

## Ejemplo de Persistencia con JAVA

### Que se busca:

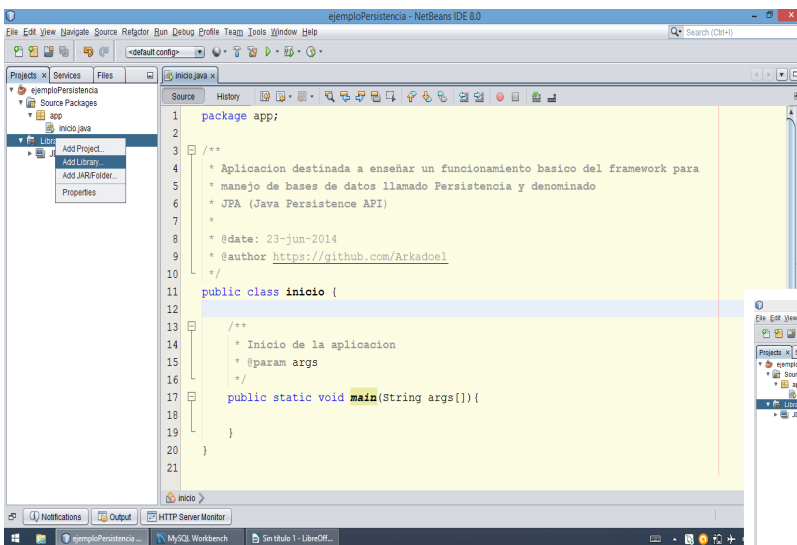
Dar un conocimiento básico acerca de como usar fácilmente bases de datos con java sin volverse loco.

### Software usado

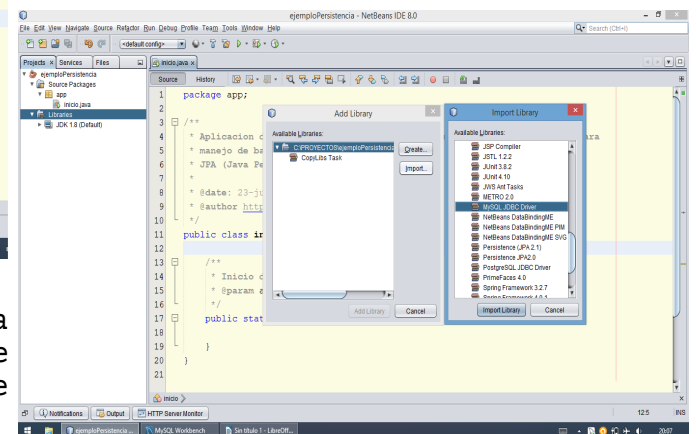
- Netbeans 8.0
- MySQL 5.6 con workbench

### Comencemos

Lo primero que haremos es ir a netbeans y crear un proyecto java de escritorio normal y corriente. Una vez lo hayamos hecho podremos comenzar a poner librerías, para ello daremos boton derecho en la carpeta del proyecto llamada “Libraries” y seleccionaremos el menú “Add Library”.

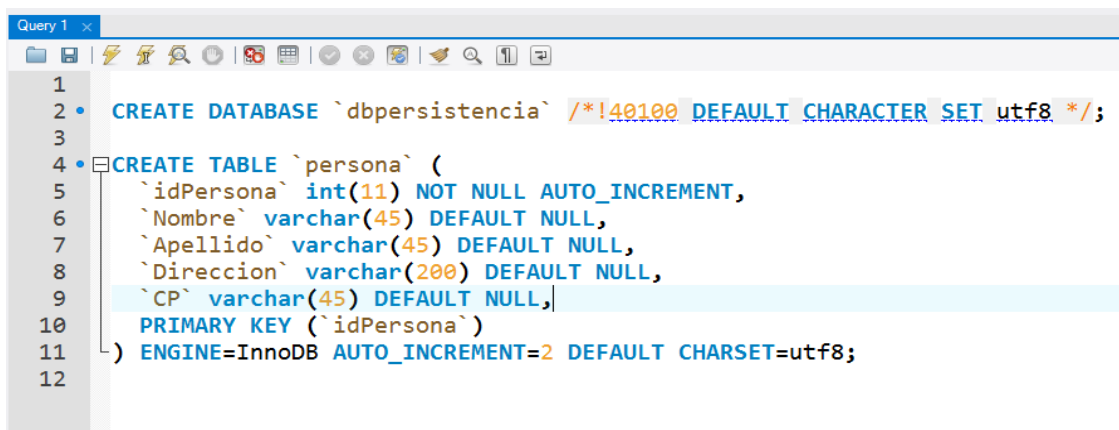


Una vez hayamos seleccionado el menú nos aparecera la pantalla de añadir librerías, como lo mas probable es que falte la librería de JDBC para MySQL deberemos de dar a “Import”, buscar la librería y añadirla.

