 Universidad Católica de Córdoba

Trabajo final

Base de Datos Aplicada

Nombre del proyecto

Docente: Ing. Mariano, García Mattío

Fecha: 19/11/2013

Autores: Antonel Stefano, [stefanoantonel@gmail.com](mailto:stefanoantonel@gmail.com)

Bonansea María Florencia, florenciabonansea@gmail.com

# Índice

[Introducción 4](#_Toc372449608)

[Alcances y limites 5](#_Toc372449609)

[Ámbito de la aplicación: 6](#_Toc372449610)

[Funcionamiento 7](#_Toc372449611)

[Módulo Login: 7](#_Toc372449612)

[Modulo Registro: 7](#_Toc372449613)

[Modulo ABM Notas: 7](#_Toc372449614)

[Modulo Búsqueda: 8](#_Toc372449615)

[Módulo de filtros: 8](#_Toc372449616)

[Bibliografía 9](#_Toc372449617)

[Anexo 1 10](#_Toc372449618)

[Diagrama de Casos de uso: 10](#_Toc372449619)

[Anexo 2 17](#_Toc372449625)

[Diagrama de clases: 17](#_Toc372449626)

[Anexo 3 18](#_Toc372449627)

[Diagrama E-R 18](#_Toc372449628)

[Anexo 4 19](#_Toc372449629)

[El entorno de desarrollo 19](#_Toc372449630)

[Anexo 5 20](#_Toc372449631)

[Explicación de los archivos utilizados. 20](#_Toc372449632)

[Conclusión 22](#_Toc372449633)

# Introducción

El objetivo de la aplicación es proporcionar al usuario una forma simple, agradable e intuitiva para administrar notas de usuarios.

Nuestra aplicación permite agregar, eliminar y modificar sus notas en la medida en que cada usuario lo desee. Como así también marcar las notas que fueron realizadas permitiendo una fácil visualización de las mismas.

La aplicación permite que el usuario se sienta dueño de la misma, ya que puede configurar el Look&Feel según una serie de opciones que proporciona la aplicación.

Facilita al usuario la gestión de las mismas con un buscador, para que escribiendo alguna palabra o parte de una palabra pueda encontrar la nota que necesita, de una forma fácil, rápida e intuitiva.

# Alcances y limites

La aplicación gestiona las anotaciones que los usuarios ingresan, permite su modificación, eliminación, y marcado como realizada. Además, permite una búsqueda por título o por el cuerpo de la nota.

Adicionalmente la aplicación permite el cambio de claves en caso de superar un periodo de vencimiento de la contraseña.

También permite la activación de la cuenta vía mail, permitiendo que al registrar un nuevo usuario, el sistema le envíe un email con un link para poder activar la cuenta.

# Ámbito de la aplicación:

Esta aplicación está pensada para el uso cotidiano de las personas de cualquier edad, con conocimientos mínimos de computación. Al estar en un servidor web, se pensó la aplicación para que pueda ser accedida desde cualquier dispositivo con acceso a internet (con los requerimientos de software necesarios). La aplicación fue creada para que sea lo más intuitiva posible, sin interacciones engorrosas, o botones que pueden sacar el foco a lo que es necesario.

# Funcionamiento

## Módulo Login:

Este modulo permite el ingreso de usuarios a aplicación para que pueda comenzar administrar sus notas.

Los datos que se deben rellenar en este formulario son el usuario y la contraseña. Si es usuario se encuentra en condición (su cuenta se ha activado correctamente), se le permitirá ingresar a la aplicación. Las contraseñas tienen una validez de 30 días (modificable) y luego del paso de este periodo, el usuario debe ingresar una clave diferente.

## Modulo Registro:

Este modulo se encuentra disponible para que los usuarios que deseen usar la aplicación puedan crear una cuenta en la misma y de esta forma poder administrar sus notas. (Este registro es obligatorio).

Para poder registrarse es necesario rellenar un formulario en el cual se pide el nombre, clave y email. Este email es requerido porque se le va a enviar un correo electrónico a esa cuenta de email para poder activar su cuenta.

Este correo tiene una validez de 24 hs, de forma que si el usuario no activo su cuenta durante ese periodo, tiene abrir el mail y presionar sobre la tecla de activación, lo que desencadena que se le envíe un nuevo email para validar su cuenta, el cual también tiene una validez de 24 hs.

