



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

APLICACIÓN WEB PARA LA GESTIÓN DE LA PLANIFICACIÓN DOCENTE DE LOS GRADOS

Daniel Ignacio Salazar Recio

12 de diciembre de 2011



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERO TÉCNICO EN INFORMÁTICA DE SISTEMAS

APLICACIÓN WEB PARA LA GESTIÓN DE LA PLANIFICACIÓN DOCENTE DE LOS GRADOS

- Departamento: Lenguajes y sistemas informáticos
- Director del proyecto: Juan José Domínguez Jiménez
- Autor del proyecto: Daniel Ignacio Salazar Recio

Cádiz, 12 de diciembre de 2011

Fdo: Daniel Ignacio Salazar Recio

Agradecimientos

Me gustaria agradecer y/o dedicar este texto a ...

Licencia

Este documento ha sido liberado bajo Licencia GFDL 1.3 (GNU Free Documentation License). Se incluyen los términos de la licencia en inglés al final del mismo.

Copyright (c) 2011 Daniel Ignacio Salazar Recio.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Notación y formato

Aquí incluiremos los aspectos relevantes a la notación y el formato a lo largo del documento. Para simplificar podemos generar comandos nuevos que nos ayuden a ello, ver `comandos.sty` para más información.

Cuando nos refiramos a un programa en concreto, utilizaremos la notación:

emacs.

Cuando nos refiramos a un comando, o función de un lenguaje, usaremos la notación:

`quicksort`.

Índice general

1. Introducción	1
1.1. Objetivos y alcance	1
1.2. Estructura del documento	1
1.3. Definiciones y acrónimos	2
2. Planificación	3
2.1. Fase inicial	3
2.2. Fase de análisis	3
2.3. Fase de aprendizaje	3
2.4. Fase de diseño	3
2.5. Implementación	3
2.6. Pruebas	4
2.7. Redacción de la memoria	4
2.8. Diagrama de Gantt	4
3. Descripción general del proyecto	7
3.1. Descripción	7
3.2. Perfiles de usuario	7
3.2.1. Perfil Administrador	7
3.2.2. Perfil Planificador	7
3.2.3. Perfil Profesor	7
3.2.4. Perfil Alumno	7
3.3. Interfaz de usuario	8
3.4. Software	8
4. Análisis	9
4.1. Metodología de desarrollo	9
4.2. Especificación de requisitos del sistema	9
4.2.1. Gestión de usuarios	10
4.2.2. Gestión de titulaciones	12
4.3. Modelo conceptual de datos	38
4.3.1. Diagrama de clases conceptuales	38
4.4. Modelo de comportamiento del sistema	38
4.4.1. Caso de uso: Registrar titulación	39
4.4.2. Caso de uso: Registrar asignatura	40
4.4.3. Caso de uso: Editar asignatura	41
4.4.4. Caso de uso: Borrar asignatura	42
4.4.5. Caso de uso: Importar asignatura	42
4.4.6. Caso de uso: Crear plan docente	43

4.4.7.	Caso de uso: Generar informe de asignatura	44
4.4.8.	Caso de uso: Añadir evento al calendario	44
4.4.9.	Caso de uso: Exportar calendario	45
4.4.10.	Caso de uso: Seleccionar grupos de teoría	45
4.4.11.	Caso de uso: Añadir grupo de teoría	46
4.4.12.	Caso de uso: Eliminar grupo de teoría	46
4.4.13.	Caso de uso: Ubicar slot de horario	47
4.4.14.	Caso de uso: Editar horario tipo	48
4.4.15.	Caso de uso: Comprobar grupo de teoría	48
4.4.16.	Caso de uso: Crear aula	49
4.4.17.	Caso de uso: Borrar aula	49
4.4.18.	Caso de uso: Editar aula	50
5.	Diseño	51
5.1.	Controladores	51
5.1.1.	Users	51
5.1.2.	Login	52
5.1.3.	Logout	52
5.1.4.	Admin	52
5.1.5.	Titulaciones	52
5.1.6.	Asignaturas	53
5.1.7.	Aulas	53
5.1.8.	Cursos	53
5.1.9.	Eventos	53
5.1.10.	Horarios	54
5.1.11.	PlanesDocentes	54
5.2.	Base de datos	55
5.2.1.	Modelo entidad-relación	57
5.2.2.	Tablas y atributos	58
6.	Implementación	63
6.1.	Lenguajes	63
6.2.	Extensiones y librerías	64
6.3.	Herramientas utilizadas	64
6.4.	Detalles de la implementación de la arquitectura del sistema	65
6.4.1.	Capa modelo	65
6.4.2.	Capa controlador	66
6.4.3.	Capa vista	67
7.	Pruebas	71
7.1.	Pruebas sobre los datos	71
7.2.	Especificación del diseño de pruebas	71
7.3.	Especificación de los procedimientos de prueba	72
7.4.	Documentación de la ejecución de las pruebas	73
7.5.	Herramientas utilizadas para las pruebas	73
8.	Conclusiones	75
8.1.	Opinión personal	75
8.2.	Ampliaciones futuras	75

Bibliografía y referencias	77
GNU Free Documentation License	79
1. APPLICABILITY AND DEFINITIONS	79
2. VERBATIM COPYING	80
3. COPYING IN QUANTITY	80
4. MODIFICATIONS	81
5. COMBINING DOCUMENTS	82
6. COLLECTIONS OF DOCUMENTS	83
7. AGGREGATION WITH INDEPENDENT WORKS	83
8. TRANSLATION	83
9. TERMINATION	83
10. FUTURE REVISIONS OF THIS LICENSE	84
11. RELICENSING	84
ADDENDUM: How to use this License for your documents	84

Indice de figuras

2.1. Diagrama de Gannt. Desarrollo del proyecto 1/2	5
2.2. Diagrama de Gannt. Desarrollo del proyecto 1/2	6
4.1. Diagrama de casos de uso de la gestión de usuarios	10
4.2. Diagrama de casos de uso de la gestión de titulaciones	12
4.3. Diagrama de casos de uso de la gestión de asignaturas	16
4.4. Diagrama de casos de uso de la gestión de cursos	20
4.5. Diagrama de casos de uso de la gestión de la planificación docente	23
4.6. Diagrama de casos de uso de la gestión del calendario	27
4.7. Diagrama de casos de uso de la gestión de horarios	30
4.8. Diagrama de casos de uso de la gestión de aulas	35
4.9. Diagrama del modelo conceptual de datos	38
4.10. Diagrama de secuencia del caso de uso Registrar titulación	39
4.11. Diagrama de secuencia del caso de uso Registrar asignatura	40
4.12. Diagrama de secuencia del caso de uso Editar asignatura	41
4.13. Diagrama de secuencia del caso de uso Borrar asignatura	42
4.14. Diagrama de secuencia del caso de uso Importar asignatura	42
4.15. Diagrama de secuencia del caso de uso Crear plan docente	43
4.16. Diagrama de secuencia del caso de uso Generar informe de asignatura	44
4.17. Diagrama de secuencia del caso de uso Añadir evento al calendario	44
4.18. Diagrama de secuencia del caso de uso Exportar calendario	45
4.19. Diagrama de secuencia del caso de uso Seleccionar grupos de teoría	45
4.20. Diagrama de secuencia del caso de uso Añadir grupo de teoría	46
4.21. Diagrama de secuencia del caso de uso Eliminar grupo de teoría	46
4.22. Diagrama de secuencia del caso de uso Ubicar slot de horario	47
4.23. Diagrama de secuencia del caso de uso Editar horario tipo	48
4.24. Diagrama de secuencia del caso de uso Comprobar grupo de teoría	48
4.25. Diagrama de secuencia del caso de uso Crear aula	49
4.26. Diagrama de secuencia del caso de uso Borrar aula	49
4.27. Diagrama de secuencia del caso de uso Editar aula	50
5.1. Diagrama Entidad-Relación	57
6.1. Captura de la estructura de una página	68
6.2. Captura del listado de titulaciones	68
6.3. Captura de página de horarios	69

Indice de tablas

5.1. Estructura de la tabla actividades	58
5.2. Estructura de la tabla asignaturas	58
5.3. Estructura de la tabla aulaactividades	58
5.4. Estructura de la tabla aulas	59
5.5. Estructura de la tabla ci_sessions	59
5.6. Estructura de la tabla cursos	59
5.7. Estructura de la tabla cursos_compartidos	60
5.8. Estructura de la tabla eventos	60
5.9. Estructura de la tabla horarios	60
5.10. Estructura de la tabla horario_reference	60
5.11. Estructura de la tabla lineashorarios	61
5.12. Estructura de la tabla planactividades	61
5.13. Estructura de la tabla planesdocentes	61
5.14. Estructura de la tabla titulaciones	61
5.14. Estructura de la tabla titulaciones (continúa)	62
5.15. Estructura de la tabla users	62

Capítulo 1

Introducción

Con este Proyecto de Fin de Carrera se pretende la consecución de dos objetivos fundamentales: poner en práctica los conocimientos adquiridos en la titulación de Ingeniería Técnica en Informática de Sistemas y buscar un incremento de los conocimientos en la rama del desarrollo web, al no haber estudiado nada de este tema durante la carrera.

1.1. Objetivos y alcance

El proyecto consiste en la creación de un software que ayude a los coordinadores de las titulaciones de grado a programar la planificación docente. En principio está pensado únicamente para el contexto de la Escuela Superior de Ingeniería, aunque debería ser fácilmente adaptable a otras facultades.

Actualmente para hacer esta planificación se utilizan hojas de cálculo de Microsoft Excel o OpenOffice, haciendo que el trabajo sea algo tedioso al tener que comprobar multitud de factores manualmente. La aplicación pretende facilitar esta labor, realizando esas comprobaciones automáticamente. Por ejemplo la tarea de realizar un horario y comprobar que un aula no esté ya ocupada por otra asignatura. Para ello el objetivo es crear una aplicación web de código abierto.

Otro objetivo que se pretende con este proyecto es hacerlo escalable, para que en un futuro se le puedan realizar las ampliaciones necesarias sin necesidad de cambiar demasiado lo que está ya hecho.

1.2. Estructura del documento

El documento se compone de los siguientes capítulos:

- **Introducción:** descripción del proyecto, objetivos y alcance del mismo y estructura básica del documento.
- **Planificación:** descripción del desarrollo de la planificación temporal y plazos de realización.
- **Descripción general:** descripción detallada sobre el proyecto, especificando tecnologías y herramientas usadas para su desarrollo.
- **Análisis:** fase de análisis del sistema, empleando la metodología seleccionada. Definición de requisitos funcionales del sistema, modelo conceptual y modelo de comportamiento.

- **Diseño:** fase de diseño del sistema, diseño de la base de datos y diagramas de clase aplicadas al diseño.
- **Implementación:** aspectos más relevantes de la fase de implementación del sistema y explicación de los problemas encontrados durante el desarrollo.
- **Pruebas y validaciones:** pruebas realizadas al software para verificar que todo funciona correctamente y según lo esperado.
- **Conclusiones:** valoración y conclusiones personales obtenidas tras la realización del proyecto.
- **Apéndices:**
 - **Manual de instalación:** manual para instalar correctamente la aplicación.
 - **Manual de usuario:** manual para ayudar al usuario en el uso de la aplicación.
 - **Manual de importación:** ejemplos de los archivos admitidos por la aplicación para importar y exportar datos.
 - **Exportación:** ejemplos de los archivos exportados por la aplicación.
- **Bibliografía:** libros y referencias consultadas durante la realización del proyecto.
- **Licencia GPL 3:** texto completo sobre la licencia GPL 3, por la cual se rige el proyecto.

1.3. Definiciones y acrónimos

A continuación se detallan las abreviaturas y acrónimos utilizados a lo largo de todo el documento.

- **PHP:** PHP: Hypertext Preprocessor. Es un lenguaje de scripting del lado del servidor.
- **XHTML:** eXtensible Hypertext Markup Language. Lenguaje de marcado para estructurar las vistas de un documento web.
- **IDE:** Entorno de desarrollo integrado. Es una aplicación con herramientas para facilitar el trabajo de un desarrollador.
- **SQL:** Lenguaje de consulta estructurado. Lenguaje para realizar operaciones sobre una base de datos.
- **MySQL:** Sistema de gestión de base de datos relacional.
- **CSS:** Cascading Style Sheets, hojas de estilo en cascada. Utilizadas para definir el estilo de un documento web.
- **ER:** Entidad-relación. Diagrama utilizado para mostrar la especificación de una base de datos.
- **ESI:** Escuela Superior de Ingeniería.
- **MVC:** Modelo Vista Controlador. Patrón de diseño arquitectónico utilizado a la hora de hacer el diseño de un sistema.

Capítulo 2

Planificación

La planificación se divide en varias fases, a continuación se explicará en detalle cada una de ellas.

2.1. Fase inicial

La primera fase consistió en el planteamiento de la idea del proyecto, que en principio era desarrollar una aplicación web. El tutor finalmente decide proponer este proyecto, dejando al alumno la libre elección de las tecnologías utilizadas para su desarrollo.

2.2. Fase de análisis

Se realizan diversas reuniones con el tutor para hacer el planteamiento y una especificación informal de los requisitos. Debido a la complejidad de la aplicación fue una labor compleja que necesitó de varias reuniones y en las que hubo cambios en los requisitos debido a la forma en la que se realiza la planificación docente de las titulaciones de grado, que ha cambiado durante los últimos años.

2.3. Fase de aprendizaje

Para la realización del proyecto usaron tecnologías de las que no se tenían conocimiento, por lo tanto fue necesaria una amplia fase de aprendizaje. Esta fase se puede dividir en tres partes, el aprendizaje de PHP como lenguaje, aprendizaje del framework CodeIgniter y del ORM Doctrine, y finalmente de los lenguajes de la parte del cliente, es decir, HTML y JavaScript.

2.4. Fase de diseño

Fase en la que se realiza el diseño de la aplicación. Es importante hacer un buen diseño aquí para que no surjan problemas más adelante y se haga necesario hacer cambios muy costosos en el diseño.

2.5. Implementación

Fase más extensa del desarrollo del proyecto. Consiste en implementar los requisitos especificados en la fase de análisis siguiendo para ello el diseño realizado en la fase anterior, procurando que la aplicación final satisfaga las necesidades.

2.6. Pruebas

Etapa importante en la que se comprueba una por una las funcionalidades del sistema verificando que no hay errores y que todo funciona como debe.

2.7. Redacción de la memoria

Esta fase se ha ido solapando con las demás ya que se ha realizado conjuntamente a las otras a medida que se iba desarrollando el proyecto.

2.8. Diagrama de Gantt

A continuación se muestra el diagrama de Gantt realizado con la herramienta Planner, en el que se puede comprobar los plazos utilizados para las fases del desarrollo del proyecto.

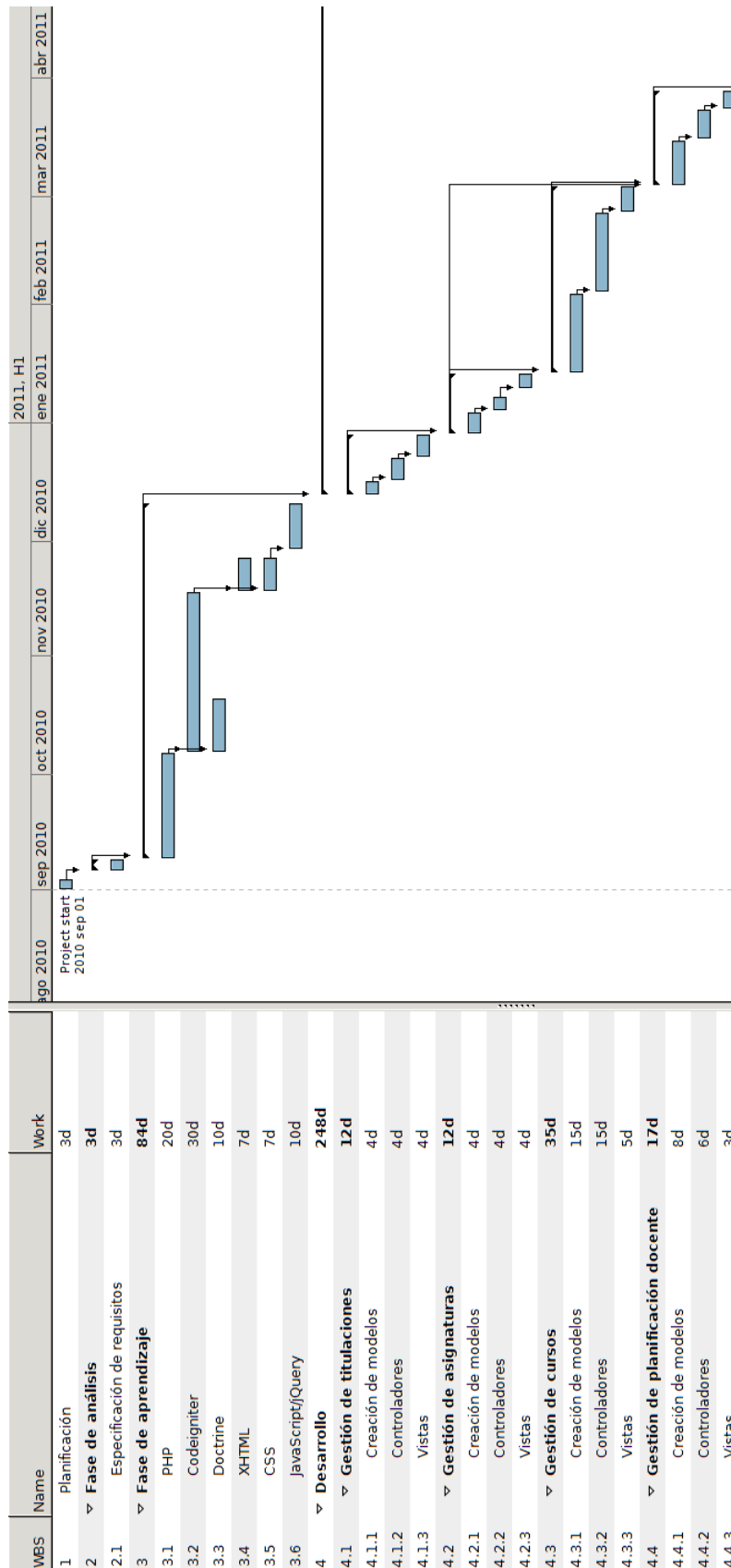


Figura 2.1: Diagrama de Gannt. Desarrollo del proyecto 1/2

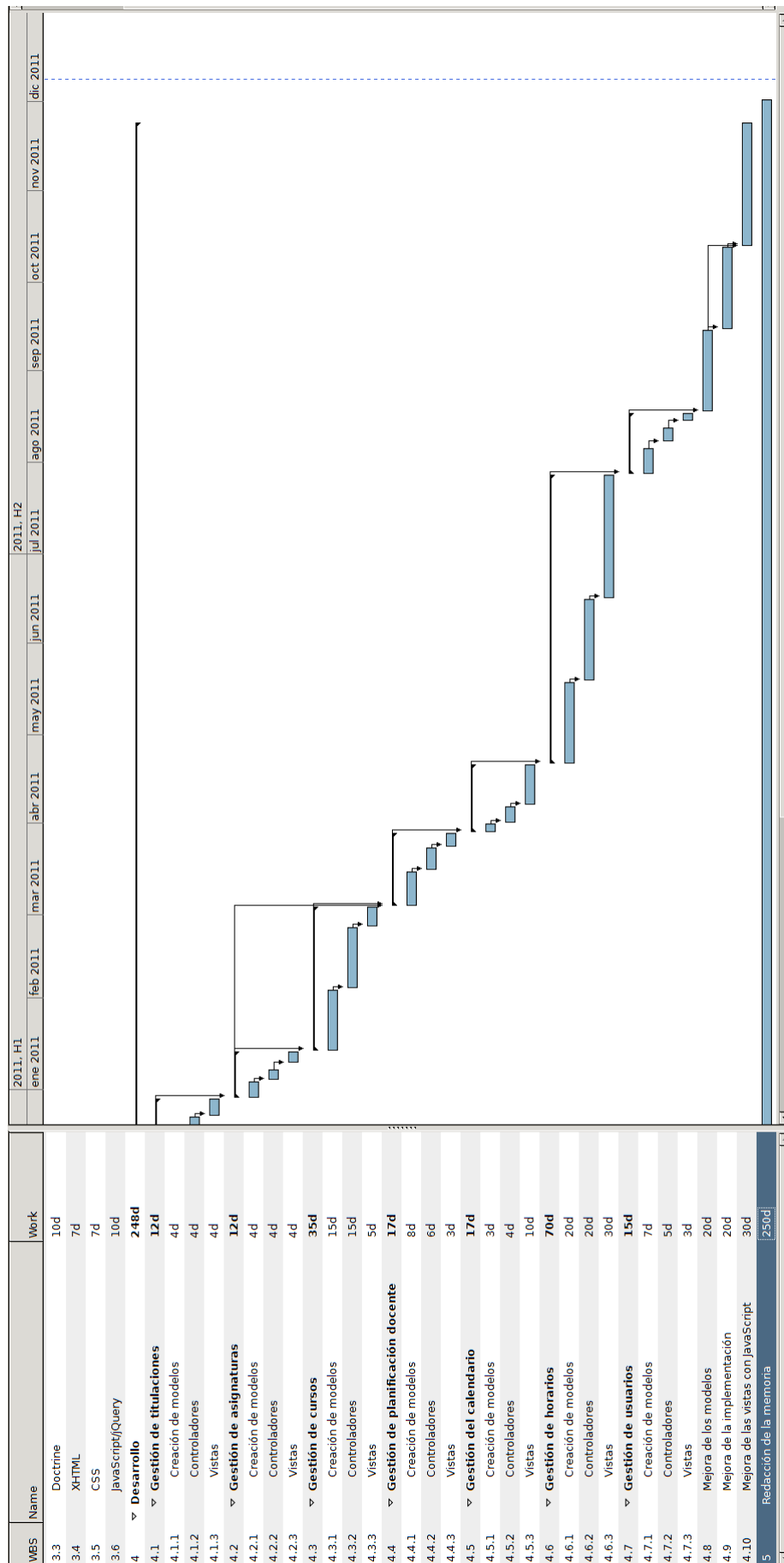


Figura 2.2: Diagrama de Gannt. Desarrollo del proyecto 1/2

Capítulo 3

Descripción general del proyecto

Este proyecto tiene la condición de *Software Libre*, por lo que en caso de necesitar ser ampliado, cualquier persona podría hacerlo. El proyecto es una aplicación nueva, no es continuación de otro proyecto.

