

En este taller utilizaremos PostgreSQL, un DBMS de código abierto. Éste se encuentra instalado en Linux en la sala Ada Lovelace y tiene creado un usuario *lovelace* y una base de datos por defecto del mismo nombre. También utilizamos pgAdmin4, una aplicación web para administrar bases de datos en PostgreSQL. Utilizaremos también java, que se encuentra instalado en la misma sala en Linux. Para editar puede usar Eclipse u otro editor de su preferencia. También editaremos archivos HTML para lo cual puede usar cualquier editor de texto.

## 1. Iniciando con Formularios

1. Descargue el archivo de ejemplo de formulario HTML que encontrará en e-aulas.
2. Explore el archivo y asegúrese de entender todas las etiquetas.
3. Cree un nuevo archivo en el que incorpore un formulario que permita buscar estudiantes por nombre, apellido o número de documento.
4. Este nuevo formulario debe tener un botón que permita hacer la solicitud para buscar estudiantes.
5. Cree un nuevo archivo, esta vez con un formulario que permita buscar cursos.
6. Cada uno de los archivos anteriores debe llamar a una aplicación diferente en el servidor.

## 2. Iniciando con Servlets

1. Descargue el archivo de ejemplo de un servlet (.java) de e-aulas.
2. Explore el archivo y asegúrese de entender todas las instrucciones.
3. Cree un nuevo servlet que permita responder a una búsqueda de estudiantes por nombre, apellido o número de documento. Inicialmente responda con un HTML de prueba.
4. Cree un nuevo servlet que permita responder a una búsqueda de cursos. Inicialmente responda con un HTML de prueba.

## 3. Instalando y corriendo Apache Tomcat

1. Descargue la última versión de Apache Tomcat y descomprímala. A este directorio le llamaremos TOMCAT\_HOME. Seleccione el archivo con extensión .tar.gz para Linux. Para Windows seleccione el instalador adecuado.
2. Asegúrese de establecer la variable de entorno JAVA\_HOME, cuyo valor debe ser la carpeta donde reside su instalación de java. Por ejemplo, en Ubuntu puede ser `/usr/lib/jvm/java-11-openjdk-amd64`

3. Lo anterior puede hacerlo, desde una terminal, como

```
JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
```

4. Inicie su servidor Tomcat, ejecutando el archivo `startup.sh` que está ubicado en la carpeta `bin` del directorio `TOMCAT_HOME`.
5. Pruebe que su servidor funciona abriendo el navegador y buscando la página `http://localhost:8080`
6. Detenga su servidor Tomcat, ejecutando el archivo `shutdown.sh` que está ubicado en la carpeta `bin` del directorio `TOMCAT_HOME`.

## 4. Creando y ejecutando aplicaciones web

1. En la carpeta `TOMCAT_HOME` identifique la carpeta `webapps`, donde se almacenan las aplicaciones web que el servidor puede servir.
2. Dentro de la carpeta `TOMCAT_HOME` cree la siguiente estructura de directorios

```
test
test\WEB-INF
test\WEB-INF\classes
```

3. En la carpeta `TOMCAT_HOME\webapps\test` cree un archivo `index.html` con el siguiente código

```
<html>
  <head><title>Mi página</title></head>
  <body>
    <h1>Esta es mi pagina de bienvenida.</h1>
  </body>
</html>
```

4. En la carpeta `TOMCAT_HOME\webapps\test` cree otro archivo `otra.html` con el siguiente código

```
<html>
  <head><title>Mi otra página</title></head>
  <body>
    <h1>Esta es mi otra pagina de bienvenida.</h1>
  </body>
</html>
```

5. Compile el ejemplo de servlet descargado de e-aulas `ConsultaPersonaServlet.java`. Para esto necesita usar la librería `servlet-api.jar` que encuentra en la carpeta `lib` en `TOMCAT_HOME`. Puede hacerlo en su IDE o en terminal con los comandos

```
javac -cp $TOMCAT_HOME/lib/servlet-api.jar ConsultaPersonaServlet.  
java
```

Copie el archivo binario `ConsultaPersonaServlet.class` a la carpeta `TOMCAT_HOME\webapps\test\WEB-INF\classes`. Al copiar, incluya la carpeta contenedora del paquete si definió uno (e.g., en el ejemplo de e-aulas debe incluir la carpeta `testServlet`, que contiene el binario).

6. Verifique que el formulario en HTML que descargó de e-aulas dirija la consulta a la dirección apropiada, es decir,

```
<form action="http://localhost:8080/test/testServlet/  
consultaPersona" method=get>
```

7. Note que el archivo ejemplo de e-aulas es parte de un paquete de java `testServlet`. Esto hace que la anotación definida incluya la ruta con el paquete, y que al copiar los binarios a la carpeta `classes`, se incluya también la carpeta.
8. Realice el mismo procedimiento anterior para ejecutar los dos servlets que definió anteriormente. Pruebe los servlets con los formularios definidos.

## 5. Aplicaciones y bases de datos

1. Active el servidor de bases de datos PostgreSQL y la base de datos asociada al archivo `Taller4\_base.sql`. Si no tiene una BD creada, créela con este archivo.
2. Modifique su servlet de búsqueda de estudiantes para que se conecte a la base de datos y extraiga la lista de los cursos que ha tomado el estudiante. Retorne un HTML con esta información. Si el estudiante no se encuentra, debe retornar un mensaje.
3. Antes de reiniciar el servidor Tomcat, agregue la librería con el driver de PostgreSQL a la carpeta `TOMCAT_HOME/lib`. Ésta es la misma librería que usó para compilar el código de Java que se conecta a la base de datos PostgreSQL. Por ejemplo, en mi caso es `postgresql-42.2.5.jar`. Esta librería le permitirá a Tomcat comunicarse con el servidor de bases de datos de PostgreSQL.
4. Modifique su servlet de búsqueda de cursos para que se conecte a la base de datos y extraiga la lista de estudiantes que están tomando el curso. Retorne un HTML con esta información. Si el curso no se encuentra, debe retornar un mensaje.