



**UNIVERSIDAD
REY JUAN CARLOS**

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INFORMÁTICA**

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

Curso Académico 2009/2010

Proyecto de Fin de Carrera

Mejora de la herramienta de autor de CoMoLE

Autor: Víctor Martín Arrabal

Tutor: Estefanía Martín Barroso

RESUMEN

El uso de Internet para el acceso a información es a día de hoy una finalidad extendida. La gran cantidad de información existente en la red de redes hace que el conocimiento sea compartido entre todos los usuarios. Esto provoca que no toda la información pueda ser útil y resulte complicado el acceso a un tipo concreto de información. Por ello han surgido áreas especializadas de información por individuos y empresas que permiten el acceso a un tipo determinado de información.

El incesante auge tecnológico proporciona nuevas formas de aprendizaje y enseñanza, si a esto le sumamos el uso de tecnologías móviles hace que la información requiera de una adecuación de contenidos. Por tanto surge la necesidad del uso de aplicaciones en el ámbito educativo que faciliten esta tarea. CoMoLE (“Context-based adaptative Mobile Learning Environments”) es un sistema de adaptación que permite realizar distintos tipos de actividades a los usuarios recomendándoles las actividades más adecuadas dependiendo de sus características personales, acciones realizadas mientras interactúan con el sistema y contexto en un momento determinado.

Este proyecto consiste en mejorar la herramienta de autor del sistema CoMoLE existente, la cual permite a los profesores definir las características de los entornos adaptativos móviles, mejorando las funcionalidades existentes de la anterior herramienta e incorporando nuevas funciones. El objetivo principal de esta herramienta es facilitar las labores de diseño a los profesores de estos entornos, realizando una interfaz mucho más usable e intuitiva.

ÍNDICE

1. Introducción	1
1.1. Descripción de CoMoLE	1
1.2. Presentación del problema.....	2
1.3. Motivaciones y objetivos.....	3
1.4. Estructura del documento	3
2. Objetivos	5
2.1. Descripción del problema.....	5
2.2. Características generales de los entornos.....	5
2.3. Estudio de alternativas	19
2.4. CGI vs Servlets	19
2.5. Metodología empleada	21
2.6. Herramientas empleadas.....	22
3. Descripción informática	27
3.1. Requisitos	27
3.2. Análisis.....	31
3.3. Diseño	38
3.4. Implementación	39
3.5. Pruebas	52
4. Conclusiones y trabajo futuro	53
4.1. Conclusiones	53
4.2. Trabajos futuros	54
5. Bibliografía	55
6. Anexos	57
A. Manual de instalación.....	57

ÍNDICE DE FIGURAS

FIGURA 1: CICLO DE VIDA DEL PROCESO UNIFICADO.....	22
FIGURA 2: DIAGRAMA DE CASOS DE USO.....	29
FIGURA 3: DIAGRAMA DE ESTADOS DEL CASO DE USO IDENTIFICAR.....	30
FIGURA 4: ESTEREOTIPOS DE CLASES.....	31
FIGURA 5: DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO CREAR ENTORNO.....	32
FIGURA 6: DIAGRAMA DE COLABORACIÓN ASOCIADO AL CASO DE USO CREAR ENTORNO, REPRESENTACIÓN DEL CAMINO BÁSICO.....	32
FIGURA 7: DIAGRAMA DE COLABORACIÓN ASOCIADO AL CASO DE USO CREAR ENTORNO, REPRESENTACIÓN DEL CAMINO ALTERNATIVO "CANCELAR CREAR ENTORNO".....	33
FIGURA 8: DIAGRAMA DE COLABORACIÓN ASOCIADO AL CASO DE USO CREAR ENTORNO, REPRESENTACIÓN DEL CAMINO ALTERNATIVO "ERROR EN LOS DATOS".....	33
FIGURA 9: DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO MODIFICAR ENTORNO.....	34
FIGURA 10: DIAGRAMA DE COLABORACIÓN ASOCIADO AL CASO DE USO MODIFICAR ENTORNO, REPRESENTACIÓN DEL CAMINO BÁSICO.....	34
FIGURA 11: DIAGRAMA DE COLABORACIÓN ASOCIADO AL CASO DE USO MODIFICAR ENTORNO, REPRESENTACIÓN DEL CAMINO ALTERNATIVO "ERROR EN LOS DATOS".....	35
FIGURA 12: DIAGRAMA DE COLABORACIÓN ASOCIADO AL CASO DE USO MODIFICAR ENTORNO, REPRESENTACIÓN DEL CAMINO ALTERNATIVO "CANCELAR MODIFICACIONES".....	35
FIGURA 13: DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO VER ENTORNO.....	36
FIGURA 14: DIAGRAMA DE COLABORACIÓN ASOCIADO AL CASO DE USO VER ENTORNO, REPRESENTACIÓN DEL CAMINO BÁSICO.....	36
FIGURA 15: DIAGRAMA DE COLABORACIÓN ASOCIADO AL CASO DE USO VER ENTORNO, REPRESENTACIÓN DEL CAMINO ALTERNATIVO "ERROR EN LA LECTURA".....	36
FIGURA 16: DIAGRAMA DE CLASES DE ANÁLISIS DEL CASO DE USO ELIMINAR ENTORNO.....	37
FIGURA 17: DIAGRAMA DE COLABORACIÓN ASOCIADO AL CASO DE USO ELIMINAR ENTORNO - CAMINO BÁSICO.....	37
FIGURA 18: DIAGRAMA DE COLABORACIÓN ASOCIADO AL CASO DE USO ELIMINAR ENTORNO - CAMINO ALTERNATIVO "ERROR EN LA LECTURA".....	37
FIGURA 19: DIAGRAMA DE COLABORACIÓN ASOCIADO AL CASO DE USO ELIMINAR ENTORNO - CAMINO ALTERNATIVO "CANCELAR ELIMINAR".....	38
FIGURA 20: DIAGRAMA DE SECUENCIA DEL CASO DE USO ELIMINAR ENTORNO, SECUENCIA CORRECTA.....	39
FIGURA 21: EJEMPLO DE DATOS DE PROFESORES.....	39
FIGURA 22: PANTALLA DE IDENTIFICACIÓN.....	40
FIGURA 23: EJEMPLO DE PÁGINA DE SELECCIÓN DE OPERACIÓN.....	40
FIGURA 24: EJEMPLO DE INFORMACIÓN DISEÑADORES AUTORIZADOS DE LOS ENTORNOS.....	41
FIGURA 25: EJEMPLO DE PANTALLA DE DEFINICIÓN DE CARACTERÍSTICAS GENERALES DEL ENTORNO.....	42
FIGURA 26: EJEMPLO DEFINICIÓN DE RASGOS DE ADAPTACIÓN EN LA PÁGINA DE DEFINICIÓN DE REGLAS GENERALES.....	43
FIGURA 27: EJEMPLO DE PÁGINA DE DEFINICIÓN DE FILTROS GENERALES DE CONTEXTO.....	45
FIGURA 28: EJEMPLO PÁGINA DE ACTIVIDADES.....	48
FIGURA 29: EJEMPLO DE PÁGINA DE CONTENIDOS.....	49
FIGURA 30: EJEMPLO DE PÁGINA PARA AÑADIR ACTIVIDADES A UNA DETERMINADA ACTIVIDAD O REGLA ...	50

ÍNDICE DE TABLAS

TABLA 1: EJEMPLO DE CARACTERÍSTICAS GENERALES DEL ENTORNO ADAPTATIVO MÓVIL ASIGNATURA1	7
TABLA 2: EJEMPLO DE REGLAS GENERALES DE CONTEXTO DEL CURSO (ASIGNATURA1)	8
TABLA 3: EJEMPLOS DE ACTIVIDADES.....	12
TABLA 4: EJEMPLOS REGLAS INDIVIDUALES	13
TABLA 5: EJEMPLO DE REGLAS ESTRUCTURALES.....	15
TABLA 6: EJEMPLO RE REGLAS DE ESPACIOS DE TRABAJO COLABORATIVOS.....	17
TABLA 7: EJEMPLOS DE REGLAS DE HERRAMIENTAS COLABORATIVAS.....	19
TABLA 8: RASGOS DE ADAPTACIÓN PREDEFINIDOS.....	42
TABLA 9: FILTROS GENERALES POR DEFECTO.....	44
TABLA 10: REGLAS APLICABLES A CADA TIPO DE ACTIVIDAD	46

1. Introducción

En este capítulo se da una visión global de los aspectos más relevantes necesarios para la resolución del proyecto. Se hará una presentación del problema, una descripción de CoMoLE y se definirán los objetivos y motivaciones del proyecto.

1.1. Descripción de CoMoLE

CoMoLE (“*Context-based adaptative Mobile Learning Environment*”) es un sistema que permite la recomendación y realización de actividades individuales y colaborativas en diferentes entornos adaptativos. Está basado en el área de enseñanza y se engloba en el conjunto de los sistemas hipermedia adaptativos, los cuales tienen en cuenta las diferentes características de los usuarios para determinar sus necesidades educativas en cada momento.

Para la realización de cada una de las actividades, CoMoLE genera espacios de trabajo de forma dinámica. Permitiendo la inclusión o exclusión de actividades a los distintos usuarios según sus características o requisitos individuales.

Esta recomendación la realiza basándose en los rasgos de adaptación que se tienen en cuenta dentro de los criterios definidos por el profesor a través de la herramienta de autor. Estos rasgos de adaptación están relacionados con las características personales del alumno, las acciones que realiza mientras se encuentra interactuando con el entorno, el contexto en el que se encuentra (dispositivo utilizado, tiempo disponible y localización física). Además de estos criterios definidos explícitamente por el profesor, CoMoLE puede efectuar recomendaciones basándose en las acciones previas de usuarios similares en contextos análogos.

Durante la realización de actividades CoMoLE además de guiar al alumno en las actividades a realizar mientras interactúa con el entorno, almacena sus acciones para poder realizar recomendaciones a otros usuarios en situaciones parecidas.

El acceso a las actividades se realiza a través de un navegador Web, y se puede hacer mediante los diferentes dispositivos que soporta el sistema CoMoLE (PDA, PC, portátil, o teléfono móvil).

Resumiendo, el mecanismo de adaptación de CoMoLE permite:

- La inclusión o exclusión de actividades, según diversos requisitos o rasgos de adaptación.



- Agrupación y organización de actividades relacionadas.
- Aplicar recomendaciones generales sobre los tipos de actividades más adecuadas dependiendo del contexto de los usuarios y/o de características personales.
- Establecer dependencias entre actividades.
- Gestionar grupos de usuarios y actividades colaborativas.
- Generar espacios de trabajo adecuados.

1.2. Presentación del problema

La creación de entornos de trabajo adaptativos requiere de una herramienta que nos permita ayudarnos en la definición de los mismos, permitiéndonos (o impidiéndonos) definir los criterios y características en los que se va a basar el entorno. La herramienta de autor, como ya se ha dicho, permite la creación de entornos de trabajo adaptativos. Esta utilidad está destinada al uso exclusivo de profesores y personal docente, pues estos son los encargados de definir las reglas y criterios que regirán tanto el uso de cada uno de los contenidos de cada materia impartida como la guía de navegación ofrecida a los alumnos.

El problema principal a resolver es por tanto la necesidad de dar soporte a la creación y configuración de entornos adaptativos con el objetivo de recomendar actividades según las distintas situaciones del futuro usuario.

De este problema principal surgen los siguientes subproblemas que la aplicación ha de dar soporte:

- Gestionar la creación, modificación o eliminación de entornos adaptativos móviles (en adelante también llamados cursos).
- Establecer un acceso por un medio común a la herramienta.
- Acceso único y distintivo para cada uno de los profesores.
- Almacenamiento de los cursos y los datos o contenidos asociados de los distintos profesores de forma persistente.
- Permitir la visualización de entornos adaptativos móviles previamente creados de forma organizada e intuitiva.
- Posibilidad de acceso a cursos por uno o varios profesores (por pertenecer a un mismo departamento o ser varios profesores de una misma asignatura).



- Dar soporte a la creación de actividades (derivadas de los entornos) colectivas o en grupo.

Todos estos objetivos se han de conseguir de forma que el profesor pueda utilizar la herramienta de forma simple, y poder realizar cualquiera de las anteriores tareas mencionadas. Cabe anotar que la herramienta define una serie de características de cada entorno (explicadas en el apartado 2.2) que requiere de un conocimiento previo, aunque no muy avanzado, del tipo de reglas, puesto que desconocer su funcionalidad puede originar problemas de entendimiento del uso y finalidad de cualquiera de las utilidades.

En todo momento se trata de facilitar el uso de la herramienta de autor por todo tipo de usuarios (tanto expertos como inexpertos). Dando ayudas para definir y combinar las características del entorno adaptativo a definir. El diseñador se debe sentir a gusto con la herramienta ofreciendo un alto grado de confiabilidad, evitando miedos a realizar posibles cambios en los cursos, y avisando siempre de las opciones disponibles.

1.3. Motivaciones y objetivos

La herramienta de autor, como ya se ha dicho, es la encargada de generar los ficheros necesarios con la información sobre las reglas y características de las que finalmente el sistema CoMoLE recogerá para poder generar los espacios de trabajo. La creación de entornos de aprendizaje adaptativos móviles no es una tarea sencilla y para los diseñadores de estos entornos, el uso de una herramienta que les asesore durante la definición de estos entornos de aprendizaje se hace totalmente necesario. La herramienta de autor facilitara al diseñador la creación de entornos adaptativos móviles. Los profesores podrán gestionar los distintos criterios de adaptación y actualizar las características que el sistema tiene en cuenta a la hora de realizar la recomendación de las actividades más adecuadas. Además, podrá gestionar los contenidos asociados a cada una de las actividades de forma fácil en cualquier momento.

1.4. Estructura del documento

Este documento se compone de los siguientes capítulos:

Capítulo 1: En el capítulo actual se ha descrito el propósito general del proyecto, para ello se ha realizado una descripción de la herramienta CoMoLE, pasando a continuación a describir el problema junto con sus motivaciones y objetivos.



Capítulo 2: En este capítulo se ha resumido los objetivos del proyecto, realizando un análisis de las posibles alternativas para abordar estos objetivos, y describiendo brevemente la metodología y herramientas empleadas.

Capítulo 3: Este capítulo contiene toda la descripción informática que resume el desarrollo del proyecto, por ello se ha estructurado en las partes de especificación de requisitos, análisis, diseño, implementación y pruebas correspondientes a las fases de diseño software.

Capítulo 4: En este capítulo se presenta de forma desglosada las conclusiones y posibles trabajos futuros.

Capítulo 5 y 6: En estos capítulos se recogen las fuentes para realizar este trabajo y se incluye un manual de usuario de la aplicación y un apartado explicativo de los contenidos del CD adjunto.



2. Objetivos

En este capítulo se definirán los objetivos marcados para la realización del proyecto. En primer lugar se realizará una descripción del problema definiendo los objetivos propuestos, a continuación se presentarán las posibles alternativas para abordar el problema y finalmente se describe la metodología empleada, dando una breve descripción de las herramientas empleadas para el desarrollo del proyecto.

2.1. Descripción del problema

Para lograr crear los entornos adaptativos se diseña la herramienta de autor, la cual permite la creación de los cursos. Los diseñadores del entorno son los responsables de definir de forma apropiada los elementos del entorno, por tanto deberán conocer el significado y utilidad de cada una de las opciones. Los cursos tienen una serie de características de configuración las cuales se pasan a detallar a continuación.

2.2. Características generales de los entornos

Todos los entornos que se creen deberán tener una serie de características básicas, las cuales serán aplicadas a todas las actividades que conformen el curso. Todas las características generales son de obligatoria definición. Las características generales de los cursos conforman la base del mecanismo de recomendación de las actividades que toman parte del curso.

Nombre o identificador del entorno: Es el nombre asociado al entorno, es lo que le hace distintivo del resto de cursos creados. Al ser la característica que lo hace diferente al resto será único entre todos ellos. Se trata de una sola palabra identificativa.

