

Desarrollo de una aplicación Web para la gestión de hipódromos mediante la tecnología J2EE

ibTrot

Santiago Serradilla Fernández
ITIG

Consultor: Albert Grau Perisé
18/06/2007

© **Santiago Serradilla Fernández** (el autor/a)
sannti@gmail.com

Reservados todos los derechos. Está prohibida la reproducción total o parcial de esta obra para cualquier medio o procedimiento, compresos la impresión, la reprografía, el microfilm, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorizan la Ley de Propiedad Intelectual.

Creative Commons

Este trabajo está sujeto – excepto que se indique lo contrario- a una licencia de Reconocimiento-NoComercial-SenseObraDerivada 2.5 España de Creative Commons. Podéis copiarlo, distribuirlos y transmitirlos públicamente siempre que citéis el autor i 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/2.5/es/deed.es>

(Licencia GPL)

Puede copiar y distribuir el Programa (o un trabajo basado en él, según se especifica en el apartado 2, como código objeto o en formato ejecutable según los términos de los apartados 1 y 2, supuesto que además cumpla una de las siguientes condiciones:

- 1. Acompañarlo con el código fuente completo correspondiente, en formato electrónico, que debe ser distribuido según se especifica en los apartados 1 y 2 de esta Licencia en un medio habitualmente utilizado para el intercambio de programas, o*
- 2. Acompañarlo con una oferta por escrito, válida durante al menos tres años, de proporcionar a cualquier tercera parte una copia completa en formato electrónico del código fuente correspondiente, a un coste no mayor que el de realizar físicamente la distribución del fuente, que será distribuido bajo las condiciones descritas en los apartados 1 y 2 anteriores, en un medio habitualmente utilizado para el intercambio de programas, o*
- 3. Acompañarlo con la información que recibiste ofreciendo distribuir el código fuente correspondiente. (Esta opción se permite sólo para distribución no comercial y sólo si usted recibió el programa como código objeto o en formato ejecutable con tal oferta, de acuerdo con el apartado b anterior).*

Resumen del trabajo de fin de carrera. J2EE

El presente documento se basa en las etapas de recogida y documentación de requisitos, análisis, diseño, implementación y testing de una aplicación Web utilizando la tecnología J2EE.

El desarrollo de las etapas de requisitos, análisis y diseño son independientes de la tecnología que será utilizada para la implementación de la aplicación Web. En estas etapas (o fases) se ha potenciado el paradigma de la programación orientada a objetos ya que define los programas en términos de "clases de objetos". Estos objetos son entidades que combinan *estado* (datos), *comportamiento* (procedimientos) e *identidad* (propiedades del objeto que lo diferencia del resto). Esta cualidad permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

Para el desarrollo de la etapa de implementación se ha utilizado el lenguaje *java* junto con otras tecnologías que serán explicadas más adelante. Cabe destacar el uso de patrones de diseño utilizados para el desarrollo de la aplicación, en especial el patrón de diseño MVC que ha sido desarrollado a partir del framework *Struts*. Se ha elegido este framework ya que es utilizado ampliamente en el sector empresarial y es considerado de gran solidez.

ÍNDICE

1. Introducción.....	5
1.1 Justificación del TFC	5
1.2 Objetivos	5
1.3 Enfoque y método seguido	6
1.4 Planificación del proyecto	6
1.5 Productos obtenidos	8
1.6 Descripción del resto de capítulos de la memoria	8
2. Especificación y análisis de requisitos.....	9
2.1 Preámbulo	9
2.2 Descripción del proyecto	9
2.3 Composición del software: subsistemas	9
2.4 Descripción de las funcionalidades por subsistema	10
2.5 Actores del sistema	11
2.6 Seguridad	11
2.7 Diagramas de caso de uso por subsistema.....	11
2.7.1 Subsistema de conexión	11
2.7.2 Subsistema de mantenimiento	12
2.7.3 Subsistema de gestión	13
2.7.4 Subsistema de información	14
2.8 Descripción de los casos de uso por subsistema.....	14
2.8.1 Subsistema de conexión	14
2.8.2 Subsistema de mantenimiento	14
2.8.3 Subsistema de gestión	23
2.8.4 Subsistema de información	30
3. Diseño.....	33
3.1 Diagrama de clases	33
3.1.1 Diagrama de entidades	33
3.1.2 Diagrama de jerarquías	33
3.1.2.1 Ventanas principales de la aplicación	34
3.1.2.2 Ventanas de la aplicación por subsistema	34
3.2 Diagramas de secuencia.....	36
3.3 Diseño de la base de datos	42
3.4 Arquitectura	45
3.5 Patrones de diseño	46
4. Implementación.....	49
4.1 Tecnología y herramientas utilizadas	49
5. Valoración económica.....	51
6. Conclusiones.....	52
7. Glosario.....	53

8. Bibliografía.....	54
9. Anexo.....	56
9.1 Manual del usuario	56
9.2 Licencia del framework Common Controls	56

ÍNDICE DE FIGURAS

Figura 1 - Caso de uso conexión.....	11
Figura 2 - Caso de uso mantenimiento	12
Figura 3 - Caso de uso gestión	13
Figura 4 - Caso de uso información	14
Figura 5 - Diagrama de entidades.....	33
Figura 6 - Ventanas principales	34
Figura 7 - Ventanas subsistema conexión	34
Figura 8 - Ventanas subsistema información	34
Figura 9 - Ventanas subsistema mantenimiento	35
Figura 10 - Ventanas subsistema gestión.....	36
Figura 11 – Diagrama de secuencias: Listado de caballos	37
Figura 12 – Diagrama de secuencias: Actualizar listado de caballos	37
Figura 13 – Diagrama de secuencias: Alta de caballos	38
Figura 14 – Diagrama de secuencias: Editar caballos	39
Figura 15 – Diagrama de secuencias: Eliminar caballo.....	40
Figura 16 – Diagrama de secuencias: Información detallada del caballo	41
Figura 17 – Diagrama arquitectura 3 capas	45
Figura 18 – Diagrama Struts MVC	47

1. INTRODUCCIÓN

1.1 Justificación del TFC: punto de partida y aportación

La realización del TFC parte de la necesidad de demostrar los conocimientos que ha adquirido el alumno a través de las distintas asignaturas de Ingeniería Técnica de Informática.

Como punto de partida son esenciales los conocimientos adquiridos en las asignaturas: programación orientada a objetos, base de datos I, base de datos II, estructura de la información, ingeniería del software y técnicas de desarrollo de software para el desarrollo de este TFC.

Ha sido necesario un gran esfuerzo por parte del alumno el ir aprendiendo la tecnología J2EE a medida que se han desarrollado las distintas etapas de requisitos, análisis y diseño.

El desarrollo del TFC me ha aportado unos conocimientos generales sobre la tecnología J2EE, patrones de diseño y frameworks como *Struts* y *Common-Controls* (www.common-controls.com) que sin ningún tipo de duda me serán de gran utilidad en mi vida laboral.

1.2 Objetivos

El TFC tiene como objetivo principal poner de manifiesto el aprendizaje que ha realizado el alumno a lo largo de los estudios de Ingeniería Técnica de Informática. En mi caso los estudios de ITI de Gestión.

El desarrollo de aplicaciones empresariales está cada vez más enfocado hacia el uso de una estructura distribuida con el fin de aprovechar las ventajas que nos aporta la red Internet. El uso de la arquitectura J2EE se ha establecido como un estándar a seguir para el desarrollo de aplicaciones empresariales distribuidas.

La tecnología Java aporta las herramientas necesarias para desarrollar aplicaciones mediante la Programación Orientada a Objetos con independencia del sistema operativo en el que se trabaje.

Con el desarrollo del TFC el alumno adquiere los conocimientos generales sobre dicha arquitectura, de forma que le sean útiles en el mundo laboral.

1.3 Enfoque y método seguido.

El método seguido para desarrollar el TFC es el propio del ciclo de vida del desarrollo de software orientado a objetos que divide el desarrollo de la aplicación Web en distintas etapas.

El ciclo de vida elegido para el desarrollo de la aplicación Web ha sido el ciclo de vida “clásico” o también llamado “en cascada” de forma que en cada etapa se obtiene una serie de documentos que sirven de base para la siguiente etapa y que por tanto no podemos empezar una etapa sin haber finalizado la etapa anterior. Las etapas desarrolladas y su orden es el siguiente:

1. Recogida y documentación de requisitos
2. Análisis
3. Diseño
4. Implementación
5. Testing

Para el desarrollo de estas distintas etapas ha sido esencial los conocimientos adquiridos en las asignaturas ingeniería del software y técnicas de desarrollo de software.

La curva de aprendizaje de la tecnología J2EE se ha realizado paralelamente a las cuatro primeras etapas.

1.4 Planificación del proyecto

- **Especificación y análisis de requisitos**

- | | |
|---|-------------------------|
| 1. Descripción del proyecto | (1 día: 19/03) |
| 2. Resumen esquemático de las necesidades | (2 días: 20/03 - 21/03) |
| 3. Subsistemas – Funcionalidades por subsistema | (2 días: 22/03 - 23/03) |
| 4. Identificar las entidades de la aplicación | (1 día: 24/03) |
| 5. Diagrama de clases inicial | (2 días: 25/03 - 26/03) |
| 6. Actores del sistema | (1 día: 27/03) |
| 7. Diagrama de casos de uso | (4 días: 28/03 - 31/03) |
| 8. Especificación formal de los casos de uso | (6 días: 01/04 - 06/04) |
| 9. Planificación del proyecto | (2 días: 07/04 - 08/04) |
| 10. Glosario | (1 día: 09/04) |
| 11. Preparación y entrega PAC1 | (2 días: 10/04 - 11/04) |

Total días asignados: **24**

Intervalo de fechas: 19/03/2007 – 12/04/2007

- **Diseño**

- | | |
|---|-------------------------|
| 1. Revisión detallada del diagrama de clases | (2 días: 13/04 - 14/04) |
| 2. Diagramas de secuencia | (4 días: 15/04 - 18/04) |
| 3. Estudio de patrones de diseño | (3 días: 19/04 - 21/04) |
| 4. Definición de la arquitectura y tecnología | (3 días: 22/04 - 24/04) |
| 5. Diseño de la base de datos | (3 días: 25/04 - 27/04) |
| 6. Diseño de la interfaz gráfica de usuario | (2 días: 28/04 - 29/04) |
| 7. Preparación y entrega PAC2 | (4 días: 30/04 - 03/05) |

Total días asignados: **21**

Intervalo de fechas: 13/04/2007 – 3/05/2007

- **Implementación y testing**

- | | |
|---|--------------------------|
| 1. Instalación y configuración del software | (8 días: 04/05 - 11/05) |
| 2. Implementación | (23 días: 12/05 - 03/06) |
| 3. Preparación y entrega PAC3 | (1 día: 04/06) |
| 4. Juego de pruebas | (3 días: 05/06 - 07/06) |

Total días asignados: **35**

Intervalo de fechas: 4/05/2007 – 7/06/2007

- **Memoria**

- | | |
|---------------------------------------|-------------------------|
| 1. Realización de la documentación | (8 días: 08/06 - 15/06) |
| 2. Realización de la presentación | (2 días: 16/06 - 17/06) |
| 3. Preparación y entrega del proyecto | (1 día: 18/06) |

Total días asignados: **12**

Intervalo de fechas: 8/06/2007 – 18/06/2007

Cabe destacar que la curva de aprendizaje de la tecnología J2EE se ha realizado de forma paralela a las fases de análisis, requisitos y diseño, al igual que la instalación y configuración de software necesario para las pruebas que en un principio se había asignado a la fase de implementación.

Se han tenido en cuenta todos los días hábiles del curso a la hora de realizar la planificación inicial del proyecto. Aún así, se intentará dejar los domingos libres para el descanso del alumno. La pérdida de estos días se recompensará realizando en un mismo día varias tareas asignadas a distintos días.

1.5 Productos obtenidos

Los elementos que forman el trabajo de fin de carrera son los siguientes:

- La aplicación Web *ibTrot*, formada por:
 1. Los ficheros *ibTrot.jar* e *ibTrot.war* (ejecutables)
 2. Clases ejecutables y código fuente (.class y .java)
 3. Clases de presentación al usuario (.jsp)
 4. Librerías necesarias para su ejecución (.jar, .tld)
 5. Ficheros de configuración y despliegue (.properties, .xml)
 6. Imágenes de la aplicación
- Documentación formada por el *javadoc* de las clases de la aplicación.
- Script de creación de las tablas de la base de datos e inserción de datos en ellas.
- Memoria del proyecto, la cuál explica el proceso seguido para la construcción de la aplicación Web.
- La presentación (.ppt) que resume de una forma clara el proyecto realizado.
- Manual de usuario.

1.6 Descripción del resto de capítulos de la memoria

El resto de capítulos de la memoria explican el ciclo de vida seguido para el desarrollo de la aplicación Web: análisis y documentación de requisitos, diseño, implementación y testing.

2. ESPECIFICACIÓN Y ANÁLISIS DE REQUISITOS

2.1 Preámbulo

El proyecto a realizar surge como idea inicial de mi suegro, que es socio y miembro del equipo de gestión del hipódromo “Son Pardo” de Palma de Mallorca, debido a la necesidad de compartir una cierta información (boletines) accesibles desde los distintos hipódromos de Mallorca.

El desarrollo del proyecto podrá tener un uso final real, siendo de gran ayuda para la gestión interna de un hipódromo. También cabe destacar que se realizarán nuevas funcionalidades en un futuro próximo (como por ejemplo definir una estructura para poder compartir información entre varios hipódromos).

2.2 Descripción del proyecto

La idea consiste en realizar una aplicación Web accesible desde la red Internet que permita administrar la gestión interna de un hipódromo (podrá ser usada a modo de Intranet).

Actualmente tienen una pequeña aplicación en Access para almacenar diversos datos del hipódromo (miembros, caballos y carreras) mientras que todo el resto de información que manejan (boletines, sanciones, incidencias...) se encuentra en papel.

Inicialmente la aplicación servirá para tener un control-almacenaje de los datos internos del hipódromo y para la generación de “Boletines” accesibles desde la red Internet.

2.3 Composición del software: subsistemas

El software se encuentra dividido en 4 subsistemas:

- **Subsistema de Conexión:** se encarga del acceso y validación de los usuarios en la aplicación. Su función principal es la seguridad de la aplicación.
- **Subsistema de Mantenimiento:** se encarga de realizar las acciones asociadas a las siguientes entidades de la aplicación: hipódromos, miembros, caballos y usuarios (actores).
- **Subsistema de Gestión:** se encarga de realizar las acciones asociadas a las siguientes entidades de la aplicación: carreras, incidencias.
- **Subsistema de Información:** se encarga de la generación de boletines.

2.4 Descripción de las funcionalidades por subsistema

2.4.1 Subsistema de conexión

- Autenticar el acceso de los usuarios a la aplicación: permite acceder a los usuarios a la aplicación mediante la combinación usuario-contraseña.
- Sistema de seguridad

2.4.2 Subsistema de mantenimiento

- Gestión de hipódromos: permite realizar el alta, baja, modificación, información detallada y listado de los hipódromos. También nos permite refrescar la lista de hipódromos por si se han producido cambios en la base de datos.
- Gestión de miembros: permite realizar el alta, baja, modificación, información detallada y listado de los miembros. También nos permite refrescar la lista de miembros por si se han producido cambios en la base de datos.
- Gestión de caballos: permite realizar el alta, baja, modificación, información detallada y listado de los caballos. También nos permite refrescar la lista de caballos por si se han producido cambios en la base de datos.
- Gestión de usuarios: permite realizar el alta, baja, modificación, información detallada y listado de los usuarios (actores del sistema). También nos permite refrescar la lista de usuarios por si se han producido cambios en la base de datos.