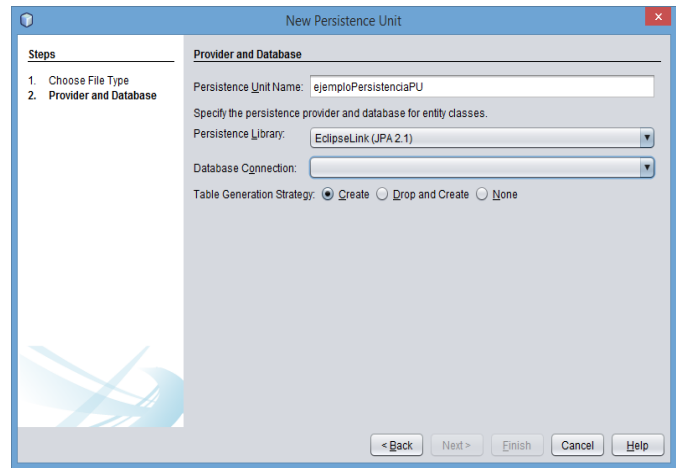
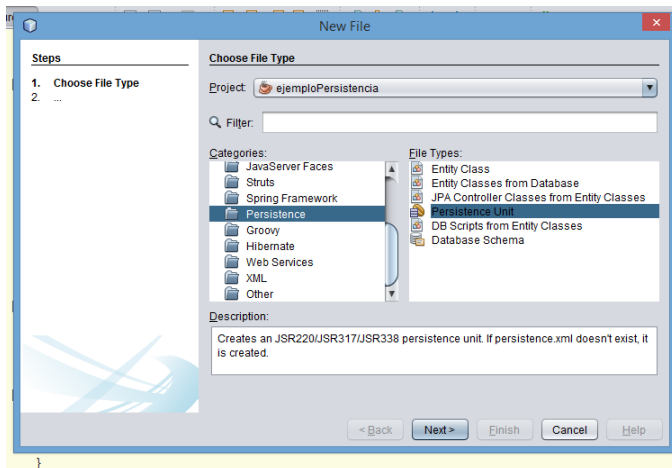


Tambien deberemos añadir una librería llamada Persistence JPA (2.1) que es la que nos permitirá acceder facilmente a la base de datos.

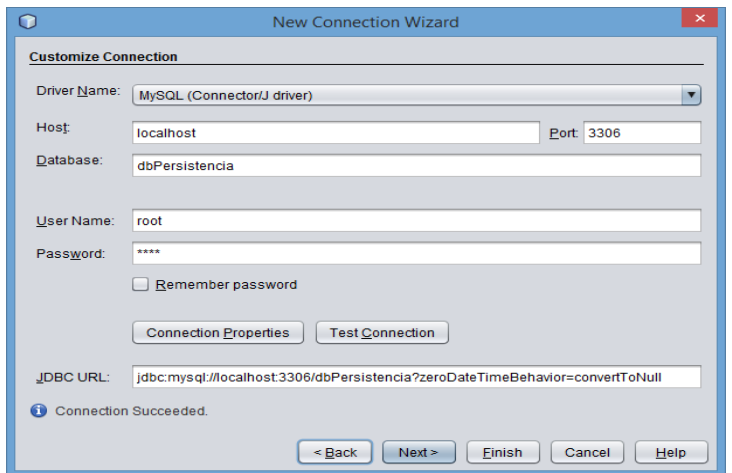
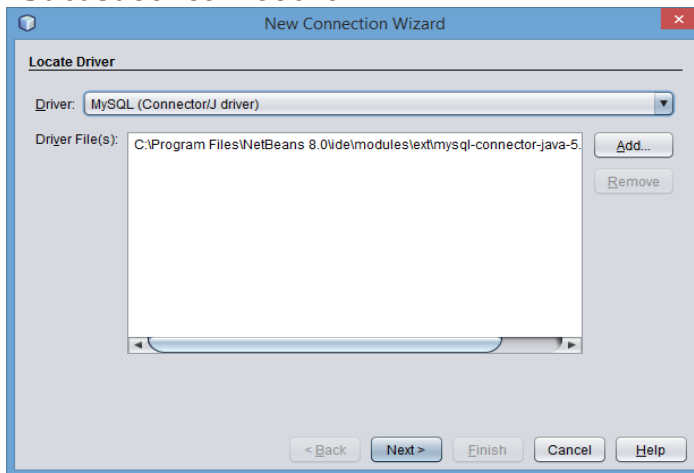
Creamos la base de datos en MySQL



