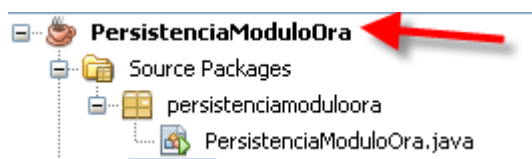


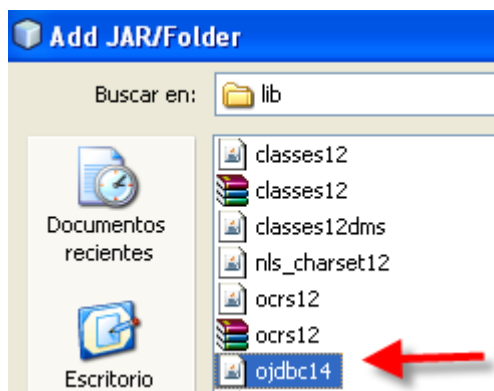
Práctica 5. Crear una Entidad persistente **con oracle.**

La práctica consiste en crear una entidad llamada módulo y persistirla en la base de datos como una tabla llamada modulo.

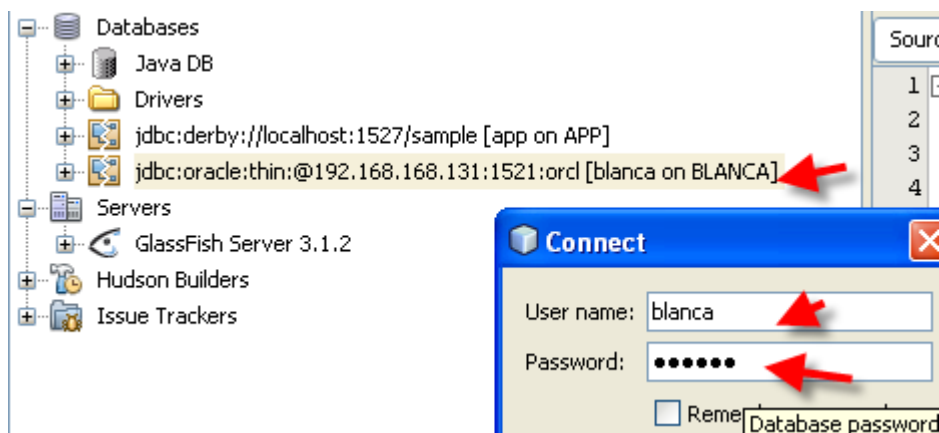
Crea un proyecto llamado PersistenciaModuloOra