Idiomas: Los entornos tienen definidos los idiomas en los que la aplicación da soporte (contenidos e interfaz). Todos los cursos tendrán al menos un idioma en el que se desarrollarán las actividades.

Descripción en los idiomas: Los entornos deberán tener descripciones en los diversos idiomas en los que se imparta la asignatura, cada descripción está formada por una o varias frases que resuma el entorno. El número de descripciones de los idiomas deberá corresponderse con la cantidad de idiomas.



Opciones generales de actividades colaborativas: Las actividades colaborativas son aquellas que se pueden realizar entre varios usuarios y grupos. Estas opciones son solo aplicadas a las actividades que sean de tipo colaborativo.

Las opciones generales de actividades colaborativas lo conforman el tamaño de grupos, el papel o rol que tendrán asignados los miembros de grupos, la generación automática o no automática de grupos y la definición de herramientas colaborativas. La selección del tamaño de grupos la define su tamaño mínimo y máximo. El papel o rol de grupos define la asignación de un líder o no. La definición de herramientas colaborativas permite nombrar las herramientas que se pueden emplear en las actividades colaborativas que conforman el entorno adaptativo móvil.

Rasgos de adaptación: Sirven para definir la información sobre los usuarios que van a realizar las actividades. Los rasgos de adaptación se pueden englobar en 4 tipos: Información personal (información del alumno relevante a lo largo del tiempo), preferencias (la elección de alternativas es un rasgo cambiante), acciones (comportamiento en la realización de actividades) y rasgos de contexto (localización y dispositivo al efectuar actividades). Representan los rasgos del usuario que se pueden resumir en diversos valores, a través de un conjunto de datos numérico o estereotipado. Los rasgos numéricos definirán el intervalo de valores numéricos que puede tomar el rasgo. Los rasgos estereotipados definirán un conjunto definido de valores aislados que puede tomar el rasgo. Estos rasgos son los utilizados para definir todas las reglas del entorno (explicados más adelante) y actuarán como una de las bases del sistema de recomendación de CoMoLE de actividades. Existen rasgos de adaptación predefinidos por su uso común, estos son el emplazamiento o lugar, el tiempo disponible, el dispositivo empleado, así como otros para el tipo de aprendizaje del alumno (visual/verbal, secuencial/global,...).

Un ejemplo de las características generales previamente explicadas se puede ver en la definición del entorno adaptativo Asignatura1 (ver tabla 1). En ella se puede observar que los contenidos de las actividades estarán disponibles en inglés y español. En cuanto a las opciones colaborativas el tamaño de grupos de trabajo será de al menos 5 alumnos y como máximo 9, ningún alumno tendrá que asumir el papel de líder, los grupos no se generarán de forma dinámica y el conjunto de herramientas disponibles para la realización de actividades serán Hc1, Hc2, Hc3. Los rasgos de adaptación definidos son dispositivo, lugar, tiempo disponible e información sobre si es repetidor o no. Estos rasgos de adaptación son los que se usarán para definir las características de las actividades y rasgos del curso.



CARACTERÍSTICA	VALOR/VALORES
Nombre	Asignatura1
Idiomas	Español, Ingles
Descripciones	Asignatura de ejemplo Example subject
Opciones colaborativas	
– Tamaño de grupos	Entre 5 y 9
– Rol de los miembros	Ninguno
– Generación automática de grupos	No
– Herramientas colaborativas	Hc1, Hc2, Hc3
Rasgos de adaptación	
1. Dispositivo	PC, PDA, portátil, teléfono.
2. Lugar	Casa, clase, laboratorio, desconocido
3. Tiempo disponible	Entre 1 y 240 minutos
4. Repetidor	Si, No

Tabla 1: Ejemplo de características generales del entorno adaptativo móvil Asignatural

2.2.1. Reglas generales de contexto del entorno.

Las reglas o filtros generales de contexto sirven para delimitar las actividades que pueden ser accesibles o inaccesibles. Las actividades conforman cada una de las partes, operaciones o tareas que se pueden llevar a cabo en el entorno. Cada curso puede tener uno, ninguno o varios filtros generales de contexto. Cada filtro general de contexto está formado por:

Tipos de actividades: Tipos a los que se desea aplicar la regla de contexto, los tipos sirven para aplicar las reglas a actividades con características comunes. Los tipos de actividades que se pueden dar son: actividades compuestas, teoría, ejemplo, simulación, test, texto corto, atenea, tutoría, revisión, mensaje, archivo, colaborativa, otra. La regla general de contexto se podrá aplicar a uno o varios tipos de actividades de entre el conjunto anterior.

Condición de activación: Es la condición que se debe cumplir o incumplir para acceder a el conjunto de actividades definido. Las condiciones pueden ser simples o compuestas. Las condiciones simples son aquellas formadas por un rasgo de adaptación (uno de los previamente definidos), un comparador y un valor. Para los rasgos estereotipados el comparador será un “=” (igual) y el valor uno de los definidos. Para los rasgos numéricos se usan los comparadores “<” (menor), “>” (mayor), “<=” (menor o igual), “>=” (mayor o igual) e “=” (igual) y el valor un número acotado en el intervalo



definido por el rasgo de adaptación. Las condiciones compuestas están formadas por varias condiciones simples unidas entre ellas por un operador lógico, estos operadores son “y” (conjunción lógica) y “o” (disyunción lógica). Una condición formada por una conjunción lógica resulta verdadero (se cumple) si se cumplen ambas condiciones simples, mientras que una condición con una disyunción lógica se cumple si una condición simple o ambas son verdaderas o se cumplen. Al cumplirse o incumplirse la condición las actividades definidas estarán disponibles para el usuario.

Recomendación: La recomendación indica si las actividades de los tipos escogidos en la regla son recomendados o no al cumplirse la condición de activación.

La tabla2 muestra un ejemplo de filtros generales de contexto del entorno del ejemplo anterior (Asignatura1). Las reglas de contexto definidas como se puede ver hacen uso de rasgos de adaptación definidos en las reglas generales del curso.

La primera regla está formada por una condición simple, indica que no es recomendable realizar actividades de tipo colaborativa o de archivo si el dispositivo empleado es una PDA.

La segunda regla es una condición compuesta unidas por una conjunción lógica, establece que son recomendables las actividades de tipo tutoría, test y ejemplo si se cumple que la actividad se realiza en clase y además el tiempo disponible es menor de 30 minutos (el tiempo por defecto es en minutos).

La tercera regla es una condición compuesta formada por una disyunción lógica y establece que las actividades que sean de teoría son recomendables si se cumple que, o bien, el alumno es repetidor, o el usuario no es repetidor pero dispone de más de 50 minutos.

CONDICIÓN	RECOMENDACIÓN	TIPOS DE ACTIVIDADES
(dispositivo = PDA)	No	Colaborativa, archivo.
(lugar = clase) Y (tiempo < 30)	Si	Tutoría, test, ejemplo.
(repetidor = si)	Si	Teoría.
O ((repetidor = no) Y (tiempo > 50))		

Tabla 2: Ejemplo de reglas generales de contexto del curso (Asignatura1)



2.2.2. Definición de actividades y reglas.

Los cursos están formadas por un conjunto de partes que lo conforman llamados actividades, las actividades conformarán el conjunto de información disponible del curso. El alumno accederá desde su dispositivo a cada una de las actividades que forman parte del entorno para poder acceder a los contenidos e información. Todas las actividades de los entornos tienen una serie de características comunes que son de obligatoria definición:

Identificador: Cada actividad tiene un identificador numérico que sirve para distinguir las actividades entre sí. Este identificador deberá ser único entre las actividades que conformen el entorno a desarrollar.

Nombre: Las actividades tienen que tener un nombre que será el mostrado al usuario cuando quiera realizar la actividad. El nombre de la actividad es único entre todas las actividades del curso creado.

Descripción en los idiomas: Las actividades tienen una descripción que resume el objetivo de la actividad. Estas descripciones deberán ser únicamente las definidas en las reglas generales del entorno, cada actividad tendrá una descripción por cada idioma que el curso al que pertenece da soporte.

Lugar de realización de la actividad: Cada actividad tiene un lugar predefinido donde realizarse. Un alumno al tratar de realizar una actividad si no se encuentra en el lugar adecuado para su realización no se recomendará. Los lugares de realización de las actividades pueden ser: clase, laboratorio, casa, otro o cualquiera (esta última recomendada para cualquier emplazamiento).

Fecha límite: La fecha límite determina hasta que momento la actividad puede estar disponible para realizarse, es útil para actividades con fecha de entrega o actividades con una determinada vida útil. En cualquier caso las actividades tienen que tener determinada una fecha límite de realización.

Tiempo máximo de realización: El tiempo máximo representa lo que puede tardar un alumno en realizar una actividad desde su inicio. Es útil para posibles test o actividades que puedan evaluar los conceptos aprendidos del usuario, estableciendo un tope de tiempo para su ejecución.



Tipo de actividad: El tipo de actividad determinará a su vez el tipo de reglas que se podrán aplicar a cada actividad. En general las actividades que se podrán realizar se pueden clasificar en:

- Actividades individuales: Agrupan los ejercicios que no requerirán de otros alumnos para realizarse.
- Actividades colaborativas o colectivas: Son aquellas que requiere de generar grupos de trabajo para realizar las actividades, estas actividades la forman un tamaño de grupos definido en las reglas principales del entorno.
- Actividades relacionadas con otros usuarios: Son aquellas que no requieren de grupos de trabajo para realizarse pero se generan dinámicamente entre todos los usuarios. Los usuarios pueden interactuar entre sí, accediendo cada uno en diversos momentos a las actividades.
- Actividades externas: Aquellas que el alumno realizará en otros sistemas.

Atendiendo a este tipo de clasificación el mecanismo de recomendación establece los siguientes tipos de actividades. Los tipos de reglas que pueden tener cada tipo se explicarán después.

- i. Actividades compuestas: Las actividades compuestas engloban varias actividades de otros tipos. Las actividades pertenecientes a una actividad de este tipo pueden ser de cualquier tipo (también de conjunto). Las actividades compuestas vienen definidas por reglas estructurales y reglas individuales. Las reglas estructurales especificarán las actividades pertenecientes a la actividad compuesta. Las actividades compuestas deberán contener al menos una “subactividad” de cualquier tipo.
- ii. Actividades colaborativas: Como se ha dicho anteriormente las actividades colaborativas son para grupos de usuarios. Por ser para un grupo de usuarios y tener que disponer de herramientas para realizar actividades, hay que definir las reglas que rijan el acceso a las distintas herramientas que compongan la actividad. A estas reglas se les llaman reglas de espacios colaborativos.
- iii. Resto de actividades: El resto de actividades están compuestas por reglas individuales, las cuales serán el único mecanismo de recomendación de estas actividades. Los tipos con solamente reglas individuales son: actividades de teoría, ejemplo, simulación, test, texto corto, atenea, tutoría, revisión, mensaje, archivo y otra.



Contenidos: Las actividades pueden tener asociados contenidos multimedia. Estos contenidos serán diferentes para cada tipo de actividad, dispositivo para los que están diseñados, y características personales del alumno. Con independencia del tipo de actividad cada contenido puede tener distintos tipos de materiales, tal como texto, imágenes, programas... Cada contenido puede compartirse por varias actividades y tener diferentes versiones según el dispositivo. Ahora bien, no todas las actividades pueden tener contenidos, debido a la orientación y objetivo de la actividad. Las actividades de tipo compuesto, como ya se ha dicho, son las que están formadas por varias “subactividades”, por tanto estas subactividades (que no sean compuestas) serán las que podrán tener contenidos asociados. Las actividades colaborativas son un caso especial en que deberán compartirse contenidos para el grupo de trabajo formado en una actividad de este tipo. Por tanto, los contenidos de las actividades colaborativas vendrán definidos según las reglas de espacios colaborativos oportunas, las cuales se explican al final del apartado actual. Esto nos permitirá adaptar los contenidos a las características conjuntas del grupo y de los usuarios.

Los contenidos por tanto estarán asociados a un dispositivo en el que se podrán emplear, estos posibles dispositivos son: PC, PDA, laptop (ordenador de sobremesa), portátil y cualquiera de los anteriores. De esta manera los contenidos serán mostrados para cada tipo de dispositivo empleado en cada momento. Los contenidos tendrán las características definidas por la actividad, esto quiere decir que si por ejemplo se ha excedido el tiempo máximo de realización, o se ha sobrepasado la fecha límite de la actividad los contenidos dejarán de estar disponibles.

La tabla 3 muestra ejemplos de actividades de distintos tipos creadas y de sus características. Estas actividades formarán parte de un entorno. Las descripciones que poseen son las mismas que posee el curso, por tanto el entorno al que pertenezcan deberá tener descripciones en inglés y español. Por ejemplo pueden pertenecer al entorno del ejemplo anterior de nombre Asignatura1.

La actividad de identificador 1 tiene el nombre Actv1, y es de tipo Teoría, por tanto podrá disponer de contenidos, que en este caso será Tema1.pdf que se podrá visualizar en cualquier dispositivo y repasoClase.ppt que requerirá de un ordenador de sobremesa para acceder al contenido.



La actividad de identificador 2 tiene el nombre Actv2 y se trata de unos ejercicios propuestos para realizar en clase. Al ser de tipo ejercicios puede tener contenidos, que en este caso son 3 archivos con extensión pdf.

La actividad de identificador 3 es una actividad compuesta pensada para realizar en cualquier lugar, sirve para englobar toda la parte práctica de la asignatura. Al ser una actividad compuesta no puede disponer de contenidos (podrán disponer de contenidos sus subactividades) y dispone de 3 subactividades: 2 prácticas y un examen. Estas subactividades en este ejemplo no están definidas pero es obligatorio que a su vez estén definidas sus características.

La actividad de identificador 4 y nombre Actv4 es una actividad colaborativa. Este tipo de actividades, tal como se ha dicho, tienen reglas de espacios de trabajo que determinarán sus contenidos y no poseen subactividades. Hacen uso de las herramientas colaborativas definidas en las características generales del entorno al que pertenezcan.









Id	Nombr e	Descripción en idiomas	Lugar	Fecha límite	Tiempo Max. (min.)	Tipo	Contenidos y (dispositivo) O subactividades
1	Actv1	 Tema 1  Topic 1	casa	11/06/ 2010	120	Teoría	CONTENIDOS 1. Tema1.pdf (cualquiera) 2. RepasoClase.ppt (PC)
2	Actv2	 Ejercicios en clase  Class exercises	clase	30/06/ 2010	55	Ejercic ios	CONTENIDOS 1. Hoja1.pdf (cualquiera) 2. Hoja2.pdf (cualquiera) 3. Hoja3.pdf (cualquiera)
3	Actv3	 Parte practica  Practice part	cualqui era	23/04/ 2010	180	Comp uesta	SUBACTIVIDADES 1. Práctica1 2. Práctica2 3. Examen práctico
4	Actv4	 Repaso en grupos  Group review	laborat orio	25/05/ 2010	110	Colabo rativa	Ninguno

Tabla 3: Ejemplos de actividades



2.2.2.1. Reglas individuales.

Las reglas individuales sirven para recomendar o restringir el acceso a una determinada actividad. Son las reglas más simples y constan únicamente de las siguientes partes:

Identificador: Es un valor numérico que identifica a cada una de las reglas que se pueden dar en la actividad. El identificador numérico tiene que ser único y distintivo entre el resto de reglas de la actividad.

Nombre de actividad: Corresponde a la actividad que tiene la regla individual. El nombre de la actividad es único entre todas las actividades del entorno creado.

Condición de activación: La condición de activación tiene el mismo formato que la condición de las reglas generales de contexto (Véase apartado 2.1.2). La condición de activación no puede ser una condición vacía, ya que estas reglas solo tienen condición de activación. No tienen campo de recomendación, por tanto al activarse pasa a recomendarse la actividad.