2.4.3 Subsistema de gestión

- Gestión de carreras: permite realizar el alta, baja, modificación, información detallada y listado de las carreras. También nos permite refrescar la lista de carreras por si se han producido cambios en la base de datos.
- Gestión de incidencias: permite realizar el alta, baja, modificación, información detallada y listado de las incidencias. También nos permite refrescar la lista de incidencias por si se han producido cambios en la base de datos.
- Gestión de sanciones: permite realizar el alta, baja, modificación, información detallada y listado de las sanciones. También nos permite refrescar la lista de sanciones por si se han producido cambios en la base de datos.

2.4.4 Subsistema de información

- Gestión de boletines: permite realizar el alta, baja, modificación información detallada y listado de los boletines. . También nos permite refrescar la lista de boletines por si se han producido cambios en la base de datos.

2.5 Actores del sistema

El sistema tiene 3 tipos de actores: *administrador*, *gestor* y *auxiliar*. La diferencia entre ellos se debe a las funcionalidades que pueden ejecutar cada uno de ellos. El *auxiliar* solo puede ejecutar las funcionalidades asociadas a los subsistemas de conexión e información. El *gestor* solo puede ejecutar las funcionalidades asociadas a los subsistemas de conexión, gestión y mantenimiento (exceptuando el mantenimiento de usuarios). El *administrador* puede ejecutar todas las funcionalidades de la aplicación.

Nota: Debido a la amplitud del proyecto y el tiempo limitado para el desarrollo del mismo, no se han implementado las funcionalidades propias que distinguan que tipo de usuario se ha autenticado en la aplicación (distinción de roles). Dicha acción se propone para versiones posteriores.

2.6 Seguridad

La seguridad de la aplicación se basará en la introducción de un usuario-contraseña que será validado por el sistema, de forma que si los datos introducidos por el usuario son correctos se le dará acceso a la aplicación mientras que si son incorrectos se le denegará el acceso.

Cabe destacar que todas las clases *Action* de la aplicación heredarán de la clase *BaseAction* que en todo momento comprobarán que el usuario este validado en el sistema (comprobando si existe una sesión o no), de forma que si no se encuentra validado lo redireccionará a la pantalla inicial de *login*.

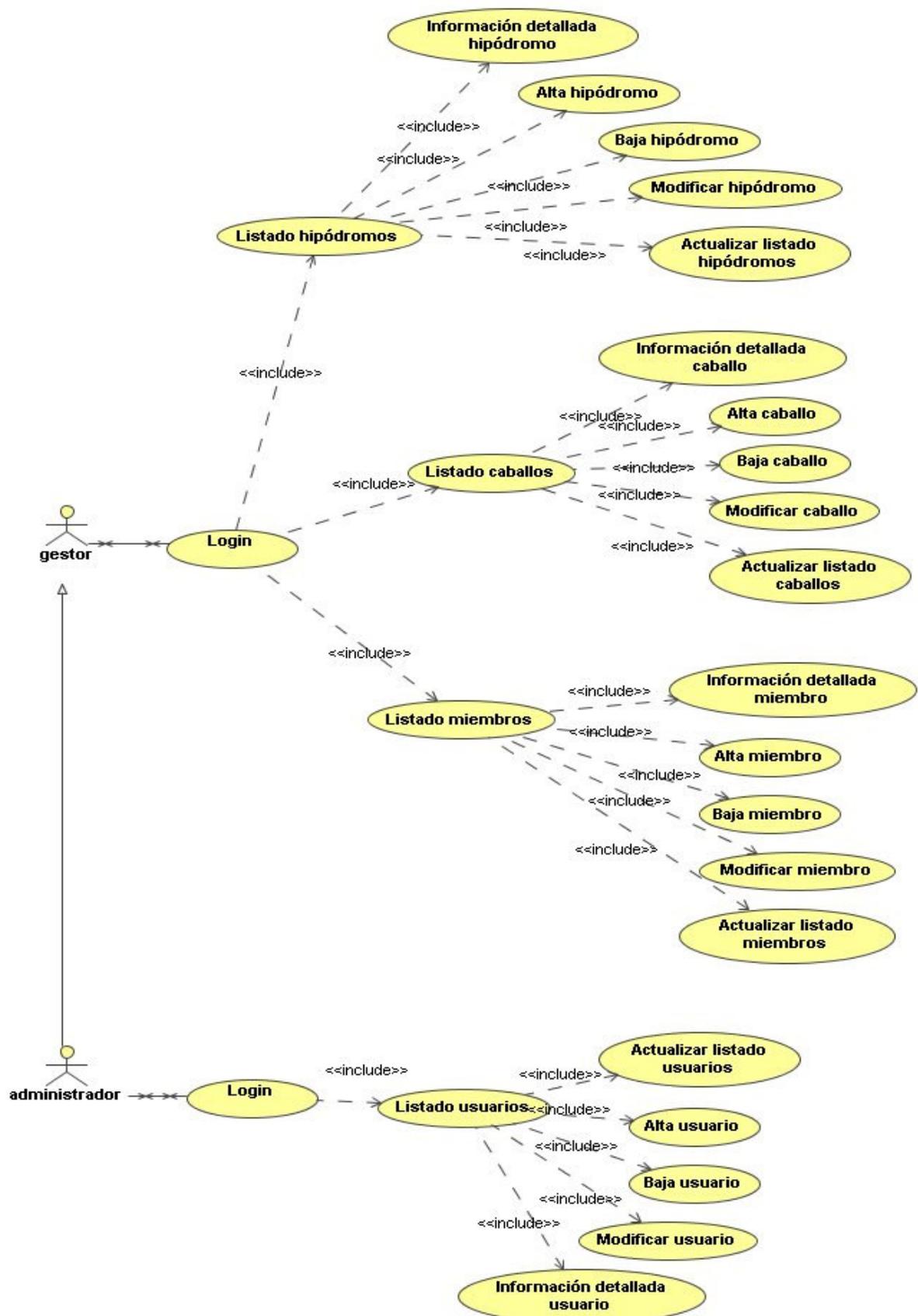
2.7 Diagramas de caso de uso por subsistema

Cada entidad de la aplicación da lugar a 6 casos de uso.

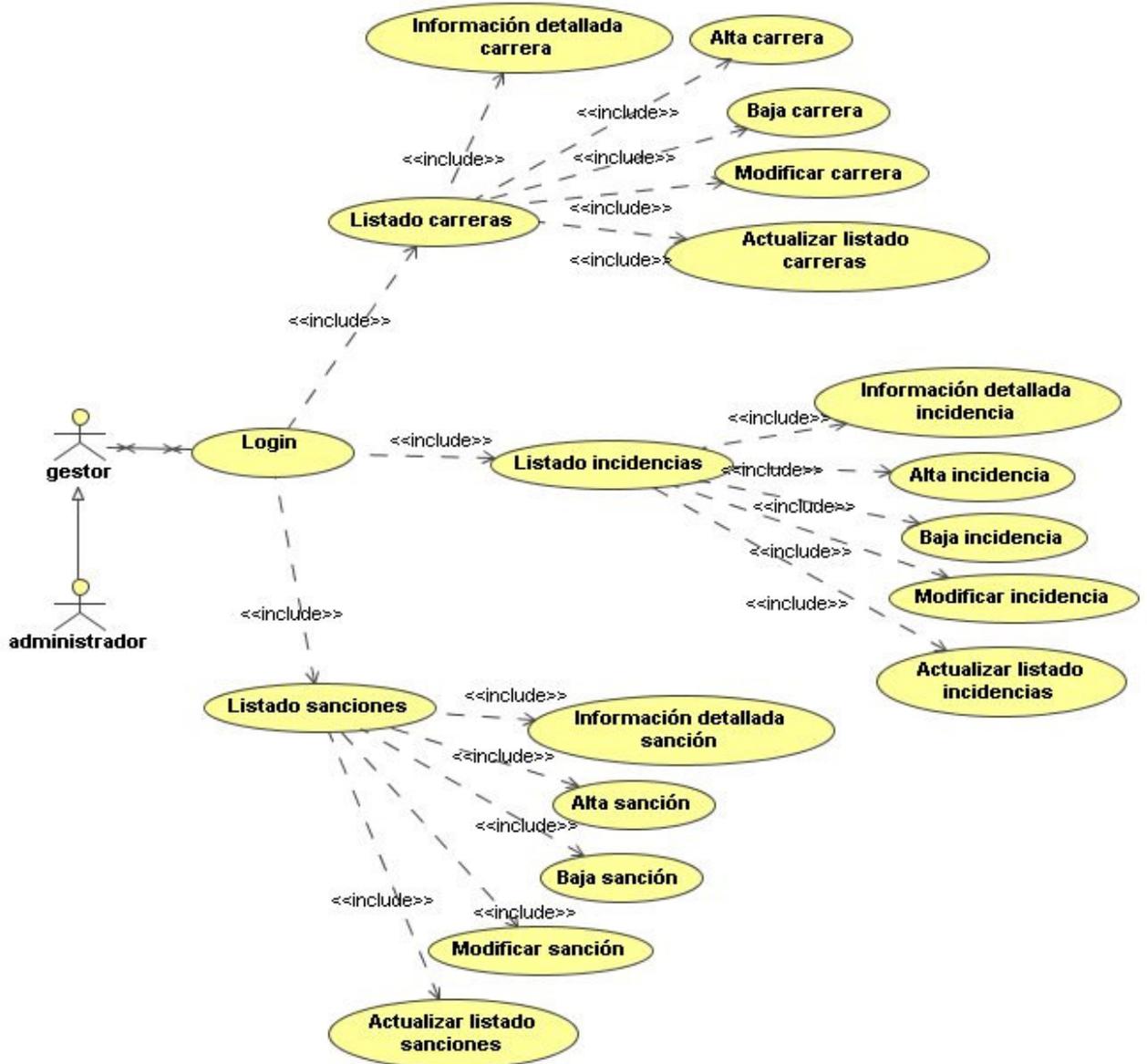
2.7.1 Subsistema de conexión



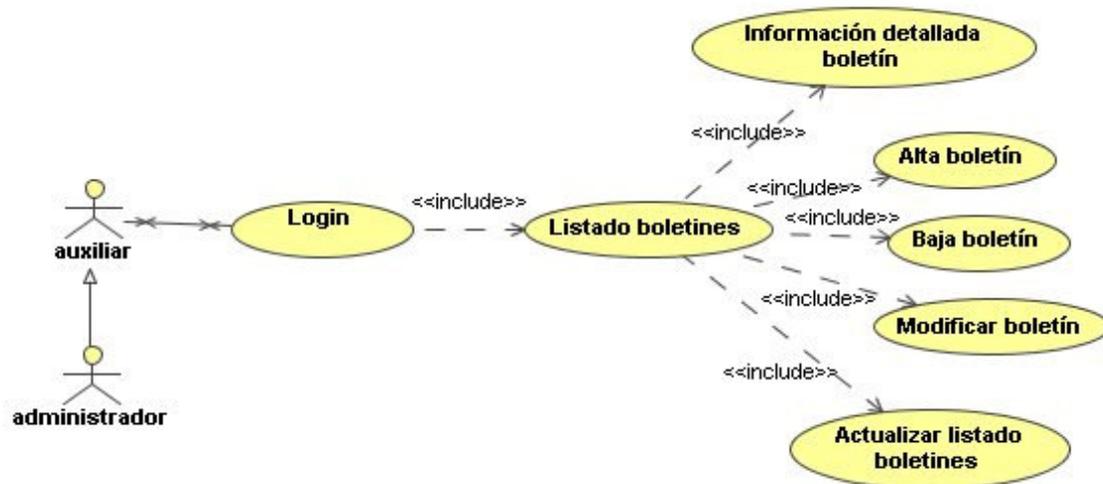
2.7.2 Subsistema de mantenimiento



2.7.3 Subsistema de gestión



2.7.4 Subsistema de información



2.8 Descripción de los casos de uso por subsistema

2.8.1 Subsistema de conexión

1. Caso de uso "Login"

Resumen de la funcionalidad: permite acceder a los usuarios a la aplicación.

Actores: *administrador, gestor, auxiliar.*

Casos de uso relacionados: ninguno (aunque se debe pasar por éste para acceder a todos los demás).

Precondición: el usuario es un *administrador, gestor* o *auxiliar*, está dado de alta en el sistema y su fecha de baja debe ser nula (*persona.FechaBaja = NULL*)

Poscondición: Al introducir el usuario-contraseña se accederá a la aplicación.

Descripción: el usuario debe introducir su usuario (identificado por su DNI) y su contraseña. El sistema validará los datos introducidos y si estos son correctos se le dará acceso a la aplicación. En caso contrario se le denegará el acceso.

2.8.2 Subsistema de mantenimiento

1. Caso de uso "Listado de hipódromos"

Resumen de la funcionalidad: permite visualizar por pantalla todos los hipódromos del sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: alta de hipódromos, baja de hipódromos, modificación de hipódromos, información detallada hipódromo, actualizar listado hipódromos

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El usuario seleccionará la opción de menú mantenimiento de hipódromos.

Poscondición: Se mostrará un listado de todos los hipódromos por pantalla.

Descripción: el usuario accederá al menú [mantenimiento] → [hipódromos] de forma que se mostrarán todos los hipódromos del sistema por pantalla. La ventana “Listado de hipódromos” nos permitirá acceder al alta, baja, modificación e información detallada de hipódromos. También permite refrescar la lista de hipódromos por si se han producido cambios en la BBDD.

2. Caso de uso “Alta de hipódromos”

Resumen de la funcionalidad: permite dar de alta un hipódromo en el sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de hipódromos

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El hipódromo a crear es identificado por su ID, debe ser válido y único en el sistema.

Poscondición: Si se han introducido todos los datos correctamente se dará de alta el hipódromo en el sistema; en caso contrario se informará del error ocurrido al usuario.

Descripción: una vez que se ha entrado en la ventana “Listado de hipódromos” accederemos a la ventana “Alta de hipódromos”. Deberemos introducir todos los datos solicitados y se seleccionará el botón “Guardar” de forma que se dará de alta el hipódromo en el sistema. Si se selecciona el botón “Atrás” se saldrá de la ventana “Alta de hipódromos” sin crear el hipódromo en el sistema.

3. Caso de uso “Baja de hipódromos”

Resumen de la funcionalidad: permite eliminar un hipódromo del sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de hipódromos

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El hipódromo a eliminar de la base de datos es identificado por su ID y debe existir en el sistema.

Poscondición: el hipódromo es eliminado del sistema.

Descripción: una vez que se ha entrado en la ventana “Listado de hipódromos” seleccionamos el hipódromo a eliminar, el sistema nos mostrará un mensaje por pantalla indicando si queremos eliminar el hipódromo seleccionado. En caso de aceptar se eliminará del sistema, en caso contrario no se eliminará.

4. Caso de uso “Modificar hipódromos”

Resumen de la funcionalidad: permite modificar los datos de un hipódromo a excepción de su identificador.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de hipódromos

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El hipódromo a modificar es identificado por su ID y debe existir en el sistema.

Poscondición: se guardan en el sistema los nuevos datos del hipódromo.

Descripción: una vez que se ha entrado en la ventana “Listado de hipódromos” seleccionamos el hipódromo a modificar de forma que se accede a la ventana “Modificación de hipódromos” y se modifican los datos deseados. Para confirmar los cambios se selecciona el botón “Guardar” volviendo a la ventana “Listado de hipódromos”. En el caso de que se seleccione el botón “Atrás” se saldrá de la ventana “Modificación de hipódromos” sin realizar ningún cambio.

