



**ESCUELA SUPERIOR DE INGENIERÍA**

**INGENIERÍA INFORMÁTICA**

**EvalCourse: un Lenguaje Específico de Dominio para obtener indicadores del desarrollo de competencias a través de un LMS**

Antonio Balderas Alberico

12 de julio de 2013





ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA EN INFORMÁTICA

**EvalCourse: un Lenguaje Específico de Dominio para obtener indicadores del desarrollo de competencias a través de un LMS**

- Departamento: Ingeniería Informática
- Directores del proyecto: Iván Ruíz Rube y Juan Manuel Doderó Beardo
- Autor del proyecto: Antonio Balderas Alberico

Cádiz, 12 de julio de 2013

Fdo: Iván Ruiz Rube | Fdo: Juan M. Doderó Beardo | Fdo: Antonio Balderas Alberico



## *Agradecimientos*

*A mi familia.*

## Resumen

Hoy en día es casi tan importante que un estudiante adquiera los conocimientos necesarios para ser un buen profesional en lo que estudia, como que adquiera las competencias necesarias, genéricas y específicas, que a un profesional la rama objeto de estudio se le deben suponer. Pero la evaluación de estas competencias no es una tarea sencilla, y muchas veces puede basarse en la simple observación que el docente hace sobre el trabajo del alumno en clase, llegando a ser una evaluación muy subjetiva. Con EvalCourse, se pretende proporcionar una herramienta que otorgue al docente unos criterios cuantificables para evaluar el desempeño de los estudiantes en base a dichas competencias.

EvalCourse se va a implementar como un Lenguaje de Dominio Específico (DSL). Un DSL es un lenguaje de programación orientado a un problema específico y con una semántica orientada al dominio para el que se diseña. En nuestro caso, este dominio es la evaluación de indicadores de competencias de los estudiantes. El objetivo de EvalCourse es ayudar al docente en la evaluación del desempeño de los alumnos en las competencias que éstos deben desarrollar a lo largo del curso. Para llevar a cabo esta función tomará los datos que necesite de un LMS (Learning Management System). Estos datos deberán servir al docente como indicadores del desarrollo de dichas competencias. El LMS para el que se ha creado este software inicialmente es Moodle, ya que es la plataforma virtual que se utiliza en la Universidad de Cádiz y con el que todos estamos más familiarizados. De todas formas, la idea inicial con la que se ha desarrollado este proyecto, es que el DSL sea genérico y pueda ser utilizado con otros sistemas, no sólo de tipo LMS, sino también con sistemas que fomenten el trabajo colaborativo, como pueden ser los wikis. Por ejemplo, MediaWiki es utilizado también en el ámbito de la docencia en nuestra Universidad.

Para su desarrollo utilizaremos Eclipse Modeling Framework (EMF). EMF es el núcleo de la herramienta Eclipse para el desarrollo dirigido por modelos. El metamodelo de EvalCourse será desarrollado con Ecore. Y para especificar un lenguaje, la gramática será descrita utilizando el lenguaje Xtext. El producto final se instala como un plugin adicional en la herramienta Eclipse que deberá tener el docente o usuario del programa. Para su utilización éste deberá crear, utilizando nuestro DSL, una sentencia indicando la información que se desea obtener. Como resultado se obtiene un fichero para el programa Kettle de Pentaho, un software que sirve para generar informes y para minería de datos. Desde EvalCourse a través de este fichero se podrán obtener tanto los datos de la evaluación de indicadores de competencias en un formato textual, como en formato gráfico.

Para obtener información de Moodle, el software extraerá de la base de datos los datos de la participación, entrega de tareas o acceso de los alumnos a Moodle. Estos datos podrán ser utilizados por el docente como indicadores de las competencias. En los próximos capítulos explicaremos la sintaxis del DSL, sus aspectos técnicos y toda la información relativa al desarrollo e implementación de la herramienta.

**Palabras clave:** EvalCourse, DSL, Lenguajes Específicos de Dominio, indicadores, evidencias, evaluación, competencias, LMS, Moodle.

# Índice general

<b>I Prolegómeno</b>	<b>1</b>
<b>1. Introducción</b>	<b>5</b>
1.1. Motivación . . . . .	5
1.2. Descripción del sistema actual . . . . .	6
1.3. Descripción del problema . . . . .	6
1.3.1. Diagnóstico del problema . . . . .	7
1.3.2. Propuesta de estrategia . . . . .	7
1.3.3. Beneficios esperados . . . . .	8
1.4. Objetivos y alcance del proyecto . . . . .	8
1.5. Organización del documento . . . . .	9
<b>2. Planificación</b>	<b>11</b>
2.1. Metodología de desarrollo . . . . .	11
2.1.1. Primer ciclo . . . . .	11
2.1.2. Segundo ciclo . . . . .	11
2.1.3. Tercer ciclo . . . . .	12
2.1.4. Cuarto ciclo . . . . .	12
2.1.5. Quinto ciclo . . . . .	12
2.2. Planificación del proyecto . . . . .	13
2.3. Costes . . . . .	19
<b>3. Ingeniería Dirigida por Modelos</b>	<b>21</b>
3.1. MDE: Ingeniería Dirigida por Modelos . . . . .	21
3.2. DSL: Lenguaje de Dominio Específico . . . . .	21
3.2.1. Fases en el desarrollo de un DSL externo . . . . .	22
3.3. MDE y DSL . . . . .	23
<b>II Desarrollo</b>	<b>25</b>
<b>4. Análisis de Requisitos</b>	<b>29</b>
4.1. Catálogo de actores . . . . .	29
4.2. Requisitos funcionales . . . . .	29
4.2.1. Caso de uso general: interacción . . . . .	31
4.2.2. Caso de uso: consultar evidencias en actividades . . . . .	31
4.2.3. Caso de uso: consultar evidencias en foros . . . . .	32
4.2.4. Caso de uso: consultar evidencias en workshops . . . . .	32

4.2.5. Caso de uso: evidencias de campus . . . . .	33
4.3. Requisitos de información . . . . .	33
4.3.1. Moodle . . . . .	34
4.3.2. Metamodelo . . . . .	41
4.4. Requisitos no funcionales . . . . .	41
4.5. Estudio de alternativas tecnológicas . . . . .	43
<b>5. Diseño del Sistema</b>	<b>45</b>
5.1. Diseño de la arquitectura . . . . .	45
5.2. Arquitectura física . . . . .	45
5.3. Arquitectura lógica . . . . .	45
5.3.1. EMF: Eclipse Modeling Framework . . . . .	47
5.3.2. Ecore Tools . . . . .	48
5.3.3. Xtext . . . . .	49
5.3.4. Xtend . . . . .	50
5.3.5. Transformaciones a Pentaho . . . . .	50
5.4. Arquitectura de diseño . . . . .	57
<b>6. Implementación del Sistema</b>	<b>59</b>
6.1. Introducción . . . . .	59
6.2. Modelo . . . . .	59
6.3. Proyecto EMF . . . . .	59
6.4. DSL . . . . .	63
6.5. Traducción . . . . .	67
<b>7. Pruebas del Sistema</b>	<b>71</b>
7.1. Pruebas unitarias . . . . .	71
7.1.1. Metamodelo . . . . .	71
7.1.2. Editor . . . . .	71
7.1.3. Transformación . . . . .	71
7.1.4. Resultados en xml . . . . .	72
7.1.5. Resultados en gráfico . . . . .	72
7.2. Pruebas de integración . . . . .	73
7.3. Pruebas de sistema . . . . .	73
7.3.1. Pruebas funcionales . . . . .	73
7.4. Pruebas de aceptación . . . . .	77
7.4.1. Recopilación de información . . . . .	77
7.4.2. Análisis de resultados . . . . .	79
<b>III Epílogo</b>	<b>83</b>
<b>8. Manual de usuario</b>	<b>87</b>
8.1. Introducción . . . . .	87
8.2. Características . . . . .	87
8.3. Requisitos previos . . . . .	88
8.4. Utilización . . . . .	88

<b>9. Manual de instalación y explotación</b>	<b>93</b>
9.1. Introducción . . . . .	93
9.2. Requisitos previos . . . . .	93
9.3. Procedimientos de instalación . . . . .	94
<b>10. Conclusiones</b>	<b>97</b>
10.1. Objetivos . . . . .	97
10.2. Lecciones aprendidas . . . . .	98
10.2.1. Planificación . . . . .	98
10.2.2. Costes . . . . .	98
10.2.3. Experiencia . . . . .	102
10.3. Trabajo futuro . . . . .	102
<b>Bibliografía</b>	<b>105</b>
<b>IV Apéndice</b>	<b>107</b>
.1. Contenido de fichero xml de Pentaho (ejemplo) . . . . .	109
.2. Contenido del fichero EvalCourse.ecore . . . . .	123
.3. GNU General Public License . . . . .	125
<b>Artículo enviado a EC-TEL 2013</b>	<b>137</b>
<b>Artículo enviado a ISELEAR'13</b>	<b>139</b>



# Índice de figuras

2.1. Diagrama de Gantt . . . . .	14
2.2. Diagrama de Gantt - Parte 1 . . . . .	15
2.3. Diagrama de Gantt - Parte 2 . . . . .	16
2.4. Diagrama de Gantt - Parte 3 . . . . .	17
2.5. Resumen de fechas planificadas para el desarrollo del proyecto . . . . .	18
3.1. Gráfico de relaciones entre los modelos dirigidos . . . . .	23
4.1. Casos de uso Interacción . . . . .	30
4.2. Casos de uso Realizar Informes . . . . .	30
4.3. Herencia de casos de usos del súper-caso de interacción . . . . .	31
4.4. Herencia de casos de usos de consultar evidencias en actividades . . . . .	32
4.5. Herencia de casos de usos de consultar evidencia en foros . . . . .	33
4.6. Herencia de casos de usos de consultar evidencias en workshops . . . . .	34
4.7. Herencia de casos de usos de consultar evidencias en campus . . . . .	35
4.8. Diagrama de Moodle V.2 . . . . .	36
4.9. Diagrama del módulo de usuarios de Moodle V.2 . . . . .	37
4.10. Diagrama del módulo de roles de Moodle V.2 . . . . .	38
4.11. Diagrama del módulo de foros de Moodle V.2 . . . . .	39
4.12. Diagrama del módulo de actividades de Moodle V.2 . . . . .	39
4.13. Diagrama del módulo de workshop de Moodle V.2 . . . . .	40
4.14. Diagrama del módulo de logging de Moodle V.2 . . . . .	40
4.15. Metamodelo de EvalCourse . . . . .	42
4.16. Filtros para la obtención de informes en Moodle . . . . .	43
5.1. Arquitectura software. Proceso de desarrollo . . . . .	46
5.2. Diagrama de la arquitectura física . . . . .	46
5.3. Diagrama de la arquitectura lógica . . . . .	47
5.4. Diagrama de componentes de la arquitectura lógica de EvalCourse . . . . .	48
5.5. Vista del editor gráfico de Ecore . . . . .	48
5.6. Jerarquía de componentes Ecore . . . . .	49
5.7. Diagrama de la arquitectura de diseño . . . . .	58
6.1. Creación proyecto EMF . . . . .	60
6.2. Creación diagrama en Ecore . . . . .	60
6.3. Vista gráfica del metamodelo . . . . .	61
6.4. Archivo del metamodelo Ecore . . . . .	61
6.5. Creación fichero de generación de modelo . . . . .	62

6.6.	Generador del metamodelo . . . . .	63
6.7.	Proyecto del metamodelo . . . . .	64
6.8.	Creación de proyecto Xtext basado en Ecore . . . . .	64
6.9.	Generación infraestructura DSL . . . . .	66
6.10.	Generación infraestructura DSL . . . . .	67
6.11.	Transformación en Pentaho . . . . .	67
6.12.	Componentes de la transformación en Pentaho . . . . .	68
6.13.	Clases que conforman el paquete utils . . . . .	69
6.14.	Creación del plugin del DSL . . . . .	69
7.1.	Validación de un modelo conforme al metamodelo de EvalCourse. . . . .	71
7.2.	Gráfico del desempeño de los estudiantes en la entrega de actividades. . . . .	75
7.3.	Gráfico del desempeño de los estudiantes en la participación en el foro. . . . .	77
7.4.	Gráfico del desempeño de los estudiantes en el trabajo en el foro. . . . .	81
8.1.	Creación de un nuevo proyecto de EvalCourse . . . . .	89
8.2.	Dando nombre a un nuevo proyecto de EvalCourse . . . . .	89
8.3.	Estructura de ficheros que se crea con un nuevo proyecto de EvalCourse . . . . .	90
8.4.	Cosulta escrita en fichero process.evc . . . . .	90
8.5.	Ficheros creados en carpeta src-gen . . . . .	91
8.6.	Ficheros de salida . . . . .	91
8.7.	Resultado gráfico en software Gephi . . . . .	92
8.8.	Resultado textual en xml - lado izquierdo . . . . .	92
8.9.	Resultado textual en xml - lado derecho . . . . .	92
9.1.	Página de descarga de EvalCourse . . . . .	95
9.2.	Carpeta de plugins de Eclipse . . . . .	95
10.1.	Diagrama de Gantt del tiempo empleado . . . . .	99
10.2.	Diagrama de Gantt - Comparacion . . . . .	100
10.3.	Resumen de fechas una vez acabado el proyecto . . . . .	101

# Índice de tablas

4.1. Caso de uso Interacción . . . . .	31
4.2. Caso de uso Realizar informes . . . . .	32
4.3. Caso de uso general: Entrega de hitos . . . . .	33
4.4. Caso de uso general: Participación en actividades . . . . .	33
4.5. Caso de uso general: Acceso a las actividades . . . . .	34
4.6. Caso de uso general: Participación en foros . . . . .	34
4.7. Caso de uso general: Acceso al foro . . . . .	35
4.8. Caso de uso general: Entrega de hitos en workshop . . . . .	35
4.9. Caso de uso general: Participación en actividades en workshop . . . . .	37
4.10. Caso de uso general: Acceso al workshop . . . . .	37
4.11. Caso de uso general: Participación activa en el campus . . . . .	38
4.12. Caso de uso general: Seguimiento del campus . . . . .	38
4.13. Caso de uso general: Acceso al campus . . . . .	39
7.1. Listado de pruebas unitarias de código. . . . .	72
7.2. Listado de pruebas unitarias de transformaciones. . . . .	72
7.3. Listado de resultados de ficheros xml. . . . .	72
7.4. Listado de resultados de ficheros xml. . . . .	73
7.5. Pruebas de integración de los módulos. . . . .	73
7.6. Listado del desempeño de los estudiantes en la entrega de actividades. . . . .	75
7.7. Listado del desempeño de los estudiantes en la participación en el foro. . . . .	76
7.8. Tabla explicativa de cómo calificamos las competencias en base a los indicadores. EvalCourse no genera esta tabla. . . . .	78
7.9. Listado del desempeño de los equipos de estudiantes en planificación y trabajo en equipo. . . . .	79
7.10. List of students' performance in forum participation in case study. . . . .	80



Parte I  
Prolegómeno







# Capítulo 1

## Introducción

### 1.1. Motivación

Las universidades hacen cada día más énfasis en la importancia de las competencias como elemento central del desarrollo del alumno. Pero estas competencias no son siempre fáciles de evaluar por parte del profesorado. Hoy en día se disponen de potentes sistemas de aprendizaje, herramientas online que los profesores utilizan para gestionar sus asignaturas. El resultado de la aplicación de esta herramienta, como normal general, proporciona y facilita el trabajo al docente a la hora de evaluar, corregir y medir numéricamente en base a un resultado el trabajo de un alumno. Sin embargo, consideramos que la información contenida en una base de datos de un sistema de este tipo podría ser mucho más aprovechada. Realizando un análisis de los datos se podría conocer gran cantidad de información que nos podría servir de indicador para medir el trabajo del alumno y el desarrollo de ciertas competencias.

La realización de este trabajo comprende, por tanto, no únicamente el desarrollo de un sistema software (análisis de requisitos, diseño, implementación,... etc.), sino también la inmersión en el mundo de las competencias. Saber cuáles son, estimar cómo se pueden medir, colaborar con otros profesores en qué es importante a la hora de valorar lo que han hecho los estudiantes, y ver cómo puede estar plasmado todo eso en la información contenida en la base de datos para aprovecharlo. Con este proyecto no solo aplico y amplío los conocimientos adquiridos en estos años de carrera, además aprendo a gestionar y valorar estas competencias, algo que me será muy útil en mi carrera docente.

La herramienta DSL que hemos desarrollado es EvalCourse. EvalCourse surge en el contexto del aprovechamiento de la tecnología para la mejora del aprendizaje (TEL: Technology-Enhanced Learning). Este es un campo de la investigación que ha traído consigo grandes cambios en colegios, universidades y lugares de trabajo, y que se encuentra envuelto en una revolución tecnológica constante. EvalCourse, es una herramienta que será utilizada para tomar datos que nos sirvan como indicadores del trabajo del desempeño de los alumnos en diversas competencias. Los datos que se puedan tomar o la adaptación de estas herramientas para ser utilizada con otros sistemas, entre otros, son campos que quedan abiertos para futuros trabajos de investigación relacionados con este trabajo, y que serán explicados en mayor detalle en trabajos futuros.

## 1.2. Descripción del sistema actual

Hablar del sistema actual como tal no tendría sentido, ya que no existe un sistema que realice esta tarea. Pero sí, de un modo u otro, los profesores tratan de evaluar dichas competencias. Quizás en algunos casos, de una forma un tanto manual, se establecen indicadores del trabajo de los estudiantes, tanto de la participación de los alumnos en clase presencial, como en los sistemas online. Pero esta forma de medir el desempeño de un alumno con respecto a competencias no es fiable. Se te pueden escapar muchas consideraciones. Uno no puede saber a ciencia cierta, si el esfuerzo en un trabajo en grupo ha sido desarrollado de una manera equitativa por los miembros del equipo. Por ejemplo, podemos basarnos en las competencias de Proyecto Tuning [Tuning, 2012]. Entre ellas, aparece la competencia número 17 (Teamwork: trabajo en equipo). Si un compañero entrega las tareas del grupo fuera del tiempo establecido por el profesor, y se supone que otro compañero necesita estas tareas para completar la suya, el primer no estará siendo un buen compañero. En este caso, el profesor deberá mirar en el campus virtual la fecha de las entregas, para ver si son o no posteriores.

Por esto, con EvalCourse trataremos de encontrar indicadores dentro de los sistemas de aprendizaje que ayuden al docente a valorar el esfuerzo y desempeño de los estudiantes en dichas competencias.

En la Universidad de Cádiz hemos desarrollado algunas herramientas que ayudan al docente en la evaluación de competencias. Estas herramientas se han utilizado en su mayoría aplicadas a wikis. Por un lado, tenemos StatMediaWiki [Palomo-Duarte et al., 2011], herramienta que recopila información de una instalación MediaWiki y la resume en tablas y gráficos que permitan analizar su desarrollo y estado. Esta herramienta permite medir cuantitativamente el trabajo de los alumnos en base a la cantidad de información que ha introducido cada uno en los diferentes artículos del wiki. La experiencia llevada a cabo con esta herramienta ha sido descrita en este artículo [Palomo-Duarte et al., 2012]. Por otro lado, tratando de buscar un punto de vista cualitativo a la información aportada por los alumnos, se desarrolló AssessMediaWiki [Balderas et al., 2012a]. AssessMediaWiki se conectaba también a una instalación de MediaWiki, permitiendo la evaluación de los cambios realizados por los estudiantes entre dos ediciones de un artículo de la wiki, dando la posibilidad de realizar heteroevaluación, autoevaluación y la evaluación entre iguales [Balderas et al., 2012b].

Estas aplicaciones resuelven en cierto modo la problemática para MediaWiki de cómo evaluar las aportaciones a este sistema realizadas por los estudiantes. Pero EvalCourse va más allá, ya que su aplicación va orientada a la búsqueda de indicadores en cualquier sistema LMS. Aunque, en este proyecto, para empezar, se ha desarrollado para Moodle.

## 1.3. Descripción del problema

En este apartado vamos a describir cuales consideramos que son los problemas o necesidades que nos han empujado a desarrollar este proyecto. Como se ha comentado con anterioridad, todo gira alrededor de la evaluación de competencias genéricas.

### 1.3.1. Diagnóstico del problema

La evaluación de competencias genéricas por parte del docente es una tarea difícil. Por un lado, por el carácter subjetivo de la misma, sujeto a criterios no establecidos a priori y que debe resolver el docente. Segundo, por lo no escalable de la tarea. Si ya es difícil, en muchas ocasiones, llegar a poder impartir todo el temario a tiempo, más difícil lo puede ser si es labor del docente diseñar y corregir tareas para evaluar las competencias.

En este trabajo se propone la utilización de los registros de interacción de los alumnos con Moodle. Por tanto, podríamos añadir como otro problema derivado de la pretensión de uso de estos registros, el hecho de que el acceso a la información de los registros de Moodle no está al alcance de un usuario no experimentado.

En resumen, podemos decir que son tres los problemas principales:

- Subjetividad de las evaluaciones de competencias
- Escalabilidad del trabajo de evaluación de competencias
- Complejidad de procesar la información contenida en los registros de Moodle

### 1.3.2. Propuesta de estrategia

En este apartado vamos a describir cada una de las estrategias a seguir para resolver cada problema, así como la justificación de la misma.

#### Subjetividad de las evaluaciones de competencias

<b>Problema</b>	Subjetividad de las evaluaciones de competencias
<b>Propuesta</b>	Evaluar las competencias mediante el uso de indicadores de su desempeño basados en la interacción de los alumnos con el campus virtual
<b>Justificación</b>	A pesar de su importancia, las competencias genéricas son difíciles de evaluar objetivamente. De los registros de actividad del uso del Campus virtual se pueden extraer indicadores objetivos para evaluarlas. Será necesario relacionar indicadores de competencias con actividades de Moodle (foros, taller, etc.) para obtener automáticamente indicadores objetivos que permitan la evaluación de una serie de competencias genéricas.

## Escalabilidad del trabajo de evaluación de competencias

<b>Problema</b>	Escalabilidad del trabajo de evaluación de competencias
<b>Propuesta</b>	Recolección automática de información de los registros del campus virtual
<b>Justificación</b>	La labor del docente es conseguir que sus alumnos consigan las competencias profesionales que capaciten al alumno para el desempeño de la profesión. El profesor, justifica a posteriori una nota para sus estudiantes por medio de exámenes. Si a esta labor, de docencia, diseño y corrección de actividades, hemos de sumar la labor de fomentar el desarrollo y la evaluación de unas competencias genéricas, podríamos decir que el trabajo del profesor deja de ser escalable. Por eso, si aprovechamos la interacción del alumno con el campus virtual para evaluar sus competencias, el trabajo podría automatizarse.

## Complejidad de procesar la información contenida en los registros de Moodle

<b>Problema</b>	Complejidad de procesar la información contenida en los registros de Moodle
<b>Propuesta</b>	Crear un lenguaje de dominio específico
<b>Justificación</b>	Un DSL ofrece un alto nivel de abstracción al usuario, ya que están dirigidos a expertos en el dominio. La semántica del lenguaje es muy cercana al dominio del problema para el que se diseña. Por este motivo, se considera que un DSL es la mejor opción, ya que se usará un lenguaje muy cercano al lenguaje del profesor.

### 1.3.3. Beneficios esperados

Los beneficios esperados son los siguientes:

- Evaluación objetivas de competencias
- Posibilidad de evaluar las competencias genéricas de todos los alumnos de un curso sin que esta tarea suponga un trabajo inabordable
- Sencillez en el trabajo con el DSL para la obtención de la información necesaria

## 1.4. Objetivos y alcance del proyecto

### Objetivo principal

- Creación de un DSL con una sintaxis sencilla e intuitiva que extraiga información importante para el docente sobre la participación de los alumnos en Moodle.

### Subobjetivos

- Crear un metamodelo lo suficientemente genérico para nuestro DSL, de forma que sea válido no sólo para Moodle, sino también para futuras versiones del DSL utilizables en otros sistemas.
- La sintaxis de nuestro DSL deberá ser sencilla y cercana al lenguaje humano.
- Investigar qué información sería de importancia desde el punto de vista de las competencias en un sistema Moodle para generar en nuestro DSL las consultas que nos devuelvan dicha información.
- Investigar la estructura de los ficheros fuente del software Kettle de Pentaho para que nuestro DSL pueda proporcionar como salida un fichero válido para este software.
- Integrar el software como un todo: Eclipse, EvalCourse y Kettle para facilitar al usuario la instalación e implantación del sistema.

## 1.5. Organización del documento

En el documento presentamos primero la planificación llevada a cabo para el desarrollo del proyecto, tanto la estimada a priori como la efectiva a posteriori, así como los costes del mismo. En el siguiente apartado, y para cerrar el prolegómeno, se introduce el enfoque MDE (Model-Driven Engineering), que es el que ha sido el utilizado en este trabajo.

En el apartado de desarrollo se comienza con el análisis de requisitos, incluyendo un análisis del sistema Moodle y del proceso de selección de los datos necesarios para obtener los posibles indicadores de competencias. En el quinto capítulo se trata el diseño del sistema, y en el sexto la implementación del mismo. Concluyendo esta parte con el capítulo de pruebas.

Cerramos el trabajo con la sección del epílogo, en la que se pueden encontrar: el manual de usuario, el manual de instalación y explotación, las conclusiones, la bibliografía utilizada y los artículos enviados a dos congresos a partir de este trabajo.



## Capítulo 2

# Planificación

### 2.1. Metodología de desarrollo

El modelo de desarrollo de software utilizado ha sido el iterativo o incremental. Con este modelo de desarrollo se han creado versiones del producto que poco a poco han ido creciendo en prestaciones hasta llegar al producto final. A continuación se describen las funcionalidades que tenían cada una de estas versiones.

#### 2.1.1. Primer ciclo

El proyecto EvalCourse abarca un conjunto de herramientas software y de pasos que a priori puede resultar difícil de abarcar, por lo que se optó que la primera iteración de desarrollo consistiese únicamente en el cierre del ciclo del producto desde que se crea el metamodelo hasta que se genera el fichero fuente del Pentaho. Además, este primer ciclo comprende el estudio de la herramienta Eclipse Modeling Framework (EMF) [[Gronback and C.,](#) ], Ecore, Ecore Diagram, Xtext y Xtend, así como el manejo del software Pentaho.

- Creación del metamodelo en Ecore
- Creación del proyecto Xtext basado en un metamodelo Ecore
- Estudio del fichero de entrada del software Pentaho
- Programación básica del DSL en Xtend

#### 2.1.2. Segundo ciclo

Viendo que el software es posible se comienzan a plantear los requisitos que este debe cumplir. Se añaden prestaciones como la conexión a la base de datos, mejora de la sintaxis del DSL y la estructura de los ficheros de Pentaho.

- Mejora del metamodelo Ecore
- Regeneración del proyecto Xtext basado en el metamodelo Ecore
- Mejora de la sintaxis del DSL
- Estudio de base de datos Moodle

- Conexión a base de datos Moodle desde el código fuente
- Mejoras en mapeo a Pentaho

### **2.1.3. Tercer ciclo**

En este punto se decide generalizar más el DSL para no pensar tanto en Moodle, sino en un LMS genérico, como es el objetivo del proyecto. A su vez se dan más mejoras en otros aspectos del software que se detallan a continuación.

- Mejora del metamodelo Ecore
- Regeneración del proyecto Xtext basado en el nuevo metamodelo
- Modificación de la sintaxis del DSL basada en el nuevo metamodelo
- Conexión a base de datos mediante fichero externo
- Mejoras en la compilación, permitiéndose el uso de más operadores de Pentaho
- Pruebas

### **2.1.4. Cuarto ciclo**

En este ciclo se pone a punto el software para su finalización.

- Pequeños retoques en la sintaxis del metamodelo
- Se mejora la compilación del DSL
- Se exporta el plugin creado y se personaliza
- Se estudia el software Gephi para su integración con EvalCourse y la generación de gráficos

### **2.1.5. Quinto ciclo**

Finalización, refactorización del código y pruebas del software final.

- Integración de bibliotecas de Pentaho
- Generación de XML Gephi
- Refactorización del código fuente de todo el DSL
- Realización de batería de pruebas

## 2.2. Planificación del proyecto

Se muestra un diagrama de Gantt con la planificación para el desarrollo del proyecto (figura 10.1) y el listado de tareas junto con su estimación temporal detallada (figura 2.5). El tiempo requerido es de 160 días. Como el diagrama de Gantt es muy grande, después de mostrarse al completo, se divide en tres y se muestra en las hojas siguientes (figuras 2.2, 2.3 y 2.4).

