



UNIVERSIDAD DE JAÉN
ESCUELA POLITÉCNICA SUPERIOR
DE JAÉN DEPARTAMENTO DE
INGENIERÍA ELECTRÓNICA Y
AUTOMÁTICA

TESIS DOCTORAL

INTEGRACIÓN DE LABORATORIOS *ONLINE*
DE AUTOMÁTICA Y TELECOMUNICACIÓN EN
LOS SISTEMAS DE GESTIÓN DE
APRENDIZAJE MEDIANTE SCORM

PRESENTADA POR:
ILDEFONSO RUANO RUANO

DIRIGIDA POR:
DR. D. JUAN GÓMEZ ORTEGA
DR. D. JAVIER GÁMEZ GARCÍA

JAÉN, 23 DE NOVIEMBRE DE 2016

ISBN 978-84-9159-052-1

Tesis Doctoral

INTEGRACIÓN DE LABORATORIOS *ONLINE* DE AUTOMÁTICA Y
TELECOMUNICACIÓN EN LOS SISTEMAS DE GESTIÓN DE
APRENDIZAJE MEDIANTE SCORM

Autor:

D. Ildfonso Ruano Ruano

Directores:

Dr. D. Juan Gómez Ortega
Dr. D. Javier Gámez García

Programa de Doctorado en Tecnologías de la Información y la Comunicación (R.D. 99/2011)
Escuela Politécnica Superior de Jaén
Universidad de Jaén

2016

Tesis Doctoral: Integración de Laboratorios *Online* de Automática y Telecomunicación en los Sistemas de Gestión de Aprendizaje mediante SCORM

Autor: D. Ildefonso Ruano Ruano

Directores: Dr. D. Juan Gómez Ortega
Dr. D. Javier Gámez García

El tribunal compuesto por los siguientes doctores:

Presidenta/e: Sebastián Dormido Bencomo

Vocal: Sergio Fabio Gómez-Estern Aguilar

Secretaria: Elisabet Estévez Estévez

Acuerdan otorgarle la calificación de:

Elisabet Estévez Estévez
La secretaria del Tribunal

Fecha: 23/11/2016

“Me lo contaron y lo olvidé; lo vi y lo entendí; lo hice y lo aprendí”.
Confucio (551 AC-478 AC)

“Yo no enseño a mis alumnos, solo les proporciono las condiciones en las que puedan aprender”.
Albert Einstein (1879-1955)

Índice de Contenidos

Índice de Contenidos.....	iii
Resumen.....	vii
Agradecimientos	xi
Lista de Figuras.....	xiii
Lista de Tablas	xix
Abreviaciones y Acrónimos	xxi
Símbolos	xxv
1. Introducción.....	1
1.1. Contextualización	3
1.2. Objetivos del Trabajo de Tesis.....	9
1.3. Descripción del Trabajo y Metodologías Utilizadas	11
1.3.1. Fase 1. Iniciación, Estudio y Obtención de Prototipos	11
1.3.2. Fase 2. Análisis, Herramientas y Metodología	14
1.3.3. Fase 3. Desarrollo de WebLabs, Depuración y Difusión	17
1.3.1. Fase 4. Ajuste, Reestructuración y Finalización.....	18
1.4. Contribuciones de la Tesis.....	22
1.4.1. Contribuciones Conceptuales.....	22
1.4.2. Contribuciones en Congresos y Revistas	23
1.4.3. Tabla-Resumen de Contribuciones-Comunicaciones	25
1.5. Estructura del Documento	26
2. Estado del Arte y Tecnológico.....	29
2.1. <i>E-learning</i> y Sistemas de Gestión de Aprendizaje	31
2.1.1. Sistemas de Gestión de Aprendizaje (LMSs).....	33
2.1.1. Estándares de <i>e-learning</i>	37

2.1.2.	SCORM.....	47
2.2.	Laboratorios Docentes <i>Online</i>	64
2.2.1.	Tipos de Laboratorios <i>online</i> y Terminología	64
2.2.2.	Tecnologías de Desarrollo y Comunicación.....	69
2.2.3.	Iniciativas, Arquitecturas, Redes y Plataformas	70
2.2.4.	Evaluación de la Efectividad del Aprendizaje.....	75
2.2.5.	Revisiones y Tendencias.....	76
2.2.6.	Otras Líneas de Trabajo.....	77
2.3.	Integración de Laboratorios <i>online</i> y LMSs.....	79
2.3.1.	Evolución de Integración de Laboratorios <i>Online</i> y LMSs	79
2.3.2.	Integración Lab-LMS y SCORM.....	86
3.	Integración Lab-LMS.....	91
3.1.	Objetivos de la Integración Lab-LMS	95
3.2.	Propuesta de Modos de Integración Lab-LMS.....	98
3.3.	Paradigma de Integración Lab-LMS Mediante SCORM	104
4.	Herramientas y Utilidades Desarrolladas	111
4.1.	Java Frente a JavaScript	114
4.2.	Comunicaciones SCO-LMS	115
4.3.	Herramienta Soporte de Comunicaciones Java: scormRTE	117
4.3.1.	Método para la Invocación de Funciones JavaScript	122
4.3.2.	Métodos Constructores	122
4.3.3.	Métodos de Sesión SCORM.....	122
4.3.4.	Métodos de Recuperación de Datos del LMS	123
4.3.5.	Métodos de Almacenamiento de Datos en el LMS.....	124
4.3.6.	Métodos de Soporte para las Comunicaciones.....	126
4.3.7.	Métodos de Ayuda en Comunicaciones y Formato de Datos ...	126
4.3.8.	Requisitos, Consejos y Ejemplos de Uso	126
4.4.	Herramientas de Soporte para JavaScript	132
4.4.1.	Librería JavaScript <i>RTE.js</i> para Comunicaciones SCO-LMS.....	133

4.4.2. Librerías JavaScript para Creación de <i>Tests</i> Automatizados	137
4.5. Otras Utilidades.....	144
4.5.1. Paquetes SCORM de Comprobación	145
4.5.2. Plantillas de Paquetes SCORM.....	147
4.5.3. Documentos de Ayuda	150
5. Metodología Desarrollada para la Creación de WebLabs.....	153
5.1. Introducción a la Metodología	155
5.2. Fases de la Metodología.....	155
5.2.1. Fase A. Objetivos/Competencias.....	157
5.2.2. Fase B. Diseño de Experimentos.....	157
5.2.3. Fase C. Decisiones Técnicas	157
5.2.4. Fase D. VRL	158
5.2.5. Fase E. Diseño del Plan de Aprendizaje	161
5.2.6. Fase F: Recursos Asociados	163
5.2.7. Fase G: Implementación	167
5.2.8. Fase H: Comprobación y Depuración	168
5.1.	168
5.2.	168
5.3. Características de la Metodología.....	168
6. Aplicaciones y Laboratorios Desarrollados	171
6.1. Aplicaciones EJS.....	173
6.1.1. Simulación de Sistema Intercambiador de Calor (SIC)	173
6.1.2. Simulación de Sistema de Péndulo Invertido en Carril (SPIC) ..	176
6.2. Laboratorios Virtuales	179
6.2.1. Sistema de Suspensión de la rueda de un coche.....	180
6.2.2. Modelado de Sistemas Dinámicos: Motor CC	188
6.2.3. Control PID de un Motor CC	206
6.3. Laboratorios Remotos	228
6.3.1. Estructura y Arquitectura de Comunicaciones	229

6.3.2. Modelado del Sistema Dinámico de un Motor CC	232
6.3.3. Comunicaciones SIP	237
6.4. Laboratorios Híbridos	251
6.4.1. Control de Temperatura sobre un Invernadero Remoto	253
6.4.2. Fotosíntesis	255
7. Conclusiones y Líneas de Futuro	259
7.1. Conclusiones	261
7.2. Líneas de Futuro.....	264
Referencias	267

Resumen

Las prácticas de laboratorio constituyen un recurso docente muy importante en la docencia universitaria. En los estudios universitarios de ciencias y tecnologías la importancia de los laboratorios es aún mayor, ya que sus actividades suelen ser eminentemente prácticas. De hecho, el trabajo de laboratorio ayuda a los alumnos a entender la realidad a la que se deben enfrentar, estableciendo un nexo de unión entre el mundo teórico docente y el mundo real. En los laboratorios docentes se puede observar, desde el punto de vista tecnológico, una evolución. A partir de los laboratorios presenciales, en los que se trabaja de forma presencial con recursos reales, han surgido (1) laboratorios virtuales en los que se trabaja con simulaciones de otros recursos, (2) laboratorios remotos en los que se trabaja a distancia con recursos reales e incluso (3) laboratorios híbridos donde se puede trabajar con recursos reales y simulaciones. Esta evolución se ha producido, en parte, gracias a Internet, permitiendo crear laboratorios *online* (virtuales, remotos o híbridos) que cuando se ofrecen a los alumnos utilizando la tecnología Web también se conocen con el término WebLabs.

Actualmente, la educación a distancia, electrónica, digital o *e-learning* es ya una realidad en todas las universidades modernas, ya sea de forma totalmente virtual o en forma de apoyo a la educación presencial (*blended learning*). Para ello, el recurso tecnológico más importante que ha impulsado esta nueva herramienta son los Sistemas de Gestión de Aprendizaje, más conocidos por sus siglas en inglés como LMS (*Learning Management System*). Un LMS es *software*, que se presenta como una plataforma Web, y que permite administrar, documentar, monitorizar, distribuir y controlar actividades de formación y cursos en una institución u organismo.

Durante un tiempo los laboratorios *online* y los LMS han convivido de forma independiente; sin embargo, su convergencia era un hecho inevitable que se empezó a implementar recientemente. La simbiosis WebLab-LMS se puede producir en diferentes niveles. Sin embargo, todos ellos conllevan una serie de ventajas entre las que destacan el hecho de ofrecer contenido docente en un marco conocido por el alumno y permitir la identificación de los usuarios que acceden al WebLab. El nivel más bajo parte de la inclusión de un WebLab en un LMS, sin que se produzca ningún tipo de interacción entre ambos, estaría la posibilidad de enlazar un laboratorio *online* en un LMS; En el otro extremo se encuentra la integración avanzada WebLab-LMS, para que se produzca este último caso es necesario que se establezca una comunicación entre ambos, permitiendo todo tipo de interacciones. En este caso la lista de ventajas aumenta, por ejemplo, facilitando la personalización de los experimentos basada en la identificación de los usuarios y mostrando el resultado de evaluaciones en el LMS a alumnos y tutores. En este trabajo se analizan los modos de integración Laboratorio-LMS y se trabaja en un modo de integración avanzado basado en estándares de *e-learning*.

Esta tesis trata de facilitar la integración de WebLabs de cualquier tipo, remotos, virtuales o híbridos, en los LMSs en un nivel avanzado haciendo uso de SCORM (*Shared Content Object Reference Model*). SCORM es un conjunto de estándares de contenido de *e-learning* que define especificaciones para la agregación de contenidos, para la presentación dinámica de contenidos y para establecer comunicaciones entre los contenidos y el LMS. Es el estándar de facto de contenidos de *e-learning*, ya que es soportado por la mayoría de los LMSs del mercado. Además, el uso de WebLabs en formato SCORM añade una ventaja muy importante: la posibilidad de reutilizar un mismo WebLab en cualquier LMS que soporte SCORM.

En este trabajo se ha definido una metodología de creación de WebLabs que tiene en cuenta aspectos docentes y pedagógicos con el fin de obtener WebLabs efectivos y de calidad. Para ello se ha utilizado una estructura de contenidos SCORM que incluye, además del propio

laboratorio *online*, una serie de contenidos de apoyo. Entre estos, destacan los contenidos informativos y generalistas, teóricos, de ayuda a las prácticas y de evaluación automatizada. También se han desarrollado una serie de herramientas que facilitan la programación de las páginas que forman el WebLab y el *software* de laboratorio. Concretamente ayudan al desarrollo de las comunicaciones con el LMS, el manejo de los datos del modelo SCORM y la creación de pruebas de evaluación automatizadas. Estas herramientas se pueden aplicar a *software* de laboratorio desarrollado con tecnología Java, como por ejemplo EJS (*Easy Java Simulations*), y también con JavaScript.

Para la validación experimental de esta propuesta se han creado diversos WebLabs virtuales, remotos e híbridos. En el proceso de desarrollo de los mismos se ha seguido dicha metodología y se han utilizado las herramientas que facilitan su creación, incluyéndose finalmente en módulos SCORM. Algunos de estos laboratorios han servido para que alumnos de los Grados de Ingeniería Industrial, Ingeniería Mecánica, Ingeniería Eléctrica, Ingeniería en Organización Industrial e Ingeniería Electrónica, todas ellas titulaciones de la Escuela Politécnica Superior de Jaén y alumnos de los Grados de Ingeniería Telemática y de Ingeniería de Telecomunicaciones de la Escuela Politécnica Superior de Linares realicen prácticas en laboratorios *online*. Se ha analizado la eficiencia de los WebLabs ofrecidos a los estudiantes utilizando distintos métodos: resultados conseguidos por los estudiantes, nivel de satisfacción de los mismos y análisis objetivo de las características técnicas de los WebLabs. Los resultados obtenidos han sido siempre muy positivos, aun así se han analizado para poder corregir errores y mejorar el proceso de creación de nuevos laboratorios, permitiendo contrastar mejoras en las últimas versiones de los laboratorios implementados.

Agradecimientos

A toda mi familia, especialmente a Mari Carmen, Alonso y Alberto.

A todos los amigos que me han ayudado a realizar este “trabajo”: directores de tesis, innovadores liantes, miembros de GRAV, y a mis amigos de la UNED, Sebastián y Luis.

Lista de Figuras

Figura 1. Laboratorios Presenciales de la Escuela Politécnica Superior de Jaén	3
Figura 2: Ventajas de los Laboratorios online	5
Figura 3. Esquema Genérico de un Laboratorio Virtual y un Laboratorio Remoto	6
Figura 4. Consorcios, Redes y Plataformas de Laboratorios online docentes más importantes	7
Figura 5: Uso de Plataforma de Docencia Virtual (PDV) por PDI	8
Figura 6. Algunos de los LMSs que soportan los estándares SCORM	9
Figura 7. Herramientas de búsqueda y gestión bibliográfica utilizadas	12
Figura 8. Logotipo de ADL	13
Figura 9. Software EJS/EJSs	13
Figura 10. Logo del LMS ILIAS	13
Figura 11. Modelo básico obtenido en la fase 1	14
Figura 12. Ejemplo de secuencia de Aprendizaje que incluye un VRL	15
Figura 13. Ejemplo de modelo SCORM de 4 páginas para VRL Java y JavaScript (JS)	16
Figura 14. Ejemplo de Servidor y equipo real de WebLab remoto: Motor CC	18
Figura 15. VL de sistema de suspensión de rueda de coche	19
Figura 16. Infraestructura para ubicación y protección de servidores y equipos de laboratorios remotos de la UJA	20
Figura 17. Objetivos y fases en las que se han desarrollado	21
Figura 18. Indicadores de descripción del objetivo 1.2 (Fte: UNIVERSITIC 2015 [13])	32
Figura 19. Porcentajes de buenas prácticas relacionadas con docencia virtual destacadas (Fte: UNIVERSITIC 2015 [13])	32
Figura 20. Uso del LMS institucional (Fte: UNIVERSITIC 2015 [13])	33
Figura 21. Tendencia de búsqueda global del término LMS en Internet (2004-Oct2016)	36
Figura 22. Estadísticas de Moodle (Gráfico de elaboración propia a partir de datos tomados de Moodle.net en Agosto de 2016)	36
Figura 23. Instalaciones ILIAS en el mundo por tipo de usuario	37
Figura 24. Estadísticas de uso diario y anual del LMS ILIAS principal de la UJA (22-9-2016)	38
Figura 25. Esquema de funcionamiento de xAPI	44
Figura 26. Esquema de funcionamiento de LTI	46
Figura 27. Evolución de las versiones de SCORM	48
Figura 28. Ejemplos de Assets	49
Figura 29. Ejemplo de SCO	50
Figura 30. Ejemplo de Agregación	50

Figura 31. Ejemplo de Paquete de contenido SCORM	50
Figura 32. Captura de pantalla parcial de ejemplo de fichero imsmanifest.xml	52
Figura 33. Modelo temporal del RTE SCORM	60
Figura 34. Tipos de Laboratorios online	66
Figura 35. Arquitectura del Sistema Gateway4Labs	89
Figura 36. Comunicaciones LMS-Laboratorio	92
Figura 37. Esquemas de modos de integración 0 a 2	101
Figura 38. Esquemas de modos de Integración 3 a 5 (avanzados)	103
Figura 39. Permisos del recurso WebLab (paquete SCORM) en el LMS ILIAS de la UJA	106
Figura 40. Planificación temporal del recurso WebLab (paquete SCORM) en el LMS ILIAS de la UJA	107
Figura 41. Precondiciones del recurso WebLab (paquete SCORM) en el LMS ILIAS de la UJA	107
Figura 42. Ejemplo de esquema del modo de integración 3 desarrollado para la UJA	108
Figura 43. Captura parcial del DOM de página Web de un SCORM 2004 en ejecución (Objeto API)	116
Figura 44. Ejemplo de ejecución de Paquete SCORM con 2 SCO y funciones JavaScript	117
Figura 45. Ejemplo de ejecución de Paquete SCORM con 1 SCO y un applet Java que importa el paquete scormRTE	120
Figura 46. Ejemplo de diagrama de secuencia UML con entidades y métodos relacionados	121
Figura 47. Pestaña "Opciones de Ejecución" de EJS	127
Figura 48. Vista de página de variables en EJS	127
Figura 49. Vista del código de Inicialización en EJS	128
Figura 50. Ventajas de la firma de applets Java	132
Figura 51. Ejemplo de mensaje de petición de Ejecución de applet Java	133
Figura 52. RTE.js como capa intermedia de comunicaciones RTE SCORM con el LMS	134
Figura 53. Ejemplo de código JavaScript embebido en una página HTML SCORM usando la librería RTE.js para obtener el nombre del estudiante del LMS	136
Figura 54. Diagrama de secuencia UML de un ejemplo de sesión de comunicación SCORM con entidades y funciones relacionadas	137
Figura 55. Esquema de comunicaciones y librerías usadas en la generación automatizada de tests	140
Figura 56. Ejemplo de fichero QTI de definición de preguntas de test	141
Figura 57. Captura de pantalla de un segmento de test generado automáticamente	144
Figura 58. Captura de pantalla de las constantes de sesión en SCORM1.2	146
Figura 59. Captura de pantalla de las variables de sólo lectura y de lectura/escritura de SCORM1.2	147
Figura 60. Comentarios, interacciones, Objetivos, Errores/Diagnóstico y Terminación de sesión	147
Figura 61. Estructuras de directorios de los modelos SCORM	148

Figura 62. Ejemplo de SCO y estructura de carpetas de paquete SCORM1.2 _____	149
Figura 63. Captura parcial de la Tabla-resumen del paquete Java scormRTE_____	151
Figura 64. Captura parcial de Tabla-Resumen de variables de librería RTE.js _____	151
Figura 65. Captura parcial de Tabla-resumen de funciones de librería RTE.js _____	151
Figura 66. Captura parcial de Tabla-Resumen de variables y funciones de librería RTE.js _____	152
Figura 67. Metodología de creación de laboratorios online: Fases, actores y resultados obtenidos en cada fase_____	156
Figura 68. Ciclo de Aprendizaje Experimental de Kolb_____	162
Figura 69. Sistema Intercambiador de Calor (SIC) _____	173
Figura 70. Esquema del proceso termal_____	174
Figura 71. GUI de la simulación EJS del SIC _____	176
Figura 72. Sistema de Péndulo Invertido en Carro Lineal (SPIC) _____	177
Figura 73. Detalle del carro con péndulos, variables, constantes y ecuaciones del SPIC_____	177
Figura 74. Ventana principal de la GUI de la simulación del SPIC_____	178
Figura 75. Ventana de gráficos del SPICL_____	179
Figura 76. Ventana de parámetros del SPICL_____	179
Figura 77. Estructura de Lab "Sistema de suspensión de la rueda de un coche" y comunicaciones LMS_____	181
Figura 78. Captura de pantalla del SCO1 con la teoría específica del laboratorio _____	181
Figura 79. Captura de pantalla parcial de SCO2. Guiones de prácticas_____	182
Figura 80. Sistema real, esquema del sistema de suspensión y esquema general del sistema _____	182
Figura 81. Captura de pantalla del applet Java del VL del sistema de suspensión de una rueda_____	183
Figura 82. Captura de pantalla de la ventana emergente de gráficos del applet_____	185
Figura 83. Captura de pantalla parcial de SCO3. Evaluación final de prácticas _____	186
Figura 84. Ejemplo de vista del progreso de aprendizaje ofrecida a un estudiante sobre su trabajo _____	187
Figura 85. Ejemplo de vista de profesor del informe básico de capítulos SCORM en ILIAS _____	187
Figura 86. Vista del progreso de aprendizaje de los usuarios del laboratorio online para un profesor _____	188
Figura 87. Unidad mecánica con Motor CC 33-100 de Feedback Instrument _____	189
Figura 88. Estructura de Lab "Modelado de Sistemas Dinámicos: MotorCC" y comunicaciones LMS_____	190
Figura 89. Captura de pantalla del SCO1 con la información genérica y estructura del laboratorio _____	191
Figura 90. Captura de pantalla parcial de la teoría presentada en el SCO2 _____	192
Figura 91. Captura de pantalla parcial del test pre-VL presentado en el SCO2_____	193

Figura 92. Captura de pantalla parcial de los guiones de prácticas presentados en el SCO3 _____	193
Figura 93. applet Java del VL con práctica 1ª y ventana emergente de paso a la 2ª __	194
Figura 94. Esquema del sistema y ecuación de transferencia del sistema de Motor CC simplificado _____	194
Figura 95. applet Java del VL con práctica 2ª y ventana emergente de paso al SCO3 _	195
Figura 96. Captura de pantalla parcial del SCO4 con el test de evaluación post-VL ___	196
Figura 97. Secuenciación y Navegación del paquete SCORM del Laboratorio Virtual del Motor CC _____	196
Figura 98. Captura de pantalla de carpeta virtual en ILIAS tipo A _____	198
Figura 99. Contenidos del grupo B y secuencia de aprendizaje y navegación completa (incluyendo recursos del LMS y en el interior del paquete SCORM) _____	200
Figura 100. Mejora producida en las calificaciones _____	203
Figura 101. Opinión de alumnado: Adecuación de contenidos del WebLab (SCORM) _	204
Figura 102. Opinión de alumnado: Valoración de aspectos del WebLab (SCORM) ____	205
Figura 103. Opinión de alumnado: Valoración general final del WebLab (SCORM) ___	205
Figura 104. Captura de pantalla de página 1 del WebLab Control PID de Motor CC: Introducción _____	209
Figura 105. Captura de pantalla parcial de página 2 del WebLab de PID de Motor CC (Teoría) _____	209
Figura 106. Captura de pantalla parcial de página 2 del WebLab de PID de Motor CC (Test pre-VL) _____	210
Figura 107. Ventana emergente de aviso de puntuación anterior en VL _____	211
Figura 108. Captura de pantalla parcial de página 3 del WebLab de PID de Motor CC (Intro) _____	211
Figura 109. Captura de pantalla parcial de página 3 del WebLab de PID de Motor CC (Guiones) _____	212
Figura 110. Modelos del Sistema de control del WebLab de PID de Motor CC _____	213
Figura 111. applet Java del VL con práctica 1ª de control P y ventana emergente de paso a la de Control PD _____	214
Figura 112. applet Java del VL con práctica 2ª de control PD y ventana emergente de paso a la de Control PI _____	215
Figura 113. applet Java del VL con práctica 3ª de control PI y ventana emergente de paso al SCO4 _____	215
Figura 114. Ventana de comentarios del VL de Control PID de Motor CC _____	216
Figura 115. Captura de pantalla parcial de página 4 del WebLab de PID de Motor CC: Evaluación _____	217
Figura 116. Captura de pantalla de grupo en ILIAS para WebLab de 3 páginas _____	218
Figura 117. Resumen de resultados del GC _____	221
Figura 118. Resumen de resultados del GE _____	221
Figura 119. Resumen de resultados GC-GE y Examen final _____	222

Figura 120. Opinión del alumnado del WebLab de Control PID de Motor CC (diversos aspectos)	225
Figura 121. Opinión del alumnado del WebLab de Control PID de Motor CC (Valoración Final)	226
Figura 122. Captura de pantalla de la ventana principal del WebLab en UNILabs	227
Figura 123. Captura de pantalla de la ejecución del WebLab en UNILabs (Página 3. Paso de experimento 2 (Control PD) a 3 (Control PI)	228
Figura 124. Infraestructura de Laboratorios Remotos	230
Figura 125. Switch (Conmutador) NORTEL BayStack 5510-24T	230
Figura 126. Esquema de comunicaciones para laboratorios remotos de la UJA	231
Figura 127. Arquitectura y Sistema de comunicaciones del Laboratorio remoto de modelado de Motor CC	232
Figura 128. Interfaz de usuario administrador del software de servidor	233
Figura 129. Equipos del laboratorio remoto en las instalaciones	235
Figura 130. Estructura y comunicaciones de Lab remoto Modelado del Sistema Dinámico de un Motor CC	235
Figura 131. Interfaz de usuario de aplicación applet Java de RL	236
Figura 132. Estructura y comunicaciones de Lab remoto SIP	239
Figura 133. Captura de pantalla del SCO1 con la información genérica y estructura del laboratorio remoto SIP	240
Figura 134. Captura de pantalla parcial del SCO2 del laboratorio remoto SIP (Teoría).	241
Figura 135. Captura de pantalla parcial del SCO2 del laboratorio remoto SIP (Test pre-RL)	241
Figura 136. Captura de pantalla parcial del SCO3 del laboratorio remoto SIP (Introducción a las prácticas)	242
Figura 137. Interfaz de usuario del applet Java del servidor proxy SIP	243
Figura 138. Interfaz de usuario del applet Java del cliente SIP (softphone)	243
Figura 139. Captura de pantalla de espacio virtual del WebLab SIP	244
Figura 140. Datos de uso y resultados obtenidos por estudiantes del WebLab "Comunicaciones SIP"	245
Figura 141. Opinión de alumnado: Adecuación de contenidos del WebLab SIP (SCORM)	247
Figura 142. Opinión del alumnado del WebLab de Comunicaciones SIP (diversos aspectos)	248
Figura 143. Opinión del alumnado: Nivel de satisfacción final del WebLab de Comunicaciones SIP	248
Figura 144. Estructura y comunicaciones de Lab remoto SIP JavaScript	249
Figura 145. Captura parcial de pantalla de SCO3 del WebLab de comunicaciones SIP JS	250
Figura 146. Interfaz de cliente SIP JavaScript con vista de mensajes SIP completa	251

Figura 147. Equipos remotos de laboratorios híbridos en balda superior de infraestructura _____	252
Figura 148. Arquitectura y Sistema de comunicaciones de los Laboratorios híbridos _	252
Figura 149. Estructura y comunicaciones del laboratorio híbrido de control PID de temperatura _____	253
Figura 150. Interfaz de VL en forma de applet Java para el control PID de la temperatura _____	254
Figura 151. Interfaz de RL en forma de applet Java para el control remoto de actuadores y visión de valores en los sensores del invernadero _____	255
Figura 152. Estructura y comunicaciones del laboratorio híbrido de fotosíntesis ____	256
Figura 153. Interfaz de VL en forma de applet Java para fotosíntesis _____	256
Figura 154. Interfaz de RL en forma de applet Java para el control remoto de actuadores y visión de gráficas de parámetros en WebLab Fotosíntesis _____	257

Lista de Tablas

Tabla 1. <i>Comunicaciones en congresos/revistas y contribuciones conceptuales</i>	26
Tabla 2. <i>Eventos de Navegación SCORM</i>	57
Tabla 3. <i>Categorías y Rangos de Códigos de Error de la API SCORM</i>	62
Tabla 4. <i>Elementos del modelo de datos RTE</i>	63
Tabla 5: <i>Modos de Integración Lab-LMS y características de cada uno</i>	100
Tabla 6. <i>Elementos de la función RTE</i>	134
Tabla 7. <i>Lista de variables del sistema</i>	174
Tabla 8. <i>Variables y constantes del SPIC</i>	178
Tabla 9. <i>Parámetros del Sistema de suspensión.</i>	183
Tabla 10. <i>Resumen de uso y resultados obtenidos en el VL de Modelado de Motor CC 201</i>	
Tabla 11. <i>Función de los contenidos del WebLab modelado de Motor CC</i>	204
Tabla 12. <i>Evaluación de WebLab: Método 2</i>	206

Abreviaciones y Acrónimos

ADL: *Advanced Distributed Learning* (Aprendizaje Distribuido Avanzado).

AENOR: Asociación Española para la Normalización y Certificación

AICC: *Aviation Industry Computer-Based Training Committee* (Comité de Formación basada en Ordenador de la Industria de la Aviación).

AJAX: *Asynchronous JavaScript And XML* (JavaScript Asíncrono y XML).

API: *Application Program Interface* (Interfaz del Programa Aplicación).

ARL: *Augmented Remote Lab* (Laboratorio Remoto de Realidad Aumentada)

BTH: Instituto de Tecnología de Blekinge (*Blekinge Institute of Technology*, de Suecia).

CAM: *Content Agregation Model* (Modelo de Agregación de Contenidos).

CEA: Comité Español de Automática.

CEN: Comité Europeo de Normalización

CMI: *Computer Managed Instruction* (Formación Gestionada por Ordenador).

CORDRA: *Content Object Repository Discovery and Registration/Resolution Architecture*.

CRUE: Conferencia de Rectores de Universidad de España.

CSS: *Cascade Style Sheet* (Hoja de Estilo en Cascada).

DCL: *Distributed Control Lab* (Laboratorio de Control Distribuido).

DCMI: *Dublin Core Metadata Initiative*.

DIA: Departamento de Informática y Automática (UNED).

DIBE: *Department of Biophysical and Electronic Engineering* (Departamento de Ingeniería Electrónica y Biofísica).

DNS: *Domain Name System* (Sistema de Nombres de Dominio).

DoD: *Department of Defense* (Departamento de Defensa de los EE.UU.).

DOM: *Document Object Model* (Modelo de Objetos del Documento).

EDNA: *Education Network Australia*.

EFQM: Fundación Europea para la Gestión de la Calidad.

EFQUEL: Fundación Europea para la Calidad en el *E-learning*.

EJS: *Easy Java Simulations*.

EJS: *Easy Java/JavaScript Simulations*.

EPSJ: Escuela Politécnica Superior de Jaén.

EPSL: Escuela Politécnica Superior de Linares.

Go-Lab: *Global Online Science Labs for Inquiry Learning at School* (Laboratorios online de Ciencia globales para aprendizaje de investigación en la escuela).

GOLC: *Global Online Laboratory Consortium* (Consortio Mundial de Laboratorios Online).

GUI: *Graphic User Interface* (Interfase Gráfica de Usuario).

HPI: *Hasso-Plattner-Institut*.

IEEE/LTSC: *Institute of Electrical and Electronic Engineers/Learning Technologies Standardization Committee*.

ICT: Ver TIC.

IEEE: *Institute of Electrical and Electronics Engineers*.

IMS: *Instructional Management System* (Sistema de Gestión Didáctico).

IMS Global o GLC: *IMS Global Learning Consortium* (Consortio de Aprendizaje Mundial IMS).

ISA: *iLab Shared Architecture* (Arquitectura compartida iLab)

ISILab: *Internet Shared Instrumentation Laboratory* (Laboratorio de instrumentación compartido en Internet).

ISO/EIC: *International Standards Organisation/International Electrotechnical Commission*.

JCR: *Journal Citation Report*.

JVM: *Java Virtual Machine* (Máquina Virtual de Java).

LaaS: *Laboratory as a Service* (Laboratorio como Servicio).

LAN: *Local Area Network* (Red de Área Local).

LiLa: *Library of Labs* (Librería de laboratorios).

LMS: *Learning Management System* (Sistema de Gestión de Aprendizaje).

LO: *Learning Object* (Objeto de Aprendizaje).

LLO: *Lab Learning Object* (Objeto de Aprendizaje tipo Laboratorio).

LoL: *Laboratorio online*.

LRS: *Learning Record Store* (Almacén de Registros de Aprendizaje).

LTI: *Learning Tool Interoperability* (Interoperabilidad entre Herramientas de Aprendizaje).

MIT: *Massachusetts Institute of Technology* (Instituto Tecnológico de Massachusetts, de EE.UU.)

Lab: Laboratorio.

LORN: *Learning Object Repositories Network*.

OAI: *Open Archives Initiative*.

OCELOT: *Open and Collaborative Environment for the Leverage of Online instrumentation* (Entorno abierto y colaborativo para el aprovechamiento de la instrumentación en línea).

- ODRL:** *Open Digital Rights Language*.
- OKI:** *Open Knowledge Initiative*.
- OMA:** *Open Mobile Alliance*.
- OS:** Ver SO.
- OSL:** *Open Source license*, (Licencia de Código Abierto).
- OSM:** *Operating Systems and Middleware*.
- PDI:** Personal Docente e Investigador.
- PDV:** Plataforma de Docencia Virtual.
- PID:** Proyecto de Innovación Docente.
- PIF:** *Packet Interchange File*.
- PTZ:** *Pan Tilt Zoom* (Tipo de cámara con rotación en plazo horizontal, vertical y zoom)
- QTI:** *Question & Test Interoperability* (Interoperabilidad de Tests y Preguntas).
- RLMS:** *Remote Laboratory Management System*, Sistema de Gestión de laboratorios Remotos
- RTE:** *Run-Time Environment* (Entorno de Tiempo de Ejecución).
- SCI:** Servicios Centrales de Informática.
- SCO:** *Shared Content Object* (Objeto de Contenido Compartido).
- SCORM:** *Shared Content Object Reference Model* (Modelo de Referencia de Objetos de Contenido Compartidos).
- SD:** *Standard Deviation* (Desviación Estándar).
- SIC:** Sistema Intercambiador de Calor (*HES, Heat Exchanger System*).
- SIF:** *Systems Interoperability Framework*.
- SIP:** *Session Initiation Protocol* (Protocolo de Inicio de Sesiones).
- SN:** *Sequencing and Navigation* (Secuenciación y Navegación).
- SO:** Sistema operativo (*OS, Operative System*).
- SPIC:** Sistema de Péndulo Invertido sobre Carril Lineal.
- STA:** *Science, Technology and Engineering* (Ciencias, tecnología e ingeniería).
- STAM:** *Science, Technology, Engineering and Mathematics* (Ciencias, tecnología, ingeniería y matemáticas).
- TCAPI:** Tin Can API, también conocido como xAPI.
- TCP/IP:** *Transport Control Protocol/Internet Protocol* (Protocolo de Control de Transporte/Protocolo Internet).
- TI:** Tecnologías de la Información.
- TIC:** Tecnologías de la Información y las Comunicaciones (*ICT, Information and Communication Technologies* en inglés).
- UI:** *User Interface* (Interfaz de usuario).

UJA: Universidad de Jaén.

UNED: Universidad Nacional de Educación a Distancia.

VCL: *Virtual Computer Laboratory* (Laboratorio de Computación Virtual).

VISIR: *Virtual Instrument Systems in Reality*.

VL: *Virtual Laboratory* (Laboratorio Virtual).

VLE: *Virtual Learning Environment*.

VNC: *Virtual Network Computer*, (Ordenador Virtual en Red).

VoIP: *Voice over IP* (Voz sobre IP).

VR: *Remote Laboratory* (Laboratorio Remoto).

VRL: *Virtual and/or Remote Laboratory* (Laboratorio Remoto y/o Virtual).

W3C: *World Wide Web Consortium* (Consortio WWW).

WebLab: *Web Laboratory* (Laboratorio Web).

xAPI: *eXperience API*, también conocido como TCAP

XML: *eXtensible Markup Language*

Símbolos

Sistema Intercambiador de Calor (SIC) - 6.1.1:

T_1 : Temperatura de entrada del circuito primario del SIC.
 T_2 : Temperatura de salida del circuito primario del SIC.
 T_3 : Temperatura de entrada del circuito secundario del SIC.
 T_4 : Temperatura de salida del enfriador.
 T_{env} : Temperatura del entorno.
 m_p, m_s : Capacitancia (Líquido en circuito 1).
 m_{ip}, m_{is} : Masas de agua en circuitos primario y secundario del SIC.
 c_p : Calor específico del agua.
 \dot{Q}_i : Tasa de flujo de calor del intercambiador.
 \dot{Q}_c : Transferencia de calor (Calentador).
 \dot{Q}_v : Transferencia de calor disipada del intercambiador.
 AU_j : Coeficiente de transferencia de calor del intercambiador.
 AU_f : Coeficiente de transferencia de calor del enfriador.

Sistema de Péndulo Invertido en Carril (SPIC) - 6.1.2

g : Aceleración de la gravedad.
 l : longitud del palo del péndulo.
 M : masa del carro.
 I : Momento de inercia del carro.
 m : Masa del palo (del péndulo).
 d : Coeficiente de amortiguamiento del péndulo.
 F : Fuerza ejercida en carro.
 x : Posición del carro.
 u : Señal de control (tensión).
 θ : Ángulo del péndulo.
 k_{Fu} : Ganancia entre la derivada de la tensión de entrada y la fuerza.

Sistema de Suspensión de la Rueda de un Coche - 6.2.1

g : Aceleración de la gravedad (9,81 m/s²)
 L_0 : Longitud del muelle en reposo.
 m : Masa que soporta la rueda (350 Kg).
 K : Constante del muelle.
 B : Constante de amortiguamiento del muelle.
 u : Distancia del asfalto al punto de referencia.

y: Distancia de la masa al punto de referencia.

Modelado de Sistemas Dinámicos: Motor CC - 6.2.2

t = Tiempo.

U_a = Entrada al sistema.

ω = Salida del Sistema (Velocidad angular).

K = Ganancia estática del sistema.

T = Constante de tiempo del sistema.

Control PID de un Motor CC - 6.2.3

t = Tiempo.

Ref(t) = Entrada de referencia.

$e(t)$ = Señal de entrada al controlador.

$\omega(t)$ = Salida del Sistema (Velocidad angular).

$\theta(t)$ = Salida del Sistema (Posición angular).

K_m = Ganancia estática del sistema.

T_m = Constante de tiempo del sistema.

K_p = Constante proporcional del controlador.

K_D = Constante derivativa del controlador.

K_I = Constante integral del controlador.

Cap.1

1. Introducción

- 1.1. Contextualización
 - 1.2. Objetivos del Trabajo de Tesis
 - 1.3. Descripción del Trabajo y Metodologías Utilizadas
 - 1.4. Contribuciones de la Tesis
 - 1.5. Estructura del Documento
-

Este capítulo de introducción adentra al lector en la temática que aborda este trabajo de tesis y da una visión general del trabajo realizado y los logros que lo avalan. Comienza contextualizando el trabajo realizado, explicando la justificación y motivaciones con las que se ha llevado a cabo. El segundo apartado muestra los objetivos que se han marcado y conseguido mientras que el siguiente prosigue con una breve descripción del trabajo realizado y las metodologías seguidas. Después se analiza y describe las principales

contribuciones que se han logrado tanto a nivel conceptual como en forma de publicaciones realizadas en el ámbito científico. Para finalizar el capítulo se ha incluido una breve explicación de los contenidos incluidos en el resto del documento.

1.1. Contextualización

La importancia del trabajo práctico es innegable en muchos estudios universitarios [1]. Este hecho se hace especialmente patente en la mayoría de las asignaturas de ingenierías y ciencias aplicadas [2], [3]. Tradicionalmente, esta formación se ha venido realizando en laboratorios presenciales (**Figura 1**, *hands-on laboratories* en inglés).



Figura 1. Laboratorios Presenciales de la Escuela Politécnica Superior de Jaén

Algunos autores han llegado a afirmar que las experiencias de laboratorio constituyen el corazón del aprendizaje científico [4], o que las experiencias de laboratorio hacen que la ciencia “cobre vida” [5]. El trabajo práctico constituye un pilar esencial en la formación de los estudiantes de ingeniería, que deben tener un conocimiento que vaya más allá de la propia teoría [2]. La comunidad científica presenta un acuerdo explícito sobre los beneficios que aportan los laboratorios en la enseñanza de ciencias e ingenierías [3], [6], [7], [8], [9].

Por otro lado, los profesionales de ingeniería llevan mucho tiempo siendo artífices de muchos cambios que se han producido gracias a la explotación de avances tecnológicos y sus innovaciones relacionadas [10]. La aplicación de estos avances tecnológicos en la educación, especialmente de las **Tecnologías de la Información y las Comunicaciones (TIC)**, que posibilitan la formación

online, es una línea de actuación muy importante que fomentan ampliamente organismos públicos y privados, tanto internacionalmente [11], [12] como en España [13]. Se ha demostrado que el uso de las TICs puede aportar mejoras significativas en el proceso educativo [14], hecho que ha sido percibido por los docentes [15]. Las TICs también se han aplicado a los laboratorios docentes. Esto ha hecho no sólo que cambie la forma de realizar el trabajo práctico, sino que se encuentra en un proceso de cambio continuo cuyo principal culpable lo constituye Internet y su evolución imparable [16]. El avance y la aplicación de las TICs ha traído consigo la aparición de nuevos tipos de laboratorios con los que los alumnos pueden trabajar a través de Internet en lo que se ha venido a llamar **laboratorios online** (LoL) o en línea [17]. Un factor clave que ha ayudado al desarrollo de este tipo de laboratorios ha sido, sin duda, el alto coste que supone la implantación y mantenimiento de un laboratorio presencial, principalmente el asociado al espacio físico o estancia que debe ocupar, al equipamiento que contiene y al personal de mantenimiento del mismo [18]. El ahorro del coste no es la única ventaja que aportan los laboratorios *online* respecto a los laboratorios presenciales; de hecho existen trabajos en los que se ha tratado este tema y lo han puesto en entredicho bajo ciertas circunstancias [19]. Afortunadamente existen otra serie de **beneficios** [20] (**Figura 2**) que han facilitado su desarrollo y adopción en las instituciones educativas como son:

- **Disponibilidad espacial y temporal.** Se aumenta la disponibilidad de uso de los laboratorios al posibilitar la ejecución a distancia desde cualquier ubicación (libertad espacial) y a cualquier hora (libertad temporal).
- **Seguridad.** Los laboratorios *online* aumentan la seguridad a la hora de realizar las prácticas, algunos experimentos presenciales pueden resultar peligrosos para los estudiantes y/o personal técnico de laboratorio.
- **Extensión del uso de recursos escasos.** Pudiendo justificarse la compra de un dispositivo en una institución u organismo al aumentar las posibilidades de ser utilizado gracias a su integración en forma de laboratorio *online*.
- **Accesibilidad.** Las personas discapacitadas aumentan las posibilidades de poder trabajar en este tipo de laboratorios si se programa una interfaz adecuada a sus características (por ejemplo, el uso de un ayudante virtual que muestre textos en caso de un estudiante con discapacidad auditiva).

- **Observación/Grabación.** Las sesiones de laboratorios *online* pueden ser observadas e incluso grabadas remotamente.
- **Compartición.** Los laboratorios *online* pueden ser compartidos entre diferentes instituciones.
- **Motivaciones.** A muchos estudiantes les motiva el hecho de trabajar en laboratorios *online* frente a los presenciales. Además permiten la inclusión de elementos extra que los hagan ver como “juegos”, o *gamification* en inglés [21], [22].



Figura 2: Ventajas de los Laboratorios online

Sin embargo, no todos son ventajas: algunas habilidades manuales y manejo de equipos son más difíciles de adquirir con este tipo de laboratorios [23]. Hoy día, los laboratorios *online* y los presenciales conviven en armonía, pudiendo encontrarse ambas modalidades de forma generalizada [9], [7], o incluso de forma combinada [23].

Para que los alumnos puedan trabajar en un laboratorio *online* deben hacer uso de un *software* de laboratorio a través del cual pueden interactuar con un sistema real o simulado. En el primer caso se dice que se trata de un **laboratorio remoto** o *Remote Lab* en inglés, RL, mientras que en el segundo se dice que es un **laboratorio virtual** o *Virtual Lab* en inglés, VL. También puede darse el caso en el que interactúen con sistemas reales y simulados a la vez (laboratorio híbrido o *Hybrid Lab* en inglés, HL). De forma genérica se puede referir a la aplicación de un laboratorio *online* como VRL (*Virtual/Remote Lab* en inglés). La estructura de un laboratorio *online* se muestra en la **Figura 3**.

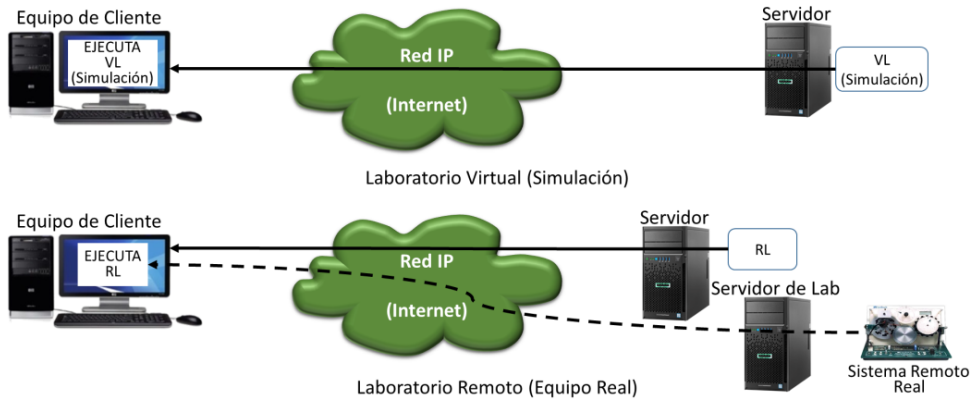


Figura 3. Esquema Genérico de un Laboratorio Virtual y un Laboratorio Remoto

Como se puede ver en la **Figura 3**, en un laboratorio virtual el usuario debe conectarse a un servidor y descargar el VL que ejecuta en su dispositivo para interactuar con la simulación. Por otro lado, el proceso equivalente en un laboratorio remoto, aunque para el usuario pueda parecer similar, resulta algo más complejo: el estudiante se descarga el RL tras conectarse a un servidor, lo ejecuta en su dispositivo para interactuar con un Sistema Remoto Real a través de un servidor de laboratorio, que es el encargado de establecer y mantener la conexión entre el Sistema Remoto Real y el RL a través de Internet. El servidor de laboratorio puede ser el mismo que usa el usuario para descargarse el *software* del RL.

Universidades y organizaciones educativas prestigiosas en todo el mundo se han dedicado a promover y compartir los laboratorios *online* que han desarrollado en **portales y/o redes nacionales e internacionales de laboratorios online** (**Figura 4**) tales como el Proyecto iLab [24] del MIT (*Massachusetts Institute of Technology*), VISIR (*Virtual Instrument Systems In Reality*) Open Lab Platform [25] del Instituto de Tecnología de Blekinge (BTH, *Blekinge Institute of Technology*) en Suecia, LiLa (*Library of Labs*) [26] que es un consorcio liderado por la Universidad de Stuttgart de Alemania, UNILabs (*University Network of Interactive Labs*) [27] de la UNED (Universidad Nacional de Educación a Distancia) en España, Lab2go [28] de la Universidad Carinthia de Ciencias Aplicadas en Austria, ISILab (*Internet Shared Instrumentation Laboratory*) [29] del DIBE (*Department of Biophysical and Electronic Engineering*) de la Universidad de Génova en Italia, NetLab [30] de la Universidad del Sur de Australia (UniSA), DCL (*Distributed Control Lab*) [31] del grupo OSM (*Operating Systems and Middleware*) del HPI (*Hasso-Plattner-Institut*) en Alemania, WebLab

Deusto [32] de la Universidad de Deusto en España, y el Proyecto Labshare (Sahara) [33] del gobierno Australiano, concluido en el 2011 pero soportado desde el 2015 por el Instituto LabShare. Se han llegado a crear consorcios internacionales como GOLC (*Global Online Laboratory Consortium*) [34], con el objetivo principal de promover el desarrollo, compartición y la investigación en laboratorios accesibles remotamente para la educación y cuyos miembros son, entre otras, las universidades más involucradas en los proyectos antes mencionados (MIT, UNED, Universidad de Deusto, Universidad de Stuttgart, BTH, ...).



Figura 4. Consorcios, Redes y Plataformas de Laboratorios online docentes más importantes

Los laboratorios *online* han proliferado en los estudios de ingeniería en mayor medida que en otras titulaciones, presentándose en diferentes modalidades e implementándose con múltiples tecnologías [35], [36], [36], [37].

Sin embargo, el mayor exponente de la aplicación de las TICs en la educación universitaria no lo constituyen los laboratorios *online*, sino que son las plataformas de docencia virtual o **Sistemas de Gestión de Aprendizaje** (*Learning Management Systems* en inglés o simplemente LMS). La docencia virtual, o *e-learning*, es hoy día posible tal como la conocemos gracias a los LMSs [38]. Los LMSs constituyen el elemento central esencial de *e-learning* [39]. Ofrecen un espacio web en el que los alumnos y profesores se encuentran, pueden comunicarse, crear, compartir y usar recursos de aprendizaje, realizar evaluaciones, encontrar enlaces a otros recursos externos, integrar aplicaciones externas y mucho más [40]. En el último informe UNIVERSITIC 2015 [13], publicado en 2016, que refleja el estado de las TICs en las universidades españolas se dice que “La práctica totalidad del profesorado y de los estudiantes utilizan la plataforma de docencia virtual institucional...” y añade que el 91,81% del personal docente e investigador (PDI) hace uso de la misma (**Figura 5**).

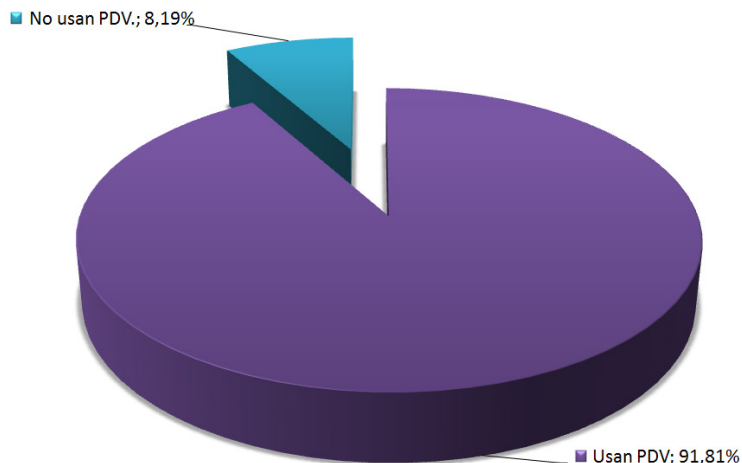


Figura 5: Uso de Plataforma de Docencia Virtual (PDV) por PDI

Los LMSs y los laboratorios *online* ofrecen servicios complementarios cuya convergencia fue señalada como una tendencia a seguir por Gravier en el 2006 [41]. Otros autores han dedicado tiempo y esfuerzo a esta tarea [42], [43], [44]. Sin embargo, la **integración avanzada de los laboratorios *online* y los LMSs** basado en estándares no es un problema de fácil solución [45], [46]. Existen muchas formas de llevarla a cabo y para un profano en la materia puede ser un tema bastante complejo. Algunas soluciones apostaron por el uso de estándares para permitir la reutilización de laboratorios en distintos LMSs [47], mientras que otros apostaron por crear soluciones dedicadas que permitieran la comunicación entre el laboratorio y el LMS [42].

Esta tesis se centra en este problema, aportando **soluciones al reto de la integración de los laboratorios *online* en los LMSs**. Propone una solución de integración avanzada con comunicación Laboratorio-LMS basada en estándares que es reutilizable. Para ello, en este trabajo, se ha analizado el problema de la integración VRL-LMS distinguiendo diferentes modalidades o niveles y que van desde una integración mínima hasta una integración avanzada [46]. Tras establecer esta clasificación se ha aportado una solución basada en los estándares SCORM (*Shared Content Object Reference Model*) [48] y el desarrollo de una serie de herramientas que facilitan la creación de laboratorios *online* con integración avanzada en LMS [45], [49], [50]. Una de las principales ventajas que presentan los laboratorios desarrollados de este modo es la posibilidad de reutilización en los LMSs que soporten los estándares SCORM (**Figura 6**), manteniendo todas sus capacidades de comunicación laboratorio-LMS [51].

Otra no menos importante es la posibilidad de incluir en el mismo formato SCORM otros recursos asociados al *software* del laboratorio que pueden ayudar a maximizar la efectividad de aprendizaje [50].



Figura 6. Algunos de los LMSs que soportan los estándares SCORM

Finalmente, se ha conseguido definir una metodología que ayuda a analizar los problemas que conlleva la creación de un laboratorio *online* identificando las fases necesarias para llevar a cabo una integración, los agentes involucrados en cada fase y los resultados que deben obtenerse después de cada una de ellas [52].

1.2. Objetivos del Trabajo de Tesis

El objetivo principal de este trabajo es la **Integración avanzada de laboratorios docentes online efectivos en LMSs basada en los estándares SCORM**. Como objetivos secundarios, que han sido necesarios para llevar a cabo el objetivo principal con garantías, se pueden enumerar los siguientes:

- Objetivo 1.** Buscar y realizar un análisis bibliográfico del estado del arte en laboratorios *online* y su integración con los LMSs.
- Objetivo 2.** Analizar las características básicas y beneficios que aportan los LMSs en la docencia práctica a través de Internet y los estándares de e-

learning, especialmente los que están más estrechamente relacionados con los laboratorios *online*.

Objetivo 3. Analizar y conseguir manejo de práctico de los estándares de *e-learning* SCORM aplicables a los laboratorios *online*.

Objetivo 4. Analizar la aplicabilidad y utilizar del *software Easy Java/JavaScript Simulations*, EJS (antes *Easy Java Simulations*, EJS) [53] para el desarrollo de VRL (virtuales y remotos) y otro *software* asociado: Desarrollar modelos/simulaciones de sistemas reales e interfaces para interaccionar con sistemas remotos reales.

Objetivo 5. Crear laboratorios *online* embebidos en paquetes de formato SCORM de prueba (virtuales y remotos).

Objetivo 6. Desarrollar una clasificación de los diferentes modos de integración de los laboratorios *online* en los LMSs con análisis de las ventajas e inconvenientes que presentan.

Objetivo 7. Analizar pedagógicamente los recursos asociados a los laboratorios *online*; crear secuencias de aprendizaje a través de estos recursos y el propio VRL en formato SCORM, y de forma combinada con recursos ubicados en un LMS.

Objetivo 8. Desarrollar recursos para simplificar y facilitar la puesta en marcha de nuevos laboratorios *online* integrados en LMS mediante SCORM:

- a. Plantillas de paquetes SCORM para las versiones 1.2 y 2004.
- b. Modelos de testeo en forma de paquete SCORM para las versiones 1.2 y 2004.
- c. Paquete Java para desarrollo simplificado de comunicaciones VRL-LMS basadas en SCORM (ver. 1.2 y 2004) en entornos de programación Java.
- d. Librerías JavaScript para desarrollo simplificado de comunicaciones y evaluaciones VRL-LMS y SCO¹-LMS en entornos de programación JavaScript.
- e. Documentos de ayuda para el uso de las librerías JavaScript y el paquete Java.

Objetivo 9. Desarrollar una metodología para la creación de laboratorios *online* efectivos desde el punto de vista del aprendizaje del alumno, con integración avanzada en LMS.

¹ SCO: *Shared Content Object*, elemento de SCORM más simple capaz de establecer comunicaciones con el LMS en el que está alojado, típicamente es una página Web.

Objetivo 10. Crear/Usar/Evaluar laboratorios docentes *online* con integración avanzada en LMS basada en SCORM para su aplicación en asignaturas regladas de Grados de Ingeniería.

Objetivo 11. Diseñar y Crear una infraestructura física y lógica para la instalación de laboratorios remotos accesibles por Internet.

Objetivo 12. Difundir los resultados y herramientas obtenidos para extender su uso en otros organismos y/o entornos.

1.3. Descripción del Trabajo y Metodologías Utilizadas

Este trabajo de tesis tiene por título “Integración de Laboratorios *Online* de Automática y Telecomunicación en los Sistemas de Gestión de Aprendizaje mediante SCORM”. Se han realizado una serie de trabajos dirigidos a la consecución del objetivo principal de este trabajo: **la obtención de una metodología de desarrollo de laboratorios Web docentes efectivos con integración avanzada en LMS**. Parte de estos trabajos se han realizado en el marco de una Acción de Innovación y Mejora Docente de la convocatoria 2012/13 y dos Proyectos de Innovación Docente (PID) de las convocatorias de 2012-14 [54] y 2014-16 [55] que fueron concedidos por la Universidad de Jaén con el objetivo de mejorar y promover la implantación práctica de innovaciones en la docencia universitaria. Actualmente, se ha solicitado un tercer PID para dar continuidad a los mismos. Los trabajos desarrollados han seguido varias fases metodológicas con el fin de alcanzar los objetivos señalados en el apartado anterior. Estas fases se describen a continuación junto a los trabajos y objetivos asociados a cada una de ellas.

1.3.1. Fase 1. Iniciación, Estudio y Obtención de Prototipos

La primera fase se centró principalmente en la exploración bibliográfica de trabajos relacionados con los laboratorios docentes utilizados a través de Internet, también conocidos como laboratorios *online*, laboratorios Web o WebLabs. De este modo se han analizado diferentes tipos y modelos de laboratorios *online* poniendo especial énfasis en aquellos que podían establecer alguna relación con un LMS y el modo en que ésta se producía. En estos casos se analizaron los beneficios que obtienen los laboratorios que cumplían estos requisitos. Por otro lado, se analizaron estándares de *e-learning*, especialmente los más utilizados y extendidos en los LMSs como son los estándares SCORM. Se analizó la posibilidad de realizar un diseño, desde un punto de vista teórico-

pedagógico, de una estructura de contenidos SCORM que pudiera dar soporte a un laboratorio *online*. Esto incluía la creación de modelos de prueba de contenido y funcionalidades necesarias para realizar la integración en LMS mediante SCORM. Para ello, en esta fase también se realizó el análisis y prueba del *software* EJS con el que se obtuvieron varios VRL y las primeras versiones de las herramientas que facilitan la integración.

Análisis bibliográfico. Se descargaron y analizaron documentos científicos relacionados con la temática, para obtenerlos se trabajó a través de Internet realizando búsquedas por diferentes métodos:

- Directa en buscadores Google web [56] y Google Académico [57] basadas en cadenas relacionadas (*laboratory, virtual, remote, experiment, etc.*).
- En portales especializados como la librería digital *Institute of Electrical and Electronics Engineers (IEEE) Xplore* [58] y otros repositorios en los que se tiene acceso por ser miembro de la Universidad de Jaén (UJA), como el mantenido por la editorial Elsevier [59].
- Analizando las referencias de los trabajos encontrados por los métodos anteriores.

Para la gestión de documentos bibliográficos y referencias se utilizó el *software* de gestión bibliográfica Mendeley en sus versiones Web y de escritorio [60].



Figura 7. Herramientas de búsqueda y gestión bibliográfica utilizadas

Análisis de estándares de e-learning. El estudio se ha centrado en los que se utilizan para facilitar interacciones entre sistemas y en los utilizados para la creación de contenidos y evaluación automática, ya que son los más relacionados con la creación de laboratorios *online* y el propósito de integrarlos con LMSs. Se ha tenido como objetivo que la integración Laboratorio-LMS fuera avanzada y la portabilidad del laboratorio, es decir, que un mismo laboratorio pudiera ser utilizado en diferentes LMSs. El resultado de este análisis ha sido la

elección de SCORM como estándar para realizar la creación de laboratorios *online* con integración avanzada.

Estudio de Estándares SCORM. Se descargaron y analizaron los recursos relacionados con SCORM encontrados en el portal web de *Advanced Distributed Learning* (ADL) [61], una organización dependiente de la Secretaría de defensa de los EE.UU. de América, encargada de modernizar la gestión y distribución de la formación y educación.



Figura 8. Logotipo de ADL

Para ello, entre otras funciones, se encarga del mantenimiento y desarrollo de estos estándares. Entre los recursos descargados destacan las especificaciones técnicas de los estándares, guías de usuario y plantillas de un paquete SCORM básico.

Análisis y prueba del software EJS/EJSs. Se trata de una plataforma de *software* libre y abierta desarrollada por investigadores de la Universidad de Murcia, que facilita la creación aplicaciones y simulaciones interactivas en lenguaje Java y, desde la versión 5.0, en lenguaje JavaScript. Se tuvo conocimiento de este *software* tras algunas reuniones y encuentros con expertos en laboratorios *online* auspiciadas por el Grupo de educación en automática del Comité Español de Automática, CEA [62].

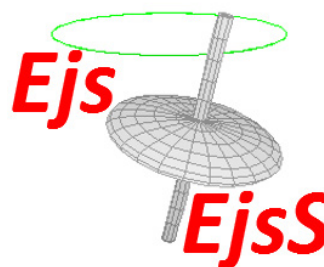


Figura 9. Software EJS/EJSs

El *software* EJS fue descargado de su página Web [63], utilizado para adquirir destreza en el manejo del mismo y crear diferentes simulaciones interactivas en Java. Estas simulaciones fueron los primeros VL desarrollados en el ámbito de este trabajo.

Modelo inicial de pruebas. Modificando una plantilla SCORM se hicieron diferentes pruebas en las que se comprobó la comunicación SCORM-LMS. Para ello se utilizó el LMS institucional y el de evaluación de la Universidad de Jaén [64]; Versiones del LMS ILIAS (*Integriertes Lern-,Informations- und Arbeitskooperations-System*) [65] desarrollado por la Universidad de Colonia que soporta las versiones 1.2



Figura 10. Logo del LMS ILIAS

y 2004 Ed.4 de SCORM. El trabajo se realizó editando documentos con códigos XML, HTML y JavaScript. Los VRL desarrollados programando en lenguaje Java con EJS se usaron para ser incluidos como parte de un paquete SCORM.

Primeras herramientas. Cuando se incrustó VRL como un *applet* de Java desarrollado con EJS y se comprobó la complejidad que entrañaba el intercambio de datos VRL-LMS se vio la necesidad de desarrollar herramientas que facilitaran estas comunicaciones. Se hicieron las primeras versiones en forma de métodos Java incluidos como código propio en EJS. La **Figura 11** muestra un esquema del modelo básico obtenido en la fase 1.

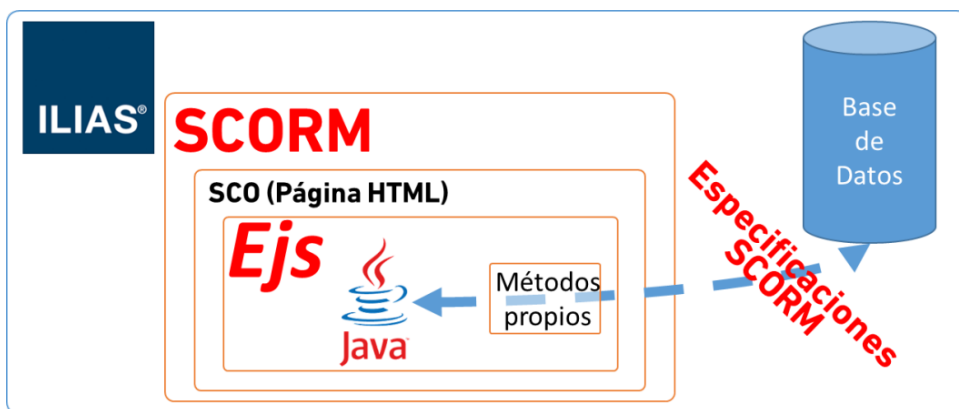


Figura 11. Modelo básico obtenido en la fase 1

Objetivos relacionados con la Fase 1. En esta primera fase se alcanzaron los Objetivos 1 a 5 parcialmente ya que algunos de los mismos se extendieron en el resto de fases. Por ejemplo, el análisis bibliográfico ha sido realizado de forma continua para conocer los avances relacionados incluidos en las publicaciones científicas o el desarrollo de *software* VRL que fue necesario para crear los laboratorios *online* desarrollados en el resto de fases.

1.3.2. Fase 2. Análisis, Herramientas y Metodología

En esta fase, una vez lograda una experiencia de integración avanzada Laboratorio-LMS, se realizaron trabajos relacionados con los objetivos de mejorarla desde el punto de vista de la efectividad del aprendizaje para facilitar la creación de nuevos laboratorios. Para ello, en primer lugar, se realizó un análisis de las posibilidades de integración que se pueden obtener y los beneficios que aportan cada una de ellas. Se analizaron los recursos

pedagógicos asociados a los VRL, se diseñó un modelo de estructura SCORM a desarrollar, se analizaron las funcionalidades de la integración de VRL en LMS y se obtuvo una metodología de creación de laboratorios *online* efectivos e integrados en LMSs. A raíz de los resultados y análisis de datos obtenidos se mejoraron las herramientas desarrolladas en la fase anterior, se crearon otras y se obtuvieron WebLabs con integración avanzada, diseñados para ofrecer un aprendizaje más efectivo.

Modos de Integración. Analizando los artículos revisados en el análisis bibliográfico y la documentación de diferentes estándares de *e-learning* y aplicables [66], [67], se han identificado diferentes formas de realizar la integración laboratorio-LMS. Se ha establecido una clasificación de modos de integración que se ha documentado también de forma gráfica usando las herramientas adecuadas [46].

Secuencias de Recursos asociados a los VRL. Los VRL no deben presentarse de forma aislada a los estudiantes. Múltiples estudios y la experiencia vivida por estudiantes y tutores han demostrado que la efectividad de un laboratorio aumenta al ser apoyado por una serie de recursos auxiliares como son: una correcta base teórica, un guion de prácticas adecuado, manuales de los recursos que se deben utilizar, etc. [68], [69]. A raíz de a) los datos obtenidos por el estudio bibliográfico realizado en la fase anterior, que mostraba diferentes formas de realizar el proceso cognitivo relacionado con el trabajo práctico [70], [71], y b) la experiencia propia en docencia virtual [72], [73], [74] y docencia de prácticas en diferentes modalidades, se identificaron los recursos que pueden/deben acompañar a un VRL. También se ha analizado la forma de presentación de los mismos creando una secuencia de recursos de aprendizaje entre los que se encuentra el propio *software* del VRL [75] (**Figura 12**).

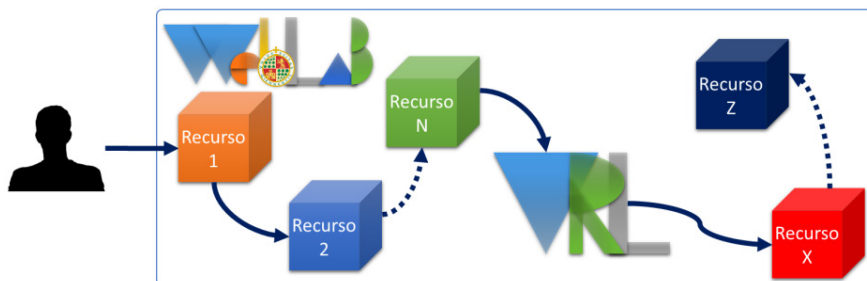


Figura 12. Ejemplo de secuencia de Aprendizaje que incluye un VRL

Se realizaron pruebas de modelos de paquetes SCORM en los que incluyeron estos recursos y se implementaron secuencias y recorridos de aprendizaje. La **Figura 13** muestra dos ejemplos de modelos de WebLab SCORM de 4 páginas, uno que incluye un VRL desarrollado en Java que usa el paquete scormRTE.jar y otro que incluye un VRL desarrollado en JavaScript que usa la librería RTE.js.

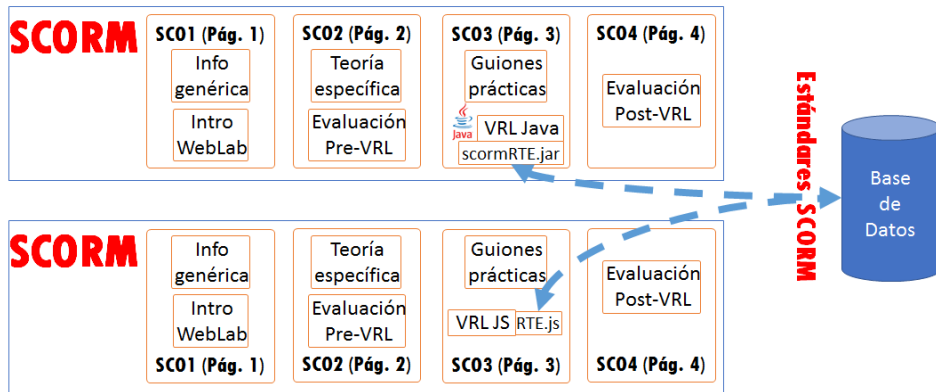


Figura 13. Ejemplo de modelo SCORM de 4 páginas para VRL Java y JavaScript (JS)

Para realizar estos trabajos se ha usado programación en HTML, XML, Java y JavaScript. También se han creado documentos en distintos formatos. Se ha utilizado el LMS ILIAS como soporte y herramienta de enlace entre recursos.

Metodología de creación de Laboratorios online. A partir del análisis de otras propuestas y el trabajo propio, basado en la creación de WebLabs integrados con LMSs, se ha documentado una metodología genérica de creación de laboratorios *online* integrados en LMSs en la que se definen las fases, agentes involucrados en cada fase y resultados producidos por cada una [52]. Para el caso concreto de la integración basada en ILIAS se detectó la necesidad de simplificar el proceso mediante el desarrollo de herramientas.

Herramientas. Para facilitar la creación de laboratorios *online* usando SCORM se han creado nuevas herramientas de ayuda, algunas de ellas a partir de los métodos Java obtenidos en la fase 1 y otras de forma totalmente novedosa. Para ello, se han programado diferentes herramientas en HTML, Java, JavaScript y XML (*eXtensible Markup Language*). Entre ellas destacan el paquete Java scormRTE.jar y la librería JavaScript rte.js que facilitan las comunicaciones Java-LMS y JavaScript-LMS respectivamente.

Objetivos relacionados con la Fase 2. En la segunda fase se realizaron trabajos orientados principalmente a la consecución de los Objetivos 6 a 9; aunque también estuvieron relacionados con los objetivos 1, 3, 4 y 5.

1.3.3. Fase 3. Desarrollo de WebLabs, Depuración y Difusión

En esta fase se han creado WebLabs, se han depurado y mejorado las herramientas de la fase anterior, se ha diseñado una infraestructura lógica para laboratorios remotos y se han realizado publicaciones de los resultados obtenidos en congresos.

Creación de WebLabs e infraestructura lógica de laboratorios remotos. Se han creado WebLabs de prueba en el LMS ILIAS aplicando la metodología de creación de laboratorios *online* (obtenida en la fase 2) al caso particular en el que se usan los estándares SCORM, el *software* EJS y las herramientas de ayuda (desarrolladas en las fases 2 y 3). Para la obtención de los laboratorios remotos también ha sido necesario desarrollar el *software* utilizado en el servidor de laboratorio, el cual permite el acceso y manipulación de los sistemas remotos reales (**Figura 14**). Además, en el caso de los laboratorios remotos, fue necesario establecer contacto con los Servicios Centrales de Informática (SCI) de la UJA, quienes realizaron la configuración necesaria y proporcionaron direcciones y puertos de acceso para los servidores tras reuniones mantenidas en las que se diseñó una infraestructura lógica para la instalación de laboratorios remotos.

Para todo ello ha sido necesario usar editores HTML, XML, Java, JavaScript, C y el LMS institucional de la Universidad de Jaén, ILIAS. De este modo se han obtenido varios laboratorios que han permitido depurar la metodología de creación de laboratorios *online* y de las herramientas desarrolladas.

Depuración y ampliación. Durante el desarrollo de los WebLabs se han depurado y ampliado las herramientas, creando ficheros de ayuda y modelos nuevos de SCORM en distintas versiones, creando nuevas herramientas de ayuda para realizar evaluaciones y obteniendo versiones corregidas y ampliadas de las ya existentes. Esta depuración se ha basado, además de en la experiencia obtenida, en la valoración de aspectos pedagógicos y el estudio de los recursos que pueden presentarse asociados a los VRL con el fin de mejorar la experiencia

docente y los resultados de aprendizaje de los usuarios que realicen sus prácticas en los laboratorios.



Figura 14. Ejemplo de Servidor y equipo real de WebLab remoto: Motor CC

Difusión de resultados. Se han realizado las publicaciones en congresos nacionales e internacionales explicando ejemplos obtenidos y herramientas desarrolladas. Como ejemplo, la **Figura 15** muestra una captura de pantalla del VL del sistema de suspensión de la rueda de un coche que fue utilizado en la comunicación *“Integration of Online Laboratories - LMS via SCORM”* que fue aceptada en el año 2013 en la conferencia internacional *2013 IEEE International Conference on Systems, Man and Cybernetics – SMC*, celebrada en Manchester.

Objetivos relacionados con la Fase 3. En la segunda fase se realizaron trabajos orientados principalmente a la consecución de los Objetivos 8, 10, 11 y 12, aunque también estuvieron relacionados con los objetivos 4, 5, 7 y 9.

1.3.1. Fase 4. Ajuste, Reestructuración y Finalización

Se utilizaron todos los recursos obtenidos en las fases anteriores (modelos, infraestructuras, metodología y las herramientas creadas) para obtener WebLabs que se pudieran utilizar en docencia reglada. Antes de crearlos fue necesario establecer reuniones y contactos con los encargados de las asignaturas en las que se iban a usar dichos contenidos con el fin de asegurar su efectividad para el aprendizaje. Tras el uso docente de los laboratorios se realizó

una evaluación de los mismos y de los resultados obtenidos, se realizaron algunos cambios y la difusión de los resultados obtenidos.

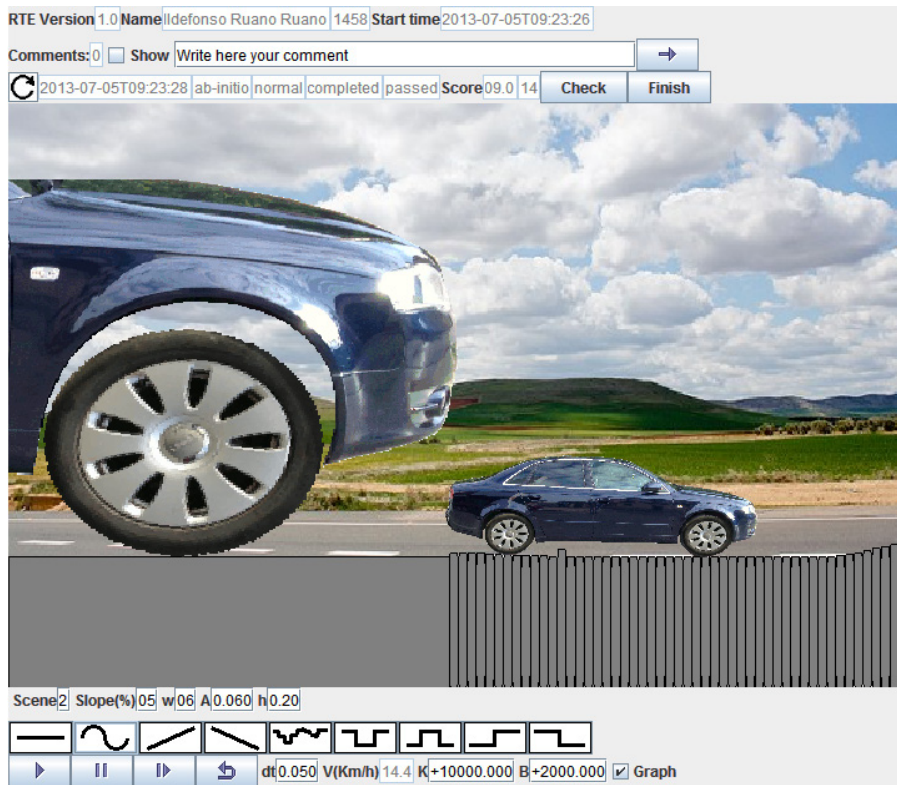


Figura 15. VL de sistema de suspensión de rueda de coche

Laboratorios docentes. La metodología y herramientas desarrolladas en las fases anteriores se utilizaron para la creación de laboratorios *online* teniendo en cuenta las ideas y objetivos marcados por los docentes responsables de las asignaturas en las que se pretendían aplicar. Además, se ha comprobado la reutilización de algunos de los laboratorios mediante la incorporación y utilización en otros LMS como Moodle, el LMS más utilizado en el mundo. Para ello se hizo uso de la plataforma UNILabs [27] gestionada por la UNED.

Uso de laboratorios. Los laboratorios obtenidos se aplicaron en algunas asignaturas de Grados de ingeniería del área de Automática en la Escuela Politécnica Superior de Jaén (EPSJ) y la Escuela Politécnica Superior de Linares (EPSL) y de Telecomunicaciones en la EPSL, en todos los casos de la Universidad de Jaén.

Evaluaciones y validaciones. Se realizó la evaluación y validación de los propios laboratorios, los resultados obtenidos por los alumnos e indirectamente las ventajas esperables que aportan la incorporación de la integración WebLab-LMS mediante SCORM. Para ello se han aplicado diferentes estrategias para medir la eficiencia de aprendizaje de los recursos creados basadas en los resultados obtenidos, las opiniones de los alumnos y el análisis de las características de los laboratorios obtenidos.

Cambios y Mejoras. El análisis de los resultados obtenidos en las evaluaciones permitió validar las propuestas y herramientas, corregir errores y mejorarlas. Cuando fue necesario se realizó una redefinición de los modelados UML y se modificaron los recursos desarrollados. Entre las mejoras, destaca el diseño y creación de una infraestructura para la ubicación protegida de los servidores y equipos reales de los laboratorios remotos **Figura 16**. En la misma infraestructura, los SCI de la UJA instalaron un equipo de telecomunicaciones exclusivo y se asignó la dirección Web DNS <http://weblab.ujaen.es> para el acceso a los equipos.



Figura 16. Infraestructura para ubicación y protección de servidores y equipos de laboratorios remotos de la UJA

Difusión de resultados. Una vez depuradas las propuestas y recursos obtenidos se realizó una difusión de las mismas y los resultados obtenidos en congresos y revistas de nivel científico elevado.

Objetivos relacionados con la Fase 4. En la segunda fase se realizaron trabajos orientados principalmente a la consecución de los Objetivos 10, 11 y 12, aunque también estuvieron relacionados con los objetivos 4, 7, 8 y 9.

El esquema de la **Figura 17** muestra objetivos y fases en las que se desarrollan.

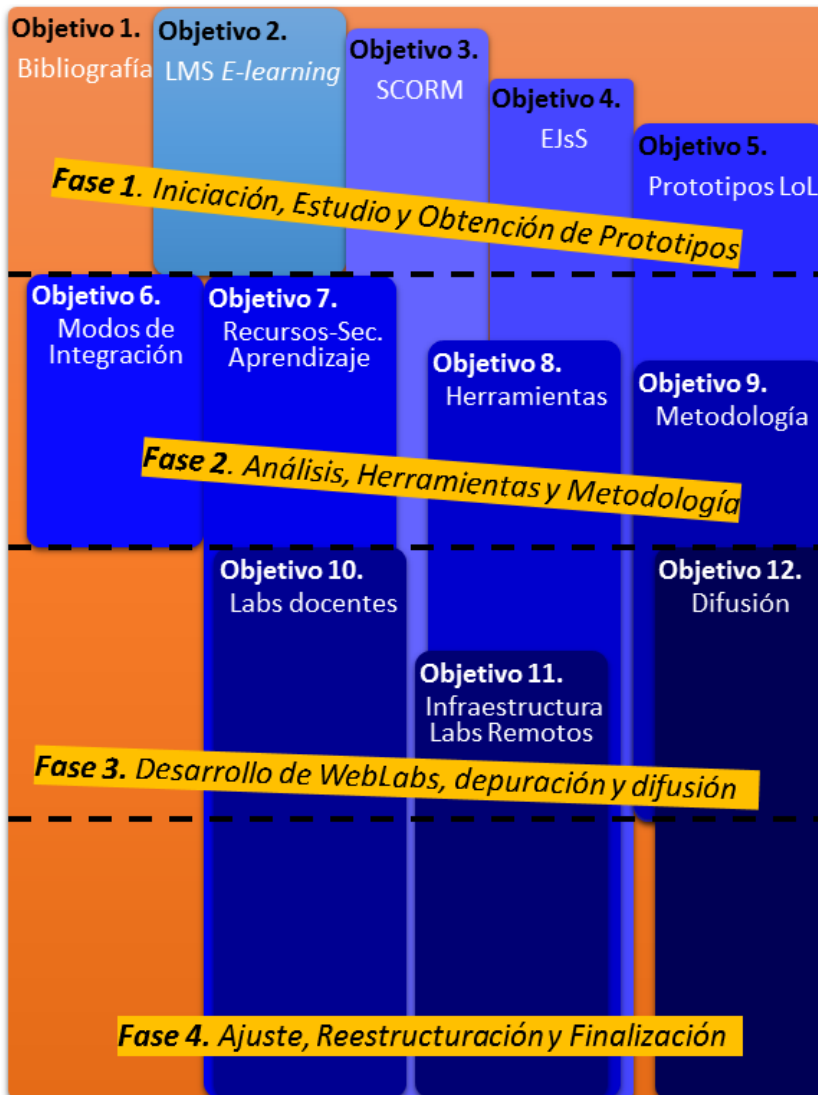


Figura 17. Objetivos y fases en las que se han desarrollado

1.4. Contribuciones de la Tesis

En esta sección se explican las principales contribuciones que aporta este trabajo de tesis de forma conceptual y en forma de comunicaciones en congresos nacionales e internacionales y revistas indexadas en el *Journal Citation Report* (JCR).

1.4.1. Contribuciones Conceptuales

El presente trabajo de tesis ha aportado una serie de contribuciones conceptuales que se enumeran a continuación:

- **Modos de integración VRL-LMS.** El análisis de las diferentes formas de Integración entre un laboratorio *online* y un LMS realizado a partir del análisis bibliográfico originó una clasificación de modos de integración que es analizada y expuesta con detalle.
- **Integración avanzada basada en SCORM.** La propuesta mostrada en este trabajo para conseguir la integración avanzada está basada en el uso de los estándares SCORM como objetos contenedores del VRL. Gracias a ellos se puede establecer la comunicación Laboratorio-LMS. Otra de las ventajas que ofrece SCORM es la posibilidad de crear secuencias de aprendizaje en un único paquete SCORM en el que está incrustado, junto a otros recursos asociados, el *software* del VRL.
- **Herramientas para integración VRL-LMS.** En el ámbito de este trabajo de tesis se han desarrollado herramientas que facilitan la integración avanzada de VRL basada en SCORM simplificando las comunicaciones. Se trata de plantillas SCORM, paquetes Java, librerías JavaScript y documentos de ayuda.
- **Metodología de creación de Laboratorios docentes *online* con integración avanzada en LMS.** Se presenta una metodología de creación de laboratorios *online* con integración avanzada en LMS que presenta fases, actores involucrados en las fases y los resultados generados en cada una de las mismas. Esta metodología aconseja no presentar el *software* del VRL de forma aislada, sino asociado a otros recursos necesarios para el establecimiento de secuencias de aprendizaje que maximizan la efectividad del aprendizaje. El objetivo final de la metodología es obtener laboratorios docentes más efectivos desde el

punto de vista del aprendizaje del alumno, además de poner bases que facilitan la presencia de otras características ventajosas como son, entre otras, la reutilización en diferentes LMSs, el acceso universal, el funcionamiento del software acorde a los gustos y/o necesidades de los estudiantes y la personalización de experimentos basada en el alumno.

- **Creación de Laboratorios *online* con integración avanzada en LMS.** La aplicación de la integración avanzada basada en SCORM descrita en el capítulo 3, la metodología propuesta en el 4 y las herramientas del 5 han permitido crear una serie de laboratorios *online* con integración avanzada en LMS basada en SCORM. Algunos de éstos han sido aplicados en la docencia de asignaturas regladas en la UJA.
- **Análisis de efectividad de laboratorios *online* integrados.** Con el fin de comprobar la efectividad y calidad de los laboratorios, que han sido utilizados en la docencia reglada, se han aplicado diferentes métodos de medición que tienen en cuenta diferentes factores: opinión de alumnos, resultados obtenidos en el laboratorio, resultados obtenidos en la asignatura y criterios de definición de calidad del laboratorio.
- **Infraestructura para laboratorios remotos.** Esta contribución tiene menor impacto científico aunque bastante importancia en el ámbito de la Universidad de Jaén. Se ha dotado a la Universidad de Jaén de una infraestructura lógica de comunicaciones y física mediante una estructura de ubicación-protección de equipos remotos que permite el desarrollo y mantenimiento futuro de otros laboratorios remotos.

1.4.2. Contribuciones en Congresos y Revistas

Este apartado muestra los trabajos que se han llevado a cabo en revistas y congresos Internacionales y nacionales con el fin de conseguir los objetivos descritos en el apartado 1.2 y que describen los desarrollos de las contribuciones descritas anteriormente.

Congresos Internacionales

[76] Ruano-Ruano, I.; Cano-Marchal, P.; Gámez-García, J.; Gomez-Ortega, J. "PID Control WebLab with LMS Integration Using SCORM". IFAC Workshop on Internet Based Control Education IBCE15 — Brescia (Italy), November 4-6, 2015. *IFAC-PapersOnline* Vol.48, Issue 29, 2015, P.301–306. doi: 10.1016/j.ifacol.2015.11.252

[50] Ruano-Ruano, I., Gamez-Garcia, J., Bencomo, S. D., & Gomez Ortega, J. (2014, October). "A SCORM based package model for WebLabs". In *Frontiers in Education Conference (FIE), 2014 IEEE* (pp. 47-50). IEEE. Madrid, España. ISBN: 978-1-4799-3921-3. Acceso Web

[77] Ruano Ruano, I., Garcia Gamez, J., & Gomez Ortega, J. (2014, October). "Building SCORM embedded WebLabs with LMS interaction". In *Frontiers in Education Conference (FIE), 2014 IEEE* (pp. 43-46). IEEE. Madrid, España. ISBN: 978-1-4799-3921-3. Acceso Web

[78] Ruano-Ruano, I., Gómez-Ortega, J., Gámez-García, J., & Estevez-Estevez, E. (2013, October). "Integration of *Online* Laboratories-LMS via SCORM". In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on* (pp. 3163-3167). IEEE. Manchester, Reino Unido. Acceso Web

Congresos Nacionales:

[46] Ruano-Ruano, I., Gámez-García, J. & Gómez-Ortega, J. (2016, Septiembre). "Modos de integración de laboratorios *online* en sistemas de gestión de aprendizaje" XXXVII Jornadas de Automática, 2016. Madrid, España.

[49] Ruano-Ruano, I., Gómez-Ortega, J., Gámez-García, J., & Estevez-Estevez, E. (2013, Septiembre). "Herramienta basada en SCORM para la integración automática de Laboratorios *Online* en LMS" XXXIV Jornadas de Automática, 2013. Tarrasa, Barcelona, España. Acceso Web

Revistas indexadas en JCR (*Journal Citation Report*):

Tres de las comunicaciones surgidas a partir del presente trabajo han conseguido la aceptación oficial en 3 revistas que se encuentran indexadas en la edición de ciencias (*Science*) del JCR dentro de varias categorías. A continuación se enumeran las comunicaciones y un breve resumen de las contribuciones asociadas.

[45] Ruano-Ruano, I.; Cano-Marchal, P.; Gámez-García, J.; Gomez-Ortega, J. "**Advanced LMS Integration of SCORM Web Laboratories**". IEEE Access. Aceptado-Publicación: <http://ieeexplore.ieee.org/document/7506221/> doi:10.1109/ACCESS.2016.2587805. JCR(2015): 1,249

Este artículo muestra algunas de las herramientas desarrolladas en el ámbito de esta tesis para facilitar la integración de VRL en LMS usando los estándares

SCORM. Especialmente el paquete Java scormRTE.jar que simplifica las comunicaciones entre un VRL desarrollado con el lenguaje Java en forma de *applet* y la librería JavaScript RTE.js que puede ser utilizado desde páginas web de un SCORM o VRL desarrollados en lenguaje JavaScript para facilitar y simplificar las comunicaciones con el LMS basadas en SCORM.

[52] Ruano-Ruano, I., Gamez-Garcia, J., Bencomo, S. D., & Gomez Ortega, J. **“A methodology to Obtain Learning Effective Laboratories with Learning Management System Integration”**. IEEE Transactions on Learning Technologies. Aceptado-Publicación: <http://ieeexplore.ieee.org/document/7523439/> doi:10.1109/TLT.2016.2594771. JCR(2015): 1,129

En este artículo se explica la metodología de desarrollo de laboratorios *online* identificando las distintas fases que la componen, los actores principales que pueden intervenir en las mismas y los resultados que se obtienen en cada una de las fases.

[51] Ruano Ruano, I.; Gámez García, J.; Gómez Ortega, J. **“Laboratorio Web SCORM de Control PID con Integración Avanzada”**. Revista Iberoamericana de Automática e Informática industrial-RIAI-ISSN 1697-7912. Volume 13, issue 4, pp. 472-483, <http://dx.doi.org/10.1016/j.riai.2016.05.007> October-December 2016. JCR(2015): 0,475

Este artículo muestra un ejemplo de Laboratorio *online* (WebLab) desarrollado con la metodología y las herramientas desarrolladas al amparo de esta tesis y los datos obtenidos tras su aplicación en un curso de “Automática Industrial” impartido en la EPSJ en diferentes grados de Ingeniería Industrial. Para ello se utilizó el LMS institucional de la Universidad de Jaén [64]. También se muestra la posibilidad de reutilización del WebLab SCORM en un LMS de otra institución que además es diferente, la plataforma UNILabs [27] basada en un LMS Moodle que gestiona la UNED.

1.4.3. Tabla-Resumen de Contribuciones-Comunicaciones

La **Tabla 1** muestra la equivalencia entre las contribuciones conceptuales y los trabajos publicados en congresos y revistas que están relacionados con cada uno de ellos.

Tabla 1. Comunicaciones en congresos/revistas y contribuciones conceptuales

Contribución	Congreso						Revista JCR		
	Internacional				Nacional		Access [45]	TLT [52]	RIAI [51]
	IBCE [76]	FIE1 [50]	FIE2 [77]	SMC [78]	JA16 [46]	JA13 [49]			
Modos de integración VRL-LMS					X		X	X	
Integración LMS basada en SCORM	X	X	X	X	X	X	X	X	X
Herramientas para integración LMS	X	X	X	X		X	X	X	X
Metodología de creación de LoL								X	X
Creación de LoL integrados en LMS	X	X	X	X		X	X	X	X
Análisis de efectividad de LoL	X							X	X
Infraestructuras para RL									

1.5. Estructura del Documento

La presente tesis doctoral se ha estructurado en un formato estructurado en capítulos como sigue a continuación:

- **Capítulo 1. Introducción.** Se ha contextualizado el marco en el que se ha desarrollado el trabajo exponiendo la justificación y motivaciones, los objetivos a alcanzar, la descripción del trabajo y metodologías seguidos, las contribuciones aportadas y la estructura del documento.
- **Capítulo 2. Estado del Arte y Tecnológico.** Se realiza una revisión bibliográfica de trabajos relacionados con laboratorios docentes *online* y se aclaran algunos conceptos tecnológicos como el de *e-learning* y los estándares de *e-learning*, relacionados con los trabajos desarrollados.
- **Capítulo 3. Integración Lab-LMS.** Este capítulo analiza las ventajas que aporta la integración de los laboratorios *online* con los LMSs, analiza las diferentes formas de integración que se pueden dar y realiza una clasificación de modos de integración. Posteriormente describe y analiza el modo de integración avanzado propuesto en este trabajo que se basa en el uso de los estándares SCORM mostrando ventajas e inconvenientes.
- **Capítulo 4. Herramientas y Utilidades Desarrolladas.** Se describen las herramientas que se han desarrollado para facilitar la integración Laboratorio-LMS usando los estándares SCORM y se justifica su uso.
- **Capítulo 5. Metodología Desarrollada para la Creación de WebLabs.** Se muestra la metodología genérica propuesta para la creación de

laboratorios *online* integrados en LMSs y se analizan las características ventajosas que se pueden conseguir si se siguen las recomendaciones dadas.

- **Capítulo 6. Aplicaciones y Laboratorios Desarrollados.** Se describen los laboratorios remotos, virtuales e híbridos que se han creado. Con los remotos se muestra la infraestructura lógica y física para ubicación, protección y mantenimiento de los equipos y servidores remotos. Con aquellos que han sido utilizados en docencia, se incluyen algunos de los resultados obtenidos y la evaluación que se ha llevado a cabo.
- **Capítulo 7. Conclusiones y Líneas de Futuro.** Se incluyen las principales conclusiones que se pueden extraer del presente trabajo y se apuntan las líneas de futuro que, ya se están siguiendo, o se pueden continuar a partir de lo conseguido hasta el momento.

