

TFC J2EE: Aplicación Web de Gestión de Reservas de Pistas Deportivas

Memoria

Sergio Ruiz Japón
ITIG

Consultor: Antoni Oller Arcas

 **UOC** Universitat Oberta
de Catalunya

Junio 2013

A mi mujer, Marta, por su paciencia y comprensión, y por siempre animarme a conseguir este objetivo.

Resumen

Para la realización de este Trabajo Fin de Carrera (TFC) se ha realizado el análisis, diseño e implementación de una aplicación utilizando tecnología Java y una arquitectura J2EE. El presente documento contiene la memoria del trabajo realizado durante el desarrollo de una Aplicación Web de Gestión de Reservas de Pistas Deportivas.

El proyecto de Gestión de Reservas de Pistas Deportivas consiste en una aplicación accesible a través de Internet, en la que los usuarios pueden consultar la disponibilidad de las diferentes pistas de las que consta el centro deportivo, reservar dichas pistas, etc. Además, ofrece las opciones típicas de administración, para que los gestores del centro deportivo puedan añadir, modificar o eliminar, pistas, tarifas, usuarios, etc.

Para llevar a cabo este desarrollo se han utilizado los conocimientos adquiridos a lo largo de la carrera de Ingeniería Técnica de Informática de Gestión en programación, análisis orientado a objetos y diseño de bases de datos. A su vez, ha sido necesario el estudio de la tecnología J2EE y de los diversos frameworks y componentes disponibles en el mercado.

Área del TFC: J2EE

Palabras clave: Java, J2EE, JBoss, Seam, JavaServer Faces, Hibernate, Patrones de diseño.

Licencia: Este trabajo está sujeto – salvo que se indique lo contrario – a una licencia Creative Commons by-nc-nd 3.0 España. Reconocimiento – NoComercial – SinObraDerivada: No se permite un uso comercial de la obra original ni la generación de obras derivadas. Podéis copiarlo, distribuirlo y transmitirlo públicamente siempre que citéis el autor y la obra, no se haga un uso comercial y no se haga copia derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.es>.

Índice

Resumen	3
1. Introducción	5
1.1 Justificación del TFC y contexto en el que se desarrolla.....	5
1.2 Objetivos del TFC.....	6
1.3 Enfoque y método seguido	6
1.4 Planificación	7
1.4.1 Calendario	7
1.4.2 Planificación Temporal (Diagrama de Gantt)	9
1.5 Productos obtenidos.....	10
1.6 Descripción de los capítulos de la memoria.....	10
2. Descripción del proyecto	11
2.1 Módulos de la aplicación	12
2.1.1 Módulo Gestión de Usuarios.....	12
2.1.2 Módulo Gestión de Tipos de Pistas	12
2.1.3 Módulo Gestión de Pistas	12
2.1.4 Módulo Gestión de Tarifas.....	12
2.1.5 Módulo de Gestión de Reservas.....	12
2.1.6 Módulo de Consulta de Pistas Disponibles / Reservar Pista.....	12
3. Análisis.....	13
3.1 Actores del sistema	13
3.2 Casos de uso	13
3.2.1 Casos de uso del Invitado.....	13
3.2.2 Casos de uso del Usuario registrado	14
3.2.3 Casos de uso del Administrador	14
3.3 Descripción de los casos de uso	15
4. Diseño.....	22
4.1 Diagrama de clases.....	22
4.2 Diseño de la base de datos	22
4.2.1 Modelo Entidad-Relación.....	22
4.2.2 Modelo lógico de la base de datos.....	23
4.2.3 Arquitectura de la aplicación.....	24
4.2.4 Entorno tecnológico	25
5. Implementación.....	26
5.1 Script de creación de la base de datos	26
6. Conclusiones.....	33
Glosario	34
Bibliografía.....	35
Recursos on-line	35
Anexos.....	36
Manual Básico de Usuario.....	36

1. Introducción

1.1 Justificación del TFC y contexto en el que se desarrolla

El Trabajo de Fin de Carrera está dirigido a poner en práctica los conocimientos adquiridos durante el estudio de las diferentes asignaturas que se han cursado a lo largo de la titulación, de forma conjunta en un solo proyecto. Esto supone un trabajo eminentemente práctico y bastante más cercano que el resto de asignaturas cursadas a lo que el futuro titulado encontrará en el desarrollo de su carrera profesional.

A pesar de no tener ninguna experiencia previa en el desarrollo de aplicaciones web, puesto que siempre he trabajado con Bases de Datos, elegí este área de trabajo debido a la gran demanda de profesionales vinculados al desarrollo web que hay actualmente, aún en estos tiempos de crisis. Considero que es un importante área de conocimiento para cualquier profesional del sector y es por ello que decidí adentrarme en el aprendizaje de la tecnología J2EE.

Para elegir la temática de la aplicación, después de pensar en infinidad de ideas, decidí realizar una aplicación web de Gestión de Reservas de Pistas Deportivas por la gran demanda que en los últimos tiempos tienen deportes como el pádel, por lo que se hace bastante tedioso el encontrar pistas disponibles y el tener que estar llamando a diferentes centros deportivos en busca de una pista libre. Si resulta un proceso laborioso para los usuarios, imagino que aún peor debe ser para los trabajadores del centro deportivo que deben atender infinidad de llamadas que finalmente no se concretarán en una reserva puesto que no quedan pistas libres a la hora que el usuario deseaba.

En resumen, con esta aplicación web buscamos tanto facilitar el acceso de los usuarios del centro deportivo a las pistas del mismo, como descongestionar el trabajo del personal del centro deportivo, que ahora podrá dedicar su tiempo a otras tareas.

1.2 Objetivos del TFC

El objetivo final del TFC es profundizar en los conocimientos de la tecnología J2EE, mediante el desarrollo de una aplicación web. Con este objetivo trabajaremos en:

- Análisis, diseño, implementación y pruebas, puesta en producción, y la generación de la documentación asociada: esta memoria y la presentación virtual.
- Seguimiento del patrón de arquitectura de software Modelo Vista Controlador, separando las capas de los datos y la lógica del negocio de la aplicación de la interfaz de usuario y el módulo que gestiona los eventos. Gracias a este patrón conseguiremos una aplicación más mantenible y un código más reutilizable, debido a la separación de conceptos.
- Para poder utilizar el Modelo Vista Controlador del punto anterior necesitaremos el uso de frameworks y la comunicación de estos con bases de datos relacionales, para la persistencia de los datos.
- Uso de patrones de diseño aplicables a la arquitectura J2EE.

Finalmente, al ser este trabajo una recopilación de los conocimientos adquiridos en el resto de asignaturas de la titulación, deberemos seguir las metodologías de desarrollo y programación previamente estudiadas en otras asignaturas. Una vez finalizado este Trabajo Final de Carrera, la aplicación generada debe ser completamente funcional y realizar todas las funciones que se le presuponían inicialmente.

1.3 Enfoque y método seguido

Para el desarrollo del proyecto se ha seguido un enfoque metodológico similar al que se lleva a cabo en entornos profesionales, donde se siguen una serie de fases de forma ordenada, que permiten minimizar los riesgos del proyecto y aumentar las garantías de éxito del mismo.

Para ello hemos seguido un ciclo de vida clásico o en cascada, en el que se han previsto y realizado las siguientes etapas:

- **Definición funcional:** consiste en definir qué queremos que haga nuestra aplicación.
- **Planificación:** se trata de dimensionar y planificar de forma realista cuanto tiempo nos va a llevar cada una de las fases para conseguir hacer las tareas definidas en el punto anterior.

- **Análisis:** consiste en analizar funcionalmente el problema que resuelve el proyecto, identificar los casos de uso y dar la solución funcional. Como resultado se obtiene el diagrama de casos de uso, la descripción textual de casos de uso y un plan de pruebas que al finalizar la implementación habrán de garantizar que la aplicación funciona correctamente.
- **Diseño:** se define cómo se implementará el proyecto a nivel técnico, la arquitectura y el modelo de implementación de la base de datos. Para ello, se analizan diferentes opciones: frameworks, servidores de aplicaciones, bases de datos, etc. El resultado es de identificación de las clases principales, el diseño de la base de datos y los diagramas de secuencia y de actividad de los casos de uso principales.
- **Implementación:** en función del diseño anterior, se codifica y construye cada una de las capas necesarias para el funcionamiento de la aplicación.
- **Pruebas:** se ejecuta el plan de pruebas definido en la etapa de análisis. Se verifica que el resultado es el esperado, tanto en los casos en que se espera conseguir un error como que funcione correctamente. Se corrigen las incidencias que se identifiquen (cuando el resultado obtenido no es el esperado en el plan de pruebas).
- **Documentación:** se genera toda la documentación vinculada a la aplicación para que cualquier persona pueda conocer qué hace y cómo lo hace.

1.4 Planificación

1.4.1 Calendario

La planificación del proyecto se adapta al calendario establecido por la UOC para la entrega del trabajo y está dividido en diferentes etapas, siguiendo la metodología definida anteriormente.

Se definen en la planificación las siguientes actividades organizadas en 4 entregas:

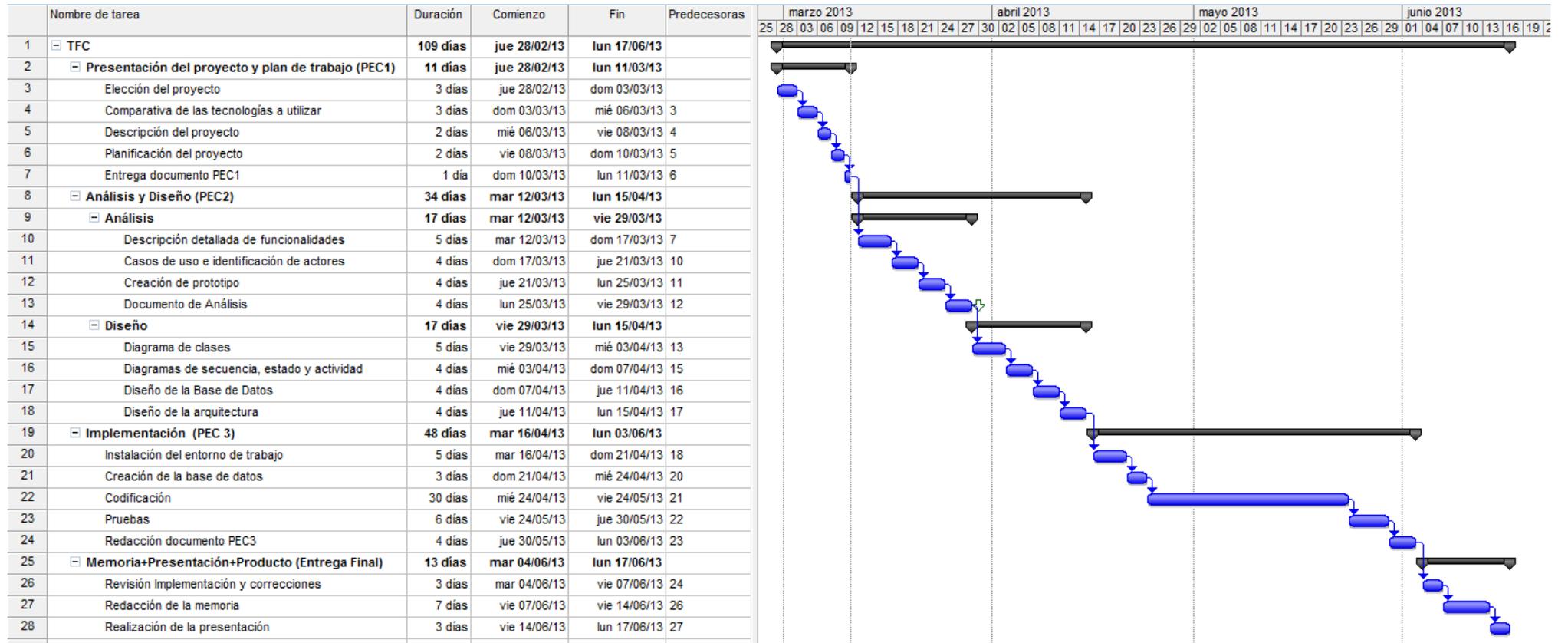
- **PEC1:** Presentación del proyecto y plan de trabajo.
- **PEC2:** Análisis y diseño.
- **PEC3:** Implementación.
- **PEC4:** Entrega final.