El listado de tareas que aparecen en el diagrama de Gantt se listan a continuación, siendo estas las mismas que componen cada uno de los ciclos descritos en el apartado anterior:

- A1: Creación del metamodelo en Ecore
- A2: Creación del proyecto Xtext basado en un metamodelo Ecore
- A3: Estudio del fichero de entrada del software Pentaho
- A4: Programación básica del DSL en Xtend
- B1: Mejora del metamodelo Ecore
- B2: Regeneración del proyecto Xtext basado en el metamodelo Ecore
- B3: Mejora de la sintaxis del DSL
- B4: Estudio de base de datos Moodle
- B5: Conexión a base de datos Moodle desde el código fuente
- B6: Mejoras en mapeo a Pentaho
- C1: Mejora del metamodelo Ecore
- C2: Regeneración del proyecto Xtext basado en el nuevo metamodelo
- C3: Modificación de la sintaxis del DSL basada en el nuevo metamodelo
- C4: Conexión a base de datos mediante fichero externo
- C5: Mejoras en la compilación, permitiéndose el uso de más operadores de Pentaho
- C6: Pruebas
- D1: Pequeños retoques en la sintaxis del metamodelo
- D2: Se mejora la compilación del DSL
- D3: Se exporta el plugin creado y se personaliza
- D4: Se estudia el software Gephi para su integración con EvalCourse y la generación de gráficos
- D5: Integración de bibliotecas de Pentaho
- D6: Generación de XML Gephi
- D7: Refactorización del código fuente de todo el DSL
- D8: Realización de batería de pruebas

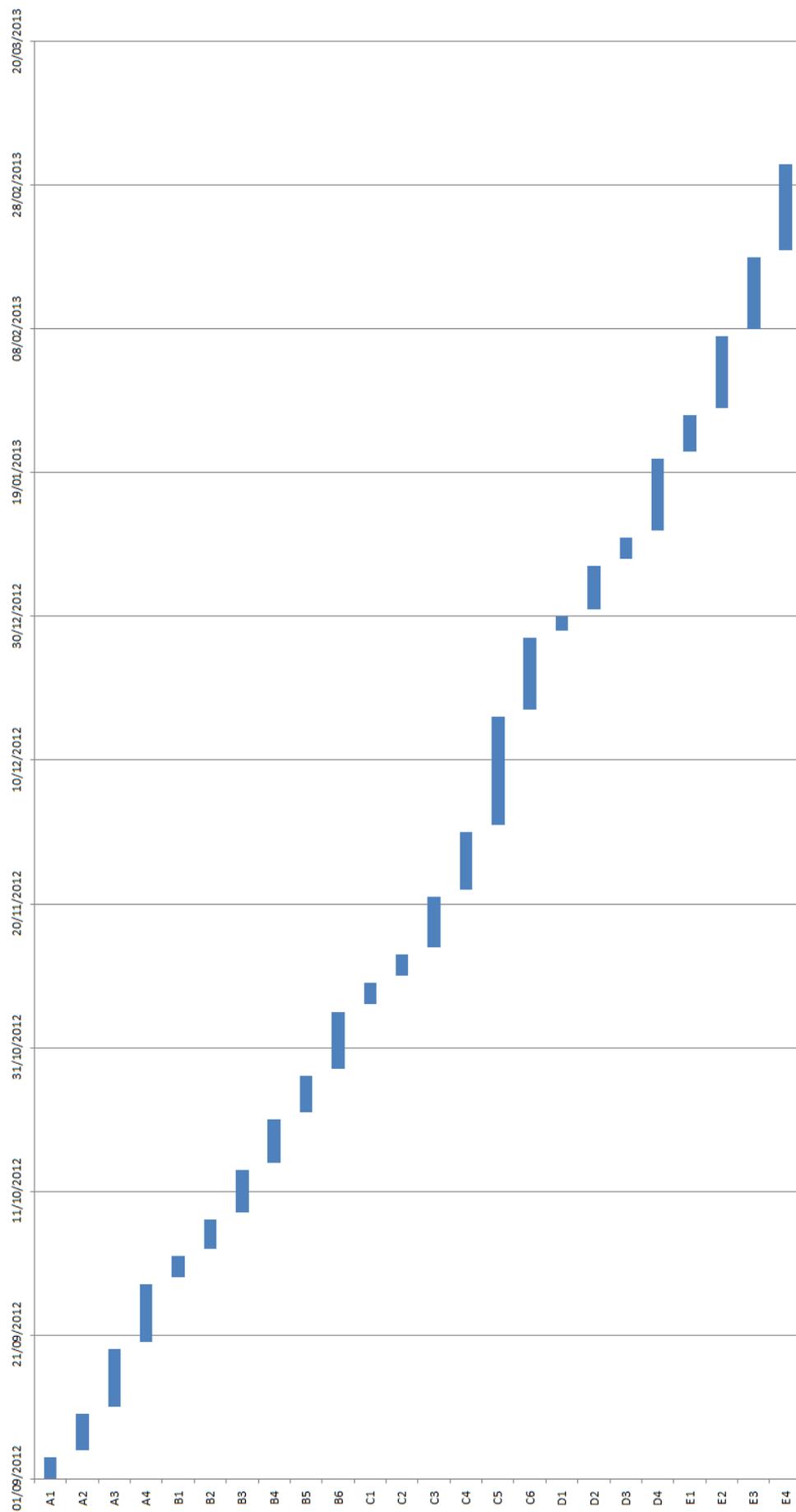


Figura 2.1: Diagrama de Gantt

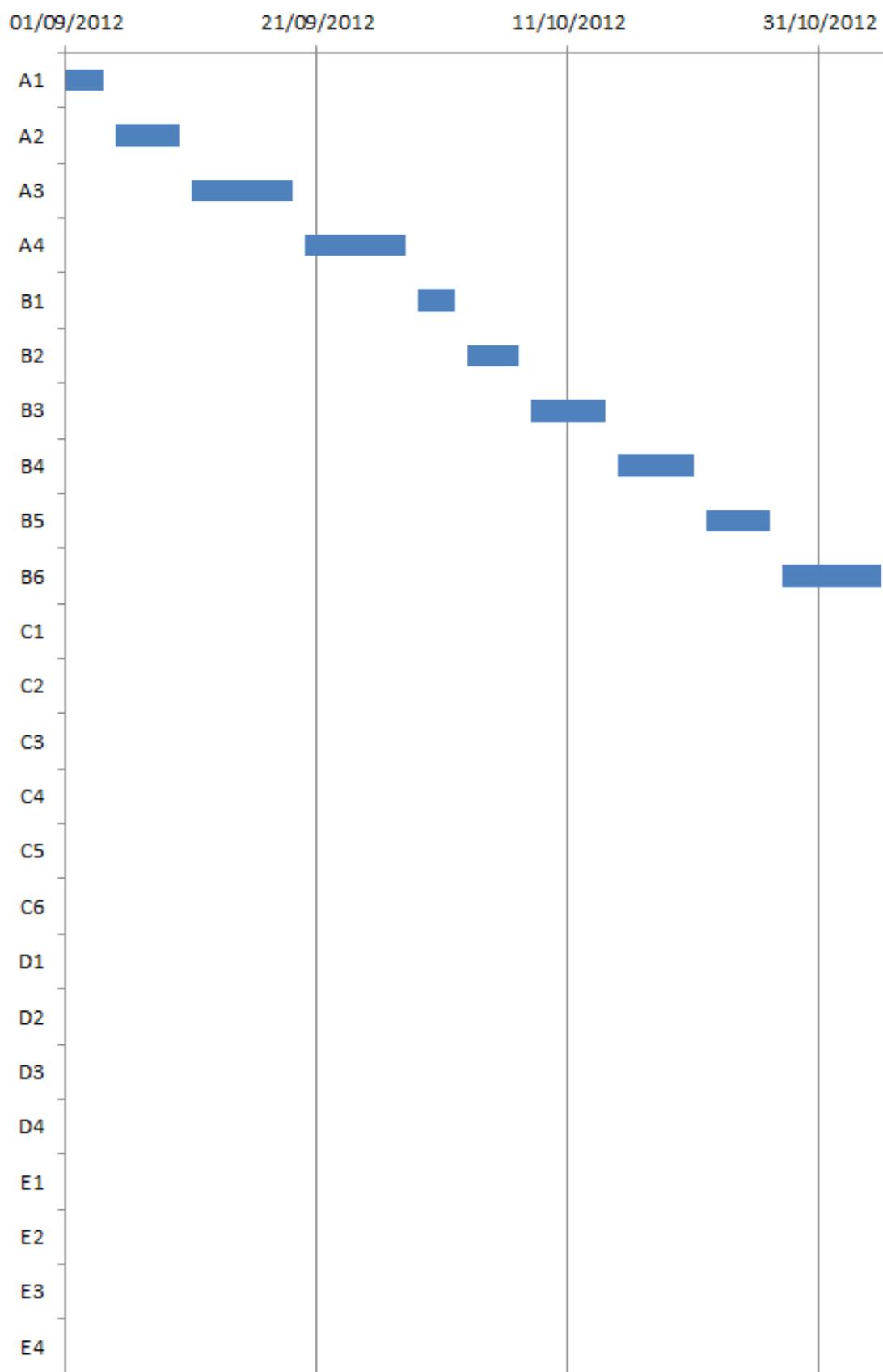


Figura 2.2: Diagrama de Gantt - Parte 1

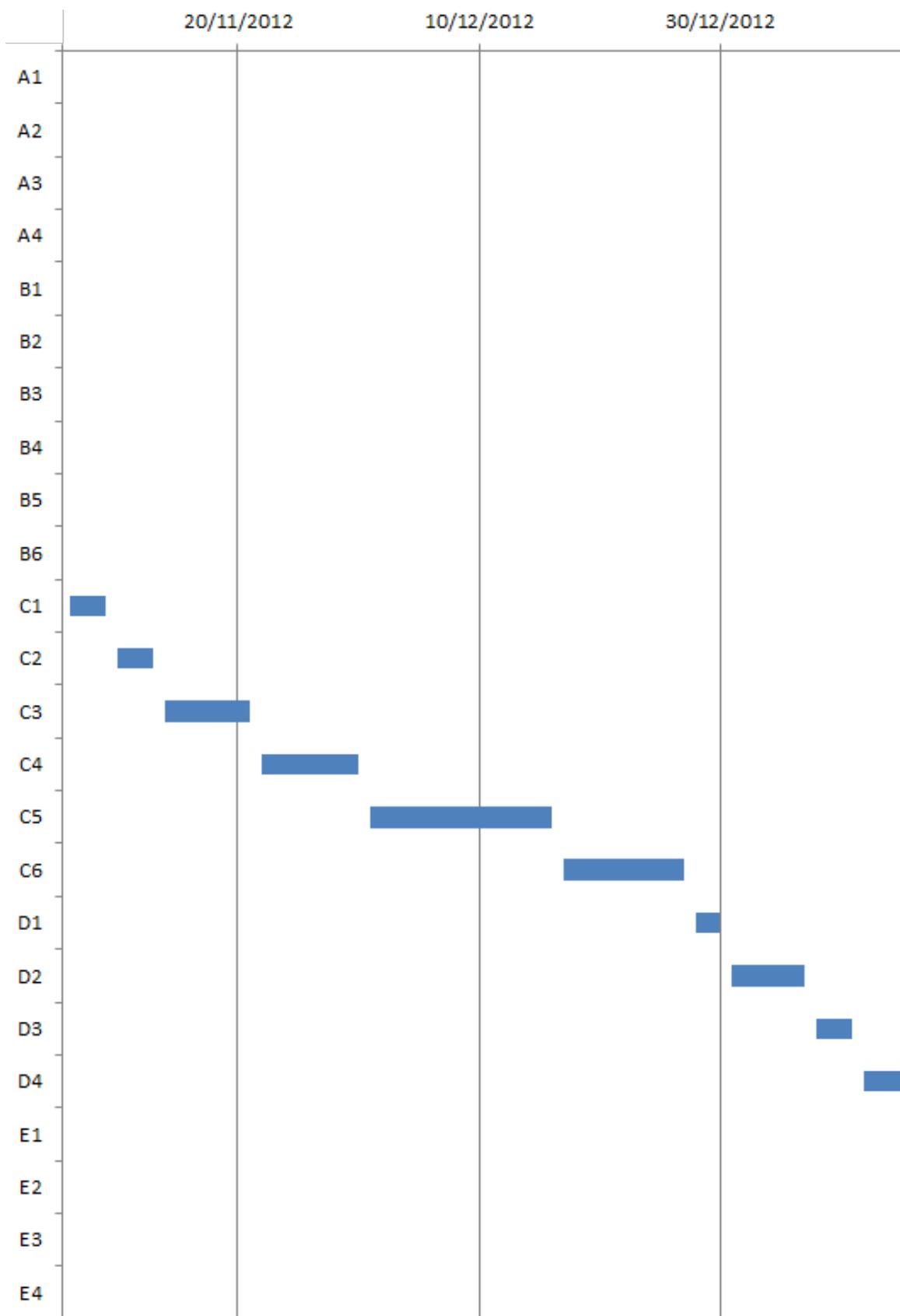


Figura 2.3: Diagrama de Gantt - Parte 2

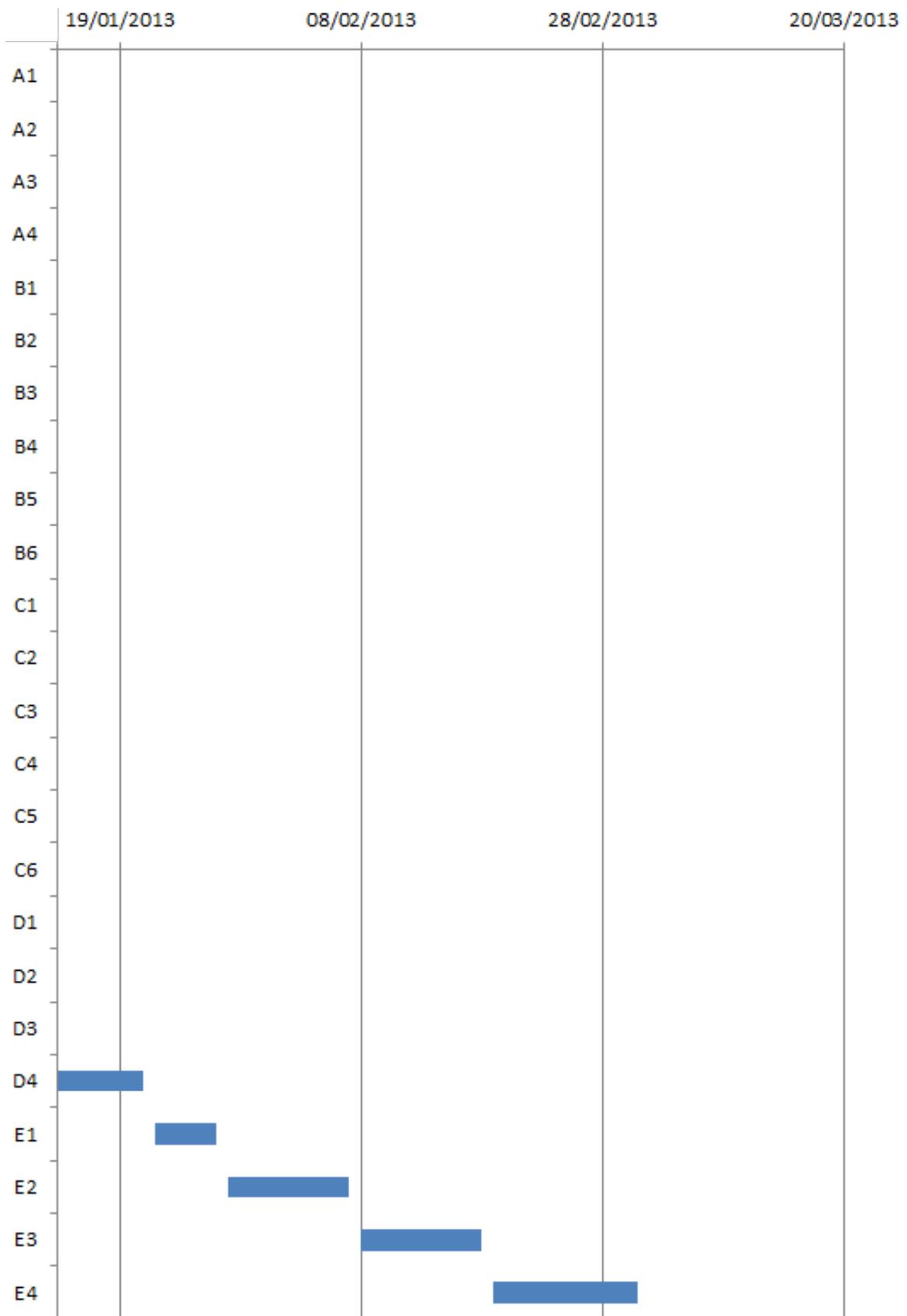


Figura 2.4: Diagrama de Gantt - Parte 3

<b>Actividades</b>	<b>Inicio</b>	<b>Duración (días)</b>	<b>Fin</b>
A1	01/09/2012	3	04/09/2012
A2	05/09/2012	5	10/09/2012
A3	11/09/2012	8	19/09/2012
A4	20/09/2012	8	28/09/2012
B1	29/09/2012	3	02/10/2012
B2	03/10/2012	4	07/10/2012
B3	08/10/2012	6	14/10/2012
B4	15/10/2012	6	21/10/2012
B5	22/10/2012	5	27/10/2012
B6	28/10/2012	8	05/11/2012
C1	06/11/2012	3	09/11/2012
C2	10/11/2012	3	13/11/2012
C3	14/11/2012	7	21/11/2012
C4	22/11/2012	8	30/11/2012
C5	01/12/2012	15	16/12/2012
C6	17/12/2012	10	27/12/2012
D1	28/12/2012	2	30/12/2012
D2	31/12/2012	6	06/01/2013
D3	07/01/2013	3	10/01/2013
D4	11/01/2013	10	21/01/2013
E1	22/01/2013	5	27/01/2013
E2	28/01/2013	10	07/02/2013
E3	08/02/2013	10	18/02/2013
E4	19/02/2013	12	03/03/2013

Figura 2.5: Resumen de fechas planificadas para el desarrollo del proyecto

### 2.3. Costes

A continuación se presenta el estudio y presupuesto de los costes de los recursos (humanos y materiales) descritos anteriormente, necesarios para el proyecto. Para el cálculo de costes de personal nos basamos en las tablas salariales de la UCA para el personal técnico de apoyo contratado laboral [CCOO, 2010].

- Para la realización del proyecto serán necesarias 5,34 personas / mes
- La categoría de los trabajadores será: T. S. Apoyo a la Docencia e Investigación
- Salario base mensual de los trabajadores + Complemento categoría: 2.746,65 €
- Total estimado = 5,34 trabajadores x 2.746,65 € = **14.648,80 €**



## Capítulo 3

# Ingeniería Dirigida por Modelos

### 3.1. MDE: Ingeniería Dirigida por Modelos

MDE<sup>1</sup> es un nuevo enfoque en Ingeniería del Software. En él se utilizan modelos como artefactos software. Es una metodología de desarrollo de software que se centra en la creación y explotación de modelos de dominio (es decir, representaciones abstractas de los conocimientos y las actividades que rigen en un dominio de aplicación particular), y no en los conceptos o algoritmos informáticos [Stahl et al., 2006].

Este enfoque se utiliza para aumentar la productividad al maximizar la compatibilidad entre los sistemas (a través de la reutilización de modelos estandarizados), lo que simplifica el proceso de diseño (a través de los modelos de los patrones de diseño recurrentes en el dominio de aplicación), y promover la comunicación entre los individuos y los equipos que trabajan en el sistema (a través de la estandarización de la terminología y de las mejores prácticas aplicadas en el dominio de aplicación).

Un paradigma de modelado para el MDE se considera eficaz si sus modelos tienen sentido desde el punto de vista de un usuario que esté familiarizado con el dominio, y si pueden servir de base para los sistemas de aplicación. El modelo se debe desarrollar a través de una amplia comunicación entre los jefes de producto, diseñadores, desarrolladores y usuarios del dominio de aplicación. A medida que los modelos se acercan a la terminación, que permiten el desarrollo de software y sistemas.

### 3.2. DSL: Lenguaje de Dominio Específico

En el desarrollo de software, un lenguaje específico de dominio (domain-specific language - DSL) [Fowler and Parsons, 2011] [dosideas.com, 2009] es un lenguaje de programación dedicado a un problema de dominio en particular, o una técnica de representación o resolución de problemas específica. Este concepto no es nuevo, ya que desde siempre existieron lenguajes de programación de propósito específico.

---

<sup>1</sup>Model-Driven Engineering

Lo opuesto a un lenguaje específico de dominio son los lenguajes de programación de propósito general, como C o Java; y los lenguajes de modelado de propósito general, como UML.

Los DSL se crean específicamente para resolver problemas dentro de un dominio en particular, y no están pensados para resolver problemas fuera de este dominio (aunque pueda ser técnicamente posible). En cambio, los lenguajes de propósito general se crean para resolver problemas en muchos dominios. Un dominio también puede ser un área de negocio específica. En ocasiones es difícil discernir las ventajas y desventajas entre las construcciones de los DSL y de los lenguajes de propósito general.

### **Ventajas de los DSL**

- Los DSL permiten expresar soluciones usando los términos y el nivel de abstracción apropiado para el dominio del problema. En consecuencia, los mismos expertos de dominio pueden comprender, validar, modificar y a menudo desarrollar programas en DSL.
- Es código auto-documentado.
- Los DSL mejoran la calidad, productividad, confianza, mantenibilidad, portabilidad y reusabilidad de las aplicaciones.
- Los DSL permiten validaciones a nivel del dominio. Mientras las construcciones del lenguaje estén correctas, cualquier sentencia escrita puede considerarse correcta.

### **Desventajas de los DSL**

- El costo de aprender un nuevo lenguaje vs. su aplicación limitada.
- El costo de diseñar, implementar y mantener un DSL y las herramientas para trabajar con él.
- Encontrar, establecer y mantener el alcance adecuado.

#### **3.2.1. Fases en el desarrollo de un DSL externo**

- Diseño del metamodelo
- Desarrollo del formato de representación
- Generación de transformaciones
- Construcción de un IDE

**Metamodelado** es el análisis, diseño y construcción de los metamodelos necesarios para cubrir un determinado tipo de problemas, así como las reglas y restricciones aplicables.

Un **metamodelo** es el conjunto de conceptos del dominio a modelar (metaclases) y las relaciones entre ellos (meta-asociaciones).

El **objetivo** es definir los conceptos y relaciones del dominio del problema que queremos abordar, mediante un diagrama de clases. Estos se definen de forma abstracta, independientemente del formato de representación deseado. Los metamodelos definirán la sintaxis abstracta de nuestro lenguaje.

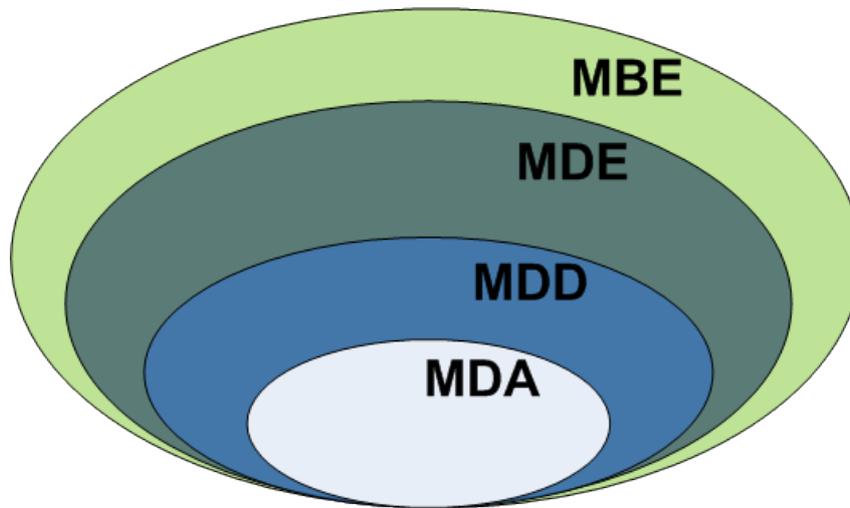


Figura 3.1: Gráfico de relaciones entre los modelos dirigidos

### 3.3. MDE y DSL

Después de leer la información anterior o como se puede leer en Model-Based DSL Frameworks [Kurtev et al., 2006], se aprecia como MDE y DSL convergen en varios aspectos. La noción de dominio es común a estas dos áreas de estudio y comparten una gran cantidad de objetivos y prácticas comunes. Del mismo modo que MDE es una generalización de la MDA (Model-Driven Architecture), la ingeniería de DSL puede ser visto como una generalización de MDE [Haan, ] (figura 3.1 de [Cabot, ]).



Parte II

Desarrollo







## Capítulo 4

# Análisis de Requisitos

### 4.1. Catálogo de actores

- Profesor: El actor profesor, docente o investigador que utilizará la herramienta para obtener las evidencias del desarrollo de competencias de sus alumnos mediante la interacción de estos con el curso de Moodle.
- LMS: El actor LMS será el sistema del que se capturarán los datos susceptibles de ser analizados y procesados. En este caso LMS siempre será Moodle.

### 4.2. Requisitos funcionales

Existen dos casos de uso principales:

- Interacción
- Realizar informes

**Interacción** (tabla 4.1). Al tratarse este proyecto de un lenguaje de programación, el escenario de los diferentes casos de uso que nos podemos encontrar es simple. La variedad radica en el conjunto de consultas que el lenguaje admite, pero la interacción es siempre la misma. Por eso se va a establecer un caso de uso general, aplicable en sus pasos a todos los casos de uso.

En este caso de uso (figura 4.1), el profesor ingresa una consulta en el sistema. El sistema procesa la consulta y genera un fichero de transformación para el programa Pentaho, un fichero XSL para la posterior traducción del listado en xml a Gephi y un fichero en Java para ejecutar el siguiente paso.

**Realizar informes** (tabla 4.2). Una vez que se ha obtenido el fichero de transformación de Pentaho, llega el momento de aplicar dicho fichero a Moodle. Esta aplicación se debe poder realizar desde el propio EvalCourse. Este caso de uso representa el proceso de la obtención de los informes.

En este caso de uso (figura 4.2) el usuario ejecuta el fichero Java, el sistema utiliza el fichero de transformación de Pentaho para consultar la base de datos del Moodle (LMS) y devuelve los datos en el formato xml. A continuación, se procesa el fichero xml con la transformación XSL y se obtiene el fichero en formato gráfico Gephi.

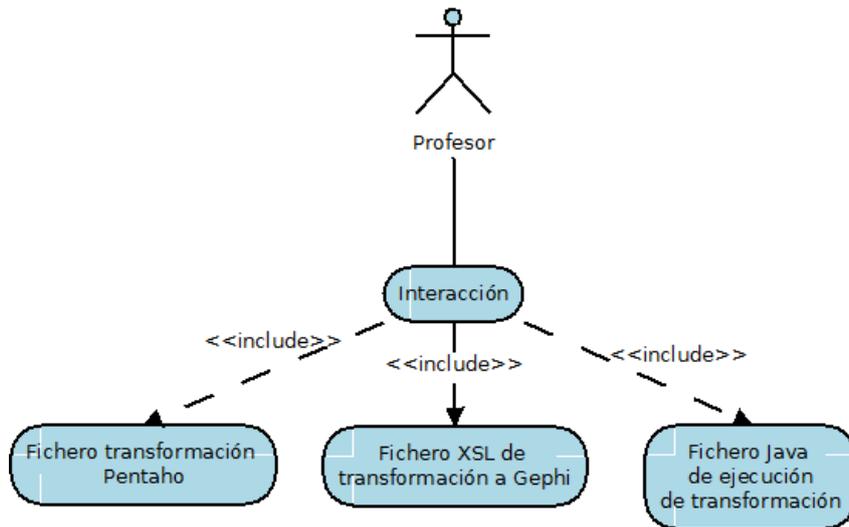


Figura 4.1: Casos de uso Interacción

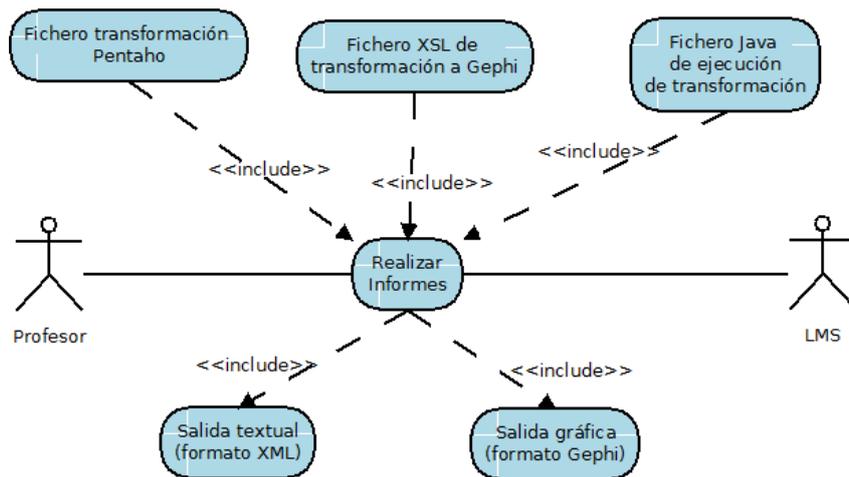


Figura 4.2: Casos de uso Realizar Informes

<b>Caso de uso</b>	Interacción.
<b>Descripción</b>	El usuario escribe una consulta en el sistema para conocer algún tipo de información referente al trabajo de los estudiantes. Este es el súper-caso del que los demás heredarán.
<b>Actores</b>	Docente.
<b>Precondiciones</b>	
<b>Pasos</b>	<ul style="list-style-type: none"> <li>▪ Análisis léxico, sintáctico y semántico de la consulta</li> <li>▪ Construcción del fichero de transformación Pentaho</li> <li>▪ Construcción del fichero de transformación XSL para transformar listado Pentaho a grafo Gephi</li> <li>▪ Construcción del fichero de transformación Pentaho para utilizar el XSL</li> <li>▪ Construcción de fichero Java para ejecutar las transformaciones</li> </ul>

Cuadro 4.1: Caso de uso Interacción



Figura 4.3: Herencia de casos de usos del súper-caso de interacción

#### 4.2.1. Caso de uso general: interacción

*Interacción* es un súper-caso de uso. A partir de él se describirán el resto de casos de uso en base a las consultas que se podrán realizar. Esta situación es la que se conoce como herencia de casos de uso. Un súper-caso general del que heredan los demás casos de uso se detalla en la figura 4.3.

#### 4.2.2. Caso de uso: consultar evidencias en actividades

El caso de uso *consultar evidencias en actividades* es el caso de uso general para las consultas que extraerán información referente a actividades. De éste heredarán otros tres casos de uso (figura 4.4):

- Entrega de hitos (cuadro 4.3)
- Participación en actividades (cuadro 4.4)
- Acceso a las actividades (cuadro 4.5)

<b>Caso de uso</b>	Realizar informes
<b>Descripción</b>	El usuario ejecuta el fichero de transformación que obtiene como salida a la consulta y recibe como salida dos ficheros: un listado con la información solicitada y una representación gráfica de dicha información.
<b>Actores</b>	Docente.
<b>Precondiciones</b>	Deben haberse generado el fichero de transformación en Pentaho, de extensión .ktr, y el fichero Java necesario para ejecutar la transformación.
<b>Pasos</b>	<ul style="list-style-type: none"> <li>▪ Ejecución fichero Java</li> <li>▪ Generación listado en formato xml</li> <li>▪ Generación gráfico en formato Gephi</li> </ul>

Cuadro 4.2: Caso de uso Realizar informes



Figura 4.4: Herencia de casos de usos de consultar evidencias en actividades

#### 4.2.3. Caso de uso: consultar evidencias en foros

El caso de uso *consultar evidencias en foros* es el caso de uso general para las consultas que extraerán información de los foros. De éste heredarán otros dos casos de uso (figura 4.5):

- Participación en foros
- Acceso al foro

#### 4.2.4. Caso de uso: consultar evidencias en workshops

El caso de uso *consultar evidencias en workshops* es el caso de uso general para las consultas que extraerán información referente a workshops. De éste heredarán otros tres casos de uso (figura 4.5):

- Entrega de hitos en workshop (cuadro 4.8)
- Participación en actividades en workshop (cuadro 4.9)
- Acceso al workshop (cuadro 4.10)

<b>Caso de uso</b>	Entrega de hitos
<b>Descripción</b>	Solicitud de información sobre la entrega a tiempo de las actividades. Esta consulta puede ir dirigida a un conjunto de actividades referenciadas por su id, o a todas las actividades del curso.

Cuadro 4.3: Caso de uso general: Entrega de hitos

<b>Caso de uso</b>	Participación en actividades
<b>Descripción</b>	Solicitud de información sobre la participación de los alumnos en las actividades. Esta consulta puede ir dirigida a un conjunto de actividades referenciadas por su id, o a todas las actividades del curso.

Cuadro 4.4: Caso de uso general: Participación en actividades

#### 4.2.5. Caso de uso: evidencias de campus

El caso de uso *consultar evidencias en campus* es el caso de uso general para las consultas que extraerán información referente a la actividad en el campus virtual. De éste heredarán otros tres casos de uso (figura 4.7):

- Participación activa en el campus (cuadro 4.11)
- Seguimiento del campus (cuadro 4.12)
- Acceso al campus (cuadro 4.13)

### 4.3. Requisitos de información

Los datos que EvalCourse necesita gestionar son los registros que almacenan información sobre la interacción de los alumnos con Moodle. Por esto, el diagrama conceptual que estudiaremos es el del entorno de aprendizaje virtual Moodle. A partir de éste, es necesario realizar un mapeo



Figura 4.5: Herencia de casos de usos de consultar evidencia en foros

<b>Caso de uso</b>	Acceso a las actividades
<b>Descripción</b>	Solicitud de información sobre el acceso de los alumnos a las actividades. Esta consulta puede ir dirigida a un conjunto de actividades referenciadas por su id, o a todas las actividades del curso.

Cuadro 4.5: Caso de uso general: Acceso a las actividades

<b>Caso de uso</b>	Participación en foros
<b>Descripción</b>	Solicitud de información sobre los alumnos que escriben en el foro. Esta consulta puede ir dirigida a un conjunto de actividades referenciadas por su id, o a todas las actividades del curso.

Cuadro 4.6: Caso de uso general: Participación en foros

entre qué queremos obtener, y en qué registro de Moodle nos proporcionaría dicha información.

Los lenguajes específicos de dominio (DSL) son herramientas para abordar problemas concretos. Para construir un DSL partiendo de un enfoque MDE, lo primero que hemos de hacer es diseñar el metamodelo. El metamodelo es el modelo del modelo, es el conjunto de conceptos del dominio a modelar (metaclases) y las relaciones entre ellos. Nuestra metamodelo deberá abarcar el problema de la evaluación de evidencias, así como las reglas y las restricciones a aplicar.

### 4.3.1. Moodle

Moodle cuenta con 186 tablas. Puede ver su diagrama en la figura 4.8. Una estructura de base de datos de un tamaño bastante considerable. EvalCourse solo necesitará parte de la información ahí almacenada. Por esto, será necesario un estudio de las estructuras de datos que almacenan dicha información. En los siguientes puntos se detallarán estos almacenes.

