

Universitat Oberta de Catalunya

**Ingeniería Técnica en Informática de
Sistemas**

Servidor de Gestión y Control de Cursos
de una Escuela de Formación

Rafael Díaz Heredia
Junio del 2006

Dirigido por Jordi Ceballos Villach

Resumen

El objetivo principal de este proyecto era consolidar los conocimientos adquiridos a lo largo de la carrera y a la vez conseguir un cierto dominio en el entorno de programación *Microsoft Visual Studio 2005*© en el ámbito *.NET*.

El proyecto desarrolla una pequeña aplicación para una escuela de formación en la que se tratará de gestionar las asistencias a los cursos. Dado el alto grado de absentismo, será importante un control paralelo de las recuperaciones de los temas no vistos. Será una aplicación que separará por un lado las operaciones a nivel de usuario normal y por otra las de un usuario especial llamado coordinador que podrá administrar el estado de los cursos y usuarios.

Se ha optado por realizar la aplicación en *ASP NET 2.0* debido a que es más sencillo actualizar y modificarla en caso de necesidad y debido también a que todos los usuarios dispondrán de una conexión a red y de un navegador actualizado, como el *Microsoft Internet Explorer 6.0*.

Se ha realizado una separación de la Vista y el Controlador, realizando la primera en código XHTML y CSS y la segunda aplicando la tecnología *.NET* y bajo lenguaje *C#*, separando las tareas propias de un programador de las de un diseñador gráfico.

Para la base de datos, se ha optado por el *Microsoft SQL Server 2005*© que parece el idóneo para una aplicación de estas características.

Decir que todo el proyecto está realizado con programación orientada a objetos y por tanto todo el código está encapsulado en clases, con todas las ventajas que esto implica.

Tabla de Contenidos

Resumen	2
1. Justificación del proyecto: punto de partida y aportación	6
1.1. Descripción de la escuela de formación	6
1.2. Requisitos	8
1.3. Objetivos	9
1.4. Enfoque y método	10
1.5. Planificación del proyecto	10
1.6. Productos obtenidos	12
1.7. Descripción de los capítulos siguientes	12
2. Infraestructura de la aplicación	13
2.1. Arquitectura de la aplicación	13
2.2. Diseño lógico de la base de datos	13
2.3. Descripción de las tablas	15
2.3.1. Tabla Usuarios	15
2.3.2. Tabla Centrales	15
2.3.3. Tabla Turnos	15
2.3.4. Tabla Alumnos	16
2.3.5. Tabla Años	16
2.3.6. Tabla Contadores	16
2.3.7. Tabla Sesiones	17
2.3.8. Tabla Temas	18
2.3.9. Tabla TemasRealizados	18
2.3.10. Tabla Convocatorias	19
2.3.11. Tabla TemasConvocados	19
2.4. Diseño de clases	20
2.4.1. Clases del Modelo	20
2.4.2. Diagrama de clases de las clases de entidades del problema	21
2.4.3. Diagrama de clases de las clases gestoras	22
2.4.4. Clases Controlador y Vista	23
2.5. Seguridad	23
2.5.1. Autenticación	24
2.5.2. Inyección SQL	24
2.5.3. Perfiles de usuario	25
2.5.4. Mejoras en la seguridad	26
2.6. Requisitos técnicos	27
2.7. Componentes externos	27
2.7.1. MessageBox	27
2.7.2. PopUp Calendar	28
2.7.3. SetFocus	29
2.7.4. msgBox: MessageBox & ConfirmationBox	29
2.7.5. Enhanced List Box Control	30
2.7.6. Librería Interop de Microsoft Office 2003	31
3. Descripción de la aplicación “Gestión de Cursos”	32

3.1. Pantalla Inicio	32
3.2. Pantalla Principal	33
3.3. Pantalla de cambio de contraseña	34
3.4. Pantalla Imparticiones	35
3.5. Pantalla Pendientes	39
3.6. Pantalla Convocatorias	40
3.7. Pantalla Informe	40
3.8. Pantalla Sesiones	43
3.9. Pantalla Alumnos	44
3.10. Pantalla Usuarios	45
4. Líneas de desarrollo futuro y posibles mejoras	47
5. Conclusiones	48
6. Bibliografía y referencias de consulta	49

Índice de Figuras

Figura 1: Ejemplo de planificación de sesiones de Ascó en el año 2006	7
Figura 2: Diagrama de Gantt del proyecto	11
Figura 3: Diagrama lógico de la base de datos	14
Figura 4: Diagrama de clases de las entidades del problema	21
Figura 5: Diagrama de clases de las entidades gestoras	22
Figura 6: Ejemplo del control MessageBox	28
Figura 7: Ejemplo de uso del control PopUp Calendar	28
Figura 8: Aspecto del control PopUp Calendar	29
Figura 9: Aspecto del control msgBox en su funcionalidad de ConfirmationBox	29
Figura 10: Ejemplo de uso del control Enhanced List Box Control	31
Figura 11: Pantalla Inicio	33
Figura 12: Pantalla principal con barra de pestañas de coordinador	34
Figura 13: Detalle de la barra de pestañas para un instructor no coordinador	34
Figura 14: Pantalla de cambio de contraseña	35
Figura 15: Pantalla Imparticiones	35
Figura 16: Introducción de la asistencia a una sesión en fecha programada	36
Figura 17: Introducción de la asistencia de una sesión en fecha no programada	37
Figura 18: Hoja oficial de firmas de asistencia	38
Figura 19: Temas lectivos recuperables pendientes de un alumno en la pantalla Pendientes	39
Figura 20: Detalle de la pantalla Pendientes donde se realiza una convocatoria de recuperación	40
Figura 21: Detalle de la pantalla Convocatorias donde se observa una convocatoria de recuperación de un alumno	40
Figura 22: Pantalla Informe	41
Figura 23: Detalle de los totalizadores de horas de un turno en el informe anual	41
Figura 24: Detalle de una sesión realizada por un turno en el informe anual	42
Figura 25: Pantalla Sesiones: Cuadro resumen de sesiones	43
Figura 26: Introducción de una nueva sesión en la pantalla Sesiones	44
Figura 27: Gestión de alumnos en la pantalla Alumnos	45
Figura 28: Pantalla Usuarios	46

1. Justificación del proyecto: punto de partida y aportación

La idea de la realización de este proyecto salió en la empresa en la cual trabajo. Hablando con algunos compañeros surgió la idea de realizar una aplicación que nos permitiese automatizar de una manera sencilla el control y la gestión de los cursos, ya que, hasta ahora, este control se realiza de una manera totalmente manual, en un fichero *MSEXcel*, siempre laborioso y propenso a tener errores.

La idea sería poder tener un servidor que se dedicara a contener la aplicación y la base de datos y que desde cualquier ordenador de la empresa un usuario pudiera conectarse a él para introducir la asistencia de los alumnos a sus clases, ver el estado de las ausencias y programar convocatorias para recuperar los temas no vistos.

La aportación principal de este proyecto será precisamente su producto, ya que espero poder utilizarlo en mi trabajo diario ahorrándome tiempo y esfuerzo en una tarea que, a mi modo de ver, no debería hacernos gastar tantos recursos.

A nivel personal, este proyecto también sirve para hacer que conozca el lenguaje de programación *C#* y la tecnología *.NET* y sus posibilidades, totalmente desconocidos por mí antes de comenzar.

1.1. Descripción de la escuela de formación

Dada la particularidad de la Escuela de Formación, se ha de describir el funcionamiento de ésta para entender mejor el objetivo y la funcionalidad del aplicativo.

La Escuela de Formación está ubicada en Tarragona. Entre otras muchas cosas, su función más importante es dar la formación obligatoria al personal de Sala de Control de las centrales nucleares de Ascó y Vandellós II. Esta formación obligatoria, según la Guía 1.1 del Consejo de Seguridad Nuclear (CSN) debe ser de 120 horas mínimas, de las cuales, como mínimo 20 deben ser prácticas en un simulador de alcance total^[28].

El personal de Sala de Control de las centrales está asignado a un turno. De momento, cada central tiene siete turnos, aunque es posible que en un futuro se aumente este número a ocho.

Cada turno está compuesto por un jefe de turno, un jefe de sala, un operador de reactor y un operador de turbina por cada grupo nuclear que tenga la central. Así, Vandellós II, como solo tiene un grupo nuclear (un único reactor) su turno lo componen cuatro personas. En cambio, Ascó tiene dos grupos nucleares, por lo que cada turno lo componen ocho personas: dos jefes de turno, aunque uno de ellos de

menor categoría jerárquica y se le llama ayudante del jefe de turno, dos jefes de sala, dos operadores de reactor y dos de turbina.

Según el cuadrante, un turno puede trabajar de mañana, de tarde, de noche, estar en retén, estar en formación obligatoria o estar de fiesta, en ciclos repetitivos de siete semanas. Estar de retén implica ir a trabajar en caso de tener que sustituir a alguna persona del turno que esté en ese momento en la Sala de Control por baja u otros motivos, ya que en Sala de Control debe haber en todo momento cuatro personas. Si no se tiene que sustituir a nadie, entonces va a formación con su grupo.

De modo que a formación viene personal que en su cuadrante está de formación obligatoria o de retén. Cuando están de formación obligatoria es difícil que falte alguien, pero cuando están de retén, hay muchas ausencias, ya que es probable que personal del turno de mañana, tarde o noche falte a trabajar.

Se programan al principio del año una serie de sesiones de formación, habitualmente cinco o seis, aunque podrían ser más. Las sesiones se dividen en jornadas de seis horas de duración cada una. Lo normal es que cada sesión contenga cinco jornadas, aunque podrían ser menos.

Como son ciclos de siete semanas, para que a una sesión de formación acuda todo el personal, la sesión se ha de repetir siete veces, una para cada turno. Es posible que alguna sesión coincida en el tiempo, ya que se puede estar dando formación al turno de formación obligatoria y en otra aula, al de retén. Como ejemplo se adjunta la planificación del año 2006 para la central nuclear de Ascó, en la que se observan las 6 sesiones, de siete semanas cada una.

2006 rev 0	PRIMER SEMESTRE 2006														SEGUNDO SEMESTRE 2006																																									
	ENERO			FEBRERO				MARZO				ABRIL			MAYO				JUNIO			JULIO				AGOSTO				SEPTIEMBRE			OCTUBRE			NOVIEMBRE				DICIEMBRE																
	9	16	23	30	6	13	20	27	4	11	18	25	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	4	11	18	25	1								
F.O.																																																								
S.1-Simulador (F.O.)																																																								
S.2 Simulador (F.O.) los 2 Gr																																																								
S.3- Lectiva PCD/lect (R.)																																																								
S.4-Lect/PEI-PR (R.)																																																								
S4.1. AASS																																																								
S.5-CI (R.)																																																								
RECUPERACIONES(R.)																																																								
Reten As																																																								

Figura 1: Ejemplo de planificación de sesiones de Ascó en el año 2006

Como se programan del orden de 130 horas al año, es imprescindible llevar el control de las ausencias y realizar recuperaciones de las jornadas perdidas, ya que hay que asegurar al finalizar el año, que todos y cada uno han recibido las 120 horas mínimas obligatorias.

De cara a una auditoria del CSN, lo único oficial es un formato en el que para cada sesión, el jefe de turno certifica con su firma los asistentes a las jornadas de la sesión y los temas recibidos por cada componente de su turno.

1.2. Requisitos

La aplicación deberá llevar el control de temas y horas de formación que cada alumno recibe y llevar un control sobre las recuperaciones de los temas perdidos de cada alumno, ya que al final de año se intenta que todos los alumnos hayan visto todos los temas programados.

Habrán dos módulos, uno de usuario normal o instructor y otro de administrador o coordinador. Además, en función del perfil del usuario registrado, sea coordinador o instructor, verá solo los datos de su central asociada, sea Ascó o Vandellós. Se incorpora un módulo para que un coordinador pueda dar de alta a otra central, cosa improbable por ahora, pero se deja la puerta abierta a una ampliación del programa. Además, se entiende que un mismo instructor no puede estar asociado a más de una central. Si lo estuviese se le darían dos perfiles de entrada.

Así, cuando entre un usuario se habrá de registrar para ver que central tiene asignada y si es coordinador o no.

Un instructor normal podrá:

- Rellenar los datos para una sesión programada y dada de alta en el sistema. Estos datos serán: la fecha de cada jornada, los temas que se imparten en cada jornada, el turno y la asistencia de cada día. Además, y dada la posibilidad de que una persona repita el mismo tema dos veces (solo para temas prácticos), deberá dejar constancia de esto. Una vez ha asignado la asistencia de toda la sesión, podrá imprimir la hoja de firmas para que los alumnos confirmen su asistencia a las clases.
- Recuperar una persona o varias con temas pendientes de alguna sesión. Para ello introducirá el día de recuperación y los temas recuperados. Después podrá generar la hoja individual de asistencia.
- Ver los temas pendientes y recuperables que le queda a cada persona.
- Programar una convocatoria a una persona que le falten horas recuperables. Esto es solo para contabilidad de las estadísticas finales, pues es interesante ver el número de convocatorias hechas a un alumno para recuperar un tema (o lo que es lo mismo, el número de veces que no ha asistido a clase).
- Ver el estado actual de la formación por grupo y por persona, las horas realizadas y las horas totales programadas.
- Además, podrá cambiar su contraseña de entrada en la aplicación.

Un coordinador podrá realizar las mismas funciones que el instructor, por tanto será una especialización de éste, y además podrá:

- Dar de alta a nuevos instructores o coordinadores.
- Dar de alta a una nueva sesión. Para ello introducirá el número de temas y la descripción de estos, las horas asignadas a cada tema, si son horas teóricas o prácticas y si estas horas son recuperables o no. Además, deberá introducir la fecha de la semana prevista para cada turno.

- Modificar una sesión. Cambiar alguno de los parámetros asignados a una sesión, siempre y cuando no haya habido ninguna impartición de ésta.
- Gestión de alumnos. Podrá asignar a cada alumno a un turno, dar de alta o de baja a nuevos o antiguos alumnos, cambiarles de turno o incluso asignarles un puesto dentro del turno.
- Podrá dar de alta a una nueva central en caso de que hubiese necesidad.

La aplicación estará contenida en un servidor, que es donde estará la base de datos de alumnos y cursos y utilizará la tecnología *ASP.NET*. Por tanto, será sencillo el mantenerla y actualizarla, ya que no se necesita instalar nada en los ordenadores de los instructores o de las aulas, simplemente necesitarán un navegador actualizado.

Al ser una aplicación de código abierto hay que tener en cuenta algunos aspectos de la programación:

- El código ha de estar bien documentado para facilitar posibles ampliaciones.
- El código ha de ser muy modular y fácil de modificar y ampliar.
- En la medida de lo posible se ha de fomentar la reutilización de las clases más genéricas.
- La aplicación ha de ser lo más genérica posible.
- Si bien es importante la fiabilidad de la aplicación, se adoptará un criterio más relajado en favor de la sencillez y claridad del código.

El lenguaje de programación, si bien puede ser cualquiera compatible con la tecnología .NET, será *C#*.

Los datos estarán guardados en una base de datos SQL a la que se podrá acceder mediante *SQL Server 2005*, instalado en el servidor.

1.3. Objetivos

Los objetivos de este proyecto son dos:

- Conocer y familiarizarme con la tecnología .NET. Siempre me ha llamado la atención programar, aunque nunca he tenido la oportunidad de hacerlo. Además, utilizar la tecnología .NET es para mí un atractivo, ya que he leído mucho de ella y creo que los resultados pueden ser muy satisfactorios.
- Crear una herramienta que, como instructor, me simplifique la faena en mi puesto de trabajo, ya que se pierden más horas en el control del personal y de los cursos que en la preparación de estos.
- Aplicar los conocimientos aprendidos durante el estudio de la carrera de *ETIS*.

1.4. Enfoque y método

El enfoque y el método se basan en el estándar del desarrollo de aplicaciones, es decir, en las fases de análisis, diseño, implementación y prueba.

La ventaja de este proyecto es que los requisitos de la aplicación son dados por el propio desarrollador, que además, va a ser usuario, con lo que el producto resultante estará muy cerca del deseado por la mayoría de usuarios sin necesidad de reuniones de definición ni alcance.

Se pretende conseguir una aplicación suficientemente intuitiva para todos los usuarios, que sea autoexplicativa y no necesite de ayudas. A pesar de todo, se realizará un manual de funcionamiento para asegurarnos el perfecto uso de la aplicación.

1.5. Planificación del proyecto

La planificación viene determinada por el calendario de PACS que hay que ir entregando a lo largo del trimestre y que marcarán las fechas de los hitos más importantes.

- 1. Entrega del plan de trabajo (13 de Marzo):** Este documento presenta la descripción general del proyecto así como su planificación. Para realizarlo es necesario instalar y aprender lo básico del MS Project 2003, necesario para realizar el diagrama de Gannt.
- 2. Entrega del análisis y primer prototipo (3 de Abril):** En este documento se especifican las funcionalidades de la aplicación mediante casos de uso y se diseña una primera versión de la interfície.
- 3. Entrega del diseño (18 de Abril):** Este documento definirá como se implementan las funcionalidades descritas en la anterior fase. Se adjuntará el diagrama de clases y el diseño de la base de datos.
- 4. Entrega de la implementación (22 de Mayo):** Se entregará la aplicación, y, aunque no será la definitiva, si tendrá que ser muy próxima a ésta.
- 5. Entrega de la memoria y la presentación (19 de Junio):** Se entregará el resultado final de la aplicación, como el documento final donde se recogerá todo el trabajo y una pequeña presentación para mostrar a los evaluadores todo el proceso. Después de esta entrega, habrá un periodo de dos semanas en la que el tribunal evaluará el proyecto y podrá remitir preguntas al alumno.

Una de las tareas que directamente no llevan a un hito pero que es importante resaltar es el aprendizaje y comprensión del lenguaje de programación *C#* y de *ASP.NET*, que será de unos dos meses desde el inicio del proyecto y que debe finalizar antes de la entrega de la implementación.

El diagrama de Gannt obtenido es el siguiente:

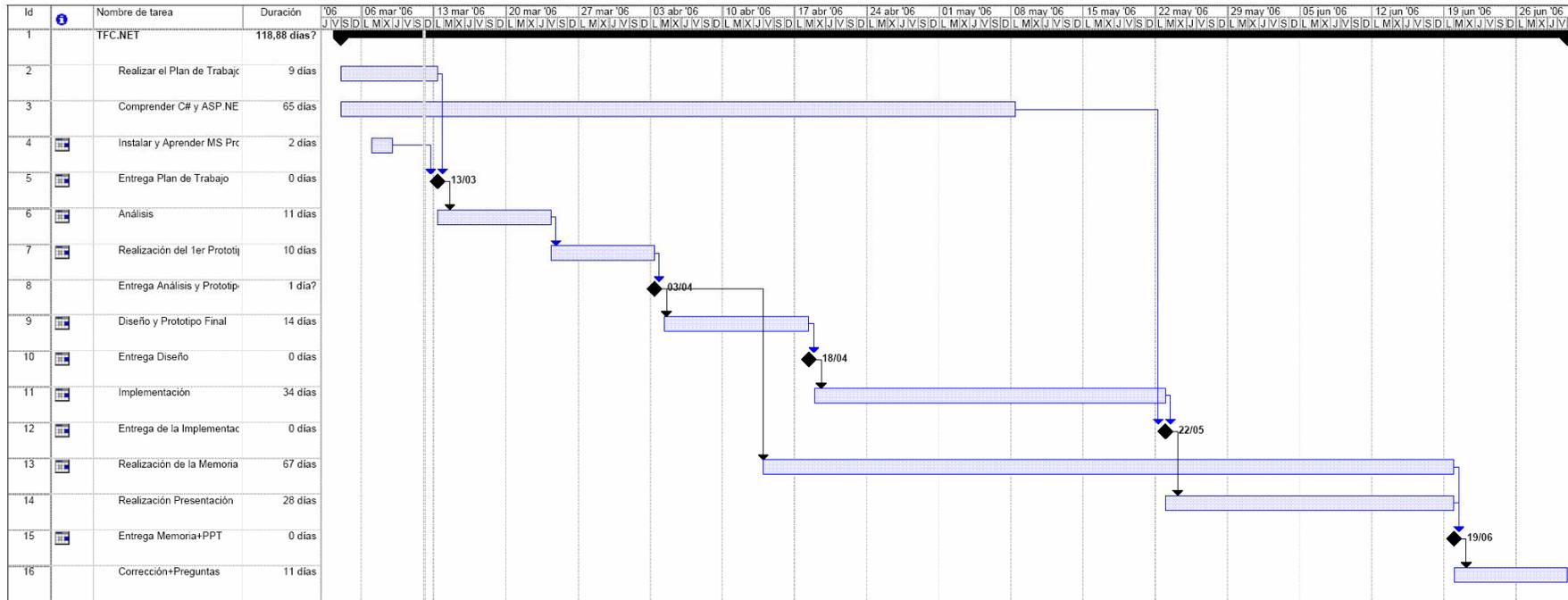


Figura 2: Diagrama de Gannt del proyecto

1.6. Productos obtenidos

Durante el desarrollo del proyecto se han obtenido los siguientes productos ya descritos en el apartado anterior de planificación:

- Plan de trabajo: Breve descripción del proyecto y planificación temporal
- Análisis: Conjunto de requisitos y funcionalidades representados mediante casos de uso
- Prototipo en HTML: diseño de la interficie y ejemplos de las funcionalidades en HTML estático
- Diseño: diagrama de clases y diseño de la base de datos
- Implementación: aplicación totalmente operativa
- Manual de instalación: breve documento que recoge los requisitos y guía de aplicaciones y componentes a instalar
- Memoria del proyecto: Documento final que resume todo el trabajo
- Presentación del proyecto: Documento que recoge los aspectos más importantes para resaltar en una presentación y defensa del proyecto

1.7. Descripción de los capítulos siguientes

En el siguiente capítulo se hablará de la arquitectura de la aplicación, que criterios se han seguido y los motivos. Se hablará del diseño de clases, sus relaciones, datos y servicios.

Se dará el modelo entidad relación y se detallarán todas y cada una de las tablas con sus columnas, claves primarias y foráneas.

Se hablará de la seguridad de la aplicación, así como de posibles mejoras de ésta de cara a una futura ampliación de la aplicación.

En el tercer capítulo se podrá ver la estructura de las páginas así como un detalle de cada una para hacerse una idea muy aproximada de la aplicación.

En el cuarto capítulo se hablará de las líneas abiertas y posibles mejoras y ampliaciones que se podrían realizar en un futuro.

En el quinto capítulo se expondrán las conclusiones.

Y finalmente se recogerá toda la bibliografía y direcciones de Internet usadas para realizar este proyecto.

2. Infraestructura de la aplicación

Este capítulo describe la arquitectura utilizada en el proyecto así como el detalle de las clases y tablas que contiene.

2.1. Arquitectura de la aplicación

Para desarrollar este proyecto se utilizará la arquitectura modelo-vista-controlador (MVC).

El objetivo es intentar separar lo máximo posible la interfície, la lógica y los datos para dar una mayor facilidad de mantenimiento y para reutilizar el código allí donde se pueda en futuras aplicaciones.

Con este patrón conseguimos separación de responsabilidades. Se trata de dividir la aplicación en la vista (el interface), el controlador (lo que maneja la vista), y el modelo (lo que utiliza el controlador para generar nuevas vistas).

En nuestro caso, al utilizar el Framework .NET, el controlador, cada vista tendrá un controlador propio (Page Controller) gracias al code-behind, que son los que gestionan las acciones de los usuarios en esa página.

- Vista: Las páginas .aspx que el servidor web manda a los usuarios y que captura las acciones de éstos, pasándola al controlador de esa página.
- Controlador: Las clases aspx.cs que interaccionan con el modelo de datos y con la vista.
- Modelo: Las clases que representan el modelo de datos y la lógica de la aplicación.

2.2. Diseño lógico de la base de datos

El diagrama lógico de la base de datos se obtiene directamente del modelo relacional que se presenta en el documento de Diseño.

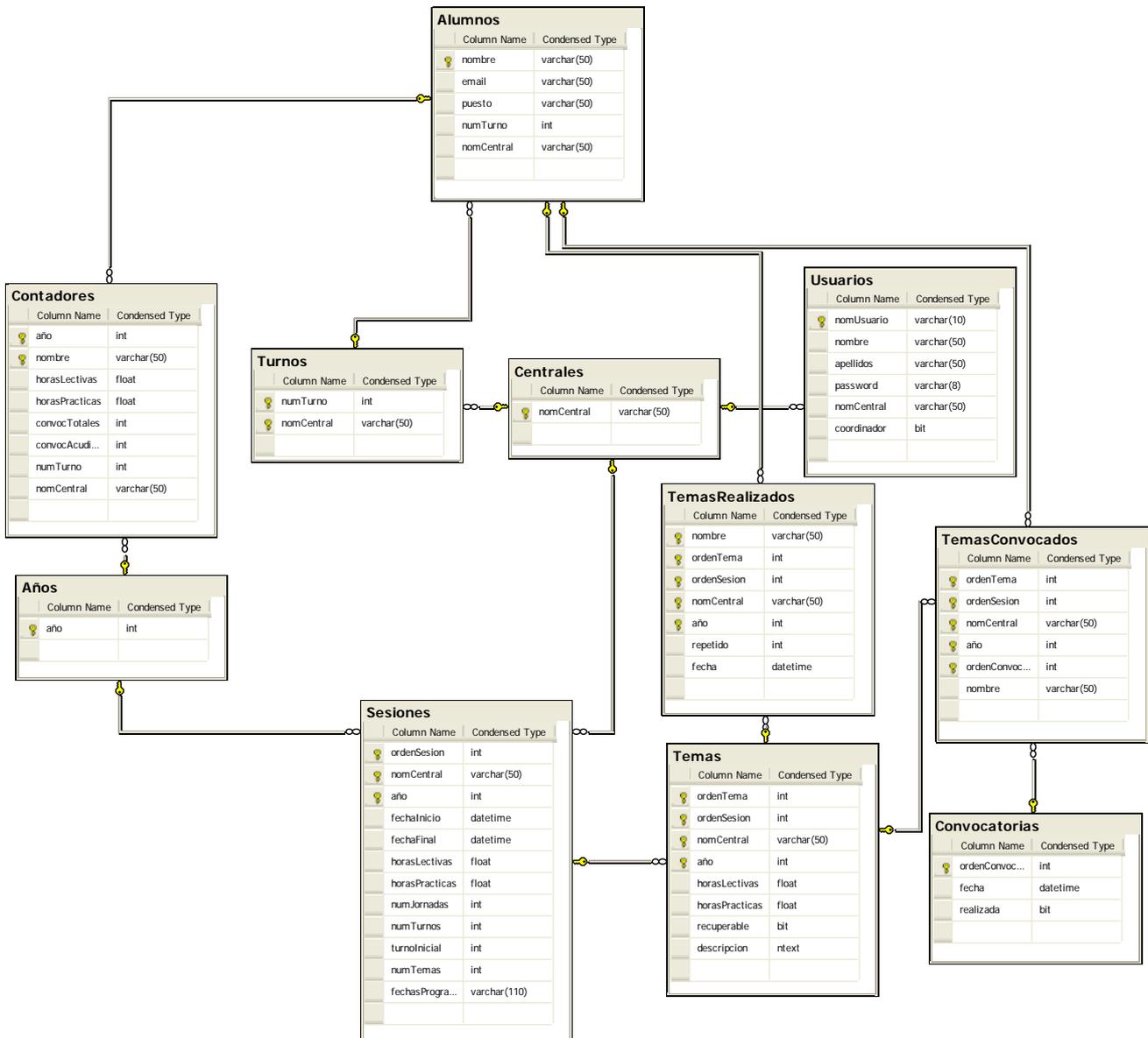


Figura 3: Diagrama lógico de la base de datos

2.3. Descripción de las tablas

2.3.1. Tabla Usuarios

La tabla Usuarios es la que contiene la información de los usuarios del sistema, instructores y coordinadores. Esta información será el nombre y apellidos del usuario, el nombre de usuario, la contraseña de entrada y si el usuario es coordinador o no.

Columna	Tipo	Clave	Descripción
nomUsuario	char(10)	PK	Es el nombre de usuario del instructor. Se utiliza para identificar la persona que accede al sistema.
nombre	char(20)		Nombre real de la persona que accede al sistema.
apellidos	char(30)		Apellidos del usuario.
password	char(10)		Contraseña de entrada.
nomCentral	char(20)	FK	Central a la cual está asignado el usuario. Solo podrá ver los datos de dicha central.
coordinador	bit		Booleano que se usará para ver si el usuario es coordinador o no.

2.3.2. Tabla Centrales

En esta tabla se guardarán los nombres de las centrales nucleares de las que se dispondrán. En principio, este número es fijo e igual a dos, pero se deja la posibilidad de ampliar en un futuro.

Columna	Tipo	Clave	Descripción
nomCentral	char(20)	PK	Nombre de la central nuclear.

2.3.3. Tabla Turnos

En esta tabla se guardará el número de turnos que hay en cada central. En principio son siete, pero existe la posibilidad de que se amplíe en un futuro. Cada turno tendrá asignado un número de alumnos.

Columna	Tipo	Clave	Descripción
numTurno	int	PK	Es el número de orden del turno.
nomCentral	char(20)	PK,FK	Central a la cual está asignado el turno.

2.3.4. Tabla Alumnos

Esta tabla guarda la información de todos los alumnos. Se necesita saber el nombre, el puesto, el email, el turno en el que está en ese momento y la central que tienen asignada. Como un alumno no puede cambiar de central, este último valor servirá solo para poder acceder más rápidamente a los datos de un usuario.

Columna	Tipo	Clave	Descripción
nombre	char(30)	PK	Nombre y apellidos del alumno.
email	char(20)		Dirección de correo del alumno.
puesto	char(4)		Siglas del puesto que ocupa: JT – Jefe de Turno, AJT – Ayudante de Jefe de Turno, JS – Jefe de Sala, OR – Operador de Reactor, OT – Operador de Turbina, OP – Operador en prácticas. Las siglas podrán acabar con un número, 1 ó 2, en caso de que haya dos grupos nucleares en la misma central. En el caso de los Operadores en Prácticas, el número indicará el orden de éstos dentro del grupo, ya que puede haber más de uno.
numTurno	int	FK	Número del turno actual del alumno.
nomCentral	char(20)	FK	Central a la cual pertenece el alumno.

2.3.5. Tabla Años

La tabla de años recoge el número de años de los que se guarda registro.

Columna	Tipo	Clave	Descripción
años	int	PK	Contiene el año.

2.3.6. Tabla Contadores

En esta tabla se recogerá la historia de cada alumno. Dada un nombre de alumno y un año tendremos las horas realizadas, tanto lectivas como prácticas, las convocatorias totales y acudidas y, además, el turno en el que estuvo a finales de ese año. Este último valor es necesario para poder generar los listados y estadísticas de otro año anterior.

Columna	Tipo	Clave	Descripción
año	int	PK, FK	El año del que queremos ver los contadores.
nombre	char(30)	PK	Nombre del alumno.
horasLectivas	float		Horas lectivas realizadas.
horasPracticas	float		Horas prácticas realizadas.

Columna	Tipo	Clave	Descripción
convocTotales	int		Número total de convocatorias para recuperar algún tema en ese año. No se tendrán en cuenta las convocatorias de las sesiones programadas oficiales.
convocAcudidas	int		Número de convocatorias para recuperar algún tema a las que realmente asistió.
numTurno	int		Número del turno al cual perteneció el alumno a final de ese año.
nomCentral	char(20)		Nombre de la central a la que pertenece el alumno.

2.3.7. Tabla Sesiones

En esta tabla se recoge la información de las sesiones que se tienen cada año para cada central. Se identifican por un número, que no tiene por qué ser correlativo en el tiempo, pero sí único cada año para cada central.

Columna	Tipo	Clave	Descripción
ordenSesion	int	PK	Número que identifica a una sesión en un año y para una central.
nomCentral	char(20)	PK, FK	Nombre de la central a la que pertenece la sesión.
año	int	PK, FK	El año.
fechaInicio	datetime		Fecha programada en la que comienza el primer grupo.
fechaFinal	datetime		Fecha programada en la que acude el último grupo. Será la última semana de impartición.
horasLectivas	float		Horas lectivas programadas.
horasPracticas	float		Horas prácticas programadas.
numJornadas	int		Número de jornadas o días programados.
numTurnos	int		Número de turnos convocados. En principio, siempre serán siete.
turnoInicial	int		Número del turno que comienza la sesión. Este valor junto con el anterior sirven para saber el orden en que acudirán los turnos, ya que siempre son consecutivos empezando por uno cualquiera.
numTemas	int		Número de temas que tiene la sesión.

Columna	Tipo	Clave	Descripción
fechasProgramadas	char(110)		Son las fechas programadas de todos los turnos. Se guarda en un string con el formato dd/mm/aaaa para cada fecha de cada turno separados por un espacio. Así, son 11 posiciones para la fecha más el espacio. La longitud del string es de 110 para guardar 10 turnos como máximo. La primera fecha será la del primer turno y las siguientes serán las de los demás turnos en orden consecutivo. Es necesario este string ya que es posible que una misma sesión no se realice en semanas consecutivas. Así, es necesario guardar todas las fechas para cada turno.

2.3.8. Tabla Temas

En esta tabla se guarda la información de todos los temas de todas las sesiones. Para identificar un tema, se necesitará el número de orden del tema dentro de la sesión, el número de la sesión en el que está incluido, la central a la cual pertenece la sesión y el año lectivo.

Columna	Tipo	Clave	Descripción
ordenTema	int	PK	Número que identifica a un tema dentro de la sesión. Será consecutivo según los vaya introduciendo el coordinador al dar de alta la sesión.
ordenSesion	int	PK, FK	Número que identifica a una sesión en un año y para una central.
nomCentral	char(20)	PK, FK	Nombre de la central a la que pertenece el tema.
año	int	PK, FK	El año.
horasLectivas	float		Horas lectivas del tema.
horasPracticas	float		Horas prácticas del tema. Si es diferente de cero, se considerará un tema práctico.
recuperable	bit		Indica si el tema puede ser recuperable o no.
descripcion	ntext		Texto que describe el tema. Puede ser de gran extensión.

2.3.9. Tabla TemasRealizados

En esta tabla se guardará la información de los temas que realiza cada alumno cada año y las veces que lo repite, en caso de que lo haga.

Columna	Tipo	Clave	Descripción
nombre	char(30)	PK, FK	Nombre del alumno.
ordenTema	int	PK, FK	Número que identifica a un tema dentro de la sesión.
ordenSesion	int	PK, FK	Número que identifica a una sesión en un año y para una central.
nomCentral	char(20)	PK, FK	Nombre de la central a la que pertenece el tema.
año	int	PK, FK	El año.
repetido	int		Número de veces que el alumno repite el tema. Su valor por defecto es 0, que indicará que sólo lo ha hecho una vez.
fecha	datetime		Fecha en la cual se realiza el tema. Necesaria para saber si se ha hecho en fecha programada o en recuperación.

2.3.10. Tabla Convocatorias

Se guardan las convocatorias para recuperar algún tema que se van realizando en el orden en que los instructores las van introduciendo en el sistema. Sólo se necesita saber la fecha de la convocatoria y si se realiza o no.

Columna	Tipo	Clave	Descripción
ordenConvocatoria	int	PK	El número de orden de la convocatoria.
fecha	datetime		Fecha en la que se ha convocado a recuperar algún tema a un alumno.
realizada	bit		Indica si el alumno acudió o no a la convocatoria.

2.3.11. Tabla TemasConvocados

En esta tabla se guardarán los temas a los que se convoca cada alumno en una convocatoria de recuperación. Una convocatoria puede ser un único tema de una única sesión o bien varios temas de sesiones diferentes.

Columna	Tipo	Clave	Descripción
nombre	char(30)	FK	Nombre del alumno.
ordenTema	int	PK, FK	Número que identifica a un tema dentro de la sesión.
ordenSesion	int	PK, FK	Número que identifica a una sesión en un año y para una central.
nomCentral	char(20)	PK, FK	Nombre de la central a la que pertenece el tema.

Columna	Tipo	Clave	Descripción
año	int	PK, FK	El año.
ordenConvocatoria	int	PK, FK	El orden de la convocatoria.

2.4. Diseño de clases

2.4.1. Clases del Modelo

Las clases del modelo son todas las que tienen que ver con los datos del problema. Se dividen en dos grupos:

- Las clases que corresponden a entidades del problema
- Las clases gestoras correspondientes a las entidades que requieren operaciones en la base de datos. Estas clases se llamarán GestorX donde X corresponderá a una clase de la entidad correspondiente.

2.4.2. Diagrama de clases de las clases de entidades del problema

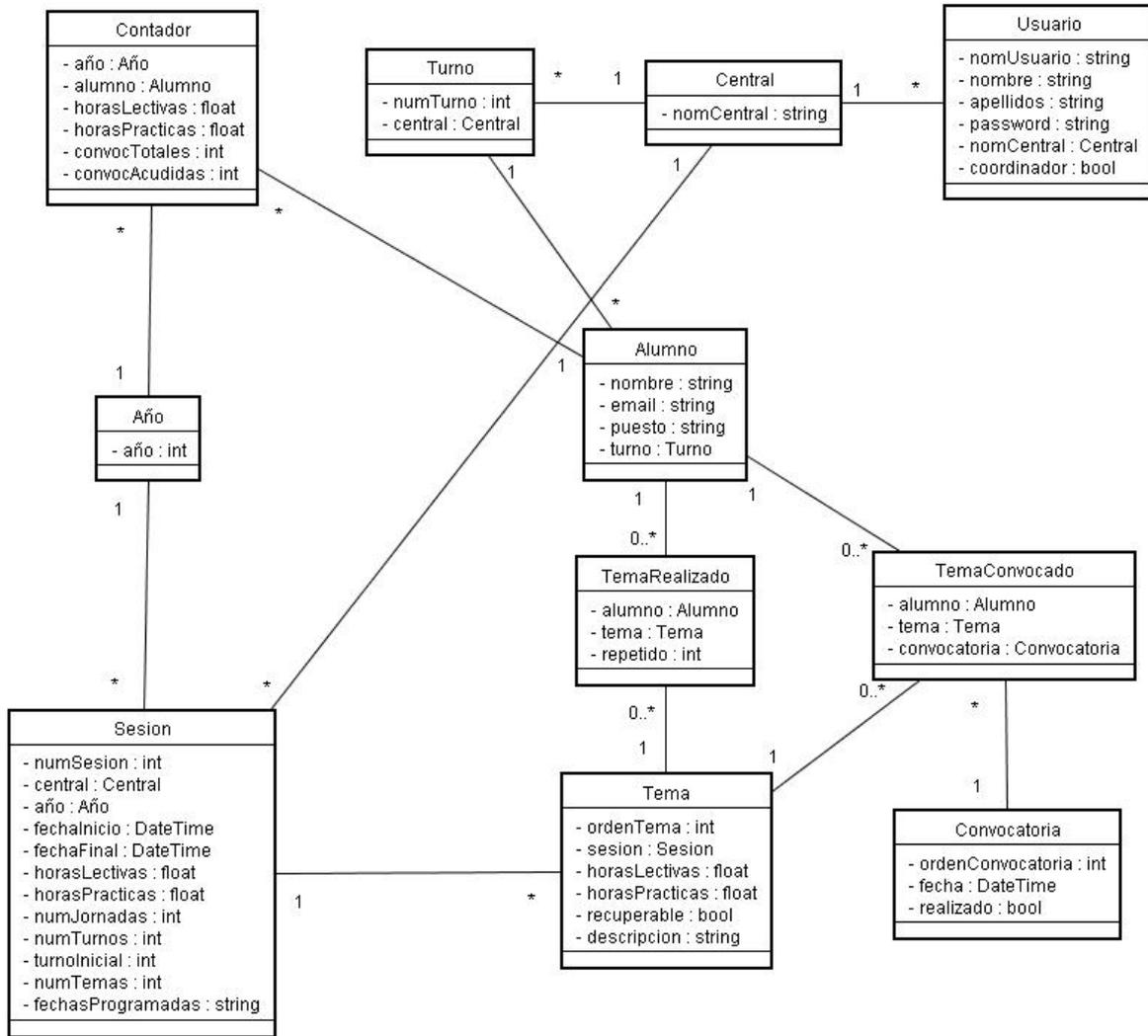


Figura 4: Diagrama de clases de las entidades del problema

No se incluyen los constructores de clase para simplificar. Todos los atributos se implementarán como propiedades.

2.4.3. Diagrama de clases de las clases gestoras

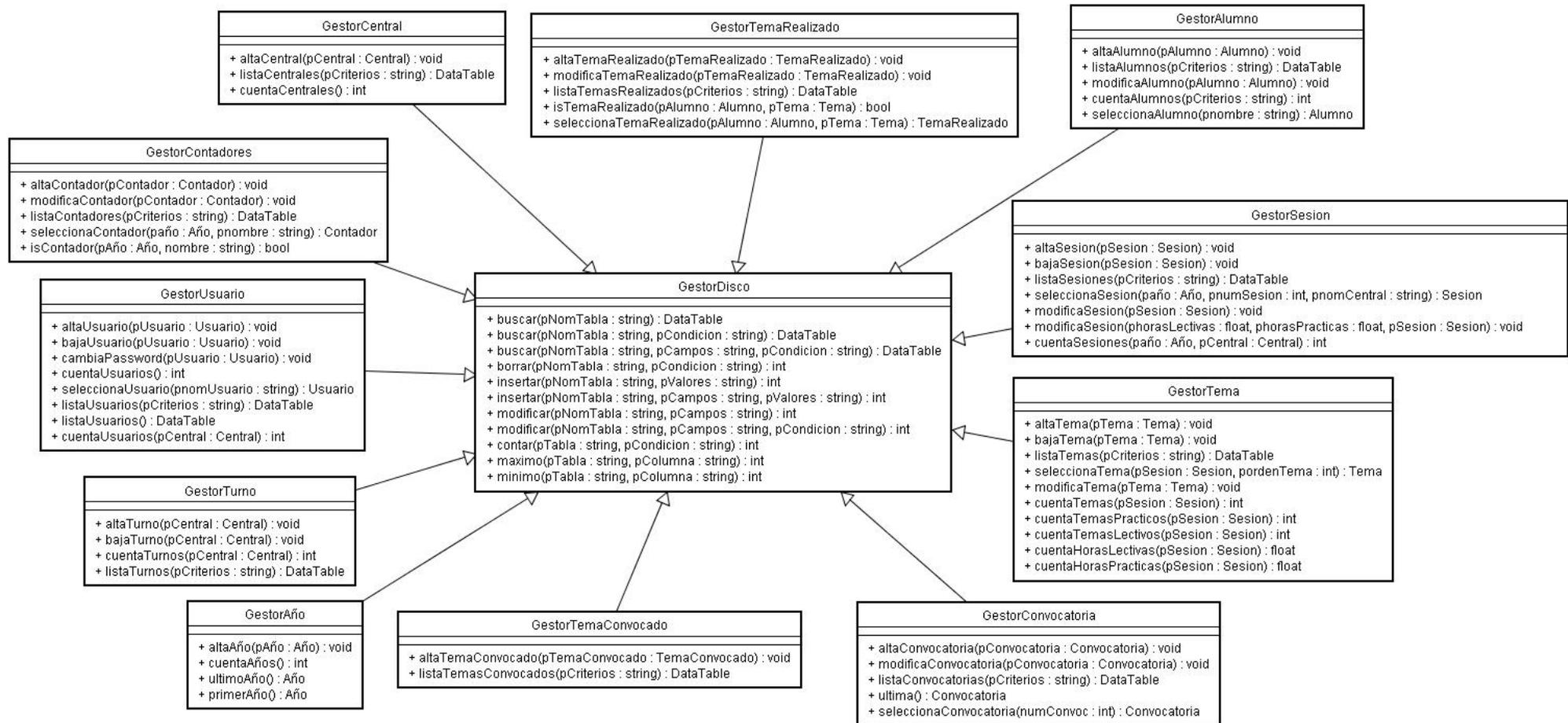


Figura 5: Diagrama de clases de las entidades gestoras

2.4.4. Clases Controlador y Vista

Gracias a la arquitectura “core-behind” de *ASP.NET*, a cada vista le corresponde un controlador que gestionará las acciones que el usuario realice en la vista.

A partir del prototipo de la aplicación diseñado, se obtiene:

Fichero Vista	Clase Controlador	Descripción
alumnos.aspx	alumnos.aspx.cs	Pantalla que permite a un coordinador gestionar los alumnos y turnos de su central.
convocatorias.aspx	convocatorias.aspx.cs	Permite a un instructor ver el estado de las convocatorias de cualquier alumno de su central.
imparticiones.aspx	imparticiones.aspx.cs	Permite a un instructor la gestión de las asistencias a los temas impartidos.
informe.aspx	informe.aspx.cs	Permite a un instructor generar en un fichero Excel el estado de las imparticiones del año seleccionado.
inicio.aspx	inicio.aspx.cs	Pantalla inicial donde el usuario introducirá su nombre y contraseña.
password.aspx	password.aspx.cs	Pantalla de cambio de contraseña.
pendientes.aspx	pendientes.aspx.cs	Permite a un instructor ver los temas recuperables pendientes que tienen los alumnos de su central.
principal.aspx	principal.aspx.cs	Pantalla inicial una vez el usuario ha introducido su contraseña. Muestra el menú de pestañas.
sesiones.aspx	sesiones.aspx.cs	Permite a un coordinador gestionar las sesiones a impartir por los instructores.
usuarios.aspx	usuarios.aspx.cs	Permite a los coordinadores ver los datos de los usuarios del sistema, así como dar de alta nuevos usuarios. Además, podrán dar de alta una nueva central y asignarle desde allí nuevos usuarios.

2.5. Seguridad

La aplicación va a ejecutarse únicamente desde una intranet corporativa y no se prevé su uso externo desde Internet, por lo que no se ha realizado un gran esfuerzo en este apartado. Se han tratado temas básicos de seguridad más relacionados con el funcionamiento correcto y seguro de la aplicación que contra ataques externos, ya que no se éstos no se entienden.

2.5.1. Autenticación

La autenticación de la aplicación se realiza mediante autenticación Windows y variables de sesión.

La autenticación Windows se realiza mediante la frase en el archivo web.config de la aplicación:

```
<authentication mode="Windows" />
```

Es decir, cualquier usuario podrá acceder a la página inicial desde un enlace o bien introduciéndola directamente en la dirección del explorador. Una vez allí, deberá introducir su nombre de usuario y contraseña, y se mira en la base de datos que exista. Si existe, se le da acceso a la página principal y se guarda como variable de sesión el nombre de usuario y la central asociada, que serán necesarios para acceder a todas y cada una de las páginas de la aplicación.

Al iniciar una sesión en *ASP.NET* el servidor le añade una *cookie* llamada *ASP.NET_SessionID* *cookie* si el usuario tiene las *cookies* activadas o bien utilizamos la sesión sin *cookie*:

```
<system.web>  
<sessionState sqlConnectionString="Data Source=(local)\; Integrated  
Security=yes;"  
    cookieless="AutoDetect" />
```

Para utilizar la aplicación sin *cookies*, se ha tenido en cuenta en llamar a todas las direcciones como direcciones relativas.

De una forma u otra, se guarda en la sesión dos variables que son necesarias para acceder a cada página. Así, si no se ha pasado por la página inicial, no se han guardado las variables de sesión y aunque el usuario intente entrar directamente a una página sin pasar por la inicial se generaría un error.

2.5.2. Inyección SQL

Para evitar inyección SQL se ha seguido el documento de referencia de M. Harper donde se explica como hacer frente y evitar este tipo de ataques.

Los usuarios no son ni mucho menos expertos en base de datos y es muy difícil, por no decir imposible, que alguien intente un tipo de ataque contra esta aplicación.

Sin embargo, se han seguido unas guías mínimas para evitar errores en las inserciones y búsquedas en la base de datos frutos de errores en la introducción de datos por parte de los usuarios.

Así, se limita la longitud de los controles *TextBox* al mínimo posible. Solo en un caso es posible introducir una gran cantidad de caracteres en el *TextBox* necesarios para la introducción de la descripción de un tema nuevo.

En todos los controles *TextBox* se ha realizado una subrutina de eliminación del carácter “” propio del lenguaje SQL.

En los *TextBox* que son obligatorios que el usuario introduzca un número o una fecha, se verifica que los datos estén bien introducidos antes de guardarlos en la base de datos.

En cuanto a la prohibición de palabras claves de SQL tipo SELECT, DROP, INSERT o DELETE, por ejemplo se ha dejado que el usuario las introduzca, ya que en las funciones de inserción de datos en la base de datos, todos los valores de todos los campos se realizan con el método *insertar* de la clase *GestorDisco*:

```
public int insertar(string pNomTabla, string pCampos, string pValores)
{
    int numRows = 0;
    string strSelect = "INSERT INTO " + pNomTabla + "(" +
        pCampos + ") VALUES(" + pValores + ")";

    SqlConnection Conexion = new SqlConnection(strConexion);
    SqlCommand Comando = new SqlCommand(strSelect, Conexion);

    try
    {
        Conexion.Open();
        numRows = Comando.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        ERROR = ex.ToString();
    }
    finally
    {
        Conexion.Close();
    }
    return numRows;
}
```

Con lo que las palabras claves que el usuario pueda introducir, al eliminar el carácter “”, quedan siempre dentro de la variable *pValores*, siempre entre comillas simples introducidas por la aplicación, por lo que el lenguaje SQL nunca las tendrá en cuenta y las introducirá en la base de datos como si fuesen cualquier otra palabra.

2.5.3. Perfiles de usuario

Para el funcionamiento de la aplicación se han tenido en cuenta dos perfiles de usuarios distintos:

- Usuario normal o instructor: Este usuario solo tendrá acceso a la parte de gestión de las asistencias y las convocatorias, así como a generar el informe del estado actual.
- Usuario gestor o coordinador: Además de todo lo anterior, tendrá acceso a la gestión de las sesiones, de los alumnos, de las centrales y de los usuarios.

Para identificar uno u otro, al entrar en la aplicación, cada usuario tiene definido en la base de datos una variable booleana, *coordinador*, que le dará acceso a las pantallas propias de un coordinador o no. Esta variable también se guarda como variable de estado, por lo que al generarse cada vista, se comprueba si el usuario es coordinador y no y mostrará solo las pestañas de acceso a los distintos apartados según el nivel de usuario.

2.5.4. Mejoras en la seguridad

Con el tratamiento realizado en el proyecto, se cree que es suficiente para la ejecución normal en la intranet de la empresa, ya que serán los administradores de red los que controlen la posible intrusión a ésta desde el exterior. Sin embargo, si se quisiese hacer funcionar en Internet para tener acceso desde cualquier punto, habría que seguir una serie de mejoras en la seguridad, como podrían ser:

- Cifrado de la base de datos: Ahora mismo la base de datos no está cifrada para permitir una cómoda gestión. Sin embargo, es posible cifrarla. Por ejemplo, usando el algoritmo simétrico 3DES.
- Denegación de Servicio (DoS): Se produce cuando hay multitud de máquinas haciendo peticiones HTTP al servidor. Al utilizar IIS no hay mucho problema, ya que solo permite un máximo de 10 usuarios, que son mucho más que los instructores existentes. Aunque si quisiéramos aumentar este número, se podría migrar a *Windows Server 2003*, por ejemplo, que no tiene esta restricción. En este caso, tendríamos que hacer frente a este potencial ataque mediante el uso de algún *firewall* que disponga de una protección frente este tipo de ataques.
- Seguridad SSL: En una intranet no es obligatorio el uso de SSL, ya que se utilizan *switchs* que realizan la segmentación de red y por tanto, un atacante que quiera utilizar un *sniffer* para robar una contraseña debería tener acceso físico a la red (que no es *wireless*), cosa bastante improbable debido al tipo de empresa y sector. En cambio, en el uso de Internet sí que sería altamente recomendable
- Autenticación mediante *forms*: Hemos visto que ahora la autenticación a la base de datos es mediante Windows y gestionada por IIS y a la aplicación mediante un login de entrada y variables de sesión. Es posible depurar esta última técnica, haciendo una autenticación mediante *forms*, con lo que al intentar abrir cualquier página sin haber pasado por la página de login, te redirecciona a ésta sin dar ningún mensaje de error.

2.6. Requisitos técnicos

El *software* a utilizar será todo **Microsoft**® ya que será el que mejor se integre:

- Lenguaje de desarrollo: **C#** y **ASP.NET 2.0**
- Entorno de desarrollo: **Visual Studio 2005**

Este *software* es necesario solo para el desarrollo, y no será necesario tenerlo instalado en el servidor una vez se haya realizado.

En el servidor se necesita:

- **.NET Framework 2.0 de Microsoft**
- Servidor web: **Internet Information Services** ofrecido por **Windows XP Professional**
- Sistema Gestor de Base de Datos: **SQL Server 2005**
- Gestor ofimática: **Microsoft Office 2003**. Sólo serán necesarios los programas **MS Excel** y **MS Outlook**, que se utilizan para la realización del informe final y para el envío de las convocatorias de recuperación a cada alumno.

En los clientes:

- Navegador actualizado: **Internet Explorer 6.0** o **Mozilla Firefox 1.5.0.x**

En cuanto al *hardware*, se necesitará un servidor con una configuración media del mercado, memoria RAM normal y poco espacio en el disco duro, solo el necesario para la base de datos, que con 1Gb será más que suficiente.

2.7. Componentes externos

Con el objeto de obtener mayor productividad se ha intentado utilizar los recursos disponibles en Internet. Esta práctica es habitual en entornos profesionales, aunque en este proyecto hay una limitación económica.

Se verán ahora los componentes externos utilizados.

2.7.1. MessageBox

MessageBox^[15] es una clase que permite utilizar en ASP.NET la funcionalidad ofrecida por la API de Windows de mostrar un mensaje de información al usuario.

Se utiliza múltiples veces a lo largo de la aplicación para advertir al usuario un error en los datos introducidos, o simplemente para informarle que su acción ha sido realizada con éxito.



Figura 6: Ejemplo del control MessageBox

2.7.2. PopUp Calendar

PopUp Calendar^[16] se utiliza tanto como calendario como para “Date Picker”, es decir, tanto para mostrar al usuario las fechas como para facilitarle la introducción de fechas en controles **TextBox**. Se utiliza usando una pequeña llamada Javascript en el enlace del calendario para que abra la pantalla ASP.NET con el control **Calendar**, que se encargará de “fabricar” el calendario y mostrarlo al usuario.

El control coloca la fecha seleccionada en el **TextBox** padre con el formato dd/mm/aaaa.

Decir que esta funcionalidad es una ayuda para el usuario, ya que siempre puede introducir las fechas directamente en el **TextBox** y la aplicación verificará que se ha introducido correctamente.

En la siguiente figura se muestra un ejemplo de su uso.

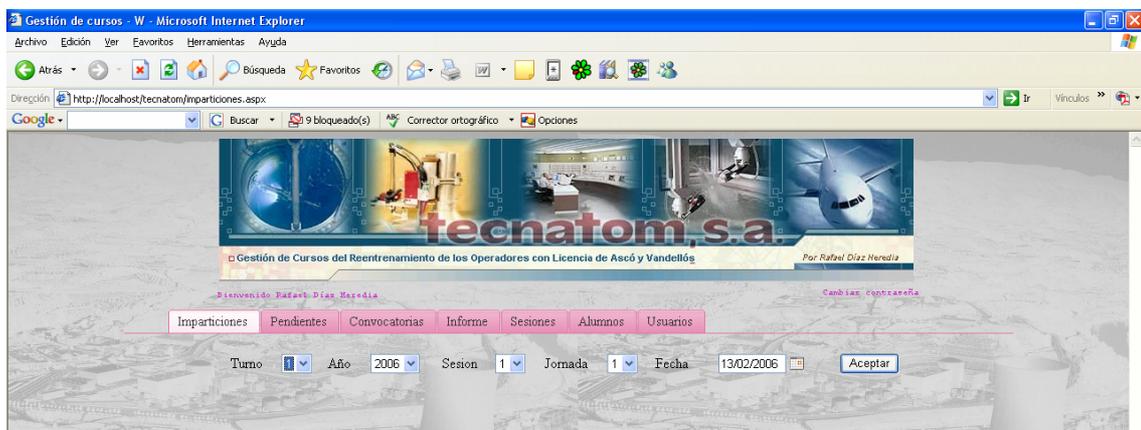


Figura 7: Ejemplo de uso del control PopUp Calendar



Figura 8: Aspecto del control PopUp Calendar

En la página origen, este control se daba en *VisualBasic*, pero se ha preferido traducirlo a *C#*, para que todo el proyecto tenga una coherencia.

2.7.3. SetFocus

En *ASP.NET* no se puede establecer por defecto que componente tendrá el “focus” una vez se carga la página. Así, utilizamos este control, **SetFocus**^[17] para darle la vista al componente que más nos interese al cargar una página.

2.7.4. msgBox: MessageBox & ConfirmationBox

Este control tiene una doble funcionalidad: la de mostrar un mensaje de información al usuario (como el control *MessageBox* anteriormente citado) y la de pedir una confirmación del usuario para realizar una acción que será crítica.

Por ejemplo, eliminar un turno tendrá unas grandes consecuencias en la gestión, y hay que estar seguro que el usuario desea eliminarlo conscientemente y no es un error. Por eso se le pide confirmación mediante este control.



Figura 9: Aspecto del control msgBox en su funcionalidad de ConfirmationBox

Este control estaba programado para *ASP.NET 1.1* y para poder utilizarlo en *ASP.NET 2.0* hay que realizar una rebaja de la seguridad de la página:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="alumnos.aspx.cs" Inherits="alumnos"
EnableEventValidation="false" %>
```

```
<%@ Register Assembly="msgBox" Namespace="BunnyBear" TagPrefix="ccl" %>
```

Esta es la cabecera de una página ASPX que utiliza el componente **msgBox**.

Se ha de poner el atributo *EnableEventValidation* en valor **false**. Este atributo es propio de *ASP.NET 2.0* y se utiliza para habilitar la validación de eventos en los escenarios de devolución de datos y devolución de llamada, por lo que, al ponerlo en **false**, estamos reduciendo la seguridad.

Así, solo utilizaremos el control **msgBox**^[18] en la variante de confirmación, y nunca como ventana de información. Para ello utilizaremos el anterior componente **MessageBox**.

Este componente se usa agregando a la lista de componentes de *ASP.NET* uno nuevo llamado **msgBox**, a partir de ese momento, se puede utilizar igual que un control estandar de *ASP.NET*:

```
<ccl:msgBox ID="MsgBox1" runat="server" />
```

En el código, se podrá llamar a la función *confirm* del control, pasando como parámetros la frase que quieres que salga en la ventana de advertencia y un control oculto que sirve para ver si el usuario ha aceptado o no.

```
MsgBox1.confirm("¿De verdad quieres modificar la composición de los  
turnos? Esto permitirá aumentar o disminuir el número de turnos.",  
"hid_f1");
```

2.7.5. Enhanced List Box Control

Este control^[19] sirve para manejar dos listas, o controles **ListBox**, y poder pasar elementos de una a otra, con los típicos botones de izquierda y derecha, o subir o bajar elementos dentro de una misma lista, con los botones de subir o bajar.

Se utiliza sólo en la página de control de alumnos, y sirve para cambiar la composición de los turnos. Tenemos en un lado el turno seleccionado, y al otro los alumnos no asignados a ningún turno (porque son nuevos, están fuera del turno estudiando para ascender de puesto, etc.). Podemos pasar alumnos de un lado a otro y en la lista del turno seleccionado, modificar los puestos para que se correspondan con los actuales.



Figura 10: Ejemplo de uso del control Enhanced List Box Control

2.7.6. Librería Interop de Microsoft Office 2003

Se utiliza la librería *Interop* de *Microsoft Office 2003* para poder llamar a clases *Excel* o *Outlook* para las funcionalidades de crear el informe final o enviar los e-mails a los alumnos al convocarles a una recuperación.

Este el paquete ofimático utilizado en la empresa, así que no supone un gasto extra a los usuarios.

3. Descripción de la aplicación “Gestión de Cursos”

La aplicación pretende realizar el control de una escuela de formación que imparte cursos a una empresa externa. Se necesita llevar un control muy exhaustivo e individual de cada alumno, ya que se han de justificar unas horas mínimas obligatorias al finalizar un año, tanto prácticas como lectivas.

La aplicación tendrá dos perfiles de usuario, uno de instructor, que permite realizar el control de las asistencias y las convocatorias de las clases de recuperación, y otro de coordinador, que, además de lo anterior, permite dar de alta a nuevos cursos (o sesiones), modificar la composición de los grupos de alumnos y gestionar los usuarios de la aplicación.

Para movernos por todas las funcionalidades de la aplicación, se usará un sistema de pestañas. En las pestañas podremos acceder a las distintas partes siempre y cuando tengamos el perfil necesario. Si no tenemos ese perfil, simplemente no saldrán las pestañas que dan acceso a las aplicaciones de los administradores o coordinadores.

Esta descripción será simplemente para conocer el estado final de cada pantalla con unas breves nociones de la funcionalidad. Si se desea una descripción más detallada sobre esta última, habrá que dirigirse al documento de Análisis.

3.1. Pantalla Inicio

Cualquier usuario ha de estar registrado, ya que, cada usuario, tiene asignado una central y sólo podrá manejar los datos asociados a esa central. Nunca verá nada relativo al resto de centrales de la aplicación. Esta asociación se realiza al dar de alta al usuario en el sistema por parte de un coordinador y será irremplazable. Un usuario tendrá una única central asociada. Si por cualquier motivo, hay un instructor que necesita tener acceso a más de una central, se le daría de alta en el sistema con tantos nombres de usuario como centrales requiera.

Así, se pedirá nombre de usuario y contraseña.

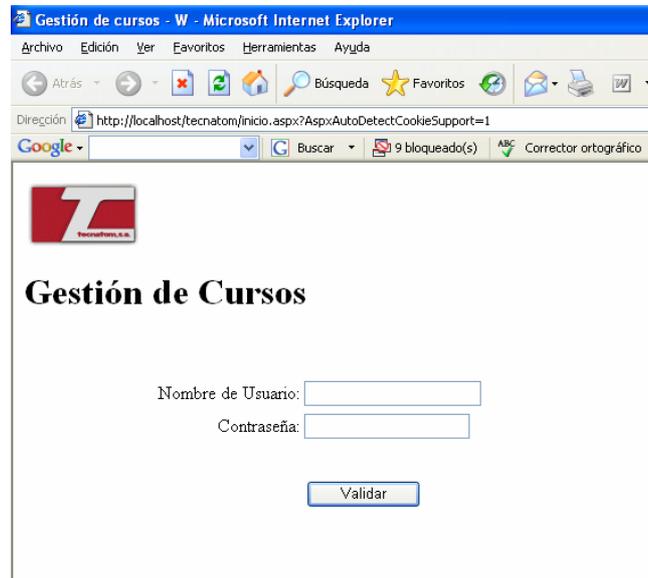


Figura 11: Pantalla Inicio

3.2. Pantalla Principal

A esta pantalla solo se podrá acceder desde la primera vez que un usuario se valida en la aplicación o desde la pantalla de cambio de contraseña. Solo sirve para mostrar la imagen de bienvenida, y la barra de pestañas para acceder a las distintas partes de la aplicación. Esta barra será diferente según el perfil del usuario.

En la parte superior aparece una imagen fija, que es el logo de la empresa. Además, aparecerá un mensaje de bienvenida con el nombre del usuario y, en el lado contrario, un link para que el usuario pueda cambiar la contraseña. Tanto el logo, como estas dos etiquetas serán fijas para toda la sesión del usuario.

La barra de pestañas, en cambio, reflejará en cada momento en que parte de la aplicación estamos, cambiando la etiqueta correspondiente de color.

Esta barra de etiquetas no es ningún control *ASP.NET*. Se ha conseguido mediante *CSS* y código *JavaScript*. Se ha preferido así por dar un toque estético más profesional.

En la figura 12 se muestra esta pantalla con la barra de pestañas para un coordinador.

En la figura 13, se observa el detalle de esta barra para un instructor sin perfil de coordinador.

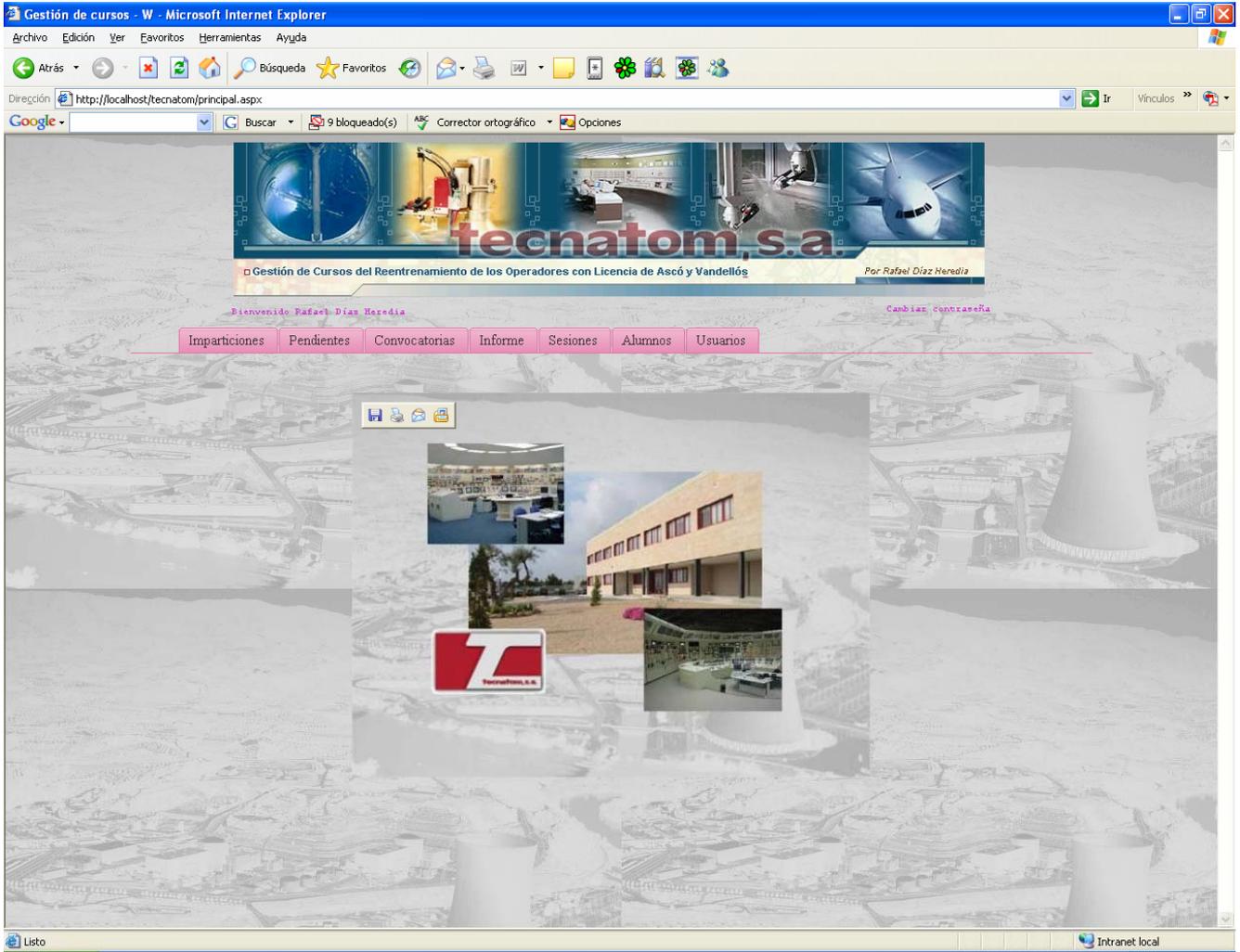


Figura 12: Pantalla principal con barra de pestañas de coordinador



Figura 13: Detalle de la barra de pestañas para un instructor no coordinador

3.3. Pantalla de cambio de contraseña

A esta pantalla se puede acceder desde cualquier punto de la aplicación una vez se haya validado el usuario. Sirve para que el usuario cambie su contraseña si lo

requiere. Se le pedirá que introduzca dos veces la nueva contraseña para verificar que no se ha equivocado.

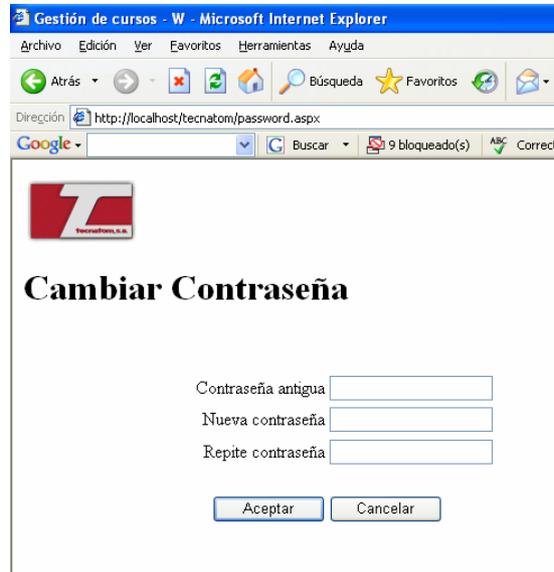


Figura 14: Pantalla de cambio de contraseña

3.4. Pantalla Imparticiones

En esta pantalla un usuario normal podrá gestionar las asistencias a los cursos, tanto a los programados como a los convocados.

Deberá introducir el número del turno del cual quiere introducir las asistencias, el año, el número de la sesión a impartir, y la jornada dentro de la semana. En el campo fecha saldrá la fecha programada para impartir esa sesión a ese turno. Si el instructor la cambia, se considerará una recuperación y será tratada como tal (sólo podrá impartir clases prácticas y teóricas recuperables, nunca temas no recuperables, ya que es imposible darlo en fechas distintas a las programadas).

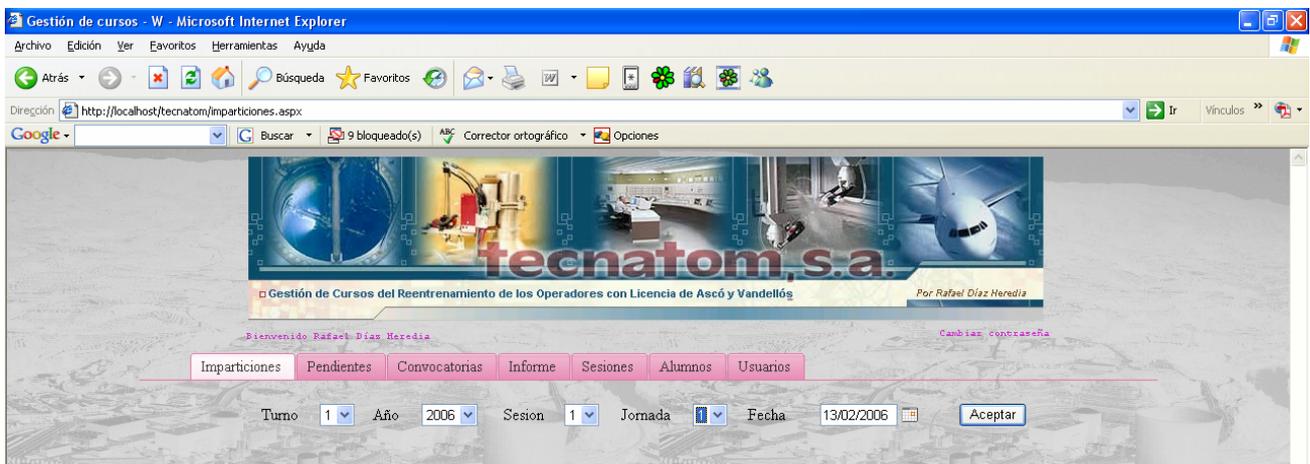


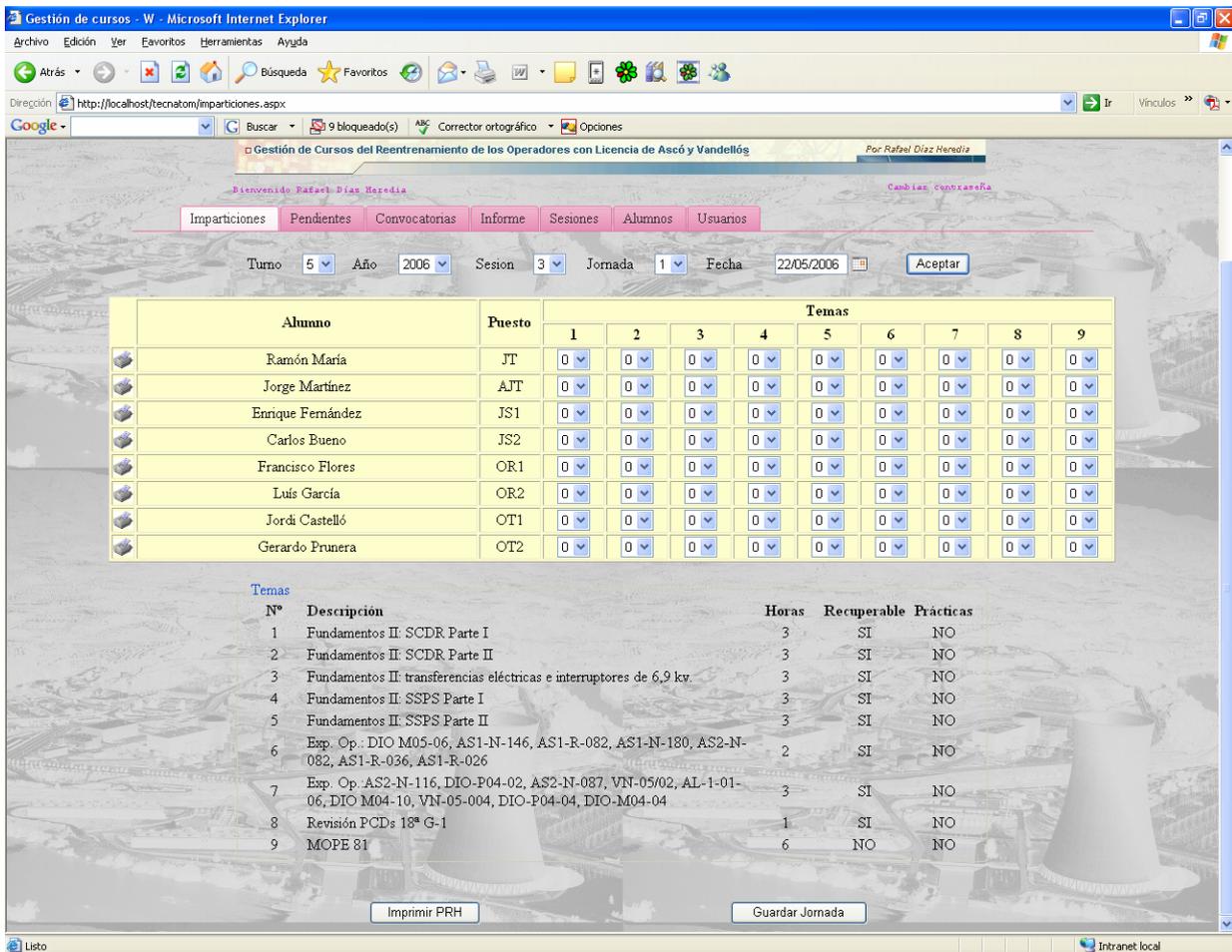
Figura 15: Pantalla Imparticiones

Al aceptar, saldrá una primera tabla con el listado de alumnos del turno seleccionado y con tantos desplegables como temas haya en la sesión. El desplegable indicará las veces que ha visto el alumno ese tema en esa jornada (lo normal es 1, que lo ha visto, ó 0, que no ha asistido; pero para los temas prácticos existe la posibilidad de 2 por si lo repite el mismo día).

Decir que si el día no coincide con el programado por el coordinador, solo saldrán activos los temas teóricos recuperables no vistos por el personal del turno y todos los prácticos.

Después, saldrá una tabla informativa en la que se relacionan los temas de la sesión y se indica, además de su descripción, el número de horas y si son lectivos o prácticos y recuperables o no.

En la Figura 16 se puede observar como se introducen las asistencias para una sesión en la fecha programada por el coordinador al dar de alta a la sesión. Todos los temas están activos. En cambio, en la Figura 17, se observa el control de las asistencias de un turno en fechas no programadas. Se observa que solo se pueden introducir los temas prácticos y los lectivos recuperables no vistos por algún alumno.



Gestión de Cursos del Reentrenamiento de los Operadores con Licencia de Ascó y Vandellós

Turno: 5 | Año: 2006 | Sesión: 3 | Jornada: 1 | Fecha: 22/05/2006

Alumno	Puesto	Temas								
		1	2	3	4	5	6	7	8	9
Ramón María	JT	0	0	0	0	0	0	0	0	0
Jorge Martínez	AJT	0	0	0	0	0	0	0	0	0
Enrique Fernández	JS1	0	0	0	0	0	0	0	0	0
Carlos Bueno	JS2	0	0	0	0	0	0	0	0	0
Francisco Flores	OR1	0	0	0	0	0	0	0	0	0
Luis García	OR2	0	0	0	0	0	0	0	0	0
Jordi Castelló	OT1	0	0	0	0	0	0	0	0	0
Gerardo Prunera	OT2	0	0	0	0	0	0	0	0	0

Nº	Descripción	Horas	Recuperable	Prácticas
1	Fundamentos II: SCDR Parte I	3	SI	NO
2	Fundamentos II: SCDR Parte II	3	SI	NO
3	Fundamentos II: transferencias eléctricas e interruptores de 6,9 kv	3	SI	NO
4	Fundamentos II: SSPS Parte I	3	SI	NO
5	Fundamentos II: SSPS Parte II	3	SI	NO
6	Exp. Op.: DIO M05-06, AS1-N-146, AS1-R-082, AS1-N-130, AS2-N-082, AS1-R-036, AS1-R-026	2	SI	NO
7	Exp. Op.: AS2-N-116, DIO-P04-02, AS2-N-087, VN-05/02, AL-1-01-06, DIO M04-10, VN-05-004, DIO-P04-04, DIO-M04-04	3	SI	NO
8	Revisión PCDs 18ª G-1	1	SI	NO
9	MOPE 81	6	NO	NO

Figura 16: Introducción de la asistencia a una sesión en fecha programada

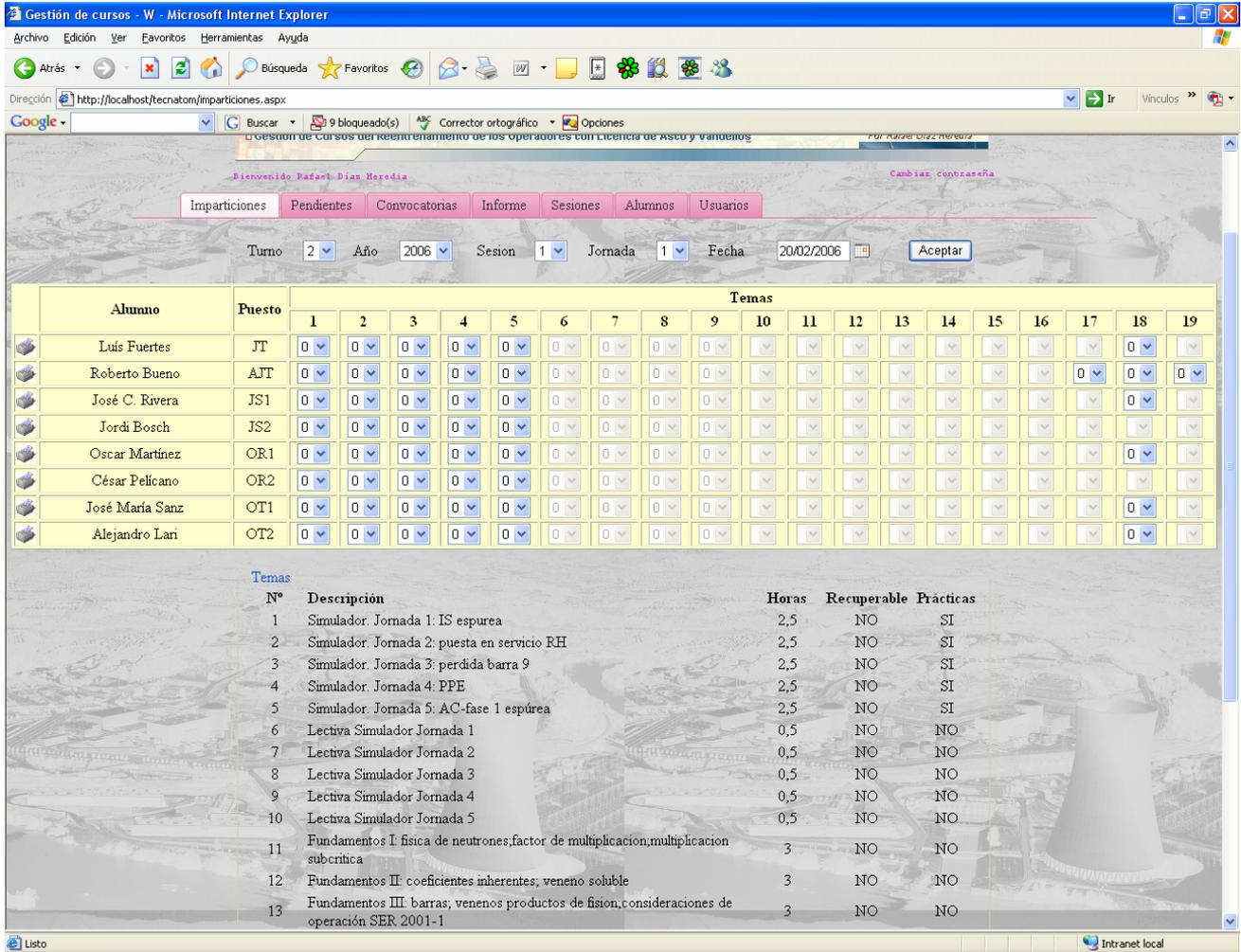


Figura 17: Introducción de la asistencia de una sesión en fecha no programada

Al final de las tablas habrá un botón para guardar el estado de la sesión (si no los cambios no se guardarán) y otro para imprimir la hoja de firmas en la que los alumnos confirman su asistencia. Esta hoja de firmas está hecha en Excel por si el instructor desea guardarse una copia, aunque lo normal es que la imprima directamente.

En la Figura 18 se adjunta un ejemplo de la hoja de firmas.

		PRH 5.09								
SESION 1										
TURNO 2		TEMAS Semana 20/02/2006			Horas teóricas	Horas prácticas	Horas teóricas realizadas	Horas prácticas realizadas	Fecha recuperación	Observaciones
Luis Fuertes	JT	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19			17,5	12,5	16,5	12,5		
Roberto Bueno	AJT	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16			17,5	12,5	14,5	12,5		
José C. Rivera	JS1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19			17,5	12,5	16,5	12,5		
Jordi Bosch	JS2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16			17,5	12,5	14,5	12,5		
Oscar Martínez	OR1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19			17,5	12,5	16,5	12,5		
César Pelicano	OR2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16			17,5	12,5	14,5	12,5		
José María Sanz	OT1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19			17,5	12,5	16,5	12,5		
Alejandro Lari	OT2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19			17,5	12,5	16,5	12,5		
TEMAS SESION 1										
		DIA	HORAS	NUMERO TEMA	DESCRIPCION_TEMA					
			2,5	1	Simulador. Jornada 1: IS espúrea					
			2,5	2	Simulador. Jornada 2: puesta en servicio RH					
			2,5	3	Simulador. Jornada 3: pérdida barra 9					
			2,5	4	Simulador. Jornada 4: PPE					
			2,5	5	Simulador. Jornada 5: AC-fase 1 espúrea					
			0,5	6	Lectiva Simulador Jornada 1					
			0,5	7	Lectiva Simulador Jornada 2					
			0,5	8	Lectiva Simulador Jornada 3					
			0,5	9	Lectiva Simulador Jornada 4					
			0,5	10	Lectiva Simulador Jornada 5					
			3	11	Fundamentos I: física de neutrones:factor de multiplicacion:multiplicacion subcritica					
			3	12	Fundamentos II: coeficientes inherentes; veneno soluble					
			3	13	Fundamentos III: barras; venenos productos de fisión;consideraciones de operación SER 2001-1					
			1	14	Modificaciones de diseño de la 16ª Recarga del Grupo II: PCDs 20073, 20341, 20335, 20423, 20964, 21147, 21247, 21408, 21454, 21481, 21597, 21637					
			1	15	Modificación de diseño de la Inyección de Zinc: PCDs 21332 y 21332-A. Fundamentos del Zinc.					
			1	16	Avance de Modificaciones de diseño de la 18ª Recarga del grupo I: PCDs 20115, 20162, 20186, 20229, 20302-6, 20352, 20498, 20562, 20987, 20995, 21000, 21035, 21051, 21159-A, 21246, 21385, 21657, 21698					
			1	17	Modificación de diseño de la sustitución de los Split Pins, PCD 21752					
			1	18	Maniobras de Recarga: Descripción de las actividades más importantes de la recarga. Vídeo de la Recarga de Ascó.					
			1	19	Maniobras de la Tapa Ligera de Ascó. Estudio de Seguridad. IOP 1.12 apartados 6.7, 6.8 y 6.9					
INSTRUCTOR										
Nombre	Fecha	Firma			ALUMNO			JEFE DE TURNO		
					Nombre	Fecha	Firma	Nombre	Fecha	Firma
JEFE DE OPERACIÓN										
JEFE DE FORMACION										
Nombre	Fecha	Firma			Nombre	Fecha	Firma			

Figura 18: Hoja oficial de firmas de asistencia

Además, se podrá generar la hoja individual, para cada alumno en las recuperaciones, haciendo un clic en la imagen de la impresora que tiene cada uno a la izquierda de su nombre.

3.5. Pantalla Pendientes

En esta pantalla se podrán ver los temas recuperables que tienen los alumnos. Se podrán ver todos los alumnos de un turno a la vez, o bien un alumno en particular.

En la Figura 19 se muestra un ejemplo de búsqueda para un alumno.

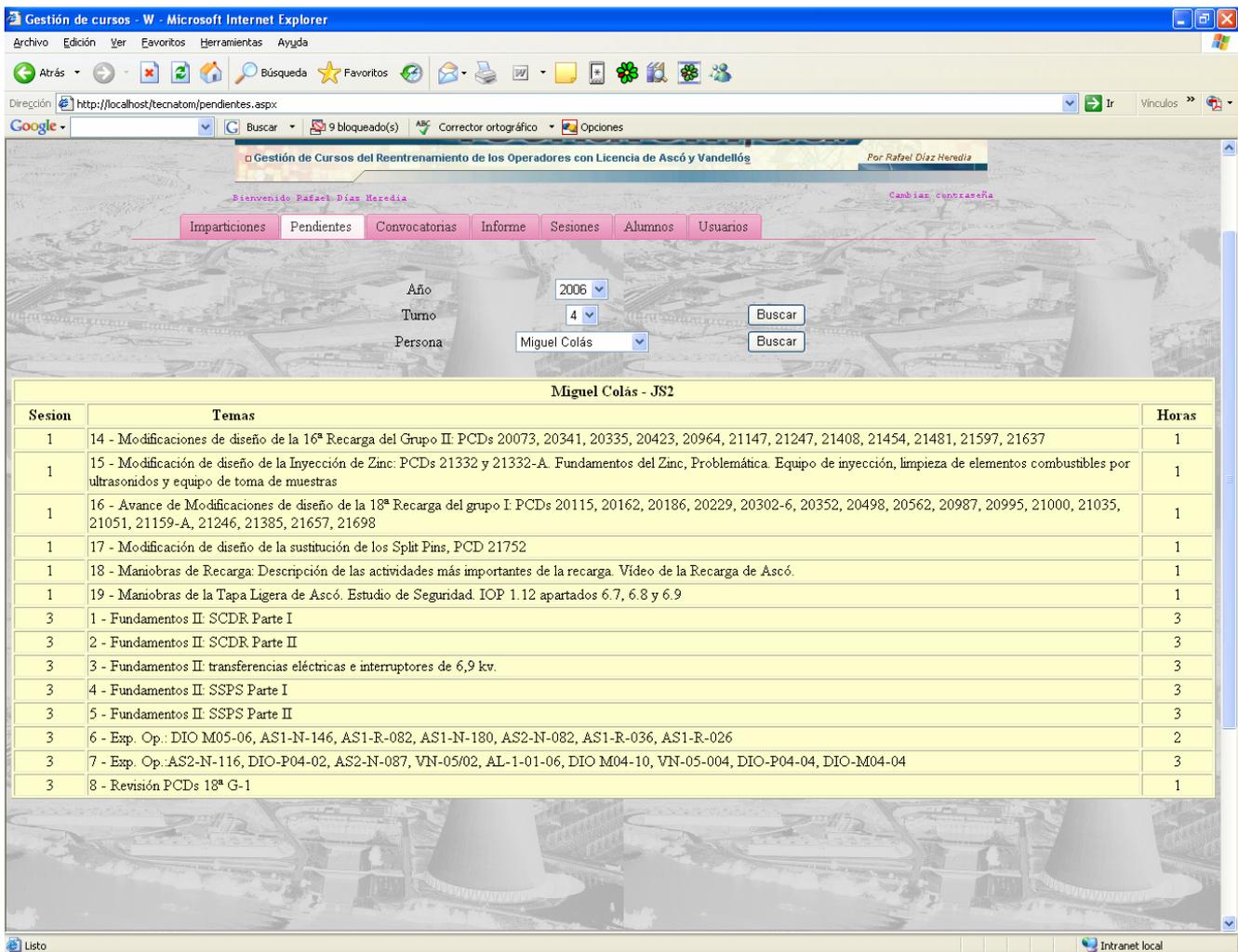


Figura 19: Temas lectivos recuperables pendientes de un alumno en la pantalla Pendientes

Al realizar la búsqueda de una única persona, aparecerá un recuadro en la parte inferior en la que se puede programar una recuperación de los temas no vistos.

Se seleccionan tantos temas como se quieran convocar, se asigna la fecha de la convocatoria y al dar al botón “Convocar” se genera un email al interesado convocándole el día elegido para realizar los temas seleccionados.

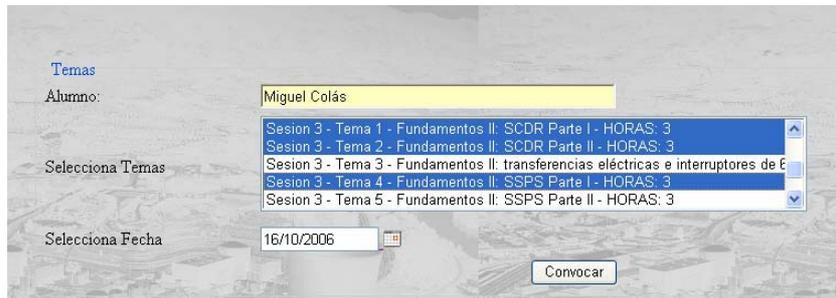


Figura 20: Detalle de la pantalla Pendientes donde se realiza una convocatoria de recuperación

3.6. Pantalla Convocatorias

Esta pestaña está muy ligada a la anterior. En ella, un instructor puede llevar un control del estado de las convocatorias de recuperación. Para cada alumno de un turno, podrá ver cuantas convocatorias tiene y si ha asistido o no a esa convocatoria. Podrá, además, confirmar la asistencia de un alumno.

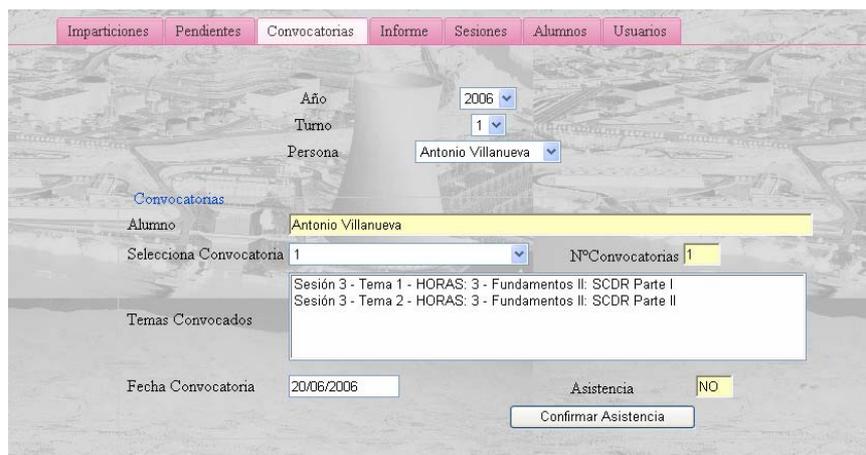


Figura 21: Detalle de la pantalla Convocatorias donde se observa una convocatoria de recuperación de un alumno

3.7. Pantalla Informe

En esta pestaña se genera, en **Excel** un informe del estado de los cursos, para todos los turnos. En el informe se ven los temas vistos en el día de la convocatoria, así como las horas pendientes y las fechas en las que ha recuperado si lo ha hecho.

Al final y para cada turno, hay unos contadores que muestran las horas teóricas y prácticas programadas y las que ha realizado cada alumno.

Este informe se genera en **Excel** por requisito del cliente, y para él, esta aplicación debe ser totalmente transparente. Así, se ha decidido generar el informe en el mismo formato que el que ahora se utiliza.



Figura 22: Pantalla Informe

En la Figura 23 se observa el detalle de los totalizadores de las horas realizadas por los alumnos de un turno.

En la Figura 24 se observa un detalle del informe final en el que se representa un turno y una sesión del año seleccionado. No se representan todas las sesiones por falta de espacio.

TOTALES			
Horas teóricas totales	Horas prácticas totales	Horas teóricas realizadas	Horas prácticas realizadas
44,5	12,5	17,5	12,5
44,5	12,5	17,5	12,5
44,5	12,5	17,5	12,5
44,5	12,5	17,5	15
44,5	12,5	17,5	12,5
44,5	12,5	17,5	12,5
44,5	12,5	17,5	12,5
44,5	12,5	17,5	12,5

Figura 23: Detalle de los totalizadores de horas de un turno en el informe anual

		SESIÓN 1						
Turno 7	TEMAS Semana 27/03/2006	Horas teóricas	Horas prácticas	Horas teóricas realizadas	Horas prácticas realizadas	Fecha de Recuperación	Observaciones	
							Pendientes recup	Ausencias a lectivas no recup
Alberto Harguindey	JT	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19	17,5	12,5	17,5	12,5		
Jordi Morales	AJT	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19	17,5	12,5	17,5	12,5		
Angel Torné	JS1	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19	17,5	12,5	17,5	12,5		
Pedro Fernández	JS2	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19	17,5	12,5	17,5	15		
Josep Olivé	OR1	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19	17,5	12,5	17,5	12,5		
Javier Ferrándiz	OR2	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19	17,5	12,5	17,5	12,5		
Carlos Moreno	OT1	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19	17,5	12,5	17,5	12,5		
Jordi Prats	OT2	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19	17,5	12,5	17,5	12,5		

TEMAS SESIÓN 1			
DÍAS	HORAS	IMERO	TEI DESCRIPCIÓN
	2,5	1	Simulador. Jornada 1: IS espúrea
	2,5	2	Simulador. Jornada 2: puesta en servicio RH
	2,5	3	Simulador. Jornada 3: perdida barra 9
	2,5	4	Simulador. Jornada 4: PPE
	2,5	5	Simulador. Jornada 5: AC-fase 1 espúrea
	0,5	6	Lectiva Simulador Jornada 1
	0,5	7	Lectiva Simulador Jornada 2
	0,5	8	Lectiva Simulador Jornada 3
	0,5	9	Lectiva Simulador Jornada 4
	0,5	10	Lectiva Simulador Jornada 5
	3	11	Fundamentos I: física de neutrones;factor de multiplicacion;multiplicacion subcritica
	3	12	Fundamentos II: coeficientes inherentes; veneno soluble
	3	13	Fundamentos III: barras; venenos productos de fisión;consideraciones de operación SER 2001-1
	1	14	Modificaciones de diseño de la 16ª Recarga del Grupo II: PCDs 20073, 20341, 20335, 20423, 20964, 21147, 21247, 21408, 21454, 21481, 21597, 21637
	1	15	Modificación de diseño de la Inyección de Zinc: PCDs 21332 y 21332-A. Fundamentos del Zinc, Problemática. Equipo de inyección, limpieza de elementos combustibles por ultrasonidos y equipo de toma de muestras
	1	16	Avance de Modificaciones de diseño de la 18ª Recarga del grupo I: PCDs 20115, 20162, 20186, 20229, 20302-6, 20352, 20498, 20562, 20987, 20995, 21000, 21035, 21051, 21159-A, 21246, 21385, 21657, 21698
	1	17	Modificación de diseño de la sustitución de los Split Pins, PCD 21752
	1	18	Maniobras de Recarga: Descripción de las actividades más importantes de la recarga. Vídeo de la Recarga de Ascó.
	1	19	Maniobras de la Tapa Ligera de Ascó. Estudio de Seguridad. IOP 1.12 apartados 6.7, 6.8 y 6.9

Figura 24: Detalle de una sesión realizada por un turno en el informe anual

3.8. Pantalla Sesiones

Esta pestaña es exclusiva de los coordinadores. En ella, un coordinador podrá dar de alta a una sesión o modificar una que aún no se haya empezado a impartir.

Primero saldrá un resumen de las existentes en el año seleccionado, como se observa en la Figura 25. Solo se podrán elegir aquellos años que tengan sesión. Al crear una sesión con fecha de impartición otro año, automáticamente se dará de alta el nuevo año en la base de datos.

Para dar de alta una nueva sesión, se necesita que el coordinador introduzca los temas, le asigne un número a la sesión e introduzca el número de jornadas y las fechas programadas para cada turno.



Figura 25: Pantalla Sesiones: Cuadro resumen de sesiones

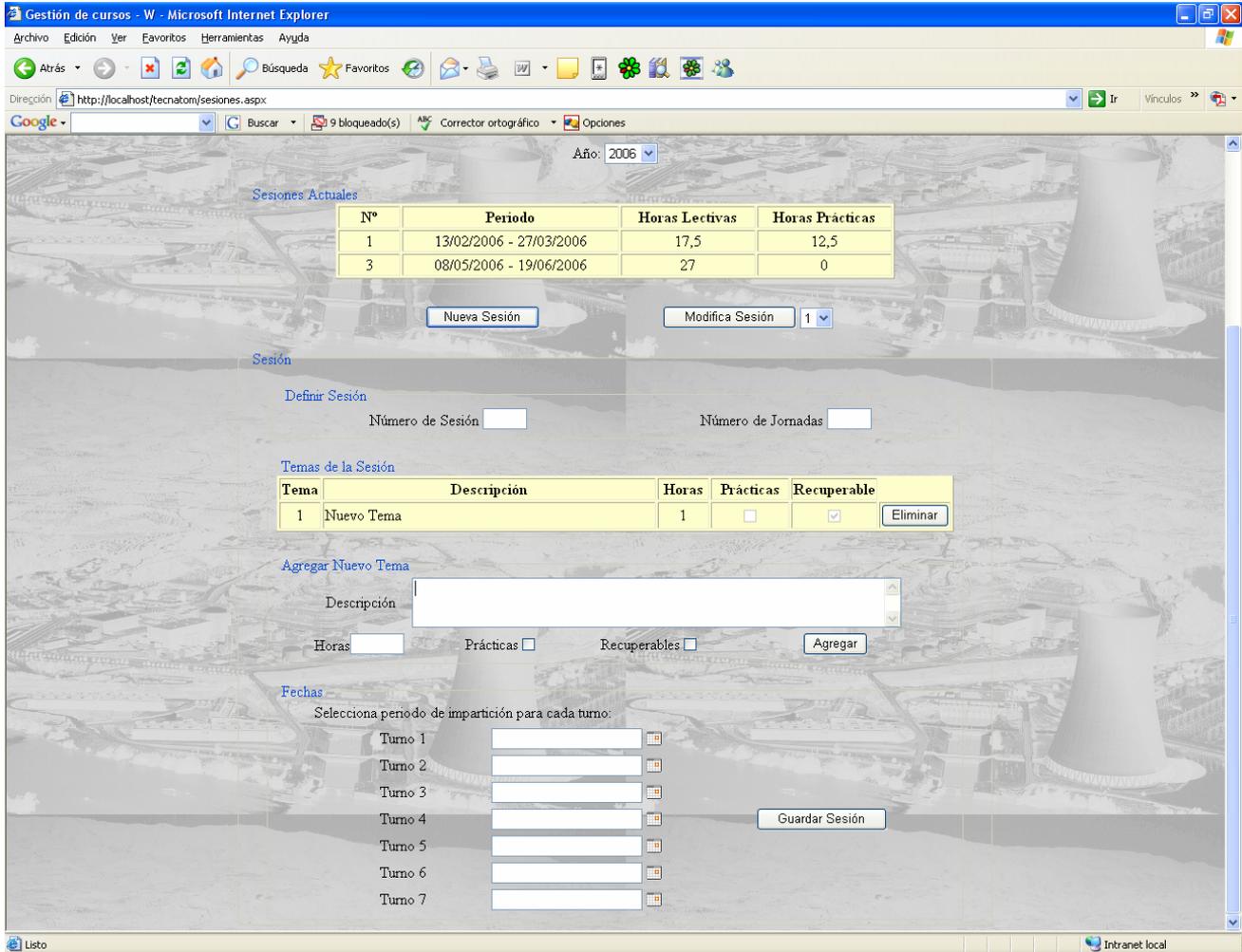


Figura 26: Introducción de una nueva sesión en la pantalla Sesiones

3.9. Pantalla Alumnos

En esta pestaña, un coordinador podrá modificar la composición de los turnos de su central asignada y dar de alta a nuevos alumnos. Se crea un turno 0 que es el que recoge los alumnos desasignados o los que ya no están en el turno de operación.

No se permite la eliminación de ninguno, pues se desea guardar toda la historia de la gestión de cursos desde que comienza a funcionar la aplicación.

Además, y como algo excepcional, se podrá incrementar o disminuir el número de turnos de la central. Esto es bastante improbable, pero se deja abierta la posibilidad para no perder la funcionalidad de la aplicación si esto ocurriese en un futuro.

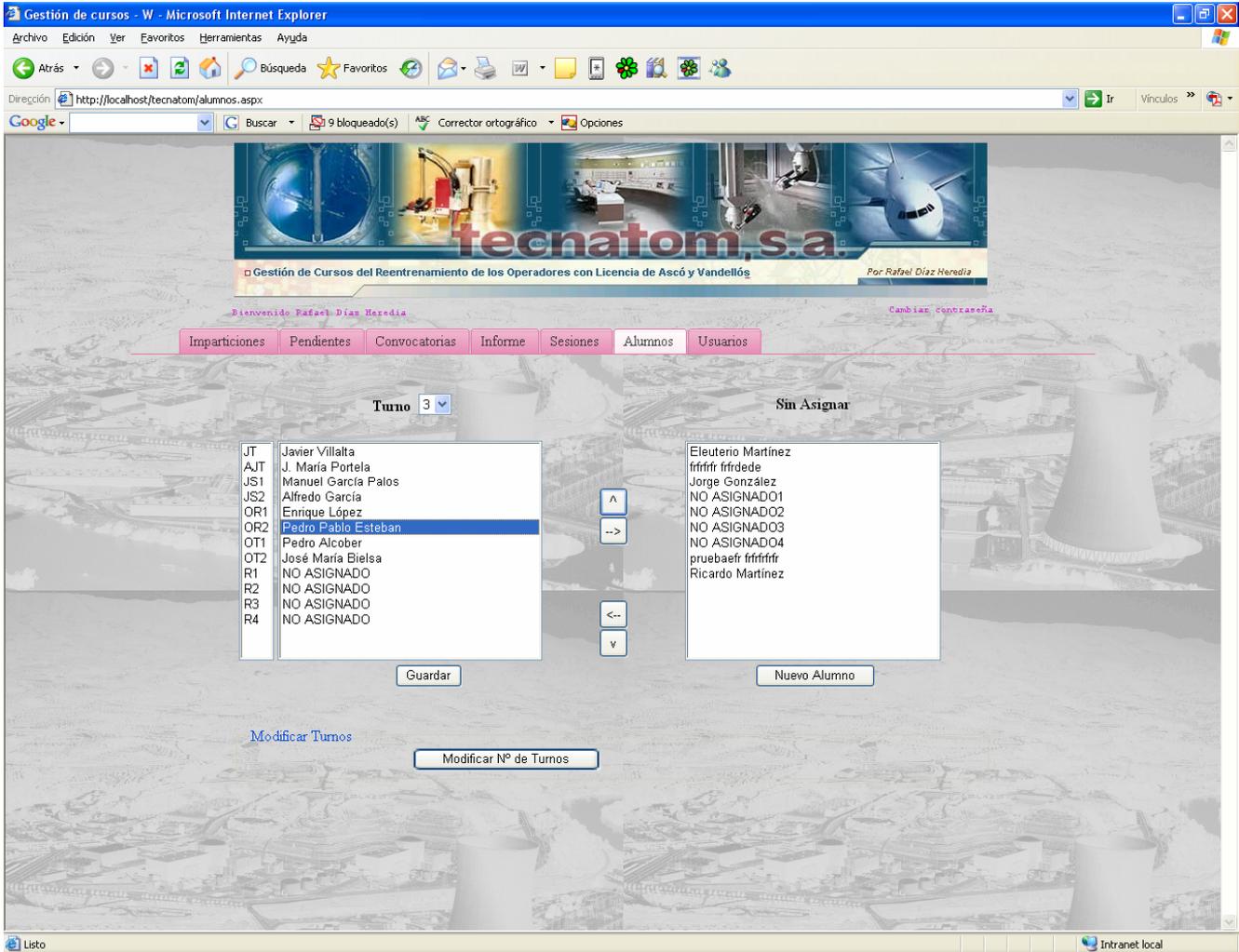


Figura 27: Gestión de alumnos en la pantalla Alumnos

3.10. Pantalla Usuarios

En ella, un coordinador podrá ver los datos de todos los usuarios de la aplicación, incluida las contraseñas (por si alguien se olvida de ellas), dar de alta a un nuevo usuario, asignándole una central o incluso dar de alta a una nueva central nuclear. Esto es bastante improbable, ya que la escuela de formación solo trabaja con Ascó y Vandellós II, pero se deja la posibilidad abierta.

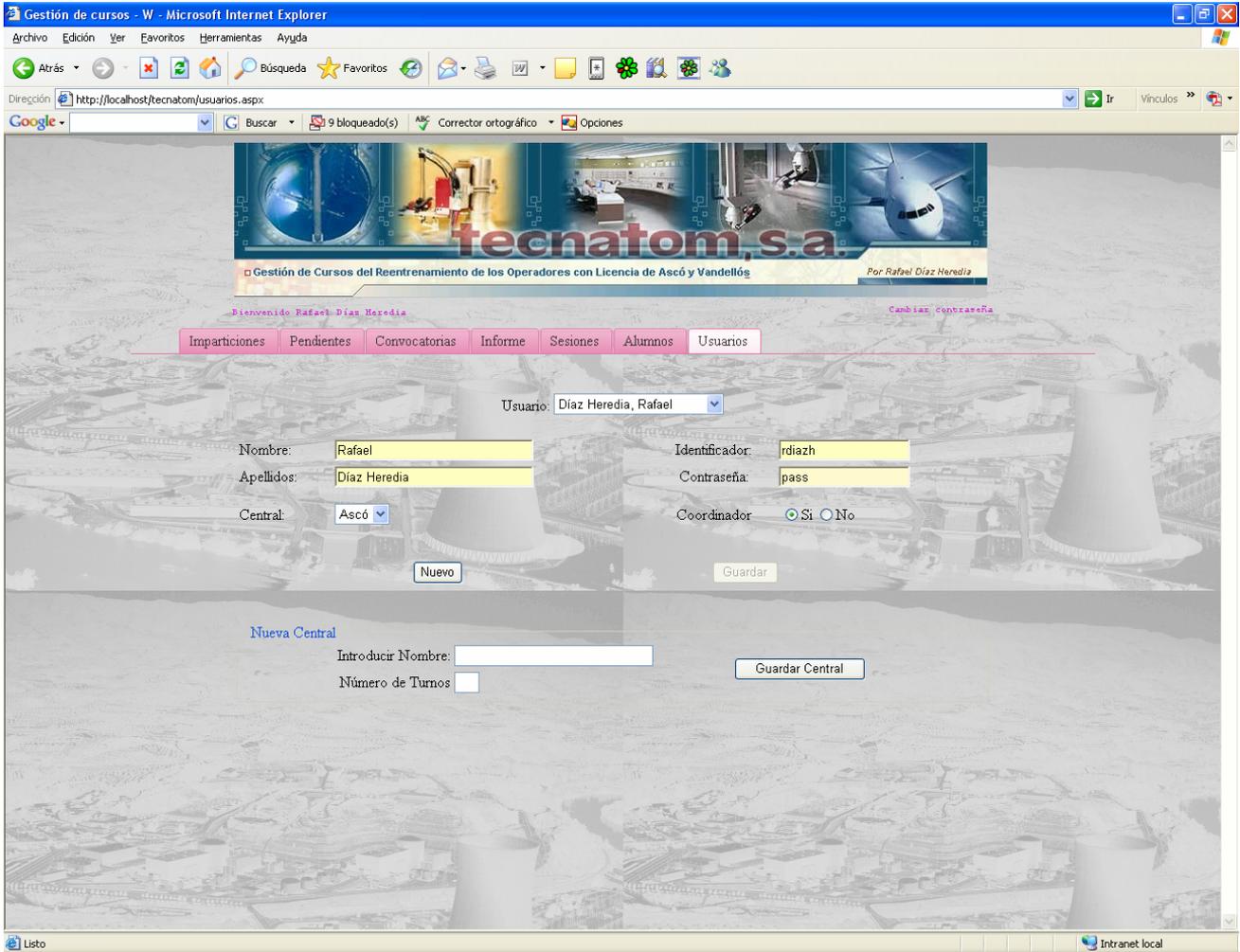


Figura 28: Pantalla Usuarios

4. Líneas de desarrollo futuro y posibles mejoras

La aplicación ha nacido para hacer frente a un problema muy concreto dentro de la Escuela de Formación, como son los cursos al personal con Licencia de las Centrales Nucleares.

Como aplicación dentro de un TFC de la carrera de Ingeniería Técnica de Sistemas, el proyecto tenía un límite temporal y no podía demorarse ni ser excesivamente grande.

Por tanto, existen bastantes mejoras y ampliaciones para que la aplicación sea una herramienta de gestión mucho más amplia dentro de la Escuela de Formación.

Estas ampliaciones, o futuras líneas y mejoras podrían ser:

- Mejorar el apartado de convocatorias: Se podría introducir el calendario oficial de los turnos para poder realizar las convocatorias en fechas en los que los alumnos están según cuadrante en formación o retén. Ahora podría ser que el instructor se equivocara al introducir una fecha y al alumno le llegue la convocatoria en un día al que no puede asistir por estar a turno.
- Mejorar el aspecto del informe final: Ahora se ha optado por dejar el informe final en *Excel* y que el cliente no note diferencia alguna si se utiliza esta aplicación como si no. En un futuro es posible hacer informes más consistentes utilizando HTML o incluso un *PDF*, con la posibilidad de introducir gráficas de asistencia a las convocatorias.
- Introducir un apartado para el personal sin Licencia: Al igual que el personal con licencia, la Escuela de Formación también ofrece cursos a otros puestos de las centrales. Se podría gestionar un apartado para ver que cursos se imparten y cual son los alumnos, aunque no se necesite un control tan riguroso.
- Introducir comentarios: Dejar la posibilidad de que un instructor añada comentarios en las jornadas de impartición. Estos comentarios podrían aparecer en el informe final, en la columna comentarios, junto con los temas pendientes de cada alumno.
- Mejora de la generación del formato oficial de firmas, tanto el del grupo como el individual: Ahora se genera utilizando una hoja *Excel* existente en el servidor y rellenándola utilizando llamadas *ADO.NET*. Es posible mejorar esta rutina generando dinámicamente la hoja sin necesidad de tener una copia en el servidor, como se hace en la generación del informe final.
- Mejoras en la seguridad vistas en el apartado 2: Utilización de SSL, autenticación por formulario, etc.

5. Conclusiones

El objetivo final era poner en práctica la mayor parte de conocimientos adquiridos durante los estudios. Para la realización de este proyecto, se ha utilizado especialmente conceptos utilizados en las áreas de:

- Ingeniería del software
- Fundamentos de programación
- Programación orientada a objetos
- Bases de datos
- Técnicas de desarrollo del software

El producto final es una aplicación que, con unos pocos retoques, será utilizable por una empresa, por lo que, a nivel personal, creo que este trabajo me ha aportado, a parte de los conocimientos y de la oportunidad de implementar un proyecto real, una herramienta útil en mi trabajo.

Las herramientas utilizadas para la realización de este proyecto son tan de actualidad que, en algunos casos como el *Visual Studio 2005*, existe poca información publicada. Agradecer a la *Universitat Oberta de Catalunya* el mérito de poder facilitar esas herramientas.

Y agradecer, finalmente, al consultor Jordi Ceballos por la ayuda y ánimos ofrecidos en los momentos malos, que los ha habido.

6. Bibliografía y referencias de consulta

A continuación se muestran tanto la documentación utilizada como consulta, como las direcciones de Internet visitadas y utilizadas de un modo u otro.

[1] Jesse Liberty y Dan Hurwitz; **“Programming ASP.NET”**, O’Reilly 2005, ISBN: 0-596-00916-X

[2] Stephen Walther; **“ASP.NET Al descubierto”**, Prentice Hall 2002, ISBN: 84-205-3472-2

[3] J. Ferguson, B. Patterson, J. Beres, P. Boutquin, M. Gupta; **“La biblia de C#”**, Anaya 2003, ISBN 84-415-1484-4

[4] Andrew Watt, **“Microsoft SQL Server 2005 for Dummies”**, Wiley Publishing 2006, ISBN: 0-7645-7755-7

[5] Programación Fácil: Tutorial C#
<<http://www.programacionfacil.com/csharpnet/indice.htm>>

[6] Belén Albeza, **“XHTML+CSS de una maldita vez!”**, 2004
<http://www.bergantells.net/~jander/weblog/tuto-xhtml_a3.pdf>

[7] Víctor Cuervo, Aulambra.com, **“Eliminar el subrayado de los enlaces”**
<<http://www.aulambra.com/ver.asp?id=77>>

[8] Web Taller: Manual de CSS
<<http://www.webtaller.com/manual-css/imprimible/tag-link.php>>

[9] w3.org: Comenzando con HTML+CSS
<<http://www.w3.org/style/examples/011/firstcss.es.html>>

[10] sidar.org: Manual de CSS
<<http://www.sidar.org/recur/desdi/mcss/manual/indice.php>>

[11] F. Berzal y F.J. Cortijo; **“Curso de C#”**
<<http://elvex.ugr.es/decsai/csharp/>>

[12] gotdotnet.com: **Server-Side Data Acces**
<<http://samples.gotdotnet.com/quickstart/aspplus/doc/webdataaccess.aspx>>

[13] Sergio José Villaneda Ávila, **“Operaciones básicas con bases de datos en ASP.NET (INSERT, UPDATE, SELECT, DELETE)”**
<http://www.netveloper.com/contenido2.aspx?IDC=275_0>

- [14] gotdotnet.com: **ADO.NET: Utilizar transacciones de base de datos**
<<http://es.gotdotnet.com/quickstart/howto/doc/adoplus/DatabaseTrans.aspx>>
- [15] Lee Gunn, **“Simple MessageBox functionality in ASP.NET”**
<<http://www.codeproject.com/aspnet/AspNetMsgBox.asp>>
- [16] Colt Kwong, **“PopUp Calendar”**
<<http://www.123aspx.com/redir.aspx?res=31081>>
- [17] Ryan Farley, **“Set Focus to an ASP.NET Control”**
<<http://ryanfarley.com/blog/archive/2004/12/21/1325.aspx>>
- [18] Ning Liao, Liang Yang , **“A Simple ASP.NET Server Control: Message Box & Confirmation Box”**
<<http://www.codeproject.com/aspnet/NingLiangSimpleControl.asp>>
- [19] Mankayarkarasi, **“Enhanced List Box Control”**
<<http://www.codeproject.com/aspnet/MCEnhancedListBoxControl.asp>>
- [20] Guillermo Som, **“Acceder a datos con ADO.NET”**
<http://www.netveloper.com/imprimir.aspx?IDC=181_0>
- [21] Jorge Ojeda García, **“Procesos Excel en ASP.NET”**
<http://www.elguille.info/colabora/NET2006/joga_ProcesosExcelASPNET.htm>
- [22] gotdotnet.com: **“Utilizar el archivo global.asax”**
<<http://es.gotdotnet.com/quickstart/aspplus/doc/globalasax.aspx#staticobjects>>
- [23] Gabriel Rodríguez Plancarte, **“Redireccionamiento en Session_End”**
<http://www.elguille.info/colabora/NET2005/GRP_Redireccion_en_Session_End.htm>
- [24] Microsoft.com: **“Cómo automatizar Microsoft Excel de Microsoft Visual C # .NET”**
<<http://support.microsoft.com/?scid=kb;es;302084>>
- [25] Victor H. Passador y Carlos A. Walzer, **“Accediendo a datos en Excel utilizando ADO.NET y ASP.NET”**
<<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art05.asp>>
- [26] mvps.org, **“Color Palette and the 56 Excel ColorIndex Colors”**
< <http://www.mvps.org/dmccritchie/excel/colors.htm>>
- [27] Microsoft.com: **“Cómo utilizar la Biblioteca de Objeto de Microsoft Outlook para enviar un mensaje que tiene datos adjuntos al utilizar Visual C”**
<<http://support.microsoft.com/?scid=kb;es;310263>>

[28] Guía de Seguridad del Consejo de Seguridad Nuclear 1.01, “**Cualificaciones para la obtención y uso de licencias de personal de operación en centrales nucleares**”

<http://www.csn.es/plantillas/frame_publicaciones.jsp?id_nodo=280&&tipo=GSG&keyword=&auditoria=F>