

# Guía de Aplicación Cliente-Servidor usando EJB

creado por Carlos Eduardo Castelán Vázquez

adaptado por Marco Antonio Gutiérrez Cárdenas del tutorial de aplicación cliente - servidor con EJB

El siguiente tutorial se realizó utilizando macOS High Sierra, aunque los pasos de configuración y las herramientas que se utilizaron también se pueden utilizar en Windows.

- JDK 8 // *Tiene que ser la versión de Oracle. Es posible descargar esta versión sin tener que hacer una cuenta con Oracle.*
- Table Plus (opcional) // *Esta es un workbench para MySQL, pero solo la podemos usaremos para hacer queries sencillos*
- Homebrew (solamente para macOS, para instalar MySQL)
- Chocolatey (solamente para Windows, para instalar MySQL y/o JDK8)
- MySQL 8.0.18
- NetBeans 8.2 (descargar todos los bundles) /\* *Aquí supongo que será donde dice 'All'* \*/
  - GlassFish 4.1.1 (viene incluido como un bundle de NetBeans)

Se puede utilizar en Windows, no se recomienda llevar este tutorial en Ubuntu ya que la instalación del JDK 8 y Java 8 distribuida por Oracle tiene un mayor nivel de complejidad.

## Instalación

Antes de comenzar necesitamos tener instalado Java 8 y el JDK 8, si cuenta con alguna versión más nueva tendrá que desinstalar. Para revisar la versión de java con la que cuenta ejecute en la terminal los siguientes comandos: `java -version` y `javac -version`.

Si la versión que obtenemos al ejecutar los comandos es la 1.8.0\_231 podemos continuar, si no cuenta con esta versión o no tiene Java instalado utilice los links 1 y 2 para descargar e instalar estas versiones de java (si utiliza alguna distribución de linux no utilizar OpenJDK). Lo importante es que en los settings de macOS salga el icono de Java.



Después de instalar Java, instalaremos MySQL si cuenta con un sistema operativo diferente a macOS utilizar el link de descarga 5, para macOS utilizaremos el gestor de paquetes Homebrew. Para instalar Homebrew tenemos que seguir las instrucciones proporcionadas en el link 4, una vez instalado homebrew ejecutaremos el siguiente comando `brew install mysql`

Se recomienda ejecutar el siguiente comando para no tener que iniciar mysql cada vez que iniciamos el equipo `brew services start mysql`  
Si no, ejecuta el comando anterior ejecutar `brew mysql.server start`

Una vez iniciado MySQL tenemos que configurar una contraseña para nuestra cuenta así que ejecutaremos `mysql_secure_installation`

La contraseña elegida para root fue `ejb` (todo en minúsculas). VALIDATE PASSWORD COMPONENT solo sirve para que tus contraseñas sean mínimas de 8 caracteres, tengan signos de puntuación y esas cosas de seguridad. Para esta práctica no es necesario. Es posible volver a correr el `mysql_secure_installation` en caso de haberse equivocado (por ejemplo, para permitir usuarios anónimos) pero necesitarás recordar tu contraseña de root.

```
(base) SAM:~ mgtca$ mysql_secure_installation
Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: n
Please set the password for root here.

New password:
Sorry, you can't use an empty password here.

New password:
Re-enter new password:
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : n
... skipping.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : n
... skipping.
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

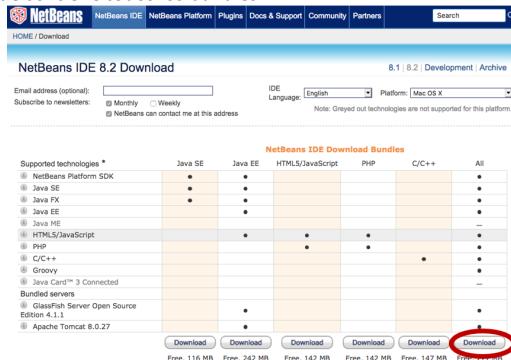
All done!
(base) SAM:~ mgtca$
```

Al tener configurado MySQL, crearemos una nueva base de datos para esto podemos correr el script que viene con esta guía llamado `schema.sql` entrando a `mysql` en el mismo directorio donde se encuentre este script y usando el comando `source schema.sql`.

El nombre de la base de datos será `Prueba_EJB`, a continuación, el código para crear la base de datos y la tabla usuario que utilizaremos en esta guía. Usuario está conformado por un id y un nombre. El script está hecho para poder correrse varias veces en caso de falla y volver a dejar todo como se necesita.

```
DROP DATABASE `Prueba_EJB`;
CREATE DATABASE `Prueba_EJB`;
USE `Prueba_EJB`;
CREATE TABLE `Prueba_EJB`.`Usuario`(`id` serial, `Nombre` varchar(255) NOT NULL, PRIMARY KEY (id));
```

Al tener la base de datos creada iniciaremos con nuestra aplicación de EJB, la cual estará conformada por una aplicación cliente que se conecta a una aplicación de Java EE que se encuentra en un servidor Glassfish. Para esto primero tenemos que instalar NetBeans haciendo uso del link 7, descargamos la opción que contiene todos los bundles.