#### Gestión de usuarios

Toda la información que obtenemos del sistema es siempre sobre un conjunto de usuarios. Esos usuarios pertenecerán a un curso concreto, supuestamente al curso que imparte el profesor que está usando la herramienta. Además, esos usuarios en el curso cumplen un rol en particular, el



Figura 4.6: Herencia de casos de usos de consultar evidencias en workshops

<b>Caso de uso</b>	Acceso al foro
<b>Descripción</b>	Solicitud de información sobre los alumnos que acceden en el foro. Esta consulta puede ir dirigida a un conjunto de actividades referenciadas por su id, o a todas las actividades del curso.

Cuadro 4.7: Caso de uso general: Acceso al foro

<b>Caso de uso</b>	Entrega de hitos en workshop
<b>Descripción</b>	Solicitud de información sobre las entrega de actividades en el taller. Esta consulta puede ir dirigida a un conjunto de actividades referenciadas por su id, o a todas las actividades del curso.

Cuadro 4.8: Caso de uso general: Entrega de hitos en workshop

rol de estudiantes. Estos datos hay que tenerlos en cuenta a la hora de filtrar la información. Los fragmentos del diagrama que intervienen son los de los módulos de usuarios 4.9 y de roles 4.10. Las tablas que se necesitan son las siguientes:

- **mdl\_user**: Esta tabla almacena toda la información de los usuarios. Sera necesaria para relacionarla en consultas posteriores con los usuarios a los que pertenecen las evaluaciones obtenidas de otras fuentes de información.
- **mdl\_course**: Esta tabla almacena toda la información de los cursos. Sera necesaria para relacionarla en consultas posteriores con la pertenencia de los usuarios, actividades, foros, ... etc., con respecto al curso que se está evaluando.
- **mdl\_role y mdl\_role\_assignments**: Estas tablas son necesarias para conocer el rol del usuario que se evalúa. No habría otra forma de saber si un usuario es profesor o alumno.
- **mdl\_context**: Indica hasta qué nivel puede actuar un usuario con un rol dado.

### Gestión de foros

La información que se necesita del sistema es de tipo cuantitativa. Por tanto, lo que los alumnos escriban en el foro no será de interés para la aplicación. Al menos, teniendo en cuenta su alcance actual. El único requisito necesario es que debemos saber quién escribe y dónde lo escribe. El

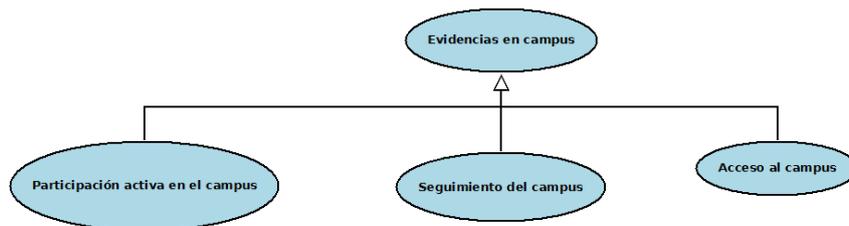


Figura 4.7: Herencia de casos de usos de consultar evidencias en campus

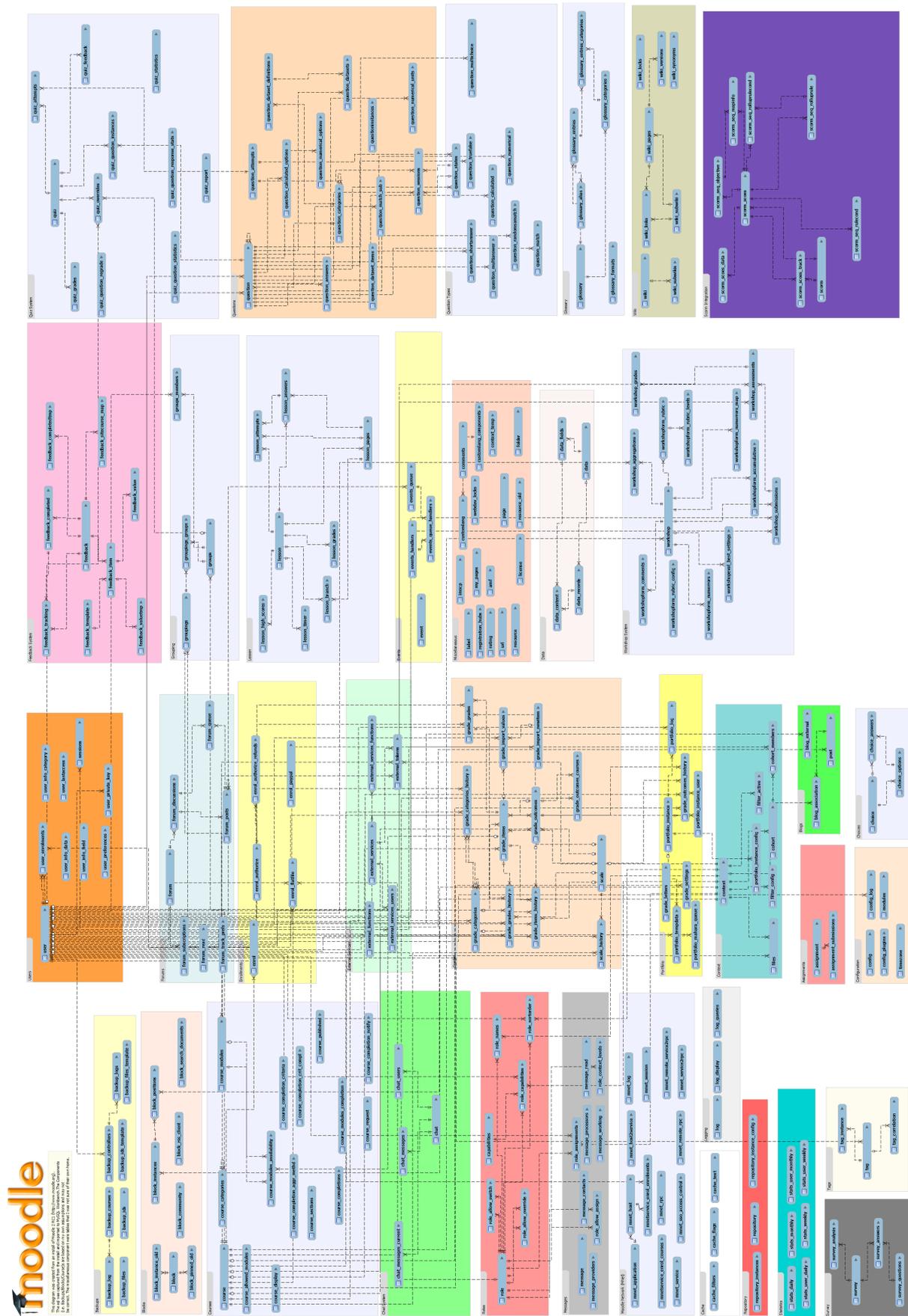


Figura 4.8: Diagrama de Moodle V.2

<b>Caso de uso</b>	Participación en actividades en workshop
<b>Descripción</b>	Solicitud de información sobre las realización de actividades en el taller. Esta consulta puede ir dirigida a un conjunto de actividades referenciadas por su id, o a todas las actividades del curso.

Cuadro 4.9: Caso de uso general: Participación en actividades en workshop

<b>Caso de uso</b>	Acceso al workshop
<b>Descripción</b>	Solicitud de información sobre el acceso a las actividades en el taller. Esta consulta puede ir dirigida a un conjunto de actividades referenciadas por su id, o a todas las actividades del curso.

Cuadro 4.10: Caso de uso general: Acceso al workshop

quién ya se obtuvo con anterioridad de la tabla de usuarios. Pero para saber en qué foro se hace es necesario acceder a la tablas de los mensajes en el foro 4.11. Las tabla que se necesita es la siguiente:

- **mdl\_forum\_posts**: Esta tabla almacena toda la información de los mensajes de los usuarios. Entre toda esa información está la que nos interesa. A partir del atributo *userid* se sabe qué usuario escribió. A partir del atributo *parent* se sabe si el usuario inició un debate o participó en él. Si *parent* es cero, significa que es el comentario raíz del que parte la conversación, por lo que sabemos que el usuario inició el debate. Sin embargo, si es distinto de cero, significa que el comentario es respuesta a un mensaje anterior. El origen del mensaje podría ser de utilidad al docente para según qué competencia quiera evaluar.

## Gestión de actividades

Los estudiantes realizan una serie de actividades a lo largo del curso. Estas actividades suelen consistir en un fichero o conjunto de ficheros que éstos han de subir a la plataforma virtual y que

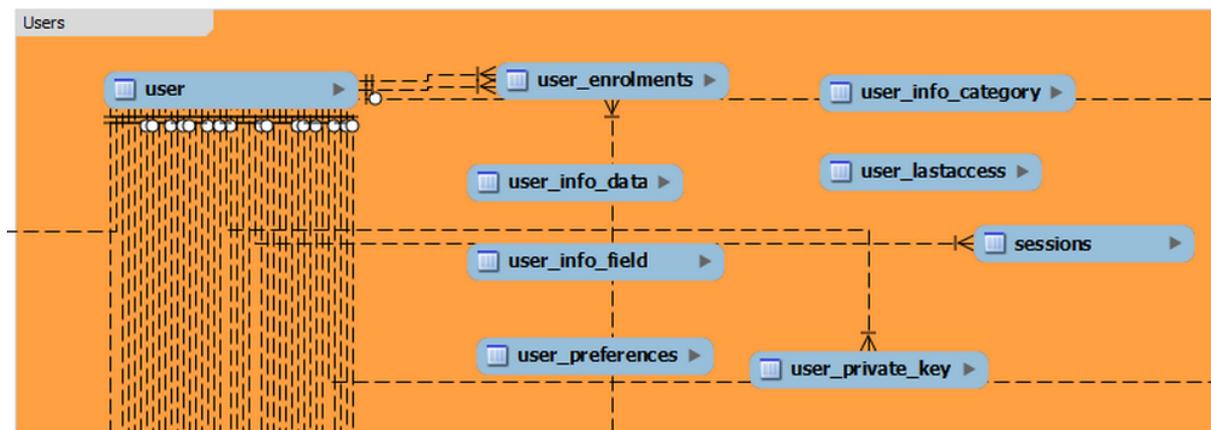


Figura 4.9: Diagrama del módulo de usuarios de Moodle V.2

<b>Caso de uso</b>	Participación activa en el campus
<b>Descripción</b>	Solicitud de información sobre la participación activa en el campus. Esta consulta puede ir dirigida a un conjunto de actividades referenciadas por su id, o a todas las actividades del curso.

Cuadro 4.11: Caso de uso general: Participación activa en el campus

<b>Caso de uso</b>	Seguimiento del campus
<b>Descripción</b>	Solicitud de información sobre la participación pasiva en el campus. Esta consulta puede ir dirigida a un conjunto de actividades referenciadas por su id, o a todas las actividades del curso.

Cuadro 4.12: Caso de uso general: Seguimiento del campus

posteriormente el docente evalúa. Lo que se pretende de las actividades es saber qué cantidad de estas ha entregado un estudiante. De las que ha entregado, cuáles han sido dentro y fuera del plazo establecido. Las tablas que almacenan esta información de actividades 4.12 son las siguientes:

- **mdl\_assign (assignment)**: Almacena información general de las tareas, su relación con el curso, las fechas disponibles de entrega,... etc.
- **mdl\_assign\_submission**: Esta tabla almacena información concreta sobre la entrega de cada usuario. En qué fecha lo hizo, qué entregó, ... etc.

## Gestión de talleres

El taller (workshop) es una actividad muy interesante que nos ofrece la plataforma virtual. En él los estudiantes han de realizar alguna actividad, como en las mencionadas en el apartado anterior. La diferencia radica en que en este caso puedes establecer diferentes formas de evaluación. Puedes indicar que sean otros alumnos los que evalúen a sus compañeros, o que cada alumno evalúe a varios alumnos y luego hacer la media de esta nota. Esta actividad, que es mucho más potente de lo que aquí se describe, permite al docente un gran abanico de posibilidades a la hora de evaluar competencias del alumnado. El módulo del diagrama que interviene en el taller es el del

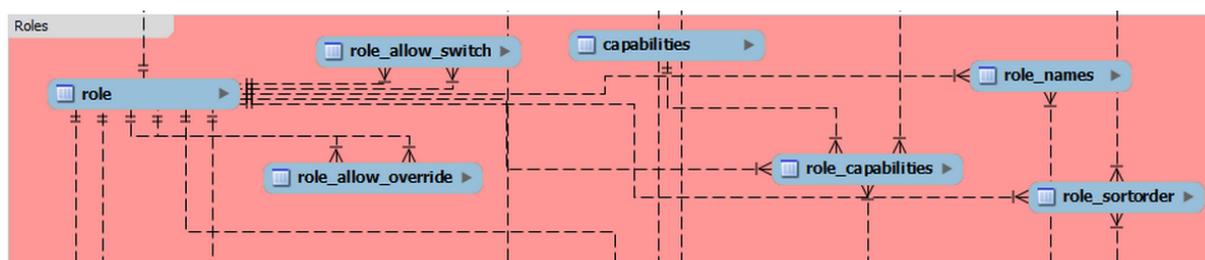


Figura 4.10: Diagrama del módulo de roles de Moodle V.2

<b>Caso de uso</b>	Acceso al campus
<b>Descripción</b>	Solicitud de información sobre los accesos al campus virtual.

Cuadro 4.13: Caso de uso general: Acceso al campus

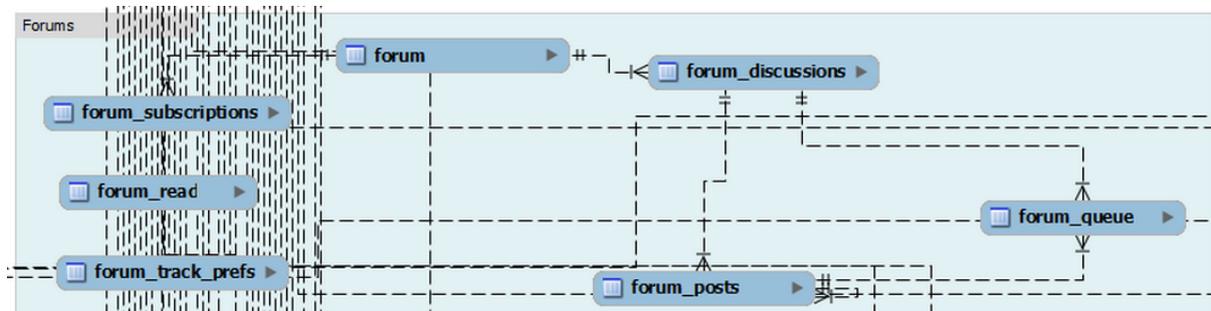


Figura 4.11: Diagrama del módulo de foros de Moodle V.2

workshop 4.13. Aunque dada su complejidad, se disponen de más tablas que en las actividades corrientes, para nuestro caso las tablas cumplen la misma función que las anteriores.

- **mdl\_workshop\_assessments**: Almacena información general de las tareas, su relación con el curso, las fechas disponibles de entrega,... etc.
- **mdl\_workshop\_submission**: Esta tabla almacena información concreta sobre la entrega de cada usuario y sobre el usuario que la corrige. En qué fecha lo hizo, qué entregó, ... etc.

### Gestión de accesos

Uno de los aspectos a controlar por parte de la herramienta es la cantidad de veces que acceden los usuarios a los cursos. Para eso es necesario acceder al módulo de logging 4.14. La tabla que necesitaremos es la siguiente:

- **mdl\_log**: Almacena información acerca de la acción que realizó el usuario, en qué fecha y desde dónde lo hizo. Importante para saber si los alumnos acceden al campus virtual, si acceden al foro, a los documentos,... etc.

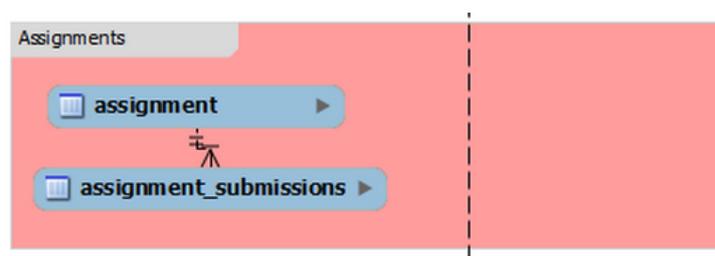


Figura 4.12: Diagrama del módulo de actividades de Moodle V.2

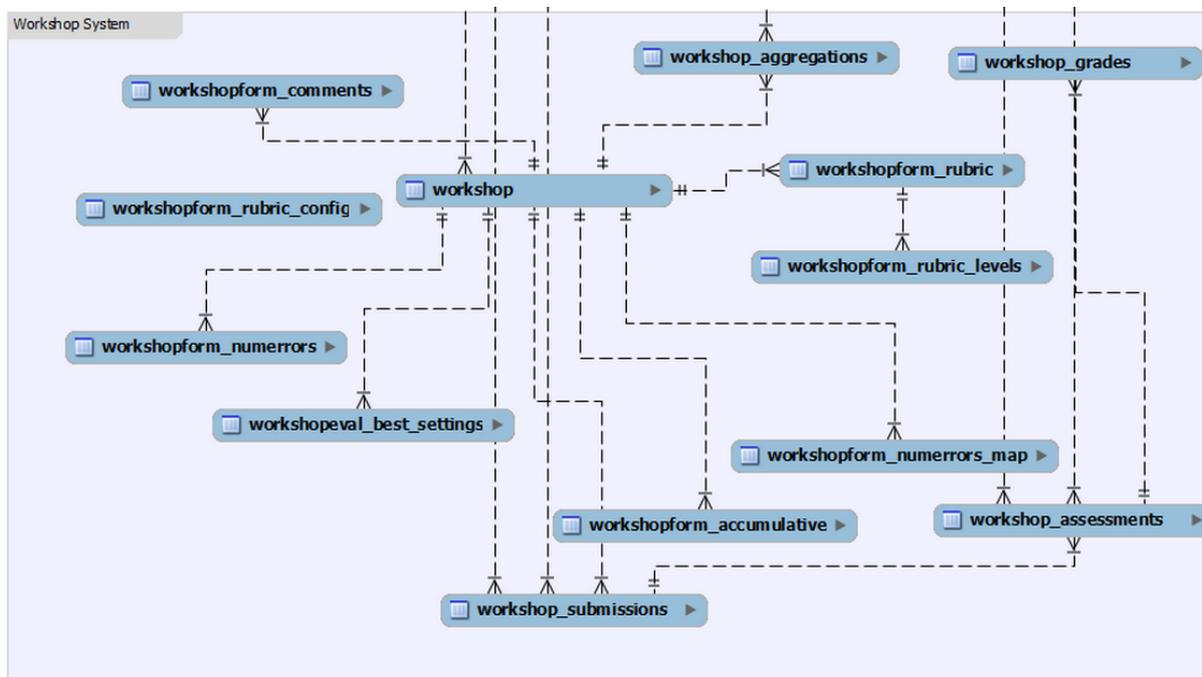


Figura 4.13: Diagrama del módulo de workshop de Moodle V.2



Figura 4.14: Diagrama del módulo de logging de Moodle V.2

### 4.3.2. Metamodelo

El metamodelo de EvalCourse 4.15 define los conceptos y relaciones del dominio del problema de la evaluación de evidencias del desempeño de los alumnos en el campus virtual. Los conceptos que formarán parte del mismo son los siguientes:

- **Evidence:** es el concepto que identifica la evidencia que queremos evaluar. Se trata del concepto raíz del que todos parten. Su único atributo es el nombre de la evidencia, que a su vez dará nombre a los ficheros de salida que generará el sistema.
- **Student:** representa a los elementos que queremos evaluar. En este caso serán los estudiantes.
- **Target:** representa el objetivo que queremos extraer de los estudiantes, obtenerlos como tal o contarlos.
- **Indication:** representa el elemento con el que vamos a evaluar al concepto Student. Por un lado necesita que se defina con qué indicio (sign) evaluaremos. Por otro lado, hay que definir la herramienta (tool) que utilizaremos para evaluar. También se puede definir entre qué fechas se obtendría la información.
- **Entity:** este concepto, que puede estar o no, indica sobre qué entidades actuará la evidencia. Si no se indica nada, por ejemplo, tomará los datos de todas las actividades. Pero si se indican uno o varios identificadores, actuará sobre las herramientas que tengan dichos identificadores.
- **Connection:** concepto que representa la conexión con la base de datos del entorno de aprendizaje virtual Moodle. Tiene como parámetros todos los valores necesarios para establecer la conexión. El usuario deberá conocer estos datos para poder conectar la herramienta y extraer la información.

## 4.4. Requisitos no funcionales

Otros requisitos no funcionales que debe satisfacer el sistema son los siguientes:

- **Portabilidad:** El software debe ser capaz de ejecutarse en las diferentes plataformas para PC que habitualmente utilizan los profesores. El software ha sido desarrollado con Eclipse Modeling Framework, y se utiliza como un plugin de Eclipse. Por tanto, si en la plataforma a ser utilizado el programa se puede instalar Eclipse, entonces, EvalCourse podrá ser utilizado en los sistemas operativos más utilizados en el mercado (Linux, Windows y Mac OS).
- **Seguridad:** EvalCourse se va a conectar a una plataforma en la que conviven muchos cursos. EvalCourse solo debe poder tener acceso a la información del curso al que está enlazada en ese momento, que es el curso al que pertenece el docente que está usando el DSL.
- **Fiabilidad:** EvalCourse trabajará con una gran cantidad de información. La interacción de un grupo de 30 alumnos con un curso del campus virtual genera muchísimos registros. Registros a los que tendrá acceso EvalCourse y que ha de manejar de forma fiable. Los resultados han de ser los que se esperan y deben representar la realidad del curso. De ahí,

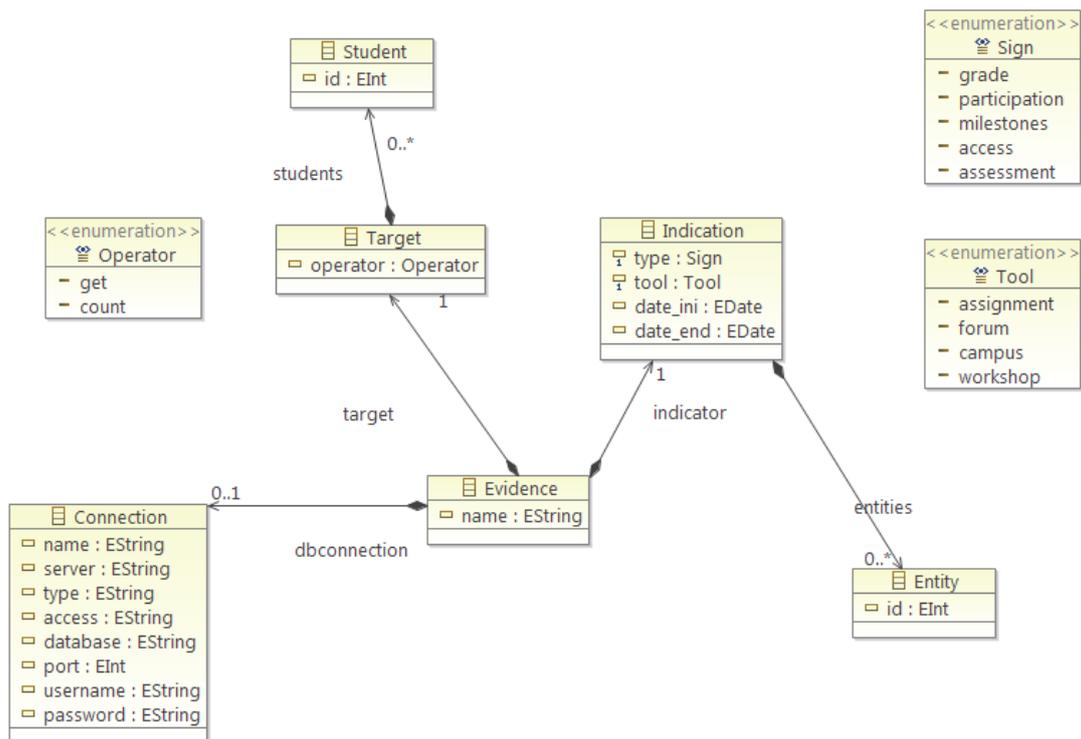


Figura 4.15: Metamodelo de EvalCourse

la importancia de un buen análisis de la base de datos de Moodle para que la posterior recolección de información sea fiable.

- Comunicaciones con sistemas externos:** EvalCourse se conectará a la base de datos de Moodle. Los permisos para poder conectarse o no a la herramienta quedan fuera del ámbito del proyecto. En nuestro caso, la conexión se realiza por medio del fichero de transformación que se crea en Pentaho. Pero el pc desde el que se ejecuta EvalCourse ha de tener permisos en el sistema que almacena Moodle para poder interactuar.
- Extensibilidad:** El proyecto que aquí se presenta es sólo la primera piedra de un software más ambicioso. Las competencias han de depurarse, de manera que se puedan extraer otros indicadores de la herramienta Moodle, así como extender el DSL para ser utilizado con otros entornos de aprendizaje.
- Usabilidad:** La facilidad con la que los docentes puedan utilizar EvalCourse para extraer los indicadores de competencias es uno de los objetivos principales de la herramienta. EvalCourse está orientado a cualquier usuario, no teniendo porque estar relacionado con el área de la informática. Debe ser fácil de usar, entendible e intuitivo.
- Privacidad:** Cumplir la LOPD. La Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal, (LOPD), es una Ley Orgánica española que tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor, intimidad y privacidad personal y familiar. El único dato de carácter personal que va a tratar EvalCourse es el nombre y apellidos de los alumnos. El resto de datos

Seleccione los registros que desea ver:

	Todos los participantes	Hoy, 1 de julio de 2013	
Todas las actividades	Todas las acciones	Mostrar en página	Conseguir estos registros

Figura 4.16: Filtros para la obtención de informes en Moodle

personales, aun estando muchos de ellos disponibles en Moodle. No son del ámbito del DSL desarrollado, pues como se especificó con anterioridad, la recogida de datos es meramente con fines cuantitativos. Por esto, el contenido de mensajes en correos, en foros, datos geográficos y toda aquella información que pueda relacionar a un usuario queda fuera del ámbito del proyecto.

## 4.5. Estudio de alternativas tecnológicas

La alternativa tecnológica a EvalCourse está en el propio Moodle. Tal y cómo se indicó en los requisitos no funcionales, una de las características más importantes es la usabilidad. EvalCourse va a ser utilizado por docentes no tecnólogos. Éstos han de poder usar la herramienta con facilidad. Moodle ofrece mecanismos para obtener gran parte de la información que aquí se detalla. Si vamos a la zona lateral del campus, veremos un enlace que es informes. Si pinchamos ahí, iremos a la pantalla 4.16 de los informes. Podremos aplicar una serie de filtros y obtener información sobre el acceso de alumnos a curso, actividades, foros, documentos,... etc. Además, se puede ver desde dónde se ha hecho.

Consideramos que esta alternativa no es óptima por varios motivos:

- El generador de informes es difícil de encontrar si no lo utilizas a menudo.
- Su interfaz es muy poco intuitiva.
- Los informes son poco personalizables y contienen información de poca utilidad.



## Capítulo 5

# Diseño del Sistema

En este capítulo se recoge la arquitectura general del sistema de información y el diseño de componentes software.

### 5.1. Diseño de la arquitectura

El diseño de la arquitectura se basa en diversos componentes del software Eclipse destinados al desarrollo dirigido por modelos. Tiene las etapas que se pueden ver en la figura 5.1 y que se detallarán más adelante.

### 5.2. Arquitectura física

Los componentes hardware 5.2 que forman la arquitectura física de nuestro sistema son básicamente dos:

- Base de datos de Moodle: La base de datos puede estar en el propio ordenador local del usuario o en un servidor independiente.
- Ordenador del usuario: El equipo del profesor que va a utilizar la aplicación.

### 5.3. Arquitectura lógica

La arquitectura lógica de EvalCourse, con carácter más general, es la descrita en la figura 5.3. En ella se aprecia en un diagrama de interacción los componentes que intervienen y que son necesarios para cumplir todos los requisitos de la aplicación. Estos elementos software son los que se describen en la siguiente lista:

- Eclipse: es el programa base sobre el que se instalará la aplicación.
- EvalCourse: es el software desarrollado, que se distribuye como un plugin de Eclipse.
- Moodle: sistema web de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea. Este tipo de plataformas tecnológicas también se conoce como LMS (Learning Management System).

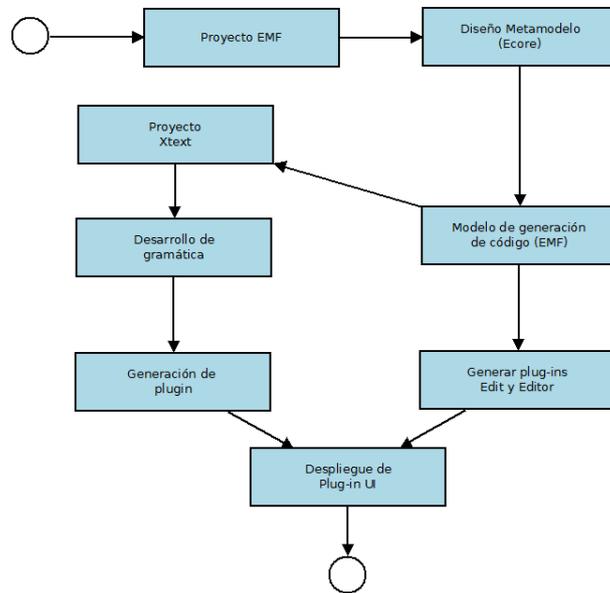


Figura 5.1: Arquitectura software. Proceso de desarrollo

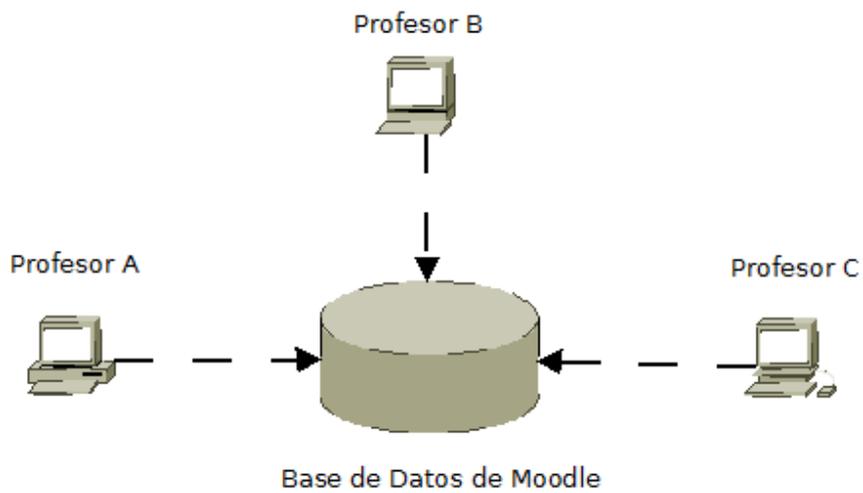


Figura 5.2: Diagrama de la arquitectura física

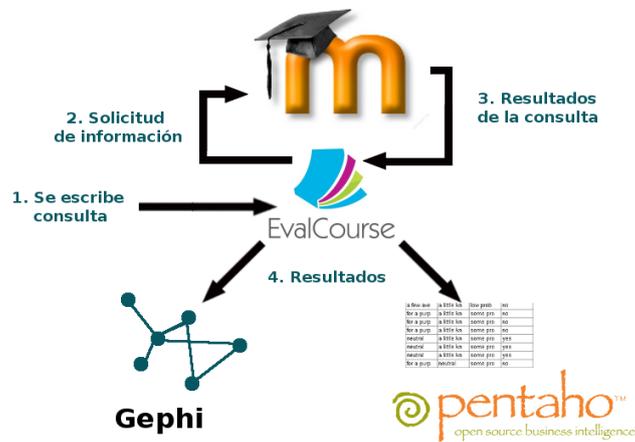


Figura 5.3: Diagrama de la arquitectura lógica

- Pentaho: conjunto de programas libres para generar inteligencia empresarial que incluye herramientas integradas para generar informes, minería de datos, ETL, etc.
- Gephi: Gephi es un programa para visualización interactiva y exploración de todo tipo de redes y sistemas complejos, dinámicos y gráficos jerárquicos.

