

Proyecto kda.com

Un sitio donde compartir aficiones

Consultor: Antoni Oller Arcas
Memoria del proyecto

Índice de contenido

Descripción del proyecto.....	4
Introducción.....	4
Funcionalidad del sitio:.....	4
Parte técnica:.....	4
Herramientas para la documentación:.....	5
Páginas implementadas.....	6
Tecnologías y herramientas del proyecto.....	7
Herramientas de programación:.....	7
Librerías y tecnologías empleadas:.....	8
Gestión y documentación del proyecto.....	9
Análisis funcional y casos de uso.....	10
Introducción.....	10
Análisis funcional.....	10
Casos de uso.....	12
Registro de usuarios.....	12
Modificación de los datos de usuario.....	13
Planificar actividades.....	14
Suscribirse a actividad.....	15
Comentar actividad.....	16
Descartar una actividad.....	17
Acceder al espacio de usuario.....	18
Listado de actividades de usuario.....	19
Listado de nuevas actividades para el usuario.....	20
Entrar en el sistema.....	21
Salir del sistema.....	22
Restricciones iniciales.....	23
Interfície de usuario.....	24
Registro de usuarios.....	24
Modificación datos de usuarios.....	25
Planificar actividad.....	25
Suscribirse a actividad.....	26
Comentar actividad.....	26
Descartar actividad.....	27
Acceder al espacio de usuario.....	28
Consulta de los datos de usuario.....	28
Listado actividades usuario.....	29
Listado avisos nuevas actividades.....	29
Entrar en el sistema.....	29
Diagrama de clases del modelo.....	31
Atributos de las clases del modelo.....	32
Usuario.....	32
AgrupacionAficiones.....	33
Aficion.....	33

Provincia.....	34
Ubicacion.....	34
.....	34
Actividad.....	35
ActividadSuscrita.....	35
ActividadesDescartadas.....	36
ComentarioActividad.....	36
Notas importantes con respecto al modelo.....	37
Patrones de diseño e implementación de Kda.....	38
Patrones de diseño adoptados en la aplicación.....	38
Ejemplo de funcionamiento mediante Diagrama de secuencia.....	41
Diseño de base de datos.....	42
Diseño lógico de la base de datos.....	43
Diseño de las tablas de la base de datos.....	46
Mapa del sitio web.....	49
Portada (inicio).....	49
Pizarra.....	50
Datos de Usuario.....	51
Datos de la Actividad.....	52
Funcionamiento de la suscripción a una actividad.....	53
Funcionamiento del descarte de una actividad.....	54
Comentar Actividad.....	55
Instalación de la aplicación.....	56
Creación de la base de datos kda.....	56
Relleno de la información inicial.....	59
Historial de la instalación del entorno de trabajo.....	61
Creación del archivo war y ejecución de la aplicación.....	62
Ver la aplicación en funcionamiento.....	62
Planificación temporal del TFC.....	64
Planificación seguida según las fechas de entrega para las Pecs.....	64
Diagrama de Gantt.....	66
Conclusiones y futuras modificaciones.....	67
Futuro de Kda.....	68
Bibliografía y recursos.....	69

Descripción del proyecto

Introducción

Funcionalidad del sitio:

El sitio web implementado pretende crear un lugar donde la gente pueda indicar cuales son sus aficiones, sus lugares de posible reunión y organizar grupos para poder compartir dichas aficiones.

Los usuarios se registran, introducen sus datos de localización y contacto, e indicarán cuales son sus aficiones. La cantidad de aficiones que un usuario puede indicar está limitada en número. El sistema asocia en grupos a los usuarios por población y afición.

Cualquiera de los usuarios del sitio puede proponer una actividad para un lugar y una hora concreta. El sistema avisará a los usuarios del grupo indicando que hay una nueva actividad. Estos podrán suscribirse confirmando su asistencia y realizando los comentarios que estimen oportunos. También pueden descartar dicha actividad, en caso de no querer suscribirse, para que desaparezca de su espacio de usuario.

En todo momento, los usuarios incluidos en un grupo podrán consultar la lista de usuarios adheridos a una actividad y los detalles de la misma.

Parte técnica:

Se pretendía realizar todo el proyecto haciendo uso unicamente de software gratuito, y a ser posible, de código libre y libre distribución. Todas las herramientas y librerías utilizadas debían pertenecer al ecosistema J2EE, estar ampliamente instaladas y ser estables. Y además, la aplicación no tenía que requerir usar un servidor de aplicaciones, con un contenedor de servlets y librerías adicionales debía ser suficiente.

Para la parte visual, se ha querido utilizar la especificación de referencia JSF 2.0 que ya aporta todas las herramientas necesarias para la visualización, validación, composición y control de las pantallas.

La grabación de la información debía ser agnóstica en cuanto a la tecnología utilizada para la persistencia. Con JPA cubrimos ese requerimiento ya que, con una mínima configuración, "escondemos" toda la complejidad asociada a la transformación de las entidades en tablas de base de datos.

Otra de las intenciones del proyecto, era independizar la parte de visualización de la parte de la lógica de negocio adoptando el patrón Modelo-Vista-Controlador (MVC). Con JSF, ya se conseguía, ya que está basado en dicho patrón. Pero además, se pretendían implementar otra serie de patrones que hicieran más independientes cada una de las partes con el fin de obtener una aplicación muy modularizada. De esta forma podremos modificar cualquiera de las tecnologías utilizadas en la aplicación en un futuro sin que ninguna de las demás se vea afectada.

La construcción y generación del .war de la aplicación debía ser lo más automatizada posible. Es decir, la gestión de dependencias de librerías y el empaquetamiento de la aplicación debían poderse realizar un número infinito de veces sin que eso supusiera descargas o copias de archivos reiterados. Maven surgió como solución a este planteamiento.

Por último, en cuanto a la parte técnica, la base de datos debía cumplir también los requisitos mínimos de gratuidad y libre distribución. Se barajaron una serie de base de datos, pero al final se decidió optar por MySql Community Edition.

Herramientas para la documentación:

En cuanto a las herramientas para la elaboración de documentos y diagramas, también debían ser gratuitas y de eficiencia contrastadas.

En este caso estaba claro, OpenOffice y StarUML llevan años ofreciendo todo lo necesario para realizar tanto la presentación como los diagramas y documentos a incorporar en el proyecto.

Páginas implementadas

Los apartados en los que se divide el sitio web son los siguientes:

- Portada (inicio): página inicial donde se explica la funcionalidad del sitio, se exponen las características del mismo y se permite el acceso y registro de los usuarios. Se trata de la parte pública de la aplicación y no ha sido desarrollada en su presentación, pero sí en su funcionalidad.
- Registro (datos_usuario): página encargada de la solicitud de los datos de los usuarios, sus aficiones y alta de los mismos en la base de datos. Cuando se selecciona un tipo de afición se cargan automáticamente (usando AJAX) las aficiones asociadas. La misma funcionalidad se realiza con las Provincias y las Ubicaciones, que también se cargan dinámicamente. A esta página se puede acceder también desde el espacio de usuario para editar los datos asociados al usuario activo.
- Pagina personal (pizarra): página en la que los usuarios pueden gestionar sus datos, dar de alta actividades, ver aquellas a las que están suscritos, descartar actividades que no interesen y realizar comentarios en las actividades en las que están o han estado suscritos.
- Alta de una Actividad (datos_actividad): cualquier usuario registrado puede dar de alta una actividad en el sistema. Una vez añadida, todos los usuarios de la localidad con la misma afición de la actividad son informados en su página personal de que pueden suscribirse o descartarse. Se indica la fecha de celebración, lugar exacto donde se realiza y una descripción de la misma.
- Alta de un Comentario (comentario_actividad): un usuario suscrito a una actividad, puede realizar comentarios sobre la misma. Dichos comentarios aparecen automáticamente en el espacio de usuario de los demás usuarios suscritos a la actividad.

Tecnologías y herramientas del proyecto

Para el desarrollo del proyecto se han utilizado tecnologías de la plataforma J2EE que permiten independizar la vista del modelo y gestionar, mediante controladores, el acceso a las diferentes capas implementando así una arquitectura MVC (modelo, vista, controlador).

Herramientas de programación:

Todas las herramientas de desarrollo utilizadas son gratuitas y pueden ser descargadas desde las páginas web de los proyectos:

- La aplicación se ha desplegado tanto en un servidor Tomcat 6.0 como Tomcat 7.0. Tomcat 7.0 dentro del entorno de desarrollo Eclipse y Tomcat 6.0 fuera del mismo. Se ha elegido esta versión como versión final del servidor de aplicaciones ya que se considera estable y con suficiente historia para presentar pocas incidencias. Página del proyecto:

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

- La versión de java utilizada ha sido la "Java EE 6 SDK" que actualmente es la versión empresarial estándar. Página del proyecto:

<http://www.oracle.com/technetwork/java/javaee/tech/index.html>

- Como entorno de desarrollo se ha utilizado "Eclipse Java EE IDE for Web Developers" versión Juno con el plugin de maven para Eclipse para gestionar las dependencias del proyecto. Página del proyecto:

<http://www.eclipse.org/juno/>

- La base de datos donde se han creado las tablas e índices necesarios para el almacenaje de la información ha sido "MySQL Community Server 5.6.10". Se trata de la versión gratuita de esta base de datos ampliamente instalada y testada. Página del proyecto:

<http://www.mysql.com/products/community/>

- Como herramienta de gestión de las librerías de las que depende el proyecto y de construcción del producto final, se ha utilizado el plugin de maven para Eclipse. Este plugin permite especificar

cuales son las librerías que deben descargarse y que versión de las mismas se utiliza para la construcción del archivo war final de la aplicación. En eclipse se especificará como url de instalación:

<http://download.eclipse.org/technology/m2e/releases/1.3/1.3.1.20130219-1424>

Librerías y tecnologías empleadas:

Con el fin de simplificar el desarrollo y dotar de una funcionalidad extra a la aplicación, se han incorporado al proyecto una serie de librerías que extienden las características básicas de las tecnologías utilizadas. Estas librerías se incorporan al proyecto mediante la especificación de dependencias dentro del archivo pom.xml de maven:

- Para la persistencia de los datos se ha utilizado la librería "OpenJPA 2.2" que implementa la especificación "JSR-317 Java Persistence 2.0". No se han utilizado todas las características de JPA con el fin de permitir la portabilidad de la aplicación a otro tipo de almacenes de datos que no sean bases de datos relacionales.
- Para la parte visual y de controlador se ha usado la librería "PrimeFaces 3.5" que implementa la especificación JSF 2. La librería tiene una gran cantidad de componentes y soporta Ajax nativo, además de poder configurar una gran cantidad de temas. Además de la propia librería de PrimeFaces, se han incorporado al proyecto las librerías de JSF 2. Se han utilizado en el proyecto las capacidades de JSF para soportar multiidioma en la aplicación. No se han habilitado más idiomas que el español, pero ya se han guardado todos los literales en un archivo de idioma para poder dotar de esta funcionalidad al sitio.
- Para la conexión con la base de datos se han incorporado las librerías de mysql-connector para Java.
- Para la inyección de las dependencias entre los diferente objetos, además de la inyección básica proporcionada por JSF 2, se ha utilizado Spring 3.1.2.

También se han incluido en el proyecto, aunque actualmente no se están utilizando, las librerías de PrettyFaces que ocultan las extensiones de los archivos utilizados para la visualización de las páginas web. Debido a un

problema detectado en la navegación, se prescindió de su uso.

Gestión y documentación del proyecto

En cuanto a las herramientas para la gestión, elaboración de los diagramas y la documentación se han utilizado:

- Para la elaboración de los diagramas de clases, de actividad, de secuencia, etc. del proyecto se ha utilizado "Astar community 6.6", antiguo JUDE Community.
- Para la gestión del proyecto y la elaboración de los diagramas de Gantt se ha empleado "Open Workbench 1.1.4".
- Y por último, para la creación de los documentos de entrega para las diferentes Pecs se ha utilizado "Apache Open Office 3".

Análisis funcional y casos de uso

Introducción

El sitio web implementado tiene como objetivo la gestión de actividades relacionadas con las aficiones de los usuarios registrados. A partir de un perfil de usuario donde se han indicado las localidades que quedan cercanas al lugar donde se reside y una serie de aficiones, los usuarios pueden gestionar su suscripción a actividades, ya sean creadas por ellos mismos o por otros usuarios.

La aplicación consta de una serie de pantallas para realizar las funciones de registro y modificación de datos de usuario, alta de actividad, comentar actividad, suscribirse o descartar actividad. A todas esas pantallas se accede desde el espacio de usuario llamado pizarra. Dicho espacio de usuario presenta la información mínima necesaria para consultar y gestionar las actividades de un usuario, así como sus datos personales. Por esta razón, el flujo de navegación siempre es el mismo y parte desde la pizarra hacia todas las restantes consultas posibles.

Análisis funcional

Las diferentes funcionalidades implementadas en el proyecto son:

- Registro de los usuarios: un usuario anónimo que acceda a la página principal del sitio puede registrarse para comenzar a ser un usuario de la aplicación. Cuando se realiza el registro, se da de alta el usuario en el sistema habiendo introducido sus datos, sus aficiones y accediendo a la página personal.
- Modificación de los datos del usuario: un usuario registrado puede modificar aquellos datos que deban ser actualizados o no sean correctos.
- Planificar actividades: en función de las aficiones asociadas, un usuario puede planificar actividades en el sistema de cara a que otros usuarios puedan suscribirse.
- Suscribirse a actividades: en el momento que un usuario detecte o

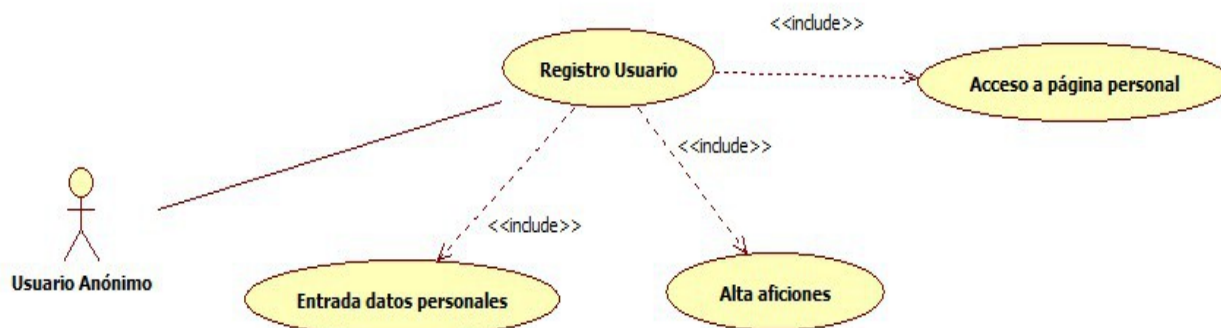
sea informado de una actividad que le interesa, podrá suscribirse para participar en la misma.

