



UNIVERSIDAD
POLITECNICA
DE VALENCIA

Escuela Técnica
Superior de Ingeniería
Informática



PROYECTO FINAL DE CARRERA

DESARROLLO DE UN SITIO WEB PARA UN COLEGIO

CEIP Montealegre de L'Eliana

Autor:

Víctor Pérez Pérez

Director:

Felix Buendía García

ÍNDICE

<u>1. INTRODUCCIÓN</u>	4
1.1 Presentación y objetivos	4
1.2 Contexto	4
1.3 Planteamiento del problema	4
1.4 Estructura del documento	5
<u>2. ESPECIFICACIÓN DE REQUISITOS</u>	6
2.1 Introducción	6
2.1.1 Propósito	6
2.1.2 Ámbito	6
2.1.3 Definiciones, acrónimos y abreviaturas	6
2.1.4 Referencias	6
2.1.5 Visión global	7
2.2 Descripción general	7
2.2.1 Perspectiva del producto	7
2.2.2 Funciones del producto	7
2.2.3 Características del usuario	9
2.2.4 Restricciones generales	9
2.2.5 Supuestos y dependencias	9
2.3 Requisitos específicos	10
2.3.1 Requerimientos funcionales	10
2.3.2 Requerimientos de interfaces externos	16
2.3.2.1 Interfaces de usuario	16
2.3.2.2 Interfaces hardware	16
2.3.2.3 Interfaces software	16
2.3.2.4 Interfaces de comunicaciones	17
2.3.3 Requerimientos de eficiencia	17
2.3.4 Obligaciones del diseño	17
2.3.4.1 Estándares cumplidos	17
2.3.4.2 Limitaciones hardware	17
2.3.5 Atributos	17
2.3.5.1 Seguridad	17
2.3.5.2 Facilidades de mantenimiento	17
2.3.5.3 Portabilidad	18
2.3.5.4 Otros requerimientos	18
<u>3. ANÁLISIS</u>	19
3.1 Introducción	19
3.2 Diagrama de clases	19
3.3 Diagrama de casos de uso	20
3.3.1 Actores	21
3.3.2 Casos de uso del usuario	21

3.3.3 Casos de uso del usuario registrado	22
3.3.4 Casos de uso del alumno	22
3.3.5 Casos de uso del profesor	22
3.3.6 Casos de uso del administrador	23
<u>4. DISEÑO</u>	24
4.1 Introducción	24
4.2 Capa de presentación	24
4.3 Capa de negocio o lógica de la aplicación	26
4.4 Capa de persistencia o datos	27
<u>5. IMPLEMENTACIÓN</u>	29
5.1 Tecnologías utilizadas en el desarrollo del proyecto	29
5.1.1 HTML y XHTML	29
5.1.2 CSS	29
5.1.3 JavaScript, jQuery y AJAX	30
5.1.4 PHP	30
5.1.5 SQL	31
5.1.6 MySQL	31
5.1.7 PhpMyAdmin	31
5.1.8 StarUML	32
5.1.9 Adobe Dreamweaver	32
5.1.10 DBDesigner	32
5.2 Descripción del proyecto	32
5.2.1 Capa de presentación	32
5.2.2 Capa de negocio o lógica de la aplicación	36
5.2.3 Capa de persistencia o datos	40
<u>6. EVALUACIÓN</u>	41
6.1 Introducción	41
6.2 Validación de CSS	41
6.3 Validación de enlaces	42
6.4 Validación de la resolución	42
6.5 Validación de navegadores	42
6.6 Validación de la seguridad en el acceso a la zona de administración	45
<u>7. CONCLUSIÓN</u>	46
7.1 Valoración personal del trabajo realizado	46
7.2 Posibles ampliaciones	46
<u>8. BIBLIOGRAFÍA</u>	47

ÍNDICE DE FIGURAS

Figura 1. Zonas de la interfaz gráfica	16
Figura 2. Diagrama de clases	20
Figura 3. Actores	21
Figura 4. Casos de uso del usuario	21
Figura 5. Casos de uso del usuario registrado	22
Figura 6. Casos de uso del alumno	22
Figura 7. Casos de uso del profesor	22
Figura 8. Casos de uso del administrador	23
Figura 9. Esquema básico de la arquitectura de tres capas	24
Figura 10. Formato de las páginas	25
Figura 11. Diagrama de navegabilidad de la página de Inicio	25
Figura 12. Capas del proyecto	26
Figura 13. Diagrama entidad-relación	28
Figura 14. Tablas de la base de datos	40
Figura 15. Validación de CSS	41
Figura 16. Validación de enlaces	42
Figura 17. Validación de navegadores (Microsoft Internet Explorer)	43
Figura 18. Validación de navegadores (Mozilla Firefox)	43
Figura 19. Validación de navegadores (Google Chrome)	44
Figura 20. Validación de navegadores (Safari)	44
Figura 21. Validación de navegadores (Opera)	45

1. INTRODUCCIÓN

1.1 Presentación y objetivos

Este documento describe el trabajo realizado en el proyecto final de carrera de ingeniería técnica en informática de gestión. El proyecto consiste en el desarrollo del sitio web del colegio público Monte alegre, situado en la localidad de L'Eliaana (Camp de Túria, Valencia). El objetivo de dicha web es dar una información completa sobre el colegio y los servicios de los que dispone, así como proporcionar una serie de funcionalidades a los alumnos y profesores del centro educativo.

La web es accesible desde cualquier navegador por Internet y tiene información referente a su localización, sus miembros (alumnos y profesores) y su proyecto educativo. Se trata de dar una visión lo más completa posible de la institución y de su funcionamiento.

En cuanto a los usuarios, la página tiene varios tipos. Por un lado, están los usuarios no registrados, que únicamente pueden acceder a información general sobre el centro. Por otro, están los usuarios registrados, que pueden acceder a una mayor funcionalidad dentro de la aplicación. Entre estos últimos se encuentra el usuario alumno/padre, el usuario profesor y el usuario administrador. Los usuarios alumno/padre y profesor pueden realizar acciones tales como listar los alumnos o los profesores del colegio, mientras que el usuario administrador gestiona la base de datos del sistema.

1.2 Contexto

El proyecto ha sido realizado para el colegio de educación infantil y primaria Montealegre, situado en la localidad de L'Eliaana.

Para el desarrollo de la aplicación realicé diversas reuniones con la directora del colegio y estudié cómo se estaban realizando las tareas para después poder informatizarlas adecuadamente. La mayoría de las decisiones de estética y funcionales de la aplicación fueron tomadas junto con la directora, que además me proporcionó toda la documentación y fotografías del colegio.

Una vez realizadas las reuniones con la directora y estudiada la situación, distribuí el trabajo en 3 partes: el diseño de la web, la intranet de los usuarios alumno y profesor y la del administrador.

1.3 Planteamiento del problema

El problema que se nos ha planteado es la construcción del sitio web del colegio. La funcionalidad de la aplicación, a grandes rasgos, debía ser:

- Mostrar información general del centro.
- Permitir a los alumnos realizar consultas sobre sus asignaturas y profesores.
- Permitir a los profesores realizar consultas sobre sus alumnos y gestionar las notas y faltas de asistencia de estos.
- Permitir al administrador de la web gestionar la base de datos del sistema.

1.4 Estructura del documento

El presente documento está dividido en una serie de capítulos que corresponden, básicamente, a las distintas etapas que conforman el proceso de desarrollo del proyecto. Estas etapas han sido:

- **Especificación de requisitos:** Se redactó de una manera global una primera visión del proyecto donde señalamos los requisitos que debía cumplir. La finalidad de esta etapa es plasmar el acuerdo entre el desarrollador y el cliente acerca de las funcionalidades del proyecto. En nuestro caso el visto bueno nos lo dió la directora del centro educativo.
- **Análisis:** Se realizó el modelado conceptual de la futura solución mediante el uso de diagramas (diagrama de clases y diagramas de casos de uso). Los modelos ayudan a visualizar como es el sistema, proporcionando plantillas que sirven de guía en la construcción de la aplicación. En esta etapa se especifica qué debe hacer la aplicación pero no cómo debe hacerlo.
- **Diseño:** Se utilizaron los elementos y modelos obtenidos durante el análisis para transformarlos en mecanismos que puedan ser utilizados en un entorno web con las características y condiciones que establecen este tipo de entornos. Se diseñaron todos los niveles de los que consta la aplicación (nivel de presentación, nivel lógico y nivel de persistencia).

Tanto la etapa del análisis como la del diseño están desprovistas de código. Un buen análisis y un buen diseño son la mejor forma de llegar a producir software de calidad.

- **Implementación:** Se utilizaron los elementos obtenidos en el diseño para permitir la elaboración del producto o prototipo funcional, es decir, que puede ser puesto en marcha y sometido a pruebas. Para ello se consideraron las diversas tecnologías que han intervenido en la elaboración de dicho producto. Todo lo desarrollado en las etapas del análisis y del diseño, se tradujo a código.
- **Evaluación y pruebas:** Esta fase se centró en la comprobación del correcto funcionamiento del producto desarrollado mediante una serie de pruebas.

Tras estas tareas pudimos dar por concluido el proyecto, por lo que en último lugar mostramos las conclusiones obtenidas y listamos la bibliografía utilizada durante la realización del mismo.

2. ESPECIFICACIÓN DE REQUISITOS

2.1 Introducción

2.1.1 Propósito

El propósito de la especificación de requisitos es definir cuales son los requerimientos que debe tener la aplicación que se va a desarrollar y describir la funcionalidad del usuario a lo largo de ella.

2.1.2 Ámbito

El desarrollo del sitio web está orientado a ofrecer diversos contenidos y funcionalidades que ayuden a poder obtener información sobre los servicios de los que dispone el colegio al usuario anónimo y a obtener información sobre alumnos y profesores al usuario registrado. En la aplicación se diferencian, por tanto, dos partes bastante claras: la pública y la privada. La parte pública (portal) será accesible por todo el mundo y tendrá información general sobre el colegio. La parte privada (intranet) será para uso exclusivo de usuarios registrados y contendrá información más específica.

2.1.3 Definiciones, siglas y abreviaturas

Sitio web: Conjunto de archivos electrónicos y páginas web referentes a un tema en particular que incluye una página inicial de bienvenida, con un nombre de dominio y dirección en Internet específicos.

Interfaz: Parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario.

Intranet: Parte privada de la aplicación donde sólo tendrán acceso a la información los usuarios que estén registrados.

Navegador: Permite al usuario recuperar y visualizar páginas web a través de Internet.

Servidor web: Se trata de un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

Centro: Sitios donde se imparten los niveles de educación primaria.

Etapas educativas: Cada uno de los tramos en los que se estructura el sistema educativo español.

Proyecto curricular: Documento donde se indican los objetivos, estrategias y contenidos en la práctica docente.

Curso: Unidad de impartición de la enseñanza.

2.1.4 Referencias

- ANSI/IEEE Std. 830 – 1984. Guía del IEEE para la especificación de requerimientos software.

- Una guía para la realización y supervisión de proyectos final de carrera en el ámbito de la web.
- Apuntes de las asignaturas ISG.
- Ejemplos de otros proyectos.

2.1.5 Visión global

A continuación se realizará la descripción general del sistema desarrollado con sus funciones, características del usuario, restricciones, supuestos y dependencias. También se expondrá una especificación detallada de los requisitos detectados.

2.2 Descripción general

2.2.1 Perspectiva del producto

La aplicación desarrollada pretende dar información general sobre el colegio así como información más específica para cada tipo de usuario que esté registrado. Se podrá acceder a la aplicación desde cualquier sistema operativo que tenga conexión a Internet utilizando un navegador web.

2.2.2 Funciones del producto

A continuación se muestran las funciones que conforman la aplicación, según el tipo de usuario que se encuentre conectado.

Usuario anónimo

- **Autenticación:** Se muestra un par de cajas de texto en las que el usuario ha de introducir su nombre de usuario y contraseña para acceder a la parte privada de la aplicación.

Usuario registrado

- **Cerrar sesión:** Se muestra un botón para que el usuario cierre su sesión. Cuando se pulsa se le redirige a la página de inicio.

Usuario alumno/padre

- **Consultar horario:** Se muestra un listado de las asignaturas en las que el alumno conectado se encuentra matriculado y cuando haga clic sobre una de ellas se muestra el horario correspondiente a esa asignatura (día de la semana, hora de inicio y hora de fin).

- **Listar alumnos de clase:** Se listan los alumnos que comparten clase con el alumno conectado (nombre y apellidos).

- **Listar profesores:** Se listarán los profesores que imparten clase al alumno conectado (nombre, apellidos y nombre de la asignatura que le imparten).

- **Listar notas:** Se muestra un listado de las asignaturas en las que el alumno conectado se encuentra matriculado y cuando haga clic sobre una de ellas, se muestran las calificaciones correspondientes a esa asignatura (trimestre y nota).

- **Listar faltas de asistencia:** Se muestra un listado de las asignaturas en las que el alumno conectado se encuentra matriculado y cuando haga clic sobre una de ellas se muestran las faltas de asistencia correspondientes a esa asignatura (fecha y si está o no justificada).

Usuario profesor

- **Listar alumnos:** Se muestra el listado de las asignaturas que imparte el profesor conectado y cuando selecciona una de ellas se muestran los alumnos que se encuentran matriculados en esa asignatura (nombre y apellidos).

- **Listar profesores:** Se listan los profesores que imparten clases en el centro (nombre y apellidos).

- **Poner notas:** Se muestra el listado de las asignaturas que imparte el profesor conectado para que cuando seleccione una de ellas pueda elegir un alumno de los que se encuentran matriculados en esa asignatura y ponerle una nota.

- **Poner faltas de asistencia:** Se muestra el listado de las asignaturas que imparte el profesor conectado para que cuando seleccione una de ellas pueda elegir un alumno de los que se encuentran matriculados en esa asignatura y ponerle una falta de asistencia.

