



UNIVERSIDAD DE CÓRDOBA
ESCUELA POLITÉCNICA SUPERIOR

PROYECTO FIN DE CARRERA
INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

Editor colaborativo de código fuente en lenguaje C

Manual Técnico

Alumno: Antonio Jesús González León
Director: Cristóbal Romero Morales
Fecha: 3 de septiembre de 2012

D. CRISTÓBAL ROMERO MORALES
Profesor Titular de la Escuela Politécnica Superior
Departamento de Informática y Análisis Numérico
Universidad de Córdoba

CERTIFICA: Que la memoria titulada “*Editor colaborativo de código fuente en lenguaje C*” ha sido realizada por ANTONIO JESÚS GONZÁLEZ LEÓN bajo mi dirección y constituye su Proyecto de Fin de Carrera de Ingeniería Técnica Informática de Gestión.

En Córdoba, a 3 de septiembre de 2012

D. CRISTÓBAL ROMERO MORALES
Director del proyecto

Agradecimientos

- A D. Cristóbal Romero Morales, profesor titular de la Escuela Politécnica Superior, por su colaboración y tiempo prestado para la realización de este proyecto.
- A todos mis profesores, desde preescolar hasta la Universidad, por hacerme el regalo más valioso como es el conocimiento.
- A mi novia, familia y amigos por todo el apoyo y comprensión demostrada durante todo este tiempo y por todo el tiempo que no he podido estar con ellos y que me hubiera gustado dedicarles.
- A mis antiguos compañeros de trabajo por todo el conocimiento adquirido gracias a ellos y sus valiosos consejos.

Índice general

1. Introducción al problema	1
1.1. Introducción	1
1.2. Estudio del problema	3
1.2.1. Identificación del problema real	3
1.2.2. Identificación del problema técnico	4
1.3. Objetivos	9
1.4. Antecedentes	10
1.5. Restricciones	11
1.5.1. Factores dato	11
1.5.2. Factores estratégicos	12
1.6. Recursos	13
1.6.1. Recursos humanos	14
1.6.2. Recursos Hardware	14
1.6.3. Recursos Software	14
2. Análisis de plataformas e-learning	17
2.1. Moodle	18
2.1.1. Introducción	18
2.1.2. Requerimientos	18
2.1.3. Seguridad	19
2.1.4. Soporte	19

2.1.5.	Facilidad de uso	20
2.1.6.	Funciones	20
2.1.7.	Administración	20
2.1.8.	Interoperatividad	20
2.1.9.	Aplicaciones soportadas	21
2.2.	WebCT	21
2.2.1.	Introducción	21
2.2.2.	Requerimientos	21
2.2.3.	Seguridad	22
2.2.4.	Soporte	23
2.2.5.	Facilidad de uso	23
2.2.6.	Funciones	23
2.2.7.	Administración	24
2.2.8.	Interoperatividad	24
2.2.9.	Aplicaciones soportadas	24
2.3.	Ilias	25
2.3.1.	Introducción	25
2.3.2.	Requerimientos	25
2.3.3.	Seguridad	26
2.3.4.	Soporte	26
2.3.5.	Facilidad de uso	26
2.3.6.	Funciones	26
2.3.7.	Administración	26
2.3.8.	Interoperatividad	27
2.3.9.	Aplicaciones soportadas	27
2.4.	Proyecto Sakai	27
2.4.1.	Introducción	27
2.4.2.	Requerimientos	27

2.4.3.	Seguridad	28
2.4.4.	Soporte	29
2.4.5.	Facilidad de uso	29
2.4.6.	Funciones	29
2.4.7.	Administración	29
2.4.8.	Interoperatividad	30
2.4.9.	Aplicaciones soportadas	30
2.5.	Atutor	30
2.5.1.	Introducción	30
2.5.2.	Requerimientos	30
2.5.3.	Soporte	31
2.5.4.	Facilidad de uso	32
2.5.5.	Funciones	32
2.5.6.	Administración	32
2.5.7.	Interoperatividad	32
2.5.8.	Aplicaciones soportadas	32
2.6.	eGroupWare	33
2.6.1.	Introducción	33
2.6.2.	Requerimientos	33
2.6.3.	Seguridad	34
2.6.4.	Soporte	35
2.6.5.	Facilidad de uso	35
2.6.6.	Funciones	36
2.6.7.	Administración	36
2.6.8.	Interoperatividad	37
2.6.9.	Aplicaciones soportadas	37
2.7.	Comparativa de la plataforma para la gestión docente	38
2.7.1.	Acceso	39

2.7.2. Seguimiento	40
2.7.3. Soporte del Software	41
2.7.4. Funciones a disposición del alumno	41
2.7.5. Administración	43
2.7.6. Facilidad de uso	44
2.7.7. Otros	45
2.8. Elección de la plataforma para la gestión docente	46
3. Especificación de requisitos	49
3.1. Descripción del dominio de información del sistema	49
3.1.1. Subsistema Wikicode-Mod	50
3.1.2. Subsistema Wikicode-Synchronizer	54
3.2. Especificación de la interfaz	56
3.3. Modelo funcional	60
3.3.1. Identificación de actores	60
3.3.2. Casos de uso de trazo grueso	62
3.3.3. Diagramas de caso de uso	68
3.3.4. Matriz requisito / Caso de uso	73
3.3.5. Especificación de los casos de uso	75
3.4. Modelo conceptual	85
3.4.1. Diccionario de datos para el sistema Wikicode-Mod	85
3.4.2. Diccionario de datos para el sistema Wikicode-Synchronizer	111
3.4.3. Diagrama de clases	114
3.5. Especificación del Modelo Objeto-Comportamiento	115
3.5.1. Diagrama de colaboración: Configuración de la Wikicode	115
3.5.2. Diagrama de colaboración: Visualización del Código Fuente	116
3.5.3. Diagrama de colaboración: Compilación de Código Fuente	117
3.5.4. Diagrama de colaboración: Edición de código	117

4. Diseño del sistema	119
4.1. Diseño de la base de datos	119
4.1.1. Modelo Conceptual	119
4.1.2. Modelo Relacional	122
4.1.3. Normalización del modelo	123
4.2. Descripción del resultado final del interfaz	124
4.2.1. Descripción del interfaz	124
4.2.2. Descripción de las pantallas del interfaz	125
4.3. Especificación procedimental	133
4.3.1. Diagrama de actividades para el sistema Wikicode-Mod	134
4.3.2. Diagrama de actividades para el sistema Wikicode-Synchronizer	141
4.4. Modelo de paquetes	143
4.4.1. Paquete Cliente de Wikicode	143
4.4.2. Paquete Servidor/Cliente Wikicode-Mod de Wikicode	143
4.4.3. Paquete Servidor Wikicode-Synchronizer de Wikicode	144
4.4.4. Diagrama de paquetes	145
5. Implementación del sistema	147
6. Pruebas del sistema	149
6.1. Objetivos de las pruebas	149
6.2. Diseño de los casos de pruebas	150
6.3. Pruebas estructurales de la caja blanca	150
6.4. Pruebas funcionales o de la caja negra	151
6.4.1. Análisis de los valores límites	151
6.5. Pruebas del desarrollador	152
6.5.1. Pruebas con un curso real	152
6.5.2. Prueba de compilación utilizando Cross Compiling	155
6.5.3. Comprobación de estadísticas	155

6.6. Corrección de errores	156
6.7. Prueba de aceptación	157
7. Conclusiones y futuras mejoras	159
7.1. Conclusiones	159
7.2. Posibles ampliaciones o mejoras	160

Índice de figuras

2.1. Plataforma e-learning Moodle	19
2.2. Plataforma e-learning WebCT	22
2.3. Plataforma e-learning Ilias	25
2.4. Plataforma e-learning Sakai	28
2.5. Plataforma e-learning ATutor	31
2.6. Plataforma e-learning EGroupWare	34
3.1. Gráfico de Subsistemas	50
3.2. Gráfico de Subsistema Wikicode-Mod	51
3.3. Gráfico de Subsistema Wikicode-Synchronizer	55
3.4. Estructura general de la interfaz del curso	57
3.5. Estructura de la interfaz de creación	58
3.6. Estructura de la interfaz de vista	58
3.7. Estructura de la interfaz de edición del código fuente	59
3.8. Estructura de la interfaz del histórico de un código fuente	59
3.9. Estructura de la interfaz de la información estadística de una Wikicode	60
3.10. Diagrama inicial para el sistema Wikicode	68
3.11. Diagrama Caso de uso C1	68
3.12. Diagrama Caso de uso C2	69
3.13. Diagrama Caso de uso C3	69
3.14. Diagrama Caso de uso C4	70

3.15. Diagrama Caso de uso C5	70
3.16. Diagrama Caso de uso C6	71
3.17. Diagrama Caso de uso C7	71
3.18. Diagrama Caso de uso C8	72
3.19. Diagrama Caso de uso C9	72
3.20. Diagrama Caso de uso C10	73
3.21. Class mod_wikicode_renderer	87
3.22. Class page_wikicode	89
3.23. Class page_wikicode_view	91
3.24. Class page_wikicode_edit	92
3.25. Class page_wikicode_log	93
3.26. Class page_wikicode_search	94
3.27. Class page_wikicode_create	96
3.28. Class page_wikicode_compile	97
3.29. Class page_wikicode_diff	98
3.30. Class page_wikicode_history	100
3.31. Class page_wikicode_restoreversion	101
3.32. Class page_wikicode_save	102
3.33. Class page_wikicode_viewversion	103
3.34. Class page_wikicode_confirmrestore	104
3.35. Class page_wikicode_admin	105
3.36. Class html_table	107
3.37. Class html_write	108
3.38. Class html_cell	110
3.39. Class html_row	111
3.40. Class editorlib	113
3.41. Diagrama de clases	114
3.42. Diagrama de colaboración Configuración de la Wikicode	115

3.43. Diagrama de colaboración Visualización del Código Fuente	116
3.44. Diagrama de colaboración Compilación de Código Fuente	117
3.45. Diagrama de colaboración Edición de código	118
4.1. Esquema EE-R del modelo conceptual del módulo Wikicode	120
4.2. Diagrama relacional para el módulo Wikicode	123
4.3. Pantalla principal de un curso Moodle añadiendo una Wikicode	125
4.4. Pantalla de configuración y parametrización Wikicode	126
4.5. Pantalla de ayuda a la configuración Wikicode	126
4.6. Pantalla cabecera Wikicode	127
4.7. Pantalla visualización Wikicode	128
4.8. Pantalla editor Wikicode.	129
4.9. Pantalla desbloqueo Wikicode.	130
4.10. Pantalla chat Wikicode.	131
4.11. Pantalla compilación Wikicode.	131
4.12. Pantalla histórico Wikicode.	132
4.13. Pantalla restauración histórico Wikicode.	132
4.14. Pantalla comparación histórico Wikicode.	133
4.15. Pantalla estadística Wikicode.	133
4.16. Actividad crear, configurar y parametrizar la Wikicode	134
4.17. Actividad visualizar código fuente	135
4.18. Actividad editar código fuente	136
4.19. Actividad chat	137
4.20. Actividad compilar código fuente	138
4.21. Actividad visualizar histórico	139
4.22. Actividad visualizar estadísticas	140
4.23. Actividad sincronizar código	141
4.24. Actividad actualizar estadísticas	142

4.25. Diagrama de paquetes	145
5.1. Diagrama de Componentes	147
6.1. Pantalla principal de un curso Moodle	152
6.2. Edición de código con una función bloqueada	153
6.3. Edición de código. Descarga y ejecución de una aplicación creada.	154
6.4. Edición de código. Chat.	154
6.5. Edición de código. Cross Compiling.	155
6.6. Log de información de una Wikicode.	156

Índice de cuadros

2.1. Comparativa de Acceso	39
2.2. Comparativa de Seguimiento	40
2.3. Comparativa de Soporte del Software	41
2.4. Comparativa de Funciones a disposición del alumno	43
2.5. Comparativa de Administración	43
2.6. Comparativa de Facilidad de uso	45
2.7. Comparativa de otras características	46
3.1. Descripción del Tutor	61
3.2. Descripción del Tutor	61
3.3. Descripción del Módulo Wikicode-Mod	62
3.4. Descripción del Módulo Wikicode-Synchronizer	62
3.5. Caso de Uso C1: Creación y Parametrización de la actividad Wikicode . . .	64
3.6. Caso de Uso C2: Visualización del código desarrollado	64
3.7. Caso de Uso C3: Edición del Código	65
3.8. Caso de Uso C4: Bloqueo del Código	65
3.9. Caso de Uso C5: Desbloqueo del Código	65
3.10. Caso de Uso C6: Salvar Código	66
3.11. Caso de Uso C7: Interacción mediante Chat con otros usuarios	66
3.12. Caso de Uso C8: Compilación de código fuente	66
3.13. Caso de Uso C9: Consulta de histórico y restauración de versiones	67

3.14. Caso de Uso C10: Consulta de Estadísticas	67
3.15. Matriz requisito / caso de uso	74
3.16. Especificación caso de uso C1	75
3.17. Especificación caso de uso C2	76
3.18. Especificación caso de uso C3	78
3.19. Especificación caso de uso C4	79
3.20. Especificación caso de uso C5	80
3.21. Especificación caso de uso C6	81
3.22. Especificación caso de uso C7	82
3.23. Especificación caso de uso C8	83
3.24. Especificación caso de uso C9	84
3.25. Especificación caso de uso C10	85

Capítulo 1

Introducción al problema

1.1. Introducción

Los avances en la tecnología en los últimos años han aportado nuevas perspectivas y desafíos al proceso de enseñanza y aprendizaje. Ya nadie pone en duda que el ordenador es un medio didáctico muy potente y que puede, y de hecho lo está haciendo, cambiar la forma de enseñar de los profesores y el modo de aprender de los alumnos. Una parte fundamental en este nuevo proceso de aprendizaje es sin duda Internet. Con la facilidad de adquisición de la información en esta red global, el profesor pasa a tener el papel fundamental de ayudar al alumno a interpretar los datos, a relacionarlos y a contextualizarlos. Sin embargo, el volumen de informaciones no permite alcanzar todos los contenidos que caracterizan un área de conocimiento. Por lo tanto, profesores y alumnos necesitan aprender a aprender cómo acceder a la información, dónde buscarla y qué hacer con ella.

Ahora, cuando un alumno procede a buscar información en Internet o trabaja con un software educativo, el protagonista ya no es el profesor sino el propio alumno, quien por su cuenta investiga, comprende, lee y aprende.

Para poner en práctica todo esto, y ayudar tanto al profesor como al alumno, existen un gran número sistemas de gestión de aprendizaje. Existen numerosas plataformas (LMS, por sus siglas en inglés), entre las que podemos nombrar: Atutor [atu12], WebCT [wct12], Moodle [mdl12], Ilias [ili12], etc. Aunque cada una de ellas tiene sus propias peculiaridades, todas poseen unos fundamentos teóricos comunes. Dichos fundamentos están relacionados con el uso de aprendizaje colaborativo, con la gestión y organización de los procesos de trabajo así como con la importancia del proceso de aprendizaje.

Sin embargo, aunque todas estas plataformas ofrecen una serie de herramientas para ser utilizadas en el aprendizaje colaborativo, no todas son igual de propicias para dicho aprendizaje.

El aprendizaje colaborativo es un tipo de aprendizaje en grupo en el que los alumnos aprenden gradualmente, y en el que no existe ningún tipo de competencia entre ellos. Dicho de otro modo, los alumnos colaboran para mejorar el conocimiento grupal y pueden ser elementos de ayuda para el aprendizaje de otros compañeros.

Por ende, es evidente que el elemento básico para aprender colectivamente se trata de la interacción y la comunicación entre los alumnos, ayudándose entre todos del conocimiento común y aportando individualmente.

En cuanto a la gestión y organización, la tecnología e-learning no nos proporciona únicamente la ayuda para la interacción de los alumnos, sino que aprovecha sus ventajas para gestionar, organizar y facilitar el trabajo del grupo en las tareas de aprendizaje. Además, nos permite monitorizar, analizar y mejorar los procedimientos de trabajo.

Sin apartarnos de la línea argumental de esta introducción, también podemos observar como la programación informática ha pasado a ser un área estratégica importante dentro de cualquier sector técnico o científico en nuestra sociedad. Es por este motivo que no se da docencia de ella únicamente en los grados informáticos, sino que cada vez es más común su derivación hacia otras áreas.

Sin embargo, es bastante amplia la dificultad que supone para alguien que no haya estado en contacto con el sector informático aprender a programar en la gran mayoría de ocasiones. Esto no sucede con todos los casos, muchas veces se puede aprovechar la capacidad de algún alumno que ya maneje estos conocimientos con anterioridad o simplemente tenga la habilidad para ello en beneficio del resto de compañeros.

Es por ello que una vez conocidas las principales características de las plataformas e-learning, podemos observar que se trata de una herramienta idónea para que los alumnos más aventajados sobre el sector de la programación puedan compartir sus conocimientos e interactuar de una manera que el profesor considere oportuna con el resto de compañeros. A su vez, se trata de un sistema perfecto para que el profesor pueda evaluar los conocimientos de todos los alumnos sin necesidad de valorarlos individualmente.

1.2. Estudio del problema

En esta sección se tratarán de identificar las necesidades que pretendemos cubrir así como de definir el principal objetivo a alcanzar con el desarrollo del proyecto, mostrando las alternativas y posibilidades mediante las cuales será posible conseguir el resultado que deseamos.

La enumeración de las necesidades y la definición del problema podrán llevarse a cabo desde dos puntos de vista diferentes. Por un lado, se intentará identificar el problema desde la perspectiva del cliente (problema real) y, por otro, se intentará presentar la forma de dar solución a dichas necesidades desde un punto de vista más técnico.

1.2.1. Identificación del problema real

Como hemos comentado anteriormente, este trabajo se basa fundamentalmente en la colaboración de los alumnos en una comunidad virtual con respecto a la programación informática. En el campo de la enseñanza es tan importante el producto final como el proceso que se ha seguido para obtener dicho conocimiento.

Actualmente se sigue una gran cantidad de distintos métodos lectivos para hacer más llevadero el conocimiento de este área a todo tipo de alumnos. El resultado normalmente es siempre el mismo, un alumno que ya sabe programar con suficiencia realiza el trabajo que el profesor ha preparado para el aprendizaje del grupo, y el resto de usuarios más inexpertos simplemente copia esa práctica y la entrega. La consecuencia de esto es simple, a la hora de evaluar los conocimientos individualmente las calificaciones son bajas, pues en ningún momento han aprendido a programar y en muchos casos ni se ha llegado a hacer el intento.

La programación informática, al contrario que otras materias, requiere de un conocimiento informático previo puesto que el alumno debe de ser capaz de montar un entorno de desarrollo que muchas veces no llega ni a conseguir. Muchos alumnos tienen adversidad por todo lo relacionado con la materia y despejarle el camino es la manera más productiva de encontrar buenos resultados.

De esta manera, se ha desarrollado un modelo que nos permite que el usuario programe desde su casa, cuando considere oportuno, sin necesidad de instalar nada en su equipo. Del mismo modo, este sistema obliga al alumno a no copiar e incluso a colaborar para realizar prácticas informáticas. Dicho de otro modo, el sistema ha conseguido despejar el camino y hacer más activo al alumno.

1.2.2. Identificación del problema técnico

Para describir los condicionantes del problema técnico se utilizará una técnica denominada PDS (Product Design Specification) que contiene los apartados detallados a continuación.

1.2.2.1. Funcionamiento

El modelo desarrollado proporcionará un nuevo módulo para el entorno Moodle destinado a crear un entorno de desarrollo de programación en C para una edición colaborativa. Dicho módulo deberá tener varias opciones de creación para el profesor y que pueda decidir que tipo de metodología usar (grupal, individual, grupal colaborativa...). A su vez, aprovechando la ventaja que nos ofrece Moodle, proporcionar una serie de estadísticas que puedan resultar útiles de analizar de cara a un futuro.

La edición de código fuente será colaborativa y en tiempo de ejecución, de modo que los alumnos puedan contemplar que van programando sus compañeros mientras ellos están desarrollando. Debe facilitar la comunicación entre los alumnos y a la vez proporcionar las herramientas correspondientes a cualquier entorno: Compilación, Prueba de ejecutable, Histórico de versiones, etc.

1.2.2.2. Entorno

Como hemos comentado en el punto anterior, el módulo desarrollado se encuentra dentro de Moodle. Por tanto, para acceder a dicho módulo habrá que acceder previamente al curso que debe encontrarse bajo un entorno Moodle. Una vez dentro, el usuario podrá elegir entre varias opciones:

- Si el usuario es profesor/administrador podrá crear tantas Wikis de edición de código como considere oportuno. En su creación, podrá configurar los parámetros del modo que considere adecuado. Además podrá consultar y modificar cada una de ellas cuando considere oportuno.
- Si el usuario es un alumno podrá acceder a todas las Wikis de edición de código que el profesor haya considerado oportuno. Una vez dentro de ella podrá elegir:
 - Visualizar el código fuente.
 - Editar el código fuente.
 - Bloquear o desbloquear partes del código para su edición.

- Hablar con el resto de alumnos que estén modificando el código fuente.
- Compilar el código fuente.
- Descargar el ejecutable creado.
- Consultar el histórico de versiones y restaurar, si así lo desea, alguna versión más antigua.

1.2.2.3. Vida esperada

La estimación de la vida de un software es un parámetro muy importante pues puede que no merezca la pena realizar un gran esfuerzo en el diseño de un software si después va a tener una vida muy corta. Sin embargo, este parámetro es difícil de calcular pues depende de varios factores, como pueden ser su mantenimiento o nuevas necesidades por parte de los usuarios.

En cuanto al módulo de Moodle, al tratarse de un software desarrollado para una plataforma determinada de e-learning, se estima que la vida esperada de la aplicación será totalmente dependiente y estará ligada a la vida esperada de dicha plataforma. Debido al auge que ha tenido Moodle en los últimos años así como a la cantidad de ventajas que ofrece, se puede suponer que tendrá una vida muy elevada por lo que nuestra aplicación también la tendrá.

Sin embargo esto no quiere decir que la aplicación no pueda modificarse, ampliarse, añadir nuevas funcionalidades e incluso variar su front-end¹ para adaptarlo a otros sistemas e-learning.

1.2.2.4. Ciclo de mantenimiento

Aunque no se espera que el software desarrollado sufra modificaciones, cabe la posibilidad de que este sea modificado o actualizado en futuros proyectos e investigaciones, pues está diseñado para que se puedan añadir nuevas características, e incluso dar la posibilidad de seleccionar un lenguaje de programación distinto. Además, se podrá utilizar la potencia de Moodle para permitir la importación de ficheros, la edición de proyectos de varios ficheros, la exportación, etc.

¹**Front-end:** Parte del software que interactúa con el o los usuarios.

1.2.2.5. Competencia

Pese a que actualmente se está produciendo un gran auge de las comunidades virtuales, aún no hay casos de aplicaciones externas que mezclen una comunidad virtual y un editor de código fuente colaborativo. Si existen numerosos casos de editores colaborativos, los cuales pueden ser online como el nuestro o mediante software. Ejemplo de ellos pueden ser Ethercodes[etc12], Notapipe[npp12] o Amy Editor[amy12]. Estos casos serán comentados con mayor profundidad en capítulos posteriores.

1.2.2.6. Aspecto Externo

La aplicación desarrollada será presentada en CD-ROM, pues aunque la capacidad de almacenamiento que requiere la misma no es muy elevada, se considera más oportuno este dispositivo por ser más resistente, duradero y manejable.

Además, la aplicación irá acompañada de un manual de usuario, en el que se explicará la instalación, configuración y manejo del programa de la forma más sencilla y clara posible. Junto con este manual, se entregará un manual de código, por si el usuario desea realizar alguna consulta más técnica.

1.2.2.7. Estandarización

El diseño de la aplicación se ha realizado de forma tal que la aplicación usuario pueda ejecutarse en cualquier Sistema Operativo, ya sea de tipo Windows, Linux, Mac o de cualquier otro. Además, se han procurado utilizar los elementos de la forma más estándar posible, como es el caso del mismo Moodle así como la filosofía de comunicación utilizada (cliente-servidor), los lenguajes de programación utilizados (PHP y Javascript), la base de datos (mySQL), el formato en el que se ofrecen los ejecutables (.exe para Windows, .out para los OS basados en Unix), etc.

1.2.2.8. Calidad y fiabilidad

La calidad y la fiabilidad son campos que cada vez van tomando mayor importancia en la sociedad. Por ello, ambos han sido muy tenidos en cuenta a la hora de diseñar la aplicación.

De esta manera, este proyecto ha sido desarrollado identificando los puntos o elementos de riesgo o de mayor probabilidad de fallo e intentando minimizar esta probabilidad.

El principal riesgo es la sincronización de código, tanto con el sistema como con el resto de usuarios, de modo que dos usuarios que no estén comunicados entre si puedan estar intentando modificar lo mismo. Para esto se han creado una serie de restricciones, dotando al sistema de bloqueos y desbloqueos que son debidamente comunicados a los usuarios. El sistema se encargará de que ningún usuario pueda modificar una parte de código bloqueada por otro usuario.

Otro de los posibles riesgos es la sobrecarga del servidor, ya que varios alumnos pueden estar intentando paralelamente comunicarse con este para ir guardando su código. Sin embargo, el sistema ha sido programado para comunicarse con el servidor exclusivamente en situaciones críticas, como puede ser el bloqueo o desbloqueo de funciones o las modificaciones de bloques en el código. De este modo, el sistema queda libre de la mayoría del porcentaje de carga que este módulo le pueda ocasionar.

1.2.2.9. Programa de tareas

A continuación se va a desarrollar el programa detallado de la realización del proyecto a lo largo del tiempo:

- **Fase de preparación:** Esta primera fase supone un proceso de investigación y de documentación del material, tanto gráfico como interactivo, que se ha utilizado en el desarrollo del software y que se expondrán más adelante en el apartado de bibliografía.
- **Elección de plataforma e-learning:** En dicha fase se realizará un estudio de las distintas plataformas e-learning existentes en la actualidad así como un estudio comparativo de las mismas. De esta manera, se expondrá cual de las plataformas ha sido elegida y cuales han sido los motivos que han dado lugar a dicha elección.
- **Análisis y diseño del módulo y del editor colaborativo:** En esta fase se realizará el análisis y el diseño tanto del módulo que sirve como interfaz para acceder a la aplicación de edición de código como de la aplicación que realiza el propio editor colaborativo.
- **Implementación del módulo y del editor colaborativo:** En esta fase se llevará a cabo la implementación del software que permitirá acceder al editor desde la plataforma seleccionada así como la implementación del editor de código fuente colaborativo.

- **Realización de pruebas:** Para asegurar el correcto comportamiento de toda la aplicación se deberán realizar una serie de pruebas y un análisis exhaustivo de los resultados.
- **Documentación de la memoria y anexos:** En esta fase se ha elaborado la memoria, el manual de usuario y el manual de código. La elaboración de la mayoría de las partes de la memoria se ha realizado conjuntamente con el diseño de la aplicación para una mayor coherencia del documento.

1.2.2.10. Pruebas

Las pruebas que se van a realizar comenzarán por las de la caja negra o pruebas funcionales. Se tendrá especial cuidado con los valores límite y con la reacción del sistema a cualquier tipo de acción que realice el usuario.

Se realizará también la prueba de la caja blanca en aquellas partes donde la caja negra no ha obtenido unos resultados óptimos o en aquellas partes donde haya un funcionamiento complicado.

Por último, se realizará la prueba de aceptación, en donde intervendrán una o varias personas distintas al autor, y la prueba del sistema, para comprobar que se cumplen todos los requisitos y que el propio sistema responde a situaciones límite y de sobrecarga.

La aplicación se dará por válida cuando se hayan corregido todos los errores encontrados en las pruebas anteriores y se cumplan con los objetivos para los cuales se diseñó.

1.2.2.11. Seguridad

El módulo desarrollado se ha diseñado de tal forma que el usuario no pueda cometer errores que afecten al sistema y si los comete que éste no se vea involucrado. A su vez se limitará al usuario a realizar una serie de acciones que puedan poner en peligro la estabilidad del sistema.

De la misma manera, como la aplicación es de libre distribución no tiene sentido la seguridad contra copias no autorizadas.

1.3. Objetivos

Con la realización de este proyecto se han intentando alcanzar los siguientes objetivos:

- Utilizar la filosofía Open Source, término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de poder acceder al código, que a las cuestiones éticas y morales las cuales se destacan en el software libre.
- Crear un sistema, que pese a estar embebido dentro un sistema de E-learning, nos proporcione una imagen similar a la de una plataforma de desarrollo. Entre sus características nos debemos encontrar:
 - Capacidad de almacenar un histórico de código fuente al igual que un servidor SVN.
 - Mantener toda la filosofía del sistema e-learning. No debe de modificar su interfaz, su sistema de instalación, su sistema de permisos y no planteará problema de compatibilidad más allá que los del propio sistema e-learning.
 - Utilización de un editor que reconozca palabras claves, funciones, estructuras, símbolos, cadenas, etc. de modo que parsee² el código y ayude al usuario dando formato a este.
 - Reconocer en el editor estructuras más complejas como comentarios o definiciones y mostrarlos en un color diferente para que el usuario pueda distinguirlos.
 - Tener un sistema de bloqueos parciales dentro del código, delimitado por funciones, de modo que se permita la programación simultánea del mismo fichero. El usuario podrá ver claramente las partes que son modificables y a la vez podrá ir viendo las modificaciones de otros usuarios sin necesidad de actualizar.
 - Tener un sistema de comunicación que permita a los usuarios conversar entre ellos de modo síncrono.
 - El editor debe facilitar la programación del usuario, informando a este sobre que línea se encuentra y permitiendo la búsqueda en el interior del código fuente.
 - El sistema debe informar de los mensajes de error para que el usuario pueda corregirlos, informando de la línea donde se encuentra.
- Utilizar la nomenclatura, tanto en código como en el modelo de datos, de la plataforma e-learning seleccionada.

²**Parser:** Analizador sintáctico que convierte el texto de entrada en otras estructuras.

1.4. Antecedentes

Como se ha comentado en los apartados anteriores, la edición de código fuente de modo colaborativo es un tema ampliamente abordado y que se encuentra en pleno auge en la actualidad. Este proyecto ha tomado la idea de numerosos sistemas utilizados para tareas similares. Dentro de éstos se pueden destacar los siguientes:

- **EclipseGavab[GG12]:** Entorno de desarrollo integrado con características colaborativas que permiten la implementación del Aprendizaje Basado en Proyectos en titulaciones online de Informática. Dispone, entre otras características, de mensajería instantánea y edición compartida de código, funcionalidades que permiten que el profesor supervise el trabajo de forma telemática y que los alumnos colaboran virtualmente. EclipseGavab soporta la programación en varios lenguajes de programación de distintos paradigmas y características, entre ellos Pascal, C y Java, de forma que pueda utilizarse en diferentes asignatura a lo largo de la titulación.
- **AmyEditor[amy12]:** Herramienta web para desarrolladores de edición colaborativa de código fuente. Tiene varias características útiles como la sangría inteligente, resaltado de sintaxis, plegado de código, atajos de teclado personalizables y muchos más. Se pueden organizar los documentos en proyectos y abrir varios ficheros a la vez utilizando pestañas. Este sistema actualmente soporta JavaScript, Ruby, PHP, C#, Java, HTML (HyperText Markup Language), Python etc.
- **Ethercodes[etc12]:** Editor de código fuente para codificar en la nube³ aún en etapa alfa. No compila ni interpreta código alguno. En este sentido, es una herramienta complementaria para el desarrollador.
- **Notapipe[npp12]:** Editor de texto online colaborativo en tiempo real para desarrollar sitios webs. Varios usuarios pueden editar archivos al mismo tiempo y ver en tiempo real los cambios en los archivos de edición. Basado en la Web, soporta Firefox, Internet Explorer, Safari, Google Chrome y Opera. Tiene resaltado de sintaxis en lenguajes de programación web: HTML, CSS, JS y PHP. En la versión gratuita sólo pueden trabajar tres usuarios simultáneamente, es en versiones de pago donde pueden trabajar desde 6 hasta 9 usuarios en *realtime*.
- **JSBin[jsb12]:** Servicio online que nos permite introducir código HTML, CSS y JavaScript y verificar su funcionamiento. Está pensado para que los desarrolladores

³**Computación en la nube:** Concepto conocido también bajo los términos servicios en la nube, informática en la nube, nube de cómputo o nube de conceptos, del inglés cloud computing, es un paradigma que permite ofrecer servicios de computación a través de Internet.

puedan trabajar de manera colaborativa pero, es abierto a cualquier usuario. Podemos editar sin dificultades, incluir alguna de las librerías más comunes e incluso, guardar la URL del proyecto y hacerla pública.

- **Mozilla Skywriter[skw12]:** Entorno de desarrollo en el que los datos se pueden acceder desde cualquier máquina. Esto permite a los desarrolladores colaborar en proyectos a través de una interfaz única accediendo a ella por un navegador web. Soporta HTML, CSS, PHP, Python, C#, C, Ruby y JavaScript. No compila ni interpreta código alguno.
- **Open Cooperative Web Framework[ocw12]:** Entorno de desarrollo que permite interacción real entre usuarios remotos y fuentes de datos externas. El framework⁴ se encarga de notificar a los usuarios y apoyar en la colaboración a los usuarios. No compila ni interpreta código alguno.

Estos ejemplos han sido el punto de partida para la realización de este proyecto, pues el desarrollo del Software se ha llevado a cabo de tal forma que intente imitar el funcionamiento de las mismas, aunque se han intentando realizar una serie de mejoras y adaptaciones, como es el caso de embeberlo dentro de la plataforma de e-learning Moodle.

1.5. Restricciones

En este apartado se expondrán todos aquellos factores que condicionarán la estrategia de diseño en la realización del proyecto. Estas limitaciones se pueden dividir principalmente en dos, la identificación de los factores dato y la identificación de los factores estratégicos. Las restricciones dato, que no podrán ser modificadas, comprenderán sobre todo a las limitaciones presupuestarias y de tiempo, el tipo de hardware existente, el sistema operativo, etc. En cuanto a las restricciones estratégicas, en las que habrá que elegir entre varias posibilidades para cada una, se encontrarán la elección del entorno de trabajo, las herramientas utilizadas, el lenguaje de programación, el tipo de interfaz de usuario, etc.

1.5.1. Factores dato

En cuanto a los factores dato, que son aquellos que no pueden ser modificados a lo largo del desarrollo del proyecto tenemos los siguientes:

⁴**Framework:** estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.

- Como factor inicial tenemos principalmente la realización de un editor, de tipo colaborativo, de código fuente en lenguaje C.
- Otro de los factores iniciales que se puede considerar es que el módulo que se desarrolle debe ser intuitivo y accesible, de manera que no acarree ningún tipo de equívoco.
- La utilización de la arquitectura de comunicación cliente-servidor para realizar la comunicación entre el módulo y el usuario final.
- Además, habrá que añadir todas aquellas indicaciones que se describan en el capítulo *Especificación de Requisitos*.

