

**exafarm**  
programa preparador  
de exámenes de **farmacología**

Estudiante: Francisco José Jimeno Demuth

Tema: J2EE

Consultor: Salvador Campo Mazarico

20 de junio de 2011

A Vega y Sandra

## Resumen

El departamento de farmacología de la Universidad de Oviedo posee un sistema “manual” de generación de exámenes en el que los profesores copian y pegan las preguntas en editores de texto para crear finalmente un examen con diferentes versiones que imprimirán en papel.

Exafarm es un programa que busca facilitar y disminuir la carga de trabajo de los profesores automatizando el proceso de generación de exámenes tipo test.

El marco de trabajo elegido para este proyecto es J2EE, dentro de un entorno JavaServer Faces 2.0, basado en presentaciones web, junto con EJB y JPA, todos ellos estándares de programación en JEE6. Trabajar en Java, junto con estos estándares aporta múltiples ventajas, como la de ser un entorno de trabajo multiplataforma, basado en la tecnología cliente-servidor, con modularidad y escalabilidad del proyecto enfocado a Java para entornos empresariales.

La técnica de programación utilizada en el programa se basa en el modelo de programación de 3 capas (MVC) que permite separar de forma clara los diferentes bloques de programación.

La persistencia del programa se realiza a través del gestor de base de datos relacional MySQL por su gratuidad, fiabilidad, eficiencia y facilidad de uso.

Todo el entorno se manejará a través del servidor de aplicaciones Glassfish v.3.1 en su versión de código abierto, que cumple con los últimos estándares de programación en JEE 6.

Para la realización de este proyecto se han utilizado conocimientos adquiridos a lo largo del estudio de Ingeniería Técnica de Informática de Gestión de la Universitat Oberta de Catalunya.

**Área de TFC:** J2EE

**Palabras clave:** Java, J2EE, JavaServer Faces, EJB, Tests

**Licencia:** El trabajo está sujeto –salvo que se indique lo contrario– a una licencia de Reconocimiento –No comercial– Sin obras derivadas 2.5 España de Creative Commons. Podéis copiarlo, distribuirlo y transmitirlo públicamente siempre que

citéis el autor y la obra, no se haga uso comercial y no se haga copia derivada. La licencia completa se puede consultar en <http://creativecommons.org/licenses/by-cn-nd/2.5/es/deed.es>.

## Tabla de contenido

<b>Resumen.....</b>	<b>3</b>
<b>Introducción.....</b>	<b>8</b>
Justificación del TFC y contexto en el que se desarrolla .....	8
Objetivos .....	9
Enfoque y método seguido .....	9
Internacionalización .....	10
Entorno de trabajo.....	11
Gestor de base de datos .....	11
<b>Planificación del proyecto.....</b>	<b>11</b>
<b>Composición del programa .....</b>	<b>12</b>
Arquitectura básica del proyecto .....	12
<b>Diagrama de paquetes .....</b>	<b>13</b>
<b>Casos de uso .....</b>	<b>15</b>
Identificación de los actores.....	15
Casos de uso .....	15
Fichas de los casos de uso.....	17
<b>Diagramas de actividad.....</b>	<b>29</b>
Selección de preguntas.....	30
Generar examen .....	31
<b>Estructura de la base de datos .....</b>	<b>32</b>
<b>Requerimientos de software .....</b>	<b>32</b>
Requerimientos para el servidor.....	33
Máquina virtual de Java (JVM) .....	33
Servidor de aplicaciones .....	33
Requerimientos para el cliente .....	33
<b>Conclusiones .....</b>	<b>33</b>
<b>Bibliografía.....</b>	<b>35</b>
<b>Anexo I: Instalación del programa .....</b>	<b>36</b>
Preparación de la base de datos .....	36
Creación de la base de datos .....	36

<b>Servidor de aplicaciones.....</b>	<b>51</b>
<b>Paquete del programa .....</b>	<b>52</b>

## Tabla de ilustraciones

Ilustración 1. Tabla de temporalización y funciones del programa.....	12
Ilustración 2. Arquitectura básica del proyecto .....	13
Ilustración 3. Diagrama de paquetes .....	14
Ilustración 4. Relación de actores del programa.....	15
Ilustración 5. Diagrama general de casos de uso .....	16
Ilustración 6. Pantalla de caso de uso Identificación.....	17
Ilustración 7. Pantalla de añadir pregunta.....	18
Ilustración 8. Pantalla de listado de preguntas.....	19
Ilustración 9. Pantalla de listado de exámenes activos .....	20
Ilustración 10. Pantalla de modificación de pregunta .....	21
Ilustración 11. Pantalla de modificación de datos de usuario.....	22
Ilustración 12. Pantalla de listado de exámenes .....	23
Ilustración 13. Pantalla de generación de exámenes .....	24
Ilustración 14. Pantalla de añadir examen.....	25
Ilustración 15. Pantalla de modificar bloque.....	26
Ilustración 16. Pantalla de añadir bloque .....	27
Ilustración 17. Pantalla de listar temas .....	28
Ilustración 18. Pantalla de añadir tema .....	29
Ilustración 19. Diagrama de actividad de selección de preguntas .....	30
Ilustración 20. Diagrama de actividad de generación de examen .....	31
Ilustración 21. Esquema de la base de datos .....	32

## Introducción

### Justificación del TFC y contexto en el que se desarrolla

Desde hace años el departamento de Farmacología realiza de forma bianual exámenes de tipo test en papel a los alumnos de 5º de medicina de la Universidad de Oviedo. Para la generación de un examen los diferentes profesores de la asignatura envían las preguntas en un fichero de texto al coordinador que prepara los exámenes de forma “artesanal”. Con un procesador de textos añade las preguntas, que se copian y pegan para realizar las diferentes versiones del examen. Esta tarea supone una dedicación casi exclusiva del coordinador de la asignatura de varios días para la creación del examen, con una probabilidad elevada de confundirse en la generación de las diferentes versiones que contienen las mismas preguntas pero ordenadas en diferente orden. La realización de estos exámenes, además de ser complicada, tiene múltiples inconvenientes como que no permite un control sencillo de las preguntas que se pusieron en años anteriores, con lo que es difícil evitar que las preguntas se repitan en años posteriores.

Entre las funciones más importantes del programa encontraremos:

- Limitación de acceso a la aplicación con usuario y contraseña.
- Posibilidad de gestión de usuarios por parte de un administrador del programa.
- Gestión de temas de la asignatura distribuidos en diferentes bloques por parte del administrador del programa.
- Gestión efectiva y cómoda de las preguntas introducidas por los diferentes profesores de la asignatura.
- Creación de exámenes para una determinada convocatoria con un número determinado de preguntas de tipo test. La generación de los exámenes se realizará de forma automática produciéndose un documento PDF de cada versión del examen.
- Control de qué preguntas han aparecido en exámenes anteriores.



El programa permitirá que los diferentes profesores puedan incluir preguntas de examen en una base de datos que luego se encargará de generar los exámenes con sus diferentes versiones de forma totalmente automatizada.

## Objetivos

El objetivo de este proyecto es realizar un programa, llamado exafarm, con la idea de disminuir la carga de trabajo de los docentes el máximo posible. El programa se encargará de realizar diferentes procesos de forma automática ayudando así a que la generación de los exámenes no sea tan laboriosa como lo ha sido durante estos últimos años.

Realizar este proyecto me ayudará a profundizar en aspectos de la programación para entornos JEE, una tecnología muy extendida en el ámbito profesional y con una tendencia a aumentar en los próximos años.

## Enfoque y método seguido

En el momento inicial, al plantear la planificación del proyecto, realicé una lectura extensa de diferentes herramientas que permiten realizar programas relacionados con la programación en JEE. Dado que mi experiencia en la programación en entornos JEE era casi nula, y que existe una elevada cantidad de herramientas para la elaboración de programas en este entorno, fue necesario mucho tiempo para documentarme. Las fuentes de información fueron desde páginas web hasta libros relacionados con el tema.

La programación en JEE permite la programación de aplicaciones en el lenguaje Java, desarrollado inicialmente por la empresa Sun Microsystems, que recientemente fue adquirida por Oracle. Este entorno de trabajo se ejecuta sobre un servidor de aplicaciones y sigue un estándar de funcionalidad que es controlado por la Java Community Process.

La programación en J2EE inicialmente era bastante complicada y confusa, por lo que rápidamente aparecieron herramientas o frameworks que hacían más sencilla y amigable su utilización, fue el momento en el que aparecieron herramientas como Struts, Spring, Hibernate... A raíz de esto, en futuras versiones de JEE, Sun

creo unas nuevas especificaciones que ayudaron a mejorar y facilitar el trabajo en este entorno. La versión JEE 6 añadía diferentes especificaciones nuevas entre las que se encuentra JavaServer Faces(JSF) 2.0, EJB 3.1... herramientas que he utilizado para la realización de mi proyecto. Me decidí a programar en JSF en su versión 2.0 debido a que esta tecnología recoge muchos de los aspectos positivos de frameworks con más años de experiencia. JSF no se trata de un framework propiamente dicho, sino que es más bien una especificación, por lo que existen múltiples implementaciones del mismo que deben seguir unas normas pautadas por Sun. El que se trate de una tecnología más nueva, comparada con Struts o Spring, por ejemplo, le aporta múltiples ventajas, pero también inconvenientes. JSF carece de la experiencia que puede tener Struts, que ya tiene versiones más avanzadas y depuradas, esto hace que en ocasiones pueda haber problemas de estabilidad y errores que han de ser subsanados. Una especificación tan nueva también hace que exista menos documentación que se pueda consultar. En la versión 2.0 (aparecida a finales del año 2009) hubo muchos cambios, métodos utilizados anteriormente han dejado de recomendarse. Aunque se mantienen especificaciones de versiones anteriores por cuestiones de compatibilidad, estas ya no se recomiendan, como es el caso de la programación de páginas web en JSP, sustituida en la versión nueva por facelets, más intuitivos y sencillos de usar.

Utilizar una tecnología tan nueva no quiere decir, ni mucho menos, que ésta esté exenta de problemas. Al inicio del proyecto el soporte a esta tecnología no estaba soportado por muchos programas y, por ejemplo, el servidor de aplicaciones JBoss únicamente era compatible con JEE 6 en su última versión, que era una versión Beta, con lo que podían aparecer problemas difíciles de solventar. El servidor de aplicaciones Glassfish sí que soporta esta versión desde la 3.0, pero tampoco está exento de problemas.

### **Internacionalización**

El programa realizado está pensado para poder ser utilizado en diferentes idiomas. Durante la implementación se ha tenido en cuenta este aspecto y se han utilizado ficheros de texto para mostrar todos los mensajes en los diferentes idiomas en los

que se pueden mostrar. En la actualidad únicamente se muestran los mensajes en castellano, aunque no es difícil añadir otros idiomas.