- **Modificar notas:** Se muestra el listado de las asignaturas que imparte el profesor conectado para que cuando seleccione una de ellas pueda elegir un alumno de los que se encuentran matriculados en esa asignatura y modificarle la nota.

- **Modificar faltas de asistencia:** Se muestra el listado de las asignaturas que imparte el profesor conectado para que cuando seleccione una de ellas pueda elegir un alumno de los que se encuentran matriculados en esa asignatura y modificar una falta de asistencia.

- **Listar faltas de asistencia:** Se muestra el listado de las asignaturas que imparte el profesor conectado para que cuando seleccione una de ellas aparezca el listado de alumnos que se encuentran matriculados en esa asignatura, con su nombre, apellidos y número de faltas de asistencia.

Usuario administrador

- **Alta de usuario:** Se podrá dar de alta usuarios.

- **Alta de asignatura:** Se podrá dar de alta asignaturas.

- **Matricular alumno en asignatura:** Se podrá matricular a los alumnos en las distintas asignaturas.

- **Baja de usuario:** Se podrá dar de baja un usuario.

- **Baja de asignatura:** Se podrá dar de baja una asignatura.

- **Modificar datos usuario:** Se podrá modificar los datos de un usuario.
- **Modificar datos asignatura:** Se podrá modificar los datos de un asignatura.

2.2.3 Características del usuario

Se puede diferenciar entre dos tipos de usuarios, los usuarios no registrados y los usuarios registrados. Dentro de los usuarios registrados podemos diferenciar, además, más tipos de usuarios.

Usuarios no registrados

Este tipo de usuarios solamente tendrán acceso a la información general del centro.

Usuarios registrados

Este tipo de usuarios son los que tienen acceso a la intranet. Hay distintos tipos de usuarios registrados según la función que tengan que realizar. Se puede diferenciar el usuario alumno/padre, el usuario profesor y el usuario administrador.

Usuario alumno/padre

Este tipo de usuario podrá consultar el horario de clase, con qué otros alumnos comparte clase, qué profesores le imparten docencia, qué notas ha obtenido y qué faltas de asistencia tiene.

Usuario profesor

Este tipo de usuario podrá listar los alumnos a los que imparte clase, listar los profesores que imparten clase en el centro y gestionar las notas y faltas de asistencia de sus alumnos.

Usuario administrador

Este tipo de usuario se encargará de la gestión de la base de datos del sistema. Es decir, efectuará el alta y la baja de los usuarios y asignaturas así como las modificaciones sobre la información referente a ellos/as.

2.2.4 Restricciones

Al tratarse de una aplicación web, se requiere un ordenador con un navegador convencional y una conexión a Internet básica.

2.2.5 Supuestos y dependencias

La aplicación desarrollada trabaja al margen de cualquier hardware o software ofreciendo así un soporte multiplataforma. La única dependencia importante que podemos encontrar está relacionada con el servidor web donde se encuentre alojado nuestro portal, el cual ha de ser capaz de soportar PHP y MySQL.

Menos importante pero a tener en cuenta es el hecho de que el diseño de la interfaz de la aplicación ha sido realizada sobre el navegador Firefox, de manera que algunos aspectos como los bordes redondeados de los divs y que la posición de algunos botones salga en la posición correcta dependen de si el usuario utilizar este navegador para visualizar la web o no.

2.3 Requisitos específicos

2.3.1. Requerimientos funcionales

A continuación se describen las diversas funciones que ofrece la aplicación web, clasificadas según el tipo de usuario que accede al servicio.

Usuarios no registrados

La única función disponible para los usuarios no registrados es la de autenticarse para acceder a la parte privada de la aplicación.

Apartado	Descripción
Título	Autenticación
Propósito	Acceder a la intranet de la aplicación
Entrada	Nombre de usuario y contraseña
Proceso	Se comprueba que el usuario existe
Salida	Confirmación de acceso a la intranet o aviso de que el nombre de usuario o contraseña no es/son válidos

Usuarios registrados

Los usuarios registrados en general sólo pueden cerrar su sesión para abandonar la intranet y volver de nuevo a la parte pública de la aplicación.

Apartado	Descripción
Título	Cerrar sesión
Propósito	Salir de la intranet de la aplicación
Entrada	-
Proceso	Se redirige a la página de inicio
Salida	-

Usuario alumno/padre

Sin embargo, si el usuario registrado es un usuario de tipo alumno/padre puede consultar su horario de clase, listar los alumnos con los que comparte clase, listar los profesores que le imparten docencia y consultar sus notas y faltas de asistencia.

Apartado	Descripción
Título	Consultar horario
Propósito	Consultar el horario de clase
Entrada	-
Proceso	Listar las asignaturas en las que se encuentra matriculado el alumno conectado Elegir la asignatura de la cual quiere ver el horario
Salida	El horario de clase (día, hora de inicio y hora de fin)
Apartado	Descripción
Título	Listar alumnos de clase
Propósito	Listar los alumnos con los que comparte clase
Entrada	-
Proceso	Recuperar de la base de datos los alumnos con los que comparte clase el alumno conectado
Salida	El nombre y los apellidos de los alumnos con los que comparte clase

Apartado	Descripción
Título	Listar profesores
Propósito	Listar los profesores que le imparten clase
Entrada	-
Proceso	Recuperar de la base de datos los profesores que imparten clase al alumno conectado y el nombre de la asignatura que le imparten
Salida	El nombre y los apellidos de los profesores y los nombres de las asignaturas

Apartado	Descripción
Título	Listar notas
Propósito	Listar las notas
Entrada	-
Proceso	Listar las asignaturas en las que se encuentra matriculado el alumno conectado

	Elegir la asignatura de la cual quiere ver la nota
Salida	Las notas

Apartado	Descripción
Título	Listar faltas de asistencia
Propósito	Listar las faltas de asistencia
Entrada	-
Proceso	Listar las asignaturas en las que se encuentra matriculado el alumno conectado Elegir la asignatura de la cual quiere ver las faltas de asistencia que pudiera tener
Salida	Las faltas de asistencia

Usuario profesor

Si el usuario registrado es del tipo profesor entonces puede listar los alumnos a los que imparte clase, listar los profesores que dan clase en el colegio y poner y modificar las notas y faltas de asistencia de sus alumnos.

Apartado	Descripción
Título	Listar alumnos
Propósito	Listar los alumnos a los que imparte clase
Entrada	-
Proceso	Listar las asignaturas en las que imparte clase el profesor conectado Elegir la asignatura de la cual quiere ver los alumnos que se encuentran matriculados
Salida	El nombre y los apellidos de los alumnos a los que imparte clase

Apartado	Descripción
Título	Listar profesores
Propósito	Listar los profesores que imparten clase en el centro
Entrada	-
Proceso	Recuperar de la base de datos los profesores que imparten clase en el colegio
Salida	El nombre y los apellidos de los profesores que imparten clase en el centro

Apartado	Descripción
Título	Poner notas
Propósito	Poner la nota de una asignatura a un alumno
Entrada	Nota
Proceso	Elegir la asignatura Elegir el alumno y el trimestre Insertar la nota en la base de datos
Salida	Confirmación de que se ha insertado la nota o aviso de que no se ha podido realizar la operación

Apartado	Descripción
Título	Poner faltas de asistencia
Propósito	Poner una falta de asistencia a un alumno
Entrada	-
Proceso	Elegir la asignatura Elegir el alumno Insertar en la base de datos la falta de asistencia
Salida	Confirmación de que se ha insertado la falta de asistencia o aviso de que no se ha podido realizar la operación

Apartado	Descripción
Título	Modificar notas
Propósito	Modificar la nota de una asignatura a un alumno
Entrada	Nota
Proceso	Elegir la asignatura Elegir el alumno y el trimestre Modificar la nota en la base de datos
Salida	Confirmación de que se ha modificado la nota o aviso de que no se ha podido realizar la operación

Apartado	Descripción
Título	Modificar faltas de asistencia
Propósito	Modificar una falta de asistencia a un alumno
Entrada	Sí o No
Proceso	Elegir la asignatura Elegir el alumno Modificar en la base de datos la falta de asistencia

Salida	Confirmación de que se ha modificado la falta de asistencia o aviso de que no se ha podido realizar la operación
--------	--

Apartado	Descripción
Título	Listar faltas de asistencia
Propósito	Listar las falta de asistencia de los alumnos a los que imparte clase el profesor conectado
Entrada	-
Proceso	Elegir la asignatura Se muestra el listado de alumnos (nombre, apellidos y número de faltas de asistencia)
Salida	-

Usuario administrador

El usuario registrado de tipo administrador dispone de las funciones de dar el alta y la baja de los usuarios (alumno y profesor) y de las asignaturas, de matricular a un alumno en una asignatura y de modificar los datos de los usuarios y asignaturas.

Apartado	Descripción
Título	Alta de usuario
Propósito	Dar de alta un usuario
Entrada	Login, contraseña, nombre, apellidos y nivel del nuevo usuario
Proceso	Insertar el usuario en la base de datos.
Salida	Confirmación de que se ha insertado el usuario o aviso de que no se ha podido realizar la operación

Apartado	Descripción
Título	Alta de asignatura
Propósito	Dar de alta una asignatura
Entrada	Nombre de la asignatura, nivel en el que se encuentra y profesor que la imparte
Proceso	Insertar la asignatura en la base de datos
Salida	Confirmación de que se ha insertado la asignatura o aviso de que no se ha podido realizar la operación
Apartado	Descripción

Título	Matricular alumno en asignatura
Propósito	Matricular a un alumno en una asignatura
Entrada	Nombre y apellidos del alumno y nombre de la asignatura de la cual se le quiere matricular
Proceso	Inserta la información en la base de datos
Salida	Confirmación de que se ha matriculado al alumno en la asignatura seleccionada o aviso de que no se ha podido realizar la operación

Apartado	Descripción
Título	Baja de usuario
Propósito	Dar de baja un usuario
Entrada	-
Proceso	Elegir el usuario que se quiere dar de baja Borrar el usuario de la base de datos
Salida	Confirmación de que se ha borrado el usuario o aviso de que no se ha podido realizar la operación

Apartado	Descripción
Título	Modificar datos de un usuario
Propósito	Modificar los datos de un usuario
Entrada	Login, contraseña, nombre, apellidos y nivel
Proceso	Elegir un usuario Modificar los datos del usuario seleccionado con los datos introducidos
Salida	Confirmación de que se han modificado los datos del usuario o aviso de que no se ha podido realizar la operación

Apartado	Descripción
Título	Modificar datos asignatura
Propósito	Modificar los datos de una asignatura
Entrada	Nombre, nivel y profesor
Proceso	Elegir una asignatura Modificar los datos de la asignatura seleccionada con los datos introducidos
Salida	Confirmación de que se han modificado los datos de la asignatura o aviso de que no se ha podido realizar la operación

Apartado	Descripción
----------	-------------

Título	Baja de asignatura
Propósito	Dar de baja una asignatura
Entrada	-
Proceso	Elegir la asignatura que se quiere dar de baja Borrar la asignatura de la base de datos
Salida	Confirmación de que se ha borrado la asignatura o aviso de que no se ha podido realizar la operación

2.3.2 Requerimientos de interfaces externos

2.3.2.1 Interfaces de usuario

A continuación se muestra una captura de pantalla (figura 1) en la que se especifican las zonas que conforman la interfaz gráfica con la que interactuará el usuario que haga uso del portal web.

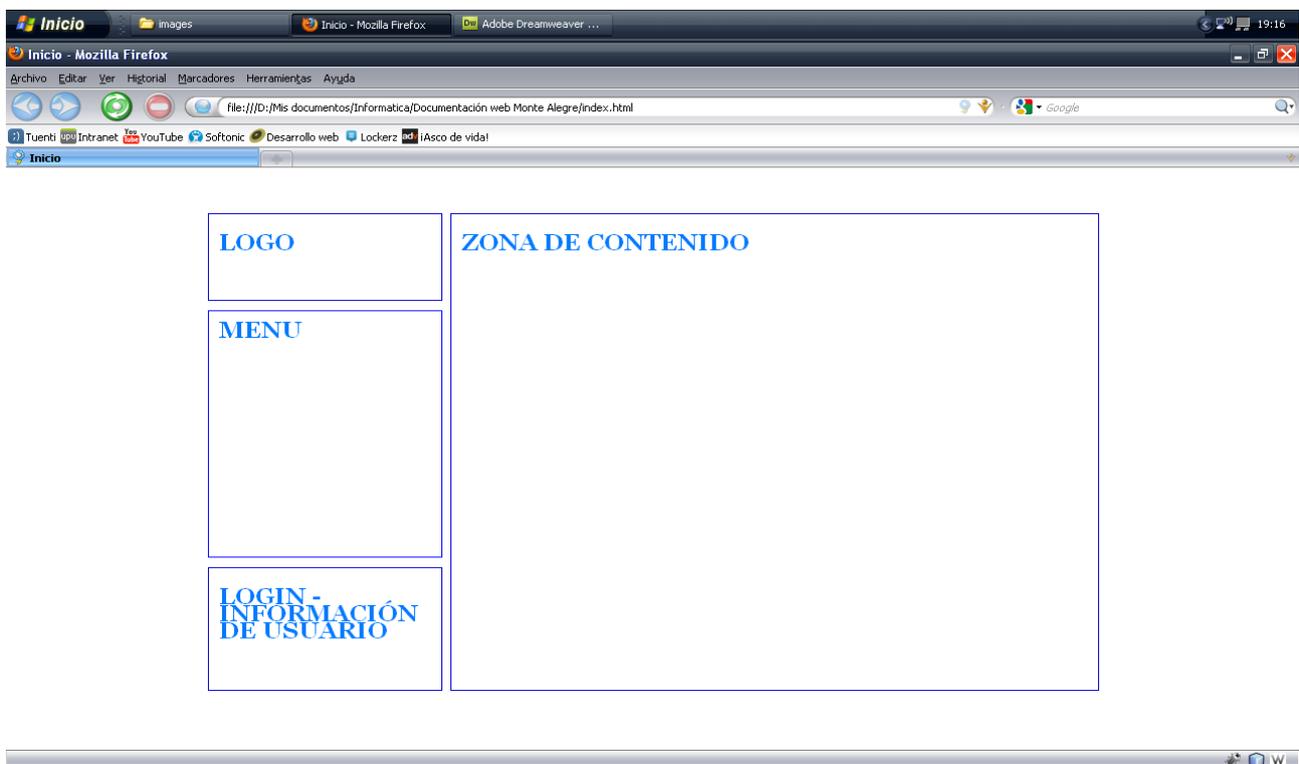


Figura 1. Zonas de la interfaz gráfica

2.3.2.2 Interfaces hardware

Al tratarse de una aplicación web, se podrá visualizar sobre cualquier sistema operativo.