Teniendo en cuenta la descripción de nuestro proyecto y los objetivos que deseamos conseguir, procedemos a establecer una planificación que nos permita realizar con éxito nuestro TFC.

En la tabla siguiente podemos ver la planificación de las distintas etapas del proyecto, y a continuación, el diagrama de Gantt correspondiente.

Tarea	Tiempo Estimado	Fecha Inicio	Fecha Fin
1.- Presentación del proyecto y plan de trabajo (PEC1)	11 días	28/02/2013	11/03/2013
Elección del proyecto	3 días	28/02/2013	03/03/2013
Comparativa de las tecnologías a utilizar	3 días	03/03/2013	06/03/2013
Descripción del proyecto	2 días	06/03/2013	08/03/2013
Planificación del proyecto	2 días	08/03/2013	10/03/2013
Entrega documento PEC1	1 día	10/03/2013	11/03/2013
2.- Análisis y Diseño (PEC2)	34 días	12/03/2013	15/04/2013
Análisis	17 días	12/03/2013	29/03/2013
Descripción detallada de funcionalidades	5 días	12/03/2013	17/03/2013
Casos de uso e identificación de actores	4 días	17/03/2013	21/03/2013
Creación de prototipo	4 días	21/03/2013	25/03/2013
Documento de Análisis	4 días	25/03/2013	29/03/2013
Diseño	17 días	29/03/2013	15/04/2013
Diagrama de clases	5 días	29/03/2013	03/04/2013
Diagramas de secuencia, estado y actividad	4 días	03/04/2013	07/04/2013
Diseño de la Base de Datos	4 días	07/04/2013	11/04/2013
Diseño de la arquitectura	4 días	11/04/2013	15/04/2013
3.- Implementación (PEC 3)	48 días	16/04/2013	03/06/2013
Instalación del entorno de trabajo	5 días	16/04/2013	21/04/2013
Creación de la base de datos	3 días	21/04/2013	24/04/2013
Codificación	30 días	24/04/2013	24/05/2013
Pruebas	6 días	24/05/2013	30/05/2013
Redacción documento PEC3	4 días	30/05/2013	03/06/2013
4.- Memoria + Presentación + Producto (Entrega Final)	13 días	04/06/2013	17/06/2013
Revisión Implementación y correcciones	3 días	04/06/2013	07/06/2013
Redacción de la memoria	7 días	07/06/2013	14/06/2013
Realización de la presentación	3 días	14/06/2013	17/06/2013

1.4.2 Planificación Temporal (Diagrama de Gantt)



1.5 Productos obtenidos

Los productos obtenidos en la realización de este Trabajo Final de Carrera son los que se detallan a continuación:

- El producto más importante es la aplicación web desarrollada, que permite la gestión de reservas de pistas deportivas.
- La presente memoria del proyecto, que explica el proyecto en detalle: objetivos, planificación inicial y las diferentes etapas de la metodología seguida para obtener el resultado final.
- Presentación virtual que sintetiza el trabajo realizado en el proyecto y los resultados obtenidos.

1.6 Descripción de los capítulos de la memoria

En el resto de capítulos de la memoria se describe ampliamente el proyecto. Está organizada en los siguientes capítulos:

- **Descripción del proyecto:** especifica y delimita el alcance del proyecto desde un punto de vista funcional, es decir, qué hará y qué no hará nuestra aplicación.
- **Análisis:** Descripción del funcionamiento de los casos de uso identificados, así como los diagramas de estado, de secuencia y de actividades de los casos de uso más relevantes.
- **Diseño:** Definición del modelo de entidades y del diseño de la base de datos que permita la persistencia de los datos necesarios. Diseño de la arquitectura tecnológica utilizada en el proyecto que permitirá la implementación de los casos de uso y facilite un ciclo de vida del software equilibrado mediante una separación de capas bien definida.
- **Implementación:** se explican las decisiones de diseño e implementación tomadas durante el desarrollo. Así mismo, se indican los requerimientos de software y los pasos a realizar para una correcta configuración y ejecución de la aplicación.
- **Conclusiones:** se exponen las principales conclusiones obtenidas una vez finalizado el proyecto.

2. Descripción del proyecto

El presente proyecto consiste en el desarrollo de una aplicación web para la gestión de las reservas de las pistas de cualquier complejo polideportivo. El objetivo es facilitar el disfrute de las pistas deportivas por parte de los usuarios, además de optimizar la gestión para el máximo aprovechamiento de las instalaciones.

Los usuarios de la web, una vez identificados en el sistema, podrán comprobar la disponibilidad de las diferentes pistas del complejo deportivo, y si estuvieran interesados podrán realizar la reserva de una pista disponible, que a partir de ese momento aparecerá como reservada, y por lo tanto, no disponible para el resto de usuarios. Será necesario que los usuarios estén identificados en el sistema a la hora de realizar una reserva para así evitar conflictos con reservas en falso o la incomparecencia de los jugadores a la hora reservada, en cuyo caso se podrá cobrar el alquiler de la pista al socio que hizo la reserva.

Además, el usuario podrá ver todos los datos de las reservas que tiene vigentes y del histórico de las reservas que ha realizado.

El administrador del sistema podrá añadir nuevas instalaciones, establecer los horarios y precios para disponer de las pistas o de los complementos (luz artificial, etc).

Debido al espectacular crecimiento en los últimos tiempos del número de usuarios de pistas deportivas, especialmente de pádel, y a la dificultad para poder reservar una pista, creemos que el proyecto nos ofrece una gran variedad de posibilidades para poder reservar una pista sin tener que realizar varias llamadas telefónicas, y en las que en muchos casos no se logra conseguir el objetivo.

2.1 Módulos de la aplicación

2.1.1 Módulo Gestión de Usuarios

Mediante este módulo el administrador del sistema podrá gestionar todo lo relacionado con los usuarios del sistema: altas, bajas, modificaciones, reseteo de contraseñas, etc.

2.1.2 Módulo Gestión de Tipos de Pistas

Usando este módulo el administrador gestionará los distintos tipos de pista disponibles en la instalación deportiva.

2.1.3 Módulo Gestión de Pistas

A través de este módulo el administrador del sistema gestionará las pistas del complejo deportivo. Podrá añadir realizar la gestión completa (altas, bajas y modificaciones) de las pistas del sistema.

2.1.4 Módulo Gestión de Tarifas

En este módulo el administrador fija los horarios en que estarán disponibles y los precios para cada tipo de pista.

2.1.5 Módulo de Gestión de Reservas

Usando este módulo el usuario podrá consultar, modificar o anular sus reservas. El administrador del sistema podrá hacer todas estas operaciones con el añadido de que podrá gestionar las reservas de cualquier usuario.

2.1.6 Módulo de Consulta de Pistas Disponibles / Reservar Pista

Este módulo permitirá consultar la disponibilidad de pistas y un hacer la reserva de una pista que esté libre.

3. Análisis

En este apartado se describe el Sistema de Gestión de Reservas de Pistas Deportivas, se identificarán y describirán los diferentes actores que intervienen en el mismo, se mostrará el diagrama de casos de uso y se realizará una descripción detallada de cada caso de uso.

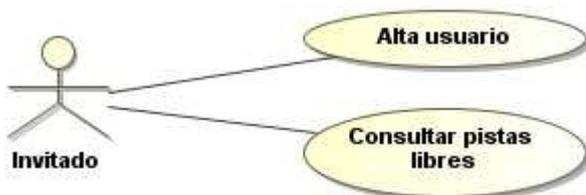
3.1 Actores del sistema

Después de analizar el sistema concluimos que intervienen los siguientes actores:

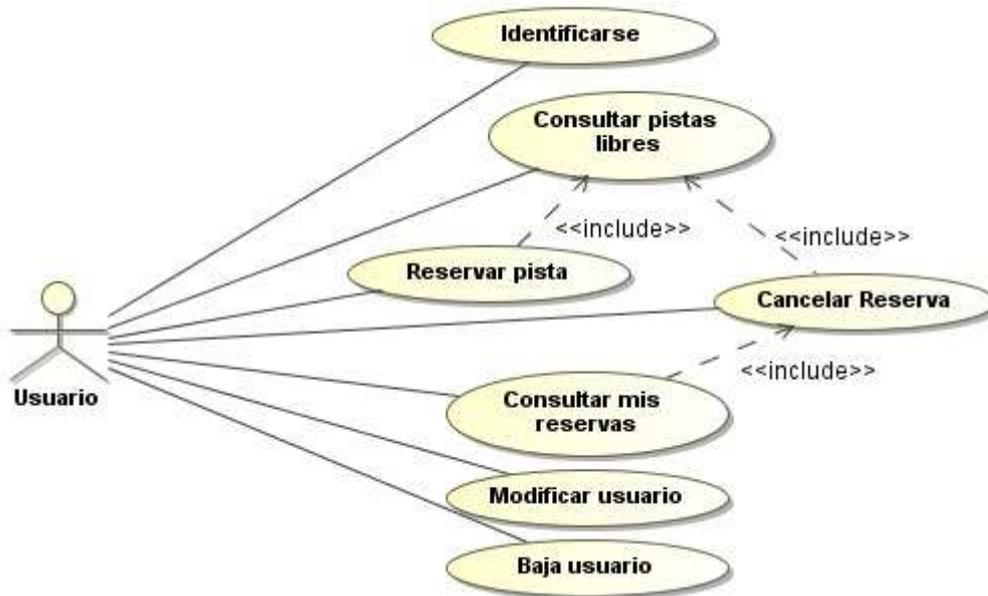
- **Invitados:** Son aquellos usuarios del sistema que acceden a este para consultar la disponibilidad de pistas libres sin necesidad de identificarse. Sólo pueden hacer esta consulta o registrarse como usuarios del sistema.
- **Usuarios registrados:** son los usuarios que utilizarán el sistema para realizar las reservas de las pistas deportivas, por lo tanto, serán mayoría frente al resto de usuarios. Para poder realizar reservas deben registrarse e identificarse adecuadamente en el sistema.
- **Administrador:** es el usuario responsable del correcto funcionamiento del sistema, además de ocuparse de gestionar las pistas disponibles, fechas y horas en que se podrán alquilar las pistas, el precio de las mismas, etc.