3.1. Descripción

El proyecto consiste en una aplicación web, con distintos perfiles de usuario, en la que se llevará a cabo la configuración de la planificación docente de las distintas titulaciones de grado de la ESI. Como se ha dicho existirán varios perfiles de usuario, cada uno tendrá un cometido.

3.2. Perfiles de usuario

A continuación se expondrán los diferentes perfiles detallando a que funcionalidad tendrá acceso cada uno.

3.2.1. Perfil Administrador

El administrador solo tendrá acceso a la gestión de usuarios y a la creación de copias de seguridad de la base de datos. Por tanto el administrador podra crear nuevos usuarios con los perfiles que considere necesarios.

3.2.2. Perfil Planificador

Es el perfil que tiene acceso a más funcionalidades del sistema, llevará a cabo la gestión de titulaciones, asignaturas, planificación docente, calendario, aulas y horarios, realizando la configuración de todo. Es el perfil principal de la aplicación.

3.2.3. Perfil Profesor

Únicamente tendrá acceso a la visualización de la planificación docente de una titulación concreta.

3.2.4. Perfil Alumno

El alumno solo tendrá acceso a la visualización de los horarios. Podrá configurar un horario con las asignaturas y grupos pertenecientes a su titulación, generando un horario únicamente con las asignaturas que el quiera consultar.

3.3. Interfaz de usuario

La interfaz será algo simple, visualizada en un navegador web, con un menú principal en el que se tendrá acceso a las diferentes funcionalidades, estando ocultas las que no pertenezcan al perfil del usuario.

3.4. Software

Al ser una aplicación web, ésta será multiplataforma, pudiendo funcionar sobre cualquier navegador actual, ya que cumple los estándares de la W3C.

Como lenguaje de servidor la aplicación utiliza PHP, se toma la decisión de utilizarlo por la amplia documentación que hay disponible, además de la multitud de librerías que existen para simplificar su utilización. Además se ha utilizado el framework MVC CodeIgniter, que simplifica muchas tareas que de implementarlas únicamente con PHP sin la ayuda de ninguna librería se harían muy tediosas.

Para las vistas se ha utilizado XHTML y CSS, por su facilidad para estructurar los documentos y darles un estilo adecuado.

En la parte de los datos se ha usado MySQL como SGBD, utilizando Doctrine como un ORM para abstraer el uso de la base de datos dentro de la aplicación.

Capítulo 4

Análisis

4.1. Metodología de desarrollo

Para la realización del proyecto y su documentación se ha utilizado el *Rational Unified Process (RUP)*, junto con el *Lenguaje Unificado de Modelado (UML)*. Se ha elegido este sistema ya que es la metodología estándar más utilizada, además de ser un grupo de metodologías que se adaptan muy bien a las necesidades de un producto.

4.2. Especificación de requisitos del sistema

A continuación se enumeran los requisitos funcionales que se consideran fundamentales para el sistema. Éstos serán detallados utilizando casos de uso, describiendo tanto su escenario principal como sus posibles flujos alternativos. Además se detallará cada caso de uso con su diagrama de secuencia correspondiente.

4.2.1. Gestión de usuarios

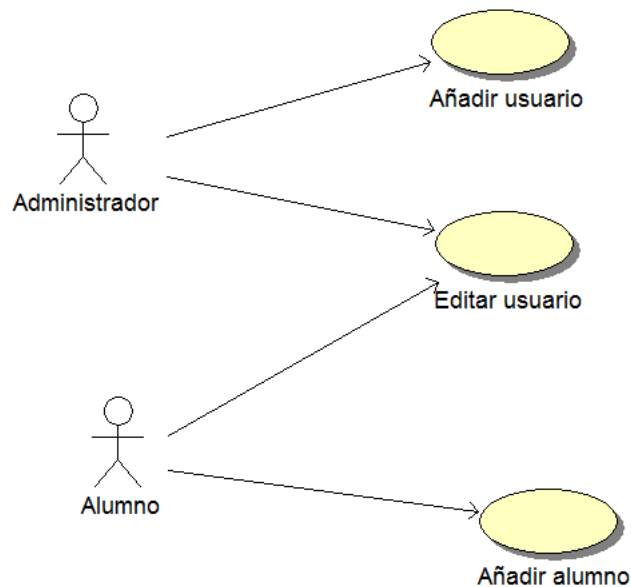


Figura 4.1: Diagrama de casos de uso de la gestión de usuarios

Caso de uso: Añadir usuario

- **Descripción:** Caso de uso para la creación de un usuario.
- **Actores:** Administrador
- **Precondiciones:** El administrador se ha identificado correctamente en el sistema.
- **Postcondiciones:** Se crea un usuario con el perfil correspondiente.
- **Escenario principal:**
 1. El administrador introduce los datos del usuario y el nivel de privilegios.
 2. El sistema valida que los datos son correctos y no hay ningún usuario con el mismo email y DNI
 3. El sistema crea el usuario y envía por correo el password al usuario.
- **Escenarios alternativos:**
 - 2.a. Alguno de los datos no es correcto.
 1. El sistema indica el error y el caso de uso vuelve al paso anterior.
 - 2.b. Ya existe algún usuario con el mismo email o DNI.
 1. El sistema indica el error y el caso de uso vuelve al paso anterior.

*.a. En cualquier momento el administrador decide cancelar el proceso.

1. El caso de uso finaliza.

Caso de uso: Registro de alumno

- **Descripción:** Caso de uso para la creación de un alumno.
- **Actores:** Alumno.
- **Precondiciones:** No hay ningún usuario identificado en el sistema.
- **Postcondiciones:** Se crea un usuario con el perfil de alumno.
- **Escenario principal:**
 1. El alumno introduce sus datos, DNI, titulación y email.
 2. El sistema valida que los datos son correctos y no hay ningún usuario con el mismo email y DNI.
 3. El sistema crea el usuario y envía por correo el password al usuario
- **Escenarios alternativos:**
 - 2.a. Alguno de los datos no es correcto.
 1. El sistema indica el error y el caso de uso vuelve al paso anterior.
 - 2.b. Ya existe algún usuario con el mismo email o DNI.
 1. El sistema indica el error y el caso de uso vuelve al paso anterior.
- *.a. En cualquier momento el alumno decide cancelar el proceso.
 1. El caso de uso finaliza.

Caso de uso: Editar usuario

- **Descripción:** Caso de uso para la edición de un usuario.
- **Actores:** Usuario.
- **Precondiciones:** El usuario que se intenta editar coincide con el identificado en el sistema o bien el usuario identificado es un administrador.
- **Postcondiciones:** Se actualizan los datos del usuario.
- **Escenario principal:**
 1. El sistema muestra los datos actuales del usuario.
 2. El usuario modifica sus datos, DNI, titulación y email.
 3. El sistema valida que los datos son correctos y no hay ningún usuario con el mismo email y DNI.
 4. El sistema modifica el usuario.
- **Escenarios alternativos:**
 - 2.a. Alguno de los datos no es correcto.
 1. El sistema indica el error y el caso de uso vuelve al paso anterior.

2.b. Ya existe algún usuario con el mismo email o DNI.

1. El sistema indica el error y el caso de uso vuelve al paso anterior.

*.a. En cualquier momento el usuario decide cancelar el proceso.

1. El caso de uso finaliza.

4.2.2. Gestión de titulaciones

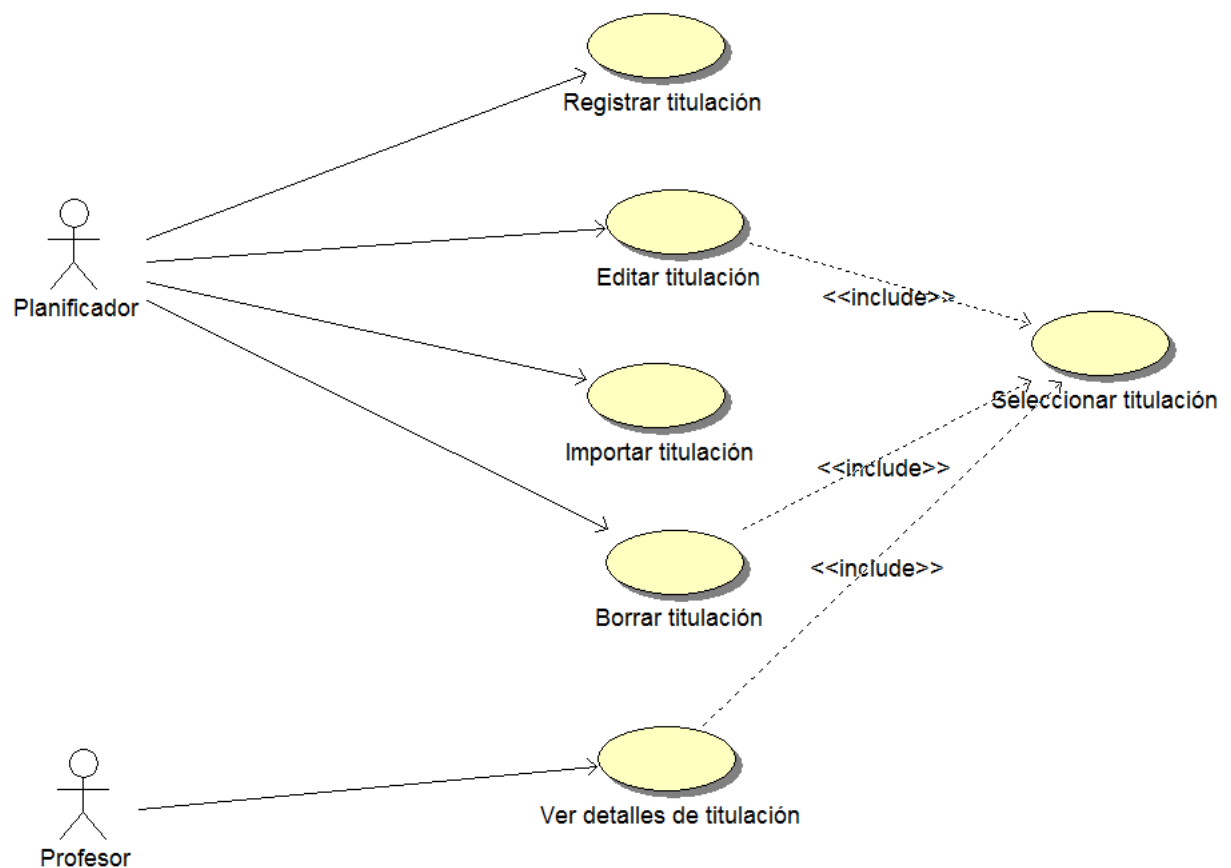


Figura 4.2: Diagrama de casos de uso de la gestión de titulaciones

Caso de uso: Seleccionar titulación

- **Descripción:** Caso de uso abstracto incluido en otros casos de uso para seleccionar una titulación de una lista de disponibles.
- **Actores:** Planificador.
- **Precondiciones:** El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se selecciona una titulación para su uso en otra finalidad.
- **Escenario principal:**
 1. El sistema muestra un listado de las titulaciones disponibles.

2. El usuario selecciona la titulación deseada.

■ **Escenarios alternativos:**

1.a. No hay ninguna titulación registrada.

1. El sistema indica el error y el caso de uso finaliza.

Caso de uso: Registrar titulación

■ **Descripción:** Registra una nueva titulación en el sistema.

■ **Actores:** Planificador.

■ **Precondiciones:** El usuario identificado en el sistema es un planificador.

■ **Postcondiciones:** La titulación queda registrada.

■ **Escenario principal:**

1. El planificador introduce el código, el nombre y todos los demás datos de la titulación.

2. El sistema comprueba que los datos cumplen el formato.

3. El sistema confirma el alta de la titulación mostrando un mensaje.

■ **Escenarios alternativos:**

2.a. Alguno de los datos introducidos tiene un formato incorrecto.

1. El sistema lo indica mostrando un mensaje de error y se vuelve al paso anterior.

2.b. Falta algún campo obligatorio.

1. El sistema lo indica mostrando un mensaje de error y se vuelve al paso anterior.

2.c. Ya existe alguna titulación con ese código o nombre.

1. El sistema indica el error y se vuelve al paso anterior.

*a. El planificador decide cancelar el registro en cualquier momento, el caso de uso termina.

Caso de uso: Editar titulación

■ **Descripción:** Edita una titulación existente en el sistema modificando sus datos.

■ **Actores:** Planificador.

■ **Precondiciones:** El usuario identificado en el sistema es un planificador.

■ **Postcondiciones:** La titulación queda modificada en el sistema.

■ **Escenario principal:**

1. Se realiza el caso de uso *Seleccionar titulación*.

2. El sistema muestra sus datos actuales, permitiendo su edición.

3. El planificador modifica los datos.

4. El sistema comprueba que todos los datos son correctos.

5. El sistema muestra un mensaje indicando que la edición se ha completado.

■ **Escenarios alternativos:**

- 4.a. Alguno de los datos tiene un formato incorrecto.
 - 1. El sistema muestra un mensaje de error indicándolo, a continuación se vuelve al paso anterior.
- 4.b. Falta algún campo obligatorio por rellenar.
 - 1. El sistema muestra un mensaje de error indicándolo, a continuación se vuelve al paso anterior.
- 4.c. Ya existe alguna titulación con el nombre o código introducidos.
 - 1. El sistema indica el error y se vuelve al paso anterior.
- *a. En cualquier momento el planificador decide cancelar la edición, el caso de uso se da por terminado.

Caso de uso: Borrar titulación

- **Descripción:** Borra una titulación del sistema.
- **Actores:** Planificador.
- **Precondiciones:** La titulación existe en el sistema. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** La titulación queda eliminada del sistema.
- **Escenario principal:**
 - 1. Se realiza el caso de uso *Seleccionar titulación*.
 - 2. El sistema muestra un diálogo de confirmación.
 - 3. El planificador confirma que quiere eliminar la titulación del sistema.
 - 4. El sistema elimina la titulación.
 - 5. El sistema muestra un mensaje confirmando que se ha eliminado la titulación.
- **Escenarios alternativos:**
 - 3.a. El planificador selecciona que no desea eliminar la titulación.
 - 1. El caso de uso se reinicia.
 - *a. En cualquier momento el planificador decide cancelar la eliminación.
 - 1. El caso de uso se termina.

Caso de uso: Ver detalles de titulación

- **Descripción:** Muestra los datos de una titulación en detalle, así como sus asignaturas.
- **Actores:** Planificador o profesor.
- **Precondiciones:** La titulación existe en el sistema. El usuario identificado en el sistema es un planificador o un profesor.
- **Postcondiciones:** Los datos de la titulación se muestran por pantalla.
- **Escenario principal:**
 - 1. Se realiza el caso de uso *Seleccionar titulación*.

2. El sistema muestra los datos de la titulación y un listado de sus asignaturas si las tiene, junto con su planificación docente.

- **Escenarios alternativos:**

- *a. En cualquier momento el usuario decide cancelar el proceso, el caso de uso se termina.

Caso de uso: Importar titulación

- **Descripción:** Caso de uso para importar titulaciones de forma masiva desde un archivo csv con un formato concreto.

- **Actores:** Planificador.

- **Precondiciones:** El archivo tiene el formato correcto. El usuario identificado en el sistema es un planificador.

- **Postcondiciones:** Se crean las titulaciones indicadas en el archivo.

- **Escenario principal:**

1. El planificador selecciona el archivo.
2. El sistema comprueba que cada línea tenga el formato correcto.
3. El sistema crea una titulación por cada línea con los datos indicados en el archivo.

- **Escenarios alternativos:**

- 2.a. alguna línea no cumple el formato

- 1. El sistema indica el error y el caso de uso finaliza.

- 2.b. Ya existe una titulación creada con el mismo identificador.

- 1. El sistema lo indica y el caso de uso finaliza.

Gestión de asignaturas

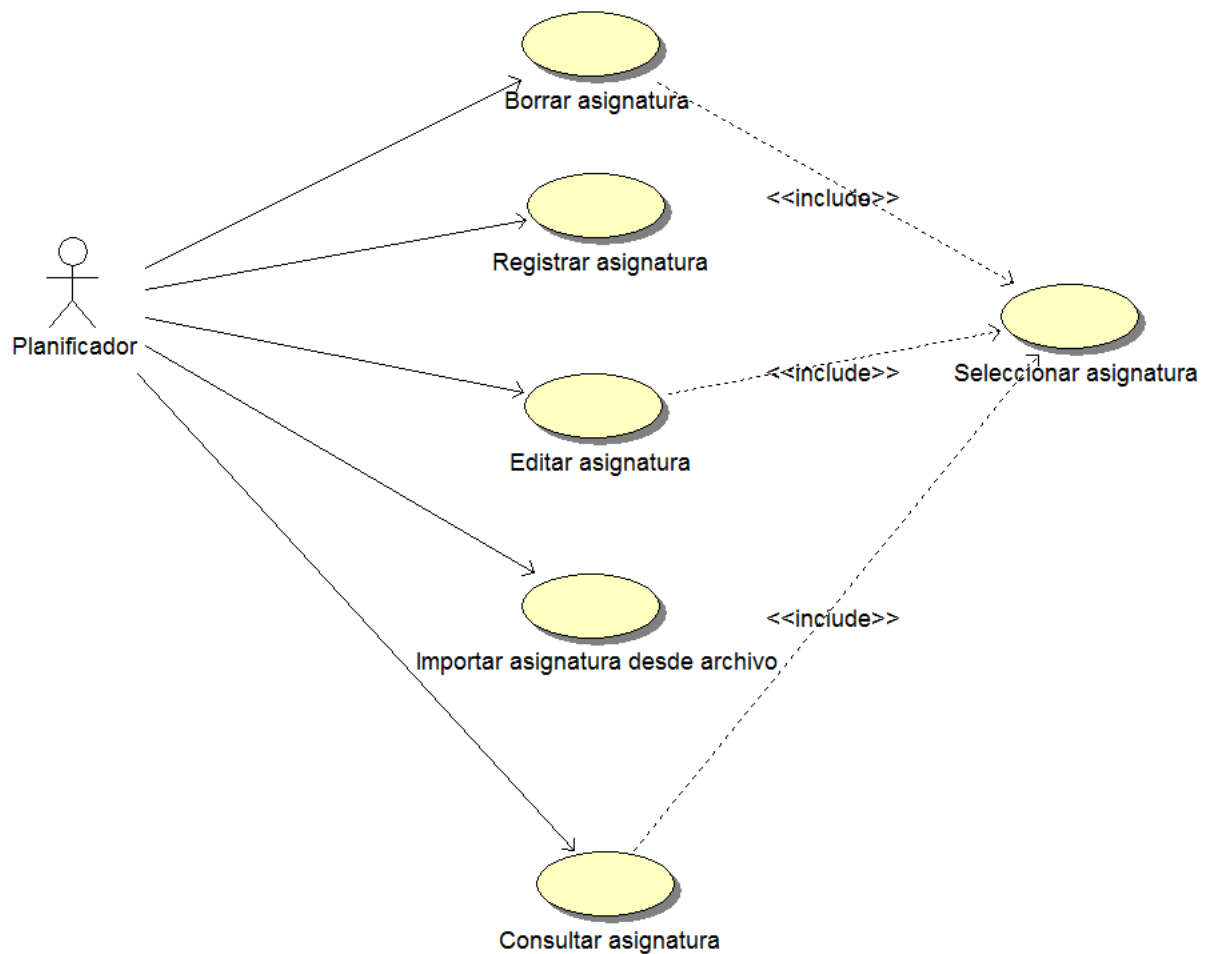


Figura 4.3: Diagrama de casos de uso de la gestión de asignaturas

Caso de uso: Seleccionar asignatura

- **Descripción:** Caso de uso abstracto incluido por otros casos de uso para seleccionar una asignatura del listado de las que tiene disponibles una titulación concreta.
- **Actores:** Usuario.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** Queda seleccionada una asignatura para algún fin concreto de otro caso de uso.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar titulación*.
 2. El sistema muestra un listado de las asignaturas disponibles asociadas a la titulación seleccionada.
 3. El usuario selecciona una asignatura de la lista.

- **Escenarios alternativos:**

- 2.a. No hay ninguna asignatura registrada en esa titulación.

- 1. El sistema indica el error y el caso de uso finaliza.

Caso de uso: Registrar asignatura

- **Descripción:** Se da de alta una nueva asignatura en el sistema.

- **Actores:** Planificador.

- **Precondiciones:** Existe alguna titulación con la que asociar la asignatura. El usuario identificado en el sistema es un planificador.

- **Postcondiciones:** La asignatura queda registrada en el sistema.

- **Escenario principal:**

- 1. Se realiza el caso de uso *Seleccionar titulación*.

- 2. El sistema muestra un formulario para introducir los datos.

- 3. El planificador introduce el código, el nombre y todos los demás datos de la asignatura.

- 4. El sistema comprueba que todos los datos cumplen el formato requerido.

- 5. El sistema registra la asignatura y muestra un mensaje confirmándolo.

- **Escenarios alternativos:**

- 3.a. El planificador selecciona que desea tomar los datos de otra asignatura.

- 1. Se realiza el caso de uso *Duplicar asignatura*.

- 4.a. Alguno de los datos no cumple el formato correcto.

- 1. El sistema indica el error y se vuelve al paso anterior.

- 4.b. Falta por rellenar algún campo obligatorio.

- 1. El sistema indica el error y se vuelve al paso anterior.

- 4.c. Ya existe alguna asignatura con ese código o nombre.

- 1. El sistema indica el error y se vuelve al paso anterior.

- *a. En cualquier momento el planificador decide cancelar el proceso.

- 1. El caso de uso se cancela.

Caso de uso: Editar Asignatura

- **Descripción:** Se modifican los datos de una asignatura existente en el sistema.

- **Actores:** Planificador.

- **Precondiciones:** El usuario identificado en el sistema es un planificador.

- **Postcondiciones:** La asignatura queda modificada en el sistema.

- **Escenario principal:**

- 1. Se realiza el caso de uso *Seleccionar asignatura*.

2. El sistema muestra los datos de la asignatura en un formato editable.
3. El planificador hace las modificaciones que considere necesarias.
4. El sistema comprueba que los datos modificados cumplen el formato requerido.
5. El sistema guarda la asignatura y muestra un mensaje confirmándolo.

■ **Escenarios alternativos:**

- 4.a. Alguno de los datos no cumple el formato correcto.
 1. El sistema indica el error y se vuelve al paso anterior.
- 4.b. Falta por rellenar algún campo obligatorio.
 1. El sistema indica el error y se vuelve al paso anterior.
- 4.c. Ya existe alguna asignatura con ese código o nombre.
 1. El sistema indica el error y se vuelve al paso anterior.
- *a. En cualquier momento el planificador decide cancelar el proceso.
 1. El caso de uso se cancela.

