

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



TRABAJO FIN DE GRADO

**Gestión académica y de comunicación directa entre centros
educativos y familias**

Marcos Muñoz Iglesias

JUNIO 2015

Tutor: Simone Santini

**Gestión académica y de comunicación directa entre centros
educativos y familias**

AUTOR: Marcos Muñoz Iglesias

TUTOR: Simone Santini

**Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid**

Fecha: Junio 2015

Resumen

Un sistema de gestión académica y de comunicación directa entre centros educativos y familias es una aplicación que permite que los distintos miembros de la comunidad educativa, como son profesores, alumnos y familia de éstos, estén en continuo contacto para conocer las novedades que involucran a los alumnos.

Actualmente existen herramientas en el mercado que ayudan a los centros educativos a gestionar su personal docente, sus alumnos, sus cuentas financieras, etc. Aun así, no todos los centros educativos hacen uso de estas herramientas por diversos motivos.

En relación con las necesidades que puedan llegar a tener estos centros se ha desarrollado **SchoolPipe**, una herramienta web que pretende integrar las nuevas tecnologías en la educación, hacer más fácil la gestión de un centro escolar y mejorar la comunicación entre las familias y los profesores. Esta herramienta es intuitiva y fácil de usar: no provoca dudas en el usuario.

La herramienta, además de prestar el servicio de gestión y comunicación de un colegio, está implementada alrededor de un núcleo genérico que le permita ser implantada en otro sistema configurándola a medida.

El sistema es utilizado mediante clientes web por sus características y por ser uno de los medios más utilizados en el mundo de Internet. Los motivos por los que se ha elegido esta tecnología son su bajo coste y su gran facilidad para ser implantado en diferentes navegadores.

Como principal línea de trabajo futuro queda implementar más funcionalidad que pueda servir de ayuda a los centros educativos además de mejorar la funcionalidad de esta primera versión. En particular, por ejemplo, gestión más detallada de exámenes, datos de matriculación completos, etc.

Palabras clave: Sistemas de información, Tecnologías de la Información, Gestión escolar, flujo de información, comunicación, educación online, plataformas, LMS, Web 2.0, SIMS, TIC, e-learning

Abstract

A School Information Management System (SIMS) is a platform that facilitates the management of educational institutions that increases direct communication between all education players such as school center teachers, families and students, in order to improve the teaching and learning processes.

Currently, there are some tools in the market that help school centers a manage educators, students, financial accounts, etc. Even so, not all educational institutions use these tools for a variety of reasons, including cost, difficulty of use, resistance to change, etc.

In order to fulfill the needs of the educational institutions, we have developed **SchoolPipe**, a web tool that pretends to integrate new technologies in education, facilitate school centers management and improve the communication between families and teachers. This intuitive tool is easy to use and to install and is designed to provide a pleasant and intuitive user experience.

The tool can also be used in other scenarios and markets, thanks to its general-purpose kernel, which allows its implementation in other systems in a customized way.

The characteristics of the system suggested its implementation as a server-side application accessed through web browsers one of the most important and most used media in internet world. This yielded the additional advantages of low cost and high accessibility.

This work results in a complete and usable application. Nevertheless as all projects developed in the dynamic environment of the web, this application is work in progress. Envisioned future development include extending its functionality, improving the management of test and adding the possibilities of meaning academic fees.

Keywords: Information Systems; Information Technology; Education Management; Information flow, communication, online education, LMS, Web 2.0, SISM, TIC, e-learning

Agradecimientos

Mucho ha llovido hasta el día de hoy. Desde aquel mes de Septiembre del año 2009 en el que esta gran etapa empezaba en mi vida: el maravillo mundo de la informática que tanto me ha dado, tanto me ha quitado y tanto me dará en el futuro.

Quiero empezar este punto agradeciendo a mis padres la oportunidad de haber estudiado en la universidad. Siempre me apoyaron pese a todo y por ello les estoy muy agradecido. Ha habido temporadas en los que me pasaba meses sin verlos por estar estudiando, haciendo prácticas y, por qué no decirlo, con los compañeros de carrera. Y puede parecer extraño leer que “he estado meses sin ver a mis padres”. Pero no lo es cuando en 2009 uno decide irse del “pueblo” donde ha pasado los últimos ocho años de su vida para venir a la sierra de Madrid con mis tíos. En especial, quería agradecer a mi tía Mamen por toooooo lo que ha hecho por mí inculcándome la universidad desde bien joven para intentar llegar a ser como ella es, una crack en todo. Gracias también a mi tío y a mis primos pequeños por aguantarme durante seis largos años. Aunque quizá también deberían de darme ellos las gracias a mi por aguantar tanto llanto inoportuno de niño pequeño en épocas de exámenes y entregas.

No me puedo dejar a mi hermana, siempre tan pendiente de mí, más que yo de ella. Gracias por haber estado siempre ahí, incluso cuando te fuiste a Londres, estabas ahí para lo que necesitara.

Agradecer también al resto de mi familia su comprensión y apoyo en los duros y no tan duros momentos que esta carrera te hace vivir.

Toda esa panda de 5 compañeros (Jose, Luci, Diego, Rafa y Borja) que empezamos juntos la carrera, gracias. Gracias a todos por aquellos inicios.

Todos los que están ahora terminando conmigo, que no son pocos y que nos hemos hecho íntimos en los últimos 2-3 años, llegando a formar el gran Ciudaddeanos Team... ¡¡Menudo equipazo!! En especial gracias a Javi, Sergio, Carlos, Mario, Gallas, Alberto, Raúl,... por haberme dado tantos momentos de compañía, sufrimiento y alegrías. GRACIAS chicos.

Quiero destacar al capitán del equipo, Fernando, por su apoyo durante los últimos 3 años de carrera y aquellas asignaturas donde los dos trabajamos de forma muy cómoda y que sacamos con creces. ¡Gracias amigo!

Gracias a Simone por haberme apoyado en este proyecto que yo mismo saqué de la manga y que pocos profesores quisieron aceptar. Gracias por tu tiempo y dedicación.

Gracias a Idoia, por, sin necesidad alguna, ayudarme a controlar este documento con una buena plantilla y con unas buenas correcciones a tiempo. Gracias.

Gracias a todos los profesores que han dejado huella sobre mí con sus enseñanzas, consejos, riñas, correcciones... Todos vosotros, uno a uno, habéis conseguido que aprenda algo de cada uno de vosotros.

Hace ya 2 años empecé a trabajar con el maravilloso equipo de Aprendum. Gracias a todos mis compañeros por aguantar mis cotilleos, bromas, madridismo... En especial quiero acordarme de mis actuales compañeros del departamento técnico, Aída y Alexandre, y mis excompañeros, Fernando y Dani por iniciarme en el mundo laboral y confiar en mí cuando apenas era un cachorro. Gracias al señor CTO de Aprendum, El señor Don Alejandro Yusta por ser mi jefe y además, pese a la edad, mi amigo. Gracias Alex, por todo lo que me has enseñado, apoyado y por todo lo que hemos bromeado juntos.

Gracias a Miguel por ayudarme con los trabajos de economía y hacerme reír mucho, a Sergito, a Sara, a Lorena, a Ada, a Daniele, a Melisa, a Juan, a Gregorio... en fin, Gracias a todo el equipo de Aprendum.

Gracias también a Alexandre, última incorporación del departamento técnico de Aprendum que tanto me ha ayudado a realizar este proyecto con sus pruebas para romper todo el sistema y sus ayudas con determinados conceptos.

Por último y no por ello menos importante, quiero darle las gracias a la que, quizá haya sido la persona más importante durante toda la carrera desde el año 2011. Gracias a Esther por absolutamente todo. Por todo su apoyo durante aquel fatídico año cuando, con su ayuda, dedicación e insistencia, conseguimos, los dos, sacar todas las asignaturas adelante. Desde entonces he cambiado mi forma de afrontar las cosas y ahora voy más de cara. Gracias por compartir tu vida conmigo y gracias por tanto. Hay tanto que decir de ella, que por mucho que quiera sintetizar, sería imposible. Por tanto, un enorme GRACIAS.

Seguro que me dejo muchos compañeros y/o personas que me han dado ánimos, apoyo y ayuda durante todos estos años de carrera, para ellos, aunque no los mencione uno a uno, otro GRACIAS.

Índice de Contenidos

1	INTRODUCCIÓN	1
1.1	MARCO DEL PROYECTO.....	1
1.2	MOTIVACIÓN	1
1.3	OBJETIVOS.....	2
1.4	ESTRUCTURA DEL DOCUMENTO	3
2	ESTADO DEL ARTE.....	5
2.1	PLATAFORMAS EDUCATIVAS Y SUS FUNCIONES TÍPICAS	5
2.2	ERP (ENTERPRISE RESOURCE PLANNING)	5
2.3	HERRAMIENTAS UTILIZADAS ACTUALMENTE PARA GESTIONAR CENTROS EDUCATIVOS.....	6
2.4	PLATAFORMAS E-LEARNING.....	8
2.5	OPINIONES Y CONCLUSIONES.....	8
3	OBJETIVOS/FUNCIONALIDADES.....	9
3.1	OBJETIVOS ESPECÍFICOS Y FUNCIONALIDADES	9
3.1.1	<i>Objetivos propios de la aplicación</i>	<i>9</i>
3.1.2	<i>Objetivos genéricos</i>	<i>10</i>
4	DEFINICIÓN DEL PROYECTO	11
4.1	METODOLOGÍA.....	11
4.2	HERRAMIENTAS USADAS	13
5	ANÁLISIS	19
5.1	INTRODUCCIÓN.....	19
5.2	CASOS DE USO	19
5.3	CATÁLOGO DE REQUISITOS	22
5.3.1	<i>Requisitos Funcionales</i>	<i>22</i>
5.3.2	<i>Requisitos No funcionales.....</i>	<i>26</i>
6	DISEÑO E IMPLEMENTACIÓN.....	27
6.1	INTRODUCCIÓN.....	27
6.2	ARQUITECTURA DE LA APLICACIÓN.....	27
6.3	BASE DE DATOS.....	28
6.3.1	<i>Conexión a la base de datos</i>	<i>36</i>
6.4	INTERFAZ DE USUARIO.....	36
6.5	ESTRUCTURA DE LA APLICACIÓN	37
6.5.1	<i>Diagrama de clases de la aplicación</i>	<i>38</i>
6.5.2	<i>Diseño e implementación del modelo de la aplicación</i>	<i>39</i>
6.5.3	<i>Implementación de modelos más destacados.....</i>	<i>42</i>
6.6	GENERACIÓN DE PDF.....	50
7	VALIDACIÓN Y VERIFICACIÓN	51
7.1	ALCANCE DE LAS PRUEBAS	51
7.1.1	<i>Funcionalidad.....</i>	<i>51</i>

7.1.2	<i>Compatibilidad</i>	51
7.1.3	<i>Accesibilidad</i>	51
7.2	VERIFICACIÓN.....	51
7.2.1	<i>Estrategia de pruebas y desarrollo de las mismas</i>	51
7.3	VALIDACIÓN.....	53
7.3.1	<i>Estrategia y desarrollo de validación</i>	53
8	EXTENSIBILIDAD	55
9	EVALUACIÓN	57
9.1	EVALUACIÓN	57
9.2	BENEFICIOS DE LA APLICACIÓN	57
10	CONCLUSIONES Y LÍNEAS FUTURAS	59
10.1	CONCLUSIONES.....	59
10.2	TRABAJO FUTURO.....	59
11	REFERENCIAS	61
	ANEXO A: DIAGRAMA ESTRUCTURAL DE LA BASE DE DATOS	A
	ANEXO B: RESUMEN DE CÓMO TRABAJA EL SISTEMA DE REJILLA DE BOOTSTRAP EN FUNCIÓN DE LOS DISPOSITIVOS	B
	ANEXO C: ALGORITMOS DISEÑADOS EN LA APLICACIÓN Y OTRAS CAPTURAS DE INTERÉS	C
	ANEXO D: GESTIÓN DEL MENÚ	H
	ANEXO E: DESARROLLO DE PRUEBAS ATENDIENDO A LOS REQUISITOS	I
	ANEXO F: RESULTADOS DE LAS PRUEBAS DE SQL INJECTION Y DE VULNERABILIDADES SOBRE SERVIDOR	Q

Índice de Figuras

FIGURA 1. DIAGRAMA DE GANTT.....	12
FIGURA 2. CICLO DE VIDA EN CASCADA CON REALIMENTACIÓN.....	12
FIGURA 3. COMPONENTES DE UN LAMP.....	14
FIGURA 4. ALGUNOS SERVICIOS DE AWS.....	15
FIGURA 5. DIFERENCIA ENTRE UNA APLICACIÓN WEB CON AJAX Y UNA PLATAFORMA WEB CLÁSICA	16
FIGURA 6. EJEMPLO DE UNA PÁGINA WEB USANDO EL FRAMEWORK DE BOOTSTRAP EN MOZILLA FIREFOX.....	17
FIGURA 7. EJEMPLO DEL EDITOR CKEDITOR.....	18
FIGURA 8. DIAGRAMA DE CASO DE USO 1. USUARIO SIN INICIAR SESIÓN EN EL SISTEMA.....	20
FIGURA 9. DIAGRAMA DE CASO DE USO 2. ACCIONES COMUNES DE TODOS LOS USUARIOS	20
FIGURA 10. DIAGRAMA DE CASO DE USO 3. ACCIONES NO COMUNES DE LOS USUARIOS	21
FIGURA 11. DIFERENTES NAVEGADORES WEB	22
FIGURA 12. ARQUITECTURA CLIENTE-SERVIDOR DE UNA APLICACIÓN TIPO LAMP.....	28
FIGURA 13. MODELO DE 3 CAPAS EN UNA ARQUITECTURA CLIENTE-SERVIDOR.....	28
FIGURA 14. ESTRUCTURA DE LA PÁGINA WEB.....	36
FIGURA 15. ESTRUCTURA DE LA HOME DE SCHOOLPIPE	37
FIGURA 16. DIAGRAMA DE CLASES DE LA APLICACIÓN.....	38
FIGURA 17. INDEX DE LA APLICACIÓN.....	40
FIGURA 18 ESTRUCTURA Y PASOS DEL MODELO-VISTA-CONTROLADOR	42
FIGURA 19. OPCIONES DE FUNCIONALIDAD DE LA GESTIÓN Y ADMINISTRACIÓN DE TABLAS	42
FIGURA 20. ENTIDAD QUE GESTIONA ASIGNATURAS.....	43
FIGURA 21. PANTALLA DE ADMINISTRACIÓN DE TABLAS.....	44
FIGURA 22. LISTADO DE DATOS CON 10 REGISTROS POR PÁGINA.....	44

FIGURA 23. LISTADO DE DATOS CON 5 REGISTROS POR PÁGINA	45
FIGURA 24. PÁGINA DE INSERCIÓN Y EDICIÓN DE REGISTROS	45
FIGURA 25. TIPO DE CAMPO <i>DBLINK</i>	45
FIGURA 26. TIPO DE CAMPO <i>DBLINKSPECIAL</i>	46
FIGURA 27. TIPO DE CAMPO <i>LIST</i>	46
FIGURA 28. FUNCIONES QUE AÑADEN Y EDITAR UN EVENTO EXISTENTE	47
FIGURA 29. VENTANA MODAL PARA RELLENAR LOS DATOS DE UN EVENTO	48
FIGURA 30. PÁGINA DE COMUNICADOS	48
FIGURA 31. PANTALLA DE RESPUESTA DE UN MENSAJE	49
FIGURA 32. PÁGINA DE GESTIÓN DE EXÁMENES	50
FIGURA 33. PÁGINA DE CREACIÓN DE EXÁMENES.	50
FIGURA 34. PÁGINA DE CALIFICACIONES DE EXAMEN	50
FIGURA 35. CAPTURA DE LA PÁGINA PRINCIPAL DE LA APLICACIÓN DE GESTIÓN DE INVITADOS A UNA BODA	55
FIGURA 36. ESQUEMA RELACIONAL DE <i>SCHOOLPIPE_MASTER</i>	A
FIGURA 37. CLASIFICACIÓN DE OPCIONES DE BOOTSTRAP SEGÚN EL DISPOSITIVO.....	B
FIGURA 38. DISTRIBUCIÓN DE COLUMNAS EN BOOTSTRAP	B
FIGURA 39. CLASE DATABASE -WRAPPER DE MYSQLI.....	C
FIGURA 40. FORMA DE CONECTAR A LA BASE DE DATOS	C
FIGURA 41 CARGADOR DE SESIÓN.....	D
FIGURA 42. CARGADOR DE PARÁMETROS	D
FIGURA 43. MÉTODO ACTION Y RENDER DE LA CLASE LOGIN.....	D
FIGURA 44. DISPOSICIÓN DE CARPETAS	E
FIGURA 45. CLASE TEMPLATECONTEXT	F
FIGURA 46. CÓMO ENVIAR Y RECIBIR VARIABLES A LAS PLANTILLAS	F

FIGURA 47. MUESTRA DE CORREO RECIBIDO. CORREO DE AVISO DE MENSAJE NUEVO EN LA APLICACIÓN.	G
FIGURA 48. EJEMPLO DE CÓMO GENERAR UN PDF CON WKPDF.....	G
FIGURA 49. REPRESENTACIÓN DE LA PANTALLA QUE ADMINISTRA EL MENÚ	H
FIGURA 50. PÁGINA PRINCIPAL DEL ROL ADMINISTRADOR EN GOOGLE CHROME	L
FIGURA 51. PÁGINA PRINCIPAL DEL ROL PROFESOR EN FIREFOX.....	L
FIGURA 52. PÁGINA PRINCIPAL EN FIREFOX MOBILE.....	M
FIGURA 53. PÁGINA PRINCIPAL EN SAFARI (IPHONE).....	M
FIGURA 54. PANTALLA QUE SE MUESTRA CUANDO LAS CALIFICACIONES HAN SIDO INSERTADAS CORRECTAMENTE	M
FIGURA 55. SI LA CALIFICACIÓN ES MAYOR QUE 10, SALE UN ERROR	M
FIGURA 56. PANTALLA DE CAMBIO DE CONTRASEÑA.....	N
FIGURA 57. CAMBIO DE CONTRASEÑA INCORRECTO	N
FIGURA 58. CAMBIO DE CONTRASEÑA CORRECTO.....	N
FIGURA 59. MUESTRA DE DATOS DE USUARIO.....	N
FIGURA 60. HORARIO DE UN GRUPO DE ALUMNOS	O
FIGURA 61. DATOS ERRÓNEO EN EL FORMULARIO QUE GENERA UN EVENTO.....	O
FIGURA 62. LOGIN ERRÓNEO.....	P
FIGURA 63. REENVÍO DE LA CONTRASEÑA OLVIDADA SATISFACTORIO.....	P
FIGURA 64. RESULTADO DE ATAQUE DE VULNERABILIDAD DEL SERVIDOR	T

Índice de Tablas

TABLA 1. TABLA ASIGNATURAS	29
TABLA 2. TABLA BACK939_ENTITY	30
TABLA 3. TABLA BACK939_MENU	30
TABLA 4. TABLA BACK939_ROLES.....	30
TABLA 5. TABLA BACK939_TABLES.....	31
TABLA 6. TABLA BACK939_USER.....	31
TABLA 7. TABLA FALTAS_ALUMNOS.....	31
TABLA 8. TABLA CALIFICACIONES	32
TABLA 9. TABLA EVENTS	32
TABLA 10. TABLA EXAMENES	32
TABLA 11. TABLA BACK939_TABLE_FIELDS.....	33
TABLA 12. TABLA BACK939_USER_ACTIVITY	34
TABLA 13. TABLA CUSTOMERS	34
TABLA 14. TABLA MESSAGES	35
TABLA 15. TABLA TIMETABLE	35
TABLA 16. LISTADO DE PRUEBAS REALIZADAS	K

Glosario

AJAX (*Asynchronous JavaScript And XML*): técnica de desarrollo web para crear funcionalidad interactiva.

AWS (*Amazon Web Services*): colección de servicios web que unidos forman una plataforma de computación en la nube.

B-learning (*Blended Learning*): aprendizaje semipresencial que combina diferentes métodos de impartición de la enseñanza.

Bootstrap: es un *framework* para crear interfaces web con JavaScript y CSS.

E-learning (*Electronic Learning*): es el aprendizaje electrónico o tipo de educación que se realiza a distancia a través de canales electrónicos utilizando herramientas como soporte de los procesos de enseñanza-aprendizaje.

ERP (*Enterprise Resource Planning*): Planificación de Recursos de la Empresa. Es un software que facilita la integración entre las distintas áreas de una empresa.

Framework: estructura conceptual y tecnológica de soporte definido que sirve como base para la organización y desarrollo de software.

LAMP (*Linux Apache Mysql PHP*): acrónimo que describe un sistema de infraestructura de Internet.

LCMS (*Learning Content Management System*): sistema de gestión de contenidos.

LMS (*Learning Management System*): sistema de gestión de aprendizaje. Es un software de servidor empleado para administrar y distribuir las actividades no presenciales de una institución.

MVC (*Model-View-Controller*): patrón de arquitectura software.

MySQL: sistema de gestión de base de datos.

MySQLi (*MySQL improved*): extensión de PHP que permite acceder a la funcionalidad proporcionada por MySQL.

NETBEANS: entorno de programación.

PHP (*PHP Hypertext Preprocessor*): lenguaje de programación.

SIMS (*School Information Management System*): sistema de gestión y de información escolar.

1 Introducción

1.1 Marco del proyecto

Este Trabajo de Fin de Grado tiene como propósito la implementación de una plataforma cuya finalidad es la comunicación directa e interna entre los distintos miembros de la comunidad educativa. De esta manera los miembros se pueden comunicar con mensajes enviados y recibidos desde la propia aplicación. Además, la plataforma ofrece funciones de gestión educativa como la creación de exámenes, inserción de notas, calendario o gestión de datos de todos los miembros que componen el centro educativo. Otra de las finalidades que se quiere conseguir es que la aplicación sea online con una interfaz intuitiva, rápida y atractiva.

A pesar de estar pensado para una aplicación específica, uno de los requisitos prioritarios de este trabajo ha sido que la aplicación tuviera una parte que fuera lo más general posible de manera que, con simples cambios y casi sin necesidad de escribir código, se pudieran modificar muchas partes de la configuración de la aplicación. Esta funcionalidad permitiría por ejemplo (con pocas modificaciones) crear una aplicación para gestionar un back-office de una tienda online, los invitados de una boda o un equipo de fútbol. Todo ello es configurable mediante módulos que sí hay que diseñar.

1.2 Motivación

En los últimos años el número de plataformas de gestión educativa ha aumentado de forma considerable. Constituyen, actualmente, una realidad tecnológica creada en Internet, que facilita la gestión académica y proporciona un útil medio de comunicación entre los distintos miembros de la comunidad educativa.

Por otro lado, el aprendizaje a través de las TIC (Tecnologías de la información y Comunicación) es un paso más de la evolución de la educación a distancia [1]. Esta forma de aprendizaje se denomina *e-learning*. El *e-learning* proporciona al estudiante los escenarios necesarios para crear un ambiente de aprendizaje. Estos escenarios tienen la característica de ser fácilmente accesibles, distribuidos e interactivos. Una **plataforma *e-learning*** es una aplicación que integra un conjunto de herramientas para la enseñanza-aprendizaje en línea, permitiendo una enseñanza no presencial (*e-learning*) y/o una enseñanza mixta (*b-learning*), donde se combina la enseñanza en Internet con experiencias en la clase presencial [2].