Una vez creada la base de datos, vamos a nuestro proyecto en Netbeans y dando boton derecho encima del proyecto, vamos a “New → Other → Persistence” seleccionando una nueva unidad de persistencia:



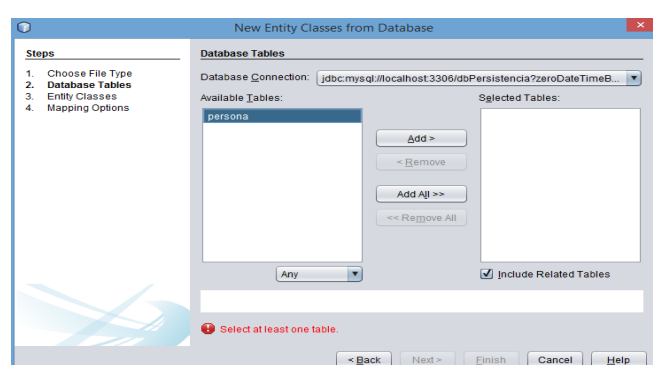
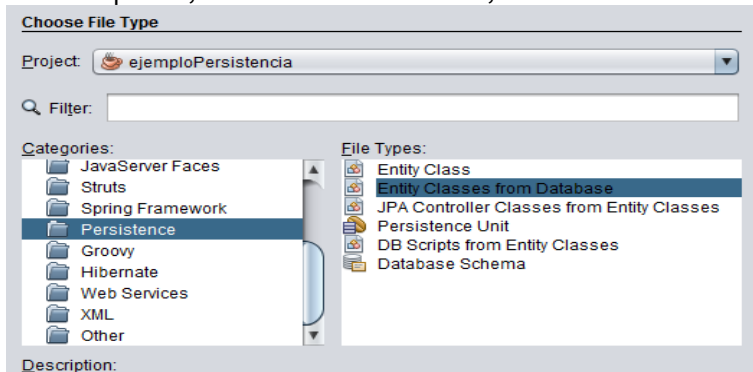
Como no tenemos ninguna conexión a base de datos, la creamos en el combo “Database Connection”