## Entorno de trabajo

Para poder realizar un programa en JEE mínimamente extenso, es necesario disponer de una herramienta de trabajo que facilite las tareas de configuración y generación del proyecto. Existen múltiples entornos de desarrollo integrado (IDE) que van desde programas comerciales hasta programas de código abierto. Para la realización de este proyecto me planteé 2 opciones, por un lado el IDE de Eclipse y por otro Netbeans. Aunque ambos se adaptaban bien a mis necesidades, me decanté por Netbeans debido a que su integración en JEE 6 estaba más avanzada y era menos compleja que Eclipse. Este IDE viene con el servidor de aplicaciones Glassfish, adaptado a los nuevos estándares de JEE 6.

Para mejorar el aspecto del programa y facilitar el manejo de datos, me he apoyado en la extensión primefaces, un framework que recoge diferentes herramientas relacionadas con JSF 2 y la librería en javascript jQuery.

Para el control de versiones he utilizado git y para el control de librerías maven.

## Gestor de base de datos

En el marco de las bases de datos me encontré con el mismo problema que con los IDE, existen múltiples programas que se adaptaban a las necesidades de exafarm. Sistemas de gestión de bases de datos (SGBD) como MySQL, Postgres o incluso Derby cumplían con todos los requisitos necesarios para ser candidatos a utilizarse junto con exafarm. Finalmente me decidí por MySQL por tratarse de un SGBD con licencia de uso gratuita plenamente integrable en JEE.

## Planificación del proyecto

En la planificación del proyecto se han tenido en cuenta diferentes fases de trabajo y temporalización que se muestran a continuación.

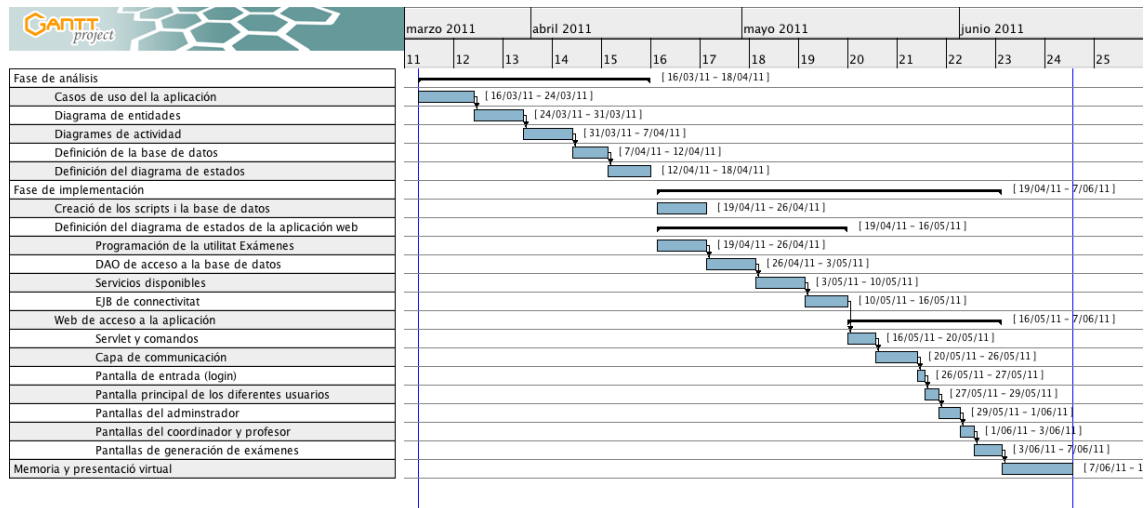


Ilustración 1. Tabla de temporalización y funciones del programa

## Composición del programa

El programa se puede dividir en diferentes módulos según su diferente funcionalidad:

- **Módulo de conexión y seguridad:** Este módulo realiza tareas de autenticación y autorización de los usuarios y aspectos relacionados con la seguridad. Sigue las recomendaciones de seguridad de JAAS un protocolo de JEE 6.
- **Módulo de gestión de usuarios y temas:** Este módulo se encarga de la gestión de usuarios (alta, baja) y del alta y la baja de los temas de la asignatura.
- **Módulo de gestión de exámenes:** Relacionado con la gestión y generación de exámenes y sus diferentes versiones.
- **Módulo de gestión de preguntas:** Relacionado con la gestión de las preguntas que se pueden agregar a los exámenes.

## Arquitectura básica del proyecto

El programa se basa en una arquitectura por capas siguiendo un modelo MVC (Modelo, Vista, Controlador).

En el servidor Java EE tenemos el contenedor EJB que es el único que accede a la capa de datos, en este contenedor es donde desplegaremos el módulo EJB. El contenedor web atiende las peticiones http y es donde se despliega el módulo Web. Los usuarios podrán acceder al sistema remotamente a través de un navegador web utilizando peticiones http.

En el siguiente esquema se puede observar la arquitectura utilizada en el proyecto.

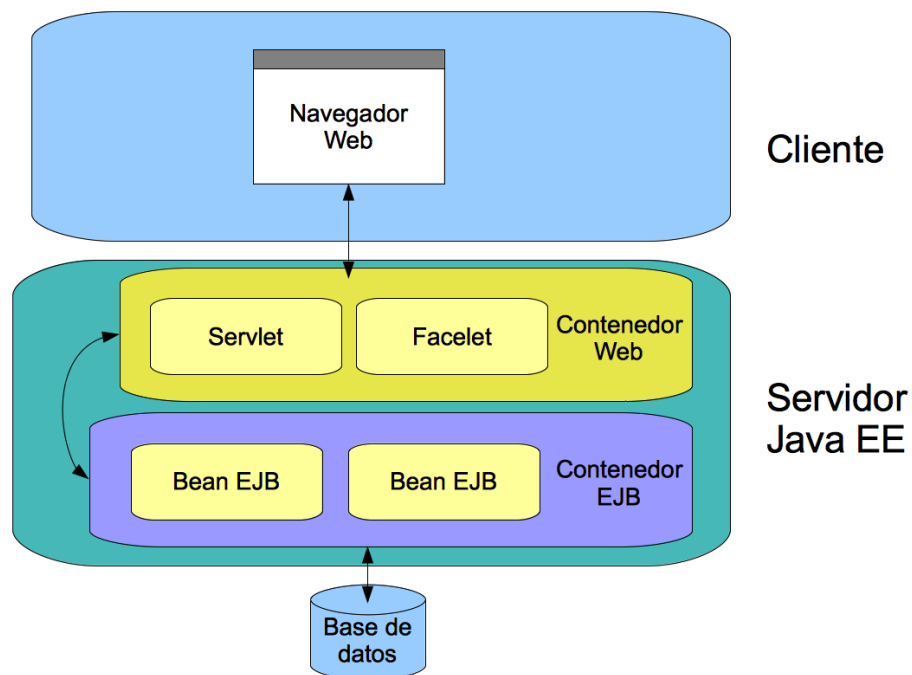
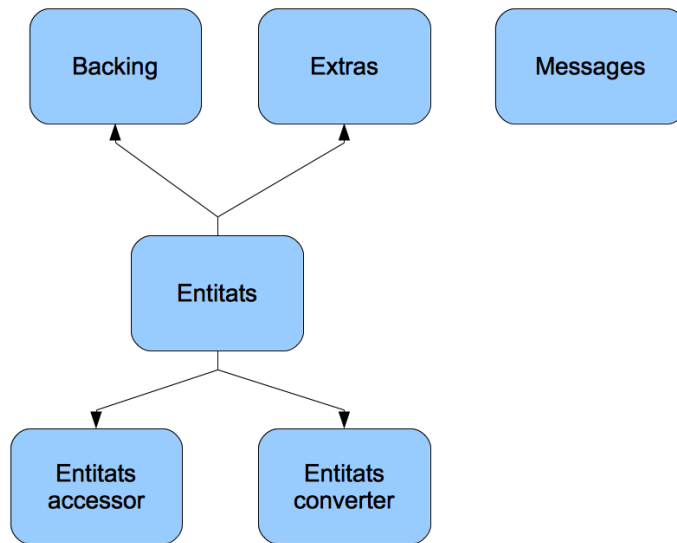


Ilustración 2. Arquitectura básica del proyecto

## Diagrama de paquetes

Para facilitar el desarrollo de la aplicación se han creado diferentes paquetes que agrupan las clases. En el siguiente esquema se muestran estos paquetes:



**Ilustración 3. Diagrama de paquetes**

**Entitats:** Paquete que agrupa todas las clases de entidad, que serán las que se corresponderán con la capa de persistencia.

**Entitats accessor:** Paquete que agrupa las clases encargadas del acceso a la base de datos.

**Entitats converter:** Clases encargadas de la conversión de tipos para la presentación web.

**Backing:** Backing beans, relacionada con los Managedbeans.

Extras: Clases con funcionalidad extra.

**Messages:** Clase que agrupa los ficheros de internacionalización.

## Casos de uso

### Identificación de los actores

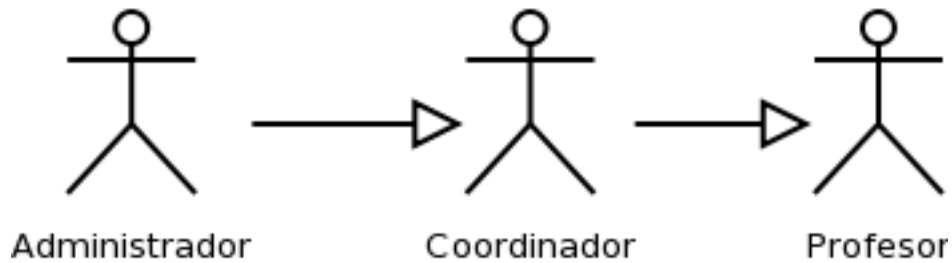


Ilustración 4. Relación de actores del programa

El programa acepta 3 tipos de actores cuyas características principales se describen a continuación:

- **Administrador:** Encargado de la gestión de usuarios, gestión de temas y bloques. Tiene privilegios para modificar y desactivar preguntas de todos los usuarios.
- **Coordinador:** Encargado de la gestión de exámenes y generación de los mismos. Tiene privilegios para modificar y desactivar preguntas de todos los usuarios.
- **Profesor:** Encargado de la gestión de preguntas propias y selección de preguntas para determinados exámenes. Únicamente pueden gestionar las preguntas propias.