Las plataformas de gestión educativa han sido implantadas con ampliación de funcionalidad *e-learning*, sobre todo a nivel universitario y enseñanza secundaria [3]. Existe una disconformidad de los centros educativos de enseñanza primaria [4] con las actuales herramientas de las que disponen. Son muchos los centros que no cuentan con las herramientas de las TIC apropiadas para poder gestionar de una manera más eficiente todos los procesos que ocurren día a día en la escuela.

Las TIC están teniendo un papel muy importante en el sistema educativo. Actualmente, este es uno de los aspectos en los que un centro educativo se puede diferenciar de su competencia.

Las TIC sirven para manejar diversos ámbitos educativos como por ejemplo:

- Comunicaciones entre profesorado y familias sobre temas administrativos o educativos.
- Aprendizaje a partir de los contenidos digitales (*e-learning*).
- Aspectos económicos o de gestión académica del centro educativo.
- Aspectos administrativos y legislativos del centro educativo.

Dado que las plataformas de gestión informatizada en la educación no se han difundido lo suficiente, la comunicación entre el centro escolar y las familias tiene como canal principal al estudiante que los relaciona. Esto provoca que en ocasiones dicha comunicación sea defectuosa o incluso inexistente. Por tanto, es deseable tener un canal de comunicación directo y asíncrono entre familia y escuela. Es bueno que sea asíncrono porque no requiere que una de las dos partes esté activa en el momento de la comunicación. Así las familias podrían atender los comunicados cuando tuvieran tiempo disponible.

1.3 **Objetivos**

El objetivo principal de este proyecto es diseñar una herramienta para los centros educativos que gestione de una manera eficiente los datos de los estudiantes e implante un sistema de comunicación informatizada entre profesores y familias de los estudiantes. Además, con esta herramienta se pretende impulsar la evolución tecnológica en los centros educativos integrando las TIC para mejorar su rendimiento interno y externo. Para lograr este objetivo, la aplicación se apoya en una solución tecnológica diseñada para responder de forma global e integradora a todas las necesidades de un centro escolar mediante un sistema web versátil y potente.

La aplicación debe poder proporcionar las herramientas básicas para poder generar una comunicación entre las familias y el centro educativo mediante mensajería privada interna, generación de eventos, gestión de datos del personal del centro y de los alumnos y listados que permitan visualizar esos datos, como por ejemplo boletines de notas o faltas de asistencia.

Todos estos objetivos se concentran en **SchoolPipe**. **SchoolPipe es la herramienta de “Gestión académica y de comunicación directa entre centros educativos y familias” desarrollada para este Trabajo de Fin de Grado.** SchoolPipe es capaz de gestionar los usuarios y que éstos puedan ser diferenciados unos de otros mediante el concepto de **rol**. Se tendrá en cuenta el usuario conectado para mostrar una configuración u otra. La plataforma no incluye en esta versión un sistema de *e-learning*. El sistema cuenta con una interfaz gráfica basada en el *framework* de Bootstrap [5] que hace que su manejo sea mucho más sencillo y agradable a la vista del usuario.

1.4 Estructura del documento

Este documento se estructura de la siguiente manera:

En el *capítulo 2* se hace un análisis del estado del arte, el cual incluirá la situación actual de las plataformas educativas y algunos ejemplos de éstas.

En el *capítulo 3* se desarrollan los objetivos de este Trabajo de Fin de Grado.

El *capítulo 4* definirá la aplicación que se ha diseñado, así como su metodología y todas las herramientas utilizadas para su realización.

Los capítulos 5, 6 y 7 engloban gran parte del contenido de este documento. En el *capítulo 5* se hablará del análisis realizado de la aplicación incluyendo el catálogo de requisitos y los casos de uso. En el *capítulo 6* se desarrollará el diseño y la implementación de la aplicación. Aquí se describe la arquitectura, el modelo implementado, el desarrollo de la base de datos, el diagrama de clases de la aplicación, la explicación de algunos de los módulos más destacables, etc. El *capítulo 7* detalla la verificación y validación de la aplicación mediante un conjunto de pruebas realizadas sobre el sistema.

El *capítulo 8* hablará sobre las posibilidades que tiene esta herramienta fuera del marco de este proyecto, es decir, como se explicará más adelante, la aplicación tiene una parte que la hace ser reutilizable para otras temáticas.

En el *capítulo 9* se evaluará el resultado final y se verá cuál es su beneficio.

Para finalizar, en el *capítulo 10* se exponen las conclusiones obtenidas tras el desarrollo del proyecto y se mencionarán las posibles líneas de trabajo futuro.

2 Estado del arte

2.1 Plataformas Educativas y sus funciones típicas

Como dice Rodríguez Diéguez, J.I *et al.* en su libro *Tecnología Educativa y Nuevas Tecnologías aplicadas a la educación*: “Una plataforma educativa es una herramienta física, virtual o una combinación de ambas, que brinda la capacidad de interactuar con uno o varios usuarios con fines pedagógicos. Se considera además, que contribuyen en la evolución de los procesos de aprendizaje y enseñanza, complementando o presentando alternativas a las prácticas de educación tradicional” [6].

Como se introduce en el capítulo 1, en los últimos años son muchos los centros educativos que han incorporado a su forma de trabajar un sistema que les ayude con la gestión educativa, financiera, administrativa y de comunicación entre las personas adscritas a dicho centro educativo. Sin embargo, este tipo de sistemas no se usan de forma generalizada en todos los centros, por diferentes motivos, lo cual induce a pensar que quizá una herramienta de este tipo de fácil uso, versátil, barata y eficiente pudiera llegar a usarse de manera más universal en los centros educativos.

Las plataformas suelen tener diferentes elementos y características [7] como los que se explican a continuación:

- **LMS (*Learning Management System*)**: es el punto de contacto entre los usuarios de la plataforma (profesores, estudiantes y empleados). Se encarga, entre otras cosas, de presentar los cursos a los usuarios, del seguimiento de la actividad del alumno, etc.
- **LCMS (*Learning Content Management System*)**: engloba, aspectos directamente relacionados con la gestión y publicación de contenidos.
- **Herramientas de comunicación**: para que los participantes de una actividad formativa puedan comunicarse y trabajar en común, deben proporcionarse los mecanismos necesarios para ello (chat, foros, correo electrónico, intercambio de ficheros, etc.).
- **Herramientas de administración**: son esenciales para la asignación de permisos dentro de cada uno de los cursos o para poder controlar la inscripción y el acceso a las diferentes etapas del curso.

2.2 ERP (*Enterprise Resource Planning*)

Algunos centros educativos utilizan unas herramientas un poco más completas que se denominan ERP. Un ERP (*Enterprise Resource Planning* o Planificación de Recursos de la Empresa) es un software que facilita la integración entre las distintas áreas de una empresa: facturación, financiera, contable, Recursos Humanos, fabricación, suministros, atención al cliente, calidad, etc. [8]. Por la definición de ERP, quizá sea una herramienta demasiado compleja para lo que un centro educativo puede llegar a necesitar. Por ello, han aparecido en el mercado los llamados ERP Educativos o también llamados Sistemas de gestión de información escolar (*SIMS- School Information Management Systems*) [9].

Los centros educativos se han adaptado a este tipo de herramienta, ya que sus necesidades así lo requieren [8]. Las áreas de gestión que son diferentes en un centro educativo y que por tanto son las que mayor atención se debe prestar a la hora de elegir una aplicación de gestión son:

- Informes de documentación oficial interna del centro.
- CRM (*Customer Relationship Management*), o de atención al cliente entendiendo como tal el módulo de comunicación con familias o con alumnos (aspectos administrativos y educativos).
- Gestión de Horarios.
- Facturación.
- Repositorio de documentos.

Existen dos características sin las cuales una plataforma no se puede considerar un verdadero ERP:

1. **Base de datos única**, es decir, la información está unificada en una única base de datos.
2. **Parametrización**: Un ERP no es un software que se saca de una caja o se descarga y se instala en 4-5 pasos. Para usar un ERP es necesario un proceso de configuración o parametrización con los datos personales del centro que va a utilizar el producto.

Además de estas dos características será necesario a estas alturas que la aplicación funcione en la nube, es decir, se ejecute en Internet en un navegador.

A continuación se expone una comparativa de las actuales herramientas de gestión educativa que existen en el mercado a nivel nacional y en algunos casos internacionales. Las hay de todas las clases: plataformas, software, en la nube, en servidores, adaptadas para dispositivos móviles, etc. y la mayoría están compuestas de módulos que pueden ser contratados de manera independiente. Así, los propios centros podrán escoger la instalación de aquellos módulos que vayan a necesitar.

2.3 Herramientas utilizadas actualmente para gestionar centros educativos

Alexia



Alexia es una plataforma que integra el área de gestión académico-administrativa, la comunicación entre centro y familias y además dispone de app para dispositivos móviles [10]. Permite hacer seguimiento del alumno y gestión del personal. Alexia tiene la posibilidad de integrar herramientas como por ejemplo un generador de horarios o un gestor de biblioteca [11].

Aula1



Aula1 es una solución ERP educativa diseñada por Detecsys Soluciones TI sobre Microsoft Azure [12] [13]. Aula1 ofrece los siguientes módulos: económico, académico y comunicación con las familias.

Además, al tratarse de un sistema abierto, se integra con Moodle [14], Sage Murano [15], Google Apps [16] y Office365 Educación [17]. Las funcionalidades que ofrece Aula1 para padres, profesores y alumnos son muy extensas disponiendo además de aplicación para dispositivos móviles.

Globaleduca



Globaleduca [18] está compuesta por seis módulos que pueden integrarse entre sí para la gestión del centro educativo. Varios de ellos son los que se encargan de las tareas administrativas y comunicación familia-escuela. También ofrece una intranet para gestionar toda la organización interna del centro y un portal para la comunicación con las familias.

School Alive



School Alive es una plataforma multilingüe que dispone de varios módulos de gestión, todos ellos personalizables según los deseos de los centros educativos, como por ejemplo gestión del personal, de la instalación, del alumnado, de la comunicación con las familias [4].

Cabe destacar que incluye diferentes herramientas (por ejemplo, blogs) y permite cargar contenidos desde Google Apps.

Educolombia



Educolombia se encarga de ofrecer servicios básicos para el colegio [19]. Algunas de sus funciones son: calendario de eventos de diferentes tipos, directorio de personal, formulario de contacto entre los distintos miembros, etc. Destaca su módulo de notas basado en las normas del Ministerio de Educación Nacional.

Web School



Web School es un ERP educativo creado por desarrolladores indonesios que ofrece la posibilidad de tener un control total de todo el centro educativo [20]. Además, los padres pueden

tener total acceso a la información de sus hijos.

Fedena



Fedena es otra aplicación ERP de origen indio cuyo código es *open source* [21]. Está siendo utilizada por el gobierno del estado hindú de

Kerala y por más de 40.000 instituciones en todo el mundo [22]. Fedena se integra con Moodle, con Google Apps y con una pasarela de pagos.

SIMS

CAPITA



SIMS es un software desarrollado por la empresa inglesa Cápita que se basa en Microsoft SQL Server y cuenta con 30 años de historia [23]. La cuota de mercado es del 80% en los colegios de Inglaterra y Gales [24].

2.4 Plataformas E-learning

Muchas de las plataformas anteriormente mencionadas incluyen entre sus posibilidades un módulo de *e-learning*. Algunas plataformas destinadas al *e-learning* son:



Moodle [14] destaca como plataforma *e-learning* [25] utilizada internacionalmente por muchos centros universitarios. Moodle se encuentra instalado en más de 54.000 sitios y en 225 países. [26]



Blackboard Learn [27] es un software utilizado por miles de organizaciones para lograr que su experiencia educativa sea más confortable.

Blackboard

2.5 Opiniones y conclusiones

La implementación de nuevas plataformas educativas informáticas para la mejora de la educación está puesta en entredicho por personas que opinan que estas plataformas sólo son un apoyo o complemento a la educación tradicional [28]. Además, existen argumentos que indican que estas plataformas pretenden mercantilizar la enseñanza [29]. Otros opinan que la asociación entre casa y escuela es ahora una necesidad para el progreso de la educación [30] [31].

Desde mi punto de vista, estoy de acuerdo con la primera afirmación. Es cierto que estas herramientas son un apoyo para la educación presencial, pero nunca deberían ser un sustituto. Si bien, aunque sean un apoyo, las veo totalmente necesarias en centros de educación infantil y primaria (CEIP). No hay que responsabilizar a un niño de entre cinco y doce años de que se acuerde de decir en casa que hay un examen de matemáticas en tres días o de que hay que abonar el precio de una excursión al zoo.

Por tanto todas las herramientas similares a las mencionadas en este capítulo son relativamente necesarias y los centros educativos deberían poder acceder a ellas a un precio más razonable pues hay algunas extremadamente caras.

3 Objetivos/Funcionalidades

Como se ha especificado en el capítulo 1, el propósito de este Trabajo de Fin de Grado es definir, implementar y presentar un sistema software de gestión académica y comunicación entre centro educativo y familias. Para cumplir este objetivo la aplicación necesita un sistema de roles que clasifique los privilegios para realizar determinadas operaciones en el sistema.

El sistema cuenta con los siguientes componentes:

- **Usuarios:** Son los que interactúan con la plataforma y se debe validar qué pueden o no hacer.
- **Perfiles o roles:** Son los roles que se le asignarán a los distintos usuarios. Los roles que esta aplicación tiene en cuenta son: Administrador, Director, Profesor, Familia y Alumno.
- **Funcionalidades:** Son los distintos objetivos de la aplicación.
- **Permisos:** Son las acciones concretas que se asignan sobre cualquier funcionalidad del sistema.

Para poder implementar el sistema, éste deberá cumplir unos objetivos que enumeramos a continuación.

3.1 Objetivos específicos y funcionalidades

Existen dos tipos de objetivos en la aplicación:

- **Objetivos propios de la aplicación:** La aplicación tiene unos objetivos específicos para la implementación del sistema software de gestión académica y comunicación entre centro educativo y familias.
- **Objetivos genéricos:** La aplicación tiene un sistema de administración con el cual se podrá modificar parte de la configuración de cómo se muestran determinados datos de la aplicación.

A continuación se muestran los dos tipos de objetivos explicados anteriormente.

3.1.1 Objetivos propios de la aplicación

1. **Acceder a los listados de datos de:**
 - a. **Usuarios de la aplicación:** Se puede ver un listado de todos los usuarios y acceder a la edición de los datos de cada usuario.
 - b. **Cursos, grupos y asignaturas:** Se puede ver y editar los cursos, grupos y asignaturas del centro educativo, asignarle un tutor diferente a cada grupo y asignar determinados alumnos a grupos.
2. **Listado de alumnos por grupo:** Esta funcionalidad permite a los profesores ver los alumnos que tienen asignados y acceder a sus datos.
3. **Pasar Lista:** Los profesores pueden asignar faltas de asistencia en días concretos a sus alumnos. Esta información es útil para la familia del alumno.
4. **Generar eventos:** Cualquier usuario puede generar eventos en su calendario personal rellenando un formulario, con los datos de dicho evento.

5. **Envío de mensajes:** El sistema es capaz de gestionar una bandeja de entrada, de salida y papelera interna. Esto permite enviar mensajes privados a usuarios de la aplicación (solicitud de reunión, comunicados, etc.). Este sistema envía un correo electrónico al usuario que recibe el mensaje privado notificando dicha acción.
6. **Generación de exámenes y asignación de notas:** Los profesores pueden generar exámenes de las asignaturas que imparten y poner la nota de sus alumnos en dichos exámenes.
7. **Ver Horario de clases:** Los usuarios de la aplicación pueden ver su horario de clases personal (alumno-familia-profesor) o ver el horario de un determinado grupo (director-profesor tutor).
8. **Ver y descargar el boletín de notas:** La familia puede ver y descargar el boletín de notas de los estudiantes relacionados con ellos.
9. **Ver perfil:** Cualquier usuario puede ver su perfil con sus datos y un acceso a un calendario de eventos.
10. **Cambio de contraseña:** Los usuarios pueden modificar su contraseña en caso de necesidad u olvido. El segundo caso envía al correo electrónico del usuario que la solicita la nueva contraseña de acceso.

3.1.2 Objetivos genéricos

Como se ha dicho anteriormente, uno de los objetivos de la aplicación es realizar una base lo más general posible para que el sistema sea replicable bajo otra temática. Esta parte consiste en poder configurar sin codificar las opciones del menú de navegación y asignar permisos a esas opciones. Además, y como parte más compleja, el sistema es capaz de administrar las tablas de la base de datos en modo vistas. Esto quiere decir que una misma tabla de la base de datos puede tener distintas configuraciones dependiendo del usuario que quiera acceder a los datos almacenados en la tabla.

Existen dos objetivos a diferenciar en este punto:

1. **Administración de tablas de la base de datos:** Esta es una funcionalidad muy completa, la cual proporciona al administrador acceso sobre gran parte de la configuración de las tablas de la base de datos sin necesidad de clientes externos o modificar líneas de código. Cualquier tabla de la base de datos se puede insertar en lo que se llamará “*Administración de Tablas*”. Cuando una tabla se inserta en dicha administración se puede configurar una serie de opciones, que se verán más adelante, configurables y que servirán para mostrar listados de datos ordenables, insertar y editar dichos datos o asociar a la tabla administrada ficheros adjuntos. Por último se pueden hacer exportaciones e importaciones de las configuraciones de las tablas entre distintos sistemas.
2. **Administración del menú de navegación:** La aplicación tiene la posibilidad de administrar el menú que puede ver cada tipo de usuario. Esta opción lo que permite al administrador es generar una opción en el menú y asignar permisos de visibilidad.

4 Definición del Proyecto

4.1 Metodología

En este apartado se definirán con detalle y en orden de ejecución cada una de las etapas que se han seguido durante el desarrollo del proyecto.

Las fases del proyecto coinciden con las que normalmente se utilizan en la mayoría de proyectos software [32]:

1. **Análisis:** La tarea principal de esta fase es la captura de requisitos. Estos deben delimitar exactamente cómo ha de funcionar la aplicación.
2. **Diseño:** Esta tarea consiste en realizar el diseño completo de la aplicación. Esto implica el diseño de las bases de datos que almacenarán la información, diseño de navegación y diseño de las clases que se utilizarán (dado que gran parte de la aplicación se va a desarrollar en PHP).
3. **Implementación:** Durante esta fase se han codificado los diseños definidos durante la fase anterior. Además, se realizarán parte de las pruebas que más adelante habrá que documentar. Como se explica a continuación en el modelo de ciclo de vida escogido, no supondrá un gran inconveniente realizar modificaciones en esta fase, ni en las anteriores.
4. **Pruebas:** Aunque durante la fase de implementación se realizarán muchas de las pruebas que se van a llevar a cabo, en ésta se tratará de completar y repasar ese proceso para asegurar que todas las funciones, módulos y partes de la aplicación realizan exactamente su función como se desea, probando todos los casos y documentando todos los resultados obtenidos.
5. **Documentación:** Aparte de toda la documentación de requisitos, de diseño y de código que se ha ido realizando durante las fases anteriores, se deberá llevar a cabo una documentación general y más completa, en la que se recogerán todos los aspectos del proyecto y se relacionarán todos los puntos importantes de las fases, así como los resultados de las mismas y las conclusiones obtenidas.
6. **Mantenimiento:** Normalmente, esta es una de las fases más largas de los proyectos software. Se realizan todo tipo de mejoras, actualizaciones y modificación en la aplicación. En el caso de esta aplicación, esta fase del proyecto se trabajará en el futuro. Por tanto, es importante dejar el proyecto bien documentado.

En el diagrama de Gantt de la figura 1 pueden observarse estas etapas y la distribución temporal de las tareas más generales llevadas a cabo, así como la dependencia entre ellas.

Como se aprecia, se sigue un modelo de ciclo de vida en cascada. Dado que sólo había un recurso humano en la fase de codificación, se decidió intercalar varias de las tareas que se planificaron para esta fase.

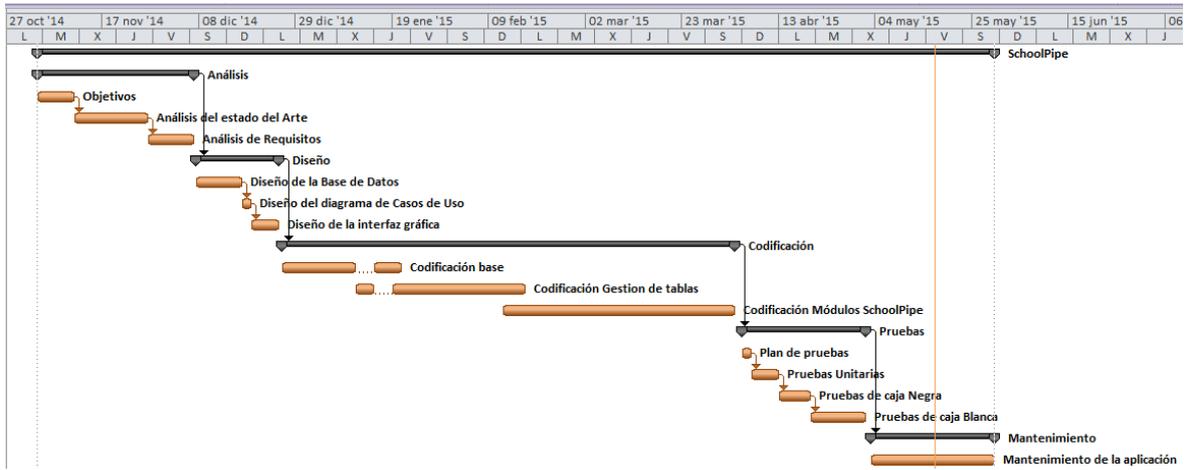


Figura 1. Diagrama de Gantt

A continuación se procederá a justificar el ciclo de vida [33] utilizado en este proyecto.

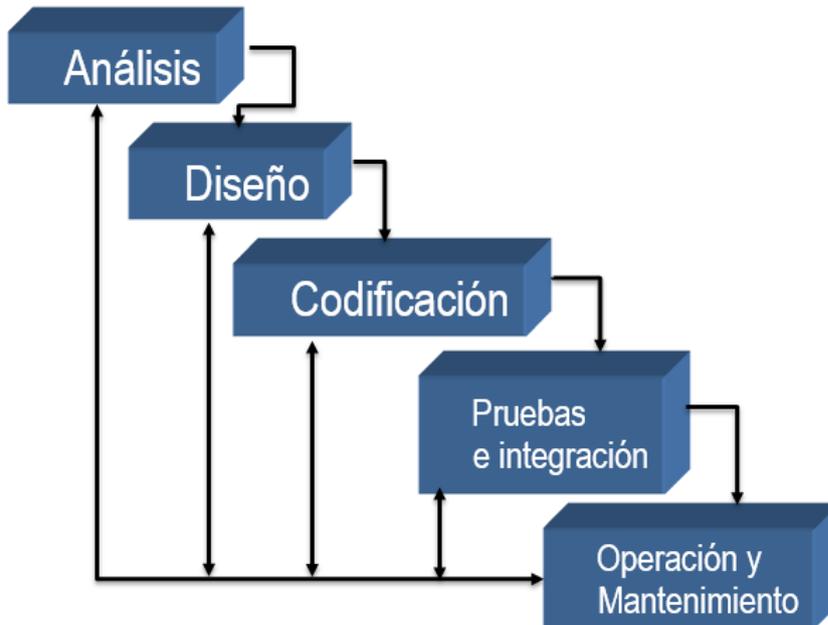


Figura 2. Ciclo de vida en cascada con realimentación

Se ha escogido una variante bastante común del modelo de ciclo de vida en cascada, añadiéndole una mejora iterativa que permite ir generando los productos de las diferentes etapas e irlos refinando y mejorando. Con ello se evitan algunas de las limitaciones que se producen al utilizar el modelo de cascada clásico. La ventaja de este modelo es que permite volver atrás en cualquier momento para refinar o modificar algún requisito que previamente podría haber sido olvidado. Una vez fijado los requisitos, se repetirán las fases de diseño, desarrollo y pruebas hasta cumplir los resultados esperados.

