

Análisis, diseño e implementación de una aplicación Web para el seguimiento del proceso académico haciendo uso de la arquitectura .NET

Universitat Oberta de Catalunya
Ingeniería técnica de Informática de Sistemas

Alumno: Omar Benjumea Gómez
Consultor : David Gañán Jiménez

Trabajo final de carrera
Curso 2005-2006 (09/01/2005)

RESUMEN

Análisis, diseño e implementación de una aplicación Web para el seguimiento del proceso académico haciendo uso de la arquitectura .NET

En este trabajo final de carrera se ha desarrollado una aplicación Web para el seguimiento vía Internet del proceso académico de los alumnos, tanto por ellos mismos como por parte de sus padres. La aplicación se ha implementado usando la arquitectura .NET de Microsoft para el desarrollo y la tecnología ADO para el acceso a datos. Todo ello se ha realizado usando el lenguaje de programación orientado a objetos C#. El trabajo perseguía principalmente dos objetivos: por un lado realizar todo el ciclo de vida de la aplicación Web, haciendo uso de los conocimientos adquiridos a lo largo de la carrera en diferentes asignaturas, y aprender a trabajar con el entorno orientado a objetos de Microsoft llamado .NET y con la tecnología ADO.NET para acceder a las bases de datos. En la memoria se describen todas las etapas llevadas a cabo para crear la aplicación: captura de requisitos, análisis, diseño y implementación, y se puede apreciar claramente como cada una de estas etapas nos encamina hacia la siguiente. También se hace una breve explicación introductoria de las tecnologías usadas, centrándonos especialmente en ASP.NET. En el último capítulo del documento se analizan los resultados obtenidos y cómo el desarrollo de la aplicación nos ha permitido aprender la tecnología propuesta y alcanzar todos los objetivos iniciales del trabajo. Finalmente se presentan algunos futuros cambios y añadidos que se podrían realizar para mejorar el resultado obtenido.

ÍNDICE DE CONTENIDOS

1. Introducción	5
1.1 Contexto y justificación del trabajo	5
1.2 Objetivos del trabajo	6
1.3 Enfoque y método seguido	6
1.4 Planificación del trabajo	7
1.5 Productos obtenidos	9
1.6 Descripción del resto de capítulos	10
2. Tecnologías empleadas	11
2.1 Microsoft.NET	11
2.2 ASP.NET	12
2.3 ADO.NET	13
2.4 C#	13
3. Análisis de requisitos	14
3.1 Visión general	14
3.2 Requisitos funcionales	14
3.3 Requisitos de la interfaz de usuario	16
3.3.1 Identificación de las restricciones técnicas	16
3.3.2 Perfiles de usuario	16
3.3.3 Documentación de las tareas	17
3.4 Especificaciones de la usabilidad	18
3.5 Casos de uso	19
3.6 Diagrama de clases	29
4. Diseño	30
4.1 Diseño de la implementación de los casos de uso	30
4.2 Diseño de la persistencia	40
4.3 Diseño de la interfaz	42
5. Implementación	57
6. Conclusiones	59
6.1 Conclusión y objetivos alcanzados	59
6.2 Recomendaciones	59
7. Glosario	61
8. Bibliografía	62
8.1 Libros usados	62
8.2 Fuentes de Internet	62

ÍNDICE DE FIGURAS

Tabla de planificación del proyecto	8
Esquema del funcionamiento del ASP	12
Diagrama de los casos de uso de padre y alumno	19
Diagrama de los casos de uso del profesor	20
Diagrama de los casos de uso del administrador	20
Diagrama estático de clases	29
Diagrama de colaboración de la creación de nuevos usuarios	32
Diagrama de la base de datos	40
Mapa de la aplicación web	42
Diseño de la pantalla de login	43
Diseño de la pantalla de cambio de contraseña	44
Diseño de la pantalla principal del administrador	45
Diseño de la pantalla de creación de nuevos usuarios	46
Diseño de la pantalla de borrado de usuarios	47
Diseño de la pantalla de creación de nuevas aulas	48
Diseño de la pantalla principal del profesor	49
Diseño de la pantalla de añadir alumnos a un aula	50
Diseño de la pantalla de crear nuevos mensajes	51
Diseño de la pantalla de insertar o modificar notas	52
Diseño de la pantalla de insertar faltas de asistencia	53
Diseño de la pantalla principal del padre o alumno	54
Diseño de la pantalla de consulta de mensajes	55
Diseño de la pantalla de consulta de notas	56
Diseño de la pantalla de consulta de faltas de asistencia	56

1 INTRODUCCIÓN

1.1 Contexto y justificación del trabajo

Se ha escogido esta aplicación como tema del trabajo final de carrera por la interesante oportunidad que nos ofrece para conocer las posibilidades de la plataforma .NET de Microsoft. Hay que aclarar que la obtención de la aplicación dista mucho de ser el objetivo principal del trabajo. Dicho objetivo es aprender a trabajar con el entorno orientado a objetos de Microsoft y, más concretamente, explorar el potencial que presenta esta plataforma a la hora de desarrollar páginas Web interactivas y aplicaciones de servidor. También resulta especialmente atractiva la posibilidad de utilizar y asimilar una tecnología de enlace con bases de datos como ADO.NET, ya que es cada vez más común la necesidad de usar servidores de bases de datos en las aplicaciones.

El hecho de que el objetivo central sea aprender estas tecnologías nos obliga a buscar una aplicación sencilla, que nos permita simplificar lo máximo posible las fases de análisis y diseño para centrarnos en la parte que más nos interesa: la implementación.

La aplicación que se ha escogido para desarrollar permite el seguimiento de la trayectoria académica de los alumnos, tanto por ellos mismos como por sus padres, a través de Internet. Para ello se pueden consultar las calificaciones, ver las faltas de asistencia o leer los mensajes escritos por los profesores. Aunque ya existen numerosas aplicaciones Web con funcionalidades semejantes, la principal novedad aportada por esta aplicación es la posibilidad que ofrece de seguir la evolución académica de los hijos a través de Internet.

1.2 Objetivos del trabajo

El objetivo principal del trabajo es alcanzar un buen conocimiento de la plataforma .NET de Microsoft, concretamente utilizando el lenguaje de programación C# y aprender a utilizar la tecnología ADO.NET.

Para ello se plantea la realización de todo el ciclo de vida de una aplicación Web usando las tecnologías que se desea aprender. La aplicación permitirá seguir el desarrollo académico de los alumnos a través de Internet.

El desarrollo de la aplicación crea los siguientes objetivos secundarios:

- Definir detalladamente los requisitos funcionales de la aplicación.
- Confeccionar un análisis preciso de la aplicación.
- Realizar el diseño utilizando los conceptos de la arquitectura .NET.
- Implementar la aplicación.

1.3 Enfoque y método seguido

El método seguido para la realización del trabajo ha venido marcado por las limitaciones de tiempo y la necesidad de cumplir los plazos de entregas. Este hecho, unido a las entregas intermedias que se tenían que hacer de las diferentes PACs, ha provocado la decisión de dividir el proyecto en tres grandes bloques o tareas:

- Captura de requisitos, análisis y diseño de la aplicación.
- Implementación de la interfaz de usuario.
- Implementación de las funcionalidades de la aplicación.

Dentro de cada uno de estos tres bloques se ha seguido un proceso paralelo en su realización, debido a la existencia de tres tipos de usuarios muy diferenciados por las tareas que realizarán en la aplicación: el administrador, los profesores y los alumnos/padres. A grandes rasgos se podría decir que las tareas del administrador serán las de definir al resto de usuarios, los profesores realizarán siempre una tarea de inserción de datos y los alumnos y padres tareas de consultas de datos. Por ello el orden seguido en cada uno de los bloques definidos anteriormente ha sido:

- 1º Tareas relacionadas con el administrador
- 2º Tareas relacionadas con los profesores
- 3º Tareas relacionadas con los alumnos y padres

Esto ha facilitado también la etapa de pruebas realizada en cada una de las fases de la implementación, ya que podíamos hacer las pruebas con datos insertados directamente desde la aplicación sin necesidad de trabajar desde el servidor de la base de datos.

1.4 Planificación del trabajo

La planificación temporal del trabajo ha venido marcada por las fechas de entrega fijadas de las diferentes PACs. En la siguiente tabla se muestra con detalle esta planificación:

Tarea	Duración	Inicio	Fin
Elección del TFC	3	23/9/05	25/9/05
Redacción del plan de trabajo	3	25/9/05	27/9/05
Entrega de la PAC1	28/9/05		
Análisis y diseño de la base de datos (BD)	5	29/9/05	3/10/05
Creación de la BD	3	4/10/05	6/10/05
Análisis y diseño de la aplicación	5	7/10/05	11/10/05
Desarrollo de la interfaz	21	12/10/05	1/11/05
Principal	2	12/10/05	13/10/05
Login	1	14/10/05	14/10/05
Pantallas de administrador	6	15/10/05	20/10/05
Pantallas de profesor	4	21/10/05	24/10/05
Pantallas de alumno	4	25/10/05	28/10/05
Pantallas de padre	4	29/10/05	1/11/05
Entrega de la PAC2	2/11/05		
Implementación de funcionalidades	37	2/11/05	7/12/05
Principal y login (+ pruebas)	5	2/11/05	6/11/05
Pantallas de administrador (+ pruebas)	8	7/11/05	14/11/05
Pantallas de profesor (+ pruebas)	8	15/11/05	22/11/05
Pantallas de alumno (+ pruebas)	8	23/11/05	30/11/05
Pantallas de padre (+ pruebas)	8	1/12/05	8/12/05
Etapas de pruebas	4	9/12/05	12/12/05
Entrega de la PAC3	12/12/05		
Redacción de la memoria	15	13/12/05	27/12/06
Creación de la presentación virtual	13	28/12/05	9/1/06
Entrega final	9/1/06		

1.5 Productos obtenidos

Como resultado del trabajo se han obtenido dos productos: por un lado el código de la aplicación Web implementada y por el otro toda la documentación generada en las fases de análisis y diseño.

El código se divide en los siguientes archivos:

- Los archivos .aspx donde está el código que define los Web Forms o interfaz gráfica de cada página
Los ficheros son: *Añadir_alumno.aspx*, *Añadir_faltas.aspx*, *Añadir_mensajes.aspx*, *consultar_mensajes.aspx*, *CrearAula.aspx*, *Default.aspx*, *eliminarusuario.aspx*, *insertar_notas.aspx*, *login.aspx*, *Nuevo_usuario.aspx*, *ver_faltas.aspx* y *ver_notas.aspx*
- Los archivos .aspx.cs donde está el código que implementa las funcionalidades de cada página
Los ficheros son: *Añadir_alumno.aspx.cs*, *Añadir_faltas.aspx.cs*, *Añadir_mensajes.aspx.cs*, *consultar_mensajes.aspx.cs*, *CrearAula.aspx.cs*, *Default.aspx.cs*, *eliminarusuario.aspx.cs*, *insertar_notas.aspx.cs*, *login.aspx.cs*, *Nuevo_usuario.aspx.cs*, *ver_faltas.aspx.cs* y *ver_notas.aspx.cs*
- Los archivos de la carpeta imagen, en los que se encuentran los gráficos usados en la aplicación.
Los ficheros son: *fondo_formularios.jpg*, *fondo_menu.jpg*, *login.jpg*, *logoprincipal.jpg*

- Los archivos .cs que definen clases no vinculadas a ninguna página.
Los ficheros son: *Aula.cs* y *Usuario.cs*
- El archivo Web.config en el que se guarda la configuración general de la aplicación web.