### Casos de uso

En el siguiente diagrama se muestran los casos de uso más significativos del programa:

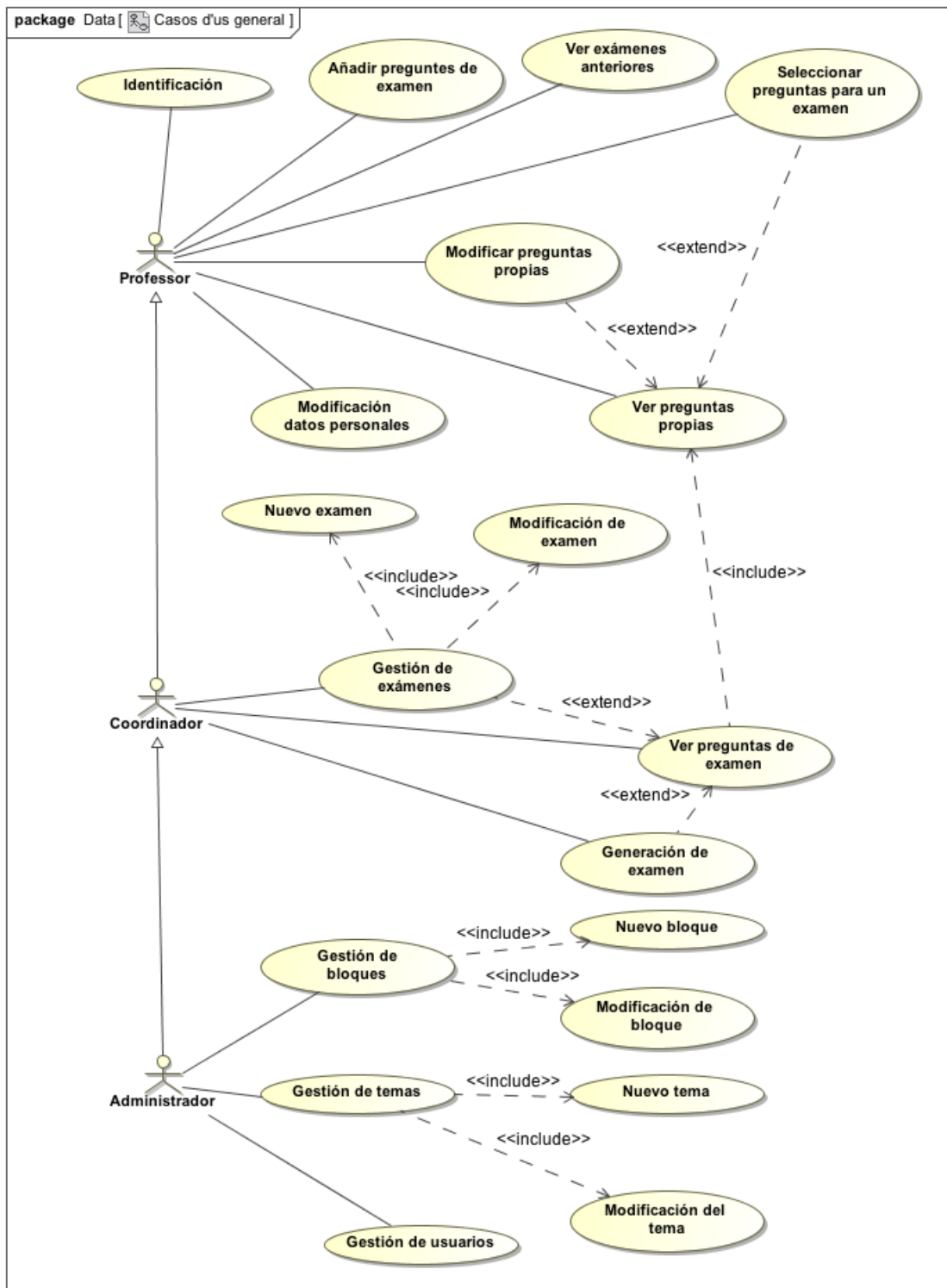


Ilustración 5. Diagrama general de casos de uso



## Fichas de los casos de uso

1. **Caso de uso:** Identificación
  - 1.1. **Actor principal:** Profesor
  - 1.2. **Precondición:** El usuario ha de estar dado de alta y ha de estar activo.
  - 1.3. **Postcondición:** El profesor estará identificado como usuario.
  - 1.4. **Casos de uso relacionados:** Todos los demás casos de uso.
2. Escenario principal
  - 2.1. Aparecen 2 campos de formulario.
  - 2.2. El usuario ha de rellenar los campos de nombre de usuario y clave
  - 2.3. El usuario pulsa sobre el botón “Enviar”
  - 2.4. El usuario es identificado y autenticado y es dirigido a la página principal según su función (Administrador, Coordinador o profesor)
3. Flujos alternativos
  - 3.1. El profesor puede desconectarse del sistema en cualquier momento pulsando sobre el enlace “Desconectar” en la cabecera de la página.
  - 3.2. En este caso dejará de poder acceder al sistema.



The screenshot shows the login interface for 'exafarm'. At the top, the logo 'exafarm' is displayed in a grey sans-serif font, with the 'x' in green. Below the logo, the text 'programa preparador de exámenes de farmacología' is shown, with 'farmacología' in green. A message in grey text reads: 'Por favor, introduzca nombre de usuario y contraseña para acceder a la aplicación.' Below this, there are two input fields: 'Usuario:' followed by a text box, and 'Clave:' followed by a text box. A grey button labeled 'Enviar' is positioned below the password field.

Ilustración 6. Pantalla de caso de uso Identificación

1. **Caso de uso:** Añadir preguntas de examen
  - 1.1. **Actor principal:** Profesor
  - 1.2. **Precondición:** El profesor ha de estar dado de alta, estar activo e identificado por el sistema
  - 1.3. **Postcondición:** En la base de datos habrá una nueva pregunta
  - 1.4. **Casos de uso relacionados:** Identificación
2. Escenario principal
  - 2.1. El profesor introduce la pregunta y las respuestas.
  - 2.2. Es necesario que introduzca texto en todos los campos.
  - 2.3. Ha de seleccionar un tema para la pregunta
  - 2.4. Ha de seleccionar al menos una respuesta correcta
3. Flujos alternativos
  - 3.1. Si el profesor sale sin guardar, la pregunta no se agregará a la base de datos

Exafarm - Añadir pregunta

Pregunta:  Activa:

Respuesta 1:  Correcta:

Respuesta 2:  Correcta:

Respuesta 3:  Correcta:


Respuesta 4:  Correcta:

Respuesta 5:  Correcta:

Tema:

Mezclar las respuestas:

Ilustración 7. Pantalla de añadir pregunta

1. **Caso de uso:** Ver preguntas
  - 1.1. **Actor principal:** Profesor
  - 1.2. **Precondición:** El profesor ha de estar activo e identificado.
  - 1.3. **Postcondición:** ---
  - 1.4. **Casos de uso relacionados:** Ver preguntas, Modificar preguntas, Identificación
2. Escenario principal
  - 2.1. Aparece un listado con las preguntas del profesor. La lista muestra el id de la pregunta, el texto de la pregunta y respuestas, el tema y bloque, el profesor que ha realizado la pregunta (únicamente en el caso de coordinadores o administradores), un botón para modificar datos de la pregunta (tema y profesor), un botón para modificar el texto de la pregunta  (solo en el caso de que la pregunta no haya sido incluida en ningún examen cerrado).
  - 2.2. Aparecerá un botón de duplicar en el caso de que la pregunta se encuentre en un examen cerrado.
  - 2.3. Es posible filtrar el listado por Id, temas.
  - 2.4. En el caso de que el profesor tenga permisos de coordinador o administrador, también podrá filtrar por profesor.








Listado de preguntas								
<span style="float: right;">1 2 3 &gt;&gt; &lt;&lt;</span>								
Id	Pregunta	Tema-Bloque Todos	Profesor Todos	Activa	Mezclable	Modificar	Ver	Duplicar
1	Señale la afirmación falsa: a. En el inicio de maniobras de RCP se considera implícito el consentimiento por parte del paciente. b. No es necesaria una orden médica para iniciar la RCP. c. Es necesaria una orden médica para interrumpir las maniobras de RCP. d. Aunque las secuelas pueden ser graves tras una RCP, debe realizarse en todas las PCR. e. El papel del público no sanitario, en el medio extrahospitalario, es fundamental en el inicio y realización de la REP.	Definición y objetivos-Teoría	Gloria Manso Rodríguez	Sí	Sí			
2	Cual de estas afirmaciones es cierta: a. La RCP básica carece de utilidad. b. En España los testigos de un paro cardíaco habitualmente no realizan la RCP básica. c. La desfibrilación semiautomática solo debe realizarla el médico. d. La RCP avanzada solo debe efectuarse en el hospital. e. El objetivo de posibilitar la desfibrilación en los primeros 5 min. de un paro cardíaco es una recomendación con bajo nivel de evidencia.	Información y automedicación-Teoría	Franz Jimeno Demuth	Sí	No			
3	En situación de RCP avanzada, señale la correcta: a. La monitorización eléctrica inicial se debe realizar a través de electrodos. b. Una vez que se ha hecho una descarga a través de pañales, no podemos monitorizar de nuevo con ellas. c. No podemos conocer el ritmo cardíaco si monitorizamos con pañales. d. No podemos conocer el verdadero ritmo cardíaco si no realizamos un ECG de 12 derivaciones. e. La monitorización a través de electrodos no debe interrumpir las	Definición y objetivos-Teoría	Gloria Manso Rodríguez	Sí	No			

Ilustración 8. Pantalla de listado de preguntas

1. **Caso de uso:** Seleccionar preguntas para examen
  - 1.1. **Actor principal:** Profesor
  - 1.2. **Precondición:** El profesor ha de estar activo e identificado.
  - 1.3. **Postcondición:** Las preguntas seleccionadas se incluirán en el examen activo.
  - 1.4. **Casos de uso relacionados:** Identificación
2. Escenario principal
  - 2.1. Aparece una tabla con los exámenes activos.
  - 2.2. El usuario seleccionará el examen activo
  - 2.3. Aparecerá una tabla con las preguntas activas del usuario con un botón de selección junto a cada pregunta
  - 2.4. Una vez seleccionadas las preguntas el usuario pulsará sobre el botón de guardar.
3. Flujos alternativos
  - 3.1. El usuario si no pulsa sobre el botón de guardar, no se guardará la selección de preguntas.

Listado de exámenes activos.			
Id	Fecha	Activo	Seleccionar
1	29 de enero de 2010	Si	<input type="button" value="⌵"/>
2	9 de enero de 2011	No	<input type="button" value="⌵"/>
3	19 de enero de 2011	No	<input type="button" value="⌵"/>
4	4 de enero de 2012	No	<input type="button" value="⌵"/>
5	1 de enero de 2010	No	<input type="button" value="⌵"/>

Ilustración 9. Pantalla de listado de exámenes activos

1. **Caso de uso:** Modificar preguntas propias
  - 1.1. **Actor principal:** Profesor
  - 1.2. **Precondición:** El profesor ha de estar activo e identificado.
  - 1.3. **Postcondición:** La pregunta seleccionada se habrá modificado en la base de datos
  - 1.4. **Casos de uso relacionados:** Ver preguntas propias, Identificación
2. Escenario principal
  - 2.1. Desde el listado de preguntas propias el profesor pulsa sobre el botón de modificar.
  - 2.2. Aparece un formulario con el texto para la pregunta, las respuestas y tema.
  - 2.3. Existen unos botones de selección para seleccionar la respuesta correcta.
  - 2.4. El usuario pulsará sobre guardar para que la pregunta quede guardada en la base de datos.
3. Flujos alternativos
  - 3.1. El usuario si no pulsa sobre el botón de guardar, no se guardarán los cambios realizados en la pregunta.