Caso de uso: Borrar asignatura

- **Descripción:** Se borra una asignatura del sistema.
- **Actores:** Planificador.
- **Precondiciones:** El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** La asignatura queda eliminada del sistema.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar asignatura*.
 2. El sistema muestra un diálogo de confirmación.
 3. El planificador confirma que desea borrar la asignatura.
 4. El sistema borra la asignatura y muestra un mensaje confirmándolo.
- **Escenarios alternativos:**
 - 3.a. El planificador selecciona que no desea eliminar la asignatura.
 1. El caso de uso se reinicia.
 - *a. En cualquier momento el planificador decide cancelar la eliminación.
 1. El caso de uso se termina.

Caso de uso: Consultar asignatura

- **Descripción:** Muestra los datos en detalle de una asignatura.
- **Actores:** Usuario.
- **Precondiciones:** El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se muestran los datos de la asignatura por pantalla.

- **Escenario principal:**

1. Se realiza el caso de uso *Seleccionar asignatura*.
2. El sistema muestra la información relacionada con la asignatura.

- **Escenarios alternativos:**

- *a. En cualquier momento el planificador decide cancelar el proceso.
 1. El caso de uso se termina.

Caso de uso: Importar asignatura desde archivo

- **Descripción:** Caso de uso para importar asignaturas de forma masiva desde un archivo csv con un formato concreto.

- **Actores:** planificador

- **Precondiciones:** El archivo tiene el formato correcto.

- **Postcondiciones:** Se crean las asignaturas indicadas en el archivo.

- **Escenario principal:**

1. El planificador selecciona el archivo.
2. El sistema comprueba que cada línea tenga el formato correcto.
3. El sistema crea una asignatura por cada línea con los datos indicados en el archivo.

- **Escenarios alternativos:**

- 2.a. Alguna línea no cumple el formato

1. El sistema indica el error y el caso de uso finaliza.

- 2.b. Ya existe una asignatura creada con el mismo identificador.

1. El sistema lo indica y el caso de uso finaliza.

Gestión de cursos

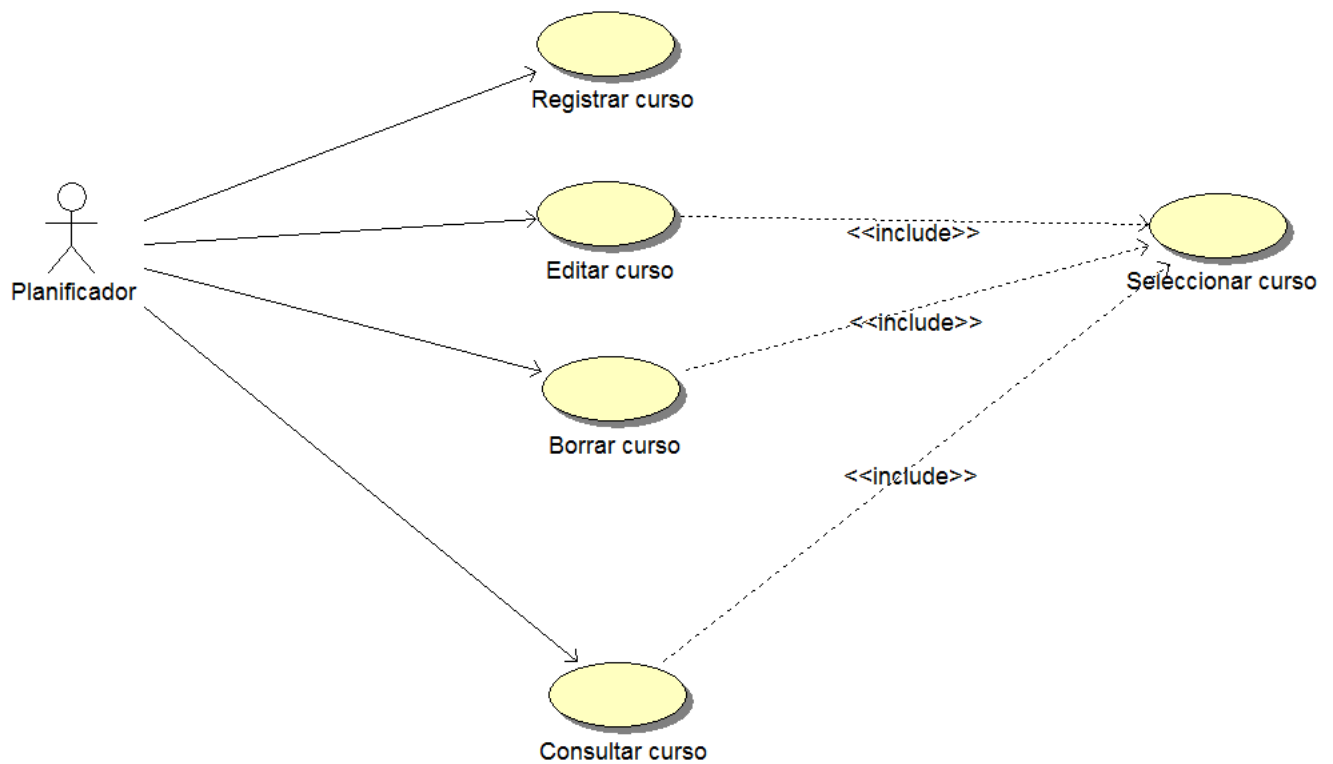


Figura 4.4: Diagrama de casos de uso de la gestión de cursos

Caso de uso: Seleccionar curso

- **Descripción:** Caso de uso abstracto que es incluido en otros casos de uso.
- **Actores:** Usuario.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** Queda seleccionado un curso para su uso con algún fin.
- **Escenario principal:**
 1. El sistema muestra un listado con los cursos disponibles.
 2. El usuario selecciona un curso.
- **Escenarios alternativos:**
 - 1.a. No hay ningún curso registrado en el sistema.
 1. El sistema indica el error y el caso de uso finaliza.

Caso de uso: Registrar curso

- **Descripción:** Se da de alta en el sistema la configuración de un nuevo curso.
- **Actores:** Planificador.

- **Precondiciones:** El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Queda registrado en el sistema el curso.
- **Escenario principal:**
 1. El planificador introduce los datos de configuración del curso, incluyendo fecha de inicio y final de curso.
 2. El sistema comprueba que los datos sean correctos.
 3. El sistema informa de que el curso ha sido registrado con éxito.
- **Escenarios alternativos:**
 - 2.a. Alguno de los datos introducidos no cumple el formato correcto.
 1. El sistema indica el error y se vuelve al paso anterior.
 - 2.b. Ya existe un curso registrado que empieza o termina en el mismo año que el introducido.
 1. El sistema indica el error y se vuelve al paso anterior.
 - 2.c. Los años introducidos de inicio y fin del curso no son consecutivos.
 1. El sistema indica el error y se vuelve al paso anterior.
 - 2.d. alguna de las fechas de exámenes introducidas no están comprendidas entre la duración del curso.
 1. El sistema indica el error y se vuelve al paso anterior.
 - *a. En cualquier momento el planificador decide cancelar el proceso.
 1. El caso de uso finaliza.

Caso de uso: Editar curso

- **Descripción:** Se edita en el sistema la configuración de un curso.
- **Actores:** Planificador.
- **Precondiciones:** El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Quedan registradas en el sistema las modificaciones realizadas.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar curso*.
 2. El sistema muestra los datos en forma editable.
 3. El planificador modifica los datos de configuración del curso que considere necesarios.
 4. El sistema comprueba que los datos sean correctos.
 5. El sistema informa de que el curso ha sido registrado con éxito.
- **Escenarios alternativos:**
 - 4.a. Alguno de los datos introducidos no cumple el formato correcto.
 1. El sistema indica el error y se vuelve al paso anterior.
 - 4.b. Ya existe un curso registrado que empieza o termina en el mismo año que el introducido.
 1. El sistema indica el error y se vuelve al paso anterior.

- 4.c. Los años introducidos de inicio y fin del curso no son consecutivos.
 - 1. El sistema indica el error y se vuelve al paso anterior.
- 4.d. alguna de las fechas de exámenes introducidas no están comprendidas entre la duración del curso.
 - 1. El sistema indica el error y se vuelve al paso anterior.
- *a. En cualquier momento el planificador decide cancelar el proceso.
 - 1. El caso de uso finaliza.

Caso de uso: Consultar curso

- **Descripción:** Se consulta la configuración de un curso mostrándola por pantalla
- **Actores:** Planificador.
- **Precondiciones:** El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se muestra por pantalla la configuración del curso seleccionado.
- **Escenario principal:**
 - 1. Se realiza el caso de uso *Seleccionar curso*.
 - 2. El sistema muestra los datos del curso.
- **Escenarios alternativos:**
 - *a. En cualquier momento el planificador decide cancelar el proceso.
 - 1. El caso de uso finaliza.

Caso de uso: Borrar curso

- **Descripción:** Se elimina un curso registrado en el sistema.
- **Actores:** Planificador
- **Precondiciones:** El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** El curso seleccionado queda eliminado del sistema
- **Escenario principal:**
 - 1. Se realiza el caso de uso *Seleccionar curso*.
 - 2. El sistema muestra un diálogo pidiendo la confirmación del borrado.
 - 3. El planificador selecciona que desea confirmar el borrado.
 - 4. El sistema muestra un mensaje confirmando el éxito en la operación.
- **Escenarios alternativos:**
 - 3.a. El planificador selecciona que no desea confirmar el borrado.
 - 1. El caso de uso se reinicia.
 - *a. En cualquier momento el planificador cancela el proceso.
 - 1. El caso de uso finaliza.

Gestión de planificación docente

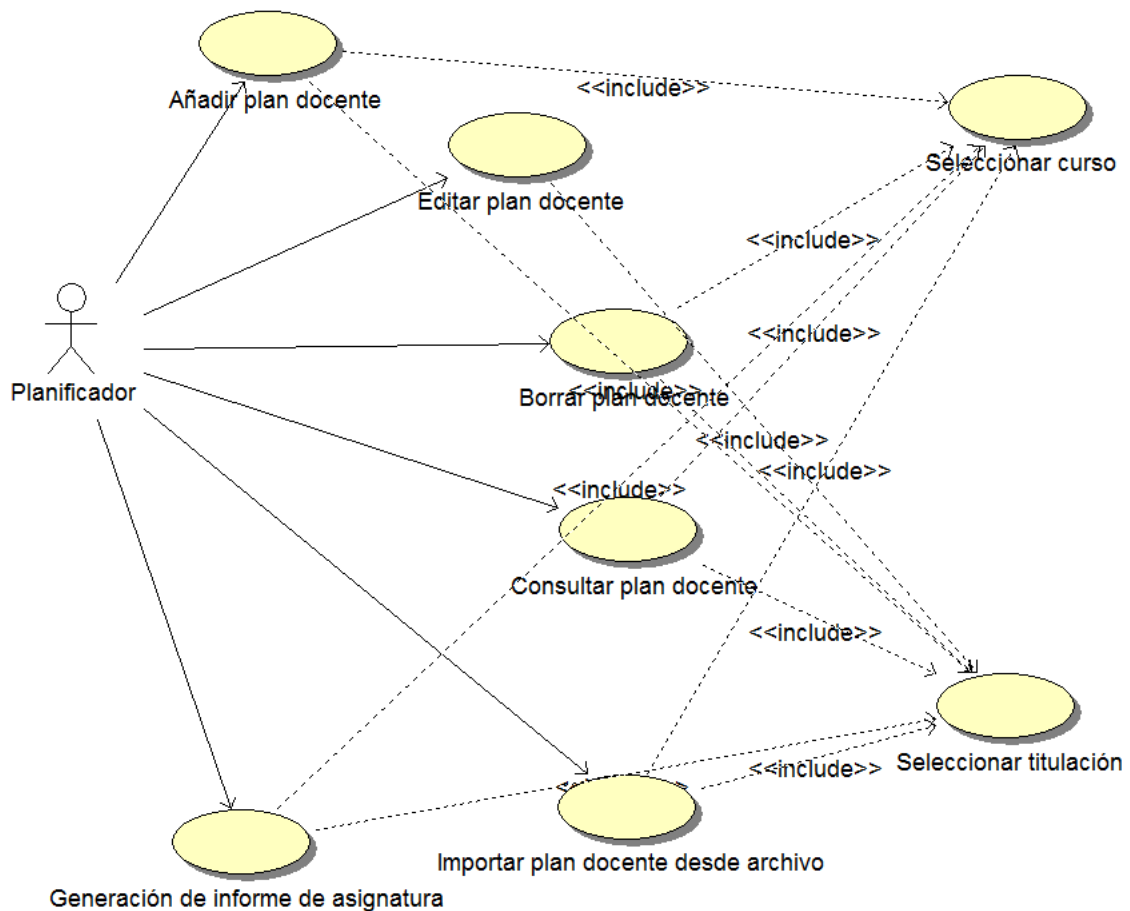


Figura 4.5: Diagrama de casos de uso de la gestión de la planificación docente

Caso de uso: Añadir plan docente

- **Descripción:** Se añaden los detalles del plan docente para una asignatura en un curso determinado.
- **Actores:** Planificador.
- **Precondiciones:** El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** El plan docente queda registrado en el sistema, asociado a una asignatura y un curso determinado.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar asignatura*.
 2. Se realiza el caso de uso *Seleccionar curso*.
 3. El sistema comprueba que no exista ya un plan docente asociado a ese curso.
 4. El usuario introduce los datos del plan docente.
 5. El sistema comprueba que los datos cumplen el formato requerido.

6. El sistema guarda la carga de trabajo y muestra un mensaje confirmándolo.

■ **Escenarios alternativos:**

3.a. Ya existe una carga de trabajo establecida para el curso seleccionado.

1. El sistema indica el error y el caso de uso vuelve al paso anterior.

4.a. El usuario indica que quiere tomar los datos de otra carga de un curso anterior.

1. Se realiza el caso de uso *Duplicar carga de trabajo*

5.a. Alguno de los datos introducidos no cumple el formato correcto.

1. El sistema indica el error y se vuelve al paso anterior.

5.b. Alguno de los campos obligatorios no ha sido rellenado.

1. El sistema indica el error y se vuelve al paso anterior.

*a. En cualquier momento el planificador decide cancelar el proceso.

1. El caso de uso se termina.

Caso de uso: Editar plan docente

■ **Descripción:** Se edita una carga de trabajo existente para un curso determinado.

■ **Actores:** Planificador.

■ **Precondiciones:** El usuario identificado en el sistema es un planificador.

■ **Postcondiciones:** La carga de trabajo queda modificada en el sistema.

■ **Pasos:**

1. Se realiza el caso de uso *Seleccionar asignatura*.

2. Se realiza el caso de uso *Seleccionar curso*.

3. El sistema comprueba que exista una carga asociada a ese curso.

4. El sistema muestra los datos de la carga en un formato editable.

5. El usuario modifica los datos.

6. El sistema comprueba que los datos cumplen el formato requerido.

7. El sistema guarda los cambios y muestra un mensaje confirmándolo.

■ **Escenarios alternativos:**

3.a. No existe una carga de trabajo establecida para el curso seleccionado.

1. El sistema indica el error y el caso de uso vuelve al paso anterior.

5.a. Alguno de los datos introducidos no cumple el formato correcto.

1. El sistema indica el error y se vuelve al paso anterior.

5.b. Alguno de los campos obligatorios no ha sido rellenado.

1. El sistema indica el error y se vuelve al paso anterior.

*a. En cualquier momento el planificador decide cancelar el proceso.

1. El caso de uso se termina.

Caso de uso: Borrar plan docente

- **Descripción:** Se borra una carga de trabajo existente en el sistema asociada a un curso.
- **Actores:** Planificador.
- **Precondiciones:** El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** La carga de trabajo asociada a la asignatura y curso seleccionados queda eliminada del sistema.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar asignatura*
 2. Se realiza el caso de uso *Seleccionar curso*.
 3. El sistema comprueba que exista una carga asociada a ese curso.
 4. El sistema muestra un diálogo de confirmación.
 5. El usuario confirma que desea borrar la carga.
 6. El sistema muestra un mensaje confirmando la eliminación y borra la carga.
- **Escenarios alternativos:**
 - 3.a. No existe una carga de trabajo establecida para el curso seleccionado.
 1. El sistema indica el error y el caso de uso vuelve al paso anterior.
 - 5.a. El administrador selecciona que no desea eliminar la carga.
 1. El caso de uso se reinicia.
 - *a. En cualquier momento el administrador decide cancelar la eliminación.
 1. El caso de uso se termina.

Caso de uso: Consultar plan docente

- **Descripción:** Se consulta la carga de trabajo de una asignatura para un curso determinado.
- **Actores:** Planificador.
- **Precondiciones:** El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se muestran los datos al usuario por pantalla.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar asignatura*
 2. Se realiza el caso de uso *Seleccionar curso*.
 3. El sistema comprueba que exista una carga asociada a ese curso.
 4. La carga existe y es mostrada al usuario.
- **Escenarios alternativos:**
 - 3.a. No existe ninguna carga asociada a ese curso.
 1. El sistema muestra un mensaje informando del error y vuelve al paso anterior.
 - *a. En cualquier momento el usuario decide cancelar el proceso.
 1. El caso de uso se cancela.

Caso de uso: Importar plan docente desde archivo

- **Descripción:** Caso de uso para importar planes docentes de forma masiva desde un archivo csv con un formato concreto.
- **Actores:** Planificador.
- **Precondiciones:** El archivo tiene el formato correcto. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se crean los planes docentes indicados en el archivo.
- **Escenario principal:**
 1. El administrador selecciona el archivo.
 2. El sistema comprueba que cada línea tenga el formato correcto.
 3. El sistema crea un plan docente por cada línea con los datos indicados en el archivo.
- **Escenarios alternativos:**
 - 2.a. Alguna línea no cumple el formato
 1. El sistema indica el error y el caso de uso finaliza.
 - 2.b. Ya existe un plan docente creado para la asignatura indicada y para ese curso.
 1. El sistema lo indica y el caso de uso finaliza.

Caso de uso: Generación de informe de asignatura

- **Descripción:** Caso de uso para generar un informe de las horas asignadas a una asignatura.
- **Actores:** Planificador.
- **Precondiciones:** La asignatura tiene un plan docente creado y asignadas horas en los horarios. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se genera un informe en pdf permitiendo su descarga.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar curso*.
 2. Se realiza el caso de uso *Seleccionar asignatura*.
 3. El sistema comprueba el plan docente de la asignatura y las asignaciones en los horarios.
 4. El sistema muestra un desglose de las horas de cada actividad de la asignatura y de cada semana teniendo en cuenta los eventos del calendario, permitiendo la descarga del archivo.
- **Escenarios alternativos:**
 - *.a. En cualquier momento se decide parar el proceso.
 1. El caso de uso finaliza.
 - 3.a La asignatura no tiene un plan docente asignado.
 1. El caso de uso finaliza.

Gestión del calendario

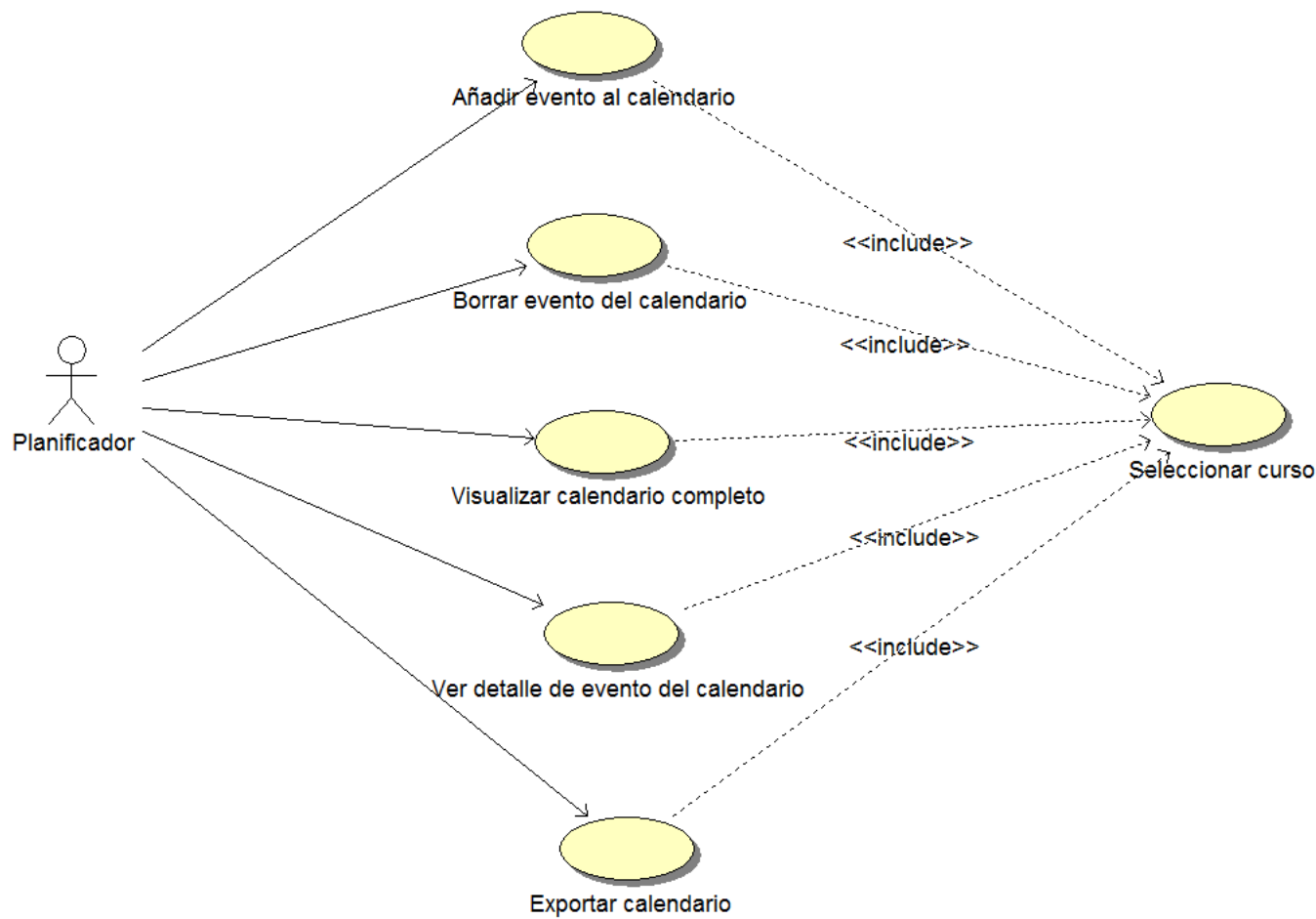


Figura 4.6: Diagrama de casos de uso de la gestión del calendario

Caso de uso: Añadir evento al calendario

- **Descripción:** Añade un evento de fechas al calendario, que puede ser una festividad o un período de vacaciones.
- **Actores:** Planificador.
- **Precondiciones:** El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** El evento queda registrado para el calendario de un curso concreto.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar curso*.
 2. El subdirector introduce los datos correspondientes al evento, que serían, el nombre del evento, el tipo de evento, si es una fecha concreta o un rango de ellas, y su fecha de inicio y finalización.
 3. El sistema comprueba que los datos son correctos y que no se solapen con otros eventos.

