

Proyecto Fin de Carrera



Universidad Carlos III de Madrid

Escuela Politécnica Superior

Ingeniería Informática

**SeSo-GEO: E-learning apoyado por la web
semántica y social. Herramienta de Gestión de
Exámenes On-Line.**

Realizado por:

Borja Blanco Iglesias <borja@babieca.org>

Dirigido por:

Ricardo Colomo Palacios <rcolomo@inf.uc3m.es>

Octubre de 2009

“No hay nostalgia peor que añorar lo que nunca jamás sucedió.”
Joaquín Ramón Martínez Sabina

“Paren el mundo que me bajo.”
Groucho Marx

Agradecimientos

Como casi todo en la vida, creo que lo correcto es empezar por el principio, en este caso, dar las gracias a mis padres por todo lo que me han dado, que me ha permitido estar hoy aquí. A ellos les tengo que agradecer muchas cosas, pero especialmente me gustaría agradecerles la educación y los valores que me han inculcado durante tanto tiempo. De igual manera, debo agradecerles a mis hermanos todo el apoyo y el cariño recibido, que ha hecho que ellos siempre hayan sido y seguirán siendo, entre otras muchas cosas, mis mejor amigos. Así mismo, mi agradecimiento a mi tía Amparo, mi segunda madre, y al resto de familia de quien siempre me ha llegado su apoyo.

En toda esta larga etapa universitaria, he conocido a una gran cantidad de personas que de múltiples maneras me han ayudado a que hoy pueda escribir estas líneas. Es realmente difícil acordarse de todas ellas, sin embargo, creo que lo mínimo que puedo hacer es el esfuerzo por recordarlas. Pido perdón por adelantado si dejo en el olvido a alguien.

Mil gracias a “mis Fermines” Fran, Jorge y Víctor; que tantos y tan buenos momentos me han regalado en estos años. Especial cariño guardo de las noches, madrugadas y desayunos de esa primera época universitaria en Aranjuez. En aquellos años conocí a Eneida y a sus hijos, todo un ejemplo de lucha y superación, de quien espero no olvidarme nunca. Así mismo, injusto sería si omito en estos agradecimientos a todos los profesores del CES, especialmente a Diego y Fran. La vida nos separó más de lo que habría querido, pero siento la necesidad desde aquí, de reconocer todo el trabajo que realizan y todo lo que han influido en tantos logros personales que he tenido.

Esa misma vida que a veces te separa de personas de las que no quisieras alejarte, a veces te presenta, de la manera más sorprendente, a personajes curiosos que dejan huella. Claro ejemplo son los “dos Danis” con quien compartí varios años de mi vida y con quien he vivido momentos realmente curiosos. Otro ejemplo de estas personas peculiares, que de alguna forma llegan a nosotros, es Giovanni. Él debe ocupar un papel importante en estos agradecimientos ya que ha estado en la mayor parte de esta andadura, siempre con un espíritu generoso dispuesto a ayudarme en todo momento.

En el terreno profesional, me viene a la mente mi primer jefe José Antonio en Telefónica I+D. A él debo agradecerle su gran confianza depositada en mí y su empujoncito en mi carrera. De otra beca, en este caso en Meta4, me llevé una gran amistad. Siempre he escuchado que los amigos están cuando se les necesitan, y es precisamente cuando Ernesto, casi exclusivamente me atrevería a decir, ha estado ahí. Con toda la sinceridad del mundo muchas gracias Ernesto.

Entrando ya en los dos últimos años de esta carrera de fondo, debo agradecer los grandes momentos que he vivido en mi actual proyecto, desde estas líneas dar las gracias a todo el equipo de Alfa; en especial a Gabri y Nacho que tanto han hecho por mí, y a quien siempre consideraré como mis mentores profesionales. Así mismo, un especial agradecimiento a Isma y Edu, de los que el destino me ha separado, pero con quienes espero seguir en contacto muchos años.

Para ir cerrando estos agradecimientos me gustaría expresar mi gratitud a una persona con nombre y apellidos, Jorge Arribas, quien siempre ha confiado en mí y ha estado ahí con una dosis de motivación cuando la he necesitado.

Como es lógico, en un proyecto fin de carrera nunca debe faltar unas palabras de gratitud hacia la persona que de manera desinteresada ofrece su mano, su ayuda, su experiencia y su consejo para que este proyecto sea una realidad, muchas gracias Ricardo.

Y si alguien ha estado ahí desde el primer día, hasta el último, ese es Diego, la otra parte de este proyecto, muchas gracias por todo compañero.

Por último y como no podía ser de otra manera, dar las gracias a Irene, la niña de mis ojos, tu has hecho, entre tantas otras cosas, que esto cobre un sentido, muchas gracias princesa.

Resumen

La educación a distancia, o e-learning, se ha convertido en los últimos años, en una de las grandes alternativas a la educación presencial tradicional. Este hecho, ha provocado la proliferación de diversos sistemas de evaluación en línea que se han ido desarrollando conjuntamente con los propios sistemas e-learning. Paralelamente al concepto e-learning, han surgido y se han ido formando otros conceptos innovadores entorno a la Web, que han sido bautizados como “Web Social” y “Web Semántica” respectivamente. El presente proyecto, expone un sistema de evaluación de contenidos, en el cual se han aplicado sendos conceptos. Así, SeSo-GEO es una herramienta capaz de generar y compartir a través de búsquedas semánticas, un conocimiento entre distintas plataformas centradas en la enseñanza asistida por ordenador.

Abstract

The e-learning has become a major alternative to traditional classroom education during recent years. This fact has led to the emergence of different online assessment systems that have been developed in conjunction with e-learning systems themselves. Parallel to the e-learning movement, other innovative concepts related to web have appeared, namely "Social Web" and "Semantic Web". This project presents an e-learning system that implements the two concepts. As a result of this effort, SeSo-GEO is a tool capable of generating and sharing through semantic search, knowledge across different platforms devoted to e-learning.

Índice

1. INTRODUCCIÓN.....	2
1.1. DESCRIPCIÓN DEL ÁMBITO DE ESTUDIO.....	2
1.2. PROBLEMÁTICA	4
1.3. DELIMITACIÓN DE LA SOLUCIÓN	6
1.4. ESTRUCTURA DE LA MEMORIA	7
2. ESTADO DEL ARTE	8
2.1. E-LEARNING.....	8
2.1.1. <i>Herramientas E-Learning</i>	11
2.1.2. <i>Evaluación On-Line</i>	16
2.1.3. <i>Estandarización E-Learning</i>	18
2.2. WEB SEMÁNTICA.....	22
2.2.1. <i>Tecnologías de Web Semántica</i>	22
2.2.2. <i>Ontología para la Ingeniería del Software (SEOntology)</i>	32
2.3. WEB SOCIAL.....	34
2.3.1. <i>Tecnologías de Web Social</i>	36
2.3.2. <i>Ejemplos de aplicaciones de Web Social</i>	40
2.4. CONCLUSIONES.....	45
3. DESCRIPCIÓN DE LA SOLUCIÓN.....	48
3.1. DESCRIPCIÓN FUNCIONAL	48
3.1.1. <i>Identificación de roles</i>	48
3.1.2. <i>Definición de elementos de gestión</i>	49
3.1.3. <i>Definición de permisos</i>	51
3.1.4. <i>Definición del gestor de conocimiento</i>	51
3.1.5. <i>Definición del modelo de distribución de datos</i>	52
3.1.6. <i>Definición del motor semántico</i>	53
3.1.7. <i>Definición de la estandarización de información</i>	54
3.2. DESCRIPCIÓN TÉCNICA.....	55
3.2.1. <i>Herramientas de modelado y generación de código</i>	57
3.2.2. <i>Herramientas de desarrollo</i>	57
3.2.3. <i>Frameworks y API's de desarrollo</i>	59
3.2.4. <i>Herramientas de gestión de datos</i>	62
3.2.5. <i>Servidor de Aplicaciones</i>	63
3.2.6. <i>Herramientas de colaboración</i>	63
3.3. METODOLOGÍA DE DESARROLLO	66
3.3.1. <i>Metodologías ágiles</i>	66
3.3.2. <i>XP (Extreme Programming)</i>	68
3.3.3. <i>Ciclo de Vida de XP</i>	69
4. CONCLUSIONES.....	74
5. LÍNEAS FUTURAS	76
BIBLIOGRAFÍA Y REFERENCIAS.....	78
ANEXOS	84
ANÁLISIS DEL SISTEMA DE INFORMACIÓN	84
DOCUMENTO DE ARQUITECTURA DEL SISTEMA	102
DOCUMENTO TÉCNICO DE LA ITERACIÓN 1	132
DOCUMENTO TÉCNICO DE LA ITERACIÓN 2	158
DOCUMENTO TÉCNICO DE LA ITERACIÓN 3	186
DOCUMENTO TÉCNICO DE LA ITERACIÓN 4.....	216
ESTIMACIÓN DE COSTES	230
ANÁLISIS DE COSTES	238

Índice de Figuras

FIGURA 1. INTERFAZ PRINCIPAL DE LA HERRAMIENTA MOODLE	12
FIGURA 2. INTERFAZ PRINCIPAL DE LA HERRAMIENTA CLAROLINE	14
FIGURA 3. DIAGRAMA DE ROLES QUE INTERVIENEN EN QTI	20
FIGURA 4. REPRESENTACIÓN GRÁFICA DE EXPRESIONES EN RDF.....	23
FIGURA 5. EJEMPLO DE GRÁFICO RDF	24
FIGURA 6. EJEMPLO PARA RANGO RDFS	27
FIGURA 7. RELACIONES DE LOS LENGUAJES OWL	31
FIGURA 8. IMAGEN DE LA WIKIPEDIA EN CASTELLANO	40
FIGURA 9. PORTADA DE YOUTUBE EN CASTELLANO	41
FIGURA 10. PORTADA DEL RINCÓN DEL VAGO.	43
FIGURA 11. CONCEPTOS FUNDAMENTALES DE GEO	45
FIGURA 12. COMUNICACIÓN DE SISTEMAS CON GEO.....	55
FIGURA 13. ARQUITECTURA DE GEO.....	56
FIGURA 14. COMPARATIVA ENTRE LAS DIFERENTES METODOLOGÍAS.....	68
FIGURA 15. CICLO DE VIDA DE XP	71
FIGURA 16. CICLO DE VIDA DEL PROYECTO GEO.....	71

Índice de Tablas

TABLA 1. LISTADO LMS'S	15
TABLA 2. EJEMPLO DE ESPACIO DE VALORES.....	25
TABLA 3. COMPARATIVA DE VERSIONES RSS Y RECOMENDACIONES.....	38
TABLA 4. ANÁLISIS DE LA WIKIPEDIA.....	41
TABLA 5. ANÁLISIS DE YOUTUBE	42
TABLA 6. ANÁLISIS DEL RINCÓN DEL VAGO.....	44
TABLA 7. GESTIÓN DE PERMISOS ASOCIADOS A FUNCIONALIDAD	51
TABLA 8. COMPARATIVA DE LAS METODOLOGÍAS ÁGILES FRENTE A LAS TRADICIONALES	66

1. Introducción

Muchas de las investigaciones que se están realizando en los últimos años en el ámbito de la informática, están enfocadas en tres pilares fundamentales: el e-learning o educación a distancia, la Web semántica y la Web social. El presente Proyecto Fin de Carrera tiene como objetivo investigar estos tres conceptos y unificarlos en el desarrollo de un sistema de evaluación online. Dicho desarrollo se realiza tomando como base las innovadoras metodologías ágiles de desarrollo así como las últimas tecnologías Java EE.

1.1. Descripción del ámbito de Estudio

Tal como se describe en la introducción de la sección, este proyecto fin de carrera se marca desde sus comienzos un ambicioso ámbito de estudio, y es que asume como temas centrales, conceptos tan amplios, dispares, debatidos y actuales dentro de la investigación informática, como son el e-learning, la Web Semántica y la Web Social. A continuación se ofrecerán pequeñas pinceladas extraídas de distintas publicaciones sobre cada uno de estos temas.

e-Learning o Educación en línea.

Extraído del “e-learning como herramienta estratégica del grupo Telefónica” del Boletín de la Sociedad de la información [1]. Es en la década de los 80 cuando se comenzaron a diseñar, para diferentes empresas, los primeros módulos de lo que entonces se conocía como “Enseñanza asistida por ordenador” (EAO). La segunda mitad de los 90 marca el inicio del aprendizaje on-line. Las empresas vinculadas al sector tecnológico introducen el uso de Internet en su oferta formativa. Ya en el año 2000 las grandes empresas han sistematizado el aprendizaje on-line incorporando los cursos en sus redes corporativas o intranets ofreciendo una alternativa a la formación continua presencial. La expresión inglesa eLearning (aprendizaje electrónico) se escucha cada vez más en los despachos de los responsables de RRHH y de formación de las empresas.

Son años en los surge un debate importante en torno a la calidad de los contenidos on-line. Las prisas por disponer de los materiales habían ido en detrimento del componente didáctico. Una vez dentro del nuevo siglo no podemos obviar los avances de la iniciativa ADL SCORM ni de la proliferación posterior de las plataformas y herramientas de código abierto. A finales de 2003 ya era posible cursar una titulación universitaria oficial, un máster o un postgrado a través de la Web.

En la actualidad, todos los agentes implicados en el aprendizaje online miran con expectación el impulso que los gobiernos puedan dar a este sector. Las diferentes administraciones públicas tienen como objetivo fundamental, incorporar las nuevas tecnologías en la escuela y los hogares. Ahí tienen mucho que decir los desarrolladores, investigadores y proveedores de contenidos.

Web Social o Web 2.0

Extraído de “Web 2.0” de Wikipedia, la enciclopedia libre [2]. El término de Web 2.0 fue acuñado por Dale Dougherty de O'Reilly Media en una lluvia de ideas con Craig Cline de MediaLive para desarrollar ideas para una conferencia. Dougherty sugirió que la Web estaba en un renacimiento, con reglas que cambiaban y modelos de

negocio que evolucionaban. Dougherty puso ejemplos — "DoubleClick era la Web 1.0; Google AdSense es la Web 2.0. Ofoto es Web 1.0; Flickr es Web 2.0." — en vez de definiciones, y reclutó a John Battelle para dar una perspectiva empresarial, y O'Reilly Media, Battelle, y MediaLive lanzó su primera conferencia sobre la Web 2.0 en Octubre del 2004. La segunda conferencia se celebró en octubre de 2005.

En su conferencia, O'Reilly y Battelle resumieron los principios clave que creen que caracterizan a las aplicaciones web 2.0: la Web como plataforma; efectos de red conducidos por una arquitectura de participación; innovación y desarrolladores independientes; pequeños modelos de negocio capaces de redifundir servicios y contenidos; el perpetuo beta; software por encima de un solo aparato.

En general, cuando mencionamos el término Web 2.0 nos referimos a una serie de aplicaciones y páginas de Internet que utilizan la inteligencia colectiva para proporcionar servicios interactivos en red dando al usuario el control de sus datos. Así, podemos entender como 2.0 -"todas aquellas utilidades y servicios de Internet que se sustentan en una base de datos, la cual puede ser modificada por los usuarios del servicio, ya sea en su contenido (añadiendo, cambiando o borrando información o asociando datos a la información existente), bien en la forma de presentarlos, o en contenido y forma simultáneamente".

Web Semántica

La llegada de la Web Semántica representa una revolución en la forma de acceder y almacenar la información. El término "Web Semántica" fue acuñado por Berners-Lee, Hendler y Lassila [3], para describir la evolución que sigue un documento basado en la Web hacia un nuevo paradigma en el que el documento es capaz de ser manipulado por un ordenador a través de datos e información incluida. La Web Semántica permite acceder de manera automática a la información, a través de la semántica implícita en distintos datos procesables. La Web Semántica fue definida por estos autores como "una extensión de la web actual en la que se da significado al contenido," y puede "permitir que los equipos puedan trabajar en cooperación con un mejor funcionamiento". La Web Semántica ofrece una visión complementaria a un entorno de gestión del conocimiento (Warren, 2006) [4] que, en muchos casos ha sido ampliado e incluso sustituido el conocimiento previo, por arquetipos de gestión de la información (Davies, Lytras y Sheth, 2007) [5]. Naeve (2005) [6] sostiene que la Web Semántica ha iniciado un cambio entre los paradigmas de "impulsar el conocimiento" a los "conocimientos de atracción", como resultado de sus capacidades avanzadas para la integración de información automática. Del mismo modo, Fensel y Musen (2001) [7] consideran a la Web Semántica como "un cerebro para la humanidad", y algunos autores incluso han ampliado esta definición a una "red semántica humana." (Naeve, 2005 [5]; Vossen et al, 2007 [8]).

1.2. Problemática

Tal como se ha expuesto anteriormente, el e-learning o educación a distancia, no es un nuevo concepto que ha aparecido en nuestros días de la nada, sino que poco a poco con el paso de los años, ha llegado a estar en boca de todas las entidades educativas y departamentos de formación empresarial. Si bien es cierto que muchas entidades ya utilizan el concepto en sus ciclos formativos, con el transcurso de los años todavía no se ha dado con “la fórmula” que lleve a la educación a distancia a la consecución de las expectativas que durante tanto tiempo se ha tenido en ella. Si intentamos buscar un único culpable a este hecho, nos será imposible pues existe una gran cantidad de factores y condicionantes que han influido en que el e-learning no haya alcanzado el potencial que verdaderamente esconde. ¿Cuáles son estos factores? ¿Acaso no estaba preparada la sociedad? ¿Acaso no estábamos preparados tecnológicamente? ¿Es éste el momento? La identificación de estos factores, es un tema realmente controvertido en el que podríamos contar con tantas opiniones como investigadores hay actualmente estudiando el tema. Si bien es cierto que sería realmente interesante entrar a estudiar en profundidad estas causas, éste no es el objetivo del proyecto, por lo que a continuación únicamente se va a intentar exponer brevemente la opinión del autor al respecto.

A lo largo de la historia, la enseñanza siempre ha ido ligada y condicionada implícitamente a la sociedad y como todos los cambios sociales, los cambios sobre la educación deben pasar por un tiempo de adaptación y aceptación social. Desde hace unos años, nuestra sociedad se encuentra inmersa en un gran cambio, que algunos han atribuido incluso el grado de revolución, en el que la tecnología y las telecomunicaciones están cambiando nuestra forma de vida, y en un mundo globalizado como el nuestro, están llegando a cambiar todos los ámbitos de la misma, desde el trabajo, los transportes, la medicina y por supuesto la educación. Si bien es cierto, que no podría calificarse como revolucionario el cambio en la enseñanza, también es cierto que poco a poco la sociedad empieza a aceptar un modelo educativo en el que alumnos y profesores toman como respaldo a la formación tradicional, sistemas de e-learning; lo que supone un cambio y un avance al respecto. Sin embargo, son muchos los que piensan que este paso adelante es demasiado pequeño, que aún se ve lejos una enseñanza en el que la figura física del profesor sea totalmente prescindible. Para ello, la educación a distancia necesita un impulso no solo social, sino tecnológico llevando al ámbito de la investigación educativa conceptos tan de moda como Web 2.0 o Web semántica, dotando a los sistemas e-learning de un componente social que involucre al conjunto del personal docente, ofreciéndoles contenidos clasificables semánticamente que facilite su edición y uso a los propios usuarios.

Hay que reconocer que desde los años 80, en los que aparecieron esas primeras plataformas de “Enseñanza asistida por Ordenador”, se ha avanzado mucho hasta llegar a nuestros días. Iniciativas como ADL SCORM han ayudado enormemente a que diversos sistemas e-learning hayan podido adaptar y estandarizar sus contenidos para ser utilizados y compartidos por diversos sistemas. No obstante, a pesar de implementar la funcionalidad necesaria para este intercambio, pocos sistemas, por no decir ninguno, han conseguido implantar política alguna de contenido libre para ser utilizados por otras aplicaciones. En pocas palabras, a pesar de existir el medio a través de un modelo de estandarización creado, todavía no se ha podido implantar una filosofía de Web Social sobre el desarrollo de sistemas e-learning.

Un último punto tecnológico a tener en cuenta es la escasa o nula adaptación de los sistemas de educación a distancia a un entorno de Web Semántica. Si empezamos a hablar de un conocimiento global compartido, en el que un gran número de usuarios pueden acceder a través de diversos sistemas a contenidos educativos, hay que tener en cuenta la importancia de tener un motor semántico a la hora de indexar y tratar las consultas sobre esos contenidos. Es cierto que los motores semánticos todavía no han sido extendidos como el medio demandado, sin embargo es realmente necesario prestar especial atención a este concepto a la hora de diseñar y desarrollar los futuros sistemas de educación online.

Con este pequeño punto introductorio en el que se ha dado a conocer el problema que se plantea en el presente proyecto fin de carrera, se ha intentado dar respuesta a tres de las cuatro preguntas planteadas al principio de la sección. A la finalización del proyecto se espera que el lector pueda responder afirmativamente por muto propio a esa última cuestión: ¿Es éste el momento?

1.3. Delimitación de la solución

Dado lo amplio del ámbito de estudio, así como la generalidad del problema planteado, una de las principales dificultades que se encuentra a la hora de abordar el presente proyecto es la delimitación de la solución; es decir, abordar el problema de una manera realista, analizarlo y plantear un posible desarrollo que dé respuesta a dicho problema desde una perspectiva realista, con una estimación de esfuerzo viable.

Si bien en una primera aproximación, se propuso la implementación de un sistema de e-learning, capaz de gestionar los contenidos semánticamente, para poder ser compartidos con el resto de sistemas, pronto se observó la magnitud del desarrollo así como la inviabilidad del mismo. Es por ello, por lo que se decidió centrarse en un ámbito más reducido del e-learning como es la evaluación Online.

Así, en el presente proyecto se va a abordar el desarrollo de un Sistema de Gestión de Exámenes Online (SeSo-GEO). Dicho sistema, no solo gestionará los datos implicados en este tipo de sistemas, sino que además será capaz de generar exámenes pseudo aleatorios, a partir de una configuración base dada, así como evaluarlos automáticamente.

GEO se desmarca de los sistemas tradicionales de evaluación cuya única función es la generación de exámenes a partir de unas preguntas dadas. Incorporando las últimas tendencias de Web Semántica y Web Social, se introduce el concepto de gestor de conocimiento centralizado y compartido sobre el que realizar búsquedas semánticas de preguntas.

En resumen, se pretende el desarrollo de un sistema de evaluación online, el cual genere contenidos bajo un estándar con meta información para poder ser intercambiados y compartidos con otros sistemas, a la vez que permita la búsqueda semántica del contenido generado.

1.4. Estructura de la memoria

Además de la presente introducción que sirve de presentación al resto de la memoria, ésta se encuentra dividida en cuatro capítulos más: estado del arte, solución planteada, conclusiones y líneas futuras.

En primer lugar, en el capítulo del estado del arte se presentan los resultados de las investigaciones realizadas sobre las tres grandes áreas que componen el presente proyecto: e-learning, Web semántica y Web social. Estos resultados engloban una descripción de cada uno de los puntos, un pequeño estudio de mercado de las distintas herramientas que existen en la actualidad relacionadas con el desarrollo a realizar, así como un análisis de diversos conceptos que serán utilizados con posterioridad en el desarrollo del sistema planteado como solución. Por último y a modo de resumen se sintetizan en un último apartado los puntos más importantes de las tres áreas analizadas.

A continuación, se plantea brevemente la solución propuesta al problema planteado en la presente introducción. Éste tercer capítulo contempla tanto la parte funcional como técnica de la solución, separada en dos secciones propias, además se incluye un tercer punto donde se describe la metodología utilizada en el desarrollo.

Los dos últimos capítulos describen brevemente las conclusiones más relevantes extraídas del desarrollo del proyecto así como las líneas futuras de desarrollo que se dejan planteadas.

Para finalizar la memoria se han incluido a modo de apéndices distintos documentos fruto de las tareas planteadas por la metodología utilizada.

2. Estado del Arte

Uno de los principales puntos del presente proyecto fin de carrera, es el estudio previo sobre el estado actual de cada uno de los principales temas que abarca el proyecto. Como ya se mencionó anteriormente, estos temas son: e-learning, Web semántica y Web Social. A continuación, se profundizará en cada uno de estos temas, dando una visión global sobre lo que existe en la actualidad referente a cada uno, así como las líneas de investigación actuales y las relaciones con el presente proyecto. Al final de la presente sección, se presentará a modo de resumen, un punto en el que se expondrán las conclusiones extraídas del mismo y se detallarán aquellos puntos en común de las distintas partes de estudio.

2.1. E-Learning

Las nuevas tecnologías en general, y el uso de Internet en particular, ha ido penetrando de manera progresiva en cada una de las actividades cotidianas de las personas en los últimos años. Así, actividades tan cotidianas como hacer la compra, ir a los bancos, o comprar el periódico, se están sustituyendo por la compra on-Line, el uso de la banca electrónica o la visualización de contenidos en línea. Como no podía ser de otra manera, la educación también se ha visto “afectada” por la llegada de estas nuevas tecnologías.

Tal como describe D. Francisco José García Peñalvo, profesor de la Universidad de Salamanca, en su artículo “Estado actual de los sistemas de e-Learning” [9]: el sector educativo ha encontrado en estas nuevas tecnologías un excelente medio para romper con las limitaciones geográficas y temporales que los esquemas tradicionales de enseñanza-aprendizaje conllevan, revolucionando, y cambiando a la vez, el concepto de educación a distancia. Su adopción y uso han sido amplios, lo que ha permitido un desarrollo rápido y consistente en el que la Web ha ido tomando distintas formas dentro de los procesos educativos.

La Web se convierte en la infraestructura básica para desarrollar los procesos de enseñanza-aprendizaje no presenciales, combinando servicios síncronos y asíncronos, lo que ha dado lugar a un modelo conocido como e-formación o e-learning, cada vez más valorado, no como sustituto de la formación presencial tradicional, sino más como un complemento que se ha de adaptar según las necesidades y nivel de madurez del público receptor de esta formación [10], que puede ir desde ser una actividad complementaria muy concreta y residual en los estudios de primaria y secundaria, a ser un modelo únicamente no presencial en la formación a distancia o formación continua empresarial. No obstante, las aproximaciones mixtas, que combinan actividades formativas presenciales y no presenciales, toman cada vez más fuerza y se posicionan como una importante alternativa ante los grandes retos que estamos afrontando con la integración del sistema universitario al nuevo Espacio Europeo de Educación Superior y el creciente peso de la formación a lo largo de toda la vida.

Para intentar definir el concepto anteriormente nombrado de e-learning y cuya descripción es el objetivo de este punto, de nuevo se hace referencia al artículo del profesor Peñalvo [9]. Según el mismo, el concepto de e-learning se define de muchas formas diferentes fundamentalmente debido a que los actores que de él hacen uso son muy diversos, cada uno con su idiosincrasia y su ámbito de aplicación.

Desde la perspectiva de su concepción y desarrollo como herramienta formativa, los sistemas de e-learning tienen una dualidad pedagógica y tecnológica. Pedagógica en cuanto a que estos sistemas no deben ser meros contenedores de información digital, sino que ésta debe ser transmitida de acuerdo a unos modelos y patrones pedagógicamente definidos para afrontar los retos de estos nuevos contextos. Tecnológica en cuanto que todo el proceso de enseñanza-aprendizaje se sustenta en aplicaciones software, principalmente desarrolladas en ambientes Web, lo que le vale a estos sistemas el sobrenombre de plataformas de formación.

Desde la perspectiva de su uso se podría distinguir la visión que tienen sus usuarios finales, que con independencia de su madurez y formación, verán al sistema e-learning como una fuente de servicios para alcanzar su cometido formativo. No obstante, también es factible diferenciar una visión de organización, en la que se definen el alcance y los objetivos buscados con la formación basada en estos sistemas, distinguiéndose una visión académica y una visión empresarial.

Si se toma como referencia la raíz de la palabra, e-learning se traduce como “aprendizaje electrónico”, y como tal, en su concepto más amplio puede comprender cualquier actividad educativa que utilice medios electrónicos para realizar todo o parte del proceso formativo.

Existen definiciones que abren el espectro del e-learning a prácticamente a cualquier proceso relacionado con educación y tecnologías, como por ejemplo la definición de la *American Society of Training and Development* que lo define como “término que cubre un amplio grupo de aplicaciones y procesos, tales como aprendizaje basado en web, aprendizaje basado en ordenadores, aulas virtuales y colaboración digital. Incluye entrega de contenidos vía Internet, intranet/extranet, audio y vídeo grabaciones, transmisiones satelitales, TV interactiva, CD-ROM y más”.

Otros autores acotan más el alcance del e-learning reduciéndolo exclusivamente al ámbito de Internet, como Rosenberg [11] que lo define como: “el uso de tecnologías Internet para la entrega de un amplio rango de soluciones que mejoran el conocimiento y el rendimiento. Está basado en tres criterios fundamentales:

1. El e-learning trabaja en red, lo que lo hace capaz de ser instantáneamente actualizado, almacenado, recuperado, distribuido y permite compartir instrucción o información.
2. Es entregado al usuario final a través del uso de ordenadores utilizando tecnología estándar de Internet.
3. Se enfoca en la visión más amplia del aprendizaje que van más allá de los paradigmas tradicionales de capacitación”.

Desde la perspectiva que ofrece la experiencia en el desarrollo y explotación de plataformas e-learning, podríamos aventurarnos a dar nuestra propia definición de e-learning como la capacitación no presencial que, a través de plataformas tecnológicas, posibilita y flexibiliza el acceso y el tiempo en el proceso de enseñanza-aprendizaje, adecuándolos a las habilidades, necesidades y disponibilidades de cada discente, además de garantizar ambientes de aprendizaje colaborativos mediante el uso de herramientas de comunicación síncrona y asíncrona, potenciando en suma el proceso de gestión basado en competencias.

En todas estas definiciones, así como en otras que se pueden encontrar en la bibliografía especializada, se acaba haciendo mención explícita o implícita a lo que se viene llamando en triángulo del e-learning [12], formado por la tecnología (plataformas, campus virtuales...), los contenidos (calidad y estructuración de los mismos se toman como elementos capitales para el éxito de una iniciativa de e-formación) y los servicios (siendo el elemento más variopinto que engloba la acción de los profesores, elementos de gestión, elementos de comunicación, elementos de evaluación...). Variando el peso de estos tres componentes se obtienen diferentes modelos de e-formación, de igual forma que variando las variables y recursos con los que cuenta un profesor se obtienen diferentes políticas de docencia presencial.

En la práctica, para llevar a cabo un programa de formación basado en e-learning, se hace uso de plataformas o sistemas de software que permiten la comunicación e interacción entre profesores, alumnos y contenidos. Se tienen principalmente dos tipos de plataformas: las que se utilizan para impartir y dar seguimiento administrativo a los cursos en línea o LMS (*Learning Management Systems*) y, por otro lado, las que se utilizan para la gestión de los contenidos digitales o LCMS (*Learning Content Management Systems*).

Entre las herramientas más utilizadas para los ambientes o sistemas e-learning están los Sistemas de Administración de Aprendizaje o LMS, también ampliamente conocidos como plataformas de aprendizaje. Un LMS es un software basado en un servidor web que provee módulos para los procesos administrativos y de seguimiento que se requieren para un sistema de enseñanza, simplificando el control de estas tareas. Los módulos administrativos permiten, por ejemplo, configurar cursos, matricular alumnos, registrar profesores, asignar cursos a un alumno, llevar informes de progreso y calificaciones. También facilitan el aprendizaje distribuido y colaborativo a partir de actividades y contenidos preelaborados, de forma síncrona o asíncrona, utilizando los servicios de comunicación de Internet como el correo, los foros, las videoconferencias o el chat.

El alumno interactúa con la plataforma a través de una interfaz web que le permite seguir las lecciones del curso, realizar las actividades programadas, comunicarse con el profesor y con otros alumnos, así como dar seguimiento a su propio progreso con datos estadísticos y calificaciones. La complejidad y las capacidades de las plataformas varían de un sistema a otro, pero en general todas cuentan con funciones básicas como las que se han mencionado.

Los Sistemas de Administración de Contenidos de Aprendizaje o LCMS tienen su origen en los CMS (*Content Management System*) cuyo objetivo es simplificar la creación y la administración de los contenidos en línea, y han sido utilizados principalmente en publicaciones periódicas (artículos, informes, fotografías...). En la mayoría de los casos lo que hacen los CMS es separar los contenidos de su presentación y también facilitar un mecanismo de trabajo para la gestión de una publicación web. Los LCMS siguen el concepto básico de los CMS, que es la administración de contenidos, pero enfocados al ámbito educativo, administrando y concentrando únicamente recursos educativos y no todo tipo de información.

En esencia, se define entonces un LCMS como un sistema basado en web que es utilizado para crear, aprobar, publicar, administrar y almacenar recursos educativos y cursos en línea [13]. Los principales usuarios son los diseñadores instruccionales que

utilizan los contenidos para estructurar los cursos, los profesores que utilizan los contenidos para complementar su material de clase e incluso los alumnos en algún momento pueden acceder a la herramienta para desarrollar sus tareas o completar sus conocimientos.

Los contenidos usualmente se almacenan como objetos descritos e identificables de forma única. En un LCMS se tienen contenedores o repositorios para almacenar los recursos, que pueden ser utilizados de manera independiente o directamente asociados a la creación de cursos dentro del mismo sistema. Es decir que el repositorio puede estar disponible para que los profesores organicen los cursos o también pueden estar abiertos para que cualquier usuario recupere recursos no vinculados a ningún curso en particular, pero que les pueden ser de utilidad para reforzar los aprendidos sobre algún tema.

El proceso de trabajo dentro de un LCMS requiere de control en cada fase del contenido, esto conlleva un proceso editorial para controlar la calidad de los contenidos creados, así como para permitir y organizar su publicación.

Siguiendo con el completo artículo del profesor Peñalvo [9], podemos concluir que los sistemas que promueven los procesos de enseñanza-aprendizaje a través de sistemas de e-learning tienen una gran importancia para consolidar la denominada Sociedad del Conocimiento. Estos medios abren la puerta para la formación básica o avanzada a una importante cantidad de personas, que pueden ver mejorada su cualificación personal o su situación profesional. Estos sistemas tienen un campo enorme de aplicación ya que la formación puede orientarse de forma complementaria a nivel de educación primaria y secundaria, de forma complementaria o exclusiva a nivel universitario, de postgrado o de formación continua, y de formación especial a medida en las empresas. No obstante, el campo del e-learning está en sus fases iniciales y le falta un largo camino por recorrer hasta alcanzar su madurez y consolidación.

2.1.1. Herramientas E-Learning

Actualmente nos encontramos en pleno auge de los sistemas de gestión de formación en línea, lo que supone que día a día vayan apareciendo nuevas herramientas tanto de libre distribución como de desarrollo privado.

Tras un arduo análisis de mercado de estas herramientas, se ha optado por profundizar en dos de las herramientas de libre distribución más extendidas hasta el momento como son Moodle y Claroline.

La elección de estas dos plataformas de e-learning para el presente estudio, se basa en la gratuidad de las mismas, la gran implantación que se está realizando en la actualidad, así como los amplios servicios de que disponen y que permitirán mostrar la imagen típica de un LMS.

Para la realización de este análisis de las herramientas, se ha procedido a la implantación de las mismas en la máquina de desarrollo del proyecto “desarrollo.babioca.org”.

Moodle

Moodle, cuyo nombre viene del acrónimo inglés “*Modular Object-Oriented Dynamic Learning Environment*” (Entorno Modular de Aprendizaje Dinámico

Orientado a Objetos), es un LMS desarrollado inicialmente en agosto de 2002 en la Universidad Tecnológica de Curtin. Su creador, Martin Dougiamas; diseñó un sistema bajo una corriente de pensamiento que se ha denominado "pedagogía constructorista social" [14] que ha llevado a este sistema a convertirse en uno de los entornos de e-learning más utilizados actualmente.

Podríamos resumir los aspectos técnicos muy brevemente, dejando constancia de que es un sistema desarrollado íntegramente en PHP abstrayéndose completamente del SGBD y utilizando una única BBDD para la persistencia de todos los datos.

Además de la filosofía de desarrollo del sistema, uno de los aspectos más importante de Moodle es la enorme funcionalidad y flexibilidad de la que está dotado el sistema. Dicha funcionalidad permite al administrador, tal como veremos posteriormente, configurar absolutamente todos los aspectos relevantes de un LMS así como crear una completa agenda de cada curso añadiendo recursos y actividades.



Figura 1. Interfaz principal de la Herramienta Moodle

La administración cuenta con una infinidad de opciones de configuración, sin embargo la exposición detallada de cada una de estas opciones recae fuera del ámbito de este proyecto. A continuación se enumerarán las opciones generales:

- Gestión de usuarios
- Gestión de cursos
- Administración de calificaciones
- Administración de ubicación e idioma.
- Administración módulos.
- Administración de seguridad.
- Administración de apariencia y portada.
- Administrar del servidor.

- Administrar listas de enlaces.
- Administración de red.
- Generación de informes.

Dentro de la gestión y configuración de cada curso encontramos las siguientes opciones:

- Configuración del curso.
- Asignación de roles.
- Gestión de Calificaciones.
- Gestión de grupos.
- Gestión de copias de seguridad.
- Administración de informes

Cada curso se encuentra estructurado a través de una agenda en la cual diariamente pueden ser añadidos recursos como etiquetas, páginas de texto y HTML, enlaces a documentos, directorios y paquetes con contenido IMS; así como actividades del tipo de Chat, consultas, cuestionarios, encuestas, foros, glosarios, lecciones y contenidos SCORM.

Para el análisis de esta herramienta se ha implantado en una máquina de desarrollo del proyecto accesible desde la URL <http://moodle.desarrollo.babioca.org>.

Tras la evaluación de cada una de las funcionalidades del sistema, concluimos que Moodle es uno de los LMS más completos, con mayor posibilidad de parametrización y todo bajo una filosofía *free software* lo que hace que sea uno de las herramientas e-learning más extendidas. Como puntos fuertes podríamos destacar varios: la adaptación a los estándares QTI y SCORM le permiten la importación de contenidos de una manera realmente sencilla a la vez que permitirán que los contenidos sean utilizados por otros. La disposición de los cursos en forma de agenda dentro de cada curso, permite al alumno tener una visión cronológica del propio curso de un solo vistazo. En lo referente a los recursos, cabe destacar la inclusión de BBDD, que permite la creación de manera totalmente dinámica de tablas de contenidos personales. Además de otros recursos típicos de un LMS se han incluido una gran variedad de actividades que el alumno puede ir realizando cada día. Entre estas actividades cabe destacar la inclusión de un Chat para una comunicación más ágil a modo de tutorías, así como foros y glosarios.

Para terminar y centrarnos en el desarrollo posterior del proyecto, cabe destacar el completo sistema de evaluación mediante cuestionarios del que dispone el sistema. En él se contemplan gran variedad de tipos de preguntas como preguntas calculadas, descripciones, ensayos, emparejamientos, respuestas incrustadas, opción múltiple o respuestas cortas. Este sistema será analizado de una manera más completa posteriormente.

Claroline

El proyecto Claroline, cuyo logotipo es el rostro de Calíope, la musa griega de la poesía épica y la elocuencia; fue iniciado en el año 2000, en el Instituto Pedagógico

Universitario de Multimedia de la Universidad Católica de Lovain (Bélgica). A él se han ido uniendo una gran comunidad de profesores y desarrolladores y a partir del año 2004 el Centro de Investigación y Desarrollo del Instituto Superior de Ingeniería Belga.

Técnicamente cabe destacar que la plataforma se encuentra escrita íntegramente en PHP y utiliza como sistema gestor de base de datos MySQL, lo cual contribuye en gran medida a su amplia distribución.

Uno de los aspectos más interesantes de esta plataforma, es la implementación dentro de las especificaciones de SCORM e IMS. Tal como se verán en secciones posteriores estas especificaciones marcan las bases de la accesibilidad, interoperabilidad y reusabilidad de los contenidos utilizados; además de ser uno de los principales objetivos de análisis del proyecto.

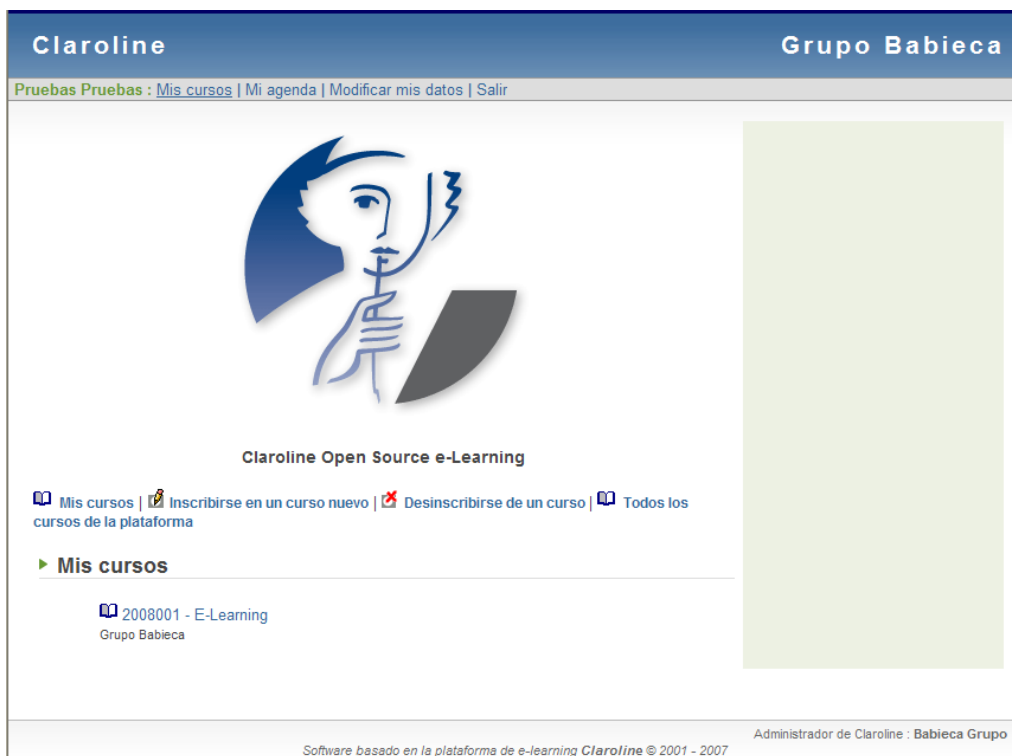


Figura 2. Interfaz principal de la Herramienta Claroline

Funcionalmente presenta las características propias de un sistema de gestión de contenidos (LMS). Las principales funcionalidades son:

- Gestión de cursos
- Gestión de agenda con tareas y plazos.
- Gestión de anuncios incluyendo el envío vía e-mail a usuarios registrados.
- Gestión de documentos y enlaces.
- Administración de cuestionarios y sistema de evaluación, incluyendo estadísticas de los mismos.
- Administración de secuencias de aprendizaje.
- Confeción de ejercicios para que sean realizados por los estudiantes.

- Administrar foros de discusión tanto públicos como privados.
- Administrar listas de enlaces.
- Administración de grupos de estudiantes.
- Gestión los envíos de los estudiantes: documentos, tareas, trabajos, etc.
- Crear y guardar chats.
- Creación de un conocimiento compartido a través de “Wiki’s”.

Para el análisis de esta herramienta se ha implantado en una máquina de desarrollo del proyecto accesible desde la URL <http://claroline.desarrollo.babieca.org>.

Tras la evaluación de cada una de las funcionalidades, cabe destacar el sistema de gestión de cuestionarios que incluye, el cual permite la gestión de varios parámetros del cuestionario como son las fechas de realización, el tiempo, el número de intentos, etc. Además cuenta con diversos tipos de preguntas: elección simple, elección múltiple, rellenar huecos y relacionar. Por último en lo referente al sistema de evaluación, volver a hacer referencia en este punto a su adaptación al estándar QTI de IMS lo que le permite exportar los cuestionarios y ser interpretados por otro sistema.

Para finalizar el análisis de la herramienta Claroline destacar también la gestión de ejercicios, que permite la configuración de varios parámetros como son: fechas de entrega, tipo de envío, ejercicio en grupo o individual, etc. Además permite al profesor la evaluación de cada uno de las entregas añadiendo la calificación y comentarios de los mismos.

Otras herramientas de e-learning

Dado que un mayor análisis de las herramientas de e-learning encontradas, recae fuera del alcance del estudio, se presenta la siguiente tabla a modo de resumen, donde el lector podrá encontrar más información relativa a dichas herramientas.

Nombre	Referencia	Libre
.LRN	http://dotlrn.org/	x
ATutor	http://www.atutor.it	x
Authorware	http://www.adobe.com/	
Blackboard	http://www.blackboard.com/us/index.bbb	
Claroline	http://www.claroline.net/	x
Delfos LMS	http://franco-atigroup.com/delfos/	
Desire2Learn	http://www.desire2learn.com/	
Docebo	http://www.docebo.org	x
Dokeos	http://sourceforge.net/projects/dokeos	x
EKP	http://www.netdimensions.com/	
ILIAS	http://www.ilias.de	x
LON-CAPA	http://www.lon-capa.org/	x
Moodle	http://moodle.org/	x
SITEA	http://www.sitea.net/	
Skillfactory	http://www.skillfactory.com.mx/	
Vertice	http://www.verticelearning.com	

Tabla 1. Listado LMS's

2.1.2. Evaluación On-Line

No debemos olvidar que dentro del ámbito del e-learning, el presente proyecto se centra sobre el importantísimo proceso de evaluación on-line. Y es que tal como expone Juan Carlos Lozano, en su artículo “Técnicas y Herramientas de Evaluación On-Line” [15]: evaluar es una de las etapas más importantes dentro del proceso de enseñanza-aprendizaje y no se debe confundir evaluación con calificación, ya que sólo es un aspecto más del proceso evaluativo que está relacionado con la valoración o notas finales y tiene una función exclusivamente acreditativa.

Es decir, no sólo se evalúa al alumno para certificar el nivel de conocimientos adquirido a lo largo de todo el proceso, sino que también se evalúa para ofrecerle una retroalimentación sobre su aprendizaje, para que los docentes conozcan la efectividad de su actuación, para certificar los resultados, para evaluar la calidad de la metodología empleada, etc.

La formación online supera a otros tipos de formación tradicionales porque en ella se pueden evaluar, incluso en muchos casos de forma automática, los siguientes aspectos:

- La asistencia: se puede conocer el número de accesos, el tiempo empleado por los diferentes participantes de la acción formativa, etc. y esto puede servir para justificar las horas lectivas del curso.
- Las aportaciones: se puede conocer también el grado participación los participantes de la acción formativa (alumnos, docentes y coordinador), el número de mensajes enviados, intervenciones en los foros, etc.
- Los conocimientos: a través de técnicas e instrumentos de evaluación como las autoevaluaciones, ejercicios, exámenes, etc. se puede medir el grado de aprendizaje alcanzado por el alumno.
- El proceso formativo en su totalidad: se puede medir el grado de eficacia y eficiencia del curso, su atractivo, su usabilidad, etc.

La formación online cuenta con unas posibilidades casi ilimitadas para realizar la evaluación. Tanto en las plataformas como en los contenidos online se pueden incluir herramientas de evaluación interactivas y dinámicas que ofrecen por un lado, un feedback inmediato al alumno sobre los resultados alcanzados, y por otro lado, permiten a los gestores de la formación disponer de datos cuantitativos generados automáticamente por el sistema, que facilitan enormemente la tarea de evaluar.

A pesar de estas facilidades tampoco habrá que descartar en muchos casos la intervención y juicio de un docente o tutor que evalúe aspectos cualitativos, a través de la actuación del alumno en los distintos contextos y por supuesto mediante pruebas de toda la vida, que implique una mayor elaboración por parte del alumno, como por ejemplo, un proyecto fin de curso.

De este modo nos encontramos con técnicas objetivas (valoración cuantitativa), técnicas subjetivas (valoración cualitativa) y técnicas mixtas.

Siguiendo con la clasificación del profesor Lozano, encontramos técnicas objetivas como los cuestionarios y ejercicios interactivos del tipo de: preguntas de elección múltiple, doble alternativa, asociación de parejas, rellenado de huecos,

ordenación de elementos, identificación de términos y clasificación de elementos. Otras técnicas objetivas pueden ser juegos interactivos como son los crucigramas, sopa de letras rompecabezas y puzles.

En el otro lado de las técnicas encontramos las subjetivas, como son la exposición oral y redacción escrita, resolución de problemas, dinámica de grupos, representación de roles, casos prácticos así como el proyecto fin de curso.

Para obtener una descripción de estas técnicas puede consultarse el artículo de Juan Carlos Lozano en su totalidad [15].

Actualmente están empezando a proliferar gran cantidad de herramientas de pago incluso algunas de código libre y/o libre distribución para la evaluación on-line. A continuación se expondrán de manera muy breve una pequeña representación de las herramientas encontradas que han sido elegidas como representación de los múltiples sistemas analizados:

Autor Vértice

La Herramienta *Autor* [16] es un software desarrollado por vértice e-learning cuya última versión es la 2.1. *Autor* es compatible con la versión 1.2 de SCORM por lo que se podrán generar y exportar contenidos bajo esta especificación. El sistema dispone de editores de contenidos, esquemas flash y evaluaciones integrados. Además cuenta con una sencilla edición de páginas así como el uso de plantillas para la generación de contenidos.

Perception Scantron

Perception [17] es la herramienta desarrollada por Scantron para la creación de test y encuestas a través de Internet y que completa su gama de software de evaluación. *Perception* cuenta con gran variedad de preguntas, configuraciones de test, opciones de seguridad e incluso opciones de reporte. Todas estas características se encuentran en la página oficial del sistema [17]. Además de *Perception* cabe la pena destacar el software de generación de exámenes *ExamView* de Scantron que permite generar exámenes automáticamente de 2000 bancos de preguntas.

CrearTest.com

CrearTest.com [18] es una novedosa iniciativa online para la creación y realización de test y a través de Internet. Si bien no añade ninguna novedad tecnológica ni funcional significativa a las herramientas analizadas, se ha querido incluir en este pequeño análisis por ser unas de las primeras herramientas de evaluación en aplicar el nuevo concepto de Web Social; concepto tan importante en este proyecto y que será estudiado en más profundidad en el próximo punto del presente Estado del Arte.

CrearTest es una sencilla herramienta a través de la cual, cualquier persona puede crear y compartir sus propios exámenes para que otros usuarios los pueda realizar. La funcionalidad que ofrece es muy limitada ya que únicamente se contempla la creación y realización de exámenes. Además no hay ningún control sobre los contenidos publicados lo que hace que la información publicada sea escasamente útil, y su recuperación altamente compleja. Por último resaltar la poco creativa interfaz de usuario lo que hace poco atractivo su uso.

iTest

iTest [19] es la continuidad del Centro de Estudios Felipe II adscrito a la Universidad Complutense de Madrid al proyecto Exanet [21] desarrollado por alumnos de este mismo centro durante el curso académico 2003/04 tal como se describe en un artículo publicado por la profesora Nuria Joglar [20] en 2007.

iTest permite la evaluación y la auto-evaluación en academia a todos los niveles educativos. iTest genera exámenes tipo test al azar, los corrige automáticamente, permite la revisión de los mismos *a posteriori* por parte del alumno y genera listados de calificaciones y estadísticas sobre los resultados para el profesor.

Por último, no podemos olvidar que todo buen sistema LMS debe tener un motor de evaluación. Es por esto por lo que también deben tenerse en cuenta aquellos sistemas LMS referenciados con anterioridad y que no se han expuesto en esta sección por no caer en la redundancia en las referencias.

2.1.3. Estandarización E-Learning

Uno de los puntos más importantes y sobre el que más hincapié se está haciendo en el desarrollo de las nuevas tecnologías en general, y del e-learning en particular, es la definición de estándares. Mediante la utilización de estos estándares, se garantiza la independencia de los contenidos generados por los LCMS y utilizados por los LMS, de forma que se cumplan estas cuatro especificaciones que debería cumplir cualquier CMS:

- Accesibilidad: los contenidos son independientes de la plataforma en la que estén.
- Interoperabilidad: el contenido puede ser usado en diferentes plataformas
- Reusabilidad: los contenidos pueden ser utilizados en diferentes sistemas.
- Durabilidad: el contenido podrá utilizarse sin importar los cambios en la tecnología con la cual se elaboró.

Con la aplicación de los estándares se posibilita la libre elección de los proveedores de contenidos y herramientas, así como la reutilización de los cursos en diferentes plataformas y sistemas.

Actualmente hay diversos estándares utilizables, como son el AICC (desarrollado por la industria de la aviación de EEUU), IEEE LTSC (Instituto de Ingenieros Electrónicos e Informáticos), IMS (del Global Learning Consortium), y el más utilizado y extendido: SCORM. A continuación se explicará brevemente este último, así como IMS ya que será el estándar sobre el que se basará el presente proyecto.

SCORM

El Modelo Referenciado de Objetos de Contenido Compartible (**SCORM** , por sus siglas en inglés, *Sharable Content Object Reference Model.*) representa el conjunto de especificaciones que permiten desarrollar, empaquetar y entregar materiales educativos. Estos paquetes pueden incluir páginas Web, gráficos, Javascript, Flash y cualquier otro contenido que pueda ser representado en un navegador Web.

Las especificaciones de SCORM [22], distribuidas por ADL, detallan cómo deben de publicarse los contenidos y usarse los metadatos; también, incluyen las especificaciones para representar la estructura de los cursos por medio de XML y el uso de API.

Se puede decir que SCORM consta de tres componentes:

- Empaquetamiento de contenidos: se refiere a la manera en que se guardan los contenidos de un curso, el modo en que están ligados entre sí y la forma en la que se entregará la información al usuario. Todos estos datos se concentran en un archivo llamado *manifest.xml*
- Ejecución de comunicaciones: detalla el ambiente para ejecutar la información y consta de dos partes, los comandos de ejecución y los metadatos del estudiante.
- Metadatos del curso: son de dos tipos, los que incluyen la información del curso en sí, y los que contienen el material del estudiante.

Con la aplicación de SCORM se hace posible el crear contenidos que puedan importarse dentro de sistemas de gestión de aprendizaje diferentes, siempre que estos soporten el estándar.

Puesto que SCORM no será utilizado en el presente proyecto, no se ha creído oportuno una extensión sobre los detalles técnicos y funcionales del mismo, para una mayor información puede consultarse la página oficial de la especificación [22].

IMS

El IMS Global Learning Consortium(IMS GLC) [23] es una compañía global, sin fines de lucro cuyo objetivo principal es la creación de estándares en el ámbito del e-learning. Actualmente IMS GLC está formada por más de 140 organizaciones del ámbito tecnológico y de la educación.

Desde su creación en 1997, IMS ha desarrollado alrededor de 20 normas las cuales son ampliamente utilizadas en aplicaciones tecnológicas de aprendizaje de todo el mundo. Estas normas incluyen estandarización de meta-datos, empaquetado de contenidos, servicios empresariales, preguntas y test, información de alumnos, definición de vocabulario, herramientas de interoperabilidad así como otros aspectos dentro del e-learning. Si bien es realmente interesante cada una de estas normas, por recaer fuera del alcance de este estudio, vamos a realizar una pequeña descripción de una de ellas, ya que es el estándar a desarrollar dentro del presente proyecto: IMS Question & Test Interoperability Specification (QTI) [24].

La especificación QTI describe un modelo de representación de datos para las preguntas, los test, los datos generados así como los correspondientes informes de resultados. Por lo tanto esta especificación permite el intercambio de datos entre sistemas LCMS, LMS así como cualquier otra herramienta que gestione este tipo de información.

El modelo de datos QTI está descrito de una manera abstracta, utilizando UML para facilitar la unión de una amplia gama de modelo de datos, herramientas y lenguaje de programación, además, para el intercambio de datos se utiliza XML. Además hay que tener en cuenta que QTI ha sido diseñado para apoyar la interoperabilidad y la innovación por lo que han sido definidos varios puntos de extensión del estándar que pueden ser utilizados para un diseño a medida de cada herramienta.

Básicamente QTI está diseñado para:

- Proporcionar un formato del contenido bien documentado a través del cual se pueda almacenar e intercambiar test y preguntas independientemente de la herramienta utilizada para crearlos.
- Apoyar el despliegue de los datos a través de una amplia gama de LMS.
- Proporcionar un sistema con la capacidad de informar de los resultados obtenidos en los test de una forma coherente.

