

Manual de configuración de Hibernate



Suponemos que el usuario tiene instalados todos los componentes necesarios para la realización de la aplicación: Netbeans con los complementos de web e Hibernate, Java...

En primer lugar, debemos crear la base de datos que vamos a usar en nuestra aplicación.

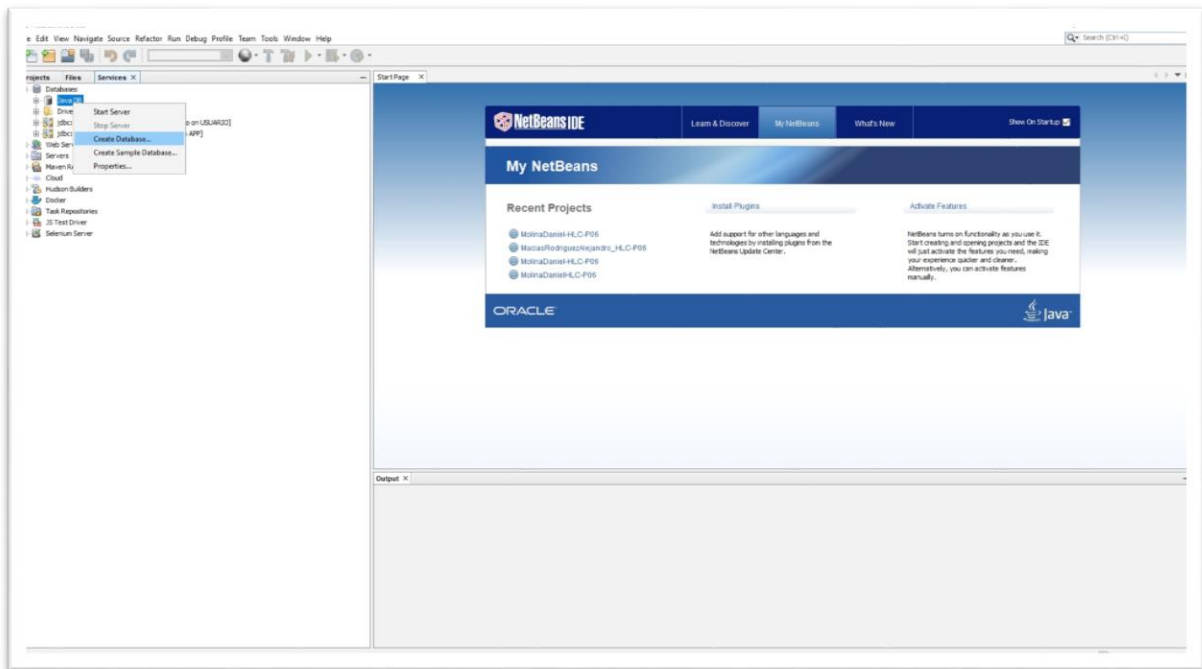


Imagen 1: creación de base de datos.

Creamos la base de datos, en nuestro caso con las siguientes credenciales:

- Database name: colegio.
- User name: usuario.
- Password: usuario.

Nos conectamos a la base de datos haciendo click derecho en ella, seleccionando "Connect" y poniendo las credenciales que hemos definido.

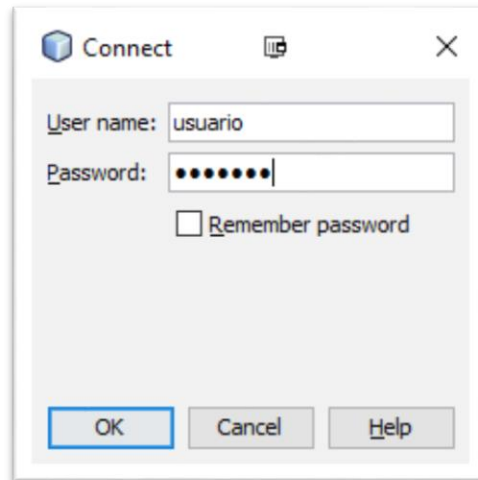


Imagen 2: acceso a la base de datos.