4. El sistema indica que todo es correcto y muestra un mensaje confirmando el éxito de la operación.

■ **Escenarios alternativos:**

- 3.a. Alguno de los datos tiene un formato incorrecto o algún campo está en blanco.
 1. El sistema lo indica y vuelve al paso anterior.
- 3.b. Las fechas del evento se solapan con algún otro evento o fecha del curso.
 1. El sistema lo indica y vuelve al paso anterior.
- *a. En cualquier momento el usuario decide cancelar el proceso.
 1. El caso de uso finaliza.

Caso de uso: Borrar evento del calendario

- **Descripción:** El usuario selecciona alguno de los eventos del calendario para eliminarlo y este queda eliminado del sistema.
- **Actores:** Planificador.
- **Precondiciones:** El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** El evento queda eliminado del calendario del curso concreto.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar curso*.
 2. El sistema muestra un listado de fechas de eventos.
 3. El subdirector selecciona la fecha que desea eliminar.
 4. El sistema muestra un mensaje confirmando que el evento ha sido eliminado.
- **Escenarios alternativos:**
 - 2.a. No hay ningún evento registrado en el calendario de ese curso.
 1. El sistema lo indica y el caso de uso finaliza.
 - *a. En cualquier momento el subdirector decide cancelar el proceso.
 1. El caso de uso finaliza sin eliminar el evento.

Caso de uso: Visualizar calendario completo

- **Descripción:** Se muestra un calendario completo con las fechas marcadas.
- **Actores:** Planificador.
- **Precondiciones:** El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se muestra el calendario por pantalla.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar curso*.
 2. El sistema muestra un calendario con las fechas de los eventos marcadas.
- **Escenarios alternativos:**

Caso de uso: Ver detalle de evento del calendario

- **Descripción:** Se muestran los datos detallados de un evento del calendario
- **Actores:** Planificador.
- **Precondiciones:** Hay eventos registrados en el sistema. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se muestran los datos.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar curso*.
 2. El sistema muestra el calendario para el curso actual con los eventos creados sobre él.
 3. El usuario selecciona un evento
 4. El sistema muestra los datos del evento, título, razón, etc.
- **Escenarios alternativos:**

Caso de uso: Exportar calendario

- **Descripción:** Caso de uso para exportar el calendario de un curso a un formato externo como csv o una hoja de cálculo.
- **Actores:** Planificador.
- **Precondiciones:** El curso existe y tiene eventos creados. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se exporta un archivo con un formato adecuado.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar curso*.
 2. Se muestra y permite descarga del archivo exportado.
- **Escenarios alternativos:**
 - *.a. En cualquier momento el administrador decide cancelar el proceso.
 1. Se finaliza el caso de uso.

Gestión de horarios

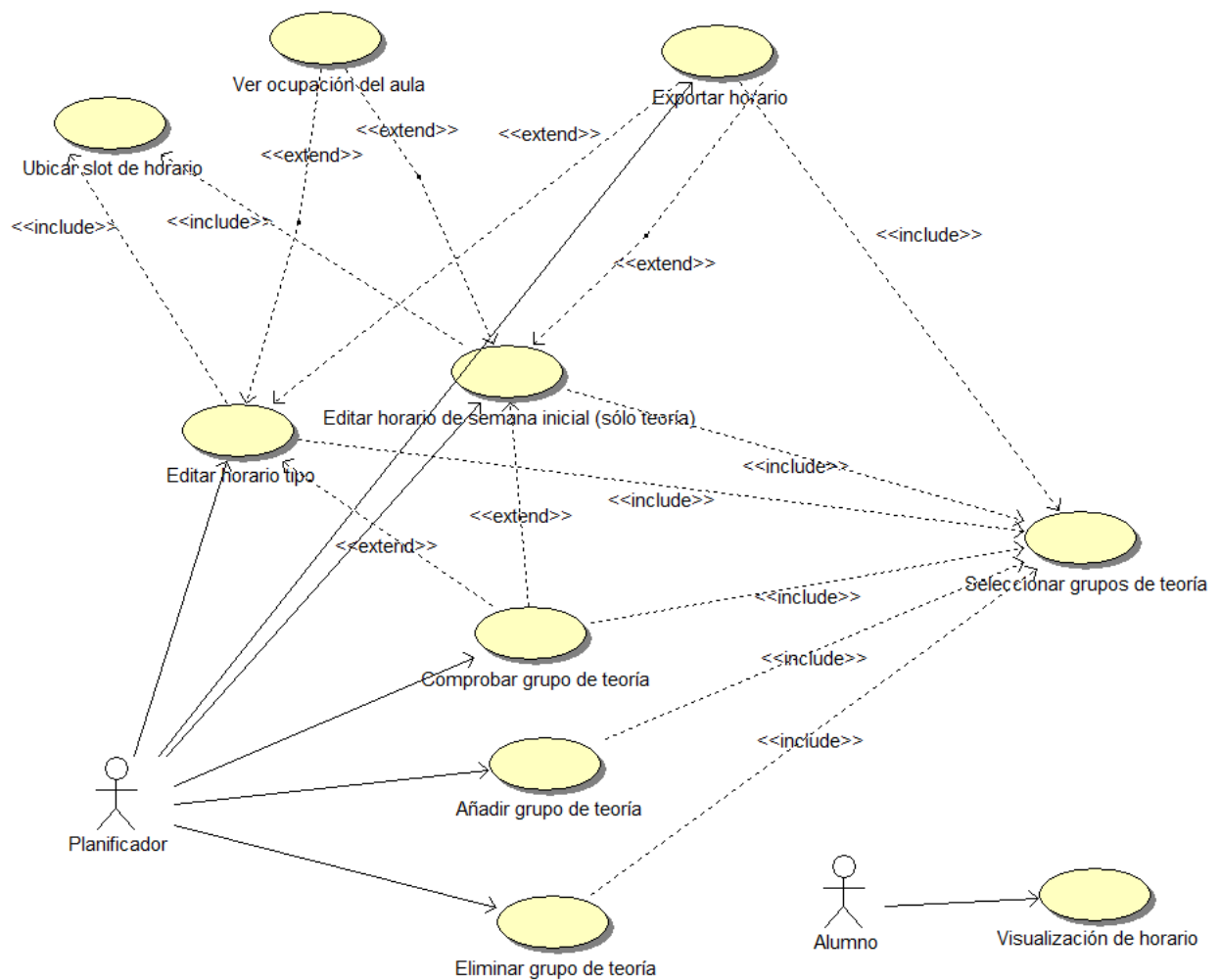


Figura 4.7: Diagrama de casos de uso de la gestión de horarios

Caso de uso: Seleccionar grupos de teoría

- **Descripción:** Se muestran los grupos de teoría de una titulación. Este caso de uso es incluido por más casos de uso.
- **Actores:** Planificador.
- **Precondiciones:** Hay titulaciones y cursos creados. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se muestra la información
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar curso*.
 2. Se realiza el caso de uso *Seleccionar titulación*.

3. El sistema muestra un listado con los cursos de la titulación y el número de grupos de cada curso.
4. El administrador selecciona el grupo deseado.

- **Escenarios alternativos:**

Caso de uso: Ubicar slot de horario

- **Descripción:** Caso de uso abstracto que es incluido por otros casos de uso. Se utiliza para ubicar en un horario un slot de una actividad de una asignatura.
- **Actores:** Planificador.
- **Precondiciones:** La asignatura existe y tiene asignadas horas en el plan docente para esa actividad. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** El slot de la actividad queda guardado en el horario.
- **Escenario principal:**
 1. El administrador selecciona una asignatura y su actividad
 2. El sistema comprueba que tenga asignadas horas en el plan docente
 3. El administrador selecciona el aula en la que se guardará el slot
 4. El administrador selecciona el lugar que ocupará en el horario
 5. El sistema comprueba que el aula no esté ocupada en ese momento y que si el slot es de teoría que no se solape con otros slots.
 6. El sistema guarda el slot en el horario.
- **Escenarios alternativos:**
 - 2.a. La asignatura no tiene horas asignadas para esa actividad.
 1. El sistema lo indica y se vuelve al paso anterior.
 - 5.a. El aula está ocupada en ese horario.
 1. El sistema lo indica y se vuelve al paso 3.
 - 5.b. El slot es de teoría y se solapa con otros slots.
 1. El sistema lo indica y se vuelve al paso anterior.

Caso de uso: Editar horario tipo

- **Descripción:** Se muestra el horario de un grupo de teoría en un formato editable.
- **Actores:** Planificador.
- **Precondiciones:** Hay algún grupo creado para la titulación, semestre, curso y año académico. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se muestran las asignaturas disponibles, permitiendo su edición. Se muestran tanto los slots de teoría como los de las demás actividades
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar grupos de teoría*.

2. El administrador selecciona el curso y semestre deseado para editar su horario tipo.
3. El sistema comprueba que exista el grupo y se que no exista un horario ya empezado.
4. Se realiza el caso de uso *Ubicar slot de horario*.
5. El administrador realiza el paso anterior las veces que sean necesarias hasta completar el horario.

■ **Escenarios alternativos:**

- *.a. En cualquier momento el administrador decide parar el proceso.
 1. El sistema guarda los cambios hechos hasta ahora y el caso de uso finaliza.
- 2.a No ha sido creado ningún grupo para ese curso.
 1. El sistema indica el error y finaliza el caso de uso.
- 3.a Ya hay un horario empezado para ese grupo y semestre
 1. El sistema muestra el horario ya empezado ubicando los slots donde estaban en el antiguo horario y el caso de uso continua.

Caso de uso: Editar horario de semana inicial (sólo teoría)

- **Descripción:** Se muestra para su edición el horario de un grupo de teoría en una semana en la que sólo se imparte teoría
- **Actores:** Planificador.
- **Precondiciones:** Ya ha sido creado el grupo y editado el horario tipo correspondiente. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se muestra el horario con los slots de teoría del horario tipo ubicados en su sitio, salvo los que no sean posible por no ser día lectivo

■ **Escenario principal:**

1. Se realiza el caso de uso *Seleccionar grupo de teoría*.
2. El administrador selecciona el curso y semestre además de la semana que se editará.
3. El sistema comprueba que exista el grupo y se que no exista un horario ya empezado.
4. El sistema muestra los slots de teoría del horario tipo correspondiente, sacando del horario los que no sean posibles ubicar automáticamente por que el día sea no lectivo.
5. Se realiza el caso de uso *Ubicar slot de horario*.
6. Se realiza el paso anterior las veces que se consideren necesarias hasta completar el horario.

■ **Escenarios alternativos:**

- *.a. En cualquier momento el administrador decide parar el proceso.
 1. El sistema guarda los cambios hechos hasta ahora y el caso de uso finaliza.
- 2.a No ha sido creado ningún grupo para ese curso.
 1. El sistema indica el error y finaliza el caso de uso.
- 3.a Ya hay un horario empezado para ese grupo, semestre y semana
 1. El sistema muestra el horario ya empezado ubicando los slots donde estaban en el antiguo horario y el caso de uso continua.

Caso de uso: Añadir grupo de teoría

- **Descripción:** Se añade un grupo de teoría a un curso de una titulación para poder editar más adelante su horario
- **Actores:** Planificador.
- **Precondiciones:** La titulación existe y tiene asignaturas con planes docentes creados para el curso actual. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** El grupo se añade al curso correspondiente.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar grupo de teoría*.
 2. El administrador selecciona el curso de la titulación al que se va a añadir el grupo.
 3. El sistema comprueba que no se haya sobrepasado el número de grupos indicado en el plan docente y guarda el grupo.
- **Escenarios alternativos:**
 - 2.a. Se han sobrepasado el número de grupos indicado en el plan docente.
 1. El sistema advierte del problema pero el caso de uso continúa.

Caso de uso: Eliminar grupo de teoría

- **Descripción:** Se elimina un grupo de teoría de un curso de una titulación
- **Actores:** Planificador.
- **Precondiciones:** Hay grupos creados para la titulación y curso. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se elimina el grupo así como toda su información asociada (horarios, informes, etc).
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar grupo de teoría*.
 2. El administrador selecciona el curso de la titulación al que se va a eliminar el grupo.
 3. El sistema pide confirmación de la eliminación.
 4. El administrador confirma que desea eliminar el grupo.
 5. Queda borrado el grupo y toda la información asociada y el sistema lo indica.
- **Escenarios alternativos:**
 - 4.a. El administrador cancela el proceso.
 1. El caso de uso finaliza.

Caso de uso: Exportar horario

- **Descripción:** Caso de uso para exportar un horario creado a un formato externo como csv o una hoja de cálculo.
- **Actores:** Planificador.
- **Precondiciones:** Existe un grupo creado y está creado el horario. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se exporta un archivo con un formato adecuado.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar grupo de teoría*.
 2. El administrador selecciona la semana, semestre y curso.
 3. El sistema busca el horario elegido.
 4. Se muestra y permite descarga del archivo exportado.
- **Escenarios alternativos:**
 - *.a. En cualquier momento el administrador decide cancelar el proceso.
 1. El caso de uso finaliza.
 - 3.a. No existe un horario creado para los datos elegidos.
 1. El sistema lo indica y se vuelve al paso anterior.

Caso de uso: Comprobar grupo de teoría

- **Descripción:** Caso de uso para comprobar que el número de horas introducidas en los horarios de un grupo coincide con lo indicado en el plan docente de cada asignatura.
- **Actores:** Planificador.
- **Precondiciones:** El grupo existe y tiene horarios creados. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se muestra la comprobación indicando si faltan o sobran horas en las asignaturas y actividades.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar grupo de teoría*.
 2. El sistema muestra el cálculo de horas de un grupo mediante los horarios de esa semana, teniendo en cuenta los eventos asignados en el calendario e indica dónde faltan horas por asignar para que sea solucionado.
- **Escenarios alternativos:**
 - *.a. En cualquier momento se decide parar el proceso.
 1. El caso de uso finaliza.

Gestión de aulas

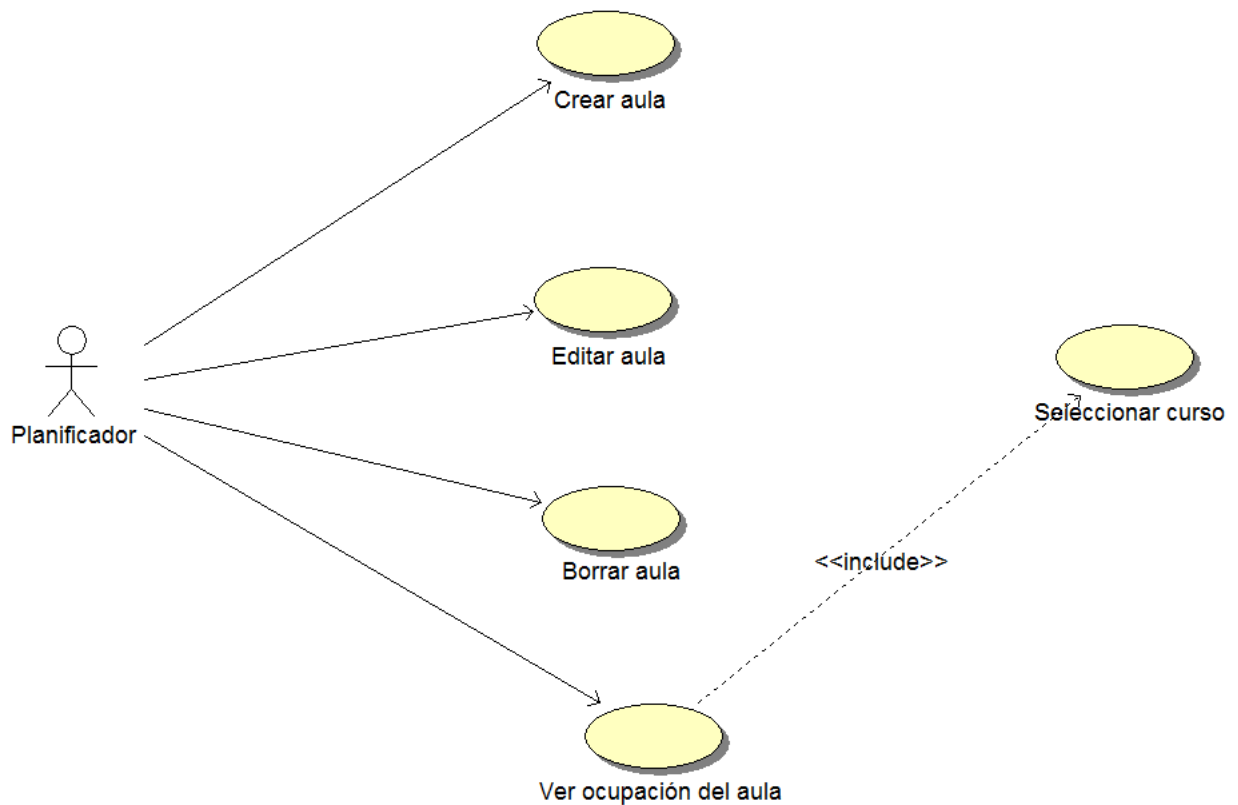


Figura 4.8: Diagrama de casos de uso de la gestión de aulas

Caso de uso: Crear aula

- **Descripción:** Caso de uso para crear un aula a la que asignar slots de horario
- **Actores:** Planificador.
- **Precondiciones:** El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** El aula queda creada
- **Escenario principal:**
 1. El administrador introduce los datos del aula: nombre y actividades para las que sirve
 2. El sistema comprueba que no exista ningún aula con ese nombre y el aula queda creada.
- **Escenarios alternativos:**
 - *.a. En cualquier momento el administrador decide cancelar el proceso.
 1. El caso de uso finaliza.
 - 2.a. Ya existe un aula con ese nombre.
 1. El sistema lo indica y se vuelve al paso anterior.

Caso de uso: Editar aula

- **Descripción:** Caso de uso para editar los datos de un aula ya creada
- **Actores:** Planificador.
- **Precondiciones:** El aula existe. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** El aula queda guardada con los nuevos datos.
- **Escenario principal:**
 1. El sistema muestra los datos actuales del aula permitiendo su edición
 2. El administrador edita el nombre y añade o elimina actividades.
 3. El sistema valida que el nuevo nombre no exista y guarda los datos
- **Escenarios alternativos:**
 - *.a. En cualquier momento el administrador decide cancelar el proceso
 1. El caso de uso finaliza.
 - 3.a. Ya existe un aula con ese nombre
 1. El sistema lo indica y se vuelve al paso anterior

Caso de uso: Eliminar aula

- **Descripción:** Caso de uso para eliminar un aula del sistema.
- **Actores:** Planificador.
- **Precondiciones:** El aula existe en el sistema. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** El aula queda eliminada del sistema y todas las referencias a ésta quedan borradas.
- **Escenario principal:**
 1. Se selecciona el aula a eliminar
 2. El sistema pide confirmación
 3. El administrador confirma la eliminación
 4. El sistema elimina el aula y borra todas las referencias a ésta.
- **Escenarios alternativos:**
 - 3.a. El administrador decide que no desea eliminar el aula.
 1. El caso de uso finaliza

Caso de uso: Ver ocupación de aula

- **Descripción:** Caso de uso para ver la ocupación actual de un aula según los horarios creados para el curso actual.
- **Actores:** Planificador.
- **Precondiciones:** El aula existe en el sistema. El usuario identificado en el sistema es un planificador.
- **Postcondiciones:** Se muestran todos los slots que están asignados a ese aula junto con su horario.
- **Escenario principal:**
 1. El administrador selecciona el aula para la que desea ver la ocupación.
 2. El administrador selecciona el semestre y la semana para el que desea ver la ocupación.
 3. El sistema muestra la ocupación actual.
- **Escenarios alternativos:**
 - *.a. En cualquier momento el administrador decide cancelar el proceso.
 1. El caso de uso finaliza.

Caso de uso: Visualización de horario

- **Descripción:** Caso de uso para visualizar un horario por parte de un alumno personalizado con las asignaturas seleccionadas por él.
- **Actores:** Alumno.
- **Precondiciones:** El usuario identificado en el sistema es un alumno.
- **Postcondiciones:** Se muestran todos los slots que ha seleccionado el alumno.
- **Escenario principal:**
 1. Se realiza el caso de uso *Seleccionar asignatura*.
 2. Se repite el paso anterior hasta seleccionar todas las asignaturas deseadas.
 3. El sistema muestra un horario personalizado según las asignaturas elegidas.
- **Escenarios alternativos:**
 - *.a. En cualquier momento el alumno decide cancelar el proceso.
 1. El caso de uso finaliza.