Además de los componentes descritos anteriormente. EvalCourse necesita de otra serie de componentes para su desarrollo. En los siguientes puntos se hablará de ellos. Nos referimos a EMF, Ecore, Xtext y Xtend (figura 5.4). Además describiremos el software Pentaho, ya que hay que incluir en el cómo serán los ficheros de transformación que generará el DSL.

### 5.3.1. EMF: Eclipse Modeling Framework

EMF [Eclipse, a] es un framework de modelado y facilidad de generación de código para construir herramientas y otras aplicaciones basadas en un modelo de datos estructurado. Desde una especificación del modelo descrita en XMI, EMF suministra herramientas y soporte para producir un conjunto de clases Java para el modelo, así como un conjunto de clases que permitan visualización y edición basándose en comandos del modelo, y un editor básico. Los Modelos pueden ser especificados usando Anotación Java, documentos XML, o herramientas de modelado, y después ser importados a EMF. Lo más importante de todo, EMF suministra las bases para la interoperabilidad con otras herramientas y aplicaciones basadas en EMF [Steinberg et al., 2009]. Sus características principales son las siguientes:

- Framework para el desarrollo de metamodelos (sintaxis abstractas).
- Permite generar automáticamente clases de implementación en Java para los elementos de nuestros metamodelos



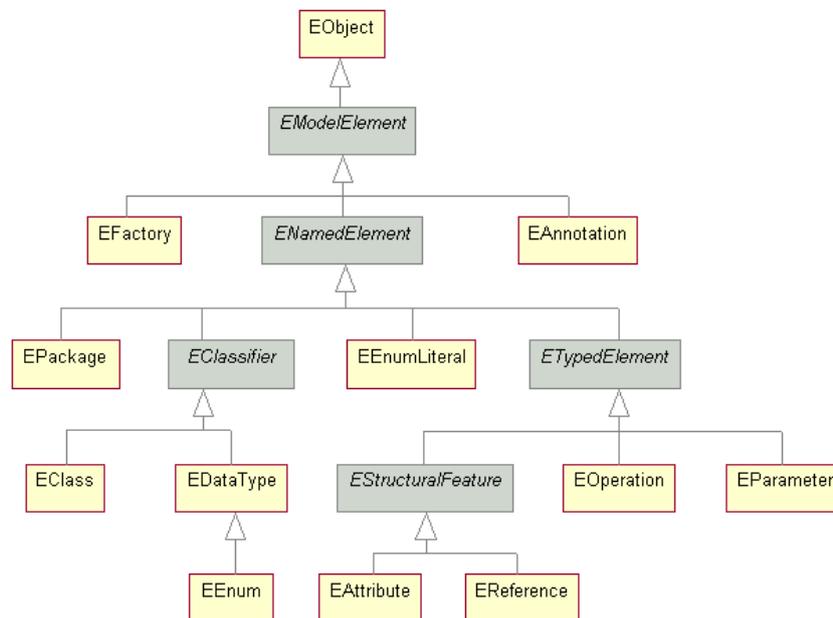


Figura 5.6: Jerarquía de componentes Ecore

- EPackage: componente que permite organizar clases y tipos de datos.
- EClass: conceptos en el metamodelo
- EReference: asociación entre conceptos
- EAttribute: propiedades de los conceptos
- EDataType: tipo de un atributo.

### 5.3.3. Xtext

Xtext es un framework para el desarrollo de lenguajes de programación y lenguajes específicos de dominio (DSL) [Bettini, ], liderados por la empresa Itemis AG.

- Xtext genera automáticamente los componentes necesarios para trabajar con los DSL: parser, analizador estático, formateador de código, generador de código, etc.
- Pueden utilizarse desde dentro de Eclipse o de forma independiente.
- Xtext incluye un lenguaje para diseñar gramáticas y una amplia API para definir los diferentes aspectos de los DSL.
- Utiliza Google Guice, un framework ligero para la inyección de dependencias, con el cual podemos sustituir fácilmente cualquier comportamiento por defecto.
- Se basa en ANTLR para generar los parsers LL(k).

- Xtext construye automáticamente la infraestructura software necesaria para dar soporte a los nuevos lenguajes.
- Para configurar la generación automática del código de soporte, Xtext utiliza un DSL especial, denominado MWE2.
- MWE2 permite componer objetos en Java, pero de forma declarativa y sin tanta verbosidad.

Abarca todos los aspectos de una infraestructura completa del lenguaje, desde los analizadores, enlazador, compilador o intérprete. Finalmente se integra con el IDE de Eclipse. Algunas de sus características son:

- Coloreado de sintaxis
- Autocompletar
- Integración con Java
- Integración con otras herramientas de Eclipse
- Validación y corrección rápida

#### 5.3.4. Xtend

Xtend es un lenguaje de programación de alto nivel de propósito general para la máquina virtual de Java. Xtend tiene sintáctica y semánticamente sus raíces en el lenguaje de programación Java, pero se centra en una sintaxis más concisa y alguna funcionalidad adicional, como la inferencia de tipos, métodos de extensión, y la sobrecarga de operadores. Se podría decir que Xtend es un dialecto flexible y expresivo de Java, de hecho, así lo pone en su página web [\[Eclipse, c\]](#). Además de ser un lenguaje orientado a objetos, también integra características conocidas de la programación funcional, por ejemplo, expresiones lambda. Xtend utiliza sistema de tipos de Java. Se integra perfectamente con todas las bibliotecas de Java existentes.

#### 5.3.5. Transformaciones a Pentaho

Para poder realizar el proceso de transformación de las consultas a ficheros de Pentaho hay que conocer los componentes de que consta éste. Todos los pasos de los casos de uso descritos en el capítulo anterior, tienen una etapa en la que se construye la transformación Pentaho. En este apartado vamos a describir por un lado, los pasos de Pentaho que se van a utilizar; y después mostraremos cómo deberá ser cada fichero de transformación según la consulta.

#### Componentes de Pentaho

El primero componente que tendrá cualquiera de las transformaciones es un *Table input*. En él se define la consulta SQL que extraerá la información de la base de datos de Moodle. Este componente es de tipo *input*.



Table input

Otro componente que tendrán todas las transformaciones será el de salida (de tipo output). Como se ha mencionado, la salida serán en formato xml, por lo que el componente es *XML output*.



XML Output

*Group by* te permite calcular valores sobre un grupo definido de campos. Funcionamiento similar a la cláusula group by de una consulta SQL. Este componente es de tipo *Statistics*.



Group by

El paso calculadora (*calculator*) proporciona funciones predefinidas que se pueden ejecutar en los valores de campo de entrada. Además de los argumentos (campos A, B y C) también se debe especificar el tipo de retorno de la función.



Calculator

El paso *Join Rows (cartesian product)* realiza un producto cartesiano entre los datos provenientes de varias entradas.



Join Rows (cartesian product)

*Load file in content memory* carga el contenido de un fichero cualquiera en memoria.



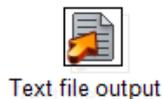
Load file content in memory

*XSL Transformation* es un componente que se encarga de realizar una transformación XSL. Recibe dos ficheros, por un lado el XML a transformar y por otra el XSL que tiene el esquema de transformación.



XSL Transformation

*Text file output* es un componente de salida (output). Su finalidad es devolver como salida la información de la transformación en un fichero de texto.



## Componentes necesarios para cada transformación Pentaho

A continuación se describen los componentes necesarios para la generación de cada fichero de transformación Pentaho.

### Transformaciones en Pentaho para actividades

- *input table*: Realiza consulta a la base de datos para obtener la información de las actividades
- *group by*: Obtiene dos agrupamientos en base a la consulta anterior. Por un lado calcula cuántas entregas han hecho los alumnos y por otro lado cuántas han sido a tiempo.
- *group by*: Obtiene el máximo número de entregas que había que hacer
- *Join Rows*: Se realiza un producto cartesiano entre el campo máximo y todos los registros de la consulta
- *calculator*: Calcula la cantidad de actividades pendientes de entrega y el porcentaje que supone del total
- *XML Output*: Devuelve la información en fichero XML

### Transformaciones en Pentaho para talleres (workshop)

- *input table*: Realiza consulta a la base de datos para obtener la información de los talleres. Qué alumno corrige a cada estudiante
- *XML Output*: Devuelve la información en fichero XML

### Transformaciones en Pentaho para foros

- *input table*: Realiza consulta a la base de datos para obtener la información de los foros
- *group by*: Suma por una lado las participaciones de los alumnos en los foros y por otro la cantidad de debates que han iniciado
- *calculator*: Calcula el total de intervenciones sumando las participaciones y los inicios de debates
- *XML Output*: Devuelve la información en fichero XML

### Transformaciones en Pentaho para campus

- *input table*: Realiza la consulta de la base de datos para obtener información del log del campus
- *group by*: Suma la cantidad de apariciones por estudiante
- *XML Output*: Devuelve la información en fichero XML

### Transformaciones para obtener gráfico

- *Load file in content memory*: Carga en memoria el xml con los registros
- *Load file in content memory*: Carga en memoria del xsl con el esquema de transformación
- *Join Rows*: Unimos el contenido de ambos ficheros en un único registro con un atributo para el contenido de cada fichero
- *XSL Transformation*: Realizamos la transformación XSL sobre los registros contenido en el fichero XML
- *Text file output*: Asignamos el fichero transformado a un fichero de texto con extensión gexf (Formato de Gephi)

### Mapeo

A continuación se describe el mapeo que se realiza entre el código EvalCourse y el xml de Pentaho. En el código 5.1 se puede ver las diferentes alternativas que ofrece el DSL.

```

1 Evidence name_of_the_indicator:
2     get students
3     show milestones | participation | access
4     in assignment | forum | campus [list of ids].

```

Listing 5.1: Formato y palabras reservadas

La estructura del XML de Pentaho es la que se puede ver en la figura 5.2.

```

1 <transformation>
2     <info>
3         [...]
4     </info>
5     <connection>
6         [...]
7     </connection>
8     <order>
9         [...]
10    </order>
11    <step>
12        [...]
13    </step>
14    <step>
15        [...]
16    </step>
17    <step>
18        [...]
19    </step>
20 </transformation>

```

---

## Listing 5.2: Estructura XML Pentaho

Explicación del XML:

- `transformation`: elemento raíz que contiene el cuerpo del documento XML de Pentaho.
- `info`: elemento que contiene información general de la transformación, entre ella el nombre de la misma.
- `connection`: elemento que contiene la información de conexión, si la hubiere, a la base de datos. Tomará sus datos del fichero de conexión que deberemos situar en la carpeta raíz de Eclipse.
- `order`: elemento en el que se indican los saltos (hops) o enlaces entre etapas (steps) de que constará la transformación.
- `step`: elemento que engloba las características de cada etapa (step) y que se corresponderá a algún componente Pentaho de los que comentamos en el subapartado anterior.

### Mapeo de entrega de hitos:

Los ficheros XML son bastantes largos, por lo que vamos a poner en el apéndice al completo el correspondiente a esta transformación. Vamos a mostrar en el cuerpo de este mapeo (código 1) los elementos de etapas (steps) como los de transición (hops) y para el resto de mapeos sólo mosteremos los steps. El código que generará el fichero de transformación para la entrega de hitos es el que se ve en el código 5.3.

```
1 Evidence indicador_hitos :
2   get students
3   show milestones
4   in assignment.
```

Listing 5.3: Consulta para la entrega de hitos

```
1 <transformation>
2   [...]
3 <order>
4     <hop>
5       <from>query </from>
6       <to>milestones </to>
7       <enabled>Y</enabled>
8     </hop>
9     <hop>
10      <from>percentage </from>
11      <to>indicador_hitos </to>
12      <enabled>Y</enabled>
13    </hop>
14    <hop>
15      <from>milestones </from>
16      <to>max </to>
17      <enabled>Y</enabled>
18    </hop>
19    <hop>
20      <from>milestones </from>
```

```

21         <to>join </to>
22         <enabled>Y</enabled>
23     </hop>
24     <hop>
25         <from>max</from>
26         <to>join </to>
27         <enabled>Y</enabled>
28     </hop>
29     <hop>
30         <from>join </from>
31         <to>percentage </to>
32         <enabled>Y</enabled>
33     </hop>
34
35 </order>
36 <step>
37     <name>query </name>
38     <type>TableInput </type>
39     [...]
40 </step>
41 <step>
42     <name>milestones </name>
43     <type>GroupBy</type>
44     [...]
45 </step>
46 <step>
47     <name>max</name>
48     <type>GroupBy</type>
49     [...]
50 </step>
51 <step>
52     <name>join </name>
53     <type>JoinRows</type>
54     [...]
55 </step>
56 <step>
57     <name>percentage </name>
58     <type>Calculator </type>
59     [...]
60 </step>
61 <step>
62     <name>indicador_hitos </name>
63     <type>XMLOutput</type>
64     [...]
65 </step>
66 </transformation>

```

Listing 5.4: Estructura steps de XML Pentaho para entrega de hitos

textbfMapeo de evaluación de workshop:

El código que generará el fichero de transformación para la evaluación de los workshops es el que se ve en el código 5.5. Las etapas de la transformación se pueden ver en el código 5.6.

```

1 Evidence indicador_workshop :
2     get students
3     show assessment
4     in workshop.

```

Listing 5.5: Consulta para el trabajo en los workshops

```
1 <transformation>
2 [...]
3 <step>
4     <name>query </name>
5     <type>TableInput </type>
6     [...]
7 </step>
8 <step>
9     <name>indicador_workshop </name>
10    <type>XMLOutput </type>
11    [...]
12 </step>
13
14 </transformation>
```

Listing 5.6: Estructura steps XML Pentaho para las actividades del workshop

textbfMapeo de participación en foros:

El código que generará el fichero de transformación para la participación en foros es el que se ve en el código 5.7. Las etapas de la transformación se pueden ver en el código 5.8.

```
1 Evidence indicador_foro:
2     get students
3     show participation
4     in forum.
```

Listing 5.7: Consulta para la participación en foros

```
1 <transformation>
2 [...]
3 <step>
4     <name>query </name>
5     <type>TableInput </type>
6     [...]
7 </step>
8 <step>
9     <name>participation </name>
10    <type>GroupBy </type>
11    [...]
12 </step>
13 <step>
14    <name>total </name>
15    <type>Calculator </type>
16    [...]
17 </step>
18 <step>
19    <name>indicador_foro </name>
20    <type>XMLOutput </type>
21    [...]
22 </step>
23
24 </transformation>
```

Listing 5.8: Estructura steps XML Pentaho para la participación en el foro

textbfMapeo de interacción con el campus:

El código que generará el fichero de transformación para la interacción con el campus es el que se ve en el código 5.9. Las etapas de la transformación se pueden ver en el código 5.10.

```
1 Evidence indicador_campus :
2     get_students
3     show_access
4     in_campus.
```

Listing 5.9: Consulta para el acceso al campus

```
1 <transformation>
2   [...]
3 <step>
4     <name>query </name>
5     <type>TableInput </type>
6     [...]
7 </step>
8 <step>
9     <name>access </name>
10    <type>GroupBy </type>
11    [...]
12 </step>
13 <step>
14    <name>indicador_campus </name>
15    <type>XMLOutput </type>
16    [...]
17 </step>
18 </transformation>
```

Listing 5.10: Estructura steps XML Pentaho para acceso al campus

## 5.4. Arquitectura de diseño

El patrón arquitectónico que sigue este proyecto es el patrón pipeline. El patrón pipeline consiste en una serie de transformaciones sobre sucesivas etapas de los datos de entrada. Los datos entran al sistema y fluyen a través de los componentes. Una tubería (pipeline) es una estructura

común que conecta componentes computacionales (filtros) a través de conectores (pipes), de modo que las computaciones se ejecutan a la manera de un flujo. Los datos se transportan a través de las tuberías entre los filtros, transformando gradualmente las entradas en salidas. Es una arquitectura muy intuitiva, debido a su simplicidad y su facilidad para captar una funcionalidad. Al introducirse la consulta EvalCourse en el sistema, esta va pasando por las diferentes etapas

del compilador hasta generarse finalmente el código en XML que podremos ver en Pentaho 5.7.

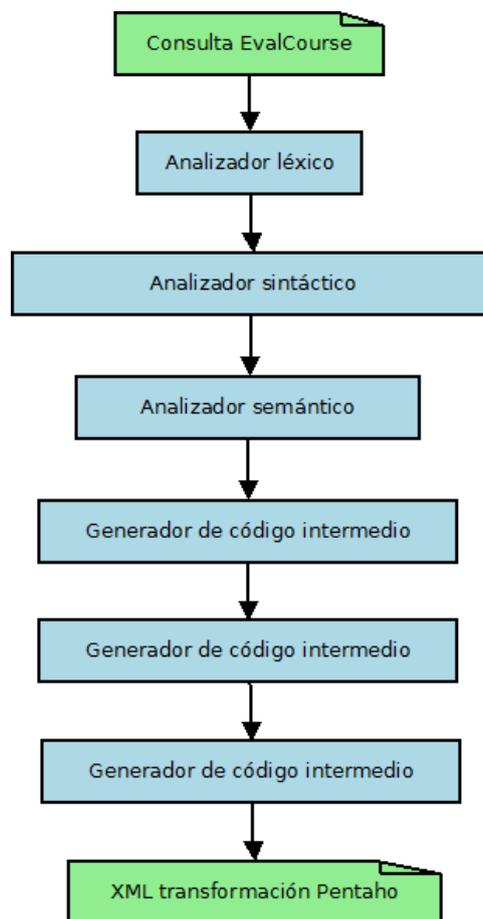


Figura 5.7: Diagrama de la arquitectura de diseño

## Capítulo 6

# Implementación del Sistema

### 6.1. Introducción

La implementación de un DSL tiene básicamente tres pasos:

- Por un lado, tenemos la creación del **modelo**. Como se ha mencionado varias veces en este documento, nuestro DSL será útil en un determinado dominio. Lo primero es modelar dicho dominio, crear el metamodelo que caracterice a todas las instancias de dicho dominio. Este es el primer paso que debemos hacer, ya que queremos modelar un lenguaje para un dominio para el que no existe. Si tuviéramos el modelo, podríamos omitir este paso.
- En segundo lugar, queremos crear el **lenguaje de dominio específico (DSL)**. Crearemos un lenguaje que se base en el metamodelo creado, definiendo su sintaxis y el modo en el que interactuarán los operadores y tokens que participan.
- Y por último habrá que definir las diferentes **operaciones** que se llevarán a cabo a partir de la utilización de nuestro lenguaje. Como y en qué se transformará el código que introduzca un usuario utilizando la sintaxis que definimos en el apartado anterior.

### 6.2. Modelo

Para crear el modelo se siguieron los siguientes pasos:

- Creación de proyecto EMF
- Creación de metamodelo mediante Ecore
- Generación del código Java

### 6.3. Proyecto EMF

Para la creación del metamodelo que vamos a utilizar en nuestro DSL, será necesario crear un proyecto EMF vacío 6.1.

Ahora debemos crear las clases Java de nuestro modelo. Pero como hemos dicho, para el modelo vamos a utilizar Ecore, y del conjunto de herramientas que conforman el componente Ecore,

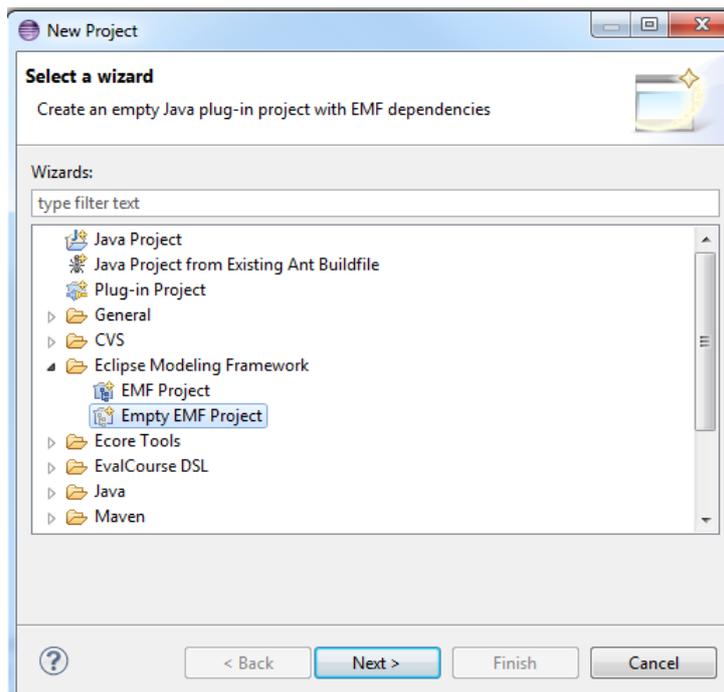


Figura 6.1: Creación proyecto EMF

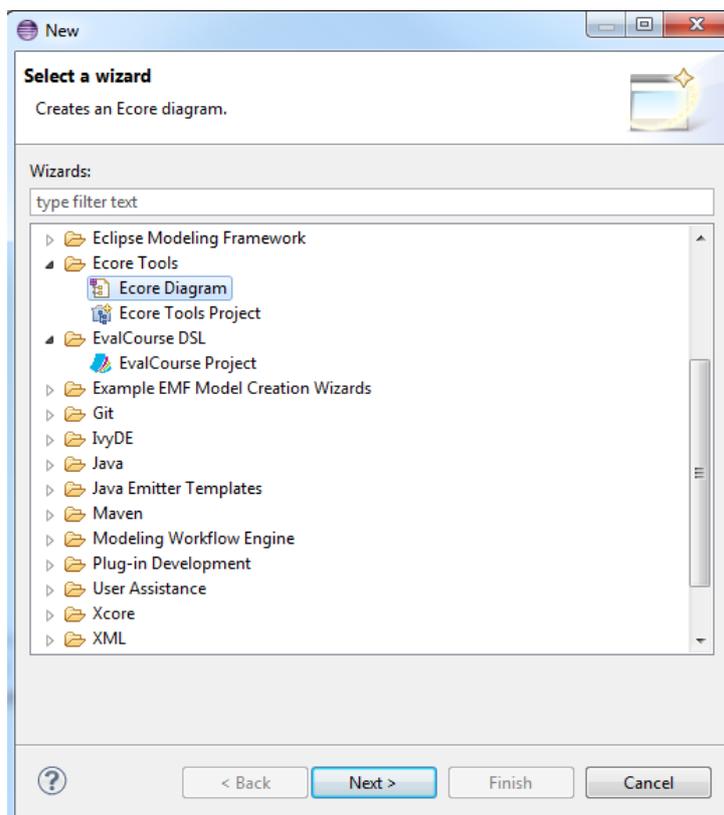


Figura 6.2: Creación diagrama en Ecore

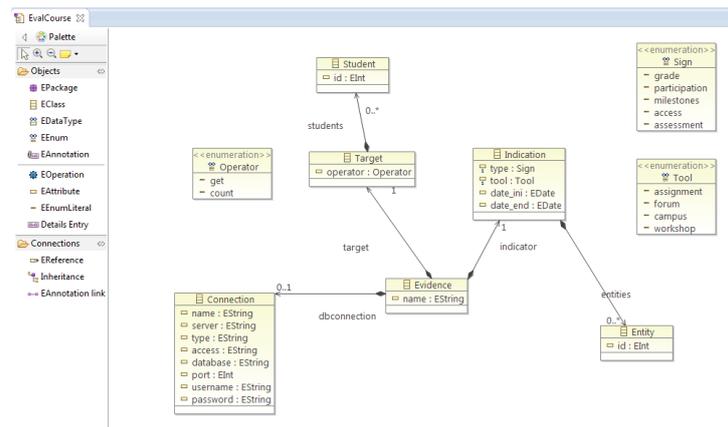


Figura 6.3: Vista gráfica del metamodelo

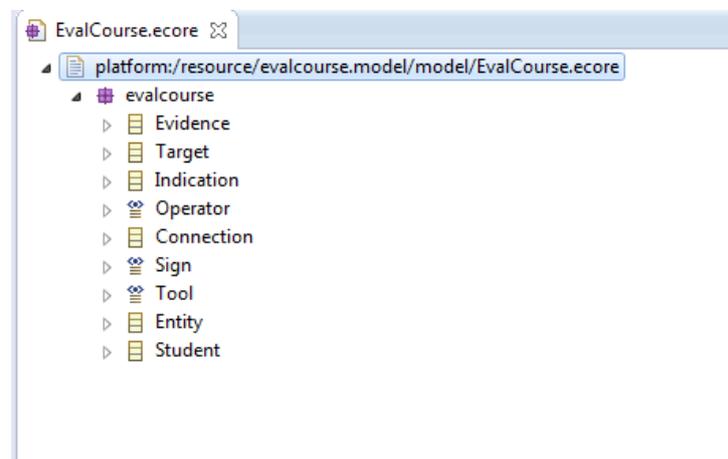


Figura 6.4: Archivo del metamodelo Ecore

utilizaremos *Ecore diagram* 6.2. Dentro crearemos una carpeta *model* que será el lugar en el que se guardará el metamodelo.

El modelo lo creamos mediante un fichero de extensión XMI. El principal objetivo de XMI es permitir un intercambio de metainformación entre herramientas de modelado, basados en UML, y repositorios de metainformación, basados en MOF, en heterogéneos entornos distribuidos. Realizando este intercambio mediante streams o ficheros con formato estándar basado en XML. Gracias al componente Ecore no tendremos que utilizar el editor para escribir en XMI. El metamodelo que crearemos usando *Ecore diagram* es el que aparece en la figura 6.3.

Conforme vamos trabajando y guardando, automáticamente se actualiza el fichero Ecore que contiene realmente el metamodelo (figura 6.4). Si abrimos este fichero, podremos comprobar cómo se listan los conceptos que hemos representado en el metamodelo.

Detrás de esta representación del fichero, como se comentó anteriormente, está la implementación en XMI (Ver en apéndice 2).

Para poder utilizar el modelo necesitamos las clases Java, y EMF nos proporciona una herramienta para generar las clases Java a partir de nuestro modelo Ecore: *EMF Generator Model*. Creamos un fichero generador basado en nuestro metamodelo Ecore 6.5.

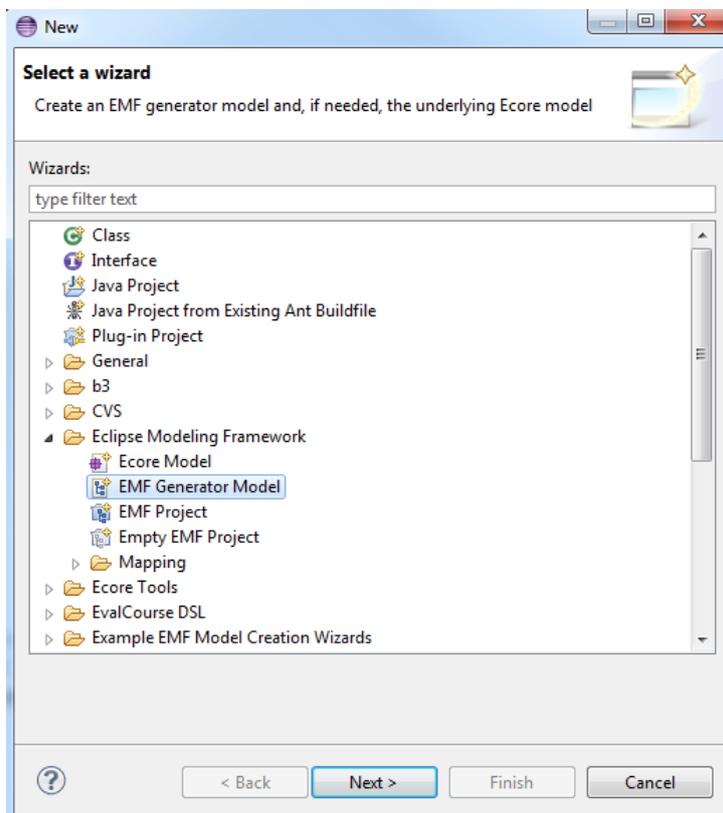


Figura 6.5: Creación fichero de generación de modelo

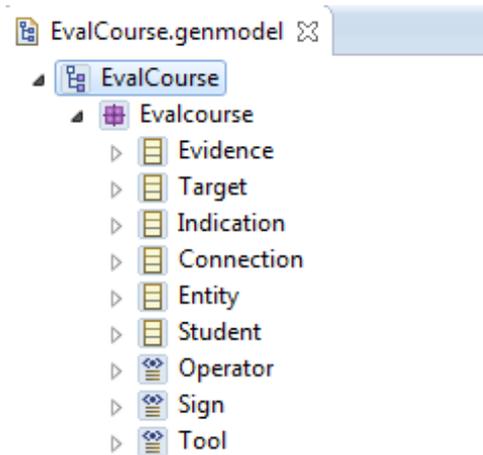


Figura 6.6: Generador del metamodelo

Para generar el código Java habrá que abrir el fichero `EvalCourse.genmodel` 6.6, pinchar en la clase principal con el botón derecho y darle *Generate all*. Tras unos segundos se habrán generado todas las clases Java.

Finalmente, nuestro directorio *model* con todas las carpetas necesarias para la creación del metamodelo quedó como se puede ver en la figura 6.7.

## 6.4. DSL

En esta fase generaremos el lenguaje. Las etapas de las que se compone esta fase son dos:

- Creación de nuevo proyecto Xtext basado en el metamodelo Ecore
- Generación de la infraestructura del lenguaje

Debemos crear un proyecto Xtext basado en un modelo definido en Ecore 6.8. Al hacerlo así, EMF nos creará una sintaxis del lenguaje por defecto, a partir de la cual podremos empezar a trabajar y modelar la sintaxis según nuestro planteamiento.