4.2 Herramientas usadas

En este apartado se enumeran las diferentes herramientas y tecnologías utilizadas durante el desarrollo del proyecto.

Netbeans

Netbeans es un entorno de programación integrado (IDE) libre que sirve para desarrollar aplicaciones [34]. Está compuesto por un conjunto de módulos. Un módulo es un archivo escrito en JAVA que contiene clases para interactuar con el API de Netbeans. Tiene una interfaz intuitiva y agradable con un gran número de opciones de configuración. Netbeans tiene un gran soporte para aplicaciones web escritas en PHP5, un buen depurador integrado y proporciona soporte para AJAX.

LAMP

Es el acrónimo utilizado para describir un sistema de infraestructura de internet. Este paquete está compuesto de herramientas de software libre. Esto implica que se pueden distribuir gratuitamente y son totalmente modificables por el usuario. El paquete está formado por las siguientes herramientas:

Linux

Es el sistema operativo utilizado. Está basado en los estándares Unix. Linux es el núcleo del sistema operativo y el sistema completo es GNU con la adición de dicho núcleo. Por ello a este sistema también se le conoce como GNU/Linux. Está perfectamente adaptado al ámbito de los servidores web. Actualmente Linux es el sistema más usado en servidores web [35]. Amazon y Google utilizan este sistema.

Apache

Apache es el servidor web por excelencia [36]. Es el componente principal del paquete y responde a las solicitudes web (HTTP/HTTPS) de los navegadores. Se le pueden incluir módulos de software para aumentar la funcionalidad. Actualmente el 51% de los sitios activos trabaja sobre Apache [37]. Es un servidor fiable y, como se ha dicho anteriormente, gratuito.

MySQL

MySQL [38] se utiliza como gestor de Base de Datos relacionales. Permite crear, acceder y administrar bases de datos. Es una gran herramienta para los sitios web dinámicos porque dependen del uso de una base de datos. Este gestor se creó pensando en la rapidez y robustez y no posee muchas de las características que otros gestores comerciales como Oracle o SQL Server sí que disponen. Aun así, dado que las aplicaciones web no requieren muchas características avanzadas, hacen que MySQL sea un servidor de base de datos muy apropiado, y cuenta con más de 12 millones de instalaciones activas, según MySQL AB [39].

PHP, Perl o Python

Cualquiera de estos tres lenguajes de programación podría ser la “P” del acrónimo de LAMP.

- **Perl:** Es un lenguaje de programación que se usa para escribir scripts CGI. Estos scripts se utilizan para relacionar las respuestas de programas ejecutados en el servidor con páginas web dinámicas.
- **Python:** Es un lenguaje muy sencillo de aprender. Se usa para generar aplicaciones web que se ejecutan directamente en la memoria del navegador.
- **PHP:** es un lenguaje de programación que ejecuta instrucciones en el servidor a medida que el navegador va leyendo la página web. Su sintaxis es parecida a Java.

Cabe destacar que en este proyecto el lenguaje de programación principal utilizado ha sido PHP debido a la mayor experiencia con este lenguaje. Dado que es software libre permite instalar rápidamente, por ejemplo, un comercio electrónico con osCommerce, un weblog con PHP, Nuke o un portal web para administrar las bases de datos MySQL con phpMyAdmin, del cual se hablará más adelante.



Figura 3. Componentes de un LAMP

(Fuente: Programming4Us)

A continuación se numeran algunas ventajas e inconvenientes de utilizar LAMP [40]:

Ventajas:

1. LAMP da soporte a arquitecturas, como por ejemplo Intel, Mips o PPC (Macintosh).
2. Si existen problemas de seguridad estos, habitualmente, son solventados con rapidez.
3. Actualización de los programas vía internet.

Inconvenientes:

1. El funcionamiento de LAMP es muy distinto de un Windows. Si el usuario que lo está ejecutando o probando está muy acostumbrado a Windows dificultará su trabajo. Es importante decir que para distribuciones de Windows existe WAMP y para distribuciones Macintosh existe MAMP.
2. Para actualizar los programas es necesario tener conocimientos sobre órdenes de terminal en Linux, aunque desde las últimas versiones, con su nueva interfaz gráfica de actualizaciones, esto ya no es tan necesario.
3. Configurar algunos servicios de red requiere de más tiempo que en Windows.

phpMyAdmin

phpMyAdmin es una herramienta escrita en PHP que sirve para administrar bases de datos MySQL o MariaDB a través de páginas web [41]. Tiene todas las acciones básicas que se pueden realizar sobre una base de datos: crear, eliminar, alterar tablas, añadir, eliminar y editar nuevos campos de una tabla, ejecutar sentencias SQL, obtener diagramas de la base de datos, importar y exportar datos en CSV o SQL, etc.

Asembla y Subversion

Asembla es un portal web que proporciona herramientas para gestionar proyectos [42]. Se puede acceder a un panel de tareas de proyecto, *Wiki's* propias o lo más importante, un repositorio del código del proyecto.

Subversion es un sistema de control de versiones con el cual se puede no solo trabajar en grupo, si no tener accesible las diferentes versiones que vayamos generando del proyecto [43].

Amazon Web Services (AWS)



Figura 4. Algunos Servicios de AWS
(Fuente: Amazon Web Services)

Para la instalación de la plataforma en un entorno de producción se ha utilizado Amazon Web Services [44], de ahora en adelante AWS. AWS es una colección de servicios web que unidos forman una plataforma de computación en la nube. Algunos de sus servicios disponibles son:

- Servidores web
- Servidores de Bases de datos
- Archivado de datos
- Correo electrónico
- Gestor de DNS

Además, cuenta con gran cantidad de software de apoyo instalable en sus servidores.

Timetable Web

Timetable Web se ha utilizado como recurso de apoyo para la generación de los horarios de los profesores y de los grupos del centro educativo [45].

Lenguajes y técnicas de diseño Web

Para el desarrollo de la interfaz de la web se ha utilizado un conjunto de lenguajes bastante típico en este tipo de aplicaciones como son:

HTML

Este estándar sirve de referencia para elaborar páginas web. Al tratarse de una aplicación web, se utiliza este lenguaje para definir un código y una estructura básica del contenido de la misma.

Javascript, AJAX y JQuery

Javascript es un lenguaje de programación interpretado. Se ejecuta directamente en el lado del cliente o navegador web. Esto implica que es el navegador el que soporta la carga de procesamiento, y de este modo, se libera al servidor de gran parte de la carga de trabajo.

AJAX (*Asynchronous JavaScript And XML*) es una técnica que permite, mediante programas escritos en Javascript, que un servidor y un cliente (navegador) intercambien datos en formato texto, HTML, JSON o XML, de forma asíncrona [46]. Todo este proceso sucede sin necesidad de realizar demasiadas llamadas al servidor (recargas de página, por ejemplo). En la figura 5 se ve la diferencia entre utilizar un modelo clásico de aplicación web y utilizar un modelo que implementa AJAX.

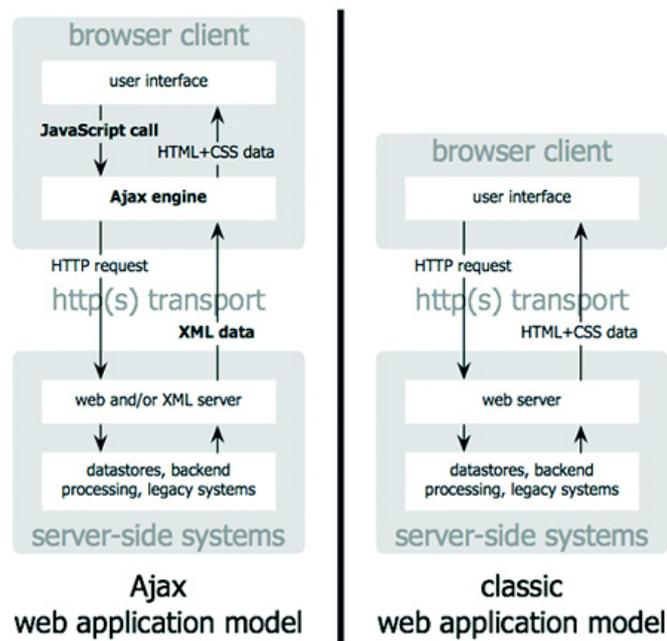


Figura 5. Diferencia entre una aplicación web con Ajax y una plataforma Web clásica

(Fuente: QArea)

Jquery es una librería de JavaScript que ayuda a hacer más simple la interacción con documentos HTML, manejar eventos, gestionar animaciones y agregar dinamismo utilizando AJAX [47]. Resulta muy útil en partes de la aplicación en las que hay que manejar animaciones y en las partes de la aplicación en las que hay que utilizar AJAX. Además de ser compatible con todos los navegadores.

CSS, el Framework Twitter Bootstrap y Font-Awesome

CSS es un lenguaje utilizado para definir la presentación de un documento escrito en HTML o XML [48]. El CSS es utilizado para separar la estructura de un documento de la presentación del mismo.

Twitter Bootstrap es un *framework* para crear interfaces web con JavaScript y CSS [49]. Contiene plantillas de diseño con tipo de letra propio, formularios, botones, cuadros, menús de navegación y una interminable lista de elementos de diseño. Su labor es facilitar y agilizar las tareas de dar formato y buen aspecto a una aplicación web. Además, Bootstrap integra *jQuery*, lo que le da a la herramienta muchos elementos adicionales como *Modals* (modales), *Tabs* (pestañas), *Tooltips* (descripciones emergentes), *Popovers* (ventanas emergentes), etc.

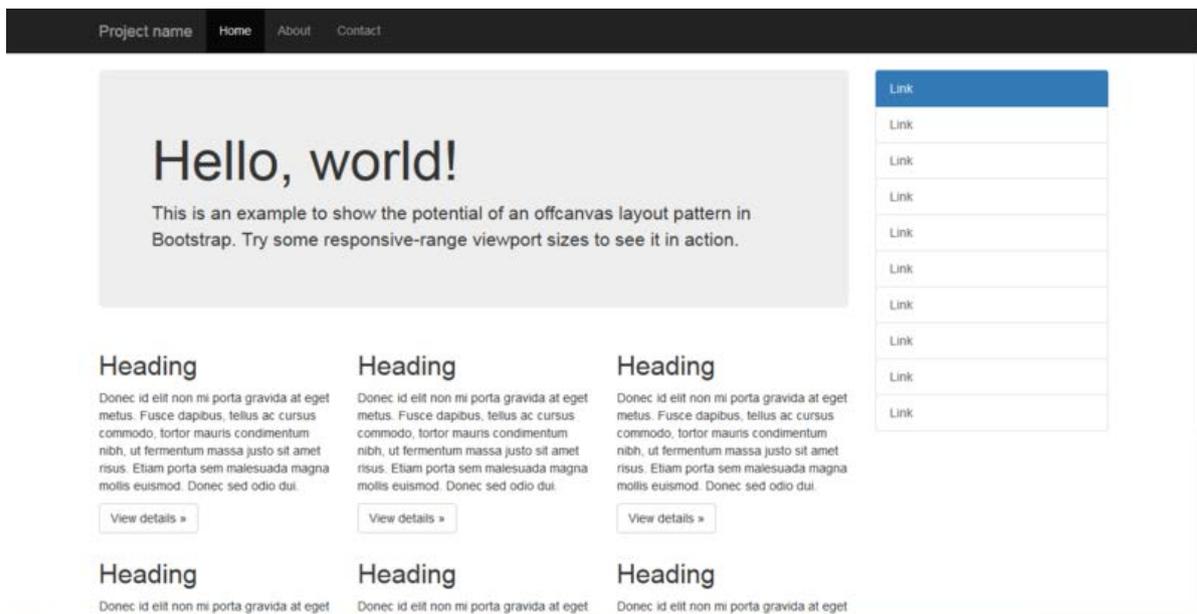


Figura 6. Ejemplo de una página web usando el Framework de Bootstrap en Mozilla Firefox
(Fuente: Bootstrap)

Font-Awesome es una fuente de iconos mediante la cual se pueden insertar algunos símbolos en lugar de imágenes [50]. Al ser una fuente, hace que la página sea más ligera. Su uso está preparado para funcionar y fusionarse fácilmente con el *framework* de Bootstrap pero también se puede usar sin incluir este *framework*.

CKEditor

CKEditor es un editor HTML *open source*, que permite introducir texto con todas las propiedades de otros editores como Microsoft Word sin necesidad de instalarlo en el dispositivo del usuario que lo utiliza [51]. Se ha utilizado este editor para la redacción de los mensajes privados entre los usuarios de la aplicación.

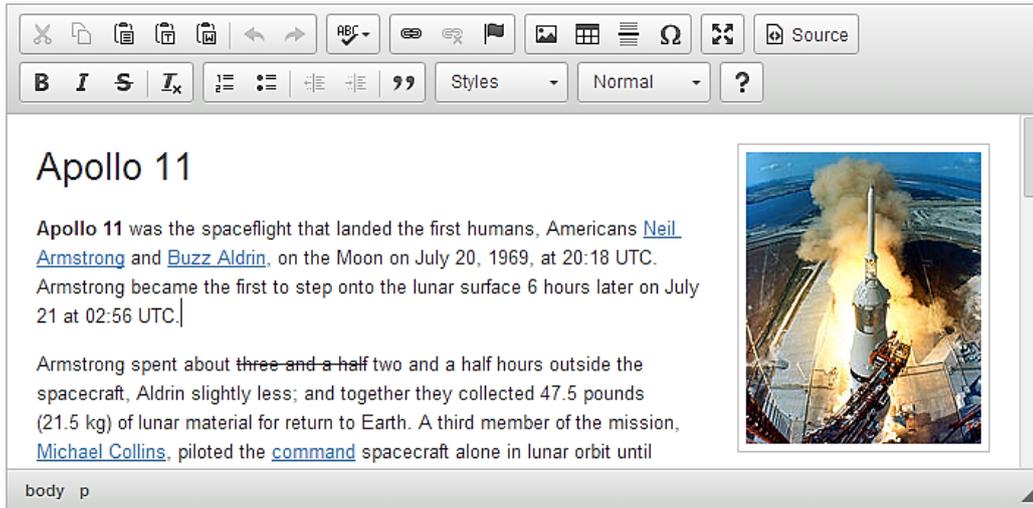


Figura 7. Ejemplo del editor CKEditor
(Fuente: CKeditor)

5.1 Introducción

En este capítulo se presenta la fase de análisis de este proyecto software especificando los casos de uso de la aplicación. También se definirá el catálogo de requisitos, el cual se tendrá en cuenta como base para el diseño de la aplicación. Es por ello por lo que la fase de análisis es de suma importancia para el devenir de todo producto software. Es en este capítulo donde se deben asentar las bases del proyecto y a partir de las cuales se construirá todo lo demás.

El proceso a seguir se basa en primer lugar en una buena definición de las diferentes acciones que pueden realizar los distintos roles. Esto se puede ver mediante unos casos de uso de la aplicación. Seguidamente, y basándonos en estos casos de uso, se especificarán los requisitos específicos de la aplicación.

5.2 Casos de Uso

Los casos de uso de la aplicación se presentarán en tres diagramas siendo cada uno los siguientes:

- Diagrama de caso de uso de usuario sin iniciar sesión.
- Diagrama de usuario que ha iniciado sesión.
 - Acciones comunes de estos usuarios.
 - Acciones no comunes de estos usuarios.

Para entender los diagramas es necesario comprender los objetos que en él aparecen:

- Muñeco: Actor que interviene en los casos de uso.
- Globo: Acciones.
- Flechas: relación entre actor y acción.
 - Continua: el actor debe realizar alguna acción para continuar a la siguiente acción.
 - Discontinua: Por haber realizado una acción, el sistema irá automáticamente a la siguiente acción.

En la figura 8 se muestra el primer diagrama. Existe un usuario con dos acciones claras: iniciar sesión (de ahora en adelante **login**) o recordar la contraseña. Si el login es satisfactorio se devuelve la sesión iniciada y accede a la página principal de la aplicación. En caso contrario se devolverá error. Si el usuario no recuerda su contraseña podrá recibir un e-mail con una nueva si introduce su e-mail en el apartado “Recordar contraseña”.

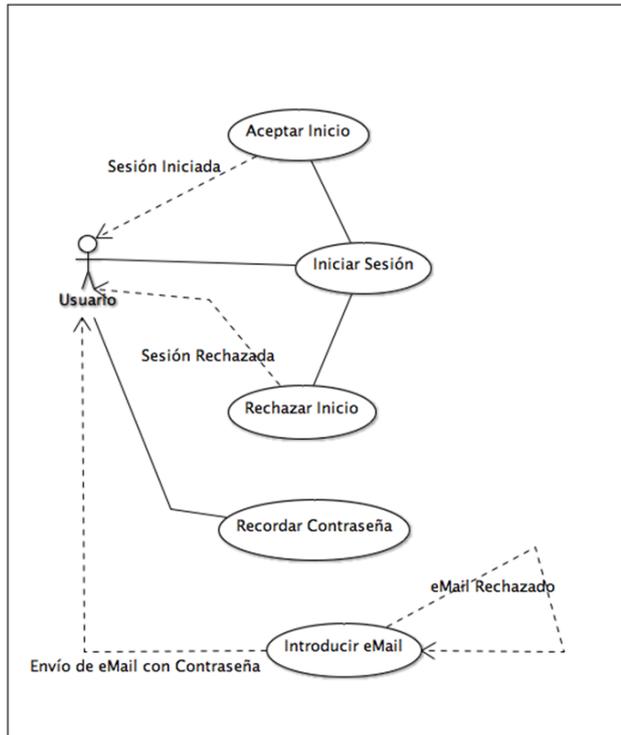


Figura 8. Diagrama de caso de uso 1. Usuario sin iniciar sesión en el sistema

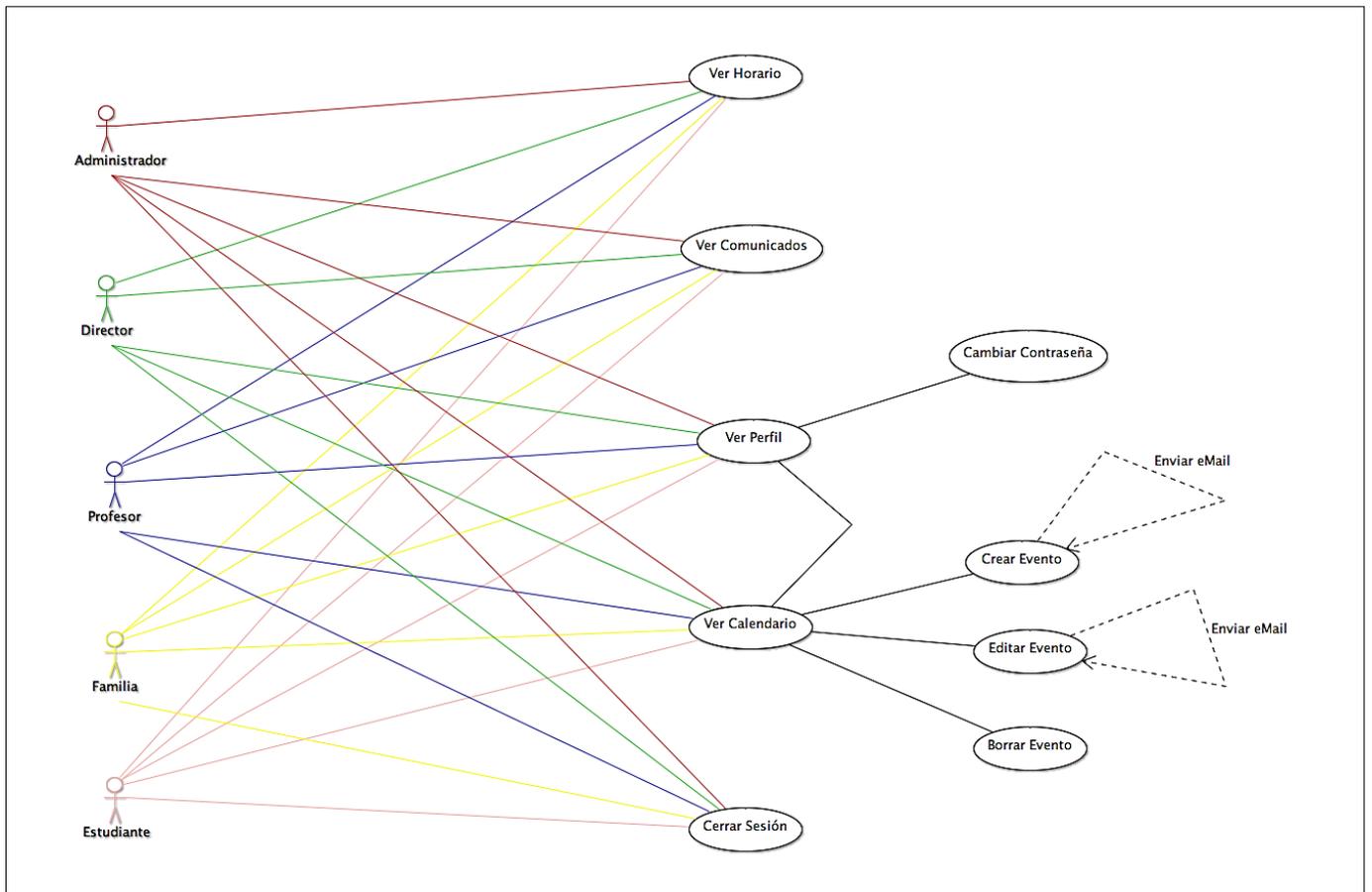


Figura 9. Diagrama de caso de uso 2. Acciones comunes de todos los usuarios

En la figura 9 se muestran todas las acciones que pueden hacer todos los usuarios de la aplicación. Todos los usuarios pueden: ver su horario, ver sus comunicados y generar nuevos, ver su perfil y modificar su contraseña, ver su calendario y generar nuevos eventos o editar los ya existentes y cerrar sesión.

En la figura 10 se muestran todas las acciones que pueden hacerse dependiendo del tipo de usuario que ha hecho login en la aplicación. Los **estudiantes** y **familia** tienen acceso a toda la gestión de los propios estudiantes como son las notas, las faltas de asistencia o ver información sobre los hijos. Los **profesores**, además, pueden acceder a la gestión de exámenes y pueden asignar faltas de asistencia a los alumnos. El **director** puede acceder a toda la gestión del centro educativo como son los grupos, cursos, asignaturas o personal. Por último, el **administrador**, además de todo lo anteriormente mencionado, podrá acceder a una sección especial de configuración del sistema, que se verá más adelante.



Figura 10. Diagrama de caso de uso 3. Acciones no comunes de los usuarios

A partir de todos los casos de uso recientemente explicados se ha de generar un catálogo de requisitos específico para este proyecto software.

5.3 Catálogo de requisitos

Para este proyecto se ha generado un catálogo de requisitos donde se distinguen dos tipos principales de requisitos:

- **Requisitos funcionales:** especifican cada una de las funcionalidades del sistema.
- **Requisitos no funcionales:** Son todos los requisitos que no describen funcionalidades de la aplicación. En este paquete entran, por ejemplo, los siguientes tipos de requisitos: Rendimiento, diseño, seguridad, usabilidad, estabilidad, accesibilidad,...., etc.

A continuación se define el catálogo de requisitos para describir la funcionalidad y las características.

5.3.1 Requisitos Funcionales

Se dividirán los requisitos funcionales en requisitos de módulo para facilitar su comprensión y que el catálogo de requisitos esté bien estructurado.