4.3. Modelo conceptual de datos

4.3.1. Diagrama de clases conceptuales

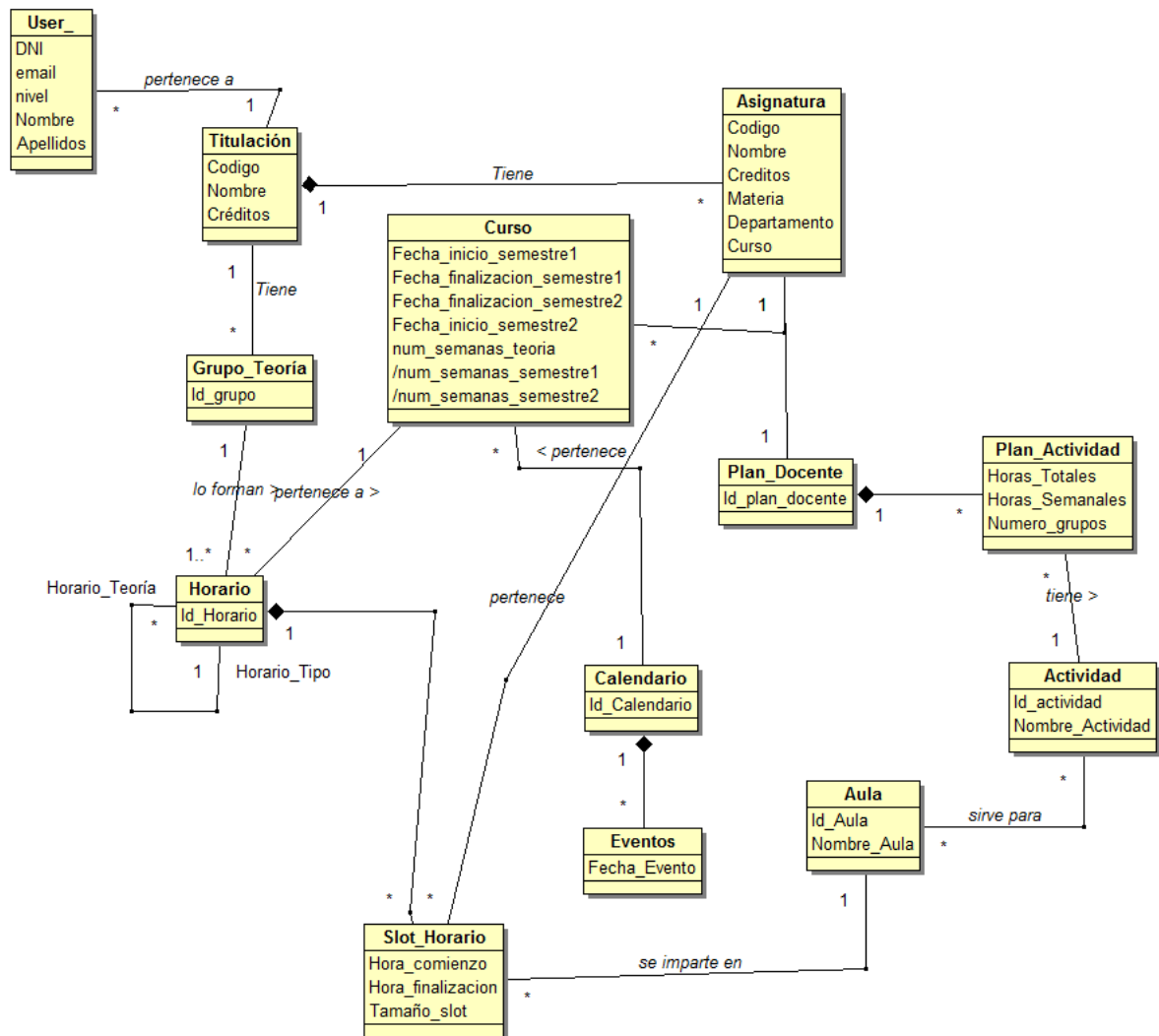


Figura 4.9: Diagrama del modelo conceptual de datos

4.4. Modelo de comportamiento del sistema

Para el modelo de comportamiento del sistema se mostrarán diferentes diagramas de secuencia del sistema. El diagrama define las interacciones entre actores y sistema, también se detallarán los contratos de las operaciones del sistema, para describir en detalle qué hace cada operación.

Al existir muchos casos de uso similares, sólo se detallarán los más relevantes de cada subsistema.

4.4.1. Caso de uso: Registrar titulación

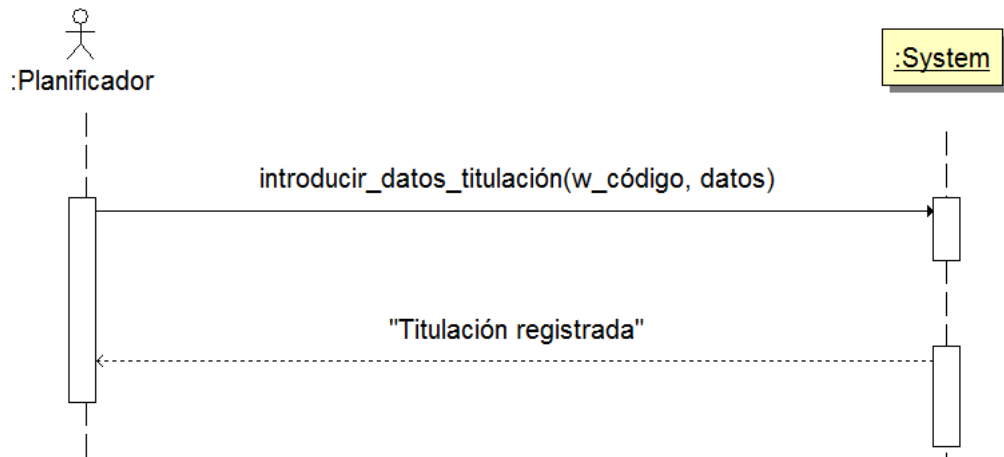


Figura 4.10: Diagrama de secuencia del caso de uso Registrar titulación

Contrato de la operación: introducir_datos_titulación

- **Responsabilidades:** Registrar una titulación en el sistema.
- **Referencias cruzadas:** Caso de uso *registrar titulación*, Caso de uso *editar titulación*.
- **Precondiciones:** No existe ninguna titulación con código = w_código.
- **Postcondiciones:** Se crea una instancia T de Titulación. Se asignan w_código y datos a T.

4.4.2. Caso de uso: Registrar asignatura

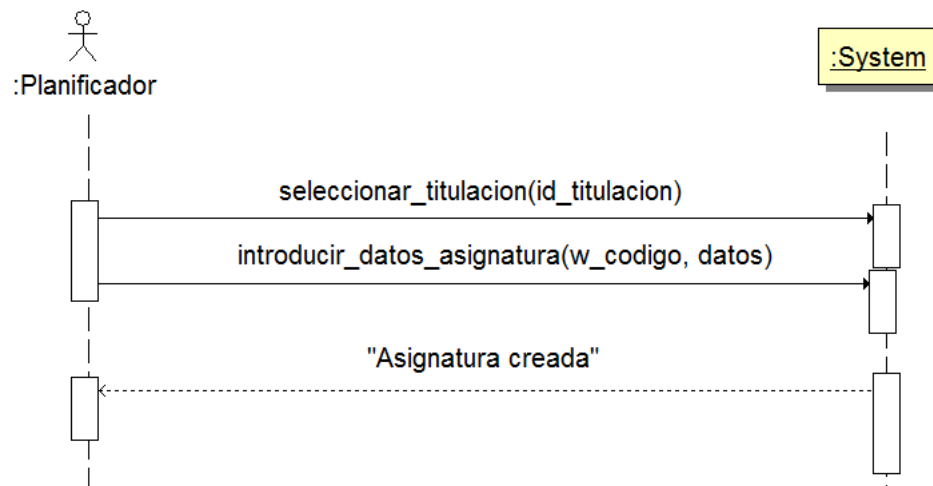


Figura 4.11: Diagrama de secuencia del caso de uso Registrar asignatura

Contrato de la operación: seleccionar_titulacion

- **Responsabilidades:** Seleccionar una titulación para añadirle una asignatura.
- **Referencias cruzadas:** Caso de uso *registrar asignatura*, Caso de uso *editar asignatura*
- **Precondiciones:** Existe una titulación con $t.id = id_titulación$.
- **Postcondiciones:** Se devuelve una instancia T de Titulación con $T.id = id_titulación$.

Contrato de la operación: introducir_datos_asignatura

- **Responsabilidades:** Registrar una asignatura y asociarla a una titulación.
- **Referencias cruzadas:** Caso de uso *registrar asignatura*, Caso de uso *editar asignatura*
- **Precondiciones:** No existe ninguna asignatura con código = w_codigo.
- **Postcondiciones:** Se crea una instancia A de Asignatura. Se asigna $A.codigo = w_codigo$ y $A.datos = datos$.

4.4.3. Caso de uso: Editar asignatura

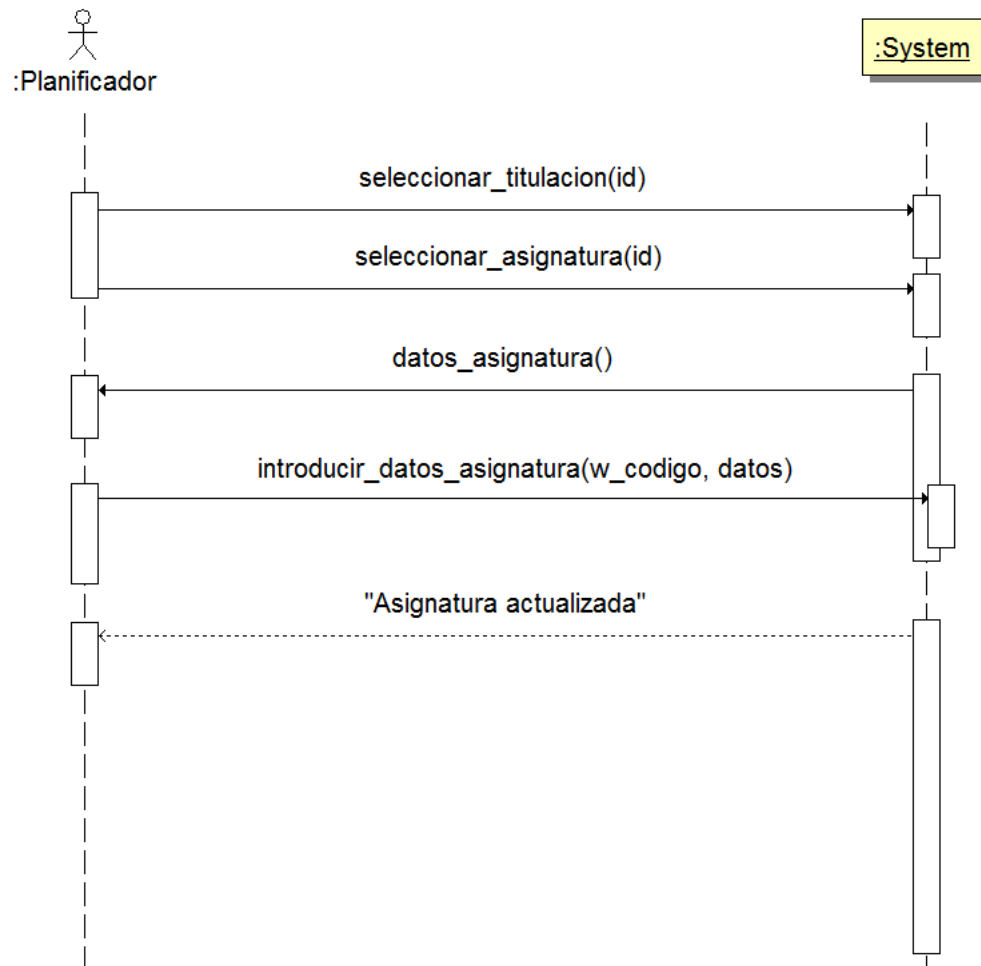


Figura 4.12: Diagrama de secuencia del caso de uso Editar asignatura

Contrato de la operación: seleccionar_asignatura

- **Responsabilidades:** Devolver una instancia de Asignatura para poder editar sus datos.
- **Referencias cruzadas:** Caso de uso *editar asignatura*, Caso de uso *borrar asignatura*.
- **Precondiciones:** Debe existir una Asignatura registrada en el sistema con A.id = id_asignatura.
- **Postcondiciones:** Se devuelve una instancia A de Asignatura con A.id = id_asignatura.

4.4.4. Caso de uso: Borrar asignatura

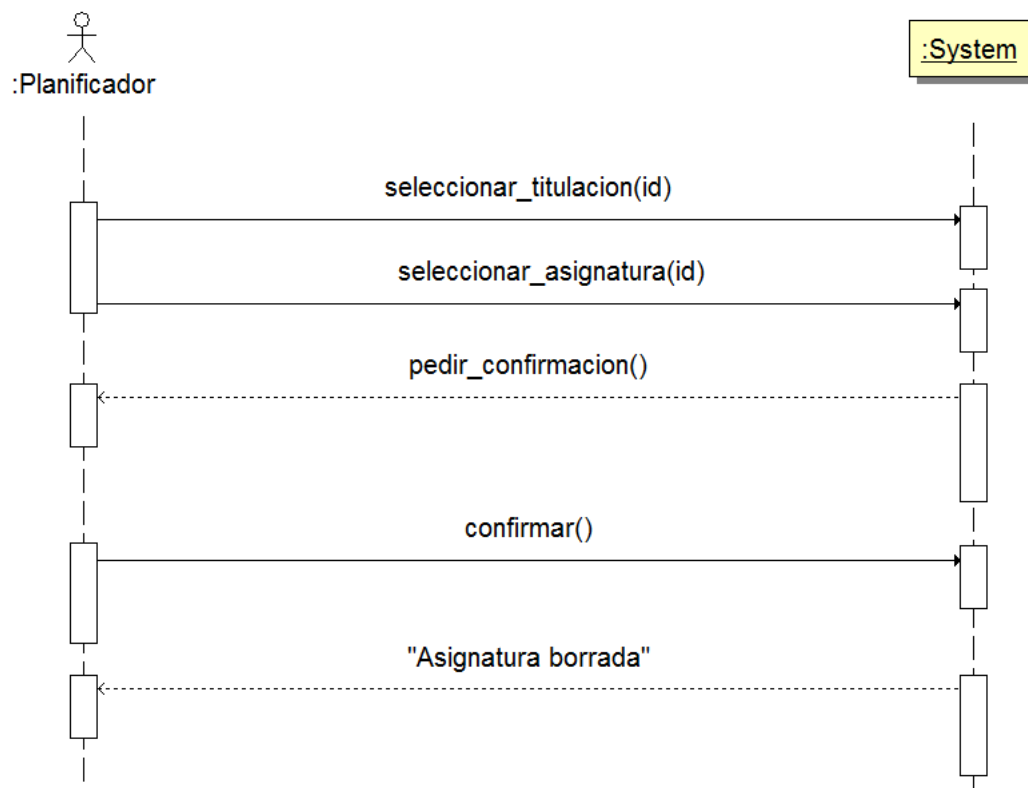


Figura 4.13: Diagrama de secuencia del caso de uso Borrar asignatura

4.4.5. Caso de uso: Importar asignatura

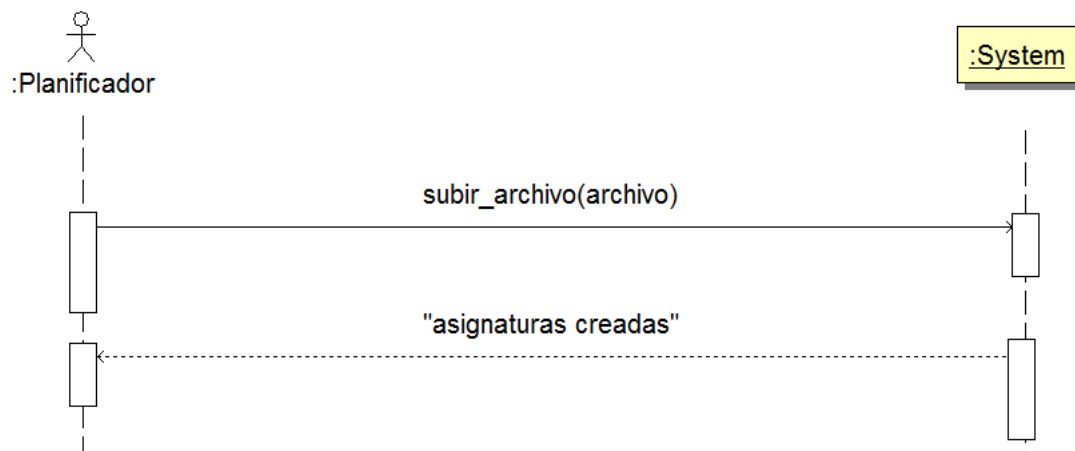


Figura 4.14: Diagrama de secuencia del caso de uso Importar asignatura

Contrato de la operación: subir_archivo

- **Responsabilidades:** Crear asignaturas masivamente capturando la información de las líneas de un archivo.
- **Referencias cruzadas:** Caso de uso *importar asignatura*, Caso de uso *importar titulación*, Caso de uso *importar plan docente*.
- **Precondiciones:** Por cada línea del archivo, no existe una instancia de la clase a la que se pretende añadir información con la misma clave que aparece en la línea.
- **Postcondiciones:** Por cada línea se crea una instancia de la clase con `id = id_línea`.

4.4.6. Caso de uso: Crear plan docente

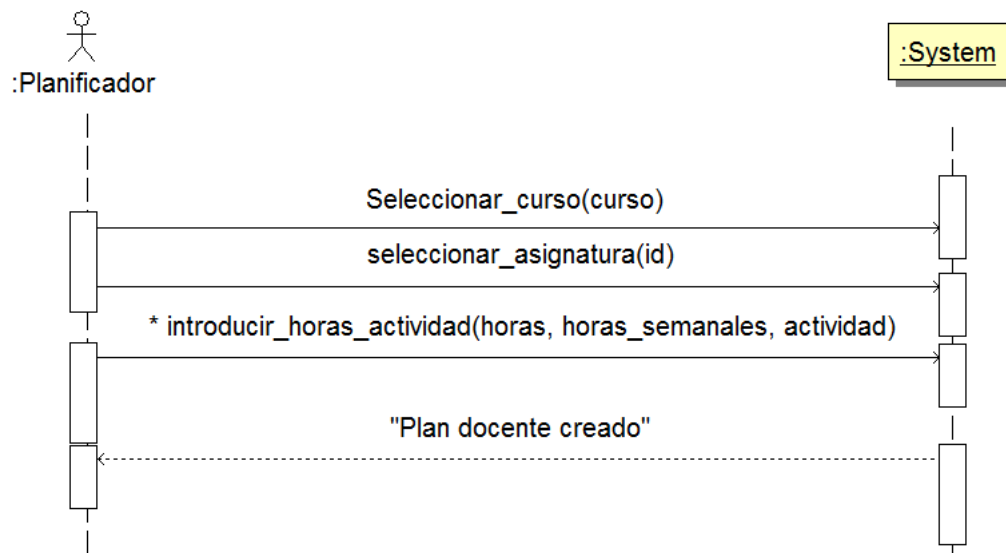


Figura 4.15: Diagrama de secuencia del caso de uso Crear plan docente

Contrato de la operación: introducir_horas_actividad

- **Responsabilidades:** Añadir la planificación docente de una actividad concreta al sistema.
- **Referencias cruzadas:** Caso de uso *crear plan docente*, Caso de uso *editar plan docente*
- **Precondiciones:** La actividad existe en el sistema.
- **Postcondiciones:** Se crea una instancia `P` de `Plan_Actividad` con `P.actividad = actividad`, `P.horas = horas` y `P.horas_semanales = horas_semanales`, además se asocia con la asignatura `A`.

4.4.7. Caso de uso: Generar informe de asignatura

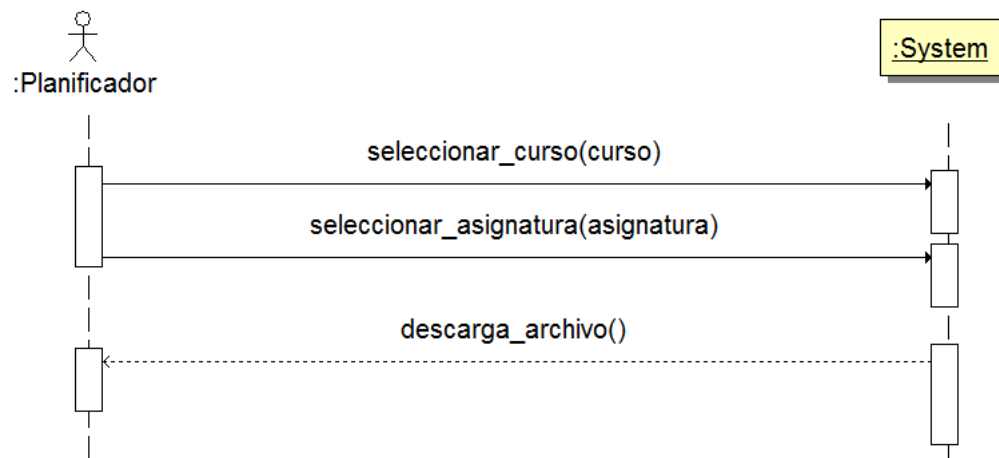


Figura 4.16: Diagrama de secuencia del caso de uso Generar informe de asignatura

4.4.8. Caso de uso: Añadir evento al calendario

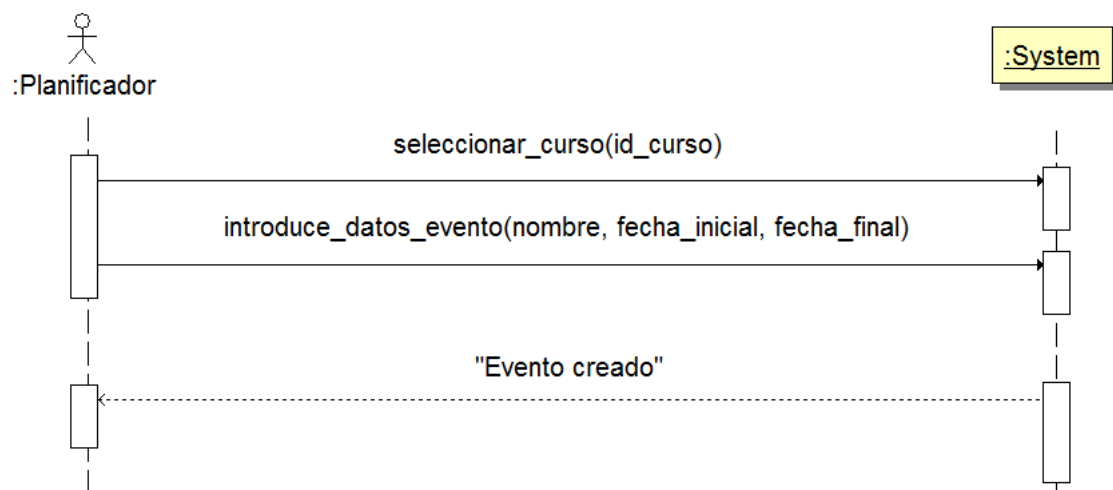


Figura 4.17: Diagrama de secuencia del caso de uso Añadir evento al calendario

Contrato de la operación: seleccionar curso

- **Responsabilidades:** Seleccionar un curso para hacer operaciones con el o asociarlo con otras clases.
- **Referencias cruzadas:** Caso de uso *Añadir evento al calendario*.
- **Precondiciones:** Existe un curso C con $C.id_curso = id_curso$.
- **Postcondiciones:** Se devuelve una instancia C de curso.