1.5.2. Factores estratégicos

Estos factores son variables, pues habrá que elegir entre varias posibilidades. De esta manera y como hemos enunciado también anteriormente existen una serie de factores estratégicos:

- **Elección de la plataforma de aprendizaje.** La elección de la plataforma de aprendizaje ha sido uno de los temas más difíciles de decidir. Si bien, a priori, se comenzó a estudiar la posibilidad de utilizar el sistema Degree para la realización del editor, más adelante se descartó pues se consideró que la opción de Moodle era mucho más ventajosa y provechosa, sobre todo porque tiene una interfaz de tecnología sencilla, ligera y compatible, porque es muy fácil de instalar, porque posee una sólida seguridad en todos sus módulos, porque las áreas de introducción de texto pueden ser editadas usando el sencillo editor de HTML y porque Moodle se ha convertido en el sistema de gestión de cursos más utilizado y conocido del mundo. Debido a que decidir sobre este tema no ha sido trivial, se ha profundizado mucho más en el siguiente capítulo.
- **Elección de un lenguaje de programación.** En cuanto a la elección del lenguaje de programación hay que tener en cuenta que se han desarrollado dos módulos totalmente independientes. Por un lado se ha desarrollado la interfaz que permite acceder al editor colaborativo. Un lenguaje estándar para la realización de esta interfaz y que es el más conocido y utilizado para este fin es **PHP**. Sin embargo, la elección de PHP está supeditada a la elección de plataforma de aprendizaje, hecho que se comentará más adelante. Por otro lado, se ha desarrollado el editor de código colaborativo. Para la implementación de este sistema se ha optado por el lenguaje

Javascript debido principalmente a su API⁵ JQuery, la cual ha servido para de enlace entre el interfaz y el sistema encargado de la edición de código. A su vez, para la programación del editor en si también nos hemos servido de Javascript.

- **Elección del gestor de la base de datos.** En cuanto a la elección del gestor de la base de datos nos hemos centrado, para tomar la decisión, en el lenguaje de desarrollo seleccionado, es decir, en PHP. PHP soporta diferentes gestores de base de datos como son MySQL, PostgreSQL, Oracle, ODCB, etc. Sin embargo, se ha decidido utilizar el gestor MySQL pues presenta una serie de ventajas como son:
 - Trabaja en múltiples plataformas.
 - Licencia de uso GPL.
 - Es un producto de excelente calidad y velocidad.
 - Permite almacenar los datos en distintas arquitecturas de almacenamiento.
 - Está muy extendido en aplicaciones de gestión de entorno web por su buena integración con el servidor Web Apache y el lenguaje de programación PHP.
- **Elección del servidor Web.** En la elección de servidor Web se han considerado tres posibilidades: ISS (Internet Information Server), Apache Web Server y Glassfish. Finalmente se ha decidido por Apache puesto que es el más extendido en la programación Web.
- **Sistema Operativo.** El Sistema Operativo que tienen instalados los equipos y bajo el cuál se desarrollará la aplicación será Mac OS X, aunque el sistema final podrá ejecutarse desde cualquier plataforma que soporte PHP y que permita disponer de una base de datos.
- **Entorno de trabajo.** Si bien se ha escogido la opción Mac OS por la comodidad que supone su manejo, se ha hecho uso de máquinas virtuales para hacer pruebas en Windows y Linux.

1.6. Recursos

A lo largo de este proyecto se han hecho uso de una serie de recursos que según su naturaleza pueden ser clasificados en recursos humanos, recursos hardware y recursos software.

⁵**API:** conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

1.6.1. Recursos humanos

Son recursos referidos al personal requerido para la creación, dirección y desarrollo del proyecto.

- Director del proyecto D. Cristóbal Romero Morales
- Autor D. Antonio Jesús González León

1.6.2. Recursos Hardware

Determinan los dispositivos hardware utilizados para la creación del sistema.

- Ordenador Macbook Pro 2,66 GHz Intel Core i7, 4 GB RAM, 300 Gb de disco duro.

1.6.3. Recursos Software

Determinan qué herramientas software van a ser necesarias para la creación y desarrollo del sistema. Entre los recursos software se encuentran cada uno de los programas, sistemas operativos, herramientas de programación y de utilidades que han sido utilizadas a lo largo del proyecto.

Los recursos software disponibles son:

- Sistema Operativo
 - Mac OS X Lion 10.7.4
 - Microsoft Windows XP
 - GNU/Linux Ubuntu 12.04 LTS
 - Oracle VM VirtualBox
- Entorno de Programación
 - Aptana Studio 3
 - MAMP Server 2.0
 - Apache/2.2.21 (Unix) mod_ssl/2.2.21 OpenSSL/0.9.8r DAV/2 PHP/5.3.6
 - MySQL 5.5.9
 - PHP Version 5.3.6
 - Mozilla Firefox

- Internet Explorer
- Google Chrome
- Safari
- Extensión Firebug para Mozilla Firefox
- Moodle v 1.9
- Moodle v 2.1
- Documentación
 - Mactext
 - Latexian
 - L^AT_EX
 - Bibtex
 - DIA
 - GNU Image Manipulation Program (GIMP)
 - Adobe Photoshop CS5
 - Adobe Acrobat 9.0

Capítulo 2

Análisis de plataformas e-learning

En este capítulo se va a realizar un análisis de las plataformas CMS (Course Management Systems) más utilizadas en la actualidad para seleccionar aquella que mejor se adapte a las necesidades de nuestro proyecto, indicando además los motivos de dicha selección. El análisis de las plataformas se realizará teniendo en cuenta las siguientes características:

- **Requerimientos:** se tendrán en cuenta los requerimientos software y hardware, así como el tipo de licencia de la plataforma.
- **Seguridad:** debido a que se trata de una aplicación orientada a Internet es importante determinar la seguridad que ofrece.
- **Soporte:** en este apartado se analizará el tipo de soporte ofrecido por la plataforma, como pueden ser manuales, foros, etc.
- **Facilidad de uso:** al ser una plataforma orientada al aprendizaje, es decir, que va a ser constantemente accedida por profesores y alumnos que en la mayoría de los casos no serán expertos en e-learning, la facilidad de uso de la misma es un punto de vital importancia.
- **Funciones:** en este apartado se verán las funciones puramente informáticas de la plataforma, como pueden ser balanceo de carga, exportación del contenido de la base de datos, etc.
- **Administración:** la administración es uno de los puntos más importantes dentro de una plataforma. En este apartado se verán las capacidades de administración que tiene cada una.

- **Interoperatividad:** esta característica favorece el aprendizaje al fomentar la comunicación entre alumno-profesor y alumno-alumno. Por esta razón, se analizará y se valorará como punto favorable.
- **Aplicaciones soportadas:** además de la propia funcionalidad ofrecida por la plataforma, también se tendrá en cuenta la posibilidad de añadir aplicaciones extras dentro del entorno de la misma.

2.1. Moodle

2.1.1. Introducción

Moodle[mdl12] es una plataforma de aprendizaje a distancia basada en software libre, es decir, una aplicación diseñada para ayudar a los educadores a crear cursos de calidad en línea. Moodle fue creado por Martin Dougiamas, antiguo administrador de WebCT en la Universidad Curtin. Se basó en trabajos sobre el constructivismo en pedagogía, que afirman que el conocimiento se construye en la mente del estudiante en lugar de ser transmitido sin cambios a partir de libros o enseñanzas. Moodle ha ido evolucionando desde su creación en 1999 y actualmente se siguen desarrollando nuevas versiones cada vez con más frecuencia. En enero de 2005, la base de usuarios registrados incluye 2600 sitios en más de 100 países y está traducido a más de 50 idiomas.

2.1.2. Requerimientos

- Servidor de aplicación: PHP 4.1.2+.
- Coste aproximado: Gratis.
- Base de datos: MySQL.
- Licencia: GNU GPL.
- Sistema operativo: Cualquiera.
- Lenguaje de programación: PHP.
- Servidor Web: Apache 1.3.

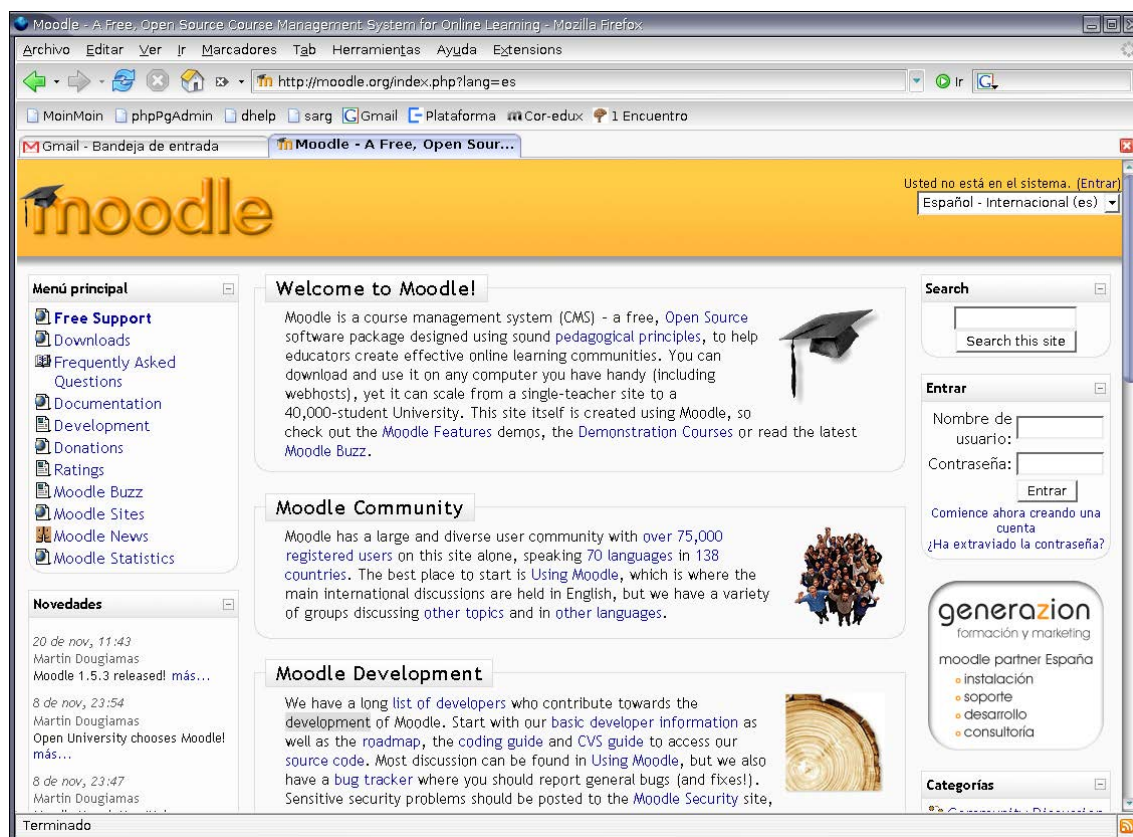


Figura 2.1: Plataforma e-learning Moodle

2.1.3. Seguridad

La seguridad en el sistema Moodle se limita a la autenticación de los usuarios mediante un servidor externo LDAP. También permite la autenticación de las cuentas creadas mediante servidores de correo o de news. Por otra parte, se permite la creación de distintos perfiles para mejorar el control de accesos a los recursos.

2.1.4. Soporte

En cuanto se refiere al soporte, es decir, lo que la organización que realiza el programa nos proporciona, tenemos que Moodle posee manuales comerciales en el mercado además de que proporciona una ayuda on-line, un foro público, una lista de correo y manuales para su utilización. Desde la web de Moodle se puede acceder a gran cantidad de documentación sobre la plataforma, así como, tener comunicación directa con desarrolladores experimentados para preguntar cualquier duda.

2.1.5. Facilidad de uso

Moodle tiene una gran facilidad de uso, permitiendo la inclusión de múltiples tipos de ficheros en los cursos, como pueden ser: test, libros digitales, ficheros HTML, ficheros PDF, hojas de cálculo EXCEL, ficheros WORD, etc. La asignación de profesores y estudiantes a los cursos, así como la creación y configuración de los mismos, no tiene ninguna complicación puesto que se realiza desde un simple menú. Posteriormente, la adición de recursos a los cursos se realiza de manera intuitiva sin ningún tipo de problema. Moodle tiene la posibilidad de cambiar su aspecto fácilmente gracias a las hojas de estilos (css).

2.1.6. Funciones

Moodle posee un sistema avanzado para cachear rápidamente contenidos de las páginas. Las plantillas para agilizar la navegación, además de permitir los duplicados de la base de datos para garantizar la seguridad. En su contra hay que destacar que permite balanceo de carga si hubiese varios servidores.

2.1.7. Administración

La administración de esta plataforma se puede dividir en administración de la plataforma, administración de usuarios y administración de cursos. La administración de la plataforma la realiza un administrador definido en la instalación de la misma. En este apartado se puede gestionar la interfaz del curso mediante la gestión de los temas y la gestión de los módulos incluidos en la plataforma. La administración de usuarios se basa en dos grandes grupos: estudiantes y profesores. Tanto unos como otros tienen un proceso de autenticación. El administrador es el encargado de asignar profesores a los cursos. Después de esto los profesores tienen control total del mismo. La administración de cursos se realiza por parte de los profesores y consiste en la creación de recursos del mismo de manera que los alumnos matriculados tengan acceso a los mismos.

2.1.8. Interoperatividad

Como herramientas de interoperatividad la plataforma Moodle posee las siguientes fundamentalmente:

- Foros de discusión. Estos foros poseen las funciones normales de los mismos, edición de mensajes, adjuntar ficheros, etc.
- Chats. Permiten una interacción fluida mediante texto síncrono.

- Wikis. Permiten el desarrollo de wikis dentro de un curso.
- Talleres. Permite la subida de documentos así como la evaluación de los mismos entre iguales y la calificación por parte del profesor.
- Glosarios. Permite la realización de un listado de términos acerca de una temática concreta así como la evaluación de los mismos y la realización de comentarios por parte de otros alumnos y del profesor.

2.1.9. Aplicaciones soportadas

Una de las ventajas de esta plataforma es la posibilidad de incluir nuevos módulos. Esta característica combinada con la amplia expansión de la misma en todo el mundo dota a la plataforma de una gran capacidad de crecimiento mediante la inclusión de nuevos módulos desarrollados o mediante el desarrollo de un nuevo módulo en PHP que satisfaga las necesidades de los usuarios.

2.2. WebCT

2.2.1. Introducción

WebCT [wct12] comenzó como un proyecto del profesor Murray Golbert de la Universidad de Columbia para estudiar los efectos del aprendizaje online. WebCT se creó en 1997 como un producto comercial y es en 1999 cuando fue comprada por la Universal Learning Technology (UTL). Dentro de la plataforma WebCT se encuentran distintos productos de e-learning como pueden ser WebCT Vista y WebCT Campus Edition. A continuación se analizarán las características de WebCT Vista que se encuentra en el grupo de CMS más populares del momento.

2.2.2. Requerimientos

- Servidor de aplicaciones: La administración se realiza desde web. Los instaladores locales realizan la instalación de aplicaciones.
- Coste aproximado: La licencia de hasta 1000 usuarios para una organización sin ánimo de lucro es de aproximadamente 5000 €.
- Base de datos: Oracle 9i. Licencia: Anual por usuarios.

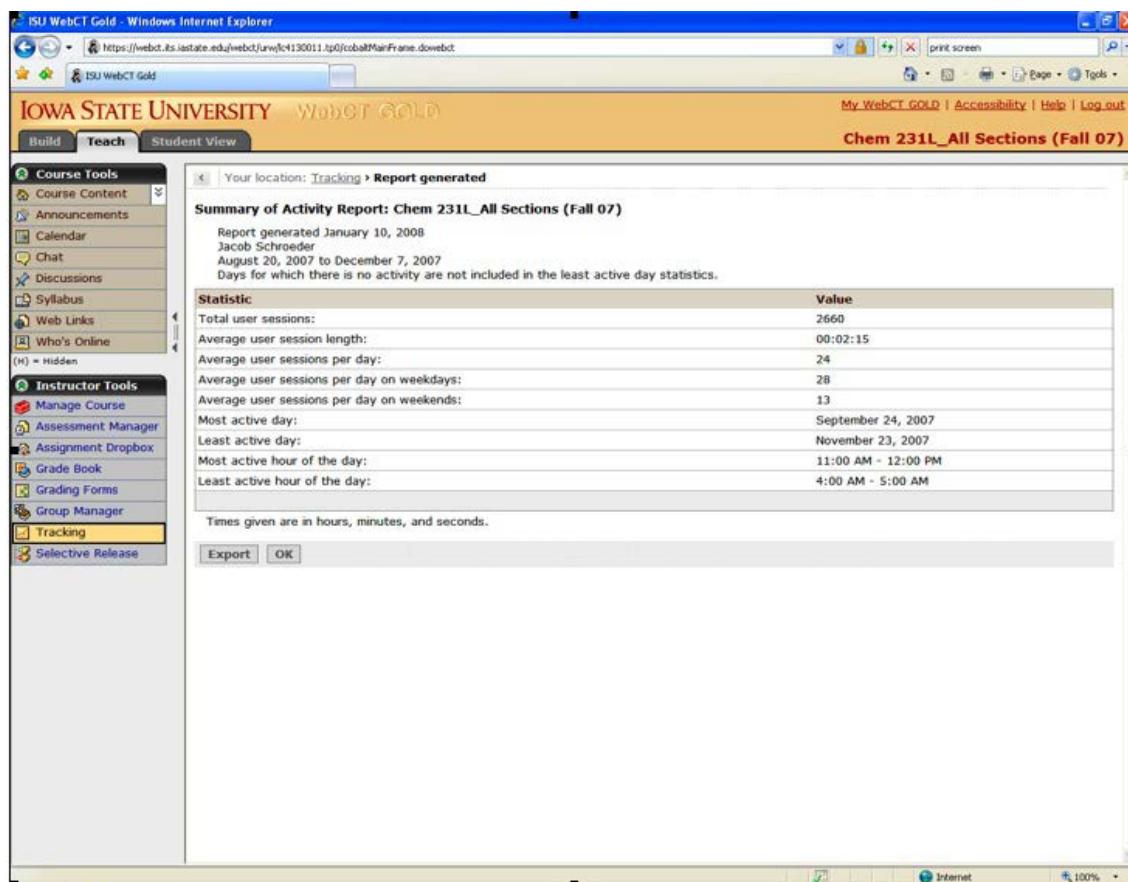


Figura 2.2: Plataforma e-learning WebCT

- Sistema operativo: RedHat Enterprise Linux, Sun SPARC Solaris 8 y 9, Windows 2000 Server SP 2 o Windows 2000 Advanced Server.
- Lenguaje de programación: Java y JavaScript. Servidor web: Apache.

2.2.3. Seguridad

Los administradores del sistema pueden proteger el acceso a los cursos mediante nombre de usuario y clave. Esta autenticación se realiza mediante un servidor LDAP externo o mediante el uso del protocolo Kerberos.

Se pueden establecer los parámetros de las claves, como longitud y cambios de la misma pasado un periodo de tiempo. Los logins de los usuarios pueden ser encriptados mediante el protocolo SSL.

Además de lo anteriormente comentado, el sistema WebCT esta completamente auditado, recogiendo registros de auditoría de todos los usuarios, así como guardando historial de accesos de los mismos.

2.2.4. Soporte

El soporte es uno de los puntos fuertes de WebCT debido a que se trata de una plataforma de pago. En este punto encontramos soporte para administradores, profesores y alumnos.

En cuanto al soporte para administradores se encuentran manuales on-line sobre administración, 'faqs' sobre administración y boletines de noticias, etc. Además de esto, encontramos soporte interactivo mediante listas de correo y foros de discusión. Por último se ofrece un área de descarga de software para administradores y un punto de contacto con el sistema de soporte de WebCT para solicitar ayuda en caso de que todo lo anterior no fuera suficiente.

El soporte ofrecido para profesores y alumnos del sistema es similar al ofrecido para los administradores, teniendo contenidos on-line, soporte interactivo y descarga de software.

2.2.5. Facilidad de uso

La creación de cursos esta basada en un editor web de uso fácil e intuitivo. La estructura básica y la navegación en los cursos no puede ser modificada, la ventaja que tiene este aspecto es que es de fácil aprendizaje para los novatos y el entorno se vuelve familiar para los alumnos avanzados. La desventaja que tiene es que se produce un proceso monótono al no poder adaptar la navegación a las necesidades del curso. El contenido de los cursos puede ser editado mediante un applet de Java que funciona como editor HTML. éste se utiliza para pequeñas modificaciones. Mientras que para modificaciones de mayor envergadura se usa el editor online instalado en el ordenador del administrador o profesor. En cuanto a la navegación de los cursos, decir que no se puede acceder a los contenidos mediante direcciones URL, posee un buen escalado de fuentes e imágenes y que los botones de retroceso y recarga en el navegador no funcionan como deberían. Al recargar una página se vuelve a la página inicial.

2.2.6. Funciones

En cuanto a las características que posee WebCT tenemos las siguientes: posee cacheado de páginas (lo cual mejora la navegación por los cursos), replicado de las bases de datos,

trabajo bajo Oracle permitiendo así realizar copias de seguridad de manera rápida y fiable y, por último, balanceo de carga (que permite repartir el trabajo para evitar la saturación del sistema). Por el contrario, tiene la desventaja de que no existe la posibilidad de exportar el contenido del curso de manera estática para usarlo en un servidor local.

2.2.7. Administración

Las posibilidades de administración que posee WebCT permiten realizar la administración online del sitio web. Una característica especial es la actualización de los contenidos de manera temporal, es decir, se permite agregar o eliminar contenidos según períodos de tiempo. Otro punto importante en cuanto a la administración es la posibilidad de creación de perfiles para usuarios y la gestión de cada conjunto independientemente. En cuanto al registro de estudiantes para los cursos, se puede realizar directamente por medio del administrador o del profesor responsable del curso o bien, crear un proceso de registro automático para que los alumnos se matriculen en el propio curso.

2.2.8. Interoperatividad

Para fomentar el trabajo de los alumnos la plataforma WebCT posee diferentes herramientas que favorecen la interoperatividad, como son los foros de discusión, en los cuales los mensajes pueden contener ficheros adjuntos, ficheros HTML, etc. Otras características son el intercambio de ficheros mediante WebDAV o las listas de correo de los cursos, mediante las cuales se puede divulgar información. Por último posee un chat en tiempo real basado en Java y una pizarra compartida en la que se pueden compartir información con los alumnos. Esta pizarra permite escribir símbolos matemáticos así como subir imágenes y anotaciones.

2.2.9. Aplicaciones soportadas

WebCT posee gran cantidad de aplicaciones que dan soporte a los cursos generados como pueden ser Chats, foros, calendarios, estadísticas, etc. Un punto importante es que permite el trabajo en grupo (GroupWare), de manera que pueden generar grupos de manera aleatoria, generados por los profesores o bien grupos generados por los propios alumnos. Otra característica que posee WebCT es la posibilidad de que cada alumno se cree su propio portfolio en formato HTML, así como un espacio común para los alumnos, de manera que puedan intercambiar información y archivos.

2.3. Ilias

2.3.1. Introducción

La plataforma ILIAS [ili12] fue desarrollada como parte del proyecto VIRTUS de la Universidad de Colonia. Es un sistema abierto usado por multitud de universidades debido a su licencia GNU. Este sistema está profundamente extendido y en los últimos tiempos está tomando importancia en multitud de universidades dentro de España. En Andalucía es el sistema que promueve la Junta de Andalucía como medio de enseñanza virtual. Como ejemplo está la Universidad de Jaén.



Figura 2.3: Plataforma e-learning Ilias

2.3.2. Requerimientos

- Servidor de aplicaciones: PHP 4.3.2.
- Coste aproximado: Gratis.
- Base de datos: MySQL 4.0.14 o superior. Licencia: GNU GPL.
- Sistema operativo: Desarrollado con RedHat Linux 7.x/8.0 pero se puede usar sobre cualquier distribución Linux, Solaris o Unix.
- Lenguaje de programación: PHP.
- Servidor web: Apache.

2.3.3. Seguridad

La seguridad en el sistema ILIAS se limita a la autenticación de los usuarios mediante un servidor externo LDAP o mediante el uso del protocolo RADIUS. Por otra parte, se permite la creación de distintos perfiles para mejorar el control de accesos a los recursos. Otro punto importante introducido en la última versión es el control de virus en la subida de ficheros.

2.3.4. Soporte

El soporte del sistema se limita a la información que contiene la web de la plataforma, <http://www.ilias.uni-koeln.de/ios/index-e.html>, en el apartado docs, que contiene documentos sobre instalación, desarrollo y migración entre distintas versiones de ILIAS. Además de estos documentos, en la misma web se encuentra un foro de discusión donde se pueden realizar consultas sobre la plataforma.

2.3.5. Facilidad de uso

ILIAS tiene una excelente facilidad de uso, permitiendo la inclusión de múltiples tipos de ficheros en los cursos, como pueden ser, módulos de aprendizaje ILIAS, test ILIAS, libros digitales, ficheros HTML, ficheros PDF, hojas de cálculo EXCEL, ficheros WORD, etc. Para la edición de los contenidos se integra un editor basado en XML con una interfaz XHTML con funciones de edición para texto, tablas, mapas, glosario, etc. Permite la integración de imágenes, applets, animaciones, video y audio. Por último decir que también permite el uso de hojas de estilos (CSS) y la reusabilidad de objetos mediante ILIAS VRI (Virtual Ressource Identifiers).

2.3.6. Funciones

La plataforma ILIAS posee distintas funciones de cacheado de páginas y la posibilidad de exportar información de manera estática para uso local. En su contra decir que no posee la función de replicado de bases de datos ni balanceo de carga.

2.3.7. Administración

La administración de la plataforma se limita a permitir o no permitir accesos a los contenidos por parte de los usuarios. Para ello jerarquiza los usuarios en función de perfiles

o roles con diferentes niveles de acceso. Los perfiles de esta plataforma son los siguientes: profesor, estudiante, diseñador e invitado.

2.3.8. Interoperatividad

Como herramientas de interoperatividad la plataforma ILIAS posee foros de discusión que pueden ser introducidos en los cursos generados. Estos foros poseen las funciones normales de los mismos, editar mensajes, adjuntar ficheros, etc. Además de estos foros de discusión, ILIAS posee listas de correo tanto internas como externas que permiten la comunicación entre los usuarios de la plataforma. Por último, decir que posee un Chat para la comunicación en tiempo real de los usuarios.

2.3.9. Aplicaciones soportadas

ILIAS posee gran cantidad de aplicaciones que dan soporte a los cursos generados, como pueden ser Chats, foros, calendarios, estadísticas, etc. La plataforma ILIAS, además, posee gran cantidad de herramientas para fomentar el trabajo en grupo, permitiendo el intercambio de ficheros, la creación de contenidos de manera cooperativa, creación de foros para grupos, creación de calendarios de grupo y un organizador de tareas. En cuanto al tratamiento de los usuarios, ILIAS realiza su administración en función de cursos o bien en función de grupos (GroupWare).

2.4. Proyecto Sakai

2.4.1. Introducción

Plataforma desarrollada en conjunto por la Universidad de Mighigan, Universidad de Indiana, el MIT, Stanford y el consorcio uPortal con el objetivo de integrar y sincronizar su software educativo en una colección pre-integrada de herramientas de Open source para e-learning.

2.4.2. Requerimientos

- Servidor de aplicaciones: J2EE.
- Coste aproximado: Gratuito o mínimo.
- Base de datos: MySQL u Oracle.

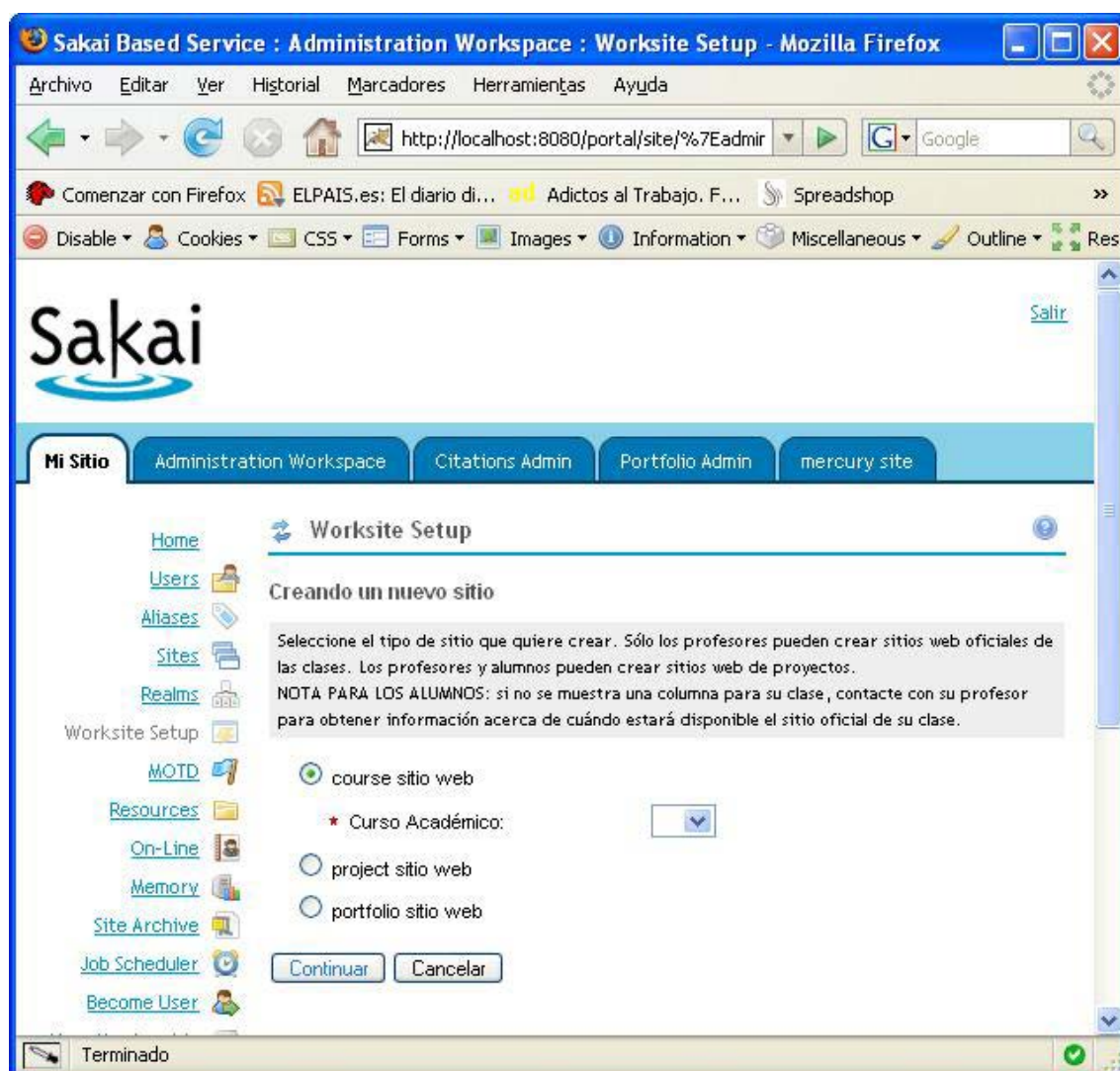


Figura 2.4: Plataforma e-learning Sakai

- Licencia: Open Source.
- Sistema operativo: Windows XP , Macintosh Mac Os, Linux o Solaris.
- Lenguaje de programación: Java.
- Servidor web: TomCat.

2.4.3. Seguridad

En cuanto a la seguridad ofrecida por SAKAI[sak12], no se ofrecen datos técnicos en cuanto a la tecnología usada, simplemente se menciona que la seguridad es gestionada

mediante Java y las tecnologías que contiene para esta función, permitiendo administración de características como autenticación. Un punto a favor de esta plataforma es que, al ser de carácter libre, se dispone del código fuente, por lo tanto en cualquier momento se pueden modificar los protocolos de seguridad y solventar problemas de seguridad del software.

2.4.4. Soporte

El soporte del sistema tiene dos apartados. Uno gratuito que se limita a la información que contiene la web de la plataforma <http://www.sakaiproject.org> que contiene documentos sobre instalación, desarrollo y migración entre distintas versiones de SAKAI. Por otro lado, tiene soporte comercial ofrecido por diferentes compañías que ofrecen hosting, instalación, migración, etc. Este soporte se ofrece con la versión comercial de la plataforma, la cual tiene un coste económico.

2.4.5. Facilidad de uso

En cuanto al uso de la plataforma SAKAI, decir que se permite la agregación de todo tipo de documentos de manera sencilla, como por ejemplo: ficheros HTML, ficheros PDF, hojas de cálculo EXCEL, ficheros WORD, etc. La interfaz de uso es una GUI (Graphical User Interface) basa en XML, mediante la cual se pueden gestionar todas las características del curso que se está creando.

2.4.6. Funciones

SAKAI posee como característica más importante la escalabilidad, puesto que posibilita el trabajo tanto con bases de datos de pequeño tamaño como de gran tamaño. Otras características importantes son que permite cacheado de la información y balanceo de carga, permitiendo así un mejor uso del sistema, disminuyendo la posibilidad de sobrecargas en el mismo.

2.4.7. Administración

La administración de los cursos en la plataforma SAKAI permite la creación de sitios individuales para los cursos realizados, la gestión de usuarios en cada uno de ellos así como la creación de subgrupos dentro de cada uno de los cursos. En cuanto a los contenidos de los cursos se permite la administración tanto de su contenido como del acceso al mismo.

2.4.8. Interoperatividad

Las herramientas para la interoperatividad ofrecidas por esta plataforma son foros de discusión, Chats en tiempo real, listas de correo, anuncios, calendarios, etc. De esta manera, se da soporte a la comunicación entre los usuarios de los cursos. Además de lo anteriormente mencionado, se permite la inclusión de información previa por parte del administrador del sistema o, incluso, por parte de cualquier usuario del sistema. La plataforma SAKAI ofrece soporte para el trabajo en grupo (Groupware), dando también soporte para el desarrollo e investigación online.

2.4.9. Aplicaciones soportadas

En cuanto a las aplicaciones soportadas por SAKAI decir que no tiene limite, puesto que funciona mediante servicios web. Por lo tanto cualquier aplicación puede ser adaptada e introducida dentro del portal como servicio web. Por otro lado, tenemos las APIs de programación de la plataforma mediante las cuales se pueden desarrollar nuevas aplicaciones para los cursos.

2.5. Atutor

2.5.1. Introducción

ATutor[atu12] es un Sistema de Gestión de Contenido Educativo Basado en la Web, diseñado en la accesibilidad y la adaptabilidad. De esta manera, permite a los educadores desarrollar fácilmente contenido en línea en un ambiente de aprendizaje estructurado y adaptable. En cuanto a los estudiantes, permite navegar a través del contenido de diferentes maneras, de acuerdo a su estilo o método preferido de aprendizaje en línea. ATutor está disponible de manera gratuita para uso no comercial bajo GNU GLP, permitiendo la distribución y modificación del código fuente provisto siempre y cuando el derecho de autor y las líneas de crédito no sean alteradas.

2.5.2. Requerimientos

- Servidor de aplicación: PHP 4.1.2+.
- Coste aproximado: Gratis.
- Base de datos: MySQL.