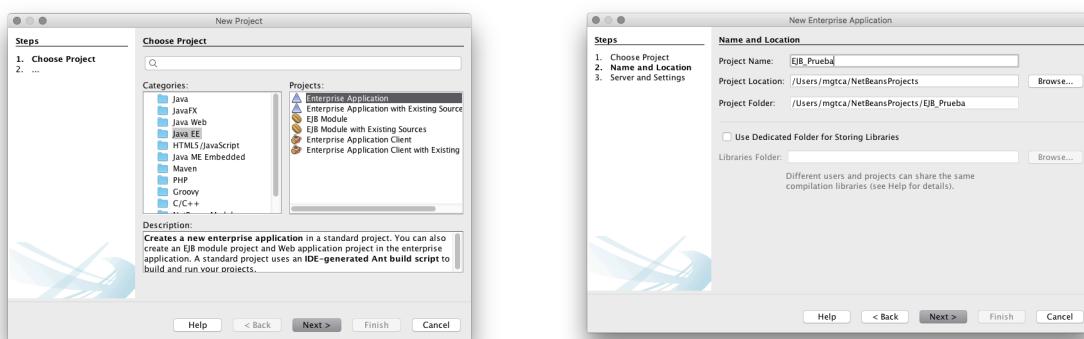


También debemos descargar de una vez el conector de MySQL con Java usando el link 6. Hay que seleccionar la opción *Platform Independent*, que es la que está destinada para Mac

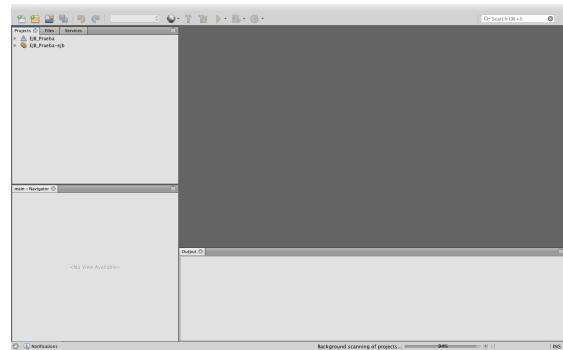
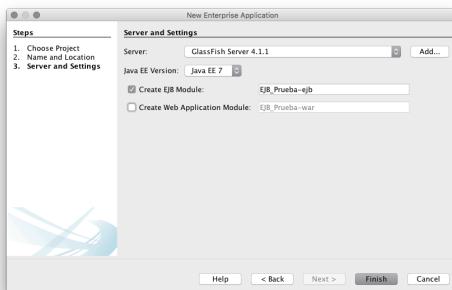


## Configuración y Desarrollo de la Aplicación

Para iniciar el desarrollo de nuestra aplicación abrimos nuestro IDE NetBeans, seleccionamos la opción de nuevo proyecto, en la ventana emergente seleccionamos la categoría Java EE y la opción de Enterprise Application. Al proyecto le llamaremos `EJB_Prueba`:

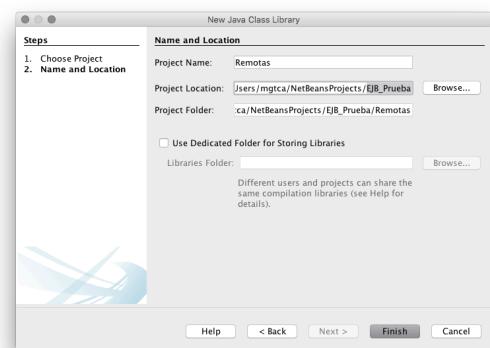
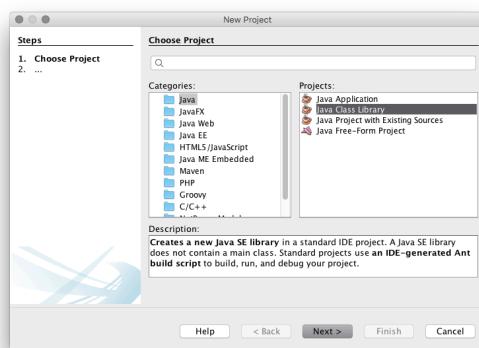


Por último, tendremos que seleccionar las siguientes opciones para crear nuestro proyecto. Una vez creado, tendremos en nuestra pestaña de proyectos los siguientes:

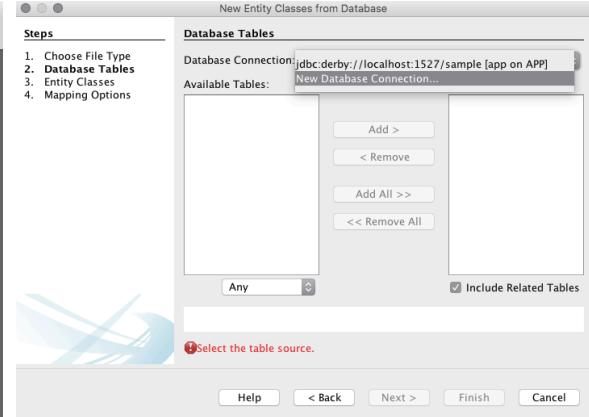
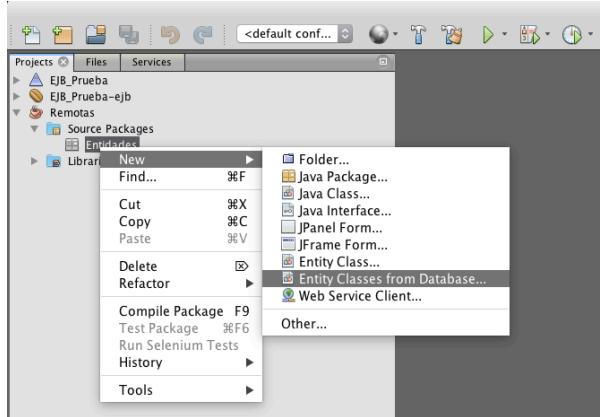


**Nota:** Si se desea realizar una web application module utilice como complemento el libro Java 13: Distributed programming and Java EE (link 8)

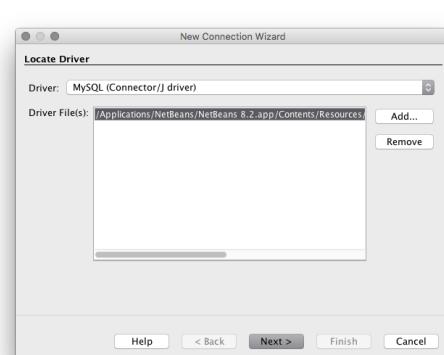
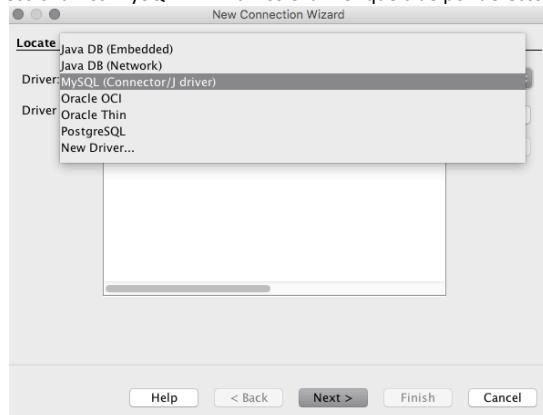
Seleccionamos crear nuevo proyecto. En esta ocasión seleccionamos la categoría Java y la opción Java Class Library. Guardaremos este proyecto en el folder de nuestro proyecto llamado EJB\_Prueba.



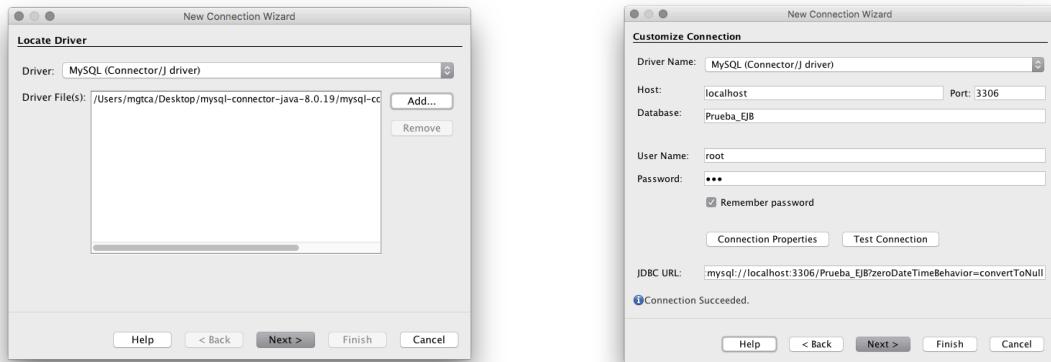
En esta librería creamos un paquete llamado Entidades, seleccionamos New y elegimos Entity Classes from Database. Seleccionamos New Database Connection.