Cap.2

2. Estado del Arte y Tecnológico

2.1. *E-learning* y Sistemas de Gestión de Aprendizaje

2.2. Laboratorios Docentes *Online*

2.3. Integración de Laboratorios *online* y LMSs

El contenido de este capítulo se puede dividir en tres grandes bloques. El primero incluye una serie de definiciones y descripciones de conceptos tecnológicos que resultan esenciales en este trabajo de tesis, como son el aprendizaje electrónico o *e-learning*, los Sistemas de Gestión de Aprendizaje y los estándares de *e-learning*. También se realiza una breve explicación del estado actual de estos conceptos, su uso en las universidades españolas y se incluye un breve manual de SCORM. Este último contenido puede resultar de

ayuda para comprender el funcionamiento de SCORM en los contenidos incluidos en el resto de capítulos. Por otro lado, el segundo bloque presenta una breve introducción a los laboratorios docentes *online* que incluye un estado del arte general de las tendencias existentes. Finalmente, el tercer bloque incluye un estado del arte sobre la integración de laboratorios *online* y LMSs poniendo especial énfasis en aquellos trabajos basados en los estándares SCORM.

2.1. *E-learning* y Sistemas de Gestión de Aprendizaje

El *e-learning*, aprendizaje electrónico o docencia virtual hace referencia al tipo de educación en el que se emplean medios electrónicos en el proceso formativo [79]. Teniendo en cuenta esta definición no implica que se tenga que hacer uso de Internet, sin embargo, el término es usado generalmente en procesos educativos que se realizan a distancia a través de Internet, y más concretamente mediante el servicio Web en lo que se ha llamado entornos de aprendizaje Web [80] [81]. En cualquier caso, el término es adecuado cuando se usa para cualquiera de sus modalidades: totalmente virtual (*e-learning* puro) o mixta (*b-learning* o *blended learning*) [82]. Lo que sí queda claro es que el término *e-learning* implica el uso de medios electrónicos.

En diciembre de 2015, la sectorial TIC de la CRUE (Conferencia de Rectores de Universidad de España) publicó el Informe UNIVERSITIC 2015, que muestra los resultados de un análisis global del estado de las Tecnologías de la Información (TI) en el Sistema Universitario Español [13]. En el estudio participaron 64 universidades (86%) que reúnen al 90% de los estudiantes universitarios de España. En este documento se indica que uno de los objetivos de las universidades participantes es “proporcionar soporte y promover la docencia no presencial” (objetivo 1.2). Relacionado con este objetivo existen una serie de indicadores que se muestran en la **Figura 18**.

La CRUE-TIC ha considerado importantes 21 buenas prácticas relacionadas con la docencia virtual, de las cuales, el 76% están implantadas o en fase de hacerlo. Destacan las mostradas en la **Figura 19**.

El informe indica que se aprecia una consolidación del soporte y promoción a la docencia no presencial, especialmente en las titulaciones no presenciales, donde el incremento es mayor. También apunta que se aprecia un uso elevado de la virtualización para la prestación de servicios para prácticas docentes. En este sentido, el uso de las plataformas de docencia virtual (LMS) institucionales es bastante alto por parte del profesorado y los alumnos como refleja la **Figura 20**.

	2015 ⁽¹⁾		evolución 2014-2015 ⁽²⁾			
	% resp.	media	% resp.	2014	2015	evolución
1.2. Proporcionar soporte y promover la docencia no presencial						
Nº de buenas prácticas relacionadas con la docencia virtual que lleva a cabo la universidad (sobre un total de 21)	97%	15,97	84%	15,89	16,16	1,7%
% de buenas prácticas relacionadas con la docencia virtual que lleva a cabo la universidad	97%	76,04%	84%	75,66%	76,94%	1,7%
Nº de PDI que utiliza la plataforma de docencia virtual institucional	89%	1.574,11	80%	1.596,38	1.592,61	-0,2%
% de PDI que utiliza la plataforma de docencia virtual institucional	89%	91,81%	80%	90,84%	91,88%	1,1%
Nº de titulaciones no presenciales	92%	9,02	81%	5,25	5,87	11,8%
% de titulaciones no presenciales	91%	8,06%	80%	5,05%	6,15%	21,7%
Nivel de madurez de la Universidad en relación con la adopción de MOOC	88%	38,89%	Nuevo Indicador 2015			
Número de cursos MOOC en las que participa (en exclusiva o compartidos) de forma activa la universidad	84%	5,91	Nuevo Indicador 2015			
Número de cursos MOOC en las que colabora de forma activa la universidad	80%	3,78	Nuevo Indicador 2015			
Porcentaje de cursos MOOC en las que colabora de forma activa la universidad, en relación con el total	61%	84,51%	Nuevo Indicador 2015			

⁽¹⁾ Los datos se corresponden con los aportados por todas las universidades participantes en la campaña 2015
⁽²⁾ Los datos pertenecen a las universidades que han aportado valor a este indicador en ambas campañas (2014 y 2015)

■ Importante tendencia positiva del indicador (por encima del +10%)
 ■ El indicador evoluciona levemente (entre el -5 y el 10%)
 ■ Tendencia preocupante del indicador (por debajo del -5%)

Figura 18. Indicadores de descripción del objetivo 1.2 (Fte: UNIVERSITIC 2015 [13])

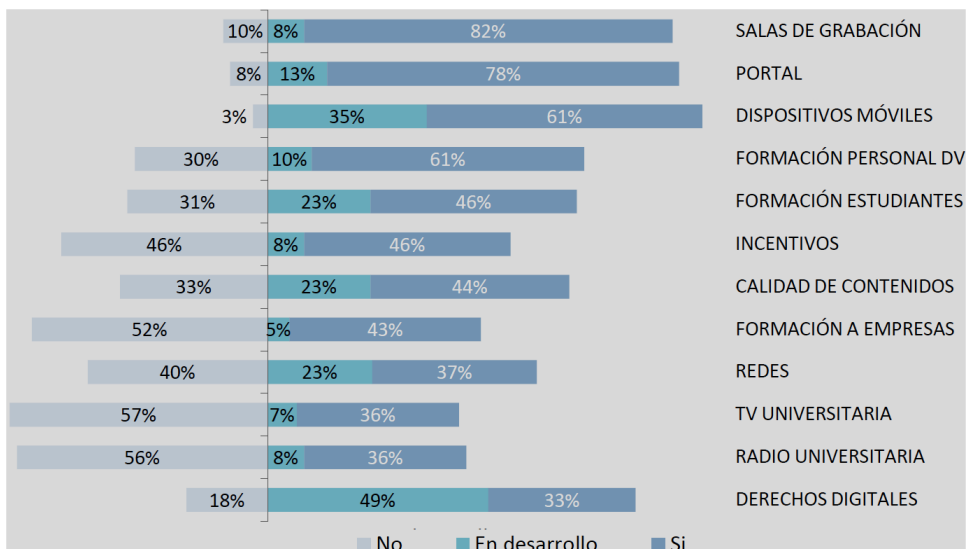


Figura 19. Porcentajes de buenas prácticas relacionadas con docencia virtual destacadas (Fte: UNIVERSITIC 2015 [13])

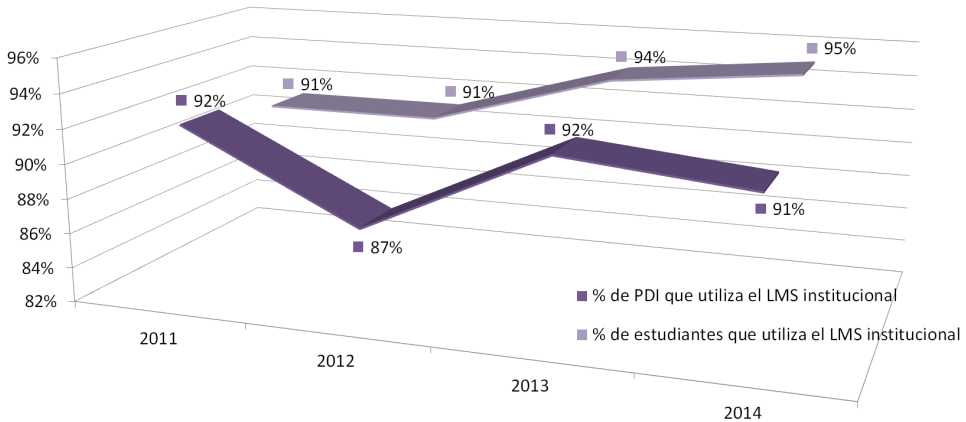


Figura 20. Uso del LMS institucional (Fte: UNIVERSITIC 2015 [13])

2.1.1. Sistemas de Gestión de Aprendizaje (LMSs)

Los Sistemas de Gestión de aprendizaje o LMS, del inglés *Learning Management System*, constituyen el mayor exponente del uso de las TICs en la educación. También son conocidos por el nombre de Plataformas o entornos de docencia virtual (*Virtual Learning Environment, VLE*). Los LMSs están plenamente instalados en las instituciones educativas y pueden suponer cambios bastante notables en el modelo pedagógico derivados de la educación a distancia [10], siempre y cuando los profesores no los utilicen como meros repositorios de contenido [83]. La educación a distancia puede suponer una sensación de abandono y soledad por parte del alumno que hay que suplir adaptando los contenidos que se les ofrece y ofreciendo una buena guía de estudio [74].

Algunos autores como Vrasidas [84] han realizado un análisis de los aspectos pedagógicos relacionados con los sistemas de *e-learning* señalando los cambios de rol que se producen en los estudiantes y alumnos, algunas circunstancias bajo las cuales los estudiantes aprenden mejor, herramientas que serían útiles para profesores en línea son aquellas que los apoyan. El mismo trabajo incluye un resumen de características que ayudan a conseguir un aprendizaje *online* eficaz junto a los requisitos que debe proporcionar un LMS en cada uno de los casos:

- Centrado en el alumno: los estudiantes organizan la información y el conocimiento, toman el control de su aprendizaje, actúan como individuos autónomos que planifican y ejecutan las tareas de aprendizaje. Un LMS debe proporcionar herramientas que permitan a

los alumnos organizar la información, aportar contenido, y participar en actividades de aprendizaje.

- Comprometidos y activos: los estudiantes participan en actividades interesantes que los motivan y emplean principios de aprendizaje activos para resolver los problemas de clase. Un LMS debe proporcionar herramientas que apoyen el aprendizaje activo y la resolución de problemas.
- Constructiva: el aprendizaje es un proceso constructivo durante el cual los estudiantes co-construyen conocimiento y significado al interactuar con sus compañeros, herramientas y contenido. Un LMS debe proporcionar herramientas que soporten varios tipos de interacciones estudiante-profesor y alumno-alumno.
- Situado y contextual: el aprendizaje se ubica en contextos del mundo real del que obtiene su significado real. Un LMS debe proporcionar herramientas que permitan a estudiantes y profesores integrar correctamente las actividades del mundo real en la planificación del curso.
- Social y Colaborativo: el aprendizaje es una actividad social y los estudiantes aprenden mejor cuando interactúan frecuentemente con maestros y compañeros. Un LMS debería permitir que los estudiantes interactúen, proporcionando herramientas de comunicación síncrona y asíncrona.
- Reflexivo: los estudiantes participan en el pensamiento reflexivo sobre sus acciones, habilidades, competencias, conocimiento, y las habilidades de meta-aprendizaje. Un LMS debe proporcionar herramientas que estructuren y apoyen una reflexión sobre el proceso de aprendizaje. p.ej. llevar un diario, preguntas de sondeo para que reflexionen sobre ello, etc.
- Requiere realimentación inmediata. Integrar la retroalimentación en lista de calificaciones. Un LMS puede utilizar agentes inteligentes para proporcionar información al trabajo de los estudiantes y ayudar al maestro para que monitoree el progreso de los estudiantes.

Hoy día los LMSs son unas herramientas esenciales en la educación universitaria que una vez realizada la identificación de un usuario, la cual puede realizarse de forma federada, pueden proporcionar las siguientes ventajas [85], [86]:

- Ofrecen bajo un mismo entorno web espacios virtuales que permiten integrar la oferta educativa de los organismos de educación superior.
- Permiten ampliar la oferta educativa y diversificarla.
- Ofrecen posibilidades de intercambio y exposición de contenidos docentes.
- Permiten establecer secuencias de aprendizaje.

- Ofrecen posibilidades de comunicación y colaboración entre usuarios.
- Ofrecen posibilidades de evaluación a los usuarios.
- Permiten realizar el seguimiento del trabajo realizado por los usuarios.
- Ofrecen soporte multilinguaje.

Para ello suelen ofrecer los siguientes recursos y servicios:

- Espacio Virtual personalizado, con posibilidad de almacenamiento *online* de contenidos privados y logros obtenidos (informes, *badges*, etc.).
- Gestión de usuarios, incluyendo su pertenencia a grupos y permisos sobre los recursos.
- Recursos de aprendizaje en diferentes formatos (propios, pdf, docx, etc.) como pueden ser módulos de teoría.
- Herramientas de comunicación como foros, chats, sesiones de videoconferencia, etc.
- Herramientas de creación de contenidos con posibilidad de permitir trabajo colaborativo como páginas web, wikis, módulos, blogs, portafolios, documentos en distintos formatos, etc.
- Herramientas de recogida de opiniones de usuarios como encuestas y/o votaciones.
- Herramientas de evaluación de usuarios como *tests*, tareas y trabajos *online*, así como de la gestión de los mismos, los resultados y las comunicaciones a los alumnos.
- Sistema de seguimiento del progreso de aprendizaje de los usuarios basados en números de accesos, tiempo de trabajo, competencias, objetivos y superación de eventos.

Para conocer la importancia que pueden tener los LMSs a nivel mundial se ha utilizado *Trend* [87], una herramienta de la empresa Google que permite representar la frecuencia con la que se ha buscado un término en Internet en porcentaje (considerando 100% el momento en el que la frecuencia de la búsqueda fue mayor). La **Figura 21** muestra los resultados obtenidos para el término LMS desde 2004 hasta el 1 de octubre 2016, estos datos confirman que los Sistemas de Gestión de Aprendizaje están en auge y en su mayor nivel de interés en el mundo (al 100%).

Existen muchos LMSs, más de 340 [88]. Atendiendo al desarrollo y propiedad del *software* asociado se pueden establecer tres tipos de LMSs: comerciales (Blackboard, Edoceo, Desire2learn, etc.), de *software* libre (Moodle, ILIAS, Dokeos, Sakai, etc.) [89] y de desarrollo propio (Ágora virtual, EducaLab, Virtaula, etc.). Algunos autores han realizado estudios e informes sobre el

estado de los LMSs en España llegando a la conclusión de que los más utilizados en las universidades de España son los de *software* libre, y especialmente Moodle, implantado en más de 35 universidades [85], [86]. En el mundo, el LMS más extendido y utilizado es también Moodle con más de 70767 instalaciones registradas en 232 países según indica su página web [90], de las cuales 7014 están en España, el segundo país que cuenta con más instalaciones de Moodle tras EE.UU (*Figura 22*).



Figura 21. Tendencia de búsqueda global del término LMS en Internet (2004-Oct2016)

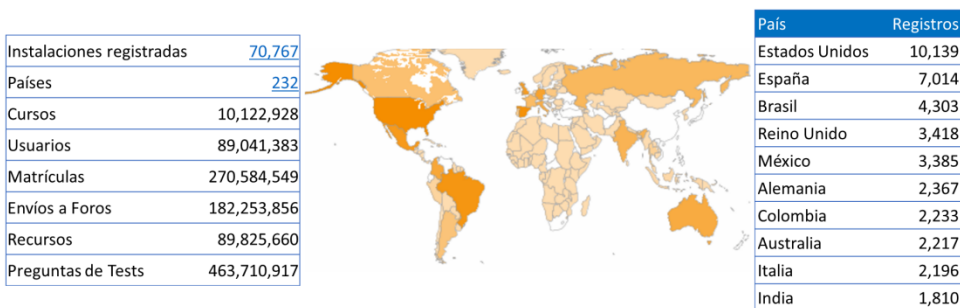


Figura 22. Estadísticas de Moodle (Gráfico de elaboración propia a partir de datos tomados de Moodle.net en Agosto de 2016)

ILIAS [65] es el LMS institucional utilizado en la Universidad de Jaén, es una plataforma de docencia virtual de código abierto creada en la Universidad de Colonia dentro del proyecto VIRTUS en el año 1998, año en el que se publicó su versión 1. En agosto de 2016 se publicó su última versión estable hasta el momento, la 5.1.9. El número de instalaciones ILIAS reconocidas en el mundo en septiembre de 2016 son 193 de 22 países (*Figura 23*)

Es un número mucho menor que el presentado por Moodle, sin embargo es una plataforma muy reconocida profesionalmente que ha conseguido la certificación de seguridad de la OTAN y es muy utilizada en universidades (47%), empresas (18%), administraciones públicas y gubernamentales (11%), centros de formación profesional (11%), escuelas (9%), y ONGs y organizaciones sin ánimo de lucro (3%) de todo el mundo. La mayoría de las instalaciones están en Europa

(95%), más concretamente en Alemania (60%). En la UJA se utilizan varias instalaciones ILIAS con fines diferentes; para docencia reglada [64], formación y evaluación de docentes [91] y circunstancialmente una tercera de evaluación de nuevas versiones de la plataforma. La versión de ILIAS instalada en la UJA actualmente es la última estable (v5.1.9) y el número de usuarios registrados es 43.633 aunque muchos de ellos no están activos de forma continua al tratarse de alumnos egresados que han terminado sus estudios recientemente y aún mantienen activos sus registros. En la **Figura 24** se pueden ver gráficas de las conexiones soportadas por el sistema LMS ILIAS principal de la UJA (dv.ujaen.es), obtenidas de <http://dv1.ujaen.es/admin/status/docencia/docencia.html>, donde se puede observar la concurrencia de usuarios en un periodo diario (promedio de 5 minutos) y anual (promedio de 1 día).

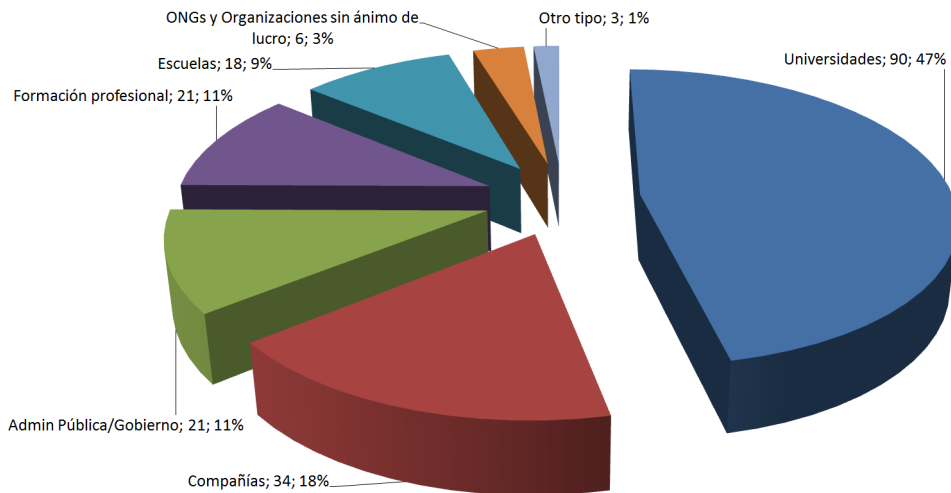


Figura 23. Instalaciones ILIAS en el mundo por tipo de usuario

2.1.1. Estándares de e-learning

El lenguaje español asumió como propia la palabra estándar procedente del término inglés *standard* que puede usarse como adjetivo o nombre. La Real Academia Española de la lengua lo define en su diccionario como adjetivo de la siguiente forma: “Que sirve como tipo, modelo, norma, patrón o referencia”, y cuando se usa como nombre simplemente con los términos “Tipo, modelo, patrón, nivel”. El **objetivo** de un estándar en el ámbito tecnológico es **establecer normas que permitan la colaboración y el entendimiento entre sistemas**. Los

estándares se encuentran en múltiples ejemplos de la vida cotidiana, probablemente el caso más conocido es el propio Internet que se puede definir como una red mundial de dispositivos heterogéneos interconectados entre sí gracias al uso común en todos ellos del conjunto de estándares TCP/IP (*Transport Control Protocol/Internet Protocol*).

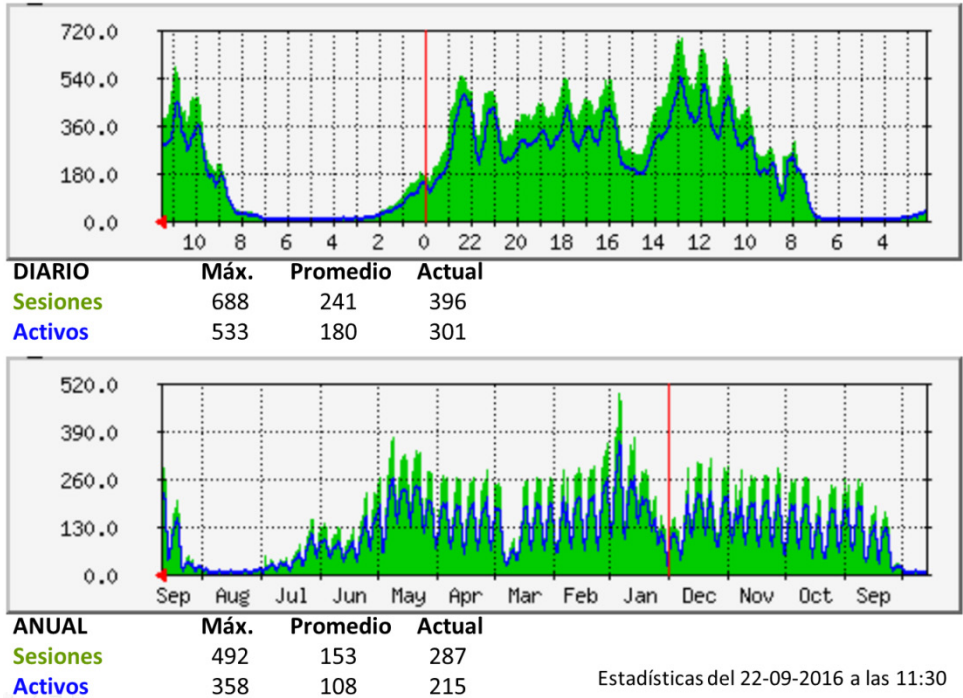


Figura 24. Estadísticas de uso diario y anual del LMS ILIAS principal de la UJA (22-9-2016)

Existen **estándares oficiales** que han sido aprobados y sancionados por un organismo oficial de estandarización nacional o internacional, y son de obligado cumplimiento y, por otro lado, los **estándares de facto**, que aunque no hayan sido sancionados por un organismo oficial se usan por voluntad propia debido a la conveniencia que conlleva su uso. Los estándares aseguran que el desarrollo de cualquier producto se ajuste a unas normas que facilitan el entendimiento del mismo (gestionable). Esto permite la cooperación entre los productores sin mermar su capacidad de competir (interoperabilidad). También se asegura la compatibilidad entre productos, lo que permite a los usuarios cambiar de proveedor manteniendo la compatibilidad (reusabilidad).

Todas las ventajas que aporta un estándar son aplicables al campo de la educación. Los estándares permiten la reutilización de los recursos de

aprendizaje, la posibilidad de poder ser utilizados en diferentes sistemas heterogéneos y que se puedan desarrollar todas las interacciones necesarias entre contenidos, aplicaciones y herramientas y sistemas utilizados. El uso de los estándares trae consigo las siguientes ventajas a todos los actores involucrados en el proceso educativo [92]:

- **Consumidores de Aplicaciones** (Organizaciones educativas). Si usan estándares en vez de tecnología propietaria se aseguran no depender de una empresa con la que pueden verse atrapados. Reducen coste si desean cambiar desarrollos propios, normalmente tienen portabilidad ante actualizaciones del estándar.
- **Productores de Aplicaciones** (Industria de Aplicaciones de *e-learning*). Si usan estándares de comunicaciones entre sistemas simplifican la integración entre productos, reducen coste de desarrollo y amplían las posibilidades de venta (mercado potencial).
- **Creadores de Contenidos de *e-learning*** (Profesorado e Industria de contenidos). Si usan estándares cuando crean un contenido se hará de la misma forma y podrá ser usado en diferentes entornos compatibles. De este modo se aumenta el mercado potencial ya que habrá más posibilidades de vender un contenido.
- **Alumnado** (Consumidores finales). La existencia de estándares asegura que la oferta de contenidos sea mayor y por tanto tendrán más posibilidades de elección. Los contenidos de un mismo estándar tendrán características similares que les exigirá menos esfuerzo cuando trabajen con ellos. También la portabilidad de los resultados de aprendizaje (créditos, informes o certificados) será mayor.

La aplicación de Internet en la educación ha traído consigo muchos beneficios, pero también algunos problemas debido al hecho de que Internet sea una red abierta, diversa y heterogénea [67]. Esto implica el uso de diferentes sistemas como, ordenadores o dispositivos en general, sistemas operativos (SO), LMSs, sistemas de almacenamiento, protocolos de comunicación, etc. Otros problemas están relacionados con el coste de desarrollo de contenidos, herramientas, y otros recursos utilizados en los cursos y de su adaptación ante cambios en el entorno tecnológico. Por todo esto es necesario que la creación de materiales sea de calidad y se asegure su reutilización en el tiempo. Gracias a los estándares se pueden solucionar muchos de estos problemas. La industria de la estandarización se ha desarrollado ampliamente en esta área y existen

muchas organizaciones dedicadas a ello: ADL (*Advanced Distributed Learning*) [61], AENOR (Asociación Española para la Normalización y Certificación) [93], AICC (*Aviation Industry Computer Based Training Comitte*), CanCore, CEN (Comité Europeo de Normalización), Content Guard, CORDRA (*Content Object Repository Discovery and Registration/Resolution Architecture*), EDNA (*Education Network Australia*), EFQM (Fundación Europea para la Gestión de la Calidad), EFQUEL (Fundación Europea para la Calidad en el *E-learning*), HR-XML, IEEE/LTSC (*Institute of Electrical and Electronic Engineers/Learning Technologies Standarization Committee*), IMS (*Instructional Management System*), ISO/EIC (*International Standards Organisation/ International Electrotechnical Commission*), LORN (*Learning Object Repositories Network*), OAI (*Open Archives Initiative*), OASIS, ODRL (*Open Digital Rights Language*), OKI (*Open Knowledge Initiative*), OMA (*Open Mobile Alliance*), ProLearn, SIF (*Systems Interoperability Framework*), W3C (*World Wide Web Consortuim*) [94], DCMI (*Dublin Core Metadata Initiative*).

La estandarización no se ha centrado sólo en la organización y descripción de los contenidos (resuelven el problema de la interoperabilidad y portabilidad) sino también en aspectos dinámicos que permiten describir procesos educativos complejos que permitan crear aplicaciones más complejas, por ejemplo que sean colaborativas o estén basadas en competencias. Las diferentes categorías que pueden ser objeto de una estandarización en el proceso educativo se muestran a continuación [66]:

- **Accesibilidad.** Posibilidad de que un servicio o producto de *e-learning* sea usado por más personas incluyendo la capacidad de ajustarse y adaptarse a las necesidades y/o gustos de los usuarios.
- **Arquitectura.** La técnica y estilo en el que se desarrolle el sistema de *e-learning*. Es recomendable que sea abierta, escalable, integrada, flexible, adaptable y global.
- **Calidad.** Métodos de medición del valor de todo el proceso de *e-learning*.
- **Competencias.** Habilidades y conocimientos medibles que los alumnos pueden adquirir en el proceso educativo.
- **Contenidos y Evaluación.** Patrones de creación de contenidos con posibilidad de empaquetado y su evaluación.
- **Derechos digitales.** Publicación, gestión/entrega y autorización de contenidos mediante derechos digitales.

- **Información del alumno.** Almacenamiento y gestión de información sobre el alumno y sus logros con privacidad de forma individual o colectiva.
- **Interactividad.** Intercambio de información entre recursos de aprendizaje y LMSs durante la ejecución de los mismos.
- **Metadatos.** Información adicional sobre las estructuras, contenidos y herramientas del proceso.
- **Proceso de Aprendizaje.** Métodos de aprendizaje basado en teorías pedagógicas.
- **Repositorios.** Descripción de contenidos y recursos de *e-learning* incluidos en los contenedores o depósitos y publicación de métodos de interoperabilidad entre ellos.
- **Vocabulario y Lenguajes.** Necesarios para la comunicación y entendimiento entre máquinas y humanos involucrados en el *e-learning*.

Los laboratorios Web, como recurso educativo que se utiliza a través de Internet, se pueden considerar un tipo de contenido de *e-learning* y por tanto se les puede aplicar todos los estándares aplicables al resto de contenidos. No existe un estándar específico o particular que aborde los contenidos de *e-learning* de tipo laboratorio. Los laboratorios constituyen un tipo de recurso muy particular que suelen presentar una variedad en su implementación y complejidad mucho mayor que el resto de contenidos, máxime cuando se trata de un laboratorio remoto que, al menos, exige el uso de comunicaciones con recursos reales distantes. En este sentido, un laboratorio abarcaría todos los aspectos utilizados por [66] para clasificar los estándares de *e-learning*. En este tipo de recursos se hace más necesaria la existencia de un método de registro del seguimiento de la experiencia que realiza el alumno. Por ello cobra más importancia los estándares basados en las interacciones y comunicaciones contenido-LMS.

Estándares para la comunicación Contenido-LMS actuales:

Existen varios estándares que permiten implementar diferentes métodos para realizar el seguimiento de los datos relacionados con un contenido docente y almacenarlos en la base de datos de un sistema de almacenamiento

centralizado, normalmente un LMS. Estas especificaciones pueden permitir realizar el seguimiento del progreso y la interacción en *e-learning*.

2.1.1.1. Computer-Managed Instruction (CMI)

El comité para la formación mediante ordenador de la industria de la aviación (AICC, *Aviation Industry Computer-Based Training Committee*) publicó en 1993 un documento titulado "CMI Guidelines for Interoperability" [95] que constituye el origen de los estándares de comunicación entre contenido y LMS. Originalmente fue una especificación basada en archivos para escritorio, aunque las revisiones posteriores se diseñaron para un navegador web. Se puede considerar que es la primera especificación y por tanto el origen de los estándares de comunicación entre contenido docente y LMS.

2.1.1.2. Shared Content Object Reference Model (SCORM)

Es el conjunto de estándares de *e-learning* más extendido y utilizado, existen varias versiones, la última que es la versión SCORM 2004v4 está formado por 3 sub-especificaciones que sirven para definir como crear un formato de contenido de *e-learning* en forma de paquete (paquete SCORM). Para ello estas especificaciones indican como crear y empaquetar los contenidos de *e-learning*, la navegación y secuenciación entre dichos contenidos y, finalmente, como el contenido puede interactuar (mediante funciones de comunicación) con el LMS en el que está alojado el paquete para intercambiar datos de un vocabulario establecido por SCORM.

SCORM partió con muy buenas expectativas a pesar de tener algunas críticas iniciales [96] [97] y de hecho se ha convertido en el estándar de facto de la creación de contenidos específicos de *e-learning*. Sin embargo, presenta unos problemas que junto a la aparición de nuevos estándares está haciendo que vuelva obsoleto rápidamente. Todo ello a pesar de que la industria de *e-learning* suele tomarse bastante tiempo para adoptar las nuevas especificaciones. Estos son los más importantes:

- El contenido debe residir en el mismo dominio que el LMS. Esto es ineficiente si se piensa en la forma de trabajar en la nube que existe hoy día. Este problema se puede superar pero exige soluciones particulares

para cada sistema y no forman parte de la especificación por lo que no pueden ser utilizadas por otros sistemas.

- La especificación SCORM es extensa y compleja, para desarrollar contenidos SCORM se requieren recursos dedicados y cumplir todas las normas.
- Diseñado para navegadores Web. El contenido SCORM incrusta su API (*Application Program Interface*), con la que se logra la comunicación entre el contenido SCORM y el LMS, en una página Web. Sin embargo, no todo el contenido de *e-learning* tiene que estar basado en Web. Por ejemplo no está pensado para ser ejecutado como aplicaciones de dispositivos móviles, los cuales se utilizan cada vez más.
- Plantea problemas de seguridad. La tecnología de SCORM está desfasada y es fácilmente accesible.
- SCORM utiliza un modelo de datos definido estrictamente que no permite intercambiar otros datos no definidos en el mismo. No todas las experiencias de *e-learning* tienen que coincidir con los datos de este modelo.
- El seguimiento que realiza SCORM se hace por estudiante a nivel individual, no proporciona ni realiza seguimiento de otros roles como tutores o profesores ni proporciona datos en grupos de alumnos.
- SCORM no proporciona ninguna guía a los LMSs para mostrar o exportar los datos SCORM. Cada LMS lo hace a su manera, mostrando los datos que considera oportunos, ocultando otros y con formatos particulares.

A pesar de estos problemas, aunque se aproveche muy poco su potencial, y se utilice principalmente como un medio simple de realizar el seguimiento del progreso de un usuario y la puntuación dentro de un curso de *e-learning*, SCORM está bastante bien afianzado y es la especificación más utilizada y extendida [98].

2.1.1.3. Tin Can API o *eXperience* API (xAPI)

xAPI fue publicado en 2012 con el nombre de Tin Can API o TCAP, aunque actualmente se le denomina con el nombre *eXperience* API o simplemente xAPI por ADL, el organismo encargado de su desarrollo y mantenimiento [99], [100]. Aunque hay foros donde se suele decir que es el sucesor de SCORM [101], xAPI no es, ni se desarrolló, para reemplazar SCORM, constituye una revolución si se

compara con SCORM. Es el primer componente de la Arquitectura de Formación y aprendizaje de ADL, sin embargo, no se ideó como una especificación para establecer comunicaciones entre un LMS y contenido de *e-learning*. xAPI aglutina las experiencias de aprendizaje definidas en un formato llamado declaraciones de estado (*statements*) y las registra en un LRS que almacena los registros de actividad (*Learning records*) en la nube (**Figura 25**). Para ello xAPI define comunicaciones entre una experiencia de aprendizaje y un Almacenamiento de Registros de Aprendizaje (LRS, *Learning Record Store*). Esto significa que no hay necesidad de usar un LMS o un navegador, se puede registrar la experiencia de aprendizaje de un estudiante desde cualquier dispositivo, aplicación o medio.

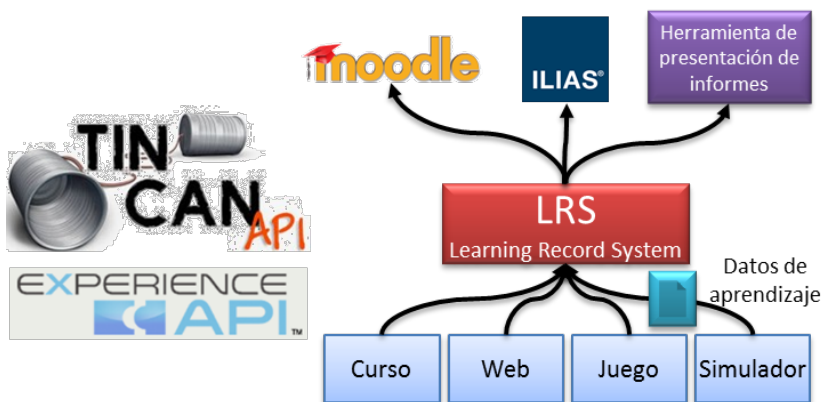


Figura 25. Esquema de funcionamiento de xAPI

Como resumen, xAPI puede usarse sin necesidad de un LMS, sin problema de dominio cruzado, sin necesidad de un navegador. Pero no todo son ventajas, respecto al nivel de complejidad xAPI supone un cambio muy importante que exige un nivel de complejidad mayor. Además sólo cubre y soluciona parte de los problemas que plantea SCORM pero no es tan completo como SCORM ni permite hacer tantas cosas.

Antes ya se comentó que xAPI no es un sustituto de SCORM, sino que con su implantación se han sentado las bases para un nuevo estándar SCORM. De hecho existe una especificación con la que un contenido SCORM puede utilizar el estándar xAPI que se explica brevemente a continuación.

2.1.1.4. *Experience API SCORM Profile (Perfil SCORM de xAPI)*

El perfil SCORM xAPI [102] es una especificación que describe cómo representar los datos de actividad y los comportamientos ante los intentos en el tiempo de ejecución de un contenido SCORM usando xAPI. Se trata de un documento adicional de la especificación xAPI cuyo objetivo es proporcionar ayuda a la comunidad SCORM en el uso de xAPI. Este perfil ofrece una guía para representar datos y eventos de SCORM en forma de declaraciones xAPI. Usando este perfil se puede integrar los datos proporcionados por un contenido SCORM en un entorno xAPI. Proporciona coherencia en los informes y datos almacenados por los paquetes SCORM tradicionalmente en un LMS, y permite el desarrollo de herramientas de interoperabilidad fuera del entorno LMS típico.

2.1.1.5. *Learning Tool Interoperability (LTI)*

El estándar LTI ha sido desarrollado por el IMS *Global Learning Consortium* (IMS) [103]. LTI tiene sus orígenes en las especificaciones de Interoperabilidad de herramientas IMS que fue lanzado en el año 2006. Más tarde se convirtió en lo que hoy día se conoce como LTI o interoperabilidad de herramientas de aprendizaje. En mayo de 2010 se publicó una versión llamada LTI básico (Basic LTI) que proporcionaba un mecanismo simple para el lanzamiento de herramientas y contenidos desde un LMS. En marzo de 2011 se renombró como LTI 1.0 y más tarde como 1.2. Actualmente la última versión publicada de LTI es la 2 [104], que data del año 2014.

El concepto principal de LTI es la creación de un estándar de integración entre aplicaciones de aprendizaje enriquecidas (que pueden haber sido creadas por cualquier empresa y método, y alojadas remotamente) con plataformas como LMSs, portales Web, repositorios de objetos de aprendizaje o cualquier otro entorno de aprendizaje. En la terminología LTI a las aplicaciones de aprendizaje se les llama **Herramientas** (creadas por **Proveedores de herramientas**) y a los LMSs, plataformas o portales Web se les llama **Consumidores de herramientas**. La **Figura 26** muestra un esquema ejemplo de uso con los actores involucrados. Se trata de unas especificaciones más utilizadas en el ámbito académico que otras áreas y menos extendida en general que las explicadas anteriormente. Este estándar permite lograr una integraciones de tipo “*plug and play*” entre un LMS y aplicaciones externas de forma rápida, fácil y barata. Para ello el LMS y la aplicación deben ser compatibles con LTI. La aplicación puede ser de cualquier tipo: e-portafolios, libro electrónico, editor de documentos o presentaciones,

herramientas de colaboración, recursos de biblioteca, herramientas de evaluación, herramientas de tareas, herramientas de clasificación, herramientas de video o herramientas de temas específicos. Sólo debe usar la API LTI para comunicarse con el LMS y así poder registrar las acciones que realiza el usuario directamente en el LMS.

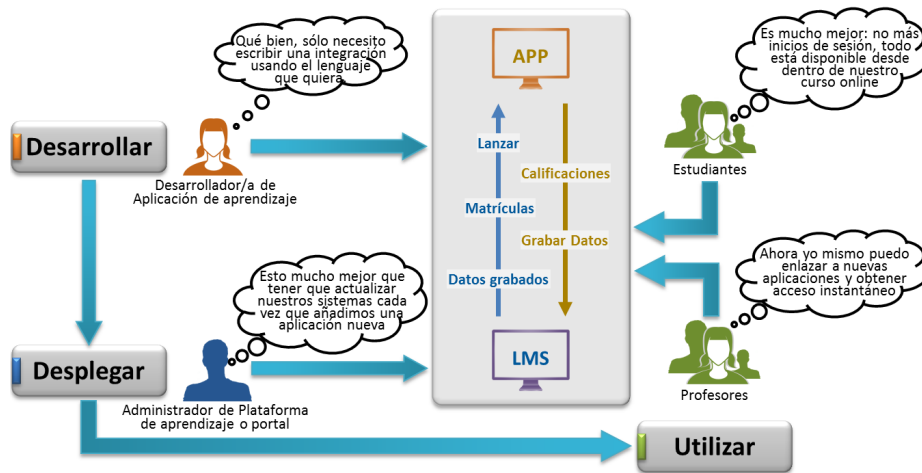


Figura 26. Esquema de funcionamiento de LTI

Aunque los objetivos de LTI 1 son los mismos que los de LTI 2, la tecnología que utiliza es muy diferente. LTI 2 introduce una arquitectura nueva para la implementación de LTI aunque es compatible con la versión anterior y permite que cualquier consumidor o proveedor de herramientas que soporte LTI2 pueda conectarse con un consumidor de herramientas o consumidor de LTI1.

2.1.1.6. OpenSocial

Se trata de una especificación pública dependiente del W3C [94] (grupo de trabajo en Web social) que define un conjunto de APIs comunes y un contenedor (entorno de hospedaje de componentes) con el que se pueden desarrollar aplicaciones Web. Las aplicaciones desarrolladas con las APIs de OpenSocial pueden interactuar con cualquier sistema de red social compatible. Aunque no se trata de una especificación propia de *e-learning* proporciona una serie de herramientas con las que se pueden establecer comunicaciones aplicadas a estos entornos o relacionar entornos educativos con redes sociales.

2.1.1.7. *cmi5*.

cmi5 es la nueva versión de la especificación CMI (*Computer-Managed Instruction*) del AICC. AICC colaboró en el desarrollo de las especificaciones SCORM y xAPI mientras desarrollaban *cmi5* por lo que han incorporado lo mejor de cada especificación. En 2014, AICC se disolvió y transfirió formalmente el proyecto *cmi5* a ADL que es el organismo actualmente responsable del mismo [105]. En 2015 se publicó una versión beta de *cmi5* llamada “*sandstone*”. Es básicamente una versión beta estable, lo que significa que pueden hacer desarrollos basados en la especificación ya que la ADL se compromete a no hacer grandes cambios. El objetivo era dar a conocer una versión final de la especificación a finales de 2015. Sin embargo, a día de hoy aún se trabaja en una versión llamada “*Quartz*” que recoge las ideas enviadas por los desarrolladores que usaron “*sandstone*”.

La especificación *cmi5* es una actualización de las necesidades tanto del LMS como del contenido. *cmi5* aprovecha la especificación xAPI para modernizar la comunicación entre los contenidos, o unidades asignables (AUs), y el LMS. Además de modernizar el modelo de comunicación, *cmi5* define una nueva estructura de curso para empaquetar y describir su contenido, y el proceso de poner en marcha e inicializar el contenido. Entre los objetivos de *CMi5* destaca que presenta un soporte multi-dispositivo (incluido dispositivos móviles), permite comunicaciones entre agentes de cualquier tipo (incluido ficheros binarios) y pueden funcionar del mismo modo desde cualquier ubicación, sin tener que residir en el propio LMS.

En resumen, *cmi5* es una revisión completa de la especificación CMI que incorpora ideas de las especificaciones SCORM y xAPI. Complementa a xAPI añadiendo, entre otras cosas, vocabulario y reglas, probablemente sea la sucesora de SCORM aunque todavía debe publicarse una versión definitiva y ver cómo funciona.

2.1.2. SCORM

En el apartado “2.1.2.2. Shared Content Object Reference Model (SCORM)” se han descrito brevemente las características de las especificaciones SCORM, en este se va a explicar con mayor detalle explicando algunos conceptos clave, las versiones del producto que se han realizado y las tres sub-especificaciones que lo componen. Se trata de unos estándares dirigidos a personal técnico como

programadores [106]. Los contenidos SCORM pueden presentarse a los estudiantes mediante cualquier LMS compatible con SCORM que utilice la misma versión.

2.1.2.1. Versiones

La primera versión de SCORM, SCORM 1.0, fue publicada en 2001 por ADL (*Advanced Distribution Learning*), una división del departamento de defensa de los EE.UU. Desde entonces aparecieron nuevas y diferentes versiones hasta que en el año 2009 se publicó la última y más avanzada: la 4ª edición de SCORM 2004. La **Figura 27** muestra las distintas versiones y los años en los que se han publicado cada una de ellas.

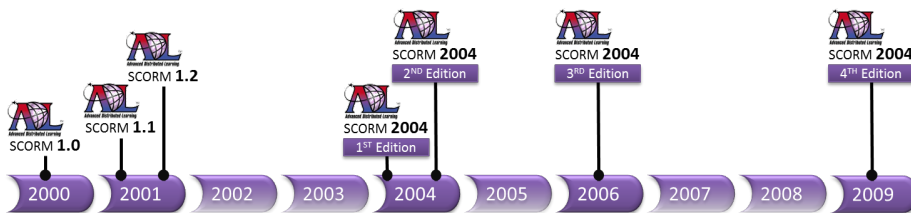


Figura 27. Evolución de las versiones de SCORM

Las versiones más populares son SCORM 1.2 y SCORM2004v4. SCORM 1.2 es un conjunto de especificaciones que sirven para describir un modelo de agregación de contenido (sub-especificación CAM, *Content Agregation Model* [107]) y como éste puede comunicarse con un LMS (sub-especificación RTE, *Run-Time Environment* [108]). SCORM 2004 mejora las especificaciones de SCORM 1.2 y añade otra que permite definir la secuenciación y navegación de los elementos que se forman el contenido (sub-especificación SN, *Sequencing and Navigation* [109]).

Al ser un estándar más antiguo y simple, existe un soporte bastante amplio para la versión 1.2 en la mayoría de los productos de edición y LMSs. También existe un amplio grupo, aunque menor, que soportan la especificación de 2004. SCORM es, hoy día, el estándar más utilizado que permite comunicaciones entre contenido de *e-learning* y un LMS.

A continuación se van a explicar las sub-especificaciones incluidas en la última versión: SCORM 2004v4.

2.1.2.2. Sub-especificación CAM

El modelo de contenido SCORM se presenta en forma de paquete comprimido en formato ZIP (fichero PIF, *Packet Interchange File*) que debe ser incorporado en un LMS para presentarse a los alumnos. La especificación CAM, Modelo de Agregación de Contenidos de SCORM [107] describe los componentes usados en un paquete SCORM, como se deben encapsular para poder ser utilizados en distintos sistemas, como describirlos para que puedan ser buscados y descubiertos y como definir la secuencia de información entre componentes.

La estructura de un elemento SCORM puede estar formada por los siguientes tipos de contenidos:

- **Asset.** Es la representación electrónica de un medio que puede ser texto, imagen, sonido, vídeo, *applet* de Java, código JavaScript o cualquier otro elemento de datos que pueda ser tratado por un navegador Web y presentado a un usuario (**Figura 28**). Los Assets pueden ser descritos con metadatos para permitir su búsqueda en repositorios y reutilizarlos.



Figura 28. Ejemplos de Assets

- **Sharable Content Object (SCO).** Es un conjunto de uno o más Assets que representa un recurso de aprendizaje con capacidad de usar la sub-especificación SCORM RTE [108] para comunicarse con un LMS (**Figura 29**). Es el elemento más sencillo al que un LMS puede realizar un seguimiento y comunicarse con él. Las comunicaciones se basan en la API ECMAScript de IEEE [110]. Para aumentar su portabilidad, un SCO debe ser independiente del resto de contenido. Al igual que los Assets pueden ser descritos mediante metadatos para poder ser descubiertos y reutilizados, un SCO debe estar creado siguiendo las normas descritas en SCORM RTE para que pueda localizar una API proporcionada por el LMS e invocar al menos las funciones de inicialización y terminación de comunicación.

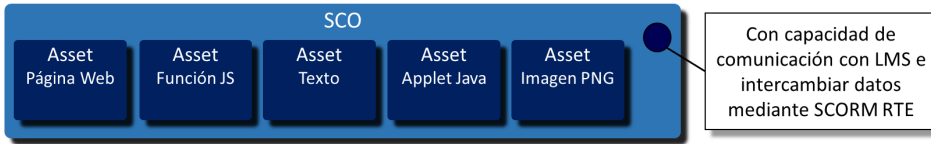


Figura 29. Ejemplo de SCO

- **Agregación.** Es una colección de SCOs relacionados entre sí (**Figura 30**). Puede considerarse equivalente a un curso o sección de un curso. Su uso no es obligatorio. Pueden anidarse.

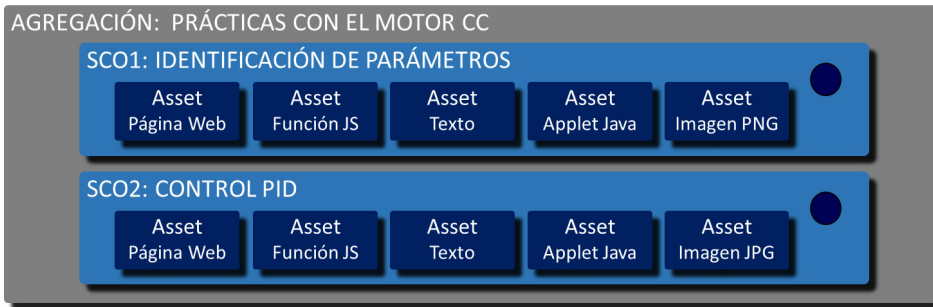


Figura 30. Ejemplo de Agregación

Además de estos tipos de contenidos que pueden aparecer en un **paquete SCORM** también es necesario que se incluya una descripción de la forma en que se van a presentar y como se relacionan, para eso es necesario que exista un fichero XML llamado `imsmanifest.xml` que debe estar obligatoriamente en la carpeta raíz de la estructura junto a sus ficheros de definición y soporte, por ejemplo, DTD, XSD. El resto de ficheros asociados a cada *Asset* es conveniente que se estructuren en carpetas para tenerlo mejor organizado y poder reutilizarlos (**Figura 31**).

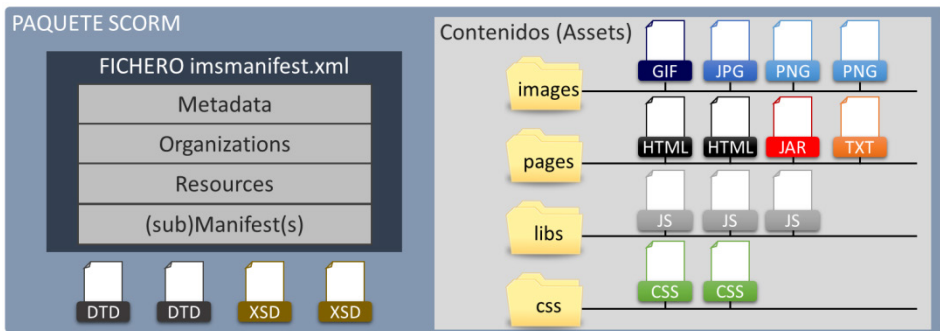


Figura 31. Ejemplo de Paquete de contenido SCORM

Un paquete de contenido SCORM representa una unidad de aprendizaje que puede ser parte de un curso o puede ser una unidad independiente. En el caso más común los SCO son ficheros HTML que llevan asociados otros *Assets* como imágenes, librerías JavaScript, hojas de estilo, etc. El fichero `imsmanifest.xml` contiene un inventario estructurado del contenido del paquete, describe que estructura existe basada en ítems que se suelen corresponderse con SCO y los *Assets* correspondientes. Un fichero `imsmanifest.xml` puede contener 4 secciones principales:

- **Metadata.** Datos que describen el paquete SCORM en su conjunto.
- **Organizations.** Contiene la estructura de contenido o la organización de los recursos de aprendizaje que constituyen una unidad o unidades de formación independientes. También puede contener la definición de la intención de secuenciación que se asocia con la estructura de contenidos.
- **Resources.** Define los recursos de aprendizaje incluidos en el paquete de contenidos SCORM.
- **(sub)Manifest(s).** Describe las unidades de formación anidadas lógicamente (que pueden ser tratadas como unidades independientes).

La **Figura 32** muestra un ejemplo de fichero `imsmanifest.xml`. Se puede observar la forma que presenta este fichero cuando se tiene un paquete SCORM con una estructura que incluye dos ítems de tipo SCO (RES1 y RES2), que en realidad son dos páginas Web (`page1.xhtml` y `page2.html`), y otros recursos en forma de librerías JavaScript (`APIWrapper.js` y `RTE.js`).

2.1.2.1. Sub-especificación SN

Esta sub-especificación SN (Secuenciación y Navegación) describe cómo el LMS debe entregar el contenido compatible con SCORM (SCOs y *Assets*) a los estudiantes durante el tiempo de ejecución mediante un conjunto de eventos de navegación que pueden ser iniciados por el sistema o por el alumno. Para ello define las reglas de secuencia que se pueden añadir al paquete en el fichero `imsmanifest.xml`, define los comportamientos que deben ocurrir en tiempo de ejecución cuando existen reglas de secuenciación en el paquete y presenta un modelo de datos de navegación. Las reglas de secuenciación que se deben incluir en el fichero `imsmanifest.xml` se deben incluir anexas a los elementos a

los que se les aplica. La sub-especificación SN presenta los siguientes contenidos:

```

1  <?xml version = "1.0" standalone = "no"?>
2  <manifest identifier = "LMSTestPackage_T-01b" version = "1.0"
3     xmlns = "http://www.imsglobal.org/xsd/imsdcp_v1p1"
4     xmlns:adlcp = "http://www.adlnet.org/xsd/adlcp_v1p3"
5     xmlns:adlseq = "http://www.adlnet.org/xsd/adlseq_v1p3"
6     xmlns:adlnav = "http://www.adlnet.org/xsd/adlnav_v1p3"
7     xmlns:imsss = "http://www.imsglobal.org/xsd/imsss"
8     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
9     xsi:schemaLocation = "http://www.imsglobal.org/xsd/imsdcp_v1p1 scorm/imsdcp_v1p1.xsd
10        http://www.adlnet.org/xsd/adlcp_v1p3 scorm/adlcp_v1p3.xsd
11        http://www.adlnet.org/xsd/adlseq_v1p3 scorm/adlseq_v1p3.xsd
12        http://www.adlnet.org/xsd/adlnav_v1p3 scorm/adlnav_v1p3.xsd
13        http://www.imsglobal.org/xsd/imsss scorm/imsss_v1p0.xsd" >
14  <metadata>
18  <organizations default="ORG-AAE040C6-5D31-FF80-7A74-C93346D3DE42">
19    <organization identifier="ORG-AAE040C6-5D31-FF80-7A74-C93346D3DE42" structure="hierarchical">
20      <title>PruebaJS SCORM2004v4-2P</title>
21      <item identifier="SCO1" identifierref="RES1" isvisible="true">
43      <item identifier="SCO2" identifierref="RES2" isvisible="true">
63    </organization>
67  </organizations>
68  <resources>
69    <resource>
70      <resource identifier="RES1" href="page1.xhtml" adlcp:scormType="sco" type="webcontent" >
75      <resource identifier="RES2" href="page2.html" adlcp:scormType="sco" type="webcontent" xml:base
80      <resource identifier="API-DEP" adlcp:scormType="asset" xml:base="scripts/" type="webcontent"
81      <resource identifier="API-DEP1" adlcp:scormType="asset" xml:base="scripts/" type="webcontent"
82    </resources>
83  </manifest>

```

Figura 32. Captura de pantalla parcial de ejemplo de fichero imsmainfest.xml

- **Terminología y conceptos relativos a la secuenciación.** Describe conceptos útiles para entender el funcionamiento de la navegación y secuenciación en SCORM SN. Define los conceptos de árbol de actividades, *cluster*, actividad de aprendizaje e intento (*attempt*). Por ejemplo, un árbol de actividades es un diagrama de estructura de contenido que describe las actividades de aprendizaje de un paquete SCORM.
- **Modelo de definición de secuencia.** Describe un conjunto de elementos que pueden ser utilizados por los desarrolladores de contenido para definir comportamientos relacionados con la secuencia de los contenidos del paquete SCORM. De este modo se pueden describir estrategias de secuenciación durante el desarrollo del contenido. Los comportamientos relativos a la secuencia que ofrece un determinado elemento no tiene por qué ser estático. Los valores de los elementos del modelo de definición de secuencia pueden verse alterados por el LMS en función del trabajo del alumno. Esto permite usar SCORM para establecer secuencias de aprendizaje dinámicas entre los contenidos del paquete. También incluye ejemplos de uso. Define los siguientes elementos:

- **Modos de Control de secuenciación.** Define modos permitidos para la secuenciación de elementos del paquete; pueden permitirse varios simultáneamente. Por ejemplo, pueden estar activos (true) los modos *Sequencing Control Choice* y *Sequencing Control Choice Exit* a la vez. De este modo se permite que una petición de navegación de tipo *choice* (Elección) se pueda activar, para ir al elemento hijo de la actividad de aprendizaje actual o para terminar la ejecución.
- **Controles de Elección limitados.** Conjunto de controles de elección que permiten poner condiciones y comportamientos al procesamiento de solicitudes de elección de secuenciación. Por ejemplo si el control de elección restringido *Prevent Activation* se configura como *True*, entonces las peticiones de secuenciación de tipo *choice* sólo debe permitir elementos que estén inmediatamente anexos a la actividad de aprendizaje activa en ese momento en el flujo definido (no puede haber “saltos” a otras actividades que no estén inmediatamente anexas a la actividad en la que se genera la petición).
- **Descripción de reglas de secuenciación.** Reglas que sirven para describir las acciones a realizar cuando se cumplen determinadas condiciones relativas a la secuenciación. Por ejemplo, *If Completed* (condición) *Then Continue* (Acción) establece que si el estudiante completa la actividad activa, es decir, el estado de terminación del intento actual es “*completed*”, la siguiente acción que se debe realizar es continuar, es decir, la siguiente actividad definida en la secuencia lógica es la que debe entregar el LMS.
- **Condiciones límite.** Permite que el desarrollador de contenidos establezca condiciones límite que cuando se cumplan impidan que un estudiante reciba un determinado contenido. Por ejemplo, si para una actividad se define el valor de *Limit Condition Attempt Control* como *true*, y el de *Limit Condition Attempt Limit* como 5, entonces el LMS no podrá entregar esa actividad al estudiante que haya alcanzado ese número de intentos.
- **Descripción de reglas agrupadas.** Cuando existen varias actividades asociadas se pueden definir reglas que las tenga en cuenta como agrupación.

- **Controles agrupados.** Permiten establecer controles sobre actividades relacionadas (p.e. actividad padre-actividades hijo).
- **Controles de consideración de agrupaciones.** Reglas que mejoran las condiciones en las que una actividad contribuye al conjunto de actividades relacionadas con su actividad padre.
- **Descripción de objetivos.** Las actividades pueden tener asociados unos objetivos de aprendizaje, a efectos de secuenciación cada objetivo de aprendizaje tiene unos datos de información de registro de estado que permite tomar decisiones de secuenciación condicionadas basadas en estos valores.
- **Controles de selección.** Permiten que los desarrolladores de contenido definan información de secuenciación que indica cuando seleccionar ciertas actividades y limitar el número de actividades que se pueden elegir. Esto permite crear reglas para establecer la implementación de secuenciación de un LMS. Por ejemplo, elegir 4 de las 6 actividades hijo en el primer intento de una actividad.
- **Controles de aleatorización.** Reglas que describen que acciones llevará a cabo y cuando el LMS para reordenar las actividades hijo disponibles de los *clusters* de actividades que se encuentre.
- **Controles de entrega.** Describen las acciones que realizarán los LMSs antes de que los estudiantes empiecen un intento (*attempt*) de una actividad y antes de que acabe el intento.
- **Comportamientos de secuenciación.** Aquí se incluye el Modelo de comportamiento de secuenciación y Requisitos y controles de navegación. Describe detalladamente la información que registra el LMS relacionada con la secuenciación y la forma en que el progreso del estudiante a través de los objetos del contenido afecta al registro que hace el LMS de esta información.
 - **Modelo de seguimiento.** Es un conjunto de elementos del modelo de datos que describen la información de seguimiento.
 - **Proceso de secuenciación general.** Proporciona el proceso de control general para que el LMS implemente la secuenciación. Define cómo se aplican los distintos comportamientos de secuenciación en el contexto de una sesión de secuenciación. El proceso general de secuenciación incluye los siguientes comportamientos:

- *Comportamiento de Navegación*: describe cómo una solicitud de navegación se valida y se traduce como peticiones de terminación y secuenciación.
 - *Comportamiento de Terminación*: describe cómo termina el intento (*attempt*) actual de un usuario con una actividad, cómo se actualiza el estado del árbol de actividades y si debe llevarse a cabo alguna acción debido a la terminación del intento del usuario (*attempt*).
 - *Comportamiento de agrupación*: describe cómo la información de seguimiento de las actividades agrupadas en *clusters* se deriva de la información de seguimiento de las actividades que lo forman.
 - *Comportamiento de selección y aleatorización*: describe cómo se deben considerar las actividades de un clúster durante el procesamiento de una solicitud de secuenciación.
 - *Comportamiento de secuenciación*: describe cómo una solicitud de secuenciación se procesa en un árbol de actividades de un intento de usuario (*attempt*) para identificar la siguiente actividad que se va a entregar.
 - *Comportamiento de entrega*: describe cómo se valida una actividad identificada para ser entregada y la forma en que un LMS debe realizar la entrega de una actividad validada.
- **Modelo de navegación SCORM.** En SCORM, la experiencia de aprendizaje que se proporciona a los alumnos consiste en la serie de actividades de aprendizaje que experimentan para cada árbol de actividades dado; es decir, en la secuencia de actividades que el secuenciador decida en particular (siguiendo las reglas programadas y las interacciones realizadas por el alumno) para ser entregadas y finalmente lanzadas por el alumno. La implementación de la secuencia del LMS es un componente pasivo del LMS, sólo actúa en respuesta a las peticiones de navegación emitidas por el LMS. La navegación es el proceso por el que un estudiante y un LMS cooperan para identificar las peticiones de navegación para realizar una experiencia de aprendizaje. El LMS puede proporcionar una *interface* de usuario (UI, *User Interface*) que el estudiante podrá usar para indicar sus peticiones de navegación

por los elementos del paquete SCORM. Los desarrolladores de contenido SCORM también pueden incorporar en los contenidos un menú de navegación y mantener el UI del LMS o indicar al LMS que no lo proporcione. Sea como fuese, en todos los casos las peticiones de navegación vienen dadas por acciones realizadas por el estudiante o programadas en el propio contenido a través del árbol de actividades.

- **Eventos de Navegación.** El modelo de navegación SCORM define una serie de 11 eventos que pueden ser desencadenados por un estudiante a través de un LMS y dispositivos UI proporcionados en el contenido o directamente por un SCO (**No se** identifica la actividad de aprendizaje que debe ser entregada: El LMS no tiene que hacer nada aunque puede presentar un aviso.
 - Se produce una excepción durante el procesamiento de la secuenciación: El LMS no tiene que hacer nada aunque podría manejar la excepción.
- **Tabla 2).** Cuando se produce uno de los eventos de navegación el LMS procesa la petición correspondiente invocando al sistema de secuenciación y el resultado del mismo puede ser una de las siguientes opciones:
 - Si recibe petición de navegación *Exit All*: Finaliza el intento actual y devuelve el control al LMS.
 - Si después de evaluar el estado actual de seguimiento y la información de secuencias válidas del árbol de actividades, el LMS determina que el procesamiento de la solicitud de navegación prevista no debe ser respetado: Ignora la petición de Navegación:
 - Si después de evaluar el estado actual de seguimiento y la información de secuencias válidas del árbol de actividades, el LMS determina que el procesamiento de la solicitud de navegación prevista puede ser respetado. El LMS invoca el procesamiento de secuenciación general y pueden pasar 3 cosas:
 - Se identifica la actividad de aprendizaje que debe ser entregada: el LMS prepara y lanza el objeto de contenido asociado a la actividad.

- No se identifica la actividad de aprendizaje que debe ser entregada: El LMS no tiene que hacer nada aunque puede presentar un aviso.
- Se produce una excepción durante el procesamiento de la secuenciación: El LMS no tiene que hacer nada aunque podría manejar la excepción.

Tabla 2. *Eventos de Navegación SCORM*

Evento de Navegación	Descripción de Comportamiento	Origen
Start	Indica deseo de identificar la primera actividad disponible en el árbol de actividades. Generado por el LMS cuando el estudiante empieza un nuevo intento en la actividad raíz del árbol de actividades. Este evento genera una petición de navegación <i>Start</i> .	LMS
Resume All	Indica deseo de continuar un intento anteriormente suspendido en la actividad raíz del árbol de actividades. Es generado automáticamente por el LMS cuando el estudiante reanuda un intento anteriormente suspendido en el árbol de actividades. Este evento genera una petición de navegación <i>Resume</i> .	LMS
Continue	Indica deseo de identificar la próxima (respecto a la actividad en la que está) actividad de aprendizaje lógica disponible en el árbol de actividades. Este evento genera una petición de navegación <i>Continue</i> .	LMS o SCO
Previous	Indica deseo de identificar la actividad de aprendizaje anterior (respecto a la actividad en la que está) lógica disponible en el árbol de actividades. Este evento genera una petición de navegación <i>Previous</i> .	LMS o SCO
Choose	Indica deseo de acceder directamente a una actividad de aprendizaje específica del árbol de actividades. Este evento genera una petición de navegación <i>Choice</i> para una determinada actividad.	LMS o SCO
Jump	Indica deseo de saltar directamente a una actividad de aprendizaje específica del árbol de actividades. Este evento genera una petición de navegación <i>Jump</i> para una determinada actividad.	SCO
Abandon	Indica deseo de terminar el intento actual de forma prematura o anormalmente en el objeto de contenido que está actualmente sin intención de reanudarlo más tarde. Este evento finaliza el intento sobre la actividad en ejecución. Si la actividad tiene un padre el intento en la actividad padre no se finaliza. <i>Abandon</i> no tiene efectos inmediatos en ninguna de las actividades predecesoras (padres). Un intento abandonado se considera un intento. Este evento genera una petición de navegación <i>Abandon</i> .	LMS or SCO
Abandon All	Indica deseo de terminar el intento actual y todas las actividades de aprendizaje activas de forma prematura o anormalmente en el objeto de contenido que está actualmente sin intención de reanudarlo más tarde. Este evento finaliza el intento sobre la actividad en ejecución. Un intento abandonado se considera un intento. Este evento genera una petición de navegación <i>Abandon All</i> .	LMS or SCO

Suspend All	Indica deseo de detener el intento actual con intención de reanudarlo más tarde. Suspende todas las actividades activas. Si el próximo intento se inicia con un evento <i>Resume All</i> , los intentos en todas las actividades suspendidas por este evento se reanudarán. En ningún caso se perderá la información de seguimiento que hubiera sido grabada antes en el LMS. Este evento genera una petición de navegación <i>Suspend All</i> .	LMS or SCO
Unqualified Exit	Indica que el intento actual en la actividad activa ha finalizado normalmente, sin que haya sido desencadenado por otro evento de navegación como <i>Continue</i> , <i>Previous</i> , <i>Jump</i> o <i>Choose</i> . Este evento genera una petición de navegación <i>Exit</i> .	LMS or SCO
Exit All	Indica que el intento actual en la actividad raíz del árbol de actividades ha finalizado normalmente, también finaliza todas las actividades de aprendizaje activas. Este evento genera una petición de navegación <i>Exit All</i> .	LMS or SCO

- **Comunicaciones en tiempo de ejecución de peticiones de navegación.** Un SCO, o un *Asset* activo incluido en un SCO (por ejemplo un *applet* de Java o una aplicación JavaScript, a través de la API SCORM del SCO puede comunicarse directamente con el LMS. Esta comunicación permite indicar una petición de navegación consultar cuáles son las posibilidades de navegación posibles. Para ello debe hacer uso de funciones de comunicación definidas en la sub-especificación SCORM RTE, y datos del modelo de datos navegación que se verán a continuación:
 - **Modelo de datos de navegación.** Tiene los siguientes elementos:
 - **adl.nav.request.** Es un elemento de lectura/escritura que puede tener uno de los siguientes valores: *continue*, *previous*, *choice*, *jump*, *exit*, *exitAll*, *abandon*, *abandonAll*, *suspendAll* o *_none_*. Cuando un SCO tiene una sesión de comunicación con el LMS puede actualizar el valor de este elemento en el LMS poniendo uno de estos valores está haciendo que se active el evento correspondiente (No se identifica la actividad de aprendizaje que debe ser entregada: El LMS no tiene que hacer nada aunque puede presentar un aviso.
 - Se produce una excepción durante el procesamiento de la secuenciación: El LMS no tiene que hacer nada aunque podría manejar la excepción.

- **Tabla 2)** para que el LMS ejecute la petición de navegación correspondiente tras terminar la sesión de comunicación SCO-LMS. Al leerlo puede conocer cuál es el siguiente destino de navegación programado tras el SCO actual.
- **adl.nav.request_valid.continue,**
adl.nav.request_valid.previous,
adl.nav.request_valid.choice.{target=<string>}
y adl.nav.request_valid.jump.{target=<string>}
Son elementos de sólo lectura que puede tener uno de los siguientes valores: *true*, *false* o *unknown*. Cuando desde un LMS se lee el valor de uno de estos datos almacenado en el LMS puede conocer la posibilidad de hacer una petición de navegación del tipo correspondiente.

2.1.2.2. Sub-especificación RTE

La sub-especificación SCORM RTE, *Run Time Environment* (Entorno en tiempo de ejecución) [108], describe los requisitos que debe cumplir un LMS compatible para gestionar la ejecución de los paquetes SCORM y los requisitos que debe cumplir un SCO para usar la API SCORM (*Application Program Interface*) que proporciona el LMS. La API permite que se produzcan las comunicaciones SCO-LMS con el fin de intercambiar datos del modelo de datos RTE. Para ello presenta 3 secciones principales donde se explica la entrega de los objetos de contenido SCORM que realiza el LMS al navegador web del estudiante (gestión del RTE), la forma de comunicarse entre un objeto de contenido con el LMS (API SCORM) y la información almacenada por el LMS para cada objeto de contenido (Modelo de datos RTE).

- **Gestión del RTE.** Cuando un alumno interactúa con un contenido SCORM, el LMS evalúa el rendimiento del alumno y las peticiones de navegación (Sub-especificación SN). Cuando el LMS identifica una actividad que debe entregar al alumno, lanza el contenido asociado a esa actividad para presentárselo al alumno. El contenido, que puede ser de tipo SCO o *Asset* se presenta en el entorno del navegador del alumno. Para entender el proceso de seguimiento del alumno durante

una experiencia de aprendizaje se han definido unos términos que se muestran gráficamente en el modelo temporal de la **Figura 33**:

- **Sesión de login en LMS**: periodo de tiempo durante el cual el alumno se identifica en un LMS hasta que sale del sistema.
- **Intento de Alumno**: experiencia de aprendizaje en un paquete SCORM en el que trata de cumplir los requerimientos de una actividad de aprendizaje que utiliza un objeto de contenido. Un intento puede contener una o más sesiones de alumno y puede suspenderse entre sesiones.
- **Sesión de Alumno**: periodo de tiempo ininterrumpido durante el cual el alumno está accediendo a un objeto de contenido, normalmente un SCO, en este caso el LMS debe proporcionar una instancia de la API SCORM para permitir las comunicaciones SCO-LMS.
- **Sesión de Comunicación**: una conexión activa entre un objeto de contenido (por ejemplo un SCO) y una instancia de la API SCORM. Comienza en el momento en el que desde el SCO se hace una llamada a la función *Initialize()* con éxito y finaliza al llamar a la función *Terminate()*, ambas de la API SCORM.

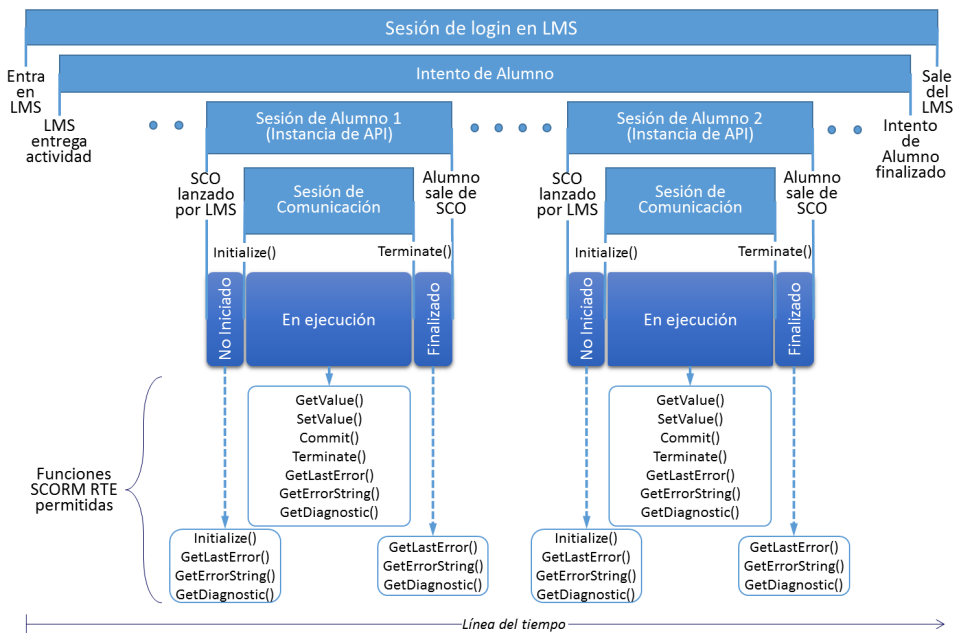


Figura 33. Modelo temporal del RTE SCORM

- **API SCORM:** la API SCORM consiste en una serie de 8 funciones predefinidas que los LMSs deben implementar como estimen oportuno para poder proporcionar a todos los SCO una instancia operativa de la misma. Para que los SCO puedan usar la implementación de la API SCORM deben hacer una búsqueda de la misma. Los navegadores web proporcionan un Modelo de Objetos del Documento, más conocido como DOM (*Document Object Model*), que permite acceder a todos los elementos de una página Web. Cuando el LMS entrega el objeto de contenido al alumno incluye una instancia de la API SCORM en el DOM que el SCO debe encontrar para poder usar las funciones. Las 8 funciones de la API se pueden estructurar de la siguiente forma:
 - **Funciones de sesión:** se usan para marcar el principio y fin de una sesión de comunicación entre un SCO y el LMS a través de la API SCORM. Son dos:
 - **Initialize().** Se usa para iniciar una sesión de comunicación SCO-LMS. Devuelve “*true*” si tiene éxito, y en caso contrario “*false*” y almacena un código de error que podrá ser recuperado con la función *GetLastError()*.
 - **Terminate().** Se usa para cerrar una sesión de comunicación SCO-LMS. Devuelve “*true*” si tiene éxito, y en caso contrario “*false*” y almacena un código de error que podrá ser recuperado con la función *GetLastError()*.
 - **Funciones de transferencia de datos:** se utilizan para intercambiar datos del modelo de datos RTE entre un SCO y un LMS a través de la API SCORM. Son 3:
 - **GetValue(parameter).** Se usa para pedir el valor de un dato almacenado en el LMS del modelo de datos RTE de SCORM. Devuelve “*true*” si tiene éxito, y en caso contrario “*false*” y almacena un código de error que podrá ser recuperado con la función *GetLastError()*.
 - **SetValue(parameter1, parameter2).** Se usa para pedir al LMS que almacene el valor indicado en *parameter2* en el dato del modelo de datos RTE de SCORM indicado en *parameter1*. Devuelve “*true*” si tiene éxito, y en caso contrario “*false*” y almacena un código de error que podrá ser recuperado con la función *GetLastError()*.
 - **Commit(parameter).** Se usa para asegurar el almacenamiento de datos persistente para todos los

valores que se hayan pedido almacenar y pudieran estar en la caché de la instancia de la API SCORM desde la última vez que se utilizó esta misma función o *Initialize()*, lo que sea lo más reciente. Si la instancia de la API no guarda datos en caché devuelve “true” y el código de error 0, también devuelve “true” si tiene éxito almacenando los datos, y en caso contrario “false” y almacena un código de error que podrá ser recuperado con la función *GetLastError()*.

- **Funciones de soporte:** se usan para comunicaciones auxiliares SCO-LMS a través de la instancia de la API SCORM. Son 3:
 - **GetLastError().** Se usa para pedir el código de error del estado de error actual de la instancia de la API que es el valor que devuelve (**Tabla 3**).
 - **GetErrorString(parameter).** Se usa para pedir un texto con la descripción del estado de error indicado en *parameter* que es el valor que devuelve en forma de cadena de caracteres.
 - **GetDiagnostic(parameter).** Es una función de uso específico del LMS que le permite definir una información de diagnóstico adicional para el error que se indique en *parameter* (máximo 255 caracteres) que es lo que devuelve.

Tabla 3. Categorías y Rangos de Códigos de Error de la API SCORM

Categoría de Código de Error	Rango de Código de Error
No hay Error	0
Errores Generales	100-199
Errores de Sintaxis	200-299
Errores RTS	300-399
Errores del Modelo de Datos	400-499
Errores definidos por implementación	1000-65535

Estas funciones no se pueden usar todas en todo momento, depende del estado de la sesión de comunicación. En la **Figura 33** se pueden ver las funciones que se pueden usar en cada uno de los estados en los que se puede encontrar una sesión de comunicación:

- **No Iniciado.** Desde que el LMS lanza el SCO y éste encuentra la instancia de la API SCORM hasta que se realiza una llamada a la función *Initialize()* con éxito.
- **En Ejecución.** Desde que el SCO realiza con éxito una llamada a la función *Initialize()* hasta que se realiza otra llamada con éxito a la función *Terminate()*.
- **Finalizado.** Desde que el SCO realiza con éxito una llamada a la función *Terminate()* hasta que el alumno sale del SCO.
- **Modelo de datos RTE.** El propósito de establecer un modelo de datos común es asegurar que exista definido un conjunto de información sobre los SCOs que pueda ser utilizado por diferentes sistemas LMS para realizar el registro del seguimiento de las experiencias de aprendizaje de la misma forma. El uso de los elementos del modelo de datos es totalmente opcional por parte de los SCOs, el único requisito que tienen es el uso de las funciones *Initiate()* y *Finalize()*. Los SCOs pueden afectar directamente a la secuenciación del paquete SCORM registrando los resultados de las interacciones del alumno durante las sesiones de alumno en el mismo SCO. La **Tabla 4** muestra los elementos que hay en el modelo de datos RTE de SCORM.

Tabla 4. Elementos del modelo de datos RTE

Elemento del Modelo de Datos	Notación oficial del Elemento	Descripción
Comments From Learner	cmi.comments_from_learner	Texto del estudiante.
Comments From LMS	cmi.comments_from_lms	Comentarios del LMS para el estudiante.
Completion Status	cmi.completion_status	Indica el estado de terminación del SCO.
Completion Threshold	cmi.completion_threshold	Valor usado para determinar si el SCO debe considerarse completado.
Credit	cmi.credit	Indica si al estudiante se le acreditará su trabajo en el SCO.
Entry	cmi.entry	Indica si el estudiante ha entrado antes en el SCO.
Exit	cmi.exit	Indica como o porque el estudiante dejó el SCO.
Interactions	cmi.interactions	información relativa a una interacción con propósito de medición o evaluación.
Launch Data	cmi.launch_data	Datos específicos de un SCO que puede usar para la inicialización.
Learner Id	cmi.learner_id	Identificación del estudiante en el LMS.
Learner Name	cmi.learner_name	Nombre del estudiante en el LMS.
Learner Preference	cmi.learner_preference	Preferencias del alumno sobre el uso del SCO.
Location	cmi.location	Localización en el SCO.

Maximum Time Allowed	cmi.max_time_allowed	Tiempo acumulado que el alumno puede pasar en el SCO en el intento.
Mode	cmi.mode	Identifica uno de los tres modos en que se puede presentar el SCO al alumno.
Objectives	cmi.objectives	Objetivos de aprendizaje o rendimiento asociados con un SCO.
Progress Measure	cmi.progress_measure	Identifica una medida del progreso que el alumno ha realizado para completar el SCO.
Scaled Passing Score	cmi.scaled_passing_score	Puntuación de superación del SCO normalizada.
Score	cmi.score	Puntuación alcanzada por el estudiante en el SCO.
Session Time	cmi.session_time	Tiempo que el alumno ha estado en la sesión de alumno actual en el SCO.
Success Status	cmi.success_status	Indica si el alumno ha dominado el SCO.
Suspend Data	cmi.suspend_data	Información creada por un SCO como resultado del acceso o interacción del alumno en el SCO.
Time Limit Action	cmi.time_limit_action	Indicates what the SCO should do when the maximum time allowed is exceeded.
Total Time	cmi.total_time	Suma total de los tiempos de sesión de alumno acumulados en el intento anterior al actual.

La mayoría de los elementos del modelo de datos RTE son simples, pero los siguientes se presentan en colecciones que se manejan como *arrays* indexados a partir del elemento 0:

- Comments from learner (cmi.comments_from_learner)
- Comments from LMS (cmi.comments_from_lms)
- Objectives (cmi.objectives)
- Interactions (cmi.interactions)
- Shared Data Stores (adl.data)

2.2. Laboratorios Docentes *Online*

Como ya se ha comentado en la sección 1.1 de este trabajo (Contextualización), en los programas universitarios de ingenierías y ciencias aplicadas el trabajo experimental realizado en los laboratorios es esencial para la formación de los estudiantes [3], [17], [111].

2.2.1. Tipos de Laboratorios *online* y Terminología

Existen diferentes formas de presentar los laboratorios a los estudiantes que han traído consigo diferentes modalidades de laboratorios. La caracterización

de estas modalidades se ha basado tradicionalmente en dos criterios, que ya apuntó Dormido [1], y que han sido utilizados por otros autores [18], [28]:

- Según el modo de acceso a los recursos: local o remoto.
- Según la naturaleza física del recurso: real o simulado.

Esto ha permitido identificar los tipos de laboratorios estableciendo 4 modalidades básicas:

- **Laboratorios Presenciales en planta real (*hands-on labs*):** acceso local a recurso real.
- **Laboratorios Presenciales en simulación:** acceso local a recurso simulado en PC de laboratorio, también llamado laboratorio virtual mono-usuario.
- **Laboratorio remoto en planta real:** acceso remoto a recurso real, se suele denominar simplemente como **laboratorio remoto**.
- **Laboratorio virtual en simulación:** acceso remoto a recurso simulado multi-usuario, se suele llamar simplemente **laboratorio virtual**.

Sin embargo, en la literatura hay algunas discrepancias relacionadas con el uso de estos términos para definir diferentes tipos de laboratorios; por ejemplo, en [35], se usa el término “*virtual lab*” para referirse a laboratorios remotos, mientras que en [23] se usa el término “*online lab*” para referirse exclusivamente a experimentos remotos en plantas reales cuando normalmente se usa para describir laboratorios utilizados a través de Internet sin distinguir la naturaleza del recurso (real o simulado).

Además, el avance de la tecnología ha traído consigo otra serie de criterios que también se pueden tener en cuenta a la hora de definir los entornos y características de los laboratorios como puede ser el uso de alguna tecnología como la realidad virtual, tecnología Web, integración con LMS, etc.

Por este motivo se ha creído necesario definir los siguientes términos utilizados para referirse a laboratorios *online* (**Figura 34**):

- **Laboratorios en línea, Laboratorio *online* o LoL (*online labs, Virtual/Remote Lab o VRL*):** laboratorio utilizado a través de Internet sin distinción del tipo de recurso al que se accede (puede ser real, simulado o ambos) [112], [113], [114].

- **Laboratorios Virtuales** (*Virtual Lab* o VL). VRL con el que se accede a una simulación [115], [116], [115].
- **Laboratorios Remotos** (*Remote Lab* o RL). VRL con el que se accede a un sistema real [75], [117], [118].
- **Laboratorios Híbridos** (*Hybrid Lab* o HL). VRL en el que se accede a un sistema real y a una simulación en el mismo entorno [119], [28], [120].
- **Laboratorio Remoto de Realidad Aumentada** (*Augmented Remote Lab* o ARL). RL en el que se ofrece una visión de un entorno físico real en la que mediante ordenador se ha añadido algún elemento sensorial en forma de sonido, gráfico o texto [121], [122]. Por definición los ARL son RL, aunque en los casos en los que el elemento añadido por el equipo sea una simulación se podría considerar un HL.
- **Laboratorio Web** (*Web-based lab*, *Web Laboratory* o **WebLab**). VRL al que se accede mediante tecnología Web, normalmente un navegador [76], [123], [124], [125].

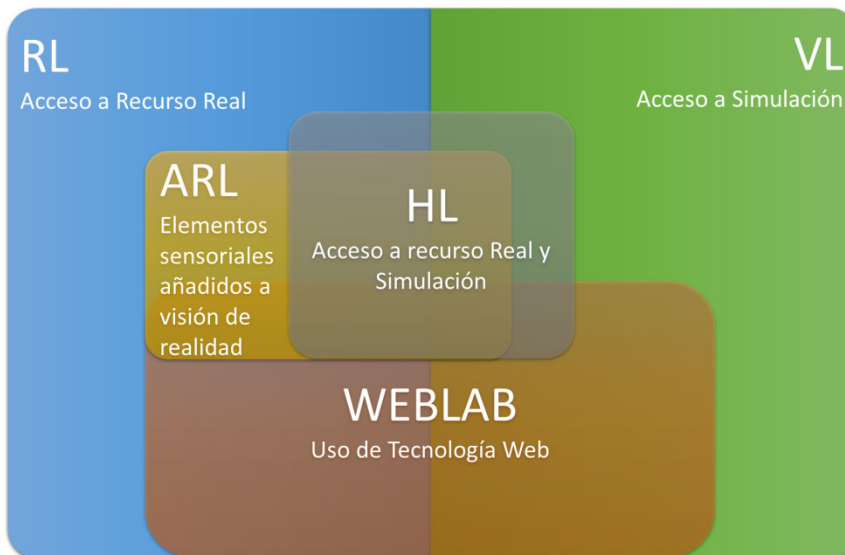


Figura 34. Tipos de Laboratorios online

Los laboratorios *online* presentan un amplio conjunto de ventajas que han sido señaladas por múltiples autores como son la disponibilidad espacial y temporal, seguridad, extensión del uso de recursos escasos, accesibilidad, observación/grabación, compartición, motivaciones [20] y algunos problemas como la dificultad de adquirir habilidades hápticas y las sensaciones que dan el manejo real de la instrumentación [23].

Sin embargo, dependiendo de la modalidad de laboratorio utilizada pueden presentar ventajas o inconvenientes asociados. Los laboratorios virtuales añaden algunas ventajas a las mostradas anteriormente que no presentan los laboratorios remotos, como son [126], [127], [128], [129]:

- **Modelos personalizables.** Permiten simplificar o modificar los modelos de los sistemas reales con fines educacionales, resaltando y haciendo más visibles aquellas características de la experimentación que más interesan para lograr que la efectividad del aprendizaje sea mayor o se adapte a las capacidades y niveles del alumno.
- **Mejor visibilidad.** Permiten mostrar fenómenos, dispositivos y experimentos que bien por su tamaño (muy pequeño o muy grande), o dificultad de acceso (el interior de un motor en funcionamiento), son imposibles de apreciar [130] [131]. Por otro lado, también permite eliminar o minimizar detalles que puedan confundir al estudiante [132].
- **Configuraciones imposibles.** Se pueden modificar los valores de los parámetros o combinar dispositivos que en el mundo real sería imposible de realizar. Por ejemplo, cambiando la escala temporal se puede comprobar el efecto del tiempo, como el paso de años, en un instante.
- **Preparación pre-laboratorio.** Los laboratorios virtuales son ideales para preparar al alumno antes de acceder a un laboratorio presencial o remoto en el que deba trabajar con equipos potencialmente peligrosos o complicados [23], [133], [134].
- **Resistencia a daños.** Ningún equipo real puede salir estropeado al trabajar con simulaciones.
- **Acceso múltiple.** No existe ningún tipo de limitación respecto al número de usuarios que pueden acceder a la vez a un equipo virtual al ser una simulación que se ejecuta localmente una vez descargada.
- **Repetitividad.** Permite repetir un mayor número de experimentos de forma rápida y fácil, comprobando los resultados obtenidos, modificando los parámetros necesarios y reproducir el experimento inmediatamente.
- **Coste.** Aunque es una característica general de los laboratorios *online* en el caso de los virtuales es más apreciable al no tener que tratar con sistemas reales, siempre más costosos.

Por otro lado también se tienen algunas **desventajas en laboratorios virtuales**:

- **Requerimientos.** Algunas simulaciones pueden requerir ciertos recursos computacionales que son muy costosos y no funcionan bien sin el equipo adecuado. Esto suele ser habitual en sistemas modelados de realidad virtual 3D.
- **Actitud de estudiantes.** Al no tratarse de sistemas reales y ya que los estudiantes son conscientes de ello pueden actuar con falta de seriedad, sin la debida responsabilidad y de forma descuidada. Cuando se trabaja con sistemas reales, ya sea de forma remota o local, los estudiantes suelen tener otra actitud ante los experimentos.
- **Falta de realidad.** Algunas simulaciones no se realizan correctamente de forma que dejan de mostrar ciertos aspectos que sí se dan al trabajar con sistemas reales como puede ser la existencia de ruidos y perturbaciones.
- **Experiencia incompleta.** Normalmente la última fase de un proceso de formación requiere el uso o manipulación de equipos reales, algo que en ningún caso se podrá conseguir con las simulaciones.

Por otro lado también existen **desventajas en el uso de los laboratorios remotos** que se pueden considerar exclusivas de esta modalidad:

- **Necesidad de regular su uso.** Ya que en los laboratorios remotos se va a utilizar recursos reales que normalmente no existen en número ilimitado es necesario implementar un sistema de reservas o regular el acceso a los mismos mediante intervalos temporales limitados.
- **Coste de desarrollo y mantenimiento.** Normalmente el montaje de un laboratorio remoto requiere un esfuerzo importante en trabajo técnico de personal, tiempo y económico que suele depender del equipo a manejar remotamente. Además se debe disponer de un espacio físico en el que se deben instalar los equipos reales y al que se debe proporcionar acceso de red. Una vez desarrollado y en funcionamiento también suele requerir un coste de mantenimiento, aunque en este caso puede llegar a ser bastante escaso.

Diversos autores han analizado los resultados que se obtienen en los laboratorios *online* y en los laboratorios presenciales y han llegado a la conclusión de que son comparables [25], [135], [136], [137], [138], [139], [140] y que aportan ventajas a la formación tradicional [128]. En el año 2015, Brinson [141] realizó una revisión de la investigación empírica y llegó a la conclusión de

que en la mayoría de los estudios los resultados obtenidos por los estudiantes en los laboratorios no tradicionales (*online* o VRL) son iguales o superiores a los obtenidos en los tradicionales o presenciales. Esta conclusión se mantiene para todas las categorías de resultados contemplados: conocimiento y comprensión, habilidades de investigación, habilidades prácticas, percepción, capacidad de análisis, y comunicación científica y social.

2.2.2. Tecnologías de Desarrollo y Comunicación

Los primeros laboratorios *online* se implementaron con comunicaciones específicas implementadas sobre TCP/IP, los protocolos base de Internet [142] [143]. Sin embargo, pronto empezaron a ser accedidos vía web [144], [145], [146] y a tratar las tecnologías con las que se implementaban los laboratorios remotos [18], como son el modelo cliente-servidor [145]. Durante un tiempo los trabajos de investigación se centraron en la parte técnica del desarrollo y creación de los laboratorios *online*. En [147] se hace una revisión de las tecnologías utilizadas, analiza *software* para crear clientes y servidores de laboratorios, considera Java, .NET y Python y tecnologías de comunicación como *Web services*.

Sin duda las tecnologías más utilizadas para crear y desarrollar los laboratorios *online* son las siguientes:

- **LabVIEW**, [144], [148], [149], [150], [151], [152], [153], [154], [155], [156], [157].
- **Matlab/Simulink**, [150], [152], [153], [158], [159], [160], [161], [162], [163], [164], [165].
- **Java**, [35], [160], [166], [167], [168], [169]. En este sentido conviene destacar el uso del *software EJS* [63], para la obtención de programas Java utilizados sobre como clientes de laboratorios *online* [53], [77], [115], [152], [161], [170], [171], [172], [173], [174], [175], [176], [177].