- Comentar actividad: antes y después de haberse realizado una actividad, ésta puede ser comentada. Se pueden añadir comentarios para su preparación y también una vez finalizada.
- Descartar una actividad: cuando el usuario es informado de una nueva actividad que puede ser de su interés, éste puede descartar dicha actividad, ya que no piensa suscribirse a la misma.
- Acceder al espacio de usuario: una vez logado en el sistema, el usuario accede a su espacio de usuario llamado pizarra. Se trata de un panel de control donde se visualiza la información más relevante asociada al usuario. Se tratan de consultas limitadas en tamaño, con la información más reciente, para poder ofrecer una gestión más dinámica. Dicho espacio consta de las siguientes consultas:
 1. Consulta de los datos del usuario: se muestran los datos más relevantes del usuario y se permite acceder al área de modificación.
 2. Listado de las actividades: Se presentan las actividades pendientes de realizar por parte del usuario indicando sus datos de planificación.
 3. Listado de comentarios: se listan los comentarios de las actividades a las que el usuario está suscrito, independientemente de que ya hayan finalizado, ya que pueden seguir haciéndose comentarios sobre la actividad. Los comentarios que se muestran son los posteriores a la suscripción del usuario a la actividad.
 4. Avisos de nuevas actividades: espacio destinado a mostrar los avisos de nuevas actividades creadas por otros usuarios para la localidad y para alguna de las aficiones del usuario. Estos avisos se utilizarán para realizar la suscripción o descarte.
- Entrar en el sistema: los usuarios ya registrados acceden a su área introduciendo su correo y contraseña.
- Salir del sistema: los usuarios pueden desconectarse del sistema cerrando la sesión.

Casos de uso

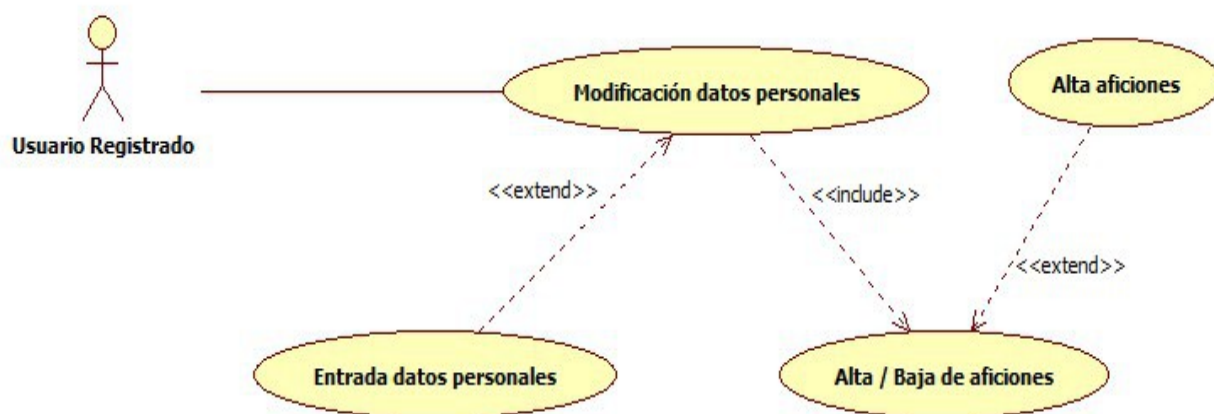
Registro de usuarios

Registro de usuarios	
Funcionalidad	Registrar un usuario en el sistema
Descripción	Un usuario anónimo de la web se registra en el sistema indicando una serie de datos personales para poder acceder a su espacio de trabajo.
Actores	Usuario anónimo
Precondición	El usuario no está registrado
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona el link de “Registrarse”. 2. El usuario indica sus datos personales. 3. Se valida que el correo no exista en el sistema. 4. Se valida que todos los datos obligatorios han sido rellenados y que son correctos. 5. Se da de alta el usuario y sus datos en las tablas de la aplicación. 6. Se muestra un mensaje de bienvenida al sitio. 7. El usuario accede a su espacio de trabajo (Pizarra).
Flujo Alternativo	<ol style="list-style-type: none"> 3. El correo ya existe en el sistema. 4. Se muestra un mensaje indicando que ya existe el usuario. 4. No se han tecleado todos los datos obligatorios o hay alguno incorrecto. 5. Se muestra un mensaje indicando donde se encuentra el error.
Postcondición	El usuario se encuentra registrado en el sistema y en su espacio de trabajo.



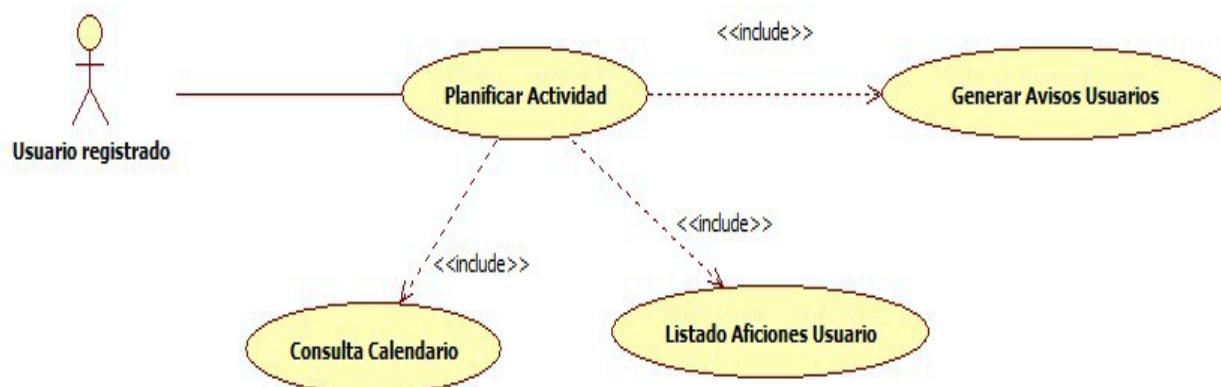
Modificación de los datos de usuario

Modificación de los datos de usuario	
Funcionalidad	Modificar los datos personales o de las aficiones de un usuario en el sistema
Descripción	Un usuario registrado en la web modifica sus datos personales, o bien, sus aficiones en el sistema.
Actores	Usuario registrado
Precondición	El usuario está registrado y se encuentra en su área de la aplicación
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona el link de “Editar” en sus datos de usuario. 2. El usuario modifica sus datos personales. 3. Se valida que todos los datos obligatorios han sido rellenados y que son correctos. 5. Se actualizan los datos en las tablas de la aplicación. 6. Se muestra un mensaje de modificación realizada con éxito. 7. El usuario accede a su espacio de trabajo (Pizarra).
Flujo Alternativo	<ol style="list-style-type: none"> 3. No se han teclado todos los datos obligatorios o hay alguno incorrecto. 4. Se muestra un mensaje indicando donde se encuentra el error.
Postcondición	El usuario ha actualizado sus datos en el sistema y regresa a su espacio de trabajo.



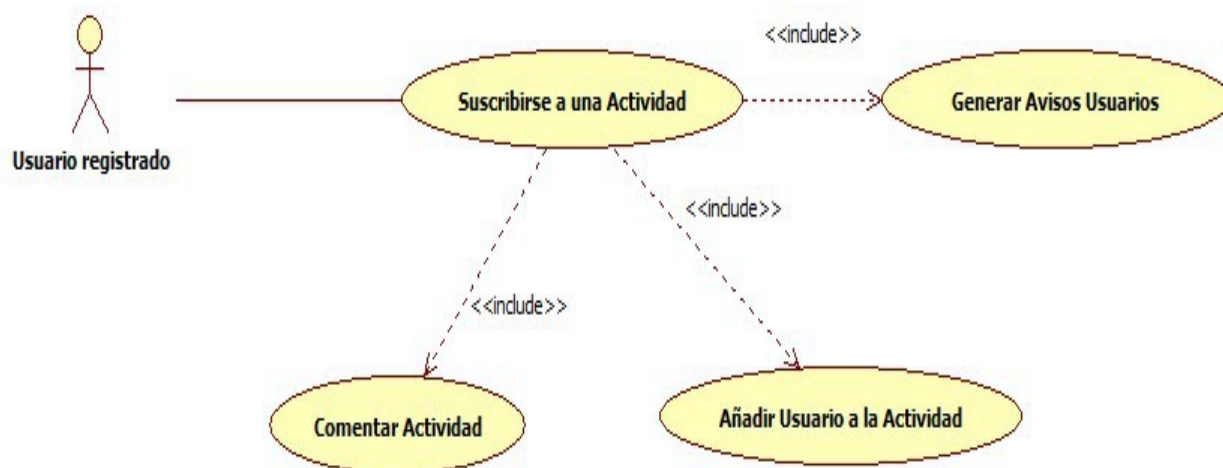
Planificar actividades

Planificar actividades	
Funcionalidad	Planificar una actividad dentro del sistema.
Descripción	Un usuario registrado en la web planifica una actividad para una localidad y una afición en concreto.
Actores	Usuario registrado
Precondición	El usuario está registrado y se encuentra en su área de la aplicación.
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona el link de “Nueva Actividad”. 2. El usuario selecciona una de sus aficiones, una fecha, una hora, un lugar exacto para la actividad y una descripción ampliada. 3. Se valida que todos los datos hayan sido informados y que son correctos. 5. Se crea la actividad y se generan los avisos para los usuarios de la localidad y con la misma afición de la actividad creada. 6. El usuario accede a su espacio de trabajo (Pizarra). 7. Se muestra la nueva actividad creada.
Flujo Alternativo	<ol style="list-style-type: none"> 3. Se ha tecleado algún dato incorrecto para la actividad. 4. Se muestra un mensaje indicando donde se encuentra el error. 3. El usuario cancela la planificación de la actividad.
Postcondición	El usuario ha creado una nueva actividad en el sistema y regresa a su espacio de trabajo.



Suscribirse a actividad

Suscribirse a actividad	
Funcionalidad	Suscribirse a una actividad existente en el sistema.
Descripción	Un usuario registrado en la web se suscribe a una actividad con el fin de participar en la misma.
Actores	Usuario registrado
Precondición	El usuario está registrado y, debido a que ha recibido una notificación de una actividad a realizar en alguna de sus localidades para una de sus aficiones, puede suscribirse a la actividad.
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona el link de “Suscribirse” en una notificación recibida en su espacio de usuario. 2. Se añade el usuario a la lista de usuarios suscritos a la actividad. 3. Se muestra la actividad a la que se ha suscrito en la lista de actividades pendientes al refrescarse su espacio de usuario.
Flujo Alternativo	No se ha descrito.
Postcondición	El usuario se ha suscrito a la actividad.



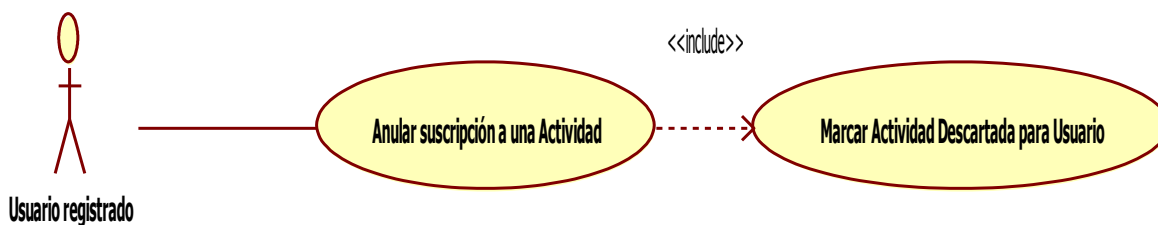
Comentar actividad

Comentar actividad	
Funcionalidad	Comentar una actividad a la que se está suscrito.
Descripción	Un usuario registrado y suscrito a una actividad realiza un comentario sobre la misma.
Actores	Usuario registrado
Precondición	El usuario está registrado y suscrito a la actividad que pretende comentar.
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona el link de “Comentar Actividad” en su lista de actividades a las que está suscrito. 2. El usuario puede realizar un comentario para la actividad. 3. Se añade el comentario en la lista de comentarios de la actividad. 4. Se generan los avisos para que los usuarios suscritos a la actividad sepan que se ha añadido un nuevo comentario. 5. El usuario accede a su espacio de trabajo (Pizarra).
Flujo Alternativo	2. El usuario cancela el comentario.
Postcondición	El usuario ha comentado la actividad.



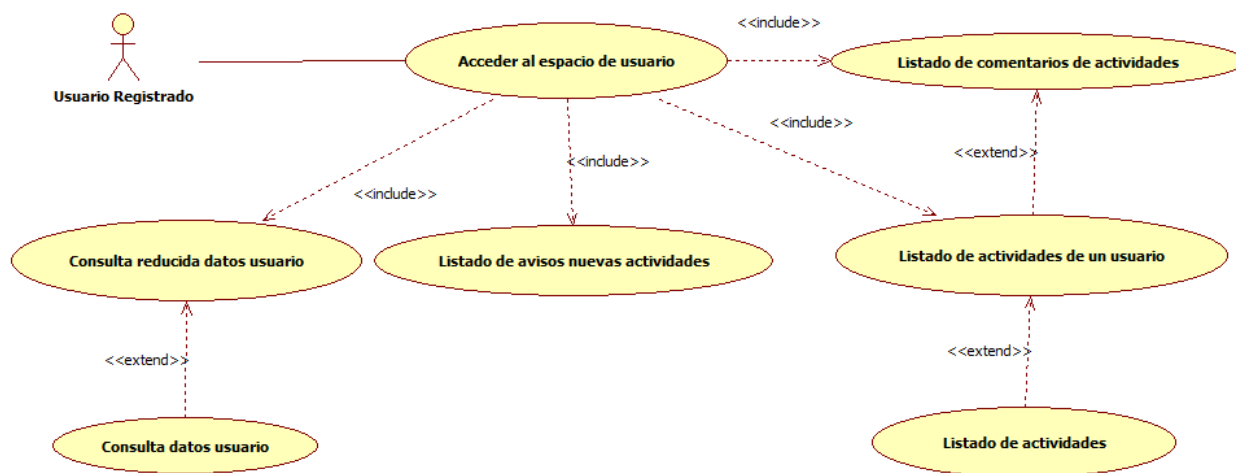
Descartar una actividad

Descartar una actividad	
Funcionalidad	Descartar la suscripción a una actividad.
Descripción	Un usuario registrado y que ha recibido un aviso de una nueva actividad, descarta dicha actividad para no recibir más avisos sobre la misma.
Actores	Usuario registrado
Precondición	El usuario está registrado y las características de la actividad coinciden con las del usuario.
Flujo	1. El usuario selecciona el link de “Descartar” en su lista de avisos de actividades. 2. Se refresca el espacio de trabajo (Pizarra).
Flujo Alternativo	No se ha descrito.
Postcondición	El usuario ha descartado la actividad.



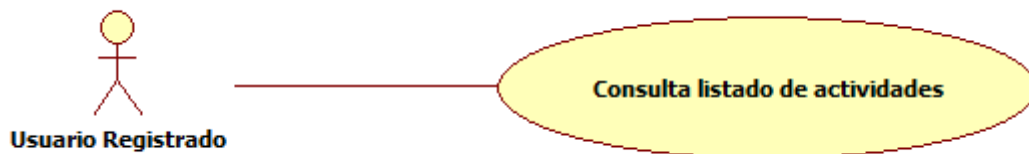
Acceder al espacio de usuario

Acceder al espacio de usuario	
Funcionalidad	Visualizar el espacio de trabajo del usuario.
Descripción	Un usuario registrado accede a su espacio de usuario donde se visualizan sus datos básicos, la lista de actividades y los últimos comentarios de las actividades.
Actores	Usuario registrado
Precondición	El usuario está logado en el sistema.
Flujo	1. El usuario accede a su espacio de trabajo. 2. Se visualizan las consultas de: <ul style="list-style-type: none"> • Datos básicos de usuario. • Listado de actividades de un usuario. • Listado de los comentarios de las actividades de un usuario. • Avisos de nuevas actividades interesantes para el usuario.
Flujo Alternativo	No se ha descrito.
Postcondición	El usuario se encuentra en su espacio de aplicación.