Una vez dentro de la base de datos, procedemos a crear las tablas y alimentarlas con datos, esto lo podemos hacer haciendo uso de varias opciones del IDE, en nuestro caso vamos a hacerlo manualmente. Para ello, expandimos la conexión en el árbol de servicios haciendo click en “+”, expandimos nuestro usuario, hacemos click derecho en “tablas” y seleccionamos “Execute Command”, y ahí podemos ejecutar todo el código SQL que queramos.

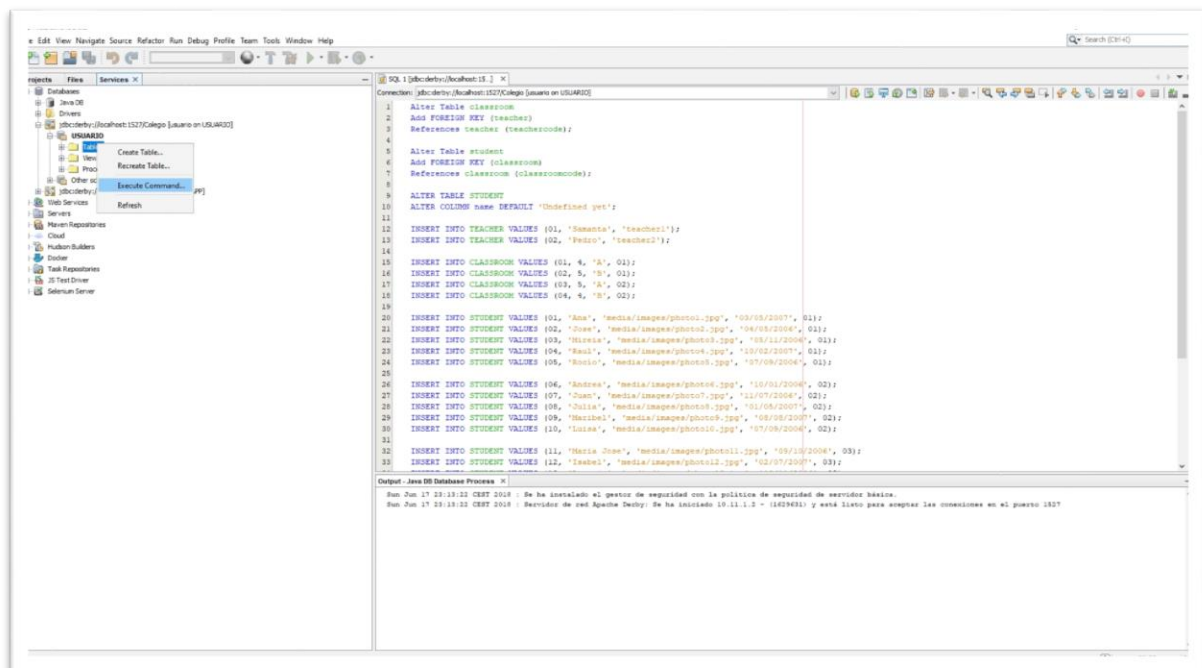


Imagen 3: creación de tablas.

Una vez creada nuestra base de datos, vamos a comenzar a hacer la aplicación con la que nos vamos a conectar a ella.

Elegimos Files->New project->Java Web->Web Application.

Pulsamos “Next”, rellenamos con los datos que correspondan, pulsamos “Next”, nos aseguramos que tenemos elegido como Server GlassFish Server, pulsamos “Next”, elegimos los frameworks que vamos a usar, en nuestro caso “Hibernate 4.3.1” y lo configuramos para que se conecte a nuestra base de datos.

