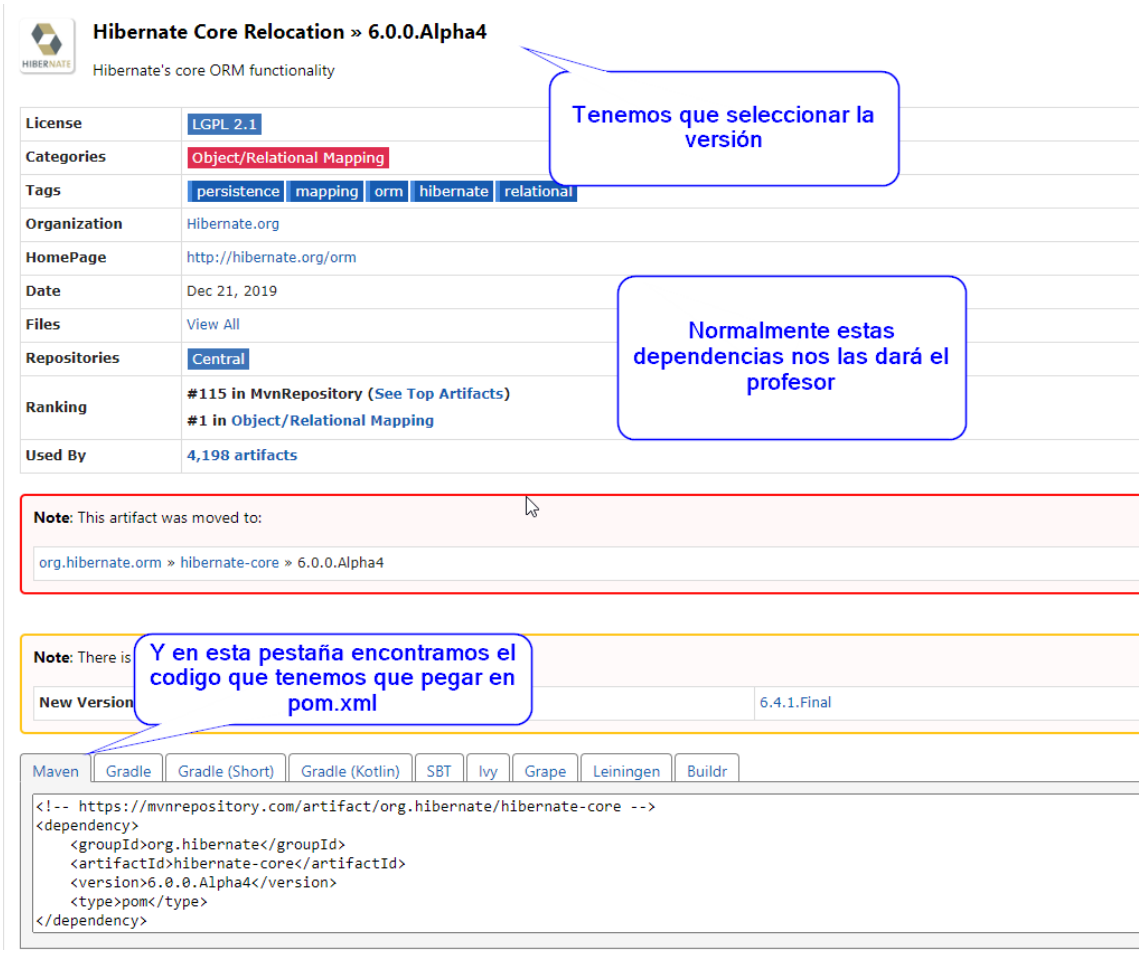
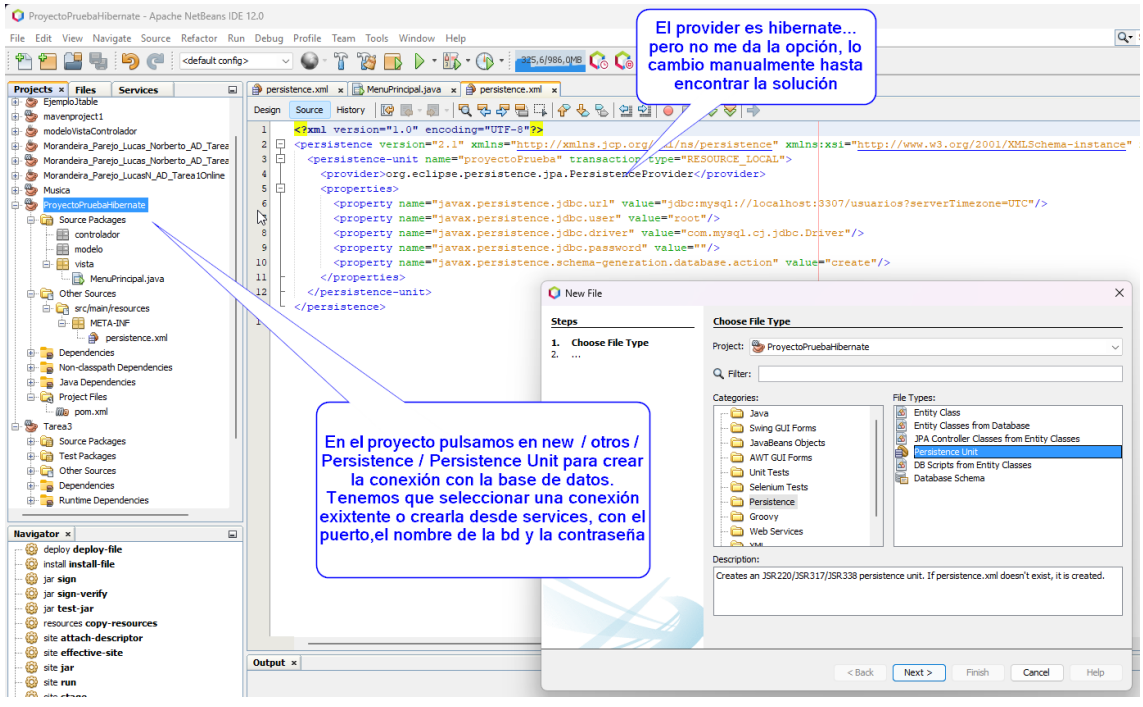
Consultas HQL Hibernate