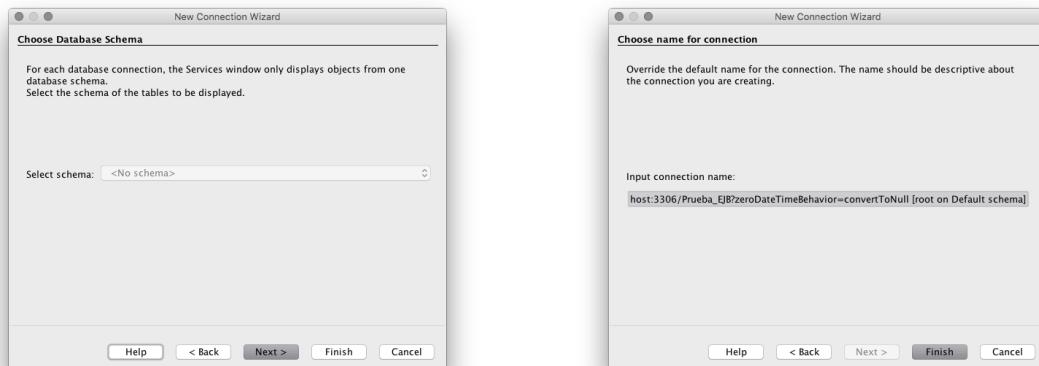
Seleccionamos MySQL. Eliminamos el driver que trae por defecto:



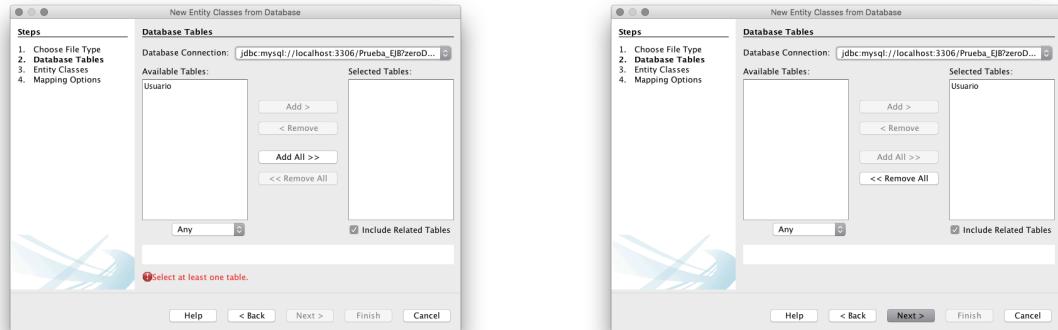
Y añadimos el que descargamos previamente. Una vez hecho esto configuramos la conexión y la probamos:



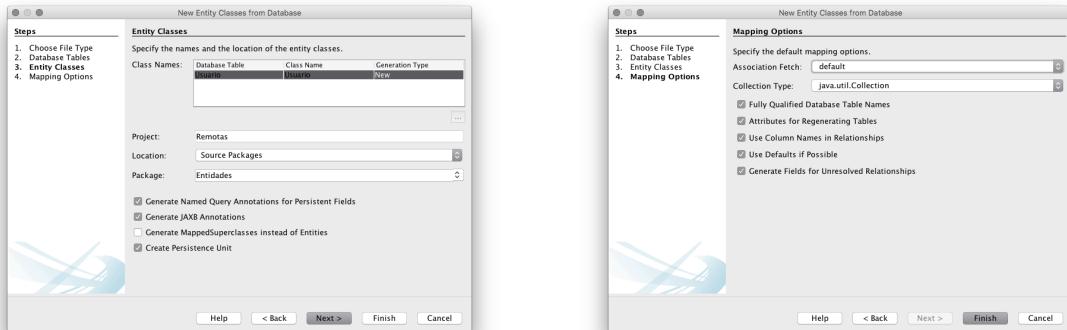
Damos dos veces next y finish:



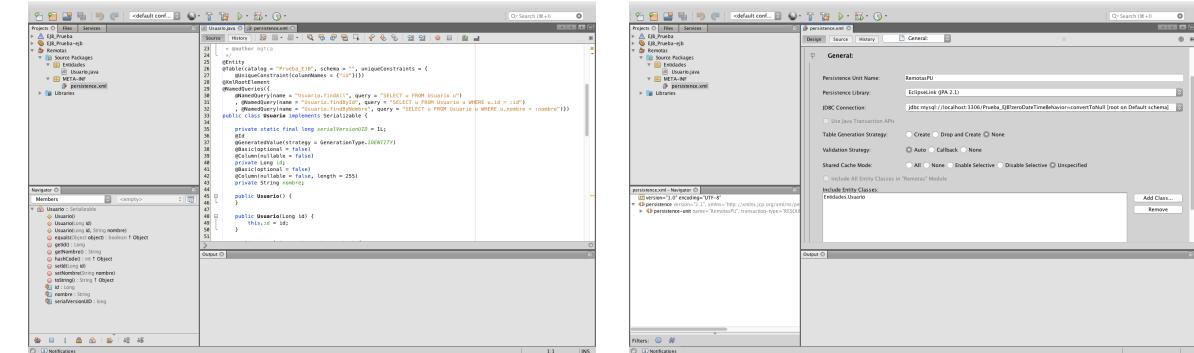
Seleccionamos la tabla usuario y la añadimos:



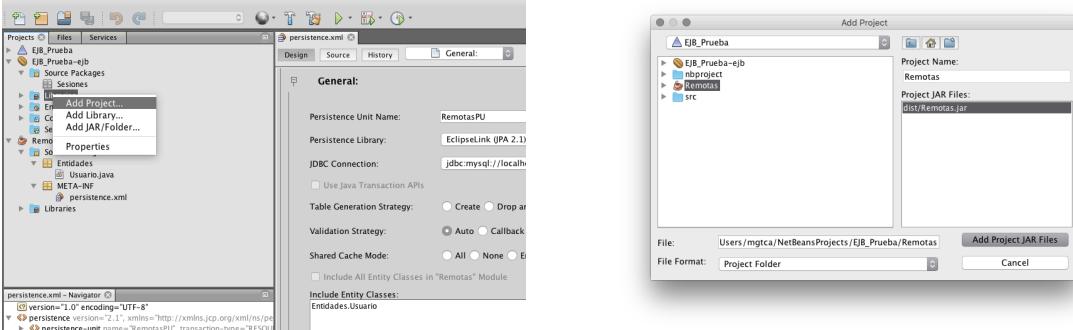
Creamos las Clases de Entidad. Configuramos las opciones de mapeo:



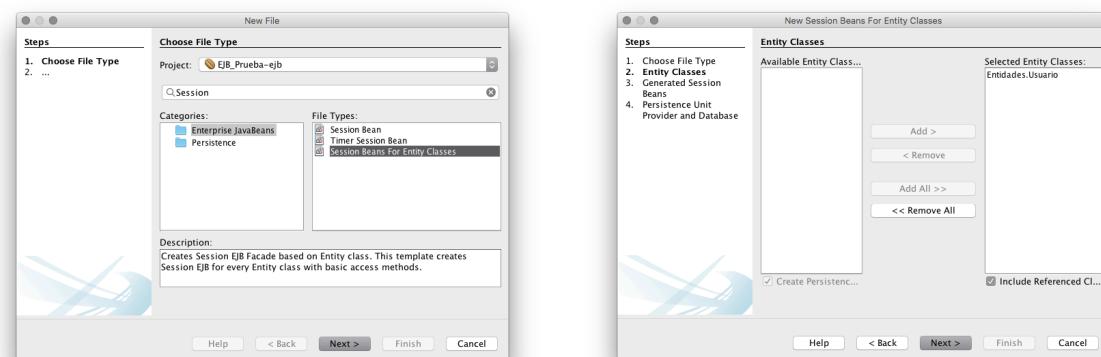
Como resultado tenemos la clase Usuario.java y el paquete META-INF con el archivo persistence.xml



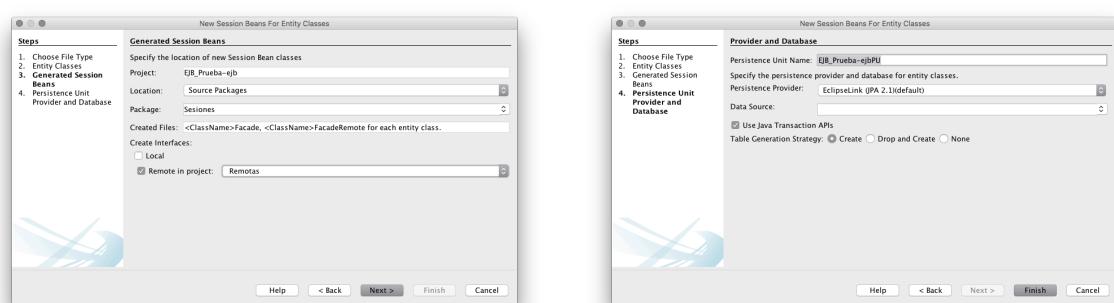
En el proyecto EJB\_Prueba-ejb creamos el paquete Sesiones y añadimos la librería Remotas a ese proyecto:



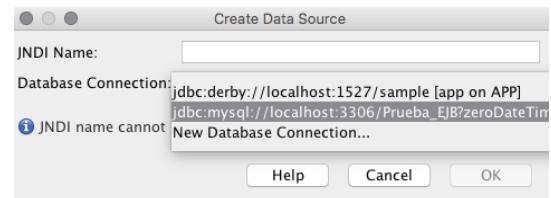
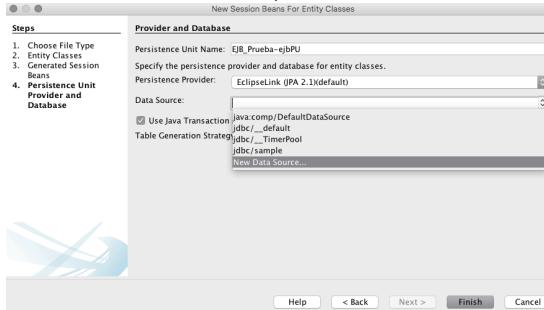
Ahora click derecho en el paquete se Sesiones > Session Bean for Entity Classes. Importante En caso de no encontrarlo, seleccionar en other y buscarlo. Seleccionamos las entity classes que utilizaremos, en este caso usuario.



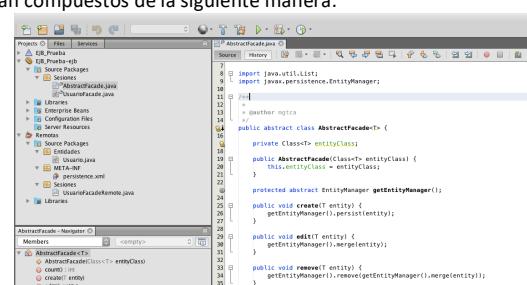
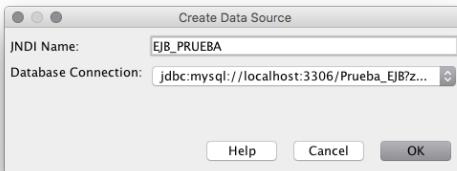
En create interfaces seleccionamos remote y se añadirá a la librería de Remotas. Le damos next y seleccionaremos un Data Source.



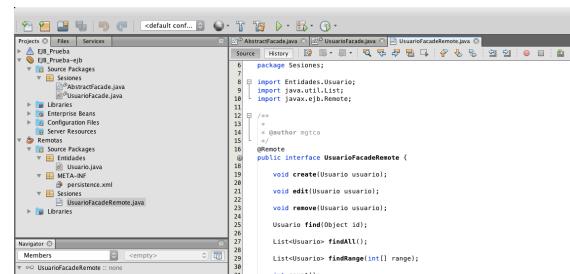
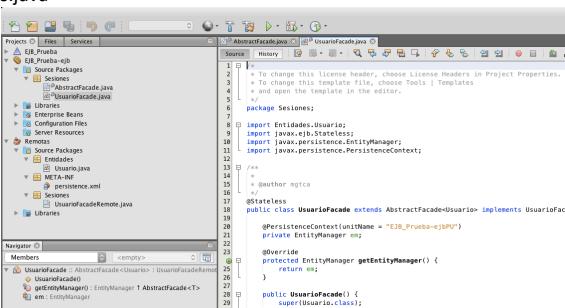
Generamos un nuevo *data source*, la conexión a la base de datos será la que creamos previamente.



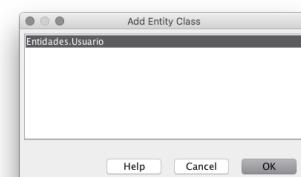
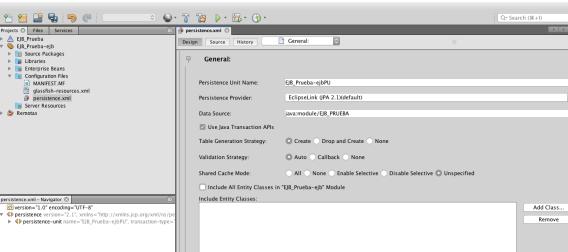
De nombre le pondremos EJB\_PRUEBA. Como resultado en nuestro paquete de sesiones de nuestro proyecto EJB\_Prueba-ejb tendremos dos archivos AbstractFacade.java y UsuarioFacade.java, los cuales están compuestos de la siguiente manera:



Abstract facade, cuenta con todos los métodos que tienen por defecto los session beans, en cuanto a UsuarioFacade se añaden los demás métodos que se requieran. Por último, en nuestra librería de Remotas se añade el paquete sesiones con el archivo UsuarioFacadeRemotas.java

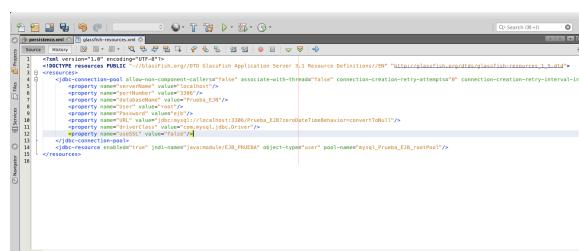
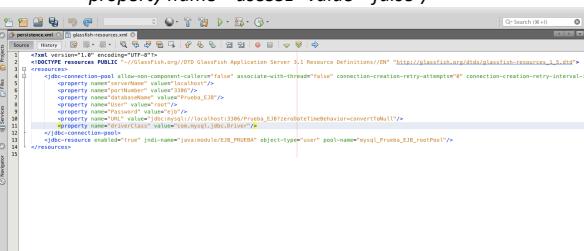