Requisitos funcionales generales

- **[RF-SP-GE-001] Conexión a internet:**
 - Es necesario que el usuario que quiera utilizar la aplicación mantenga una conexión a internet para poder acceder a los servidores. Su ejecución es 100% en la nube.
- **[RF-SP-GE-002] Menú de opciones fácilmente accesible:**
 - Las principales funcionalidades de cada rol de la aplicación estarán disponibles en la página *Home*.
- **[RF-SP-GE-003] Compatibilidad:**
 - La aplicación web será compatible con los principales clientes web de escritorio del actual mercado además de con los navegadores de plataformas móviles en plataformas Android e IOS.



Figura 11. Diferentes Navegadores web

(Fuente: Tapernet)

Requisitos de la gestión de los usuarios

- **[RF-SP-GU-001] Crear usuario (parte 1):**
 - El administrador tiene la opción, desde la gestión de usuarios de la aplicación, de crear un nuevo usuario.
- **[RF-SP-GU-002] Crear usuario (parte 2):**
 - El director tiene la opción, una vez que el administrador ha creado el usuario, de crear todos los datos personales asociados a un usuario.

- **[RF-SP-GU-003] Editar datos del usuario:**
 - El director puede editar cualquiera de los datos personales de un usuario.
- **[RF-SP-GU-004] Asignar alumnos a un determinado grupo:**
 - El director puede editar cualquiera de los datos personales de un usuario.
- **[RF-SP-GU-005] Consultar datos de un usuario:**
 - Todos los usuarios tienen acceso a su perfil donde verán sus datos personales.
- **[RF-SP-GU-006] Cambiar su contraseña de acceso:**
 - En referencia a **[RF-SP-GU-005]** cada usuario, desde su panel personal, puede modificar su contraseña de acceso.
- **[RF-SP-GU-008] Ver horario:**
 - Cualquier persona del centro educativo puede ver el horario de clases siguiendo la siguiente lógica:
 - Administrador y Director: Pueden ver los horarios asignados a cada grupo de la institución.
 - Profesor: Puede ver su horario personal de clases. En caso de ser tutor de un grupo, puede ver también el horario de su grupo.
 - Alumno: Puede ver su horario.
 - Familia: Puede ver el horario de los alumnos relacionados.
- **[RF-SP-GU-009] Autenticar usuarios:**
 - Para funcionar con la página será necesario iniciar sesión. Cuando el usuario inicia sesión la aplicación reconoce su rol para mostrar la información a la que puede acceder.
- **[RF-SP-GU-010] Cerrar sesión del usuario:**
 - Cualquier usuario puede cerrar su sesión en el sistema.
- **[RF-SP-GU-011] Pasar lista a Alumnos:**
 - Sólo los profesores podrán utilizar la funcionalidad de asignar faltas de asistencia a los alumnos de sus clases en determinados días.

Requisitos de gestión de la sesión

- **[RF-SP-SES-001] Información del usuario en la sesión:**
 - Todos los datos básicos del usuario que ha iniciado sesión se guardan en la sesión para que éstos sean accesibles desde cualquier parte de la página sin necesidad de tener que enviarlas por parámetros.
- **[RF-SP-SES-002] Información del menú de usuario en la sesión:**
 - Como se dijo anteriormente, cada usuario tiene asociado unas opciones de menú. Todas estas opciones, con el fin de liberar a la base de datos de buscar en cada interacción con la página cuál es el menú del usuario que ha iniciado sesión, se cargan en la sesión la primera vez que el usuario accede.

Requisitos de gestión de comunicados

- **[RF-SP-COM-001] Comunicación con otros usuarios:**
 - Los usuarios pueden comunicarse con otros usuarios de la aplicación mediante mensajes privados.
- **[RF-SP-COM-002] Redactar nuevos mensajes:**
 - Cualquier usuario puede redactar nuevos mensajes privados. En caso de que el usuario conectado sea un profesor tutor de un grupo, podrá enviar el mensaje a todos los alumnos relacionados con ese grupo.
- **[RF-SP-COM-003] Visualizar mensajes:**
 - Cualquier usuario puede visualizar el contenido de los mensajes privados que reciba.
- **[RF-SP-COM-004] Responder mensajes:**
 - Cualquier usuario que reciba mensajes puede responderlos.
- **[RF-SP-COM-005] Marcar mensajes como “No Leído”:**
 - Cualquier usuario puede marcar un mensaje como “No Leído” si lo desea.
- **[RF-SP-COM-006] Borrar mensajes:**
 - Cualquier usuario puede borrar un mensaje para enviarlo a la papelera y una vez allí puede borrarlo definitivamente de la aplicación. Estos mensajes sólo son recuperados por el administrador.
- **[RF-SP-COM-007] Envío de notificaciones:**
 - Cuando se genera un mensaje privado se envía un e-mail al correo de los receptores de dicho mensaje.

Requisitos de gestión de exámenes

- **[RF-SP-EX-001] Crear nuevo examen:**
 - Los profesores pueden crear un nuevo examen rellenando una serie de datos que relacionan dicho examen con un grupo, asignatura, tipo, etc.
- **[RF-SP-EX-002] Editar examen:**
 - Los profesores pueden editar datos referentes a un examen ya creado.
- **[RF-SP-EX-003] Calificar examen:**
 - Los profesores podrán seleccionar un examen creado por ellos y asignar notas a los alumnos del grupo de dicho examen.
- **[RF-SP-EX-004] Descargar boletín de notas:**
 - Las familias podrán descargar un archivo PDF con un resumen de todas las notas del alumno asociado.

Requisitos de gestión de cursos, grupos y asignaturas

- **[RF-SP-CGA-001] Inserción de nuevos datos:**
 - La aplicación permite introducir un nuevo curso en el sistema, un nuevo grupo asociado un curso o una asignatura.
- **[RF-SP-CGA-002] Edición de datos:**
 - La aplicación permite editar los datos de un curso, un grupo asociado un curso o una asignatura ya existentes.

Requisitos de gestión del calendario y de los eventos

- **[RF-SP-CAL-001] Ver calendario:**
 - Cualquier usuario puede acceder a su calendario para ver los eventos que en él se tienen programados. Además, la vista de calendario puede hacerse por día, semana, mes y año. En la pantalla de visión general aparece un listado de los eventos futuros. En el calendario aparecen los eventos creados por el usuario o aquellos en los que se ve involucrado.
- **[RF-SP-CAL-002] Crear eventos:**
 - Cualquier usuario puede crear nuevos eventos y, según el tipo de evento, invitar a participantes.
- **[RF-SP-CAL-003] Editar eventos:**
 - Cualquier usuario puede editar los eventos que él mismo ha creado.
- **[RF-SP-CAL-004] Borrar eventos:**
 - Cualquier usuario puede borrar sólo los eventos creados por sí mismo.
- **[RF-SP-CAL-005] Envío de notificaciones:**
 - Cuando se genera un evento, si éste tiene invitados, se envía un e-mail al correo de los invitados.

Requisitos del Administrador

- **[RF-SP-ADM-001] Configuración de la aplicación a nivel Administrador:**
 - Existe un rol (administrador) que tiene acceso a toda la configuración de la aplicación.
- **[RF-SP-ADM-002] Gestión de menú:**
 - El administrador puede configurar las opciones del menú que podrá ver cada rol de la aplicación.
- **[RF-SP-ADM-003] Crear nueva opción de menú:**
 - Se pueden crear nuevas opciones que aparecerán en el menú.
- **[RF-SP-ADM-004] Editar opción de menú:**
 - Se pueden editar las opciones ya creadas del menú.
- **[RF-SP-ADM-005] Eliminar opción de menú:**
 - Se puede eliminar cualquier opción del menú.
- **[RF-SP-ADM-006] Gestión de entidades:**
 - Como se verá más adelante, la aplicación está formada por módulos o entidades. Existe un panel de administración de las entidades de la aplicación.
- **[RF-SP-ADM-007] Administración de la configuración de las tablas de la BBDD:**
 - El administrador tiene acceso a un panel donde podrá añadir nuevas tablas a la administración de tablas, editar la configuración de una tabla o importar y exportar la configuración de dichas tablas. Más adelante se verá este requisito con más descripción.

5.3.2 Requisitos No funcionales

- **[RNFSP-001] Interfaz web sencilla:**
 - Menús claros y concisos para asegurar que el usuario navega con comodidad.
- **[RNFSP-002] Base de datos estable:**
 - Por debajo de la aplicación funciona una base de datos capaz de albergar gran cantidad de datos.
- **[RNFSP-003] Copias de seguridad:**
 - Periódicamente se realizan copias de seguridad de la base de datos
- **[RNFSP-004] Máquina con potencia para multi-conexión:**
 - La aplicación debe ser servida desde una máquina con los suficientes recursos como para poder admitir que muchos usuarios estén conectados simultáneamente.
- **[RNFSP-005] Aplicación con base genérica:**
 - La aplicación se ha diseñado de forma que sea fácilmente replicable bajo otra temática teniendo una parte común que no sea necesaria reprogramar.
- **[RNFSP-006] Diseño modular:**
 - En referencia al **[RNFSP-005]**, la aplicación se ha realizado bajo un diseño modular para facilitar la incorporación de nueva funcionalidad futura. Estos módulos se llamarán **entidades** de ahora en adelante.
- **[RNFSP-007] Seguridad en el *login*:**
 - Para aumentar la seguridad de los datos de los usuarios, la aplicación solicita una contraseña de acceso. Dicha contraseña es generada por la aplicación cuando el administrador crea el usuario. La contraseña está generada bajo el algoritmo MD5 dado que este tipo de encriptación es unidireccional, es decir, es irreversible.
- **[RNFSP-008] Introducción de datos sencilla:**
 - La introducción de determinados datos se ha realizado de tal forma que el usuario no cometa demasiados errores promoviendo el uso de elegir datos mediante formularios seleccionables.
- **[RNFSP-009] Volumen de datos:**
 - No existe límite en la creación de usuarios (profesores, alumnos, representantes de familia...), asignaturas o cursos.
- **[RNFSP-010]: Operatividad**
 - La información proporcionada por la página debe estar operativa para que quienes accedan a ella puedan acceder a dicha información sin ninguna anomalía.
- **[RNFSP-011]: Privacidad**
 - La información de los usuarios sólo será accesible si previamente se inicia sesión.

6 Diseño e Implementación

6.1 Introducción

En esta sección se tratará todo lo referente al diseño de este proyecto web incluyendo la implementación de todo el sistema. Se especificará el diseño de la arquitectura de la aplicación, el diseño e implementación de todos los módulos relevantes, el diseño de la base de datos, el diagrama de clases utilizadas y se detallará el diseño de la codificación empleada.

6.2 Arquitectura de la aplicación

Para cumplir con los requisitos especificados en el capítulo anterior y las funcionalidades definidas se debe elegir una arquitectura de desarrollo acorde con el proyecto. La aplicación está basada en una arquitectura Cliente-Servidor, es decir la aplicación gestiona el servidor web y utiliza como interfaz páginas web. Esta arquitectura se denomina también modelo de dos capas.

SchoolPipe está formado por:

- Una base de datos relacional en la que almacenar los datos de los usuarios, del centro educativo y de configuración del sistema.
- Un cliente (navegador) que envía peticiones.
- Un servidor que recibe y responde a las peticiones de los clientes (navegadores).

Las aplicaciones web son utilizadas por usuarios que se conectan desde cualquier punto a través de sus clientes web y además es proporcionada gracias a un servidor web. Por lo tanto, la arquitectura de tipo Cliente-Servidor consta de dos componentes básicos: un servidor web y uno o más clientes. Además, ambos componentes necesitan de una conexión de red para funcionar.

Se ha contratado una máquina en la plataforma de AWS con el fin de alojar en él **SchoolPipe**. Dicha máquina, tras ser configurada, cuenta con los elementos que un LAMP exige: la versión 14.04 de Ubuntu Server, Apache como servidor web y PHP5 para interpretar el código. Además, en AWS existe la posibilidad de elegir una máquina aparte donde va a estar el servidor de base de datos.

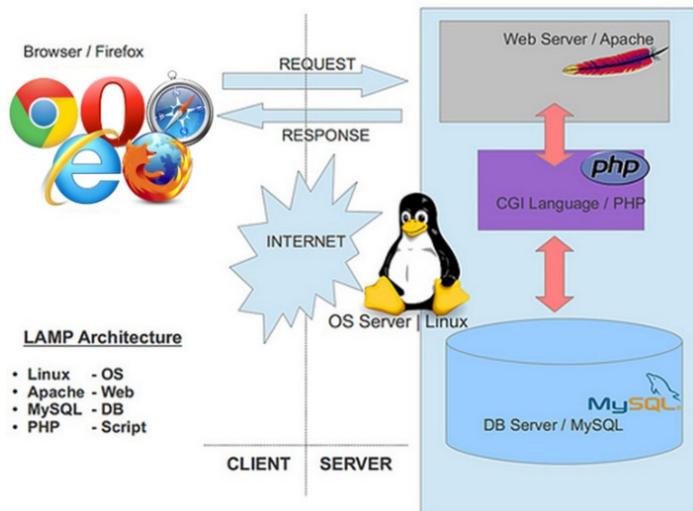


Figura 12. Arquitectura Cliente-Servidor de una aplicación tipo LAMP

En la figura 12 se ve cómo el servidor (Apache) distribuye la información a los clientes (navegadores) que la soliciten de manera que éstos interpretan dicha información. El resultado de la información interpretada por el cliente es la página web.

Existe una alternativa a este modelo y es el modelo de tres capas. Entre el cliente y el servidor se introduce una capa de procesamiento de datos y así los procesos pueden ser manejados de forma independiente a la interfaz de usuario y a los datos propiamente dichos. En la se figura 13 ve este modelo.

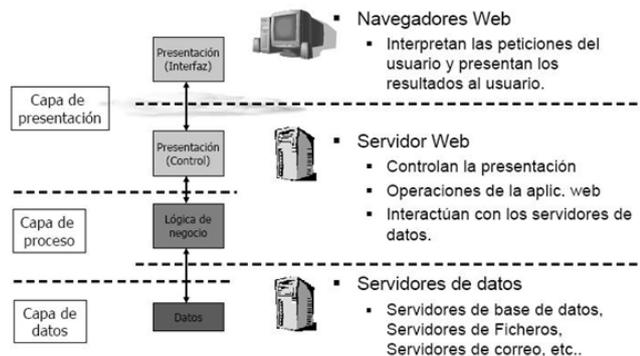


Figura 13. Modelo de 3 capas en una arquitectura Cliente-Servidor

6.3 Base de datos

En este apartado se tratan todos los aspectos de diseño relacionados con la base de datos. En primer lugar se especificará la tecnología a usar para su desarrollo, justificando la elección y posteriormente se detallará cada una de las partes que componen la base de datos y qué datos se almacenan en las tablas que componen la base de datos.

En el anexo A se muestra el diseño de la base de datos en un diagrama estructural.

La base de datos **schoolpipe_master** está implementada de forma que casi todo está relacionado con los usuarios. Como excepción, hay tablas de configuración o historial de acciones que no están interconectadas con ninguna otra tabla.

Distinguimos tres módulos importantes en la base de datos:

- Tablas de la configuración de la aplicación a bajo nivel.
- Tablas de la configuración de la aplicación a nivel de gestión de entidades.
- Tablas relacionadas con la gestión de usuarios, cursos, asignaturas y todo lo referente al centro escolar.

El *primer módulo* está compuesto por tres tablas que guardan la configuración de los campos de las tablas de base de la base de datos.

El *segundo módulo* está compuesto por las tablas que manejan las entidades que gestiona la aplicación, es decir, tablas de configuración de lo que hay disponible en la aplicación, de cómo se montan los menús de los usuarios, de qué rol tienen los usuarios, etc.

Por último, el *tercer módulo* está compuesto por las tablas que guardan todos los datos de los usuarios, de los cursos, los grupos, las asignaturas, los exámenes, los horarios, los mensajes entre usuarios, etc.

A continuación se explican las tablas y los campos más significativos de la aplicación. No aparecen en la descripción de atributos el campo identificador: todos los registros de las tablas tienen uno. El funcionamiento completo de las tablas que gestionan la funcionalidad genérica de la aplicación se explica en las tablas tabla 5 y tabla 11

Nombre	asignaturas ^(*)
Descripción	Almacena la información de las asignaturas.
Atributos	<ul style="list-style-type: none">• asignatura_label: el nombre de la asignatura.• asignatura_code: el código de la asignatura.• color: Color que recibe la asignatura cuando es pintada en el horario.

Tabla 1. Tabla asignaturas

^(*) Las tablas “Grupos” y “Cursos” son muy similares a la descrita aquí.

Nombre	back939_entity
Descripción	Almacena todas las entidades-módulos que soporta el sistema.
Atributos	<ul style="list-style-type: none"> • package: representa el paquete (carpeta) en la que se encuentra situada dicha entidad. • name: Es el nombre de la entidad. • label: etiqueta de la entidad. • type: Es el tipo de la entidad, por ejemplo PHP. • params: parámetros que tiene la entidad. • menú: si sale o no en el menú. • security: Si se puede ver sin iniciar sesión

Tabla 2. Tabla back939_entity

Nombre	back939_menu
Descripción	Almacena la información del menú y sus elementos.
Atributos	<ul style="list-style-type: none"> • parent_id: 0 si es un nodo raíz del menú o un ID indicando que ese nodo pertenece a un padre. • name: Es el nombre del elemento del menú. • label: etiqueta del elemento del menú. • roles: roles que tienen acceso al elemento de menú. • icon: icono, en formato proporcionado por font-awesome, que se pinta en el menú. • entity: entidad a la que hace referencia el elemento de menú.

Tabla 3. Tabla back939_menu

Nombre	back939_roles
Descripción	Almacena la información de los roles de los que dispone la aplicación.
Atributos	<ul style="list-style-type: none"> • name: Es el nombre del rol. • description: nota descriptiva del rol.

Tabla 4. Tabla back939_roles

Nombre	back939_tables
Descripción	Almacena la información de las tablas de la base de datos que pueden ser gestionadas directamente desde la aplicación.
Atributos	<ul style="list-style-type: none"> • DBNAME: Nombre de la base de datos a la que pertenece la tabla. • DBTABLE: Nombre de la tabla en la base de datos. • TABLENAME: Nombre o etiqueta que se le asigna a la tabla a administrar. • DESCRIPTION: Descripción de la tabla. • UNIQUEKEYFIELD: Campo de clave primaria. • USECOUNTER: Si es una tabla auto-incrementable. • QUERYMAXROWS: Número máximo de registros que devuelve esta tabla ante una consulta a dicha tabla. • INDICES: Índices de la tabla a administrar. • HASFILES: Indica si hay ficheros relacionados con dicha tabla, por ejemplo imágenes.

Tabla 5. Tabla back939_tables

Nombre	back939_user
Descripción	Almacena la información del usuario de la aplicación
Atributos	<ul style="list-style-type: none"> • user: nombre de usuario. • password: contraseña del usuario, en texto plano o md5. • name; nombre del usuario. • back939_rol_id: rol del usuario.

Tabla 6. Tabla back939_user

Nombre	faltas_alumnos
Descripción	Almacena las faltas de asistencia de los alumnos.
Atributos	<ul style="list-style-type: none"> • alumno_id: el alumno que ha faltado. • profesor_id: el profesor que pone la falta. • grupo_id: el grupo al que pertenece el alumno. • asignatura_id: asignatura de la falta • dia: fecha de la falta.

Tabla 7. Tabla faltas_alumnos

Nombre	calificaciones
Descripción	Almacena las calificaciones de los alumnos.
Atributos	<ul style="list-style-type: none"> • alumno_id: el alumno que ha obtenido la calificación. • profesor_id: el profesor que pone la calificación. • grupo_id: el grupo al que pertenece el alumno. • asignatura_id: asignatura a la que pertenece la calificación. • examen_id: id del examen. • nota_examen: calificación del alumno en el examen.

Tabla 8. Tabla calificaciones

Nombre	events
Descripción	Almacena los eventos de la aplicación.
Atributos	<ul style="list-style-type: none"> • event_owner: usuario que ha creado el evento. • title: título del evento. • body: cuerpo del evento. • class: tipo de evento creado. • start: fecha de inicio del evento. • end: fecha de fin del evento. • event_type_rol: indica los roles que pueden ver el evento. • event_user_id: usuarios que pueden ver el evento.

Tabla 9. Tabla events

Nombre	exámenes
Descripción	Almacena los exámenes.
Atributos	<ul style="list-style-type: none"> • examen_label: nombre del examen. • profesor_id: el profesor que pone el examen. • grupo_id: el grupo al cual va dirigido el examen. • asignatura_id: asignatura a la que pertenece el examen. • comentarios: comentarios de apoyo sobre el examen. • trimestre: trimestre en el que el examen será contabilizado. • tipo: tipo del examen (parcial/final) • fecha_examen: fecha del examen.

Tabla 10. Tabla exámenes

Nombre	back939_table_fields
Descripción	Almacena las columnas de la tabla administrada en la tabla anteriormente descrita.
Atributos	<ul style="list-style-type: none"> • tablename: Nombre o etiqueta de la tabla a la que pertenece el campo. • order_field: orden en el que aparecen los campos cuando se administran. • name: nombre del campo. • sqlPK: Si el campo es o no clave primaria. • sqlNullable: si el campo puede ser <i>NULL</i>. • label_field: etiqueta que tiene el campo. • group_field: grupo de campos al que pertenece el campo. Este valor sirve para agrupar todos los campos que tienen el mismo grupo para cuando se presentan los datos. • type: es el tipo de valor que acepta el campo. • min: Es el orden en el que el campo va a ser mostrado en el listado de resultados de la tabla administrada. Si no se desea que este campo salga en el listado, no se rellena valor en el min. • formType: en la ventana de inserción y edición de datos, este campo será mostrado en el formato que especifique formType. • size: tamaño máximo que permite el campo. • multi: si el campo permite valores múltiples. • separ: si el campo es múltiple, entonces los distintos valores están separados con un separador. • mandatory: si es obligatorio. • insert_field: si el campo aparece en el formulario de inserción. • update_field: si el campo aparece en el formulario de edición. • control: otro campo muy importante. Permite introducir valores especiales para parametrizar el formType. Además añade otros comportamientos como por ejemplo, introducir la fecha actual.

Tabla 11. Tabla back939_table_fields

Nombre	back939_user_activity
Descripción	Almacena la información de navegación del usuario por la plataforma.
Atributos	<ul style="list-style-type: none"> • id_user: usuario que ha iniciado sesión. • ip: ip del usuario que interactúa. • entity: entidad por la que pasa el usuario. • action: acción que hace en la entidad que está siendo solicitada. • request: parámetros de la url. • date_action: fecha en que ocurre el acceso.

Tabla 12. Tabla back939_user_activity

Nombre	customers
Descripción	Almacena los datos personales de los usuarios de la aplicación.
Atributos	<ul style="list-style-type: none"> • parent_id: 0 si no tiene padre o el id del padre en caso de ser alumno. • customers_gender: género del usuario. • customers_firstname: nombre del usuario. • customers_lastname: apellidos del usuario. • customers_dob: fecha de nacimiento del usuario. • customers_email_address: dirección de correo del usuario. • customers_nif: DNI del usuario. • customers_telephone_emergency1: primer teléfono de emergencia. • customers_telephone_emergency2: segundo teléfono de emergencia • customers_date_of_last_logon: fecha en que se ha registrado el último inicio de sesión del usuario. • customers_date_account_created: fecha en que se crea la cuenta del usuario. • customers_date_account_last_modified: fecha en la que los datos del usuario han sido modificados.