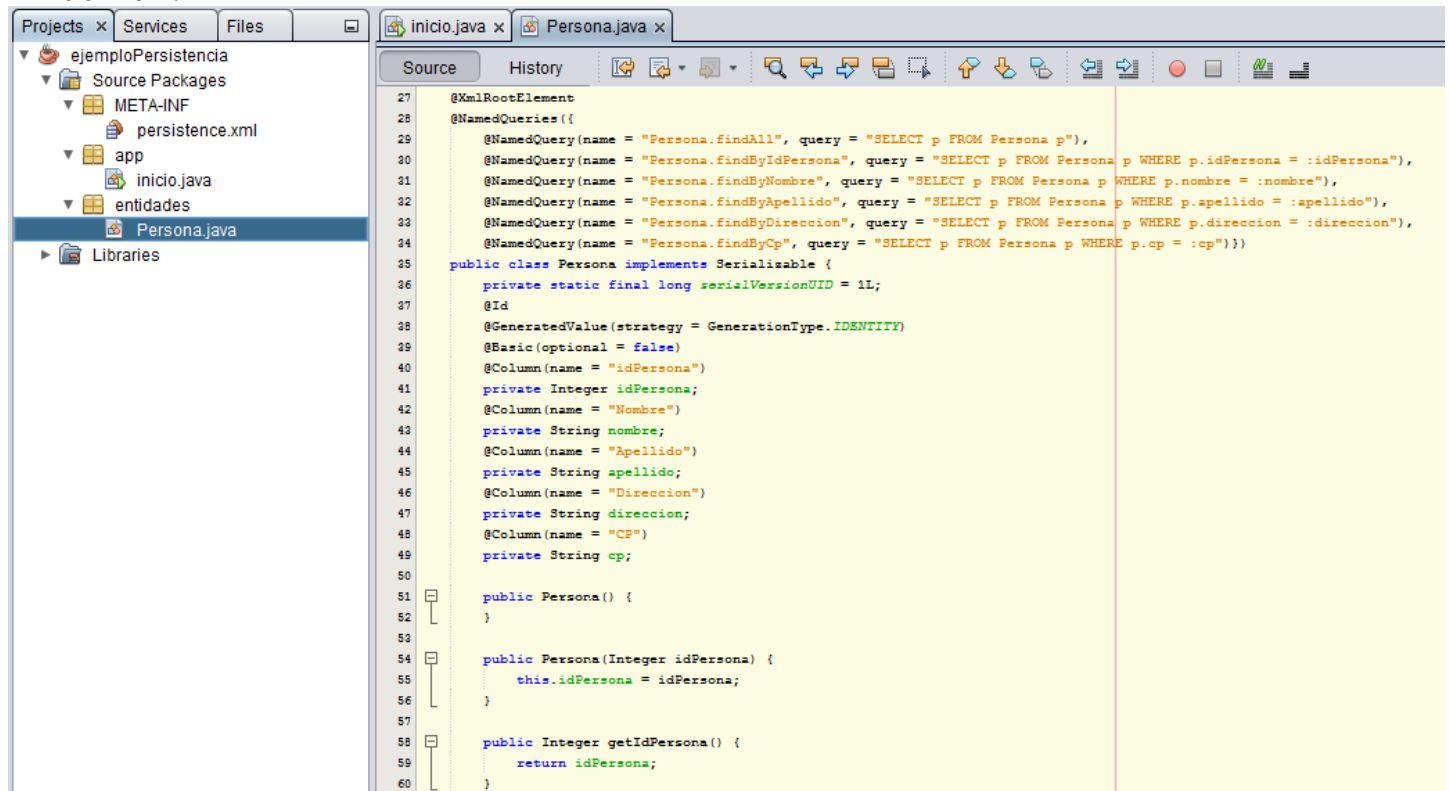
Nos aparecera un nuevo archivo persistence.xml que contiene la unidad de persistencia.



Ahora crearemos un paquete que contendra las distintas tablas de la base de datos en forma de objetos, lo que comunmente se denomina objetos entidad de una base de datos. Para ello damos boton derecho en el proyecto y volvemos a ir hasta la zona en donde dimos a “Persistence unit” para, en esta ocasión, dar a “Entities from database”



Añadimos las tablas que queramos usar, en este caso solo tenemos una así que no hay problema. El resto de pantallas las dejamos como están y ya podemos finalizar.



```

27  @XmlElement
28  @NamedQueries({
29      @NamedQuery(name = "Persona.findAll", query = "SELECT p FROM Persona p"),
30      @NamedQuery(name = "Persona.findByIdPersona", query = "SELECT p FROM Persona p WHERE p.idPersona = :idPersona"),
31      @NamedQuery(name = "Persona.findByName", query = "SELECT p FROM Persona p WHERE p.nombre = :nombre"),
32      @NamedQuery(name = "Persona.findByApellido", query = "SELECT p FROM Persona p WHERE p.apellido = :apellido"),
33      @NamedQuery(name = "Persona.findByDireccion", query = "SELECT p FROM Persona p WHERE p.direccion = :direccion"),
34      @NamedQuery(name = "Persona.findByCp", query = "SELECT p FROM Persona p WHERE p.cp = :cp"))
35  public class Persona implements Serializable {
36      private static final long serialVersionUID = 1L;
37      @Id
38      @GeneratedValue(strategy = GenerationType.IDENTITY)
39      @Basic(optional = false)
40      @Column(name = "idPersona")
41      private Integer idPersona;
42      @Column(name = "Nombre")
43      private String nombre;
44      @Column(name = "Apellido")
45      private String apellido;
46      @Column(name = "Direccion")
47      private String direccion;
48      @Column(name = "CP")
49      private String cp;
50
51      public Persona() {
52      }
53
54      public Persona(Integer idPersona) {
55          this.idPersona = idPersona;
56      }
57
58      public Integer getIdPersona() {
59          return idPersona;
60      }

```

Esta operación nos habrá generado una serie de objetos entidad que nos ayudaran a manejar la base de datos de forma sencilla y sin demasiadas complicaciones.

Crearemos ahora una clase dentro del paquete "app" que sera la encargada de las distintas operaciones en la base de datos.

Junto a este manual hay un proyecto para netbeans en el que podrá ver las distintas operaciones a realizar.