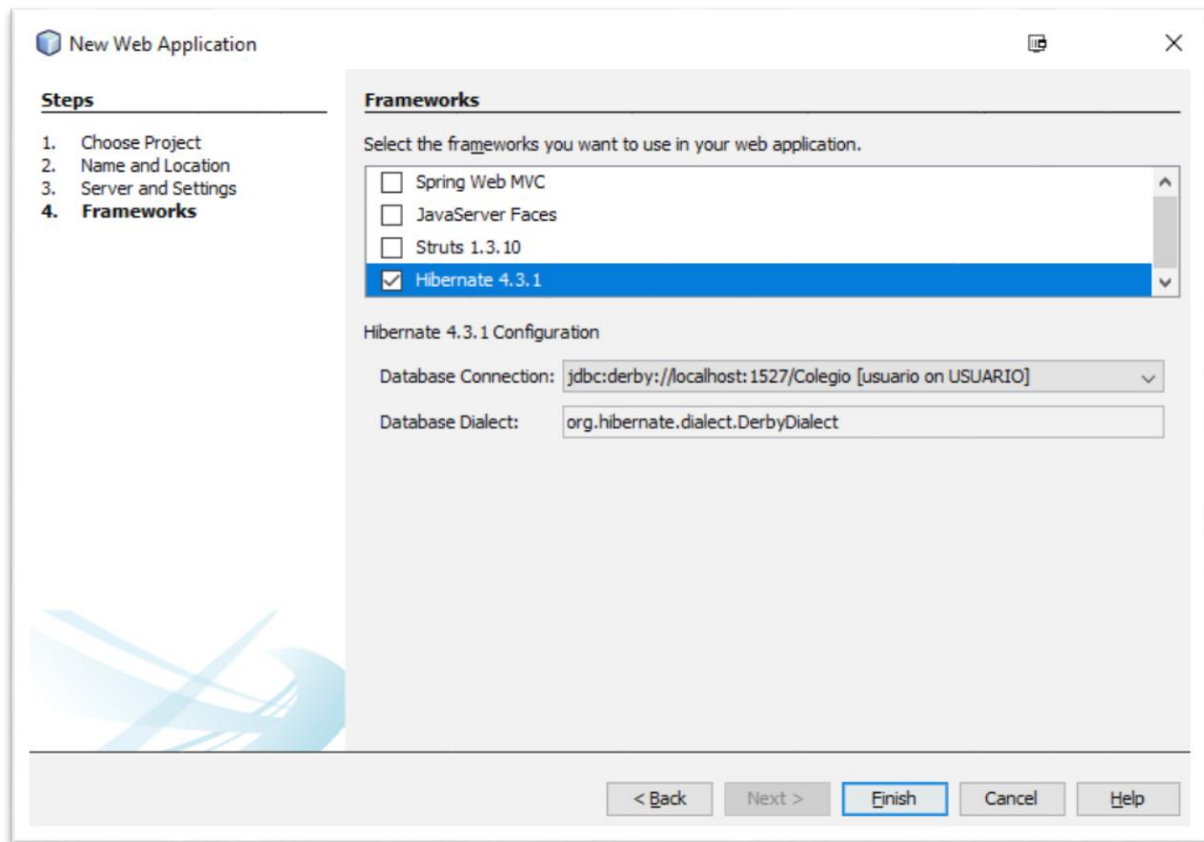


Imagen 4: adición de frameworks a nuestra aplicación.

En “Database Connection” elegimos la conexión que hemos creado anteriormente y le damos a “Finish”.

Procedemos a crear los paquetes donde van a ir nuestras clases, serán “Persistence”, “Model” y “Dao”, y moveremos el archivo “hibernate.cfg.xml”, generado automáticamente, a “Persistence”.

A continuación, vamos a editar el archivo anteriormente mencionado, le hacemos doble click, nos aseguramos de que en la parte de “JDBC Properties” aparecen los siguientes campos, si no, los introduciremos manualmente haciendo click en “Add...”