Además de las tecnologías señaladas se han utilizado otras como **Flash** [178], [179], [180], ahora en desuso, o **AJAX** [113], [181], [182], en pleno auge, resultado del uso de JavaScript asíncrono y XML. Otra tecnología que también es muy utilizada son los **Web services**, empleados para comunicaciones a través de la red, sobre todo como forma de acceso a laboratorios y entre sistemas que se quieren compartir [148], [179], [183]. Todas estas tecnologías no son

excluyentes, sobre todo cuando se trata de laboratorios remotos es frecuente que se utilicen simultáneamente varias de los métodos indicados [148], [150], [152], [153]. En este sentido conviene señalar el desarrollo del servidor **JIM** [36], que permite utilizar Java como *software* de cliente de un laboratorio remoto para conectarse con el servidor cuando se ha implementado con Matlab/Simulink, y el servidor **JIL** [154], que permite utilizar Java como *software* de cliente de un laboratorio remoto para conectarse con el servidor cuando se ha implementado con LabVIEW. En el lado del cliente, cuando se trata de un Weblab, además de utilizar alguna de las tecnologías indicadas, siempre ha sido necesario el uso de **HTML** y hojas de estilo en cascada (**CSS**) para crear las páginas web, y JavaScript con bastante frecuencia [165]. A finales de 2015, algunos navegadores empezaron a dar problemas con las tecnologías basadas en el uso de *plugins* de navegador como los necesarios para ejecutar *applets* de Java o Flash. Este hecho es un factor importante que ha determinado un aumento en el uso de **JavaScript** [184], [181]. Aunque el uso de JavaScript no se limita al cliente ya que también hay propuestas donde los servidores se han desarrollado empleando esta tecnología [185].

2.2.3. Iniciativas, Arquitecturas, Redes y Plataformas

Con el fin de compartir recursos asociados a los laboratorios *online* existen centros y organizaciones que han desarrollado iniciativas, redes, plataformas, arquitecturas y/o interfaces que se ponen a disposición de otros organismos para acceder a experimentos, crear laboratorios remotos o acceder a dispositivos físicos remotamente. El término RLMS (*Remote Laboratory Management System*, Sistema de Gestión de laboratorios Remotos) se usa para referirse a un sistema que administra diferentes laboratorios remotos, proporciona autenticación, autorización, gestión de usuarios, registro de usuarios, planificación y APIs para desarrollar nuevos laboratorios sobre las mismas. Algunos ejemplos de RLMSs son el ILab del MIT, Labshare Sahara y WebLab-Deusto que se comentarán más adelante en este mismo apartado. En [127] se incluye una recopilación bastante amplia de los asociados a laboratorios virtuales, y en [186] se hace una comparativa de las principales arquitecturas de sistemas para laboratorios remotos. A continuación se describen brevemente algunas de las iniciativas, redes, plataformas y arquitecturas más importantes que existen:

DCL (*Distributed Control Lab*, Laboratorio de Control Distribuido, <http://www.dcl.hpi.uni-potsdam.de/research/dcl/>), [31]. Se trata de una infraestructura abierta para la realización de experimentos de control y robótica basado en *Web services*. El DCL ha sido desarrollado por el *Operating Systems and Middleware Group* (Grupo de Sistemas Operativos y Middleware) del HPI (*Hasso Plattner Institut*) de la Universidad de Postdam en Alemania. Este grupo, junto a 4 universidades alemanas, están desarrollando el laboratorio distribuido DISCOURSE en el que se quieren integrar los experimentos del DCL. Algunos de los trabajos desarrollados sobre este sistema son [187] y [188]. A pesar de tener el mismo acrónimo, no se debe confundir con el *Distributed Control Lab* (DCL) de la Universidad de Purdue en Estados Unidos, un laboratorio dedicado a la investigación en el diseño y control de sistemas de redes complejas de gran escala.

Go-Lab (*Global Online Science Labs for Inquiry Learning at School*, Laboratorios online de Ciencia globales para aprendizaje de investigación en la escuela, <http://www.golabz.eu>), [189]. Es un proyecto financiado con fondos europeos para la creación de un portal web en el que se ofrecen laboratorios de ciencias en línea (laboratorios remotos y virtuales) para su uso a gran escala en la educación escolar. El objetivo general del proyecto es animar a los jóvenes de edades comprendidas de 10 a 18 para participar en la ciencia, adquirir habilidades de investigación científica, y experimentar la cultura de hacer ciencia mediante la realización de la experimentación guiada activa. Aunque está dirigido a jóvenes se ha incluido porque tiene conexiones con otras de las redes presentadas en esta relación.

iLab (<http://icampus.mit.edu/projects/iLabs.shtml>), [24]. Es un proyecto desarrollado en el MIT (Instituto Tecnológico de Massachusetts), cuyo mayor exponente es la arquitectura ISA (*iLab Shared Architecture*, Arquitectura iLab compartida) que fue definida por Harvard en el 2008 [183] ISA facilita el desarrollo de nuevos laboratorios web y proporciona mecanismos para que los estudiantes de una universidad puedan utilizar experimentos y dispositivos físicos de otra universidad. Para ello ISA presenta una estructura parecida a una arquitectura web de 3 niveles (experimento por lotes): el primer nivel lo constituye la aplicación cliente del usuario que ejecuta en su puesto de trabajo tras descargarlo de la web. El nivel intermedio se llama *Service Broker* (agente servicio). Esta apoyado por una base de datos relacional estándar tipo SQL Server u Oracle. El *Service Broker* se encarga de la autenticación del usuario,

comprueba sus permisos para acceder a un determinado servidor de laboratorio, acepta la especificación del experimento que se le envía y espera la respuesta del sistema para enviarla al cliente. El tercer nivel lo constituye el servidor de laboratorio que ejecuta los experimentos especificados y notifica al *Service Broker* cuándo puede recoger los resultados. Este proyecto está aún muy activo y probablemente sea el más extendido y utilizado en el mundo. Además permite integración con otras redes [148] y sistemas como VISIR [190] o ISILab. Existen múltiples trabajos científicos en los que se ha trabajado con esta arquitectura [191], [185], [192].

ISILab (*Internet Shared Instrumentation Laboratory*, Laboratorio de Instrumentación compartido en Internet), <http://isilab.dibe.unige.it/>), [29]. Esta iniciativa depende del DIBE (*Department of Biophysical and Electronic Engineering*) de la Universidad de Génova en Italia. El proyecto original data de 1998, año en el que se desarrolló un laboratorio remoto que evolucionó para dar soporte multiusuario con una aplicación de cliente Java y LabVIEW en el servidor. Para aumentar su usabilidad ha tenido actualizaciones en 2002 para obtener soporte con XML [193] y en 2004 para trabajar con *Web services*. Hoy día se ha producido una integración con el proyecto iLab del MIT. Existen diversos trabajos en los que se menciona el uso de ISILab y su integración con un LMS Moodle [69] o comparándolos con otros sistemas [194].

Lab2go (<http://www.lab2go.net>). Este proyecto está basado en la creación y mantenimiento de un repositorio creado en el marco del proyecto Ontowiki financiado por la UE (Colaboración Semántica para la Gestión del Conocimiento Empresarial, *E-learning* y E-Turismo). Este repositorio en línea ofrece un marco común para recoger y describir los datos de laboratorio de diferentes proveedores de laboratorio ubicados en todo el mundo. El objetivo del mismo es promover el desarrollo de laboratorios y el intercambio de conocimientos entre las partes interesadas. El Portal Web aloja información sobre proyectos actuales de investigación, investigadores, organizaciones y estado actual de las tecnologías más avanzadas con el fin de fortalecer la colaboración en este campo de la ciencia. Un objetivo de la plataforma es la mejora del proceso de búsqueda de laboratorios en línea con el uso de las tecnologías de la Web Semántica. Mediante descripciones de Web semántica se obtienen nuevas formas de realizar mecanismos de búsqueda como es la navegación basada en facetas, que permite al usuario buscar información en función de las propiedades de un objeto especial. Además, es posible consultar recursos en

base a criterios específicos en la siguiente dirección web: aunque no parece estar operativa de forma continua [28].

Labshare (Sahara, <http://www.labshare.edu.au/>), [195], [196], [197]. Se trata de una iniciativa conjunta de la red de la tecnología de Australia dirigida por la Universidad de Tecnología de Sydney en la que participan otros centros: la Universidad de Tecnología de Curtin, la Universidad de Tecnología de Queensland, La Universidad RMIT y la Universidad de Australia del Sur. Este proyecto tiene como objetivo crear una red nacional de laboratorios de acceso remoto compartido. Esta red ofrece experimentos educativos basados en laboratorio de alta calidad que están disponibles para los estudiantes universitarios y de secundaria de cualquier parte de Australia y del mundo.

LiLa (*Library of Laboratories*, Biblioteca de Laboratorios, <http://www.lila-project.org/>), [26]. La Unión Europea mediante el programa comunitario eContentPlus co-financia a un consorcio formado por universidades y empresas de 7 países europeos liderados por la Universidad de Stuttgart encargada de llevar a cabo el proyecto LiLa. Se trata de un portal de recursos de laboratorios en línea para fomentar el intercambio de experiencias entre las instituciones. Para ello se hace uso del estándar de *e-learning* SCORM [48] como contenedor de laboratorios remotos y JavaScript para iniciar los experimentos. Se han realizado bastantes publicaciones relacionadas con este proyecto [47], [112], [198], alguna incluso utilizando otros estándares como LTI [199].

NetLab (<http://netlab.unisa.edu.au/>) [30]. Es un Proyecto de laboratorios remotos desarrollado y mantenido por la Universidad del Sur de Australia. Es utilizado por el profesorado y los alumnos de la Universidad del Sur de Australia para realizar experimentos en los que controlan remotamente. Existen varios trabajos realizados en el ámbito de este sistema [200], [201]. Existe otra iniciativa que usa la misma nomenclatura, desarrollada en el Instituto de Tecnología de la India que no debe ser confundida con la de Australia, en este caso se trata de un Sistema de Gestión de Laboratorios *online* [202].

OCELOT (*Open and Collaborative Environment for the Leverage of Online instrumentation*, Entorno abierto y colaborativo para el aprovechamiento de la instrumentación en línea, <http://ocelot.ow2.org>). Es un marco de trabajo con el que se puede controlar fácilmente dispositivos *online* utilizando herramientas multimedia interactivas con realidad aumentada. El sistema contempla 4 tipos de laboratorios: presenciales, VL, RL y HL. Surgió con el trabajo de tesis de

Christophe Gravier [203]. En la actualidad está siendo desarrollado por su autor en Télécom Saint-Etienne (Francia) bajo licencia de código abierto [204].

UniLabs, (*University Network of Interactive Labs*, Red universitaria de laboratorios interactivos, <http://unilabs.dia.uned.es/>), [27]. Es una red formada por muchas universidades en la que comparten laboratorios virtuales y remotos con los que se puede interactuar con equipos y dispositivos reales de forma individual o colaborativa. Está mantenido y gestionado por la UNED mediante una plataforma Moodle adaptada. Esta red es la evolución de otra serie de redes y plataformas, la última de las cuales se llamada UNEDLabs [176] [205], que son mantenidas por la UNED, una de las instituciones pioneras y punteras en laboratorios *online* en España. Algunos de los últimos trabajos en los que se ha tratado esta red son [206] y [207].

VISIR (*Integration of Virtual Instrument Systems in Reality*, Integración de Sistema de Instrumentación virtual con realidad, <http://openlabs.bth.se/electronics/>). Se trata de un laboratorio de código abierto dedicado a experimentos remotos de electrónica analógica. Fue lanzado a finales del año 2006 por el Departamento de Procesamiento de la Señal (ASB) del Instituto de Tecnología de Blekinge (BTH) en Suecia junto con las empresas *National Instrument* de EE.UU. y *Axiom EduTECH* de Suecia. Esta infraestructura ha permitido crear trabajos conjuntos con otras redes y arquitecturas mostradas en esta lista. Algunos de los trabajos publicados sobre este sistema son [208], [209] y [210].

WebLab-Deusto, (<https://www.weblab.deusto.es/>), [32]. Es un proyecto de código abierto desarrollado y mantenido por la Universidad de Deusto que se inició en el año 2000. El objetivo del proyecto es mejorar las experiencias de aprendizaje mediante el uso y desarrollo de laboratorios remotos. Para ello ofrecen varios laboratorios que se pueden utilizar a través de Internet de forma gratuita utilizando para ello un *software* que está disponible bajo licencia OSL (*Open Source License*) que puede ser duplicado. Han adoptado el proyecto VISIR adquiriendo una copia de su equipamiento para utilizarlo con los alumnos de la Universidad de Deusto. En el ámbito de este proyecto han surgido numerosas publicaciones [211], [118], [212].

2.2.4. Evaluación de la Efectividad del Aprendizaje

Uno de los primeros trabajos en los que se evaluó la efectividad de los laboratorios *online* fue el realizado por Ogot en 2003 [213] y Corter en el 2004 [7] en donde concluyen que los resultados obtenidos en un laboratorio remoto no son peores que los obtenidos en uno presencial. Wiesner en 2004 [214] realizó un estudio equivalente con laboratorios virtuales en el que concluye que los resultados obtenidos en el laboratorio virtual no son peores que en el presencial. Para ello tuvo en cuenta los resultados obtenidos por los estudiantes en el examen final, en una encuesta de nivel de satisfacción y entrevistas personales. Lindsay y Good publicaron en el 2005 un trabajo en el que compararon la efectividad de un laboratorio remoto, otro virtual y otro presencial llegando a la conclusión de que no eran equivalentes y cada uno de ellos presentaba ciertas debilidades y fortalezas [136].

A pesar de estos trabajos, como ya se ha comentado anteriormente en el epígrafe “2.2.2. Tecnologías de Desarrollo y Comunicación”, hubo unos años en los que los trabajos trataban más el aspecto técnico y de la arquitectura necesaria para el desarrollo tecnológico de estos laboratorios, olvidándose en parte de los aspectos pedagógicos y la efectividad del aprendizaje conseguida en los laboratorios *online*. Algunos trabajos resaltaron este y otros asuntos relacionados [2], [9] (ya había sido señalado mucho antes [215]) como tendencias a seguir y empezaron a aparecer más investigaciones que, además de mostrar laboratorios *online*, ofrecían los resultados obtenidos en los mismos tratando su efectividad. En este sentido han surgido muchos trabajos entre los cuales cabe destacar los siguientes:

En el año 2006 Tzafestas y otros presentaron un trabajo sobre unos experimentos de laboratorio que realizaron los alumnos en 3 modalidades (presencial, remota y virtual) centrándose en el impacto educativo basado en los resultados obtenidos por los alumnos [216]. A pesar de existir ciertas diferencias concluye indicando que los resultados son comparables y que es necesario realizar más trabajos y estudios. Un año más tarde, Nickerson y otros presentaron un modelo para evaluar la efectividad de los laboratorios remotos y virtuales que utilizaron para comparar las versiones presencial y remota de un laboratorio [138]. Obtuvieron datos de los resultados obtenidos por los estudiantes (en el laboratorio y en el curso) y de sus niveles de satisfacción a partir de encuestas llegando a la conclusión de que eran comparables o mejores para la versión remota. Otros trabajos han comparado y combinado los

laboratorios virtuales y remotos [217] concluyendo que los estudiantes obtenían una mejor comprensión que con los laboratorios presenciales. En 2009 Abdulwahed y Nagy realizaron un estudio empírico [218] para medir la efectividad pedagógica de laboratorios *online* y presenciales integrados para poder adaptarse al modelo de enseñanza indicada por Kolb [71] llegando a la conclusión de que se complementan. En otros trabajos realizados por estos autores [219], [23] también han utilizado distintas modalidades de laboratorios con el fin de aprovechar las ventajas de un laboratorio híbrido. En 2013 Stefanovic [220] compara diferentes arquitecturas de laboratorios *online*, muestra los resultados obtenidos y concluye que para medir correctamente la efectividad de un sistema se debe partir del mismo punto y es muy importante la comprobación de que se han cumplido los objetivos y el propósito del mismo. Esta misma idea es compartida por Maiti en la comparativa que realiza en 2013 entre algunas de las arquitecturas de sistemas para laboratorios remotos más importantes del mundo cuando trata de comprobar la efectividad docente de las mismas [186] incluyendo la frase de Arango [221] “*the pedagogical effectiveness of any educational activity is judged by whether or not the intended learning outcomes are achieved*”, que en español podría traducirse como “la eficacia pedagógica de cualquier actividad educativa se juzga en función de si se logran o no los resultados de aprendizaje esperados”.

2.2.5. Revisiones y Tendencias

Una de las primeras revisiones sobre laboratorios *online* se encuentra en el trabajo de Dormido [1] sobre educación en control. Desde entonces se han publicado otras muchas revisiones y comparativas sobre trabajos relacionados con los laboratorios *online*, destacan las siguientes:

- Comparativa de laboratorios presenciales, virtuales y remotos, [9] realizada por Ma y Nickerson en 2006.
- Revisión y evaluación de tendencias en laboratorios remotos [18] realizada por Gomes and Bogosyan en 2009.
- Revisión y comparativa de tecnologías de desarrollo de *software* de cliente y servidor en laboratorios remotos [147] realizada en 2009 por García-Zubía y otros.
- Revisión de laboratorios remotos y virtuales [178] realizada por Chen y otros en 2010.

- Revisión de laboratorios remotos y virtuales [222] realizada por Sancristobal y otros en 2012.
- Revisión de arquitecturas de sistemas para laboratorios remotos [186] que hizo Maiti en el año 2013.
- Revisión de resultados de aprendizaje obtenido por los laboratorios virtuales y remotos frente a los presenciales [141] que hizo Brinson en 2015.
- Revisión laboratorios virtuales de STA (*Science, Technology and Engineering*, Ciencias, Tecnología e Ingeniería) [127] que fue realizado recientemente por Potkonjak y otros y publicada en el año 2016.

Conviene resaltar el análisis bibliométrico [126] realizado en 2016 por Heradio y otros, que partiendo del trabajo de revisión bibliográfica [223] que realizaron Zappatore y otros en 2015 hacen una revisión bibliográfica bastante más amplia sobre la experimentación en laboratorios remotos y virtuales, identifica los trabajos más influyentes, los autores más prolíficos e influyentes, las revistas y congresos más importantes y la evolución que han tenido las diferentes áreas de interés relacionadas con los laboratorios *online*.

2.2.6. Otras Líneas de Trabajo

Otros campos de interés bastante prolíficos en número de trabajos se refieren a la **realidad virtual** y la **realidad aumentada**, dos términos que se pueden confundir con frecuencia. La realidad aumentada pretende perfeccionar una realidad añadiéndole elementos virtuales, mientras que la realidad virtual intenta presentar una realidad mediante dispositivos o técnicas virtuales. Aplicado a los laboratorios *online* la realidad aumentada exige que existan elementos físicos reales, luego se aplica a laboratorios remotos mientras que la realidad virtual se aplica a laboratorios virtuales. La primera referencia sobre realidad virtual y laboratorios la publicaron Vouk y otros en 1999 al describir un proyecto centrado en el desarrollo de laboratorios virtuales para la enseñanza de ingeniería mecánica [224]. Desde entonces se han publicado otros trabajos que han tratado este tema como [225], [226]. El concepto de realidad aumentada es más moderno aunque también existen bastantes trabajos que lo han tratado en el ámbito de los laboratorios *online* [227], [121], [122].

Otro tema considerado en las investigaciones relacionadas con los laboratorios *online* ha sido la posibilidad de **cooperación** entre estudiantes. En las prácticas

presenciales es común que existe una realimentación y apoyo entre los estudiantes entre sí que incluso en muchas ocasiones trabajan de forma conjunta en la consecución de los trabajos de prácticas. La idea de evitar el aislamiento de los alumnos en la educación a distancia a través de Internet no es nueva, la primera referencia encontrada sobre laboratorios virtuales y cooperación se publicó en 1999 [228]. Desde entonces, la posibilidad de permitir una cooperación entre estudiantes a la hora de trabajar en los laboratorios *online* ha sido tratada e implementada en bastantes trabajos [70], [174], [229], [230], [231], [180], [232], [233]. Existen propuestas de laboratorios que han considerado la posibilidad de proporcionar aprendizaje cooperativo de forma complementaria aplicando diferentes soluciones como son la de desarrollar laboratorios que permitan ser utilizados por varios estudiantes al mismo tiempo [234], [235], integrando los laboratorios en mundos virtuales que proporcionan mecanismos de cooperación [236], [237], [229] o integrando los laboratorios *online* en LMSs [117], [238].

A partir del año 2010, la **Integración de laboratorios *online* y LMSs** ha sido otro de los campos de trabajo más importantes como se señala en el trabajo de Heradio y otros [126]. En el apartado siguiente se tratará con más detalle por estar directamente relacionado con este trabajo de tesis.

Finalmente, existen numerosas tesis doctorales relacionadas con los laboratorios *online*. A continuación se referencian algunas de las presentadas en el entorno de la UNED:

- En [239] se presenta un nuevo enfoque metodológico para la enseñanza basado en un laboratorio virtual y remoto para el estudio de la automática.
- En [240] se hace una especificación formal de laboratorios virtuales y remotos aplicada a la ingeniería de control.
- En [241] se presenta un modelado Orientado a Objetos de Laboratorios Virtuales para la Educación en Control Automático.
- En [242] se muestra un entorno integral basado en web para la educación en ingeniería de control.
- En [243] se introduce el concepto de LaaS (*Laboratory as a Service*, Laboratorio como servicio) un paradigma para desarrollar e implementar laboratorios remotos modulares.

Y otras más directamente relacionadas con este trabajo al tratar la integración de laboratorios *online* y sistemas educativos y LMS:

- En [244] se presenta una metodología, estructura y desarrollo de interfaces intermedias para la conexión de laboratorios remotos y virtuales a plataformas educativas.
- En [245] se muestran unas plataformas interactivas de experimentación virtual y remota con aplicaciones de control y robótica.
- En [42] se presenta el desarrollo de una nueva generación de laboratorios *online* que se integran en LMSs gracias al desarrollo de un módulo propio para el LMS Moodle.

2.3. Integración de Laboratorios *online* y LMSs

En esta sección se van a mostrar trabajos relacionados con la integración de los laboratorios *online* y los LMSs. En primer lugar se va a tratar la evolución que ha sufrido esta integración y se va a hacer un recorrido temporal mostrando los trabajos más relevantes. Posteriormente, se mostrarán los trabajos de integración en los que los estándares SCORM han tenido un papel destacado.

2.3.1. Evolución de Integración de Laboratorios *Online* y LMSs

Las primeras relaciones entre plataformas de formación y laboratorios *online* se produjeron cuando se trató de ofrecer diferentes laboratorios *online* en un mismo entorno dedicado. De este modo surgieron algunas plataformas de laboratorios *online* independientes que ofrecían a los alumnos la posibilidad de utilizar diferentes laboratorios en un mismo entorno (por ejemplo AutomatL@bs [246]). Estas plataformas fueron evolucionando para integrarse con los LMS institucionales de las organizaciones docentes tratando de resolver los problemas que surgían, un ejemplo de estos intentos lo constituye el trabajo de Ponta [194] que examina las posibilidades de ofrecer experimentos remotos a usuarios finales como servicios distribuidos utilizando la tecnología de servicios Web. Este trabajo analizó las arquitecturas de 3 de los sistemas de laboratorios remotos más conocidos, como son ISILab del DIBE [29], HPI DCL [31] y el iLab del MIT [24] llegando a la conclusión de que uno de los primeros pasos en realizar debe ser utilizar mecanismos de descubrimiento para poder obtener información de los experimentos disponibles.

El mejor ejemplo de esta evolución se tiene en los sistemas y proyectos que ha presentado el Departamento de Informática y Automática (DIA) de la UNED, uno de los organismos más avanzados a nivel mundial en laboratorios *online* que ha liderado a un grupo de universidades en España para compartir laboratorios *online* [247]:

- 2004. Portal remoto del DIA.
- 2006. Proyecto AutomatL@bs para experimentos en ingeniería de control [248].
- 2006. Entorno de e-Mersion [37].
- 2009. Proyecto FisL@bs para experimentos en física [249].
- 2011. Entorno UNEDLabs [205], [176].
- 2014. UNILabs [27], [206].

La integración WebLab-LMS es un tema recurrente que aparece en múltiples trabajos realizados sobre laboratorios docentes *online* desde que fue señalado como tal en el trabajo de **Gravier y otros** en 2006 [41]. En dicho trabajo señalaba la falta de disponibilidad de laboratorios *online* genéricos y colaborativos, prácticamente la totalidad de los laboratorios existentes eran específicos del dispositivo físico con el que se experimenta y no presentaban colaboración síncrona. Esto exige reinventar sistemas cada vez que se quiere implementar otro laboratorio. Por otro lado indicaba la necesidad de integrar los LMS y los RL como objetos de aprendizaje definidos y tratados en los LMS junto al resto de recursos de aprendizaje y nunca aislados.

Rapuno y Zoino [250] presentaron en 2006 una propuesta de integración de laboratorios remotos desarrollados con LabVIEW y un LMS basado en un módulo específico o *plugin* desarrollado en el LMS para encapsular el laboratorio remoto. El nuevo módulo permite disfrutar de cuatro tipos de experiencias a los estudiantes: 1) laboratorio virtual síncrono, 2) visualización de un experimento, 3) control de experimento y 4) creación de experimento.

En 2006, **Donzellini y Ponta** [69] exploraron los aspectos del uso pedagógico de laboratorios virtuales y remotos en el campo de la ingeniería electrónica, utilizando como ejemplos dos proyectos. En el segundo de ellos, ISILab, se presenta al RL como un elemento no aislado que forma parte de una estrategia de tres fases a) Pre-Lab, b) Lab y c) Post-Lab. Para ello realizó lo que denominó una integración Lab-LMS. Utilizó el LMS Moodle para crear una secuencia de aprendizaje en la que se ofrecen los recursos Pre-Lab; fundamentos teóricos y

un test que debían superar los alumnos para permitirles el acceso a un experimento. Tras realizar los experimentos se usa Moodle de nuevo para recoger los informes que los alumnos realizaron sobre el experimento completado y se recogió su opinión sobre la efectividad del laboratorio mediante un módulo personalizado también en el LMS.

Andria y otros, en 2007 [251], describen la integración de un LMS con un laboratorio remoto de medidas eléctricas y electrónicas basado en *web services*. El *software* de cliente tiene dos versiones en ActiveX y en *applet* de Java y está ubicado en un LMS específico al que se ha creado un subsistema para visualizar los experimentos y conectarse con un servidor de laboratorios que actúa como Servicio puente con el servidor de medidas que está desarrollado con LabVIEW.

En 2008, **San Cristóbal y otros** [252] describen una solución para integrar laboratorios virtuales en LMS mediante el desarrollo de un middleware con el que los laboratorios virtuales puedan consumir los servicios que ofrece el LMS. Constituye una intención de integración basada en el desarrollo de un *plugin* específico de dotLRN (.LRN), el LMS elegido para implementarlo.

Lowe y otros [253], en el año 2009, publican un artículo en el que analizan la evolución de las arquitecturas de laboratorio remoto, sus problemas y las posibilidades de superarlas basadas en las tecnologías emergentes de Internet. Para ello primero analizan algunas arquitecturas como son iLab y la utilizada en su universidad (UTS, Universidad Tecnológica de Sidney), llegando a la conclusión de que algunos problemas pueden resolverse utilizando las últimas tecnologías que estaban surgiendo en Internet. En relación a la integración de laboratorios y LMS añaden que las arquitecturas analizadas no presentan un buen soporte para su integración con los LMS, a pesar de presentar un gran número de funcionalidades capaces de enriquecer enormemente a los laboratorios remotos.

En 2009, **Cmuk y otros** presentaron MIRACLE [254], un modelo de educación de tipo *b-learning* para ingeniería eléctrica en el que presentaron una serie de 14 guías de implementación como recomendaciones para que el profesor actúe ante la implementación de un RL. En las guías se recomendó la presentación de una serie de recursos del LMS con los que los alumnos deben trabajar o superar antes de acceder al laboratorio y tras trabajar en el mismo, por ejemplo, leer textos de teoría, *tests*, informes en wikis, foros, y otros elementos del LMS.

En el año 2010, **Abdellaoui y otros** [255] presentaron un estudio sobre la interoperabilidad de LMSs y laboratorios remotos en una integración sin fisuras. Se enfatizó la necesidad de ver a los laboratorios remotos como materiales pedagógicos en el interior de los LMS y presentaron una propuesta para lograr la comunicación entre ambos mediante una relación de acoplamiento débil o ligero. La justificación de que este acoplamiento debía ser débil se debe a la dificultad de que un LMS pudiera alojar todos los trabajos prácticos en un LMS. Propuso el desarrollo de *plugins* en los LMS para conectar RL y LMS.

Bochicchio y Longo [117] en el año 2010 publicaron un trabajo en el que señalaron la falta de integración entre los LMS y los laboratorios remotos y la falta de interacción síncrona entre los participantes en el experimento (profesores y estudiantes). Presentaron los principales aspectos y resultados que obtuvieron en un proyecto de integración entre el LMS Moodle y el laboratorio remoto MicroNET de la Universidad de Salento. Para lograrlo analizaron el *software* del laboratorio y sus componentes y trataron de lograr la integración de cada uno de ellos con Moodle paso a paso. Para ello fue necesario el desarrollo de *plugins* de Moodle. Un año más tarde, en 2012, los mismos autores presentaron un trabajo [256] describiendo una arquitectura de sistema basada en requerimientos técnicos y pedagógicos y analizaron los resultados obtenidos en la implementación de un prototipo usado en un escenario académico real.

Pesquera y otros [257], en el año 2011, mostraron el diseño y desarrollo de una arquitectura para conectar el LMS dotLRN con WebLabs de distintas universidades. En dotLRN crearon una nueva estructura (comunidad) llamada *Laboratory* y tipo de grupo al que le dieron características, roles y permisos para dar soporte a los laboratorios y desarrollaron un nuevo modelo que encapsularon en un paquete que contenía el modelo de datos del laboratorio y un API. Definieron distintos tipos de acceso para permitir adaptarse a diferentes laboratorios remotos soportando varios estándares.

En el año 2011, **Vargas y otros** [248] presentaron el proyecto AutomatL@abs desarrollado en el entorno de la UNED. Se trataba de un proyecto innovador en el campo de la experimentación remota aplicada a la educación en Ingeniería de Control en el que participan 7 universidades españolas para compartir sus laboratorios remotos. Los autores presentaron varios ejemplos y mostraron la integración con un LMS dedicado, eMersion, que gestionaba el sistema de reservas y el acceso de usuarios a los experimentos además de presentar

contenidos y herramientas de comunicación. La comunicación del *software* que utilizaban los estudiantes, EJS, con los servidores de laboratorio, que utilizaban LabVIEW, se realizó mediante el uso del middleware JIL. Para finalizar presentaron y analizaron la evaluación del sistema basado en la percepción de los estudiantes sobre la calidad del entorno experimental, los cuales mostraron un alto nivel de satisfacción.

En el año 2012, **de la Torre y otros** [53] presentaron un trabajo en el que mostraron la integración de laboratorios *online* EJS en un LMS de tipo Moodle [176]. Para ello presentaron el portal de experimentación UNEDLabs, en el que los estudiantes podían encontrar VRLs en los que trabajar realizando experimentos virtuales y remotos y los profesores tenían una herramienta como es el *plugin* VRLab de Moodle y un procedimiento con los que podían integrar los laboratorios desarrollados con EJS como *applets* Java en el LMS fácilmente. El portal permitía además la creación de todo tipo de recursos de *e-learning* soportados por el LMS Moodle para que acompañaran a los experimentos: documentación, herramientas colaborativas y sociales, repositorio de ficheros privado, un calendario que señala las fechas de entrega de las actividades, vídeos, etc. Este trabajo fue ampliado el mismo año en la tesis doctoral del autor principal, en la que se mostraron con más detalle todos los desarrollos y múltiples ejemplos de uso [42]. Siguiendo esta línea de trabajo, **Guinaldo y otros** [258] publicaron en el año 2013 un laboratorio virtual y remoto basado en web para el control de un sistema de bola y barra. Se utilizó el *software* EJS para crear los *applets* Java del VL y el RL que debía utilizar el alumno. Estos laboratorios se integraron en la plataforma UNEDLabs basada en el LMS Moodle utilizando el *plugin* dedicado EJSApp. Este módulo permitía insertar *applet* Java desde el propio dispositivo u otras localizaciones *online*, controlar el acceso de los usuarios al laboratorio, facilidades de mantenimiento y empaquetado en los back up de Moodle y estadísticas. **Chaos y otros** [152] publicaron, también en el año 2013, un trabajo sobre un laboratorio de robótica remoto y virtual utilizando EJS, Matlab y LabVIEW. Fue integrado en UNEDLabs utilizando el mismo sistema descrito en los trabajos comentados anteriormente. Se señalaba la posibilidad que ofrecía el sistema Moodle de presentar junto al VRL otros recursos como guiones de prácticas, guías de usuario, documentos técnicos y usar el sistema de reservas que implementa la plataforma.

También en el año 2012, **Orduña y otros** [259] presentan un trabajo en el que estudiaron el uso de modelos de federaciones para integrar RLMS (Sistemas de

gestión de laboratorios remotos) en LMSs y muestra 2 casos de estudio con los LMS Joomla y Moodle.

Tawfik y otros, en el año 2012, realizaron una publicación incluyendo un informe que mostraba soluciones emergentes para la implementación y despliegue eficiente de laboratorios remotos de formación en ingeniería. También analizó diferentes escenarios de integración incluyendo características, limitaciones y desafíos futuros.

En el año 2012, **Fernández y otros** [168] describieron una plataforma desarrollada para integrar el acceso a diferentes laboratorios virtuales y remotos. También desarrollaron un *plugin* para permitir el acceso a la plataforma, y por tanto a los laboratorios, desde Atenea, el LMS que estaba implementado en una plataforma Moodle. El sistema fue probado y evaluado mediante una serie de encuestas que demostraron la eficiencia del mismo.

Rodríguez-Gil y otros [260] publicaron en el año 2012 un ejemplo de integración avanzada de dos sistemas de laboratorios, VISIR y WebLab-Deusto. Aunque VISIR podía funcionar de forma independiente, ya que es capaz de realizar la autenticación de usuarios, gestión de usuarios y planificación entre otras funciones, se pensó en ofrecer parte de sus experimentos en el marco de un RLMS. Este trabajo describió los esfuerzos realizados para lograr la integración VISIR/WebLab-Deusto aportando a VISIR la posibilidad de integrarse con redes sociales compatibles y LMSs como Moodle.

En el año 2012, **Lerro y otros** [43] presentaron un proyecto de integración de un laboratorio remoto de Física electrónica y un LMS (e-educativa) para un curso experimental, llevado a cabo por una empresa tecnológica y la Universidad Nacional de Rosario en Argentina. Básicamente, consistía en el desarrollo de un interface basado en un protocolo de comunicación estandarizado que permitía el intercambio de datos entre el servidor de laboratorio y el LMS. Más tarde, en el año 2013, **Lerro y otros** [261] presentaron un sistema de autorización y autenticación centralizado que podía ser utilizado para compartir laboratorios. El sistema permitía utilizar sistemas de identificación centralizados como los usados por las universidades (Shibboleth o OpenID) y las redes sociales (Facebook Connect o OAuth de Google) o proporcionar uno propio para centros que no cuenten con uno. De este modo los usuarios identificados podían consumir los servicios de los RLMS y acceder a sus laboratorios. En el año 2014, presentaron otro trabajo [262] sobre el mismo tema en el que se hacía un

diseño mejorado del *interface* del laboratorio remoto basado en CSS y HTML5 para adaptarse al LMS e-educativa. La integración se producía en forma de módulos del LMS como actividades que debe crear un profesor. El sistema era capaz de generar un informe del resultado de la evaluación que se realizaba.

Kafadarova y otros [263], en el año 2013, publicaron un trabajo dentro del proyecto RIPLECS en el que se describía un laboratorio remoto de comunicaciones inalámbricas. En este laboratorio los estudiantes podían acceder para trabajar remotamente con una serie de antenas accediendo desde LMS sin requerir *software* adicional.

En el año 2015, **Wuttke y otros** [264] describieron un nuevo enfoque general para integrar los laboratorios *online* en el proceso educativo con el objetivo de evaluar los conocimientos de los estudiantes, no sólo en los niveles más bajos, como se hace con preguntas de múltiple elección y similares, sino las habilidades de pensamiento de nivel superior. Para ello utilizaron principalmente un LMS Moodle y un laboratorio remoto que eran compatibles con xAPI para poder almacenar registros en un LRS (*Learning Record Store*), además utilizaron una serie de herramientas de evaluación que soportaban xAPI. El LMS incluía contenidos de aprendizaje, enlaces relevantes e interconexión con los programas externos mediante componentes. La integración LMS-Lab consistía en ofrecer un único portal de entrada y registro de usuarios (Moodle) desde el que se llamaba al resto de sistemas.

Colbran y Schulz [185], en el año 2015, mostraron una actualización de la arquitectura ISA utilizada por el *Service broker* de iLab para que, además de soportar la API tradicional, pudiera utilizar otra API basada en una arquitectura RESTful y usar JSON para codificar los datos. De este modo se utilizaba un nuevo servicio *software* llamado agente que ofrece una capa envolvente al *Service broker* y permite realizar modificaciones programables de peticiones. Para demostrar su uso se realizó un ejemplo de integración de un laboratorio remoto con un LMS Blackboard utilizando el protocolo LTI para la autenticación de usuario.

También, en el año 2015, el trabajo de **Ozvoidova y Ondrušek** [265] trató la integración de los laboratorios *online* en sistemas de educación mostrando algunos ejemplos y analizando como se puede lograr a una escala global. En el trabajo se dijo que, desde el punto de vista del alumno, la integración sería de alta calidad si es imperceptible, es decir, si no observaba diferencias de

tecnología, cambios estéticos, ni necesitaba realizar registros adicionales. Además de la integración visual también distinguió otro factor determinante en la calidad de la misma, referido a la capacidad de interaccionar que pueda tener el laboratorio con el LMS.

En 2016, **Sáenz y otros** [206] presentaron un curso de libre acceso en UNILabs que ofrecía numerosos laboratorios remotos y virtuales sobre control automático. Detallaban estos laboratorios y las actividades que se podían hacer. Los laboratorios fueron desarrollados utilizando una solución de bajo coste, utilizaron el *software* EJS (evolución de EJS que permite crear aplicaciones en JavaScript además de Java) para crear los interfaces de usuario y una placa de plataforma de desarrollo económica, llamada BeagleBone Black. Los VRL se presentaron en un LMS Moodle (UNILabs) de acceso libre a través de la web que facilitaba su gestión y mantenimiento. La comunicación entre la aplicación EJS que ejecutaba el estudiante y la placa BeagleBone se realizaba gracias a un *middleware* desarrollado por los autores. Añadían que al encontrarse en un LMS los laboratorios se ven enriquecidos, facilitando y promoviendo las herramientas para que los estudiantes intercambiaran ideas con otros compañeros y los profesores sobre los experimentos, compartieran los informes de sus laboratorios y trabajaran en sesiones colaborativas. Además se podían usar todos los tipos de recursos que ofrece un LMS como Moodle para cualquier actividad relacionada con los laboratorios.

2.3.2. Integración Lab-LMS y SCORM

SCORM es uno de los estándares de *e-learning* más extendido y utilizado en LMS, el número de sistemas que soportan SCORM es muy alto [266]. El estándar SCORM se utiliza para definir 1) contenido de *e-learning*, 2) la secuenciación y navegación de sus componentes, y 3) el lanzamiento y comunicación de los contenidos con el sistema que los hospeda. Los laboratorios *online* se pueden considerar un tipo de contenido de *e-learning*, probablemente el más complejo y avanzado. Por eso no es extraño que se haya querido desarrollar laboratorios *online* utilizando SCORM. Existen muchos trabajos en los que se ha relacionado a SCORM con los laboratorios *online*, aquí se presenta una muestra de los más representativos:

En 2007, **Özdogru y Cagiltay** [267] mostraron el diseño e implementación del proceso de integración de un LMS Moodle con una plataforma de laboratorios

remotos. Propusieron los estándares SCORM para crear los cursos y contenidos almacenados en un LMS tipo Moodle y los interfaces necesarios para integrarse con los laboratorios aunque no se concretó ninguna solución real.

Gomes y Bogosyan [18], en el año 2009, presentaron un estado del arte de las tecnologías utilizadas en los laboratorios, presentaron algunos ejemplos de laboratorios remotos de educación en electrónica industrial y apuntaron algunas tendencias y retos. Entre otras cosas, señalaban la necesidad de describir los laboratorios remotos en términos de sus metadatos mediante el uso de estándares como SCORM. También indicaron que una de las características de los laboratorios remotos es que ofrecían soluciones particulares, con una capacidad limitada o inexistente de cooperar con otras plataformas al haber sido desarrolladas como soluciones especiales o ad-hoc. Señalaban como reto la estandarización, modularidad, portabilidad, escalabilidad e interoperabilidad entre las diferentes soluciones. En este sentido, presentaban a los web services, sistemas de metadatos como SCORM y arquitecturas abiertas como aspectos clave para evitar estas limitaciones.

En el año 2010, **Sancristobal** y otros describieron en [44] una técnica para presentar en un navegador un WebLab incrustado como parte de un paquete SCORM que es entregado por un LMS, y una arquitectura orientada a servicios para integrar LMSs con iLab y otros laboratorios remotos. Al incrustar el laboratorio en un SCORM que estaba en un LMS, era el LMS el encargado de controlar la autenticación, además, si el Lab podía invocar la API SCORM podría comunicarse con el LMS y hacer que el profesor pudiera ver el progreso del usuario. Se indicaba la posibilidad que tienen los laboratorios, una vez incrustados en el paquete SCORM, de intercambiar datos con el LMS usando las funciones JavaScript de la API que suministraba el LMS. Sin embargo, si el lenguaje elegido para crear el laboratorio no permitía realizar llamadas a código, la comunicación no era posible salvo que se integrara el laboratorio en un *iframe* de la página Web y fuera éste el que realizara el intercambio de datos con el LMS. En este último caso no se podía considerar que sea el *software* del laboratorio quien se comunicaba con el LMS, sino su entorno web.

En el año 2011, **Richter** y otros [47] publicaron un trabajo en el que presentaron el Proyecto LiLa (*Library of Labs*) [26], describiendo sus objetivos, propósitos, diseño *software* y algunas de las primeras experiencias. LiLa también funcionaba como un repositorio de laboratorios incrustados en paquetes SCORM que se podían descargar para ser utilizados en LMSs. El mismo año, publicaron un

trabajo [268] en el que se mostraba la integración de laboratorios virtuales y experimentos remotos de LiLa. Fue en un curso de Física para ingenieros de la Universidad de Stuttgart en el que presentaban los resultados obtenidos en una fase piloto. Esta integración se produjo en un LMS ILIAS en el que se usaron otros recursos propios de la plataforma además de los paquetes SCORM de LiLa. De este modo, para cada experimento, establecieron un flujo de trabajo que los alumnos debían cumplir: una fase de orientación (resumen del experimento, descripción del experimento y tarea a realizar y un pre-test), una fase de ejecución (experimento *online*) y fase de revisión (test). En el año 2014, publicaron otro trabajo [199] en el que muestran las limitaciones que presentaba SCORM para implementar un laboratorio de programación *online* basado en VipLab, una herramienta de programación basada en web. Para solucionarlo, desarrollaron una solución específica utilizada en la Universidad de Stuttgart, consistente en un *plugin* de VipLab para el LMS ILIAS, que permite insertar un *applet* de Java y código JavaScript para insertar VipLab como un nuevo tipo de pregunta. Consideraron LTI como una posible solución a la integración en el futuro.

En el año 2012, **Bellido y otros** [269] presentaron un trabajo de acceso remoto a un laboratorio de redes de computación basado en el sistema LiLa. Al acceder al LMS utilizando un protocolo federado como Shibboleth se conseguían los permisos necesarios para poder acceder al laboratorio remoto, el estudiante sólo tenía que acceder a un módulo SCORM en el que estaba incrustado el *software* del laboratorio remoto en forma de LLO (*Lab Learning Object*). El navegador del estudiante descargaba el LLO y contactaba con el sistema de reservas para comprobar los permisos de acceso que, en caso de ser correctos (tener una reserva), le permitían el acceso al laboratorio remoto. Existían 5 tipos de LLO en función del escenario que se presentaba a los alumnos, en todos ellos lanzaba un VNC (*Virtual Network Computer*, Ordenador Virtual en Red) a través de un cliente ssh por el puerto que se le había indicado.

Barra y otros [270], en 2013, mostraron un trabajo en el que usaban paquetes SCORM para integrarlos con juegos e interactuar con los mismos ofreciendo contenidos educativos.

En el año 2013, **Orduña y otros** [271] describieron la iniciativa Go-Lab [272], llamada gateway4labs [273], que en los años 2014 [274] y 2015 [275], fue actualizada. Esta iniciativa estaba dedicada a extender la integración de RLMS (Sistema de gestión de laboratorios remotos) o múltiples laboratorios remotos

con múltiples LMSs. Para ello la arquitectura (**Figura 35**) contemplaba la posibilidad de usar los estándares LTI (*Learning Tool Interoperability*), Open Social o soluciones dedicadas basadas en HTTP y *plugins* en los LMS para conectarse a un gestor de laboratorios. Este gestor actuaba como elemento central de una arquitectura que permitía acceder a los laboratorios remotos mediante *web services* o el protocolo necesario para conectarse con el RLMS que los gestionaban. Cuando los LRMS requerían identificación debían soportar protocolos de federación que permitieran usar en el RLMS la identificación que el usuario introduce en su LMS. Cuando el LMS no era compatible con las tecnologías LTI o Open Social debía implementarse un acceso mediante SCORM en el LMS que debía usar un *plugin* específico para poder comunicarse con el *Lab manager*. El problema que planteaban estas soluciones es que el contenido SCORM en el que está incrustado el laboratorio no se comunicaba utilizando la API original de SCORM, sino una API implementada por el *plugin* de servidor del LMS que constituye una solución particular y no reutilizable salvo en un LMS del mismo tipo que tenga el mismo *plugin*.

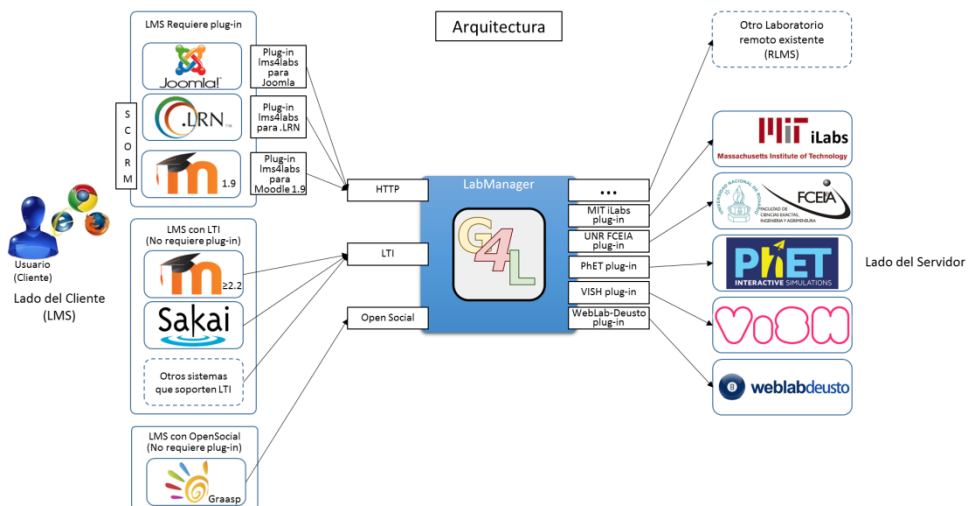


Figura 35. Arquitectura del Sistema Gateway4Labs

Cap.3

3. Integración Lab-LMS

- 3.1. Objetivos de la Integración Lab-LMS
 - 3.2. Propuesta de Modos de Integración Lab-LMS
 - 3.3. Paradigma de Integración Lab-LMS Mediante SCORM
-

Integrar es una palabra que tiene su origen en el latín *integrare* ‘renovar’ o ‘completar’. El Diccionario de la Real Academia Española de la lengua (DRAE) [276] muestra varias acepciones, pero la más correcta para el caso que nos ocupa es la que indica “Constituir un todo” o “Hacer que algo pase a formar parte de un todo”. Aplicado a un laboratorio *online* y un LMS se entiende que se trata de hacer que el laboratorio *online* pase a formar parte del LMS. Sin embargo, este no es un concepto claro, podría entenderse que se consigue la integración Laboratorio-LMS simplemente con alojar un laboratorio *online* en un LMS o con proporcionar un vínculo que los relacione en un conjunto. Además, dependiendo del punto de vista con que se mire, **para un alumno** se puede

considerar que una integración de calidad se puede conseguir cuando no perciba la diferencia entre el LMS y el laboratorio [265], se podría añadir que también valoran cuando pueden interactuar con compañeros y tutores para comunicarse y comprobar el resultado del trabajo que realizan en el laboratorio *online* lo más rápidamente posible. Por otro lado, para un **desarrollador de laboratorios *online***, la integración se puede entender desde un punto de vista técnico y de las tecnologías utilizadas. Cuando se trata de laboratorios remotos, o simulados gestionados remotamente por algún repositorio externo al LMS, la integración de las comunicaciones con el LMS puede llegar a ser bastante más compleja. Esto es debido, entre otras cosas, a la variedad de comunicaciones que se pueden llegar a producir con el LMS (**Figura 36**).

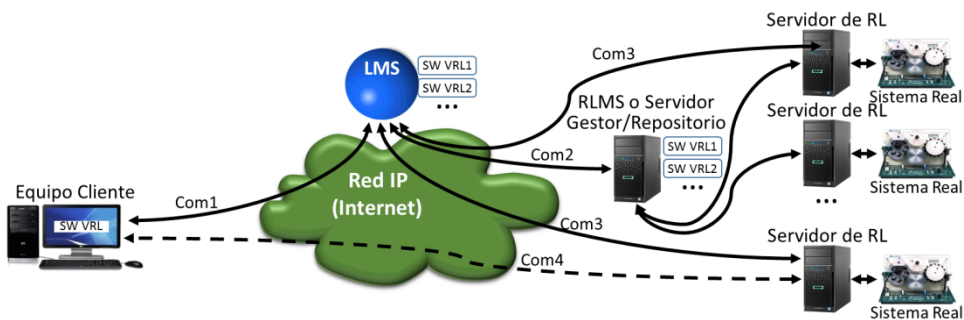


Figura 36. Comunicaciones LMS-Laboratorio

Las comunicaciones tipo 'Com1' se producen entre el *software* que presenta la *interface* de usuario con el que el estudiante interactúa con el sistema real y el LMS, las tipo 'Com 2' entre el LMS y los sistemas de gestión de laboratorios *online* y las tipo 'Com3' entre el LMS y los servidores de RL, las cuales no siempre se producen cuando se cuenta con un sistema intermedio. Se han dejado al margen las comunicaciones tipo 'Com4', que se producen entre el *software* que presenta la *interface* de usuario con la que el estudiante interactúa con el sistema real y el servidor de RL, las cuales no necesitan establecerse a través del LMS.

En la sección "2.3. Integración de Laboratorios *online* y LMS", muchos de los trabajos referenciados que trataban la integración laboratorio-LMS se centraban sólo en explicar las comunicaciones tipo 'Com2' y 'Com3' referidas a una determinada arquitectura, repositorio o sistema RLMS.

El tercer actor más importante en la docencia de laboratorios *online* es el **tutor o profesor**. Para ellos un buen nivel de integración con el LMS se logra cuando

se puede insertar fácilmente un laboratorio *online* en el espacio virtual de su docencia, puede relacionarlo con otros recursos que considere oportunos y comprobar fácilmente el trabajo realizado por sus alumnos.

En cualquier caso, lo que sí es evidente es que se pueden establecer diferentes grados, niveles o modos de integración. Antes de analizarlos es necesario examinar las ventajas que se pueden obtener de una integración. Esto es lo que se hace en el apartado siguiente. Posteriormente, se ha realizado un análisis y clasificación de las posibilidades de integración Laboratorio-LMS que se ha incluido en el apartado 3.2. de este trabajo. En este análisis el nivel de integración más alto se consigue cuando se logra una integración avanzada, entendiéndose como tal a la posibilidad de que el laboratorio pueda interactuar con el LMS estableciendo comunicaciones bidireccionales. Para terminar este capítulo, el apartado 3.3 explica cómo se puede lograr una integración avanzada WebLab-LMS utilizando los estándares SCORM. Esta es la propuesta por la que se ha optado en este trabajo. Se trata de una opción que, entre otras características, permite integrar plenamente el laboratorio en el LMS, con un aprendizaje centrado en el alumno, y que los alumnos y tutores puedan comprobar el seguimiento del trabajo realizado en el mismo LMS.

3.1. Objetivos de la Integración Lab-LMS

La integración tratada en este trabajo de tesis no se refiere a las plataformas específicas de laboratorios *online*, sino a la que se produce entre el *software* de laboratorios *online* donde los estudiantes interactúan para ejecutar sus experimentos prácticos y los LMS generalistas, normalmente los institucionales de las organizaciones responsables de la docencia. Para entender mejor el motivo que ha llevado a trabajar en esta integración hay que comprender los beneficios que se derivan de la misma ya que son los objetivos que se marcan en este trabajo. Por ejemplo, una de las principales ventajas que pueden obtenerse cuando se produce una integración WebLab-LMS, aunque no la única, es la posibilidad de establecer rutas de aprendizaje que consideren contenidos incluidos en un LMS y un laboratorio *online*. A continuación se enumeran las principales ventajas que pueden obtenerse cuando un laboratorio *online* es integrado en un LMS. Hay que destacar que no todas estas ventajas se presentan en todos los modos de integración que existen. La integración propuesta en este trabajo de tesis tiene todos estos objetivos asignados:

- **Ejecución en entorno conocido.** Los LMS constituyen el elemento tecnológico principal de la docencia universitaria, los alumnos están acostumbrados a su uso diario y cualquier recurso, como un laboratorio *online*, que se les presente en el LMS o a través del mismo supone un cambio menor de lo que podría suponer si se presentara en cualquier otro entorno.
- **Identificación de alumnos.** Los LMS realizan una identificación de usuarios, normalmente basado en el sistema usuario-contraseña, que proporciona seguridad en diversos ámbitos. Cuando se presenta un laboratorio *online* a través de un LMS la identificación es realizada por el LMS, liberando al laboratorio *online* de esta función, los usuarios del laboratorio *online* podrán utilizarlo sin tener que introducir sus credenciales. No es necesario que el laboratorio *online* se encuentre ubicado en el interior LMS, en el caso de que un laboratorio *online* se encuentra en una ubicación externa al LMS, puede recoger las credenciales del usuario, establecer una comunicación con el LMS para enviársela y que sea éste quien realice la identificación. Sin embargo este caso requiere implementar cifrado en el intercambio de información laboratorio-LMS para lograr un nivel de seguridad parecido.

- **Formación adaptada al alumno.** Si un laboratorio *online* es capaz de comunicarse con el LMS para obtener la identificación del alumno tendrá la posibilidad de adaptarse al mismo. En este caso podrá ofrecer diferentes experimentos, o variaciones de los mismos, basados en la identificación del alumno o el perfil docente del alumno obtenido de ejecuciones pasadas del alumno en el mismo laboratorio.
- **Ejecución Personalizada/Accesible.** De igual forma que la descrita en el punto anterior, si el laboratorio puede comunicarse con el LMS y almacenar las preferencias del alumno en asuntos como son el idioma, subtítulos, volumen, velocidad de ejecución o cualquier otra característica, entonces se podrá presentar una ejecución del laboratorio acorde a los gustos y/o necesidades del alumno en la misma ejecución y/o posteriores.
- **Presentación con otros elementos del LMS.** Numerosos estudios han tenido en cuenta la presentación de los laboratorios junto a otros recursos y han promovido el hecho de que el laboratorio no debe ser un elemento aislado en el aprendizaje del alumno, sino que debe presentarse junto a otros recursos [69], [68], [179]. En un LMS se pueden utilizar muchos tipos de recursos para implementar los objetos de aprendizaje que pueden ser útiles para la ejecución de un laboratorio. De este modo, junto al laboratorio *online*, se pueden presentar los siguientes elementos:
 - *Recursos de contenido.* Ficheros en formato propietario de la plataforma o algún estándar como PDF o SCORM. Para incluir explicaciones de requerimientos técnicos o formativos para el acceso a un laboratorio *online*, explicaciones de las características del laboratorio *online*, objetivos buscados, estructura del mismo, principio de funcionamiento, manuales de instrucciones del Laboratorio que expliquen el funcionamiento del *software* del laboratorio, guiones de prácticas que expliquen a los alumnos en qué consisten las prácticas y el trabajo que deben realizar, teoría específica del trabajo de laboratorio o soluciones a los experimentos propuestos.
 - *Herramientas de evaluación.* Tests o pruebas de evaluación para comprobar el nivel de conocimiento antes de acceder a un laboratorio o tras su realización para comprobar el nivel adquirido.

- *Herramientas de comunicación.* Permiten establecer comunicaciones alumno-profesor para realizar tutorías *online* sobre el laboratorio o alumno-alumno para permitir su coordinación con el objetivo de ejecutar el laboratorio *online* de forma conjunta.
 - *Asíncronos.* Son los Foros, Correo electrónico y otros medios de comunicación que permiten establecer comunicaciones sin ser necesario la coincidencia temporal de los intervinientes en la comunicación.
 - *Síncronos.* Chats, herramientas de video-conferencias, VoIP y otros medios de comunicación en el que los actores intervinientes deben coincidir en el tiempo.
- *Videos, Podcasts y otros recursos multimedia.* Pueden utilizarse para las mismas funciones que las explicadas en los recursos de contenido.
- *Encuestas y herramientas de realimentación.* Permiten obtener la opinión de los alumnos antes, durante o después de ejecutar la propuesta de aprendizaje.
- *Herramientas de edición de contenidos.* Permiten que los estudiantes aporten información sobre el trabajo realizado, dependiendo de la plataforma pueden utilizarse distintos formatos tipo texto, gráfico, wiki, glosario, etc.
- **Establecimiento de rutas, caminos o recorridos de aprendizaje.** Los diferentes recursos que se pueden asociar con un laboratorio *online*, enumerados en el punto anterior, se pueden presentar sin establecer relaciones entre ellos o, por el contrario, se puede formar con ellos un plan de aprendizaje o recorrido de aprendizaje estableciendo una secuencia para trabajar con los mismos. Estas secuencias de aprendizaje pueden ser autorreguladas por el propio alumno [277] sin establecer ninguna restricción o se puede hacer uso de las facilidades que suelen proporcionar los LMS para controlar la secuencia de acceso a los recursos. En este caso es necesario configurar el control de acceso de cada recurso en la secuencia basado en una temporalización, uso y/o en la superación de otros recursos. Cuando el laboratorio *online* está ubicado en el LMS podrá ser incluido como otro recurso más en la secuencia y, dependiendo de la posibilidad de comunicación VRL-LMS para poder establecer el estado de uso y superación de cada alumno en el laboratorio, podrá usarse para establecer secuencias de aprendizaje

post-VRL. En caso contrario (si el VRL no es capaz de comunicarse con el LMS para establecer el uso y estado de superación del alumno), sólo se podrán establecer secuencias de aprendizaje Pre-VRL (hasta acceder al VRL).

- **Seguimiento del trabajo realizado en el laboratorio.** Para que se realice este seguimiento es necesario que existan comunicaciones VRL-LMS que permitan al VRL enviar información relativa al uso y superación de los experimentos programados y que el LMS sea capaz de registrarla y mostrarla de forma adecuada. En este caso, se podrá acceder de forma inmediata a esta información por parte de alumnos y tutores, con las ventajas que supone para ambos.
- **Almacenamiento de resultados y calificaciones.** En el caso de que el LMS pueda realizar el seguimiento del trabajo realizado se tendrá un almacenamiento de los resultados de uso y superación que normalmente podrán ser exportados del LMS para ser incorporados en el cálculo de las calificaciones correspondientes.

3.2. Propuesta de Modos de Integración Lab-LMS

La integración Lab-LMS se ha acometido siguiendo diferentes estrategias. Las primeras aproximaciones se basaban en usar los LMS como simples contenedores de los laboratorios *online*, pero sin establecer comunicaciones entre el *software* del laboratorio y el LMS en el que estaban alojados. Se produjo una mejora sustancial en el proceso de integración Lab-LMS cuando se empezaron a desarrollar laboratorios que eran capaces de comunicarse con el LMS. En este sentido, se han realizado propuestas basadas en el desarrollo de *software* específico para la adaptación de los laboratorios en un determinado LMS [42] [278] y otras basadas en el uso de estándares que permiten la reutilización del laboratorio en los LMS compatibles [78]. Esta última posibilidad puede implementarse de dos formas diferentes que se diferencian en la ubicación del laboratorio; alojado en el mismo LMS [51] o ejecutándose de forma totalmente independiente al LMS, aunque utilizando estándares de comunicación con el LMS que permiten su integración en el mismo [275]. En [279] se propone LaaS (*Laboratory as a Service*), un modelo de desarrollo de laboratorios remotos como componentes modulares con capacidad de interactuar con sistemas remotos, lo cual permite tanto la comunicación con el recurso remoto como su integración con LMSs.

En el apartado anterior, se han tratado los beneficios que se pueden obtener con la integración Laboratorio-LMS que se corresponden con los objetivos de la propuesta de integración tratada en este trabajo de tesis. Los beneficios proporcionados por la integración dependen del modo en que ésta se produce. Se han analizado las diferentes formas en que se pueden relacionar los laboratorios *online* con los LMS y se ha realizado una propuesta de modos de integración Lab-LMS. Antes de presentar los modos de integración y beneficios que se pueden obtener en cada uno de ellos es importante recordar las siguientes ideas:

- Un LMS es el elemento tecnológico central de la vida docente universitaria, en él los alumnos encuentran los recursos y herramientas necesarias para realizar sus tareas estudiantiles y los profesores y tutores sus labores docentes.
- Un laboratorio *online* implica la existencia de una aplicación o *software* de laboratorio que a partir de este momento se va a llamar VRL. El VRL presenta siempre una interfaz gráfica de usuario (GUI en inglés) a través de la cual el usuario interactúa con un sistema real, en caso de tratarse de un VL, o con una simulación que debe estar incluida en el mismo, en caso de tratarse de un VL (si fuera un laboratorio híbrido la GUI se usaría para interactuar con un sistema real y una simulación simultáneamente).
- El VRL es un contenido de aprendizaje que, desde el punto de vista pedagógico, no debe presentarse de forma aislada a los alumnos. Al VRL le deben acompañar otros recursos que lo apoyen y sirvan para mejorar la efectividad del aprendizaje que se pretende obtener. Sólo por señalar algunos, se puede mencionar los siguientes: lista de requisitos técnicos para ejecutar el VRL, manual de uso del VRL, información sobre el laboratorio *online* y los objetivos y/o competencias que se pretenden obtener, contenidos teóricos de la materia relacionada, guiones de las prácticas que deben realizarse, pruebas de evaluación, ...

Se ha establecido una clasificación de modos de integración de un VRL con un LMS. La **Tabla 5** muestra un resumen de las principales características que presentan cada uno de los modos. El modo 0 equivale a un modo sin integración y según se incrementa el número, aumenta el grado desde un nivel de integración mínima (1) a avanzada que puede establecerse de 3 modos

diferentes (3, 4 o 5) sin que, en este caso, un número mayor implique mayor grado de integración.

Tabla 5: Modos de Integración Lab-LMS y características de cada uno

Integración	Modo	Ubicación	Comunicación con LMS	Recursos Asociados		Identificación	Portabilidad	Personalizar
				Pre-VRL	Post-VRL			
Nula	0	Ext	No	No	No	No	Sí	No
Mínima	1	Ext. (enlace)	No	No	No	No ²	No	No
Media	2	LMS	No	Posible	No	LMS	No	No
Avanzada	3	LMS	Estándar	Posible	Posible	LMS	Posible	Posible
	4	LMS	Propia	Posible	Posible	LMS	No ³	Posible
	5	Ext. y LMS ⁴	Estándar	Posible ³	Posible ³	VRL-LMS	Posible	Posible

Modo 0. No integración. El LMS y el VRL no establecen ninguna relación, ni siquiera de enlace de uno al otro ni por supuesto con ningún otro contenido del LMS. No existe ningún beneficio al no existir ningún tipo de integración.

Modo 1. VRL enlazado en LMS sin comunicaciones. En el espacio virtual de la asignatura o materia de la que forma parte el VRL, que está en el LMS, existe un enlace al VRL, el cual se encuentra ubicado en otro servidor. Esta integración es mínima, el único beneficio que se aporta es la facilidad de encontrar el VRL a través del LMS, que es un elemento habitual de uso.

Modo 2. LMS como contenedor del VRL. El VRL se encuentra ubicado en el LMS, en el espacio virtual de la asignatura o materia junto al resto de recursos de la misma. Esta integración es algo mayor, el beneficio está claro, por un lado se asegura que la ejecución del VRL sólo se produce por usuarios identificados por el LMS. Por otro lado, el alumno encuentra todos los recursos en el mismo entorno sin necesidad de acceder a otras direcciones en Internet. Además, si el LMS lo permite, algo que suele ser habitual, se pueden establecer secuencias o caminos de aprendizaje para controlar el acceso al VRL basado en la superación de otro u otros recursos (control de secuencia Pre-VRL). Un ejemplo de este tipo de integración es el que se da cuando se incrusta un VRL como parte de

² Aunque el enlace se encuentre únicamente en el LMS no se puede asegurar el control de acceso al VRL.

³ Cuando se hace uso de estándares la portabilidad entre LMSs compatibles está asegurada, al usar desarrollos propios no.

⁴ Si el LMS implementa un recurso para el estándar y permite establecer control de acceso con otros recursos asociados.

contenido SCORM pero sin utilizar la sub-especificación RTE para establecer comunicaciones con el LMS que lo aloja.

La **Figura 37** muestra un esquema de cada uno de los tres primeros modos de integración (modos 0 a 2).

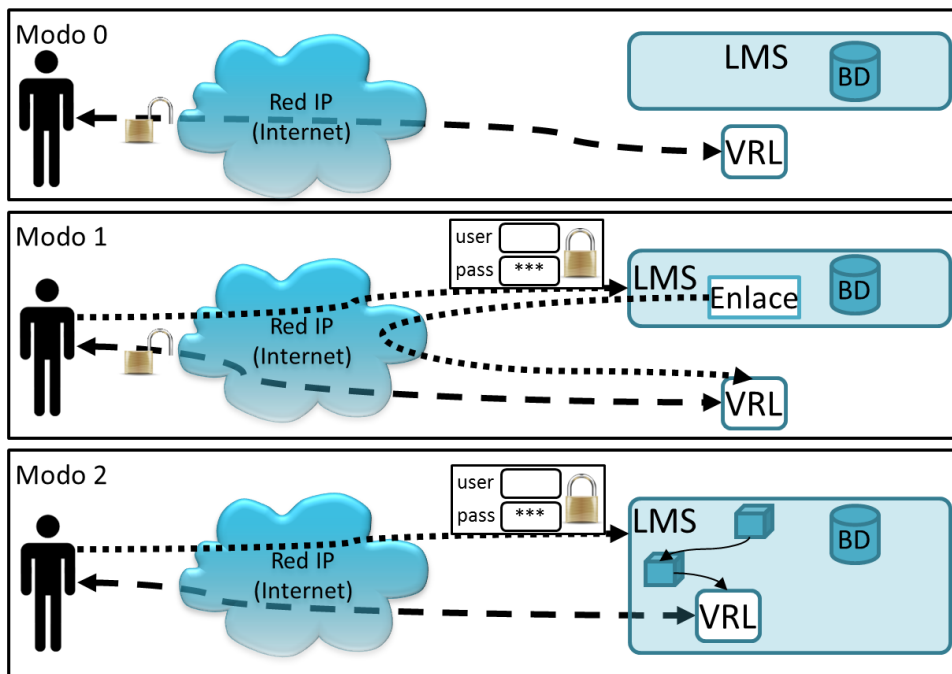


Figura 37. Esquemas de modos de integración 0 a 2

Una integración avanzada en la que existe comunicación Lab-LMS permite que la aplicación del laboratorio pueda obtener, entre otros, datos que permitan identificar al alumno. Estos datos son indispensables si se quiere proporcionar un aprendizaje personalizado y almacenar el resultado del trabajo realizado por cada alumno en el LMS. Los modos que se describen a continuación son todos avanzados y el valor de la numeración utilizada no implica, como ha ocurrido hasta el momento, que se trate de una integración superior o más avanzada.

Modo 3. LMS como contenedor del VRL que interacciona con el LMS mediante un estándar de comunicaciones. Este caso es el mismo que el anterior pero se ha añadido la posibilidad de que el VRL y el LMS se comuniquen. Si el LMS transmite la identificación del alumno al VRL, le permitiría realizar un aprendizaje adaptado al mismo y, una vez que el alumno termine de trabajar realizando los experimentos, comunicar la calificación y su resultado al LMS. De

este modo se podría usar este resultado en el LMS para controlar el acceso a otros recursos asociados incluidos en el LMS (control de acceso a recursos asociados Post-VRL). Al usarse estándares de comunicación se podría exportar el VRL para poder usarse en otros LMS que soporten el mismo estándar. Un ejemplo de este tipo de integración es el que se da cuando se incrusta un VRL como parte de contenido SCORM que utiliza la sub-especificación RTE para establecer comunicaciones con el LMS que lo aloja. En este caso, la comunicación está limitada al intercambio de información que pertenezcan al modelo de datos RTE de SCORM. También se podría realizar utilizando el estándar xAPI, aunque en este caso es necesario que se utilice un LRS (*Learning Record Store*) que puede estar implementado en el mismo LMS para que se haga cargo del almacenamiento de los registros de la actividad de aprendizaje generados por el VRL y establecer un “vocabulario” que permita definir los datos que se quieren intercambiar entre el VRL y el LMS.

Modo 4. LMS como contenedor del VRL que interacciona con el LMS mediante el desarrollo de una extensión del LMS. Este modo sería muy similar al anterior salvo que en vez de usar un estándar para establecer las comunicaciones VRL-LMS se usaría una extensión que se debe añadir al LMS para que sea capaz de comunicarse con los VRL que sean creados de un modo específico. Presenta el problema de que los VRL sólo pueden ser reutilizado en otros LMSs que sean del mismo tipo y se les haya instalado la misma extensión.

Modo 5. VRL externo al LMS que interacciona con el LMS mediante un estándar de comunicaciones. En este modo, el LMS se ejecuta de forma totalmente externa al LMS, puede hacerlo en cualquier dispositivo para el que se haya programado el VRL (móvil, Tablet, PC, etc.). El VRL debe recoger las credenciales del usuario y usar un estándar de comunicaciones que soporte el LMS (la identificación podría realizarla el LMS con las credenciales transmitidas el VRL), habrá que configurar en el LMS un objeto relacionado con el estándar en el que se configurará los parámetros de la comunicación que se va a realizar con el VRL. Entre estos parámetros es necesario utilizar un “léxico” que permita la comunicación VRL-LMS. Si el LMS lo permite, este objeto podría relacionarse con los otros objetos relacionados con el VRL que hay en el LMS estableciendo secuencias de aprendizaje Pre-VRL y Post-VRL. Además, el VRL sería reutilizable en todos los LMS que cumplieran el estándar utilizado por el VRL. Al igual que los modos 3 y 4 podría permitir un aprendizaje personalizado. Un ejemplo de este tipo de integración es el que se da cuando se tiene un VRL que utiliza el

estándar LTI para establecer comunicaciones con el LMS que lo aloja. También se podría utilizar el estándar xAPI que se comentó en el modo 3, ya que no es necesario que el contenido se encuentre en el mismo LMS.

La **Figura 38** muestra esquemas de los modos de integración 3 a 5.

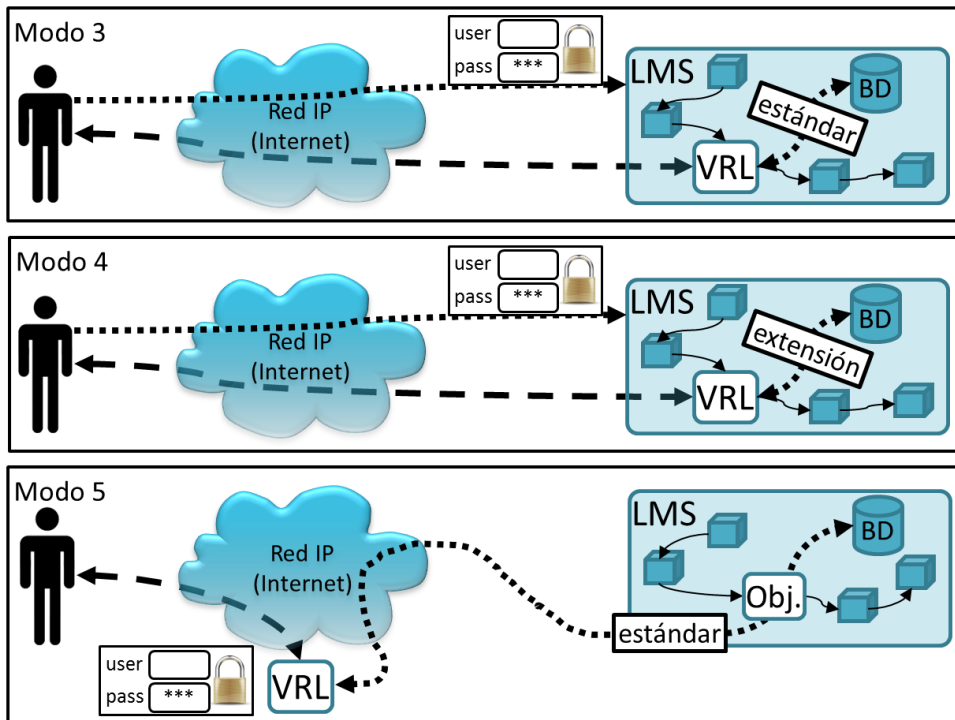


Figura 38. Esquemas de modos de Integración 3 a 5 (avanzados)

La personalización del aprendizaje o funcionamiento adaptado al estudiante que pueden ofrecer los VRL en los modos 3 a 5 debe programarse en el propio VRL. Cuando esta personalización se quiera reflejar en el LMS deberá tenerse en cuenta el vocabulario que tenga en cada caso particular, el cual puede depender del estándar o extensión utilizada. Entre las opciones de personalización que se podrían dar estarían las siguientes:

- Personalización de los experimentos a realizar: cada alumno realizaría un experimento diferente en función de su identificación.
- Posibilidad de almacenar notas personales y/o resultados del trabajo realizado en el LMS para su recuperación en ejecuciones futuras.
- Posibilidad de personalizar el funcionamiento del VRL; por ejemplo, permitir la selección de lenguaje, rapidez de ejecución, volumen,

aparición de subtítulos y de almacenar estas preferencias en el LMS para ejecuciones posteriores.

- Posibilidad de personalizar la evaluación de las prácticas y almacenar dichos valores en el LMS. No existe ninguna definición oficial de lo que es una integración avanzada entre un laboratorio *online* y un LMS, pero se puede decir que ésta se produce cuando existe una comunicación entre ambos que permite cumplir al menos esta propiedad.

Todas ellas son características deseables desde un punto de vista pedagógico que permiten elaborar contenidos docentes de mayor calidad.

3.3. Paradigma de Integración Lab-LMS Mediante SCORM

La principal estandarización elegida para el desarrollo del presente trabajo ha sido el conjunto de especificaciones SCORM. Los motivos que llevaron a usar estas normativas cuando se inició este trabajo son varios:

- Era y sigue siendo el estándar propio de *e-learning* más extendido y utilizado por los LMS, la mayoría de los LMS del mercado soportan SCORM.
- El LMS institucional de la Universidad de Jaén, ILIAS [65], soporta los estándares SCORM.
- Es un estándar “completo” que abarca diferentes aspectos del proceso normativo de contenido docente: Contenidos y evaluación, Navegación y Secuenciación e Interactividad. Estas características aporta varias ventajas:
 - Permite crear secuencias de aprendizaje que incluyan diversos recursos, uno de los cuales puede ser el VRL, en un mismo paquete SCORM. Dicha secuencia, o camino de aprendizaje, será reutilizable en cualquier LMS compatible si se exporta el paquete SCORM para ser importado en otro LMS.
 - Permite, a través de las comunicaciones SCORM, que el VRL intercambie datos con el LMS. De este modo el VRL puede ofrecer una formación adaptada al alumno y un funcionamiento personalizado basado en las preferencias del alumno y, por otro lado, el LMS puede realizar un registro del trabajo realizado por el alumno y ofrecer un seguimiento del mismo y los resultados obtenidos a alumnos y profesores.

Con este trabajo se ha logrado obtener una integración avanzada Lab-LMS en la que se pueden obtener las ventajas señaladas en el apartado “3.1. Objetivos de la Integración Lab-LMS”. Esta integración se corresponde con el modo 3 (avanzado), con la particularidad principal, que no única, de utilizar los estándares SCORM. Se usan las tres especificaciones de SCORM para encapsular el VRL junto a un subconjunto de los recursos asociados en un paquete SCORM (WebLab) formado por distintos SCOs (páginas Web) que tienen la posibilidad de comunicarse, al igual que el VRL, con el LMS para presentarse en forma de secuencia de aprendizaje variable dependiente del trabajo realizado por cada alumno:

- La sub-especificación CAM [107] permite definir una estructura de contenidos de aprendizaje basado en páginas Web y los elementos incluidos en dichas páginas, los cuales se empaquetan en un fichero de formato ZIP llamado paquete SCORM.
- La sub-especificación SN (*Sequencing and Navigation*) [109] permite establecer la forma en la que se puede navegar entre los objetos y recursos incluidos en el paquete SCORM. Es decir, se pueden establecer secuencias de aprendizaje internas al SCORM.
- Finalmente la sub-especificación RTE [108] define el modo en que el LMS ofrece los contenidos del paquete SCORM y la forma de comunicación que se puede establecer entre los contenidos incluidos en el paquete SCORM y el LMS basado en el modelo de datos predefinido (modelo de datos RTE).

Cuando un LMS soporta SCORM, los paquetes SCORM son reconocidos como recursos de aprendizaje. Su tratamiento dentro del LMS depende de la implementación y las características del LMS, pero en la mayoría de los casos los LMS permiten al menos configurar los paquetes SCORM para controlar su disponibilidad en los siguientes términos:

- **Permisos:** basado en los roles del LMS se pueden configurar diferentes permisos para los recursos (visibilidad, lectura, escritura, etc.). En este caso, los WebLabs, implementados como paquetes SCORM. Por ejemplo, permisos del WebLab en ILIAS, en la **Figura 39** se puede ver la configuración de permisos del WebLab para los roles del contexto (el WebLab se ofrece en un grupo para lograr que tengan acceso sólo un determinado número de alumnos), administrador (profesor encargado

de la docencia del WebLab en el grupo) y miembro (alumnos del grupo, que deben realizar el Weblab).

Tutor del curso	Alumno del curso	Administrador	Miembro
<input type="checkbox"/> Visible	<input type="checkbox"/> Visible	<input checked="" type="checkbox"/> Visible	<input checked="" type="checkbox"/> Visible
<input type="checkbox"/> Lectura	<input type="checkbox"/> Lectura	<input checked="" type="checkbox"/> Lectura	<input checked="" type="checkbox"/> Lectura
<input type="checkbox"/> Ver progreso de aprendizaje de otros usuarios	<input type="checkbox"/> Ver progreso de aprendizaje de otros usuarios	<input checked="" type="checkbox"/> Ver progreso de aprendizaje de otros usuarios	<input type="checkbox"/> Ver progreso de aprendizaje de otros usuarios
<input type="checkbox"/> Editar progreso de aprendizaje	<input type="checkbox"/> Editar progreso de aprendizaje	<input checked="" type="checkbox"/> Editar progreso de aprendizaje	<input type="checkbox"/> Editar progreso de aprendizaje
<input type="checkbox"/> Copiar	<input type="checkbox"/> Copiar	<input checked="" type="checkbox"/> Copiar	<input type="checkbox"/> Copiar
<input type="checkbox"/> Cambiar configuración	<input type="checkbox"/> Cambiar configuración	<input checked="" type="checkbox"/> Cambiar configuración	<input type="checkbox"/> Cambiar configuración
<input type="checkbox"/> Eliminar	<input type="checkbox"/> Eliminar	<input checked="" type="checkbox"/> Eliminar	<input type="checkbox"/> Eliminar
<input type="checkbox"/> Cambiar permisos	<input type="checkbox"/> Cambiar permisos	<input checked="" type="checkbox"/> Cambiar permisos	<input type="checkbox"/> Cambiar permisos
<input type="checkbox"/> Seleccionar todo	<input type="checkbox"/> Seleccionar todo	<input type="checkbox"/> Seleccionar todo	<input type="checkbox"/> Seleccionar todo

[Guardar](#)

Figura 39. Permisos del recurso WebLab (paquete SCORM) en el LMS ILIAS de la UJA

- Planificación Temporal:** se puede determinar un espacio temporal en el que el paquete SCORM (WebLab) estará disponible para acceder y ver o trabajar en el mismo. Por ejemplo, planificación temporal del WebLab en ILIAS. En la **Figura 40** se puede ver la planificación temporal del WebLab, que estará visible siempre para los roles con permisos para ello, pero que no podrán acceder al mismo nada más que desde las 12:26 horas del 20-09-2016 hasta las 00:59 del 23-11-2016.
- Precondiciones:** se puede configurar el acceso a un módulo SCORM para que dependa del estado asociado a otros recursos del LMS (accedido/superado/terminado). Por ejemplo, precondiciones del WebLab en ILIAS, en la **Figura 41** se puede ver las opciones de precondición del WebLab, que determinan que no será accesible hasta que el estudiante cumpla todas las precondiciones: haber cumplido el

progreso de aprendizaje del recurso “Pasos Previos a Ejecución de WebLab” (es un módulo de contenido que se debe leer) y finalizado el recurso tipo test “Test Previo a WebLab de PID de Motor CC”.

WEPLAB PID MOTOR CC: EDITAR PLANIFICACIÓN TEMPORAL

Planificación Temporal

Planificación Temporal deshabilitada
Escoja esta opción para deshabilitar cualquier planificación temporal.

Disponible temporalmente

Tiempo de Comienzo

20 Septiembre 2016

12 26

(HH:MM)

Fecha de Fin

23 Noviembre 2016

00 59

(HH:MM)

Visibilidad Visibilidad

Tiempo objetivo habilitado
Escoja esta opción para ofrecer a los alumnos sugerencias sobre las fechas de entrega/finalización

Figura 40. Planificación temporal del recurso WebLab (paquete SCORM) en el LMS ILIAS de la UJA

Crear nueva asociación

OPCIONES DE PRECONDICIÓN Guardar

Ocultar objeto

Esta opción permite ocultar el objeto a los usuarios que actualmente no hayan cumplido las precondiciones. Por lo tanto, no será visible el contenido que no es accesible.

Modo Todas las precondiciones
Todas las precondiciones deben satisfacerse para conseguir acceso.

Subconjunto de precondiciones
Se necesita un subconjunto de precondiciones para conseguir acceso. Algunas precondiciones pueden estar activadas como obligatorias

Guardar

ACTIVAR PRECONDICIONES
(1 - 2 de 2)

Eliminar

Condición de activación ↑	Condición	Obligatorio	Acciones
<input type="checkbox"/> Pasos Previos a Ejecución de WebLab Instrucciones de verificación, instalación y configuración de JAVA para la ejecución de applets (con imágenes y enlaces)	Determinado por el Progreso de aprendizaje	<input checked="" type="checkbox"/>	Editar
<input type="checkbox"/> Test Previo a WebLab de PID de Motor CC	Finalizada	<input checked="" type="checkbox"/>	Editar
<input type="checkbox"/> Seleccionar todo			

Eliminar

(1 - 2 de 2)

Figura 41. Precondiciones del recurso WebLab (paquete SCORM) en el LMS ILIAS de la UJA

Gracias a estas posibilidades de configuración de la disponibilidad de acceso a los recursos que ofrecen los LMSs se pueden crear secuencias de aprendizaje pre-Lab.

Por otro lado, con la integración avanzada obtenida con este trabajo por las comunicaciones SCORM-LMS, también se puede actualizar el estado de un paquete SCORM (WebLab) en función del trabajo realizado por el alumno en el mismo, por ejemplo, si realiza correctamente los experimentos, el estado asociado al paquete SCORM del VRL será “superado”. Esto permite crear asociaciones con otros recursos post-lab del LMS creando secuencias de aprendizaje completas. Por ejemplo, si el estudiante ha superado el SCORM puede acceder a un vídeo, que muestra la mejor solución al WebLab, y realizar una encuesta de satisfacción para recabar su opinión sobre el WebLab.

La **Figura 42** muestra un ejemplo del modo de integración desarrollado en este trabajo para crear los laboratorios *online* de la Universidad de Jaén. En el LMS ILIAS se ha creado una secuencia de aprendizaje formada por los recursos 1 a 4 del LMS (Pre-WebLab: Rec1 y Rec2 y Post-WebLab: Rec3 y Rec4) entre los que se encuentra un paquete SCORM formado por 4 SCOs, equivalentes a 4 páginas web (P1 a P4). En estas páginas Web se pueden incluir otros recursos asociados al VRL para crear otra secuencia de aprendizaje interna. Cada una de las páginas Web y el propio *software* del VRL se pueden comunicar con el LMS utilizando el estándar RTE de SCORM.

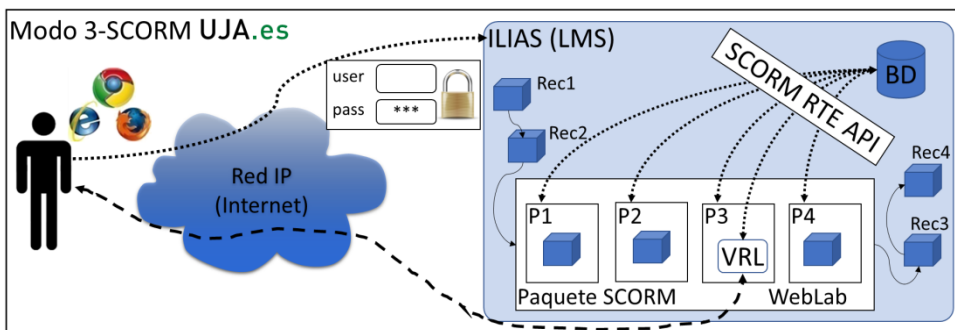


Figura 42. Ejemplo de esquema del modo de integración 3 desarrollado para la UJA

En este ejemplo no se ha especificado qué tipo de recursos son cada uno. A continuación, se muestra una posibilidad que podría haberse dado. Está basada en la plantilla SCORM de la UJA más utilizada, que está formada por 4 páginas en las que se aconseja que se incluyan los recursos que se indican a continuación:

- Rec1: un fichero PDF con una lista de requerimientos técnicos para ejecutar el VRL.

- Rec2: un test para comprobar el nivel inicial de los alumnos antes de acceder al WebLab.
- Página 1 (P1). Información genérica del Laboratorio que trata de definir y enmarcar la experiencia práctica para que los estudiantes entiendan para que sirve, cómo lo van a realizar y que se van a encontrar.
- Página 2 (P2). Contenidos teóricos específicos del VRL y los experimentos que van a realizar posteriormente, incluye recordatorio de la teoría y extensiones de la explicada en clases de teoría más relacionada con los experimentos. Prueba de evaluación para comprobar que el estudiantes posee unos conocimientos mínimos antes de acceder al VRL.
- Página 3 (P3). Breve manual de usuario de la GUI del VRL, guiones de prácticas que explican los experimentos a realizar y el propio VRL.
- Página 4 (P4). Prueba de evaluación de competencias/objetivos conseguidos por el alumno tras la ejecución del VRL.
- Rec3: vídeo con la solución correcta.
- Rec4: encuesta de opinión para conocer nivel de satisfacción del alumno.

Cap.4

4. Herramientas y Utilidades Desarrolladas

4.1. Java Frente a JavaScript

4.2. Comunicaciones SCO-LMS

0.

0.

Herramientas de Soporte para JavaScript

4.5. Otras Utilidades

En el apartado “3.3. Paradigma de Integración Lab-LMS Mediante SCORM”, del capítulo anterior, se ha explicado el modo de integración en el que se basa este trabajo que queda reflejado por el esquema mostrado en la “**Figura 42**”. Ejemplo de esquema del modo de integración 3 desarrollado para la UJA”. SCORM no es un formato muy amigable ni fácil de usar, sobre todo si no se tiene experiencia. Por ello en el ámbito de este trabajo de tesis se han creado una serie de herramientas que facilitan la comunicación entre SCORM y el LMS [77], otras que ayudan a crear contenidos de evaluación en el propio SCORM y varios modelos de paquetes SCORM que sirven como plantillas para la creación de futuros WebLabs [50]. Estas herramientas se describen con detalle en este capítulo. Para crear el *software* que deben utilizar los alumnos de un laboratorio *online* se debe considerar un lenguaje de programación. Como se ha visto en el capítulo 2, en el que se ha tratado el estado del arte, existen muchas posibilidades a la hora de elegir dicho lenguaje, en [147] se hace un análisis de los lenguajes más utilizados en su momento tanto en el lado del cliente como en el del servidor. Los primeros WebLabs desarrollados utilizaron VRL desarrollados en Java con la herramienta EJS. Para facilitar la comunicación desde el código Java del VRL hasta el LMS se creó un paquete Java que simplificaba enormemente estas transmisiones en ambos sentidos usando el estándar SCORM. Hoy en día se usa EJS (*Easy Java/JavaScript Simulations*) para crear VRL en lenguaje JavaScript y por ello se ha creado una librería JavaScript que facilita las comunicaciones VRL-LMS. Todas estas comunicaciones son las que hacen que los WebLabs desarrollados con este sistema consigan una integración avanzada, permitiendo realizar aprendizajes adaptados al estudiante. También se han creado librerías JavaScript que facilitan la creación de *tests* o pruebas de evaluación para ser insertados en los SCOs (páginas Web) que forman el paquete SCORM. Estas librerías se encargan de realizar las comunicaciones SCORM con el LMS para que almacene datos relacionados con las preguntas, respuestas y puntos conseguidos por cada usuario. Resumiendo, estas son el conjunto de herramientas que facilitan la creación del paquete SCORM y el *software* VRL que se van a describir en las próximas secciones:

- Herramientas *software* en forma de paquete Java y librerías JavaScript, los lenguajes elegidos para el desarrollo de los VRL en el lado del cliente.
- Paquetes SCORM de testeo de la implementación realizada por los LMS.
- Una serie de plantillas y modelos de paquetes SCORM.
- Documentación asociada al *software*.

Antes de describir las herramientas que se han creado, se procederá a justificar el uso de los lenguajes utilizados, Java y JavaScript, las ventajas que hoy en día ofrece uno sobre otro y, ya que la mayor parte de las herramientas que se van a mostrar están orientadas a facilitar la comunicación SCO-LMS, también se describirá su funcionamiento con detalle y las dificultades que presenta.

4.1. Java Frente a JavaScript

Cuando se inició este trabajo de tesis, el lenguaje elegido para la creación del *software* de cliente que los estudiantes debían ejecutar para acceder a los experimentos (virtuales o remotos) fue Java. Los motivos eran variados: era un lenguaje muy adecuado para escribir *software* integrado en páginas web, presentaba un amplio soporte dado por la comunidad Java en Internet, existía un gran número de implementaciones realizadas en el campo de los laboratorios *online*, la existencia de un *software* como EJS facilitaba la programación y ofrecía múltiples ejemplos y, finalmente, ofrece posibilidades de comunicación con el LMS utilizando la sub-especificación RTE de SCORM [108].

Java es una tecnología que nació en 1995 con la presentación del lenguaje de programación Java y los *applets* de Java [280]. Los *applets* de Java son programas encapsulados que se incrustan en páginas Web con el fin de proporcionar contenidos dinámicos. Existen otras tecnologías que proporcionan servicios similares que permiten ejecutar contenidos que se incluyen embebidos en contenido HTML, como son los Flash y Silverlight. Para su reproducción es necesario que los navegadores incluyan un código añadido a su programación nativa que sea capaz de ejecutar estos contenidos, es lo que se conoce como *plugin* de navegador. En el caso de los *applets* de Java estos *plugins* hacen uso de un *software* llamado Máquina Virtual de Java (*Java Virtual Machine* o JVM), encargado de permitir la ejecución del código Java de alto nivel en una determinada máquina actuando como intermediario. Los *applets* se ejecutan dentro de los confines del propio JVM, que es un programa que emula la arquitectura del dispositivo en el que se está ejecutando el *applet*. Los *applets* de Java se han utilizado durante mucho tiempo de esta forma, pero no han envejecido bien. No han evolucionado como deberían y por ello hace tiempo que no son considerados una plataforma de desarrollo moderno. Los *applets* Java persisten hoy día ya que su uso está muy extendido, su implantación es muy importante [199], [170], [166] y hay usuarios que se resisten a su cambio.