**Modificar la pregunta 1**

Pregunta

Señale la afirmación falsa:

Respuestas	
Respuesta	Correcta
En el inicio de maniobras de RCP se considera implícito el consentimiento por parte del paciente.	<input type="checkbox"/>
No es necesaria una orden médica para iniciar la RCP.	<input type="checkbox"/>
Es necesaria una orden médica para interrumpir las maniobras de RCP.	<input type="checkbox"/>
Aunque las secuelas pueden ser graves tras una RCP, debe realizarse en todas las PCR.	<input checked="" type="checkbox"/>
El papel del público no sanitario, en el medio extrahospitalario, es fundamental en el inicio y realización de la REP.	<input type="checkbox"/>

Ilustración 10. Pantalla de modificación de pregunta

1. **Caso de uso:** Modificar datos personales
  - 1.1. **Actor principal:** Profesor
  - 1.2. **Precondición:** El profesor ha de estar activo e identificado.
  - 1.3. **Postcondición:** Los datos del profesor quedarán modificados en la base de datos
  - 1.4. **Casos de uso relacionados:** Identificación
2. Escenario principal
  - 2.1. Aparece un formulario con campos para la modificación de los datos del usuario siguientes:
    - Nombre
    - Apellidos
    - Email
    - Clave
  - 2.2. El usuario podrá modificar sus datos personales
  - 2.3. Una vez realizados los cambios el usuario pulsará sobre el botón de “Modificar”.
3. Flujos alternativos
  - 3.1. El usuario si no pulsa sobre el botón de modificar, no se guardará la modificación de los datos.



The screenshot shows a web form titled "Modificar datos de usuario". It contains four input fields: "Nombre" with the value "Franz", "Apellidos" with "Jimeno Demuth", "Email" with "franz@uniovi.es", and "Clave" with a masked password. A "Modificar" button is located below the password field.

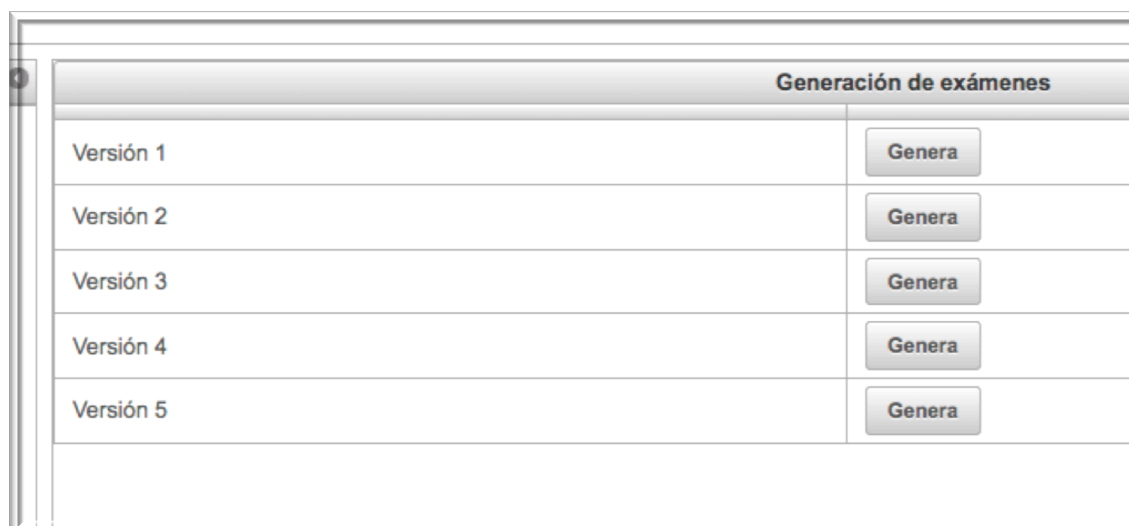
Ilustración 11. Pantalla de modificación de datos de usuario

1. **Caso de uso:** Ver examen
  - 1.1. **Actor principal:** Coordinador
  - 1.2. **Precondición:** El usuario ha de estar activo e identificado como coordinador.
  - 1.3. **Postcondición:** ---
  - 1.4. **Casos de uso relacionados:** Identificación, listado de exámenes, generación de examen
2. Escenario principal
  - 2.1. Aparece una tabla con todos los exámenes existentes mostrando datos de la fecha de convocatoria, número de preguntas seleccionadas para el examen y el número de preguntas total, si el examen está activo y si el examen está cerrado.
  - 2.2. Se pueden modificar los datos del examamen pulsando sobre el botón de modificar.
  - 2.3. En el caso de que un examen esté activo, será posible ver las preguntas.
  - 2.4. En el caso de que un examen no esté cerrado, será posible generar el examen.
3. Flujos alternativos
  - 3.1. El usuario pulsa sobre alguna opción del menú y no selecciona ningún examen.

Listado de Exámenes								
	Fecha	Preguntas Seleccionadas	Número de versiones	Activo	Cerrado	Modificar	Ver	Generar
1	29 de enero de 2010	1/70	6	No	Sí			
2	9 de enero de 2011	0/60	5	No	No			
3	19 de enero de 2011	0/10	3	No	No			
4	4 de enero de 2012	0/4	12	No	No			
5	4 de enero de 2012	0/12	4	No	No			
6	5 de febrero de 2012	0/10	23	No	No			
7	10 de enero de 2011	0/23	99	Sí	No			
8	1 de enero de 2011	0/33	332	Sí	No			
9	1 de enero de 2011	0/3	22	Sí	No			

Ilustración 12. Pantalla de listado de exámenes

1. **Caso de uso:** Generar examen
  - 1.1. **Actor principal:** Coordinador
  - 1.2. **Precondición:** El usuario ha de estar identificado como coordinador
  - 1.3. **Postcondición:** Se habrán generado las versiones de examen seleccionadas en formato PDF con plantilla de respuestas y listado de respuestas correctas.
  - 1.4. **Casos de uso relacionados:** Identificación, Ver examen
2. Escenario principal
  - 2.1. Aparece una tabla con las versiones que se quieren generar.
  - 2.2. El usuario podrá pulsar sobre el botón de generar para descargar la versión en formato PDF, con plantilla de respuestas y listado de respuestas correctas.
3. Flujos alternativos
  - 3.1. El usuario si no pulsa sobre el botón de generar, no se generarán las versiones.



Generación de exámenes	
Versión 1	<input type="button" value="Genera"/>
Versión 2	<input type="button" value="Genera"/>
Versión 3	<input type="button" value="Genera"/>
Versión 4	<input type="button" value="Genera"/>
Versión 5	<input type="button" value="Genera"/>

Ilustración 13. Pantalla de generación de exámenes



1. **Caso de uso:** Añadir examen
  - 1.1. **Actor principal:** Coordinador
  - 1.2. **Precondición:** El usuario ha de estar identificado como coordinador.
  - 1.3. **Postcondición:** El examen se añadirá a la base de datos.
  - 1.4. **Casos de uso relacionados:** Identificación
2. Escenario principal
  - 2.1. Aparece un formulario con los datos del examen:
    - Fecha de convocatoria
    - Número de preguntas
    - Número de versiones
    - Activo
    - Cerrado
  - 2.2. El usuario introducirá los datos del examen.
  - 2.3. Una vez rellenados los campos el usuario pulsará sobre el botón de añadir.
3. Flujos alternativos
  - 3.1. El usuario si no pulsa sobre el botón de añadir, no se guardará el examen nuevo.

**Añadir examen**

Fecha de convocatoria (dd/mm/aaaa):

Número de preguntas:

Número de versiones:

Activo:

Cerrado:

**Añadir**

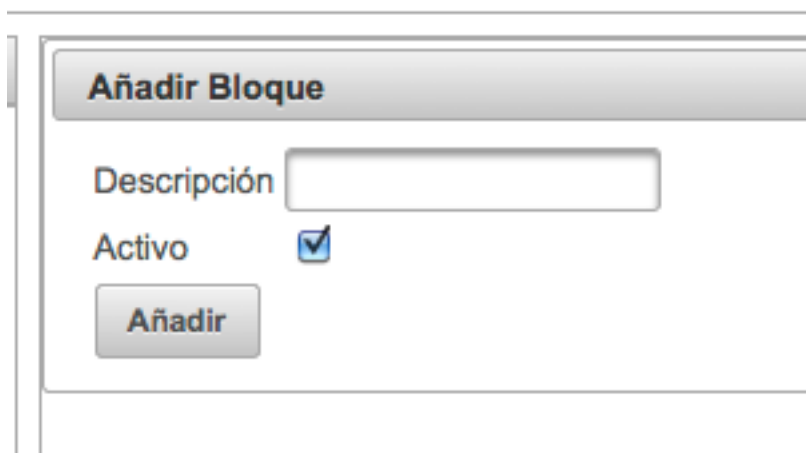
Ilustración 14. Pantalla de añadir examen

1. **Caso de uso:** Ver bloques
  - 1.1. **Actor principal:** Coordinador
  - 1.2. **Precondición:** El usuario ha de estar identificado como Administrador
  - 1.3. **Postcondición:** ---
  - 1.4. **Casos de uso relacionados:** Identificación
2. Escenario principal
  - 2.1. Aparece una tabla con los bloques que existen para la asignatura.
  - 2.2. El usuario podrá modificar los datos pulsando sobre el botón de modificar .
  - 2.3. En el caso de modificar un bloque el usuario deberá pulsar sobre el botón de aceptar para guardar las modificaciones.
  - 2.4. En el caso de pulsar sobre cancelar, no se guardarán los cambios.

Modificar bloque		
Listado de bloques		
Descripción	Activo	Modificar
Teoría	<input checked="" type="checkbox"/>	✓ ✕
Seminarios	Sí	✎

Ilustración 15. Pantalla de modificar bloque

1. **Caso de uso:** Añadir bloque
  - 1.1. **Actor principal:** Administrador
  - 1.2. **Precondición:** El usuario ha de estar identificado como administrador.
  - 1.3. **Postcondición:** Los datos del bloque se añadirán a la base de datos.
  - 1.4. **Casos de uso relacionados:** Identificación
2. Escenario principal
  - 2.1. Aparece un formulario con los datos del bloque:
    - Descripción
    - Activo
  - 2.2. El usuario introducirá los datos en los campos.
  - 2.3. Para guardar los datos el usuario pulsará sobre Añadir
3. Flujos alternativos
  - 3.1. El usuario si no pulsa sobre el botón de añadir, no se guardará el bloque.