2.3.2.3 Interfaces software

La aplicación funcionará en cualquier máquina con un navegador web y conexión a Internet.

2.3.2.4 Interfaces de comunicaciones

Las comunicaciones se efectuarán siguiendo el protocolo HTTP mediante conexiones TCP/IP.

2.3.3 Requerimientos de eficiencia

Debido a la metodología empleada en el diseño de la aplicación, el portal web tiene un rendimiento del 100% independientemente del hardware y software utilizado. Los únicos factores que pueden influir son las velocidades de conexión del cliente y del servidor. En cuanto al servidor, actualmente la mayoría cuentan con velocidades de subida más que suficientes para nuestras necesidades. Respecto al cliente, las velocidades ofrecidas actualmente por los ISP también son más que suficientes para la correcta transmisión de datos entre el cliente y el servidor.

Otro aspecto que hemos tratado de cuidar especialmente es la resolución del monitor. La mejor resolución para visualizar nuestra aplicación es 1280x800, pudiéndose visualizar, sin embargo, el contenido de igual modo con resoluciones inferiores.

2.3.4 Obligaciones del diseño

2.3.4.1 Estándares cumplidos

Se han intentado cumplir los estándares de cualquier web con acceso seguro, creando un sistema de autenticación para que nadie pueda acceder a una zona de la web a la que no tiene permiso de acceso. El idioma elegido para la presentación de las páginas ha sido el castellano.

2.3.4.2 Limitaciones hardware

Al tratarse de una aplicación web no se requiere un hardware específico. El servidor que albergará la base de datos del sistema deberá permanecer conectado a Internet las 24 horas, puesto que este host será quien atienda las peticiones de lectura y escritura de los usuarios que accedan a la intranet.

2.3.5 Atributos

2.3.5.1 Seguridad

La seguridad es un componente fundamental en el portal. La administración del sitio web está sujeta a la identificación satisfactoria del administrador en la aplicación, de forma que ningún usuario anónimo o visitante pueda, por ejemplo, añadir o eliminar usuarios en la base de datos. Para asegurar la identidad del administrador se requerirá un nombre de usuario y contraseña que autenticará a éste en la aplicación. La información acerca de la cuenta se guardará en la base de datos. El proceso de login o autenticación llevará al administrador al panel de administración desde donde podrá gestionar el sitio web del centro.

2.3.5.2 Facilidades de mantenimiento

El mantenimiento básico de la aplicación se llevará a cabo por el administrador. Sin embargo, cualquier cambio que se deseara introducir y requiriese de la modificación de la base de datos, así como implementar nueva funcionalidad, deberá ser llevada a cabo por el programador de la web.

2.3.5.3 Portabilidad

La aplicación ha sido diseñada con tecnología libre, luego podrá ser soportada por cualquier plataforma y sistema operativo. Por lo mismo, se podrá acceder a ella desde cualquier navegador.

2.3.5.4 Otros requerimientos

El portal usa una base de datos MySQL donde se almacena toda la información referida al centro como usuarios, servicios, etc. Las consultas a la base de datos se realizan por parte del servidor web mediante PHP y su API de acceso a bases de datos MySQL.

Por otro lado, ya hemos comentado que para poder introducir datos en la base de datos se debe ser el usuario administrador. Así pues, la primera vez que entremos a la aplicación se nos ha de haber creado un usuario de este tipo, puesto que si no es así no podremos crear nuevos usuarios, entre otras cosas. La aplicación se entregará, por tanto, con este usuario ya creado.

3. ANÁLISIS

3.1 Introducción

Para realizar el análisis de esta aplicación web se ha optado por seguir las recomendaciones definidas por UML (Unified Modeling Language o Lenguaje Unificado de Modelado en castellano). Este modelo dispone de multitud de diagramas que ayudan a comprender la complejidad del futuro sistema, permitiéndonos plasmar en un lenguaje estándar aquellas funcionalidades, requisitos y demás características que hemos detectado en el sistema.

Aunque UML define una gran cantidad de diagramas para representar los distintos aspectos del desarrollo de la aplicación, en nuestro caso nos hemos centrado en dos tipos de diagramas: el diagrama de clases y el diagrama de casos de usos.

3.2 Diagrama de clases

El diagrama de clases (figura 2) describe la estructura del sistema mostrando las relaciones (asociación, agregación y composición) entre los distintos elementos que conforman la aplicación, expresados estos como clases con sus atributos. A continuación describiremos brevemente dichas clases:

Colegio: Representa al colegio público Monte Alegre.

Alumno: Representa a los alumnos matriculados en el centro. Pueden acceder a la intranet de la aplicación a través de su nombre de usuario o login y contraseña.

Profesor: Representa a los profesores que imparten docencia en el centro. Pueden acceder a la intranet de la aplicación a través de su login y contraseña.

Administrador: Representa al administrador del sistema. Puede acceder a la intranet de la aplicación a través de su login y contraseña.. Es el encargado de la gestión de la base de datos.

Nivel: Contendrá la información necesaria para identificar a los distintos grupos de alumnos que conforman el colegio. El nivel puede ser infantil o primaria y dependiendo de este el curso puede ser 3, 4 o 5 años para los de infantil o 1º, 2º, 3º, 4º, 5º o 6º para los de primaria.

Asignatura: Representa las materias que se imparten en el centro y a las que asisten los alumnos y que son impartidas por los profesores del centro.

Horario: Contiene la información acerca del horario en el que se imparte una asignatura.

Nota: Representa la nota de un alumno en una asignatura.

Falta de asistencia: Permite identificar las ausencias de los alumnos en horario escolar, ya sean sin justificar o justificadas. De esta forma se podrá llevar un control de las asistencias de los alumnos a clase y que podrá ser consultado por sus padres o tutores.

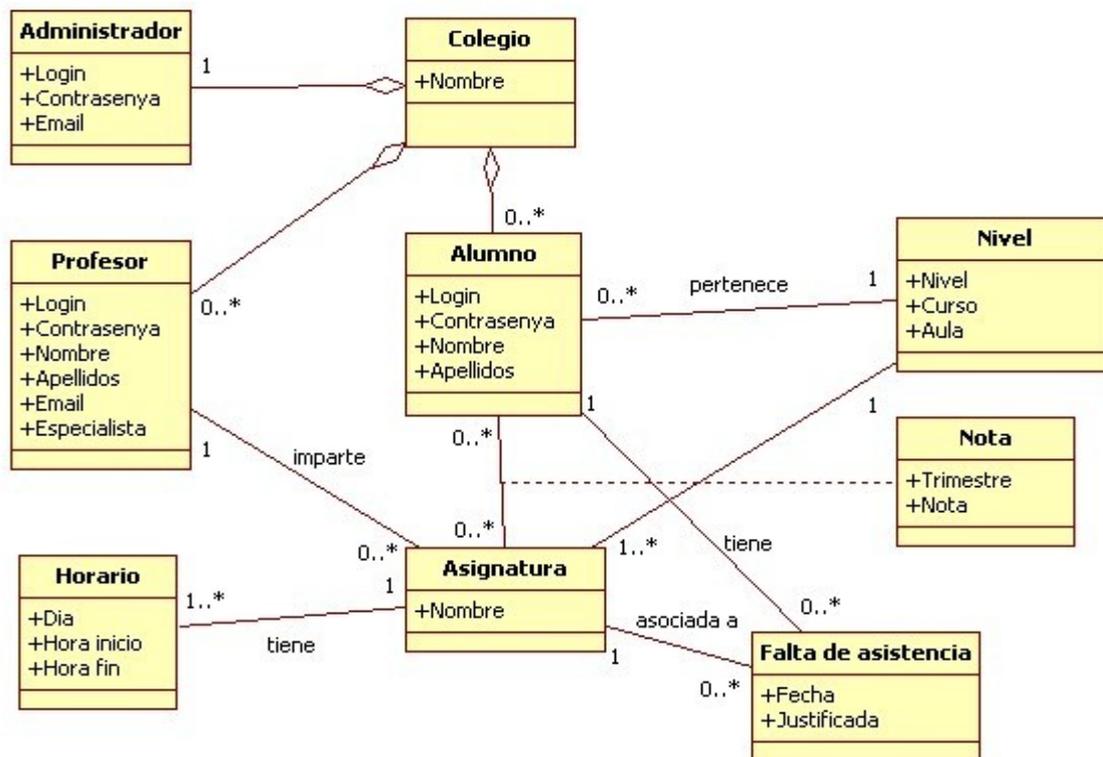


Figura 2. Diagrama de clases

3.3 Diagrama de casos de uso

Los diagramas de casos de uso nos permiten diferenciar los actores que interactúan con nuestra aplicación, las relaciones entre ellos y las acciones que puede realizar cada uno dentro del sistema. Este tipo de diagramas son fácilmente comprensibles tanto por clientes como por usuarios, representan los requisitos funcionales del sistema y se utilizan como base para un desarrollo iterativo e incremental. Los diagramas de casos de uso tienen tres elementos:

- **Actores:** Son los usuarios del sistema (figura 3). Un actor puede ser una persona, un conjunto de personas, un sistema hardware o un sistema software. Los actores representan un rol, que puede desempeñar alguien que necesita intercambiar información con el sistema.
- **Casos de uso:** Un caso de uso describe una forma concreta de utilizar parte de la funcionalidad de un sistema. La colección de todos los casos de uso describe toda la funcionalidad del sistema.
- **Comunicación entre actores y casos de uso:** Cada actor ejecuta un número específico de casos de uso en la aplicación. Por eso decimos que hay comunicación entre actores y casos de uso.

3.3.1 Actores

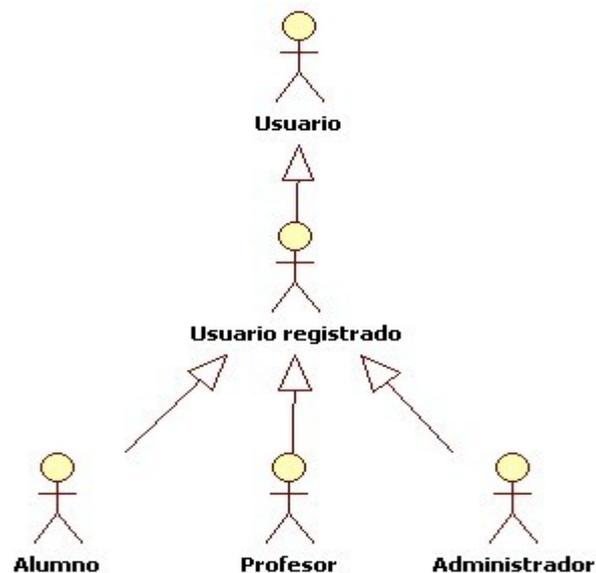


Figura 3. Actores

Los usuarios son todas aquellas personas que acceden al portal web a través de Internet sin necesidad de autenticarse. Los usuarios registrados son los alumnos y profesores del centro educativo y el administrador, que acceden a la intranet de la aplicación mediante un nombre de usuario y contraseña.

3.3.2 Casos de uso del usuario

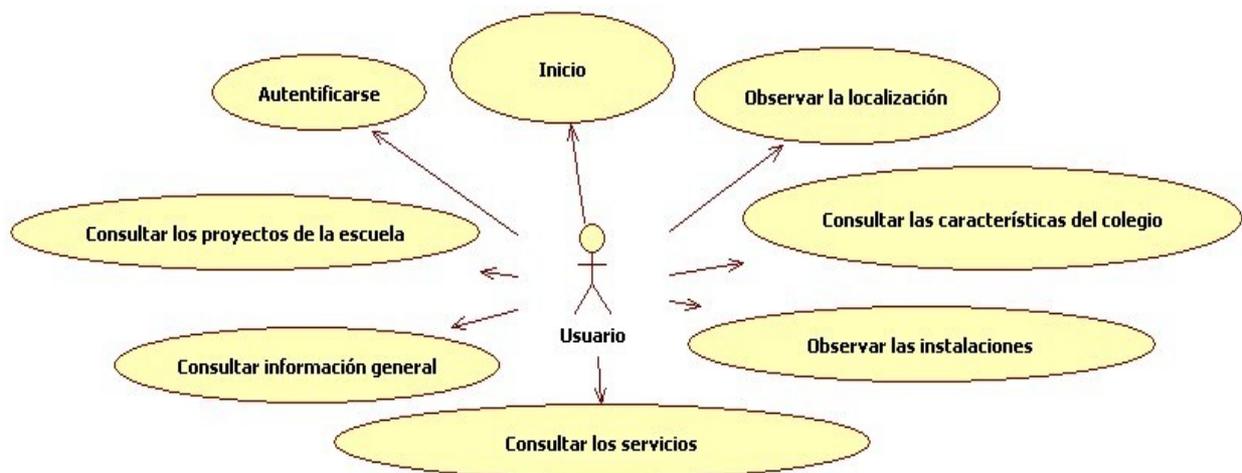


Figura 4. Casos de uso del usuario

Dado que este tipo de actor, que también podría denominarse usuario anónimo o visitante, no se ha identificado ante el sistema, las acciones que podrá realizar serán muy limitadas, centrándose en la obtención de información general del centro. Sin embargo, puede acceder a la zona de autenticación para autenticarse y entrar en la zona privada de la aplicación, accediendo, por tanto, a una mayor funcionalidad, si es que dispone de un nombre de usuario y contraseña válidos.