5. Caso de uso “Información detallada del hipódromo”

Resumen de la funcionalidad: permite visualizar por pantalla todos los datos de un hipódromo.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de hipódromos

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El hipódromo a mostrar es identificado por su ID y debe existir en el sistema.

Poscondición: se muestran los datos del hipódromo.

Descripción: una vez que se ha entrado en la ventana “Listado de hipódromos” seleccionamos el identificador del hipódromo en cuestión de forma que se accede a la ventana “Información detallada”. Se visualiza en pantalla todos los datos asociados al hipódromo. Al seleccionar el botón “Atrás” volvemos a la ventana “Listado de hipódromos”.

6. Caso de uso “Actualizar listado hipódromos”

Resumen de la funcionalidad: permite actualizar la lista de hipódromos por si se ha producido algún cambio en la BBDD.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de hipódromos

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación.

Poscondición: se muestra nuevamente el listado de hipódromos.

Descripción: una vez que se ha entrado en la ventana “Listado de hipódromos” seleccionamos el icono en forma de *flecha circular* de forma que se cargan nuevamente en la lista los hipódromos del sistema.

7. Caso de uso “Listado de miembros”

Resumen de la funcionalidad: permite visualizar por pantalla todos los miembros del sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: alta de miembros, baja de miembros, modificación de miembros, información detallada miembro, actualizar listado miembros

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El usuario seleccionará la opción de menú mantenimiento de miembros.

Poscondición: Se mostrará un listado de todos los miembros por pantalla.

Descripción: el usuario accederá al menú [mantenimiento] → [miembros] de forma que se mostrarán todos los miembros del sistema por pantalla. La ventana “Listado de miembros” nos permitirá acceder al alta, baja, modificación e información detallada de miembros. También permite refrescar la lista de miembros por si se han producido cambios en la BBDD.

8. Caso de uso “Alta de miembros”

Resumen de la funcionalidad: permite dar de alta un miembro en el sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de miembros

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El miembro a crear es identificado por su DNI, debe ser válido y único en el sistema.

Poscondición: Si se han introducido todos los datos correctamente se dará de alta el miembro en el sistema; en caso contrario se informará del error ocurrido al usuario.

Descripción: una vez que se ha entrado en la ventana “Listado de miembros” accederemos a la ventana “Alta de miembros”. Debemos introducir todos los datos solicitados y se seleccionará el botón “Guardar” de forma que se dará de alta el miembro en el sistema. Si se selecciona el botón “Atrás” se saldrá de la ventana “Alta de miembros” sin crear el miembro en el sistema.

9. Caso de uso “Baja de miembros”

Resumen de la funcionalidad: permite eliminar un miembro del sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de miembros

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El miembro a eliminar de la base de datos es identificado por su DNI y debe existir en el sistema.

Poscondición: el miembro se da de baja del sistema.

Descripción: una vez que se ha entrado en la ventana “Listado de miembros” seleccionamos el miembro a eliminar, el sistema nos mostrará un mensaje por pantalla indicando si queremos eliminar el miembro seleccionado. En caso de aceptar se dará de baja el miembro del sistema, en caso contrario no se eliminará.

10. Caso de uso “Modificar miembros”

Resumen de la funcionalidad: permite modificar los datos de un miembro a excepción de su identificador (DNI) y carnet.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de miembros

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El miembro a modificar es identificado por su DNI y debe existir en el sistema.

Poscondición: se guardan en el sistema los nuevos datos del miembro.

Descripción: una vez que se ha entrado en la ventana “Listado de miembros” seleccionamos el miembro a modificar de forma que se accede a la ventana “Modificación de miembros” y se modifican los datos deseados. Para confirmar los cambios se selecciona el botón “Guardar” volviendo a la ventana “Listado de miembros”. En el caso de que se seleccione el botón “Atrás” se saldrá de la ventana “Modificación de miembros” sin realizar ningún cambio.

11. Caso de uso “Información detallada del miembro”

Resumen de la funcionalidad: permite visualizar por pantalla todos los datos de un miembro.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de miembros

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El miembro a mostrar es identificado por su DNI y debe existir en el sistema.

Poscondición: se muestran los datos del miembro.

Descripción: una vez que se ha entrado en la ventana “Listado de miembros” seleccionamos el identificador del miembro en cuestión de forma que se accede a la ventana “Información detallada”. Se visualiza en pantalla todos los datos asociados al miembro. Al seleccionar el botón “Atrás” volvemos a la ventana “Listado de miembros”.

12. Caso de uso “Actualizar listado miembros”

Resumen de la funcionalidad: permite actualizar la lista de miembros por si se ha producido algún cambio en la BBDD.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de miembros

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación.

Poscondición: se muestra nuevamente el listado de miembros.

Descripción: una vez que se ha entrado en la ventana “Listado de miembros” seleccionamos el icono en forma de *flecha circular* de forma que se cargan nuevamente en la lista los miembros del sistema.

13. Caso de uso “Listado de caballos”

Resumen de la funcionalidad: permite visualizar por pantalla todos los caballos del sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: alta de caballos, baja de caballos, modificación de caballos, información detallada caballo, actualizar listado caballos

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El usuario seleccionará la opción de menú mantenimiento de caballos.

Poscondición: Se mostrará un listado de todos los caballos por pantalla.

Descripción: el usuario accederá al menú [mantenimiento] → [caballos] de forma que se mostrarán todos los caballos del sistema por pantalla. La ventana “Listado de caballos” nos permitirá acceder al alta, baja, modificación e información detallada de caballos. También permite refrescar la lista de caballos por si se han producido cambios en la BBDD.

14. Caso de uso “Alta de caballos”

Resumen de la funcionalidad: permite dar de alta un caballo en el sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de caballos

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El caballo a crear es identificado por su ID, debe ser válido y único en el sistema.

Poscondición: Si se han introducido todos los datos correctamente se dará de alta el caballo en el sistema; en caso contrario se informará del error ocurrido al usuario.

Descripción: una vez que se ha entrado en la ventana “Listado de caballos” accederemos a la ventana “Alta de caballos”. Deberemos introducir todos los

datos solicitados y se seleccionará el botón “Guardar” de forma que se dará de alta el caballo en el sistema. Si se selecciona el botón “Atrás” se saldrá de la ventana “Alta de caballos” sin crear el caballo en el sistema.

15. Caso de uso “Baja de caballos”

Resumen de la funcionalidad: permite eliminar un caballo del sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de caballos

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El caballo a eliminar de la base de datos es identificado por su ID y debe existir en el sistema.

Poscondición: el caballo se da de baja del sistema.

Descripción: una vez que se ha entrado en la ventana “Listado de caballos” seleccionamos el caballo a eliminar, el sistema nos mostrará un mensaje por pantalla indicando si queremos eliminar el caballo seleccionado. En caso de aceptar se dará de baja el caballo del sistema, en caso contrario no se eliminará.

16. Caso de uso “Modificar caballos”

Resumen de la funcionalidad: permite modificar los datos de un caballo a excepción de su identificador.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de caballos

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El caballo a modificar es identificado por su ID y debe existir en el sistema.

Poscondición: se guardan en el sistema los nuevos datos del caballo.

Descripción: una vez que se ha entrado en la ventana “Listado de caballos” seleccionamos el caballo a modificar de forma que se accede a la ventana “Modificación de caballos” y se modifican los datos deseados. Para confirmar los cambios se selecciona el botón “Guardar” volviendo a la ventana “Listado de caballos”. En el caso de que se seleccione el botón “Atrás” se saldrá de la ventana “Modificación de caballos” sin realizar ningún cambio.

17. Caso de uso “Información detallada del caballo”

Resumen de la funcionalidad: permite visualizar por pantalla todos los datos de un caballo.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de caballos

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El caballo a mostrar es identificado por su ID y debe existir en el sistema.

Poscondición: se muestran los datos del caballo.

Descripción: una vez que se ha entrado en la ventana “Listado de caballos” seleccionamos el identificador del caballo de forma que se accede a la ventana “Información detallada”. Se visualiza en pantalla todos los datos asociados al caballo. Al seleccionar el botón “Atrás” volvemos a la ventana “Listado de caballos”.

18. Caso de uso “Actualizar listado caballos”

Resumen de la funcionalidad: permite actualizar la lista de caballos por si se ha producido algún cambio en la BBDD.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de caballos

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación.

Poscondición: se muestra nuevamente el listado de caballos.

Descripción: una vez que se ha entrado en la ventana “Listado de caballos” seleccionamos el icono en forma de *flecha circular* de forma que se cargan nuevamente en la lista los caballos del sistema.

19. Caso de uso “Listado de usuarios”

Resumen de la funcionalidad: permite visualizar por pantalla todos los usuarios del sistema.

Actores: *administrador*

Casos de uso relacionados: alta de usuarios, baja de usuarios, modificación de usuarios, información detallada de usuario, actualizar listado usuarios

Precondición: el usuario es *administrador* y se ha identificado en la aplicación. El usuario seleccionará la opción de menú mantenimiento de usuarios.

Poscondición: Se mostrará un listado de todos los usuarios por pantalla.

Descripción: el usuario accederá al menú [mantenimiento] → [usuarios] de forma que se mostrarán todos los usuarios del sistema por pantalla. La ventana “Listado de usuarios” nos permitirá acceder al alta, baja, modificación e información detallada de usuarios. También permite refrescar la lista de usuarios por si se han producido cambios en la BBDD.

20. Caso de uso “Alta de usuarios”

Resumen de la funcionalidad: permite dar de alta un usuario en el sistema.

Actores: *administrador*

Casos de uso relacionados: listado de usuarios

Precondición: el usuario es *administrador* y se ha identificado en la aplicación. El usuario a crear es identificado por su DNI, debe ser válido y único en el sistema.

Poscondición: Si se han introducido todos los datos correctamente se dará de alta el usuario en el sistema; en caso contrario se informará del error ocurrido al usuario.

Descripción: una vez que se ha entrado en la ventana “Listado de usuarios” accederemos a la ventana “Alta de usuarios”. Deberemos introducir todos los datos solicitados y se seleccionará el botón “Guardar” de forma que se dará de alta el usuario en el sistema. Si se selecciona el botón “Atrás” se saldrá de la ventana “Alta de usuarios” sin crear el usuario en el sistema.

NOTA: para una mayor seguridad el campo “password” del usuario se insertará en blanco en la base de datos, de forma que sea el administrador de la aplicación el que deberá especificar la contraseña (por ejemplo directamente desde el MySQL) y comunicársela al usuario por escrito

21. Caso de uso “Baja de usuarios”

Resumen de la funcionalidad: permite eliminar un usuario del sistema.

Actores: *administrador*

Casos de uso relacionados: listado de usuarios

Precondición: el usuario es *administrador* y se ha identificado en la aplicación. El usuario a eliminar de la base de datos es identificado por su DNI y debe existir en el sistema.

Poscondición: el usuario se da de baja del sistema.

Descripción: una vez que se ha entrado en la ventana “Listado de usuarios” seleccionamos el usuario a eliminar, el sistema nos mostrará un mensaje por pantalla indicando si queremos eliminar el usuario seleccionado. En caso de aceptar se dará de baja el usuario del sistema, en caso contrario no se eliminará.

22. Caso de uso “Modificar usuarios”

Resumen de la funcionalidad: permite modificar los datos de un usuario a excepción de su identificador: DNI.

Actores: *administrador*

Casos de uso relacionados: listado de usuarios

Precondición: el usuario es *administrador* y se ha identificado en la aplicación. El usuario a modificar es identificado por su DNI y debe existir en el sistema.

Poscondición: se guardan en el sistema los nuevos datos del usuario.

Descripción: una vez que se ha entrado en la ventana “Listado de usuarios” seleccionamos el usuario a modificar de forma que se accede a la ventana

“Modificación de usuarios” y se modifican los datos deseados. Para confirmar los cambios se selecciona el botón “Guardar” volviendo a la ventana “Listado de usuarios”. En el caso de que se seleccione el botón “Atrás” se saldrá de la ventana “Modificación de caballos” sin realizar ningún cambio.

23. Caso de uso “Información detallada del usuario”

Resumen de la funcionalidad: permite visualizar por pantalla todos los datos de un usuario.

Actores: *administrador*

Casos de uso relacionados: listado de usuarios

Precondición: el usuario es un *administrador* y se ha identificado en la aplicación. El usuario a mostrar es identificado por su DNI y debe existir en el sistema.

Poscondición: se muestran los datos del usuario.

Descripción: una vez que se ha entrado en la ventana “Listado de usuarios” seleccionamos el identificador del usuario de forma que se accede a la ventana “Información detallada”. Se visualiza en pantalla todos los datos asociados al usuario. Al seleccionar el botón “Atrás” volvemos a la ventana “Listado de usuarios”.

24. Caso de uso “Actualizar listado usuarios”

Resumen de la funcionalidad: permite actualizar la lista de usuarios por si se ha producido algún cambio en la BBDD.

Actores: *administrador*

Casos de uso relacionados: listado de usuarios

Precondición: el usuario es un *administrador* y se ha identificado en la aplicación.

Poscondición: se muestra nuevamente el listado de usuarios.

Descripción: una vez que se ha entrado en la ventana “Listado de usuarios” seleccionamos el icono en forma de *flecha circular* de forma que se cargan nuevamente en la lista los usuarios del sistema.

2.8.3 Subsistema de gestión

1. Caso de uso “Listado de carreras”

Resumen de la funcionalidad: permite visualizar por pantalla todas las carreras del sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: alta de carreras, baja de carreras, modificación de carreras, información detallada carrera, actualizar listado carreras

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El usuario seleccionará la opción de menú gestión de carreras.

Poscondición: Se mostrará un listado de todas las carreras por pantalla.

Descripción: el usuario accederá al menú [gestión] → [carreras] de forma que se mostrarán todas las carreras del sistema por pantalla. La ventana “Listado de carreras” nos permitirá acceder al alta, baja, modificación e información detallada de carreras. También permite refrescar la lista de carreras por si se han producido cambios en la BBDD.

2. Caso de uso “Alta de carreras”

Resumen de la funcionalidad: permite dar de alta una carrera en el sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de carreras

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. La carrera a crear es identificada por su ID, debe ser válida y única en el sistema.

Poscondición: Si se han introducido todos los datos correctamente se dará de alta la carrera en el sistema; en caso contrario se informará del error ocurrido al usuario.

Descripción: una vez que se ha entrado en la ventana “Listado de carreras” accederemos a la ventana “Alta de carreras”. Debemos introducir todos los datos solicitados y se seleccionará el botón “Guardar” de forma que se dará de alta la carrera en el sistema. Si se selecciona el botón “Atrás” se saldrá de la ventana “Alta de carreras” sin crear la carrera en el sistema.

3. Caso de uso “Baja de carreras”

Resumen de la funcionalidad: permite eliminar una carrera del sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de carreras

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. La carrera a eliminar de la base de datos es identificada por su ID y debe existir en el sistema.

Poscondición: la carrera es eliminada del sistema.

Descripción: una vez que se ha entrado en la ventana “Listado de carreras” seleccionamos la carrera a eliminar, el sistema nos mostrará un mensaje por pantalla indicando si queremos eliminar la carrera seleccionada. En caso de aceptar se eliminará del sistema, en caso contrario no se eliminará.

4. Caso de uso “Modificar carrera”

Resumen de la funcionalidad: permite modificar los datos de una carrera a excepción de su identificador.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de carreras

Precondición: el usuario es un *administrador o gestor* y se ha identificado en la aplicación. La carrera a modificar es identificada por su ID y debe existir en el sistema.