## Modulo ABM Notas:

Este es el modulo principal de la aplicación, en el se permite la adición, modificación, eliminación, marcado como realizado de notas.

Para agregar notas, en la pantalla se muestra un botón “Agregar Nota”, la cual nos crea una nueva nota, la misma posee un titulo y un cuerpo que pueden ser modificados.

Para modificar una nota, solo basta con escribir en el titulo o en el cuerpo y cuando se le quita el foco del lugar de escritura, la nota se modifica.

Para poder eliminar las notas, la aplicación brinda un botón (X) en cada nota para poder eliminarla, solo basta con apretarlo y la nota se elimina.

Para marcar como realizado una nota, lo que se tiene que hacer es presionar sobre un checkbox que posee cada nota. Esta nota marcada puede ser desmarcada para mostrar que no se ha realizado aún.

## Modulo Búsqueda:

Para poder realizar una búsqueda, solo basta con escribir la palabra o parte de las palabras que se desea encontrar y la aplicación va a mostrar solo las notas que coincidan con los parámetros de búsqueda.

Para las búsquedas es necesario que se ingresen al menos 2 letras, de la palabra que se está buscando.

Una vez hecha la búsqueda, el usuario puede ingresar notas, modificarlas, eliminarlas, etc. de la misma forma que como si no estuvieran en una búsqueda.

## Módulo de filtros:

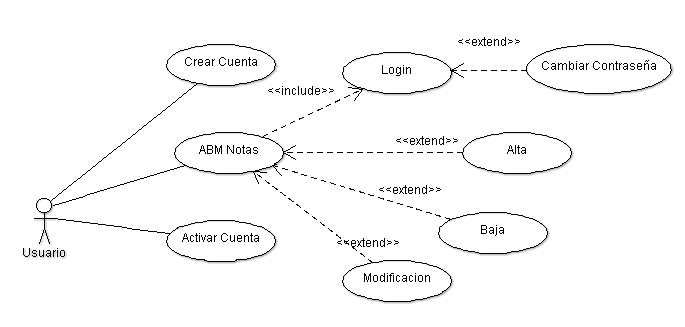
En este módulo, la aplicación va a mostrar 3 botones en la parte superior de la ventana de notas, uno de los cuales es para que muestre todas las notas, otro para que muestre solo las que faltan de realizar y por ultimo uno que muestre las que están listas (las cuales están marcadas con la tilde o checkboxs)