Contrato de la operación: introduce_datos_evento

- **Responsabilidades:** Crear un evento en el calendario del curso C.
- **Referencias cruzadas:** Caso de uso *Añadir evento al calendario*.
- **Precondiciones:** No existe ningún evento con fechas entre las introducidas.
- **Postcondiciones:** Se crea una instancia E de Evento con E.fecha_inicial = fecha_inicial, E.fecha_final = fecha_final y E.nombre = nombre y se asocia a C.

4.4.9. Caso de uso: Exportar calendario

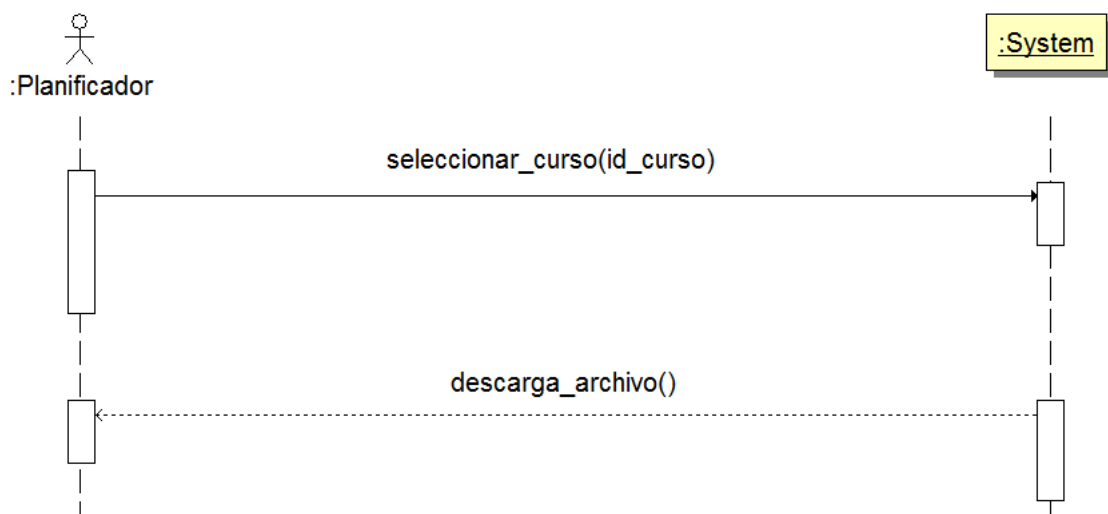


Figura 4.18: Diagrama de secuencia del caso de uso Exportar calendario

4.4.10. Caso de uso: Seleccionar grupos de teoría

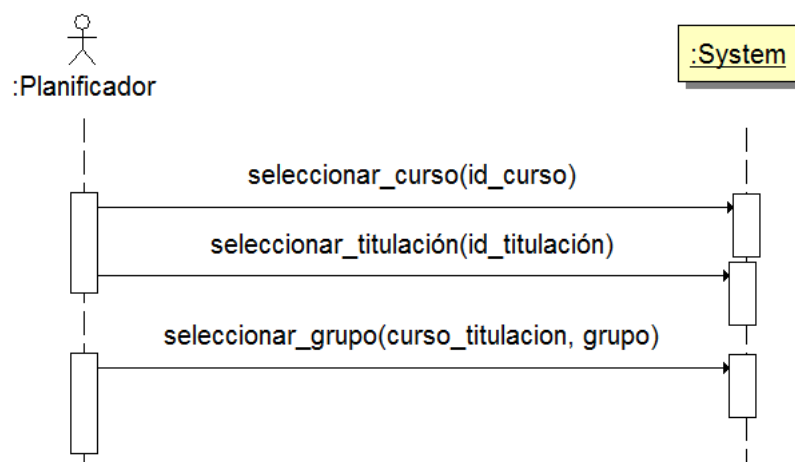


Figura 4.19: Diagrama de secuencia del caso de uso Seleccionar grupos de teoría

Contrato de la operación: seleccionar_grupo

- **Responsabilidades:** Seleccionar un grupo de una titulación para hacer uso de él en otro caso de uso.
- **Referencias cruzadas:** Caso de uso *seleccionar grupos de teoría*
- **Precondiciones:** El curso indicado está dentro de los cursos de la titulación.
El grupo ha sido creado.
- **Postcondiciones:** Se devuelve una instancia G del Grupo.

4.4.11. Caso de uso: Añadir grupo de teoría

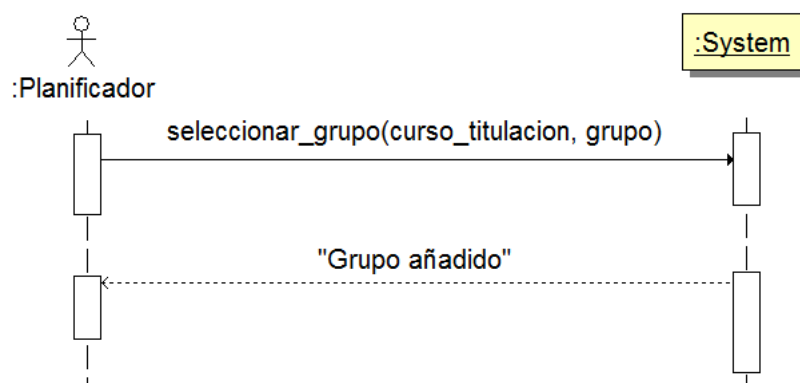


Figura 4.20: Diagrama de secuencia del caso de uso Añadir grupo de teoría

4.4.12. Caso de uso: Eliminar grupo de teoría

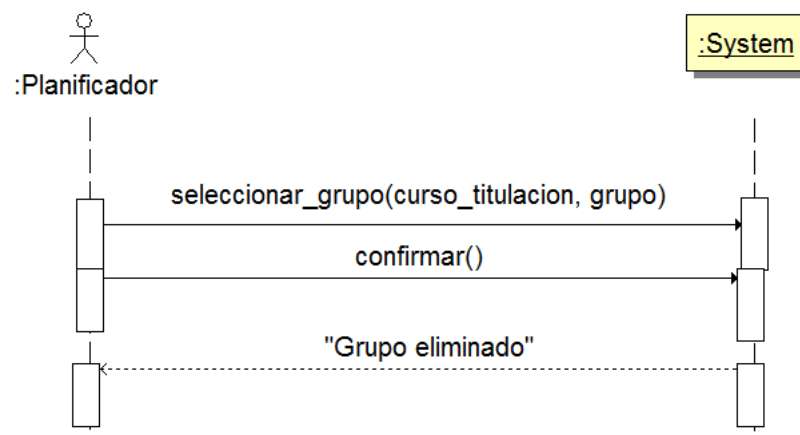


Figura 4.21: Diagrama de secuencia del caso de uso Eliminar grupo de teoría

4.4.13. Caso de uso: Ubicar slot de horario

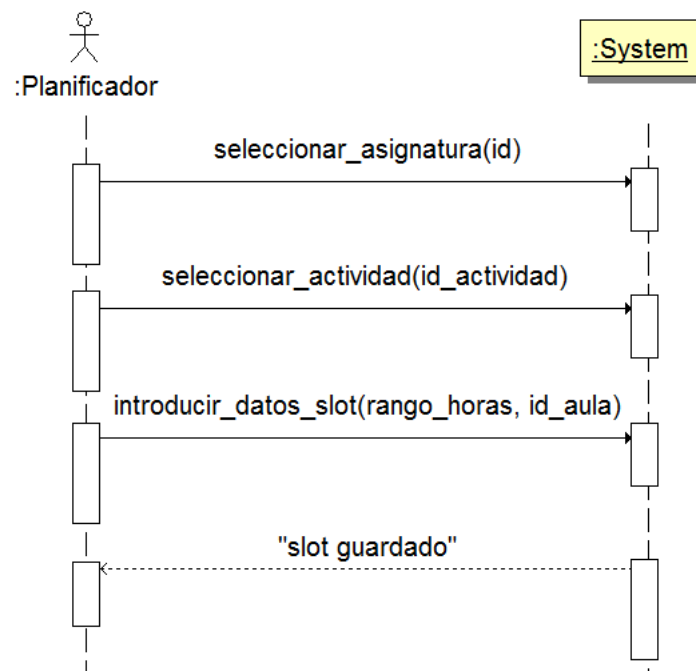


Figura 4.22: Diagrama de secuencia del caso de uso Ubicar slot de horario

Contrato de la operación: seleccionar_actividad

- **Responsabilidades:** Seleccionar una actividad para introducirla en el horario
- **Referencias cruzadas:** Caso de uso *ubicar slot de horario*.
- **Precondiciones:** La asignatura elegida dispone de horas en el plan docente para esa actividad.
- **Postcondiciones:** Se devuelve una instancia de un Slot de horario para esa asignatura y actividad.

Contrato de la operación: introducir_datos_slot

- **Responsabilidades:** Ubicar en el horario un slot de una actividad de una asignatura.
- **Referencias cruzadas:** Caso de uso *ubicar slot de horario*
- **Precondiciones:** Si el slot es de teoría no se solapa con otro slot. No se debe solapar con otro slot con el mismo id_aula.
- **Postcondiciones:** El slot queda ubicado en el horario.

4.4.14. Caso de uso: Editar horario tipo

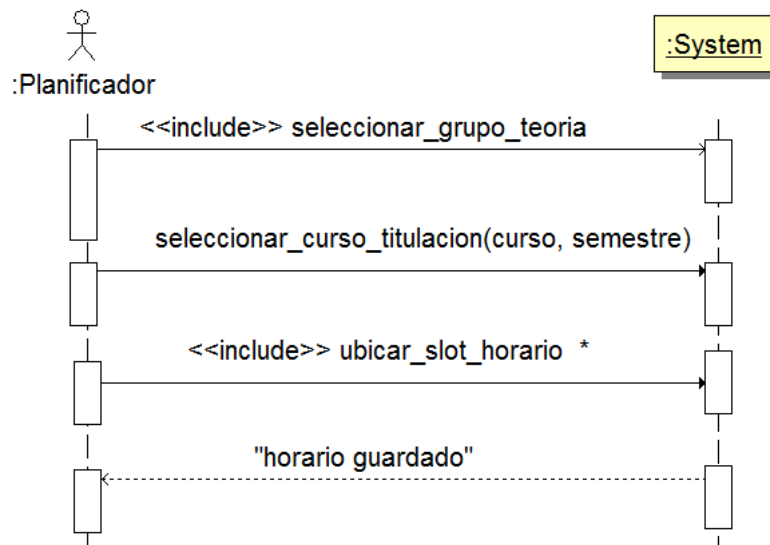


Figura 4.23: Diagrama de secuencia del caso de uso Editar horario tipo

Contrato de la operación: seleccionar_curso_titulacion

- **Responsabilidades:** Seleccionar una instancia del Horario
- **Referencias cruzadas:** Caso de uso *editar horario tipo*, Caso de uso *editar horario semana inicial*.
- **Precondiciones:** El curso está dentro de los que tiene la titulación.
- **Postcondiciones:** Se devuelve una instancia de Horario (o se crea una si no existe).

4.4.15. Caso de uso: Comprobar grupo de teoría

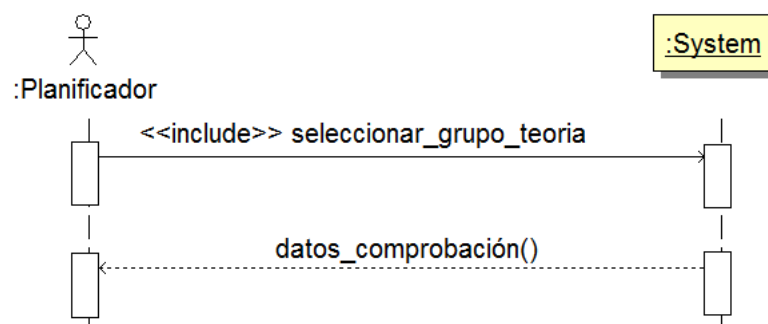


Figura 4.24: Diagrama de secuencia del caso de uso Comprobar grupo de teoría

4.4.16. Caso de uso: Crear aula

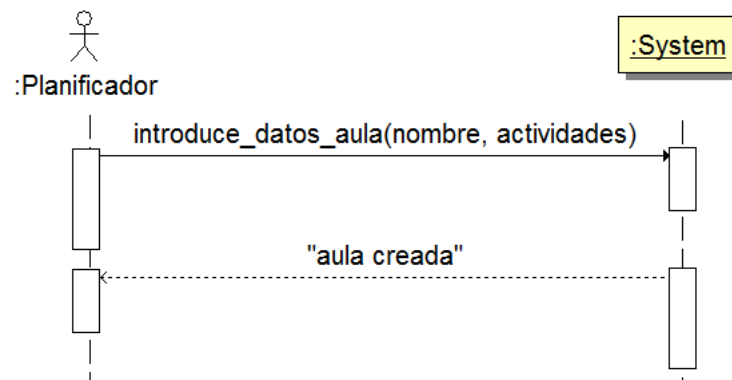


Figura 4.25: Diagrama de secuencia del caso de uso Crear aula

Contrato de la operación: introducir_datos_aula

- **Responsabilidades:** Crear un aula en el sistema.
- **Referencias cruzadas:** Caso de uso *crear aula*, Caso de uso *Editar aula*.
- **Precondiciones:** No existe ningún aula en el sistema con A.nombre = nombre.
- **Postcondiciones:** Se crea una instancia A de Aula con A.nombre = nombre y A.actividades = actividades.

4.4.17. Caso de uso: Borrar aula

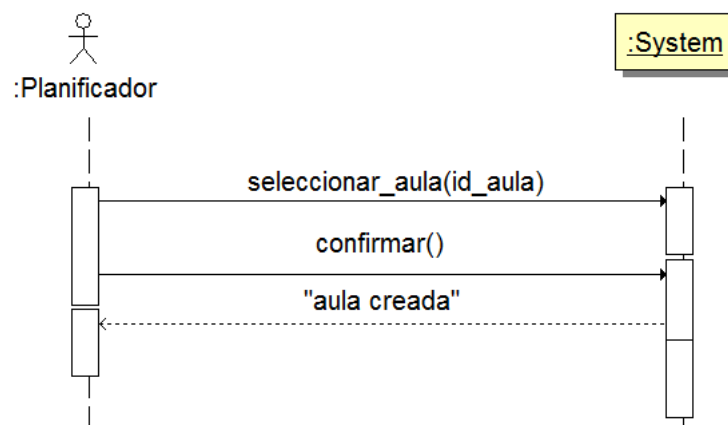


Figura 4.26: Diagrama de secuencia del caso de uso Borrar aula

Contrato de la operación: seleccionar_aula

- **Responsabilidades:** Seleccionar un aula para borrarla o editarla.
- **Referencias cruzadas:** Caso de uso *borrar aula*, Caso de uso *editar aula*.

- **Precondiciones:** Existe un aula en el sistema con A.id_aula = id_aula.
- **Postcondiciones:** Se devuelve una instancia A de aula con A.id_aula = id_aula.

4.4.18. Caso de uso: Editar aula

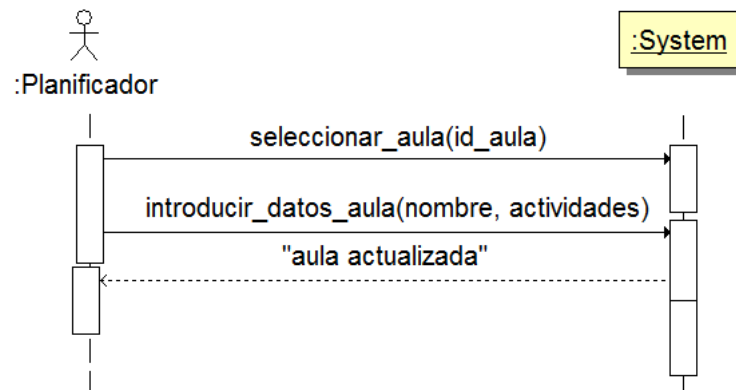


Figura 4.27: Diagrama de secuencia del caso de uso Editar aula

Capítulo 5

Diseño

La fase de diseño consiste en aplicar una serie de técnicas para transformar los requisitos elicitados en la fase de análisis en una estructura detallada para el sistema de forma que se pueda implementar fácilmente a partir de ese diseño.

Siguiendo los requisitos de la fase anterior, esta tarea es relativamente sencilla y debe resultar en una serie de diagramas y especificaciones que sirvan como guión y documentación a las personas que vayan a participar, ahora o en un futuro en el desarrollo del sistema.

La aplicación se rige por el patrón arquitectónico MVC. La capa del modelo equivale al SGBD y al ORM, mientras el controlador es el responsable de hacer de intermediario entre vista y modelo, llevando la lógica de la aplicación. Las vistas serán los ficheros XHTML junto con sus estilos.

5.1. Controladores

A continuación, se comentarán las acciones más importantes que componen cada uno de los controladores. Normalmente cada controlador tiene asociado un modelo, y cada acción del controlador una vista.

5.1.1. Users

Controlador encargado de llevar toda la gestión de usuarios

- `users::add()` - Muestra el formulario de creación de un usuario.
- `users::create()` - Crea un usuario según los datos introducidos en el formulario.
- `users::edit()` - Muestra un formulario con los datos del usuario a editar sobre él para poder ser modificados.
- `users::update()` - Actualiza un usuario según los datos introducidos.
- `users::recuperar_password()` - Muestra el formulario para recuperar una contraseña.
- `users::envio_recuperar()` - Envía y cambia la contraseña a un usuario por email a la dirección introducida en el formulario.

5.1.2. Login

Controlador encargado de gestionar la identificación de usuarios en el sistema.

- `login::index()` - Muestra el formulario de solicitud de datos de acceso.
- `login::submit()` - Comprueba los datos introducidos en el formulario anterior e identifica al usuario si son correctos, sino recarga el formulario y lo indica.

5.1.3. Logout

Controlador encargado de gestionar la salida de los usuarios del sistema

- `logout::index()` - Provoca la salida del usuario del sistema.

5.1.4. Admin

Controlador encargado de algunas funciones de administración.

- `admin::restaurar_backup()` - Muestra un formulario para cargar un archivo sql con una copia de seguridad de la base de datos, restaurando los datos que estuvieran en ese archivo.
- `admin::restaurar()` - Restaura la base de datos con el archivo cargado en el formulario anterior.
- `admin::backup()` - Permite la descarga de un backup del estado actual de la base de datos.

5.1.5. Titulaciones

Controlador encargado de llevar a cabo toda la gestión de las titulaciones.

- `titulaciones::index()` - Muestra las titulaciones que hay creadas en el sistema.
- `titulaciones::show_informes()` - Muestra un listado de las asignaturas de la titulación permitiendo generar un informe con las que se quieran seleccionar.
- `titulaciones::add()` - Muestra un formulario de creación de titulaciones.
- `titulaciones::create()` - Crea una titulación a partir de los datos introducidos en el formulario.
- `titulaciones::edit()` - Permite la edición de una titulación mostrando un formulario.
- `titulaciones::update()` - Actualiza los datos de una titulación existente.
- `titulaciones::delete()` - Borra una titulación del sistema.
- `titulaciones::show()` - Muestra las asignaturas de esa titulación.
- `titulaciones::show_planificacion()` - Muestra una tabla con la planificación docente de una titulación, con una fila por asignatura.
- `titulaciones::exportar_planificacion()` - Permite la descarga de un archivo csv con la planificación docente completa de todas las asignaturas de la titulación.
- `titulaciones::select_titulacion()` - Muestra un listado de las titulaciones permitiendo enrutar hacia a otra acción que necesite seleccionar una titulación.

5.1.6. Asignaturas

Controlador encargado de llevar a cabo toda la gestión de las asignaturas.

- `asignaturas::show()` - Muestra la planificación docente de una asignatura para un curso concreto.
- `asignaturas::add_to()` - Añade una asignatura a una titulación dada mostrando un formulario.
- `asignaturas::create()` - Crea una asignatura siguiendo los datos introducidos en el formulario.

5.1.7. Aulas

Controlador encargado de llevar a cabo la gestión de las aulas.

- `aulas::add()` - Muestra el formulario de creación de un aula.
- `aulas::create()` - Crea un aula a partir de los datos introducidos en el formulario.
- `aulas::index()` - Muestra un listado de todas las aulas.
- `aulas::exportar_ocupacion()` - Exporta a un archivo csv la ocupación de un aula según los horarios de un curso concreto.

5.1.8. Cursos

Controlador que lleva toda la gestión de los cursos.

- `cursos::add()` - Muestra el formulario para la creación de un curso nuevo.
- `cursos::create()` - Crea un curso con los datos introducidos.
- `cursos::edit()` - Muestra un formulario para editar un curso ya creado.
- `cursos::update()` - Actualiza un curso existente con los datos introducidos.
- `cursos::delete()` - Borra un curso existente.
- `cursos::select_curso()` - Muestra un listado de los cursos permitiendo enrutar hacia a otra acción que necesite seleccionar uno.

5.1.9. Eventos

Controlador encargado de la gestión del calendario.

- `eventos::index()` - Muestra el calendario del sistema permitiendo crear nuevos eventos sobre el.
- `eventos::add()` - Muestra un formulario para la creación de eventos.
- `eventos::create()` - Crea un evento con los datos introducidos en el formulario.
- `eventos::delete()` - Borra un evento existente.
- `eventos::fetch_events()` - Función que devuelve en formato JSON los datos de los eventos del sistema para mostrarlos sobreimpresos en el calendario.
- `eventos::export_calendar()` - Exporta y permite la descarga del calendario en formato csv.

5.1.10. Horarios

Controlador encargado de la gestión de los horarios.

- `horarios::select_grupo()` - Muestra los cursos de una titulación y sus grupos creados, con enlaces a los diferentes horarios.
- `horarios::add_grupo()` - Añade un grupo a un curso de una titulación concreta.
- `horarios::edit()` - Muestra un horario editable permitiendo ubicar los diferentes slots de las asignaturas.
- `horarios::ocupacion_aula()` - Muestra la ocupación de un aula concreta.
- `horarios::exportar_ocupacion()` - Permite exportar y descargar un archivo csv con la ocupación de un aula.
- `horarios::edit_teoría()` - Muestra y permite editar un horario de una semana que solo tiene asignaturas de teoría.
- `horarios::check_grupo()` - Realiza la comprobación de las horas planificadas y asignadas en los horarios de un grupo de una titulación, mostrando el resultado.
- `horarios::save_line()` - Guarda un slot de una asignatura en un horario concreto.
- `horarios::delete()` - Borra un horario.
- `horarios::delete_line()` - Borra un slot de un horario, dejándolo sin asignar.
- `horarios::exportar()` - Permite exportar un horario concreto a csv permitiendo la descarga del archivo.
- `horarios::add_extra_slot()` - Permite añadir un slot extra de una asignatura en un horario de una semana de teoría.
- `horarios::visualizacion_asignaturas()` - Permite a un alumno la visualización de un horario personalizado. Para ello esta función muestra un listado de las asignaturas disponibles permitiendo seleccionar las deseadas.
- `horarios::visualizacion_mostrar_grupos()` - Muestra las asignaturas seleccionadas en el paso anterior y permite seleccionar los grupos deseados.
- `horarios::visualizacion_mostrar_horario()` - Muestra el horario personalizado en los dos pasos anteriores.