Cuando creamos un proyecto maven nos crea en Proyect Files el archivo pom.xml donde tenemos que indicar las dependencias, estas se encuentran en la página https://mvnrepository.com/

Buscando hibernate, mysql, persistence, etc y en maven nos dá el código que tendríamos que poner





org.hibernate.jpa.HibernatePersistenceProvider

tenemos que crear las entidades, una por cada tabla de la base de datos teniendo en cuanta los tipos de datos, las primary key y las relaciones si hay más de una tabla

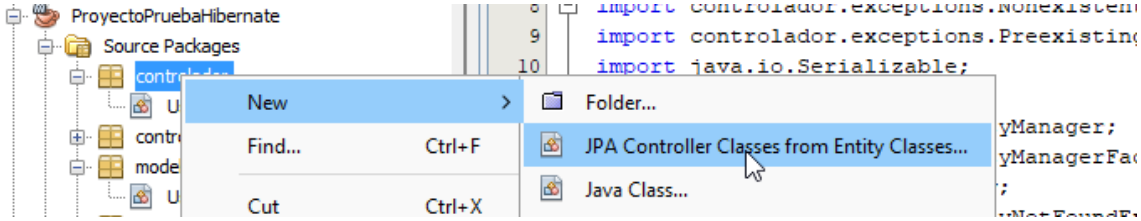
```
@Entity
@Table (name="empleados")
public class Empleado implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @Column (name = "emp_no")
    private Short emp_no;

    @Column (name = "apellido")
    private String apellido;

    @Column (name = "comision")
    private Float comision;

    @ManyToOne
    @JoinColumn (name = "dept_no")
    private Departamento dept_no;
```

Nombre de la tabla en la BD

Esto siempre hay que ponerlo

Luego creamos varios constructores, los geter y seter y también en insert nos da la opción de añadir el código de unos métodos que se llaman hashCode y equals. Los añadimos

Una vez terminada la Entidad podemos crear el controlador



Aquí añadiríamos los métodos con el código HQL con las consultas a la base de datos. Igualmente nos da algunos métodos creados con los que se pueden hacer casi todas las operaciones más usuales

```
public class UsuarioJpaController implements Serializable {

    public UsuarioJpaController(EntityManagerFactory emf) {
        this.emf = emf;
    }

    public UsuarioJpaController () {
        emf = Persistence.createEntityManagerFactory("usuarios");
    }
```

Tenemos que añadir este constructor sin parámetros, el segundo.

En las clases en las que vamos a necesitar acceso a la base de datos creamos un objeto del controlador con el constructor que hemos creado antes

```
public class ControladorMenu implements ActionListener{

    private MenuPrincipal vistaMenu;
    private UsuarioJpaController usuarioJpaController;

    public ControladorMenu(MenuPrincipal menu) {
        this.vistaMenu = menu;
        usuarioJpaController = new UsuarioJpaController();

        this.vistaMenu.btn_add.addActionListener(this);
    }
```