3.2 Casos de uso

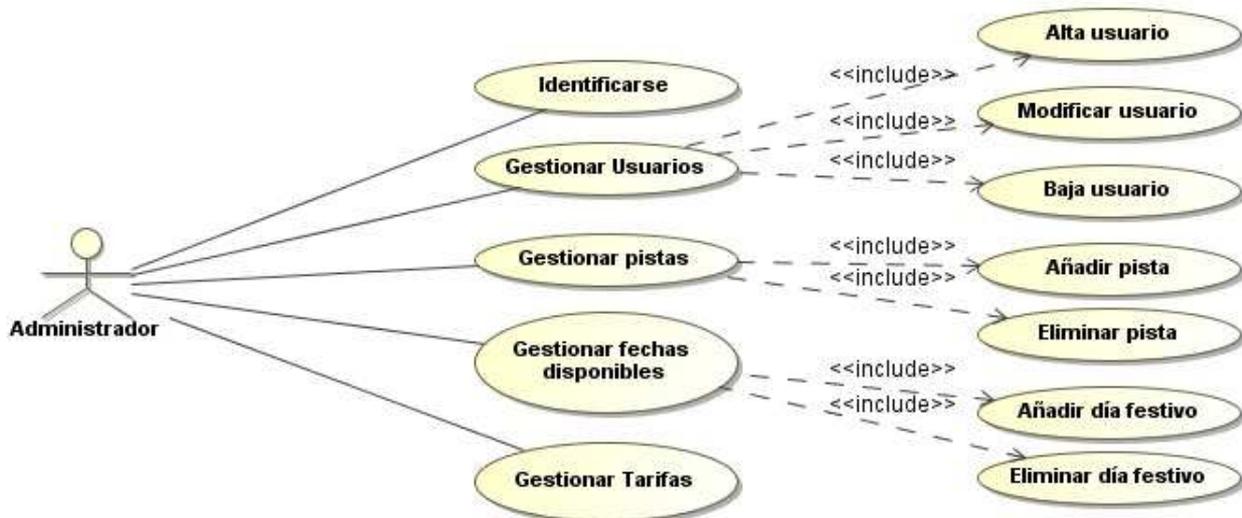
3.2.1 Casos de uso del Invitado



3.2.2 Casos de uso del Usuario registrado



3.2.3 Casos de uso del Administrador



3.3 Descripción de los casos de uso

Caso de uso	Alta usuario
Resumen de la funcionalidad:	Añade un nuevo usuario registrado al sistema.
Papel dentro del trabajo del usuario:	Poco frecuente.
Actores:	Invitado, Administrador.
Casos de uso relacionados:	Identificarse
Precondición:	No existe el usuario.
Postcondición:	El usuario ha quedado registrado en el sistema, por lo que hay un nuevo usuario.
Flujo normal:	<ol style="list-style-type: none"> 1.- El usuario (invitado o administrador) accede al sistema y a la página de registro. 2.- Introduce los datos de la persona a registrar. 3.- Envía los datos. 4.- Se crea el usuario.
Flujo alternativo:	<ol style="list-style-type: none"> 2.- El invitado (o el administrador) sale de la página de registro sin completar el proceso. 3.- El invitado (o el administrador) pulsa el botón Cancelar. 3.- El nombre de usuario o el correo electrónico ya existen en el sistema. 3.- Los datos introducidos contienen algún error o falta algún dato.
Observaciones:	Todos los usuarios registrados en el sistema tendrán que utilizar este proceso, aunque sólo una vez.

Caso de uso	Consultar pistas libres
Resumen de la funcionalidad:	Muestra el estado de ocupación de las pistas para una fecha determinada.
Papel dentro del trabajo del usuario:	Muy frecuente.
Actores:	Invitado, Usuario registrado, Administrador.
Casos de uso relacionados:	Reservar pista, Cancelar reserva
Precondición:	Ninguna, al ser una consulta.
Postcondición:	Ninguna, al ser una consulta.
Flujo normal:	<ol style="list-style-type: none"> 1.- El usuario accede a la aplicación.

	<p>2.- Selecciona el tipo de pista que quiere consultar.</p> <p>3.- Selecciona la fecha que quiere consultar.</p> <p>4.- Se muestra el estado en que se encuentran todas las pistas del tipo elegido para la fecha indicada.</p>
Flujo alternativo:	
Observaciones:	<p>Será una de las funcionalidades más utilizadas de la aplicación, puesto que para realizar una reserva, todos los usuarios consultarán previamente el estado de las pistas.</p>

Caso de uso	Identificarse
Resumen de la funcionalidad:	El usuario introduce su nombre de usuario y contraseña para acceder al sistema con los privilegios de su tipo de usuario.
Papel dentro del trabajo del usuario:	Muy frecuente.
Actores:	Usuario registrado, Administrador.
Casos de uso relacionados:	Alta usuario
Precondición:	El usuario existe en el sistema.
Postcondición:	El usuario se identifica en el sistema correctamente.
Flujo normal:	<p>1.- El usuario (usuario registrado o administrador) accede al sistema.</p> <p>2.- El usuario va a la página de login.</p> <p>3.- Introduce su nombre de usuario y contraseña.</p> <p>4.- Pulsa el botón Enviar.</p> <p>5.- El sistema reconoce al usuario mostrándole un mensaje de bienvenida.</p>
Flujo alternativo:	<p>3.- El usuario sale de la página de login sin identificarse.</p> <p>4.- El usuario pulsa el botón Cancelar.</p> <p>5.- El nombre de usuario no existe o bien la contraseña no coincide, por lo que el sistema muestra un mensaje de error.</p>
Observaciones:	También será una funcionalidad muy utilizada puesto que para poder realizar una reserva es necesario que el usuario se identifique previamente.

Caso de uso	Reservar pista
Resumen de la funcionalidad:	El usuario hace una reserva de una pista para una fecha y hora concreta.
Papel dentro del trabajo del usuario:	Muy frecuente.
Actores:	Usuario registrado, Administrador.
Casos de uso relacionados:	Identificarse, Consultar pistas libres
Precondición:	El usuario debe estar identificado en el sistema.
Postcondición:	El usuario ha realizado la reserva de una pista para una fecha y hora.
Flujo normal:	1.- El usuario consulta la disponibilidad de las pistas para una fecha determinada y pulsa el botón Reservar pista para una hora en concreto. 2.- El usuario realiza la reserva de la pista a su nombre, y la pista ya no estará disponible a partir de ahora.
Flujo alternativo:	2.- Si el usuario no estaba identificado en el sistema se le pide que se identifique o que se registre.
Observaciones:	Otra de las funcionalidades más utilizadas puesto que es el objetivo final de nuestra aplicación.

Caso de uso	Cancelar reserva
Resumen de la funcionalidad:	El usuario cancela la reserva de una pista que había realizado previamente.
Papel dentro del trabajo del usuario:	Poco frecuente.
Actores:	Usuario registrado, Administrador.
Casos de uso relacionados:	Identificarse, Consultar pistas libres, Realizar reserva
Precondición:	El usuario debe estar identificado en el sistema y haber realizado una reserva a su nombre.
Postcondición:	El usuario ha cancelado una reserva de una pista para una fecha y hora.
Flujo normal:	1.- El usuario puede cancelar la reserva bien desde la pantalla "Consultar mis reservas" o bien desde la pantalla "Consulta pistas libres" para una fecha determinada, donde le aparecerá el botón Cancelar reserva para las pistas que tenga reservada a su nombre. 2.- El usuario ha cancelado la reserva de la pista a su nombre, y la pista vuelve a estar libre a partir de ahora.
Flujo alternativo:	2.- Si el tiempo que falta para el inicio de la reserva es menor a un tiempo fijado como parámetro por el administrador, no se permitirá la cancelación de la reserva, por lo que el usuario nos adeudará el importe de la misma.
Observaciones:	

Caso de uso	Consultar mis reservas
Resumen de la funcionalidad:	El usuario consulta las reservas que ha realizado previamente.
Papel dentro del trabajo del usuario:	Poco frecuente.
Actores:	Usuario registrado, Administrador.
Casos de uso relacionados:	Realizar reserva
Precondición:	Ninguna, al ser una consulta.
Postcondición:	Ninguna, al ser una consulta.
Flujo normal:	1.- El usuario
Flujo alternativo:	
Observaciones:	

Caso de uso	Modificar usuario
Resumen de la funcionalidad:	El usuario modifica alguno de sus datos o su contraseña.
Papel dentro del trabajo del usuario:	Poco frecuente.
Actores:	Usuario registrado, Administrador.
Casos de uso relacionados:	Alta usuario
Precondición:	El usuario debe estar identificado en el sistema.
Postcondición:	El sistema modifica los datos del usuario en la base de datos.
Flujo normal:	<ol style="list-style-type: none"> 1.- El usuario accede a la aplicación. 2.- Pulsa el botón Modificar usuario. 3.- Se muestran los datos del usuario tal y como están en la base de datos. 4.- El usuario modifica alguno de los campos permitidos. 5.- El usuario pulsa el botón Guardar, para que se guarden los cambios introducidos.
Flujo alternativo:	<ol style="list-style-type: none"> 5.- El usuario sale sin pulsar el botón Guardar. 5.- El usuario pulsa el botón Cancelar.
Observaciones:	

Caso de uso	Baja usuario
Resumen de la funcionalidad:	El usuario se da de baja del sistema
Papel dentro del trabajo del usuario:	Poco frecuente.
Actores:	Usuario registrado, Administrador.
Casos de uso relacionados:	Alta usuario
Precondición:	El usuario debe estar identificado en el sistema.
Postcondición:	El sistema borra los datos del usuario de la base de datos.
Flujo normal:	<ol style="list-style-type: none"> 1.- El usuario accede a la aplicación. 2.- Pulsa el botón Baja usuario. 3.- Se muestra un mensaje de advertencia para que sepa que si borra su usuario es un cambio irreversible y no podrá volver a acceder al sistema. 4.- El usuario pulsa el botón Aceptar, para que se borren sus datos.
Flujo alternativo:	<ol style="list-style-type: none"> 4.- El usuario sale sin pulsar el botón Aceptar. 4.- El usuario pulsa el botón Cancelar.
Observaciones:	

Caso de uso	Gestionar usuarios
Resumen de la funcionalidad:	El administrador accede a un menú específico para gestionar usuarios.
Papel dentro del trabajo del usuario:	Poco frecuente.
Actores:	Administrador.
Casos de uso relacionados:	Alta usuario, Baja usuario, Modificar usuario
Precondición:	El administrador debe estar identificado en el sistema.
Postcondición:	El sistema muestra el menú de administración de usuario
Flujo normal:	<ol style="list-style-type: none"> 1.- El administrador accede a la aplicación. 2.- Pulsa el botón Gestionar usuarios. 3.- Se muestran las diferentes opciones del menú.
Flujo alternativo:	
Observaciones:	El administrador podrá dar de alta nuevos usuarios, modificar los datos de los usuarios existentes, por ejemplo, una contraseña bloqueada, y eliminar usuarios que notifiquen que no quieren continuar en el sistema.

Caso de uso	Gestionar pistas
Resumen de la funcionalidad:	El administrador accede a un menú específico para gestionar pistas.
Papel dentro del trabajo del usuario:	Poco frecuente.
Actores:	Administrador.
Casos de uso relacionados:	Añadir pista, Eliminar pista
Precondición:	El administrador debe estar identificado en el sistema.
Postcondición:	El sistema muestra el menú de administración de pistas
Flujo normal:	<ol style="list-style-type: none"> 1.- El administrador accede a la aplicación. 2.- Pulsa el botón Gestionar pistas. 3.- Se muestran las diferentes opciones del menú.
Flujo alternativo:	
Observaciones:	El administrador podrá dar de alta nuevas pistas o eliminarlas porque ya no estén disponibles.