Una vez teniendo esto, en el proyecto EJB\_Prueba-ejb vamos a la carpeta Configuration Files y seleccionamos persistence.xml. Deseleccionamos Include All Entity Classes in "EJB\_Prueba-ejb Module y seleccionamos Add Class y añadimos Usuario si se tuvieran otras para añadir tambien se seleccionan:



Ahora en la misma carpeta seleccionamos glassfish-resources.xml. A este archivo le añadiremos lo siguiente:

```
<property name="useSSL" value="false"/>
```



Antes de proseguir configuraremos glassfish, el cual se ubica dentro de la la carpeta de netbeans, en macOS la ruta es:

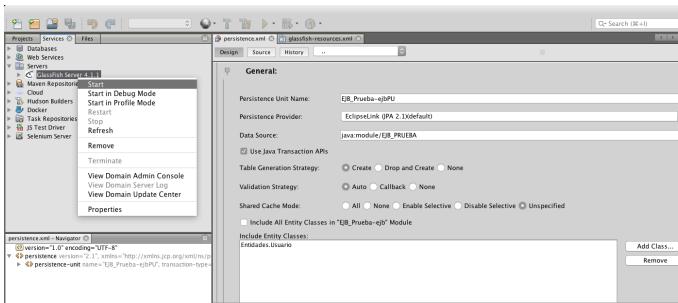
`/Applications/NetBeans/glassfish-4.1.1/glassfish/domains/domain1/config/domain.xml`

Tenemos que modificar el siguiente archivo. Añadiremos antes del tag de cierre </resources> el código del archivo `glassfish-resources.xml`. Ya modificado lo guardamos.

## Ahora en la carpeta

/Applications/NetBeans/glassfish-4.1.1/glassfish/domains/domain1/lib/ext

añadimos el conector de mysql. **Importante** poner el .jar y no el folder completo. Ya configurado lo anterior iniciamos glassfish para esto, regresamos a netbeans en la pestaña services seleccionamos *Servers* e iniciamos glassfish.



Ya iniciado abrimos el navegador y accedemos a localhost:4848 esperamos a que cargue y nos dirigimos a:

## *Resources -> JDBC -> JDBC Connection Pools*

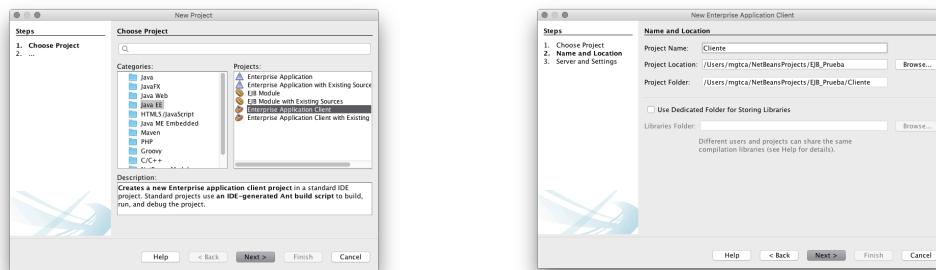
y seleccionamos `mysql_Prueba_EJB_rootPool`. En `Datasource Classname` cambiamos el nombre que aparece por el siguiente:(esto para la versión 8). Si la versión es 5 dejar la que está.

*com.mysql.cj.jdbc.MysqlDataSource*

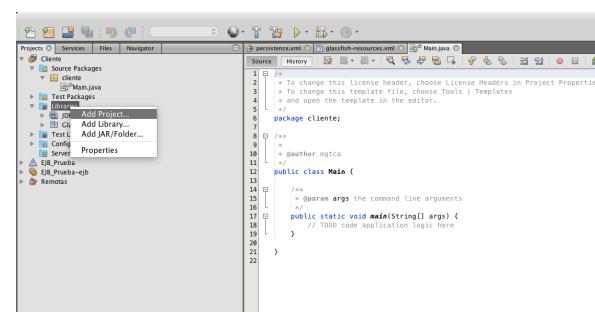
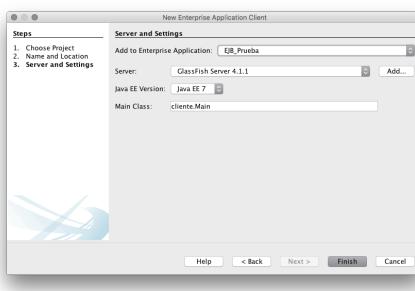
Al haber cambiado el nombre le damos save y luego ping, si el ping es exitoso podemos seguir con la aplicación en NetBeans. Se cambia el nombre de la clase puesto que en el driver que descargamos ya no se llama igual la clase.