Si una actividad no tiene ninguna regla individual es recomendado a todos los alumnos en cualquier circunstancia.

En la tabla 4 se han creado ejemplos de reglas individuales de las actividades del ejemplo de la tabla3.

La regla individual de la actividad de nombre Actv1, establece que para poder realizar la actividad se tiene que estar conectado desde un ordenador de sobremesa.

Las reglas individuales de la Actv2, tienen identificador 1 y 2, aunque una regla tiene el mismo identificador que la regla de la Actv1 es correcto al ser distintas actividades. Las reglas individuales recomiendan la actividad cuando se cumpla que:

- I. Se accede desde cualquier dispositivo y se dispone de más de 30 minutos
- II. El alumno es repetidor, o, no es repetidor y dispone de más de 20 minutos.

Id	Nombre de actividad	Condición de activación
1	Actv1	(dispositivo = PC)
1	Actv2	(dispositivo = cualquiera) Y (tiempo > 30)
2	Actv2	(repetidor = si) O ((repetidor = no) Y (tiempo>20))

Tabla 4: Ejemplos reglas individuales



2.2.2.2. *Reglas estructurales.*

Las reglas estructurales indican el modo en que se organizan las actividades compuestas y las condiciones que estas deben cumplir para recomendar una determinada subactividad que compone la actividad compuesta. Cada actividad compuesta se puede descomponer de diversas formas, permitiendo una gran flexibilidad en la descomposición y el orden de acceso a las subactividades que la formen. Cada regla estructural está formada por una serie de características comunes que han de ser definidas por parte del diseñador:

Identificador de regla: Se trata de un valor numérico que identifica a cada una de las reglas que se pueden dar en cada actividad compuesta. El identificador numérico tiene que ser único y distintivo entre el resto de reglas de una misma actividad de tipo compuesto.

Condición de activación: La condición de activación tiene el mismo formato que la condición de las reglas generales de contexto (Véase apartado 2.1.2). Si la regla no tiene ninguna condición de activación se recomienda a todos los alumnos que acceden a las subactividades que compongan la regla. Estas condiciones no tienen campo de recomendación, por tanto al activarse se recomendarán las subactividades.

Actividades que componen la actividad compuesta: Cada regla estructural puede tener una o varias subactividades. Las actividades que conforman cada regla estructural tendrán sus propias características (las mismas que una actividad normal) y ser de cualquier tipo, si una subactividad es de tipo compuesta esta a su vez podrá tener sus propias reglas estructurales.

Secuencia de navegación: La secuencia de navegación indica el orden en que se deberán desarrollar las distintas subactividades. Las posibles secuencias permitidas son: guía directa y guía flexible. Si la guía de una regla estructural es directa los alumnos deberán realizar las subactividades en el orden propuesto. Si la guía, en cambio, es flexible los alumnos serán libres de realizar las subactividades en el orden que más les convenga.

La tabla 5 recoge tres ejemplos de reglas estructurales de actividades compuestas. Estas reglas pertenecen a dos actividades de tipo compuesto.

La regla de identificador 1 es una regla estructural perteneciente a la actividad de nombre ActvComp1, esta regla estructural indica que si el dispositivo para realizar la



actividad es un ordenador de sobremesa la guía es directa, teniéndose que realizarse las subactividades Práctica1, Practica2 y Examen práctica en ese estricto orden

La regla de identificador 2 es una regla estructural que pertenece a la misma actividad que la regla anterior (ActvComp1) y su condición de activación es una condición vacía, lo cual implica que serán recomendadas las subactividades que la componen en cualquier caso. Está formado por dos subactividades: Teoría1 y ActividadCol1. Su guía es flexible lo que implica que los alumnos podrán realizar las subactividades que lo componen en el orden que les interese.

La regla de identificador 3 es una regla de la actividad ActvComp2 con una condición compuesta que recomienda las subactividades que la componen si ocurre una de las dos condiciones: el lugar de realización de la actividad es en casa y el dispositivo empleado es un ordenador de sobremesa, o, el dispositivo empleado es una PDA y el tiempo disponible es mayor de 15 minutos. Si una de estas condiciones se da lugar son recomendadas las subactividades Teoría2, Ejercicios1, Contenidos1, Teoría3 en orden explícito, teniendo que realizarse las actividades en el orden descrito, para poder acceder a las actividades que precedan a una se deberán completar las actividades anteriores.

Id.	Nombre	Condición	Guía	Subactividades
1	ActvComp1	(dispositivo = PC)	Directa	Práctica1, Práctica2, Examen práctico
2	ActvComp1	-	Flexible	Teoría1, ActividadCol1.
3	ActvComp2	(lugar=casa) Y (dispositivo=PC) O (dispositivo=PDA) Y (tiempo>15)	Directa	Teoría2, Ejercicios1, Contenidos1, Teoria3.

Tabla 5: Ejemplo de reglas estructurales

2.2.2.3. Reglas de adaptación para espacios de trabajo colaborativos.

Las reglas de espacios colaborativos son un caso especial de reglas en los que hay que definir las reglas que van a regir la recomendación de los grupos de trabajo. Este tipo de reglas se da únicamente en las actividades de tipo colaborativo. Los grupos de trabajo son generados según el tamaño mínimo y máximo definido en las reglas generales del entorno y son creados de forma automática o libre.

Cada espacio de trabajo estará formado por el enunciado de la actividad, las herramientas principales de las que hace uso (definidas en las reglas generales del



entorno) y herramientas adicionales correspondientes a las herramientas principales. Las herramientas principales constituirán la interfaz principal del entorno de trabajo, mientras que las herramientas adicionales estará situadas en una interfaz secundaria que será ofrecidas a estudiantes que necesiten actividades extra de apoyo. Para poder definir los criterios de adaptación y generación de los espacios colaborativos, se permiten especificar reglas que especifiquen los criterios de uso y acceso a las distintas herramientas y entornos colaborativos.

Para la definición de todas estas características y criterios se dan dos tipos de reglas de adaptación para entornos colaborativos: reglas de espacios de trabajo y reglas de herramientas colaborativas. Si al crear las reglas generales del entorno se decidió no usar conjunto de herramientas, no se podrán agregar reglas de espacios de trabajo, ni reglas de herramientas colaborativas, ya que ambas hacen uso de los conjuntos de herramientas.

Las reglas de espacios de trabajo colaborativos detallan el enunciado (contenido accesible) y conjunto de herramientas que se podrán usar para realizar una actividad colaborativa. Cada regla de espacio de trabajo está definida por unas características comunes:

Identificador: Al igual que en el resto de reglas el identificador es numérico y permite distinguir cada regla entre las demás reglas de espacios de trabajo.

Nombre actividad: Determina la actividad a la que pertenece cada regla de espacio de trabajo. La actividad a la que pertenezca la regla tiene que ser de tipo colaborativo para que se pueda dar este tipo de regla.

Conjunto de herramientas: El conjunto de herramientas a utilizar cuando se cumpla la condición de activación. Tiene que ser un conjunto de herramientas definido en las reglas generales del entorno. Este conjunto de herramientas corresponderá a la interfaz principal del usuario.

Enunciado de la actividad: Presentan el contenido que tendrán accesibles los estudiantes que cumplan la condición. El enunciado corresponde al ejercicio o tarea que realizarán en el entorno.

Condición de activación: La condición de activación tiene el mismo formato que todas las condiciones anteriores. Si la regla no tiene ninguna condición de activación se permite a todos los alumnos el acceso al enunciado de la regla del espacio de trabajo



colaborativo que quieran realizar. Estas condiciones no poseen campo de recomendación, por tanto al activarse se recomendará el uso de la herramienta.

En la tabla 6 se puede ver un ejemplo de reglas de espacios de trabajo colaborativo. En ella se han creado cuatro reglas para distintas actividades que a continuación pasan a explicarse.

La regla de identificador 1 de la actividad Actv1Col recomienda el uso de la herramienta Ch1 con los contenidos Ejercicios 1 si se cumple que el usuario se encuentra en casa y ha accedido a la actividad mediante un ordenador de sobremesa.

La regla de identificador 2 de la actividad Actv1Col recomienda el uso del conjunto de herramientas Ch2 con los contenidos Teoría2 a todos los alumnos.

La actividad de nombre Actv2Col tiene también dos reglas de espacios de trabajo para un mismo conjunto de herramientas, por lo que todos los estudiantes utilizarán a priori el mismo conjunto de herramientas. La regla de identificador 1 recomienda el acceso a EjerciciosAvanz si el usuario es repetidor y dispone de más de 30 minutos, en cambio recomienda realizar el enunciado EjerciciosBasic si no es repetidor o si es repetidor y tiene menos de 30 minutos. Estas condiciones son entre sí excluyentes y no se puede dar el hecho de acceder a los 2 enunciados de forma simultánea.

Id	Nombre actividad	Conjunto de herramientas	Enunciado	Condición de activación.
1	Actv1Col	Ch1	Ejercicios1	(dispositivo=PC) Y (lugar=casa)
2	Actv1Col	Ch2	Teoria2	
1	Actv2Col	Ch2	EjerciciosAvanz	(repetidor=si) Y (tiempo>30)
2	Actv2Col	Ch2	EjerciciosBasic	(repetidor=no) O ((repetidor=si) Y (tiempo<=30))

Tabla 6: Ejemplo re reglas de espacios de trabajo colaborativos

Las reglas de herramientas colaborativas definen, partiendo de un conjunto de herramientas, las herramientas más adecuadas para cada grupo de estudiantes. Estableciendo herramientas secundarias que podrán ser utilizadas para la realización de una actividad colaborativa. Estas herramientas servirán de ayuda para las herramientas de las que derivan. A su vez estas herramientas secundarias definidas pueden ser simples o compuestas (conjunto de herramientas), por lo que si son un conjunto se podrán especificar reglas de herramientas colaborativas para estas herramientas secundarias. Este tipo de reglas constan de:



Identificador: El identificador es un valor numérico único entre las reglas de herramientas colaborativas. Permite que el sistema distinga el orden en que se han definido las reglas.

Nombre actividad: Representa la actividad (previamente definida por el diseñador) a la que corresponde la regla. Esta actividad tiene que ser colaborativa para que se puedan aplicar este tipo de reglas.

Conjunto de herramientas: El conjunto de herramientas con las que se podrá acceder a las herramientas secundarias cuando se cumpla la condición de activación. Puede ser un conjunto de herramientas definido en las reglas generales del entorno o una herramienta definida en otra regla.

Herramientas: Las herramientas o herramienta que se recomendará su uso cuando se cumpla la condición de activación. Se tiene que especificar al menos una herramienta para cada regla de herramienta colaborativa. Estas herramientas serán mostradas en una interfaz secundaria al alumno.

Condición de activación: La condición de activación tiene el mismo formato que todas las anteriores. Si la regla está vacía se recomienda a todos los usuarios el acceso a las herramientas secundarias definidas para la regla. No tienen campo de recomendación, por tanto al activarse se recomendarán el uso de todas las herramientas secundarias.

Algunos ejemplos de reglas de herramientas colaborativas se pueden ver en la tabla 7.

En el ejemplo de identificador 1 de la Actv1Col se utiliza el conjunto de herramientas Ch1 y este conjunto de herramientas utiliza las herramientas Chat y corrección, las cuales serán recomendadas a todos los alumnos. El ejemplo de identificador 2 de la Actv1Col define la herramienta de la anterior regla de identificador 1. La herramienta Corrección es un conjunto de herramientas que a su vez dispone de una Interfaz para la corrección y un foro para debatir la corrección, para poder utilizar esta herramienta se debe tener un PC o una PDA y disponer más de 35 minutos para la corrección.

El ejemplo de la actividad Actv2Col hace uso del conjunto de herramientas Ch2, la cual se compone de Foro y Email, se recomendará el uso de estas herramientas cuando el usuario acceda a ellas desde un ordenador. Al no definirse nuevas reglas de herramientas colaborativas de estas nuevas herramientas se entiende que son herramientas simples que no hacen uso de herramientas adicionales.



Id	Nombre Actividad	Conjunto de herramientas	Herramientas	Condición de activación
1	Actv1Col	Ch1	Chat, Corrección	
2	Actv1Col	Corrección	InterfazCorrección, ForoDiscusión.	(dispositivo=PC) O ((dispositivo=PDA) Y (tiempo>35))
1	Actv2Col	Ch2	Foro, Email.	(dispositivo=PC)

Tabla 7: Ejemplos de reglas de herramientas colaborativas

2.3. Estudio de alternativas

Después de explicar las características que tienen los entornos adaptativos móviles, se pasa a analizar las posibles opciones de diseño. Al analizar la funcionalidad de la anterior herramienta y sus posibles utilidades ventajosas se decidió, debido a los problemas que ocasionaba, rediseñar la herramienta partiendo de cero sin reutilizar nada de la anterior. La anterior herramienta estaba programada en lenguaje Java mediante el uso de tecnología *servlets* ya que su librería de funcionalidad básica a su vez estaba escrita en lenguaje Java. Como alternativa al uso de *servlets* se puede utilizar los programas o tecnologías CGI, que soportan el uso de Java, las cuales tras analizarlo (véase siguiente apartado) se desechó el uso de esta tecnología.

2.4. CGI vs Servlets

Los programas CGI se encargan de recibir del servidor los datos de un formulario. Cada formulario lleva incluido un campo de nombre “*action*” con el que se asocia el nombre de programa en el servidor. El servidor arranca dicho programa y le pasa los datos que han llegado con el formulario. Los datos son pasados al programa CGI de dos maneras distintas:

1. Usando una variable de entorno del sistema operativo del servidor (método GET).
2. Por medio de un flujo de caracteres que llega por la entrada estándar (método POST).

En ambos casos, la información introducida en el formulario llega en la forma de una única cadena de caracteres en la que el nombre de cada campo del formulario se asocia con el valor asignado por el usuario, y en la que los blancos y ciertos caracteres especiales se han sustituido por secuencias de caracteres de acuerdo con una determinada codificación, el programa CGI decodifica esta información y separa los valores de los distintos campos. Finalmente almacena los valores o realiza la operación



solicitada el programa CGI. De ordinario, el programa CGI termina enviando al cliente una página HTML a través de la salida estándar (asociada a la pantalla) en la que le informa de las tareas realizadas, y posibles fallos.

Es importante resaltar que estos procesos tienen lugar en el servidor. Esto a su vez puede resultar un problema, ya que al tener múltiples clientes conectados al servidor, el programa CGI puede estar siendo llamado simultáneamente por varios clientes, con el riesgo de que el servidor se llegue a saturar. Téngase en cuenta que cada vez que se recibe un requerimiento se arranca una nueva copia del programa CGI.

Los *servlets* son programas escritos en Java que se ejecuta en el marco de un servicio de red, y que recibe y responde a las peticiones de uno o más clientes. El funcionamiento del tratado de información es bastante similar a los programas CGI pero con muchas mejoras:

1. Los *servlets* son independientes del servidor y sistema operativo, y al estar escritos en java el servidor puede estar escrito en cualquier lenguaje con el mismo resultado, los programas CGI hay que compilarlos en el lenguaje del servidor.
2. Los programas CGI, tal como se ha dicho, tienen que cargarse tantas veces como peticiones realicen los clientes, mientras que los *servlets* quedan activos en memoria una vez que se ejecutan por primera vez, optimizando el uso de memoria del sistema.
3. Los *servlets* al utilizar programas en Java, absorben las ventajas que proporciona este lenguaje (conectar con bases de datos, seguridad en la información, ejecución de métodos en ordenadores remotos...).

Resumiendo los *servlets* son más prácticos y de mayor utilidad que los programas CGI, esto sumado a las facilidades que nos proporciona el entorno de desarrollo NetBeans para el uso de *servlets* ha provocado la decisión de usar *servlets*. NetBeans da soporte completo a la programación de *servlets* y uso de librerías y proyectos en java, así como la creación de páginas mediante archivos JSP, lo cual facilita sustancialmente el desarrollo del proyecto.