A continuación se presenta un pequeño diagrama con los roles y sistemas representados en QTI:

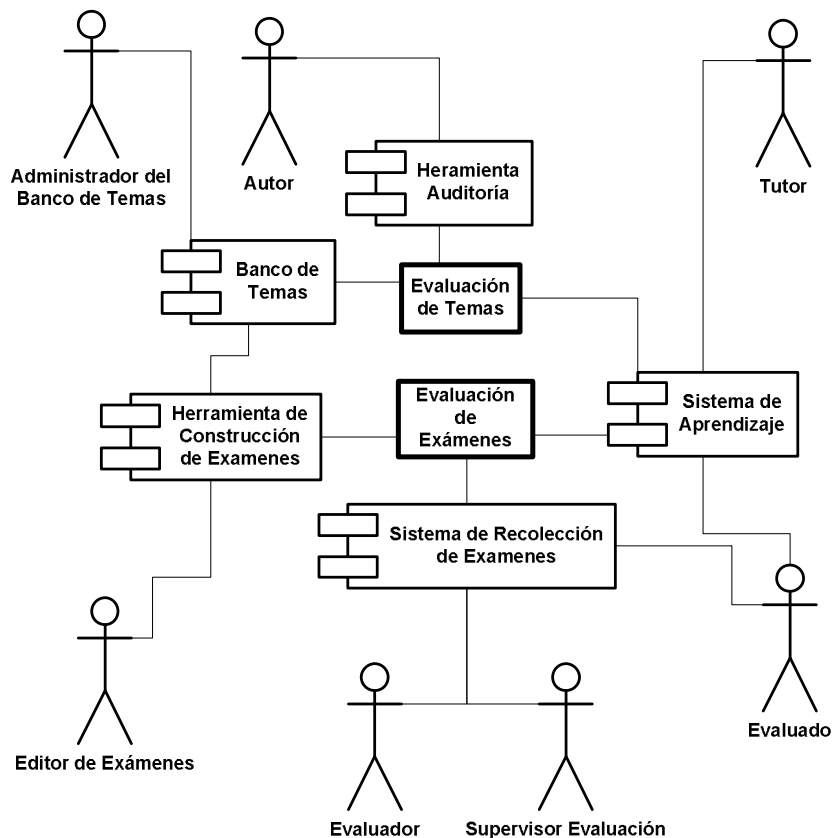


Figura 3. Diagrama de Roles que intervienen en QTI

Sistemas identificados por la especificación QTI:

- Herramienta de Auditoria: Sistema utilizado por un autor para crear o modificar una evaluación.
- Banco de Temas: Un sistema de recopilación y gestión de temas.
- Herramienta de construcción de exámenes: Un sistema de creación de exámenes.
- Sistema de recolección de exámenes: Sistema que gestiona automáticamente la entrega de los resultados de una evaluación al evaluador.
- Sistema de aprendizaje: Lo que comúnmente hemos llamado LMS. Aquel sistema que gestiona todo lo relacionado con las tareas de gestión de aprendizaje.

Roles identificados por la especificación QTI:

- Autor: En situaciones simples los temas pueden ser creados por un solo autor, sin embargo en situaciones más complejas un tema puede tener que pasar a través de un proceso de creación con un control de calidad asociado. En esta situación podemos identificar la figura del autor que a través de la herramienta de auditoría valida esos temas creados.
- Administrador del banco de temas: Encargado de la gestión de los elementos de las evaluaciones.
- Editor de Exámenes: Se encarga de la construcción de exámenes, extrayendo los temas del banco de temas.
- Supervisor de Evaluación: Aquella persona o sistema encargado de la supervisión de la correcta entrega de una evaluación.
- Evaluador: Persona o sistema encargado de realizar la evaluación de una evaluación.
- Tutor: Cualquier persona que participe en la gestión, dirección y tareas de apoyo en el proceso de aprendizaje del alumno.
- Evaluado: Aquella persona que está siendo evaluada.

2.2. Web Semántica.

A menudo, el concepto de “Web semántica” resulta confuso y difuso. Conviene definir, antes de comenzar, qué es exactamente. Como ya se mencionó anteriormente y según la “Guía Breve de Web Semántica” del W3C [25], “La Web Semántica es una Web extendida, dotada de mayor significado en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida. Al dotar a la Web de más significado y, por lo tanto, de más semántica, se pueden obtener soluciones a problemas habituales en la búsqueda de información gracias a la utilización de una infraestructura común, mediante la cual, es posible compartir, procesar y transferir información de forma sencilla. Esta Web extendida y basada en el significado, se apoya en lenguajes universales que resuelven los problemas ocasionados por una Web carente de semántica en la que, en ocasiones, el acceso a la información se convierte en una tarea difícil y frustrante”.

Es decir, consiste en dotar en la Web de significado a la información publicada, para facilitar la tarea de los sistemas de recuperación y acceso a la información. Esta incorporación de semántica a los datos, se lleva a cabo mediante la inclusión de “meta-datos”, es decir, “datos sobre los datos”, que nos indican información que, si bien un humano puede deducir del contexto, y la “lectura entre líneas”, una máquina necesita que sean especificados. La comprensión de esos datos por parte de las máquinas, se basa en las definiciones y relaciones entre los datos y meta datos, y las propias búsquedas. “No se trata de una inteligencia artificial mágica que permita a las máquinas entender las palabras de los usuarios, es sólo la habilidad de una máquina para resolver problemas bien definidos, a través de operaciones bien definidas que se llevarán a cabo sobre datos existentes bien definidos” [25].

2.2.1. Tecnologías de Web Semántica.

Una vez que nos hemos centrado en qué es exactamente la Web Semántica, podemos pasar a describir cómo funciona con mayor detalle. Se podría decir que por un lado tenemos las tecnologías de definición de meta datos, y por otro, las tecnologías capaces de interpretar esos meta datos para definir y resolver problemas.

La Web Semántica utiliza esencialmente RDF, SPARQL, y OWL.

- RDF proporciona información descriptiva simple sobre los recursos que se encuentran en la Web y que se utiliza, por ejemplo, en catálogos de libros, directorios, colecciones personales de música, fotos, eventos, etc.
- SPARQL es lenguaje de consulta sobre RDF, que permite hacer búsquedas sobre los recursos de la Web Semántica utilizando distintas fuentes datos.
- OWL es un mecanismo para desarrollar temas o vocabularios específicos en los que asociar esos recursos. Lo que hace OWL es proporcionar un lenguaje para definir ontologías estructuradas que pueden ser utilizadas a través de diferentes sistemas. Las ontologías, que se encargan de definir los términos utilizados para describir y representar un área de conocimiento, son utilizadas por los usuarios, las bases de datos y las aplicaciones que necesitan compartir información específica, es decir, en

un campo determinado como puede ser el de las finanzas, medicina, deporte, etc. Las ontologías incluyen definiciones de conceptos básicos en un campo determinado y la relación entre ellos.

En resumen: RDF permite mostrar conocimiento. SPARQL permite hacer consultas sobre ese conocimiento, y OWL permite definir las relaciones entre distintos términos del conocimiento mostrado. Es necesario remarcar que estas tres tecnologías son complementarias al pertenecer a ámbitos distintos (representación, consulta y estructuración). A continuación, entraremos más en detalle en cada una de las tecnologías.

Resource Description Framework (RDF).

Según la especificación RDF del W3C [26], el “Resource Description Framework” es un marco de trabajo para representar información en Web. Como adelantamos en el apartado anterior, se encarga de “mostrar la información”.

El desarrollo de RDF ha sido motivado para varios usos, entre otros, proveer información sobre recursos Web, y los sistemas que los usan; permitir el desarrollo de aplicaciones que precisen modelos de información abiertos, hacer procesable la información para máquinas a nivel global, permitir la combinación de información de diferentes aplicaciones para obtener una nueva información, y hacer posible que los agentes software automaticen tareas de acceso a información. RDF viene a ser un lenguaje común, basando en XML, mediante el cual todos estos elementos pueden entenderse.

RDF está basado en un modelo gráfico de datos. La estructura de cada expresión está basada en un conjunto de tres elementos: un sujeto, un predicado, y un objeto. A su vez, un conjunto de estas ternas, forman un Gráfico RDF. La notación de estos elementos suele ser mediante dos nodos unidos por una flecha, que siempre apunta al objeto.

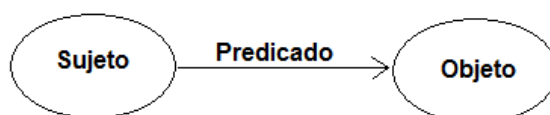


Figura 4. Representación Gráfica de expresiones en RDF

La afirmación de una terna implica la afirmación de la relación expresada por el predicado, entre el sujeto y el objeto. Un gráfico RDF expresa la conjunción lógica de todas las expresiones que contiene.

Un ejemplo de expresión sería “El análisis es una fase de desarrollo software”. En este caso, el sujeto sería “El análisis”, el predicado “es una”, y el objeto “fase de desarrollo software”. Para representar todas las fases de desarrollo software, necesitaríamos una terna por cada una de ellas. La conjunción de todas, nos expresaría la información que queremos representar.

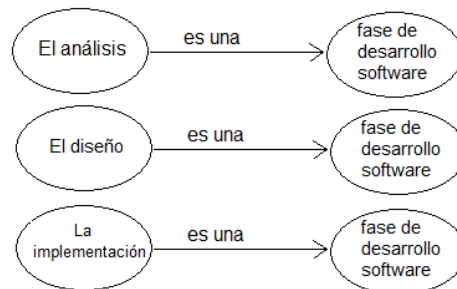


Figura 5. Ejemplo de gráfico RDF

Cada objeto puede ser a su vez el sujeto de otros predicados, o un “literal”. Siguiendo con nuestro ejemplo, podríamos tener la expresión “Las fases de desarrollo software producen documentos”; en la que “Fase de desarrollo software” pasaría a ser el sujeto. Si construyéramos esa expresión, y la combináramos con las anteriores, podríamos ver que “El análisis”, es una “fase de desarrollo software”, y por el hecho de serlo, “produce documentos”. Podríamos seguir ampliando los nodos del gráfico RDF incluyendo más expresiones, hasta conformar un vocabulario completo.

Como seguramente habrá apreciado el lector, existirían otras formas diferentes para la representación de la información que se han usado; por ejemplo, modificando las expresiones de forma que el objeto pase a ser el sujeto. Esto nos muestra que no existe una forma determinista de representar mediante RDF un vocabulario. Una mala elección de sujetos, predicados, u objetos, puede repercutir en que una representación no sea válida para todos los contextos. A la hora de elegir una representación RDF de un dominio, hemos de valorar las diferentes opciones a nuestro alcance, y usar la que nos resulte más provechosa.

Para un ser humano, la comprensión del sujeto “El análisis”, o simplemente del predicado “es una”, no reviste mayor dificultad; sin embargo, una máquina necesita que se le explique qué es cada elemento. RDF resuelve esta cuestión a través de las referencias URI. Una referencia URI, o literal, usada como nodo, identifica qué representa el nodo. Una URI usada en un predicado representa la relación existente entre los elementos representados por los nodos que conecta. Un predicado URI podría ser a su vez un nodo en el gráfico. Cada URI identifica un recurso expresado por RDF.

En RDF se usan tipos (datatypes) para definir los posibles valores que puede tomar un nodo. Un tipo de datos consta de un espacio léxico, un espacio de valores, y un mapeo léxico-valor. Esto posibilita el referirse a un mismo elemento de diferentes formas. Según la especificación del W3C [26], la definición de tipos en RDF es compatible con los tipos en los XML Schemas. Cada elemento del espacio de valores puede estar relacionado con de 0 a N elementos del espacio léxico, mientras que cada elemento del espacio léxico está relacionado unívocamente con un elemento del espacio de valores.

La forma más clara de verlo es a través de un ejemplo: si quisiéramos representar los números naturales 1, 2 y 3 en binario y en decimal, nuestro tipo de datos podría ser el que se muestra bajo estas líneas. Se ha forzado a dos bits para diferenciar la representación binaria de decimal en el 1.

Espacio de valores	{UNO, DOS, TRES}
Espacio léxico	{"1", "2", "3", "4", "01", "10", "11"}
Mapeo	{<"1", UNO>, <"2", DOS>, <"3", TRES>, <"01", UNO>, <"10", DOS>, <"11", TRES>}

Tabla 2. Ejemplo de Espacio de Valores

Con este tipo de datos, decimos que cada vez que usemos DOS, podremos estar hablando tanto del número decimal "2" como del número binario "10".

Los tipos de datos se definen de forma separada en RDF, y son referenciados en los grafos mediante URIs.

Para identificar como parte del léxico elementos como números o palabras, se usan literales. Cualquier elemento representado por un literal, puede ser a su vez representado por una URI, aunque la especificación del W3C [26] recomienda el uso de literales por ser a menudo más conveniente e intuitivo. Un literal puede ser el objeto de una expresión, pero nunca el sujeto o predicado.

Las ideas sobre el significado e inferencia en RDF se sostienen sobre el concepto formal de vinculación (entailment), tal y como se discute en otras recomendaciones del W3C [27]. Una expresión RDF vincula a otra expresión RDF si todas las posibles combinaciones de cosas que hacen a la primera verdadera, lo hacen también a la segunda. Sobre esta base, si la verdad de la primera es presumida o demostrada, esto puede transmitirse a la segunda.

Esta sección no pretende ser un manual de RDF, de modo que no se analizarán las vicisitudes de la sintaxis de este lenguaje. Para obtener información sobre la sintaxis, puede consultarse la recomendación extendidas del W3C sobre la misma [28].

RDF Vocabulary Description Language: RDF Schema.

Como se vio en la sección anterior, RDF es un lenguaje de propósito general para representar información en la Web. La especificación de RDF para definir vocabularios es RDF Schema (RDFS).

Según la descripción que se hace en el W3C [29], RDFS usa un sistema basado en clases y propiedades, similar al de la programación orientada a objetos, con la diferencia de que, en lugar de definir una clase en función de las propiedades que puedan tener sus instancias, RDFS define las propiedades según las clases de recursos a las que pueden aplicarse. Estos son los mecanismos de rol y ámbito. La cuestión es que las propiedades se aglutinan para formar clases, con unos ámbitos de aplicación. Básicamente, RDFS es una extensión semántica de RDF, que mediante recursos definidos con RDF permite la especificación de un vocabulario. Si recordamos los tipos de datos de RDF (datatypes), podemos decir que todas las clases son tipos de datos, mientras que las propiedades son predicados.

Los recursos pueden dividirse en grupos llamados clases, siendo llamados sus miembros instancias. Cada clase es, a su vez, un recurso que define sus propiedades mediante RDF, y suele identificarse con una URI. Para especificar que un recurso es una instancia de una clase, se usa la propiedad **rdf:type**.

Se denomina extensión de la clase al conjunto de instancias de una clase. Por ejemplo, en el ámbito de la Ingeniería del Software, podría existir la clase “Participantes en el Análisis”, cuya extensión podría estar formada (sin entrar en sutilezas) por “Jefe de proyecto”, “Analista” y “Cliente”. Una instancia puede pertenecer a más de una extensión de clase. Continuando con el ejemplo anterior, los elementos citados podrían pertenecer, a excepción de “Cliente”, a la extensión de la clase “Participantes en el Diseño”. Incluso puede darse el caso de que dos clases tengan exactamente la misma extensión de clases. El grupo de instancias de una clase se denota mediante la propiedad **rdfs:Class**.

Una clase puede contener subclases, que contendrán la propiedad **rdfs:subClassOf**. Del mismo modo, podemos usar el término “superclase” para referirnos a la inversa. La pertenencia a una clase es hereditaria de superclase a subclase. A modo de ejemplo, si “Tareas del análisis” es una subclase de “Tareas de un proyecto”, todas las instancias de la primera serán a su vez instancias de la segunda.

Todas las clases descritas por RDF son subclases de la clase **rdfs:Resource**, que a su vez es una instancia de **rdfs:Class**. Otras clases genéricas son **rdfs:Literal** (superclase de valores literales, como cadenas de texto o números), **rdfs:Datatype** (clase a la que pertenecen los tipos), **rdf:XMLLiteral** (clase de los literales con formato XML) y **rdf:Property** (clase de las propiedades). Como podemos observar, todo el esquema de RDF para definir vocabularios, está definido a su vez en mediante RDFS.

A semejanza de las clases, las propiedades también pueden tener subpropiedades, existiendo una herencia en cuanto a los elementos que cumplen una propiedad (“Si una propiedad P es una subpropiedad de la propiedad P’, entonces todos los pares de recursos que están relacionados con P están también relacionados con P’.”). A diferencia de las clases, no se ha definido una superpropiedad suprema de la que todas las propiedades sean subpropiedades.

En el árbol de clases y propiedades que podría formarse mediante todas las jerarquías imaginables, aparecen los ámbitos (ranges) y dominios (domains) para seleccionar exactamente los elementos que queremos que formen parte de un vocabulario.

Para establecer que los valores de una propiedad son instancia de una clase, usamos la propiedad **rdfs:range**. La pertenencia a un ámbito establece que todos los elementos que sean el objeto en una expresión RDF con cierto predicado, pertenecen a la clase especificada.

Por ejemplo, si escribiéramos:

“es revisado por” rdfs:range “Responsables de documento”

Estaríamos diciendo, según este hipotético (e inexacto) vocabulario de Ingeniería del Software, que todos los elementos “a la derecha” (los objetos) de las expresiones RDF que contengan el predicado “es revisado por”, pertenecen a la clase “Responsables de documento”.

Dicho de forma gráfica, si nos encontramos con:

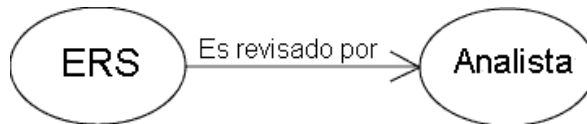


Figura 6. Ejemplo para rango RDFS

Sabríamos que “Analista” es una instancia de la clase “Responsables de documento” gracias a la definición del ámbito anterior.

Los dominios son semejantes a los ámbitos, con la diferencia de que, en vez de indicar la clase a la que pertenece el objeto, están relacionados con el sujeto. Denotamos el dominio mediante la propiedad **rdfs:domain**.

Siguiendo con el ejemplo anterior, podríamos expresar lo siguiente:

“es revisado por” rdfs:domain “Documentos”

Con lo que indicaríamos que todos los elementos que “son revisados por alguien”, pertenecen a la clase “Documentos”. Si se hubiera establecido este dominio, la anterior ilustración, nos estaría diciendo que la ERS (especificación de requisitos software) es una instancia de la clase “Documentos”.

Es importante destacar que RDFS únicamente define las relaciones entre diferentes elementos de un vocabulario, y limitaciones en cuanto a tipos o propiedades, pero no establece cómo deberían ser usadas estas relaciones. Dicho de otra forma, tan sólo da información gramatical, sin entrar en la semántica.

SPARQL Language for RDF.

Siguiendo las estandarizaciones del W3C [30], SPARQL está diseñado para expresar consultas a través de diversas fuentes de datos almacenados como RDF o traducidos a RDF. SPARQL permite consultar patrones requeridos y opcionales a través de conjunciones y disyunciones. Los datos pueden mostrarse como conjuntos de resultados o grafos RDF.

La mayoría de las consultas en SPARQL contienen un conjunto de patrones triples (patrón de grafo básico). Los patrones triples son como las ternas de RDF, exceptuando que cada sujeto, predicado, y objeto, puede ser una variable. Un patrón básico de grafo básico coincide con un subgrafo de los datos RDF cuando los términos RDF de dicho subgrafo pueden ser sustituidos por las variables, y el resultado es un grafo RDF equivalente al subgrafo. Esto significa que si fijamos el sujeto, predicado, objeto, o cualquier combinación de estos elementos, la consulta nos devolverá el subconjunto, o subconjuntos, de datos del grafo que coinciden con ese patrón.

Por ejemplo, si dispusiéramos de unos datos RDF sobre ingeniería del software, en el que hubiéramos definido todos los documentos que produce cada fase de desarrollo, y quisiéramos saber los documentos que produce la fase de “análisis”, la consulta podría ser la siguiente:

```
SELECT ?v WHERE { “análisis” “produce” ?v }
```

Podemos observar la estructura de una consulta típica. Tras la palabra “select”, indicamos los elementos que queremos mostrar en los resultados (en este caso, la

variable *v*). En la cláusula *where*, indicamos las restricciones en este orden: sujeto, predicado, objeto. Suponiendo que el grafo contenga ternas del tipo fase – produce – documento, si fijamos la fase a “análisis”, y el predicado a “produce”, tan sólo dejamos variable el objeto (el documento producido), de modo que los resultados obtenidos corresponderán exactamente con los documentos producidos durante la fase de análisis. Del mismo modo, podríamos consultar qué fase de desarrollo produce el documento que se llame “ERS” con esta consulta:

```
SELECT ?v WHERE { ?v “produce” “ERS” }
```

O podríamos consultar todos los documentos producidos por todas las fases:

```
SELECT ?x ?y WHERE { ?x “produce” ?y }
```

De esta forma, podemos probar a fijar todos los elementos que necesitemos para obtener los resultados deseados. Se permite no fijar ningún elemento, aunque técnicamente no tendría sentido consultar “todo lo que está relacionado de alguna forma con todo”.

Con total seguridad, no nos conformaremos con que las búsquedas se basen en coincidencias exactas en los literales. Por este motivo, SPARQL permite el uso de expresiones regulares a través de los filtros. Para ver un ejemplo, supongamos que nuestro lenguaje de Ingeniería del Software incluye una definición de los roles existentes en un proyecto, siendo cada expresión de la forma (nombre del rol) (gestiona) (ámbito). Ahora, supongamos que queremos saber qué ámbitos gestionan cada uno de los roles que comienzan por la palabra “jefe”. La consulta sería la siguiente:

```
SELECT ?rol WHERE { ?rol “gestiona” ?ambito  
  
                FILTER regex (?rol, “^jefe”)  
}
```

La sintaxis de las expresiones regulares es la misma que se usa para XQuery y XPath, y está basada en las XML Schema Regular Expressions. Por supuesto, podemos especificar en qué lenguaje ha de consultarse un literal, por ejemplo, podemos especificar que hablamos de “phase” en el lenguaje ois (una hipotética Ontología de Ingeniería del Software) codificándolo de la siguiente manera: “phase”@ois.

Del mismo modo que tenemos filtros para cadenas de texto, también existen para campos numéricos. Por ejemplo, si queremos filtrar una variable *?v*, podríamos usar en el *WHERE* el filtro *FILTER (?v > 10000)*.

Si quisiéramos que la salida, en vez de mostrar un conjunto de resultados, nos devuelva un grafo RDF, tan sólo tenemos que sustituir la palabra *SELECT* por *CONSTRUCT*. También es posible que la salida sea un grafo RDF descriptivo de la solución; en este caso habría que usar *DESCRIBE*. Si tan sólo queremos saber si existen resultados a una consulta, usamos *ASK*.

Otra de las ventajas de SPARQL es que permite que las consultas incluyan patrones o datos opcionales. Esto permite obtener datos en el resultado aunque la expresión RDF no tenga disponible este dato; de otro modo, los datos se descartarían para el resultado.

Si se usan patrones opcionales, los datos se mapearán a un grafo RDF siempre que estén disponibles, si no lo estuvieran, no llegará a hacerse el binding, pero no se descartarán como parte de los resultados.

Continuando con el ejemplo de los roles en un proyecto de ingeniería del software, podríamos querer obtener información sobre los roles que participan en la implementación, y a quién supervisa cada uno. Si no usáramos patrones opcionales, los becarios no aparecerían en los resultados al no supervisar a nadie. La forma de hacer la consulta sería la siguiente:

```
SELECT ?rol ?supervisado  
  
WHERE {    ?rol "participa en" "implementacion"  
  
OPTIONAL { ?rol "supervisa" ?supervisado} }
```

Por supuesto, podemos incluir tantas cláusulas opcionales como queramos, y filtros de igual modo que en unas restricciones normales.

A parte de restricciones en la consulta, podemos especificar la forma en que queremos que se muestren los resultados. Las existentes son las siguientes:

- **ORDER BY:** permite que los resultados sigan un orden basado en el valor de una o varias variable. Se puede especificar si la ordenación será ascendente o descendente (*ASC(?nombre_variable)* y *DESC(?nombre_variable)*) aunque por defecto lo hace de forma ascendente.
- **DISTINCT:** elimina los resultados repetidos (*SELECT DISTINCT ?nombre_variable... WHERE...*)
- **REDUCED:** elimina algunos de los resultados repetidos (*SELECT REDUCED ?nombre_variable... WHERE...*)
- **LIMIT:** reduce el número de resultados mostrados a los indicados por el límite (*SELECT ... WHERE {...} LIMIT i*)
- **OFFSET:** los resultados se mostrarán a partir del i-ésimo, que no será incluido (*SELECT ... WHERE {...} OFFSET i*).

Todas las restricciones de resultados pueden combinarse entre sí. Por ejemplo, para saber cuál es el tercer rol de Ingeniería del Software que participa en el análisis, ordenado alfabéticamente por el nombre, la consulta sería la siguiente:

```
SELECT ?rol WHERE {?rol "participa en" "análisis"}  
  
ORDER BY ?rol OFFSET 2 LIMIT 1
```

Existen muchas más consideraciones y particularidades sobre SPARQL, pero quedan fuera del ámbito de este estudio sobre la Web Semántica. Para más información, pueden consultarse las referencias.

Web Ontology Language (OWL).

Haciendo referencia nuevamente a los estándares del W3C [31], “El Lenguaje de Ontologías Web (OWL) está diseñado para ser usado en aplicaciones que necesitan procesar el contenido de la información en lugar de únicamente representar información para los humanos”. Si bien RDF era el lenguaje que permitía la descripción de recursos, OWL permite la definición de vocabularios complejos sobre ámbitos concretos (ontologías).

Antes de definir el lenguaje de ontologías, deberíamos puntualizar qué es en sí una ontología. En las preguntas frecuentes de OWL del W3C [32], podemos encontrar la siguiente definición: “Una ontología define los términos a utilizar para describir y representar un área de conocimiento. Las ontologías son utilizadas por las personas, las bases de datos, y las aplicaciones que necesitan compartir un dominio de información (un dominio es simplemente un área de temática específica o un área de conocimiento, tales como medicina, fabricación de herramientas, bienes inmuebles, reparación automovilística, gestión financiera, etc.). Las ontologías incluyen definiciones de conceptos básicos del dominio, y las relaciones entre ellos, que son útiles para los ordenadores [...]. Codifican el conocimiento de un dominio y también el conocimiento que extiende los dominios. En este sentido, hacen el conocimiento reutilizable”.

De este modo, OWL se encarga de enfocar los recursos definidos mediante RDF hacia la especificación completa de un vocabulario sobre un dominio específico.

Anteriormente, existían otros lenguajes de ontologías, pero eran desarrollados con objetivos muy concretos, alejados de la estandarización. El uso de RDF orientado a la especificación de ontologías permitió que las ontologías dispusieran de las siguientes capacidades:

- Capacidad de ser distribuidas a través de varios sistemas
- Escalable a las necesidades de la Web
- Compatible con los estándares Web de accesibilidad e internacionalización
- Abierto y extensible

Cuando se analizó la importancia de RDF en el apartado dedicado a dicho lenguaje, pudo parecer que bastaba para la definición de vocabularios; sin embargo, “OWL extiende RDFS para permitir la expresión de relaciones complejas entre diferentes clases RDFS, y mayor precisión en las restricciones de clases y propiedades específicas. Esto incluye por ejemplo: - los recursos para limitar las propiedades de clases con respecto a número y tipo, - los recursos para inferir qué elementos que tienen varias propiedades son miembros de una clase en particular - los recursos para determinar si todos los miembros de una clase tendrán una propiedad en particular, o si puede ser que sólo algunos la tengan - los recursos para distinguir entre relaciones uno-a-uno, varios-a-uno o uno-a-varios, permitiendo que las "claves externas" de las bases de datos puedan representarse en una ontología - los recursos para expresar relaciones entre clases definidas en diferentes documentos en la Web - los recursos para construir nuevas clases a partir de uniones, intersecciones y complementos de otras, y - los recursos para restringir rangos y dominios para especificar combinaciones de clases y

propiedades”. En resumen, RDF es un lenguaje que se limita a exponer datos, usando una capacidad de expresión semántica muy reducida.

Además, si nos guiamos por el W3C [31]: “La Web semántica se basará en la capacidad de XML para definir esquemas de etiquetas a medida y en la aproximación flexible de RDF para representar datos. El primer nivel requerido por encima de RDF para la Web semántica es un lenguaje de ontologías que pueda describir formalmente el significado de la terminología usada en los documentos Web. Si se espera que las máquinas hagan tareas útiles de razonamiento sobre estos documentos, el lenguaje debe ir más allá de las semánticas básicas del RDF Schema”. Esto quiere decir que, si pensásemos en las herramientas de Web Semántica como una jerarquía de capas, RDFS se limitaría a exponer recursos, que son usados por OWL para dotarles de una semántica completa.

Una vez descrito lo que es una ontología, su utilidad, y la necesidad de crear un lenguaje estándar para definirla, nos centraremos en el análisis de OWL.

OWL proporciona tres lenguajes, cada uno con nivel de expresividad mayor que el anterior, diseñados para ser usados por comunidades específicas de desarrolladores y usuarios. Estos lenguajes son OWL Lite, OWL DL y OWL Full. Sus principales diferencias radican en las restricciones que aplican a su vez sobre la especificación de OWL.

OWL Lite está pensado para ontologías con restricciones y clasificaciones jerárquicas simples. Al no ofrecer mucha complejidad, las herramientas que pueden procesar estos lenguajes son a su vez simples.

OWL DL (Description Logics) busca expresar mayor complejidad descriptiva sin ganar en complejidad computacional. Todas las conclusiones con computables y resolubles en tiempo finito.

OWL Full permite expresar relaciones más abstractas y mayor libertad sintáctica. La especificación del W3C [31] nos indica que: “Es poco probable que cualquier software de razonamiento sea capaz de obtener un razonamiento completo para cada característica de OWL Full”.

Los lenguajes que pertenecen a cada rama de OWL están incluidos en la mayor, de este modo, un lenguaje de OWL Lite es a su vez OWL DL, y por lo tanto, también OWL Full. Las relaciones entre los distintos “dialectos” se expresan en el siguiente diagrama.

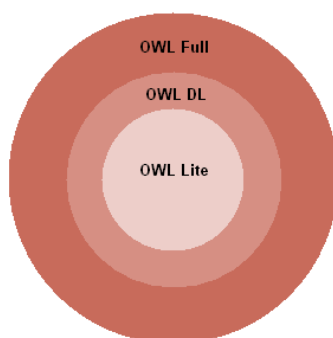


Figura 7. Relaciones de los lenguajes OWL

De las descripciones de los lenguajes OWL, podemos deducir que debe existir un compromiso entre la riqueza de expresión que queramos dar a una ontología, y la capacidad de ser procesada que deseemos para ella. Podríamos hacer un símil con los distintos idiomas del lenguaje natural. Cuando más compleja es la gramática de una lengua, y más dada a los giros, nos permite una riqueza de expresión mayor, pero es más difícil de aprender para un hablante no nativo (y no digamos para ser comprendida por una máquina).

A diferencia del resto de tecnologías investigadas previamente, no se hará hincapié en la sintaxis y diferentes propiedades de OWL al no considerarse necesario para la comprensión de su utilidad y funcionamiento. Tan sólo es necesario tener en cuenta que extiende propiedades de RDFS, permitiendo la expresión de cardinalidades, además de relaciones lógicas entre propiedades (inversas, simétricas...). Por otro lado, permite el empleo de restricciones numerales (para todo, existe uno...), y operadores lógicos entre clases, tales como uniones o intersecciones. Para más información, se remite puede consultarse los estándares anteriormente descritos [31].

2.2.2. Ontología para la Ingeniería del Software (SEOntology).

A lo largo del análisis de las diferentes tecnologías implicadas en la Web Semántica, se ha podido apreciar que todos los ejemplos iban encaminados al dominio de la Ingeniería del Software. Esto se debe a que será el dominio que se tomará como base para el desarrollo del Servidor on-line de Exámenes.

Una ontología de Ingeniería del Software típicamente provee conceptos de dicho dominio (qué son, cómo están relacionados, y cómo pueden relacionarse con otros) para representar y comunicar el conocimiento de Ingeniería del Software, e información de proyectos, a través de Internet.

Una buena práctica sería el desarrollo de una nueva ontología para la Ingeniería del Software, pero se trataría de un desarrollo complejo que formaría un proyecto completo por sí mismo. Se ha decidido basarse en una ontología ya desarrollada. En este apartado se analizarán los aspectos básicos de la Software Engineering Ontology (SEOntology) [33], la primera ontología de Ingeniería del Software cuya implementación está disponible on-line.

SEOntology nació con el objetivo de proporcionar un lenguaje común para la compartición de conocimiento entre diferentes núcleos de un equipo de desarrollo involucrado en un proyecto de Ingeniería del Software.

La necesidad de este lenguaje común radica en la tendencia actual de la diversificación de entornos durante el ciclo de desarrollo de los proyectos. No es raro que una empresa se dedique al análisis y diseño, y se encargue la implementación, o las pruebas, a una empresa externa, que no tiene por qué estar geográficamente próxima a la empresa inicial.

Por otro lado, SEOntology pretende que los conceptos teóricos de la Ingeniería del Software sean comprendidos y seguidos, ya que no es raro que cada empresa (incluso cada ingeniero del software) los a sus necesidades o costumbres, provocándose inconsistencias en la documentación [34].

De forma esquemática, las razones para el desarrollo de SEOntology son:

- Definir los conceptos de Ingeniería del Software de una forma más precisa, permitiendo un mejor procesamiento automático tal y como han evolucionado.
- Hacer explícitos los supuestos en el dominio de la Ingeniería del Software.
- Separar el conocimiento del dominio y subdominio del conocimiento de la instancia.
- Crear una referencia multi-site de desarrollo de software para las aplicaciones.
- Compartir una comprensión consistente del significado de la información de un proyecto de ingeniería.

El proyecto SEOntology no se ha limitado a la creación de la ontología en sí, sino que ha desarrollado una serie de aplicaciones que permiten la gestión de proyectos en entornos colaborativos basándose en la propia ontología [33].

El conjunto completo de conceptos de Ingeniería del Software se capturan como un dominio de conocimiento. Obviamente, no todos los proyectos usarán todos los conceptos reflejados en la ontología, sino que cada uno de ellos estará más cerca de uno de los subdominios (por ejemplo, la metodología de orientación a objetos). Los conceptos específicos del conocimiento se expresan como subdominios.

Cada instancia de conocimiento en SEOntology representa las particularidades y necesidades específicas de un proyecto. En este caso, los creadores han decidido separar completamente el dominio de conocimiento de las instancias de conocimiento [35]. Esta separación permite a su vez delimitar por un lado el conocimiento teórico (conceptos, relaciones, subdominios, etc), y por otro los datos de proyectos, permitiendo diferentes instanciaciones en función del proyecto con el que se relacionen. Por ejemplo, pertenecería al dominio el concepto “Use_case”, y una de sus instancias sería cada uno de los casos de uso presentes en un proyecto.

Se han publicado dos niveles de abstracción: el genérico y el específico. El genérico representa los conceptos que son comunes al conjunto de conceptos de Ingeniería del Software, mientras que el específico representa el conjunto de conceptos que son específicamente usados para algunas categorías de proyectos particulares [36].

A través de SEOntology, un texto sobre Ingeniería del Software puede ser parseado para descubrir de forma dinámica y automática los conceptos relevantes y la relación entre ellos. Si además se especifican las instancias presentes en ese texto, se puede representar el mismo de forma esquemática, a través de los conceptos y relaciones de la ontología, para asegurar su comprensión y validez semántica [36].

Fijando la vista en el caso particular del Servidor on-line de Exámenes, podemos deducir que, gracias a SEOntology, podrá optimizarse el acceso a los recursos publicados, ya que la indexación podría seguir una estructura semejante a la ontología. Al mismo tiempo, podrá validarse automáticamente la relevancia de los contenidos publicables respecto al dominio de la Ingeniería del Software.

2.3. Web Social

La Web Social, también llamada Web 2.0, es según Christian Van Der Hest S. [37]: “la transición que se ha dado de aplicaciones tradicionales hacia aplicaciones que funcionan a través del web enfocadas al usuario final. Se trata de aplicaciones que generen colaboración y de servicios que reemplacen las aplicaciones de escritorio”. En otras palabras, se trata del conjunto de aplicaciones Web en las que la experiencia del usuario final cobra protagonismo. También añade que “El Web 2.0 no es precisamente una tecnología, sino es la actitud con la que debemos trabajar para desarrollar en Internet”. Esto nos indica que puede ser complicado definir qué es exactamente la Web Social.

Tim O'Reilly, autor del término “Web 2.0”, nos dice en su artículo original al respecto [38] que la Web 2.0 no tiene una clara frontera, sino un núcleo gravitacional. Puede verse la Web 2.0 como un conjunto de principios y prácticas alrededor del cual giran las aplicaciones, a mayor o menor distancia.

No es sencillo definir a la Web Social, al menos en aspectos prácticos. A veces, es complicado definir cuándo un sitio queda encuadrado dentro de la “Web 1.0” y cuándo puede considerarse miembro de pleno derecho de la Web Social. Si nos basamos en el artículo escrito por Paul Graham [39], podemos llegar a la conclusión de que hay tres tendencias fundamentales en los sitios que consideramos de Web 2.0: “Ajax, democracia, y no tratar mal a los usuarios”.

Con el término Ajax se hace referencia al hecho de hacer posible que las páginas Web se parezcan cada vez más a aplicaciones de escritorio. Considera que se está produciendo una generación nueva de software, de las que no había desde que aparecieron los primeros microcomputadores.

Graham hace gran hincapié en la democratización de la información que supone la Web 2.0. Dice que ha quedado demostrado que el trabajo de los aficionados puede superar el de los profesionales, cuando disponen del sistema adecuado para canalizar sus esfuerzos. El autor del artículo nos hace ver cómo un filtro de calidad compuesto por usuarios masivos puede ser mucho más eficiente que el filtro que pueda imponer un editor profesional. Por supuesto, no hay que olvidar que todos los contenidos publicados en la Web Social tienen mayor o menor calidad, pero aquellos que realmente valen la pena son los que prevalecen con el tiempo.

Siguiendo con el artículo de Graham, llama la atención la característica de “no tratar mal a los usuarios” a la que el autor dedica una sección completa del artículo. Compara los sitios que se crearon durante la burbuja tecnológica con los actuales sitios de Web 2.0. Las de aquel entonces tenían, según el autor, diseños insultantes, y transmitían el mensaje de “Esta es nuestra Web, no la tuya”. Graham cree que la raíz del problema es que las compañías de aquel entonces tenían la sensación de estar dando algo gratis, lo que les hacía comportarse de forma arrogante. Podían llegar a pensar que cuanto más daño se hiciera al usuario (en forma de anuncios, registros, spam, etc), mejor. La Web 2.0 está cambiando todo eso. Las Webs ya no son de las compañías, sino de los propios usuarios, quienes generan los contenidos. El autor da una serie de consejos para todas las “startups” relacionadas con Web Social que aparezcan:

- No ser nunca despótico.

- No hacer que los usuarios se registren a menos que sea necesario.
- No pedir la dirección a menos que haga falta por alguna razón.
- No hacer preguntas innecesarias.
- No enviar e-mails a menos que los usuarios los pidan explícitamente.
- No abrir los links en frames o en otras ventanas.
- Si hay versión gratuita y de pago, no ser muy restrictivo.
- Pecar de generosidad en cuanto a lo que se permite a los usuarios.

Graham resume su opinión en que Web 2.0 significa usar la Web en la forma en que debe ser usada. Las tendencias que vemos ahora no son más que la naturaleza inherente a la Web, emergiendo de los viejos modelos que se impusieron durante la Burbuja.

Si volvemos al artículo de O'Reilly [38], vemos como el padre del término Web 2.0 insiste en la importancia de los datos, o mejor dicho, del control de los mismos. En una plataforma en la que los usuarios generan datos que otros usuarios consumen, ¿Quién es el dueño de los mismos? O'Reilly prevé que a medida que las empresas comiencen a darse cuenta de que el control sobre datos puede ser su principal fuente de ventaja competitiva, podremos encontrar mayores intentos de control. También nos dice que el control de una fuente de datos única y difícil de replicar, que se enriquezca con el uso de los usuarios, puede ser el secreto del éxito corporativo en los nuevos tiempos.

O'Reilly da dos directrices para las empresas que quieran embarcarse en proyectos de Web Social:

- Las operaciones deben convertirse en una competencia central de la compañía (core competence). Esto significa que, si bien el mantenimiento ya era algo importante, ahora debe de ser lo más frecuente posible, llegando incluso a automatizarlo.
- Los usuarios deben ser tratados como codesarrolladores. Ya sea liberando el código, teniendo en cuenta la importancia de sus aportaciones de contenidos, o simplemente permitiendo un desarrollo abierto, en el que cualquiera pueda aprovechar nuestros servicios en sus propios sitios (caso de Google Maps, cuyos mapas son usados como base en otros sistemas).

Los análisis previos hacen bastante obvia la segunda afirmación, pero la primera es sorprendente. El autor presenta el ejemplo de Google, al igual que hace en la mayoría de sus postulados. Si Google no rastreara continuamente la Web para actualizar sus índices, ni buscara sin descanso formas de evitar los trucos ilegales de posicionamiento, no habría llegado a sus niveles de calidad actuales. Como bien dice el autor, posiblemente su topología de red, sistema de administración, y algoritmos de balanceo, sean secretos mejor guardados que los propios algoritmos de búsqueda.

Una vez descrito lo que es el concepto de Web Social en sí, y las implicaciones empresariales que supone, podemos adentrarnos en las diferentes tecnologías, cuya existencia, unida a la evolución de la Web como si fuera un ser vivo, han desencadenado el paso a la Web 2.0.

2.3.1. Tecnologías de Web Social

A partir del artículo de O'Reilly [38], se puede deducir que las tecnologías que suelen estar presentes en los proyectos de Web Social son:

- Ajax
- Sindicación (RSS)
- Orientación a servicios

Conviene aclarar que la presencia de todas estas tecnologías en cierto sistema no implica necesariamente que pertenezca a la Web Social. Es más, no es necesario que todas estén presentes, ya que como dice O'Reilly, “la Web 2.0 no es una tecnología, sino una actitud”. Veamos a mayor profundidad cada una de estas tecnologías, o conjunto de ellas.

Ajax

Según publica Jesse James Garret en su artículo “Ajax: A new approach to Web Applications” [40]: “Ajax no es una tecnología. Es realmente muchas tecnologías, cada una floreciendo por su propio mérito, uniéndose de poderosas nuevas formas”.

Ajax es el acrónimo de Asynchronous JavaScript + XML. En términos generales, consiste en enviar una petición directamente al servidor a través de JavaScript, pudiendo tratar la respuesta para cargar tan sólo partes de una página, sin necesidad de cargarla entera. El problema de la actualización parcial de contenidos ha sido un lastre para los programadores de aplicaciones Web. Ajax ha resuelto este escollo que separaba a las aplicaciones Web de los programadores de aplicaciones de escritorio mediante peticiones asíncronas.

Volviendo al artículo de Garret [40], vemos que forma esquemática lo que ofrece de nuevo Ajax:

- Presentación basada en estándares usando XHTML y CSS
- Exhibición e interacción dinámicas usando el Document Object Model.
- Intercambio y manipulación de datos usando XML y XSLT
- Recuperación de datos asíncrona usando XMLHttpRequest
- y JavaScript uniendo todo.

En lugar de cargar un pagina Web, al inicio de la sesión, el navegador carga el motor Ajax (escrito en JavaScript y habitualmente metido en un frame oculto). Este motor es el responsable de renderizar la interfaz que el usuario ve, y comunicarse con el servidor en nombre del usuario.

El motor AJAX permite que la interacción del usuario con la aplicación suceda asincrónicamente (independientemente de la comunicación con el servidor). Así el usuario nunca estará mirando una ventana en blanco del navegador a que el servidor haga algo.

Cada acción del usuario que normalmente generaría una petición HTTP se transforma en una llamada JavaScript al motor Ajax. Cualquier respuesta a una acción de usuario que no necesite un viaje de vuelta al servidor (como una simple validación de datos, editar datos en memoria, e incluso alguna navegación) es manejada por el motor por su cuenta. Si el motor necesita algo del servidor para responder, hace esa petición asincrónicamente, normalmente mediante XML de forma transparente al usuario.

Como se adelantó unos párrafos más atrás, Ajax consigue que las aplicaciones Web se parezcan cada vez más a las aplicaciones clásicas de escritorio, y puedan funcionar como tales. Si la Web Social pretende crear un entorno amigable para los usuarios, es decir, que sientan que “están en casa”, Ajax puede ser el primer paso para conseguirlo.

Sindicación (RSS)

La sindicación (o más correctamente “redifusión”) de información en Web consiste en la transmisión de información desde una fuente de la misma hasta otros entornos, que se convierten a su vez en emisores al ofrecerla a más usuarios.

Los dos formatos que se usan más frecuentemente para la sindicación Web son RSS y Atom. En este documento se describirá RSS por ser el único que ha sido especialmente referenciado en los documentos leídos hasta el momento.

Según Mark Pilgrim [41]: RSS es un formato para syndicar noticias y el contenido de sitios que las publiquen. Estos sitios pueden ser muy diferentes, desde grandes agencias de noticias, hasta blogs. Puede sindicarse cualquier cosa que pueda ser dividida en elementos textuales discretos. Una vez que la información de cada elemento está en formato RSS, un lector de RSS puede comprobar los cambios en el feed RSS del modo adecuado, pudiéndose publicar la información en multitud de sitios heterogéneos de forma simultánea.

Existen varias versiones de RSS, aunque las más frecuentes, hoy día, son la 1.0 y la 2.0. La 1.0 está basada en RDF, mientras que la 2.0 es un formato por sí mismo, tan sólo basado en XML. Bajo estas líneas podemos ver una tabla comparativa de las versiones de RSS extraída del artículo de Mark Pilgrim [41]:

Versión	Propietario	Ventajas	Estatus	Recomendación
0.90	Netscape		Obsoleto por 1.0	No usar
0.91	UserLand	Muy simple.	Oficialmente obsoleta por 2.0, pero todavía bastante popular.	Usar para sindicación básica. Fácil de migrar a 2.0 si se necesita más flexibilidad.
0.92, 0.93, 0.94	UserLand	Permite metadatos más ricos que 0.91	Obsoleta por 2.0	Usar 2.0 en su lugar
1.0	RSS-DEV Working Group	Extensibilidad basada en RDF vía módulos, no controlada por una única distribuidora.	Núcleo estable, desarrollo activo de módulos.	Usar para aplicaciones basadas en RDF o si necesitas módulos avanzados específicos de RDF.
2.0	UserLand	Extensibilidad vía módulos, ruta fácil de migración desde 0.9x	Núcleo estable, desarrollo activo de módulos.	Usar para sindicación rica en metadatos de propósito general.

Tabla 3. Comparativa de versiones RSS y recomendaciones

De esta tabla deducimos que hoy en día conviene centrarse en generar sindicación para las versiones de RSS 1.0 y 2.0, por ser las que se usan de forma más generalizada. Aunque la 0.91 siga usándose, el estar “oficialmente obsoleta” en el momento de la redacción del artículo fuente, le da pocas garantías de futuro.

La sindicación ha permitido que cualquier usuario sea avisado de actualizaciones en los sitios Web que elija sin necesidad de visitarlos uno a uno. Según O’Reilly [38], es el avance más significativo en la evolución desde la arquitectura de la Web 1.0 a la 2.0.

Orientación a servicios

La “Guía breve de los Servicios Web” del W3C [42], define los Servicios Web como “conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web”. En la misma fuente, encontramos que “[...] proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar”.

En el paradigma clásico, o la Web 1.0, las aplicaciones eran entes independientes completos, donde las interacciones con otras eran fruto de un diseño que lo buscara expresamente. Las arquitecturas de cada una eran herméticas hacia el exterior, y el usuario final sólo podía aprovechar lo que ofreciera cada una, y de la forma en que lo ofreciera.

La Web 2.0 busca la reutilización de elementos ya existentes (servicios). Si una compañía ya proporciona unos datos, o un tratamiento efectivo de los mismos ¿Por qué volver a desarrollarlo? La reutilización siempre ha sido uno de los pilares del desarrollo, pero solía reducirse al ámbito de una misma compañía o equipo. La orientación a servicios es mucho más que desarrollar un elemento software para que otros lo añadan;

es dejar abiertas partes de nuestro sistemas para que otros las añadan al suyo, y consigan una innovación beneficiosa mediante el ensamblado de servicios Web.

O'Reilly habla del ensamblado de servicios en estos términos [38]:

“Cuando los componentes de tipo materia prima son abundantes, se puede crear valor simplemente ensamblándolas de forma novedosa o eficaz. Al igual que la revolución del PC proporcionó muchas oportunidades para la innovación en el ensamblado del hardware de tipo *commodity* (con compañías como Dell, que hizo de dicho ensamblado una ciencia, derrotando a compañías cuyo modelo de negocio requería de innovación en el desarrollo del producto), creemos que la Web 2.0 ofrecerá oportunidades a las empresas para superar a la competencia, siendo mejores en el aprovechamiento y la integración de los servicios proporcionados por otros”.

Más adelante añade: “Las aplicaciones Web 2.0 se construyen a partir de una red de servicios de datos que cooperan. Por tanto: ofrezca interfaces de Web Services y sindicación de contenidos, y reutilice los servicios de datos de otros. Apoye los modelos de programación ligeros que permitan sistemas débilmente acoplados”.

En otras palabras, O'Reilly aconseja que las arquitecturas de las aplicaciones sean muy desacopladas, y que se permita que otros sistemas puedan aprovechar nuestros servicios de una forma sencilla, a la vez que nosotros aprovechamos los servicios que brindan otros.

La cuestión clave viene definida en cuanto a qué beneficio nos puede brindar el que nuestros servicios sean aprovechados por otros (y más si es de forma gratuita). En este punto, es necesario volver a citar a O'Reilly [38] cuando dijo que “el control de una fuente de datos única y difícil de replicar, que se enriquezca con el uso de los usuarios, puede ser el secreto del éxito corporativo en los nuevos tiempos”. Si nosotros ofrecemos un servicio que usan otras personas; aunque no nos dé un beneficio directo, nos da el control del mismo; este control puede suponer una ventaja estratégica importante.

Los dos tipos de Servicios Web más extendidos hoy día son:

- Servicios Ligeros (REST)
- Servicios basados en SOAP (“Web Services”)

Los servicios REST son cronológicamente los primeros. Se basan en un mero intercambio de mensajes HTTP con un servidor, sin seguir un estándar específico. Sus características principales es que se usan en comunicaciones cliente-servidor y que no mantienen estado; es decir, cada petición es independiente de otras.

Los servicios basados en SOAP son a los que se refiere la gente comúnmente cuando habla de *Web Services* o Servicios Web. Las características funcionales, en cuanto a comunicaciones cliente-servidor, y falta de estado, son similares a los servicios REST, con la aportación de un estándar para los mensajes y la implementación de los mismos.

Se podría pensar que los servicios REST son adecuados para comunicaciones que no tengan valor empresarial, y los *Web Services* para aquellos que precisen de mayor seguridad o control del flujo de información.

2.3.2. Ejemplos de aplicaciones de Web Social

Ahora que sabemos lo que es la Web Social, o Web 2.0, y que tenemos nociones de las tecnologías más habituales que emplean, vamos a ver algunos ejemplos de aplicaciones que podrían considerarse que pertenecen a este género.

Analizaremos algunos de los buques insignia de la Web Social: Wikipedia y Youtube, además de un sitio Web con un propósito más o menos cercano al del Servidor online de Exámenes, con mucha experiencia y cuyo nombre habla por sí solo: “El rincón del vago”.

De cada ejemplo estudiado, se hablará sobre su aspecto general, el nivel de colaboración del usuario y las tecnologías que usa. Se añadirá el ranking que les da Alexa y su pagerank de Google como indicativo de su popularidad.

Wikipedia, la enciclopedia libre.

La Wikipedia se define a sí misma como “[...] una enciclopedia libre y políglota basada en la colaboración de sus contribuyentes por medio de la tecnología wiki”.

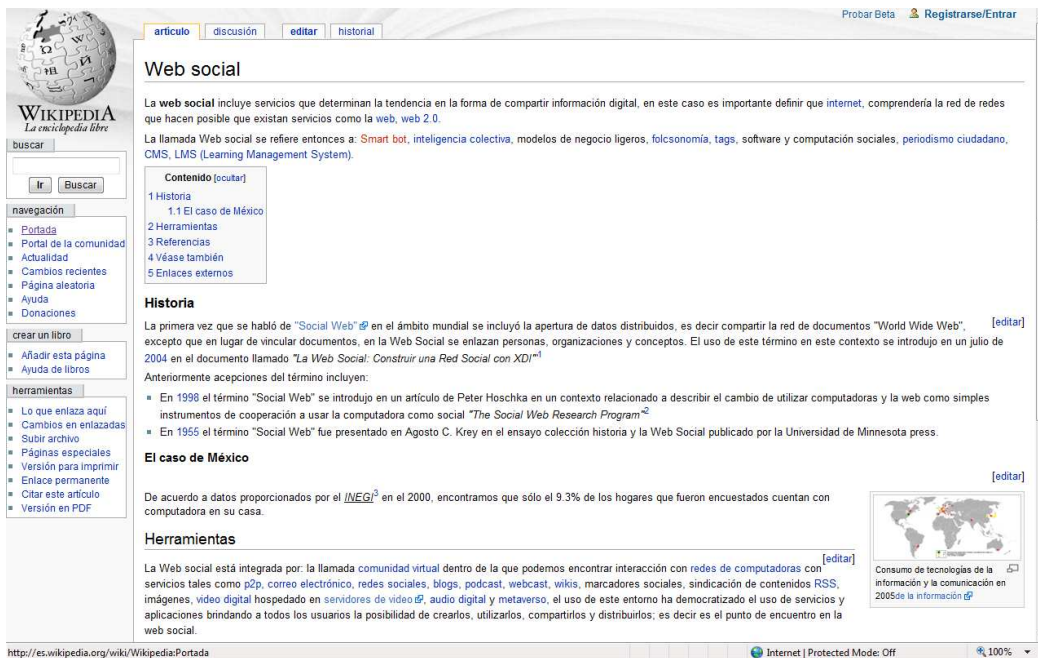


Figura 8. Imagen de la Wikipedia en castellano

Es un proyecto que se puso en marcha durante el año 2001, y cuyo objetivo es construir y mantener una enciclopedia libre gracias a la colaboración de los propios usuarios. Su idiosincrasia le permite estar actualizada casi inmediatamente con todos los acontecimientos que van teniendo lugar, a diferencia de las enciclopedias clásicas, que dependían de un grupo de editores. Por otro lado, estas mismas características hacen que la información que publica no sea siempre fiable.

El aspecto que presenta la aplicación es limpio y amigable. Es sencillo encontrar lo que se busque, y la estrecha interrelación entre todos los artículos permite ir pasando del tema inicial que se buscó a otros relacionados, siguiendo los enlaces. La financiación completa corre a cargo de donaciones, de modo que no existe ningún elemento publicitario más allá del anuncio de las propias donaciones.

Título:	Wikipedia
URL:	http://wikipedia.org
Versión WAP:	Sí
Ranking Alexa:	8
Pagerank Google:	9
Idioma:	Aplicaciones distintas para cada idioma
Código abierto:	Sí. El motor es descargable y se puede usar.
Acceso libre:	Sí
Publicación libre:	Sí
Servicios web:	Búsqueda en portales con misma tecnología
Ajax:	No
Sindicación:	RSS, Atom
Más información:	Estándar propio para publicación de artículos (Wikipedia)

Tabla 4. Análisis de la Wikipedia

Aunque la Wikipedia no use Ajax, ni haga un uso extendido de servicios web, no se puede dudar que pertenece al mundo de la Web Social. Ofrece sindicación, pero está restringida a los historiales de los artículos. Es un claro ejemplo de una aplicación que pertenece a la Web 2.0 no por las tecnologías que usa, sino por su propósito y forma de empleo.

Youtube

Es el portal de videos por excelencia. Un sitio donde cualquiera puede publicar sus videos para que otros usuarios los vean (siempre que no infrinjan las reglas básicas de Youtube).

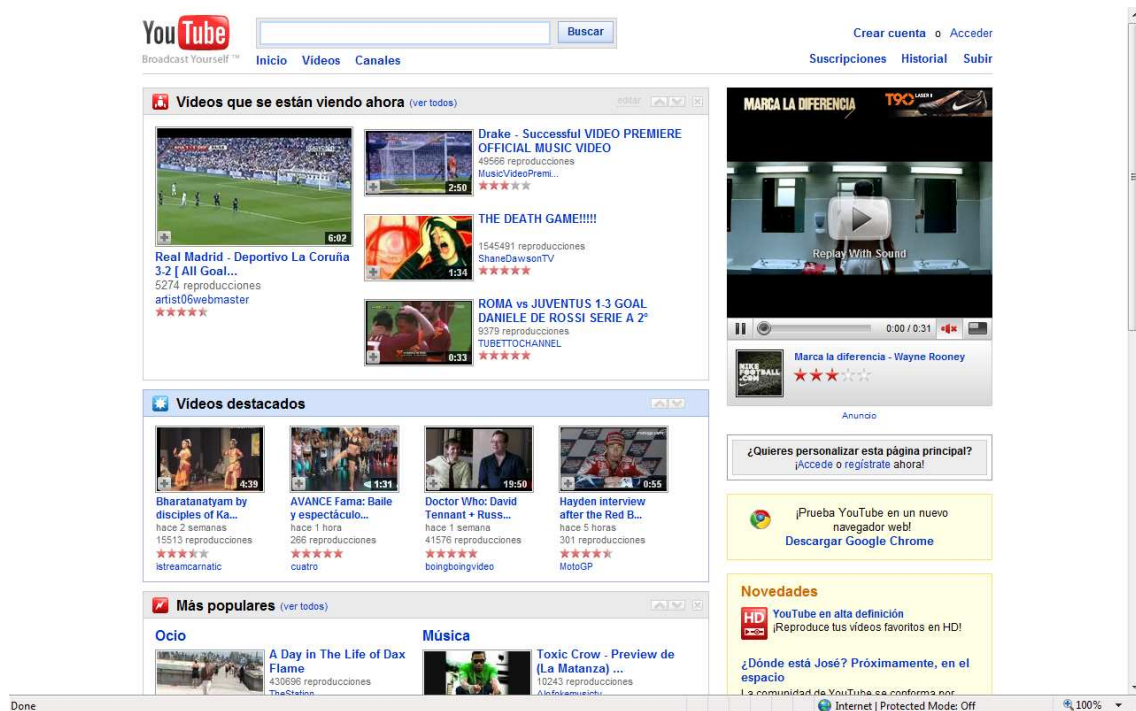


Figura 9. Portada de Youtube en castellano

Inicialmente fue fundado por tres ex empleados de Paypal en 2005. La compañía fue creciendo hasta que fue comprada por Google. Las claves de su éxito residen en que se evita la necesidad de descargarse un video, o enviarlo mediante correo electrónico. Al difundirse por streaming, y ser convertidos a un formato flash, la duración y calidad del original es prácticamente irrelevante.