1.6 Descripción del resto de capítulos

En los siguientes capítulos se explicará de forma detallada las diferentes etapas que se han seguido para desarrollar el trabajo.

Antes de detallar el desarrollo de la aplicación, se hará un breve resumen explicativo de las tecnologías usadas en él: ASP.NET y ADO.NET.

El detalle de la aplicación empezará por dar una visión general de las funcionalidades esperadas de ella, lo que nos ha de conducir a detallar de una forma más concreta sus requisitos funcionales y de usabilidad hallados durante la fase de análisis. Estos requisitos funcionales desembocarán en la redacción de los casos de uso requeridos por la aplicación.

En el siguiente capítulo se detallará el diseño de la aplicación, concretamente el diseño de la implementación de los casos de uso, el diseño de la persistencia y el diseño de la interfaz de usuario.

Seguidamente serán mostrados detalles de la codificación de la aplicación pertenecientes a la fase de implementación del trabajo.

Y finalmente se explicarán las conclusiones resultantes del trabajo, así como futuras mejoras y añadidos que serían posibles para optimizar la aplicación obtenida.

2 TECNOLOGÍAS EMPLEADAS

2.1 Microsoft.NET

Microsoft.NET es la arquitectura o conjunto de nuevas tecnologías lanzadas por Microsoft con el objetivo de mejorar tanto su sistema operativo como su modelo de componentes para obtener una plataforma con la que sea sencillo el desarrollo de software en forma de servicios web.

El acceso a estos servicios se realiza en base a estándares de Internet, como son diferentes mecanismos del protocolo HTTP (GET y PUT) o el novedoso protocolo RPC conocido como SOAP (Simple Access Object Protocol), que no es más que una combinación de estándares como HTTP y XML para realizar llamadas a los miembros de estos servicios web.

Pero la plataforma .NET no son sólo los servicios web, sino que también ofrece numerosos servicios a las aplicaciones que para ella se escriban, como por ejemplo un recolector de basura, independencia de la plataforma y total integración entre lenguajes. De hecho sería posible, por ejemplo, escribir una clase en C# que derive de otra escrita en Visual Basic.NET que a su vez derive de otra escrita en Cobol.

2.2 ASP.NET

El ASP (Active Server Pages) es el componente de Microsoft.Net totalmente orientado al desarrollo de aplicaciones Web. El funcionamiento de esta tecnología es el siguiente: cuando el navegador de un usuario realiza una petición a una página ASP, el servidor devuelve una página html generada en ese mismo momento basándose en el código ASP alojado en el servidor. Esto permite alcanzar un grado de interactividad y personalización total entre la aplicación y el usuario. El siguiente diagrama servirá de ayuda para entender mejor el funcionamiento de esta tecnología:



Algunas de las posibilidades más interesantes que nos permite el ASP son el poder acceder a bases de datos o generar gráficos y correos electrónicos de forma dinámica.

2.3 ADO.NET

La tecnología ADO.NET es la solución propuesta por Microsoft para permitir el acceso a bases de datos desde su plataforma .NET. Lo que hacen los objetos ADO (Objetos de Datos ActiveX) es permitir que una aplicación escrita en un lenguaje de programación cualquiera, pueda comunicarse con una base de datos usando el SQL (Structured Query Language). Se podría decir que ADO.NET es un conjunto de clases que exponen servicios de acceso a datos al programador de .NET.

Un aspecto importante de ADO.NET es el hecho de que utilice un modelo de acceso a datos pensado para entornos desconectados. Esto quiere decir que la aplicación se conecta al origen de datos, realiza las operaciones que necesite, carga en memoria los resultados obtenidos utilizando variables y se desconecta del origen de datos.

2.4 C#

Ya se ha visto que es posible programar en la plataforma .NET en prácticamente cualquier lenguaje, sin embargo el escogido para desarrollar este trabajo ha sido el C#.

Este lenguaje ha sido creado por Microsoft al considerar conveniente el poder disponer de un lenguaje diseñado desde cero para ser utilizado en .NET. Es un lenguaje que no cuenta con elementos heredados de versiones anteriores e innecesarios en esta plataforma y que por lo tanto es lo más sencillo posible para programarla aprovechando toda su potencia y versatilidad.

3 ANÁLISIS DE REQUISITOS

3.1 Visión general

En primer lugar hay que tener en cuenta que la aplicación va a ser administrada desde el propio centro educativo en el que sea instalada, por lo que su administración debe ser sencilla e intuitiva, y hallarse integrada dentro de la misma aplicación. Hay que tener en cuenta también que los usuarios finales son un grupo totalmente heterogéneo, por lo que la usabilidad de la interfaz debe ser lo más clara y fácil de usar que sea posible. En cuanto a las funciones que realizará la Web, serán de administración de la propia aplicación, de introducción de datos o de lectura de datos, por lo que la mayor parte de las funcionalidades tendrán que acceder a la base de datos para resolverse.

3.2 Requisitos funcionales

En la siguiente lista se dan los requisitos funcionales que ha de cumplir la aplicación agrupados según el tipo de usuario que necesitará dichas funcionalidades:

Administrador:

- *Crear un nuevo usuario:* Crea un usuario de tipo alumno o profesor.

Al crear un nuevo alumno, también se creará un nuevo usuario del tipo padre.

- *Borrar un usuario:* Elimina una cuenta de usuario de la base de datos. Podrá borrarse cualquier usuario excepto el mismo administrador.
- *Crear nueva asignatura:* Crea una nueva asignatura y le asigna un profesor.

Profesor:

- *Añadir alumnos a sus asignaturas:* Se usa a principio del curso para asignar los alumnos que tendrá el profesor en sus asignaturas.
- *Dejar mensaje en el tablón:* Dejará un mensaje en el tablón de una asignatura para que pueda ser consultado por padres, alumnos o ambos.
- *Introducir faltas de asistencia de los alumnos:* inserta las faltas de asistencia de los alumnos para que sean consultadas por los padres.
- *Insertar notas:* permite insertar las notas de sus alumnos.

Alumno:

- *Consultar el tablón para alumnos de una asignatura:* permite ver los mensajes dejados por el profesor para todos los alumnos.
- *Consultar las notas:* permite consultar las notas sacadas por el alumno en las diferentes asignaturas.
- *Consultar las faltas de asistencia:* permite consultar las faltas de asistencia del alumno en las diferentes asignaturas.

Padre:

- Consultar el tablón para padres de una asignatura: permite ver los mensajes dejados por el profesor para todos los padres.
- Consultar las notas: permite consultar las notas sacadas por el hijo en las diferentes asignaturas.
- Consultar las faltas de asistencia: permite consultar las faltas de asistencia del hijo en las diferentes asignaturas.

3.3 Requisitos de la interfaz de usuario

3.3.1 Identificación de las restricciones técnicas:

Hay que tener en cuenta que esta aplicación web ha de ser usada por todo tipo de usuarios utilizando conexiones de velocidad totalmente variable. Por ello se ha de optimizar el apartado gráfico, así como la información de ida y vuelta al servidor, para que no resulte excesivamente lenta la navegación por la aplicación.

3.3.2 Perfiles de usuario

Es necesario hacer una distinción entre los tres tipos de usuarios que usarán la aplicación:

Administrador: Sólo existe una cuenta de usuario administrador y es creada en el momento de instalación de la aplicación. Es el encargado de crear las cuentas del resto de usuarios que tengan que usar la aplicación.

Profesor: Son los usuarios encargados de gestionar la entrada de datos en la aplicación: introducir notas, introducir faltas de asistencia o dejar mensajes en los tableros de anuncios.

Alumno/Padre: Son los usuarios mayoritarios. Son usuarios básicamente pasivos, en el sentido que la mayoría de las funciones que tienen a su disposición sirven para hacer una consulta de datos introducidos por un usuario profesor.

3.3.3 Documentación de las tareas:

Tarea	Frecuencia (1-4)	Usuarios	Entradas	Salidas	Problemas posibles
login	3	Todos	Username y Contraseña	Devuelve una autenticación de usuario	La entrada no coincide con la BD
cambiar contraseña	2	Todos	contraseña antigua y contraseña nueva	---	---
crear nuevo usuario	3	administrador	Datos del nuevo usuario	---	El usuario ya había sido dado de alta.
borrar usuario	1	administrador	nombre, apellidos o DNI del usuario	---	---
crear aula	2	administrador	datos del aula	---	El aula ya existía

añadir alumno a aula	2	profesor	username del alumno y ID del aula	---	El alumno ya pertenecía al aula
añadir mensaje	3	profesor	ID del aula y texto del mensaje	---	---
añadir notas	2	profesor	notas, ID del aula y username del alumno	---	---
añadir faltas	2	profesor	fecha de la falta y username del alumno	---	---
consultar notas	4	alumno/ padre	username del usuario	lista con las notas del alumno	---
consultar faltas	4	alumno/ padre	username del usuario	lista con las faltas del alumno	---
consultar mensajes	4	alumno/ padre	ID del aula	Mensajes que corresponden a ese aula	---
logout	3	todos	---	---	---