Sin embargo, más de 20 años después de la aparición de los *applets* de Java, la mayoría de las empresas responsables de los navegadores web han anunciado que eliminaban o marcaban un calendario para la eliminación del soporte de *plugins* basados en estándares de contenidos embebidos tipo *applets*, Flash o Silverlight [281] [282].

Hoy en día la mejor alternativa para poder proporcionar contenido dinámico a una página Web en el lado del cliente lo constituyen las tecnologías script, se trata de códigos de programación incrustados en el interior de un documento HTML que son interpretados y ejecutados por el navegador web del usuario. El lenguaje script más famoso, extendido, utilizado y recomendado es JavaScript. Los navegadores modernos están contruidos con soporte JavaScript incorporado. Esto significa que son capaces de interpretar y ejecutar directamente en la máquina del cliente el código JavaScript que se encuentre incluido en las páginas Web. JavaScript es un lenguaje de programación muy potente y flexible, es un lenguaje increíblemente complejo y flexible, y que está bien adaptado a la experimentación [283]. Además, JavaScript es el modo nativo de comunicación que presenta la sub-especificación RTE de SCORM [108] a través de la API JavaScript que presenta el LMS, por lo que se facilita el desarrollo de las comunicaciones VRL-LMS.

Estos son los motivos por los que, aunque los primeros desarrollos relacionados con este trabajo de tesis se hayan enfocado al lenguaje Java, se haya decidido a trabajar también con JavaScript con el fin de proporcionar herramientas con una mejor previsión de uso en el futuro.

4.2. Comunicaciones SCO-LMS

En el modelo de comunicación elegido para lograr la integración (Modo 3 – SCORM UJA.es) se hace uso de los estándares SCORM de forma que el *software* de VRL, que ejecuta el estudiante, establezca comunicaciones con el LMS. La API del RTE de SCORM, que es proporcionada por el LMS para establecer esta comunicación, consiste en el conjunto de funciones JavaScript que se describieron en el apartado “2.1.2.2.Sub-especificación RTE”. La forma en que un LMS entrega estas funciones se describe a continuación.

Cuando un LMS lanza un SCO al navegador de un estudiante (cliente), lo que hace es enviarle una página Web que incluye un objeto JavaScript con un nombre específico en el DOM del navegador. El DOM de un documento web, página HTML o XHTML, es un conjunto de utilidades específicamente diseñadas para manipularlo. Técnicamente, el DOM es una API de funciones que se pueden utilizar para manipular las páginas web de forma rápida y eficiente. La API SCORM es añadida por el LMS como un objeto del DOM de la página web del SCO, así se puede encontrar para utilizar las funciones desde el código

JavaScript que exista en la página. En la **Figura 43** se puede ver una captura de pantalla de la herramienta *Firebug* del navegador Firefox en la que se muestra el DOM de una página web de un paquete SCORM en ejecución. El objeto con el nombre `API_1484_11` es la API RTE de SCORM 2004.

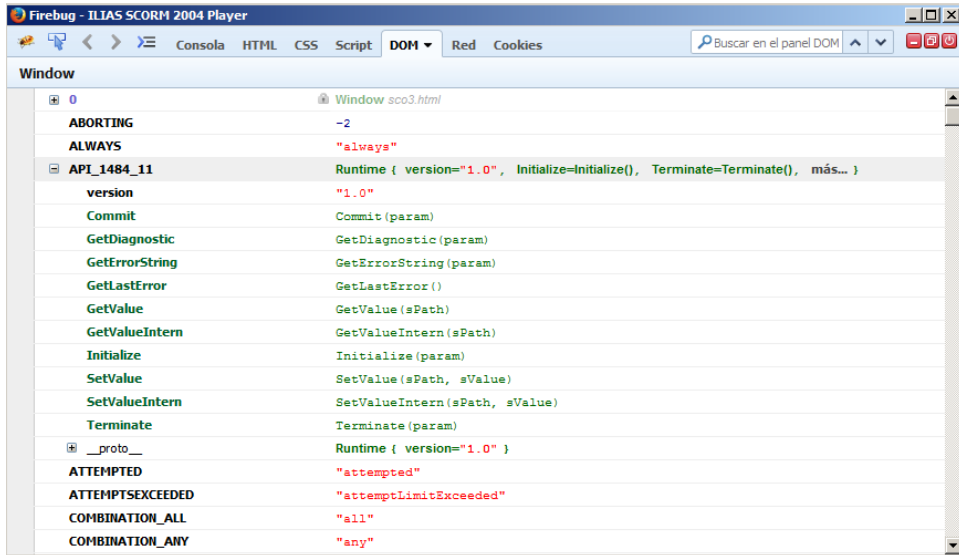


Figura 43. Captura parcial del DOM de página Web de un SCORM 2004 en ejecución (Objeto API)

ADL, que es la iniciativa, dependiente del DoD (*Department of Defense*), responsable del desarrollo y mantenimiento de SCORM, proporciona unas plantillas de paquetes SCORM y recomiendan que para crear un paquete SCORM nuevo se parta de estos modelos. Estos paquetes incluyen una librería JavaScript llamada *APIWrapper.js* que se encarga de realizar el descubrimiento del objeto API y facilita el uso de las 8 funciones básicas del RTE de SCORM. Para ello, implementa otras funciones equivalentes que, además de realizar la función de transferencia correspondiente, actualizan los códigos de error. Esta librería es importada en las páginas web de todos los SCO que componen el paquete SCORM de forma que presentan unas funciones de API equivalentes a las originales añadiendo una "capa" o envoltura a las funciones de la API RTE de SCORM. La **Figura 44** refleja esta situación mostrando un ejemplo de un paquete SCORM de 2 páginas web (SCOs), el SCORM se está ejecutando en el navegador de un usuario. Tras importar la librería *APIWrapper.js*, se puede hacer uso de la función *doInitialize()* para ejecutar la función *Initialize()* de la API que ha proporcionado el LMS. La función *Initialize()* es de obligada ejecución en todo SCO que quiera intercambiar datos con un LMS ya que inicializa las

comunicaciones. Después de su llamada se podrán ejecutar las funciones de transferencia de datos como se explicó en la **Figura 33**.

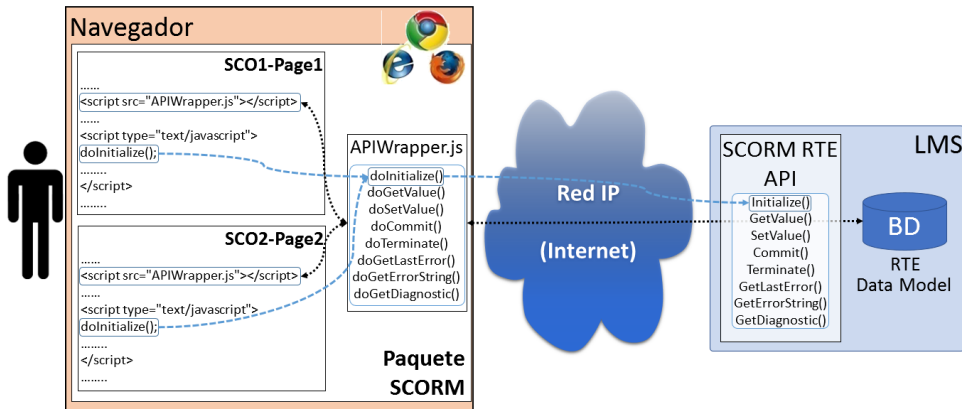


Figura 44. Ejemplo de ejecución de Paquete SCORM con 2 SCO y funciones JavaScript

4.3. Herramienta Soporte de Comunicaciones Java: scormRTE

Cuando se tiene un *applet* de Java incrustado en un SCO el esquema mostrado en la **Figura 44** se complica ya que un *applet* de Java no puede ejecutar directamente las funciones JavaScript de la librería *APIWrapper.js*. Para lograr que un *applet* de Java que está incrustado en un SCO pueda intercambiar datos con el LMS es necesario que puedan ejecutar las funciones JavaScript de la API SCORM desde el código Java. Con el objetivo de invocar funciones JavaScript desde Java se han seguido las instrucciones que se indican desde Oracle [284]. En el código del *applet* es necesario importar el paquete *netscape.javascript.** para poder crear un objeto de la clase *netscape.javascript.JSObject* con el que se podrá recuperar una referencia a los objetos JavaScript de la página web en la que se encuentra el *applet* e interactuar con la página Web. La API SCORM se presenta a la página web del SCO como un elemento del DOM, luego se podrá acceder a la misma desde el código Java. Este proceso, aunque no es complicado, añade complejidad a cualquier comunicación que se quiera realizar tanto en sentido VRL-LMS como en el sentido LMS-VRL.

Aunque inicialmente se utilizaron unas funciones propias que solucionaban este problema [78], se decidió que lo más operativo era crear un paquete Java que facilitara el proceso de toda comunicación Java-LMS. De este modo, se abstrae al programador de laboratorios Java de tener que poseer conocimientos del

estándar RTE SCORM. Así no tendrá que realizar ningún tipo de programación JavaScript. Otra ventaja adicional que presenta el uso de esta herramienta *software* es que simplifica el acceso a los datos compuestos del modelo. Un ejemplo de esta última característica se da cuando se usa la versión 2004 de SCORM desde el VRL para almacenar un comentario nuevo de estudiante en el LMS. Para hacer esto en JavaScript con el código nativo de SCORM hay que ejecutar distintas funciones:

1. Ejecución de función de lectura para obtener el número de comentarios guardados hasta el momento. Se obtiene así la posición en la que se podría guardar el nuevo comentario sin sobrescribir otro existente.
2. Ejecución de función de escritura para almacenar el comentario en la posición obtenida en el punto 1.
3. Ejecución de función de escritura para almacenar la localización desde la cual se escribe el comentario en la posición obtenida en el punto 1.
4. Cálculo de la hora y fecha actual y transformación en el formato SCORM 2004 (*Timestamp*).
5. Ejecución de función de escritura para almacenar el momento (*Timestamp*) en el que se escribe el comentario en la posición obtenida en el punto 1.

Para lograr completar estas acciones desde Java habría que hacer una invocación a cada una de las funciones utilizadas en los puntos anteriores, sin embargo, se puede crear un único método Java que tenga 2 argumentos (el comentario del alumno y la localización), que calculara automáticamente el *Timestamp* en el formato adecuado y realizara todas las acciones antes indicadas. De este modo, se simplifica enormemente la programación del laboratorio en código Java.

Así surgió el paquete JAVA ***scormRTE.jar***. Este paquete contiene dos clases. Cada una de las clases realiza una importación del paquete Java *netscape.javascript.** para poder invocar funciones JavaScript, incluye variables para almacenar todos los datos del modelo RTE SCORM y métodos Java necesarios para establecer todos los posibles intercambios de información VRL-LMS y LMS-VRL. Cada dato del modelo tiene asociado un método (si el dato es de sólo lectura o sólo escritura) o dos métodos (si es un dato de escritura/lectura). Existen otros métodos auxiliares, unos facilitan la invocación de funciones JavaScript, otros se usan para realizar el tratamiento conjunto de varios datos del modelo cuando están relacionados, y finalmente otros que

ayudan con algunos formatos específicos del estándar SCORM, como por ejemplo, la forma de expresar un intervalo temporal o un determinado momento en el tiempo (*Timestamp*). Estas son las dos clases incluidas en *scormRTE.jar*:

- La clase ***scormRTE.class*** para la versión SCORM 2004. Incluye 74 variables públicas, 30 de las cuales son arrays (2 de ellos de 2 dimensiones), que se corresponden con los datos del modelo RTE (69) y SN (5) de SCORM 2004 y 122 métodos públicos, todos ellos accesibles desde cualquier código Java que realice una importación de la clase.
- La clase ***scormRTE12.class*** para la versión SCORM 1.2. Incluye 49 variables públicas, 17 de las cuales son arrays (2 de ellos de 2 dimensiones), que se corresponden con los datos del modelo RTE de SCORM 1.2 y 75 métodos públicos, todos ellos accesibles desde cualquier código Java que realice una importación de la clase.

Un *applet* Java que haga uso del paquete *scormRTE.java* y se incruste en la página web de un SCO de un paquete SCORM podrá crear un objeto de la clase correspondiente a la versión de SCORM utilizada, *scormRTE* o *scormRTE12*, para poder usar los métodos y lograr comunicarse con el LMS fácilmente. El paquete *scormRTE.jar* se puede ver como un *software* envolvente de las comunicaciones SCORM RTE, uno más tras la librería *APIWrapper.js* que importan las páginas web de los SCOs que forman un paquete SCORM. La **Figura 45** muestra un ejemplo equivalente al mostrado en la **Figura 44** para un paquete SCORM de sólo un SCO que contiene un *applet* Java que está utilizando el paquete *scormRTE* para establecer comunicaciones con la API RTE de SCORM y recoger el nombre del usuario (elemento *cmi_learner_name* del modelo de datos RTE de SCORM).

La **Figura 46** muestra el diagrama de secuencia UML con entidades y métodos/funciones relacionados del ejemplo de la **Figura 45**. Se trata de una sesión de trabajo de un estudiante en un laboratorio *online* que está incrustado en un paquete SCORM. El estudiante (*Learner*) se conecta al LMS, se identifica, abre un módulo SCORM y entra en un SCO. La página web del SCO carga el *APIWrapper.js*, inicializa las comunicaciones SCORM (función *doInitiate()*) y ejecuta un *applet* de Java que es el VRL. El *applet* de JAVA carga el paquete *scormRTE*, crea un objeto de la clase *scormRTE* y ejecuta métodos como *rteGetLearnerName()*. Se evalúa la expresión en JavaScript (gracias al paquete *netscape.javascript* que se ha omitido aquí para no complicar el diagrama) y se

ejecuta la función *doGetValue()* del *APIWrapper* que a su vez ejecuta la función *GetValue()* de la API de SCORM (ambas con el argumento *cmi_learner_name*). Esta función obtiene el valor del nombre del usuario que ha iniciado la sesión y está almacenado en el modelo de datos del RTE de SCORM que gestiona el LMS y lo devuelve a la variable *name* del *applet* de JAVA, en este caso un VRL.

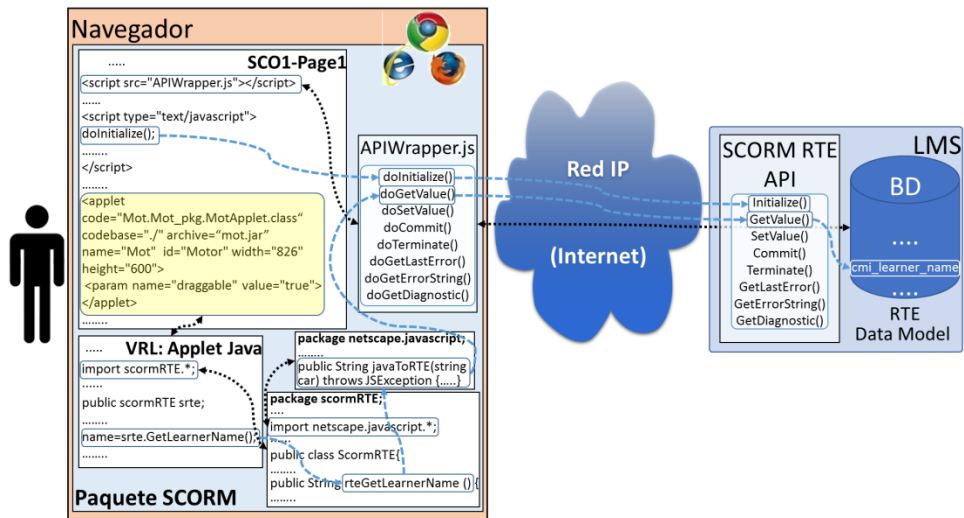


Figura 45. Ejemplo de ejecución de Paquete SCORM con 1 SCO y un applet Java que importa el paquete scormRTE

La **Figura 46** también muestra cómo se guarda la puntuación alcanzada por el estudiante (*points=10*) en el elemento *cmi_score.raw* con el método *rteSetScoreRaw(points)*. Se termina la sesión de comunicación del SCO ejecutando el método *rteTerminate()* desde el *applet* de Java.

El nombre dado a los métodos de las clases *scormRTE* y *scorm12RTE* en ambas versiones (SCORM2004 y SCORM1.2 respectivamente) ha sido uniforme, tratando que coincida siempre que ha sido posible. Los casos en los que no ha podido hacerse de este modo han sido debidos principalmente a dos causas: a) existencia de elementos en un formato que no se encuentran en el otro (esto ha sido más común para SCORM2004, que tiene un número mayor de elementos) y b) diferente implementación o características en un elemento existente en las dos versiones, por ejemplo en SCORM 1.2 se puede utilizar un elemento de tipo comentario de estudiante que es simple y único, mientras que en SCORM2004 el elemento comentario está compuesto por 3 sub-elementos (comentario, *timestamp* y localización) y es un array que permite utilizar hasta 256 comentarios. El hecho de utilizar, cuando ha sido posible, un mismo nombre de

método en las 2 versiones de SCORM permite la reutilización, con un número mínimo de cambios, del código Java generado en una versión para la otra. De este modo, la adaptación de un WebLab desarrollado originalmente para SCORM 1.2 a la versión SCORM es bastante simple ya que sólo requiere cambiar la declaración del objeto de clase *scormRTE12* por otra en la que se utilice la clase *scormRTE* y repasar los métodos utilizados que se refieren a datos del modelo que no tienen las mismas propiedades en ambas versiones. El cambio de SCORM2004 a SCORM1.2 requiere el cambio opuesto de clases y el repaso se complica porque además de comprobar los datos con propiedades distintas hay que comprobar el uso de elementos que no existen en la versión 1.2. Por este motivo se aconseja a los programadores de *applets* Java que deseen lograr una portabilidad entre versiones, que tengan en cuenta estos aspectos antes de comenzar sus desarrollos y, en caso de ser posible, traten de utilizar sólo los elementos compatibles.

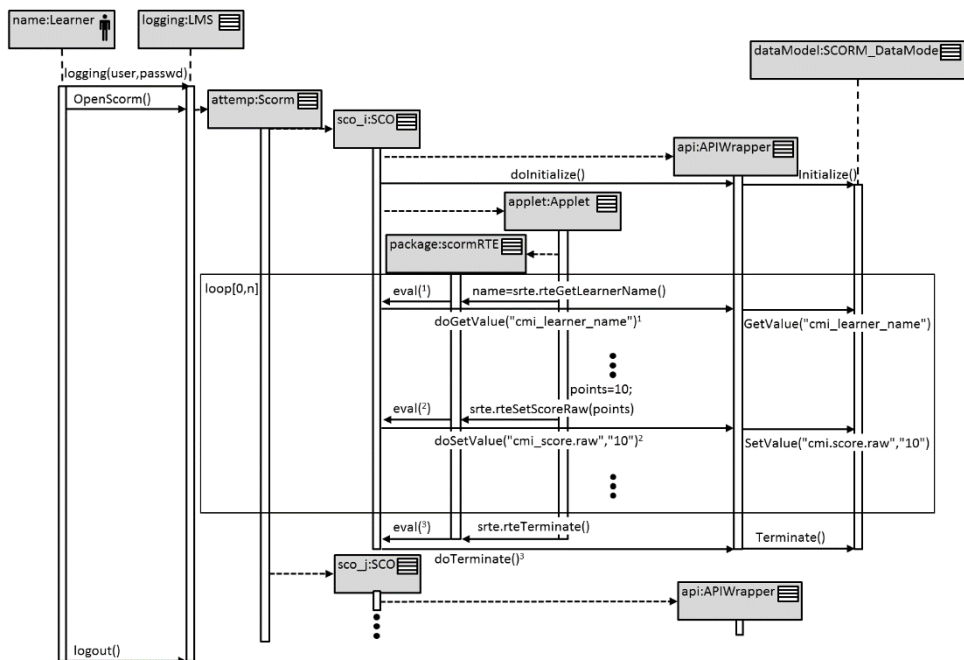


Figura 46. Ejemplo de diagrama de secuencia UML con entidades y métodos relacionados

Los métodos incluidos en las clases *scormRTE* y *scormRTE12* se pueden clasificar en 7 categorías que se enumeran y explican a continuación.

4.3.1. Método para la Invocación de Funciones JavaScript

Este método (*javaToRTE(data)*) es utilizado desde Java cada vez que se quiere invocar una función JavaScript de la librería *APIWrapper.js*, incluyendo los métodos de sesión SCORM, los métodos de almacenamiento y recuperación de datos a/desde el LMS y los métodos de soporte. El código de este método es el siguiente:

```
1: public String javaToRTE(String data) throws JSEException {  
2:     String methodOutput="";  
3:     methodOutput = (String) jso.eval(data);  
4:     return methodOutput;  
5: }
```

En la cadena *data* se incluye la cadena de código JavaScript (normalmente funciones JavaScript de la librería *APIWrapper*) que se quiere invocar en el entorno marcado por el DOM, es decir, la página web HTML en la que se encuentra el *applet* Java que contiene el código. El objeto *jso* es de la clase *JObject* que proporciona el paquete Java *netscape.javascript*, antes mencionado.

4.3.2. Métodos Constructores

Estos son métodos de tipo constructor de Java que permiten la creación de objetos *scormRTE* y *scormRTE12* con unas determinadas características antes de usar sus variables y métodos. Estos métodos, además de crear los objetos, también inicializan algunas de sus propiedades con información recuperada del LMS que podría ser requerida por el WebLab. Estos datos recuperados por los métodos constructores del LMS son todos constantes e invariantes durante la sesión de comunicación SCO-LMS. Entre estos están la versión del modelo de datos SCORM utilizada por el LMS, el nombre e identificación del estudiante, el método de acceso al SCO y el máximo tiempo permitido para completar el SCO.

4.3.3. Métodos de Sesión SCORM

Estos métodos permiten establecer y terminar las sesiones de comunicación SCORM que se pueden dar entre el SCO que contiene el *applet* Java y el LMS que lo aloja. No es necesario hacer uso de estos métodos si la página Web del SCO en la que está incrustado el *applet* Java realiza directamente las llamadas a las funciones JavaScript de iniciación y/o terminación de la sesión de

comunicación. En este sentido, se permite cualquier combinación (entre llamadas desde JavaScript y desde Java usando el paquete *scormRTE*). Es decir, la sesión de comunicación SCORM puede ser establecida por el SCO ejecutando la función JavaScript *doInitiate()* o desde el código Java del *applet* usando el método *rteInitiate()* y se puede terminar dicha sesión desde el SCO ejecutando la función JavaScript *doTerminate()* o desde el código Java del *applet* usando el método *rteTerminate()*. En el ejemplo de la **Figura 46**, la sesión de comunicación es iniciada por el SCO y terminada desde el *applet*. Este ha sido el funcionamiento programado en los laboratorios desarrollados, aunque es conveniente, para prevenir la posibilidad de finalización inesperada de la ejecución del *applet* Java sin haber podido terminar la sesión de comunicación, que se añada en el SCO un código que asegure el cierre de la sesión ante posibles fallos inesperados. Un ejemplo de tratamiento de estas situaciones es crear una función *unloadPage()* (que se ejecuta antes de descargar la página web del navegador) en la que se compruebe si se ha cerrado la sesión de comunicación y, en caso contrario, llame a la función *doTerminate()*. Para cerrar la sesión de comunicación SCORM.

4.3.4. Métodos de Recuperación de Datos del LMS

Permiten leer los datos almacenados en el modelo SCORM que gestiona el LMS, para ello hacen uso del método de invocación de funciones JavaScript descrito en el apartado 4.3.1. La función básica utilizada por estos métodos es *doGetValue(data)*, de la librería *APIWrapper*, que internamente llama a la función *GetValue(data)* de la API SCORM. Todas las funciones están personalizadas para recuperar un tipo de dato específico y, de este modo, simplificar su uso haciendo innecesario conocer todas las características del modelo de datos SCORM y su acceso. La **Figura 46** muestra un ejemplo del uso de una de estas funciones: *rteGetLearnerName()*. Cuando este método es utilizado en el código Java de una aplicación de laboratorio tipo *applet*, por ejemplo:

```
name = rteGetLearnerName();
```

Produce la invocación de la función JavaScript *doGetValue("cmi_learner_name")* desde el HTML del SCO en el que está incrustado el VRL Java. A su vez, esta función, que está definida en la librería *APIWrapper*, llama a la función *GetValue("cmi_learner_name")* de la API SCORM que suministró el LMS en el que está alojado el HTML cuando lanzó la página al navegador del estudiante. Al

ejecutarse `GetValue("cmi_learner_name")`, el LMS obtiene el nombre del estudiante que se identificó en el LMS y abrió el SCO, y lo asigna a la variable Java `name`.

De este modo sólo es necesario conocer los datos del modelo RTE de SCORM que se quieren actualizar para buscar el método que tienen que usar. Cada elemento individual del modelo, siempre que sea de lectura o lectura/escritura, tiene una función que permite leer su valor.

Adicionalmente, se han desarrollado una serie de métodos Java que cuando se utilizan permiten leer los ítems asociados a un elemento compuesto con una sólo línea de código. Esto permite simplificar enormemente el código Java de la aplicación de laboratorio. Por ejemplo, cuando se utiliza el método `rteGetCommentById(id)` se realiza internamente, y de forma transparente al programador, una llamada a los métodos `rteGetCommentsLearnerTs(id)`, `rteGetCommentsLearner(id)`, y `rteGetCommentsLearnerLoc(id)`, para obtener el comentario de estudiante almacenado en la posición `id` del modelo de datos mantenido por el LMS, el momento en que se almacenó (*Timestamp*) y su localización, cada uno de ellos de la misma forma en que se ha recuperado el valor del nombre del estudiante en párrafos anteriores. Estos tres valores son almacenados en las variables correspondientes del objeto de la clase `scormRTE` utilizada en la aplicación.

4.3.5. Métodos de Almacenamiento de Datos en el LMS

Permiten escribir valores en los datos del modelo SCORM del LMS. De forma similar a los métodos de recuperación de datos del LMS, los métodos hacen uso del método de invocación de funciones JavaScript descrito en el apartado 4.3.1. En este caso, la función JavaScript básica del `APIWrapper` utilizada es `doSetValue(data, value)`, que a su vez llama a la función `SetValue(data, value)` de la API SCORM para almacenar la información generada por la aplicación de laboratorio Java en el LMS. Estos métodos están personalizados para almacenar datos específicos del modelo de datos RTE de SCORM, facilitando su uso y reduciendo la necesidad de conocer las características detalladas del modelo y las comunicaciones. La **Figura 46** también muestra un ejemplo del uso de una de estas funciones: `rteSetScoreRaw(points)`. Donde `point` es una variable Java en la que previamente se ha asignado la puntuación alcanzada por el alumno en el laboratorio (`points=10;`). La cadena de funciones JavaScript ejecutadas es:

`doSetValue("cmi_score.raw","10");` en el *APIWrapper*, llama a la siguiente:

`SetValue("cmi_score.raw","10");` en la API SCORM, almacena el valor "10" en el sub-elemento *raw* del elemento *cmi_score* del modelo de datos RTE SCORM que mantiene y gestiona el LMS.

En esta categoría también se han creado otros métodos que facilitan el almacenamiento de elementos compuestos del modelo de datos RTE de SCORM. Por ejemplo, como ya se explicó al comienzo del apartado "0.

", cuando se quiere almacenar un comentario de estudiante nuevo en el LMS y se está utilizando la versión 2004 de SCORM se tienen que usar las siguientes funciones (se supone que se ha creado un objeto de la clase *ScormRTE* llamado *srte*):

1: `id=srte.rteGetCommentsLearnerCount();`

usada para conocer el número de comentarios almacenados y guardar en la variable *id* la posición donde se puede almacenar uno nuevo. Como el índice empieza en 0 el número de comentarios que hay coincide con la primera posición libre en la que se puede almacenar un comentario nuevo.

2: `srte.rteSetCommentsLearner (comment, id);`

usada para grabar el comentario, almacenado en la variable *comment*, en la posición *id* del array de comentarios de estudiante que hay en el LMS.

3: `timestamp = fechaHoraScorm ();`

usada para calcular la fecha y hora en formato SCORM2004, y guardarlo en la variable *timestamp*, que será utilizada a continuación.

4: `srte.rteSetCommentsTs (timestamp, id);`

usada para grabar la fecha y hora en el que se está haciendo este comentario, almacenado en la variable *timestamp*, en la posición *id* del array de fecha y hora en las que se realizan los comentarios de estudiante que hay en el LMS.

5: `srte.rteSetCommentsLoc (location, id);`

usada para grabar la localización desde la que se está haciendo este comentario, almacenado en la variable *location*, en la posición *id* del array de localizaciones de comentarios de estudiante que hay en el LMS.

Todo esto se puede hacer de forma automática con un único método, si se usa *rteSetCommentsLearnerNew (comment, location);* tiene el mismo efecto.

4.3.6. Métodos de Soporte para las Comunicaciones

Estos métodos llaman a funciones de soporte de las comunicaciones de la API de SCORM que sirven para asegurar la consistencia de los datos almacenados (*rteCommit()*) y el tratamiento y diagnóstico de los errores que se pueden producir en las comunicaciones SCO-LMS comprobando el estado de la conexión y el resultado de las transferencias de datos (*rteGetLastError ()*, *rteGetErrorString (errorCode)* y *rteGetDiagnostic (errorCode)*).

4.3.7. Métodos de Ayuda en Comunicaciones y Formato de Datos

Estos métodos simplifican el tratamiento de los formatos de algunos datos requeridos por los modelos SCORM. Por ejemplo, el método *fechaHoraScorm()* devuelve una cadena de caracteres con la fecha y hora en el formato SCORM 2004: “yyyy-MM-ddTHH:mm:ss”, o con la hora en el formato SCORM 1.2.: “HH:mm:ss.SS”, dependiendo de la clase utilizada para crear el objeto.

4.3.8. Requisitos, Consejos y Ejemplos de Uso

A continuación se va a explicar los pasos mínimos que habría que dar para crear un *applet* Java con soporte de comunicaciones SCORM y algunos ejemplos de código para usar los datos más característicos de un laboratorio. Para ilustrar estos ejemplos se ha supuesto que se utiliza el *software* EJS para programar el laboratorio.

Importación del paquete scormRTE. El primer requisito para crear un VRL Java es tener localizado el paquete *netscape.javascript* y *scormRTE* para poder importarlos desde la plataforma de desarrollo. Si el paquete *netscape.javascript* está instalado en Java, como *scormRTE* contiene una línea de importación al mismo no sería necesario añadirlo en el código Java. Para lograr la importación del paquete *scormRTE* en EJS hay que escribir “scormRTE.*;” en el campo “Imports” de la pestaña “Opciones de Ejecución” como muestra la **Figura 47**.

Además la Ubicación del paquete *scormRTE* debe ser:

```
<carpeta_EJS>/bin/extensions/scormRTE.jar
```

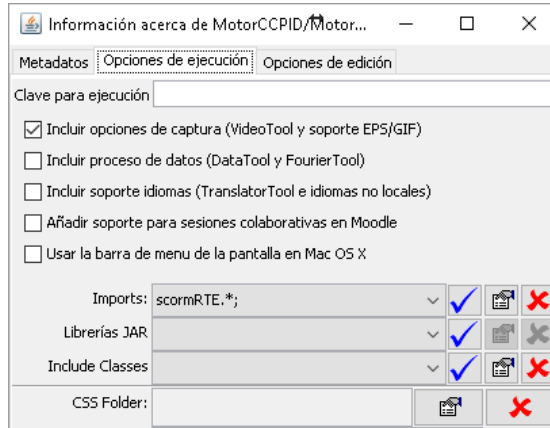


Figura 47. Pestaña "Opciones de Ejecución" de EJS

Declaración del objeto SCORM. En la opción "Modelo" – "Variables" se recomienda crear una página para las variables relacionadas con las comunicaciones SCORM. En la **Figura 48** se muestra cómo en el ejemplo se ha creado una página llamada "RTE" en la que se ha creado una variable llamada *srte* (se podía haber puesto cualquier nombre de variable) a la que se le ha asignado el tipo objeto y se ha escrito el nombre de la clase *ScormRTE* porque se quiere seguir la versión SCORM2004. En caso de querer utilizar la versión SCORM1.2 debería haberse escrito un nombre de clase *ScormRTE12*.



Figura 48. Vista de página de variables en EJS

Inicialización. En la opción "Modelo" – "Inicialización" se debe escribir un código relacionado con las comunicaciones SCORM para comprobar que el código Java se está ejecutando como un *applet*. En la **Figura 49** se muestra

cómo se hace esta comprobación, y en caso afirmativo, se inicializa el objeto *srte* de la clase *ScormRTE* mediante el método constructor *ScormRTE(applet)*.



Figura 49. Vista del código de Inicialización en EJS

Se debe tener en cuenta que las comunicaciones Java-LMS con SCORM sólo funcionan cuando se ha identificado correctamente en un LMS y se ha abierto un *applet* insertado en una página Web que sea un SCO de un paquete SCORM que ha sido instalado correctamente en un LMS y lanzado por el mismo al navegador. Por eso hay que comprobar que se trata de un *applet* en ejecución. Además, para que el *applet* se ejecute es necesario que el dispositivo del usuario tenga instalada la JVM y un navegador adecuado configurado con el *plugin* de Java activo.

En este momento se debe inicializar la sesión de comunicación SCO-LMS, en el código de la **Figura 49** no se ha incluido ya que se supone que ha sido realizado desde el SCO en el que está insertado el *applet* al cargar la página HTML. Otras de las acciones que ya se pueden realizar cuando se ha inicializado el objeto *scormRTE* es recuperar del LMS variables que permanezcan constantes en toda la sesión de comunicación:

- En la variable local *datetime* se guarda la fecha y hora de entrada (SCORM2004) o sólo la hora de entrada (SCORM1.2) en el formato SCORM correspondiente a la versión utilizada. Esta línea se puede usar en las 2 versiones.
- En la variable local *version* se guarda la versión del modelo de datos SCORM 2004 que se usa en el LMS (esto no se puede hacer en la versión SCORM1.2 ya que no tiene el elemento *version*). Esta línea debería eliminarse o comentarse en caso de usar la versión SCORM1.2.

- En las variables locales *learnerName* y *learnerId* se guardan el nombre y el identificador del estudiante en el LMS y el modelo SCORM. Estas líneas se pueden usar en las 2 versiones. Esta información es muy importante ya que permite realizar **experimentos personalizados adaptando el contenido a cada usuario** y/o cambiar parámetros de los experimentos en función de la persona que los realice.
- En la variable local *mode* se guarda el modo en el que se presenta al estudiante el SCO (puede ser *browse*, *normal* o *review*), es un dato del modelo SCORM. Esta línea se puede usar en las 2 versiones.

Desarrollo de la sesión. SCORM incluye elementos en el modelo RTE para almacenar las preferencias de los estudiantes de idioma, velocidad de entrega de contenido, volumen del audio y subtítulos. Estos elementos pueden ser preguntados al estudiante en el VRL y almacenados en el LMS para que se recuerde en futuras ejecuciones, estas son las funciones a utilizar:

- 1: `srte.rteSetLearnerPreferenceAudio(audio);`
- 2: `srte.rteSetLearnerPreferenceLanguage(language);`
- 3: `srte.rteSetLearnerPreferenceSpeed(speed);`
- 4: `srte.rteSetLearnerPreferenceCaptioning (captioning);`

De este modo, si el paquete SCORM permite múltiples ejecuciones, en las ejecuciones posteriores del SCO se puede recuperar esta información almacenada en el LMS para usarla en consecuencia mediante los siguientes métodos:

- 5: `audio=srte.rteGetLearnerPreferenceAudio();`
- 6: `language=srte.rteGetLearnerPreferenceLanguage();`
- 7: `speed=srte.rteGetLearnerPreferenceSpeed();`
- 8: `captioning=srte.rteGetLearnerPreferenceCaptioning();`

Cuando en el desarrollo del VRL se ofrece una pregunta de test o prueba de evaluación parcial se puede guardar constancia de ello en el LMS así como de la respuesta y resultado obtenido por el estudiante como fruto de su trabajo. En SCORM estas pruebas se conocen como interacciones (*interactions*). Por ejemplo, si se quiere poner una pregunta (identificada como *id="prueba1"*), en la que se va a pedir al estudiante que introduzca un determinado valor numérico (*type="numeric"*), que debe calcular después de realizar un determinado experimento, cuyo valor debe estar comprendido entre 4 y 5 para

ser correcto (*pattern="4:5"*), no hay más soluciones correctas posibles (*num patt=0*) y la pregunta tiene un peso del 50% de la nota final (*weight="50"*) se puede guardar constancia del mismo en el LMS de la siguiente forma:

```
1: num=srte.rteGetInteractionsCount();
2: srte.rteSetInteractionsId(id, num);
3: srte.rteSetInteractionsType(type, num);
4: srte.rteSetInteractionsTs(num);
5: srte.rteSetInteractionsCorrectRespPattern(pattern, num, numpatt);
6: srte.rteSetInteraccionsoftwareigh(weight, num);
```

Del mismo modo, si un estudiante realiza el experimento y accede a la prueba para responder en el VRL un determinado valor (*response="4.5"*) pasados 5 minutos desde que se le presentó la prueba (*latency="PT0H5M"*) y obteniendo una puntuación de 10 (*result=10*), se podrá guardar constancia en el LMS de este modo:

```
7: srte.rteSetInteractionsResponse(response, num);
8: srte.rteSetInteractionsResult(result, num);
9: srte.rteSetInteractionsLatency(latency, num);
```

En la versión SCORM2004 se podría acceder más tarde a estos datos, almacenados en el LMS, desde el mismo VRL, por ejemplo, para generar vistas resumen del trabajo del alumno antes de acabar el laboratorio. Para ello hay que utilizar las funciones tipo *Get* correspondientes, en la versión 1.2 el número de sub-elementos de tipo interacciones que se pueden leer está muy limitado.

También se pueden almacenar comentarios del estudiante en el LMS durante la ejecución del VRL y recuperarlos más tarde (en SCORM 1.2 sólo se permite uno, en SCORM 2004 hasta 256). Los comentarios pueden ser usados por el programador del VRL para guardar información que se puede recuperar en posteriores ejecuciones referidas al trabajo del alumno o el estado del sistema.

Estos son sólo algunos ejemplos de los intercambios de datos que se pueden realizar durante la ejecución, hay múltiple elementos del modelo de datos que no se han mencionado.

Finalización. Antes de terminar la ejecución del *applet* del VRL es conveniente almacenar el resultado del trabajo realizado por el estudiante en el LMS para que esta información pueda estar disponible para el estudiante y profesorado

en el futuro. La información típica que se suele guardar es la puntuación final, si ha completado todas las tareas asignadas y si el estudiante ha superado el laboratorio. Esto se puede lograr con las siguientes líneas de código:

```
1: scoreRaw=srte.rteSetScoreRaw(points);
2: completionStatus="completed";
3: successStatus="passed";
4: completionStatus=srte.rteSetCompletionStatus (completionStatus);
5: successStatus=srte.rteSetSuccessStatus(successStatus);
```

Si se está utilizando SCORM2004, se puede utilizar la sub-especificación SN, que controla la secuenciación y navegación entre los SCOs que componen el paquete SCORM del WebLab, para programar el próximo destino al que se desea que acceda el estudiante en función del resultado de su trabajo en el *applet* del VRL. Por ejemplo, basándose en la puntuación final obtenida por el estudiante, si obtuvo menos de un 5 se cerrará todo el paquete SCORM y el estudiante deberá abandonar el WebLab, pero en caso contrario se le presentará el siguiente SCO. Para ello se deben añadir estas líneas en el código que en principio no tendrán efecto hasta que se llame a la función de terminación de la sesión de comunicación SCO-LMS (*Terminate()*):

```
1: navReq=(points<5)? "exitAll":"continue";
2: srte.rteSetNavRequest(navReq);
```

Ahora ya sí se podría dar por terminada la sesión de comunicación SCO-LMS, como se indicó anteriormente se puede realizar desde el *applet* VRL o desde la página HTML en la que está insertado. Si el control de la secuencia de trabajo se ha realizado desde el *applet* lo lógico es que sea el mismo *applet* el que cierre esta sesión. Para ello se usa la siguiente línea de código:

```
1: srte.rteTerminate();
```

Para terminar este apartado en el que se indican consejos de uso y programación de un *applet* de Java como VRL hay que indicar que es muy aconsejable realizar la firma del código para evitar problemas de seguridad a la hora de su ejecución. Cuando se ejecuta un *applet* Java en un navegador lo hace en un entorno restringido llamado "*sandbox*" (caja de arena). Esto evita que el *applet* JAVA tenga acceso a recursos del sistema: ficheros, conexiones de red, impresoras, cámaras, micrófonos, etc. También hace que la ejecución de *applets*

Java sea más segura. Si un *applet* necesita acceder a alguno de estos recursos debe estar firmado digitalmente como muestra la **Figura 50**.

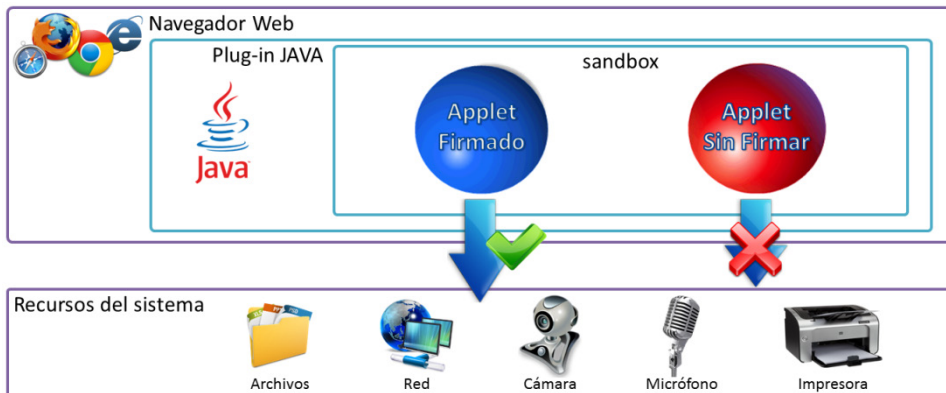


Figura 50. Ventajas de la firma de *applets* Java

En el caso de que el VRL sea de un laboratorio remoto la firma de código resulta indispensable.

Para realizar la firma de código Java en la Universidad de Jaén se tuvo que realizar una petición de un certificado de firma electrónica de código que fue solicitada a través de RedIRIS y suministrada por la empresa *Comodo Security Services* (<https://www.comodo.com/>). Todos los *applets* Java utilizados en los laboratorios remotos y virtuales han sido firmados digitalmente siguiendo el procedimiento adecuado para proporcionar mayor seguridad. Por ello a la hora de ejecutar un *applet* de Java aparece en el mensaje de petición de permiso de ejecución que el *applet* ha sido firmado por la Universidad de Jaén, como muestra el ejemplo de la figura **Figura 51**.

4.4. Herramientas de Soporte para JavaScript

JavaScript es el lenguaje de programación más utilizado en desarrollo y creación de documentos Web. Esto se debe a que junto a HTML y CSS forman el trío de tecnologías que todo programador Web debe conocer y dominar. Existen múltiples tecnologías que lo utilizan de forma nativa como son AJAX y HTML5. Además es el lenguaje natural utilizado para las comunicaciones RTE de SCORM. Es soportado por todos los navegadores modernos y más utilizados, que poseen intérpretes del mismo sin necesidad de utilizar ningún tipo de *plugin* o añadido al *software*. Esta característica hace que cualquier estudiante pueda abrir su

navegador y ejecutar código JavaScript incluido en páginas Web sin necesidad de hacer ninguna instalación ni configuración de *software* adicional.

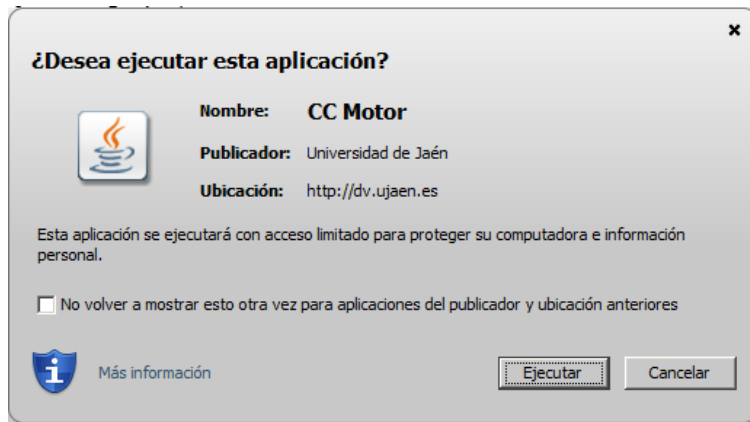


Figura 51. Ejemplo de mensaje de petición de Ejecución de applet Java

Se han desarrollado una serie de librerías orientadas a distintos fines: a) facilitar la comunicación entre aplicaciones JavaScript incluidas en un SCO (página Web) y el LMS en el que está alojado y b) facilitar la creación automatizada de *tests* de evaluación utilizando las comunicaciones RTE de SCORM. En los siguientes subapartados se va a realizar una explicación de las mismas.

4.4.1. Librería JavaScript *RTE.js* para Comunicaciones SCO-LMS

Se ha creado una librería JavaScript llamada RTE.js para facilitar las comunicaciones de código JavaScript con el LMS usando SCORM. RTE.js ofrece las mismas prestaciones que las que se han explicado para el paquete Java scormRTE en el apartado anterior (0.

), aunque ofrece algunas ventajas significativas y un funcionamiento diferente.

La librería *RTE.js* facilita la integración de código JavaScript incluido en el interior de un SCO (página web) de un paquete SCORM, con el LMS donde está alojado. *RTE.js* puede utilizarse también por aplicaciones de VRL embebidas en la misma página Web, ya sean desarrolladas en lenguaje JavaScript o con lenguaje Java en forma de *applet*. Ofrece una capa intermedia de comunicaciones previa al uso del *APIWrapper* tal y como muestra la **Figura 52**.

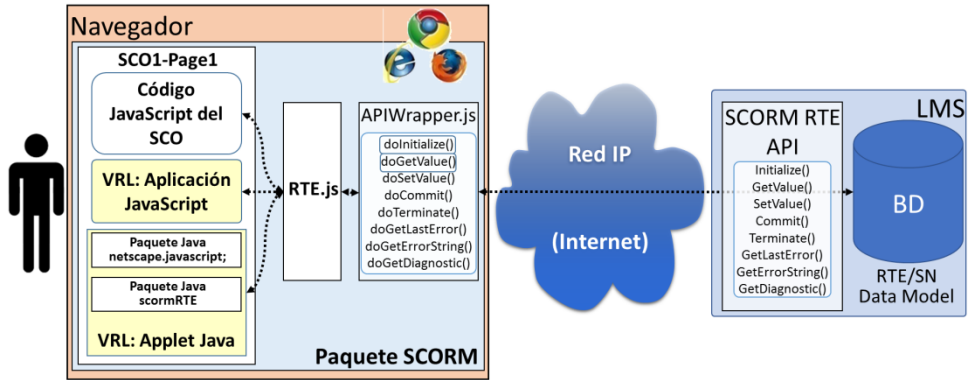


Figura 52. RTE.js como capa intermedia de comunicaciones RTE SCORM con el LMS

Como se puede ver en la **Figura 52**, el uso de *RTE.js* es compatible con el uso del paquete Java *scormRTE*. Para usarla desde Java hay que usar la función de invocación de código JavaScript (*javaToRTE(expresión)*) poniendo como expresión la función de *RTE.js* deseada.

A pesar de poder utilizarse desde distintas fuentes, el uso más común para el que fue diseñado es como capa intermedia facilitadora de comunicaciones SCORM para aplicaciones VRL escritas en lenguaje JavaScript embebidas en la página web de un SCO. *RTE.js* incluye una función llamada RTE que ha sido definida para trabajar como una clase. La función RTE incluye 83 variables y 146 funciones que proporcionan el soporte para las versiones SCORM2004 y SCORM1.2 aunque algunos de estos elementos son exclusivos de una de las versiones como se puede ver en la **Tabla 6**.

Tabla 6. Elementos de la función RTE

Versiónes SCORM	Relación directa con modelo de datos RTE/SN	Variables	Funciones
Comunes	Sí	45	71
(SCORM2001/1.2)	No	5	10
SCORM2004	Sí	29	60
SCORM1.2	Sí	4	5
Totales		83	146

La función *RTE* sólo tiene un argumento llamado *version* que indica la versión de SCORM utilizada en el código (“2004” o “1.2”). Esto permite utilizar el mismo

código para las dos versiones siempre que se cambie el valor del argumento *version* y se tenga cuidado para no utilizar elementos exclusivos de una versión.

Variables de RTE. Para optimizar la ejecución de la librería *RTE.js* las variables de la función *RTE* son declaradas pero no definidas hasta que se realiza su primer uso. Si una de las variables no se utiliza en una sesión SCORM no será definida. La mayoría de las variables tienen una correspondencia directa con un elemento del modelo de datos SCORM. Solamente hay 5 variables que se usan internamente en la librería y no necesitan esta correspondencia. Hay 45 variables comunes a las 2 versiones de SCORM, 29 exclusivas de SCORM2004 y 4 variables exclusivas de SCORM1.2. De las 83 variables que existen 31 de ellas son *arrays*, de los que a su vez 13 son exclusivos de SCORM2004 y el resto son comunes a ambas versiones.

Funciones de RTE. Las 146 funciones implementadas en *RTE* realizan tareas similares a las descritas en la sección dedicada al paquete *JAVA scormRTE* sobre los métodos de las clases *scormRTE*. Se pueden clasificar de la misma forma salvo los utilizados para invocar funciones JavaScript, que no son necesarios al tratarse de código JavaScript, y los Constructores, que tampoco son necesarios al no existir la filosofía de creación de objetos, por lo que en este caso resultan sólo 5 categorías, en vez de las 7 que existen en el paquete *Java scormRTE*:

Funciones de sesión SCORM. Inician y terminan sesiones de comunicación SCORM que se pueden dar entre el SCO que contiene el código Java y el LMS que lo aloja.

Funciones de recuperación de datos del LMS. Leen datos almacenados en el modelo SCORM que gestiona el LMS usando la función *doGetValue(data)*, de la librería *APIwrapper*, que a su vez llama a la función *GetValue(data)* de la API SCORM. La **Figura 53** muestra un ejemplo de uso de estas funciones, antes se debe declarar una variable de la clase RTE (en el ejemplo *miRTE*) e indicar la versión de SCORM utilizada como parámetro ("2004"). En la variable *Nombre* se quiere obtener el nombre del estudiante, por lo que se utiliza la función *getLearnerName()* de la librería *RTE.js*, que llama a la función *doGetValue("cmi_learner_name")* de la librería *APIWrapper*, que hace uso de la función *GetValue("cmi_learner_name")* de la API SCORM, la cual accede a la base de datos mantenida por el LMS y devuelve el valor del elemento *cmi_learner_name* del modelo RTE SCORM, el nombre del

estudiante en el LMS, en este caso “Juan Maroto” que es asignado a la variable *Nombre* en la aplicación de laboratorio JavaScript.

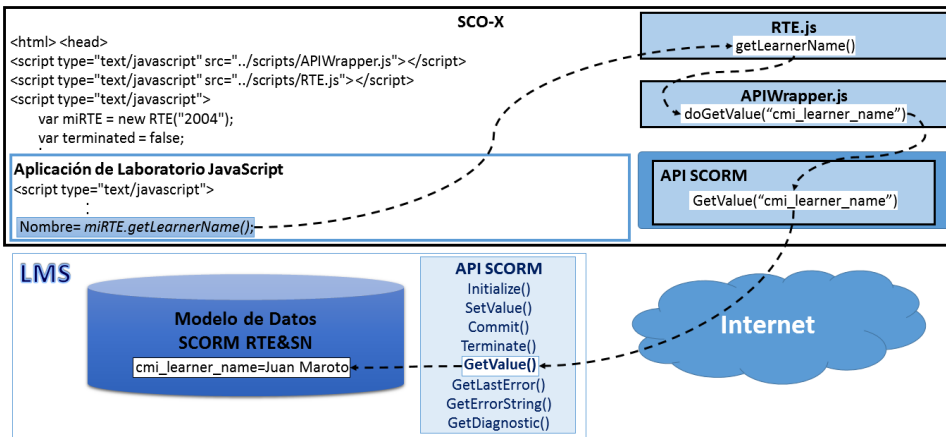


Figura 53. Ejemplo de código JavaScript embebido en una página HTML SCORM usando la librería RTE.js para obtener el nombre del estudiante del LMS

Se han creado un mayor número de funciones para lectura de datos múltiples en elementos compuestos como son los comentarios en SCORM2004, las interacciones, las preferencias de usuario y los objetivos.

Funciones de almacenamiento de datos en el LMS. Permiten escribir valores en los datos del modelo SCORM del LMS. En este caso, la función JavaScript básica del *APIWrapper* utilizada es *doSetValue(data, value)*, que a su vez llama a la función *SetValue(data, value)* de la API SCORM para almacenar la información generada por la aplicación de laboratorio Java en el LMS. Al igual que en la categoría anterior, también se han creado un mayor número de funciones para escritura de datos múltiples en elementos compuestos como son los comentarios en SCORM2004, las interacciones, las preferencias de usuario y los objetivos.

La **Figura 54** muestra el diagrama de secuencia UML correspondiente a un ejemplo de sesión de comunicación SCORM en el que se puede ver la creación de la variable función *RTE* para la versión 2004 de SCORM (*miRTE("2004")*), la llamada a la función de iniciación de sesión (*miRTE.Initialize()*), la lectura del nombre del estudiante participante en la sesión (*miRTE.getLearnerName()*), el almacenamiento

de la calificación alcanzada por el estudiante en el laboratorio (*miRTE.setScoreRaw(points)*) y la terminación de la sesión de comunicación (*miRTE.Terminate()*).

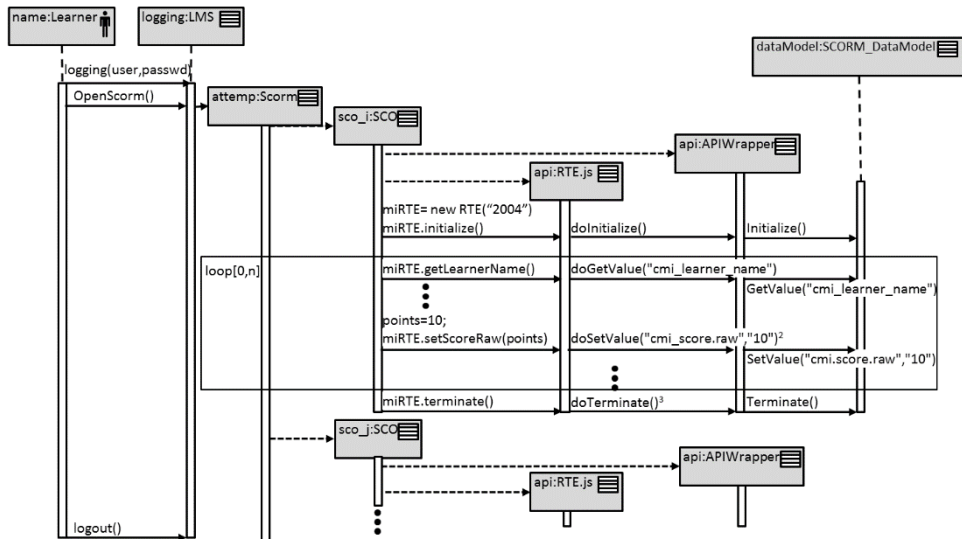


Figura 54. Diagrama de secuencia UML de un ejemplo de sesión de comunicación SCORM con entidades y funciones relacionadas

Funciones de soporte de las comunicaciones. Llamamos a funciones de soporte de las comunicaciones de la API de SCORM para el almacenamiento consistente de los datos en el LMS y el tratamiento de errores de comunicaciones SCORM.

Funciones de ayuda en comunicaciones y formato de datos. Simplifican el tratamiento de los formatos de algunos datos requeridos por los modelos SCORM.

4.4.2. Librerías JavaScript para Creación de Tests Automatizados

La evaluación del trabajo realizado por el alumno en sus trabajos experimentales es una fase, dentro de la labor docente, de gran importancia ya que sirve para comprobar el nivel de los conocimientos adquiridos y, con ello, la efectividad de la experiencia de aprendizaje implementada. Por otro lado, la autoevaluación ha demostrado ser una gran herramienta para que los estudiantes puedan conocer el nivel alcanzado y con ello, si tienen necesidades que deban completar.

Teniendo en cuenta el modelo de integración de laboratorio *online* con LMS elegido (modo 3 basado en uso de estándares), las opciones de evaluación del trabajo realizado por los estudiantes son múltiples, aunque limitadas:

- Evaluación en laboratorio. El propio *software* del VRL puede realizar pruebas de evaluación comprobando el trabajo que realiza el estudiante mientras trabaja.
- Evaluación en SCORM. En el modelo de integración VRL-LMS utilizado por la UJA, basado en SCORM, las pruebas de evaluación también pueden realizarse como contenido Web en otra parte del paquete SCORM en el que está incrustado el VRL.
- Evaluación en LMS. Ofrece varias alternativas que se pueden aplicar a los experimentos *online*, como el envío de trabajos/actividades, edición de wikis u otros tipos de documentos soportados por el LMS, participación en foros, número/tiempo de accesos a un recurso o *tests*. Hay pocas automatizadas que no requieran la participación del equipo docente, la más habitual es el uso de *tests*, el tipo de preguntas está limitado aunque se buscan nuevas alternativas [285] de forma que sean útiles para los docentes y se puedan implementar en los LMS [286].
- Evaluación externa. Se han desarrollado sistemas de evaluación automatizada *online* [287] que pueden ser utilizadas en laboratorios *online* [288], [289]. Desde el *software* del laboratorio *online* se puede acceder a estos servicios externos al LMS y al laboratorio que realicen la evaluación.

Se han desarrollado una serie de 4 librerías JavaScript (*principal.js*, *creaPreguntas2004.js*, *creaPreguntas12.js* y *creaPreguntasSinCom.js*) para generar de forma automática preguntas de test en páginas Web o aplicaciones JavaScript con las siguientes características. A partir de un fichero de definición de preguntas genera código HTML capaz de mostrar un *test* de preguntas interactivo totalmente operativo en una página Web. Además, en caso de estar incluido en un SCORM, es capaz de establecer comunicaciones con el LMS para registrar las interacciones. Las **características** detalladas se muestran a continuación:

- Soporta preguntas de test de 4 tipos: opción múltiple-respuesta simple, opción múltiple-respuesta-múltiple, respuesta numérica y relleno de huecos.

- Para definir las preguntas se utiliza el estándar QTI [290]. QTI es un estándar de formato para la representación de contenidos y resultados de evaluaciones educativas, permitiendo el intercambio de *tests* y preguntas de *test* entre LMSs.
- Permite seleccionar las preguntas, y las características de presentación y ejecución del test:
 - Se puede determinar el fichero de definición de preguntas QTI del que se van a obtener las preguntas a utilizar en el test.
 - Definir el número de preguntas que van a aparecer en el test.
 - Definir el tiempo máximo de trabajo en minutos de un alumno en el test.
 - Aleatorizar la aparición de las preguntas del test.
 - Aleatorizar el orden de aparición de las opciones de respuesta en las preguntas de opción múltiple.
 - Determinar si se van a realizar comunicaciones entre el SCO de la página web donde se encuentra el test y el LMS.
- Una vez comenzado un test aparece en la esquina superior derecha una ventana flotante semitransparente tipo *alert* de JavaScript, que muestra la puntuación alcanzada hasta el momento y la máxima disponible. También aparece otra ventana similar en la zona superior izquierda mostrando el paso del tiempo desde que se muestra la primera pregunta del *test*, cuando se le ha configurado un tiempo máximo para su ejecución. Se ha utilizado la librería *smoke.js*, obtenida de Internet (<http://smoke-js.com/>), para implementar estos efectos.
- Utiliza unos ficheros CSS, que deben incluirse en el paquete SCORM, en donde se definen los estilos utilizados y que pueden cambiarse para adaptarse al resto de contenido HTML en el que se quieran insertar: *estilo.css* y *smoke.css*.
- Las preguntas van apareciendo de una en una ofreciendo la posibilidad de responder y a) **validar** la respuesta, en cuyo caso se hará la corrección automática, b) **borrar** la respuesta, que volverá a mostrar el formulario de la pregunta en su posición inicial, borrando todos los datos y selecciones marcadas o c) **posponer** la pregunta haciendo que aparezca la siguiente.
- En caso de usar las comunicaciones SCORM debe incluirse en el paquete SCORM la librería *RTE.js* descrita en el apartado “4.4.1. Librería JavaScript *RTE.js* para Comunicaciones SCO-LMS”. Como *RTE.js* hace uso

de *APIWrapper.js* también deberá añadirse ésta. Soporta las versiones SCORM 2004 y SCORM1.2 de comunicaciones SCO-LMS. Cuando se han configurado comunicaciones, de forma automática, al aparecer cada pregunta se envían al LMS los datos que la caracterizan (identificador, descripción⁵, tipo de pregunta, respuesta/s correcta/s, etc.) y al responder y validar el estudiante la respuesta dada, los puntos conseguidos y la latencia producida (tiempo que ha tardado en responder la pregunta desde que se le presentó).

- Permite la inserción de varios *tests* en una misma página, cada uno de ellos con características diferentes en función de las opciones de configuración deseadas. Estos tests pueden generarse desde el código Java de un *applet* de VRL (invocando código JavaScript), desde una aplicación VRL JavaScript o desde el propio código JavaScript de un SCO (página Web). Sin embargo, la inserción del test debe producirse siempre en código HTML en la página Web del SCO. La **Figura 55** muestra un esquema donde se puede ver las tres posibilidades de generación de *tests* automáticos, los paquetes Java necesarios, la estructura de librerías JavaScript y las dependencias entre las mismas. En verde se muestran las librerías desarrolladas para la generación de *tests* interactivos automáticos con SCORM. El uso de estas librerías no limita el uso normal de la librería *RTE.js* mostrado en la **Figura 52**.

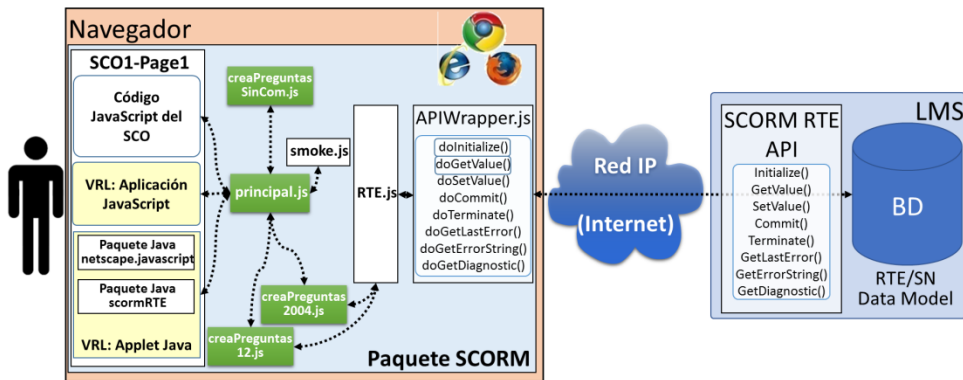


Figura 55. Esquema de comunicaciones y librerías usadas en la generación automatizada de tests

Funcionamiento y Ejemplo de uso. Para insertar un test interactivo con comunicaciones SCORM en una página web o una aplicación VRL de un SCO hay que hacer lo indicado en los siguientes pasos:

⁵ No disponible en SCORM 1.2

1. **Fichero QTI.** Se debe tener un fichero .xml en formato QTI de definición de preguntas que hay que incluir en el paquete SCORM. La forma más fácil de obtenerlo es exportarlo de un banco de preguntas de un LMS. La creación de preguntas así se ve facilitada enormemente. La mayoría de LMS (Moodle, ILIAS, etc.) utilizan el formato SCORM y permiten su exportación de forma directa. Otra opción posible es editar manualmente un fichero de texto para darle formato QTI e incluir dentro las preguntas deseadas (esta opción es muy laboriosa aunque no es complicada). En la **Figura 56** se puede ver un ejemplo de fichero QTI con 23 preguntas de *test*.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <!DOCTYPE questestinterop SYSTEM "ims_qtiasivlp2pl.dtd">
3 <!--Generated by ILIAS XmlWriter-->
4 <questestinterop>
5   <item ident="il 1946 qst 1476278" title="Señalización " maxattempts="1">
114   <item ident="il 1946 qst 1476284" title="Técnicas de Codificación" maxattempts="1">
223   <item ident="il 1946 qst 1476287" title="¿Quién inventó el teléfono?" maxattempts="1">
351   <item ident="il 1946 qst 1476288" title="¿Que servicio?" maxattempts="1">
479   <item ident="il 1946 qst 1476295" title="Marcación" maxattempts="1">
588   <item ident="il 1946 qst 1476298" title="P1-Temal-Sistema de marcación DTMF" maxattempts="1">
716   <item ident="il 1946 qst 1476308" title="Pregunta 5-Conector teléfono" maxattempts="1">
844   <item ident="il 1946 qst 1476277" title="Comutación" maxattempts="1">
1004   <item ident="il 1946 qst 1476283" title="Codecs" maxattempts="1">
1164   <item ident="il 1946 qst 1476286" title="¿De que tipo?" maxattempts="1">
1324   <item ident="il 1946 qst 1476297" title="Comutación (3)" maxattempts="1">
1457   <item ident="il 1946 qst 1476299" title="P3-Temal-Comutación" maxattempts="1">
1725   <item ident="il 1946 qst 1476302" title="P4-Temal-Características de técnicas de codificación
1885   <item ident="il 1946 qst 1476306" title="Pregunta 3-Redes de paquetes" maxattempts="1">
2045   <item ident="il 1946 qst 1476307" title="Pregunta 4-Redes de circuitos" maxattempts="1">
2205   <item ident="il 1946 qst 1476280" title="Comutación (2)" maxattempts="1">
2297   <item ident="il 1946 qst 1476281" title="Teorema de Nyquist" maxattempts="1">
2389   <item ident="il 1946 qst 1476293" title="Actualidad de la telefonía" maxattempts="1">
2461   <item ident="il 1946 qst 1476296" title="Teorema de Nyquist (3)" maxattempts="1">
2593   <item ident="il 1946 qst 1476301" title="P5-Temal-Señalización telefónica" maxattempts="1">
2713   <item ident="il 1946 qst 1476305" title="Pregunta 2 Marcación" maxattempts="1">
2812   <item ident="il 1946 qst 1476315" title="Trama MIC 1" maxattempts="1">
2982   <item ident="il 1946 qst 1476316" title="Tramas MIC" maxattempts="1">
3302 </questestinterop>

```

Figura 56. Ejemplo de fichero QTI de definición de preguntas de *test*

2. **Librerías JavaScript.** Se tienen que incluir en el paquete SCORM los siguientes ficheros: *principal.js*, *smoke.js*, *creaPreguntas2004.js*, *creaPreguntas12.js*, *creaPreguntasSinCom.js*, *RTE.js*. y *APIWrapper.js*, correspondientes a librerías JavaScript. Tienen que estar todas en la misma carpeta: “_ejs_library/scripts/”.
3. **Hojas de Estilo CSS y Edición CSS.** Se tienen que incluir en el paquete SCORM los siguientes ficheros: *estilo.css* y *smoke.css* correspondientes a hojas de estilo en cascada. Si se quiere modificar la apariencia de los *tests* que se quieran generar (colores, tipos de letra, formatos, etc.), se deben editar al gusto o necesidades para que su integración con el resto del WebLab sea mayor. Es recomendable que estén todos en la misma carpeta, por ejemplo “_ejs_library/CSS/”.

4. **Edición de SCO.** Se tiene que editar el fichero HTML del SCO en el que se quiere insertar el test para realizar las siguientes acciones:

a. **Importar librerías.** Hay que insertar código para importar las librerías JavaScript mínimas necesarias:

01: <script type="text/javascript" src="_ejs_library/scripts/APIWrapper.js"></script>

02: <script type="text/javascript" src="_ejs_library/scripts/RTE.js"></script>

03: <script type="text/javascript" src="_ejs_library/scripts/principal.js"></script>

04: <script type="text/javascript" src="_ejs_library/scripts/smoke.js "></script>

b. **Importar librerías.** Hay que insertar código para importar las hojas de estilo mínimas necesarias:

01: <link href="_ejs_library/css/estilo.css" rel="stylesheet" type="text/css">

02: <link href="_ejs_library/css/smoke.css" rel="stylesheet" type="text/css">

c. **Opciones del Test-Creación de variables.** En cada página Web del SCORM en el que se desee insertar un test es necesario añadir seis variables que sirven para definir sus características:

- **banco.** Esta variable hace referencia al fichero, incluyendo la ruta o path, que contiene las cuestiones de las que se compondrá el test. Es una cadena de caracteres.
- **numeroPr.** Indica el número de preguntas que debe contener el test. Es un entero.
- **time.** Sirve para indicar si no se quiere controlar el tiempo de que disponen los estudiantes para contestar el test (time=0) o si, por el contrario, se quiere limitar el tiempo entonces se debe indicar el número de minutos del que disponen los alumnos. Es un entero.
- **prAleat.** Indica si se va a aleatorizar las preguntas ("sí") o no ("no"). Es una cadena de caracteres con el valor "sí" o "no".
- **opAleat.** Indica si se va a aleatorizar las opciones de respuesta de las preguntas de opción múltiple ("sí") o no ("no"). Es una cadena de caracteres con el valor "sí" o "no".
- **comunicaciones.** Sirve para indicar si no se quiere utilizar las comunicaciones SCORM para que el LMS guarde registro de las preguntas ("sí") o, por el contrario, si se quieren usar comunicaciones SCORM. En este caso se debe indicar la versión deseada ("2004" o "12").

Las siguientes líneas muestran un ejemplo de asignación de estas variables:

```
01: var banco = '_ejs_library/qti/tema1.xml';
02: var numeroPr = 10;
03: var time = 0;
04: var prAleat = "si";
05: var opAleat = "si";
06: var comunicaciones = "2004";
```

- d. **Creación de tabla y llamada a Función *loadJsLib()***. La función *loadJsLib()* está incluida en la librería principal.js y es la función que se debe llamar dentro de código JavaScript (entre etiquetas <script>) cuando se quiera insertar un test en una página web de un SCO o una aplicación VRL. Tiene seis argumentos que permiten realizar la configuración del test conforme a los gustos del programador. Estos argumentos se corresponden con las seis variables creadas en el punto anterior. El código JavaScript de llamada a esta función debe incluirse dentro de una tabla con el identificador "tabla". Un ejemplo de la llamada a esta función se muestra a continuación.

```
01: <table id="tabla" align="center"><tr>
02:   <script type="text/javascript">;
03:   loadJsLib(banco,numeroPr,time,prAleat,opAleat,comunicaciones);
03:   </script>
04: </tr></table>
```

Una vez ejecutado un SCORM que cumpla lo indicado en los pasos 1 (Fichero QTI), 2 (Librerías JavaScript) y 3 (Hojas de Estilo CSS y Edición CSS), y se cargue un SCO editado según el punto 4 (Edición de SCO) ofrecerá un test como el mostrado en la **Figura 57**. El estilo podrá cambiar en función de la edición de las hojas de estilos realizada en el paso 3, es recomendable para adaptar visualmente el *test* al resto de contenidos del SCO y que se integre con normalidad.

The screenshot shows a test interface with a yellow background. At the top, it displays '9. P4-Tema1-Características de técnicas de codificación'. On the left, a 'Tiempo' (Time) box shows '06.52'. On the right, a 'Puntuación' (Score) box shows '47.75/55'. The question text is: 'Selección de entre las siguientes características cuáles son propias de una técnica de codificación vocoder?'. Below the text are four radio button options: 'Calidad regular' (checked), 'Mayor ancho de banda', 'Poco ancho de banda' (checked), and 'Buena calidad'. At the bottom of this section are three buttons: 'Validar', 'Borrar', and 'Posponer'.

The second question is titled '10. Tramas MIC'. The text reads: 'En una trama MIC E1 se produce un flujo de información a 2048 Kbps'. Below this, it says: 'Como la codificación de cada canal original (antes de multiplexarlo) se realiza usando 8 bytes por muestra, significa que el muestreo de canal se realiza cada 0.125 ms'. At the bottom, it states: 'Para obtener la información de una trama MIC E1 se debería hacer un muestreo cada 3.9 μs'. Similar to the first question, there are 'Validar', 'Borrar', and 'Posponer' buttons at the bottom.

Figura 57. Captura de pantalla de un segmento de test generado automáticamente

Se puede encontrar información más detallada sobre las librerías JavaScript para creación de test interactivos automatizados SCORM en [291].

4.5. Otras Utilidades

Hasta ahora se han mostrado las siguientes herramientas orientadas a la integración VRL-LMS:

- Paquete Java *scormRTE.jar*, que facilita las comunicaciones entre un VRL Java insertado como *applet* en un SCO de un paquete SCORM y el LMS que lo aloja.
- Librería *RTE.js*, que simplifica las comunicaciones entre el código JavaScript de un SCO de un paquete SCORM y el LMS que lo aloja.
- Librerías *principal.js*, *creaPreguntas2004.js*, *creaPreguntas12.js* y *creaPreguntasSinCom.js*, que generan *tests* interactivos en páginas web a partir de ficheros QTI con posibilidad de realizar comunicaciones basadas en SCORM con el LMS de forma automática.

También se han creado otras herramientas que, aunque no han resultado tan costosas de obtener, sí ofrecen una serie de utilidades indispensables que facilitan enormemente el desarrollo de WebLabs basados en SCORM con integración avanzada. Se enumeran a continuación y se explican con detalle en los sub-apartados siguientes:

Paquetes SCORM de Comprobación Sirven para comprobar el correcto funcionamiento de las comunicaciones SCORM-LMS de las versiones 1.2 y 2004 antes de empezar un nuevo desarrollo.

Plantillas de Paquetes SCORM Son paquetes SCORM sin contenido que presentan una estructura completa para ser rellenados con formatos de 1, 2, 3 y 4 páginas. Los paquetes se presentan para las versiones SCORM2004 y SCORM1.2.

Documentos de Ayuda Son documentos resumen y explicativos del paquete Java *scormRTE* y de la librería JavaScript *RTE.js*. Ayudan al programador de WebLabs a conocer las posibilidades de ambos.

4.5.1. Paquetes SCORM de Comprobación

No todos los LMSs implementan las versiones SCORM de la misma forma, existen algunos sub-elementos del modelo de datos RTE que pueden ser soportados o no a criterio del LMS. ADL [61], la agencia encargada del mantenimiento y control de SCORM distingue 3 categorías de productos SCORM de menor a mayor requerimiento:

- **SCORM conformant.** El producto debe superar una prueba con el Conjunto de Pruebas de Conformidad de ADL de la versión correspondiente.
- **SCORM adopter.** El producto debe haber sido considerado *SCORM conformant* previamente, se deben enviar a ADL los resultados de las pruebas con el Conjunto de Pruebas de Conformidad de ADL de la versión correspondiente y éstos deben ser positivos, pasan a incluirse en la lista de productos *SCORM adopter* [266].
- **SCORM certified.** Se trata de una verificación de la conformidad SCORM por una tercera parte independiente. También está basado en los resultados de las pruebas con el Conjunto de Pruebas de Conformidad de ADL de la versión correspondiente y los productos certificados pasan a formar parte de la lista *SCORM certified* [292].

Aunque un producto haya obtenido uno de estos grados no se puede asegurar que los cambios de versión en el LMS puedan haber producido algún cambio en el mismo. Por ello se han desarrollado dos paquetes SCORM de comprobación, uno para cada versión SCORM contemplada: SCORM2004 y SCORM1.2. Estos

paquetes utilizan la librería *RTE.js* para realizar comunicaciones con el LMS, consultar datos del modelo SCORM que sean de sólo lectura y grabar aquellos que son de escritura. Se trata de paquetes SCORM que contienen un único SCO muy simple:

El paquete “SCORM1.2 Test Page”. Permite comprobar las comunicaciones SCORM-LMS siguiendo la versión 1.2. Muestra una tabla en la que se muestran los datos del modelo:

- Constantes de sesión (datos que no varían durante toda la sesión de comunicación RTE), como el nombre e identificación del estudiante, la forma de entrar en el SCO, el máximo tiempo de trabajo permitido, el modo de visualización del SCO y los sub-elementos soportados por los elementos compuestos. La tabla se obtiene realizando una consulta de cada uno de estos datos, en la columna izquierda de la tabla aparece el nombre del dato y en la derecha el valor consultado en la base de datos del LMS. La **Figura 58** muestra una captura parcial de esta aparte de la página.

SCORM1.2 TEST PAGE

SCORM 1.2 Session Constants	
startTimeStamp:	11:47:54
learnerName:	Ruano Ruano, Ildefonso
learnerId:	1458

Figura 58. Captura de pantalla de las constantes de sesión en SCORM1.2

- Variables de sólo lectura de SCORM1.2. Son sólo 2: *interactionsCount* y *objectivesCount*, el número de interacciones y objetivos del SCO respectivamente, se pueden ver en la parte superior de la **Figura 59**.
- Variables de lectura/escritura de SCORM1.2. En este caso, en la columna izquierda de la tabla, se presenta un formulario con el que se puede rellenar los datos a grabar en el LMS y en la parte derecha se muestra el valor de la consulta. La parte inferior de la **Figura 59** muestra una captura parcial de estos datos.
- Comentarios de estudiante y de LMS, interacciones, objetivos, errores/diagnóstico y terminación de sesión. En la parte inferior de la tabla del SCO se muestra información de los datos relacionados con los elementos enumerados en este epígrafe, con los formularios se puede almacenar información en el LMS y recuperarla luego, los datos

relacionados con la terminación permiten establecer condiciones y el tiempo de la sesión y terminar la sesión de comunicación (**Figura 60**).

SCORM 1.2 Read only Variables	
Get Value interactionsCount: 0	
Get Value objectivesCount: 0	
SCORM 1.2 R/W Variables	
test de prueba de SCORM 1.2 Set Get value location: test de prueba de SCORM 1.2	
Datos para suspensión suspendData: Set Get value	Datos para suspensión
passed Set Get value completionStatus:	passed
10 Set Get Value	10
scoreRaw:	
0 Set Get Value	0
scoreMin:	

Figura 59. Captura de pantalla de las variables de sólo lectura y de lectura/escritura de SCORM1.2

Comments from Learner	
Comentario de usuario Set Get Value CommentFromLearner:	Comentario de usuario
Get Value CommentsFromLMS:	
Interactions	
Identificación1 true-false 10 true	
false wrong Set	
1	
Objectives	
identificacion 10 0 10 passed Set 0 Get Objective	
#.0. Id:identificacion. scoreRaw:10. scoreMin:0. scoreMax:10. Status:passed	
Error/Diagnostic	
Recupera error/Diagnóstico	
Error: 0 (No Error) Diagnostic: No Error	
Termination	
suspend Set ext Set session Time Terminar	

Figura 60. Comentarios, interacciones, Objetivos, Errores/Diagnóstico y Terminación de sesión

El paquete “SCORM2004 Test Page”. Permite comprobar las comunicaciones SCORM-LMS siguiendo la versión 2004. Muestra una tabla en la que se muestran los datos del modelo de forma similar a la mostrada para 1.2. Con la diferencia de los datos ampliados existentes en la nueva versión.

4.5.2. Plantillas de Paquetes SCORM

El propósito de estos paquetes es ofrecer una estructura SCORM a los programadores de WebLabs en la que se pueda empezar a trabajar para crear sus laboratorios *online* si tener muchos conocimientos del estándar. Existen

plantillas de paquetes SCORM para la versión 2004 (4) y para la versión 1.2 (4). Los hay de 1 a 4 SCOs (páginas Web). Los programadores de un WebLab que deseen usar SCORM y las herramientas explicadas en este capítulo simplemente tienen que elegir el formato SCORM deseado (2004 o 1.2), elegir la estructura de WebLab de 1 a 4 páginas (en función de los contenidos que deseen introducir en el WebLab) e introducir los contenidos en formato web (HTML, HTML5 o como deseen) en el contenido de las páginas Web principales de los SCOs. La **Figura 61** muestra los diferentes modelos que existe y la estructura de carpetas y directorios de cada una de las versiones SCORM.

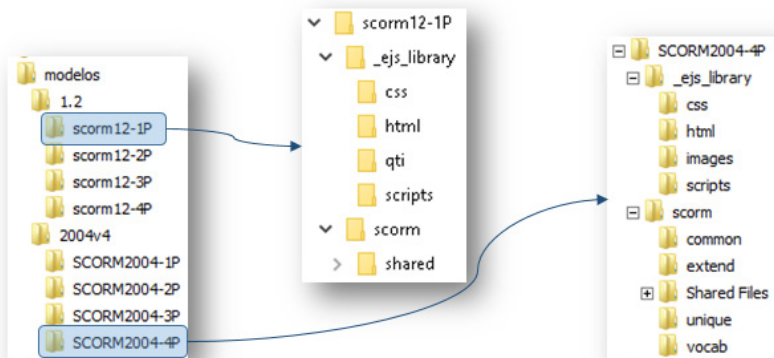


Figura 61. Estructuras de directorios de los modelos SCORM

El fichero *imsmanifest.xml* que define la estructura SCORM y todos los ficheros principales de los SCOs se sitúan en la carpeta principal de cada estructura.

Todos los paquetes incluyen las siguientes librerías y hojas de estilo, descritas en apartados anteriores: *APIWrapper.js*, *RTE.js*, *smoke.js*, *principal.js*, *creaPreguntas2004.js*, *creaPreguntas12.js* y *creaPreguntasSinCom.js.*, *estilo.css* y *smoke.css*. En la **Figura 62** se muestra un ejemplo de SCO en forma de HTML con las líneas de código necesarias para poder usar estos ficheros y asegurarse un funcionamiento mínimo y la estructura de directorios de los paquetes de la versión SCORM 1.2. El fichero *imsmanifest.xml* que define la estructura SCORM y todos los ficheros principales de los SCOs se sitúan en la carpeta principal que en el ejemplo de la **Figura 62** es la carpeta “scorm12-1p”.

A continuación se describe todo lo indicado en la **Figura 62**:

①: líneas de código necesarias para importar las librerías, que deben estar incluidas en la carpeta “_ejs_library/scripts”.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <title>Page1</title>
5 <script type="text/javascript" src="_ejs_library/scripts/APIWrapper.js"></script>
6 <script type="text/javascript" src="_ejs_library/scripts/RTE.js"></script>
7 <script type="text/javascript" src="_ejs_library/scripts/smoke.js"></script>
8 <script type="text/javascript" src="_ejs_library/scripts/principal.js"></script>
9 <link href="_ejs_library/css/estilo.css" rel="stylesheet" type="text/css">
10 <link href="_ejs_library/css/smoke.css" rel="stylesheet" type="text/css">
11 <script type="text/javascript">
12     var miRTE = new RTE("1.2");
13     var terminated = false;
14     var banco = '_ejs_library/qti/temal.xml';
15     var numeroPr = 10;
16     var time = 0;
17     var prAleat = 'si';
18     var opAleat = 'si';
19     var comunicaciones = '12';
20     function loadPage() {
21         var result = miRTE.initialize();
22         miRTE.getCompletionStatus();
23         if (miRTE.completionStatus == "not attempted"){
24             miRTE.setCompletionStatus("incomplete");
25         } else
26             if (miRTE.completionStatus == "passed"){
27                 terminated=true;
28             }
29     }
30     function unloadPage() {
31         if (terminated != true) {
32             terminated = true;
33             miRTE.setExit("");
34             miRTE.setCompletionStatus("passed");
35             miRTE.setSessionTime();
36             miRTE.terminate();
37         }
38     }
39 </script>
40 </head>
41 <body onLoad="loadPage()" onbeforeunload="unloadPage()">
42 <table align="center" width="80%"><tr>
43 <td align="center"><p align="center">Page 1</p></td>
44 <td align="center"><p align="center"><b>TITLE</b></p></td>
45 <td align="center"><p align="center">SubTitle</p></td></tr>
46 </table></br>
47 <table id="tabla" align="center"><tr>
48 <script type="text/javascript">
49     loadJsLib(banco, numeroPr, time, prAleat, opAleat, comunicaciones);
50 </script></tr>
51 </table></br>
52 </body>
53 </html>

```

Estructura de carpetas del paquete SCORM1.2

- ▼ scorm12-1P
 - ▼ _ejs_library
 - css
 - html
 - qti
 - scripts
 - ▼ scorm
 - > shared

Figura 62. Ejemplo de SCO y estructura de carpetas de paquete SCORM1.2

- 2: líneas de código necesarias para usar las hojas de estilo, que deben estar incluidas en la carpeta “_ejs_library/css”.
- 3: definición de la variable *miRTE*, de tipo *RTE* que se inicia con el argumento correspondiente a la versión SCORM (“1.2”). Este “objeto” es el que se usará para establecer comunicaciones con el LMS usando *RTE.js*.
- 4: definición de variables a utilizar si se desea crear un test automático interactivo usando la función *loadJsLib* (no incluida en el código). La ubicación del archivo XML QTI debe ser coherente con la estructura de directorios, en el ejemplo se ha supuesto que se encuentra en la carpeta “_ejs_library/qti”.
- 5: función JavaScript *loadPage()* que inicia una sesión de comunicación SCORM, obtiene del LMS el estado de terminación del SCO para el estudiante y en función de su valor actualiza otros datos del modelo. Esta función se ejecuta

al producirse el evento JavaScript “*onload*”, es decir, siempre que se carga la página Web. Sirve para asegurarse de que se inicia una sesión de comunicación SCORM al entrar en la página web.

6: función JavaScript *unloadPage()* que comprueba si se ha terminado la sesión de comunicación SCORM (para que funcione bien se debe actualizar la variable *terminated* si se termina la sesión desde otra parte del código) y en caso de no estar finalizada actualiza variables del modelo SCORM en el LMS y termina la sesión de comunicación. Esta función se ejecuta al producirse el evento JavaScript “*onbeforeunload*”, es decir, justo antes de descargar o salir de la página Web. Sirve para asegurarse de que no queda abierta la sesión de comunicación SCORM cuando se sale de la página web.

7: Tabla con *id=table* y función JavaScript *loadJsLib()* que inserta como contenido del HTML un test interactivo con la configuración definida en las variables de 4.

4.5.3. Documentos de Ayuda

Se han elaborado una serie de documentos PDF que sirven de ayuda para el conocimiento y uso de las herramientas de comunicación SCO-LMS, es decir, el paquete Java *scormRTE* y la librería *RTE.js*. Son los siguientes:

- **Tabla-resumen del paquete Java *scormRTE* (Figura 63).** Es una tabla que muestra las variables de las clases *scormRTE* y *scormRTE12* equivalentes a elementos del modelo de datos de SCORM, el tipo de dato con el que se ha implementado, las versiones SCORM para las que son válidas (SCORM1.2 y/o 2004) y el modo de acceso (lectura y/o escritura), en nombre oficial del elemento en SCORM, un comentario explicativo y los métodos disponibles para manipularlo (recuperar/almacenar del/en LMS).
- **Tablas resumen de librería JavaScript *RTE.js*.** Son varias tablas, algunas de las cuales muestran las variables de la librería mientras que otras muestran las funciones de la librería. Las siguientes figuras muestran ejemplos de ambos tipos de tablas:
 - En la **Figura 64**, se muestran las variables de la librería *RTE.js* ordenadas con diferentes criterios indicando si se corresponden con elementos del modelo SCORM, su compatibilidad en las

versiones 2004 y 1.2, los permisos de lectura (R) y escritura (W) en cada versión, si son constantes en la sesión de comunicación SCORM (C), si son *arrays* (A) y si son comunes a las 2 versiones (B).

scormRTE Packet		version		Dot Notation Binding		Comment
Name	Type	1.2	2004			
commentsLearner	String	RW		cmi.comments		Contains text from the learner.
	public String	rtGetCommentsLearner()				
	public String	rtSetCommentsLearner(String comment)				
commentsLearner	String[]	RW		cmi.comments_from_learner.n.comments		Array containing learner comments that are being stored by the LMS for the SCO.
	public String	rtGetCommentsLearner(int numcom)				
	public String	rtSetCommentsLearner(String comment, int numcom)				
commentsLearnerChildren	String	R		cmi.comments_from_learner.children		Indicates comments_from_learner elements supported by the LMS: comment, location, timestamp.
	public String	rtGetCommentsLearnerChildren()				
commentsLearnerCount	int	R		cmi.comments_from_learner.count		Indicates current number of learner comments that are being stored by the LMS for the SCO.
	public int	rtGetCommentsLearnerCount()				
						throws NumberFormatException
commentsLearnerLoc	String[]	RW		cmi.comments_from_learner.n.location		Indicates the point in the SCO to which the comment applies. If no value is specified for location, then the comment is applicable to the entire SCO.
	public String	rtGetCommentsLearnerLoc(int numcom)				
	public String	rtSetCommentsLearnerLoc(String location, int numcom)				
commentsLearnerTs	String[]	RW		cmi.comments_from_learner.n.timestamp		Indicates the point in time at which the comment was created or most recently changed. Format: YYYY-MM-DD[THH:mm:ss[.sTSO]]
	public String	rtGetCommentsLearnerTs(int numcom)				
	public String	rtSetCommentsLearnerTs(int numcom)				

Figura 63. Captura parcial de la Tabla-resumen del paquete Java scormRTE

RTE Variables	SCORM Data element		Permissions				C	A	B
	Version		2004		1.2				
	2004	1.2	R	W	R	W			
startDate	1	1	1		1				
startMS	1	1	1		1				
verScorm	1	1	1		1				
learnerName	cmi.learner_name	cmi.core.student_name	1		1		1	1	
learnerId	cmi.learner_id	cmi.core.student_id	1		1		1	1	
completionThreshold	cmi.completion_threshold	cmi.student_data.mastery_score	1		1		1	1	
credit	cmi.credit	cmi.core.credit	1		1		1	1	
entry	cmi.entry	cmi.core.entry	1		1		1	1	

Figura 64. Captura parcial de Tabla-Resumen de variables de librería RTE.js

- En la **Figura 65**, se detallan las funciones de la librería RTE.js agrupadas según uso básico (funciones auxiliares, básicas de SCORM, de recuperación de datos y de almacenamiento de datos) indicando su compatibilidad en las versiones 2004 y 1.2, si se corresponden con variables de RTE, si afectan a datos del modelo SCORM y si son exclusivas a alguna de las versiones.

Type	Function	2004	1.2	RTE variable	Both		Exclusive	
					Data	No D	2004	1.2
Aux	getWorkingSeconds(msec)	X	X		2			1
	getLatencyFromSeconds(secs)	X	X					1
Basic	initialize()	X	X					1
	terminate()	X	X					1
	commit()	X	X					1
	getLastError()	X	X					1
	getErrorString()	X	X					1
	getDiagnostic()	X	X					1
	getData(data)	X	X					1
	setData(data, value)	X	X					1
	getLearnerName()	X	X	learnerName				1
	getLearnerId()	X	X	learnerId				1
getCompletionThreshold()	X	X	completionThreshold				1	

Figura 65. Captura parcial de Tabla-resumen de funciones de librería RTE.js

- **Tabla-resumen de la librería JavaScript RTE.js (Figura 66)**. Es una tabla equivalente a la tabla mostrada en la **Figura 63** para el paquete

scormRTE, muestra las variables de la librería *RTE.js* equivalentes a elementos del modelo de datos de SCORM, el tipo de dato con el que se ha implementado, las versiones SCORM para las que son válidas (SCORM1.2 y/o 2004) y el modo de acceso (lectura y/o escritura), en nombre oficial del elemento en SCORM, un comentario explicativo y las funciones disponibles para manipularlo (recuperar/almacenar del/en LMS).