La sintaxis de nuestro lenguaje debe ser la mostrada en 8.1. La explicación de la sintaxis es la siguiente:

- En la primera línea se declara el nombre de la evidencia o indicador, terminado en dos puntos para indicar que desde ahí comienza la declaración: *Evidence name\_of\_the\_indicator:*
- En la segunda línea lo que se escribe es una cadena constante. No interviene en el resultado, pero de cara a que la consulta resulte más clara se decidió añadir: *get students*
- En la tercera línea se define qué operador queremos que nos devuelva la evidencia. Puede ser la entrega de hitos (milestones), la participación (participation) o el acceso (access) *show milestones / workshop / participation / access*

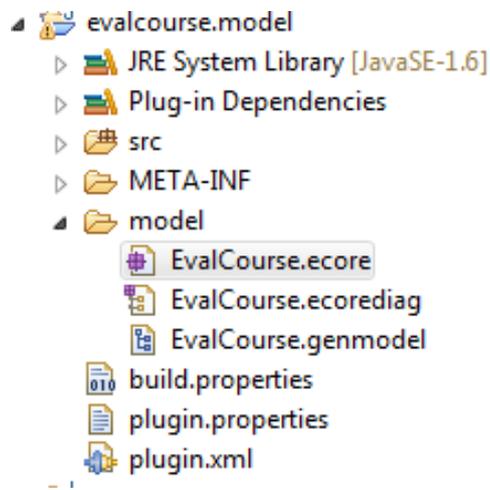


Figura 6.7: Proyecto del metamodelo

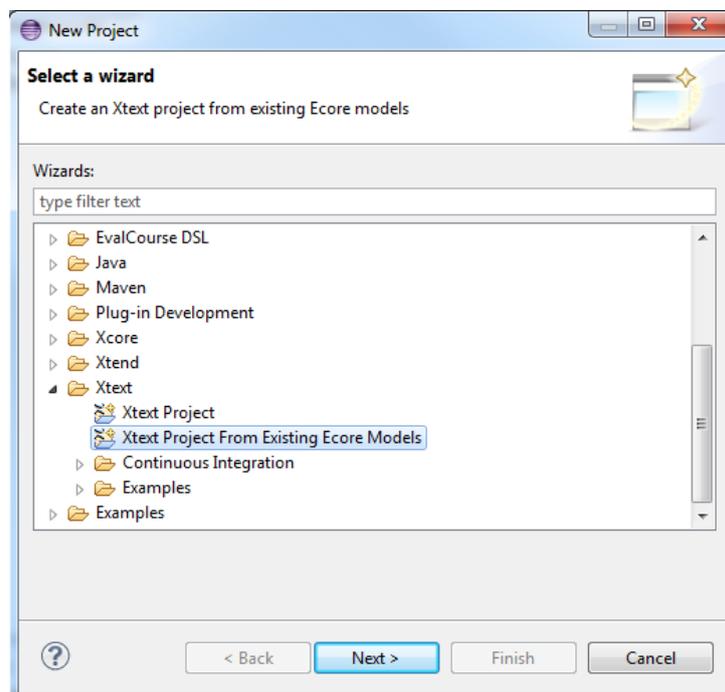


Figura 6.8: Creación de proyecto Xtext basado en Ecore

- En la cuarta línea se indica sobre qué concepto actuará el operador anterior. Puede ser sobre las tareas (assignment), talleres (workshop), foros (forum) o plataforma virtual (campus). Además si se indican a continuación uno o varios identificadores, se actuará sobre el concepto con dicha o dichas identidades. Si no se especifica se actuará sobre todas las del curso. La finalización de la consulta se indica con un punto *in assignment | workshop | forum | campus [list of ids]*.

```

1 Evidence name_of_the_indicator :
2     get students
3     show milestones | participation | access
4     in assignment | workshop | forum | campus [list of ids].

```

Listing 6.1: Sintaxis del lenguaje y palabras reservadas

Debemos crear un fichero en formato Xtext, en el caso de nuestro proyecto el fichero se denomina *EvalCourse.xtext* |refcode:xtext en el que se define nuestro DSL.

```

1 // automatically generated by Xtext
2 grammar
3     evalcourse.dsl.EvalCourse with org.eclipse.xtext.common.Terminals
4
5 import "http://evalcourse/1.0"
6 import "http://www.eclipse.org/emf/2002/Ecore" as ecore
7
8
9 Evidence returns Evidence :
10     'Evidence'
11     name=EString
12     ' ';
13         target=Target 'students'
14         indicator=Indication
15         ('dbconnection' dbconnection=Connection)?
16     ' . ' ;
17
18 Entity returns Entity :
19     {Entity}
20     id=EInt ;
21
22 Target returns Target :
23     {Target}
24     (operator=Operator)?
25     (students+=Student ("," students+=Student)*)?
26     ;
27
28 Student returns Student :
29     {Student}
30     id=EInt
31     ;
32
33 EString returns ecore::EString :
34     STRING | ID ;
35
36 Indication returns Indication :
37     {Indication}
38     'show' type=Sign

```

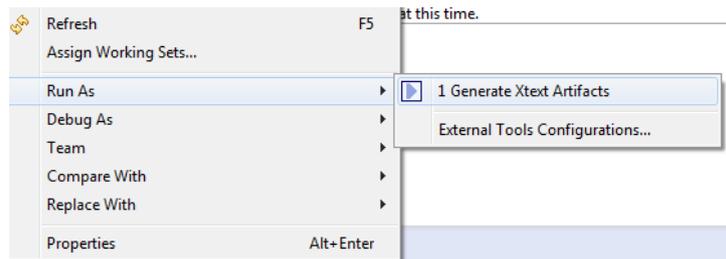


Figura 6.9: Generación infraestructura DSL

```

39         'in' tool=Tool
40         ('between' date_ini=EDate 'and' date_end=EDate)?
41         (entities+=Entity ("," entities+=Entity)*)?
42     ;
43
44 Connection returns Connection:
45     {Connection}
46     'Connection'
47     name=EString
48     '{'
49         ('server' server=EString)?
50         ('type' type=EString)?
51         ('access' access=EString)?
52         ('database' database=EString)?
53         ('port' port=EInt)?
54         ('username' username=EString)?
55         ('password' password=EString)?
56     '}' ;
57
58 Operator returns Operator:
59     'get' | 'count' ;
60
61 EInt returns ecore::EInt:
62     '-'? INT;
63
64 EDate returns ecore::EDate:
65     INT'-'INT'-'INT'
66 ;
67
68 Tool returns Tool:
69     'assignment' | 'forum' | 'campus' | 'workshop';
70
71 Sign returns Sign:
72     'grade' | 'participation' | 'milestones' | 'access' | 'assessment';

```

Listing 6.2: EvalCourse.xtext

Una vez terminado el fichero Xtext, generaremos la infraestructura del lenguaje, pinchando con el botón derecho sobre el fichero *EvalCourse.xtext* (Run As -> Generate Xtext Artifacts) 6.9. Se generan ficheros Java y una infraestructura de lenguaje ejecutable con un editor de Eclipse en el que utilizar nuestro lenguaje.

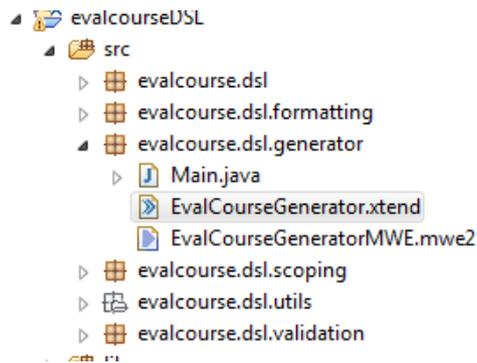


Figura 6.10: Generación infraestructura DSL

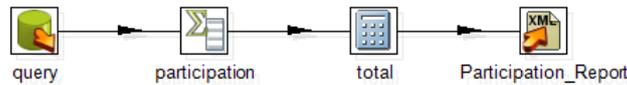


Figura 6.11: Transformación en Pentaho

## 6.5. Traducción

En la generación de la infraestructura a partir del fichero Xtext se generó, entre otros, un paquete llamado *EvalCourse.dsl.generator*, que contiene un fichero llamado *EvalCourse.xtend* 6.10, que será en el que deberemos comenzar a programar la transformación ya que es el fichero principal. A partir de este podremos crear otros paquetes y ficheros en el que programar diferentes partes de nuestro DSL.

Lo que queremos hacer aquí es transformar una consulta del tipo que vimos anteriormente en fichero de transformación Pentaho. Cuando abrimos una transformación en Pentaho tiene el aspecto que podemos ver en la figura 6.11.

Una transformación en Pentaho se compone de varios componentes 6.12. Deberemos mapear estos componentes desde nuestro lenguaje. Estos componentes son los siguientes:

- transformation: componente principal que inicia y finaliza una transformación
- connection: componente que se encarga de la conexión con la base de datos
- step: cada uno de los pasos de los que se compone la transformación
- hop: cada uno de los saltos que va de un *step* al siguiente

Para implementar todos estos componentes se crean varias clases dentro de un paquete especial llamado *utils* 6.13. Estas clases son las siguientes:

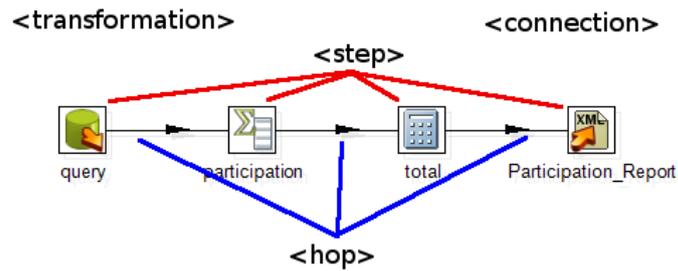


Figura 6.12: Componentes de la transformación en Pentaho

- *Classpath.xtend*: Clase para incluir las bibliotecas necesarias para ejecutar transformaciones de Pentaho en el proyecto EvalCourse que cree el usuario
- *DBconnection.xtend*: Clase para incluir en la transformación de Pentaho el componente de conexión con la base de datos
- *gephi.xtend*: Clase para generar el fichero de transformación en Pentaho para generar el XML del fichero gráfico Gephi a partir de un fichero de transformación XSLT
- *Hop.xtend*: Clase para generar los saltos entre las diferentes etapas del fichero de transformación en Pentaho
- *Step.xtend*: Clase para generar las etapas de que constará la transformación en Pentaho
- *Transformation.xtend*: Clase para generar los parámetros generales del fichero de transformación de Pentaho
- *Xml.xtend*: Clase para generar el resto de características por defecto del fichero de transformación de Pentaho

Una vez terminado el proyecto se exporta como plugin de Eclipse, para después poder ser utilizado como tal en cualquier instalación de Eclipse que cuenta con EMF 6.14.

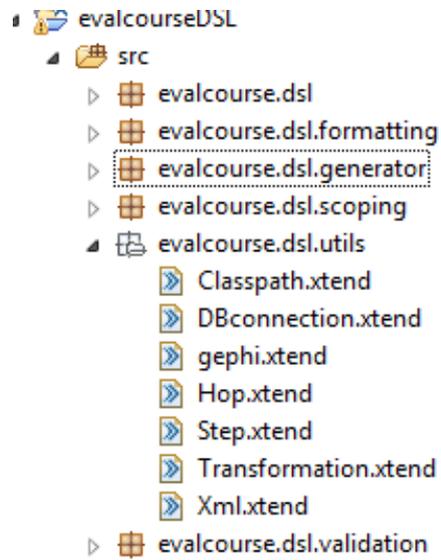


Figura 6.13: Clases que conforman el paquete utils

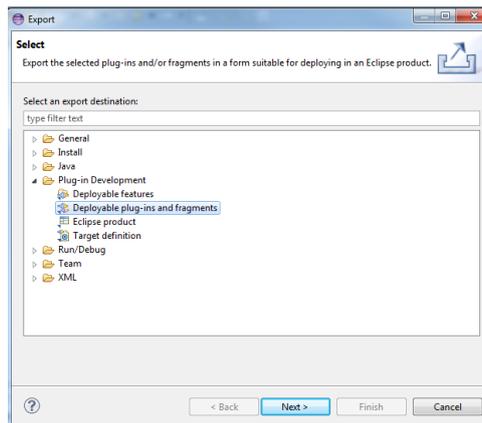


Figura 6.14: Creación del plugin del DSL



## Capítulo 7

# Pruebas del Sistema

### 7.1. Pruebas unitarias

Las pruebas unitarias tienen por objetivo localizar errores en cada nuevo artefacto software desarrollado, antes de que se produzca la integración con el resto de artefactos del sistema.

#### 7.1.1. Metamodelo

Una vez creado el metamodelo se generan automáticamente utilizando el asistente de Eclipse varios modelos conforme a dicho Metamodelo. Todos fueron validados utilizando el validador de Eclipse, por lo que el metamodelo superó las pruebas unitarias (figura 7.1).

#### 7.1.2. Editor

El editor generado por la herramienta para EvalCourse indica cuando hay errores en la sintaxis. Todas las consultas conforme a la gramática que se han probado son admitidas por el DSL (tabla 7.1).

#### 7.1.3. Transformación

En este apartado se evalúan los ficheros de Pentaho obtenidos a partir de las consultas del programa, para comprobar si realmente devuelven el resultado esperado (tabla 7.2). Todos los

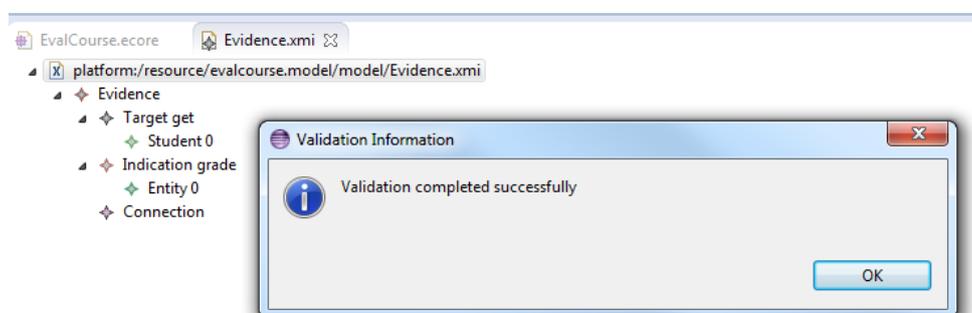


Figura 7.1: Validación de un modelo conforme al metamodelo de EvalCourse.

Cuadro 7.1: Listado de pruebas unitarias de código.

Consulta	Verificada
Evidence <i>name</i> : get students show milestones in assignments.	✓
Evidence <i>name</i> : get students show assessment in workshop.	✓
Evidence <i>name</i> : get students show participation in forum.	✓
Evidence <i>name</i> : get students show access in campus.	✓
Evidence <i>name</i> : get students show participation in campus.	✓
Evidence <i>name</i> : get students show access in forum.	✓

Cuadro 7.2: Listado de pruebas unitarias de transformaciones.

Transformación	Funcionamiento
Obtención de entrega de hitos	✓
Evaluación por pares en workshops.	✓
Participación en los foros.	✓
Interacción con el campus.	✓
Acceso a los foros.	✓

ficheros generados funcionan en Pentaho, luego se han pasado las pruebas unitarias para este módulo.

#### 7.1.4. Resultados en xml

En este apartado se evalúan los ficheros xml obtenidos a partir la ejecución de la transformación Pentaho (tabla 7.3). Todas las pruebas generan unos resultados que se han verificado conforme a los datos reales del campus, luego se han pasado las pruebas unitarias para este módulo.

#### 7.1.5. Resultados en gráfico

En este apartado se evalúan los ficheros gráficos obtenidos a partir la ejecución de la transformación XSL del resultado XML (tabla 7.4). Todas las pruebas generan unos resultados que se han verificado conforme a los datos reales del campus, luego se han pasado las pruebas unitarias para este módulo.

Cuadro 7.3: Listado de resultados de ficheros xml.

Listados XML	Funcionamiento
Listado de entrega de hitos	✓
Listado de la evaluación por pares en workshops.	✓
Listado de la participación en los foros.	✓
Listado de la interacción con el campus.	✓
Listado de acceso a los foros.	✓

Cuadro 7.4: Listado de resultados de ficheros xml.

Gráfico	Funcionamiento
Gráfico de entrega de hitos	✓
Gráfico de la evaluación por pares en workshops.	✓
Gráfico de la participación en los foros.	✓
Gráfico de la interacción con el campus.	✓
Gráfico de acceso a los foros.	✓

Cuadro 7.5: Pruebas de integración de los módulos.

Módulos	Integración
Metamodelo - Editor (Xtext)	✓
Metamodelo - Editor (Xtext) - Transformación (Xtend)	✓
Transformación (Xtend) - Resultados en xml	✓
Resultados en xml - Resultados en gráfico	✓
Metamodelo - Editor - Transformación - Resultados xml - Resultados gráfico	✓

## 7.2. Pruebas de integración

Este tipo de pruebas tienen por objetivo localizar errores en módulos o subsistemas completos, analizando la interacción entre varios artefactos software. Los distintos módulos se fueron integrando hasta conseguir el funcionamiento del sistema completo. A continuación se muestra un cuadro (tabla 7.5) que muestra la verificación de la interacción entre varios de sus componentes, pero será en el siguiente apartado, en las pruebas de sistema, donde se vea el sistema funcionando al completo.

## 7.3. Pruebas de sistema

En esta actividad se realizan las pruebas del sistema de modo que se asegure que el sistema (tanto el software como el hardware) cumplan con todos los requisitos establecidos.

### 7.3.1. Pruebas funcionales

Para las pruebas funcionales se eligieron dos competencias a evaluar:

- Planificación y gestión del tiempo
- Solución de problemas

A continuación se describen los indicadores utilizados en cada uno de los casos y los resultados obtenidos. Los datos sobre los que se trabajan son unos datos de prueba generadores de manera aleatoria.

## Planificación y gestión del tiempo

La capacidad del alumno para gestionar su tiempo de una manera eficaz y productiva está directamente relacionado con su desempeño académico: *cuánto mejor gestiona su tiempo un estudiante, mejor nota obtiene*. Además, estará menos estresado no sólo en lo relacionado a su vida académica, sino también en lo que a lo personal se refiere [Britton and Tesser, 1991].

Los profesores suelen definir durante el curso diferentes actividades que los estudiantes tienen que realizar y subir al campus virtual antes de una fecha tope. Vamos a considerar cada una de estas entregas de actividades como unos si fueran unos hitos. Al final del curso mediremos la cantidad de actividades o hitos que los estudiantes han entregado a tiempo, y la cantidad de ellos que han sido entregados fuera de tiempo.

Una de las características de las actividades que definimos en el campus virtual es que se puede o no permitir el envío de tareas retrasadas. Para poder realizar el estudio, permitimos que los estudiantes entreguen sus tareas tardes si no han tenido tiempo. Las fechas topes hay que cumplirlas, pero el sistema permitirá su entrega aunque haya pasado la fecha.

En este ejemplo suponemos que los estudiantes han enviado varias tareas durante el curso y ahora vamos a evaluar su desempeño en la competencia de planificación y gestión del tiempo. Usando EvalCourse, vamos a consultar la información utilizando el código mostrado en el ejemplo 7.1. Declaramos una evidencia llamada *Milestone\_list*.

```
1 Evidence Milestone_list :
2     get_students
3     show_milestones
4     in_assignment.
```

Listing 7.1: Consulta para obtener el desempeño de los alumnos en la entrega de actividades

Este código nos devolverá el desempeño de los estudiantes en dos archivos. Por un lado tendremos un XML con la información mostrada en la tabla 7.6. En cada fila se muestra los hitos que tiene que completar cada estudiante (Total), la cantidad de estos que se han entregado a tiempo (In-time) y la cantidad de estos que no han sido entregados aún (Pending).

La primera fila del listado comienza con dos columnas identificando al estudiante con su identificador interno de Moodle y su nombre de usuario. La columna *Total* indica el número de actividades que los estudiantes tienen que subir durante el curso (cinco en nuestro caso). La cuarta columna, *In-time*, indica que *Student1* envió tres actividades antes de la fecha límite; *Overdue* indica que no ha enviado actividades fuera de tiempo, mientras que *Pending* indica que dos actividades no han sido aún entregadas. Finalmente, *In-rate* indica el porcentaje de actividades que se han entregado antes de la fecha límite con respecto al total que deberían haber entregado.

En el segundo fichero tenemos un gráfico que proporciona una visión complementaria de la información del desempeño de los estudiantes (figure 7.2). El vértice central del gráfico representa la actividad que estamos considerando, en este caso, la actividad es la entrega de las asignaciones de los estudiantes. Los otros vértices representan los estudiantes inscritos en el curso. Cada

Cuadro 7.6: Listado del desempeño de los estudiantes en la entrega de actividades.

id	username	Total	In-time	Overdue	Pending	In-time Rate
1	Student1	5	3	0	2	60 %
2	Student2	5	1	0	4	20 %
3	Student3	5	3	2	0	60 %
4	Student4	5	3	2	0	60 %
5	Student5	5	3	2	0	60 %
6	Student6	5	3	1	1	60 %
7	Student7	5	3	2	0	60 %
8	Student8	5	3	2	0	60 %
9	Student9	5	3	2	0	60 %

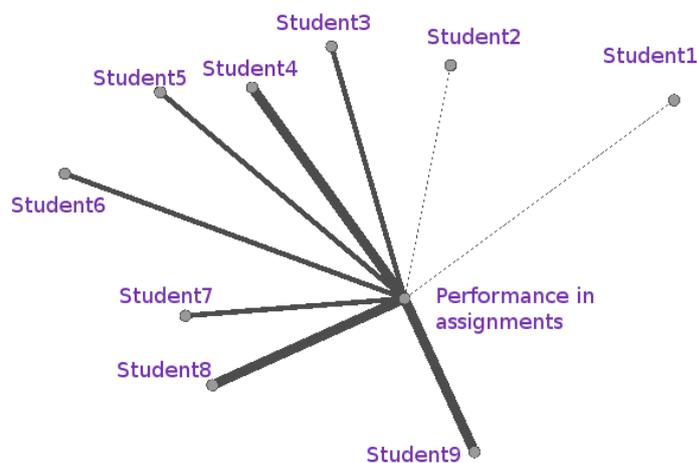


Figura 7.2: Gráfico del desempeño de los estudiantes en la entrega de actividades.

estudiante está vinculado a la actividad central. La anchura del enlace indica desempeño de los estudiantes en el indicador. En el ejemplo, el enlace del primer estudiante es más delgado que la del resto. Eso significa que su rendimiento era menor que la de los otros estudiantes.

### Resolución de problemas

Para evaluar la resolución de problemas vamos a centrarnos en el trabajo de los estudiantes en los foros [Jacob et al., 2009]. Suponemos que los profesores exponen los problemas que hay que resolver en un foro. Los estudiantes tienen que estudiar antes de empezar a trabajar en sus soluciones. Además, se supone que los alumnos habrán sido informados a principios del curso de que los profesores tendrán en cuenta la participación en el foro para evaluar algunos resultados. Por lo tanto, vamos a utilizar el registro de su participación del foro como un indicador relacionado con la resolución de problemas. También los estudiantes pueden plantear problemas en el foro que surjan como fruto de su estudio. Estas participaciones también serán valoradas.

Podría ser que el estudiante participase en un foro escribiendo un comentario que no contribuyera a la solución del problema. Pero, por un lado, podemos considerarlo como una señal de la

Cuadro 7.7: Listado del desempeño de los estudiantes en la participación en el foro.

id	username	Debate-starter	Debate-participation
1	Student1	0	0
2	Student2	0	0
3	Student3	2	0
4	Student4	0	1
5	Student5	1	0
6	Student6	0	3
7	Student7	1	1
8	Student8	0	1
9	Student9	0	1

voluntad de los estudiantes. Y, por otro lado, esto es sólo un indicador. Corresponde al profesor tenerlo en cuenta o no. Por ejemplo, el profesor puede considerar que los estudiantes por debajo de una cantidad mínima de mensajes se deban ser considerados como inactivos y, por tanto, no participaron en el curso. Por el contrario, el profesor podría evaluar como un indicador positivo si se sabe que un determinado alumno alcanza cierto grado de participación.

Para evaluar el desempeño estudiantil en las competencias genéricas de resolución de problemas mediante la participación foro utilizaremos el siguiente código 7.2.

```

1 Evidence Participation_list :
2   get students
3   show participation
4   in forum.
```

Listing 7.2: Código para recoger la participación de lo estudiantes en los foros

Como resultado se obtiene la tabla 7.7. Cada estudiante tiene dos atributos. *Debate-starter* indica los debates que cada alumno ha creado, es decir, creación de nuevas discusiones o debates en el foro. Mientras que *Debate-participación* mide la participación del alumno en las conversaciones para resolver problemas. Vamos a utilizar estos valores como indicadores de desempeño de los estudiantes en la competencia.

También obtenemos un gráfico mostrado en la figura 7.3. Aquí podemos ver las dos actividades representadas como nodos, cada uno en un lado de la gráfica. A cada estudiante se representa como un nodo que está vinculado con las actividades. Cuanto más gruesa sea la relación, más participación de dicho estudiante. En nuestro caso, sólo tres estudiantes comenzaron algún debate. Ellos son los únicos (Student3, Student5 y Student7) con líneas unidas al nodo del lado derecho (*debate-starter*). En cuanto a los enlaces en el lado izquierdo, hasta cinco estudiantes participaron en los debates (*debate-participación*). Podemos observar cómo las líneas más gruesas muestran una mayor actividad (Student3 en el debate de arranque y Student6 en debate-participación).

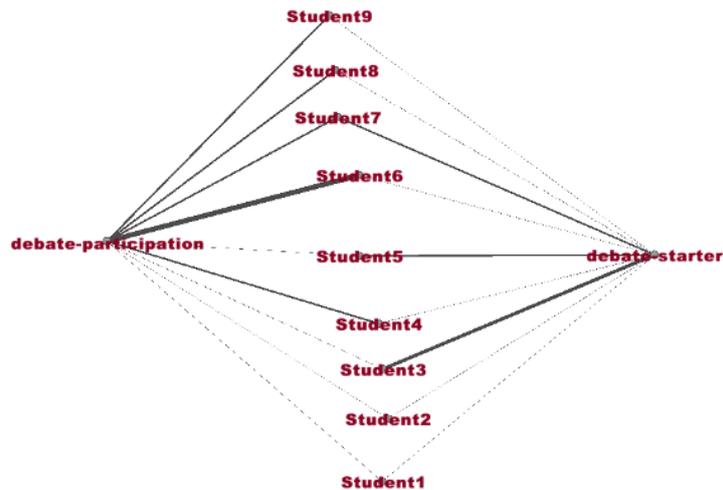


Figura 7.3: Gráfico del desempeño de los estudiantes en la participación en el foro.

## Análisis de resultados

Los resultados obtenidos son los esperados. Están basados en una instalación local de Moodle, en la que manualmente se metieron varios estudiantes y se crearon varios mensajes de estos estudiantes. También se crearon actividades y se simuló como algunos lo enviaban tarde.

Los resultados obtenidos de las diferentes aplicaciones de EvalCourse, de estos dos ejemplos, y de otros cuantos creados coinciden con los datos almacenados en el campus virtual. Por tanto, podemos concluir que el DSL funciona conforme a los requisitos establecidos, y por tanto, el resultado de las pruebas funcionales son satisfactorios.

## 7.4. Pruebas de aceptación

### 7.4.1. Recopilación de información

Para las pruebas de aceptación vamos a utilizar datos reales obtenidos de un curso del campus virtual de la asignatura Procesadores de Lenguajes II, del último curso de la titulación Ingeniería en Informática, de la Universidad de Cádiz. Después, instalamos en nuestro ordenador la plataforma virtual Moodle e importamos dicho curso. Esto debió ser así porque obviamente no tenemos permiso para conectarnos directamente a la base de datos de la plataforma en producción.

Durante el curso, los estudiantes tuvieron que realizar varias tareas obligatorias y un examen al final del semestre. Estas proporcionan una nota sobre el conocimiento de los estudiantes. Utilizamos EvalCourse para obtener indicadores que ayudan en la evaluación de las competencias. A continuación, se describen los indicadores extraídos utilizando EvalCourse.

Cuadro 7.8: Tabla explicativa de cómo calificamos las competencias en base a los indicadores. EvalCourse no genera esta tabla.

Hitos a tiempo	Enviados 5 actividades	Planificación	Trabajo en equipo
Sí	Sí	+	+
no	Sí	-	+
Sí	no	+	-
no	no	-	-

### Extracción de indicadores de los hitos o actividades

Durante el semestre, los estudiantes tenían que trabajar en pequeños equipo de dos o tres miembros. Cada equipo tuvo que realizar varios hitos durante el curso. Estos hitos son partes de un proyecto de software que tuvieron que desarrollar y subir a la actividad correspondiente. Cada hito tenía una fecha límite, aunque el sistema permite a los estudiantes que presenten estas actividades después de este plazo. Para cada equipo, sólo uno de los miembros tuvo que cargar cada actividad. Por ejemplo, si tenemos un equipo con miembros de la S1, S2 y S3, podrían presentar las actividades con las siguientes distribuciones: S1 presenta las cinco actividades, pero S2 y S3 no presentan nada, o S1 presenta la primera actividad, S2 presenta la segunda y S3 el resto, etc. Lo que es importante es que todas las actividades se presentan en cada equipo, no importa quién lo haga. Todos los compañeros de equipo compartirán la calificación.

Las competencias que queríamos evaluar fueron planificación y gestión del tiempo y trabajo en equipo, y para ello tuvimos en cuenta si las actividades se habían entregado o no a tiempo. Como podemos ver en la tabla 7.8, aquellos equipos que hayan presentado todos sus hitos en el tiempo y que entregaron exactamente las cinco actividades fueron evaluadas en las competencias con una calificación positiva. Si un equipo presenta exactamente cinco actividades, pero no en tiempo, entonces se evaluaron en trabajo en equipo con una calificación positiva, pero no en planificación y gestión del tiempo. Si enviaron todos los hitos a tiempo, pero más de cinco actividades (porque más de un compañero haya subido la misma actividad), serán evaluados positivamente en la planificación y gestión del tiempo, pero negativamente en trabajo en equipo. Por supuesto, si los hitos no se presentaron a tiempo y más o menos de cinco actividades fueron presentadas por un grupo, ambas competencias fueron evaluadas como negativas.

### Extracción de indicadores del foro

Como se ha comentado antes, los estudiantes trabajan en equipos pequeños. Cada equipo en el curso tuvo un foro para la comunicación interna. Se utilizó el foro para obtener indicadores para evaluar un par de competencias: habilidades interpersonales y de liderazgo.

Las habilidades interpersonales de cada estudiante se calculan utilizando como indicador la participación total en el foro. Un estudiante que tiene tres o más participaciones en el foro tendrá calificación positiva en la competencia de habilidad interpersonal.

Como indicador del liderazgo tomamos en cuenta la cantidad de debates que cada estudiante comenzó. Es decir, se utiliza el valor *debate-stater* para evaluarlo. Un estudiante que ha iniciado

Cuadro 7.9: Listado del desempeño de los equipos de estudiantes en planificación y trabajo en equipo.

Team	Total	In-time	Overdue	Pending	Planning	Teamwork
team01	5	5	0	0	+	+
team02	5	2	1	2	-	-
team03	5	4	1	0	+	+
team04	5	5	0	0	+	+
team05	5	4	1	0	+	+
team06	5	1	1	3	-	-
team07	5	5	0	0	+	+
team08	5	2	2	1	-	-
team09	5	4	1	0	+	+
team10	5	5	0	0	+	+
team11	5	2	1	2	-	-
team12	5	6	1	0	+	-
team13	5	4	0	1	-	-
team14	5	4	1	0	+	+

dos o más debates tendrá una calificación positiva en la competencia de liderazgo.

#### 7.4.2. Análisis de resultados

##### Extracción de indicadores de los hitos o actividades

Moodle en su configuración permite la creación de grupos de trabajo. Desafortunadamente, no lo definimos explícitamente en el sistema. Así que tuvimos que obtener la lista de los estudiantes de las tareas usando EvalCourse, y de forma manual destacar el equipo de cada estudiante.

En la tabla 7.9 podemos ver el resultado final del desempeño de los equipos en competencias de acuerdo a los indicadores. Hay ocho equipos que obtuvieron la marca positiva en la competencia de planificación y gestión del tiempo y en la de trabajo en equipo, ya que presentaron todos los hitos en tiempo y presentaron exactamente los cinco hitos. Hay cinco equipos que tienen una nota negativa en ambas competencias, ya que no presentaron todas las actividades. Sólo un grupo tiene una calificación positiva en la competencia de planificación y gestión del tiempo, pero una calificación negativa en la competencia de trabajo en equipo. Esto es así porque no estaban bien coordinados y presentaron actividades repetidas.

##### Extracción de indicadores del foro

Sólo 18 alumnos de la 36 matriculados en el curso participaron en el foro. Esto no significa que no hayan trabajado estas competencias, sólo significa que no han demostrado que lo hicieron con el uso de la herramienta. Podemos analizar los resultados del uso de EvalCourse en el trabajo en el foro con la tabla 7.10 y en el gráfico 7.4. Muy pocos estudiantes pudieron ser evaluados de estas competencias. Hay muchos que destacaron sobre resto. A pesar de que el foro haya sido utilizado por un grupo reducido de alumnos, hemos observado que los estudiantes que obtuvieron buena nota en estos indicadores realmente tuvieron un buen desempeño durante el

Cuadro 7.10: List of students' performance in forum participation in case study.

id	username	Debate-starter	Debate-participation	Total
43	S2	0	1	1
44	S3	3	4	7
45	S4	2	2	4
46	S5	0	1	1
48	S7	4	3	7
50	S9	6	8	14
51	S10	1	1	2
53	S12	0	1	1
55	S14	1	1	2
59	S18	1	2	3
60	S19	2	0	2
61	S20	0	2	2
62	S21	0	1	1
67	S26	0	1	1
70	S29	0	1	1
71	S30	2	1	3
75	S34	1	5	6
78	S37	2	4	6

curso en la consecución de estas competencias. Por lo tanto, podemos decir que los indicadores que hemos elegido parecen buenas evidencias sobre las competencias logradas por los estudiantes.