3.3.3 Casos de uso del usuario registrado

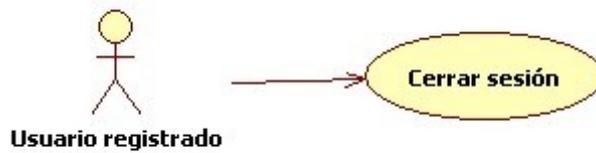


Figura 5. Casos de uso del usuario registrado

Este tipo de usuario sólo puede realizar la acción de cerrar su sesión porque es la única funcionalidad que tienen en común los usuarios alumno, profesor y administrador.

3.3.4 Casos de uso del alumno



Figura 6. Casos de uso del alumno

Este tipo de usuario representa a un alumno matriculado en el colegio. Podrá realizar las acciones de mostrar su horario de clases, listar los alumnos con los que comparte clase, listar los profesores que le imparten clase y consultar sus notas y faltas de asistencia.

3.3.5 Casos de uso del profesor

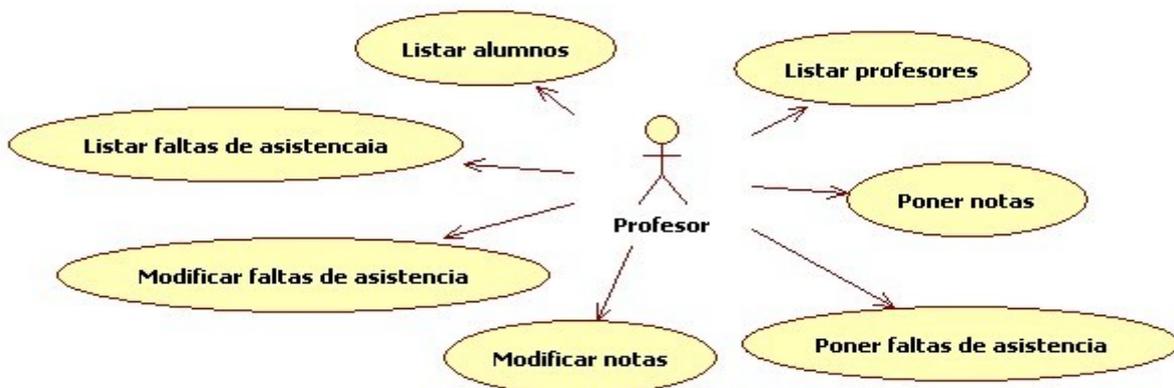


Figura 7. Casos de uso del profesor

Este tipo de usuario representa a un profesor que imparte clases en el colegio. Podrá realizar las acciones de listar los alumnos matriculados en el centro, listar los profesores que trabajan en el colegio, poner, modificar y listar las notas a los alumnos a los que imparte clase y poner y modificar faltas de asistencia a los alumnos que falten a alguna de sus clases.

3.3.6 Casos de uso del administrador



Figura 8. Casos de uso del administrador

Este tipo de usuario representa al administrador de la aplicación. Podrá realizar las acciones de dar el alta y baja de usuarios, matricular un alumno en una asignatura y modificar los datos de los usuarios y asignaturas.

4. DISEÑO

4.1 Introducción

El diseño es el proceso que extiende, refina y reorganiza los aspectos detectados en el proceso de modelado conceptual para generar una especificación rigurosa del sistema de información siempre orientada a la obtención de la solución del sistema software.

Para el diseño del sitio web se ha escogido una arquitectura de tres capas (presentación, aplicación y persistencia). La utilización de esta arquitectura (figura 9) se debe a que los distintos niveles son independientes unos de otros de manera que, por ejemplo, se puede cambiar fácilmente el comportamiento de las clases en el nivel de aplicación sin que ello influya en las otras capas.

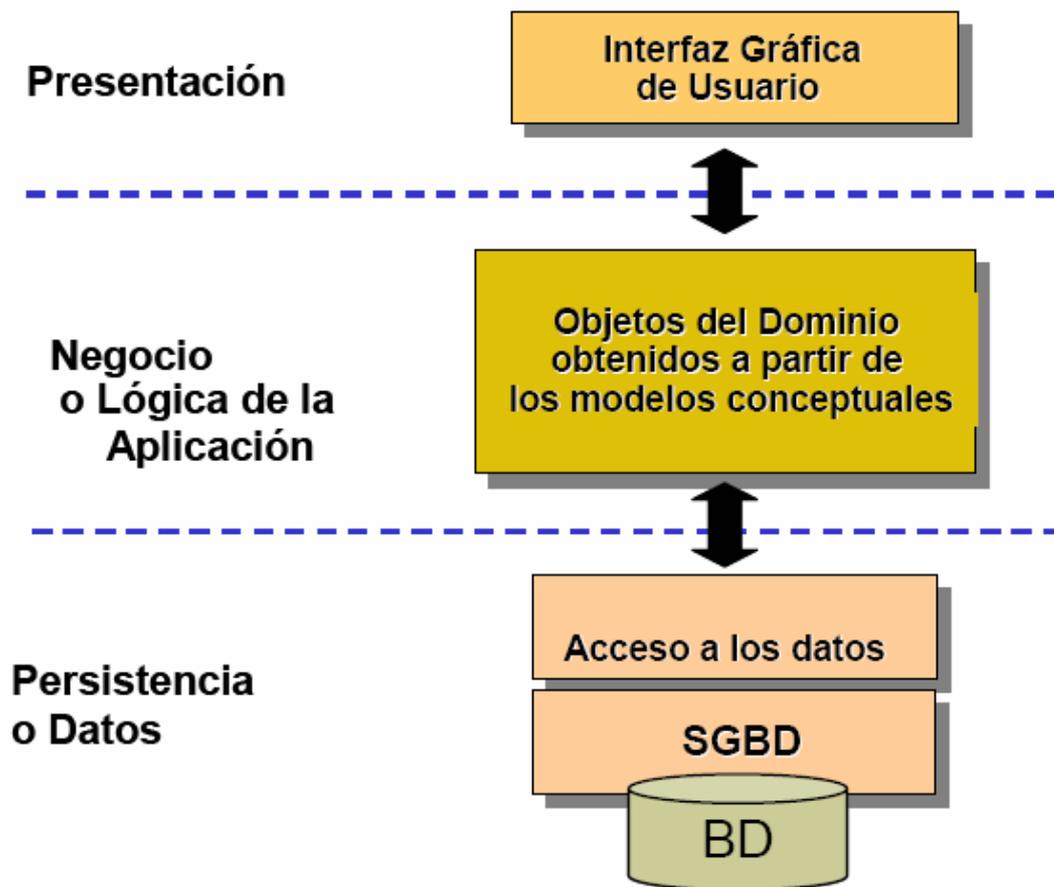


Figura 9. Esquema básico de la arquitectura de tres capas

4.2 Capa de presentación

La capa de presentación se puede definir como el conjunto de componentes software que implementan la interacción con los usuarios a través de una representación visual de la aplicación, proporcionando a los usuarios una forma de acceder y controlar los datos y los servicios de los objetos. A partir de la interfaz gráfica, el usuario podrá navegar por las distintas páginas para poder obtener toda la información que desee, o aportarla en caso de ser necesario.

A continuación se muestra el formato que se ha utilizado para las distintas páginas web (figura 10) y el diagrama de navegabilidad de la página de Inicio (figura 11).

ZONA DEL LOGO



Características generales

Código del centro: 46024503	CIF: Q4601134B
Dirección: C/José Alegre Rubio, nº 11	CP: 46183
Teléfono: 962743609	Móvil: 638494644
Fax: 962758780	Email: 46024503@edu.gva.es

Modalidad lingüística:
PIP (Programa de Incorporación Progresiva)

Educación infantil - 2º ciclo:
3 unidades

Educación primaria:
6 unidades

Rasgos de identidad

- o Pluralismo
- o Democracia
- o Coeducación
- o Aconfesionalidad
- o Participación
- o Ecología
- o Integración
- o Paz
- o Cooperación
- o Solidaridad

ZONA DE CONTENIDO



Localización

Características

Instalaciones

Servicios

Información

Proyectos de escuela

ZONA DEL MENU

Tipo

Usuario

Clave

ZONA DE AUTENTICACIÓN

Figura 10. Formato de las páginas

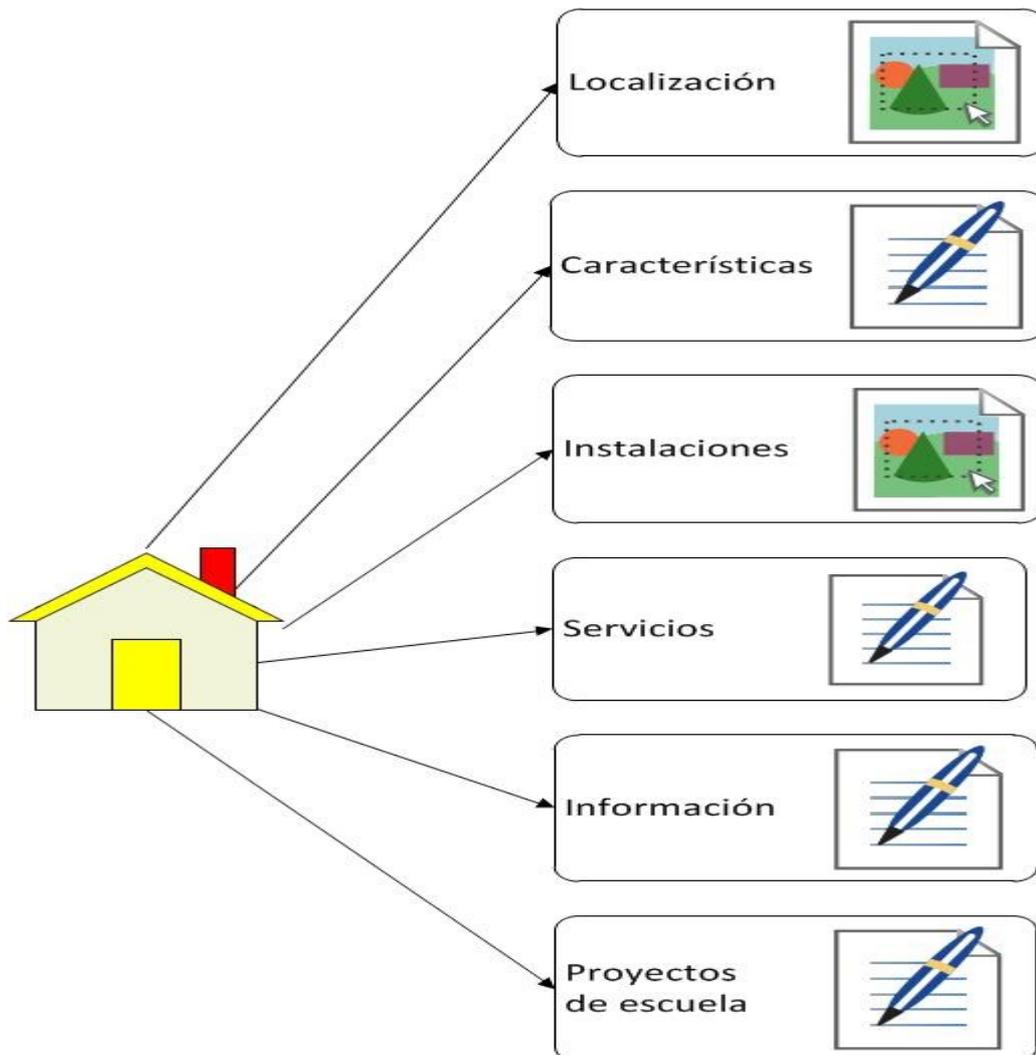


Figura 11. Diagrama de navegabilidad de la página de Inicio

4.3 Capa de negocio o lógica de la aplicación

La capa de negocio se puede definir como el conjunto de componentes software que implementan completamente el comportamiento de las clases del dominio, especificadas en la fase de modelado conceptual. Es en este nivel, por tanto, donde se implementa la funcionalidad de la aplicación web. Esta capa sirve de enlace entre los niveles de presentación y de persistencia (figura 12), ya que la capa de presentación no accede a la base de datos directamente, sino que se comunica con la capa de aplicación para demandarle el servicio deseado y es la capa de aplicación la que se comunica con la capa de persistencia para recuperar los datos necesarios.