RTE JS function		version		Dot Notation Binding	Comment
Name	In charge	1.2	2004		
commentsLearner	SCO	RW	-	cmi.comments	Contains text from the learner.
				getCommentsLearner()	
				setCommentsLearner (comment)	
commentsLearner	SCO	-	RW	cmi.comments_from_learner.n.comments	Array containing learner comments that are being stored by the LMS for the SCO.
				getCommentsLearner (aux)	
				setCommentsLearner (comment, numcom)	
commentsLearnerChildren	LMS	-	R	cmi.comments_from_learner_children	Indicates comments_from_learner elements supported by the LMS: comment, location, timestamp.
				getCommentsLearnerChildren ()	
commentsLearnerCount	LMS	-	R	cmi.comments_from_learner_count	Indicates current number of learner comments that are being stored by the LMS for the SCO.
				getCommentsLearnerCount()	
commentsLearnerLoc	SCO	-	RW	cmi.comments_from_learner.n.location	Indicates the point in the SCO to which the comment applies. If no value is specified for location, then the comment is applicable to the entire SCO.
				getCommentsLearnerLoc (aux)	
				setCommentsLearnerLoc (location, numcom)	
commentsLearnerTs	SCO	-	RW	cmi.comments_from_learner.n.timestamp	Indicates the point in time at which the comment was created or most recently changed. Format: YYYY-MM-DD[Thh:mm[-ss[-t[STZ]]]]
				getCommentsLearnerTs (aux)	
				setCommentsLearnerTs (numcom)	

Figura 66. Captura parcial de Tabla-Resumen de variables y funciones de librería *RTE.js*

Cap.5

5. Metodología Desarrollada para la Creación de WebLabs

5.1. Introducción a la Metodología

5.2. Fases de la Metodología

5.3. Características de la Metodología

Los primeros laboratorios desarrollados en el ámbito de esta tesis fueron realizados siguiendo criterios propios a partir de la experiencia propia y los trabajos recopilados en las revisiones bibliográficas realizadas. Sin embargo, tras obtener los primeros prototipos de laboratorios *online* con integración avanzada

basada en estándares SCORM y se comprobó el proceso realizado hasta la obtención de los mismos, se vio la necesidad de revisar todo el proceso y tratar de generalizarlo. De este modo surgieron los contenidos del presente capítulo que incluye una metodología de creación de laboratorios online diseñada con el objetivo de obtener un producto, como es un WebLab, eficiente y que tenga todas las posibilidades de cumplir características siempre deseables en los laboratorios *online*.

5.1. Introducción a la Metodología

Esta metodología se ha desarrollado gracias a la experiencia obtenida en el diseño y desarrollo de laboratorios *online* y la aplicación de conceptos y teorías didácticas y pedagógicas. Se trata de una **metodología genérica que no impone ninguna tecnología, estándar ni arquitectura y puede ser aplicada para crear cualquier tipo de laboratorio**. Esta metodología puede ayudar al personal docente que no tenga mucha experiencia a entender cómo se puede realizar el diseño y la creación de los laboratorios *online*.

Un concepto clave que se ha tenido en cuenta a la hora de crear esta metodología es la efectividad del aprendizaje. Un laboratorio *online* es sólo una herramienta al servicio de las actividades de aprendizaje, por eso debería diseñarse a partir del plan de estudios, fijándose en el propósito final que se desea que tenga el laboratorio para que pueda cubrir alguna necesidad planteadas por el plan de estudios y teniendo en cuenta el resto de contenidos docentes [75]. Un laboratorio *online* no es un elemento aislado, debe estar integrado con el resto de recursos de aprendizaje y convertirse en un elemento más de un plan de aprendizaje en el que se presente la secuencia de aprendizaje programada por el equipo docente. Este proceso puede ser dividido en tres fases: Pre-VRL, VRL y post-VRL [69], [233]. En este sentido, las sesiones de laboratorio docente deben ser guiadas por un plan integrado de clases [271] que define una estructura adaptativa que incluya todos los recursos necesarios para que los estudiantes puedan conseguir las competencias y objetivos deseados.

5.2. Fases de la Metodología

Esta metodología se ha estructurado en ocho fases secuenciales (A-H), en cada una de las cuales participan una serie de actores para generar unos resultados. Estos resultados se deben entregar a los actores participantes de la siguiente fase para que puedan realizar sus labores correspondientes de forma que al finalizar las ocho fases se obtenga un laboratorio *online* con integración LMS que además sea efectivo para el aprendizaje del alumno. La **Figura 67** muestra un esquema de la metodología detallando las fases, actores y resultados parciales y en los sub-apartados siguientes se explican con detalle las ocho fases componentes de la metodología.

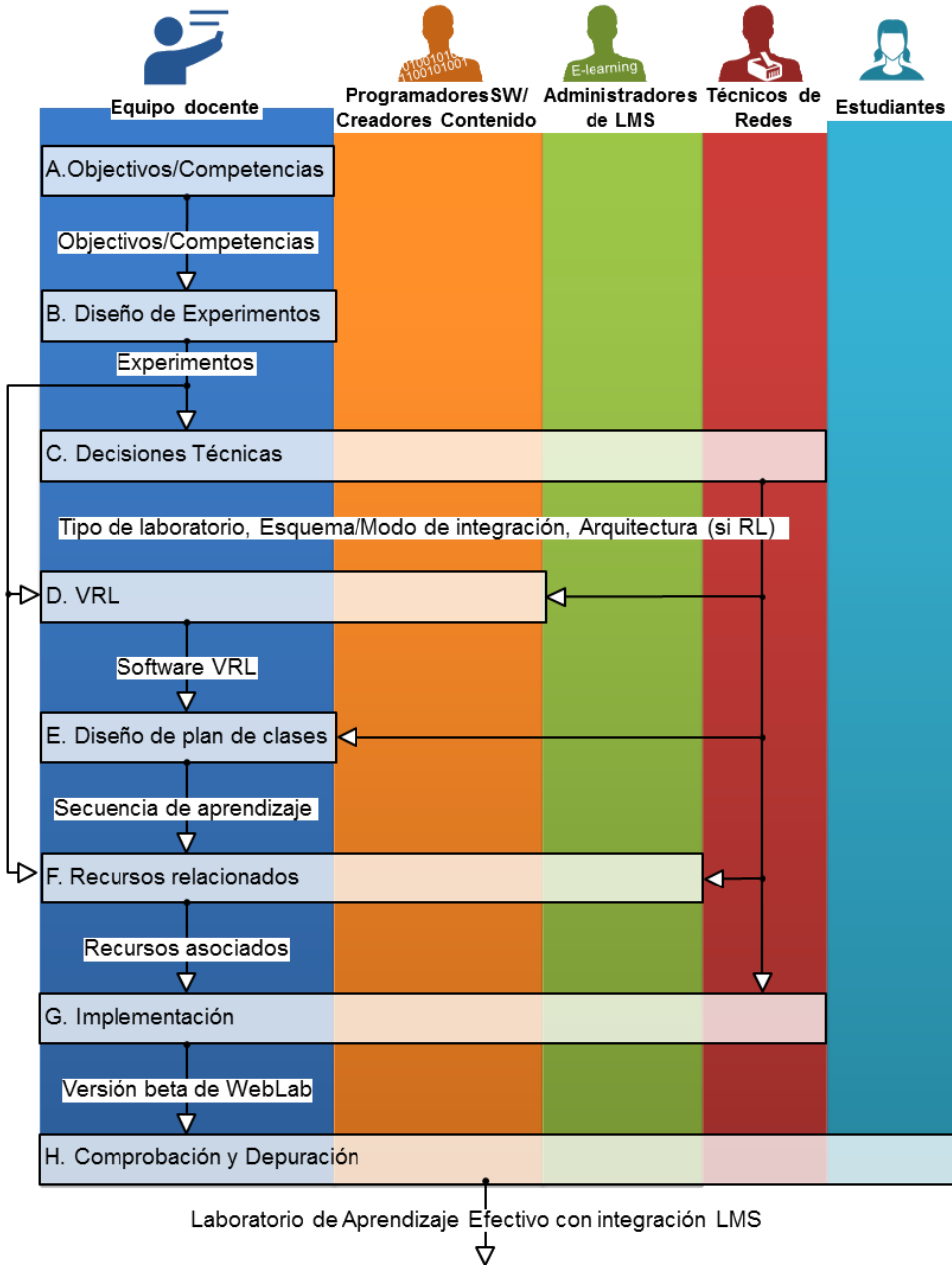


Figura 67. Metodología de creación de laboratorios online: Fases, actores y resultados obtenidos en cada fase

5.2.1. Fase A. Objetivos/Competencias

Esta fase está justificada por una obviedad señalada en [2] y repetida en [25]: *“If you don’t know where you want to go, you won’t know which road to take and you won’t know if you have arrived”* (“si no sabes dónde quieres ir, no sabrás que camino debes tomar ni cuando has llegado”). Aplicado a la educación sugiere que **la definición clara y concisa de los objetivos es esencial para diseñar un sistema de aprendizaje efectivo**. La existencia de un laboratorio docente está justificada por los experimentos que los estudiantes puedan realizar en los mismos, mientras que a su vez los experimentos están justificados por los objetivos y competencias que los estudiantes puedan conseguir al realizarlos. En los sistemas de educación actuales, los objetivos y competencias vienen definidos en el plan de estudios de la asignatura en la que se quiera encuadrar el laboratorio *online*. Por este motivo, el punto de partida para el diseño de un laboratorio docente es el plan de estudios de la asignatura.

La fase A debe **ser realizada por el equipo docente responsable** de la asignatura en la que se va a utilizar el laboratorio. Deben seleccionar del plan de estudios los objetivos que deben ser alcanzados mediante el trabajo práctico en un laboratorio y especificar claramente los objetivos de aprendizaje, habilidades y competencias que proporcionará el laboratorio a los estudiantes que lo utilicen y superen satisfactoriamente.

5.2.2. Fase B. Diseño de Experimentos

El **equipo docente**, a partir de la lista de objetivos y competencias generados en la fase anterior (fase A), **deberá diseñar los experimentos** con los que consideren que los estudiantes van a adquirir todos los objetivos y competencias especificadas. Estos experimentos y trabajos prácticos son los que deben ser proporcionados por el laboratorio *online* que debe ser creado en fases posteriores junto a los recursos y secuencias de trabajo relacionadas.

5.2.3. Fase C. Decisiones Técnicas

Para cada experimento, o grupo de experimentos similares que se hayan diseñado en la fase B, hay que **determinar el tipo de laboratorio** más apropiado (virtual, remoto o híbrido) **para implementar los experimentos y la arquitectura, tecnologías y estándares que se deben utilizar**. Esta decisión está

determinada muy fuertemente por la infraestructura tecnológica y la experiencia en desarrollos tecnológicos realizados en el pasado por el organismo o institución involucrado en la creación del laboratorio. La siguiente frase publicada en [41] se ajusta perfectamente con este hecho: “*Nobody wants to reinvent the wheel each time a device goes online*”, (“*Nadie quiere reinventar la rueda cada vez que tiene que poner un dispositivo en línea*”). Si la institución ya ha utilizado una determinada tecnología o arquitectura, un LMS particular, y/o ha desarrollado un determinado *plugin* específico para su LMS con el fin de integrar los laboratorios *online* con el LMS, es mejor utilizar esta propuesta, siempre que sea posible.

Cuando no existen estas limitaciones, se deben tomar una serie de decisiones:

1. **Elegir el tipo de laboratorio:** virtual, remoto o híbrido.
2. **Elegir las tecnologías y arquitectura.** En caso de que se tengan dispositivos físicos remotos estas decisiones son más complejas, entre otras cosas hay que determinar si se quieren usar soluciones ad hoc o basadas en sistemas estándares como son los *Web services*, Matlab o LabVIEW.
3. **Elegir el Modo de integración Laboratorio-LMS**, que puede ser básico (**Figura 37**) o avanzado (**Figura 38**). Esta metodología recomienda que se use un modo avanzado por las ventajas que conlleva. Además, siempre que se pueda es preferible utilizar un sistema que permita ubicar la aplicación VRL que debe utilizar el alumno en un LMS, ya que será más fácil la integración el mismo y el uso de las facilidades y recursos que aportan los LMS.
4. **Elegir el tipo de interacción:** *ad hoc* o basada en estándares.

En el caso más general, **estas decisiones deben ser tomadas por consenso entre el equipo docente, los programadores *software*, los técnicos de red y comunicaciones de la institución y los administradores del LMS** de la institución que implementará el laboratorio *online*. En la mayoría de los casos muchos de estos roles pueden recaer en una misma persona.

5.2.4. Fase D. VRL

El principal elemento de un laboratorio *online* es el *software* de laboratorio (VRL), entendiendo como tal la aplicación que deben utilizar los estudiantes

para llevar a cabo los experimentos programados. Existen algunas diferencias derivadas de los tipos de VRL que se pueden encontrar:

- **VL.** Cuando se trata de un laboratorio Virtual *online*, el *software* que utilizan los estudiantes debe ofrecer al menos una interfaz gráfica de usuario (GUI) con la que puedan interactuar con simulaciones de experimentos que deben completar.
- **RL.** En el caso de que el laboratorio online sea remoto el *software* se complica, ya que además de incluir una GUI, debe proporcionar servicios de comunicación con el equipo remoto en el que el alumno debe realizar los experimentos interactuando con ellos a través de la GUI. Hay que tener en cuenta que siempre que se realizan conexiones remotas se pueden producir problemas de seguridad que deberán ser tenidos en cuenta.
- **HL.** Si el VRL es un laboratorio *online* híbrido entonces el *software* incluirá, además de la GUI a través de la cual los estudiantes interactúan para realizar los experimentos, simulaciones y conexiones externas con un equipo real remoto.

Sea cual sea el tipo de *software* de VRL debe ser programado utilizando una tecnología compatible con la Web como JavaScript, que pueda ser incrustada en una página Web. Las tecnologías que requieren el uso de un *plugin* en el navegador Web, como los *applets* de Java o Flash, no son convenientes para el desarrollo de nuevos laboratorios, ya que las empresas responsables de los navegadores Web (como Internet Explorer, Google Chrome o Mozilla Firefox) han eliminado o anunciado fechas venideras para la eliminación del soporte de estas tecnologías [281], [282]. Existe la posibilidad de desarrollar *software* de VRL que no tenga que ser incrustado en una página Web y ser utilizado con un navegador Web, por ejemplo los basados en plataformas de móviles y tabletas; Sin embargo, si se desea obtener una integración avanzada con un LMS, se requiere una comunicación entre el VRL y el LMS, que debe producirse usando tecnologías Web y, preferiblemente, algún estándar como SCORM, LTI, xAPI, *cmi5* o similar.

A continuación se van a explicar algunas **propiedades convenientes y deseables** en los VRL para aumentar la efectividad del aprendizaje en los laboratorios online. Esta metodología aconseja a los creadores de laboratorios que implementen sus laboratorios teniendo en cuenta estas características:

- **Asistencia.** Para ayudar a los estudiantes a manipular correctamente los equipos de laboratorios y las simulaciones, todos los elementos de la GUI del *software* de VRL, deben tener textos de ayuda emergentes que indiquen su función.
- **Personalización-Accesibilidad.** El funcionamiento del *software* de VRL debe permitir una personalización basado en los gustos y necesidades del estudiante, al menos lenguaje, velocidad de presentación, subtítulos y volumen de audio. Para aumentar la accesibilidad, en los casos en los que sea aplicable, también puede considerarse la posibilidad de añadir audio descriptivo y cambiar tamaños de letra. Otra característica positiva, relacionada con este elemento, es la posibilidad de almacenar las opciones seleccionadas para poder ser recordadas en el futuro si se permiten accesos posteriores al laboratorio.
- **Experimentos individualizados.** Se refiere a la posibilidad de mostrar experimentos distintos a cada estudiante. Cada uno ejecutará un experimento personalizado cambiando los parámetros para cada ejecución o basado en el alumno y sus posibilidades.
- **Evaluación automática.** El trabajo realizado por cada estudiante en los experimentos es analizado y evaluado de forma automática por el propio *software* de VRL.
- **Registro del progreso de aprendizaje.** Consiste en el almacenamiento de las características de la sesión de laboratorio (hora de acceso, tiempo empleado, etc.), los logros alcanzados y los resultados obtenidos por cada alumno.

La integración avanzada del laboratorio online y el LMS puede facilitar enormemente la implementación de algunas de estas características. Cuando existe una comunicación LMS-VRL se puede identificar a los estudiantes, posibilitando, por tanto, la personalización del funcionamiento del VRL, los experimentos individualizados, auto-evaluaciones y almacenar los datos de uso y resultados obtenidos en el LMS por cada alumno. El tipo de interacción y el modo de integración lab-LMS viene determinado por las decisiones tomadas en la fase C (Decisiones Técnicas), y puede basarse en soluciones *ad hoc* o en el uso de estándares de *e-learning* como los comentados en el apartado anterior. Además, si el VRL está ubicado en el LMS (recomendado) se podrán establecer relaciones con otros recursos del LMS y crear secuencias de aprendizaje para facilitar la configuración del plan de aprendizaje que se presenta en la siguiente fase.

Los encargados de desarrollar el *SOFTWARE* VRL son principalmente los programadores *software* en estrecha colaboración con el equipo docente, aunque deben tener muy en cuenta las decisiones técnicas apuntadas por los administradores de LMS y los técnicos de red en caso de tratarse de un laboratorio remoto.

5.2.5. Fase E. Diseño del Plan de Aprendizaje

El *software* de VRL generado en la fase anterior no debe ser el único recurso que los estudiantes deberían usar. Para aumentar la efectividad del VRL deberían incluirse otros recursos creados y presentados junto al mismo. En esta fase el equipo docente debe determinar y diseñar un plan de aprendizaje formado por diferentes recursos junto a una secuencia de aprendizaje y la temporalización a través de los mismos, siempre a criterio de los docentes. Un camino de aprendizaje (o secuencia curricular) es un conjunto de etapas creadas para guiar a un estudiante para adquirir de forma efectiva conocimientos y habilidades. En cada etapa se presenta a los estudiantes un contenido con el que deben trabajar y tras cada una debería existir una evaluación que compruebe el progreso de aprendizaje del estudiante [293].

En este sentido se pueden implementar muchas teorías pedagógicas, como el ciclo de aprendizaje experimental de Kolb [71], que ha sido utilizado por diversos autores para diseñar las experiencias en laboratorios online [70], [218]. Kolb afirmaba que el conocimiento es resultado de la interacción entre la teoría y la experiencia, la experiencia se transforma en conocimiento mediante el proceso de aprendizaje. También afirmaba que se requiere una participación activa del estudiante en oposición con la participación pasiva asociada a la formación dirigida por el maestro y que el aprendizaje se lleva a cabo en las cuatro etapas mostradas en la **Figura 68**.

Independientemente del modelo pedagógico que quiera ser implementado por el equipo docente, la responsabilidad del equipo docente en esta fase es la de crear un plan de aprendizaje que se verá reflejado en una secuencia de aprendizaje (puede que reiterativa, como el ciclo de Kolb) formada por una serie de recursos entre los cuales se encuentra el VRL, las relaciones existentes entre dichos recursos y la navegación y secuenciación entre los mismos.

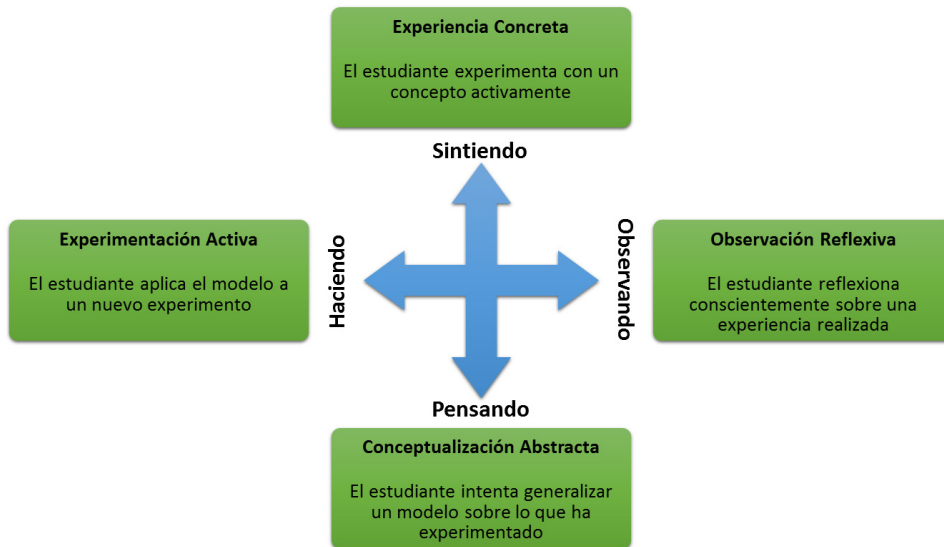


Figura 68. Ciclo de Aprendizaje Experimental de Kolb

Sin embargo, la creación del plan de aprendizaje viene determinada por las opciones de realización posibles, las cuales muchas veces pueden verse limitadas por las decisiones tomadas en fases anteriores (arquitectura, tipo de LMS, etc.). En el mejor de los casos el VRL está ubicado en el LMS junto a todos los recursos asociados y, de este modo, el plan de aprendizaje se puede establecer configurando la disponibilidad, temporización y pre-condiciones de acceso de los recursos que muy frecuentemente implementan los LMS. En todo caso, incluso si se va a ubicar el VRL en un LMS, hay que estudiar las capacidades que presenta el LMS.

En este sentido, SCORM está preparado para implementar secuencias de aprendizaje. Los paquetes SCORM pueden estructurarse en diferentes páginas e incluir diferentes tipos de recursos, como un VRL para el que se puede determinar la secuenciación y navegación entre los SCOs. Esto proporciona más flexibilidad a la hora de diseñar un plan de aprendizaje, al disponer de las siguientes opciones:

- Diseñar el WebLab como un único recurso en forma de paquete SCORM en el LMS, que incluya una secuencia de aprendizaje completa entendiendo como tal a un VRL y un conjunto de recursos asociados.
- Diseñar el WebLab en el LMS como una secuencia de aprendizaje formada por recursos del LMS entre los que se incluya un módulo

SCORM en el que, a su vez, está incrustado el VRL junto a otros recursos asociados.

- Diseñar el WebLab en el LMS como una secuencia de aprendizaje formada por recursos del LMS entre los que se incluya un módulo SCORM en el que únicamente se encuentra el VRL.

5.2.6. Fase F: Recursos Asociados

Como se ha comentado, los recursos asociados con el VRL, diseñados en la fase anterior (fase E), se pueden clasificar según su posición en la secuencia de aprendizaje como Pre-VRL, VRL o Post-VRL. A continuación se van a mostrar algunos recursos directamente relacionados con los VRL, que deberían presentarse junto a los mismos en cada una de las categorías justificando su existencia por la mejora en la efectividad que pueden producir:

Recursos Pre-VRL. Se trata de recursos que los estudiantes deberían trabajar antes de acceder al VRL. Los conocimientos previos influyen en su proceso de aprendizaje; este conocimiento previo además puede contener ideas falsas o estar mal construido en su memoria a largo plazo. Las actividades de aprendizaje pre-laboratorio aumentan la preparación del estudiante antes de que realicen el trabajo práctico y les ayudan a disponer de más capacidad de memoria de trabajo para el aprendizaje que realizan en el laboratorio. Algunos de los recursos que entran en esta fase son los siguientes:

- **Información General.** Los estudiantes trabajan mejor y menos estresados cuando son capaces de responder a las siguientes cuestiones antes de acceder al resto de recursos: ¿Qué voy a hacer?, ¿Cómo está relacionado este trabajo con la asignatura y mis estudios?, ¿Qué tipo de laboratorio voy a utilizar?, ¿Estoy capacitado para trabajar y entender este laboratorio?, ¿Qué propósito tiene este laboratorio?, ¿Qué voy a aprender?, ¿Qué habilidades y competencias voy a adquirir si trabajo en este laboratorio?, ¿Cómo se van a presentar estos contenidos?. El equipo docente debe dar respuesta a todas estas preguntas antes de presentar el VRL a los estudiantes. Así conseguirá darles confianza y una sensación de seguridad en sus capacidades. Para ello debe crear un recurso en el que se incluya esta información general sobre el laboratorio:
 - Enmarcar el VRL dentro del plan de estudios de la asignatura.

- Describir el laboratorio (tipo y estructura), los contenidos asociados y la navegación y secuenciación entre los mismos.
 - Mostrar los conocimientos previos recomendados para trabajar en el VRL que deben poseer para poder entenderlo.
 - Mostrar los objetivos y competencias/habilidades que van a adquirir si trabajan en el VRL y lo superan satisfactoriamente.
- **Conocimiento teórico previo.** Los laboratorios presentados a los estudiantes normalmente se basan en conocimientos teóricos que son explicados en las clases de teoría. Como se dice en [2] *“A common goal is to relate theory and practice or to bring the ‘real world’ into an otherwise theoretical education”* (*“Un objetivo común es relacionar la teoría y la práctica para llevar el ‘mundo real’ a una educación que, de lo contrario, sería solamente teórica”*). El tiempo limitado del que disponen los estudiantes en las sesiones pre-laboratorio no permiten que los profesores puedan clarificar los conceptos preconcebidos ni proporcionan la oportunidad de refrescar sus conocimientos haciendo las conexiones necesarias entre los conocimientos previos y los nuevos. Por eso, los estudiantes, normalmente, siguen las instrucciones del manual del laboratorio y los guiones de prácticas mecánicamente para realizar sus experimentos [294]. Los estudiantes que carecen de una sólida formación teórica no pueden comprender el objetivo del experimento que deben realizar ni entender los conceptos subyacentes [295]. Para evitarlo, es conveniente proporcionar las bases teóricas básicas, máxime cuando el laboratorio precisa de contenido teórico específico no explicado con la suficiente extensión o directamente no explicado anteriormente. En estos casos, es necesario añadir un recurso con la información teórica previa que el equipo docente considere indispensable para poder acometer los experimentos con éxito. Esta información puede extender la lección teórica del curso que el VRL está enriqueciendo o particularizar las ideas más generales dadas en las clases de teoría para el caso específico de los experimentos incluidos en el VRL.
- **Herramienta de evaluación Pre-VRL.** Los estudiantes entenderán mejor los experimentos del VRL y estarán más preparados para completar el laboratorio si han adquirido el conocimiento previo requerido para ello. Es necesario asegurarse de que los estudiantes poseen el conocimiento específico del VRL requerido para usar el VRL. El equipo docente debe diseñar un recurso que sea capaz de evaluar el conocimiento y las

capacidades de los estudiantes antes de acceder al VRL, en otras palabras, para acceder al lab debe darse una evaluación positiva [69]. Esta herramienta puede implementarse de muchas formas aunque la más común es en forma de test.

- **Requisitos de ejecución.** En algunos casos, es necesario explicar los requisitos necesarios para que el funcionamiento del *software* del VRL sea el adecuado. Por ejemplo, el uso de un determinado sistema operativo, navegador, *plugin*, *software* especial, configuración en el dispositivo o navegador, velocidad de la conexión u otras condiciones de la conexión como la apertura de puertos del nivel de transporte. La explicación de estas necesidades y las instrucciones necesarias para solventarlas pueden evitar frustraciones y enfados innecesarios a los estudiantes. En caso necesario, debe proporcionarse un documento que cumpla este cometido.
- **Manual del software.** El GUI de la aplicación de VRL permite interactuar con los elementos remotos o virtuales del laboratorio para realizar los experimentos. Debería proporcionarse un manual que explique su funcionamiento con detalle y guíe al estudiante en el manejo del VRL. Este recurso también debería estar disponible durante la realización de los experimentos. Este recurso, al igual que el anterior, debería crearse siguiendo las indicaciones de los programadores del *software* de VRL.

Recursos VRL. Son recursos que los estudiantes utilizan mientras trabajan con el VRL. Estos son algunos de ellos:

- **Guiones de prácticas.** Las guías de experimentación o guiones de prácticas, ayudan a los estudiantes a realizar los experimentos. Es un elemento esencial que explica los pasos que los estudiantes deben llevar a cabo para completar los experimentos. Consiste en una secuencia de instrucciones que describen sin ambigüedades el trabajo que deben realizar los estudiantes. El número de prácticas incluidas y el nivel de definición de las instrucciones deben ser determinados por el equipo docente teniendo en cuenta la posibilidad de incluirlos en el VRL.
- **Herramientas de Comunicación y Colaboración.** Las comunicaciones estudiante-estudiante y estudiante-tutor constituyen un elemento muy positivo en cualquier experiencia educativa [234], [232]. El aprendizaje es una actividad social y los estudiantes aprenden mejor cuando

interactúan frecuentemente con maestros y compañeros [84]. Un VRL, o su entorno, debería permitir que los estudiantes interactúen, proporcionando herramientas de comunicación síncronas y/o asíncronas. Los tutores deben ser capaces de resolver cualquier duda que pudiera surgir a los alumnos durante sus sesiones de trabajo con los laboratorios online. Por otro lado, si se pretende conseguir una cooperación entre estudiantes para la realización de unas prácticas es esencial disponer de herramientas de comunicación síncronas. Aunque estas herramientas pueden ser proporcionadas por el propio VRL, normalmente añaden una complejidad al desarrollo de la aplicación que puede ahorrarse si se utilizan las herramientas de comunicación que suelen proporcionar los LMS como son los foros, correo electrónico, chats, aulas virtuales y/o sesiones de videoconferencias. La ventaja de usar las herramientas de comunicación del LMS es que su uso se puede extender a las fases pre-VRL y post-VRL.

Recursos post-VRL. Son recursos que los estudiantes deben utilizar una vez hayan trabajado con el laboratorio. Entre estos se encuentran los siguientes:

- **Herramienta de evaluación post-VRL.** Una vez terminado el trabajo en el laboratorio online, es necesario verificar que los estudiantes han conseguido dominar los conceptos y han obtenido las habilidades deseadas, es decir, que se han conseguido los objetivos de aprendizaje buscados con el VRL. No se debe confundir con la evaluación que puede realizar el VRL, la cual debe estar diseñada para comprobar que se han realizado correctamente los experimentos. Un alumno puede realizar las prácticas correctamente y aun así no haber obtenido las competencias y objetivos para los que se diseñó el laboratorio. De este modo, esta herramienta constituye un elemento esencial a la hora de comprobar la efectividad del laboratorio en su conjunto, incluyendo todos los recursos proporcionados junto a la aplicación del VRL. Existen varias formas de implementar esta evaluación, mediante *tests*, entrevistas personales, encuestas o entrega de informes.
- **Solución recomendada.** Este recurso consiste en una descripción de una sesión de trabajo en el VRL en la que se lleva a cabo una ejecución correcta de los experimentos. No debe estar accesible hasta que haya finalizado el periodo de evaluación final del VRL. Tiene varios objetivos, al alumno que completó correctamente los experimentos le sirve para

recordar el trabajo realizado y clarificar ideas. A los que no lo terminaron correctamente les sirve para comprobar cómo debería haberse realizado el trabajo de laboratorio, entender y alcanzar los objetivos propuestos. Puede implementarse de varias formas aunque lo más adecuado es utilizar herramientas multimedia tipo *screencast* con un audio que describa las acciones realizadas puntualizando los hechos más relevantes y así ofrecerse después como un vídeo.

- **Recabar opinión de alumnado.** Sirve para conocer la opinión del alumnado sobre el laboratorio en su conjunto; el VRL y los recursos asociados. Los resultados permitirán conocer los puntos débiles y fuertes de la propuesta de cara a mejorar en el futuro y constituye otro elemento a tener en cuenta para “medir” la efectividad del laboratorio *online*. Puede implementarse como una encuesta de satisfacción en el propio LMS, entrevistas personales o entrega de formularios tradicionales.

Todos los recursos relacionados con el VRL que se acaban de describir deben crearse con un entorno y formato compatible entre sí, y además se tienen que implementar teniendo en cuenta criterios de usabilidad Web [296]. Lo más recomendable es que todos estos recursos sean creados por creadores de contenido especializados siguiendo los consejos y directrices dados por el personal docente. Lamentablemente, en la mayoría de los casos este trabajo es realizado por el propio personal docente que, muy frecuentemente, no posee los conocimientos necesarios para aplicar los criterios antes mencionados.

5.2.7. Fase G: Implementación

Todos los recursos asociados al VRL y la propia aplicación VRL deben instalarse y configurarse según la arquitectura y estándares elegidos, ya sea de forma distribuida o integrada en un LMS. Se recomienda el uso de estándares de *e-learning* para ello, con todas las ventajas que éstos aportan. También se deben establecer las relaciones de acceso entre recursos que puedan permitir la implementación de la/s secuencia/s de aprendizaje diseñada/s en el plan de aprendizaje. Esta fase depende mucho de las decisiones tomadas en las fases C (decisiones técnicas) y E (plan de aprendizaje). En esta fase se deben involucrar principalmente el personal docente, y en menor medida el administrador de LMS y los técnicos de redes y comunicaciones.

5.2.8. Fase H: Comprobación y Depuración

En esta fase debe comprobarse el correcto funcionamiento de todos los recursos, individualmente y en su conjunto, incluyendo la secuenciación y navegación entre ellos. En un principio el profesorado debe actuar como estudiantes probando todos los recursos incluidos y comprobando su perfecto funcionamiento, idealmente en todos los casos posibles, aunque, según el experimento, puede ser inviable debido a las múltiples posibilidades que se pueden dar. Adicionalmente se puede involucrar a ciertos estudiantes para que participen en la comprobación de las últimas versiones. Ante cualquier fallo se deberá realizar el arreglo necesario que podrá necesitar la ayuda de cualquiera de los roles de la *Figura 67*.

5.3. Características de la Metodología

La validación de la metodología descrita en este capítulo es un proceso complicado ya que depende de la implementación particular que se realice para cualquier laboratorio *online*. El método utilizado en este trabajo se basa en un conjunto de propiedades deseables que debería tener un laboratorio desarrollado utilizando la metodología. Se han analizado algunas de estas propiedades para comprobar cuando se puede asegurar que están presentes en un WebLab desarrollado con esta metodología:

- **Laboratorio multi-tipo.** Esta metodología no impone ningún tipo de restricción al tipo de laboratorio, puede ser aplicada para crear un laboratorio de cualquier tipo: remoto, virtual o híbrido.
- **Acceso a distancia Universal.** La metodología propuesta permite el desarrollo de laboratorios accesibles desde cualquier localización a través de Internet. No existen límites físicos.
- **Efectividad de aprendizaje.** Los principios en los que se basa la metodología se basan en factores que ayudan a la consecución de la efectividad del aprendizaje en el laboratorio. Las ocho fases en las que se estructura aseguran que los experimentos son diseñados con un objetivo principal: que los estudiantes logren los objetivos y obtengan las competencias específicas para los que se diseña el laboratorio. Siguiendo los principios dados en la metodología, el VRL se presenta a los estudiantes junto a una serie de recursos pedagógicos relacionados

con el laboratorio para cumplir un plan de aprendizaje que también tiene como objetivo aumentar la efectividad de la propuesta pedagógica.

- **Integración VRL-LMS.** La metodología apuesta por una integración avanzada VRL-LMS basada en el intercambio de información como es la identificación del estudiante entre otras.
- **Entorno conocido.** La recomendación de ubicar el WebLab en el LMS supone presentarlo en un entorno de trabajo conocido en el que los actores principales de la propuesta pedagógica; profesores, tutores y estudiantes, estarán más cómodos y relajados a la hora de realizar sus respectivas funciones facilitándoles su trabajo.
- **Estandarización.** Esta metodología no impone la utilización de ningún estándar pero recomienda el uso de estándares de *e-learning* para la integración VRL-LMS y la creación de materiales y que para la implementación de los laboratorios *online* se usen sistemas y estándares consolidados.
- **Personalización y Accesibilidad.** La identificación de los estudiantes permite la personalización del funcionamiento de los contenidos, especialmente de la aplicación de VRL, permitiendo su ejecución conforme a los gustos y posibilidades de acceso de cada persona.
- **Prácticas adaptadas.** De nuevo la identificación del estudiante permite realizar itinerarios de aprendizaje adaptadas a cada persona y experimentos caracterizados por cada identificación en función del trabajo que vaya realizando cada alumno. Se podrán ofrecer prácticas personalizadas.
- **Colaborativo.** Tanto si se implementan herramientas de comunicación para el trabajo conjunto de los estudiantes en la aplicación del VRL, como si se utilizan otras herramientas proporcionadas por el LMS, se puede conseguir esta propiedad que ha demostrado ser muy apreciada por proporcionar resultados muy positivos en los laboratorios donde se ha conseguido.
- **Evaluación automática.** La metodología recomienda que se programe el VRL para que realice autoevaluaciones automatizadas. Si se ubica en un LMS y los recursos asociados se implementan como recursos del LMS, entonces lo más probable es que el LMS pueda realizar evaluaciones automatizadas del uso y resultados obtenidos en esos recursos. Y si se utiliza el estándar SCORM también se pueden realizar evaluaciones

automatizadas del uso y resultados obtenidos en todos los recursos incluidos en el mismo, incluido el VRL.

- **Registro del progreso de aprendizaje automatizado.** La monitorización automatizada del trabajo realizado por los estudiantes es una propiedad deseable por los tutores y estudiantes, no sólo para los experimentos del laboratorio, sino para todos los recursos asociados. Al equipo docente le ahorra tiempo y trabajo de recopilación de datos y a los alumnos les permite obtener los resultados de su trabajo con el fin de mejorar y ahorrarse el estrés de las esperas. Si se siguen las directrices y recomendaciones de la metodología para la programación del VRL (Fase D), el diseño del plan de aprendizaje (Fase E) y la implementación de los recursos asociados (Fase G) en el LMS, entonces el progreso de aprendizaje puede ser registrado fácilmente de forma automática por el LMS.
- **Compartible y Reutilizable.** El desarrollo y puesta en marcha de un laboratorio online es bastante costoso, por ese motivo esta propiedad es muy deseable. Los laboratorios creados usando esta metodología que siguen las recomendaciones de uso de estándares y sistemas consolidados son más fáciles de compartir y reutilizar. En el caso concreto de utilizar el estándar SCORM, existe un gran número de sistemas LMS compatibles [266], [292], en los que la reutilización de un laboratorio *online* SCORM es un proceso trivial e inmediato.

El uso de la metodología propuesta asegura que los laboratorios online desarrollados cumplan directamente muchas de las propiedades que se acaban de describir y facilitan la consecución de todas ellas.

Cap.6

6. Aplicaciones y Laboratorios Desarrollados

6.1. Aplicaciones EJS

6.2. Laboratorios Virtuales

6.3. Laboratorios Remotos

6.4. Laboratorios Híbridos

Este capítulo muestra aplicaciones y laboratorios *online* que se han desarrollado en la Universidad de Jaén con el objetivo final de obtener una serie de laboratorios operativos para la docencia reglada en las Áreas de Telecomunicaciones y el Área de Ingeniería de Sistemas y Automática. En el

primer apartado (6.1) se van a mostrar algunas aplicaciones desarrolladas con el software *Easy Java Simulations* [53] que fueron orientadas a obtener destreza con el mismo para poder crear las simulaciones de sistemas reales que se han incluido en los WebLabs. Los siguientes apartados muestran los laboratorios *online* creados en la UJA, están estructurados por el tipo de laboratorio: virtuales (6.2), remotos (6.3) e híbridos (6.4). En cada uno de ellos se presentan distintos laboratorios cuyo objetivo final ha sido muy diverso. Los primeros que se obtuvieron han servido para comprobar el uso de SCORM como soporte para la presentación de laboratorios online, los siguientes han servido como prototipos que se han utilizado en docencia reglada en asignaturas impartidas en la Escuela Politécnica Superior de Jaén [297] (EPSJ) y la Escuela Politécnica Superior de Linares [298] (EPSL); y los últimos se han desarrollado siguiendo la metodología mostrada en el capítulo anterior, algunos de los cuales han podido ser utilizados también en la docencia reglada de la EPSJ y EPSL, obteniéndose interesantes datos de uso.

6.1. Aplicaciones EJS

Se trata de simulaciones realizadas en el lenguaje de programación Java con la herramienta EJS de sistemas reales existente en el Laboratorio de Control de Procesos situado en la EPSJ.

6.1.1. Simulación de Sistema Intercambiador de Calor (SIC)

En el Laboratorio de Control de Procesos de la EPSJ se encuentra la plataforma de pruebas mostrada en la **Figura 69**. Estos sistemas son muy comunes en los procesos industriales, especialmente de la industria alimenticia, por ejemplo en el proceso de extracción del aceite de oliva. Se realizó un modelo que fue utilizado para realizar una simulación del sistema con EJS.

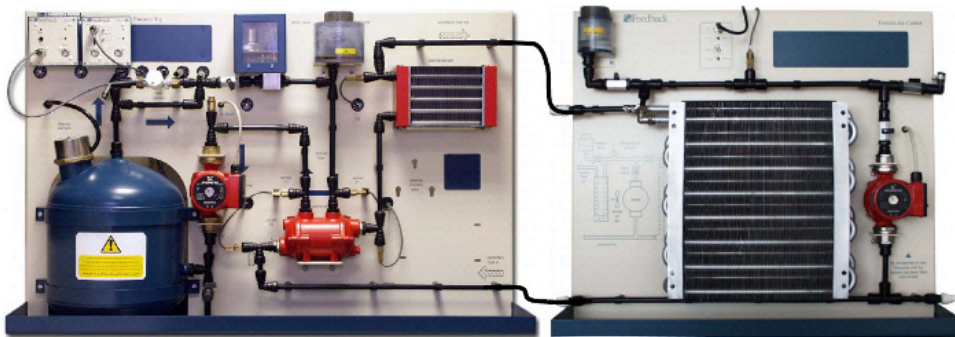


Figura 69. Sistema Intercambiador de Calor (SIC)

El sistema real es la combinación de dos productos de la empresa *Feedback-Instrument* [299]: el sistema de formación de procesos de temperatura completo (cod. 38-002) y el refrigerador de aire forzado (cod. 38-610), aunque no se ha tenido en cuenta el segundo para la simulación. El sistema está formado por dos circuitos formados por tubos rellenos de agua conectados a través de un intercambiador de calor a contracorriente (circuito primario, en la izquierda de la **Figura 69**). El primer circuito (primario) contiene una resistencia eléctrica que calienta el agua; mientras que, el segundo circuito (secundario) tiene un radiador de aire que enfría el fluido. El esquema del proceso termal se muestra en la **Figura 70**.

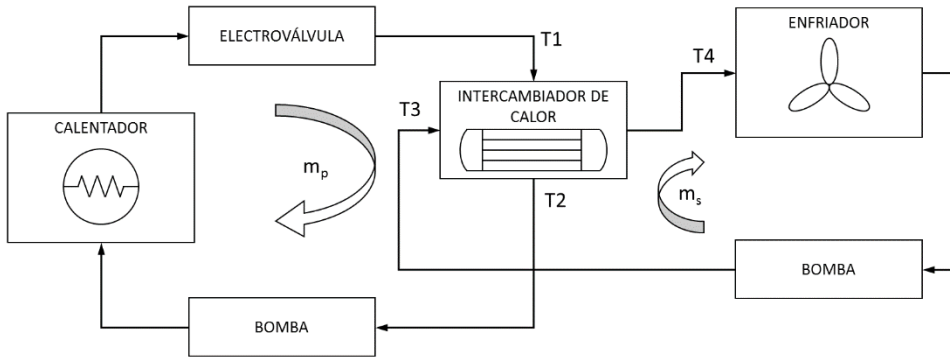


Figura 70. Esquema del proceso termal

Las ecuaciones dinámicas que definen el comportamiento del proceso son las mostradas en las ecuaciones (ec.1-ec.4, el significado de las variables está en la **Tabla 7**):

$$m_p c_p \dot{T}_1 = \dot{m}_p c_p (T_2 - T_1) + \dot{Q}_c \quad (\text{ec.1})$$

$$m_{ip} c_p \dot{T}_2 = \dot{m}_p c_p (T_1 - T_2) - \dot{Q}_i \quad (\text{ec.2})$$

$$m_{is} c_p \dot{T}_4 = \dot{m}_s c_p (T_3 - T_4) + \dot{Q}_i \quad (\text{ec.3})$$

$$m_s c_p \dot{T}_3 = \dot{m}_s c_p (T_4 - T_3) + \dot{Q}_v \quad (\text{ec.4})$$

Tabla 7. Lista de variables del sistema

Variable	Definición
T_1	Temperatura de entrada del circuito primario del SIC.
T_2	Temperatura de salida del circuito primario del SIC.
T_3	Temperatura de entrada del circuito secundario del SIC.
T_4	Temperatura de salida del enfriador.
T_{env}	Temperatura del entorno.
m_{pr}, m_s	Capacitancia (Líquido en circuito 1).
m_{ipr}, m_{is}	Masas de agua en circuitos primario y secundario del SIC.
c_p	Calor específico del agua.
\dot{Q}_i	Tasa de flujo de calor del intercambiador.
\dot{Q}_c	Transferencia de calor (Calentador).
\dot{Q}_v	Transferencia de calor disipada del intercambiador.
AU_i	Coefficiente de transferencia de calor del intercambiador.
AU_r	Coefficiente de transferencia de calor del enfriador.

El calor transferido en el intercambiador se define por la ecuación característica de un intercambiador de calor de carcasa y tubo, que es la mostrada en las ecuaciones (ec.5 y ec.6).

$$\dot{Q}_i = AU_i \Delta T_{lm} \quad (\text{ec.5})$$

$$\Delta T_{lm} = \frac{(T_1 - T_4) - (T_2 - T_3)}{\ln \frac{(T_1 - T_4)}{(T_2 - T_3)}} \quad (\text{ec.6})$$

Y la ecuación (ec.7) muestra la transferencia de calor disipada del enfriador:

$$\dot{Q}_v = AU_r (u_{speed})(T_4 - T_{env}) \quad (\text{ec.7})$$

Donde las señales de actuación son la tasa de flujo del circuito primario y la velocidad del enfriador del circuito secundario (u_{speed}).

Se creó una simulación con la herramienta EJS, en la que se implementó el modelo del SIC totalmente operativa. La GUI presenta cuatro zonas diferenciadas como se puede observar en la **Figura 71**:

- *Zona superior*: aparece un panel con una botonera de control de la puesta en marcha y funcionamiento de la simulación (ejecutar, pausar, ejecutar paso a paso, resetear e inicializar) y mostraba la temperatura ambiente (T_{env}) y las de las distintas partes del SIC (T_1 a T_4).
- *Zona media-alta*: muestra una imagen fija del SIC real sobre la cual se han solapado elementos representativos de las temperaturas existentes en las distintas partes del SIC (T_1 a T_4). Estos elementos cambian de color y de tamaño en función del valor de la temperatura correspondiente.
- *Zona media-baja*: contiene un panel de control con los actuadores del sistema, el calentador y el enfriador, pudiendo seleccionar su activación y el valor de su constante representativa, que puede introducirse escribiendo su valor en el campo específico o deslizando el controlador correspondiente.
- *Zona inferior*: muestra en gráficas la evolución de las 4 temperaturas en función del tiempo.

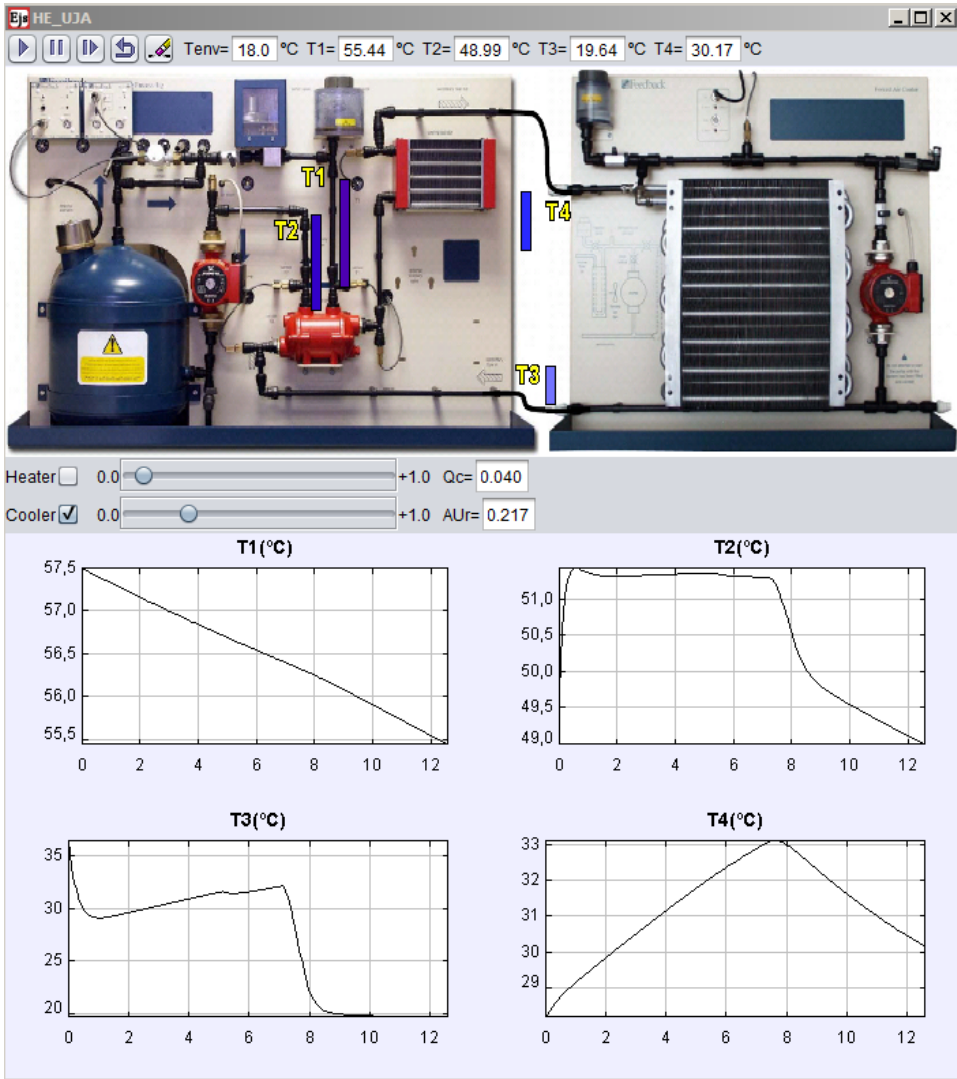


Figura 71. GUI de la simulación EJS del SIC

6.1.2. Simulación de Sistema de Péndulo Invertido en Carril (SPIC)

El sistema de carro-péndulo invertido en carril lineal consiste en un carro que se mueve a lo largo de un carril de un metro de largo (**Figura 72**). El carro tiene un eje en el que están enlazados dos péndulos que son capaces de girar libremente. El carro se puede mover hacia atrás y hacia adelante provocando el giro de los péndulos. El movimiento del carro se produce por el desplazamiento de la cinta en dos direcciones causado por el motor DC situado al final del carril. Al aplicar una tensión en el motor se controla la fuerza con la que el carro se

mueve. El valor de la fuerza depende del valor de la tensión de control. La tensión es, por tanto, la variable manipulada por el controlador.

En el sistema real las dos variables que se leen desde el péndulo (usando codificadores digitales) son la posición del péndulo (ángulo) y la posición del carro en el carril. La tarea del controlador debe cambiar la tensión del motor DC dependiendo de estas dos variables, de forma que se complete la tarea de control deseada (control de estabilización de los péndulos en una posición vertical superior, control de girado o control del carro).

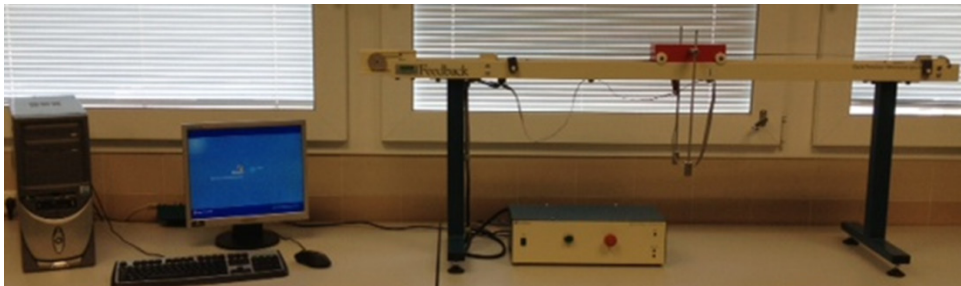


Figura 72. Sistema de Péndulo Invertido en Carro Lineal (SPIC)

El proceso mecánico del péndulo es el presentado en la **Figura 73**, que también incluye las ecuaciones de movimiento no lineales, que se obtienen sumando las fuerzas que actúan en el sistema del péndulo, el carro y los momentos (ec.8 y ec.9), y la ecuación de la fuerza (ec.10).

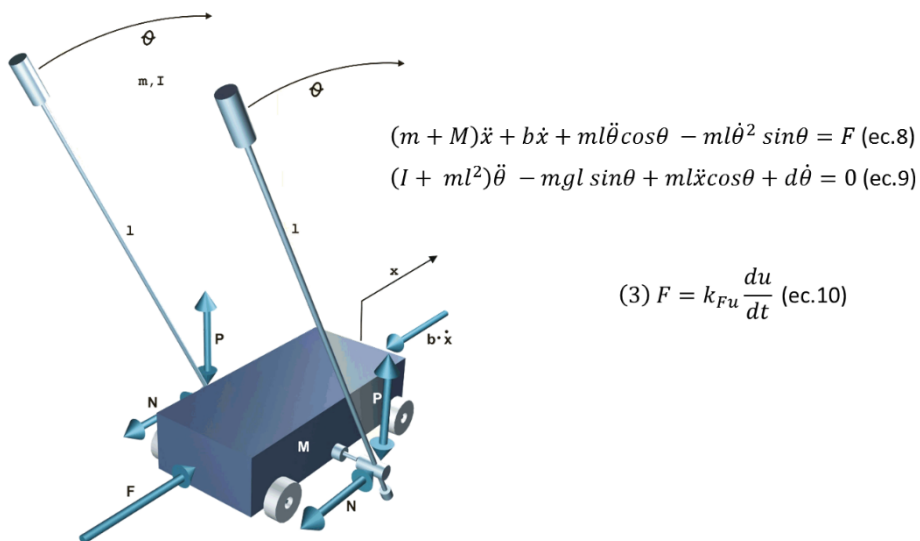


Figura 73. Detalle del carro con péndulos, variables, constantes y ecuaciones del SPIC

El significado y valor de los parámetros constantes se describe en la **Tabla 8**.

Tabla 8. Variables y constantes del SPIC

Parámetro	Valor
g: Gravedad.	9,81 m/s ²
l: longitud del palo del péndulo.	0,36-0,4 m (depende de configuración)
M: masa del carro.	2,4 Kg
I: Momento de inercia del carro.	~0,099 Kg/m ² (depende de configuración)
m: masa del palo (del péndulo).	0,23 Kg
d: Coeficiente de amortiguamiento del péndulo.	0,005 Nms/rad
b: Coeficiente de fricción del carro.	0,05 Ns/m
F: Fuerza ejercida en carro.	
x: Posición del carro.	
u: Señal de control (tensión).	
θ: Ángulo del péndulo.	
k_{fu}: Ganancia entre la derivada de la tensión de entrada y la fuerza	

Se creó una simulación con la herramienta EJS, en la que se implementó el modelo del SPIC totalmente operativa. La GUI presenta tres ventanas, la principal (Modelo SISPEN, **Figura 74**) donde se muestra la imagen del sistema y el panel de control de la simulación desde el que se puede controlar la visualización de las otras 2 ventanas (“Gráficos” y “Modelos”), descritas a continuación.



Figura 74. Ventana principal de la GUI de la simulación del SPIC

La ventana “Gráficos” (**Figura 75**) muestra gráficos de la posición, velocidad y aceleración del carro y del ángulo, velocidad angular y aceleración angular del péndulo.

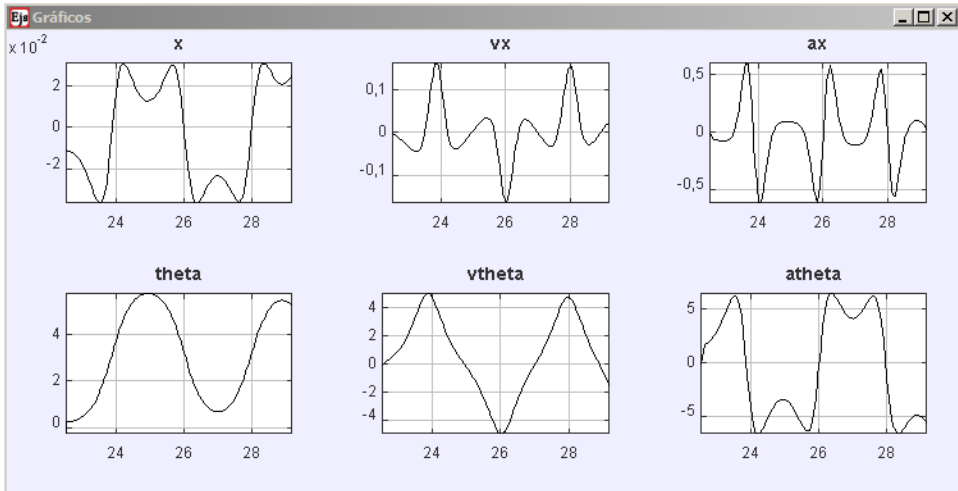


Figura 75. Ventana de gráficos del SPICL

La ventana “Modelo” (**Figura 76**) muestra el valor de los parámetros constantes y permite cambiar su valor.

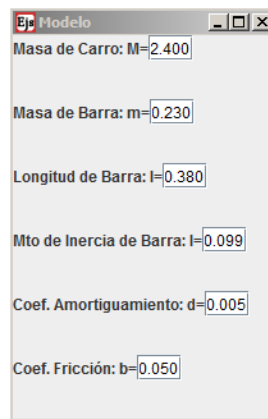


Figura 76. Ventana de parámetros del SPICL

6.2. Laboratorios Virtuales

En este apartado se muestran laboratorios virtuales desarrollados con integración en LMS mediante su incrustación en paquetes SCORM y el uso de las

herramientas y utilidades descritas en el capítulo “4. Herramientas y Utilidades Desarrolladas”.

6.2.1. Sistema de Suspensión de la rueda de un coche

Este laboratorio fue el primero creado como paquete SCORM 2004 integrado en un LMS, la plataforma institucional ILIAS de la UJA. El VL realiza la simulación del funcionamiento del amortiguador para la rueda de un automóvil con la posibilidad de trabajar en distintos escenarios. Este laboratorio virtual se diseñó para ser utilizado en la asignatura Automática Industrial que se imparte en el segundo curso de todos los grados de la rama de Ingeniería industrial ofertados en la Universidad de Jaén (ingeniería eléctrica, electrónica industrial, mecánica y organización industrial). Dicha asignatura tiene como finalidad general proporcionar al alumno una visión global de la aplicabilidad de la automática en entornos industriales y la ofrecer los conocimientos básicos sobre las diferentes tecnologías aplicadas al control de procesos continuos y discretos. Esta práctica se llevaría a cabo en el módulo de identificación y caracterización de sistemas dinámicos. Tiene como objetivos principales:

- Afianzar en el alumno el concepto de sistema dinámico.
- Diferenciar y extraer las características fundamentales de las distintas respuestas temporales. Para ello se le da la opción de ajustar los parámetros del modelo dinámico que caracteriza el sistema.

Estructura del Paquete SCORM

Está formada por 3 SCOs en los que se incluye una presentación teórica del laboratorio (SCO1), los guiones de prácticas y una aplicación VL desarrollada con EJS (SCO2) y una evaluación final consistente en un test de 6 preguntas que se corrige automáticamente. La estructura del paquete SCORM se creó a partir de una plantilla descargada de la Web de ADL [61]. Los tres SCOs que forman el paquete SCORM utilizan el RTE de SCORM para comunicarse con el LMS a través de la librería JavaScript *APIWrapper.js* que usa la API SCORM, esto significa que se llaman a las funciones de *APIWrapper.js* desde código JavaScript incrustado en el código HTML de las páginas Web asociadas a cada SCO. Además se utilizaron métodos Java de desarrollo propio en el código del VL desarrollado

con EJS con el que se invocan las funciones de APIWrapper.js. La **Figura 77** muestra esta estructura y las comunicaciones SCORM establecidas entre el WebLab y el LMS ILIAS.

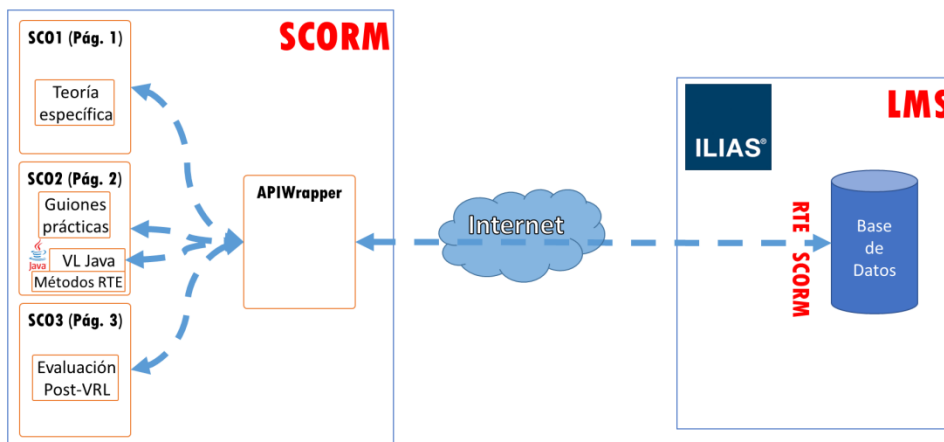


Figura 77. Estructura de Lab “Sistema de suspensión de la rueda de un coche” y comunicaciones LMS

El **SCO1 (Figura 78)** muestra la teoría básica del sistema de suspensión, inicia una sesión de comunicación y ofrece un botón al final de la página para comprobar el tiempo pasado desde la carga de la página, cuando es superior a un minuto considera que el SCO está superado y almacena en el LMS el estado de superación, “passed”, el de terminación, “completed”, y la puntuación alcanzada, “5”, para el estudiante que se haya identificado.

SCORM -EJS

Modelo de masa simple para simular el Sistema de suspensión de la rueda de un coche

La siguiente figura muestra una representación del sistema de suspensión de un coche simplificado. K es la constante del muelle (de longitud en reposo L0) modela la elasticidad del neumático y del muelle de la suspensión. La constante de amortiguamiento B modela el amortiguador. Las masas de la rueda, del neumático y del eje son despreciadas, la masa m representa una cuarta parte de la masa del coche (1400 Kg / 4 = 350 Kg). La variable u representa la altura que presenta el asfalto respecto a una posición inicial tomada como sistema de referencia de todo el sistema y la variable y representa la altura que toma el vehículo respecto al mismo sistema.

Lo que se pretende con un sistema de amortiguación es que diferencia y-u evolucione para evitar movimientos verticales bruscos en la masa.

Para ello se debe obtener la ecuación del sistema aunque antes se debe entender el funcionamiento del amortiguador, el cual trata de eliminar las diferencias de velocidad que se producen en la masa (el vehículo) cuando el asfalto deja de ser plano y ofrece discontinuidades (subidas, bajadas, piedras, huecos, etc.). La fuerza que ejerce el amortiguador tiene una magnitud de valor B*(dy-du) y actúa en la dirección en la que reduce la diferencia de velocidad en sentido vertical.

La ecuación que gobierna el sistema es la siguiente:

$$m\ddot{y} = -mg - K(y - u - L_0) - B(\dot{y} - \dot{u})$$

Superación de esta parte

Para superar esta parte de teoría, debes estar al menos 5 minutos en esta página (300 segundos), además conseguirás los 5 puntos correspondientes a la parte teórica. Aprovecha el tiempo para leer los párrafos anteriores y comprender cual es el principio básico del sistema de suspensión que se va a trabajar en la página siguiente del módulo donde hay un laboratorio virtual basado en la simulación de un sistema de suspensión en el que podrás ver su funcionamiento y deberás realizar una serie de prácticas. Puedes comprobar el tiempo que llevas en esta página pulsando sobre el siguiente botón:

Comprobar Evaluación y Ver el tiempo pasado

Fecha y hora de entrada a SCO1: 2016-10-03T12:23:57
60.093 segundos (passed)

Figura 78. Captura de pantalla del SCO1 con la teoría específica del laboratorio

El **SCO2** contiene los guiones de prácticas que debe realizar el alumno (**Figura 79**) y el *applet* Java del VL en el que debe realizar las prácticas. En el primer experimento se le pide que experimente con diferentes escenarios (tipos de terreno o eventos en el mismo tipo bache o piedra) y observe lo que ocurre, la evaluación que realiza el laboratorio se basa en la comprobación de la ejecución de la simulación, añadiendo un punto por cada escenario probado. En el segundo se le pide que consiga una determinada configuración mediante la asignación de valores a la constante del muelle (K) y de amortiguamiento (B). En el caso de elegir unos valores dentro de los márgenes establecidos se consiguen 5 puntos adicionales por lo que el máximo sería 14 (9+5).

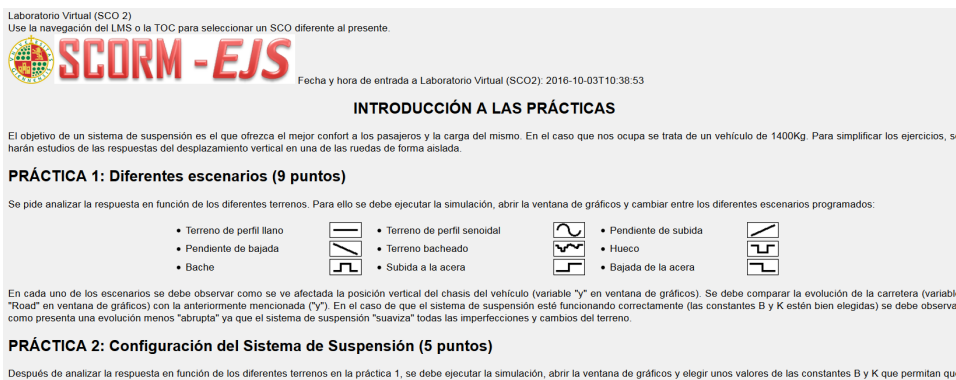


Figura 79. Captura de pantalla parcial de SCO2. Guiones de prácticas

VL. El laboratorio virtual desarrollado muestra el funcionamiento de un amortiguador de coche permitiendo al alumno modificar sus parámetros (K , constante del muelle y B , constante de amortiguamiento) y seleccionar distintos escenarios de funcionamiento. La **Figura 80** muestra una imagen del sistema real (a), el esquema del sistema de suspensión (b) y el esquema del sistema completo (c).

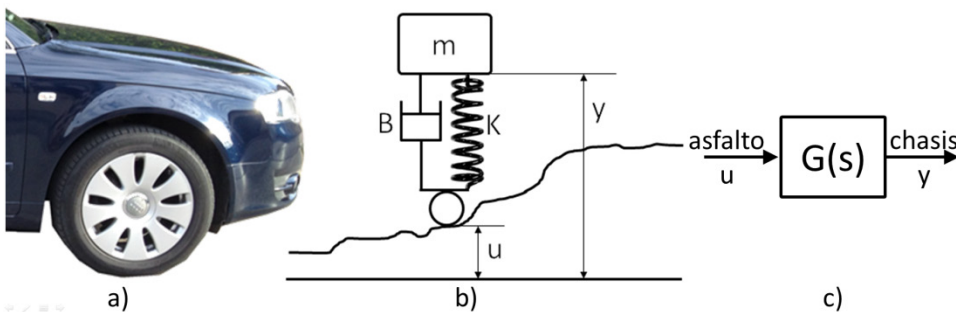


Figura 80. Sistema real, esquema del sistema de suspensión y esquema general del sistema

Se ha utilizado un ejemplo sencillo que sirva para mostrar el trabajo desarrollado de forma simple, la ecuación del sistema de suspensión (ec.1) y su función de transferencia (ec.12) se muestran a continuación, y en la **Tabla 9** el significado de las constantes y variables utilizadas.

$$m\ddot{y} = -mg - K(y - u - L0) - B(\dot{y} - \dot{u}) \quad (\text{ec.11})$$

$$G(s) = \frac{Y(s)}{U(s)} = \frac{Bs+K}{ms^2+Bs+K} \quad (\text{ec.12})$$

Tabla 9. Parámetros del Sistema de suspensión.

Parámetro	Significado
g	Aceleración de la gravedad (9,81 m/s ²)
L0	Longitud del muelle en reposo.
m	Masa que soporta la rueda (350 Kg).
K	Constante del muelle.~
B	Constante de amortiguamiento del muelle.
u	Distancia del asfalto al punto de referencia.
y	Distancia de la masa al punto de referencia.

La **Figura 81** muestra una captura de pantalla del *applet* EJS del VL incrustado en el SCO2.

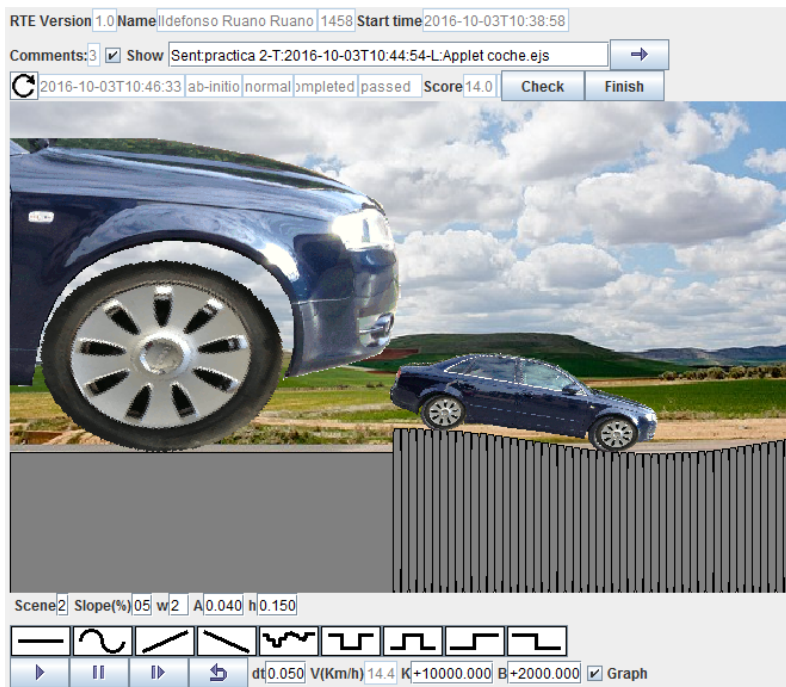


Figura 81. Captura de pantalla del *applet* Java del VL del sistema de suspensión de una rueda

En la parte superior se muestra un panel en el que se muestra información recogida por el *applet* del LMS en el momento de su lanzamiento (que puede ser actualizada en cualquier momento) y una zona en la que el alumno que esté trabajando puede guardar los comentarios que estime oportuno en el LMS para recuperarlos cuando desee (para ello debe abrir una ventana supletoria seleccionando la casilla de verificación desde el mismo panel). En la parte central se muestran una serie de imágenes que ayudan al alumno a reconocer la escena y el funcionamiento que se logra en el amortiguador del coche mientras que en la parte inferior aparecen tres paneles desde los que se pueden configurar las características del experimento:

- *Panel superior.* Los parámetros de las distintas escenas que se pueden establecer para el funcionamiento del amortiguador.
- *Panel medio.* Permite seleccionar los escenarios de uso del sistema.
- *Panel inferior.* Controla la puesta en marcha y desarrollo de la simulación así como cambiar los parámetros de configuración del sistema de amortiguación y la activación de una ventana de gráficas de resultados.

Para poder hacer esto se han utilizado métodos Java para interactuar con el LMS que permiten obtener y mostrar en el *applet* los siguientes datos:

- El nombre e identificador del alumno que accede al módulo SCORM.
- La versión de API que se está ejecutando (ofrecida por el LMS).
- El estado de terminación del laboratorio (si está superado).
- La fecha y hora de entrada en el laboratorio.
- Los comentarios guardados por el alumno sobre el laboratorio en el LMS.

Por otro lado también se han desarrollado otros métodos que permiten almacenar en el LMS información obtenida en el *applet* Java:

- Comentarios del alumno que permanecen de forma persistente en el LMS para su consulta en sesiones e intentos posteriores por parte del laboratorio online.
- Información referente a la puntuación alcanzada por el alumno en el intento de superación de pruebas (para lograr una evaluación del mismo).
- Información referente a la superación del laboratorio.

En la **Figura 82** se muestra una captura de la ventana de gráficos (des/seleccionable desde la ventana principal) que muestra la evolución de la altura del chasis del vehículo (y), de la carretera (Road), y de la diferencia entre ambas (y -Road). En esta figura se puede observar como la evolución del chasis (y , en color azul) es más “suave” que la que presenta la carretera (Road, línea inferior en la gráfica) debido al efecto que realiza el sistema de suspensión.

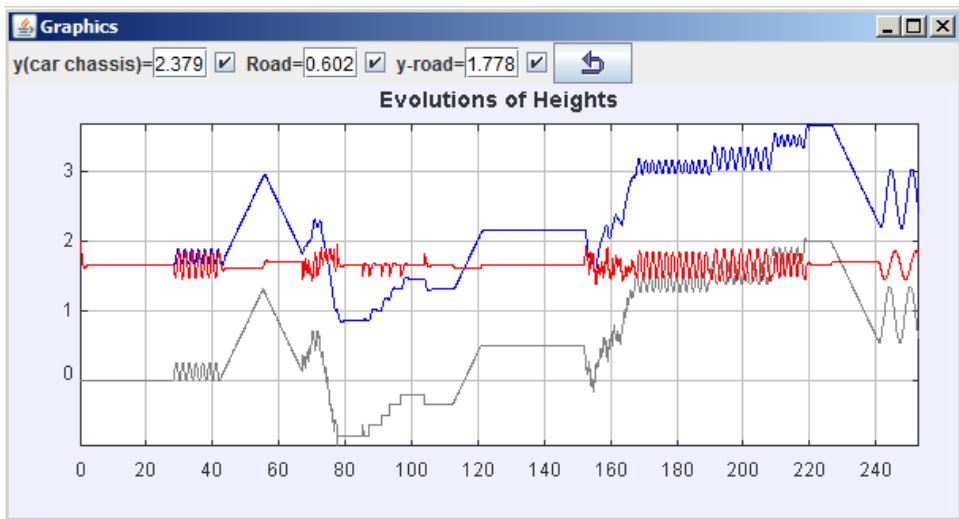


Figura 82. Captura de pantalla de la ventana emergente de gráficos del applet


Se pueden seleccionar nueve escenarios diferentes en los que se pueden realizar los experimentos, incluso combinarlos secuencialmente, como se desee. Con estas escenas y la posibilidad de cambiar las características de todas ellas según los parámetros indicados a continuación se pueden diseñar múltiples tipos de prácticas que muestren distintos comportamientos del sistema ante diferentes entradas:

- Firme plano. Permite estudiar el comportamiento del sistema ante una entrada constante. Sirve para comprobar el chasis permanece en su posición de reposo (a una distancia L_0 del asfalto).
- Subida/Bajada de acera (se puede modificar la altura de la acera con el campo “ h ”). Permite estudiar el comportamiento del sistema ante una entrada de tipo escalón positivo/negativo.
- Obstáculo/Hueco en firme (se puede modificar la altura/profundidad con el campo “ h ”). Permite estudiar el comportamiento del sistema ante una entrada de tipo impulso positivo/negativo.

- Firme ascendente/descendente (se puede modificar la pendiente con el campo “Slope”). Permite estudiar el comportamiento del sistema ante una entrada tipo rampa ascendente/descendente.
- Firme sinusoidal (se puede modificar frecuencia con el campo “w” y amplitud de la oscilación del perfil del firme con el campo “A”). Permite estudiar el comportamiento del sistema ante una entrada sinusoidal.
- Firme irregular. Permite estudiar el comportamiento del sistema ante una entrada variable aleatoria.

El **SCO3 (Figura 83)** contiene un test con 6 preguntas de tipo opción múltiple (4 opciones cada una), respuesta única con el que se trata de valorar los conocimientos adquiridos por el alumno tras su trabajo en el VL, puede obtener hasta 30 puntos.

Evaluar (SCO3)
 Use la navegación del LMS o la TOC para seleccionar un SCO diferente al presente



Fecha y hora de entrada a Evaluar (SCO3): 2016-10-03T10:36:43

TEST DE EVALUACIÓN DE LAS PRÁCTICAS

MUY IMPORTANTE, LEER ANTES DE EMPEZAR A CONTESTAR

- Presta atención a las siguientes preguntas, lee tranquilamente el enunciado y contesta con tranquilidad.
- El objetivo de las mismas es saber si has asimilado correctamente los conocimientos que debes obtener.
- Fíjate en que si respondes correctamente recibirás 5 puntos por cada pregunta, pero si lo haces de forma incorrecta te restará un punto.
- Para poder superar el test es necesario que obtengas al menos 15 puntos (módulo 30).
- Solo se tiene una oportunidad para contestar a cada una de las preguntas, una vez pases el botón "Contestar" ya no podrás volver a hacerlo.
- Si has señalado una opción pero te has arrepentido y deseas dejar la respuesta en blanco pulsa sobre "Reiniciar".
- Al final de la página debes pulsar sobre el botón "Contestar Todo y Enviar" para mostrar tus resultados finales y enviarlos al LMS.
- Se considerará que has contestado al test si al menos has contestado a una de las preguntas.

Pregunta 1

PREGUNTA 1: Constante del muelle de sistema de segundo orden sub-amortiguado (OK:5 puntos - Mal:-1 punto)

Calcular el valor mínimo que ha de tener la constante de muelle para que la gráfica que representa la amortiguación se comporte como un sistema de segundo orden sub-amortiguado con sentido ante un asfalto llano.

17.000 N/m-1
 17.150 N/m-1
 17.300 N/m-1
 17.450 N/m-1

Contestar | Reiniciar

Resultado Pregunta 1

Ya has contestado a esta pregunta, no puedes cambiar tu respuesta ==> Puntos Pregunta 5 == Puntos Total: 18

Pregunta 2

PREGUNTA 2: Comportamiento del sistema ante Subida y Bajada de acera (OK:5 puntos - Mal:-1 punto)

Señala cual es el comportamiento del sistema de amortiguación ante una subida y bajada de acera de 2 cm de amplitud

Sistema críticamente estable

Figura 83. Captura de pantalla parcial de SCO3. Evaluación final de prácticas

Informes en el LMS

Una de las principales ventajas que aporta el hecho de que exista una integración avanzada entre el laboratorio online y el LMS es que al poder comunicarse se puede almacenar los resultados del trabajo de cada alumno en el LMS. Para que alumnos y tutores o profesores puedan acceder después a esta información se debe comprobar las posibilidades que ofrece el LMS. Los LMS no ofrecen información de todos los datos guardados en la base de datos de SCORM que guardan para cada módulo. Depende de cada LMS el conjunto de datos que ofrece y la forma de mostrarlos. Además los datos mostrados no son

los mismos para alumnos y tutores. A modo de ejemplo, la **Figura 84** muestra los datos mostrados a los alumnos que trabajaron en el LMS en el laboratorio: el estado de superación de cada SCO y la puntuación alcanzada en cada uno.

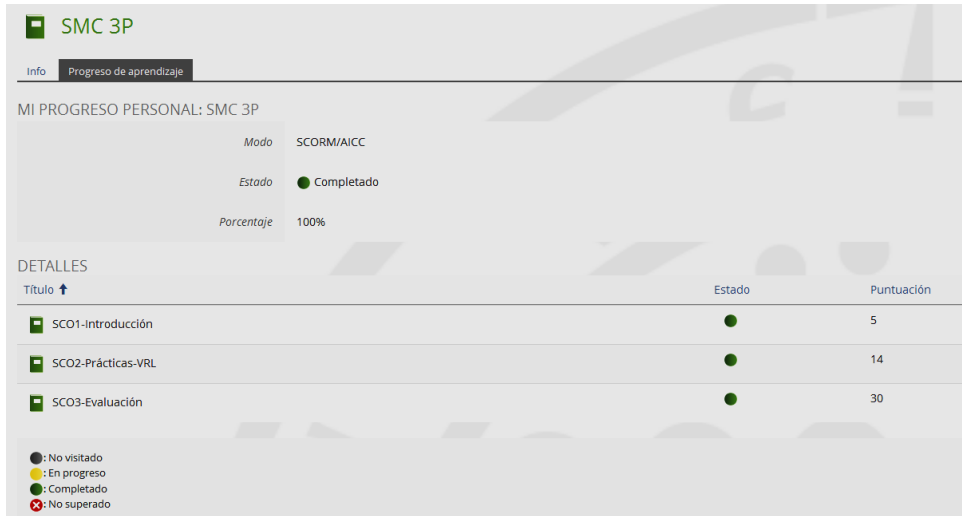


Figura 84. Ejemplo de vista del progreso de aprendizaje ofrecida a un estudiante sobre su trabajo

La información mostrada al profesorado es más completa, en el caso concreto del LMS ILIAS se muestra desde dos pantallas, una específica de SCORM y otra de progreso de aprendizaje genérica de la plataforma. En la **Figura 85** se muestra un ejemplo de la vista de informe básico de capítulos al que puede acceder el profesorado, existen otros informes seleccionables referentes a la superación, interacciones, objetivos y objetivos globales.

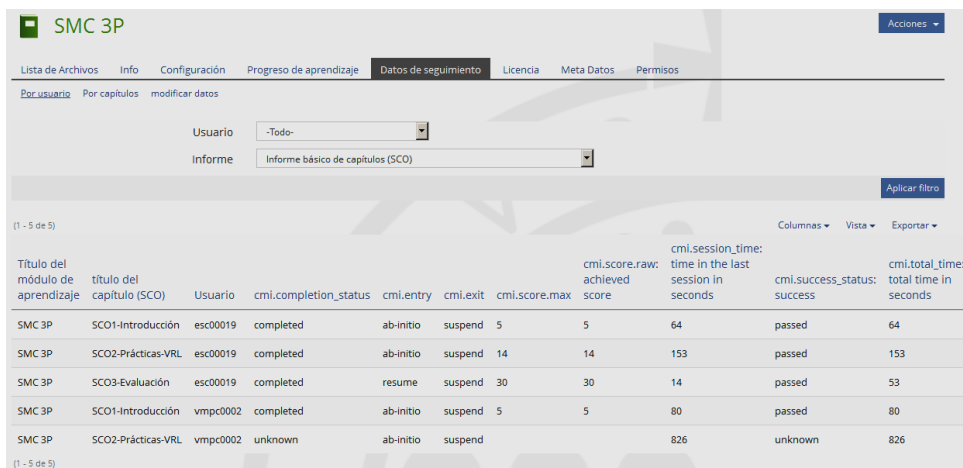


Figura 85. Ejemplo de vista de profesor del informe básico de capítulos SCORM en ILIAS

La **Figura 86** muestra la vista del progreso de aprendizaje a la que puede acceder un profesor, se incluyen los SCOs y su estado de superación mediante colores, además de datos realizados por el LMS al margen de SCORM, como son el número de accesos y el tiempo empleado.

The screenshot shows a web interface for tracking user progress. At the top, there are navigation tabs: 'Progreso de los usuarios', 'Vista matricial' (selected), 'Resumen', and 'Configuración'. Below the tabs, the title is 'VISTA MATRICIAL: SMC 3P' with 'Modo: SCORM/AICC' and '(1 - 11 de 11)'. There are buttons for 'Enviar e-mail' and 'Mostrar filtro', 'Columnas', 'Vista', and 'Exportar'. The main area is a table with columns: 'Usuario', 'Estado', 'Número de Accesos', 'Tiempo Empleado', and 'Porcentaje'. Each user row has three colored circles representing progress status. A legend at the bottom left explains the colors: black for 'No visitado', yellow for 'En progreso', green for 'Completado', and red with a cross for 'No superado'.

Usuario	Estado	Número de Accesos	Tiempo Empleado	Porcentaje
asgarcia	● ● ● ●			0%
avr00019	● ● ● ●			0%
casacar	● ● ● ●			0%
ceov0001	● ● ● ●			0%
esc00019	● ● ● ●	1	4 Minutos 5 Segundos	100%
jbo00001	● ● ● ●			0%
jfvz0002	● ● ● ●			0%
juango	● ● ● ●			0%
lmnieto2	● ● ● ●			0%
pcano	● ● ● ●			0%
vmpc0002	● ● ● ●	1	15 Minutos 8 Segundos	33%

Figura 86. Vista del progreso de aprendizaje de los usuarios del laboratorio online para un profesor

Este laboratorio se ha utilizado para la elaboración de varios trabajos publicados en congresos y jornadas nacionales e internacionales [78], [49], [77].

6.2.2. Modelado de Sistemas Dinámicos: Motor CC

Este laboratorio *online* fue pionero en muchos sentidos. Fue el primero, creado como paquete SCORM 2004 integrado en un LMS, que se ofreció a alumnos de una asignatura reglada, y fue el primero en el que se diseñaron dos planes de aprendizaje consistentes en dos secuencias de recursos de aprendizaje con los que los alumnos, divididos en dos grupos, debían trabajar. También fue el primero en el que se utilizó el paquete Java *scormRTE.jar* para facilitar las comunicaciones VL-LMS y la metodología empleada para su creación constituyó la base de la primera versión de la metodología propuesta en el capítulo 5.

En el laboratorio en sí mismo se incluyen unas prácticas en la que se muestra, mediante una aplicación de VL desarrollada en EJS, el modelo virtual de un

motor de CC y las ecuaciones que lo definen. Los estudiantes de la asignatura “Automática Industrial” de la Universidad de Jaén trabajan presencialmente en el laboratorio de control con un sistema de planta de unidad mecánica 33-100 de la empresa *Feedback Instrument* [299] que incluye un motor de CC (**Figura 87**), este es el motivo por el que se ha creado la simulación basada en este dispositivo. Los alumnos deben identificar los parámetros del modelo a partir de su comportamiento dinámico. Se ofreció a los alumnos de la asignatura “Automática Industrial” de los grados en Ingeniería Industrial, Mecánica, Eléctrica y Electrónica en la EPS de Jaén y fue utilizado por 273 estudiantes. Los objetivos del WebLab se estructuran en **globales**: a) Conocer y Aprender a modelar y simular sistemas y b) Conocimiento básico de control automático y su aplicación a la automatización industrial, y **específicos**: a) Estudio del comportamiento dinámico de un Motor CC (Corriente Continua) e b) Identificación de los parámetros que definen el modelo dinámico del sistema (motor CC).

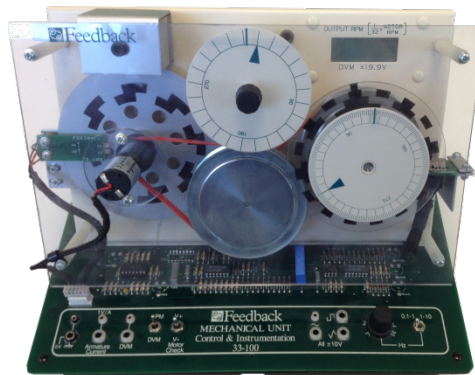


Figura 87. Unidad mecánica con Motor CC 33-100 de Feedback Instrument

Estructura del Paquete SCORM

Está formada por 4 SCOs en los que se incluye información genérica del laboratorio y la estructura del WebLab (SCO1), presentación teórica del laboratorio y un test pre-VRL (sco2), los guiones de prácticas y una aplicación VL desarrollada con EJS (SCO3) y una evaluación final consistente en un test de 7 preguntas que se corrige automáticamente. La estructura del paquete SCORM se creó a partir de una plantilla descargada de la Web de ADL [61]. Los cuatro SCOs que forman el paquete SCORM utilizan el RTE de SCORM para comunicarse con el LMS a través de la librería JavaScript *APIWrapper.js*, esto significa que se

llaman a las funciones de *APIWrapper.js* desde código JavaScript incrustado en el código HTML de las páginas Web asociadas a cada SCO. Además se utilizó el paquete Java *scormRTE.jar*, explicado en el apartado 0, desarrollado para facilitar la invocación de funciones de *APIWrapper.js*. La **Figura 88** muestra esta estructura y las comunicaciones SCORM establecidas entre el WebLab y el LMS ILIAS.

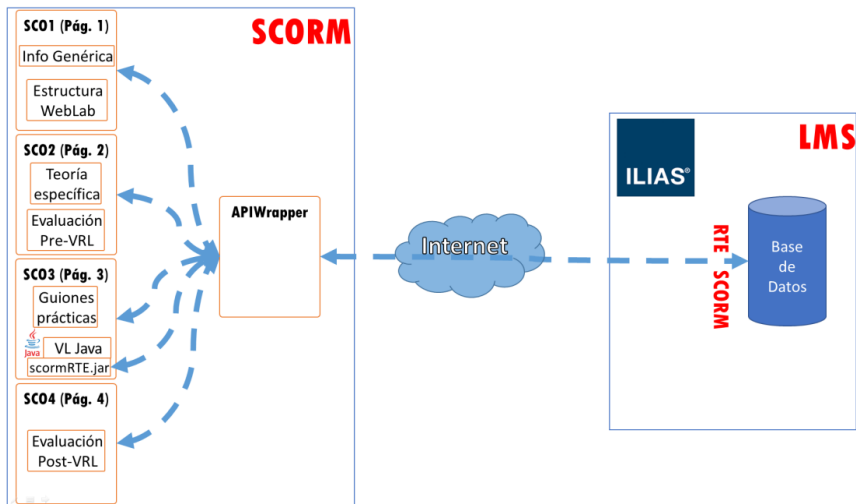



Figura 88. Estructura de Lab "Modelado de Sistemas Dinámicos: MotorCC" y comunicaciones LMS

La página 1 (**SCO1**) (**Figura 89**) muestra una tabla que incluye información general y básica del laboratorio para que el alumno pueda contextualizarlo dentro de los contenidos teóricos de la asignatura, conozca los objetivos que persiguen con el mismo y las competencias que puede adquirir. También se muestra información sobre la estructura del laboratorio, los contenidos del mismo y la forma de superarlo. Se añade un botón para acceder a la siguiente página (SCO2) del laboratorio. Se supera simplemente con estar más de 5 segundos en la página y no tiene ninguna puntuación asociada.


La página 2 (**SCO2**) contiene información teórica específica del laboratorio y un test pre-VL. En la parte de teoría (**Figura 90**) se muestra el modelo dinámico del motor CC en el que está basada la práctica (motor CC del fabricante *Feedback Instrument*) y el modelo dinámico de primer orden que se va a simular en el VL. También se definen la ganancia estática y la constante de tiempo del sistema y la forma de calcularlas gráficamente ante la respuesta que presenta el sistema cuando se aplica una entrada escalón unitario. El test (**Figura 91**) consiste en 7 preguntas de opción múltiple, respuesta simple con el que se trata de asegurar

que los estudiantes tienen los conocimientos previos mínimos para trabajar en el VL.

Introducción al WebLab (Página1) Introducción al WebLab (Página1)



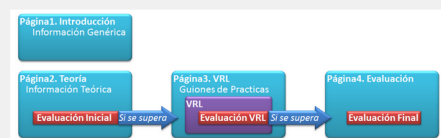
MODELADO DE SISTEMAS DINÁMICOS: MOTOR CC



UNIVERSIDAD:	UNIVERSIDAD DE JAÉN
TITULACIONES:	GRADOS EN INGENIERÍA INDUSTRIAL, MECÁNICA, ELÉCTRICA Y ELECTRÓNICA
ASIGNATURA:	AUTOMÁTICA INDUSTRIAL (NIVEL BÁSICO)
PLANIFICACIÓN:	2º CURSO // 2º CUATRIMESTRE
CONTEXTO DEL EJERCICIO:	BLOQUE III. MÉTODOS DE CONTROL TEMA 6. MODELADO DE SISTEMAS
OBJETIVOS Y COMPETENCIAS:	<ul style="list-style-type: none"> • GLOBALES: <ul style="list-style-type: none"> ◦ Conocer y Aprender a modelar y simular sistemas. ◦ Conocimiento básico de control automático y su aplicación a la automatización industrial. • ESPECÍFICOS: <ul style="list-style-type: none"> ◦ Estudio del comportamiento dinámico de un Motor CC (Corriente Continua). ◦ Identificación de los parámetros que definen el modelo dinámico del sistema (motor CC).
TEORÍA:	Enlace a la carpeta de teoría: BLOQUE III. MÉTODOS DE CONTROL

Estructura del WebLab (Laboratorio Web) y Modo de superarlo

La siguiente figura muestra la organización de los contenidos del WebLab:



El Contenido está estructurado en 4 secciones (Páginas):

- **Página1.Introducción (Esta Página):** Proporciona información general sobre el WebLab y se enmarca dentro de los estudios de grado.
- **Página2.Teoría (Página siguiente):** Proporciona información teórica relacionada con el modelado de los sistemas dinámicos, especialmente sobre el modelado de un sistema de Motor CC. También contiene un test inicial que trata de asegurar que se han asimilado los conceptos teóricos explicados antes de acceder al laboratorio en sí con el objetivo de que no pueda acceder al laboratorio alguien que no ha adquirido los conocimientos necesarios para usarlo y aprovecharlo.
- **Página3.VRL:** (Para acceder a esta sección es necesario haber superado el test inicial de la sección Página2.Teoría) Proporciona los guiones de las prácticas que se deben realizar con el laboratorio y el propio VRL (Virtual Remote Lab) en el que se podrá trabajar con el fin de superar las prácticas que se deben realizar por cada alumno.
- **Página4.Evaluación (Última página):** (Para acceder a esta sección es necesario haber superado la evaluación automática de la sección Página3.VRL.) Muestra un test final relacionado con todo el trabajo e información previa incluida en el WebLab que los estudiantes deben superar.

