

# Memoria Proyecto Final de Carrera

Portal web para un colegio:  
Internado Meinston

Dirigido por: Félix Buendía

**Lidia Contreras Ochando**  
Ingeniería Técnica en Informática de gestión  
Junio 2011



**INDICE**

<b>1. INTRODUCCIÓN</b>	<b>7</b>
1.1 OBJETIVOS DEL PROYECTO	7
1.2 CONTEXTO	7
1.3 ESTRUCTURA DEL DOCUMENTO	8
<b>2. ESPECIFICACIÓN DE REQUISITOS</b>	<b>9</b>
2.1 INTRODUCCIÓN	9
2.1.1 PROPÓSITO	9
2.1.2 ÁMBITO	9
2.1.3 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS	9
2.1.4 REFERENCIAS	10
2.1.5 VISIÓN GLOBAL	10
2.2 DESCRIPCIÓN GENERAL	10
2.2.1 PERSPECTIVA DEL PRODUCTO	10
2.2.2 FUNCIONES DEL PRODUCTO	10
2.2.3 CARACTERÍSTICAS DE USUARIO	11
2.2.4 RESTRICCIONES GENERALES	12
2.2.5 SUPUESTOS Y DEPENDENCIAS	12
2.3 REQUISITOS ESPECÍFICOS	12
2.3.1 REQUISITOS DE INTERFACES EXTERNAS	12
2.3.2 REQUISITOS FUNCIONALES	12
2.3.3 RESTRICCIONES DE DISEÑO	24
2.3.4 ATRIBUTOS	24
<b>3. ANÁLISIS</b>	<b>25</b>
3.1 UML	25
3.1.1 DIAGRAMA DE CLASES	25
3.1.2 DIAGRAMAS DE CASOS DE USO	27

<b>4. DISEÑO</b>	<b>33</b>
<b>4.1 ARQUITECTURA DEL SISTEMA</b>	<b>33</b>
4.1.1 NIVEL DE PRESENTACIÓN	34
4.1.2 NIVEL DE APLICACIÓN	36
4.1.3 NIVEL DE PERSISTENCIA	37
<b>5. IMPLEMENTACIÓN</b>	<b>38</b>
<b>5.1 TECNOLOGÍAS UTILIZADAS</b>	<b>38</b>
5.1.1 HTML	39
5.1.2 CSS	39
5.1.3 PHP	39
5.1.4 JQUERY (JAVASCRIPT)	39
5.1.5 MYSQL	40
<b>5.2 HERRAMIENTAS UTILIZADAS</b>	<b>40</b>
<b>5.3 IMPLEMENTACIÓN DETALLADA</b>	<b>40</b>
5.3.1 CAPA DE PRESENTACIÓN	40
5.3.2 CAPA DE APLICACIÓN	45
5.3.3 CAPA DE PERSISTENCIA	54
<b>6. EVALUACIÓN Y PRUEBAS</b>	<b>56</b>
<b>6.1 PRUEBA DE ENLACES ROTOS</b>	<b>56</b>
<b>6.2 VALIDACIÓN DE HTML</b>	<b>57</b>
<b>6.3 VALIDACIÓN DE CSS</b>	<b>57</b>
<b>6.4 PRUEBA DE COMPATIBILIDAD PARA MÓVIL</b>	<b>58</b>
<b>6.5 COMPATIBILIDAD CON LOS NAVEGADORES</b>	<b>59</b>
6.5.1 MOZILLA FIREFOX	59
6.5.2 INTERNET EXPLORER	60
6.5.3 GOOGLE CHROME	61
6.5.4 OPERA	62
<b>6.6 COMPATIBILIDAD CON DIFERENTES RESOLUCIONES</b>	<b>63</b>

<b>7. CONCLUSIONES</b>	<b>64</b>
<b>8. BIBLIOGRAFÍA</b>	<b>65</b>
<b>9. ANEXOS</b>	<b>66</b>
<b>9.1 ANEXO A: CASOS DE USO</b>	<b>66</b>
<b>9.2 ANEXO B: DISEÑO CONCEPTUAL DE LA BASE DE DATOS</b>	<b>83</b>
<b>9.3 ANEXO C: DOMINIOS BASE DE DATOS</b>	<b>86</b>
<b>9.4 ANEXO D: FICHERO DE CREACIÓN DE LA BASE DE DATOS</b>	<b>87</b>
<b>9.5 ANEXO E: ESTRUCTURA DE ARCHIVOS</b>	<b>94</b>
<b>9.6 ANEXO F: MANUAL DE USUARIO</b>	<b>96</b>
9.6.1 PÁGINA PRINCIPAL	96
9.6.2 INTRANET	100
<b>9.5 ANEXO G: ÍNDICE DE ILUSTRACIONES</b>	<b>115</b>



## 1. Introducción

Este documento es el Proyecto Final de Carrera de Ingeniería Técnica en Informática de Gestión, cursado en la Escuela Técnica Superior de Ingeniería informática de la Universidad Politécnica de Valencia.

Durante el resto del documento podremos ver la evolución que ha sufrido el proyecto para su desarrollo, desde el planteamiento y análisis de requisitos, hasta su implementación y diseño de la interfaz visible para el usuario final.

### 1.1 Objetivos del proyecto

La finalidad es crear un portal web para un colegio-internado, en el que los familiares puedan tener un seguimiento de la educación y la evolución de sus hijos; los alumnos puedan comunicarse con sus familiares, además de ver sus notas y su progreso; y los profesores puedan poner a sus alumnos las notas correspondientes, las observaciones requeridas o puedan evaluar el seguimiento de sus alumnos tutelados.

El portal del colegio, en adelante *Meinston*, tratará de cumplir dos objetivos esenciales:

- 1- Servir de portal informativo, claro y ordenado, en el que quede patente para todo el que lo visite, de las intenciones y las cualidades del colegio.
- 2- Albergar una intranet para personal docente, alumnos y familiares, donde poder comunicarse entre ellos de forma rápida y simple, donde los padres puedan ver el progreso de sus hijos y los profesores comunicarse con ellos.

### 1.2 Contexto

El colegio Meinston, es un internado ubicado en los Pirineos, en un inmejorable entorno natural, ofreciendo un amplio programa educativo al aire libre.

Enseña a alumnos desde los 2 años hasta 2º de Bachillerato, es bilingüe español-inglés y sirve de residencia a los alumnos que deciden vivir allí el año escolar entero.

La directiva del colegio ha decidido darse a conocer a través de las nuevas tecnologías, ya que hasta ahora no han tenido página web, y quieren crear una plataforma que sirva tanto de intranet para las personas del colegio, como informativa para los visitantes.



Ilustración 1-1 - Colegio Meinston

### 1.3 Estructura del documento

Este documento presenta una estructura en etapas, común en el desarrollo de un proyecto software. Estas etapas serán las siguientes:

- **Especificación de requisitos** → En esta fase se reúnen todos los requisitos funcionales que debe tener el producto final. En la especificación de requisitos participa activamente el cliente, de modo que pueda aportar sus ideas.
- **Análisis** → El Análisis describe la estructura y funcionalidad del producto mediante diagramas que permiten comprender el sistema y su funcionamiento. Se incluyen diagrama de clases UML y casos de uso, que describen los principales comportamientos del portal.
- **Diseño** → En esta fase se explican los diferentes niveles que componen la arquitectura de la aplicación (nivel de presentación, nivel lógico y nivel de persistencia).
- **Implementación** → En esta etapa, se describen las diferentes tecnologías utilizadas, las herramientas usadas en el desarrollo del proyecto, y se explican detalladamente la implementación y el código asociado.
- **Evaluación y pruebas** → En este apartado se efectúan una serie de pruebas para evaluar el correcto funcionamiento del sistema.

Finalmente el documento terminará con un apartado de conclusiones, bibliografía y anexos, incluyendo estos últimos los diagramas UML, diseños conceptuales de la base de datos, relación de ilustraciones, etc.

## 2. Especificación de requisitos

### 2.1 Introducción

#### 2.1.1 Propósito

La especificación de requisitos tiene como finalidad conocer lo que se demanda y espera obtener de la aplicación, muy importante para el desarrollo de la misma.

#### 2.1.2 Ámbito

El colegio *Meinston* es un internado para niños desde infantil hasta bachillerato, ubicado en los Pirineos. La directiva del colegio ha decidido darse a conocer a través de las nuevas tecnologías en internet y a la vez servir de puente entre profesores y alumnos, con los familiares de estos últimos, de manera que esa comunicación se pueda hacer en tiempo real.

#### 2.1.3 Definiciones, Acrónimos y abreviaturas

**Internado** → Centro de enseñanza donde los niños además de dar clase, viven allí durante el periodo escolar.

**Alumno** → Todas las personas que están matriculadas en el internado con la finalidad de estudiar.

**Profesor** → Todas aquellas personas que trabajan en el internado como docentes de alguna materia.

**Monitor** → Todas aquellas personas que trabajan en el internado como monitor de algún deporte.

**Familiar** → La persona que se responsabiliza de uno o más alumnos del internado.

**Intranet** → Una intranet es una red privada que utiliza tecnología Internet para compartir, dentro de una organización, parte de sus sistemas de información y sistemas operacionales.

**Anónimo** → Toda persona que accede a la web sin identificarse.

**Comunicado** → Aquel mensaje que puede ser enviado:

- Desde un alumno a su familiar.
- Desde un alumno a su tutor.
- Desde el familiar a algún alumno suyo.
- Desde un profesor a un alumno tutelado.

**Observación** → Aquel mensaje que puede ser enviado por el personal del colegio a cualquier alumno o de un alumno al personal del colegio, siempre que este último sea profesor o monitor suyo.

**Informe** → Un informe que un profesor deberá escribir y enviar a cada uno de los familiares de sus alumnos tutelados, informando del progreso, comportamiento y estado del alumno.

### 2.1.4 Referencias

- IEEE Std. 830-1998 Guía del IEEE para la Especificación de Requisitos Software.
- Ejemplos IEEE

### 2.1.5 Visión global

El producto a desarrollar es un portal web para un colegio.

El objetivo final es dar a conocer el colegio y procurar una intranet para alumnos, familiares y personal del colegio.

Los visitantes de la web podrán informarse sobre los aspectos esenciales del colegio.

Para acceder a la intranet, el visitante deberá estar registrado previamente y dado de alta en la base de datos. Para ello se le facilitará una clave de acceso.

## 2.2 Descripción general

### 2.2.1 Perspectiva del producto

La aplicación que desarrollamos es totalmente independiente.

El sitio web podrá ser visitado por cualquier usuario, independiente de la plataforma y del navegador que utilice.

### 2.2.2 Funciones del producto

#### 2.2.2.1 Operaciones permitidas

En este apartado se enumerarán las operaciones permitidas para cada uno de los usuarios.

- Anónimo:
  - Solicitar información.
  - Mostrar información.
- Alumno:
  - Acceder a la intranet mediante su DNI.
  - Listar observaciones.
  - Listar notas.
  - Listar grupo asignado y asignaturas del grupo.
  - Listar deportes a los que está apuntado.
  - Listar comunicados.
  - Escribir comunicado.
  - Borrar comunicado.
  - Inscribirse en deporte.
  - Borrarse de deporte.
  - Listar exámenes.
  - Listar datos personales
  - Consultar tutor.
- Familiar:
  - Acceder a la intranet mediante su clave.
  - Listar alumnos asignados.
  - Ver informe del alumno seleccionado.
  - Listar comunicados del alumno.

- Escribir comunicado al alumno.
- Listar observaciones del alumno.
- Listar notas del alumno.
- Listar exámenes del alumno.
- Listar deportes a los que el alumno está apuntado.
- Modificar datos personales del alumno.
- Profesor:
  - Acceder a la intranet mediante su DNI.
  - Listar asignaturas asignadas.
  - Listar exámenes.
  - Listar alumnos tutelados.
  - Listar comunicados.
  - Escribir comunicados.
  - Escribir informe a los alumnos tutelados.
  - Escribir observación a alumno.
  - Listar observaciones.
  - Modificar nota de alumno.
  - Modificar sus datos personales.
  - Listar sus datos personales.
- Administrador:
  - Listar/consultar todos los datos de la base de datos.
  - Añadir personal.
  - Añadir alumnos.
  - Añadir familiares
  - Asignar claves a los usuarios.
  - Borrar usuarios.
  - Modificar cualquier dato de la base de datos.

### 2.2.3 Características de usuario

Podemos distinguir entre varios tipos de persona que pueden acceder a la web:

- Anónimo: Cualquier persona no identificada que visite la web.
- Usuario registrado: Aquella persona que accede a la intranet mediante un usuario y una clave.
  - Alumno: Aquel que estudia en el colegio.
  - Familiar: Persona responsable de un alumno.
  - Personal:
    - Profesor: Persona que da clase en el colegio.
    - Monitor deportivo: Persona que enseña un deporte en el colegio,
    - Personal sanitario: Persona que trabaja como médico en el colegio.
    - Psicólogo: Persona que trabaja como psicólogo en el colegio.
  - Administrador: Persona encargada del mantenimiento de la web.

### 2.2.4 Restricciones generales

Para acceder a la web no se requiere ningún hardware específico, solamente acceso a Internet y un navegador, ya que el servidor es el que se encarga de gestionar las transacciones sobre la base de datos y de interpretar el código PHP de las páginas.

Solo el administrador, al identificarse tiene posibilidad de modificar cualquier dato de la web o la base de datos. Los demás usuarios registrados (excepto los alumnos) solo podrán modificar sus propios datos personales.

### 2.2.5 Supuestos y dependencias

Deberá haber un correcto funcionamiento de la base de datos y se deberán realizar copias de seguridad periódicamente.

## 2.3 Requisitos específicos

### 2.3.1 Requisitos de interfaces externas

#### 2.3.1.1 Interfaces de usuario

La aplicación se visualizará en la pantalla del usuario, de forma sencilla e intuitiva.

#### 2.3.1.2 Interfaces hardware

En nuestra aplicación no se necesita ningún interfaz hardware específico, puede ser visitada desde cualquier dispositivo que pueda conectarse a internet y navegar.

#### 2.3.1.3 Interfaces software

La aplicación está desarrollada en PHP y JQuery, sobre Windows 7, mediante Dreamweaver CS3 y Eclipse. Su base de datos se basa en MySQL. De este modo puede funcionar sobre cualquier navegador y cualquier sistema operativo.

### 2.3.2 Requisitos funcionales

#### 2.3.2.1 Almacenamiento y mantenimiento de los datos

## 1. Operaciones permitidas para usuario anónimo:

### 1.1. Solicitar información.

**Introducción:** Se rellena un formulario con los datos necesarios para que el colegio pueda ponerse en contacto con el solicitante.

**Entradas:** Datos de contacto.

**Proceso:** Se envía un formulario de contacto.

**Salida:** Datos del solicitante.

### 1.2. Mostrar información.

**Introducción:** Visualizar por pantalla el portal web.

**Entradas:** Dirección de la web.

**Proceso:** Mostrar la web.

**Salida:** Visualización de la web dependiendo del medio desde el que se acceda.

## 2. Operaciones permitidas para Alumno:

### 2.1. Acceder a la intranet mediante su DNI.

**Introducción:** Identificarse mediante su DNI y contraseña.

**Entradas:** DNI y contraseña.

**Proceso:** Conectarse a la base de datos mediante su DNI para acceder a la intranet.

**Salida:** Intranet del alumno.

### 2.2. Listar observaciones.

**Introducción:** Visualizar las observaciones pertenecientes a ese alumno.

**Entradas:** Datos del alumno.

**Proceso:** Consultar observaciones dado un alumno.

**Salida:** Observaciones pertenecientes al alumno.

### 2.3. Listar notas.

**Introducción:** Visualizar las notas relativas al curso actual y el alumno que ha iniciado sesión.

**Entradas:** Datos del alumno.

**Proceso:** Consultar las notas dado un alumno.

**Salida:** Notas del alumno.

### 2.4. Listar grupo asignado y asignaturas del grupo.

**Introducción:** Visualizar el nombre del grupo al que ha sido asignado el alumno, así como las asignaturas pertenecientes al grupo.