Youtube no sólo está encaminado al ocio, sino que muchas empresas ya lo han adoptado como escaparate; incluso se usa para hacer campaña política. Un video publicado en Youtube puede ser visto por millones de personas en poco tiempo.

El interfaz gráfico es la sencillez elevada a la máxima potencia. Si no se encuentra lo que se busca a la primera, o en los videos relacionados, seguramente es que no haya sido publicado. La integración con Google tras la compra del portal original ha ayudado a mejorar el sistema de búsqueda de vídeos.

Como suele ser habitual en las aplicaciones de Google, no hay publicidad visible, o al menos molesta, en ninguna de las páginas de Youtube. Tan sólo aparece la típica publicidad interna de otros productos de Google.

Título:	Youtube
URL:	http://www.youtube.com
Versión WAP:	Sí
Ranking Alexa:	3
Pagerank Google:	9
Idioma:	Aplicación única multi-idioma.
Código abierto:	No
Acceso libre:	Sí
Publicación libre:	Usuarios registrados
Servicios Web:	Reproductor incrustado de videos en otras páginas
Ajax:	Sí
Sindicación:	RSS
Más información:	Integración con servicios y usuarios Google

Tabla 5. Análisis de Youtube

Es uno de los ejemplos más claros y populares de aplicación de la Web Social. No sólo usa las tecnologías que suelen llevar asociadas, sino que su propio fin es el de compartir información entre los usuarios.

Aunque no sea de código abierto, su sencillez hace muy sencillo indagar en su funcionamiento interno. Un ejemplo de esto es que existen aplicaciones que son capaces de llegar hasta el archivo de vídeo y permitir al usuario que lo descargue, aunque Youtube no lo permita, o mejor dicho, “no lo ofrezca”.

Rincón del vago

Es una de las aplicaciones con más solera en el ámbito de la Web en castellano. Pone a disposición de los usuarios una extensa base documental orientada a trabajos, apuntes, exámenes, y otros materiales docentes.



Figura 10. Portada del Rincón del Vago.

Fue creada en 1998 por dos jóvenes de Salamanca, con la intención de que sirviera para que cualquier alumno tuviera a su disposición los trabajos y apuntes que otros alumnos pudieran prestarle. Más tarde pasó a formar parte de la red de portales de Orange.

Desde su creación, la polémica ha acompañado a este sitio web. Sus detractores siempre han esgrimido que facilita, incluso promueve, la copia de trabajos. Por otro lado, sus defensores afirman que tan sólo es un sitio donde documentarse.

El interfaz gráfico deja bastante que desear. Puede resultar confuso en ocasiones, y el abuso en el empleo de publicidad reduce sensiblemente el aprovechamiento óptimo del espacio. En algunos puntos cuesta distinguir qué parte es publicidad, y cuál es contenido.

La clasificación de la información se basa en el Sistema Educativo Español, de modo que dificulta su aprovechamiento para estudiantes de otras naciones. Esta propia clasificación tiende a ser un poco confusa en algunas secciones, de modo que es casi obligatorio el uso del buscador. Aún así, no se garantiza que el elemento obtenido mediante búsqueda sea relevante.

Título:	Rincón del vago
URL:	http://www.rincondelvago.com
Versión WAP:	No
Ranking Alexa:	736
Pagerank Google:	5
Idioma:	Interfaz en castellano, contenido multilingüe.
Código abierto:	No
Acceso libre:	Sí
Publicación libre:	Usuarios registrados
Servicios Web:	No
Ajax:	No
Sindicación:	No
Más información:	Incluye secciones de ocio

Tabla 6. Análisis del Rincón del Vago

Clasificar “El Rincón del Vago” como una aplicación de Web Social sería bastante riguroso. Los contenidos son creados y usados por los usuarios. No usa ninguna de las tecnologías que suelen asociarse con las aplicaciones de Web 2.0. Si tenemos en cuenta que lleva más de 10 años en línea, esto podría deberse a que en su fundación aún prevalecían los paradigmas de la llamada Web 1.0. Se podría decir que este es un ejemplo de un sitio de Web Social que se denomina así ateniéndose únicamente a sus principios.

2.4. Conclusiones

En los apartados anteriores del Estado del Arte se han descrito los tres pilares fundamentales en los que se basa GEO:

- El e-learning o educación a distancia.
- La Web Semántica.
- La Web Social.

A continuación se establecerán las relaciones existentes entre estas tres grandes áreas y que son las que permiten construir la base sobre la que se desarrollará el resto del proyecto. La siguiente figura muestra el entorno en el que se ubica y fundamenta GEO.

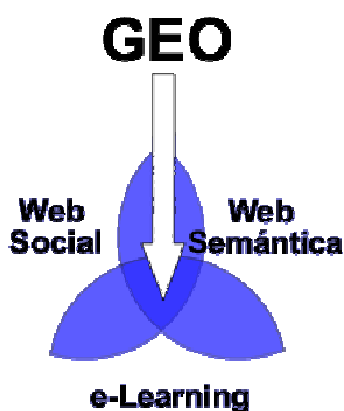


Figura 11. Conceptos fundamentales de GEO

En la actualidad existen un gran número de herramientas, ya sean de pago, gratuitas o de código abierto; que permiten la realización de pruebas de evaluación en línea. Estos sistemas ofrecen un gran abanico de opciones a la hora de construir y realizar un examen, acompañados además de una serie de funcionalidades que permiten una completa gestión de evaluaciones, calificaciones, matriculas y diversos datos implicados en el propio proceso de evaluación. Incluso algunas de estas herramientas están integradas en completos sistemas e-learning lo que les dota de una mayor versatilidad funcional.

Algunos de estos sistemas, incluyen en sus desarrollos módulos de importación y exportación de datos a través de estándares como SCORM e IMS. Estos estándares suponen un gran avance hacia el camino del conocimiento compartido, pues permiten a las herramientas generar contenidos que posteriormente puedan ser utilizados por terceros sistemas. Sin embargo, en la actualidad estas opciones no son utilizadas, en su mayor parte por la falta de funcionalidad en la metodología utilizada para compartir esta información, y en la mayoría de las ocasiones son utilizadas como meras copias de seguridad de los contenidos.

Si se quiere seguir el camino del conocimiento compartido, y poder beneficiarse de las ventajas que ofrece el mismo, el diseño de estos sistemas debe evolucionar hacia una filosofía de Web 2.0. Para ello, los diseñadores y desarrolladores de sistemas deben centrarse en compartir conocimiento como principal objetivo, imponiendo las mínimas restricciones a la hora de gestionar el conocimiento por parte de los usuarios, siempre dentro de los requisitos de seguridad que implícitamente se desprenden del ámbito en el que se encuentra la evaluación en línea. Es decir, a la hora de diseñar sistemas de evaluación los esfuerzos deben centrarse en la generación de un conocimiento estándar así como en los mecanismos que deben desarrollarse para que ese contenido pueda estar siempre disponible por todos los usuarios a nivel global así como por terceros sistemas de una manera ágil y sencilla.

Dando por supuesto que se alcanzara el desarrollo de sistemas de evaluación en línea capaces de compartir el conocimiento generado entre ellos, surgiría un nuevo problema con el que se han encontrado desarrolladores de sistemas bajo los estándares de Web Social: la gestión del conocimiento. En todo sistema en el que los usuarios pueden generar y compartir el conocimiento surge el problema de cómo tratar toda esa información generada ya que la información “en bruto” no sirve de nada. Antes de poder ser puesta a disposición del resto de usuarios, la información debe pasar por un proceso de indexación, validación y catalogación. En respuesta a este problema aparece el concepto de Web Semántica. A través de un motor semántico es posible indexar semánticamente la información, clasificarla, valorarla y lo más impórtate, recuperarla a partir de palabras con contenido semántico y no solo léxico como se hace en la mayor parte de los buscadores actuales. Es por esto por lo que es importante incluir en la gestión del conocimiento las herramientas semánticas necesarias para poder recuperar y gestionar los contenidos generados en todo sistema bajo la filosofía de Web Social.

En Resumen, el sistema de Gestión de Exámenes Online (GEO) a desarrollar, nace de la conjunción de estos tres pilares fundamentales de estudio, en el que se toma la Web Social y la Web Semántica, como solución fundamental para el desarrollo de los futuros sistemas de e-learning.

3. Descripción de la Solución

La descripción de la solución se expone en dos puntos principales. Una solución funcional donde se explica brevemente qué se va a hacer y porqué, justificando cada una de las decisiones tomadas dentro de la funcionalidad del sistema. Además de una solución técnica, donde se expone a alto nivel cómo se va a desarrollar la solución funcional propuesta, así como la descripción de las herramientas que se van a utilizar para su implementación. Por último, por tratarse de un desarrollo software, es imprescindible la descripción de la metodología de desarrollo a utilizar, incluyendo el ciclo de vida que se seguirá en el proceso de desarrollo.

3.1. Descripción Funcional

En la presente sección se presentarán aquellas soluciones funcionales que han sido estudiadas y analizadas como solución al problema propuesto, así como la elección de la misma describiendo los motivos que justifican dicha elección.

Tras un primer análisis, es obvio que la solución pasa por la implementación de una Aplicación Web para la gestión de exámenes en línea, capaz de generar y ofrecer al usuario la posibilidad de realizar el examen para una posterior corrección automática. Dicha aplicación debe tener al menos cuatro módulos:

- Gestión de exámenes, que incluye la funcionalidad básica de generación, realización y corrección de exámenes, así como la gestión de los datos implícitos como son los exámenes, preguntas, respuestas, etc.
- Importación y exportación de los datos generados según un estándar definido previamente y que sea reconocido lo más ampliamente posible.
- Motor semántico para la elección automática de las preguntas a partir de un tema propuesto por el usuario.
- Módulo de comunicación con sistemas externos con el que poder compartir el conocimiento generado.

Partiendo como base de estos axiomas para el inicio de la definición funcional de la solución, se presentan grandes incógnitas a resolver. A continuación, se irán exponiendo éstas incógnitas, dando respuesta a los problemas que presentan llegando a la solución final.

3.1.1. Identificación de roles

Uno de los principales puntos en el análisis preliminar de un sistema, es la identificación del usuario. Si bien es cierto que no encontramos ningún punto en el problema que obligue a recoger esta distinción de roles en la solución, se ha creído oportuno una diferenciación clara de usuarios contemplando una gestión de permisos dependiendo del rol, lo que dotará al sistema de grandes posibilidades de expansión como herramienta de evaluación. Los roles identificados son:

- Alumno. El Alumno será el rol con mayor limitación de permisos limitándose a realizar exámenes y ver la corrección de los mismos cuando proceda.
- Profesor. El profesor será la pieza fundamental en la gestión del conocimiento referente a las preguntas y exámenes propiamente dichas. Será el encargado de añadir conocimiento aportando las preguntas así como la meta información de las mismas que ayude en su clasificación; además de compartir ese conocimiento con otros usuarios. Por supuesto, tendrá otras tareas como la de crear y configurar exámenes, o ver las calificaciones de los mismos.
- Administrador del sistema. Por último, se contempla un rol con todos los permisos para la gestión de profesores, y cualquier dato implicado en el sistema.

3.1.2. Definición de elementos de gestión

A la hora de hablar de gestión de exámenes en línea, hay que definir, identificar y acotar aquellos datos implicados en el sistema cuya gestión caerá dentro del ámbito de desarrollo. Para abordar este punto se han planteado varias soluciones, dependiendo del grado de gestión, que han sido resumidas en tres:

- Gestión Mínima. Únicamente se gestiona aquellas entidades que son directamente utilizadas en la gestión de exámenes, como son las preguntas, y las configuraciones propias que marcan las directrices de un examen.
- Gestión Rígida. Se gestionan aquellas entidades directamente e indirectamente utilizadas en la gestión de exámenes tales como alumnos, profesores, matriculaciones,... además de los datos implícitos en los propios exámenes.
- Gestión parametrizable. Además de gestionar todos los datos implicados en la realización de exámenes, se gestionan las parametrizaciones de las entidades a través del propio administrador. Así se permitirá modificar por ejemplo los datos almacenados de un profesor, una asignatura, etc.

Una vez analizados el esfuerzo que sería necesario para cada una de las opciones, se elige el desarrollo de una gestión parametrizable. Esta decisión ha sido tomada teniendo en cuenta la metodología de desarrollo elegida, que será expuesta en secciones posteriores, y que incentiva un desarrollo evolutivo lo que nos permitirá ir añadiendo los módulos de gestión de una manera dinámica a medida que se va avanzando en el propio desarrollo. Además, esta solución aporta un gran valor añadido al sistema, con un esfuerzo de diseño e implementación medio/bajo dada la reutilización que puede llevarse a cabo de casa uno de los módulos.

A continuación se describen brevemente los módulos de gestión a desarrollar identificados, así como las entidades a las que hace referencia:

- Usuarios: módulo para la gestión de los usuarios del sistema.

- Alumnos y parametrización de los datos: módulo para la gestión de los alumnos, así como la parametrización de los datos que se almacenarán de los mismos.
- Profesores y parametrización de los datos: módulo para la gestión de los profesores incluyendo las asignaturas que imparten y de las que podrán generar exámenes, así como la parametrización de los datos que se almacenarán de los mismos.
- Administradores: módulo para la gestión de los administradores del sistema.
- Asignaturas y parametrización de los datos: módulo para la gestión de las asignaturas, así como la parametrización de los datos que se almacenarán de los mismos.
- Matriculas y parametrización de los datos: módulo para la gestión de las asignaturas, así como la parametrización de los datos que se almacenarán de los mismos.
- Preguntas: modulo para la gestión de la base de conocimiento de preguntas. Este módulo incluirá comunicación con sistemas externos para poder compartir la base de conocimiento, además de contener un subsistema para la estandarización del conocimiento generado y poder comprender el recibido.
- Configuración de examen y parametrización de los datos: módulo para la gestión de los parámetros para la generación y realización de exámenes, como son el número de preguntas, el tipo, tiempo de examen, fecha de realización del mismo, etc. Además se incluye la parametrización de estos datos.
- Generación de Examen: modulo para la generación de un examen individual para un alumno, a partir de los datos de configuración del examen. Dicho módulo tendrá un motor semántico, para la búsqueda y selección de preguntas.
- Corrección de Exámenes: módulo de calificación automática del examen a partir de un examen realizado por el alumno.
- Revisión de Exámenes: módulo para la visualización y revisión de un examen realizado.
- Estadísticas aplicación: módulo de generación y visualización de estadísticas generales de la aplicación.
- Estadísticas asignatura: módulo de generación y visualización de estadísticas individuales de cada asignatura.
- Accesos aplicación: módulo para el control de accesos y permisos a la aplicación y a cada uno de sus módulos.

3.1.3. Definición de permisos

A continuación se exponen en forma de tabla, los permisos de cada rol asociados a cada una de las funcionalidades de gestión identificadas. La tabla identifica cada rol por una letra, así tenemos Administrador (A), Profesor (P) y Alumno (L). Además existen dos tipos de permisos: total (x) y parcial (O).

Funcionalidad	Añadir			Eliminar			Modificar			Ver/Buscar		
	A	P	L	A	P	L	A	P	L	A	P	L
Usuarios	X			x			x			x		
Alumnos	X			x			x		O	x	O	O
Profesores	X			x			x	O		x	O	
Administradores	X			x			x			x		
Asignaturas	X			x			x			x	O	O
Matriculas	X			x			x			x	O	O
Preguntas	X	x		x	x		x	x		x	x	
Configuración de Exámenes	X	x		x	x		x	x		x	x	
Generación de examen												
Corrección de examen												
Revisión de examen	X			x			x			x	O	O
Estadísticas Globales Aplicación										x		
Gestión Estadísticas Asignatura										x	O	
Accesos aplicación												

Tabla 7. Gestión de permisos asociados a funcionalidad

3.1.4. Definición del gestor de conocimiento

Para que el sistema a desarrollar pueda ser de utilidad y novedoso, hay que definir una base de conocimiento de preguntas lo más completa posible y que pueda ser enriquecida diariamente con las aportaciones de los profesores. Es decir, cuantas más preguntas se almacenen en el sistema más posibilidades de realización de exámenes distintos existen. Para que esto ocurra, es necesario maximizar el número profesores que puedan trabajar sobre una base de conocimiento, y yendo más allá en el concepto, se podría extender a varios sistemas trabajando sobre una base de conocimiento común. Esta idea nos lleva a uno de los principales axiomas de la Web 2.0: compartir conocimiento.

En primer lugar, y antes de decantarnos por compartir conocimiento, debemos pararnos a pensar los pros y los contras que existen con la aplicación de este concepto al dominio que nos ocupa. Como ya se ha comentado anteriormente, el hecho de poder compartir el conocimiento generado con otros usuarios a través de diversos sistemas, abre una puerta de acceso a una base de datos a nivel global en el que poder recuperar una ingente cantidad de preguntas con las que configurar exámenes aumentando enormemente las capacidades de generación de exámenes a nivel sistema e incluso las capacidades de entrenamiento a nivel usuario; permitiendo a profesores poder acceder a un gran número de preguntas a partir de las cuales configurar sus exámenes y a los propios alumnos simular la realización de exámenes enriqueciéndose de los contenidos demandados en las simulaciones sin entrar en conflicto con los exámenes generados por el profesor.

Por otro lado, es posible que el hecho de compartir conocimiento en el ámbito de la evaluación, pueda producir el efecto no deseado de que la información pueda ser vista

por otros usuarios que no debieran tener acceso. Es decir, un alumno podría acceder a preguntas que posteriormente formen parte del examen a realizar. Como ya se ha comentado con anterioridad una base de datos lo suficientemente grande, puede hacer que estadísticamente la probabilidad de acceso a preguntas comprometidas sea lo suficientemente pequeña para que deje de ser un problema. Sin embargo, también es cierto que dependiendo del dominio es posible que el número de preguntas no pueda ser lo suficientemente grande y ese conocimiento deba ser protegido frente a usuarios no autorizados.

Teniendo en cuenta estos dos puntos, y sobre todo el enriquecimiento que aporta el poder compartir conocimiento con otros usuarios y sistemas, se ha optado por dotar al sistema de la funcionalidad necesaria para poder tener un acceso compartido a la información generada, además de los soportes necesarios de seguridad para tener la posibilidad de que cierta información sólo sea compartida por un determinado número de usuarios y/o sistemas.

3.1.5. Definición del modelo de distribución de datos

Una vez se ha decidido que el sistema podrá compartir el conocimiento generado, la siguiente tarea es la definición del modelo de distribución de datos, es decir, definir qué datos y cómo se van a compartir.

Si recordamos las distintas funcionalidades que fueron descritas al comienzo de la sección, nos encontramos con datos que sin ningún tipo de dudas no van a ser compartidos, como son usuarios, alumnos, profesores, administradores, matriculas, estadísticas y exámenes generados y realizados. También existen otra serie de datos como son las configuraciones de exámenes y las asignaturas que si bien podrían compartirse se ha tomado la decisión de no hacerlo por no encontrar un estándar o una definición universal sobre los mismos. Por lo tanto, la única entidad que será compartida con otros usuarios y sistemas y que formará nuestro conocimiento compartido son las preguntas, las cuales serán compartidas utilizando el estándar QTI [24].

A la hora de analizar cómo serán compartidos estos datos surgen tres modelos básicos de distribución:

- Distribución total entre sistemas. Cada sistema contiene su propio modelo de datos donde son añadidas aquellas preguntas que se van generando. Cada vez que se quiera hacer una consulta, se realiza sobre aquellos sistemas conocidos.
- Centralizado en el servidor. Los datos de cada sistema son guardados en un único sistema centralizado. Cada vez que se añade una pregunta se hace sobre un servidor. Del mismo modo cada vez que se realice una consulta, todos los sistemas la hacen sobre un único servidor.
- Modelo mixto. Como tercer modelo existe un modelo que combina los dos anteriores. Cada sistema tiene su propio modelo de datos. Al añadir preguntas, éstas son añadidas a su modelo existiendo la posibilidad de subirse al servidor. Así mismo, las consultas son realizadas tanto a su modelo como al del servidor.

Si entramos en detalle sobre cada uno de los modelos, nos encontramos con que cada uno de ellos tiene sus propias ventajas e inconvenientes. Una vez estudiadas las diversas variantes de cada uno en profundidad, se ha optado por utilizar un modelo centralizado en un servidor. Dicho modelo permite centralizar en un único modelo los datos manejados por el resto, escalando la gestión de datos sobre un único sistema disminuyendo en gran medida los problemas de unicidad, duplicidad de preguntas, y cualquier otro problema generado por la redundancia de datos. Por el contrario, este modelo añade una dependencia hacia terceros sistemas y una posible disminución de rendimiento. Ambos parámetros son considerados tolerables por el diseño funcional de la aplicación.

A pesar de que el diseño funcional del servidor recae fuera del alcance del presente proyecto (éste es desarrollado por Diego Jiménez como proyecto paralelo) cabe resaltar algunos detalles del servidor para sentar la base de comunicación en el presente análisis, que dará respuesta a los problemas de seguridad y privacidad de ciertos contenidos planteados con anterioridad. El diseño del servidor de preguntas seguirá un modelo muy utilizado en las recientes aplicaciones Web 2.0 como por ejemplo YouTube. Cada uno de los usuarios, puede subir y clasificar un contenido como público, privado y solo visible por sus “amigos”. De esta forma cada profesor, puede marcar preguntas como privadas o protegidas para que no sean visibles por sus alumnos pero si por otros profesores. Además la existencia de sistemas de votaciones permite una búsqueda en el servidor por distintos parámetros y no solo por la relevancia de la búsqueda. Esta comunicación será descrita detalladamente en los documentos de diseño del sistema incluidos en la presente memoria como anexos.

3.1.6. Definición del motor semántico

Otra de las mayores aportaciones del presente proyecto fin de carrera, es la inclusión de un motor semántico, que a partir de una meta información de la base de conocimiento, pueda gestionar de una manera más eficiente la generación y corrección de exámenes. A partir de esta información, el profesor podría realizar búsquedas de preguntas sobre un tema concreto, a la hora de definir una configuración de examen. Así por ejemplo, un profesor de Ingeniería del Software puede definir un examen que contenga veinte preguntas sobre las fases de la ingeniería del software. Haciendo uso de este motor y de una ontología que defina la ingeniería del software, el sistema buscaría en su base de conocimiento preguntas que contengan términos relacionados con las fases de la ingeniería del software, y devolvería aquellas primeras veinte preguntas más relevantes para la búsqueda establecida. Además, haciendo uso de la meta información aportada por el propio servidor de conocimiento (tal como se ha visto en la sección anterior) se podrían realizar búsquedas por distintos criterios de búsqueda, como son votaciones, usos, de un usuario o grupo de usuario concreto, etc. del tipo “quiero veinte preguntas aleatorias sobre las metodologías de desarrollo software, de entre las cien preguntas más votadas”. Este tipo de búsquedas y meta información será detallada en el proceso de análisis y diseño del sistema que será entregado como anexo a la presente memoria.

A pesar de que en un primer momento, se contempló la implementación del motor semántico como un módulo dentro del desarrollo y ámbito del presente sistema, la aplicación de los diversos conceptos que existen en torno a la Web 2.0 sugirió un nuevo punto de vista más acorde con las decisiones tomadas en torno a la distribución de los datos. Este nuevo enfoque apunta al uso de servicios externos al sistema no solo

para el almacenamiento y recuperación de datos, sino también para la implementación de los diversos algoritmos de búsqueda semántica. Es decir, utilizar servidores de conocimiento a modo de buscadores semánticos, y no solo como meros sistemas de almacenamiento de información. Con esta idea, todo lo referente a la gestión del conocimiento sobre las preguntas recae fuera del alcance del sistema, recurriendo a meras llamadas a servicios ofrecidos por sistemas externos, encargados de gestionar y recuperar la información generada desde el sistema. El desarrollo de este sistema servidor externo, es realizado paralelamente al presente proyecto por Diego Jiménez.

3.1.7. Definición de la estandarización de información

A pesar de que no es el objetivo de esta sección especificar las características de intercambio de información entre los sistemas, sí que se ha creído oportuno mencionar la existencia de un módulo de importación y exportación de preguntas según un estándar reconocido. Tras analizar los diversos estándares actualmente soportados por las aplicaciones de evaluación, se decidió la utilización de QTI (Question and Test Interoperability) especificación creado por el IMS Global Learning Consortium precisamente para la definición de este tipo de información. En posteriores secciones se explicará cómo ha sido utilizado este estándar.

Por último, debe reiterarse el hecho de que la presente sección no se ha querido reflejar detalles propios del análisis del sistema, sino simplemente ofrecer una descripción funcional a alto nivel para sentar las bases del futuro sistema a desarrollar.

3.2. Descripción Técnica

En la presente sección se expondrá una solución técnica, sin entrar en detalles de diseño e implementación, que responda al problema definido anteriormente. Para ello, se describirá brevemente la solución elegida así como las herramientas a utilizar en el desarrollo de la solución, clasificadas por el papel que desempeñan dentro del proyecto.

A muy alto nivel, podríamos describir la solución elegida, como la implementación de una aplicación Web bajo los estándares Java EE [43]. Según se ha definido en la solución funcional, la aplicación de Gestión de Exámenes Online (GEO) quedaría encuadrada dentro del siguiente esquema de comunicación.

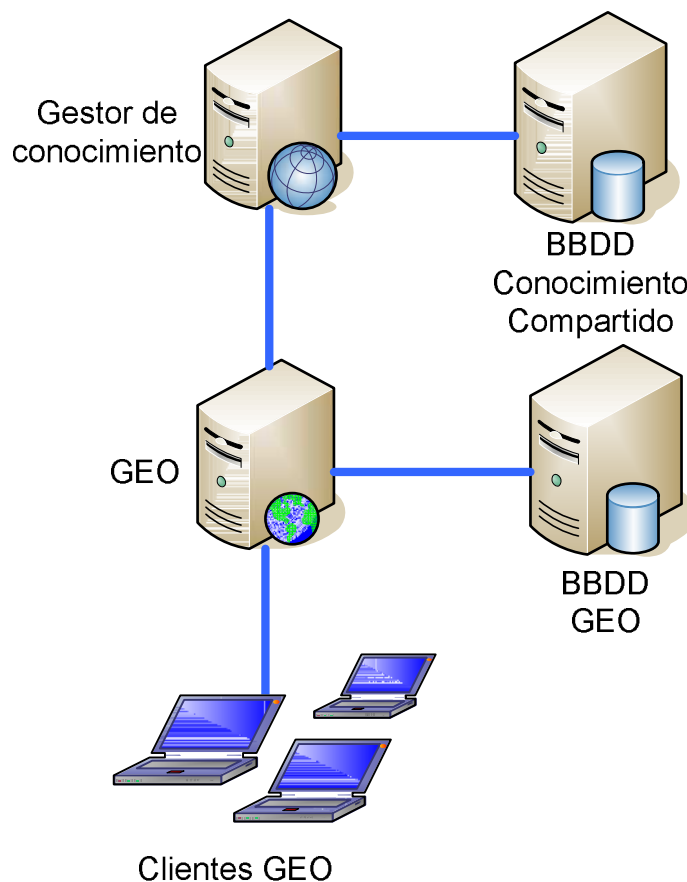


Figura 12. Comunicación de sistemas con GEO

Los clientes del sistema (navegadores Web) lanzan peticiones contra el servidor de aplicaciones GEO. Dependiendo de la petición, dicho servidor puede resolverla localmente, utilizando un gestor de BBDD que puede ser local o distribuido, o en el caso de que requiera la utilización del gestor de conocimiento, lanzar a su vez una petición al mismo. Nótese que en el ámbito del proyecto únicamente recae el desarrollo del sistema GEO así como de su gestor de BBDD.

Una vez definido a alto nivel la comunicación de GEO con el exterior, a continuación se explicará brevemente la solución técnica que se ha definido, siempre desde una visión a alto nivel sin entrar en detalles propios del diseño del sistema que serán adjuntados como anexos a la presente memoria.

El diseño de los componentes de negocio del sistema se realizará mediante una arquitectura dirigida por el modelo utilizando una herramienta de modelado UML como MagicDraw [44] que permitirá un diseño bajo la arquitectura MDA y la generación de código automática con una herramienta como AndroMDA [45].

Como servidor de aplicaciones se utilizará el servidor de Sun GlassFish[46]. La comunicación entre el servidor de aplicaciones y los sistemas externos se hará mediante una filosofía SOA a través de Servicios Web implementados en parte mediante frameworks de desarrollo como Spring integrado con la herramienta AndroMDA .

El Backend de la aplicación se desarrollará utilizando un mapeo de datos objeto-modelo relacional mediante la herramienta Hibernate [47], utilizando MySQL [48] como gestor de BBDD.

El Frontend de la aplicación se implementará mediante un Framework de desarrollo AJAX como ZK [49] integrado en el entorno de desarrollo Eclipse a través del plugin ZK Studio. También se utilizará la API de Struts [50] para la implementación del control del MVC.

En el siguiente diagrama puede verse a muy alto nivel la arquitectura que se seguirá en el desarrollo del sistema, siguiendo el típico modelo de arquitectura Java EE.

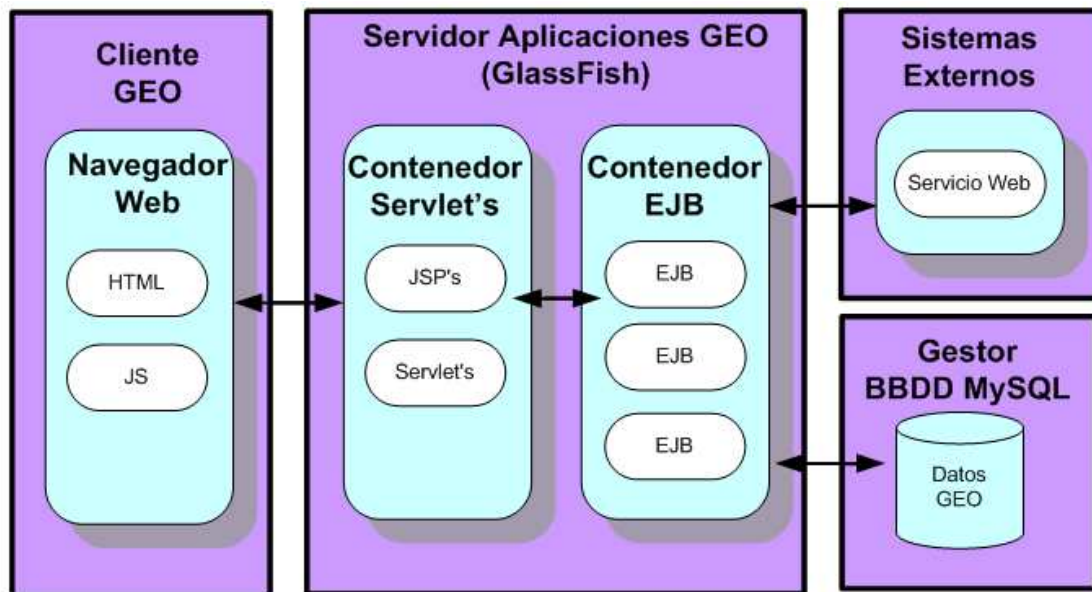


Figura 13. Arquitectura de GEO

Una vez definidos los conceptos básicos de la solución técnica propuesta, a continuación se describirán brevemente las herramientas seleccionadas para el desarrollo del sistema siguiendo un agrupamiento lógico por la función que desempeñan dentro del desarrollo.

3.2.1. Herramientas de modelado y generación de código

Las herramientas de modelado nos permiten realizar los diseños de las aplicaciones siguiendo estándares como UML para que posteriormente podamos pasar el modelo a código a través de herramientas de generación automáticas.

MagicDraw UML Community Edition

MagicDraw UML [44] es un programa para ser utilizado como modelado de UML. Provee soporte completo para metamodelos UML 2.0, incluyendo diagramas de clases, casos de uso, comunicación, secuencia, estado, actividad, implantación, paquetes, componentes, estructuras compuestas y de distribución. Adicionalmente, MagicDraw provee soporte explícito para perfiles UML y diagramas personalizables dando significado al modelamiento visual de arquitecturas.

MagicDraw genera automáticamente sus partes del modelo, todo de acuerdo con los patrones de diseño establecidos: GoF, Java, EJB, JUnit, Esquemas XML, WSDL, CORBA, IDL y cualquier otro patrón personalizable.

Además tiene infinidad de características más que pueden consultarse en la página oficial de la aplicación [44].

AndroMDA

AndroMDA [45] es un framework MDA Open Source, cuya función principal es la de generar código a partir de un modelo, generalmente UML en formato XMI producido por herramientas de modelado. Existen gran cantidad de plugins de AndroMDA (cartuchos y librerías) a través de los cuales se puede generar diversos componentes en forma de código para un gran número de lenguajes entre los que se encuentra Java.

A pesar de que AndroMDA puede ser utilizado en cualquier aplicación, se suele utilizar más en aplicaciones con tecnología Java EE [43]. AndroMDA puede crear un nuevo proyecto Java EE desde cero, en donde el código es generado a partir de un modelo realizado en UML. Entre las grandes posibilidades que ofrece AndroMDA está la de generar el código para Hibernate [47], EJB, Spring [51], WebServices y Struts [50]; integrando el código generado automáticamente al proceso de construcción. AndroMDA es muy eficiente generando código para la aplicación a partir de un modelo, generando la columna vertebral del proyecto, lo que permite que los desarrolladores se mantengan enfocados a la lógica de negocio.

3.2.2. Herramientas de desarrollo

A pesar de que la mayoría de herramientas descritas en esta sección podrían clasificarse como herramientas de desarrollo, únicamente se han incluido en este punto aquellas utilizadas para la edición de código, compilación, y depuración ya sea en un entorno visual o no.

Java Enterprise Edition 5 SDK

Java Platform, Enterprise Edition o Java EE [43] (anteriormente conocido como Java 2 Platform, Enterprise Edition o J2EE hasta la versión 1.4), es una plataforma de programación—parte de la Plataforma Java—para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles

distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

La plataforma Java EE está definida por una especificación similar a otras especificaciones del Java Community Process. Java EE es también considerada informalmente como un estándar debido a que los suministradores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE.

Java EE incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, etc y define cómo coordinarlos. Java EE también configura algunas especificaciones únicas para Java EE para componentes. Estas incluyen Enterprise JavaBeans (EJB), servlets, portlets (siguiendo la especificación de Portlets Java), JavaServer Pages (JSP) y varias tecnologías de servicios web. Esto permite al desarrollador crear una Aplicación de Empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel.

En su versión 5, la plataforma Java EE pasa a ser el estándar de la industria para la aplicación de la arquitectura orientada a servicios (SOA) así como de la próxima generación de aplicaciones Web. Se centra en hacer más fácil el desarrollo conservando la riqueza de la plataforma J2EE 1.4. Además ofrece nuevas y actualizadas funciones, como Enterprise JavaBeans (EJB) 3.0, tecnología JavaServer Faces (JSF), y la última API de servicios Web.

Eclipse Ganymede

Eclipse [52] es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar aplicaciones. Desarrollado originalmente por IBM, actualmente es desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

La versión actual de Eclipse dispone de un gran número de características como son: editor de texto, resaltado de sintaxis, pruebas unitarias con JUnit, control de versiones CVS, integración con Ant, asistentes para la creación de proyectos, refactorización, etc.

Asimismo, existen multitud de plugins para diversas tareas como la integración con Hibernate [47] y Subversión [53] que permiten extender la funcionalidad del entorno.

ZK

ZK [49] es un framework AJAX, escrito completamente en Java, de código abierto, que permite el desarrollo de una rica interfaz de usuario para aplicaciones Web sin usar JavaScript y con una programación simple y escasa.

El núcleo de ZK es un mecanismo conducido por eventos basado en AJAX, sustentado sobre 70 componentes XUL y 80 componentes XHTML, y un lenguaje de

marcación para diseñar interfaces de usuario. Los programadores diseñan las páginas de su aplicación en componentes XUL/XHTML con una gran cantidad de características, y los manipulan con eventos disparados por la actividad del usuario final. Es similar al modelo de programación encontrado en las aplicaciones basadas en GUI de escritorio.

ZK utiliza el acercamiento llamado centrado-en-el-servidor para la sincronización de componentes, el pipelining entre clientes y servidores se hace automáticamente por el motor, y los códigos de Ajax son completamente transparentes para los desarrolladores de aplicaciones Web. Por lo tanto, los usuarios finales obtienen una interacción y respuesta similar a las de una aplicación de escritorio, mientras que la complejidad del desarrollo es similar a la que tendría la codificación de aplicaciones de escritorio.

Además de la programación basada en componentes, de manera similar a Swing, ZK soporta un lenguaje de marcación para la definición de una potente interfaz de usuario llamada ZUML.

- ZUML está diseñado para que desarrolladores no expertos diseñen interfaces de usuario de forma eficiente.
- ZUML permite a un desarrollador mezclar diferentes tipos de lenguaje de marcación, tales como el lenguaje XUL de Mozilla y XHTML, todos ellos en la misma página.
- ZUML permite a los desarrolladores embeber scripts en lenguaje Java (interpretado por BeanShell) y usar expresiones para manipular los componentes y acceder a los datos.

El desarrollo con ZK estará integrado en el IDE Eclipse, a través de un plugin de dicho entorno llamado ZK Studio que forma parte del propio desarrollo de ZK.

3.2.3. Frameworks y API's de desarrollo

Además de las herramientas de desarrollo descritos anteriormente, se han utilizado una serie de API's y frameworks de desarrollo en Java para facilitar algunas tareas.

Struts

Struts [50] es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma Java EE [43]. Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache Struts.

Struts se basa en el patrón del Modelo-Vista-Controlador (MVC). Cuando se programan aplicaciones Web con el patrón MVC con un único controlador, nos encontramos con un grave problema, ya que nuestro controlador se convierte en lo que se conoce como "fat controller", es decir un controlador de peticiones, Struts surge como la solución a este problema ya que implementa un solo controlador (ActionServlet) que evalúa las peticiones del usuario mediante un archivo configurable (struts-config.xml).

Struts permite reducir el tiempo de desarrollo. Su carácter de "software libre" y su compatibilidad con todas las plataformas en que Java Enterprise esté disponible, lo convierte en una herramienta altamente disponible.

Spring

El Spring Framework [51] (también conocido simplemente como Spring) es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java.

A pesar de que Spring Framework no obliga a usar un modelo de programación en particular, se ha popularizado en la comunidad de programadores en Java al considerársele una alternativa y sustituto del modelo de Enterprise JavaBean. Por su diseño el framework ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria.

Mientras que las características fundamentales de este framework pueden emplearse en cualquier aplicación hecha en Java, existen muchas extensiones y mejoras para construir aplicaciones basadas en Web por encima de la plataforma empresarial de Java (Java Enterprise Platform).

Hibernate

Hibernate [47] es una herramienta de mapeo objeto-relacional para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

Hibernate ofrece también un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "criteria").

Hibernate para Java puede ser utilizado en aplicaciones Java independientes o en aplicaciones Java EE, mediante el componente Hibernate Annotations que implementa el estándar JPA, que es parte de esta plataforma.

JAXB

JAXB [54] es una API de Java para el tratamiento de información XML proporcionando un mapeo directo entre XML y los objetos Java. Dado un esquema, que especifica la estructura de los datos XML, el compilador JAXB genera un conjunto de clases de Java que contienen todo el código para analizar los documentos XML basados en el esquema. Una aplicación que utilice las clases generadas puede construir un árbol de objetos Java que representa un documento XML, manipular el contenido del árbol, y regenerar los documentos del árbol, todo ello en XML sin requerir que el desarrollador escriba código de análisis y de proceso complejo.

JAXB tiene infinidad de ventajas entre las que se encuentra, el uso de tecnología Java y XML, garantía de datos válidos, rapidez y facilidad de uso, restricción de datos, personalizable y altamente extensible.

JQTI

Básicamente JQTI [55] es un intérprete para el estándar IMS QTI v2.1 el cual está desarrollado sobre XML. QTI es bastante diferente a la mayoría de estándares sobre XML, ya que no sólo contiene datos, sino también contiene instrucciones para el procesamiento de estos datos. Estas instrucciones, que hacen referencia a las evaluaciones en sí, especifican básicamente cómo han de presentarse las preguntas, así como los contenidos evaluados. De hecho podrías definir la especificación QTI como un propio lenguaje de programación, cuya especificación viene dada por el propio documento XML.

El intérprete JQTI incluye una serie de características que lo hace realmente interesante para el procesamiento de la información bajo el estándar QTI: incluye apoyo para leer, analizar e interpretar los documentos XML QTI, así como para la creación de nuevos contenidos en QTI sobre XML; además de permitir la validación y el informe de errores de los documentos en QTI.

Por último resalta la distribución de JQTI bajo la licencia BSD lo que permite usarlo libremente desde otros desarrollos como el de GEO.

Log4j

Log4j[56] es una biblioteca open source desarrollada en Java por la Apache Software Foundation que permite a los desarrolladores de software elegir la salida y el nivel de granularidad de los mensajes o “logs” a tiempo de ejecución y no a tiempo de compilación como es comúnmente realizado. La configuración de salida y granularidad de los mensajes es realizada a tiempo de ejecución mediante el uso de archivos de configuración externos.

Log4j tiene una serie de características que hace realmente interesante su uso para la generación de ficheros de control y monitorización del sistema. Incluye seis niveles de prioridad de mensaje, se pueden configurar varias salidas de texto simultáneamente, incluso configurar nuestra propia salida tipo servidor remoto, base de datos etc. además es totalmente configurable desde un fichero XML lo que hace que su configuración sea altamente flexible y con una gran facilidad de uso gracias a la API que dispone.

Spring Security 2.0

Spring Security 2.0 [57] es uno de los más desarrollados y ampliamente utilizados proyectos de Spring. Fundado en 2003 y mantenido activamente por SpringSource, hoy en día es utilizado para asegurar numerosos entornos tan exigentes como pueden ser organismos gubernamentales, aplicaciones militares así como empresas bancarias.

Spring Security 2.0 es fácil de aprender, implementar y administrar proporcionando directrices para las operaciones más comunes, permitiendo implementar la seguridad de la aplicación completa en tan sólo unas pocas líneas de XML. Además Spring Security está rodeado de herramientas de integración que permiten facilitar aún más la tarea de configuración del framework y adaptación a cualquier aplicación. Por último resaltar la integración con tecnologías Spring como pueden ser Spring Web Flow, Spring Web Services, SpringSource Enterprise, SpringSource Application Management Suite y SpringSource TC Server.

3.2.4. Herramientas de gestión de datos

Uno de los principales puntos en todo sistema es la persistencia de datos. Como en la mayoría de aplicaciones, para el presente sistema se hará uso de un gestor relacional de base de datos encargado de almacenar los datos de la aplicación en tablas relacionales.

MySQL

MySQL [48] es un gestor de base de datos sencillo de usar y increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales.

Las características principales de MySQL son las siguientes:

- Es un gestor de base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.
- Es Open Source. El código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL para aplicaciones no comerciales.
- Es una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet.
- Existe una gran cantidad de API's para todos los lenguajes (incluido Java) que facilita en gran medida la comunicación con el gestor.

3.2.5. Servidor de Aplicaciones

Tal y como se ha comentado insistentemente, el sistema a desarrollar se trata de una aplicación Web por lo que otro punto importante en la solución tecnológica es la elección del servidor de aplicaciones sobre el que se desplegará la aplicación. A pesar de que en un primer momento se pensó en la utilización de Tomcat por su simplicidad, facilidad de uso y extensión entre los desarrolladores; se abandonó la idea una vez que se decidió la utilización de EJB's para la implementación de los servicios de negocio del sistema ya que Tomcat únicamente es un contenedor de Servlets y no de EJB's.

Glass Fish

GlassFish [46] es un servidor de aplicaciones que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación, tomando como base de desarrollo el Application Server de Sun Microsystems. Es gratuito, de código libre y se distribuye bajo la licencia CDDL y la GNU GPL.

Glass Fish implementa la plataforma Java EE 5, por lo que soporta las últimas versiones de tecnologías como: JSP, JSF, Servlets, EJBs, Java API para Servicios Web (JAX-WS), Arquitectura Java para Enlaces XML (JAXB), Metadatos de Servicios Web para la Plataforma Java 1.0, y muchas otras tecnologías. El servidor Glass Fish está rodeado por una gran comunidad de desarrolladores con el mismo nombre, que están en continuo desarrollo detectando y corrigiendo errores a la vez que dotan al servidor de nuevas funcionalidades.

3.2.6. Herramientas de colaboración

Unas de las herramientas más importantes y muchas veces olvidadas en todo desarrollo software, son las herramientas de gestión de proyectos y colaboración. Este tipo de herramientas facilitan el trabajo en entornos colaborativos, además de facilitar el acceso al código y documentación generada.

SVN / Tortoise

Subversión [53] es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS, el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

Las principales características de subversión son las siguientes:

- Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- Las modificaciones (incluyendo cambios a varios archivos) son atómicas.
- La creación de ramas y etiquetas es una operación más eficiente; Tiene costo de complejidad constante ($O(1)$) y no lineal ($O(n)$) como en CVS.

- Se envían sólo las diferencias en ambas direcciones (en CVS siempre se envían al servidor archivos completos).
- Puede ser servido mediante Apache, sobre WebDAV/DeltaV. Esto permite que clientes WebDAV utilicen Subversion en forma transparente.
- Maneja eficientemente archivos binarios (a diferencia de CVS que los trata internamente como si fueran de texto).
- Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.
- Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autenticar archivos (SQL, LDAP, etc.).

Como clientes de subversión se utilizará Subclipse [58] plugin integrado con Eclipse, así como TortoiseSVN [59] para ser utilizado visualmente integrado con el sistema operativo.

dotProject

dotProject [60] es una herramienta orientada a la Gestión de Proyectos. Para eso se orienta a la administración de recursos para desarrollar un producto, cuya producción requiera de un conjunto de actividades o tareas que se desarrollen entre ellas en forma paralela o independiente.

La aplicación consta de un conjunto de entidades ordenadas jerárquicamente las cuales permiten brindar la funcionalidad del producto. A continuación se mencionan las entidades más importantes de dotProject:

- Compañías: Son las entidades que agrupan proyectos, actividades y usuarios.
- Departamentos: Son áreas dentro de las compañías, que permiten agrupar usuarios en dicho nivel.
- Usuarios/Contactos: dotProject tiene usuarios los cuales son capaces de loguearse a dotProject y trabajar dentro del esquema de permisos que posea el rol de dicho usuario. Los contactos son usuarios especiales que asignados a un determinado proyecto pueden recibir por ejemplo: correo, actualizaciones y noticias pero no necesariamente deben tener acceso al sistema dotProject. Los usuarios y contactos pertenecen a una compañía.
- Proyectos: Es la entidad que contiene el grupo de tareas necesarias para desarrollar un determinado producto.
- Actividades: son las tareas asignadas dentro de un proyecto. Son los componentes sobre los cuales se controla: la duración, dependencias, recursos asignados y progreso. Las actividades deben de pertenecer a un único proyecto.

Descripción de la Solución

- Diagramas de Gantt: Permite ver en forma grafica las actividades ordenadas jerárquicamente, mostrando las dependencias y solapamientos de las mismas.
- Tickets: para administrar todos los problemas relacionados a un proyecto.
- Archivos: Permite almacenar archivos dentro de un proyecto permitiendo un versionado básico de los mismos.
- Foros: Permite la creación de foros de discusión dentro de cada proyecto para distribuir información y discutir temas relativos al proyecto del foro.
- Administración del Sistema: Contiene la actividades relacionadas a la administración de usuarios, roles y configuración del sistema.

3.3. Metodología de Desarrollo

Tras analizar diferentes metodologías tradicionales de desarrollo tales como Métrica v3 [61], Merise[62], así como los Estándares de Desarrollo Software propuestos por la ESA [63]; se ha concluido que las características del proyecto a desarrollar como solución al problema planteado, no se adaptaban a este tipo de metodologías. Es por ello por lo que se ha buscado un nuevo tipo de metodologías más flexibles como son las metodologías ágiles.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas de prácticas en producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
El cliente es parte del equipo de desarrollo	Existe un contrato prefijado, el cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos
Tiempos Flexibles	Tiempos Estrictos
Simplicidad de aplicación	Complejas estructuras

Tabla 8. Comparativa de las metodologías Ágiles frente a las Tradicionales

3.3.1. Metodologías ágiles

Las metodologías ágiles empezaron a desarrollarse en el año 2001 cuando diecisiete expertos en los modelos de desarrollo software, se reunieron para definir los métodos que estaban surgiendo frente a las metodologías tradicionales. De esta reunión surgió el siguiente *Manifiesto Ágil* [64]:

“Estamos poniendo al descubierto mejores métodos para desarrollar software, haciéndolo y ayudando a otros a que lo hagan. Con este trabajo hemos llegado a valorar:

- A los individuos y su interacción, por encima de los procesos y las herramientas.
- El software que funciona, por encima de la documentación exhaustiva.
- La colaboración con el cliente, por encima de la negociación contractual.
- La respuesta al cambio, por encima del seguimiento de un plan.

Aunque hay valor en los elementos de la derecha, valoramos más los de la izquierda.”

Tras los cuatro valores descritos, los firmantes redactaron los siguientes, como los principios que de ellos se derivan:

- Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software de valor.

Descripción de la Solución

- Son bienvenidos los requisitos cambiantes, incluso si llegan tarde al desarrollo. Los procesos ágiles se dobligan al cambio como ventaja competitiva para el cliente.
- Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses, con preferencia en los periodos breves.
- Las personas del negocio y los desarrolladores deben trabajar juntos de forma cotidiana a través del proyecto.
- Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
- La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
- El software que funciona es la principal medida del progreso.
- Los procesos ágiles promueven el desarrollo sostenido. Los patrocinadores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
- La atención continua a la excelencia técnica enaltece la agilidad.
- La simplicidad como arte de maximizar la cantidad de trabajo que no se hace, es esencial.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos que se auto-organizan.
- En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y ajusta su conducta en consecuencia.

A pesar de que este tipo de metodologías centra como uno de los puntos principales la comunicación con el cliente, y el trabajo en contacto con él, punto que obviamente no se cumplirá en el presente desarrollo por no elegir tal figura; se ha elegido este nuevo concepto de metodología de desarrollo por aportar la suficiente flexibilidad en el trabajo y diseño que el ámbito del proyecto necesita.

Tras analizar las principales propuestas de metodologías ágiles, finalmente se ha optado por seguir una de las pioneras en este ámbito: XP (Extreme Programming) [65]. A continuación se expondrás las principales características de esta metodología así como la descripción del ciclo de vida que se propone y seguirá el sistema a desarrollar.

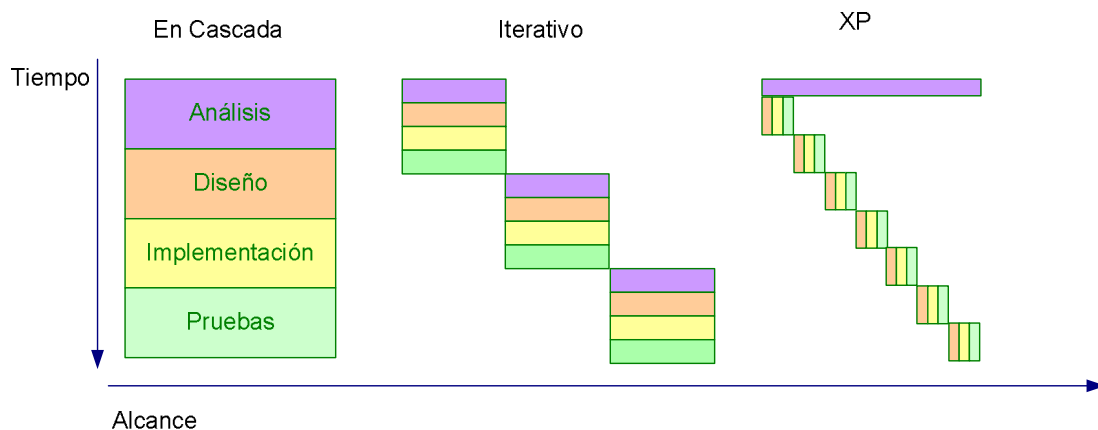
3.3.2. XP (Extreme Programming)

La programación extrema es una metodología ágil desarrollada por Kent Back para la construcción de software. La misma se basa en la simplicidad, comunicación y reutilización del código desarrollado. Su idea básicamente consiste en desarrollar sub-versiones del software, realizando entregas con frecuencia (cada dos semanas o menos). Cada sub-versión, se debe codificar de manera simple para que funcione correctamente. El diseño (o mini-diseño) se hace a medida que se avanzan sobre las sub-versiones. El código y el diseño se van modificando continuamente, añadiéndoles funcionalidades.

A pesar de que XP es especialmente flexible en este punto, se definen los siguientes roles: programador, cliente, encargado de pruebas, encargado de seguimiento, responsable global y gestor. Para el presente proyecto todos estos roles son asumidos por los dos desarrolladores del proyecto, a excepción del rol de cliente y encargado de seguimiento que será compartido con el propio tutor del mismo.

A diferencia de las metodologías tradicionales donde se plantea un desarrollo en vertical, con cuatro procesos claramente diferenciados (análisis, diseño, implementación y pruebas) los cuales se realizan de manera secuencial en grandes periodos de tiempo cada uno; o de las metodologías iterativas las cuales proponen un desarrollo en espiral, cerrando mini-proyectos a partir de pequeños análisis; XP propone una única etapa de análisis en la cual se defina el alcance del proyecto, para después ir resolviendo pequeños problemas planteados en el análisis, realizando paralelamente los procesos de diseño, implementación y pruebas lo que desemboca en pequeñas entregas que van resolviendo requisitos de usuario y son validadas por el propio usuario.

En el gráfico adjunto puede observarse claramente las características comentadas de cada una de las metodologías:



Compilación en tiempo real

Figura 14. Comparativa entre las diferentes metodologías

Para llevar a cabo XP se basa en trece prácticas básica:

1. Equipo completo: está formado por el equipo de desarrollo, el cliente y el responsable del proyecto.
2. Planificación: se observan las historias de usuario y se planifica en qué orden se van a realizar. Puesto que la planificación puede o no respetarse, es necesario revisarla constantemente.

3. Pruebas del cliente: el cliente, con la ayuda de los desarrolladores, propone sus propias pruebas.
4. Versiones pequeñas: las sub-versiones deben ser lo suficientemente pequeñas como para poder implantarse y probarse en un periodo de no más de una semana.
5. Diseño simple: a medida que se implementa el código se deben de ir dando algunos retoques a la interfaz, analizando cada una de ellas y eliminando partes innecesarias de la misma.
6. Pareja de programadores: dos personas aportan más al código que una sola.
7. Desarrollo guiado por las pruebas automáticas: es necesario usar un programa que testee las historias de usuarios.
8. Mejorar el código: mientras se codifica debe mejorarse el código; extraer funcionalidades comunes, eliminar código innecesario, etc.
9. Integración continua: cuando se tenga una nueva funcionalidad, debe recompilarse y probarse.
10. El código es de todos: cualquier persona dentro del equipo de desarrollo deber conocer el código y modificarlo.
11. Normas de codificación: se debe definir un estilo común de codificación.
12. Metáforas: hay que buscar frases que definan cómo funcionan las distintas partes del programa, así todo el equipo entenderá de qué se está hablando.
13. 0 horas extras y días 0: se quiere señalar que no deben realizarse horas extras, y no dejar días sin hacer nada. Tiempo: aproximadamente 40 horas por semana.

3.3.3. Ciclo de Vida de XP

El ciclo de vida ideal de XP consiste de seis fases [66]:

1. Exploración: En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.
2. Planificación de la Entrega (*Release*): En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada

una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación.

3. Iteraciones: Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción. Los elementos que deben tomarse en cuenta durante la elaboración del Plan de la Iteración son: historias de usuario no abordadas, velocidad del proyecto, pruebas de aceptación no superadas en la iteración anterior y tareas no terminadas en la iteración anterior. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.
4. Producción: La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase. Es posible que se rebaje el tiempo que toma cada iteración, de tres a una semana. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento).
5. Mantenimiento: Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción.

Descripción de la Solución

La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

6. Muerte del Proyecto: Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

Tomando estas fases como base de desarrollo, el ciclo de vida de XP viene definido mediante el siguiente diagrama.