Para superar el WebLab se debe llegar hasta la sección Página4 Evaluación y superar el test de evaluación final.

Figura 89. Captura de pantalla del SCO1 con la información genérica y estructura del laboratorio

El test pre-VL asigna una puntuación a los estudiantes que lo contestan de forma automática en función de sus respuestas. Si la puntuación es superior a 14 (el máximo es 28), se almacenará en el LMS el estado superado del SCO y la puntuación alcanzada, y se permitirá acceder a la siguiente página (SCO3).

La página 3 (**SCO3**) incluye los guiones de prácticas del VL (**Figura 92**) y el propio VL incrustado como un *applet* de Java. Los estudiantes deben realizar 2 prácticas, en la primera de ellas se trata de realizar pruebas con la GUI del simulador, constituye un manual del *software* que debe manipularse.

En la parte inferior del SCO3 se encuentra el *applet* Java del VL del Motor CC (**Figura 93**). La GUI está estructurada en tres zonas:

Teoría del WebLab (Página2)
Teoría del WebLab (Página2)

MODELADO DE SISTEMAS DINÁMICOS: MOTOR CC

Introducción

El objetivo de este ejercicio es iniciar al estudiante en el modelado de sistemas dinámicos en general, y, en particular, al modelado de un motor electro-mecánico. Con este propósito se presenta el modelado de un motor de corriente continua (Motor CC) como el mostrado en la figura 1.

En el Laboratorio de Control Automático de la Universidad de Jaén hay varias unidades mecánicas del fabricante Feedback Instrument Limited que incluyen un motor CC. Entre los elementos electromecánicos que incluyen estas unidades se puede encontrar un motor CC, un tacómetro analógico, potenciómetros de entrada y salida analógicos, codificadores absolutos e incrementales y un freno magnético.



Figura 1. Motor CC.

Los estudiantes usarán una aplicación EJS (*Easy Java Simulations*) que está en la página siguiente (Página3), para utilizar un modelo virtual de motor CC que les permitirá identificar su modelo dinámico. Antes de identificar el modelo se deben calcular las ecuaciones que definen su dinámica. Esta dinámica es un sistema lineal de segundo orden (de primer orden si se desprecia la inductancia, L).

A continuación se muestran las ecuaciones que gobiernan el comportamiento de un motor CC. Más adelante se obtendrá el modelo simplificado de primer orden.

Modelo Dinámico

El siguiente esquema representa el modelo dinámico de un motor CC:



Figura 2. Modelo dinámico de un motor CC.

Figura 90. Captura de pantalla parcial de la teoría presentada en el SCO2

- **Superior.** Muestra dos paneles (uno sobre otro), en el superior se muestran datos del modelo SCORM recuperados por el VL del LMS que muestran la versión del modelo de datos RTE, el nombre y la identificación del estudiante, el modo de entrada al SCO y la fecha y hora (timestamp) de entrada en el VL. En el inferior se muestra una botonera con los controles de ejecución de la aplicación (ejecución pausa, paso a paso y reseteo), campos para ver y manipular el tiempo (“t”), la entrada al sistema (“U_a”), la salida del sistema (“W”), casillas de activación/desactivación de las gráficas de “U_a” y “W” en la zona de gráficos, un botón para borrar las gráficas de la zona de gráficos y si la aplicación se está ejecutando como *applet* con comunicaciones LMS.
- **Media.** Muestra dos paneles (uno sobre otro). En el superior se distinguen dos partes, en izquierda aparece una fotografía del motor CC con elementos móviles dependientes de la velocidad alcanzada y en la derecha las gráficas que se hayan seleccionado en la zona superior (la tensión de entrada al sistema y/o la velocidad angular). En el inferior se muestra un panel de control de visualización de la gráfica.

Test previo al Laboratorio (Evaluación Inicial)

MUY IMPORTANTE, LEER ANTES DE EMPEZAR A RESPONDER EL TEST

- Presta atención a las siguientes preguntas, lee tranquilamente el enunciado y contesta con tranquilidad.
- El objetivo de las mismas es saber si has asimilado correctamente los conocimientos que debes obtener.
- Fíjate en que si respondes correctamente recibirás 4 puntos por cada pregunta, pero si lo haces de forma incorrecta te restará un punto.
- Para poder superar el test es necesario que obtengas al menos 14 puntos (máximo 28).
- Sólo se tiene una oportunidad para contestar a cada una de las preguntas, una vez pulses el botón "Contestar" ya no podrás volver a hacerlo.
- Si has señalado una opción pero te has arrepentido y deseas dejar la respuesta en blanco pulsa sobre "Reiniciar".
- Al final de la página debes pulsar sobre el botón "Contestar Todo y Enviar" para mostrar tus resultados finales y enviarlos al LMS.
- Se considerará que has contestado al test si al menos has contestado a una de las preguntas.
- No refresques el contenido de la página en el navegador, si lo haces se cargará de nuevo la Página1-Introducción y deberás volver a empezar el WebLab.

Pregunta 1

PREGUNTA 1: Linealidad del Sistema.
(Bien:4 puntos - Error:-1 punto)

¿Es lineal la dinámica del sistema estudiado?

Sí. Además puede aproximarse a un sistema de primer orden.
 No. Es un proceso no lineal.

Resultado de pregunta 1

¡Estupendo!, sigue así ==> Puntos obtenidos en esta pregunta:4 == Total de Puntos en el test:4

Pregunta 2

PREGUNTA 2: Definición de sistema de primer orden.
(Bien:4 puntos - Error:-1 punto)

Un sistema de primer orden se define usando los siguientes parámetros...

La constante de tiempo del sistema

Figura 91. Captura de pantalla parcial del test pre-VL presentado en el SCO2

MODELADO DE SISTEMAS DINÁMICOS: MOTOR CC

Introducción a las Prácticas

El objetivo de este ejercicio es simular el comportamiento de un motor de corriente continua (CC) como el mostrado en la Figura 1 (de la marca Feedback) e identificar algunos de los parámetros que caracterizan su comportamiento.

Figura 1. Motor de corriente continua (CC).

Practica 1: Pruebas del simulador del Motor CC.

Con el fin de ganar confianza y aprender el funcionamiento de este simulador de Motor CC se debe trabajar y dominar la interfaz que se ofrece mediante la realización de una serie de pruebas que se describen a continuación.

Antes de ello, debes saber que todos los elementos del interfaz tienen un texto de ayuda asociado. Esta ayuda se muestra en forma de texto emergente que aparecerá cuando el puntero del ratón se sitúa inmóvil sobre cada uno de los elementos durante un tiempo (aproximadamente un segundo).

Figura 92. Captura de pantalla parcial de los guiones de prácticas presentados en el SCO3

- **Inferior.** Muestra el panel de la práctica en ejecución, la práctica 1ª (Figura 93) muestra campos para definir el periodo y la amplitud de la señal de entrada, 2 botones para definir una señal “U_a” (sinusoidal o tren cuadrado de pulsos), y un botón para pasar a la práctica 2ª.

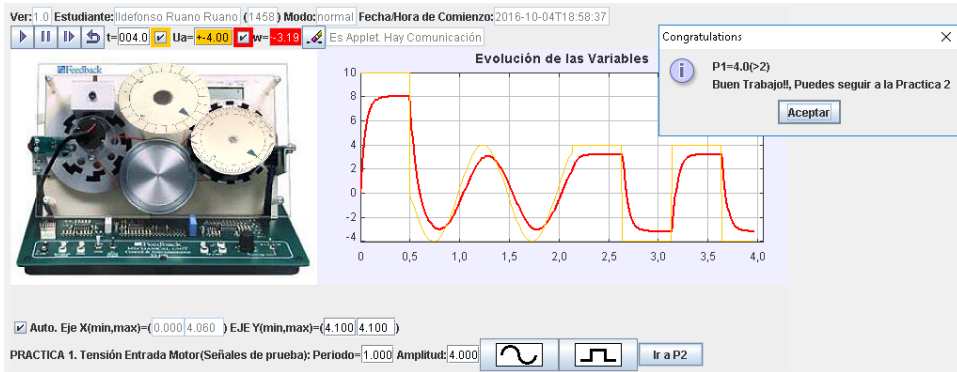


Figura 93. applet Java del VL con práctica 1ª y ventana emergente de paso a la 2ª

Cuando el estudiante pulsa el botón “Ir a P2” para realizar la práctica 2, el VL comprueba que el estudiante haya realizado correctamente las pruebas de la GUI pedidas y muestra una ventana emergente en la que informa de la puntuación alcanzada (máximo 4 puntos), en caso de no llegar al mínimo exigido (2 puntos) el alumno se debe mantenerse en la práctica 1 hasta alcanzar el mínimo exigible.

Si el estudiante alcanza el mínimo exigido, tras cerrar la ventana de información, se le permite realizar la práctica 2. Para realizar la práctica 2 deben basarse en la teoría mostrada en el SCO2: $G(s)$, la función de transferencia del sistema, tiene una ecuación como la mostrada en la **Figura 94**. El estudiante debe identificar los parámetros K (ganancia estática) y T (constante de tiempo) que definen el comportamiento dinámico del motor aplicando una tensión de entrada (“ u_a ”) constante de 1 voltio en continua (entrada escalón unitario) y midiendo la respuesta del motor (“ ω ”).

$$U_a(s) \rightarrow \boxed{G(s)} \rightarrow \Omega(s) \qquad G(s) = \frac{\Omega(s)}{U_a(s)} = \frac{K}{1+Ts}$$

Figura 94. Esquema del sistema y ecuación de transferencia del sistema de Motor CC simplificado

El panel inferior de la GUI cambia cuando se realiza la práctica 2 (**Figura 95**). Muestra dos campos en donde puede introducir los parámetros K y T que deben calcular midiendo sobre las gráficas de la zona media de la GUI. Cada estudiante realiza un experimento individualizado: en función de su identificación tiene un par K - T de parámetros personalizado que debe calcular. Cuando tienen la respuesta correcta deben pulsar en el botón “Comprobar”, aparecerá una ventana emergente que les mostrará el resultado obtenido, el cual también se

mostrará en un campo del panel inferior. Si no se llega al mínimo exigido deberán intentarlo de nuevo, para ello disponen de 3 intentos, si se consumen sin llegar al mínimo el VL vuelve a mostrar la GUI de la práctica 1 para que el estudiante comience de nuevo a trabajar en el VL.

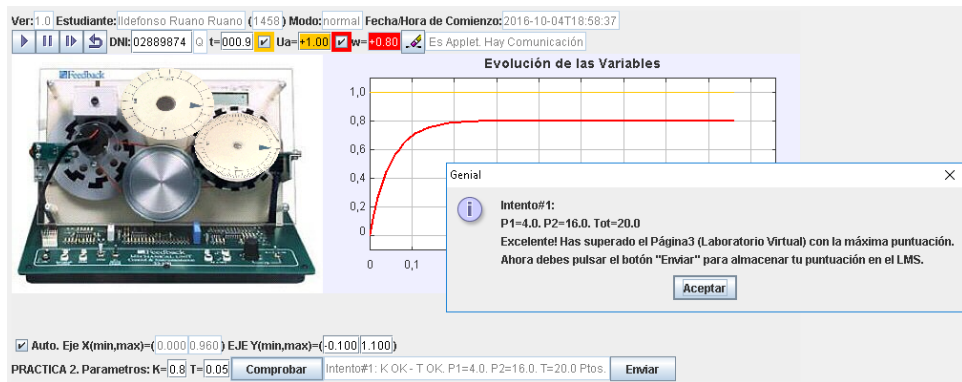


Figura 95. applet Java del VL con práctica 2ª y ventana emergente de paso al SCO3

Cuando el estudiante supera la práctica 2, en alguno de sus 3 intentos, y pulsa el botón “Comprobar”, aparece una ventana emergente como la mostrada en la **Figura 95** y un botón con el texto “Enviar” en el panel inferior de la GUI. Cuando se cierra la ventana emergente, y se pulsa este botón, se envía la puntuación alcanzada desde el VL al LMS junto al estado de superación “*passed*” y de terminación “*completed*”. Estos datos serán almacenados por el LMS en la base de datos que mantiene del modelo RTE de SCORM. El VL antes de cerrar la sesión de comunicación SCORM programa el siguiente destino para que el estudiante acceda al SCO4.

La página 3 (**SCO4**) (**Figura 96**) contiene un test con 7 preguntas de tipo opción múltiple, respuesta única, test post-VL, con el que se trata de valorar los conocimientos adquiridos por el alumno tras su trabajo en el VL, puede obtener hasta 28 puntos que también son almacenados en el LMS por el SCO4 junto a los estados de terminación y superación.

La programación de la secuenciación y navegación entre SCOs que se ha realizado en el propio paquete SCORM ha incluido una secuencia de aprendizaje de los recursos incluidos que se acaban de describir, esta secuencia se muestra con más claridad en la **Figura 97**, la cual también incluye la secuenciación entre prácticas programada en el interior del código java del VL.

Evaluación (Página4)
Evaluación (Página4)

MODELADO DE SISTEMAS DINÁMICOS: MOTOR CC

Evaluación final de las Prácticas del WebLab

MUY IMPORTANTE, LEER ANTES DE EMPEZAR A RESPONDER EL TEST

- Presta atención a las siguientes preguntas, lee tranquilamente el enunciado y contesta con tranquilidad.
- El objetivo de las mismas es saber si has asimilado correctamente los conocimientos que debes obtener.
- Fíjate en que si respondes correctamente recibirás 4 puntos por cada pregunta, pero si lo haces de forma incorrecta te restará un punto.
- Para poder superar el test es necesario que obtengas al menos 14 puntos (máximo 28).
- Sólo se tiene una oportunidad para contestar a cada una de las preguntas, una vez pulses el botón "Contestar" ya no podrás volver a hacerlo.
- Si has señalado una opción pero te has arrepentido y deseas dejar la respuesta en blanco pulsa sobre "Reiniciar".
- Al final de la página debes pulsar sobre el botón "Contestar Todo y Enviar" para mostrar tus resultados finales y enviarlos al LMS.
- Se considerará que has contestado al test si al menos has contestado a una de las preguntas.
- No refresques el contenido de la página en el navegador, si lo haces se cargará la Página1-Introducción y deberás volver a empezar todo el WebLab.

Pregunta 1

PREGUNTA 1: Salida a entrada escalón.
(Bien:4 puntos - Error:-1 puntos)

Se tiene un sistema de primer orden al que se le aplica una entrada en el dominio temporal como la mostrada en la siguiente figura:

Indica cual de las siguientes gráficas muestra la forma que debe mostrar la salida del sistema también en el dominio temporal.

Figura 96. Captura de pantalla parcial del SCO4 con el test de evaluación post-VL

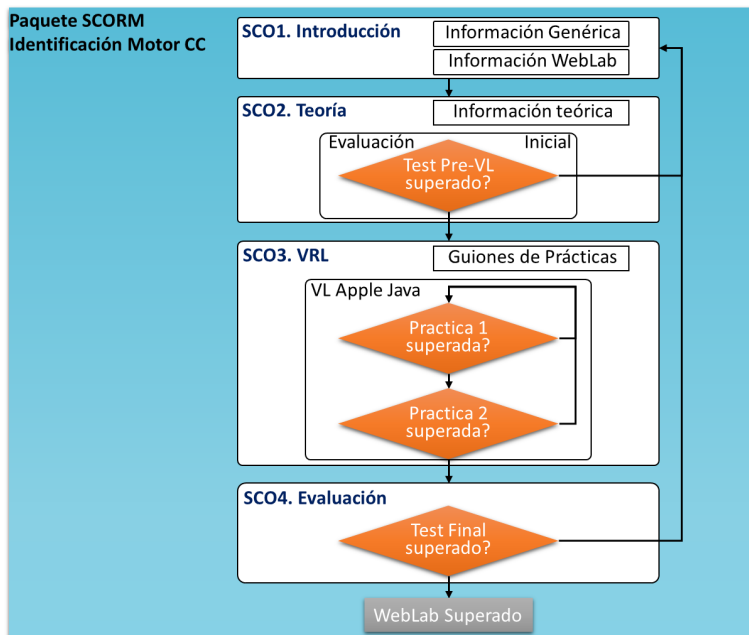


Figura 97. Secuenciación y Navegación del paquete SCORM del Laboratorio Virtual del Motor CC

Este WebLab, entendiendo como tal al paquete SCORM, se presentó a los alumnos en el LMS ILIAS inmerso en un plan de aprendizaje que consideraba dos secuencias de aprendizaje distintas que se van a explicar a continuación.

Presentación del Laboratorio online a los Estudiantes

La asignatura “Automática Industrial” se imparte en el 2º cuatrimestre del 2º año de los Grados en Ingeniería Industrial, Mecánica, Eléctrica y Electrónica de la EPS de Jaén. Proporciona una introducción a la aplicación de la automática en entornos industriales. También proporciona conocimientos básicos de diferentes tecnologías de control aplicadas a procesos continuos y discretos. En el curso 2013-14 se matricularon 369 estudiantes de los que 356 estaban identificados en ILIAS, el LMS institucional de la UJA en el que los profesores de la asignatura proporcionaban herramientas y recursos asociados con la misma. 338 de los estudiantes identificados en ILIAS se unieron a uno de los 17 grupos de prácticas que se crearon para organizar la docencia, cada uno de los cuales posee un espacio virtual en ILIAS al que sólo pueden acceder los miembros de cada grupo.

En cada uno de estos espacios virtuales se creó una única carpeta en la que se incluyeron los recursos asociados al WebLab, se crearon carpetas de dos tipos, A y B, de forma que los 17 grupos se repartieron para tener una carpeta de uno de estos tipos. El contenido existente en las carpetas de tipo A se muestra en la **Figura 98**.

Se incluye una explicación de los recursos que se les presenta indicando la secuencia de aprendizaje que deben realizar y los propios recursos:

1. Un test de comprobación de conocimientos previos, “Test WebLab MotorCCES”, que deben ejecutar antes de iniciar cualquier acción formativa relacionada con el WebLab. Se trata del mismo test de evaluación de conocimientos adquiridos con el VL, el test post-VL incluido en la página 4 (SCO4) del paquete SCORM creado con la herramienta de creación de tests que ofrece ILIAS.
2. Un documento con instrucciones de verificación, instalación y configuración de Java para que no tuvieran problemas a la hora de ejecutar el *applet* Java incrustado en el SCORM.

Modelado de Sistemas Dinámicos. Motor CC
WebLab "Modelado de Sistemas Dinámicos. Motor CC". Versión: ESP

Acciones ▾

Contenido Info Configuración Progreso de aprendizaje Exportar Permisos

Ver Gestionar contenido Ordenación Editor Texto/Multimedia

Añadir nuevo elemento ▾

Carpeta de WebLab: "Modelado de Sistemas Dinámicos. Motor CC". Versión: ESP

En esta carpeta están las instrucciones y recursos que necesitas para realizar el WebLab de Modelado de un Motor de Corriente Continua como el de la siguiente figura:

Instrucciones

A continuación se describen los pasos que debes realizar para completar con éxito el WebLab:

1. Ejecuta el test "**Test WebLab MotorCCES**". Se trata de un test que sirve para conocer tus conocimientos previos a la realización del laboratorio. El resultado que obtengas en este test no se tendrá en cuenta para tu evaluación en la asignatura. Hasta que no finalices el test no podrás continuar y ejecutar el WebLab.
2. Abre y lee el módulo "**Pasos Previos a Ejecución de WebLab**". En este módulo se explica que características debe tener el PC en el que ejecutes el WebLab para que no te de problemas de compatibilidad o falta de recursos (Explica como instalar JAVA y configurarlo correctamente).
3. Abre y Ejecuta el módulo SCORM "**UJACCES**" (Recuerda que hasta que no finalices el test no podrás acceder a este recurso). Este es el módulo del WebLab, contiene 4 páginas Web en las que debes leer y realizar las acciones del laboratorio que se piden, hay teoría necesaria para hacer el laboratorio virtual, tests de evaluación, guiones de prácticas y el propio laboratorio. No tengas prisa ya que no hay límite de tiempo.
4. Cuando superes el Laboratorio podrás acceder al último elemento: la "**Encuesta WebLab**", se trata de una encuesta sencilla en la que deberás dar tu opinión sobre el módulo UJACCES. Sólo podrás ejecutarla una vez y no tardarás mucho en hacerla.

Ánimo y suerte

CONTENIDO

- Test WebLab MotorCCES
- Pasos Previos a Ejecución de WebLab
Instrucciones de verificación, instalación y configuración de JAVA para la ejecución de applets en ILIAS (con imágenes y en...
Tipo: Módulo de Aprendizaje
- UJACCES
WebLab: "Modelado de Sistemas Dinámicos: Motor CC". Versión: ESP
Tipo: Módulo de aprendizaje SCORM/AICC Finalizada: Descargar informe
Progreso de aprendizaje: ●
Precondiciones que deben cumplirse para tener acceso a estos elementos: » Mostrar
- Encuesta WebLab
Sirve para dar la opinión sobre el WebLab UJACCES una vez completado
completada

Figura 98. Captura de pantalla de carpeta virtual en ILIAS tipo A

3. El paquete SCORM del WebLab descrito anteriormente, incluyendo el VL y todos los recursos asociados.
4. Una encuesta de opinión para conocer el nivel de satisfacción de los alumnos que completan el WebLab SCORM.

Además de estos contenidos, los estudiantes contaban con un foro común a los dos grupos en el que podían intercambiar experiencias y establecer comunicaciones con el profesorado sobre el laboratorio. El contenido de las carpetas creadas a los grupos de tipo B fue el mismo que el mostrado para el

tipo A, salvo que **no incluía el test previo** “Test WebLab MotorCCES”. En la **Figura 99** se puede ver los contenidos y la secuencia de aprendizaje que se estableció teniendo en cuenta las posibilidades que ofrece el LMS para crear precondiciones de acceso a un recurso.

Evaluación de la Efectividad del WebLab

Uno de los objetivos más importantes de un laboratorio *online* docente, probablemente el más importante, es que aumente el conocimiento y comprensión de la naturaleza de los experimentos realizados [9], [300]. La efectividad de un laboratorio debería medirse en función del entendimiento alcanzado. Sin embargo, es un elemento muy difícil de cuantificar. Para evaluar este laboratorio se han tenido en cuenta distintos factores:

- **Resultados.** Los resultados absolutos obtenidos por los 273 estudiantes basados en la evaluación del trabajo realizado en el WebLab [36]. Aquí se tendrá en cuenta los resultados de la evaluación automática realizada por los SCOs del paquete SCORM y almacenada en el LMS. La **Tabla 10** muestra un resumen de los datos de uso y resultados obtenidos por los alumnos en este laboratorio *online*. Por otro lado también se han considerado los resultados obtenidos por los estudiantes que completaron el WebLab respecto al conocimiento inicial con el que iniciaron la propuesta de aprendizaje. Los grupos A, grupo de control, y B, grupo experimental, fueron creados aleatoriamente intentando que tuvieran el mismo número de estudiantes, se supone que cuentan con un nivel inicial de conocimientos similar. El propósito de crear estas diferencias era evaluar el conocimiento inicial de los estudiantes, dado por los resultados medios que obtuvieron los estudiantes del grupo A en el test previo se supone similar al de los estudiantes del grupo B. De este modo si se comparan el resultado promedios alcanzado por los estudiantes del grupo A en el test previo al WebLab SCORM con el resultado promedio alcanzado por los estudiantes del grupo B en el mismo test, que ejecutaron tras trabajar en el VL (test post-VL del SCORM) se obtendrá la mejora de conocimiento promedio alcanzada.
- **Encuesta.** La satisfacción de los estudiantes basada en las opiniones recogidas por la encuesta sobre el WebLab que completaron 128 de los estudiantes que completaron el WebLab (al ser voluntaria no fue realizada por todos los estudiantes).

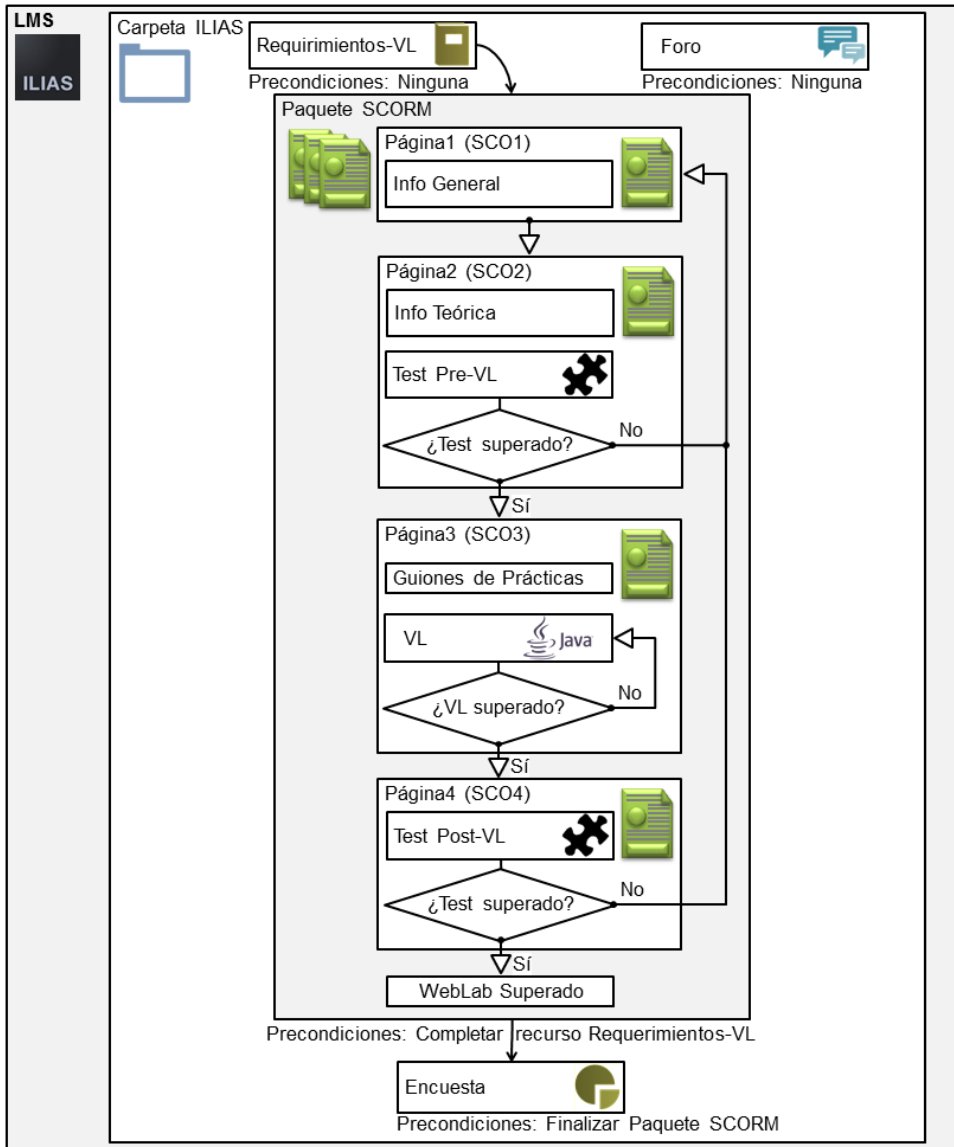


Figura 99. Contenidos del grupo B y secuencia de aprendizaje y navegación completa (incluyendo recursos del LMS y en el interior del paquete SCORM)

- **Otros.** Aplicación de criterios de evaluación de la efectividad basada en aspectos pedagógicos [301] y en la calidad basada en aspectos administrativos, técnicos y educativos [302].

Resultados. Los hechos y las conclusiones más sobresalientes que se pueden deducir de los resultados obtenidos son los siguientes (se han resaltado los datos que se van a comentar mediante un código de colores en la **Tabla 10**):

Tabla 10. Resumen de uso y resultados obtenidos en el VL de Modelado de Motor CC

Item	Grupo A	Grupo B	Total
Estudiantes (Est.)	177 (100%)	161 (100%)	338 (100%)
Est. acceden test previo	145 (81.9%)	-	-
Est. superan test previo	111 (62.71%)	-	-
Puntos/Est. (1-10)	6.4 (2.6 SD)	-	-
Paquete SCORM Completo			
Est. acceden WebLab	130 (73.4%)(100%)	143 (88.8%)(100%)	273 (80.8%)(100%)
Promedio intentos/ Est.	5.7 (5.0 SD)	6.8 (6.2 SD)	6.3 (5.7 SD)
Promedio tiempo/ Est.	67.4 (60.5 SD)	93.2 (79.9 SD)	80.8 (72.3 SD)
Promedio tiempo/(Intento Est.)	18.1 (21.7 SD)	22.8 (41.0 SD)	20.7 (33.4 SD)
Promedio puntos/Est. (0-10)	7.9 (1.8 SD)	7.5 (2.0 SD)	7.7 (1.9 SD)
SCO2. Teoría			
Est. acceden	130 (73.4%)(100%)	141 (87.6%)(98.6%)	271 (80.2%)(99.3%)
Promedio time/ Est.	17.7 (38.7 SD)	24.2 (26.9 SD)	21.1 (33.2 SD)
Promedio tiempo/(Intento Est.)	5.3 (11.1 SD)	6.7 (17.2 SD)	6.1 (14.7 SD)
Est. consiguen puntos	128 (72.3%)(98.5%)	139 (86.3%)(97.2%)	267 (79.0%)(97.8%)
Promedio puntos/Est. (0-10)	8.8 (1.7 SD)	8.9 (1.5 SD)	8.9 (1.6 SD)
SCO3. VRL			
Est. acceden	127 (71.8%)(97.7%)	139 (86.3%)(97.2%)	266 (78.7%)(97.4%)
Promedio time/ Est.	44.0 (36.3 SD)	58.4 (61.2 SD)	51.5 (51.3 SD)
Promedio tiempo/(Intento Est.)	11.2 (13.1 SD)	13.6 (24.1 SD)	12.5 (19.7 SD)
Est. consiguen puntos	123 (69.5%)(94.6%)	128 (79.6%)(89.5%)	251 (74.3%)(91.9%)
Promedio puntos/Est. (0-10)	7.5 (2.3 SD)	7.0 (2.3 SD)	7.16 (2.3 SD)
SCO4. Evaluación			
Est. acceden	119 (67.2%)(91.5%)	121 (75.2%)(84.6%)	240 (71.0%)(87.9%)
Promedio time/ Est.	7.4 (7.9 SD)	13.3 (12.2 SD)	10.4 (10.7 SD)
Promedio tiempo/(Intento Est.)	2.0 (2.6 SD)	3.2 (4.1 SD)	2.6 (3.5 SD)
Est. consiguen puntos	119 (67.2%)(91.5%)	121 (75.2%)(84.6%)	240 (71.0%)(87.9%)
Promedio puntos/Est. (0-10)	8.2 (1.52 SD)	8.0 (1.8 SD)	8.1 (1.7 SD)
Est. superan WebLab	118 (66.7%)(90.8%)	118 (73.3%)(82.5%)	236 (69.8%)(86.4%)

Hecho 1º. El número de estudiantes del grupo B que acceden al WebLab es más de un 15% superior a los del grupo A (88,8% frente a 73,4%), sin embargo el 90% de los estudiantes del grupo A que acceden al WebLab lo superan y con una buena nota media (7,9) mientras que sólo el 82,5% de los estudiantes del grupo B que acceden al WebLab lo superan y con peor nota media (7,4). Se trata de porcentajes significativos. **Explicación:** Los estudiantes del grupo A, que tuvieron que realizar el test previo al WebLab, tuvieron excesivo respeto o miedo al fracaso, y no participaron tanto como los estudiantes del grupo B, que no tuvieron que hacerlo. Por otro lado, los estudiantes del grupo A que después de responder al test previo entraron en el WebLab sacaron mejores porcentajes de superación y calificaciones de media que los del grupo B ya que se trataba de alumnos, de media, más cualificados, que además repetían el test como evaluación final del WebLab.

Hecho 2º. Los estudiantes emplearon más tiempo en trabajar en el SCO3, página en la que se encontraba el VL (más del 50% del tiempo empleado en el WebLab). La desviación estándar (SD) es muy alta en todos los valores temporales. **Explicación:** Cuando los alumnos trabajan a su ritmo sin límites de tiempo ni de uso o repetición, frecuentemente dejan sesiones de trabajo en los navegadores abiertas mientras hacen otras cosas. Esto motiva las altas desviaciones producidas. De todos modos esto no influye en el hecho de que la parte del WebLab en la que se emplea más tiempo es en la página que aloja al *applet* Java del laboratorio con el que deben realizar los experimentos, algo deseable en cualquier modelo de laboratorio.

Hecho 3º. La puntuación media obtenida por los estudiantes del grupo A que ejecutaron el test previo al WebLab fue 6,4 (SD 2,6), mientras que la puntuación media que obtuvieron en el mismo test los estudiantes del grupo B tras realizar el laboratorio fue 8,0 (SD 1,8). **Explicación:** La ejecución y superación con éxito del laboratorio online ha proporcionado un conocimiento extra a los estudiantes. El uso del WebLab es positivo para los alumnos que han participado. Suponiendo que el conocimiento inicial de los alumnos del grupo B fuera el mismo que los del grupo A la mejora producida es superior al 25% como puede verse gráficamente en la *Figura 100*.

Hecho 4º. El 86,4 % de los estudiantes que accedieron al WebLab lo superaron, y la mayoría de ellos obtuvieron calificaciones bastante positivas con una media de 7,7 (SD 1,9). **Explicación:** El WebLab ha resultado una experiencia positiva para la mayoría de los estudiantes que lo completaron.

Encuesta. 128 de los 236 estudiantes que accedieron y superaron el WebLab participaron en una encuesta. Respondieron preguntas con el fin de obtener su opinión sobre el laboratorio y el modelo propuesto en el que se basa. La encuesta, de 23 preguntas, estaba estructurada en 10 páginas para promover la participación y aumentar su usabilidad.

Las primeras 5 páginas (11 preguntas) se usaron para conocer las características de los participantes y los entornos en los que trabajaron con el laboratorio. La *página 1* contenía 3 preguntas personales sobre los encuestados que demostraron que la mayoría tenían entre 20 y 24 años (73%), eran chicos ((84%) y estaban estudiando el grado de Ingeniería Mecánica (52%). La *página 2* tenía una pregunta para comprobar que el encuestado había accedido al WebLab (100%). La *página 3* tenía 2 preguntas usadas para comprobar el trabajo

realizado por los estudiantes previo al uso del WebLab, la mayoría (97,7%) habían leído el tema 6 de teoría en el que se basa el WebLab y una mayoría algo menor afirmaba haber asistido a las clases presenciales de teoría en la que se explicó este tema (62%). La **página 4** incluía una pregunta usada para comprobar que el encuestado había superado el WebLab (100%) y la única pregunta incluida en la **página 5** mostraba que el 70% de los encuestados había ejecutado el WebLab más de una vez.

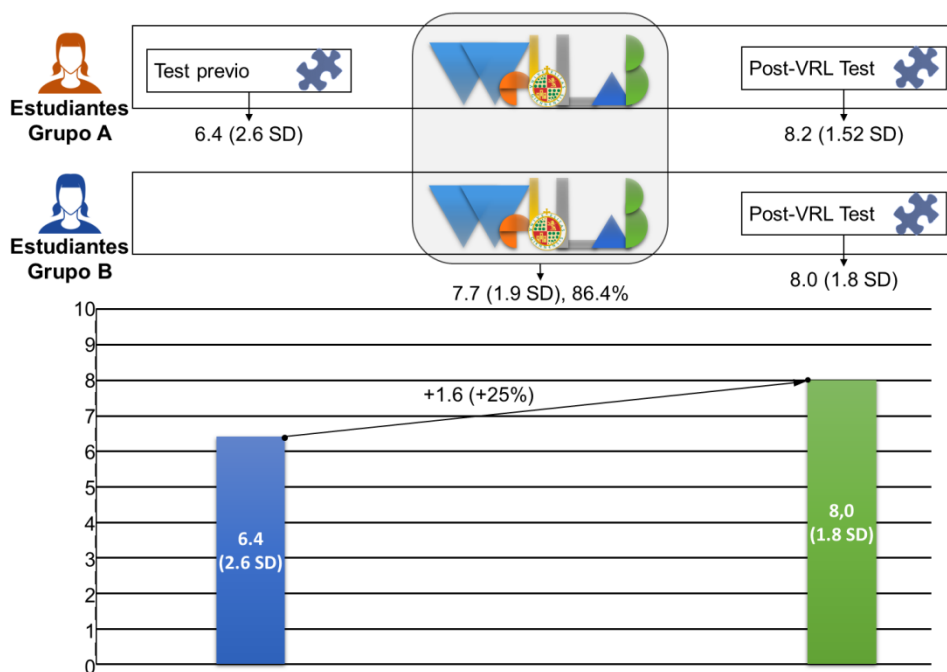


Figura 100. Mejora producida en las calificaciones

Las **páginas 6 a 9** (12 preguntas) se realizaron siguiendo la estructura del paquete SCORM del WebLab, la pagina 6 para el SCO1, la pagina 7 para el SCO2, la pagina 8 para el SCO3 y la pagina 9 para el SCO4. Se mostraban capturas de pantalla de las páginas del SCORM. Con dos propósitos, por un lado para recabar la opinión sobre si los contenidos habían cumplido su función (**Tabla 11**) y, por otro para que valoraran si se trataba de contenidos apropiados usando una escala Likert 1-5. Los resultados, mostrados en la **Figura 101**, son todos bastante positivos situándose todos en el “Notable” al estar al menos en el 3,5 sobre 5 (7 sobre 10).

Tabla 11. Función de los contenidos del WebLab modelado de Motor CC

SCO	Pregunta	Respuestas
1	¿Has entendido cuales eran los objetivos y competencias que podías obtener si trabajabas el WebLab y dónde se enmarcaba su contenido en tus estudios?	Sí 89,1 % No 10,9 %
1	¿Has entendido cuál es la estructura del WebLab?	Sí 93,7 % No 6,3 %
2	¿Has estudiado y comprendido la teoría sobre el laboratorio que se encontraba en la página 2 del WebLab?	Sí 58,6 % Parte 40,6 % No 0,8 %
3	¿Has leído y comprendido los guiones de prácticas que se encuentran en la página 3 del WebLab?	Sí 61,7 % Parte 37,5 % No 0,8 %

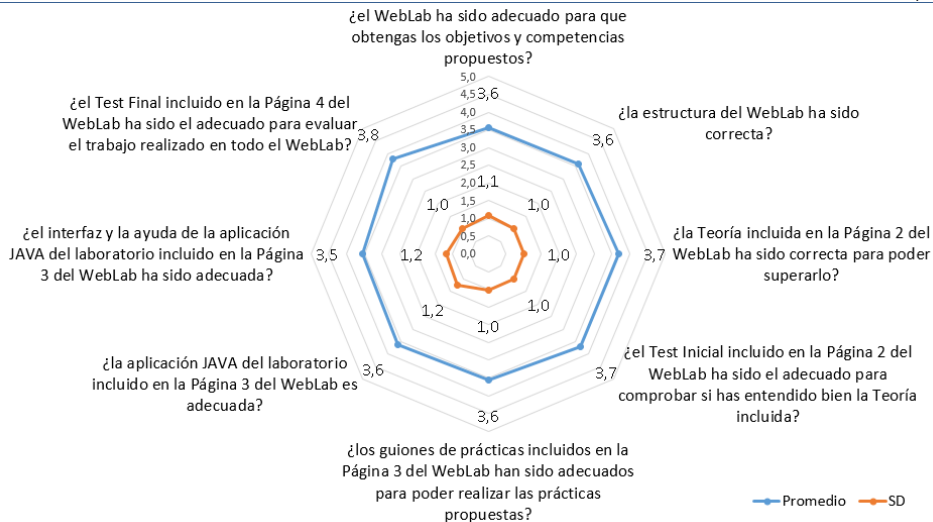


Figura 101. Opinión de alumnado: Adecuación de contenidos del WebLab (SCORM)

La **página 10**, última de la encuesta, presentaba 3 preguntas. La primera se usó para los estudiantes dieran su valoración sobre distintos aspectos del WebLab en general (todo el paquete SCORM, incluyendo el VL) utilizando una escala Likert 1-5. Los aspectos considerados fueron Acceso cómodo, Diseño Web adecuado (colores, tipos de letra, claridad, etc...), Estructura adecuada, Navegación adecuada, Facilidad de uso adecuada, Sensación de inmersión adecuada, Tiempo requerido adecuado, Instrucciones claras y precisas, Fiabilidad del sistema adecuado y, finalmente, Utilidad adecuada (**Figura 102**). Todos los aspectos obtuvieron una valoración positiva por encima de 3,3 sobre 6 (6,6 sobre 10).

La segunda pregunta de la página 10, penúltima de la encuesta, sirvió para obtener una valoración global final del WebLab siguiendo de nuevo una escala Likert 1-5. La **Figura 103** muestra como esta valoración fue bastante positiva obteniendo un 3,67 sobre 5, un “notable” de 7,34 sobre 10, lo que supone un

resultado muy satisfactorio. Para finalizar la encuesta, la pregunta 10 (opcional) decía: “En tu opinión, ¿Qué aspectos del WebLab deberían mejorarse?”. Respondieron 76 estudiantes de los cuales 52 no indicaron que debiera realizarse ninguna mejora. Aunque las respuestas del resto de estudiantes fue muy variada, algunos de ellos se pusieron de acuerdo en señalar algunos aspectos que, a su juicio, necesitaban mejoras como son la interface de usuario (16 estudiantes), la instalación y ejecución de Java (15 estudiantes) y los guiones de prácticas (7 estudiantes). También vale la pena señalar que 13 estudiantes indicaron que les gustaba el WebLab y que este sistema debería ser extendido a otras materias.

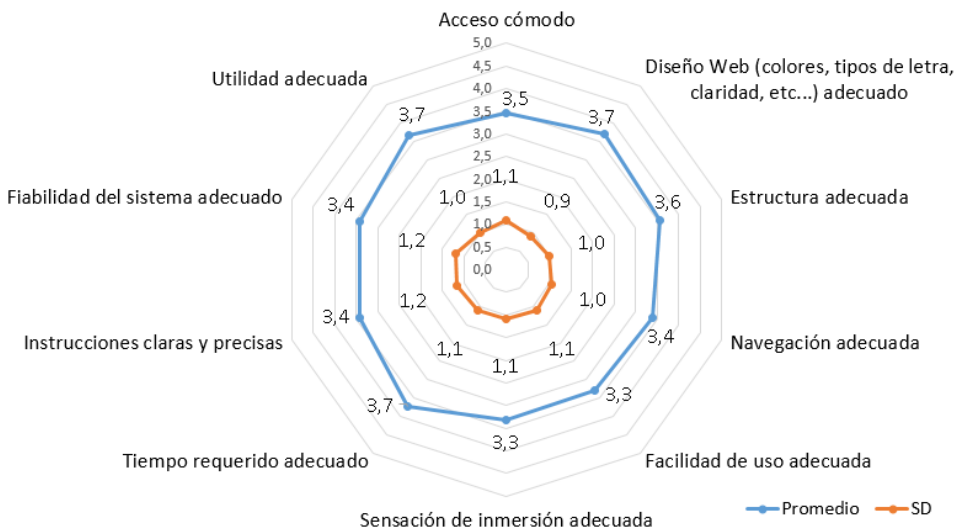


Figura 102. Opinión de alumnado: Valoración de aspectos del WebLab (SCORM)

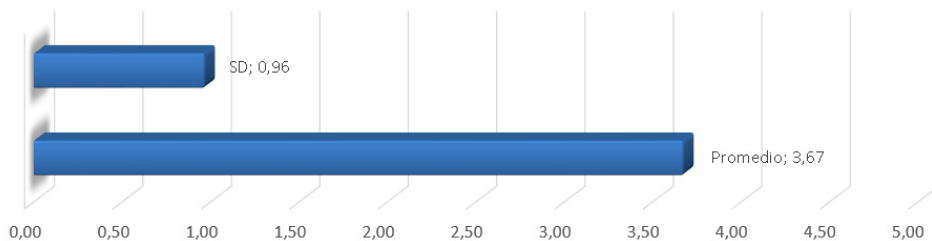


Figura 103. Opinión de alumnado: Valoración general final del WebLab (SCORM)

Otros métodos de evaluación. Aparte del uso de encuestas y entrevistas con el alumnado para conocer su opinión, y el análisis de los resultados obtenidos por

los alumnos existen pocos métodos de evaluación de la efectividad de los mismos. Este laboratorio también se evaluó siguiendo alguna de estas metodologías, la señalada por Rashid y otros [301] (Método 1) y por Amigud y otros [302] (Método 2). Estos métodos se basan en la identificación del laboratorio y los experimentos (de forma administrativa y pedagógica), la relación entre teoría y lab/experimentos, el entorno de aprendizaje, el manual del lab, la evaluación de los estudiantes, usabilidad técnica e información pedagógica. Al aplicar los métodos indicados se obtuvieron los siguientes resultados:

- *Método 1.* El WebLab cumple todos los requisitos de información administrativa. Recibió una valoración de 25 puntos en usabilidad técnica (el máximo). En cuanto a la información educativa del WebLab los objetivos están claramente definidos, los estudiantes son evaluados automáticamente varias veces y se han utilizado 3 de los 4 estilos educativos (todos menos el auditivo).
- *Método 2.* El WebLab incluye dos prácticas (experimentos). La evaluación de los aspectos pedagógicos de los dos experimentos se muestra en la **Tabla 12**. El entorno de aprendizaje del WebLab en este método se basa en la interacción entre profesores, la colaboración entre estudiantes y la evaluación. Las relaciones entre profesores y la colaboración entre estudiantes no son compatibles con este WebLab directamente, pero pueden ser proporcionadas por las múltiples herramientas de comunicación del LMS (miembro del equipo docente-estudiante y estudiante-estudiante). En cuanto a la efectividad del manual de laboratorio esta WebLab llega a 5 de los 7 atributos medidos.

Tabla 12. Evaluación de WebLab: Método 2

#	Nombre de Experimento	Relación T ^a	Nivel de contenido	Tipo de Actividad
1	Prueba de uso de GUI	Independencia total	Complicación baja	Tutorial
2	Identificación de los parámetros K y T	Fuerte acoplamiento	Complicación media	Experimentación y Análisis

Los trabajos relacionados con este laboratorio y los resultados obtenidos han sido utilizados en varias publicaciones internacionales [50], [46].

6.2.3. Control PID de un Motor CC

Este laboratorio fue el primero que se creó siguiendo todos los pasos de la metodología propuesta en el capítulo 5. También se creó como paquete SCORM

2004 integrado en un LMS, la plataforma institucional ILIAS de la UJA, pero añadiendo y utilizando más recursos del LMS que en los laboratorios anteriores. Además fue el primero en el que se realizó una adaptación a la versión SCORM 1.2 para poder probar la portabilidad en otras plataformas (La portabilidad entre plataformas del mismo tipo ya había sido probado con anterioridad). Para ello se usó el sistema Moodle [90] que mantiene la UNED para la compartición de laboratorios online universitarios en UNILabs [27]. El VL realiza la simulación del control PID del motor descrito en el apartado “6.2.2. Modelado de Sistemas Dinámicos: Motor CC”. El WebLab contiene un plan de aprendizaje que incluye una serie de recursos de utilidad docente como teoría de control PID, pruebas de evaluación, un laboratorio virtual de control PID de un motor de corriente continua y experimentos personalizados para cada alumno cuyos resultados son almacenados en el LMS. Este WebLab se ha presentado en el LMS institucional de la Universidad de Jaén a 340 alumnos de la asignatura “Automática Industrial” en el curso 2014-15, la misma asignatura en la que se presentó el laboratorio descrito en el apartado 6.3.2 de este trabajo. Los datos de uso han permitido realizar diversas evaluaciones que demuestran que los alumnos que lo han completado han obtenido un rendimiento excelente en el propio WebLab, han conseguido unos resultados muy superiores al resto de alumnos en la evaluación final de la asignatura y lo han valorado muy positivamente.

Dentro del WebLab hay cuatro páginas en las que los estudiantes pueden encontrar, además de un laboratorio virtual (Virtual Lab, VL) de control PID, recursos útiles relacionados con los experimentos. Los alumnos pueden navegar por estas páginas accediendo a estos recursos a través del itinerario de aprendizaje programado en secuencia (página 1 a 4). Esta secuencia puede modificarse en función de las acciones y calificaciones que obtengan los alumnos. Todas las páginas del WebLab incluyen una evaluación automática, las calificaciones resultantes obtenidas en cada página son almacenadas en el LMS ILIAS para que los tutores del curso puedan acceder a las mismas. El VL es un *applet* de Java desarrollado con el software EJS [63] que está incrustado en la página 3 del WebLab. Este VL se comunica con el LMS utilizando la sub-especificación RTE, responsable de las comunicaciones de SCORM, para obtener la identificación del usuario. Cuando un alumno ejecuta el VL éste le presenta 3 experimentos personalizados basados en su identificación y almacena en el LMS las calificaciones que consigue y algunas notas en forma de comentarios sobre el trabajo realizado por el alumno en cada intento de superación de los experimentos. El alumno puede recuperar y visualizar estos comentarios

durante la ejecución del VL o en ejecuciones posteriores. Los objetivos y competencias globales y específicos seleccionados para este WebLab fueron:

- Conocimiento básico de control automático y su aplicación a la automatización industrial.
- Aprender para qué sirve un regulador (controlador).
- Conocer/comprender reguladores tipo P, PD, PI y PID.
- Sintonizar un controlador para un sistema servo con un motor de corriente continua (motor de CC).
- Realizar el ajuste de parámetros de estos reguladores.
- Consolidar el concepto de tiempo de establecimiento a un % de error de la señal de un sistema.

Estructura del Paquete SCORM

Como ya se ha comentado, está formada por 4 SCOs que cumplen la misma estructura descrita en la **Figura 88** y la misma forma de establecer las comunicaciones SCORM-LMS.

La página 1 (**SCO1**) **Figura 104** contiene una tabla con información genérica que enmarca el laboratorio en el plan de estudios y una zona en la mitad inferior donde describe el WebLab: su estructura en páginas (breve descripción del contenido de cada una), navegación básica y forma de superarlo. Con estar sólo 5 segundos se considera superado y se asigna 4 puntos. Esta información se almacena en el LMS.

La página 2 (**SCO2**) contiene la información teórica necesaria para completar el VL y un test pre-VL con sus 7 preguntas de opción múltiple-respuesta única para comprobar el conocimiento adquirido por el alumno sobre los contenidos teóricos de la página.

Las subsecciones de contenido teórico incluidas en esta página son: 1) Introducción, 2) Tipos de controladores, 3) Controlador P (Incluye un ejemplo de un controlador P para controlar la posición de un motor de CC), 4) Controlador PD (Incluye un ejemplo de un controlador PD de la posición de un motor de CC usando el criterio de cancelación cero-polo), 5) Controlador PI (Incluye un ejemplo de un controlador PI para controlar la velocidad de un motor de CC usando el criterio de cancelación cero-polo) y 6) Controlador PID (**Figura 105**).

Introducción al WebLab (PAGINA 1)
Introducción al WebLab (PAGINA 1)

CONTROL PID DE SISTEMAS DINÁMICOS: MOTOR CC

UNIVERSIDAD:	UNIVERSIDAD DE JAÉN
TITULACIONES:	GRADOS EN INGENIERIA INDUSTRIAL, MECANICA, ELECTRICA, ORGANIZACION INDUSTRIAL Y ELECTRONICA
ASIGNATURA:	AUTOMATICA INDUSTRIAL (NIVEL BASICO)
PLANIFICACION:	2º CURSO // 2º CUATRIMESTRE
TIPO DE EJERCICIO/EXPERIMENTO:	LABORATORIO VIRTUAL
CONTEXTO DEL EJERCICIO:	BLOQUE III. METODOS DE CONTROL TEMA 7. DISEÑO DEL CONTROLADOR PID.
OBJETIVOS Y COMPETENCIAS:	<ul style="list-style-type: none"> • GLOBALES: <ul style="list-style-type: none"> ◦ Conocimiento básico de control automático y su aplicación a la automatización industrial. ◦ Aprender para qué sirve un regulador (controlador). • ESPECIFICOS: <ul style="list-style-type: none"> ◦ Conocer y comprender los reguladores de tipo P, PD, PI y PID. ◦ Aprender técnicas de ajuste de los parámetros de estos reguladores. ◦ Consolidar en concepto de tiempo de establecimiento a un % de error de la señal de un sistema. ◦ Sintonzar un controlador para un sistema servo con un motor de corriente continua (motor CC).
TEORIA:	Enlace a la carpeta de teoría: BLOQUE III. METODOS DE CONTROL

Estructura del WebLab (Laboratorio Web) y Modo de superarlo

Este WebLab se ha creado utilizando el formato SCORM. La siguiente figura muestra la organización básica que se utiliza en el departamento para diseñar Laboratorios Web en un paquete SCORM:

El Contenido se estructura en 4 secciones (páginas web):

- **PAGINA 1. Introducción (Esta Página):** Proporciona información general sobre el WebLab y se enmarca dentro de los estudios de grado.
- **PAGINA 2. Teoría (Página siguiente):** Proporciona información teórica relacionada con el control de los sistemas dinámicos, especialmente sobre el control PID de un sistema de Motor CC. También contiene un test inicial que trata de asegurar que se han asimilado los conceptos técnicos explicados antes de acceder al laboratorio en sí con el objetivo de que no pueda acceder al laboratorio alguien que no ha adquirido los conocimientos necesarios para usarlo y aprovecharlo.
- **PAGINA 3. VRL:** (Para acceder a esta sección es necesario haber superado el test inicial de la sección PAGINA 2. Teoría). Proporciona los guiones de las prácticas que se deben realizar con el laboratorio y el propio VRL (Virtual Remote Lab, Laboratorio Virtual o Remoto) en el que se podrá trabajar con el fin de superar las prácticas propuestas. El VRL realiza una evaluación automática del trabajo realizado por cada alumno.
- **PAGINA 4. Evaluación (Última página):** (Para acceder a esta sección es necesario haber superado la evaluación automática de la sección PAGINA 3 VRL). Muestra un test final relacionado con todo el trabajo e información previa incluida en el WebLab que los estudiantes deben superar.

Para superar el WebLab se debe llegar hasta la sección PAGINA 4. Evaluación y superar el test de evaluación final.

Figura 104. Captura de pantalla de página 1 del WebLab Control PID de Motor CC: Introducción

Teoría del WebLab (PAGINA 2)
Teoría del WebLab (PAGINA 2)

CONTROL PID DE SISTEMAS DINÁMICOS: MOTOR CC

Introducción

Un controlador automático compara el valor real de la salida de una planta con la entrada de referencia (el valor deseado), determina el error entre ambas señales ($e(t) = \text{referencia}(t) - \text{salida}(t)$) y produce una señal de control que reduce la diferencia a cero o a un valor pequeño. **La acción de control es la forma con la cual el controlador automático produce la señal de control. La figura siguiente muestra el diagrama de bloques de un sistema de control industrial que consiste en un controlador automático, un actuador, una planta y un sensor (elemento de medición).**

Figura 1. Esquema de Bloques de un Controlador.

Por un lado, la salida del controlador se alimenta a un actuador, como por ejemplo una válvula. Mientras que el sensor, o elemento de medición, es un dispositivo que convierte una magnitud en una señal, por ejemplo un desplazamiento en metros pasaría a una señal en voltios, mA o % del alcance del sensor.

Tipos de Controladores

Los controladores industriales más importantes se clasifican en función de las acciones de control que implementan del siguiente modo:

- Controladores On-off.
- Controladores Proporcionales.
- Controladores Proporcionales-Derivativos (PD).
- Controladores Proporcionales-Integrales (PI).
- Controladores Proporcionales-Integrales-Derivativos (PID).

Controlador Proporcional

Figura 105. Captura de pantalla parcial de página 2 del WebLab de PID de Motor CC (Teoría)

Las 7 preguntas del test tratan conceptos explicados en la teoría, mostrada en la misma página, que el estudiante debe dominar (**Figura 106**):

1. ¿Saben que es $e(t)$?
2. ¿Saben calcular las constantes de un controlador PID a partir de su ley de control?
3. ¿Saben distinguir entre los diferentes tipos de control (P, PD, PI, PID)?
4. ¿Saben cuál es la ecuación que define el tiempo de establecimiento al 5% del error en un sistema de primer orden?
5. ¿Saben qué tipo de control PID opera teniendo en cuenta sólo el pasado y el presente?
6. ¿Saben qué tipo de acción de control PID opera como un offset autoajutable?
7. ¿Saben qué tipo de acción de control PID hay que añadir para mejorar la respuesta en el transitorio?

Al terminar el test se almacena en el LMS la puntuación obtenida (máximo 28 puntos), el estado de superación y el de terminación, y si se supera (más de 14 puntos).

Test previo al Laboratorio (Evaluación Inicial)

MUY IMPORTANTE, LEER ANTES DE EMPEZAR A RESPONDER EL TEST

- Presta atención a las siguientes preguntas, lee tranquilamente el enunciado y contesta con tranquilidad.
- El objetivo de las mismas es saber si has asimilado correctamente los conocimientos que debes obtener para acceder al laboratorio.
- Fíjate en que si respondes correctamente recibirás 4 puntos por cada pregunta, pero si lo haces de forma incorrecta te restará un punto.
- También tienes la opción de responder señalando la opción "No sabe/No contesta", en cuyo caso no se sumará ni se restará ningún punto.
- Para poder superar el test es necesario que obtengas al menos 14 puntos (máximo 28).
- Sólo se tiene una oportunidad para contestar a cada una de las preguntas, una vez pulses el botón "Contestar" ya no podrás volver a hacerlo.
- Si has señalado una opción pero te has arrepentido y deseas dejar la respuesta en blanco pulsa sobre "Reiniciar" o la opción "No sabe/No contesta".
- Al final de la página debes pulsar sobre el botón "Contestar Todo y Enviar" para mostrar tus resultados finales y enviarlos al LMS.
- Debes contestar a todas las preguntas.
- No actualices el contenido de la página en el navegador, si lo haces se cargará de nuevo el PAGINA 1-Introducción y deberás volver a empezar el WebLab.

Pregunta 1

PREGUNTA 1: Linealidad del Sistema.
(Bien:4 puntos - Error:-1 punto)

¿Qué es $e(t)$?

$e(t)$ es la señal de referencia (valor deseado).

$e(t)$ es la diferencia que existe entre la variable controlada y la señal de referencia.

$e(t)$ es la diferencia que existe entre la variable controlada y la entrada del sistema a controlar.

$e(t)$ es la entrada del sistema a controlar.

No sabe/No contesta.

Enviar Reiniciar

Resultado de pregunta 1

¡Estupendo!, sigue así ==> Puntos obtenidos en esta pregunta:4 == Total de Puntos en el test:4

Figura 106. Captura de pantalla parcial de página 2 del WebLab de PID de Motor CC (Test pre-VL)

La Página 3 (SCO3) incluye los elementos necesarios para el trabajo experimental del alumno. Nada más entrar y la misma y cargarse el *applet* puede aparecer una ventana emergente si se ha ejecutado antes el VL indicando la puntuación alcanzada (**Figura 107**).

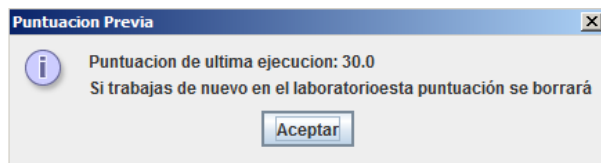


Figura 107. Ventana emergente de aviso de puntuación anterior en VL

En su parte superior (**Figura 108**) hay una introducción y un breve manual del software del VL que muestra la GUI del mismo incluyendo imágenes descriptivas. A continuación, se encuentran los guiones de prácticas de los 3 experimentos que los alumnos deben realizar en el VL (**Figura 109**).

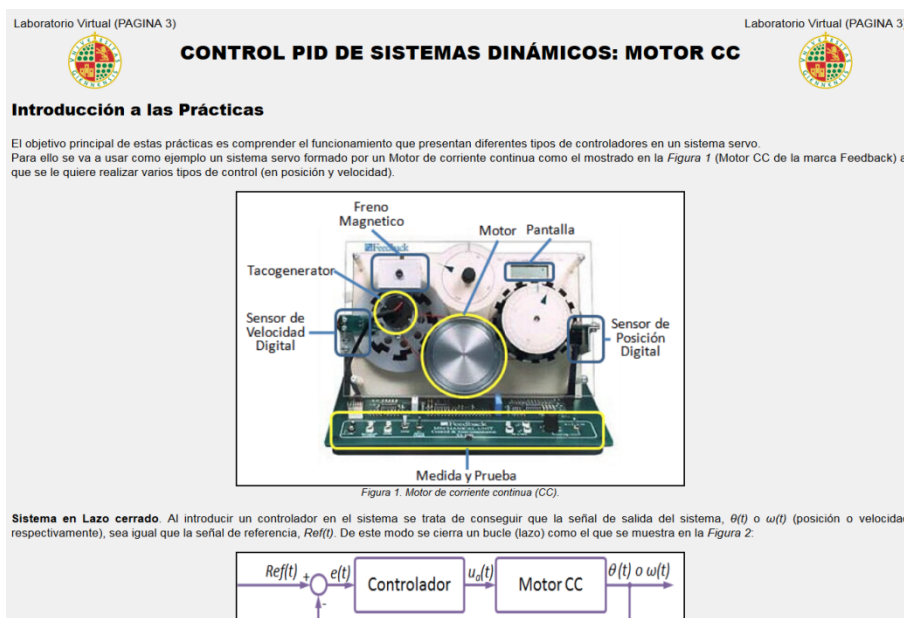


Figura 108. Captura de pantalla parcial de página 3 del WebLab de PID de Motor CC (Intro)

En esta zona se ha utilizado un código de colores que facilita al estudiante el entendimiento de las prácticas y acciones a realizar (**Figura 109**):

- **Blanco:** Explicaciones normales y acciones paso a paso que el alumno debe realizar en el VL.

- **Amarillo:** Observaciones que debe tener en cuenta el alumno tras realizar las acciones que se le piden.
- **Naranja:** Información importante y explicaciones sobre la evaluación de los apartados/experimentos.

Acciones a realizar en el Laboratorio Virtual

1. Comprobar que la entrada al sistema (señal de referencia) tiene el valor 10. En caso contrario introducir Ref(t)=10 (pulsar el botón intro/enter del teclado).
2. Comprobar que la constante de control proporcional es 1: $K_P=1$. (Si $K_P=0$ entonces la entrada al motor CC sería 0 y la salida también).
3. Pulsar el botón de ejecución
4. Observar las gráficas que muestran el comportamiento del sistema.
5. Pulsar el botón "Limpiar" que está justo a la derecha de K_P para borrar las gráficas, ajustar los ejes y reiniciar valores antes de introducir un nuevo valor de K_P .
6. Cambiar el valor de K_P aumentando su valor (P.e. 2 - 4 - 8 - 10 - 20 - 50 ... pulsar el botón intro/enter del teclado cada vez que se modifique el valor para que tenga efecto en la aplicación), y repetir los pasos dados hasta que quede claro el comportamiento del sistema.

Observaciones en el Laboratorio Virtual

Al aumentar el valor de K_P se consigue que la señal de salida $\theta(t)$ alcance antes en un régimen permanente el valor de la señal de Referencia Ref(t).
 Cuando el valor de K_P crece llega un momento en el que se produce sobrepasamiento (sobreimpulso) y oscilaciones en la señal de salida como se puede ver en la siguiente gráfica (Esto es debido a que el sistema en lazo cerrado se comporta como un sistema de segundo orden subamortiguado):

Si se desea que la variable controlada (posición) alcance el valor de la señal de referencia lo antes posible sin sobreesalaciones en el régimen transitorio el sistema ha de ser críticamente amortiguado.

Evaluación del Apartado 1

Calcular el valor de K_P con el que el sistema está críticamente amortiguado cuando la entrada de referencia es una señal escalón de amplitud 10.
 Introducir el valor calculado (K_{Pca}) y pulsar el botón de comprobación para validar el resultado.

Figura 109. Captura de pantalla parcial de página 3 del WebLab de PID de Motor CC (Guiones)

Finalmente, en la zona inferior de la página web se encuentra incrustado el *applet* Java que constituye la aplicación del VL con la que el alumno debe interactuar para hacer los 3 experimentos. Los experimentos se implementaron creando un laboratorio virtual (VL) con el software EJS para obtener un *applet* de VL. El *applet* fue programado en JAVA con una interfaz gráfica de usuario (GUI) sencilla y fácil de utilizar, se importó el paquete Java *scormRTE* (descrito en el apartado 0) para poder crear un objeto de tipo *scormRTE* que facilita las comunicaciones VL-LMS a través de la API de SCORM. Esto es muy importante ya que cuando el VL se ejecuta como *applet* incrustado en una página Web, una de las primeras cosas que hace es iniciar una comunicación con el LMS para obtener la identificación del estudiante. De este modo, el funcionamiento del VL se personaliza para cada alumno:

- La simulación de un motor de CC con la que debe trabajar cada estudiante es diferente para cada uno: Los parámetros K_m (ganancia

estática del sistema) y T_m (constante de tiempo del sistema) se calculan en función de la identificación del estudiante dada por el LMS al VL y así cada estudiante trabaja con un sistema personalizado.

- Los resultados de las acciones realizadas por cada estudiante en cada experimento son almacenados en el LMS como comentarios SCORM indicando fecha y hora. Más tarde, los estudiantes pueden revisar estos datos privadamente.
- El trabajo realizado por los estudiantes en cada uno de los 3 experimentos es evaluado de forma automática por el VL y la calificación que obtiene es almacenada en el LMS. Los tutores pueden comprobar estos datos.

La GUI del VL cambia con cada experimento aunque todos ellos se han realizado usando los modelos de primer orden simplificado del motor de la **Figura 110**.

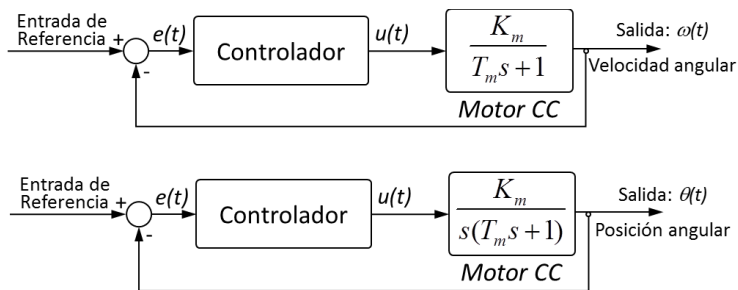


Figura 110. Modelos del Sistema de control del WebLab de PID de Motor CC

A continuación se describen cada uno de los experimentos y se muestra la GUI implementada para cada uno:

1. Control de la posición angular de un motor de CC usando un controlador P: Los alumnos deben observar el comportamiento del sistema cuando se cambia el valor de la constante proporcional (K_p) y calcular el valor de K_p que hace que el sistema sea críticamente amortiguado. La GUI (**Figura 111**) es similar a la mostrada en el laboratorio de identificación del modelo (apartado 6.3.2), se ha actualizado las opciones de control de ejes en los gráficos y el panel de prácticas añadiendo el esquema de control.
2. Control de la posición angular de un Motor de CC usando un control PD con restricciones temporales: Los alumnos deben reducir el orden de la ecuación del sistema usando el criterio de cancelación cero-polo [303] y

calcular los valores de las acciones Proporcional e Integral del controlador para que la salida del sistema alcance el régimen permanente en un tiempo acorde a la especificación dada para el tiempo de establecimiento al 2% o al 5% del error. El VL determina los valores de K_m y T_m en función de la identificación del alumno y el valor de la forma de cálculo del tiempo de establecimiento (2% o 5%) de forma aleatorio y la GUI, **Figura 112**, los muestra para que el estudiante pueda realizar sus cálculos con todos los datos. Los alumnos tienen 3 intentos para responder correctamente, sino el VL los devuelve a la práctica 1.

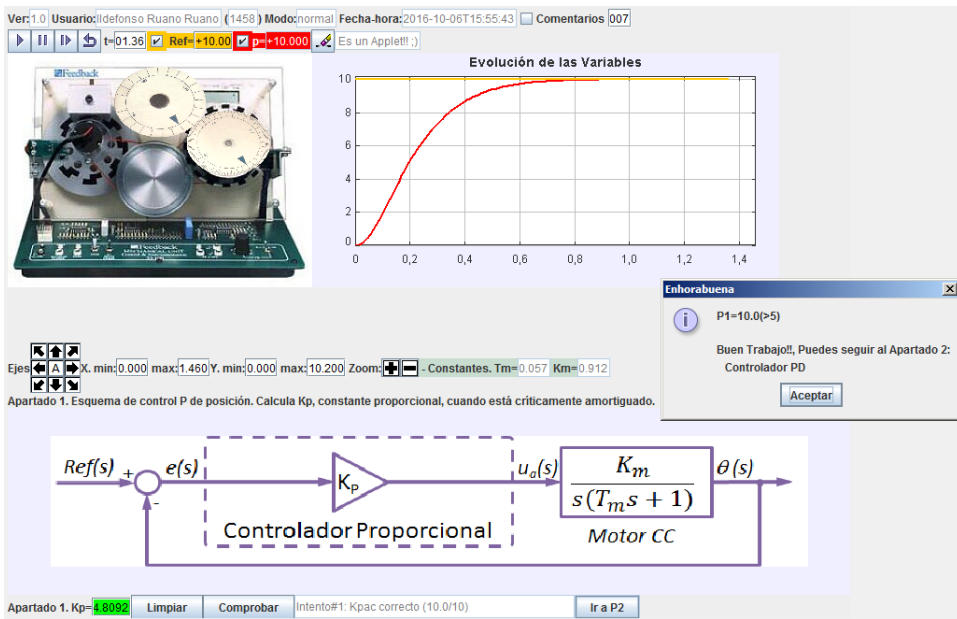


Figura 111. applet Java del VL con práctica 1ª de control P y ventana emergente de paso a la de Control PD

- Control de la velocidad angular de un motor de CC usando un controlador PI. En este caso la dinámica pasa a ser un sistema de primer orden, ya que al ser la velocidad la derivada de la posición, el polo en el origen para esta nueva dinámica desaparece. Los estudiantes deben calcular los valores de K_p y K_i con los que el sistema responde a una entrada escalón con un tiempo de establecimiento al 2% o al 5% del error. Al igual que en el experimento anterior, el VL calcula los parámetros K_m y T_m y el tipo de tiempo de establecimiento y la GUI

(Figura 113) los muestra al estudiante. Los alumnos tienen 3 intentos para responder correctamente, sino el VL los devuelve a la práctica 2.

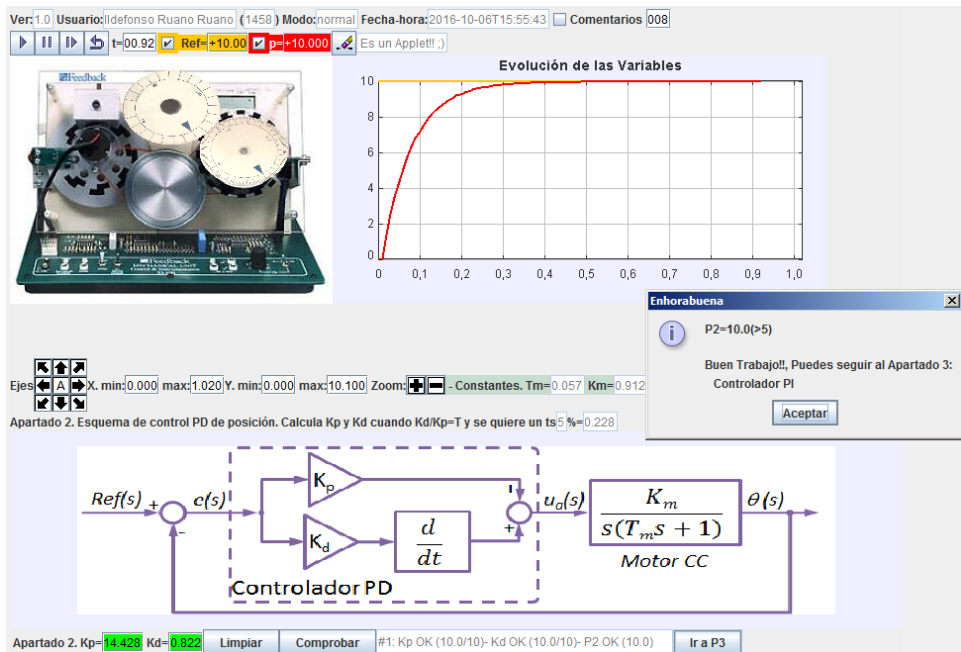


Figura 112. applet Java del VL con práctica 2ª de control PD y ventana emergente de paso a la de Control PI

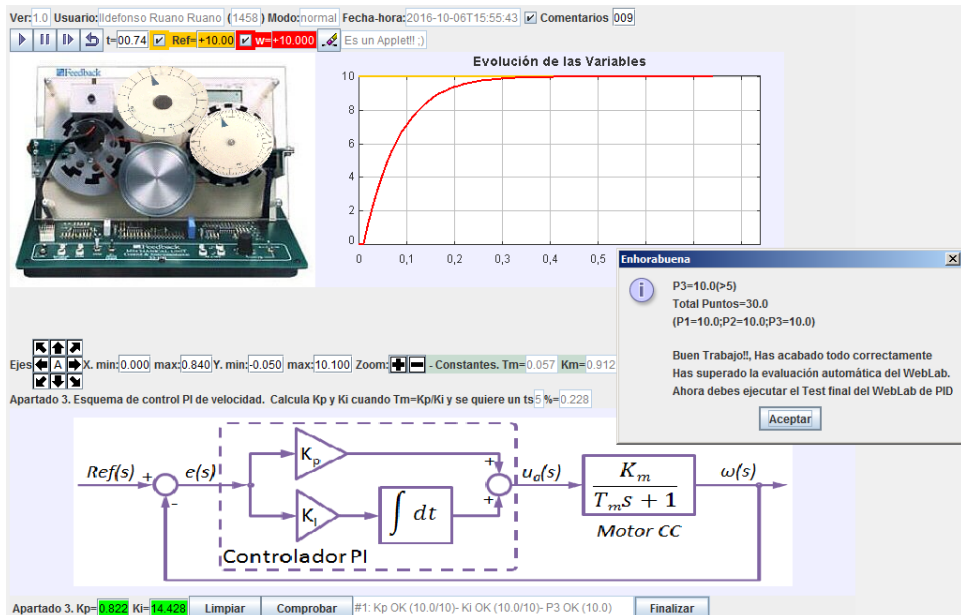
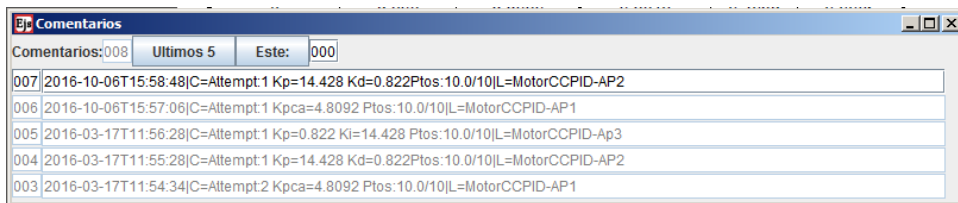


Figura 113. applet Java del VL con práctica 3ª de control PI y ventana emergente de paso al SCO4

El VL almacena automáticamente en el LMS comentarios que describen los intentos realizados por cada estudiante en cada experimento y el resultado conseguido. Estos comentarios pueden ser recuperados por los estudiantes activando la ventana de comentarios (**Figura 114**) en la GUI del VL. Cada experimento tiene un valor de 10 puntos, una vez acabado el trabajo del estudiante en el VL, al pulsar el botón “Terminar” provoca que el VL almacene en el LMS el estado de superación y terminación del lab y la calificación obtenida. En caso de que se haya superado se pasará a la página 4.



ID	Timestamp	Attempt	Kp	Ki	Kd	Puntos	Lab
007	2016-10-06T15:58:48	Attempt 1	14.428	0.822	0.822	10.0/10	MotorCCPID-AP2
006	2016-10-06T15:57:06	Attempt 1	4.8092			10.0/10	MotorCCPID-AP1
005	2016-03-17T11:56:28	Attempt 1	0.822	14.428		10.0/10	MotorCCPID-AP3
004	2016-03-17T11:55:28	Attempt 1	14.428	0.822	0.822	10.0/10	MotorCCPID-AP2
003	2016-03-17T11:54:34	Attempt 2	4.8092			10.0/10	MotorCCPID-AP1

Figura 114. Ventana de comentarios del VL de Control PID de Motor CC

La página 4 (**SCO4**) sólo incluye el test post-VL que está formado por 7 preguntas de opción múltiple,-respuesta única sobre lo que el alumno ha trabajado (**Figura 115**). Su propósito es comprobar que el alumno domina los conceptos y ha conseguido las habilidades que debe proporcionar el WebLab:

1. ¿Pueden distinguir el comportamiento dinámico de un sistema sub-amortiguado?
2. ¿Conocen la definición correcta de “tiempo de establecimiento al X% del error”?
3. ¿Pueden calcular la función de transferencia de un sistema con control PI dado en lazo cerrado?
4. ¿Pueden calcular la función de transferencia de un sistema con control PD dado en lazo abierto?
5. ¿Pueden identificar los parámetros de un PID a partir de una ley de control?
6. ¿Saben qué tipo de control (P, PI, PD, PID) es el más adecuado y simple para conseguir que un sistema dado cumpla ciertas especificaciones conociendo su función de transferencia? (2 preguntas de este tipo)

La programación de la secuenciación y navegación entre SCOs que se ha realizado en el propio paquete SCORM coincide con la mostrada en la **Figura 97** para el laboratorio de modelado del Motor CC, aunque la secuenciación entre

prácticas programada en el interior del código java del VL incluye las 3 prácticas en vez de las 2 mostradas en el otro laboratorio.

Evaluación (Página4)
Evaluación (Página4)

MODELADO DE SISTEMAS DINÁMICOS: MOTOR CC

Evaluación final de las Prácticas del WebLab

MUY IMPORTANTE, LEER ANTES DE EMPEZAR A RESPONDER EL TEST

- Presta atención a las siguientes preguntas, lee tranquilamente el enunciado y contesta con tranquilidad.
- El objetivo de las mismas es saber si has asimilado correctamente los conocimientos que debes obtener.
- Fíjate en que si respondes correctamente recibirás 4 puntos por cada pregunta, pero si lo haces de forma incorrecta te restará un punto.
También puedes seleccionar la opción No Sabe/No Contesta", que no suma ni resta puntos.
- Para poder superar el test es necesario que obtengas al menos 14 puntos (máximo 28).
- Sólo se tiene una oportunidad para contestar a cada una de las preguntas, una vez pulses el botón "Enviar" ya no podrás volver a hacerlo.
- Si has señalado una opción pero te has arrepentido y deseas dejar la respuesta en blanco pulsa sobre "Reiniciar".
- Al final de la página debes pulsar sobre el botón "Contestar Todo y Enviar" para mostrar tus resultados finales y enviarlos al LMS.
- Se considerará que has contestado al test si al menos has contestado a una de las preguntas.
- No refresques el contenido de la página en el navegador, si lo haces se cargará la Página1-Introducción y deberás volver a empezar todo el WebLab.

Pregunta 1

PREGUNTA 1: Sistema Subamortiguado.
(Bien:4 puntos - Error:-1 puntos)

Las siguientes imágenes se corresponden con la señal de salida de un mismo sistema controlado mediante un controlador proporcional en el que se ha estado cambiando el valor de la constante proporcional K_P con el fin de obtener distintos comportamientos.
Indica cual de las siguientes gráficas muestra la forma que debe mostrar la salida del **sistema subamortiguado**.

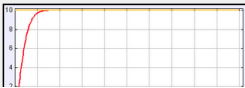


Figura 115. Captura de pantalla parcial de página 4 del WebLab de PID de Motor CC: Evaluación

Presentación del Laboratorio online a los estudiantes

Los 340 alumnos de la asignatura “Automática Industrial” de la Universidad de Jaén en el curso 2014-15 tuvieron acceso a este WebLab. 339 de ellos estaban registrados en el espacio virtual de la asignatura en ILIAS. Los 339 alumnos registrados en ILIAS fueron divididos en dos grupos virtuales (grupo experimental y grupo de control) en el LMS con el fin de analizar el rendimiento y la efectividad del laboratorio en dos escenarios diferentes. Para ello se les asignaron unos espacios privados en donde se incluyeron una serie de recursos compartidos para cada grupo:

- Grupo experimental (GE):** 169 estudiantes. Este grupo ha accedido a una experiencia de WebLab completa. Para medir correctamente la diferencia de conocimiento adquirido por los estudiantes, el test post-VL se retiró del paquete SCORM del WebLab y se incluyó en la carpeta del grupo como un test ILIAS con las mismas preguntas y características.

En la parte superior del espacio virtual se incluyeron algunas instrucciones, a continuación se incluyeron todos los recursos obligatorios del itinerario docente en orden de uso (**Figura 116**).



Figura 116. Captura de pantalla de grupo en ILIAS para WebLab de 3 páginas

- *Test previo a WebLab de PID de Motor de CC.* Test ILIAS idéntico al test Post-VL que se usa para medir el conocimiento de los alumnos antes de entrar al WebLab.
- *Pasos Previos a Ejecución de WebLab.* Este es el documento de requisitos del VL que se ha implementado como un módulo ILIAS.
- *WebLab PID Motor de CC.* Este es el paquete SCORM del WebLab pero sólo con las 3 primeras páginas (sin la página 4).
- *Test Final de WebLab de PID de Motor de CC.* Test Post-VL incluido en la página 4 del WebLab original.
- *Encuesta WebLab PID Motor de CC.* Encuesta utilizada para conocer la opinión de los estudiantes sobre el WebLab.
- *Foro de dudas del WebLab.* Foro usado para ayudar a los estudiantes.
- *Vídeo con la solución al VL.* Este recurso no se ofreció a los estudiantes antes de que se acabara el plazo de uso de la propuesta. Es un módulo ILIAS donde se ha incluido un vídeo creado con el software de videoconferencias *Adobe Connect* en el que se muestra una presentación explicativa de la solución y un ejemplo de uso del WebLab (página 3).
- **Grupo de Control (GC):** 170 estudiantes. Los miembros de este grupo han tenido acceso a una experiencia del WebLab limitada. El formato y los recursos presentados a este grupo eran exactamente los mismos que lo que se ha descrito para el grupo experimental, sin embargo, el recurso WebLab PID Motor de CC al que accedieron era una versión reducida en la que sólo estaba la página 3 del VL (sin las páginas 1, 2 ni 4).

La participación en esta propuesta fue voluntaria y totalmente opcional por lo que hubo muchos alumnos que, aun teniendo la oportunidad de acceder a los recursos de su grupo correspondiente, no han participado. La superación del WebLab y las calificaciones obtenidas no han sido tenidas en cuenta en la nota del alumno en la asignatura.

El motivo de crear dos grupos de alumnos con diferentes escenarios es múltiple. Se ha tratado de demostrar dos hipótesis:

- Un WebLab basada en una estructura de varias páginas que incluyen recursos relacionados con el VL (sección 3.1.) ayuda a completar los

experimentos del laboratorio y, por tanto, a superar el VL. Los alumnos del grupo experimental (3P) obtendrán mejores resultados que los de control (1P).

- La superación de los experimentos del laboratorio (VL) en cualquiera de los 2 escenarios ayuda a superar la propia asignatura.

Datos de uso, Resultados y Conclusiones

Se obtuvieron los siguientes datos de uso y resultados obtenidos por los alumnos de los GC y GE:

- Los datos promedio temporales no son muy fiables: la desviación estándar es muy elevada. Debe ser debido a que hay alumnos que entran en las páginas y después de trabajar no las cierran y las dejan contando un tiempo que no es real, de ahí la desviación tan grande.
- Acceso inicial escaso: Realizan el test previo (primer recurso) 71 alumnos de 170 en el GC (41,76%) y 69 de 169 en el GE (40,82%).
- Nivel de conocimiento inicial escaso en ambos grupos: Nota media obtenida en el test previo de 4,16 (SD 0,85) en el grupo de control y 4,49 (SD 2,15) en el GE. Alumnos que superan el test:
 - GC: 22 (30,99% de los que acceden al test) con nota media de 6,36 (SD 0,85).
 - GE: 29 (42,03% de los que acceden al test) con nota media de 6,60 (SD 1,17).
- Acceso al WebLab muy escaso en ambos grupos: Entran 50 alumnos de 170 en el GC (29,41%, 70,4% de los que hicieron el test previo) y 57 de 169 en el GE (33,73%, 82,6% de los que hicieron el test previo).
- Mejores resultados en el laboratorio virtual de los alumnos del GE. Lo superan más alumnos y con mejor nota:
 - GC: 22 (44% de los que entran en el WebLab, 31% de los que hicieron el test inicial) con nota media de 6,36 (SD 0,85).
 - GE: 41 (71,93% de los que entran en el WebLab, 59,4% de los que hicieron el test inicial) con nota media de 9,18 (SD 0,68).
- Buen nivel de conocimiento alcanzado por los alumnos que llegaron a realizar el test final en ambos grupos, pero con mejores resultados en el GE: Nota media obtenida en el test final de 6,88 (SD 2,32) en el grupo de control y 7,35 (SD 2,49) en el GE. Alumnos que superan el test:
 - GC: 16 (72,73% de los que acceden al test) con nota media de 7,95 (SD 1,73).
 - GE: 32 (78,05% de los que acceden al test) con nota media de 8,39 (SD 1,65).
- Aumenta al menos un 20% el nivel de conocimiento final alcanzado por los alumnos que llegaron a realizar el test final en ambos grupos:

- GC: Mejora Promedio de 2,01 (SD 3,29) (Mejora de 20,13%).
- GE: Mejora Promedio de 2,23 (SD 2,8) (Mejora de 22,30%).
- El número de alumnos que completa toda la propuesta didáctica superando el test final es bajo aunque el número de alumnos fue muy superior en el GE respecto al grupo de control:
 - GC: 16 (32% de los que acceden al WebLab, 22,5% de los que accedieron al test previo).
 - GE: 32 (57,14% de los que acceden al WebLab, 46,4% de los que accedieron al test previo).

Estos datos se muestran gráficamente para el grupo de control y experimental en la **Figura 117** y la **Figura 118** respectivamente.

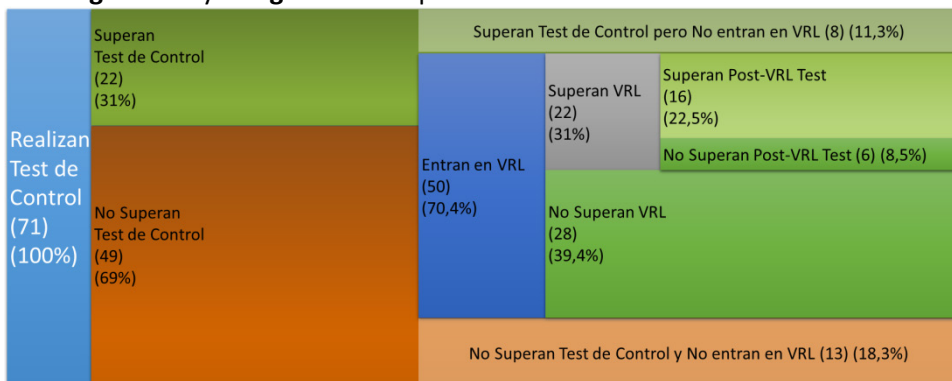


Figura 117. Resumen de resultados del GC

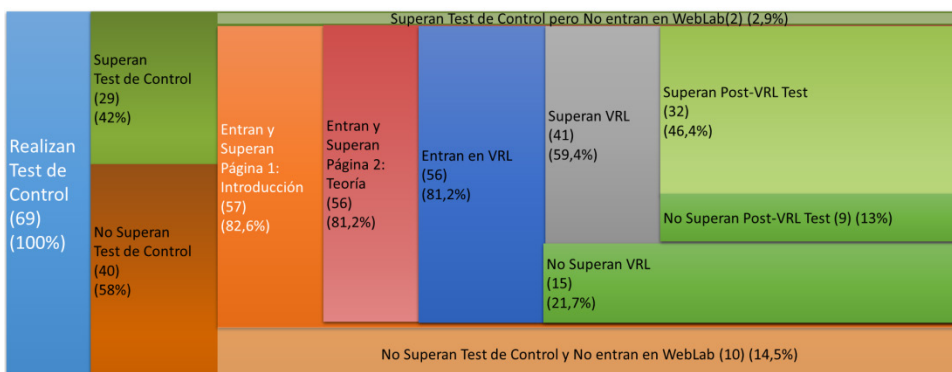


Figura 118. Resumen de resultados del GE

Como resumen de conclusiones sobre los datos de uso del WebLab se puede decir lo siguiente:

- Los tiempos medio de las medidas de tiempo no son muy fiables.
- La participación inicial es baja.
- Los resultados obtenidos en el VL del grupo experimental son mejores que en el grupo de control.

- El número de alumnos que ha completado y superado la propuesta de aprendizaje es bajo.
- Las calificaciones obtenidas en el WebLab por los alumnos que completaron satisfactoriamente todos los recursos son altas.

Por otro lado, también se analizaron los resultados que obtuvieron los alumnos en la evaluación final del curso. A continuación se resumen las conclusiones alcanzadas y que la **Figura 119** muestra gráficamente:

- La asistencia al examen de la primera convocatoria de la asignatura no ha sido muy alta, 197 de 340 alumnos (57,94%), y los resultados no han sido muy positivos ya que sólo aprobaron 61 (17,94%) con una nota media de 6,52 (SD 0,76).
- Los alumnos que trabajaron algún recurso asociado al WebLab (al menos el test previo) obtuvieron mejores resultados que la media del curso:
 - GC: 23 alumnos superaron el examen (38,33% de los que se presentaron) con una nota media de 6,4 (SD 2,18).
 - GE: 20 alumnos superaron el examen (37,04% de los que se presentaron) con una nota media de 6,73 (SD 0,92).
- Los alumnos que superaron el WebLab y el test final de evaluación obtuvieron los mejores resultados:
 - GC: 9 alumnos superaron el examen (64,29% de los que se presentaron) con una nota media de 6,84 (SD 0,47).
 - GE: 14 alumnos superaron el examen (50% de los que se presentaron) con una nota media de 6,88 (SD 1,02).

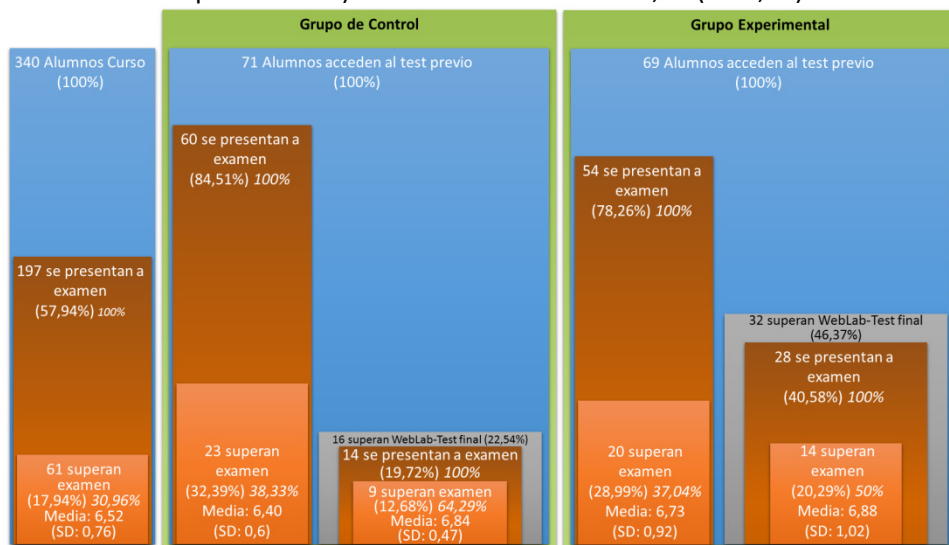


Figura 119. Resumen de resultados GC-GE y Examen final

Evaluación de la efectividad del WebLab

La evaluación de un laboratorio es un proceso complejo que puede ser realizado aplicando diferentes metodologías en función de los aspectos del laboratorio que se quieran evaluar. En la literatura especializada se puede encontrar una amplia variedad [304], [301], [302], [138]. En este laboratorio se muestran 3 evaluaciones, fruto de la aplicación de tres metodologías que se han realizado basadas en el resultado que han obtenido los alumnos en el propio WebLab, en los efectos del WebLab en el examen final de la asignatura y en la satisfacción del alumnado que ha completado la propuesta docente.