Poscondición: se guardan en el sistema los nuevos datos de la carrera.

Descripción: una vez que se ha entrado en la ventana “Listado de carreras” seleccionamos la carrera a modificar de forma que se accede a la ventana “Modificación de carreras” y se modifican los datos deseados. Para confirmar los cambios se selecciona el botón “Guardar” volviendo a la ventana “Listado de carreras”. En el caso de que se seleccione el botón “Atrás” se saldrá de la ventana “Modificación de carreras” sin realizar ningún cambio.

5. Caso de uso “Información detallada carrera”

Resumen de la funcionalidad: permite visualizar por pantalla todos los datos de una carrera.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de carreras

Precondición: el usuario es un *administrador o gestor* y se ha identificado en la aplicación. La carrera a mostrar es identificada por su ID y debe existir en el sistema.

Poscondición: se muestran los datos de la carrera.

Descripción: una vez que se ha entrado en la ventana “Listado de carreras” seleccionamos el identificador de la carrera de forma que se accede a la ventana “Información detallada”. Se visualiza en pantalla todos los datos asociados a la carrera. Al seleccionar el botón “Atrás” volvemos a la ventana “Listado de carreras”.

6. Caso de uso “Actualizar listado carreras”

Resumen de la funcionalidad: permite actualizar la lista de carreras por si se ha producido algún cambio en la BBDD.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de carreras

Precondición: el usuario es un *administrador o gestor* y se ha identificado en la aplicación.

Poscondición: se muestra nuevamente el listado de carreras.

Descripción: una vez que se ha entrado en la ventana “Listado de carreras” seleccionamos el icono en forma de *flecha circular* de forma que se cargan nuevamente en la lista las carreras del sistema.

7. Caso de uso “Listado de incidencias”

Resumen de la funcionalidad: permite visualizar por pantalla todas las incidencias del sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: alta de incidencias, baja de incidencias, modificación de incidencias, información detallada incidencia, actualizar listado incidencias

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El usuario seleccionará la opción de menú gestión de incidencias.

Poscondición: Se mostrará un listado de todas las incidencias por pantalla.

Descripción: el usuario accederá al menú [gestión] → [incidencias] de forma que se mostrarán todas las incidencias del sistema por pantalla. La ventana “Listado de incidencias” nos permitirá acceder al alta, baja, modificación e información detallada de incidencias. También permite refrescar la lista de incidencias por si se han producido cambios en la BBDD.

8. Caso de uso “Alta de incidencias”

Resumen de la funcionalidad: permite dar de alta una incidencia en el sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de incidencias

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. La incidencia a crear es identificada por su ID, debe ser válida y única en el sistema.

Poscondición: Si se han introducido todos los datos correctamente se dará de alta la incidencia en el sistema; en caso contrario se informará del error ocurrido al usuario.

Descripción: una vez que se ha entrado en la ventana “Listado de incidencias” accederemos a la ventana “Alta de incidencias”. Debemos introducir todos los datos solicitados y se seleccionará el botón “Guardar” de forma que se dará de alta la incidencia en el sistema. Si se selecciona el botón “Atrás” se saldrá de la ventana “Alta de incidencias” sin crear la incidencia en el sistema.

9. Caso de uso “Baja de incidencias”

Resumen de la funcionalidad: permite eliminar una incidencia del sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de incidencias

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. La incidencia a eliminar de la base de datos es identificada por su ID y debe existir en el sistema.

Poscondición: la incidencia es eliminada del sistema.

Descripción: una vez que se ha entrado en la ventana “Listado de incidencias” seleccionamos la incidencia a eliminar, el sistema nos mostrará un mensaje por pantalla indicando si queremos eliminar la incidencia seleccionada. En caso de aceptar se eliminará del sistema, en caso contrario no se eliminará.

10. Caso de uso “Modificar incidencia”

Resumen de la funcionalidad: permite modificar los datos de una incidencia a excepción de su identificador.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de incidencias

Precondición: el usuario es un *administrador o gestor* y se ha identificado en la aplicación. La incidencia a modificar es identificada por su ID y debe existir en el sistema.

Poscondición: se guardan en el sistema los nuevos datos de la incidencia.

Descripción: una vez que se ha entrado en la ventana “Listado de incidencias” seleccionamos la incidencia a modificar de forma que se accede a la ventana “Modificación de incidencias” y se modifican los datos deseados. Para confirmar los cambios se selecciona el botón “Guardar” volviendo a la ventana “Listado de incidencias”. En el caso de que se seleccione el botón “Atrás” se saldrá de la ventana “Modificación de incidencias” sin realizar ningún cambio.

11. Caso de uso “Información detallada incidencia”

Resumen de la funcionalidad: permite visualizar por pantalla todos los datos de una incidencia.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de incidencias

Precondición: el usuario es un *administrador o gestor* y se ha identificado en la aplicación. La incidencia a mostrar es identificada por su ID y debe existir en el sistema.

Poscondición: se muestran los datos de la incidencia.

Descripción: una vez que se ha entrado en la ventana “Listado de incidencias” seleccionamos el identificador de la incidencia de forma que se accede a la ventana “Información detallada”. Se visualiza en pantalla todos los datos asociados a la incidencia. Al seleccionar el botón “Atrás” volvemos a la ventana “Listado de incidencias”.

12. Caso de uso “Actualizar listado incidencias”

Resumen de la funcionalidad: permite actualizar la lista de incidencias por si se ha producido algún cambio en la BBDD.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de incidencias

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación.

Poscondición: se muestra nuevamente el listado de incidencias.

Descripción: una vez que se ha entrado en la ventana “Listado de incidencias” seleccionamos el icono en forma de *flecha circular* de forma que se cargan nuevamente en la lista las incidencias del sistema.

13. Caso de uso “Listado de sanciones”

Resumen de la funcionalidad: permite visualizar por pantalla todas las sanciones del sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: alta de sanciones, baja de sanciones, modificación de sanciones, información detallada sanción, actualizar listado sanciones

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. El usuario seleccionará la opción de menú gestión de sanciones.

Poscondición: Se mostrará un listado de todas las sanciones por pantalla.

Descripción: el usuario accederá al menú [gestión] → [sanciones] de forma que se mostrarán todas las sanciones del sistema por pantalla. La ventana “Listado de sanciones” nos permitirá acceder al alta, baja, modificación e información detallada de sanciones. También permite refrescar la lista de sanciones por si se han producido cambios en la BBDD.

14. Caso de uso “Alta de sanciones”

Resumen de la funcionalidad: permite dar de alta una sanción en el sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de sanciones

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. La sanción a crear es identificada por su ID, debe ser válida y única en el sistema.

Poscondición: Si se han introducido todos los datos correctamente se dará de alta la sanción en el sistema; en caso contrario se informará del error ocurrido al usuario.

Descripción: una vez que se ha entrado en la ventana “Listado de sanciones” accederemos a la ventana “Alta de sanciones”. Deberemos introducir todos los datos solicitados y se seleccionará el botón “Guardar” de forma que se dará de alta la sanción en el sistema. Si se selecciona el botón “Atrás” se saldrá de la ventana “Alta de sanciones” sin crear la sanción en el sistema.

15. Caso de uso “Baja de sanciones”

Resumen de la funcionalidad: permite eliminar una sanción del sistema.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de sanciones

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. La sanción a eliminar de la base de datos es identificada por su ID y debe existir en el sistema.

Poscondición: la sanción es eliminada del sistema.

Descripción: una vez que se ha entrado en la ventana “Listado de sanciones” seleccionamos la sanción a eliminar, el sistema nos mostrará un mensaje por pantalla indicando si queremos eliminar la sanción seleccionada. En caso de aceptar se eliminará del sistema, en caso contrario no se eliminará.

16. Caso de uso “Modificar sanción”

Resumen de la funcionalidad: permite modificar los datos de una sanción a excepción de su identificador.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de sanciones

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. La sanción a modificar es identificada por su ID y debe existir en el sistema.

Poscondición: se guardan en el sistema los nuevos datos de la sanción.

Descripción: una vez que se ha entrado en la ventana “Listado de sanciones” seleccionamos la sanción a modificar de forma que se accede a la ventana “Modificación de sanciones” y se modifican los datos deseados. Para confirmar los cambios se selecciona el botón “Guardar” volviendo a la ventana “Listado de sanciones”. En el caso de que se seleccione el botón “Atrás” se saldrá de la ventana “Modificación de sanciones” sin realizar ningún cambio.

17. Caso de uso “Información detallada sanción”

Resumen de la funcionalidad: permite visualizar por pantalla todos los datos de una sanción.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de sanciones

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación. La sanción a mostrar es identificada por su ID y debe existir en el sistema.

Poscondición: se muestran los datos de la sanción

Descripción: una vez que se ha entrado en la ventana “Listado de sanciones” seleccionamos el identificador de la sanción de forma que se accede a la ventana “Información detallada”. Se visualiza en pantalla todos los datos asociados a la sanción. Al seleccionar el botón “Atrás” volvemos a la ventana “Listado de sanciones”.

18. Caso de uso “Actualizar listado sanciones”

Resumen de la funcionalidad: permite actualizar la lista de sanciones por si se ha producido algún cambio en la BBDD.

Actores: *administrador, gestor*

Casos de uso relacionados: listado de sanciones

Precondición: el usuario es un *administrador* o *gestor* y se ha identificado en la aplicación.

Poscondición: se muestra nuevamente el listado de sanciones.

Descripción: una vez que se ha entrado en la ventana “Listado de sanciones” seleccionamos el icono en forma de *flecha circular* de forma que se cargan nuevamente en la lista las sanciones del sistema.

2.8.4 Subsistema de información

1. Caso de uso “Listado de boletines”

Resumen de la funcionalidad: permite visualizar por pantalla todos los boletines del sistema.

Actores: *administrador, auxiliar*

Casos de uso relacionados: alta de boletines, baja de boletines, modificación de boletines, información detallada boletín, actualizar listado boletines

Precondición: el usuario es *administrador* o *auxiliar* y se ha identificado en la aplicación. El usuario seleccionará la opción de menú boletines.

Poscondición: Se mostrará un listado de todos los boletines por pantalla.

Descripción: el usuario accederá al menú [boletines] → [listado boletines] de forma que se mostrarán todos los boletines del sistema por pantalla. La ventana “Listado de boletines” nos permitirá acceder al alta, baja, modificación e información detallada de boletines. También permite refrescar la lista de boletines por si se han producido cambios en la BBDD.

2. Caso de uso “Alta de boletines”

Resumen de la funcionalidad: permite dar de alta un boletín en el sistema.

Actores: *administrador, auxiliar*

Casos de uso relacionados: listado de boletines

Precondición: el usuario es *administrador* o *auxiliar* y se ha identificado en la aplicación. El boletín a crear es identificado por su ID, debe ser válido y único en el sistema.

Poscondición: Si se han introducido todos los datos correctamente se dará de alta el boletín en el sistema; en caso contrario se informará del error ocurrido al usuario.

Descripción: una vez que se ha entrado en la ventana “Listado de boletines” accederemos a la ventana “Alta de boletines”. Deberemos introducir todos los datos solicitados y se seleccionará el botón “Guardar” de forma que se dará de alta el boletín en el sistema. Si se selecciona el botón “Atrás” se saldrá de la ventana “Alta de boletines” sin crear el boletín en el sistema.

3. Caso de uso “Baja de boletines”

Resumen de la funcionalidad: permite eliminar un boletín del sistema.

Actores: *administrador, auxiliar*

Casos de uso relacionados: listado de boletines

Precondición: el usuario es *administrador o auxiliar* y se ha identificado en la aplicación. El boletín a eliminar de la base de datos es identificado por su ID y debe existir en el sistema.

Poscondición: el boletín se elimina del sistema.

Descripción: una vez que se ha entrado en la ventana “Listado de boletines” seleccionamos el boletín a eliminar, el sistema nos mostrará un mensaje por pantalla indicando si queremos eliminar el boletín seleccionado. En caso de aceptar se eliminará el boletín del sistema, en caso contrario no se eliminará.

4. Caso de uso “Modificar boletines”

Resumen de la funcionalidad: permite modificar los datos de un boletín a excepción de su identificador.

Actores: *administrador, auxiliar*

Casos de uso relacionados: listado de boletines

Precondición: el usuario es *administrador o auxiliar* y se ha identificado en la aplicación. El boletín a modificar es identificado por su ID y debe existir en el sistema.

Poscondición: se guardan en el sistema los nuevos datos del boletín.

Descripción: una vez que se ha entrado en la ventana “Listado de boletines” seleccionamos el boletín a modificar de forma que se accede a la ventana “Modificación de boletines” y se modifican los datos deseados. Para confirmar los cambios se selecciona el botón “Guardar” volviendo a la ventana “Listado de boletines”. En el caso de que se seleccione el botón “Atrás” se saldrá de la ventana “Modificación de boletines” sin realizar ningún cambio.

5. Caso de uso “Información detallada boletín”

Resumen de la funcionalidad: permite visualizar por pantalla todos los datos de un boletín.

Actores: *administrador, auxiliar*

Casos de uso relacionados: listado de boletines

Precondición: el usuario es un *administrador* o *auxiliar* y se ha identificado en la aplicación. El boletín a mostrar es identificado por su ID y debe existir en el sistema.

Poscondición: se muestran los datos del boletín.

Descripción: una vez que se ha entrado en la ventana “Listado de boletines” seleccionamos el identificador del boletín de forma que se accede a la ventana “Información detallada”. Se visualiza en pantalla todos los datos asociados al boletín. Al seleccionar el botón “Atrás” volvemos a la ventana “Listado de boletines”.

6. Caso de uso “Actualizar listado boletines”

Resumen de la funcionalidad: permite actualizar la lista de boletines por si se ha producido algún cambio en la BBDD.

Actores: *administrador, auxiliar*

Casos de uso relacionados: listado de boletines

Precondición: el usuario es un *administrador* o *auxiliar* y se ha identificado en la aplicación.