Más abajo, en Optional Properties/Configuration Properties, añadimos otros dos campos, y más abajo, en Miscellaneous Properties, otro, de manera que nos quede como en la siguiente imagen:

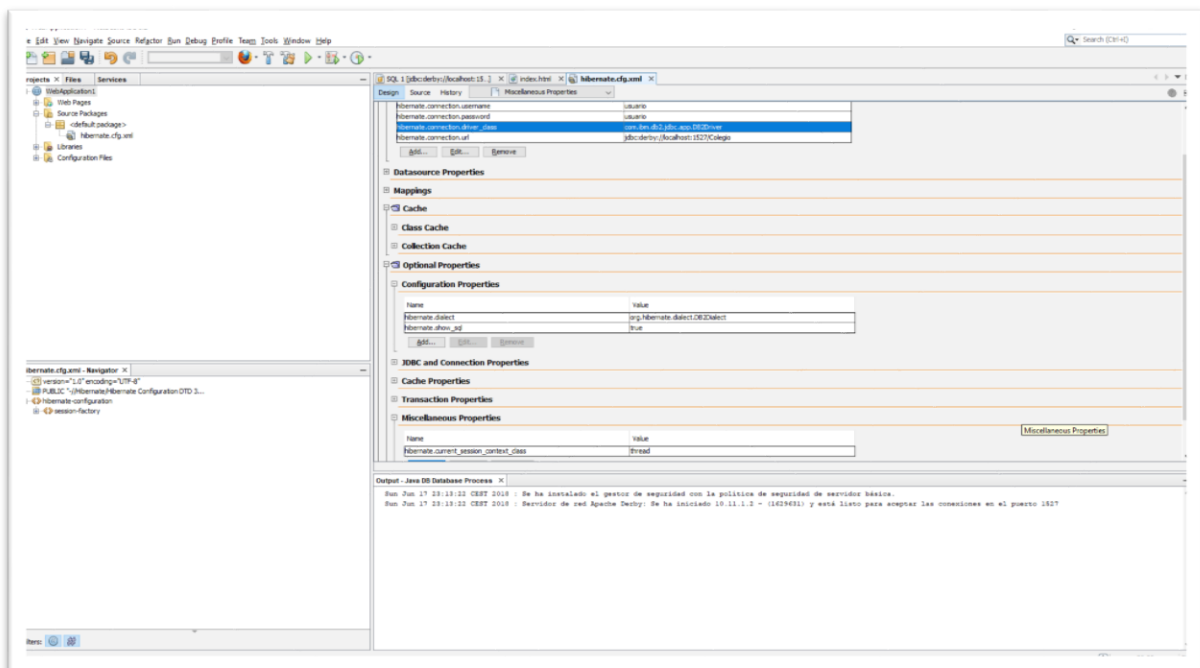


Imagen 5: editando el archivo hibernate.cfg.xml.

Una vez hecho esto, creamos los demás archivos necesarios para la configuración de nuestra aplicación:

Click derecho en el paquete “Persistence”->New->Other.

Buscamos HibernateUtil y lo creamos tal cual, haciendo click en “Finish” sin tocar nada.