Medida del incremento de conocimiento adquirido. A efectos de evaluación se extrajo la información incluida en la página 4 del modelo de WebLab original, (test final de evaluación de conocimientos adquiridos), que se ofreció como un test independiente ILIAS. Para poder medir el efecto del WebLab en el conocimiento adquirido por los alumnos que participaron en el WebLab se compararon los resultados que obtuvieron en el mismo test antes y después de superar satisfactoriamente el WebLab. La diferencia entre la nota media obtenida entre estos tests (previo y final) es un indicativo de la efectividad del laboratorio. En los dos grupos a los que se les presentó el VL (por si sólo o en la modalidad de 3 páginas) se produjo un incremento en la nota media aproximado al 30% (29,9% en el GE y 31,3% en el GC) que puede entenderse como un incremento del conocimiento adquirido por los alumnos que lo superaron en las dos modalidades. Esto confirma que el WebLab y los recursos asociados han producido un incremento en el nivel de conocimiento de los estudiantes que siguieron las dos propuestas de aprendizaje: GC (WebLab sólo con VL en 1 página) y GE (WebLab con VL en 3 páginas) y que los WebLabs son efectivos.

Puede sorprender que en el GC los resultados fueran ligeramente mejores, 31,3% frente a 29,9%, diferencia de 0,14 puntos, 1,4%. Sin embargo si se miran en su conjunto y se tiene en cuenta el número de alumnos que superaron totalmente cada propuesta (32 alumnos en GE y 16 en GC, justo la mitad) se torna esta idea. En el GE el 56,14% de los alumnos que entraron en el WebLab completaron la propuesta superando el test final (32 de 57) mientras que en el GC sólo fue sólo el 32% (16 de 50), una diferencia de 24,14%.

El VL es efectivo porque los estudiantes que lo han superado han incrementado su conocimiento cerca de un 30% (diferencia de 1,4% mejor en el GC). Pero la efectividad del modelo de WebLab basado en el paquete SCORM de 4 páginas es mucho mayor que la efectividad del VL por sí sólo (WebLab de 1 página) ya que la diferencia de porcentajes de alumnos que lo superan es un 24,14%.

WebLab y Evaluación final de la asignatura. Los datos y resultados obtenidos en el examen final de la asignatura por los estudiantes que completaron satisfactoriamente el WebLab, en cualquiera de sus modalidades, son mucho mejores que los obtenidos como media de los estudiantes de la asignatura (incluyendo a todos los matriculados, participaron en la experiencia o no):

- El porcentaje de alumnos que participaron en el examen final es mucho mayor: 28 de 32 en el GE y 14 de 16 en el GC (87,5% en ambos casos) frente a los 197 de 340 (57,94%) del curso entero.
- El porcentaje de alumnos que superaron el examen final es también mucho mayor: 14 de 28 (50%) en el GE y 9 de 14 (64,29%) en el GC frente a los 61 de 197 (30,96%) del curso entero, una diferencia de 19,04% (GE) y 33,33% (GC).
- La nota media obtenida por los alumnos que superaron el WebLab es ligeramente mayor: 6,88 (SD 1,02) en el GE y 6,84 (SD 0,47) en el GC frente a 6,52 (SD 0,76).

Los alumnos que completaron el WebLab se presentaron al examen final en un porcentaje mayor (diferencia entre porcentajes de 29,56%), además superaron el examen con unos porcentajes mucho mayores y obtuvieron una nota media ligeramente superior. No hay que dar de lado el hecho de que, al tratarse de una actividad voluntaria, los alumnos que participaron en el WebLab fueron aquellos que se suelen calificar como “mejores” alumnos: los que suelen participar más y obtener mejores resultados. Sin embargo no se puede negar que existe una correspondencia directa entre la superación de este WebLab y la obtención de mejores resultados cualitativos y cuantitativos que, proporcionalmente, es debida a este hecho.

Encuesta. Los resultados obtenidos en las encuestas que completaron los alumnos de los grupos experimental y de control que superaron los WebLabs ayudan a validar la efectividad de esta propuesta docente. La encuesta del grupo experimental tenía 22 preguntas y la del grupo de control 15 ya que no se incluyeron las preguntas relativas a las páginas 1 y 2 que no se incluyeron en el

WebLab de este grupo. Se han obtenido múltiples datos y todos ellos se pueden considerar positivos.

La encuesta es similar a la utilizada en el laboratorio de modelado del motor CC del apartado anterior por lo que se ha reducido la exposición de resultados de la misma a las preguntas finales en las que se les pedía la valoración de diversos aspectos de los WebLabs en una escala Likert de 1 a 5. Todos los aspectos evaluados en los WebLabs han obtenido una valoración positiva, han obtenido una valoración igual o superior a 3,46 (SD 1,07, “Fiabilidad del sistema”) en el grupo experimental y 3,25 (SD 1,21, “Acceso cómodo”) en el grupo de control. Los aspectos mejor evaluados han sido “Diseño Web” en el grupo experimental (3,95 SD 0,78) y “Utilidad” en el grupo de control (4,15 SD 1,04). La **Figura 120** muestra estos datos con más detalle para el WebLab del grupo experimental y el grupo de control.

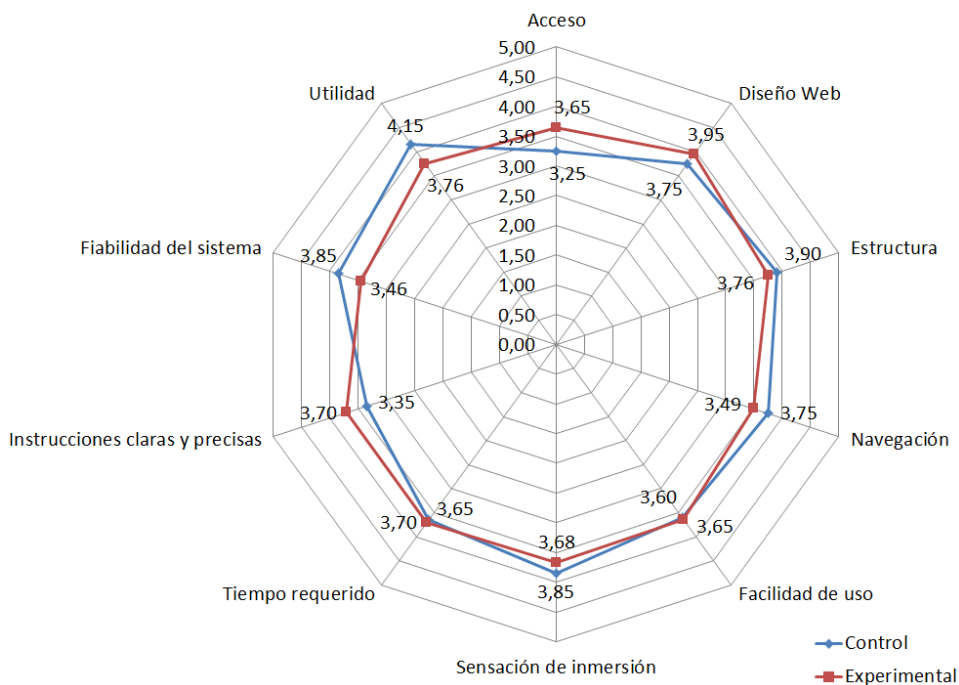


Figura 120. Opinión del alumnado del WebLab de Control PID de Motor CC (diversos aspectos)

El resultado más representativo de estas evaluaciones es la valoración final general que se hace de los WebLabs que ha sido de 3,78 (SD 0,79) en el grupo experimental y de 3,85 (SD 0,67) en el grupo de control como muestra la **Figura 121**.

Reusabilidad del WebLab

Una de las características de este laboratorio es que ha sido el primero que, una vez operativo, se decidió probar su importación a otros LMS para comprobar la reusabilidad de los paquetes SCORM. Para ello se pensó en utilizar el paquete SCORM del WebLab en una plataforma de tipo Moodle, el LMS más extendido a nivel mundial. El WebLab descrito en las secciones anteriores ha sido desarrollado siguiendo la Edición 4ª de la versión 2004 de SCORM (última versión publicada), la cual es soportada por ILIAS. Sin embargo, Moodle sólo soporta la versión SCORM 1.2 (más antigua y con menos posibilidades).

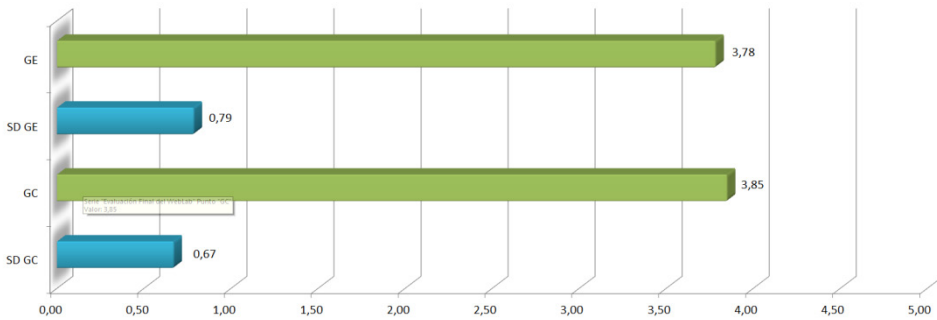


Figura 121. Opinión del alumnado del WebLab de Control PID de Motor CC (Valoración Final)

Por ello se tuvo que realizar una modificación del paquete SCORM original para convertirlo a la versión 1.2. Esto planteó una serie de cambios, principalmente derivados por la imposibilidad de controlar la secuenciación y navegación entre las páginas, ya que no se incluye ninguna sub-especificación a este respecto en la versión 1.2:

- Es necesario usar un empaquetado diferente usando un nuevo modelo base de la versión 1.2 en el que se pueden incluir las mismas 4 páginas de contenido usadas.
- Las llamadas a funciones JavaScript del RTE que hay en las páginas de contenido se deben comprobar y reescribir con las funciones de la versión 1.2. cuando tienen equivalente, o eliminarlas en caso de no existir (p.e. las relativas a la secuenciación y navegación).
- En el software del VL creado con EJS hay que usar un objeto de la clase `scormRTE12` para llamar a las funciones de la versión 1.2 y comprobar las funciones no implementadas para su eliminación o adaptación.

- Simplificación del software del VL en la sección de comentarios SCORM (la versión 1.3 sólo admite un comentario).

De este modo se obtuvo una versión SCORM 1.2 del WebLab. El mismo paquete SCORM fue importado y utilizado en las plataformas ILIAS de la Universidad de Jaén y Moodle de la red UNILabs de laboratorios universitarios compartidos que gestiona la UNED. Ambos se encuentran ubicados con acceso libre (es necesario registrarse en el sistema UNILabs y matricularse en el “Curso de prueba para la Universidad de Jaén”). La **Figura 122** muestra una captura de pantalla cuando se selecciona el paquete SCORM en el LMS Moodle de UNILabs antes de entrar en el mismo. Muestra un resumen sobre las actuaciones pasadas y los resultados obtenidos.

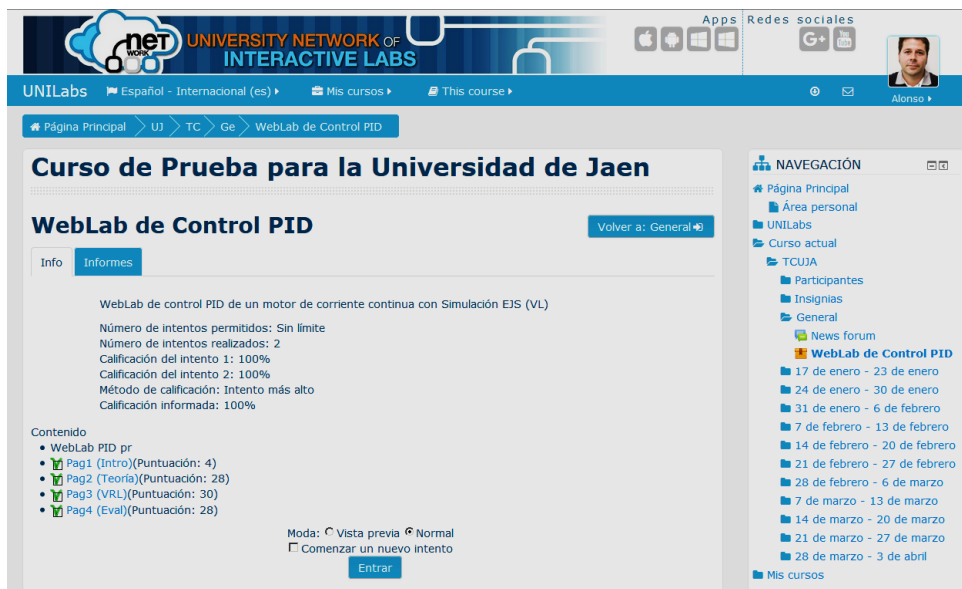


Figura 122. Captura de pantalla de la ventana principal del WebLab en UNILabs

Su funcionamiento fue similar, la secuenciación y navegación en la versión 1.2 es realizada y gestionada por el propio LMS, lo que limita a los creadores del WebLab. La única diferencia observada en la ejecución del WebLab en las dos plataformas radica en el tratamiento que se hace del elemento de SCORM *cmi.comments*. Este dato es una cadena de caracteres con un tamaño máximo de 4096 que se almacena y gestiona por el LMS que puede ser leída o actualizada por las páginas del SCORM, al ser actualizada en ILIAS se reescribe el contenido previamente añadido mientras que la implementación de Moodle concatena los contenidos (comentarios añadidos). La **Figura 123** muestra un

ejemplo de ejecución del WebLab en UNILabs, si se compara con la imagen equivalente de la ejecución de la misma parte del WebLab en ILIAS, **Figura 112**, se puede observar que el sistema a controlar tiene distintos parámetros K_m y T_m , ya que, aunque se trata del mismo usuario, al tratarse de sistemas distintos (UNILabs e ILIAS) la identificación, en la que se basa el VL para personalizar el experimento, es distinta.

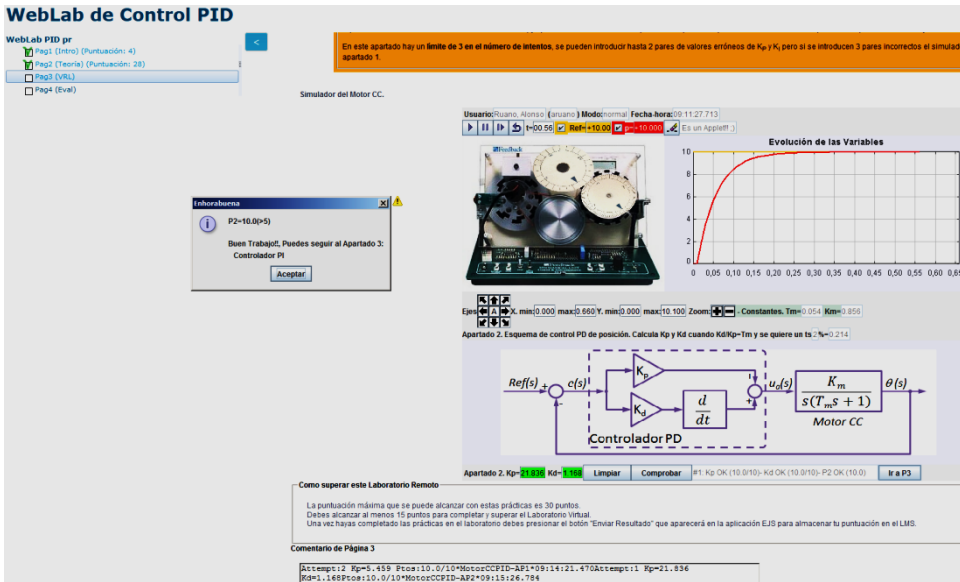


Figura 123. Captura de pantalla de la ejecución del WebLab en UNILabs (Página 3. Paso de experimento 2 (Control PD) a 3 (Control PI)

6.3. Laboratorios Remotos

Los laboratorios remotos añaden una complejidad extra a la existente en los laboratorios virtuales ya que exigen la existencia de una ubicación en la que se debe instalar los dispositivos remotos reales con los que los alumnos deben realizar los trabajos experimentales y una arquitectura de acceso a los mismos. Este último requisito exige la participación de los técnicos y responsables de comunicaciones del organismo en el que se vayan a configurar estas instalaciones.

Antes de explicar los laboratorios remotos desarrollados en la UJA se va a describir la infraestructura y arquitectura de comunicaciones existente en la UJA para la instalación y mantenimiento de los laboratorios remotos.

6.3.1. Estructura y Arquitectura de Comunicaciones

La ubicación de los servidores de laboratorios remotos y los dispositivos y plantas reales a controlar es el Laboratorio de Control de Procesos en la EPS de Jaén. Este laboratorio es una estancia docente en la que imparten clases prácticas presenciales de diferentes grados de ingeniería. Fue el lugar en el que se decidió instalar los dispositivos físicos a utilizar en los laboratorios remotos ya que muchos de los recursos susceptibles de ser conectados y controlados remotamente se encontraban allí. Se plantearon dos problemas:

- La coincidencia de alumnos en clases presenciales con las instalaciones de laboratorios remotos.
- La falta de comunicaciones de red para ofrecer remotamente los recursos.

El primero de los problemas se solucionó con el diseño y construcción de una estructura metálica en aluminio con paneles de plástico transparente que se creó en una zona del mismo laboratorio. Esta estructura cuenta con unas puertas con cerradura y en su interior hay dos baldas de madera a distintas alturas en las que se pueden situar los dispositivos, recursos y/u ordenadores necesarios para la implementación de laboratorios remotos. En la misma estructura se ha realizado una instalación eléctrica consistente en líneas que recorren todas las baldas y terminan en múltiples regletas para facilitar la alimentación eléctrica de cualquier dispositivo que se desee emplear. *La Figura 124* muestra una fotografía de dicha estructura antes del montaje eléctrico (a) y después del mismo (b), con los recursos incluidos.

Con el fin de que los estudiantes puedan hacer uso de los laboratorios remotos desde sus domicilios es necesario ofrecer los servidores a través de Internet. Esto supone la instalación de servidores accesibles en la Red Informática de la UJA (RIUJA) accesibles desde su exterior. La normativa interna de la UJA prohíbe la instalación de servidores sin el permiso necesario. Para resolver el segundo de los problemas se contactó con los Servicios Centrales de Informática (SCI) de la UJA para comunicarles la intención de montar una serie de servicios que debían ser accesibles local y remotamente a RIUJA. La solución que se ofreció fue instalar un conmutador en la propia infraestructura que se situó bajo la balda inferior. Este equipo es un *switch* NORTEL *BayStack* 5510-24T (*Figura 125*), de 24 puertos que permite la conexión en el interior de la infraestructura de hasta

22 puntos de red (2 de las tomas se han reservado para funciones de control y prueba de comunicaciones locales). Se sitúa pegado bajo la balda inferior en la zona izquierda de la infraestructura.

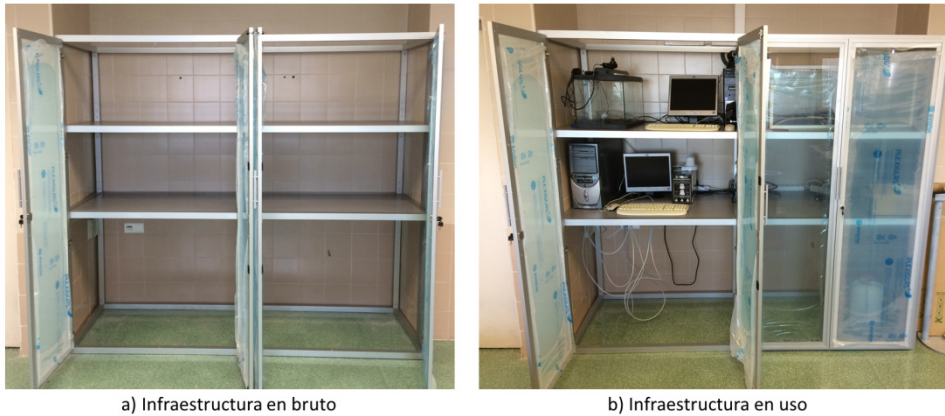


Figura 124. Infraestructura de Laboratorios Remotos



Figura 125. Switch (Conmutador) NORTEL BayStack 5510-24T

En la infraestructura se mantiene una red de área local (LAN, *Local Area Network*) no conectada directamente a Internet, que usa direcciones IP de red privadas de tipo 192.168.X.X, la dirección de la LAN asignada es 192.168.133.0/24. Todos los equipos instalados en la infraestructura (servidores de laboratorio y cámaras IP) deben poseer una dirección de esta LAN y estar configurados para comunicarse a través de la puerta de enlace (*gateway*) 192.168.133.254, que es un equipo intermedio de comunicaciones que realiza además funciones de seguridad y el mapeo de puertos. Este *gateway* está conectado al *switch* mediante 2 líneas de 1 Gbps. El *gateway* tiene una dirección IP pública que es accesible desde cualquier parte de Internet, es la 150.214.97.114, que además tiene asignada la dirección DNS (*Domain Name System*) weblab.ujaen.es que es la dirección pública que deben utilizar todas las aplicaciones VR de cliente que quieran acceder a un equipo remoto.

Aunque todas las aplicaciones utilicen la misma dirección (weblab.ujaen.es o 150.214.97.114), se accederá a un laboratorio u otro en función del puerto de nivel de transporte utilizado. El equipo *gateway*, que recoge todas las solicitudes de comunicación enviadas a esta dirección, se encarga de hacer llegar a la dirección de LAN privada correspondiente en la red 192.168.133.0/24 y permitir las comunicaciones en ambos sentidos.

A modo de ejemplo, si se implementa un laboratorio remoto en la posición intermedia de la infraestructura (Equipos en Red de Lab remoto 3) que tiene una cámara IP y un PC servidor del equipo remoto, es necesario asignarles una dirección IP y un puerto a cada uno de ellos, Cámara IP: 192.168.133.1:8000 y servidor PC: 192.168.133.2:8001. Los dos se tendrán que conectar a cualquiera de los puertos no utilizados del *switch* en la propia infraestructura mediante un cable de par trenzado. Todos los puertos que no se utilizan, por motivos de seguridad, están cerrados, por lo que habrá que ponerse en contacto con el SCI de la UJA para indicarles que configuren el *gateway* y redireccione todo lo que entre en la IP 150.214.97.114:8000 hacia la IP 192.168.133.1:8000 y todo lo que entre por la IP 150.214.97.114:8001 hacia la IP 192.168.133.2:8001. El programador de la aplicación VR sólo tendrá que programar las comunicaciones usando la dirección 150.214.97.114 en los puertos indicados (8000 y 8001 en el ejemplo).

La **Figura 126** muestra el esquema de comunicaciones creado para conectar la infraestructura de laboratorios remotos y la separación de la misma suponiendo que se tienen 6 laboratorios remotos.

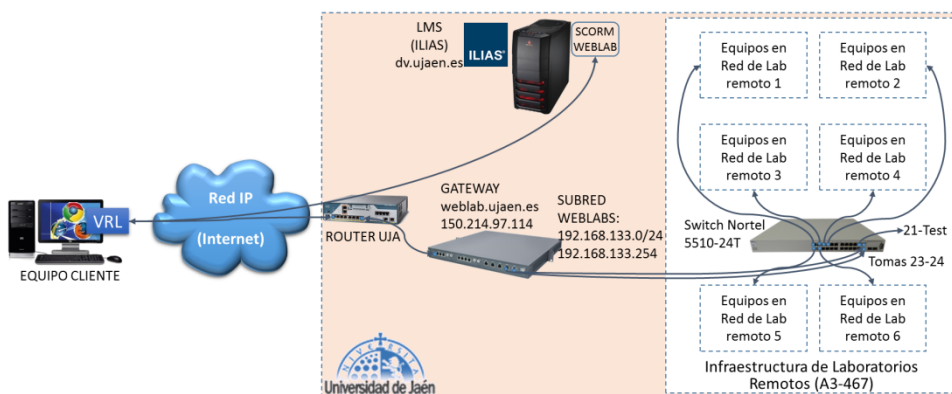


Figura 126. Esquema de comunicaciones para laboratorios remotos de la UJA

6.3.2. Modelado del Sistema Dinámico de un Motor CC

Este laboratorio fue el primero desarrollado en la UJA usando las herramientas y metodologías propuestas en este trabajo. Se trata de una versión remota del laboratorio virtual de modelado del sistema dinámico del motor CC descrito en el apartado 6.2.2.

Descripción de Arquitectura

La arquitectura de la parte remota está descrita visualmente por la **Figura 127**. El modo de integración laboratorio-LMS es el mismo que el mostrado para los laboratorios virtuales; el estudiante abre su navegador e inicia una sesión en el LMS ILIAS de la UJA para descargarse un paquete SCORM, en una de las páginas existe incrustada una aplicación Java VR en forma de *applet* que interactúa con un sistema real remoto. Esta interacción se establece siguiendo el modelo cliente-servidor, en el que el cliente es la aplicación Java VR que utiliza el estudiante y el servidor es un programa C ubicado en un PC servidor. En el apartado anterior se explicó el sistema de comunicaciones genérico que a continuación se va a particularizar para este laboratorio remoto.

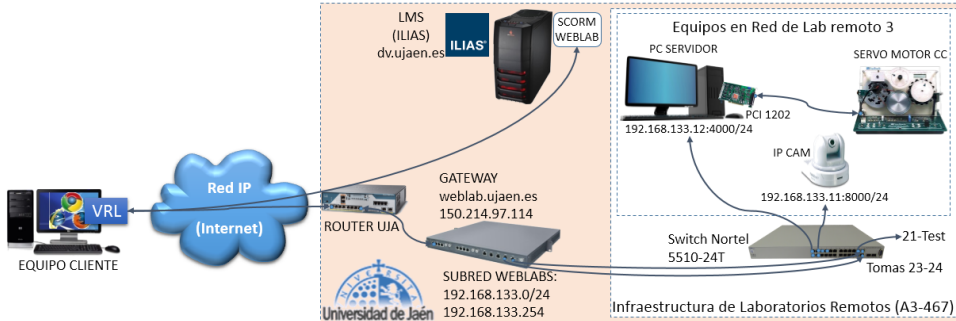


Figura 127. Arquitectura y Sistema de comunicaciones del Laboratorio remoto de modelado de Motor CC

Los dispositivos y aparatos instalados en la zona descrita como “Equipos en Red de Lab remoto 3” en la **Figura 126** son los siguientes:

- **PC servidor:** Se trata de un PC genérico con el S.O. Windows 7 en el que se está ejecutando un *software* de servidor escrito en lenguaje C. Este programa contiene los siguientes elementos que le permiten realizar una serie de tareas:

- **Un interface de configuración y puesta en marcha del software de servidor.** Este interface permite atender las órdenes en modo comando de un administrador. Accediendo al propio PC se puede configurar el puerto en el que se quiere atender las conexiones de la aplicación VR que usan los alumnos, el tiempo asignado a cada conexión, iniciar el programa servidor o salir del mismo. La **Figura 128** muestra la interfaz mostrado al administrador.

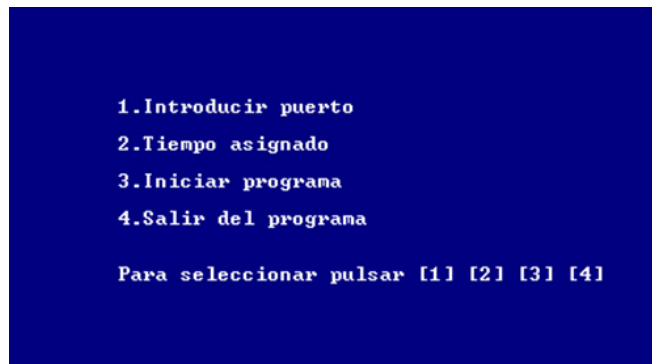


Figura 128. Interfaz de usuario administrador del software de servidor

- **Un *middleware* de comunicaciones *socket*.** Desarrollado para establecer conexiones con una aplicación VR en forma de *applet* Java desarrollada en EJS e incrustada en una página de un paquete SCORM que es manejada por los estudiantes. Permite hacer las siguientes funciones: **a) *gestionar conexiones remotas*** de la aplicación VR utilizada por los estudiantes: una vez iniciado se queda a la espera de conexiones en el puerto configurado, una vez establecida la conexión el estudiante podrá trabajar con el motor CC, pasado el tiempo el servidor cerrará la conexión y quedará a la espera de activar otra (el socket es no bloqueante y si existe otra petición de conexión mientras se mantiene una activa se le indica este hecho y que debe esperar), **b) *Intercambiar datos con el VR de los estudiantes mientras estén conectados***. Mientras un estudiante está conectado podrá indicar la tensión que se desea introducir en el sistema y recibir la salida del sistema (velocidad angular del motor CC).

- **Funciones de control de velocidad.** La generación de la señal de tensión que es la entrada del sistema servomotor se realiza en el programa a partir de las directrices marcadas desde la aplicación VR EJS del estudiante.
- **Funciones de lectura/escritura de datos externos.** Basadas en la librería C *P1202.lib* permiten gestionar una tarjeta de adquisición de datos (descrita más adelante) con la que se puede gestionar la tensión de entrada del sistema servomotor (señal de entrada del sistema) y conocer la velocidad angular del motor CC.
- **Tarjeta de adquisición de datos PCI 1202.** Instalada en un slot PCI del PC y usada mediante la librería *P1202.lib* en lenguaje C. Está configurada para trabajar con entradas analógicas simples y con un rango de tensión de salida de $\pm 10V$). Dispone un conector DB-37.
- **Una unidad Servomotor 33-100 de la marca *Feedback Instrument*.** Es la misma que se simuló en el laboratorio virtual de modelado del sistema dinámico del motor CC descrito en el apartado 6.2.2.
- **Un cable plano de 34 cables.** Encargado de conectar la tarjeta PCI1202 con la unidad servomotor 33-100.
- **Una fuente de alimentación externa DC 01-100.** También de la marca *Feedback Instrument*, de +15V y -15V de 1,5A y de +5V de 0.5A. Es necesaria para alimentar al sistema Servomotor de CC y lograr su correcto funcionamiento.
- **Una cámara IP PTZ modelo TV-IP410 de la marca TrendNET.** Apunta directamente al sistema servomotor para mostrar todos sus movimientos. Una vez conectada a la red, se ha configurado como un servidor web a la espera de peticiones de conexiones de clientes identificados. La aplicación VR del estudiante, al iniciarse, se conecta automáticamente, se identifica y muestra las imágenes que recibe en la interfaz de usuario pudiéndose visualizar el movimiento del sistema servomotor.

La fotografía de la **Figura 129** muestra estos componentes en el espacio reservado que existe en la infraestructura de laboratorios remotos descrita anteriormente.

Estructura del paquete SCORM

En el lado del cliente, El paquete SCORM tiene la misma estructura y contenido que el mostrado en el apartado 6.3.2. La única diferencia radica en el *applet* Java RL y las conexiones externas que realiza al PC servidor y a la cámara IP ubicadas en la infraestructura de laboratorios remotos como se muestra en la **Figura 130**.



Figura 129. Equipos del laboratorio remoto en las instalaciones

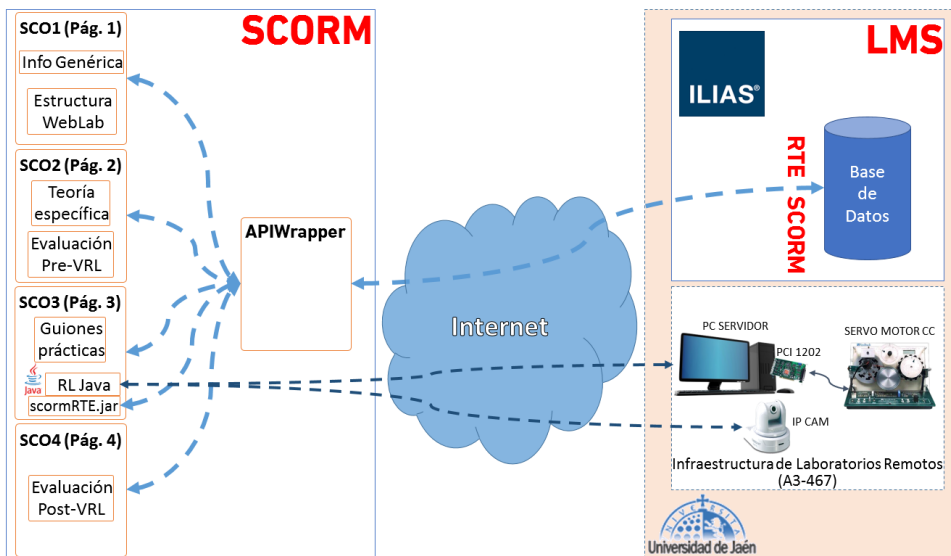


Figura 130. Estructura y comunicaciones de Lab remoto Modelado del Sistema Dinámico de un Motor CC

RL

El *applet* Java desarrollado en EJS presenta la interfaz de usuario mostrado en la **Figura 131**, es muy similar a la interfaz de usuario mostrado en el laboratorio virtual de modelado de este mismo sistema servomotor (**Figura 93**).

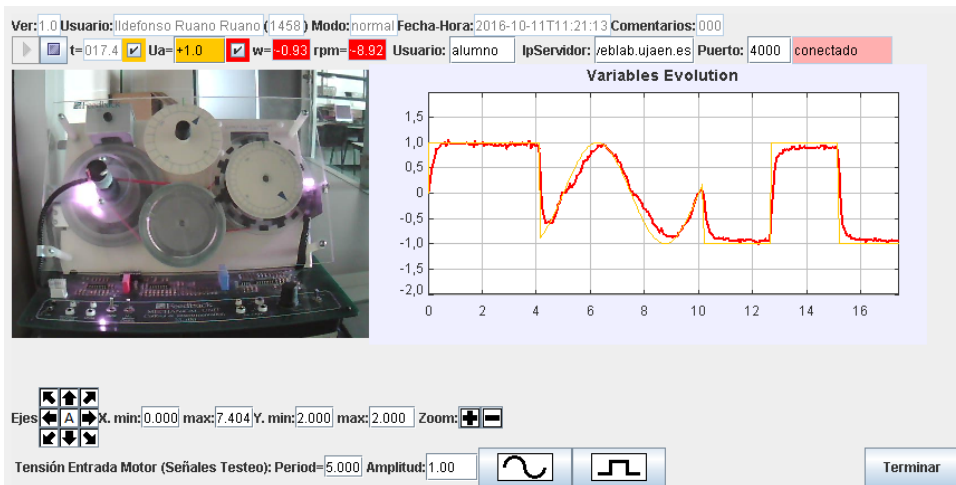


Figura 131. Interfaz de usuario de aplicación *applet* Java de RL

La única diferencia consiste en que al pulsar el botón de ejecutar (botón *play*) se establece una conexión con el servidor indicado por la dirección IP del campo “*IPservidor*” y el puerto del campo homónimo usando la acreditación de usuario indicada en el campo correspondiente. Todos estos campos se muestran por ser una versión prototipo (no utilizada aún por alumnos), en la versión final serán ocultos. Si se establece la conexión se mostrará la palabra “conectado” como se muestra en la **Figura 131** y se realizará el envío de la consigna seleccionada: el valor constante indicado por el campo u_a , tensión alterna o de tensión continua cuadrada con la amplitud y periodo introducidos.

La imagen del motor se recibe de la cámara IP y se muestra nada más cargar y empezar a ejecutarse el *applet* en la zona media-izquierda del interfaz, así como el valor de la consigna y velocidad angular del sistema en el gráfico que existe en la zona media derecha. Fíjese en las irregularidades que muestra la gráfica de la velocidad angular al no realizarse ningún filtrado de la misma.

La conexión e intercambio de datos se produce durante el tiempo configurado en el servidor (normalmente un minuto), pasado ese tiempo se recibe un aviso de desconexión.

Para una descripción más completa del software de servidor y la aplicación EJS cliente se puede encontrar más información en [305].

6.3.3. Comunicaciones SIP

Este laboratorio fue el primero desarrollado y utilizado en Telecomunicaciones, especialidad en Telemática. Se creó siguiendo todos los pasos de la metodología propuesta en el capítulo 5 como paquete SCORM 2004 integrado en un LMS, la plataforma institucional ILIAS de la UJA, añadiendo y utilizando recursos del LMS. Hay dos versiones del mismo, a continuación se va a describir el primero de ellos [306].

El RL contiene una serie de prácticas orientadas a trabajar con un cliente y un servidor SIP (*Session Initiation Protocol*), SIP es el Protocolo de Inicio de Sesiones, desarrollado por el IETF (*Internet Engineering Task Force*) que sirve para iniciar, modificar y finalizar sesiones interactivas de comunicaciones multimedia entre usuarios (principalmente VoIP, Voz sobre IP o telefonía por Internet). El WebLab contiene un plan de aprendizaje que incluye una serie de recursos de utilidad docente como teoría sobre SIP, pruebas de evaluación, un laboratorio remoto de SIP formado por dos *applets* Java (cliente y servidor SIP) y experimentos personalizados para cada alumno cuyos resultados son almacenados en el LMS. Este WebLab se ha presentado en el LMS institucional de la Universidad de Jaén a los 17 alumnos de la asignatura “Sistemas de Telefonía” en el curso 2015-16. Los datos de uso han permitido realizar diversas evaluaciones que demuestran que los alumnos que lo han completado han obtenido un rendimiento excelente en el propio WebLab, han conseguido unos resultados muy buenos en la evaluación final de la asignatura y lo han valorado muy positivamente.

Dentro del WebLab hay cuatro páginas en las que los estudiantes pueden encontrar, además del laboratorio remoto (RL) de SIP, recursos útiles relacionados con los experimentos. Los alumnos pueden navegar por estas páginas accediendo a estos recursos a través del itinerario de aprendizaje programado en secuencia (página 1 a 4). Esta secuencia puede modificarse en

función de las acciones y calificaciones que obtengan los alumnos. Todas las páginas del WebLab incluyen una evaluación automática, las calificaciones resultantes obtenidas en cada página son almacenadas en el LMS ILIAS para que los tutores del curso puedan acceder a las mismas. El RL está formado por dos *applets* de Java que están incrustados en la página 3 del WebLab. Este RL se comunica con el LMS utilizando la sub-especificación RTE, responsable de las comunicaciones de SCORM, para obtener la identificación del usuario. Cuando un alumno ejecuta el RL, éste le presenta 3 prácticas personalizadas basados en su identificación y almacena en el LMS las calificaciones que consigue. Los objetivos y competencias globales y específicos seleccionados para este WebLab fueron:

- Identificar los distintos elementos que intervienen en las comunicaciones multimedia.
- Conocer los protocolos utilizados para el intercambio de información multimedia.
- Conocer las técnicas y protocolos del transporte de audio y vídeo.
- Conocer los protocolos para el establecimiento y control de sesiones multimedia.
- Conocer los distintos protocolos utilizados en la transmisión de voz sobre redes TCP/IP.
- Conocer los elementos que intervienen en una comunicación de VoIP.
- Configurar un servidor *proxy* y estudiar su comportamiento.
- Configurar un *softphone* y estudiar su comportamiento.
- Visualizar los paquetes del protocolo SIP intercambiados durante una comunicación.
- Identificar los distintos campos existentes en cada paquete SIP.

Descripción de Arquitectura de Comunicaciones y Estructura del WebLab

Se puede decir que este laboratorio es remoto porque se hacen uso de recursos remotos al dispositivo que utiliza el estudiante (normalmente un PC), aunque no es un laboratorio remoto tradicional como el mostrado en el apartado anterior (6.3.2). Este es un laboratorio de comunicaciones, y como tal, los alumnos deben realizar conexiones cliente-servidor utilizando el protocolo SIP y entender su funcionamiento. En este laboratorio el recurso externo con el que interactúan los estudiantes es el *software* ejecutado por otro compañero,

también estudiante, como muestra la **Figura 132**. Existen prácticas en las que estas conexiones se producen de forma local, en el mismo dispositivo en el que está trabajando el alumno (cuando un alumno usa su cliente para conectarse con su propio servidor) (1), y otras en las que se producen de forma remota, conectándose dos clientes entre sí (uno remoto al otro) gracias a un servidor que puede ser local (para el estudiante 1 la comunicación 2-3 y para el estudiante 2 la 2-3) o remoto (para el estudiante 2 la comunicación 2-3 y para el estudiante 1 la 2-3)

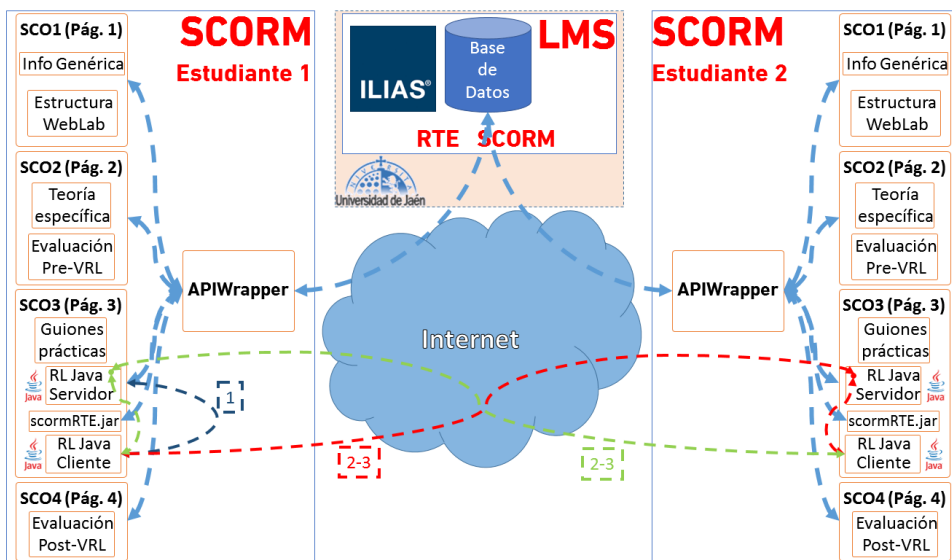


Figura 132. Estructura y comunicaciones de Lab remoto SIP

La página 1 (**SCO1**) (**Figura 133**) muestra una tabla que incluye información general y básica del laboratorio para que el alumno pueda contextualizarlo dentro de los contenidos teóricos de la asignatura, conozca los objetivos que persiguen con el mismo y las competencias que puede adquirir. También se muestra información sobre la estructura del laboratorio, los contenidos del mismo y la forma de superarlo. Se añade un botón para acceder a la siguiente página (SCO2) del laboratorio. Se supera simplemente con estar más de 3 segundos en la página y no tiene ninguna puntuación asociada.

La página 2 (**SCO2**) contiene información teórica específica del laboratorio y un test pre-RL. En la parte de teoría (**Figura 134**) se define el protocolo SIP y su situación en la torre de protocolos de comunicaciones, los explican elementos de una arquitectura SIP, los tipos de servidores y terminales, las funciones SIP, las direcciones SIP y el formato de mensajes SIP. El test (**Figura 135**) consiste en

7 preguntas de opción múltiple, 6 de respuesta simple y una de respuesta múltiple, con el que se trata de asegurarse que los estudiantes tienen los conocimientos previos mínimos para trabajar en el RL de SIP. La puntuación alcanzada (el máximo son 28 puntos), el estado de superación y el de terminación son almacenados por el SCO2 en el LMS.

Introducción al WebLab (Página1)
Introducción al WebLab (Página1)

COMUNICACIONES SIP

UNIVERSIDAD:	UNIVERSIDAD DE JAÉN
TITULACIONES:	GRADO EN INGENIERÍA TELEMÁTICA
ASIGNATURA:	SISTEMAS DE TELEFONÍA
LABORATORIO:	LABORATORIO REMOTO
PLANIFICACIÓN:	4º CURSO // 1º CUATRIMESTRE
CONTEXTO DEL EJERCICIO:	TEMA 5. ARQUITECTURA DE VOIP SIP
OBJETIVOS Y COMPETENCIAS:	<ul style="list-style-type: none"> • GLOBALES: <ul style="list-style-type: none"> ◦ Identificar los distintos elementos que intervienen en las comunicaciones multimedia. ◦ Conocer los protocolos utilizados para el intercambio de información multimedia. ◦ Conocer las técnicas y protocolos del transporte de audio y vídeo ◦ Conocer los protocolos para el establecimiento y control de sesiones multimedia. ◦ Conocer los distintos protocolos utilizados en la transmisión de voz sobre redes TCP/IP. ◦ Conocer los elementos que intervienen en una comunicación de VoIP. • ESPECÍFICOS: <ul style="list-style-type: none"> ◦ Configurar un servidor proxy y estudiar su comportamiento. ◦ Configurar un softphone y estudiar su comportamiento. ◦ Visualizar los paquetes del protocolo SIP intercambiados durante una comunicación. ◦ Identificar los distintos campos existentes en cada paquete SIP.
TEORÍA:	Enlace a la carpeta de la asignatura: SISTEMAS DE TELEFONÍA Enlace a la carpeta de teoría: TEMA 5. ARQUITECTURA DE VOIP SIP

Estructura del WebLab (Laboratorio Web) y Modo de superarlo

La siguiente figura muestra la organización de los contenidos del WebLab:

El Contenido está estructurado en 4 secciones (Páginas):

- **Página1.Introducción (Esta Página):** Proporciona información general sobre el WebLab y se enmarca dentro de los estudios de grado.
- **Página2.Teoría (Página siguiente):** Proporciona información teórica relacionada con el funcionamiento del protocolo SIP y los mensajes intercambiados. También contiene un test inicial que trata de asegurar que se han asimilado los conceptos teóricos explicados antes de acceder al laboratorio en sí con el objetivo de que no pueda acceder al laboratorio alguien que no ha adquirido los conocimientos necesarios para usarlo y aprovecharlo.
- **Página3.VRL:** (Para acceder a este sección es necesario haber superado el test inicial de la sección Página2.Teoría) Proporciona los guiones de las prácticas que se deben realizar con el laboratorio y el propio VRL (Virtual Remote Lab) en el que se podrá trabajar con el fin de superar las prácticas propuestas. El VRL realiza una evaluación automática del trabajo realizado por cada alumno.
- **Página4.Evaluación (Última página):** (Para acceder a este sección es necesario haber superado la evaluación automática de la sección Página3.VRL) Muestra un test final relacionado con todo el trabajo e información previa incluida en el WebLab que los estudiantes deben superar.

Para superar el WebLab se debe llegar hasta la sección Página4.Evaluación y superar el test de evaluación final.

Figura 133. Captura de pantalla del SCO1 con la información genérica y estructura del laboratorio remoto SIP

Teoría del Weblab (Página2)
Teoría del Weblab (Página2)

COMUNICACIONES SIP

Introducción

El **objetivo** de este ejercicio es **iniciar al estudiante en el funcionamiento del protocolo SIP** en general, y, en particular, en el aprendizaje de los mensajes intercambiados entre dos clientes y un proxy en una **comunicación VoIP** mediante el protocolo SIP.

Session Initiation Protocol (Protocolo de Inicio de Sesiones) es un protocolo de control y señalización usado mayoritariamente en los sistemas de Telefonía IP, que fue desarrollado por el IETF (RFC 3261). Dicho protocolo permite crear, modificar y finalizar sesiones multimedia con uno o más participantes y sus mayores ventajas recaen en su simplicidad y consistencia.

Hasta la fecha, existían múltiples protocolos de señalización tales como el H.323 de la ITU, el SCCP de Cisco, o el MGCP, pero parece que poco a poco SIP está ganando la batalla del estándar. Cisco está progresivamente adoptando SIP como protocolo en sus sistemas de telefonía IP en detrimento de H.323 y SCCP, Microsoft ha elegido SIP como protocolo para su nuevo OCS (Office Communication Server), y los operadores (de móvil y fijo) también están implantando SIP dentro de su estrategia de convergencia, aprovechando de este modo la escalabilidad y interoperabilidad que nos proporciona el protocolo SIP.

Figura 134. Captura de pantalla parcial del SCO2 del laboratorio remoto SIP (Teoría).

Test previo al Laboratorio (Evaluación Inicial)

MUY IMPORTANTE, LEER ANTES DE EMPEZAR A RESPONDER EL TEST

- Presta atención a las siguientes preguntas, lee tranquilamente el enunciado y contesta con tranquilidad.
- El objetivo de las mismas es saber si has asimilado correctamente los conocimientos que debes obtener.
- Fíjate en que si respondes correctamente recibirás 4 puntos por cada pregunta, pero si lo haces de forma incorrecta te restará un punto.
- También tienes la opción de responder señalando la opción "No sabe/No contesta", en cuyo caso no se sumará ni se restará ningún punto.
- Para poder superar el test es necesario que obtengas al menos 14 puntos (máximo 28).
- Sólo se tiene una oportunidad para contestar a cada una de las preguntas, una vez pulses el botón "Contestar" ya no podrás volver a hacerlo.
- Si has señalado una opción pero te has arrepentido y deseas dejar la respuesta en blanco pulsa sobre "Reiniciar".
- Al final de la página debes pulsar sobre el botón "Contestar Todo y Enviar" para mostrar tus resultados finales y enviarlos al LMS.
- Se considerará que has contestado al test si al menos has contestado a una de las preguntas.
- No refresques el contenido de la página en el navegador, si lo haces se cargará de nuevo la Página1-Introducción y deberás volver a empezar el Weblab.

Pregunta 1

PREGUNTA 1: Formato SIP.
(Bien:4 puntos - Error:-1 punto)

¿Cuál es el primer componente del formato de un mensaje SIP?

- Cabecera
- Línea en blanco
- Línea de inicio
- Cuerpo del mensaje
- No sabe/No contesta

Figura 135. Captura de pantalla parcial del SCO2 del laboratorio remoto SIP (Test pre-RL)

La página 3 (**SCO3**) incluye un pequeño manual de uso de la GUI de las aplicaciones Java que deben usar, los guiones de prácticas del RL (**Figura 136**) y las aplicaciones del propio RL incrustadas como dos *applets* de Java.

En la descripción de las prácticas se ha utilizado el mismo código de colores que se utilizó en el laboratorio de control PID (apartado 6.2.3) para facilitar al estudiante el entendimiento de las prácticas y acciones a realizar (**Figura 109**).

Tras los guiones de prácticas, en la parte inferior del SCO3, se encuentran los *applets* Java del RL, el servidor proxy SIP y el cliente SIP. Los *applets* de Java fueron creados a partir de versiones *freeware* de un cliente SIP (*softphone*) y un

servidor SIP *proxy* que fueron adaptados para poder mostrar los contenidos y formatos de los mensajes SIP y presentar una interfaz acorde a las necesidades de las prácticas que se habían diseñado. Los estudiantes debían realizar 3 prácticas:



Figura 136. Captura de pantalla parcial del SCO3 del laboratorio remoto SIP (Introducción a las prácticas)

- Práctica 1. Activación del *proxy* SIP y Registro del *softphone*.** El objetivo de esta primera práctica es que el estudiante adquiera familiaridad con el simulador y conozca su manejo. Para ello debe activar el servidor *proxy* SIP en el *applet* Java del servidor (**Figura 137**) y después manipular el *applet* Java del cliente (**Figura 138**) para rellenar los campos necesarios para registrarse en el servidor que acaba de activar (conexión local). Si se hace correctamente se consiguen 8 puntos.
- Práctica 2: Comunicación con otro estudiante mediante chat.** Esta segunda parte de la práctica consiste en una comunicación entre dos estudiantes, primero cada uno registrado en su propio servidor *proxy* y utilizando la dirección IP del otro estudiante y, a continuación, usando el *proxy* de uno de ellos como servidor. En esta práctica hay 4 tareas (cada una de ellas suma 2 puntos): enviar un mensaje de texto a un estudiante registrado en otro *proxy* distinto, recibir un mensaje de texto de un estudiante registrado en otro *proxy* distinto, enviar un mensaje de texto a un estudiante registrado en nuestro mismo *proxy* y recibir un mensaje de texto de un estudiante registrado en nuestro mismo *proxy*.

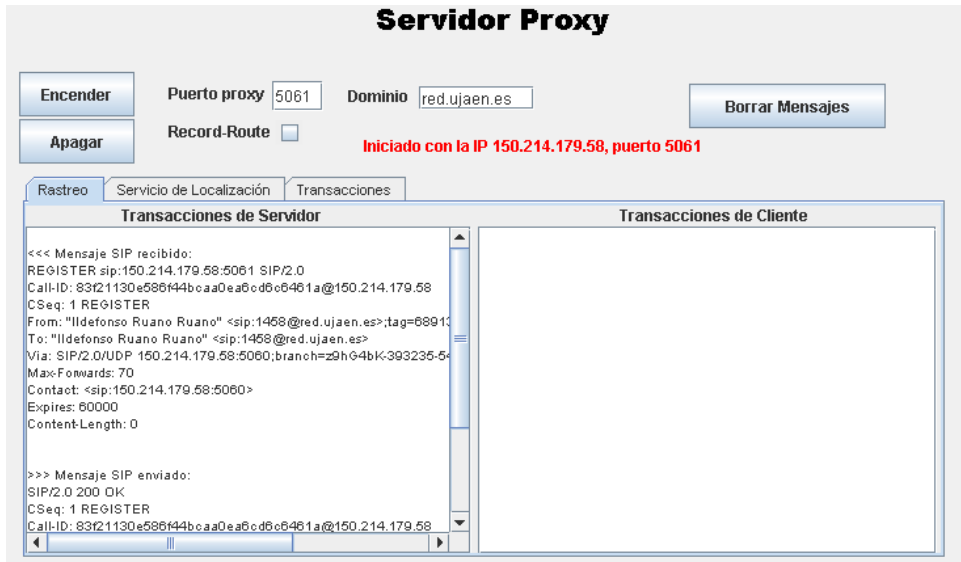


Figura 137. Interfaz de usuario del applet Java del servidor proxy SIP



Figura 138. Interfaz de usuario del applet Java del cliente SIP (softphone)

- Práctica 3: Comunicación con otro estudiante mediante una llamada.**
 En esta práctica, dos estudiantes volverán a comunicarse entre sí pero, en esta ocasión, experimentarán con los mensajes SIP que se envían mediante la simulación de una llamada de audio o de audio y video. Tienen que realizar tres tareas (cada una de ellas suma 4 puntos):

realizar una llamada a otro estudiante, aceptar una llamada entrante, una vez establecida una llamada, ser el estudiante que cuelga.

El máximo número de puntos que se pueden alcanzar es 28. Una vez que se tengan puntos suficientes para superar la práctica (la mitad del máximo, es decir, 14), aparecerá un nuevo botón con el texto “Finalizar” en el *applet* del cliente. La puntuación alcanzada y los estados de terminación y superación serán almacenados en el LMS.

Presentación del Laboratorio online a los Estudiantes

Los 17 alumnos de la asignatura “Sistemas de telefonía” de la Universidad de Jaén en el curso 2015-16 tuvieron acceso a este WebLab. Debido al número tan bajo de estudiantes no se creyó posible dividirlos en el LMS en dos grupos virtuales (grupo experimental y grupo de control) como se ha hecho en otros laboratorios con el fin de analizar el rendimiento y la efectividad del laboratorio en dos escenarios diferentes. A los 17 se les presentó una carpeta virtual en el LMS ILIAS (**Figura 139**) que contenía los siguientes recursos:

- *Pasos Previos a Ejecución de WebLab*. Este es el documento de requisitos del VL que se ha implementado como un módulo ILIAS.
- *WebLab PID Motor de CC*. Este es el paquete SCORM del WebLab completo con las 4 páginas.
- *Encuesta WebLab*. Encuesta utilizada para conocer la opinión de los estudiantes sobre el Weblab.



Figura 139. Captura de pantalla de espacio virtual del WebLab SIP

14 de los 17 alumnos hicieron uso de estos recursos, entraron en el WebLab y respondieron a la encuesta de opinión.

Datos de Uso, Resultados y Conclusiones

A partir de los datos obtenidos del LMS ILIAS se crearon tablas de uso y resultados obtenidos en el WebLab que pueden verse en la **Figura 140**. Algunos de los datos más notables del mismo son los siguientes:

#	Último acceso	Intentos	SCO1			SCO2			SCO3			SCO4			TOTAL			Nota Prácticas Conv1 2015-15	Nota Asignatura Conv1 2015-16
			Tpo	Tpo	Pts	Tpo	Pts	Tpo	Pts	Tpo	Pts	Tpo	Pts	Tpo	Pts	Nota			
1	18. Dic 2015, 18:46	5	2	10	23	29	24	12	24	53	71	8,45	6,75	3,90					
2	18. Dic 2015, 18:54	5	4	5	28	82	24	9	18	100	70	8,33	9,25	8,50					
3	18. Dic 2015, 19:29	2	4	18	24	56	24	10	18	88	66	7,86	9,50	9,26					
4	18. Dic 2015, 18:45	4	5	12	23	76	26	11	23	104	72	8,57	7,00	7,22					
5	18. Dic 2015, 18:54	11	1	16	28	79	24	9	23	105	75	8,93	9,25	7,99					
6	18. Dic 2015, 19:27	3	4	12	23	39	24	9	18	64	65	7,74	9,50	8,14					
7	18. Dic 2015, 18:55	11	4	11	23	91	24	9	28	115	75	8,93	9,75	9,28					
8	18. Dic 2015, 18:54	14	3	12	18	109	24	9	23	133	65	7,74	9,75	8,62					
9	18. Dic 2015, 18:40	6	1	12	28	70	24	6	28	89	80	9,52	9,75	7,79					
10	18. Dic 2015, 18:35	4	3	7	28	43	28	11	28	64	84	10,00	9,50	8,80					
11	18. Dic 2015, 19:18	20	1	11	28	97	24	14	13	123	65	7,74	9,75	7,19					
12	18. Abr 2016, 13:37	11	1	16	28	61	28	16	23	94	79	9,40	9,50	8,05					
13	18. Dic 2015, 19:00	6	1	8	18	143	26	2	28	154	72	8,57	9,50	7,10					
14	18. Dic 2015, 19:01	4	3	20	23	37	24	14	23	74	70	8,33	7,00	7,73					
Promedios		7,57	2,64	12,14	24,50	72,29	24,86	10,07	22,71	97,14	72,07	8,58	8,98	7,83					
SD		5,11	1,45	4,19	3,61	31,56	1,51	3,50	4,60	28,37	5,99	0,71	1,13	1,33					

Nota media Prácticas	8,17	8,83	8,98
SD	0,59	2,66	1,13
Curso académico	2013-14	2014-15	2015-16
	Nota Prácticas Conv1		

7,11	7,21	7,83	Nota media Asignatura
0,72	2,24	1,33	SD
2013-14	2014-15	2015-16	Curso Académico
Nota Asignatura Conv1			

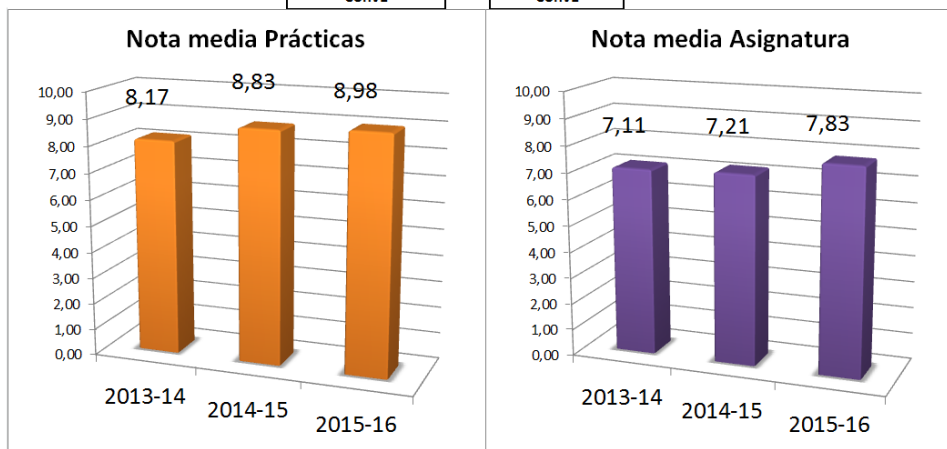


Figura 140. Datos de uso y resultados obtenidos por estudiantes del WebLab "Comunicaciones SIP"

- El WebLab se podía repetir tantas veces como se quisiera, el promedio de intentos por alumno fue de 7,57.
- El tiempo medio de trabajo en el WebLab fue de más de 97 minutos, suponiendo el número de intentos del punto anterior (7,57) tardan de media menos de 13 minutos en cada intento.
- La página en la que se invierte más tiempo es la tercera, (SCO3), en la que tienen que realizar las prácticas, algo deseable en un laboratorio.
- Todas las páginas fueron superadas por todos los alumnos que accedieron a WebLab, además con buenas medias (8,75 en SCO2, 8,88 en SCO3 y 8,11 en SCO4).
- El WebLab en general fue superado por todos y con una nota media bastante buena de 8,58, siendo la nota más baja un 7,74.
- Todos los alumnos que superaron el WebLab superaron también la parte práctica de la asignatura con una media de 8,98 y la asignatura con una media de 7,83.
- La evolución de las notas prácticas y finales de la asignatura de “Sistemas de Telefonía” no se ha visto alterada por la aplicación del WebLab y han mantenido una línea creciente desde que empezó a impartirse en el año 2013/14.

Evaluación de la Efectividad del WebLab

Al no poder establecer un grupo de control por el escaso número de alumnos, se hace más difícil medir la efectividad del WebLab remoto. Los únicos métodos que se han utilizado se basan en los buenos resultados obtenidos por todos los alumnos que trabajaron en el WebLab, que se ha mostrado en el sub-apartado anterior y en los resultados de una encuesta que se pasó a los alumnos tras finalizar su trabajo en el WebLab. Es, básicamente, la misma encuesta utilizada en el laboratorio virtual de la sección “6.2.2. Modelado de Sistemas Dinámicos: Motor CC” y por tanto tiene las mismas preguntas adaptadas a este WebLab. A continuación se muestran los gráficos obtenidos como promedios de valoración en escala Likert dados por los alumnos sobre los contenidos incluidos en el WebLab (**Figura 141**), diversos aspectos sobre el WebLab en general (**Figura 142**) y la valoración final del WebLab (**Figura 143**).

Como ya se comentó en la sección 4.1, el uso de aplicaciones Java en forma de *applets* incrustadas en una página Web es una tecnología que se ha vuelto

obsoleta, muchos navegadores prohíben su ejecución y otros ponen bastantes dificultades. Este hecho se ha visto reflejado en los resultados de la encuesta de opinión. El tipo de contenido menos adecuado del WebLab según la opinión de los alumnos han sido los *applets* Java del RL con bastante diferencia: Los *applets* JAVA del RL han sido adecuados: 3,9 con SD 1,2 frente al 4,2 del siguiente contenido peor valorado como se puede observar en la **Figura 141**. El problema, siempre a criterio de los alumnos, parece estar en la fiabilidad del sistema basado en los *applets* de Java, es el aspecto peor valorado de todo el WebLab con sólo un 4,0 frente al 4,4 que obtienen otros aspectos que le siguen como peor valorados. Por otro lado, el tiempo requerido, las instrucciones claras y precisas y el acceso cómodo son los aspectos mejor valorados como puede verse en la **Figura 143**.

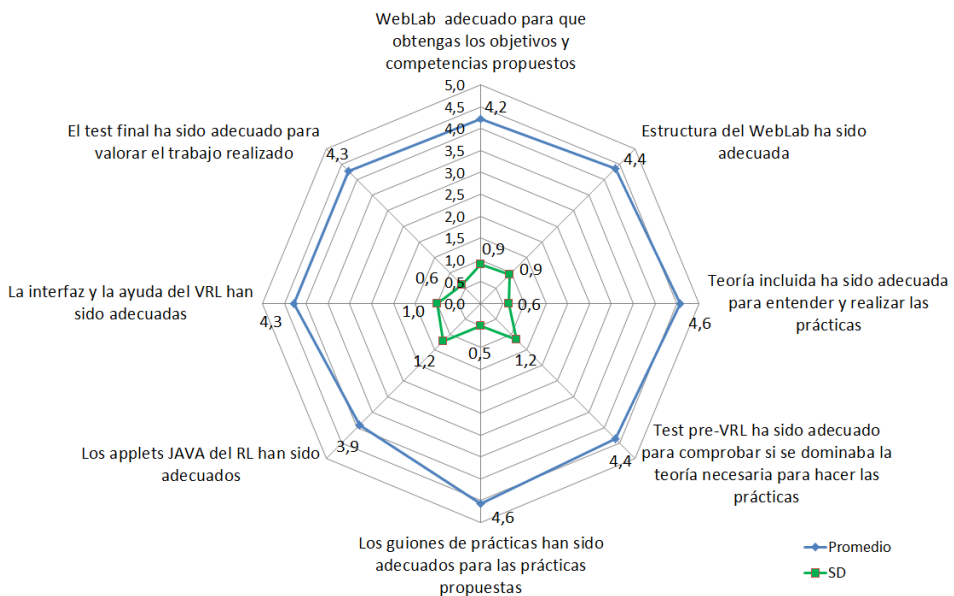


Figura 141. Opinión de alumnado: Adecuación de contenidos del WebLab SIP (SCORM)

La opinión final de los estudiantes sobre el WebLab en general obtiene una muy buena calificación de 4,21 que se puede trasladar a un 8,42 sobre 10, cercano al sobresaliente.

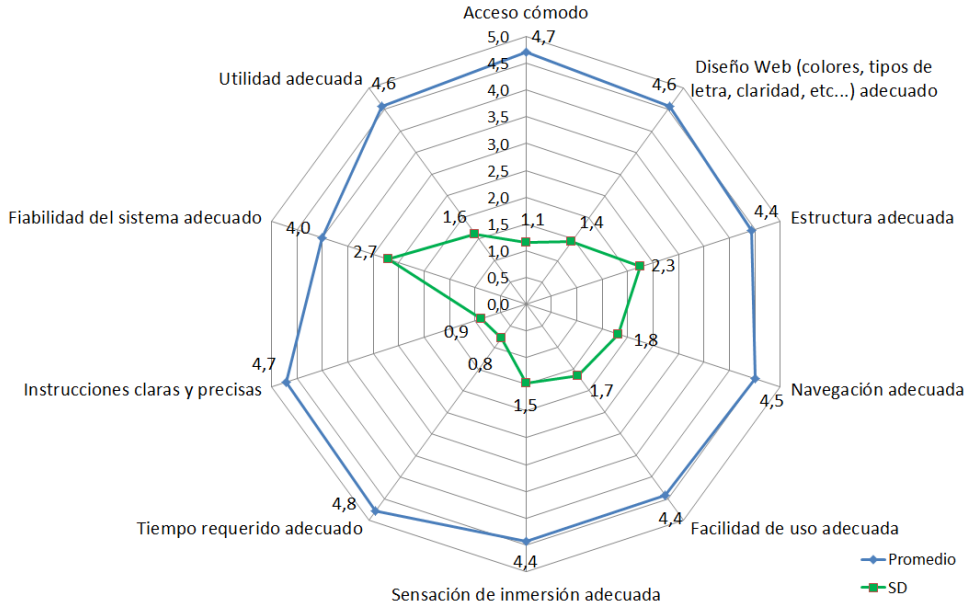


Figura 142. Opinión del alumnado del WebLab de Comunicaciones SIP (diversos aspectos)

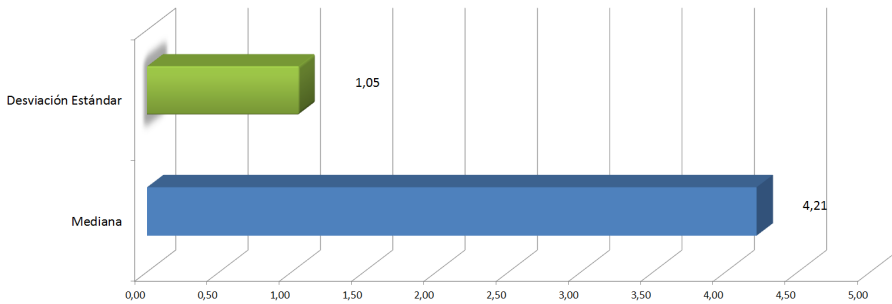


Figura 143. Opinión del alumnado: Nivel de satisfacción final del WebLab de Comunicaciones SIP

Versión JavaScript del WebLab Comunicaciones SIP

La versión del WebLab de comunicaciones SIP basada en aplicaciones Java presenta un par de problemas, uno de los cuales ya ha sido explicado en el apartado anterior; el uso de Java como tecnología para crear las aplicaciones que debe utilizar el estudiante para realizar sus prácticas. El otro problema radica en el hecho de que la configuración de comunicaciones exigibles para realizar la práctica, según mostraba la **Figura 132**, para poder completar las prácticas en los experimentos 2 y 3 se exige el funcionamiento de un servidor *proxy* SIP en uno de los dispositivos de los alumnos. Cuando los alumnos realizan los experimentos en el laboratorio de manera presencial no existe

ningún problema ya que todos trabajan en la misma red. Pero cuando intentan hacerlo desde sus domicilios es necesario configurar los *routers* que les proporcionan el acceso a Internet para que les permitan recibir conexiones externas. Aunque no es algo imposible de conseguir, requiere conocimientos avanzados que no se puede asegurar que posean los estudiantes y conocimiento del funcionamiento particular del *router*, algo que no siempre se puede conseguir.

La solución al primero de los problemas se basó en la creación de nuevas aplicaciones cliente tipo *softphone* utilizando otro lenguaje de programación, concretamente JavaScript. Para ello se utilizó un cliente SIP JavaScript *freeware* que se modificó para poder acceder al contenido de los paquetes SIP y mostrarlos en una nueva interfaz personalizada. La solución al segundo problema fue utilizar un servidor proxy real en el dispositivo del estudiante para las pruebas locales y otro servidor accesible en RIUJA para los experimentos remotos (**Figura 144**).

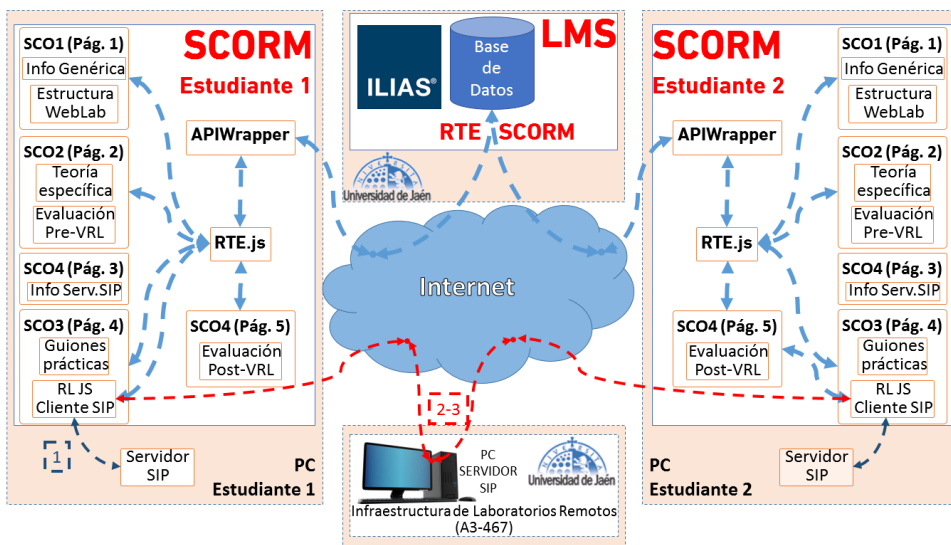


Figura 144. Estructura y comunicaciones de Lab remoto SIP JavaScript

Ante estos cambios se decidió crear una versión renovada del WebLab de Comunicaciones SIP [307]. El WebLab ha cambiado mostrando una nueva imagen y estructura basada en 5 páginas al añadir una página tras la teoría y test-pre-RL del SCO2 que sirve para explicar la instalación y configuración del servidor *proxy* SIP en el dispositivo local (**Figura 145**). También han cambiado los contenidos de teoría y el test pre-RL y los experimentos que aunque son

parecidos, resultan distintos evaluando en forma de preguntas de test. Otra ventaja que presenta el cliente SIP es que permite establecer comunicaciones vocales y con vídeo.



Figura 145. Captura parcial de pantalla de SCO3 del WebLab de comunicaciones SIP JS

Las prácticas que se presentan ahora a los estudiantes son las siguientes:

- **Practica 1: Registro en servidor local.** El estudiante se debe registrar en el servidor local que ha debido instalar en su mismo dispositivo (normalmente PC) con una de las identificaciones que haya creado. En el interfaz del cliente SIP podrá ver los mensajes que se envían para realizar este registro (**Figura 146**). Se le realizarán 7 preguntas relacionadas con estos mensajes que deberá superar y con las que podrá obtener hasta 3,5 puntos.
- **Practica 2: Registro en servidor remoto y comunicación SIP.** En esta práctica deben ponerse de acuerdo dos estudiantes para registrarse en el servidor *proxy* SIP que existe en *weblab.ujaen.es*, cada uno con las credenciales personales que les proporcione el profesor. Deben intercambiarse mensajes usando el interfaz SIP. Se les presentarán 6 preguntas, 3 de ellas referidas a los mensajes generados por los envíos y las otras 3 a los mensajes generados por los mensajes recibidos. Podrá obtener hasta 3 puntos.

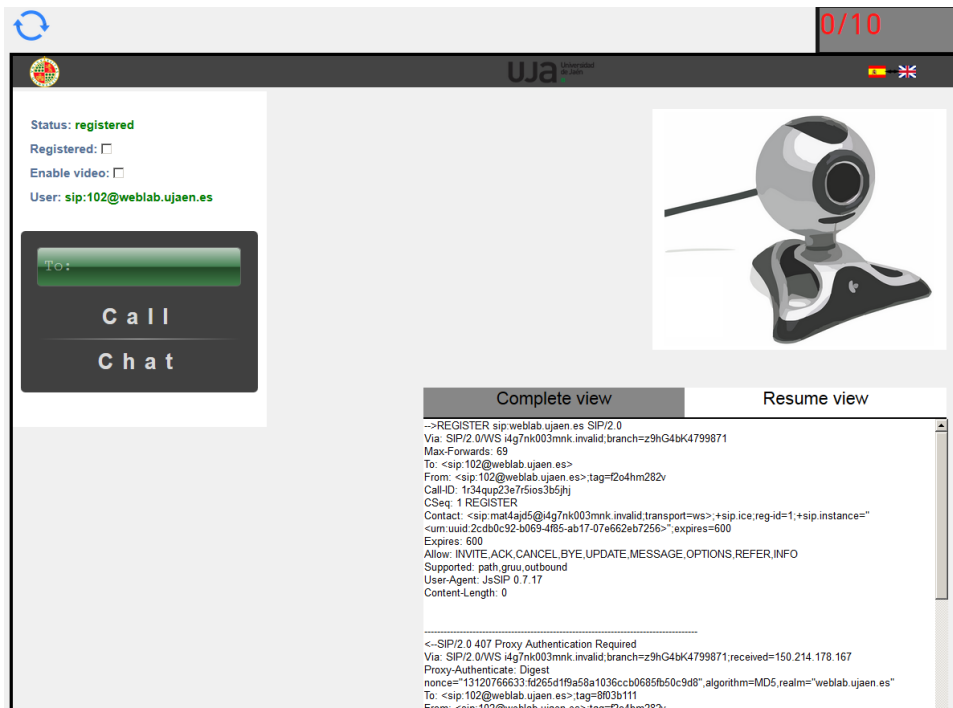


Figura 146. Interfaz de cliente SIP JavaScript con vista de mensajes SIP completa

- Practica 3: Comunicaciones SIP vocales.** En la práctica final los estudiantes deben establecer llamadas de VoIP SIP o videollamadas SIP usando el servidor *proxy* SIP de weblab.ujaen.es y las mismas credenciales usadas en la práctica anterior. Los estudiantes deben responder 3 preguntas cuando realicen la llamada y 4 cuando la respondan, en total todos deberán responder a 7 preguntas referidas a los contenidos de los mensajes SIP generados que pueden ver en la interfaz del cliente (**Figura 146**). Los estudiantes podrán obtener hasta 3,5 puntos.

En total se pueden obtener hasta 10 puntos, si se logran más de 5 se podrá pasar a la página siguiente (SCO4) en la que se presenta un test post-RL.

6.4. Laboratorios Híbridos

Siguiendo los principios y herramientas mostrados en el resto de capítulos también se ha promovido la creación de WebLabs en la modalidad híbrida. Se consideran como tales ya que en la misma estructura SCORM se ha

contemplado el uso de aplicaciones de VL y de RL. Se han creado dos laboratorios híbridos basados en el uso y control de “invernaderos” controlados remotamente. Los invernaderos invernadero son en realidad peceras en la que se ha incluido una serie de sensores y actuadores utilizando placas de desarrollo de *hardware* y *software* Arduino MEGA. Para aprovechar el espacio de la infraestructura de laboratorios remotos descrita en la sección 6.3.1 y aprovechar recursos los dos laboratorios hacen uso de un mismo PC servidor, aunque debido a sus dimensiones ocupan los espacios definidos como Laboratorio remoto 1 y 2 en la balda superior, como puede verse en la **Figura 147**.

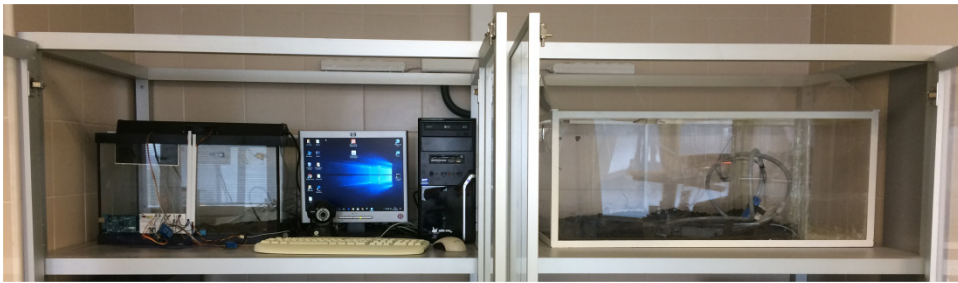


Figura 147. Equipos remotos de laboratorios híbridos en balda superior de infraestructura

La arquitectura y el esquema de comunicaciones de ambos sistemas se muestran en la **Figura 148**.

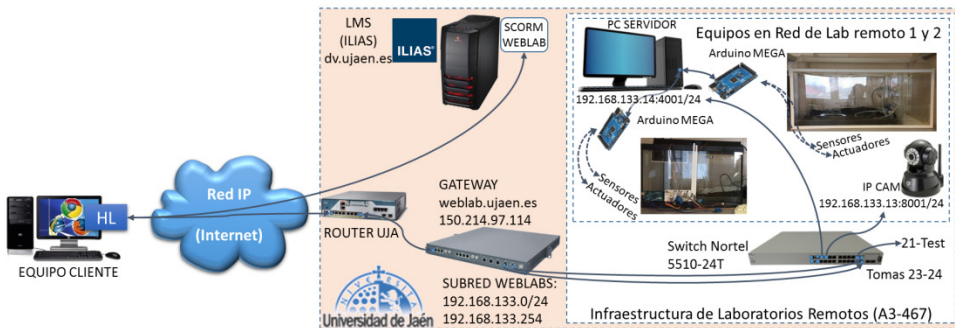


Figura 148. Arquitectura y Sistema de comunicaciones de los Laboratorios híbridos

El primero de los laboratorios fue enfocado a estudios tecnológicos de grados en Ingeniería Industrial Eléctrica, Electrónica, Mecánica y Organización Industrial mostrando un control PID de temperatura, se explicará brevemente en el apartado 6.4.1. El segundo se centró en estudios de Ciencias de la Educación para ser utilizado en asignaturas del Grado en Educación Primaria y el Máster de

Formación de Profesorado mostrando análisis de Fotosíntesis, se explicará en el apartado 6.4.2.

6.4.1. Control de Temperatura sobre un Invernadero Remoto

Este es el primer laboratorio híbrido construido con la metodología y herramientas propuestas en este trabajo de tesis. El WebLab tiene forma de paquete SCORM de 5 páginas, todas las páginas realizan comunicaciones con el LMS intercambiando datos del modelo RTE y SN de SCORM. Existen 2 aplicaciones en forma de *applet* de Java, una es un VL y la otra un RL, ambas se comunican con el LMS haciendo uso del paquete Java *scormRTE.jar*. Los objetivos del laboratorio son: a) conocer y simular sistemas de primer orden, b) conocimiento básico de control automático y su aplicación a la automatización industrial, c) estudiar y diseñar un controlador PID y d) optimizar el controlador implementado. Para obtener información más completa sobre este WebLab consultar [308]. Este laboratorio tiene la estructura y comunicaciones mostrada en la **Figura 149**.

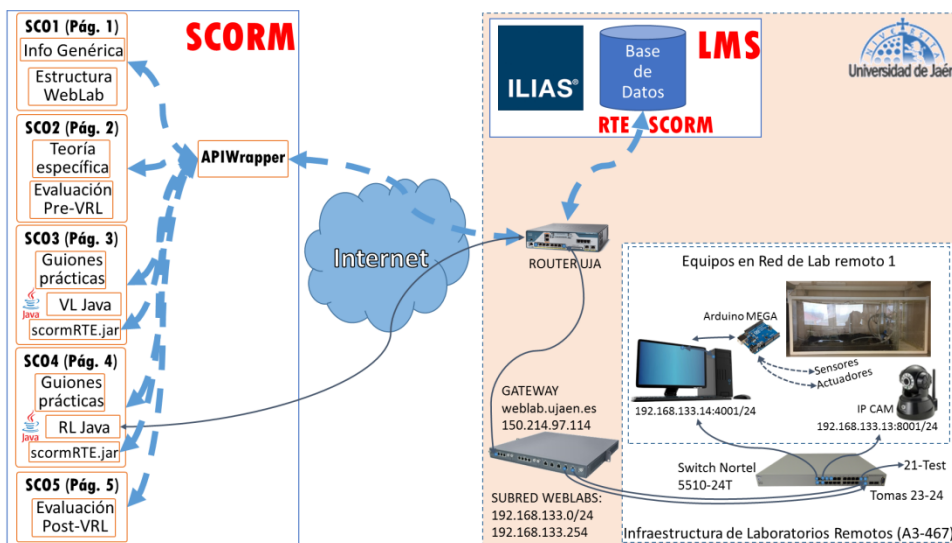


Figura 149. Estructura y comunicaciones del laboratorio híbrido de control PID de temperatura

La metodología docente y los elementos que lo componen es muy similar a los laboratorios mostrados hasta ahora, diferencia más sustancial consiste en la inclusión de 2 aplicaciones de laboratorios en forma de *applet* de Java. El laboratorio en su conjunto contiene 3 prácticas:

Práctica 1. Control PID de temperatura. El VL de la página 3 presenta la primera práctica, que consiste en la sintonización del PID conforme a los requerimientos pedidos (relación de amortiguamiento = 4). El interfaz del VL se puede ver en la **Figura 150**.

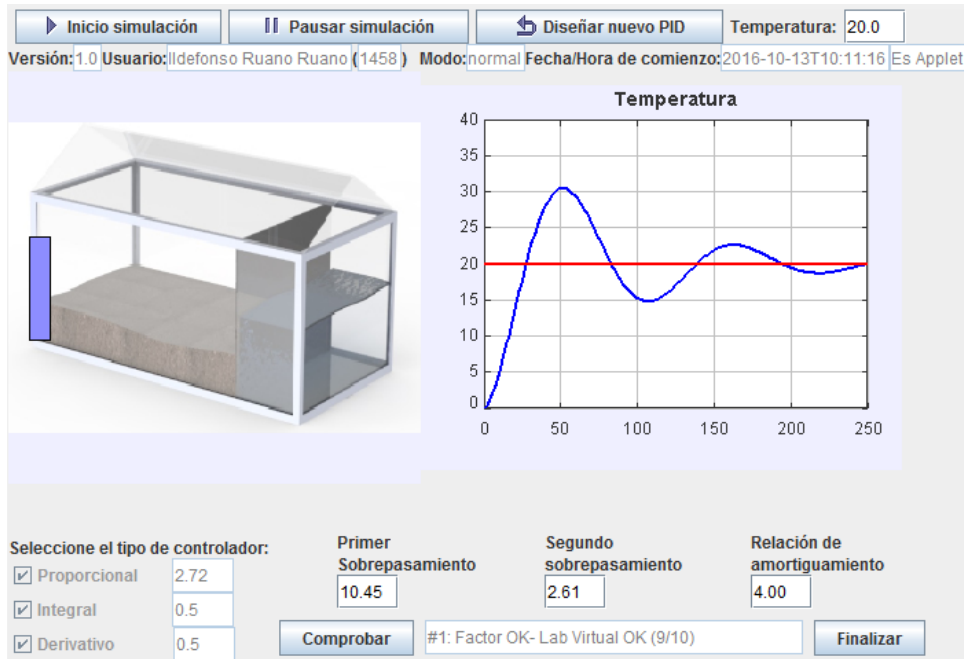


Figura 150. Interfaz de VL en forma de applet Java para el control PID de la temperatura

Las prácticas 2 y 3 se realizan en la página 4, donde se encuentra el *applet* Java del RL.

Práctica 2. Manual de RL. Esta práctica consiste en dominar la interfaz que se ofrece en el software del RL (**Figura 151**). Se deben utilizar todos los elementos para obtener, al menos, la puntuación mínima que permite acceder a la práctica siguiente.

Práctica 3. Interacción con actuadores. Consiste simplemente en interactuar con los actuadores del sistema invernadero remoto y observar como evolucionan las variables medidas por medio de los sensores. Los actuadores que existen son una bomba de riego, un sistema de iluminación y un ventilador. Además de temperatura (exterior e interior al invernadero) existen sensores de humedad exterior/interior/terreno (miden la humedad fuera del invernadero, en el interior del invernadero y del terreno existente dentro del invernadero), de oxígeno, de dióxido de carbono y de luminosidad.

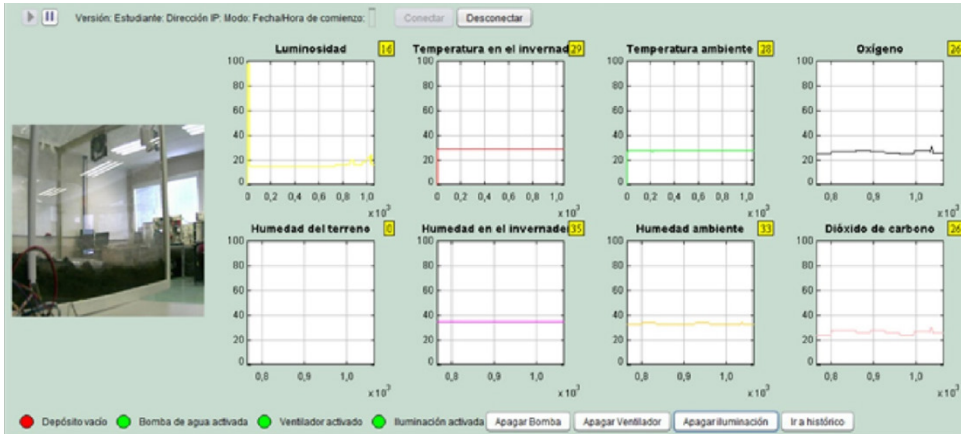


Figura 151. Interfaz de RL en forma de applet Java para el control remoto de actuadores y visión de valores en los sensores del invernadero

6.4.2. Fotosíntesis

Este WebLab es el segundo que se crea híbrido construido con la metodología y herramientas propuestas en este trabajo de tesis. Al igual que el anterior, el WebLab tiene forma de paquete SCORM de 5 páginas, todas las páginas realizan comunicaciones con el LMS intercambiando datos del modelo RTE y SN de SCORM y existen 2 aplicaciones en forma de *applet* de Java, VL y RL, que se comunican con el LMS haciendo uso del paquete Java scormRTE.jar. El laboratorio trata el tema de la fotosíntesis, un contenido curricular básico en la Educación Primaria y por tanto fundamental para los futuros maestros a los que va dirigido este laboratorio. El objetivo básico es dominar toda la temática relacionada con la fotosíntesis. Para la construcción de este laboratorio se ha seguido una concepción pedagógica diferente a la utilizada hasta el momento. La primera página contiene una breve descripción del laboratorio que lo enmarca en los estudios al igual que se ha hecho en todos los laboratorios mostrados hasta ahora. Sin embargo el contenido del resto de páginas no coincide con lo explicado hasta ahora. Para obtener una información más completa del mismo consultar [309]. Este laboratorio tiene la estructura y comunicaciones mostrada en la **Figura 152**.

La página 2, en vez de mostrar una teoría y un test pre-HL contiene un cuestionario de ideas previas cuyo objetivo es explorar las ideas previas que tiene el alumno sobre la fotosíntesis. En la página 3 se presentan experimentos

virtuales y remotos relacionados con la fotosíntesis. Solamente se presenta un experimento virtual utilizando el VL con el interfaz mostrado en la **Figura 153**.

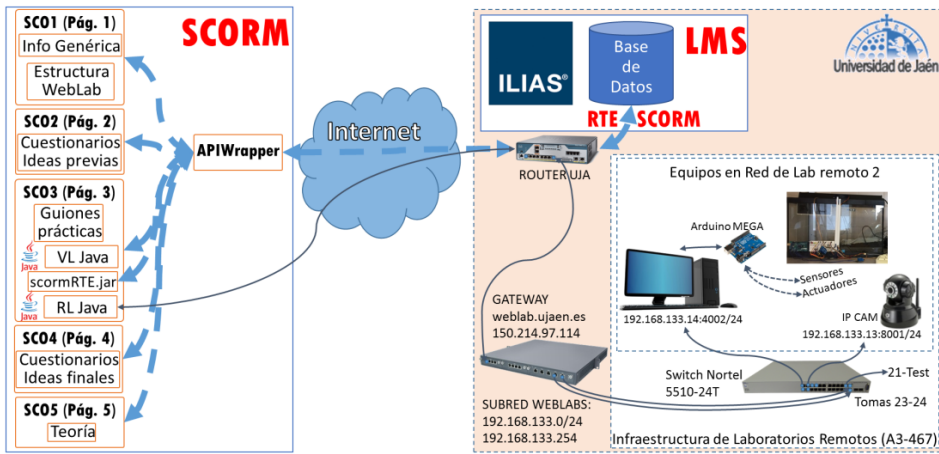


Figura 152. Estructura y comunicaciones del laboratorio híbrido de fotosíntesis

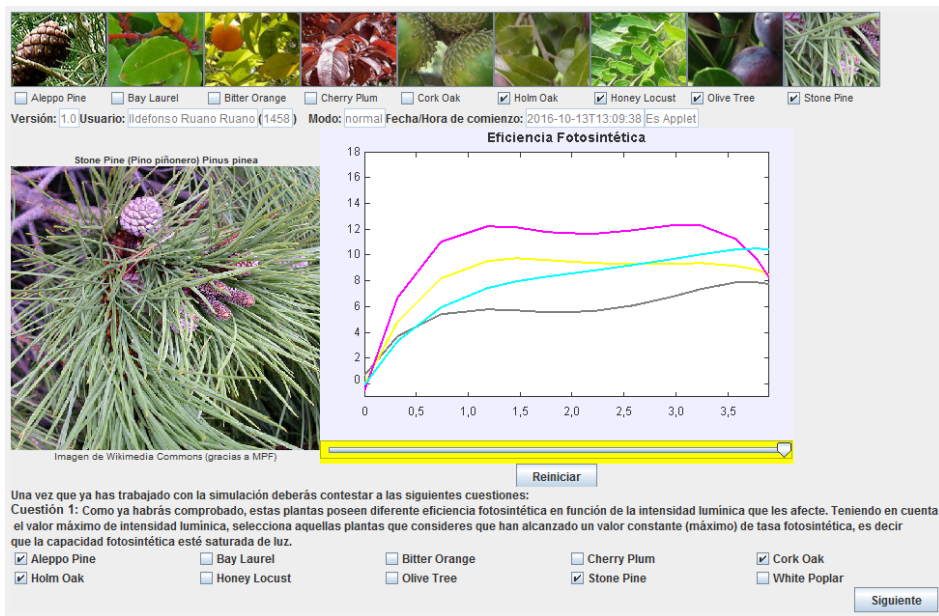


Figura 153. Interfaz de VL en forma de applet Java para fotosíntesis

El simulador muestra una gráfica en la que se dibuja la curva de eficiencia fotosintética (eje Y) en función de la intensidad lumínica (eje X) para las especies que se señalen. Se debe responder a una pregunta de opción múltiple, respuesta múltiple seleccionando las especies cuya curva de eficiencia fotosintética cumple una determinada característica.

El laboratorio remoto se basa en una serie de 3 experimentos a realizar con el *applet* de RL mostrado en la **Figura 154**, cuyo panel inferior va cambiando en función del experimento en curso. Los tres experimentos tratan el control remoto de la iluminación, riego y climatización del invernadero respectivamente, debiendo observar después el efecto que se tiene mediante la visualización de gráficas y responder a las preguntas indicadas.