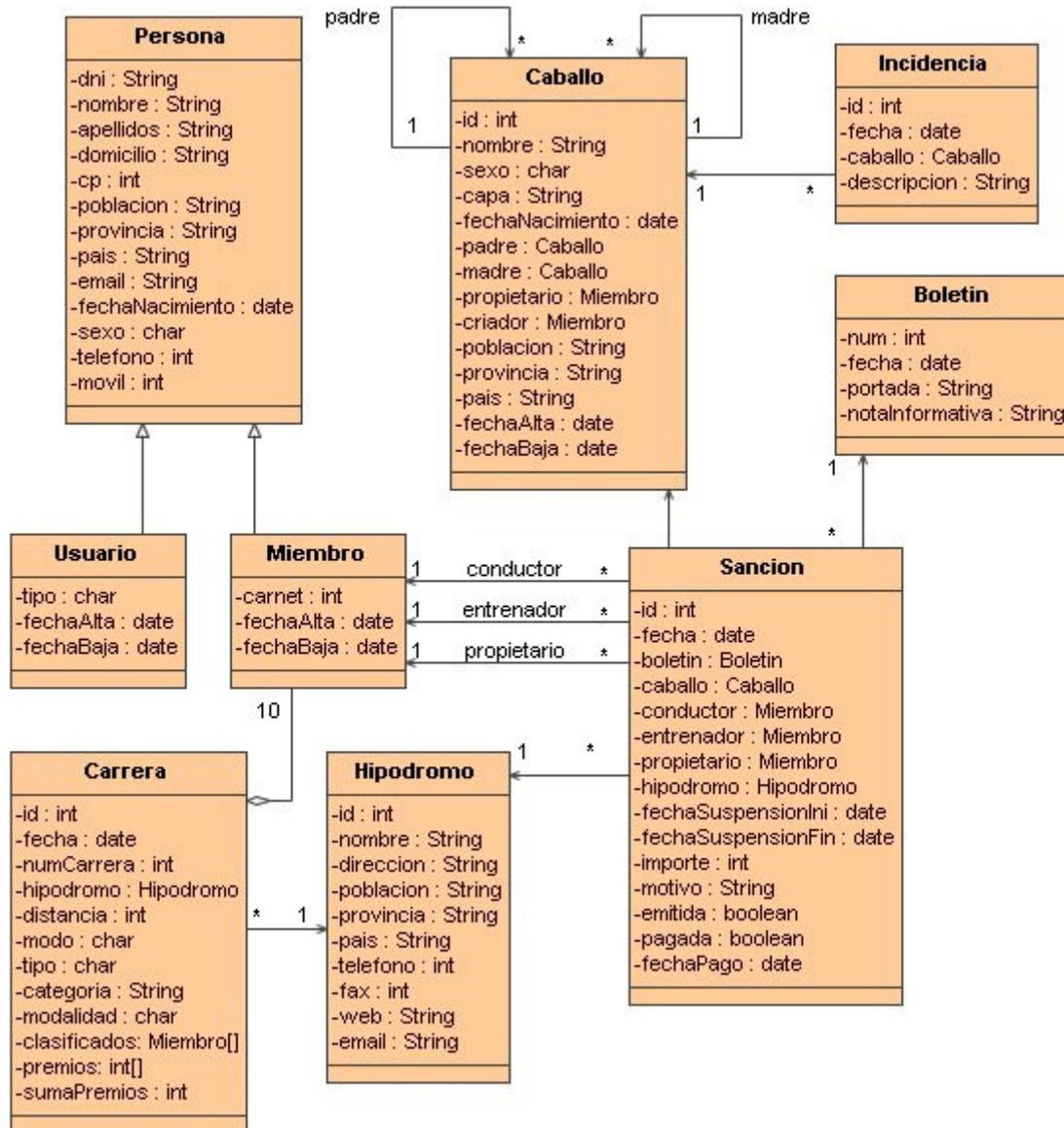
Poscondición: se muestra nuevamente el listado de boletines.

Descripción: una vez que se ha entrado en la ventana “Listado de boletines” seleccionamos el icono en forma de *flecha circular* de forma que se cargan nuevamente en la lista los boletines del sistema.

3. DISEÑO

3.1 Diagrama de clases

3.1.1 Diagrama de entidades

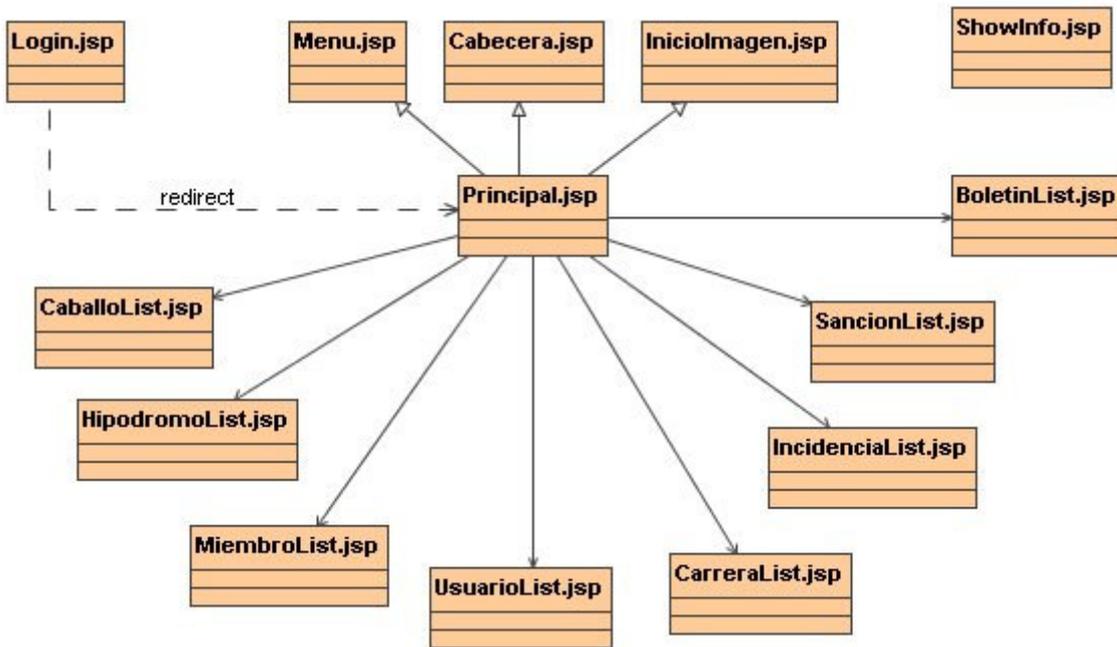


3.1.2 Diagrama de jerarquías

Todas las clases especificadas a continuación (a excepción de las clases *Login.jsp* y *ShowInfo.jsp*) incluyen las JSPs *Menu.jsp* y *Cabecera.jsp* aunque no se hayan indicado específicamente. Esta acción se ha realizado con el objetivo de tener una visión más clara de la jerarquía de pantallas de la aplicación.

La clase *ShowInfo.jsp* tiene relación con todas las clases JSP de la aplicación, ya que servirá para mostrar los mensajes de información o error que se produzcan en la aplicación.