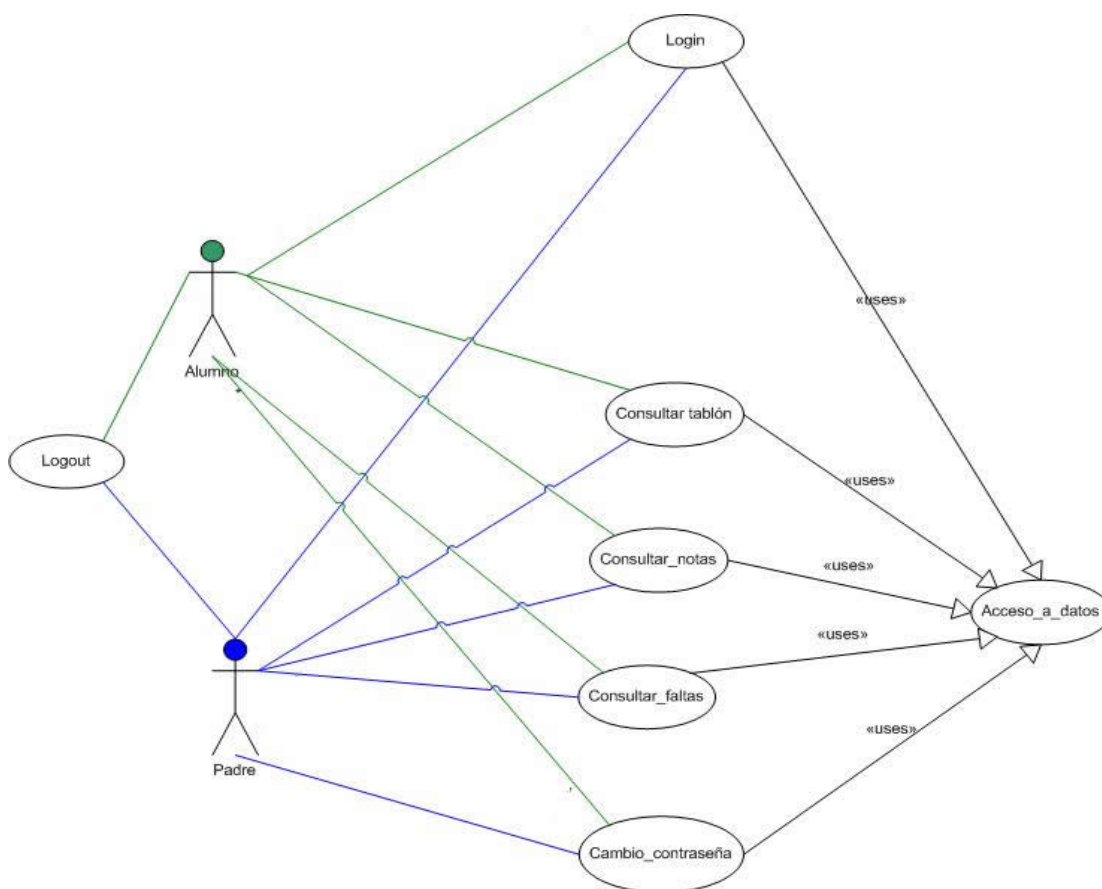
3.4 Especificaciones de usabilidad

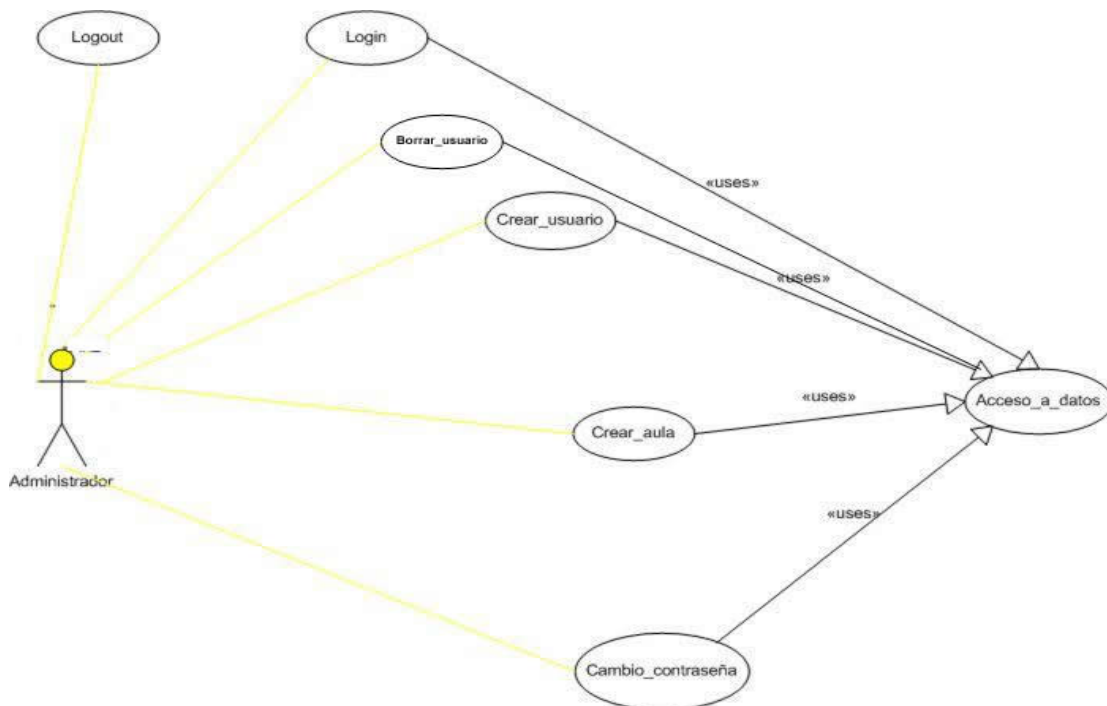
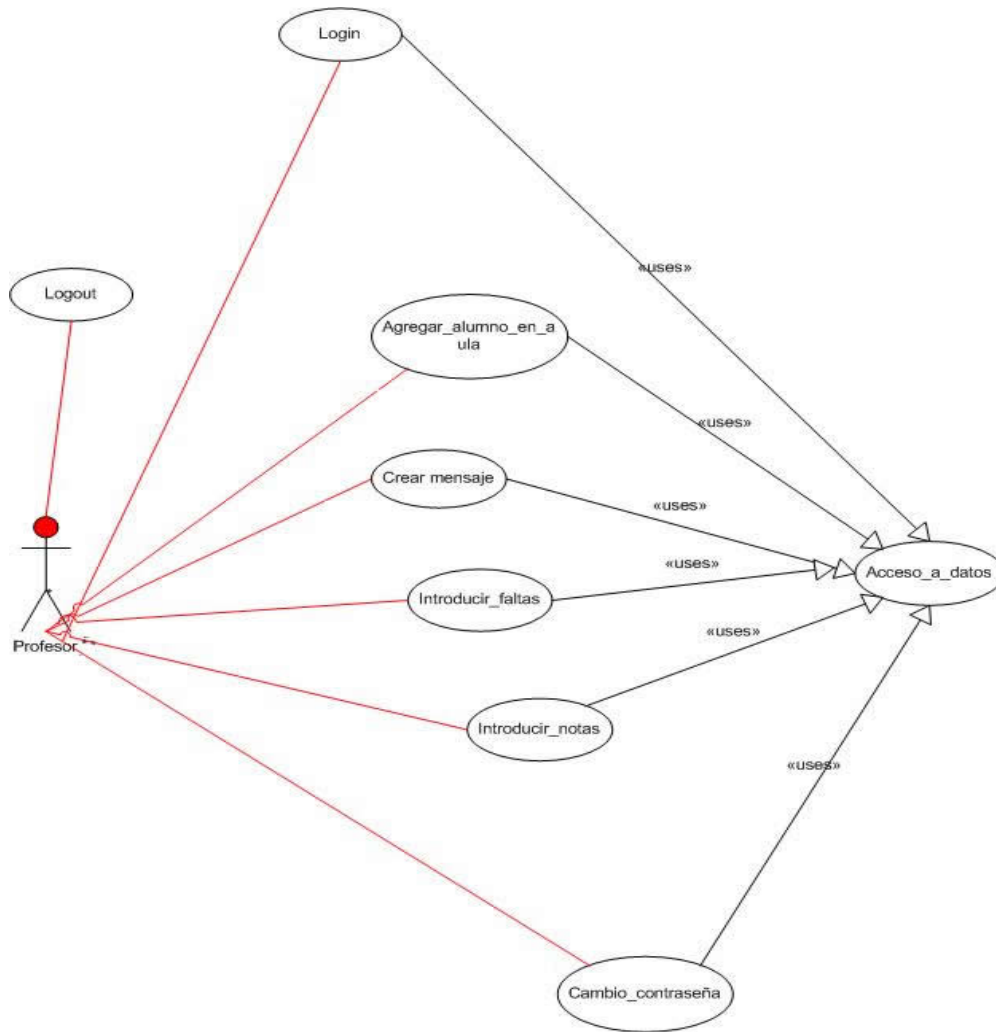
La aplicación ha de ser suficientemente intuitiva para que pueda ser utilizada por usuarios sin conocimientos informáticos elevados. Ha de ponerse especial cuidado en optimizar la usabilidad de las tareas a realizar por los usuarios del tipo Padre, que es un tipo de usuario totalmente heterogéneo, por lo que se puede dar el caso que sea un usuario sin ningún tipo de conocimiento informático.

Para conseguir esto se ha optado por utilizar una interfaz gráfica simple, con botones no demasiado pequeños y lo suficientemente descriptivos

como para que el usuario pueda saber intuitivamente para que sirven. Por otro lado, el hecho de que las funciones de estos usuarios sean exclusivamente de consulta de datos, evita que el uso incorrecto de la aplicación por su parte pueda suponer un problema grave.

3.5 Casos de uso





Caso de uso número 1: "Login"

Resumen de la funcionalidad: Identifica al usuario mediante un nombre de usuario y una contraseña.

Actores: **Administrador, alumno, profesor y padre.**

Casos de uso relacionados: Acceso a base de datos.

Precondiciones: El usuario debe ser el administrador o haber sido creado por este.

Postcondiciones: Se crea un objeto Usuario cargando los datos de la base de datos. Se autentifica al usuario.

El usuario introduce su nombre de usuario y su contraseña, y ambos son validados tras consultar la base de datos. Se carga en pantalla la página principal con el menú correspondiente según cual sea el tipo de usuario (administrador, profesor, alumno o padre).

Caso de uso número 2: "Cambiar contraseña"

Resumen de la funcionalidad: Cambia en la base de datos la contraseña de un usuario.

Actores: **Administrador, alumno, profesor y padre.**

Casos de uso relacionados: Acceso a base de datos.

Precondiciones: El usuario debe existir en la base de datos. El usuario debe facilitar la antigua contraseña que será cotejada con la que está almacenada en la base de datos.

Postcondiciones: La antigua contraseña queda sustituida por la nueva en la base de datos.

El usuario introduce la antigua y la nueva contraseñas, y si la antigua contraseña coincide con la que está almacenada en la base de datos, esta será sustituida por la nueva.

Caso de uso número 3: "Crear nuevo usuario"

Resumen de la funcionalidad: Crea un nuevo usuario y lo guarda en la base de datos.

Actores: **Administrador**.

Casos de uso relacionados: Acceso a base de datos.

Precondiciones: No debe haber ningún usuario en la base de datos con el mismo DNI.

Postcondiciones: Se habrá insertado un nuevo usuario en la base de datos.

El administrador deberá introducir los datos del nuevo usuario en un formulario que variará ligeramente según el tipo de usuario que se desee añadir (profesor, o alumno y padre). El nombre de usuario será igual al DNI del usuario, excepto en el caso de los usuarios padre cuyo

nombre de usuario será p+el DNI de su hijo. La contraseña inicial siempre será igual al segundo apellido del usuario.

Caso de uso número 4: "Borrar usuario"

Resumen de la funcionalidad: Elimina un usuario de la base de datos.

Actores: **Administrador**.

Casos de uso relacionados: Acceso a base de datos.

Precondiciones: El usuario debe existir en la base de datos. Si el usuario es de tipo profesor, aún no debe haber sido asignado a ningún aula.

Postcondiciones: El usuario habrá sido borrado de la base de datos.

El administrador deberá introducir el nombre, apellidos o DNI del usuario que desea borrar. Se hará un listado con todos los usuarios coincidentes con los criterios de la búsqueda, y el administrador seleccionará uno de ellos. Finalmente, el usuario quedará eliminado de la base de datos. Si el usuario era de tipo alumno, también será borrado el usuario padre, las asignaciones de aula y las notas que estuvieran asociadas a él.

Caso de uso número 5: "Crear Aula"

Resumen de la funcionalidad: crea un nuevo objeto aula y lo inserta en la base de datos.

Actores: **Administrador**.

Casos de uso relacionados: Acceso a base de datos.

Postcondiciones: Se habrá creado un nuevo registro en la tabla aula de la base de datos.

El administrador deberá introducir el año, el profesor que impartirá clase en este aula, y seleccionar el curso, y la asignatura que formarán el aula. El campo ID_aula se generará automáticamente.

Caso de uso número 6: "Añadir alumno a Aula"

Resumen de la funcionalidad: Asocia un determinado Alumno a un Aula.