Los *servlets* permiten la creación de páginas Web dinámicas en diversos lenguajes: HTML, PHP, ASP, PYTHON, RUBY,... Este código dinámico genera páginas Web en lenguaje HTML.



La interfaz con el usuario se decidió implementarla completamente en inglés, ya que no suponía ningún problema desarrollarla la aplicación en este idioma, y la mayoría de aplicaciones de *software* están en inglés. Permitiendo un posible uso más entendido de la aplicación.

En cuanto al almacenamiento persistente de los datos se realizará en ficheros XML, ya que el sistema CoMoLE procesa este tipo de ficheros y el proyecto base del que se parte tiene métodos implementados para almacenar los entornos creados en ficheros XML.

2.5. *Metodología empleada*

Entre todas las metodologías existentes, la empleada para el desarrollo del proyecto ha sido la del Proceso Unificado o Proceso Unificado de Desarrollo Software (abreviado como UP). Este marco de desarrollo está centrado en la arquitectura, es un desarrollo iterativo e incremental, está dirigida por casos de uso y enfocada en los riesgos.

El proceso unificado está centrado en la arquitectura, por lo que existen varios modelos y vistas que cubren las funcionalidades del sistema. El modo en que se organiza cada una de las fases depende de los casos de uso.

El UP está dirigido por casos de uso, mediante los cuales se capturan los requisitos funcionales y se definen los contenidos de cada iteración. Se pretende que de este modo se realicen casos de uso en cada uno de los pasos que van a darse en el ciclo de vida del proyecto software. De este modo se obliga a pensar en los requisitos importantes para el usuario además de los que se podrían considerar oportunos.

El proceso unificado está enfocado a los riesgos, y exige que se definan los posibles riesgos en fases tempranas de cada fase. De esta manera se puede establecer un orden de cada una de las fases según el control de los riesgos principales identificados.

El UP es un marco iterativo e incremental formado por las fases de Inicio, elaboración, construcción y transición. Cada fase se divide en iteraciones, cada vez que se termina una fase se produce un incremento que ofrece mejoras y funcionalidades en el sistema de desarrollo. Las iteraciones corresponden a una secuencia ordenada de requisitos, análisis, diseño, implementación y pruebas cada iteración implica un mayor o menor esfuerzo según la fase, véase figura 1.



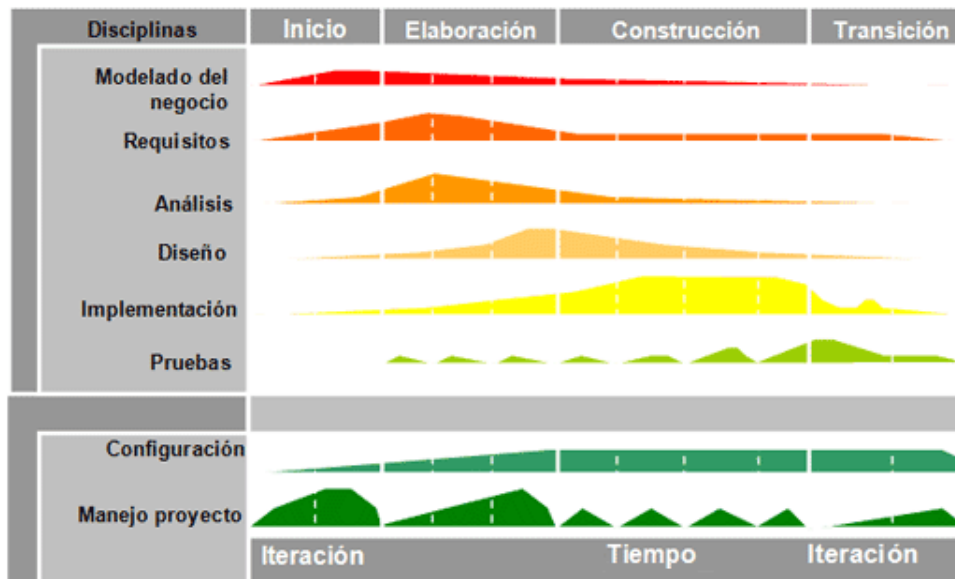


Figura 1: Ciclo de vida del proceso unificado

2.6. Herramientas empleadas

Para la realización del proyecto se han empleado diversas tecnologías, entornos, lenguajes y demás que requieren de una breve explicación de su utilidad y aportes por los que se ha considerado oportuno emplear:

2.6.1. Java

Java es un lenguaje de programación orientado a objetos creado por Sun Microsystems. Java toma mucha de la sintaxis y utilidad de C y C++. Pero tiene sus propias características las cuales a continuación se resumen:

Simple: Java elimina muchos de los problemas y características más confusas. Añade características útiles como el reciclador de memoria (*garbage collector*).

Orientado a objetos: Java trabaja con sus datos como objetos y con interfaces a estos. Soporta las tres características de su paradigma: encapsulación, herencia y polimorfismo.

Distribuido: Se ha construido con extensa capacidad de interconexión TCP/IP. Existen librerías y rutinas para acceder e interactuar con protocolos como http y ftp.

Robusto: Java realiza verificaciones buscando problemas en tiempo de compilación y ejecución. La comprobación de tipos ayuda a detectar errores durante el desarrollo.



Seguro: El código Java pasa por muchas verificaciones antes de ejecutarse. Se pasa a través de un verificador de *bytecode* que comprueba errores.

Interpretado: El intérprete de Java (*run-time*) puede ejecutar directamente el código objeto. Enlazar un programa consume menos recursos que compilarlo.

Multitarea: Java permite realizar múltiples actividades simultáneamente en un programa. El beneficio consiste en un mejor rendimiento interactivo y mejor comportamiento.

Dinámico: Java se beneficia de la tecnología orientada a objetos y no intenta conectar todos los módulos que comprenden la aplicación hasta el tiempo de ejecución.

Arquitectura neutral: El compilador compila el código a un fichero objeto independiente de la arquitectura de la máquina donde se ejecuta. Cualquier máquina que tenga el sistema de ejecución *run-time* podrá ejecutar el código objeto

2.6.2. XML

XML son las siglas en inglés de *Extensible Markup Language* (lenguaje de marcas extensible). Un lenguaje de marcado se puede definir como una forma de codificar un documento, donde a cada texto se le incorporan etiquetas con información adicional relativa a la estructura del texto. La información almacenada por el lenguaje de marcado tiene obligatoriamente que seguir una estructura para luego al leer la información se pueda realizar de forma correcta (a esto se le llama su DTD).

En el desarrollo del proyecto se ha empleado XML para almacenar los datos de los entornos generados de forma persistente. La lectura/escritura de ficheros XML requiere en java de funciones adicional, por ello se ha empleado para este fin la librería JDOM. JDOM es una librería de código abierto de java y diseñada específicamente para Java.

2.6.3. Servlet

Los *servlets* son clases normales Java que se crean cuando se necesitan y se destruyen cuando no se necesitan. Los *servlets* se cargan en un motor especial que se encarga de la creación y destrucción de cada *servlet* de la aplicación Web. Este motor se encarga de mantener el ciclo de vida del *servlet*, el motor oficial de Sun es el contenedor *Jakarta-Tomcat*, el cual se ha empleado para la realización del proyecto. En este caso se ha empleado la clase *HttpServlet* de la que heredarán todos los *servlets* del proyecto. La



clase `HttpServlet` es muy útil para el procesamiento de formularios HTML y utiliza el protocolo HTTP.

HTTP es un protocolo orientado a petición-respuesta. Una petición está formada por unos campos cabecera y un cuerpo, que puede estar vacío. Una respuesta HTTP contiene un código de resultado y también una cabecera y cuerpo.

2.6.4. NetBeans

NetBeans IDE (entorno de desarrollo integrado) es un entorno de desarrollo para programadores para escribir, compilar, probar, depurar, analizar y distribuir programas. Está escrito en java y es gratuito y de código abierto, Sun Microsystems fue el fundador del proyecto. Da soporte a muchos más lenguajes que java y dispone de una gran cantidad de módulos para extender su funcionamiento. NetBeans ha sido utilizado durante el desarrollo de todo el proyecto por su facilidad de uso y la ayuda que ofrece al tratamiento de *servlets*.

2.6.5. HTML

HTML corresponden a las siglas *HyperText Markup Language* (lenguaje de marcado de hipertexto). Como su nombre indica es un lenguaje de marcado, pero está enfocado a la creación de páginas Web en lugar del almacenamiento de datos como el lenguaje XML. Se escribe entre etiquetas para describir la estructura y el contenido. HTML puede describir la apariencia de un documento y puede incluir scripts que pueden afectar al comportamiento de los navegadores al procesar el HTML.

2.6.6. JSP

JSP corresponde a las siglas *Java Server Pages*. Esta tecnología ayuda a los diseñadores y desarrolladores de aplicaciones Web separando la lógica de la presentación de la lógica de la aplicación, para permitir separar el contenido estático del dinámico de una página HTML. Mediante esta tecnología es posible cambiar el diseño de la página sin tener que modificar el código que genera la página.

2.6.7. Servidor

Un servidor es un computador (generalmente más potente que un computador normal) que forma parte de una red y provee servicios a otras computadoras llamados clientes. Sigue el modelo cliente-servidor en el que se atiende una petición realizada por el cliente. Esta petición puede ser tanto de acceso a contenidos como de servicio de



aplicaciones, en general tareas en beneficio de otras aplicaciones. El contenedor escogido para el desarrollo del proyecto ha sido *tomcat*, el cual es un servidor Web utilizado en combinación con el servidor Web apache. Las peticiones realizadas a estos servidores Web se realizan mediante el protocolo HTTP.



3. Descripción informática

Este apartado se describe por fases la especificación de requisitos, análisis, diseño e implementación que se ha realizado para llevar a cabo el proyecto, en cada fase se argumenta las decisiones tomadas y se adjuntan diagramas específicos de cada modelo de fase que clarifican los detalles.

3.1. Requisitos

Este apartado expone los requisitos que se han establecido para la realización del proyecto. El objetivo principal del análisis de requisitos del sistema es la identificación de los requisitos básicos que el sistema ha de dar soporte, separando los detalles procedimentales de cada funcionalidad.

La herramienta de autor está diseñada para la creación y mantenimiento de los entornos de trabajo adaptativos. Cada autor dispondrá de las opciones necesarias para mantener sus propios entornos diseñados. La herramienta de autor sólo se recomienda utilizarse en dispositivos PC u ordenador portátil, debido a la cantidad de información que se tiene que especificar y por tanto se mostrará por pantalla. Durante la especificación de requisitos se decidió darle soporte solo a los dispositivos previamente mencionados.

Las funcionalidades principales que permite la herramienta de autor son:

1. Diseñar nuevos entornos.
2. Modificar entornos previamente creados.
3. Ver detalles de entornos previamente especificados.
4. Eliminar cursos existentes.

Todas estas funcionalidades siempre se podrán realizar bajo una previa identificación. La identificación de cada profesor permite mantener organizados y separados los entornos pertenecientes a cada diseñador. De esta manera cada profesor sólo puede realizar cualquiera de las funcionalidades anteriores en un espacio de trabajo que le concierne. Los datos de cada profesor por tanto deberán estar almacenados en un medio físico para que se pueda disponer de ellos de forma estable y duradera y no perder información al reiniciar el sistema. La herramienta de autor no puede dar de alta nuevos profesores, puesto que ésta sólo le corresponde verificar y controlar el acceso de los diseñadores previamente agregados. Por tanto, al no ser objetivo de la herramienta de



autor el dar de alta usuarios en el sistema no se dispondrá de una funcionalidad con dicho objetivo.

Los requisitos funcionales son aquellas características que la aplicación a desarrollar debe incorporar, como acciones que el sistema debe ser capaz de desempeñar. En estos requisitos se muestra como ha de ser el trabajo final que se entregue al cliente.

Los requisitos funcionales de estas funcionalidades principales son:

R1. El método de acceso a la herramienta de autor será vía Internet. Por lo que para usar la herramienta de autor el profesor deberá estar conectado a Internet.

R2. El profesor deberá introducir sus datos antes en el sistema para poder realizar cualquier operación en la herramienta de autor. La validación consistirá en la correcta identificación (inserción de los datos) de un profesor.

R3. Si el profesor desea modificar, ver o eliminar un entorno adaptativo deberá tener algún curso creado.

R4. Si el profesor desea ver, modificar o eliminar un entorno deberá aparecer como diseñador autorizado a realizar esa operación para el curso seleccionado en los datos internos de la herramienta.

La figura 2 muestra un diagrama de casos de uso que conforman las funcionalidades principales que da soporte la aplicación y los requisitos funcionales que cada caso de uso requiere para poder realizarse. El diagrama de casos de uso representa la interacción que tiene el actor (en este caso el profesor de los cursos adaptativos) con el sistema para realizar una determinada tarea. Este diagrama está formado por 4 casos de uso (modificar, ver, crear y eliminar entorno), que representan a cada unidad coherente de funcionalidad, y un caso de uso de inclusión (relación *include*), lo cual significa que el actor (diseñador o profesor) también hace uso de ese caso de uso al acceder a cualquiera de los que lo incluyen. Este último caso será la identificación del profesor que está interactuando con la herramienta de autor.

Los casos de uso que forman parte del diagrama son:

Caso de uso *Crear entorno*:

- Actor: Diseñador



- Descripción: El diseñador o profesor decide crear un nuevo entorno adaptativo, acción que requiere de la identificación del profesor.

Caso de uso *Ver entorno*:

- Actor: Diseñador
- Descripción: El diseñador o profesor desea ver un entorno previamente creado, y al que debe tener acceso. Esta operación requiere una previa identificación.

Caso de uso *Modificar entorno*:

- Actor: Diseñador
- Descripción: El diseñador o profesor modifica un curso que previamente se ha creado, y a la cual tiene acceso, para ello previamente se ha tenido que identificar para poder realizar la acción.

Caso de uso *Eliminar entorno*:

- Actor: Diseñador
- Descripción: El diseñador o profesor desea eliminar un curso previamente creado. Esto requiere de su identificación en el sistema.

Caso de uso *Identificación*:

- Actor: Ninguno
- Descripción: Para poder entrar en el sistema y realizar cualquier operación de las anteriores se requiere de una correcta identificación y que la herramienta valide que el profesor tiene permisos sobre un determinado curso o entorno.

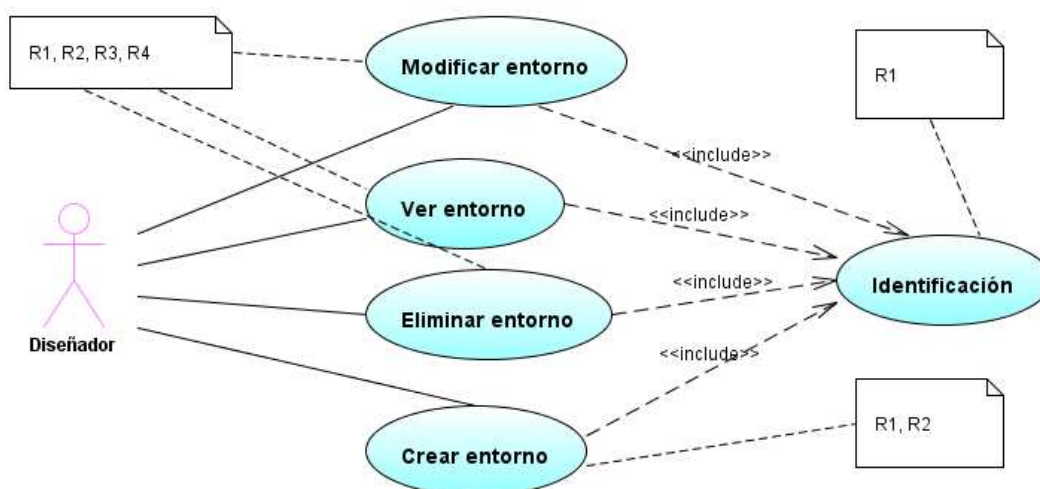


Figura 2: Diagrama de casos de uso

Tal y como se puede ver en la figura anterior para que el diseñador pueda realizar cualquier tarea siempre requiere que se identifique previamente. Por tanto la secuencia de operaciones que realizará el usuario sea cual sea la operación que quiera realizar será siempre la misma.