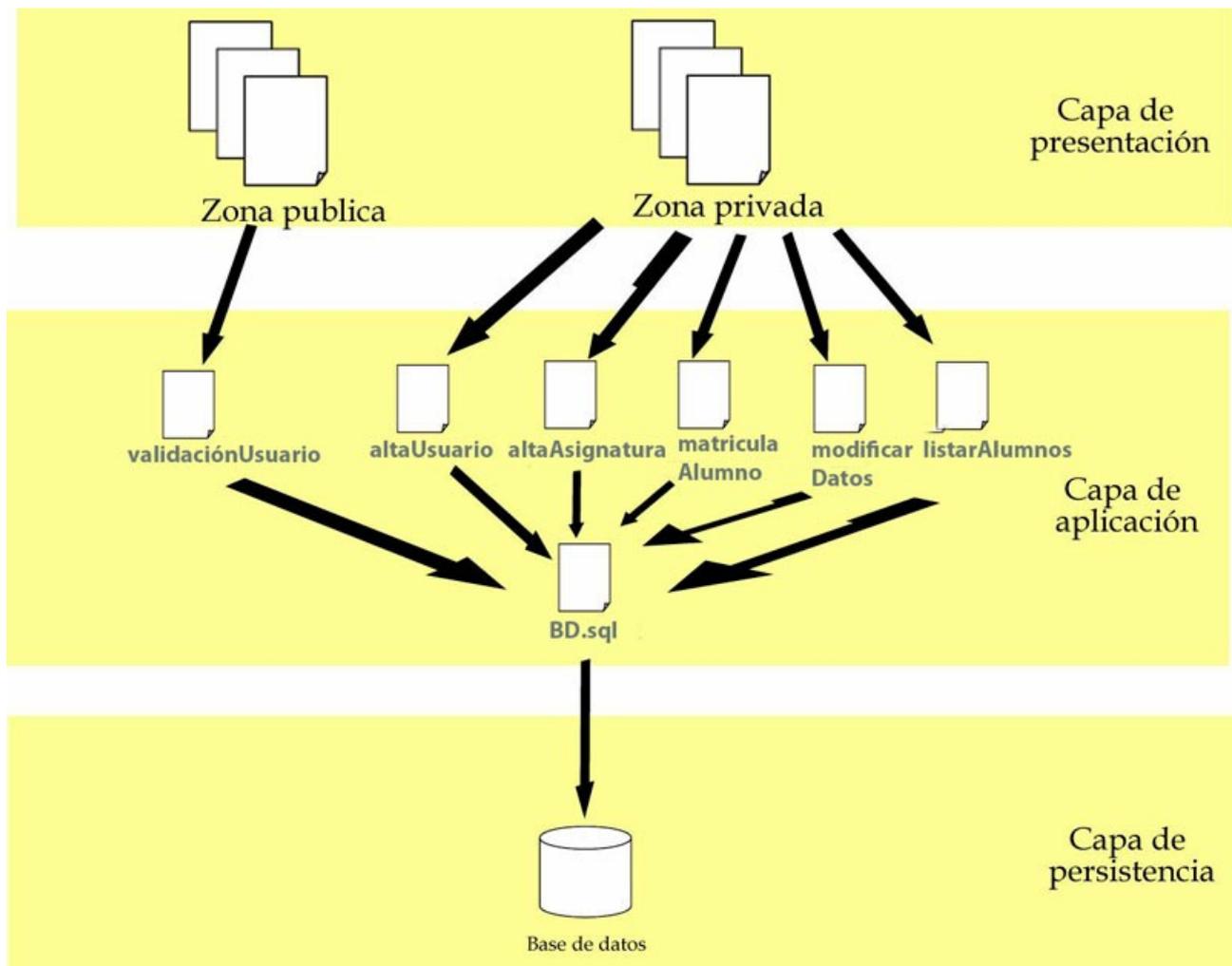


Figura 12. Capas del proyecto

En el proyecto la capa de negocio se corresponde con las clases, es decir, con los ficheros que representan los distintos elementos del diagrama de clases y sus colecciones, almacenadas en la carpeta "Clases". Hay dos tipos de estructura, dependiendo de si se trata de la definición de un elemento o la definición de una colección de elementos.

Los elementos son Administrador, Alumno, Profesor, Asignatura, Nivel, Horario, Nota y FaltaAsistencia y tienen la siguiente estructura:

- **Atributos:** Define las propiedades del elemento.

- **Método constructor:** Crea una instancia del elemento y da valor a sus atributos.
- **Métodos get y set:** Devuelven o establecen el valor de los atributos del elemento.
- **Método leer:** Dado un identificador, recupera de la base de datos las propiedades del elemento con ese identificador.
- **Método insertar:** Inserta en la base de datos un elemento con los atributos que se supone se han establecido previamente con llamadas a métodos set.
- **Método modificar:** Modifica en la base de datos un elemento con los atributos que se supone se han establecido previamente con llamadas a métodos set.
- **Método eliminar:** Dado un identificador, elimina de la base de datos el elemento con ese identificador.
- **Método visualizar:** Muestra el valor de los atributos de un elemento.

Las colecciones de elementos son `coleccionAlumnos`, `coleccionProfesores`, `coleccionAsignaturas`, `coleccionNiveles`, `coleccionHorarios`, `coleccionNotas` y `coleccionFaltasAsistencia` y tienen la siguiente estructura:

- **Atributos:** Define las propiedades de la colección de elementos.
- **Método constructor:** Crea una instancia de la colección y la “rellena” con los elementos que toque en función del valor de los parámetros que se le han pasado.
- **Método obtenerLista:** Devuelve una lista con los distintos elementos que conforman la colección.
- **Método cantidadElementos:** Devuelve la cantidad de elementos que hay en la colección.
- **Método visualizar:** Realiza la llamada al método `visualizar` sobre los distintos elementos que conforman la colección y devuelve una tabla con los datos.

4.4 Capa de persistencia o datos

La capa de persistencia se puede definir como el conjunto de componentes software que proporcionan una serie de servicios que permiten a los objetos del dominio interactuar con su repositorio permanente asociado.

En el proyecto la capa de persistencia se corresponde con la base de datos de la aplicación y las distintas tablas que la conforman. Estas son `administrador`, `alumno`, `profesor`, `asignatura`, `alumno_asignatura`, `nivel`, `horario`, `nota` y `falta_asistencia`.

A continuación se muestra el diagrama entidad-relación (figura 12) utilizado para la implementación de la base de datos de nuestro proyecto.

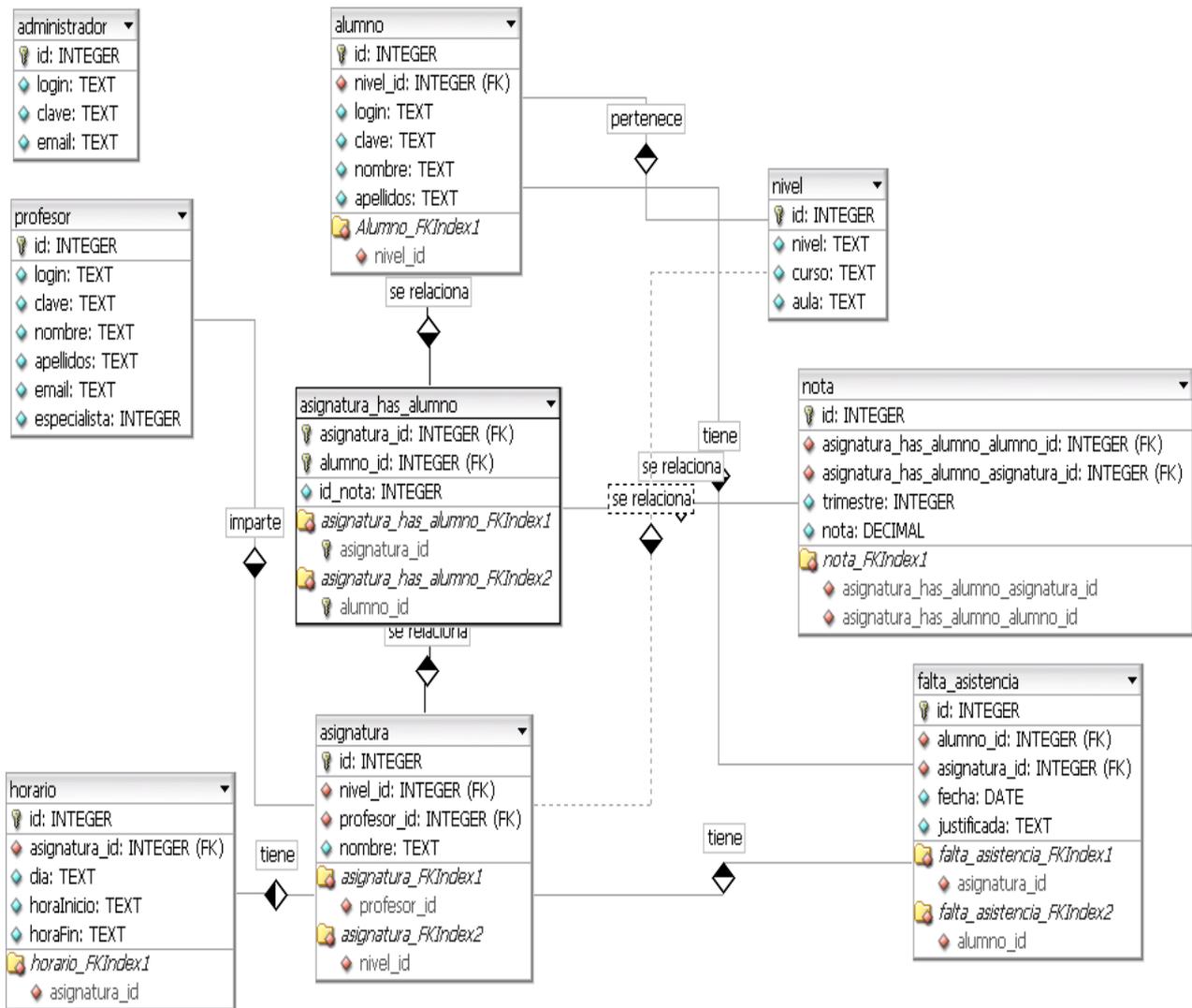


Figura 13. Diagrama entidad-relación

5. IMPLEMENTACIÓN

5.1 Tecnologías utilizadas en el desarrollo del proyecto

5.1.1 HTML y XHTML

HTML, siglas de HyperText Markup Language (Lenguaje de Marcado de Hipertexto), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>).

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium, más conocido como W3C. Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo. El propio W3C define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global". Por convención, los archivos de formato HTML usan la extensión .htm o .html.

XHTML es el sucesor de HTML. XHTML, acrónimo inglés de eXtensible Hypertext Markup Language (lenguaje extensible de marcado de hipertexto), es el lenguaje de marcado pensado para sustituir a HTML como estándar para las páginas web. XHTML es la versión XML de HTML, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones, más estrictas, de XML.

XHTML tiene como objetivo avanzar en el proyecto del W3C de lograr una web semántica, donde la información, y la forma de presentarla estén claramente separadas. En este sentido, XHTML serviría únicamente para transmitir la información que contiene un documento, dejando para hojas de estilo y JavaScript su aspecto y diseño en distintos medios.

5.1.2 CSS

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML. En este último podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- Control centralizado de la presentación de un sitio web completo, con lo que se agiliza de forma considerable la actualización del mismo.
- Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web remoto, con lo que aumenta considerablemente la accesibilidad.

- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

5.1.3 JavaScript, jQuery y AJAX

JavaScript es un lenguaje interpretado utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java. Sin embargo, al contrario que Java, JavaScript no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia. Es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.

Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM (Modelo de Objetos del Documento). JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

jQuery es una biblioteca o framework de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

5.1.4 PHP

PHP es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Aunque todo en su diseño está orientado a facilitar la creación de página web, es posible crear aplicaciones con una interfaz gráfica para el usuario, utilizando la extensión PHP-Qt o PHP-GTK. También puede ser usado desde la línea de órdenes, de la misma manera como Perl o Python pueden hacerlo; a esta versión de PHP se la llama PHP-CLI (Command Line Interface).

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente. Mediante extensiones es también posible la generación de archivos PDF, Flash, así como imágenes en diferentes formatos.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos, tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

PHP es una alternativa a las tecnologías de Microsoft ASP y ASP.NET, a ColdFusion de la compañía Adobe, a JSP/Java de Sun Microsystems, y a CGI/Perl.

[5.1.5 SQL](#)

El lenguaje de consulta estructurado (SQL Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar de una forma sencilla información de interés de una base de datos, así como también hacer cambios sobre la misma.

[5.1.6 MySQL](#)

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

MySQL es muy utilizado en aplicaciones web como MediaWiki, Amazon, Yahoo, Flickr o Drupal; en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP.

[5.1.7 PhpMyAdmin](#)

PhpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar bases de datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas. Se encuentra disponible bajo la licencia GPL.

[5.1.8 StarUML](#)

StarUML es un proyecto de software libre que intenta reemplazar las herramientas de UML comerciales tales como Rational Rose. Se ha utilizado para desarrollar el diagrama de clases y los diagramas de casos de uso en la etapa de análisis del proyecto. Debido a su interfaz intuitiva, se trata de una herramienta fácil de manejar para el usuario.

[5.1.9 Adobe Dreamweaver](#)

Adobe Dreamweaver es una aplicación en forma de estudio enfocada a la construcción y edición de sitios y aplicaciones web basados en estándares. Es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del W3C. Su principal competidor es Microsoft Expression Web y tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras.

Las versiones originales de la aplicación se utilizaban como simples editores WYSIWYG. Sin embargo, versiones más recientes soportan otras tecnologías web como CSS, JavaScript y algunos frameworks del lado servidor. Esta aplicación está disponible tanto para la plataforma MAC como para Windows, aunque también se puede ejecutar en plataformas basadas en UNIX utilizando programas que implementan las API's de Windows, tipo Wine.

[5.1.10 DBDesigner](#)

DBDesigner es un sistema totalmente visual de diseño de bases de datos, que combina características y funciones profesionales con un diseño simple, muy claro y fácil de usar, a fin de ofrecer un método efectivo para gestionar una base de datos. Permite administrar la base de datos, diseñar tablas, hacer peticiones SQL manuales, ingeniería inversa en MySQL, Oracle, MSSQL y otras bases de datos ODBC, modelos XML y soporte para la función drag-and-drop.

5.2 Descripción del proyecto

Las páginas Web que componen la aplicación están implementadas siguiendo una estrategia basada en contenidos, es decir, las páginas Web se estructuran en zonas (divs) donde cada una es responsable de proporcionar cierta información sobre un contenido en concreto.

A continuación va a procederse a hacer una descripción del proyecto siguiendo el esquema por capas que se ha seguido para implementarlo. Para cada capa, se ha elegido un fichero de los que la conforman para explicar su estructura, ya que el número de zonas o divs es siempre el mismo.

[5.2.1 Capa de presentación](#)

A continuación se muestra el contenido de uno de los ficheros que conforman la capa de presentación. Para poder explicarlo mejor, he dividido el fichero y he añadido un breve texto explicativo antes de cada una de las partes en las que ha sido dividido.

En la cabecera del documento, se “importan” el fichero CSS que define el estilo de la página y el script JavaScript que nos permitirá hacer la presentación de la web más dinámica y amigable.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Inicio</title>
    <link href="CSS/estilo.css" rel="stylesheet" type="text/css"/>
    <script type="text/javascript"
src="http://cdn.jquerytools.org/1.2.5/full/jquery.tools.min.js"></script>

</head>

```

La primera zona del cuerpo del documento el contenedor, dentro de la cual se añadirán todas las zonas que conforman la aplicación. La primera en añadirse es la zona correspondiente al logo del colegio, compuesta únicamente por una imagen que sirve de enlace a la página de inicio.