Caso de uso	Gestionar fechas disponibles
Resumen de la funcionalidad:	El administrador accede a un menú específico para gestionar las fechas en que las pistas estarán disponibles.
Papel dentro del trabajo del usuario:	Poco frecuente.
Actores:	Administrador.
Casos de uso relacionados:	Añadir día festivo, Eliminar día festivo
Precondición:	El administrador debe estar identificado en el sistema.
Postcondición:	El sistema muestra el menú de gestión de fechas disponibles.
Flujo normal:	<ol style="list-style-type: none"> 1.- El administrador accede a la aplicación. 2.- Pulsa el botón Gestionar fechas disponibles. 3.- Se muestran las diferentes opciones del menú.
Flujo alternativo:	
Observaciones:	El administrador podrá establecer los días de la semana en los que las pistas del club están disponibles, así como añadir días en los que el club permanecerá cerrado por cualquier motivo.

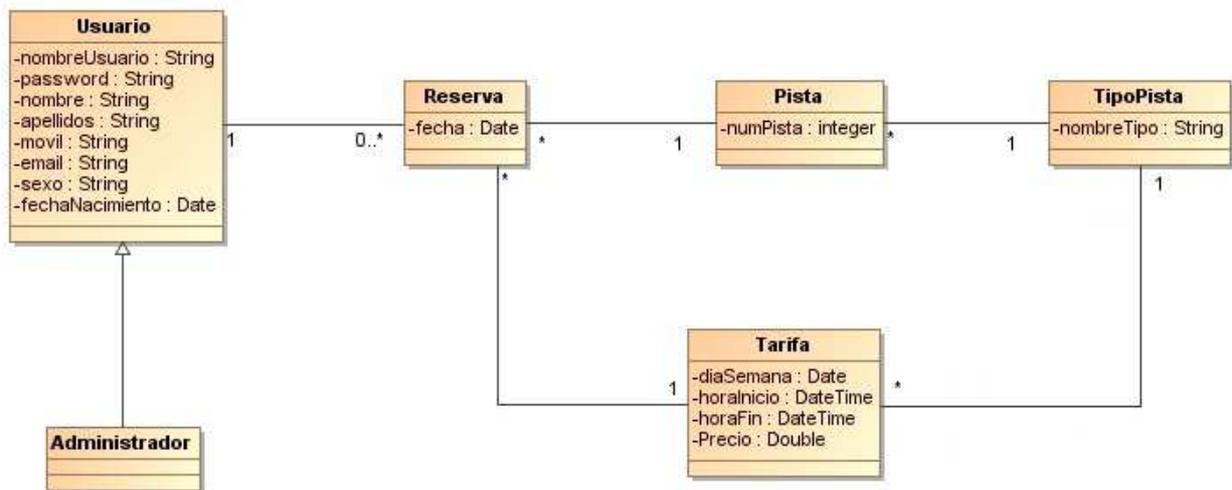
Caso de uso	Gestionar tarifas
Resumen de la funcionalidad:	El administrador accede a un menú específico para gestionar las horas en que las pistas estarán disponibles en los distintos días y las tarifas de alquiler de cada pista en cada horario.
Papel dentro del trabajo del usuario:	Poco frecuente.
Actores:	Administrador.
Casos de uso relacionados:	
Precondición:	El administrador debe estar identificado en el sistema.
Postcondición:	El sistema muestra el menú de gestión de horas disponibles.
Flujo normal:	<ol style="list-style-type: none"> 1.- El administrador accede a la aplicación. 2.- Pulsa el botón Gestionar tarifas. 3.- Se muestran las diferentes opciones del menú.
Flujo alternativo:	
Observaciones:	El administrador podrá establecer las horas en las que las pistas del club están disponibles en los distintos días de la semana y el precio de las mismas.

4. Diseño

Una vez finalizada la fase de análisis, comenzamos con la etapa de diseño de la nueva aplicación:

4.1 Diagrama de clases

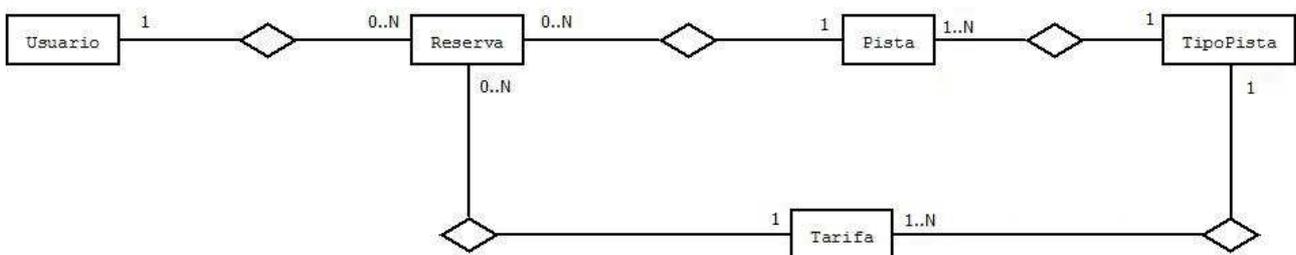
A continuación se presenta el diagrama de clases de entidad junto con sus atributos y las relaciones entre las clases:



4.2 Diseño de la base de datos

4.2.1 Modelo Entidad-Relación

Para almacenar toda la información utilizada en la aplicación vamos a utilizar una base de datos, almacenada en un SGBD, en nuestro caso Oracle. A continuación presentamos el diseño de la base de datos, empezando por su modelo entidad-relación (E-R):



4.2.2 Modelo lógico de la base de datos

USUARIO(id_usuario, nombre_usuario, password, nombre, apellidos, movil, email, sexo, fecha_nacimiento, es_administrador)

TIPO_PISTA(id_tipo_pista, nombre_tipo)

PISTA(id_pista, num_pista, id_tipo_pista)

donde {id_tipo_pista} referencia Tipo_pista

TARIFA(id_tarifa, hora_inicio, hora_fin, precio)

RESERVA(id_usuario, id_pista, id_tarifa, fecha)

donde {id_usuario} referencia Usuario

donde {id_pista} referencia Pista

donde {id_tarifa} referencia Tarifa

Consideraciones:

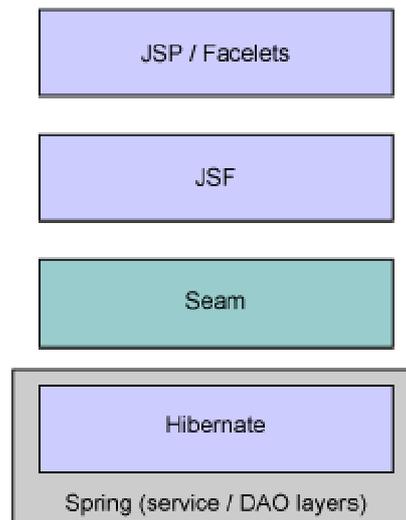
Se han utilizado claves primarias numéricas, a raíz de una secuencia por las siguientes razones:

- Independizar la clave de los datos, de forma que no haga falta cambiar la clave (tanto en la tabla principal, como en todas las relacionadas) en caso de que hubiera cualquier tipo de error en los datos. Esta clave no la utilizaremos para hacer búsquedas, pero sí que podemos usarla para comparar objetos de forma sencilla.
- Utilizamos la cláusula FOREIGN KEY para identificar las claves ajenas que hacen referencia a otras tablas.
- No he considerado necesario crear una tabla "Administradores", puesto que todos los atributos serían los de la tabla "Usuario", por lo que he aprovechado esta última para añadir un campo que diga si el usuario es también administrador.

4.2.3 Arquitectura de la aplicación

Para el desarrollo de nuestro proyecto utilizaremos la arquitectura J2EE, que nos brinda una arquitectura multicapa y distribuida, que nos permite acceder a nuestra aplicación a través de Internet. Esto es el patrón conocido como Modelo – Vista – Controlador o MVC.

Esta arquitectura de separación en capas conlleva grandes ventajas respecto a la programación en una sola capa, ya que nos permite una mejor estructuración y modularización de los programas. Eso supone que es mucho más sencillo la reutilización de código y el mantenimiento y sustitución de componentes más pequeños y específicos, ya que los cambios necesarios se localizan en una sola capa, de manera que no se ven afectadas el resto de capas.



- **Vista o capa de presentación:** es la interfaz gráfica a través de la cual los usuarios de nuestra aplicación interactúan con ella. Se ocupa de presentar y gestionar las pantallas que visualiza el usuario de la aplicación, recogiendo los eventos que se produzcan y las interacciones con el usuario. Un buen diseño de la capa de presentación ayudará mucho a la usabilidad y facilidad de uso de la aplicación, y por tanto es una parte esencial para el éxito del proyecto.
- **Controlador o capa de negocio:** es quien contiene las reglas de negocio del sistema. Se encarga de recibir los datos enviados desde la Vista, tratarlos, y enviar el resultado al modelo. A su vez puede volver a presentar una nueva vista con los resultados obtenidos. Digamos que hace de enlace entre la Vista y el Modelo.
- **Modelo o capa de persistencia:** se encarga de almacenar la información con la cual el sistema opera. Aquí se gestiona el flujo de información con las bases de datos, y se devuelven como se necesiten desde el programa principal.

4.2.4 Entorno tecnológico

Ya hemos dicho anteriormente, que para el desarrollo de nuestro proyecto utilizaremos la arquitectura J2EE, siguiendo el patrón de diseño de software del Modelo – Vista – Controlador o MVC. Para implementar físicamente este patrón hemos necesitado un entorno tecnológico, que seguidamente vamos a detallar:

- **IDE de desarrollo:** He utilizado JBoss Developer Studio Version: 3.0.1.GA, que es un entorno de desarrollo que integra Eclipse con infinidad de componentes y herramientas para el desarrollo Web. Entre ellas:
- **Framework de desarrollo:** JBoss Seam. Es un framework que integra JSF y EJB3. Tuvo bastante éxito hace unos años e incluso su filosofía se incluye en el estándar de JEE 6. Decidí utilizar este framework y no otro, debido a que es el utilizado en mi empresa, por lo que una vez finalizado este TFC, considero que estoy capacitado para asumir tareas de desarrollo en el mismo framework, que era mi idea original.
 - **Framework de persistencia de datos:** Hibernate Tools for Eclipse. Version: 3.3.1.
 - **Servidor de aplicaciones:** JBoss 5.1 Runtime Server
 - **Java Development Kit (JDK):** 1.6.0_23
 - **Conector de base de datos:** ojdbc5.jar
- **Servidor de Bases de Datos:** Oracle 10 XE. He elegido este servidor de bases de datos, aún a sabiendas de que para la explotación comercial de la aplicación es bastante costoso, pero podemos suponer que la entidad responsable del centro deportivo, probablemente un ayuntamiento, ya posee máquinas con bases de datos Oracle, y sus técnicos están familiarizados con el entorno, por lo que lo consideré una imposición del cliente. Por otra parte, dejando de lado el inconveniente económico, ofrece mayor seguridad y Oracle ofrece un soporte del que no dispondríamos en otros Sistemas Gestores de Bases de Datos gratuitos.
- **Entorno de administración de Bases de Datos:** He utilizado indistintamente Toad y sqldeveloper.
- **Hardware:** He utilizado mi portátil personal, un Dell, con Sistema Operativo Windows 7 Professional, Procesador Intel Core 2 Duo T8300 a 2.40 GHz y 3 GB de RAM, así como eventualmente un ordenador sobremesa.