En conclusión, y de manera informal, el profesor del curso confirmó que las notas obtenidas mediante los indicadores son muy parecidas a las que él calculó de forma manual para cada estudiante en el curso.

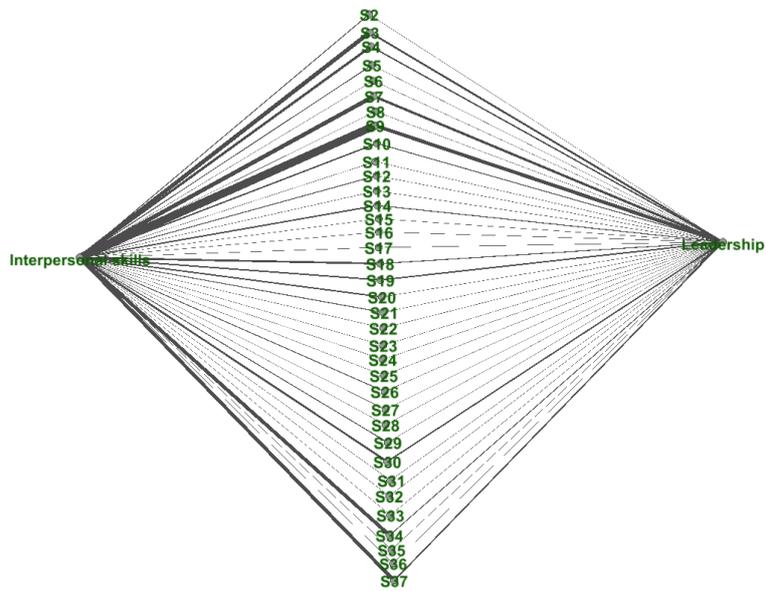


Figura 7.4: Gráfico del desempeño de los estudiantes en el trabajo en el foro.



Parte III

Epílogo







## Capítulo 8

# Manual de usuario

Las instrucciones de uso del sistema se detallan a continuación.

### 8.1. Introducción

EvalCourse es un DSL creado para ayudar al docente en la obtención de indicadores en el desempeño de sus alumnos en competencias genéricas. Estos indicadores los tomará del trabajo de los estudiantes en un curso de la plataforma virtual Moodle.

Medir las competencias genéricas de un alumno puede ser un trabajo demasiado subjetivo y poco escalable. Saber cómo es un alumno trabajando en equipo o cómo se organiza el trabajo son competencias difíciles de medir con una mera observación. Cómo estas dos hay otras muchas competencias en la misma situación. Son características que conoces de una persona cuando realmente la conoces. Es imposible saber cómo son todos tus alumnos y menos aun cuando el número de estudiantes es muy grande.

Moodle almacena gran cantidad de información acerca de la interacción de cada alumno con el curso. Esta interacción nos puede dar pistas sobre cómo trabaja un alumno en diferentes competencias. Es cuestión de definir indicadores de dichas competencias en base a dicha interacción. Es trabajo del docente declarar sus indicadores. Con EvalCourse, se da la opción a los docentes de escribir sus indicadores y ejecutarlos frente a su curso del campus virtual.

### 8.2. Características

Las principales funcionalidades del sistema son:

- Consulta de indicadores del desempeño de los estudiantes en los siguientes conceptos:
  - Actividades
  - Foro
  - Campus
  - Talleres
- En dichos conceptos se podrá consultar lo siguiente:

- Acceso
  - Participación
  - Entrega de hitos
- Creación de fichero de transformación de Pentaho
  - Ejecución de transformación de Pentaho y obtención de:
    - Listado de desempeño de alumnos en competencias en formato XML
    - Gráfico de desempeño de alumnos en formato Gephi

### 8.3. Requisitos previos

Requisitos software:

- Eclipse
- Herramienta EMF
- Pentaho (necesario si desea ejecutar las transformaciones fuera de EvalCourse)
- Gephi (necesario para poder visualizar los gráficos)

### 8.4. Utilización

Para comenzar debemos abrir Eclipse y crear un nuevo proyecto. Tras seleccionar *EvalCourse project* como el tipo de proyecto EvalCourse 8.1, deberemos dar a siguiente y darle un nombre al proyecto EvalCourse 8.2. Ahora pulsamos en finalizar y esperamos a que se cree toda la estructura de archivos del proyecto EvalCourse.

Se crea una estructura de carpetas. Entre estas carpetas, hay una de ella que es *src*. Ahí estará el fichero *process.evc*, que será el fichero en el que debemos escribir la consulta. Nos damos cuenta que al crear el proyecto, el editor nos marca un error en el archivo *process.evc* 8.3. El entorno de trabajo marcará con un error el fichero *process.evc* mientras éste no contenga una consulta que satisfaga la sintaxis del lenguaje.

Para crear una consulta debemos utilizar la sintaxis especificada en 8.1.

```

1 Evidence name_of_the_indicator:
2     get students
3     show milestones | participation | access
4     in assignment | forum | campus [list of ids].

```

Listing 8.1: Formato y palabras reservadas

Un ejemplo de consulta sería el que aparece en la figura 8.4. Vemos que una vez que la consulta satisface la especificación del lenguaje deja de avisar del error tal y como veíamos en la figura anterior.

Cuando guardamos el trabajo escrito en el fichero *process.evc* automáticamente se generan los ficheros de transformación de Pentaho en la carpeta *src-gen* (figura 8.5). Estos ficheros son:

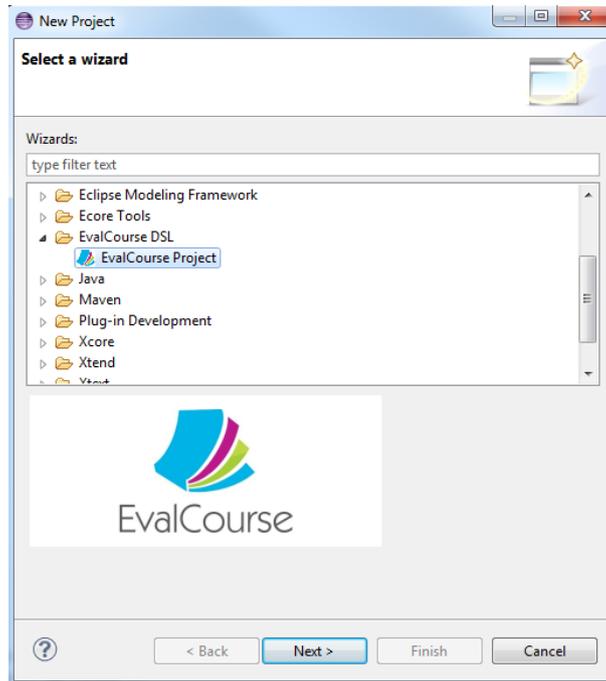


Figura 8.1: Creación de un nuevo proyecto de EvalCourse

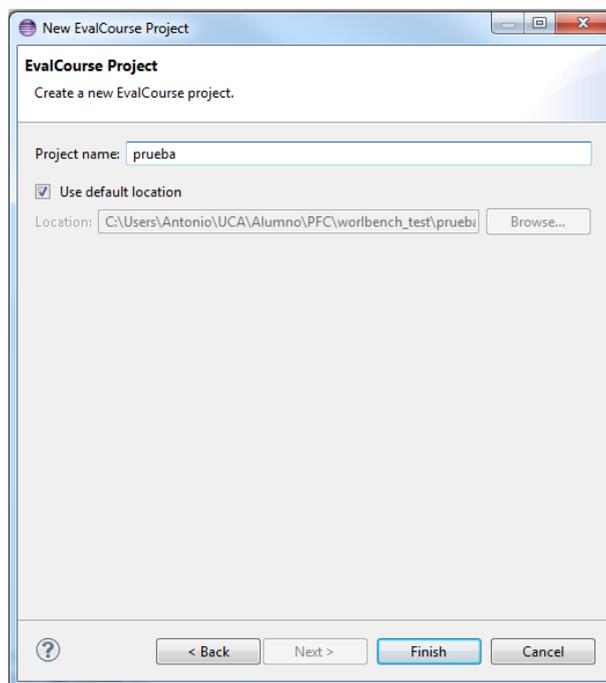


Figura 8.2: Dando nombre a un nuevo proyecto de EvalCourse

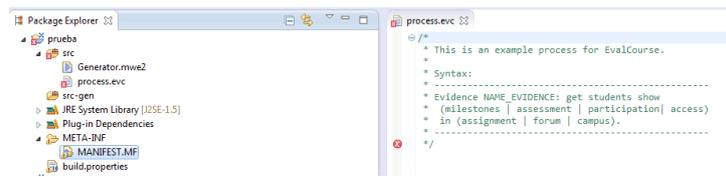


Figura 8.3: Estructura de ficheros que se crea con un nuevo proyecto de EvalCourse

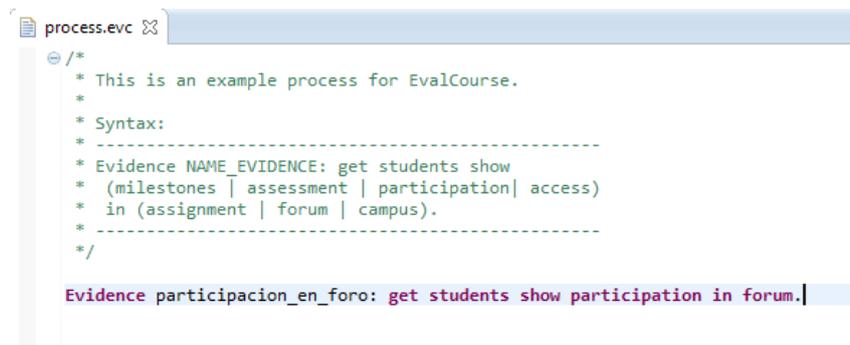


Figura 8.4: Consulta escrita en fichero process.evc

- *participacion\_en\_foro.ktr*: Transformación en formato Pentaho para extraer la información escrita en la consulta del sistema LMS.
- *participacion\_en\_foro.xsl*: XSL que transforma la información de un fichero xml a un fichero en formato Gephi.
- *participacion\_en\_foroToGephi.ktr*: Transformación en formato Pentaho que utilizará la transformación XSL para obtener el fichero en formato Gephi.
- *Transformation.java*: Programa escrito en Java que utilizando las bibliotecas de Pentaho permitirá realizar las transformaciones desde EvalCourse, y así no tener que abrir el programa Pentaho.

Para obtener los resultados debemos ejecutar el fichero Transformation.java. Pinchamos en él y le damos a ejecutar. En este momento comenzarán a ejecutarse las transformaciones, y finalmente obtendremos los ficheros con los resultados solicitados al inicio 8.6.

Los dos ficheros de que dispondremos tendrán el nombre de la evidencia que declaramos en el código y dos extensiones. El primero de ellos es *.gexf*, para el gráfico. Si abrimos este fichero con el programa Gephi veremos algo así: 8.7. Cada vértice representa un estudiante o una actividad. Cada estudiante está unido con las dos actividades que se evalúan, y dependiendo del grosor de la arista que los une el desempeño del estudiante en esa actividad habrá sido mayor o menos.

Por otro lado tenemos el fichero de extensión *.xml*, que será el listado textual con la misma información (figuras 8.8 y 8.9).

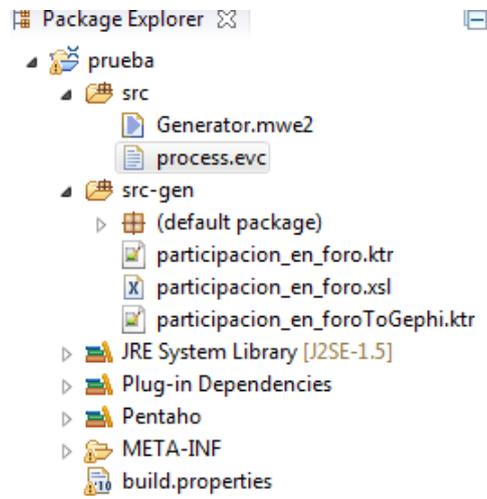


Figura 8.5: Ficheros creados en carpeta src-gen

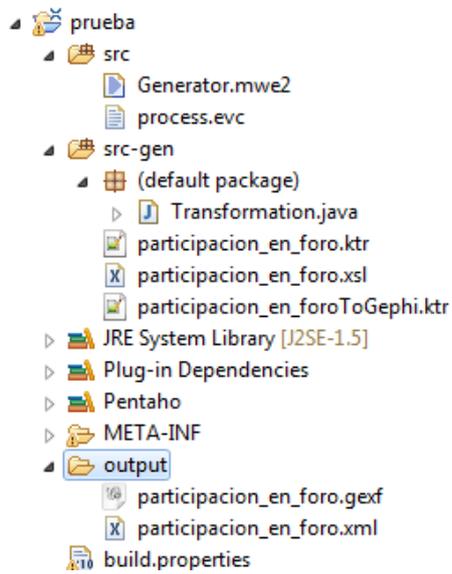


Figura 8.6: Ficheros de salida

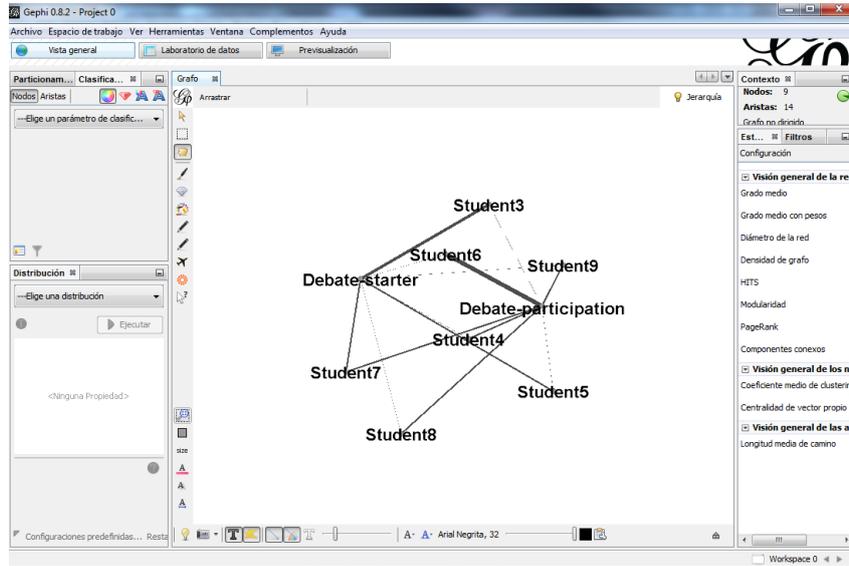


Figura 8.7: Resultado gráfico en software Gephi

```

participacion_en_foro.xml
<?xml version="1.0" encoding="UTF-8"?>
<Rows>
<Row><id> 00000000000005</id> <username>Student3</username> <firstname>Student3</firstname> <lastname>Student3</lastname>
<Row><id> 00000000000006</id> <username>Student4</username> <firstname>Student4</firstname> <lastname>Student4</lastname>
<Row><id> 00000000000007</id> <username>Student5</username> <firstname>Student5</firstname> <lastname>Student5</lastname>
<Row><id> 00000000000008</id> <username>Student6</username> <firstname>Student6</firstname> <lastname>Student6</lastname>
<Row><id> 00000000000009</id> <username>Student7</username> <firstname>Student7</firstname> <lastname>Student7</lastname>
<Row><id> 00000000000010</id> <username>Student8</username> <firstname>Student8</firstname> <lastname>Student8</lastname>
<Row><id> 00000000000011</id> <username>Student9</username> <firstname>Student9</firstname> <lastname>Student9</lastname>
</Rows>

```

Figura 8.8: Resultado textual en xml - lado izquierdo

```

<Debate-starter> 2</Debate-starter> <Debate-participation> 0</Debate-participation> <Total> 2</Total> </Row>
<Debate-starter> 0</Debate-starter> <Debate-participation> 1</Debate-participation> <Total> 1</Total> </Row>
<Debate-starter> 1</Debate-starter> <Debate-participation> 0</Debate-participation> <Total> 1</Total> </Row>
<Debate-starter> 0</Debate-starter> <Debate-participation> 3</Debate-participation> <Total> 3</Total> </Row>
<Debate-starter> 1</Debate-starter> <Debate-participation> 1</Debate-participation> <Total> 2</Total> </Row>
<Debate-starter> 0</Debate-starter> <Debate-participation> 1</Debate-participation> <Total> 1</Total> </Row>
<Debate-starter> 0</Debate-starter> <Debate-participation> 1</Debate-participation> <Total> 1</Total> </Row>

```

Figura 8.9: Resultado textual en xml - lado derecho

## Capítulo 9

# Manual de instalación y explotación

Las instrucciones de instalación y explotación del sistema se detallan a continuación.

### 9.1. Introducción

El principal objetivo de EvalCourse es un DSL es la obtención de indicadores en el desempeño de sus alumnos en competencias genéricas. Estos indicadores se extraerán del trabajo de los estudiantes en un curso de la plataforma virtual Moodle.

El ámbito de aplicación de EvalCourse es la docencia universitaria. Los profesores lo podrán utilizar para evaluar el desarrollo de competencias de aquellos de sus estudiantes que intervengan en el campus virtual.

### 9.2. Requisitos previos

Requisitos software:

- Eclipse, con la herramienta EMF y el plugin de EvalCourse.
- Pentaho (necesario si desea ejecutar las transformaciones fuera de EvalCourse)
- Gephi (necesario para poder visualizar los gráficos)

Para que la aplicación funcione correctamente será necesario configurar el acceso a la base de datos. Este se consigue mediante la definición de los parámetros de conexión en el fichero *dbconexion.csv* 9.1, que se encuentra en la carpeta raíz de Eclipse. Un ejemplo de fichero se ve en el fragmento de código 9.2.

Listing 9.1: Esquema del archivo dbconexion.csv

```
1 name_of_the_connection : db_server : db_engine : db_mode : db_name : db_port :
2       db_username : encrypted_password
```

Listing 9.2: Ejemplo de fichero dbconexion.csv

```
1 mudel : localhost : MYSQL : Native : moodle : 3306 :
2       moodle : Encrypted 2e7fad9850be9c1869b2fad76e9b487ee
```

- *name\_of\_the\_connection*: Nombre de la conexión de la base de datos. El usuario del programa puede poner el nombre que quiera.
- *db\_server*: Nombre del servidor que contiene la base de datos. Podrá ser una dirección IP o localhost.
- *db\_engine*: Motor de base de datos. La Aplicación solo ha sido probada con el sistema MYSQL.
- *db\_mode*: Modo de conexión. Por defecto 'Native'.
- *db\_name*: Nombre de la base de datos que contiene la base de datos de Moodle.
- *db\_port*: Puerto por el que nos conectaremos a la base de datos.
- *db\_username*: Usuario de la base de datos.
- *encrypted\_password*: Encrypted + la clave encriptada.

### 9.3. Procedimientos de instalación

- Si dispone de Eclipse instalado:
  - Descargar los plugins de EvalCourse (figura 9.1)
  - Copiar los plugins de EvalCourse en la carpeta plugins ubicada en la carpeta raíz de Eclipse (figura 9.2)
  - Ya puede ejecutar Eclipse y seleccionar un proyecto de tipo EvalCourse
- Si no dispone de Eclipse instalado, se puede descargar la versión de Eclipse que incorpora ya EvalCourse.

#### Enlaces de interés

Las aplicaciones las puede descargar de los siguientes enlaces:

- EvalCourse: <https://www.assembla.com/spaces/evalcourse/documents>
- Pentaho: <http://www.pentaho.com>
- Gephi: <https://gephi.org/>

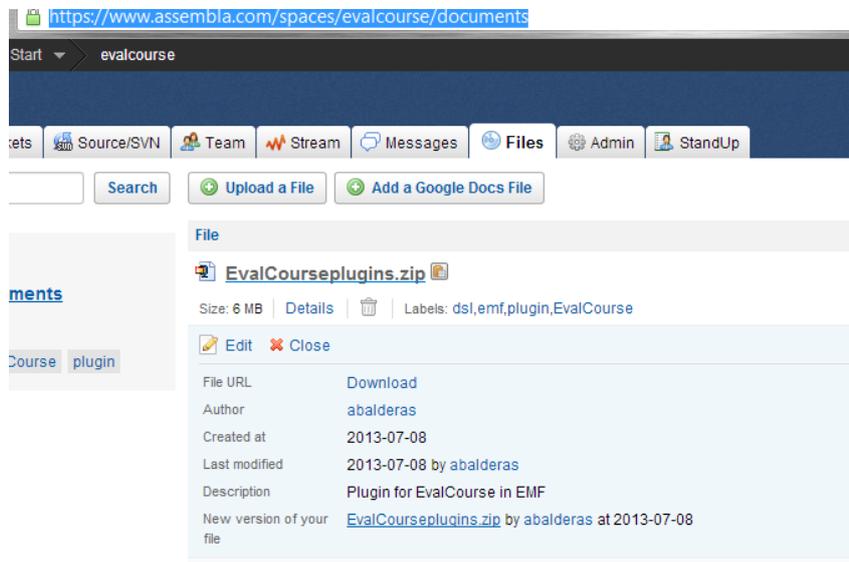


Figura 9.1: Página de descarga de EvalCourse

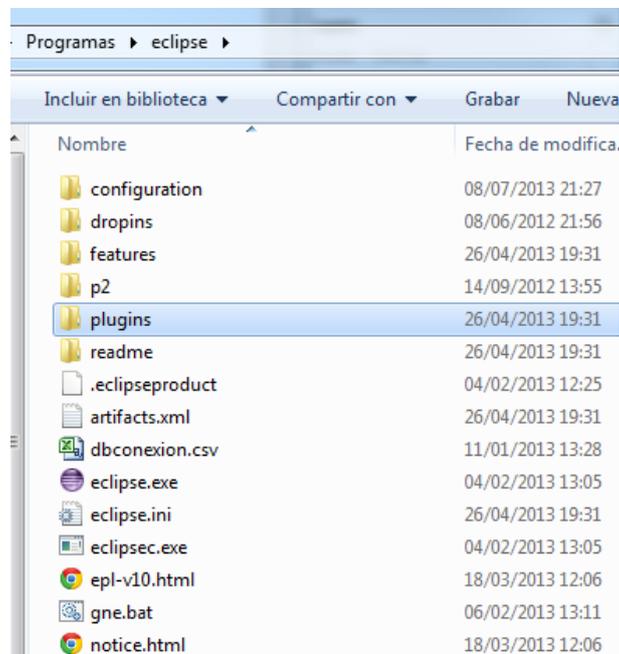


Figura 9.2: Carpeta de plugins de Eclipse



# Capítulo 10

## Conclusiones

En este último capítulo se detallan las lecciones aprendidas tras el desarrollo del presente proyecto y se identifican las posibles oportunidades de mejora sobre el software desarrollado.

### 10.1. Objetivos

#### Objetivo principal

- Creación de un DSL con una sintaxis sencilla e intuitiva que extraiga información importante para el docente sobre la participación de los alumnos en Moodle: se ha creado un DSL que realiza esta función. No es menos cierto que es misión del profesor definir sus indicadores de competencias, o si la información que recibe de unos indicadores le son o no válidos. Cada curso es diferente y que un profesor fomente en mayor o menor medida el uso del curso del campus virtual influirá en los resultados. De cualquier forma, la herramienta se ha desarrollado, se ha utilizado y se ha probado con datos reales que han sido enviados ([Balderas et al., 2013a] y [Balderas et al., 2013b] y aceptados([Balderas et al., 2013b] en congresos de TEL<sup>1</sup>, demostrando su viabilidad. En trabajos futuros se darán más pistas sobre los próximos pasos.

#### Subobjetivos

- Crear un metamodelo lo suficientemente genérico para nuestro DSL, de forma que sea válido no sólo para Moodle, sino también para futuras versiones del DSL utilizables en otros sistemas: se ha creado un metamodelo cuya última versión consideramos que es apta para ser utilizada en cualquier LMS 4.15. Desde luego, en Moodle nos ha sido de utilidad. Para verificar su viabilidad para otros LMS habrá que esperar a ver los resultados de su aplicación en trabajos futuros.
- La sintaxis de nuestro DSL deberá ser sencilla y cercana al lenguaje humano: se ha creado una sintaxis compuesta por palabras del idioma inglés, cuya construcción literal indica claramente qué se pretende obtener. Por tanto consideremos que el objetivo de que sea de sencilla utilidad para personas no familiarizadas con el entorno tecnológico se ha conseguido.
- Investigar qué información sería de importancia desde el punto de vista de las competencias en un sistema Moodle para generar en nuestro DSL las consultas que nos devuelvan

---

<sup>1</sup>Technology Enhanced Learning

dicha información: se hizo una consulta a varios profesores del Departamento de Ingeniería Informática para conocer su opinión y qué parámetros creían válidos para evaluar las competencias a través de la información de Moodle. Con esta información se adoptaron para esta primera versión del software los indicadores de los foros, actividades, accesos al campus y talleres.

- Investigar estructura de los ficheros fuente del software Kettle de Pentaho para que nuestro DSL pueda proporcionar como salida un fichero válido para este software: mediante la realización de muchos ejemplos y estudio del xml que hay detrás de las transformaciones de Pentaho, se consiguieron crear varios algoritmos para mapear las consultas de EvalCourse a un formato válido y funcional de dichos ficheros.
- Integrar el software como un todo: Eclipse, EvalCourse y Kettle para facilitar al usuario la instalación e implantación del sistema: Con la ayuda de EMF y la bibliotecas de Kettle conseguimos el objetivo de integrar todo en un único IDE.

## 10.2. Lecciones aprendidas

### 10.2.1. Planificación

Se muestra un diagrama de Gantt del tiempo que se ha empleado (figura 10.1) en el desarrollo del proyecto y otro con la comparación entre el tiempo planificado y el que en realidad se ha empleado (figura 10.2).

El tiempo esperado para la realización del proyecto era de 160 días (figura 2.5). Sin embargo, se tardó en completar 207 días (figura 10.3). El retraso de 47 días se debe sobre todo a labores de investigación que llevaron más tiempo del que se pensaba inicialmente. Las áreas que más retraso supusieron fueron, por un lado, la integración de las librerías de Pentaho con el DSL para poder ejecutar la transformación desde el propio DSL; y, por otra parte, el estudio y mapeo de los ficheros XML para Pentaho [Bouman and Dogen, ] y Gephi.

### 10.2.2. Costes

Vamos a comparar la diferencia de costes entre los calculados a priori y los que han sido realmente:

- Se tardó en realizar el proyecto 207 días = 6,9 meses
- Por lo que hubieran sido necesarias 6,9 personas / mes
- Siendo el salario base mensual de los trabajadores + Complemento categoría: 2.746,65 €
- Total real = 6,9 trabajadores x 2.746,65 € = **18.951,88 €**
- Sin embargo, la estimación temporal previa a la realización del proyecto fue menor: 5,34 personas / mes
- La estimación ascendía a **14.648,80 €**
- La diferencia entre el coste estimado y el coste real ha sido de **4.303,08 €**