Listado de actividades de usuario

Listado de actividades de usuario	
Funcionalidad	Listar las actividades a las que está suscrito un usuario.
Descripción	Un usuario registrado visualiza el listado de las actividades a las que está suscrito.
Actores	Usuario registrado
Precondición	El usuario está logado en el sistema.
Flujo	<ol style="list-style-type: none"> 1. El usuario se encuentra en el espacio de usuario (Pizarra). 2. Se visualizan todas las actividades que están pendientes a las que el usuario está suscrito.
Flujo Alternativo	No se ha descrito.
Postcondición	El usuario visualiza el listado de actividades.



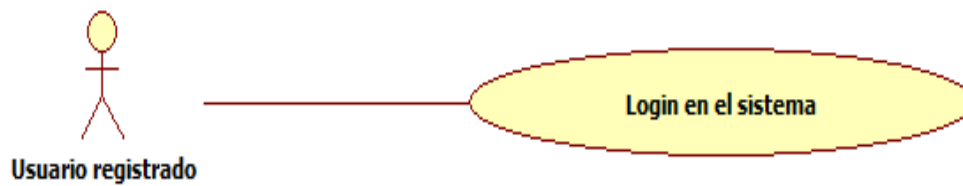
Listado de nuevas actividades para el usuario

Listado de nuevas actividades para el usuario	
Funcionalidad	Listar las nuevas actividades para las localidades y alguna de las aficiones del usuario.
Descripción	Un usuario registrado visualiza el listado de las nuevas actividades creadas por otros usuarios para sus localidades y para alguna de sus aficiones.
Actores	Usuario registrado.
Precondición	El usuario está logado en el sistema.
Flujo	1. El usuario se encuentra en el espacio de usuario (Pizarra). 2. Se visualizan todas las nuevas actividades en las localidades del usuario para alguna de sus aficiones.
Flujo Alternativo	No se ha descrito.
Postcondición	El usuario se encuentra en el listado de nuevas actividades.



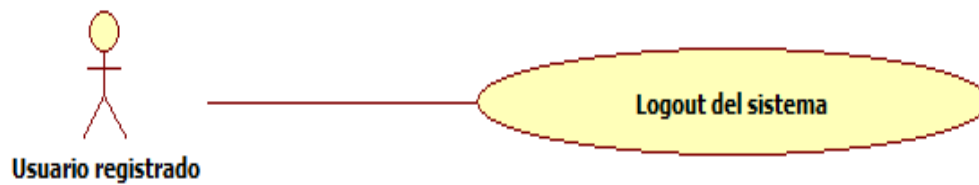
Entrar en el sistema

Entrar en el sistema	
Funcionalidad	Login del usuario en el sistema.
Descripción	Un usuario anónimo indica sus datos de usuario y contraseña de la aplicación y entra en el sistema accediendo a su espacio de usuario.
Actores	Usuario registrado
Precondición	El usuario está registrado en el sistema.
Flujo	1. El usuario introduce su mail y contraseña y acepta la entrada al sistema. 2. El usuario accede a su espacio de trabajo (Pizarra).
Flujo Alternativo	2. No coincide el mail con la contraseña y los campos de login advierten del error.
Postcondición	El usuario se encuentra en su espacio de aplicación.



Salir del sistema

Salir del sistema	
Funcionalidad	Logout del usuario en el sistema.
Descripción	Un usuario registrado y logado en la aplicación sale de la misma.
Actores	Usuario registrado
Precondición	El usuario está registrado y dentro del sistema.
Flujo	1. El usuario selecciona el link de “Logout” en su espacio de trabajo. 2. El usuario se redirige a la pantalla principal de inicio.
Flujo Alternativo	No se ha descrito.
Postcondición	El usuario se encuentra en la página principal de la aplicación.



Restricciones iniciales

Se establecen una serie de restricciones debidas al alcance inicial del proyecto:

- Cada usuario tendrá un máximo de 5 localidades para la creación o suscripción de actividades.
- Cada usuario tendrá un máximo de 5 aficiones para la creación o suscripción de actividades.
- Los grupos de las aficiones (deportes, juegos, etc.) estarán limitados y serán los cargados inicialmente en la aplicación.
- Las aficiones estarán limitadas, al igual que los grupos, y estarán limitados siendo los cargados inicialmente en la aplicación.
- Inicialmente se restringen las localidades al espacio geográfico de España. Se hará una carga inicial de las provincias y municipios a partir de un archivo de municipios.
- De momento no se limita el número de comentarios que un usuario puede realizar sobre una actividad ni de los comentarios que puede tener una actividad.

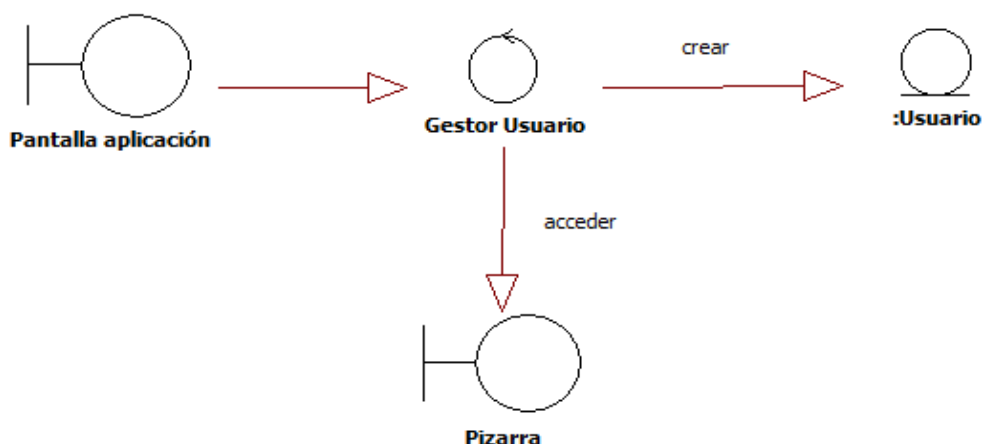
Interfície de usuario

La interfície de la aplicación, es exclusivamente web y, al usar componentes web JSF estándar, debe ser totalmente funcional y compatible con la mayoría de navegadores web. Se ha probado en Firefox y Chrome y los componentes funcionan correctamente.

Se hace uso de la funcionalidad AJAX que tienen algunos de los componentes existentes en PrimeFaces. En particular, los refrescos automáticos de determinadas áreas de pantalla o la carga de opciones de selección una vez seleccionado un determinado valor. Cuando se selecciona una provincia, se cargan automáticamente sus municipios, y al seleccionar una agrupación de aficiones, se cargan las aficiones asociadas.

Al utilizar el concepto de "Pizarra" (espacio de usuario) para toda la gestión de las actividades y las aficiones del usuario, la navegación siempre se realizará desde dicha pantalla. Todas las acciones comenzarán y finalizarán en la pantalla "Pizarra". Ésta, se refrescará para mostrar los valores actualizados una vez que se regrese desde cualquier otra acción.

Registro de usuarios



Desde la pantalla del sitio web se puede comenzar el proceso de registro

accediendo a la pantalla de introducción de los datos de usuario (personales y aficiones). Una vez finalizada la introducción de los datos se accederá al espacio de usuario.

Modificación datos de usuarios

Desde el espacio de usuario se podrá acceder a la pantalla de modificación de los datos de usuario así como de sus aficiones.



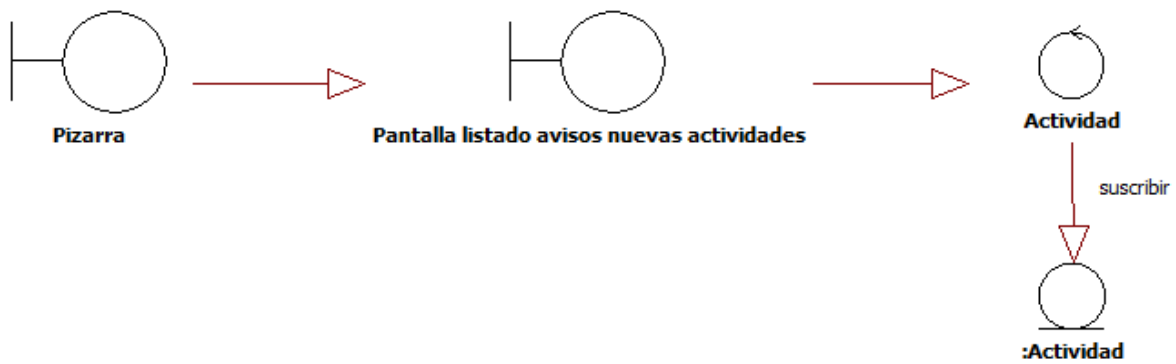
Planificar actividad

Desde el espacio de usuario se podrá acceder a la pantalla de creación de una actividad para una determinada afición del usuario.



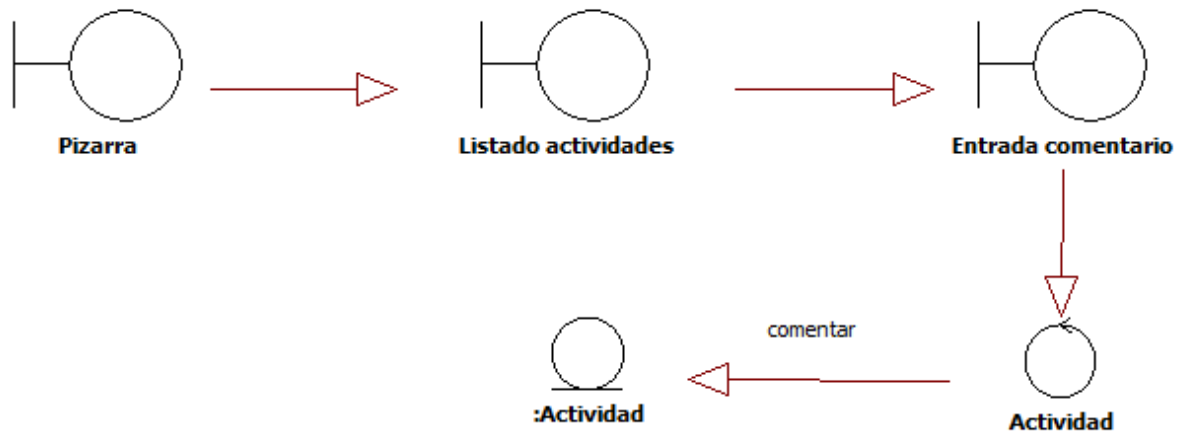
Suscribirse a actividad

Dentro del espacio de usuario, en el listado de nuevas actividades, se reciben los avisos de las nuevas actividades creadas por otros usuarios. En el propio listado, el usuario podrá suscribirse o descartarse de una actividad. Se mostrará toda la información necesaria de una actividad para que el usuario pueda tomar una decisión con respecto a la suscripción o descarte de la misma.



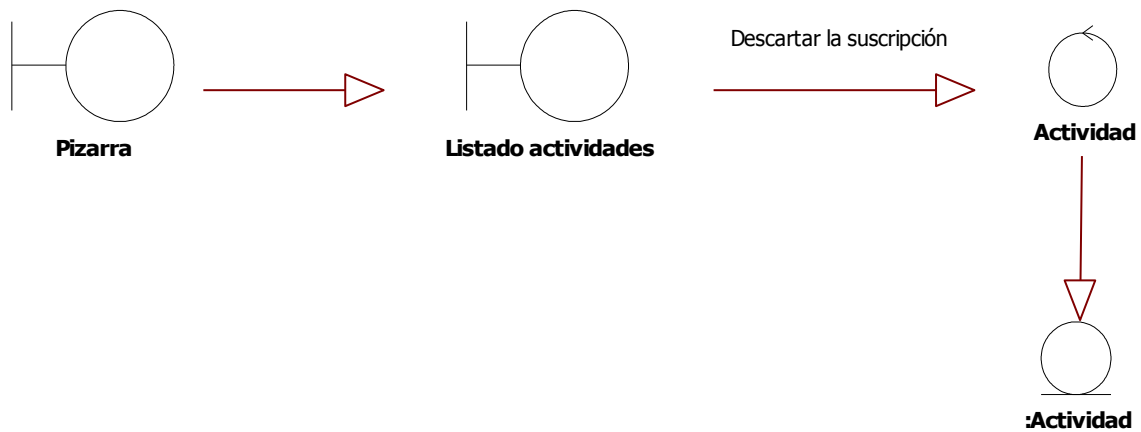
Comentar actividad

Desde el espacio de usuario, a través de la lista de actividades a las que está suscrito, o bien, a través de la lista de comentarios de las actividades a las que el usuario se ha suscrito, se pueden realizar comentarios que recibirán el resto de suscriptores a dicha actividad.



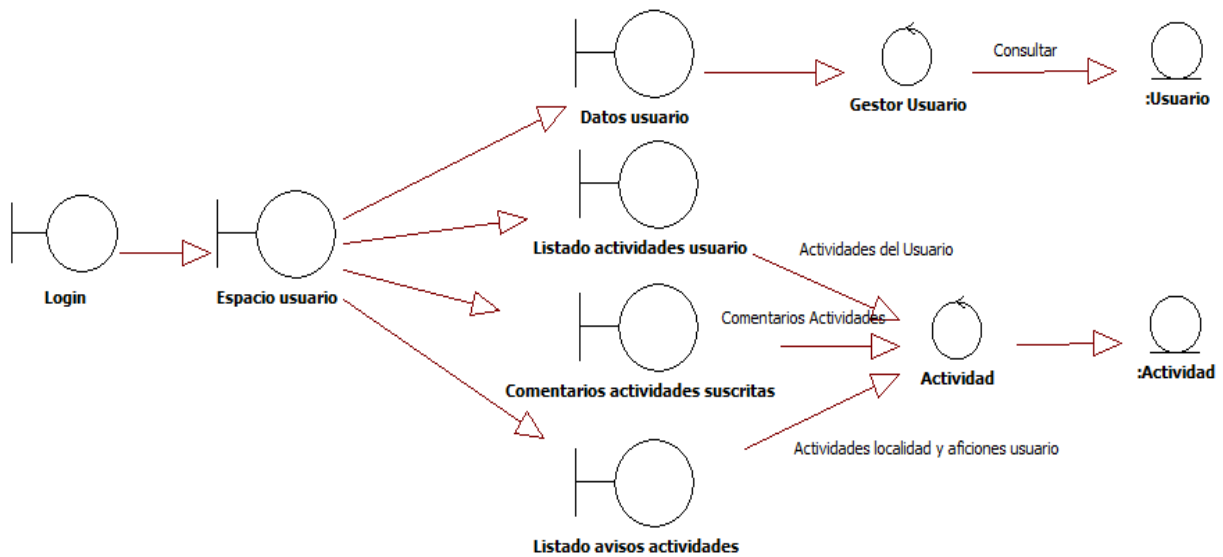
Descartar actividad

Desde el espacio de usuario, a través de la lista de avisos de actividades a las que se puede suscribir el usuario, se podrá descartar la suscripción a una actividad para no volver a visualizar el aviso en un futuro.