5. Implementación

Lo primero a destacar en la fase de implementación es la herramienta seam-gen incluida en JBoss Seam, la cual, una vez configurada dónde se encuentra la base de datos con las tablas ya creadas, genera una estructura básica de mantenimiento de la aplicación para las tablas existentes en el esquema, lo cual facilita bastante el trabajo, ya que hace todo el mapeo de las tablas con todos sus campos, con sus respectivos métodos getters y setters.

Así mismo quiero destacar el uso de un Repositorio de Copias de Seguridad. Para almacenar copias de seguridad del proyecto, así como para poder cambiar del ordenador sobremesa al portátil y viceversa, disponiendo siempre de la última versión del código fuente generado, he utilizado un repositorio SVN gratuito, disponible en la web www.assembla.com. Así, de forma bastante sencilla, siempre tenía los datos actualizados, con sólo subir los cambios al repositorio al dejar un ordenador y descargar la última versión en el otro. Igualmente, nos sirve para tener un historial de versiones estables, a las que volver en caso de que haya algún problema.

5.1 Script de creación de la base de datos

```
--
-- TIPO_PISTA (Table)
--
CREATE TABLE TIPO_PISTA
(
  ID_TIPO_PISTA  NUMBER                NOT NULL,
  NOMBRE_TIPO    VARCHAR2(50 BYTE)     NOT NULL
);

--
-- PK_TIPO_PISTA (Index)
--
CREATE UNIQUE INDEX PK_TIPO_PISTA ON TIPO_PISTA
(ID_TIPO_PISTA);

--
-- Non Foreign Key Constraints for Table TIPO_PISTA
--
ALTER TABLE TIPO_PISTA ADD (
  CONSTRAINT PK_TIPO_PISTA PRIMARY KEY (ID_TIPO_PISTA)
  USING INDEX) ;

--
-- PISTA (Table)
--
CREATE TABLE PISTA
(
  ID_PISTA        NUMBER                NOT NULL,
  NUM_PISTA       NUMBER                NOT NULL,
  ID_TIPO_PISTA   NUMBER                NOT NULL
);
```

```

--
-- PK_PISTA (Index)
--
CREATE UNIQUE INDEX PK_PISTA ON PISTA
(ID_PISTA);

--
-- Non Foreign Key Constraints for Table PISTA
--
ALTER TABLE PISTA ADD (
  CONSTRAINT PK_PISTA PRIMARY KEY (ID_PISTA)
  USING INDEX );

--
-- Foreign Key Constraints for Table PISTA
--
ALTER TABLE PISTA ADD (
  CONSTRAINT FK_PISTA_TIPO_PISTA FOREIGN KEY (ID_TIPO_PISTA)
  REFERENCES TIPO_PISTA (ID_TIPO_PISTA));

--
-- USUARIO (Table)
--
CREATE TABLE USUARIO
(
  ID_USUARIO          NUMBER          NOT NULL,
  NOMBRE_USUARIO     VARCHAR2(30 BYTE) NOT NULL,
  PASSWORD            VARCHAR2(30 BYTE) NOT NULL,
  NOMBRE              VARCHAR2(50 BYTE) NOT NULL,
  APELLIDOS           VARCHAR2(80 BYTE) NOT NULL,
  MOVIL               VARCHAR2(12 BYTE) NOT NULL,
  EMAIL              VARCHAR2(25 BYTE) NOT NULL,
  SEXO                VARCHAR2(1 BYTE) NOT NULL,
  FECHA_NACIMIENTO   DATE             NOT NULL,
  ES_ADMINISTRADOR   VARCHAR2(1 BYTE) NOT NULL,
  FECHA_ALTA         DATE             NOT NULL
);

--
-- PK_USUARIO (Index)
--
CREATE UNIQUE INDEX PK_USUARIO ON USUARIO
(ID_USUARIO);

--
-- Non Foreign Key Constraints for Table USUARIO
--
ALTER TABLE USUARIO ADD (
  CONSTRAINT PK_USUARIO PRIMARY KEY (ID_USUARIO)
  USING INDEX );

--
-- TARIFA (Table)
--
CREATE TABLE TARIFA
(
  ID_TARIFA          NUMBER          NOT NULL,
  ID_TIPO_PISTA     NUMBER          NOT NULL,

```

```
HORA_INICIO    VARCHAR2(5 BYTE)          NOT NULL,
HORA_FIN       VARCHAR2(5 BYTE)        NOT NULL,
PRECIO         NUMBER(6,2)             NOT NULL
);

--
-- PK_TARIFA (Index)
--
CREATE UNIQUE INDEX PK_TARIFA ON TARIFA
(ID_TARIFA);

--
-- Non Foreign Key Constraints for Table TARIFA
--
ALTER TABLE TARIFA ADD (
  CONSTRAINT PK_TARIFA PRIMARY KEY (ID_TARIFA)
  USING INDEX );

--
-- Foreign Key Constraints for Table TARIFA
--
ALTER TABLE TARIFA ADD (
  CONSTRAINT FK_TARIFA_TIPO_PISTA FOREIGN KEY (ID_TIPO_PISTA)
  REFERENCES TIPO_PISTA (ID_TIPO_PISTA));

--
-- RESERVA (Table)
--
CREATE TABLE RESERVA
(
  ID_RESERVA    NUMBER          NOT NULL,
  ID_USUARIO    NUMBER          NOT NULL,
  ID_PISTA      NUMBER          NOT NULL,
  ID_TARIFA     NUMBER          NOT NULL,
  FECHA         DATE           NOT NULL
);

--
-- PK_RESERVA (Index)
--
CREATE UNIQUE INDEX PK_RESERVA ON RESERVA
(ID_RESERVA);

--
-- Non Foreign Key Constraints for Table RESERVA
--
ALTER TABLE RESERVA ADD (
  CONSTRAINT PK_RESERVA PRIMARY KEY (ID_RESERVA)
  USING INDEX );

--
-- Foreign Key Constraints for Table RESERVA
--
ALTER TABLE RESERVA ADD (
  CONSTRAINT FK_RESERVA_USUARIO FOREIGN KEY (ID_USUARIO)
  REFERENCES USUARIO (ID_USUARIO));

ALTER TABLE RESERVA ADD (
  CONSTRAINT FK_RESERVA_PISTA FOREIGN KEY (ID_PISTA)
  REFERENCES PISTA (ID_PISTA));

ALTER TABLE RESERVA ADD (
```



```

Insert into tarifa(ID_TARIFA, ID_TIPO_PISTA, HORA_INICIO, HORA_FIN, PRECIO)
Values (TARIFA_ID_SEQ.NEXTVAL,3,'16:00','17:00',15);
Insert into tarifa(ID_TARIFA, ID_TIPO_PISTA, HORA_INICIO, HORA_FIN, PRECIO)
Values (TARIFA_ID_SEQ.NEXTVAL,3,'17:00','18:00',15);
Insert into tarifa(ID_TARIFA, ID_TIPO_PISTA, HORA_INICIO, HORA_FIN, PRECIO)
Values (TARIFA_ID_SEQ.NEXTVAL,3,'18:00','19:00',15);
Insert into tarifa(ID_TARIFA, ID_TIPO_PISTA, HORA_INICIO, HORA_FIN, PRECIO)
Values (TARIFA_ID_SEQ.NEXTVAL,3,'19:00','20:00',15);
Insert into tarifa(ID_TARIFA, ID_TIPO_PISTA, HORA_INICIO, HORA_FIN, PRECIO)
Values (TARIFA_ID_SEQ.NEXTVAL,3,'20:00','21:00',30);
Insert into tarifa(ID_TARIFA, ID_TIPO_PISTA, HORA_INICIO, HORA_FIN, PRECIO)
Values (TARIFA_ID_SEQ.NEXTVAL,3,'21:00','22:00',30);
Insert into tarifa(ID_TARIFA, ID_TIPO_PISTA, HORA_INICIO, HORA_FIN, PRECIO)
Values (TARIFA_ID_SEQ.NEXTVAL,3,'22:00','23:00',30);

-- INSERTING into PISTA
Insert into PISTA (ID_PISTA,NUM_PISTA,ID_TIPO_PISTA) values
(PISTA_ID_SEQ.NEXTVAL,1,1);
Insert into PISTA (ID_PISTA,NUM_PISTA,ID_TIPO_PISTA) values
(PISTA_ID_SEQ.NEXTVAL,2,1);
Insert into PISTA (ID_PISTA,NUM_PISTA,ID_TIPO_PISTA) values
(PISTA_ID_SEQ.NEXTVAL,1,2);
Insert into PISTA (ID_PISTA,NUM_PISTA,ID_TIPO_PISTA) values
(PISTA_ID_SEQ.NEXTVAL,2,2);
Insert into PISTA (ID_PISTA,NUM_PISTA,ID_TIPO_PISTA) values
(PISTA_ID_SEQ.NEXTVAL,1,3);

-- INSERTING into USUARIO
Insert into USUARIO
(ID_USUARIO, NOMBRE_USUARIO, PASSWORD, NOMBRE, APELLIDOS, MOVIL, EMAIL, SEXO,
FECHA_NACIMIENTO, ES_ADMINISTRADOR, FECHA_ALTA)
Values
(USUARIO_ID_SEQ.NEXTVAL, 'admin', 'admin', 'Usuario', 'Administrador',
'6666666666', 'admin@admin.es', 'H', TO_DATE('01/01/1980', 'MM/DD/RRRR'), 'S',
TO_DATE('01/01/2013', 'MM/DD/RRRR'));
Insert into USUARIO
(ID_USUARIO, NOMBRE_USUARIO, PASSWORD, NOMBRE, APELLIDOS, MOVIL, EMAIL, SEXO,
FECHA_NACIMIENTO, ES_ADMINISTRADOR, FECHA_ALTA)
Values
(USUARIO_ID_SEQ.NEXTVAL, 'USER', 'USER', 'Usuario', 'Normal', '6666666666',
'user@user.es', 'H', TO_DATE('01/01/1990', 'MM/DD/RRRR'), 'N',
TO_DATE('01/01/2013', 'MM/DD/RRRR'));

-- INSERTING into RESERVA
Insert into RESERVA
(ID_RESERVA, ID_USUARIO, ID_PISTA, ID_TARIFA, FECHA)
Values
(RESERVA_ID_SEQ.NEXTVAL, 1, 1, 1, TO_DATE('06/13/2013', 'MM/DD/YYYY'));
Insert into RESERVA
(ID_RESERVA, ID_USUARIO, ID_PISTA, ID_TARIFA, FECHA)
Values
(RESERVA_ID_SEQ.NEXTVAL, 2, 2, 2, TO_DATE('06/13/2013', 'MM/DD/YYYY'));
Insert into RESERVA
(ID_RESERVA, ID_USUARIO, ID_PISTA, ID_TARIFA, FECHA)
Values
(RESERVA_ID_SEQ.NEXTVAL, 2, 2, 1, TO_DATE('06/13/2013', 'MM/DD/YYYY'));
Insert into RESERVA
(ID_RESERVA, ID_USUARIO, ID_PISTA, ID_TARIFA, FECHA)
Values