**Añadir Bloque**

Descripción

Activo

**Añadir**

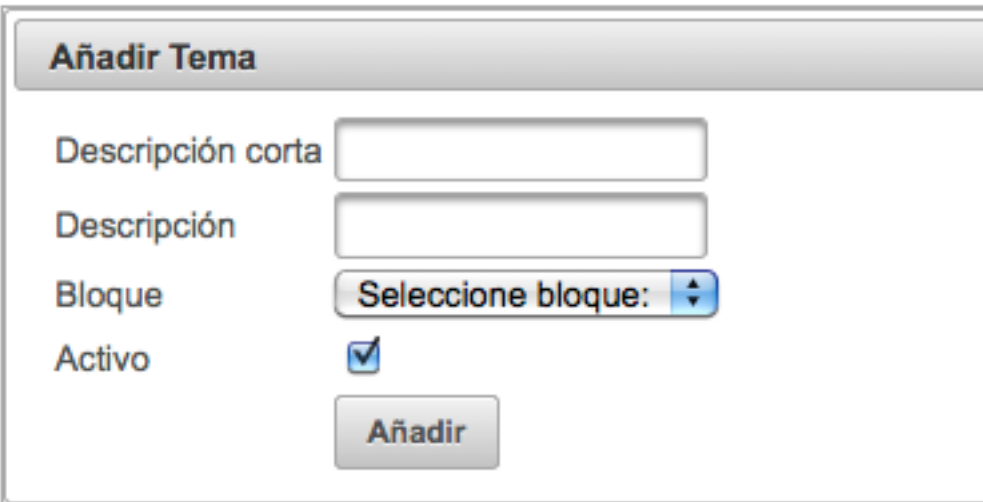
Ilustración 16. Pantalla de añadir bloque

1. **Caso de uso:** Ver temas
  - 1.1. **Actor principal:** Administrador
  - 1.2. **Precondición:** El usuario ha de estar identificado como administrador.
  - 1.3. **Postcondición:** Los temas se habrán modificado si el usuario los modificó.
  - 1.4. **Casos de uso relacionados:** Identificación
2. Escenario principal
  - 2.1. Aparece una tabla con los temas de la base de datos.
  - 2.2. La tabla se puede ordenar por Descripción corta, Descripción, Bloque y Activo.
  - 2.3. El usuario podrá modificar los temas de la tabla pulsando sobre el botón de modificar.
  - 2.4. Para guardar los cambios pulsará sobre el botón de aceptar.
  - 2.5. Para cancelar los cambios pulsará sobre el botón de cancelar.

Listado de Temas				
Descripción corta	Descripción	Bloque	Activo	Modificar
Definición y objetivos	Definición y objetivos de la Farmacología Clínica	Teoría	<input checked="" type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>
Información y automedicación	Información sobre medicamentos. Automedicación.	Teoría	Sí	<input type="checkbox"/>
Cumplimiento	Cumplimiento de la prescripción.	Teoría	Sí	<input type="checkbox"/>
Especialidades farmacéuticas	Especialidades Farmacéuticas. Tipos.	Teoría	Sí	<input type="checkbox"/>

Ilustración 17. Pantalla de listar temas

1. **Caso de uso:** Añadir tema
  - 1.1. **Actor principal:** Administrador
  - 1.2. **Precondición:** El usuario ha de estar identificado como administrador.
  - 1.3. **Postcondición:** El tema se habrá añadido a la base de datos.
  - 1.4. **Casos de uso relacionados:** Identificación
2. Escenario principal
  - 2.1. Aparece una tabla con los temas de la base de datos.
  - 2.2. La tabla se puede ordenar por Descripción corta, Descripción, Bloque y Activo.
  - 2.3. El usuario podrá modificar los temas de la tabla pulsando sobre el botón de modificar.
  - 2.4. Para guardar los cambios pulsará sobre el botón de aceptar.
  - 2.5. Para cancelar los cambios pulsará sobre el botón de cancelar.



**Añadir Tema**

Descripción corta

Descripción

Bloque

Activo

Ilustración 18. Pantalla de añadir tema

## Diagramas de actividad

A continuación se muestran los diagramas de actividad más significativos de la aplicación:

## Selección de preguntas

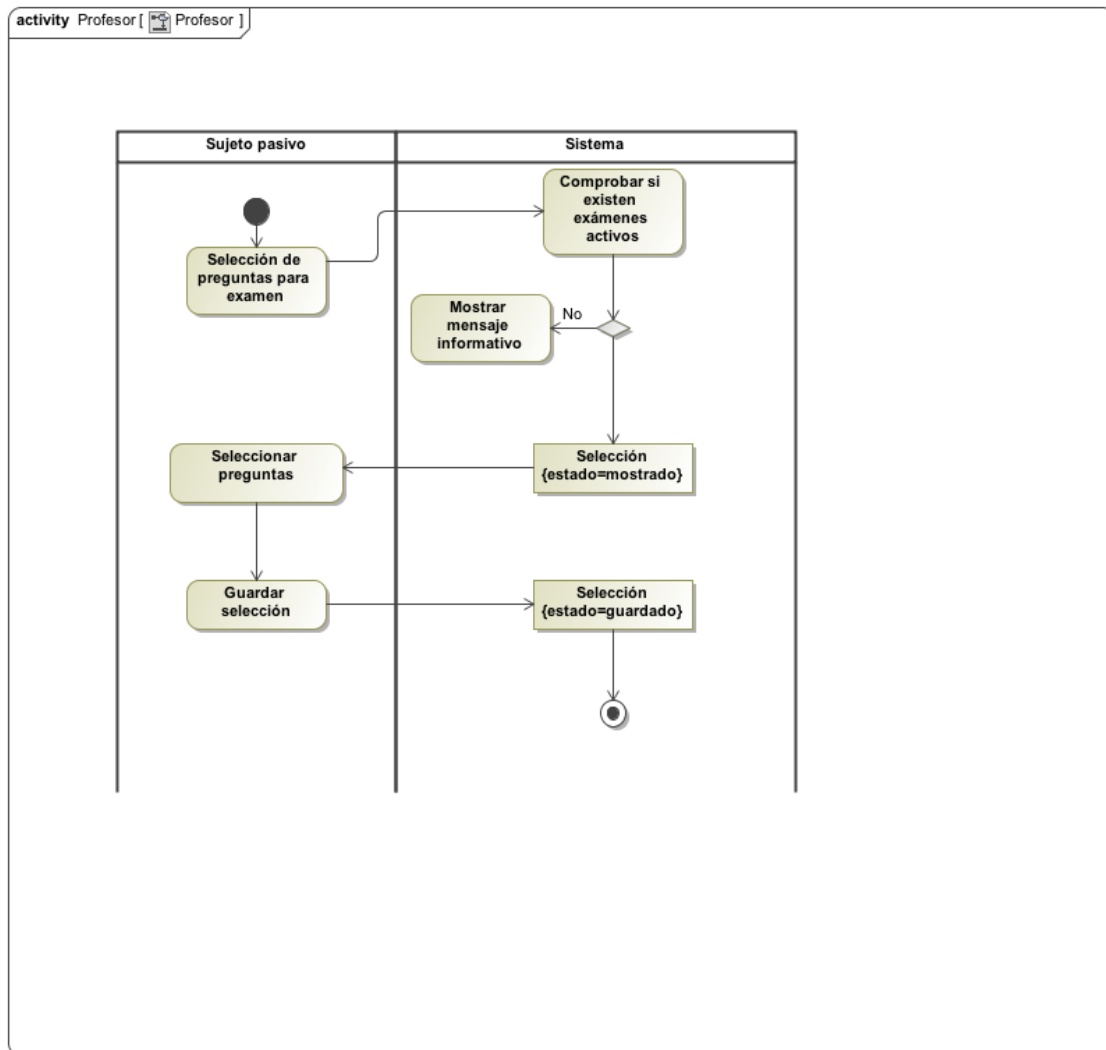


Ilustración 19. Diagrama de actividad de selección de preguntas

## Generar examen

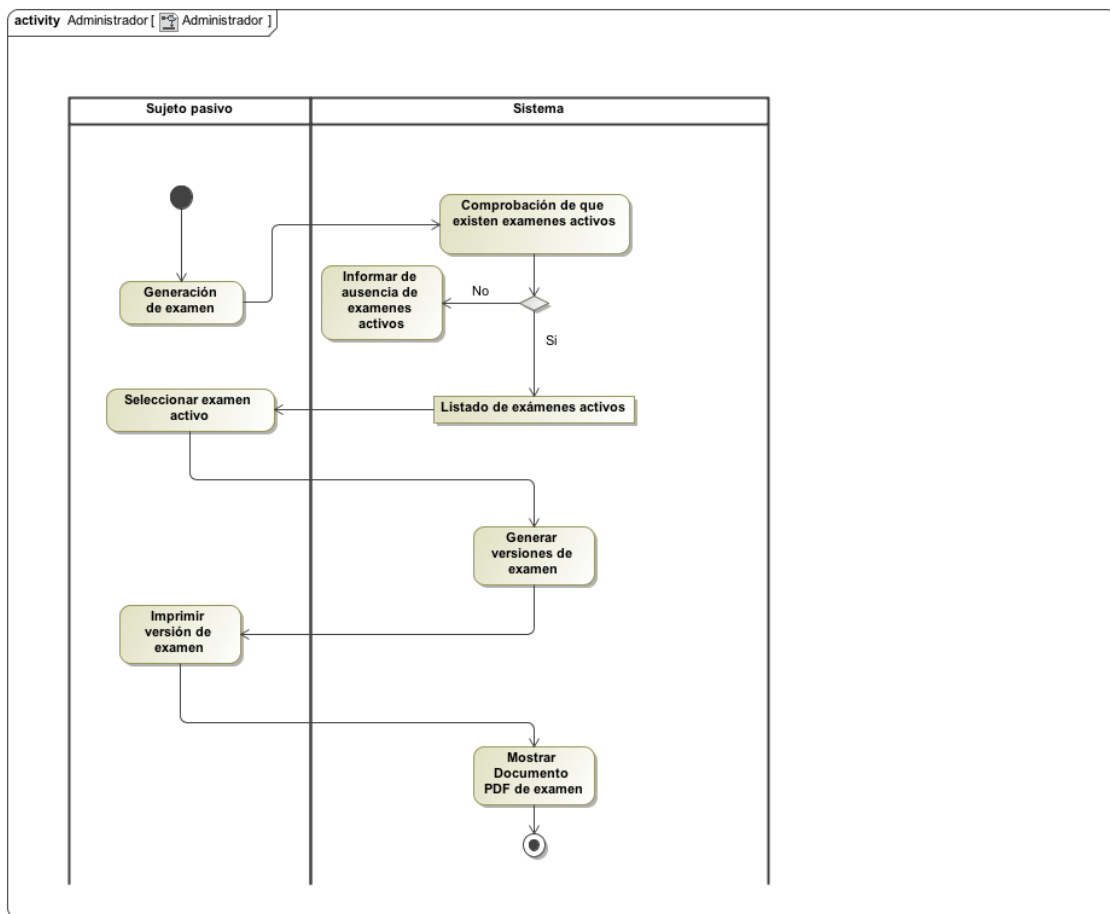


Ilustración 20. Diagrama de actividad de generación de examen

## Estructura de la base de datos

En la siguiente figura se muestra la estructura de la base de datos relacional:

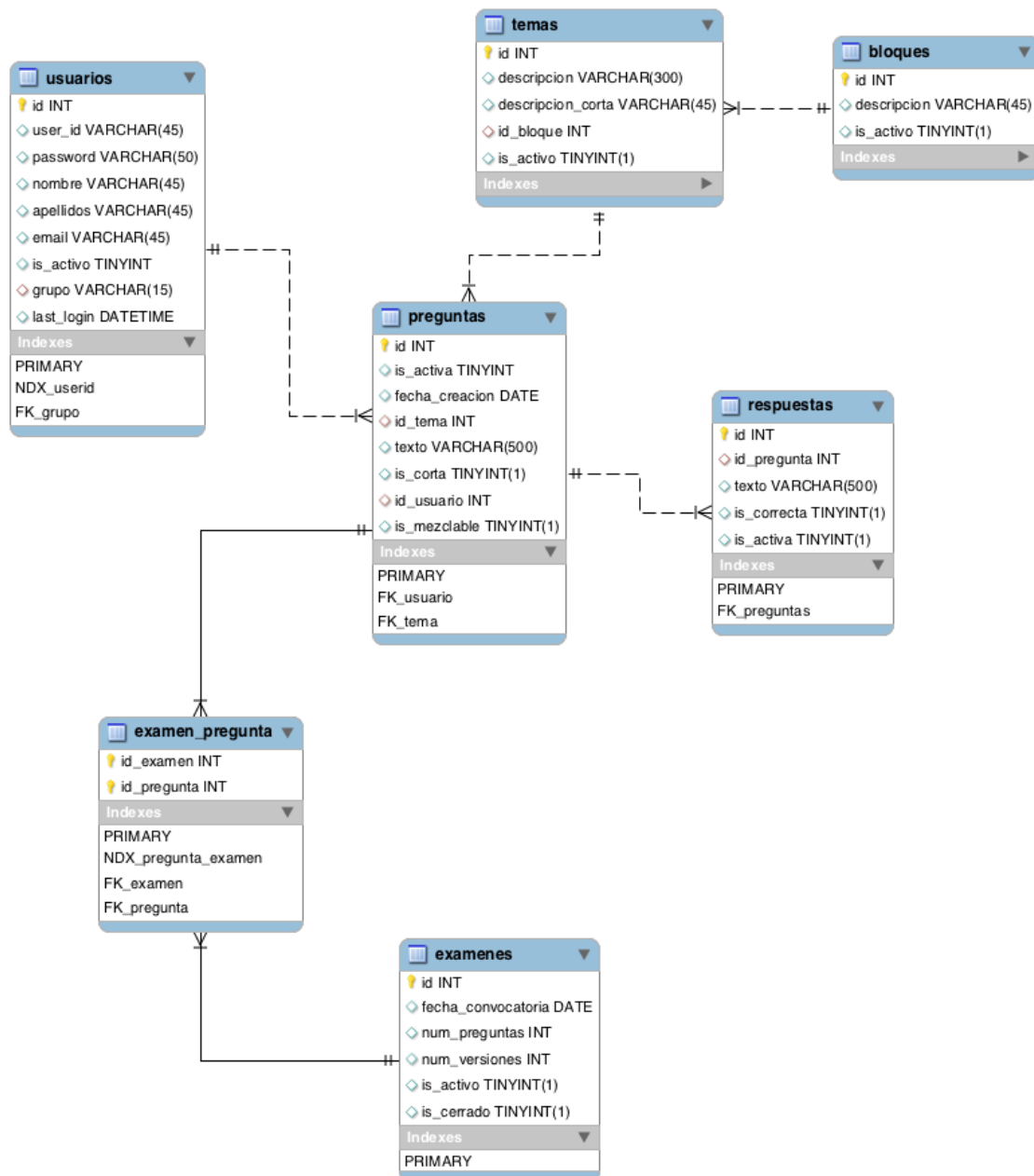


Ilustración 21. Esquema de la base de datos

## Requerimientos de software

Para el correcto funcionamiento del programa son necesarios diferentes programas que deberán estar instalados en el servidor y el cliente.



## Requerimientos para el servidor

### Máquina virtual de Java (JVM)

Es necesario haber instalado en el servidor una JVM que cumpla con los estándares JEE 6, en este caso se recomienda la última versión en el momento de la creación de este proyecto (Versión 6 Update 26).

Ésta se puede descargar de:

<http://www.java.com/es/download>

### Servidor de aplicaciones

El servidor de aplicaciones utilizado en este proyecto es Glassfish 3.1, que cumple con los requisitos para poder utilizar todos los requerimientos de JEE 6.

## Requerimientos para el cliente

En el lado del cliente únicamente es necesario un navegador web que soporte javascript. El diseño de la página ha sido realizado a en una resolución de 1280x800 puntos, aunque se ha tenido en cuenta que las pantallas podían ser menores, adaptándose la presentación a la resolución de las mismas mostrando barras de desplazamiento.

## Conclusiones

Considero que el programa que he realizado como proyecto de final de carrera puede llegar a ser un programa de gran utilidad para el profesorado, facilitando el trabajo de los mismos. En esta primera versión el programa cubre las necesidades básicas, aunque en versiones posteriores se podrán tener en cuenta nuevos aspectos que harán la aplicación más flexible.

En el momento de decidir el tema de trabajo del proyecto final de carrera, me pareció que J2EE me podía ser de utilidad en futuros trabajos. El haber programado en Java durante toda la carrera influyó en mi decisión. Durante la fase de documentación y aprendizaje me di cuenta de que J2EE es mucho más que un lenguaje de programación, es un modo de programar, una forma de pensar que hasta ese momento no me había planteado. Ese aspecto hizo que el aprendizaje

inicial fuera difícil. El que existieran múltiples versiones de los diferentes frameworks y especificaciones lo hizo todo más complicado. Afortunadamente, la curva de aprendizaje del lenguaje fue disminuyendo la pendiente conforme avanzaba en mis estudios. Esta forma de programar tan estructurada, una vez se entiende y se utiliza, facilita mucho la tarea del programador y permite que proyectos de gran envergadura puedan ser gestionados de forma mucho más sencilla, sin llegar a la locura una vez se adentra uno en el código fuente.

## Bibliografía

BOOCH, GRADY; RUMBAUGH, JAMES; JACOBSON, IVAR. *El lenguaje unificado de modelado*. Addison Wesley

BURNS, ED; SCHALK, CHRIS. *JavaServer Faces 2.0, The Complete Reference*. Ed. Mc Graw Hill

CEBALLOS, FRANCISCO JAVIER. *Java 2 Curso de programación*. Ed. Ra-Ma

GEARY, DAVID M.; *Core JavaServer Faces*. Ed. Prentice Hall 3rd edition

GONCALVEZ, ANTONIO. *Beginning Java EE 6 Platform with GlassFish 3*. Ed. Apress

JENDROCK, ERIC; EVANS, IAN; GOLLAPUDI, DEVIKA; HAASE, KIM; SRIVATHSA, CHIMAYEE *The Java EE 6 Tutorial*. Ed. Oracle  
(<http://download.oracle.com/javaee/6/tutorial/doc/>) (consultado en mayo 2011)

LIMA DIAZ, FELIPE; *Java 6*. Ed. Anaya Multimedia

## Anexo I: Instalación del programa

### Preparación de la base de datos

El SGBD utilizado para este programa es MySQL. Es necesario tenerlo instalado. El programa de instalación del servidor se puede encontrar en:

<http://www.mysql.com/downloads/mysql/>

(dirección válida el 16/6/2011)

### Creación de la base de datos

En el fichero zip adjunto se encuentran los archivos usuarios.sql (sentencias ddl), estructura.sql(sentencias ddl), datos.sql (sentencias dml). Estos contienen las instrucciones para la creación de usuarios, la generación de la base de datos y tablas, y la inserción de datos de ejemplo de la aplicación.

A continuación se muestran los mismos en formato texto:

Fichero *usuarios.sql* crea 2 usuarios. Por un lado *exafarm\_user* que será el usuario utilizado por el programa *exafarm* con permisos para insertar seleccionar y modificar datos y *exafar\_admin*, usuario con todos los privilegios para poder gestionar la base de datos desde el propio SGBD:

```
CREATE USER `exafarm_user` IDENTIFIED BY 'UOC';

grant INSERT on TABLE `exafarm`.`bloques` to exafarm_user;
grant SELECT on TABLE `exafarm`.`bloques` to exafarm_user;
grant UPDATE on TABLE `exafarm`.`bloques` to exafarm_user;
grant INSERT on TABLE `exafarm`.`examenes` to exafarm_user;
grant SELECT on TABLE `exafarm`.`examenes` to exafarm_user;
grant UPDATE on TABLE `exafarm`.`examenes` to exafarm_user;
grant INSERT on TABLE `exafarm`.`examen_pregunta` to exafarm_user;
grant SELECT on TABLE `exafarm`.`examen_pregunta` to exafarm_user;
grant UPDATE on TABLE `exafarm`.`examen_pregunta` to exafarm_user;
```

```

grant INSERT on TABLE `exafarm`.`grupos` to exafarm_user;
grant SELECT on TABLE `exafarm`.`grupos` to exafarm_user;
grant UPDATE on TABLE `exafarm`.`grupos` to exafarm_user;
grant INSERT on TABLE `exafarm`.`preguntas` to exafarm_user;
grant SELECT on TABLE `exafarm`.`preguntas` to exafarm_user;
grant UPDATE on TABLE `exafarm`.`preguntas` to exafarm_user;
grant INSERT on TABLE `exafarm`.`respuestas` to exafarm_user;
grant SELECT on TABLE `exafarm`.`respuestas` to exafarm_user;
grant UPDATE on TABLE `exafarm`.`respuestas` to exafarm_user;
grant INSERT on TABLE `exafarm`.`temas` to exafarm_user;
grant SELECT on TABLE `exafarm`.`temas` to exafarm_user;
grant UPDATE on TABLE `exafarm`.`temas` to exafarm_user;
grant INSERT on TABLE `exafarm`.`usuarios` to exafarm_user;
grant SELECT on TABLE `exafarm`.`usuarios` to exafarm_user;
grant UPDATE on TABLE `exafarm`.`usuarios` to exafarm_user;
CREATE USER `exafarm_admin` IDENTIFIED BY 'UOC';

grant ALL on TABLE `exafarm`.`bloques` to exafarm_admin;
grant ALL on TABLE `exafarm`.`examenes` to exafarm_admin;
grant ALL on TABLE `exafarm`.`examen_pregunta` to exafarm_admin;
grant ALL on TABLE `exafarm`.`grupos` to exafarm_admin;
grant ALL on TABLE `exafarm`.`preguntas` to exafarm_admin;
grant ALL on TABLE `exafarm`.`respuestas` to exafarm_admin;
grant ALL on TABLE `exafarm`.`temas` to exafarm_admin;
grant ALL on TABLE `exafarm`.`usuarios` to exafarm_admin;

```

### Fichero “*estructura.sql*”:

```

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

DROP SCHEMA IF EXISTS `exafarm` ;
CREATE SCHEMA IF NOT EXISTS `exafarm` DEFAULT CHARACTER SET utf8
COLLATE utf8_spanish_ci ;
USE `exafarm` ;

-- -----
-- Table `exafarm`.`grupos`
-- -----

```

```

DROP TABLE IF EXISTS `exafarm`.`grupos` ;

CREATE TABLE IF NOT EXISTS `exafarm`.`grupos` (
  `id_grupo` VARCHAR(15) NOT NULL ,
  `descripcion` VARCHAR(45) NULL ,
  PRIMARY KEY (`id_grupo`) )
ENGINE = InnoDB;

-----
-- Table `exafarm`.`usuarios`
-----

DROP TABLE IF EXISTS `exafarm`.`usuarios` ;

CREATE TABLE IF NOT EXISTS `exafarm`.`usuarios` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `user_id` VARCHAR(45) NULL ,
  `password` VARCHAR(50) NULL ,
  `nombre` VARCHAR(45) NULL ,
  `apellidos` VARCHAR(45) NULL ,
  `email` VARCHAR(45) NULL ,
  `is_activo` TINYINT NULL ,
  `grupo` VARCHAR(15) NULL ,
  `last_login` DATETIME NULL ,
  PRIMARY KEY (`id`) ,
  INDEX `NDX_userid` (`user_id` ASC) ,
  INDEX `FK_grupo` (`grupo` ASC) ,
  CONSTRAINT `FK_grupo`
    FOREIGN KEY (`grupo`)
    REFERENCES `exafarm`.`grupos` (`id_grupo`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB,
COMMENT = 'Datos de usuario' ;

-----
-- Table `exafarm`.`examenes`
-----

DROP TABLE IF EXISTS `exafarm`.`examenes` ;

```

```

CREATE TABLE IF NOT EXISTS `exafarm`.`examenes` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `fecha_convocatoria` DATE NULL ,
  `num_preguntas` INT NULL ,
  `num_versiones` INT NULL ,
  `is_activo` TINYINT(1) NULL ,
  `is_cerrado` TINYINT(1) NULL ,
  PRIMARY KEY (`id`) )
ENGINE = InnoDB;

```

```

-----
-- Table `exafarm`.`bloques`
-----

```

```

DROP TABLE IF EXISTS `exafarm`.`bloques` ;

```

```

CREATE TABLE IF NOT EXISTS `exafarm`.`bloques` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `descripcion` VARCHAR(45) NULL ,
  `is_activo` TINYINT(1) NULL ,
  PRIMARY KEY (`id`) )
ENGINE = InnoDB;

```

```

-----
-- Table `exafarm`.`temas`
-----

```

```

DROP TABLE IF EXISTS `exafarm`.`temas` ;

```

```

CREATE TABLE IF NOT EXISTS `exafarm`.`temas` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `descripcion` VARCHAR(300) NULL ,
  `descripcion_corta` VARCHAR(45) NULL ,
  `id_bloque` INT NULL ,
  `is_activo` TINYINT(1) NULL ,
  PRIMARY KEY (`id`) ,
  INDEX `FK_bloques` (`id_bloque` ASC) ,
  CONSTRAINT `FK_bloques`
    FOREIGN KEY (`id_bloque`)
    REFERENCES `exafarm`.`bloques` (`id`)
    ON DELETE NO ACTION

```

```

        ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

-- Table `exafarm`.`preguntas`
-----

DROP TABLE IF EXISTS `exafarm`.`preguntas` ;

CREATE TABLE IF NOT EXISTS `exafarm`.`preguntas` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `is_activa` TINYINT NULL ,
  `fecha_creacion` DATE NULL ,
  `id_tema` INT NULL ,
  `texto` VARCHAR(500) NULL ,
  `is_corta` TINYINT(1) NULL ,
  `id_usuario` INT NULL ,
  `is_mezclable` TINYINT(1) NULL ,
  PRIMARY KEY (`id`) ,
  INDEX `FK_usuario` (`id_usuario` ASC) ,
  INDEX `FK_tema` (`id_tema` ASC) ,
  CONSTRAINT `FK_usuario`
    FOREIGN KEY (`id_usuario`)
    REFERENCES `exafarm`.`usuarios` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `FK_tema`
    FOREIGN KEY (`id_tema`)
    REFERENCES `exafarm`.`temas` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----

-- Table `exafarm`.`respuestas`
-----

DROP TABLE IF EXISTS `exafarm`.`respuestas` ;

CREATE TABLE IF NOT EXISTS `exafarm`.`respuestas` (
  `id` INT NOT NULL AUTO_INCREMENT ,

```



```

`id_pregunta` INT NULL ,
`texto` VARCHAR(500) NULL ,
`is_correcta` TINYINT(1) NULL ,
`is_activa` TINYINT(1) NULL ,
PRIMARY KEY (`id`) ,
INDEX `FK_preguntas` (`id_pregunta` ASC) ,
CONSTRAINT `FK_preguntas`
  FOREIGN KEY (`id_pregunta` )
  REFERENCES `exafarm`.`preguntas` (`id` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `exafarm`.`examen_pregunta`
-----

DROP TABLE IF EXISTS `exafarm`.`examen_pregunta` ;

CREATE TABLE IF NOT EXISTS `exafarm`.`examen_pregunta` (
  `id_examen` INT NOT NULL ,
  `id_pregunta` INT NOT NULL ,
  PRIMARY KEY (`id_examen`, `id_pregunta`) ,
  INDEX `NDX_pregunta_examen` (`id_pregunta` ASC, `id_examen` ASC) ,
  INDEX `FK_examen` (`id_examen` ASC) ,
  INDEX `FK_pregunta` (`id_pregunta` ASC) ,
  CONSTRAINT `FK_examen`
    FOREIGN KEY (`id_examen` )
    REFERENCES `exafarm`.`examenes` (`id` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `FK_pregunta`
    FOREIGN KEY (`id_pregunta` )
    REFERENCES `exafarm`.`preguntas` (`id` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

Fichero “*datos.sql*”:

```
SET SQL_MODE=@OLD_SQL_MODE;
```

```

SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

-----

-- Data for table `exafarm`.`grupos`
-----

START TRANSACTION;
USE `exafarm`;
INSERT INTO `exafarm`.`grupos` (`id_grupo`, `descripcion`) VALUES
('admin', 'Administrador');
INSERT INTO `exafarm`.`grupos` (`id_grupo`, `descripcion`) VALUES
('coord', 'Coordinador');
INSERT INTO `exafarm`.`grupos` (`id_grupo`, `descripcion`) VALUES
('profe', 'Profesor');
INSERT INTO `exafarm`.`grupos` (`id_grupo`, `descripcion`) VALUES
('ningu', 'Ninguno');

COMMIT;

-----

-- Data for table `exafarm`.`usuarios`
-----

START TRANSACTION;
USE `exafarm`;
INSERT INTO `exafarm`.`usuarios` (`id`, `user_id`, `password`,
`nombre`, `apellidos`, `email`, `is_activo`, `grupo`, `last_login`)
VALUES (1, 'gloria', 'OLwUomij2j7PPbDAioJybCRhk7o+lkyRZWkOF/v/24g=',
'Gloria', 'Manso Rodríguez', 'gloria@uniovi.es', 1, 'coord', NULL);
INSERT INTO `exafarm`.`usuarios` (`id`, `user_id`, `password`,
`nombre`, `apellidos`, `email`, `is_activo`, `grupo`, `last_login`)
VALUES (2, 'franz', 'A61rMz1EUJnH3+D/dF7FzBMw0Unvds82w67U7+oT9yU=',
'Franz', 'Jimeno Demuth', 'franz@uniovi.es', 1, 'admin', NULL);
INSERT INTO `exafarm`.`usuarios` (`id`, `user_id`, `password`,
`nombre`, `apellidos`, `email`, `is_activo`, `grupo`, `last_login`)
VALUES (3, 'esther', 'qMM+b/vcsRxn1AYf/tV4ColFgbC472ciayn6LKvVN/k=',
'Esther', 'Salgueiro', 'esther@uniovi.es', 0, 'profe', NULL);
INSERT INTO `exafarm`.`usuarios` (`id`, `user_id`, `password`,
`nombre`, `apellidos`, `email`, `is_activo`, `grupo`, `last_login`)
VALUES (4, 'admin', 'jG125bVBBBW96Qi9Te4V37Fnqchz/Eu4qB9vKrRIqRg=',
'Franz', 'Jimeno Demuth', 'franz@uniovi.es', 1, 'admin', NULL);

```

```
COMMIT;

-----
-- Data for table `exafarm`.`examenes`
-----

START TRANSACTION;
USE `exafarm`;
INSERT INTO `exafarm`.`examenes` (`id`, `fecha_convocatoria`,
`num_preguntas`, `num_versiones`, `is_activo`, `is_cerrado`) VALUES
(NULL, '2011-05-12', 70, 6, 1, 0);

COMMIT;

-----
-- Data for table `exafarm`.`bloques`
-----

START TRANSACTION;
USE `exafarm`;
INSERT INTO `exafarm`.`bloques` (`id`, `descripcion`, `is_activo`)
VALUES (1, 'Teoría', 1);
INSERT INTO `exafarm`.`bloques` (`id`, `descripcion`, `is_activo`)
VALUES (2, 'Seminarios', 1);

COMMIT;

-----
-- Data for table `exafarm`.`temas`
-----

START TRANSACTION;
USE `exafarm`;
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (1, 'Definición
y objetivos de la Farmacología Clínica', 'Definición y objetivos', 1,
1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (2, 'Información
sobre medicamentos. Automedicación.', 'Información y automedicación',
1, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (3,
'Cumplimiento de la prescripción.', 'Cumplimiento', 1, 1);
```

```
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (4,
'Especialidades Farmacéuticas. Tipos.', 'Especialidades
farmacéuticas', 1, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (5,
'Farmacocinética Clínica. Modelos farmacocinéticos elementales.',
'Farmacocinética', 1, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (6,
'Variabilidad de la respuesta a los fármacos. Individualización del
tratamiento. Monitorización. Ajuste de dosis.', 'Variabilidad', 1, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (7,
'Interacciones medicamentosas y su repercusión en la terapéutica.',
'Interacciones', 1, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (8, 'Bases de la
utilización de medicamentos en situaciones especiales: niños,
ancianos, embarazadas, enfermos renales y enfermos hepáticos.',
'Medicamentos en situaciones especiales', 1, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (9, 'Desarrollo
de nuevos fármacos: Fases I, II y III. ', 'Desarrollo de fármacos', 1,
1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (10, 'Evaluación
del ensayo clínico. Procedimientos de registro.', 'Ensayos clínicos',
1, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (11, 'El efecto
placebo.', 'Placebo', 1, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (12, 'Fase IV:
Estudios postautorización. Estudios de utilización de medicamentos.',
'Fase IV', 1, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (13,
'Farmacovigilancia. Bases farmacológicas de las reacciones adversas a
los medicamentos.', 'Farmacovigilancia', 1, 1);
```