Acceder al espacio de usuario

Una vez logado el usuario en el sistema, se accede al espacio de usuario donde se visualizarán los datos básicos del usuario, las actividades a las que está suscrito, los últimos comentarios de dichas actividades y los avisos de las nuevas actividades que, por sus características, pueden



interesarle.

Consulta de los datos de usuario

Desde el espacio de usuario se podrán consultar sus datos personales así como las aficiones seleccionadas en la ficha.



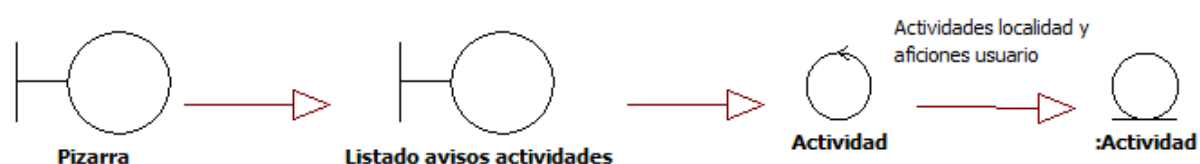
Listado actividades usuario

Desde el espacio de usuario se podrá consultar el listado de actividades a las que está suscrito el usuario ya sea porque las ha planificado él o bien porque se ha suscrito a las de otros usuarios.



Listado avisos nuevas actividades

Desde el espacio de usuario se puede consultar el listado de nuevas actividades, que pueden ser interesantes debido a las ubicaciones y las aficiones del usuario activo, ya que se planifican en las mismas localidades y para sus mismas aficiones.



Entrar en el sistema

Desde la portada del sitio web un usuario podrá logarse con su correo y su contraseña. Una vez autenticado en el sistema accederá directamente a su espacio de usuario.

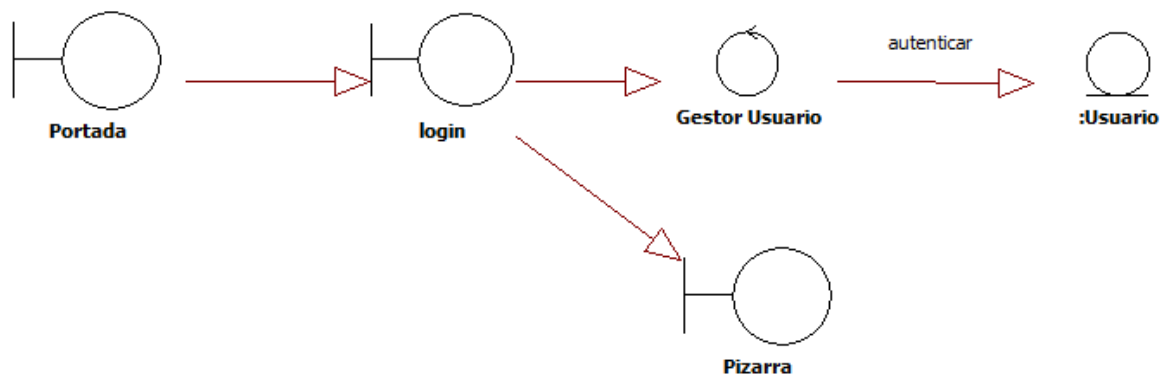
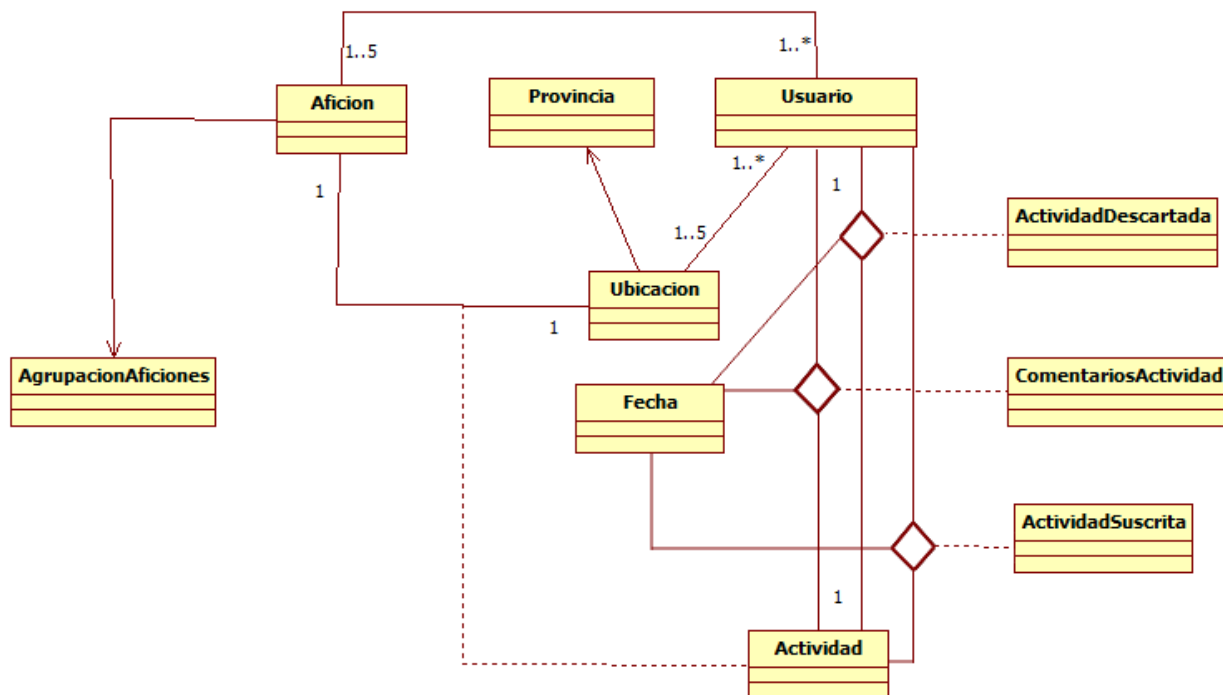


Diagrama de clases del modelo

El diagrama de clases resultado de la lógica de la aplicación será:



Donde se indica lo siguiente:

- Cada afición pertenece a una Agrupación de aficiones (deportes, juegos, etc.)
- Cada Usuario debe tener al menos una afición y un máximo de 5.
- Cada Usuario debe tener al menos una ubicación definida y un máximo de 5.
- Una Actividad será creada por un Usuario para una Afición y una Ubicación.
- Un Usuario puede suscribirse a una Actividad en una fecha.
- Un Usuario puede descartar una Actividad en una fecha.
- Un Comentario a una Actividad será creado por un Usuario para una Actividad y para una Fecha / Hora.

Atributos de las clases del modelo

En función de las clases detectadas, la relación entre las mismas y la funcionalidad de la aplicación los atributos serán los siguientes (todos serán privados y se accederá a ellos mediante setters y getters):

Usuario

Usuario
-mail: String -password: String -nombre: String -apellido1: String -apellido2: String -telefono: String -movil: String -extracto: String -aficiones: List<Aficion> -ubicaciones: List<Aficion> -foto: ByteArray

- mail: se utilizará como nombre de usuario para realizar el login en la aplicación.
- password: se guardará en la base de datos para poder validar el login en la aplicación.
- nombre: Nombre del usuario.
- apellido1: primer apellido. Será opcional.
- apellido2: segundo apellido. Será opcional.
- telefono: teléfono fijo del usuario. Será opcional.
- movil: teléfono móvil del usuario. Será opcional.
- extracto: descripción breve del usuario indicando sus aficiones y su disponibilidad para realizar actividades.

- aficiones: lista con las aficiones del usuario.
- ubicaciones: lista con las ubicaciones donde el usuario puede realizar actividades.
- foto: imagen del usuario.

AgrupacionAficiones

AgrupacionAficiones
-nombre: String

- nombre: nombre identificativo del grupo para las aficiones.

Aficion

Aficion
-grupoAficion: AgrupacionAficiones -nombreAficion: String -descripcion: String

- grupoAficion: al que pertenece la afición.
- nombreAficion: nombre identificativo de la afición.
- descripcion: breve descripción de la afición en caso de necesitarse. Será opcional.

Provincia

Provincia
-provincia: String -provinciaseo: String -provincia3: String

- provincia: nombre de provincia.
- provinciaseo: nombre de provincia normalizado.
- Provincia3: código de 3 caracteres para identificar la provincia.

Ubicacion

Ubicacion
-provincia: String -poblacion: String -codigoPostal: Integer

- provincia: donde se encuentra la población.
- poblacion: nombre de municipio.
- codigoPostal: código postal del municipio.

Actividad

Actividad
-usuarioCreador: Usuario -aficion: Aficion -ubicacion: Ubicacion -fechaCelebracion: Date -quedarEn: String -descripcion: String

- usuarioCreador: referencia al usuario que ha creado la actividad.
- aficion: referencia a la afición para la que se ha creado la actividad.
- ubicación: referencia a la ubicación donde se realizará la actividad.
- fechaCelebracion: fecha y hora en la que se celebrará la actividad.
- quedarEn: lugar exacto de la ubicación (calle o sitio) donde se realizará la actividad.
- descripcion: comentario sobre que se va a hacer en la actividad.

ActividadSuscrita

ActividadSuscrita
-usuario: Usuario -actividad: Actividad -fecha: Date

- usuario: referencia al usuario que se ha suscrito.

- actividad: referencia a la actividad para la que se ha suscrito.
- fecha: fecha y hora en la que se ha realizado la suscripción.

ActividadesDescartadas

ActividadDescartada
-usuario: Usuario -actividad: Actividad -fecha: Date

- usuario: referencia al usuario que se ha suscrito.
- actividad: referencia a la actividad para la que se ha suscrito.
- fecha: fecha y hora en la que se ha realizado la suscripción.

ComentarioActividad

ComentarioActividad
-usuario: Usuario -actividad: Actividad -fecha: Date -comentario: String -comentarioSuscripcion: Boolean

- usuario: referencia al usuario que ha creado el comentario.
- actividad: referencia a la actividad para la que se hace el comentario.

- fecha: fecha y hora en la que se ha realizado el comentario.
- comentario: comentario sobre la actividad.
- comentarioSuscripcion: se indica si se trata del comentario asociado a la suscripción. Actualmente no se utilizará, ya que para las suscripciones se utiliza la clase ActividadSuscrita.

Notas importantes con respecto al modelo

Debido a que se utiliza JPA para la persistencia de los objetos y de que se pretende hacer la aplicación portable a bases de datos noSQL, que pueden no implementar un modelo relacional, no se ha usado la funcionalidad que ofrece JPA para realizar cargas de objetos asociados a otros objetos (uno a uno, uno a muchos, etc.). Un ejemplo de las limitaciones que pueden encontrarse en almacenamientos noSQL en la nube, y de los pros y los contras de usar almacenamiento SQL y noSQL sobre Google App Engine se comenta en los artículos "SQL or NoSQL: Google App Engine" en <http://www.sitepoint.com/nosql-and-the-appengine-datastore/>. Por esta razón, los objetos asociados al modelo de la aplicación se dividen en dos tipos:

- Los objetos persistentes: son objetos del modelo propiamente dicho, aquellos que se persisten realmente en la base de datos. Son objetos cuya relación con otros objetos se realiza mediante el identificador único de los mismos y no con el objeto completo asociado.
- Los objetos "con datos": objetos que heredan de los objetos persistentes y que poseen atributos adicionales de los objetos relacionados y que se utilizarán en la parte de la vista.

Patrones de diseño e implementación de Kda

En la aplicación se han adoptado una serie de patrones de diseño:

“Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. ”

que nos sirven para estructurar la aplicación, solucionar la comunicación entre la vista, los controladores y el modelo, y simplificar el tratamiento de los datos (guardado, actualización y recuperación). El uso reiterado de patrones para el desarrollo de la aplicación facilita la resolución de problemas repetitivos y simplifica la resolución de incidencias.

Patrones de diseño adoptados en la aplicación

Los patrones utilizados en la aplicación son:

- Inversion of Control: usando el patrón de inyección de dependencias independizamos el uso de los servicios que implementan el patrón “Service Facade” de su instanciación dentro de la aplicación. Básicamente lo que se hace es definir que objetos, que implementan una determinada interface, serán usados en otros objetos. El inyector (en este caso Spring) busca los objetos definidos para una anotación en concreto y, una vez realizada la construcción del objeto que se está cargando, realiza el set de las instancias que se han definido.
- Entity: el patrón entidad sirve para identificar claramente las clases del modelo de negocio que serán guardadas y recuperadas de la base de datos. Todas las clases de la aplicación para la implementación de este patrón se encuentran en el paquete: *com.kda.data.entities*
- DAO: este patrón provee de una forma totalmente abstracta de acceso a datos. En nuestra aplicación JPA nos ofrece esta funcionalidad. Hemos creado una clase que actúa de gestora para realizar tanto la persistencia como el lanzamiento de consultas sobre el almacén de datos. Gracias a este patrón, la aplicación puede cambiar sin problemas

el lugar donde residen los datos de la misma. La implementación de este patrón se encuentra en las librerías utilizadas por la aplicación y en el paquete: *com.kda.data.utils*.

– Service Facade: este patrón sirve para exponer una serie de servicios ocultando la complejidad que se encuentra detrás de los mismos a través de una interface. En la aplicación se han especificado una serie de interfaces llamadas **Managers* que se han implementado mediante clases llamadas **ManagersImpl*. Estas clases ofrecen los servicios necesarios para que los controladores de las pantallas obtengan la información a mostrar, validen los datos tecleados y guarden todo lo necesario en la base de datos. También se encargan de la gestión de la sesión y de mantener los datos del usuario en la misma. Todas las clases de la aplicación para la implementación de este patrón se encuentran en el paquete: *com.kda.data.managers*.

– Controller: Se encarga de definir el comportamiento de la vista y de comunicarse con los servicios para obtener y guardar la información. Cada vista de la aplicación tiene asociada una clase que implementa este patrón. Todas las clases de la aplicación para la implementación de este patrón se encuentran en el paquete: *com.kda.display.controllers*.

– View: datos a visualizar y rellenar en la vista. Se trata de encapsular en beans los campos a visualizar o rellenar en la pantalla. Estos son los que realmente se visualizan y disponen de funciones para convertir un/unos objetos en formulario y un formulario en el objeto pertinente. Todas las clases de la aplicación para la implementación de este patrón se encuentran en el paquete: *com.kda.display.forms*.