**Entradas:** Datos del alumno.

**Proceso:** Consultar grupo y asignaturas del alumno.

**Salida:** Lista de asignaturas del grupo.

### 2.5. Listar deportes a los que está apuntado.

**Introducción:** Visualizar los deportes a los que está apuntado el alumno.

**Entradas:** Datos del alumno.

**Proceso:** Consultar deportes donde el alumno esté matriculado.

**Salida:** Lista de deportes donde el alumno está matriculado.

### **2.6. Listar comunicados del familiar/tutor.**

**Introducción:** Visualizar los comunicados que el alumno ha recibido o enviado a su familiar/tutor.

**Entradas:** Datos del alumno.

**Proceso:** Consultar los comunicados.

**Salida:** Comunicados del alumno.

### **2.7. Escribir comunicado al familiar/tutor.**

**Introducción:** Escribir un comunicado para el familiar/tutor.

**Entradas:** Datos del comunicado.

**Proceso:** Enviar el comunicado al familiar/tutor.

**Salida:** Confirmación del envío.

### **2.8. Borrar comunicado.**

**Introducción:** Borrar un comunicado enviado o recibido.

**Entradas:** Comunicado seleccionado.

**Proceso:** Eliminar la tupla del comunicado seleccionado.

**Salida:** Confirmación del borrado.

### **2.9. Inscribirse en deporte.**

**Introducción:** Inscribir al alumno en el deporte seleccionado.

**Entradas:** Datos del alumno, deporte seleccionado.

**Proceso:** Inscribir al alumno en el deporte.

**Salida:** Confirmación de inscripción.

### **2.10. Borrarse de deporte.**

**Introducción:** Borrar al alumno del deporte seleccionado.

**Entradas:** Datos del alumno, deporte seleccionado.

**Proceso:** Eliminar la tupla del alumno en el deporte seleccionado.

**Salida:** Confirmación de borrado.

### **2.11. Listar exámenes.**

**Introducción:** Visualizar el calendario de exámenes con las fechas de estos.

**Entradas:** Datos del alumno.

**Proceso:** Consultar calendario de exámenes.

**Salida:** Fechas de los exámenes del alumno.

**2.12. Listar datos personales.**

**Introducción:** Visualizar los datos personales del alumno.

**Entradas:** Datos del alumno.

**Proceso:** Consultar datos personales.

**Salida:** Lista de datos personales.

**2.13. Consultar tutor.**

**Introducción:** Visualizar tutor correspondiente al alumno.

**Entradas:** Datos del alumno.

**Proceso:** Consultar tutor del alumno.

**Salida:** Datos del tutor.

**3. Operaciones permitidas para Familiar:**

**3.1. Acceder a la intranet mediante su clave.**

**Introducción:** Mostrar la intranet del familiar.

**Entradas:** DNI del familiar y clave.

**Proceso:** Comprobar usuario y clave y mostrar la intranet.

**Salida:** Intranet del familiar.

**3.2. Listar alumnos asignados.**

**Introducción:** Visualizar los alumnos que el familiar que ha iniciado sesión tiene asignados.

**Entradas:** Datos del familiar.

**Proceso:** Consultar alumnos asignados.

**Salida:** Lista de alumnos asignados.

**3.3. Ver informe del alumno seleccionado.**

**Introducción:** Visualizar el informe del alumno seleccionado.

**Entradas:** Datos del alumno.

**Proceso:** Consultar informe del alumno.

**Salida:** Informe.

### **3.4. Listar comunicados.**

**Introducción:** Visualizar los comunicados enviados o recibidos.

**Entradas:** Datos del familiar.

**Proceso:** Consultar los comunicados.

**Salida:** Lista de comunicados.

### **3.5. Escribir comunicado al alumno.**

**Introducción:** Escribir un comunicado para el alumno seleccionado.

**Entradas:** Datos del alumno, datos del comunicado.

**Proceso:** Enviar comunicado al alumno, introducir nueva tupla.

**Salida:** Confirmación de envío.

### **3.6. Listar observaciones del alumno.**

**Introducción:** Consultar las observaciones enviadas al alumno seleccionado.

**Entradas:** Datos del alumno.

**Proceso:** Consultar observaciones del alumno.

**Salida:** Lista de observaciones.

### **3.7. Listar notas del alumno.**

**Introducción:** Consultar las notas pertenecientes al alumno seleccionado.

**Entradas:** Datos del alumno.

**Proceso:** Consultar las notas del alumno.

**Salida:** Lista de las notas del alumno.

### **3.8. Listar exámenes del alumno.**

**Introducción:** Consultar las fechas de los exámenes del alumno.

**Entradas:** Datos del alumno.

**Proceso:** Consultar calendario de exámenes.

**Salida:** Lista de exámenes.

### **3.9. Listar deportes a los que el alumno está apuntado.**

**Introducción:** Consultar los deportes a los que está apuntado el alumno seleccionado.

**Entradas:** Datos del alumno.

**Proceso:** Consultar deportes.

**Salida:** Lista de deportes.

**3.10. Modificar datos personales del alumno.**

**Introducción:** Modificar los datos del alumno seleccionado.

**Entradas:** Datos del alumno, nuevos datos.

**Proceso:** Modificar las tuplas de los datos del alumno.

**Salida:** Confirmación de modificación, Lista de los nuevos datos.

**4. Operaciones permitidas para Profesor:**

**4.1. Acceder a la intranet mediante su DNI.**

**Introducción:** Acceder a la intranet del profesor mediante su clave de acceso.

**Entradas:** DNI del profesor y clave de acceso.

**Proceso:** Acceder a la intranet y conectarse a la base de datos.

**Salida:** Visualizar la intranet.

**4.2. Listar asignaturas asignadas.**

**Introducción:** Consultar las asignaturas en las que el profesor debe dar clase.

**Entradas:** Datos del profesor.

**Proceso:** Listar asignaturas asignadas al profesor que ha iniciado sesión.

**Salida:** Lista de las asignaturas asignadas.

**4.3. Listar exámenes.**

**Introducción:** Consultar los exámenes a los que debe acudir el profesor.

**Entradas:** Datos del profesor.

**Proceso:** Consultar las fechas de los exámenes del profesor.

**Salida:** Lista de los exámenes.

**4.4. Listar alumnos tutelados.**

**Introducción:** Consultar los alumnos bajo su tutela.

**Entradas:** Datos del profesor.

**Proceso:** Consultar los alumnos que el profesor tiene asignados como tutor.

**Salida:** Lista de alumnos tutelados.

#### **4.5. Listar comunicados.**

**Introducción:** Consultar los comunicados enviados y recibidos.

**Entradas:** Datos del profesor.

**Proceso:** Consultar los comunicados.

**Salida:** Lista comunicados.

#### **4.6. Escribir comunicados a alumnos.**

**Introducción:** Escribir un comunicado a los alumnos tutelados.

**Entradas:** Datos del alumno.

**Proceso:** Enviar comunicado.

**Salida:** Confirmación de envío.

#### **4.7. Escribir informe a los alumnos tutelados.**

**Introducción:** Escribir el informe a los familiares de los alumnos tutelados.

**Entradas:** Datos del alumno.

**Proceso:** Enviar informes.

**Salida:** Confirmación de envío.

#### **4.8. Escribir observación a alumno.**

**Introducción:** Escribir una observación a un alumno.

**Entradas:** Datos del alumno.

**Proceso:** Consultar los comunicados recibidos.

**Salida:** Lista comunicados.

#### **4.9. Modificar nota de alumno.**

**Introducción:** Modificar la nota de un alumno de alguna asignatura del profesor.

**Entradas:** Datos del profesor, datos del alumno.

**Proceso:** Modificar tupla de nota.

**Salida:** Lista de notas, confirmación de cambio.

#### **4.10. Modificar sus datos personales.**

**Introducción:** Modificar los datos personales del profesor.

**Entradas:** Datos del profesor, nuevos datos del profesor.

**Proceso:** Modificar tuplas.

**Salida:** Lista de datos personales, confirmación de cambio.

**4.11. Listar sus datos personales.**

**Introducción:** Consultar los datos personales del profesor.

**Entradas:** Datos del profesor.

**Proceso:** Consultar los datos personales.

**Salida:** Lista de datos personales.

**5. Operaciones permitidas para Administrador:**

**5.1. Buscar alumno por DNI.**

**Introducción:** Consultar los datos de un alumno mediante su DNI.

**Entradas:** DNI del alumno.

**Proceso:** Consultar los datos del alumno

**Salida:** Lista de datos personales del alumno.

**5.2. Buscar alumno por nombre.**

**Introducción:** Consultar los datos de un alumno mediante su nombre.

**Entradas:** Nombre (o apellido) del alumno.

**Proceso:** Consultar los datos del alumno

**Salida:** Lista de datos personales del alumno.

**5.3. Buscar personal por DNI.**

**Introducción:** Consultar los datos de un miembro del personal mediante su DNI.

**Entradas:** DNI del miembro del personal.

**Proceso:** Consultar los datos del miembro del personal.

**Salida:** Lista de datos personales del miembro del personal.

**5.4. Buscar alumno por nombre.**

**Introducción:** Consultar los datos de un miembro del personal mediante su nombre.

**Entradas:** Nombre (o apellido) del miembro del personal.

**Proceso:** Consultar los datos del miembro del personal.

**Salida:** Lista de datos personales del miembro del personal.

#### **5.5. Buscar familiar por DNI.**

**Introducción:** Consultar los datos de un familiar mediante su DNI.

**Entradas:** DNI del familiar.

**Proceso:** Consultar los datos del familiar.

**Salida:** Lista de datos personales del familiar.

#### **5.6. Buscar familiar por nombre.**

**Introducción:** Consultar los datos de un familiar mediante su nombre.

**Entradas:** Nombre (o apellido) del familiar.

**Proceso:** Consultar los datos del familiar.

**Salida:** Lista de datos personales del familiar.

#### **5.7. Buscar familiar por DNI de alumno.**

**Introducción:** Consultar los datos de un familiar mediante del DNI de un alumno.

**Entradas:** DNI del alumno.

**Proceso:** Consultar los datos del familiar de un alumno.

**Salida:** Lista de datos personales del familiar.

#### **5.8. Listar alumnos dado un familiar.**

**Introducción:** Consultar los alumnos asignados a un familiar concreto.

**Entradas:** Datos del familiar.

**Proceso:** Consultar los alumnos.

**Salida:** Lista de alumnos.

#### **5.9. Buscar familiar por nombre de alumno.**

**Introducción:** Consultar los datos de un familiar mediante el nombre de un alumno.

**Entradas:** Nombre (o apellido) del alumno.

**Proceso:** Consultar los datos del familiar.

**Salida:** Lista de datos personales del familiar.

#### **5.10. Consultar número total de alumnos/personal/familiares.**

**Introducción:** Consultar el número total de alumnos/personal/familiares registrados en la base de datos.

**Entradas:** Tipo de dato (Alumno/personal/familiar).

**Proceso:** Consultar cantidad de datos registrados.

**Salida:** Número de datos.

**5.11. Listar todos los alumnos/personal/familiares.**

**Introducción:** Consultar la totalidad de los alumnos/personal/familiares registrados en la web.

**Entradas:** Tipo de dato (Alumno/personal/familiar).

**Proceso:** Listar todos los alumnos/personal/familiares.

**Salida:** Lista de nombres y DNI.

**5.12. Listar alumnos por curso/asignatura/deporte.**

**Introducción:** Consultar los alumnos pertenecientes a un curso/asignatura/deporte en concreto.

**Entradas:** Tipo de dato (Curso/Asignatura/Deporte).

**Proceso:** Listar los alumnos.

**Salida:** Lista de nombres y DNI.

**5.13. Listar alumnos por fecha de ingreso.**

**Introducción:** Consultar los alumnos que se registraron en la web en una fecha concreta.

**Entradas:** Fecha o intervalo de tiempo.

**Proceso:** Listar los alumnos.

**Salida:** Lista de nombres y DNI.

**5.14. Listar alumnos por edad/año de nacimiento.**

**Introducción:** Consultar los alumnos que nacieron el mismo año.

**Entradas:** Edad/año de nacimiento.

**Proceso:** Listar los alumnos.

**Salida:** Lista de nombres y DNI.

**5.15. Listar alumnos por sexo.**

**Introducción:** Consultar los alumnos del mismo sexo.

**Entradas:** Sexo (Hombre/mujer).

**Proceso:** Listar los alumnos.

**Salida:** Lista de nombres y DNI.

**5.16. Listar alumnos tutelados por un profesor.**

**Introducción:** Consultar los alumnos que comparten el mismo tutor.

**Entradas:** Datos del profesor.

**Proceso:** Listar los alumnos.

**Salida:** Lista de nombres y DNI.

**5.17. Consultar tutor dado un alumno.**

**Introducción:** Consultar el tutor del alumno seleccionado.

**Entradas:** Datos del alumno.

**Proceso:** Listar datos del profesor.

**Salida:** Datos del profesor.

**5.18. Listar notas dado un alumno.**

**Introducción:** Consultar las notas de un alumno.

**Entradas:** Datos del alumno.

**Proceso:** Listar notas.

**Salida:** Lista de notas.

**5.19. Listar profesores por curso/asignatura.**

**Introducción:** Consultar los profesores asignados a un curso o a una asignatura.

**Entradas:** Tipo de dato (Curso/Asignatura).

**Proceso:** Listar los profesores.

**Salida:** Lista de nombres y DNI.

**5.20. Listar monitores deportivos por deporte.**

**Introducción:** Consultar el monitor deportivo asignados a un deporte.

**Entradas:** Deporte.

**Proceso:** Listar los monitores deportivos.

**Salida:** Lista de nombres y DNI.

**5.21. Listar deportes/asignaturas.**

**Introducción:** Consultar todos los deportes/asignaturas impartidos en el colegio.

**Entradas:** Base de datos.

**Proceso:** Listar deportes/asignaturas.

**Salida:** Lista de deportes/asignaturas.

**5.22. Añadir personal/alumno**

**Introducción:** Introducir los datos de un nuevo personal/alumno.

**Entradas:** Datos de la persona.

**Proceso:** Crear nuevas tuplas.

**Salida:** Datos introducidos, confirmación de inserción.

**5.23. Añadir familiar**

**Introducción:** Introducir los datos de un nuevo familiar.

**Entradas:** Datos de la persona, datos del alumno.

**Proceso:** Crear nuevas tuplas.

**Salida:** Datos introducidos, confirmación de inserción.

**5.24. Modificar usuarios.**

**Introducción:** Modificar los datos de un usuario.

**Entradas:** Datos del usuario, nuevos datos del usuario.

**Proceso:** Modificar las tuplas.

**Salida:** Confirmación de modificación, nuevos datos.

**5.25. Asignar/modificar claves a los usuarios.**

**Introducción:** Modificar la clave de un usuario.

**Entradas:** Datos del usuario, nueva clave.

**Proceso:** Modificar las tuplas.

**Salida:** Confirmación de modificación, nuevos datos.

**5.26. Borrar usuarios.**

**Introducción:** Borrar los datos de un usuario.

**Entradas:** Datos del usuario.

**Proceso:** Eliminar las tuplas.

**Salida:** Confirmación de borrado.

### **2.3.3 Restricciones de diseño**

Se realizará un diseño general para cualquier situación.

### **2.3.4 Atributos**

#### **2.3.4.1 Seguridad**

La aplicación no mostrará en ningún momento los datos de sus usuarios para protegerlos de posibles fraudes con sus datos personales, para ello se utilizarán los servicios que ofrece SSL.

El usuario que desee utilizar los servicios de la Web tendrá que identificarse dentro de la página con su DNI y clave (asignados al registrarse en la web).

El servidor protegerá todas las transacciones que se puedan realizar tanto del usuario hacia el servidor como del servidor al usuario.

#### **2.3.4.2 Mantenimiento**

El mantenimiento se llevará a cabo por la persona asignada y autorizada previamente, con conocimientos sobre el sistema.

Se realizarán las pertinentes copias de seguridad de la base de datos periódicamente.

### 3. Análisis

El siguiente apartado nos describe la fase de análisis del proyecto. En esta etapa se analiza la aplicación que vamos a crear y se describe su estructura y funcionalidad mediante diagramas UML, con los que entender su funcionalidad nos será más sencillo.

Al final de esta fase tendremos un modelo con los diferentes actores que interactúan con los objetos del sistema a través de diversas relaciones y las acciones que cada uno de ellos podrá llevar a cabo.

#### 3.1 UML

UML es un lenguaje de modelado que utilizaremos para describir nuestro sistema, para detallar y construir los métodos y atributos que estarán presentes en él. Realizaremos varios diagramas para poder entender mejor nuestra aplicación.

##### 3.1.1 Diagrama de clases

Un diagrama de clases describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

En la ilustración 3-1 podemos ver el diagrama de clases de nuestra aplicación.

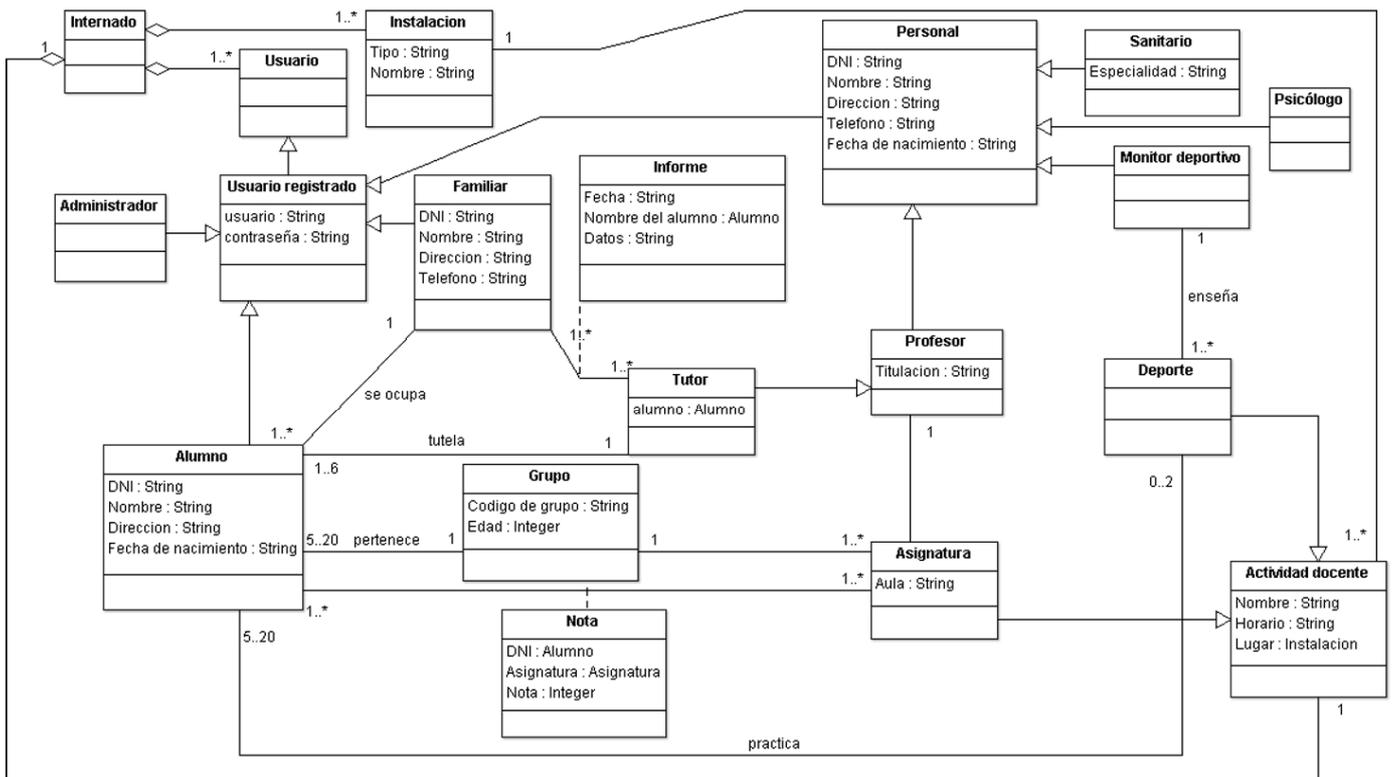


Ilustración 3-1 - Diagrama de clases

De la ilustración 3-1 podemos extraer que el **internado** tiene una serie de **instalaciones** y **usuarios**.

Un **usuario** se considerará anónimo mientras no se identifique en la aplicación. Cuando lo haga pasará a ser **usuario registrado**.

El **administrador** es aquel usuario registrado con capacidad y permisos totales para modificar cualquier elemento de la aplicación.

Del **alumno** conocemos sus datos personales, así como su **familiar** asignado. Un alumno tiene un **tutor** que se ocupa de enviar un **informe** a su familiar. Puede enviar comunicados tanto a su familiar como a su tutor, así como recibirlos de ambos. También puede recibir **observaciones** de cualquier otro miembro del **personal** del colegio.

Un alumno pertenece a un **grupo**, que se compone de varias **asignaturas**. De cada una de estas asignaturas obtendrá una **calificación** que solo podrá poner el **profesor** asignado a esa asignatura. También podrá practicar hasta dos **deportes**, a los que podrá inscribirse o borrarse mediante su **intranet**.

Cada grupo tendrá un código que le identificará por la edad de sus alumnos, que serán como máximo 20.

Cada asignatura pertenecerá a un solo grupo, y cada una de ellas será impartida por un profesor.

Un profesor podrá impartir más de una asignatura y deberá ser tutor de al menos un alumno, pero de cómo máximo 6.

Cada **monitor deportivo** impartirá al menos un deporte, en el que como máximo habrá 20 alumnos. Si no se llegase a 5 el deporte no se impartiría.

Todos los usuarios registrados se identificarán con su **DNI** y **clave** proporcionada por el administrador.

El envío de comunicados, informes y observaciones se realiza de la siguiente manera:

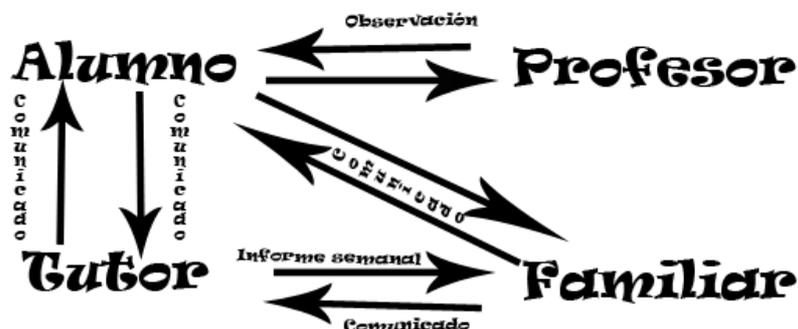


Ilustración 3-2 - Comunicaciones

### 3.1.2 Diagramas de casos de uso

Los diagramas de casos de uso describen el comportamiento de la aplicación, enfatizan lo que debe suceder en el sistema modelado.

En estos diagramas podemos observar varios actores, aquellos que podrán actuar con el sistema y se describe de este modo qué podrán hacer.

En nuestro sistema podemos distinguir varios actores:

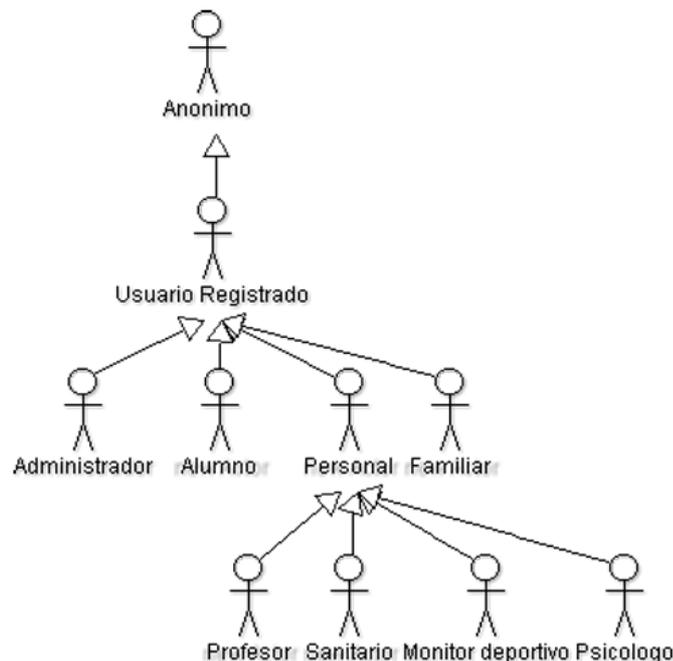


Ilustración 3-3 - Casos de uso, Actores

**Anónimo** es toda persona que accede a la aplicación sin identificarse, este actor podrá navegar por la web, pero no acceder a la intranet. Si podrá, en cambio, cualquier usuario registrado.

Este actor puede a su vez ser **Administrador**, **Alumno**, **Personal** o **Familiar**. Según el papel que desempeñe tendrá unos privilegios u otros. El **Administrador** es aquel con acceso total, que puede modificar por completo la aplicación, es el que se encarga de su mantenimiento.

El **Alumno** accede a la aplicación para comunicarse con el **Personal** o el **Familiar**, para ver sus calificaciones o para inscribirse en algún deporte.

El **Familiar** entrará para comunicarse con el **Alumno** o el **Personal** y para informarse de los progresos de los **Alumnos** que tenga asignados.

El ultimo actor es el **Personal**, este es aquella persona que trabaja en el internado ya sea **Profesor**, **Sanitario**, **Monitor deportivo** o **Psicólogo**. En este proyecto nos ocuparemos del **Profesor**, ya que es el que puede interactuar con los **Familiares** y los **Alumnos**, así como modificar calificaciones, etc.

A continuación mostramos los casos de uso según cada actor.

### 3.1.2.1 Caso de uso: Anónimo

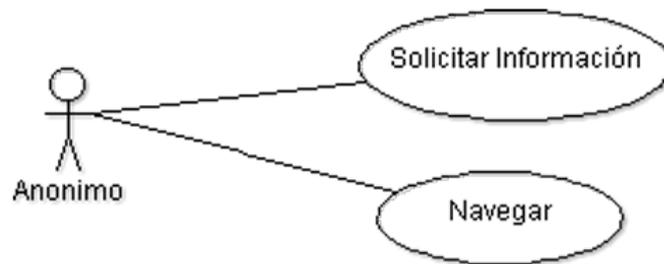


Ilustración 3-4 - Caso de uso: Anónimo

Se puede observar que el usuario anónimo solo puede realizar dos acciones en la aplicación, navegar por la web o solicitar información sobre el internado.

### 3.1.2.2 Caso de uso: Usuario registrado



Ilustración 3-5 - Caso de uso: Usuario registrado

El usuario registrado debe identificarse mediante su usuario y contraseña para poder acceder a la intranet. Cualquier usuario que se identifique podrá consultar sus datos personales, pero solo aquellos con permisos podrán modificarlos.

## 3.1.2.3 Caso de uso: Familiar

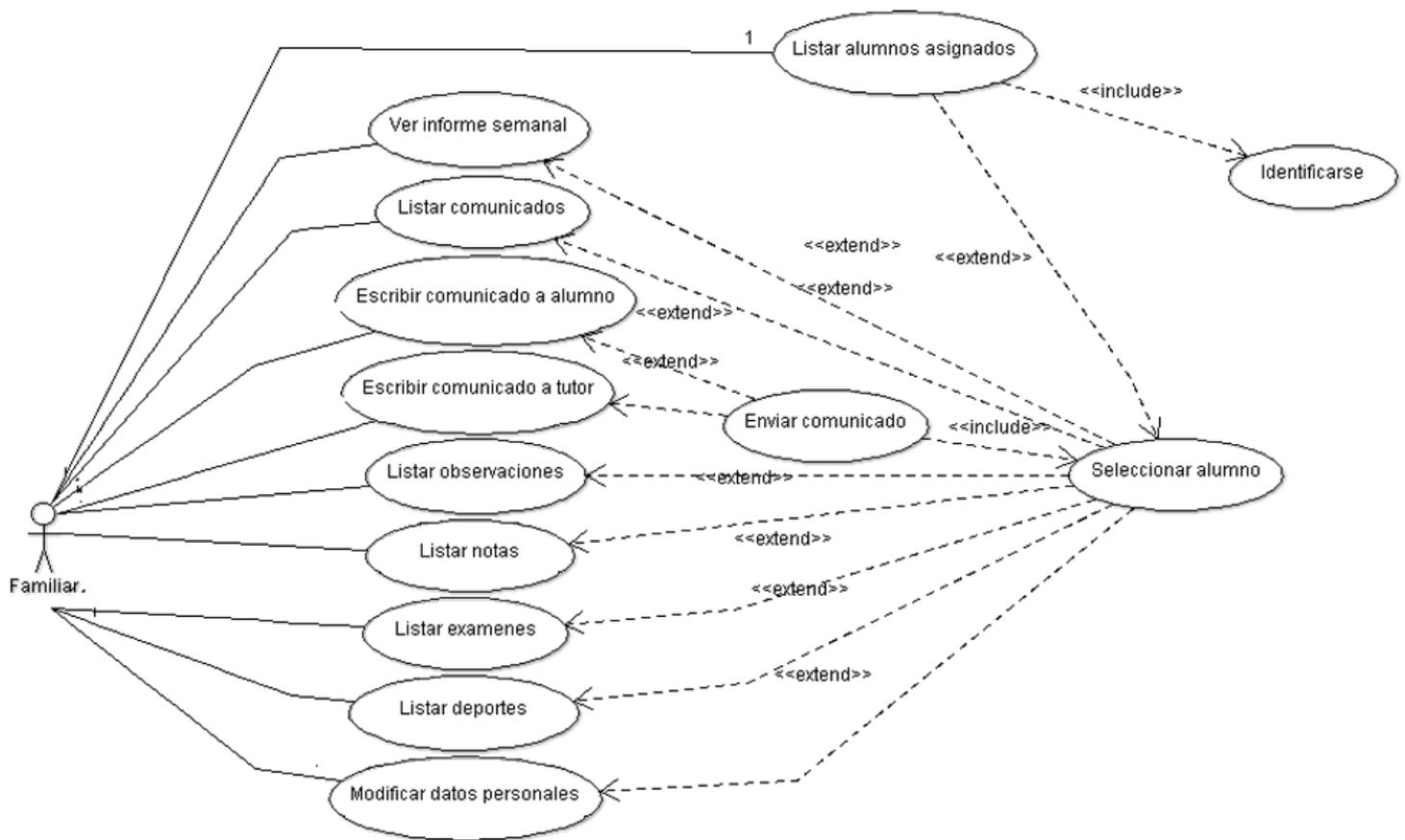


Ilustración 3-6 - Caso de uso: Familiar

El familiar puede ver cualquier información relacionada con sus alumnos, y enviarles comunicados, así como a los tutores de los mismos, para ello tendrán que seleccionar el alumno del que quieren ver la información.

3.1.2.4 Caso de uso: Alumno

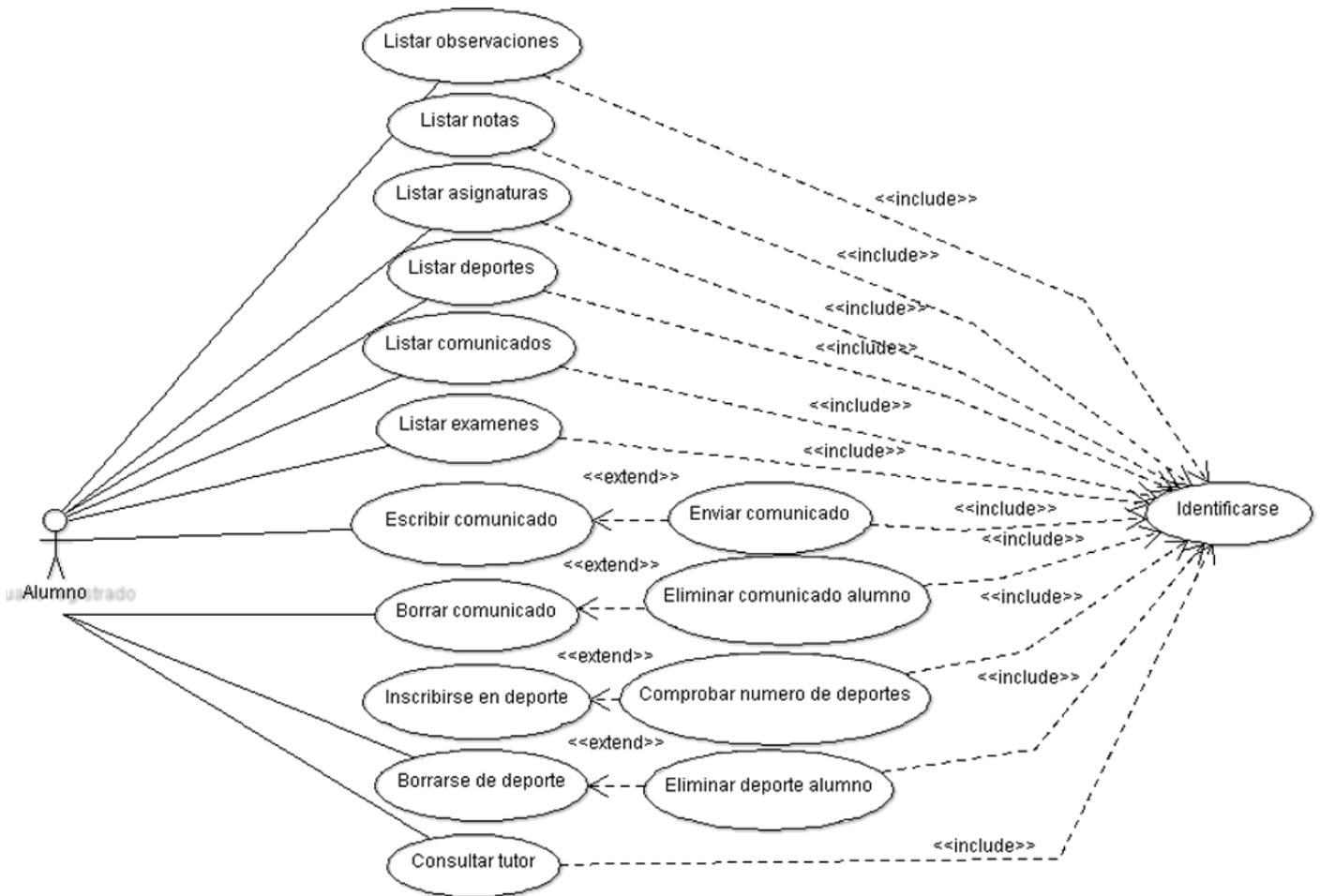


Ilustración 3-7 - Caso de uso: Alumno

El alumno podrá consultar cualquier información referente a él, interactuar con familiar y tutor mediante comunicados y modificar su estado en los deportes.

## 3.1.2.5 Caso de uso: Profesor

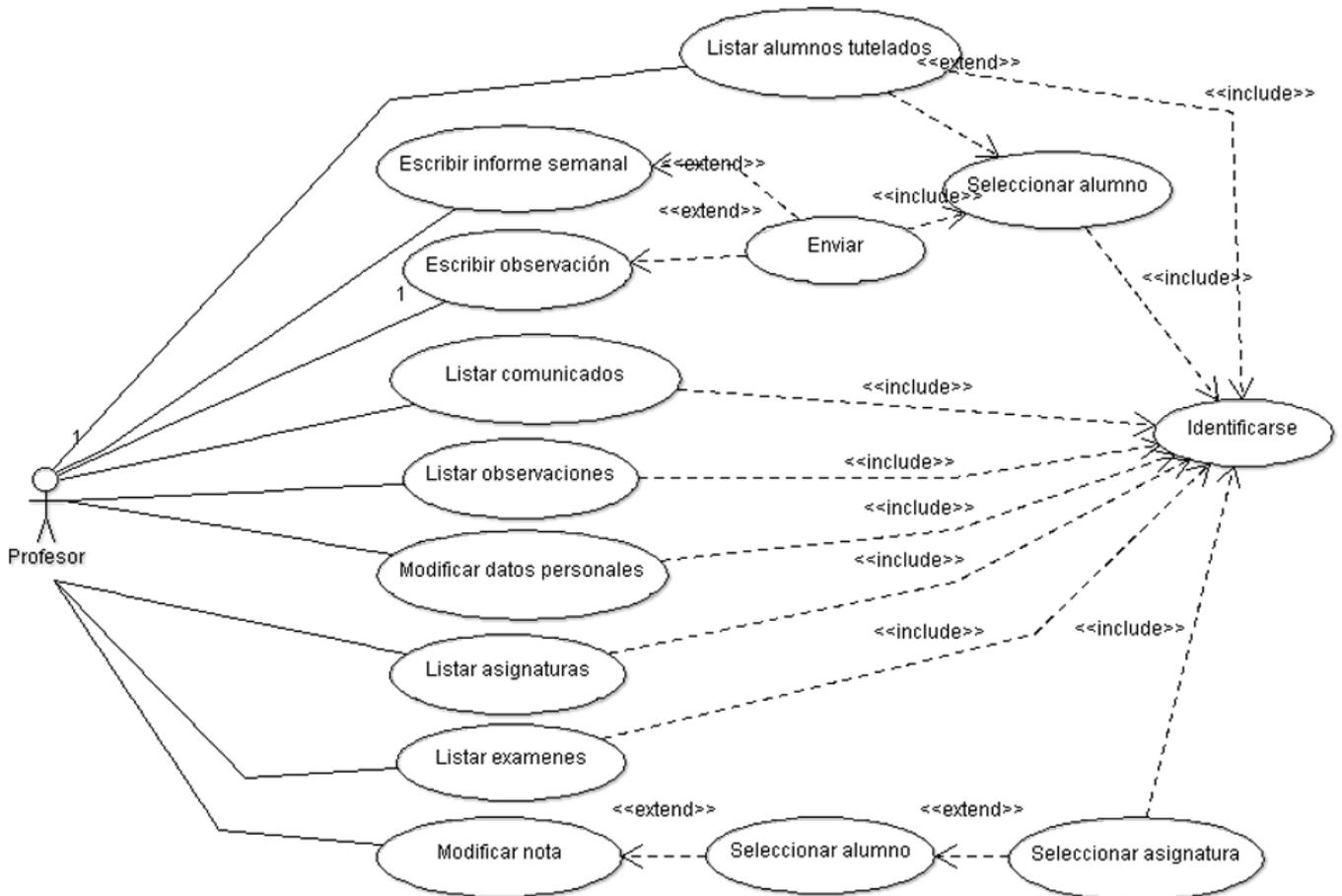


Ilustración 3-8 - Caso de uso: Profesor

El profesor puede listar las asignaturas que imparte, los exámenes de dichas asignaturas, escribir observaciones a sus alumnos e informes semanales a sus alumnos tutelados, modificar notas, etc.