5.1.11. PlanesDocentes

Controlador encargado de la gestión de la planificación docente.

- `planesdocentes::add_carga()` - Muestra un formulario para añadir la planificación docente de un curso a una asignatura.
- `planesdocentes::create()` - Crea en el sistema la planificación docente según los datos introducidos en el formulario.
- `planesdocentes::edit()` - Muestra un formulario para editar la planificación docente ya existente de una asignatura.

- `planesdocentes::update()` - Actualiza la planificación docente de una asignatura según los datos ya introducidos.
- `planesdocentes::make_upload()` - Muestra un formulario para subir un archivo con un csv con la planificación docente de una o varias asignaturas.
- `planesdocentes::upload_file()` - Parsea el fichero subido en el paso anterior y crea los planes docentes nuevos.
- `planesdocentes::informe_asignatura()` - Genera un informe en pdf de una o varias asignaturas, permitiendo su descarga.

5.2. Base de datos

Para el diseño de la base de datos en la que se guardarán los datos manejados por la aplicación se usará un modelo relacional. Se usará MySQL como sistema de gestión de base de datos.

5.2.1. Modelo entidad-relación

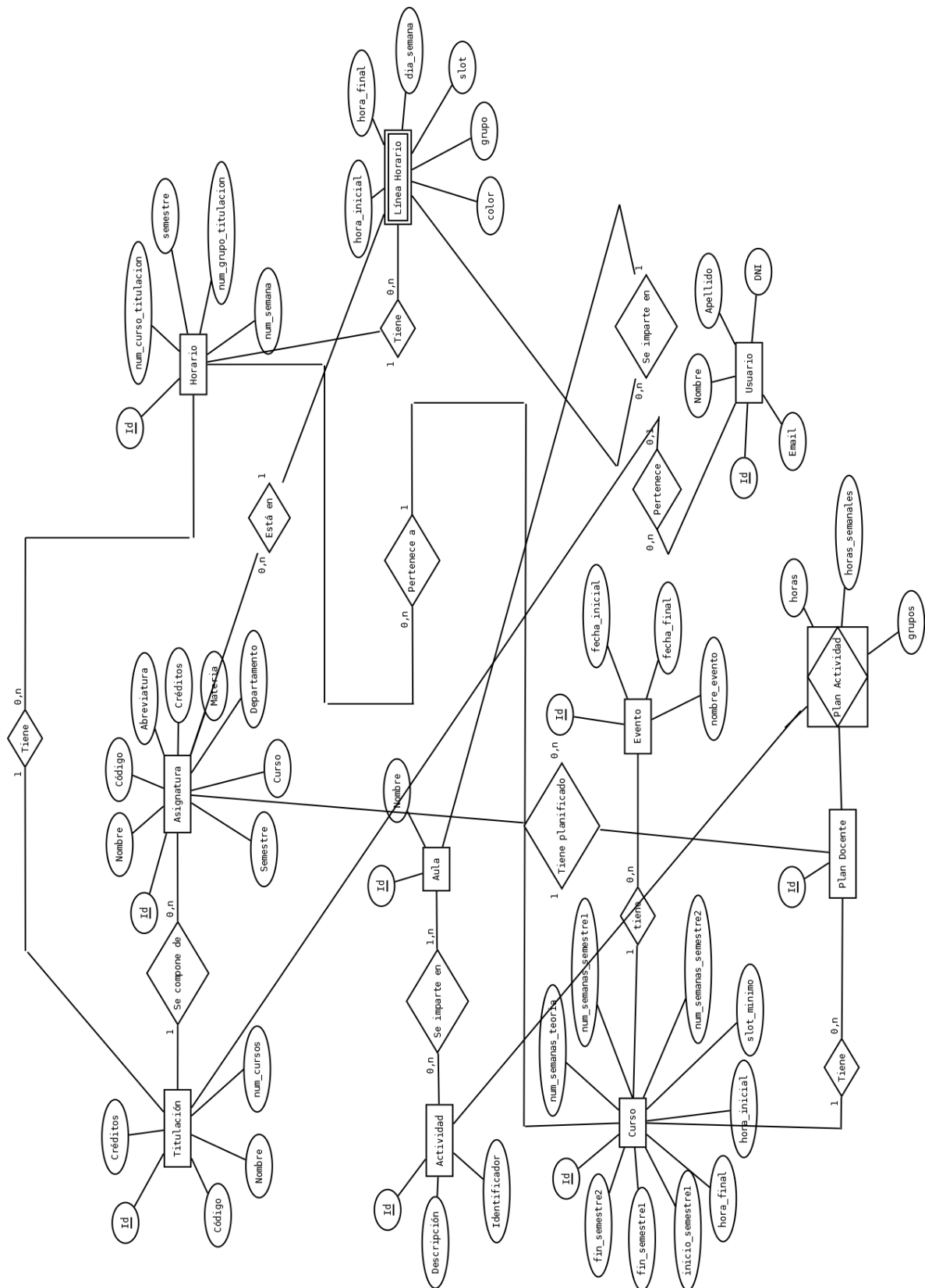


Figura 5.1: Diagrama Entidad-Relación

5.2.2. Tablas y atributos

Tabla actividades

En esta tabla se definen todas las actividades que tendrán las asignaturas, en principio son fijas y la tabla se utiliza simplemente para guardar los códigos por los que son identificadas en la universidad.

Tabla 5.1: Estructura de la tabla actividades

Columna	Tipo	Nulo	Descripción
<i>id</i>	bigint(20)	No	Identificador autoincremental de la tabla
descripcion	varchar(100)	No	Nombre completo de la actividad
identificador	varchar(1)	No	Identificador alfabético de la actividad

Tabla asignaturas

En esta tabla se almacenan todas las asignaturas de la titulación, guardando además la clave foránea con la relación con titulaciones.

Tabla 5.2: Estructura de la tabla asignaturas

Columna	Tipo	Nulo	Descripción
<i>id</i>	int(11)	No	Identificador autoincremental de la tabla
<i>codigo</i>	varchar(3)	No	Código utilizado por la Universidad para la asignatura
nombre	varchar(200)	No	Nombre completo de la asignatura
abreviatura	varchar(5)	No	Abreviatura de la asignatura
creditos	int(11)	No	Créditos que tiene la asignatura dentro de la titulación
materia	varchar(100)	No	Materia de la asignatura
departamento	varchar(200)	No	Departamento al que pertenece la asignatura
curso	int(10)	No	Curso dentro de la titulación al que pertenece
semestre	varchar(255)	No	Semestre en el que se imparte la asignatura
titulacion_id	int(11)	No	Clave foránea de la relación de la tabla con <i>titulaciones</i>

Tabla aulaactividades

Esta es una tabla de asociación, sirve para relacionar la tabla aulas con la tabla actividades, y saber de esta forma que actividades se pueden impartir en cada aula.

Tabla 5.3: Estructura de la tabla aulaactividades

Columna	Tipo	Nulo	Descripción
<i>id_actividad</i>	bigint(20)	No	Clave foránea en la relación con actividades
<i>id_aula</i>	bigint(20)	No	Clave foránea en la relación con aulas

Tabla aulas

Tabla para almacenar las aulas que se usarán en el sistema.

Tabla 5.4: Estructura de la tabla aulas

Columna	Tipo	Nulo	Descripción
<i>id</i>	bigint(20)	No	Identificador autoincremental de la tabla
nombre	varchar(100)	No	Nombre del aula

Tabla ci_sessions

Tabla utilizada por el framework para almacenar la información de las sesiones.

Tabla 5.5: Estructura de la tabla ci_sessions

Columna	Tipo	Nulo	Descripción
<i>session_id</i>	varchar(40)	No	Id autoincremental de la tabla
ip_address	varchar(16)	No	Dirección IP de la sesión
user_agent	varchar(120)	No	Agente de usuario de la sesión (Navegador, etc)
last_activity	int(10)	No	Valor de la última actividad
user_data	text	Sí	Datos del usuario que deben ser almacenados

Tabla cursos

Tabla utilizada para almacenar toda la información de la configuración de un curso.

Tabla 5.6: Estructura de la tabla cursos

Columna	Tipo	Nulo	Descripción
<i>id</i>	int(11)	No	Id autoincremental de la tabla
num_semanas_teoría	int(11)	No	Número de semanas de solo horas de teoría del curso
num_semanas_semestre1	int(11)	No	Número de semanas que tendrá el semestre 1
num_semanas_semestre2	int(11)	No	Número de semanas que tendrá el semestre 2
horas_por_credito	int(11)	No	El número de horas que tendrá cada crédito de una asignatura
slot_minimo	bigint(20)	No	Slot mínimo que tendrá una asignatura en un horario, en minutos
hora_inicial	time	No	Hora inicial de los horarios de ese curso
hora_final	time	No	Hora final de los horarios de ese curso
inicio_semestre1	date	No	Fecha de inicio del primer semestre
fin_semestre1	date	No	Fecha de finalización del primer semestre
inicio_semestre2	date	No	Fecha de inicio del segundo semestre
fin_semestre2	date	No	Fecha de finalización del segundo semestre
inicio_exámenes_enero	date	No	Fecha de inicio de los exámenes de enero
fin_exámenes_enero	date	No	Fecha de finalización de los exámenes de enero
inicio_exámenes_junio	date	No	Fecha de inicio de los exámenes de junio
fin_exámenes_junio	date	No	Fecha de finalización de los exámenes de junio
inicio_exámenes_sept	date	No	Fecha de inicio de los exámenes de septiembre
fin_exámenes_sept	date	No	Fecha de finalización de los exámenes de septiembre

Tabla cursos_compartidos

Tabla 5.7: Estructura de la tabla cursos_compartidos

Columna	Tipo	Nulo	Descripción
<i>id_plandocente</i>	int(11)	No	Id autoincremental de la tabla
<i>num_curso_compartido</i>	int(11)	No	Número del curso con el que se comparte el plan docente

Tabla eventos

Tabla donde se almacenan los eventos de un curso para completar el calendario.

Tabla 5.8: Estructura de la tabla eventos

Columna	Tipo	Nulo	Descripción
<i>id</i>	int(11)	No	Id autoincremental de la tabla
nombre_evento	varchar(255)	No	Nombre del evento
tipo_evento	varchar(255)	No	Tipo de evento
fecha_individual	tinyint(1)	No	Booleano para saber si el evento es una única fecha o un rango de ellas
fecha_inicial	date	No	Fecha inicial del rango que compone el evento
fecha_final	date	No	Fecha final del rango que compone el evento
curso_id	int(11)	No	Id del curso al que pertenece (Clave foránea)

Tabla horarios

Tabla que almacena la información principal de un horario.

Tabla 5.9: Estructura de la tabla horarios

Columna	Tipo	Nulo	Descripción
<i>id</i>	int(11)	No	Identificador autoincremental de la tabla
id_curso	int(11)	No	Identificador del curso al que pertenece (clave foránea)
id_titulacion	int(11)	No	Identificador de la titulación a la que pertenece (clave foránea)
num_curso_titulacion	int(11)	No	Número del curso de la titulación al que pertenece el horario
semestre	varchar(255)	No	Semestre del horario
num_grupo_titulacion	int(11)	No	Grupo de teoría de la titulación
num_semana	int(10)	No	El número de la semana en el que se imparte

Tabla horario_reference

Tabla de asociación de la tabla horarios consigo misma, para indicar las relaciones entre horarios tipo y horarios de semanas iniciales.

Tabla 5.10: Estructura de la tabla horario_reference

Columna	Tipo	Nulo	Descripción
<i>id_tipo</i>	bigint(20)	No	Id del horario con el rol de tipo en la relación
<i>id_teoria</i>	bigint(20)	No	Id del horario con el rol de semana inicial

Tabla lineashorarios

Tabla de los slots que componen un horario.

Tabla 5.11: Estructura de la tabla lineashorarios

Columna	Tipo	Nulo	Descripción
<i>id</i>	int(11)	No	Identificador autoincremental de la tabla
id_horario	int(11)	No	Identificador del horario al que pertenece el slot (clave foránea)
id_asignatura	int(11)	No	Identificador de la asignatura del slot (clave foránea)
hora_inicial	time	Sí	Hora inicial en la que se imparte el slot
hora_final	time	Sí	Hora de finalización del slot
dia_semana	tinyint(3)	Sí	Día de la semana (0-4/L-V) en el que se imparte
id_actividad	bigint(20)	Sí	Identificador de la actividad que indica el slot
num_grupo_actividad	bigint(20)	No	Número del grupo de la actividad a la que pertenece
slot_minimo	float(18,2)	No	Tamaño del slot del horario
color	varchar(7)	Sí	Color con el que aparecerá en el horario
id_aula	bigint(20)	Sí	Identificador del aula en el que se impartirá

Tabla planactividades

Tabla que indica la planificación docente de una actividad.

Tabla 5.12: Estructura de la tabla planactividades

Columna	Tipo	Nulo	Descripción
<i>id</i>	int(11)	No	Identificador autoincremental de la tabla
id_plandocente	int(11)	No	Identificador del plan docente al que pertenece
id_actividad	bigint(20)	No	Identificador de la actividad a la que pertenece
horas	int(10)	No	Número de horas de la planificación para el curso
grupos	int(10)	No	Número de grupos que tendrá la actividad
horas_semanales	int(10)	No	Número de horas semanales que se impartirán
alternas	tinyint(1)	Sí	Si se impartirá o no en semanas alternas

Tabla planesdocentes

Tabla que almacena la información de un plan docente de una asignatura.

Tabla 5.13: Estructura de la tabla planesdocentes

Columna	Tipo	Nulo	Descripción
<i>id</i>	int(11)	No	Identificador autoincremental de la tabla
id_asignatura	int(11)	No	Identificador de la asignatura a la que pertenece
id_curso	int(11)	No	Identificador del curso al que pertenece

Tabla titulaciones

Tabla que almacena la información de una titulación.

Tabla 5.14: Estructura de la tabla titulaciones

Columna	Tipo	Nulo	Descripción
<i>id</i>	int(11)	No	Identificador autoincremental de la tabla

Tabla 5.14: Estructura de la tabla titulaciones (continúa)

Columna	Tipo	Nulo	Descripción
codigo	varchar(4)	No	Código de la titulación utilizado por la universidad
nombre	varchar(200)	No	Nombre de la titulación
creditos	int(10)	No	Número de créditos totales
num_cursos	int(10)	No	Número de cursos que tendrá la titulación

Tabla usuarios

Tabla que almacena la información de los usuarios del sistema.

Tabla 5.15: Estructura de la tabla users

Columna	Tipo	Nulo	Descripción
id	int(11)	No	Identificador autoincremental de la tabla
password	varchar(255)	No	Password codificado en md5
nombre	varchar(50)	No	Nombre del usuario
apellidos	varchar(50)	No	Apellidos del usuario
dni	varchar(9)	No	DNI del usuario
email	varchar(30)	No	email del usuario
id_titulacion	int(11)	No	Titulación a la que pertenece el alumno (si el usuario es alumno)
level	int(10)	No	Nivel de privilegios del usuario 0:admin, 1:planificador, 2:profesor, 3:alumno

Capítulo 6

Implementación

Llegada la etapa de implementación, hay que transformar a código lo analizado y diseñado en etapas anteriores. La petición que se nos hacía era que fuera una aplicación web, para ello se toman una serie de decisiones en cuanto a herramientas y lenguajes que se detallarán a continuación.

6.1. Lenguajes

Durante la carrera hay muy poca formación en cuanto a desarrollo web se refiere, solo una asignatura, que toca por encima el desarrollo en el lado del servidor con algunas nociones de PHP. La razón por la cual se elige entonces PHP para el desarrollo del proyecto es ampliar los conocimientos en dicho lenguaje, además de ser uno de los más usados, lo que implica que será sencillo encontrar solución a los posibles problemas que vayan apareciendo debido a la amplia documentación que hay disponible.

PHP es un lenguaje fácil de aprender, dado que posee mucha similitud con lenguajes como C, que si han sido aprendidos durante la carrera.

Tras comenzar el aprendizaje del lenguaje, surge el problema de que para hacer el desarrollo mínimamente organizado habría que implementar una serie de clases base que nos ayudaran con la implementación del sistema. Esto implica un trabajo tedioso que puede ahorrarse utilizando algún framework que nos proporcione herramientas que hagan mucho más fácil el trabajo. Es por ello que se decide utilizar CodeIgniter.

CodeIgniter es un framework escrito en PHP, basado en el patrón arquitectónico MVC, a diferencia de otros frameworks existentes para PHP, es una herramienta realmente ligera, poco intrusiva y que facilita muchísimo el trabajo. Para ello pone a disposición algunas librerías y helpers, aumentando notablemente la productividad del desarrollador.

Otra gran ventaja de CodeIgniter es su documentación y su comunidad de desarrolladores, además de la Wiki en la que los usuarios van publicando plugins que pueden ser útiles para nuestro trabajo.

Uno de los problemas que encontramos con CodeIgniter es que a diferencia de otros frameworks MVC, no disponían de un ORM, es decir un sistema de persistencia que nos permitiera obtener elementos de la base de datos y transformarlos en objetos de nuestro sistema. Para ello se decide utilizar Doctrine, un ORM fácilmente integrable con CodeIgniter, y que nos permitía tener una base de datos virtual sobre nuestro sistema.

Pero la parte del servidor no es la única que hay que implementar, además necesitamos un lenguaje para las vistas. Para ello se elige XHTML, lenguaje de marcado ampliamente usado y recomendado por la W3C.

Otro objetivo es tener una buena interacción con el usuario, para ello es necesario el uso de un lenguaje que pueda interactuar con el DOM y modificarlo sin necesidad de pasar por el servidor, para ello elegimos JavaScript, soportado por la inmensa mayoría de navegadores. Se hace necesario para mejorar la interacción, el uso de una librería que facilite el trabajo con este lenguaje, para ello elegimos jQuery, la librería de JavaScript más usada, con multitud de plugins y una documentación muy bien estructurada.

6.2. Extensiones y librerías

El proyecto requiere la realización de algunas tareas complejas, como la generación de informes y la configuración de horarios, para ello se han utilizado una serie de librerías para obtener algunas funcionalidades difíciles de implementar. Éstas son:

- **FullCalendar** - Plugin para jQuery que permite mostrar un calendario interactivo, con una API a disposición del desarrollador para responder a multitud de eventos, lo que permite una máxima personalización.
- **jQueryUI** - Plugin para jQuery con una gran variedad de widgets para mejorar la interacción con el usuario.
- **Farbtastic** - Plugin para jQuery que nos permite integrar un selector de color en una página.
- **FPDF** - Librería para PHP para la generación de documentos PDF.
- **PHPMailer** - Librería para PHP que facilita el envío de correos electrónicos.

6.3. Herramientas utilizadas

Para el desarrollo del proyecto se hace necesario el uso de una serie de herramientas, como editores de código, sistemas de control de versiones, etc. A continuación se detallarán todas las herramientas usadas en este proyecto.

La herramienta principal que se ha utilizado ha sido un IDE, en este caso NetBeans en su versión PHP. NetBeans es un entorno escrito en Java y pensado en un principio para desarrollar en este mismo lenguaje, pero conforme ha avanzado el tiempo se ha ido ampliando a más lenguajes, como por ejemplo PHP. La integración con este último es perfecta, proporcionando útiles herramientas como el autocompletado.

Para la detección de errores se hace casi obligado el uso de un *debugger*, en este caso hemos usado XDebugger que viene integrado en NetBeans, permitiendo utilizar puntos de ruptura en el código para comprobar el estado del sistema en un momento dado.

Para el despliegue de la aplicación se ha utilizado un entorno compuesto por un servidor Apache, base de datos MySQL y el intérprete de PHP, todo ello sobre un sistema GNU/Linux.

Otra herramienta utilizada que facilita el trabajo enormemente ha sido **Git**. Git es un sistema de control de versiones que facilita el desarrollo colaborativo y el mantenimiento de un software, versionando

todos los cambios que se vayan produciendo en el código. Esto permite que si queremos volver a una versión anterior del sistema podamos hacerlo sin problema alguno, además de la creación de ramas de desarrollo, pudiendo fusionar ramas sin problema alguno.

6.4. Detalles de la implementación de la arquitectura del sistema

6.4.1. Capa modelo

Como hemos dicho, se ha utilizado la librería Doctrine como ORM. Esta librería sustituye completamente a la capa modelo del MVC de CodeIgniter, para ello, cada tabla de la base de datos se corresponde con un modelo. Para construir un modelo definimos sus atributos en una clase, además de sus relaciones, de esta forma, Doctrine generará automáticamente las tablas de la base de datos a partir de los modelos, aplicando todas las reglas de integridad definidas.