3.1.2.1 Ventanas principales de la aplicación

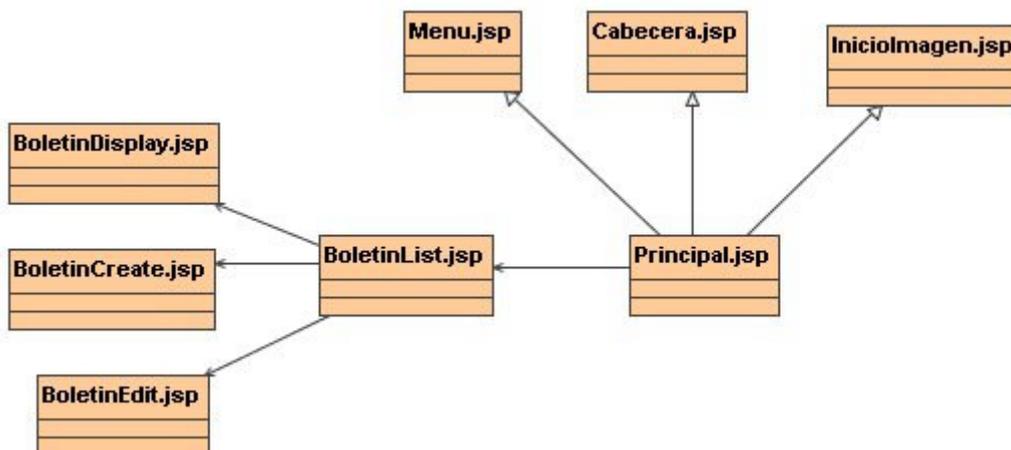


3.1.2.2 Ventanas de la aplicación por subsistema

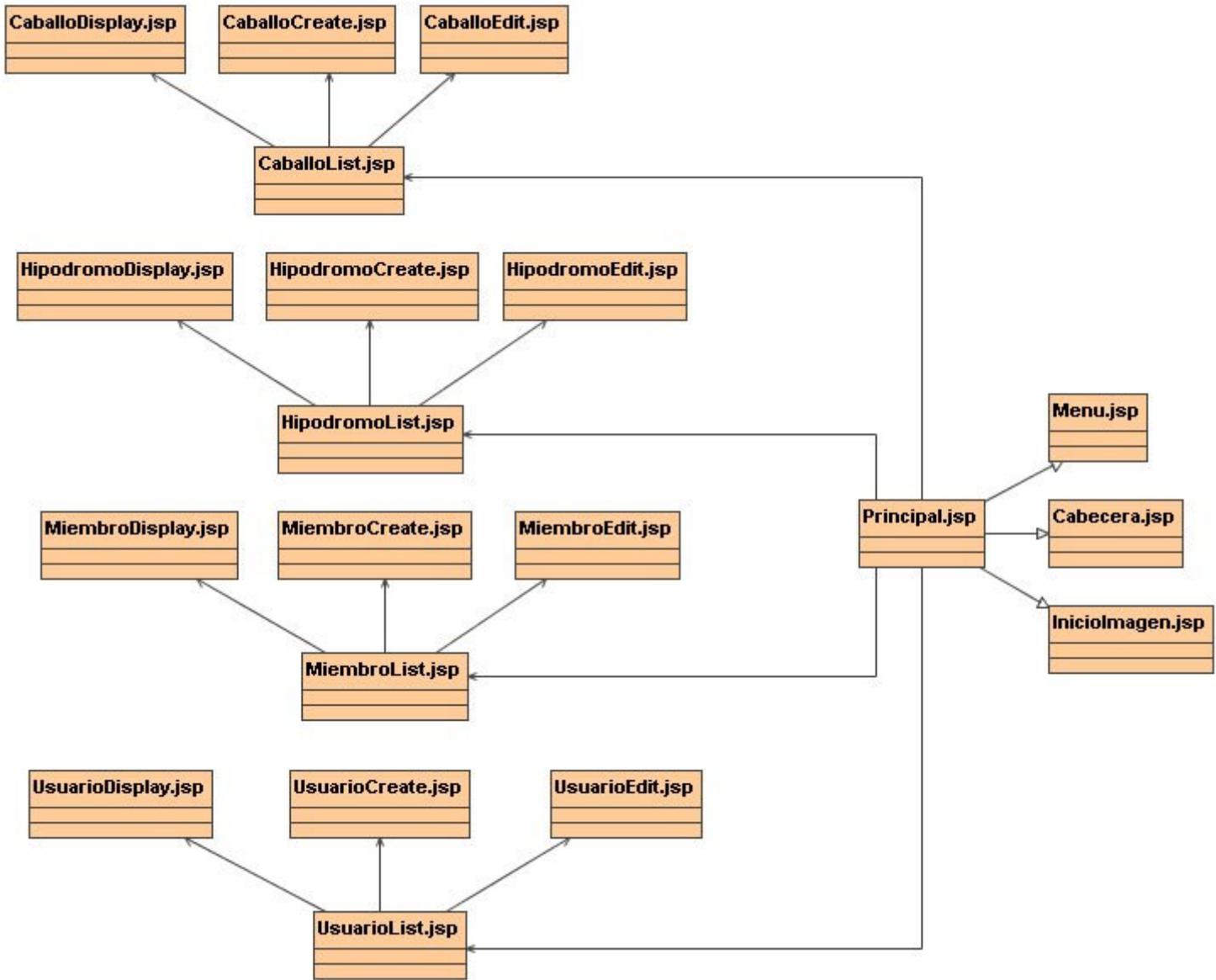
Subsistema de conexión



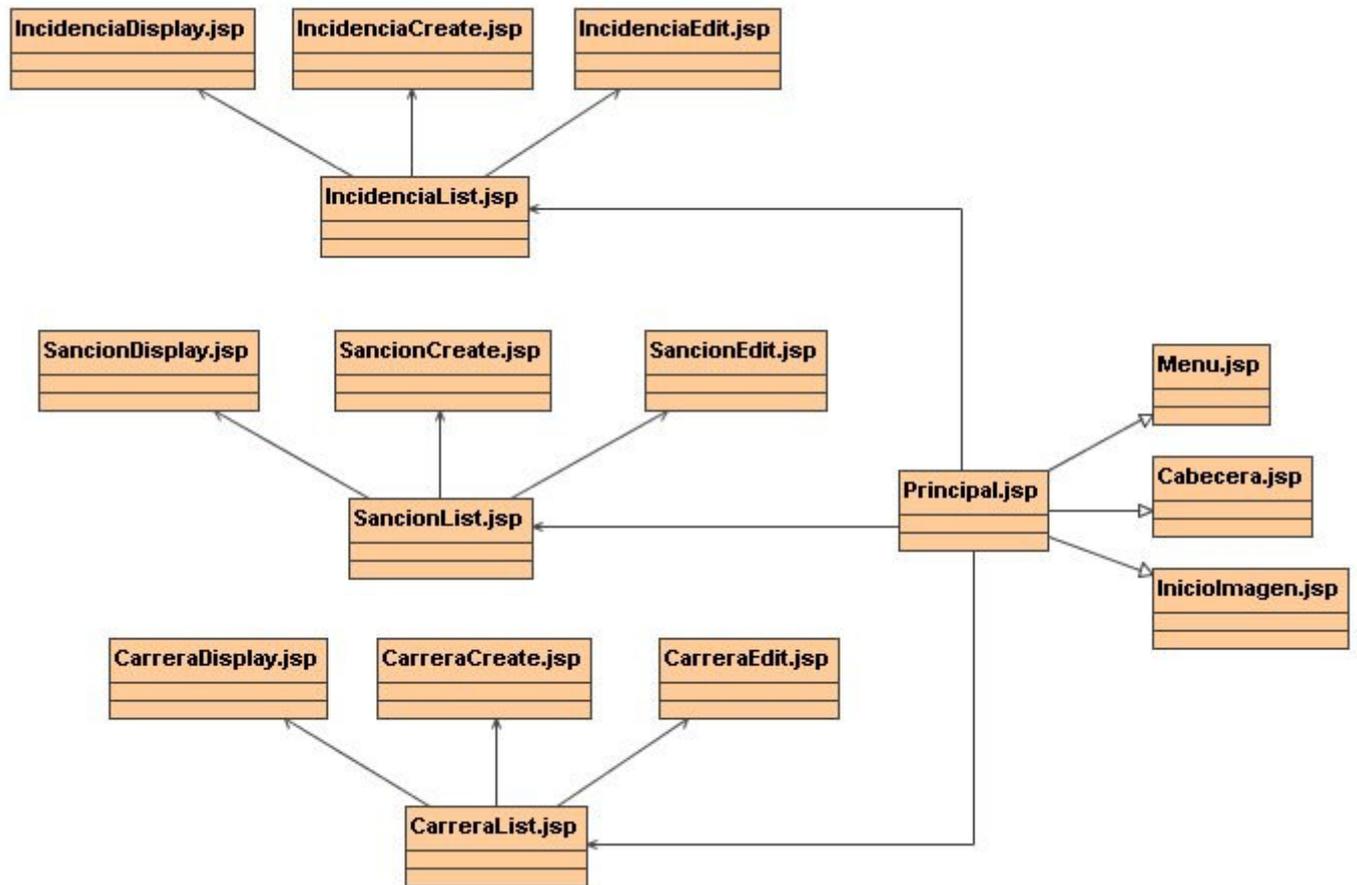
Subsistema de información



Subsistema de mantenimiento



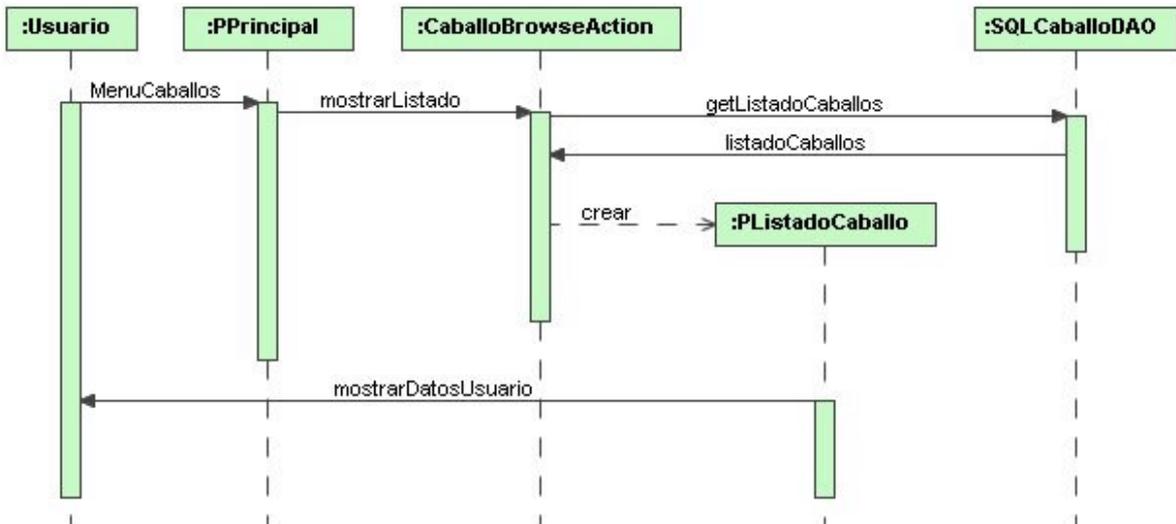
Subsistema de gestión



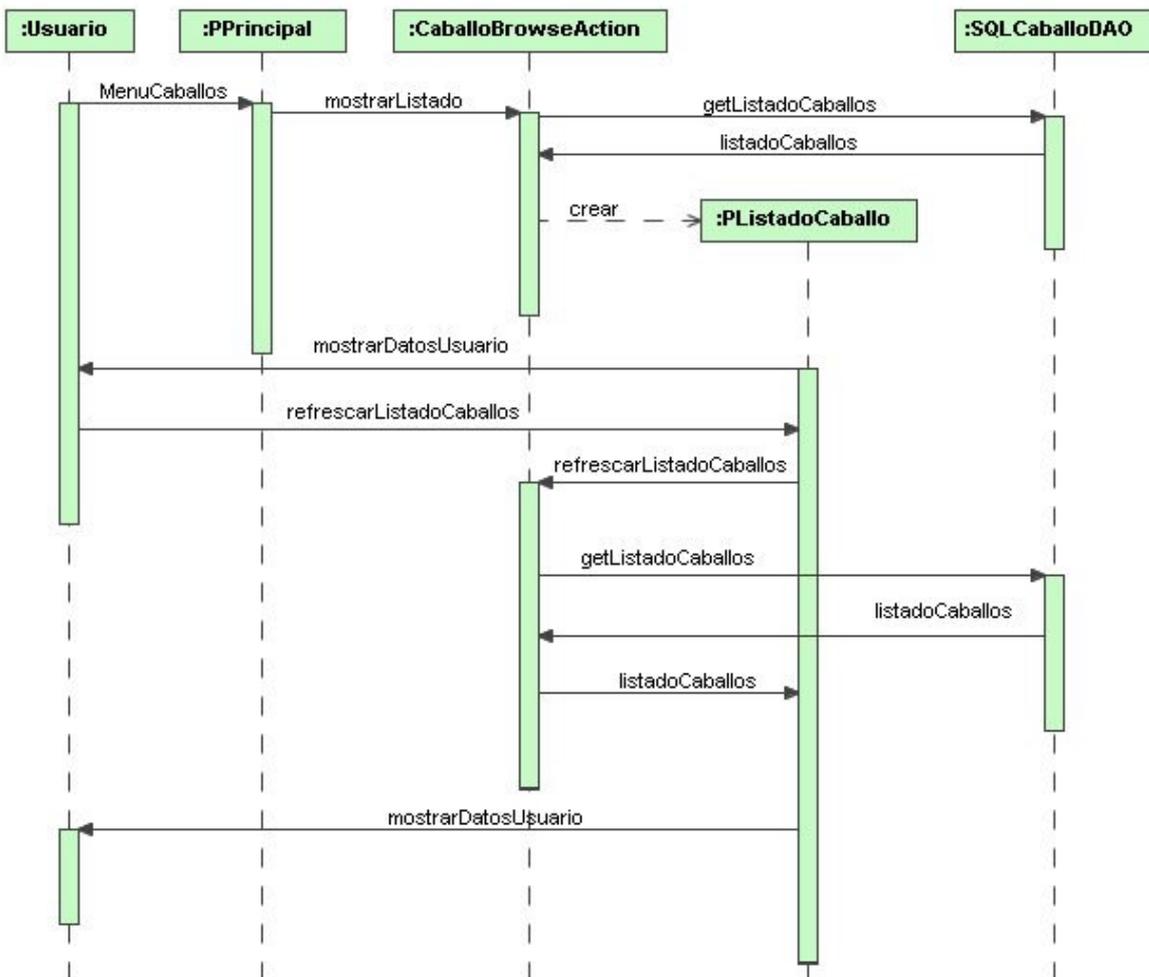
3.2 Diagramas de secuencia

Debido al gran número de casos de uso que tiene la aplicación, un total de 6 casos de uso por las 8 entidades que tiene la aplicación hacen un total de 48 casos de uso, solo se especificarán los diagramas de secuencia asociados a los casos de uso de la entidad *Caballo*. A partir de éstos nos podemos hacer una idea de los diagramas de secuencia asociados a los casos de uso de las demás entidades.