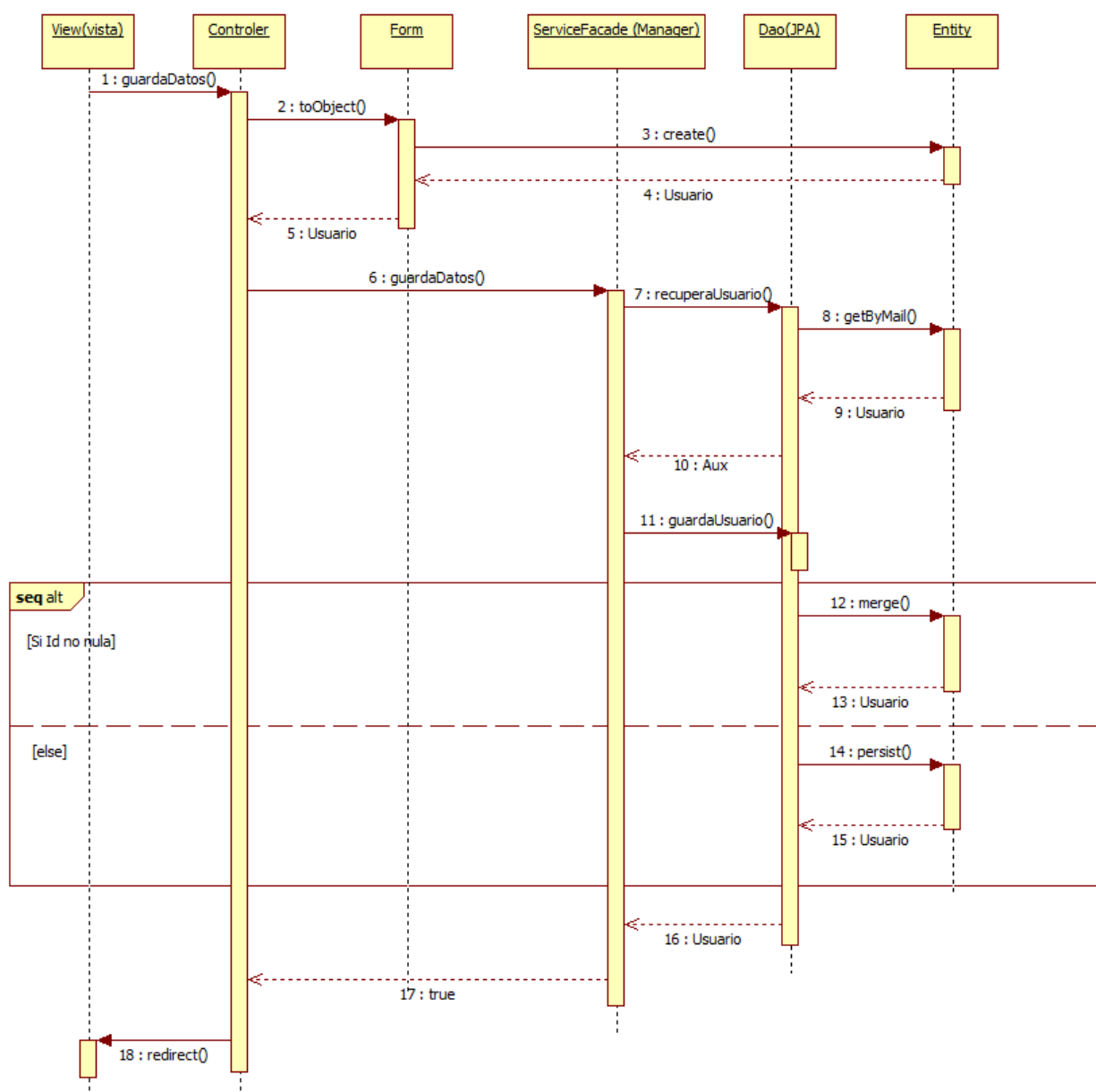
En general, todas las páginas de la aplicación funcionan de la misma forma:

1. La Vista tiene un Controlador asociado que gestiona la carga de los datos de la pantalla.
2. Al Controlador se le inyectan todos los Managers necesarios para poder recuperar la información a mostrar y gestionar su proceso.

3. Una vez creado el Controlador e inyectados los Managers, en un método `@PostConstruct` se carga toda la información a mostrar y gestionar y se rellena un Formulario que será el utilizado en la página para mostrar los datos.
4. En función de las acciones en pantalla, se realizan las llamadas por AJAX a métodos del Controler para refrescar o validar información.
5. Por último se somete la información de la vista cargada en el Formulario y el Controlador utiliza los Managers pertinentes para procesarla.
6. En caso de no haber errores, se redirecciona la vista a la página que convenga.

Ejemplo de funcionamiento mediante Diagrama de secuencia

Para mostrar el funcionamiento general de las pantallas con los distintos elementos, se incluye un diagrama de secuencia para definir el comportamiento de la función guardarUsuario() de la pantalla datos_usuario.xhtml del Controlador DatosUsuarioControler:



Diseño de base de datos

El diseño de la base de datos es el resultado de la definición de las tablas de la misma a partir del modelo de clases propuesto.

En primer lugar, tendremos una serie de maestros que servirán para visualizar los selectores de la aplicación y que se rellenarán directamente durante la creación de la base de datos. Para estas tablas, de momento, no hay mantenimiento y sus registros se insertan desde un script:

- AGRUPACION_AFICIONES
- AFICION
- PROVINCIA
- UBICACION

En segundo lugar, están las tablas de transacciones para la aplicación. Estas tablas son rellenadas por los usuarios mediante su registro, alta de actividades y comentarios sobre actividades. La tabla de las fotos de los usuarios se grabará en una tabla a parte de la de los datos de usuario:

- USUARIO
- ACTIVIDAD
- ACTIVIDAD_SUSCRITA
- ACTIVIDAD_DESCARTADA
- COMENTARIO_ACTIVIDAD
- USUARIO_FOTO

Por último tendremos las tablas que se utilizan para asociar los usuarios a las Aficiones y a los municipios:

- USUARIO_AFICION
- USUARIO_UBICACION

Para todas las tablas, además de los campos necesarios para guardar los datos, se han creado campos de identificación únicos por registro a partir de una secuencia. Estos campos únicos se usarán para las referencias entre tablas mediante claves foráneas.

Diseño lógico de la base de datos

A partir del diagrama de clases presentado en el apartado anterior se obtiene el siguiente diseño lógico:

PROVINCIA

```
(  
    id, provincia, provinciaseo, provincia3  
)
```

UBICACION

```
(  
    id, idprovincia, poblacion, poblacionseo, postal, latitud, longitud  
    ON {idprovincia} REFERENCIA PROVINCIA  
)
```

AGRUPACION_AFICIONES

```
(  
    id, nombre  
)
```

AFICION

```
(  
    id, agrupacion_id, nombre, descripcion  
    ON {agrupacion_id} REFERENCIA AGRUPACION_AFICIONES  
)
```

USUARIO

```
(  
    id, mail, password, nombre, apellido1, apellido2, telefono, movil,  
    extracto  
)
```

ACTIVIDAD

```
(  
    id, usuarioCreador, aficion_id, ubicacion_id, fechaCelebracion,  
    quedarEn, descripcion  
    ON {usuarioCreador} REFERENCIA USUARIO  
    y {aficion_id} REFERENCIA AFICION  
    y {ubicacion_id} REFERENCIA UBICACION  
)
```

COMENTARIO_ACTIVIDAD

```
(  
    id, usuario_id, actividad_id, fecha, comentario,  
    comentarioSuscripcion  
    ON {usuario_id} REFERENCIA USUARIO  
    y {actividad_id} REFERENCIA ACTIVIDAD  
)
```

ACTIVIDAD_SUSCRITA

```
(  
    id, usuario_id, actividad_id, fecha  
    ON {usuario_id} REFERENCIA USUARIO  
    y {actividad_id} REFERENCIA ACTIVIDAD  
)
```

ACTIVIDAD_DESCARTADA

```
(
```

```
    id, usuario_id, actividad_id, fecha  
    ON {usuario_id} REFERENCIA USUARIO  
    y {actividad_id} REFERENCIA ACTIVIDAD  
)
```

USUARIO_FOTO

```
(  
    id, usuario_id, foto  
    ON {usuario_id} REFERENCIA USUARIO  
)
```

USUARIO_AFICION

```
(  
    id, usuario_id, aficion_id  
    ON {usuario_id} REFERENCIA USUARIO  
    y {aficion_id} REFERENCIA AFICION  
)
```

USUARIO_UBICACION

```
(  
    id, usuario_id, municipio_id  
    ON {usuario_id} REFERENCIA USUARIO  
    y {municipio_id} REFERENCIA MUNICIPIO  
)
```

Diseño de las tablas de la base de datos

Una vez definido el diseño lógico se incluye el script de creación de las diferentes tablas de la aplicación. Todas tienen como clave primaria el identificador único autoincremental y, como claves foráneas, hacen referencia a esta identificación cuando sea necesario:

```

CREATE TABLE kda.AGRUPACION_AFICIONES
(
    id            int UNSIGNED NOT NULL auto_increment,
    nombre       VARCHAR(50) NOT NULL,
    PRIMARY KEY (id)
);

CREATE TABLE kda.AFICION
(
    id                int UNSIGNED NOT NULL auto_increment,
    agrupacion_id    int UNSIGNED,
    nombre           VARCHAR(50) NOT NULL,
    descripcion      VARCHAR(100) DEFAULT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (agrupacion_id) REFERENCES AGRUPACION_AFICIONES(id)
);

CREATE TABLE kda.PROVINCIA (
    id                int unsigned NOT NULL auto_increment,
    provincia         varchar(50) NOT NULL,
    provinciaseo     varchar(50) NOT NULL,
    provincia3       char(3) default NULL,
    PRIMARY KEY (id),
    UNIQUE KEY provinciaseo (provinciaseo)
);

CREATE TABLE kda.UBICACION (
    id                int unsigned NOT NULL auto_increment,
    idprovincia       int unsigned NOT NULL,
    poblacion         varchar(150) NOT NULL,
    poblacionseo     varchar(150) default NULL,
    postal            int(5) unsigned zerofill default NULL,
    latitud           decimal(9,6) default NULL,
    longitud          decimal(9,6) default NULL,
    PRIMARY KEY (id),
    UNIQUE KEY poblacionseo (poblacionseo),
    UNIQUE KEY lugar (latitud,longitud),
    KEY idprovincia (idprovincia)
);

CREATE TABLE kda.USUARIO
(
    id                int UNSIGNED NOT NULL auto_increment,

```

```

mail          VARCHAR(50) NOT NULL,
password      VARCHAR(50) NOT NULL,
nombre        VARCHAR(50) NOT NULL,
apellido1     VARCHAR(50) DEFAULT NULL,
apellido2     VARCHAR(50) DEFAULT NULL,
telefono      VARCHAR(20) DEFAULT NULL,
movil         VARCHAR(20) DEFAULT NULL,
extracto      VARCHAR(500) NOT NULL,
PRIMARY KEY (id)
);

CREATE TABLE kda.ACTIVIDAD
(
  id            int UNSIGNED NOT NULL auto_increment,
  usuarioCreador int UNSIGNED,
  aficion_id    int UNSIGNED,
  ubicacion_id int(10) UNSIGNED,
  fechaCelebracion timeStamp NOT NULL,
  quedarEn     VARCHAR(200) NOT NULL,
  descripcion   VARCHAR(500) NOT NULL,
PRIMARY KEY (id),
FOREIGN KEY (usuarioCreador) REFERENCES USUARIO(id),
FOREIGN KEY (aficion_id)     REFERENCES AFICION(id),
FOREIGN KEY (ubicacion_id)   REFERENCES UBICACION(id)
);

CREATE TABLE kda.COMENTARIO_ACTIVIDAD
(
  id            int UNSIGNED NOT NULL auto_increment,
  usuario_id    int UNSIGNED,
  actividad_id  int UNSIGNED,
  fecha         timestamp NOT NULL,
  comentario    VARCHAR(500) NOT NULL,
  comentarioSuscripcion int UNSIGNED,
PRIMARY KEY (id),
FOREIGN KEY (usuario_id) REFERENCES USUARIO(id),
FOREIGN KEY (actividad_id) REFERENCES ACTIVIDAD(id)
);

CREATE TABLE kda.ACTIVIDAD_SUSCRITA
(
  id            int UNSIGNED NOT NULL auto_increment,
  usuario_id    int UNSIGNED,
  actividad_id  int UNSIGNED,
  fecha         timestamp NOT NULL,
PRIMARY KEY (id),
FOREIGN KEY (usuario_id) REFERENCES USUARIO(id),
FOREIGN KEY (actividad_id) REFERENCES ACTIVIDAD(id)
);

CREATE TABLE kda.ACTIVIDAD_DESCARTADA
(
  id            int UNSIGNED NOT NULL auto_increment,
  usuario_id    int UNSIGNED,
  actividad_id  int UNSIGNED,

```

```
        fecha                timestamp NOT NULL,  
        PRIMARY KEY (id),  
        FOREIGN KEY (usuario_id) REFERENCES USUARIO(id),  
        FOREIGN KEY (actividad_id) REFERENCES ACTIVIDAD(id)  
    );  
  
CREATE TABLE kda.USUARIO_FOTO  
(  
    id                int UNSIGNED NOT NULL auto_increment,  
    usuario_id        int UNSIGNED,  
    foto              LONGBLOB NOT NULL,  
    PRIMARY KEY (id),  
    FOREIGN KEY (usuario_id) REFERENCES USUARIO(id)  
);  
  
CREATE TABLE kda.USUARIO_AFICION  
(  
    id                int UNSIGNED NOT NULL auto_increment,  
    usuario_id        int UNSIGNED,  
    aficion_id        int UNSIGNED,  
    PRIMARY KEY (id),  
    FOREIGN KEY (usuario_id) REFERENCES USUARIO(id),  
    FOREIGN KEY (aficion_id) REFERENCES AFICION(id)  
);  
  
CREATE TABLE kda.USUARIO_UBICACION  
(  
    id                int UNSIGNED NOT NULL auto_increment,  
    usuario_id        int UNSIGNED,  
    ubicacion_id     int UNSIGNED,  
    PRIMARY KEY (id),  
    FOREIGN KEY (usuario_id) REFERENCES USUARIO(id),  
    FOREIGN KEY (ubicacion_id) REFERENCES UBICACION(id)  
);
```

Para poder gestionar los identificadores únicos de cada tabla, OpenJPA necesita la definición de una tabla que será la encargada de mantener el identificador autoincremental.

```
CREATE TABLE kda.OPENJPA_SEQUENCE_TABLE  
(  
    id                tinyint(4) NOT NULL,  
    SEQUENCE_VALUE    bigint(20) default NULL,  
    PRIMARY KEY (id)  
)
```


Mapa del sitio web

Kda consta de una parte pública, que será el lugar donde se definen y presentan las bondades de la aplicación, y una parte privada, que es el lugar donde los usuarios compartirán sus aficiones y definirán las actividades.