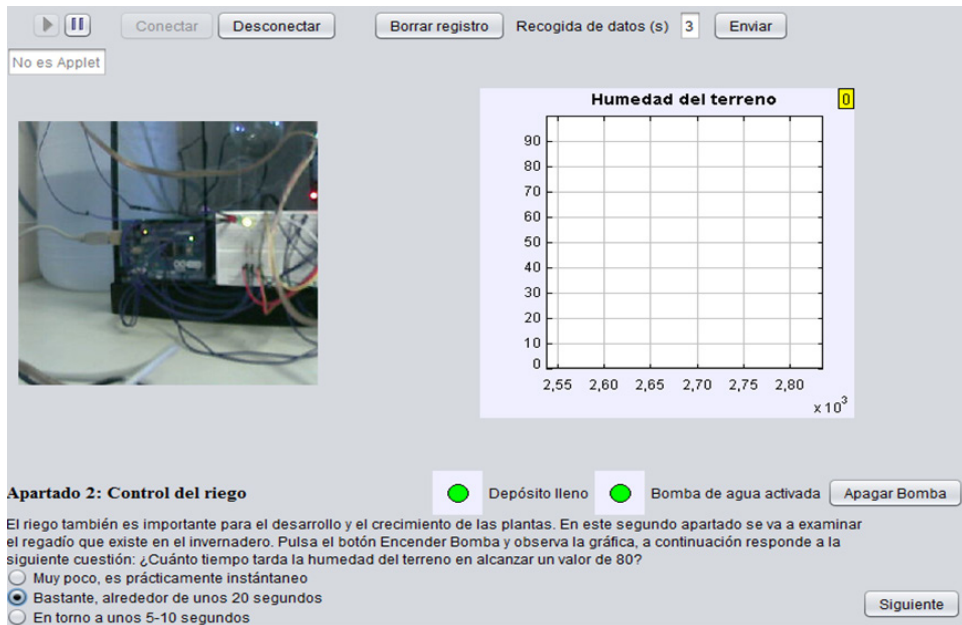


Figura 154. Interfaz de RL en forma de *applet* Java para el control remoto de actuadores y visión de gráficas de parámetros en WebLab Fotosíntesis

La página 4 del WebLab incluye cuestiones de ideas finales que sirven para comprobar como el trabajo realizado con la simulación y el RL ha cambiado las ideas preconcebidas que se cuestionaron en la página 2 y comprobar el nivel de conocimientos alcanzado.

Finalmente, la página 5 presenta contenidos teóricos que dan respuesta a todas las preguntas que se pueden plantear.

Cap.7

7. Conclusiones y Líneas de Futuro

7.1. Conclusiones

7.2. Líneas de Futuro

Para finalizar, en este capítulo se resumen los resultados más importantes que se han alcanzado en este trabajo de tesis, y se muestran las conclusiones más importantes que se derivan de los mismos. También se analizan las posibles líneas de futuro que se pueden desarrollar a partir de lo expuesto en este documento, algunas de las cuales ya se están llevando a cabo con el fin de avanzar y mejorar lo realizado o adaptarse al avance imparable de las nuevas tendencias y estándares tecnológicos.

7.1. Conclusiones

Este trabajo de tesis constituye una fase más de una trayectoria docente, innovadora e investigadora en docencia virtual o *e-learning*. Para poder realizar esta tesis, antes de empezar la etapa que concluye con esta publicación, se partió de una experiencia como profesor de asignaturas totalmente virtuales en el Campus Andaluz Virtual (CAV) [72], [310], e innovador/investigador en distintas facetas de la docencia virtual que dieron su fruto en comunicaciones de revistas, congresos nacionales e internacionales:

- Evaluación de las acciones formativas de *e-learning* [311], [312], [313], [314].
- Búsqueda de la calidad en *e-learning* mediante el uso de guías de estudio adaptadas [73], [315], [316], [74].
- Análisis y desarrollo de nuevas pruebas de evaluación online automatizadas en LMSs [317], [318].
- Cursos y metodologías para formación del profesorado universitario en la modalidad de b-learning [319], [320], [321].
- Desarrollo de redes de colaboración interuniversitaria en innovación educativa [322], [323].
- Creación de mini-vídeos docentes de bajo coste [324].

La aplicación de los conocimientos derivados de estas temáticas, pedagógicos y tecnológicos, se han aplicado al tema central de este trabajo de tesis que no es otro más que la integración avanzada de laboratorios online en los LMSs.

El análisis bibliográfico realizado ha constatado la **importancia de los laboratorios *online* y la integración de los mismos con los LMSs en la investigación actual**. Aunque no es el tema que presente mayor número de artículos científicos en el campo de los laboratorios *online*, si es uno de los más prolíficos. Esto demuestra el interés que existe en lograr la integración Lab-LMS, debido a las múltiples ventajas que se pueden obtener. El mismo análisis muestra que para lograr **las diferentes integraciones Lab-LMS se han implementado aplicando múltiples soluciones**. Algunas de estas soluciones se han basado en el uso de estándares de *e-learning*, como SCORM, LTI, xAPI o *cmi5*. Muchos de los estándares de *e-learning* han surgido en los últimos años y algunos de ellos aún no se han terminado de definir, están en continua evolución.

La clasificación de modos de integración incluida en este trabajo muestra las diferentes formas en que se puede producir la implementación de la integración Lab-LMS. Cada modo presenta una serie de ventajas e inconvenientes; **una integración avanzada Lab-LMS permite conseguir laboratorios *online* que presentan el mayor conjunto de ventajas posible**. Esto se logra cuando existe intercambio de información entre el laboratorio y el LMS.

El uso de los estándares SCORM permite crear laboratorios *online* logrando una integración avanzada Lab-LMS. La sub-especificación RTE de los estándares SCORM sirve para lograr el intercambio de datos Lab-LMS eficientemente. El intercambio de información Lab-LMS se ve restringido a los elementos de los modelos de datos SN y RTE, algo que limita en parte las posibilidades, por ejemplo, la obtención de resultados en formato distinto de texto como pueda ser audio, gráfico o vídeo. Sin embargo, permite obtener laboratorios que presentan muchas características deseables: identificación de estudiantes, presentación de un entorno conocido, personalización de experimentos, creación de secuencias de aprendizaje con recursos del LMS y en el propio paquete SCORM, portabilidad y reutilización en diferentes LMSs, evaluaciones automatizadas y personalizadas, seguimiento de resultados en el LMS y el almacenamiento de calificaciones en el LMS entre otras.

La complejidad que presenta el uso de los estándares SCORM para crear cualquier tipo de contenido, y laboratorios *online* en particular, es un hándicap importante. Aunque por otro lado, esta complejidad se ve simplificada por el uso de las herramientas creadas en el ámbito de esta tesis:

- *scormRTE*, el paquete Java de comunicaciones facilita, a los programadores Java de LoL, la implementación de las comunicaciones Lab-LMS abstrayéndoles del uso de JavaScript.
- *RTE.js*, la librería JavaScript de comunicaciones facilita, a los creadores de contenido SCORM y a los programadores JavaScript de LoL, la implementación de las comunicaciones Lab-LMS.
- Los documentos de ayuda asisten a los programadores para entender el uso de *scormRTE* y *RTE.js*.
- Las librerías JavaScript para creación de *tests* automatizados y los ficheros CSS asociados a las mismas facilitan la creación de evaluaciones automatizadas y reutilizar recursos ya existentes (preguntas en formato QTI). Presentan ciertas limitaciones referidas a los tipos de preguntas de

test que pueden usarse, ya que sólo permite cuatro tipos de preguntas, pero son las más utilizadas normalmente.

Para concluir respecto a **las herramientas presentadas en este trabajo**, se puede afirmar que **simplifican la creación de WebLabs, ahorrando tiempo de trabajo y posibilitando la presentación de códigos más claros y sencillos.**

La metodología de creación de laboratorios *online* incluida en este trabajo muestra los pasos, los actores que intervienen en cada uno y los resultados que se deben obtener en cada paso, para conseguir laboratorios efectivos desde el punto de vista del aprendizaje. Esta metodología no puede asegurar por sí misma que los laboratorios, obtenidos siguiendo los pasos y consejos incluidos en la misma, presenten todas las características deseables en un laboratorio. Sin embargo, **la metodología propuesta ayuda a los creadores de laboratorios *online* a recordar cuestiones esenciales que, si son seguidas y tenidas en cuenta con las particularidades propias de cada implementación, mejoran las prestaciones mostradas por los laboratorios.**

Los laboratorios *online* con integración avanzada basada en SCORM, creados siguiendo la metodología propuesta y utilizando las herramientas presentadas en este trabajo, que se han presentado a los alumnos de enseñanzas regladas de la UJA en el LMS ILIAS han ofrecido unas prestaciones muy buenas. Los resultados docentes de los estudiantes que han participado en estas experiencias han sido siempre positivos en diferentes aspectos: un porcentaje muy alto de los alumnos que han trabajado en los mismos han obtenido muy buenos resultados (nota de laboratorio al menos un notable alto) y buen resultado en pruebas finales del curso cuando se ha realizado la comparativa (mejor que los estudiantes que no lo hicieron). En estos laboratorios también se han realizado encuestas de satisfacción que han resultado muy positivas siempre con notas dentro del entorno del notable. Estos laboratorios han mostrado la mayoría de las ventajas que se pueden obtener de la integración avanzada. Por estos motivos se puede concluir que **siguiendo la metodología propuesta y utilizando las herramientas desarrolladas en el ámbito de este trabajo se pueden obtener WebLabs con integración avanzada basada en SCORM que son efectivos y presentan características muy deseables.**

7.2. Líneas de Futuro

Este proyecto de tesis ha mantenido un carácter evidentemente práctico, por la temática y la dependencia tecnológica de los contenidos no se puede pretender que sea algo cerrado ni finalizado. La obsolescencia tecnológica obliga a adaptar los sistemas sin más remedio. Por ello es necesario contemplar estas adaptaciones y mejoras como trabajos a realizar en un futuro cercano con el fin de mantener el correcto funcionamiento de todo lo conseguido. Teniendo en cuenta esto, existe un amplio abanico de posibilidades a la hora de dar una continuación a este trabajo de tesis. A continuación se explican algunas de las líneas de trabajo futuro más importantes.

Uso de otros estándares para la Comunicación Lab-LMS. El método utilizado para lograr la comunicación Lab-LMS, y con ello la integración avanzada, ha sido el uso de la sub-especificación RTE de SCORM. Constituye una solución viable y efectiva. Sin embargo, existen bastantes voces críticas sobre el uso de SCORM debido a los problemas planteados en la sección 2.1.1.2. Algunos de estos problemas se han solventado o minimizado con la aparición de las herramientas señaladas en el capítulo 4 de esta tesis pero la existencia de otros estándares de *e-learning* más modernos y avanzados como [LTI](#) [104], [xAPI](#) [100] o [cmi5](#) [105] hace que merezca la pena investigar sus posibilidades como sustituto de RTE para lograr las comunicaciones que permitan la integración avanzada Lab-LMS.

Establecimiento de Léxico Común para Laboratorios. Esta línea de futuro está muy relacionada con la anterior, en la que se ha planteado la posibilidad de utilizar otro estándar para las comunicaciones Lab-LMS. La sub-especificación RTE de SCORM define un modelo de datos para la comunicación SCORM-LMS, esto limita el número y tipo de datos que pueden intercambiarse, pero asegura que en todos los sistemas se va a disponer al menos de los que se han definido. Todos los estándares señalados se centran en la comunicación, pero no en los datos que se van a intercambiar el Lab y el LMS. En caso de utilizar uno de los estándares alternativos a SCORM habría que definir un modelo de datos a intercambiar entre un Lab y un LMS. Sin embargo, esto puede dar lugar a una multitud de propuestas que añadieran más caos y dificultades. Por eso sería bueno que existiese una definición, al menos mínima, de un modelo de datos que se use para los intercambios de información Lab-LMS independientemente del estándar utilizado para lograr los intercambios. Esto además, posibilitaría la creación de una capa envolvente de las comunicaciones que permitiera utilizar diferentes estándares sin tener que cambiar el código del laboratorio más que

en una variable o argumento para indicar el método de comunicación o estándar deseado. Aumentando la capacidad de reutilización de los laboratorios en distintos LMSs.

Ampliar Posibilidades de Herramientas. Las [librerías JavaScript desarrolladas](#) se han enfocado a facilitar la comunicación Lab-LMS y la construcción de *tests* interactivos. Se plantean dos líneas de futuro relacionadas con la mejora de estas librerías:

- Las versiones actuales permiten trabajar con comunicaciones SCORM2004 y SCORM1.2. Se pueden hacer ampliaciones de las mismas para soportar otro tipo de estándares intentando mantener la nomenclatura de funciones actual para evitar o minimizar el número de cambios en los laboratorios actuales.
- El número de tipos de preguntas soportado por las librerías de construcción automática de *tests* con comunicaciones LMS es 4: preguntas de opción múltiple (respuesta simple y respuesta múltiple), relleno de huecos y numérica. Se puede ampliar para lograr soporte de más tipos de preguntas como ordenación vertical u horizontal, unir parejas, de mapa de imagen, de añadir subconjunto, de fórmula, etc. Se puede hacer una mejora de la librería para soportar otros tipos de preguntas y mejorar las existentes, por ejemplo, añadiendo la posibilidad de definir opciones de respuesta de forma gráfica.

Conversión Java-JavaScript. En la sección 4.1 ya se comentó el dilema Java vs JavaScript y se explicó la necesidad de realizar los nuevos desarrollos utilizando JavaScript y olvidarse de Java debido a su obsolescencia en su forma de *applets* incrustadas en páginas Web. Se prevé que para el año 2017 Java deje de ser soportado por los navegadores más extendidos. Esto obliga a migrar todos los desarrollos hechos con Java a JavaScript si se quieren usar con normalidad, todos los trabajos relacionados con esta temática suponen otra línea de futuro.

Integración EjsS. Se están manteniendo comunicaciones con los desarrolladores del software *Easy Java/JavaScript Simulations* (EJS) para estudiar la posibilidad de integrar en el mismo las herramientas desarrolladas en el ámbito de esta tesis. Esto permitiría que los programadores que usen EJS para crear las aplicaciones VRL en JavaScript puedan usar directamente las funciones de las librerías de comunicaciones y creación de *tests* en sus códigos y que se incluya una opción para exportar sus VRL incrustados en un paquete SCORM basado en

una de las plantillas explicadas en la sección 4.5.2. De este modo el paquete SCORM generado por EJS podrá ser importado en un LMS incluyendo un VRL con comunicaciones SCORM y *tests* interactivos automatizados.

Desarrollo de Nuevos Laboratorios. Los contenidos y herramientas generados en esta tesis facilitan la creación de nuevos laboratorios online. En función de las necesidades docentes y con el espíritu de mejora de la calidad de la educación se deben estudiar las posibilidades de creación de nuevos laboratorios y acometer estas tareas cuando sean factibles.

Efectividad de Laboratorios. La última línea de futuro de este trabajo de tesis se refiere a un tema necesario para poder comprobar la calidad de los laboratorios online. Ya se ha comentado que es un tema complicado sobre el que no existe una metodología clara y precisa sobre cómo debe realizarse al existir distintos métodos y criterios como son comprobar el nivel de conocimiento alcanzado tras una propuesta de aprendizaje, el nivel de satisfacción dado por los participantes o la aplicación de plantillas que definan las características y propiedades de los laboratorios de forma objetiva. En cualquier caso depende del laboratorio a evaluar y debido a la gran heterogeneidad existente no siempre se podrá alcanzar unanimidad en el establecimiento de un método concreto. Probablemente una combinación de las líneas dadas fuera lo más correcto aunque para eso se requiere un estudio más avanzado.

Referencias

- [1] S. Dormido, "Control learning: present and future," *Annu. Rev. Control*, vol. 28, no. 1, pp. 71–93, Jan. 2004.
- [2] L. D. FEISEL and A. J. ROSA, "The Role of the Laboratory in Undergraduate Engineering Education," *J. Eng. Educ.*, pp. 121–130, 2005.
- [3] S. R. Singer, M. L. Hilton, and H. A. Schweingruber, *America's Lab Report*. Washington, D.C.: National Academies Press, 2005.
- [4] N. J. Nersessian, "Conceptual change in science and in science education," *Synthese*, vol. 80, no. 1, pp. 163–183, Jul. 1989.
- [5] M. P. Clough, "Using the laboratory to enhance student learning," *Learn. Sci. Sci. Learn.*, no. JANUARY 2002, pp. 85–94, 2002.
- [6] J. E. Corter, J. V. Nickerson, S. K. Esche, C. Chassapis, S. Im, and J. Ma, "Constructing reality: A Study of Remote, Hands-On, and Simulated Laboratories," *ACM Trans. Comput. Interact.*, vol. 14, no. 2, p. 7–es, Aug. 2007.
- [7] J. E. Corter, J. V. Nickerson, S. K. Esche, and C. Chassapis, "Remote versus hands-on labs: a comparative study," in *34th Annual Frontiers in Education, 2004. FIE 2004.*, 2004, pp. 595–599.
- [8] S. Chen, "The view of scientific inquiry conveyed by simulation-based virtual laboratories," *Comput. Educ.*, vol. 55, no. 3, pp. 1123–1130, Nov. 2010.
- [9] J. Ma and J. V. Nickerson, "Hands-on, simulated, and remote laboratories: A Comparative Literature Review," *ACM Comput. Surv.*, vol. 38, no. 3, p. 7–es, Sep. 2006.
- [10] H. Fry, S. Ketteridge, and S. Marshall, *A Handbook for Teaching and Learning in Higher Education*. 2009.
- [11] I. E. Allen, J. Seaman, R. Poulin, and T. T. Straut, "Online report card: Tracking online education in the United States," 2016.

- [12] E. Commission, "ICT in Education," *Digital Single Market. Digital Economy & Society*. [Online]. Available: <https://ec.europa.eu/digital-single-market/en/ict-education>.
- [13] CRUE UNIVERSIDADES ESPAÑOLAS, "UNIVERSITIC 2015. Analisis de las TIC en las Universidades Españolas," 2015.
- [14] S. Livingstone, "Critical reflections on the benefits of ICT in education," *Oxford Rev. Educ.*, vol. 38, no. 1, pp. 9–24, 2012.
- [15] R. Scherer, F. Siddiq, and T. Teo, "Becoming more specific: Measuring and modeling teachers' perceived usefulness of ICT in the context of teaching and learning," *Comput. Educ.*, vol. 88, pp. 202–214, Oct. 2015.
- [16] S. N. Dorogovtsev and J. F. F. Mendes, *Evolution of Networks: From Biological Nets to the Internet and WWW*, vol. 57. OUP Oxford, 2003.
- [17] D. Olin, J. Bourne, F. Mayadas, and S. Consortium, "Online engineering education : Learning anywhere , anytime.," *J. Eng. Educ.*, no. January, 2005.
- [18] L. Gomes and S. Bogosyan, "Current Trends in Remote Laboratories," *Ind. Electron. IEEE Trans.*, vol. 56, no. 12, pp. 4744–4756, 2009.
- [19] D. Lowe, M. De La Villefromoy, K. Jona, and L. Yeoh, "Remote laboratories: Uncovering the true costs," *2012 9th Int. Conf. Remote Eng. Virtual Instrumentation, REV 2012*, 2012.
- [20] C. Gravier, J. Fayolle, B. Bayard, M. Ates, and J. Lardon, "State of the Art About Remote Laboratories Paradigms – Foundations of Ongoing Mutations," *Int. J. Online Eng.*, vol. 4, no. 1, pp. 19–25, 2008.
- [21] A. Domínguez, J. Saenz-De-Navarrete, L. De-Marcos, L. Fernández-Sanz, C. Pagés, and J. J. Martínez-Herráiz, "Gamifying learning experiences: Practical implications and outcomes," *Comput. Educ.*, vol. 63, pp. 380–392, 2013.
- [22] J. C. Burguillo, "Using game theory and Competition-based Learning to stimulate student motivation and performance," *Comput. Educ.*, vol. 55, no. 2, pp. 566–575, 2010.
- [23] M. Abdulwahed and Z. K. Nagy, "The TriLab, a novel ICT based triple access mode laboratory education model," *Comput. Educ.*, vol. 56, no. 1, pp. 262–274, 2011.
- [24] M. I. of T. MIT, "The iLAB Project." [Online]. Available:

- <http://ilab.mit.edu/wiki>. [Accessed: 26-Jul-2016].
- [25] I. Gustavsson, K. Nilsson, J. Zackrisson, J. García-Zubía, U. Hernandez-Jayo, A. Nafalski, Z. Nedic, O. Gol, J. Machotka, M. I. Pettersson, T. Lago, and L. Hkansson, "On Objectives of Instructional Laboratories, Individual Assessment, and Use of Collaborative Remote Laboratories," *IEEE Trans. Learn. Technol.*, vol. 2, no. 4, pp. 263–274, Oct. 2009.
- [26] D. B. (coordinator), "LiLa - Library of Labs." [Online]. Available: <http://www.lila-project.org/>. [Accessed: 26-Jul-2016].
- [27] U. N. de E. a D. (UNED), "University Network of Interactive Labs (UNILabs)." [Online]. Available: <http://unilabs.dia.uned.es/>. [Accessed: 26-Jul-2016].
- [28] C. Maier and M. Niederstätter, "Lab2go – A Repository to Locate Online Laboratories," *Int. J. Online Eng.*, vol. 6, no. 1, pp. 12–17, 2010.
- [29] D. of B. and E. E.-U. de G. (DIBE), "Internet Shared Instrumentation Laboratory (ISILab)." [Online]. Available: <http://isilab.dibe.unige.it/>. [Accessed: 26-Jul-2016].
- [30] U. of S. A. (UniSA), "NetLab." [Online]. Available: <http://netlab.unisa.edu.au/>. [Accessed: 26-Jul-2016].
- [31] O. S. and M. G. at HPI, "Distributed Control Lab - DCL." [Online]. Available: <http://www.dcl.hpi.uni-potsdam.de/research/dcl/>. [Accessed: 26-Jul-2016].
- [32] U. of Deusto, "WebLab-Deusto." [Online]. Available: <http://weblab.deusto.es/website/>. [Accessed: 26-Jul-2016].
- [33] A. G. D. and S. A. F. (DSA), "The LabShare Project (Sahara)." [Online]. Available: <http://www.labshare.edu.au/project/>. [Accessed: 26-Jul-2016].
- [34] GOLC, "Global Online Laboratory Consortium (GOLC)." [Online]. Available: http://www.online-engineering.org/GOLC_about.php. [Accessed: 26-Jul-2016].
- [35] E. Guimarães, A. Maffeis, E. Guimaraes, J. Pereira, B. Russo, E. Cardozo, M. Bergerman, and M. F. Magalhaes, "REAL: a virtual laboratory for mobile robot experiments," *IEEE Trans. Educ.*, vol. 46, no. 1, pp. 37–42, 2003.
- [36] E. Fabregas, G. Farias, S. Dormido-Canto, S. Dormido, and F. Esquembre,

- “Developing a remote laboratory for engineering education,” *Comput. Educ.*, vol. 57, no. 2, pp. 1686–1697, Sep. 2011.
- [37] J. L. Guzmán, H. Vargas-Oyarzún, J. Sánchez, M. Berenguel, S. Dormido, and F. Rodríguez, “Education research in engineering studies: Interactivity, virtual and remote labs,” in *Distance Education Issues and Challenges*, 2007th ed., Nova Science Publishers, 2007, pp. 131–167.
- [38] B. Rubin, R. Fernandes, M. D. Avgerinou, and J. Moore, “The effect of learning management systems on student and faculty outcomes,” *Internet High. Educ.*, vol. 13, no. 1–2, pp. 82–83, Jan. 2010.
- [39] H. coates, R. James, and G. Baldwin, “A Critical Examination Of The Effects Of Learning Management Systems On University Teaching And Learning,” *Tert. Educ. Manag.*, vol. 11, no. 1, pp. 19–36, Mar. 2005.
- [40] W. R. Watson and S. L. Watson, “An Argument for Clarity: What are Learning Management Systems, What are They Not, and What Should They Become?,” *TechTrends*, vol. 51, no. 2, pp. 28–34, Mar. 2007.
- [41] C. Gravier, J. Fayolle, G. Noyel, A. Lelevé, and H. Benmohamed, “Distance learning: Closing the gap between remote labs and learning management systems,” *2006 1st IEEE Int. Conf. E-Learning Ind. Electron. ICELIE*, pp. 130–134, 2006.
- [42] L. de la Torre, “New Generation Virtual and Remote Laboratories: Integration Into Web Environments 2.0 With Learning Management Systems,” SPANISH OPEN UNIVERSITY (UNED), 2013.
- [43] F. Lerro, S. Marchisio, S. Martini, H. Massacesi, E. Perretta, A. Gimenez, N. Aimetti, and J. I. Oshiro, “Integration of an e-learning platform and a remote laboratory for the experimental training at distance in engineering education,” *2012 9th Int. Conf. Remote Eng. Virtual Instrumentation, REV 2012*, 2012.
- [44] E. Sancristobal Ruiz, M. Castro, J. Harward, P. Baley, K. DeLong, and J. Hardison, “Integration view of Web Labs and Learning Management Systems,” in *IEEE EDUCON 2010 Conference*, 2010, vol. 81, pp. 1409–1417.
- [45] I. Ruano, P. Cano, J. Gamez, and J. Gomez, “Advanced LMS Integration of SCORM Web Laboratories,” *IEEE Access*, pp. 1–1, 2016.
- [46] I. Ruano-Ruano, J. Gámez-García, and J. Gómez-Ortega, “Modos de integración de laboratorios online en sistemas de gestión de aprendizaje,” in *XXXVII Jornadas de Automática*, 2016, pp. 653–660.

-
- [47] T. Richter, Y. Tetour, and D. Boehringer, "Library of Labs - A European Project on the Dissemination of Remote Experiments and Virtual Laboratories," in *2011 IEEE International Symposium on Multimedia*, 2011, pp. 543–548.
- [48] Advanced Distributed Learning (ADL), "SCORM 2004 4th Ed. (Shared Content Object reference model)," 2009. [Online]. Available: <https://www.adlnet.gov/adl-research/scorm/scorm-2004-4th-edition/>. [Accessed: 21-Jun-2016].
- [49] I. Ruano-Ruano, J. Gómez-Ortega, J. Gámez-García, and E. Estévez-Estévez, "Herramienta basada en SCORM para la integración automática de Laboratorios Online en LMS," in *XXXIV Jornadas de Automática*, 2013, pp. 275–282.
- [50] I. Ruano-Ruano, J. Gámez-García, S. Dormido, and J. Gómez-Ortega, "A SCORM based package model for WebLabs," in *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, 2014, pp. 1–4.
- [51] I. Ruano Ruano, J. Gámez García, and J. Gómez Ortega, "Laboratorio Web SCORM de Control PID con Integración Avanzada," *Rev. Iberoam. Automática e Informática Ind. RIAI*, vol. 13, no. 4, pp. 472–483, Oct. 2016.
- [52] I. Ruano, J. Gamez, S. Dormido, and J. Gomez, "A Methodology to Obtain Learning Effective Laboratories with Learning Management System Integration," *IEEE Trans. Learn. Technol.*, pp. 1–1, 2016.
- [53] F. Esquembre, "Easy Java Simulations: a software tool to create scientific simulations in Java," *Comput. Phys. Commun.*, vol. 156, no. 2, pp. 199–204, Jan. 2004.
- [54] J. Gómez-Ortega, I. Ruano-Ruano, J. Gámez-García, J. R. Balsas-Almagro, and A. Sánchez-García, "Laboratorio Remoto de un sistema de péndulo invertido sobre carro de trayectoria lineal." Jaén, p. 105, 2014.
- [55] J. Gámez-García, I. Ruano-Ruano, J. Gómez-Ortega, A. Sánchez-García, E. Estévez-Estévez, M. Romero-Ariza, and A. Quesada-Armenteros, "Diseño, desarrollo, implementación y evaluación de Laboratorios Web." Universidad de Jaén, Jaén, p. 497, 2016.
- [56] Google, "Buscador Google." [Online]. Available: <https://www.google.es/>. [Accessed: 28-Jul-2016].
- [57] Google, "Google Académico." [Online]. Available: <https://scholar.google.es/>. [Accessed: 28-Jul-2016].

- [58] I. of E. and E. E. (IEEE), "IEEE Xplore." [Online]. Available: <http://ieeexplore.ieee.org>. [Accessed: 28-Jul-2016].
- [59] Elsevier, "Librería Digital Elsevier." [Online]. Available: <https://www.elsevier.com/>. [Accessed: 28-Jul-2016].
- [60] Mendeley, "Mendeley." [Online]. Available: <https://www.mendeley.com/>. [Accessed: 01-Aug-2016].
- [61] A. D. L. (ADL), "Advanced Distributed Learning (ADL)." [Online]. Available: <https://www.adlnet.gov/>. [Accessed: 28-Jul-2016].
- [62] C. E. de A. (CEA), "Comité Español de Automática (CEA)." [Online]. Available: <http://www.ceautomatica.es/>. [Accessed: 28-Jul-2016].
- [63] F. Esquembre, "Easy Java Simulations (EJS)." [Online]. Available: <http://fem.um.es/Ejs/>. [Accessed: 28-Jul-2016].
- [64] U. de Jaén, "LMS ILIAS de la UJA." [Online]. Available: <http://dv.ujaen.es>. [Accessed: 28-Jul-2016].
- [65] ILIAS, "Integriertes Lern-,Informations- und Arbeitskooperations-System (ILIAS)." [Online]. Available: <http://www.ilias.de>. [Accessed: 28-Jul-2016].
- [66] J. Hilera and R. Hoya, *Estándares de e-learning: Guía de consulta*, vol. 106, no. 493. 2010.
- [67] A. Berlanga Reyes, B. Fernández Manjón, I. Martínez Ortiz, P. Moreno Ger, and J. L. Sierra Rodríguez, "Estándares en e-learning y diseño educativo," 2011.
- [68] D. Lowe, G. Bharathy, B. Stumpers, and H. Yeung, "Laboratory lesson plans: Opportunities created by remote laboratories," in *2012 9th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2012, pp. 1–6.
- [69] G. Donzellini and D. Ponta, "The electronic laboratory: traditional, simulated or remote?," in *Advances on remote Laboratories and E ...*, L. Gomes and J. García-Zubía, Eds. Deusto (Spain): Deusto Publicaciones, 2006, pp. 223–247.
- [70] A. Konak, T. K. Clark, and M. Nasereddin, "Using Kolb's Experiential Learning Cycle to improve student learning in virtual computer laboratories," *Comput. Educ.*, vol. 72, pp. 11–22, 2014.
- [71] D. KOLB, "Experiential learning: Experience as the source of learning and development," in *Strategic Learning in a Knowledge Economy*, Elsevier,

- 1984, pp. 20–38.
- [72] I. Ruano-Ruano and L. García-Cabrera, “Experiencia didáctica sobre la asignatura del CAV ‘Internet, Fundamentos y Servicios,’” in *Buenas Prácticas en teleformación en las diez universidades andaluzas*, 1st ed., J. I. Aguaded and A. Infante, Eds. La Coruña (Spain): Junta de Andalucía, 2009, p. 417.
- [73] I. Ruano-Ruano, L. García-Cabrera, J. M. Ortega-Tudela, and M. Á. Peña-Hita, “Garantizar la Calidad de una Asignatura Virtual Mediante su Guía de Estudio,” *II Foro Buenas prácticas en Teleformación*, vol. 1, no. 2, 2009.
- [74] L. García-Cabrera, J. M. Ortega-Tudela, M. Á. Peña-Hita, I. Ruano-Ruano, and A. M. Ortiz, “The Quality of Virtual Learning: the Study Guide Importance,” *Pixel-Bit*, no. 37, pp. 77–92, 2010.
- [75] F. Lerro, S. Marchisio, E. Perretta, M. Plano, and M. Protano, “Using the Remote Lab of Electronics Physics (‘Laboratorio Remoto de Física Electrónica’) to Support Teaching and Learning Processes in Engineering Courses,” in *Using Remote Labs in Education*, J. G. Zubía and G. R. Alves, Eds. Deusto (Spain): Deusto Digital, 2011, pp. 211–230.
- [76] I. Ruano-Ruano, P. Cano-Marchal, J. Gámez-García, and J. Gómez-Ortega, “PID Control WebLab with LMS Integration Using SCORM,” *IFAC-PapersOnLine*, vol. 48, no. 29, pp. 301–306, 2015.
- [77] I. Ruano-Ruano, J. Gámez-García, and J. Gómez-Ortega, “Building SCORM embedded WebLabs with LMS interaction,” in *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, 2014, pp. 1–4.
- [78] I. Ruano-Ruano, J. Gómez-Ortega, J. Gámez-García, and E. Estévez-Estévez, “Integration of Online Laboratories - LMS via SCORM,” in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 3163–3167.
- [79] J.-E. Pelet, *E-Learning 2.0 Technologies and Web Applications in Higher Education*. IGI Global, 2014.
- [80] B. H. Khan, *Web Based Instruction*. New Jersey (USA): Educational Technology Publications, 1997.
- [81] D. Mioduser, R. Nachmias, O. Lahav, and A. Oren, “Web-Based Learning Environments,” *J. Res. Comput. Educ.*, vol. 33, no. 1, pp. 55–76, Sep. 2000.

- [82] N. Hoic-Bozic, M. Holenko Dlab, and V. Mornar, "Recommender System and Web 2.0 Tools to Enhance a Blended Learning Model," *IEEE Trans. Educ.*, vol. 59, no. 1, pp. 39–44, Feb. 2016.
- [83] A. J. Swart, "The effective use of a learning management system still promotes student engagement !," in *2016 IEEE Global Engineering Education Conference (EDUCON)*, 2016, no. April, pp. 40–44.
- [84] C. Vrasidas, "Issues of pedagogy and design in e-learning systems," *Proc. 2004 ACM Symp. Appl. Comput.*, pp. 911–915, 2004.
- [85] M. P. Prendes, "Plataformas de campus virtual con herramientas de software libre. Análisis comparativo de la situación actual en las universidades españolas," 2009.
- [86] SCOPEO, "Aproximación pedagógica a las plataformas open source en la universidad española," 2011.
- [87] G. Labs, "Google Trends." [Online]. Available: <https://www.google.es/trends/>. [Accessed: 03-Jul-2016].
- [88] Capterra, "Top LMS Software," 2016. [Online]. Available: <http://www.capterra.com/learning-management-system-software/>. [Accessed: 04-Aug-2016].
- [89] M. Aberdour, "Open source learning management systems," 2007.
- [90] Moodle, "Moodle Statistic." [Online]. Available: <https://moodle.net/stats/>. [Accessed: 12-Aug-2016].
- [91] Universidad de Jaén, "LMS ILIAS de Evaluación de la UJA." [Online]. Available: <http://ev.ujaen.es>. [Accessed: 06-Sep-2016].
- [92] B. Fernandez M, P. Moreno G, J. L. Sierra R, and I. Martínez O, "Uso de Estándares aplicados a TIC en Educación," *Design*, p. 187, 2008.
- [93] A. E. de N. y C. (Aenor), "Asociación Española de Normalización y Certificación (Aenor)." .
- [94] W. W. W. C. (W3C), "World Wide Web Consortium (W3C)." [Online]. Available: <https://www.w3.org/>. [Accessed: 12-Aug-2016].
- [95] AICC CMI Subcommittee and S. Bergstrom, "CMI Guidelines for Interoperability AICC." AICC, Aviation Industry Computer-Based Training Committee, p. 241, 1993.
- [96] O. Bohl, J. Scheuhase, R. Sengler, and U. Winand, "The sharable content

- object reference model (SCORM) - a critical review," in *International Conference on Computers in Education, 2002. Proceedings.*, 2002, vol. 1, pp. 950–951.
- [97] A. Parmar, "Paper Review on Sharable Content Object Reference Model (SCORM): Framework for E-learning Standard," in *2012 Second International Conference on Advanced Computing & Communication Technologies*, 2012, pp. 409–411.
- [98] C.-Y. Chang and S.-F. Cheng, "Implementing an SCORM-based e-learning system for a multinational enterprise," in *2015 IEEE International Conference on Consumer Electronics - Taiwan*, 2015, pp. 472–473.
- [99] Advanced Distributed Learning (ADL), "Experience API," 2013. [Online]. Available: <https://www.adlnet.gov/adl-research/performance-tracking-analysis/experience-api/>. [Accessed: 12-Aug-2016].
- [100] Advanced Distributed Learning (ADL), "xAPI 1.0.2." Advanced Distributed Learning Initiative (ADL), 2013.
- [101] J. Poltrack, N. Hruska, A. Johnson, and J. Haag, "The Next Generation of SCORM: Innovation for the Global Force," in *Interservice/Industry Training, Simulation, and Education Conference*, 2012.
- [102] Advanced Distributed Learning (ADL), "Experience API SCORM Profile." Advanced Distributed Learning Initiative (ADL), 2016.
- [103] IMS Global Learning Consortium (IMS), "IMS Global Learning Consortium (IMS)." [Online]. Available: <https://www.imsglobal.org/>. [Accessed: 12-Aug-2016].
- [104] IMS Global Learning Consortium, "Learning Tools Interoperability (LTI) v2.0," 2014. [Online]. Available: <https://www.imsglobal.org/specs/ltiv2p0>. [Accessed: 21-Jul-2016].
- [105] Advanced Distributed Learning (ADL), "Computer-Managed Instruction 5 (cmi5)," 2015. [Online]. Available: <https://adlnet.gov/adl-research/performance-tracking-analysis/cmi5/>. [Accessed: 21-Jul-2016].
- [106] Advanced Distributed Learning, "SCORM Users Guide for Instructional Designers," 2011.
- [107] Advanced Distributed Learning (ADL), "Content Aggregation Model (CAM) SCORM 2004 4th Ed. v1.1." p. 234, 2009.
- [108] Advanced Distributed Learning (ADL), "Run-Time Environment (RTE)

- SCORM 2004 4th Ed. v1.1." p. 244, 2009.
- [109] Advanced Distributed Learning (ADL), "Sequencing and Navigation (SN) SCORM 2004 4th Ed. v1.1." p. 234, 2009.
- [110] I. of E. and E. E. (IEEE), "1484.11.2-2003 IEEE Standard for Learning Technology-ECMA Script Application Programming Interface for Content to Runtime Services Communication," vol. 2004, no. September. IEEE, 2004.
- [111] B. Balamuralithara and P. C. Woods, "Virtual laboratories in engineering education: The simulation lab and remote lab," *Comput. Appl. Eng. Educ.*, vol. 17, no. 1, pp. 108–118, Mar. 2009.
- [112] V. Mateos, A. Gallardo, and T. Richter, "LiLa Booking System: Architecture and Conceptual Model of a Rig Booking System for On-Line Laboratories," *Int. J. Online Eng.*, vol. 7, no. 4, 2011.
- [113] A. Gallardo, T. Richter, P. Debicki, L. Bellido, V. Mateos, and V. Villagra, "A rig booking system for on-line laboratories," *2011 IEEE Glob. Eng. Educ. Conf.*, pp. 643–648, Apr. 2011.
- [114] P. P. Grube, D. Boehringer, T. Richter, N. Natho, C. Spiecker, C. Maier, and D. Zutin, "A Metadata Model for Online Laboratories," in *IEEE Global Engineering Education Conference (EDUCON)*, 2011, pp. 618–622.
- [115] R. Pastor, J. Sánchez, and S. Dormido, "Web-based virtual lab and remote experimentation using easy java simulations," in *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 2005, vol. 16, no. c, pp. 103–108.
- [116] C. Martin-Villalba, A. Urquia, and S. Dormido, "Development of an industrial boiler virtual-lab for control education using Modelica," *Comput. Appl. Eng. Educ.*, vol. 21, no. 1, pp. 36–45, Mar. 2013.
- [117] M. A. Bochicchio and A. Longo, "Extending LMS with collaborative remote lab features," *Proc. - 10th IEEE Int. Conf. Adv. Learn. Technol. ICALT 2010*, pp. 310–314, 2010.
- [118] J. García-Zubía, I. Angulo, L. Rodríguez-Gil, P. Orduna, and O. Dziabenko, "Boole-WebLab-Deusto: Integration of a remote lab in a tool for digital circuits design," *Proc. - Front. Educ. Conf. FIE*, pp. 848–854, 2013.
- [119] L. Rodriguez-Gil, P. Orduna, J. Garcia-Zubia, I. Angulo, and D. Lopez-de-Ipina, "Graphic technologies for virtual, remote and hybrid laboratories: WebLab-FPGA hybrid lab," in *2014 11th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2014, no.

February, pp. 163–166.

- [120] L. Rodriguez-Gil, J. Garcia-Zubia, P. Orduna, and D. Lopez-de-Ipina, "Towards new multiplatform hybrid online laboratory models," *IEEE Trans. Learn. Technol.*, vol. 1382, no. c, pp. 1–1, 2016.
- [121] J. M. Andujar, A. Mejias, and M. A. Marquez, "Augmented Reality for the Improvement of Remote Laboratories: An Augmented Remote Laboratory," *IEEE Trans. Educ.*, vol. 54, no. 3, pp. 492–500, Aug. 2011.
- [122] M.-P. Chen and B.-C. Liao, "Augmented Reality Laboratory for High School Electrochemistry Course," in *2015 IEEE 15th International Conference on Advanced Learning Technologies, 2015*, pp. 132–136.
- [123] J. Garcia-Zubia, D. Lopez-de-Ipina, and P. Orduna, "Towards a canonical software architecture for multi-device WebLabs," in *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.*, 2005, p. 6 pp.
- [124] R. Costa, G. Alves, and M. Zenha-Rela, "Reconfigurable weblabs based on the IEEE1451 Std.," *2010 IEEE Educ. Eng. Conf. EDUCON 2010*, pp. 1359–1366, 2010.
- [125] C. Ko, B. M. Chen, Y. Zhuang, and K. Chen Tan, "Development of a web-based laboratory for control experiments on a coupled tank apparatus," *IEEE Trans. Educ.*, vol. 44, no. 1, pp. 76–86, 2001.
- [126] R. Heradio, L. de la Torre, D. Galan, F. J. Cabrerizo, E. Herrera-Viedma, and S. Dormido, "Virtual and remote labs in education: A bibliometric analysis," *Comput. Educ.*, vol. 98, pp. 14–38, Jul. 2016.
- [127] V. Potkonjak, M. Gardner, V. Callaghan, P. Mattila, C. Guetl, V. M. Petrović, and K. Jovanović, "Virtual Laboratories for Education in Science, Technology, and Engineering: a Review," *Comput. Educ.*, vol. 95, pp. 309–327, Apr. 2016.
- [128] N. Rutten, W. R. Van Joolingen, and J. T. Van Der Veen, "The learning effects of computer simulations in science education," *Comput. Educ.*, vol. 58, no. 1, pp. 136–153, 2012.
- [129] A. Ferrero and V. Piuri, "A simulation tool for virtual laboratory experiments in a WWW environment," *IEEE Trans. Instrum. Meas.*, vol. 48, no. 3, pp. 741–746, Jun. 1999.
- [130] J. L. Chiu, C. J. DeJaegher, and J. Chao, "The effects of augmented virtual science laboratories on middle school students' understanding of gas

- properties," *Comput. Educ.*, vol. 85, pp. 59–73, Jul. 2015.
- [131] T. Jaakkola, S. Nurmi, and K. Veermans, "A comparison of students' conceptual understanding of electric circuits in simulation only and simulation-laboratory contexts," *J. Res. Sci. Teach.*, vol. 48, no. 1, pp. 71–93, Jan. 2011.
- [132] K. C. Trundle and R. L. Bell, "The use of a computer simulation to promote conceptual change: A quasi-experimental study," *Comput. Educ.*, vol. 54, no. 4, pp. 1078–1088, May 2010.
- [133] B. Dalgarno, A. G. Bishop, W. Adlong, and D. R. Bedgood, "Effectiveness of a Virtual Laboratory as a preparatory resource for Distance Education chemistry students," *Comput. Educ.*, vol. 53, no. 3, pp. 853–865, 2009.
- [134] J. C. Van Staden, M. W. H. Braun, and B. J. E. Van Tonder, "Computerized pendulum experiment for the introductory physics laboratory," *Comput. Educ.*, vol. 11, no. 4, pp. 281–292, Jan. 1987.
- [135] N. Kostaras, M. Xenos, and A. N. Skodras, "Evaluating Usability in a Distance Digital Systems Laboratory Class," *IEEE Trans. Educ.*, vol. 54, no. 2, pp. 308–313, May 2011.
- [136] E. D. Lindsay and M. C. Good, "Effects of Laboratory Access Modes Upon Learning Outcomes," *IEEE Trans. Educ.*, vol. 48, no. 4, pp. 619–631, Nov. 2005.
- [137] Z. Nedic, J. Machotka, and A. Nafalski, "Remote laboratories versus virtual and real laboratories," in *33rd Annual Frontiers in Education, 2003. FIE 2003.*, 2003, vol. 1, p. T3E_1-T3E_6.
- [138] J. V. Nickerson, J. E. Corter, S. K. Esche, and C. Chassapis, "A model for evaluating the effectiveness of remote engineering laboratories and simulations in education," *Comput. Educ.*, vol. 49, no. 3, pp. 708–725, 2007.
- [139] D. C. Sicker, T. Lookabaugh, J. Santos, and F. Barnes, "Assessing the Effectiveness of Remote Networking Laboratories," in *Proceedings Frontiers in Education 35th Annual Conference*, 2005, p. S3F–7–S3F–12.
- [140] K. Sun, Y. Lin, and C. Yu, "A study on learning effect among different learning styles in a Web-based lab of science for elementary school students," *Comput. Educ.*, vol. 50, no. 4, pp. 1411–1422, 2008.
- [141] J. R. Brinson, "Learning outcome achievement in non-traditional (virtual and remote) versus traditional (hands-on) laboratories: A review of the

- empirical research,” *Comput. Educ.*, vol. 87, pp. 218–237, Sep. 2015.
- [142] B. Aktan, C. A. Bohus, L. A. Crowl, and M. H. Shor, “Distance learning applied to control engineering laboratories,” *IEEE Trans. Educ.*, vol. 39, no. 3, pp. 320–326, 1996.
- [143] D. Harrison and J. M. Pitre, “Computers in a teaching laboratory: Just another piece of apparatus,” *Comput. Educ.*, vol. 12, no. 2, pp. 261–267, 1988.
- [144] A. Ferrero and V. Piuri, “A simulation tool for virtual laboratory experiments in a WWW environment,” *IEEE Trans. Instrum. Meas.*, vol. 48, no. 3, pp. 741–746, Jun. 1999.
- [145] Hong Shen, Zheng Xu, B. Dalager, V. Kristiansen, O. Strom, M. S. Shur, T. A. Fjeldly, Jian-Qiang Lu, and T. Ytterdal, “Conducting laboratory experiments over the Internet,” *IEEE Trans. Educ.*, vol. 42, no. 3, pp. 180–185, 1999.
- [146] S. G. Tzafestas, “Teaching Control and Robotics Using the Web,” in *Web-Based Control and Robotics Education*, S. G. Tzafestas, Ed. Springer, 2009, pp. 1–38.
- [147] J. Garcia-Zubia, P. Orduna, D. Lopez-de-Ipina, and G. R. Alves, “Addressing Software Impact in the Design of Remote Laboratories,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4757–4767, Dec. 2009.
- [148] L. A. Mendes, L. Li, P. H. Bailey, K. R. DeLong, and J. A. del Alamo, “Experiment lab server architecture: A web services approach to supporting interactive LabVIEW-based remote experiments under MIT’s iLab shared architecture,” in *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2016, no. February, pp. 293–305.
- [149] M. Stefanovic, V. Cvijetkovic, M. Matijevic, and V. Simic, “A LabVIEW-based remote laboratory experiments for control engineering education,” *Comput. Appl. Eng. Educ.*, vol. 19, no. 3, pp. 538–549, Sep. 2011.
- [150] D. Hercog, B. Gergic, S. Uran, and K. Jezernik, “A DSP-Based Remote Control Laboratory,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 6, pp. 3057–3068, Dec. 2007.
- [151] N. Ertugrul, “Towards Virtual Laboratories: A Survey of LabVIEW-based Teaching/ Learning Tools and Future Trends,” *Int. J. Eng. Educ.*, vol. 16, no. 3, pp. 171–180, 2000.

- [152] D. Chaos, J. Chacón, J. Lopez-Orozco, and S. Dormido, "Virtual and Remote Robotic Laboratory Using EJS, MATLAB and LabVIEW," *Sensors*, vol. 13, no. 2, pp. 2595–2612, Feb. 2013.
- [153] R. Costa-Castello, M. Vallés, L. M. Jiménez, L. Díaz-Guerra, A. Valera, and R. Puerto, "Integración de dispositivos físicos en un laboratorio remoto de control mediante diferentes plataformas: Labview, Matlab y C/C++," *Rev. Iberoam. Automática e Informática Ind.*, vol. 7, no. 1, pp. 23–34, Jan. 2010.
- [154] J. Chacon, H. Vargas-Oyarzún, G. Farias, J. Sanchez, and S. Dormido, "EJS, JIL Server, and LabVIEW: An Architecture for Rapid Development of Remote Labs," *IEEE Trans. Learn. Technol.*, vol. 8, no. 4, pp. 393–401, Oct. 2015.
- [155] Chi Chung Ko, B. M. Chen, Shaoyan Hu, V. Ramakrishnan, Chang Dong Cheng, Yuan Zhuang, and Jianping Chen, "A web-based virtual laboratory on a frequency modulation experiment," *IEEE Trans. Syst. Man Cybern. Part C (Applications Rev.)*, vol. 31, no. 3, pp. 295–303, 2001.
- [156] M. M. Albu, K. E. Holbert, G. T. Heydt, S. D. Grigorescu, and V. Trusca, "Embedding Remote Experimentation in Power Engineering Education," *IEEE Trans. Power Syst.*, vol. 19, no. 1, pp. 139–143, Feb. 2004.
- [157] M. T. Restivo, J. Mendes, A. M. Lopes, C. M. Silva, and F. Chouzal, "A Remote Laboratory in Engineering Measurement," *IEEE Trans. Ind. Electron.*, vol. 56, no. 12, pp. 4836–4843, Dec. 2009.
- [158] A. Valera, J. L. Diez, M. Valles, and P. Albertos, "Virtual and remote control laboratory development," *IEEE Control Syst. Mag.*, vol. 25, no. 1, pp. 35–39, Feb. 2005.
- [159] R. Puerto, L. M. Jiménez, and O. Reinoso, "Remote control laboratory via Internet using Matlab and Simulink," *Comput. Appl. Eng. Educ.*, vol. 18, no. 4, pp. 694–702, Dec. 2010.
- [160] J. Sánchez, S. Dormido, R. Pastor, and F. Morilla, "A Java / Matlab-Based Environment for Remote Control System Laboratories : Illustrated With an Inverted Pendulum," *IEEE Trans. Educ.*, vol. 47, no. 3, pp. 321–329, 2004.
- [161] G. Farias, R. De Keyser, S. Dormido, and F. Esquembre, "Developing Networked Control Labs: A Matlab and Easy Java Simulations Approach," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3266–3275, Oct. 2010.
- [162] L. Jiménez, R. Puerto, and O. Reinoso, "Recolab: laboratorio remoto de

- control utilizando Matlab y Simulink,” *Rev. Iberoam. Automática e Informática Ind.*, vol. 2, pp. 64–72, 2010.
- [163] A. Valera, M. Vallés, and J. L. Díez, “SIMULACIÓN Y CONTROL DE PROCESOS FÍSICOS DE FORMA REMOTA,” *Rev. Iberoam. Automática e Informática Ind. RIAI*, vol. 2, no. 2, pp. 20–29, 2005.
- [164] M. Casini, D. Prattichizzo, and A. Vicino, “The automatic control telelab: a user-friendly interface for distance learning,” *IEEE Trans. Educ.*, vol. 46, no. 2, pp. 252–257, May 2003.
- [165] C. Schmid, “Remote experimentation techniques for teaching control engineering,” *Fourth Int. Sci. ...*, no. June, 2000.
- [166] J. Sanchez, F. Morilla, S. Dormido, J. Aranda, and P. Ruiperez, “Virtual and remote control labs using Java: a qualitative approach,” *IEEE Control Syst. Mag.*, vol. 22, no. 2, pp. 8–20, Apr. 2002.
- [167] F. J. González-Castaño, L. Anido-Rifón, J. Vales-Alonso, M. J. Fernández-Iglesias, M. Llamas Nistal, P. Rodríguez-Hernández, and J. M. Pousada-Carballo, “Internet access to real equipment at computer architecture laboratories using the Java/CORBA paradigm,” *Comput. Educ.*, vol. 36, no. 2, pp. 151–170, 2001.
- [168] J. Fernandez, R. Bragós, M. Cabrera, A. Abelló, N. Arroyo, D. González, F. Garófano, A. Cortés, and A. Fabra, “Interoperability platform for virtual and remote laboratories,” *2012 9th Int. Conf. Remote Eng. Virtual Instrumentation, REV 2012*, 2012.
- [169] G. Viedma, I. J. Dancy, and K. H. Lundberg, “A web-based linear-systems iLab,” in *Proceedings of the 2005, American Control Conference, 2005.*, 2005, pp. 5139–5144.
- [170] C. a. Jara, F. a. Candelas, J. Pomares, and F. Torres, “Java software platform for the development of advanced robotic virtual laboratories,” *Comput. Appl. Eng. Educ.*, p. n/a-n/a, Mar. 2011.
- [171] G. Farias and F. Esquembre, “Desarrollo de laboratorios virtuales, interactivos y remotos utilizando easy java simulations y modelos simulink,” *12th Latin-American Congr. Autom. Control*, pp. 336–341, 2006.
- [172] E. Fabregas, N. Duro, R. Dormido, S. Dormido-Canto, H. Vargas-Oyarzún, and S. Dormido, “Virtual and remote experimentation with the Ball and Hoop system,” in *2009 IEEE Conference on Emerging Technologies & Factory Automation*, 2009, pp. 1–8.

- [173] N. Carrero and R. Costa-castell, "Using interactive tools to teach / learn Sliding Mode Control," *49th IEEE Conf. Decis. Control*, vol. 0, no. x, pp. 5132–5137, Dec. 2010.
- [174] C. a. Jara, F. a. Candelas, F. Torres, S. Dormido, F. Esquembre, and O. Reinoso, "Real-time collaboration of virtual laboratories through the Internet," *Comput. Educ.*, vol. 52, no. 1, pp. 126–140, Jan. 2009.
- [175] S. Dormido, H. Vargas-Oyarzún, J. Sanchez, N. Duro, R. Dormido, S. Dormido-Canto, and F. Esquembre, "Using Web-based laboratories for control engineering education," in *International Conference on Engineering Education – ICEE 2007*, 2007, pp. 5–10.
- [176] L. de la Torre, J. P. Sanchez, R. Heradio, C. Carreras, M. Yuste, J. Sanchez, and S. Dormido, "UNEDLABS: An Example of EJS Labs Integration into Moodle," in *World Conference on Physics Education*, 2012, no. July, pp. 537–546.
- [177] H. Vargas-Oyarzún, J. Sánchez-Moreno, S. Dormido, C. Salzmann, D. Gillet, and F. Esquembre, "Web-Enabled Remote Scientific Environments," *Comput. Sci. Eng.*, vol. 11, no. 3, pp. 36–46, May 2009.
- [178] X. Chen, G. Song, and Y. Zhang, "Virtual and Remote Laboratory Development : A Review," in *Earth and Space 2010: Engineering; Science; and Operations in Challenging Environments*, 2010, pp. 3843–3852.
- [179] F. Luthon and B. Larroque, "LaboREM-A Remote Laboratory for Game-Like Training in Electronics," *IEEE Trans. Learn. Technol.*, vol. 8, no. 3, pp. 311–321, Jul. 2015.
- [180] M. a. Bochicchio and A. Longo, "Hands-On Remote Labs: Collaborative Web Laboratories as a Case Study for IT Engineering Classes," *IEEE Trans. Learn. Technol.*, vol. 2, no. 4, pp. 320–330, Oct. 2009.
- [181] M. Kaluz, J. García-Zubía, M. Fikar, and L. Cirka, "A Flexible and Configurable Architecture for Automatic Control Remote Laboratories," *IEEE Trans. Learn. Technol.*, vol. 8, no. 3, pp. 299–310, Jul. 2015.
- [182] M. Tawfik, C. Salzmann, D. Gillet, D. Lowe, H. Saliyah-Hassane, E. Sancristobal Ruiz, and M. Castro, "Laboratory as a service (LaaS): A novel paradigm for developing and implementing modular remote laboratories," *Int. J. Online Eng.*, vol. 10, no. 4, pp. 13–21, 2014.
- [183] V. J. Harward, J. A. del Alamo, S. R. Lerman, P. H. Bailey, J. Carpenter, K. DeLong, C. Felknor, J. Hardison, B. Harrison, I. Jabbour, P. D. Long, L. Naamani, J. Northridge, M. Schulz, D. Talavera, C. Varadharajan, K. Yehia,

- R. Zbib, and D. Zych, "The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories," *Proc. IEEE*, vol. 96, no. 6, pp. 931–950, Jun. 2008.
- [184] J. Saenz, F. Esquembre, F. J. Garcia, L. de la Torre, and S. Dormido, "A new model for a remote connection with hardware devices using Javascript," in *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2016, pp. 143–144.
- [185] S. Colbran and M. Schulz, "An update to the software architecture of the iLab Service Broker," in *Proceedings of 2015 12th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2015, no. February, pp. 90–93.
- [186] A. Maiti, A. D. Maxwell, and A. A. Kist, "An overview of system architectures for Remote Laboratories," in *Proceedings of 2013 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, 2013, pp. 661–666.
- [187] A. R. B. R. M. von L. J. M. A. Polze, "Real-time robotics and process control experiments in the Distributed Control Lab," *IEE Proc. - Softw.*, vol. 152, no. 5, pp. 229–235, 2005.
- [188] A. Rasche, P. Tr, M. Dirska, and A. Polze, "Foucault ' s Pendulum in the Distributed Control Lab," in *Ninth IEEE International Workshop on*, 2003, p. 299.
- [189] T. de Jong, D. Gillet, and S. Sotiriou, "Web de Go-Lab." [Online]. Available: <http://www.golabz.eu/>. [Accessed: 15-Sep-2016].
- [190] M. Gourmaj, A. Naddami, A. Fahli, and M. Moussetad, "Integration of virtual instrument systems in reality (VISIR) OpenLabs with Khouribga OnlineLab," in *2015 International Conference on Interactive Collaborative Learning (ICL)*, 2015, no. September, pp. 793–797.
- [191] K. P. Ayodele, I. A. Inyang, and L. O. Kehinde, "An iLab for Teaching Advanced Logic Concepts With Hardware Descriptive Languages," *IEEE Trans. Educ.*, vol. 58, no. 4, pp. 262–268, 2015.
- [192] D. G. Zutin and M. Auer, "A Simple LabVIEW based Framework to Facilitate the deployment of iLab Batch Lab Servers," in *Remote Engineering and Virtual Instrumentation (REV), 2014 11th International Conference on*, 2014, no. February.
- [193] A. Bagnasco, M. Chirico, and A. M. Scapolla, "XML Technologies to Design Didactical Distributed Measurement Laboratories," in *Instrumentation*

- and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE, 2002, no. 1, pp. 21–23.*
- [194] D. Ponta, A. M. Scapolla, and P. Buschiazzo, "Survey of Remote Laboratories Using Service Oriented Architectures," *Int. J. Online Eng.*, vol. 5, no. 1, p. 6 pp.-pp., Feb. 2009.
- [195] M. Tawfik, E. S. Cristóbal, A. Pesquera, R. Gil, S. Martin, G. Diaz, J. Peire, M. Castro, R. Pastor, S. Ros, and R. Hernandez, "Shareable educational architectures for remote laboratories," in *Proceedings - 2012 Technologies Applied to Electronics Teaching, TAE 2012, 2012*, pp. 122–127.
- [196] B. Pradarelli, L. Latorre, M. L. Flottes, Y. Bertrand, and P. Nouet, "Remote labs for industrial IC testing," *IEEE Trans. Learn. Technol.*, vol. 2, no. 4, pp. 304–311, Oct. 2009.
- [197] D. B. Lowe, S. Murray, L. Weber, M. De La Villefromoy, A. Johnston, E. Lindsay, W. Nageswaran, and A. Nafalski, "LabShare: Towards a national approach to laboratory sharing," *20th Annu. Conf. Australas. Assoc. Eng. Educ.*, no. September 2016, pp. 458–463, 2009.
- [198] L. Bellido, V. Villagra, and V. Mateos, "Federated authentication and authorization for reusable learning objects," in *IEEE EDUCON 2010 Conference, 2010*, pp. 1071–1074.
- [199] T. Richter, P. Grube, and D. Boehringer, "Integrating an online programming lab into ILIAS," in *2014 11th International Conference on Remote Engineering and Virtual Instrumentation (REV), 2014*, vol. 1, no. February, pp. 31–34.
- [200] Z. Nedic and J. Machotka, "Remote Laboratory NetLab for Effective Teaching of 1 st Year Engineering Students," *Int. J. Online Eng.*, vol. 3, no. 3, pp. 1–6, 2007.
- [201] M. Teng, H. Considine, Z. Nedic, and A. Nafalski, "Current and Future Developments in the Remote Laboratory NetLab," *Int. J. Online Eng.*, vol. 12, no. 8, pp. 4–12, 2016.
- [202] A. Maiti, "NETLab: An online laboratory management system," in *IEEE EDUCON 2010 Conference, 2010*, pp. 1351–1358.
- [203] C. Gravier, "Vers la généralisation de manipulations distantes et collaboratives d'instruments de haute technologie," Université Jean Monnet - Saint-Etienne, 2007.

-
- [204] Telecom St. Etienne, "Ocelot: Opensource software." [Online]. Available: <http://ocelot.telecom-st-etienne.fr/>. [Accessed: 21-Oct-2016].
- [205] S. Dormido, J. Sánchez, L. De la Torre, R. Heradio, C. Carreras, J. P. Sánchez, and M. Yuste, "Physics Experiments at the UNEDLabs Portal," *Int. J. Online Eng.*, vol. 8, no. S1, pp. 26–27, Jan. 2012.
- [206] J. Saenz, J. Chacon, L. De La Torre, A. Visioli, and S. Dormido, "Open and Low-Cost Virtual and Remote Labs on Control Engineering," *IEEE Access*, vol. 3, no. 99, pp. 805–814, 2015.
- [207] R. Heradio, L. De La Torre, J. Sanchez, and S. Dormido, "Making EJS applications at the OSP digital library available from Moodle," *Proc. 2014 11th Int. Conf. Remote Eng. Virtual Instrumentation, REV 2014*, no. February, pp. 112–116, 2014.
- [208] M. A. Marques, M. C. Viegas, M. C. Costa-Lobo, A. V. Fidalgo, G. R. Alves, J. S. Rocha, and I. Gustavsson, "How remote labs impact on course outcomes: Various practices using VISIR," *IEEE Trans. Educ.*, vol. 57, no. 3, pp. 151–159, 2014.
- [209] I. Gustavsson, K. Nilsson, J. Zackrisson, G. R. Alves, A. V. Fidalgo, L. Claesson, J. G. Zubia, U. H. Jayo, M. Castro, G. Diaz Orueta, and F. G. Loro, "Lab sessions in VISIR laboratories," in *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2016, pp. 350–352.
- [210] L. Rodriguez-Gil, P. Orduna, J. Garcia-Zubia, and D. Lopez-de-Ipina, "Advanced integration of OpenLabs VISIR (Virtual Instrument Systems in Reality) with Weblab-Deusto," in *2012 9th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2012, pp. 1–7.
- [211] J. García-Zubía and P. Orduña, "Integración del laboratorio remoto WebLab-Deusto en Moodle," *WebLab-Deusto ...*, 2009.
- [212] J. Garcia-Zubia, P. Orduna, I. Angulo, U. Hernandez, O. Dziabenko, D. Lopez-Ipina, and L. Rodriguez-Gil, "Application and user perceptions of using the WebLab-Deusto-PLD in technical education," in *2011 First Global Online Laboratory Consortium Remote Laboratories Workshop*, 2011, pp. 1–6.
- [213] M. Ogot, G. Elliott, and N. Glumac, "An Assessment of In-Person and Remotely Operated Laboratories," *J. Eng. Educ.*, vol. 92, no. 1, pp. 57–64, Jan. 2003.
- [214] T. F. Wiesner and W. Lan, "Comparison of Student Learning in Physical

- and Simulated Unit Operations Experiments," *J. Eng. Educ.*, vol. 93, no. 3, pp. 195–204, Jul. 2004.
- [215] L. D. Feisel, G. D. Peterson, O. Arnas, L. Carter, A. Rosa, and W. Worek, "Learning objectives for engineering education laboratories," *Front. Educ. 2002 FIE 2002 32nd Annu. Conf.*, vol. 2, p. F1D–1, 2002.
- [216] C. S. Tzafestas, N. Palaiologou, and M. Alifragis, "Virtual and Remote Robotic Laboratory: Comparative Experimental Evaluation," *IEEE Trans. Educ.*, vol. 49, no. 3, pp. 360–369, Aug. 2006.
- [217] Z. C. Zacharia, "Comparing and combining real and virtual experimentation: an effort to enhance students' conceptual understanding of electric circuits," *J. Comput. Assist. Learn.*, vol. 23, no. 2, pp. 120–132, Jan. 2007.
- [218] M. Abdulwahed and Z. K. Nagy, "Applying Kolb's Experiential Learning Cycle for Laboratory Education," *J. Eng. Educ.*, vol. 98, no. 3, pp. 283–294, Jul. 2009.
- [219] M. Abdulwahed and Z. K. Nagy, "The impact of different preparation modes on enhancing the undergraduate process control engineering laboratory: A comparative study," *Comput. Appl. Eng. Educ.*, vol. 22, no. 1, pp. 110–119, 2014.
- [220] M. Stefanovic, "The objectives, architectures and effects of distance learning laboratories for industrial engineering education," *Comput. Educ.*, vol. 69, pp. 250–262, 2013.
- [221] F. Arango, Chenghung Chang, S. K. Esche, and C. Chassapis, "A scenario for collaborative learning in virtual engineering laboratories," in *2007 37th annual frontiers in education conference - global engineering: knowledge without borders, opportunities without passports*, 2007, p. F3G–7–F3G–12.
- [222] E. Sancristobal Ruiz, S. Martin, R. Gil, P. Orduna, M. Tawfik, A. Pesquera, G. Diaz, A. Colmenar, J. García-Zubía, and M. Castro, "State of Art, Initiatives and New Challenges for Virtual and Remote Labs," in *2012 IEEE 12th International Conference on Advanced Learning Technologies*, 2012, pp. 714–715.
- [223] M. Zappatore, A. Longo, and M. A. Bochicchio, "The bibliographic reference collection GRC2014 for the Online Laboratory Research community," in *Proceedings of 2015 12th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2015, no.

February, pp. 24–31.

- [224] J. P. N. Firmeza and F. M. S. Ramos, “TeleProd: An Internet based Teleproductics system,” in *Tenth International IFIP WG5.2/5.3 International Conference PROLAMAT 98*, 1998, pp. 81–92.
- [225] H. Hoyer, A. Jochheim, C. Rohrig, and A. Bischoff, “A Multiuser Virtual-Reality Environment for a Tele-Operated Laboratory,” *IEEE Trans. Educ.*, vol. 47, no. 1, pp. 121–126, Feb. 2004.
- [226] T. D. Murphey, “Teaching Rigid Body Mechanics Using Student-Created Virtual Environments,” *IEEE Trans. Educ.*, vol. 51, no. 1, pp. 45–52, 2008.
- [227] D. R. A. Rambli, M. Y. Nayan, and S. Sulaiman, “A Portable Augmented Reality Lab,” *1st Int. Malaysian Educ. Technol. Conv.*, vol. 1, pp. 176–183, 2007.
- [228] M. A. Vouk, D. L. Bitzer, and R. L. Klevans, “Workflow and end-user quality of service issues in Web-based education,” *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 4, pp. 673–687, 1999.
- [229] F. Arango, Chenghung Chang, S. K. Esche, and C. Chassapis, “A scenario for collaborative learning in virtual engineering laboratories,” in *2007 37th annual frontiers in education conference - global engineering: knowledge without borders, opportunities without passports*, 2007, p. F3G–7–F3G–12.
- [230] W. R. van Joolingen, T. de Jong, A. W. Lazonder, E. R. Savelsbergh, and S. Manlove, “Co-Lab: research and development of an online learning environment for collaborative scientific discovery learning,” *Comput. Human Behav.*, vol. 21, no. 4, pp. 671–688, Jul. 2005.
- [231] L. M. Regueras, E. Verdu, M. J. Verdu, and J. P. de Castro, “Design of a Competitive and Collaborative Learning Strategy in a Communication Networks Course,” *IEEE Trans. Educ.*, vol. 54, no. 2, pp. 302–307, May 2011.
- [232] D. Gillet, A. Ngoc, and Y. Rekik, “Collaborative web-based experimentation in flexible engineering education,” *Educ. IEEE Trans.*, vol. 48, no. 4, pp. 696–704, Nov. 2005.
- [233] B. Barros, T. Read, and M. F. Verdejo, “Virtual Collaborative Experimentation: An Approach Combining Remote and Local Labs,” *IEEE Trans. Educ.*, vol. 51, no. 2, pp. 242–250, 2008.
- [234] C. a. Jara, F. a. Candelas, F. Torres, S. Dormido, and F. Esquembre,

- "Synchronous collaboration of virtual and remote laboratories," *Comput. Appl. Eng. Educ.*, vol. 20, no. 1, pp. 124–136, Mar. 2012.
- [235] L. de la Torre, R. Heradio, C. A. Jara, J. Sanchez, S. Dormido, F. Torres, and F. A. Candelas, "Providing collaborative support to virtual and remote laboratories," *IEEE Trans. Learn. Technol.*, vol. 6, no. 4, pp. 312–323, Oct. 2013.
- [236] J. Garcia-Zubia, J. Irurzun, I. Angulo, U. Hernandez, M. Castro, E. Sancristobal, P. Orduna, and J. Ruiz-de-Garibay, "SecondLab: A remote laboratory under Second Life," in *IEEE EDUCON 2010 Conference*, 2010, pp. 351–356.
- [237] J. Fayolle, C. Gravier, and B. Jailly, "Collaborative Remote Laboratory in Virtual World," *Proc. 10th WSEAS Int. Conf. Appl. Informatics Commun.*, pp. 396–402, 2010.
- [238] L. de la Torre, M. Guinaldo, R. Heradio, and S. Dormido, "The Ball and Beam System: A Case Study of Virtual and Remote Lab Enhancement With Moodle," *IEEE Trans. Ind. Informatics*, vol. 11, no. 4, pp. 934–945, Aug. 2015.
- [239] J. Sánchez Moreno, "Un nuevo enfoque metodológico para la enseñanza a distancia de asignaturas experimentales. Análisis, diseño y desarrollo de un laboratorio virtual y remoto para el estudio de la automática a través de Internet," Universidad Nacional de Educación a Distancia, UNED, 2001.
- [240] R. Pastor, "Especificación formal de laboratorios virtuales y remotos. Aplicación a la ingeniería de control," Universidad Nacional de Educación a Distancia, UNED, 2006.
- [241] C. Martin-Villalba, "Modelado Orientado a Objetos de Laboratorios Virtuales para la Educación en Control Automático," Universidad Nacional de Educación a Distancia, 2007.
- [242] H. Vargas-Oyarzún, "Entorno Integral Basado en Web para la Educación en Ingeniería de Control," Universidad Nacional de Educación a Distancia, UNED, 2010.
- [243] M. S. T. Abuelela, "Laboratory as a Service (LaaS): a Paradigm for Developing and Implementing Modular Remote Laboratories," Universidad Nacional de Educación a Distancia (UNED), 2013.
- [244] E. Sancristobal Ruiz, "METODOLOGÍA, ESTRUCTURA Y DESARROLLO DE INTERFACES INTERMEDIAS PARA LA CONEXIÓN DE LABORATORIOS

- REMOTOS Y VIRTUALES A PLATAFORMAS EDUCATIVAS,” Universidad Nacional de Educación a Distancia (UNED), 2012.
- [245] E. Fabregas, “PLATAFORMAS INTERACTIVAS DE EXPERIMENTACIÓN VIRTUAL Y REMOTA: APLICACIONES DE CONTROL Y ROBÓTICA,” Universidad Nacional de Educación a Distancia (UNED), 2013.
- [246] H. Vargas-Oyarzún, “An Integral Web-based Environment for Control Engineering Education,” Universidad Nacional de Educación a Distancia (UNED), 2010.
- [247] S. Dormido, J. Sánchez–Moreno, H. Vargas, L. De La Torre, and R. Heradio, “UNED Labs: a network of virtual and remote laboratories,” in *Using remote labs in education: two little ducks in remote experimentation*, J. García-Zubía and G. R. Alves, Eds. Bilbao: Deusto Digital, 2011, pp. 253–270.
- [248] H. Vargas-Oyarzún, J. Sánchez, C. A. Jara, F. A. Candelas, F. Torres, and S. Dormido, “A Network of Automatic Control Web-Based Laboratories,” *IEEE Trans. Learn. Technol.*, vol. 4, no. 3, pp. 197–208, 2011.
- [249] L. de la Torre, J. Sánchez, S. Dormido, J. P. Sánchez, M. Yuste, and C. Carreras, “Two web-based laboratories of the FisL@bs network: Hooke’s and Snell’s laws,” *Eur. J. Phys.*, vol. 32, no. 2, pp. 571–584, 2011.
- [250] S. Rapuano and F. Zoino, “A Learning Management System Including Laboratory Experiments on Measurement Instrumentation,” *IEEE Trans. Instrum. Meas.*, vol. 55, no. 5, pp. 1757–1766, Oct. 2006.
- [251] G. Andria, A. Baccigalupi, M. Borsic, P. Carbone, P. Daponte, C. De Capua, A. Ferrero, D. Grimaldi, A. Liccardo, N. Locci, A. M. L. Lanzolla, D. Macii, C. Muscas, L. Peretto, D. Petri, S. Rapuano, M. Riccio, S. Salicone, and F. Stefani, “Remote Didactic Laboratory ‘G. Savastano,’ The Italian Experience for E-Learning at the Technical Universities in the Field of Electrical and Electronic Measurement: Architecture and Optimization of the Communication Performa,” *IEEE Trans. Instrum. Meas.*, vol. 56, no. 4, pp. 1135–1147, Apr. 2007.
- [252] E. Sancristobal Ruiz, S. Martin, R. Gil, G. Díaz, A. Colmenar, M. Castro, J. Peire, J. M. Gómez, E. López, and P. López, “Integration of internet based labs and open source LMS,” *Proc. - 3rd Int. Conf. Internet Web Appl. Serv. ICIW 2008*, no. Figure 2, pp. 217–222, 2008.
- [253] D. Lowe, S. Murray, and E. Lindsay, “Evolving Remote Laboratory Architectures to Leverage Emerging Internet Technologies,” *IEEE Trans.*

- Learn. Technol.*, vol. 2, no. 4, pp. 289–294, Oct. 2009.
- [254] D. Cmur, T. Mutapcic, and I. Bilic, “MIRACLE—Model for Integration of Remote Laboratories in Courses that Use Laboratory and e-Learning Systems,” *IEEE Trans. Learn. Technol.*, vol. 2, no. 4, pp. 275–288, Oct. 2009.
- [255] N. Abdellaoui, C. Gravier, B. Belmekki, and J. Fayolle, “Towards the loose coupling between LMS and Remote Laboratories in online engineering education,” in *IEEE EDUCON 2010 Conference*, 2010, pp. 1935–1940.
- [256] M. a. Bochicchio and A. Longo, “Delivering Collaborative Web Labs as a Service for Engineering Education,” *Int. J. Online Eng.*, vol. 8, no. 2, pp. 4–10, May 2012.
- [257] A. Pesquera, R. Morales, R. Pastor, S. Ros, R. Hernandez, E. Sancristobal Ruiz, and M. Castro, “dotLAB: Integrating remote labs in dotLRN,” in *2011 IEEE Global Engineering Education Conference (EDUCON)*, 2011, pp. 111–117.
- [258] M. Guinaldo, L. De La Torre, R. Heradio, and S. Dormido, “A virtual and remote control laboratory in moodle: The ball and beam system,” *IFAC Proc. Vol.*, vol. 10, no. PART 1, pp. 72–77, 2013.
- [259] P. Orduna, E. Sancristobal Ruiz, M. Emaldi, M. Castro, D. Lopez-De-Ipina, and J. García-Zubía, “Modelling remote laboratories integrations in e-learning tools through remote laboratories federation protocols,” *Proc. - Front. Educ. Conf. FIE*, pp. 1–6, 2012.
- [260] L. Rodriguez-Gil, P. Orduna, J. Garcia-Zubia, and D. Lopez-de-Ipina, “Advanced integration of OpenLabs VISIR (Virtual Instrument Systems in Reality) with Weblab-Deusto,” in *2012 9th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2012, pp. 1–7.
- [261] F. Lerro, P. Orduña, and S. Marchisio, “Development of a remote laboratory management system and integration with social networks,” *Int. J. Recent Contrib. from Eng. Sci. IT*, pp. 1–5, 2013.
- [262] F. Lerro, S. Marchisio, S. Martini, H. Massacessi, E. Perretta, A. Gimenez, N. Aimetti, and J. I. Oshiro, “Performing Real Experiments From a Remote Learning Management System,” *IEEE Rev. Iberoam. Tecnol. del Aprendiz.*, vol. 9, no. 1, pp. 23–27, Feb. 2014.
- [263] N. Kafadarova, N. Mileva, and S. Stoyanova, “Remote Wireless Communications lab in real time,” *IEEE Glob. Eng. Educ. Conf. EDUCON*, pp. 69–74, 2013.

-
- [264] H. Wuttke, M. Hamann, and K. Henke, "Integration of remote and virtual laboratories in the educational process," in *Proceedings of 2015 12th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2015, vol. 11, no. February, pp. 157–162.
- [265] M. Ozvoldova and P. Ondrušek, "Integration of Online Labs into Educational Systems," *Int. J. Online Eng.*, vol. 11, no. 6, p. 54, Nov. 2015.
- [266] Advanced Distributed Learning (ADL), "SCORM Adopters List," Advanced Distributed Learning Initiative (ADL), Report, 2016.
- [267] B. Ozdogru and N. E. Cagiltay, "How Content Management Problem of a Remote Laboratory System can be handled by integrating an open source learning management system? Problems and solutions," in *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2007, pp. 1–5.
- [268] Y. Tetour, D. Boehringer, and T. Richter, "Integration of virtual and remote experiments into undergraduate engineering courses," in *2011 First Global Online Laboratory Consortium Remote Laboratories Workshop*, 2011, pp. 1–6.
- [269] L. Bellido, V. Mateos, V. A. Villagr a, D. Fern andez, and O. Walid, "Remote access to computer networking laboratories," *2012 9th Int. Conf. Remote Eng. Virtual Instrumentation, REV 2012*, 2012.
- [270] E. Barra, A. Gordillo, D. Gallego, and J. Quemada, "Integration of SCORM packages into web games," *Proc. - Front. Educ. Conf. FIE*, pp. 685–690, 2013.
- [271] P. Ordu a, S. B. Uribe, N. H. Isaza, E. Sancristobal Ruiz, M. Emaldi, A. P. Martin, K. DeLong, P. Bailey, D. L pez-De-Ipi a, M. Castro, and J. Garc a-Zub a, "Generic integration of remote laboratories in learning and content management systems through federation protocols," *Proc. - Front. Educ. Conf. FIE*, pp. 1372–1378, 2013.
- [272] T. de Jong, D. Gillet, and S. Sotiriou, "Web de Go-Lab."
- [273] "Web de documentaci n de Gateway4labs." [Online]. Available: <https://gateway4labs.readthedocs.io>. [Accessed: 15-Sep-2015].
- [274] P. Orduna, A. Caminero, I. Lequerica, D. G. Zutin, P. Bailey, E. Sancristobal Ruiz, L. Rodriguez-Gil, A. Robles-Gomez, M. Latorre, K. DeLong, L. Tobarra, S. Ros, M. Castro, D. Lopez-de-Ipina, and J. Garcia-Zubia, "Generic integration of remote laboratories in public learning tools: Organizational and technical challenges," in *2014 IEEE Frontiers in*

Education Conference (FIE) Proceedings, 2014, pp. 1–7.

- [275] P. Orduna, D. G. Zutin, S. Govaerts, I. L. Zorrozuza, P. H. Bailey, E. Sancristobal Ruiz, C. Salzmann, L. Rodriguez-Gil, K. DeLong, D. Gillet, M. Castro, D. Lopez-De-Ipina, and J. García-Zubía, “An Extensible Architecture for the Integration of Remote and Virtual Laboratories in Public Learning Tools,” *Rev. Iberoam. Tecnol. del Aprendiz.*, vol. 10, no. 4, pp. 223–233, 2015.
- [276] Real Academia Española, *Diccionario de la Lengua Espanola*, 23rd ed. 2014.
- [277] Y. Vovides, S. Sanchez-Alonso, V. Mitropoulou, and G. Nickmans, “The use of e-learning course management systems to support learning strategies and to improve self-regulated learning,” *Educ. Res. Rev.*, vol. 2, no. 1, pp. 64–74, 2007.
- [278] M. A. Trenas, J. Ramos, E. D. Gutiérrez, S. Romero, and F. Corbera, “Use of a new moodle module for improving the teaching of a basic course on computer architecture,” *IEEE Trans. Educ.*, vol. 54, no. 2, pp. 222–228, May 2011.
- [279] M. Tawfik, C. Salzmann, D. Gillet, D. Lowe, H. Saliyah-Hassane, E. Sancristobal Ruiz, and M. Castro, “Laboratory as a Service (LaaS): A model for developing and implementing remote laboratories as modular components,” in *2014 11th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, 2014, vol. 10, no. 4, pp. 11–20.
- [280] B. Evans, *Java: The Legend*, vol. 53. O’Reilly Media, 2015.
- [281] Dalibor, “Moving to a Plugin-Free Web,” *Topic-Oracle*, 2016. [Online]. Available: https://blogs.oracle.com/java-platform-group/entry/moving_to_a_plugin_free. [Accessed: 20-Jul-2016].
- [282] M. Byrne, “The Rise and Fall of the Java Applet: Creative Coding’s Awkward Little Square,” *MotherBoard*, 2016. [Online]. Available: <http://motherboard.vice.com/read/a-brief-history-of-the-java-applet>. [Accessed: 20-Jul-2016].
- [283] R. Harmes and D. Diaz, *Pro JavaScript design Patterns*. Apress, 2008.
- [284] Oracle, “Invoking JavaScript Code From an Applet,” *Java Documentation*. [Online]. Available: <https://docs.oracle.com/javase/tutorial/deployment/applet/invokingJavaScriptFromApplet.html>. [Accessed: 19-Sep-2016].

-
- [285] L. García-Cabrera, J. M. Ortega-Tudela, J. R. Balsas-Almagro, I. Ruano-Ruano, M. Á. Peña-Hita, and J. C. Cuevas-Martínez, "New Assessment Tools in Learning Management Systems," in *The future of Education International Conference 2nd Ed. (2012)*, 2012, pp. 182–187.
- [286] I. Ruano-ruano and L. García-cabrera, "Estudio y parametrización de pruebas de evaluación para su implementación en docencia virtual," pp. 1–10.
- [287] D. Muñoz de la Peña, F. Gómez-Estern, and S. Dormido, "A new Internet tool for automatic evaluation in Control Systems and Programming," *Comput. Educ.*, vol. 59, no. 2, pp. 535–550, Sep. 2012.
- [288] M. López-Martínez, F. Gómez-Estern, and D. Muñoz, "Sistema de Evaluación Automática Vía Web en Asignaturas Prácticas de Ingeniería," *Rev. Iberoam. automática e informática Ind.*, vol. 7, no. 3, pp. 111–119, Jul. 2010.
- [289] A. Muñoz de la Peña, D. González-Gómez, D. Muñoz de la Peña, and F. Gómez-Estern, "Evaluación automática de ejercicios experimentales de análisis químico instrumental," in *Jornadas de Automática*, 2011.
- [290] IMS Global Learning Consortium (IMS), "IMS Question & Test Interoperability® Specification." [Online]. Available: <https://www.imsglobal.org/question/index.html>. [Accessed: 24-Sep-2016].
- [291] S. Díaz Fuentes, "Creación de Test interactivos SCORM." Universidad de Jaén, Jaén, p. 200, 2016.
- [292] Advanced Distributed Learning (ADL), "SCORM Certified List," Advanced Distributed Learning Initiative (ADL), Report, 2016.
- [293] F. Yang, F. W. B. Li, and R. W. H. Lau, "A Fine-Grained Outcome-Based Learning Path Model," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 44, no. 2, pp. 235–245, Feb. 2014.
- [294] J. Garratt, "Virtual Investigations: ways to accelerate experience," *Univ. Chem. Educ.*, vol. 1, no. 1, pp. 19–27, 1997.
- [295] M. Limniou, N. Papadopoulos, and C. Whitehead, "Integration of simulation into pre-laboratory chemical course: Computer cluster versus WebCT," *Comput. Educ.*, vol. 52, no. 1, pp. 45–52, 2009.
- [296] J. Morkes and J. Nielsen, "Concise, scannable, and objective: How to write for the Web," *Useit. com*, pp. 1–17, 1997.

- [297] U. de Jaén, "Web de la EPS de Jaén." [Online]. Available: <http://eps.ujaen.es/>. [Accessed: 30-Sep-2016].
- [298] U. de Jaén, "Web de la EPS de Linares." [Online]. Available: <http://www10.ujaen.es/conocenos/centros/eps/>. [Accessed: 30-Sep-2016].
- [299] F. Instrument, "Web de FeedBack-Instrument." [Online]. Available: <http://www.feedback-instruments.com/>. [Accessed: 30-Sep-2016].
- [300] J. E. Corter, S. K. Esche, C. Chassapis, J. Ma, and J. V. Nickerson, "Process and learning outcomes from remotely-operated, simulated, and hands-on student laboratories," *Comput. Educ.*, vol. 57, no. 3, pp. 2054–2067, Nov. 2011.
- [301] M. Rashid, I. A. Tasadduq, Y. I. Zia, M. Al-Turkistany, and S. Rashid, "Evaluation of engineering laboratories," *2012 Int. Conf. Educ. e-Learning Innov. ICEELI 2012*, 2012.
- [302] Y. Amigud, G. Archer, J. Smith, M. Szymanski, and B. Servatius, "Assessing the quality of web-enabled laboratories in undergraduate education," *32nd Annu. Front. Educ.*, vol. 2, pp. 12–16, 2002.
- [303] K. Ogata, *Ingeniería de control moderna*, 5^a. Madrid (Spain): Prentice Hall, 2010.
- [304] F. A. Candelas, F. Torres, P. Gil, F. Ortiz, S. Puente, and J. Pomares, "Laboratorio Virtual Remoto para Robótica y Evaluación de su Impacto en la Docencia," *Rev. Iberoam. Automática e Informática Ind.*, vol. 1, no. 2, pp. 49–57, 2004.
- [305] M. A. Bonache Gómez, "Diseño de un middleware de comunicaciones para la gestión remota de equipos docentes de control de procesos." Universidad de Jaén, p. 176, 2014.
- [306] I. Malpica Herrera, "WebLab SIP-Java basado en SCORM." Universidad de Jaén, p. 118, 2015.
- [307] F. Puentes Marchal, "WebLab SCORM de clientes SIP." Universidad de Jaén, Linares, p. 84, 2016.
- [308] J. F. Vico Zafra, "Desarrollo de laboratorio web (WebLab) basado en SCORM con capacidad de interactuar con LMS." Universidad de Jaén, Jaén, p. 242, 2015.
- [309] A. Vela Ruíz, "Desarrollo de laboratorio web (WebLab) basado en SCORM

- con capacidad de interaccionar con LMS.” Universidad de Jaén, Jaén, p. 235, 2015.
- [310] L. García-Cabrera and I. Ruano-Ruano, “Algunas claves para garantizar el éxito en la teleformación: «Sistemas Hipermedia»,” in *Buenas Prácticas en teleformación en las diez universidades andaluzas*, 1st ed., J. I. Aguaded and A. Infante, Eds. La Coruña (Spain): Junta de Andalucía, 2009.
- [311] L. García-Cabrera, I. Ruano-Ruano, J. M. Ortega-Tudela, M. Á. Peña-Hita, and J. R. Balsas-Almagro, “Aplicación y resultados del procedimiento de evaluación y mejora de la calidad en las acciones formativas on-line del campus andaluz virtual,” in *VII Foro de Evaluación de la Calidad de la Educación Superior y de la Investigación*, 2010, no. 1, pp. 43–44.
- [312] J. R. Balsas-Almagro, I. Ruano-Ruano, and L. García-Cabrera, “La evaluación de las acciones formativas del campus andaluz virtual ayuda a mejorar su calidad,” in *II Jornadas de Campus Virtuales. Granada 2010*, 2010.
- [313] L. Pérez-Pérez, M. P. Andrés-Romero, J. R. Balsas-Almagro, I. Ruano-Ruano, O. Martín-Rodríguez, and L. García-Cabrera, “A Proposal on E-learning Quality Assessment in Higher Education Initiative of Andalusian Virtual Campus,” in *The future of Education International Conference 1st Ed. (2011)*, 2011, 1st ed.
- [314] J. R. Balsas-Almagro, I. Ruano-Ruano, and L. García-Cabrera, “Análisis de Evaluación Inicial de Asignaturas Virtuales,” in *III JORNADAS INTERNACIONALES DE CAMPUS VIRTUALES*, 2012, pp. 69–72.
- [315] L. García-Cabrera, I. Ruano-Ruano, J. M. Ortega-Tudela, and M. Á. Peña-Hita, “GUÍA DE ESTUDIO DE UNA ASIGNATURA PARA EL CAMPUS ANDALUZ VIRTUAL,” *II Foro Buenas prácticas en Teleformación*, vol. 1, no. 2, 2009.
- [316] L. García-Cabrera, J. M. Ortega-Tudela, M. Á. Peña-Hita, I. Ruano-Ruano, and A. M. Ortiz-Colón, “Guías de estudio para la calidad en la educación,” *Iniciación a la Investig.*, vol. 0, no. e4, 2010.
- [317] I. Ruano-Ruano, J. R. Balsas-Almagro, L. García-Cabrera, J. C. Cuevas-Martínez, J. M. Ortega-Tudela, and M. Á. Peña-Hita, “Estudio y parametrización de pruebas de evaluación para su implementación en docencia virtual,” in *Congreso Internacional de Innovación Docente (CIID 2011)*, 2011, pp. 1383–1394.

- [318] I. Ruano-Ruano, J. R. Balsas-Almagro, L. García-Cabrera, and J. C. Cuevas-Martínez, "Nuevas Pruebas de Evaluación en Plataformas de Docencia Virtual para la Educación Universitaria," in *I Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2011)*, 2011, no. 1, p. 660.
- [319] L. García-Cabrera, I. Ruano-Ruano, and J. R. Balsas-Almagro, "Curso de formación para profesorado universitario en modalidad de b-learning sobre manejo avanzado de un entorno," in *I Jornadas Internacionales de Innovación Docente Universitaria en Entornos de Aprendizaje Enriquecidos*, 2012, 1st ed., pp. 306–308.
- [320] L. García-Cabrera, I. Ruano-Ruano, and J. R. Balsas-Almagro, "A B-Learning Methodology Case for Faculty at High Education," *J. Cases Inf. Technol.*, vol. 15, no. 1, pp. 19–35, Jan. 2013.
- [321] L. García-Cabrera, J. R. Balsas-Almagro, and I. Ruano-Ruano, "Lifelong Learning to Academic Staff by means of Blended Learning & Tailor-Made Courses," in *International GUIDE (Global Universities in Distance Education) Conference 2013*, 2013.
- [322] J. R. Balsas-Almagro, I. Ruano-Ruano, and L. García-Cabrera, "A Proposal for Enhancing Teaching Innovation Through Collaboration Networks on Internet," in *International Conference on Education and New Developments (END 2013)*, 2013, pp. 150–154.
- [323] L. García-Cabrera, J. R. Balsas-Almagro, and I. Ruano-Ruano, "A higher education social network to share and promote teaching innovation experiences," in *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, 2014, pp. 1–8.
- [324] J. R. Balsas-Almagro, I. Ruano-Ruano, L. García-Cabrera, J. M. Ortega-Tudela, and R. Fuentes-Martínez, "Creating self learning, accesible and low-cost mini-videos," in *International Conference on Education and New Developments 2016 (END 2016)*, 2016, pp. 231–235.