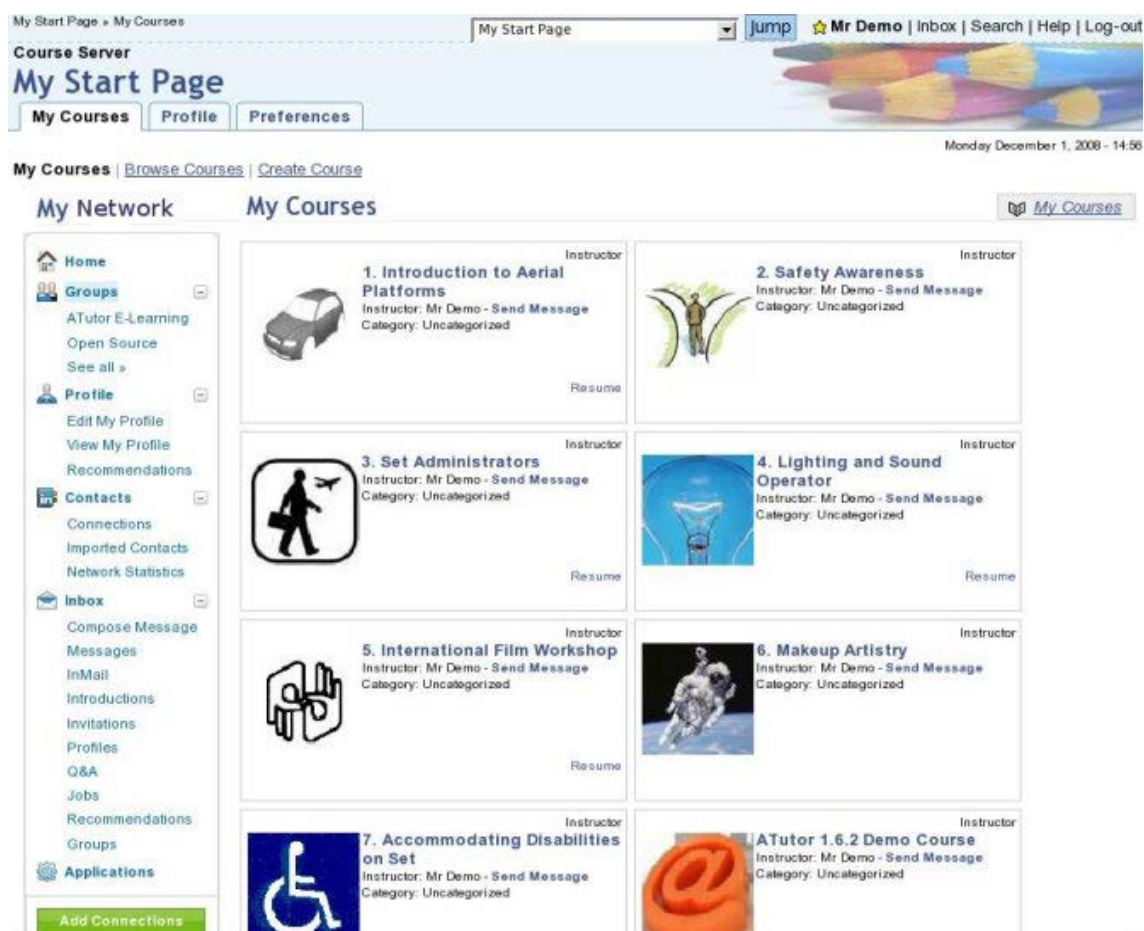


Figura 2.5: Plataforma e-learning ATutor

- Licencia: GNU GPL.
- Sistema operativo: Cualquiera.
- Lenguaje de programación: PHP.
- Servidor Web: Apache 1.3.

2.5.3. Soporte

En cuanto a lo que se refiere al soporte, Atutor no tiene ningún manual comercial en el mercado aunque sí proporciona una ayuda on-line, un foro público, una lista de correo y manuales para su utilización.

2.5.4. Facilidad de uso

Es un LMS (Learning Management System) con gran soporte para realizar cursos y caracterizado por poseer funciones de colaboración. Además, está diseñado para una fácil accesibilidad, para su uso en varias lenguas y para el cumplimiento de especificaciones e-learning. Posee una buena ayuda on-line y un tutorial para nuevos usuarios. Además se sustenta en los mejores recursos open source, Apache, PHP y MySQL.

2.5.5. Funciones

Esta plataforma posee varias funciones siempre orientadas a mejorar la accesibilidad y facilitar el uso. En cuanto a aspectos relacionados con este apartado encontramos que posee navegación adaptativa, permitiendo a cada usuario configurar los contenidos de sus cursos. Otra función importante es la posibilidad de exportación de los contenidos de la base de datos.

2.5.6. Administración

El apartado de administración del aTutor posee menos características con respecto a los CMS vistos, esto es debido a que .aTutor.es una plataforma orientada al aprendizaje (LMS). Aunque también hay que decir que la funcionalidad incluida en el apartado de administración es la idónea para el correcto funcionamiento de los cursos desarrollados.

2.5.7. Interoperatividad

En este apartado aTutor supera a las plataformas anteriores permitiendo la comunicación mediante las aplicaciones vistas en los anteriores CMS, como son Chats, foros, etc. Pero además posee un módulo de trabajo colaborativo que está integrado dentro de la plataforma (aCollab).

2.5.8. Aplicaciones soportadas

aTutor posee una serie de aplicaciones, cada una de ellas con una funcionalidad clara definida:

- aCollab: aplicación que permite el trabajo colaborativo así como la creación de grupos dentro de los cursos.

- aChat: esta aplicación como se puede observar posee un nombre bastante identificativo de su función, es decir, permite crear Chats dentro de los cursos.
- aComm: esta aplicación permite el uso de una pizarra, además de permitir la comunicación entre usuarios.
- aTalker: aplicación que realiza las funciones de "specher", es decir, lee un texto escrito dentro del curso.

2.6. eGroupWare

2.6.1. Introducción

eGroupware[egw12] es una sistema de trabajo en grupo vía web, de código abierto. Está escrita en PHP utilizando bases de datos, tales como PostgreSQL o MySQL. Incluye un calendario, una libreta de direcciones, un gestor de contactos, un cliente de correo electrónico IMAP, un InfoLog, funciones de CRM (Customer Relationship Management), un gestor de proyectos, un gestor de recursos, un gestor de ficheros, una plantilla de tiempos, un wiki, una base de conocimiento y un motor de flujos de trabajo.

2.6.2. Requerimientos

- Servidor de aplicación: PHP 4.3+ como mínimo, aunque se recomienda PHP 5.1+ pues se puede requerir alguna funcionalidad de esta versión como son las notificaciones y las utilidades de exportar e importar.
- Coste aproximado: Gratis.
- Base de datos: MySQL 4.1+, aunque se recomienda la versión 5.0+ (la mayor parte del trabajo funciona correctamente en la versión 4.0 pero módulos como la lista de precios en el gestor de proyectos así como la base de conocimiento no. Otra opción es utilizar Postgres 8.0+.
- Licencia: GNU GPL.
- Sistema operativo: Cualquiera.
- Lenguaje de programación: PHP.
- Servidor Web: Cualquiera que soporte PHP. Entre los más importantes se encuentran:

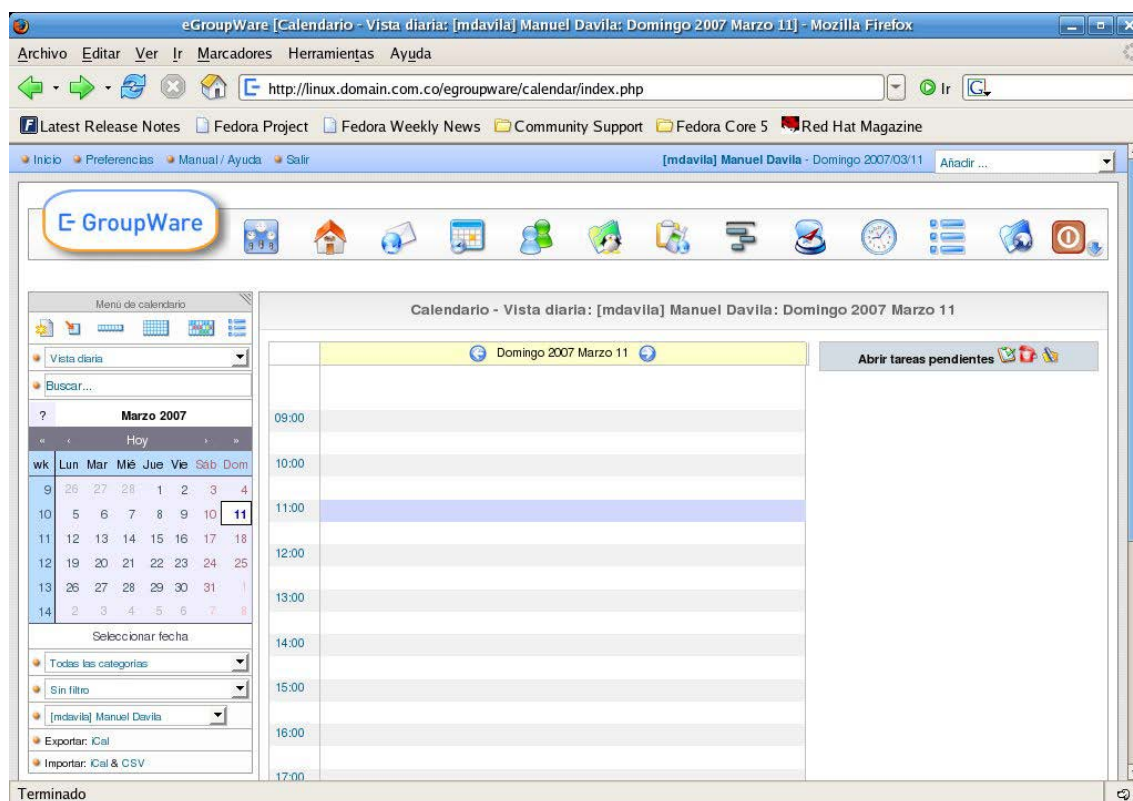


Figura 2.6: Plataforma e-learning EGroupWare

- Apache version 1.33+
- IIS
- Roxen

2.6.3. Seguridad

La seguridad en el sistema eGroupWare está bastante definida y ofrece una serie de servicios, de entre los que destaca:

- Seguridad básica en el servidor. Se pueden elegir muchas posibilidades para tener seguridad en el servidor:
 - La medida de seguridad más importante que se puede hacer es mantener la instalación actualizada. Para estar informado de las nuevas actualizaciones existe una lista de correo a la que cualquier usuario del sistema puede suscribirse.
 - Comprobar los servicios que se ejecutan y puertos abiertos.
 - Desinstalar programas que no se necesitan en el servidor.

- Comprobación local de existencia de un rootkit.
 - Administración de un servidor seguro, realizando conexiones en modo seguro, trabajando con pares de claves SSH, etc.
 - Instalar software para monitorizar los registros del sistema.
 - Entorno de detección de intrusos.
 - Seguridad de los servicios en demonios.
 - Cortafuegos.
- Seguridad en aplicaciones web. Con un software de seguridad en aplicaciones, se pueden ejecutar de forma segura las aplicaciones basadas en web, como eGroupWare, desde ejecuciones SQL, Cross Side Scripting y otros ataques. Hay varias aplicaciones en el mercado Apache e IIS.
 - Optimización y seguridad en Apache. Para la seguridad del servidor web, se deben desactivar todos los módulos innecesarios así como activar sólo los que se necesite para ejecutar el servidor web. Ejecutarlo con menos módulos, también mejora el rendimiento.
 - Seguridad de la instalación PHP.
 - Crear un certificado para el servidor web. Para proteger su privacidad, si se conecta a una instalación eGroupWare, se puede usar un certificado en el servidor. Con un certificado, se puede conectar al servidor web con una conexión cifrada (https en vez de http). Sin una conexión https, otras personas pueden usar un Sniffer para averiguar la contraseña, u otra información.

2.6.4. Soporte

En cuanto al soporte, eGroupWare posee un sitio web oficial muy completo en el que existe una amplia variedad de manuales así como un foro público, una lista de correo, ayuda on-line, etc.

Además, desde la misma web de eGroupWare se puede también tener comunicación directa con desarrolladores experimentados, concertar citas o entrevistas, etc.

2.6.5. Facilidad de uso

El sistema eGroupWare posee una interfaz muy intuitiva y fácil de utilizar. De esta manera, el menú principal proporciona rápidos enlaces hacia la página personal del usuario,

hacia el módulo general de preferencias personales, hacia información relativa a un módulo específico así como para funciones de cierre de sesión.

2.6.6. Funciones

Como se ha comentado anteriormente, el sistema eGroupWare ofrece una funcionalidad muy extensa y variada. Entre estas funciones destacan:

- Calendario.
- Libreta de direcciones.
- Gestor de contactos.
- Cliente de correo electrónico IMAP. InfoLog.
- Funciones de CRM.
- Gestor de proyectos.
- Gestor de recursos.
- Gestor de ficheros.
- Plantilla de tiempos.
- Una wiki.
- Una base de Conocimiento.
- Un motor de flujos de trabajo.

2.6.7. Administración

En lo que se refiere a administración, en eGroupWare se permitirá administrar tanto las cuentas de usuario como los grupos de usuarios.

Hay que tener en cuenta que la administración de grupos de usuarios es más compleja que la de cuentas de usuario, pues estas últimas únicamente necesitan su creación, su posible modificación y su cancelación. Sin embargo, la administración de grupos requiere otras operaciones.

Separar los usuarios en grupos proporciona una manera de controlar qué módulos están disponibles para cada grupo (Correo, calendario, administrador de ficheros). También proporciona al administrador una forma de controlar qué grupos tienen acceso a otros grupos

o información de usuarios. Al igual que una empresa tiene distintos departamentos, también los tiene eGroupWare. Por ejemplo, el departamento de compras puede no necesitar el módulo del Administrador de ficheros, y el departamento legal sí lo necesita para compartir documentos entre miembros o grupos del departamento. Al configurar el grupo del departamento legal, se marcaría el módulo del administrador de ficheros. Al configurar el grupo de compras, se asegura no marcar el módulo del administrador de ficheros.

eGroupWare permite poder tener todos los módulos disponibles para todos los grupos, pero el administrador controlaría qué grupo tiene acceso a la información de otro grupo mediante ACLs (Listas de control de acceso).

Es necesario realizar una preparación y una planificación para implantar eGroupWare en una empresa. Como todos los proyectos, en esta parte es donde se dedica la mayor parte del tiempo. Organizar los grupos y planear los privilegios de acceso. Preguntar quién necesita qué y por qué es un comienzo.

2.6.8. Interoperatividad

Como herramientas de interoperatividad, la plataforma eGroupWare posee foros de discusión y wikis. Estos foros poseen las funciones normales de los mismos, edición de mensajes, adjuntar ficheros, etc. En cuanto a los wikis ofrecen una herramienta para la construcción de conocimiento para la comunidad. Cada página en un Wiki es parecida a una página web excepto en que todo el mundo puede editarla. Esto hace a los Wikis una herramienta ideal para capturar conocimiento.

Ademas de todo esto posee funciones de CRM (Customer Relationship Management), un gestor de proyectos (a partir de diagramas de Gantt), un gestor de recursos, etc.

2.6.9. Aplicaciones soportadas

El sistema eGroupWare está formado por un conjunto de aplicaciones individuales e instalables. Todas estas aplicaciones tienen como objetivo dar solución a un problema relativo al trabajo colaborativo. Las aplicaciones más importantes dentro de este sistema pueden ser:

- Calendario.
- Libro de direcciones. Es una aplicación para la gestión de clientes u otra relativa a la información de contactos, y enlazarla con otras aplicaciones egroupware. Se

pueden importar y exportar contactos en formatos como LDIF, CVS o VCard y administrarlos por SQL o LDAP.

- FelaMiMail. Es una aplicación e-mail para el trabajo colaborativo que se puede enlazar con otras aplicaciones como es el caso del libro de direcciones.
- InfoLog

2.7. Comparativa de la plataforma para la gestión docente

Lo primero que debe tenerse en cuenta al tomar dicha decisión es que la plataforma debe ser gratuita y que tenga los requisitos software para favorecer la minimización de costes en el desarrollo del proyecto.

De esta manera y fijándonos en el coste de la licencia, se va a desechar el uso de WebCT, puesto que se trata de una plataforma de pago.

En este punto nos encontramos ante la decisión de usar una de las siguientes plataformas: ATutor, Moodle, Ilias y eGroupWare. La primera que descartaríamos, de entre estas cuatro, es eGroupWare pues aunque permite trabajo colaborativo no está destinado al aprendizaje. Por tanto nos quedarían sólo las tres primeras como opciones candidatas. Para la elección de una de ellas, realizaremos una comparativa detallada de características y funcionalidades de las tres, analizando los siguientes aspectos:

- Acceso.
- Seguimiento.
- Soporte del software.
- Funciones a disposición del alumno. Administración.
- Facilidad de uso.
- Otros.

2.7.1. Acceso

	ATutor	Ilias	Moodle
Autenticación	Pide el usuario y clave, además de tener la opción de login automático.	Pide usuario y clave utilizando el protocolo LDAP. Además el administrador puede crear grupos con distintos privilegios.	Pide usuario y clave utilizando el protocolo LDAP. Además permite el registro mediante la confirmación de clave por e-mail.
Invitados	Permite el acceso a todos los contenidos de los cursos públicos.	Solo permite el acceso a la sección de registro on-line.	Permite el acceso a todos los contenidos de los cursos públicos.
Acceso al curso	Se pueden crear distintos tipos de cursos según sus privilegios de acceso: público (puede entrar cualquier usuario), protegido (sólo se permite el acceso de los alumnos registrados en el curso) y privado (solo se permite el acceso de aquellos alumnos registrados y autorizados por el profesor del curso).	Sólo se permite el acceso de los usuario registrados.	Se pueden crear cursos públicos y privados. Los públicos permiten el acceso de usuarios registrados y de invitados. Los privados sólo pueden ser vistos por los usuarios registrados, es decir, alumnos confirmados por los profesores.

Cuadro 2.1: Comparativa de Acceso

2.7.2. Seguimiento

	ATutor	Ilias	Moodle
Seguimiento en el acceso	El seguimiento en un curso debe estar autorizado por el administrador mediante una opción de configuración. A partir de este momento el seguimiento del usuario que ha entrado en el curso es completo, pudiendo ver todas las actividades que realiza, etc. Además, tiene la opción de ver un histograma y exportar a un fichero Excel.	El administrador decide en cada curso cuáles son los parámetros del seguimiento: acceso, foros, etc. El profesor puede ver las estadísticas de su propio curso.	Tanto los profesores como el administrador pueden ver el seguimiento del alumno. Los profesores tienen acceso sólo a los cursos de los que son profesores. El seguimiento es completo sobre todas las actividades que realiza el alumno y se puede visualizar con respecto a cada actividad o dato preciso.

Cuadro 2.2: Comparativa de Seguimiento

2.7.3. Soporte del Software

	ATutor	Ilias	Moodle
Documentación	La documentación oficial de Atutor es completa y fácil de usar. Existen cursos sobre Atutor pero hay que estar registrados.	La documentación oficial no es muy descriptiva y específica, sólo ofrece una visión global. Hay documentación no oficial que está más completa (en inglés).	A parte de cursos de Moodle en la página oficial, también existe documentación completa y detallada tanto oficial como no oficial para el administrador, profesores y usuarios.
Soporte técnico	Existen foros en la comunidad de Atutor de soporte.	Existen foros en la comunidad de Atutor de soporte.	Igual que los anteriores, además posee Chats que, a horas específicas, están disponibles para cualquier duda.

Cuadro 2.3: Comparativa de Soporte del Software

2.7.4. Funciones a disposición del alumno

	ATutor	Ilias	Moodle
Foro	Posee foros típicos por medio de hilos. Puede ser público o privado dependiendo del curso.	Los Foros vienen dados por preguntas. La persona que crea el curso decide en qué cursos lo quiere crear. Se pueden crear cursos para grupos.	Foro basado en hilos. También usado para las noticias y novedades. Configurable dependiendo de qué usuarios lo utilizarán si profesores o alumnos. También se pueden evaluar.

Sigue en la página siguiente.

	ATutor	Ilias	Moodle
Descarga de archivos	Cada profesor pone a disposición de los alumnos los archivos.	Tanto el profesor como un grupo puede cargar y poner una dirección URL para bajarse el archivo. No se carga al servidor.	Tanto los profesores como los alumnos pueden dejar archivos en foros o en las actividades donde puedan y los profesores lo permitan.
Lista de correo	Posee mensajería interna entre dos personas. Sin embargo, no se puede enviar de uno a varios, aunque sí se puede a todos los inscritos a un curso.	El administrador decide el acceso a los correos. Se pueden mandar tanto correos como mensajería privada a una o varias personas y a un grupo.	Se pueden enviar tanto correos, como mensajería privada entre dos personas.
Test on line	Los profesores pueden crear tests on-line y asignarle un tiempo de vida. Los alumnos no pueden saber las respuestas cuando terminan el test.	Los profesores pueden crear tests on-line y asignarle un tiempo de respuesta. Ofrece la posibilidad de linkar un documento que contenga la respuesta. Una vez terminado el test indica las respuestas correctas y cuales son las que ha puesto el estudiante.	Los profesores pueden crear tests on-line y asignarle un tiempo de vida. Los alumnos sí pueden saber las respuestas cuando terminan el test. También permiten bajar un link con un documento donde están las respuestas.
Chat	Posee un Chat basado en PHP en el que se puede entrar una vez matriculado en el curso.	Posee Chat mediante la instalación de un módulo.	Posee Chat, siempre que el profesor lo haya creado en un curso.

Sigue en la página siguiente.

	ATutor	Ilias	Moodle
Otros	Posee diferentes paquetes como pueden ser speaker, Chat, grupos colaborativos, pizarra, etc, pero pocos traducidas al español.	Posee Chat y otras actividades como módulos complementarios.	Módulos como el de reunión, libro, flash, wikis, encuestas, talleres, tareas, etc.

Cuadro 2.4: Comparativa de Funciones a disposición del alumno

2.7.5. Administración

	ATutor	Ilias	Moodle
Gestión y mantenimiento del LCMS	Apenas se accede al sistema como administrador se visualizan las opciones de cambio de rol y las características generales del sitio. El administrador tiene la posibilidad de editar sus datos, así como su rol, modificar los datos de un curso, el lenguaje, etc. La complejidad de esta labor depende del tamaño del portal.	El administrador tiene la responsabilidad de todo y además puede hacer todo lo que haga un docente. Puede gestionar cuentas, módulos de Ilias, los foros, el sistema (aspecto, lenguaje, base de datos, etc.), el seguimiento, etc.	Parecida a la de Atutor, puede gestionar todas las cuentas, todos los cursos, las variables del sistema, también puede cambiar de rol. La complejidad de esta labor depende del tamaño del portal.

Cuadro 2.5: Comparativa de Administración

2.7.6. Facilidad de uso

	ATutor	Ilias	Moodle
Alumnos	En general en todas las tareas comunes de los alumnos como son: crear una cuenta, acceder a un curso, hacer una actividad, participar en un foro o chat, mandar un mensaje privado o resolver algún problema con la ayuda de otros alumnos, se hace sin ningún problema.	En general en todas las tareas comunes de los alumnos como son, crear una cuenta, acceder a un curso, hacer una actividad, participar en un foro o chat, mandar un mensaje privado o resolver algún problema con la ayuda de otros alumnos se hace sin ningún problema.	En general en todas las tareas comunes de los alumnos como son, crear una cuenta, acceder a un curso, hacer una actividad, participar en un foro o chat, mandar un mensaje privado o resolver algún problema con la ayuda de otros alumnos se hace sin ningún problema. Moodle posee además una ayuda en cada contenido.
Profesores	Para las funciones de los profesores como son crear una cuenta, un curso, configurar éste, agregar actividades, etc, es amigable y fácil de usar. Posee un interfaz diferente al de los alumnos con lo que queda un poco engorroso.	Para las funciones de los profesores como son crear una cuenta, un curso, configurar éste, agregar actividades, etc, es amigable y no tiene mucha dificultad.	Para las funciones de los profesores como son crear una cuenta, un curso, configurar éste, agregar actividades etc. es amigable y fácil de usar, aunque pierde un poco de privilegios con respecto a Atutor, como puede ser a la hora de modificar el interfaz de la página.

Sigue en la página siguiente.

	ATutor	Ilias	Moodle
Administradores	Las funciones de administración en Atutor no tienen una complejidad mayor que las demás.	Complejidad similar a la de Atutor.	Los tres están en la misma línea, aunque Moodle mantiene el mismo interfaz independientemente del usuario.

Cuadro 2.6: Comparativa de Facilidad de uso

2.7.7. Otros

	ATutor	Ilias	Moodle
Idioma	Posee diferentes paquetes de distintas lenguas. En la versión analizada tenemos 8 lenguas, pero en desarrollo hay muchas más.	El castellano está disponible pero con algunas deficiencias.	Posee 34 lenguas, entre ellas el castellano, y se pueden especificar en cada curso.
Soporte Multi-plataforma	Soporta servidores Windows, Linux, y Unix.	Soporta Linux y Solaris por igual.	Soporta servidores Windows, Linux y Unix.
Experiencia instalada	Posee una gran comunidad y está numerosamente difundida en todo el mundo. La comunidad española no es muy grande.	Poco difundido y la comunidad es prácticamente alemana.	Posee una gran comunidad en castellano, además de su gran difusión, mucho mayor que las dos anteriores.

Sigue en la página siguiente.

	ATutor	Ilias	Moodle
Posibilidad de la importación de paquetes y módulos	Atutor puede importar cursos IMS. Una de sus puntos flacos es que no puede incorporar módulos como lo hace Moodle, sino que posee diferentes plataformas que se unen entre sí como son ACollab o AChat y que poseen interfaz diferente.	Ilias puede importar cursos IMS, también tiene la posibilidad de importar algunos módulos.	Puede importar cualquier paquete SCORM y próximamente podrá exportar e importar IMS. En lo que respecta a los módulos posee una gran extendibilidad en este sentido ya que tiene la posibilidad de instalar y desinstalar estos módulos que están en continuo desarrollo.

Cuadro 2.7: Comparativa de otras características

2.8. Elección de la plataforma para la gestión docente

Después de la comparación de todas estas características, la primera descartada sería Ilias debido a su pobre comunidad española y su insuficiente documentación y funcionalidad, además de su poca difusión.

Después de descartar Ilias, la elección estaría entre Moodle y Atutor. Aunque parezca una elección difícil, esto no es así. La elección sería Moodle fundamentalmente por las siguientes características:

- Moodle posee una filosofía constructora porque se centra en actividades interactivas del estudiante mientras que Atutor era fundamentalmente un entorno para "colgar apuntes" que con las nuevas versiones ha mejorado mucho, pero que, sin embargo, sigue sustentándose en una filosofía que afronta los cursos separando las actividades por tipo, es decir, están los foros por un lado, los exámenes por otro, los Chats por otro, etc. Esto en Moodle no es así pudiendo seguir una temática o un tiempo dependiendo de la configuración del curso aunque todo esto no es obligatorio.

- Moodle es modular. El profesor puede agregar o quitar actividades a su curso. A su vez, la plataforma también funciona de la misma manera pudiéndose desarrollar nuevos módulos y ser instalados. Atutor es monolítico, pues el profesor no tiene el diseño del aprendizaje en sus manos.
- Aunque ya ha sido evaluado en las tablas anteriores, la facilidad de uso en Moodle puede ser menos dificultosa que en Atutor debido a que éste cambia el interfaz dependiendo del usuario.
- La comunidad en castellano es muchísimo más superior la que tiene moodle que aTutor sin duda alguna, y el respaldo de la comunidad hace tener un número superior de foros que hacen que el desarrollo y la expansión de Moodle sea superior, sobre todo aquí en España.

Por supuesto, el hecho de elegir Moodle no es todo ventajas. También Atutor tiene algunas ventajas sobre Moodle como son:

- La posibilidad de integrar IMS, que sin duda es el futuro de los estándares e-learning después de que SCORM esté siendo desechado. El equipo de Moodle trabaja sobre ello y dentro de poco estará operativo en forma de módulo.

Capítulo 3

Especificación de requisitos

Corresponde a esta parte del documento revisar, desde el punto de vista técnico, las condiciones obtenidas durante el análisis de requisitos, deteniéndose en qué debe hacer el sistema y no en cómo debe hacerlo, que será objeto de la especificación del sistema. Primeramente, se realizará una descripción general o reconocimiento del problema, especificando cuáles son los objetivos y realizando una descripción del dominio de información del sistema. Seguidamente vamos a exponer los casos principales de uso y después comentaremos las clases utilizadas así como el diagrama de clases resultantes.

3.1. Descripción del dominio de información del sistema

Como se comentó en el capítulo de introducción, el principal objetivo de este proyecto es realizar un editor de código fuente en lenguaje C de modo colaborativo y que éste estuviera embebido dentro de una plataforma e-learning. A su vez este sistema tendría que darnos todas las posibilidades de un entorno de desarrollo y posibilitar la recolección de estadísticas sobre el código. Del mismo modo, debe permitir comunicarse a los usuarios de modo sencillo.

Para satisfacer estos requisitos, se ha propuesto dividir el proyecto en dos subsistemas que pueden ser analizados de modo independiente pero que están interconectados entre si. Estos son:

1. Subsistema Wikicode-Mod.
2. Subsistema Wikicode-Synchronizer.

De este modo, cada uno de estos subsistemas tendrá sus propios requisitos funcionales, su propio modelo conceptual y estará dividido en una serie de módulos específicos.

Hay que comentar de modo que sirva como nota aclaratoria para el resto del documento que ambos subsistemas pertenecen al módulo creado para Moodle en el proyecto y que no se necesita de la instalación de dos aplicaciones independientes. La división en dos subsistemas distinto se hace para mayor claridad en la documentación y debido a que la sincronización no está realizada para ninguna plataforma e-learning concreta.

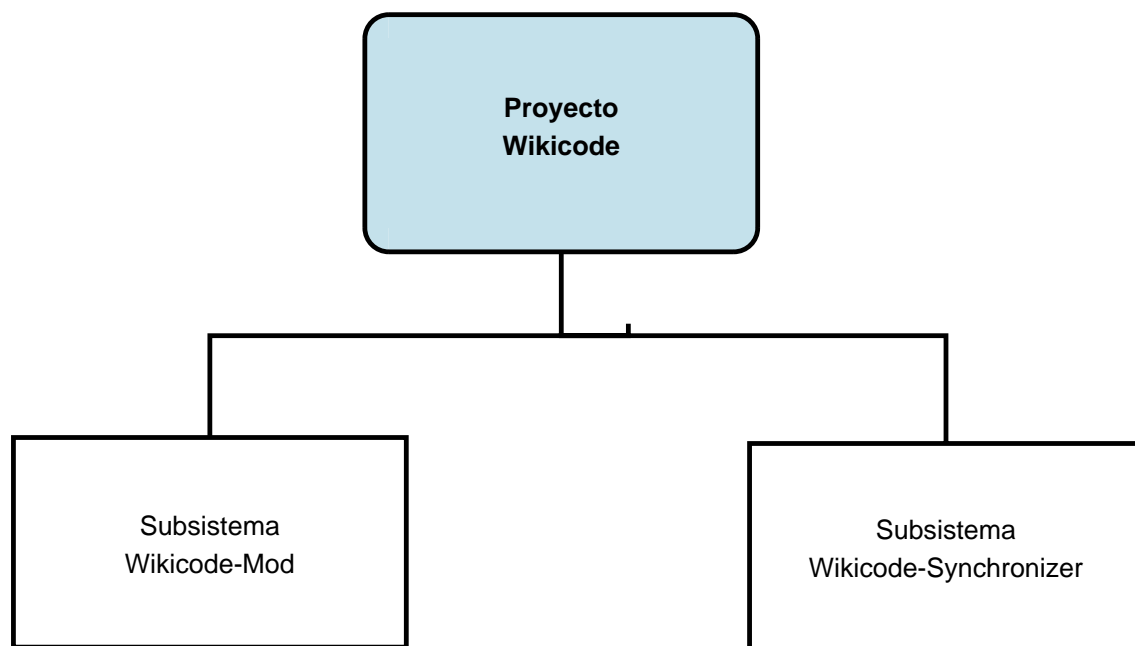


Figura 3.1: Gráfico de Subsistemas

3.1.1. Subsistema Wikicode-Mod

El subsistema *Wikicode-Mod* es el encargado de interactuar con el usuario y de aceptar o denegar las peticiones que este le pida. Será también el artífice de llevar la información al subsistema encargado de la sincronización del código fuente y de recibir el código actualizado y mostrarlo al usuario final. Dicho de otra manera, el subsistema *Wikicode-Mod* servirá de interfaz entre los usuarios y el entorno de programación embebido en Moodle.

Este subsistema se puede dividir en estos ocho módulos para realizar lo expuesto anteriormente:

1. Módulo para configurar el módulo dentro de la plataforma e-learning. A partir de este momento, se le llamará subsistema *Configurar*.

2. Módulo para visualizar el código fuente en texto plano. A partir de este momento, se le llamará subsistema *Visualización del Código*.
3. Módulo para editar el código fuente en lenguaje C. A partir de este momento, se le llamará subsistema *Edición del Código*.
4. Módulo para que los usuarios puedan interactuar de modo síncrono. A partir de este momento, se le llamará subsistema *Chat*.
5. Módulo para compilar el código fuente desarrollado en el proyecto. A partir de este momento, se le llamará subsistema *Compilación del Código*.
6. Módulo para visualizar las versiones antiguas del código fuente. A partir de este momento, se le llamará subsistema *Histórico de Código*.
7. Módulo para visualizar las estadísticas del código fuente. A partir de este momento, se le llamará subsistema *Log Proyecto*.
8. Módulo para administrar el proyecto. A partir de este momento, se le llamará subsistema *Administración*.

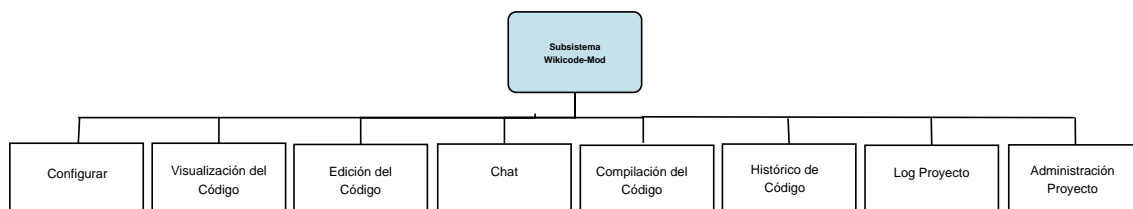


Figura 3.2: Gráfico de Subsistema Wikicode-Mod

A continuación se van a describir la principales funcionalidades y requisitos para cada subsistema:

3.1.1.1. Configurar

Módulo encargado de crear y parametrizar de los proyectos Wikicode embebidos en Moodle.

Los requisitos del módulo *Configurar* son:

- **RC1:** Se debe indicar la ruta del compilador o compiladores con los que se quiere crear la aplicación.

- **RC2:** Se debe posibilitar al tutor de personalizar si quiere que el código se edite de modo individual o grupal.
- **RC3:** Se debe permitir la parametrización de otros campos informativos como Nombre, Descripción, etc.

3.1.1.2. Visualización del Código

Módulo encargado de mostrar al usuario el código fuente desarrollado en texto plano.

Los requisitos del módulo *Visualización del Código* son:

- **RVC1:** Se debe mostrar el código fuente desarrollado sin ninguna etiqueta.
- **RVC2:** Se debe permitir al tutor la posibilidad de elección de grupo en caso de que el módulo se haya desarrollado de modo grupal.