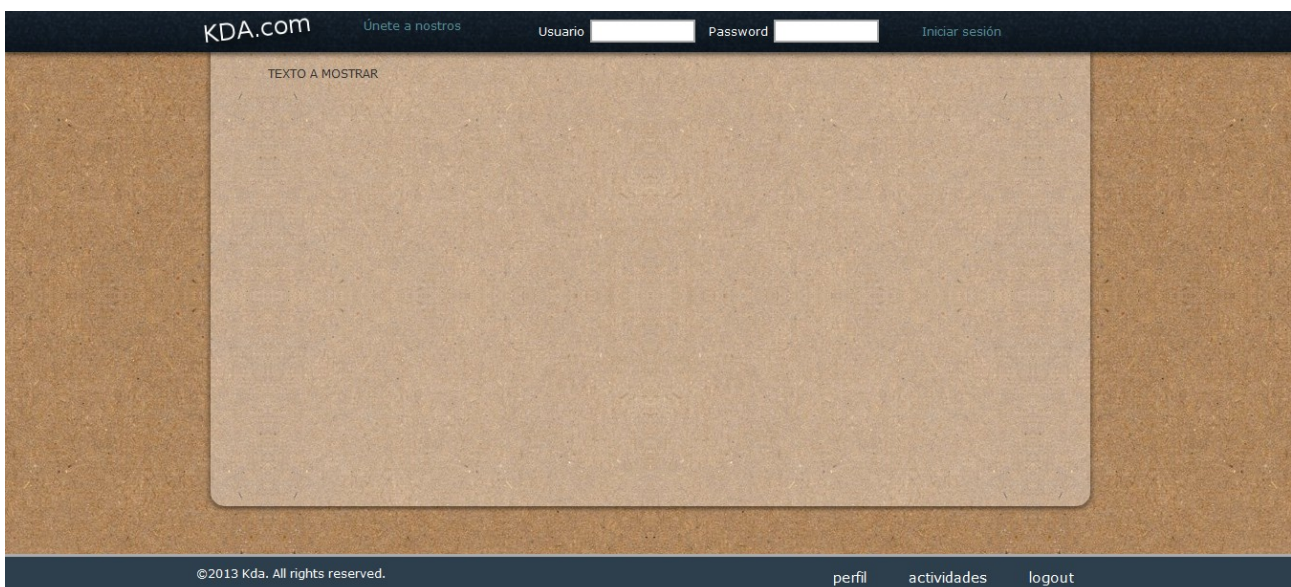
Los apartados en los que se divide el sitio y componen el mapa de la web son los siguientes:

Portada (inicio)

Esta página sería la portada del sitio y es donde se introduciría publicidad, breves explicaciones del funcionamiento de Kda, etc. No se ha implementado una portada como tal ya que la parte pública no era el objetivo final del proyecto.

En esta pantalla se incluyen dos links imprescindibles para la aplicación:

- El registro: desde esta pantalla se accede a la pantalla de entrada de datos del usuario para darlo de alta en el sistema.
- El Login: introduciendo el usuario y la contraseña se puede acceder al espacio de trabajo.



Pizarra

Se trata de la página principal de la aplicación. El espacio de usuario donde se realizan todas las funcionalidades de las que dispone la aplicación: compartir aficiones, crear actividades, suscribirse o descartarse de las actividades de las que somos avisados y comentar actividades a las que estamos suscritos. Se pueden modificar datos del perfil para cambiar nuestro "Extracto", nuestras ubicaciones donde realizar actividades o modificar nuestras aficiones.

Se divide en cuatro apartados bien diferenciados:

- Datos del perfil.
- Nuevos avisos: de actividades.
- Próximas actividades: información de las actividades a las que estamos suscritos y van a ser realizadas
- Últimos comentarios: de las actividades a las que estamos o hemos estado suscritos.

The screenshot shows the user profile page for 'Jorge' on KDA.com. The page is divided into four main sections:

- Profile Summary:** Shows the user's name 'Jorge', a profile picture, and an 'Editar' button. Below this is an 'Extracto' section with a short bio: 'Me gusta mucho el baloncesto, por eso busco gente para jugar partidos de vez en cuando. Tambien corro un rato todas las tardes. Pero tambien me gusta salir de fiesta y ver las estrellas y los pla'.
- Nuevos avisos (New Alerts):** A section with a warning icon containing two alerts. The first is for 'Baloncesto' in Calviá, created by Rosi, with a 'Suscribir' (Subscribe) and 'Descartar' (Dismiss) button. The second is for 'Astronomia' in Na Burguesa, also created by Rosi, with similar buttons.
- Tus próximas actividades (Your upcoming activities):** A section with a clock icon and a 'Nueva Actividad' button. It lists two activities: 'Atletismo' in Calviá on Fri Jun 14 18:06:00 CEST 2013, and 'Baloncesto' in Palma de Mallorca on Sat Jun 22 13:00:00 CEST 2013. Each activity has 'Comentar Actividad' and 'Más detalles' links.
- Los últimos comentarios (Latest comments):** A section with a speech bubble icon showing three comments from other users (Rosi, Jorge, and Jorge) on various activities, each with a 'Comentar Actividad' link.

The footer of the page includes the copyright notice '©2013 Kda. All rights reserved.' and navigation links for 'perfil', 'actividades', and 'logout'.

Datos de Usuario

Es el lugar donde se introduce toda la información del usuario: mail, nombre, apellidos, contraseña, teléfonos, un extracto o pequeña explicación de nuestras aficiones y las localidades y aficiones para las que queremos recibir avisos o de las que daremos de alta actividades. Esta pantalla servirá para el registro y la modificación de los datos del usuario.

The screenshot shows the 'Datos de Usuario' (User Data) form on the KDA.com website. The form is set against a brown, textured background. At the top left is the 'KDA.com' logo, and at the top right are 'Logout' and 'Pizarra' links. The form is titled 'Datos de Usuario' and contains several sections:

- Mail:** A text input field containing 'j.merino@uoc.com'. A red 'Requerido' label is to its right.
- Password:** A text input field with masked characters. A red 'Requerido' label is to its right.
- Repite Password:** A text input field with masked characters. A red 'Requerido' label is to its right.
- Extracto:** A text area containing the text: 'Me gusta mucho el baloncesto, por eso busco gente para jugar partidos de vez en cuando. Tambien corro un rato todas las tardes. Pero tambien me gusta salir de fiesta y ver las estrellas y los pla'. A red 'Requerido' label is to its right. Below the text area is a note: '5 caracteres.'.
- Nombre:** A text input field containing 'Jorge'. A red 'Requerido' label is to its right.
- Móvil:** A text input field containing '677-986532'.
- Primer apellido:** A text input field containing 'M.'.
- Teléfono:** An empty text input field.
- Segundo apellido:** A text input field containing 'L.'.
- Hobbies Section:** A grid of dropdown menus.
 - Left column: 'Tipo de Afición' (4 instances) with values: Deportes, Deportes, Actividades Nocturnas, Actividades Nocturnas, Select One.
 - Right column: 'Afición' (4 instances) with values: Atletismo, Baloncesto, Salir de Fiesta, Astronomia, Select One.
- Location Section:** A grid of dropdown menus.
 - Left column: 'Provincia' (4 instances) with values: Baleares, Baleares, Madrid, Select One, Select One.
 - Right column: 'Ubicacion' (4 instances) with values: Calvià, Palma de Mallorca, Alcalá de Henares, Select One, Select One.

At the bottom right of the form, there are two buttons: 'Cancelar' and 'Guardar'. The 'Guardar' button is highlighted with a green circle.

At the bottom of the page, there is a dark blue footer containing:

- ©2013 Kda. All rights reserved.
- perfil actividades logout

Datos de la Actividad

Página donde se dan de alta o se consultan los detalles de una actividad. Cuando se da de alta una actividad, se presentan sólo las aficiones y las localidades del usuario que la da de alta. En caso de que se trate de una consulta, se oculta el botón para guardar la información.

The screenshot shows a web form titled "Datos de la Actividad" on the KDA.com website. The form is set against a brown, textured background. At the top left of the page is the KDA.com logo, and at the top right are links for "Logout" and "Pizarra". The form fields are as follows:

- Fecha Celebración** (Required): A text input field containing "30/06/2013 15:00".
- Afición**: A dropdown menu with "Baloncesto" selected.
- Ubicación**: A dropdown menu with "Calvià" selected.
- Quedamos en** (Required): A text input field containing "Polideportivo de Calvià".
- Descripción** (Required): A text area containing the text "Hemos quedado con los de Capdella y habra partido intenso." Below the text area is a character count: "142 caracteres."

At the bottom right of the form, there are two buttons: "Cancelar" and "Guardar". The "Guardar" button is circled in green, indicating it is the primary action for saving the activity.

©2013 Kda. All rights reserved. perfil actividades logout

Funcionamiento de la suscripción a una actividad

En la Pizarra podemos suscribirnos a una de las actividades que aparece en el apartado "Nuevos avisos". Una vez suscritos, aparecerá una nueva sección en el apartado de "Tus próximas actividades":

The screenshot shows the KDA.com user interface. At the top, there is a navigation bar with "KDA.com", "Logout", and "Pizarra". The main content area is divided into several sections:

- User Profile:** A profile card for "Jorge" with an "Editar" button. Below it is an "Extracto" section with a short bio about basketball and running.
- Nuevos avisos:** A section with a yellow warning icon. It contains a notice from "Rosi" about an astronomy event in Calvià on Friday, June 28, 2013. There are "Suscribir" and "Descartar" buttons at the bottom.
- Tus próximas actividades:** A section with a clock icon and a "Nueva Actividad" button. It lists three upcoming activities:
 - Atletismo:** Faltan 2 días para tu próxima actividad de Atletismo. Lugar: Pabellón de Calvià. Fecha: Fri Jun 14 18:06:00 CEST 2013.
 - Baloncesto:** Faltan 7 días para tu próxima actividad de Baloncesto. Lugar: Pabellon de San Jose Obrero. Fecha: Wed Jun 19 10:00:00 CEST 2013. This entry is highlighted with a green border.
 - Baloncesto:** Faltan 10 días para tu próxima actividad de Baloncesto. Lugar: Palma Arena. Fecha: Sat Jun 22 13:00:00 CEST 2013.
- Los últimos comentarios:** A section with a speech bubble icon showing three recent comments from users Rosi and Jorge regarding the activities.

At the bottom of the page, there is a footer with "©2013 Kda. All rights reserved." and navigation links for "perfil", "actividades", and "logout".

Funcionamiento del descarte de una actividad

En la Pizarra podemos descartarnos de una de las actividades que aparece en el apartado "Nuevos avisos". Una vez descartados, el aviso desaparecerá y no volverá a mostrarse más:

The screenshot shows the user profile page for 'Jorge' on KDA.com. The page is set against a corkboard background. At the top, there are links for 'Logout' and 'Pizarra'. The profile section includes a name 'Jorge', an 'Editar' button, and an 'Extracto' with a short bio. Below this, the 'Nuevos avisos' section is highlighted with a green border, showing a notification icon and the text 'Nuevos avisos'. To the right, the 'Tus próximas actividades' section lists three upcoming events: 'Atletismo' (2 days away), 'Baloncesto' (7 days away), and another 'Baloncesto' (10 days away). At the bottom, the 'Los últimos comentarios' section shows three recent comments from other users.

KDA.com Logout Pizarra

Jorge Editar

Extracto

Me gusta mucho el baloncesto, por eso busco gente para jugar partidos de vez en cuando. Tambien corro un rato todas las tardes. Pero tambien me gusta salir de fiesta y ver las estrellas y los pla

Nuevos avisos

Tus próximas actividades Nueva Actividad

Faltan **2 días** para tu próxima actividad de **Atletismo**

Lugar **Pabellón de Calviá** en la localidad de **Calviá**.
Fecha **Fri Jun 14 18:06:00 CEST 2013**

[Comentar Actividad](#) [Más detalles](#)

Faltan **7 días** para tu próxima actividad de **Baloncesto**

Lugar **Pabellon de San Jose Obrero** en la localidad de **Palma de Mallorca**.
Fecha **Wed Jun 19 10:00:00 CEST 2013**

[Comentar Actividad](#) [Más detalles](#)

Faltan **10 días** para tu próxima actividad de **Baloncesto**

Lugar **Palma Arena** en la localidad de **Palma de Mallorca**.
Fecha **Sat Jun 22 13:00:00 CEST 2013**

[Comentar Actividad](#) [Más detalles](#)

Los últimos comentarios

Rosi
De la actividad de **Atletismo** del **Fri Jun 14 18:06:00 CEST 2013**
Yo vengo pero no tengo pelota.
[Comentar Actividad](#)
Sun Jun 09 18:07:43 CEST 2013

Jorge
De la actividad de **Baloncesto** del **Wed Jun 05 16:00:00 CEST 2013**
Pues traete pan.
[Comentar Actividad](#)
Mon Jun 03 23:31:31 CEST 2013

Jorge
De la actividad de **Baloncesto** del **Wed Jun 05 16:00:00 CEST 2013**
Me apunto. ¿Qué traigo?
[Comentar Actividad](#)
Mon Jun 03 23:31:07 CEST 2013

©2013 Kda. All rights reserved. perfil actividades logout

Comentar Actividad

En la Pizarra, tanto en apartado de "Tus próximas actividades" como en apartado de "Últimos comentarios", tenemos la opción de comentar una actividad a la que estamos suscritos. En esta pantalla podemos añadir un comentario que aparecerá en la lista de comentarios de todos los usuarios suscritos a la actividad:

KDA.com Logout Pizarra

Comentar Actividad

De la actividad de **Baloncesto**

del **Sat Jun 22 13:00:00 CEST 2013**

Descripción
Vamos a jugar a Basquet.

Comentario Requiendo

Me apunto.
Llevo la pelota y las bebidas.

158 caracteres.