Para describir esta secuencia de realización de una operación se ha definido un diagrama de estados. El diagrama de estados permite mostrar el comportamiento dinámico del sistema. Cada diagrama de estado está asociado a un caso de uso del sistema. Está formado por un estado inicial, un conjunto de transiciones entre estados intermedios y un estado final. En cada transición se puede mostrar el acontecimiento ocurrido para darse esa transición de un estado a otro. Cada estado, representado con una caja de texto, puede tener diferentes acciones: *Entry* (acción realizada instantáneamente al entrar en el estado), *Do* (acción realizada durante el estado, termina al producirse otra transición), *Exit* (acción realizada a la salida del estado).

En la figura 3 se muestra el diagrama de estados para el caso de uso Identificación. En este caso en primer lugar se solicitaría los datos de acceso a la herramienta, al introducir los datos se confirmarían si son correctos o no. Si estos datos no son correctos habrá una transición de estados en el que se avisará al diseñador la incorrecta identificación, si la identificación en cambio es correcta se produce una transición a un estado en el que se esperará la selección de la opción a realizar con la herramienta de autor. Una vez seleccionada, se realizará la operación y se llegará al estado final.

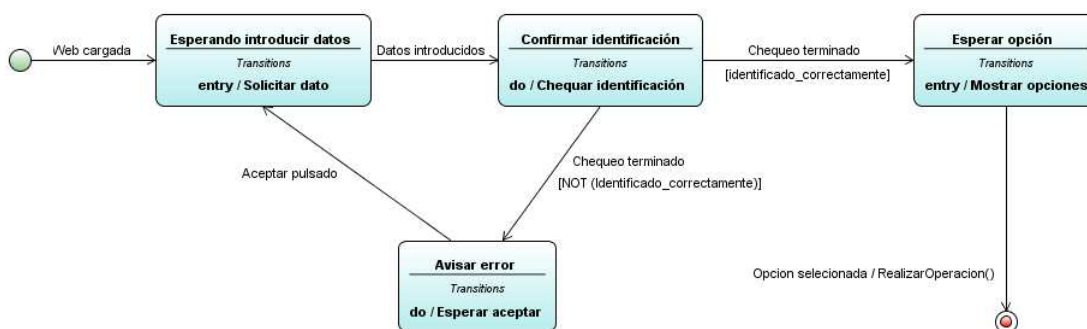


Figura 3: Diagrama de estados del caso de uso Identificar

Los requisitos no funcionales identifican las características no funcionales del sistema (restricciones de la plataforma, seguridad, tiempos de acceso,...). Las características no funcionales esperadas en la herramienta de autor son:



RNF1: Debe ser intuitivo en toda la interfaz.

RNF2: En la interfaz gráfica debe estar presente todas las funcionalidades deseadas y conseguir una buena adaptación y colocación de los contenidos.

RNF3: El tiempo de acceso de una sesión tendrá un tiempo límite.

RNF4: Soportará conexión de múltiples usuarios en el mismo momento.

RNF5: El acceso a los datos de cada usuario se realizará en el servidor para evitar agujeros en la seguridad.

3.2. *Análisis*

La fase de análisis consiste en el estudio de los requisitos descritos en la captura de requisitos, refinándolos y organizando su estructura. El modelo de análisis representa una orientación al entorno de la implementación. Para ello estructura los paquetes de análisis jerárquicamente. Cada paquete de análisis contiene clases del análisis (diferentes subsistemas) y realizaciones de los casos de usos, es el mecanismo para organizar los elementos del análisis. Las realizaciones de los casos de uso corresponden a una colaboración que describe como se realiza el caso de uso, desarrollando su iteración.

El análisis se realiza mediante los diagramas de clases y los diagramas de colaboración. Los diagramas de clases están formados por (ver figura 4):

- Clases entidad: derivan de la descripción del caso de uso, representa la información persistente en el sistema. En nuestro caso la base de datos consistirá en los datos almacenados en los ficheros XML y ficheros asociados a las actividades.
- Clases interfaz: representan la interacción entre el sistema y el actor. En nuestro caso toda la interacción se realiza mediante una interfaz gráfica.
- Clases de control: gobierna el flujo del caso de uso. Representarán la comunicación entre la interfaz y la entidad.



Figura 4: Estereotipos de clases

Los diagramas de colaboración son diagramas de interacción que complementan los diagramas de clases. Representan la secuencia de acciones en un caso de uso. Los diagramas de colaboración contienen los elementos de los diagramas de clases de análisis más flechas que indican la secuencia y orden de iteración. Pueden aparecer distintos diagramas de colaboración para mostrar diferentes opciones o posibles fallos a la hora de realizar el caso de uso.

A continuación se realizarán los diagramas de clases de análisis y colaboración de los casos de uso: crear, modificar, ver y eliminar entorno. Todos los casos de uso mencionado hacen uso del caso de uso de identificación, por lo que se mostrará también en los diagramas de colaboración los pasos previos de identificación en el sistema.

3.2.1. Caso de uso crear nuevo entorno

La figura 5 muestra el diagrama de clases del caso de uso crear entorno, en el cual se puede ver que para crear un nuevo entorno se requiere del gestor y la base de datos para almacenar el curso.



Figura 5: Diagrama de clases de análisis del caso de uso crear entorno

La figura 6 representa el diagrama de colaboración asociada al camino básico del análisis del caso de uso crear entorno. El camino básico determina los pasos seguidos para la correcta creación de un entorno y una identificación correcta del diseñador del entorno.

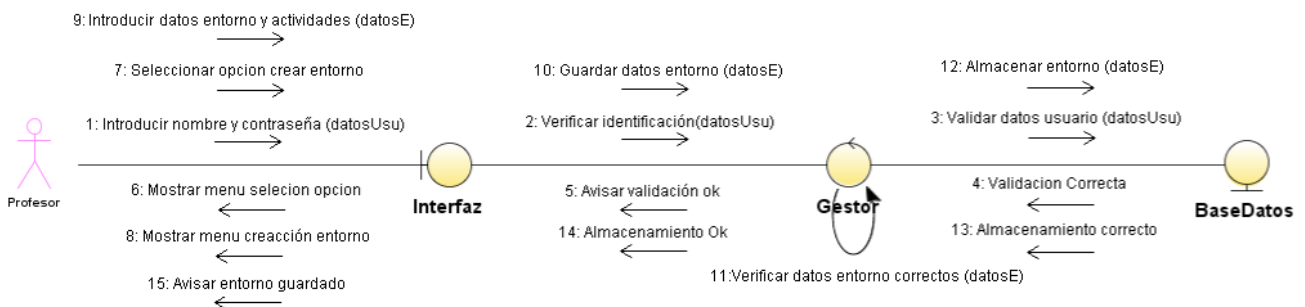


Figura 6: Diagrama de colaboración asociado al caso de uso crear entorno, representación del camino básico

La figura 7 muestra el diagrama de colaboración del camino alternativo “Cancelar creación”. Este camino muestra los pasos que se producen cuando un usuario al seleccionar crear un entorno decide finalmente no almacenarla. Tal y como se puede ver en el diagrama se avisará al profesor de que la operación se cancela y no se almacenará el curso creado.

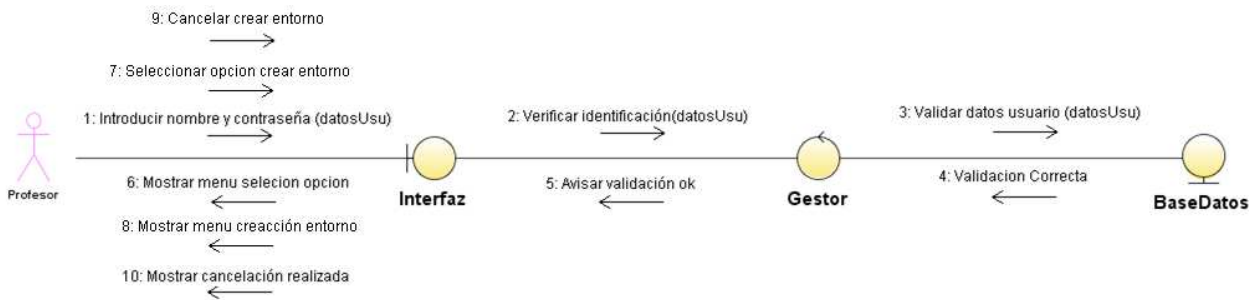


Figura 7: Diagrama de colaboración asociado al caso de uso crear entorno, representación del camino alternativo "Cancelar crear entorno"

En la figura 8 se muestra el camino alternativo del diagrama de colaboración asociado a la creación de un nuevo entorno con errores. Al verificar que existe algún error en el curso se avisa al diseñador de los errores ocurridos. Antes de guardar el entorno se verifica su correcto diseño.

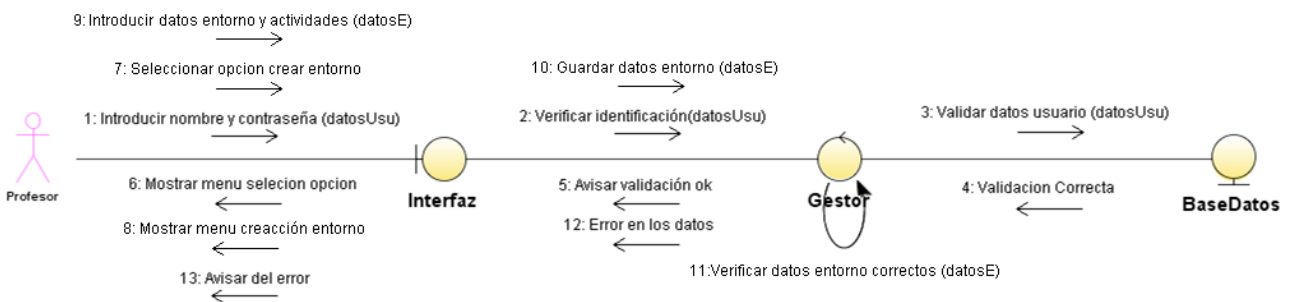


Figura 8: Diagrama de colaboración asociado al caso de uso crear entorno, representación del camino alternativo "Error en los datos"

3.2.2. Caso de uso modificar entorno

Para modificar un entorno hay que cargar un curso previamente almacenado y mostrar los datos para realizar los cambios deseados, una vez aplicados los cambios guardar el entorno. Por tanto el diagrama de clases del caso de uso modificar entorno, hará uso de



la interfaz, gestor y la base de datos. La figura 9 muestra el diagrama de clases de uso de la modificación de un entorno existente.

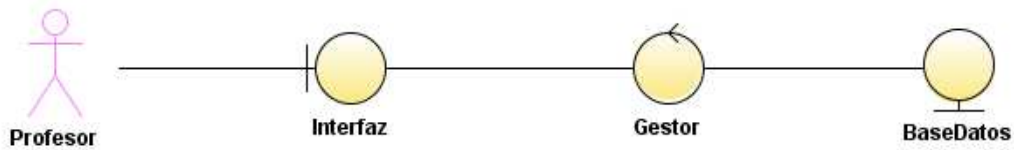


Figura 9: Diagrama de clases de análisis del caso de uso modificar entorno

La figura 10 muestra el camino básico del diagrama de colaboración del caso de uso modificar entorno. Este camino muestra los pasos que se darían si no ocurriera ningún problema para realizar y aplicar los cambios.

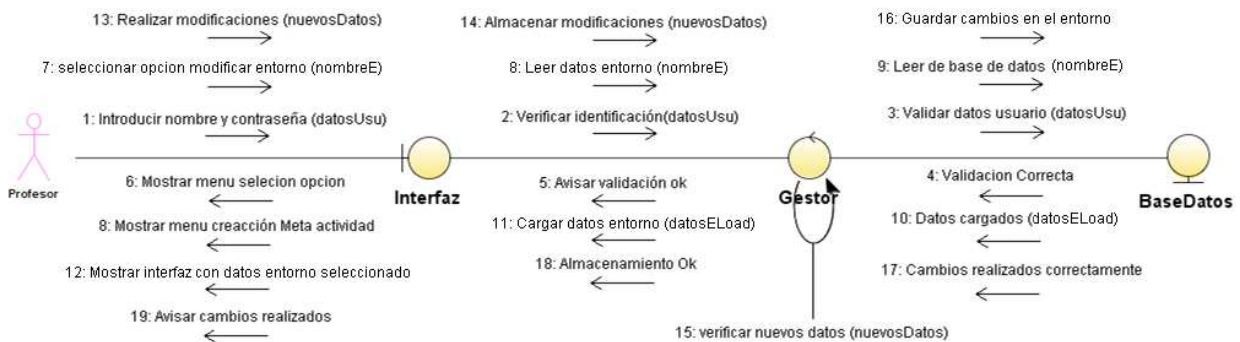


Figura 10: Diagrama de colaboración asociado al caso de uso modificar entorno, representación del camino básico

El diagrama de colaboración del camino alternativo “*error en los datos*” viene representado en la figura 11. Este camino se producirá cuando el diseñador o profesor introduza unos cambios en el entorno que no puedan ser aplicados y por tanto almacenados. Al darse este caso, el curso no se modificará (permaneciendo guardada la información previamente almacenada a la modificación) y se avisará al profesor de la incorrecta definición de los datos del entorno que se han realizado las modificaciones.

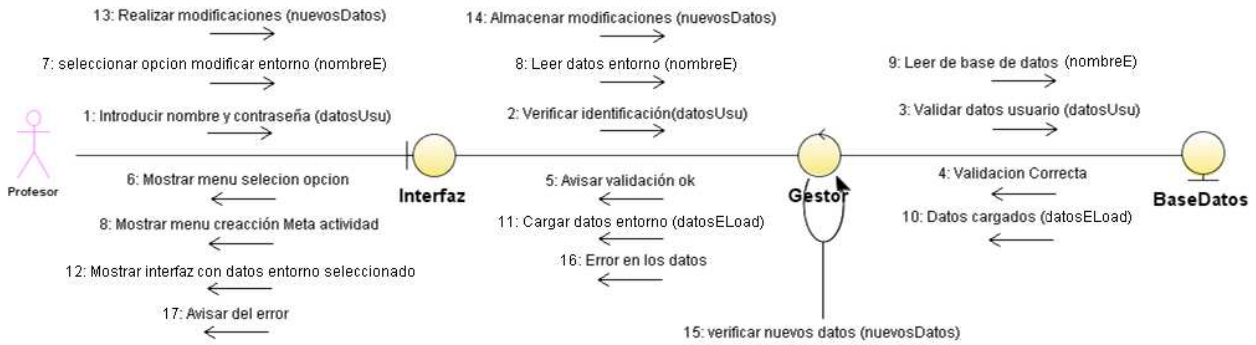


Figura 11: Diagrama de colaboración asociado al caso de uso modificar entorno, representación del camino alternativo "Error en los datos"

La figura 12 muestra el diagrama de colaboración del camino alternativo “Cancelar modificaciones”, se produce cuando el diseñador decide no aplicar los cambios que ha realizado en el entorno deseado, y la información sobre el entorno original permanece almacenada.