3.1.1.3. Edición del Código

Módulo encargado de permitir la escritura al usuario de código fuente y a su vez de impedir acciones no permitidas. También debe de informar de una serie de requisitos que se marcaban como objetivos.

Los requisitos del módulo *Edición del Código* son:

- **REC1:** Permitir al usuario el desarrollo de código fuente en lenguaje C.
- **REC2:** Se debe proporcionar al usuario información para ayudarlo en la escritura del código fuente, mostrando la línea de código sobre la que nos encontramos actualmente.
- **REC3:** Se ha de parsear el código, ayudándose de colores y estilos tipográficos, para informar al usuario final de palabras claves, cadenas, definiciones, comentarios y funciones.
- **REC4:** Se debe impedir al usuario editar una parte bloqueada por otro usuario.
- **REC5:** Se debe de informar al usuario sobre qué partes están bloqueadas y qué usuario la tiene bloqueada.
- **REC6:** Se debe de bloquear automáticamente la función que el usuario está modificando para impedir que otros usuarios puedan cambiarla.

3.1.1.4. Chat

Módulo encargado de permitir la interlocución entre los distintos usuarios de un proyecto Wikicode.

Los requisitos del módulo *Chat* son:

- **RCH1:** Se debe permitir enviar mensajes y recibirlos de forma grupal y mostrarlos en el interfaz del proyecto.
- **RCH2:** Se deben de conservar los mensajes enviados por todos los usuarios para poder mostrarse de modo posterior.
- **RCH3:** Se debe de mostrar además del mensaje el usuario que lo ha enviado y el momento en que fue enviado.

3.1.1.5. Compilación del Código

Módulo encargado de compilar el código fuente y de mostrarnos los errores en caso de que lo hubiera.

Los requisitos del módulo *Compilación del Código* son:

- **RCC1:** Se debe de enviar al compilador el código fuente conforme a la versión más actualizada del proyecto, teniendo en cuenta las posibles modificaciones de otros usuarios.
- **RCC2:** En caso de que haya errores en la programación del código, se nos debe de informar de en que línea se han producido dichos errores y mostrarnos una ayuda de como debemos solucionarlos.

3.1.1.6. Histórico de Código

Módulo encargado de informar de la historia del código fuente y de informarnos de los cambios realizados.

Los requisitos del módulo *Histórico de Código* son:

- **RHC1:** Debe mostrar todas las versiones del código desde su creación hasta el actual.
- **RHC2:** Debe permitir al usuario restaurar una versión antigua.

3.1.1.7. Log Proyecto

Módulo encargado de ofrecer estadísticas que se han ido recopilando durante la creación y modificación del código fuente.

Los requisitos del módulo *Log Proyecto* son:

- **RLP1:** Se debe de informar del tiempo empleado para la realización del código fuente.
- **RLP2:** Se debe de comunicar al usuario los errores compilando que se han producido durante la realización del proyecto.

3.1.1.8. Administración

Módulo encargado de facilitar al tutor las opciones necesarias para administrar el proyecto Wikicode.

Los requisitos del módulo *Administración* son:

- **RA1:** No se debe permitir el acceso a usuarios sin el rol de profesor o administrador.
- **RA2:** Permitir al tutor/administrador modificar las opciones creadas durante la configuración y, del mismo modo, eliminar información que no desee almacenar.

3.1.2. Subsistema Wikicode-Synchronizer

El subsistema *Wikicode-Synchronizer* es el encargado de interactuar con la base de datos y crear los tags oportunos en el código para permitir los bloqueos y desbloques necesarios en el código fuente. Dicho de otro modo, es el encargado de hacer de interfaz entre el módulo de Moodle y los datos, modificándolos de manera abstracta al usuario final.

Debido a la naturaleza de los datos, se ha considerado oportuno el dividir este subsistema en los siguiente módulos.

1. Módulo que realiza la interacción con la base de datos para salvar/cargar el código. A partir de este momento, se le llamará subsistema *Salvar código*.
2. Módulo para bloquear y desbloquear funciones dentro del código. A partir de este momento, se le llamará subsistema *Locker/Unlocker código*.

3. Módulo para actualizar las estadísticas referentes al código que se está tratando. A partir de este momento, se le llamará subsistema *Actualizar Log*.

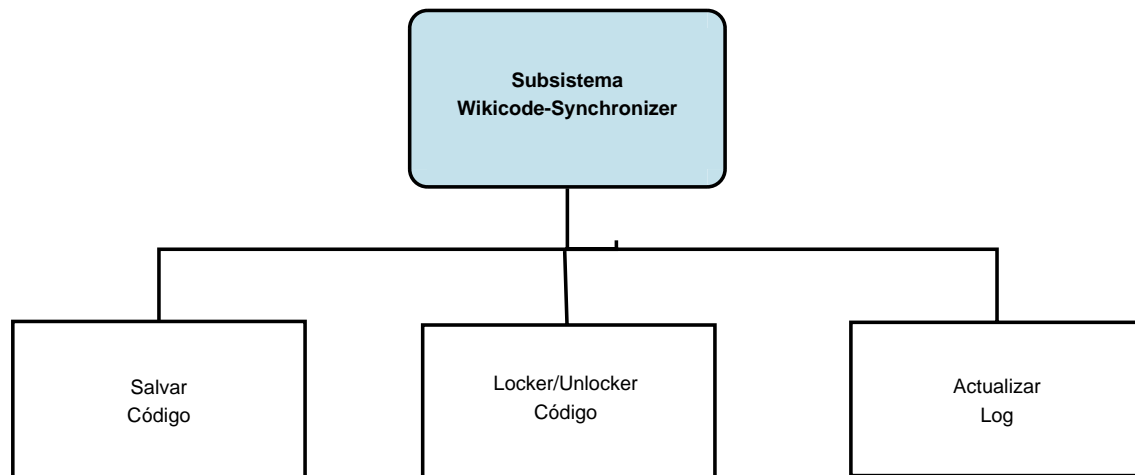


Figura 3.3: Gráfico de Subsistema Wikicode-Synchronizer

A continuación se van a describir la principales funcionalidades y requisitos para cada subsistema:

3.1.2.1. Salvar código

Módulo encargado de hacer de interfaz entre la base de datos y el proyecto Wikicode.

Los requisitos del módulo *Salvar código* son:

- **RSC1:** Se debe de obtener el código fuente del editor cada vez que se pulse una tecla clave.
- **RSC2:** Se considerará tecla clave el punto y coma (;), el intro y la apertura de corchetes ({}).
- **RSC3:** Se debe, tras comprobar que la función está bloqueada por el usuario, cambiar la función que ha sido modificada por el usuario en el código fuente global.
- **RSC4:** Se debe, si la función está bloqueada por otro usuario, devolver un código de error y no almacenar nada en base de datos.
- **RSC5:** Se debe, si el código ha sido modificado, enviar al editor el nuevo código fuente modificado. Este código debe tener embebida la información sobre sus bloqueos.

3.1.2.2. Locker/Unlocker código

Módulo encargado de recibir el código fuente y bloquear o desbloquear funciones del código.

Los requisitos del módulo *Locker/Unlocker código* son:

- **RLU1:** Se debe de bloquear el código de una función, si el usuario así lo desea, desde el principio de sus comentarios hasta la finalización del último corchete.
- **RLU2:** Si se desea bloquear una función ya bloqueada el sistema debe devolver un mensaje de error.
- **RLU3:** Se debe de permitir a cualquier usuario bloquear una función si esta ha sido desbloqueada.
- **RLU4:** Se debe de permitir a un usuario que tenga una función bloqueada desbloquearla cuando este desee.

3.1.2.3. Actualizar Log

Módulo encargado de actualizar la información referente a las estadísticas del código sobre el que se está trabajando.

Los requisitos del módulo *Actualizar Log* son:

- **RAL1:** La duración del proyecto se contabilizará desde el inicio de su escritura hasta la última modificación realizada.
- **RAL2:** Cada vez que exista una compilación errónea se aumentará en uno este dato. Se inicializará a cero.

3.2. Especificación de la interfaz

En este apartado se pretende analizar la interfaz de la aplicación a desarrollar. Partiendo de la interfaz utilizada por la plataforma Moodle, se ha intentado mantener la sencillez y claridad de la misma. Realmente, la interfaz de nuestra aplicación consiste en un módulo de Moodle que, como hemos comentado, mantendrá su sencillez, claridad y estilo. Este módulo contiene fundamentalmente una serie de acciones que son las que puede elegir el usuario: Creación, Visualización, Edición, Visualización del Histórico y Visualización de Estadísticas.

De esta manera, las interfaces que se encontrará el usuario al acceder a un curso determinado a través de moodle son:

- Estructura general de la interfaz del curso. Esta estructura vendrá presentada en la siguiente figura:



Figura 3.4: Estructura general de la interfaz del curso

Cuando un usuario desee acceder a la aplicación desarrollada o un tutor desee tanto entrar como crear un nuevo código fuente para que lo editen los usuarios lo hará desde la zona de recursos del curso.

- Estructura general de creación de un nuevo módulo Wikicode. Esta estructura vendrá presentada en la siguiente figura:

En este esquema podemos observar que se presenta un formulario a rellenar con los datos necesarios para la creación de un nuevo módulo Wikicode. Una vez queden introducidos o seleccionados, habrá que confirmar los cambios para que queden registrados en el sistema.

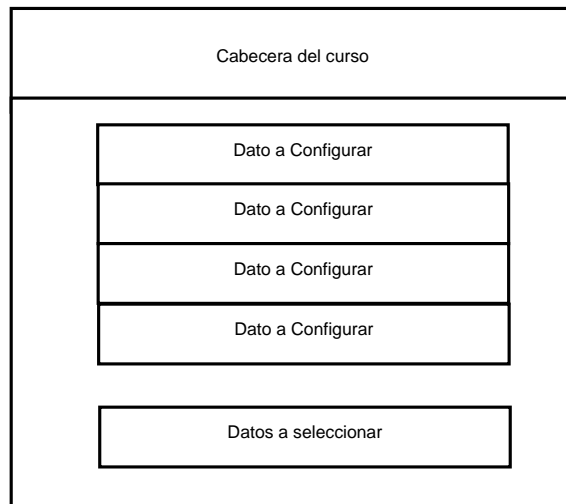


Figura 3.5: Estructura de la interfaz de creación

- Estructura general de visualización del código fuente en un módulo Wikicode. Esta estructura vendrá presentada en la siguiente figura:

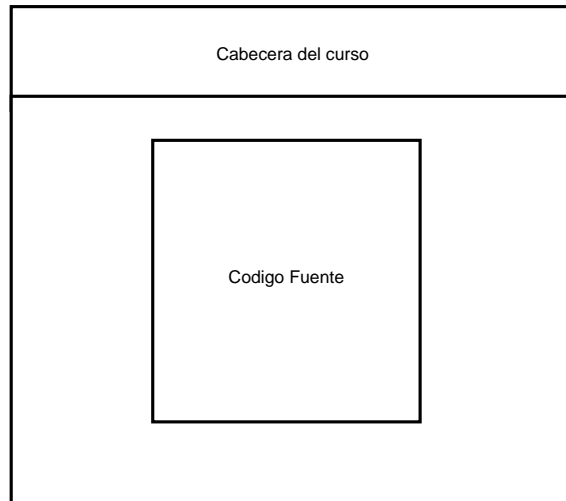


Figura 3.6: Estructura de la interfaz de vista

En este esquema podemos observar que se mantiene la cabecera del curso, en el que tendremos un menú superior para navegar por las opciones de la Wikicode, y un cuadro de texto con el código fuente que ha sido desarrollado.

- Estructura general de edición del código fuente en un módulo Wikicode. Esta estructura vendrá presentada en la siguiente figura:

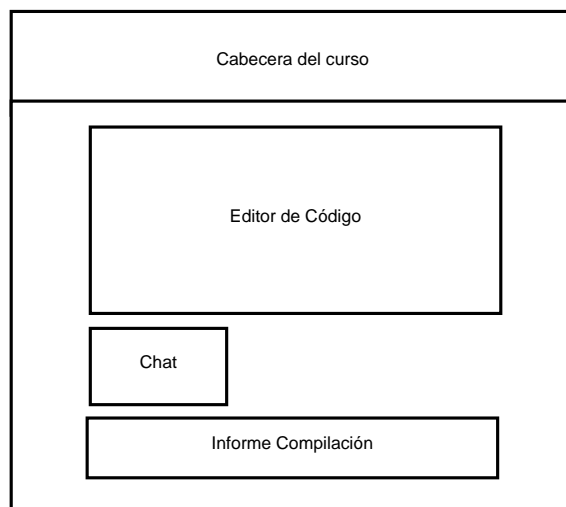


Figura 3.7: Estructura de la interfaz de edición del código fuente

En esta estructura, vemos como en la parte superior conservamos la cabecera del curso para poder navegar por sus opciones. En el resto del cuerpo tenemos un editor de código para que el usuario pueda escribir el programa, un chat para comunicarse de modo síncrono con el resto de usuarios e información sobre los posibles errores de compilación.

- Estructura general del histórico en un módulo Wikicode. Esta estructura vendrá presentada en la siguiente figura:

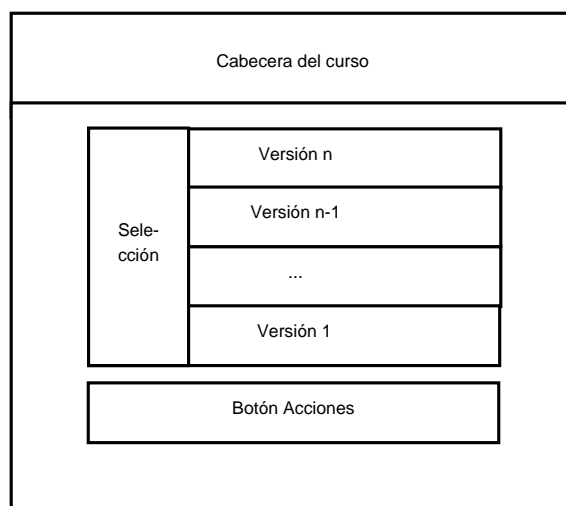


Figura 3.8: Estructura de la interfaz del histórico de un código fuente

Al igual que con las anteriores interfaces en esta seguimos manteniendo la misma estructura, un menú superior con información sobre el curso y la posibilidad de navegar más

rápidamente. En el cuerpo se mostrarán todas las versiones desarrolladas por los usuarios, un *radiobutton* para posibilitar elegir alguna en concreto y una botonera para poder realizar acciones sobre ellas.

- Estructura general de visualización de las estadísticas de un módulo Wikicode. Esta estructura vendrá presentada en la siguiente figura:

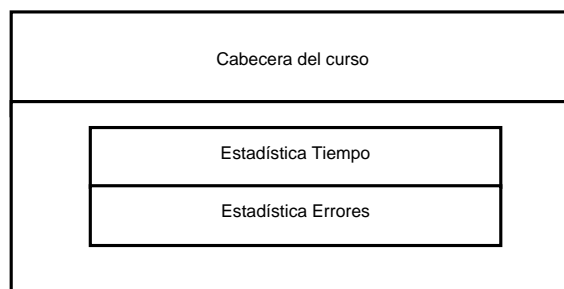


Figura 3.9: Estructura de la interfaz de la información estadística de una Wikicode

De nuevo mantenemos la cabecera en la parte superior del interfaz, con las ventajas que ya hemos comentado con anterioridad de navegación, y en el cuerpo de esta mostraremos los resultados respecto a las estadísticas calculadas. De esta manera en primer lugar se presentará el tiempo necesitado y a continuación se informará del número de errores producidos.

3.3. Modelo funcional

Para explicar el modelo funcional utilizado vamos a aplicar casos de uso. Un caso de uso se define como una colección de escenarios con éxitos y fallo relacionados, que describe a los actores utilizando un sistema para satisfacer un objetivo.

Como hemos comentado anteriormente, el proyecto se ha dividido en dos subsistemas que pueden ser estudiados por separado: Wkicode-Mod *Módulo de Moodle* y Wikicode-Synchronizer *Tratamiento de datos*. Por tanto, especificaremos los casos de uso así como los actores que están involucrados para cada uno de los sistemas.

3.3.1. Identificación de actores

Antes de nada tendremos que identificar los actores que interactuarán con nuestros sistemas. De esta manera, contaremos con los siguientes actores:

3.3.1.1. Tutor

A continuación se especifican las características que debe tener el actor *Tutor*:

ACTOR: Tutor	
Identificador	Tutor
Id-Padre	No tiene
Descripción	Persona encargada de crear la Wikicode y de configurar los parámetros necesarios. Está involucrada con el sistema Wikicode-Mod.

Cuadro 3.1: Descripción del Tutor

3.3.1.2. Usuario

A continuación se especifican las características que debe tener el actor *Usuario*:

ACTOR: Usuario	
Identificador	Usuario
Id-Padre	No tiene
Descripción	Persona encargada de editar el código fuente, comunicarse vía chat con otros usuarios y llamar al compilador de Código fuente. También tiene la posibilidad de navegar por el histórico del código y visualizar las estadísticas de la Wikicode. Está involucrada con el sistema Wikicode-Mod.

Cuadro 3.2: Descripción del Tutor

3.3.1.3. Módulo Wikicode-Mod

A continuación se especifican las características que debe tener el actor *Módulo Wikicode-Mod*:

ACTOR: Módulo Wikicode-Mod	
Identificador	Módulo Wkicode-Mod
Id-Padre	No tiene
Descripción	Módulo encargado de recibir las peticiones del usuario, así como la configuración para la creación del mismo. Es el encargado de llamar al compilador y de mostrarnos el mensaje de este. A su vez es el que se ocupa de escribir y leer los ficheros referentes al chat. También se encargará de hacer las peticiones necesarias al subsistema encargado de sincronizar y almacenar el código, así como de leer y mostrar el código fuente una vez haya sido devuelto. Está involucrada con el sistema Wikicode-Synchronizer.

Cuadro 3.3: Descripción del Módulo Wikicode-Mod

3.3.1.4. Módulo Wikicode-Synchronizer

A continuación se especifican las características que debe tener el actor *Módulo Wikicode-Synchronizer*:

ACTOR: Módulo Wikicode-Synchronizer	
Identificador	Módulo Wikicode-Synchronizer
Id-Padre	No tiene
Descripción	Módulo encargado de hacer las modificaciones necesarias en el código fuente, así como desbloquear y bloquear para un usuario específico partes de este. Es el encargado de actualizar las estadísticas relacionadas con la Wikicode que se está trabajando. También debe de enviar al subsistema encargado de mostrar el código éste una vez actualizado. Está involucrada con el sistema Wikicode-Mod.

Cuadro 3.4: Descripción del Módulo Wikicode-Synchronizer

3.3.2. Casos de uso de trazo grueso

El siguiente paso para describir el modelo funcional es identificar los principales casos de uso.

Para el sistema Wikicode éstos serán los siguientes:

- C1. Caso de uso Creación y Parametrización de la actividad Wikicode. Cuando el *Tutor* seleccione en la pantalla principal del curso que desea crear una nueva Wikicode para los usuarios del curso deberá configurar una serie de opciones. Una vez creada, el *Tutor* también podrá administrar dichas opciones.
- C2. Caso de uso Visualización del código desarrollado. Tanto el *Tutor* como el *Alumno* pueden acceder a visualizar una Wikicode sin necesidad de querer realizar una acción sobre ella.
- C3. Caso de uso Edición del Código. Una vez esté creada una Wikicode, lo primero que hay que hacer es insertar contenido en ella, la cual se hace desde un editor web embebido dentro de la propia plataforma e-learning. El *Alumno* o grupo de estos podrá desarrollar libremente el código fuente. Durante la edición se actualizarán estadísticas.
- C4. Caso de uso Bloqueo del Código. Una vez que el *Alumno* empieza a editar un código fuente, automáticamente se bloquea la función que está modificando o creando para que otros usuarios no puedan acceder a ella.
- C5. Caso de uso Desbloqueo del Código. Una vez que el *Alumno* ha terminado una función dentro de un código fuente, debe acudir a la opción que tiene en su interfaz de desbloquear funciones y marcar cual quiere desbloquear para que otro *Alumno* pueda modificarla.
- C6. Caso de uso Salvar Código. Cada vez que un *Alumno* pulse una tecla clave en el editor, este realizará una petición al subsistema encargado de sincronizar su información y esta se almacenará en base de datos.
- C7. Caso de uso Interacción mediante Chat con otros usuarios. Un *Alumno* puede tener la necesidad de comunicarse con otros miembros del grupo que estén colaborando para modificar una Wikicode, y puede hacerlo a través de esta herramienta que de tipo chat que llegará como mensaje de aviso al resto usuarios.
- C8. Caso de uso Compilación de código fuente. Un *Alumno* que así lo desee puede llamar a un compilador y este, a través del interfaz diseñado para la edición de código fuente, mostrar los resultados de la compilación de su código. Si la compilación es correcta puede descargar el ejecutable si así lo desea.
- C9. Caso de uso Consulta de histórico y restauración de versiones. Tanto un *Alumno* como un *Tutor* pueden acceder al histórico de una Wikicode para consultar el código

en otras fechas. Una vez dentro de una versión antigua comprobarán el código y lo restaurarán si así se desea.

- C10. Caso de uso Consulta de Estadísticas. Un *Alumno*, así como un *Tutor*, pueden comprobar la información estadística de una Wikicode si así lo desean. En ella pueden ver la duración de edición de código y el número de compilaciones erróneas.

3.3.2.1. Casos de uso para el sistema Wikicode

CASO DE USO C1: Creación y Parametrización de la actividad Wikicode.

Este es el diagrama correspondiente al caso de uso C1:

CASO DE USO C1: Creación y Parametrización de la actividad Wikicode	
Identificador	C1
Id-Padre	No tiene
Id-Requisito de Soporte	RC1, RC2, RC3, RA1, RA2
Descripción	Se describe el proceso de creación y parametrización de la actividad Wikicode.

Cuadro 3.5: Caso de Uso C1: Creación y Parametrización de la actividad Wikicode

CASO DE USO C2: Visualización del código desarrollado. Este es el diagrama correspondiente al caso de uso C2:

CASO DE USO C2: Visualización del código desarrollado	
Identificador	C2
Id-Padre	No tiene
Id-Requisito de Soporte	RVC1, RVC2
Descripción	Se describe el proceso de visualización del código desarrollado.

Cuadro 3.6: Caso de Uso C2: Visualización del código desarrollado

CASO DE USO C3: Edición del código desarrollado. Este es el diagrama correspondiente al caso de uso C3:

CASO DE USO C3: Edición del Código	
Identificador	C3
Id-Padre	No tiene
Id-Requisito de Soporte	REC1, REC2, REC3, REC4, REC5, RAL1
Descripción	Se describe el proceso de edición del código fuente.

Cuadro 3.7: Caso de Uso C3: Edición del Código

CASO DE USO C4: Bloqueo del Código. Este es el diagrama correspondiente al caso de uso C4:

CASO DE USO C4: Bloqueo del Código	
Identificador	C4
Id-Padre	No tiene
Id-Requisito de Soporte	REC6, RLU1, RLU2, RLU3
Descripción	Se describe el proceso de bloqueo del código fuente.

Cuadro 3.8: Caso de Uso C4: Bloqueo del Código

CASO DE USO C5: Desbloqueo del Código. Este es el diagrama correspondiente al caso de uso C5:

CASO DE USO C5: Desbloqueo del Código	
Identificador	C5
Id-Padre	No tiene
Id-Requisito de Soporte	RLU4
Descripción	Se describe el proceso de desbloqueo del código fuente.

Cuadro 3.9: Caso de Uso C5: Desbloqueo del Código

CASO DE USO C6: Salvar Código. Este es el diagrama correspondiente al caso de uso C6:

CASO DE USO C6: Salvar Código	
Identificador	C6
Id-Padre	No tiene
Id-Requisito de Soporte	RSC1, RSC2, RSC3, RSC4, RSC5
Descripción	Se describe el proceso de salvar el código fuente.

Cuadro 3.10: Caso de Uso C6: Salvar Código

CASO DE USO C7: Interacción mediante Chat con otros usuarios. Este es el diagrama correspondiente al caso de uso C7:

CASO DE USO C7: Interacción mediante Chat con otros usuarios	
Identificador	C7
Id-Padre	No tiene
Id-Requisito de Soporte	RCH1, RCH2, RCH3
Descripción	Se describe el proceso de interacción mediante chat con otros usuarios.

Cuadro 3.11: Caso de Uso C7: Interacción mediante Chat con otros usuarios

CASO DE USO C8: Compilación de código fuente. Este es el diagrama correspondiente al caso de uso C8:

CASO DE USO C8: Compilación de código fuente	
Identificador	C8
Id-Padre	No tiene
Id-Requisito de Soporte	RCC1, RCC2, RAL2
Descripción	Se describe el proceso de compilación de código fuente.

Cuadro 3.12: Caso de Uso C8: Compilación de código fuente

CASO DE USO C9: Consulta de histórico y restauración de versiones. Este es el diagrama correspondiente al caso de uso C9:

CASO DE USO C9: Consulta de histórico y restauración de versiones	
Identificador	C9
Id-Padre	No tiene
Id-Requisito de Soporte	RHC1, RHC2
Descripción	Se describe el proceso de consulta de histórico y restauración de versiones.

Cuadro 3.13: Caso de Uso C9: Consulta de histórico y restauración de versiones

CASO DE USO C10: Consulta de Estadísticas. Este es el diagrama correspondiente al caso de uso C10:

CASO DE USO C10: Consulta de Estadísticas	
Identificador	C10
Id-Padre	No tiene
Id-Requisito de Soporte	RLP1, RLP2
Descripción	Se describe el proceso de consulta de Estadísticas.

Cuadro 3.14: Caso de Uso C10: Consulta de Estadísticas

3.3.3. Diagramas de caso de uso

3.3.3.1. Diagramas para el sistema Wikicode

- Diagrama inicial del sistema Wikicode:

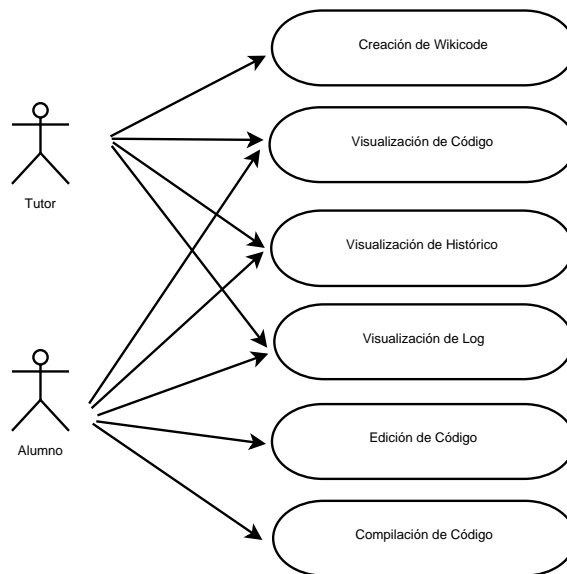


Figura 3.10: Diagrama inicial para el sistema Wikicode

- Diagrama del Caso de uso C1:

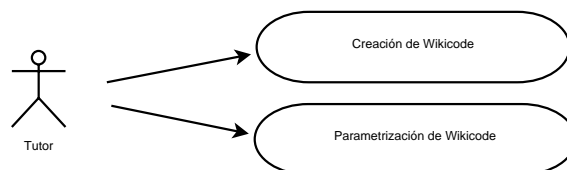


Figura 3.11: Diagrama Caso de uso C1

- Diagrama del Caso de uso C2:

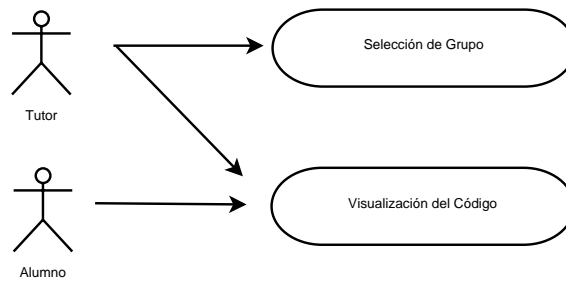


Figura 3.12: Diagrama Caso de uso C2

- Diagrama del Caso de uso C3:

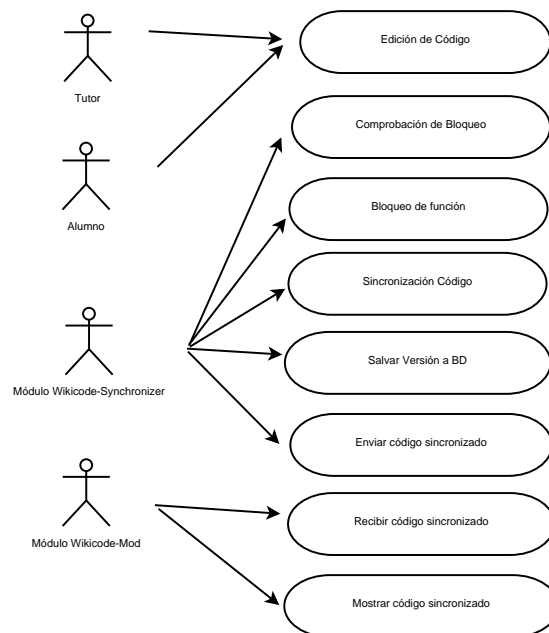


Figura 3.13: Diagrama Caso de uso C3

■ Diagrama del Caso de uso C4:

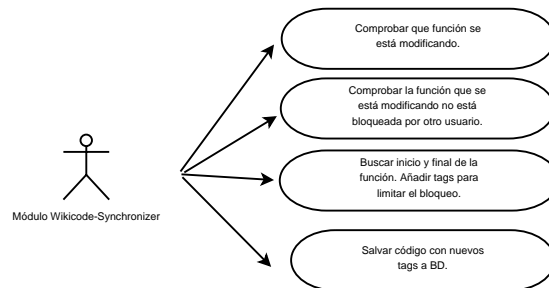


Figura 3.14: Diagrama Caso de uso C4

■ Diagrama del Caso de uso C5:

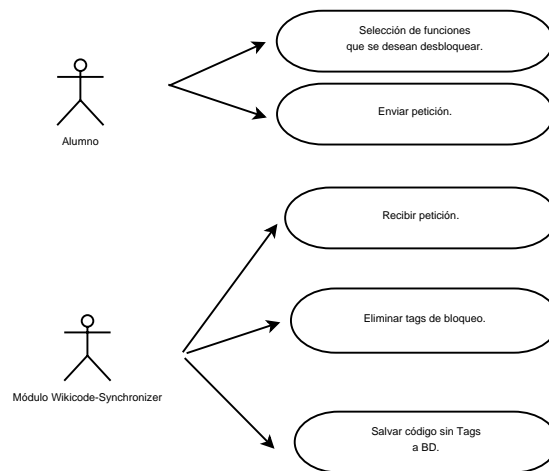


Figura 3.15: Diagrama Caso de uso C5

■ Diagrama del Caso de uso C6:

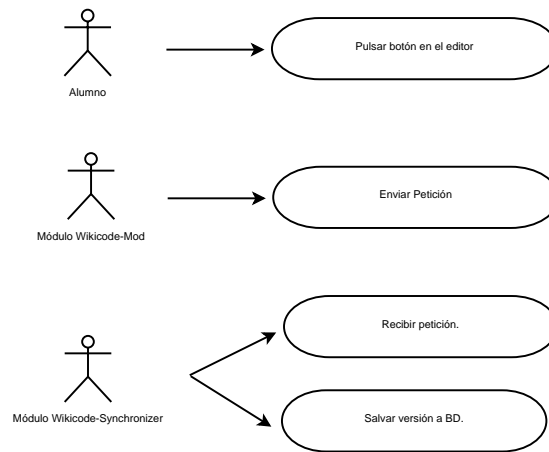


Figura 3.16: Diagrama Caso de uso C6

■ Diagrama del Caso de uso C7:

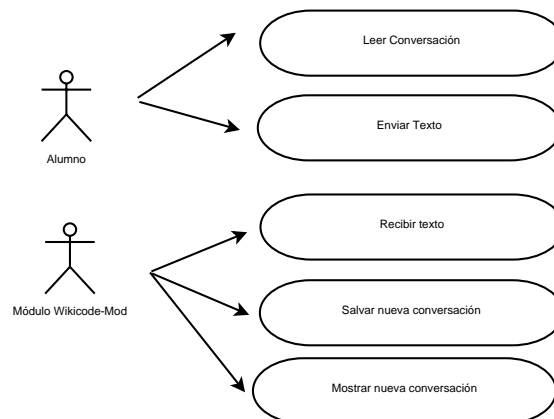


Figura 3.17: Diagrama Caso de uso C7

■ Diagrama del Caso de uso C8:

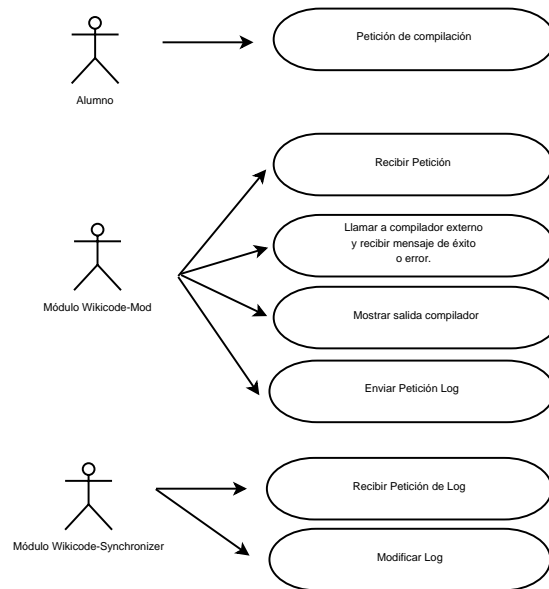


Figura 3.18: Diagrama Caso de uso C8

■ Diagrama del Caso de uso C9:

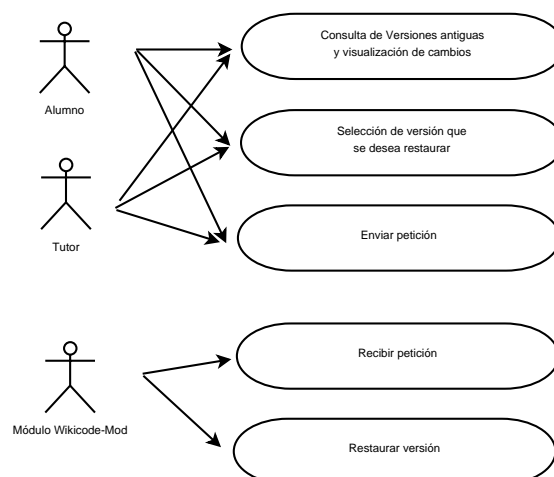


Figura 3.19: Diagrama Caso de uso C9

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
RCC1								X		
RCC2								X		
RHC1									X	
RHC2									X	
RLP1										X
RLP2										X
RA1	X									
RA2	X									
RSC1						X				
RSC2						X				
RSC3						X				
RSC4						X				
RSC5						X				
RLU1				X						
RLU2				X						
RLU3				X						
RLU4					X					
RAL1			X							
RAL2								X		

Cuadro 3.15: Matriz requisito / caso de uso

3.3.5. Especificación de los casos de uso

3.3.5.1. C1: Caso de uso Creación y Parametrización de la actividad Wikicode

La especificación del caso de uso C1 quedará especificada en la siguiente tabla:

ID	C1
Nombre	Creación y Parametrización de la actividad Wikicode
Descripción	El <i>Tutor</i> crea una nueva actividad Wikicode e introduce las opciones de configuración sobre las que desea trabajar
Actores	Tutor
Asunciones	Acceso correcto a la base de datos y a los ficheros de registro
Pasos	<ol style="list-style-type: none">1. El <i>Tutor</i> selecciona la creación de una nueva actividad Wikicode.2. El <i>Tutor</i> selecciona los parámetros de configuración que considere oportunos.
Variaciones	-
Requisitos no funcionales	-
Cuestiones	-

Cuadro 3.16: Especificación caso de uso C1

3.3.5.2. C2: Caso de uso Visualización del código desarrollado

La especificación del caso de uso C2 quedará especificada en la siguiente tabla:

ID	C2
Nombre	Visualización del código desarrollado
Descripción	Tanto el <i>Tutor</i> como el <i>Alumno</i> pueden acceder a visualizar el código desarrollado en la actividad Wikicode.
Actores	Tutor. Alumno.
Asunciones	Acceso correcto a la base de datos
Pasos	<ol style="list-style-type: none">1. El <i>Tutor</i> o el <i>Usuario</i> seleccionan esta opción en el bloque de Moodle correspondiente.2. Se visualiza el código fuente de la Wikicode seleccionada.
Variaciones	-
Requisitos no funcionales	-
Cuestiones	-

Cuadro 3.17: Especificación caso de uso C2

3.3.5.3. C3: Caso de uso Edición del código desarrollado

La especificación del caso de uso C3 quedará especificada en la siguiente tabla:

ID	C3
Nombre	Edición del código desarrollado
Descripción	El <i>Alumno</i> editará el código en un campo de texto preparado para ello, a su vez el <i>Tutor</i> también podrá realizar dicha acción si la considera oportuno. Del mismo modo de manera automática se llamará al subsistema encargado de la sincronización y los bloqueos y devolverá el código actualizado al módulo encargado de llevarlo al interfaz.
Actores	Tutor. Alumno. Módulo Wikicode-Synchronizer. Módulo Wikicode-Mod.
Asunciones	Acceso correcto a la base de datos y a los ficheros de registro. Conexión entre subsistemas mediante el envío de ficheros de tipo JSON.
Pasos	<ol style="list-style-type: none">1. El <i>Alumno</i> o <i>Tutor</i> editan una función en el editor de Código.2. El <i>Alumno</i> o <i>Tutor</i> pulsan una tecla clave: Punto y coma, Intro, Apertura de Corchete, Retroceso.3. El <i>Módulo Wikicode-Mod</i> detecta esta acción y envía al Módulo Wikicode-Synchronizer el código correspondiente al editor y la posición del cursor dentro del código.
Continúa en la siguiente página	

Pasos	<ol style="list-style-type: none">4. El <i>Módulo Wikicode-Synchronizer</i> comprueba si la función no está bloqueada, en caso negativo la bloquea, y sincroniza el código con el del resto de usuarios.5. El <i>Módulo Wikicode-Synchronizer</i> salva el código en base de datos y envía el código sincronizado al <i>Módulo Wikicode-Mod</i>.6. El <i>Módulo Wikicode-Mod</i> recibe el código y actualiza el editor con el código sincronizado al resto de usuarios.
Variaciones	-
Requisitos no funcionales	-
Cuestiones	-

Cuadro 3.18: Especificación caso de uso C3

3.3.5.4. C4: Caso de uso Bloqueo del código

La especificación del caso de uso C4 quedará especificada en la siguiente tabla:

ID	C4
Nombre	Bloqueo del Código
Descripción	El <i>emph</i> módulo <i>Wikicode-Synchronizer</i> recibirá el carácter que se está modificando, buscará a que función pertenece y tras comprobar que esta no está bloqueada la bloqueará para el usuario que ha ordenado la acción.
Actores	Módulo <i>Wikicode-Synchronizer</i>
Asunciones	Acceso correcto a la base de datos
Pasos	<ol style="list-style-type: none">1. El <i>Módulo Wikicode-Synchronizer</i> comprueba que función se está modificando.2. Se comprueba que esta función no esté bloqueada por ningún otro usuario, y en caso de que esta condición se cumpla, se delimitan la función mediante la inserción de tags para marcar el inicio y el fin del bloqueo. Dentro de esos tags también se indica el usuario que la ha bloqueada para facilitar las acciones.3. Se salva el código con los nuevos tags a la base de datos.
Variaciones	-
Requisitos no funcionales	-
Cuestiones	-

Cuadro 3.19: Especificación caso de uso C4

3.3.5.5. C5: Caso de uso Desbloqueo del código

La especificación del caso de uso C5 quedará especificada en la siguiente tabla:

ID	C5
Nombre	Desbloqueo de Código
Descripción	El <i>emphAlumno</i> seleccionará las funciones que desea desbloquear de su lista de funciones bloqueadas, las cuales se envían al submódulo encargado de desbloquear el código.
Actores	Alumno. Módulo Wikicode-Mod. Módulo Wikicode-Synchronizer.
Asunciones	Acceso correcto a la base de datos. Envío correcto de fichero JSON.
Pasos	<ol style="list-style-type: none"> 1. El <i>Alumno</i> pulsa el botón correspondiente a la acción de desbloquear. 2. El <i>Módulo Wikicode-Mod</i> muestra al <i>Alumno</i> una lista con todas las funciones que tiene bloqueadas. El alumno las selecciona y estas se envían mediante un fichero JSON al <i>módulo Wikicode-Synchronizer</i>. 3. El <i>Módulo Wikicode-Synchronizer</i> tras recibir la petición elimina los tags de bloqueo y salva el código a base de datos.
Variaciones	-
Requisitos no funcionales	-
Cuestiones	-

Cuadro 3.20: Especificación caso de uso C5

3.3.5.6. C6: Caso de uso Salvar Código

La especificación del caso de uso C6 quedará especificada en la siguiente tabla:

ID	C6
Nombre	Salvar Código
Descripción	El <i>Alumno</i> hace una llamada al subsistema encargado del módulo Wikicode y este llamará al subsistema que debe salvar el código en base de datos.
Actores	Alumno. Módulo Wikicode-Mod. Módulo Wikicode-Synchronizer.
Asunciones	Acceso correcto a la base de datos. Envío correcto de fichero JSON.
Pasos	<ol style="list-style-type: none">1. El <i>Alumno</i> selecciona que desea salvar al código y esta es comunicada al <i>Módulo Wikicode-Mod</i>.2. El <i>Módulo Wikicode-Mod</i> prepara un fichero JSON con la información que desea salvar y se envía al <i>Módulo Wikicode-Mod</i>.3. Este tras recibir la petición almacena en base de datos una versión del histórico para que pueda ser consultada posteriormente.
Variaciones	-
Requisitos no funcionales	-
Cuestiones	-

Cuadro 3.21: Especificación caso de uso C6

3.3.5.7. C7: Caso de uso Interacción mediante Chat con otros usuarios

La especificación del caso de uso C7 quedará especificada en la siguiente tabla:

ID	C7
Nombre	Interacción mediante Chat con otros usuarios
Descripción	El <i>Alumno</i> leerá la información del chat y, si así lo desea, podrá enviar un mensaje a la conversación. Este mensaje se realizará a través del subsistema <i>Wikicode-Mod</i> .
Actores	Alumno. Módulo Wikicode-Mod.
Asunciones	Acceso correcto a los ficheros de registro
Pasos	<ol style="list-style-type: none">1. El <i>Módulo Wikicode-Mod</i> lee el fichero que se corresponde a la Wikicode que estamos trabajando y la muestra en pantalla.2. El <i>Alumno</i> lee la conversación y si así lo desea puede enviar un nuevo mensaje a esta.3. El <i>Módulo Wikicode-Mod</i> lee el mensaje enviado y modifica el fichero de conversación con el nuevo mensaje.
Variaciones	-
Requisitos no funcionales	Esta operación se repite cada 5 segundos para actualizar la información.
Cuestiones	-

Cuadro 3.22: Especificación caso de uso C7

3.3.5.8. C8: Caso de uso Compilación de código fuente

La especificación del caso de uso C8 quedará especificada en la siguiente tabla:

ID	C8
Nombre	Compilación de código fuente
Descripción	El <i>Alumno</i> llamará al compilador a través de un botón. El módulo <i>Wikicode-Mod</i> hará una llamada al compilador pasándole como argumento el código fuente y mostrará su salida al usuario. También se aumentará el contador de estadísticas.
Actores	Alumno. Módulo Wikicode-Mod. Módulo Wikicode-Synchronizer.
Asunciones	Acceso correcto al compilador y a la base de datos. Acceso correcto a los ficheros de intercambio con el compilador.
Pasos	<ol style="list-style-type: none"> 1. El <i>Alumno</i> pulsa el botón de Compilar y este se comunica con el <i>módulo Wikicode-Mod</i>. 2. El <i>módulo Wikicode-Mod</i> crea un fichero con el código fuente y llama al compilador para que lo tome como parámetro. 3. El compilador crea un fichero de salida con sus posibles errores y el <i>módulo Wikicode-Mod</i> lo lee y se muestra al usuario. 4. El <i>módulo Wikicode-Mod</i> llama al <i>módulo Wikicode-Synchronizer</i> para que este aumente las estadísticas correspondientes a la compilación.
Variaciones	-
Requisitos no funcionales	-
Cuestiones	-

Cuadro 3.23: Especificación caso de uso C8

3.3.5.9. C9: Caso de uso Consulta de histórico y restauración de versiones

La especificación del caso de uso C9 quedará especificada en la siguiente tabla:

ID	C9
Nombre	Consulta de histórico y restauración de versiones
Descripción	El <i>Alumno</i> o el <i>Tutor</i> podrán consultar las versiones antiguas del código y restaurar alguna de ellas si así lo desean.
Actores	Alumno. Tutor. Módulo Wikicode-Mod.
Asunciones	Acceso correcto a la base de datos y a los ficheros de registro
Pasos	<ol style="list-style-type: none"> 1. El <i>Tutor</i> o el <i>Usuario</i> seleccionan esta opción en el bloque de Moodle correspondiente. 2. Se visualizan el histórico de la Wikicode seleccionada. 3. El <i>Tutor</i> o el <i>Usuario</i> seleccionan una versión y comprueban el código fuente de ésta. 4. El <i>Tutor</i> o el <i>Usuario</i> seleccionan si desean restaurar esta versión. 5. Si la acción anterior ha sido seleccionada como si, el módulo Wikicode-Mod recibe la petición y modifica el código fuente actual por el de la versión más antigua.
Variaciones	-
Requisitos no funcionales	-
Cuestiones	-

Cuadro 3.24: Especificación caso de uso C9

3.3.5.10. C10: Caso de uso Consulta de Estadísticas

La especificación del caso de uso C10 quedará especificada en la siguiente tabla:

ID	C10
Nombre	Consulta de Estadísticas
Descripción	El <i>Alumno</i> o el <i>Tutor</i> podrán consultar las estadísticas del código.
Actores	Alumno. Tutor.
Asunciones	Acceso correcto a la base de datos y a los ficheros de registro
Pasos	<ol style="list-style-type: none"> 1. El <i>Tutor</i> o el <i>Usuario</i> seleccionan esta opción en el bloque de Moodle correspondiente. 2. Se visualizan las estadísticas de la Wikicode seleccionada.
Variaciones	-
Requisitos no funcionales	-
Cuestiones	-

Cuadro 3.25: Especificación caso de uso C10

3.4. Modelo conceptual

Para analizar el modelo conceptual realizado se comentarán las diferentes clases que se han desarrollado y se mostrará el diagrama de clases resultante. Como se indicó en apartados anteriores, el proyecto se encuentra fundamentalmente dividido en dos grandes módulos o sistemas. Por tanto, las clases serán explicadas por separado y se realizará un diagrama de clases para cada uno de los sistemas.

De esta manera, una vez extraídas las clases necesarias y suficientes, se va a definir, primeramente, un diccionario de datos expresando el significado concreto de cada clase para evitar posibles interpretaciones erróneas. Como tenemos dos sistemas bien diferenciados, se definirá un diccionario de datos para cada uno de ellos.

3.4.1. Diccionario de datos para el sistema Wikicode-Mod

Este sistema está compuesto por varias clases, las cuales hacen de interfaz entre el usuario y el sistema. Se puede dividir según las opciones que se van a utilizar:

3.4.1.1. Class `mod_wikicode_renderer`

- Descripción: Se trata de una clase que genera el interfaz gráfico del módulo a través de un modelo previamente creado.
- Observaciones: Esta clase se utiliza para que el módulo Wikicode tenga una apariencia similar a la del resto de módulos Moodle. Hereda de la clase base de moodle encargada del renderizado y se sobrecarga una serie de funciones.
- Atributos: No tiene atributos.
- Operaciones:
 - **page_index:** Devuelve el código HTML de una página principal con el formato de Moodle.
 - **search_result:** Devuelve el código HTML de una búsqueda con el formato de Moodle.
 - **diff:** Devuelve el código HTML con la diferencia en el contenido de dos Wikicodes.
 - **diff_paging_bar:** Devuelve el código HTML para proveernos acceso a otras versiones.
 - **wikicode_info:** Devuelve el código HTML de un cuadro con información sobre la Wikicode.
 - **tabs:** Devuelve el código HTML de un menú superior con las opciones de navegación de una Wikicode.
 - **wikicode_print_subwiki_selector:** Devuelve el código HTML de un menú con las opciones de navegación entre distintas Wikicodes.
 - **menu_admin:** Devuelve el código HTML del menú que se mostrará en caso de que el usuario tenga rol de administrador a la hora de navegar por una Wikicode.
- Relaciones:
 - **plugin_renderer_base:** Hereda de esta clase.

- Diagrama de la clase:

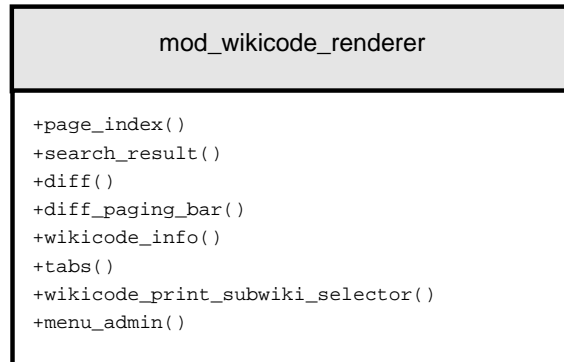


Figura 3.21: Class `mod_wikicode_renderer`

3.4.1.2. class `page_wikicode`

- Descripción: Contiene el código común entre todos los tipos de páginas
- Observaciones: Se trata de una clase abstracta que sirve de interfaz para el resto de clases que van a crear una página.
- Atributos:
 - `subwiki`:
 - Descripción: Identificador de la Wiki actualmente seleccionada.
 - Dominio: Variable de tipo entero.
 - `page`:
 - Descripción: Identificador de la página Wiki actualmente seleccionada.
 - Dominio: Variable de tipo entero.
 - `title`:
 - Descripción: Título de la página actual.
 - Dominio: Variable de tipo string.
 - `gid`:
 - Descripción: Identificador del grupo actualmente seleccionado.
 - Dominio: Variable de tipo entero.
 - `modcontext`:
 - Descripción: Contexto del módulo.
 - Dominio: Variable de tipo objeto.

- uid:
 - Descripción: Identificador del usuario actual.
 - Dominio: Variable de tipo entero.
 - tabs:
 - Descripción: Objeto con las opciones que son mostradas en el menú superior.
 - Dominio: Variable de tipo objeto.
 - tabs_options:
 - Descripción: Objeto con las opciones del menú superior.
 - Dominio: Variable de tipo objeto.
 - wikioutput:
 - Descripción: Objeto con las opciones de renderizado del módulo Wikicode seleccionado.
 - Dominio: Variable de tipo objeto.
- Operaciones:
- **print_header:** Esta operación devuelve el código HTML que imprime la parte superior de la página.
 - **print_pagetitle:** Método para imprimir el título de la página.
 - **setup_tabs:** Método para configurar el menú superior y dejar marcada la opción sobre la que estamos navegando.
 - **print_content:** Operación que debe ser sobrescrita por las clases que hereden de esta para imprimir el código HTML correspondiente al cuerpo central de la página.
 - **set_page:** Método para asignar la página actual.
 - **set_title:** Método para asignar el título de la página actual.
 - **set_gid:** Método para asignar el identificador de grupo.
 - **set_uid:** Método para asignar el identificador de usuario.
 - **print_footer:** Esta operación devuelve el código HTML que imprime la parte inferior de la página.
 - **set_session_url:** Operación para indicar a la variable global de Moodle \$SESSION la página web actual sobre la que estamos navegando.
 - **print_tab:** Imprime el menú superior con las opciones configuradas en la página.

■ Relaciones:

- **page_wikicode_view:** Padre de esta clase.
- **page_wikicode_edit:** Padre de esta clase.
- **page_wikicode_log:** Padre de esta clase.
- **page_wikicode_search:** Padre de esta clase.
- **page_wikicode_create:** Padre de esta clase.
- **page_wikicode_diff:** Padre de esta clase.
- **page_wikicode_history:** Padre de esta clase.
- **page_wikicode_restoreversion:** Padre de esta clase.
- **page_wikicode_viewversion:** Padre de esta clase.
- **page_wikicode_admin:** Padre de esta clase.

■ Diagrama de la clase:



Figura 3.22: Class page_wikicode

3.4.1.3. Class `page_wikicode_view`

- Descripción: Clase para mostrarnos una página de Wikicode en la que se muestra el código fuente como texto plano.
- Observaciones: Se considera que esta clase es la que hace de interfaz en el caso de uso *Visualización de código desarrollado*.
- Atributos:
 - `coursemodule`:
 - Descripción: Identificar del curso sobre el que está creado la Wikicode.
 - Dominio: Variable de tipo entero.
- Operaciones:
 - **`print_header`**: Esta operación devuelve el código HTML que imprime la parte superior de la página.
 - **`wikicode_print_subwiki_selector`**: Esta operación devuelve el código HTML con el menú para seleccionar sobre que Wikicode queremos trabajar.
 - **`print_content`**: Operación devuelve el código HTML que imprime la parte central de la página.
 - **`set_url`**: Operación que modifica las variables globales para indicar a Moodle en que página nos encontramos.
 - **`set_course_module`**: Método para asignar el identificador de curso de la clase.
 - **`create_navbar`**: Método para añadir al menú de navegación la opción de Vista.
- Relaciones:
 - **`page_wikicode`**: Hereda de esta clase.

- Diagrama de la clase:

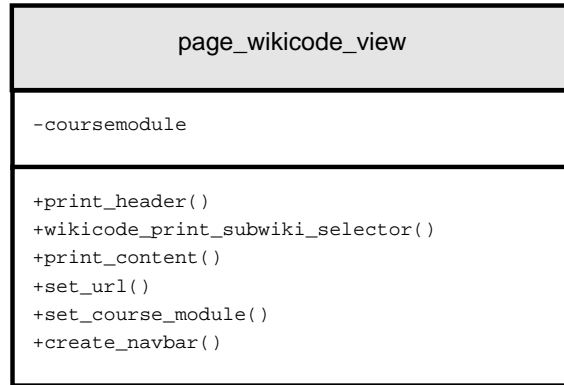


Figura 3.23: Class `page_wikicode_view`

3.4.1.4. Class `page_wikicode_edit`

- Descripción: Clase para mostrarnos una página de edición de Wikicode.
- Observaciones: Se considera que esta clase es la que hace de interfaz en los casos de uso *Edición de código desarrollado*, *Interacción mediante chat con otros usuarios* y *Compilación de código fuente*.
- Atributos:
 - compiled:
 - Descripción: Variable que nos describe si la página actual se está mostrando tras una compilación o aún no se ha intentando compilar.
 - Dominio: Variable de tipo entero.
- Operaciones:
 - **print_header:** Esta operación devuelve el código HTML que imprime la parte superior de la página.
 - **print_pagetitle:** Método para imprimir el título de la página.
 - **print_content:** Operación devuelve el código HTML que imprime la parte central de la página.
 - **set_url:** Operación que modifica las variables globales para indicar a Moodle en que página nos encontramos.
 - **set_session_url:** Método para modificar la variable global de Moodle referente a la dirección web de la sesión.

- **set_compiled:** Método para asignar la información referente a la variable compiled.
 - **print_edit:** Método que crea el código HTML que involucra al editor web, el chat y el compilador.
- Relaciones:
- **page_wikicode:** Hereda de esta clase.
 - **page_wikicode_compile:** Padre de esta clase.
 - **page_wikicode_save:** Padre de esta clase.
- Diagrama de la clase:

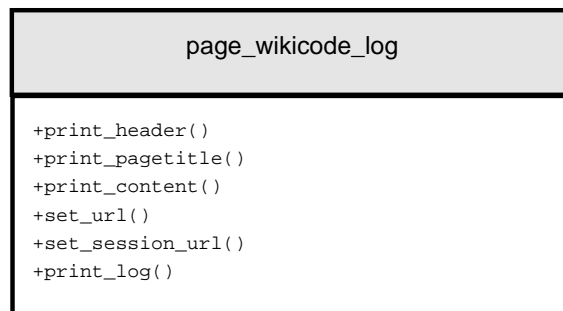


Figura 3.24: Class page_wikicode_edit

3.4.1.5. Class page_wikicode_log

- Descripción: Clase para mostrarnos una página de estadísticas sobre una Wikicode.
- Observaciones: Se considera que esta clase es la que hace de interfaz con el caso de uso *Consulta de estadísticas*.
- Atributos: No hay atributos.
- Operaciones:
- **print_header:** Esta operación devuelve el código HTML que imprime la parte superior de la página.
 - **print_pagetitle:** Método para imprimir el título de la página.

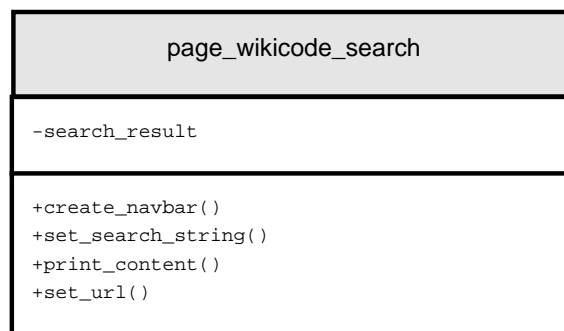
- **print_content:** Operación devuelve el código HTML que imprime la parte central de la página.
 - **set_url:** Operación que modifica las variables globales para indicar a Moodle en que página nos encontramos.
 - **set_session_url:** Método para modificar la variable global de Moodle referente a la dirección web de la sesión.
 - **print_log:** Método que crea el código HTML que nos muestra las estadísticas de la página Wikicode actual.
- Relaciones:
- **page_wikicode:** Hereda de esta clase.
- Diagrama de la clase:

Figura 3.25: Class `page_wikicode_log`

3.4.1.6. Class `page_wikicode_search`

- Descripción: Clase para mostrarnos una búsqueda dentro de las páginas de Wikicode.
- Observaciones: Esta clase sólo es utilizada a partir de la versión 2 de Moodle.
- Atributos:
- **search_result:**
 - Descripción: Variable que nos almacena un registro de base de datos con el resultado de la búsqueda que se desee.
 - Dominio: Variable de tipo objeto.
- Operaciones:

- **create_navbar:** Esta operación devuelve el código HTML de una barra de navegación.
 - **set_search_string:** Función que llama a base de datos para conseguir los datos de la búsqueda. Recibe como parámetros la cadena a buscar y el dominio sobre el que se quiere encontrar la información.
 - **print_content:** Operación devuelve el código HTML que imprime la parte central de la página.
 - **set_url:** Operación que modifica las variables globales para indicar a Moodle en que página nos encontramos.
- Relaciones:
 - **page_wikicode:** Hereda de esta clase.
 - Diagrama de la clase:

Figura 3.26: Class `page_wikicode_search`

3.4.1.7. Class `page_wikicode_create`

- Descripción: Clase para crear una nueva Wikicode dentro de un curso.
- Observaciones: Se considera que esta clase es la que hace de interfaz con el caso de uso *Creación y Parametrización de la actividad Wikicode*.
- Atributos:
 - format:
 - Descripción: Formato de la Wikicode que se va a crear. Actualmente sólo existe uno, pero se deja el código preparada por si se desean crear más formatos en el futuro.

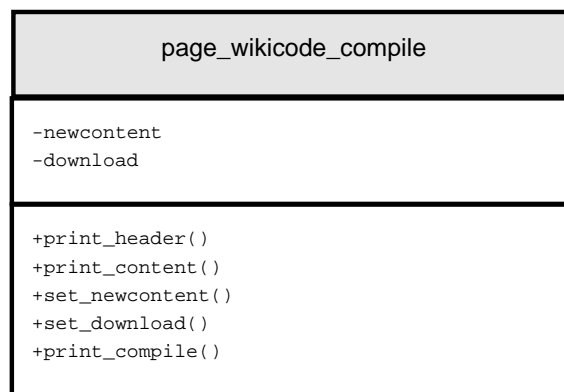
- Dominio: Variable de tipo string.
- swid:
 - Descripción: Identificador de la página subwiki seleccionada.
 - Dominio: Variable de tipo entero.
- wid:
 - Descripción: Identificador de la página wiki seleccionada.
 - Dominio: Variable de tipo entero.
- action:
 - Descripción: Cadena con la información sobre si se va a crear una página nueva o se está modificando una anterior.
 - Dominio: Variable de tipo string.
- mform:
 - Descripción: Objeto con la información sobre la nueva página que se está creando.
 - Dominio: Variable de tipo objeto.
- Operaciones:
 - **print_header**: Esta operación devuelve el código HTML que imprime la parte superior de la página.
 - **print_content**: Operación devuelve el código HTML que imprime la parte central de la página.
 - **create_navbar**: Esta operación devuelve el código HTML de una barra de navegación.
 - **set_url**: Operación que modifica las variables globales para indicar a Moodle en que página nos encontramos.
 - **create_page**: Método que crea el código HTML que nos muestra la página de parametrización de la Wikicode.
 - **set_format**: Método para asignar el formato de la Wikicode.
 - **set_wid**: Método para asignar el identificador de la Wikicode actual.
 - **set_swid**: Método para asignar el identificador de la subwiki actual.
 - **set_action**: Método para asignar la acción que se va a realizar.
- Relaciones:
 - **page_wikicode**: Hereda de esta clase.
- Diagrama de la clase:

Figura 3.27: Class `page_wikicode_create`

3.4.1.8. Class `page_wikicode_compile`

- Descripción: Clase que llama al compilador y nos muestra el intercambio de información con éste.
- Observaciones: Se considera que esta clase es la que hace de interfaz entre el *módulo Wikicode-Mod* y el compilador de C.
- Atributos:
 - `newcontent`:
 - Descripción: Código fuente sincronizado del editor que va a ser compilado.
 - Dominio: Variable de tipo string.
 - `download`:
 - Descripción: Identificador con la información sobre si se desea descargar el ejecutable en caso de que la compilación sea correcta.
 - Dominio: Variable de tipo entero.
- Operaciones:
 - **`print_header`**: Esta operación devuelve el código HTML que imprime la parte superior de la página.
 - **`print_content`**: Operación devuelve el código HTML que imprime la parte central de la página.

- **set_newcontent:** Operación que asigna el código fuente al atributo de la clase.
 - **set_download:** Método para asignar si se desea descargar el ejecutable.
 - **print_compile:** Operación de llamada al compilador y de recoger sus resultados. Para finalizar nos redirige a la página de edición con la nueva información.
- Relaciones:
- **page_wikicode_edit:** Hereda de esta clase.
- Diagrama de la clase:

Figura 3.28: Class `page_wikicode_compile`

3.4.1.9. Class `page_wikicode_diff`

- Descripción: Clase que nos muestra a través de un interfaz gráfico las diferencias entre dos versiones.
- Observaciones: Sólo disponible a partir de la versión 2 de Moodle.
- Atributos:
- **compare:**
 - Descripción: Número de versión que quiere que se compare.
 - Dominio: Variable de tipo entero.
 - **comparewith:**
 - Descripción: Número de versión con el que se quiere comparar el atributo *compare*.
 - Dominio: Variable de tipo entero.

■ Operaciones:

- **print_header:** Esta operación devuelve el código HTML que imprime la parte superior de la página.
- **print_content:** Operación devuelve el código HTML que imprime la parte central de la página.
- **set_url:** Operación que modifica las variables globales para indicar a Moodle en que página nos encontramos.
- **set_comparasion:** Operación que asigna las versiones que se quieren comparar.
- **print_diff_content:** Operación que compara ambas versiones, encuentra sus diferencias y nos devuelve el código HTML para mostrarlas con el interfaz de Moodle.

■ Relaciones:

- **page_wikicode:** Hereda de esta clase.

■ Diagrama de la clase:

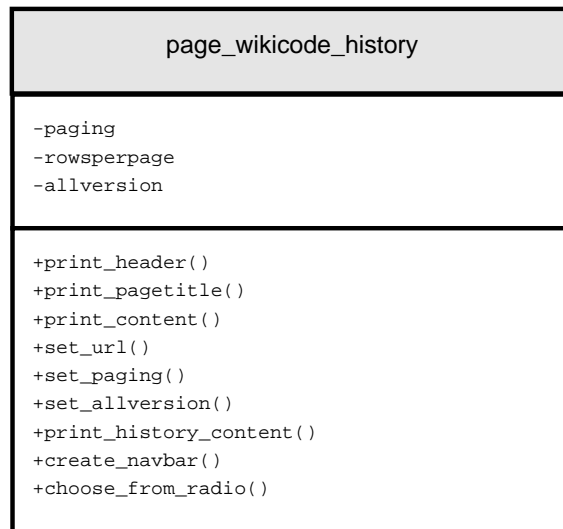


Figura 3.29: Class page_wikicode_diff

3.4.1.10. Class page_wikicode_history

- Descripción: Clase que nos muestra el histórico de versiones de un módulo Wikicode.
- Observaciones: Se considera que esta clase es la que hace una parte de interfaz con el caso de uso *Consulta de histórico y restauración de versiones*.
- Atributos:
 - paging:

- Descripción: Versión actual.
 - Dominio: Variable de tipo entero.
- **rowspage:**
 - Descripción: Versiones que se desean visualizar por página.
 - Dominio: Variable de tipo entero.
- **allversion:**
 - Descripción: Si esta variable es distinta a 0, todas las versiones serán visualizadas en una tabla.
 - Dominio: Variable de tipo entero.
- Operaciones:
 - **print_header:** Esta operación devuelve el código HTML que imprime la parte superior de la página.
 - **print_pagetitle:** Método para imprimir el título de la página.
 - **print_content:** Operación devuelve el código HTML que imprime la parte central de la página.
 - **set_url:** Operación que modifica las variables globales para indicar a Moodle en que página nos encontramos.
 - **set_paging:** Operación que asigna la versión actual sobre la que nos encontramos.
 - **set_allversion:** Operación que asigna si deseamos ver todas las versiones o un número limitado de estas.
 - **print_history_content:** Operación que nos muestra el contenido de un histórico.
 - **create_navbar:** Esta operación devuelve el código HTML de una barra de navegación.
 - **choose_from_radio:** Devuelve el código HTML tras crear un grupo de radio-button con las opciones indicadas.
- Relaciones:
 - **page_wikicode:** Hereda de esta clase.
- Diagrama de la clase:

Figura 3.30: Class `page_wikicode_history`

3.4.1.11. Class `page_wikicode_restoreversion`

- Descripción: Clase que restaura una antigua versión de una Wikicode.
- Observaciones: Se considera que esta clase es la que hace una parte del interfaz con el caso de uso *Consulta de histórico y restauración de versiones*.
- Atributos:
 - version:
 - Descripción: Identificador de versión que será restaurada.
 - Dominio: Variable de tipo entero.
- Operaciones:
 - **print_header:** Esta operación devuelve el código HTML que imprime la parte superior de la página.
 - **print_content:** Operación devuelve el código HTML que imprime la parte central de la página.
 - **set_url:** Operación que modifica las variables globales para indicar a Moodle en que página nos encontramos.
 - **set_versionid:** Operación que asigna la versión que va a ser restaurada.
 - **print_restoreversion:** Imprime el contenido de la versión que se va a restaurar.
- Relaciones:

- **page_wikicode:** Hereda de esta clase.

- Diagrama de la clase:



Figura 3.31: Class `page_wikicode_restoreversion`

3.4.1.12. Class `page_wikicode_save`

- Descripción: Clase que llama salva el código fuente en el histórico.
- Observaciones: Se considera que esta clase es la que hace de interfaz con el caso de uso *Salvar Código*.
- Atributos:
 - **newcontent:**
 - Descripción: Código fuente sincronizado del editor que va a ser salvado.
 - Dominio: Variable de tipo string.
- Operaciones:
 - **print_header:** Esta operación devuelve el código HTML que imprime la parte superior de la página.
 - **print_content:** Operación devuelve el código HTML que imprime la parte central de la página.
 - **set_newcontent:** Operación que asigna el código fuente al atributo de la clase.
 - **print_save:** Operación que salva en el histórico una nueva versión con el código fuente actual.
- Relaciones:
 - **page_wikicode_edit:** Hereda de esta clase.

- **page_wikicode_confirmrestore:** Padre de esta clase.

- Diagrama de la clase:

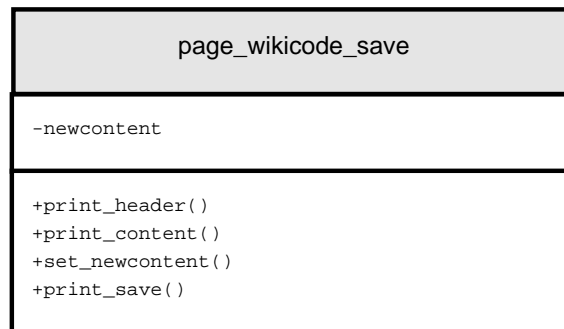


Figura 3.32: Class `page_wikicode_save`

3.4.1.13. Class `page_wikicode_viewversion`

- Descripción: Clase que nos muestra una antigua versión de una Wikicode.
- Observaciones: Se considera que esta clase es la que hace una parte del interfaz con el caso de uso *Consulta de histórico y restauración de versiones*.
- Atributos:
 - **version:**
 - Descripción: Identificador de versión que será visualizada.
 - Dominio: Variable de tipo entero.
- Operaciones:
 - **print_header:** Esta operación devuelve el código HTML que imprime la parte superior de la página.
 - **print_content:** Operación devuelve el código HTML que imprime la parte central de la página.
 - **set_url:** Operación que modifica las variables globales para indicar a Moodle en que página nos encontramos.
 - **set_versionid:** Operación que asigna la versión que va a ser restaurada.
 - **print_version_view:** Imprime el contenido de una versión seleccionada.
- Relaciones:

- **page_wikicode:** Hereda de esta clase.