Tabla 13. Tabla customers

Nombre	messages
Descripción	Almacena los mensajes que se envían los usuarios de la aplicación.
Atributos	<ul style="list-style-type: none"> • message_from: emisor del mensaje. • message_to: destinatario del mensaje. • message_date_send_received: fecha de envío del mensaje por parte del emisor. • message_subject: asunto del mensaje: • message_content: contenido del mensaje. • message_status: mensaje leído o no leído. • message_trash_sender: si el mensaje está o no en la papelera del emisor. • message_trash_recipient: si el mensaje está o no en la papelera del receptor. • message_date_delete_sender: fecha en la que el emisor del mensaje ha borrado el mensaje. • message_date_delete_recipient: fecha en la que el receptor del mensaje ha borrado el mensaje

Tabla 14. Tabla messages

Nombre	timetable
Descripción	Almacena la organización de los horarios de las clases de los miembros del centro educativo. Cada registro representa un hueco en el horario.
Atributos	<ul style="list-style-type: none"> • dia_number: día de la semana. • hora_inicio: hora a la que empieza la clase. • hora_fin: hora a la que finaliza la clase. • profesor_id: el profesor que imparte la clase. • grupo_id: el grupo donde se imparte la clase. • asignatura_id: asignatura que se imparte.

Tabla 15. Tabla timetable

6.3.1 Conexión a la base de datos

La aplicación cuenta con un *wrapper* de Mysql. Las figuras 39 y 40 del apartado *Conexión a la base de datos* del anexo C ilustran cómo conectar la aplicación a la base de datos para poder hacer operaciones sobre ella.

6.4 Interfaz de Usuario

En este apartado se detallará cómo es la interfaz gráfica de la página web. Posteriormente se explicarán algunas de las pantallas de esta aplicación. Para desarrollar la interfaz se ha utilizado el *framework* de Bootstrap, el cual proporciona al usuario sencillez en la navegación al utilizar elementos muy intuitivos.

Además, Bootstrap aporta al programador las siguientes facilidades:

- Es un *framework* fácil e intuitivo de usar siguiendo las pautas descritas en la documentación [5].
- Gracias a Bootstrap, el desarrollo de la interfaz es idéntico en todos los navegadores incluso en los de los dispositivos móviles.
- Bootstrap se encuentra optimizado para los dispositivos móviles: proporciona a la propia página web el denominado *diseño responsive*, es decir, que la página se adapte al tamaño del dispositivo que reproduce la web.
- Es un *framework* actualizado constantemente por una gran compañía como es Twitter.

La figura 14 muestra cómo es la estructura de la página web de este proyecto software programado con Bootstrap.

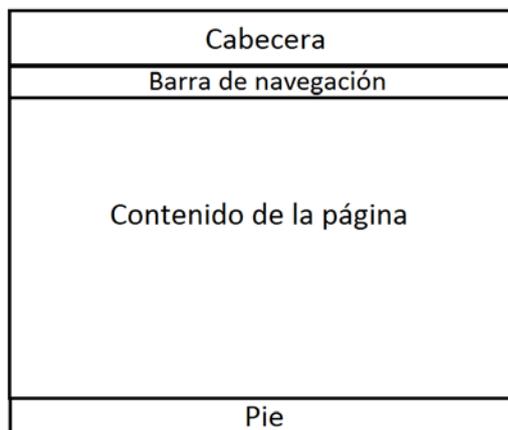


Figura 14. Estructura de la página web

Tanto la cabecera, como el pie y la barra de navegación son partes de la página que no cambian con la navegación del usuario, pero el contenido central sí que cambia.

Bootstrap forma sus páginas de la siguiente manera: existe un contenedor principal que limita el ancho de la página. El contenedor puede tener en su interior filas de *12 huecos*. Bootstrap tiene módulos de tamaño (columnas) entre el uno y el doce. Es decir,

lo ideal cuando se diseña en Bootstrap es tener columnas que encajen en una rejilla de doce elementos (véase el anexo B). Las filas no se superponen unas a otras.

La página principal de **SchoolPipe** se estructura como muestra la figura 15. Los rectángulos exteriores representan los contenedores de la página, los interiores alargados indican las filas y el rectángulo interior de color verde, las columnas dentro de cada fila. Bootstrap tiene la peculiaridad de que, en el sistema de columnas, cuando una columna no entra en una fila (por tamaño), se genera una nueva. En esta imagen, el contenido central de la página está dividido en dos columnas de seis huecos cada una.

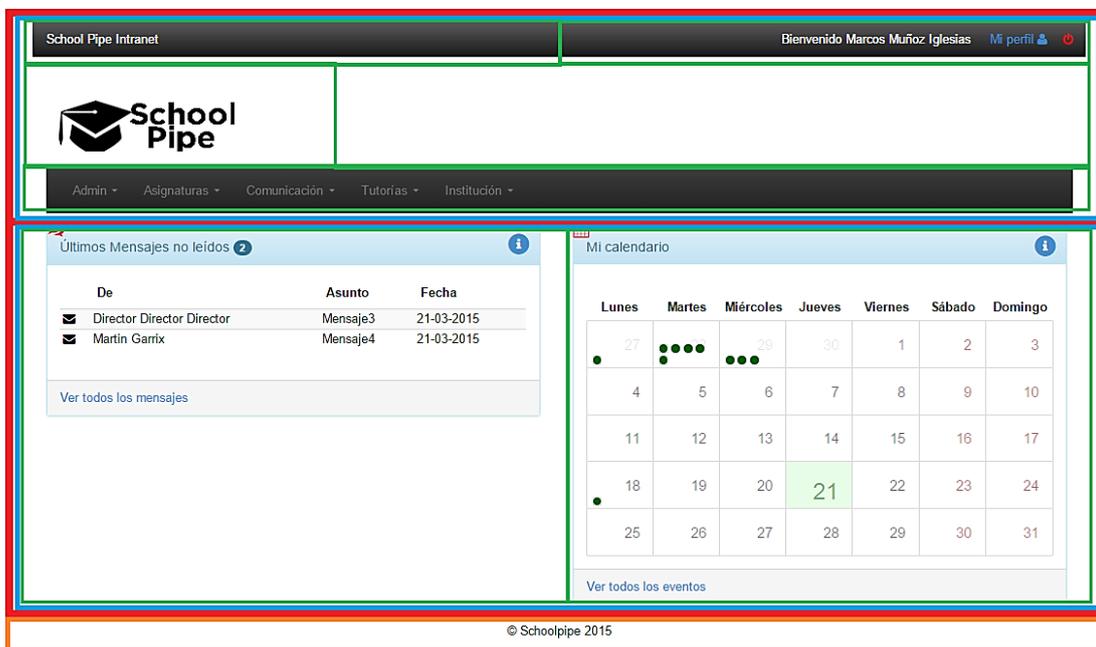


Figura 15. Estructura de la Home de SchoolPipe

6.5 Estructura de la aplicación

Como ya se ha mencionado, la aplicación se ha implementado en PHP, como lenguaje de programación principal en el lado del servidor. Este lenguaje es muy utilizado en la programación web y cuenta con soporte para muchas operaciones como son conexión y manejo con bases de datos, interpretación de ficheros XML, JSON, diseño del Modelo-Vista-Controlador y un sinnúmero de librerías auxiliares, como por ejemplo, detección de dispositivos móviles, envío de e-mails mediante el servicio que proporciona Amazon SES o generación de documentos en PDF.

El código de esta aplicación incluye todo lo relacionado con conexión, acceso y operaciones con la base de datos, la implementación del Modelo-Vista-Controlador mediante entidades, de lo cual se hablará más adelante.

En el lado del cliente se ejecutan otros lenguajes y técnicas de diseño web como son HTML, Javascript, CSS, AJAX, etc. En esta sección se verá el diagrama de clases de la aplicación resultante, el modelo utilizado para implementar el sistema y cómo se ha diseñado e implementado dicho modelo.

- **TemplateContext:** es la clase que permite añadir y recoger variables en el contexto de las plantillas.

La tercera sección es la que engloba todos los módulos que engloban la aplicación. Se parte de una clase abstracta principal que es *GenericEntity*. Todo módulo que quiera ser implementado en la aplicación debe extender de esta clase e implementar tres métodos, que son abstractos en la clase padre, para poder funcionar. El funcionamiento de estos métodos se describe en el siguiente punto. De estas clases que heredan de *GenericEntity*, hay varias que son para administrar y configurar la aplicación y otras que son más propias de la funcionalidad que **SchoolPipe** quiere hacer llegar al usuario que no es administrador.

6.5.2 Diseño e implementación del modelo de la aplicación

Esta plataforma web ha sido diseñada e implementada utilizando el modelo de arquitectura Modelo-Vista-Controlador (MVC). Esto quiere decir que la aplicación separa tres componentes principales: el modelo, la vista y el controlador.

Las aplicaciones web escritas con el *framework* MVC pueden actualizarse y mantenerse fácilmente. Por ejemplo, los programadores pueden implementar nuevos módulos con sus respectivas vistas sin necesidad de reescribir toda la aplicación, es decir, que el *framework* está basado en la reutilización de código. En el ámbito web, el MVC tanto el controlador, como el modelo y gran parte de la vista están alojados en el servidor. El MVC permite que ciertos componentes como Javascript o AJAX se ejecuten en el lado del cliente.

A continuación se detallan las características de cada una de estas capas de la arquitectura y cómo han sido implementadas en este proyecto:

Controlador

El controlador es aquel componente que responde a los eventos generados por el usuario en la página web. El controlador realiza llamadas al *Modelo* para, por ejemplo, hacer todo el procesamiento de datos. También puede enviar información a “su *Vista*”, por tanto, el controlador es la conexión entre el *Modelo* y la *Vista*.

Modelo

El modelo es aquella parte que gestiona toda la información con la que trabaja el sistema: actualizaciones, consultas, etc. En este proyecto, estas acciones están restringidas a los privilegios según se han definido en 3.1 Objetivos específicos y funcionalidades. Desde esta parte se envía a la vista la información que fue solicitada por el usuario. Como hemos mencionado en el punto anterior, las peticiones de acceso llegan a través del *Controlador*.

Vista

La vista presenta el *Modelo* en un formato en que el usuario pueda interactuar con los datos que éste solicita al sistema.

Implementación del modelo de la aplicación

En este punto se verá con detalle cómo se ha implementado el flujo que hace la aplicación.

La figura 17 muestra la codificación del punto de entrada (*index.php*) de la aplicación. Cualquier petición realizada, va a este fichero. Podríamos deducir que este punto de entrada es parte del controlador general pues es aquí donde se decide a qué controlador específico está llamando el usuario. Este proceso es lo que se conoce como *bootstrapping*, es decir, en ámbito de PHP, un fichero que maneja todas las solicitudes dinámicas por parte del usuario.

```
1 <?php
2 include_once './app/code/includes/Util.php';
3 ini_set('memory_limit', '2048M');
4 header('Content-Type: text/html; charset=windows-1252');
5 header('X-UA-Compatible: IE=edge,chrome=1');
6 Util::loadBasic(); //Se cargan los ficheros y configuraciones básicas.
7 session_start(); //iniciamos la sesión después de cargar los ficheros, importante
8 $backSession = Util::loadSession(); // se carga la sesión guardada si hubiera.
9 $backSession->loadParams();
10
11 // saco entity
12 $entityId = Util::getEntityParam();
13 $entity = Util::getEntityInstance($entityId);
14 if ($entity->checkAccess()) { //permiso??
15     $entity->log();
16     $entity->action();
17     Util::printHead();
18     // el menu y la cabecera se pinta despues del action para tener en cuenta los
19     // cambios producidos por el action de la entidad
20     Util::printMenu();
21     $entity->render(); //pintará los mensajes y las plantillas
22 }
23 } elseif ($backSession->getUser() == null) { //Sesión Cerrada!!!!!!
24     Util::redirect("index.php?entity=1");
25 } else { //Simplemente no tiene permisos. Pintamos lo necesario con el botón de volver...
26     Util::printHead();
27     // el menu y la cabecera se pinta despues del action para tener en cuenta los
28     // cambios producidos por el action de la entidad
29     Util::printMenu();
30     Util::printMessage();
31 }
32 Util::printFooter();
33 Util::end();
```

Figura 17. Index de la aplicación

Este fichero lo que hace es incluir varios ficheros y configuraciones específicas, iniciar una sesión nueva y decidir si en dicha sesión se carga una nueva o se reanuda una ya existente. Esto último sólo ocurre en el caso de que no exista cierto parámetro en la sesión. Si existiera indicaría que ya hay una sesión previa y lo que se hace es cargar dicha sesión.

Con la sesión cargada se lee la entidad y la acción que viajan por el *GET* y el *POST*.

Una vez que se tiene el parámetro que indica la entidad que hay que instanciar, se instancia y se comprueba lo siguiente:

- Si el usuario que está ejecutando dicha entidad tiene permiso para acceder a ella.
- Si hay un usuario con la sesión iniciada.

En caso de que el usuario haya iniciado sesión y tenga permisos, se suceden siempre las mismas operaciones de la entidad elegida:

- Se guarda el registro de la actividad del usuario.
- Se ejecuta el método *action*. Este método es lo que consideraremos el Modelo de la arquitectura.
- Se pinta el menú de la aplicación.
- Se ejecuta el método *render*. Este método es lo que consideraremos la *Vista* de la arquitectura.

El método *action* se encarga de realizar operaciones de base de datos, interpretado de datos, actualizaciones, etc.

El método *render* se encarga de imprimir los mensajes (operaciones correctas, erróneas) que hubiera podido generar el método *action* y de seleccionar la plantilla que se va a pintar por pantalla.

La gestión de las plantillas se hace mediante la clase *TemplateContext*. Los ficheros de plantilla se guardan en la carpeta *design*. Si la plantilla es global se guarda dentro de la carpeta *global* (menú, pie, ..., etc.) si es específica de una plantilla se coloca dentro de la carpeta *entity*. Las plantillas tienen la posibilidad de recibir variables enviadas con el controlador y el modelo.

En el anexo C se muestran capturas ejemplo de los métodos *action* y *render*, envío y recepción de variables a las plantillas y cómo es la distribución de las carpetas de este proyecto.

Flujo de control del MVC

1. El usuario, a través de la interfaz de usuario, solicita cierta información cuando pulsa, por ejemplo, un botón de guardar cierta acción.
2. El punto de entrada de la aplicación recibe la información de la entidad y la acción solicitada (mediante la interfaz).
3. De aquí se deriva al controlador correspondiente (entidad) y se ejecuta el modelo con la acción solicitada.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene los datos del modelo para generar la interfaz apropiada para el usuario donde se reflejen los cambios en el modelo.
5. La interfaz de usuario se recarga con las modificaciones esperadas y espera nuevas interacciones para comenzar el ciclo nuevamente.

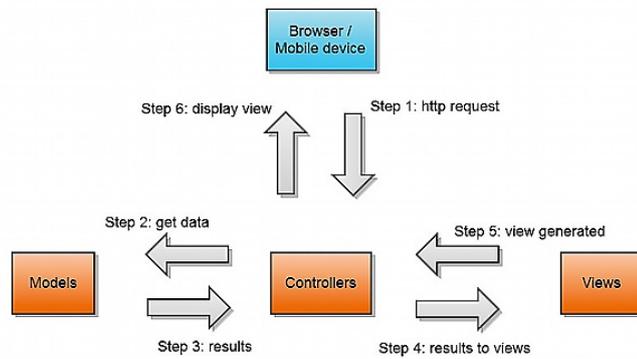


Figura 18 Estructura y pasos del Modelo-Vista-Controlador
(Fuente: *Yii 4 Beginners*)

6.5.3 Implementación de modelos más destacados

En este punto se va a explicar con detalle la implementación de aquellos módulos que se han considerado que son más relevantes.

Entidad gestor de tablas y editor de las mismas

Como se ha comentado en abundantes puntos de esta memoria, la aplicación tiene implementado un sistema de gestión de tablas de la base de datos. Esta funcionalidad no es específica del sistema de comunicación interna y gestión académica de un centro escolar, sino que es una funcionalidad más global y reusable en cualquier otro sistema.

En este punto se pretende que el lector comprenda el funcionamiento y alcance que tiene esta funcionalidad.



Figura 19. Opciones de funcionalidad de la gestión y administración de tablas

La figura 19 muestra las cuatro opciones que tiene esta funcionalidad.

El sistema permite que las configuraciones sean replicables entre sistemas que compartan la misma base de datos. Esto, por ejemplo se puede utilizar para traspasar

información entre entornos de desarrollo y entornos de producción. Para ello utilizamos la sección azul (rectángulo inferior) que se ve en la figura 19. Cuando se exporta la configuración de una tabla seleccionada, ésta aparece en el cuadro de color rojo. Para importar esa configuración habría que pegar la configuración exportada en el cuadro de color naranja.

Además, el sistema permite, mediante las opciones que se pueden encontrar en el cuadro de color verde, seleccionar cualquier tabla de la base de datos e introducirla a la administración de tablas.

Se puede decir que cada administración de una tabla es, si se desea, una posible forma de ver la información de una tabla, es decir, pueden existir infinitas configuraciones de la misma tabla. Esto podría servir para que la tabla *usuarios* sea vista y administrada de una manera para el usuario con rol “Director” y otra diferente para el usuario con rol “Administrador”.

id	package	name	label	type	params	menu	security
15	ediciontablas	EdicionTablas	Editor de asignaturas	PHP	table=asignaturas		true

Figura 20. Entidad que gestiona asignaturas

La figura 21 muestra la pantalla de una configuración de una tabla. En la parte superior (color rojo) se muestran los datos que se va a guardar en la tabla 5, como por ejemplo la etiqueta de la tabla, valor muy importante si se quiere acceder al visionado y la edición de los registros de esta tabla. Para acceder a estos registros se debe crear una nueva entidad en la tabla 2 tal y como se ve en la figura 20. Otro campo a tener en cuenta es el que indica si la tabla que se está configurando tendrá la opción de subir ficheros. Por ejemplo, la tabla de usuarios tendría el *check* de *HASHFILES* activo porque, en el panel de edición de un usuario se podría subir una foto de dicho usuario.

Como se aprecia en la imagen, en el valor de los parámetros se introduce el valor *table=asignaturas*, es decir, cualquier tabla administrada debe incluir en el campo *parámetros* la tabla a administrar para que el sistema pueda localizar los datos de esta tabla. Para poder acceder a los registros de la tabla administrada, la entidad que se genera debe ser incluida en el menú de navegación.

También es importante el valor de líneas por página. Esto hace referencia al número de registros que salen por página cuando se muestra la información de la entidad.

En la parte inferior (color verde) de la figura 21 es donde se encuentra la información de los campos de la tabla a administrar. Estos datos están explicados en la tabla 11, además de ser ahí donde se guardan cuando el usuario termina la edición.

En este punto se verá la misión que tienen seis de los campos.

1. El campo **label** permite poner una etiqueta que todo usuario pueda entender al campo de la tabla.
2. El campo **group** permite agrupar varios campos sobre un mismo fragmento de formulario. Se explica este concepto más adelante.
3. El campo **min** permite elegir si se ve o no se ve el campo en el listado de registros. Si se quiere ver, sólo hay que especificar una posición.

- Los campos *ins* y *act* permiten elegir si los campos son visibles en el formulario de inserción y en el formulario de edición.
- El campo *formType* permite elegir en qué tipo de campo de un formulario se representa el campo.
- El campo *control* permite seleccionar alguna configuración especial para el *formType*. Más adelante se especificará más información sobre los posibles controles implementados hasta la fecha.

Table -> asignaturas

Nombre Database: Contador:

Nombre Tabla: Líneas por página:

Etiqueta de Tabla: Versión de control:

Descripción: Incremento interno:

Clave Única: ¿Tiene ficheros?:

[Actualizar tabla](#)

Nº	Campo	Label	sqlPK	sqlNull	group	Tipo	Min	formType	Tam	Multi	Sep	Obli	Ins	Act	Bus	Vis	Uni	Control
1	asignatura_id	Identificador	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Datos	integ	1	Tex	11	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	asignatura_lal	Nombre	<input type="checkbox"/>	<input type="checkbox"/>	Datos	texto	2	Tex	50	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	asignatura_co	Código	<input type="checkbox"/>	<input type="checkbox"/>	Datos	texto	3	Tex	255	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	color	Color de Hora	<input type="checkbox"/>	<input type="checkbox"/>		texto		Tex	255	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

[Actualizar tabla](#)

Figura 21. Pantalla de administración de tablas

Una vez que la configuración ha sido guardada, la entidad ha sido creada en la base de datos y ya es accesible desde el menú, los registros se pueden visitar. En el listado que se muestra en la figura 22 aparecen los diez registros que se configuraron en la figura 21. Además, se puede buscar y ordenar por cualquiera de los campos que se muestran en el listado. Si en la configuración de la tabla se hubieran escogido mostrar sólo cinco registros por página, su resultado sería el que se muestra en la figura 23. Aquí se muestra cómo, al no caber todos los registros en una página, existe una página más en la sección de paginación.

[🔍](#)

	Identificador *	Nombre *	Código *	
1	0	Libre		<input checked="" type="checkbox"/>
2	1	Ciencias Naturales	NATU	<input checked="" type="checkbox"/>
3	2	Educación artística : plástica y música	EDUART	<input checked="" type="checkbox"/>
4	3	Educación física	EDUCAFIS	<input checked="" type="checkbox"/>
5	4	Valores Sociales y Cívicos	VALCIV	<input checked="" type="checkbox"/>
6	5	Lengua castellana y literatura	LENGCAS	<input checked="" type="checkbox"/>
7	6	Lengua extranjera	LENGEXT	<input checked="" type="checkbox"/>
8	7	Matemáticas	MATEMAT	<input checked="" type="checkbox"/>
9	8	Religión	RELIG	<input checked="" type="checkbox"/>
10	9	Ciencias Sociales	SOCI	<input checked="" type="checkbox"/>

[-- Primera](#)
[◀](#)
[1](#)
[▶](#)
[-- Última](#)

Figura 22. Listado de datos con 10 registros por página

	Identificador ^	Nombre ^	
1	0	Libre	
2	1	Ciencias Naturales	
3	2	Educación artística : plástica y música	
4	3	Educación física	
5	4	Valores Sociales y Cívicos	

Figura 23. Listado de datos con 5 registros por página

Si se edita un registro (o insertamos uno nuevo) la página que se ve es la que se muestra en la figura 24. Se aprecia perfectamente como se ha creado un grupo de datos que se llama **Datos** tal y como se especificó en la figura 21. Además, también se especificó que el campo identificador no se mostrara en la actualización de un registro (campo **ins**).

Edición asignaturas

Datos

Nombre	Código
Libre <input type="text"/>	<input type="text"/>

Color de Horario
#F7FF90 <input type="text"/>

Figura 24. Página de inserción y edición de registros

A continuación se van a explicar los diferentes tipos de valores que el campo **control** acepta y cuál es su funcionalidad.

- Si el campo **formType** es de tipo **dblink** significa que ese campo esconde realmente un id asociado con otra tabla. Por ejemplo, en la tabla que relaciona un alumno con un grupo, para saber a qué grupo pertenece, en la base de datos lo que se guardan son el id del alumno y el id del grupo. Si el director quisiera administrar esta tabla desde la aplicación sería muy complicado que supiera qué id corresponde con qué alumno. Para salvar este problema en el campo **control** se escribe la siguiente sintaxis:

tabla_referenciada:id_en_la_tabla,campo_referenciado

Así se consigue que a partir del id, se busque en la tabla el campo referenciado. En el caso del id del grupo habría que escribir lo siguiente:

grupos:grupo_id,grupo_name

En el formulario de datos (figura 25) esto aparecería como un selector de los grupos disponibles en la tabla **grupos**.