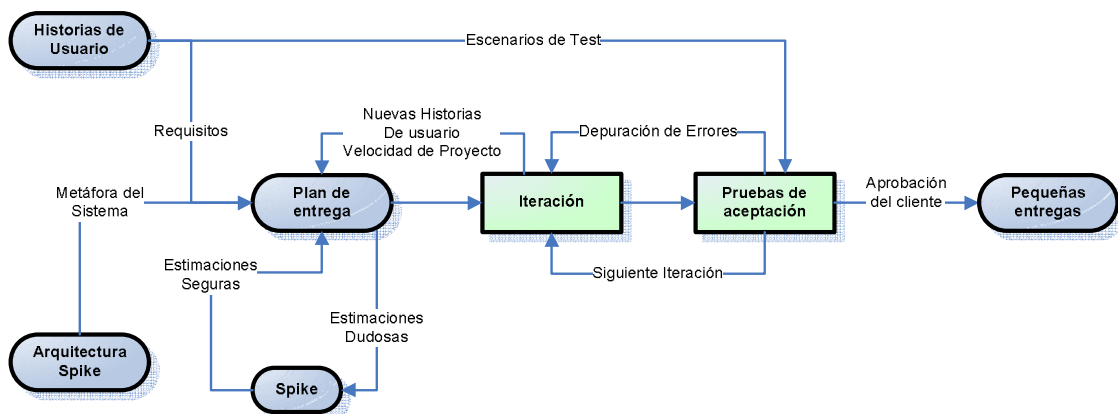


Figura 15. Ciclo de Vida de XP

Las condiciones especiales del proyecto a desarrollar, donde no hay un cliente que demande un software, sino que los requisitos de usuario nacen a partir de la experiencia de los desarrolladores y tutor del proyecto, así como de un estudio de mercado previo; hace que sea imposible aplicar el presente ciclo de vida directamente. Por este motivo se ha creado un ciclo de vida específico para el presente proyecto, partiendo como base el propuesto por XP.

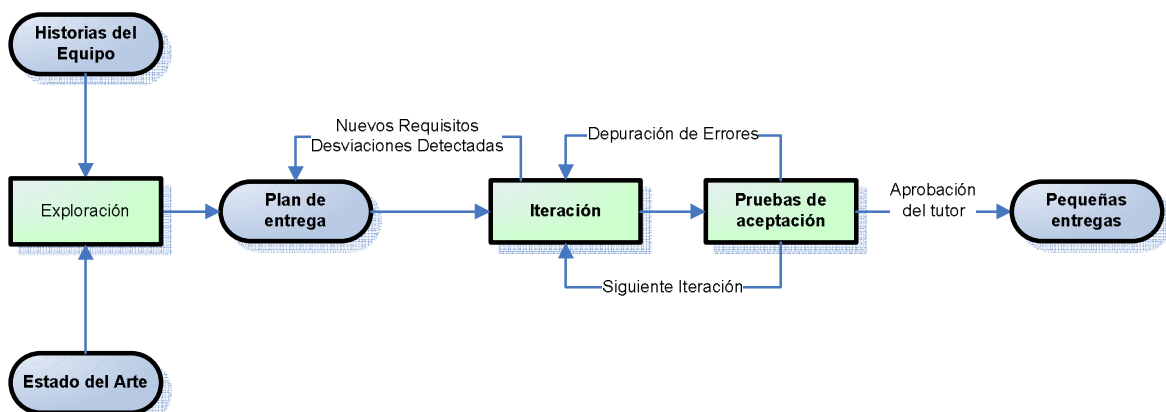


Figura 16. Ciclo de Vida del proyecto GEO

Los principales cambios que encontramos son la desaparición del usuario como fuente de requisitos así como los sistemas de pruebas automáticas, a la vez que se han incluido nuevas fuentes de requisitos como es el estudio de mercado y la experiencia previa del equipo en este tipo de herramientas. También se ha incluido la figura del tutor como parte fundamental en la aprobación de cada uno de los elementos desarrollados.

También es importante resaltar la inclusión explícita de la fase de exploración anteriormente descrita y no incluida en el ciclo de vida de XP

4. Conclusiones

El estilo de vida actual, altamente competitivo, en el que el tiempo ha pasado a convertirse en el mejor activo de cada persona, ha hecho que propuestas tecnológicas como la educación a distancia, o e-learning, pase a convertirse en una alternativa a tener en cuenta, frente a la educación presencial tradicional. Distintos campos de la enseñanza, desde academias privadas, universidades, enseñanzas medias, y prácticamente cualquier organización, tanto pública como privada relacionada con la formación, empieza a admitir y prepararse para el cambio que supone este nuevo concepto de educación. Una de las necesidades surgidas de esta revolución, es la creación de sistemas capaces de evaluar al alumno de una manera no presencial, además de contribuir en el proceso de aprendizaje mediante distintas pruebas auto-evaluables, caracterizadas por contenidos de calidad, que no simplemente sean una mera prueba frente a exámenes reales, sino que aporten un conocimiento de base al alumno.

La solución propuesta en el presente proyecto, deja de lado la percepción de un sistema de evaluación puro, en el que se ha venido trabajando en los últimos años, añadiendo otros conceptos incorporados de novedosas ideas como la Web Social o la Web Semántica, que se han ido desarrollando en los últimos tiempos, paralelamente al e-learning. Tal como marca el fundamento del primero de estos dos conceptos, se presenta un modelo de sistema en el que lo importante no es la evaluación propiamente dicha, sino el conocimiento creado, así como la capacidad de generar colaboración entre usuarios de la herramienta. Con todo esto, se expone una aplicación bajo un entorno colaborativo, en donde el usuario tiene a su disposición un motor semántico, accesible por medio de un servicio proporcionado por un sistema externo, en donde se abandona el concepto de búsqueda a través del léxico de las palabras, utilizando la semántica de las mismas para realizar consultas de contenidos relacionados semánticamente con el texto de la búsqueda introducida.

Resaltar la utilización de los estándares IMS QTI, ampliamente aceptados por las herramientas actuales de evaluación on-line, en el diseño y construcción de la aplicación, lo que permite una independencia de contenidos respecto el resto de sistemas y garantiza una comunicación entre los mismos.

El proceso de desarrollo del sistema se ha realizado siguiendo las directrices marcadas en XP, innovadora metodología ágil, que ha permitido un desarrollo flexible y claro en donde se ha premiado la simplicidad en las tareas y documentación, así como la reutilización de código, en la construcción. Dicha metodología, se ha adaptado a la perfección con la arquitectura MDA utilizada en el diseño del modelo, la cual permite un diseño totalmente flexible y modificable a lo largo del proyecto tal como exige XP. Para el resto de tareas de diseño y codificación, se han seleccionado distintas tecnologías, todas ellas innovadoras en el mundo JEE, que garantizan el mantenimiento simple y flexible de la aplicación.

Por último, destacar con gran satisfacción el trabajo en equipo realizado con Diego Jiménez, al realizar de manera conjunta y bajo un mismo estandarte sendos proyectos fin de carrera.

5. Líneas Futuras

El amplio dominio que abarca el presente proyecto, deja abierta la puerta a una gran cantidad de futuras líneas de investigación y desarrollo sobre el trabajo realizado.

Centrándonos primeramente en aquellas líneas meramente funcionales referentes al desarrollo del sistema, podemos citar las siguientes mejoras y desarrollos a realizar en el futuro:

- Generación de diversas estadísticas sobre la utilización del sistema así como de los resultados obtenidos por los alumnos. La generación de estas estadísticas permitiría al profesor observar con una gran rapidez una información tan valiosa como el índice de suspensos, las preguntas con mayor número de aciertos o los errores más comunes de los alumnos. Así mismo, pueden generarse otras estadísticas dirigidas a un perfil de administración, que permitan por ejemplo conocer el índice de alumnos presentados así como la utilización del sistema entre los alumnos.
- Generación de formulas matemáticas a partir de un editor integrado en el sistema, el cual pueda generar un resultado estándar para ser procesado y compartido con otras aplicaciones siguiendo los estándares de Web 2.0. Mediante este editor, el sistema simplificaría en gran medida la generación de contenido Web que incluya formulas matemáticas, quedando abierto el uso del sistema a las ramas de las matemáticas y la química entre otras.
- Dotación al profesor de mayor control sobre sus alumnos, permitiendo ver los detalles de cada uno de ellos así como facilitando la comunicación entre los mismos. A través de un listado de alumnos por asignatura impartida por el profesor, así como la posibilidad de ver los exámenes realizados por cada uno de ellos, permitiría al profesor ejercer una tarea de tutor, más allá de la de mero “diseñador” y revisor de exámenes.
- A través de una mayor integración con los servicios SOLE, se permitiría acentuar el modelo de Web 2.0; permitiendo gestionar las redes sociales de cada usuario, controlar el contenido generado por cada profesor, editar el mismo, mantener el contacto con otros editores y otra serie de funcionalidades propias del concepto Web 2.0.
- Uno de los puntos “débiles” del sistema de evaluación desarrollado, frente a muchos de los sistemas analizados en el transcurso del proyecto, es la limitación a las preguntas tipo test. Incluyendo otro tipo de preguntas como preguntas de relacionar conceptos, ordenación de respuestas por criterios, y preguntas de rellena de palabras; dotarían al sistema de un gran dinamismo y versatilidad ampliando enormemente la variedad de los contenidos generados.
- Como última línea de desarrollo propuesta se hace referencia a la publicación de comentarios generales capaz de llegar a todos los alumnos de igual forma que se hace en las revisiones individuales.

Abandonando el mero desarrollo del sistema y centrándonos en las futuras líneas de investigación, cabe resaltar como una de las líneas más interesantes, la generación automática de preguntas. Por medio de esta línea, se propone el desarrollo de un servicio, capaz de generar de forma automática contenidos de evaluación; tomando como base para ello las propias ontologías que dispone el sistema SOLE. El desarrollo de este sistema, permitiría la concepción de una herramienta, que a partir de ontologías fuera capaz de generar exámenes ilimitados, lo que supone un evaluador totalmente automatizado, capaz de no solo evaluar, sino entrenar al alumno en cualquier dominio definible por una ontología.

Siguiendo la línea anterior y ampliando los horizontes del desarrollo, se propone el reto de no solo generar contenidos tipo test, o respuestas “auto-correctibles”; sino desarrollar un servicio que a través de un sistema de IA sea capaz de generar y corregir automáticamente preguntas con contenido a desarrollar por los alumnos.

Para finalizar la presente sección y de forma paralela a las propuestas descritas con anterioridad, cabe contemplar como nueva línea de investigación la inclusión de un subsistema de reconocimiento de voz, el cual permitiría que el sistema fuera usado por personas con discapacidad visual.

Bibliografía y Referencias

- [1] Hortensia Mañas. Universidad de Zaragoza. *El e-learning como herramienta estratégica en el grupo telefónica. Boletín de la Sociedad de la Información: Tecnología e Innovación.*
<http://sociedaddelainformacion.telefonica.es/jsp/articulos/detalle.jsp?elem=2383>
- [2] Artículo “Web 2.0” de Wikipedia la Enciclopedia Libre.
http://es.wikipedia.org/wiki/Web_2.0
- [3] Berners-Lee, T., Hendler, J. & Lassila, O. (2001). *The Semantic Web. Scientific American*, 284(5), 34-43.
- [4] Warren, P. (2006). *Knowledge Management and the Semantic Web: From Scenario to Technology.* IEEE Intelligent Systems, 21(1), 53-59.
- [5] Naeve, A. (2005). *The Human Semantic Web Shifting from Knowledge Push to Knowledge Pull.* International Journal of Semantic Web & Information Systems, 1(3), 1-30.
- [6] Davies, J., Lytras, M.D. & Sheth, A.P. (2007). *Semantic-Web-Based Knowledge Management.* IEEE Internet Computing, 11(5), 14-16.
- [7] Fensel, D., Munsen, M.A. (2001). *The Semantic Web: A Brain for Humankind.* IEEE Intelligent Systems, 16(2), 24-25.
- [8] Vossen, G., Lytras, M. D. & Koudas, N. (2007). *Editorial: Revisiting the (Machine) Semantic Web: The Missing Layers for the Human Semantic Web.* IEEE Transactions on Knowledge and Data Engineering, 19(2), 145-148.
- [9] Francisco José García Peñalvo Universidad de Salamanca. *Estado actual de los sistemas e-learning.*
http://www.usal.es/rev_numero_06_2/n6_02_art_garcia_penalvo.htm
- [10] García Peñalvo, F.J. y García Carrasco, J. (2001) *Los espacios virtuales educativos en el ámbito de Internet: Un refuerzo a la formación tradicional.*, Teoría de la Educación. Educación y Cultura en la Sociedad de la Información.
http://www3.usal.es/~teoriaeducacion/rev_numero_03/n3_art_garcia-garcia.htm
- [11] Rosenberg, M.J. (2001) *E-learning strategies for delivering knowledge in the digital age.* McGraw-Hill.
- [12] Lozano Galera, J. (2004) *El triángulo del e-learning.*
- [13] Rengarajan, R. (2001) *LCMS and LMS: Taking advantage of tight integration.*
http://www.e-learn.cz/soubory/lcms_and_lms.pdf
- [14] Filosofía Moodle.
<http://docs.moodle.org/es/Filosof%C3%Ada>

- [15] Juan Carlos Lozano. *Técnicas y Herramientas de evaluación online*.
http://www.verticelearning.com/articulos/tecnicas_y_herramientas_de_evaluacion_on_line.html.
- [16] Herramienta de Evaluación OnLine Autor.
http://www.verticelearning.com/e-learning_Herramienta_autor_online.html
- [17] Software de Evaluación OnLine Perception.
<http://espanol.scantron.com/software/perception.htm>
- [18] Web de realización y publicación de Test.
<http://www.createst.com>
- [19] Web del grupo de Investigación HEOL del Centro de Estudios Felipe II.
<http://dosi.itis.cesfelipesecondo.com/heol/index.php>
- [20] Joglar Prieto, N. y otros. *Evaluación Online en educación Secundaria y Universitaria: Una experiencia de enseñanza-aprendizaje*.
http://dosi.itis.cesfelipesecondo.com/heol/archivos/Joglar_et_al_Evaluacion_Online_Anales_ITIS.pdf
- [21] Web del Proyecto Exanet.
<http://sourceforge.net/projects/exanet/>
- [22] Especificación del Estándar SCORM.
<http://www.adlnet.gov/scorm/>
- [23] IMS Global Learning Consortium
<http://www.imsglobal.org>
- [24] IMS Question & Test Interoperability Specification
<http://www.imsglobal.org/question/index.html>
- [25] Guía Breve de la Web Semántica. (2008). W3C.
<http://www.w3c.es/Divulgacion/Guiasbreves/WebSemantica>.
- [26] *Resource Description Framework (RDF): Concepts and Abstract Syntax*.
Resource Description Framework (RDF) (2004)
<http://www.w3.org/TR/rdf-concepts/>.
- [27] *RDF-Semantics*. (2004), W3C.
<http://www.w3.org/TR/2004/REC-rdf-mt-20040210/>.
- [28] *RDF/XML Syntax Specification (Revised)*. (2004). W3C.
<http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/>.
- [29] *RDF Vocabulary Description Language 1.0: RDF Schema*. (2004).
<http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>.
- [30] *SPARQL query language for RDF*. *SPARQL query language for RDF* (2008), W3C. <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>.

- [31] *Visión general del Lenguaje de Ontologías Web (OWL)*.(2004), W3C.
<http://www.w3.org/2007/09/OWL-Overview-es.html>.
- [32] Preguntas frecuentes sobre el Lenguaje de Ontologías Web (OWL) del W3C, (2004). <http://www.w3c.es/Traducciones/es/SW/2005/owlfaq>.
- [33] Wongthongtham P, Chang E, Dillon T y Sommerville I. *Software Engineering Ontology – the Instance Knowledge* (Part I). 2, s.l. : IJCSNS International Journal of Computer Science and Network Security, 2007, Vol. 7.
- [34] Wongthongtham, Dr. P. *Ontology-based Software Engineering. SEOntology: The Software Engineering Ontology*. (2007).
<http://www.seontology.org/doc/Onto-basedSE.pdf>.
- [35] *SEOntology. Introduction to the Software Engineering Ontology*.
http://www.seontology.org/index.php?option=com_content&task=view&id=15&Itemid=31#s.
- [36] Wongthongtham, Pornpit, Chang, Elizabeth y Sommerville, Ian. *Software Engineering Ontology for Software Engineering Knowledge*. Budapest, Hungría : 10th International Protégé Conference, 2007
- [37] Van Der Hest S., Christian. *¿Qué es la Web 2.0?* Maestros del Web, (200. <http://www.maestrosdelweb.com/editorial/web2/>.
- [38] O'Reilly, Tim. *What is Web 2.0*. (2005)
<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>.
- [39] Graham, Paul. *Web 2.0*. (2005)
<http://www.paulgraham.com/web20.html>.
- [40] Garrett, Jesse James. *Ajax: A New Approach to Web Applications*. Adaptive Path. (2005).
<http://adaptivepath.com/ideas/essays/archives/000385.php>.
- [41] Pilgrim, Mark. *What Is RSS*. XML.com. (2002).
<http://www.xml.com/lpt/a/2002/12/18/dive-into-xml.html>.
- [42] *Guía breve de los Servicios Web*. (2008), W3C.
<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
- [43] Java Platform, Enterprise Edition o Java EE
<http://java.sun.com/javaee/>
- [44] MagicDraw, herramienta de modelado UML.
<http://www.magicdraw.com/>
- [45] AndroMDA, framework de generación de código con un modelo MDA.
<http://www.andromda.org/>

- [46] GlassFish, servidor de aplicaciones desarrollado por Sun Microsystems.
<https://glassfish.dev.java.net/>
- [47] Hibernate, framework de persistencia de datos.
<https://www.hibernate.org/>
- [48] MySQL, gestor de Bases de Datos.
<http://www.mysql.com/>
- [49] ZK, framework de desarrollo Ajax.
<http://www.zkoss.org/>
- [50] Struts, framework de desarrollo de aplicaciones J2EE bajo el patrón MVC. <http://struts.apache.org/>
- [51] Spring, framework de desarrollo de aplicaciones J2EE.
<http://www.springsource.org/>
- [52] Entorno de desarrollo Eclipse.
<http://www.eclipse.org/>
- [53] Sistema de control de versiones SubVersion.
<http://subversion.tigris.org/>
- [54] Jaxb, librería Java para el tratamiento de información XML.
<https://jaxb.dev.java.net/>
- [55] Jqti, librería Java para el tratamiento de información bajo el estándar QTI. <http://jqti.qtitools.org/>
- [56] Log4j, librería Java para la generación de ficheros de Log.
<http://logging.apache.org/log4j/1.2/index.html>
- [57] Spring Security, framework para el desarrollo de la seguridad sobre aplicaciones J2EE.
<http://static.springsource.org/spring-security/site/>
- [58] Cliente de subversion en forma de Plugin para Eclipse.
<http://subclipse.tigris.org/>
- [59] Cliente gráfico de subversion integrado en el sistema de ventanas del SO.
<http://tortoissvn.tigris.org/>
- [60] dotProject.NET, herramienta OnLine para la gestión de proyectos.
<http://www.dotproject.net/>
- [61] Metodología de desarrollo software Métrica v3.
<http://www.csi.map.es/csi/metrica3/index.html>
- [62] Metodología de desarrollo software Merise.
<http://www.commentcamarche.net/contents/merise/concintro.php3>

- [63] Metodología de desarrollo software de la ESA.
http://www.esa.int/TEC/Software_engineering_and_standardisation/TECP5EU_XBQE_1.html
- [64] Manifiesto del desarrollo software ágil.
<http://www.agilemanifesto.org/>
- [65] Metodología de desarrollo software XP.
<http://www.extremeprogramming.org/>
- [66] Patricio Letelier y M^a Carmen Penadés. *Métodologías Ágiles para el desarrollo de software: eXtreme Programming (XP)*. (2004)
<http://www.willydev.net/descargas/masyxp.pdf>

Anexos

Análisis del Sistema de Información



Sistema de Gestión de Exámenes Online GEO

**Análisis del Sistema de
Información
Madrid, 05 de Marzo de 2009**

Autor	Borja Blanco Iglesias
Fecha de Creación	05/03/2009
Fecha de actualización	20/03/2009
Nombre de documento	ASI_GEO.doc
Versión	1.0

Histórico de Versiones

Fecha	Autor	Versión	Comentario
20/03/2009	Borja Blanco	1.0	Versión inicial del documento.

Revisores

Nombre	Posición
Borja Blanco Iglesias	Autor del Proyecto
Ricardo Colomo Palacios	Tutor del Proyecto

Lista de distribución

Copia N°	Nombre	Posición
1.	Borja Blanco Iglesias	Autor del Proyecto
2.	Ricardo Colomo Palacios	Tutor del Proyecto
3.	Diego Jiménez López	Colaborador
4.		
5.		

Introducción

El Análisis de Sistemas trata básicamente de determinar los objetivos y límites del sistema, caracterizar su estructura y funcionamiento, marcar las directrices que permitan alcanzar los objetivos propuestos y evaluar los pasos previos al diseño de un nuevo sistema-producto.

Objetivos del Documento

El presente documento tiene como objetivo principal, especificar y documentar los resultados obtenidos en el proceso de análisis realizado, sirviendo como referencia de las tareas e iteraciones a realizar en el futuro desarrollo del sistema bajo la metodología XP.

En las distintas secciones del documento, quedará definido el alcance del sistema así como la descripción de cada uno de los módulos a realizar, definiendo la prioridad de cada uno y dando como resultado de esa categorización la estrategia de desarrollo modular e iterativo a seguir en las futuras fases de desarrollo.

Audiencia

Éste documento, de carácter general, va dirigido a cualquier persona que forme parte de las futuras tareas de desarrollo, pruebas y mantenimiento del sistema; principalmente orientado a los encargados de las fases de diseño, que deberán tomar el presente documento como base de su trabajo.

Definiciones, acrónimos, abreviaturas y referencias

A continuación se incluye información relativa a la terminología utilizada en el documento así como a las referencias realizadas en el mismo.

- *IMS QTI(Question & Test Interoperability)*: Especificación del IMS Global Learning Consortium para la definición de formularios de evaluación.
- *SOLE*: Servidor OnLine de Exámenes.
- *XP (Extreme Programming)*: Metodología de desarrollo ágil cuya traducción al español es Programación Extrema.

Definición del Alcance del Sistema

A muy alto nivel se podría definir el alcance del sistema como el desarrollo de un sistema de gestión de exámenes online, capaz de gestionar no solo los datos implicados en este tipo de sistemas, sino que además sea capaz de generar exámenes pseudo aleatorios a partir de una configuración base dada, así como evaluarlos automáticamente. Además, los exámenes se deben generar bajo un estándar con meta información para poder ser intercambiados y compartidos con otros sistemas, a la vez que permita la búsqueda semántica del contenido generado. A continuación se expondrá aquellos requisitos que debe contemplar el sistema distribuidos en módulos funcionales, a la vez que se establecerá una prioridad y estimación de esfuerzo de cada uno.

Identificación de los módulos de desarrollo

Siguiendo con la metodología definida para el desarrollo del proyecto, en la presente sección se definirán aquellos módulos funcionales de los que se compone el sistema. Dichos módulos cubren en su totalidad la funcionalidad a desarrollar identificada en el análisis realizado sobre el estudio de mercado, así como las “historias” extraídas de las experiencias previas de los desarrolladores. A continuación se presentarán los cinco módulos identificados.

Gestión de Exámenes

Funcionalmente hablando, la gestión de exámenes es el módulo principal de la aplicación. Incluye toda la funcionalidad de generación, realización y corrección de exámenes, así como la gestión de todos los datos persistentes relacionados. Dada la complejidad del módulo se ha dividido en los siguientes submódulos:

Control de acceso:

El acceso a la aplicación deber realizarse a través del nombre de usuario y password asociado al mismo. El usuario debe estar previamente registrado por un administrador, y asociado a alguno de los siguientes roles:

- *Alumno*. Rol con mayor limitación de permisos limitándose a realizar exámenes y ver la corrección de los mismos cuando proceda.
- *Profesor*. Rol encargado de añadir preguntas así como la meta información de las mismas que ayude en su clasificación; además de realizar otras tareas como la de crear y configurar exámenes, o ver las calificaciones de los mismos.
- *Administrador del sistema*. Rol con todos los permisos para la gestión de profesores, y cualquier dato implicado en el sistema.

A continuación se definirán los escenarios asociados a cada rol a través de los siguientes diagramas de casos de uso.

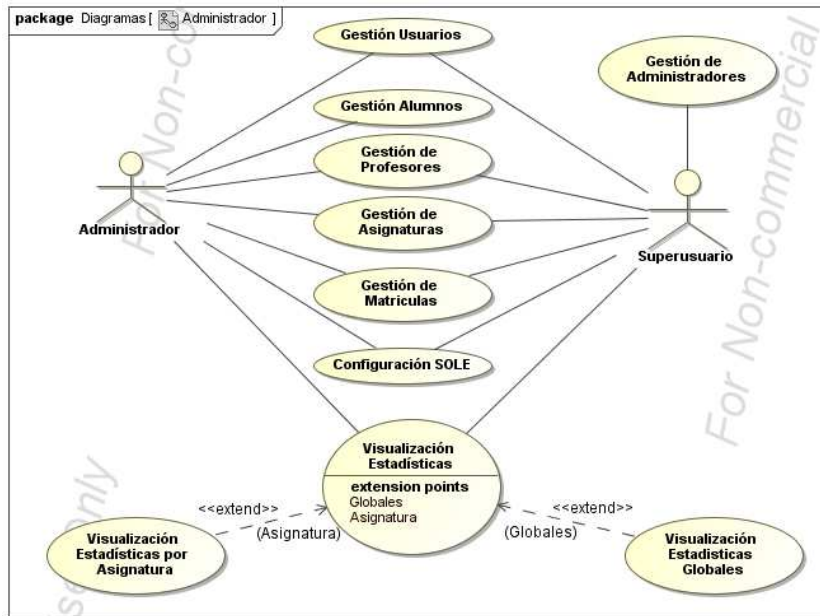


Figura 17. Escenario Administrador

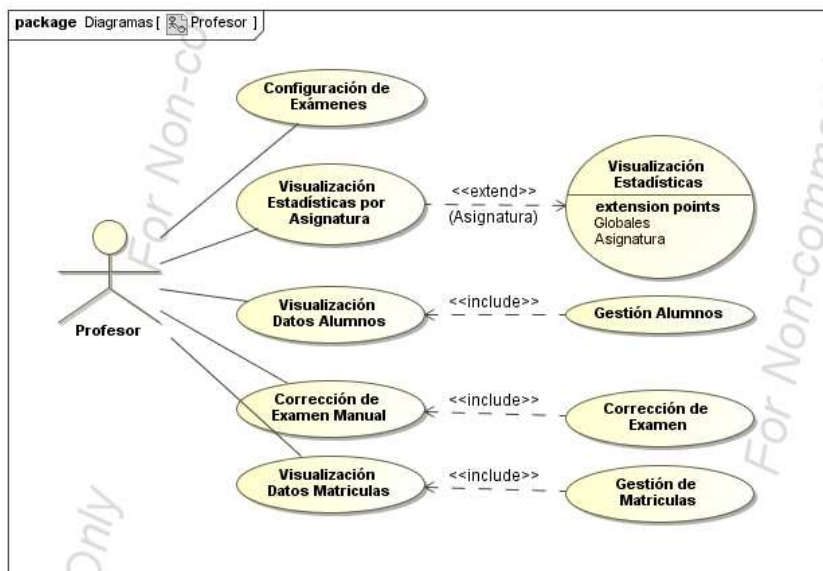


Figura 18. Escenario Profesor



Figura 19. Escenario Alumno

Gestión de Usuarios:

Módulo para la gestión de los usuarios. Para cada usuario deben almacenarse los siguientes datos:

- Identificador del usuario.
- Nombre único de usuario.
- Password de usuario
- Correo electrónico.

Gestión de Administradores:

A través de éste módulo se pueden dar de alta, modificar y eliminar administradores del sistema. Para que un administrador pueda gestionar otros administradores, debe ser súper usuario. Los datos de un administrador son los siguientes:

- Identificador. Identificador que permita identificar un administrador unívocamente.
- Usuario. Aquel usuario al que se asocia el administrador.
- Súper usuario. Marcador para saber si el administrador es súper usuario o no.
- Datos autenticación servidor remoto. Nombre de usuario y contraseña del servidor remoto SOLE.

Gestión de Profesores y parametrización de los datos:

Módulo para la gestión de los profesores, así como la parametrización de los datos que se almacenarán de los mismos. Para cada profesor deben almacenarse los siguientes datos:

- Identificador del profesor.
- Usuario asociado al profesor.
- Nombre y apellidos del profesor.
- Dirección.
- Teléfono.
- Datos autenticación servidor remoto. Nombre de usuario y contraseña del servidor remoto SOLE.
- Asignaturas que imparte el profesor.

Así mismo, los administradores pueden definir campos opcionales que pueden almacenarse de cada profesor.

Gestión de alumnos y parametrización de los datos:

Módulo para la gestión de los alumnos, así como la parametrización de los datos que se almacenarán de los mismos. Para cada alumno los administradores deben gestionar los siguientes datos que serán visibles por los profesores que impartan alguna asignatura en la que esté matriculado el alumno.

- Identificador del alumno.
- Usuario asociado al alumno.
- Nombre y apellidos del alumno.
- Dirección.
- Teléfono.
- Asignaturas en las que está matriculado, incluyendo todos los datos de matriculación.
- Calificaciones obtenidas en cada uno de los exámenes mostrando: asignatura, examen, fecha, calificación.

Así mismo, los administradores pueden definir campos opcionales que pueden almacenarse de cada alumno.

Gestión de Asignaturas y parametrización de los datos:

Módulo para la gestión de las asignaturas, así como la parametrización de los datos que se almacenarán de los mismos. Para cada alumno deben almacenarse los siguientes datos.

- Identificador de la asignatura.
- Nombre de la asignatura.
- Profesores que la imparten, mostrando nombre y apellidos.
- Alumnos matriculados en la asignatura, mostrando nombre y apellidos.

Así mismo, los administradores pueden definir campos opcionales que pueden almacenarse de cada asignatura.

Gestión de Matriculas y parametrización de los datos:

Módulo para el registro de matriculas de un determinado alumno, en una determinada asignatura. Dicho módulo no será independiente, y vendrá implícito en la gestión de alumnos y asignaturas. Para cada matricula debe registrarse:

- Alumno.
- Asignatura.
- Curso académico.

Configuración de examen y parametrización de los datos:

Este módulo al que sólo tiene acceso el rol profesor, es el encargado de establecer y modificar los datos relacionados con la configuración de un examen. Para cada configuración de examen deben almacenarse los siguientes datos:

- Asignatura a la que corresponde el examen.

- Nombre identificativo del examen.
- Definición del periodo de realización del examen.
- Definición del periodo de revisión del examen.
- Tiempo límite para hacer el examen.
- Puntuación total del examen.
- Porcentaje que resta cada respuesta errónea.
- Corrección automática instantánea. Siempre que el tipo de todas las preguntas sea de corrección automática, se puede definir que el sistema informe de la nota una vez terminado el examen, o no.
- Preguntas del examen. El usuario debe poder añadir preguntas a un examen, de dos maneras distintas:
 - Añadiendo manualmente la pregunta. En este caso, el usuario irá añadiendo preguntas a una tabla. En cada pregunta debe definir, el tipo de pregunta, el enunciado, las respuestas a mostrar (en el caso correspondiente dependiendo del tipo) y la o las respuestas correctas. Así mismo debe especificar si la pregunta creada será compartida o no, y en el caso de serlo debe establecer la visibilidad de la misma. Este punto será descrito en el módulo de comunicación.
 - Realizando una búsqueda semántica al motor de preguntas. Este motor devolverá una tabla, que incluirá una pre visualización de la pregunta dando la posibilidad de seleccionar las preguntas a realizar. Esta pre visualización permitirá visualizar la meta información de la pregunta así como valorar y comentar la pregunta a través del módulo de comunicación tal como se expondrá en la descripción de dicho módulo.

Resaltar que para esta primera versión del sistema, únicamente se tendrán en cuenta preguntas tipo test.

- Número de preguntas que componen el examen. En el caso de que el examen deba tener menos preguntas de las seleccionadas, puede generarse un examen con un número determinado de preguntas seleccionadas aleatoriamente.

Así mismo, los administradores pueden definir campos opcionales que pueden almacenarse de cada configuración.

Generación y realización del Examen:

El alumno puede solicitar la realización de un examen que esté en vigor, el cual pertenezca a una asignatura de la que esté matriculado. Una vez validado los permisos del usuario, se procede a generar el examen, mostrando las preguntas correspondientes. El sistema generará un examen aleatorio con las preguntas seleccionadas por el profesor, en el caso de que en el momento de la configuración el profesor lo haya establecido así. El alumno dispondrá del tiempo definido en la configuración del examen para realizarlo. Una vez expirado dicho tiempo, el examen finalizará registrando las respuestas dadas hasta el momento por el alumno.

Corrección de Exámenes:

Una vez realizado un examen, expirado el plazo de realización, abierto el plazo de revisión y en el caso de que la configuración del examen lo permita, el sistema corregirá el examen automáticamente, registrando los resultados y la calificación para una posterior revisión del alumno. En el caso de tratarse de una corrección manual, una vez realizado el examen, el profesor podrá realizar la corrección del mismo puntuando cada pregunta e incluso escribiendo comentarios a cada respuesta dada por el alumno.

Revisión de Exámenes:

Módulo para la visualización y revisión de un examen realizado. Una vez expirado el plazo de realización de un determinado examen, se abre el plazo de revisión, dando la posibilidad a los alumnos de revisar las respuestas correctas, comparar con sus respuestas, ver la puntuación de cada pregunta, y puntuación total obtenida, así como de escribir sus propios comentarios de revisión a cada una de las preguntas.

Visualización de estadísticas de la aplicación:

El sistema debe contar con un módulo de estadísticas capaz de mostrar una serie de gráficos configurables para los siguientes muestreos:

- Índice de accesos por asignatura.
- Índice de cada una de las calificaciones por asignatura.
- Índice de presentados por asignatura.

Visualización de estadísticas de cada asignatura:

El sistema debe contar con un módulo de estadísticas, capaz de mostrar una serie de gráficos configurables para los siguientes muestreos:

- Reparto de calificaciones para cada examen.
- Reparto de calificaciones globales.

Registro accesos aplicación:

Módulo para realizar el registro de acceso a la aplicación. Debe funcionar a modo de histórico, registrando los accesos de los usuarios del sistema así como las operaciones realizadas, mediante el registro de los siguientes eventos:

- Acceso a la aplicación.
- Salida de la aplicación.
- Modificación de cualquier registro de la aplicación: administradores, alumnos, asignaturas, configuración de exámenes, matrículas de alumnos, preguntas, profesores y usuarios.
- Inicio y finalización de la realización de un examen.
- Visualización de una corrección del examen.
- Búsqueda remota de preguntas.
- Importación y exportación de datos.

Así mismo, para cada uno de los eventos anteriores, deben registrarse los siguientes datos:

- Fecha y hora del evento.
- Usuario.
- Tipo de evento.
- Datos implicados en el evento.

Gestión de la configuración SOLE:

Debe existir un módulo de configuración en el que se establezca la URL del servidor SOLE asociado al sistema, a partir del cual se realizarán las búsquedas Online de contenidos.

Importación y exportación de datos

Una de las principales características del sistema debe ser la estandarización de datos a partir de la cual poder compartir los datos generados a la hora de componer un examen. Para ello debe existir un módulo de importación y exportación de datos capaz de transformar los datos introducidos por el usuario, a ficheros XML definidos bajo el estándar IMS QTI 2.1, y a la inversa. Este módulo será utilizado internamente por otros módulos del sistema, como por ejemplo el módulo de comunicación, por lo que no debe contemplarse una interacción directa del usuario con él mismo.

Módulo de comunicación

Siguiendo la idea del módulo anterior, es importante recalcar que debe existir un módulo encargado de las comunicaciones del sistema con sistemas externos. En este caso, estas comunicaciones serán con un sistema servidor de contenidos remoto, SOLE. En principio se definen cinco comunicaciones con este sistema:

- En el ámbito del submódulo de gestión de exámenes, cuando un profesor vaya a realizar una configuración de examen, puede realizar una búsqueda de preguntas. Esta búsqueda lanza una consulta remota a través del módulo de comunicación al servidor de preguntas, para obtener un listado de preguntas categorizadas sobre la búsqueda realizada. La búsqueda contendrá los siguientes campos:
 - Autenticación del usuario que lanza la pregunta, en este caso el profesor.
 - Cadena de consulta, con el texto a buscar en el servidor.
 - Nivel de privacidad:
 - Búsqueda global en el sistema.
 - Búsqueda privada del usuario.
 - Búsqueda en redes a las que esté agregado el usuario.
 - Dominio de búsqueda. Uri con el dominio asociado a la ontología a buscar.
 - Número de preguntas a buscar.
 - Criterio de ordenación.

Esa misma búsqueda devolverá un listado de preguntas con la información relevante de cada pregunta en formato QTI. Cada pregunta estará identificada unívocamente por un ID dado por el propio servidor SOLE.

- Siguiendo con el ámbito del submódulo de configuración de exámenes, cuando un profesor opte por crear preguntas nuevas, se debe producir una nueva comunicación con el servidor de preguntas para compartir la pregunta generada. En esta comunicación el usuario puede optar por ocultar la información, hacerla pública o sólo visible por su grupo. Los campos que deben especificarse en esta llamada deben ser:
 - *Autenticación remota.* Usuario y contraseña del servidor SOLE.
 - *QTI con la pregunta.* Texto XML en formato QTI con la pregunta a compartir.
 - *Dominio de conocimiento.* URI con el dominio donde se incluirá la pregunta generada. Previamente debe haber una consulta al servidor para saber los dominios disponibles del servidor.

Como respuesta a esta petición de compartición de conocimiento, el sistema debe responder con el ID único que ha asignado a dicha pregunta.

- Consulta de dominios disponibles en el sistema. Para realizar algunas de las consultas y peticiones a un servidor SOLE es necesario adjuntar el dominio de conocimiento asociado a la consulta o petición. Es por esto que se debe realizar una consulta previa al sistema para conocer los dominios que ofrece el sistema.
- Al pre visualizar una pregunta ofrecida por el servidor SOLE a la hora de configurar un examen, es necesario mostrar meta información no incluida en la respuesta de búsqueda del servidor. Para conocer esta meta información (valoración y comentarios de la pregunta) es necesario realizar una consulta previa, en la que, a partir del identificador de la pregunta, el servidor devuelve la meta información.
- Al igual que la consulta anterior, a la hora de pre visualizar la información de la pregunta, debe darse la posibilidad al usuario de valorar y comentar la pregunta descargada, para ello se realiza una petición de valoración adjuntando la puntuación y los comentarios, así como el identificador de la pregunta a valorar.

Es importante saber que en cada envío o petición realizada al servidor SOLE es necesario adjuntar los datos necesarios para que se realice la autenticación del usuario en el sistema remoto SOLE.

Buscador Semántico

El sistema debe realizar búsquedas semánticas a la hora de elegir las preguntas de un examen dentro de la configuración del mismo. Este sistema de búsqueda semántica no será incluido en el desarrollo del sistema sino que se accederá a través de un servicio ofrecido por el servidor SOLE por medio de la consulta de búsqueda del módulo de comunicación.

Priorización y estimación de Módulos

Una vez definida la funcionalidad de cada módulo, se procederá a otorgar una prioridad y estimación de esfuerzo a cada uno de los módulos y submódulos descritos. Sin embargo, para poder definir la prioridad de cada uno, es necesario establecer las dependencias entre los mismos a través del siguiente diagrama.

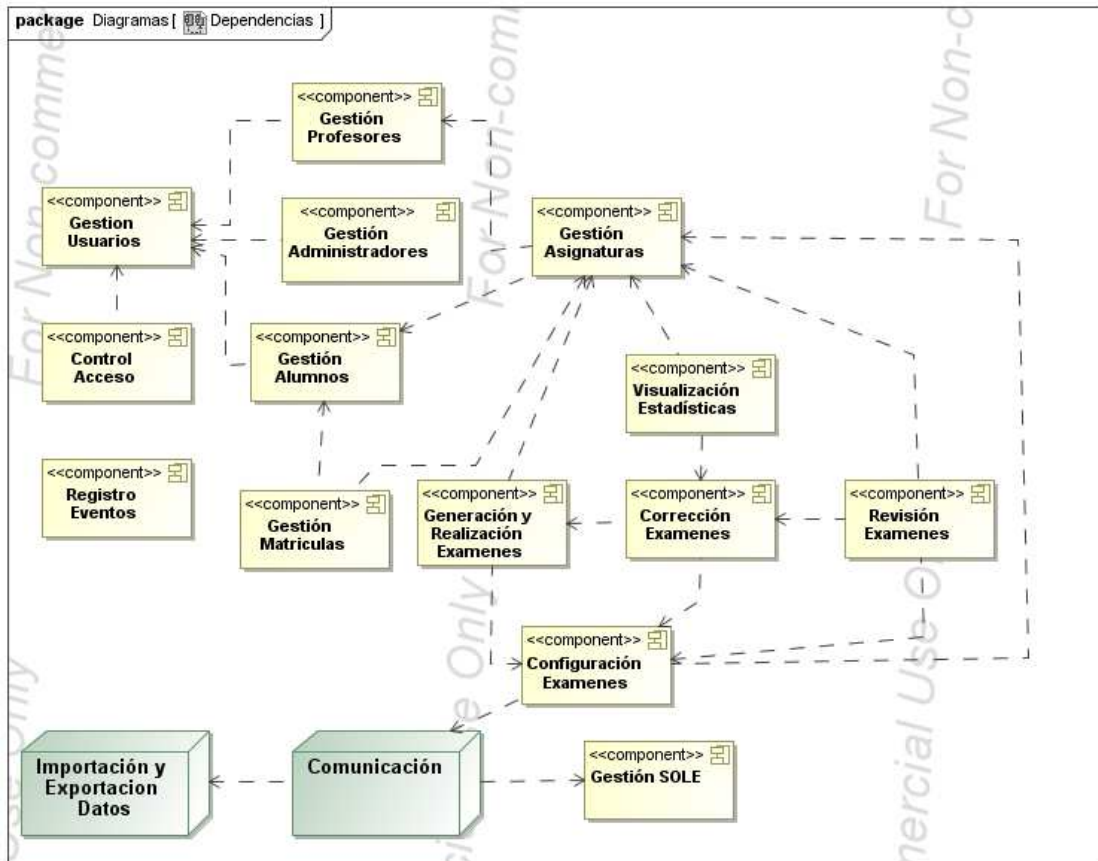


Figura 20. Dependencias entre módulos.

Basándonos en las dependencias descritas, se procede a especificar la prioridad y esfuerzo necesario mediante los siguientes parámetros.

La prioridad queda definida por un número del uno al tres donde:

- Prioridad uno. El resto de módulos dependen de él o bien es un módulo de vital importancia para el proyecto por diversos factores. Se debe priorizar el desarrollo de este módulo sobre el resto ya que paraliza el desarrollo del sistema.
- Prioridad dos. Otros módulos dependen de él, debe tenerse en cuenta para la siguiente iteración.
- Prioridad tres. Es un módulo independiente, ningún módulo depende de él. Su no desarrollo no obstaculiza el desarrollo de ningún otro módulo.

Así mismo, la estimación del esfuerzo se realiza utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Los módulos funcionales generalmente valen de 1 a 3 puntos.

<i>Modulo</i>	<i>Submódulo</i>	<i>Prioridad</i>	<i>Estimación Esfuerzo</i>
Configuración Exámenes	Control Acceso	3	1
	Gestión Usuarios	1	0,5
	Gestión Administradores	2	0,5
	Gestión profesores	2	1
	Gestión Alumnos	2	1
	Gestión Asignaturas	1	1
	Gestión Matriculas	1	0,5
	Configurar Exámenes	1	2
	Realizar Exámenes	2	4
	Corregir Exámenes	2	1
	Revisar Exámenes	3	2
	Visualizar Estadísticas Globales	3	1
	Visualizar Estadísticas por Asignatura	3	1
	Registro Eventos Aplicación	3	1
Gestión SOLE	2	0,5	
Importación y Exportación de datos	-	2	3
Comunicación	-	2	3

Tabla 9. Prioridad y estimación de módulos

Tomando las estimaciones realizadas como base para una primera planificación nos encontraríamos con el siguiente cronograma inicial y orientativo.

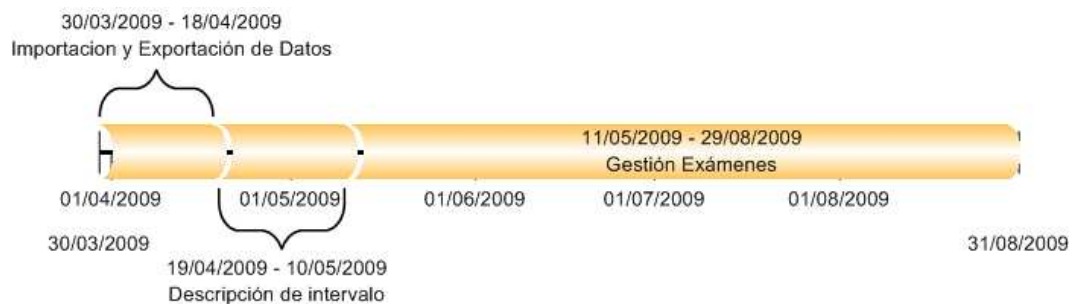


Figura 21. Dependencias entre módulos.

En la sección posterior se procederá a refinar esta planificación incluyendo las iteraciones en el desarrollo así como las fases de desarrollo marcadas en la metodología utilizada.

Planificación

Una vez se ha asignado una prioridad a cada uno de los módulos y submódulos que componen el sistema y se ha estimado el esfuerzo individualmente, basándose en la experiencia de desarrollo de proyectos anteriores; se establecerá la planificación de las tareas de desarrollo teniendo en cuenta la división de iteraciones que marca la metodología XP. Para ello, en primer lugar, se definirá cada una de las iteraciones, así como los módulos implicados en cada una.

Definición de Iteraciones

Atendiendo a las prioridades establecidas, relaciones entre módulos, carácter del proyecto y otra serie de factores, se ha creído oportuno la división del proyecto en cinco iteraciones de diversa duración (entre una y cinco semanas) cada una.

En la siguiente tabla puede verse esta división en iteraciones, así como la agrupación de módulos por cada iteración. Cabe resaltar que en la columna de estimación de esfuerzo, siguiendo con la métrica utilizada en la sección anterior, se ha incluido la suma de la estimación inicial de cada uno de los módulos, así como una reestimación ajustada a la planificación teniendo en cuenta diversos factores como la reutilización de código.

<i>Iteración</i>	<i>Módulo/Submódulo</i>	<i>Prioridad</i>	<i>Estimación Esfuerzo</i>
1	Gestión Usuarios	1	4,5 / 4
	Gestión Administradores	2	
	Gestión profesores	2	
	Gestión Alumnos	2	
	Gestión Asignaturas	1	
	Gestión Matriculas	1	
2	Importación y Exportación de datos	2	8,5 / 6
	Comunicación	2	
	Configurar Exámenes	1	
	Gestión SOLE	2	
3	Realizar Exámenes	2	5/ 4
	Corregir Exámenes	2	
4	Revisar Exámenes	3	4/ 2
	Visualizar Estadísticas Globales	3	
	Visualizar Estadísticas por Asignatura	3	
5	Control Acceso	3	1/1

Tabla 10. Definición de iteraciones

Nótese la desaparición del submódulo de registro de eventos de la aplicación con respecto a la tabla de la sección anterior. Dicho módulo seguirá un desarrollo en horizontal durante el desarrollo del resto de módulos.

Planificación del sistema

El desarrollo del proyecto se lleva a cabo siguiendo la metodología ágil XP, que marca un ciclo de vida iterativo según marca el siguiente diagrama:

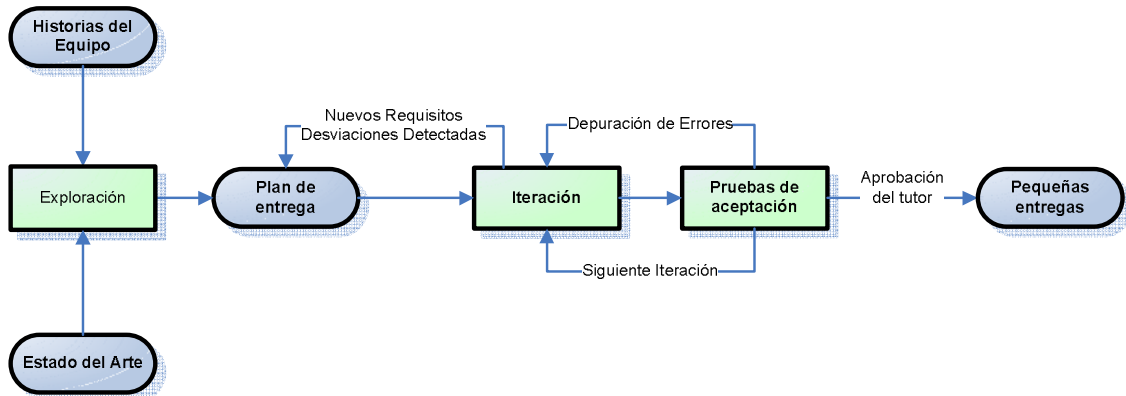


Figura 22. Ciclo de Vida del proyecto GEO

En este diagrama del ciclo de vida que marca XP adaptado a las necesidades del proyecto GEO, observamos tres fases principales.

En una primera fase de *exploración y planificación*, el equipo realiza una investigación sobre el estado del arte relevante al dominio del proyecto, dando lugar a distintas "historias" que serán tenidas en cuenta a modo de requisitos funcionales del sistema. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema. Fruto de esta fase, nace la planificación y división del sistema en iteraciones, a través de otorgar prioridades y estimaciones de esfuerzo a los módulos detectados, dando lugar a los distintos hitos de entrega.

A través de una segunda y tercera *fase iterativa*, el sistema se va *construyendo* por el equipo y *probando* por el responsable de aceptación, a través de pequeñas entregas definidas en la planificación. Cada una de las iteración esta prefijada con anterioridad, dando lugar a un sistema autosuficiente en el cual deben probarse cada uno de los requisitos establecidos sin depender de ninguna otra iteración posterior. Cada vez que una entrega esté validada, se pasará a la siguiente hasta que el sistema se dé por finalizado.

Como norma general, cabe la posibilidad de afrontar una cuarta etapa post desarrollo, que incluiría una puesta en producción y mantenimiento del sistema; sin embargo no se han contemplado estas fases dentro del presente proyecto.

A continuación, y para finalizar la presente planificación, se incluye un cronograma explicativo, en el que se detallan las fechas en las que se realizará cada una de estas fases, así como las distintas iteraciones contempladas anteriormente.

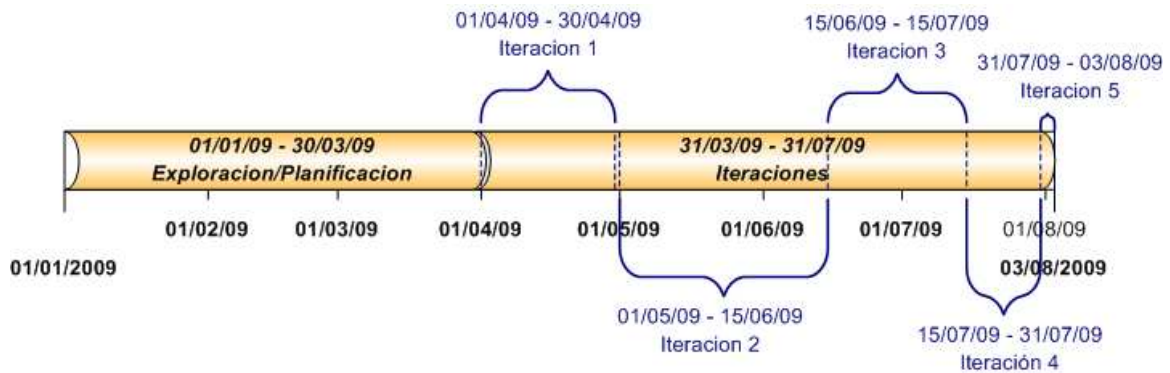


Figura 23. Cronograma con la planificación del proyecto

Tal como puede verse en el anterior diagrama, con la redacción de éste documento finaliza la fase de exploración y planificación, lo que dará lugar al comienzo del desarrollo iterativo del proyecto compuesto de las cinco iteraciones expuestas anteriormente.

En próximas fechas, y coincidiendo con el diseño de los componentes a crear en la primera iteración, se definirá la arquitectura del sistema así como los requisitos técnicos detectados sobre la fase de diseño.

Por último, y a pesar de no haber realizado un análisis de viabilidad y riesgos del proyecto, sí que es importante resaltar un riesgo crítico detectado en la presente planificación que podría llevar al fracaso del proyecto. Al comienzo de la segunda iteración del proyecto, a la hora de diseñar la comunicación del sistema con sistemas externos, debe haber quedado definida la comunicación del sistema SOLE desarrollado paralelamente por Diego Jiménez. En el caso de no haberse definido tal comunicación, el proyecto podría quedar paralizado a falta de tal definición. Como medida proactiva, se fija una reunión semanal con Diego Jiménez, para conocer el estado de su proyecto y de esta forma poder prever el riesgo, y en el caso de cumplirse poder crear con anterioridad un simulador para la funcionalidad necesitada por el sistema.

Validación del Documento

Próximos Hitos e Incidencias

ID	Tema	Resolución	Responsable	Fecha Objetivo	Fecha de Impacto
2	Arquitectura de Sistema	Pendiente iteración 1	Borja Blanco	10/04/09	01/05/09
3	Definición servicios SOLE		Diego Jiménez	15/04/09	01/05/09
4	Desarrollo iteración 1		Borja Blanco	01/05/09	01/05/09

Hitos e Incidencias Cerrados

ID	Tema	Resolución	Responsable	Fecha Objetivo	Fecha Cierre
1	Análisis y Planificación	Finalizado	Borja Blanco	01/04/09	

Aprobación del Documento

Rol	Nombre	Firma	Fecha
Autor del Proyecto	Borja Blanco Iglesias		
Tutor del Proyecto	Ricardo Colomo Palacios		

Documento de Arquitectura del Sistema



**Sistema de Gestión de
Exámenes Online
GEO**

**Documento de Arquitectura
del Sistema
Madrid, 09 de Mayo de 2009**

Autor	Borja Blanco Iglesias
Fecha de Creación	09/05/2009
Fecha de actualización	09/05/2009
Nombre de documento	DA_GEO.doc
Versión	1.0

Histórico de Versiones

Fecha	Autor	Versión	Comentarios
09/05/2009	Borja Blanco	1.0	Versión inicial del documento.

Revisores

Nombre	Posición
Borja Blanco Iglesias	Autor del Proyecto
Ricardo Colomo Palacios	Tutor del Proyecto

Lista de distribución

Copia N°	Nombre	Posición
1.	Borja Blanco Iglesias	Autor del Proyecto
2.	Ricardo Colomo Palacios	Tutor del Proyecto
3.	Diego Jiménez López	Colaborador
4.		
5.		

Introducción

A través de la arquitectura del sistema se especifican las bases de diseño e implementación a seguir a lo largo del desarrollo del sistema, definiendo las diversas vistas arquitectónicas, así como los estándares a aplicar y el entorno tecnológico que rodeará las diversas tareas de desarrollo.

Objetivos del Documento

El presente documento tiene como objetivo principal, especificar y documentar los resultados obtenidos en el proceso de diseño arquitectónico realizado a partir de las tareas de análisis e implementación de maquetas, sirviendo como referencia para las futuras tareas de diseño e implementación de iteraciones a realizar en el desarrollo del sistema bajo la metodología XP.

En las distintas secciones del documento, quedarán definidas las diversas vistas que componen la arquitectura del sistema así como los estándares de diseño y desarrollo a aplicar en las diversas tareas de las futuras iteraciones. Así mismo, se describirá tanto los entornos tecnológicos y de desarrollo a utilizar además de exponer brevemente el entorno tecnológico que se utilizará a lo largo del proyecto.

Audiencia

Éste documento, de carácter general, va dirigido a cualquier persona que forme parte de las futuras tareas de desarrollo, pruebas y mantenimiento del sistema; principalmente orientado a los encargados de las fases de diseño, que deberán tomar el presente documento como base de su trabajo.

Definiciones, acrónimos, abreviaturas y referencias

A continuación se incluye información relativa a la terminología utilizada en el documento así como a las referencias realizadas en el mismo.

- *SOLE*: Servidor OnLine de Exámenes.
- *MDA* (Model Driven Architecture): Arquitectura dirigida por el modelo.
- *UML* (Unified Modeling Language): Lenguaje unificado de modelado.
- *SOA* (service oriented architecture): Arquitectura orientada a servicios.
- *EIS*: Sistemas de Información Empresariales
- *XMI* (XML Metadata Interchange): Intercambio de datos XML.
- *XSL-FO* (XSL-Formatting Objects): Tecnología XML para el formateo y presentación de datos.
- *XSLT* (Extensible Stylesheet Language Transformations): Tecnología XML para la transformación de datos.

Vistas de Arquitectura del Sistema

Existen diversas formas de representar una arquitectura. Cada una de estas formas de representación se conoce como vista de la arquitectura.