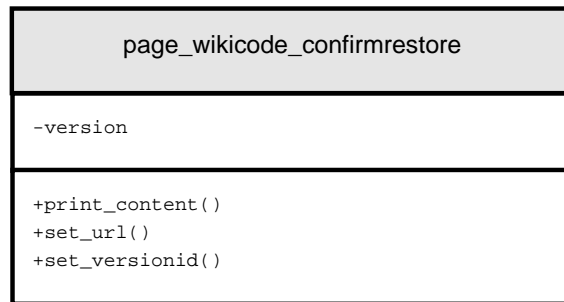
- Diagrama de la clase:



Figura 3.33: Class page_wikicode_viewversion

3.4.1.14. Class page_wikicode_confirmrestore

- Descripción: Clase que nos pide confirmación sobre la restauración de una versión anterior y en caso de respuesta afirmativa la restaura.
- Observaciones: Se considera que esta clase es la que hace una parte del interfaz con el caso de uso *Consulta de histórico y restauración de versiones*.
- Atributos:
 - version:
 - Descripción: Identificador de versión que será visualizada para la restauración.
 - Dominio: Variable de tipo entero.
- Operaciones:
 - **print_content:** Operación devuelve el código HTML que imprime la parte central de la página.
 - **set_url:** Operación que modifica las variables globales para indicar a Moodle en que página nos encontramos.
 - **set_versionid:** Operación que asigna la versión que va a ser restaurada.
- Relaciones:
 - **page_wikicode_save:** Hereda de esta clase.
- Diagrama de la clase:

Figura 3.34: Class `page_wikicode_confirmrestore`

3.4.1.15. Class `page_wikicode_admin`

- Descripción: Clase que nos muestra opciones de administración sobre nuestra Wikicode.
- Observaciones: Para visualizar la página ofrecida por esta clase es necesario tener rol de tutor en el curso.
- Atributos:
 - `action`:
 - Descripción: Información sobre la acción que va a realizar el administrador.
 - Dominio: Variable de tipo string.
- Operaciones:
 - **`print_header`**: Esta operación devuelve el código HTML que imprime la parte superior de la página.
 - **`print_content`**: Operación que realiza la acción requerida por el administrador.
 - **`print_delete_content`**: Operación que elimina el contenido de una Wikicode.
 - **`print_delete_version`**: Operación que elimina una versión de Wikicode.
- Relaciones:
 - **`page_wikicode`**: Hereda de esta clase.
- Diagrama de la clase:

3.4.1.16. Class `html_table`

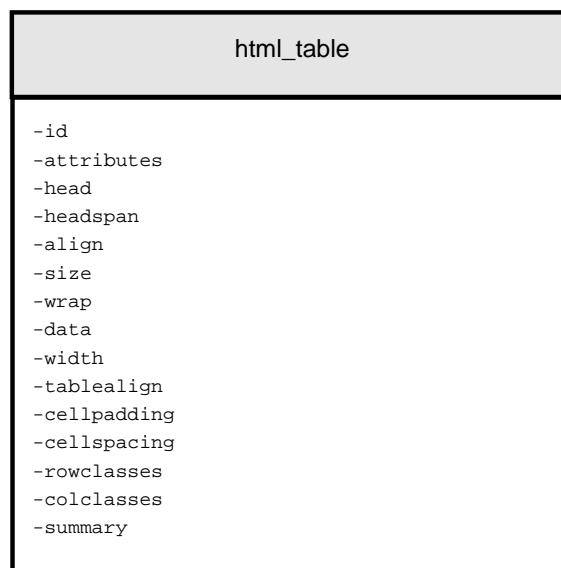
- Descripción: Clase que crea una tabla HTML.

Figura 3.35: Class `page_wikicode_admin`

- Observaciones: Esta clase sirve de esqueleto para diseñar una tabla HTML.
- Atributos:
 - `id`:
 - Descripción: Información sobre el atributo ID de la tabla.
 - Dominio: Variable de tipo string.
 - `attributes`:
 - Descripción: Información sobre los atributos HTML de la tabla.
 - Dominio: Variable de tipo array.
 - `head`:
 - Descripción: Objeto con información sobre la cabecera de la tabla.
 - Dominio: Variable de tipo array.
 - `headspan`:
 - Descripción: Objeto con información sobre la cabecera de la tabla que puede ser utilizado para múltiples columnas.
 - Dominio: Variable de tipo array.
 - `align`:
 - Descripción: Objeto con información sobre la alineación de la tabla.
 - Dominio: Variable de tipo array.
 - `size`:
 - Descripción: Objeto con información sobre el tamaño de la tabla.
 - Dominio: Variable de tipo array.

- wrap:
 - Descripción: Información sobre la propiedad wrap de la tabla.
 - Dominio: Variable de tipo array.
- data:
 - Descripción: Información sobre los datos de la tabla.
 - Dominio: Variable de tipo array.
- width:
 - Descripción: Información sobre el tamaño de la tabla. Obsoleto a partir de la versión 2 de Moodle que utiliza hojas de estilo CSS.
 - Dominio: Variable de tipo string.
- tablealign:
 - Descripción: Información sobre la alineación de la tabla. Obsoleto a partir de la versión 2 de Moodle que utiliza hojas de estilo CSS.
 - Dominio: Variable de tipo string.
- cellpadding:
 - Descripción: Información sobre el margen de las celdas de la tabla. Obsoleto a partir de la versión 2 de Moodle que utiliza hojas de estilo CSS.
 - Dominio: Variable de tipo entero.
- cellspacing:
 - Descripción: Información sobre el espacio entre celdas de la tabla. Obsoleto a partir de la versión 2 de Moodle que utiliza hojas de estilo CSS.
 - Dominio: Variable de tipo entero.
- rowclasses:
 - Descripción: Información sobre las clases a las que pertenece cada celda de la tabla.
 - Dominio: Variable de tipo array.
- colclasses:
 - Descripción: Información sobre las clases a las que pertenece cada columna de la tabla.
 - Dominio: Variable de tipo array.

- summary:
 - Descripción: Descripción de la tabla.
 - Dominio: Variable de tipo string.
- Operaciones: -
- Relaciones: -
- Diagrama de la clase:

Figura 3.36: Class `html_table`

3.4.1.17. Class `html_write`

- Descripción: Clase que nos devuelve el código HTML de distintos objetos web.
- Observaciones: Esta clase está disponible en el propio Moodle a partir de la versión 2, sin embargo se ha adaptado parcialmente y distribuido para la versión 1 de Moodle.
- Atributos: -
- Operaciones:
 - **tag**: Devuelve el código HTML de un campo tipo tag.
 - **start_tag**: Devuelve la apertura del código HTML de un campo tipo tag.
 - **end_tag**: Devuelve el cerrado del código HTML de un campo tipo tag.

- **checkbox:** Devuelve el código HTML de un campo tipo checkbox.
 - **table:** Devuelve el código HTML renderizado de un campo tipo table. Hace uso de la clase *html_table* como parámetro de entrada.
 - **label:** Devuelve el código HTML de un campo tipo label.
- Relaciones: -
 - Diagrama de la clase:

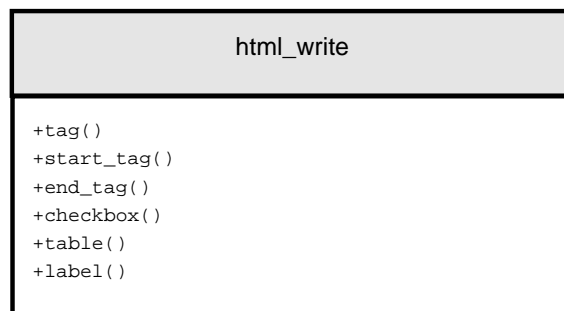


Figura 3.37: Class html_write

3.4.1.18. Class html_table_cell

- Descripción: Clase que define una celda de una tabla HTML.
- Observaciones: Esta clase está disponible en el propio Moodle a partir de la versión 2, sin embargo se ha adaptado parcialmente y distribuido para la versión 1 de Moodle.
- Atributos:
 - id:
 - Descripción: Información sobre el atributo ID de la celda.
 - Dominio: Variable de tipo string.
 - text:
 - Descripción: Información sobre el contenido de la celda.
 - Dominio: Variable de tipo string.
 - abbr:
 - Descripción: Información sobre el contenido abreviado de la celda.
 - Dominio: Variable de tipo string.

- colspan:
 - Descripción: Información sobre el número de columnas de la celda.
 - Dominio: Variable de tipo string.
- rowspan:
 - Descripción: Información sobre el número de filas de la celda.
 - Dominio: Variable de tipo string.
- scope:
 - Descripción: Define la forma de asociar la cabecera de las celdas y sus datos en una tabla.
 - Dominio: Variable de tipo string.
- header:
 - Descripción: Información sobre si tiene cabecera la celda.
 - Dominio: Variable de tipo booleana.
- style:
 - Descripción: Información sobre el contenido del atributo valor style de la celda.
 - Dominio: Variable de tipo string.
- attributes:
 - Descripción: Información sobre el contenido de otros atributos de la celda.
 - Dominio: Variable de tipo objeto.
- Operaciones: -
- Relaciones: Esta clase está únicamente creada para que pueda ser utilizada por la clase `html_row`.

- Diagrama de la clase:

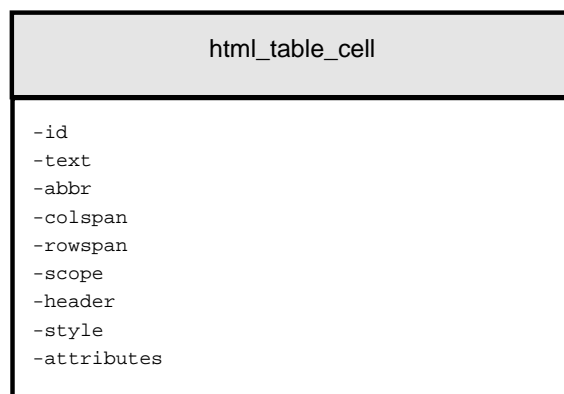
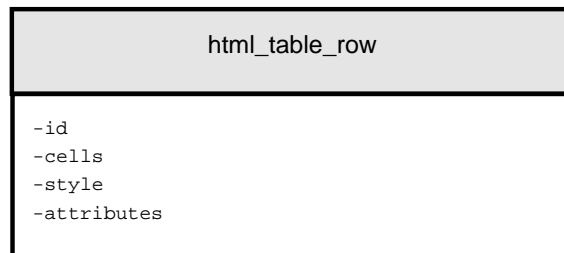


Figura 3.38: Class `html_cell`

3.4.1.19. Class `html_table_row`

- Descripción: Clase que define una fila de una tabla HTML.
- Observaciones: Esta clase está disponible en el propio Moodle a partir de la versión 2, sin embargo se ha adaptado parcialmente y distribuido para la versión 1 de Moodle.
- Atributos:
 - `id`:
 - Descripción: Información sobre el atributo ID de la celda.
 - Dominio: Variable de tipo string.
 - `cells`:
 - Descripción: Vector con información sobre las celdas de la fila.
 - Dominio: Variable de tipo objeto.
 - `style`:
 - Descripción: Información sobre el contenido del atributo valor `style` de la celda.
 - Dominio: Variable de tipo string.
 - `attributes`:
 - Descripción: Información sobre el contenido de otros atributos de la celda.
 - Dominio: Variable de tipo objeto.

- Operaciones: -
- Relaciones: Esta clase está únicamente creada para que pueda ser utilizada por la clase `html_table`.
- Diagrama de la clase:

Figura 3.39: Class `html_row`

3.4.2. Diccionario de datos para el sistema Wikicode-Synchronizer

Todas las funcionalidades de este subsistema han sido integrados en una única clase almacenada en un fichero PHP. Su descripción es la siguiente:

3.4.2.1. Class `editorlib`

- Descripción: Clase que hace de interfaz entre el módulo de Moodle y los datos para su sincronización.
- Observaciones: Esta clase puede ser exportada para trabajar bajo otro sistema e-learning.
- Atributos:
 - `editorCode`:
 - Descripción: Código fuente que se quiere sincronizar.
 - Dominio: Variable de tipo string.
 - `position`:
 - Descripción: Información sobre la posición de la tecla que ha sido pulsada para llamar al sincronizador.
 - Dominio: Variable de tipo entero.
 - `pageid`:

- Descripción: Información del identificador del código fuente.
 - Dominio: Variable de tipo entero.
- del:
 - Descripción: Información sobre si la operación realizada en el editor es de borrado.
 - Dominio: Variable de tipo booleana.
- char_del:
 - Descripción: Código decimal del carácter que ha sido eliminado del código fuente.
 - Dominio: Variable de tipo entero.
- Operaciones:
 - **deleteEnterRepeat:** Elimina la aparición de varios saltos de línea repetidos.
 - **getFunction:** Devuelve el nombre de la función al que pertenece una determinada posición dentro de un código fuente.
 - **getLevel:** Devuelve el nivel de indexación en una determinada posición de un cursor dentro de un código fuente.
 - **isLocked:** Información sobre si una determinada función está bloqueada dentro de un código fuente.
 - **isLockedByUser:** Información sobre si una determinada función está bloqueada dentro de un código fuente por el usuario que hace la llamada al sincronizador.
 - **lockFunction:** Operación que bloquea una determinada función dentro de un código fuente.
 - **unlockFunction:** Operación que desbloquea una determinada función dentro de un código fuente.
 - **getLocksByUser:** Operación que devuelve el nombre de las funciones que tiene bloqueada un usuario dentro de un código fuente.
 - **getInitFunctionLock:** Operación que devuelve la posición inicial del bloqueo de una función dentro de un código fuente.
 - **getEndFunctionLock:** Operación que devuelve la posición final del bloqueo de una función dentro de un código fuente.
 - **getFunctionLock:** Operación que devuelve una cadena con la función bloqueada delimitada por sus tags.

- **changeFunctionLock:** Operación que modifica una función bloqueada por otra dentro de un determinado código fuente.
 - **getCodigoFromDB:** Operación que devuelve el código fuente almacenado en base de datos.
 - **saveCodigoToDB:** Operación que salva el código fuente a la base de datos.
 - **keypress:** Trigger lanzado cuando una tecla clave es pulsada en el editor asociado.
 - **removePairKey:** Trigger lanzado cuando el usuario intenta eliminar una parte necesaria del código.
 - **unlockFunctionDB:** Procedimiento lanzado cuando un usuario quiere desbloquear una serie de funciones que tiene bloqueada.
- Relaciones: -
 - Diagrama de la clase:



Figura 3.40: Class editorlib

3.4.3. Diagrama de clases

El diagrama de clases es el siguiente:

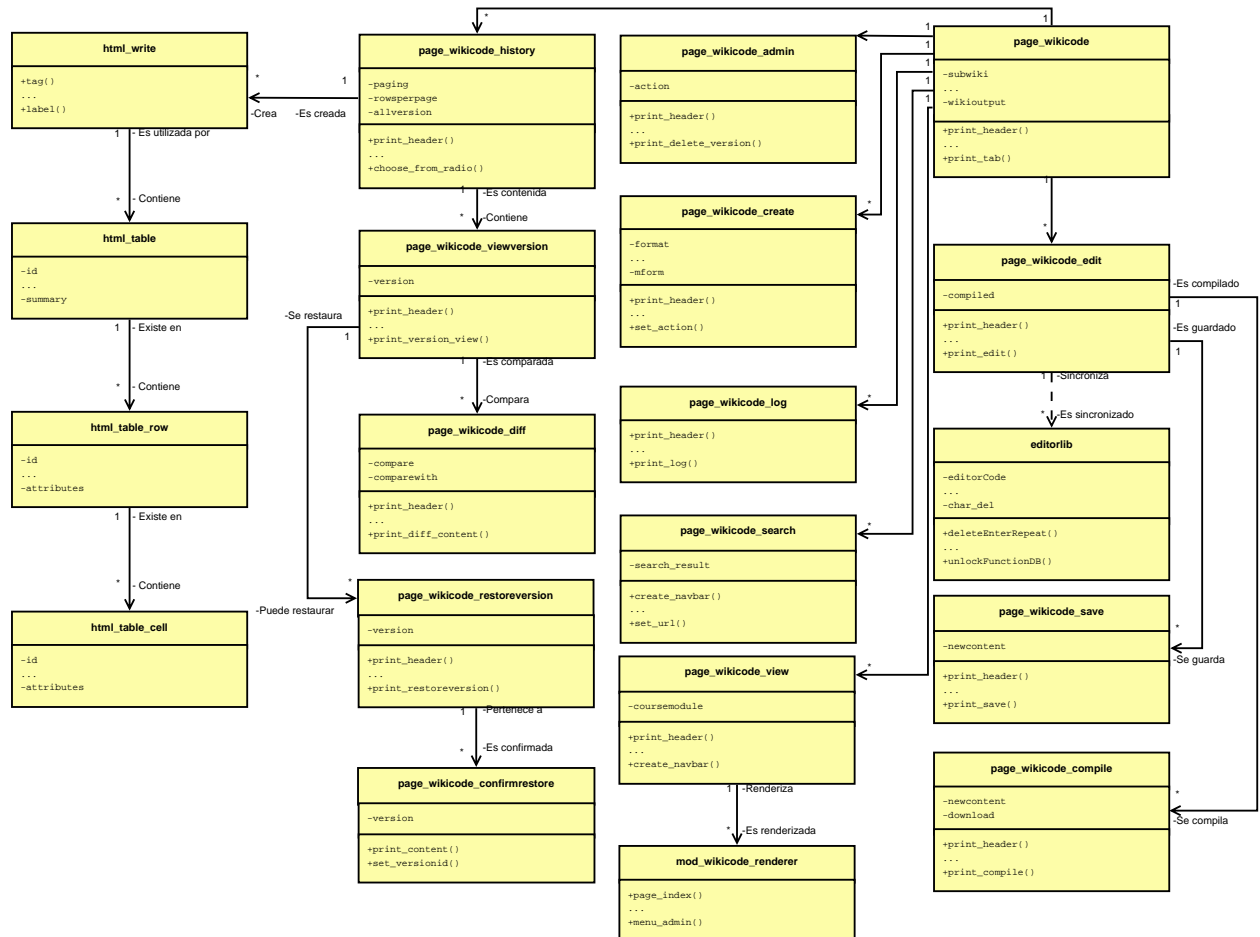


Figura 3.41: Diagrama de clases

3.5. Especificación del Modelo Objeto-Comportamiento

Una vez representados los elementos estáticos del sistema ahora es el momento de hacer una transición a su comportamiento dinámico. El aspecto dinámico del sistema combina la parte estructural con la dimensión del tiempo.

El modelo objeto-comportamiento representa cómo interaccionan las clases, especificando un flujo de control (secuencia de mensajes entre objetos) a lo largo del tiempo, para la ejecución de una tarea.

UML (*Unified Modeling Language*) 2.0 proporciona dos tipos de diagramas de interacción: los diagramas de secuencia y los diagramas de colaboración. Se ha decidido usar diagramas de colaboración porque en ellos se destaca la organización estructural de los objetos que envían y reciben mensajes. En los apartados siguientes se desarrollan los diagramas de interacción más significativos dentro de la funcionalidad requerida.

3.5.1. Diagrama de colaboración: Configuración de la Wikicode

En este diagrama se pretende mostrar el flujo de eventos relacionado con la configuración de una nueva Wikicode.

Flujo de sucesos:

1. Envía la configuración de parámetros.
2. Confirma que la configuración es válida.

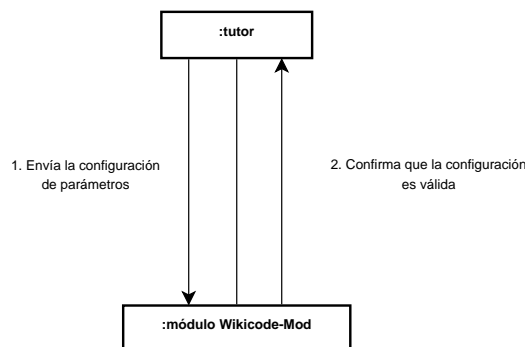


Figura 3.42: Diagrama de colaboración Configuración de la Wikicode

3.5.2. Diagrama de colaboración: Visualización del Código Fuente

En este diagrama se pretende mostrar el flujo de eventos relacionado con la visualización del código fuente dentro de Wikicode.

Flujo de sucesos:

1. Solicita la visualización del código fuente.
2. Se muestra el código.

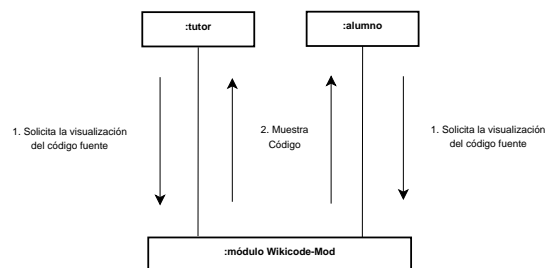


Figura 3.43: Diagrama de colaboración Visualización del Código Fuente

3.5.3. Diagrama de colaboración: Compilación de Código Fuente

En este diagrama se pretende mostrar el flujo de eventos relacionado con la compilación de código fuente dentro de Wikicode.

Flujo de sucesos:

1. Solicita la compilación del código fuente.
2. Creación de los ficheros necesarios y petición de compilación al compilador.
3. Se recoge la información de la compilación y se almacena.
4. Se envía la información sobre la compilación.
5. Se muestra la información sobre la compilación.

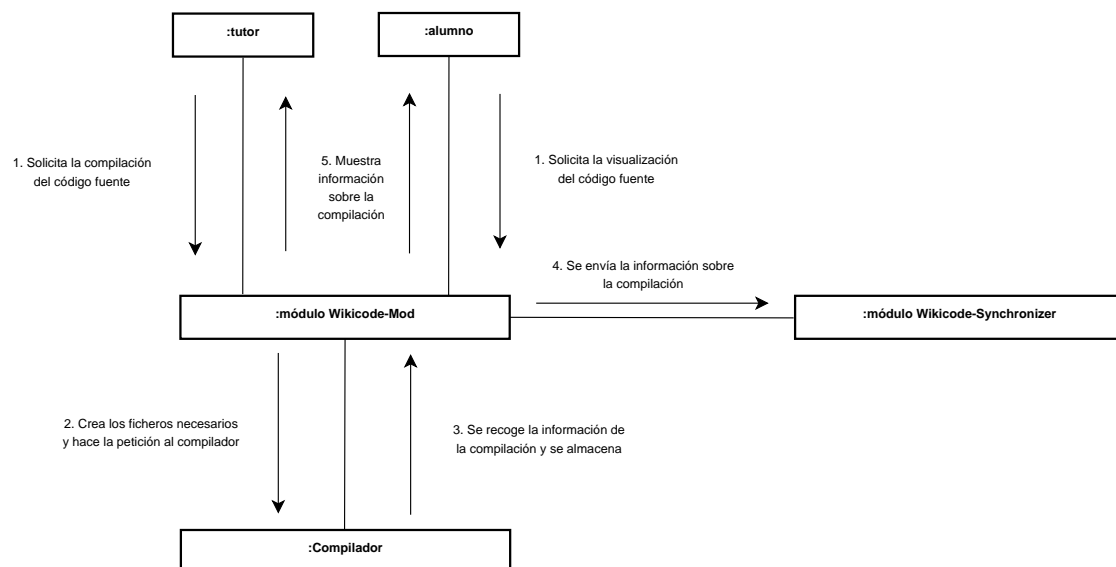


Figura 3.44: Diagrama de colaboración Compilación de Código Fuente

3.5.4. Diagrama de colaboración: Edición de código

En este diagrama se pretende mostrar el flujo de eventos relacionado con la edición de código dentro de Wikicode.

Flujo de sucesos:

1. Se edita el código y se pulsa una tecla clave.

2. Se envía el nuevo código y la posición del carácter que ha sido modificado.
3. Se sincroniza el código, se almacena en base de datos y se envía el nuevo código fuente.
4. Se muestra el nuevo código sincronizado.

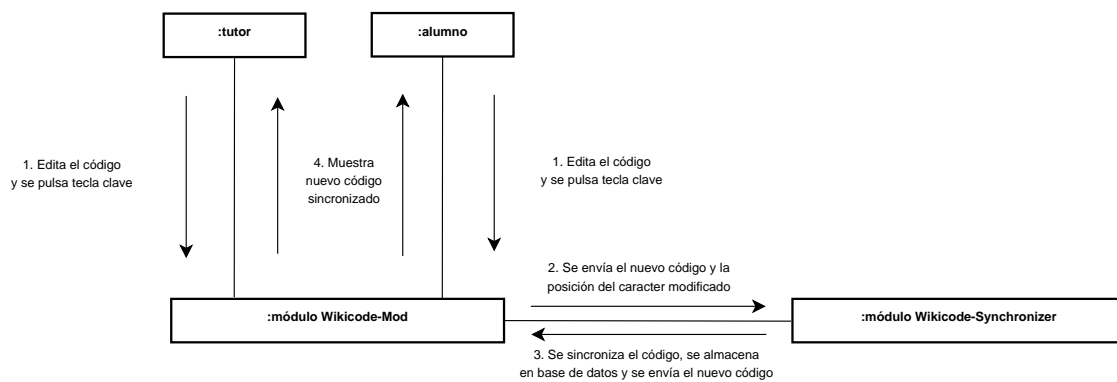


Figura 3.45: Diagrama de colaboración Edición de código

Capítulo 4

Diseño del sistema

4.1. Diseño de la base de datos

4.1.1. Modelo Conceptual

Se trata de representar la información de las distintas Wikicodes que puede almacenar un sistema e-learning, para que el sistema pueda acceder a la información referente al código fuente que están modificando los usuarios, así como para posibilitar una mejor consulta de estadísticas, histórico. También nos servirá para almacenar la información respecto a la compilación del código.

4.1.1.1. Análisis de los tipos de entidad

En principio, del análisis de requisitos del capítulo anterior pueden ser extraídos los siguientes tipos de entidad, como se muestra en la figura 4.1:

Tipo de entidad *wikicode*: el cual representa el módulo que se ha creado durante este proyecto y que se puede definir como “*el entorno de programación colaborativo*”. Se van a considerar como atributos de esta entidad un identificador secuencial que nos proporcione una clave para poder hacer referencia a él. También necesitaremos la información que se corresponde al curso en el que ha querido ser creado.

Posteriormente necesitaremos una serie de atributos informativos como es la hora de creación, de modificación, nombre, lenguaje de la Wikicode y tipo de Wikicode.

También contará con los atributos correspondiente a la parametrización del compilador, donde dividiremos en el que se querrá usar para Windows y el que se querrá usar para entornos Unix.

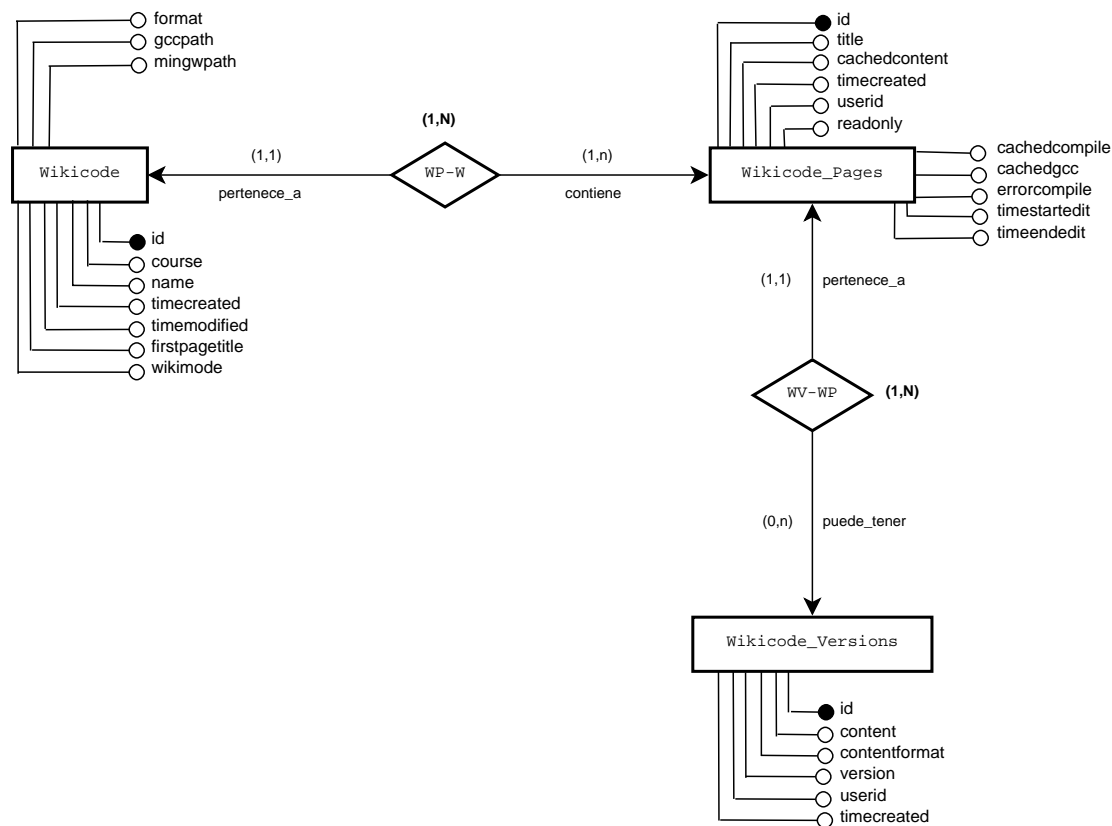


Figura 4.1: Esquema EE-R del modelo conceptual del módulo Wikicode

Tipo de entidad *wikicode_pages*: el cual representa al objeto del mundo real *página* que representa “*el código fuente editado de modo colaborativo en lenguaje C*”. Se van a considerar como atributos el título que se haya considerado al crearse así como el contenido del código. También será necesario una serie de información estadística y de información como es saber el identificador del usuario que lo ha creado, el número de visualizaciones que ha tenido ese código y si sólo queremos que sea de lectura.

También es importante contar con la información del compilador, por lo que se almacenará el código que se ha enviado al compilador y la salida que ha producido este al enviarle dicho código como parámetro. Por último también contaremos con la información estadística necesaria para la visualización de los logs: Hora de primera edición, Hora de última edición y Número de errores de compilación.

Tipo de entidad *wikicode_versions*: el cual representa el objeto del mundo real *version* que representa “*el código fuente colaborativo en lenguaje C en un momento específico del tiempo*”. Esta información es necesaria para mantener un histórico por lo que únicamente contaremos como atributos el código fuente de dicha versión, el tipo de

contenido, el número de versión que se trata, la hora en que fue creada dicha versión y la información referente al usuario que creó esta entrada.

4.1.1.2. Análisis de los tipos de interrelación

Los tipos de entidad antes mencionados se encuentran relacionados de la siguiente forma:

Tipo de interrelación *Wikicode/Wikicode_Pages* W-WP: el cual relaciona los tipos de entidad *Wikicode* y *Wikicode_Pages*, la interrelación es del tipo $1:N$, ya que una página pertenece a una única wikicode, por tanto, el tipo de entidad *Wikicode_Pages* participa con las cardinalidades $(1,1)$ en este tipo de interrelación, mientras que en una Wikicode pueden existir muchas páginas debido a la creación de grupos. Se considera que una Wikicode no tiene sentido sin que existan páginas en su interior, por lo que el tipo de entidad *Wikicode* participara en el tipo de interrelación *W-WP* con las cardinalidades $(1,n)$ como se puede observar en la figura 4.1.

Tipo de interrelación *Wikicode_Pages/Wikicode_Version* WP-WV: el cual relaciona los tipos de entidad *Wikicode_Pages* y *Wikicode_Versions*, la interrelación es del tipo $1:N$, ya que una versión del histórico pertenece a un único código fuente, por tanto, el tipo de entidad *Wikicode_Versions* participa con las cardinalidades $(1,1)$ en este tipo de interrelación, mientras que de un código fuente pueden existir muchas versiones en un histórico. No hay necesidad de que un código fuente tenga almacenada información en su histórico, por lo que el tipo de entidad *Wikicode_Pages* participara en el tipo de interrelación *WP-WV* con las cardinalidades $(0,n)$ como se puede observar en la figura 4.1.

4.1.2. Modelo Relacional

Una vez realizado el modelo conceptual se va a construir el modelo relacional del problema considerado.

4.1.2.1. Tabla Wikicode

La tabla *Wikicode* se forma a partir del tipo de entidad *Wikicode*. Esta tabla incluye los atributos *id*, *course*, *name*, *timecreated*, *timemodified*, *firstpagetitle*, *wikimode*, *format*, *gccpath* y *mingwpath* que son tomados del tipo de entidad antes mencionado (regla RTECAR-1).

La clave principal de la tabla es el atributo *id*, no considerándose, en este caso, ningún otro atributo que pueda ser utilizado como identificador alternativo, y ninguno de los atributos de esta tabla mantiene referencia alguna con los atributos de las otras tablas. La tabla *Wikicode* queda de la siguiente forma:

Wikicode (*id*, *course*, *name*, *timecreated*, *timemodified*, *firstpagetitle*, *wikimode*, *format*, *gccpath*, *mingwpath*)

4.1.2.2. Tabla Wikicode_Pages

La tabla *Wikicode_Pages* se forma a partir del tipo de entidad del mismo nombre tomando los atributos de este tipo de entidad (regla RTECAR-1), además de:

- El atributo identificador de la tabla *Wikicode* con la cual mantiene una relación uno a muchos (regla RTECAR-3.1).

La clave principal de la tabla es el atributo *id*, no considerándose, en este caso, ningún otro atributo que pueda ser utilizado como identificador alternativo.

Wikicode_Pages (*id*, *title*, *cachedcontent*, *timecreated*, *userid*, *readonly*, *cachedcompile*, *cachedgcc*, *errorcompile*, *timestartedit*, *timeendedit*, ***subwikiid***)

4.1.2.3. Tabla Wikicode_Versions

La tabla *Wikicode_Versions* se forma a partir del tipo de entidad del mismo nombre tomando los atributos de este tipo de entidad (regla RTECAR-1), además de:

- El atributo identificador de la tabla *Wikicode_Pages* con la cual mantiene una relación uno a muchos (regla RTECAR-3.1).

La clave principal de la tabla es el atributo *id*, no considerándose, en este caso, ningún otro atributo que pueda ser utilizado como identificador alternativo.

Wikicode_Pages (*id*, *content*, *contentformat*, *version*, *timecreated*, *pageid*)

4.1.3. Normalización del modelo

Table Wikicode: esta tabla se encuentra en FNBC, pues todos los atributos son atómicos (FN1) y las dependencias funcionales son completas.

Table Wikicode_Pages: al igual que en el caso anterior, esta tabla se encuentra en FNBC.

Table Wikicode_Versions: por el mismo razonamiento esta tabla se encuentra en FNBC.