# Bibliografía

https://forja.rediris.es/docman/view.php/282/.../manualDesarrollador.pdf‎

# Anexo 1

## Diagrama de Casos de uso:



Granularidades Finas:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. Nivel del Use Case**:  Negocio  Sistema de Información** | | | | |
| **Nombre del Use Case**: **Activar Cuenta** | | | | **Nro. de Orden**: 1 |
| **Prioridad**:  Alta  Media  Baja | | | | |
| **Complejidad**:  Alta  Media  Baja | | | | |
| **Actor Principal**: Usuario | | **Actor Secundario**: No aplica | | |
| **Tipo de Use Case**:  Concreto  Abstracto | | | | |
| **Objetivo**: Activar la cuenta del usuario (U) para futuros accesos de la misma | | | | |
| **Precondiciones**: El U debe haber creado una cuenta. | | | | |
| **Post- Condiciones** | Éxito: El sistema (S) redirige al Login con un cartel de éxito | | | |
| Fracaso: El sistema (S) redirige al Login con un cartel de volver a controlar el mail. | | | |
| 1. **Curso Normal** | | | 1. **Alternativas** | |
| 1. El use case comienza cuando el U abre el mail enviado previamente por el S, y realiza click en Activar. | | |  | |
| 1. El Sistema (S) controla que la fecha de activación no haya expirado | | | 2.A.1. La fecha de activación expiró.  2.A.2. El S envía un mail al U de activación  2.A.3. El S redirige al Login con un cartel: “La fecha de activación ha expirado, por favor revise su email”  2.A.4. fin use case | |
| 1. El S se redirige al Login con un cartel: “Su cuenta se ha activado Correctamente” | | |  | |
| 1. Fin use case | | |  | |
| **Asociaciones de Extensión: no aplica** | | | | |
| **Asociaciones de Inclusión: no aplica** | | | | |
| **Use Case donde se incluye: no aplica** | | | | |
| **Use Case al que extiende: no aplica** | | | | |
| 1. **Use Case de Generalización: no aplica** | | | | |
| 1. **Autor: Antonel, Bonansea** | | | 1. **FUM: 12-11-2013** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. Nivel del Use Case**:  Negocio  Sistema de Información** | | | | |
| **Nombre del Use Case**: **Alta de notas** | | | | **Nro. de Orden**: 1 |
| **Prioridad**:  Alta  Media  Baja | | | | |
| **Complejidad**:  Alta  Media  Baja | | | | |
| **Actor Principal**: Usuario | | **Actor Secundario**: No aplica | | |
| **Tipo de Use Case**:  Concreto  Abstracto | | | | |
| **Objetivo**: El usuario (U) crea una nueva nota | | | | |
| **Precondiciones**: El U debe estar logueado en el sistema | | | | |
| **Post- Condiciones** | Éxito: El sistema (S) actualiza la pagina con una nueva nota debajo | | | |
| Fracaso: No aplica. | | | |
| 1. **Curso Normal** | | | 1. **Alternativas** | |
| 1. El use case comienza cuando el U realiza click | | |  | |
| 1. El Sistema (S) controla que la fecha de activación no haya expirado | | | 2.A.1. La fecha de activación expiró.  2.A.2. El S envía un mail al U de activación  2.A.3. El S redirige al Login con un cartel: “La fecha de activación ha expirado, por favor revise su email”  2.A.4. fin use case | |
| 1. El S se redirige al Login con un cartel: “Su cuenta se ha activado Correctamente” | | |  | |
| 1. Fin use case | | |  | |
| **Asociaciones de Extensión: no aplica** | | | | |
| **Asociaciones de Inclusión: no aplica** | | | | |
| **Use Case donde se incluye: no aplica** | | | | |
| **Use Case al que extiende: no aplica** | | | | |
| 1. **Use Case de Generalización: no aplica** | | | | |
| 1. **Autor: Antonel, Bonansea** | | | 1. **FUM: 12-11-2013** | |