Según el modelo de Kruchten, una vista es “una presentación de un modelo, la cual es una descripción completa de un sistema desde una particular perspectiva”. El modelo más aceptado a la hora de establecer las vistas necesarias para describir una arquitectura software es el modelo 4+1 del propio Kruchten. Este modelo define 4 vistas principales:

1. **Vista Lógica**, modelo de objetos, clases, entidad – relación, etc.
2. **Vista de Proceso**, modelo de concurrencia y sincronización.
3. **Vista de Desarrollo**, organización estática del software en su entorno de desarrollo (librerías, componentes, .ear, .jar, etc.).
4. **Vista Física**, modelo de correspondencia software - hardware (aspectos de distribución en máquinas, por ejemplo)

Una vista más, la "+1", es la que muestra y traza en cada una de las anteriores y que está formada por las necesidades funcionales que cubre el sistema, y que en ocasiones se identifica como vista de "casos de uso".

No todas las arquitecturas de software requieren todas las vistas del modelo 4+1, algunas vistas pueden ser omitidas, pero todos los escenarios se pueden encontrar en ellas. En el presente documento se estudiará la vista lógica, de desarrollo y física.

La finalidad es la de unificar los criterios de modelado, la necesidad de facilitar la comunicación entre los diseñadores y establecer estándares variados, lo más claro posibles que permitan una mayor comprensión del sistema que se esté modelando.

Vista Lógica

La vista lógica soporta el análisis y la especificación de los requisitos funcionales: lo que el sistema debería proporcionar en términos de servicios a sus usuarios. El sistema se descompone en un conjunto de abstracciones clave tomadas mayormente del dominio del problema, en forma de objetos o clases.

El dominio del sistema se puede descomponer en:

1. Persistencia en Bases de Datos
2. Integración con Sistemas Externos
3. Interfaz gráfica de usuario con AJAX.
4. Ejecuciones automáticas.
5. Importación y exportación de datos.
6. Generación de documentos.

Persistencia en Base de Datos

Es necesaria la existencia de un módulo que se encargue de la gestión de los objetos persistentes. Este módulo aporta al sistema las funcionales CRUD (Create, read, update and delete) de las entidades en base de datos. También es el encargado de las funcionalidades de búsqueda de dichas entidades en el sistema de persistencia.

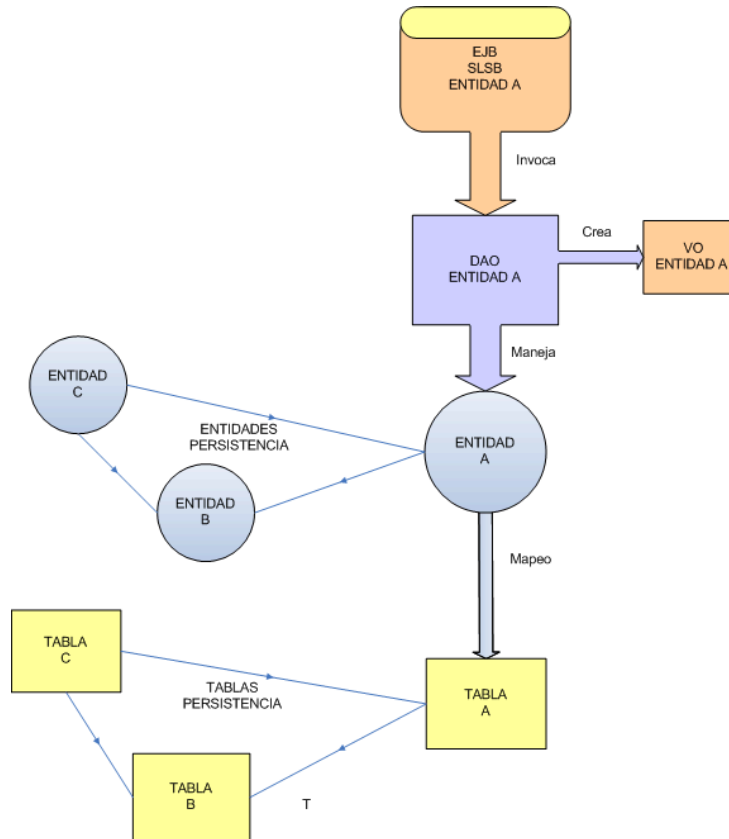


Figura 24. Arquitectura del módulo de Persistencia

El módulo se basa en una arquitectura dirigida por modelos (MDA). Esto permite que mediante el diseño de clases en UML se pueda generar una implementación basada en patrones estándares J2EE e implementada con frameworks tan extendidos como son Hibernate y Spring. La herramienta que facilita este trabajo es AndroMDA, que permite a partir de una definición XMI (XML Metadata Interchange) del diagrama de clases, generar una implementación inicial basada en los frameworks mencionados anteriormente. Por norma general el diseño de este módulo es el que determinará el modelo de tablas de la base de datos.

Integración con sistemas externos

Toda la integración con sistemas externos se agrupa dentro de este módulo. El objetivo de este módulo es separar la lógica propia del proceso de negocio de la integración con otros sistemas. Dicha integración debe resolver la forma en la que se accede a la información o al servicio del sistema externo y transformar la información intercambiada entre el formato necesario para el sistema externo y el formato necesario para el sistema.

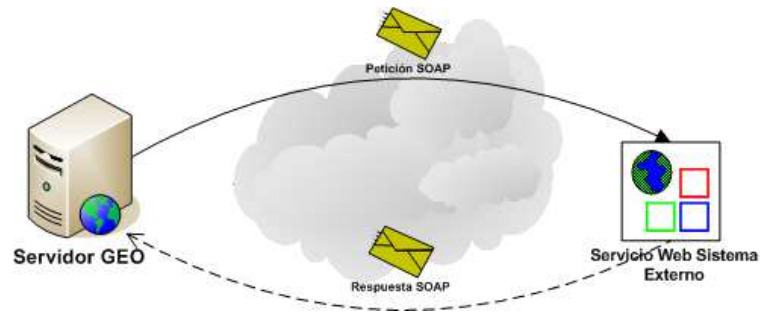


Figura 25. Integración Sistemas Externos con Web Service

El módulo se basa en el uso de Servicios Web para dicha comunicación utilizando el estándar IMS QTI 2.1 embebido en los mensajes SOAP, para intercambiar la información con otros sistemas.

Interfaz gráfica de usuario con AJAX

Los requisitos recogidos en el análisis en torno a la generación, realización y corrección de exámenes dinámicamente, así como otra serie de requerimientos no funcionales, hacen de vital importancia el uso de una interfaz ligera usando tecnología AJAX.

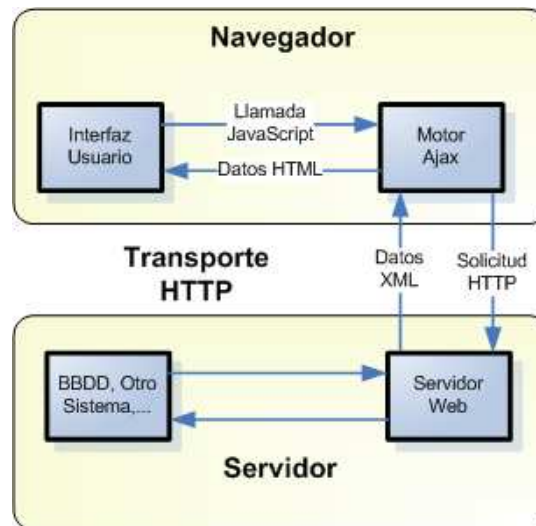


Figura 26. Comunicación Cliente/Servidor Con Ajax

Para la implementación de esta interfaz se utilizará el de framework de Ajax ZK, escrito completamente en Java, de código abierto, que permite el desarrollo de una rica interfaz de usuario para aplicaciones Web sin usar JavaScript y con una programación simple y escasa. Además, mediante ZK permite la generación de gráficos dinámicamente implementando Ajax, lo que nos permitirá responder a aquellos requisitos recogidos en el análisis en relación al mostrado de estadísticas del sistema gráficamente.

Ejecuciones Automáticas programadas

A la hora de realizar exámenes así como en otra serie de tareas, es necesario programar eventos temporales, periódicos o no; que permitan al sistema realizar tareas automáticas en un instante de tiempo concreto.

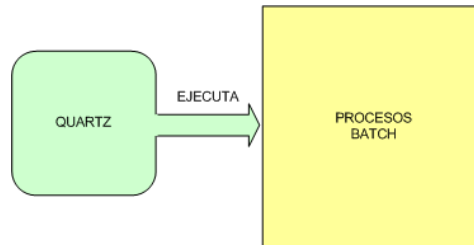


Figura 27. Ejecución Proceso Batch

Dicha tarea será implementada mediante procesos Batch que serán ejecutados por Quartz, herramienta de código abierto que permite ser integrado con cualquier aplicación J2EE de una manera fácil y sencilla.

Importación y exportación de datos

Uno de los puntos más importante recogidos en el análisis es la estandarización de contenidos mediante el estándar IMS QTI v1.2 para el intercambio de datos con sistemas externos. Teniendo en cuenta que dicho estándar está basado en tecnologías de marcado XML se implementará con la api Jaxb.

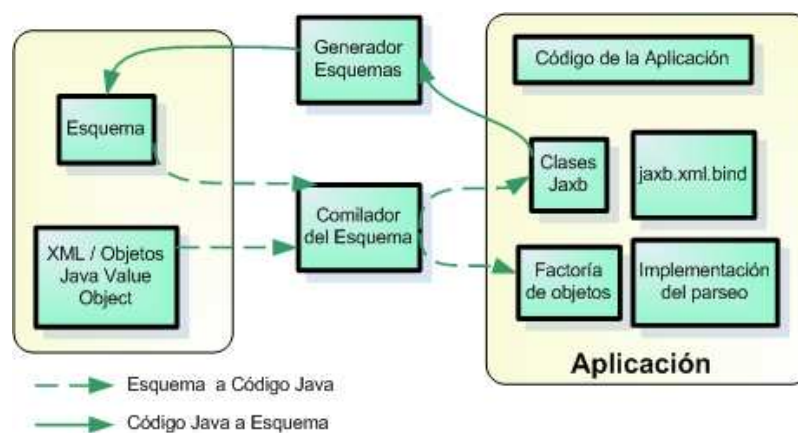


Figura 28. Uso de Jaxb para la importación de datos

Jaxb permite la transformación de código Java a XML y a la inversa, además de generar las clases necesarias para trabajar con código XML utilizando directamente objetos Java.

Generación de documentos

El módulo de generación de documentos es el encargado de generar dinámicamente la documentación a partir de unas plantillas y la información aportada por el sistema.

Los documentos se definen en un formato intermedio, XSL-FO, que independiza del formato de salida final. El procesador de XSL-FO, en nuestro caso Apache FOP, es

el encargado de la transformación al formato final. Apache FOP permite transformar a formatos PDF, RTF, TIFF/PNG, XML y TXT. La generación dinámica del documento XSL-FO se realiza mediante plantillas XSLT que son capaces de generar este documento intermedio a partir de datos XML recuperados del sistema.

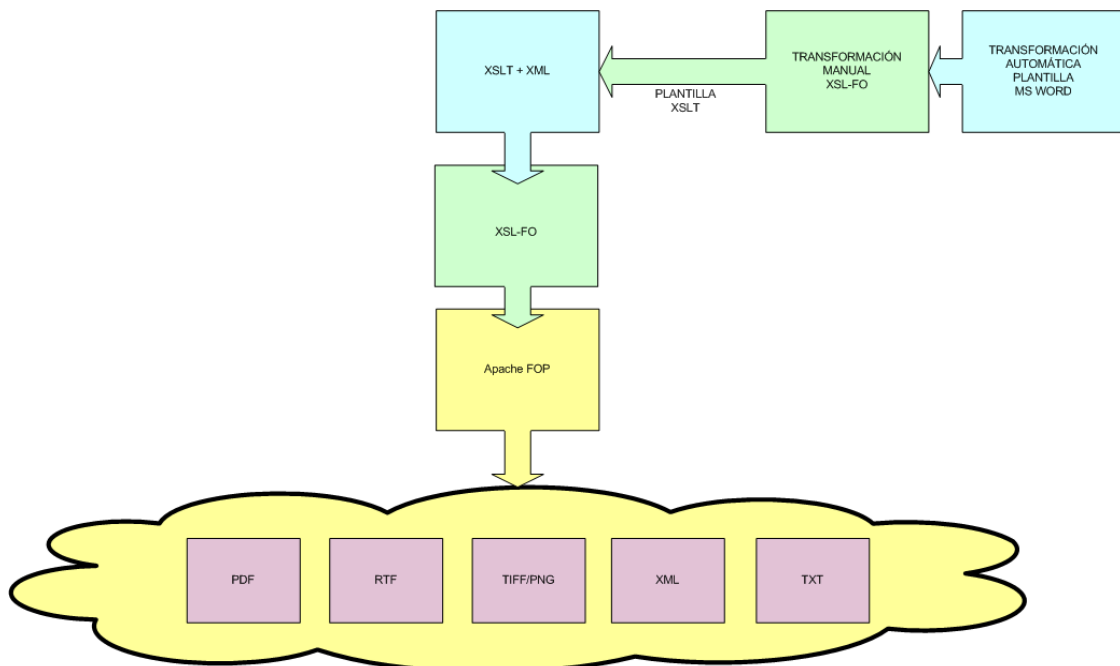


Figura 29. Generación de Documentos

Para los documentos predefinidos en formato PDF Forms esta solución no es la más óptima. Para estos casos la mejor solución es usar la librería iText que permite el manejo de este tipo de documentos.

Vista de Desarrollo

La vista de desarrollo o despliegue se enfoca en la organización de los módulos software en el entorno de desarrollo. El software es empaquetado en pequeños trozos (librerías de programa, subsistemas, componentes, etc.), los subsistemas se organizan en capas jerárquicas, y cada capa proporciona una interfaz bien definida a sus capas superiores

Para describir esta vista, la arquitectura de la aplicación se ha dividido en tres capas:

- Presentación e interfaz.
- Lógica de negocio
- Acceso a datos e integración

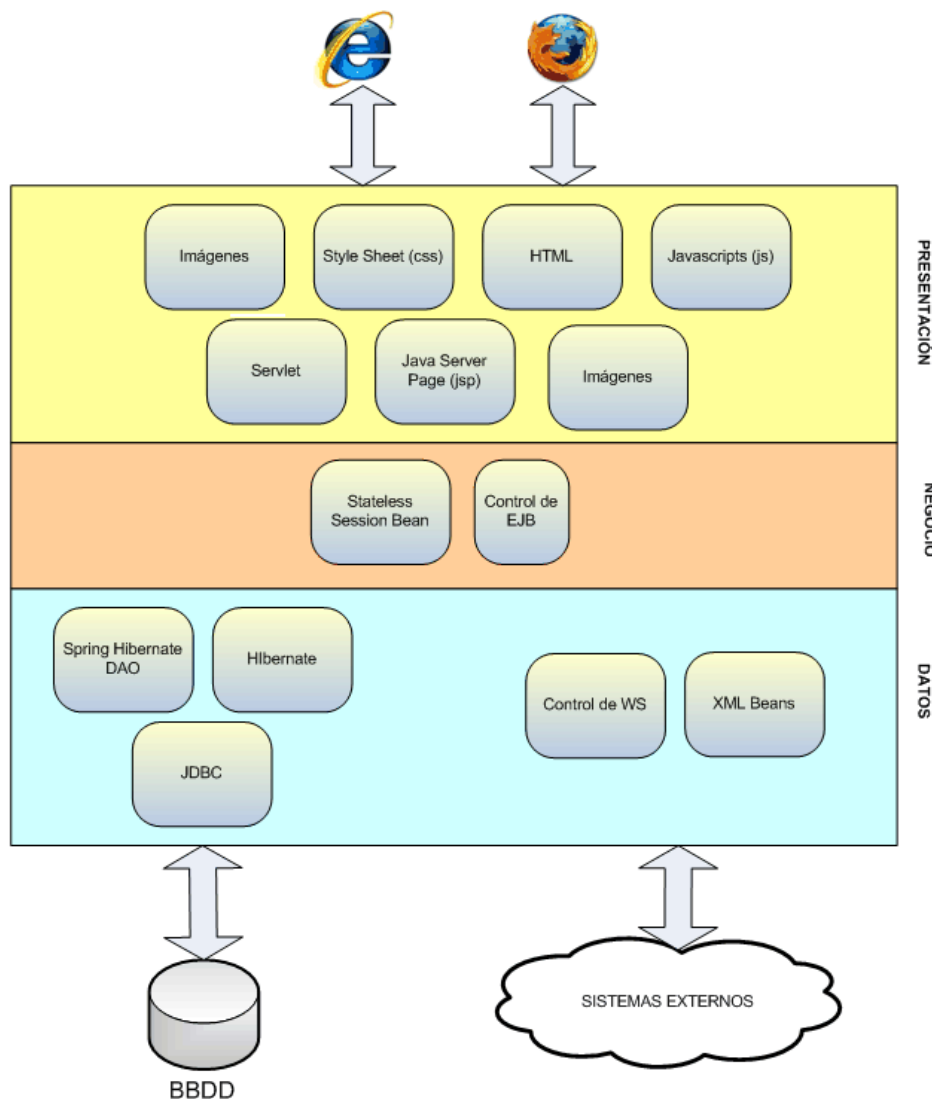


Figura 30. Arquitectura por capas

Cada una de estas capas provee de un conjunto de funcionalidades específicas sobre toda la arquitectura de la aplicación. Dividir y modularizar los componentes de la aplicación mediante estas capas permite que el sistema sea más fácil de mantener. Además en el caso del desarrollo de grandes aplicaciones, permite dividir a los equipos

de desarrollo asignándoles tareas sobre alguna de estas capas, siempre y cuando las interacciones entre las capas estén bien definidas.

No todos módulos de la aplicación necesitarán la existencia de cada una de estas capas, por ejemplo existirán módulos que no tengan porque acceder a la capa de datos al igual que existirán módulos que no tengan capa de presentación ya que solo publicarán una serie de servicios consumidos por el resto.

La aplicación debe ser construida de forma que un componente solo acceda a componentes dentro de su capa o de capas inferiores. Aunque está permitido que los componentes accedan a capas inferiores no adyacentes, es importante recordar que las capas existen por alguna razón y no deben ser saltadas arbitrariamente.

Los componentes de las capas de negocio pueden ser expuestos a través de interfaces programáticas, de forma que la capa de presentación haga uso de esos interfaces como muestra la siguiente figura. La exposición de interfaces permite que la aplicación sea partícipe como proveedor, de una arquitectura orientada al servicio (SOA). Además a través de la capa de acceso a datos e integración se podrán consumir servicios de otros sistemas dentro de SOA.

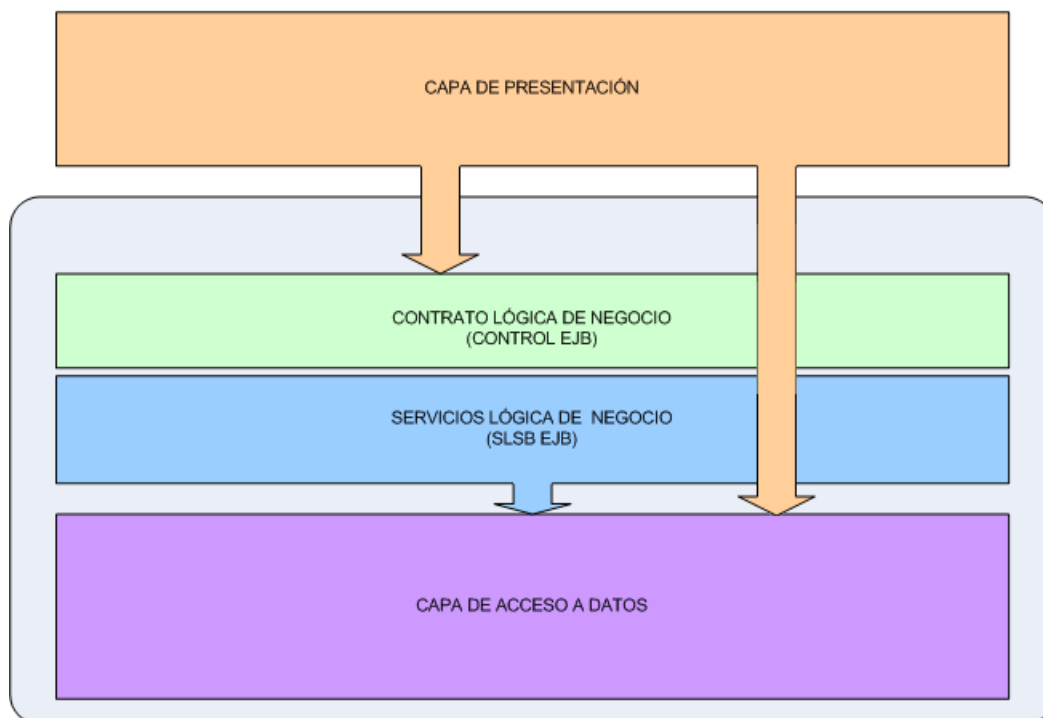


Figura 31. Arquitectura de servicios por capas

Capa de Presentación y de interfaz

La capa de presentación e interfaz es la responsable de proveer de un interfaz con la aplicación. En el sistema GEO, las interacciones de usuario se realizarán mediante un navegador web que se ejecuta de forma remota a la aplicación y que se conecta a ella vía http.

El navegador web proporciona al usuario la forma más sencilla de comunicarse con la aplicación.

La siguiente lista describe cada uno de los componentes mostrados en la capa de presentación e interfaz, la interacción de los componentes con el resto de la aplicación y cualquiera consideración de diseño asociada con el componente.

- **HTML**, las páginas HTML son usadas para proveer de contenido estático.
- **Imágenes**, las imágenes proporcionan contenido estático a una página.
- **Hojas de Estilos**, las hojas de estilos en cascada son usadas para crear estilos consistentes en la aplicación web.
- **Javascript**, añadir código javascript ejecutado en el cliente permite modificar el comportamiento por defecto estático de las páginas.
- **Zk**, el uso de librerías AJAX como ZK mejora el mantenimiento de este tipo de código. Además con el lenguaje zul permite definir e implementar las páginas de contenidos dinámicos con un código claro y sencillo.
- **Servlets**, aunque ya no se usan normalmente, puede que sean necesarios para ejecutar algún tipo de acción que no tenga mejor solución.
- **Java Server Pages (Jsps)**, las páginas Jsps se usan para proporcionar a las páginas de contenido dinámico. Se debe evitar introducir código java directamente en las JSPs. Se recomienda el uso de tag libraries como NETUI. Las Jsps solo deben incluir código para acceder y mostrar información; no deben proveer de lógica de negocio ni tampoco de lógica de navegación de pantallas.

Capa de Negocio

La lógica de negocio se implementa en esta capa. Cuando se procesa una petición, un componente de esta capa valida la información acorde a unas reglas de negocio, realiza la lógica de negocio solicitada, y posiblemente solicita a la capa de persistencia e integración la creación, lectura, modificación o borrado de la entidad externa apropiada.

Los componentes de la lógica de negocio son invocados desde la capa de presentación. La definición de lógica común de varios procesos en esta capa permite su reutilización.

- **Bean de sesión sin estado (SLSB)**, son usados para encapsular lógica de negocio y tienen grandes características de escalabilidad y rendimiento.
- **Control de acceso al EJB**, estos controles son usados para acceder a los beans de sesión y son los encargados de realizar la búsqueda en el “initial context”

Capa de Acceso a datos e Integración

Los componentes de esta capa envían y reciben información desde los sistemas de información empresariales (EIS) externos a la aplicación. El EIS más común es el sistema de gestión de base de datos. Esta capa existe para encapsular la complejidad de la comunicación con los variados EIS.

El acceso a datos de la base de datos se realizará a través del patrón J2EE Data Access Object (DAO), usando Hibernate en vez de JDBC para la implementación.

- **Hibernate:** es una herramienta de Mapeo objeto-relacional para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional y el modelo de objetos de una aplicación.
- **Spring DAO:** ofrece una serie de módulos para simplificar nuestra labor a la hora de programar, en este caso en la creación de DAOs.
- **Cliente servicio web:** permite acceder a servicios web.
- **XMLBeans:** facilita el uso de XML en desarrollo mediante un interfaz amigable.

Vista Física

La vista física muestra el hardware en el que la aplicación se va a instalar. También se describe la relación de despliegue entre las partes de la aplicación y el hardware.

Despliegue Hardware

A continuación se esquematiza el despliegue hardware del entorno de desarrollo y preproducción.



Figura 32. Entorno desarrollo preproducción

Clientes

El acceso a la aplicación se realizará mediante cualquier Navegador Web que soporte el estándar HTTP 1.1.

Capa Web

Su función principal es proveer de una capa de seguridad al sistema, aunque también es útil para funciones de balanceo de carga y distribución de contenidos estáticos.

Capa de aplicación

En esta capa se desplegará la aplicación sobre un servidor de aplicaciones GlassFish. No es recomendable separar el contenedor web del contenedor de EJBs, se producen grandes pérdidas de rendimiento al ser necesario más tráfico de red y más tiempo de proceso para serializar y deserializar la información en la red.

Capa EIS

En esta capa se sitúa el almacenamiento principal de toda la información del dominio de negocio, en nuestro caso la base de datos, y los sistemas proveedores de información externos.

Despliegue Software

En los entornos de desarrollo y reproducción el despliegue de la aplicación se realizará sobre un único servidor, dejando la opción del despliegue en modo cluster para futuras implantaciones cuyos requisitos de rendimiento aconseje dicho uso.

Aplicación

La aplicación se desplegará empaquetada como en un fichero EAR (geo_Vx_y_ear.ear), donde se incluirán todos los servicios y la aplicación web. En conjunto a este fichero se desplegarán varias librerías web del proyecto compartidas con cada uno de los módulos web del sistema.

Librerías

El despliegue de librerías de soporte (COTS) se realizará mediante el uso de librerías compartidas del servidor.

- geo_common_lib_v1.0.ear
- geo_andromda_lib_v3.3.ear
- geo_documents_lib_v1.0.ear
- geo_hibernate_lib_v3.2.ear
- geo_spring_lib_v2.5.ear

Arquitectura y estándares de desarrollo

A continuación se definen los estándares técnicos y de nomenclatura, normas y recomendaciones. Estos estándares están relacionados con la adopción o diseño de la arquitectura o infraestructura tecnológica, y condicionan el diseño o la construcción del sistema.

Idioma

Dado el carácter colaborativo internacional del que se quiere dotar al proyecto en el futuro, se elige el inglés como idioma a utilizar, en la medida de lo posible, para el nombrado y codificación del sistema. Se excluye de esta norma la implementación de la capa de presentación, cuyo lenguaje original será el castellano.

Catálogo de Excepciones

En este apartado se definen los comportamientos no habituales del sistema, que reflejan situaciones anómalas o secundarias en el funcionamiento y ejecución del sistema.

<i>Modulo</i>	<i>Excepciones</i>
General	<i>Introducción de datos no válidos por el usuario</i> <i>Problemas con datos necesarios en la lógica de negocio</i>
Documentos	<i>Problemas en la generación de documentación.</i>
Interfaces	<i>Problemas en el acceso al sistema externo</i> <i>Problemas en la transformación de los datos</i>
Persistencia	<i>Problemas en el acceso a la BBDD</i> <i>Problemas por restricciones de la BBDD</i>

Tabla 11. Excepciones

Normas para extensiones de ficheros

En la siguiente tabla se enumeran las extensiones correctas para los tipos de ficheros identificados como necesarios en el desarrollo del sistema.

<i>Tipo de fichero</i>	<i>Extensión recomendada</i>
XML	<i>.xml</i>
Schema XML	<i>.xsd</i>
Plantilla XSLT	<i>.xsl</i>
Propiedades de configuración	<i>.properties</i>
Javascript	<i>.js</i>
Hojas de estilos	<i>.css</i>
Librerías y de aplicación (j2ee)	<i>.ear, .war, .jar</i>
JSP	<i>.jsp</i>
ZUL	<i>.zul</i>
HTML	<i>.html</i>

Tabla 12. Extensión de ficheros

Normas para nombrado de ficheros

En general es importante dar nombres significativos a los ficheros de la aplicación. Además en algunos casos es importante que en el fichero se incluya información de la versión de dicho fichero.

En la **Error! Reference source not found.** se especifican los formatos para algunos de los ficheros más característicos del sistema.

<i>Tipo de fichero</i>	<i>Formato del nombre</i>
Ficheros de propiedades	<i>geo_ "módulo".properties</i>
Ficheros de librerías y de aplicación	<i>geo_ "nombre" _Vx_y.ear</i>
	<i>geo_ "nombre" _Vx_y.war</i>
	<i>geo_ "nombre" _Vx_y.jar</i>

Tabla 13. Nombrado de ficheros

Normas para nombrado de paquetes

El paquete base para todo el proyecto será **org.babieca.geo**. Dentro de este paquete se subdividirá en otros paquetes por módulos de diseño:

<i>Módulos</i>	<i>Paquetes</i>	
<i>Interfaz Gráfica</i>	<i>org.babieca.geo</i>	<i>gui</i>
<i>Persistencia</i>		<i>dadb</i>
<i>Timers</i>		<i>timers</i>
<i>Integración y estandarización</i>		<i>interfaces</i>
<i>Documentos</i>		<i>documents</i>

Tabla 14. Definición de paquetes

Nombrado de Clases

Se aplicarán reglas de nombrado para las clases que tengan que cumplan un patrón de diseño, o que debido a la tecnología que empleen sea necesario identificarlas fácilmente.

<i>Tipo de Instancia</i>	<i>Nombre</i>
<i>Fachadas de servicio</i>	<i>XxxxxService</i>
<i>DAO</i>	<i>XxxxxDAO</i>
<i>VO</i>	<i>XxxxxxValueObject</i>

Tabla 15. Nombres de Clases

Estándares de codificación Java

Se seguirá el estándar de codificación definido por Sun, en el documento “Java Code Conventions” publicado en septiembre de 1997, accesible desde la siguiente página Web: <http://java.sun.com/docs/codeconv/CodeConventions.pdf>

Nomenclatura en la BBDD

Puesto que la interacción con la base de datos viene definida e implementada por Hibernate, únicamente debe tenerse en cuenta el uso del prefijo “GEO_” en el nombrado de las tablas que componen el modelo del sistema.

Formato de trazas

Es necesario normalizar el formato en el que se generarán las trazas del sistema. El formato propuesto es el siguiente:

<p><nivel de log> <fecha horas:minutos:segundos> <aplicación> <Módulo> <submódulo> <Mensaje de traza> <datos relevantes> <clase> <nº de línea></p>
--

donde:

<nivel de log> es el nivel de log de entre los existentes, p.e ERROR, WARN...
<fecha horas:minutos:segundos> presenta el momento en el que se generó la traza.
<aplicación> indica la aplicación a la que pertenece la traza, en nuestro caso GEO.
<módulo> módulo de la aplicación de entre los existentes.
<submódulo> submódulo de la aplicación si aplica.
<Mensaje de traza> Mensaje claro y conciso del motivo de la traza.
<datos relevantes> datos que puedan ayudar a interpretar la traza.
<clase> Clase desde la que se escribe la traza
<nº de línea> de la clase desde la que se escribe la traza.

Existirá un fichero de traza para cada uno de los módulos de diseño identificados.

Patrones de diseño

En el diseño del sistema se recomienda, en la medida de lo posible, el empleo de patrones de diseño existentes entre los patrones GoF (Gang of Four) y los patrones J2EE.

Entornos Hardware

A continuación se detallan los detalles de las máquinas donde se desplegará la aplicación para realizar las distintas pruebas funcionales y de integración en las distintas tareas de desarrollo del sistema. Resaltar que por motivos de rendimiento, en todos los entornos la aplicación será desplegada en una única máquina, tanto el gestor de BBDD como todo aquel software necesario para su puesta en marcha.

Máquina de Desarrollo

Durante el desarrollo de una iteración no se contempla el despliegue público (accesible desde el exterior de la red de desarrollo) de la aplicación. El sistema correrá en el entorno de desarrollo local del desarrollador.

<i>Máquina de Desarrollo</i>	
Nombre del Servidor:	<i>bbi.babioca.org</i>
Arquitectura:	<i>Intel Centrino</i>
URL:	<i>-</i>
Memoria RAM:	<i>4 Gbytes</i>
Sistema Operativo:	<i>Windows Vista Enterprise</i>
Servidor Web:	<i>[No aplica]</i>
Servidor de Aplicaciones:	<i>Jboss 5.0.1</i>
Gestor de BBDD:	<i>MySQL 5.1.32</i>

Tabla 16. Datos de la máquina de desarrollo

Pre producción

El sistema correrá en una máquina de la red de desarrollo, con acceso público por medio de una conexión susceptible de caídas.

<i>Máquina de Desarrollo</i>	
Nombre del Servidor:	<i>desarrollo.babioca.org</i>
Arquitectura:	<i>Intel Core Duo</i>
URL:	<i>http://geo.desarrollo.babioca.org</i>
Memoria RAM:	<i>2 Gbytes</i>
Sistema Operativo:	<i>Linux 2.6.24</i>
Servidor Web:	<i>[No aplica]</i>
Servidor de Aplicaciones:	<i>Sun GlassFish 2.1</i>
Gestor de BBDD:	<i>MySQL 5.1.34</i>

Tabla 17. Datos de la máquina de preproducción

Producción

El sistema será desplegado en una máquina externa a la red de desarrollo, en un entorno seguro bajo una red segura de errores.

<i>Máquina de Desarrollo</i>	
Nombre del Servidor:	<i>babieca.org</i>
Arquitectura:	<i>Intel Core Duo (IBM server SerieX)</i>
URL:	<i>http://geo.babieca.org</i>
Memoria RAM:	<i>1 Gbytes</i>
Sistema Operativo:	<i>Linux 2.6.20</i>
Servidor Web:	<i>[No aplica]</i>
Servidor de Aplicaciones:	<i>Sun GlassFish 2.1</i>
Gestor de BBDD:	<i>MySQL 5.0.27</i>

Tabla 18. Datos de la máquina de producción

Pila tecnológica

En la presente sección se expondrán y describirán todas aquellas herramientas y frameworks utilizados durante el proceso de desarrollo del sistema, agrupadas por el papel que desempeñan dentro del propio proceso.

Herramientas de modelado y generación de código

Las herramientas de modelado nos permiten realizar los diseños de las aplicaciones siguiendo estándares como UML para que posteriormente podamos pasar el modelo a código a través de herramientas de generación automáticas.

MagicDraw UML Community Edition

MagicDraw UML es un programa para ser utilizado como modelado de UML. Provee soporte completo para meta modelos UML 2.0, incluyendo diagramas de clases, casos de uso, comunicación, secuencia, estado, actividad, implantación, paquetes, componentes, estructuras compuestas y de distribución. Adicionalmente, MagicDraw provee soporte explícito para perfiles UML y diagramas personalizables dando significado al modelamiento visual de arquitecturas.

MagicDraw genera automáticamente sus partes del modelo, todo de acuerdo con los patrones de diseño establecidos: GoF, Java, EJB, JUnit, Esquemas XML, WSDL, CORBA, IDL y cualquier otro patrón personalizable.

Además tiene infinidad de características más que pueden consultarse en la página oficial de la aplicación.

AndroMDA

AndroMDA es un framework MDA Open Source, cuya función principal es la de generar código a partir de un modelo, generalmente UML en formato XMI producido por herramientas de modelado. Existen gran cantidad de plugins de AndroMDA (cartuchos y librerías) a través de los cuales se puede generar diversos componentes en forma de código para un gran número de lenguajes entre los que se encuentra Java.

A pesar de que AndroMDA puede ser utilizado en cualquier aplicación, se suele utilizar más en aplicaciones con tecnología Java EE. AndroMDA puede crear un nuevo proyecto Java EE desde cero, en donde el código es generado a partir de un modelo realizado en UML. Entre las grandes posibilidades que ofrece AndroMDA está la de generar el código para Hibernate, EJB, Spring, WebServices y Struts; integrando el código generado automáticamente al proceso de construcción. AndroMDA es muy eficiente generando código para la aplicación a partir de un modelo, generando la columna vertebral del proyecto, lo que permite que los desarrolladores se mantengan enfocados a la lógica de negocio.

Herramientas de desarrollo

A pesar de que la mayoría de herramientas descritas en esta sección podrían clasificarse como herramientas de desarrollo, únicamente se han incluido en este punto aquellas utilizadas para la edición de código, compilación, y depuración ya sea en un entorno visual o no.

Java Enterprise Edition 5 SDK

Java Platform, Enterprise Edition o Java EE (anteriormente conocido como Java 2 Platform, Enterprise Edition o J2EE hasta la versión 1.4), es una plataforma de programación—parte de la Plataforma Java—para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

La plataforma Java EE está definida por una especificación similar a otras especificaciones del Java Community Process. Java EE es también considerada informalmente como un estándar debido a que los suministradores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son conformes a Java EE.

Java EE incluye varias especificaciones de API, tales como JDBC, RMI, e-mail, JMS, Servicios Web, XML, etc y define cómo coordinarlos. Java EE también configura algunas especificaciones únicas para Java EE para componentes. Estas incluyen Enterprise JavaBeans (EJB), servlets, portlets (siguiendo la especificación de Portlets Java), JavaServer Pages (JSP) y varias tecnologías de servicios web. Esto permite al desarrollador crear una Aplicación de Empresa portable entre plataformas y escalable, a la vez que integrable con tecnologías anteriores. Otros beneficios añadidos son, por ejemplo, que el servidor de aplicaciones puede manejar transacciones, la seguridad, escalabilidad, concurrencia y gestión de los componentes desplegados, significando que los desarrolladores pueden concentrarse más en la lógica de negocio de los componentes en lugar de en tareas de mantenimiento de bajo nivel.

En su versión 5, la plataforma Java EE pasa a ser el estándar de la industria para la aplicación de la arquitectura orientada a servicios (SOA) así como de la próxima generación de aplicaciones Web. Se centra en hacer más fácil el desarrollo conservando la riqueza de la plataforma J2EE 1.4. Además ofrece nuevas y actualizadas funciones, como Enterprise JavaBeans (EJB) 3.0, tecnología JavaServer Faces (JSF), y la última API de servicios Web.

Eclipse Ganymede

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar aplicaciones. Desarrollado originalmente por IBM, actualmente es desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

La versión actual de Eclipse dispone de un gran número de características como son: editor de texto, resaltado de sintaxis, pruebas unitarias con JUnit, control de versiones CVS, integración con Ant, asistentes para la creación de proyectos, refactorización, etc.

Asimismo, existen multitud de plugins para diversas tareas como la integración con Hibernate y Subversión que permiten extender la funcionalidad del entorno.

ZK

ZK es un framework AJAX, escrito completamente en Java, de código abierto, que permite el desarrollo de una rica interfaz de usuario para aplicaciones Web sin usar JavaScript y con una programación simple y escasa.

El núcleo de ZK es un mecanismo conducido por eventos basado en AJAX, sustentado sobre 70 componentes XUL y 80 componentes XHTML, y un lenguaje de marcación para diseñar interfaces de usuario. Los programadores diseñan las páginas de su aplicación en componentes XUL/XHTML con una gran cantidad de características, y los manipulan con eventos disparados por la actividad del usuario final. Es similar al modelo de programación encontrado en las aplicaciones basadas en GUI de escritorio.

ZK utiliza el acercamiento llamado centrado-en-el-servidor para la sincronización de componentes, el pipelining entre clientes y servidores se hace automáticamente por el motor, y los códigos de Ajax son completamente transparentes para los desarrolladores de aplicaciones Web. Por lo tanto, los usuarios finales obtienen una interacción y respuesta similar a las de una aplicación de escritorio, mientras que la complejidad del desarrollo es similar a la que tendría la codificación de aplicaciones de escritorio.

Además de la programación basada en componentes, de manera similar a Swing, ZK soporta un lenguaje de marcación para la definición de una potente interfaz de usuario llamada ZUML.

- ZUML está diseñado para que desarrolladores no expertos diseñen interfaces de usuario de forma eficiente.
- ZUML permite a un desarrollador mezclar diferentes tipos de lenguaje de marcación, tales como el lenguaje XUL de Mozilla y XHTML, todos ellos en la misma página.
- ZUML permite a los desarrolladores embeber scripts en lenguaje Java (interpretado por BeanShell) y usar expresiones para manipular los componentes y acceder a los datos.

El desarrollo con ZK estará integrado en el IDE Eclipse, a través de un plugin de dicho entorno llamado ZK Studio que forma parte del propio desarrollo de ZK.

Frameworks y API's de desarrollo

Además de las herramientas de desarrollo descritos anteriormente, se han utilizado una serie de API's y frameworks de desarrollo en Java para facilitar algunas tareas.

Struts

Struts es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma Java EE. Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache Struts.

Struts se basa en el patrón del Modelo-Vista-Controlador (MVC). Cuando se programan aplicaciones Web con el patrón MVC con un único controlador, nos encontramos con un grave problema, ya que nuestro controlador se convierte en lo que se conoce como "fat controller", es decir un controlador de peticiones, Struts surge como la solución a este problema ya que implementa un solo controlador (ActionServlet) que evalúa las peticiones del usuario mediante un archivo configurable (struts-config.xml).

Struts permite reducir el tiempo de desarrollo. Su carácter de "software libre" y su compatibilidad con todas las plataformas en que Java Enterprise esté disponible, lo convierte en una herramienta altamente disponible.

Spring

El Spring Framework (también conocido simplemente como Spring) es un framework de código abierto de desarrollo de aplicaciones para la plataforma Java.

A pesar de que Spring Framework no obliga a usar un modelo de programación en particular, se ha popularizado en la comunidad de programadores en Java al considerársele una alternativa y sustituto del modelo de Enterprise JavaBean. Por su diseño el framework ofrece mucha libertad a los desarrolladores en Java y soluciones muy bien documentadas y fáciles de usar para las prácticas comunes en la industria.

Mientras que las características fundamentales de este framework pueden emplearse en cualquier aplicación hecha en Java, existen muchas extensiones y mejoras para construir aplicaciones basadas en Web por encima de la plataforma empresarial de Java (Java Enterprise Platform).

Hibernate

Hibernate es una herramienta de mapeo objeto-relacional para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) que permiten establecer estas relaciones.

Como todas las herramientas de su tipo, Hibernate busca solucionar el problema de la diferencia entre los dos modelos de datos coexistentes en una aplicación: el usado en la memoria de la computadora (orientación a objetos) y el usado en las bases de datos (modelo relacional). Para lograr esto permite al desarrollador detallar cómo es su modelo de datos, qué relaciones existen y qué forma tienen. Con esta información Hibernate le permite a la aplicación manipular los datos de la base operando sobre objetos, con todas las características de la POO. Hibernate convertirá los datos entre los tipos utilizados por Java y los definidos por SQL. Hibernate genera las sentencias SQL y libera al desarrollador del manejo manual de los datos que resultan de la ejecución de dichas sentencias, manteniendo la portabilidad entre todos los motores de bases de datos con un ligero incremento en el tiempo de ejecución.

Hibernate está diseñado para ser flexible en cuanto al esquema de tablas utilizado, para poder adaptarse a su uso sobre una base de datos ya existente. También tiene la funcionalidad de crear la base de datos a partir de la información disponible.

Hibernate ofrece también un lenguaje de consulta de datos llamado HQL (Hibernate Query Language), al mismo tiempo que una API para construir las consultas programáticamente (conocida como "criteria").

Hibernate para Java puede ser utilizado en aplicaciones Java independientes o en aplicaciones Java EE, mediante el componente Hibernate Annotations que implementa el estándar JPA, que es parte de esta plataforma.

JAXB

JAXB es una API de Java para el tratamiento de información XML proporcionando un mapeo directo entre XML y los objetos Java. Dado un esquema, que especifica la estructura de los datos XML, el compilador JAXB genera un conjunto de clases de Java que contienen todo el código para analizar los documentos XML basados en el esquema. Una aplicación que utilice las clases generadas puede construir un árbol de objetos Java que representa un documento XML, manipular el contenido del árbol, y regenerar los documentos del árbol, todo ello en XML sin requerir que el desarrollador escriba código de análisis y de proceso complejo.

JAXB tiene infinidad de ventajas entre las que se encuentra, el uso de tecnología Java y XML, garantía de datos válidos, rapidez y facilidad de uso, restricción de datos, personalizable y altamente extensible.

JQTI

Básicamente JQTI es un intérprete para el estándar IMS QTI v2.1 el cual está desarrollado sobre XML. QTI es bastante diferente a la mayoría de estándares sobre XML, ya que no sólo contiene datos, sino también contiene instrucciones para el procesamiento de estos datos. Estas instrucciones, que hacen referencia a las evaluaciones en sí, especifican básicamente cómo han de presentarse las preguntas, así como los contenidos evaluados. De hecho podrías definir la especificación QTI como un propio lenguaje de programación, cuya especificación viene dada por el propio documento XML.

El intérprete JQTI incluye una serie de características que lo hace realmente interesante para el procesamiento de la información bajo el estándar QTI: incluye apoyo para leer, analizar e interpretar los documentos XML QTI, así como para la creación de nuevos contenidos en QTI sobre XML; además de permitir la validación y el informe de errores de los documentos en QTI.

Por último resalta la distribución de JQTI bajo la licencia BSD lo que permite usarlo libremente desde otros desarrollos como el de GEO.

Log4j

Log4j es una biblioteca open source desarrollada en Java por la Apache Software Foundation que permite a los desarrolladores de software elegir la salida y el nivel de granularidad de los mensajes o “logs” a tiempo de ejecución y no a tiempo de compilación como es comúnmente realizado. La configuración de salida y granularidad de los mensajes es realizada a tiempo de ejecución mediante el uso de archivos de configuración externos.

Log4j tiene una serie de características que hace realmente interesante su uso para la generación de ficheros de control y monitorización del sistema. Incluye seis niveles de prioridad de mensaje, se pueden configurar varias salidas de texto simultáneamente, incluso configurar nuestra propia salida tipo servidor remoto, base de datos etc. además es totalmente configurable desde un fichero XML lo que hace que su configuración sea altamente flexible y con una gran facilidad de uso gracias a la API que dispone.

Quartz

Quartz es una librería de código abierto para la programación de la ejecución de tareas en Java de una manera realmente sencilla. La principal ventaja de Quartz es que permite definir una serie de condiciones especiales y complejas para definir el momento de ejecución de la tarea.

Herramientas de gestión de datos

Uno de los principales puntos en todo sistema es la persistencia de datos. Como en la mayoría de aplicaciones, para el presente sistema se hará uso de un gestor relacional de base de datos encargado de almacenar los datos de la aplicación en tablas relacionales.

MySQL

MySQL es un gestor de base de datos sencillo de usar y increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales.

Las características principales de MySQL son las siguientes:

- Es un gestor de base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.
- Es Open Source. El código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL para aplicaciones no comerciales.
- Es una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet.
- Existe una gran cantidad de API's para todos los lenguajes (incluido Java) que facilita en gran medida la comunicación con el gestor.

Servidor de Aplicaciones

Tal y como se ha comentado insistentemente, el sistema a desarrollar se trata de una aplicación Web por lo que otro punto importante en la solución tecnológica es la elección del servidor de aplicaciones sobre el que se desplegará la aplicación. A pesar de que en un primer momento se pensó en la utilización de Tomcat por su simplicidad, facilidad de uso y extensión entre los desarrolladores; se abandonó la idea una vez que se decidió la utilización de EJB's para la implementación de los servicios de negocio del sistema ya que Tomcat únicamente es un contenedor de Servlets y no de EJB's.

Glass Fish

GlassFish es un servidor de aplicaciones que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación, tomando como base de desarrollo el Application Server de Sun

Microsystems. Es gratuito, de código libre y se distribuye bajo la licencia CDDL y la GNU GPL.

Glash Fish implementa la plataforma Java EE 5, por lo que soporta las últimas versiones de tecnologías como: JSP, JSF, Servlets, EJBs, Java API para Servicios Web (JAX-WS), Arquitectura Java para Enlaces XML (JAXB), Metadatos de Servicios Web para la Plataforma Java 1.0, y muchas otras tecnologías. El servidor Glash Fish está rodeado por una gran comunidad de desarrolladores con el mismo nombre, que están en continuo desarrollo detectando y corrigiendo errores a la vez que dotan al servidor de nuevas funcionalidades.

Herramientas de colaboración

Unas de las herramientas más importantes y muchas veces olvidadas en todo desarrollo software, son las herramientas de gestión de proyectos y colaboración. Este tipo de herramientas facilitan el trabajo en entornos colaborativos, además de facilitar el acceso al código y documentación generada.

SVN / Tortoise

Subversión es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS, el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

Como clientes de subversión se utilizará Subclipse plugin integrado con Eclipse, así como TortoiseSVN para ser utilizado visualmente integrado con el sistema operativo.

dotProject

dotProject es una herramienta orientada a la Gestión de Proyectos. Para eso se orienta a la administración de recursos para desarrollar un producto, cuya producción requiera de un conjunto de actividades o tareas que se desarrollen entre ellas en forma paralela o independiente.

La aplicación consta de un conjunto de entidades ordenadas jerárquicamente las cuales permiten brindar la funcionalidad del producto.

Entornos de Desarrollo

En la presente sección se describirá brevemente el entorno hardware y software que rodeará a las tareas de desarrollo del sistema.

Equipamiento y software base

Los puestos de desarrollo serán PCs portátiles, equipados como mínimo con procesadores Pentium 4 y 2 Gb de memoria. También es recomendable que tanto la pantalla como la tarjeta gráfica soporten resoluciones de pantalla de 1280x1024.

El sistema operativo sobre el que se desarrollará y sobre el que cada desarrollador realizará sus pruebas unitarias es Windows Vista Enterprise.

Herramientas de desarrollo

En desarrollo del núcleo del proyecto se realizará con el IDE Eclipse en su versión Ganymede para aplicaciones JEE. Este entorno permite la instalación de plugins gratuitos para realizar diversas tareas específicas. Así por ejemplo, es recomendable la instalación de ZK Studio para la integración del IDE con el framework.

En los casos en los que no existe una dependencia con la plataforma se pueden usar herramientas adicionales, por ejemplo, Aptana Studio se usará como herramienta para la edición del javascript en caso necesario.

Existirán tareas repetitivas que se puedan definir dentro de scripts tareas “ant” o tareas que solo tengan su implementación accesible desde este tipo de tareas. Para estos casos será necesario gestionar la integración de Eclipse con ant.

En otros casos, como en el caso del desarrollo de la capa de persistencia es necesario el uso de scripts Maven que automatizan sus tareas más comunes, como son la creación de la base de datos, la generación del código fuente, su compilación y su empaquetado.

Herramientas de soporte al desarrollo

El control de versiones software se realizará mediante un servidor de Subversion, y su cliente TortoiseSVN.

Para usar subversion desde el IDE de desarrollo Eclipse es necesario añadir un plugin. En principio se recomienda el uso de Subclipse. El uso del plugin es complementario al cliente TortoiseSVN.

Las pruebas de la capa de presentación se realizan directamente sobre los navegadores IExplorer y/o Firefox. Dado el comportamiento dinámico de las páginas implementado mediante javascript, es necesario disponer de herramientas para la depuración del código dentro del navegador. IExplorer incluye un depurador, además de la posibilidad de instalar una barra de herramientas de desarrollo. Las herramientas Firefox son más avanzadas, se puede usar por ejemplo Firebug.

La herramienta con la que se realizan tanto tareas de mantenimiento de la base de datos como consultas sobre las tablas del sistema es MySQL Query Browser.

Para intercambiar ficheros con el servidor de desarrollo es necesario un cliente FTP. Un cliente FTP gratuito es Filezilla.

El formato de salida para los documentos es PDF. Es necesario el cliente de Adobe, Acrobat Reader para visualizar los documentos.

Repositorio de código fuente

El repositorio de código fuente se encuentra en el servidor subversion que se encuentra en la dirección **172.16.0.128**, dentro del repositorio **babieca-src**.

La versión en desarrollo se encontrará dentro del directorio “trunk”. Existen otros dos directorios principales, brach y tag. Branch se usa para controlar otras ramas de desarrollo diferentes a la principal, por ejemplo para mantener controlada la versión de maqueta inicial. El directorio tag mantiene copias puntuales del desarrollo principal.

Validación del Documento

Próximos Hitos e Incidencias

ID	Tema	Resolución	Responsable	Fecha Objetivo	Fecha de Impacto
3	Definición servicios SOLE		Diego Jiménez	15/04/09	18/05/09
4	Desarrollo iteración 1		Borja Blanco	01/05/09	18/05/09

Hitos e Incidencias Cerrados

ID	Tema	Resolución	Responsable	Fecha Objetivo	Fecha Cierre
1	Análisis y Planificación	Finalizado	Borja Blanco	01/04/09	01/04/09
2	Arquitectura de Sistema	Finalizado	Borja Blanco	10/04/09	09/05/09

Aprobación del Documento

Rol	Nombre	Firma	Fecha
Autor del Proyecto	Borja Blanco Iglesias		
Tutor del Proyecto	Ricardo Colomo Palacios		

Documento Técnico de la Iteración 1



**Sistema de Gestión de
Exámenes Online
GEO**

**Informe Técnico sobre
Iteración 1
Madrid, 25 de Mayo de 2009**

Autor	Borja Blanco Iglesias
Fecha de Creación	25/05/2009
Fecha de actualización	25/05/2009
Nombre de documento	IT1_GEO.doc
Versión	1.0

Histórico de Versiones

Fecha	Autor	Versión	Comentarios
25/05/2009	Borja Blanco	1.0	Versión inicial del documento.

Revisores

Nombre	Posición
Borja Blanco Iglesias	Autor del Proyecto
Ricardo Colomo Palacios	Tutor del Proyecto

Lista de distribución

Copia N°	Nombre	Posición
1.	Borja Blanco Iglesias	Autor del Proyecto
2.	Ricardo Colomo Palacios	Tutor del Proyecto
3.	Diego Jiménez López	Colaborador
4.		
5.		

Introducción

El Informe técnico de una iteración del desarrollo de un sistema bajo la metodología XP, ofrece una visión global sobre el análisis realizado así como las decisiones de diseño e implementación que se han tomado a lo largo del presente desarrollo.

Objetivos del Documento

El documento tiene como objetivo principal ofrecer una visión global sobre el estado actual del proyecto, enfocado hacia las tareas realizadas sobre la iteración uno en todo el periodo de desarrollo.

A lo largo del documento se detallarán aquellos aspectos funcionales que contempla el desarrollo realizado como consecuencia de la iteración, así como las diversas decisiones de diseño tomadas a lo largo del desarrollo. Además, se incluirán los resultados obtenidos del proceso de pruebas realizado.

Por último, cabe resaltar la inclusión en el documento, de una revisión de la planificación realizada en función a los tiempos de desarrollo obtenidos, estableciendo las comparaciones oportunas con los tiempos planificados.

Audiencia

Debido al carácter técnico, el presente documento está dirigido a aquellas personas que tomen parte en tareas de desarrollo del proyecto y necesiten acceso a la documentación sobre las tareas realizadas en la iteración uno. Así mismo, por la información general tratada en el mismo, la audiencia del documento se extiende a la dirección del proyecto quien deberá validar los requisitos definidos como base del desarrollo.

Definiciones, acrónimos, abreviaturas y referencias

A continuación se incluye información relativa a la terminología utilizada en el documento así como a las referencias realizadas en el mismo.

- **EJB:** Acrónimo del inglés *Enterprise Java bean* es una API dentro del estándar JEE para la comunicación remota de servicios.
- **MDA:** Acrónimo del inglés *Model driven architecture* cuyo significado es arquitectura dirigida por el modelo.
- **SOLE:** Servidor OnLine de Exámenes.
- **XP (Extreme Programming):** Metodología de desarrollo ágil cuya traducción al español es Programación Extrema.

Definición del alcance del desarrollo

Dentro del ámbito del proyecto la presente iteración contempla el desarrollo de los siguientes módulos:

Gestión de Usuarios

Módulo para la gestión de los usuarios. Para cada usuario deben almacenarse los siguientes datos:

- Identificador del usuario.
- Nombre único de usuario.
- Password de usuario
- Correo electrónico.

El sistema debe permitir la adición, eliminación, modificación y consulta de los datos relativos a cada usuario.

Gestión de Administradores

A través de éste módulo se pueden dar de alta, modificar y eliminar administradores del sistema. Para que un administrador pueda gestionar otros administradores, debe ser súper usuario. Los datos de un administrador son los siguientes:

- Identificador que permita identificar un administrador unívocamente.
- Aquel usuario al que se asocia el administrador.
- Marcador para saber si el administrador es súper usuario o no.
- Datos autenticación servidor remoto. Nombre de usuario y contraseña del servidor remoto SOLE.

Gestión de Profesores

Módulo para la gestión de los profesores. Para cada profesor deben almacenarse los siguientes datos:

- Identificador del profesor.
- Usuario asociado al profesor.
- Nombre y apellidos del profesor.
- Dirección.
- Teléfono.
- Datos autenticación servidor remoto. Nombre de usuario y contraseña del servidor remoto SOLE.
- Asignaturas que imparte el profesor.

El sistema debe permitir la adición, eliminación, modificación y consulta de los datos relativos a cada profesor.

Gestión de alumnos

Módulo para la gestión de los alumnos. Para cada alumno los administradores deben gestionar los siguientes datos que serán visibles por los profesores que impartan alguna asignatura en la que esté matriculado el alumno.