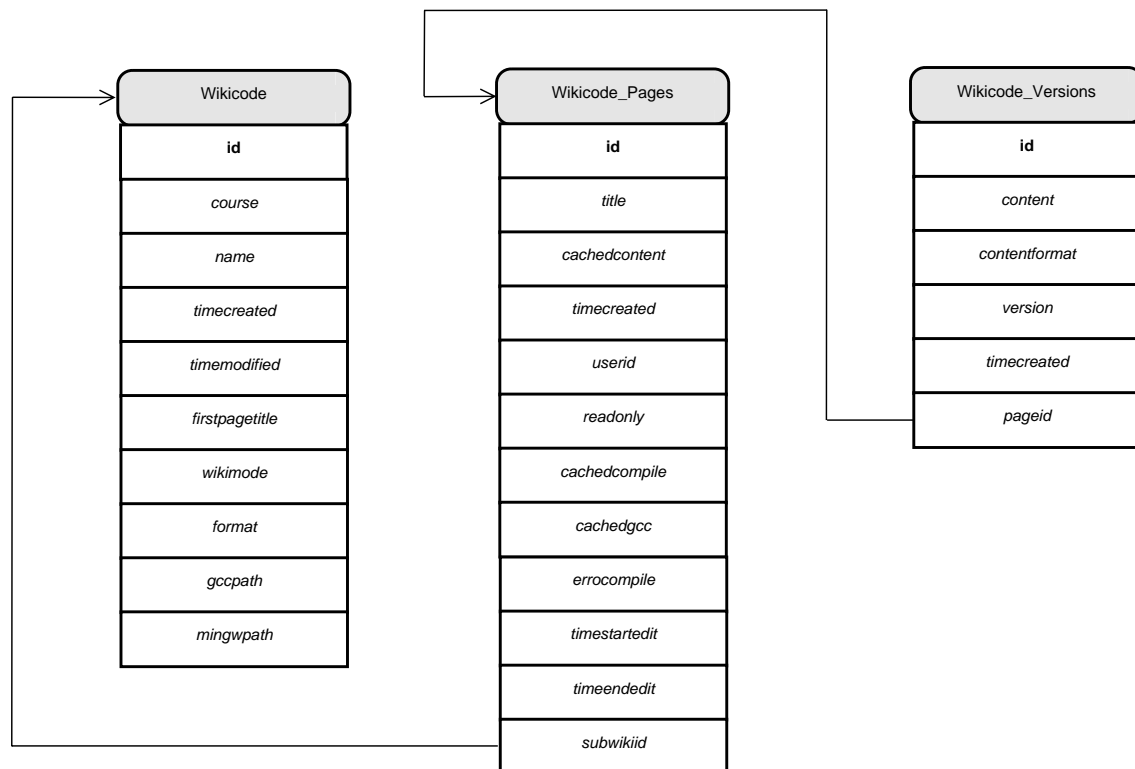


Figura 4.2: Diagrama relacional para el módulo Wikicode

4.2. Descripción del resultado final del interfaz

En este apartado se describirá cómo interactúa el sistema con el usuario, definiendo la estructura y comportamiento del interfaz. Aunque el diseño de esta interfaz no es uno de los objetivos primordiales del proyecto, se ha intentado realizar un diseño de manera que se facilite el manejo del sistema al usuario aprovechando las características del entorno Moodle. Además, para la parte en que el usuario va a editar el código fuente, se ha simulado un editor con características similares a cualquier otro editor de código que se pueda encontrar.

4.2.1. Descripción del interfaz

En cuanto a la estructura general del interfaz del módulo desarrollado, hay que tener en cuenta que nuestro proyecto se ha centrado en la realización de un módulo que pueda ser utilizado en Moodle.

De esta manera, el diseño de los módulos por defecto de Moodle se ha tomado como base para el desarrollo del nuevo bloque. Este hecho permite conseguir homogeneidad del sistema de manera que todos los bloques y elementos se encuentren en consonancia.

Por tanto, nos limitaremos a hablar del diseño realizado para el bloque en cuestión sin especificar características especiales de Moodle.

4.2.2. Descripción de las pantallas del interfaz

4.2.2.1. Pantalla principal

Al tratarse de un módulo la creación de una Wikicode y su posterior selección será igual a la otro cualquiera (Foro, Wiki, Chat, Cuestionario, etc.). En la siguiente figura se muestra como se añadiría una nueva Wikicode a un curso ya creado:

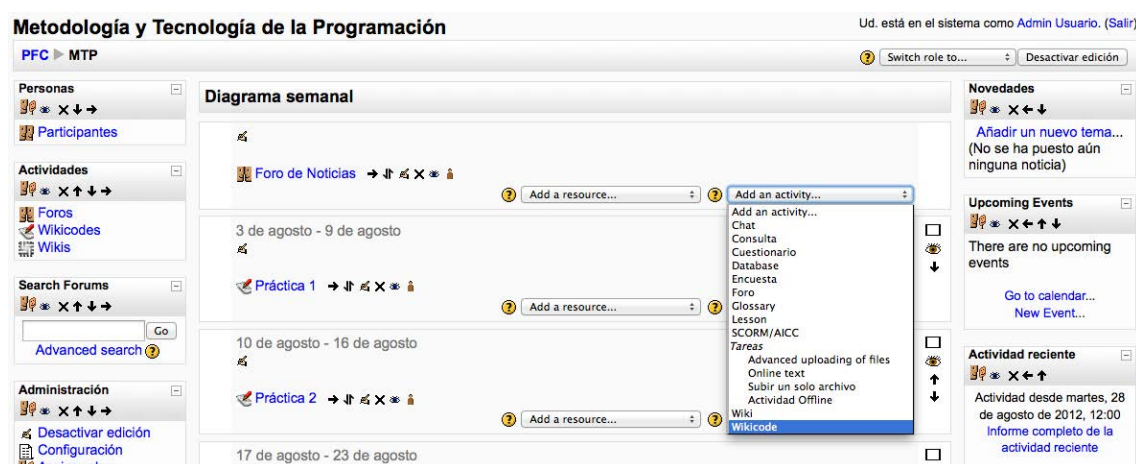


Figura 4.3: Pantalla principal de un curso Moodle añadiendo una Wikicode

4.2.2.2. Pantalla configuración y parametrización

Cuando se accede a la opción de crear una nueva Wikicode se accede a la pantalla de configuración del módulo. Este formulario contendrá todas las variables para la creación del entorno colaborativo deseado. En concreto las variables a tener en cuenta será la elección del nombre y la descripción del mismo, la posibilidad de agrupamiento con respecto a una Wikicode (eligiendo de una lista) y la parametrización del compilador que desea usarse.

Para ayudar al tutor, se ha optado por una ayuda automática a la hora de parametrizar los compiladores, tomando siempre por defecto el valor que haya sido usado un mayor número de veces con anterioridad. Si nunca se ha usado ninguno por defecto aparecerá el nombre del compilador recomendado.

En la siguiente figura se muestra la pantalla de configuración que se presenta cuando se decide crear una nueva Wikicode:

Figura 4.4: Pantalla de configuración y parametrización Wikicode

Además, a partir de la versión 2 de Moodle se ha actualizado el módulo para que cada opción traiga consigo una ayuda que nos explique la funcionalidad de cada variable.

Figura 4.5: Pantalla de ayuda a la configuración Wikicode

4.2.2.3. Pantalla cabecera superior

Como su propio nombre indica, no se trata de un interfaz completo en sí, sino de la parte superior que compartirán todas las páginas definidas dentro del módulo Wikicode. Esta cabecera, compartida, la podemos dividir en las siguientes partes.

- **Nombre del curso:** Etiqueta informativa sobre el nombre del curso al que pertenece la Wikicode.
- **Listado de navegación:** Menú de tipo lista que nos permite acceder al resto de contenidos del curso.
- **Árbol de directorio:** Nos representa un listado de donde nos encontramos ahora mismo, situándonos en el curso, la categoría y el nombre que hayamos dado al módulo. Será navegable y permite enlazar rápidamente a cada una de ellas.
- **Actualización:** Si el usuario tiene permisos de administración le aparecerá este botón, el cual permite acceder a la parametrización de la Wikicode que nos encontremos y modificar sus opciones.
- **Menú de contenido:** Menú de navegación que nos permite desplazarnos rápidamente por las opciones de nuestro editor de desarrollo. Se marcará la opción sobre la que nos encontremos en el momento.

En la siguiente figura se muestra la cabecera que se presentará en todas las opciones del módulo Wikicode:

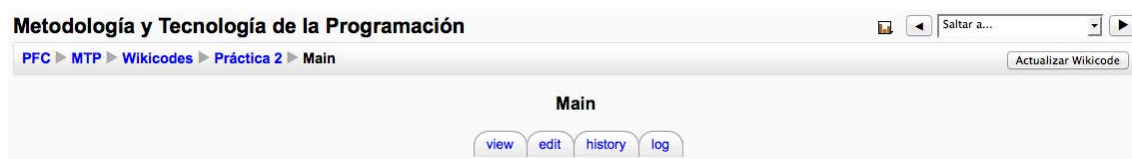


Figura 4.6: Pantalla cabecera Wikicode

4.2.2.4. Pantalla visualización

Se trata de la pantalla donde el usuario puede consultar, en formato texto plano, el código fuente de la Wikicode. La pantalla comparte el menú superior con el resto de opciones y simplemente nos mostrará esta información.

Sin embargo, para mayor utilidad de la misma, si el tutor ha decidido crear la Wikicode de modo grupal le aparecerá un menú donde podrá seleccionar el grupo que desee visualizar. Esta opción también será activa para los alumnos en caso de que el tutor así lo marcara.

En la siguiente figura se muestra el formulario de visualización con la opción de selección de grupos activada:

The screenshot shows the Moodle interface for viewing Wikicode. The page title is "Metodología y Tecnología de la Programación". The user is logged in as "Admin Usuario". The breadcrumb trail is: "Página Principal > Cursos > MTP > 25 de agosto - 31 de agosto > Práctica Documentación Grupal > View > Main > View". There is a search bar labeled "Search wikis".

On the left, there is a sidebar with "Navegación" and "Ajustes" sections. The "Ajustes" section is expanded, showing options like "Wikicode administration", "Editar ajustes", "Roles asignados localmente", "Permisos", "Compruebe los permisos", "Filtros", "Registros", "Copia de seguridad", and "Restaurar". Below this are links for "Administración del curso", "Cambiar rol a...", "Ajustes de mi perfil", and "Administración del sitio". At the bottom of the sidebar is a "Buscar" button.

The main content area has a title "Práctica de Grupo" and buttons for "View", "Edit", "History", "Log", and "Administration". Below this is a dropdown menu labeled "Grupos separados" with "Grupo A" selected. The "Main" section displays the Wikicode source code:

```
#include <stdio.h>

int main() {
    printf("Soy del grupo A y mi practica es mejor!");
    c = 1;
}

int exclusion() {
    printf("La he modificado yo!");
}
```

At the bottom of the main content area, there is a "Marcas:" label. The footer includes a link to "Moodle Docs para esta página", the user login information "Usted se ha identificado como Admin Usuario (Salir)", and a button labeled "MTP".

Figura 4.7: Pantalla visualización Wikicode

4.2.2.5. Pantalla edición

Se trata de la pantalla clave del entorno de desarrollo, en la cual podremos editar el código fuente, comunicarnos con el resto de usuarios y compilar nuestro código para ver que errores puede contener.

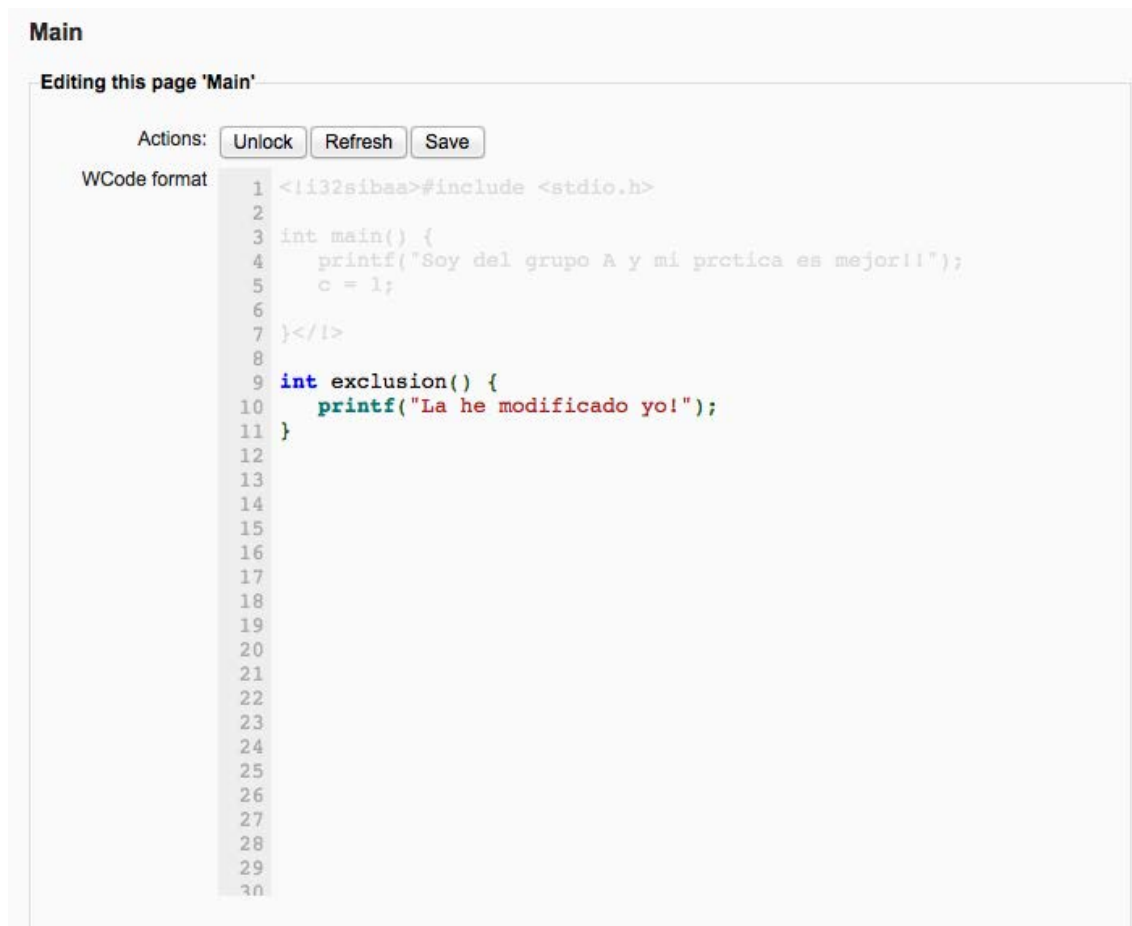


Figura 4.8: Pantalla editor Wikicode.

El editor de código se irá actualizando a medida que el usuario va escribiendo. Del mismo modo, si intentamos escribir en una parte del código bloqueada el editor nos impedirá esa acción. Las partes bloqueadas por otros usuarios están claramente diferenciadas en otro color y a su vez se nos informa que usuario es el dueño de dicha parte.

Para comodidad del usuario final, se da formato y color al código conforme a los estándares en C siguiendo este estilo:

Comentarios : Color marrón [#BB9977]

Palabra clave : Color azul y tipografía en negrita.

Cadena : Color burdeos [#AA2222]

Números : Color verde claro [#3A3]

Funciones : Color turquesa [#077] y tipografía en negrita.

Definición : Color verde oscuro.

Bloqueo : Color gris claro.

Los botones de la parte superior de este apartado actúan del siguiente modo:

Unlock : Nos abre una ventana de estilo pop-up que nos permite desbloquear las funciones que tengamos bloqueadas del código. Podemos elegir entre una, varias o todas.

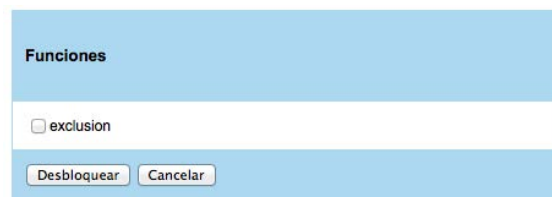


Figura 4.9: Pantalla desbloqueo Wikicode.

Refresh : Actualiza el código fuente. La actualización es semi-automática, pero si se requiere se puede efectuar esta operación manualmente.

Save : Guarda una copia del código fuente en el histórico y desbloquea todas las funciones.

Para comunicarnos con otros usuarios se ha creado un chat, el cual se dividirá en dos partes. La superior mostrará los mensajes que han sido enviados con anterioridad, informándonos del usuario y la hora a la que se envió este mensaje. En la inferior habrá un campo de texto y un botón de envío, los cuales servirán para añadir nueva información al chat.

El diseño de este elemento del formulario se puede contemplar en la siguiente figura:

Como última parte de este formulario vamos a diseñar la parte dedicada a la compilación, la cual estará dividida en dos partes:

- Mensaje: Es el resultado de nuestra compilación, en la que se informará con un mensaje si se ha podido compilar de modo correcto y se nos informará de los errores



Figura 4.10: Pantalla chat Wikicode.

en caso de que esta premisa no se cumpla. En ella se nos informará del error cometido y se nos proporcionará una pequeña ayuda para solucionarlo.

- Botones: Se tratarán de dos botones que nos permitirán dos acciones parejas. En la primera se llamará al compilador para obtener la información sobre nuestro código fuente. La segunda es una ampliación de la primera, la cual llamará al compilador y, si todo es correcto, nos permitirá descargar un ejecutable para ejecutarlo en el equipo cliente.

El diseño de este elemento se encuentra en la siguiente figura:



Figura 4.11: Pantalla compilación Wikicode.

4.2.2.6. Pantalla histórico

Como su propio nombre indica, se trata del interfaz que vamos a utilizar para consultar las versiones anteriores de un código fuente. Se proporcionará una lista de versiones y se creará una tabla informando de quién creó dicha versión y cuando la creó.



Figura 4.12: Pantalla histórico Wikicode.

Una vez pulsemos encima de una de estas versiones que se nos ofrece, podremos visualizar el código fuente que había en tal fecha y restaurarlo si así lo vemos oportuno pulsando el enlace a la izquierda de la pantalla. Dicho enlace nos llevará a otro formulario donde se nos pedirá autorización para realizar dicha acción.



Figura 4.13: Pantalla restauración histórico Wikicode.

Una de las mayores novedades que presenta el módulo para la versión 2 de Moodle es la inclusión de una potente herramienta que nos permite comparar dos códigos, señalando en el propio interfaz las diferencias entre ambos. El diseño de dicha herramienta se puede ver en la siguiente figura, y para llamarlo habrá que seleccionar dos versiones en la Figura 4.12 y pulsar el botón denominado “Comparar”.



Figura 4.14: Pantalla comparación histórico Wikicode.

4.2.2.7. Pantalla estadísticas

Este formulario, el último a diseñar, es el que se encargará de traernos la información estadística almacenada en base de datos con respecto al código fuente. Se accede a él pulsando el botón log del menú superior, y una vez dentro de él nos encontraremos una serie de etiquetas con un campo de texto informándonos de su resultado.

Como novedad, a partir de la versión 2 de Moodle se añaden unos botones de ayuda para explicar el significado de cada estadística. El diseño de dicho formulario se puede contemplar en la siguiente figura:



Figura 4.15: Pantalla estadística Wikicode.

4.3. Especificación procedimental

Llegados a este punto, trataremos de modelar el módulo viendo sus aspectos dinámicos desde un alto nivel de abstracción, es decir, nos centraremos en las actividades de negocio más importantes en el dominio del problema. Para ello no pretendemos describir en detalle los procedimientos u operaciones que tienen lugar entre dichos objetos ni los atributos que se intercambian entre sí sino simplemente mostrar las actividades que tienen lugar entre ellos. Para ello, en los siguientes apartados comentaremos los diferentes diagramas de actividades asociados a cada una de los bloques explicados anteriormente.

4.3.1. Diagrama de actividades para el sistema Wikicode-Mod

4.3.1.1. Actividad: Crear, configurar y parametrizar la Wikicode

Una de las principales actividades que debe realizar el *sistema Wikicode-Mod* es la creación de la actividad dentro de Moodle, así como configurarse correctamente para poder usarse en un futuro.

En la siguiente figura se muestra el diagrama de actividades asociado:

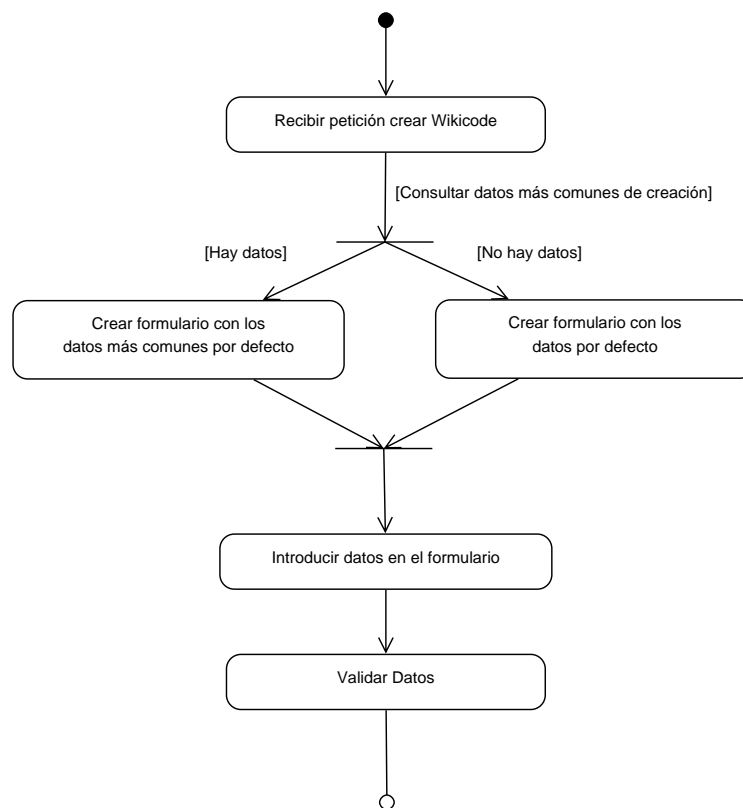


Figura 4.16: Actividad crear, configurar y parametrizar la Wikicode

4.3.1.2. Actividad: Visualizar código fuente

Es otra de las actividades que realiza el *sistema Wikicode-Mod*. Se trata de una actividad muy simple en la que se muestra el código fuente de la Wikicode.

El diagrama de actividad es:

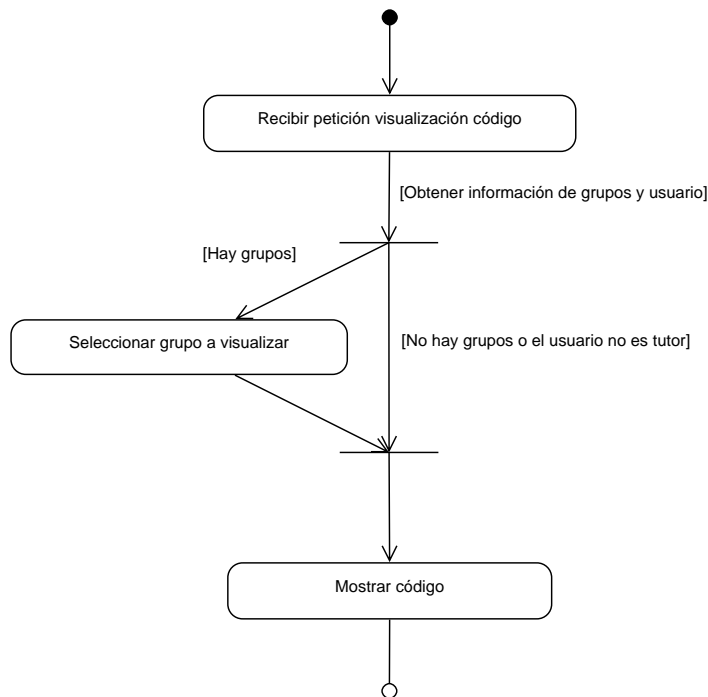


Figura 4.17: Actividad visualizar código fuente

4.3.1.3. Actividad: Editar código fuente

La actividad que nos concierne en este subapartado se trata de la principal labor del *sistema Wikicode-Mod*. Se trata de toda la actividad relacionada con la edición del código fuente, y cómo se almacena en base de datos de manera abstracta al usuario. Las teclas claves están definidas en la especificación de requisitos y la actividad de sincronización se explicará posteriormente.

El diagrama de actividad es:

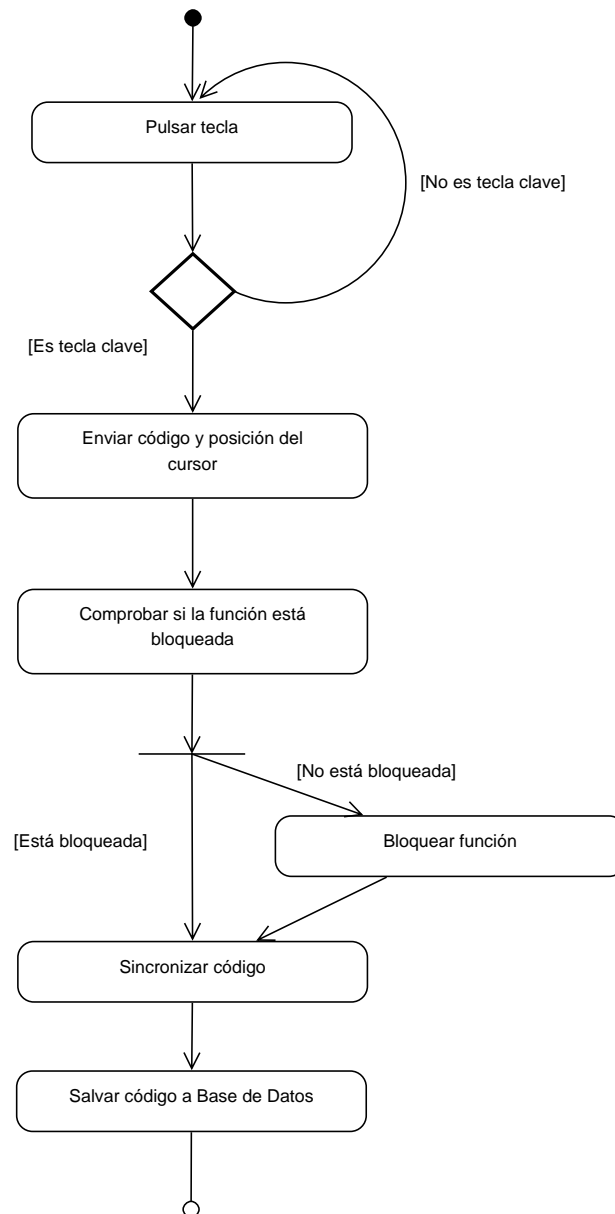


Figura 4.18: Actividad editar código fuente

4.3.1.4. Actividad: Chat

El chat es la siguiente actividad que vamos a describir, y es el módulo que utiliza el *sistema Wikicode-Mod* para comunicar a los usuarios entre sí mientras están modificando un código fuente. Es una actividad secuencial en la que únicamente hay que escribir y leer un fichero en un intervalo de tiempo.

En la siguiente figura podemos comprobar el diagrama de actividades asociado al envío de mensajes:

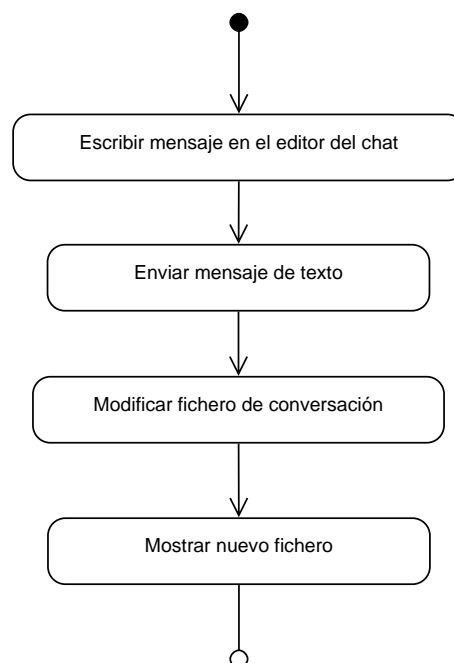


Figura 4.19: Actividad chat

4.3.1.5. Actividad: Compilar código fuente

La compilación es el proceso en el que un programa informático, conocido como compilador, traduce un programa escrito en un lenguaje de programación a otro que habitualmente suele ser el lenguaje máquina. En esta actividad el *sistema Wikicode-Mod* tomará el código escrito en el editor y se lo pasará al compilador. Posteriormente podremos comprobar la respuesta de este.

El diagrama de actividad es el siguiente:

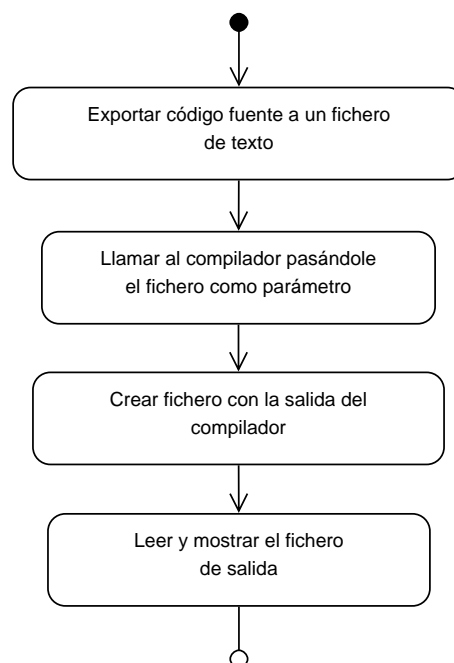


Figura 4.20: Actividad compilar código fuente

4.3.1.6. Actividad: Visualizar histórico

Es otra actividad que nos proporciona el *sistema Wikicode-Mod*, en la cual podemos ir viendo toda la historia de un código fuente. Esto nos servirá para, en caso de que hayamos cometido un error, poder hacer un rollback a una versión antigua almacenada. En el histórico no se eliminará ninguna entrada, sino que la que hayamos seleccionado en caso de restauración se creará como una nueva versión. Es decir, sería similar a si hubiéramos copiado todo ese antiguo código y hubiéramos salvado.

Una vez restaurado, volveremos al editor y se podrá volver a editar.

El diagrama de actividad que nos proporciona esta actividad se muestra en la siguiente figura:

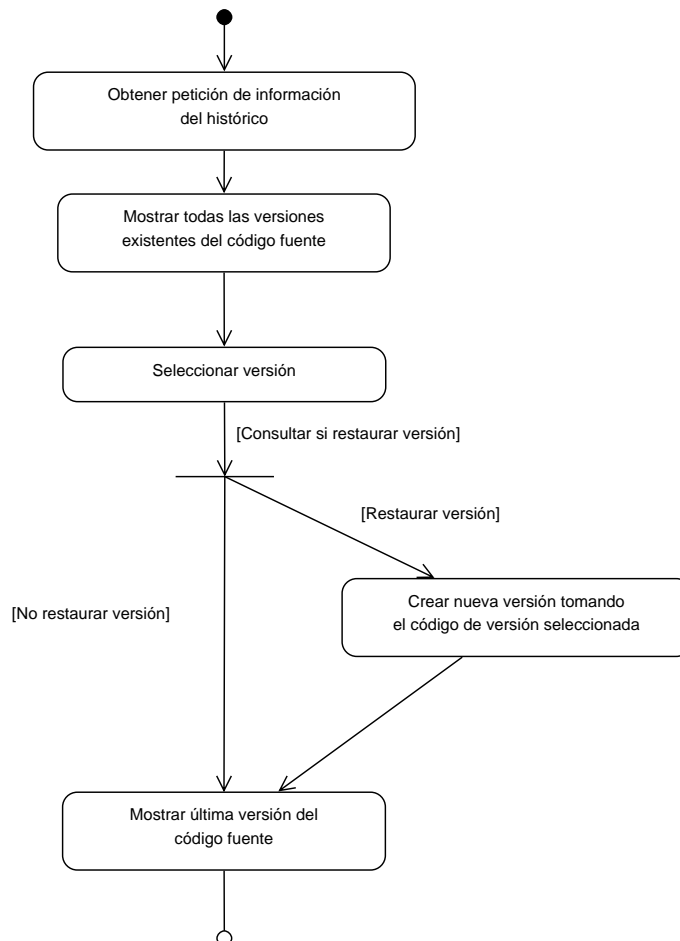


Figura 4.21: Actividad visualizar histórico

4.3.1.7. Actividad: Visualizar estadísticas

Es otra actividad del *sistema Wikicode-Mod* puede ser una de las más utilizadas por los tutores para poder valorar la realización de prácticas de programación. En ella se puede consultar todo lo referente en torno a la programación de una actividad *Wikicode* y se puede ampliar en un futuro con tantas estadísticas como se desee sin necesidad de variar el diagrama de actividad.

El diagrama de actividad se muestra en la siguiente figura:

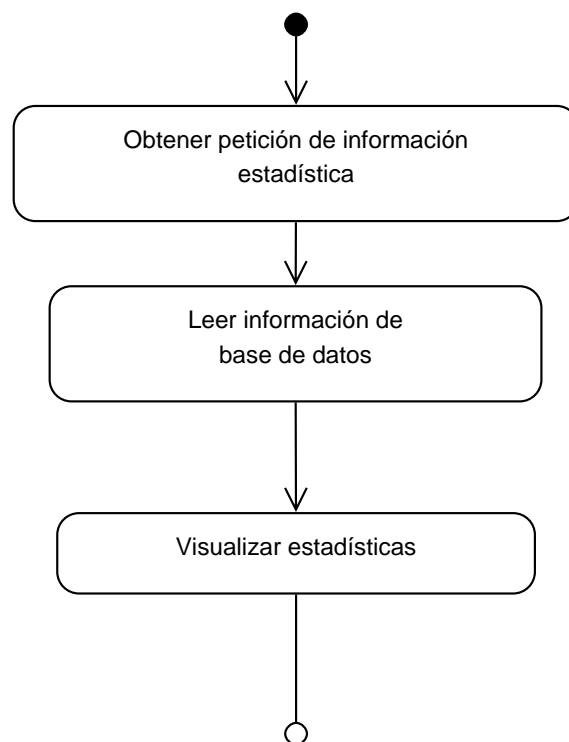


Figura 4.22: Actividad visualizar estadísticas

4.3.2. Diagrama de actividades para el sistema Wikicode-Synchronizer

4.3.2.1. Actividad: Sincronizar código

La sincronización de código es el procedimiento en el que se toma las variaciones del código fuente de un usuario y se pone en consonancia con el resto de usuarios. Del mismo modo, en ese mismo proceso se conseguirá tomar las variaciones de los otros usuarios para poder mostrárselo al usuario que está modificando su código. Así, se tratará de un proceso para poder ir viendo en tiempo real todas las modificaciones que va habiendo en un código sin necesidad de pisar el trabajo entre sí de los usuarios.

En la siguiente figura podemos comprobar el diagrama de actividades asociado a la sincronización del código fuente de un usuario con los demás:

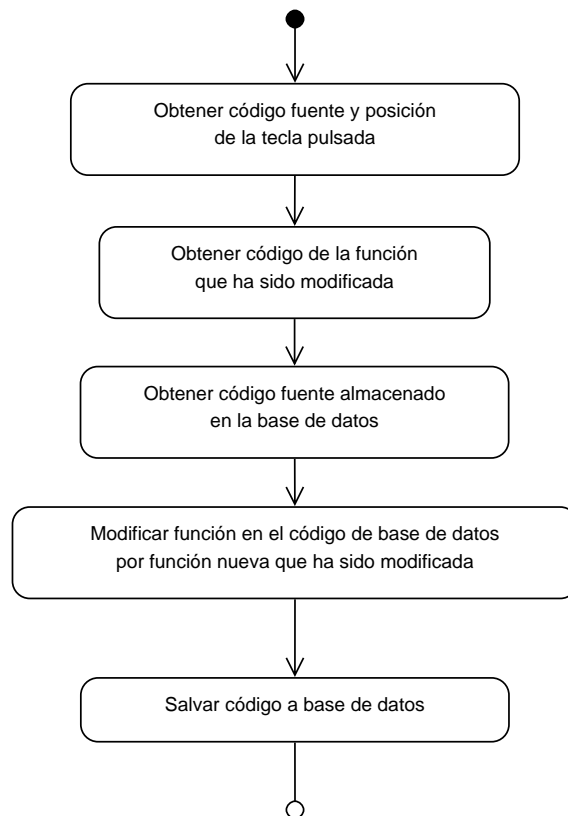


Figura 4.23: Actividad sincronizar código

4.3.2.2. Actividad: Actualizar estadísticas

Como se ha mostrado en la Figura 4.22, el usuario tendrá la posibilidad de consultar las estadísticas sobre su página de código. Estas estadísticas no son calculadas a la hora de realizar la consulta, sino que se van almacenando y actualizando en base de datos conforme se van produciendo, de este modo la carga de la operación de consulta es mucho menor y las operaciones únicamente son necesarias de realizar una vez.