3.1.2.6 Caso de uso: Administrador

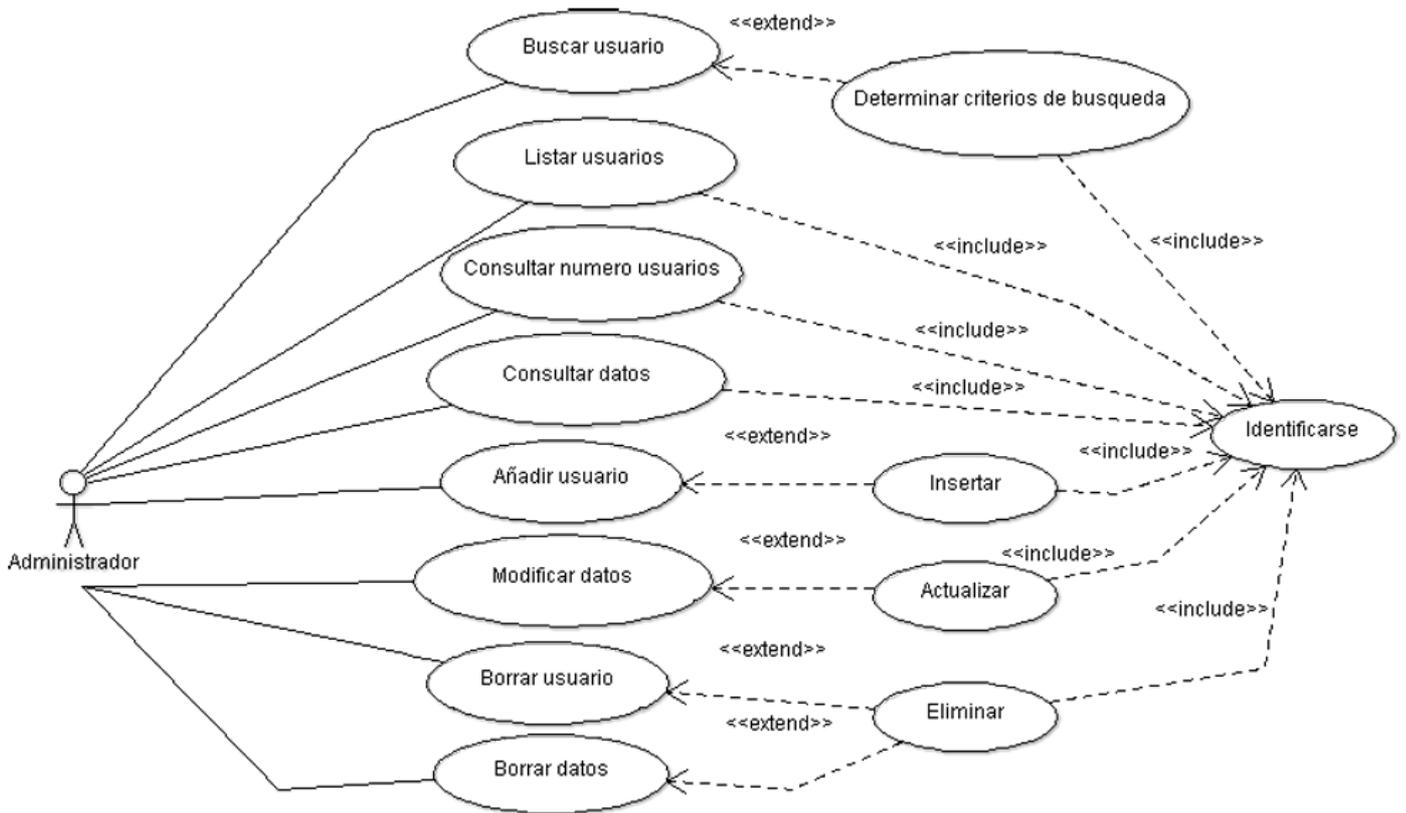


Ilustración 3-9 - Caso de Uso: Administrador

El Administrador tiene permisos totales para modificar, añadir o borrar cualquier elemento de la aplicación.

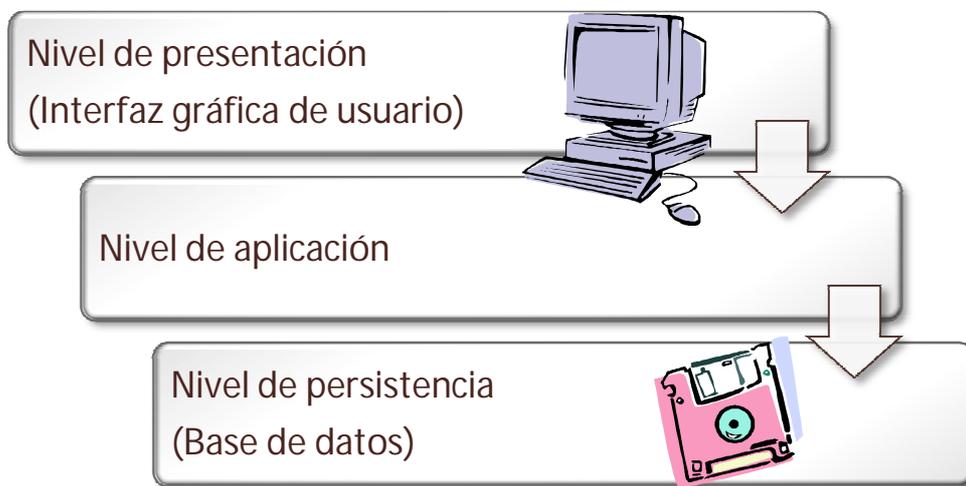
Para una información más detallada de cada uno de los casos de uso, se encontrarán sus descripciones en el Anexo A.

## 4. Diseño

### 4.1 Arquitectura del sistema

Nuestra aplicación está basado en el modelo Cliente / Servidor. Los clientes realizan peticiones Servidor Web que le ofrece la respuesta, de este modo la capacidad de proceso está repartida entre los clientes y los servidores. Mediante esta arquitectura los accesos, recursos y la integridad de los datos son controlados por el servidor, de modo que un cliente no puede acceder a aquellos datos a los que no esté autorizado.

La arquitectura de nuestra web estará estructurada de la siguiente manera:



- **Nivel de presentación** → Esta capa de nuestra arquitectura se encarga de la representación de la información para el usuario final, interactuando con él y comunicándose únicamente con el nivel de aplicación.
- **Nivel de aplicación** → Es donde se ubica el código de los programas, que se ejecutan, recibiendo las peticiones del usuario y enviándole las respuestas tras el proceso.
- **Nivel de persistencia** → En esta capa se encuentran los datos guardados en bases de datos y procesados por el nivel de aplicación.

A continuación se detallan cada uno de los niveles en nuestro proyecto.

#### 4.1.1 Nivel de presentación

El nivel de Presentación está formado por todos los documentos que envía el servidor al cliente y que se muestran al usuario final de la aplicación.

En nuestra aplicación hay dos vistas diferentes de la web. La primera para usuarios en general y la segunda, solo para registrados, la intranet.

##### 4.1.1.1 Interfaz gráfica

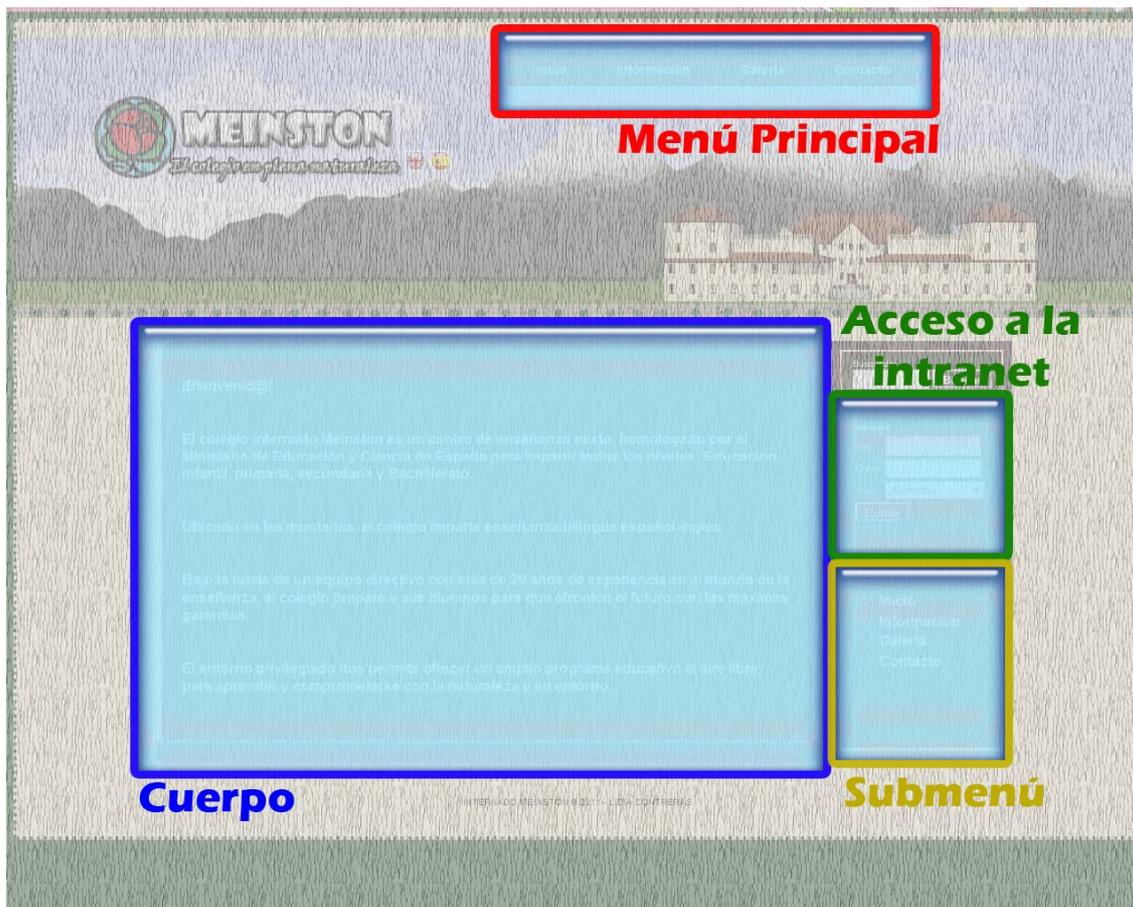


Ilustración 4-1 - Interfaz de usuario, Vista 1

La primera vista de nuestra aplicación, que se corresponde a la página de inicio, galería de imágenes, formulario de contacto e información, se divide en los siguientes apartados:

1. En la parte superior, el logotipo del colegio a la izquierda y el menú de categorías de la web a la derecha (*Ilustración 4-1, Menú principal*). Con este menú podremos acceder a la página de inicio, a la información del colegio, a la galería de imágenes y a la información de contacto.
2. En la parte izquierda de la web tenemos el cuerpo de la misma (*Ilustración 4-1, Cuerpo*), el texto que nos describe aquello que estamos viendo.
3. En la parte derecha podemos encontrar en primer lugar el buscador de la web. Seguidamente vemos el formulario de acceso a la intranet (*Ilustración 4-1, Acceso a la intranet*), donde tendremos que introducir nuestro DNI y contraseña. La última parte es el menú interno de la categoría en la que nos encontramos, si dispone de él, en caso contrario se verá el menú de la web (*Ilustración 4-1, Submenú*).

La segunda vista se corresponde a la intranet, a la que accederán solo los usuarios registrados.



Ilustración 4-2 - Interfaz de usuario, Vista 2

1. El menú de categorías se encuentra a la izquierda de la pantalla, con el podremos acceder a las diferentes funciones de la intranet (*Ilustración 4-2, Menú*). Este menú será diferente según que usuario haya iniciado sesión.
2. En la parte superior podemos ver una cabecera con el logotipo del colegio, el nombre y apellidos de la persona que ha iniciado sesión, así como su tipo de usuario y dos enlaces, para volver a la página inicial y cerrar sesión respectivamente (*Ilustración 4-2, Cabecera*).
3. La parte derecha, bajo la cabecera, es el cuerpo de la página. En este apartado veremos los datos y resultados de las consultas a las bases de datos (*Ilustración 4-2, Cuerpo*).

#### 4.1.2 Nivel de aplicación

Con esta capa definiremos el comportamiento de los objetos que interactuarán en la web. Estos objetos contienen diversos tipos de funciones que se diferencian según su propósito: mostrar información, acceder a la base de datos o procesar datos.

A continuación, se muestran las diferentes acciones que se pueden desarrollar en el sistema, asociando a cada una, los ficheros que contienen las funciones que deben utilizar.

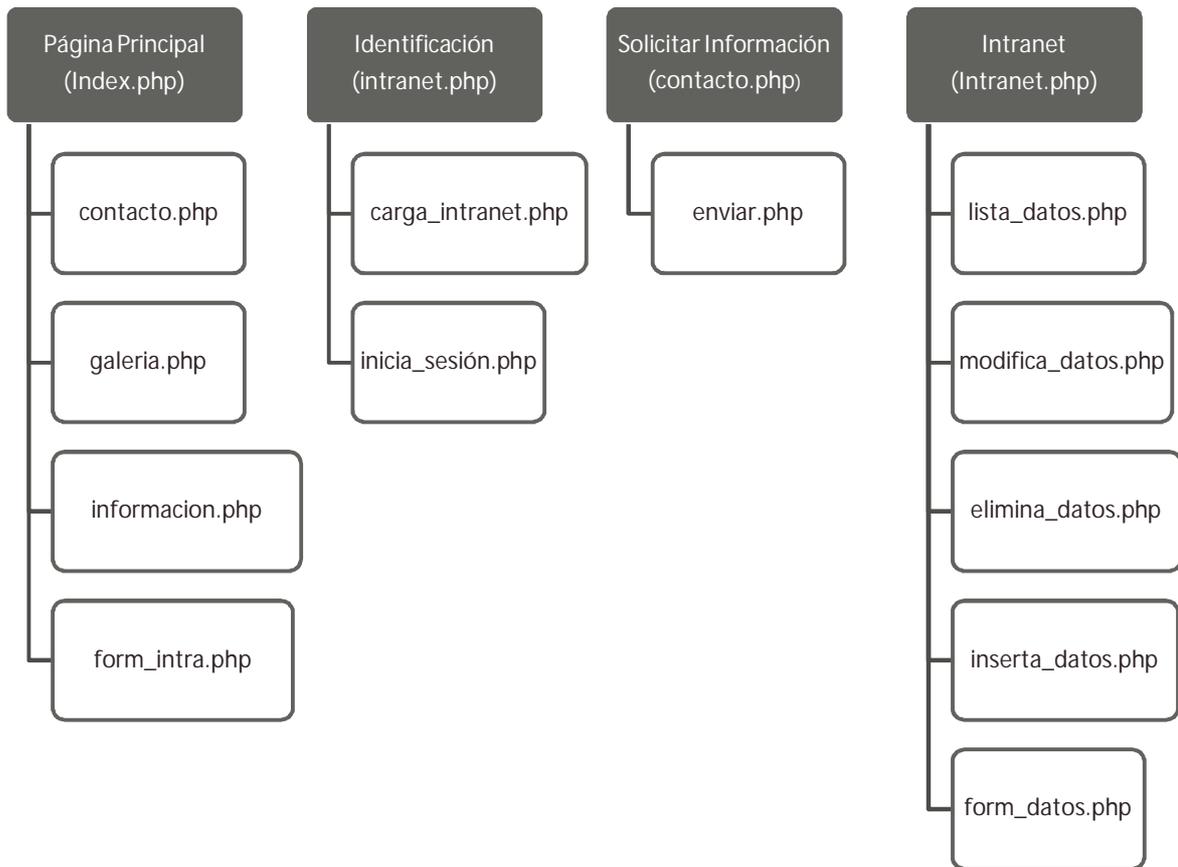


Ilustración 4-3 - Diagrama de componentes

Desde la página principal accederemos a las diversas funciones del portal.

Los archivos descritos en la ilustración 4-3, en el apartado de la página principal, mostrarán sobre todo información, sin acceder a la base de datos.

Sí accederán, sin embargo, los archivos de identificación y aquellos asociados a las funciones de la intranet.

Para mayor detalle en la estructura de archivos, hay más información en el Anexo E.

### 4.1.3 Nivel de persistencia

Es en esta capa donde los datos de nuestra aplicación están almacenados.

#### 4.1.3.1 Base de datos

Para nuestro proyecto hemos creado una base de datos llamada Internado que contendrá en cada una de sus tablas los datos tanto de alumnos, como personal del colegio, como las relaciones entre ellos.

Se muestra a continuación el diagrama entidad-relación que describe las relaciones entre los diferentes usuarios y objetos del sistema y las restricciones entre ellos.

##### 4.1.3.1.1 Diagrama Entidad-Relación

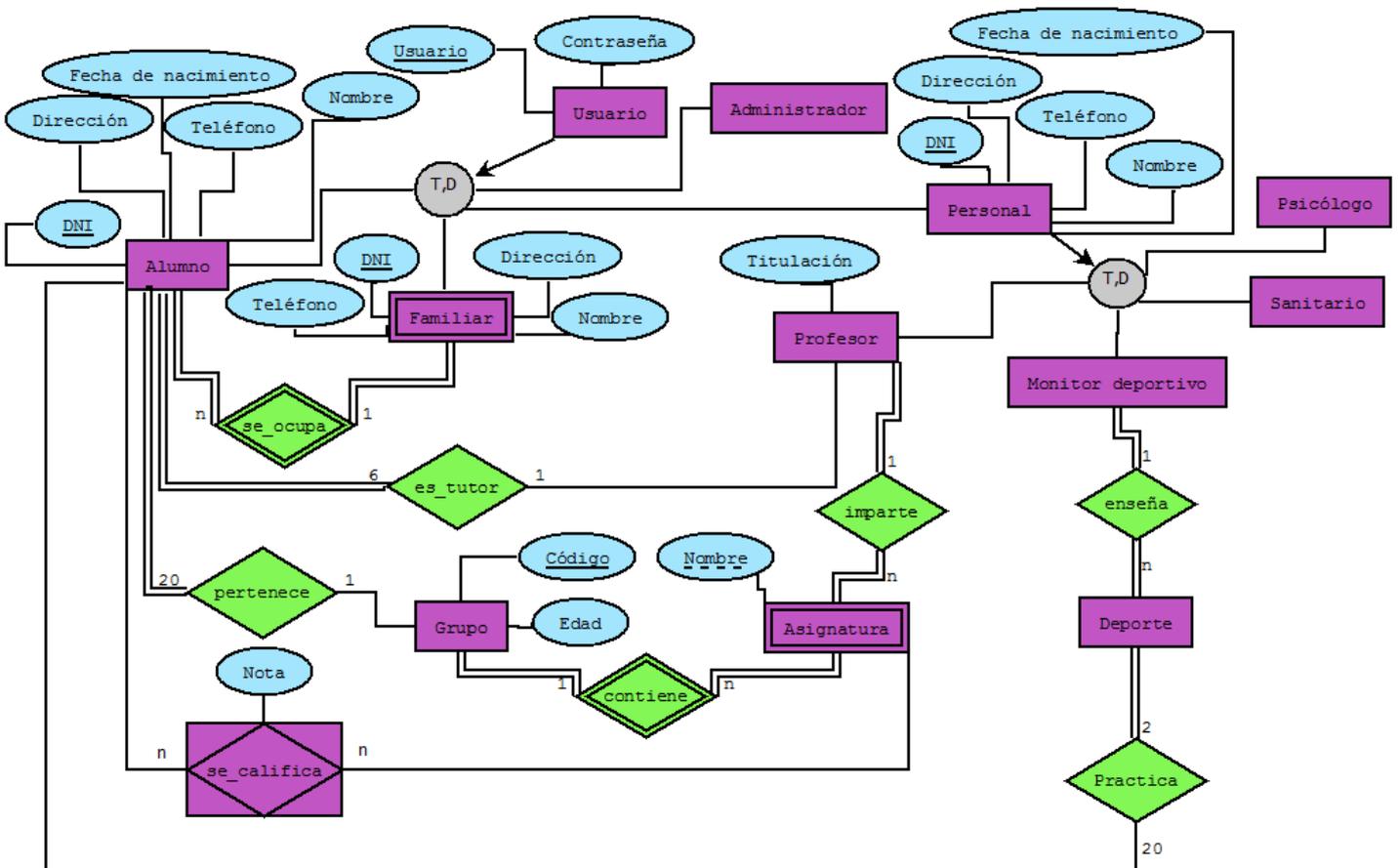


Ilustración 4-4 - Diagrama entidad-relación

Como podemos ver en la ilustración 4-4, el diagrama nos detalla los usuarios que interactuarán con el sistema (Alumno, Familiar, Personal, y administrador) y sus atributos, así como las restricciones a la hora de insertar nuevos datos o modificar los existentes.

*Para una información más detallada tanto el diagrama conceptual como los dominios están definidos en los Anexos B y C respectivamente. En el Anexo D se puede encontrar el ejecutable internado.sql para la creación de la base de datos.*

## 5. Implementación

### 5.1 Tecnologías utilizadas

Para realizar este proyecto se han utilizado varias tecnologías y lenguajes, ya sea para el aspecto visual de la web, como del comportamiento y la dinámica.

Para el aspecto visual de la página se ha utilizado en conjunto **HTML** y **Css**. El primero para la estructura y el segundo para el aspecto.

Para el comportamiento de la web se ha utilizado **PHP** en el lado del servidor y **JQuery (Javascript)** para el lado del cliente. Con PHP nos ocupamos de las acciones de acceso a la base de datos y de envío y recepción de datos desde la web a la base de datos y viceversa, así como de la comprobación de la sesión iniciada y autenticación. Con JQuery hemos conseguido que la interfaz sea dinámica, es decir, que al realizar una petición, por ejemplo hacer clic en un apartado del menú de intranet, se carguen los datos directamente en otra zona de la página, sin tener que recargar toda la web. Con el conjunto de estas dos tecnologías conseguimos que cada usuario tenga su intranet correspondiente. Para la visualización de los datos se ha utilizado un plugin de JQuery llamado **jqGrid**.

Para la base de datos se ha utilizado **MySQL**.

El funcionamiento básico de la **intranet** será el siguiente:

- Se creará mediante PHP el menú lateral y la cabecera y mediante un javascript, cargado en el *document.ready* de la página, se le darán acciones al menú.
- Mediante este último javascript, y en base a esas acciones, al hacer clic en cada uno de los apartados del menú se cargará dinámicamente un script PHP en el cuerpo de la página.
- Este último script contendrá código JQuery que cargará una tabla con los datos. Para ello se utilizarán diferentes scripts PHP:
  - o Tipo *lista\_xx.php*, donde *xx* serán los datos a cargar, devolverá a la página los datos procedentes de la base de datos.
  - o Tipo *modifica\_xx.php / inserta\_xx.php*, mandará los datos a la base de datos.
  - o Tipo *elimina\_xx.php*, borrará de la base de datos lo que se pida.
  - o Tipo *form\_xx.php*, se podrá pedir si se necesitan datos del usuario para su modificación. Estos archivos se cargarán en un diálogo de JQuery y se mostrarán por pantalla.

Veamos un poco más detalladamente cada una de ellas.

### 5.1.1 HTML

El HTML, Hyper Text Markup Language (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto utilizado normalmente en la *www* (World Wide Web). Fue creado en 1986 por el físico nuclear Tim Berners-Lee que cual tomo dos herramientas preexistentes: El concepto de Hipertexto (Conocido también como link o ancla) que permite conectar dos elementos entre si y el SGML (Lenguaje Estándar de Marcación General) que sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML es un sistema de etiquetas que no requiere compilador.

Estos documentos pueden ser mostrados por los navegadores.

### 5.1.2 CSS

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

CSS funciona a base de reglas. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne.

```
h1 {color: red;}
```

### 5.1.3 PHP

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente *PHP Tools*, o, *Personal Home Page Tools*). Fue creado originalmente por Rasmus Lerdorf en 1994.

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno.

### 5.1.4 JQuery (Javascript)

JavaScript es un lenguaje de programación interpretado.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

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

### 5.1.5 MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.

## 5.2 Herramientas utilizadas

Para la implementación de la web y la base de datos se han utilizado diferentes aplicaciones o herramientas que son las siguientes:

- Para la creación de los Scripts de PHP y Javascript se ha utilizado *Eclipse for PHP Developers*, un entorno de desarrollo integrado de código abierto multiplataforma, muy útil cuando se crean diferentes funciones en diferentes archivos.
- Para la creación de HTML y Css se ha utilizado *Adobe Dreamweaver CS3*.
- La base de datos MySQL, ha sido mantenida y modificada localmente mediante *Navicat Lite*, un software de interfaz muy sencilla e intuitiva. Para la manipulación remota se ha utilizado *PhpMyAdmin*, una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet.
- Para las pruebas y comprobaciones de la web se han utilizado distintos navegadores (*Firefox, Internet Explorer, Chrome y Opera*) y un complemento llamado *Firebug* para la comprobación de errores en el envío de datos entre los scripts PHP, por ejemplo.