- Identificador del alumno.
- Usuario asociado al alumno.
- Nombre y apellidos del alumno.
- Dirección.
- Teléfono.
- Asignaturas en las que está matriculado, incluyendo todos los datos de matriculación.

El sistema debe permitir la adición, eliminación, modificación y consulta de los datos relativos a cada alumno.

Gestión de Asignaturas

Módulo para la gestión de las asignaturas. Para cada asignatura deben almacenarse los siguientes datos.

- Identificador de la asignatura.
- Nombre de la asignatura.
- Profesores que la imparten, mostrando nombre y apellidos.
- Alumnos matriculados en la asignatura, mostrando nombre y apellidos.

El sistema debe permitir la adición, eliminación, modificación y consulta de los datos relativos a cada asignatura.

Gestión de Matriculas

Módulo para el registro de matriculas de un determinado alumno, en una determinada asignatura. Dicho módulo no será independiente, y vendrá implícito en la gestión de alumnos y asignaturas. Para cada matricula debe registrarse:

- Alumno.
- Asignatura.
- Curso académico.

El sistema debe permitir la adición, eliminación, modificación y consulta de los datos relativos a cada matricula.

Diseño del Sistema

En la presente sección se muestran los resultados obtenidos del proceso de diseño dentro de la iteración uno. Estos resultados se engloban en dos grandes áreas, el modelo de datos que formaría el *back-end* de la solución propuesta, así como el diseño de las interfaces de usuario desarrolladas, lo que formaría parte del *front-end*.

Modelo de datos

El diseño del modelo puede verse a través del siguiente diagrama de clases el cual se traduce directamente en el framework de persistencia y negocio a través de una arquitectura MDA.

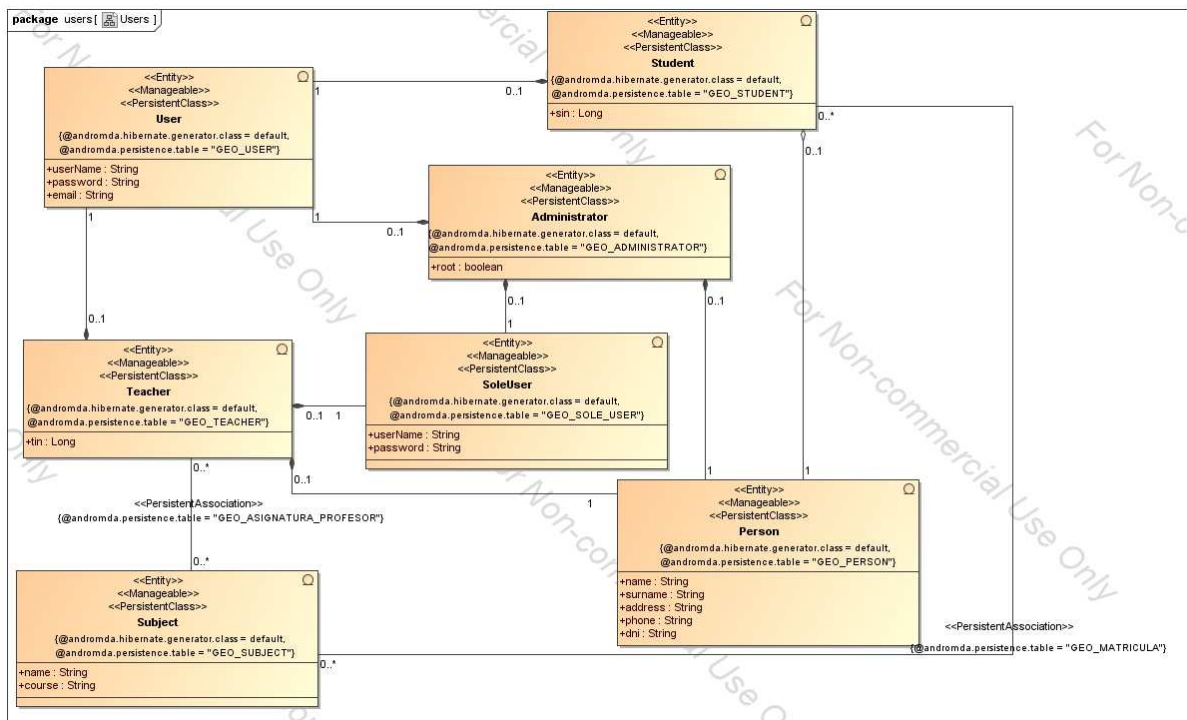


Figura 33. Modelo de negocio

El resultado de la utilización de la herramienta de generación de código AndroMDA, sobre el presente modelo de clases estereotipadas, son una serie de servicios implementados sobre EJB's, los cuales dan acceso a los datos de persistencia, encapsulados en una serie de librerías, que son importadas dentro de la aplicación desarrollada.

A continuación se hará una exposición detallada de las clases que componen el modelo de negocio utilizado de la presente iteración.

Administrator – Entidad Administrador

org.babieca.geo.ddbb.users.Administrator	
Nombre de Clase:	<ul style="list-style-type: none"> Administrator
Paquete:	<ul style="list-style-type: none"> org.babieca.geo.ddbb.users
Estereotipos:	<ul style="list-style-type: none"> Entity Manageable PersistentClass
Atributos:	<ul style="list-style-type: none"> root (boolean) – Flag que marca si es superusuario.
Métodos:	-----
Relaciones:	<ul style="list-style-type: none"> Person SoleUser User

Tabla 19. Clase Administrator

Person – Entidad Persona

org.babieca.geo.ddbb.users.Person	
Nombre de Clase:	<ul style="list-style-type: none"> Person
Paquete:	<ul style="list-style-type: none"> org.babieca.geo.ddbb.users
Estereotipos:	<ul style="list-style-type: none"> Entity Manageable PersistentClass
Atributos:	<ul style="list-style-type: none"> name (String) - Nombre surname (String) - Apellidos address (String) - Dirección phone (String) - Teléfono dni (String) – Documento nacional de identidad.
Métodos:	-----
Relaciones:	<ul style="list-style-type: none"> Administrator Student Teacher

Tabla 20. Clase Person

SoleUser – Entidad Usuario Sole

org.babieca.geo.ddbb.users.SoleUser	
Nombre de Clase:	<ul style="list-style-type: none"> • SoleUser
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.users
Estereotipos:	<ul style="list-style-type: none"> • Entity • Manageable • PersistentClass
Atributos:	<ul style="list-style-type: none"> • userName (String) – Nombre de usuario • password (String) – Clave del usuario
Métodos:	-----
Relaciones:	<ul style="list-style-type: none"> • Administrator • Teacher

Tabla 21. Clase SoleUser

Student – Entidad Estudiante

org.babieca.geo.ddbb.users.Student	
Nombre de Clase:	<ul style="list-style-type: none"> • Student
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.users
Estereotipos:	<ul style="list-style-type: none"> • Entity • Manageable • PersistentClass
Atributos:	<ul style="list-style-type: none"> • sin (String) – Número de identificación del alumno.
Métodos:	-----
Relaciones:	<ul style="list-style-type: none"> • Person • Subject • User

Tabla 22. Clase Student

Subject – Entidad Asignatura

org.babieca.geo.ddbb.users.Subject	
Nombre de Clase:	<ul style="list-style-type: none"> • Subject
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.users
Estereotipos:	<ul style="list-style-type: none"> • Entity • Manageable • PersistentClass
Atributos:	<ul style="list-style-type: none"> • name (String) – Nombre de la asignatura • course – Curso de la asignatura
Métodos:	-----
Relaciones:	<ul style="list-style-type: none"> • Administrator • Teacher

Tabla 23. Clase Subject

Teacher – Entidad Profesor

org.babieca.geo.ddbb.users.Teacher	
Nombre de Clase:	<ul style="list-style-type: none"> • Teacher
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.users
Estereotipos:	<ul style="list-style-type: none"> • Entity • Manageable • PersistentClass
Atributos:	<ul style="list-style-type: none"> • Tin (String) – Número de identificación del alumno.
Métodos:	-----
Relaciones:	<ul style="list-style-type: none"> • Person • Subject • SoleUser • User

Tabla 24. Clase Teacher

User – Entidad Usuario

org.babieca.geo.ddbb.users.User	
Nombre de Clase:	<ul style="list-style-type: none">• User
Paquete:	<ul style="list-style-type: none">• org.babieca.geo.ddbb.users
Estereotipos:	<ul style="list-style-type: none">• Entity• Manageable• PersistentClass
Atributos:	<ul style="list-style-type: none">• userName(String) – Nombre de usuario• password (String) – Clave del usuario• email - (String) – Email del usuario
Métodos:	-----
Relaciones:	<ul style="list-style-type: none">• Administrator• Student• Teacher

Tabla 25. Clase User

Interfaz Gráfica de Usuario

Completando al modelo de negocio en el ámbito del diseño, se presentan las distintas Interfaces de Usuario implementadas en la presente iteración. Éstas han sido agrupadas por funcionalidad para una mayor claridad.

Gestión de Alumnos

Tal como se ha visto en los requisitos expuestos con anterioridad, el sistema debe ser capaz de gestionar los alumnos. A continuación se muestra la interfaz principal de gestión, desde la que se listan los alumnos y a través de la cual puede accederse a las distintas funciones a ejecutar sobre ellos.

	NIA	Nombre	Apellidos	NIF	
+	100063758	Diego	Jimenez López	70665789F	
+	100063727	Borja	Blanco Iglesias	72886789S	
+	100065789	Fermin	Romero de Torres	50789457F	
+	100045678	Laura	Fernández García	67568798F	
+	100087965	Francisco	Narvaez Machón	67876543J	
+	100044567	Irene	Clemente Bueno	45876789S	

Figura 34. Listado de Alumnos

Desde la interfaz anterior puede ampliarse la información de cada alumno, dando acceso a una ventana de edición desde la que pueden modificarse todos los datos del alumno, incluyendo el listado de asignaturas en las que está matriculado.

	NIA	Nombre	Apellidos	NIF	
-	100063758	Diego	Jimenez López	70665789F	
Datos Personales					
NIA:	<input type="text" value="100063758"/>				
Nombre:	<input type="text" value="Diego"/>				
Apellidos:	<input type="text" value="Jimenez López"/>				
DNI:	<input type="text" value="70665789F"/>				
Dirección:	<input type="text" value="Kripton 12"/>				
Teléfono:	<input type="text" value="918332317"/>				
Datos de Usuario					
Alias:	<input type="text" value="diegolert"/>				
Password:	<input type="password" value="*****"/>				
Email:	<input type="text" value="rdsfdr@hj.es"/>				
<input type="button" value="Guardar"/>					
Asignaturas matriculadas					
	Asignatura	Curso			
	Tratamiento Digital de la Información	2			
	Física	1			
	Estructuras de Datos y de la Información	2			
	Teoría de Autómatas y Lenguajes Formales	4			

Figura 35. Detalle y edición datos de alumno

Para completar las operaciones de gestión existe una pantalla, también accesible desde la interfaz principal, que permite la adición de un nuevo alumno a partir de los datos introducidos.

Datos Personales	
NIA:	100076542
Nombre:	Franciso
Apellidos:	Narvaez Machón
DNI:	34567898K
Dirección:	Camino Viejo del General n121 41008
Teléfono:	555 678 654

Datos de Usuario	
Alias:	kiko
Password:	••••
Email:	kiko@babieca.org

Figura 36. Nuevo alumno

Por último se muestra la pantalla a través de la cual pueden añadirse matriculaciones del alumno en las destinas asignaturas.

Asignatura:
Física
Tratamiento Digital de la Información
Física
Estructuras de Datos y de la Información
Teoría de Automatas y Lenguajes Formales

Figura 37. Nueva matricula

Gestión de Profesores

Al igual que en el caso de los alumnos, las siguientes interfaces permiten la gestión de los profesores. Como interfaz principal y siguiendo con el ejemplo de los alumnos, encontramos el siguiente listado de profesores.

	NIP	Nombre	Apellidos	NIF	
+	10009	Boja	Blanco	76789765S	
+	1008	Jesus	Bermudo Cruz	76543456S	

Figura 38. Listado de Profesores

De manera análoga al caso de los alumnos, desde el listado puede ampliarse la información de cada profesor, dando acceso a una ventana de edición desde la que pueden modificarse todos los datos del alumno incluyendo el listado de asignaturas que imparte el profesor.

NIP	Nombre	Apellidos	NIF
10009	Borja	Blanco	76789765S

Datos Personales

NIP:

Nombre:

Apellidos:

DNI:

Dirección:

Teléfono:

Datos de Usuario

Alias:

Password:

Email:

Usuario servidor SOLE

Alias:

Password:

Asignatura	Curso
Física	1

NIP	Nombre	Apellidos	NIF
1008	Jesus	Bermudo Cruz	76543456S

Figura 39. Edición y visualización de la información relativa a un profesor

Siguiendo con el ejemplo de los alumnos, a continuación puede verse la pantalla de adición de profesores.

Nuevo Profesor

Datos Personales

NIP:

Nombre:

Apellidos:

DNI:

Dirección:

Teléfono:

Datos de Usuario

Alias:

Password:

Email:

Usuario servidor SOLE

Alias:

Password:

Figura 40. Nuevo profesor

En lugar de matricular alumnos en asignaturas, en el caso de los profesores se asignan las asignaturas que cada profesor imparte a través de la siguiente interfaz.



Figura 41. Adición de una asignatura impartida por el profesor

Gestión de Administradores

Una vez más, la gestión de administradores es similar a las anteriores, con la salvedad de alguna parametrización más propia de la entidad, así como la no relación de este rol con las asignaturas. A continuación se muestran la pantalla de acceso al listado y opciones de gestión.

	Nombre	Apellidos	NIF	Root	
+	Borja	Blanco Iglesia	50.433.221-F	✓	
+	Irene	Clemente Bueno	50456783-D	✗	

Figura 42. Listado de Administradores

La edición y visualización de los datos de cada administrador sigue un diseño común para todas las gestiones.

Nombre	Apellidos	NIF	Root	
Borja	Blanco Iglesia	50.433.221-F	✓	
Datos Personales				
Nombre:	Borja			
Apellidos:	Blanco Iglesias			
DNI:	50.433.221-F			
Dirección:	C/ Sta. Maria de la Cabeza 23456 Madrid			
Teléfono:	918 917 234			
Superusuario:	<input checked="" type="checkbox"/>			
Datos de Usuario				
Alias:	fermin			
Password:	•••••			
Email:	fermin@gmail.com			
Usuario servidor SOLE				
Alias:	feSole			
Password:	•••••			
<input type="button" value="Guardar"/>				
Irene	Clemente Bueno	50456783-D	✗	

Figura 43. Edición y visualización de la información relativa a un administrador

Para finalizar, a continuación se expone la pantalla de inserción de administradores del sistema.

Figura 44. Nuevo administrador

Gestión de Asignaturas

Finalizando los diseños gráficos de las interfaces de usuario, se muestran las relativas a la gestión de asignaturas siguiendo el modelo establecido.

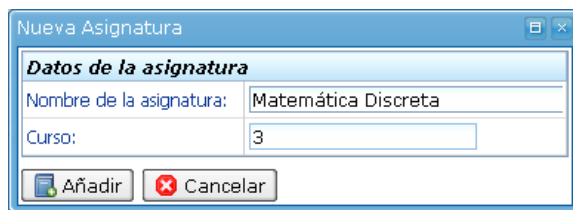
Asignatura	Curso
Tratamiento Digital de la Información	2
Física	23
Estructuras de Datos y de la Información	2
Teoría de Autómatas y Lenguajes Formales	4

Figura 45. Listado de Asignaturas

La edición es igual que la anteriores, si bien el diseño parece algo distinto por el menor número de campos que tiene esta entidad.

Figura 46. Edición y visualización de la información relativa a una asignatura

Por último se muestra la interfaz de adición de asignaturas siguiendo la tónica general.



Nueva Asignatura

Datos de la asignatura

Nombre de la asignatura: Matemática Discreta

Curso: 3

Añadir Cancelar

Figura 47. Nueva asignatura

Pruebas

En la presente sección se expondrán los resultados obtenidos de la ejecución del plan de pruebas tanto funcionales como no, sobre el desarrollo realizado dentro de la primera iteración.

Pruebas Funcionales

Las pruebas funcionales tienen como objetivo principal, la verificación de la adecuación de la solución a las especificaciones expuestas en el análisis, comprobando que el sistema funciona tal como se describe en los documentos de análisis.

Cabe resaltar que para una correcta ejecución del plan de pruebas, éste debe ser ejecutado a partir de una base de datos vacía y siguiendo un riguroso orden de ejecución tal como marca el identificador dado a cada prueba.

El Identificador de las pruebas se compone de cuatro partes:

- Los caracteres PF seguidos de un guión indican que se trata de una prueba funcional.
- Los caracteres IT1 indican que se trata una prueba de la primera iteración del desarrollo.
- Dos caracteres indicando el módulo de pruebas: gestión de asignaturas, gestión de administradores, gestión de alumnos y gestión de profesores (GA, GD, GL, GP respectivamente).
- Cuatro dígitos que indican unívocamente el número de prueba realizada.

A continuación se exponen los resultados obtenidos en las distintas pruebas:


ID:	<i>PF-IT1_GA_0001</i>	V.B. 
Nombre:	Inserción asignaturas.	
Descripción:	Se insertan tres asignaturas con los siguientes datos: <ul style="list-style-type: none"> • Nombre: <i>Asignatura 1</i> – Curso: <i>1</i> • Nombre: <i>Asignatura 2</i> – Curso: <i>2</i> • Nombre: <i>Asignatura 3</i> – Curso: <i>3</i> Se recarga el listado de asignaturas.	
Resultado Esperado:	Las asignaturas deben ir apareciendo en el listado con los datos introducidos a medida que se van insertando, además al recargarse el listado deben seguir apareciendo.	
Incidencia:	[No-Aplica]	

Figura 48. PF-IT1_GA_0001


ID:	PF-IT1_GA_0002	V.B. 
Nombre:	Modificación asignaturas.	
Descripción:	Se expande la información de la “Asignatura 3”, se modifica el nombre por “Asignatura 0” y el curso por “0”.	
Resultado Esperado:	<p>Al expandir la información de la asignatura, deben aparecer los mismos datos que en la tabla con la posibilidad de editarnos.</p> <p>Una vez modificada la información y pulsado el botón de guardar, deben modificarse los datos en el listado automáticamente.</p> <p>Al recargar el listado, el registro de la asignatura debe mantener los datos modificados.</p>	
Incidencia:	[No-Aplica]	

Figura 49. PF-IT1_GA_0002


ID:	PF-IT1_GA_0003	V.B. 
Nombre:	Eliminación asignaturas.	
Descripción:	Se elimina la “Asignatura 2”.	
Resultado Esperado:	La asignatura debe eliminarse automáticamente del listado, además al recargar el listado la asignatura ya no debe aparecer.	
Incidencia:	[No-Aplica]	

Figura 50. PF-IT1_GA_0003


ID:	PF-IT1_GD_0004	V.B. 
Nombre:	Inserción administradores.	
Descripción:	<p>Se insertan dos administradores con los siguientes nombres y datos aleatorios pero identificables:</p> <ul style="list-style-type: none"> • Nombre: <i>Administrador 1</i> • Nombre: <i>Administrador 2</i> <p>Se recarga el listado de administradores.</p>	
Resultado Esperado:	Los administradores deben ir apareciendo en el listado con los datos introducidos a medida que se van insertando, además al recargarse el listado deben seguir apareciendo.	
Incidencia:	[No-Aplica]	

Figura 51. PF-IT1_GD_0004


ID:	PF-IT1_GD_0005	V.B. 
Nombre:	Modificación administradores.	
Descripción:	Se expande la información del “Administrador 2”, se modifica el nombre por “Administrador 0” y el resto de datos por datos aleatorios pero identificables.	
Resultado Esperado:	<p>Al expandir la información del administrador, deben aparecer los mismos datos con los que se insertó el administrador con la posibilidad de editarlos.</p> <p>Una vez modificada la información y pulsado el botón de guardar, deben modificarse los datos correspondientes en el listado automáticamente.</p> <p>Al recargar el listado, el registro del administrador debe mantener los datos modificados, además al expandir el “Administrador 0” debe aparecer la misma información modificada.</p>	
Incidencia:	[No-Aplica]	

Figura 52. PF-IT1_GD_0005


ID:	PF-IT1_GD_0006	V.B. 
Nombre:	Eliminación administrador.	
Descripción:	Se elimina la “Administrador 1”.	
Resultado Esperado:	El administrador debe eliminarse automáticamente del listado, además al recargar el listado el administrador ya no debe aparecer.	
Incidencia:	[No-Aplica]	

Figura 53. PF-IT1_GD_0006


ID:	PF-IT1_GL_0007	V.B. 
Nombre:	Inserción alumnos.	
Descripción:	<p>Se insertan tres alumnos con los siguientes nombres y datos aleatorios pero identificables:</p> <ul style="list-style-type: none"> • Nombre: <i>Alumno 1</i> • Nombre: <i>Alumno 2</i> • Nombre: <i>Alumno 3</i> <p>Se recarga el listado de alumnos.</p>	
Resultado Esperado:	Los alumnos deben ir apareciendo con los datos introducidos en el listado a medida que se van insertando, además al recargarse el listado deben seguir apareciendo.	
Incidencia:	[No-Aplica]	

Figura 54. PF-IT1_GL_0007


ID:	PF-IT1_GL_0008	V.B. 
Nombre:	Modificación alumnos.	
Descripción:	Se expande la información del “Alumno 3”, se modifica el nombre por “Alumno 0” y el resto de datos por datos aleatorios pero identificables.	
Resultado Esperado:	<p>Al expandir la información del alumno, deben aparecer los mismos datos con los que se insertó el alumno con la posibilidad de editarlos.</p> <p>Una vez modificada la información y pulsado el botón de guardar, deben modificarse los datos correspondientes en el listado automáticamente.</p> <p>Al recargar el listado, el registro del alumno debe mantener los datos modificados, además al expandir el “Alumno 0” debe aparecer la misma información modificada.</p>	
Incidencia:	[No-Aplica]	

Figura 55. PF-IT1_GL_0008


ID:	PF-IT1_GL_0009	V.B. 
Nombre:	Adición matricula alumno	
Descripción:	Se expande la información del “Alumno 1”, se añade la asignatura “Asignatura 1” y posteriormente la “Asignatura 0”. Recargar el listado de alumnos, y volver a expandir la información del alumno.	
Resultado Esperado:	Al añadir cada una de las asignaturas, éstas deben ir apareciendo en el listado de matriculas del alumno. Al recargar el listado y volver a expandir la información del “Alumno 1”, las asignaturas matriculadas deben seguir apareciendo.	
Incidencia:	[No-Aplica]	

Figura 56. PF-IT1_GL_0009


ID:	PF-IT1_GL_00010	V.B. 
Nombre:	Eliminación matricula alumno	
Descripción:	Se expande la información del “Alumno 1”, se elimina la asignatura “Asignatura 1”. Recargar el listado de alumnos, y volver a expandir la información del alumno.	
Resultado Esperado:	Al eliminar la asignatura, ésta debe desapareciendo del listado de matriculas del alumno automáticamente. Al recargar el listado y volver a expandir la información del “Alumno 1”, la asignatura eliminada no debe aparecer en el listado de matriculas, apareciendo únicamente la “Asignatura 0”.	
Incidencia:	[No-Aplica]	

Figura 57. PF-IT1_GL_00010


ID:	<i>PF-IT1_GL_0011</i>	V.B. 
Nombre:	Eliminación alumno.	
Descripción:	Se elimina el “ <i>Alumno 2</i> ”.	
Resultado Esperado:	El alumno debe eliminarse automáticamente del listado, además al recargar el listado el alumno ya no debe aparecer.	
Incidencia:	[No-Aplica]	

Figura 58. PF-IT1_GL_0011


ID:	<i>PF-IT1_GP_0012</i>	V.B. 
Nombre:	Inserción profesores.	
Descripción:	<p>Se insertan tres profesores con los siguientes nombres y datos aleatorios pero identificables:</p> <ul style="list-style-type: none"> • Nombre: <i>Profesor 1</i> • Nombre: <i>Profesor 2</i> • Nombre: <i>Profesor 3</i> <p>Se recarga el listado de profesores.</p>	
Resultado Esperado:	Los Profesores deben ir apareciendo con los datos introducidos en el listado a medida que se van insertando, además al recargarse el listado deben seguir apareciendo.	
Incidencia:	[No-Aplica]	

Figura 59. PF-IT1_GP_0012


ID:	<i>PF-IT1_GP_0013</i>	V.B. 
Nombre:	Modificación profesores.	
Descripción:	Se expande la información del “ <i>Profesor 3</i> ”, se modifica el nombre por “ <i>Profesor 0</i> ” y el resto de datos por datos aleatorios pero identificables.	
Resultado Esperado:	<p>Al expandir la información del profesor, deben aparecer los mismos datos con los que se insertó el profesor con la posibilidad de editarlos.</p> <p>Una vez modificada la información y pulsado el botón de guardar, deben modificarse los datos correspondientes en el listado automáticamente.</p> <p>Al recargar el listado, el registro del profesor debe mantener los datos modificados, además al expandir el “<i>Profesor 0</i>” debe aparecer la misma información modificada.</p>	
Incidencia:	[No-Aplica]	

Figura 60. PF-IT1_GP_0013


ID:	PF-IT1_GP_0014	V.B.	
Nombre:	Adición asignación profesor		
Descripción:	Se expande la información del “ <i>Profesor 1</i> ”, se añade la asignatura “Asignatura 1” y posteriormente la “Asignatura 0”. Recargar el listado de profesores, y volver a expandir la información del profesor.		
Resultado Esperado:	Al añadir cada una de las asignaturas, éstas deben ir apareciendo en el listado de asignaciones del profesor. Al recargar el listado y volver a expandir la información del “ <i>Profesor 1</i> ”, las asignaturas matriculadas deben seguir apareciendo.		
Incidencia:	Una vez asignadas las asignaturas, recargado y el listado y expandido la información del “ <i>Profesor 1</i> ”, las asignaturas no aparecen en el listado.		

Figura 61. PF-IT1_GP_0014


ID:	PF-IT1_GP_00015	V.B.	
Nombre:	Eliminación asignación profesor		
Descripción:	Se expande la información del “ <i>Profesor 1</i> ”, se elimina la asignatura “Asignatura 1”. Recargar el listado de profesores, y volver a expandir la información del profesor.		
Resultado Esperado:	Al eliminar la asignatura, ésta debe desaparecer del listado de asignaturas del profesor automáticamente. Al recargar el listado y volver a expandir la información del “ <i>Profesor 1</i> ”, la asignatura eliminada no debe aparecer en el listado de asignaciones, apareciendo únicamente la “Asignatura 0”.		
Incidencia:	Al no haber tenido éxito la prueba PF-IT1_GP_0014, se ha procedido a insertar los registros de asignación de matriculas directamente en la BBDD para poder ejecutar ésta prueba.		

Figura 62. PF-IT1_GP_00015


ID:	PF-IT1_GP_0016	V.B.	
Nombre:	Eliminación profesor.		
Descripción:	Se elimina el “ <i>Profesor 2</i> ”.		
Resultado Esperado:	El profesor debe eliminarse automáticamente del listado, además al recargar el listado el profesor ya no debe aparecer.		
Incidencia:	[No-Aplica]		

Figura 63. PF-IT1_GP_0016

Pruebas No Funcionales

Dentro de las pruebas no funcionales se engloban todas aquellas que no guardan relación con los requisitos funcionales propiamente dichos como por ejemplo, pruebas de rendimiento, coherencia de datos en el gestor de base de datos, etc.

Al igual que ocurría en las pruebas funcionales, es importante destacar que para una correcta ejecución del plan de pruebas, éste debe seguir ejecutándose a partir de los datos residuales de las pruebas funcionales, y siguiendo el orden establecido en el documento por el identificador de las pruebas.

El Identificador se compone de cuatro partes:

- Los caracteres PN seguidos de un guión indican que la prueba es no funcional.
- Tres caracteres IT1 indicando que es la primera iteración.
- Dos caracteres indicando el ámbito de pruebas, en este caso BD indicando pruebas de base de datos.
- Cuatro dígitos que indican unívocamente el número de prueba realizada.

A continuación se exponen los resultados obtenidos en las distintas pruebas:


ID:	<i>PN-IT1_BD_00017</i> V.B. 
Nombre:	Actualización automática de datos relacionados en las eliminaciones.
Descripción:	Se eliminan todos los registros de: <ul style="list-style-type: none"> • Administradores. • Profesores. • Asignaturas. • Alumnos.
Resultado Esperado:	Los registros deben eliminarse sin dar ningún error.
Incidencia:	Al eliminar el “Profesor 1”, éste da un error. Éste error está relacionado con la incidencia de la prueba 14. Debe modificarse la actualización de las asignaturas en las entidades profesor.

Figura 64. PN-IT1_BD_00017

ID:	<i>PN-IT1_BD_00018</i>	V.B. ✓
Nombre:	Eliminación relaciones entidades.	
Descripción:	Una vez ejecutada la prueba 17, debe comprobarse el modelo en BBDD.	
Resultado Esperado:	No debe quedar ningún dato en ninguna tabla.	
Incidencia:	En la BBDD únicamente quedan los datos referentes al “Profesor 1” ya que no pudo ser eliminado por el fallo detectado en la prueba anterior. El resultado de la ésta prueba ha sido satisfactorio ya que las relaciones de todas las entidades han sido eliminadas.	

Figura 65. PN-IT1_BD_00018

Seguimiento Planificación

Tal como se marcó en la planificación anterior (véase el cronograma adjunto), la presente iteración debería haberse terminado el día 1 de mayo de 2009, sin embargo la fecha de entrega se realiza el día 30 del mismo mes lo que supone un retraso de un mes sobre la planificación establecida.

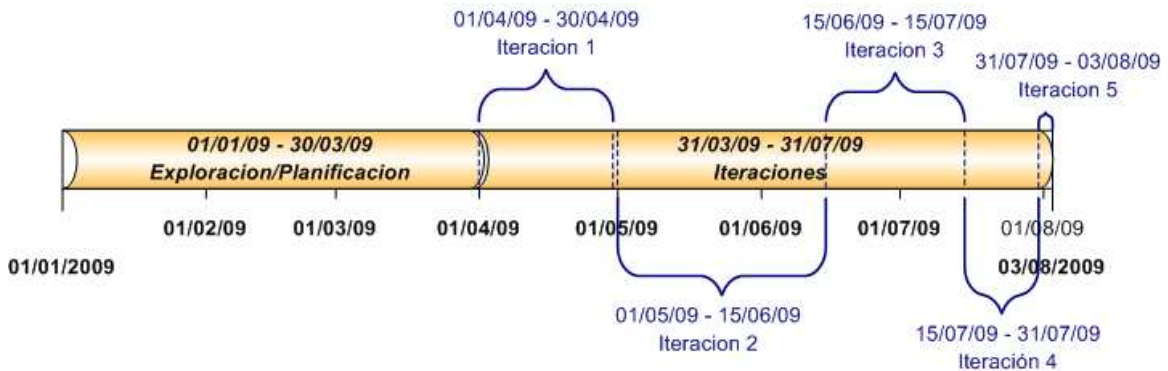


Figura 66. Cronograma con la anterior planificación del proyecto

Una vez estudiadas las posibles causas del retraso, no se contempla toma de alguna medida adicional a excepción de la extensión en el tiempo, de todos los hitos por el periodo contemplado en el retraso de un mes, resultando el siguiente cronograma.

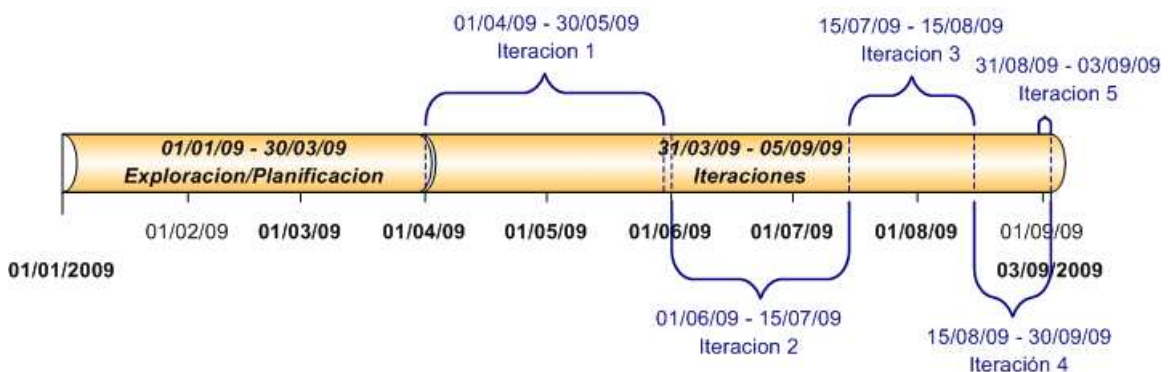


Figura 67. Cronograma con la actual planificación del proyecto

Validación del Documento

Próximos Hitos e Incidencias

ID	Tema	Resolución	Responsable	Fecha Objetivo	Fecha de Impacto
3	Definición servicios SOLE		Diego Jiménez	15/04/09	10/06/09
5	Desarrollo iteración 2		Borja Blanco	15/06/09	15/07/09
6	Error PN-IT1_BD_00017		Borja Blanco	15/07/09	15/07/09

Hitos e Incidencias Cerrados

ID	Tema	Resolución	Responsable	Fecha Objetivo	Fecha Cierre
1	Análisis y Planificación	Finalizado	Borja Blanco	01/04/09	01/04/09
2	Arquitectura de Sistema	Finalizado	Borja Blanco	10/04/09	09/05/09
4	Desarrollo iteración 1	Finalizado	Borja Blanco	01/05/09	30/05/09

Aprobación del Documento

Rol	Nombre	Firma	Fecha
Autor del Proyecto	Borja Blanco Iglesias		
Tutor del Proyecto	Ricardo Colomo Palacios		

Documento Técnico de la Iteración 2



**Sistema de Gestión de
Exámenes Online
GEO**

**Informe Técnico sobre
Iteración 2
Madrid, 15 de Julio de 2009**

Autor	Borja Blanco Iglesias
Fecha de Creación	15/07/2009
Fecha de actualización	15/07/2009
Nombre de documento	IT2_GEO.doc
Versión	1.0

Histórico de Versiones

Fecha	Autor	Versión	Comentarios
15/07/2009	Borja Blanco	1.0	Versión inicial del documento.

Revisores

Nombre	Posición
Borja Blanco Iglesias	Autor del Proyecto
Ricardo Colomo Palacios	Tutor del Proyecto

Lista de distribución

Copia N°	Nombre	Posición
1.	Borja Blanco Iglesias	Autor del Proyecto
2.	Ricardo Colomo Palacios	Tutor del Proyecto
3.	Diego Jiménez López	Colaborador
4.		
5.		

Introducción

El Informe técnico de una iteración del desarrollo de un sistema bajo la metodología XP, ofrece una visión global sobre el análisis realizado así como las decisiones de diseño e implementación que se han tomado a lo largo del presente desarrollo.

Objetivos del Documento

El documento tiene como objetivo principal ofrecer una visión global sobre el estado actual del proyecto, enfocado hacia las tareas realizadas sobre la iteración dos en todo el periodo de desarrollo.

A lo largo del documento se detallarán aquellos aspectos funcionales que contempla el desarrollo realizado como consecuencia de la iteración, así como las diversas decisiones de diseño tomadas a lo largo del desarrollo. Además, se incluirán los resultados obtenidos del proceso de pruebas realizado.

Por último, cabe resaltar la inclusión en el documento, de una revisión de la planificación realizada en función a los tiempos de desarrollo obtenidos, estableciendo las comparaciones oportunas con los tiempos planificados.

Audiencia

Debido al carácter técnico, el presente documento está dirigido a aquellas personas que tomen parte en tareas de desarrollo del proyecto y necesiten acceso a la documentación sobre las tareas realizadas en la iteración dos. Así mismo, por la información general tratada en el mismo, la audiencia del documento se extiende a la dirección del proyecto quien deberá validar los requisitos definidos como base del desarrollo.

Definiciones, acrónimos, abreviaturas y referencias

A continuación se incluye información relativa a la terminología utilizada en el documento así como a las referencias realizadas en el mismo.

- **EJB:** Acrónimo del inglés *Enterprise Java bean* es una API dentro del estándar JEE para la comunicación remota de servicios.
- **MDA:** Acrónimo del inglés *Model driven architecture* cuyo significado es arquitectura dirigida por el modelo.
- **SOLE:** Servidor OnLine de Exámenes.
- **XP (Extreme Programming):** Metodología de desarrollo ágil cuya traducción al español es Programación Extrema.

Definición del alcance del desarrollo

Dentro del ámbito del proyecto la presente iteración contempla el desarrollo de los siguientes módulos:

Configuración de examen y parametrización de los datos:

Siguiendo con las tareas de gestión de la primera iteración, el presente módulo es el encargado de establecer y modificar los datos relacionados con la configuración de un examen. Para cada configuración de examen deben almacenarse los siguientes datos:

- Asignatura a la que corresponde el examen.
- Nombre identificativo del examen.
- Definición del periodo de realización del examen.
- Definición del periodo de revisión del examen.
- Tiempo límite para hacer el examen.
- Puntuación total del examen.
- Porcentaje que resta cada respuesta errónea.
- Corrección automática instantánea. Siempre que el tipo de todas las preguntas sea de corrección automática, se puede definir que el sistema informe de la nota una vez terminado el examen, o no.
- Preguntas del examen. El usuario debe poder añadir preguntas a un examen, de dos maneras distintas:
 - Añadiendo manualmente la pregunta. En este caso, el usuario irá añadiendo preguntas a una tabla. En cada pregunta debe definir, el enunciado, las respuestas a mostrar (en el caso correspondiente dependiendo del tipo) y la o las respuestas correctas. Así mismo debe especificar si la pregunta creada será compartida o no, y en el caso de serlo debe establecer la visibilidad de la misma. Este punto será descrito en el módulo de comunicación.
 - Realizando una búsqueda semántica al motor de preguntas. Este motor devolverá una tabla, que incluirá una pre visualización de la pregunta dando la posibilidad de seleccionar las preguntas a realizar. Esta pre visualización permitirá visualizar la meta información de la pregunta así como valorar y comentar la pregunta a través del módulo de comunicación tal como se expone en la descripción de dicho módulo.
- Número de preguntas que componen el examen. En el caso de que el examen deba tener menos preguntas de las seleccionadas, puede generarse un examen con un número determinado de preguntas seleccionadas aleatoriamente.

Gestión de la configuración Sole:

Debe existir un módulo de configuración en el que se establezca la URL del servidor SOLE asociado al sistema, a partir del cual se realizarán las búsquedas Online de contenidos.

Importación y exportación de datos

Una de las principales características del sistema debe ser la estandarización de datos a partir de la cual poder compartir los datos generados a la hora de componer un examen. Para ello debe existir un módulo de importación y exportación de datos capaz de transformar los datos introducidos por el usuario, a ficheros XML definidos bajo el estándar IMS QTI 2.1, y a la inversa. Este módulo será utilizado internamente por otros módulos del sistema, como por ejemplo el módulo de comunicación, por lo que no debe contemplarse una interacción directa del usuario con él mismo.

Módulo de comunicación

Siguiendo la idea del módulo anterior, es importante recalcar que debe existir un módulo encargado de las comunicaciones del sistema con sistemas externos. En este caso, estas comunicaciones serán con un sistema servidor de contenidos remoto, SOLE. En principio se definen cinco comunicaciones con este sistema:

- En el ámbito del submódulo de gestión de exámenes, cuando un profesor vaya a realizar una configuración de examen, puede realizar una búsqueda de preguntas. Esta búsqueda lanza una consulta remota a través del módulo de comunicación al servidor de preguntas, para obtener un listado de preguntas categorizadas sobre la búsqueda realizada. La búsqueda contendrá los siguientes campos:
 - Autenticación del usuario que lanza la pregunta, en este caso el profesor.
 - Cadena de consulta, con el texto a buscar en el servidor.
 - Nivel de privacidad:
 - Búsqueda global en el sistema.
 - Búsqueda privada del usuario.
 - Búsqueda en redes a las que esté agregado el usuario.
 - Dominio de búsqueda. Uri con el dominio asociado a la ontología a buscar.
 - Número de preguntas a buscar.
 - Criterio de ordenación.

Esa misma búsqueda devolverá un listado de preguntas con la información relevante de cada pregunta en formato QTI. Cada pregunta estará identificada unívocamente por un ID dado por el propio servidor SOLE.

- Siguiendo con el ámbito del submódulo de configuración de exámenes, cuando un profesor opte por crear preguntas nuevas, se debe producir una nueva comunicación con el servidor de preguntas para compartir la pregunta generada. En esta comunicación el usuario puede optar por

ocultar la información, hacerla pública o sólo visible por su grupo. Los campos que deben especificarse en esta llamada deben ser:

- *Autenticación remota.* Usuario y contraseña del servidor SOLE.
- *QTI con la pregunta.* Texto XML en formato QTI con la pregunta a compartir.
- *Dominio de conocimiento.* URI con el dominio donde se incluirá la pregunta generada. Previamente debe haber una consulta al servidor para saber los dominios disponibles del servidor.

Como respuesta a esta petición de compartición de conocimiento, el sistema debe responder con el ID único que ha asignado a dicha pregunta.

- Consulta de dominios disponibles en el sistema. Para realizar algunas de las consultas y peticiones a un servidor SOLE es necesario adjuntar el dominio de conocimiento asociado a la consulta o petición. Es por esto que se debe realizar una consulta previa al sistema para conocer los dominios que ofrece el sistema.
- Al pre visualizar una pregunta ofrecida por el servidor SOLE a la hora de configurar un examen, es necesario mostrar meta información no incluida en la respuesta de búsqueda del servidor. Para conocer esta meta información (valoración y comentarios de la pregunta) es necesario realizar una consulta previa, en la que, a partir del identificador de la pregunta, el servidor devuelve la meta información.
- Al igual que la consulta anterior, a la hora de pre visualizar la información de la pregunta, debe darse la posibilidad al usuario de valorar y comentar la pregunta descargada, para ello se realiza una petición de valoración adjuntando la puntuación y los comentarios, así como el identificador de la pregunta a valorar.

Es importante saber que en cada envío o petición realizada al servidor SOLE es necesario adjuntar los datos necesarios para que se realice la autenticación del usuario en el sistema remoto SOLE.

Buscador Semántico

El sistema debe realizar búsquedas semánticas a la hora de elegir las preguntas de un examen dentro de la configuración del mismo. Este sistema de búsqueda semántica no será incluido en el desarrollo del sistema sino que se accederá a través de un servicio ofrecido por el servidor SOLE por medio de la consulta de búsqueda del modulo de comunicación.

Diseño del Sistema

En la presente sección se muestran los resultados obtenidos del proceso de diseño dentro de la iteración dos. Estos resultados se engloban en dos grandes áreas, el modelo de datos que formaría el *back-end* de la solución propuesta, así como el diseño de las interfaces de usuario desarrolladas, lo que formaría parte del *front-end*.

Modelo de datos

El diseño del modelo puede verse a través de los siguientes diagramas de clases. Los dos primeros de ellos se traducen directamente en el framework de persistencia y negocio a través de una arquitectura MDA.

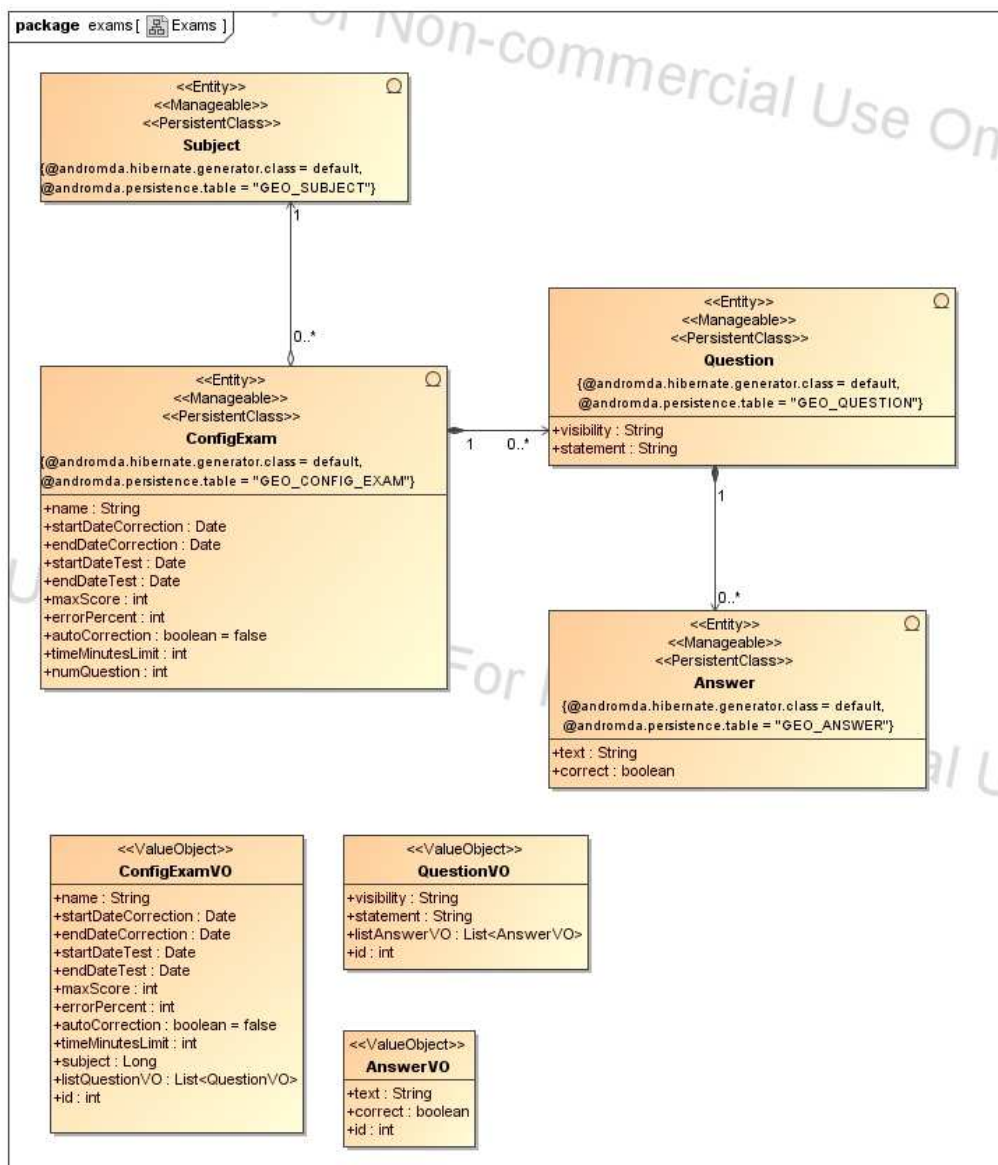


Figura 68. Modelo de negocio persistente relacionado con la configuración de exámenes

Si el diagrama anterior representaba y definía, el modelo persistente utilizado específicamente por el dominio de configuración de exámenes, a continuación se

muestra el dominio genérico para el almacenamiento de parámetros del servidor. Entre estos parámetros encontramos la URL del servidor SOLE.

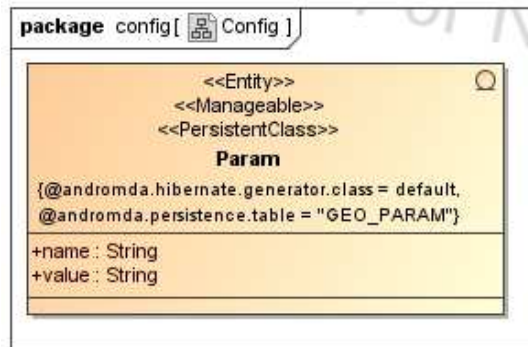


Figura 69. Modelo de negocio persistente de parámetros

El resultado de la utilización de la herramienta de generación de código AndroMDA, sobre el presente modelo de clases estereotipadas, son una serie de servicios implementados sobre EJB's, los cuales dan acceso a los datos de persistencia, encapsulados en una serie de librerías, que son importadas dentro de la aplicación desarrollada.

Para completar el modelo se expone un último diagrama el cual contiene aquellas clases utilizadas en el diseño e implementación de los módulos de comunicación e importación.

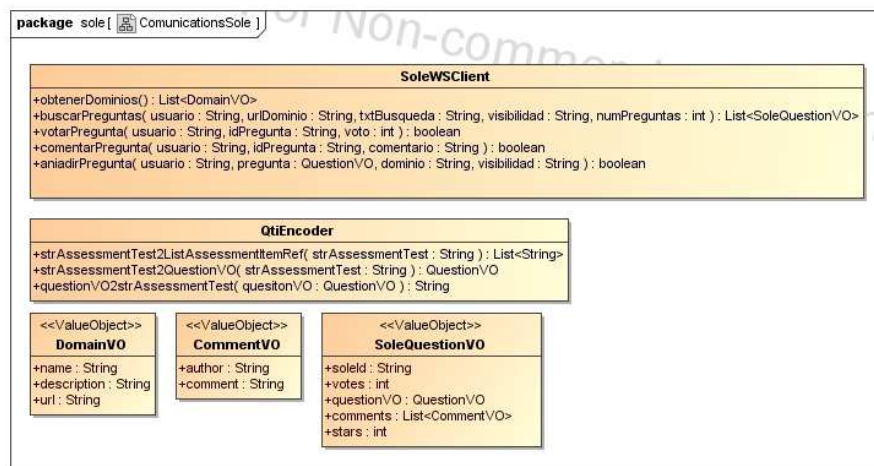


Figura 70. Modelo de negocio de la comunicación e importación

A continuación se hará una exposición detallada de las clases que componen el modelo de negocio utilizado de la presente iteración.

Answer – Entidad Respuesta

org.babieca.geo.ddbb.exams.Answer	
Nombre de Clase:	<ul style="list-style-type: none"> • Answer
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.exams
Estereotipos:	<ul style="list-style-type: none"> • Entity • Manageable • PersistentClass
Atributos:	<ul style="list-style-type: none"> • text (String) – Texto de la respuesta. • correct(boolean) – Flag que marca si una respuesta es correcta o no.
Métodos:	-----
Relaciones:	<ul style="list-style-type: none"> • Answer • ConfigExam

Tabla 26. Clase Answer

AnswerVO –Value Object de Respuesta

org.babieca.geo.ddbb.exams.AnswerVO	
Nombre de Clase:	<ul style="list-style-type: none"> • AnswerVO
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.exams
Estereotipos:	<ul style="list-style-type: none"> • ValueObject
Atributos:	<ul style="list-style-type: none"> • id (int) – Identificador de la respuesta. • text (String) – Texto de la respuesta. • correct(boolean) – Flag que marca si una respuesta es correcta o no.
Métodos:	-----
Relaciones:	-----

Tabla 27. Clase AnswerVO

ConfigExam – Entidad Configuración de Examen

org.babieca.geo.ddbb.exams.ConfigExam	
Nombre de Clase:	<ul style="list-style-type: none"> • ConfigExam
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.exams
Estereotipos:	<ul style="list-style-type: none"> • Entity • Manageable • PersistentClass
Atributos:	<ul style="list-style-type: none"> • name (String) – Nombre del examen • startDateCorrection (Date) – Dia inicial del periodo de corrección • endDateCorrection (Date) – Dia final del periodo de corrección • startDateTest (Date) – Dia inicial del periodo de realización del examen. • endDateTest (Date) – Dia final del periodo de realización del examen. • maxScore (int) – Puntuación máxima del examen. • errorPercent (int) – Porcentaje de penalización de una respuesta incorrecta. • autoCorrection (boolean) – Flag que marca si un examen es corregido inmediatamente después de finalizarse. • timeMinutesLimit (int) – Tiempo en minutos para finalizar el examen. • numQuestion (int) – Número de preguntas que componen el examen.
Métodos:	-----
Relaciones:	<ul style="list-style-type: none"> • Subject • Question

Tabla 28. Clase ConfigExam

ConfigExamVO – Value Object Configuración de Examen

org.babieca.geo.ddbb.exams.ConfigExamVO	
Nombre de Clase:	<ul style="list-style-type: none"> • ConfigExamVO
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.exams
Estereotipos:	<ul style="list-style-type: none"> • ValueObject
Atributos:	<ul style="list-style-type: none"> • name (String) – Nombre del examen • startDateCorrection (Date) – Dia inicial del periodo de corrección • endDateCorrection (Date) – Dia final del periodo de corrección • startDateTest (Date) – Dia inicial del periodo de realización del examen. • endDateTest (Date) – Dia final del periodo de realización del examen. • maxScore (int) – Puntuación máxima del examen. • errorPercent (int) – Porcentaje de penalización de una respuesta incorrecta. • autoCorrection (boolean) – Flag que marca si un examen es corregido inmediatamente después de finalizarse. • timeMinutesLimit (int) – Tiempo en minutos para finalizar el examen. • numQuestion (int) – Número de preguntas que componen el examen. • id (int) – Identificador de la configuración del Examen. • listQuestionVO (List<QuestionVO>) – Lista de preguntas que componen el examen encapsuladas en value objects.
Métodos:	-----
Relaciones:	-----

Tabla 29. Clase ConfigExamVO

Question – Entidad Pregunta

org.babieca.geo.ddbb.exams.Question	
Nombre de Clase:	<ul style="list-style-type: none"> • Question
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.exams
Estereotipos:	<ul style="list-style-type: none"> • Entity • Manageable • PersistentClass
Atributos:	<ul style="list-style-type: none"> • statement (String) – Enunciado de la pregunta. • visibility – Visibilidad de la pregunta.
Métodos:	-----
Relaciones:	<ul style="list-style-type: none"> • Answer • ConfigExam

Tabla 30. Clase Question

QuestionVO – Value Object Pregunta

org.babieca.geo.ddbb.exams.QuestionVO	
Nombre de Clase:	<ul style="list-style-type: none"> • QuestionVO
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.exams
Estereotipos:	<ul style="list-style-type: none"> • ValueObject
Atributos:	<ul style="list-style-type: none"> • statement (String) – Enunciado de la pregunta. • visibility – Visibilidad de la pregunta. • id (int) – Identificador de la pregunta. • listAnswerVO (List<AnswerVO>) – Lista de respuestas de la pregunta.
Métodos:	-----
Relaciones:	-----

Tabla 31. Clase QuestionVO

Param – Entidad Parametro

org.babieca.geo.ddbb.config.Param	
Nombre de Clase:	<ul style="list-style-type: none"> • Param
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.config
Estereotipos:	<ul style="list-style-type: none"> • Entity • Manageable • PersistentClass
Atributos:	<ul style="list-style-type: none"> • name (String) – Nombre del parámetro. • value (String) – Valor del parámetro.
Métodos:	-----
Relaciones:	-----

Tabla 32. Clase Param

CommentVO – Value Object Comentario

org.babieca.geo.comunications.sole.CommentVO	
Nombre de Clase:	<ul style="list-style-type: none"> • CommentVO
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.comunications.sole
Estereotipos:	<ul style="list-style-type: none"> • ValueObject
Atributos:	<ul style="list-style-type: none"> • author (String) – Autor del comentario. • comment (String) – Texto del comentario.
Métodos:	-----
Relaciones:	-----

Tabla 33. Clase CommentVO

DomainVO – Value Object Dominio

org.babieca.geo.comunications.sole.DomainVO	
Nombre de Clase:	• DomainVO
Paquete:	• org.babieca.geo.comunications.sole
Estereotipos:	• ValueObject
Atributos:	<ul style="list-style-type: none"> • name (String) – Nombre del dominio • description (String) – Descripción del dominio • url - (String) – Url del dominio
Métodos:	-----
Relaciones:	-----

Tabla 34. Clase DomainVO

SoleQuestionVO – Value Object Pregunta Sole

org.babieca.geo.comunications.sole.SoleQuestionVO	
Nombre de Clase:	• SoleQuestionVO
Paquete:	• org.babieca.geo.comunications.sole
Estereotipos:	• ValueObject
Atributos:	<ul style="list-style-type: none"> • soleId (String) – Identificador de la pregunta en Sole • votes (int) – Número medio de votos • questionVO (QuestionVO) – Información de la pregunta. • comments (List<CommentVO>) – Lista de comentarios. • stars (int) – Número de estrellas asignadas a la pregunta.
Métodos:	-----
Relaciones:	-----

Tabla 35. Clase SoleQuestionVO

SoleWSClient – Servicio cliente WS Sole

org.babieca.geo.comunications.sole.SoleWSClient	
Nombre de Clase:	<ul style="list-style-type: none"> • SoleWSClient
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.comunications.sole
Estereotipos:	<ul style="list-style-type: none"> • Service
Atributos:	-----
Métodos:	<ul style="list-style-type: none"> • obtenerDominios() : List<DomainVO> • buscarPreguntas(usuario : String, urlDominio : String, txtBusqueda : String, visibilidad : String, numPreguntas : int) : List<SoleQuestionVO> • votarPregunta(usuario : String, idPregunta : String, voto : int) : Boolean • comentarPregunta(usuario : String, idPregunta : String, comentario : String) : boolean • anadirPregunta(usuario : String, pregunta : QuestionVO, dominio : String, visibilidad : String) : boolean
Relaciones:	-----

Tabla 36. Clase SoleWSClient

QtiEncoder – Servicio Codificador QTI

org.babieca.geo.comunications.sole.QtiEncoder	
Nombre de Clase:	<ul style="list-style-type: none"> • QtiEncoder
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.comunications.sole
Estereotipos:	<ul style="list-style-type: none"> • Service
Atributos:	-----
Métodos:	<ul style="list-style-type: none"> • strAssessmentTest2ListAssessmentItemRef(strAssessmentTest : String) : List<String> • strAssessmentTest2QuestionVO(strAssessmentTest : String) : QuestionVO • questionVO2strAssessmentTest(quesitonVO : QuestionVO) : String
Relaciones:	-----

Tabla 37. Clase QtiEncoder

Interfaz Gráfica de Usuario

Completando al modelo de negocio en el ámbito del diseño, se presentan las distintas Interfaces de usuario implementadas en la presente iteración.

Nueva configuración de Examen

Toda nueva configuración de examen parte de una pantalla principal de creación dividida en tres apartados principales: configuración de exámenes, el cual tiene los datos principales de configuración, búsqueda de preguntas, para introducir los parámetros de búsqueda remota de preguntas, y preguntas con la referencia a aquellas asociadas al examen.

Al acceder a la pantalla únicamente puede verse expandido el panel de configuración. A través de este panel se introducen los datos de identificación del examen, los periodos de realización y revisión, así como distintos parámetros de puntuación y corrección.

Configuración del Examen	
Asignatura:	Estructuras de Datos y de la Información
Examen:	
Periodo Realización Examen:	desde 11/07/2009 08:39 PM hasta 11/07/2009 08:39 PM
Periodo Revisión Examen:	desde 11/07/2009 08:39 PM hasta 11/07/2009 08:39 PM
Puntuación Total:	10
Porcentaje de Error:	33
Tiempo Límite:	60 minutos
Corrección Automática:	<input checked="" type="checkbox"/>

Búsqueda de Preguntas

Preguntas

Guardar Cancelar

Figura 71. Nueva configuración de Examen

La búsqueda remota se compone de cuatro parámetros a través de los cuales se establece la visibilidad, dominio, número de resultados esperados así como el texto de búsqueda.

Búsqueda de Preguntas	
Visibilidad:	Público
Tema:	A software engineering ontology por ejemplo
Número Resultados:	10
Búsqueda:	METRICA

Buscar

Figura 72. Búsqueda de preguntas en servidor remoto

Como resultado de la adición de preguntas de forma manual y a través de la búsqueda remota, tal como veremos posteriormente, se irán añadiendo las preguntas en

forma de tabla tal como se muestra en la siguiente figura. A través de esta misma tabla, se pueden añadir, eliminar y modificar las preguntas.

Preguntas		
Número de Preguntas del Examen: 5		
Enunciado	Visibilidad	
¿Qué tipo de metodología es XP?	Público	
¿Qué función tiene la figura de Jefe de Proyecto?	Público	
¿Qué tarea no forma parte del DSI según METRICA?	Público	
¿Qué organismo definió METRICA?	Público	
¿Cual de las siguientes metodologías NO existe?	Público	

Figura 73. Listado de Preguntas asignadas al examen

Búsqueda de preguntas

Una vez lanzada la búsqueda remota a través de la interfaz anterior, los resultados se muestran en forma de tabla. Como resultado, además del texto de la pregunta, se muestra la visibilidad de la misma y la calificación media obtenida sobre un baremo de cinco estrellas, además de poder añadir la pregunta a la lista de preguntas para el examen.

Resultados Búsqueda			
Pregunta	Visibilidad	Calificación	
Ejemplo de Enunciado 1	Público	☆☆☆	
Ejemplo de Enunciado 2	Privado	☆☆☆☆☆	

[Volver](#)

Figura 74. Listado de preguntas encontradas

Además, desde la tabla anterior puede accederse a la información ampliada de la pregunta, donde pueden verse las respuestas a la misma así como el acceso a los comentarios realizados por otros usuarios y a las funciones de votar y comentar pregunta.

Resultados Búsqueda													
Pregunta	Visibilidad	Calificación											
Ejemplo de Enunciado 1	Público	☆☆☆											
<div style="border: 1px solid #ccc; padding: 5px;"> <p> Respuestas</p> <table border="1"> <thead> <tr> <th>Respuesta</th> <th>Correct</th> </tr> </thead> <tbody> <tr> <td>Respuesta correcta</td> <td></td> </tr> <tr> <td>Respuesta incorrecta 1</td> <td></td> </tr> <tr> <td>Respuesta incorrecta 2</td> <td></td> </tr> <tr> <td>Respuesta incorrecta 3</td> <td></td> </tr> </tbody> </table> <p> Comentarios</p> </div>				Respuesta	Correct	Respuesta correcta		Respuesta incorrecta 1		Respuesta incorrecta 2		Respuesta incorrecta 3	
Respuesta	Correct												
Respuesta correcta													
Respuesta incorrecta 1													
Respuesta incorrecta 2													
Respuesta incorrecta 3													
Ejemplo de Enunciado 2	Privado	☆☆☆☆☆											

[Volver](#)

Figura 75. Detalle de Preguntas encontradas

A continuación puede verse el panel extendido a través del cual se ven los comentarios y las funciones expuestas anteriormente.

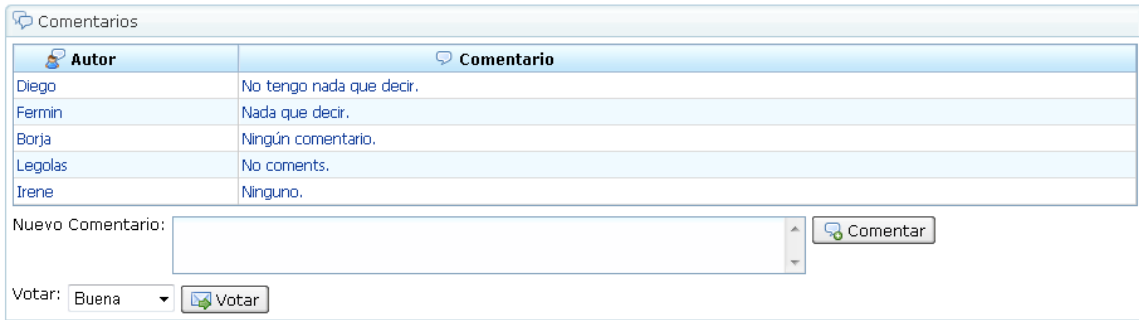


Figura 76. Comentarios realizados sobre una determinada pregunta recuperada

Nueva pregunta

Además de la adición de preguntas por búsqueda, también pueden añadirse preguntas nuevas mediante la siguiente interfaz.

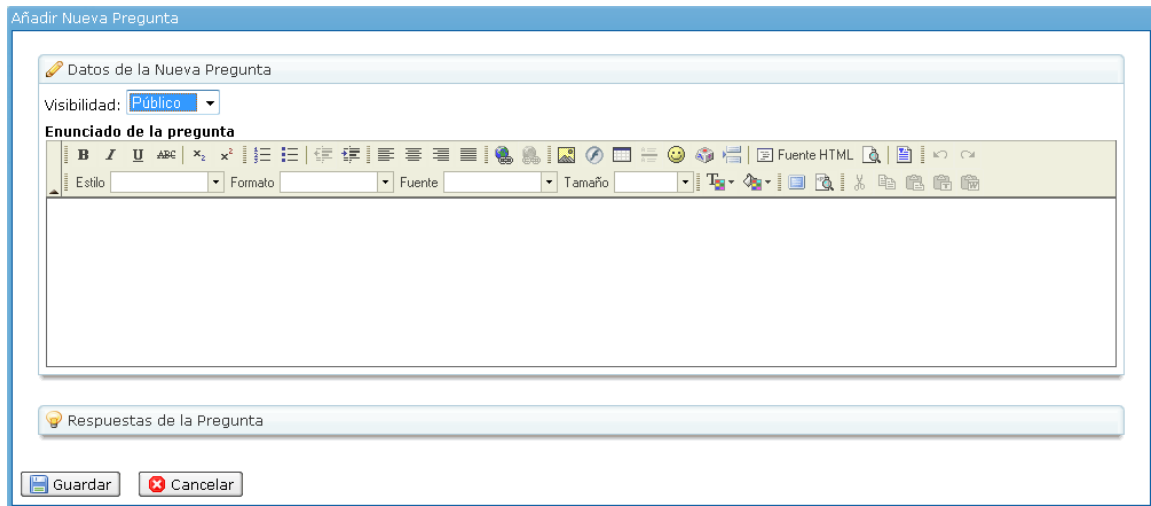


Figura 77. Nueva pregunta a añadir

Extendiendo el panel de respuestas pueden añadirse y editar aquellas que serán asociadas a la pregunta.

	Respuesta	Correcta	
+	Respuesta correcta 🧠	✓	👍
+	Respuesta incorrecta uno 🗨️	✗	👎
+	Respuesta incorrecta dos 🗨️	✗	👎
+	Respuesta incorrecta tres 🗨️	✗	👎

Figura 78. Listado de respuestas añadidas a una pregunta

Nueva respuesta

Al igual que con las preguntas y desde una pantalla análoga a las mismas, la cual se accede desde la lista de preguntas, se van añadiendo las distintas respuestas escribiendo el texto y seleccionando si es una respuesta correcta o no.

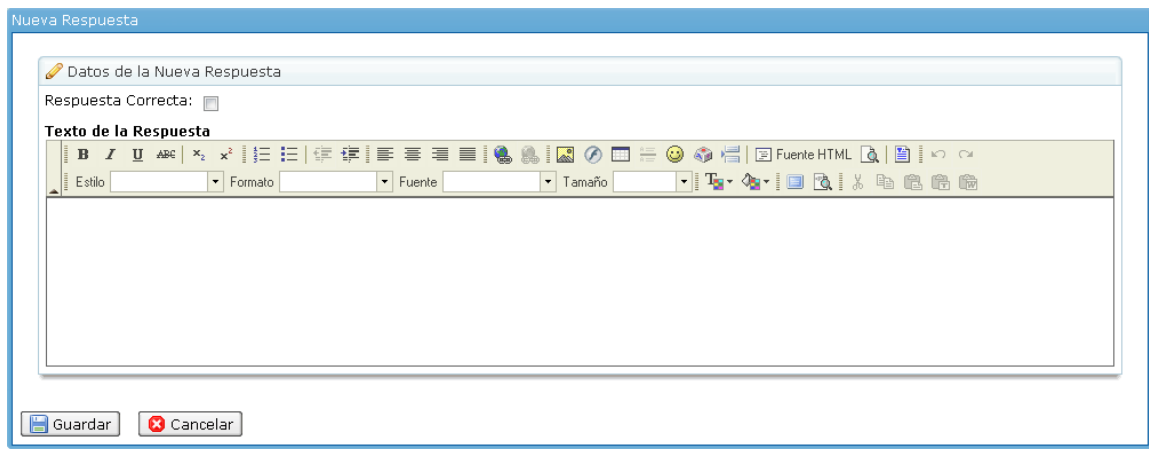


Figura 79. Nueva respuesta añadida a una pregunta

Edición de respuesta

Las respuestas previamente añadidas, pueden ser editadas a través de un panel accesible desde la propia lista de respuestas.

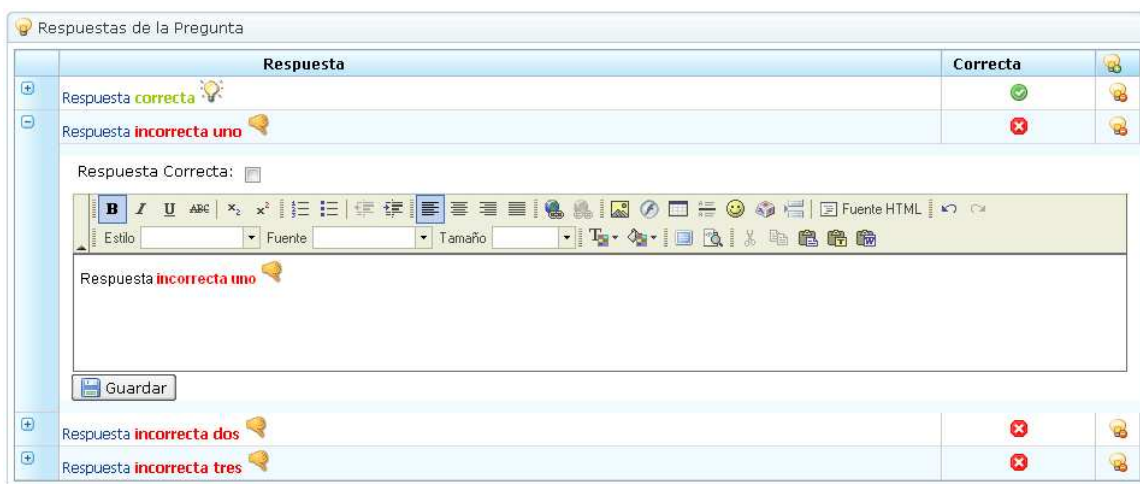


Figura 80. Edición de las respuestas añadidas

Edición de pregunta

Nuevamente, de manera análoga a las respuestas, cada uno de los parámetros que componen las preguntas, pueden ser editados a través de un panel accesible desde la tabla de preguntas que componen el examen.

Preguntas

Número de Preguntas del Examen: 1

Enunciado	Visibilidad
Texto de la Pregunta	Privado

Editar Pregunta

Visibilidad: Privado

Enunciado de la pregunta

Texto de la Pregunta

Guardar

Respuestas

Respuesta	Correcta
Respuesta correcta	✓
Respuesta incorrecta uno.	✗

Figura 81. Edición de las preguntas añadidas

Pruebas

En la presente sección se expondrán los resultados obtenidos de la ejecución del plan de pruebas tanto funcionales como no, sobre el desarrollo realizado dentro de la segunda iteración.

Pruebas Funcionales

Las pruebas funcionales tienen como objetivo principal, la verificación de la adecuación de la solución a las especificaciones expuestas en el análisis, comprobando que el sistema funciona tal como se describe en los documentos de análisis.

Cabe resaltar que para una correcta ejecución del plan de pruebas, éste debe ser ejecutado a partir de una base de datos vacía y siguiendo un riguroso orden de ejecución tal como marca el identificador dado a cada prueba.

El Identificador de las pruebas se compone de cuatro partes:

- Los caracteres PF seguidos de un guión indican que se trata de una prueba funcional.
- Los caracteres IT2 indican que se trata una prueba de la segunda iteración del desarrollo.
- Dos caracteres indicando el módulo de pruebas: configuración exámenes, configuración SOLE, módulo de comunicación y buscador semántico. (CE, CS, MC, BS respectivamente).
- Cuatro dígitos que indican unívocamente el número de prueba realizada.

A continuación se exponen los resultados obtenidos en las distintas pruebas:


ID:	<i>PF-IT2_CE_0019</i>	V.B. 
Nombre:	Generación examen sin preguntas.	
Descripción:	Se rellena el panel de Configuración de exámenes con datos aleatorios y se guarda el examen.	
Resultado Esperado:	En el selector de asignatura se deben cargar las asignaturas del sistema. Una vez pulsado el botón de guardar se debe haber creado un nuevo registro en la base de datos.	
Incidencia:	[No-Aplica]	

Figura 82. PF-IT2_CE_0019


ID:	PF-IT2_CE_0020	V.B.	
Nombre:	Validación de datos vacíos.		
Descripción:	Se dejan vacíos los campos del panel de configuración de exámenes y se pulsa el botón de guardar.		
Resultado Esperado:	El sistema debe dar errores informativos de validación y no insertar el registro del examen hasta encontrarse rellenos todos los campos.		
Incidencia:	El sistema da un error de BBDD y no está controlado a nivel de interfaz la mayoría de validaciones.		

Figura 83. PF-IT2_CE_0020


ID:	PF-IT2_CE_0021	V.B.	
Nombre:	Validación de periodos.		
Descripción:	Se introducen las fechas de periodos erróneas dentro del panel de configuración de exámenes y se pulsa el botón de guardar. Periodos erróneas son aquellos en los que la fecha de fin es anterior a la fecha de comienzo. Del mismo modo el periodo de revisión debe ser posterior o igual al periodo de realización.		
Resultado Esperado:	El sistema debe dar errores informativos de validación y no insertar el registro del examen hasta encontrarse rellenos todos los campos.		
Incidencia:	No se produce ninguna validación. Se insertar los registros sin ningún aviso ni error.		

Figura 84. PF-IT2_CE_0021


ID:	PF-IT2_CE_0022	V.B.	
Nombre:	Inserción de preguntas manualmente.		
Descripción:	Se inserta una pregunta “Publica” con el texto “Pregunta 1” y con dos respuestas una correcta y la otra no, cuyos textos serán “P1 Respuesta 1”, “P1 Respuesta 2”. Se inserta otra pregunta con texto “P2” y dos respuestas correctas con textos “P2 R1” y “P2 R2” respectivamente.		
Resultado Esperado:	Las preguntas deben aparecer en la tabla de preguntas. Del mismo modo al extender la información de las preguntas debe aparecer la información de las respuestas de cada una de ellas.		
Incidencia:	[No-Aplica]		

Figura 85. PF-IT2_CE_0022


ID:	PF-IT2_CE_0023	V.B. 
Nombre:	Edición Enunciado de la Pregunta.	
Descripción:	Se expande la edición de la pregunta. Se modifica el texto “Pregunta 1” por “P1” y se pulsa el botón guardar.	
Resultado Esperado:	El texto debe modificarse en el registro de la tabla.	
Incidencia:	[No-Aplica]	

Figura 86. PF-IT2_CE_0023


ID:	PF-IT2_CE_0024	V.B. 
Nombre:	Edición respuestas de la pregunta.	
Descripción:	Se elimina la respuesta “P1 Respuesta 2”. Se Edita el texto de la respuesta “P1 Respuesta 1” y se modifica por “P1 Respuesta 0”. Se añade la respuesta “Pr1 Respuesta 1” como pregunta incorrecta.	
Resultado Esperado:	Los cambios deben aparecer en la tabla de respuestas.	
Incidencia:	[No-Aplica]	

Figura 87. PF-IT2_CE_0024


ID:	PF-IT2_CE_0025	V.B. 
Nombre:	Eliminación de pregunta.	
Descripción:	Se elimina la pregunta con texto “P2”.	
Resultado Esperado:	La pregunta debe desaparecer de la tabla de preguntas.	
Incidencia:	[No-Aplica]	

Figura 88. PF-IT2_CE_0025


ID:	PF-IT2_CS_0026	V.B. 
Nombre:	Configuración SOLE	
Descripción:	Se estable el siguiente servidor dentro de la configuración SOLE: “http://192.168.0.128:8080/sole-ws/services/WSSole” y se pulsa el botón de guardar.	
Resultado Esperado:	El valor deber permanecer al reiniciar la página de la configuración.	
Incidencia:	[No-Aplica]	

Figura 89. PF-IT2_CS_00026


ID:	PF-IT2_MC_0027	V.B.	
Nombre:	Consulta dominios disponibles.		
Descripción:	Expandir el panel de “Búsqueda de Preguntas.		
Resultado Esperado:	Debe aparecer relleno el selector del dominio de búsqueda.		
Incidencia:	[No-Aplica]		

Figura 90. PF-IT2_MC_0027


ID:	PF-IT2_MC_00028	V.B.	
Nombre:	Búsqueda de preguntas.		
Descripción:	Lanzar una búsqueda de 10 registros, con una visibilidad “Publica” sobre el único dominio que existe para la cadena “Software”.		
Resultado Esperado:	Deben aparecer 10 preguntas relacionadas con la búsqueda.		
Incidencia:	El servicio de búsqueda no está disponible por lo que se simula la misma.		

Figura 91. PF-IT2_MC_00028


ID:	PF-IT2_BS_0029	V.B.	
Nombre:	Detalle de preguntas de búsqueda.		
Descripción:	Acceder a la información de cada resultado devuelto.		
Resultado Esperado:	Dentro de la tabla de resultados devueltos debe mostrarse el enunciado y la puntuación de la misma representado con cinco estrellas. Así mismo al expandir la información deben mostrarse las respuestas y los comentarios realizados sobre la pregunta. Los resultados devueltos deben corresponderse con la búsqueda realizada.		
Incidencia:	[No-Aplica]		

Figura 92. PF-IT2_BS_0029


ID:	PF-IT1_MC_0030	V.B.	
Nombre:	Comentar.		
Descripción:	Se introduce un comentario.		
Resultado Esperado:	El comentario debe aparecer en la tabla correspondiente.		
Incidencia:	El servicio de comentarios no está disponible.		

Figura 93. PF-IT1_MC_0030


ID:	<i>PF-IT1_MC_0031</i>	V.B. 
Nombre:	Votación.	
Descripción:	Se emite un voto aleatorio sobre la pregunta.	
Resultado Esperado:	El sistema debe validar el voto emitido desactivando la posibilidad de emitir otro voto. En el caso de producirse un error debe avisar del mismo.	
Incidencia:	El servicio de votación no está disponible.	

Figura 94. PF-IT1_MC_0031


ID:	<i>PF-IT1_CE_0032</i>	V.B. 
Nombre:	Añadir preguntas remotas	
Descripción:	Se añade una pregunta del resultado obtenido a la lista de preguntas.	
Resultado Esperado:	La pregunta debe desaparecer de los resultados obtenidos e introducirse en la tabla de preguntas del examen.	
Incidencia:	[No-Aplica]	

Figura 95. PF-IT1_CE_0032


ID:	<i>PF-IT1_CE_00033</i>	V.B. 
Nombre:	Adición Examen completo	
Descripción:	Se guarda el examen configurado.	
Resultado Esperado:	Debe almacenarse en BBDD todos los registros en las tablas de configuración, preguntas y respuestas correspondientes al examen generado cuyos datos pueden verse en el panel de configuración y en la tabla de preguntas.	
Incidencia:	[No-Aplica]	

Figura 96. PF-IT1_CE_00033


ID:	<i>PF-IT1_CE_0034</i>	V.B. 
Nombre:	Validación número de preguntas.	
Descripción:	Introducir un valor 100 en el campo “Número de preguntas del Examen” de la tabla preguntas.	
Resultado Esperado:	El sistema no debe permitir introducir un valor de preguntas por examen, superior al número de preguntas que existe en la tabla.	
Incidencia:	No se realiza dicha validación.	

Figura 97. PF-IT1_GP_0034

Seguimiento Planificación

Tal como puede observarse en el siguiente cronograma, la estimación y fecha propuesta de entrega ha cumplido con los plazos establecidos.

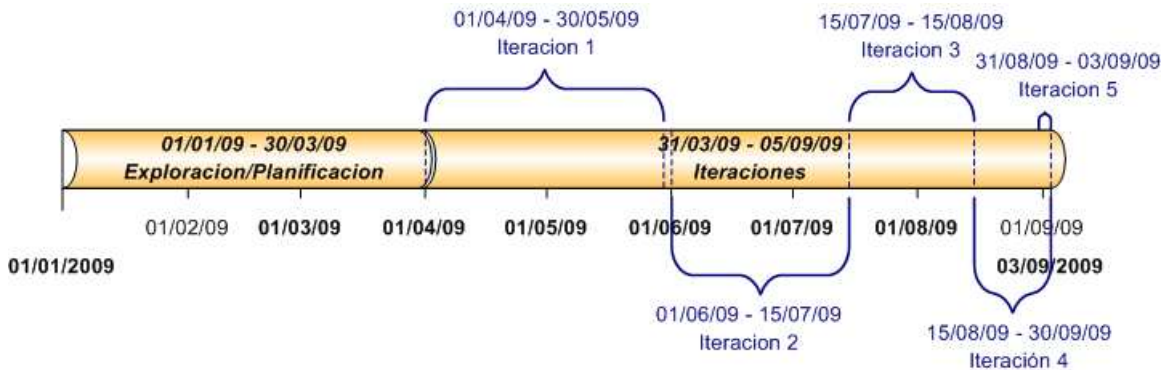


Figura 98. Cronograma con la actual planificación del proyecto

Tras una pequeña revisión de las tareas pendientes, así como de diversos factores, entre los que se incluye el próximo periodo estival y la disminución de tiempo de desarrollo y productividad; se observa un posible retraso en la entrega final del proyecto. Es por esto por lo que se decide establecer un periodo temporal de un mes desde la fecha programada a la fecha de presentación. Con todo esto, se propone el próximo día 2 de octubre como posible día de presentación del proyecto.

Validación del Documento

Próximos Hitos e Incidencias

ID	Tema	Estado	Responsable	Fecha Objetivo	Fecha de Impacto
7	Desarrollo iteración 3	Planificado	Borja Blanco	15/08/09	15/08/09
8	Error PF-IT2_CE_0020,0021,0034	Congelado	Borja Blanco	--/--/--	--/--/--
9	Error PF-IT2_MC_0030,0031	Pendiente	Diego Jiménez	10/08/09	--/--/--
10	Error PF-IT2_MC_00028	Pendiente	Diego Jiménez	11/09/09	--/--/--

Hitos e Incidencias Cerrados

ID	Tema	Resolución	Responsable	Fecha Objetivo	Fecha Cierre
1	Análisis y Planificación	Finalizado	Borja Blanco	01/04/09	01/04/09
2	Arquitectura de Sistema	Finalizado	Borja Blanco	10/04/09	09/05/09
4	Desarrollo iteración 1	Finalizado	Borja Blanco	01/05/09	30/05/09
3	Definición servicios SOLE	Finalizado	Diego Jiménez	15/04/09	01/07/09
5	Desarrollo iteración 2	Finalizado	Borja Blanco	15/07/09	15/07/09
6	Error PN-IT1_BD_00017	Finalizado	Borja Blanco	15/07/09	03/06/09

Aprobación del Documento

Rol	Nombre	Firma	Fecha
Autor del Proyecto	Borja Blanco Iglesias		
Tutor del Proyecto	Ricardo Colomo Palacios		

Documento Técnico de la Iteración 3



**Sistema de Gestión de
Exámenes Online
GEO**

**Informe Técnico sobre
Iteración 3
Madrid, 17 de Agosto de 2009**

Autor	Borja Blanco Iglesias
Fecha de Creación	19/08/2009
Fecha de actualización	19/08/2009
Nombre de documento	IT3_GEO.doc
Versión	1.0

Histórico de Versiones

Fecha	Autor	Versión	Comentarios
19/08/2009	Borja Blanco	1.0	Versión inicial del documento.

Revisores

Nombre	Posición
Borja Blanco Iglesias	Autor del Proyecto
Ricardo Colomo Palacios	Tutor del Proyecto

Lista de distribución

Copia N°	Nombre	Posición
1.	Borja Blanco Iglesias	Autor del Proyecto
2.	Ricardo Colomo Palacios	Tutor del Proyecto
3.	Diego Jiménez López	Colaborador
4.		
5.		

Introducción

El Informe técnico de una iteración del desarrollo de un sistema bajo la metodología XP, ofrece una visión global sobre el análisis realizado así como las decisiones de diseño e implementación que se han tomado a lo largo del presente desarrollo.

Objetivos del Documento

El documento tiene como objetivo principal ofrecer una visión global sobre el estado actual del proyecto, enfocado hacia las tareas realizadas sobre la iteración tres en todo el periodo de desarrollo.

A lo largo del documento se detallarán aquellos aspectos funcionales que contempla el desarrollo realizado como consecuencia de la iteración, así como las diversas decisiones de diseño tomadas a lo largo del desarrollo. Además, se incluirán los resultados obtenidos del proceso de pruebas realizado.

Por último, cabe resaltar la inclusión en el documento, de una revisión de la planificación realizada en función a los tiempos de desarrollo obtenidos, estableciendo las comparaciones oportunas con los tiempos planificados.

Audiencia

Debido al carácter técnico, el presente documento está dirigido a aquellas personas que tomen parte en tareas de desarrollo del proyecto y necesiten acceso a la documentación sobre las tareas realizadas en la iteración tres. Así mismo, por la información general tratada en el mismo, la audiencia del documento se extiende a la dirección del proyecto quien deberá validar los requisitos definidos como base del desarrollo.

Definiciones, acrónimos, abreviaturas y referencias

A continuación se incluye información relativa a la terminología utilizada en el documento así como a las referencias realizadas en el mismo.

- **EJB:** Acrónimo del inglés *Enterprise Java Bean* es una API dentro del estándar JEE para la comunicación remota de servicios.
- **MDA:** Acrónimo del inglés *Model driven architecture* cuyo significado es arquitectura dirigida por el modelo.
- **XP (Extreme Programming):** Metodología de desarrollo ágil cuya traducción al español es Programación Extrema.

Definición del alcance del desarrollo

Dentro del ámbito del proyecto la presente iteración contempla el desarrollo de los siguientes módulos:

Generación y realización del Examen:

Partiendo de las configuraciones de exámenes creadas por medio de los módulos definidos en la iteración anterior, el alumno puede solicitar la realización de un examen definido en tales configuraciones. Para ello, el examen debe estar en vigor y debe pertenecer a una asignatura en la que esté matriculado el alumno. Una vez validados los permisos del usuario, se procede a generar el examen mostrando las preguntas correspondientes. El sistema generará un examen aleatorio con las preguntas seleccionadas por el profesor, en el caso de que en el momento de la configuración el profesor lo haya establecido así. El alumno dispondrá del tiempo definido en la configuración del examen para realizarlo. Si por cualquier causa el alumno abandona la sesión, el examen estará activo y el sistema debe dar la posibilidad de seguir con el mismo con el tiempo restante para su finalización. Una vez expirado dicho tiempo, o bien una vez haya creído oportuno el alumno, el examen se dará por finalizado eliminando la posibilidad de seguir respondiendo. Si por cualquier causa, el alumno abandonara el sistema sin finalizar el examen previamente, éste se dará por finalizado automáticamente una vez finalizado el plazo con las últimas respuestas marcadas por el alumno.

Corrección de Exámenes:

Una vez finalizado un examen, ya sea por expiración del plazo establecido o manualmente por el alumno, el sistema procederá a corregirlo automáticamente. Dicha corrección registrará la nota individual de cada pregunta así como la total de la evaluación. En el caso de que el profesor lo estableciera así a la hora de crear la configuración del examen, el resultado total del examen será mostrado al usuario en una pantalla resumen al finalizar la prueba.

Revisión de Exámenes:

Tras la realización de un examen, el alumno podrá realizar la revisión del mismo siempre que lo haga en el plazo establecido a tal efecto en la configuración. En dicha revisión el alumno podrá visualizar las respuestas correctas, comparar con sus respuestas, ver la puntuación de cada pregunta, y puntuación total obtenida, así como de escribir sus propios comentarios de revisión en cada una de las preguntas y de forma general al examen. Así mismo, se dará la posibilidad de solicitar una revisión al profesor para que éste pueda valorar los comentarios realizados, contestarlos y si lo cree oportuno modificar la nota global del examen. Para ello el profesor tendrá acceso a distintos gráficos para ver de forma visual las respuestas dadas por los alumnos a cada pregunta así como la distribución de las notas.

Diseño del Sistema

En la presente sección se muestran los resultados obtenidos del proceso de diseño dentro de la iteración tres. Estos resultados se engloban en dos grandes áreas, el modelo de datos que formaría el *back-end* de la solución propuesta, así como el diseño de las interfaces de usuario desarrolladas, lo que formaría parte del *front-end*.

Modelo de datos

El diseño del modelo puede verse a través de los siguientes diagramas de clases. En ambos, puede observarse el modelo utilizado directamente en la persistencia de los datos de la iteración, así como los distintos *Value Objects* utilizados en el mapeo de datos entre Beans persistentes e interfaz gráfica.

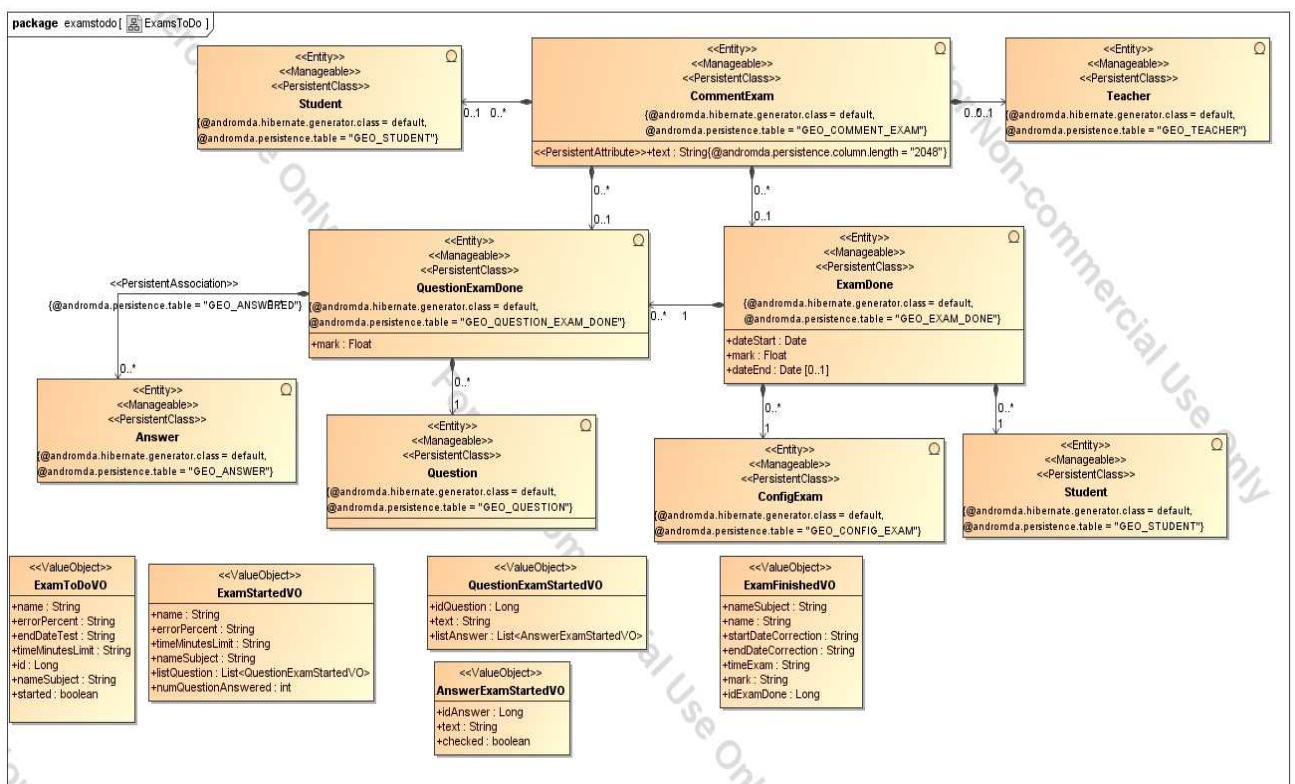


Figura 99. Modelo de negocio persistente relacionado con la realización de exámenes

Si el diagrama anterior representaba y definía el modelo persistente utilizado específicamente por el dominio de generación, realización y corrección de exámenes; a continuación se muestra el modelo utilizado por el dominio de revisión para la persistencia y consulta de los datos relacionados con el mismo.

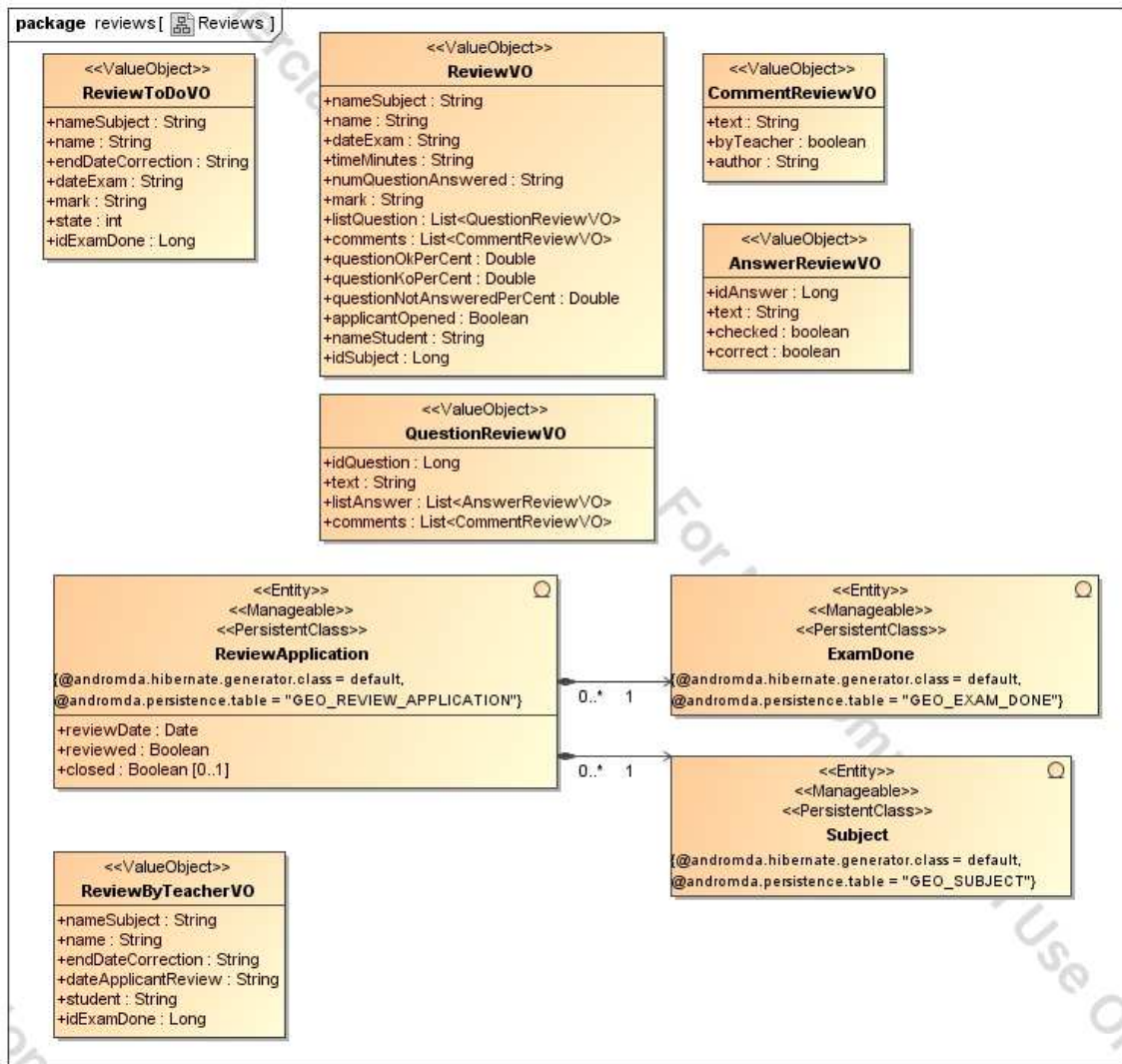


Figura 100. Modelo de negocio persistente relacionado con la revisión de exámenes

El resultado de la utilización de la herramienta de generación de código AndromDA, sobre el presente modelo de clases estereotipadas, son una serie de servicios implementados sobre EJB's, los cuales dan acceso a los datos de persistencia, encapsulados en una serie de librerías, que son importadas dentro de la aplicación desarrollada.

A continuación se hará una exposición detallada de las clases que componen el modelo de negocio utilizado en la presente iteración.

AnswerExamStartedVO – Value Object Respuesta de Examen Comenzado

org.babieca.geo.ddbb.examstodo.AnswerExamStartedVO	
Nombre de Clase:	<ul style="list-style-type: none"> • AnswerExamStartedVO
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.examstodo
Estereotipos:	<ul style="list-style-type: none"> • ValueObject
Atributos:	<ul style="list-style-type: none"> • idAnswer (Long) – Identificador de la respuesta. • text (String) – Texto de la respuesta. • checked (Boolean) – Flag que identifica si fue respondida.
Métodos:	-----
Relaciones:	-----

Tabla 38. Clase AnswerExamStartedVO

CommentExam – Entidad Comentario del examen

org.babieca.geo.ddbb.examstodo.CommentExam	
Nombre de Clase:	<ul style="list-style-type: none"> • CommentExam
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.examstodo
Estereotipos:	<ul style="list-style-type: none"> • Entity • Manageable • PersistentClass
Atributos:	<ul style="list-style-type: none"> • text (String) – Texto del comentario.
Métodos:	-----
Relaciones:	<ul style="list-style-type: none"> • Student • QuestionExamDone • ExamDone • Teacher

Tabla 39. Clase CommentExam

ExamDone – Entidad del Examen Realizado

org.babieca.geo.ddbb.examstodo.ExamDone	
Nombre de Clase:	<ul style="list-style-type: none"> ExamDone
Paquete:	<ul style="list-style-type: none"> org.babieca.geo.ddbb.examstodo
Estereotipos:	<ul style="list-style-type: none"> Entity Manageable PersistentClass
Atributos:	<ul style="list-style-type: none"> dateStart (Date) – Día de comienzo del examen. mark (Float) – Calificación del examen. dateEnt (Date) – Día de finalización del examen.
Métodos:	-----
Relaciones:	<ul style="list-style-type: none"> ConfigExam Student QuestionExamDone

Tabla 40. Clase ExamDone

ExamFinishedVO – Value Object del Examen Finalizado

org.babieca.geo.ddbb.examstodo.ExamFinishedVO	
Nombre de Clase:	<ul style="list-style-type: none"> ExamFinishedVO
Paquete:	<ul style="list-style-type: none"> org.babieca.geo.ddbb.examstodo
Estereotipos:	<ul style="list-style-type: none"> ValueObject
Atributos:	<ul style="list-style-type: none"> nameSubject (String) – Nombre de la asignatura. name (String) – Nombre descriptivo del examen. startDateCorrection (String) – Día de inicio de la corrección endDateCorrectoin (String) – Día de fin de la corrección. timeExam (String) – Duración del examen. mark (String) – Calificación del examen. idExamDone (Long) – Identificador del ExamDone asociado.
Métodos:	-----
Relaciones:	-----

Tabla 41. Clase ExamFinishedVO

ExamStartedVO – Value Object del Examen Comenzado

org.babieca.geo.ddbb.examstodo.ExamStartedVO	
Nombre de Clase:	<ul style="list-style-type: none"> • ExamStartedVO
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.examstodo
Estereotipos:	<ul style="list-style-type: none"> • ValueObject
Atributos:	<ul style="list-style-type: none"> • nameSubject (String) – Nombre de la asignatura. • name (String) – Nombre descriptivo del examen. • errorPercent (String) - Porcentaje de error de la respuesta incorrecta. • timeMinutesLimit (String) – Límite de tiempo en minutos para realizar el examen. • listQuestion (List<QuestionExamStartedVO>) – Lista de preguntas del examen. • numQuestionAnswered (int) - Número de preguntas respondidas.
Métodos:	-----
Relaciones:	-----

Tabla 42. Clase ExamStartedVO

ExamToDoVO – Value Object del Examen a Realizar

org.babieca.geo.ddbb.examstodo.ExamToDoVO	
Nombre de Clase:	<ul style="list-style-type: none"> • ExamToDoVO
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.examstodo
Estereotipos:	<ul style="list-style-type: none"> • ValueObject
Atributos:	<ul style="list-style-type: none"> • nameSubject (String) – Nombre de la asignatura. • name (String) – Nombre descriptivo del examen. • errorPercent (String) - Porcentaje de error de la respuesta incorrecta. • timeMinutesLimit (String) – Límite de tiempo en minutos para realizar el examen. • endDateTest (String) – Dia de finalización del examen. • id (Long) - Identificador del Examen a realizar. • started (boolean) – Flag que indica si ha empezado el examen o no.
Métodos:	-----
Relaciones:	-----

Tabla 43. Clase ExamToDoVO

QuestionExamDone – Entidad de la Pregunta del Examen Realizado

org.babieca.geo.ddbb.examstodo.QuestionExamDone	
Nombre de Clase:	<ul style="list-style-type: none"> • QuestionExamDoneVO
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.examstodo
Estereotipos:	<ul style="list-style-type: none"> • Entity • Manageable • PersistentClass
Atributos:	<ul style="list-style-type: none"> • mark (Float) – Calificación de la pregunta.
Métodos:	-----
Relaciones:	<ul style="list-style-type: none"> • Question. • Answer. • CommentExam • ExamDone

Tabla 44. Clase QuestionExamDone

QuestionExamStartedVO – Value Object con la pregunta del examen

org.babieca.geo.ddbb.examstodo.QuestionExamStartedVO	
Nombre de Clase:	<ul style="list-style-type: none"> • QuestionExamStartedVO
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.examstodo
Estereotipos:	<ul style="list-style-type: none"> • ValueObject
Atributos:	<ul style="list-style-type: none"> • idQuestion (Long) – Identificador de la pregunta. • text (String) – Texto de la pregunta. • listAnswer (List<AnswerExamStartedVO>) – Lista de respuestas de la pregunta.
Métodos:	-----
Relaciones:	-----

Tabla 45. Clase QuestionExamStartedVO

AnswerReviewVO – Value Object con la respuesta a Revisar

org.babieca.geo.ddbb.reviews.AnswerReviewVO	
Nombre de Clase:	<ul style="list-style-type: none"> • AnswerReviewVO
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.reviews
Estereotipos:	<ul style="list-style-type: none"> • ValueObject
Atributos:	<ul style="list-style-type: none"> • idAnswer (Long) – Identificador de la respuesta. • text (String) – Texto de la respuesta. • checked (boolean) – Flag que indica si ha sido respondida. • correct (boolean) – Flag que indica si es correcta.
Métodos:	-----
Relaciones:	-----

Tabla 46. Clase AnswerReviewVO

CommentReviewVO – Value Object con los comentarios de la revisión.

org.babieca.geo.ddbb.reviews.CommentReviewVO	
Nombre de Clase:	<ul style="list-style-type: none"> • CommentReviewVO
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.reviews
Estereotipos:	<ul style="list-style-type: none"> • ValueObject
Atributos:	<ul style="list-style-type: none"> • text (String) – Texto del comentario. • byTeacher (boolean) – Flag que indica si es un profesor el autor. • author (String) – Nombre del autor del comentario.
Métodos:	-----
Relaciones:	-----

Tabla 47. Clase CommentReviewVO

QuestionReviewVO – Value Object con la pregunta a Revisar

org.babieca.geo.ddbb.examstodo.QuestionReviewVO	
Nombre de Clase:	<ul style="list-style-type: none"> • QuestionReviewVO
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.reviews
Estereotipos:	<ul style="list-style-type: none"> • ValueObject
Atributos:	<ul style="list-style-type: none"> • idQuestion (Long) – Identificador de la pregunta a revisar. • text (String) – Texto de la pregunta. • listAnswer (List<AnswerReviewVO>) – Lista de respuestas. • comments (List<CommentReviewVO>) – Lista de comentarios.
Métodos:	-----
Relaciones:	-----

Tabla 48. Clase QuestionReviewVO

ReviewApplication – Entidad de la solicitud de revision

org.babieca.geo.ddbb.examstodo.ReviewApplication	
Nombre de Clase:	<ul style="list-style-type: none"> • ReviewApplication
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.reviews
Estereotipos:	<ul style="list-style-type: none"> • Entity • Manageable • PersistentClass
Atributos:	<ul style="list-style-type: none"> • reviewDate (Date) – Día de la revisión. • reviewed(boolean) – Flag que marca si ha sido revisado. • closed(boolean) – Flag que marca si ha sido cerrada.
Métodos:	-----
Relaciones:	<ul style="list-style-type: none"> • ExamDone • Subject

Tabla 49. Clase ReviewApplication

ReviewByTeacherVO – Value Object con la Revisión por el Profesor

org.babieca.geo.ddbb.examstodo. ReviewByTeacherVO	
Nombre de Clase:	<ul style="list-style-type: none"> • ReviewByTeacherVO
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.reviews
Estereotipos:	<ul style="list-style-type: none"> • ValueObject
Atributos:	<ul style="list-style-type: none"> • nameSubject (String) – Nombre de la asignatura. • name (String) – Nombre descriptivo del examen. • endDateCorrection (String) – Fecha de fin de la corrección • dateApplicantReview (String) – Fecha de solicitud de revisión. • student (String) – Alumno que solicita la revisión. • idExamDone (Long) – Identificador del examen del que se solicita la revisión.
Métodos:	-----
Relaciones:	-----

Tabla 50. Clase ReviewByTeacherVO

ReviewToDoVO – Value Object con la Revisión para realizar

org.babieca.geo.ddbb.examstodo. ReviewToDoVO	
Nombre de Clase:	<ul style="list-style-type: none"> ReviewToDoVO
Paquete:	<ul style="list-style-type: none"> org.babieca.geo.ddbb.reviews
Estereotipos:	<ul style="list-style-type: none"> ValueObject
Atributos:	<ul style="list-style-type: none"> nameSubject (String) – Nombre de la asignatura. name (String) – Nombre descriptivo del examen. endDateCorrection (String) – Fecha de fin de la corrección dateExam (String) – Fecha de realización del examen. mark (String) – Calificación del examen. state (int) – Estado de la revisión idExamDone (Long) – Identificador del examen del que se solicita la revisión.
Métodos:	-----
Relaciones:	-----

Tabla 51. Clase ReviewToDoVO

ReviewVO – Value Object con la Revisión

org.babieca.geo.ddbb.examstodo. ReviewVO	
Nombre de Clase:	<ul style="list-style-type: none"> • ReviewVO
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.reviews
Estereotipos:	<ul style="list-style-type: none"> • ValueObject
Atributos:	<ul style="list-style-type: none"> • nameSubject (String) – Nombre de la asignatura. • name (String) – Nombre descriptivo del examen. • timeMinutes (String) – Tiempo límite para la realización del examen. • dateExam (String) – Fecha de realización del examen. • mark (String) – Calificación del examen. • numQuestionAnswered (String) – Número de preguntas respondidas. • listQuestion (List<QuestionReviewVO>) – Lista de preguntas a revisar. • comments (List<CommentReviewVO>) – Lista de comentarios de la revisión. • questionOkPerCent (Double) – Numero de preguntas correctas. • questionKoPerCent (Double) – Numero de preguntas incorrectas. • questionNotAnswered (Double) – Numero de preguntas no respondidas. • applicantOpened (Boolean) – Flag que marca si existe una solicitud abierta. • nameStudent (String) – Nombre del alumno. • idSubject (Long) – Identificador de la asignatura.
Métodos:	-----
Relaciones:	-----

Tabla 52. Clase ReviewVO

Interfaz Gráfica de Usuario

Completando al modelo de negocio en el ámbito del diseño, se presentan las distintas Interfaces de usuario implementadas en la presente iteración.

Realización de Examen

Todo alumno tiene acceso a la lista de exámenes a realizar. Mediante esta lista, podrá ver los datos más relevantes del examen, además de tener acceso a la realización del mismo. Una vez iniciado el examen, el tiempo de realización empezará a contar. Si un alumno abandona un examen existirá la posibilidad de volver a entrar, a través del mismo listado durante el periodo en el que dure el examen. Todo examen ya iniciado es marcado en el listado a través de un símbolo de atención.

Asignatura	Examen	Fecha Fin	Tiempo	Penalización Error
Teoría de Automatas y Lenguajes Formales	Examen Test - 1	23/08/2009 11:17:00 PM	60 m.	33 %


Figura 101. Listado de Exámenes a Realizar

La pantalla de realización del examen se compone de una cabecera, capaz de ocultarse, y de un panel principal que contiene cada una de las preguntas, así como las posibles respuestas. Cuando una respuesta es marcada, ésta es registrada automáticamente por el sistema. De esta forma si un examen finaliza de manera inesperada, las últimas respuestas marcadas serán las que formen parte de la evaluación del examen.

Datos del Examen

Asignatura:	Teoría de Automatas y Lenguajes Formales
Examen:	Examen Test - 1
Tiempo Restante:	60 minutos
Preguntas Respondidas:	1 de 4
Penalización Error:	33 %

Pregunta 2



¿Cual es el nombre del caballo que aparece en la imagen al cual se hace referencia en el siguiente fragmente del Cantar del mio Cid?

*"Mío Cid el bienhadado se retrasaba:
túnica de seda viste, muy crecida trae la barba,
ya le ensillan a Babieca, muy bien que le enjaezaban,
se monta en él Mio Cid y armas de palo tomaba.
En el nombrado Babieca el Campeador cabalga,
arranca a correr y dio una carrera tan rauda
que todos los que le vieron maravillados estaban.
Desde aquel día Babieca fue famoso en toda España.
Al acabar la carrera ya Mio Cid descabalgó."*

Felipe

Babieca

Rocnante

Ginebra

Figura 102. Realización del Examen

Un examen puede finalizar por la expiración del tiempo disponible, o bien por petición del usuario. En cualquier caso, al finalizar el examen se muestra una pantalla resumen con los distintos datos de la realización del mismo.

Exam Finalizado	
Asignatura:	Teoría de Autómatas y Lenguajes Formales
Examen:	Examen Test - 1
Tiempo Realización:	3 minutos
Periodo Revisión:	Desde 17/08/2009 11:17:00 PM hasta 30/08/2009 11:17:00 PM
Calificación:	4.18 puntos sobre 10

Comentarios del examen	
La primera pregunta no pertenecía al temario anunciado. No debería entrar en el examen.	

Comentar

Figura 103. Finalización del Examen

Desde esta misma pantalla de finalización, el alumno podrá expresar algunos comentarios del examen, los cuales serán tenidos en cuenta en el momento de la revisión posterior.

Revisión del Examen por el Alumno

Al igual que ocurriera con los exámenes a realizar, el alumno también tendrá accesible el listado de aquellos exámenes realizados, que estén en periodo de revisión. Desde este listado no sólo podrá acceder a la propia revisión, sino que podrá ver los detalles más importantes del examen, así como ver el estado de aquellas revisiones solicitadas mediante un sencillo icono.

	Asignatura	Examen	Fecha Realización	Fecha Límite	Puntuacio	
	Teoría de Autómatas y Lenguajes Formales	Examen Test - 1	18/08/2009 05:59:37 PM	30/08/2009 11:17:00 PM	4.18	

Figura 104. Listado de Exámenes a Revisar por el Alumno

La pantalla de revisión se encuentra dividida en tres apartados accesibles mediante unas pestañas superiores. Así, nos encontramos con una pantalla resumen del examen, la revisión individual de cada pregunta y un último panel donde expresar y ver los comentarios generales del examen.

Al iniciar la revisión, por defecto, aparecerá el resumen del examen desde donde además de ver los detalles más importantes del mismo, se tiene acceso a un gráfico el cual ofrece una visión global de la distribución de errores, aciertos y preguntas sin responder del examen.



Figura 105. Datos Generales del Examen en la Revisión

La revisión individual de la pregunta permite comparar las respuestas dadas durante el examen con las respuestas correctas. Además, se pueden realizar comentarios específicos para cada pregunta.

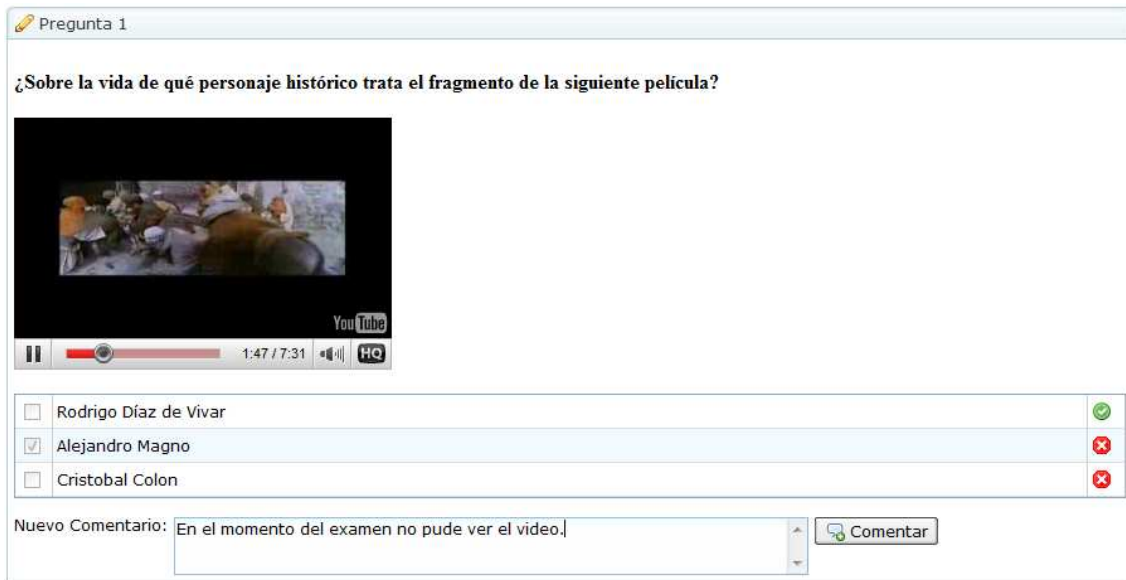


Figura 106. Revisión Individual de Preguntas

Para finalizar la revisión, existe un último panel en el que visualizar y realizar aquellos comentarios de carácter general del examen.

Datos Examen	Revisión Preguntas	Comentarios Examen
Autor	Comentario	
borjabi	La primera pregunta no pertenecía al temario anunciado. No debería entrar en el examen.	

Nuevo Comentario: Solicito la exclusión de la primera y tercera pregunta del examen. | Comentar

Solicitar Revisión

Figura 107. Comentarios Generales del Examen

Cabe resaltar que durante toda la revisión se ofrece la posibilidad de solicitar una revisión del examen por parte del profesor mediante el botón “Solicitar Revisión”.