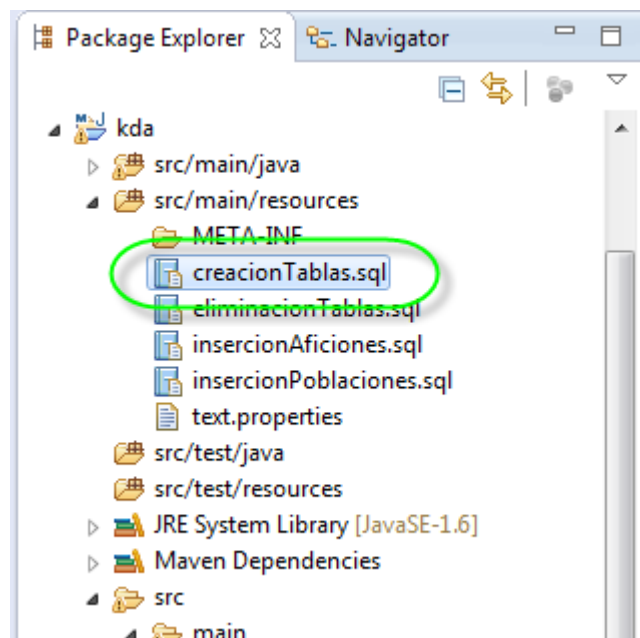
©2013 Kda. All rights reserved. perfil actividades logout

Instalación de la aplicación

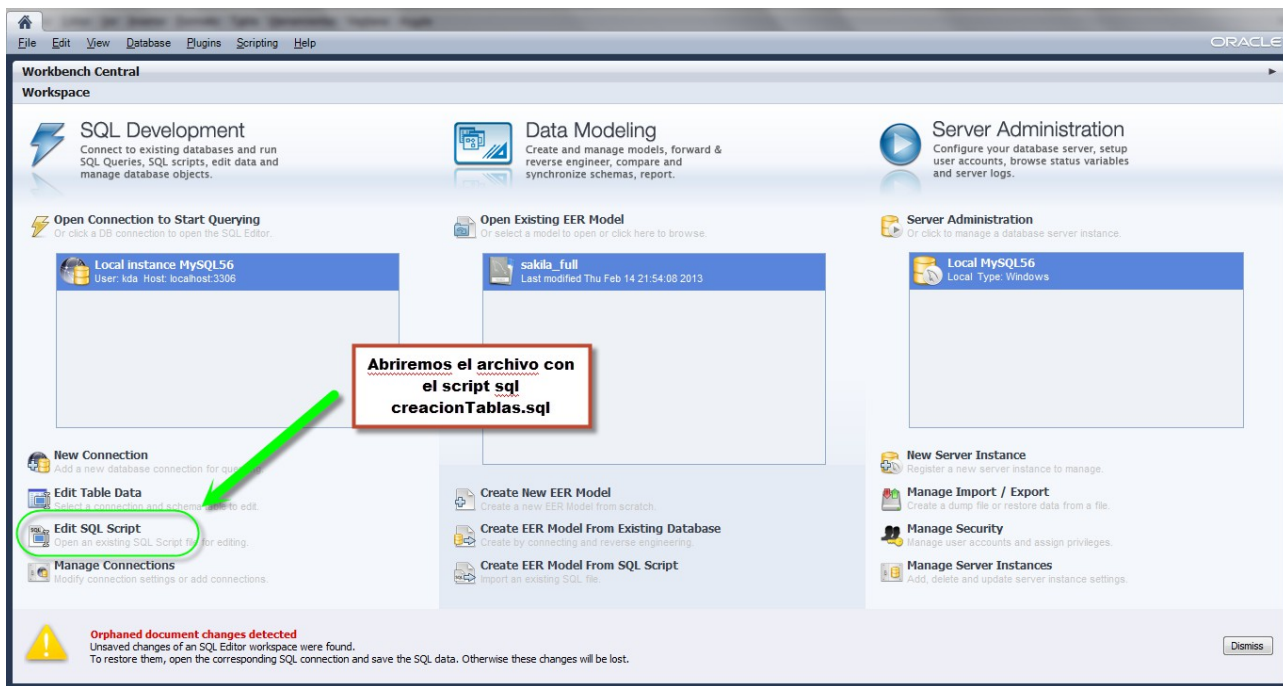
Creación de la base de datos kda

Una vez instalado MySQL y ya en el escritorio de MySQL, deberemos ejecutar el script para la creación de la base de datos que se encuentra dentro del árbol de directorios dentro de kda:

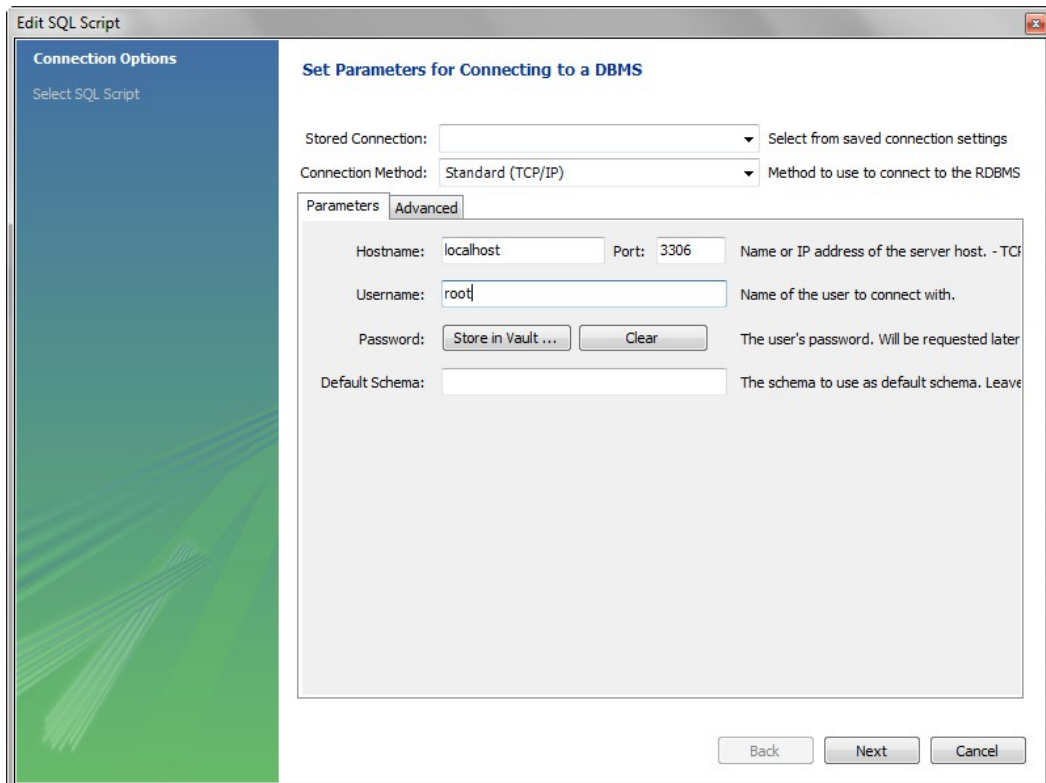
kda\src\main\resources\creacionTablas.sql

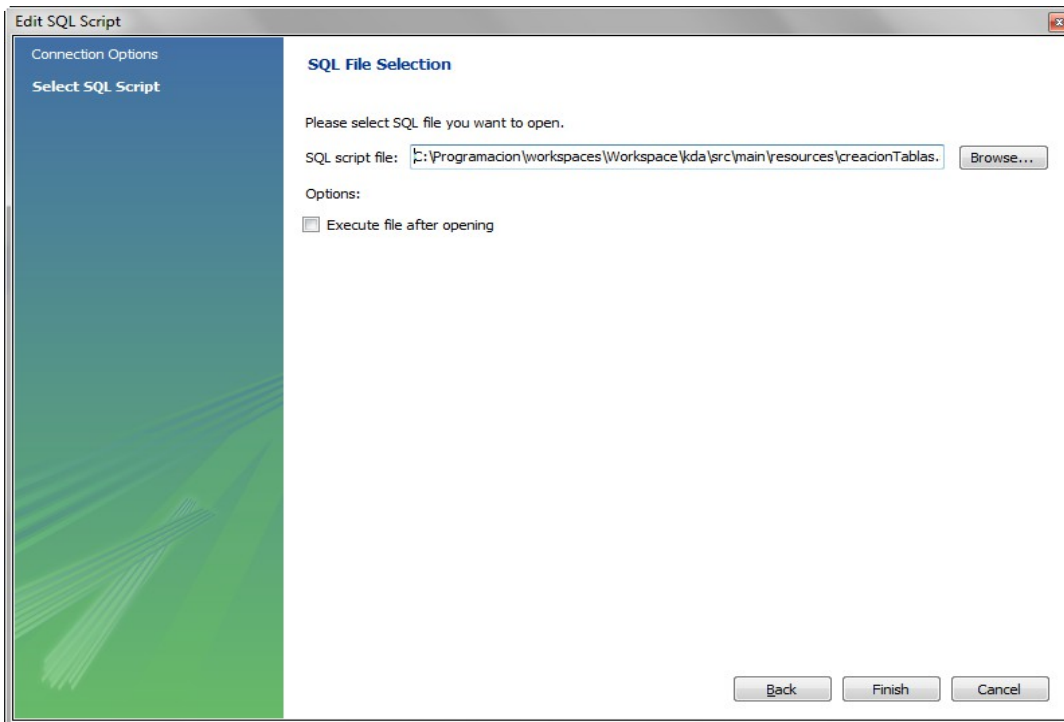


Ejecutamos la opción "Edit SQL Script"

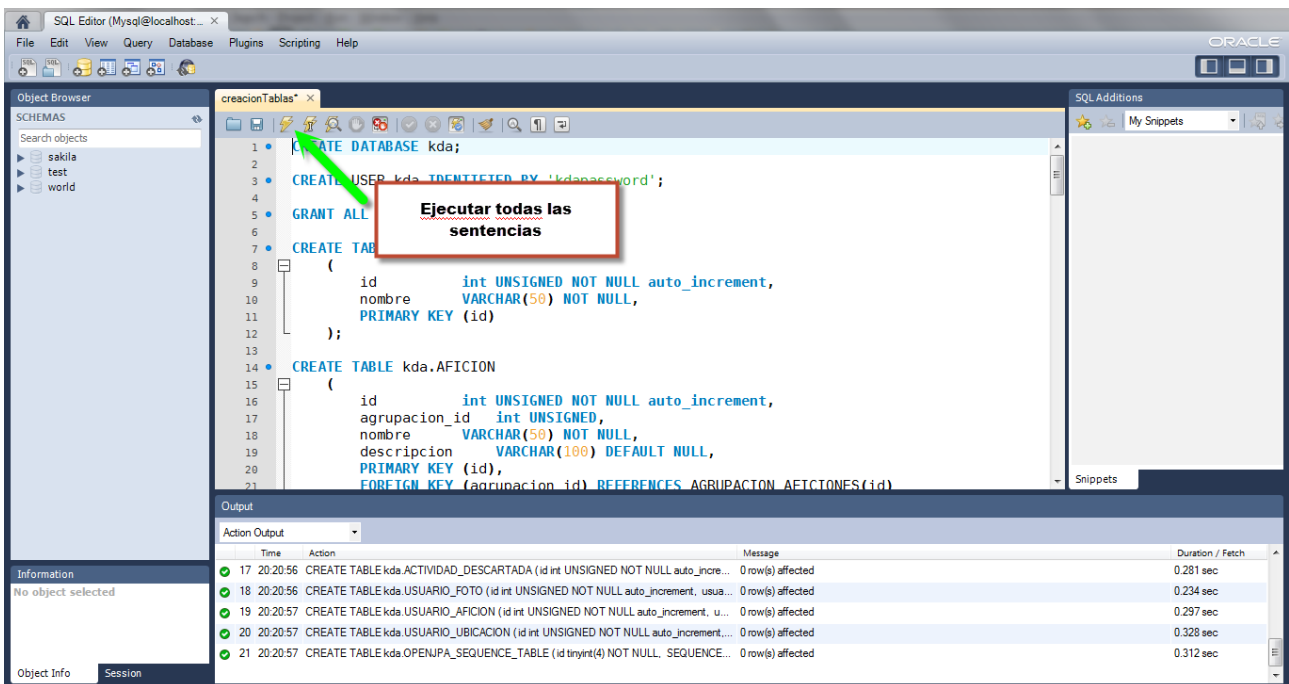


Indicamos el usuario con el que nos conectamos y el archivo que vamos a abrir:





Una vez abierto el archivo con el script bastará con ejecutar las sentencias:



Relleno de la información inicial

Hay una serie de tablas que se rellenarán con datos antes de poder hacer uso de la aplicación:

- Agrupación Aficiones.
- Aficiones.
- Provincias.
- Ubicaciones.

Para rellenarlas inicialmente, deberemos ejecutar las sentencias de los archivos del directorio kda:

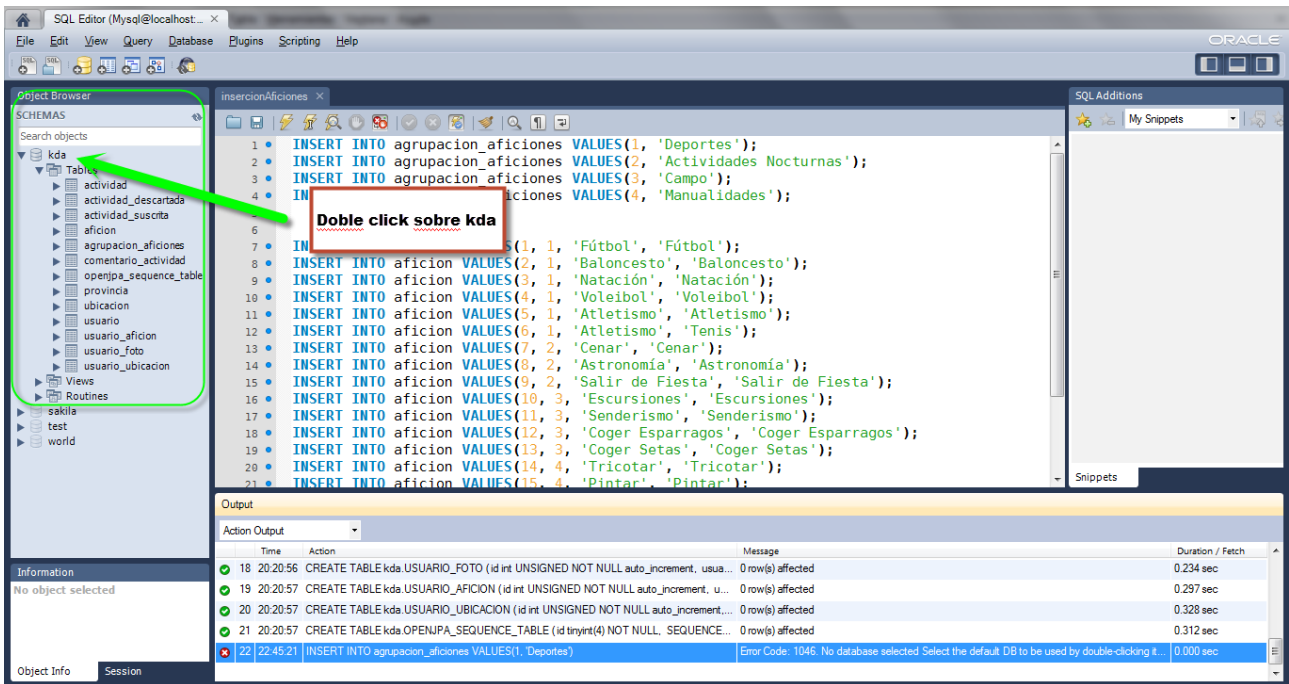
```
kda\src\main\resources\insercionAficiones.sql
```

```
kda\src\main\resources\insercionPoblaciones.sql
```

Abriremos cada uno de los archivos y ejecutaremos las sentencias de igual forma que ejecutamos las de creación de la base de datos. Es posible que se produzca un error:

Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCHEMAS list in the sidebar.

Será debido a que no tenemos ninguna base de datos seleccionada, por lo que actuaremos como nos indica el error. Refrescaremos la ventana del "Object Browser" y clicaremos 2 veces sobre la DB kda.



Una vez insertados todos los registros en la base de datos podemos proceder a la creación de la aplicación para que pueda ser desplegada en un Tomcat.