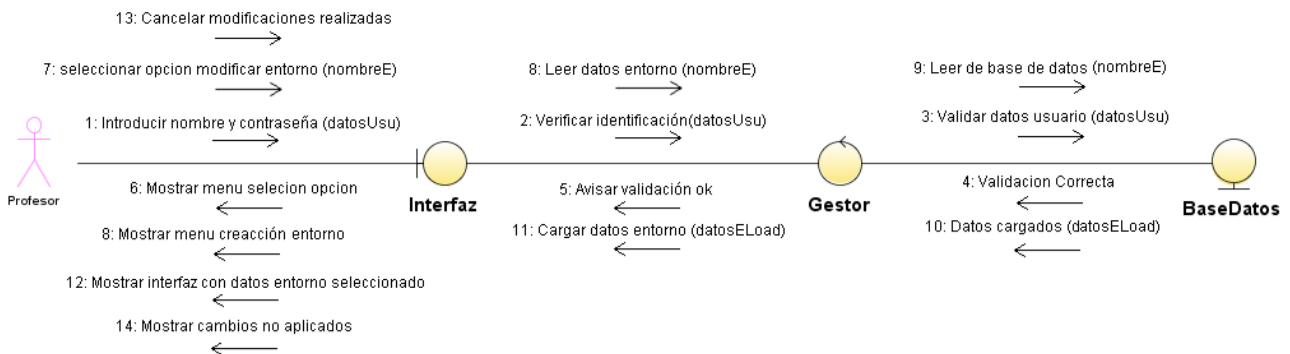


Figura 12: Diagrama de colaboración asociado al caso de uso modificar entorno, representación del camino alternativo "Cancelar modificaciones"

3.2.3. Caso de uso ver entorno

Para ver un entorno hay que cargar un curso previamente almacenado y mostrar los datos. Por tanto el diagrama de clases del caso de uso ver entorno, hará uso de la interfaz, el gestor y la base de datos. La figura 13 muestra el diagrama.



Figura 13: Diagrama de clases de análisis del caso de uso ver entorno

Cuando se desea ver los datos de un entorno previamente creado simplemente se seleccionará el curso del que se quiere ver los detalles. Al no haber detalles que se quieran modificar sólo se accede una vez a los datos del curso para cargarse y visualizarse en la interfaz gráfica. La figura 14 muestra el camino básico que se realiza para realizar esta operación de forma satisfactoria. Tras ver el entorno el usuario podrá salir de la aplicación o volver al menú principal para seleccionar otra opción. En el caso de la figura 14 el usuario decide salir de la aplicación.

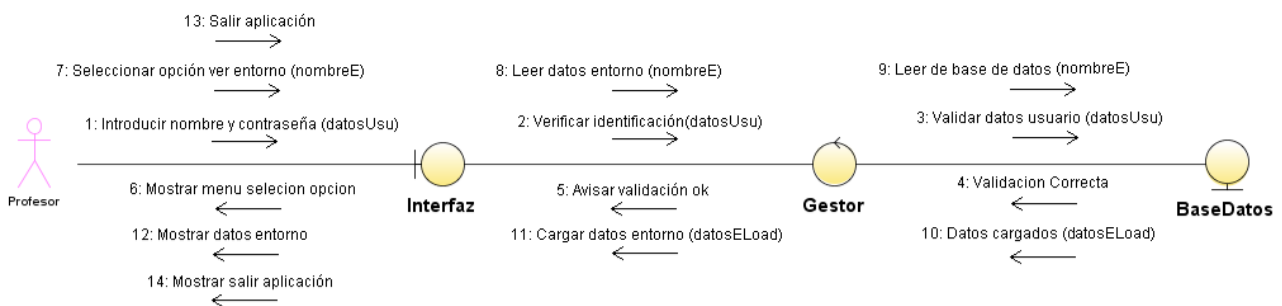


Figura 14: Diagrama de colaboración asociado al caso de uso ver entorno, representación del camino básico

Al querer ver los datos de un entorno se podrían dar fallos en la lectura (falta de autorización para ver el entorno, problemas en la información almacenada en el entorno adaptativo,...). Cuando estos fallos aparezcan, se avisará al profesor del problema surgido explicándose sus causas. La figura 15 recoge este caso en el que se podrían dar fallos en la lectura de la información almacenada sobre el entorno.

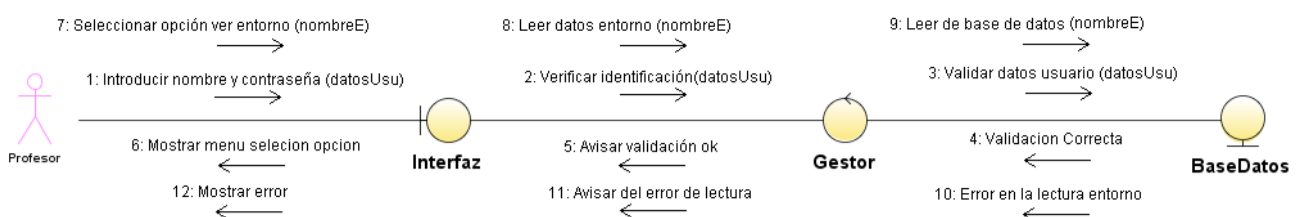


Figura 15: Diagrama de colaboración asociado al caso de uso ver entorno, representación del camino alternativo "Error en la lectura"

3.2.4. Caso de uso eliminar entorno

Para la eliminación de toda la información asociada a un entorno hay que seleccionar un curso previamente creado. El diagrama de clases de este caso de uso, utilizará la interfaz, el gestor y la base de datos. La figura 16 muestra el diagrama de clases de este caso de uso.



Figura 16: Diagrama de clases de análisis del caso de uso eliminar entorno

La eliminación consiste en el borrado de la base de datos de los datos involucrados en el entorno (profesor asociado, datos generales del entorno, contenidos asociados a cada una de las actividades, ...). Al eliminar un entorno de forma satisfactoria se produce el camino básico recogido en la figura 17.

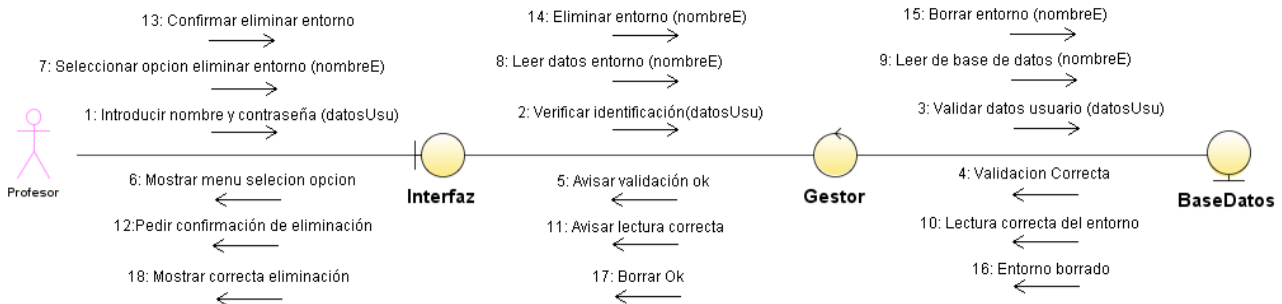


Figura 17: Diagrama de colaboración asociado al caso de uso eliminar entorno - Camino básico

El diagrama de colaboración del camino alternativo “error en la lectura” viene representado en la figura 18. Este camino ocurrirá al darse fallos en la lectura de la base de datos del entorno. Cuando se de este caso no se borrará el entorno.

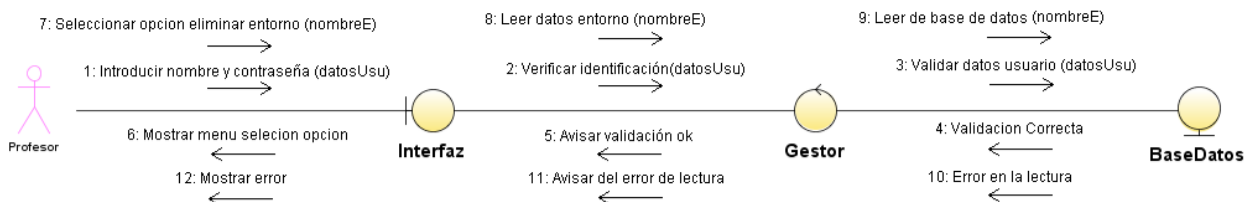


Figura 18: Diagrama de colaboración asociado al caso de uso eliminar entorno - Camino alternativo "Error en la lectura"

Cuando se quiere eliminar un entorno se pide una confirmación de la eliminación, a modo de seguridad, para evitar borrados no deseados. La figura 19 muestra el caso en el que se cancela la eliminación.

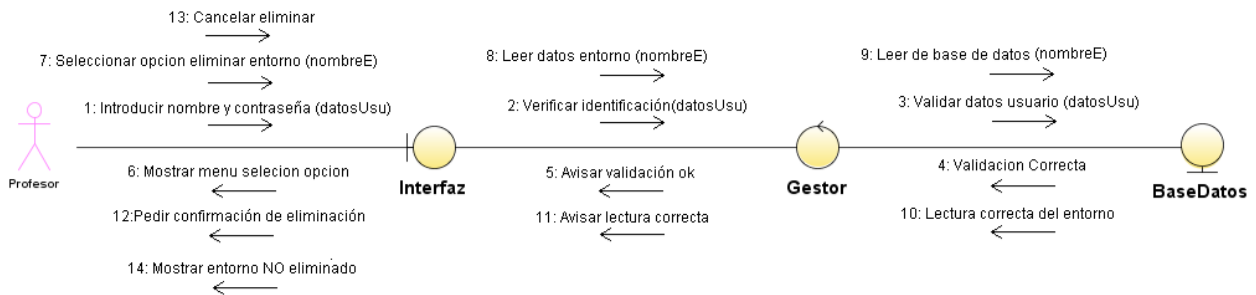


Figura 19: Diagrama de colaboración asociado al caso de uso eliminar entorno - Camino alternativo "Cancelar eliminar"

Todos estos diagramas UML pertenecientes a la fase de análisis representan la orientación a las fases de diseño e implementación. Durante la fase de diseño se muestra cómo finalmente se desarrolló la solución propuesta para realizar todos los casos de uso anteriores, y cómo se introdujeron nuevas funcionalidades no presentes en la fase de análisis y análisis de requisitos que mejoran y facilitan el uso de la herramienta de autor. Finalmente en la fase de implementación se muestra cómo se ha implementado los productos de las fases anteriores.

3.3. Diseño

El diseño de los casos de uso se va a esquematizar mediante diagramas de secuencia, los cuales nos darán una visión detallada de la secuencia de acciones cronológica que se realizará para completar cada operación. Son similares a los diagramas de colaboración, pero representa un acercamiento mucho mayor a la implementación al mostrarnos una interacción entre los objetos y también la interfaz que usará para realizar cada operación. Los objetos o interfaces se representan en cajas con líneas discontinuas verticales y los mensajes pasados entre los objetos con líneas verticales. Para todos los diagramas se mostrará solo su secuencia correcta.

Para la eliminación, modificación o visualización de los entornos se decidió durante la fase de diseño que solo se podrían realizar estas operaciones sobre los cursos de los que el diseñador tuviera privilegios. La figura 20 muestra los pasos dados para la correcta eliminación de un entorno a modo de ejemplo. Se puede ver que no existe



comprobación para saber si el diseñador dispone de privilegios sobre el entorno para eliminarlo, por lo que si decide eliminar un entorno será porque se dispone de acceso al mismo. Al finalizar y confirmar el borrado se avisa al diseñador de su correcta eliminación.

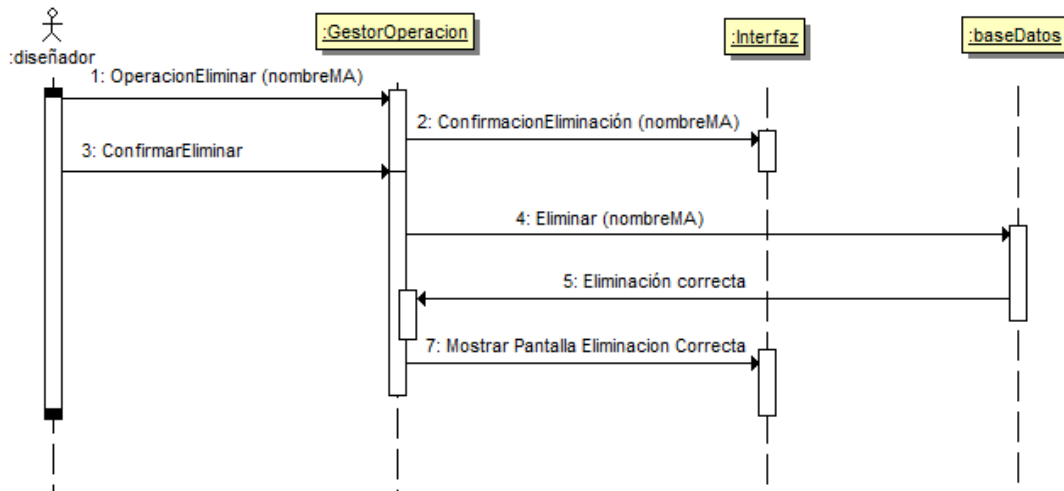


Figura 20: Diagrama de secuencia del caso de uso eliminar entorno, secuencia correcta

3.4. Implementación

En los siguientes apartados se explica la solución propuesta, el funcionamiento de la interfaz y el control realizado por la herramienta de autor para realizar las principales operaciones, se han omitido las páginas de error por motivos de espacio disponible para la realización de esta memoria.

3.4.1. Identificación

La identificación es lo primero que debe realizarse antes de poder realizar cualquier otra operación. Los datos de acceso de cada profesor que puede acceder a la herramienta de autor se encuentran almacenados en un fichero de nombre profesores.xml. La figura 21 muestra un ejemplo de datos almacenados. Al introducir los datos se compararán con los del profesores.xml para saber si el profesor tiene permiso para acceder a un determinado entorno adaptativo.

```

<Profesores>
  <profesor login="estefania" password="estefaniapwd" email="estefania@mailFalso.es" />
  <profesor login="paco" password="pacopwd" email="paco@mailFalso.es" />
  <profesor login="pepe" password="pepepwd" email="pepe@mailFalso.es" />
  <profesor login="victor" password="victorpwd" email="victor@mailFalso.es" />
</Profesores>
  
```

Figura 21: Ejemplo de datos de profesores



Para la identificación la herramienta presenta al diseñador una página como la de la figura 22. En esta pantalla se introducirá el *login* y *password*, y se procederá a comprobar los datos, si algún o ambos campos son incorrectos se avisará al profesor y podrá volver a introducir los campos, si la identificación es correcta se pasa a la siguiente página de selección de la opción que desea realizar.



Figura 22: Pantalla de identificación

3.4.2. Selección de opción

Una vez identificado correctamente el diseñador accede a una página en la que el profesor podrá seleccionar la operación deseada (crear, borrar, modificar o ver entorno), o bien, salir de la aplicación si así lo desea. Para ello la herramienta muestra al profesor una página como la de la figura 23. En la parte superior derecha de la herramienta se situará las opciones de sesión, en este caso finalizar sesión (*logout*). Para realizar las operaciones de borrar, modificar o ver entorno se mostrarán los entornos a los que se tiene acceso. En caso de no tener acceso a ningún entorno se avisará de ello.