```
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (14, 'Técnicas
de Farmacovigilancia', 'Técnicas de farmacovigilancia', 1, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (15,
'Introducción: Criterios para la selección de medicamentos. ',
'Selección de medicamentos', 2, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (16,
'Antiinfecciosos: antibacterianos, antifungicos, antivirales.',
'Antiinfecciosos', 2, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (17,
'Broncodilatadores, antiasmáticos, mucoliticos y expectorantes.',
'Broncodilatadores', 2, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (18,
'Antisecretores y fármacos que modifican la motilidad intestinal.',
'Antisecretores', 2, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (19,
'Antihipertensivos y diuréticos.', 'Antihipertensivos', 2, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (20,
'Hipolipemiantes.', 'Hipolipemiantes', 2, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (21,
'Antitrombóticos.', 'Antitrombóticos', 2, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (22,
'Hipnóticos. Ansiolíticos. ', 'Hipnóticos', 2, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (23,
'Antidepresivos.', 'Antidepresivos', 2, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (24,
'Analgésicos opiáceos y no opiáceos.', 'Analgésicos', 2, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (25,
'Antiinflamatorios esteroideos y no esteroideos.',
'Antiinflamatorios', 2, 1);
```

```

INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (26,
'Antidiabéticos.', 'Antidiabéticos', 2, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (27, 'Hormonas
esteroideas y fármacos relacionados.', 'Hormonas esteroideas', 2, 1);
INSERT INTO `exafarm`.`temas` (`id`, `descripcion`,
`descripcion_corta`, `id_bloque`, `is_activo`) VALUES (28,
'Notificación de reacciones adversas.', 'Notificación de RAM', 2, 1);

COMMIT;

-- -----
-- Data for table `exafarm`.`preguntas`
-- -----

START TRANSACTION;
USE `exafarm`;
INSERT INTO `exafarm`.`preguntas` (`id`, `is_activa`,
`fecha_creacion`, `id_tema`, `texto`, `is_corta`, `id_usuario`,
`is_mezclable`) VALUES (NULL, 1, '2011-02-05', 1, 'Señale la
afirmación falsa', 0, 1, 0);
INSERT INTO `exafarm`.`preguntas` (`id`, `is_activa`,
`fecha_creacion`, `id_tema`, `texto`, `is_corta`, `id_usuario`,
`is_mezclable`) VALUES (NULL, 1, '2011-02-05', 2, 'Cual de estas
afirmaciones es cierta:', 0, 2, 0);
INSERT INTO `exafarm`.`preguntas` (`id`, `is_activa`,
`fecha_creacion`, `id_tema`, `texto`, `is_corta`, `id_usuario`,
`is_mezclable`) VALUES (NULL, 1, '2011-02-05', 1, 'En situación de RCP
avanzada, señale la correcta:', 0, 1, 0);
INSERT INTO `exafarm`.`preguntas` (`id`, `is_activa`,
`fecha_creacion`, `id_tema`, `texto`, `is_corta`, `id_usuario`,
`is_mezclable`) VALUES (NULL, 1, '2011-02-05', 3, 'En relación a las
maniobras de RCP es cierto que:', 0, 1, 0);
INSERT INTO `exafarm`.`preguntas` (`id`, `is_activa`,
`fecha_creacion`, `id_tema`, `texto`, `is_corta`, `id_usuario`,
`is_mezclable`) VALUES (NULL, 1, '2011-02-05', 1, 'Con respecto a los
objetivos de la desfibrilación señale la afirmación correcta:', 0, 1,
0);
INSERT INTO `exafarm`.`preguntas` (`id`, `is_activa`,
`fecha_creacion`, `id_tema`, `texto`, `is_corta`, `id_usuario`,
`is_mezclable`) VALUES (NULL, 1, '2011-02-05', 1, '¿Cuál de las

```

```

siguientes actuaciones no tiene indicación en taquicardia regular de
QRS estrecho?:', 0, 1, 0);
INSERT INTO `exafarm`.`preguntas` (`id`, `is_activa`,
`fecha_creacion`, `id_tema`, `texto`, `is_corta`, `id_usuario`,
`is_mezclable`) VALUES (NULL, 1, '2011-02-05', 1, 'En la bradiarritmia
sintomática, cuál de estos fármacos no está indicado:', 0, 1, 0);
INSERT INTO `exafarm`.`preguntas` (`id`, `is_activa`,
`fecha_creacion`, `id_tema`, `texto`, `is_corta`, `id_usuario`,
`is_mezclable`) VALUES (NULL, 1, '2011-02-05', 1, 'La distensión de
las venas del cuello es un signo que orienta hacia... Señale la
incorrecta:', 0, 1, 0);
INSERT INTO `exafarm`.`preguntas` (`id`, `is_activa`,
`fecha_creacion`, `id_tema`, `texto`, `is_corta`, `id_usuario`,
`is_mezclable`) VALUES (NULL, 1, '2011-02-05', 1, 'Indique la opción
correcta. La razón por la que no se administra dosis de ataque de
fanobarbital es porque:', 0, 1, 0);

COMMIT;

-----
-- Data for table `exafarm`.`respuestas`
-----

START TRANSACTION;
USE `exafarm`;
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 1, 'En el inicio de
maniobras de RCP se considera implícito el consentimiento por parte
del paciente.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 1, 'No es necesaria una
orden médica para iniciar la RCP.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 1, 'Es necesaria una orden
médica para interrumpir las maniobras de RCP.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 1, 'Aunque las secuelas
pueden ser graves tras una RCP, debe realizarse en todas las PCR.', 1,
1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 1, 'El papel del público no

```

```
sanitario, en el medio extrahospitalario, es fundamental en el inicio
y realización de la REP.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 2, 'La RCP básica carece de
utilidad.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 2, 'En España los testigos
de un paro cardiaco habitualmente no realizan la RCP básica', 1, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 2, 'La desfibrilación
semiautomática solo debe realizarla el médico.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 2, 'La RCP avanzada solo
debe efectuarse en el hospital.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 2, 'El objetivo de
posibilitar la desfibrilación en los primeros 5 min. de un paro
cardiaco es una recomendación con bajo nivel de evidencia.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 3, 'La monitorización
eléctrica inicial se debe realizar a través de electrodos.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 3, 'Una vez que se ha hecho
una descarga a través de palas, no podemos monitorizar de nuevo con
ellas.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 3, 'No podemos conocer el
ritmo cardiaco si monitorizamos con palas.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 3, 'No podemos conocer el
verdadero ritmo cardiaco si no realizamos un ECG de 12 derivaciones.',
0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 3, 'La monitorización a
través de electrodos no debe interrumpir las maniobras de RCP', 1, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 4, 'Si disponemos de
desfibrilador, la primera medida es iniciar la RCP básica.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 4, 'El golpe precordial
resuelve el 100% de FV.', 0, 1);
```



```
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 4, 'Ante un pulso central
débil debe sospecharse FV.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 4, 'Durante la RCP hay que
corregir las causas potencialmente reversibles.', 1, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 4, 'Todas las anteriores son
falsas.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 5, 'Persigue interrumpir la
actividad eléctrica anárquica', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 5, 'Intenta restaurar la
actividad eléctrica coordinada.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 5, 'El último termino, el
objetivo es restaurar la circulación espontánea.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 5, 'Solo a y b son ciertas',
0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 5, 'a, b y c son ciertas.',
1, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 6, 'Esmolol iv', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 6, 'Verapamilo iv', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 6, 'Adenosina iv en
perfusión', 1, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 6, 'Maniobras vagales', 0,
1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 6, 'Amiodarona iv', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 7, 'Digoxina', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 7, 'Adrenalina', 0, 1);
```

```
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 7, 'Isoproterenol', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 7, 'Atropina', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 7, 'Todas las anteriores',
1, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 8, 'Neumotórax a tensión',
0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 8, 'Shock hipovolémico', 1,
1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 8, 'Taponamiento cardíaco',
0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 8, 'Disfunción cardiaca por
contusión miocárdica.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 8, 'a y c son ciertas', 0,
1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 9, 'Su vida media es de 5
días', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 9, 'Habitualmente se
administra cada 24 horas.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 9, 'Si alcanzásemos el rango
terapéutico en el primer día de tratamiento, se produciría una
sedación excesiva.', 0, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 9, 'Se administra por vía
oral', 1, 1);
INSERT INTO `exafarm`.`respuestas` (`id`, `id_pregunta`, `texto`,
`is_correcta`, `is_activa`) VALUES (NULL, 9, 'Pasa la barrera
placentaria y también a la leche materna', 0, 1);

COMMIT;
```

```
-----  
-- Data for table `exafarm`.`examen_pregunta`  
-----  
START TRANSACTION;  
USE `exafarm`;  
INSERT INTO `exafarm`.`examen_pregunta` (`id_examen`, `id_pregunta`)  
VALUES (1, 1);  
  
COMMIT;
```

### Servidor de aplicaciones

El servidor de aplicaciones utilizado es Glassfish versión 3.1 edición Open Source, dada su compatibilidad completa para aplicaciones que cumplen el estándar JEE 6 y su integración con el IDE Netbeans.

El programa de instalación del programa se puede descargar de la url mostrada a continuación, se deberá descargar el fichero que mejor se adapte al sistema en el que se quiere instalar el programa:

<http://glassfish.java.net/downloads/3.1-final.html>

(dirección válida el 16/6/2011)

Una vez instalado glassfish es necesario instalar un conector para bases de datos MySQL. Se trata de un controlador que convierte llamadas JDBC en el protocolo de conexión del SGBD de MySQL. Este se puede descargar de:

<http://www.mysql.com/downloads/connector/j/>

(dirección válida el 16/6/2011)

Se seleccionará el fichero que mejor se adapte al sistema. Se ha de extraer del archivo el fichero *mysql-connector-java-5.1.5-bin.jar* y se guardará en el directorio *lib* de glassfish.

Antes de ejecutar el programa es necesario configurar el servidor de aplicaciones. Serán necesarias activar directivas de seguridad y de acceso a la base de datos.

Se deben ejecutar los siguientes comandos desde la terminal de comandos en el directorio de glassfish:

```
asadmin create-jdbc-connection-pool --datasourceclassname
com.mysql.jdbc.jdbc2.optional.MysqlDataSource --restype
javax.sql.DataSource --property
PortNumber=3306:User=exafarm_user:Password=UOC:ServerName=localhost:Da
tabaseName=exafarm ExaFarm
```

```
asadmin create-jdbc-resource --connectionpoolid ExaFarm jdbc/exafarm
```

En el caso de utilizar otros usuarios en la base de datos, diferentes a los creados con las sentencias anteriores, será necesario introducir su nombre de usuario y password en los puntos marcados en negrita.

Para comprobar que todo ha ido bien:

```
asadmin ping-connection-pool ExaFarm
```

Crear el dominio de seguridad:

```
asadmin create-auth-realm --classname
com.sun.enterprise.security.auth.realm.jdbc.JDBCRealm --property user-
name-column=user_id:password-column=password:group-name-
column=grupo:jaas-context=jdbcRealm:datasource-
jndi="jdbc/exafarm":group-table=usuarios:user-
table=usuarios:encoding=Base64 ExaFarmSecurity
```

## Paquete del programa

El paquete war del programa se adjunta en el fichero zip entregado y tiene como nombre:

*ExaFarm-TFC.war*

Este paquete deberá ser copiado en la carpeta autodeploy del servidor de aplicaciones para poderse ejecutar. (glassfish/domains/domain1/autodeploy)

En el fichero zip se adjunta también el el código fuente del proyecto.