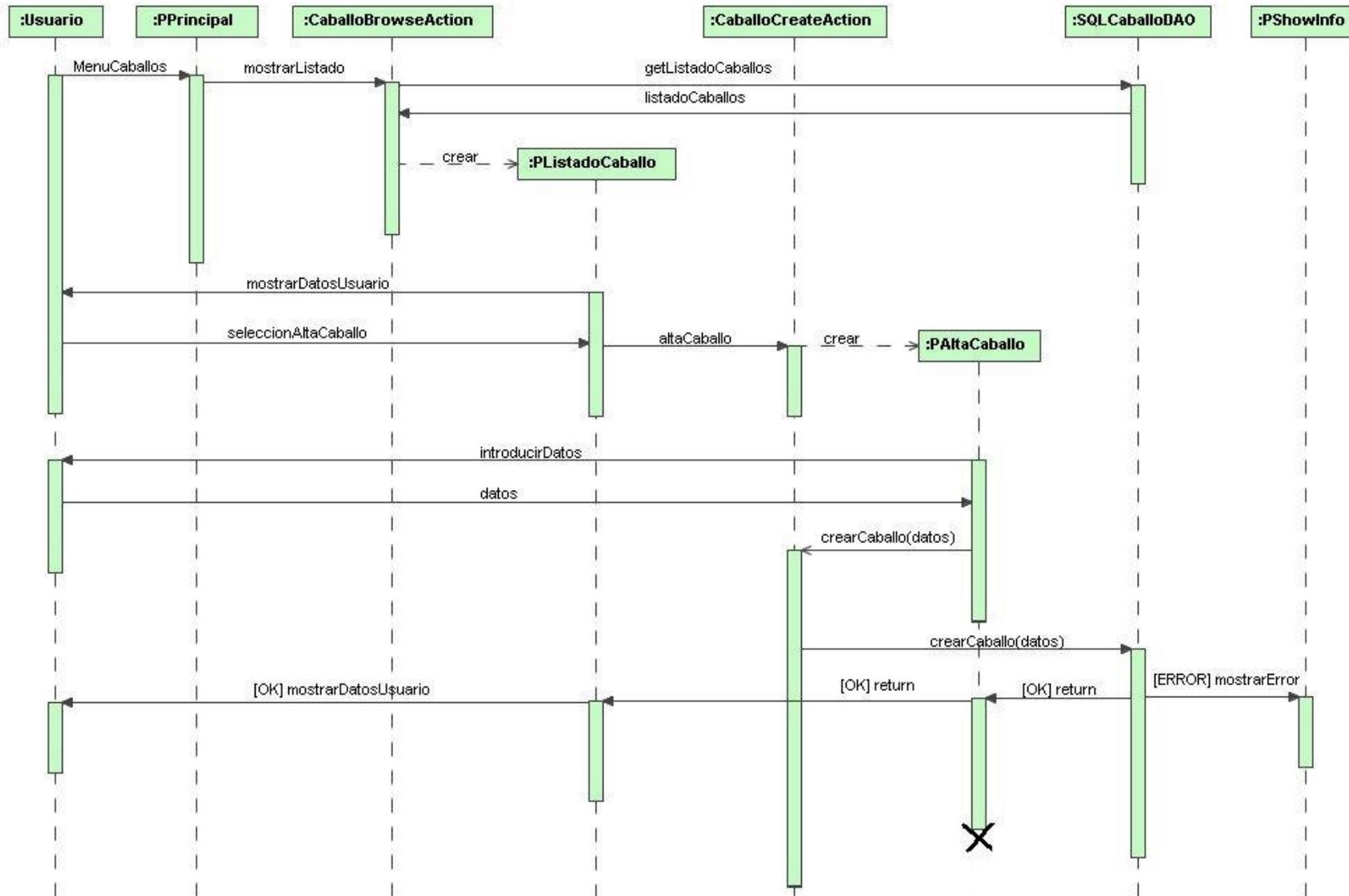
1. Listado de caballos



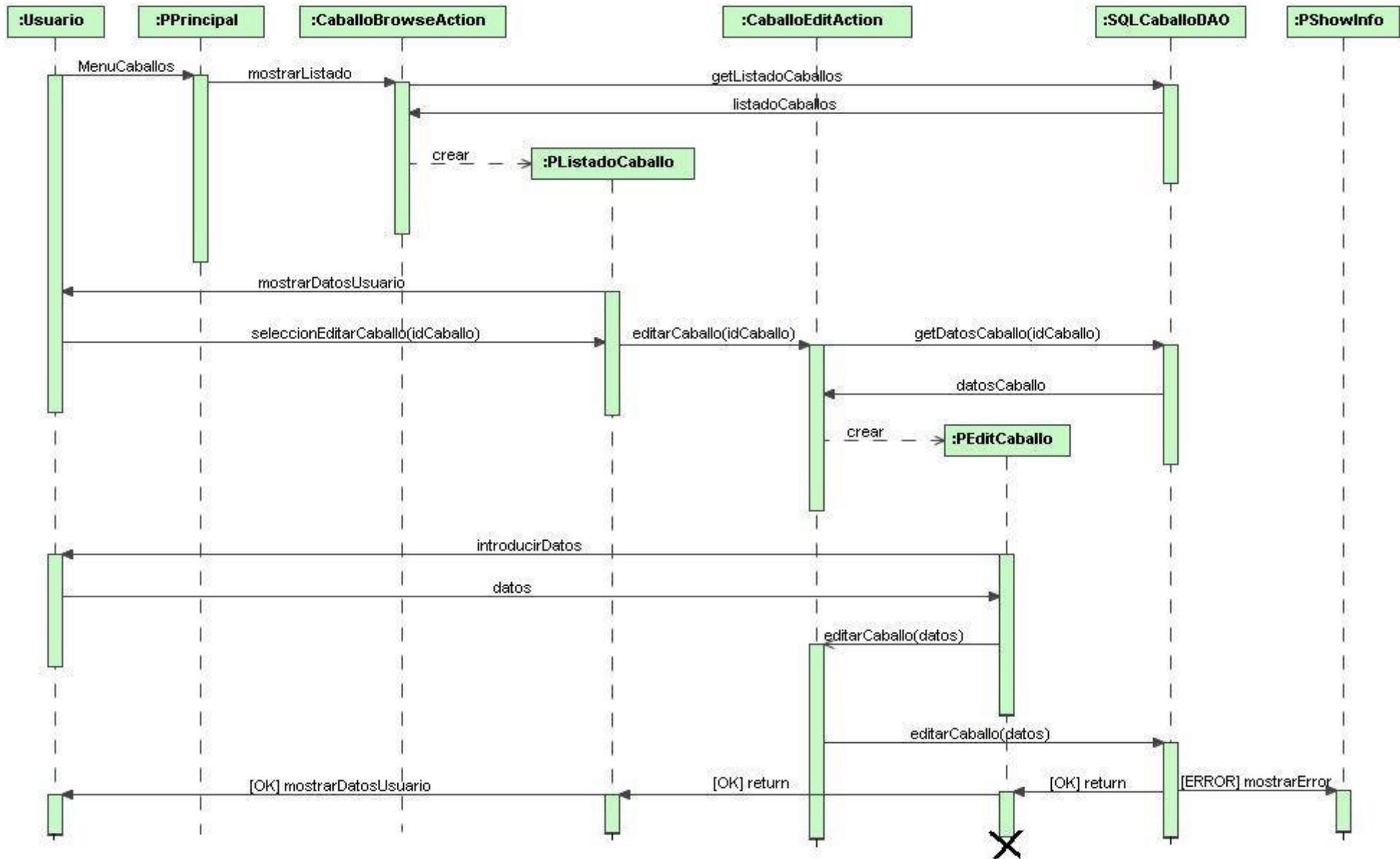
2. Actualizar listado de caballos



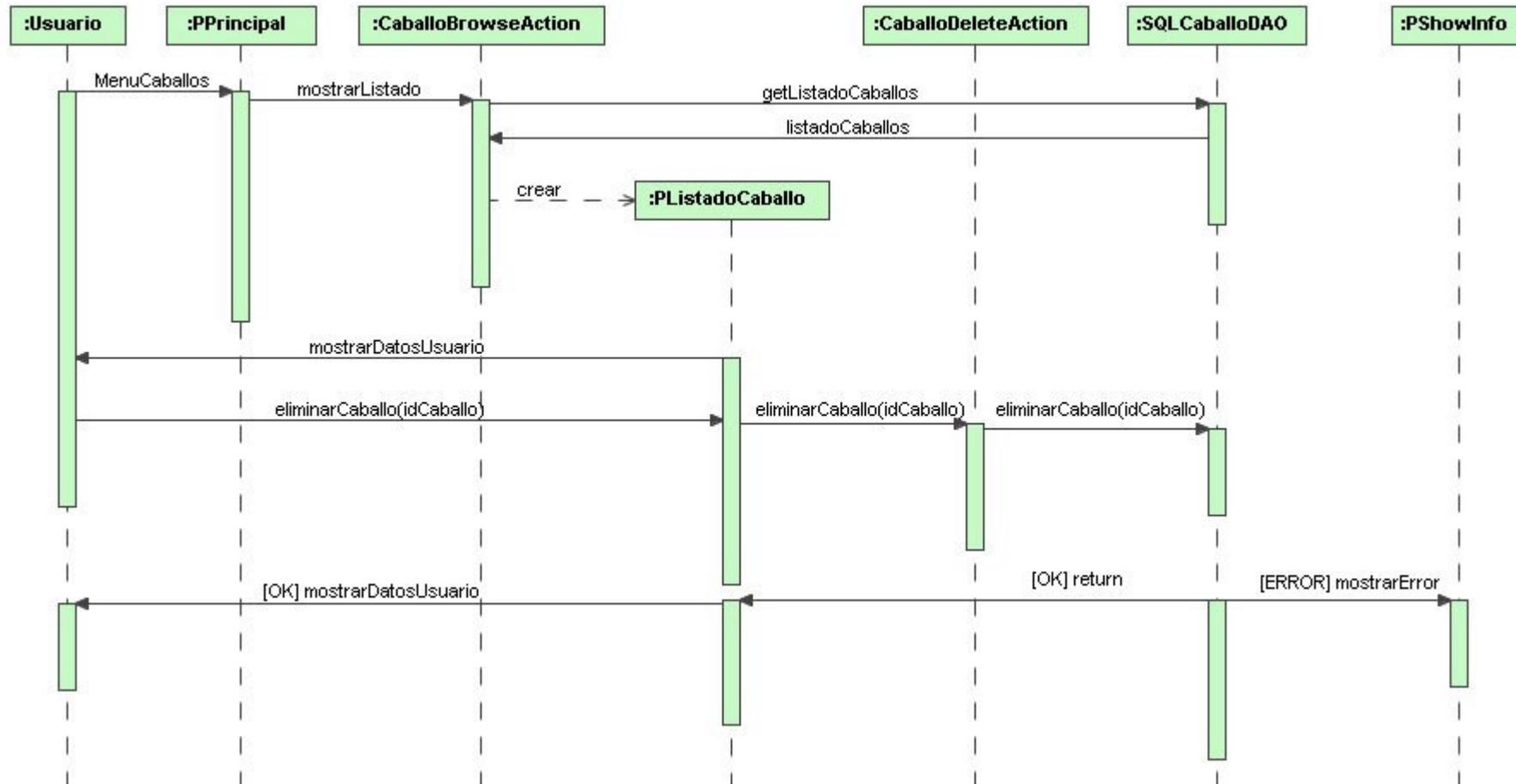
3. Alta de caballos



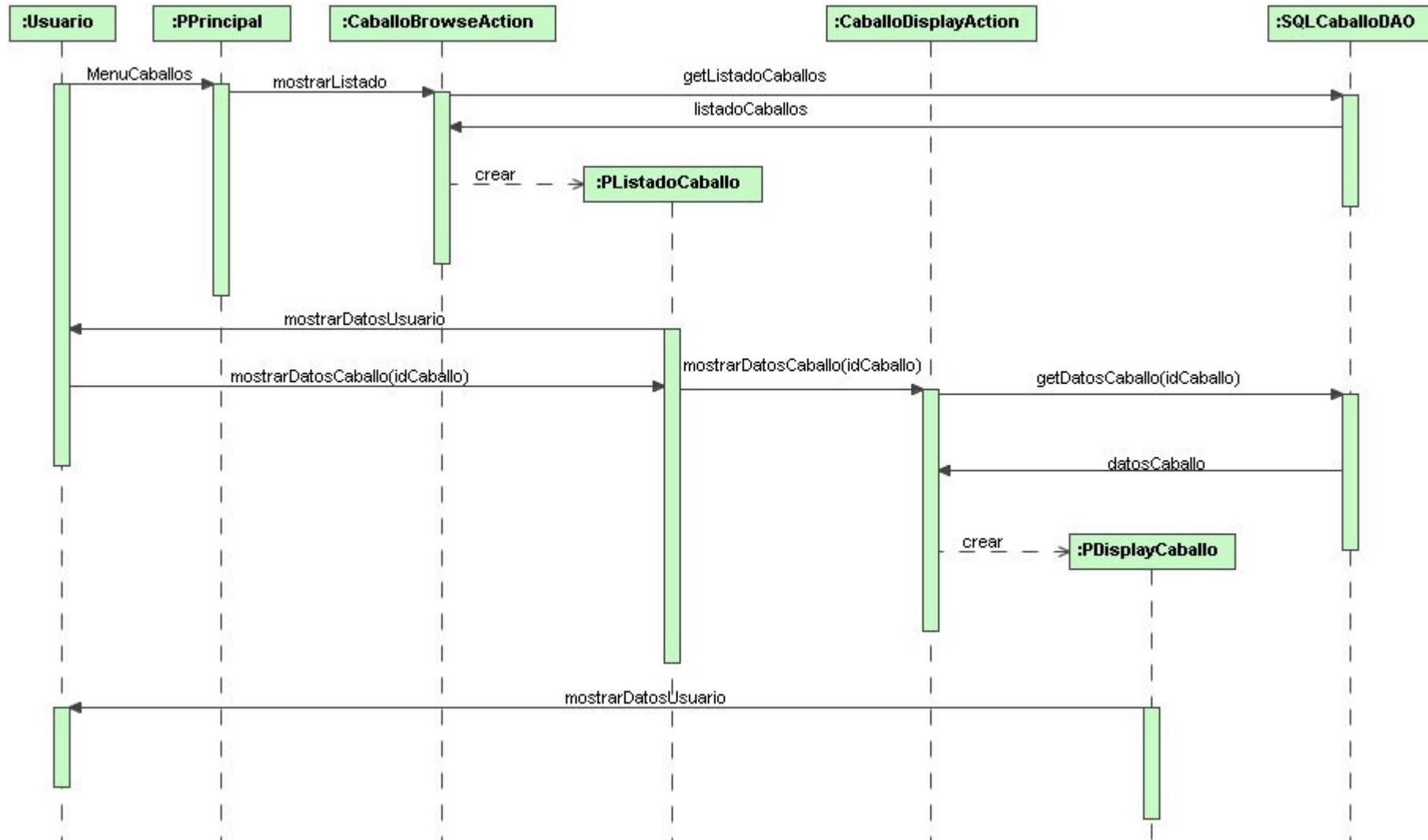
4. Editar caballos



5. Eliminar caballo



6. Información detallada del caballo



3.3 Diseño de la base de datos

Se ha optado por usar el sistema gestor de base de datos MySQL ya que es muy utilizado en aplicaciones Web y además tiene la particularidad de ser software libre. La principal característica que me ha hecho optar por este SGBD es debido a que en aplicaciones Web hay una baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

Como primer paso para el diseño de la base de datos relacional es definir como debemos tratar la jerarquía de herencia del diagrama de clases (supresión de la herencia) de la aplicación:

- “Persona – Usuario”, “Persona – Miembro”: como que la clase *Persona* no interviene en ninguna relación (no se crearán objetos *Persona*) y debido a que solo se crearán objetos de la clase *Usuario* o *Miembro*, eliminamos la clase *Persona* pasando sus atributos a su subclase. Por tanto definimos una única tabla para esta jerarquía: *Persona*. A esta tabla le añadiremos un nuevo atributo de tipo integer: *tipo*, que nos servirá para diferenciar entre los usuarios y miembros que aparecen en el sistema.

La clase *Carrera* tiene 2 atributos que contendrán distintos valores (arrays), estos son “clasificados” y “premios”. Para guardar estos distintos valores definimos una nueva tabla *CarreraPremios* que tendrá los atributos de *carrera*, *clasificados* y sus respectivos premios. Como consecuencia de esta nueva tabla desaparecen los atributos “clasificados” y “premios” de la tabla *Carrera*.

Así pues, teniendo en cuenta que el resto de elementos del diagrama se mantiene igual obtenemos un total de 8 tablas. Definimos la base de datos de la siguiente forma (se entrega el Script de creación de la BBDD con el proyecto):

Los valores que aparecen entre paréntesis son los atributos de la tabla, las claves primarias de las tablas son los atributos que aparecen subrayados y en negrita, cuando es necesario se indican las claves foráneas.

Boletin (num, fecha, portada, nota)

Caballo (id, nombre, sexo, capa, fechaNacimiento, padre, madre, propietario, criador, poblacion, provincia, pais, fechaAlta, fechaBaja)

```
{padre}          es clave foránea hacia caballo(id)
{madre}          es clave foránea hacia caballo(id)
{propietario}    es clave foránea hacia persona(dni)
{criador}        es clave foránea hacia persona(dni)
```

Carrera (id, fecha, numCarrera, hipodromo, distancia, modo, tipo, categoria, modalidad, sumaPremios)

{hipodromo} es clave foránea hacia hipodromo(id)

CarreraPremios (idCarrera, clasificado1, clasificado2, clasificado3, clasificado4, clasificado5, clasificado6, clasificado7, clasificado8, clasificado9, clasificado10, premio1, premio2, premio3, premio4, premio5, premio6, premio7, premio8, premio9, premio10)

{idCarrera} es clave foránea hacia carrera(id)
 {clasificado1} es clave foránea hacia persona(dni)
 {clasificado2} es clave foránea hacia persona(dni)
 {clasificado3} es clave foránea hacia persona(dni)
 {clasificado4} es clave foránea hacia persona(dni)
 {clasificado5} es clave foránea hacia persona(dni)
 {clasificado6} es clave foránea hacia persona(dni)
 {clasificado7} es clave foránea hacia persona(dni)
 {clasificado8} es clave foránea hacia persona(dni)
 {clasificado9} es clave foránea hacia persona(dni)
 {clasificado10} es clave foránea hacia persona(dni)

Hipodromo (id, nombre, direccion, poblacion, provincia, pais, telefono, fax, web, email)

Incidencia (id, fecha, caballo, descripcion)

{caballo} es clave foránea hacia caballo(id)

Persona (dni, nombre, apellidos, domicilio, cp, poblacion, provincia, pais, email, fechaNacimiento, sexo, telefono, movil, tipo, carnet, fechaAlta, fechaBaja)

Sancion (id, fecha, boletin, caballo, conductor, entrenador, propietario, hipodromo, fechaSuspensionIni, fechaSuspensionFin, importe, motivo, emitida, pagada, fechaPago)

{boletin} es clave foránea hacia boletin(id)
 {caballo} es clave foránea hacia caballo(id)
 {conductor} es clave foránea hacia persona(dni)
 {entrenador} es clave foránea hacia persona(dni)
 {propietario} es clave foránea hacia persona(dni)
 {hipodromo} es clave foránea hacia hipodromo(id)

Existen una serie de atributos de tipo entero (int) y carácter (char) en varias tablas de la base de datos, cuyos valores y significados son los siguientes:

Tabla persona:

- Sexo:

H: Hombre

M: Mujer

- Tipo:
 - 1: Administrador
 - 2: Gestor
 - 3: Auxiliar
 - 4: Miembro

Tabla caballo:

- Sexo:
 - H: Hembra
 - M: Macho
- Capa:
 - A: Alazán
 - AD: Alazán dorado
 - AT: Alazán tostado
 - B: Blanco
 - C: Castaño
 - CC: Castaño claro
 - CO: Castaño oscuro
 - N: Negro
 - O: Overo
 - T: Tordo
 - TO: Tordo oscuro

Tabla carrera:

- Modo:
 - A: Autostart
 - H: Handicap
- Tipo:
 - E: Trote enganchado
 - M: Trote montado
- Modalidad:
 - E: Europea
 - I: Internacional
 - N: Normal
- Categoría:
 - N: Normal
 - C: Concertada
 - L: Clásica
 - A: Amateurs
 - R: Criterium
 - P: Potros
 - E: Campeonato
 - Z: Lanzada

Tabla sancion:

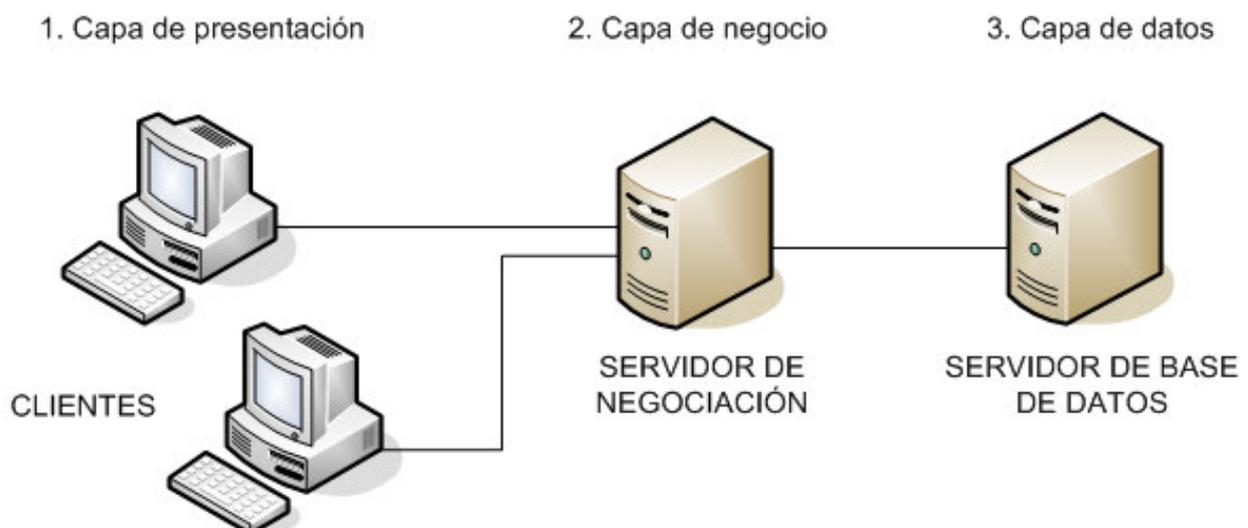
- Emitida:
 - S: Si
 - N: No
- Pagada:
 - S: Si
 - N: No

3.4 Arquitectura

El proyecto a realizar consiste en una aplicación Web accesible desde la red Internet basada en la tecnología J2EE.

Se usará la tecnología de tres capas ya que nos permite la separación de la lógica de negocio de la lógica de diseño. La ventaja principal de este estilo, es que el desarrollo se puede llevar a cabo en varios niveles y en caso de algún cambio sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Las capas son las siguientes:

- **Capa de presentación:** presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.
- **Capa de negocio:** es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. En esta capa se establecen todas las reglas que deben cumplirse. Se comunica con la capa de presentación para recibir las solicitudes del usuario y presentar los resultados, y con la capa de datos para solicitar al SGBD almacenar o recuperar datos de él.
- **Capa de datos:** es donde residen los datos. Está formada por uno o más SGBD que realiza todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. Como ya se ha comentado anteriormente el SGBD elegido para el desarrollo del proyecto es MySQL.



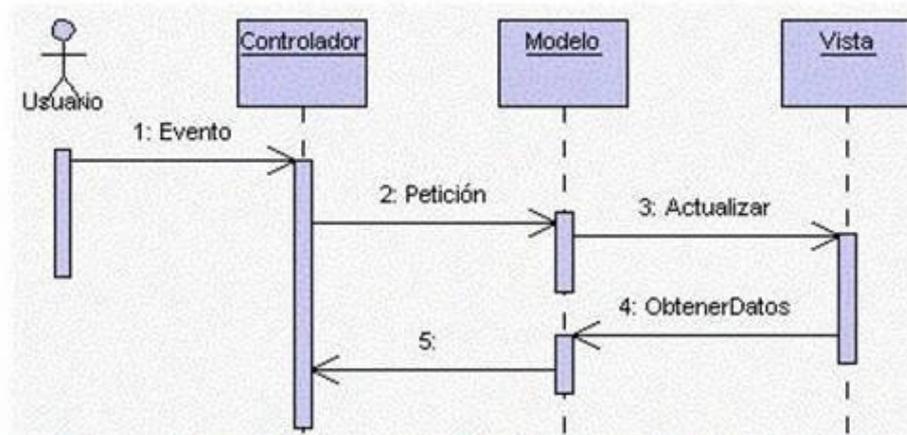
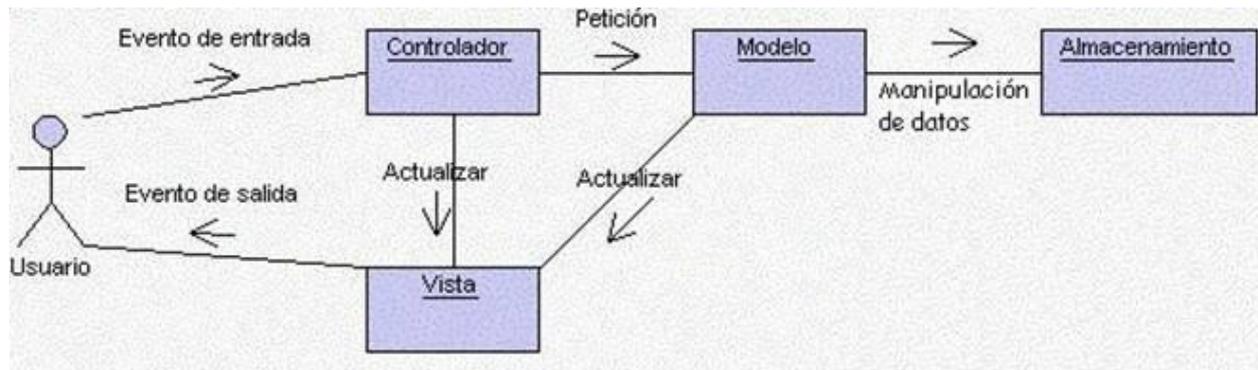
3.5 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Son una solución a un problema de diseño no trivial que es efectiva (ya se resolvió el problema satisfactoriamente en ocasiones anteriores) y reusable (se puede aplicar a diferentes problemas de diseño en distintas circunstancias).