Revisión del Examen por el Profesor

Una vez el alumno haya solicitado la revisión de un examen, ésta será visible por el profesor mediante el listado correspondiente. Dicho listado contará con los detalles más importantes del examen realizado por el alumno, así como el acceso a la propia revisión.

Asignatura	Examen	Fecha Solicitud	Fecha Límite	Alumno	
Teoría de Autómatas y Lenguajes Formales	Examen Test - 1	18/08/2009 06:23:25 PM	30/08/2009 11:17:00 PM	Borja Blanco	

Figura 108. Listado de Exámenes a Revisar por el Profesor

Al igual que ocurriera en el caso de los alumnos, la pantalla de revisión del profesor se encuentra dividida en tres apartados accesibles mediante unas pestañas superiores. De la misma forma, nos encontramos con una pantalla resumen del examen, la revisión individual de cada pregunta y un último panel donde poder visualizar los comentarios realizador por el alumno así como contestar a los mismos.

Nuevamente al iniciar la revisión, por defecto, aparecerá el resumen del examen donde además de ver los detalles más importantes del mismo, se tiene acceso a un gráfico el cual ofrece una visión global de la distribución de errores, aciertos y preguntas sin responder del examen.

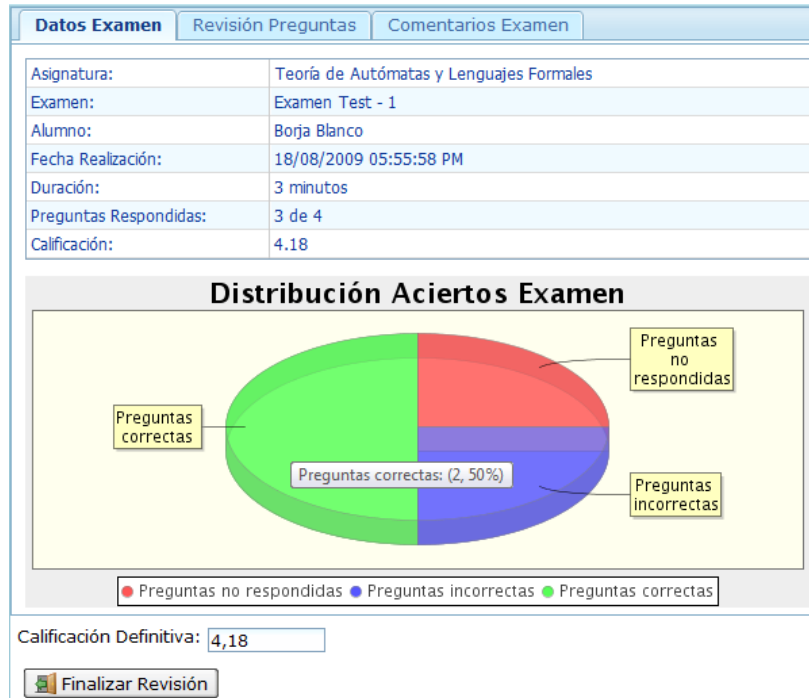


Figura 109. Datos Generales del Examen en la Revisión

De manera análoga a la revisión individual de preguntas del alumno, en este caso el profesor podrá comparar las respuestas, ver los comentarios del alumno así como añadir los suyos propios.

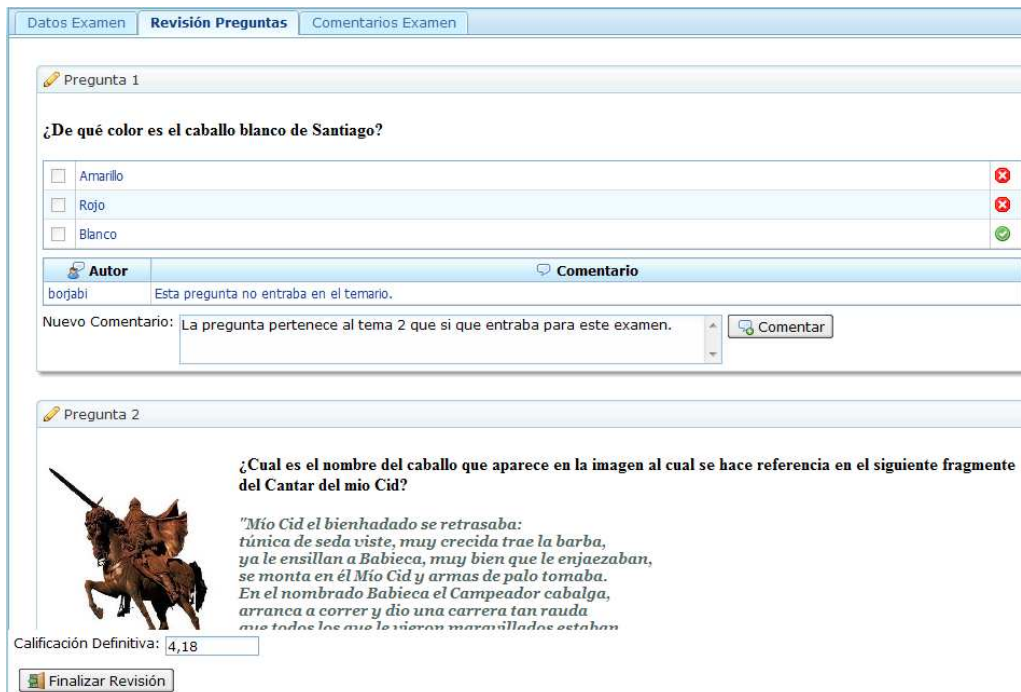
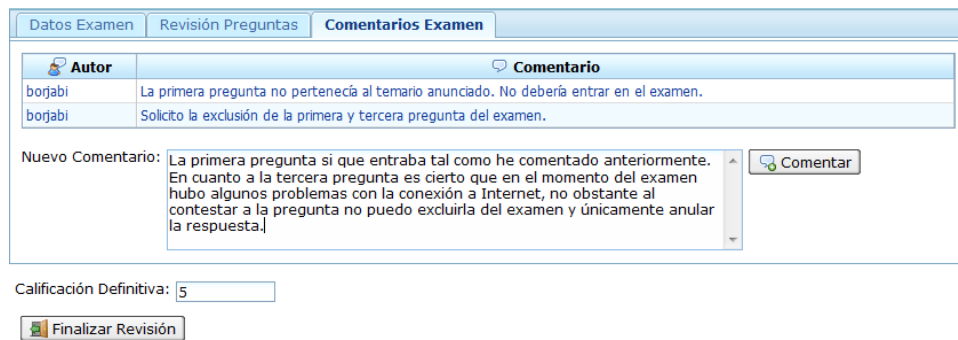


Figura 110. Revisión Individual de Preguntas

La revisión consta de una última pantalla idéntica a la del alumno, en donde el profesor podrá visualizar los comentarios del alumno realizados en todas las revisiones, así como contestar a los mismos.



The screenshot shows a web interface for exam comments. At the top, there are three tabs: 'Datos Examen', 'Revisión Preguntas', and 'Comentarios Examen'. Below the tabs is a table with two columns: 'Autor' and 'Comentario'. The table contains two rows of comments from a user named 'borjabi'. Below the table is a 'Nuevo Comentario:' section with a text area containing a new comment and a 'Comentar' button. At the bottom, there is a 'Calificación Definitiva:' field with the value '5' and a 'Finalizar Revisión' button.

Autor	Comentario
borjabi	La primera pregunta no pertenecía al temario anunciado. No deberá entrar en el examen.
borjabi	Solicito la exclusión de la primera y tercera pregunta del examen.

Nuevo Comentario: La primera pregunta si que entraba tal como he comentado anteriormente. En cuanto a la tercera pregunta es cierto que en el momento del examen hubo algunos problemas con la conexión a Internet, no obstante al contestar a la pregunta no puedo excluirla del examen y únicamente anular la respuesta.

Calificación Definitiva: 5

Finalizar Revisión

Figura 111. Comentarios Generales del Examen

Nuevamente, resaltamos la posibilidad de finalizar la revisión a través del botón dispuesto para tal efecto, disponible a lo largo de la misma. Además, el profesor podrá modificar la calificación final del examen libremente, mediante el cuadro de texto situado encima del botón de finalización.

Pruebas

En la presente sección se expondrán los resultados obtenidos de la ejecución del plan de pruebas tanto funcionales como no, sobre el desarrollo realizado dentro de la segunda iteración.

Pruebas Funcionales

Las pruebas funcionales tienen como objetivo principal, la verificación de la adecuación de la solución a las especificaciones expuestas en el análisis, comprobando que el sistema funciona tal como se describe en los documentos de análisis.

El Identificador de las pruebas se compone de cuatro partes:

- Los caracteres PF seguidos de un guión indican que se trata de una prueba funcional.
- Los caracteres IT3 indican que se trata una prueba de la tercera iteración del desarrollo.
- Dos caracteres indicando el módulo de pruebas: generación y realización de examen, corrección de exámenes y la revisión de los mismos. (GE, OE, RE respectivamente).
- Cuatro dígitos que indican unívocamente el número de prueba realizada.

Cabe resaltar que para una correcta ejecución del plan de pruebas, éste debe ser ejecutado siguiendo un riguroso orden de ejecución tal como marca el identificador dado a cada prueba y con una base de datos en la que previamente hayan sido cargados los siguientes registros:

- Asignaturas. Debe existir dos asignaturas cuyos nombres sean “Asignatura1” y “Asignatura2”.
- Alumnos. Debe existir un alumno de nombre “Alumno1” que no esté matriculado en ninguna asignatura cuyos datos personales son indiferentes.
- Profesores. Debe existir un profesor de nombre “Profesor1” que no esté asignado a ninguna asignatura cuyos datos personales son indiferentes.

Por último para evitar errores en los log de pruebas, hay que asegurarse de haber configurado correctamente la URL del servidor de preguntas SOLE.

A continuación se exponen los resultados obtenidos en las distintas pruebas:

ID:	<i>PF-IT3_GE_0035</i>	V.B. ✓
Nombre:	Filtrado de listado de exámenes por asignatura 1.	
Descripción:	Se generan dos exámenes con nombre “Examen1” y “Examen2” de la asignatura “Asignatura1” y “Asignatura2” respectivamente, con un periodo de realización dentro del actual.	
Resultado Esperado:	Al visualizar el listado de exámenes a realizar éste debe aparecer vacío.	
Incidencia:	[No-Aplica]	

Figura 112. PF-IT3_GE_0035

ID:	<i>PF-IT3_GE_0036</i>	V.B. ✓
Nombre:	Filtrado de listado de exámenes por asignatura 2.	
Descripción:	Se matricula el “Alumno1” en la “Asignatura1”.	
Resultado Esperado:	Al visualizar el listado de exámenes a realizar debe aparecer con los datos de configuración el “Examen1”.	
Incidencia:	[No-Aplica]	

Figura 113. PF-IT3_GE_0036

ID:	<i>PF-IT3_GE_0037</i>	V.B. ✓
Nombre:	Filtrado de listado de exámenes por periodo.	
Descripción:	Se crea un nuevo examen “Examen3” cuyo periodo de realización sea posterior al actual.	
Resultado Esperado:	Al visualizar el listado de exámenes a realizar debe aparecer con los datos de configuración el “Examen1”.	
Incidencia:	[No-Aplica]	

Figura 114. PF-IT3_GE_0037

ID:	<i>PF-IT3_GE_0038</i>	V.B. ✓
Nombre:	Generación Aleatoria del Examen	
Descripción:	Se crea el “Examen4” con periodos de realización actuales, con cuatro preguntas añadidas y con un tres en “Número de Preguntas del Examen”.	
Resultado Esperado:	Al realizar el examen deben aparecer tres preguntas aleatorias. Para comprobar la aleatoriedad borrar los registros de las tablas “ExamDone” y “QuestionExamDone”.	
Incidencia:	[No-Aplica]	

Figura 115. PF-IT3_GE_0038

ID:	<i>PF-IT3_GE_0039</i>	V.B. ✓
Nombre:	Parámetros de evaluación.	
Descripción:	Se crea el “Examen5” con periodos de realización actuales, con cuatro preguntas con “10” como puntuación máxima y un “20%” como penalización por error. Responder correctamente a una de las preguntas, incorrectamente a otra y dejar sin responder a la cuarta.	
Resultado Esperado:	Al finalizar el examen la nota debe ser 4.5.	
Incidencia:	[No-Aplica]	

Figura 116. PF-IT3_GE_0039

ID:	<i>PF-IT3_GE_0040</i>	V.B. ✓
Nombre:	Evaluación no automática.	
Descripción:	Se crea el “Examen6” con el parámetro de “Corrección Automática” deseleccionado y realizar el examen.	
Resultado Esperado:	Al finalizar el examen en el campo calificación debe aparecer “Disponible en la revisión”.	
Incidencia:	[No-Aplica]	

Figura 117. PF-IT3_GE_0040

ID:	<i>PF-IT3_GE_0041</i>	V.B. ✓
Nombre:	Finalización por expiración del tiempo.	
Descripción:	Se crea el “Examen7” con un tiempo de 1 minuto y se entra en el examen esperando que expire el tiempo.	
Resultado Esperado:	El examen debe finalizar automáticamente.	
Incidencia:	[No-Aplica]	

Figura 118. PF-IT3_GE_0041

ID:	<i>PF-IT3_GE_0042</i>	V.B. ✓
Nombre:	Readmisión al examen	
Descripción:	Se crea el “Examen8”. Se entra en el examen y se sale sin finalizar.	
Resultado Esperado:	Al visualizar los exámenes disponibles debe aparecer el examen, con el tiempo restante y con el aviso de examen empezado. Además debe permitir volver a entrar apareciendo el tiempo restante desde el comienzo.	
Incidencia:	[No-Aplica]	

Figura 119. PF-IT3_GE_0042


ID:	<i>PF-IT3_GE_0043</i>	V.B. 
Nombre:	Filtrado de revisiones del alumno por periodo.	
Descripción:	Se crea y se realizan el “Examen9” con un periodo de revisión actual y el “Examen10” con un periodo de revisión posterior al actual.	
Resultado Esperado:	Al visualizar las revisiones disponibles debe aparecer únicamente el “Examen9”.	
Incidencia:	[No-Aplica]	

Figura 120. PF-IT3_GE_0043


ID:	<i>PF-IT3_GE_0044</i>	V.B. 
Nombre:	Revisión Alumno.	
Descripción:	Se crea y se realizan el “Examen11” con un periodo de revisión actual. Al finalizar el examen se añaden comentarios en la pantalla resumen.	
Resultado Esperado:	Al entrar en la revisión debe visualizarse las respuestas realizadas, las respuestas correctas, los datos de realización del examen, el gráfico de respuestas así como el comentario añadido al finalizar el examen.	
Incidencia:	[No-Aplica]	

Figura 121. PF-IT3_GE_0044


ID:	<i>PF-IT3_GE_0045</i>	V.B. 
Nombre:	Solicitud de revisión.	
Descripción:	Se crea y se realizan el “Examen13”. Al realizar la revisión se comentan las distintas preguntas y se pulsa el botón de Solicitar Revisión.	
Resultado Esperado:	El profesor que imparte la “Asignatura1”, debe visualizar en el listado de revisiones la realizada por el “Alumno1”.	
Incidencia:	[No-Aplica]	

Figura 122. PF-IT3_GE_0045


ID:	<i>PF-IT3_GE_0046</i>	V.B. 
Nombre:	Comentarios profesor	
Descripción:	El profesor entra en la revisión solicitada por el “Alumno1” y responde a los comentarios.	
Resultado Esperado:	El profesor debe visualizar las respuestas del alumno, las correctas, el gráfico de respuestas así como los comentarios realizados por el alumno y los suyos propios una vez realizados.	
Incidencia:	[No-Aplica]	

Figura 123. PF-IT3_GE_0046


ID:	<i>PF-IT3_GE_0047</i>	V.B. 
Nombre:	Finalización de Revisión.	
Descripción:	El profesor pulsa sobre el botón de finalización de revisión una vez añadidos los comentarios.	
Resultado Esperado:	En el listado de revisiones del alumno, debe modificarse el icono de un reloj de arena que aparecía al solicitar la revisión y en su lugar debe aparecer un icono de información. Así mismo debe activarse de nuevo el botón de solicitud de revisión desactivado durante el periodo de espera de las contestaciones del profesor.	
Incidencia:	[No-Aplica]	

Figura 124. PF-IT3_GE_0047

Seguimiento Planificación

Si bien es cierto que tras analizar la planificación resultante de la iteración dos, pueda parecer que no se ha producido ninguna alteración sobre la misma, salvo un pequeño retraso de cuatro días en la entrega; la realidad, es que los análisis realizados en el transcurso de la presente iteración, sobre el esfuerzo necesario para alcanzar el día propuesto como objetivo de presentación, han desencadenado una alteración sustancial del plan establecido.

Las primeras conclusiones de las estimaciones de esfuerzo dejaron constancia de la inviabilidad de alcanzar la fecha establecida. Las causas que arrojaba el análisis pueden resumirse en:

- Incorrecta estimación de la iteración 5, ya que en un principio fue planificada como una implementación básica de una pantalla de login, y posteriormente se ha enfocado como la implementación de un subsistema de seguridad, que afecta al desarrollo realizado, además de ser implementado y diseñado con una tecnología (Spring Security 2.0) desconocida. Así mismo, la iteración 5 debe contemplar el diseño e implementación del portal-aplicación que dé cabida a la funcionalidad desarrollada en las anteriores iteraciones.
- Falta de estimación del periodo de pruebas finales integradas y mantenimiento del sistema. Periodo que debe ser empleado para depurar y corregir errores que puedan aparecer después del proceso de implantación.
- Falta de estimación de los trabajos de generación, corrección y depuración de la documentación final del proyecto así como de la preparación de la presentación del mismo.

Ante estos problemas, una vez analizas las diversas soluciones posibles, se propone la modificación de la planificación siguiendo las siguientes pautas:

- Agrupación de las iteraciones tres y cuatro dejando fuera del alcance del sistema los submódulos de generación de estadísticas ya que no forman parte del núcleo descrito en el análisis.
- Realización de un esfuerzo extra en el desarrollo de la iteración tres, para poder ser entregada antes del día 24 de agosto de 2009, fecha que permitirá afrontar el resto de proyecto con garantías de éxito.
- Modificar la estimación de la iteración cuatro (antigua iteración cinco) dándole un tiempo real de desarrollo y fijando su fecha de entrega para el 21 de septiembre de 2009, teniendo en cuenta que dicho periodo contempla la elaboración de la memoria del proyecto cuya entrega se realizaría el día 18 de septiembre.

Siguiendo estos puntos, y fijando el día 28 de septiembre como cierre de proyecto, se garantizaría un mínimo de siete días como periodo de pruebas y

mantenimiento; además de contemplar el desarrollo de la memoria como parte de los trabajos de la iteración cuatro y dando un margen de un mínimo de cuatro días, para la preparación de la presentación del proyecto.

Con las modificaciones establecidas el cronograma de planificación sería el siguiente:

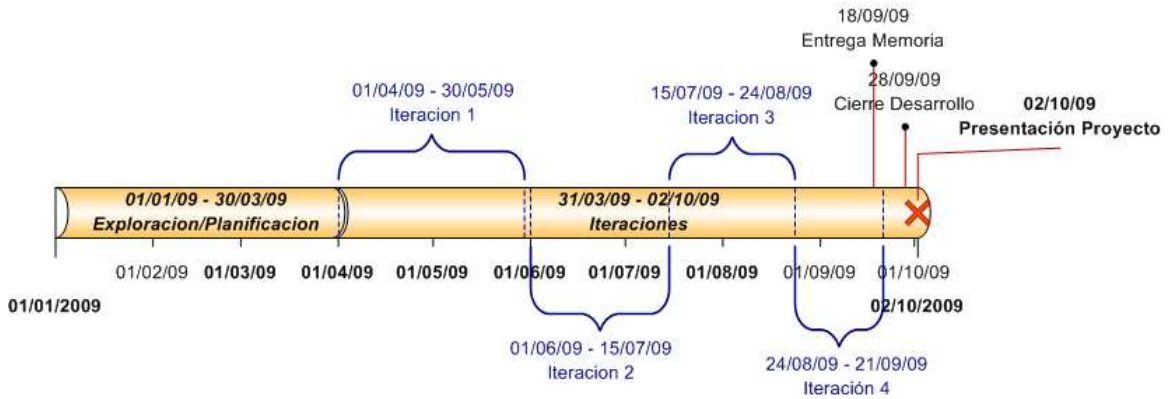


Figura 125. Cronograma con la actual planificación del proyecto

Validación del Documento

Próximos Hitos e Incidencias

ID	Tema	Estado	Responsable	Fecha Objetivo	Fecha de Impacto
9	Error PF-IT2_MC_0030,0031	Pendiente	Diego Jiménez	10/08/09	28/09/09
10	Error PF-IT2_MC_00028	Pendiente	Diego Jiménez	11/09/09	28/09/09
11	Desarrollo iteración 4	Pendiente	Borja Blanco	21/09/09	28/09/09
12	Entrega memoria proyecto	Pendiente	Borja Blanco	18/09/09	18/09/09
13	Cierre del desarrollo	Pendiente	Borja Blanco	28/09/09	28/09/09
14	Presentación proyecto	Pendiente	Borja Blanco	02/10/09	02/10/09

Hitos e Incidencias Cerrados

ID	Tema	Resolución	Responsable	Fecha Objetivo	Fecha Cierre
1	Análisis y Planificación	Finalizado	Borja Blanco	01/04/09	01/04/09
2	Arquitectura de Sistema	Finalizado	Borja Blanco	10/04/09	09/05/09
4	Desarrollo iteración 1	Finalizado	Borja Blanco	01/05/09	30/05/09
3	Definición servicios SOLE	Finalizado	Diego Jiménez	15/04/09	01/07/09
5	Desarrollo iteración 2	Finalizado	Borja Blanco	15/07/09	15/07/09
6	Error PN-IT1_BD_00017	Finalizado	Borja Blanco	15/07/09	03/06/09
7	Desarrollo iteración 3	Finalizado	Borja Blanco	15/08/09	18/08/09
8	Error PF-IT2_CE_0020,0021,0034	Finalizado	Borja Blanco	--/--/--	18/08/09

Aprobación del Documento

Rol	Nombre	Firma	Fecha
Autor del Proyecto	Borja Blanco Iglesias		
Tutor del Proyecto	Ricardo Colomo Palacios		

Documento Técnico de la Iteración 4



**Sistema de Gestión de
Exámenes Online
GEO**

**Informe Técnico sobre
Iteración 4
Madrid, 02 de Septiembre de 2009**

Autor	Borja Blanco Iglesias
Fecha de Creación	02/09/2009
Fecha de actualización	02/09/2009
Nombre de documento	IT4_GEO.doc
Versión	1.0

Histórico de Versiones

Fecha	Autor	Versión	Comentarios
02/09/2009	Borja Blanco	1.0	Versión inicial del documento.

Revisores

Nombre	Posición
Borja Blanco Iglesias	Autor del Proyecto
Ricardo Colomo Palacios	Tutor del Proyecto

Lista de distribución

Copia N°	Nombre	Posición
1.	Borja Blanco Iglesias	Autor del Proyecto
2.	Ricardo Colomo Palacios	Tutor del Proyecto
3.	Diego Jiménez López	Colaborador
4.		
5.		

Introducción

El Informe técnico de una iteración del desarrollo de un sistema bajo la metodología XP, ofrece una visión global sobre el análisis realizado así como las decisiones de diseño e implementación que se han tomado a lo largo del presente desarrollo.

Objetivos del Documento

El documento tiene como objetivo principal ofrecer una visión global sobre el estado actual del proyecto, enfocado hacia las tareas realizadas sobre la iteración cuatro en todo el periodo de desarrollo.

A lo largo del documento se detallarán aquellos aspectos funcionales que contempla el desarrollo realizado como consecuencia de la iteración, así como las diversas decisiones de diseño tomadas a lo largo del desarrollo. Además, se incluirán los resultados obtenidos del proceso de pruebas realizado.

Por último, cabe resaltar la inclusión en el documento, de una revisión de la planificación realizada en función a los tiempos de desarrollo obtenidos, estableciendo las comparaciones oportunas con los tiempos planificados.

Audiencia

Debido al carácter técnico, el presente documento está dirigido a aquellas personas que tomen parte en tareas de desarrollo del proyecto y necesiten acceso a la documentación sobre las tareas realizadas en la iteración cuatro. Así mismo, por la información general tratada en el mismo, la audiencia del documento se extiende a la dirección del proyecto quien deberá validar los requisitos definidos como base del desarrollo.

Definiciones, acrónimos, abreviaturas y referencias

A continuación se incluye información relativa a la terminología utilizada en el documento así como a las referencias realizadas en el mismo.

- **EJB:** Acrónimo del inglés *Enterprise Java Bean* es una API dentro del estándar JEE para la comunicación remota de servicios.
- **MDA:** Acrónimo del inglés *Model driven architecture* cuyo significado es arquitectura dirigida por el modelo.
- **XP (Extreme Programming):** Metodología de desarrollo ágil cuya traducción al español es Programación Extrema.

Definición del alcance del desarrollo

Dentro del ámbito del proyecto la presente iteración contempla el desarrollo de los siguientes módulos:

Control de Seguridad:

Todas las funciones desarrolladas en las iteraciones anteriores deben estar controladas por un módulo de seguridad capaz de restringir el acceso a determinados datos y operaciones a todo aquel usuario que carezca de los permisos necesarios. Como ya se definió con anterioridad, el sistema debe contemplar cuatro roles: profesor, alumno, administrador y administrador con permisos de superusuario.

Toda petición que se haga al sistema, debe estar controlada por el módulo de seguridad que se encargará de asegurar que el usuario que ha realizado la petición pertenezca a uno de los roles asignados al tipo de petición realizada.

Por último, cabe resaltar que el sistema no contempla usuarios anónimos, por lo que ningún usuario no logueado debe acceder a ninguna función del sistema.

Portal gráfico de usuario:

Como última fase del desarrollo del sistema, es imprescindible el diseño e implementación de un componente gráfico que permita mostrar a cada usuario las funciones que los roles asignados le permiten hacer dentro del sistema. Dicho componente gráfico mostrará el acceso a dichas funciones asignadas al rol del usuario. En el caso de que el usuario tenga asignados varios roles, debe mostrarse el acceso a cada uno de estos roles.

Diseño del Sistema

En la presente sección se muestran los resultados obtenidos del proceso de diseño dentro de la iteración tres. Estos resultados se engloban en dos grandes áreas, el modelo de datos que formaría el *back-end* de la solución propuesta, así como el diseño de las interfaces de usuario desarrolladas, lo que formaría parte del *front-end*.

Modelo de datos

El desarrollo del módulo de seguridad es implementado con el framework de desarrollo “Spring Security 2.0”. Dicho módulo es configurado para acceder a un proveedor JDBC cuyas tablas quedan integradas dentro del modelo del sistema.

Dicho modelo puede verse a través del siguiente diagrama de clases. En él, puede observarse el modelo utilizado directamente en la persistencia de los datos de la iteración, así como los distintos *Value Objects* utilizados en el mapeo de datos entre Beans persistentes e interfaz gráfica.

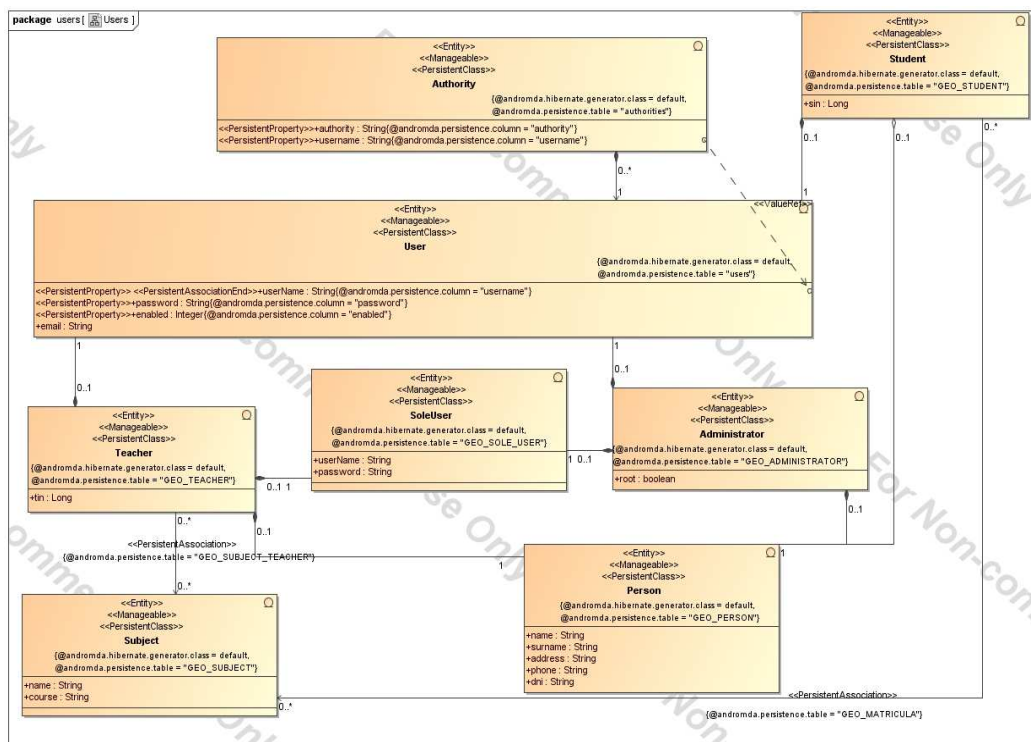


Figura 126. Modelo de negocio persistente relacionado con la el módulo de seguridad

El resultado de la utilización de la herramienta de generación de código AndroMDA, sobre el presente modelo de clases estereotipadas, son una serie de servicios implementados sobre EJB's, los cuales dan acceso a los datos de persistencia, encapsulados en una serie de librerías, que son importadas dentro de la aplicación desarrollada.

A continuación, se hará una exposición detallada de las clases que componen el modelo de negocio utilizado en la presente iteración.

Authority – Entidad de autorización

org.babieca.geo.ddbb.users.Authority	
Nombre de Clase:	<ul style="list-style-type: none"> • Authority
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.users
Estereotipos:	<ul style="list-style-type: none"> • Entity • Manageable • PersistentClass
Atributos:	<ul style="list-style-type: none"> • authority (String) – Nombre del rol. • username (String) – Nombre de usuario.
Métodos:	-----
Relaciones:	User

Tabla 53. Clase Authority

User – Entidad Usuario

org.babieca.geo.ddbb.users.User	
Nombre de Clase:	<ul style="list-style-type: none"> • User
Paquete:	<ul style="list-style-type: none"> • org.babieca.geo.ddbb.users
Estereotipos:	<ul style="list-style-type: none"> • Entity • Manageable • PersistentClass
Atributos:	<ul style="list-style-type: none"> • userName(String) – Nombre de usuario • password (String) – Clave del usuario • enabled (int) - Flag que marca si un usuario está activo • email - (String) – Email del usuario
Métodos:	-----
Relaciones:	<ul style="list-style-type: none"> • Administrator • Student • Teacher

Tabla 54. Clase User

Interfaz Gráfica de Usuario

Completando al modelo de negocio en el ámbito del diseño, se presentan las distintas Interfaces de usuario implementadas en la presente iteración.

Login

El modulo de seguridad del sistema se basa en la autenticación del usuario a través de su nombre de usuario y contraseña. A continuación, se muestra la pantalla a través de la cual un usuario anónimo podrá loguearse en el sistema.

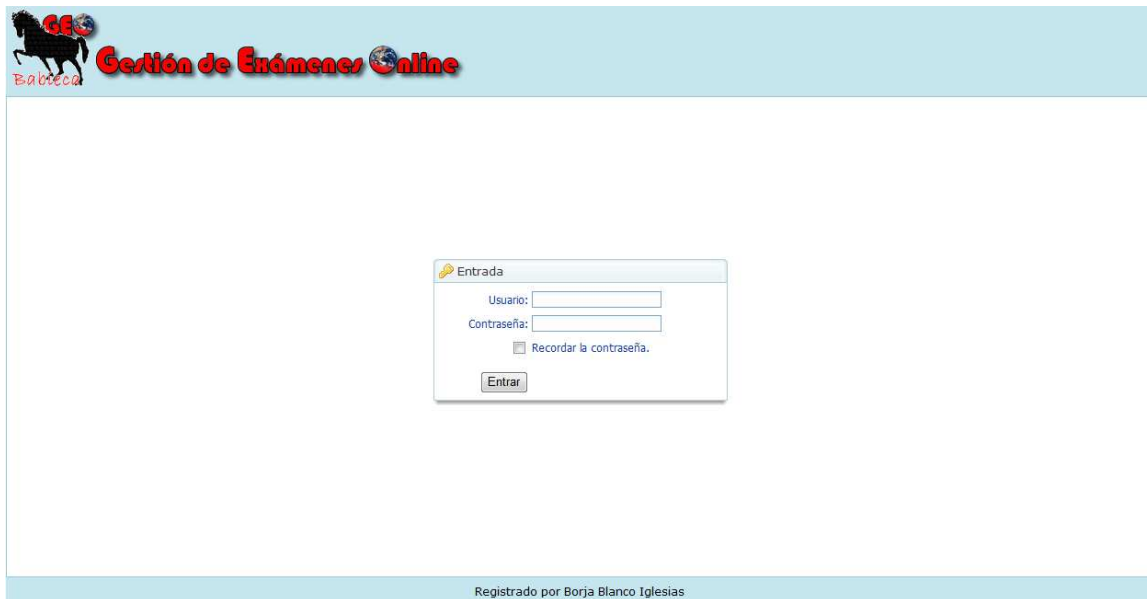


Figura 127. Pantalla de login

Una vez el usuario se haya autenticado correctamente, su nombre de usuario permanecerá almacenado en la sesión para las distintas consultas. Si se intenta acceder a cualquier función sin estar logueado, esta interfaz aparecerá automáticamente.

Portal Administrador

Al igual que ocurre con cualquier otro usuario, un administrador tiene la opción de cambiar sus datos de usuario, a excepción de los permisos asignados, función únicamente disponible como “superusuario”. Así mismo, todo administrador podrá acceder a la configuración del sistema, en donde podrá modificar la URL del servidor de consulta SOLE, así como gestionar todos aquellos datos administrables del sistema. Estos datos son: usuarios del sistema, incluyendo alumnos y profesores y asignaturas.

A continuación, puede observarse la interfaz diseñada para dar acceso a estas funciones, en la cual se ha incluido un menú en la parte izquierda agrupando dichos accesos por funcionalidad.



Figura 128. Portal del administrador

Portal Administrador superusuario

El usuario registrado con el rol especial de administración de “superusuario” podrá acceder a todas aquellas funciones propias de un administrador descritas en la sección anterior, además de la gestión de los administradores, tal como puede observarse en la siguiente figura.

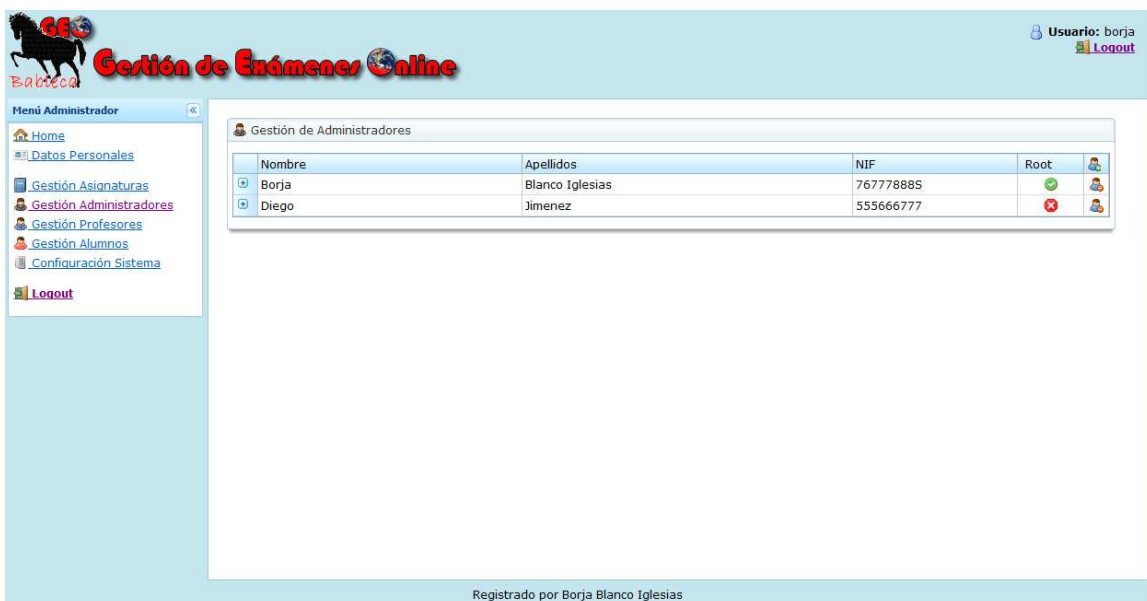


Figura 129. Portal del Administrador Superusuario

Portal Profesor

El profesor tiene dos funcionalidades propias como son la creación de exámenes y la revisión de los mismos. Ambas son accesibles desde el menú izquierdo de la pantalla. Tal como puede verse en la siguiente imagen.

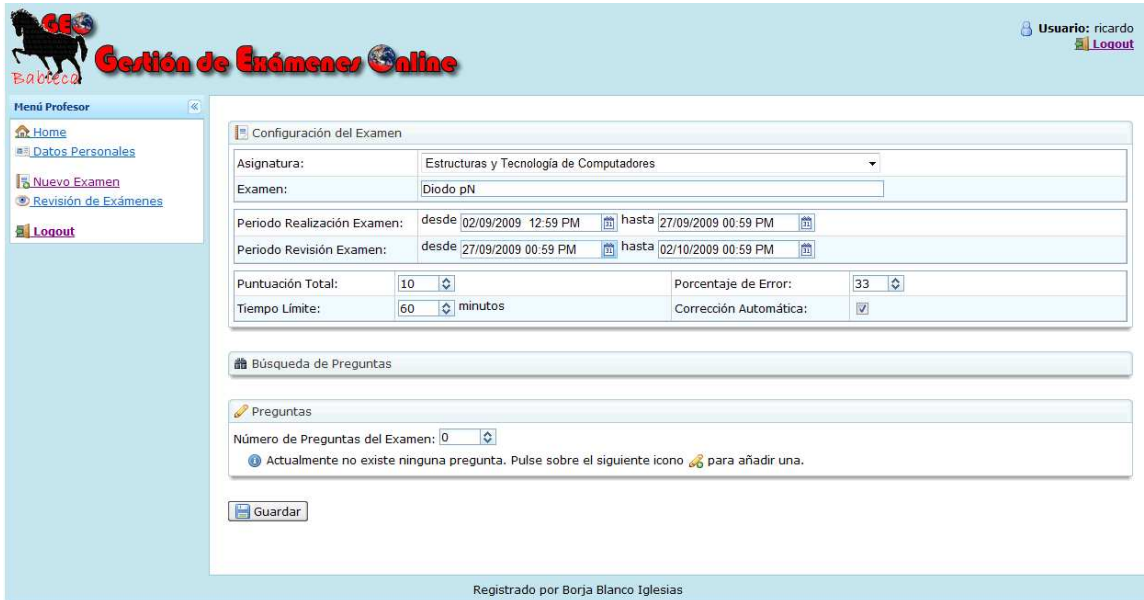


Figura 130. Portal del profesor

Portal Alumno

Del mismo modo que en el resto de usuarios, el alumno dispone de un menú, en la parte izquierda de la interfaz, desde el que acceder a sus funciones personales, que en este caso son la realización y la revisión de los exámenes.



Figura 131. Portal del alumno

Pruebas

En la presente sección se expondrán los resultados obtenidos de la ejecución del plan de pruebas tanto funcionales, sobre el desarrollo realizado dentro de la segunda iteración.

Pruebas Funcionales

Las pruebas funcionales tienen como objetivo principal, la verificación de la adecuación de la solución a las especificaciones expuestas en el análisis, comprobando que el sistema funciona tal como se describe en los documentos de análisis.

El Identificador de las pruebas se compone de cuatro partes:

- Los caracteres PF seguidos de un guión indican que se trata de una prueba funcional.
- Los caracteres IT4 indican que se trata una prueba de la cuarta iteración del desarrollo.
- Dos caracteres indicando el módulo de pruebas: seguridad y portal. (SE y PO respectivamente).
- Cuatro dígitos que indican unívocamente el número de prueba realizada.

Cabe resaltar que para una correcta ejecución del plan de pruebas, éste debe ser ejecutado siguiendo un riguroso orden de ejecución tal como marca el identificador dado a cada prueba y con una base de datos previamente vacía a excepción de un usuario con nombre “root” con los tres roles, un administrador cuyo nombre de usuario sea “administrador”, un profesor de nombre de usuario “profesor” y un alumno de nombre de usuario “alumno”.

A continuación se exponen los resultados obtenidos en las distintas pruebas:


ID:	PF-IT4_SE_0048	V.B. 
Nombre:	Autenticación de usuario.	
Descripción:	Desde la pantalla de entrada, autenticarse con el usuario “root”.	
Resultado Esperado:	En el frame superior debe aparecer el usuario “root” así como el acceso al “logout”.	
Incidencia:	[No-Aplica]	

Figura 132. PF-IT4_SE_0048

ID:	<i>PF-IT4_PO_0049</i>	V.B. ✓
Nombre:	Redirección portal administración.	
Descripción:	Hacer “logout” y autenticarse con el usuario “ <i>administrador</i> ”.	
Resultado Esperado:	Debe aparecer la pantalla principal de administración.	
Incidencia:	[No-Aplica]	

Figura 133. PF-IT4_PO_0049

ID:	<i>PF-IT4_PO_0050</i>	V.B. ✓
Nombre:	Acceso funcionalidad administración.	
Descripción:	Probar cada uno de los accesos a las funcionalidades de administración.	
Resultado Esperado:	Se debe permitir el acceso con total normalidad a cada una de las funcionalidades.	
Incidencia:	[No-Aplica]	

Figura 134. PF-IT4_PO_0050

ID:	<i>PF-IT4_SE_0051</i>	V.B. ✓
Nombre:	Bloqueo rol profesor.	
Descripción:	Acceder a la URL “ <i>/web/teacher</i> ”.	
Resultado Esperado:	Se debe bloquear e impedir el acceso a ese recurso.	
Incidencia:	[No-Aplica]	

Figura 135. PF-IT4_SE_0051

ID:	<i>PF-IT4_SE_0052</i>	V.B. ✓
Nombre:	Bloqueo rol alumno.	
Descripción:	Acceder a la URL “ <i>/web/student</i> ”.	
Resultado Esperado:	Se debe bloquear e impedir el acceso a ese recurso.	
Incidencia:	[No-Aplica]	

Figura 136. PF-IT4_SE_0052

ID:	<i>PF-IT4_PO_0053</i>	V.B. ✓
Nombre:	Redirección portal profesor.	
Descripción:	Hacer “logout” y autenticarse con el usuario “ <i>profesor</i> ”.	
Resultado Esperado:	Debe aparecer la pantalla principal del portal del profesor.	
Incidencia:	[No-Aplica]	

Figura 137. PF-IT4_PO_0053


ID:	<i>PF-IT4_PO_0054</i>	V.B. 
Nombre:	Acceso funcionalidad profesor.	
Descripción:	Probar cada uno de los accesos a las funcionalidades del profesor.	
Resultado Esperado:	Se debe permitir el acceso con total normalidad a cada una de las funcionalidades.	
Incidencia:	[No-Aplica]	

Figura 138. PF-IT4_PO_0054


ID:	<i>PF-IT4_SE_0055</i>	V.B. 
Nombre:	Bloqueo rol administración.	
Descripción:	Acceder a la URL “/web/admin”.	
Resultado Esperado:	Se debe bloquear e impedir el acceso a ese recurso.	
Incidencia:	[No-Aplica]	

Figura 139. PF-IT4_SE_0055


ID:	<i>PF-IT4_PO_0056</i>	V.B. 
Nombre:	Redirección portal alumno.	
Descripción:	Hacer “logout” y autenticarse con el usuario “alumno”.	
Resultado Esperado:	Debe aparecer la pantalla principal del portal del alumno.	
Incidencia:	[No-Aplica]	

Figura 140. PF-IT4_PO_0056


ID:	<i>PF-IT4_PO_0057</i>	V.B. 
Nombre:	Acceso funcionalidad alumno.	
Descripción:	Probar cada uno de los accesos a las funcionalidades del alumno.	
Resultado Esperado:	Se debe permitir el acceso con total normalidad a cada una de las funcionalidades.	
Incidencia:	[No-Aplica]	

Figura 141. PF-IT4_PO_0057

Seguimiento Planificación

La presente iteración ha sido finalizada con un adelanto de 17 días respecto a la planificación establecida, sin embargo, dada la carga de trabajo que supondrá la consecución de los hitos restantes, no se ha creído oportuno la modificación de ninguno de ellos. Por esto se mantiene la planificación tal como marca el siguiente cronograma:

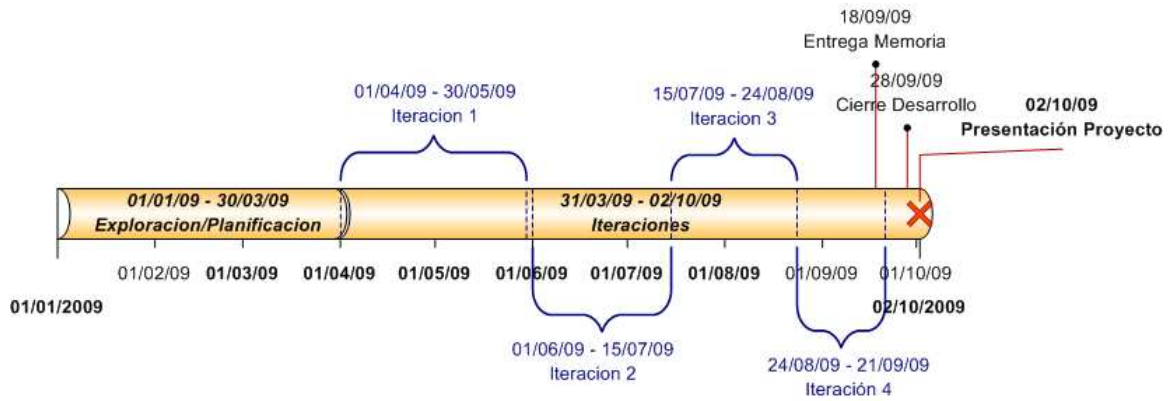


Figura 142. Cronograma con la actual planificación del proyecto

Validación del Documento

Próximos Hitos e Incidencias

ID	Tema	Estado	Responsable	Fecha Objetivo	Fecha de Impacto
9	Error PF-IT2_MC_0030,0031	Pendiente	Diego Jiménez	10/08/09	28/09/09
10	Error PF-IT2_MC_00028	Pendiente	Diego Jiménez	11/09/09	28/09/09
12	Entrega memoria proyecto	Pendiente	Borja Blanco	18/09/09	18/09/09
13	Cierre del desarrollo	Pendiente	Borja Blanco	28/09/09	28/09/09
14	Presentación proyecto	Pendiente	Borja Blanco	02/10/09	02/10/09

Hitos e Incidencias Cerrados

ID	Tema	Resolución	Responsable	Fecha Objetivo	Fecha Cierre
1	Análisis y Planificación	Finalizado	Borja Blanco	01/04/09	01/04/09
2	Arquitectura de Sistema	Finalizado	Borja Blanco	10/04/09	09/05/09
4	Desarrollo iteración 1	Finalizado	Borja Blanco	01/05/09	30/05/09
3	Definición servicios SOLE	Finalizado	Diego Jiménez	15/04/09	01/07/09
5	Desarrollo iteración 2	Finalizado	Borja Blanco	15/07/09	15/07/09
6	Error PN-IT1_BD_00017	Finalizado	Borja Blanco	15/07/09	03/06/09
7	Desarrollo iteración 3	Finalizado	Borja Blanco	15/08/09	18/08/09
8	Error PF-IT2_CE_0020,0021,0034	Finalizado	Borja Blanco	--/--/--	18/08/09
11	Desarrollo iteración 4	Pendiente	Borja Blanco	21/09/09	02/09/09

Aprobación del Documento

Rol	Nombre	Firma	Fecha
Autor del Proyecto	Borja Blanco Iglesias		
Tutor del Proyecto	Ricardo Colomo Palacios		

Estimación de Costes



Sistema de Gestión de Exámenes Online GEO

Estimación de Costes
Madrid, 20 de Marzo de 2009

Autor	Borja Blanco Iglesias
Fecha de Creación	20/03/2009
Fecha de actualización	20/03/2009
Nombre de documento	Presupuesto_GEO.doc
Versión	1.0

Histórico de Versiones

Fecha	Autor	Versión	Comentarios
20/03/2009	Borja Blanco	1.0	Versión inicial del documento.

Revisores

Nombre	Posición
Borja Blanco Iglesias	Autor del Proyecto
Ricardo Colomo Palacios	Tutor del Proyecto

Lista de distribución

Copia N°	Nombre	Posición
1.	Borja Blanco Iglesias	Autor del Proyecto
2.	Ricardo Colomo Palacios	Tutor del Proyecto

Introducción

El documento de estimación de costes permite dar una visión global de los gastos que se deberán hacer frente durante el periodo de desarrollo del sistema.

Objetivos del Documento

El objetivo de este documento es registrar los diferentes componentes del coste de mano de obra, junto con el resto de costes que incorpora el desarrollo del software.

Audiencia

El documento va dirigido a la dirección de proyecto quien debe aprobarlo para establecer la viabilidad del mismo y por lo tanto del propio proyecto.

Gastos de personal

A continuación, se encuentran el cálculo de costes relacionado con los gastos derivados del personal asignado al proyecto.

Se ha establecido que el proyecto tendrá una duración de 31 semanas, con una dedicación de 15 horas semanales de un único recurso y unos honorarios para el desarrollador de 30 € por hora trabajada. Según estos datos, los gastos de personal vienen dados por la siguiente tabla:

Actividad	Semanas Estimadas	Coste Actividad (€)
Análisis	7	3.150
Diseño Arquitectónico	7	3.150
Iteración 1	4	1.800
Iteración 2	6	2.700
Iteración 3	4	1.800
Iteración 4	2	900
Iteración 5	1	450
Total costes	31	13.950

Tabla 55. Relación de costes por actividad

Gastos de equipo informático

Los equipos utilizados en el proyecto tienen estimado un periodo de amortización lineal de 5 años. A continuación, se expone una tabla con los precios de compra así como la amortización durante el periodo de duración del proyecto.

Equipos informáticos	Precio de compra (€)	Amortización semanal (€)	Amortización del proyecto
Servidor Desarrollo (1)	600	2,31	71,3
Ordenadores personales (1)	1.500	5,76	178,84
Impresoras (1)	120	0,46	14,3
Total costes			264,44

Tabla 56. Gastos de equipos

Gastos de material fungible

En el siguiente cuadro, calculamos los costes derivados del material fungible que se utilizará a lo largo del desarrollo del proyecto.

Material	Precio unitario	Unidades	Coste total
Cartuchos de tinta	12	4	48,00
Paquetes de folios (500 uds.)	3	2	6,00
Material de oficina	-	-	50,00
Total costes			104,00

Tabla 57. Gastos de material fungible

Otros gastos

Otros gastos derivados del desarrollo del proyecto son los siguientes:

Gastos	Coste mensual (€)	Costes total (7 meses)
Hosting	15	105,00
Dominio	-	9,00
Suministros (agua, luz, teléfono, adsl, gas)	130,00	910,00
Viajes y dietas	-	100,00
Otros gastos	-	100,00
Total costes	145,00	1.224,00

Tabla 58. Otros gastos

Resumen del presupuesto

Por último, presentamos el resumen del presupuesto, para lo cual hemos ido calculando todos los gastos anteriores, con un beneficio del 20% y un índice de riesgo medio asignando un 15% al mismo.

	Estimaciones
1. Gastos de Personal	13.950,00
2. Equipos Informáticos	264,44
4. Material Fungible	104,00
6. Otros	1.224,0
Total	15.542,44
Beneficio (20%)	3.108,49
Riesgo (15%)	2.331,37
Subtotal	20.983,30
I.V.A. (16%)	3.357,733
Total presupuesto	24.340,62

Tabla 59. Resumen del presupuesto del proyecto

Validación del Documento

Aprobación del Documento

Rol	Nombre	Firma	Fecha
Autor del Proyecto	Borja Blanco Iglesias		
Tutor del Proyecto	Ricardo Colomo Palacios		

Análisis de Costes



Sistema de Gestión de Exámenes Online GEO

Análisis de Costes
Madrid, 16 de Septiembre de 2009

Autor	Borja Blanco Iglesias
Fecha de Creación	16/09/2009
Fecha de actualización	16/09/2009
Nombre de documento	Costes_GEO.doc
Versión	1.0

Histórico de Versiones

Fecha	Autor	Versión	Comentarios
16/09/2009	Borja Blanco	1.0	Versión inicial del documento.

Revisores

Nombre	Posición
Borja Blanco Iglesias	Autor del Proyecto
Ricardo Colomo Palacios	Tutor del Proyecto

Lista de distribución

Copia N°	Nombre	Posición
1.	Borja Blanco Iglesias	Autor del Proyecto
2.	Ricardo Colomo Palacios	Tutor del Proyecto

Introducción

El documento de análisis de costes permite dar una visión global de los gastos que se han hecho frente durante el periodo de desarrollo del sistema, así como compararlos con las estimaciones realizadas con anterioridad al desarrollo.

Objetivos del Documento

El objetivo de este documento es registrar los diferentes componentes del coste de mano de obra junto con el resto de costes que incorpora el desarrollo del software.

Audiencia

El documento va dirigido a la dirección de proyecto quien debe aprobarlo para establecer el beneficio obtenido con el desarrollo realizado.

Gastos de personal

A continuación, se encuentran el cálculo de costes relacionado con los gastos derivados del personal asignado al proyecto.

La duración del proyecto ha sido de 38 semanas, en lugar de las 31 estimadas previamente. Teniendo en cuenta que la dedicación del recurso asignado ha sido de 15 horas semanales con unos honorarios de 30 € por hora trabajada, nos encontramos con la siguiente tabla comparativa:

Actividades	Semanas Estimadas	Coste Actividad Estimado (€)	Semanas de Desarrollo	Coste Actividad Real (€)
Análisis	7	3.150	7	3.150
Diseño Arquitectónico	7	3.150	7	3.150
Iteración 1	4	1.800	8	3.600
Iteración 2	6	2.700	6	2.700
Iteración 3	4	1.800	6	2.700
Iteración 4	2	900	4	1800
Iteración 5	1	450	-	-
Total costes	31	13.950	38	17.100

Tabla 60. Relación de costes por actividad

Gastos de equipo informático

El material utilizado en el proyecto tiene estimado un periodo de amortización lineal de 5 años. A continuación, se expone una tabla con los precios de compra así como la amortización durante el periodo de duración del proyecto y su comparación con la estimación previa.

Equipos Informáticos	Precio de Compra (€)	Amortización Semanal (€)	Amortización Estimada	Amortización Real
Servidor Desarrollo (1)	600	2,31	71,3	87,78
Ordenadores personales (1)	1.500	5,76	178,84	218,88
Impresoras (1)	120	0,46	14,3	17,48
Total costes			264,44	324,14

Tabla 61. Gastos de equipos

Gastos de material fungible

En el siguiente cuadro, calculamos costes derivados del material fungible que se utilizará a lo largo del desarrollo del proyecto.

Material	Precio unitario	Unidades estimadas	Coste total estimado	Unidades consumidas	Coste total
Cartuchos de tinta	12	4	48,00	2	24,00
Paquetes de folios (500 uds.)	3	2	6,00	2	6,00
Material de oficina	-	-	50,00	-	0,00
Total costes			104,00		30,00

Tabla 62. Gastos de material fungible

Otros gastos

Otros gastos derivados del desarrollo del proyecto son los siguientes:

Gastos	Coste mensual (€)	Costes total estimado	Coste mensual (€)	Costes total estimado
Hosting	15	135,00	15	165,00
Dominio	-	9,00	-	9,00
Suministros (agua, luz, teléfono, adsl, gas)	130,00	910,00	130,00	1.170,00
Viajes y dietas	-	0,00	-	0,00
Otros gastos	-	95,96	-	95,96
Total costes	145,00	1.224,00	145,00	1.439,96

Tabla 63. Otros gastos

Resumen del presupuesto

Por último, presentamos el resumen del presupuesto, para lo cual hemos ido calculando todos los gastos anteriores, con un beneficio del 20% y un índice de riesgo medio asignando un 15% al mismo.

	Estimaciones	Gasto Total
1. Gastos de personal	13.950,00	17.100,00
2. Equipos	264,44	324,14
4. Fungible	104,00	30,00
6. Otros	1.224,0	1.439,96
Total	15.542,44	18.894,10
Beneficio	3.108,49 (20%)	2.089,20 (11%)
Riesgo (15%)	2.331,37	-
Subtotal	20.983,30	20.983,30
I.V.A. (16%)	3.357,733	3.357,733
Total presupuesto	24.340,62	24.340,62

Tabla 64. Resumen del presupuesto del proyecto

Validación del Documento

Aprobación del Documento

Rol	Nombre	Firma	Fecha
Autor del Proyecto	Borja Blanco Iglesias		
Tutor del Proyecto	Ricardo Colomo Palacios		