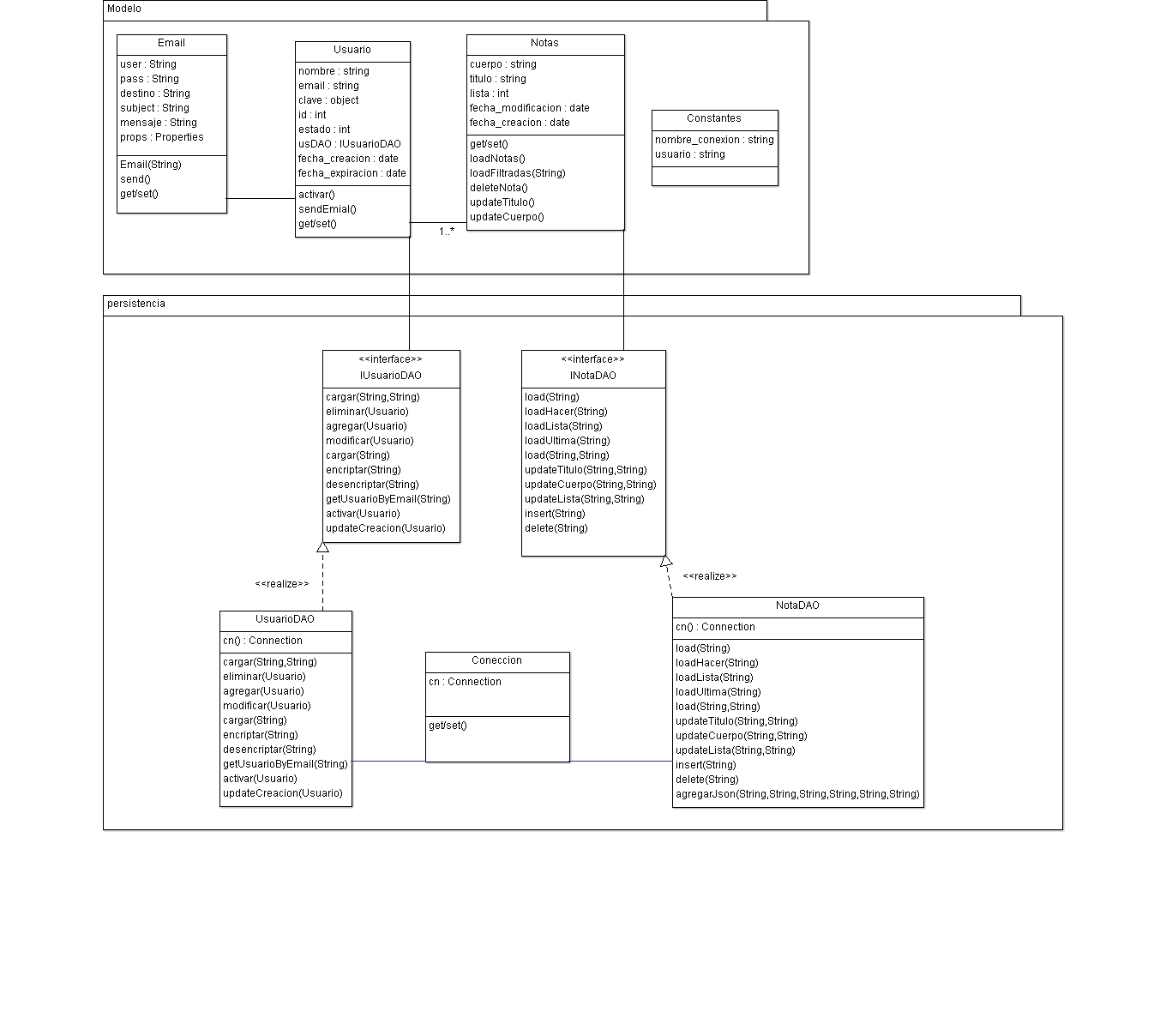
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. Nivel del Use Case**:  Negocio  Sistema de Información** | | | | |
| **Nombre del Use Case**: **Crear Cuenta** | | | | **Nro. de Orden**: 1 |
| **Prioridad**:  Alta  Media  Baja | | | | |
| **Complejidad**:  Alta  Media  Baja | | | | |
| **Actor Principal**: Usuario | | **Actor Secundario**: No aplica | | |
| **Tipo de Use Case**:  Concreto  Abstracto | | | | |
| **Objetivo**: Que el Usuario pueda crear una cuenta en el sistema | | | | |
| **Precondiciones**: No aplica. | | | | |
| **Post- Condiciones** | Éxito: El Usuario recibe un mail de activación | | | |
| Fracaso: No aplica. | | | |
| 1. **Curso Normal** | | | 1. **Alternativas** | |
| 1. El use case comienza cuando el Usuario (U) accede al sistema | | |  | |
| 1. El Sistema (S) despliega una ventana con espacio para ingresar el usuario y la contraseña, seguido de un botón Login y otro Registrarse. | | |  | |
| 1. El U realiza click en Crear una Cuenta | | |  | |
| 1. El S despliega una ventana dentro de la actual con espacios para ingresar nombre, clave dos veces e email. | | |  | |
| 1. El S controla que ambas claves coincidan. | | | 5.1.A - Las claves no coinciden, el S limpia la pantalla y muestra un cartel: “Claves ingresadas no coinciden”  5.2.A – Fin Use Case | |
| 1. El S envía un email al U para que active la cuenta. | | |  | |
| 1. Fin del Use Case. | | |  | |
| **Asociaciones de Extensión: no aplica** | | | | |
| **Asociaciones de Inclusión: no aplica** | | | | |
| **Use Case donde se incluye: no aplica** | | | | |
| **Use Case al que extiende: no aplica** | | | | |
| 1. **Use Case de Generalización: no aplica** | | | | |
| 1. **Autor: Antonel, Bonansea** | | | 1. **FUM: 14-11-2013** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. Nivel del Use Case**:  Negocio  Sistema de Información** | | | | |
| **Nombre del Use Case**: **Crear Cuenta** | | | | **Nro. de Orden**: 1 |
| **Prioridad**:  Alta  Media  Baja | | | | |
| **Complejidad**:  Alta  Media  Baja | | | | |
| **Actor Principal**: Usuario | | **Actor Secundario**: No aplica | | |
| **Tipo de Use Case**:  Concreto  Abstracto | | | | |
| **Objetivo**: Que el Usuario pueda crear una cuenta en el sistema | | | | |
| **Precondiciones**: No aplica. | | | | |
| **Post- Condiciones** | Éxito: El Usuario recibe un mail de activación | | | |
| Fracaso: No aplica. | | | |
| 1. **Curso Normal** | | | 1. **Alternativas** | |
| 1. El use case comienza cuando el Usuario (U) accede al sistema | | |  | |
| 1. El Sistema (S) despliega una ventana con espacio para ingresar el usuario y la contraseña, seguido de un botón Login y otro Registrarse. | | |  | |
| 1. El U realiza click en Crear una Cuenta | | |  | |