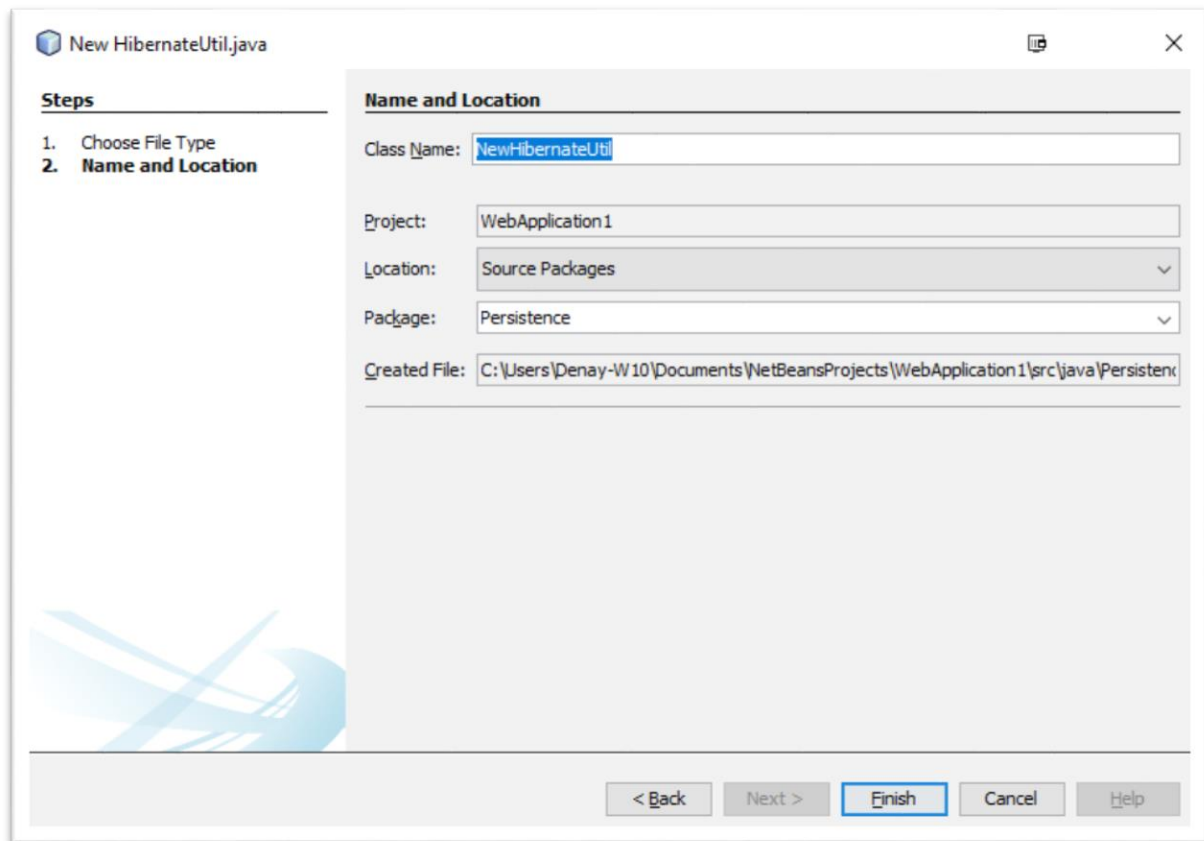


Imagen 6: creación del HibernateUtil.

Lo editamos sustituyendo la línea:

```
sessionFactory = new AnnotationConfiguration().configure().buildSessionFactory();
```

por lo siguiente:

```
sessionFactory = new  
AnnotationConfiguration().configure("Persistence/hibernate.cfg.xml").buildSession  
Factory();
```

Ahora creamos un Hibernate Reverse Engineering Wizard, también en el paquete “Persistence” y en el mismo sitio que encontramos el HibernateUtil.

Hacemos click en “Next”, luego en “Add All” y “Finish”.

Puede ser que nos salga el siguiente error:

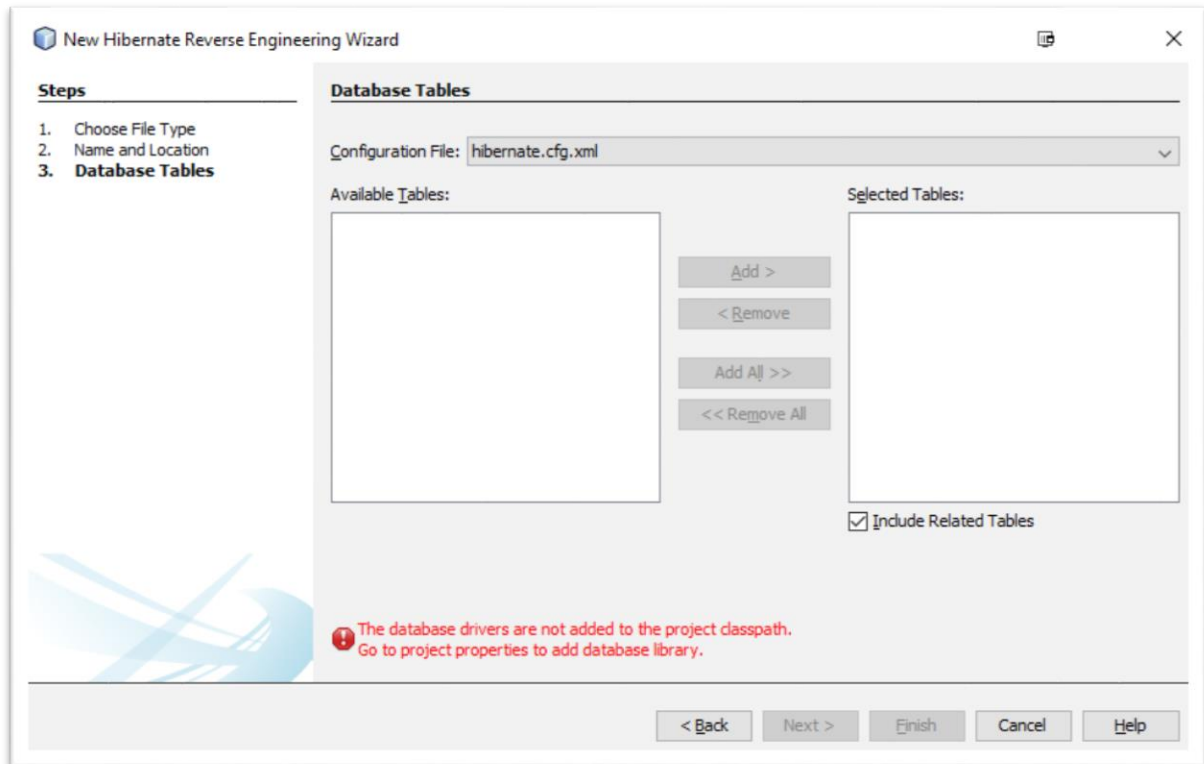


Imagen 8: Error buscando driver.

Esto se debe probablemente a que “hibernate.cfg.xml” no haya sido correctamente configurado, por lo que habría que buscar el error para seguir.

Vamos a crear el Hibernate Mapping Files and POJOs from Database, esta vez en el paquete “Model” que dejaremos tal cual, o seleccionando la opción “JDK 5 Language Features” si no lo estuviera. Hacemos click en “Finish”.

Ahora pasaremos a crear las clases Dao, que son las que nos servirán para interactuar con la base de datos, para ello crearemos una interfaz por cada tabla en la base de datos (o por cada clase en el paquete “Model”), y una clase para implementar cada interfaz.

Crearemos en cada clase las funciones que vayamos a usar en cada caso, en nuestro caso, por ejemplo:

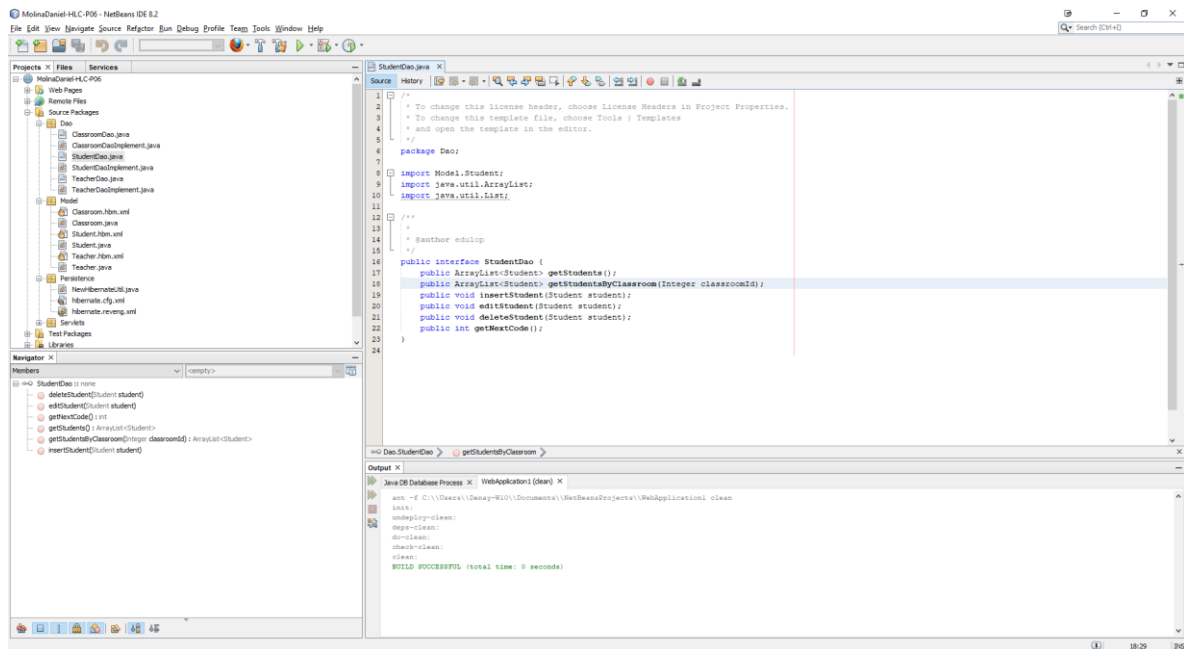


Imagen 9: Interfaz de la clase "Student"

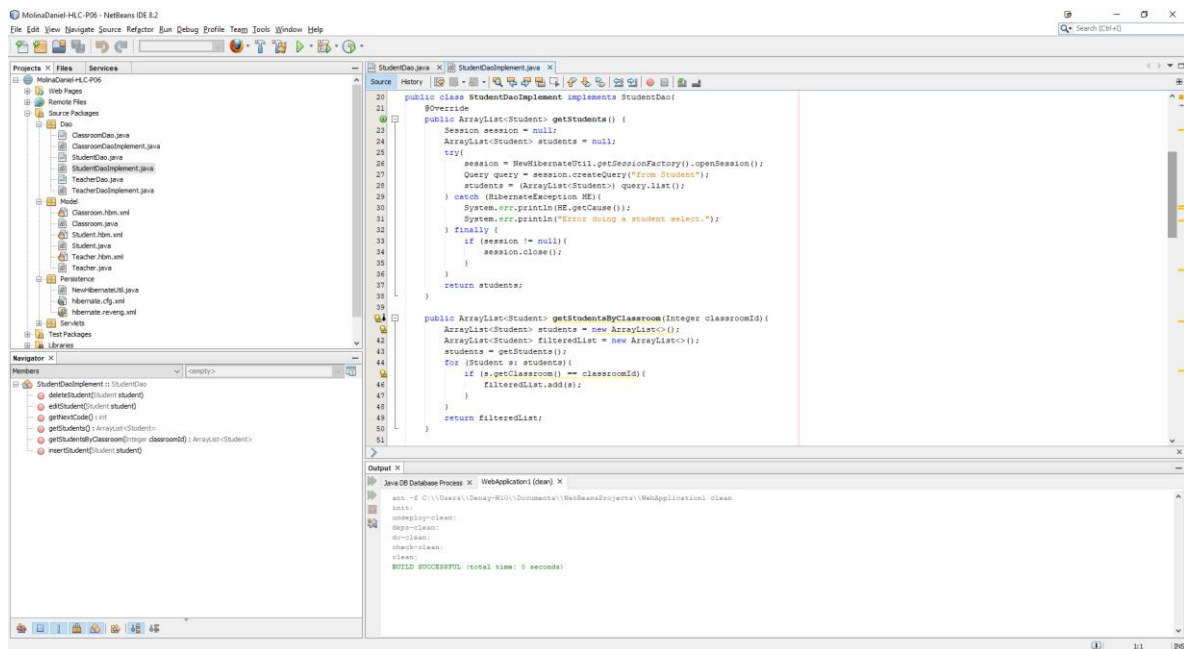


Imagen 10: Clase con interfaz implementada.

Y con esto habríamos finalizado nuestra configuración, solo nos faltaría implementarla en nuestra app web.