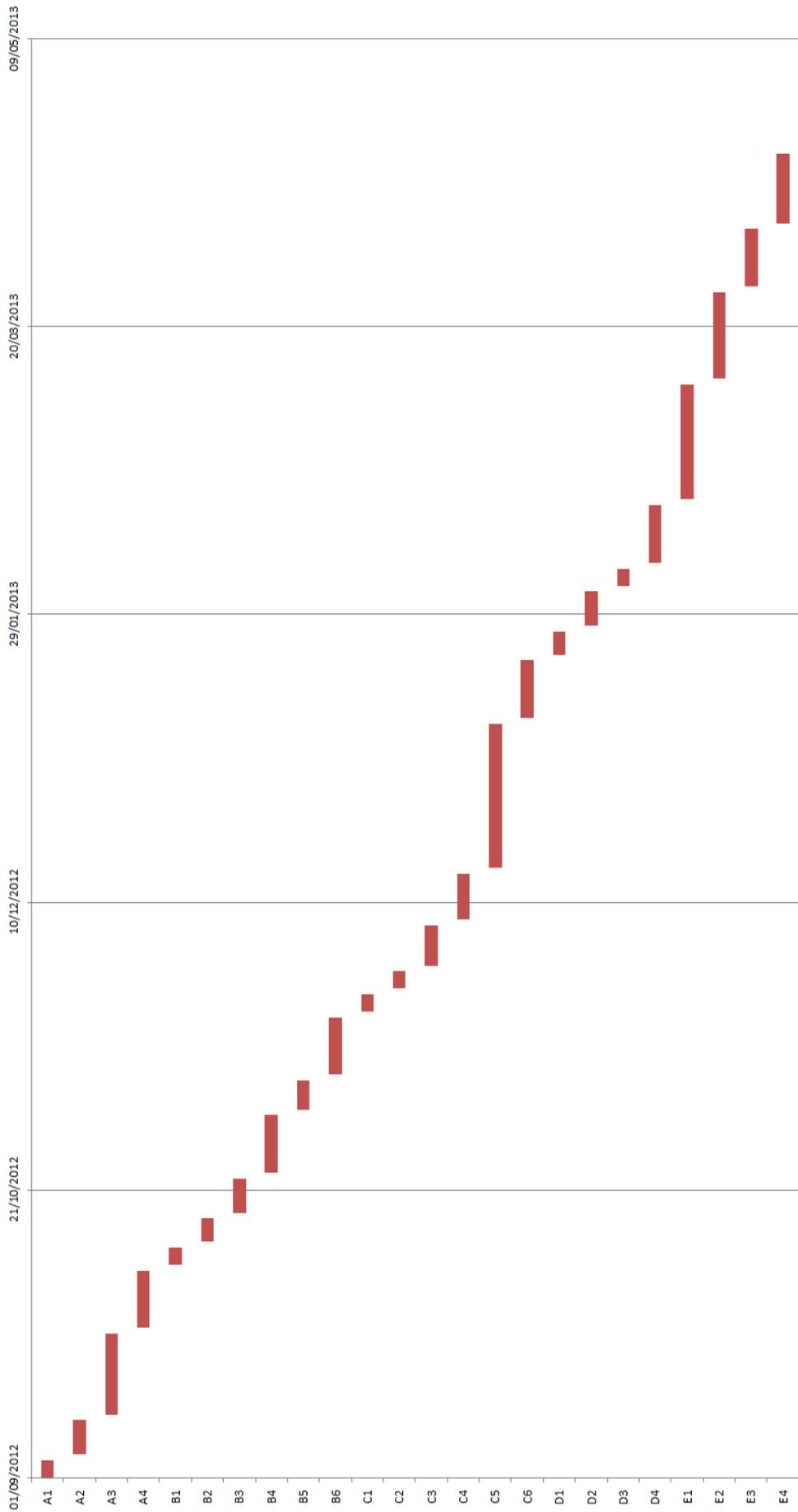


Figura 10.1: Diagrama de Gantt del tiempo empleado

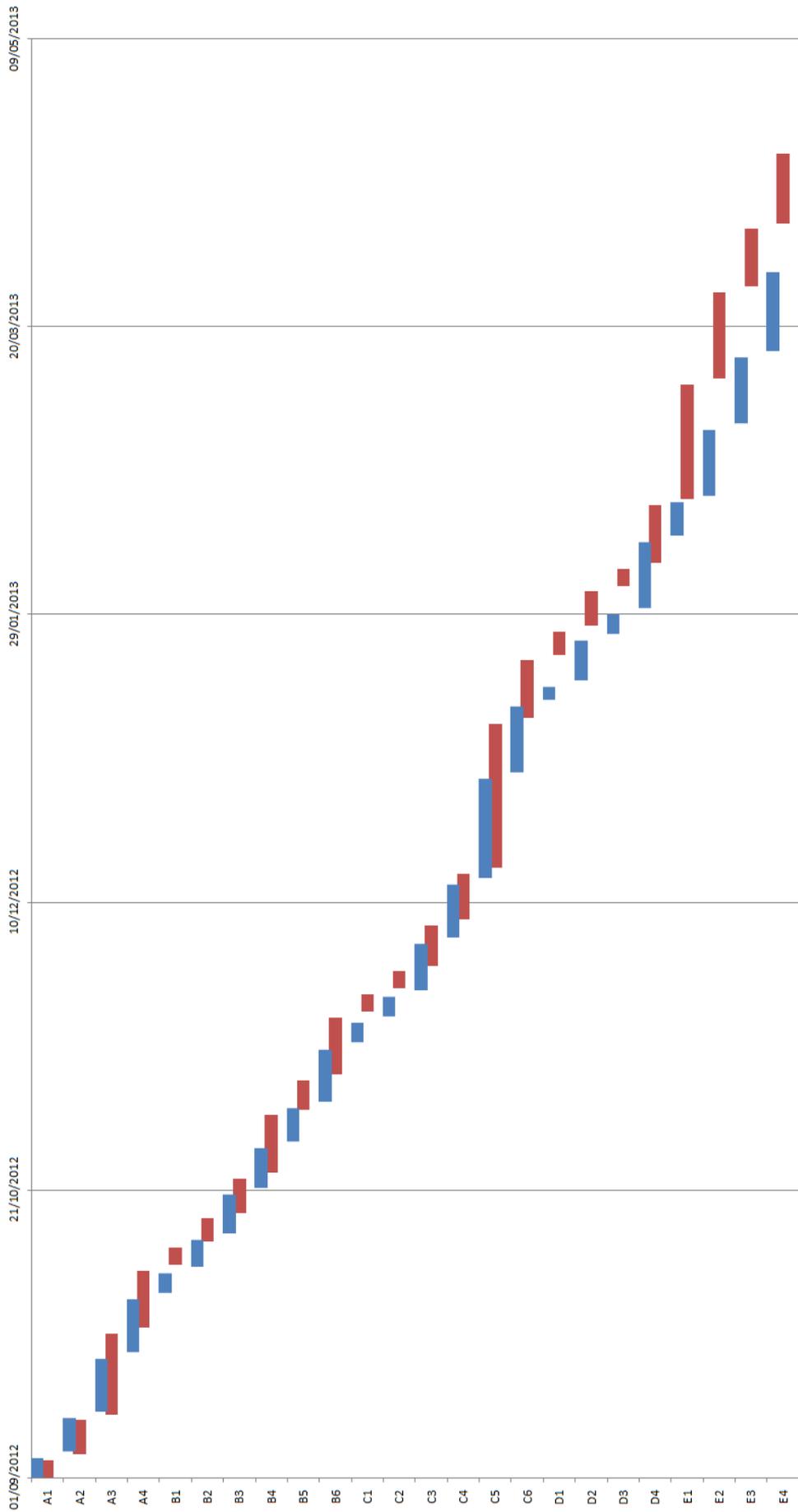


Figura 10.2: Diagrama de Gantt - Comparacion

Actividades	Inicio	Duración (días)	Fin
A1	01/09/2012	3	04/09/2012
A2	05/09/2012	6	11/09/2012
A3	12/09/2012	14	26/09/2012
A4	27/09/2012	10	07/10/2012
B1	08/10/2012	3	11/10/2012
B2	12/10/2012	4	16/10/2012
B3	17/10/2012	6	23/10/2012
B4	24/10/2012	10	03/11/2012
B5	04/11/2012	5	09/11/2012
B6	10/11/2012	10	20/11/2012
C1	21/11/2012	3	24/11/2012
C2	25/11/2012	3	28/11/2012
C3	29/11/2012	7	06/12/2012
C4	07/12/2012	8	15/12/2012
C5	16/12/2012	25	10/01/2013
C6	11/01/2013	10	21/01/2013
D1	22/01/2013	4	26/01/2013
D2	27/01/2013	6	02/02/2013
D3	03/02/2013	3	06/02/2013
D4	07/02/2013	10	17/02/2013
E1	18/02/2013	20	10/03/2013
E2	11/03/2013	15	26/03/2013
E3	27/03/2013	10	06/04/2013
E4	07/04/2013	12	19/04/2013

Figura 10.3: Resumen de fechas una vez acabado el proyecto

La brecha entre la estimación y el coste real se debe al desajuste de 47 días en la realización del proyecto. Este desajuste se ha debido sobre todo al desconocimiento inicial de la tecnología. Si tuviera que volver a presupuestar un trabajo de este tipo sería mucho más exacto en el cálculo, pues ya sí conozco la tecnología. Sin embargo, como reflexión final diría que es difícil presupuestar a priori aquello que no se domina.

### 10.2.3. Experiencia

Este proyecto ha sido para mí una primera piedra de toque a lo que debe ser mi carrera como investigador. Ha sido mucho más el tiempo invertido en investigación que en desarrollo. No sólo en la investigación de cara a la realización del proyecto, sino también en los dos artículos que hemos escrito para los congresos EC-TEL 2013 (Eight European Conference on Technology Enhanced Learning) [EC-TEL, ] e ISELEAR'13 (4th International Workshop on Software Engineering for E-learning) [ISELEAR2013, ]. Hay mucho por andar en esta área de la ciencia. Hay trabajos sobre competencias, muchos trabajos, pero muy pocos tienen una tecnología que proporcionen unos datos cuantitativos. Con EvalCourse obtenemos eso, unos datos que nos sirven de indicadores para evaluar objetivamente las competencias.

Medir el trabajo es difícil. Aunque en el apartado de planificación se hayan presentado los diagramas de Gantt, no deja de ser cierto que el trabajo se ha visto intercalado con la lectura de mucho material para los artículos.

He disfrutado con el desarrollo del DSL, y voy a seguir trabajando en él como parte de mi tesis doctoral.

## 10.3. Trabajo futuro

En la educación, el enfoque en la evaluación de experiencias de aprendizaje está cambiando del conocimiento a las competencias. Las competencias son la práctica, las habilidades y los atributos que podemos desarrollar en todos los aspectos de nuestra vida. El desarrollo tanto de competencias genéricas como claves deben integrarse en los planes de estudio junto a las competencias específicas.

La evaluación de ciertas competencias suele ser una tarea subjetiva. Por ejemplo, la obtención de un enfoque objetivo de evaluar una capacidad de estudiante en la competencia de trabajo en equipo puede ser muy difícil. Esto se vuelve aún más problemático cuando el número de estudiantes es elevado. Utilizando un LMS, los profesores podrían analizar la interacción producida por el alumno. Los LMS almacenan información acerca de las actividades del curso que requiere un filtrado antes de que se pueda utilizar. Los datos recogidos se utilizarán como indicadores del trabajo de los estudiantes.

En este proyecto se ha desarrollado EvalCourse, un DSL para definir procedimientos de evaluación. Los profesores podrán disponer de información contenida en la base de datos sin tener conocimientos técnicos en programación utilizando una sintaxis muy sencilla.

Tenemos dos líneas de trabajo futuro. Primero, identificaremos otros indicadores contenidos en Moodle y lo incluiremos en la sintaxis. En segundo lugar, nuestro lenguaje puede expresar

consultas que valdrían para otros LMS. Aunque de momento, el DSL sólo trabaja con Moodle, pensamos expandirlo para ser utilizados en otros LMS.



# Bibliografía

- [Balderas et al., 2012a] Balderas, A., Palomo-Duarte, M., Dodero, J., and Ruiz Rube, I. (2012a). Qualitative assessment of wiki-based learning processes.
- [Balderas et al., 2013a] Balderas, A., Ruiz, I., Dodero, J., and Palomo-Duarte, M. (2013a). A generative computer language to customize online learning assessments.
- [Balderas et al., 2013b] Balderas, A., Ruiz, I., Dodero, J., Palomo-Duarte, M., and Berns, A. (2013b). A generative computer language to customize online learning assessments.
- [Balderas et al., 2012b] Balderas, A., Tocino, J., Palomo-Duarte, M., Dodero, J., and Ruiz Rube, I. (2012b). Assessmediawiki.
- [Bettini, ] Bettini, L. *Implementing Domain-Specific Languages with Xtext and Xtend*.
- [Bouman and Dogen, ] Bouman, R. and Dogen, J. *Pentaho Solutions: Business Intelligence and Data Warehousing with Pentaho and MySQL*.
- [Britton and Tesser, 1991] Britton, B. K. and Tesser, A. (1991). Effects of time-management practices on college grades. *Journal of Educational Psychology*, 83(3):405–410.
- [Cabot, ] Cabot, J. Modeling languages. <http://modeling-languages.com/>.
- [CCOO, 2010] CCOO (2010). Tablas salariales 2010 IV Convenio Colectivo. <http://www.uca.es/sindicato/ccoo/documentos/tabla-salarial-pas-laboral-2010.pdf>.
- [dosideas.com, 2009] dosideas.com (2009). Los lenguajes específicos de dominio. <http://www.dosideas.com/noticias/actualidad/487-los-lenguajes-especificos-de-dominio.html>.
- [EC-TEL, ] EC-TEL. Ec-tel 2013. <http://www.ec-tel.eu/>.
- [Eclipse, a] Eclipse, E. Eclipse modeling framework. <http://www.eclipse.org/modeling/emf/>.
- [Eclipse, b] Eclipse, E. Ecore component. <http://www.eclipse.org/modeling/emft/?project=ecoretools>.
- [Eclipse, c] Eclipse, X. xtend. <http://www.eclipse.org/xtend/>.
- [Fowler and Parsons, 2011] Fowler, M. and Parsons, R. (2011). *Domain-specific languages*. Addison-Wesley, Upper Saddle River, NJ.
- [Gronback and C., ] Gronback and C., R. *Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit*.

- [Haan, ] Haan, J. d. Dsl and mde, necessary assets for model-driven approaches.
- [ISELEAR2013, ] ISELEAR2013. Iselear 2013. <http://teemconference.eu/tracks/4th-international-workshop-on-software-engineering-for-e-learning-iselear13/>.
- [Jacob et al., 2009] Jacob, S. M., Lee, B., and Lueckenhausen, G. R. (2009). Measuring critical thinking skills in engineering mathematics using online forums. *2009 International Conference on Engineering Education, ICEED2009 - Embracing New Challenges in Engineering Education*, pages 225–229.
- [Kurtev et al., 2006] Kurtev, I., Bézivin, J., Jouault, F., and Valduriez, P. (2006). Model-based dsl frameworks. In *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*, OOPSLA '06, pages 602–616.
- [Palomo-Duarte et al., 2012] Palomo-Duarte, M., Doderó, J., Medina-Bulo, I., and Rodríguez-Posada, E. (2012). Assessment of collaborative learning experiences by graphical analysis of wiki contributions. *Interactive Learning Environments*.
- [Palomo-Duarte et al., 2011] Palomo-Duarte, M., Doderó, J., Medina-Bulo, I., Rodríguez-Posada, E., Palomo, F., García, A., and Tocino, J. (2011). Statmediawiki.
- [Stahl et al., 2006] Stahl, T., Voelter, M., and Czarnecki, K. (2006). *Model-Driven Software Development: Technology, Engineering, Management*. John Wiley & Sons.
- [Steinberg et al., 2009] Steinberg, D., Budinsky, F., Paternostro, M., and Merks, E. (2009). *EMF: Eclipse Modeling Framework 2.0*. Addison-Wesley Professional, 2nd edition.
- [Tuning, 2012] Tuning, P. (2012). Competences.

Parte IV  
Apéndice



## .1. Contenido de fichero xml de Pentaho (ejemplo)

Contenido de fichero xml de Pentaho: Ejemplo indicadores\_hitos.ktr

```
1 <transformation>
2   <info>
3     <name>indicador_hitos</name>
4     <description/>
5     <extended_description/>
6     <trans_version/>
7     <trans_type>Normal</trans_type>
8     <trans_status>0</trans_status>
9     <directory>#47;</directory>
10    <parameters></parameters>
11    <log>
12      <trans-log-table>
13        <connection/>
14        <schema/>
15        <table/>
16        <size_limit_lines/>
17        <interval/>
18        <timeout_days/>
19        <field>
20          <id>ID_BATCH</id>
21          <enabled>Y</enabled>
22          <name>ID_BATCH</name>
23        </field>
24        <field>
25          <id>CHANNEL_ID</id>
26          <enabled>Y</enabled>
27          <name>CHANNEL_ID</name>
28        </field>
29        <field>
30          <id>TRANSNAME</id>
31          <enabled>Y</enabled>
32          <name>TRANSNAME</name>
33        </field>
34        <field>
35          <id>STATUS</id>
36          <enabled>Y</enabled>
37          <name>STATUS</name>
38        </field>
39        <field>
40          <id>LINES_READ</id>
41          <enabled>Y</enabled>
42          <name>LINES_READ</name>
43          <subject/>
44        </field>
45        <field>
46          <id>LINES_WRITTEN</id>
47          <enabled>Y</enabled>
48          <name>LINES_WRITTEN</name>
49          <subject/>
50        </field>
51        <field>
52          <id>LINES_UPDATED</id>
53          <enabled>Y</enabled>
54          <name>LINES_UPDATED</name>
```

```

55         <subject/>
56     </field>
57     <field>
58         <id>LINES_INPUT</id>
59         <enabled>Y</enabled>
60         <name>LINES_INPUT</name>
61         <subject/>
62     </field>
63     <field>
64         <id>LINES_OUTPUT</id>
65         <enabled>Y</enabled>
66         <name>LINES_OUTPUT</name>
67         <subject/>
68     </field>
69     <field>
70         <id>LINES_REJECTED</id>
71         <enabled>Y</enabled>
72         <name>LINES_REJECTED</name>
73         <subject/>
74     </field>
75     <field>
76         <id>ERRORS</id>
77         <enabled>Y</enabled>
78         <name>ERRORS</name>
79     </field>
80     <field>
81         <id>STARTDATE</id>
82         <enabled>Y</enabled>
83         <name>STARTDATE</name>
84     </field>
85     <field>
86         <id>ENDDATE</id>
87         <enabled>Y</enabled>
88         <name>ENDDATE</name>
89     </field>
90     <field>
91         <id>LOGDATE</id>
92         <enabled>Y</enabled>
93         <name>LOGDATE</name>
94     </field>
95     <field>
96         <id>DEPDATE</id>
97         <enabled>Y</enabled>
98         <name>DEPDATE</name>
99     </field>
100    <field>
101        <id>REPLAYDATE</id>
102        <enabled>Y</enabled>
103        <name>REPLAYDATE</name>
104    </field>
105    <field>
106        <id>LOG_FIELD</id>
107        <enabled>Y</enabled>
108        <name>LOG_FIELD</name>
109    </field>
110 </trans-log-table>
111 <perf-log-table>
112     <connection/>

```

113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170

```
<schema/>
<table/>
<interval/>
<timeout_days/>
<field>
  <id>ID_BATCH</id>
  <enabled>Y</enabled>
  <name>ID_BATCH</name>
</field>
<field>
  <id>SEQ_NR</id>
  <enabled>Y</enabled>
  <name>SEQ_NR</name>
</field>
<field>
  <id>LOGDATE</id>
  <enabled>Y</enabled>
  <name>LOGDATE</name>
</field>
<field>
  <id>TRANSNAME</id>
  <enabled>Y</enabled>
  <name>TRANSNAME</name>
</field>
<field>
  <id>STEPNAME</id>
  <enabled>Y</enabled>
  <name>STEPNAME</name>
</field>
<field>
  <id>STEP_COPY</id>
  <enabled>Y</enabled>
  <name>STEP_COPY</name>
</field>
<field>
  <id>LINES_READ</id>
  <enabled>Y</enabled>
  <name>LINES_READ</name>
</field>
<field>
  <id>LINES_WRITTEN</id>
  <enabled>Y</enabled>
  <name>LINES_WRITTEN</name>
</field>
<field>
  <id>LINES_UPDATED</id>
  <enabled>Y</enabled>
  <name>LINES_UPDATED</name>
</field>
<field>
  <id>LINES_INPUT</id>
  <enabled>Y</enabled>
  <name>LINES_INPUT</name>
</field>
<field>
  <id>LINES_OUTPUT</id>
  <enabled>Y</enabled>
  <name>LINES_OUTPUT</name>
```

```

171         </field>
172     <field>
173         <id>LINES_REJECTED</id>
174         <enabled>Y</enabled>
175         <name>LINES_REJECTED</name>
176     </field>
177     <field>
178         <id>ERRORS</id>
179         <enabled>Y</enabled>
180         <name>ERRORS</name>
181     </field>
182     <field>
183         <id>INPUT_BUFFER_ROWS</id>
184         <enabled>Y</enabled>
185         <name>INPUT_BUFFER_ROWS</name>
186     </field>
187     <field>
188         <id>OUTPUT_BUFFER_ROWS</id>
189         <enabled>Y</enabled>
190         <name>OUTPUT_BUFFER_ROWS</name>
191     </field>
192 </perf-log-table>
193 <channel-log-table>
194     <connection/>
195     <schema/>
196     <table/>
197     <timeout_days/>
198     <field>
199         <id>ID_BATCH</id>
200         <enabled>Y</enabled>
201         <name>ID_BATCH</name>
202     </field>
203     <field>
204         <id>CHANNEL_ID</id>
205         <enabled>Y</enabled>
206         <name>CHANNEL_ID</name>
207     </field>
208     <field>
209         <id>LOG_DATE</id>
210         <enabled>Y</enabled>
211         <name>LOG_DATE</name>
212     </field>
213     <field>
214         <id>LOGGING_OBJECT_TYPE</id>
215         <enabled>Y</enabled>
216         <name>LOGGING_OBJECT_TYPE</name>
217     </field>
218     <field>
219         <id>OBJECT_NAME</id>
220         <enabled>Y</enabled>
221         <name>OBJECT_NAME</name>
222     </field>
223     <field>
224         <id>OBJECT_COPY</id>
225         <enabled>Y</enabled>
226         <name>OBJECT_COPY</name>
227     </field>
228     <field>

```

```

229         <id>REPOSITORY_DIRECTORY</id>
230         <enabled>Y</enabled>
231         <name>REPOSITORY_DIRECTORY</name>
232     </field>
233     <field>
234         <id>FILENAME</id>
235         <enabled>Y</enabled>
236         <name>FILENAME</name>
237     </field>
238     <field>
239         <id>OBJECT_ID</id>
240         <enabled>Y</enabled>
241         <name>OBJECT_ID</name>
242     </field>
243     <field>
244         <id>OBJECT_REVISION</id>
245         <enabled>Y</enabled>
246         <name>OBJECT_REVISION</name>
247     </field>
248     <field>
249         <id>PARENT_CHANNEL_ID</id>
250         <enabled>Y</enabled>
251         <name>PARENT_CHANNEL_ID</name>
252     </field>
253     <field>
254         <id>ROOT_CHANNEL_ID</id>
255         <enabled>Y</enabled>
256         <name>ROOT_CHANNEL_ID</name>
257     </field>
258 </channel-log-table>
259 <step-log-table>
260     <connection/>
261     <schema/>
262     <table/>
263     <timeout_days/>
264     <field>
265         <id>ID_BATCH</id>
266         <enabled>Y</enabled>
267         <name>ID_BATCH</name>
268     </field>
269     <field>
270         <id>CHANNEL_ID</id>
271         <enabled>Y</enabled>
272         <name>CHANNEL_ID</name>
273     </field>
274     <field>
275         <id>LOG_DATE</id>
276         <enabled>Y</enabled>
277         <name>LOG_DATE</name>
278     </field>
279     <field>
280         <id>TRANSNAME</id>
281         <enabled>Y</enabled>
282         <name>TRANSNAME</name>
283     </field>
284     <field>
285         <id>STEPNAME</id>
286         <enabled>Y</enabled>

```

```

287         <name>STEPNAME</name>
288     </field>
289     <field>
290         <id>STEP_COPY</id>
291         <enabled>Y</enabled>
292         <name>STEP_COPY</name>
293     </field>
294     <field>
295         <id>LINES_READ</id>
296         <enabled>Y</enabled>
297         <name>LINES_READ</name>
298     </field>
299     <field>
300         <id>LINES_WRITTEN</id>
301         <enabled>Y</enabled>
302         <name>LINES_WRITTEN</name>
303     </field>
304     <field>
305         <id>LINES_UPDATED</id>
306         <enabled>Y</enabled>
307         <name>LINES_UPDATED</name>
308     </field>
309     <field>
310         <id>LINES_INPUT</id>
311         <enabled>Y</enabled>
312         <name>LINES_INPUT</name>
313     </field>
314     <field>
315         <id>LINES_OUTPUT</id>
316         <enabled>Y</enabled>
317         <name>LINES_OUTPUT</name>
318     </field>
319     <field>
320         <id>LINES_REJECTED</id>
321         <enabled>Y</enabled>
322         <name>LINES_REJECTED</name>
323     </field>
324     <field>
325         <id>ERRORS</id>
326         <enabled>Y</enabled>
327         <name>ERRORS</name>
328     </field>
329     <field>
330         <id>LOG_FIELD</id>
331         <enabled>N</enabled>
332         <name>LOG_FIELD</name>
333     </field>
334 </step-log-table>
335 </log>
336 <maxdate>
337     <connection/>
338 <table/>
339 <field/>
340 <offset>0.0</offset>
341 <maxdiff>0.0</maxdiff>
342 </maxdate>
343 <size_rowset>10000</size_rowset>
344 <sleep_time_empty>50</sleep_time_empty>

```

```

345     <sleep_time_full>50</sleep_time_full>
346     <unique_connections>N</unique_connections>
347     <feedback_shown>Y</feedback_shown>
348     <feedback_size>50000</feedback_size>
349     <using_thread_priorities>Y</using_thread_priorities>
350     <shared_objects_file/>
351     <capture_step_performance>N</capture_step_performance>
352     <step_performance_capturing_delay>1000
353         </step_performance_capturing_delay>
354     <step_performance_capturing_size_limit>100
355         </step_performance_capturing_size_limit>
356     <dependencies></dependencies>
357     <partitionschemas></partitionschemas>
358     <slaveservers></slaveservers>
359     <clusterschemas></clusterschemas>
360     <created_user>--</created_user>
361     <created_date>2012&#47;09&#47;06 12:04:05.323
362         </created_date>
363     <modified_user>--</modified_user>
364     <modified_date>2012&#47;09&#47;06 12:20:41.039
365         </modified_date>
366 </info>
367 <notepads>
368 </notepads>
369 <connection>
370     <name>mudel</name>
371     <server>localhost </server>
372     <type>MYSQL</type>
373     <access>Native</access>
374     <database>moodle</database>
375     <port>3306</port>
376     <username>moodle</username>
377     <password>Encrypted 2e7fad9850be9c1869b2fad76e9b487ee
378         </password>
379     <servername/>
380     <data_tablespace/>
381     <index_tablespace/>
382     <attributes>
383         <attribute>
384             <code>PORT_NUMBER</code>
385             <attribute>3306</attribute>
386         </attribute>
387     </attributes>
388 </connection>
389     <order>
390     <hop>
391         <from>query </from>
392         <to>milestones </to>
393         <enabled>Y</enabled>
394     </hop>
395     <hop>
396         <from>percentage </from>
397         <to>indicador_hitos </to>
398         <enabled>Y</enabled>
399     </hop>
400     <hop>
401         <from>milestones </from>
402         <to>max </to>

```

```

403         <enabled>Y</enabled>
404     </hop>
405     <hop>
406         <from>milestones </from>
407         <to>join </to>
408         <enabled>Y</enabled>
409     </hop>
410     <hop>
411         <from>max</from>
412         <to>join </to>
413         <enabled>Y</enabled>
414     </hop>
415     <hop>
416         <from>join </from>
417         <to>percentage </to>
418         <enabled>Y</enabled>
419     </hop>
420 </order>
421 <step>
422     <name>query </name>
423     <type>TableInput </type>
424     <description/>
425     <distributed>Y</distributed>
426     <copies>1</copies>
427     <partitioning>
428         <method>none</method>
429         <schema_name/>
430     </partitioning>
431
432     <connection>mudel</connection>
433     <sql>SELECT u.id , u.username , u.firstname , u.lastname ,
434         IF ((a.duedate - e.timemodified)
435             &gt; 0,1,0) milestone
436     FROM mdl_user u, mdl_context ct ,
437         mdl_course c, mdl_role r ,
438         mdl_role_assignments ra , mdl_assign a ,
439         mdl_assign_submission e
440     WHERE  c.id = 2
441         AND r.id = 5 AND ra.userid = u.id
442         AND ct.id = ra.contextid
443         AND c.id = ct.instanceid
444         AND r.id = ra.roleid
445         AND u.id = e.userid
446         AND a.id = e.assignment
447     ORDER BY u.id , u.username , u.firstname ,
448             u.lastname
449     </sql>
450     <limit>0</limit>
451     <lookup/>
452     <execute_each_row>N</execute_each_row>
453     <variables_active>N</variables_active>
454     <lazy_conversion_active>N</lazy_conversion_active>
455     <cluster_schema/>
456     <remotesteps>
457         <input> </input>
458         <output> </output>
459     </remotesteps>
460     <GUI>

```

```

461         <xloc>70</xloc>
462         <yloc>150</yloc>
463         <draw>Y</draw>
464     </GUI>
465
466     </step>
467     <step>
468         <name>milestones </name>
469         <type>GroupBy</type>
470         <description/>
471         <distributed>N</distributed>
472         <copies>1</copies>
473         <partitioning>
474             <method>none</method>
475             <schema_name/>
476         </partitioning>
477
478         <all_rows>N</all_rows>
479         <ignore_aggregate>N</ignore_aggregate>
480         <field_ignore/>
481         <directory>%java.io.tmpdir%</directory>
482         <prefix>grp</prefix>
483         <add_linetr>N</add_linetr>
484         <linetr_fieldname/>
485         <give_back_row>N</give_back_row>
486         <group>
487             <field>
488                 <name>id</name>
489             </field>
490             <field>
491                 <name>username</name>
492             </field>
493             <field>
494                 <name>first name</name>
495             </field>
496             <field>
497                 <name>last name</name>
498             </field>
499         </group>
500         <fields>
501             <field>
502                 <aggregate>Milestones</aggregate>
503                 <subject>milestone</subject>
504                 <type>COUNT_ALL</type>
505                 <valuefield/>
506             </field>
507             <field>
508                 <aggregate>In-time</aggregate>
509                 <subject>milestone</subject>
510                 <type>SUM</type>
511                 <valuefield/>
512             </field>
513         </fields>
514         <cluster_schema/>
515         <remotesteps>
516             <input> </input>
517             <output> </output>
518         </remotesteps> <GUI>

```

```

519         <xloc>190</xloc>
520             <yloc>150</yloc>
521         <draw>Y</draw>
522     </GUI>
523
524 </step>
525 <step>
526     <name>max</name>
527     <type>GroupBy</type>
528     <description/>
529     <distributed>Y</distributed>
530     <copies>1</copies>
531     <partitioning>
532         <method>none</method>
533         <schema_name/>
534     </partitioning>
535
536     <all_rows>N</all_rows>
537     <ignore_aggregate>N</ignore_aggregate>
538     <field_ignore/>
539     <directory>%java.io.tmpdir%/</directory>
540     <prefix>grp</prefix>
541     <add_liners>N</add_liners>
542     <liners_fieldname/>
543     <give_back_row>N</give_back_row>
544     <group> </group>
545     <fields>
546         <field>
547             <aggregate>Max</aggregate>
548             <subject>Milestones</subject>
549             <type>MAX</type>
550             <valuefield/>
551         </field>
552     </fields>
553     <cluster_schema/>
554 <remotesteps>
555     <input> </input>
556     <output> </output>
557 </remotesteps>
558     <GUI>
559         <xloc>190</xloc>
560             <yloc>30</yloc>
561         <draw>Y</draw>
562     </GUI>
563
564 </step>
565 <step>
566     <name>join</name>
567     <type>JoinRows</type>
568     <description/>
569     <distributed>Y</distributed>
570     <copies>1</copies>
571     <partitioning>
572         <method>none</method>
573         <schema_name/>
574     </partitioning>
575
576     <directory>%java.io.tmpdir%/</directory>

```

```

577         <prefix>out</prefix>
578         <cache_size>500</cache_size>
579         <main/>
580         <compare>
581     <condition>
582         <negated>N</negated>
583         <left value/>
584         <function>=</function>
585         <right value/>
586     </condition>
587     </compare>
588     <cluster_schema/>
589     <remotesteps>
590         <input>     </input>
591         <output>    </output>
592     </remotesteps>
593 <GUI>
594     <xloc>310</xloc>
595     <yloc>150</yloc>
596     <draw>Y</draw>
597 </GUI>
598 </step>
599 <step>
600     <name>percentage</name>
601     <type>Calculator</type>
602     <description/>
603     <distributed>Y</distributed>
604     <copies>1</copies>
605     <partitioning>
606         <method>none</method>
607         <schema_name/>
608     </partitioning>
609     <calculation <field_name>Pending</field_name>
610         <calc_type>SUBTRACT</calc_type>
611         <field_a>Max</field_a>
612         <field_b>Milestones</field_b>
613         <field_c/>
614         <value_type>Integer</value_type>
615         <value_length>-1</value_length>
616         <value_precision>-1</value_precision>
617         <remove>N</remove>
618         <conversion_mask/>
619         <decimal_symbol>.</decimal_symbol>
620         <grouping_symbol/>
621         <currency_symbol/>
622     </calculation>
623     <calculation>
624         <field_name>Percentage</field_name>
625         <calc_type>PERCENT_1</calc_type>
626         <field_a>In-time</field_a>
627         <field_b>Max</field_b>
628         <field_c/>
629         <value_type>Number</value_type>
630         <value_length>-1</value_length>
631         <value_precision>-1</value_precision>
632         <remove>N</remove>
633         <conversion_mask/>
634         <decimal_symbol>.</decimal_symbol>

```

```

635         <grouping_symbol/>
636         <currency_symbol/>
637         </calculation><<cluster_schema/>
638     <remotesteps>
639         <input> </input>
640         <output> </output>
641     </remotesteps>
642     <GUI>
643         <xloc>430</xloc>
644         <yloc>150</yloc>
645         <draw>Y</draw>
646     </GUI>
647 </step>
648 <step>
649     <name>indicador_hitos</name>
650     <type>XMLOutput</type>
651     <description/>
652     <distributed>Y</distributed>
653     <copies>1</copies>
654     <partitioning>
655         <method>none</method>
656         <schema_name/>
657     </partitioning>
658
659     <encoding>UTF-8</encoding>
660     <name_space/>
661     <xml_main_element>Rows</xml_main_element>
662     <xml_repeat_element>Row</xml_repeat_element>
663     <file>
664         <name>${Internal.Transformation.FileName.Directory}
665             &#47;..&#47;output&#47;indicador_hitos
666         </name>
667         <extention>xml</extention>
668         <servlet_output>N</servlet_output>
669         <do_not_open_newfile_init>N
670             </do_not_open_newfile_init>
671         <split>N</split>
672         <add_date>N</add_date>
673         <add_time>N</add_time>
674         <SpecifyFormat>N</SpecifyFormat>
675         <omit_null_values>N</omit_null_values>
676         <date_time_format/>
677         <add_to_result_filenames>N
678             </add_to_result_filenames>
679         <zipped>N</zipped>
680         <splitevery>0</splitevery>
681     </file>
682     <fields>
683         <field>
684             <name>id</name>
685             <element/>
686             <type>Integer</type>
687             <format/>
688             <currency/>
689             <decimal/>
690             <group/>
691             <nullif/>
692             <length>15</length>