## 5.3 Implementación detallada

A continuación se mostrará la descripción detallada de la implementación del proyecto, con ejemplos de código.

Se seguirá la arquitectura de diseño utilizada, basándonos en el diagrama de componentes mostrado en la ilustración 4-3.

### 5.3.1 Capa de presentación

Para la capa de presentación se ha optado por utilizar HTML para la estructura, Css para cambiar el aspecto de la página y código javascript para darle dinamismo.

#### 5.3.1.1 Página principal

La estructura de la página principal, mostrada en la ilustración 4-1, es común para el índice (index.php), la galería de imágenes (galería.php), la información del colegio (información.php) y el formulario de contacto (contacto.php).

La página se compone de contenedores html, donde uno de ellos será sustituido por la información a mostrar según el apartado en el que nos encontremos. El código es el siguiente:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>_* Internado Meinston *_</title>
<link rel="shortcut icon" href="favicon.ico" />
<link href="estilos/estilo.css" rel="stylesheet" type="text/css" media="screen" />
<script src="javascript/jquery-1.5.1.min.js" type="text/javascript"></script>
</head>
<body>
<div id="contenedor">
  <div id="cabecera">
    <div id="logo"></div>
    <div id="menu">
      <a href="index.php">Inicio</a>
      <a href="informacion.php">Informaci&oacute;n</a>
      <a href="galeria.php">Galer&iacute;a</a>
      <a href="contacto.php">Contacto</a>
    </div>
    <a href="#"></a>
    <a href="#"></a>
  </div>
  <div id="cuerpo">
    <div id="contenido">
      <textarea id="texto" class="texto" readonly="readonly">
        ...
      </textarea>
    </div>
    <div id="lateral">
      <div id="buscador">
        <p> Búsqueda: </p>
        <form id="formbus" name="formbus" action="#" method="post">
          <input type="text" class="textbusqueda" size="8" id="busqueda"
name="busqueda" />
          <input type="submit" class="botonbusqueda" value="Buscar" id="botonbus"
name="botonbus" />
        </form>
      </div>
      <div id="acceso_intranet">
        <? include "scripts/form_intra.php"; form_intra();?>
      </div>
      <div id="submenu">
        <p> Men&uacute;:: Inicio </p>
        <ul>
          <li> <a href="index.php"> Inicio </a> </li>
          <li> <a href="informacion.php"> Informaci&oacute;n </a></li>
          <li> <a href="galeria.php"> Galer&iacute;a </a></li>
          <li> <a href="contacto.php"> Contacto </a> </li>
        </ul>
      </div>
    </div>
  </div>
<div id="pie"> <p> Internado Meinston © 2011 - Lidia Contreras </p> </div>
</div>
</body>
</html>

```

Las áreas comunes a todas las páginas que comparten esta estructura (áreas sin colorear en el código), se componen básicamente de código HTML y css.

La zona coloreada de amarillo se corresponde al contenido de la página (Cuerpo, en Ilustración 4-1), aquella información que se mostrará al usuario, y que por tanto cambiará según el tema que estemos viendo. En la página de inicio y la de información del colegio, se sustituirá por un textarea con información. Para la galería se usarán div HTML que contendrán imágenes y para la página de contacto se utilizará un formulario HTML.

En estos últimos casos se utiliza javascript para dar dinamismo a la presentación de imágenes y texto. En el caso de la página de información se usa el siguiente código:

```
<div id="contenido">
  <textarea id="principios" class="texto" readonly="readonly">
    - PRINCIPIOS -
  . . .
  </textarea>
  . . .
</div>
```

El primer textarea se mostrará por defecto al cargar la página. Los demás estarán ocultos mediante `style="display:none`, que nos asegurará que, en un principio, no sean visibles. A continuación añadimos el dinamismo mediante JQuery, llamando al método `submenu_informacion()` al cargar la página.

```
<script type="text/javascript">
$(document).ready ( submenu_informacion() );
</script>
```

```
function submenu_informacion(){
  $("#submenu a").click ( function(evento){
    if( $(this).attr("href") == "principios" )
    {
      evento.preventDefault();
      $("#caracteristicas").hide();
      $("#deportes").hide();
      $("#ins_deportivas").hide();
      $("#ins_academicas").hide();
      $("#residencia").hide();
      $("#principios").fadeIn("slow");
    }
    else if( $(this).attr("href") == "caracteristicas" )
    {
      . . .
    }
  });
}
```

Para cambiar entre una información u otra, se utiliza el evento click sobre los enlaces del submenú. Cada elemento del submenú tiene su propio evento y sabremos cual es por su atributo href (`$(this).attr("href") == "principios"`). Al crearse el evento haremos que no vaya al enlace (`evento.preventDefault()`), sino que oculte todos los demás textareas (`.hide()`) y muestre el que nos pide el evento, con efecto de difuminado (`fadeIn("slow")`).

En el caso de la galería de imágenes, utilizamos un complemento de JQuery llamado Slideviewer, que nos crea una vista de las imágenes en la que podemos pasar de una a otra simplemente haciendo click en ellas.

Para conseguir este efecto creamos el siguiente código:

```
<div id="contenido" class="galeria">
    . . .
    <div id="contenedor_exterior" style="display:none;">
        <div id="galeria_exterior">
            <ul>
                <li></li>
                <li></li>
            </ul>
        </div>
    </div>
    . . .
</div>
```

Cada una de las secciones de la galería se implementa como un div oculto, que contiene una lista de las imágenes a mostrar.

```
<script type="text/javascript">
$(document).ready ( function(){
    $("#submenu a, #ini a").click ( function(evento){
        if( $(this).attr("href") == "exterior" )
        {
            evento.preventDefault();
            $("#ini").css("display", "none");
            $("#contenedor_interior").css("display", "none");
            $("#contenedor_depor").css("display", "none");
            $("#contenedor_exterior").empty().html("<div
id=\"galeria_exterior\" class=\"svw\"><ul><li><img
src=\"imagenes/imagenes/2.png\" alt=\"Vista aérea\" /></li><li><img
src=\"imagenes/imagenes/3.png\" alt=\"Fuente\"
/></li></ul></div>").css("display", "block");
            $("#galeria_exterior").slideDown();
        }
        . . .
    }); });</script>
```

Cada vez que se hace click en cada una de las secciones de imágenes, se ocultan las otras galerías (.css("display", "none")) y el contenedor a mostrar, se vacía y se rellena con la propia galería (.empty().html). Esto se hace porque si no, Slidewiever acumula las imágenes y se repiten. Una vez creada de nuevo, se le pasa al método para que la galería sea mostrada (.slideDown()).

En la página de contacto también se utiliza javascript para validar los datos en el momento. Esto lo conseguimos con el evento `blur` de JQuery. Cuando un apartado del formulario pierda el foco, se analizará su contenido y se informará si hay algún error o se ocultará si no lo hay.

```
function valida_contacto()
{
    $("#descripcion").keypress(function(event) {
        $("#caracteres").html( ($(this).val().length)+1 );
        return ( (event.which == 8) || (event.which == 9) ||
        ($(this).val().length < 249) )
    });
    $("#nombre").blur(function(){
        if( $("#nombre").val()==" " ) $("#td_nombre").show();
        else $("#td_nombre").hide();
    });
    . . .
}
```

El efecto de contar letras en el textarea del formulario se consigue mediante el código coloreado de amarillo. Cada vez que se genere el evento `keypress` en el contenido del textarea se contará el número de caracteres.

En el caso de la página de información y la galería, el submenú (Submenú, en Ilustración 4-1) coloreado de azul en el código, contendrá los diferentes apartados que se pueden ver. En la página inicial y la de contacto, en cambio, contendrá el menú de la web.

El código coloreado de verde se corresponde al formulario de acceso a la intranet, que cambiará si el usuario ha iniciado la sesión previamente, para mostrar sus datos.

### 5.3.1.2 Intranet

La estructura de la intranet, mostrada en la ilustración 4-2, es común para cualquier usuario que inicie sesión, solo cambiará el contenido del menú según el tipo de usuario que sea.

Si se ha iniciado sesión, la intranet mostrará el menú mediante php y javascript del siguiente modo:

- Mediante php se creará la estructura de la página en html según el usuario:

```
switch($_SESSION['tipo'])
{
    case 1:
    {
        $sql_alumno="SELECT Nombre, Apellidos
        FROM alumno
        WHERE NumExp LIKE '". $_SESSION['usuario']. "'";
        $select_alumno=mysql_query($sql_alumno, $conexion) or
        die("Problemas en el select de alumno:".mysql_error($conexion));
        if( mysql_num_rows($select_alumno) != 0 ) //Existe el alumno
        {
            menu_alumno();
            $reg=mysql_fetch_array($select_alumno);
            echo '
            <div id="cabecera">
            <h5>Bienvenid@ :
            ' . $reg['Nombre'] . ' ' . $reg['Apellidos'] . ' (ALUMNO)</h5>
            <h5>
            <a href="index.php">Volver a la p&aacute;gina
            principal</a> / <a href="scripts/cierra_sesion.php">Cerrar
            Sesi&oacute;n</a>
            </h5>
            </div>
            <div id="contenedor">
            </div> ' ;
        }
        else sesion_tipo_incorrecto();
    } //1
    break;
```

- Mediante php se crearán los enlaces del menú (`menu_alumno()`).

```
function menu_alumno()
{
    echo '
    <div id="menu">
    <h4 class="titulo_menu" id="titulo_menu" name="menu_alumno">MENU -
    ALUMNO<hr /></h4>
    <div class="bloque">
    <h5>Datos Personales<hr /></h5>
    <ul>
    <li id="li_datos"><a> Datos personales </a></li>
    </ul>
    </div>
    . . .
    ';
```

- Para crear la carga de cada página del menú se usa JQuery. De este modo solo tendrá que modificar el div contenedor de la intranet, insertando en él la información requerida.

```
if( $("#titulo_menu").attr("name") == "menu_alumno" )
{
    $("#contenedor").html("<h1>ALUMNO</h1>");
    $("#li_datos").click( function() { carga_intranet_alumno_datos(); } );
    . . .
}
```

- Para el contenido de cada uno de los enlaces, se ha utilizado un complemento de JQuery llamado JqGrid, que ofrece un aspecto de tablas dinámicas para mostrar datos.

### 5.3.2 Capa de aplicación

Para la capa de aplicación se ha utilizado PHP y javascript. Veámoslo según las diferentes funciones presentes en la web.

#### 5.3.2.1 Inicio de sesión

Para el inicio de sesión se ha utilizado la variable `$_SESSION['usuario']` de PHP, a la que inicializamos con el número de expediente del usuario que la inicia.

Para iniciar sesión es necesario estar en una de las páginas principales o en la propia intranet sin tener sesión ya iniciada, y rellenar los datos de acceso (DNI y contraseña).

```
<div id="acceso_intranet">
<? include "scripts/form_intra.php"; form_intra();?>
</div>
```

La función `form_intra()` comprueba si hay sesión iniciada. En caso negativo devolverá el formulario de acceso.

```
<form id="formintra" name="formintra" action="intranet.php" method="post">
  <table>
    <tr><th colspan="2"> Intranet: </th></tr>
    <tr><td> DNI: </td><td><input type="text" class="dni" size="11"
id="dni" name="dni"/></td></tr>
    <tr><td> Clave: </td><td><input type="password" class="pass" size="11"
id="pass" name="pass"/> </td></tr>
    <tr><td> Tipo: </td><td><select name="tipo">
      <option value="1">Alumno</option>
      <option value="2">Personal</option>
      <option value="3">Familiar</option>
      <option value="4">Administrador</option>
    </select>
    </td></tr>
    <tr><td colspan="2"> <input type="submit" class="botonintra"
value="Entrar" id="botonintra" name="botonintra" /></td></tr>
  </table></form>
```

Una vez que se pulse el botón de entrar, el servidor comprobará los datos mediante `php` y devolverá la respuesta.

```
if ( (isset($_POST['dni']) && $_POST['dni']!="") && (isset($_POST['pass']) &&
$_POST['pass']!="") && (isset($_POST['tipo']) && $_POST['tipo']!="") ) //No
hay sesión, pero hay datos por POST, iniciar la sesión
{
    $dni=$_POST['dni'];
    $pass=$_POST['pass'];
    $tipo=$_POST['tipo'];
    $sql_user_pass="SELECT *
FROM usuario
WHERE Usuario LIKE ' ".$dni." '
AND Contraseña LIKE ' ".$pass." '";
    $select_user_pass=mysql_query($sql_user_pass, $conexion) or
die("Problemas en el select:".mysql_error($conexion));
    if( mysql_num_rows($select_user_pass) != 0 ) //el usuario y la
clave coinciden con la bd
    {
        switch($tipo)
        {
            . . .
        }
    }
    else
    {
        sesion_incorrecta();
    }
}
//if
else //Si no hay datos por POST, error no deberia estar aqui o no a rellenado
todos los datos
{
    no_sesion();
}
//else
```

Si los datos no son correctos o no están rellenos nos devolverá otra vez el formulario de acceso mostrándonos el error.

Si son correctos, carga la intranet como hemos visto en la capa de presentación e inicia la sesión mediante `session_start()`, inicializando la variable `$_SESSION['usuario']`.

### 5.3.2.2 Cerrar sesión

Para salir de la sesión en la que nos encontremos, simplemente se vacían las variables creadas, se destruye la sesión y se redirige al usuario a la página de inicio.

```
<?php
    Session_start();
    $_SESSION['usuario']="";
    $_SESSION['tipo']="";
    $_SESSION['dni']="";
    Session_destroy();

    header("Location:../index.php");

?>
```

### 5.3.2.3 Intranet

Explicaremos ahora la implementación del contenido de la intranet mediante JqGrid.

#### 5.3.2.3.1 Listar datos

Cualquier usuario que acceda a la intranet puede ver sus datos personales, los de sus alumnos tutelados o de los que se responsabiliza si es el familiar, etc. Del mismo modo el administrador puede listar cualquier dato sobre el colegio. Para ello JqGrid tiene la funcionalidad de mostrar tablas de datos personalizadas.

```
function carga_intranet_alumno_datos()
{
    $("#contenedor").html("<h5 style=\"text-align:center\">DATOS
PERSONALES:</h5><table id=\"list\"></table><div id=\"pager\"></div>");
    $("#list").jqGrid({
        url:'scripts/intranet/lista_datos.php?tipo=alumno',
        datatype: "json",
        colNames:['NumExp','DNI','Nombre','Apellidos','Fecha nacimiento',
'Direccion','Telefono','Grupo','Familiar','Tutor'],
        colModel:[
            {name:'NumExp',index:'NumExp',width:70,align:"center"},
            {name:'DNI',index:'DNI',width:70,align:"center"},
            {name:'Nombre',index:'Nombre',width:100,align:"center"},
            {name:'Apellidos',index:'Apellidos',width:100,align:"center"},
            {name:'Fecha_nacimiento',index:'Fecha_nacimiento',width:80,
align:"center",sorttype:"date"},
            {name:'Direccion',index:'Direccion',width:100,align:"center"},
            {name:'Telefono',index:'Telefono',width:80,align:"center"},
            {name:'Grupo',index:'Grupo',width:70,align:"center"},
            {name:'Familiar',index:'Familiar',width:120,align:"center"},
            {name:'Tutor',index:'Tutor',width:120,align:"center"}
        ],
        height:"auto",
        width:"auto",
        pager: '#pager',
        toolbar: [true,"bottom"]
    });
    $("#list").jqGrid('navGrid','#pager',{search:false,edit:false,add:false,
del:false});
} //datos
```

Esta estructura se repetirá para cada uno de los contenidos de la intranet. Siempre se listarán los datos para verlos, modificarlos o borrarlos. El mecanismo utilizado es el siguiente:

- Se crean una tabla y un div vacíos que le pasamos a JqGrid para cargar los datos (color amarillo en el código).
- Para la carga de los elementos generados, en la zona coloreada de azul del código, estos serán devueltos mediante JSON por un script, que le pasamos a JqGrid en la zona verde del código.

El script tendrá siempre la siguiente estructura:

```
$sql = "
    SELECT A.Id, A.NumExp, A.DNI, A.Direccion, A.Telefono, A.Nombre,
    A.Apellidos, A.Fecha_nacimiento, A.Grupo, P.Nombre AS Tutor_nombre,
    P.Apellidos AS Tutor_apellidos, F.Nombre AS Familiar_nombre, F.Apellidos AS
    Familiar_apellidos
    FROM `alumno` A
    LEFT JOIN `familiar` F ON F.NumExp = A.Familiar
    LEFT JOIN `personal` P ON P.NumExp=A.Tutor
    WHERE 1 AND A.NumExp LIKE '".$_SESSION['usuario']."'
    ";
$result = mysql_query($sql,$conexion) or die("Couldn t execute
query.".mysql_error());
$row = mysql_fetch_array($result,MYSQL_ASSOC) ;

$response->rows[0]['id']=1;
$response->rows[0]['cell']=array(
    utf8_encode($row["NumExp"]),
    utf8_encode($row["DNI"]),
    utf8_encode($row["Nombre"]),
    utf8_encode($row["Apellidos"]),
    $row["Fecha_nacimiento"]!=NULL?date("d/m/Y",strtotime($row["Fecha_nacimi
ento"])):"- ",
    utf8_encode($row["Direccion"]),
    utf8_encode($row["Telefono"]),
    utf8_encode($row["Grupo"]),
    utf8_encode($row["Familiar_nombre"].' '.$row["Familiar_apellidos"]),
    utf8_encode($row["Tutor_nombre"].' '.$row["Tutor_apellidos"])
);
```

Estos tipos de script harán siempre las mismas acciones. Su misión es obtener de la base de datos los elementos requeridos y devolverlos (en el orden que ponemos en javascript, color azul del código) a la tabla de JqGrid (texto coloreado en amarillo).

Para que el resultado sea devuelto mediante JSON, ponemos al principio del script:

```
header("Content-type: text/json");
```

Y al final:

```
echo json_encode($response);
```

## 5.3.2.3.2 Insertar Datos

Tanto para insertar datos, como para modificarlos, en algunos casos es necesario obtener más datos de los que nos ofrece la tabla, o nos es útil tener generado un formulario que podamos mostrar para que el usuario rellene nuevos datos.

Esto lo conseguimos creando unos scripts PHP que llamaremos form\_tipo, en el que definimos los formularios necesarios para el tipo de dato que estamos manipulando. Por ejemplo, para la inserción de asignaturas creamos el siguiente formulario en form\_asignaturas.php:

```
<!-- INSERTA ASIGNATURA -->
<form id="form_admin_inserta_asignatura" style="display:none; font-size:10px;">
  <table>
    <tr><th colspan="2"> Rellena los siguientes datos: </th></tr>
    <tr><td> Nombre*: </td>
      <td><input type="text" id="form_admin_inserta_asignatura_nombre"
name="form_admin_inserta_asignatura_nombre"></input></td></tr>
    <tr><td> Grupo*: </td>
      <td><select id="form_admin_inserta_asignatura_grupos"
name="form_admin_inserta_asignatura_grupos">

<?   $sql   =   "SELECT Codigo, Edad FROM grupo WHERE 1";

      $select=mysql_query($sql, $conexion) or die("Problemas :
".mysql_error($conexion));

      while($res=mysql_fetch_array($select))
        echo '<option value="'. $res['Codigo']. '>' . $res['Codigo']. ' -
Edad: ' . $res['Edad']. '</option>';
?>
        </select></td></tr>
    <tr><td> Profesor: </td>
      <td><select id="form_admin_inserta_asignatura_profesores"
name="form_admin_inserta_asignaturas_profesores">
        <option value=""></option>
<?   $sql   =   "SELECT NumExp FROM profesor WHERE 1";

      $select=mysql_query($sql, $conexion) or die("Problemas :
".mysql_error($conexion));

      while($res=mysql_fetch_array($select))
        echo '<option
value="'. $res['NumExp']. '>' . $res['NumExp']. '</option>';
?>
        </select></td></tr>
    <tr><td> Horario: </td>
      <td><input type="text" id="form_admin_inserta_asignatura_horario"
name="form_admin_inserta_asignatura_horario"></input></td></tr>
  </table></form>
```

Cuando se haga click en el elemento del menú correspondiente, y antes de crear el jqGrid, se cargará este archivo con el formulario oculto.

```
$("#formularios").load('scripts/admin/form_asignaturas.php');
```

Para seguir con la estética del jqGrid, le añadimos un botón para la función de inserción, de modo que al hacer click en el botón se nos abrirá un dialogo con el formulario que habíamos cargado previamente, que nos facilitará la inserción.

```
.navButtonAdd('#pager',{
  caption:"",
  title:"Insertar Asignatura",
  buttonicon:"ui-icon-plus",
  onClickButton: function(){
    $("#form_admin_inserta_asignatura").dialog(
      { buttons: [
        {text: "Insertar",
        click: function(){
          var nombre=$("#form_admin_inserta_asignatura_nombre").val();
          var grupo=$("#form_admin_inserta_asignatura_grupos").val();
          var profesor=$("#form_admin_inserta_asignatura_profesores").val();
          var horario=$("#form_admin_inserta_asignatura_horario").val();
          if(nombre=="*" || grupo=="*")
            alert("Los campos con asterisco son obligatorios.");
          else{
            $.post("scripts/admin/inserta_asignaturas.php",
              {
                nombre:nombre,
                grupo:grupo,
                profesor:profesor,
                horario:horario
              },
              function(datos){
                if (datos.status=="OK")
                {
                  alert("Asignatura insertada correctamente.");
                  $("#form_admin_inserta_asignatura").dialog("close");
                  $("#form_admin_inserta_asignatura").dialog("destroy");
                  $('#list').trigger("reloadGrid");
                  $("#formularios").load('scripts/admin/
form_asignaturas.php?id='+id_fila);
                }
                else
                  alert("Error-> "+datos.texto);
              }, "json");
            }
          }
        }
      ]},
      {text: "Cancelar",
      click: function(){
        $(this).dialog("close");
        $(this).dialog("destroy");
      }
    }
  ]
},
{ modal: true },
{ title: 'Insertar Asignatura' }
);
},
position:"first"
```

Una vez se hace click en el botón insertar, se comprueba que efectivamente los campos obligatorios están rellenos (color amarillo en el código). Si no es así, se advertirá mediante una alerta (código verde).

Una vez esté todo correcto, se procede al envío de los datos mediante post, a un script que ejecutará la inserción (color azul en el código) y que devuelve, una vez más mediante JSON, el resultado de la operación (color rosa en el código).

Este script de inserción de asignatura queda del siguiente modo:

```

<?php
session_start();
header("Content-type: text/json");
error_reporting(E_ALL); ini_set("display_errors", 0);
include "../conexion.php";
include "../funciones.php";
$error = "";

if( !isset($_POST['nombre']) || $_POST['nombre']=="")
    $error = "No se recibieron los parámetros correctamente";

$conexion = conectar_mysql();
$sql = "SELECT Nombre, Grupo FROM asignatura
        WHERE 1 AND Nombre LIKE '".$_POST['nombre']."'
        AND Grupo LIKE '".$_POST['grupo']."'";
$select=mysql_query($sql, $conexion) or $error="Problemas en el select:
.mysql_error($conexion);

if (mysql_num_rows($select)>'0') $error="La asignatura ya existe";

if($error=="")
{
    $sql = "INSERT INTO asignatura
            SET     Profesor      =     '".$_POST['profesor']."',
                  Grupo          =     '".$_POST['grupo']."',
                  Nombre         =     '".$_POST['nombre']."',
                  Horario        =     '".$_POST['horario']."'";

    $select=mysql_query($sql, $conexion) or $error="Problemas en el select:
.mysql_error($conexion);
    log_modificacion('', $sql, 'asignatura', '');
}
mysql_close($conexion);

// DEVUELVO UN JSON CON ESTADO
if($error <> ""){
    $response->status = "ERROR";
    $response->texto = utf8_encode($error);
}
else{
    $response->status = "OK";
}
echo json_encode($response);
?>

```

Primero se comprueba que se hayan recibido correctamente los datos (color amarillo en el código) y seguidamente que la asignatura no exista ya (color azul en el código). Si todo está correcto, se procede a insertar la asignatura (color rosa) y se envía la respuesta mediante JSON (color verde en el código).

### 5.3.2.3.3 Modificar datos

Para modificar los datos se utiliza el mismo sistema de la inserción. La diferencia reside en que el formulario oculto contendrá los datos a modificar y que para modificar, tendremos que asegurarnos de que hay una fila seleccionada y de que el usuario está seguro de querer modificarla.

Esto lo conseguimos con javascript:

```
var id_fila=$('#list').jqGrid('getGridParam','selrow');  
if ((id_fila==null)|| (id_fila==undefined)) alert("Seleccione una fila");
```

```
if (!confirm("Modificar asignatura \""+$('#list').getCell( id_fila, 'Nombre' )+"\"?"))  
{  
    $("#form_admin_modifica_asignatura").dialog("close");  
    $("#form_admin_modifica_asignatura").dialog("destroy");  
    return;  
}
```

El formulario en este caso lo llamaremos, pasándole por GET la id del objeto a modificar:

```
$("#formularios").load('scripts/admin/form_asignaturas.php?id='+id_fila);
```

De modo que nos debemos asegurar que, al seleccionar una fila, el formulario correspondiente a ese dato esté disponible:

```
onSelectRow: function(id){  
    //Borramos los dialogos abiertos y los divs creados para que no se  
    almacenen  
    var id_fila=$('#list').jqGrid('getGridParam','selrow');  
    $("#form_admin_inserta_asignatura").dialog("destroy");  
    $("#form_admin_inserta_asignatura").remove();  
    $("#form_admin_modifica_asignatura").dialog("destroy");  
    $("#form_admin_modifica_asignatura").remove();  
    $("#formularios").load('scripts/admin/form_asignaturas.php?id='+id_fila)  
};
```

#### 5.3.2.3.4 Borrar datos

Para borrar datos se sigue el mismo procedimiento que para modificarlos, con la excepción de que, en la mayoría de los casos, no necesitamos el formulario con los datos.

Nos aseguraremos igualmente de que el usuario esté seguro de eliminar:

```
var valor=prompt("Si está seguro de que desea borrar la asignatura, escriba  
BORRAR y haga click en Aceptar:");  
if (valor=='BORRAR')  
{  
    $.post("scripts/admin/elimina_asignatura.php",  
    {  
        id:id_fila,  
        tipo:"asignatura"  
    },  
    function(datos){  
        if (datos.status=="OK")  
        {  
            alert("Asignatura eliminada correctamente.");  
            $('#list').trigger("reloadGrid");  
        }  
        else  
            alert("Error-> "+datos.texto);  
    }, "json");  
}  
else if (valor==null)  
;  
else alert("Error, el texto introducido no coincide.");  
}
```

### 5.3.3 Capa de persistencia

Para la manipulación y almacenamiento de los datos de nuestra web, se utiliza una base de datos creada en MySQL con la siguiente estructura:

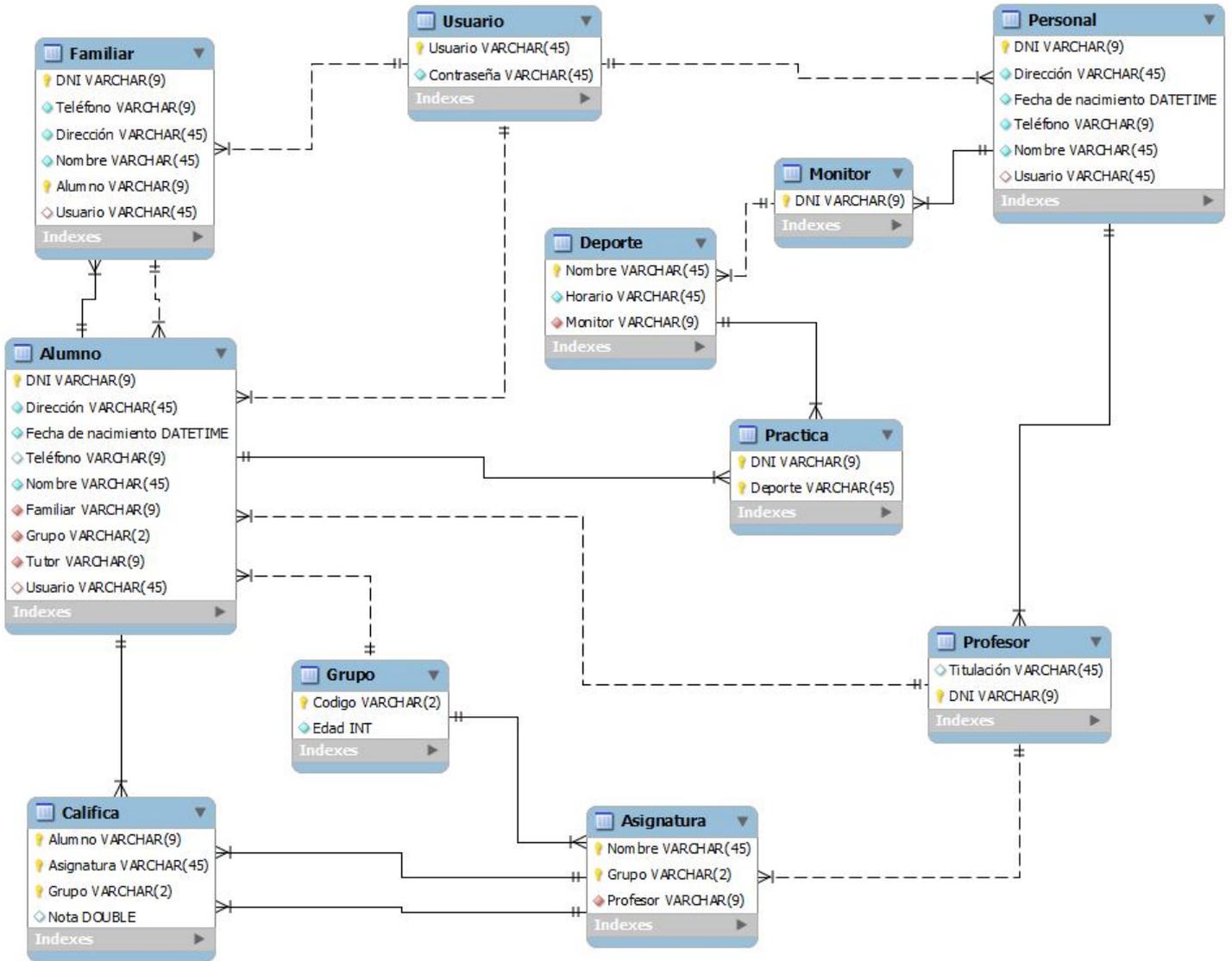


Ilustración 5-1 – Base de datos

En base a nuestro diagrama, creamos la base de datos, donde almacenaremos todos los datos de la web.

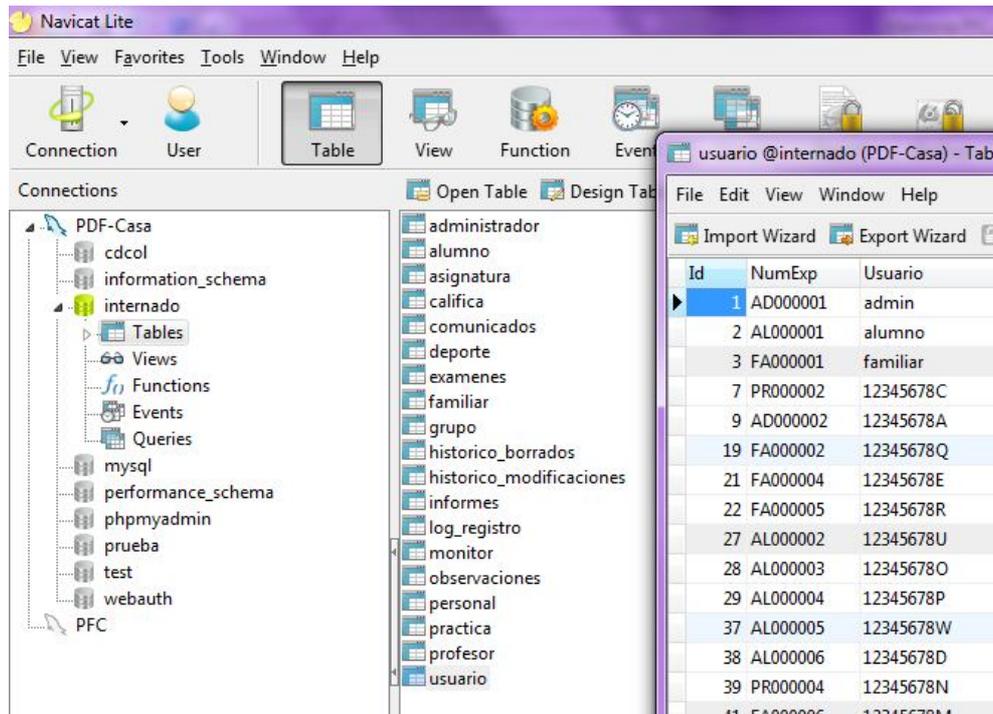


Ilustración 5-2 - Base de datos: Internado

Para conectarnos a ella, usaremos el archivo conexión.php:

```
<?php
if (!function_exists("conectar_mysql"))
{
    function conectar_mysql ()
    {
        $dbhost="localhost";
        $dbusuario="internado";
        $dbpassword="liconoc";
        $db="internado";
        $conexion = mysql_connect($dbhost, $dbusuario, $dbpassword, true)
or die ("No se puede conectar con la base de datos. Error:" .
mysql_error($conexion));
        mysql_select_db($db, $conexion) or die ("No se puede seleccionar
la base de datos");
        return $conexion;
    }
}
?>
```

## 6. Evaluación y Pruebas

Para evaluar el correcto funcionamiento del producto acabado, se han realizado una serie de pruebas donde comprobamos tanto la navegabilidad en diferentes navegadores y resoluciones, como la estructura interna de la aplicación.

### 6.1 Prueba de enlaces rotos

Realizamos esta prueba para comprobar que no haya ningún enlace roto en nuestra aplicación. Para ello utilizamos la herramienta que nos ofrece <http://validator.w3.org/checklink>.

Como ejemplo pondremos la validación de los enlaces en la página principal:

#### Processing <http://pion.disca.upv.es/internado/>

Go to [the results](#).

For reliable link checking results, check [HTML validity](#) and [CSS validity](#) first.

Back to the [link checker](#).

```
Status: Done. Document processed in 12.21 seconds.
-----
Checking link http://pion.disca.upv.es/internado/contacto.php
HEAD http://pion.disca.upv.es/internado/contacto.php fetched in 1.14 seconds

Checking link http://pion.disca.upv.es/internado/images/spa.png
HEAD http://pion.disca.upv.es/internado/images/spa.png fetched in 1.14 seconds

Checking link http://pion.disca.upv.es/internado/favicon.ico
HEAD http://pion.disca.upv.es/internado/favicon.ico fetched in 1.14 seconds

Checking link http://pion.disca.upv.es/internado/images/eng.png
HEAD http://pion.disca.upv.es/internado/images/eng.png fetched in 1.14 seconds