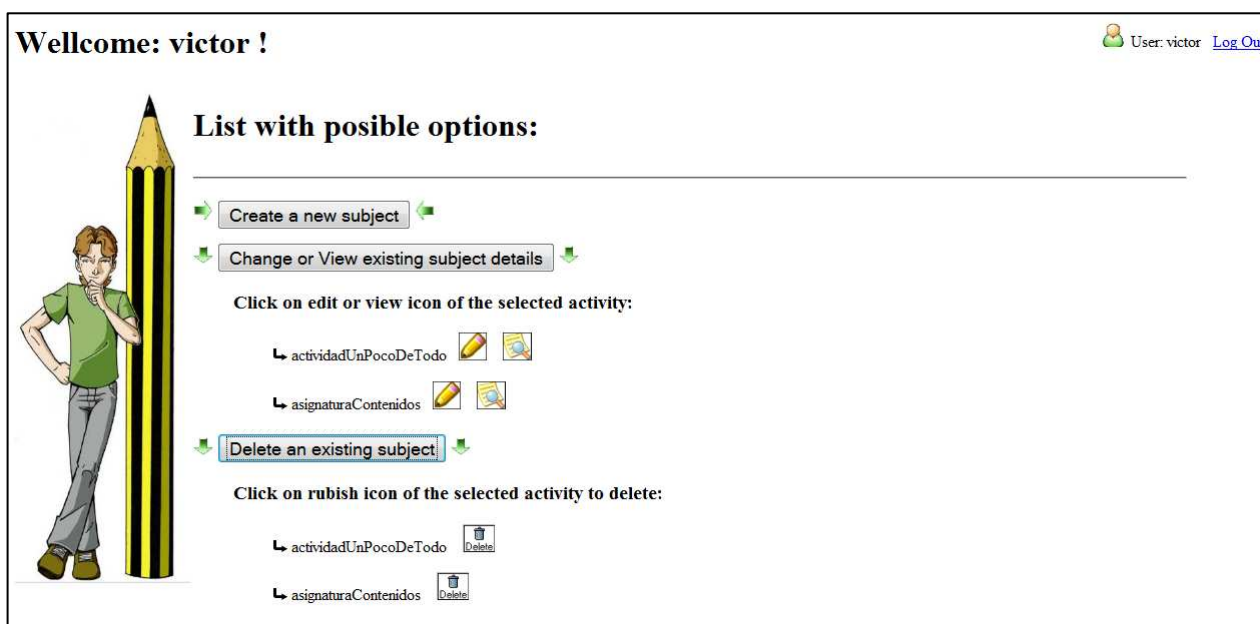


Figura 23: Ejemplo de página de selección de operación



Para controlar los entornos a los que cada profesor puede acceder y además añadir la posibilidad de permitir a varios profesores acceder a un mismo entorno, se empleó un fichero de nombre *listados.xml* como base de datos. Un ejemplo del formato de este fichero y sus etiquetas creadas para diferenciar la información se puede ver en la figura 24. Se observa que por ejemplo el diseñador de nombre victor tiene acceso a “actividadUnPocoDeTodo” y “asignaturaContenidos”. Y asignaturas como “asignatura Contenidos” es compartida por varios profesores.

```

<Lists>
  <usersL>
    <user id="estefania">
      <metaActivitiesL>
        <metaActivity>asignaturaContenidos</metaActivity>
      </metaActivitiesL>
    </user>
    <user id="victor">
      <metaActivitiesL>
        <metaActivity>actividadUnPocoDeTodo</metaActivity>
        <metaActivity>asignaturaContenidos</metaActivity>
      </metaActivitiesL>
    </user>
    <user id="pepe">
      <metaActivitiesL>
        <metaActivity>asignaturaContenidos</metaActivity>
      </metaActivitiesL>
    </user>
    <user id="paco">
      <metaActivitiesL>
        <metaActivity>asignaturaNueva</metaActivity>
      </metaActivitiesL>
    </user>
  </usersL>
</Lists>

```

Figura 24: Ejemplo de información diseñadores autorizados de los entornos



3.4.3. Crear entorno

Al seleccionar la opción crear un nuevo entorno se nos abre una página donde el diseñador introducirá las características comunes de todos los entornos. En esta pantalla se deberá introducir el nombre, descripciones, seleccionar las reglas de adaptación que se aplicarán, configuración de los entornos colaborativos (tamaño de grupo, tipo de generación, roles y herramientas colaborativas), y la agregación de los coautores a los que se permitirá el acceso al entorno. La figura 25 muestra la parte superior de la página de definición de las características comunes donde se pueden introducir estas características.



Step 1 of 4: Complete subject details

User: victor [Log Out](#) [Cancel MA creation](#)

Subject Name: Languages description (select at least one):  Spanish
  English

Adaptation Capabilities: Context
 Structural
 Individual

Collaboration Settings: Size of groups: between and
 Dynamic Worgroup Generation: No Yes
 Roles: None Leader-Regular
 Define collaborative tools: No Yes

↳ Insert name of a collaboraive Tool and click in Add botton to add:

Collaborative Tool	Delete Tool
Ch1	<input checked="" type="checkbox"/> delete
Ch2	<input checked="" type="checkbox"/> delete

Co-Author Settings: I can only modify, delete or show the subject
 Allow other users to modify, deleter or show the subject

↳ Select the author and click in Add botton to add:

Co-Author	Delete author
estefania	<input checked="" type="checkbox"/> delete
paco	<input checked="" type="checkbox"/> delete

Figura 25: Ejemplo de pantalla de definición de características generales del entorno

A continuación se definirán los rasgos de adaptación que serán posteriormente utilizados para realizar las condiciones de todas las reglas. Para facilitar esta tarea se han especificado rasgos de adaptación por defecto, que contienen los valores más utilizados (ver tabla 8).

Nombre	Valores
Dispositivo	PC, PDA, laptop, móvil
Lugar	Casa, clase, laboratorio, desconocido, otros
Visual/Verbal	Visual, Verbal
Sensitivo/Intuitivo	Sensitivo, Intuitivo
Secuencial/Global	Secuencial, Global
Activo/Reflexivo	Activo, Reflexivo
Tiempo	Entre 1 y 720

Tabla 8: Rasgos de adaptación predefinidos



En la parte inferior de la página de definición de características básicas del entorno se añadirán los rasgos de adaptación (ver figura 26), donde se podrán añadir los rasgos predefinidos anteriores o añadir nuevos rasgos de adaptación numéricos o estereotipados. En el ejemplo de la figura se puede ver como se han añadido 2 rasgos predefinidos y un rasgo numérico nuevo, y se estaba procediendo a añadir un nuevo rasgo estereotipado con valores a1 y a2.

Current Features Configuration:

Name	Type	Priority	Possible Values	Delete feature
device	Stereotype	1	Pc Pda Laptop Mobile_phone	
sensingOrIntuitive	Stereotype	1	Sensing Intuitive	
RasgoNuevo1	Numerical	2	between 12 and 45	

Features: click on Show Or Hide icon to show or hide Feature creation menu:

Default feature configuration:

- Device (pc, laptop, pda, mobile phone)
- Place (home, class, lab, unknown, others)
- Visual/verbal dimension of learning styles (visual, verbal)
- Sensing/intuitive dimension of learning styles (sensing, intuitive)
- Sequential/global dimension of learning styles (sequential, global)
- Active/reflective dimension of learning styles (active, reflective)
- Time (between 1 minute and 720 minutes)

Priority:

New feature configuration:

• Name:

• Type:

• Priority:

• Values: Insert a value to add to the new feature and click on Icon to add:

Value Name	Delete value
a1	
a2	

Confirm Feature:

[Go to next Step](#)

Figura 26: Ejemplo definición de rasgos de adaptación en la página de definición de reglas generales

En la parte superior derecha de la herramienta se dispone de opciones para cancelar la creación del entorno y cerrar sesión. Esta barra estará presente en todas las páginas que conforman el proceso de creación de un curso.

Una vez añadidas las características generales del entorno y añadidos los rasgos de adaptación que se desean utilizar en el resto de reglas, se pulsará en “Go to next step” para ir a la siguiente página, no antes de que la herramienta nos haya confirmado la



correcta definición de todos los datos (nombre no repetido, selección de idiomas...). A continuación se pasará a definir las reglas generales de contexto.

La página de definición de filtros generales de contexto se encuentra separada en 4 partes (ver figura 27). La parte superior indica el paso actual y las opciones de sesión. La segunda parte indica la (in)disposición de aplicar los filtros generales de contexto. Si se deciden no aplicar no se mostrará el resto de la página y no se aplicarán los filtros. En el ejemplo estos filtros se encuentran activados. En la siguiente parte se muestra las reglas generales que actualmente han sido especificadas. Finalmente en la parte inferior de la página se crean los filtros actuales, formado por una primera parte en la que se muestran los filtros generales de contexto por defecto. Las condiciones de estos filtros generales por defecto vienen recogidas en la tabla 9. La herramienta solo nos permitirá agregar aquellos filtros por defecto que contengan todos los rasgos incluidos en la condición.

Condición	Actividades	Recomendación
((device=PDA) OR (device=phone))	Test, review	Si
((time<10) OR (activeOrReflective=active))	Collaborative, short-text	No
((time<20) OR (activeOrReflective=reflective))	Collaborative, short-text	No
(time<10)	Test, review	Si

Tabla 9: Filtros generales por defecto

Para tener que definir una nueva regla de contexto en primer lugar se deberán añadir las actividades a las que se quiere aplicar, estas actividades se muestran en una lista desplegable en el que se añadirán los tipos deseados (sin repetir). A continuación pasa a definirse la condición de activación. Para definir el rasgo se muestra otra lista desplegable, si el rasgo es numérico (como el de la figura 27) se muestra un submenú para seleccionar el tipo de comparación, en cambio, si es de tipo estereotipado se muestra una lista desplegable con los posibles valores del rasgo seleccionado. Para añadirlo en una nueva condición (“or” o disyunción lógica) se seleccionará la opción “*add condition in a new recommendation*”, en cambio para completar la última condición (“and” o conjunción lógica) se seleccionará la opción “*add condition to the last recommendation*”. Finalmente se pulsará en el botón de nombre “*add*” para añadir la regla. Todas las condiciones, actividades y reglas definidas se podrán borrar en cualquier momento, facilitando en gran medida el manejo de la herramienta. El resto de reglas de todas las actividades tienen el mismo formato de introducción de condiciones de activación que los de las reglas estructurales.



Step 2 of 4: General Context Filters of the MetaActivity: *EntornoDePrueba*

MetaActivity Context Adaptation capabilities have been selected in MetaActivity Creation, please define context Adaptation capabilities

Click on Cancel Icon if you dont want to add General Context filters, Context Adaptation capabilities will be deselected from current Metaactivity, otherwise continue editing General Context filters or click on going to next step

Cancel:

Current Features Configuration:

Id.	No.	Possible activation condition	Activities	Recommend	Delete rule
1	1	(device = pda)	test review	yes	
	2	(device = phone)			

General Context Filters: click on Show or Hide icon, to show or Hide the context filter creation menu:

Default general context filter rule:

- Students using pdas or mobile phones can only perform test and review activities
- Collaborative activities and short-text exercises are not suitable for active students that have <10 minutes available
- Collaborative activities and short-text exercises are not suitable for reflective students that have <20 minutes available
- Students that have <10 minutes available can only do test exercises and reviews activities

Add default general context rule: [Add](#)

New General Context Filter Configuration:

First define type of activities to the new rule, then choose in recommended yes or no if these rule is suitable or not respectly. Finally create the new rule and add.

Type of activities: Add selected type:

Activity	Delete act.
simulation	
atenea	

Recommended:

Condition: To create a new condition: select the feature, then select the condition and add a new condition or complish current condition

If is

- Between and
- Higher or Equal than
- Less or Equal than
- Higher than
- Less than

Add condition in a new recommendation:

Add condition to the last recommendation:

Current General Context Rule Configuration:

Recommend or Not	Current conditions configuration			Delete Cond.
	No.	Possible activation condition text	Possible activation condition	
NOT Recommend the selected activity/activities if one of these condition happens:	1	If sensingOrIntuitive is intuitive	(sensingOrIntuitive=intuitive)	
	2	If RasgoNuevo1 is higher or equal than 20	(RasgoNuevo1>=20)	

Add Current General context rule: [Add](#)

[Go to next Step](#)

Figura 27: Ejemplo de página de definición de filtros generales de contexto

Finalmente una vez se han añadido las reglas estructurales deseadas y tras haber verificado el sistema la perfecta composición de las reglas generales de contexto se pasa a la página de creación de actividades.

La página de creación de actividades es la página más compleja y completa, en ella se describen las reglas que rigen la recomendación de cada una de las actividades. La tabla 10 resume los tipos de reglas que puede tener cada actividad. Cuando se seleccione un tipo de actividad se mostrarán solo los menús de especificación de sus reglas asociadas.



Tipo/s	Reglas Estructurales	Reglas Individuales	Reglas Espacios de Trabajo	Reglas Primarias	Herramientas Secundarias	Herramientas
Estructural	Si	Si	No	No	No	No
Colaborativ	No	No	Si	Si	Si	Si
Resto	No	Si	No	No	No	No

Tabla 10: Reglas aplicables a cada tipo de actividad

La figura 31 muestra un ejemplo de página de actividades. Esta distribuida en varias zonas, la zona de identificador ¹ especifica si las reglas individuales y estructurales serán aplicadas, si se selecciona no aplicar las reglas, no se mostrará siquiera la posibilidad de definir estas reglas, y las reglas del tipo cambiado realizadas previamente no se borrarán, pero al finalizar el paso no se aplicarán a las actividades.

La zona de identificador ² muestra un árbol de actividades creadas hasta el momento, esto sirve de orientación para al crear reglas estructurales saber que actividades están contenidas en una actividad de tipo estructural.

La zona de identificador ³ muestra las opciones que se pueden realizar con las actividades creadas. Se puede ver sus detalles, los cuales se mostrarán debajo de este menú, se puede realizar modificaciones a una actividad, para esta opción se mostrarán las características actuales en la parte inferior y se podrán cambiar y guardar o, si se da el caso, no guardar. Además se podrán borrar actividades creadas borrando también las actividades contenidas si son de tipo estructural.

La zona de identificador ⁴ se trata del menú de creación de actividades, en esta parte se especificarán los detalles de las actividades, se introducirán todos los campos de configuración de la actividad y se especificarán sus reglas a aplicar según su tipo. Para configurar la fecha en la que finaliza un curso se ha creado un calendario interactivo que emergerá al seleccionar el campo “*deadline*” permitiendo introducir la fecha de forma fácil. Para configurar cada una de las reglas los pasos a seguir son:

- Reglas individuales: Para cada regla solo se introducirá la condición de activación. No se permite introducir condiciones vacías.
- Reglas estructurales: Para cada regla se deberán agregar las subactividades que la compongan, definiendo para cada subactividad su tipo y su descripción y finalmente definiendo la condición de activación de la regla. Se permite introducir condiciones vacías.



- Reglas de espacios de trabajo: Para cada regla se seleccionará mediante una lista desplegable las herramientas definidas en las características generales del entorno y su condición de activación. No se permite introducir condiciones vacías.
- Reglas de herramientas primarias: Por cada regla se seleccionan una o varias herramientas colaborativas (las mismas que la anterior regla) agregando a cada herramienta una o varias herramientas primarias, y su condición de activación. Se permite introducir condiciones vacías.
- Reglas de herramientas secundarias: Por cada regla se seleccionará una o varias herramientas primarias (las creadas en la regla anterior) agregando a cada herramienta una o varias herramientas secundarias, junto con su condición de activación. Se permite introducir condiciones vacías.



Step 3 of 4: Activity menu creation of the MetaActivity: EntornoDePrueba

MetaActivity Structural Adaptation capabilities have been selected in MetaActivity Creation, you can define Structural Adaptation capabilities

Click on Cancel Icon if you dont want to add Structural Adaptataion capabilities, Structural Adaptation capabilities will be deselected from current Metaactivity, all Structural rules previously defined in all activities are not going to apply when you click finish.



1

MetaActivity Individual Adaptation capabilities have been selected in MetaActivity Creation, you can define individual Adaptation capabilities

Click on Cancel Icon if you dont want to add Individual Adaptataion capabilities, Individual Adaptation capabilities will be deselected from current Metaactivity, all individual rules previously defined in all activities are not going to apply when you click finish



CURRENT ACTIVITIES CONFIGURATION



2

Current Activity tree configuration:



- Actividad1
 - SubActividad1
 - Subactividad2
 - SubSubActv1
 - SubSubActv2
 - Subactividad3
- Actividad2

Select the search option, choose the activity and click on view details, delete or change icon.