```

<body>

    <div id="contenedor">

        <div id="logo">
            <a href="index.php" title="Inicio">
                
            </a>
        </div>

```

La segunda zona que se añade al contenedor de zonas es la correspondiente al menú, compuesta únicamente por una lista no numerada de enlaces a las distintas páginas que conforman la parte pública de la aplicación.

```

    <div id="menu">
        <ul>
            <li><a href="localizacion.php">Localizaci&oacute;n</a></li>
            <li><a href="caracteristicas.php">Caracter&iacute;sticas</a></li>
            <li><a href="instalaciones.php">Instalaciones</a></li>
            <li><a href="servicios.php">Servicios</a></li>
            <li><a href="informacion.php">Informaci&oacute;n</a></li>
            <li><a href="proyectosEscuela.php">Proyectos de escuela</a></li>
        </ul>
    </div>

```

La siguiente zona que se añade al contenedor de zonas es la correspondiente a la autenticación del usuario, compuesta por un par de cajas para que se escriba el nombre de usuario y la contraseña y un botón para que se envíen los datos introducidos al formulario de validación. Si los datos de acceso son correctos se redirige a la página de inicio del usuario registrado, si no, se redirige a esta página pasando como parámetro la variable error. En función del valor de esta variable se muestra un mensaje de error u otro en la zona del usuario, a continuación del botón “Enviar”.

Si en lugar de en la parte pública nos encontramos en la parte privada de la aplicación, lo que se vería en esta zona sería el nombre de usuario del usuario conectado y un enlace para cerrar sesión.

```
<div id="usuario">
  <form id="logea" class="expose" method="post" action="validacionUsuario.php">
    <label for="tipoUsuario">Tipo</label>
    <select id="tipoUsuario" name="tipoUsuario">
      <option value="alumno">Alumno</option>
      <option value="profesor">Profesor</option>
      <option value="administrador">Administrador</option>
    </select>
    <br clear="left"/>
    <label for="usuarioText">Usuario</label>
    <input id="usuarioText" type="text" name="usuario"/>
    <br clear="left"/>
    <label for="clave">Clave</label>
    <input id="clave" type="password" name="clave" />
    <br clear="left">
    <input style="width:60px;" id="submit" type="submit" value="Entrar" />
    <?php if ($_GET["error"]=="0"){ //login vacio ?>
    <span><b>Usuario vacio</b></span>
    <?php } if ($_GET["error"]=="1"){ //clave vacia ?>
    <span><b>Clave vacia</b></span>
    <?php } if ($_GET["error"]=="2"){ //login incorrecto ?>
    <span><b>Usuario incorrecto</b></span>
    <?php }else if ($_GET["error"]=="3"){ //Clave incorrecta ?>
    <span><b>Clave incorrecta</b></span>
    <?php } ?>

    <script type="text/javascript">
      jQuery(document).ready(function(){
        jQuery('form span').fadeOut();
      });
    </script>

  </form>
</div>
```

La última zona que se añade al contenedor es la correspondiente al contenido. En este caso, como nos encontramos en la página de inicio de la parte pública de la aplicación, sólo se visualiza una imagen y algunos datos de contacto del centro Monte Alegre.

```
<div id="inicio" class="contenido">
  <center>
    <h3>Bienvenido/a al CEIP Monte Alegre</h3>
  </center>
```

```
<br>
```

```
<p id="parrfoIntro">  
<b>Direcci&oacute;n:</b><br>  
C/ Jos&eacute; Alegre Rubio, nº 11<br><br>  
<b>Localidad:</b><br>  
46183 L'Eliana, Valencia<br><br>  
<b>Tel&eacute;fono:</b><br>  
962743609<br><br>  
<b>Fax:</b><br>  
962758780<br><br>  
<b>Email:</b><br>  
46024503@edu.gva.es<br>  
</p>
```

```
</div>
```

```
</div><!-- Contenedor ->
```

Por último, se ejecuta un script para hacer la presentación de la información más dinámica.

```
<script type="text/javascript">  
  
jQuery(document).ready(function(){  
    var form = jQuery(".expose").bind("click keydown", function() {  
        jQuery(this).expose({  
            onLoad: function() {  
                form.css({backgroundColor: '#eeeeee'});  
            },  
            onClose: function() {  
                form.css({backgroundColor: null});  
            }  
        });  
    });  
  
    jQuery('#inicio').fadeIn();  
  
});  
  
</script>  
  
</body>  
</html>
```

5.2.2 Capa de negocio o lógica de la aplicación

A continuación se muestra el contenido de uno de los ficheros que conforman la capa de negocio. Para poder explicarlo mejor, de la misma manera que para la capa de presentación, he dividido el fichero y he añadido un breve texto explicativo antes de cada una de las partes en las que ha sido dividido.

Se definen los atributos del elemento.

```
<?php
```

```
class Alumno {  
  
    private $id;  
    private $login;  
    private $clave;  
    private $nombre;  
    private $apellidos;  
    private $id_nivel;
```

Se implementa el método constructor. Se da valor a los atributos (en un principio iguales a 0 o a "").

```
    public function __construct($id=0, $login="", $clave="", $nombre="", $apellidos="",  
$id_nivel=0){  
        $this->id = $id;  
        $this->login = $login;  
        $this->clave = $clave;  
        $this->nombre = $nombre;  
        $this->apellidos = $apellidos;  
        $this->id_nivel = $id_nivel;  
    }  
}
```

Se implementan los métodos get y set que devuelven o establecen el valor de los atributos.

```
    public function getIdentificador(){  
        return $this->id;  
    }  
  
    public function getId($login, $clave, $nombre, $apellidos, $id_nivel){  
        $bd = new AuxBD();  
        $bd->conectar();  
        $sql = "Select id From ";  
        $sql.= "alumno Where login='".$login.'" AND ";  
        $sql.= "clave='".$clave.'" AND ";  
        $sql.= "nombre='".$nombre.'" AND ";  
        $sql.= "apellidos='".$apellidos.'" AND ";  
        $sql.= "id_nivel='".$id_nivel.'";";  
        $resultado = $bd->ejecutarSQL($sql);
```

```

        $bd->desconectar();
        $fila = $bd->siguienteFila($resultado);
        return $fila['id'];
    }

    public function getLogin(){
        return $this->login;
    }

    public function setLogin($nuevoLogin){
        $this->login = $nuevoLogin;
    }

    public function getClave(){
        return $this->clave;
    }

    public function setClave ($nuevaClave){
        $this->clave = $nuevaClave;
    }

    public function getNombre(){
        return $this->nombre;
    }

    public function setNombre ($nuevoNombre){
        $this->nombre = $nuevoNombre;
    }

    public function getApellidos(){
        return $this->apellidos;
    }

    public function setApellidos ($nuevosApellidos){
        $this->apellidos = $nuevosApellidos;
    }

    public function getIdNivel(){
        return $this->id_nivel;
    }

    public function setIdNivel ($nuevoldNivel){
        $this->id_nivel = $nuevoldNivel;
    }
}

```

Se implementa el método leer, que dado un identificador, recupera de la base de datos el valor de los atributos del elemento (en este caso alumno) con ese identificador.

```

public function leer ($id){
    $this->id = $id;
    if($this->id>0){
        $bd = new AuxBD();
        $bd->conectar();
        $sql = "Select * From ";
        $sql.= "alumno Where id='". $this->id. "'";
        $resultado = $bd->ejecutarSQL($sql);
        $bd->desconectar();
        $fila = $bd->siguienteFila($resultado);
        $this->login = $fila['login'];
        $this->clave = $fila['clave'];
        $this->nombre = $fila['nombre'];
        $this->apellidos = $fila['apellidos'];
        $this->id_nivel = $fila['id_nivel'];
    }
}

```

Se implementa el método insertar, que inserta en la base de datos un nuevo alumno. Se supone que los atributos del objeto ya recibieron las asignaciones a través de llamadas a métodos set.

```

public function insertar(){
    $bd = new AuxBD();
    $bd->conectar();
    $sql = "Insert Into alumno (id, login, clave, nombre, apellidos, id_nivel) Values ('";
    $sql.= $this->id. "',";
    $sql.= mysql_escape_string ($this->login). "',";
    $sql.= mysql_escape_string ($this->clave). "',";
    $sql.= mysql_escape_string ($this->nombre). "',";
    $sql.= mysql_escape_string ($this->apellidos). "',";
    $sql.= $this->id_nivel. "')";
    $bd->ejecutarSQL($sql);
    $bd->desconectar();
}

```

Se implementa el método modificar, que modifica en la base de datos la información de un alumno. Se supone que ya se ha hecho uso del método leer para recuperar la información de un alumno y de los métodos set para establecer los nuevos valores de los atributos.

```

public function modificar (){
    $bd = new AuxBD();
    $bd->conectar();
    $sql = "Update alumno Set login='".mysql_escape_string ($this->login). "'";
    $sql.= "Where id='". $this->id. "'";
    $bd->ejecutarSQL($sql);
    $sql = "Update alumno Set clave='".mysql_escape_string ($this->clave). "'";
    $sql.= "Where id='". $this->id. "'";
    $bd->ejecutarSQL($sql);
}

```

```

        $sql = "Update alumno Set nombre='".mysql_escape_string ($this->nombre)."'";
        $sql.= "Where id='". $this->id. "'";
        $bd->ejecutarSQL($sql);
        $sql = "Update alumno Set apellidos='".mysql_escape_string ($this->apellidos)."'";
        $sql.= "Where id='". $this->id. "'";
        $bd->ejecutarSQL($sql);
        $sql = "Update alumno Set id_nivel='". $this->id_nivel. "'";
        $sql.= "Where id='". $this->id. "'";
        $bd->ejecutarSQL($sql);
        $bd->desconectar();
    }

```

Se implementa el método eliminar, que dado un identificador, eliminar de la base de datos el alumno con ese identificador.

```

    public function eliminar ($id){
        $bd = new AuxBD();
        $bd->conectar();
        $sql = "Delete From alumno_asignatura Where id_alumno=". $id. " ";
        $bd->ejecutarSQL($sql);
        $sql = "Delete From alumno Where id=". $id. " ";
        $bd->ejecutarSQL($sql);
        $bd->desconectar();
    }

```

Se implementa el método visualizar, que muestra el valor de algunos de los atributos del elemento.