| 1. El S despliega una ventana dentro de la actual con espacios para ingresar nombre, clave dos veces e email. | | |  | |
| 1. El S controla que ambas claves coincidan. | | | 5.1.A - Las claves no coinciden, el S limpia la pantalla y muestra un cartel: “Claves ingresadas no coinciden”  5.2.A – Fin Use Case | |
| 1. El S envía un email al U para que active la cuenta. | | |  | |
| 1. Fin del Use Case. | | |  | |
| **Asociaciones de Extensión: no aplica** | | | | |
| **Asociaciones de Inclusión: no aplica** | | | | |
| **Use Case donde se incluye: no aplica** | | | | |
| **Use Case al que extiende: no aplica** | | | | |
| 1. **Use Case de Generalización: no aplica** | | | | |
| 1. **Autor: Antonel, Bonansea** | | | 1. **FUM: 14-11-2013** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. Nivel del Use Case**:  Negocio  Sistema de Información** | | | | |
| **Nombre del Use Case**: **Login** | | | | **Nro. de Orden**: 1 |
| **Prioridad**:  Alta  Media  Baja | | | | |
| **Complejidad**:  Alta  Media  Baja | | | | |
| **Actor Principal**: Usuario | | **Actor Secundario**: No aplica | | |
| **Tipo de Use Case**:  Concreto  Abstracto | | | | |
| **Objetivo**: Que el Usuario pueda autentificarse para hacer uso del sistema. | | | | |
| **Precondiciones**: No aplica. | | | | |
| **Post- Condiciones** | Éxito: El Usuario accede al menú principal del sistema. | | | |
| Fracaso: El Sistema retorna al login y muestra un cartel denegando el acceso. | | | |
| 1. **Curso Normal** | | | 1. **Alternativas** | |
| 1. El use case comienza cuando el Usuario (U) accede al sistema | | |  | |
| 1. El Sistema (S) despliega una ventana con espacio para ingresar el usuario y la contraseña, seguido de un botón Login y otro Registrarse. | | |  | |
| 1. El U ingresa su usuario y contraseña. | | |  | |
| 1. El U realiza click en Ingresar | | |  | |
| 1. El S controla que el usuario y contraseña ingresadas en el paso anterior coincidan con los existentes en la base de datos. | | | 5.1.A - El usuario no existe, ó no coincide con la contraseña.  5.2.B - El S recarga la pagina Login con un cartel diciendo: “Usuario o contraseña incorrecto”  5.3.C - Fin de Use Case | |
| 1. El S controla que la fecha de expiración no sea mayor a la fecha actual. | | | 6.1.A - La fecha de expiración es mayor a la actual.  Se invoca al Use Case “Cambiar Contraseña” | |
| 1. El S se redirige al menú principal. | | |  | |
| 1. Fin del Use Case. | | |  | |
| **Asociaciones de Extensión: no aplica** | | | | |
| **Asociaciones de Inclusión: no aplica** | | | | |
| **Use Case donde se incluye: no aplica** | | | | |
| **Use Case al que extiende: no aplica** | | | | |
| 1. **Use Case de Generalización: no aplica** | | | | |
| 1. **Autor: Antonel, Bonansea** | | | 1. **FUM: 14-11-2013** | |