Processed in 12.21 seconds.
```

### Results

#### Links

Valid links!

#### Anchors

Found 19 anchors.

Valid anchors!

Checked 1 document in 12.28 seconds.

Ilustración 6-1 - Comprobación de enlaces rotos

## 6.2 Validación de html

Se ha comprobado que el html cumple los estándares, mediante el validador <http://validator.w3.org/>.

This document was successfully checked as HTML 4.01 Strict!	
<b>Result:</b>	Passed
<b>Address :</b>	<input type="text" value="http://pion.disca.upv.es/internado/"/>
<b>Encoding :</b>	iso-8859-1 <input type="text" value="(detect automatically)"/>
<b>Doctype :</b>	HTML 4.01 Strict <input type="text" value="(detect automatically)"/>
<b>Root Element:</b>	HTML

Ilustración 6-2 - Validación de html

## 6.3 Validación de css

Igual que el código html, las hojas de estilo Css, pueden ser validadas mediante <http://jigsaw.w3.org/css-validator/>.


El Servicio de Validación de CSS del W3C  
Resultados del Validador CSS del W3C para <http://pion.disca.upv.es/internado/estilos/estilo.css> (CSS versión 2.1)

---

Ir a:
Las Advertencias (4)
Su Hoja de Estilo validada

---

Resultados del Validador CSS del W3C para <http://pion.disca.upv.es/internado/estilos/estilo.css> (CSS versión 2.1)

¡Enhorabuena! No error encontrado.

¡Este documento es [CSS versión 2.1](#) válido!

Ilustración 6-3 - Validación de Css

### 6.4 Prueba de compatibilidad para móvil

Comprobamos también si nuestro portal sería visible a través de un terminal móvil mediante <http://validator.w3.org/mobile> .

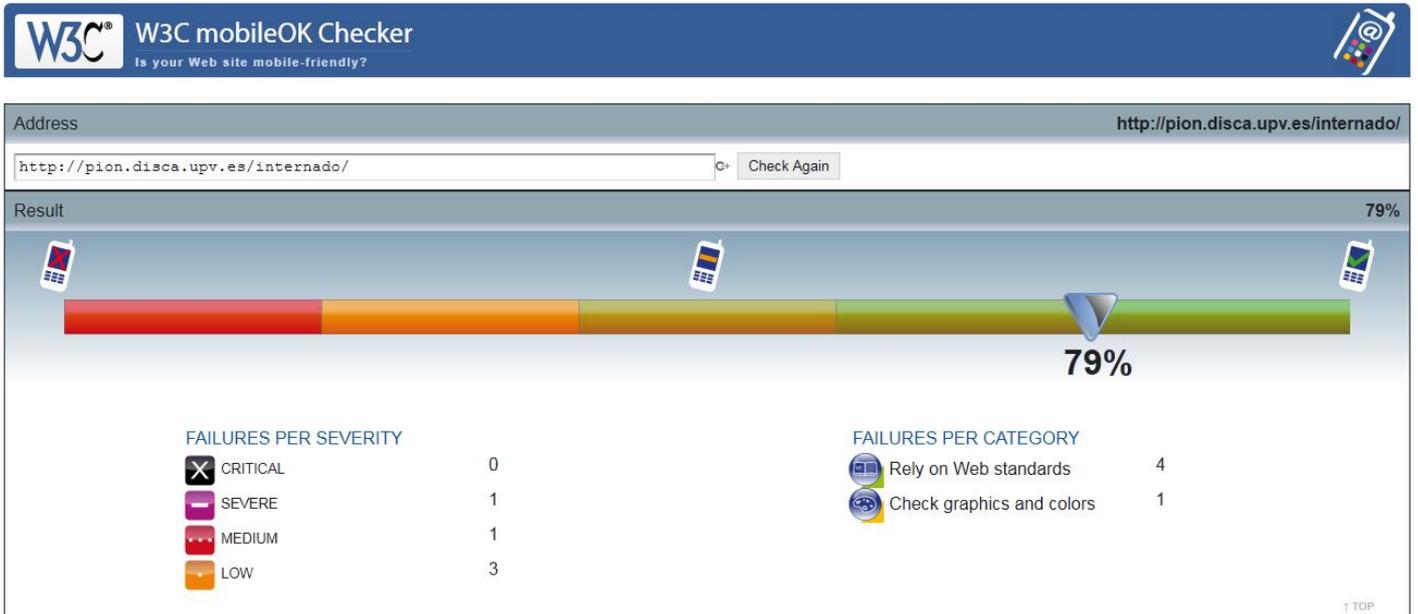


Ilustración 6-4 - Comprobación sobre móvil, página principal



Ilustración 6-5 - Comprobación sobre móvil, intranet

Como podemos ver en la ilustración 6-5, la intranet es compatible casi al 100% con los móviles, sin embargo la página principal tiene algunos errores que habría que corregir.

## 6.5 Compatibilidad con los navegadores

Comprobamos la compatibilidad de nuestra web en diferentes navegadores para asegurarnos de que cualquier usuario podrá entrar correctamente.

### 6.5.1 Mozilla Firefox



Ilustración 6-6 - Firefox: Página principal



Ilustración 6-7 - Firefox: Intranet

## 6.5.2 Internet Explorer

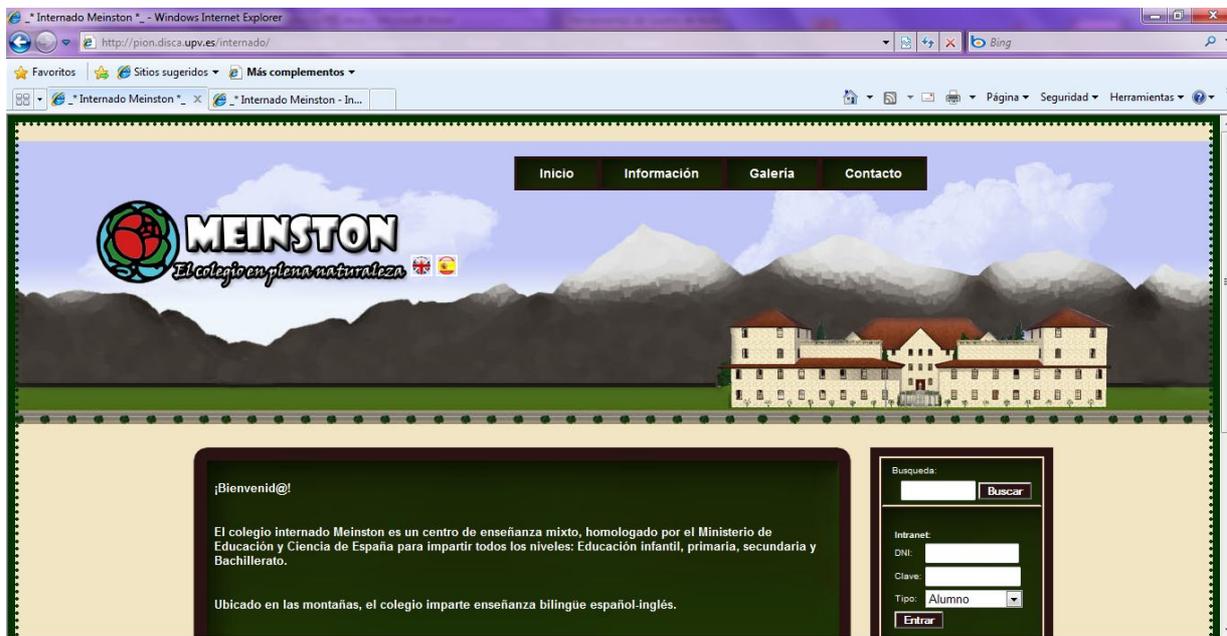


Ilustración 6-8 - Internet Explorer: Página principal

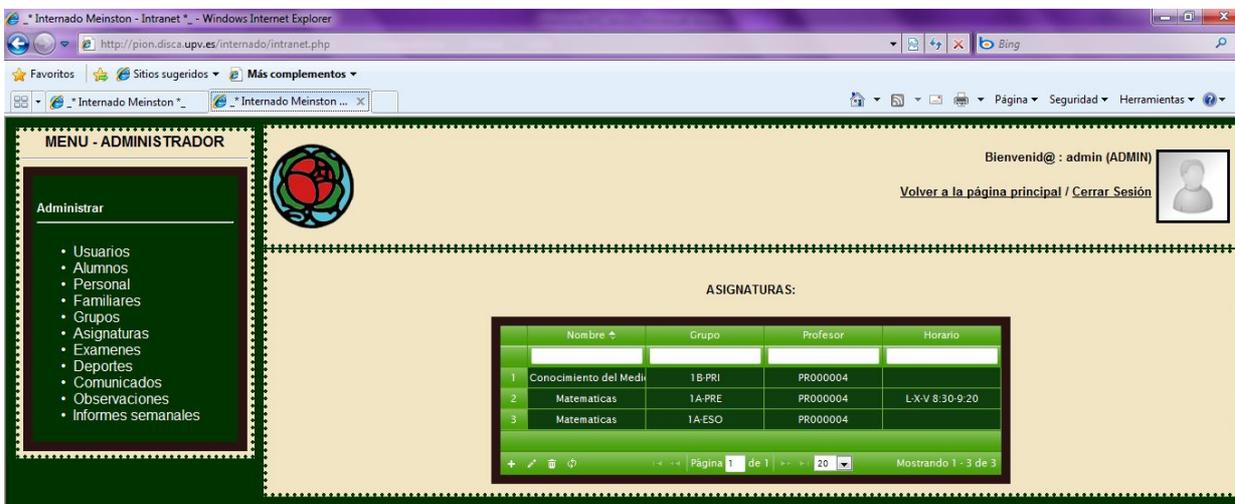


Ilustración 6-9 - Internet Explorer: Intranet

### 6.5.3 Google Chrome

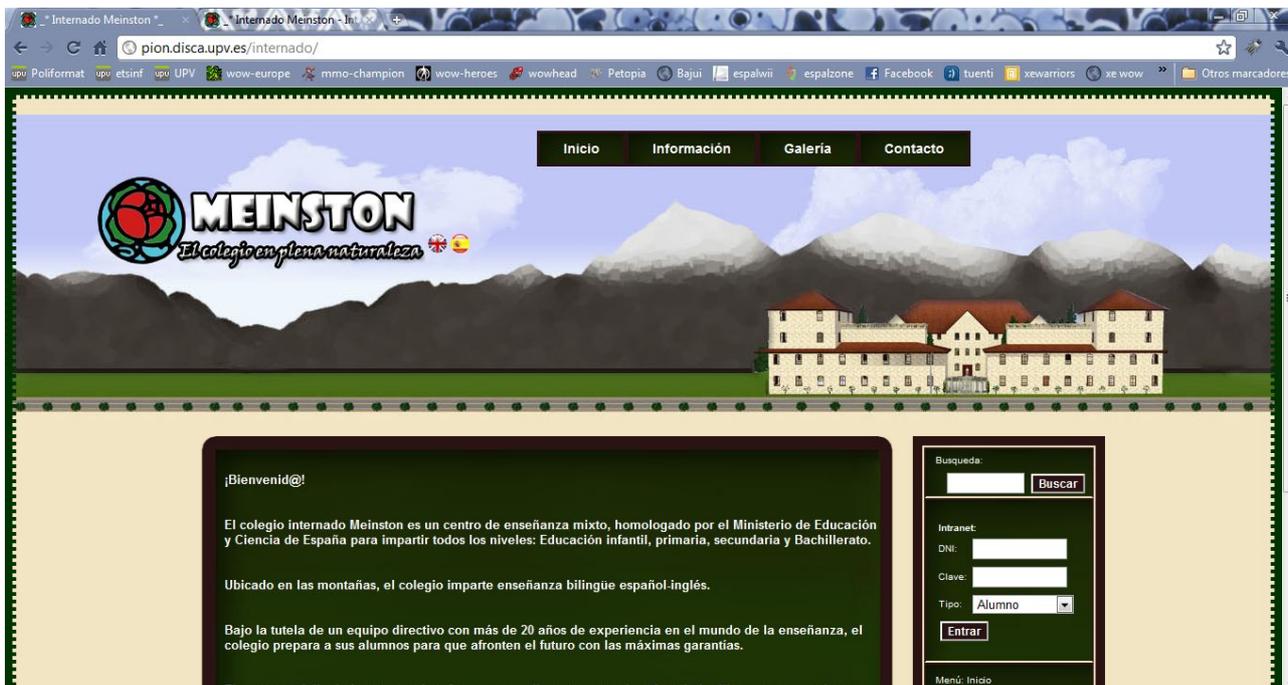


Ilustración 6–10 - Google Chrome: Página principal



Ilustración 6–11 - Google Chrome: Intranet

### 6.5.4 Opera

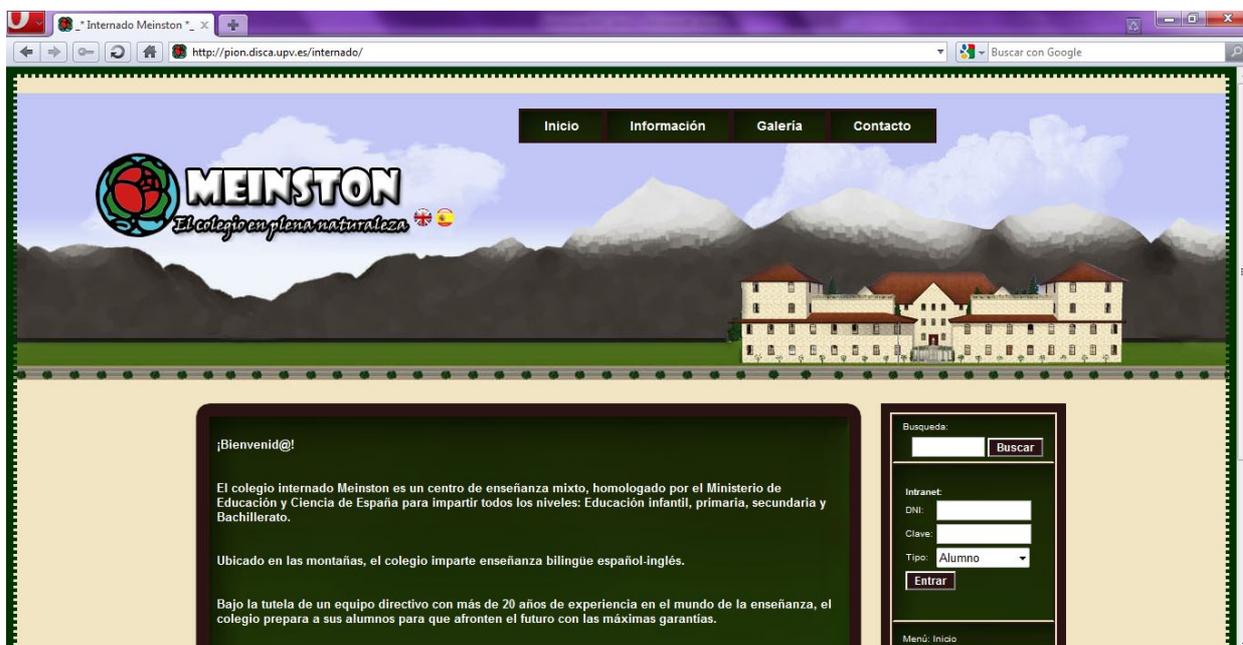


Ilustración 6-12 - Opera: Página principal



Ilustración 6-13 - Opera: Intranet

Como podemos ver, nuestro portal es compatible con todos los navegadores, tanto la página principal como la intranet.

## 6.6 Compatibilidad con diferentes resoluciones

Por último comprobamos si nuestra web es visiblemente compatible con diferentes resoluciones de pantalla, para asegurarnos de que se vea correctamente.



Ilustración 6-14 - Resolución:1366x768



Ilustración 6-15 - Resolución: 1280x1024



Ilustración 6-16 - Resolución: 1024x768

## 7. Conclusiones

Durante este proyecto, hemos creado la web del Internado Meinston, partiendo de cero. Al finalizarlo, el colegio ha entrado en internet, dándose a conocer y ofreciendo a sus alumnos, empleados y familiares un espacio virtual para poder comunicarse.

Viendo el resultado final, creo que el objetivo se ha conseguido, ya que se ha creado una interfaz sencilla, pero eficiente para el colegio.

Durante la realización de este proyecto, he aprendido muchas características de las tecnologías que he usado, en especial de PHP y JQuery que no había utilizado casi.

He intentado crear una interfaz sencilla, para un usuario final sin conocimientos informáticos, donde fuera fácil consultar, modificar o eliminar datos.

Otra de mis metas era que los usuarios se pudieran enviar mensajes entre ellos, que si bien no es tan sofisticado como lo puede ser el correo electrónico, para el uso para el que está creado creo que puede ser útil. Los mensajes se cargan dinámicamente, de modo que si un usuario le manda uno a otro usuario, este último solo tiene que actualizar la tabla para verlo.

Se podrían realizar mejoras visuales para la intranet, ya que es bastante sencilla. Sin embargo ese no era el objetivo de este proyecto y con su aspecto actual es fácil saber cómo llegar a donde se quiere ir.

JqGrid tiene muchas más opciones que no hemos utilizado en nuestra web, y que podrían ser de utilidad para las acciones que se realizan en la intranet.

En el futuro se podría mejorar el aspecto de la intranet, así como introducir nuevas acciones como ejercicios online, chat para el grupo de alumnos, tablón de anuncios, préstamo de libros, cita médica, etc.

De igual modo, se podría actualizar para que fuera accesible a personas invidentes, o adaptada a otros medios como los teléfonos móviles.

Lo que más me ha costado ha sido crear las sesiones y utilizarlas para todos los scripts, ya que nunca lo había hecho con PHP y averiguar los fallos de Javascript. También me costó al principio crear la página principal y poner cada apartado en su sitio mediante Css. Sin embargo me ha gustado realizar este proyecto y su resultado final.

## 8. Bibliografía

- **Proyectos de Fin de Carrera de años anteriores.**
- **Apuntes de asignaturas:**
  - o BDA : Bases de datos
  - o ISG: Ingeniería del software
  - o CDH: Creación de documentos de hipertexto
  - o DBD: Diseño de bases de datos
  - o PBS: Programación básica en el servidor
  - o SWE: Servidores de web
  - o ADB: Administración se sistemas de bases de datos
- **Apuntes de cursos:**
  - o CURSO DE PROGRAMACIÓN AJAX CON HTML, JAVASCRIPT Y PHP
- Conocimientos adquiridos en la realización de **prácticas de empresa** en el Colegio de Psicólogos de la Comunidad Valenciana.
- **Información de páginas web:**
  - o <http://es.wikipedia.org>
  - o <http://www.forosdelweb.com>
  - o <http://docs.jquery.com>
  - o <http://www.php.net/manual/es>
  - o <http://www.javascriptya.com.ar/jquery>
  - o <http://www.phpya.com.ar>
  - o <http://www.cssya.com.ar>
  - o <http://www.trirand.com/jqgridwiki/doku.php?id=wiki:jqgriddocs>

## 9. ANEXOS

### 9.1 ANEXO A: Casos de Uso

<b>Caso de uso</b>	Solicitar información	
<b>Actores</b>	Todos los usuarios	
<b>Propósito</b>	Obtener información acerca del colegio	
<b>Resumen</b>	Cualquier usuario puede solicitar información sobre cualquier aspecto del internado.	
<b>Precondiciones</b>	Ser usuario	
<b>Pos condiciones</b>	La administración recibirá la solicitud	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón "contacto" y accede al formulario</li> <li>2. Rellena el formulario</li> <li>3. Le da al botón enviar</li> </ol>	<ol style="list-style-type: none"> <li>4. El sistema enviará la información del formulario por e-mail a la administración.</li> </ol>
<b>Extensiones síncronas</b>	<p>Hasta el paso 3 se puede cancelar</p> <p>Si no se rellena algún dato obligatorio el sistema lo comprobará y mostrará el error</p>	

<b>Caso de uso</b>	Acceder a la intranet	
<b>Actores</b>	Cualquier usuario con clave	
<b>Propósito</b>	Acceder a la zona restringida de la web	
<b>Resumen</b>	Cualquier usuario que sea registrado puede intentar acceder a su zona privada mediante su usuario y su contraseña	
<b>Precondiciones</b>	Ser un usuario registrado	
<b>Pos condiciones</b>	El usuario accederá a la intranet	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón "Intranet"</li> <li>2. Selecciona entre "Alumno / Familiar / Personal / Admin"</li> <li>3. Rellena los campos "usuario" y "contraseña".</li> <li>4. Pulsa el botón "Entrar"</li> </ol>	<ol style="list-style-type: none"> <li>5. El sistema comprueba que el usuario y la contraseña existen y que pertenecen al tipo de cuenta seleccionado y redirige al navegador a la intranet.</li> </ol>
<b>Extensiones sincronas</b>	<p>Hasta el paso 4 se puede cancelar</p> <p>Si en el paso 5 se comprueba que algún dato no es correcto el sistema informará del error y vaciará los campos.</p>	

<b>Caso de uso</b>	Consultar datos personales propios	
<b>Actores</b>	Usuario registrado	
<b>Propósito</b>	Consultar los datos que hay registrados	
<b>Resumen</b>	El usuario registrado puede ver los datos personales que la base de datos tiene suyos	
<b>Precondiciones</b>	Ser usuario registrado y estar en la intranet	
<b>Pos condiciones</b>	El usuario ve sus datos personales	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	1. El usuario pulsa el botón "Datos personales"	2. El sistema redirige el navegador a la página adecuada.
<b>Extensiones síncronas</b>	Si la sesión ha terminado el sistema no permitirá que los datos se vean	

<b>Caso de uso</b>	Listar alumnos asignados	
<b>Actores</b>	Familiar / tutor	
<b>Propósito</b>	Ver la lista de los alumnos a su cargo	
<b>Resumen</b>	El familiar / tutor verá los alumnos que tiene asignados	
<b>Precondiciones</b>	Ser un familiar / tutor registrado y estar en la intranet	
<b>Pos condiciones</b>	El familiar / tutor ve los alumnos	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	1. El familiar / tutor pulsa el botón "ver alumnos asignados"	2. El sistema redirige el navegador a la página adecuada.
<b>Extensiones síncronas</b>	Si la sesión ha terminado el sistema no permitirá que los datos se vean	

<b>Caso de uso</b>	Seleccionar alumno	
<b>Actores</b>	Familiar o personal	
<b>Propósito</b>	Seleccionar un alumno de una lista de alumnos	
<b>Resumen</b>	El usuario accede a los datos requeridos del alumno seleccionado	
<b>Precondiciones</b>	Ser familiar del alumno o personal del colegio y estar en la intranet habiendo seleccionado previamente ver la lista de alumnos	
<b>Pos condiciones</b>	El usuario ve los datos del alumno	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	1. El usuario pulsa el botón con el nombre del alumno requerido de la lista de alumnos.	2. El sistema redirige el navegador a la página adecuada.
<b>Extensiones síncronas</b>	Si la sesión ha terminado el sistema no permitirá que los datos se vean	

<b>Caso de uso</b>	Ver informe semanal	
<b>Actores</b>	Familiar	
<b>Propósito</b>	Ver el informe semanal de un alumno a su cargo	
<b>Resumen</b>	El familiar verá el informe semanal del alumno	
<b>Precondiciones</b>	Ser un familiar registrado y estar en la intranet	
<b>Pos condiciones</b>	Poder leer el informe	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	1. El familiar pulsa el botón "Ver informe semanal"	2. El sistema redirige el navegador a la página con la lista de alumnos a cargo del familiar.
	3. El familiar pulsa el botón cuyo nombre pertenece al alumno de quien desea ver el informe.	4. El sistema redirige el navegador a la página adecuada.
<b>Extensiones síncronas</b>	Si la sesión ha terminado el sistema no permitirá que los datos se vean	

<b>Caso de uso</b>	Listar comunicados / asignaturas / deportes / notas / observaciones / exámenes del alumno	
<b>Actores</b>	Familiar	
<b>Propósito</b>	Ver los datos seleccionados del alumno a su cargo	
<b>Resumen</b>	El familiar verá los datos del alumno	
<b>Precondiciones</b>	Ser un familiar registrado y estar en la intranet	
<b>Pos condiciones</b>	Poder leer los datos	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	<p>1. El familiar pulsa el botón "Ver comunicados / asignaturas / deportes / notas / observaciones / exámenes de alumno"</p> <p>3. El familiar pulsa el botón cuyo nombre pertenece al alumno</p>	<p>2. El sistema redirige el navegador a la página con la lista de alumnos a cargo del familiar.</p> <p>4. El sistema redirige el navegador a la página adecuada.</p>
<b>Extensiones síncronas</b>	Si la sesión ha terminado el sistema no permitirá que los datos se vean	

<b>Caso de uso</b>	Escribir comunicado a un alumno	
<b>Actores</b>	Familiar o Tutor del alumno	
<b>Propósito</b>	Enviar un comunicado al alumnos seleccionado	
<b>Resumen</b>	El familiar o tutor de un alumno puede enviarle un comunicado	
<b>Precondiciones</b>	Ser un familiar registrado o tutor y estar en la intranet	
<b>Pos condiciones</b>	El alumno debe recibir el comunicado	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	<ol style="list-style-type: none"> <li>1. El familiar o tutor pulsa el botón "Enviar comunicado"</li> <li>3. Pulsa el botón cuyo nombre pertenece al alumno a quien desea enviar el comunicado</li> <li>5. Rellena el formulario de envío</li> <li>6. Pulsa el botón "Enviar"</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema redirige el navegador a la página con la lista de alumnos a cargo del familiar o tutor.</li> <li>4. El sistema redirige el navegador a la página adecuada.</li> <li>7. El sistema enviará el comunicado a la bandeja de entrada de comunicados del alumno.</li> </ol>
<b>Extensiones síncronas</b>	Si la sesión ha terminado el sistema no permitirá que los datos se envíen	

<b>Caso de uso</b>	Escribir comunicado al tutor de un alumno	
<b>Actores</b>	Familiar	
<b>Propósito</b>	Enviar un comunicado tutor del alumno seleccionado	
<b>Resumen</b>	El familiar envía un comunicado al tutor	
<b>Precondiciones</b>	Ser un familiar registrado o tutor y estar en la intranet	
<b>Pos condiciones</b>	El tutor debe recibir el comunicado	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	<ol style="list-style-type: none"> <li>1. El familiar o tutor pulsa el botón "Enviar comunicado"</li> <li>3. Pulsa el botón cuyo nombre pertenece al alumno</li> <li>5. Rellena el formulario de envío</li> <li>6. Pulsa el botón "Enviar"</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema redirige el navegador a la página con la lista de alumnos a cargo del familiar o tutor</li> <li>4. El sistema redirige el navegador a la página adecuada</li> <li>7. El sistema enviará el comunicado a la bandeja de entrada de comunicados del tutor del alumno seleccionado</li> </ol>
<b>Extensiones síncronas</b>	<p>Si la sesión ha terminado el sistema no permitirá que los datos se envíen</p> <p>Hasta el paso 5 se puede cancelar</p>	

<b>Caso de uso</b>	Modificar datos personales	
<b>Actores</b>	Familiar / Personal	
<b>Propósito</b>	Modificar los datos personales	
<b>Resumen</b>	Cualquier usuario registrado, salvo los alumnos, puede modificar sus datos	
<b>Precondiciones</b>	Ser un usuario registrado –no alumno- y estar en la intranet	
<b>Pos condiciones</b>	Actualizar sus datos	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	<p>1. El usuario pulsa el botón "Modificar datos"</p> <p>3. Rellena los campos que quiere cambiar y pulsa el botón "Cambiar"</p>	<p>2. El sistema redirige el navegador a la página con el formulario</p> <p>4. El sistema comprueba que los datos sean correctos, los actualiza en la base de datos y redirige el navegador a la página de resultados</p>
<b>Extensiones sincronas</b>	<p>Si la sesión ha terminado el sistema no permitirá que los datos se envíen</p> <p>Hasta el paso 3 se puede cancelar</p>	

<b>Caso de uso</b>	Modificar datos personales de un alumno	
<b>Actores</b>	Familiar	
<b>Propósito</b>	Modificar los datos personales de un alumno a su cargo	
<b>Resumen</b>	Cualquier familiar puede modificar los datos de los alumnos a su cargo (Solo dirección y teléfono)	
<b>Precondiciones</b>	Ser un familiar y estar en la intranet	
<b>Pos condiciones</b>	Actualizar los datos del alumno	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	<ol style="list-style-type: none"> <li>1. El usuario pulsa el botón "Modificar datos de alumno"</li> <li>3. Pulsa el botón cuyo nombre pertenece al alumno</li> <li>5. Rellena los campos que quiere cambiar y pulsa el botón "Cambiar"</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema redirige el navegador a la página con la lista de alumnos a cargo del familiar o tutor</li> <li>4. El sistema redirige el navegador a la página adecuada</li> <li>6. El sistema comprueba que los datos sean correctos, los actualiza en la base de datos y redirige el navegador a la página de resultados</li> </ol>
<b>Extensiones síncronas</b>	<p>Si la sesión ha terminado el sistema no permitirá que los datos se envíen</p> <p>Hasta el paso 5 se puede cancelar</p>	

<b>Caso de uso</b>	Listar comunicados / asignaturas / deportes / notas / observaciones / exámenes / tutor	
<b>Actores</b>	Alumno / profesor	
<b>Propósito</b>	Ver los datos seleccionados	
<b>Resumen</b>	El alumno / profesor verá sus datos	
<b>Precondiciones</b>	Ser un alumno / profesor registrado y estar en la intranet	
<b>Pos condiciones</b>	Poder leer los datos	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	1. El alumno / profesor pulsa el botón "Ver comunicados / asignaturas / deportes / notas / observaciones / exámenes / tutor"	2. El sistema redirige el navegador a la página adecuada.
<b>Extensiones sincronas</b>	Si la sesión ha terminado el sistema no permitirá que los datos se vean	

<b>Caso de uso</b>	Escribir comunicado a tutor / familiar	
<b>Actores</b>	Alumno	
<b>Propósito</b>	Enviar un comunicado al tutor / familiar	
<b>Resumen</b>	El alumno envía un comunicado	
<b>Precondiciones</b>	Ser un alumno registrado o tutor y estar en la intranet	
<b>Pos condiciones</b>	El tutor / familiar debe recibir el comunicado	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	1. El alumno pulsa el botón "Enviar comunicado a tutor / familiar"	2.. El sistema redirige el navegador a la página adecuada
	3. Rellena el formulario de envío	
	4. Pulsa el botón "Enviar"	5. El sistema enviará el comunicado a la bandeja de entrada de comunicados del tutor / Familiar del alumno
<b>Extensiones sincronas</b>	Si la sesión ha terminado el sistema no permitirá que los datos se envíen Hasta el paso 4 se puede cancelar	

<b>Caso de uso</b>	Borrar comunicado	
<b>Actores</b>	Alumno / familiar / tutor	
<b>Propósito</b>	Borrar un comunicado existente	
<b>Resumen</b>	Seleccionar un comunicado y borrarlo	
<b>Precondiciones</b>	Ser un alumno / familiar / tutor registrado y estar en la intranet	
<b>Pos condiciones</b>	El comunicado debe ser eliminado	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	1. El usuario pulsa el botón "Borrar comunicado" en la lista de comunicados	2. El sistema elimina el comunicado de la base de datos y redirige el navegador a la página adecuada
<b>Extensiones sincronas</b>	Si la sesión ha terminado el sistema no permitirá que los datos se eliminen Hasta el paso 1 se puede cancelar	

<b>Caso de uso</b>	Inscribirse en deporte	
<b>Actores</b>	Alumno	
<b>Propósito</b>	Darse de alta en un deporte	
<b>Resumen</b>	El alumno se apunta a un deporte disponible	
<b>Precondiciones</b>	Ser un alumno registrado y estar en la intranet Debe haber hueco libre en el deporte escogido	
<b>Pos condiciones</b>	El alumno debe quedar matriculado en el deporte	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	1. El alumno pulsa el botón "Ver deportes disponibles"  3. Selecciona el deporte 4. Pulsa el botón "Aceptar"	2. El sistema redirige el navegador a la página adecuada (La pagina solo dejará seleccionar aquellos deportes con huecos libres)  5. El sistema comprueba que el alumno no está apuntado al máximo de deportes permitidos y que en el deporte seleccionado haya hueco y redirige a una página con los deportes asignados
<b>Extensiones sincronas</b>	Si la sesión ha terminado el sistema no permitirá que los datos se envíen Hasta el paso 4 se puede cancelar Si en el paso 5 no se cumplen las condiciones, el sistema avisará del error y el alumno no será matriculado	

<b>Caso de uso</b>	Borrarse de deporte	
<b>Actores</b>	Alumno	
<b>Propósito</b>	Darse de baja en un deporte	
<b>Resumen</b>	El alumno se borra de un deporte matriculado	
<b>Precondiciones</b>	Ser un alumno registrado y estar en la intranet	
<b>Pos condiciones</b>	El alumno debe quedar borrado en el deporte	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	1. El alumno pulsa el botón "Ver deportes matriculados"  3. Selecciona el deporte 4. Pulsa el botón "Borrar"	2. El sistema redirige el navegador a la página adecuada  5. El sistema elimina al alumno del deporte y redirige a la página de deportes matriculados
<b>Extensiones sincronas</b>	Si la sesión ha terminado el sistema no permitirá que los datos se envíen Hasta el paso 4 se puede cancelar	

<b>Caso de uso</b>	Modificar nota	
<b>Actores</b>	Profesor	
<b>Propósito</b>	Modificar la nota de un alumno en una asignatura concreta	
<b>Resumen</b>	El profesor modifica la nota de un alumno	
<b>Precondiciones</b>	Ser un profesor registrado y estar en la intranet	
<b>Pos condiciones</b>	La nota del alumno debe cambiar	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	<p>1. El profesor pulsa el botón "Poner nota"</p> <p>3. Selecciona la asignatura</p> <p>4. Pulsa el botón "Aceptar"</p> <p>6. El profesor modifica las notas correspondientes y pulsa el botón "Aceptar"</p>	<p>2. El sistema redirige el navegador a la página con la lista de asignaturas asignadas</p> <p>5. El sistema redirige el navegador a la página con la lista de alumnos y el formulario para modificar la nota</p> <p>7. El sistema comprueba que los datos sean correctos y realiza las modificaciones. Redirige el navegador a la página con los resultados.</p>
<b>Extensiones sincronas</b>	<p>Si la sesión ha terminado el sistema no permitirá que los datos se envíen</p> <p>Hasta el paso 6 se puede cancelar</p> <p>Si en el paso 7 no se cumplen las condiciones, el sistema avisará del error y vaciará los huecos con error.</p>	

<b>Caso de uso</b>	Escribir Informe semanal	
<b>Actores</b>	Tutor	
<b>Propósito</b>	Enviar el informe semanal al familiar de un alumno	
<b>Resumen</b>	El tutor envía el informe	
<b>Precondiciones</b>	Ser un tutor registrado y estar en la intranet	
<b>Pos condiciones</b>	El familiar debe recibir el informe	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	<ol style="list-style-type: none"> <li>1. El tutor pulsa el botón "Escribir informe semanal"</li> <li>3. Selecciona el alumno y pulsa "Aceptar"</li> <li>4. Rellena el formulario de envío</li> <li>5. Pulsa el botón "Enviar"</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema redirige a la página de selección de alumno</li> <li>3.. El sistema redirige el navegador a la página adecuada</li> <li>6. El sistema enviará el informe a la bandeja de entrada de comunicados Familiar del alumno</li> </ol>
<b>Extensiones sincronas</b>	<p>Si la sesión ha terminado el sistema no permitirá que los datos se envíen</p> <p>Hasta el paso 5 se puede cancelar</p>	

<b>Caso de uso</b>	Escribir observación	
<b>Actores</b>	Personal / Alumno	
<b>Propósito</b>	Enviar una observación	
<b>Resumen</b>	Se envía una observación a la persona seleccionada	
<b>Precondiciones</b>	Ser un alumno / personal registrado y estar en la intranet	
<b>Pos condiciones</b>	La persona debe recibir el informe	
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	1. El tutor pulsa el botón "Escribir observación"  3. Selecciona el receptor y pulsa "Aceptar"  4. Rellena el formulario de envío 5. Pulsa el botón "Enviar"	2. El sistema redirige a la página de selección de receptor  3.. El sistema redirige el navegador a la página adecuada  6. El sistema enviará el informe a la bandeja de entrada de observaciones
<b>Extensiones sincronas</b>	Si la sesión ha terminado el sistema no permitirá que los datos se envíen  Hasta el paso 5 se puede cancelar	

<b>Caso de uso</b>	Buscar usuario	
<b>Actores</b>	Administrador	
<b>Propósito</b>	Buscar un usuario por nombre / DNI	
<b>Resumen</b>	El administrador puede hacer una búsqueda por diferentes criterios	
<b>Precondiciones</b>	Ser administrador y estar en la intranet	
<b>Pos condiciones</b>		
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	<ol style="list-style-type: none"> <li>1. El administrador pulsa el botón "Nueva búsqueda"</li> <li>3. Rellena el formulario y pulsa "Aceptar"</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema redirige a la página con el formulario</li> <li>3. El sistema realiza la búsqueda y redirige a una página con los resultados obtenidos</li> </ol>
<b>Extensiones sincronas</b>	<p>Si la sesión ha terminado el sistema no permitirá que los datos se envíen</p> <p>Hasta el paso 3 se puede cancelar</p>	

<b>Caso de uso</b>	Listar usuarios	
<b>Actores</b>	Administrador	
<b>Propósito</b>	Listar todos los usuarios	
<b>Resumen</b>	El administrador puede ver todos los usuarios de la aplicación	
<b>Precondiciones</b>	Ser administrador y estar en la intranet	
<b>Pos condiciones</b>		
<b>Flujo de eventos</b>	<b>Interacción con el usuario</b>	<b>Obligaciones del sistema</b>
	<ol style="list-style-type: none"> <li>1. El administrador pulsa el botón "Listar usuarios"</li> </ol>	<ol style="list-style-type: none"> <li>2. El sistema redirige a la página</li> </ol>
<b>Extensiones sincronas</b>	<p>Si la sesión ha terminado el sistema no permitirá que los datos se envíen</p>	

## 9.2 ANEXO B: Diseño conceptual de la base de datos

**USUARIO** (Id:dom\_Id, NumExp:dom\_NumExp, Usuario:dom\_DNI,  
Contraseña:dom\_contrasenya, Fecha\_Alta:dom\_Fecha\_Alta, Tipo:dom\_Tipo)  
CP{Id, NumExp, Usuario}

**PERSONAL** (Id:dom\_Id, NumExp:dom\_NumExp, DNI:dom\_DNI,  
Telefono:dom\_Telefono, Direccion:dom\_Direccion, Nombre:dom\_Nombre,  
Apellidos:dom\_Apellidos, Fecha\_nacimiento:dom\_Fecha\_nacimiento)  
CP{ID, NumExp, DNI }  
CAj: {NumExp}→USUARIO

**GRUPO** (Id:dom\_Id, Codigo:dom\_Codigo, Edad:dom\_Edad)  
CP{Id, Codigo, Edad}

**PROFESOR** (Id:dom\_Id, NumExp:dom\_NumExp , Titulacion:dom\_Titulacion)  
CP{Id, NumExp }  
CAj: {NumExp}→PERSONAL

**DEPORTE** {Id:dom\_Id , Nombre:dom\_Nombre, Horario:dom\_Horario,  
Monitor:dom\_NumExp)  
CP{Id, Nombre}  
CAj: {Monitor}→MONITOR

**MONITOR** (Id:dom\_Id, NumExp:dom\_NumExp)  
CP{Id, NumExp}  
CAj: {NumExp}→PERSONAL

**FAMILIAR** (Id:dom\_Id, NumExp:dom\_NumExp, DNI:dom\_DNI,  
Telefono:dom\_Telefono, Direccion:dom\_Direccion, Nombre:dom\_Nombre,  
Apellidos:dom\_Apellidos)  
CP{Id, DNI, NumExp}  
CAj: {NumExp}→USUARIO

**ALUMNO** (Id:dom\_Id, NumExp:dom\_NumExp, DNI:dom\_DNI, Telefono:dom\_Telefono, Direccion:dom\_Direccion, Nombre:dom\_Nombre, Apellidos:dom\_Apellidos, Fecha\_nacimiento:dom\_Fecha\_nacimiento, Familiar:dom\_NumExp, Grupo:dom\_Codigo, Tutor: dom\_NumExp)

CP{Id, NumExp, DNI }

CAj: {Familiar}→FAMILIAR

CAj: {Grupo}→GRUPO

CAj: {Tutor}→PERSONAL

**ASIGNATURA** (Id:dom\_Id, Nombre:dom\_Nombre, Grupo:dom\_Codigo, Profesor:dom\_NumExp, Horario:dom\_Horario)

CP{Id, Nombre}

CAj: {Grupo}→GRUPO

CAj: {Profesor}→PROFESOR

**CALIFICA** (Id:dom\_Id, Alumno:dom\_NumExp, Asignatura:dom\_Id, Grupo:dom\_Codigo, Nota:dom\_Nota)

CP{Id }

CAj: {Alumno}→ALUMNO

CAj: {Asignatura}→ASIGNATURA

CAj: {Grupo}→GRUPO

**PRACTICA** (Id:dom\_Id, Alumno:dom\_NumExp, Deporte:dom\_Id)

CP{Id}

CAj: {Alumno}→ALUMNO

CAj: {Deporte}→DEPORTE

**ADMINISTRADOR** (Id:dom\_Id, NumExp:dom\_NumExp, Usuario:dom\_DNI)

CP{Id, NumExp, Usuario}

CAj: {NumExp}→USUARIO

**COMUNICADOS** (Id:dom\_Id, Emisor:dom\_NumExp, Receptor:dom\_NumExp, Mensaje:dom\_Mensaje, Fecha:dom\_Fecha, Leido:dom\_Leido)

CP{Id}

CAj: {Emisor}→USUARIO

CAj: {Receptor}→USUARIO

**OBSERVACIONES** (Id:dom\_Id, Emisor:dom\_NumExp, Receptor:dom\_NumExp, Mensaje:dom\_Mensaje, Fecha:dom\_Fecha, Leido:dom\_Leido)

CP{Id}

CAj: {Emisor}→USUARIO

CAj: {Receptor}→USUARIO

**INFORMES** (Id:dom\_Id, Tutor:dom\_NumExp, Familiar:dom\_NumExp, Alumno:dom\_NumExp, Informe:dom\_Informe, Fecha:dom\_Fecha, Leido:dom\_Leido)

CP{Id}

CAj: {Tutor}→PERSONAL

CAj: {Familiar}→FAMILIAR

CAj: {Alumno}→ALUMNO

**EXAMENES** (Id:dom\_Id, Asignatura:dom\_Id, Grupo:dom\_Codigo, Fecha:dom\_Fecha, Hora:dom\_Hora, Duracion:dom\_Duracion)

CP{Id, Asignatura, Grupo}

CAj: {Asignatura}→ASIGNATURA

CAj: {Grupo}→GRUPO

### 9.3 ANEXO C: Dominios base de datos

Nombre dominio	Tipo de dato
<b>dom_DNI</b>	Cadena(9)
<b>dom_Contrasenya</b>	Cadena(45)
<b>dom_Telefono</b>	Cadena(9)
<b>dom_Direccion</b>	Cadena(45)
<b>dom_Nombre</b>	Cadena(45)
<b>dom_Fecha_nacimiento</b>	Fecha
<b>dom_Codigo</b>	Cadena(6)
<b>dom_Edad</b>	Entero(10)
<b>dom_Titulacion</b>	Cadena(45)
<b>dom_Horario</b>	Cadena(45)
<b>dom_Id</b>	Entero(11)
<b>dom_NumExp</b>	Cadena(30)
<b>dom_Fecha_Alta</b>	Fecha+Hora
<b>dom_Tipo</b>	Cadena(30)
<b>dom_Apellidos</b>	Cadena(80)
<b>dom_Nota</b>	Double
<b>dom_Mensaje</b>	Cadena(200)
<b>dom_Fecha</b>	Fecha
<b>dom_Leido</b>	Tinynt(1)
<b>dom_Informe</b>	Cadena(200)
<b>dom_Hora</b>	Hora
<b>dom_Duracion</b>	Cadena(5)

## 9.4 ANEXO D: Fichero de creación de la base de datos

```

CREATE SCHEMA IF NOT EXISTS `internado`;

-- -----
-- Table structure for `administrador`
-- -----
DROP TABLE IF EXISTS `administrador`;
CREATE TABLE `administrador` (
  `Id` int(11) NOT NULL AUTO_INCREMENT,
  `Usuario` varchar(9) NOT NULL,
  `NumExp` varchar(30) NOT NULL DEFAULT '',
  PRIMARY KEY (`Id`,`Usuario`,`NumExp`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;

-- -----
-- Table structure for `alumno`
-- -----
DROP TABLE IF EXISTS `alumno`;
CREATE TABLE `alumno` (
  `Id` int(11) NOT NULL AUTO_INCREMENT,
  `NumExp` varchar(30) NOT NULL DEFAULT '',
  `DNI` varchar(9) NOT NULL,
  `Direccion` varchar(45) CHARACTER SET utf8 NOT NULL,
  `Fecha_nacimiento` datetime NOT NULL,
  `Telefono` varchar(9) CHARACTER SET utf8 DEFAULT NULL,
  `Nombre` varchar(45) NOT NULL,
  `Apellidos` varchar(80) DEFAULT NULL,
  `Familiar` varchar(9) DEFAULT NULL,
  `Grupo` varchar(6) DEFAULT NULL,
  `Tutor` varchar(9) DEFAULT NULL,
  PRIMARY KEY (`Id`,`NumExp`,`DNI`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;

-- -----
-- Table structure for `asignatura`
-- -----
DROP TABLE IF EXISTS `asignatura`;
CREATE TABLE `asignatura` (
  `Id` int(11) NOT NULL AUTO_INCREMENT,
  `Nombre` varchar(45) NOT NULL,
  `Grupo` varchar(6) NOT NULL,
  `Profesor` varchar(9) DEFAULT NULL,
  `Horario` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`Id`,`Nombre`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;

```

```
-----  
-- Table structure for `califica`  
-----  
DROP TABLE IF EXISTS `califica`;  
CREATE TABLE `califica` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `Alumno` varchar(30) NOT NULL,  
  `Asignatura` varchar(45) NOT NULL,  
  `Grupo` varchar(6) NOT NULL,  
  `Nota` double unsigned DEFAULT NULL,  
  PRIMARY KEY (`Id`),  
  KEY `Alumno` (`Alumno`),  
  KEY `Asignatura` (`Asignatura`),  
  KEY `Grupo` (`Grupo`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;  
  
-----  
-- Table structure for `comunicados`  
-----  
DROP TABLE IF EXISTS `comunicados`;  
CREATE TABLE `comunicados` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `Emisor` varchar(30) NOT NULL,  
  `Receptor` varchar(30) NOT NULL,  
  `Mensaje` varchar(200) NOT NULL,  
  `Fecha` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',  
  `Leido` tinyint(1) NOT NULL DEFAULT '0',  
  PRIMARY KEY (`Id`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;  
  
-----  
-- Table structure for `deporte`  
-----  
DROP TABLE IF EXISTS `deporte`;  
CREATE TABLE `deporte` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `Nombre` varchar(45) NOT NULL,  
  `Horario` varchar(45) DEFAULT NULL,  
  `Monitor` varchar(30) DEFAULT NULL,  
  PRIMARY KEY (`Id`,`Nombre`),  
  KEY `Monitor` (`Monitor`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

```
-----  
-- Table structure for `examenes`  
-----  
DROP TABLE IF EXISTS `examenes`;  
CREATE TABLE `examenes` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `Asignatura` varchar(30) NOT NULL,  
  `Grupo` varchar(30) NOT NULL,  
  `Fecha` date DEFAULT NULL,  
  `Hora` time DEFAULT NULL,  
  `Duracion` varchar(5) DEFAULT NULL,  
  PRIMARY KEY (`Id`,`Asignatura`,`Grupo`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;  
  
-----  
-- Table structure for `familiar`  
-----  
DROP TABLE IF EXISTS `familiar`;  
CREATE TABLE `familiar` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `NumExp` varchar(30) NOT NULL DEFAULT '',  
  `DNI` varchar(9) NOT NULL,  
  `Telefono` varchar(9) NOT NULL,  
  `Direccion` varchar(45) NOT NULL,  
  `Nombre` varchar(45) NOT NULL,  
  `Apellidos` varchar(80) DEFAULT NULL,  
  PRIMARY KEY (`Id`,`NumExp`,`DNI`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;  
  
-----  
-- Table structure for `grupo`  
-----  
DROP TABLE IF EXISTS `grupo`;  
CREATE TABLE `grupo` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `Codigo` varchar(6) NOT NULL,  
  `Edad` int(10) unsigned NOT NULL,  
  PRIMARY KEY (`Id`,`Codigo`,`Edad`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

```
-----  
-- Table structure for `historico_borrados`  
-----  
DROP TABLE IF EXISTS `historico_borrados`;  
CREATE TABLE `historico_borrados` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `NumExp` varchar(30) NOT NULL,  
  `DNI` varchar(30) DEFAULT NULL,  
  `Tipo` varchar(30) NOT NULL,  
  `Fecha_Alta` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON  
UPDATE CURRENT_TIMESTAMP,  
  `Fecha_Baja` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',  
  `Usuario` varchar(9) DEFAULT NULL,  
  PRIMARY KEY (`Id`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

```
-----  
-- Table structure for `historico_modificaciones`  
-----  
DROP TABLE IF EXISTS `historico_modificaciones`;  
CREATE TABLE `historico_modificaciones` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `NumExp` varchar(30) NOT NULL,  
  `DNI` varchar(30) DEFAULT NULL,  
  `Accion` varchar(1000) NOT NULL,  
  `Tabla` varchar(30) NOT NULL,  
  `Fecha` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00' ON  
UPDATE CURRENT_TIMESTAMP,  
  `Usuario` varchar(9) DEFAULT NULL,  
  PRIMARY KEY (`Id`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

```
-----  
-- Table structure for `informes`  
-----  
DROP TABLE IF EXISTS `informes`;  
CREATE TABLE `informes` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `Tutor` varchar(30) NOT NULL,  
  `Familiar` varchar(30) NOT NULL,  
  `Alumno` varchar(30) NOT NULL,  
  `Informe` varchar(200) NOT NULL,  
  `Fecha` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00' ON  
UPDATE CURRENT_TIMESTAMP,  
  `Leido` tinyint(1) NOT NULL DEFAULT '0',  
  PRIMARY KEY (`Id`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

```
-----  
-- Table structure for `log_registro`  
-----  
DROP TABLE IF EXISTS `log_registro`;  
CREATE TABLE `log_registro` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `NumExp` varchar(30) COLLATE utf8_spanish_ci DEFAULT NULL,  
  `Usuario` varchar(9) COLLATE utf8_spanish_ci NOT NULL,  
  `Fecha` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,  
  PRIMARY KEY (`Id`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8  
COLLATE=utf8_spanish_ci;
```

```
-----  
-- Table structure for `monitor`  
-----  
DROP TABLE IF EXISTS `monitor`;  
CREATE TABLE `monitor` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `NumExp` varchar(30) NOT NULL DEFAULT '',  
  PRIMARY KEY (`Id`,`NumExp`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

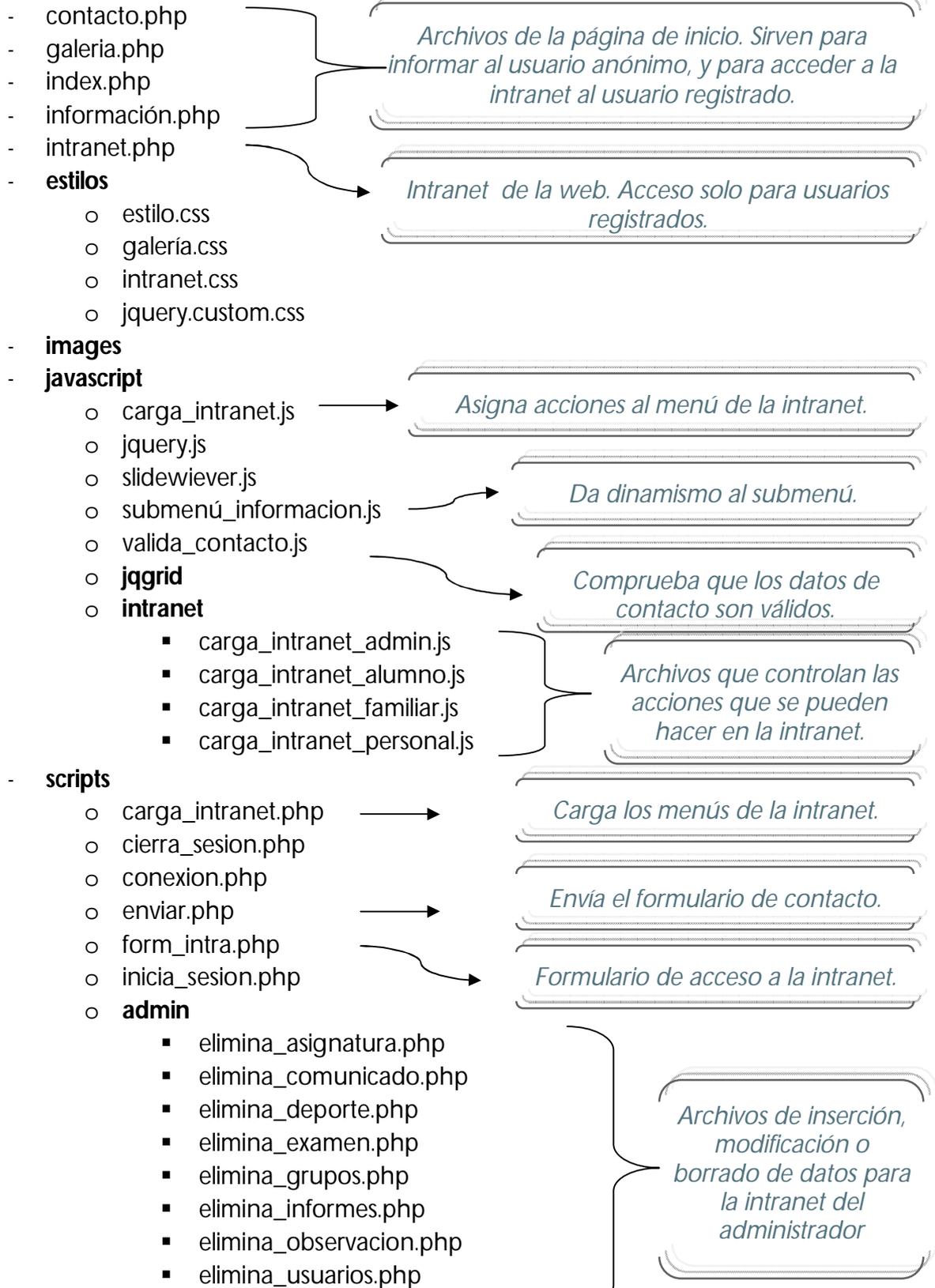
```
-----  
-- Table structure for `observaciones`  
-----  
DROP TABLE IF EXISTS `observaciones`;  
CREATE TABLE `observaciones` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `Emisor` varchar(30) NOT NULL,  
  `Receptor` varchar(30) NOT NULL,  
  `Mensaje` varchar(200) NOT NULL,  
  `Fecha` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00' ON  
UPDATE CURRENT_TIMESTAMP,  
  `Leido` tinyint(1) NOT NULL DEFAULT '0',  
  PRIMARY KEY (`Id`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

```
-- -----  
-- Table structure for `personal`  
-- -----  
DROP TABLE IF EXISTS `personal`;  
CREATE TABLE `personal` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `NumExp` varchar(30) NOT NULL DEFAULT '',  
  `DNI` varchar(9) NOT NULL,  
  `Direccion` varchar(45) NOT NULL,  
  `Fecha_nacimiento` datetime NOT NULL,  
  `Telefono` varchar(9) NOT NULL,  
  `Nombre` varchar(45) NOT NULL,  
  `Apellidos` varchar(80) DEFAULT NULL,  
  PRIMARY KEY (`Id`,`NumExp`,`DNI`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;  
  
-- -----  
-- Table structure for `practica`  
-- -----  
DROP TABLE IF EXISTS `practica`;  
CREATE TABLE `practica` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `Alumno` varchar(30) NOT NULL,  
  `Deporte` int(11) NOT NULL,  
  PRIMARY KEY (`Id`),  
  KEY `Alumno` (`Alumno`),  
  KEY `Deporte` (`Deporte`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;  
  
-- -----  
-- Table structure for `profesor`  
-- -----  
DROP TABLE IF EXISTS `profesor`;  
CREATE TABLE `profesor` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `NumExp` varchar(30) NOT NULL DEFAULT '',  
  `Titulacion` varchar(45) DEFAULT NULL,  
  PRIMARY KEY (`Id`,`NumExp`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

```
-- -----  
-- Table structure for `usuario`  
-- -----  
DROP TABLE IF EXISTS `usuario`;  
CREATE TABLE `usuario` (  
  `Id` int(11) NOT NULL AUTO_INCREMENT,  
  `NumExp` varchar(30) CHARACTER SET utf8 NOT NULL,  
  `Usuario` varchar(9) CHARACTER SET utf8 NOT NULL,  
  `Contrasenya` varchar(45) CHARACTER SET utf8 NOT NULL,  
  `Fecha_Alta` timestamp NULL DEFAULT NULL,  
  `Tipo` varchar(30) DEFAULT NULL,  
  PRIMARY KEY (`Id`,`NumExp`,`Usuario`)  
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=latin1;
```

## 9.5 ANEXO E: Estructura de archivos

### Internado



- form\_alumnos.php
  - form\_asignaturas.php
  - form\_deportes.php
  - form\_examenes.php
  - form\_familiares.php
  - form\_grupos.php
  - form\_personal.php
  - form\_usuarios.php
  - inserta\_asignaturas.php
  - inserta\_deporte.php
  - inserta\_examenes.php
  - inserta\_grupos.php
  - inserta\_usuarios.php
  - lista\_alumnos.php
  - lista\_asignaturas.php
  - lista\_comunicados.php
  - lista\_deportes.php
  - lista\_examenes.php
  - lista\_familiares.php
  - lista\_grupos.php
  - lista\_informes.php
  - lista\_observaciones.php
  - lista\_personal.php
  - lista\_usuarios.php
  - modifica\_alumno.php
  - modifica\_asignatura.php
  - modifica\_deporte.php
  - modifica\_examen.php
  - modifica\_familiar.php
  - modifica\_personal.php
  - modifica\_usuarios.php
- **intranet**
    - deportes\_alumno.php
    - elimina\_mensaje.php
    - envía\_mensaje.php
    - form\_alumno.php
    - form\_familiar.php
    - form\_personal.php
    - lee\_mensaje.php
    - lista\_datos.php
    - modifica\_notas.php

*Archivos que controlan las acciones que se pueden hacer en la intranet.*

## 9.6 ANEXO F: Manual de usuario

Podemos entrar en nuestra web mediante un navegador, en la página: <http://pion.disca.upv.es/internado>.

### 9.6.1 Página principal

Esta es nuestra página principal:



Ilustración 9-1 - Internado Meinston: Página principal

El resto de páginas de la página principal, a la que pueden acceder todos los usuarios, tienen la misma estructura, cambiando el contenido del cuerpo y el submenú lateral. Por ejemplo, podemos ver el menú del apartado información:

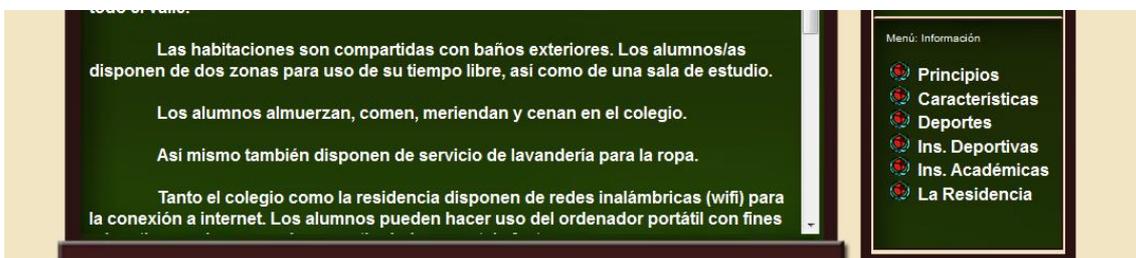


Ilustración 9-2 - Menú Información

La galería de imágenes sustituye el área de texto por una secuencia de imágenes del colegio:

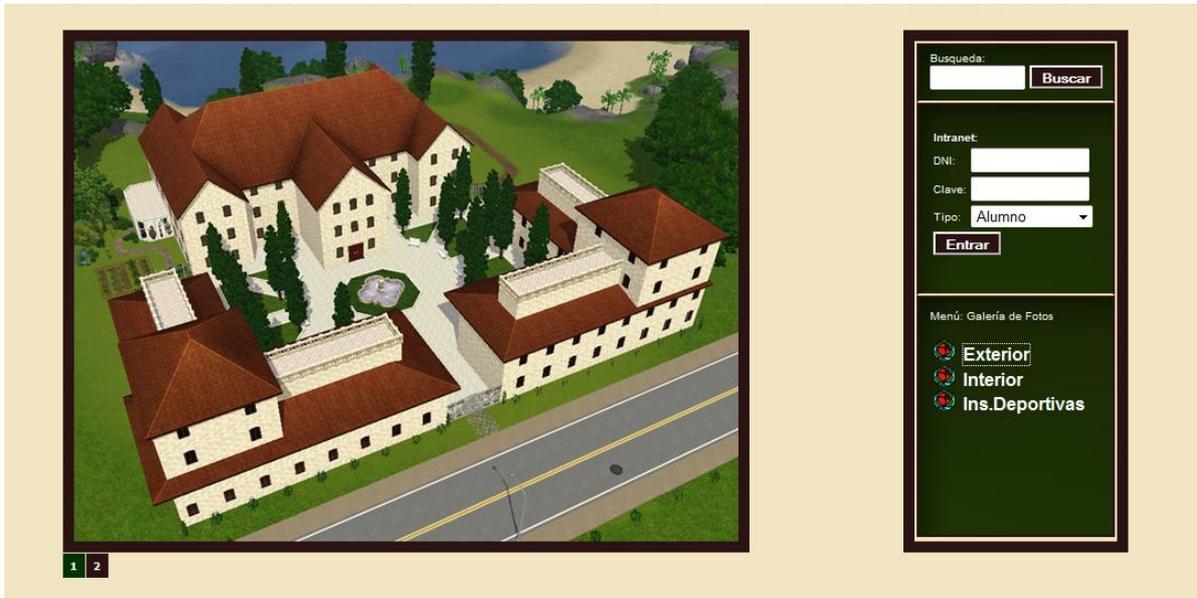


Ilustración 9-3 - Galería de imágenes

El último apartado de la página principal es el de contacto. En él podemos ver un formulario de solicitud de información:

The image shows a screenshot of a contact form. The title is 'Formulario de contacto:'. Below the title, there are five input fields: 'Nombre:' (one line), 'Apellidos:' (two lines), 'Teléfono:' (one line), 'E-mail:' (one line), and 'Descripción:' (a large text area). At the bottom left of the form, there is a small counter '0/250'. At the bottom right, there are two buttons: 'Enviar' and 'Borrar'.

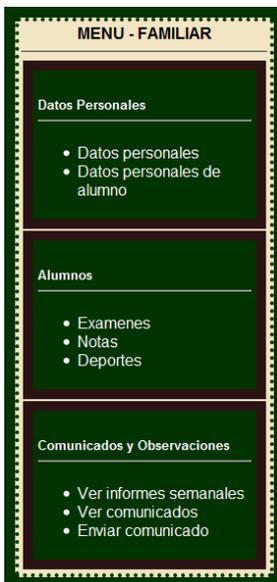
Ilustración 9-4 - Formulario de contacto

La intranet para usuarios registrados tiene el siguiente aspecto:



Ilustración 9-5 - Intranet Administrador

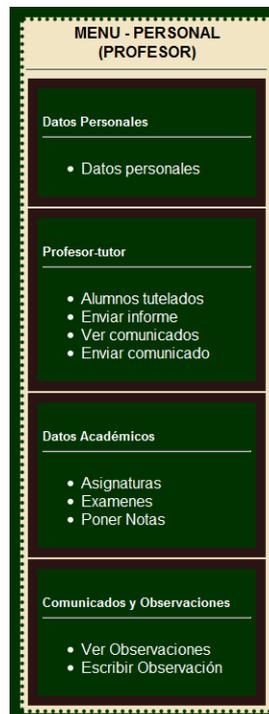
Los demás usuarios tendrán la misma intranet, cambiando el menú lateral. Veámoslo con unas imágenes:



9-8 - Menú Familiar



9-6 - Menú Alumno



9-9 - Menú Profesor

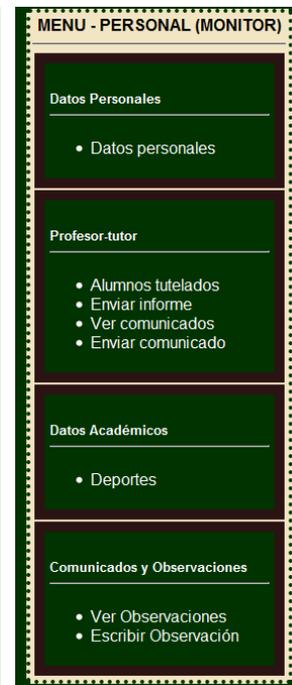


Ilustración 9-7 - Menú Personal

Para la información recuperada en cada uno de los apartados del menú se ha utilizado un formato de tablas dinámicas, como la que se muestra a continuación:

USUARIOS REGISTRADOS:

	NumExp ↑	Usuario	Contraseña	Fecha de Alta	Tipo
1	AD000001	admin	admin	20/04/2011	administrador
2	AD000002	12345678A	mantequilla	28/04/2011	administrador
3	AL000001	alumno	alumno	20/04/2011	alumno
4	AL000002	12345678U	anime	03/05/2011	alumno
5	AL000003	12345678O	monica	03/05/2011	alumno
6	AL000004	12345678P	monika	04/05/2011	alumno
7	AL000005	12345678W	manolo	10/05/2011	alumno
8	AL000006	12345678D	canguro	10/05/2011	alumno
9	AL000007	12345678V	rock	18/05/2011	alumno
10	AL000008	12345678L	musica	19/05/2011	alumno
11	AL000009	12345678K	marlin	19/05/2011	alumno
12	AL000010	12345678H	alone	19/05/2011	alumno
13	FA000001	familiar	familiar	20/04/2011	familiar
14	FA000002	12345678Q	rap	03/05/2011	familiar
15	FA000004	12345678E	cancion	03/05/2011	familiar
16	FA000005	12345678R	shakira	03/05/2011	familiar
17	FA000006	12345678M	arrebato	18/05/2011	familiar
18	PR000002	12345678C	chupachups	28/04/2011	personal
19	PR000004	12345678N	sermon	17/05/2011	personal
20	PR000005	12345678B	mundo	18/05/2011	personal

Página 1 de 2    20    Mostrando 1 - 20 de 21

Ilustración 9–10 - Tablas JQuery

Estas tablas nos permiten recuperar, manipular y borrar los datos en tiempo real.

#### 9.6.1.1 Formulario de acceso a la intranet (LOGIN)

En la página principal hay un apartado, en el menú de la derecha, que contiene un formulario con el que podemos acceder a la intranet mediante un DNI y una contraseña.

Este formulario cambiará a los datos del usuario si este ya ha iniciado sesión previamente.

Formulario de acceso a la intranet (LOGIN) con los siguientes campos:

- Intranet:
- DNI:
- Clave:
- Tipo:
- Botón:

Ilustración 9–12 - Formulario de acceso (LOGIN)

Acceso a la intranet con los siguientes elementos:

- Intranet:
- Bienvenid@: Jose Ramon El Familiar
- Enlaces: [Entrar en la intranet](#), [Cerrar Sesión](#)

Ilustración 9–11 - Acceso a la intranet

Al hacer clic, tanto en el botón Entrar, como en el enlace de la ilustración 9-11, Entrar en la intranet, se comprobarán los datos y se cargará la intranet del usuario e iniciará la sesión si no se había iniciado previamente.

Si los datos son erróneos o no se corresponden a los de la base de datos, se producirá un error.



Ilustración 9-13 - Error en inicio de sesión

### 9.6.2 Intranet

La intranet es el espacio privado de la web, reservado para los usuarios registrados en ella. Existe una intranet para cada tipo de usuario. Vamos a verlas una a una.

#### 9.6.2.1 Administrador

El administrador tiene poder total sobre los datos de los usuarios, por tanto su intranet le permite la modificación de cada uno de ellos.

##### 9.6.2.1.1 Usuarios

El administrador puede agregar nuevos usuarios, modificar los datos de los existentes y eliminarlos. Las contraseñas de usuario solo puede modificarlas él.

La página de usuarios es la siguiente:

USUARIOS REGISTRADOS:

	NumExp ↕	Usuario	Contraseña	Fecha de Alta	Tipo
1	AD000001	admin	admin	20/04/2011	administrador
2	AD000002	12345678A	mantequilla	28/04/2011	administrador
3	AL000001	alumno	alumno	20/04/2011	alumno
4	AL000002	12345678U	anime	03/05/2011	alumno
5	AL000003	12345678O	monica	03/05/2011	alumno
6	AL000004	12345678P	monika	04/05/2011	alumno
7	AL000005	12345678W	manolo	10/05/2011	alumno
8	AL000006	12345678D	canguro	10/05/2011	alumno
9	AL000007	12345678V	rock	18/05/2011	alumno
10	AL000008	12345678L	musica	19/05/2011	alumno
11	AL000009	12345678K	marilin	19/05/2011	alumno
12	AL000010	12345678H	alone	19/05/2011	alumno
13	FA000001	familiar	familiar	20/04/2011	familiar
14	FA000002	12345678Q	rap	03/05/2011	familiar
15	FA000004	12345678E	cancion	03/05/2011	familiar
16	FA000005	12345678R	shakira	03/05/2011	familiar
17	FA000006	12345678M	arrebato	18/05/2011	familiar
18	PR000002	12345678C	chupachups	28/04/2011	personal
19	PR000004	12345678N	sermon	17/05/2011	personal
20	PR000005	12345678B	undo	18/05/2011	personal

Página 1 de 2    Mostrando 1 - 20 de 21

Ilustración 9-14 - Administrador: Usuarios registrados

En ella, el administrador puede ver el usuario (DNI) y contraseña de cada uno de los usuarios, así como el tipo de usuario, la fecha de alta y el número de expediente, que será único para cada uno de ellos, se genera automáticamente y no es modificable.

A través de los espacios de escritura que podemos ver en la cabecera de la tabla, podemos realizar búsquedas más exhaustivas de los datos que hay en ellas. Por ejemplo, si quisiéramos ver solamente los alumnos:



	NumExp ↕	Usuario	Contraseña	Fecha de Alta	Tipo
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	alum
1	AL000001	alumno	alumno	20/04/2011	alumno
2	AL000002	12345678U	anime	03/05/2011	alumno
3	AL000003	12345678O	monica	03/05/2011	alumno
4	AL000004	12345678P	monika	04/05/2011	alumno
5	AL000005	12345678W	manolo	10/05/2011	alumno
6	AL000006	12345678D	canguro	10/05/2011	alumno
7	AL000007	12345678V	rock	18/05/2011	alumno
8	AL000008	12345678L	musica	19/05/2011	alumno
9	AL000009	12345678K	marilin	19/05/2011	alumno
10	AL000010	12345678H	alone	19/05/2011	alumno

Mostrando 1 - 10 de 10

Ilustración 9-15 - Búsqueda en tablas

Mediante los botones de las tablas, podremos realizar las modificaciones pertinentes en cada una de ellas.



Ilustración 9-16 - Botones de la intranet

En particular en esta tabla podremos Insertar usuarios, modificar los existentes o borrarlos.

Insertemos uno, por ejemplo de tipo Alumno.



Ilustración 9-17 - Administrador: Insertando usuario

Para introducir las fechas nos ayudaremos de un calendario.



Ilustración 9-18 – Calendario

Ilustración 9-19 - DatePicker

Si algún dato obligatorio no lo rellenáramos, la función encargada nos lo recordaría mediante un **diálogo**.

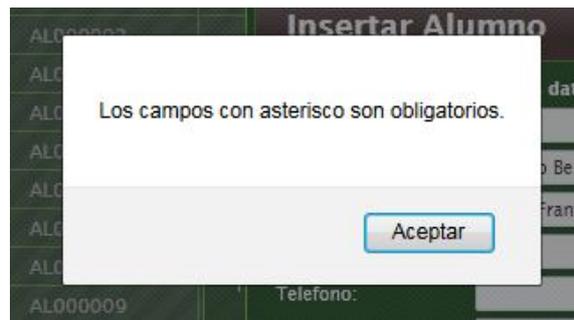


Ilustración 9-20 – Diálogo

Una vez insertado, se nos notificará y se recargará la tabla donde podremos ver el nuevo usuario:

12	AL000010	12345678H	alone	19/05/2011	alumno
13	AL000011	12345678F	pass	07/06/2011	alumno
14	FA000001	familiar	familiar	20/04/2011	familiar

Ilustración 9-21 - Administrador: Usuario insertado

Desde esta tabla, el administrador podrá modificar el usuario y contraseña de los usuarios.

Para eliminar se nos preguntará siempre si estamos seguros, y al hacerlo quedará constancia de ello en una tabla **historico\_borrados**, que nos dirá quien lo borró, y qué día.

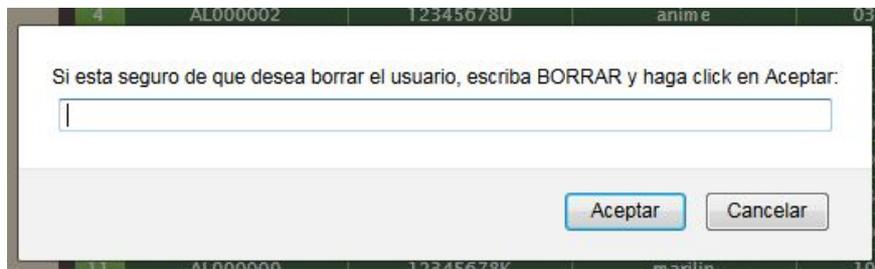


Ilustración 9–22 - Confirmación borrado

### 9.6.2.1.2 Alumnos

En la tabla de alumnos podemos modificar, insertar y borrar alumnos, así como cambiarles el tutor, el grupo y el familiar.

**ALUMNOS:**

	NumExp ↕	DNI	Nombre	Apellidos	Dirección	Fecha de nacimiento	Edad	Telefono
1	AL000001	alumno	Alum noNom bre	Alum noApellidos	Calle, Ciudad	13/04/1994	16	
2	AL000002	12345678U	Ayum i	Nuse	Japon	01/01/1970	41	
3	AL000003	12345678O	Monica		Moncada	17/05/1994	16	
4	AL000004	12345678P	Zaira	Naranjo	Barcelona	20/05/1991	19	65156151
5	AL000005	12345678W	Tatiana	Creative	C/Marbella	30/12/1998	12	633632565
6	AL000006	12345678D	Laia	Cortes Garcia	C/Del Molino	01/01/1970	41	69696964
7	AL000007	12345678V	Alexia	Aleluya	C/de atras	23/02/2004	6	
8	AL000009	12345678K	Jesus		C/pepinillo	03/05/2000	11	
9	AL000010	12345678H	Juan	Perico	c/Vistosa	02/05/2001	10	

Mostrando 1 - 9 de 9

Ilustración 9–23 - Administrador: Alumnos



Ilustración 9–24 – Cambiar familiar

Ilustración 9–25 - Modificar Alumno

### 9.6.2.1.3 Personal

En este apartado podremos insertar, modificar y eliminar personal, ya sea profesor o monitor y asignarle o quitarle asignaturas o deportes, respectivamente.



Ilustración 9-27 - Administrador: Personal



Ilustración 9-26 - Asignando deportes

### 9.6.2.1.4 Familiares

Igual que al personal, podremos modificar, añadir y eliminar, y en este caso, asignar alumnos.



Ilustración 9-28 - Administrador: Familiares

### 9.6.2.1.5 Grupos

De los grupos, el administrador podrá añadir y eliminar.

**GRUPOS:**

	Codigo	Edad ↑
1	1A-PRE	4
2	1B-PRE	4
3	2B-PRE	5
4	1B-PRE	5
5	1A-PRI	6
6	1B-PRI	6
7	2A-PRI	7
8	2B-PRI	7
9	3B-PRI	8
10	3A-PRI	8
11	4B-PRI	9
12	4A-PRI	9
13	5A-PRI	10
14	6A-PRI	11
15	1A-ESO	12
16	2A-ESO	13
17	3A-ESO	14
18	4A-ESO	15
19	1A-BAC	16
20	2A-BAC	17

Ilustración 9-29 - Administrador: Grupos

### 9.6.2.1.6 Asignaturas

Se podrán añadir, eliminar y modificar las asignaturas.

**ASIGNATURAS:**

	Nombre	Grupo	Profesor	Horario
1	Conocimiento del Medio	1B-PRI	PR000004	
2	Matematicas	1A-PRE	PR000004	L-X-V 8:30-9:20
3	Matematicas	1A-ESO	PR000004	

Mostrando 1 - 3 de 3

Ilustración 9-30 - Administrador: Asignaturas

### 9.6.2.1.7 Exámenes

Igual que las asignaturas, se podrá cambiar sus datos, añadir y borrar.

**EXAMENES:**

	Asignatura	Grupo	Fecha	Hora	Duracion
1	Conocimiento del Medio	1B-PRI	30/05/2011	13:00:00	2h
			05/2011	13:00:00	
			05/2011	09:00:00	30m

Mostrando 1 - 3 de 3

**Modificar Examen** ✕

Asignatura\*: Conocimiento del Medio - Grupo: 1B-PRI

Grupo\*: 1B-PRI - Edad: 6

Fecha: 2011-05-30

Hora: 13:00:00

Duracion: 2h

Ilustración 9-31 - Administrador: Exámenes

## 9.6.2.1.8 Deportes

De ellos podremos modificar sus datos, añadir y borrar, así como ver los alumnos que se hayan apuntado al deporte seleccionado.



Ilustración 9-32 - Administrador: Deportes

## 9.6.2.1.9 Comunicados, Observaciones e Informes

Podremos ver quién envía los mensajes y quien los recibe, si está leído y borrarlos.

	Emisor	Receptor	Mensaje	Fecha	Leído
1	PR000002	AL000002	fseaesf e	01/06/2011	Si
2	PR000002	AL000002	zsd fsd	28/05/2011	No
3	PR000004	AL000010	asdasd	28/05/2011	No
4	PR000002	AL000002	asdasdd	28/05/2011	No
5	PR000006	AL000010	¿Como te encuentras hoy? Si necesitas algo comunicalo. Un saludo.	19/05/2011	No

Ilustración 9-33 - Administrador: Comunicados, observaciones e informes

### 9.6.2.2 Alumno

El alumno no puede modificar sus datos, solo puede consultarlos. Sin embargo sí que puede inscribirse en deportes y enviar comunicados a su tutor o a su familiar, así como leer y borrar los que recibe.

#### 9.6.2.2.1 Datos personales

Aunque no pueda modificarlos, el alumno podrá ver sus datos personales.

**DATOS PERSONALES:**

NumExp	DNI	Nombre	Apellidos	Fecha nacimier	Direccion	Telefono	Grupo	Familiar	Tutor
AL000002	12345678U	Ayumi	Nuse	01/01/1970	Japon		4B-PRI	Jose Ram on El Familiar	Manolo El Del Bombo

Página 0 de 1

Ilustración 9-34 - Alumno: Datos personales

#### 9.6.2.2.2 Asignaturas, Exámenes y notas

El alumno puede consultar las asignaturas que tiene asignadas de acuerdo a su grupo actual, los exámenes de esas asignaturas y las notas que le hayan puesto los profesores.

**ASIGNATURAS:**

	Nombre	Grupo	Profesor	Horario
1	Matematicas	1A-PRE	Marisa Manguito	L-X-V 8:30-9:20

Página 1 de 1    20    Mostrando 1 - 1 de 1

Ilustración 9-35 - Alumno: Asignaturas

**EXAMENES:**

	Asignatura	Grupo	Fecha	Horario	Duracion
1	Matematicas	1A-PRE	08/05/2011	09:00:00	30m

Página 1 de 1    20    Mostrando 1 - 1 de 1

Ilustración 9-36 - Alumno: Exámenes

**NOTAS:**

	Asignatura	Grupo	Nota
1	Matematicas	1A-PRE	10

Página 1 de 1    20    Mostrando 1 - 1 de 1

Ilustración 9-37 - Alumno: Notas

### 9.6.2.2.3 Deportes

El alumno puede inscribirse y borrarse de, cómo máximo, dos deportes. Si intenta hacerlo en más de dos, el sistema le avisará del error.



Ilustración 9-38 - Alumno: Deportes

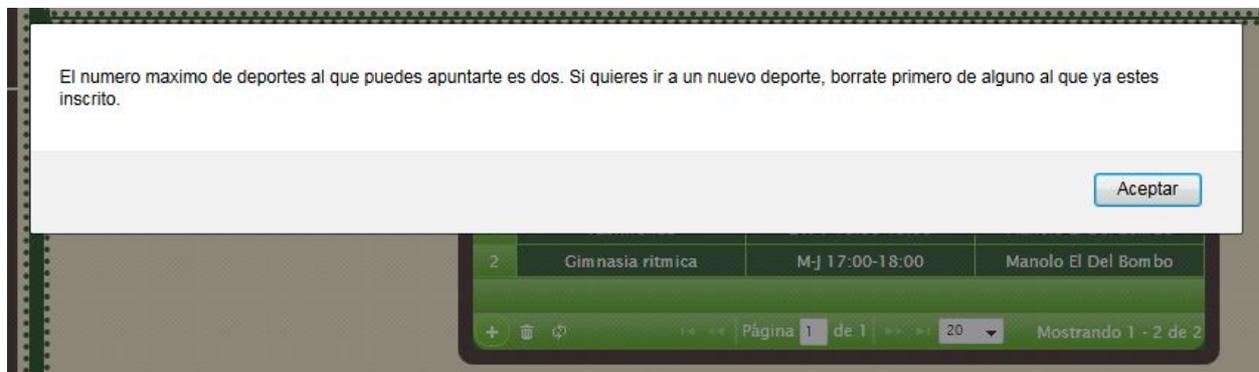


Ilustración 9-39 - Error, máximo de deportes inscritos

### 9.6.2.2.6 Observaciones y comunicados

Un alumno podrá enviar un comunicado a su tutor y a su familiar, y leer los que reciba, así como borrarlos.

Cuando un mensaje sea leído, la columna **Leído**, cambiará a **Sí** en ese mismo momento.



Ilustración 9-40 - Alumno: Observaciones y comunicados

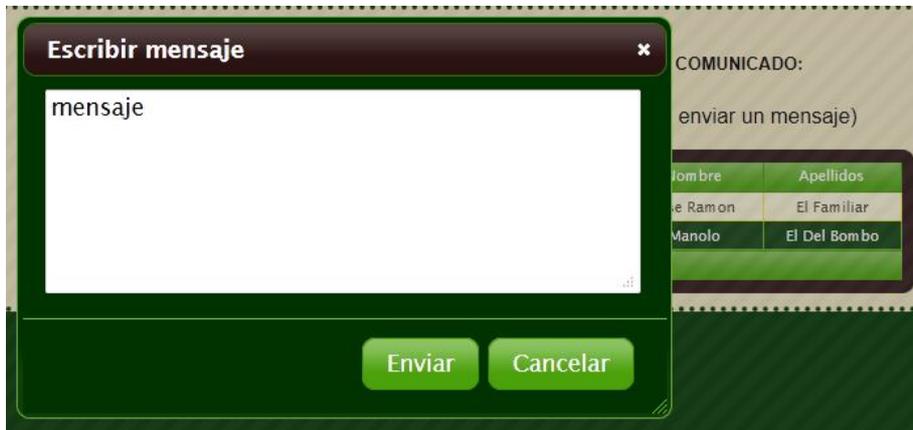


Ilustración 9-41 - Escribiendo mensaje

### 9.6.2.3 Familiar

El familiar tiene a su cargo uno o varios alumnos, de los que podrá modificar sus datos personales, y consultar sus datos académicos. También podrá enviar y recibir comunicados, así como informes del tutor de cada uno de sus alumnos.

#### 9.6.2.3.1 Datos personales

Podrá modificar sus datos y los de los alumnos de los que se responsabilice.



Ilustración 9-43 - Familiar: Datos personales



Ilustración 9-42 - Familiar: Datos personales de los alumnos

## 9.6.2.3.2 Exámenes, notas y deportes

Un familiar podrá consultar los datos académicos de los alumnos, sin modificar nada.

**EXAMENES DE LOS ALUMNOS:**

	NumExp ↑	Nombre	Apellidos	Grupo	Asignatura	Fecha	Hora	Duracion
1	AL000007	Alexia	Aleluya	1A-PRE	Matematicas	08/05/2011	09:00:00	30m
2	AL000010	Juan	Perico	1A-PRE	Matematicas	08/05/2011	09:00:00	30m

Mostrando 1 - 2 de 2

Ilustración 9-44 - Familiar: Exámenes

**NOTAS DE LOS ALUMNOS:**

	NumExp ↑	Nombre	Apellidos	Grupo	Asignatura	Nota
1	AL000007	Alexia	Aleluya	1A-PRE	Matematicas	4
2	AL000010	Juan	Perico	1A-PRE	Matematicas	6

Mostrando 1 - 2 de 2

Ilustración 9-45 - Familiar: Notas

## 9.6.2.3.3 Comunicados e Informes

El familiar puede recibir comunicados de sus alumnos, así como enviarlos e informes de los tutores de los mismos.

**INFORMES:**

**Mensaje** ✕

Quisque diam nibh, sollicitudin nec bibendum ut, vestibulum vel nibh. Vivamus ut libero eu turpis fringilla gravida. Aliquam consectetur egestas velit, id ullamcorper quam scelerisque eu. Phasellus

Borrar Mensaje

Informe	Fecha ↓	Leido
Quisque diam nibh, sollicitudin nec bibendum ut, vestibulum vel nibh. Vivamus ut libero eu turpis fringilla gravida. Aliquam consectetur egestas velit, id ullamcorper quam scelerisque eu. Phasellus	07/06/2011	Si

Mostrando 1 - 1 de 1

Ilustración 9-46 - Familiar: Comunicados e informes

### 9.6.2.4 Personal

El personal es aquella persona que trabaja en el centro. Pueden ser profesores, que impartirán asignaturas, o monitores que enseñaran algún deporte. Los dos tipos podrán ser tutores de los alumnos y por tanto enviarles comunicados e informes a sus familiares.

#### 9.6.2.4.1 Datos personales

El personal podrá modificar sus propios datos personales.



Ilustración 9-47 - Personal: Datos personales

#### 9.6.2.4.2 Tutor

Como tutor, el personal puede ver sus alumnos tutelados, enviarles comunicados y recibirlos de ellos, y enviar informes a los familiares.

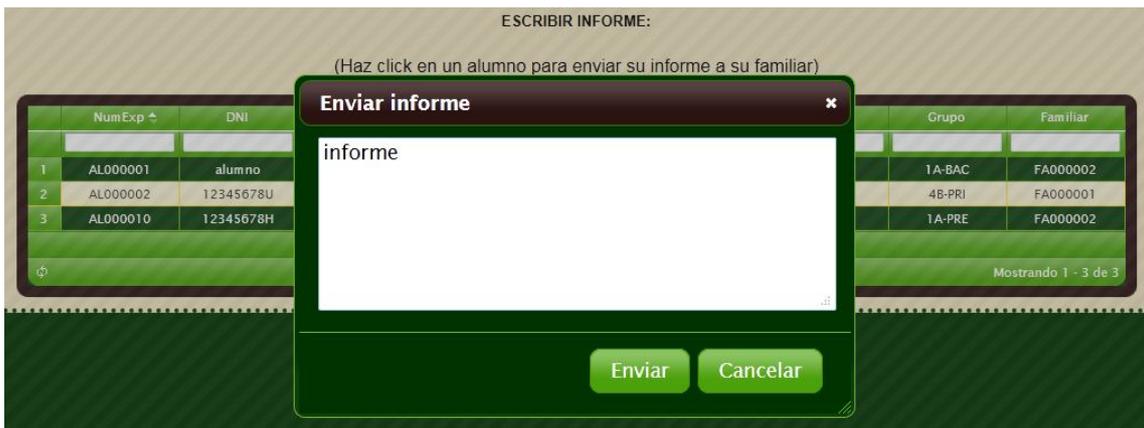


Ilustración 9-48 - Personal: Tutor

## 9.6.2.4.3 Monitor: Deportes

El monitor podrá ver los deportes que le han sido asignados, así como el número de alumnos apuntados.

**DEPORTES ASIGNADOS:**

	Nombre ↕	Horario	Alum nos
1	Atletismo	L-X-V 12:00-13:00	0
2	Baloncesto	M-J 12:00-13:00	0
3	Gimnasia rítmica	M-J 17:00-18:00	0
4	Judo	L-X-V 17:00-18:00	0
5	Natacion	M-J 18:00-20:00	0
6	Taekwondo	L-X-V 18:00-19:00	2

Página 1 de 1    20    Mostrar

Ilustración 9-49 - Monitor: Deportes

## 9.6.2.4.4 Profesor: Asignaturas, exámenes y notas

El profesor tiene unas asignaturas asignadas, con unos exámenes, y de ellos tendrá que poner nota a sus alumnos.

**ASIGNATURAS ASIGNADAS:**

	Nombre ↕	Horario	Grupo	Num Alumnos
1	Conocimiento del Medio		1B-PRI	0
2	Matematicas		1A-ESO	0
3	Matematicas	L-X-V 8:30-9:20	1A-PRE	6

Página 1 de 1    20    Mostrando 1 - 3 de 3

Ilustración 9-50 - Profesor: Asignaturas, exámenes y notas

**ASIGNATURAS ASIGNADAS:**

	Nombre ↕	Horario	Grupo	Num Alumnos
1	Con			
2				
3				

**Calificar** ✕

Rellena los siguientes datos:

AL000003-> , Monica:

AL000004-> Naranjo, Zaira:

AL000005-> Creative, Tatiana:

AL000007-> Aleluya, Alexia:

AL000009-> , Jesus:

AL000010-> Perico, Juan:

Ilustración 9-51 - Poniendo notas

Si la asignatura no tuviese alumnos matriculados, el sistema nos avisaría.

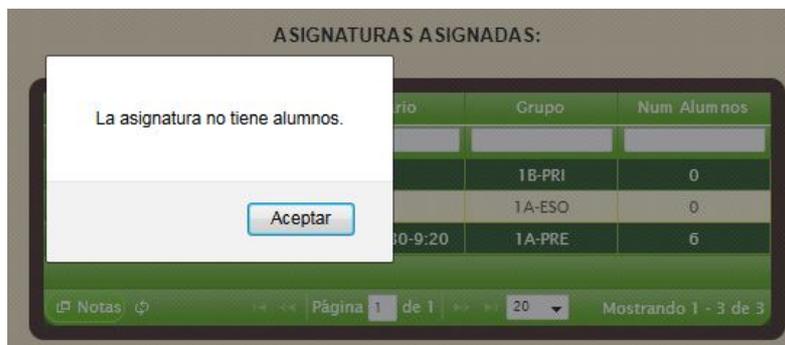


Ilustración 9-52 - Profesor: Asignatura sin alumnos

#### 9.6.2.4.5 Observaciones

El personal puede enviar observaciones a los alumnos que pertenezcan a sus asignaturas o deportes, así como recibirlos de ellos.

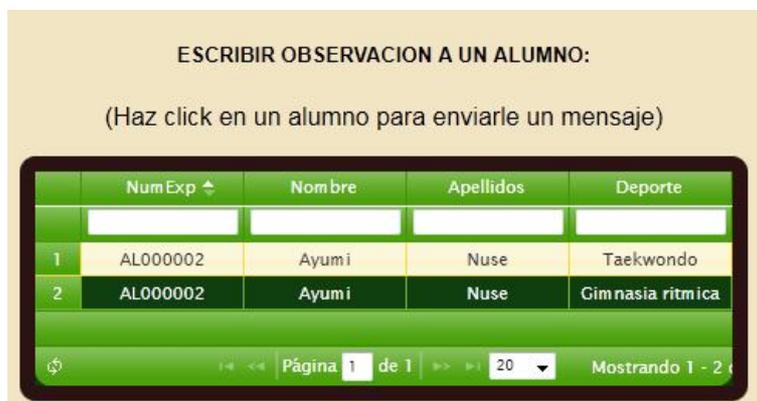


Ilustración 9-53 - Personal: Observaciones

## 9.5 ANEXO G: Índice de Ilustraciones

- COLEGIO MEINSTON	7
- DIAGRAMA DE CLASES	25
- COMUNICACIONES	26
- CASOS DE USO, ACTORES	27
- CASO DE USO: ANÓNIMO	28
- CASO DE USO: USUARIO REGISTRADO	28
- CASO DE USO: FAMILIAR	29
- CASO DE USO: ALUMNO	30
- CASO DE USO: PROFESOR	31
- CASO DE USO: ADMINISTRADOR	32
- INTERFAZ DE USUARIO, VISTA 1	34
- INTERFAZ DE USUARIO, VISTA 2	35
- DIAGRAMA DE COMPONENTES	36
- DIAGRAMA ENTIDAD-RELACIÓN	37
- BASE DE DATOS	54
- BASE DE DATOS: INTERNADO	55
- COMPROBACIÓN DE ENLACES ROTOS	56
- VALIDACIÓN DE HTML	57
- VALIDACIÓN DE CSS	57
- COMPROBACIÓN SOBRE MÓVIL, PÁGINA PRINCIPAL	58
- COMPROBACIÓN SOBRE MÓVIL, INTRANET	58
- FIREFOX: PÁGINA PRINCIPAL	59
- FIREFOX: INTRANET	59
- INTERNET EXPLORER: PÁGINA PRINCIPAL	60
- INTERNET EXPLORER: INTRANET	60
- GOOGLE CHROME: PÁGINA PRINCIPAL	61
- GOOGLE CHROME: INTRANET	61
- OPERA: PÁGINA PRINCIPAL	62
- OPERA: INTRANET	62
- RESOLUCION:1366X768	63
- RESOLUCIÓN: 1280X1024	63
- RESOLUCIÓN: 1024X768	63
- INTERNADO MEINSTON: PÁGINA PRINCIPAL	96
- MENÚ INFORMACIÓN	96
- GALERÍA DE IMÁGENES	97
- FORMULARIO DE CONTACTO	97
- INTRANET ADMINISTRADOR	98
- MENÚ ALUMNO	98
- MENÚ PERSONAL	98
- MENÚ FAMILIAR	98
- MENÚ PROFESOR	98
- TABLAS JQUERY	99
- ACCESO A LA INTRANET	99
- FORMULARIO DE ACCESO (LOGIN)	99
- ERROR EN INICIO DE SESIÓN	100
- ADMINISTRADOR: USUARIOS REGISTRADOS	100
- BÚSQUEDA EN TABLAS	101
- BOTONES DE LA INTRANET	101
- ADMINISTRADOR: INSERTANDO USUARIO	101
- CALENDARIO	102
- DATEPICKER	102
- DIÁLOGO	102
- ADMINISTRADOR: USUARIO INSERTADO	102

- CONFIRMACIÓN BORRADO	103
- ADMINISTRADOR: ALUMNOS	103
- CAMBIAR FAMILIAR ILUSTRACIÓN 9-25 - MODIFICAR ALUMNO	103
- ASIGNANDO DEPORTES	104
- ADMINISTRADOR: PERSONAL	104
- ADMINISTRADOR: FAMILIARES	105
- ADMINISTRADOR: GRUPOS	105
- ADMINISTRADOR: ASIGNATURAS	106
- ADMINISTRADOR: EXÁMENES	106
- ADMINISTRADOR: DEPORTES	107
- ADMINISTRADOR: COMUNICADOS, OBSERVACIONES E INFORMES	107
- ALUMNO: ASIGNATURAS	108
- ALUMNO: EXÁMENES	108
- ALUMNO: NOTAS	108
- ALUMNO: DATOS PERSONALES	108
- ALUMNO: DEPORTES	109
- ERROR, MÁXIMO DE DEPORTES INSCRITOS	109
- ALUMNO: OBSERVACIONES Y COMUNICADOS	109
- ESCRIBIENDO MENSAJE	110
- FAMILIAR: DATOS PERSONALES	110
- FAMILIAR: DATOS PERSONALES DE LOS ALUMNOS	110
- FAMILIAR: NOTAS	111
- FAMILIAR: COMUNICADOS E INFORMES	111
- FAMILIAR: EXÁMENES	111
- PERSONAL: DATOS PERSONALES	112
- PERSONAL: TUTOR	112
- MONITOR: DEPORTES	113
- PROFESOR: ASIGNATURAS, EXÁMENES Y NOTAS	113
- PONIENDO NOTAS	113
- PROFESOR: ASIGNATURA SIN ALUMNOS	114
- PERSONAL: OBSERVACIONES	114