Historial de la instalación del entorno de trabajo

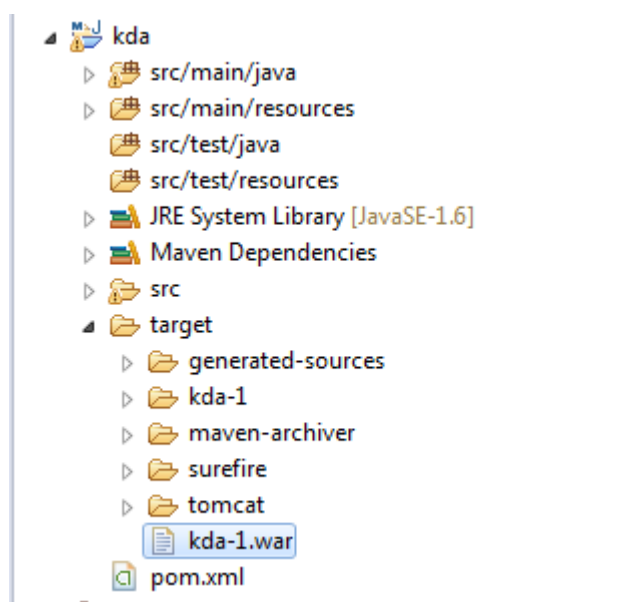
Durante el proceso de análisis y diseño de la aplicación se ha procedido a instalar una serie de herramientas y librerías para poder ir adelantando trabajo de cara a la implementación del sitio. En este apartado se comentan el resultado y las incidencias más destacadas:

- Se ha instalado como herramienta de desarrollo Eclipse. Para el mismo, se ha instalado un plugin para la utilización de Maven. Maven simplificará mucho la gestión de las dependencias existentes con las librerías utilizadas en el proyecto, así como el despliegue de la aplicación en el servidor Tomcat.
- Una vez instalado Maven, se ha configurado el archivo pom.xml para que se descarguen las librerías de JSF (jsf-api y jsf-impl), las de PrimeFaces (los componentes JSF utilizados en el proyecto) y las del servidor de servlets.
- Con el fin de probar la efectividad de la configuración realizada, se ha hecho una pequeña prueba montando una página simple con facelets, estilos y utilizando componentes JSF de PrimeFaces.
- Al intentar ejecutar el proyecto con tomcat:run se ha producido el problema de que no se cargaban correctamente las librerías de jsf 2.0 en el servidor de aplicaciones. De hecho, el war generado funcionaba correctamente en un Tomcat descargado pero no al ejecutar la aplicación con Maven desde Eclipse. Se ha localizado la solución que pasaba por ejecutar el proyecto mediante tomcat:run-war.
- Una vez solucionado el problema la aplicación se ha ejecutado respetando todas las definiciones (managebeans y estilos) realizadas.

Creación del archivo war y ejecución de la aplicación

El proyecto se ha desarrollado utilizando Eclipse e indicando durante su creación que se trataba de un proyecto Maven, por lo que todas las dependencias, repositorios desde donde extraer las librerías, los plugins y el entorno donde se va a ejecutar la aplicación han sido definidos dentro de un archivo pom.xml.

Para crear el war asociado a la aplicación, bastará con ejecutar el archivo pom.xml que se descargará las librerías de las dependencias, los plugins y el servidor Tomcat configurado para poder ejecutar directamente con "tomcat:run-war" una vez finalizada la creación del war:



El war generado será "kda-1.war" debido a que se ha configurado en el archivo pom.xml como versión 1 de la aplicación.

Ver la aplicación en funcionamiento

El war "kda-1.war" está preparado para poder ser desplegado en un Tomcat una vez colocado en el directorio "webapps" de dicho Tomcat. Para acceder a la aplicación se deberá teclear la Url en el navegador:

ipServidor:xxxx/kda-1/inicio.xhtml

Donde:

- ipServidor: es la ip del servidor donde se aloja Tomcat,"localhost" para el sistema local.
- xxxx: es el puerto donde se ha configurado el acceso al servidor.
- /kda-1/inicio.xhtml: sitio y página web para el acceso a la aplicación.

Planificación temporal del TFC

Planificación seguida según las fechas de entrega para las Pecs

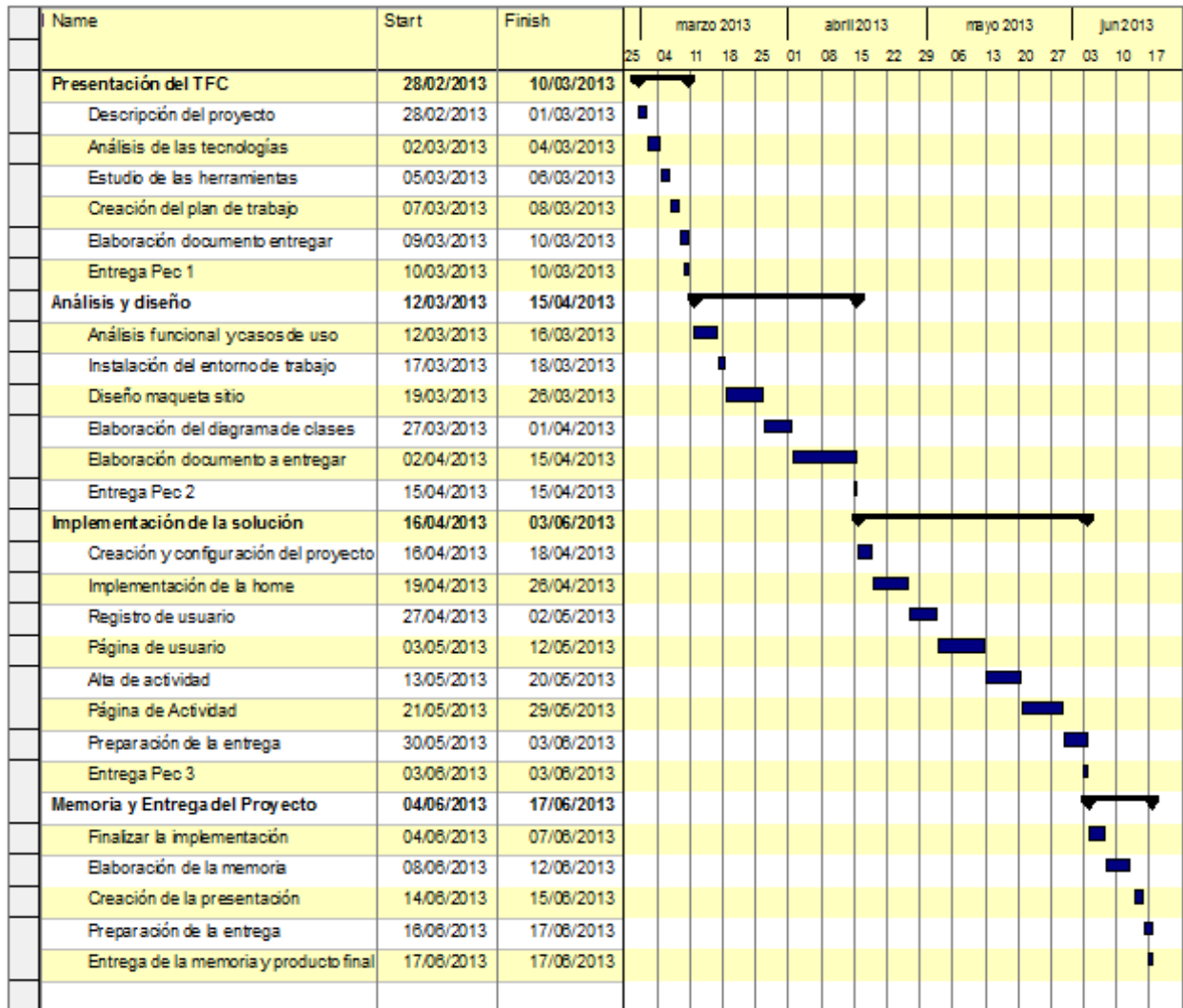
De acuerdo a las fecha de entrega de las diferentes pecs se seguido la siguiente planificación:

Práctica	Descripción	Fecha Inicio	Fecha Fin
<i>Pec 1</i>	<i>Presentación del TFC</i>	28/02/13	11/03/13
	<ul style="list-style-type: none"> • Descripción del proyecto • Análisis de las tecnologías a emplear • Estudio de las herramientas a utilizar e instalación de los programas para la creación de documentos y diagramas. • Creación del plan de trabajo y el diagrama de gantt. • Elaboración documento a entregar Pec 1 • Entrega Pec 1 	28/02/13 02/03/13 05/03/13 07/03/13 09/03/13 10/03/03	01/03/13 04/03/13 06/03/13 08/03/03 10/03/13 10/03/03
<i>Pec 2</i>	<i>Análisis y diseño</i>	12/03/13	15/04/13
	<ul style="list-style-type: none"> • Análisis funcional y casos de uso • Instalación del entorno de trabajo • Diseño maqueta sitio (no operativo) • Elaboración del diagrama de clases • Elaboración documento a entregar Pec 2 • Entrega Pec 2 	12/03/13 17/03/13 19/03/13 27/03/13 02/04/13 15/04/13	16/03/13 18/03/13 26/03/13 01/04/13 15/04/13 15/04/13
<i>Pec 3</i>	<i>Implementación de la solución</i>	16/04/13	03/06/13
	<ul style="list-style-type: none"> • Creación y configuración del proyecto • Implementación de la home 	16/04/13 19/04/13	18/04/13 26/04/13

	<ul style="list-style-type: none"> • Registro de usuario 	27/04/13	02/05/13
	<ul style="list-style-type: none"> • Página de usuario 	03/05/13	12/05/13
	<ul style="list-style-type: none"> • Alta de actividad 	13/05/13	20/05/13
	<ul style="list-style-type: none"> • Página de Actividad 	21/05/13	29/05/13
	<ul style="list-style-type: none"> • Preparación de la entrega y elaboración documento de la Pec 3 	30/05/13	02/06/13
	<ul style="list-style-type: none"> • Entrega Pec 3 	03/06/13	03/06/13
Memoria y Presentación	Memoria y Entrega del Proyecto	04/06/13	17/06/13
	<ul style="list-style-type: none"> • Elaboración de la memoria 	08/06/13	12/06/13
	<ul style="list-style-type: none"> • Creación de la presentación 	13/06/13	15/06/13
	<ul style="list-style-type: none"> • Preparación de la entrega 	16/06/13	17/06/13
	<ul style="list-style-type: none"> • Entrega de la memoria y producto final 	17/06/13	17/06/13

Diagrama de Gantt

La planificación visualizada de forma gráfica mediante un diagrama de Gantt es la siguiente:



Conclusiones y futuras modificaciones

Cuando se comenzó el proyecto se establecieron una serie de premisas para su realización:

- Basado en el ecosistema J2EE.
- Todas las herramientas y librerías debían ser gratuitas y, en la medida de lo posible, de código libre y de libre distribución.
- Separación en tres capas Modelo-Vista-Controlador.
- Uso de patrones que resuelvan problemas determinados y que ofrezcan independencia entre los módulos de la aplicación.
- Persistencia independiente de la base de datos utilizada.
- Uso de herramientas con gran aceptación y base instalada.
- Automatizar la creación del war y la recuperación de las dependencias.

Todas estas premisas han sido cubiertas en este proyecto sin problemas con las herramientas y librerías utilizadas. Se ha obtenido una aplicación completamente funcional y que se puede desplegar en cualquier servidor dedicado de un proveedor de servicios que soporte java.

Durante la realización del proyecto se han ido encontrando problemas que retrasaban su realización ligados al desconocimiento de las herramientas o librerías. Una incidencia que retrasó bastante el desarrollo fue la causada por un problema conocido como "Redirect after Post" que se suele producir al utilizar una redirección de página pero sin modificar la url del navegador. Se solventó utilizando una técnica llamada "Post-Redirect-Get" que lo que hace es indicarle a JSF que debe, además de cargar la página indicada, modificar la url del navegador realizando una redirección. Una vez comprendida la base del problema fue muy fácil arreglarlo en toda la aplicación.

En conclusión, al finalizar el proyecto, quedan totalmente cumplidas las premisas que se establecieron al principio.

Futuro de Kda

Una vez que se tiene la aplicación funcionando completamente, lo siguiente será adaptarla a otros dispositivos. Principalmente se trataría de adaptar las pantallas para que pudieran ser funcionales desde el navegador de un dispositivo móvil.

Otra opción, sería crear una serie de servicios web basados en las funcionalidades de la aplicación de cara a poder ser consultados e invocados desde clientes remotos. Dichos clientes podrían integrarse en cualquier tipo de aplicación, incluidas aplicaciones nativas de Android o iOS.

Cuando la aplicación sea estable y pueda ser utilizada desde múltiples dispositivos, lo siguiente será dotarla de más funcionalidades como poder rescatar una actividad descartada, poder descartar una aplicación a la que ya se está suscrito, cambiar una actividad de fecha informando a todos los suscritos, etc.

Por último, y no por ello menos importante, sería muy interesante poder integrarla con otras redes sociales y poder publicar actividades vía Twitter o Facebook.

Bibliografía y recursos

Toda la bibliografía y recursos utilizados han sido obtenidos directamente de Internet. Se indican las páginas más relevantes de los proyectos utilizados pero, en general, todas las librerías han sido incluidas en el proyecto a través de Maven:

J2EE:

<http://www.oracle.com/technetwork/java/javaee/downloads/index.html>

MySQL:

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

PrimeFaces:

<http://primefaces.org/downloads.html>

Tomcat:

<http://tomcat.apache.org/download-60.cgi>

Spring:

<http://www.springsource.org/spring-community-download>

OpenOffice:

<http://www.openoffice.org/download/>

StartUML:

<http://staruml.sourceforge.net/en/download.php>

Eclipse:

<http://www.eclipse.org/downloads/moreinfo/jee.php>

Gimp:

<http://www.gimp.org/downloads/>

Páginas utilizadas como bibliografía para la consecución del proyecto:

JSF2:

Como tutorial base se utilizó la página:

<http://www.coreservlets.com/JSF-Tutorial/jsf2/>

Pero inestimables han sido los artículos de Mkyong que aporta todo tipo de soluciones y tutoriales para aprovechar todas las funcionalidades de JSF:

<http://www.mkyong.com>

y principalmente la página dedicada al tutorial de JSF 2.0:

<http://www.mkyong.com/tutorials/jsf-2-0-tutorials/>

Otra fuente utilizada para ampliar el conocimiento de PrimeFaces y JSF fue:

<http://www.adictosaltrabajo.com/tutoriales.php>

En concreto:

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=introduccionPrimefaces>

Para JPA se han utilizado diversas fuentes ya que el uso de las librerías de OpenJPA se hace sobre un contenedor de Servlets y no sobre un servidor de aplicaciones, siendo diferente su comportamiento:

<http://openjpa.apache.org/getting-started.html>

<http://binodsuman.blogspot.com.es/2009/10/jpa-example-outside-container-jpa.html>

<http://devbank.wordpress.com/2011/05/25/jpa-stand-alone-project-with-hibernate-and-mysql/>

Para los patrones de diseño:

<http://msdn.microsoft.com/es-es/library/bb972240.aspx>

<http://www.lab.inf.uc3m.es/~a0080802/RAI/mvc.html>

<http://danielggarcia.wordpress.com/2009/05/13/abstraccion-de-datos-i-el-patron-dao/>

<http://blog.xebia.com/2009/05/11/jpa-implementation-patterns-service-facades-and-data-transfers-objects/>

Spring:

<http://www.mkyong.com/spring/spring-auto-wiring-beans-with-autowired-annotation/>

<http://www.roseindia.net/tutorial/spring/spring3/ioc/springautowiremode.html>

MySQL:

<http://dev.mysql.com/doc/refman/5.0/es/tutorial.html>