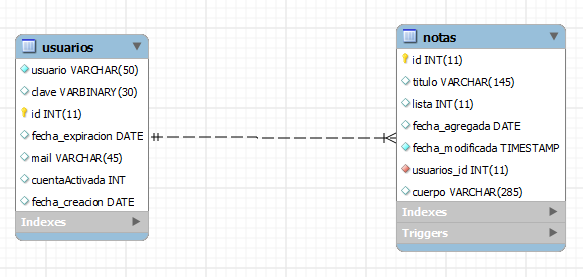
# Anexo 2

## Diagrama de clases:



# Anexo 3

## Diagrama E-R



# Anexo 4

## El entorno de desarrollo

* Eclipse

Para llevar a cabo el proyecto hemos usado el IDE de desarrollo Eclipse en su versión 4.3.

* Sistema operativo

Windows 8.1 y Windows 7.

* Apache Tomcat

Tomcat es un proyecto desarrollado por la fundación Apache. Es un contenedor de servlets que puede funcionar como servidor web autónomo pero que habitualmente se integra con el propio servidor Apache. Tiene una licencia libre de Apache. La versión utilizada para el desarrollo ha sido la 7.x

* Librería java-mail

Esta librería es la que nos permite el envio de un email cuando el usuario del sistema se quiere registrar por primera vez. En esta librería tenemos que poner el asunto, el mail del destinatario, el mail del que envía, la contraseña del que envía (clave proporcionada por google), etc.

* Boostrap

Bootstrap es un framework que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. Ha sido desarrollado por Twitter que recientemente liberó su versión 2.0. La mayor ventaja es que podemos crear interfaces que se adapten a los distintos navegadores (responsive design) apoyándonos en un framework potente con numerosos componentes webs que nos ahorrarán mucho esfuerzo y tiempo.

* Jquery

jQuery es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. jQuery es software libre y de código abierto, jQuery, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

# Anexo 5

## Explicación de los archivos utilizados.

* context.xml

La carpeta metainf contiene el fichero context.xml. En este fichero es en el que realmente especificaremos los parámetros de conexión a una base de datos determinada. Deberá existir una entrada <Resource> por cada base de datos que se vaya a emplear y su nombre deberá coincidir con el indicado en el fichero web.xml.

Las propiedades de configuración de <resource> son las siguientes:

name: nombre JNDI. Debe ser el mismo nombre que el indicado en web.xml.

auth: igual que <resauth> en el fichero web.xml.

type: igual que <restype> en el fichero web.xml.

username: usuario para acceder a la base de datos.

password: su contraseña.

driverClassName: nombre completo de la clase con el driver JDBC

url: url de conexión a la base de datos

maxActive: número máximo de instancias activas simultáneas para esta base de datos.

maxIdle: número máximo de conexiones simultáneas sin actividad.

maxWait: tiempo en milisegundos que esperará el sistema antes de generar una excepción cuando se le solicita una nueva conexión y no haya ninguna disponible.

validationQuery: Consulta SQL que puede usarse para comprobar la validez de una conexión. Debe ser un SELECT y devolver al menos una fila.

Es imprescindible por tanto que la base de datos esté correctamente configurada y que disponga de un usuario “root” cuya clave sea “root”.

* servlets

Un servlet es un objeto que se ejecuta en un servidor o contenedor JEE, fue especialmente diseñado para ofrecer contenido dinámico desde un servidor web, generalmente en HTML.

Un servlet es un objeto Java que implementa la interfaz javax.servlet.Servlet o hereda alguna de las clases más convenientes para un protocolo específico (ej: javax.servlet.HttpServlet). Al implementar esta interfaz el servlet es capaz de interpretar los objetos de tipo HttpServletRequest y HttpServletResponse quienes contienen la información de la página que invocó al servlet.

Cada servlet desempeña una función diferente y por tanto el código es sustancialmente distinto al contrario que ocurre con las clases generales y las clases \*DAO, las cuales tienen todas una estructura parecida.

* Libraries

Esta carpeta contiene las librerias externas del proyecto, necesarias para que funcione correctamente como son la JDK , las librerías propias del contenedor de servlets y servidor web Apache Tomcat.

* DAO

Este paquete contiene las clases que nos permiten manipular las tablas de la base de datos.

En ellas se encuentran los métodos básicos necesarios para trabajar con dichas tablas, como son

inserción, borrado, consulta y actualización.

Únicamente así se consigue una paquetización completa de la estructura del proyecto.

* Sistema de gestión de bases de datos:

Se decidió utilizar MySQL Workbench.  
Es una aplicación para el diseño y documentación de bases de datos pensada para ser usada con el sistema de gestión de bases de datos MySQL.  
MySQL Workbench es uno de los primeros productos de la familia MySQL que ofrece dos ediciones diferentes - una open source y una edición comercial.  
Esta es una de las principales razones por la cual se decide utilizar esta herramienta.

# Conclusión