Alumno	Grupo
esther.guerrero <input type="text"/>	<input type="text" value="Sexto A"/> <ul style="list-style-type: none"> Cuarto A Primero A Quinto A Quinto B Segundo A <li style="background-color: #007bff; color: white;">Sexto A Tercero A

Figura 25. Tipo de campo **dblink**

- Si el campo *formType* es de tipo *dblinkSpecial* ocurre lo mismo que en el caso anterior con la peculiaridad de que el selector incluye un buscador. Esto es muy útil cuando el selector ofrece muchas posibilidades. En la figura 26 se ve un ejemplo de este campo.

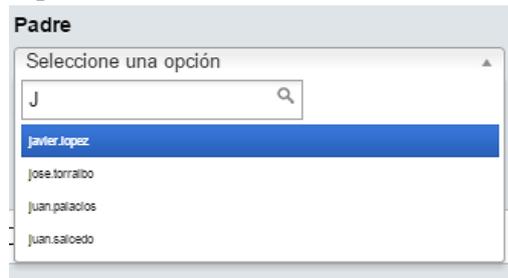


Figura 26. Tipo de campo *dblinkSpecial*

- Si el campo *formType* es de tipo *list* ocurre algo parecido a lo anterior: se genera un selector con las opciones definidas en una lista. En el campo *control* se introduce el nombre de una de las listas almacenadas en la tabla *back939_list*. El formato de estas listas es el siguiente:

Opcion1:valor1/opcion2:valor2/opcion3:valor3.../opcionN:valorN

Donde *opción* es la parte que se ve en el selector y *valor* es la parte que se guarda en la base de datos en caso de ser elegida esa opción

Por ejemplo, en esta aplicación se usa una lista para identificar los géneros de los usuarios. En la figura 27 se ve dicho ejemplo y la declaración de la lista es la que sigue:

m:m/f:f



Figura 27. Tipo de campo *list*

- Si el campo *formType* es de tipo *Fecha* y en el campo *control* se introduce el valor

ins:datetime o *upd:datetime*

Significará que el campo al que hacen referencia debe guardar la fecha en el momento de la inserción del registro o en el momento de la actualización de un registro ya existente.

Existen otras posibilidades, como por ejemplo, rellenar un campo mediante una llamada AJAX que vaya autocompletando lo que el usuario introduce en el campo. Por ejemplo, cuando hay que elegir un e-mail de entre más de 50.000 posibilidades, es recomendable hacer uso de una llamada AJAX para autocompletar el campo ya que cargar 50.000 registros en un selector podría ser demasiado pesado para el navegador que ejecuta la página.

Entidad eventos de calendario

Esta entidad se encarga de gestionar los eventos del calendario.

Para crear un evento se hace *click* sobre “Añadir evento” y se genera un evento AJAX en la aplicación. Este evento, como se aprecia en la figura 28 , hace una llamada a una entidad que gestiona eventos de tipo AJAX y pinta por pantalla una ventana modal de Bootstrap (figura 29) con los datos para rellenar eventos.

```
1 function ajax_show_add_evento() {
2   resetModal();
3   $('#modAddEvent').modal('show');
4   $.ajax({
5     url: rutaAjax + "?entity=18",
6     contentType: "application/json; charset=utf-8",
7     dataType: "json",
8     evalScripts: true,
9     success: function (response, textStatus, jqXHR) {
10      $('#modAddEvent .modal-content').html(response.html);
11    },
12    error: function (jqXHR, textStatus, errorThrown) {
13      alert("Ha habido un error al realizar la operación. Intentelo de nuevo. \n [" + errorThrown + "]);
14    }
15  });
16 }
17
18 function ajax_show_editar_evento(id) {
19   resetModal();
20   $('#modAddEvent').modal('show');
21   $.ajax({
22     url: rutaAjax + "?entity=18&eventId=" + id,
23     contentType: "application/json; charset=utf-8",
24     dataType: "json",
25     success: function (response, textStatus, jqXHR) {
26      $('#modAddEvent .modal-content').html(response.html);
27    },
28    error: function (jqXHR, textStatus, errorThrown) {
29      alert("Ha habido un error al realizar la operación. Intentelo de nuevo. \n [" + errorThrown + "]);
30    }
31  });
32 }
```

Figura 28. Funciones que añaden y editar un evento existente

Cuando se han rellenado los datos del evento que el formulario solicita se hace *click* sobre el botón *guardar*. Existe una validación en la propia página gracias a la librería Javascript *verify.js* [52]. Si todo está correcto entonces se envía al controlador la opción de guardar evento. El modelo trata todos los campos del formulario y guarda los datos en la base de datos.

Cabe destacar que en el formulario, según el tipo de evento, existe la posibilidad de invitar a otros usuarios a un evento. Para elegir las personas invitadas se utiliza el *plugin* de JQuery *autocomplete*. Dicho *plugin*, cuando el usuario introduce tres caracteres, genera un evento AJAX que busca en la base de datos todos los nombres o correos electrónicos que coinciden con dichos caracteres. Si existen resultados, se muestran en una lista con las posibilidades. Si existe una posibilidad a la que el usuario no tiene acceso, no se muestra.

Añadir un nuevo evento

Tipo de evento

¿Invitar a alguien?

Título

¿Todo el día?

Fecha inicio

Fecha Fin

Para elegir la fecha pincha sobre el icono de la derecha.

Comentarios

¿El evento es privado?

Si no es privado... ¿Quién puede ver este evento en su calendario?

- Profesores
- Padres
- Alumnos
- Todos

Guardar evento

Si lo que deseas es concertar una reunión o enviar un mensaje privado a algún usuario, pincha [aquí](#)

Figura 29. Ventana modal para rellenar los datos de un evento

Tras guardar el evento se hace uso de uno de los amplios servicios que proporciona Amazon Web Services: *Amazon Send Email Service*. Este servicio se utiliza para notificar a todos los invitados, de haberlos, que un usuario le ha invitado a un evento.

Para editar un evento, el proceso es exactamente el mismo, pero el usuario que quiere editar sólo puede editar un evento creado por él mismo. Cuando se hace *click* sobre un evento existente el evento AJAX busca sus datos en la base de datos y completa el formulario con dichos datos.

Entidad de gestión de comunicación

Esta entidad permite el envío de mensajes entre los distintos miembros del centro escolar.

Mis comunicados

Recibidos 2 | Enviados | Papelera

Borrar | Responder | Marcar como No Leído

	De	Asunto	Fecha
<input type="checkbox"/>	Director Director Director	Número de personas en listados	19-04-2015
<input type="checkbox"/>	Director Director Director	Mensaje3	13-04-2015
<input checked="" type="checkbox"/>	Javier López	Convocatoria Junio TFG	21-03-2015

Figura 30. Página de comunicados

La figura 30 ilustra la interfaz de esta entidad. A la izquierda se sitúan las tres posibilidades del gestor de mensajes: bandeja de recibidos, bandeja de salientes y papelera.

En la parte de la derecha aparecen los mensajes de la bandeja elegida, por orden de recepción. Si están en negrita significa que no se han leído. Los botones de la parte superior aparecen sólo cuando se selecciona un mensaje. Dichos botones expresan la

opción de borrar el mensaje, responderlo o marcarlo como no leído. En la parte superior, aparece la opción de redactar un nuevo mensaje y de actualizar la página.

En la figura 31 se muestra la pantalla de respuesta de un mensaje, exactamente igual que la de nuevo mensaje. En ella se ve que se puede rellenar el asunto, el listado de receptores y el cuerpo del mensaje. El editor del cuerpo, como se dijo en el apartado 4.2, se realiza mediante el editor HTML *CKEditor*.

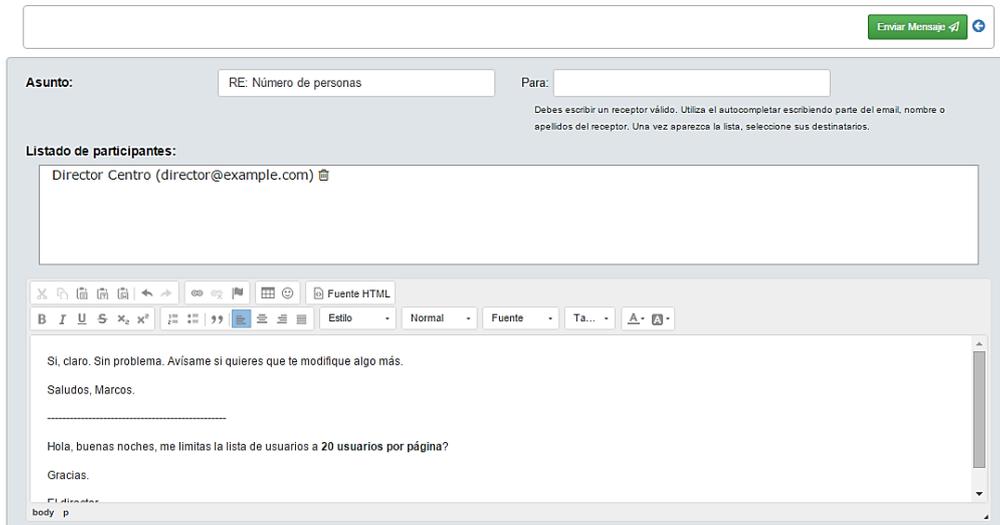


Figura 31. Pantalla de respuesta de un mensaje

Esta entidad, como la anterior explicada, también hace uso del *plugin* de autocompletar usuarios y de envío de un e-mail informativo a los receptores.

La vista de esta entidad se encarga de cargar los mensajes según sus categorías (enviado, recibido, papelera) en caso de querer mostrar la información. También tiene la capacidad de cargar la plantilla de redactar un mensaje si la acción que solicita el usuario es *viewMsg*, *newMsg* o *responderMsg*.

El modelo se encarga de guardar en la base de datos la información de los mensajes y de notificar a los receptores mediante e-mail de que tiene un mensaje privado. Se puede ver una muestra en la figura 47 del anexo C.

En la página principal de todos los usuarios aparece un resumen de los mensajes recibidos sin leer.

Entidad de gestión de exámenes

Esta entidad se encarga de gestionar las operaciones que rodean a los exámenes: creación, edición y calificación. Además, es específica del rol de profesor. Puede crear exámenes nuevos y editar los ya existentes. En la figura 32 se muestran estas opciones.

Listado de Exámenes

Elige un examen Parcial de febrero-Ciencias Naturales-Primer A Editar Examen Calificar Examen

Insertar un examen nuevo

Insertar nuevo

Figura 32. Página de gestión de exámenes

Cuando el profesor selecciona la opción de crear nuevo examen debe rellenar un formulario con los datos del examen. Cuando el profesor guarda los datos la aplicación envía un e-mail a los alumnos y familiares de éstos notificando dicho examen. En la figura 33 se muestra el formulario que ha de ser rellenado.

Crear examen

Crear examen

Grupo Primer A	Asignatura Ciencias Naturales	Nombre del examen p.e. Parcial 1
Tipo Parcial	Periodo Primer Trimestre	Día del examen AAAA-MM-DD
Comentarios y/o contenido		

Figura 33. Página de creación de exámenes.

Una vez se crea el examen, el profesor vuelve a la pantalla de la figura 32 y selecciona *Calificar examen* si quiere poner notas a los alumnos de ese grupo. En la figura 34 se muestra la pantalla en la que el profesor asigna las calificaciones del examen seleccionado.

Guardar calificaciones Volver

Calificar examen -->Parcial de febrero-Ciencias Naturales

Alumnos del grupo Primer A	Nota
Alberto Martín García	3
Javier López García	5
Miguel Pancorbo Iglesias	6

Figura 34. Página de calificaciones de examen

6.6 Generación de PDF

La aplicación incluye una librería de generación de PDF a partir de código HTML.

La aplicación dispone de un punto de entrada especial que sólo sirve para la generación de archivos PDF, es decir, sigue casi la misma lógica explicada en el apartado 6.5.2 con la peculiaridad de que se genera un objeto de tipo *WKPDF*, se llama a la entidad que muestra los datos que han de salir en el PDF y se incluye la parte de la vista de dicha entidad. Finalmente será la librería la que se encargue de realizar las operaciones que generan el PDF. En la figura 48, del anexo C, se muestra cómo funciona dicha librería.

7 Validación y Verificación

7.1 Alcance de las pruebas

En este apartado se listarán las distintas pruebas de validación ejecutadas durante la fase de pruebas.

Al haber utilizado un modelo de ciclo de vida en el que el usuario final tiene una participación muy activa, el plan de pruebas que aquí se presenta está muy relacionado con las acciones más conflictivas que el usuario final puede hacer con la aplicación, como por ejemplo, la generación de eventos, la creación de mensajes o la calificación de exámenes.

Las pruebas realizadas validan y verifican el proyecto en cuanto a funcionalidad, compatibilidad y accesibilidad.

7.1.1 Funcionalidad

La funcionalidad de la aplicación se prueba para descubrir errores que indiquen que no hay concordancia con los requisitos funcionales establecidos en el apartado 5.3.

7.1.2 Compatibilidad

La plataforma, en esta versión, está adaptada a los navegadores web más usados. Además, gran parte de las pantallas que se presentan en la aplicación están adaptadas para navegadores utilizados dispositivos móviles. En el anexo E se muestran capturas de algunas pantallas tanto en escritorio y en dispositivos móviles.

7.1.3 Accesibilidad

Se prueba para garantizar el grado en el que todas las personas pueden acceder y utilizar la aplicación con sus opciones de configuración establecidas previamente.

7.2 Verificación

El objetivo de la verificación de esta aplicación es comprobar si la funcionalidad de la misma ha sido implementada correctamente.

Se han realizado todo varios tipos de pruebas, incluyendo las pruebas de caja gris (caja blanca + caja negra), pruebas de regresión que permitan verificar nuevamente el sistema tras modificaciones o ampliaciones del mismo o pruebas de la interfaz de usuario en la que se viera que los elementos estaban correctamente situados y fueran intuitivos para el usuario. El objetivo de cada una de las pruebas es encontrar el mayor número de errores y solventarlos para entregar un producto en buen estado.

7.2.1 Estrategia de pruebas y desarrollo de las mismas

Dada la distinta naturaleza de los elementos que componen la aplicación, la estrategia de pruebas que se ha seguido no es la misma para todos ellos.

A continuación se detallan las técnicas utilizadas para la realización de las pruebas de verificación así como qué componentes han sido probados mediante estas técnicas. También se especificarán cómo se han realizado estas pruebas y qué resultados se han obtenido de las mismas.

Lectura del código

La técnica de inspección de código, consistente en una lectura del mismo con el fin de localizar errores, se ha utilizado en todas las entidades, concretamente en la sección de código que se encarga de tratar los datos que se reciben y se guardan en la base de datos, comprobando que estos sean correctos. Al tratarse de operaciones específicas, se han encontrado errores que han causado un tratamiento de datos erróneo lo cual implica que el resultado esperado no fuera el deseado. Dado que la longitud de todo el proyecto es amplia, el número de estos errores suele ser abundante, pero ya que el diseño está preparado para separar las entidades, los errores se centralizan en una única entidad. Aun así, la lectura de código ayuda a la detección de estos errores.

A medida que se iba desarrollando se iba comprobando que la sintaxis de las líneas de código era correcta o que el nombre de las variables era el correcto. Por ejemplo, en la entidad que guarda los eventos generados por el usuario existían errores de intento de tratar una variable cuando aún no contenía el valor que se esperaba.

Pruebas de caja blanca y de caja negra

Las pruebas de caja blanca son aquellas que se ejecutan teniendo en cuenta la estructura interna del programa. Estas pruebas tienen como objetivo la ejecución de todas las posibilidades existentes en el código.

Las pruebas de caja negra son aquellas que se ejecutan sin tener en cuenta la estructura interna del programa, es decir, sólo se tiene en cuenta la entrada y la salida.

En este proyecto no se han realizado ni pruebas de caja blanca ni de caja negra como tal, sino que, al ser el desarrollador el único que podría haber hecho las pruebas durante la fase de codificación, lo que se han realizado son pruebas de caja gris, es decir, se han realizado pruebas de caja negra basadas en casos de prueba realizados por personas que conocen la estructura interna del programa. Las pruebas realizadas de esta manera son más efectivas porque se conocen las partes del código que pueden resultar más conflictivas.

Con este tipo de pruebas se han probado los métodos explicados en el apartado 6.5.2: *action* y *render* de todas las entidades programadas.

Por ejemplo, en la entidad *GestionTablas* se ha probado que, en función de los parámetros que llegaban al método *action*, estos eran recogidos correctamente y procesados correctamente para guardarlos en la base de datos. Dada la complejidad de esta entidad por toda su funcionalidad comentada en el apartado 6.5.3 esta entidad ha sido, junto a la entidad *EdicionTablas*, la que más problemas ha ocasionado y por tanto, más pruebas se han realizado. Tras muchas correcciones, no sólo en estas dos entidades, todas funcionan como debieran, por tanto los resultados obtenidos son los correctos.

Pruebas de integración

Las pruebas de integración tienen el objetivo de probar varios componentes de la aplicación una vez estos han sido integrados entre sí. Para esta aplicación, esto ha servido para comprobar que todos los ficheros estaban correctamente conectados unos con otros y que, por ejemplo, se podían, desde una *clase A*, llamar a métodos escritos en la *clase B*.

Por ejemplo, en esta aplicación se ha comprobado que se podían llamar a las operaciones de base de datos desde cualquier punto de la aplicación, una vez que la clase *Database* había sido incluida al principio de la ejecución de la aplicación. Los resultados de este tipo de pruebas satisfacen la necesidad del desarrollador y de la aplicación.

Pruebas de la interfaz de usuario

Para la interfaz de la aplicación se han realizado pruebas de tipo web. La interfaz de usuario se ha verificado mediante la realización de pruebas de contenido, esto es determinar si la interfaz cuenta con todos los elementos necesarios para su correcto funcionamiento. Estos elementos son datos que la aplicación muestra al usuario o entradas de datos que el usuario debe introducir. Asimismo, se comprueba que aquellos elementos de diseño están correctamente situados en la pantalla.

Se ha probado que todas las pantallas de la aplicación se ven como el desarrollador había programado. Después de corregir ciertos matices de diseño o de funcionalidad de los elementos dinámicos, los resultados obtenidos han sido muy satisfactorios.

Pruebas de regresión

Tras encontrar los posibles errores encontrados en las pruebas realizadas, éstos se han corregido y el código ha sido nuevamente probado para comprobar si se han resuelto satisfactoriamente dichos errores. Esto es lo que se denomina pruebas de regresión.

7.3 Validación

El objetivo del proceso de validación es comprobar que se satisfacen los requisitos definidos en el apartado 5.3.

7.3.1 Estrategia y desarrollo de validación

Para la validación de esta aplicación, se ha comprobado cada uno de los requisitos desde el punto de vista de un usuario final. Esta tarea ha sido realizada primero por el desarrollador y posteriormente en colaboración con individuos que no han participado en el proceso de desarrollo. Dado que el público final de este sistema no cuenta con conocimientos de informática, a estos individuos que han hecho las pruebas se les ha tenido que dar una breve descripción de lo que era **SchoolPipe** y de lo que abarcaba en cada uno de los roles disponibles.

Tras terminar la codificación de las entidades desarrolladas se han realizado una serie de pruebas con el fin de comprobar que la entidad desarrollada no tuviera errores.

En todas las pruebas realizadas se ha ido comprobando el correcto tratamiento de los datos a la base de datos del proyecto. Además, se ha testado, por ejemplo, que un usuario no tuviera acceso a datos de otros usuarios a los que no podía tener acceso. Véase:

- Un profesor intentaba ver datos de un alumno que no era suyo.
- Un padre intentaba ver datos de un hijo que no es el suyo.

En la tabla 16 del anexo E se muestran las pruebas realizadas, una breve explicación de las mismas, unas capturas que corroboran su funcionamiento

Como se ha especificado en el apartado 6.2, la aplicación es ejecutada en un Linux 14.04 con un apache como servidor.

Para comprobar los resultados en base de datos se utilizaba la herramienta phpMyAdmin. En el sistema, para depurar errores de código además de instalar el módulo *xDebug* para PHP5, el cual sirve para pintar en el navegador aquellos errores que suceden en la ejecución siempre y cuando la directiva *display_errors* de PHP estuviera activa. Esta directiva está activa en el entorno de desarrollo, no así en el de producción. Dichos errores también se pueden ver desde el fichero *schoolpipe-error.log* con el comando *tail -f /var/log/apache2/schoolpipe-error.log*.

Por último, cabe destacar que se ha realizado un ataque de vulnerabilidades por *SQL Injection* mediante la herramienta SQLMAP y un ataque de vulnerabilidades sobre el servidor, donde se encuentra alojada la web de **SchoolPipe**, utilizando la herramienta Nessus.

En el anexo F se muestra el primer informe obtenido en el ataque de *SQL Injection* con SQLMAP. Como se puede observar en toda la traza el nivel de robustez es elevado puesto que de sólo se han encontrado dos puntos de vulnerabilidad de un total de 742 analizados.

En el anexo F también se muestra el segundo informe que atiende a las vulnerabilidades del servidor donde solamente hay dos vulnerabilidades de nivel bajo de las 20 evaluadas.

Después de llevar a cabo una intensa auditoría sobre la aplicación web, se puede determinar que el informe tiene un nivel satisfactorio, ya que las medidas de seguridad son de nivel alto, y los tipos de datos alojados en la web (según la LOPD) son de un fichero de nivel bajo.

8 Extensibilidad

Como se ha comentado en varias secciones de este documento, **la base de la aplicación es replicable bajo otros sistemas**. Para demostrar esto se ha decidido generar una aplicación auxiliar, también en sistema web y bajo el sistema LAMP que se encarga de gestionar los invitados a una boda.

La aplicación también gestiona roles: los invitados por un lado y el administrador por otro.

Esta página tiene incorporada todas las operaciones que el administrador puede hacer en **SchoolPipe** como la generación de un menú específico para los invitados o gestionar todos los invitados desde la propia aplicación. Se han incluido módulos propios y específicos como son “Información del evento” o “lista de bodas”.

La figura 35 muestra una captura de la página principal de dicha aplicación, cuyo parecido es similar al de **SchoolPipe**, pero el diseño es bien diferente.



Figura 35. Captura de la página principal de la aplicación de gestión de invitados a una boda

Como ejemplo, con esta aplicación se podría realizar una web en la que se pudiera ver toda la información de un equipo de fútbol. En este caso no existiría un inicio de sesión previo, si no que la información relevante estaría en la propia página principal. El inicio de sesión podría diferenciar a un usuario normal de un administrador, donde éste último sí tendría acceso a los menús de configuración. Se podría implementar una entidad que gestionara un CMS (*Content Management System*) para poder publicar contenido (artículos, noticias, fotos) sin necesidad de tener que programar nuevas entidades específicas para este cometido.

Otra posible idea es la posibilidad de gestionar una tienda. Aquí, los roles son los propios empleados, el gerente, etc.

Algunas de las posibles operaciones que habría serían, además de gestionar todo el personal de la tienda, generar listados de productos con información de stock, contabilizar las ventas del local y mostrar gráficos con estos datos o algo un poco más complejo: se podría generar un web Service que contactara con los proveedores para hacer solicitud de material por falta de stock.

9.1 Evaluación

Este proyecto ha sido evaluado por varios usuarios. Algunos de ellos sí tenían conocimientos sobre el mundo web, en cambio otros eran prototipos de usuarios finales de la aplicación.

En ningún caso se ha presentado un manual de usuario puesto que la aplicación será adaptada específicamente para los centros educativos y serán ellos mismos los que, con las explicaciones razonables de los desarrolladores, generen el manual de usuario que entregarán a su personal y a las familias de los alumnos matriculados en dicho centro.

