

1. Descripción de la aplicación

Se pide de desarrollar una aplicación Web que permita crear proyectos y crear tareas dentro de cada proyecto. Cada tarea será asignada a un programador y tendrá una fecha estimada de finalización y una fecha real de finalización.

En una primera parte hay que hacer la lógica de la aplicación (Servlets desde donde se accede a una BD y los JSPs que se encargan de la presentación) y en una segunda parte habrá que dotar a la aplicación de autenticación y configurar la autorización en el acceso a cada funcionalidad.

Junto con este enunciado se proporciona:

- El fichero `proyectos.sql` con la sentencias SQL que permiten la creación de la base de datos y las tablas necesarias para la aplicación. Para ello habrá que ejecutar

```
mysql -u root -p
```

y una vez estemos en el cliente MySQL habrá que ejecutar:

```
source proyectos.sql
```

Con ello se crearán una serie de tablas y se insertarán valores en las tablas relacionadas con la autenticación y autorización. Las contraseñas se han almacenado codificadas usando el código que se muestra en las transparencias del Tema 4. Las contraseñas que corresponden a cada usuario son:

Usuario	Contraseña
Liskova	pass1
Ullmanova	pass2
Turinova	pass3
Newmanova	pass4

- Se proporciona también el fichero `TrabajoFinal.zip` que contiene un proyecto Eclipse en el que ya hay implementada cierta funcionalidad de la aplicación.

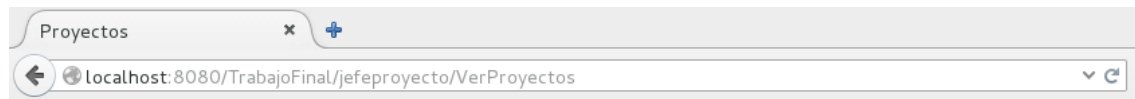
Las clases que se encuentran en el paquete `edu.uv.dawts.trabajofinal` son:

- `Proyecto.java` JavaBean que representa a un proyecto
- `Tarea.java` JavaBean que representa a una tarea
- `AccesoDatos.java` clase que representa el acceso a a la base de datos. Si se necesita alguna consulta más a la base de datos se deben añadir métodos a esta clase.
- `ContextListener.java` clase que representa un oyente de eventos que se producen en el contenedor al cargar la aplicación. En su código se crea un objeto del tipo `AccesoDatos` y se coloca en el contexto de la aplicación (*application scope* de modo que desde cualquier componente se puede obtener una referencia a ese objeto.
- `VerProyectos.java` Servlet mapeado a la URL `/jefeproyecto/VerProyectos` desde donde se obtiene una referencia a `AccesoDatos`, se obtienen todos los proyectos y se delega en un JSP la generación de la vista.
- `AddProyecto.java` Servlet mapeado a la URL `/jefeproyecto/AddProyecto` desde donde se obtiene una referencia a `AccesoDatos`, se añade el proyecto pasado como un parámetro, se obtienen todos los proyectos y se delega en un JSP la generación de la vista.

También se proporciona el JSP que muestra los proyectos y permite crear nuevos:

/WebContent/jefeproyecto/muestraProyectos.jsp

La siguiente captura muestra un ejemplo de la ejecución de la aplicación proporcionada:



Proyectos

Nombre proyecto	Acción
Desarrollo Trabajo Final	<input type="button" value="Ver tareas"/>
Monitoring Platform as a Service	<input type="button" value="Ver tareas"/>

Crear un nuevo proyecto

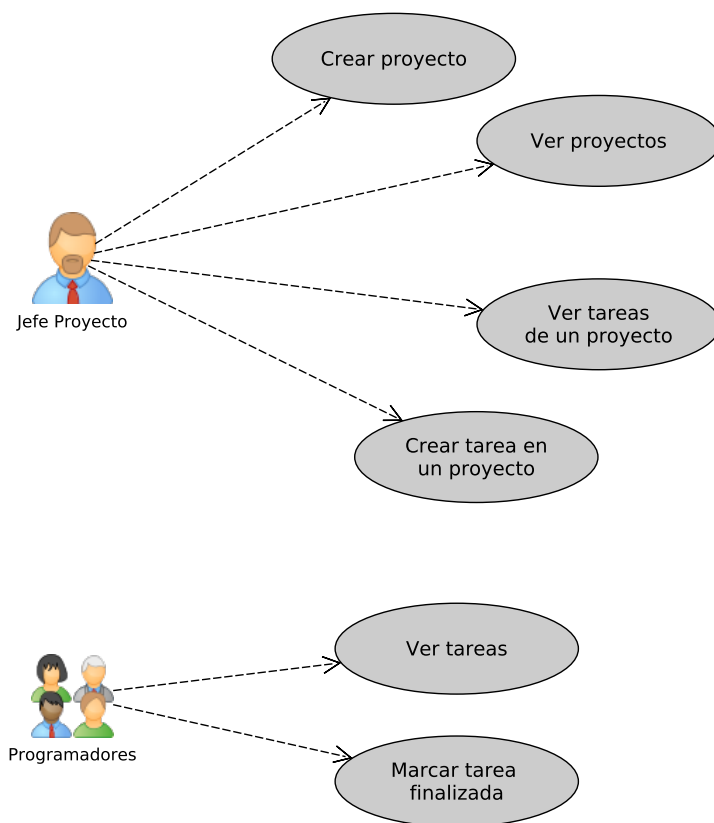
Nombre del proyecto:

2. Desarrollo de la lógica de la aplicación

Tareas:

1. Definir un pool de conexiones a la base de datos **proyectos**. El usuario y el password con el que hay que configurar el pool están en la última sentencia SQL del fichero **proyectos.sql**.
2. Definir un recurso JDBC llamado **jdbc/proyectos** para poder acceder al pool anterior.
3. Implementar los casos de uso que faltan.

CLASE



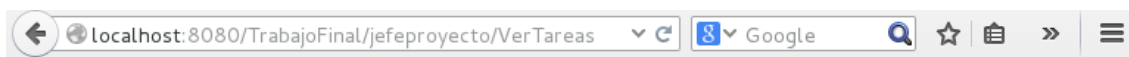
- **Jefe Proyecto:** crear proyecto, ver proyectos, ver las tareas asociadas a un proyecto y crear una nueva tarea asociada a un proyecto.
Los servlets relacionados con el Jefe de Proyecto se deben mapear a `/jefeproyecto/nombreservlet` y sus JSPs se deben guardar en `WebContent/jefeproyecto/nombre.jsp`
- **Programador:** ver las tareas (solo verá las tareas asignadas a él y si se ha sobrepasado la fecha de finalización se mostrará resaltada) y marcar una tarea como finalizada.
Los servlets relacionados con el Programador se deben mapear a `/programador/nombreservlet` y sus JSPs se deben guardar en `WebContent/programador/nombre.jsp`
Para esta primera parte se puede enviar un parámetro en las peticiones GET: `user_id` que permita poder distinguir entre los diferentes usuarios.

En todos los casos se debe usar el patrón Modelo-Vista-Controlador como en el código proporcionado.

Es posible que al seleccionar un proyecto se deba introducir en la sesión el id del proyecto y su nombre, de este modo se estará trabajando en un proyecto hasta que no se seleccione otro.

A continuación se muestra una captura con un ejemplo de cómo se pueden mostrar las tareas:

CLASE



Tareas asociadas al proyecto Desarrollo Trabajo Final

Nombre tarea	Asignada a	Fecha prevista finalización	Fecha finalizada
Escribir Servlet	2	2015-01-03	
Escribir JSP	3	2015-01-05	

Crear una nueva tarea

Nombre de la tarea:

Asignar tarea a:

Fecha prevista finalización:

Año (yyyy) Mes (mm) Día (dd)

La realización de las tareas descritas en esta sección tendrán un valor máximo de 8 puntos.

3. Autenticación y autorización

1. Definir un JDBC Realm a partir de la información que hay en la vista `v_user_role` (tal y como se describe en las transparencias del Tema 4). La codificación realizada del password en la base de datos es la que se describe en el Tema 4 y se ha obtenido realizando las siguientes transformaciones (en pseudocódigo):

`Base64(SHA-256(UTF-8(password)))`

Esta información se debe especificar al definir el JDBC Realm.

2. Definir la vinculación de roles a grupos (nombre del grupo en la base de datos) en el fichero `glassfish-web.xml`. En este caso habrá dos roles: `jefeproyecto` y `programador` que se corresponden a cada uno de los grupos que se han definido en la base de datos.
3. Definir de forma declarativa en el fichero `web.xml` la autenticación que debe realizarse mediante un formulario.
4. Definir la autorización con restricciones de seguridad en función de los roles (Jefe Proyecto o Programador). Esto se debe realizar de forma declarativa en el fichero `web.xml`.

A las URLs `/jefeproyecto/*` sólo pueden acceder los usuarios que pertenezcan al rol `jefeproyecto` y a las URLs `/programador/*` sólo pueden acceder los usuarios que pertenezcan al rol `programador`.

En este caso ya no será necesario pasar un parámetro que identifique al usuario ya que esta información la podemos obtener en el código del Servlet (y quizá se pueda colocar como un atributo de la sesión).

También es posible desarrollar un `logout` donde se borre toda la información de la sesión y relacionada con el login.

Si se realizan las tareas de la sección anterior y las de esta sección se podrá optar a una nota máxima de 10 puntos.

4. Entrega

Se debe subir a Aula Virtual un fichero zip que contenga el proyecto exportado desde Eclipse.

Entrega de la solución:

<i>January 2015</i>													
			1	2	3	4							
5	6	7	8	9	10	11							
12	13	14	15	16	17	18							
19	20	21	22	23	24	25							
26	27	28	29	30	31								

CLASE