

## EJERCICIO \_1:

Crear una base de datos llamada Librería, en puesto cliente, con dos **tablas una de libros y otra de autores**.

**Tabla Libros:** CodLibro/Título/Precio.

**Tabla Autores:** IdAutor/Nombre/Nacionalidad.

La relación entre Libros y Autores es de varios a varios: un autor puede escribir varios libros y un libro puede tener varios autores, por ello necesitamos una tercera tabla que relacione a libros y autores.

**Tabla Autores\_libros:** CodLibro/IdAutor.

Realizar un programa en java, que cargue un **driver JDBC** y establezca la conexión con **una BD MySQL**, la cual reside en la misma máquina cliente junto al **driver**, **la aplicación** y el **servidor USBW portable**, esta generará un menú que invoque a las distintas clases con sus respectivos métodos, que se exponen a continuación:

Creación de la Base de Datos y tablas.

### **MENÚ**

2- Inserción de nuevas filas.

- a. Inserción libro.
- b. Inserción autor.

3- Borrado de filas.

- a. Borrado libro.
- b. Borrado autor.

4-Modificaciones.

Introduciendo el código de un libro modifique su precio.

6- Consultas.

- a. Introduciendo el título de un libro visualice sus datos y el nombre del autor ó autores.
- b. Introduciendo el nombre de un autor visualice sus libros.
- c. Visualización de todos los Libros de la tabla.
- d. Visualización de todos los autores con sus libros.

7-Fin.

Para crear la bd después de establecer la conexión tendremos que poner el símbolo ? En lugar del nombre de la bd ejemplo:

```
String url= "jdbc:mysql://localhost:3307/?user=root&password=usbw";  
Statement st= "CREATE DATABASE";
```

Un ejemplo de creación de tablas:

```
class CreacionTablas {  
    public static void estructura(Connection conexion, Statement sentencia){  
        try{  
            sentencia.execute("CREATE TABLE IF NOT EXISTS autores ( "  
                                + "idAutor INT(11) UNSIGNED ZEROFILL NOT NULL AUTO_INCREMENT  
                                + "nombre VARCHAR(30) NOT NULL, "  
                                + "nacionalidad VARCHAR(30) NOT NULL, "  
                                + "PRIMARY KEY (idAutor));");  
            sentencia.execute("CREATE TABLE IF NOT EXISTS libros ( "  
                                + "idLibro INT(11) UNSIGNED ZEROFILL NOT NULL AUTO_INCREMENT  
                                + "titulo VARCHAR(30) NOT NULL, "  
                                + "precio FLOAT NOT NULL, "  
                                + "PRIMARY KEY (idLibro));");  
            sentencia.execute("CREATE TABLE IF NOT EXISTS librosautores ( "  
                                + "autor INT(11) UNSIGNED ZEROFILL NOT NULL, "  
                                + "libro INT(11) UNSIGNED ZEROFILL NOT NULL, "  
                                + "PRIMARY KEY (autor,libro), "  
                                + "INDEX fk_autor (autor), "  
                                + "INDEX fk_libro (libro), "  
                                + "CONSTRAINT fk_autor "  
                                + "FOREIGN KEY (autor) "  
                                + "REFERENCES autores(idAutor) "  
                                + "ON DELETE CASCADE "  
                                + "ON UPDATE CASCADE, "  
                                + "CONSTRAINT fk_libro "  
                                + "FOREIGN KEY (libro) "  
                                + "REFERENCES libros(idLibro) "  
                                + "ON DELETE CASCADE "  
                                + "ON UPDATE CASCADE );");  
        } catch (SQLException e) {  
            <
```