Los usuarios, en general, han comprobado toda la funcionalidad de la aplicación. Algunos han reportado errores, ya corregidos, que se achacaban más al diseño de la interfaz y al posicionamiento de algunas opciones que hacían la aplicación menos intuitiva de lo esperado en un primer momento.

En cuanto a la funcionalidad de las operaciones, los usuarios no han reportado muchos errores. Lo que más cabría destacar aquí es que algunos selectores de opciones de algunos formularios de edición de registros estaban vacíos, lo cual era rápidamente solucionable gracias a la configuración de tablas explicada con anterioridad.

De todas maneras, los usuarios que han probado la aplicación han destacado de manera muy positiva la funcionalidad de la administración que permite cambiar el comportamiento de los registros de las tablas de la base de datos. También han valorado positivamente algunos de los efectos CSS que se han implementado en la página principal y el funcionamiento que envuelven a los mensajes privados y los eventos.

9.2 Beneficios de la aplicación

Los beneficios que esta aplicación le dará a cualquier centro de enseñanza serán muchos de los comentados a lo largo de este documento. Supondrá una gran herramienta de ayuda para los centros educativos que la incluyan en su hábito de trabajo.

Uno de los beneficios más importantes y una de las razones por las que se decidió crear este sistema, es para que los alumnos de temprana edad no tengan la responsabilidad de informar en casa de lo que le hayan dicho en el colegio porque los padres pueden ser notificados directamente por sus profesores.

Dentro del centro escolar ayudará a que el personal esté mucho más organizado con la utilidad de calendario-agenda que incluye **SchoolPipe**. Además, se podrá gestionar toda la información de alumnos, actualizar cursos, grupos, asignaturas, etc.

Otro de los beneficios atiende al diseño: Bootstrap lo hace todo mucho más sencillo y proporciona unos diseños intuitivos y fácilmente moldeables.

Al margen de **SchoolPipe**, la versatilidad de la aplicación y que ésta sea replicable y adaptable a otras ideas similares es un beneficio a tener en cuenta en caso de tener una

idea que necesite una herramienta con funciones parecidas a las que ofrece **SchoolPipe** con gestión de roles de usuarios, módulos fácilmente instalables, configuración editable desde la propia aplicación, etc.

10 Conclusiones y líneas futuras

10.1 Conclusiones

En este trabajo se ha desarrollado un sistema integrado de gestión académica y de la comunicación para centros escolares. Una preocupación constante durante todo el proceso ha sido detectar cuáles son las necesidades que un centro escolar podría tener si desea gestionar su centro de manera informática o si desea comunicarse con la familia de determinado alumno saltando la opción de hacerlo mediante una llamada de teléfono o un correo electrónico.

Otra de las conclusiones a la que se ha llegado, después de la evaluación de algunos usuarios, es que la herramienta sería de gran utilidad en los centros educativos que la implementen. No obstante, estos usuarios han opinado que debería tener alguna opción más para complementar a la herramienta. Estos avances se ha propuesto que sean desarrollados en el futuro.

Las conclusiones finales del presente Proyecto Fin de Grado son las siguientes:

1. Se ha *desarrollado* una herramienta eficaz de comunicación entre un centro educativo y los familiares de un alumno.
2. Dicha herramienta incluye toda la *funcionalidad* propuesta en un primer momento.
3. Se ha realizado una *interfaz* para la herramienta que resulta intuitiva y manejable por cualquier usuario.
4. Se ha propuesto el *seguimiento* de una metodología software y unas pautas de mantenimiento que constituyen al éxito del proyecto.
5. Se ha priorizado el desarrollo de un software que fuera *reutilizable* y ampliable frente a la posibilidad de hacer un software de tipo *caja* (o *ad-hoc*)

10.2 Trabajo futuro

El trabajo que se espera realizar sobre la aplicación en el futuro es sobretodo mejorar y ampliar funcionalidades.

Se intentará que esta herramienta llegue a algún colegio que pueda probar el sistema en un entorno real para obtener un *feedback* real de lo que la aplicación necesita para continuar su desarrollo de cara a la implantación en centros educativos.

Se seguirá desarrollando la parte genérica de la aplicación implementando más funcionalidades para los campos de tipo *control* de la gestión de las tablas de base de datos. Se mejorarán propiedades de búsqueda de los listados de los registros de las tablas administradas. Esta sección será mucho más completa, mejorando también los formularios de inserción y edición de registros. Además, se podrán administrar tablas relacionadas. Por ejemplo: bajo un mismo listado poder administrar a la vez la tabla de usuarios y la tabla de datos de los usuarios. Ahora mismo lo que se puede hacer es administrar estas tablas por separado.

En la versión 2.0 de la aplicación aparecerán muchas más funcionalidades de la gestión académica. Se mejorará la gestión de exámenes no pudiendo introducir calificaciones a un examen que aún no ha sido realizado y quizá se implemente un algoritmo que permita a la herramienta generar horarios sin ayuda de software de terceros. Además, se incorporará la posibilidad de adjuntar archivos a los comunicados internos.

Además se podrán gestionar asignaturas de carácter extraescolar como por ejemplo actividades deportivas, idiomas extra o el sistema de comedor.

En una hipotética versión 3.0 se espera introducir un módulo financiero que permita a las familias hacer pagos online mediante pasarelas de banco o *PayPal*. Estos pagos podrán utilizarse para pagar cuotas mensuales de actividades extraescolares, excursiones o cualquier otro fin similar.

11 Referencias

- [1] J. M. Boneu, «Plataformas abiertas de e-learning para el soporte de contenidos educativos abiertos,» *Revista de la Universidad y Sociedad del Conocimiento*, p. 12, 2007.
- [2] PLS Ramboll Management, «Studies in the Context of the e-Learning Initiative: Virtual Models of European Universities,» Final Report to the EU Commission, DG Education and Culture, Aartus, 2004.
- [3] M. Molist, «Institutos y universidades apuestan por la plataforma libre de 'e-learning' Moodle,» *El País*, 13 Abril 2006.
- [4] S. Alive, «Sobre nosotros. School Alive,» Shool Alive, 2012. [En línea]. Available: <http://www.schoolalive.com/es/sobre-nosotros>. [Último acceso: 2015].
- [5] Bootstrap, «Bootstrap, a sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development.,» 2011. [En línea]. Available: <http://getbootstrap.com/>. [Último acceso: 2015].
- [6] J. Rodríguez Diéguez, O. Sáenz Barrio y otros, *Tecnología Educativa y Nuevas Tecnologías aplicadas a la educación*, Alcoy: MARfil, 1995.
- [7] Wikipedia, «Plataforma educativa - Wikipedia, la enciclopedia libre,» [En línea]. Available: http://es.wikipedia.org/wiki/Plataforma_educativa. [Último acceso: 2015].
- [8] Aula1, «¿Qué es un ERP educativo?,» 22 01 2015. [En línea]. Available: <http://www.aula1.com/blog/que-es-un-erp-educativo>. [Último acceso: 2015].
- [9] M. Telem, «The impact of the computerization of a high school's pedagogical administration on homeroom teacher–parents interrelations: A case study,» *Teaching and Teacher Education*, vol. 21, n° 6, pp. 661-678, 2005.
- [10] Cospa & Agilmic, «Alexia Educación,» 2009. [En línea]. Available: <http://www.alexiaeducacion.com/>. [Último acceso: 2015].

- [11] Cospa & Agilmic, «Alexia Educación,» 2009. [En línea]. Available: <http://www.alexiaeducacion.com/herramientas.php>. [Último acceso: 2015].
- [12] Detecsyst Soluciones TI, «Aula1,» 2008. [En línea]. Available: <http://www.aula1.com/>. [Último acceso: 2015].
- [13] Microsoft Corporation, 2010. [En línea]. Available: <http://azure.microsoft.com/es-es/>. [Último acceso: 2015].
- [14] Moodle, «Moodle,» [En línea]. Available: <http://moodle.org>. [Último acceso: 2015].
- [15] SAGE, «ERP para pymes que mejora la gestión de recursos,» [En línea]. Available: <http://www.sage.es/software/erp/mediana-empresa/sage-murano-erp>. [Último acceso: 2015].
- [16] Google Inc., «Google Apps for Work,» [En línea]. Available: <https://www.google.es/intx/es/work/apps/business/products/>. [Último acceso: 2015].
- [17] Microsoft, «Office en el ámbito educativo,» [En línea]. Available: <https://products.office.com/es-es/student/office-in-education?tab=schools>. [Último acceso: 2015].
- [18] LC Ibérica, «GlobalEduca,» [En línea]. Available: <http://www.lciberica.es/globaleduca>.
- [19] EduColombia, «Educolombia,» 2010. [En línea]. Available: <http://www.educolombia.org/>. [Último acceso: 2015].
- [20] GESCIS Technologies Pvt Ltd, [En línea]. Available: <http://www.web-school.in/>. [Último acceso: 2015].
- [21] Fedena, «Fedena,» 2009. [En línea]. Available: <http://www.fedena.com>. [Último acceso: 2015].

- [22] Wikipedia, «Fedena-Wikipedia,» [En línea]. Available: http://en.wikipedia.org/wiki/Fedena#cite_ref-4. [Último acceso: 2015].
- [23] Capita, «Capita SIMS,» 1984. [En línea]. Available: <http://www.capita-sims.co.uk/>. [Último acceso: 2015].
- [24] N. Mathiason, «Capita 'billed schools £75m too much',» *The Guardian (London)*, 06 12 2009.
- [25] J. M. Correa Gorospe, «La integración de plataformas de e-learning en la docencia universitaria: Enseñanza, aprendizaje e investigación con Moodle en la formación inicial del profesorado,» *Revista Latinoamericana de Tecnología Educativa*, vol. 4, nº 1, 2005.
- [26] Moodle, «Moodle,» [En línea]. Available: <https://moodle.net/sites/>. [Último acceso: 2015].
- [27] Blackboard Inc., «Blackboard Learn,» 1997. [En línea]. Available: <http://es.blackboard.com/sites/international/globalmaster/Platforms/Blackboard-Learn.html>. [Último acceso: 2015].
- [28] *This Will Revolutionize Education*. [Película]. Veritasium, 2014.
- [29] J. Marti, «Una visión crítica sobre la tecnología educativa,» 2013.
- [30] M. Telem, «Information technology's impact on school–parents and parents–student interrelations: a case study,» *Computers & Education*, vol. 47, nº 3, pp. 260-279, 2006.
- [31] E. Dans, «Educación online: plataformas educativas y el dilema de la apertura,» *Revista de la Universidad y Sociedad del Conocimiento (RUSC)*, 2009.
- [32] I. Jacobson, G. Booch y J. Rumbaugh, *El proceso unificado de desarrollo de software*, Reading: Addison Wesley, 2000.

- [33] J. Garzas, «Ciclos de vida para gestionar un proyecto software: cascada, espiral, iterativo, incremental o ágil 2013,» 2013. [En línea]. Available: <http://www.javiergarzas.com/2013/07/ciclos-de-vida-software.htm>.
- [34] M. Domínguez-Dorado, «NetBeans IDE 4.1. La alternativa a Eclipse,» *Todo Programación*, nº 13, pp. 32-34, 2005.
- [35] Openhost, «How to Choose Windows versus Linux Web Hosting,» [En línea]. Available: <http://www.openhost.co.nz/windows-or-linux>. [Último acceso: 2015].
- [36] Apache Software Foundation, «Apache Http Server Project,» 1997. [En línea]. Available: <http://httpd.apache.org/>. [Último acceso: 2015].
- [37] Netcraft, «April 2015 Web Server Survey,» 04 2015. [En línea]. Available: <http://news.netcraft.com/archives/2015/04/20/april-2015-web-server-survey.html>. [Último acceso: 05 2015].
- [38] «Mysql, The world's most popular open source database,» 1995. [En línea]. Available: <https://www.mysql.com/>. [Último acceso: 2015].
- [39] MySQL, «Top Reasons for VP Engineering to Embed MySQL,» [En línea]. Available: https://www.mysql.com/why-mysql/topreasons_vp.html. [Último acceso: 2015].
- [40] CiberAula, «Máster en LAMP,» 2006. [En línea]. Available: http://www.ciberaula.com/curso/lamp/que_es/.
- [41] phpMyAdmin, «phpMyAdmin,» [En línea]. Available: <http://www.phpmyadmin.net/>. [Último acceso: 2015].
- [42] Assembla, «Assembla Keeps Code, Tasks, and Teams Happily Together,» 2003. [En línea]. Available: <https://www.assembla.com>. [Último acceso: 2015].
- [43] The Apache Software Foundation, «Apache Subversion,» 2000. [En línea]. Available: <https://subversion.apache.org>.

- [44] Amazon Web Services, Inc., «Amazon Web Services (AWS) – Servicios de informática en la nube,» 2006. [En línea]. Available: <http://aws.amazon.com/es/about-aws/>. [Último acceso: 2015].
- [45] TIMETABLE WEB , «Horario escolar online,» 2012. [En línea]. Available: <http://www.timetableweb.com/es/index.php>. [Último acceso: 2015].
- [46] J. James Garret, «Adaptive Path,» 2005. [En línea]. Available: <http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications/>. [Último acceso: 2015].
- [47] The jQuery Foundation, «jQuery,» 2005. [En línea]. Available: <http://jquery.com/>. [Último acceso: 2015].
- [48] World Wide Web Consortium (W3C), «HTML & CSS,» [En línea]. Available: <http://www.w3.org/standards/webdesign/htmlcss#whatcss>. [Último acceso: 2015].
- [49] D. Cochran, «Twitter Bootstrap Web Development (1st edición),» de *Twitter Bootstrap Web Development How-To*, Packt Publishing, 2012, p. 100.
- [50] D. Gandy, «Font Awesome, the iconic font and CSS toolkit,» 2011. [En línea]. Available: <http://fontawesome.github.io/Font-Awesome/>. [Último acceso: 2015].
- [51] CKSource, «Ckeditor, The best web text editor for everyone,» [En línea]. Available: <http://ckeditor.com/>. [Último acceso: 2015].
- [52] J. Pillora, «Verify.js - A powerful, customizable, asynchronous form validation library,» 2012. [En línea]. Available: <http://verifyjs.com/>. [Último acceso: 2015].

Anexo B: Resumen de cómo trabaja el sistema de rejilla de Bootstrap en función de los dispositivos

	Extra small devices Phones (<768px)	Small devices Tablets (>=768px)	Medium devices Desktops (>=992px)	Large devices Desktops (>=1200px)
Grid behaviour	Horizontal at all times	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints
Container width	None (auto)	750px	970px	1170px
Class prefix	.col-xs-	.col-sm-	.col-md-	.col-lg-
Number of columns	12	12	12	12
Column width	Auto	~62px	~81px	~97px
Gutter width	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)
Nestable	Yes	Yes	Yes	Yes
Offsets	Yes	Yes	Yes	Yes
Column ordering	Yes	Yes	Yes	Yes

Figura 37. Clasificación de opciones de Bootstrap según el dispositivo

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-8									.col-md-4			
.col-md-4				.col-md-4				.col-md-4				
.col-md-6						.col-md-6						

Figura 38. Distribución de columnas en Bootstrap

Anexo C: Algoritmos diseñados en la aplicación y otras capturas de interés

En este anexo se presentan, en forma de ilustraciones, varios de los algoritmos comentados durante el capítulo 6.

Conexión a la base de datos

```
1 <?php
2 /**
3  * Clase Wrapper de Mysql
4  *
5  */
6  class database {
7
8      public static function connect( $server = DB_SERVER,
9                                     $username = DB_SERVER_USERNAME,
10                                    $password = DB_SERVER_PASSWORD,
11                                    $database_name = DB_DATABASE,
12                                    $db_connection = 'mysql') {
13
14          global $$db_connection, $db_tables;
15          if (isset ($$db_connection) == false || database::checkDbConnection($db_connection) == false) {
16              $$db_connection = new mysqli($server, $username, $password, $database_name);
17              if ($$db_connection->connect_errno) {
18                  throw new exception ('database::connect: error ' . $$db_connection->connect_errno);
19              }
20              if ($db_connection == "mysql") {
21                  $tables_query = $$db_connection->query("SHOW TABLES");
22                  if (is_object($tables_query) && get_class($tables_query) == "mysqli_result") {
23                      $db_tables = array();
24                      while ($tables_array = $tables_query->fetch_row()) {
25                          $db_tables[$tables_array[0]] = null;
26                      }
27                  }
28                  $tables_query->free_result();
29              }
30          }
31      }
32  }
33  ?>
```

Figura 39. Clase database -wrapper de Mysql

```
1 <?php
2 //Se incluye el fichero database
3 require_once DATABASE_FILE;
4 //Se conecta a la base de datos
5 database::connect();
```

Figura 40. Forma de conectar a la base de datos

Operaciones de Sesión y pre-cargado de datos GET/POST

```
1 <?php
2     /**
3     * Se carga la sesión en función de si el parametro sesion
4     * ya está en la sesión
5     * @return type
6     */
7     public static function loadSession() {
8
9         if (isset($_SESSION[Util::PARAM_SESSION])) {
10             self::$session = $_SESSION[Util::PARAM_SESSION];
11             return self::$session;
12         } else {
13             self::$session = BackSession::getInstance();
14             return self::$session;
15         }
16     }
```

Figura 41 Cargador de sesión

```
1 <?php
2     /**
3     * Cargamos la entity y el action
4     */
5     public function loadParams () {
6         $this->entity=Util::getEntityParam();
7         $this->action=Util::getActionParam();
8         $this->params=$_REQUEST;
9     }
```

Figura 42. Cargador de parámetros

Ejemplo de método Action y Render de la clase Login

```
1 <?php
2     /**
3     * acciones a realizar , updates , inserts , deletes...
4     */
5     public function action() {
6         $action = Util::getActionParam();
7         if ($action == 'validate') {
8             $request = Util::$session->getRequestParams();
9             $user = $request[self::PARAM_USER];
10            $password = $request[self::PARAM_PASSWORD];
11            $userRecord = $this->session->getDDBBBack()->validateUser($user, $password);
12            if ($userRecord === null) {
13                Util::addError(Util::__('Usuario/Password erroneo'));
14            } else {
15                $userRecord['userentities'] = $this->session->getDDBBBack()->getUserAccess($userRecord);
16                DBSchoolPipe::updateLogin($userRecord['back939_user_id']);
17                $this->session->setUserData($userRecord);
18                $this->session->setUser($userRecord['back939_user_user']);
19                $this->session->setMenu($this->session->getDDBBBack()->getUserMenu($userRecord['back939_roles_id']));
20
21                Util::redirect(Util::DEFAULT_PAGE);
22            }
23        }
24    }
25    /**
26    * metodo que obtiene lo que hay que pintar y llama a la plantilla html que lo hace
27    */
28    public function render() {
29        Util::printMessage();
30        $this->includeTemplate(self::TEMPLATE_LOGIN, get_class());
31    }
```

Figura 43. Método Action y Render de la clase Login

Distribución de carpetas

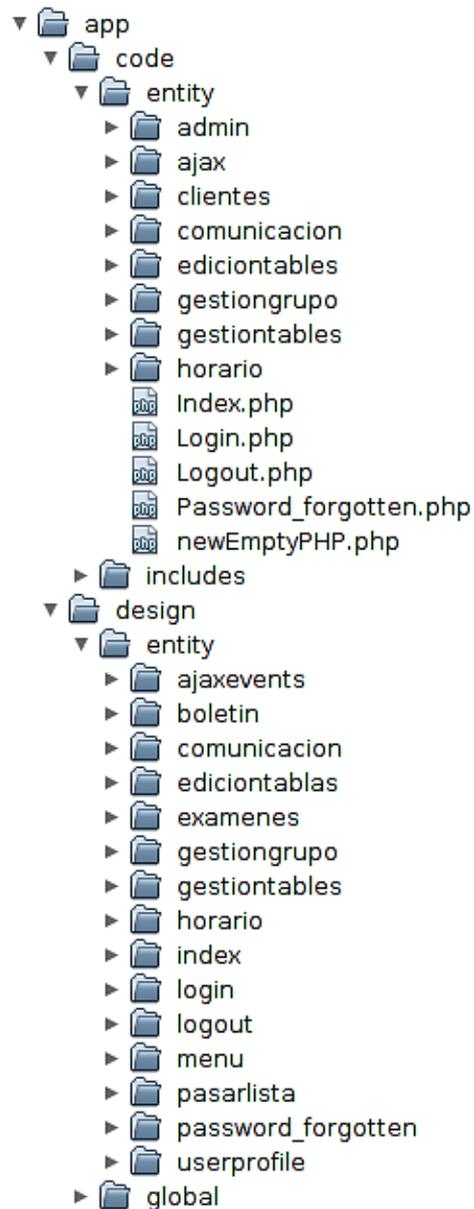


Figura 44. Disposición de carpetas

Template Context y uso de variables

```
1 <?php
2     class TemplateContext {
3
4         //Array de elementos.
5         private static $elements=array();
6
7         /**
8          * Añadir un elemento al array de elementos.
9          * @param type $clave
10         * @param type $valor
11         */
12        public static function addElement($clave,$valor){
13            self::$elements[$clave]=$valor;
14        }
15
16        /**
17         * Obtener un elemento del array de elementos.
18         * @param type $clave
19         * @return type
20         */
21        public static function getElement($clave){
22            if (isset(self::$elements[$clave])) {
23                return self::$elements[$clave];
24            } else {
25                false;
26            }
27        }
28    }
```

Figura 45. Clase TemplateContext

```
1 <?php
2     //Desde el método action o render se envían variables así
3     TemplateContext::addElement('tables', $tables);
4
5     //En el fichero plantilla se recogen las variables así
6     $variableRecibida = TemplateContext::getElement('tables');
```

Figura 46. Cómo enviar y recibir variables a las plantillas

Muestra de e-mail recibido



¡Hola **Mario!**

Tienes un nuevo mensaje de Marcos Muñoz Iglesias.

Pincha [aquí](#) para acceder a tu bandeja de entrada.

Este e-mail es automático, por favor, no respondas.

Si tienes cualquier duda o sugerencia contáctanos en info@schoolpipe.com

© Copyright 2015 SchoolPipe - Todos los derechos reservados

Figura 47. Muestra de correo recibido. Correo de aviso de mensaje nuevo en la aplicación.