Search options:

Activity: Type: set

3

View Details: Change Details: Delete:

CREATION OF NEW ACTIVITY DETAILS

Activity Name:

Languages description (introduce the description):

Type:

Accomplishment place:

Deadline:

Max. accomplishment time (in minutes):

4

Structural rules: click on Show or Hide icon, to show or Hide the structural rule creation menu:

New Structural rules Configuration:

First define subactivities to the new rule, then choose guidance type. Finally create the new rule and add. Language Description:

Subactivities: Name: Type:
Language Description:

Add type:

Guidance:

Rule: To create a new rule: select the feature, then select the condition and add a new or complish current condition

If is

Add condition in a new recommendation:

Add condition to the last recommendation:

Add Current context rule:

Individual rules: click on Show or Hide icon, to show or Hide the individual rule creation menu:

New Individual rules Configuration:

Create the new rule and add.

Rule: To create a new rule: select the feature, then select the condition and add a new or complish current condition

If is

Add condition in a new recommendation:

Add condition to the last recommendation:

Add Current Individual rule:

Add activity:

[Go to last step](#)

Figura 28: Ejemplo página de actividades



Una vez se han definido todas las reglas de cada una de las actividades y se ha confirmado la correcta configuración de las actividades, se pasa a la subida de contenidos mediante la siguiente página. Esta página tendrá el aspecto de la figura 29.

La zona de identificador **1** muestra el árbol de actividades que se ha especificado en la creación de actividades. Tiene el mismo formato que el de la página anterior

La zona de identificador **2** muestra el fragmento para añadir contenidos a actividades que no sean de tipo estructural ni colaborativo (este tipo de actividades no pueden tener contenidos la propia actividad). En primer lugar se seleccionará la actividad a la que se desea añadir el contenido y se seleccionará el tipo de dispositivo.

La zona de identificador **3** muestra la parte donde se añadirán los contenidos a las reglas de espacios colaborativos, se muestran las actividades colaborativas junto con sus reglas de espacios colaborativos definidas, teniendo que seleccionar la herramienta y el tipo de dispositivo.

En esta página se mostrarán también los contenidos agregados a cada actividad y reglas de espacios colaborativos permitiendo borrar un contenido o bien descargarlo.

Figura 29: Ejemplo de página de contenidos



Una vez seleccionada la herramienta o actividad y pulsado en “*Search content*” se redirigirá a otra página para subir contenidos a la actividad o regla de espacio colaborativa (ver figura 30). En esta página se podrá seleccionar un fichero, abriéndonos el menú de búsqueda de archivos del sistema operativo y permitiendo subir archivos. Los archivos actualmente agregados a la actividad o regla de espacio colaborativo podrán ser borrados.

Al subir contenidos los archivos se guardan en la herramienta en una carpeta temporal, puesto que hasta no llegar al paso final no se guarda el entorno diseñado. Esto se ha decidido así por si el diseñador escoge no guardar el curso.

Cuando se finalice de subir contenidos a la regla o actividad se volverá a la página anterior para poder añadir contenidos a otras actividades o reglas. Finalmente una vez se termine de agregar todos los contenidos se pulsará en “*Save Meta activity*” (ver figura 30) donde se pasará a almacenar todos los ficheros XML con todas las características y actividades creadas del entorno. Los contenidos que hayan sido subidos se guardarán en sus correspondientes ficheros XML de sus reglas y se traspasarán a la carpeta de contenidos de la actividad o regla de espacio colaborativo. Una vez se haya finalizado de realizar todo el guardado y actualización de los profesores en los listados de entornos de forma satisfactoria se nos mostrará una página que nos permitirá mostrar los datos del entorno creado (similar a ver entorno seleccionando el entorno), salir de la herramienta o volver al menú principal.

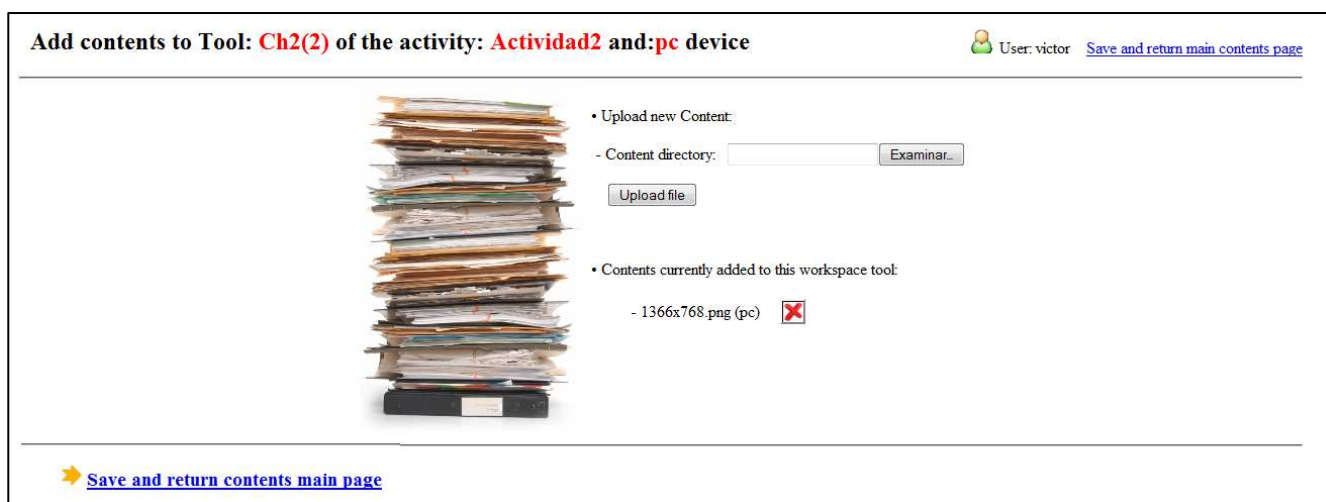


Figura 30: Ejemplo de página para añadir actividades a una determinada actividad o regla

3.4.4. Ver entorno

Al seleccionar ver entorno se nos redirigirá a una página donde se nos mostrará los detalles del entorno y actividades creadas, mostrándonos para cada actividad sus reglas y pudiendo descargar los contenidos agregados a la actividad (no se ha adjuntado una captura debido a sus enormes proporciones). Finalmente una vez se termine de visualizar todas las reglas el usuario podrá abandonar la página de visualización y regresar al menú principal.

3.4.5. Modificar entorno

La opción de modificar entorno seguirá los mismos pasos que para la creación de un nuevo curso, mostrándonos las mismas páginas pero con la configuración previamente almacenada. La modificación de las características del entorno puede repercutir sobre otras características y reglas.

Por ejemplo la eliminación de un rasgo de adaptación en la página de características generales del entorno provoca en todas las reglas que utilicen ese rasgo un cambio en su condición de activación, todas las condiciones simples que utilicen esos rasgos serán eliminadas y si se encontrasen en condiciones compuestas se recompone la condición eliminando la condición simple en la que aparezca.

También se podría dar el caso de borrar actividades o reglas que pudiesen tener contenidos asociados. Estos contenidos en la página de definición de contenidos no se muestran y una vez realizados los cambios, los contenidos desemparejados se borrarán del sistema.

Este proceso es complejo y se pueden dar múltiples consecuencias, pero se ha dado un trato especial al control de posibles fallos, la herramienta de autor asesora al diseñador para evitar problemas al realizar cambios y permitiendo al profesor realizar los cambios que estime oportunos.

Al finalizar de realizar los cambios oportunos, el proceso de aplicación de cambios consistirá en borrar el entorno original y volver a escribir el entorno nuevo con todos los cambios aplicados.

3.4.6. Eliminar entorno

La eliminación de un entorno se realiza en la página principal, para ello solo se deberá pulsar en el botón de eliminación del curso que se desea borrar. La herramienta de autor nos pedirá confirmación y tras confirmar su eliminación se eliminarán sus ficheros



XML, contenidos asociados y se eliminarán del fichero *listados.xml* a todos los profesores que disponían de privilegios sobre el entorno.

3.5. Pruebas

Durante la realización del proyecto se fue probando cada una de las funcionalidades de forma gradual según se iban desarrollando (tal y como establece el proceso unificado), esto nos ha permitido tener un gran control de fallos y errores. Una vez finalizado el proyecto se realizaron múltiples pruebas para probar todos los casos, tras probar su correcto funcionamiento podemos concluir que la herramienta es bastante estable y cumple todas las expectativas. A su vez se ha comprobado su compatibilidad con los diferentes navegadores, Internet Explorer, Mozilla Firefox, Google Chrome, dándose en todos ellos un resultado correcto en su aspecto gráfico sin problemas de compatibilidad entre las diferentes versiones.



4. Conclusiones y trabajo futuro

En este capítulo se evalúan los resultados de todo el desarrollo. Se ha dividido en una parte donde se han enumerado los logros conseguidos y se comprueban los requisitos iniciales. En segunda parte se argumenta posibles mejoras y trabajos futuros que se podrían realizar en la herramienta.

4.1. Conclusiones

Al finalizar el proyecto se ha podido comprobar que se alcanzaron todos los requisitos propuestos. Se ha conseguido crear una herramienta estable y fiable que nos permita generar los entornos de trabajo adaptativos de forma relativamente fácil. Los requisitos propuestos durante la fase de análisis de requisitos se han logrado de forma satisfactoria. Funcionalidades extra como la gestión de la sesión del usuario (activación y desactivación) y la opción de crear entornos adaptativos que puedan ser modificados por varios profesores, permiten un mayor número de opciones a tener en cuenta en la interacción con la herramienta. Se ha dado mayor soporte para la creación de actividades colaborativas, teniendo que haberse realizado cambios en la estructura de los datos almacenados para tener en cuenta las nuevas características.

Las dificultades y conclusiones aportadas en el desarrollo del proyecto han sido:

1. La herramienta de autor facilita en gran medida la definición de las características de los entornos, sin una herramienta que facilite la operación las características se tendrían que definir en los distintos ficheros XML las reglas que rijan cada entorno. La herramienta de autor presenta todas las opciones disponibles con el objetivo de no tener que modificar los datos de estos ficheros sin la herramienta.
2. La generación de entornos adaptativos móviles no es una tarea fácil, la cantidad de características y reglas que definen este tipo de entornos hacen que se requiera de un previo conocimiento de las características que conforman el entorno. Se han encontrado problemas para poder agregar estas reglas de forma fácil y más intuitiva de la que ya había.
3. La gran cantidad de contenidos mostrados en cada una de las páginas y su gran cantidad de código de control ha ocasionado problemas para controlar pequeños fallos o errores ocurridos durante el desarrollo del proyecto. Se ha resuelto este problema tratando de modularizar lo máximo posible.



4. La subida y descarga de archivos al servidor no ha sido una tarea fácil debido a las limitaciones que ofrecen sus librerías por motivos de seguridad, una de sus repercusiones es la obligatoria separación en diferentes páginas para realizar la subida de contenidos.

4.2. Trabajos futuros

Adaptar los entornos creados con la herramienta de autor al sistema CoMoLE. Durante la realización del proyecto se han modificado las funciones internas de lectura y escritura de entornos para poder dar soporte a las nuevas funcionalidades planteadas. El sistema de CoMoLE leerá de los ficheros XML las características de los entornos generados por la herramienta de autor para definir los espacios de trabajo. Al haber realizado modificaciones, el sistema de recomendación de CoMoLE deberá adaptarse a estos cambios, para poder realizar las nuevas recomendaciones (sobre todo en entornos colaborativos).

Otro posible trabajo que se podría realizar sería una herramienta de evaluación de formatos de los ficheros XML, cuya función principal sería detectar posibles fallos de construcción de ficheros XML para evitar fallos en la lectura de estos ficheros. La herramienta de autor crea y carga los contenidos de estos ficheros de forma satisfactoria, pero podría darse el caso de generar ficheros XML corruptos o bien que otra herramienta modificase de forma incorrecta un fichero.

Dar soporte a múltiples idiomas, permitiendo al diseñador seleccionar el idioma en el que va a estar diseñada la interfaz, cambiando los mensajes de error y todos los avisos que genera la herramienta.



5. Bibliografía

Martín, E. (2008). *Creación de entornos adaptativos móviles: Recomendación de actividades y generación dinámica de espacios de trabajo basadas en información sobre usuarios, grupos y contextos.*

Bloch, J. (2008). *Effective Java*. Sun Microsystems, Inc.

Jalón, J. G. (Abril de 1999). *Aprenda Servlets de Java como si estuviera en segundo.*
Url: "<http://mat21.etsii.upm.es/ayudainf/aprendainf/JavaServlets/servlets.pdf>"

Moreno, J. A. (1998). *HTML 4: Guía de referencia y tutorial*. Ra-Ma.

Quintas, A. F. (2005). *Java 2 Manual de usuario y tutorial*. Ra-Ma.



6. Anexos

A. Manual de instalación

Para poder utilizar la herramienta de autor, la cual se ha explicado su funcionamiento en este proyecto, se requiere instalar una serie de programas. Por ello a continuación se va a proceder a explicar los pasos para instalar estos programas requeridos.

Requisitos

Para realizar la instalación de la herramienta se requiere:

1. Un ordenador con una IP pública para instalar el servidor Web
2. Conexión a internet para descargar las aplicaciones

En cuanto a la instalación de aplicaciones necesarias:

1. Máquina virtual de java, JDK 1.5 o superior.
2. Servidor Web de *servlets* (Glassfish o Tomcat)

Proceso de instalación

1. Máquina virtual de java

La herramienta se ha diseñado en Java, por lo que se requiere que esté instalado en la máquina que va a hacer las operaciones de servidor. Durante la realización del proyecto se ha utilizado la versión del JDK 1.6, pero la versión 1.5 servirá también. La máquina virtual de Java se puede descargar de forma gratuita desde la página oficial de Sun: <http://java.sun.com/>

2. Servidor Web

En último lugar es necesario instalar el servidor Web, el empleado durante la realización del proyecto fue Tomcat. El servidor Web nos permitirá atender las peticiones en formato http que se reciban al servidor. Al generarse las páginas Webs de forma dinámica el servidor ejecuta el programa en java y le pasa la información de la petición. Ese programa se ejecuta y crea la página Web dinámica, la cual se envía al cliente.



Existen multitud de servidores Web disponibles y de libre distribución. El más empleado es el servidor apache, por ser de código gratuito y abierto. Para crear la aplicación Web usando *servlets* se requiere de un servidor Web que lo soporte, el servidor Web empleado ha sido Tomcat, el cual se puede descargar de forma gratuito en: <http://tomcat.apache.org/>.

Para instalar Tomcat se han de seguir los siguientes pasos:

En Unix:

1. Extraer Tomcat una vez descargado en un directorio /ruta_tomcat
2. Definir variable de ubicación de tomcat: Comando: `setenv CATALINA_HOME /ruta_tomcat`
3. Ejecutar tomcat en la consola: Comando: `$CATALINA_HOME /bin/startup.sh`

En Windows:

- Definir las variables de entorno JAVA_HOME y JRE_HOME (ambas estarán ubicadas de ordinario en la ruta C:\Program Files\Java\).
Para definir las variables de entorno pulsar botón secundario en equipo, seleccionar propiedades. A continuación ir a configuración avanzada y añadir los valores a variables del sistema.
- Finalmente ejecutar en el lugar de instalación tomcat en la carpeta bin el archivo startup.bat.

Una vez se ha instalado el servidor de forma correcta pasamos a detener el servicio (en unix: `$CATALINA_HOME/bin/shutdown.sh` y en Windows mediante la ejecución del fichero `ruta_tomcat/bin/shutdown.bat`). A continuación colocamos el fichero del proyecto “FinalProject.war” en la carpeta `ruta_tomcat/webapps` y volvemos a lanzar el servidor. Finalmente colocamos la carpeta “xmls-authoring” en la carpeta `ruta_tomcat/bin`.

Para cargar el proyecto escribiremos en el navegador: `http://localhost:8080/FinalProject` lanzándose nuestra aplicación de forma local gracias al servidor Web tomcat.