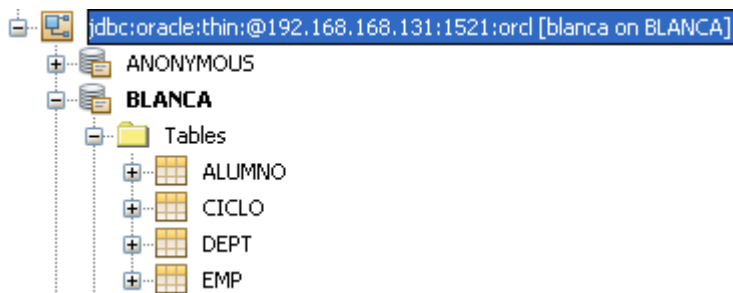
Añade el .jar de conexión a oracle



Conéctate como tu usuario oracle y su pw



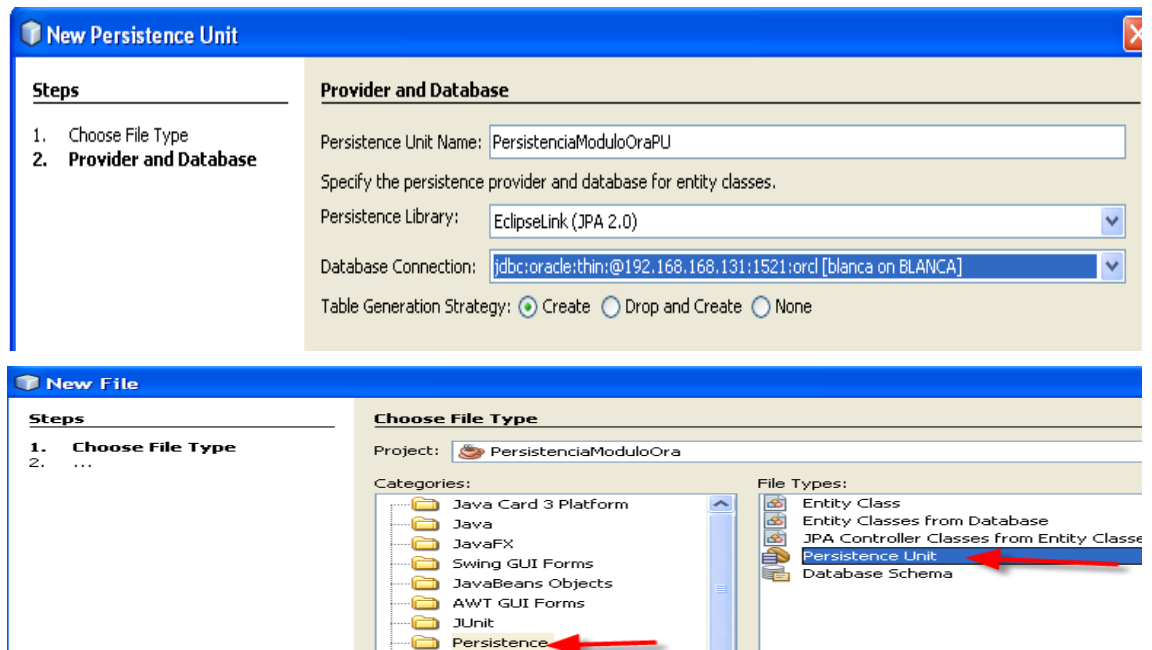
Veo las tablas que tengo antes de añadir Modulo



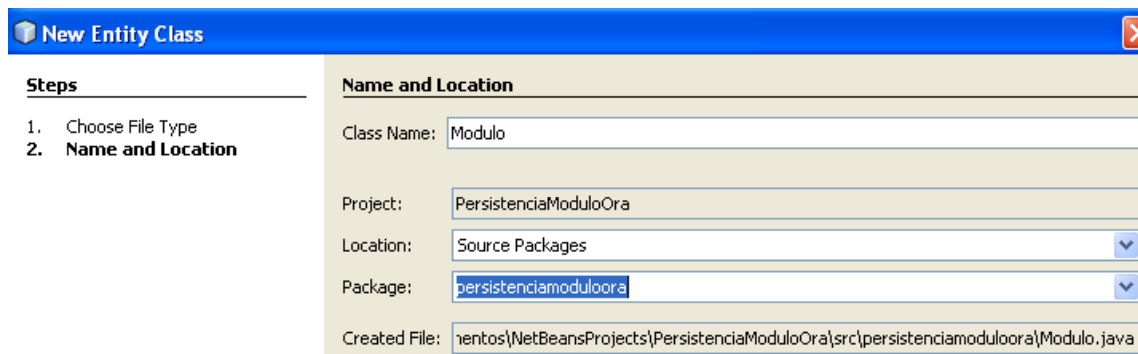
PASOS:

1. Crear unidad de persistencia
 2. Crear una entidad persistente
 3. crear un jpa Controller
 4. crear un clase aplicación
-

1. Crear unidad de persistencia



2. Crear una entidad persistente



Completo el código y añadido el atributo **private String nombre**

```
@Entity
public class Modulo implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    private String nombre;
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
}
```

y sus métodos get and set

```
public String getNombre() {
```

```
    return this.nombre;
```