```

    public function visualizar(){
        setlocale (LC_TIME, "sp_SP");
        $resultado = "<tr><td>".nl2br($this->nombre)."</td><td>&nbsp;&nbsp;&nbsp;</td>";
        $resultado.= "<td>".nl2br($this->apellidos)."</td><td>&nbsp;&nbsp;&nbsp;</td></tr>";
        return $resultado;
    }
}
?>

```

5.2.3 Capa de persistencia o de datos

A continuación se muestran las distintas tablas (figura 13) que conforman la base de datos de la aplicación:



Servidor: localhost:3306 ▶ Base de datos: montealegre

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Eliminar

Tabla	Acción	Registros	Tipo	Cotejamiento	Tamaño	Residuo a depurar
<input type="checkbox"/> administrador		1	MyISAM	latin1_spanish_ci	2.0 KB	-
<input checked="" type="checkbox"/> alumno		50	MyISAM	latin1_spanish_ci	5.1 KB	-
<input type="checkbox"/> alumno_asignatura		346	MyISAM	latin1_spanish_ci	18.2 KB	-
<input type="checkbox"/> asignatura		68	MyISAM	latin1_spanish_ci	5.9 KB	-
<input type="checkbox"/> falta_asistencia		1	MyISAM	latin1_spanish_ci	2.1 KB	36 Bytes
<input type="checkbox"/> horario		330	MyISAM	latin1_spanish_ci	22.2 KB	-
<input type="checkbox"/> nivel		9	MyISAM	latin1_spanish_ci	2.4 KB	-
<input type="checkbox"/> nota		3	MyISAM	latin1_spanish_ci	2.2 KB	132 Bytes
<input type="checkbox"/> profesor		15	MyISAM	latin1_spanish_ci	3.3 KB	-
9 tabla(s)	Número de filas	823	MyISAM	latin1_spanish_ci	63.5 KB	168 Bytes

⬆ Marcar todos/as / Desmarcar todos / Marcar las tablas con residuo a depurar Para los elementos que están marcados: ▾

Figura 14. Tablas de la base de datos

Como se aprecia en la figura, la base de datos está compuesta por nueve tablas: administrador, alumno, alumno_asignatura, asignatura, falta_asistencia, horario, nivel, nota y profesor. La tabla alumno, por ejemplo, está compuesta por los siguientes campos:

- **id**: Almacena el identificar de un alumno. Es la clave primaria.
- **login**: Almacena el nombre de usuario de un alumno.
- **clave**: Almacena la contraseña de un alumno.
- **nombre**: Almacena el nombre de un alumno.
- **apellidos**: Almacena los apellidos de un alumno.
- **id_nivel**: Almacena el identificador del nivel al que pertenece un alumno (3, 4 o 5 años de Infantil o 1°, 2°, 3°, 4°, 5° o 6° de Primaria). Es una clave ajena que apunta a la tabla Nivel.

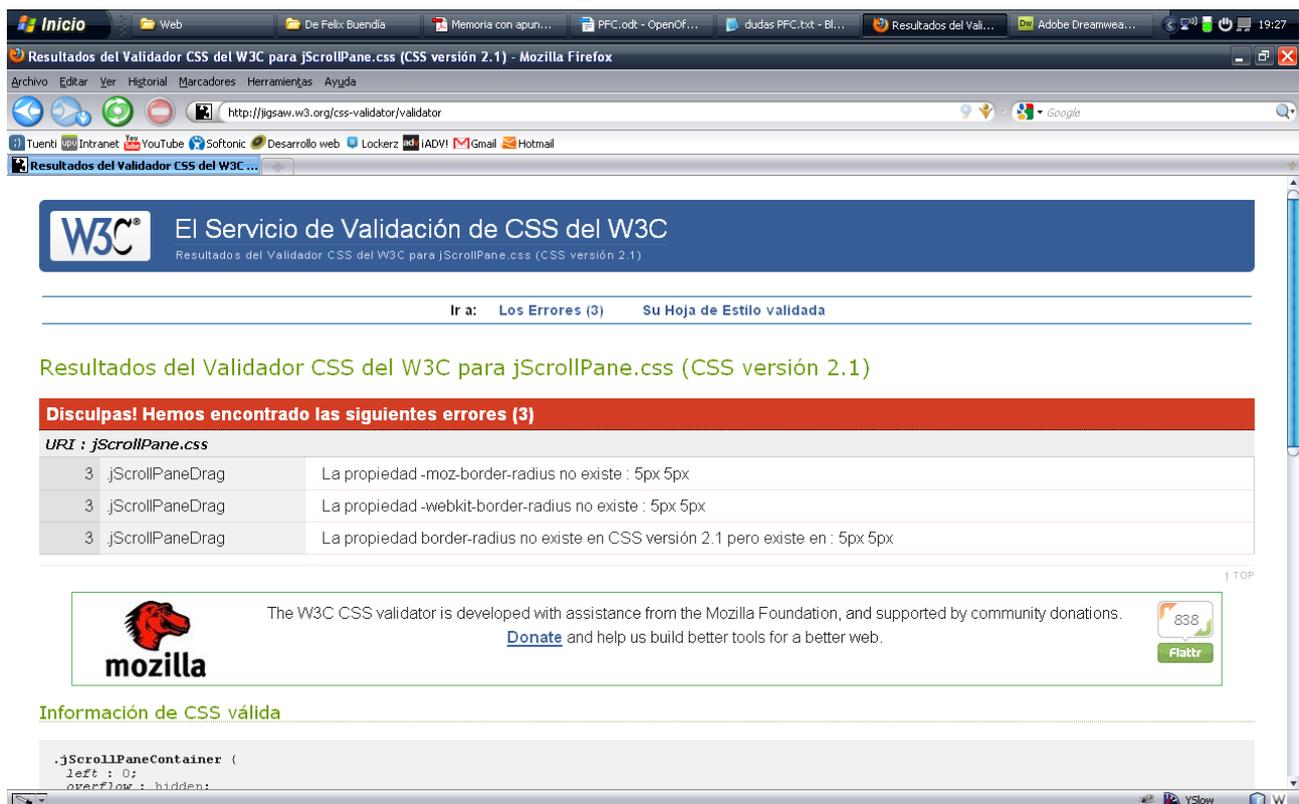
6. EVALUACIÓN

6.1 Introducción

La fase de evaluación y pruebas concluye el ciclo de vida de nuestro proyecto y lo prepara para subirlo al servidor y que se ejecute. Esta fase mide el nivel de calidad que ofrece al usuario la aplicación creada. Las herramientas que vamos a usar para realizar estas pruebas son gratuitas y las podemos encontrar online en las direcciones <http://jigsaw.w3.org/css-validator/>, <http://validator.w3.org/> y www.spoon.com/.

6.2 Validación de CSS

Dado que se han utilizado hojas de estilo para la interfaz de la aplicación, se ha realizado una prueba de validación de CSS. Para ello, se ha utilizado la página web <http://jigsaw.w3.org/css-validator/>. En cuanto al resultado (figura 14) se podría decir que la hoja de estilo ha pasado la validación a medias, ya que hay dos propiedades que el W3C no reconoce (-moz-border-radius y -webkit-border-radius) pero que son necesarias para que los bordes de los divs que hay en la página web se vean redondeados en los navegadores web Firefox, Google Chrome, Safari y Opera.



The screenshot shows the W3C CSS Validator interface. At the top, it says "El Servicio de Validación de CSS del W3C" and "Resultados del Validador CSS del W3C para jScrollPane.css (CSS versión 2.1)". Below this, it indicates "Ir a: Los Errores (3) Su Hoja de Estilo validada". The main heading is "Resultados del Validador CSS del W3C para jScrollPane.css (CSS versión 2.1)". A red banner states "Disculpas! Hemos encontrado las siguientes errores (3)". Below this, a table lists the errors:

URI : jScrollPane.css		
3	jScrollPaneDrag	La propiedad -moz-border-radius no existe : 5px 5px
3	jScrollPaneDrag	La propiedad -webkit-border-radius no existe : 5px 5px
3	jScrollPaneDrag	La propiedad border-radius no existe en CSS versión 2.1 pero existe en : 5px 5px

Below the table, there is a Mozilla logo and text: "The W3C CSS validator is developed with assistance from the Mozilla Foundation, and supported by community donations. Donate and help us build better tools for a better web." There is also a Flattr button with the number 838.

At the bottom, under "Información de CSS válida", there is a code snippet:

```
.jScrollPaneContainer {  
  left : 0;  
  overflow : hidden;
```

Figura 15. Validación de CSS

6.3 Validación de enlaces

Para esta prueba he utilizado la herramienta “Verificador de vínculos” del programa Adobe Dreamweaver. Dicho software verifica si hay enlaces rotos (enlaces incorrectos o que apuntan a una página que no existe) en el sitio web. El resultado (figura 15) de esta prueba ha sido óptimo.

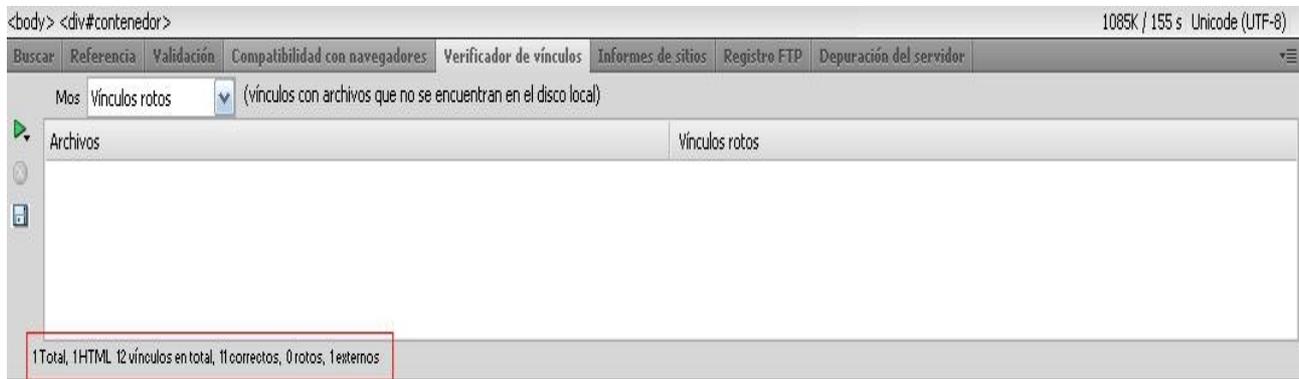


Figura 16. Validación de enlaces

6.4 Validación de la resolución

Durante la fase de especificación de requisitos establecimos que para una correcta visualización de la aplicación deberíamos optimizar la pantalla a una resolución de 1280x800 píxeles. Lo que sucede para resoluciones menores (640x480 o 800x600 píxeles) es que algunos elementos (botones sobre todo) no aparecen en la posición adecuada.

6.5 Validación de navegadores

Es importante de cara al usuario que la aplicación se visualice bien en todos los navegadores web, o al menos, en los más populares. La aplicación ha sido implementada y probada con éxito en Microsoft Internet Explorer (57.08% de cuota de mercado), Firefox (22.81%), Chrome (9.98%), Safari (5.89%) y Opera (2.23%). Para ello he utilizado la herramienta online spoon, que te permite virtualizar navegadores web de forma que puedes visualizar la página web sobre cualquier navegador sin necesidad de instalarlo.

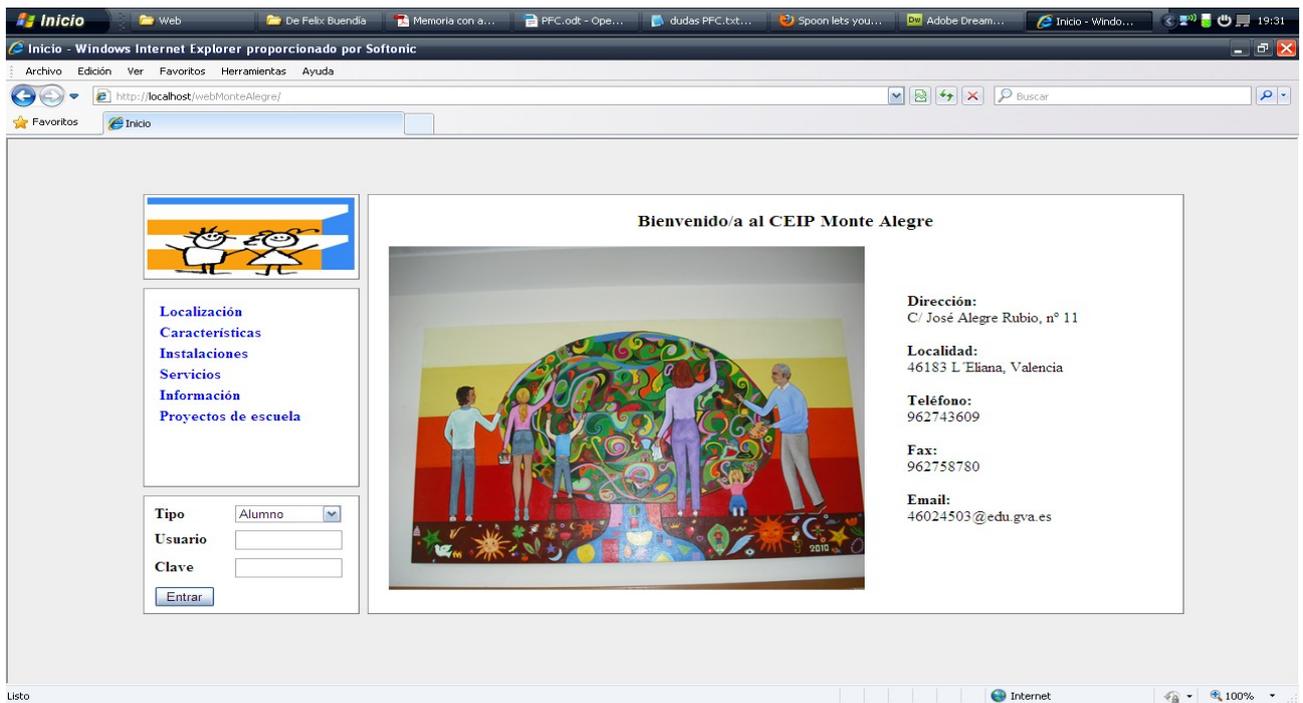


Figura 17. Validación de navegadores (Microsoft Internet Explorer)

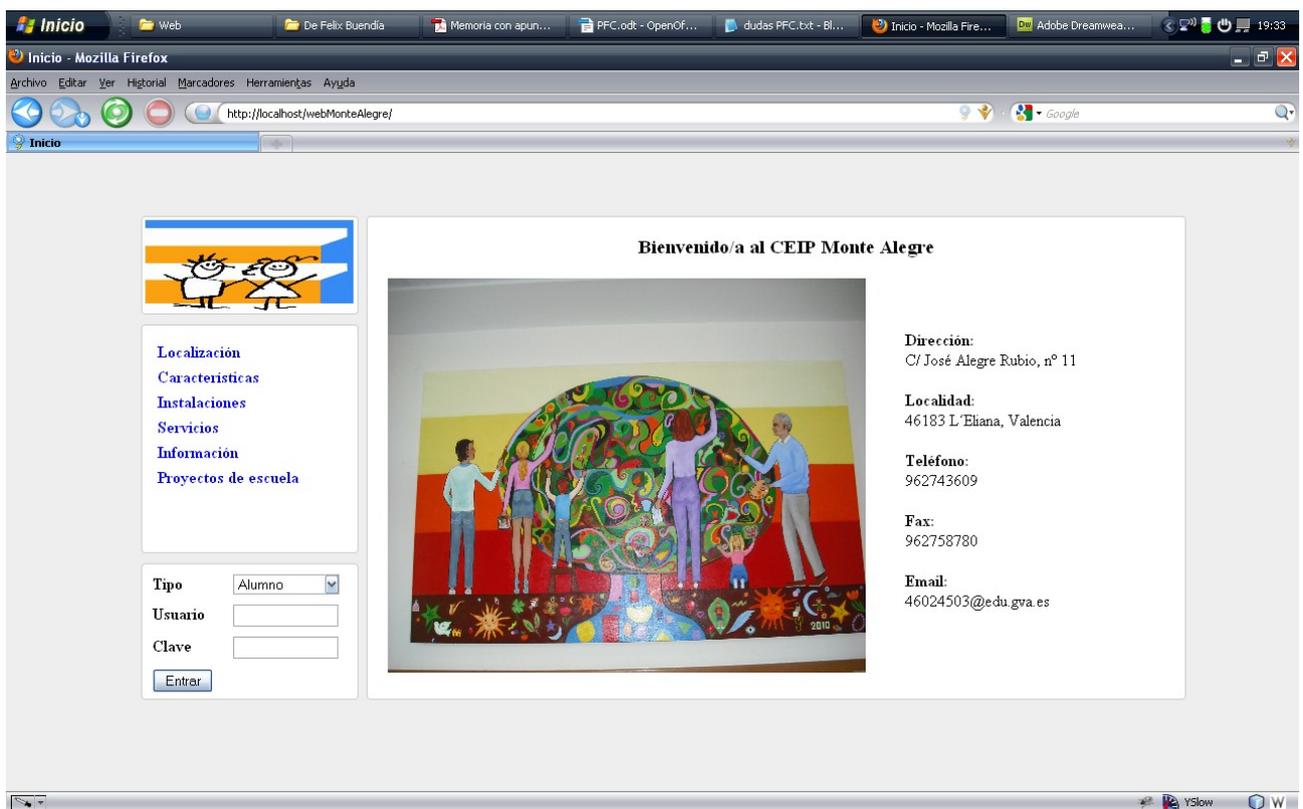


Figura 18. Validación de navegadores (Mozilla Firefox)

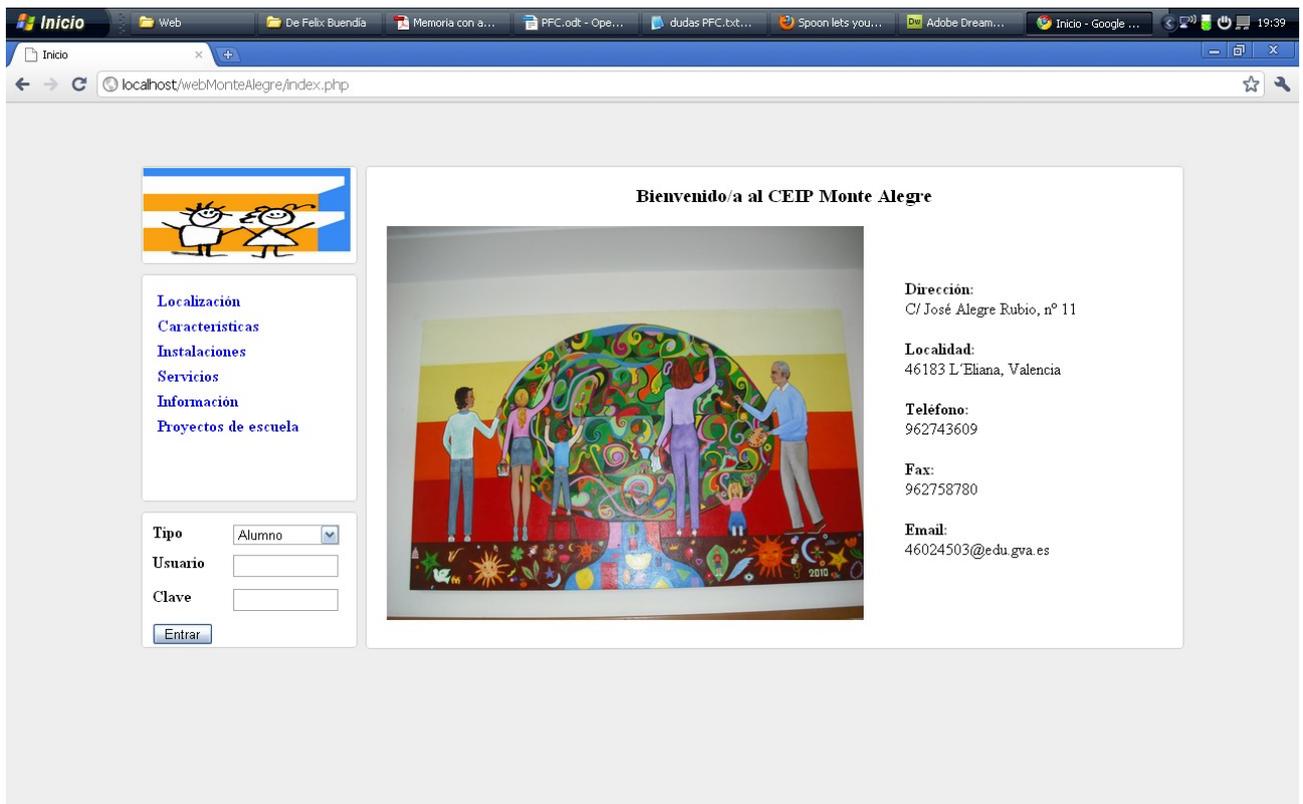


Figura 19. Validación de navegadores (Google Chrome)

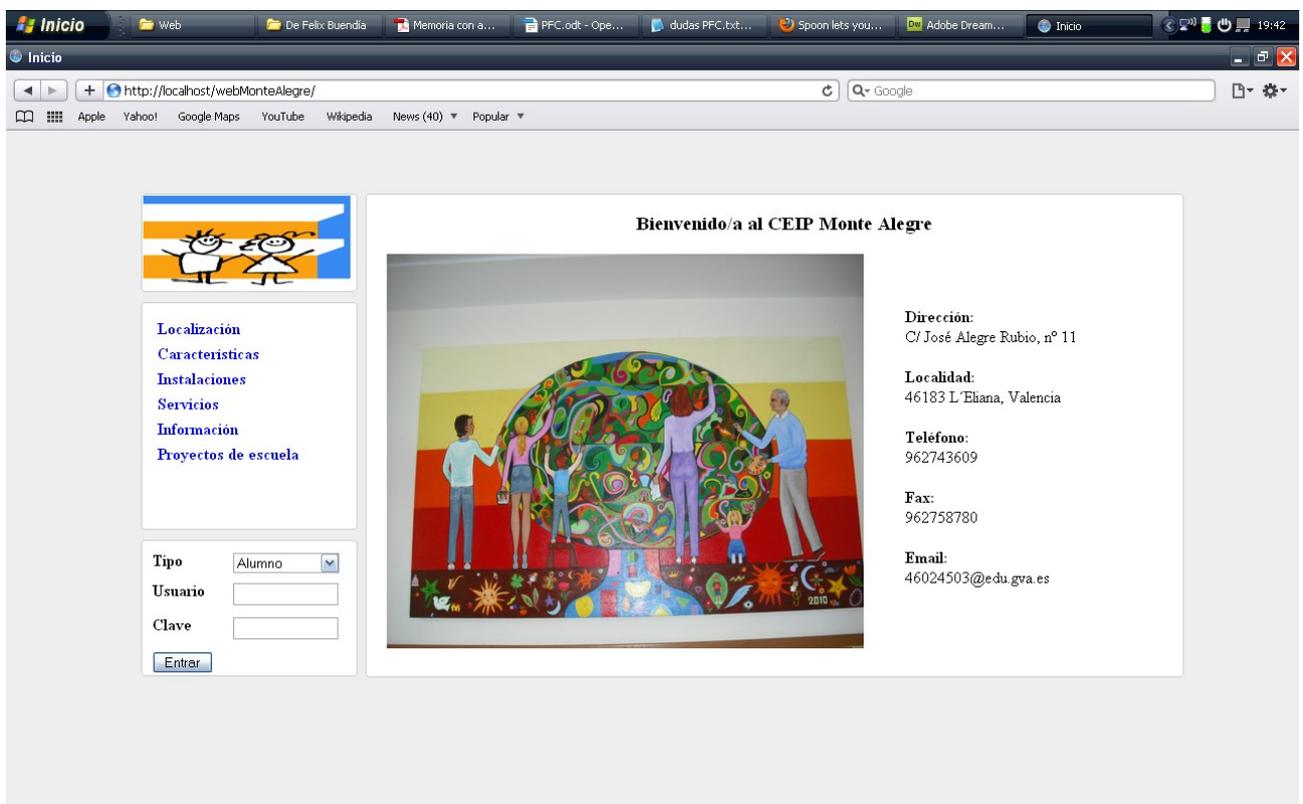


Figura 20. Validación de navegadores (Safari)

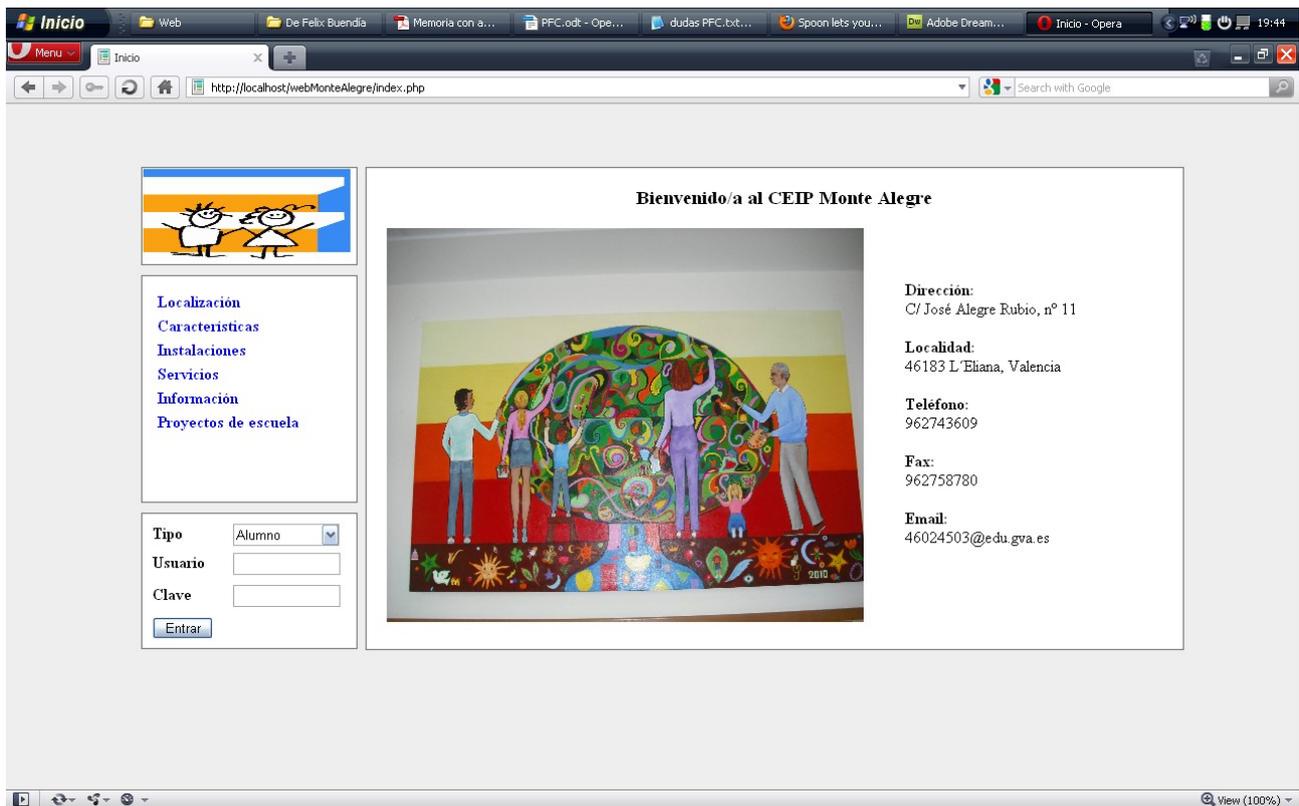


Figura 21. Validación de navegadores (Opera)

6.6 Validación de la seguridad en el acceso a la zona de administración

Para entrar en la zona de administración de la aplicación web, el usuario deberá acceder a través del div situado en el sector inferior izquierdo de la página seleccionando el tipo de usuario Administrador e introduciendo un nombre de usuario y contraseña correctos. Los datos son enviados al formulario de validación “validacionUsuario.php” que redirige a la página de inicio de la parte específica del administrador si los datos son correctos o devuelve un mensaje de error si sucede alguna de las siguientes situaciones: el usuario no ha escrito el login; el usuario no ha escrito la clave; el login no existe en la base de datos; el par login-contraseña no existe en la base de datos.

7. CONCLUSIÓN

A continuación haré un resumen del trabajo realizado. El primer paso que tomé fue establecer los requisitos que debía cumplir el sistema reuniéndome con la directora del centro educativo. En segundo lugar, realicé los modelos oportunos en UML para que me sirvieran de guía durante la implementación. Como escogí utilizar la arquitectura de tres capas, el diseño de las distintas capas que conforman la aplicación fue el siguiente paso. A continuación, llevé a cabo la etapa a la que más tiempo dediqué, la implementación del prototipo. En esta última etapa podría distinguir cuatro fases: la del diseño de la web (primera aproximación de la hoja de estilos), la de la implementación de la intranet del administrador, la de la implementación de la intranet del usuario alumno/padre y la de la implementación de la intranet del usuario profesor. Por último y para comprobar el correcto funcionamiento de todo el sistema, llevé a cabo las pruebas de la aplicación.

7.1 Validación personal del trabajo realizado

Durante la realización del proyecto he podido comprobar lo útiles que son los conocimientos adquiridos durante la carrera en asignaturas como Ingeniería del Software, Base de Datos, Diseño de base de Datos y Programación.

Considero que este proyecto me ha aportado una visión profesional del programador web, ya que he tenido que tratar directamente con el cliente y he trabajado “bajo presión”, en el sentido de que he tenido que cumplir ciertos plazos de entrega.

Cabe destacar que, por tratarse de un prototipo, la funcionalidad de la aplicación web puede no ser completa. Muchas mejoras y ampliaciones, como las que se citan en el siguiente punto, quedan fuera del alcance de este proyecto por su duración y extensión. Sin embargo, las funciones descritas en la especificación de requisitos están implementadas. Luego, bajo mi punto de vista, se trata de un trabajo realizado satisfactoriamente.

7.2 Posibles ampliaciones

A continuación se listan una serie de ampliaciones que pudieran implementarse sobre el prototipo:

- Gestión de la biblioteca.
- Tablón de anuncios y sección de noticias.
- Calendario de exámenes y excursiones.
- Hacer segura la transmisión de datos sensibles de la aplicación como puedan ser las contraseñas, mediante técnicas de encriptación de datos.

8. BIBLIOGRAFÍA

- **Apuntes de ISG, BDA, DBD y CDH.**
- **Página oficial de PHP**
www.php.org
- **Desarrollo Web**
www.desarrolloweb.com
- **Wikipedia, la enciclopedia libre**
www.wikipedia.org
- **PHP 6. Curso profesional de programación**
Edgar D´Andrea
Ediciones InforBooks
- **Una guía para la realización y supervisión de PFC en el ámbito de la Web**
Félix Buendía García
Editorial Universidad Politécnica de Valencia
- **Guía del IEEE para la especificación de requerimientos software**
ANSI/IEEE Std. 830 – 1984