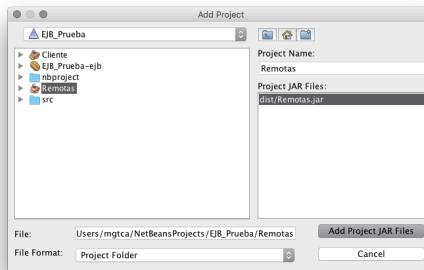


Estando en NetBeans crearemos un nuevo proyecto, la cual será una *Enterprise Application Client*. A esta aplicación se le llamará Cliente la cual se guardará en la carpeta principal de todo el proyecto

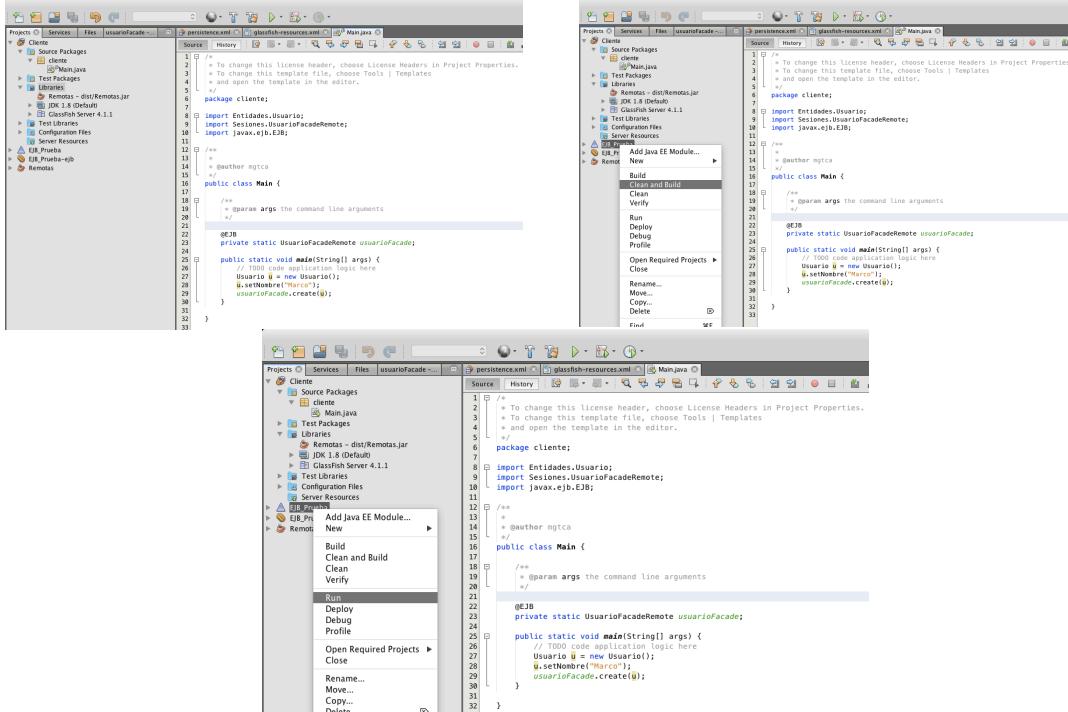


Añadimos esta aplicación a la aplicación empresarial EJB\_Prueba y le damos finish. Al crearse los archivos de este proyecto añadiremos la librería remota a nuestra aplicación cliente:





Al agregar nuestra librería modificamos la clase main de nuestra aplicación Cliente, de la siguiente manera. Para probar la aplicación seleccionamos el proyecto EJB\_Prueba, seleccionamos Clean and Build y después run.



Una vez ejecutado el programa podremos ver que funciona si en nuestra base de datos tenemos un nuevo registro con el nombre que utilizamos

```
mysql> select database();
+-----+
| database() |
+-----+
| prueba_ejb |
+-----+
1 row in set (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_prueba_ejb |
+-----+
| Usuario |
+-----+
1 row in set (0.00 sec)

mysql> select * from Usuario;
+----+----+
| id | Nombre |
+----+----+
| 1  | Marco  |
+----+----+
1 row in set (0.01 sec)

mysql>
```

## Links de Descarga

1. <https://www.java.com/es/download/>
2. <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
3. <https://tableplus.com>
4. [https://brew.sh/index\\_es](https://brew.sh/index_es)
5. <https://www.mysql.com/products/community/>
6. <https://dev.mysql.com/downloads/connector//>
7. <https://netbeans.org/downloads/8.2/>
8. <https://bookboon.com/es/java-13-distributed-programming-and-java-ebook>