En el siguiente diagrama se muestra como se van calculando dichas estadísticas y almacenando en base de datos:

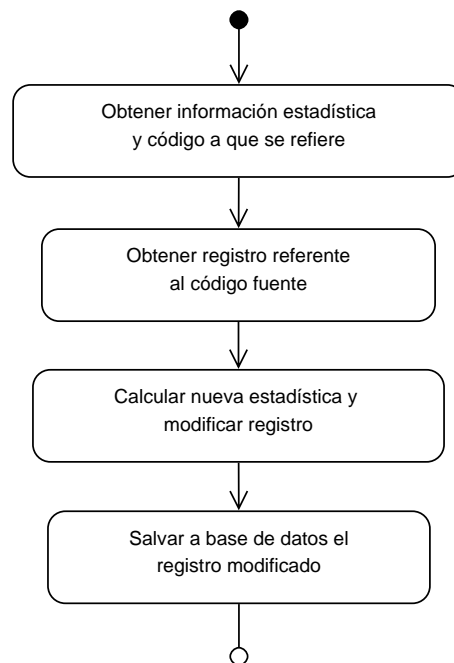


Figura 4.24: Actividad actualizar estadísticas

4.4. Modelo de paquetes

Para una mejor comprensión de la arquitectura del sistema, los elementos del modelo de objetos se agrupan en paquetes. Un paquete organiza los objetos en grupos de modo que puedan ser manipulados de forma global. Cada paquete agrupa elementos cercanos semánticamente y funcionalmente. La visibilidad de los elementos del paquete va implícita en la propia especificación de cada elemento.

La interfaz del paquete la conforman todos aquellos elementos que son visibles fuera del paquete. Cada paquete posee un nombre único dentro del sistema. Un paquete puede contener subpaquetes cuyo nombre va precedido del nombre del paquete contenedor. Cada objeto pertenece exclusivamente a un único paquete. Un paquete forma un espacio de nombres, lo que significa que la nomenclatura de los elementos del mismo grupo no debe repetirse en el contexto de su paquete contenedor. Un paquete puede importar la interfaz de otro paquete para acceder a sus elementos. Esta importación es un permiso explícito de acceso en un solo sentido.

Un paquete puede ser representado gráficamente como una proyección de su modelo de contenido. Para la representación de cada paquete se hará uso de la notación gráfica de UML 2.0.

Para el caso de este proyecto tenemos una arquitectura cliente-servidor, por tanto se debe hacer una separación clara entre la interfaz de usuario del sistema que reside en el cliente y los datos persistentes del sistema que residen en el servidor.

4.4.1. Paquete Cliente de Wikicode

Este paquete representa el bloque cliente de la arquitectura. El bloque cliente visto desde el punto de vista físico estará formado por al menos un terminal que estará ejecutando la plataforma Moodle remotamente (navegador). Este terminal estará conectado con un servidor que será el módulo de Moodle que aceptará las peticiones del usuario.

4.4.2. Paquete Servidor/Cliente Wikicode-Mod de Wikicode

Esta parte representa al módulo de Moodle que actúa como servidor aceptando las peticiones del usuario y sirviendo de interfaz de entrada para la edición de código. Sin embargo, este servidor también actuará como cliente pues será el que realice las peticiones de sincronización de datos y salvado en base de datos.

De esta manera, este módulo servidor/cliente deberá hacer lo siguiente:

- Aceptar las peticiones del usuario.
- Obtener los registros de base de datos conforme a las peticiones de los usuarios.
- Hacer de interfaz de entrada a la hora de edición del código fuente.
- Enviar la petición de sincronización de código mediante un paquete JSon y el método Get¹.
- Enviar la petición de escritura en la conversación agrupada como chat mediante el método Post².
- Llamar al compilador externo y leer la salida del fichero de intercambio.

La finalidad de este bloque es realizar una separación entre la aplicación que maneja la sincronización del código fuente y el sistema de gestión de cursos consiguiendo la independencia de ambos. Esta independencia es fundamental pues permite utilizar la aplicación de sincronización para cualquier otra plataforma que no sea Moodle.

4.4.3. Paquete Servidor Wikicode-Synchronizer de Wikicode

El tercer bloque consiste en una clase que recoge el código y la posición del carácter modificado y la petición de sincronización o bloqueo. Una vez recogida la información, realiza los algoritmos necesarios para sincronizar el código fuente para que actúe de modo colaborativo. Una vez los datos se han sincronizado se almacenan en base de datos para que el bloque *Wikicode-Mod* pueda tener acceso a ellos siempre que los necesite. También se enviará la información sincronizada como un paquete JSON para que dicho bloque no tenga que acceder de manera secuencial a base de datos.

¹**Get:** Método de petición HTTP el cual pide una representación del recurso especificado.

²**Post:** Método de petición HTTP el cual somete los datos a que sean procesados para el recurso identificado. Los datos se incluirán en el cuerpo de la petición.

4.4.4. Diagrama de paquetes

El diagrama de paquetes correspondiente es el siguiente:

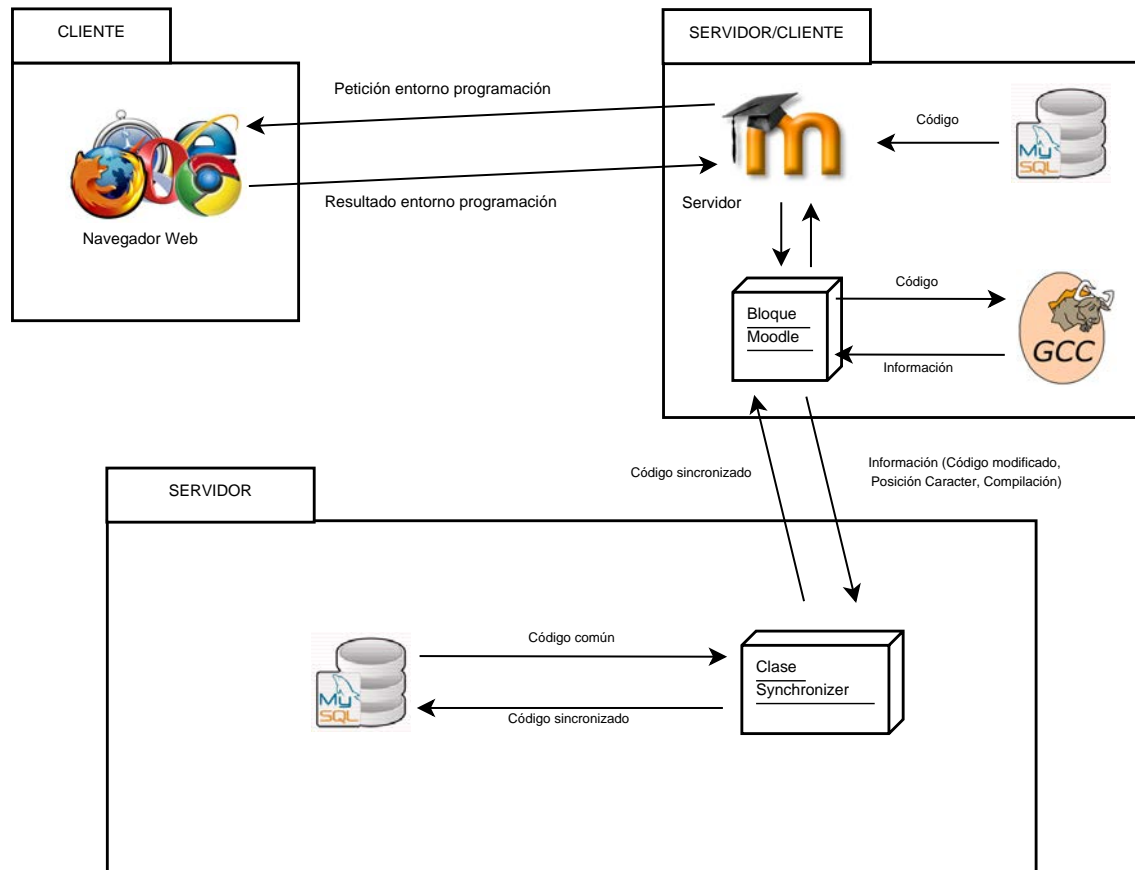


Figura 4.25: Diagrama de paquetes

Capítulo 5

Implementación del sistema

La vista de implementación se ocupa de la gestión de la configuración de las partes del sistema (componentes), las cuales pueden ensamblarse de diferentes formas para obtener el sistema ejecutable.

De esta manera, los diagramas de componentes aparecen cuando se modelan los aspectos físicos de los sistemas orientados a objetos. Así, los diagramas de componentes muestran la organización y las dependencias entre un conjunto de componentes y se utilizan para modelar la vista de implementación estática de un sistema.

Por tanto, ahora se va a comentar el diagrama de componentes asociado al sistema para especificar la implementación del sistema.

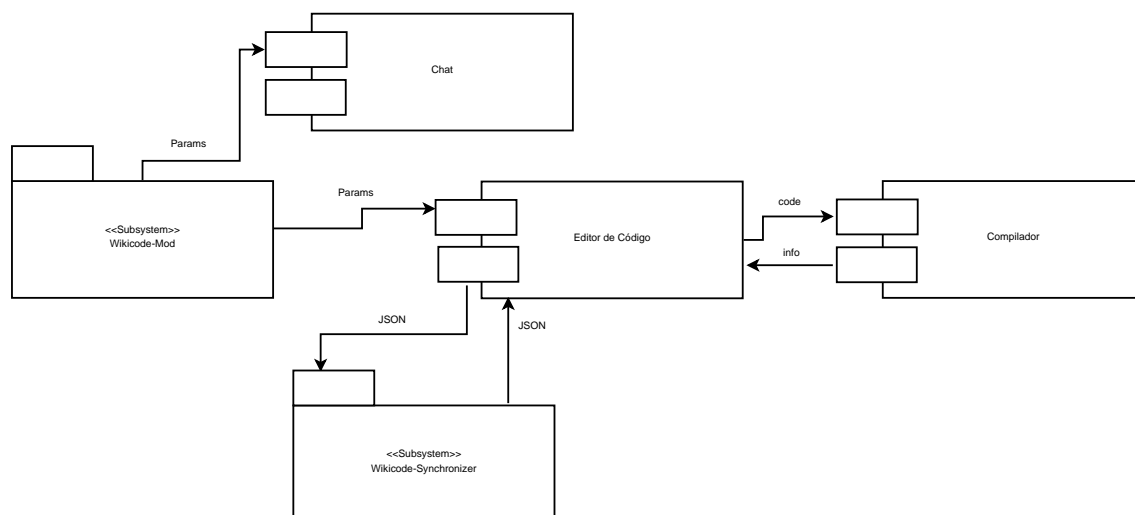


Figura 5.1: Diagrama de Componentes

Capítulo 6

Pruebas del sistema

La fase de pruebas es una fase primordial para garantizar la fiabilidad de un sistema, debido a que representa un último repaso a las especificaciones, al diseño y a la implementación del mismo. Esta parte de la documentación pretende realizar una descripción de las pruebas realizadas al software desarrollado y cuyos correctos resultados permitirán asegurar que el sistema satisface los requisitos impuestos y cumple con los objetivos deseados de manera robusta y eficiente.

6.1. Objetivos de las pruebas

Las pruebas son una actividad en el desarrollo del software en la cual el sistema o alguno de sus componentes se ejecutarán en circunstancias previamente especificadas, observando y registrando los resultados para realizar una evaluación de algún aspecto. Probar es el proceso de ejecutar un programa con el fin de encontrar errores.

En este caso, la prueba exhaustiva del sistema software con el que se trata resulta perfectamente viable debido a que se pueden probar con facilidad todas las posibilidades de su funcionamiento sin incurrir en ningún error. Por ello se plantean una serie de puntos generales para este proceso:

- El objetivo es realizar pruebas que delaten distintos errores con poco tiempo y esfuerzo.
- Secundariamente demostraremos con las pruebas qué módulos del software funcionan de acuerdo a las especificaciones y requisitos dados. El comportamiento de la aplicación nos indicará qué grado de fiabilidad tiene el programa y su calidad.

- Por último, hay que tener en cuenta que las pruebas no aseguran la falta de defectos. Sin embargo, si todo esto resultara ineficiente, se ha de recordar que tras esta fase de pruebas se lleva a cabo una etapa de explotación del sistema, con la peculiaridad de que éste se dispondrá a funcionar en el entorno propio y real de la aplicación pero bajo la supervisión y controlado por los encargados directos del mismo, pudiendo observar si verdaderamente el programa esta listo para su definitiva aceptación.

6.2. Diseño de los casos de pruebas

Analizando las especiales características del software, se llega a la conclusión de que es necesario prestar especial atención a:

- Asegurar la comunicación entre el módulo de Moodle y el sistema encargado de sincronizar el código y programar los bloqueos. Esta comunicación es la base del sistema por lo que deben estar completamente sincronizados de manera que cada petición sea servida convenientemente. Además, cada cambio en el editor introducida en el cliente debe ser indicada al servidor para que este se encargue de sincronizarlo con el resto de usuarios.
- Garantizar la correcta visualización de las pantallas, fundamentalmente la de edición.
- Garantizar el correcto acceso a la base de datos, ya sea para obtener cualquier dato de la comunidad como para almacenar y recuperar los datos referentes a la sincronización del código.
- Asegurar que un usuario no puede modificar una parte bloqueada por otro usuario, ya sea por error suyo o problemas en el sistema.
- Asegurar el correcto funcionamiento de todas las demás opciones posibles.

6.3. Pruebas estructurales de la caja blanca

Aprovechando que tanto Java como PHP permiten una sencilla y cómoda interpretación de código, gracias sobre todo a las potentes herramientas que existen para cada una de ellas y que se comentaron en el apartado de recursos software, se ha ejecutado el código al completo haciendo énfasis en bucles y condiciones forzando dicha ejecución en todas aquellas partes de código que por un motivo u otro no suelen ser accedidas. Gracias a este proceso, se han descubierto algunos fallos en dichas zonas que han sido resueltos convenientemente.

Debido a las características del software que se ha desarrollado, el enfoque de las pruebas que, teóricamente, más errores podrá detectar es el funcional o de la caja negra.

6.4. Pruebas funcionales o de la caja negra

6.4.1. Análisis de los valores límites

Al tratarse de un sistema donde la intercomunicación entre el usuario y este es constante, en este apartado es donde radica el mayor número de pruebas. Como la parte fundamental de este proyecto se trata de un editor, el usuario tiene permitido hacer en él todo lo que desee y prácticamente sin limitación alguna. Sin embargo, para evitar errores se ha impedido al usuario realizar una serie de acciones que podrían poner en peligro la estabilidad del sistema.

Muchas de estas acciones han sido camufladas como ayuda al usuario, como puede ser por ejemplo que la apertura y cerrado de funciones se hace de modo automático. De este modo, y al ser los bloqueos por funciones, es el sistema el que lo delimita automáticamente y no así el usuario.

Una vez puntualizados estos datos, se van a enumerar una serie de hechos que son candidatos a la realización de pruebas:

- Un punto en el que se han hecho pruebas exhaustivas es en el acceso a las partes bloqueadas por otro usuario del código fuente. Asimismo también se ha hecho énfasis en los puntos de sección crítica que puedan concurrir.
- Otro punto importante es la llamada al compilador, que al tratarse de una aplicación externa lo hace necesario del uso de ficheros, por lo que se ha evaluado el acceso a estos.
- Del mismo modo que el punto anterior también se presta atención al chat, que también trabaja con ficheros, por lo que hay que hacer pruebas exhaustivas para su acceso a estos.
- Otras pruebas que se han realizado han sido pruebas a la interfaz del software, es decir, se ha probado que todas las pantallas sean accesibles y que la comunicación entre ellas es eficiente.

6.5. Pruebas del desarrollador

Realmente son las pruebas más importantes que se han realizado. Básicamente estas pruebas se han centrado fundamentalmente en tres puntos comentados a continuación.

6.5.1. Pruebas con un curso real

Antes de nada comentar que la utilidad de este software se centra en las comunidades virtuales de aprendizaje en las que la colaboración es su principal objetivo. Si esto no es así, gran parte de la utilidad de esta herramienta se perdería, pues la mayor funcionalidad que expresa es su capacidad de hacer el editor colaborativo en tiempo real.

Para ello en primer lugar se ha creado un curso en Moodle, dentro del cual un profesor simulado ha ido creado una serie anotaciones y matriculando una serie de alumnos virtuales.

The screenshot shows the Moodle interface for a course titled "Metodología y Tecnología de la Programación". The user is logged in as "Antonio J. González (Salir)". The page layout includes a left sidebar with navigation links, a main content area with a calendar of activities, and a right sidebar with search and news sections.

Metodología y Tecnología de la Programación

Usted se ha identificado como [Antonio J. González \(Salir\)](#)
Español - España (es_es)

[Página Principal](#) ► [Mis cursos](#) ► **MTP**

Navegación

- [Página Principal](#)
- [Área personal](#)
- [Páginas del sitio](#)
- [Mi perfil](#)
- ▼ [Mis cursos](#)
 - ▼ **MTP**
 - [Participantes](#)
 - [General](#)
 - 18 de agosto - 24 de agosto
 - 25 de agosto - 31 de agosto
 - 1 de septiembre - 7 de septiembre
 - 8 de septiembre - 14 de septiembre
 - 15 de septiembre - 21 de septiembre
 - 22 de septiembre - 28 de septiembre
 - 29 de septiembre - 5 de octubre
 - 6 de octubre - 12 de octubre
 - 13 de octubre - 19 de octubre
 - 20 de octubre - 26 de octubre

Ajustes

Novedades

- 18 de agosto - 24 de agosto
 - [Práctica Documentación](#)
- 25 de agosto - 31 de agosto
 - [Práctica Documentación Grupal](#)
- 1 de septiembre - 7 de septiembre
- 8 de septiembre - 14 de septiembre
- 15 de septiembre - 21 de septiembre
- 22 de septiembre - 28 de septiembre
- 29 de septiembre - 5 de octubre
- 6 de octubre - 12 de octubre
- 13 de octubre - 19 de octubre
- 20 de octubre - 26 de octubre

Buscar foros

[Ir](#)

[Búsqueda avanzada](#) ?

Últimas noticias

(Sin novedades aún)

Eventos próximos

No hay eventos próximos
[Ir al calendario...](#)
[Nuevo evento...](#)

Actividad reciente

Actividad desde sábado, 18 de agosto de 2012, 18:28
[Informe completo de la actividad reciente...](#)

Actualizaciones de cursos:

Agregado Wikicode:
[Práctica Documentación Grupal](#)

Updated wiki pages:

- 20 de ago, 10:11
[Práctica Documentación](#)
- 20 de ago, 18:27
[Práctica Documentación Grupal](#)

Usted se ha identificado como [Antonio J. González \(Salir\)](#)

[Página Principal](#)

Figura 6.1: Pantalla principal de un curso Moodle

Una vez dentro del curso se han creado una serie de Wikicodes diferentes para probar las distintas opciones de configuración. Estas pruebas tienen como principales objetivos:

- Realizar una serie de códigos fuentes, tanto individualmente como de modo colaborativo, y luego comprobar si la compilación es correcta.
- Si la compilación es errónea, comprobar que las líneas referenciadas son claramente identificadas en el editor dentro de la Wikicode.
- Comprobar que todas las acciones que puedan realizarse dentro de un entorno de desarrollo pueden hacerse libremente dentro de una Wikicode, desde la restauración de una versión antigua a desbloquear partes de tu código.

Por tanto, dicho curso se ha desarrollado de manera completa y se han creado varios alumnos ficticios los cuales realizaron numerosas aportaciones y colaboraciones en la comunidad. Además, se ha utilizado para probar todas las opciones disponibles intentando barajar todas posibilidades que el usuario podría realizar en el entorno del curso.



Figura 6.2: Edición de código con una función bloqueada

También se ha probado la opción de Descarga del ejecutable una vez compilado y se ha comprobado que se ejecuta correctamente.

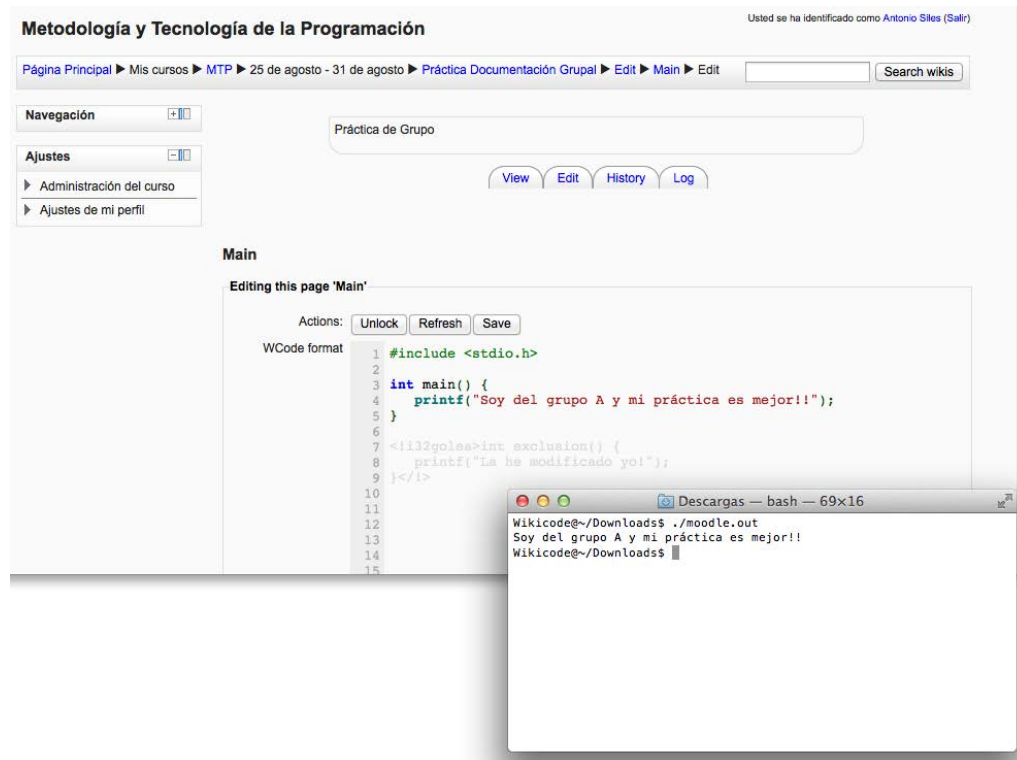


Figura 6.3: Edición de código. Descarga y ejecución de una aplicación creada.

Por último también se ha comprobado la entrada/salida de texto con respecto al chat que hemos incluido en nuestra plataforma de desarrollo antes de dar el visto bueno a la parte de edición de la plataforma.

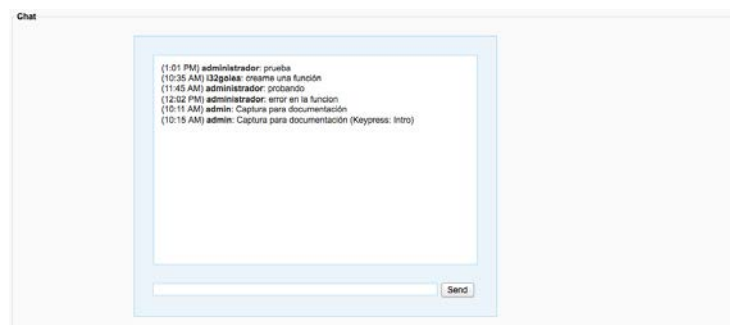


Figura 6.4: Edición de código. Chat.

6.5.2. Prueba de compilación utilizando Cross Compiling

Para hacer esta prueba se ha inicializado una máquina virtual con Windows instalado y posteriormente se ha compilado y ejecutado el ejecutable descargado. La salida es la misma que nos ha dado en el sistema operativo nativo de nuestra aplicación por lo que podemos ver que las pruebas realizadas son correctas.

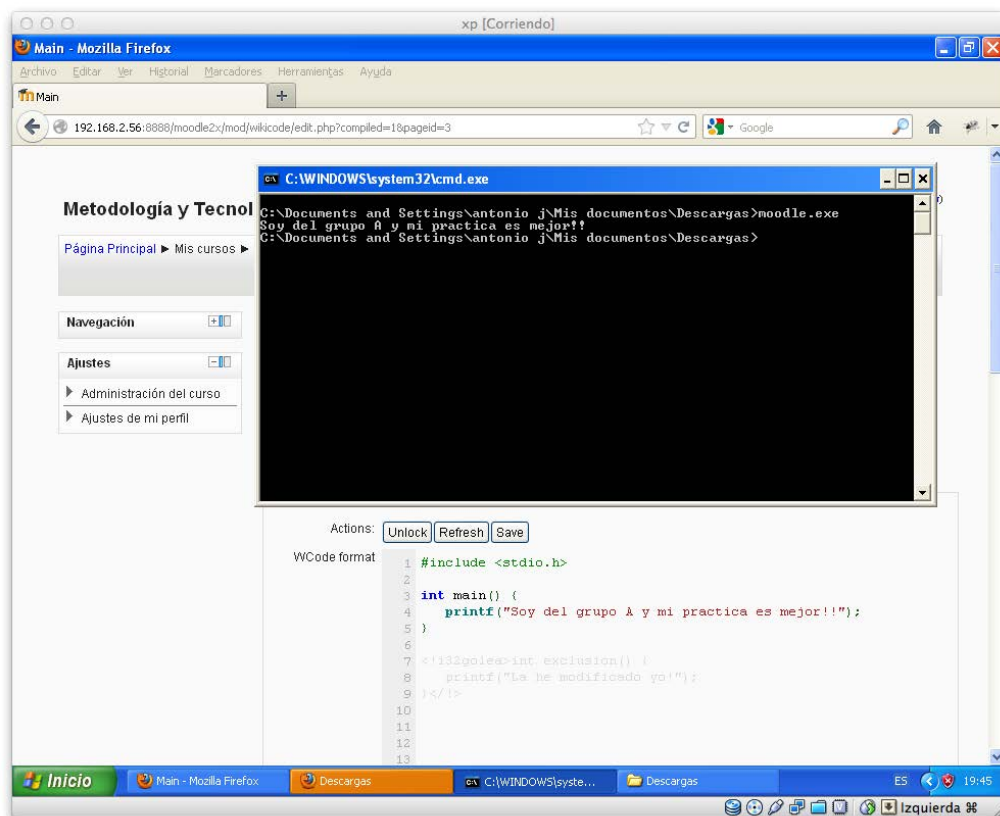


Figura 6.5: Edición de código. Cross Compiling.

6.5.3. Comprobación de estadísticas

Una parte fundamental de Wikicode es su parte estadística, donde el profesor puede comprobar la dificultad que ha tenido un alumno para realizar un determinado código y en función de esto valorar el conocimiento del alumno. Para ello se muestran dos métricas: Número de errores de compilación y Duración de desarrollo del código.

Se han hecho las pruebas oportunas, calculando el tiempo y el número de errores de modo paralelo y comprobando de la veracidad de los hechos.

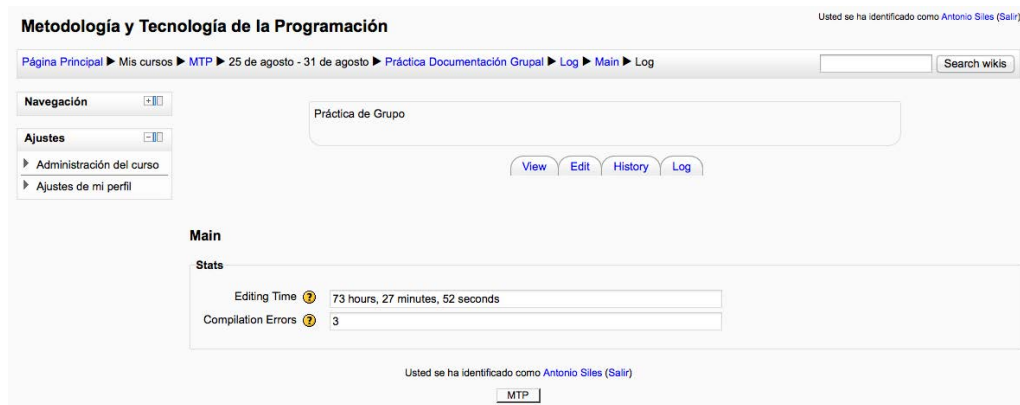


Figura 6.6: Log de información de una Wikicode.

6.6. Corrección de errores

Durante las pruebas se han detectado varios errores de programación, que se han ido corrigiendo conforme se han ido detectando. También se han corregido dos problemas a los que se ha aportado una solución funcional para protección del código, los cuales son los siguientes:

- **Borrado de caracteres especiales.** La aplicación funciona delimitándose por funciones, lo cual nos ayuda a separar los bloqueos unos de otros. Las funciones en el lenguaje de programación C se delimitan por la apertura y cerrado de corchetes, por lo que para eliminar uno de estos caracteres el contenido de su interior ahora deberá de estar vacío. Esto nos impide que el sistema desconozca el dominio de una función.
- **Auto Rollback.** Se ha podido comprobar durante las diversas pruebas que el sistema en caso de error de bloqueos vacía el buffer del código y lo deja en blanco. Se ha modificado la programación de todas las funciones para asegurarnos que siempre que un usuario modifica una función esta esté bloqueada por él, y del mismo modo, por si no se ha detectado algún caso que repercuta en el mismo error se ha programado el sistema de modo que si se vacía el buffer esta acción no se lleve a cabo y restaure el contenido que había anteriormente a la acción que lo ha propiciado.

6.7. Prueba de aceptación

Esta prueba se ha realizado con varias personas distintas del autor. El uso de la aplicación por parte de esas personas no ha ocasionado ningún error y, por tanto, se puede decir que esta prueba se ha superado de manera satisfactoria.

Capítulo 7

Conclusiones y futuras mejoras

7.1. Conclusiones

El principal objetivo de este proyecto ha sido alcanzado puesto que se ha realizado un editor de código fuente en lenguaje C, colaborativo, e integrado dentro un sistema e-learning. Para este fin se ha conseguido diseñar un módulo compatible con Moodle en todas sus versiones.

De este modo, todos los objetivos referentes al hecho de crear un editor on-line se han cumplido con todas las características de un entorno de desarrollo. También se ha permitido al tutor la configuración de una serie de parámetros que son utilizados a la hora de compilar y desarrollar los códigos fuentes. Estos parámetros nos permiten compilar en cualquier OS que deseemos instalar el sistema así como tener una serie de estadísticas.

Una de las principales ventajas que se han obtenido con el desarrollo de este proyecto es que ahora los alumnos cuentan con una gran herramienta para programar, sin necesidad de tener conocimiento informático alguno y sin necesidad de tener que instalar nada en sus equipos. A la vez, fomenta la colaboración entre los compañeros a la hora de realizar cualquier ejercicio de programación e impide hacer el típico *copy&paste* para la realización de prácticas. Por otra parte, el profesor cuenta ahora con una herramienta más potente a la hora de evaluar los conocimientos en este área de su alumnado.

Además el back-end¹ de la aplicación es totalmente independiente. De este modo, si se deseara exportar a otro sistema e-learning bastaría con diseñar un front-end dentro de ese sistema y enlazar con nuestro sistema.

¹**Back-end:** Parte de la aplicación encargada de administrar el sistema.

Otro de los objetivos cumplidos es la realización de un estudio de los sistemas e-learning más importantes en la actualidad analizando sus pros y sus contras así como realizando una explicación detallada de la elección de Moodle.

En cuanto al diseño de interfaz que se ha utilizado, como realmente lo que se ha desarrollado es un bloque para Moodle y este no tiene mucha complejidad, se ha seguido el mismo estilo gráfico de dicha plataforma. Sin embargo, también se han añadido algunos efectos visuales utilizando técnicas de diseño avanzadas como es el caso del framework JQuery.

7.2. Posibles ampliaciones o mejoras

A continuación se exponen una serie de ampliaciones o mejoras que podrían realizarse en un futuro sobre este sistema:

- Se podría aumentar el número de lenguajes de programación disponibles. Ahora mismo el sistema sólo funciona para el lenguaje **C**, y eso puede limitar la exportación de este módulo para otras asignaturas en las que se enseñe algún otro lenguaje de programación.
- Se puede permitir la realización de proyectos informáticos, dando la posibilidad de manejar varios ficheros en lugar de un único código fuente.
- Crear un sistema de importación y exportación de proyectos, de modo que el alumno pueda descargar todos los datos en su equipo para trabajar off-line.
- Crear un sistema de ejecución embebido dentro de la propia plataforma de desarrollo, así el alumno no necesitaría descargar el archivo en su equipo para ejecutarlo.

Bibliografía

- [amy12] Amy editor. <http://www.amyeditor.com/>, September 2012. 6, 10
- [atu12] Atutor learning management system. <http://atutor.ca/>, September 2012. 1, 30
- [CS11] Earle Castledine and Craig Sharkie. *jQuery: Novice to Ninja*. 2011.
- [egw12] Egroupware. <http://www.egroupware.org/>, September 2012. 33
- [etc12] Ether code. <http://ethercodes.com/>, September 2012. 6, 10
- [GG12] Patxi Gortázar and Micael Gallego. Eclipsegavab. <http://www.sidelab.es/projects/eclipsegavab/>, September 2012. 10
- [ili12] Ilias e-learning. <https://www.ilias.de/>, September 2012. 1, 25
- [jsb12] Jsbin. <http://jsbin.com/>, September 2012. 10
- [LRGN97] Irene Luque Ruiz and Miguel Ángel Gómez-Nieto. *Diseño y uso de bases de datos relacionales*. Ra-Ma Editorial, 1997.
- [LRGN99] Irene Luque Ruiz and Miguel Ángel Gómez-Nieto. *Ingeniería del Software*. Universidad de Córdoba, 1999.
- [MC10] J. Moore and M. Churchward. *Moodle 1.9 Extension Development*. PACKT, 2010.
- [mdl12] Moodle: Open-source community-based tools for learning. <http://moodle.org/>, September 2012. 1, 18
- [Mur11] Rebecca Murphey. *jQuery Fundamentals*. 2011.
- [npp12] Notapipe. <http://notapipe.net/>, September 2012. 6, 10

-
- [ocw12] Open cooperative web framework. <http://opencoweb.org/>, September 2012. 11
- [PCM96] M.G. Piattini and J.A. Calvo-Manzano. *Análisis y diseño detallado de Aplicaciones Informáticas de Gestión*. Ra-Ma Editorial, 1996.
- [Pre93] R.S. Pressman. *Ingeniería del Software*. McGraw-Hill, 1993.
- [sak12] Sakai project. <http://www.sakaiproject.org/>, September 2012. 28
- [Sch04] George Schlossnagle. *Advanced PHP Programming*. Sams, 2004.
- [skw12] Mozilla skywriter. <https://mozillalabs.com/en-US/skywriter/>, September 2012. 11
- [Som88] I. Sommerville. *Ingeniería del Software*. Addison-Wesley, 1988.
- [wct12] Web course tools. <http://www.webct.com/>, September 2012. 1, 21