```

693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750

```
<precision>0</precision>
</field>
<field>
  <name>username</name>
  <element/>
  <type>String</type>
  <format/>
  <currency/>
  <decimal/>
  <group/>
  <nullif/>
  <length>100</length>
  <precision>-1</precision>
</field>
<field>
  <name>first name</name>
  <element/>
  <type>String</type>
  <format/>
  <currency/>
  <decimal/>
  <group/>
  <nullif/>
  <length>100</length>
  <precision>-1</precision>
</field>
<field>
  <name>last name</name>
  <element/>
  <type>String</type>
  <format/>
  <currency/>
  <decimal/>
  <group/>
  <nullif/>
  <length>100</length>
  <precision>-1</precision>
</field>
<field>
  <name>Percentage</name>
  <element/>
  <type>Number</type>
  <format/>
  <currency/>
  <decimal>.</decimal>
  <group/>
  <nullif/>
  <length>-1</length>
  <precision>-1</precision>
</field>
<field>
  <name>Milestones</name>
  <element/>
  <type>Integer</type>
  <format/>
  <currency/>
  <decimal/>
  <group/>
```

```

751         <nullif/>
752         <length>10</length>
753         <precision>0</precision>
754     </field>
755     <field>
756         <name>In-time</name>
757         <element/>
758         <type>Integer</type>
759         <format/>
760         <currency/>
761         <decimal/>
762         <group/>
763         <nullif/>
764         <length>-1</length>
765         <precision>0</precision>
766     </field>
767     <field>
768         <name>Pending</name>
769         <element/>
770         <type>Number</type>
771         <format/>
772         <currency/>
773         <decimal/>
774         <group/>
775         <nullif/>
776         <length>-1</length>
777         <precision>0</precision>
778     </field></fields>
779 </cluster_schema/>
780 <remotesteps>
781     <input> </input>
782     <output> </output>
783 </remotesteps>
784 <GUI>
785     <xloc>550</xloc>
786     <yloc>150</yloc>
787     <draw>Y</draw>
788 </GUI>
789 </step>
790 </transformation>

```

Listing 1: Estructura XML Pentaho

## .2. Contenido del fichero EvalCourse.ecore

Código fuente del fichero EvalCourse.ecore

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ecore:EPackage xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xmlns:ecore="http://www.eclipse.org/emf/2002/Ecore" name="evalcourse"
5   nsURI="http://evalcourse/1.0" nsPrefix="evalcourse">
6   <eClassifiers xsi:type="ecore:EClass" name="Evidence">
7     <eStructuralFeatures xsi:type="ecore:EReference" name="target"
8       lowerBound="1" eType=":///Target" containment="true"/>
9     <eStructuralFeatures xsi:type="ecore:EAttribute" name="name"
10      eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore
11      #//EString"/>
12     <eStructuralFeatures xsi:type="ecore:EReference" name="indicator"
13       lowerBound="1" eType=":///Indication" containment="true"/>
14     <eStructuralFeatures xsi:type="ecore:EReference" name="dbconnection"
15       eType=":///Connection" containment="true"/>
16   </eClassifiers>
17   <eClassifiers xsi:type="ecore:EClass" name="Target">
18     <eStructuralFeatures xsi:type="ecore:EAttribute" name="operator"
19       eType=":///Operator"/>
20     <eStructuralFeatures xsi:type="ecore:EReference" name="students"
21       upperBound="-1" eType=":///Student" containment="true"/>
22   </eClassifiers>
23   <eClassifiers xsi:type="ecore:EClass" name="Indication">
24     <eStructuralFeatures xsi:type="ecore:EAttribute" name="type"
25       lowerBound="1" eType=":///Sign"/>
26     <eStructuralFeatures xsi:type="ecore:EAttribute" name="tool"
27       lowerBound="1" eType=":///Tool"/>
28     <eStructuralFeatures xsi:type="ecore:EAttribute" name="date_ini"
29       eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore
30       #//EDate"/>
31     <eStructuralFeatures xsi:type="ecore:EAttribute" name="date_end"
32       eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore
33       #//EDate"/>
34     <eStructuralFeatures xsi:type="ecore:EReference" name="entities"
35       upperBound="-1" eType=":///Entity" containment="true"/>
36   </eClassifiers>
37   <eClassifiers xsi:type="ecore:EEnum" name="Operator">
38     <eLiterals name="get"/>
39     <eLiterals name="count" value="1"/>
40   </eClassifiers>
41   <eClassifiers xsi:type="ecore:EClass" name="Connection">
42     <eStructuralFeatures xsi:type="ecore:EAttribute" name="name"
43       eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore
44       #//EString"/>
45     <eStructuralFeatures xsi:type="ecore:EAttribute" name="server"
46       eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore
47       #//EString"/>
48     <eStructuralFeatures xsi:type="ecore:EAttribute" name="type"
49       eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore
50       #//EString"/>
51     <eStructuralFeatures xsi:type="ecore:EAttribute" name="access"
52       eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore
53       #//EString"/>
54     <eStructuralFeatures xsi:type="ecore:EAttribute" name="database"
```

```

55         eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore
56         #//EString"/>
57     <eStructuralFeatures xsi:type="ecore:EAttribute" name="port"
58         eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore
59         #//EInt"/>
60     <eStructuralFeatures xsi:type="ecore:EAttribute" name="username"
61         eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore
62         #//EString"/>
63     <eStructuralFeatures xsi:type="ecore:EAttribute" name="password"
64         eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore
65         #//EString"/>
66 </eClassifiers>
67 <eClassifiers xsi:type="ecore:EEnum" name="Sign">
68     <eLiterals name="grade"/>
69     <eLiterals name="participation" value="1"/>
70     <eLiterals name="milestones" value="2"/>
71     <eLiterals name="access" value="3"/>
72     <eLiterals name="assessment" value="4"/>
73 </eClassifiers>
74 <eClassifiers xsi:type="ecore:EEnum" name="Tool">
75     <eLiterals name="assignment"/>
76     <eLiterals name="forum" value="1"/>
77     <eLiterals name="campus" value="2"/>
78     <eLiterals name="workshop" value="3"/>
79 </eClassifiers>
80 <eClassifiers xsi:type="ecore:EClass" name="Entity">
81     <eStructuralFeatures xsi:type="ecore:EAttribute" name="id"
82         eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore#//EInt"/>
83 </eClassifiers>
84 <eClassifiers xsi:type="ecore:EClass" name="Student">
85     <eStructuralFeatures xsi:type="ecore:EAttribute" name="id"
86         eType="ecore:EDatatype http://www.eclipse.org/emf/2002/Ecore#//EInt"/>
87 </eClassifiers>
88 </ecore:EPackage>

```

Listing 2: Fichero Ecore XMI

### .3. GNU General Public License

Copyright © 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## TERMS AND CONDITIONS

### 0. Definitions.

“This License” refers to version 3 of the GNU General Public License.

“Copyright” also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

“The Program” refers to any copyrightable work licensed under this License. Each licensee is addressed as “you”. “Licensees” and “recipients” may be individuals or organizations.

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

To “propagate” a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To “convey” a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays “Appropriate Legal Notices” to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

### 1. Source Code.

The “source code” for a work means the preferred form of the work for making modifications to it. “Object code” means any non-source form of a work.

A “Standard Interface” means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The “System Libraries” of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A “Major Component”, in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The “Corresponding Source” for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the

work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.

## 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

## 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

## 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices

of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

#### 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to “keep intact all notices”.
- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation’s users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

#### 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.
- b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than

- your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.
- c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.
  - d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.
  - e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A “User Product” is either (1) a “consumer product”, which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, “normally used” refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

“Installation Information” for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install

modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## 7. Additional Terms.

“Additional permissions” are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered “further restrictions” within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

#### 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

#### 9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

#### 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An “entity transaction” is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If

propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

## 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is “discriminatory” if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

12. No Surrender of Others’ Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

15. Disclaimer of Warranty.

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

## END OF TERMS AND CONDITIONS

### How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively state the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.>

Copyright (C) <textyear> <name of author>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

```
<program> Copyright (C) <year> <name of author>
```

```
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.  
This is free software, and you are welcome to redistribute it  
under certain conditions; type 'show c' for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <http://www.gnu.org/licenses/>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <http://www.gnu.org/philosophy/why-not-lgpl.html>.



# A generative computer language to customize online learning assessments

Antonio Balderas, Ivan Ruiz-Rube, Juan M. Dodero, Manuel Palomo-Duarte,  
and Anke Berns

University of Cádiz, Spain, Cádiz, C/Chile 1, CP 11002  
antonio.balderas@uca.es, ivan.ruiz@uca.es, juanma.dodero@uca.es,  
manuel.palomo@uca.es, anke.berns@uca.es

**Abstract.** The focus on assessment of learning experiences has shifted from knowledge to competences. Unfortunately, assessing certain competences is mainly a subjective task, being problematic for both the evaluators and the evaluated. Additionally, when the learning process is computer-supported and the number of students increases, traditional assessment procedures suffer from scalability problems. In this paper we propose a query language that supports grading learning competences according to students' performance in an online course. Using it we automatically extract different objective indicators about students work in a Learning Management System (LMS). Evaluators can use this computer programming-like language to express a number of required indicators. Such indicators are automatically obtained from the activity logs generated by the LMS.

**Keywords:** online learning, competence assessment, technological support in online education, domain-specific languages, learning analytics

## 1 Introduction

The scope of our work is related to the virtual places specially designed for collaborative tasks on LMS, which they are widely used nowadays. Each file, access or assignment done by a student is registered in the system [1]. This collected information, properly filtered [2], would be used as indicators of students' work.

## 2 EvalCourse

EvalCourse is the domain-specific language [4] that we have developed to get indicators. Language syntax and reserved words are listed in 1.1. The first line specifies the name of the evidence or indicator. The second line is a constant that indicates that we want to get information about students. The third one indicates the kind of information to be retrieved: milestones, participation or access. And the last one if it is collected from assignments, forums or campus activity. In case one or more activity identifiers are provided ([list of ids]), only information on those assignment, forum or campus will be provided. If not specified, the query acts on all activities of the indicated type.

```

1 Evidence name_of_the_indicator :
2     get students
3     show milestones | participation | access
4     in assignment | forum | campus [list of ids].

```

**Listing 1.1.** Language syntax. Reserved words are highlighted in bold format

We will illustrate language usage with a simple example on assessing problem solving competence. We can assess students' performance in this competence using their participation in forums [3] in which lecturers set out problems. We ask for the information with the code shown in 1.2, obtaining results in an XML file that can be imported in most popular spreadsheets.

```

1 Evidence Participation_list :
2     get students
3     show participation in forum.

```

**Listing 1.2.** Code to retrieve students' performance in forum participation

### 3 Conclusions and future work

Using our system, the lecturer, without any technical knowledge on system programming and using a very simple syntax, can automatically assess the performance in several competences for any student, justifying it with objective figures. At the moment, our system only works with an instance of Moodle, so our future work is implement support for other LMS.

### Acknowledgements

This work has been funded by the *PAIDI* programme of the Andalusian Government, ASCETA project (P09-TIC-5230) and University of Cadiz programme for Researching and Innovation in Education.

### References

1. H. Chebil, J. . Girardot, and C. Courtin. An ontology-based approach for sharing and analyzing learning trace corpora. In *Proceedings - IEEE 6th International Conference on Semantic Computing, ICSC 2012*, pages 101–108, 2012.
2. B. Florian, C. Glahn, H. Drachler, M. Specht, and R. Fabregat Gesa. *Activity-based learner-models or learner monitoring and recommendations in moodle*, volume 6964 LNCS of *Lecture Notes in Computer Science*. 2011.
3. S. M. Jacob, B. Lee, and G. R. Lueckenhausen. Measuring critical thinking skills in engineering mathematics using online forums. *2009 International Conference on Engineering Education, ICEED2009 - Embracing New Challenges in Engineering Education*, pages 225–229, 2009.
4. Arie van Deursen, Paul Klint, and Joost Visser. Domain-specific languages: an annotated bibliography. *SIGPLAN Not.*, 35(6):26–36, June 2000.

# A generative computer language to customize online learning assessments

Antonio Balderas  
University of Cádiz  
Spain, Cádiz, C/Chile 1, CP  
11002  
+34 956 015 692  
antonio.balderas@uca.es

Ivan Ruiz-Rube  
University of Cádiz  
Spain, Cádiz, C/Chile 1, CP  
11002  
+34 956 015 784  
ivan.ruiz@uca.es

Juan M. Dodero  
University of Cádiz  
Spain, Cádiz, C/Chile 1, CP  
11002  
+34 956 015 784  
juanma.dodero@uca.es

Manuel Palomo-Duarte  
University of Cádiz  
Spain, Cádiz, C/Chile 1, CP  
11002  
+34 956 015 483  
manuel.palomo@uca.es

## ABSTRACT

The focus on assessment of learning experiences has shifted from knowledge to competences. Unfortunately, assessing certain competences is mainly a subjective task, being problematic for both the evaluators and the evaluated. Additionally, when the learning process is computer-supported and the number of students increases, traditional assessment procedures suffer from scalability problems. In this paper we propose a methodology that supports grading learning competences according to students' performance in an online course. We automatically extract different objective indicators about students' work in a Learning Management System (LMS). Evaluators can use a computer programming-like language to express a number of required indicators. Such indicators are automatically extracted from the activity logs generated by the LMS. A case study with Moodle LMS-based courses is carried out to explain how such indicators can be obtained and how to analyze the assessment results.

## Categories and Subject Descriptors

K.3.1 [COMPUTERS AND EDUCATION]: Computer Uses in Education—*Computer-assisted instruction (CAI)*

## General Terms

Measurement, Performance, Design, Languages.

## Keywords

online learning, competence assessment, technological support in online education, domain-specific languages, learning analytics

## 1. INTRODUCTION

The interest on higher education learning has shifted from knowledge to competences. Although the importance of building-up and developing subject-specific knowledge and skills as the basis for university degree programs is widely acknowledged, it is a fact that time and attention should also be devoted to the development of generic competences or transferable skills. They are becoming more and more relevant for preparing students well for their future role in society in terms of employability and citizenship.

Unfortunately, assessing certain competences is usually a subjective task, being problematic for both evaluators and evaluated. Unless the competences were clearly linked to a specific assignment or aspect, they are difficult to grasp. Developing a detailed process for assessing students' performance in competences is a complex and time consuming task because of the different aspects to take into account. As a result, lecturers usually do not find enough time for explaining the whole established subjects, even less with other evaluation tasks to be assessed. Even supposing that students can perform a number of distinct assignments, the solely observation does not guarantee that they have got the competences specified in the syllabus. Therefore, here there is a major gap.

The scope of our work is related to the virtual places specially designed for collaborative tasks on LMS, which they are widely used nowadays. Each file, access or assignment done by a student is registered in the system. Usually researchers can analyze Technology Enhanced Learning (TEL) situations by collecting interaction traces produced by these environments [5]. However, taking advantage of this information stored about course activities requires filtering before it can be used for higher level processing [7]. This collected information would be used as indicators of students' work.

Indicators are statements that determine if the competence level is met. They have been designed so that every student will be able to demonstrate their competence regardless of their grade. As we see later, we have defined several indica-

tors to measure different competences.

We have developed a generative computer language to customize online learning assessments in Moodle. In technical terms, it is a Domain-Specific Language (DSL) [13], a formal language that can be automatically processed to execute simple sentences written using an assessment vocabulary. This language has a simple syntax, oriented to learning assessment. We implemented *EvalCourse*, a computer system that runs queries written in this language, providing in its output the information requested. This way, lecturers can easily retrieve indicators from information in the Moodle logs without any technical knowledge on databases or computer programming.

The rest of this paper is organized as follows: the second section describes the indicators and how they are connected to competences. The third section introduces EvalCourse, the domain specific language that we have used to work in the case study we will show in the fourth section with an analysis of the results. The fifth section comments related works. In the last one, we provide a discussion along with conclusions and future research lines.

## 2. INDICATORS FOR COMPETENCE ASSESSMENT

In order to obtain a meaningful data collection about student learning outcomes we need to focus on student achievement of the specific performance indicators that define the course program outcome [8]. Through experience, a lecturer can know that there are certain characteristics that can be considered as reliable indicators that a certain student has achieved a specific learning outcome. As the initial step in the present study we have detected several specific performance criteria that can be used as evidence of program learning outcomes.

*Planning and time management* [4] is the student's ability to manage their time successfully and productively, and it is explicitly related to academic performance. In the rest of the paper we refer to this competence as "Planning".

Students must submit the activities before a deadline defined by the lecturer. But LMS usually accepts them submissions after this deadline expired. We consider each one of these submissions as a milestone. At the end of the course we can measure the amount of milestones that have been submitted in time, and the amount of them that have been submitted out of time.

*Teamwork* [9] is an assignment done by several students with each doing a part that must be coordinated with that of the other members. In many cases, the tasks they have to do have interdependent. Thus, a student could had to wait that her teammate finishes her previous task so she can start hers. In this case, if that student does not finish the milestone in time, she can affect the whole group. In a LMS like Moodle we can use the deadline of the interdependent activities to set when the different teammates have to submit their tasks.

*Interpersonal skills* are the life skills we use every day to communicate and interact with other people, individually and in groups. Communication, interpersonal and decision-making skill are essential for engineering work and should be explicitly incorporated in engineering curricula [1]. We use the student participation in forums as an indicator of performance in interpersonal skills.

*Leadership* is organizing a team to achieve a common goal. Its development has become increasingly important in engineering education as companies looking for engineers with leadership, communication, and teamwork skills [11]. We can take the number of initiated debates as a indicator of the leadership competence.

## 3. EVALCOURSE

EvalCourse is the domain-specific language that we have developed to get indicators. It follows a model-driven approach [12] for modeling procedures in order to obtaining performance indicators of competences. EvalCourse<sup>1</sup> is provided freely as open-source software under the terms of the GNU GPL.

In Figure 1 we can see the EvalCourse diagram. The user requests information about an indicator (a), then EvalCourse extracts the necessary information from the LMS database (b) to meet the request for indicators. This information is collected using Pentaho libraries (c) and an intermediate Pentaho transformation (d). Finally it returns the data in two formats (e). On the one hand, the raw data in an XML file that can be processed with a analytics software, and on the other hand, a graph on a specific file for the Gephi software. Gephi is an open-source software for visualizing and analyzing large networks graphs. It must be installed to view the charts.

Language syntax and reserved words are listed in 1. The first line specifies the name of the evidence or indicator (name\_of\_the\_indicator). It will be used to name the different files produced by EvalCourse. The second line is a constant that indicates that we want to get information about students. The third one indicates the kind of information to be retrieved: milestones, participation or access. And the last one if it is collected from assignments, forums or campus activity. In case one or more activity identifiers are provided ([list of ids]), only information on those assignment, forum or campus will be provided. If not specified, the query acts on all activities of the indicated type. Bellow, two examples using synthetic information will be described.

```
1 Evidence name_of_the_indicator :
2   get students
3   show milestones | participation
4   | access
5   in assignment | forum
6   | campus [list of ids].
```

Listing 1: Reserved words and format

EvalCourse works as a Eclipse plugin. You have to create an EvalCourse project. Then you write a query in a file with

<sup>1</sup><https://www.assembla.com/spaces/evalcourse/wiki>

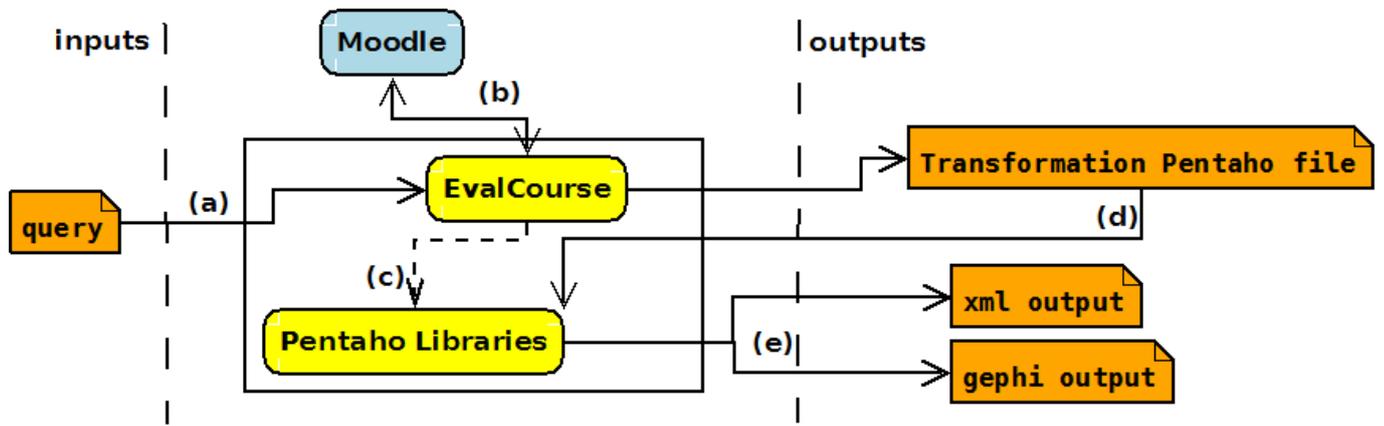


Figure 1: EvalCourse scheme.

an EVC extension. When you save this file, EvalCourse transforms the query in a Pentaho Extract, Transform and Load (ETL) process. From the program itself can execute the transformation and get the results to the query.

We will illustrate language usage with a simple example on assessing Planning competence performance of different students along a course. Using EvalCourse, we have to ask for the information with the code showed in listing 2. The evaluator declares an evidence called *Milestones.list*, that retrieves the indicator about students' performance in assignments by writing this code. The produces the information shown in table 1.

```

1 Evidence Milestone_list :
2     get students
3     show milestones
4     in assignment .

```

Listing 2: Code to retrieve students' performance in assignments

The first row of table 1 starts with two columns identifying the student by her Moodle internal identifier and her name. Then, the *Total* column indicates the number of activities that students had to submit during the course. The *In-time* column indicates that Student1 submitted three activities before their deadline; *Overdue* column indicates that she submitted no activities after deadline and *Pending* indicates that two activities had not been submitted yet. Finally, as we can work with this data in a spreadsheet, we added a new derived column: *In-time Rate*, which indicates the ratio of completed activities in time in relation to the total.

On the other hand, we are also provided a graph (figure 2). The central vertex represents the activity we are considering. In this case, activity is the delivery of assignments by students. Each other vertex represents a student enrolled in the course. Each student is linked to the central vertex activity. The thickness of the link indicates students' performance in the indicator. A visual representation gives an overview of the results that can support the interpretation of the data. In the example, the link of the first student is thinner than that of the rest. That means that her perfor-

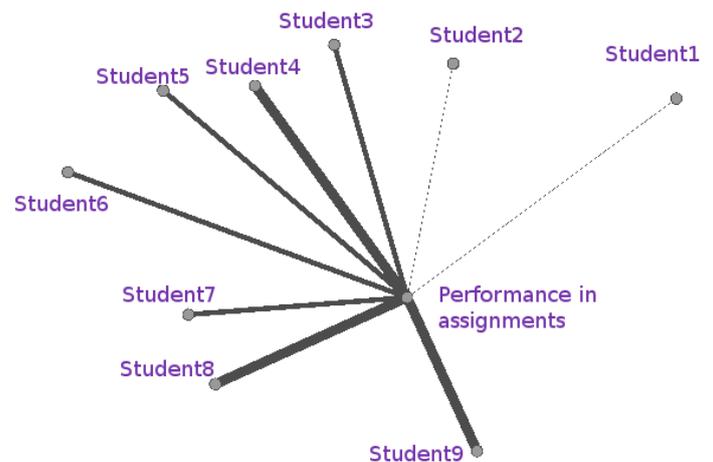


Figure 2: Graph of students' performance in assignments.

mance was lower than that of the other students.

## 4. CASE STUDY

The case study we analyze was developed in University of Cadiz, Spain. In particular, in a compulsory course on Language Processors II of its degree on Computer Science. 36 students enrolled the course in 2012/13 academic year. The course was developed in its first semester of the fifth (and last) year. The course evaluation was done manually and we later applied EvalCourse. This course was taught by one of the EvalCourse project member.

Student had several compulsory tasks over the course and an exam at the end of the semester. They provide a grade about students' knowledge. We use EvalCourse to get indicators that assist in competence assessment. Next, we describe the extracted indicators using EvalCourse.

### 4.1 Information retrieved

During the semester, students had to work in small teams of two or three members. Every team had to carry out several milestones during the course. These milestones were parts of a software development project that they had to code

**Table 1: List of students' performance in assignments.**

id	username	Total	In-time	Overdue	Pending	In-time Rate
1	Student1	5	3	0	2	60%
2	Student2	5	1	0	4	20%
3	Student3	5	3	2	0	60%
4	Student4	5	3	2	0	60%
5	Student5	5	3	2	0	60%
6	Student6	5	3	1	1	60%
7	Student7	5	3	2	0	60%
8	Student8	5	3	2	0	60%
9	Student9	5	3	2	0	60%

and upload into the appropriate activity. Every milestone had a deadline, although system allowed students to submit these activities after this deadline. For each team, only one member had to upload each activity. For example, if we have a team with members S1, S2 and S3, they can submit the activities distributing them as desired: S1 submits the five handouts, but S2 and S3 do not submit anything; or S1 submits the first handout, S2 submits the second one and S3 the rest, etc. What is important is that all the activities are submitted in each team, no matter who does it. All the teammates will share the team grade.

The competences we wanted to assess were Planning and Teamwork, and for this purpose all the activities that had been submitted in time add up. Those teams who submitted all their milestones in time and submitted exactly the five activities were evaluated in the competences with a positive grade. If a team submitted exactly five activities but not in time, then they were evaluated in Teamwork with a positive grade but not in Planning. If they submitted all milestones in time but more than five activities (because more than one teammate uploaded the same activity), they were evaluated positively in Planning but negatively in Teamwork.

## 4.2 Analysis of results

In Table 2 we can see the final result of teams' performance in competences according to indicators. There are eight teams that obtained the positive mark in both Planning and Teamwork competences, because they submitted all the milestones in time and they made exactly five submissions. There are five other teams that have a negative mark in both competences, because they did not submit all the activities.

Informally, the course supervisor confirmed that these indicators were closed to the grades he manually calculated for every student in the course.

## 5. RELATED WORKS

There are some works related to skills development. For instance, a module that helps in working and assessing the competence of teamwork is presented in [10]. Since EvalCourse works as a standalone tool, this module is integrated into Moodle. However, although EvalCourse only works with an instance of Moodle, is devised to be implemented in others LMS.

Other related work is described in [6], it uses a DSL for extracting data models out of databases. This work is focus

on the extraction of data from Web 2.0 systems, like MediaWiki wikis or Blojsom blogs. While Evalcourse provides refined reports, this proposal retrieves a model for further processing.

DSL is an approach that can be used for areas that are not connected to computer science. A DSL based on agile method adapted to the management process model of distance learning courses within the context of the Open University of Brazil is presented in [3].

The Social Networks Adapting Pedagogical Practice (SNAPP) tool was developed to provide instructors with the capacity to visualize the evolution of participant relationships within discussions forums. SNAPP 2.0 [2] shows the evolution of participant interaction over time and annotate key events that occur along this timeline. EvalCourse generates other types of displays without having to program a new application for visual analysis.

## 6. CONCLUSIONS AND FUTURE RESEARCH

In education, the focus of assessment of learning experiences has shifted from knowledge to competences. Competences are the practice, skills and attributes we can develop in every aspect of our life. The development of such key competences should be integrated through curricula, along with specific competences.

Assessing certain competences is usually a subjective task. Specially when the number of students in the course increases. When using a LMS, researchers can analyze interaction traces produced by the environment. LMS stores information about course activities that requires filtering before it can be used for higher level processing. This information will be used as indicators of students' work.

We have developed and applied EvalCourse in our case study. We assessed students' performance in the competences of Planning and Teamwork. We automatically obtained indicators that assisted as in the assessment of these competences. The interpretation that we did about indicators were closely related to the results that the lecturer noticed during the course. Of course, this is just a case study, needing further study to get a strong conclusion on the validity of the specific usage of the information retrieved by EvalCourse.

We have two future lines of work ahead. First, we will iden-

**Table 2: List of teams' performance in assignments in case study.**

Team	Total	In-time	Overdue	Pending	Planning	Teamwork
team01	5	5	0	0	+	+
team02	5	2	1	2	-	-
team03	5	4	1	0	+	+
team04	5	5	0	0	+	+
team05	5	4	1	0	+	+
team06	5	1	1	3	-	-
team07	5	5	0	0	+	+
team08	5	2	2	1	-	-
team09	5	4	1	0	+	+
team10	5	5	0	0	+	+
team11	5	2	1	2	-	-
team12	5	6	1	0	+	-
team13	5	4	0	1	-	-
team14	5	4	1	0	+	+

tify others indicators stored in LMS that can be used to assess competences, and include them in our syntax. Secondly, our language can express sentences that are general for almost any LMS. But at the moment, the generative computer language just allows lecturers to work with Moodle. So we will expand our generative computer language to be used in others LMS.

## 7. ACKNOWLEDGEMENTS

This work has been funded by the PAIDI programme of the Andalusian Government, ASCETA project (P09-TIC-5230) and University of Cadiz programme for Researching and Innovation.

## 8. REFERENCES

- [1] C. Alvarez, M. Nussbaum, M. Recabarren, F. Gomez, and D. Radovic. Teaching communication, interpersonal and decision-making skills in engineering courses supported by technology. *International Journal of Engineering Education*, 25(4):655–664, 2009.
- [2] Aneesha Bakharia and Shane Dawson. Snapp: a bird's-eye view of temporal participant interaction. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge, LAK '11*, pages 168–173, New York, NY, USA, 2011. ACM.
- [3] A. A. M. Batista, Z. Abdelouahab, D. Lopes, and P. S. Neto. *A process model for supporting the management of distance learning courses through an agile approach*, volume 152 LNEE of *Lecture Notes in Electrical Engineering*. 2013.
- [4] Bruce K Britton and Abraham Tesser. Effects of time-management practices on college grades. *Journal of Educational Psychology*, 83(3):405–410, 1991.
- [5] H. Chebil, J. . Girardot, and C. Courtin. An ontology-based approach for sharing and analyzing learning trace corpora. In *Proceedings - IEEE 6th International Conference on Semantic Computing, ICSC 2012*, pages 101–108, 2012.
- [6] O. Díaz, G. Puente, J. L. Cánovas Izquierdo, and J. García Molina. Harvesting models from web 2.0 databases. *Software and Systems Modeling*, 12(1):15–34, 2013.
- [7] B. Florian, C. Glahn, H. Drachsler, M. Specht, and R. Fabregat Gesa. *Activity-based learner-models for learner monitoring and recommendations in moodle*, volume 6964 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2011.
- [8] Rogers G. Do grades make the grade for program. pages 8–9, 2012.
- [9] J. Kay, N. Maisonneuve, K. Yacef, and P. Reimann. *The big five and visualisations of team work activity*, volume 4053 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2006.
- [10] R. E. Mohedano, A. G. Segador, and J. A. Varo. Moodle module for the management and assessment of teamwork: First results. In *Proceedings of the IADIS International Conference e-Learning 2011, Part of the IADIS Multi Conference on Computer Science and Information Systems 2011, MCCSIS 2011*, volume 2, pages 408–410, 2011.
- [11] S. Özgen, O. Sánchez-Galofré, J. R. Alabart, M. Medir, and F. Giralt. Assessment of engineering students' leadership competencies. *Leadership and Management in Engineering*, 13(2):65–75, 2013.
- [12] D. C. Schmidt. Model-driven engineering. *Computer*, 39(2):25–31, 2006.
- [13] Arie van Deursen, Paul Klint, and Joost Visser. Domain-specific languages: an annotated bibliography. *SIGPLAN Not.*, 35(6):26–36, June 2000.