Generación de PDF

```
1 <?php
2 require_once 'includes/wkhtmltopdf/wkhtml2pdf.amd64.php';
3 $pdfMaker = new WKHTML2PDF();
4 header("Content-type: application/pdf");
5 header("Content-Disposition: attachment; filename=\"\" . $_REQUEST["name"] . ".pdf\"");
6 ob_start();
7 ?>
8 <html>
9   <head>
10     <meta http-equiv="Pragma" content="no-cache">
11     <meta http-equiv="Expires" content="-1">
12     <title>Boletin de notas</title>
13     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" pageencoding="UTF-8">
14   </head>
15   <body>
16     <!--Cuerpo del PDF en HTML.-->
17     <!--Aquí va la información de las notas del alumno-->
18   </body>
19 </html>
20 <?php
21 $html = ob_get_clean();
22 $pdfMaker->set_html(utf8_encode($html));
23 $pdfMaker->render();
24 echo $pdfMaker->output(WKHTML2PDF::PDF_ASSTRING, "");
25 @ob_end_flush();
```

Figura 48. Ejemplo de cómo generar un PDF con WKHTML2PDF

Etiqueta: Admin	Nombre: admin	<input type="checkbox"/> Cambiar Roles	<input checked="" type="checkbox"/> Entidad	Menu	<input type="checkbox"/>	<input type="checkbox"/>
Etiqueta: Menu	Nombre: Menu	Icono: fa-bars	<input checked="" type="checkbox"/> Cambiar Herencia	<input checked="" type="checkbox"/> Cambiar Roles	<input checked="" type="checkbox"/> Entidad	<input checked="" type="checkbox"/> Menu
Etiqueta: Gestión de tablas	Nombre: GestionTablas	Icono: fa-database	<input checked="" type="checkbox"/> Cambiar Herencia	<input checked="" type="checkbox"/> Cambiar Roles	<input checked="" type="checkbox"/> Entidad	<input checked="" type="checkbox"/> GestionTablas
Etiqueta: Entidades	Nombre: Entidades	Icono: fa-file-o	<input checked="" type="checkbox"/> Cambiar Herencia	<input checked="" type="checkbox"/> Cambiar Roles	<input checked="" type="checkbox"/> Entidad	<input checked="" type="checkbox"/> EdicionTablas
Etiqueta: Asignaturas	Nombre: Materias	<input checked="" type="checkbox"/> Cambiar Roles	<input checked="" type="checkbox"/> Entidad	Menu	<input type="checkbox"/>	<input type="checkbox"/>
Etiqueta: Horario	Nombre: Horario	Icono: fa-calendar	<input checked="" type="checkbox"/> Cambiar Herencia	<input checked="" type="checkbox"/> Cambiar Roles	<input checked="" type="checkbox"/> Entidad	<input checked="" type="checkbox"/> Horario
Etiqueta: Comunicación	Nombre: Comunicación	<input checked="" type="checkbox"/> Cambiar Roles	<input checked="" type="checkbox"/> Entidad	Menu	<input type="checkbox"/>	<input type="checkbox"/>
Etiqueta: Mis Comunicados	Nombre: misComunicados	Icono: fa-comments	<input checked="" type="checkbox"/> Cambiar Herencia	<input checked="" type="checkbox"/> Cambiar Roles	<input checked="" type="checkbox"/> Entidad	<input checked="" type="checkbox"/> Comunicacion
Etiqueta: Tutorias	Nombre: Tutorias	<input checked="" type="checkbox"/> Cambiar Roles	<input checked="" type="checkbox"/> Entidad	Menu	<input type="checkbox"/>	<input type="checkbox"/>
Etiqueta: Mis alumnos	Nombre: Tutoria	Icono: fa-users	<input checked="" type="checkbox"/> Cambiar Herencia	<input checked="" type="checkbox"/> Cambiar Roles	<input checked="" type="checkbox"/> Entidad	<input checked="" type="checkbox"/> GestionGrupo
Etiqueta: Pasar Lista	Nombre: pasarlista	Icono: fa-list	<input checked="" type="checkbox"/> Cambiar Herencia	<input checked="" type="checkbox"/> Cambiar Roles	<input checked="" type="checkbox"/> Entidad	<input checked="" type="checkbox"/> PasarLista
Etiqueta: Exámenes	Nombre: Examen	Icono: fa-file	<input checked="" type="checkbox"/> Cambiar Herencia	<input checked="" type="checkbox"/> Cambiar Roles	<input checked="" type="checkbox"/> Entidad	<input checked="" type="checkbox"/> Exámenes
Etiqueta: Boletín	Nombre: Boletín	Icono: fa-star	<input checked="" type="checkbox"/> Cambiar Herencia	<input checked="" type="checkbox"/> Cambiar Roles	<input checked="" type="checkbox"/> Entidad	<input checked="" type="checkbox"/> Boletin

Figura 49. Representación de la pantalla que administra el menú

Anexo E: Desarrollo de pruebas atendiendo a los requisitos

En la tabla 16 se observa el listado de pruebas realizadas durante el proyecto indicando la salida esperada y la dificultad que se ha sufrido evaluada de 1-10 en dicha prueba. Cabe destacar que aquí se muestran sólo algunos requisitos. Seguidamente se muestran algunas capturas de las pruebas demostrando así que su resultado es satisfactorio.

Entrada/Requisito	Salida esperada	Posible error	Nivel de dificultad
Compatibilidad	La visualización de la aplicación se adapta correctamente a todos los navegadores.	Elementos visuales mal colocados. Errores de diseño.	4
Iniciar sesión en la aplicación	Acceso a la página principal del usuario que inicia sesión.	Mensaje erróneo en caso de introducir datos erróneos	1
Desarrollo de toda la gestión y edición de la administración de las tablas de la base de datos	La base de datos debe guardar correctamente todos los datos de las tablas configuradas. Esto es muy importante porque de esto dependen gran parte de los listados de datos, edición y creación de nuevos registros en las tablas administradas.	Datos no guardados correctamente.	10
Crear usuarios (parte I y parte II). Editar datos del usuario creado.	El usuario aparece con todos sus datos en las tablas de la base de datos cuando el administrador lo crea.	Datos no guardados correctamente.	10 ^(*)
Consultar datos de un usuario.	Se muestran los datos del usuario en modo formulario no editable.	Algún campo vacío implica que probablemente no se esté recogiendo bien de la base de datos. Esto también podría significar que ese campo no está relleno en la tabla.	4

Entrada/Requisito	Salida esperada	Posible error	Nivel de dificultad
Cambiar contraseña de acceso	Se modifica la contraseña en caso de meter la contraseña antigua de forma correcta.	Sale un mensaje de error en caso de que el campo <i>contraseña antigua</i> no coincida con la que de verdad está almacenada en la base de datos.	6
Ver horario asignado	Aparece el horario de un usuario leyendo datos de la tabla 15.	Aparece la tabla que contiene el horario vacía.	8.5
Pasar lista a los alumnos	Se almacenan en la base de datos las faltas de asistencia en caso de que se seleccione que un alumno ha faltado.	Datos no guardados correctamente.	7.5
Redacción de nuevos mensajes o contestación de mensajes ya existentes	Se almacenan en la base de datos los datos del mensaje como emisor, receptor/es, contenido del mensaje, hora de envío, etc. Además, se envía un correo a todos los receptores del mensaje.	Datos no guardados correctamente.	8
Visualizar panel de mensajes	Obtener de la base de datos los mensajes y distribuirlos por las diferentes bandejas.	Aun habiendo mensajes, las bandejas de mensajes aparecen vacías.	3
Borrar mensajes y que éstos aparezcan en la papelera.	El mensaje pasa de la bandeja donde se encuentra a la bandeja "Papelera" o de ésta última pasa a no salir en ninguna bandeja.	Los mensajes no son borrados, se mantienen en la bandeja correspondiente.	7 ^(**)
Crear exámenes o editar uno ya existente. Calificar dicho examen	Se almacenan en la base de datos los datos.	Los datos del examen o los de la calificación de los alumnos no se almacenan correctamente.	7

Entrada/Requisito	Salida esperada	Posible error	Nivel de dificultad
Descargar boletín de notas en PDF	Se descarga un fichero en formato PDF con las calificaciones del alumno.	Se descarga un fichero PDF pero su contenido está vacío.	4
Ver calendario	Se muestra el calendario con los eventos programados.	Se muestra el hueco del calendario vacío.	7
Crear eventos o editar alguno ya existente si el usuario es el creador	Los eventos se guardan en la base de datos. Además se envía un e-mail de notificación si hubiera invitados.	Datos no guardados correctamente.	9 ^(***)
Borrar eventos si el usuario es el creador	El evento ya no aparece en la base de datos.	El evento aún aparece en la base de datos.	3

Tabla 16. Listado de pruebas realizadas

(*) En esta prueba se agrupa también toda la gestión de listados, creación y edición de registros de las tablas administradas, como por ejemplo, la creación de usuarios.

(**) La gran dificultad en esta prueba fue el tratamiento de acciones Javascript.

(***) Esta prueba, por la diversidad de tipos de evento, la gran cantidad de datos a tratar, el control de envío de correos y que su funcionamiento se ha realizado bajo la tecnología de AJAX ha ocasionado que haya sido una de las más costosas de resolver. Se han invertido muchas horas de corrección de errores.

Ilustraciones de las pruebas

Prueba de compatibilidad de navegadores

School Pipe Intranet Bienvenido Marcos Muñoz Iglesias [Mi perfil](#)

School Pipe

Admin ▾ Asignaturas ▾ Comunicación ▾ Tutorías ▾ Institución ▾

Últimos Mensajes no leídos 2

De	Asunto	Fecha
Director Director Director	Mensaje3	21-03-2015
Martin Garrix	Mensaje4	21-03-2015

[Ver todos los mensajes](#)

Mi calendario

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

[Ver todos los eventos](#)

© Schoolpipe 2015

Figura 50. Página principal del rol Administrador en Google Chrome

Últimos Mensajes no leídos 0

De	Asunto	Fecha
----	--------	-------

[Ver todos los mensajes](#)

Horario

Horario personal de Alexandre

	Lunes	Martes	Miércoles	Jueves	Viernes
09:30	Libre	Religión Alexandre Sexto A	Religión Alexandre Segundo A	Libre	Libre
10:30	Libre	Religión Alexandre Quinto A	Libre	Libre	Libre
11:30	Descanso	Descanso	Descanso	Descanso	Descanso
12:00	Libre	Religión Alexandre Tercero A	Libre	Libre	Libre
13:00	Descanso	Descanso	Descanso	Descanso	Descanso
15:00	Libre	Libre	Libre	Religión Alexandre Cuarto A	Libre
15:45	Religión Alexandre Quinto B	Libre	Libre	Religión Alexandre Primero A	Libre

[Ver Horario a pantalla completa](#)

Mi calendario

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

[Ver todos los eventos](#)

© Schoolpipe 2015

Figura 51. Página principal del rol Profesor en Firefox



Figura 53. Página principal en Safari (iPhone)



Figura 52. Página principal en Firefox Mobile

Pruebas de inserción de calificaciones

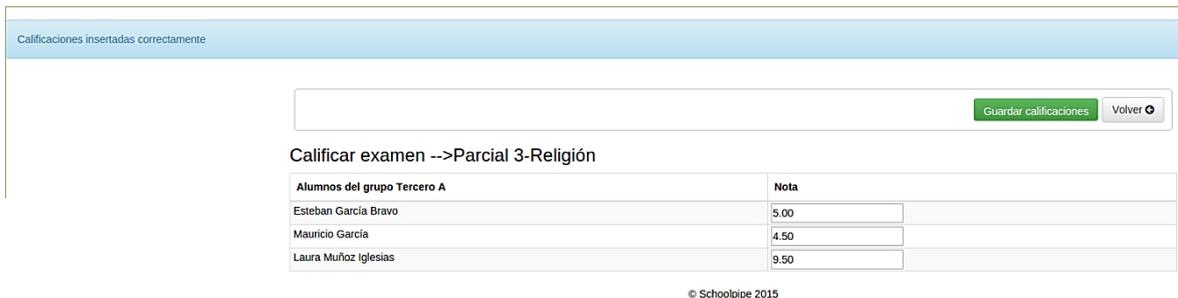


Figura 54. Pantalla que se muestra cuando las calificaciones han sido insertadas correctamente

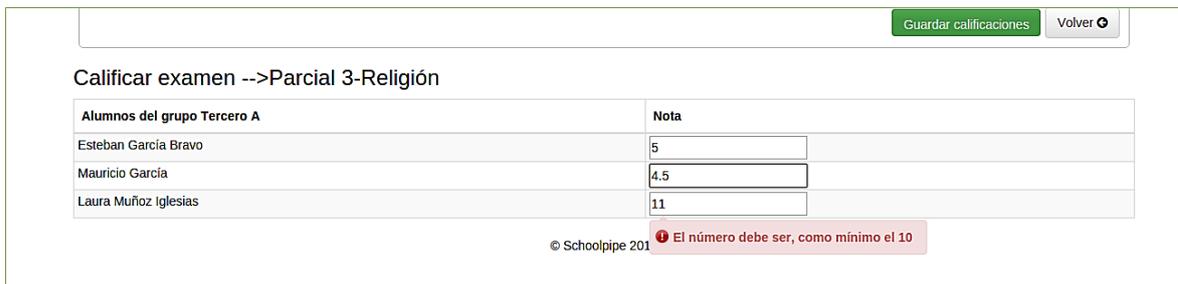


Figura 55. Si la calificación es mayor que 10, sale un error

Prueba de cambio de contraseña

Cambio de Contraseña

Contraseña antigua Contraseña nueva Repetir contraseña

© Schoolpipe 2015

Figura 56. Pantalla de cambio de contraseña

Tu contraseña antigua no coincide con la que has escrito...

Datos del perfil [Calendario](#) [Contraseña](#)

Cambio de Contraseña

Contraseña antigua Contraseña nueva Repetir contraseña

© Schoolpipe 2015

Figura 57. Cambio de contraseña incorrecto

Contraseña actualizada satisfactoriamente!

Datos del perfil [Calendario](#) [Contraseña](#)

Cambio de Contraseña

Contraseña antigua Contraseña nueva Repetir contraseña

© Schoolpipe 2015

Figura 58. Cambio de contraseña correcto

Prueba de muestra de datos

Admin ▾ Asignaturas ▾ Comunicación ▾ Tutorías ▾ Institución ▾

Datos del perfil [Calendario](#) [Contraseña](#)

Mis datos personales



Nombre <input type="text" value="Marcos"/>	Apellidos <input type="text" value="Muñoz Iglesias"/>	Nombre de Usuario <input type="text" value="admin"/>
--	---	--

[Cambia tu contraseña pinchando aquí](#)

Más datos

Género <input type="text" value="m"/>	Fecha de Nacimiento <input type="text" value="1991-01-08 00:00:00"/>	Edad <input type="text" value="24"/>
DNI <input type="text" value="03143923F"/>		

Contacto

Correo Electrónico <input type="text" value="ciltocruz@hotmail.com"/>	Teléfono1 <input type="text" value="650257175"/>	Teléfono2 <input type="text" value="6246360497"/>
---	--	---

Figura 59. Muestra de datos de usuario

Prueba de visión de horario de un grupo

Admin - Asignaturas - Comunicación - Tutorías - Institución

Volver

Horario Tercero A

	Lunes	Martes	Miércoles	Jueves	Viernes
09:30	Ciencias Sociales José Tercero A	Ciencias Naturales José Tercero A	Educación física Alejandro Tercero A	Lengua castellana y literatura José Tercero A	Lengua extranjera José Tercero A
10:30	Educación física Alejandro Tercero A	Matemáticas José Tercero A	Ciencias Naturales José Tercero A	Educación artística : plástica y música Lassana Tercero A	Lengua castellana y literatura José Tercero A
11:30	Descanso	Descanso	Descanso	Descanso	Descanso
12:00	Matemáticas José Tercero A	Valores Sociales y Cívicos Gregorio Religión Alexandre Tercero A	Lengua extranjera José Tercero A	Ciencias Naturales José Tercero A	Ciencias Sociales José Tercero A
13:00	Descanso	Descanso	Descanso	Descanso	Descanso
15:00	Educación artística : plástica y música Lassana Tercero A	Ciencias Sociales José Tercero A	Matemáticas José Tercero A	Educación física Alejandro Tercero A	Matemáticas José Tercero A
15:45	Lengua castellana y literatura José Tercero A	Lengua castellana y literatura José Tercero A	Lengua castellana y literatura José Tercero A	Lengua extranjera José Tercero A	Ciencias Naturales José Tercero A

Figura 60. Horario de un grupo de alumnos

Prueba de datos erróneos en un evento

Introduce un título

! Celda requerida

! Celda requerida

! Celda requerida ha sobre el icono c

Figura 61. Datos erróneo en el formulario que genera un evento

Prueba de Login y contraseña olvidada

Usuario/Password erroneo



The image shows a login form for 'School Pipe'. At the top left is the logo, which consists of a graduation cap icon and the text 'School Pipe'. Below the logo are two input fields: 'Email' with an envelope icon on the right, and 'Contraseña' with a magnifying glass icon on the right. Below these fields is an orange button labeled 'Acceder'. At the bottom of the form, there is a link that says '¿Has olvidado tu contraseña? Haz click aquí.'

© Schoolpipe 2015

Figura 62. Login erróneo

Contraseña modificada. Revise su bandeja de entrada.



The image shows a password reset form. At the top, it says 'Introduce tu e-mail'. Below this is a text input field. To the left of the input field is a button labeled 'Volver...' and to the right is a blue button labeled 'Enviar'.

© Schoolpipe 2015

Figura 63. Reenvío de la contraseña olvidada satisfactorio

Anexo F: Resultados de las pruebas de SQL Injection y de vulnerabilidades sobre servidor

Salida de texto de SQLMAP al realizar los ataques de SQL Injection

[23:53:06] [INFO] testing connection to the target URL

[23:53:06] [INFO] testing if the target URL is stable. This can take a couple of seconds

[23:53:07] [INFO] target URL is stable

[23:53:07] [INFO] testing if GET parameter 'entity' is dynamic

[23:53:07] [INFO] confirming that GET parameter 'entity' is dynamic

[23:53:08] [INFO] GET parameter 'entity' is dynamic

[23:53:08] [WARNING] reflective value(s) found and filtering out

[23:53:08] [ERROR] possible integer casting detected (e.g. "\$entity=intval(\$_REQUEST['entity'])") at the back-end web application

do you want to skip those kind of cases (and save scanning time)? [y/N] N

[23:53:22] [INFO] testing for SQL injection on GET parameter 'entity'

[23:53:22] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'

[23:53:24] [INFO] GET parameter 'entity' seems to be 'AND boolean-based blind - WHERE or HAVING clause' injectable

[23:53:26] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE or HAVING clause'

[23:53:26] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (EXTRACTVALUE)'

[23:53:27] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE or HAVING clause (UPDATEXML)'

[23:53:27] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'

[23:53:27] [INFO] testing 'MySQL >= 4.1 AND error-based - WHERE or HAVING clause'

[23:53:27] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'

[23:53:27] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause'

[23:53:27] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'

[23:53:28] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'

[23:53:28] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (UTL_INADDR.GET_HOST_ADDRESS)'

[23:53:28] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (CTXSYS.DRITHSX.SN)'

[23:53:28] [INFO] testing 'Firebird AND error-based - WHERE or HAVING clause'

[23:53:28] [INFO] testing 'MySQL >= 5.1 error-based - PROCEDURE ANALYSE (EXTRACTVALUE)'

[23:53:28] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace'

[23:53:29] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (EXTRACTVALUE)'

[23:53:29] [INFO] testing 'MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)'

[23:53:29] [INFO] testing 'MySQL >= 5.5 error-based - Parameter replace (BIGINT UNSIGNED)'

[23:53:29] [INFO] testing 'PostgreSQL error-based - Parameter replace'

[23:53:29] [INFO] testing 'Microsoft SQL Server/Sybase error-based - Parameter replace'

[23:53:29] [INFO] testing 'Microsoft SQL Server/Sybase error-based - Parameter replace (integer column)'

[23:53:29] [INFO] testing 'Oracle error-based - Parameter replace'

[23:53:30] [INFO] testing 'Firebird error-based - Parameter replace'

[23:53:30] [INFO] testing 'MySQL inline queries'

[23:53:30] [INFO] testing 'PostgreSQL inline queries'

[23:53:30] [INFO] testing 'Microsoft SQL Server/Sybase inline queries'

[23:53:30] [INFO] testing 'Oracle inline queries'

[23:53:30] [INFO] testing 'SQLite inline queries'

[23:53:30] [INFO] testing 'Firebird inline queries'

[23:53:31] [INFO] testing 'MySQL > 5.0.11 stacked queries'

[23:53:31] [INFO] testing 'PostgreSQL > 8.1 stacked queries'

[23:53:31] [INFO] testing 'PostgreSQL < 8.2 stacked queries (Glibc)'

[23:53:31] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries'

[23:53:31] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE)'

[23:53:31] [INFO] testing 'Oracle stacked queries (DBMS_LOCK.SLEEP)'

[23:53:32] [INFO] testing 'Oracle stacked queries (USER_LOCK.SLEEP)'

[23:53:32] [INFO] testing 'HSQLDB >= 1.7.2 stacked queries'

[23:53:32] [INFO] testing 'HSQLDB >= 2.0 stacked queries'

[23:53:32] [INFO] testing 'MySQL > 5.0.11 AND time-based blind (SELECT)'

[23:53:42] [INFO] GET parameter 'entity' seems to be 'MySQL > 5.0.11 AND time-based blind (SELECT)' injectable

[23:53:42] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'

[23:53:42] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found

sqlmap got a 302 redirect to 'http://www.schoolpipe.com:80/index.php'. Do you want to follow? [Y/n] Y

[23:53:53] [INFO] target URL appears to be UNION injectable with 8 columns

injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n] Y

[23:54:07] [WARNING] if UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--dbms=mysql')

[23:54:07] [INFO] testing 'MySQL UNION query (98) - 22 to 40 columns'

[23:54:10] [INFO] target URL appears to be UNION injectable with 39 columns

[23:54:21] [INFO] testing 'MySQL UNION query (98) - 42 to 60 columns'

[23:54:24] [INFO] testing 'MySQL UNION query (98) - 62 to 80 columns'

[23:54:26] [INFO] testing 'MySQL UNION query (98) - 82 to 100 columns'

[23:54:29] [INFO] target URL appears to be UNION injectable with 84 columns

[23:54:54] [INFO] testing 'Generic UNION query (98) - 1 to 20 columns'

[23:54:57] [INFO] testing 'Generic UNION query (98) - 22 to 40 columns'

[23:55:00] [INFO] target URL appears to be UNION injectable with 40 columns

[23:55:12] [INFO] testing 'Generic UNION query (98) - 42 to 60 columns'

[23:55:15] [INFO] target URL appears to be UNION injectable with 53 columns

[23:55:30] [INFO] testing 'Generic UNION query (98) - 62 to 80 columns'

[23:55:33] [INFO] testing 'Generic UNION query (98) - 82 to 100 columns'

[23:55:35] [INFO] checking if the injection point on GET parameter 'entity' is a false positive
GET parameter 'entity' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
N

sqlmap identified the following injection points with a total of 742 HTTP(s) requests:

Parameter: entity (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: entity=1 AND 9050=9050

Type: AND/OR time-based blind

Title: MySQL > 5.0.11 AND time-based blind (SELECT)

Payload: entity=1 AND (SELECT * FROM (SELECT(SLEEP(5)))OCxi)

[23:57:59] [INFO] the back-end DBMS is MySQL

web server operating system: Linux Ubuntu

web application technology: Apache 2.4.7

back-end DBMS: MySQL 5.0.11

Salida del ataque de vulnerabilidades del servidor realizado por Nessus

52.16.221.129					
Summary					
Critical	High	Medium	Low	Info	Total
0	0	0	2	18	20
Details					
Severity	Plugin Id	Name			
Low (2.6)	70658	SSH Server CBC Mode Ciphers Enabled			
Low (2.6)	71049	SSH Weak MAC Algorithms Enabled			
Info	10107	HTTP Server Type and Version			
Info	10267	SSH Server Type and Version Information			
Info	10287	Traceroute Information			
Info	10881	SSH Protocol Versions Supported			
Info	11219	Nessus SYN scanner			
Info	11936	OS Identification			
Info	12053	Host Fully Qualified Domain Name (FQDN) Resolution			
Info	19506	Nessus Scan Information			
Info	22964	Service Detection			
Info	24260	HyperText Transfer Protocol (HTTP) Information			
Info	25220	TCP/IP Timestamps Supported			
Info	39520	Backported Security Patch Detection (SSH)			
Info	39521	Backported Security Patch Detection (WWW)			
Info	43111	HTTP Methods Allowed (per directory)			
Info	45590	Common Platform Enumeration (CPE)			
Info	54615	Device Type			
Info	59861	Remote web server screenshot			
Info	70657	SSH Algorithms and Languages Supported			

Figura 64. Resultado de ataque de vulnerabilidad del servidor