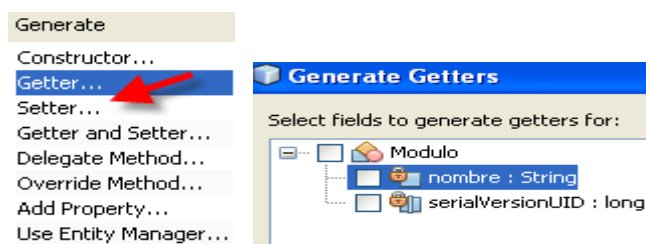
```
}
```

```
public void setNombre(final String nombre) {
```

```
    this.nombre = nombre;
```

```
}
```

Lo podemos escribir manualmente o bien sólo crear el atributo nombre y con el botón derecho ratón seleccionar insert code □ getter and setter



```

@Entity
public class Modulo implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    private String nombre;

    public Long getId() {
        return id;
    }

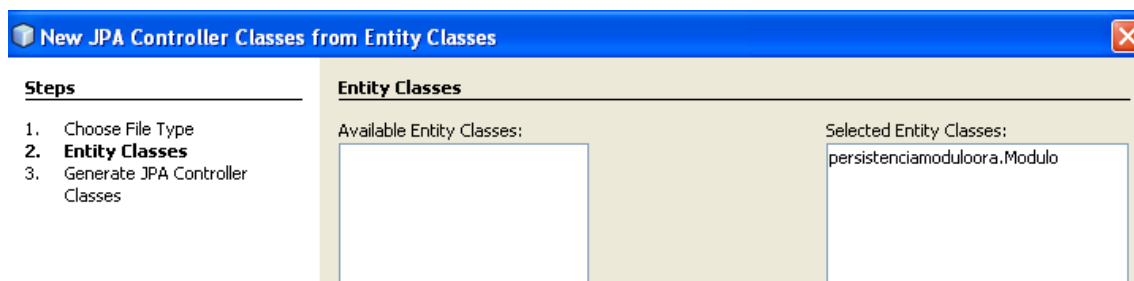
    public void setId(Long id) {
        this.id = id;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }
}

```

3. crear un jpa Controller



No olvidar añadir al fichero persistence.xml las clases

<class>Modulo.java</class>

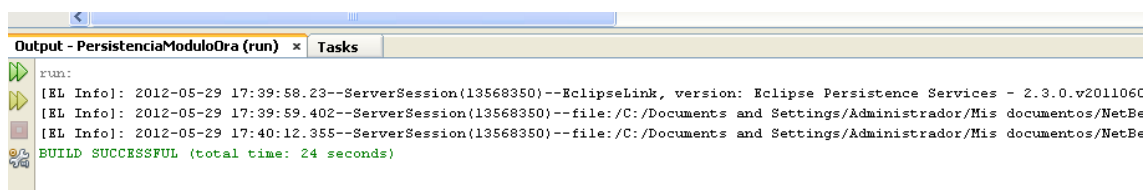
4. crear un clase aplicación

Persistir la entidad modulo en una tabla, e insertar un modulo llamado bases de datos.

Probar que se ha creado la tabla, consultando en SqlPlus de oracle y desde neatbean

Nota:

A veces para que funcione hay que ejecutar la aplicación, desconectarse de oracle, volverse a conectar y volver a ejecutar la clase aplicación



SQL> desc modulo

Nombre	¿Nulo?	Tipo
ID	NOT NULL	NUMBER(19)
NOMBRE		VARCHAR2(255)

SQL> select * from modulo;

ID
1

NOMBRE

BASES DE DATOS

The screenshot shows an IDE interface with a database connection established. The left pane displays the database structure, including a table named 'MODULO' with columns 'ID' and 'NOMBRE'. The right pane shows the SQL query 'select * from modulo;' and its result set, which contains one row with ID '1' and NOMBRE 'BASES DE DATOS'. Red arrows highlight the query and the result row.

Source History Connection: jdbc:oracle:thin:@192.168.168.131:1521:orcl

1 select * from modulo

select * from modulo

Page Size: 20

#	ID	NOMBRE
1	1	BASES DE DATOS