Los patrones utilizados para el desarrollo de la aplicación Web son los siguientes:

- **MVC:** este patrón llamado Modelo Vista Controlador es el más importante (en este proyecto) y en el cual me he basado para la implementación de la aplicación Web. Se ha implementado a partir del “framework Struts” ya que es utilizado ampliamente en el mundo empresarial y es considerado de gran solidez. Mediante este patrón podremos realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reutilización. Es muy utilizado en aplicaciones Web, donde la vista es la página HTML/JSP y el código que provee de datos dinámicos a la página, el controlador es el SGBD y el modelo es el modelo de datos.
 1. Modelo: representa la información del sistema y las reglas de negocio.
 2. Vista: muestra la información del modelo al usuario.
 3. Controlador: se encarga de procesar las entradas/peticiones del usuario.

Sus diagramas de colaboración y secuencia nos ayudaran a tener una idea general del patrón MVC:



- **Front Controller:** este patrón actúa de controlador para el manejo de las peticiones. Controla y coordina el procesamiento de todos los usuarios a través de varias peticiones. Dicho controlador maneja el control de peticiones, incluyendo la invocación de los servicios de seguridad como la autenticación y autorización, delegar el procesamiento de negocio, controlar la elección de una vista apropiada, el manejo de errores, y el control de la selección de estrategias de creación de contenido. Mediante este patrón conseguimos centralizar el control de la aplicación, mejorar la manejabilidad de la seguridad y mejorar la reutilización.
- **Session Facade:** este patrón actúa como una fachada (facade) para encapsular la complejidad de las interacciones entre los objetos de negocio participantes en un flujo de trabajo. El Session Facade maneja los objetos de negocio y proporciona un servicio de acceso uniforme a los clientes.

Mediante este patrón conseguimos introducir una capa controladora para la capa de negocio, exponer una interface uniforme, reducir el acoplamiento e incrementar la manejabilidad, mejorar el rendimiento incrementando la manejabilidad, mejorar el rendimiento reduciendo los métodos específicos, proporciona un acceso genérico, centraliza el control de seguridad, centraliza el control de transacciones y expone menos interfaces remotos a los clientes.

- **Data Access Object (DAO):** objeto de acceso a datos. Es utilizado para abstraer y encapsular todos los accesos a la fuente de datos (BBDD). El DAO oculta completamente los detalles de implementación de la fuente de datos a sus clientes. Como el interface expuesto por el DAO no cambia cuando cambia la implementación de la fuente de datos subyacente, este patrón permite al DAO adaptarse a diferentes esquemas de almacenamiento sin que esto afecte a sus clientes o componentes de negocio. Principalmente actúa como un adaptador entre el componente y la fuente de datos.

Mediante este patrón conseguimos obtener una migración más fácil, reducir la complejidad del código de los objetos de negocio y sobre todo centralizar todos los accesos a datos en una capa independiente.

- **Transfer Object (TO):** también conocido como Value Object. Permite optimizar la transferencia de datos a través de las capas de la aplicación. Este patrón es muy efectivo ya que en lugar de mandar o recibir datos de manera individual, contiene todos los datos en una estructura única (estructura de datos) solicitada ya sea por la petición o la respuesta. Se utiliza una única llamada a un método para enviar y recuperar el Transfer Object.

Mediante este patrón conseguimos transferir más datos en menos llamadas remotas, reducir el tráfico de la red, reducir la duplicación de código, accesos y transacciones concurrentes.

4. IMPLEMENTACIÓN

4.1 Tecnología y herramientas utilizadas

- **JDK 1.6.0:** versión de java.
- **Apache Ant 1.7.0:** herramienta usada en programación para la realización de tareas mecánicas y repetitivas, normalmente durante la fase de compilación y construcción. Tiene la ventaja de no depender de las órdenes de shell de cada sistema operativo, sino que se basa en archivos de configuración XML y clases Java para la realización de las distintas tareas. Esta herramienta nos ayudará a compilar y desplegar la aplicación Web de forma automática mediante el fichero de configuración *build.xml*. Es la herramienta utilizada para generar los ficheros “ibTrot.jar” e “ibTrot.war”
- **Eclipse 3.1.2:** plataforma utilizada para desarrollar entornos integrados de desarrollo. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI (mediante pluggins). Cabe destacar que se ha usado la versión 3.1.2 en lugar de la última disponible 3.2.2 debido a las incompatibilidades de esta versión con JBoss IDE.
- **Struts 1.1:** framework muy extendido en el desarrollo de aplicaciones Web bajo la plataforma J2EE. Se basa en el patrón de diseño Modelo Vista Controlador (MVC) el cual se utiliza ampliamente y es considerado de gran solidez. Nos permite realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reutilización. Oculta al programador los detalles de http, jsp, servlets...

Aún existiendo versiones superiores se ha optado por esta por ser más estable y por problemas de compatibilidad con Common Controls 1.5.011.

Sus subsistemas son:

1. **Modelo:** datos y reglas de negocio.
2. **Vista:** muestra la información del modelo al usuario.
3. **Controlador:** se encarga de gestionar las entradas/peticiones del usuario.

Nota: su funcionamiento será más detallado en la presentación del TFC (fichero power point).

- **Common-Controls 1.5.011:** framework de presentación para aplicaciones Web basadas en J2EE. Se combina con el framework Struts para implementar un desarrollo estructurado y orientado a objetos facilitando y mejorando notablemente la interfaz de usuario (capa de presentación). No se trata de un software de código abierto.
Nota: consultar anexo de la documentación.

- **JBoss AS 4.0.5:** servidor de aplicaciones J2EE de código abierto e implementado completamente en java, lo que lo caracteriza para ser utilizado bajo cualquier plataforma. Incluye el contenedor Web Apache Tomcat (servidor JSP/Servlets).
- **JBoss IDE 1.6.0:** plugin de JBoss que conecta el IDE Eclipse con el servidor de aplicaciones JBoss de forma que la depuración y otras tareas asociadas al desarrollo de la aplicación Web (como por ejemplo arrancar/parar el servidor JBoss) pueden ser realizadas desde el propio entorno Eclipse. Nos aporta las librerías J2EE 1.4 y Web Services 1.0 a nuestro proyecto.
- **MySQL 5.0.27:** sistema gestor de base de datos utilizado en el proyecto. Está caracterizado por ser software libre.
- **MySQL-gui-tools-5.0-r11a:** herramientas que nos facilitan la administración del SGBD MySQL.
- **MySQL-connector-java-5.0.3:** librería que nos permite acceder a la base de datos desde nuestro proyecto vía JDBC.
- **Magic Draw UML:** herramienta utilizada para generar los diagramas UML.

5. VALORACIÓN ECONÓMICA

Aunque este no es uno de los objetivos principales del proyecto cabe destacar que no nos encontramos en una situación muy real de poder definir el coste económico del proyecto, ya que inicialmente el alumno desconocía la tecnología J2EE y ha empleado un gran número de horas en formarse para poder desarrollar el proyecto.

Todas las herramientas/tecnologías utilizadas para el desarrollo del TFC son software libre a excepción del framework de presentación *Common Controls* y la herramienta de generación de diagramas UML *Magic Draw*.

Se ha utilizado una versión de prueba *Common Controls* por lo que se deberá consultar los términos de licencia en el anexo de esta documentación. Se ha optado por utilizar este framework debido a que el alumno ya había tenido algún contacto con él en la última empresa dónde trabajó.

Aunque para el desarrollo de los diagramas UML podríamos haber utilizado cualquier otra herramienta gratuita disponible en el mercado, se ha optado por utilizar *Magic Draw UML* debido a la experiencia obtenida en esta herramienta a lo largo de los estudios de ITIG.

Los precios de *Common Controls* se pueden consultar en:

<http://www.common-controls.com/shop/index.php>

6. CONCLUSIONES

Se han conseguido los objetivos especificados al principio de este documento. Mediante el desarrollo de este TFC el alumno ha profundizado en el mundo del J2EE conociendo las principales tecnologías y herramientas disponibles en el mercado para el desarrollo de una aplicación Web accesible desde la red Internet.

Sin ninguna duda el alumno marca la experiencia de muy positiva, ya que los conocimientos adquiridos le serán de gran utilidad en el mundo laboral.

7. GLOSARIO

Usuario: persona que tiene acceso a la aplicación *ibTrot*. Hay 3 tipos: administrador, gestor y auxiliar.

Miembro: persona identidad del sistema. Aquí están incluidos los entrenadores, propietarios, conductores y socios.

Hipódromo: entidad que representa la instalación destinada a carreras de caballos y otros ejercicios; dispone de una pista, generalmente elíptica, y de tribunas para los espectadores.

Carrera: entidad que representa la competición de velocidad entre varios caballos.

Incidencia: Hecho que se produce en lo relativo a los caballos (normalmente son incidencias de los caballos en las carreras o controles ANTIDOPPING)

Sanción: pena económica establecida para el que infringe una ley o una norma establecida.

Boletín: publicación periódica (cada martes) donde se informa de todo lo relativo al hipódromo (resultados de carreras, premios, sanciones, incidencias, etc.)

Administrador: usuario que puede ejecutar todas las funcionalidades de la aplicación.

Gestor: usuario que puede ejecutar las funcionalidades asociadas a los subsistemas de conexión, mantenimiento (exceptuando las funciones asociadas a los usuarios) y gestión.

Auxiliar: usuario que puede ejecutar las funcionalidades asociadas a los subsistemas de conexión e información.

Contraseña: palabra clave utilizado por los usuarios para acceder a la aplicación.

Entidad: Objeto del mundo real que podemos distinguir del resto de objetos y que está caracterizado por una serie de atributos (características).

BBDD: siglas de base de datos.

SGBD: siglas de sistema gestor de base de datos.

J2EE: siglas de Java 2 Enterprise Edition (edición empresarial de Java)

TFC: siglas de trabajo de fin de carrera.

8. BIBLIOGRAFÍA

Material UOC: **Enginyeria del programari**

BENET CAMPDERRICH FALGUERAS, Recerca informàtica SL

Material UOC: **Tècniques de desenvolupament de programari**

FATOS XHAFA

Material UOC: **Catálogo de patrones**

JORDI PRADEL I MIQUEL, JOSÉ ANTONIO RAYA MARTOS

Material UOC: **Ingeniería del software de componentes y sistemas distribuidos.**

PROFESORADO DE INFORMÁTICA Y MULTIMEDIA. Versión 1.12e

Material UOC: **J2EE Una plataforma de componentes distribuida**

JOSEP MARIA CAMPS I RIBA

Jakarta Struts Live

RICK HIGHTOWER

Beginning JSP 2: From Novice to Professional 2004

PETER DEN HAAN, LANCE LAVANDOWSKA, SATAHYA NARAYANA

PANDURANGA, KRISHNARAJ PERRUMAL

MySQL Technical Reference Manual

First steps with struts and JSP

<http://www.laliluna.de/struts-jsf-jsp-servlets-tutorials.html>

Documentación online Core J2EE Patterns

<http://java.sun.com/blueprints/corej2eepatterns/Patterns/index.html>

<http://www.programacion.net/java/tutorial/patrones> (castellano)

<http://www.programacion.net/java/tutorial/patrones2> (castellano)

Documentación online Struts

<http://struts.apache.org>

Documentación online JBoss

<http://www.jboss.com>

<http://www.jboss.org>

Documentación online JBoss IDE

<http://www.jboss.com/products/jbosside>

Documentación online ANT

<http://ant.apache.org/>

Documentación online HTML

<http://www.programacion.net/html/tutorial/curso>

Documentación online de hojas de estilo: CSS

<http://www.w3schools.com/css>

Documentación online API Java

<http://java.sun.com/j2se/1.4.2/docs/api>

Documentación online Struts TAGS

<http://struts.apache.org/1.2.7/userGuide/struts-html.html>

Documentación online Common-Controls framework para J2EE

<http://www.common-controls.com/en/index.php>

Documentación online Common-Controls Tag Library

<http://www.common-controls.com/en/resources/taglib/index.html>

9. ANEXO

9.1 Manual de usuario

El manual de usuario con las capturas de pantalla se entrega en un documento individual: "ManualUsuario.doc"

9.2 Licencia del framework de presentación Common Controls

Para el desarrollo de la aplicación se ha utilizado una versión de prueba, ya que no se trata de un software de código abierto. Según especifica la licencia de este producto deberá ser eliminado del PC pasados 30 días. El fichero de licencia de la versión *trial* es el siguiente:

Common Controls Trial Version License.
Copyright (c) 2000-2003 SCC Informationssysteme GmbH.

Licence terms and conditions:

This licensing contract is a legally binding contract between you (either as a natural or a judicial person) as the end-user and SCC Informationssysteme GmbH, Mühlthal (Germany).

The use of this software product pre-requires you to agree to and accept the following terms and conditions and limitations. If you do not agree to the terms and conditions of this contract, SCC Informationssysteme GmbH is not willing to license the software to you.

In that case, you are not authorized to use the software.

By downloading, copying and using the software, you declare yourself bound to the terms and conditions of this license contract.

1. GRANTING A LICENSE. ("License")

You, as the end-user, are given the simple, non-exclusive and personal right to test the software being offered here under the conditions of this contract, **on a single computer for a period of 30 days, free of cost.** At the end of this time, you must either a) purchase a registered version of the software or b) delete the software and all the documentation related to it from your computer. This also includes any backup copies in your possession.

2. COPYRIGHT.

The ownership and the copyright of the software product (including any images, photographs, animation, video, music, text and "Applets" that is present in the software module), the accompanying printed material and all copies is with SCC Informationssysteme GmbH. The software product is protected against copying by copyright, the conditions of international agreements and legal regulations. Any reproduction or any kind of reselling of the software, in whole or in part, is expressly forbidden by the applicable laws and can result in severe civil and criminal penalties.

3. FURTHER RESTRICTIONS.

The licensing agreement assumes that the licensee requires the software for use in his own operations. You are not entitled to rent, hire or lease the software or to transfer it in any other manner. Reverse engineering, decompiling and disassembling of the software are not permitted. Although the software has been extensively tested, because of the many hardware and software conditions in which the software can be used, no warranty or liability claims whatsoever can be entertained.

UNDER NO CIRCUMSTANCES CAN SCC INFORMATIONSSYSTEME GMBH AND/OR ITS SUPPLIERS BE MADE LIABLE FOR SPECIFIC OR INDIRECT DAMAGE RESULTING FROM FAILURE, LOSS OF DATA OR LOSS OF PROFIT - REGARDLESS OF WHETHER IT HAS RESULTED UNDER CONTRACT-CONFORMANT USAGE OR FROM NEGLIGENCE OR OTHER IMPERMISSIBLE ACTIONS AND OWING TO OR IN CONJUNCTION WITH THE SOFTWARE, SOFTWARE DOCUMENTS OR INFORMATION AVAILABLE ON THIS SERVER.