Un ejemplo de un modelo es el siguiente, que corresponde al de la tabla titulaciones:

```
1 class Titulacion extends Doctrine_Record
2 {
3     public function setTableDefinition()
4     {
5         $this->setTableName('titulaciones');
6         $this->hasColumn('id', 'integer', 4, array(
7             'type' => 'integer',
8             'length' => 4,
9             'primary' => true,
10            'autoincrement' => true,
11            'unsigned' => false,
12            'fixed' => false
13        ));
14        $this->hasColumn('codigo', 'string', 4, array(
15            'minlength' => 4,
16            'length' => 4,
17            'notnull',
18            'notblank',
19            'unique',
20            'regexp' => '/[0-9]{4}/',
21            'unsigned' => false
22        ));
23        $this->hasColumn('nombre', 'string', 200, array(
24            'type' => 'string',
25            'minlength' => 5,
26            'length' => 200,
27            'notnull' => true,
28            'unique' => true,
29            'notblank' => true,
30            'unsigned' => false
31        ));
32        $this->hasColumn('creditos', 'integer', 4, array(
33            'type' => 'integer',
34            'length' => 4,
35            'unsigned' => true,
36            'notnull' => true,
37            'notblank' => true
```

```

38         ));
39     $this->hasColumn('num_cursos', 'integer', 4, array(
40         'type' => 'integer',
41         'length' => 4,
42         'unsigned' => true,
43         'notnull' => true,
44         'notblank' => true
45     ));
46 }
47
48 public function getPlanificacion($id_curso)
49 {
50     $asignaturas = $this->asignaturas;
51     $salida_total = array();
52     foreach($asignaturas as $asignatura)
53     {
54         $q = Doctrine_Query::create()->select('c.*, p.*, a.descripcion'
55             )
56             ->from('PlanActividad p')
57             ->innerJoin('p.plandocente c')
58             ->innerJoin('p.actividad a')
59             ->where('c.id_curso = ? AND c.id_asignatura = ?', array
60                 ($id_curso, $asignatura->id));
61         $resultado = $q->execute();
62         $salida = array();
63         $salida[0] = $asignatura->nombre;
64         foreach($resultado as $actividad)
65         {
66             $salida[$actividad->id_actividad] = array($actividad->horas
67                 , $actividad->grupos, $actividad->horas_semanales);
68         }
69         $salida_total[] = $salida;
70     }
71
72     return $salida_total;
73 }
74
75 public function setUp()
76 {
77     parent::setUp();
78     $this->hasMany('Asignatura as asignaturas', array('local' => 'id', '
79         foreign' => 'titulacion_id', 'onDelete' => 'CASCADE'));
80 }
81 }

```

Como se puede ver en el código, se definen todos los atributos y sus reglas de integridad. En el método `setUp` definimos las relaciones, y además podemos definir nuestros propios métodos que devuelvan información personalizada de la base de datos.

6.4.2. Capa controlador

La capa controlador es la principal del sistema, la que recibe las peticiones, las procesa y las pasa a la vista para renderizar la página pedida. Al igual que en el modelo existe uno por cada subsistema.

Un controlador está compuesto de acciones, y cada una de ellas es llamada desde una URL del navegador, el controlador procesa la petición y realiza la lógica necesaria antes de devolver una respuesta, la estructura de un controlador es la siguiente:

```
1 class Titulaciones extends MY_Controller {
2
3     function __construct() {
4         parent::__construct();
5         $this->titulaciones_table = Doctrine::getTable('Titulacion');
6         $this->asignaturas_table = Doctrine::getTable('Asignatura');
7         $this->layout = '';
8         $this->notices = '';
9         $this->alerts = '';
10        $this->_filter(array('add', 'create', 'delete', 'edit', 'update', '
            show_informes', 'show', 'exportar_planificacion'), array($this,
            'authenticate'), 1);
11    }
12
13    public function index() {
14
15        $titulaciones = $this -> titulaciones_table -> findAll();
16
17        //Conseguimos los items mediante el modelo
18        $data['titulaciones'] = $titulaciones;
19        $data['page_title'] = 'INDEX TITULACIONES';
20        $data['enlace'] = 'titulaciones/show/';
21        if($this -> input -> post('js') == '1') {
22            unset($this -> layout);
23            $this -> load -> view('titulaciones/_titulaciones', $data);
24        } else {
25            //Mostramos
26            $this -> load -> view('titulaciones/index', $data);
27        }
28    }
}
```

Tenemos un constructor que es invocado en cada petición e inicializa algunos parámetros necesarios, y luego tenemos una acción, en este caso index, que busca todas las titulaciones en la base de datos y las pasa a la vista.

6.4.3. Capa vista

Esta capa es la que realmente ve el usuario, por tanto no es menos importante que las demás, aquí se utiliza un *layout* o plantilla por defecto que renderiza la parte que es común a todas las páginas, de forma que no se repite código en cada una de las vistas. Esta estructura común es la siguiente:

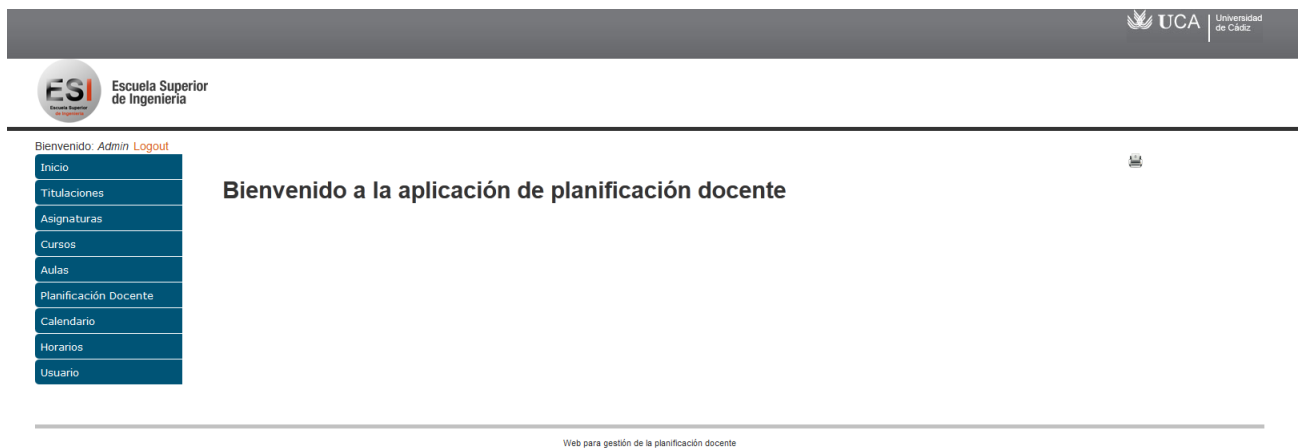


Figura 6.1: Captura de la estructura de una página

En esta estructura tenemos un menú a la izquierda, una cabecera y un pie de página, además de un panel sobre el menú que indica el usuario que está conectado y permite su salida del sistema.

Todas las páginas siguen un estilo similar, por ejemplo una página que contiene una tabla de elementos, como esta con el listado de titulaciones:

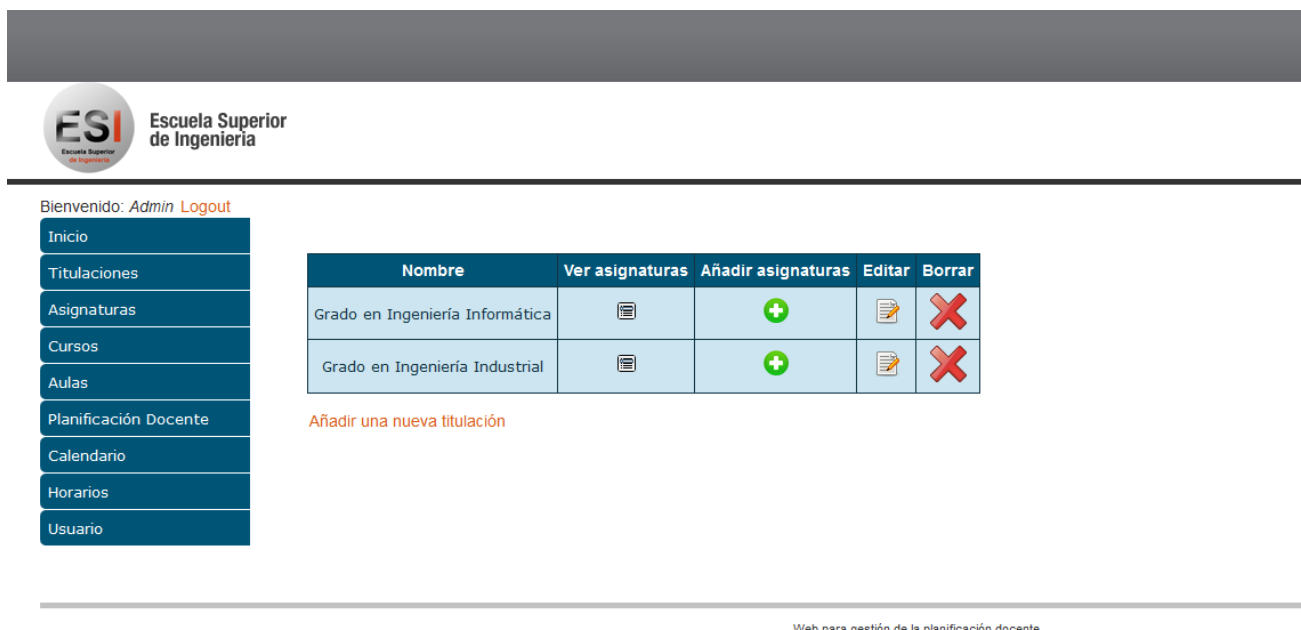


Figura 6.2: Captura del listado de titulaciones

La parte central del sistema es la gestión de horarios, para ello es necesaria una interacción sencilla con el usuario a la hora de construirlos y que no se convierta en una labor tediosa de realizar. Por ello se

pensó que lo más sencillo sería arrastrar las asignaturas al lugar deseado en el horario, es decir, lo más intuitivo posible, esto es lo que veríamos en una de las páginas de configuración de horarios:

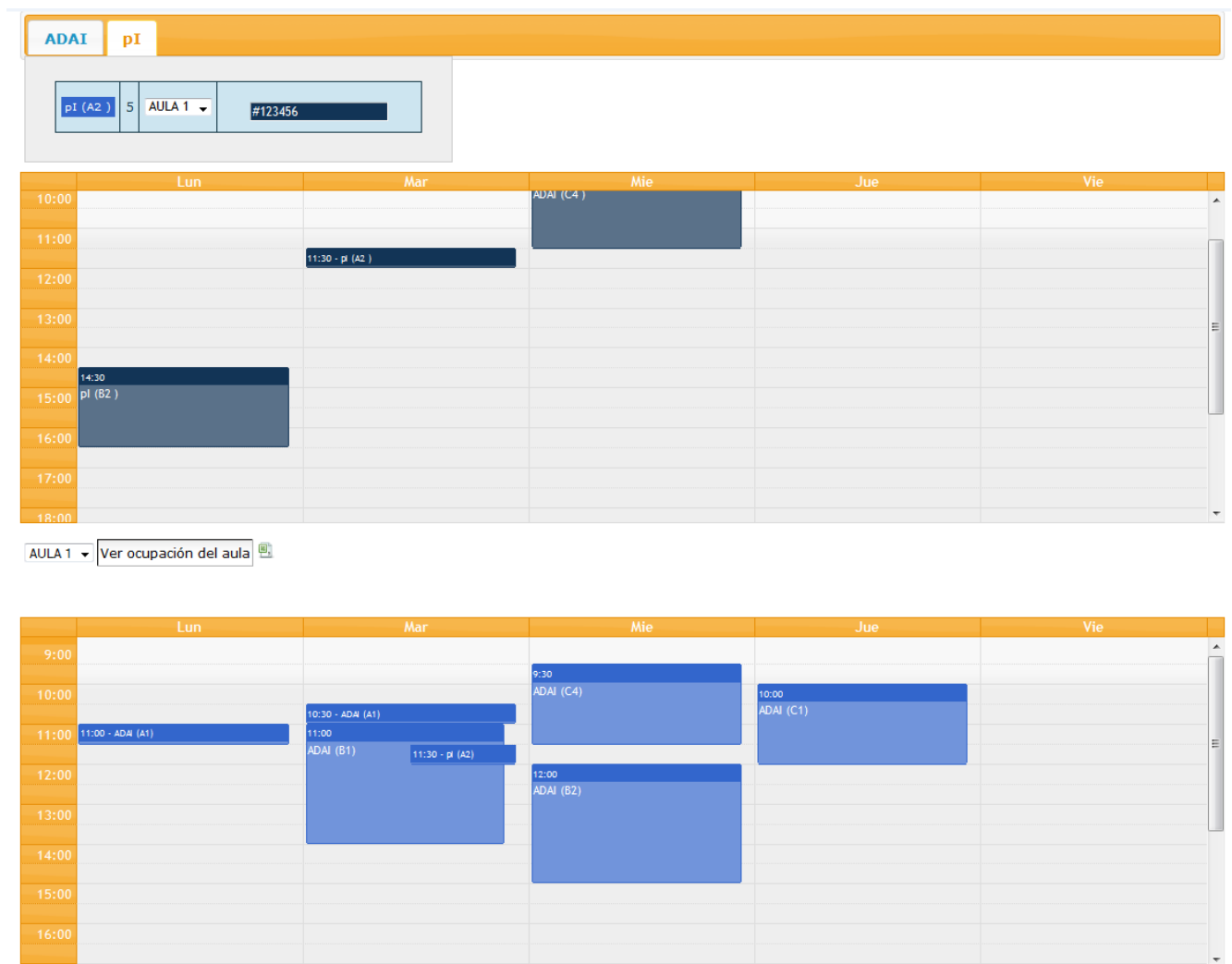


Figura 6.3: Captura de página de horarios

Tenemos tres partes diferenciadas en esta vista, un cuadro superior con las asignaturas aun no asignadas, que se podrán arrastrar al horario. Otra central con el horario de ese grupo, y otro bloque en la parte inferior en el que podemos ver la ocupación del aula que seleccionemos.

A la hora de asignar los slots en el horario hay que tener en cuenta diversos factores, como por ejemplo si el aula está ocupada, o si las asignaturas son solapables, para ello cada vez que se arrastra una asignatura al horario se hace una comprobación mediante la llamada a un servicio, devolviendo el resultado y dependiendo de si es satisfactorio o no, dejar la asignatura en su lugar o deshacer el cambio. Para deshacer el cambio, la API del plugin FullCalendar proporciona una función para invertir el proceso de la última acción realizada, eso hace el trabajo algo más sencillo.

Capítulo 7

Pruebas

Es importante en un sistema como este, y cualquiera, que todo funcione correctamente, para ello es necesario hacer una serie de pruebas para verificar el correcto funcionamiento de la aplicación y que los requisitos se cumplen tal y como fueron especificados.

7.1. Pruebas sobre los datos

Para verificar el correcto funcionamiento de la aplicación se realizaron una serie de pruebas que consistieron en las siguientes:

- **Pruebas de caja negra individual** . Cada vez que se desarrollaba una nueva clase se realizaban pruebas de caja negra, esto significa que lo único que se tiene en cuenta son los datos de entrada y la salida producida, ignorando lo que pasa internamente en el sistema.
- **Pruebas de integración de subsistema** . Al terminar un subsistema se hace necesario probar que la interacción entre todos los elementos de éste es correcta. Estas pruebas nos permiten ver si el intercambio de datos entre Modelo, Vista y Controlador es correcto.
- **Pruebas del sistema o de integración entre subsistemas** . Este tipo de pruebas se realizan sobre sistemas que trabajan de forma conjunta e intercambian información entre sí, se realizaron pruebas para comprobar que esta interacción fuera correcta. Estas pruebas fueron realizadas principalmente a la finalización del desarrollo.

7.2. Especificación del diseño de pruebas

Hay dos fases temporales donde se han realizado las pruebas:

- **Durante el desarrollo de la aplicación:** Esta etapa es la ideal para realizar las pruebas de clase individuales, ya que evitaremos propagar errores a fases posteriores. A medida que se iban desarrollando clases se iba comprobando su funcionalidad mediante la entrada de datos que cumplieran los requisitos de información especificados, así veríamos que salida daba la clase a esos datos para saber si el funcionamiento era el deseado o no, además de ver si los datos se estaban almacenando correctamente en la base de datos.
- **Una vez finalizada la aplicación:** Este es el momento de comprobar que la interacción entre los distintos subsistemas de la aplicación es correcta. Además habría que comprobar la seguridad del sistema, es decir, que un usuario con un nivel de privilegios insuficiente no pudiera acceder a un

subsistema que tuviera un acceso restringido a su rol.

El proceso a seguir para realizar estas pruebas comenzó con la entrada con el usuario administrador a la aplicación, una vez logueado, procedimos a crear un usuario con rol planificador, para poder probar el grueso de subsistemas de la aplicación.

Una vez iniciada la sesión con un usuario de rol planificador, procedimos a probar los subsistemas de titulaciones, asignaturas y cursos, ya que era necesario tener registrados algunos elementos de estas clases para probar los demás subsistemas.

Después de dar de alta varias titulaciones y asignaturas, y un curso, se procedió a introducir planes docentes para esas asignaturas, de forma que tuvieramos horas asignadas para probar los horarios más adelante.

A continuación, se probó el subsistema de gestión de calendarios, que es independiente de los planes docentes pero no del curso, se probó a introducir varios eventos, y a eliminar algunos.

Se probó también la gestión de aulas, introduciendo algunas, eliminando y editando otras.

Hecho todo esto se podía pasar a la parte de gestión de horarios, comprobando la creación de grupos, edición de horarios, y chequeo de horas asignadas. Además de comprobar una vez rellenados algunos horarios, la gestión de informes de asignatura, comprobando que se generaran correctamente con los datos introducidos en los horarios.

Hecho todo esto también se hacía necesario probar la importación y exportación de los distintos elementos del sistema, como asignaturas, calendario o horarios.

Todas estas pruebas correspondían al perfil de planificador, pero también había que probar el perfil de alumno y el de profesor. Ambos son muy similares, ya que pueden ver casi lo mismo. Principalmente pueden configurar un horario, seleccionando una serie de asignaturas y grupos, aunque en el caso del alumno había que verificar que sólo pudiera ver las asignaturas de su titulación. Además el profesor tenía una funcionalidad extra que consistía en visualizar la planificación docente de una titulación.

Además también el rol administrador tiene su propia funcionalidad, que es la de crear usuarios, eliminarlos o editarlos, cosa que tuvo que ser probada también.

7.3. Especificación de los procedimientos de prueba

Se realizaron pruebas sobre sistema operativo GNU/Linux en la distribución *Ubuntu 11.04* y en Microsoft Windows 7 sobre los siguientes navegadores:

- Mozilla Firefox 8
- Internet Explorer 8
- Opera 11.60

- Google Chrome 15.0

Algunos problemas encontrados fueron con algunas propiedades de CSS y parte del código JavaScript, que principalmente en Internet Explorer no funcionaban correctamente, se pudieron solventar estos problemas con algunos parches encontrados.

7.4. Documentación de la ejecución de las pruebas

- **Histórico de pruebas:** Muchos errores encontrados durante las pruebas fueron provocados por algún error simple en el código, que en ocasiones eran difíciles de encontrar por aparecer mensajes de error con poca información. Este tipo de casos se daba especialmente en JavaScript, ya que seguir la ejecución del código era complicado debido a que su consola de errores apenas daba información. También con la librería Doctrine hubo algunos problemas ya que a pesar de que si que mostraba errores, estos eran tremendamente crípticos, siendo en ocasiones provocados simplemente por la falta de declaración de un campo en un modelo.
- **Informe de incidentes ocurridos:** No hay incidencias que destacar. En cualquier caso, si se quisiera continuar el desarrollo, sería recomendable seguir usando el repositorio de Git, ya sea bien incorporándose al desarrollo, o bien haciendo un fork del repositorio, de forma que en caso de errores fuera posible volver a una versión anterior sin problema alguno.

7.5. Herramientas utilizadas para las pruebas

Al principio del desarrollo para trazar la funcionalidad de las distintas acciones, nos veíamos obligados a usar *echos* de las variables, mostrando su valor por el navegador, siendo este método un poco engorroso. Usando NetBeans se descubre que trae incorporada una herramienta llamada *XDebug*, que conjuntamente con una extensión del navegador Firefox llamada *EasyXDebug* facilitaba mucho el desarrollo y las pruebas, ya que nos permitía incorporar puntos de ruptura en el código, en los que usando NetBeans podíamos comprobar el estado de las variables en ese momento, sin necesidad de utilizar los ya mencionados *echo*.

Otra herramienta imprescindible para las pruebas en el desarrollo web es la extensión *FireBug*, de Mozilla Firefox que proporciona una consola de eventos de JavaScript, pudiendo además utilizar la funcionalidad de los puntos de ruptura en el código del cliente.

Capítulo 8

Conclusiones

En este capítulo se comentarán las conclusiones personales alcanzadas, así como las posibles ampliaciones futuras que se le podrían hacer a la aplicación.

8.1. Opinión personal

Con este proyecto se han abarcado aspectos tanto de software de gestión como de aplicación web, conocimientos que durante la carrera se dan de forma muy básica, por lo que proyectos como este pueden ayudarme a ampliar conocimientos para desarrollos y trabajos futuros.

Realmente es la primera vez que me enfrento a un desarrollo web relativamente grande, ya que mi conocimiento y experiencia no pasaba de la realización de algunas páginas estáticas mediante HTML. Este es, por tanto, el primer acercamiento real al lenguaje PHP, del que he descubierto su potencia en este proyecto. Es cierto que podría haber evitado usar algún framework y utilizar simplemente el lenguaje sin ninguna ayuda, pero creo que sólo hubiera hecho la labor más tediosa y aburrida, y considero que el uso de un framework debería ser considerado obligatorio por cualquier programador a la hora de realizar una aplicación web, sea en PHP o en cualquier otro lenguaje.

En cuanto a JavaScript, si bien es cierto que ya lo había usado alguna otra vez, realmente nunca había probado alguna librería como jQuery, y me ha sorprendido gratamente su potencia y facilidad de uso, y lo que se puede hacer mejorando notablemente la experiencia del usuario.

En cuanto a los sistemas de control de versiones, ya había tenido experiencia con Subversion, pero Git sorprende por su versatilidad, mostrándose mucho más fuerte a la hora de realizar desarrollos colaborativos. Se utilizó GitHub como servidor Git, ya que disponía de una amplia comunidad de usuarios y parecía ser de los más recomendados en la web, además de que disponía de un sistema de seguimiento de tareas muy bueno y fácil de usar.

En cuanto a \LaTeX no es la primera vez que lo he usado, pero esta memoria me ha servido para ampliar un poco más mi conocimiento de este lenguaje.

8.2. Ampliaciones futuras

Este proyecto está sujeto a cambios, ya que es posible que cambie la forma en la que se plantea la planificación docente de una asignatura, y aunque la aplicación se ha intentado parametrizar lo máximo

posible para poder personalizar muchos aspectos, es posible que sea necesario realizar cambios.

Entre las posibles ampliaciones que se pueden hacer está el que un alumno pueda loguearse con su usuario habitual del campus virtual, integrándolo con el LDAP de la Universidad, no se ha hecho debido a que el objetivo principal de la aplicación es ser usada por un usuario planificador para la configuración, teniendo el alumno una funcionalidad mínima.

Otra posible mejora sería una cierta automatización en la creación de los horarios, por ejemplo proponiendo un profesor su horario preferente y generándose el mejor horario posible, pudiendo ser modificado luego. En este aspecto también estaría bien la posibilidad de que un alumno propusiera una configuración mejor para un horario.

Bibliografía

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

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

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”). To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

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

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

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with . . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.