```

```
(RESERVA_ID_SEQ.NEXTVAL, 1, 1, 1, TO_DATE('06/14/2013', 'MM/DD/YYYY'));
Insert into RESERVA
  (ID_RESERVA, ID_USUARIO, ID_PISTA, ID_TARIFA, FECHA)
Values
  (RESERVA_ID_SEQ.NEXTVAL, 2, 2, 1, TO_DATE('06/14/2013', 'MM/DD/YYYY'));
Insert into RESERVA
  (ID_RESERVA, ID_USUARIO, ID_PISTA, ID_TARIFA, FECHA)
Values
  (RESERVA_ID_SEQ.NEXTVAL, 1, 1, 1, TO_DATE('06/15/2013', 'MM/DD/YYYY'));
Insert into RESERVA
  (ID_RESERVA, ID_USUARIO, ID_PISTA, ID_TARIFA, FECHA)
Values
  (RESERVA_ID_SEQ.NEXTVAL, 2, 2, 2, TO_DATE('06/15/2013', 'MM/DD/YYYY'));
Insert into RESERVA
  (ID_RESERVA, ID_USUARIO, ID_PISTA, ID_TARIFA, FECHA)
Values
  (RESERVA_ID_SEQ.NEXTVAL, 2, 2, 1, TO_DATE('06/15/2013', 'MM/DD/YYYY'));
Insert into RESERVA
  (ID_RESERVA, ID_USUARIO, ID_PISTA, ID_TARIFA, FECHA)
Values
  (RESERVA_ID_SEQ.NEXTVAL, 1, 1, 1, TO_DATE('06/16/2013', 'MM/DD/YYYY'));
```

6. Conclusiones

- Lo primero que sorprende al iniciarse en el mundo de Java y especialmente en el de J2EE, es la cantidad de frameworks disponibles en el mercado, por lo que inmediatamente asaltan las dudas: ¿Qué framework elijo? ¿Con cual será más sencillo iniciarse en el mundo de J2EE? ¿Cuál me ofrecerá mayores salidas profesionales en el futuro?
- Al final me decidí por JBoss Seam porque es el que se utiliza en mi empresa (aunque yo no lo usara porque, como ya dije, me dedico más al ámbito de las bases de datos, especialmente Oracle). Creo que fue una decisión acertada puesto que ya participo en algunos desarrollos con esta tecnología y se ha producido una retroalimentación: en mi proyecto podía utilizar los nuevos conocimientos adquiridos en el trabajo y viceversa.
- Hemos conseguido cumplir con los objetivos iniciales del proyecto: profundizar en los conocimientos de J2EE, desarrollar una aplicación con esta tecnología, gracias a los conocimientos adquiridos tanto en el estudio de la misma como a lo largo de toda la carrera.
- Considero que el desarrollo de este proyecto es todo un reto, puesto que no tenía ningún tipo de conocimiento previo de la tecnología y, a pesar de suponer un gran esfuerzo el estudio de ésta, una vez se tienen unos conocimientos mínimos es bastante sencillo desarrollar aplicaciones bastante bien construidas.

Glosario

- **Tipo de Pista:** Los distintos tipos de pista que podemos encontrar en el centro deportivo que gestiona nuestra aplicación. Por ejemplo: pádel, tenis, fútbol, baloncesto, etc.
- **Pista:** Cada una de las pistas deportivas que la aplicación pretende gestionar.
- **Tarifa:** Es el precio que vale cada tipo de pista a una determinada hora.
- **Reserva:** Es el proceso por el cual un usuario se asigna el derecho a utilizar una pista, para un día y una hora en concreto, de acuerdo con la tarifa correspondiente.

Bibliografía

Allen, Dan (2008). Seam in Action. Manning.

Bauer, Christian; King, Gavin (2005). Hibernate In Action. Manning.

Booch,Grady; Rumbaugh,James;Jacobson, Ivar(1999). El lenguaje unificado de modelado. Addison Wesley.

Mukhar, Kevin; Zelenak, Chris (2006). Beginning Java EE 5 Platform: From Novice to Professional. Apress

Pressman, Roger S.(2006). Ingeniería delsoftware:Un enfoque práctico (6ª edición). McGraw-Hill.

Roman, Ed; Patel Sriganesh, Rima; Brose,Gerald (2005). Mastering Enterprise JavaBeans(3ª edición). Wiley.

Recursos on-line

<http://www.seamframework.org>

<http://www.ibm.com/developerworks/library/j-seam1/>

Anexos

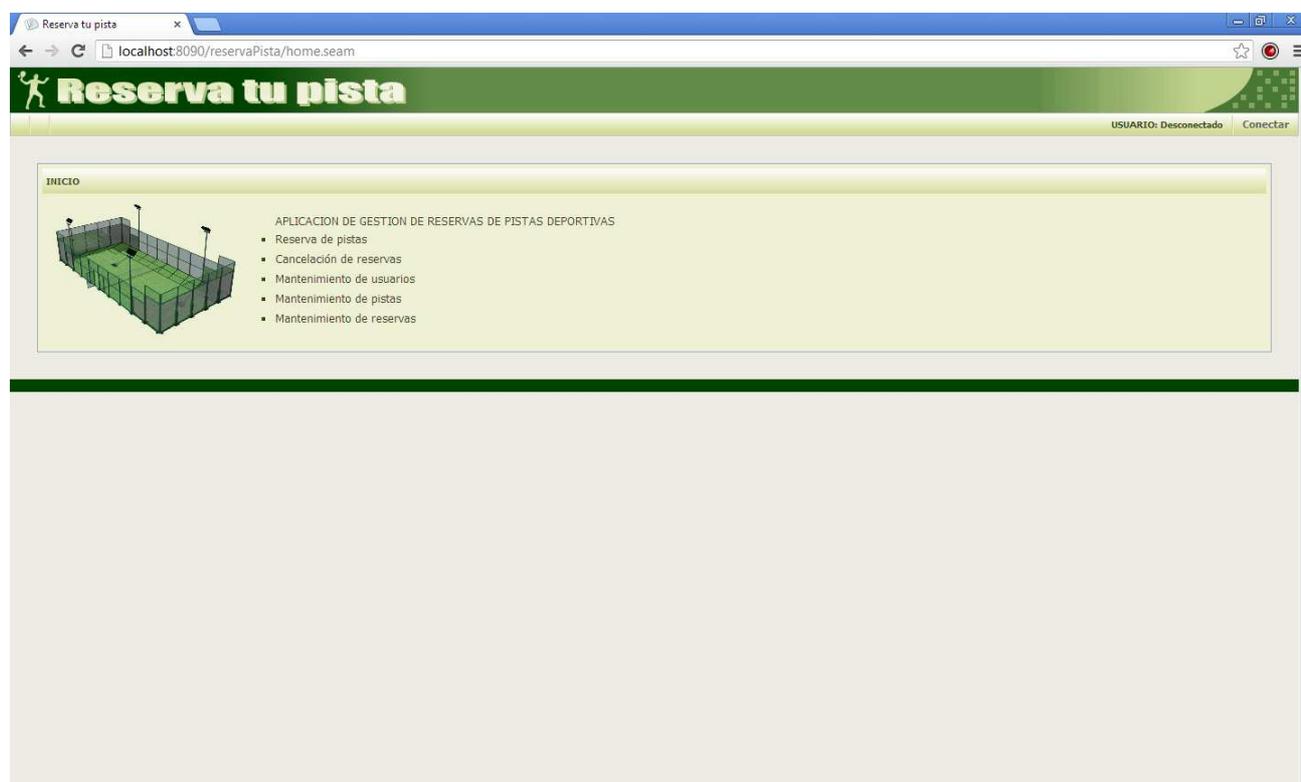
Manual Básico de Usuario

Para acceder a la aplicación deberemos abrir un navegador* y acceder a la siguiente URL:

<http://localhost:8090/reservaPista>

(*Nota: Hemos observado que la versión 10 de Internet Explorer no muestra los ModalPanel que son ampliamente utilizados en nuestra aplicación, por lo que se recomienda utilizar otro navegador o una versión anterior de Internet Explorer)

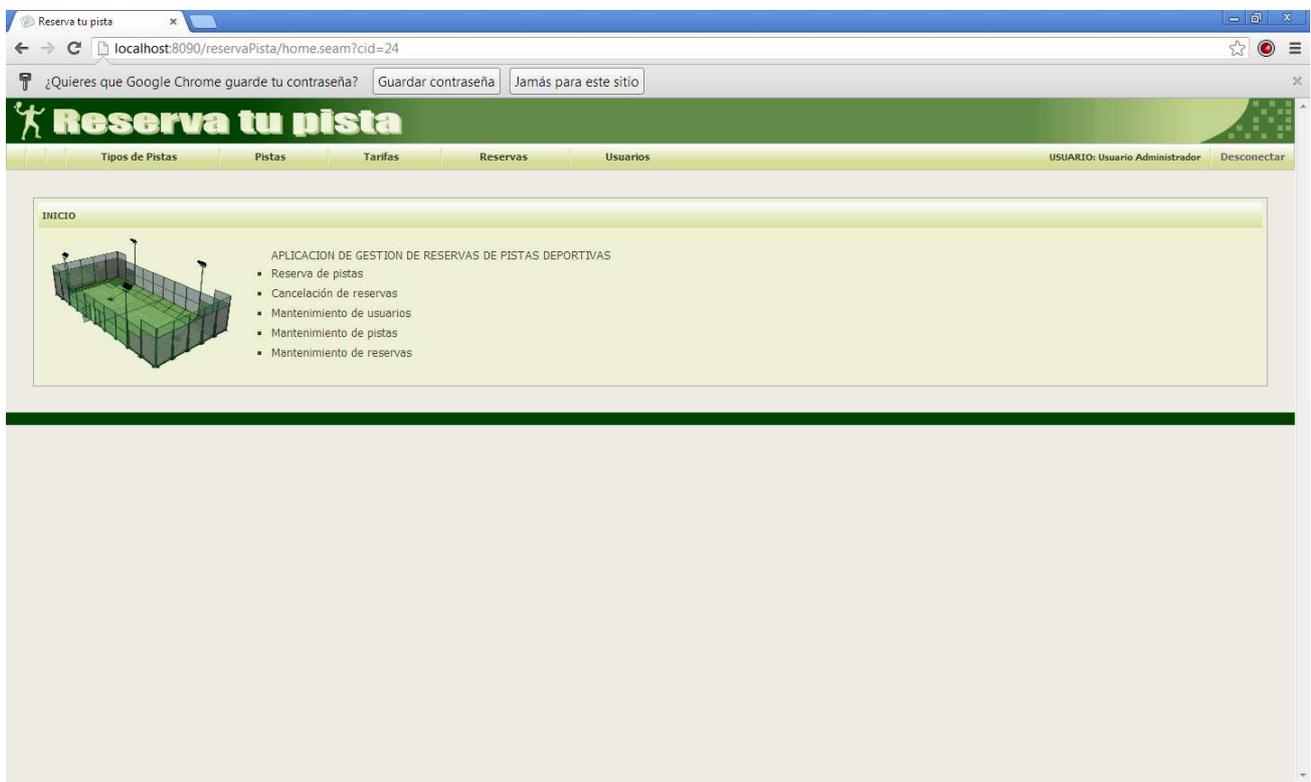
Accederemos a la siguiente pantalla:



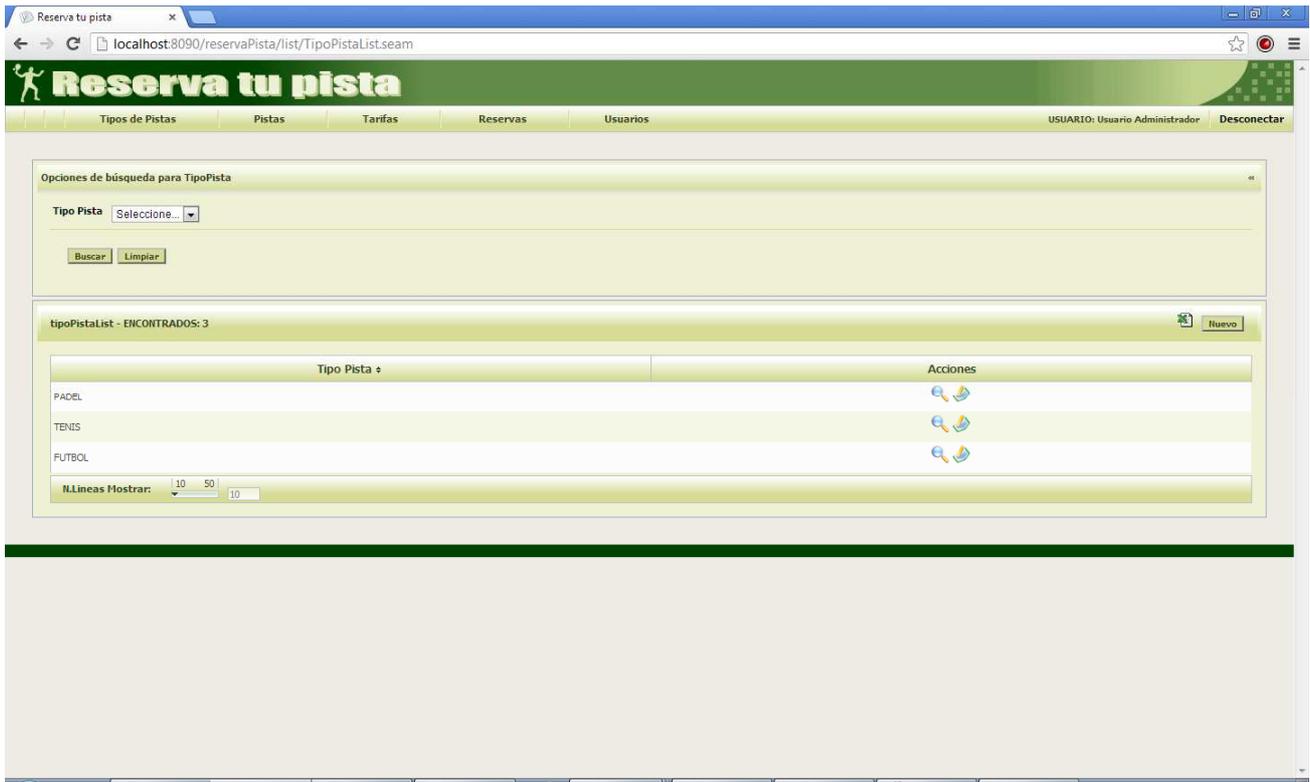
donde deberemos pulsar el botón conectar para poder logarnos:



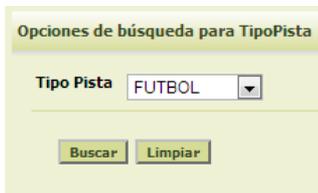
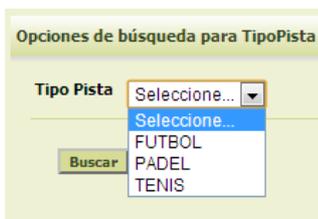
Una vez introducidos nuestro Usuario y clave pasaremos a distintas pantallas dependiendo de si somos Administrador o un usuario normal. En el caso del administrador:



Vamos ahora a la pantalla de gestión de Tipos de Pistas, clickando en la opción del menú:



Explicaremos ahora una serie de características comunes a la mayoría de pantallas. Si queremos hacer un filtrado del listado inferior de la pantalla, seleccionamos en el combo o escribimos en las cajas de texto los filtros por los que queremos buscar y pulsamos el botón Buscar y veremos como el listado cambia los resultados que muestra.

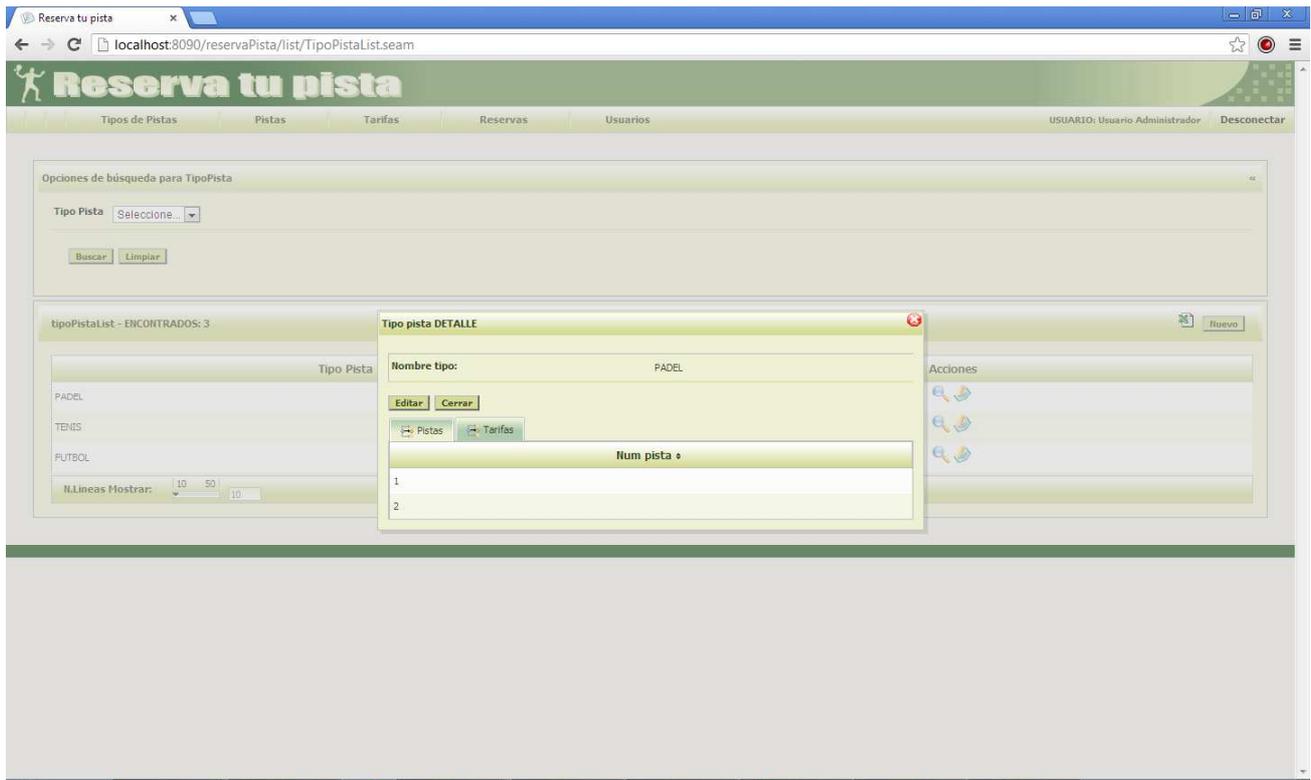


En la parte derecha de la pantalla tenemos el botón Nuevo, para añadir un nuevo registro del tipo de la pantalla en la que nos encontremos, ya sea TipoPista, Pista, Tarifa o Usuario. No se permite añadir por este método una reserva, ya que se harán por el módulo correspondiente para realizar reservas.

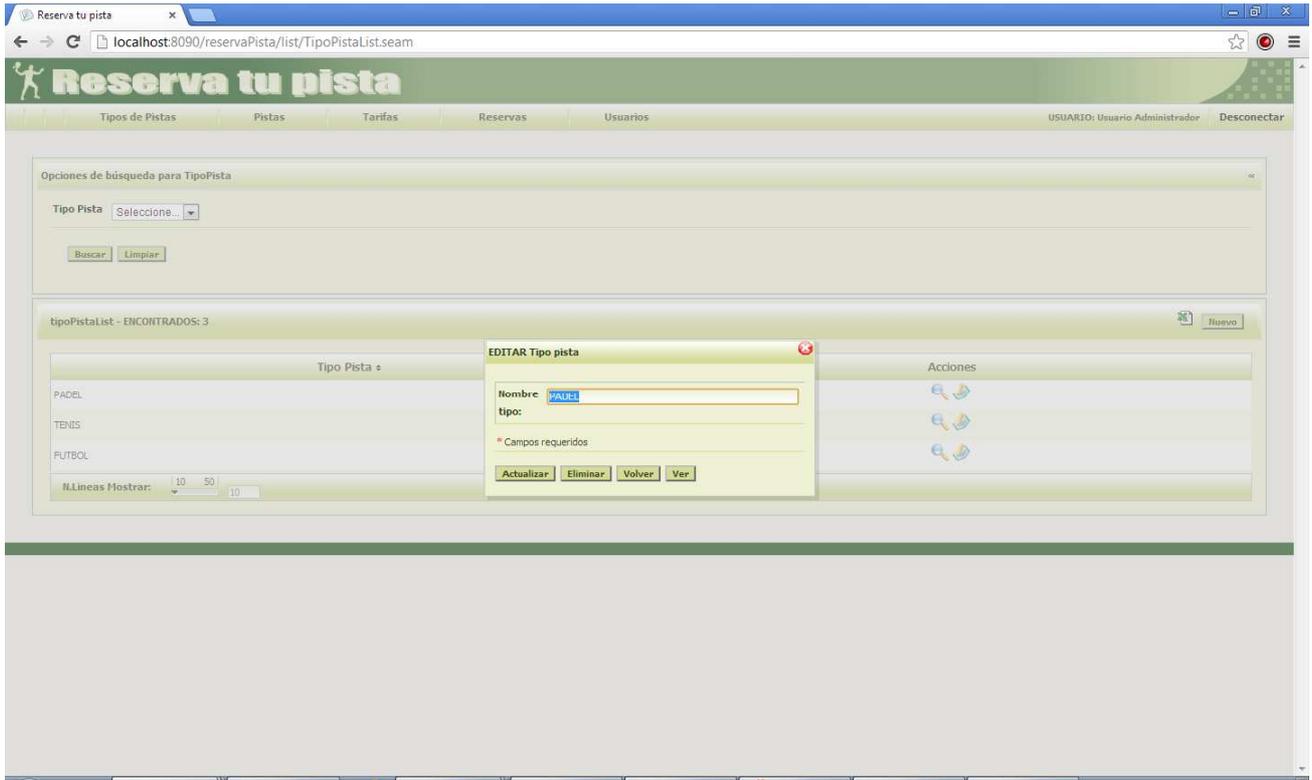


Junto al botón Nuevo encontramos un botón con el símbolo de Excel, que nos exporta el listado que se está mostrando en ese momento a un archivo Excel.

Si pulsamos el botón Ver (la lupa) en cualquier pantalla, podremos Ver los datos de la entidad que hemos seleccionado, así como las relaciones que tiene con otras entidades.



Si desde esta pantalla Ver pulsamos el botón Editar, o bien, desde cualquier listado pulsamos en el logo de Editar (el lápiz), podremos Editar los datos propios de esta entidad, para el registro seleccionado.



Vemos aquí el listado de Pistas:



Y en esta otra imagen el listado de Tarifas:

The screenshot shows the 'Reserva tu pista' web application interface. The browser address bar indicates the URL is localhost:8090/reservaPista/list/TarifaList.seam. The page title is 'Reserva tu pista'. The navigation menu includes 'Tipos de Pistas', 'Pistas', 'Tarifas', 'Reservas', and 'Usuarios'. The user is logged in as 'USUARIO: Usuario Administrador' and can click 'Desconectar'.