Actores: **Profesor**.

Casos de uso relacionados: Acceso a base de datos.

Precondiciones: El profesor debe ser responsable de algún aula. El alumno aún no pertenece a este aula.

Postcondiciones: Se habrá creado un nuevo registro en la tabla Matricula de la base de datos.

El profesor elegirá una de sus aulas, y el alumno que quiera añadir. Al añadirlo, se creará un nuevo registro en la tabla Matricula de la base de datos, que constará del username del alumno y del ID_aula del aula.

Caso de uso número 7: "Añadir mensaje"

Resumen de la funcionalidad: Añade un mensaje a un tablón de anuncios.

Actores: **Profesor**.

Casos de uso relacionados: Acceso a base de datos.

Precondiciones: El profesor debe ser responsable de algún aula.

Postcondiciones: Se creará un nuevo registro en la tabla mensaje que estará asociado a una de las aulas de las que el profesor sea responsable.

El profesor selecciona una de sus aulas e indica si será un mensaje dirigido a los padres, a los alumnos, o a ambos. Introducirá el texto con el mensaje que desee, siendo añadido como un nuevo registro de la tabla mensaje en la base de datos.

Caso de uso número 8: "Añadir nota"

Resumen de la funcionalidad: Edita las notas de un determinado aula.

Actores: **Profesor**.

Casos de uso relacionados: Acceso a base de datos.

Precondiciones: El profesor debe ser responsable de algún aula y el aula seleccionada debe tener asociado como mínimo 1 alumno.

Postcondiciones: Los cambios realizados en las notas serán también realizados en la base de datos tras la validación del profesor.

El profesor seleccionará el aula a la que quiere añadir o modificarle las notas. Tras realizar los cambios deseados, estos serán actualizados en la tabla 'nota' de la base de datos.

Caso de uso número 9: "Añadir falta"

Resumen de la funcionalidad: Añade una falta de asistencia.

Actores: **Profesor**.

Casos de uso relacionados: Acceso a base de datos.

Precondiciones: El profesor debe ser responsable de algún aula y el aula seleccionada debe tener asociado como mínimo 1 alumno.

Postcondiciones: Se añadirá una nueva falta de asistencia en la base de datos.

El profesor seleccionará el día de la falta y el alumno que la cometió. Estos datos serán insertados en la tabla 'falta' de la base de datos.

Caso de uso número 10: "Consultar notas"

Resumen de la funcionalidad: Muestra en pantalla las notas de un determinado alumno.

Actores: **Alumno y Padre.**

Casos de uso relacionados: Acceso a base de datos.

Precondiciones: El alumno debe pertenecer como mínimo a un aula.

El alumno o el padre podrán consultar las notas del alumno de todas las aulas a las que pertenece.

Caso de uso número 11: "Consultar faltas"

Resumen de la funcionalidad: Muestra en pantalla las faltas de asistencia de un determinado alumno.

Actores: **Alumno y Padre.**

Casos de uso relacionados: Acceso a base de datos.

Precondiciones: El alumno debe pertenecer como mínimo a un aula.

El alumno o el padre podrán consultar las faltas de asistencia del alumno de todas las aulas a las que pertenece.

Caso de uso número 12: "Consultar tablón"

Resumen de la funcionalidad: muestra los mensajes guardados en un determinado tablón.

Actores: **Alumno y Padre.**

Casos de uso relacionados: Acceso a base de datos.

Precondiciones: El alumno debe pertenecer como mínimo a un aula.

El alumno o el padre podrán consultar los mensajes dejados por los profesores de las diferentes aulas a las que pertenezca el alumno. En este caso de uso la aplicación distinguirá entre usuario tipo padre y tipo alumno, de manera que si es alumno le mostrará los mensajes que sean para alumnos y si es padre le mostrará los mensajes que sean para padres.

Caso de uso número 13: "Acceso a base de datos"

Resumen de la funcionalidad: Accede a la base de datos de la aplicación.

Casos de uso relacionados: Todos los demás casos de uso.

Precondiciones: Debe existir la base de datos Escuela con sus correspondientes tablas y la aplicación debe tener los permisos para acceder a ella

Se accederá a la base de datos mediante ADO.NET ya sea para realizar una consulta, modificar un registro o insertar uno nuevo.

Caso de uso número 14: "Logout"

Resumen de la funcionalidad: Vuelve a la pantalla de login.

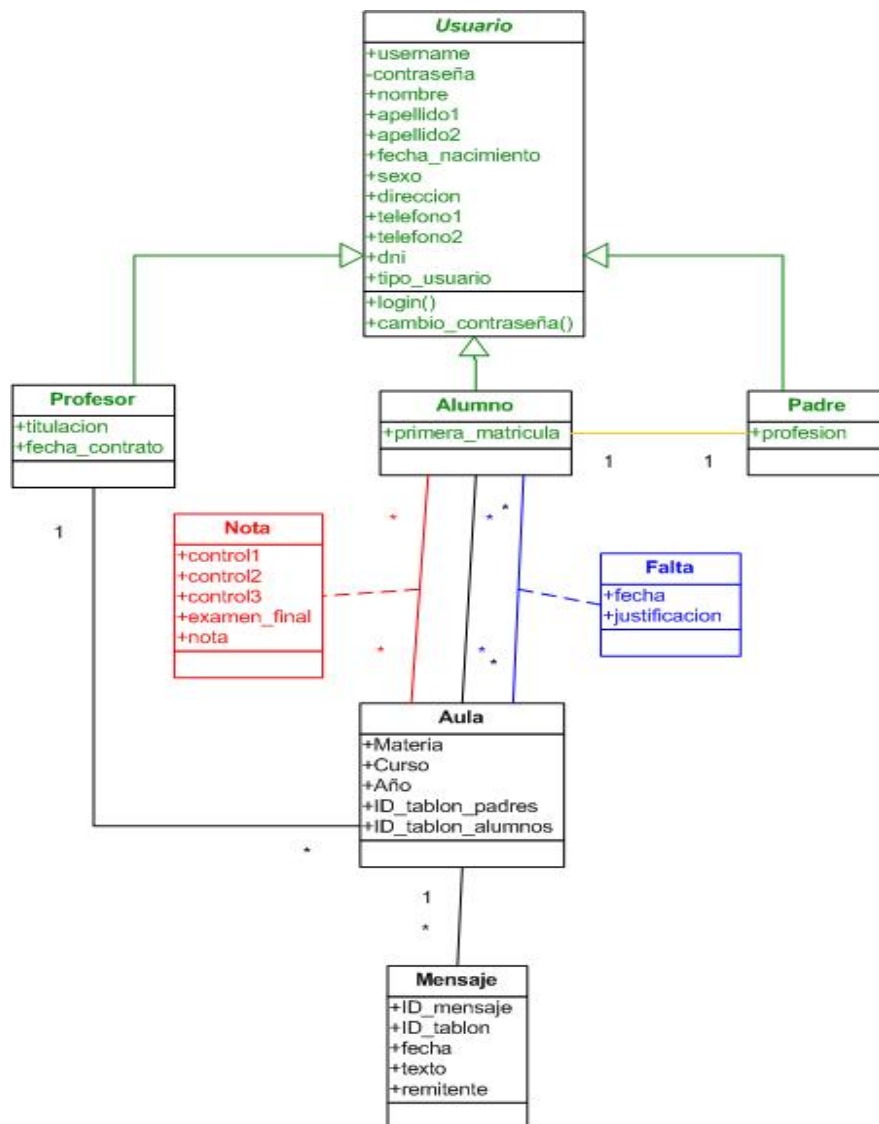
Actores: **Administrador, alumno, profesor y padre.**

Precondiciones: El usuario debe haber hecho el Login antes.

Postcondiciones: El usuario dejará de estar autenticado y regresará a la pantalla de Login.

En el caso de uso Logout, el usuario se desautentifica (se destruye la cookie) y se le reenvía a la pantalla de Login.

3.6 Diagrama de clases:



4 DISEÑO

4.1 Diseño de la implementación de los casos de uso

Caso de uso número 1: "Login"

Al cargar la página principal de la aplicación, el usuario se encontrará con una pantalla en la que tendrá que introducir su nombre de usuario y su contraseña. Al intentar autenticarse la aplicación creará una instancia de la clase usuario, buscando en la base de datos los datos del usuario cuyo username coincida con el nombre de usuario introducido. Si el nombre de usuario existe en la base de datos y la contraseña facilitada coincide con la que corresponde a ese nombre de usuario el servidor enviará la página principal del usuario. En caso contrario se le indicará que el nombre de usuario o la contraseña son incorrectos y se le permitirá que los vuelva a introducir.

Para facilitar la consulta y teniendo en cuenta que cada tipo de usuario tiene su nombre de usuario y contraseña en una tabla diferente de la base de datos, se ha optado por crear una vista que contenga únicamente los nombres de usuario y contraseñas de los diferentes tipos de usuarios insertados en la base de datos. De este modo la consulta se hace directamente sobre esta vista, y no sobre las tablas de profesor, alumno y padre, que es como se tendría que hacer de otro modo.

Para los usuarios del tipo alumno y profesor el nombre de usuario siempre será su nº de NIF, mientras que para los usuarios del tipo padre, su nombre de usuario será p + el nº de NIF de su hijo. Existe un

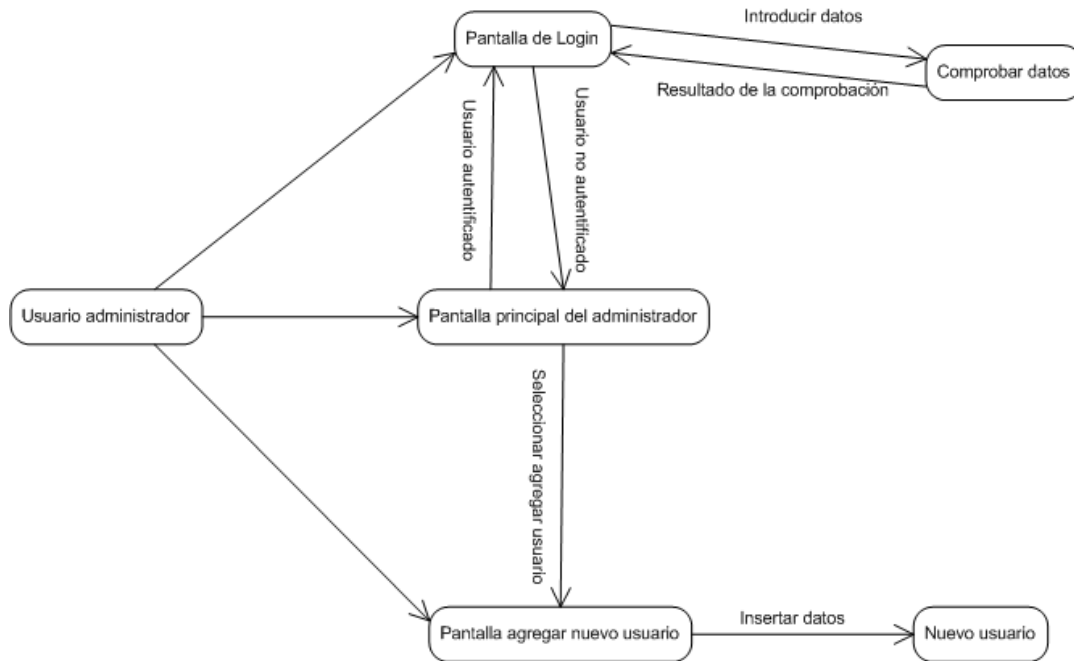
usuario especial en la aplicación que es el usuario administrador, cuyo nombre de usuario siempre será admin.

Caso de uso número 2: "Cambiar contraseña"

Todos los usuarios tendrán la posibilidad de cambiar su contraseña desde su pantalla principal de usuario. Para hacerlo tendrán que indicar cual era su antigua contraseña y cual será su nueva contraseña.

El cambio de contraseña se implementa mediante la función `cambiar_contraseña()` que pertenece a la clase usuario. Cuando el usuario valida el cambio, se realiza una consulta para comprobar que la contraseña almacenada en la base de datos y la facilitada por el usuario como antigua contraseña coincidan. Si es así, se actualiza la base de datos modificando la contraseña del usuario por la nueva que ha facilitado. Si la contraseña almacenada y la facilitada por el usuario como antigua contraseña no coinciden se da un mensaje de error por pantalla y no se realiza ninguna actualización.

Caso de uso número 3: "Crear nuevo usuario"



El usuario administrador será el encargado de crear nuevos usuarios y lo hará desde la pantalla de creación a la que accederá desde su pantalla principal de usuario. En esta pantalla seleccionará si el nuevo usuario es del tipo alumno o del tipo profesor.

Dependiendo del tipo de nuevo usuario seleccionado, se hará visible un módulo o otro, con los campos de texto para que el administrador introduzca los datos del nuevo usuario.

Si el administrador indica que el nuevo usuario es de tipo alumno, se cargarán una serie de campos de texto en los que se tendrán que ir introduciendo los datos tanto del nuevo alumno como del nuevo usuario padre, asociado a ese alumno, que se creará a la vez.

Una vez rellenos todos los campos se podrá validar la creación, de forma que el nuevo o los nuevos usuarios sean insertados en la base de datos.

Algunos de los campos de esta inserción no serán especificados por el usuario administrador sino que se cogerán por defecto. En la siguiente tabla se muestra el valor de estos campos:

Tipo de usuario:	Campo	Valor
Profesor	Nombre de usuario	su NIF
	Contraseña	su segundo apellido
Alumno	Nombre de usuario	su NIF
	Contraseña	su segundo apellido
Padre	Nombre de usuario	p + el NIF del alumno
	Contraseña	su segundo apellido
	Dirección	La misma del alumno
	Teléfonos	Los mismos que el alumno

Caso de uso número 4: "Borrar usuario"

El usuario administrador puede acceder a la pantalla de borrado de usuario desde su pantalla principal. Aquí tiene toda una lista ordenable de los usuarios tipo profesor o tipo alumno existentes en la base de datos. Puede seleccionar en todo momento uno o otro tipo, controlándose la selección gracias a las excepciones lanzadas por el `RadioButtonList` de la selección y al pulsar el botón eliminar que se encuentra al lado de cada uno de los usuarios de la lista, se comprueba cual es el username del usuario seleccionado y es eliminado de la base de datos. Al realizar una eliminación, la lista se actualiza

inmediatamente con el contenido de la base de datos, dejando de mostrar a ese usuario.

En caso de eliminar un usuario tipo alumno, en la misma función se borra de la base de datos el usuario de tipo padre asociado a ese alumno.

Caso de uso número 5: “Crear Aula”

El usuario administrador puede crear nuevas aulas. Para ello debe indicar que materia se impartirá en ese aula, a que curso pertenecerá y el año de inicio del aula. También debe escoger que profesor será el encargado de impartir las clases de ese aula.

Si todos los datos han sido correctamente introducidos y ese aula aún no existía, se insertan los datos del nuevo aula en la base de datos y el profesor que ha sido asignado como profesor del aula podrá acceder a ella desde su menú principal.

Hay que tener en cuenta que existe un tipo especial de aulas que serán las aulas de tutoría, gracias a las cuales se puede agrupar a los alumnos que pertenezcan a un mismo curso y asignarles un tutor.

Caso de uso número 6: “Añadir alumno a Aula”

Los usuarios de tipo profesor pueden acceder a la pantalla de agregar alumnos a alguna de sus aulas desde su pantalla principal, siempre y

cuando hayan sido designados como profesores de algún aula por el administrador.

Para llegar a ella tienen que seleccionar primero en cual de sus aulas desean añadir alumnos.

En esta pantalla tienen dos listados ordenables. Por un lado la lista de todos los alumnos que no pertenecen al aula, y por otro la lista de alumnos que sí que pertenecen. Desde aquí podrán añadir y quitar alumnos hasta confeccionar la lista de alumnos que quieran que pertenezcan a ese aula.

Al añadir o quitar alumnos, se hace una inserción o un borrado respectivamente, de la tabla matricula de la base de datos, usando como referencia el username que nos es devuelto por una excepción de la DataGrid al pulsar el botón. También se crea o se destruye un nuevo registro en la tabla nota, en el que se guardarán las notas de ese alumno en ese aula.

Caso de uso número 7: "Añadir mensaje"

Los usuarios de tipo profesor pueden acceder a la pantalla de añadir mensajes a alguna de sus aulas desde su pantalla principal, siempre y cuando hayan sido designados como profesores de algún aula por el administrador.

Para llegar a ella tienen que seleccionar primero en cual de sus aulas desean añadir el nuevo mensaje.

En esta pantalla deben seleccionar a quien va dirigido el mensaje (padres, alumnos o ambos), introducir un título y finalmente rellenar el campo de texto con el cuerpo del mensaje.

Al enviar el mensaje este queda insertado en la tabla mensaje de la base de datos, cogiendo un valor ID_tablon que lo relaciona con el aula seleccionada por el usuario profesor, y un valor ID_mensaje aún no existente en la base de datos. Este valor es devuelto por la función nuevo_ID() y es igual al mayor valor del campo ID_mensaje existente en la base de datos más uno. También queda registrado el día y la hora del envío.

Caso de uso número 8: “Añadir nota”

Los usuarios de tipo profesor pueden acceder a la pantalla de añadir notas a alguna de sus aulas desde su pantalla principal, siempre y cuando hayan sido designados como profesores de algún aula por el administrador.

Para llegar a ella tienen que seleccionar primero en cual de sus aulas desean modificar las notas.

En esta pantalla se le presenta una tabla en la que aparecen todos los alumnos del aula con el valor de sus notas en ese aula. El usuario profesor puede seleccionar un alumno y editar sus notas. Si en ese momento vuelve atrás o selecciona otro alumno, los cambios realizados en las notas se perderán. Para que los cambios queden guardados tiene que pulsar el botón de validar cambios, provocando que se ejecute la función update_nota() que se encarga de actualizar la tabla nota de la base de datos con los cambios realizados.

Caso de uso número 9: "Añadir falta"

Los usuarios de tipo profesor pueden acceder a la pantalla de añadir faltas de asistencia a alguna de sus aulas desde su pantalla principal, siempre y cuando hayan sido designados como profesores de algún aula por el administrador.

Para llegar a ella tienen que seleccionar primero en cual de sus aulas desean añadir las nuevas faltas.

En esta pantalla el usuario profesor tiene que seleccionar el alumno del aula al que desea añadirle una falta de asistencia, y el día en que cometió dicha falta. Al añadir la falta, se hace una nueva inserción en la tabla falta de la base de datos, guardando el username del alumno, la fecha de la falta y el aula al que corresponde la falta.

Caso de uso número 10: "Consultar notas"

Los usuarios tipo alumno o tipo padre pueden consultar las notas. Los usuarios tipo alumno consultan sus propias notas, mientras que los usuarios tipo padre consultan las notas del alumno que tengan asociado.

Al entrar en la pantalla consultar notas podrán ver una tabla en la que tendrán listadas las asignaturas a las que pertenece el alumno junto con las notas que tiene en cada una de ellas.

Caso de uso número 11: "Consultar faltas"

Los usuarios tipo alumno o tipo padre pueden consultar las faltas de asistencia. Los usuarios tipo alumno, pueden consultar sus propias faltas, mientras que los usuarios tipo padre consultan las faltas del alumno que tengan asociado.

Al entrar en la pantalla consultar faltas podrán ver una lista con las faltas de asistencia del alumno. En cada entrada de la lista se mostrará el día de la falta y el nombre de la asignatura de la que se ausentó. También se verá la justificación de la falta.

Caso de uso número 12: "Consultar tablón"

Los usuarios tipo alumno o tipo padre pueden consultar los mensajes dejados para ellos por los profesores de las aulas a las que pertenezcan o, en el caso de los usuarios tipo padre, de las aulas a las que pertenezca el alumno que tengan asociado.

Al entrar en la pantalla consultar tablón el usuario debe elegir de cual de sus aulas desea consultar los mensajes. Al seleccionar el aula aparecen todos los títulos de los mensajes de ese aula por orden de inserción. Finalmente, al seleccionar el título del mensaje que se desea consultar, el cuerpo del mensaje aparece en el campo de texto para poder ser leído.

Caso de uso número 13: "Acceso a base de datos"

El acceso a la base de datos Escuela, ya sea para insertar, actualizar o borrar datos, se realiza siempre de manera similar. Se prepara el comando sql que se desea realizar en un objeto del tipo SqlCommand, se abre la conexión con la base de datos, se ejecuta el comando guardando el resultado del mismo en un objeto del tipo DataSet y finalmente se vuelve a cerrar la conexión con la base de datos.

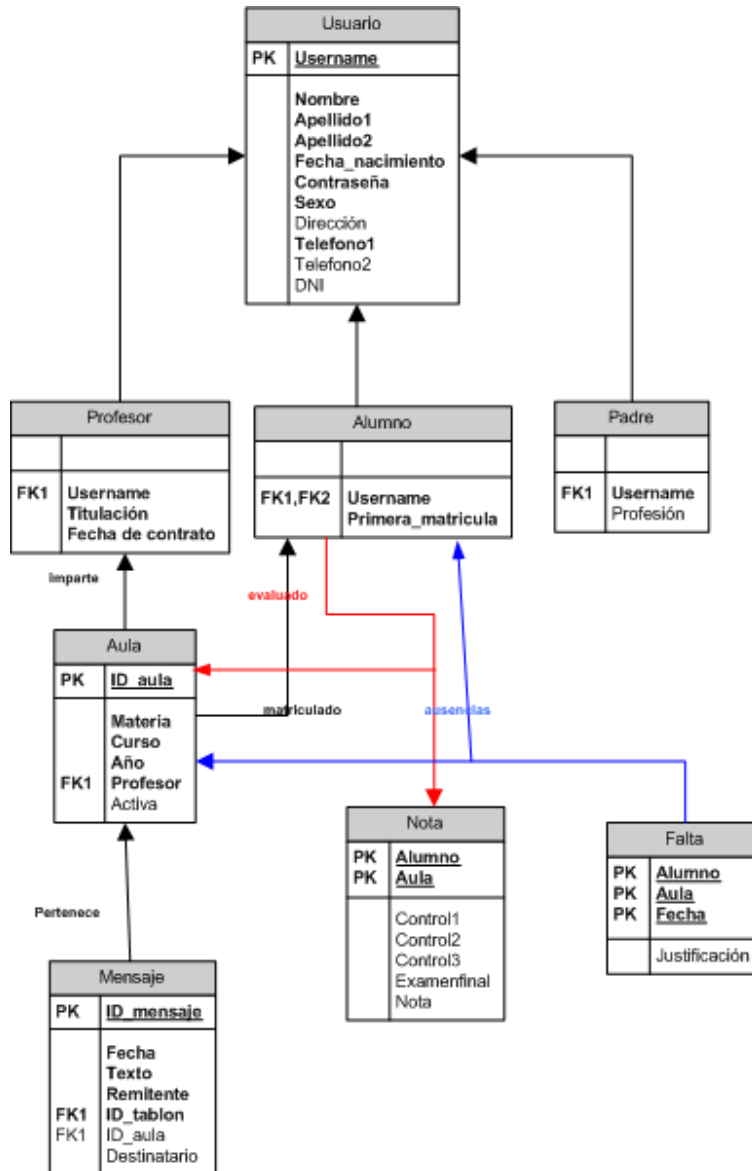
Caso de uso número 14: "Logout"

Todos los usuarios pueden cerrar su sesión en el momento que lo deseen. Al hacerlo serán desautenticados, es decir, su cookie de usuario será eliminada, la variable de sesión en la que se guarda a qué usuario pertenece esa sesión será borrada y el servidor enviará la página de inicio al navegador.

Este proceso de logout será realizado automáticamente por el servidor pasados 60 minutos desde la autenticación.

4.2 Diseño de la persistencia

Diagrama de la base de datos



Modelo Relacional de la Base de datos

PROFESOR(username, nombre, apellido1, apellido2, fecha_nacimiento, password, sexo, direccion, telefono1, telefono2, titulacion, fecha_de_contrato)

ALUMNO(username, nombre, apellido1, apellido2, fecha_nacimiento, password, sexo, direccion, telefono1, telefono2, primera_matricula)

PADRE(username, nombre, apellido1, apellido2, fecha_nacimiento, password, sexo, direccion, telefono1, telefono2, profesion, username_alumno)

Donde {username_alumno} referencia ALUMNO

AULA(materia, curso, año, username_profesor)

Donde {username_profesor} referencia PROFESOR

NOTA(username_alumno, ID_aula, control1, control2, control3, examen_final, nota)

Donde {username_alumno} referencia ALUMNO y {ID_aula} referencia AULA

FALTA(ID_aula, username_alumno, fecha, justificación)

Donde {username_alumno} referencia ALUMNO y {ID_aula} referencia AULA

MATRICULA(username_alumno, ID_aula)

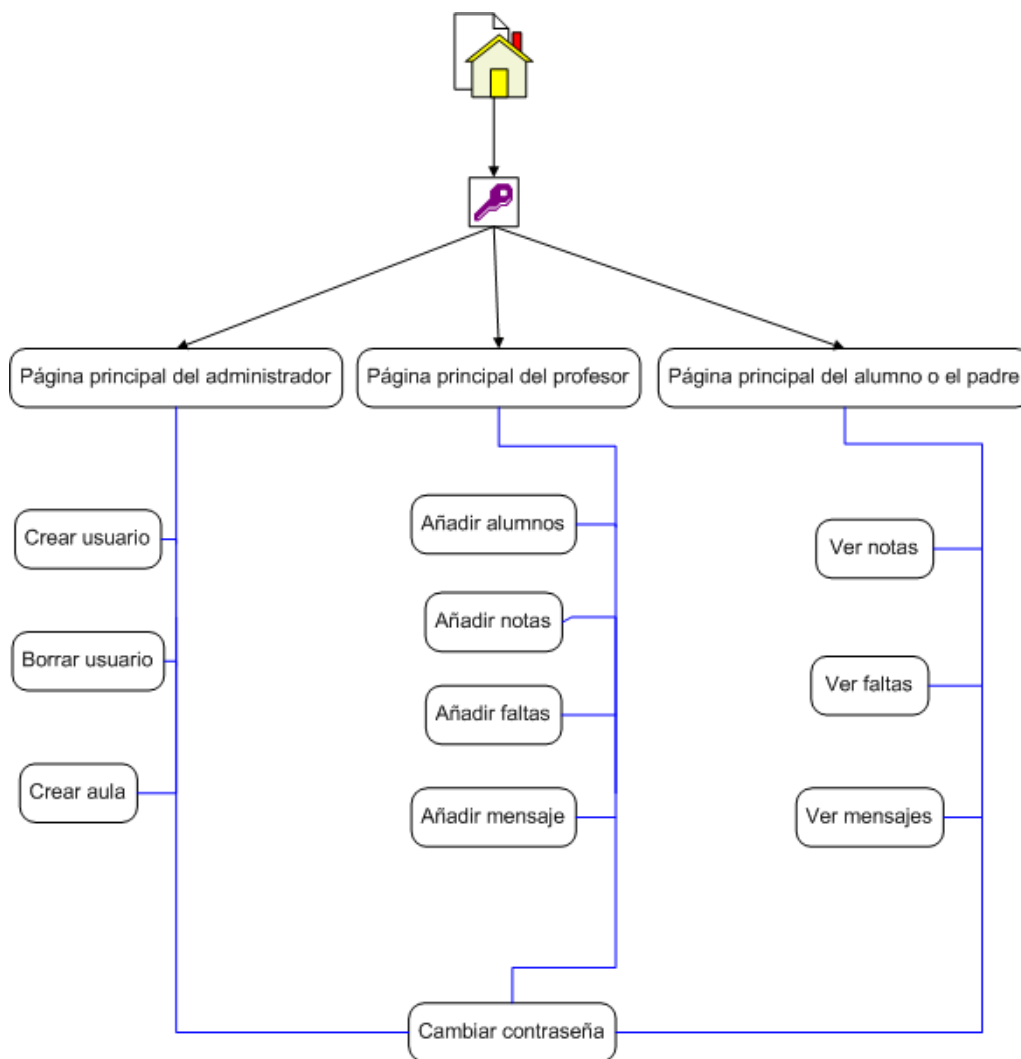
Donde {username_alumno} referencia ALUMNO y {ID_aula} referencia AULA

MENSAJE(ID_mensaje, ID_tablon, fecha, texto, remitente, titulo, destinatario)

Donde {remitente} referencia PROFESOR y {ID_tablon} referencia AULA

4.3 Diseño de la interfaz

En el siguiente diagrama se muestra el mapa web que conforma la aplicación desarrollada:



A continuación se pasará a presentar de una manera detallada el diseño de la interfaz para las diferentes pantallas de la aplicación agrupadas según los usuarios que las utilizarán.

Pantallas comunes a todos los usuarios

Pantalla de Login:



Esta es la página inicial de la aplicación. Cualquier acceso a la aplicación por parte de un usuario no autenticado es reenviado a esta página para que se autentique. La pantalla ha de constar de dos campos de texto y un botón de validación. En el primer campo de texto el usuario introduce su nombre de usuario y en el segundo su contraseña, que no es visualizada, sino que sus caracteres aparecen como * por seguridad. Finalmente se valida la autenticación mediante el botón entrar.

Pantalla de cambio de contraseña:

ESCUELA
Versión 0.1

Inicio Inicio Cambiar contraseña Bienvenido, Omar

AÑADIR ALUMNO
NUEVO MENSAJE
INSERTAR NOTAS
INSERTAR FALTAS

Contraseña antigua : *****
Contraseña nueva: *****

CAMBIAR CONTRASEÑA
CANCELAR

Esta pantalla ha de servir para que los usuarios realicen el cambio de su contraseña. Es un panel integrado dentro de la página principal de cada usuario y al activarse y hacerse visible se deshabilitan el resto de botones de la pantalla. Consta de dos campos de texto, en los que los caracteres aparecen ocultos por seguridad, y dos botones. En el primer campo de texto el usuario deberá introducir su contraseña y en el segundo la nueva contraseña que tendrá a partir de ese momento. El primer botón validará el cambio de contraseña, mientras que el segundo servirá para cancelar todo el proceso de cambio de contraseñas.

Pantallas propias del administrador

Pantalla principal del administrador:



Esta es la primera pantalla que se ha de encontrar el administrador tras autenticarse. Consta de tres botones principales situados en la parte central izquierda de la pantalla y dos secundarios colocados un poco más arriba. Los botones principales permiten ir a las pantallas de creación de nuevos usuarios, a la de borrados de usuarios y a la de creación de nuevas aulas. Los dos botone secundarios han de servir para activar el panel de cambio de contraseña o para cerrar la sesión desautenticando al usuario.

Pantalla de creación de usuarios:

ESCUELA
Versión 0.1

VOLVER A INICIO CERRAR SESIÓN

Selecciona el tipo de usuario que vas a crear: **Nuevo Profesor**

Re llena los campos siguientes:

DATOS DEL PROFESOR

Nombre:

Primer Apellido:

Segundo Apellido:

Teléfono 1:

Teléfono 2:

Fecha de nacimiento:

DNI:

Gracias a esta pantalla el administrador ha de ser capaz de introducir en la base de datos a todo el resto de usuarios que hayan de utilizar la aplicación. Consta de dos botones secundarios que permiten volver a la página principal o cerrar la sesión, desautenticando de esta manera al usuario. Bajo estos botones se encuentra un desplegable de texto en el que el administrador seleccionará si el nuevo usuario ha de ser de tipo profesor o de tipo alumno. En el primer caso se despliega un panel con los campos de texto necesarios para que el administrador introduzca los datos del nuevo profesor. En el segundo caso se despliega un panel similar pero más extenso, en el que el administrador ha de introducir los datos del nuevo alumno y del nuevo padre asociado a ese alumno.

Pantalla de borrado de usuarios:



En esta pantalla el administrador ha de ser capaz de eliminar a los usuarios que desee de la base de datos. Para ello se despliega una tabla con los usuarios de tipo profesor o de tipo alumno en la que existe un botón integrado para eliminar al usuario de una fila concreta. Por defecto se muestra la lista de profesores, pero se puede elegir entre una u otra gracias a un selector situado a la derecha de la tabla. También consta de dos botones secundarios que permiten volver a la página principal o cerrar la sesión, desautenticando de esta manera al usuario.

Pantalla de creación de aulas:

ESCUELA
Versión 0.1

VOLVER A INICIO CERRAR SESIÓN

Selecciona Asignatura
Selecciona Curso
Inserta el año de inicio del curso:
CREAR

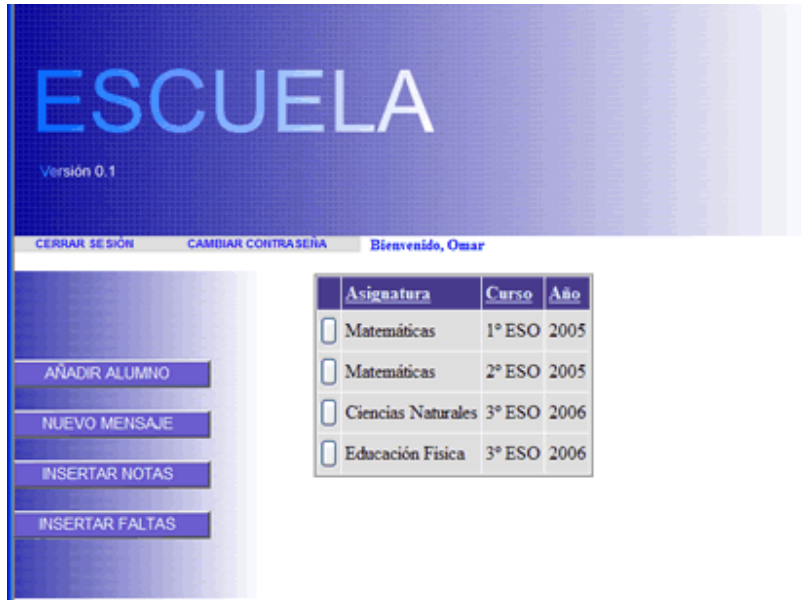
Nombre	Apellido	Segundo apellido	DNI	
<input checked="" type="checkbox"/>	Mónica	Paeo	Vispe	18043485p
<input type="checkbox"/>	Encarnación	Gómez	Rodríguez	38501311r
<input type="checkbox"/>	Omar	Benjamea	Gómez	47788420d

Materia	Curso	Profesor	Año
Matemáticas	1º ESO	Omar	2005
Matemáticas	2º ESO	Omar	2005
Filosofía	2º Bachillerato	Mónica	2006
Filosofía	1º Bachillerato	Mónica	2006
Filosofía	4º ESO	Mónica	2006
Filosofía	3º ESO	Mónica	2006
Tutoría	2º Bachillerato	Mónica	2006
Ética	1º ESO	Mónica	2006
Ciencias Sociales	2º ESO	Mónica	2006
Ciencias Naturales	3º ESO	Omar	2006
Educación Física	3º ESO	Omar	2006

Mediante esta pantalla el administrador será el encargado de crear las aulas existentes en la aplicación. La pantalla consta de dos desplegados en los que se ha de seleccionar la asignatura y el curso al que pertenecerá el aula. También existe un campo de texto en el que se ha de introducir el año en el que se iniciará ese aula. Finalmente, el administrador ha de encontrar una tabla con los profesores que haya en la base de datos para poder seleccionar el que será asignado a ese aula. Una vez seleccionados e introducidos todos los datos el nuevo aula es creada mediante el botón crear y será añadida en una tabla en la que se muestran todas las aulas creadas hasta el momento. La pantalla también consta de dos botones secundarios que permiten volver a la página principal o cerrar la sesión, desautenticando de esta manera al usuario.

Pantallas propias del profesor

Pantalla principal del profesor:



Esta es la primera pantalla que se han de encontrar los profesores tras autenticarse. Consta de cuatro botones principales situados en la parte central izquierda de la pantalla y dos secundarios colocados un poco más arriba. También ha de mostrarse una tabla con las diferentes aulas a las que está asignado el profesor. Los botones principales permiten agregar nuevos alumnos a un aula, crear nuevos mensajes para el tablón de un aula y insertar notas y faltas a los alumnos pertenecientes a una de sus aulas. Para seleccionar en cada momento sobre cual de sus aula quiere trabajar utilizará la tabla central, en la que seleccionará el aula deseada. Los dos botone secundarios han de servir para activar el panel de cambio de contraseña o para cerrar la sesión desautenticando al usuario.

Pantalla para añadir o quitar alumnos a un aula:

ESCUELA
Versión 0.1

VOLVER A INICIO CERRAR SESIÓN

Alumnos del Centro:

	Nombre	Apellido	Segundo apellido	DNI
<input type="button" value="Añadir"/>	Bombur	Benjamea	Gómez	12345678a
<input type="button" value="Añadir"/>	josaa	gomo	bOmbur	11111111c
<input type="button" value="Añadir"/>	Moniketa	Pueo	Vispe	11111111m
<input type="button" value="Añadir"/>	Omar	Benjamea	Gómez	11111111a
<input type="button" value="Añadir"/>	Pepe	Chaves	Chaves	87654321a

Alumnos asignados al Aula:

	Nombre	Apellido	Segundo apellido	DNI
<input type="button" value="Quitar"/>	Gisela	Benjamea	Gómez	99999999a
<input type="button" value="Quitar"/>	José Luis	Pueo	Vispe	18043484f

En esta pantalla el profesor ha de poder añadir nuevos alumnos a una de sus aulas o sacar de ella a alumnos ya incluidos anteriormente. La pantalla consta de dos tablas. En la primera se muestran todos los alumnos de la base de datos que no pertenecen al aula, y consta de un botón integrado en cada fila al lado del nombre de los alumnos para añadir a ese alumno al aula. En la segunda se muestran todos los alumnos que pertenecen al aula, y también consta de un botón integrado en cada fila, en este caso para excluir al alumno del aula. La pantalla también consta de dos botones secundarios que permiten volver a la página principal o cerrar la sesión, desautenticando de esta manera al usuario.

Pantalla para crear nuevos mensajes:

ESCUELA
Versión 0.1

VOLVER A INICIO CERRAR SESIÓN Mensaje insertado con éxito

Selección Destinatarios:

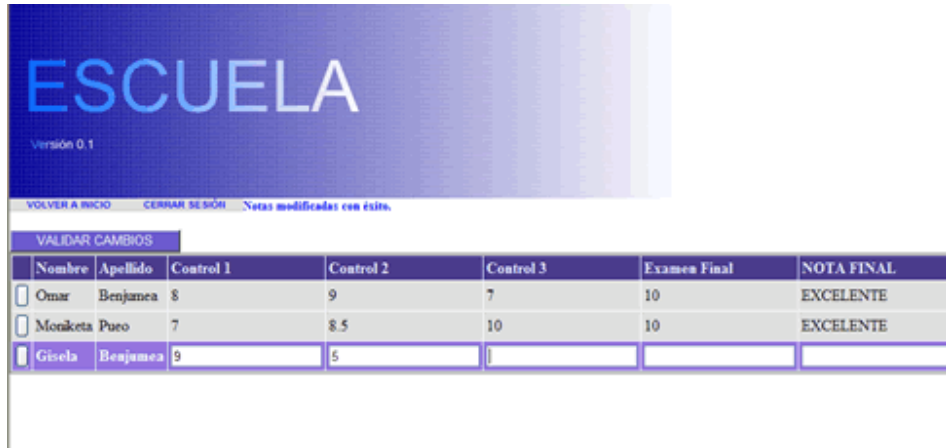
Alumnos
 Padres
 Ambos

Titulo: Mensaje de prueba ENVIAR MENSAJE

Texto: Este es un mensaje para mostrar la interfaz del envío y lectura de mensajes.

A través de esta pantalla el profesor ha de ser capaz de dejar mensajes que puedan ser leídos por los alumnos o padres de alumnos pertenecientes a una de sus aulas. Consta de un selector a la izquierda con el que el profesor ha de seleccionar al destinatario del mensaje. En la parte central de la pantalla se encuentran dos campos de texto, uno para el título y otro para el cuerpo del mensaje, y un botón para validar el mensaje una vez que haya sido redactado. La pantalla también consta de dos botones secundarios que permiten volver a la página principal o cerrar la sesión, desautenticando de esta manera al usuario.

Pantalla para insertar o modificar las notas:



En esta pantalla el profesor puede insertar las notas de los alumnos que pertenecen a una de sus aulas. Consta de una tabla en la que se muestran los alumnos que pertenecen al aula y sus notas. Para modificarlas ha de seleccionar el alumno a modificar, y cambiar los valores directamente en los campos de la tabla. Finalmente validará los cambios realizados mediante un botón situado en la parte superior de la tabla. La pantalla también consta de dos botones secundarios que permiten volver a la página principal o cerrar la sesión, desautenticando de esta manera al usuario.

Pantalla para insertar las faltas de asistencia:



Mediante esta pantalla el profesor inserta las faltas de asistencia de sus alumnos. Para ello la pantalla consta de un calendario y una tabla con la lista de los alumnos que pertenecen a su aula. El profesor ha de seleccionar el día de la falta desde el calendario y el alumno que faltó desde la tabla y validar la inserción de la falta gracias al botón desplegado a tal efecto sobre la tabla. La pantalla también consta de dos botones secundarios que permiten volver a la página principal o cerrar la sesión, desautenticando de esta manera al usuario.

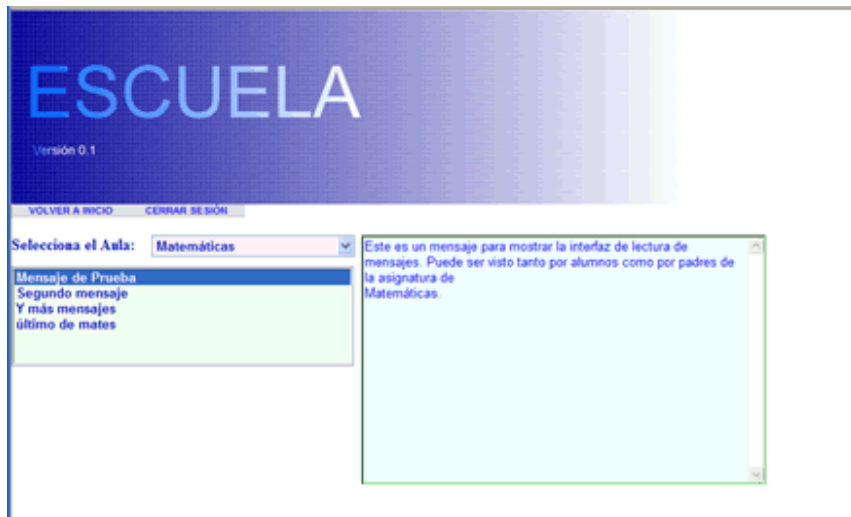
Pantallas propias del padre o alumno

Pantalla principal del padre o alumno:



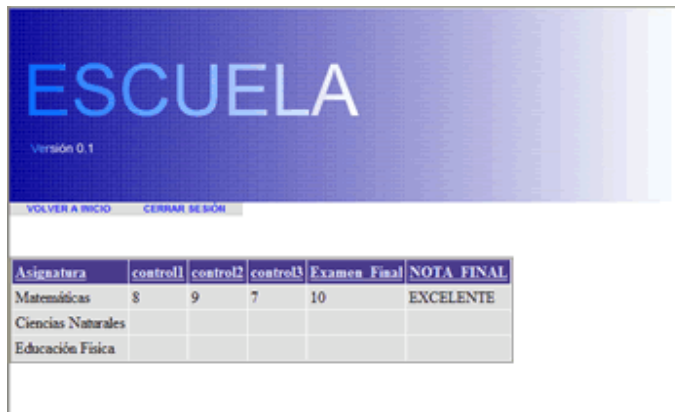
Esta es la primera pantalla que se han de encontrar los alumnos y sus padres tras autenticarse. Consta de tres botones principales situados en la parte central izquierda de la pantalla y dos secundarios colocados un poco más arriba. Los botones principales permiten acceder a las pantallas de consulta de mensajes, de consulta de notas y de consulta de faltas de asistencia. Los dos botone secundarios han de servir para activar el panel de cambio de contraseña o para cerrar la sesión desautenticando al usuario.

Pantalla para consultar los mensajes:



Mediante esta pantalla el usuario de tipo alumno o de tipo padre ha de poder consultar los mensajes dejados por sus profesores. Consta de un campo de texto desplegable mediante el cual se ha de seleccionar de cual de las aulas a las que pertenece el alumno desea ver los mensajes. En el campo de texto situado justo debajo se encuentran la lista de títulos de los mensajes que han sido dejados por el profesor. Al seleccionar cualquiera de ellos se despliega en el campo de texto más grande situado en el centro de la pantalla el cuerpo del mensaje para que pueda ser leído. La pantalla también consta de dos botones secundarios que permiten volver a la página principal o cerrar la sesión, desautenticando de esta manera al usuario.

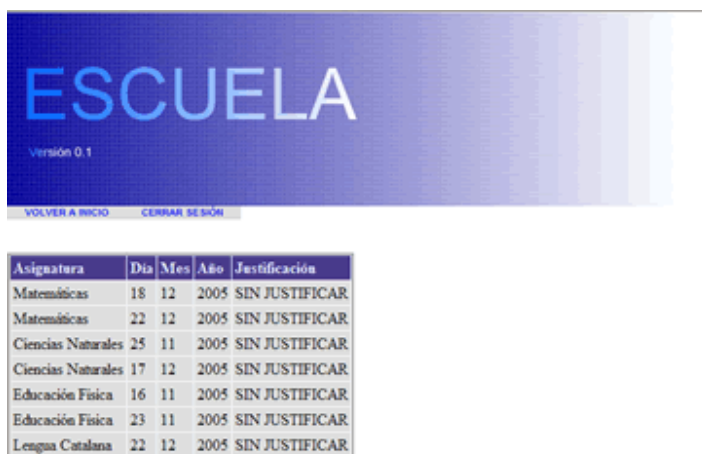
Pantalla para visualizar las notas:



Asignatura	control1	control2	control3	Examen Final	NOTA FINAL
Matemáticas	8	9	7	10	EXCELENTE
Ciencias Naturales					
Educación Física					

En esta pantalla se muestra una tabla listando todas las asignaturas a las que pertenece el alumno y las notas que existen de cada una de ellas. La pantalla también consta de dos botones secundarios que permiten volver a la página principal o cerrar la sesión, desautenticando de esta manera al usuario.

Pantalla para ver las faltas de asistencia:



Asignatura	Día	Mes	Año	Justificación
Matemáticas	18	12	2005	SIN JUSTIFICAR
Matemáticas	22	12	2005	SIN JUSTIFICAR
Ciencias Naturales	25	11	2005	SIN JUSTIFICAR
Ciencias Naturales	17	12	2005	SIN JUSTIFICAR
Educación Física	16	11	2005	SIN JUSTIFICAR
Educación Física	23	11	2005	SIN JUSTIFICAR
Lengua Catalana	22	12	2005	SIN JUSTIFICAR

En esta pantalla se pueden consultar las faltas de asistencia cometidas por el alumno. Para ello se muestra una tabla listando todas las faltas de asistencia que existen de ese alumno. La pantalla también consta de dos

botones secundarios que permiten volver a la página principal o cerrar la sesión, desautenticando de esta manera al usuario.

5 IMPLEMENTACIÓN

En esta fase del trabajo se ha obtenido el código de la aplicación a partir de la documentación obtenida durante las fases de análisis y diseño. La aplicación Web se ha realizado separando los Web Forms, guardados en archivos .aspx, de los archivos y las clases funcionales guardados en archivos .cs.

Quizá uno de los aspectos más interesantes de los utilizados en el desarrollo de la aplicación es el acceso a datos mediante objetos ADO.

A continuación se muestra, a modo de ejemplo, parte del código usado en la aplicación para este propósito:

```
//Preparamos la BD
private void preparar_BD()
{
    this.sqldataadapter = new System.Data.SqlClient.SqlDataAdapter();
    this.sqlCommand1 = new System.Data.SqlClient.SqlCommand();
    this.sqlconnection = new System.Data.SqlClient.SqlConnection();
    this.sqlCommand1.Connection = this.sqlconnection;
    this.sqldataadapter.SelectCommand=this.sqlCommand1;
    this.sqlconnection.ConnectionString = "workstation id=\"127.0.0.1\";packet
size=4096;integrated security=SSPI;initial catalog=Escuela;persist security info=False";
}
//Enlaza el datagrid Grid_nota con los alumnos que pertenecen al aula.
private void cargar_nota()
{
    preparar_BD();
    ds= new DataSet();
    this.sqlCommand1.CommandText = "SELECT a.nombre AS Nombre, a.apellido1 AS
Apellido,a.username AS DNI,n.control1,n.control2,n.control3,n.examen_final AS
Examen_Final,n.nota AS Nota_Final FROM nota n,alumno a WHERE (a.username=n.username_alumno
and n.ID_aula=@ID_aula)";
    sqlCommand1.Parameters.Add(new SqlParameter("@ID_aula", SqlDbType.Int));
    sqlCommand1.Parameters["@ID_aula"].Value = aula.ID_aula;
    this.sqldataadapter.Fill(ds);
    this.Grid_notas.DataSource=ds;
    this.Grid_notas.DataBind();
}
```

Se puede observar como en primer lugar en la función llamada preparar_BD() se preparan los objetos de conexión con la base de datos, mientras que en la función cargar_nota() se le pasa el comando

sql al objeto de tipo SqlCommand llamado sqlCommand1, que a su vez forma parte de un objeto SqlDataAdapter.

Finalmente, en el ejemplo, se asocian los datos obtenidos en el DataSet con el objeto DataGridView llamado Grid_notas para que la lista de notas sean mostradas por pantalla a través de dicho objeto.

Otro aspecto a destacar de la implementación es el uso que se ha echo de dos variables de sesión, una llamada usuario y otra llamada aula. La primera se usa en todas las sesiones. Se crea en el momento de la autenticación del usuario y guarda un objeto de tipo Usuario. Posteriormente es utilizada en todas las páginas para comprobar que el usuario realmente puede acceder a esa página. La variable llamada aula, que guarda un objeto de tipo Aula, almacena el aula de trabajo de los usuarios profesor. A continuación se muestra el código en el que se crea la variable usuario:

```
public void Login_Click(Object sender, EventArgs E)
{
    //Realizamos la conexión con la BD y obtenemos en cont el string con la contraseña que se
    //corresponde
    //al username insertado por el usuario.
    FormsAuthentication.Initialize();
    string contraseña;
    user=new Usuario(Username.Text);
    DataSet ds=user.datos;
    if (user.tipo_usuario!=-1) {contraseña=(string)ds.Tables[0].Rows[0]["contraseña"];}
    else contraseña="";
    //Comparamos la contraseña almacenada en la base de datos con la obtenida en
    //Contraseña.Text
    if (contraseña.Trim()==Contraseña.Text)
    {
        Session["usuario"]=user;
        FormsAuthentication.SetAuthCookie(Username.Text,false);
        Server.Transfer(FormsAuthentication.GetRedirectUrl(Username.Text,false));
    }
    else
    {
        usuario_incorrecto.Text = "El nombre de usuario o la contraseña no son
correctos";
    }
}
```

6 CONCLUSIONES

6.1 Conclusión y objetivos alcanzados

El resultado final del trabajo ha sido realmente muy enriquecedor ya que ha permitido tener una primera aproximación a la tecnología .NET de Microsoft, adquiriendo suficientes conocimientos como para desarrollar una aplicación Web de complejidad media. También ha servido como repaso y puesta en práctica de numerosos conceptos alcanzados en diversas asignaturas a lo largo de la carrera, especialmente de las asignaturas de programación, bases de datos y ingeniería del software. Se puede decir que se han alcanzado todos los objetivos propuestos inicialmente, tanto por el desarrollo de la aplicación como por los conocimientos adquiridos. Por otro lado, la propia redacción de la memoria es un ejercicio recomendable para alcanzar una base sólida en redacción de documentación técnica.

6.2 Recomendaciones

Se ha conseguido una aplicación Web que implementa todas las funcionalidades que se marcaron como objetivo durante la planificación del trabajo, utilizando para ello las tecnologías previstas. Sin embargo, existe toda una serie de mejoras o añadidos que se podrían realizar para mejorar el producto final.

Por ejemplo, sería interesante que se pudieran desactivar las aulas, de forma que sólo estuvieran activas las del año actual, ya que en la implementación actual de la aplicación no se hace distinción entre las asignaturas que se cursan en la actualidad o las de años pasados.

Por otro lado, sería interesante cambiar el modelo de la persistencia para que permitiera a los padres con más de un hijo matriculado en el centro tener la información de todos integrada dentro de un único

usuario, mientras que actualmente necesitaría un usuario diferente para cada hijo.

En lo que a la seguridad de la aplicación se refiere podría ser mejorable. Aunque se requiere la autenticación del usuario, y según su perfil sólo se le permite acceder a las páginas propias de ese perfil, el proceso de autenticación en el que el usuario envía su nombre de usuario y contraseña se realiza sin ningún tipo de encriptación, por lo que cualquier observador malintencionado podría ser capaz de capturar estos datos. Una manera de solucionar esto, sería implementar este proceso de autenticación usando SSL para el intercambio de información.

Otra mejora importante en el aplicativo sería el integrar el manejo de las cuentas de correo del propio centro, en caso de tener servidor de correo, dentro del aplicativo, de manera que los usuarios dispusieran de todos los servicios online dentro de la propia aplicación.

Antes de presentar comercialmente la Web como una aplicación finalizada sería conveniente añadir toda una serie de funcionalidades básicas que no han sido implementadas en esta primera versión por falta de tiempo. A continuación presentaremos algunos de estos añadidos:

- El administrador debería poder consultar las fichas de los usuarios, en las que se mostrase la información de los mismos.
- El administrador debería poder dejar avisos o alertas de mantenimiento que pudieran ser leídas por el resto de los usuarios.
- El profesor debería poder dejar mensajes personales para un determinado alumno o padre de alumno, no estar limitado a mensajes generales para todos los usuarios de un aula.
- El profesor debería poder consultar los datos personales de sus alumnos o de los padres.

- Se debería implementar un control de horarios de las asignaturas, de manera que los profesores pudieran consultar a que horas y qué días tienen que dar clases, y los alumnos consultar su horario escolar.
- Los alumnos y padres deberían poder consultar algunos datos sobre los profesores, como su especialidad, titulación, etc..
- Los padres deberían poder justificar las faltas de los alumnos directamente desde la aplicación.

7 GLOSARIO

ASP: (Active Server Pages) Componente de Microsoft totalmente orientado al desarrollo de aplicaciones Web

ADO: La tecnología ADO es la solución propuesta por Microsoft para permitir el acceso a bases de datos.

C#: Lenguaje de programación diseñado para ser utilizado en .NET.

.NET: Arquitectura de Microsoft para mejorar su sistema operativo para obtener una plataforma que facilite el desarrollo de servicios web.

SSL: (Secure Socket Layer) Es un protocolo desarrollado para dar seguridad a la transmisión de datos en transacciones comerciales en Internet. Utilizando la criptografía de llave pública, SSL provee autenticación del servidor, encriptación e integridad de los datos en las comunicaciones.

Web Forms: Es el marco de trabajo de las páginas ASP, que permite crear páginas Web programables como parte de una aplicación Web global.

8 BIBLIOGRAFÍA

8.1 Libros utilizados

Programming Microsoft Windows with C#. Charles Petzold. Microsoft Press 2002.

Professional ADO.NET Programming. Varios autores. Wrox Press Ltd 2001.

8.2 Fuentes de Internet

QuickStart Tutorials. <http://samples.gotdotnet.com/quickstart/>