Below the navigation menu is a search form titled 'Opciones de búsqueda para Tarifa'. It includes a 'Tipo Pista' dropdown menu, 'Hora inicio' and 'Hora fin' input fields, and 'Buscar' and 'Limpiar' buttons.

The main content area shows 'tarifaList - ENCONTRADOS: 10'. Below this is a table with the following columns: 'Tipo Pista', 'Hora Inicio', 'Hora Fin', 'Precio', and 'Acciones'. The table contains 10 rows of data:

Tipo Pista	Hora Inicio	Hora Fin	Precio	Acciones
PADEL	08:00	09:00	5,0 Euros	[Icons]
TENIS	09:00	10:00	9,0 Euros	[Icons]
FUTBOL	09:00	10:00	15,0 Euros	[Icons]
PADEL	09:00	10:00	6,0 Euros	[Icons]
TENIS	10:00	11:00	9,0 Euros	[Icons]
FUTBOL	10:00	11:00	15,0 Euros	[Icons]
PADEL	10:00	11:00	6,0 Euros	[Icons]
TENIS	11:00	12:00	9,0 Euros	[Icons]
FUTBOL	11:00	12:00	15,0 Euros	[Icons]
PADEL	11:00	12:00	6,0 Euros	[Icons]

At the bottom of the table, there is a pagination control showing 'Desde: 1' to 'Hasta: 10' and 'Páginas: 1 / 5'. There are also 'N.Líneas Mostrar' controls set to 10.

En esta captura podemos ver el listado de Reservas. Un usuario Administrador podrá ver las reservas de todos los usuarios, mientras que un usuario normal sólo podrá ver sus reservas:

The screenshot shows the 'Reserva tu pista' web application interface. The browser address bar indicates the URL is localhost:8090/reservaPista/list/ReservaList.seam. The page title is 'Reserva tu pista'. The navigation menu includes 'Tipos de Pistas', 'Pistas', 'Tarifas', 'Reservas', and 'Usuarios'. The user is logged in as 'USUARIO: Usuario Administrador' and can click 'Desconectar'.

Below the navigation menu is a search form titled 'Opciones de búsqueda para Reserva'. It includes a 'Tipo Pista' dropdown menu, a 'Usuario' dropdown menu, a 'Fecha' input field, and 'Buscar' and 'Limpiar' buttons.

The main content area shows 'reservaList - ENCONTRADOS: 4'. Below this is a table with the following columns: 'Pista', 'Usuario', 'Fecha', 'Hora Inicio', 'Hora Fin', and 'Acciones'. The table contains 4 rows of data:

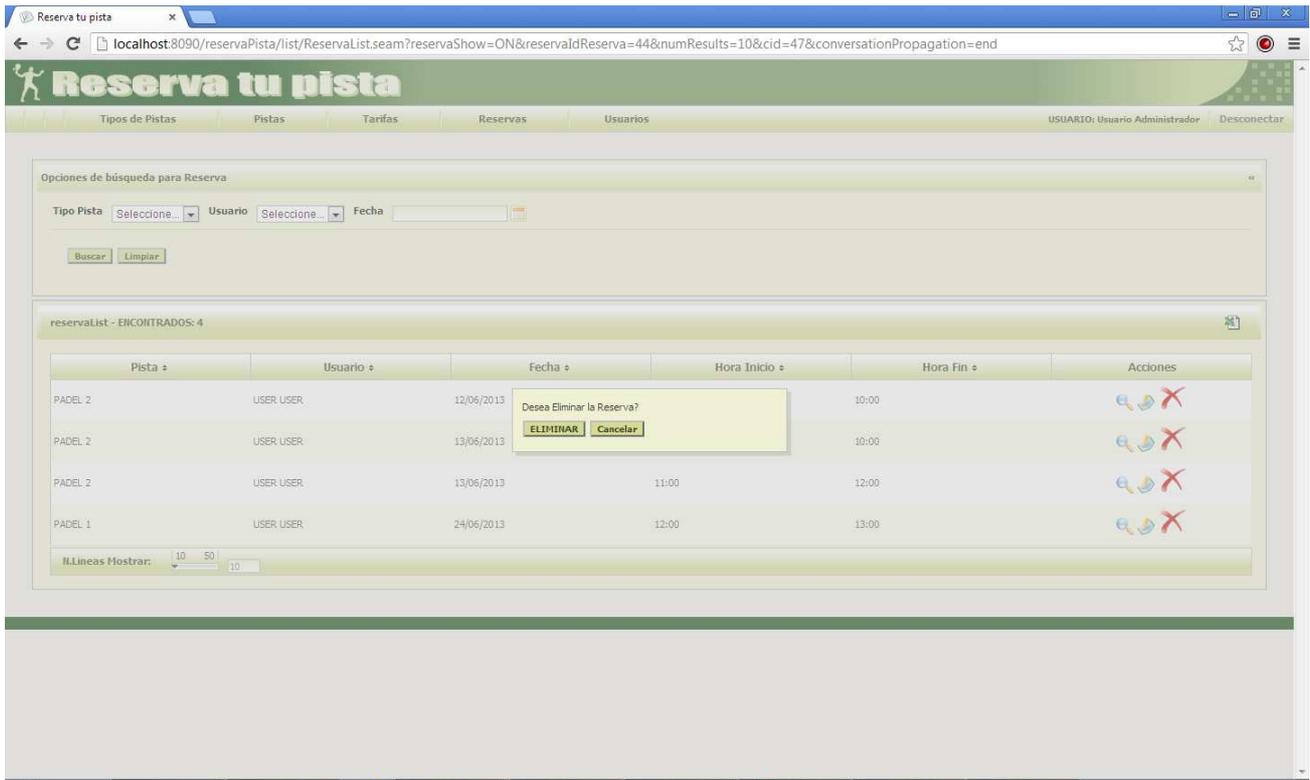
Pista	Usuario	Fecha	Hora Inicio	Hora Fin	Acciones
PADEL 2	USER USER	12/06/2013	09:00	10:00	[Icons]
PADEL 2	USER USER	13/06/2013	09:00	10:00	[Icons]
PADEL 1	USER USER	13/06/2013	12:00	13:00	[Icons]
PADEL 2	USER USER	13/06/2013	11:00	12:00	[Icons]

At the bottom of the table, there is a pagination control showing 'N.Líneas Mostrar' set to 10.

En este caso se permite eliminar la reserva, siempre que no haya pasado todavía la fecha, pulsando en el botón correspondiente (el aspa roja).



Nos pedirá una confirmación, por si le hubiésemos pulsado por error:



The screenshot shows the 'Reserva tu pista' web application interface. The browser address bar shows the URL: localhost:8090/reservaPista/list/ReservaList.seam?reservaShow=ON&reservaIdReserva=44&numResults=10&cid=47&conversationPropagation=end. The application header includes navigation tabs: Tipos de Pistas, Pistas, Tarifas, Reservas, and Usuarios. The user is logged in as 'USUARIO: Usuario Administrador' and can click 'Desconectar'. Below the header is a search section for reservations with dropdowns for 'Tipo Pista', 'Usuario', and 'Fecha', and 'Buscar' and 'Limpiar' buttons. The main content area displays a table of reservations with 4 results. A confirmation dialog is overlaid on the first row, asking 'Desea Eliminar la Reserva?' with 'ELIMINAR' and 'Cancelar' buttons. The table columns are: Pista, Usuario, Fecha, Hora Inicio, Hora Fin, and Acciones. The 'Acciones' column contains icons for search, edit, and delete.

Pista	Usuario	Fecha	Hora Inicio	Hora Fin	Acciones
PADEL 2	USER USER	12/06/2013		10:00	[Search] [Edit] [Delete]
PADEL 2	USER USER	13/06/2013		10:00	[Search] [Edit] [Delete]
PADEL 2	USER USER	13/06/2013	11:00	12:00	[Search] [Edit] [Delete]
PADEL 1	USER USER	24/06/2013	12:00	13:00	[Search] [Edit] [Delete]

La siguiente pantalla es la de administración de usuarios, que lógicamente sólo está accesible para los administradores:

Opciones de búsqueda para Usuario

Nombre usuario Password Es administrador Fecha alta

Nombre Apellidos Sexo Fecha nacimiento

Movil Email

usuarioList - ENCONTRADOS: 2

Nombre Usuario	Nombre	Apellidos	Sexo	Fecha Nacimiento	Administrador	Fecha Alta	Movil	Email	Acciones
admin	Usuario	Administrador	H	01/01/1980	<input checked="" type="checkbox"/>	01/06/2013	666666666	yo@yo.es	
USER	USER	USER	H	01/01/1982	<input type="checkbox"/>	01/06/2013	666666666	user@user.es	

N.Líneas Hostrar:

Desde el punto de vista del usuario normal de la aplicación, sólo tendremos accesibles dos pantallas. La de ver sus propias reservas que ya hemos visto anteriormente y la de Reservar una Pista. Para ello, es obligatorio que indique qué tipo de pista quiere reservar y la fecha para la que quiere hacer la reserva:

Opciones de búsqueda para Reservar Pista

Tipo Pista Fecha

N.Líneas Hostrar:

Una vez ha establecido los criterios que desea para hacer una reserva, al pulsar el botón Buscar, se muestra un listado de todas los tipos de pistas a las distintas horas del día seleccionado y la disponibilidad de dichas pistas. Si no hay ninguna pista libre de ese tipo para una hora concreta de ese día muestra un mensaje. Si hubiera alguna pista libre, nos muestra un botón Reservar:

The screenshot shows a web browser window with the URL `localhost:8090/reservaPista/list/DisponiblesList.seam`. The page title is "Reserva tu pista" and the user is logged in as "USUARIO: Usuario Normal".

Under "Opciones de búsqueda para Reservar Pista", the search criteria are:

- Tipo Pista: PADEL
- Fecha: 13/06/2013

 Buttons for "Buscar" and "Limpiar" are visible.

The main table displays the following data:

Tipo Pista	Hora Inicio	Hora Fin	Precio ↓	Disponibilidad
PADEL	08:00	09:00	5.0 Euros	Reservar
PADEL	09:00	10:00	6.0 Euros	No hay pistas libres
PADEL	10:00	11:00	6.0 Euros	Reservar
PADEL	11:00	12:00	6.0 Euros	Reservar
PADEL	12:00	13:00	6.0 Euros	Reservar
PADEL	13:00	14:00	6.0 Euros	Reservar
PADEL	14:00	15:00	6.0 Euros	Reservar
PADEL	15:00	16:00	6.0 Euros	Reservar
PADEL	16:00	17:00	6.0 Euros	Reservar
PADEL	17:00	18:00	6.0 Euros	Reservar

At the bottom of the table, there are pagination controls: "N.Líneas Mostrar:" with a dropdown set to 10, "Desde: 1" and "Hasta: 10", and "Páginas: 1 / 2".

Una vez pulsado este botón, nos muestra un listado con la disponibilidad de todas las pistas del tipo seleccionado para el día y la hora elegidos. Si está disponible, nos muestra un botón elegir (aspa verde) y si no lo estuviera muestra un prohibido. Si pulsamos el botón elegir, se inserta un nuevo registro en la tabla Reserva, de acuerdo con los datos elegidos durante el proceso. Si intentamos repetir el proceso para el mismo tipo de pista, día, hora y pista elegida, ya no nos aparecerá disponible puesto que tenemos hecha una reserva:

