

Ejercicio de base de datos.

Crear la siguiente base de datos **bbddprueba**:

```
CREATE DATABASE bbddprueba;
```

```
USE bddprueba;
```

```
CREATE TABLE contacto(
```

```
Nombre varchar(40),
```

```
correo varchar(40),
```

```
telefono varchar(24),
```

```
direccion varchar(60)
```

```
);
```

```
ALTER TABLE contacto ADD PRIMARY KEY (telefono);
```

```
INSERT INTO contacto values("Pepe","pepe@gmail.com","+698765465","C/campamento N55");
```

```
INSERT INTO contacto values("Luis","luis@gmail.com","+654654654","C/rosas N25");
```

```
INSERT INTO contacto values("Ana","ana@gmail.com","+213213213","C/leles N22");
```

Antes de empezar cargar nuestro **driver** mysql-connector-java-5.1.22-bin dentro de la biblioteca de nuestro proyecto.

Crear la siguiente clase java llamada **Contacto** con los atributos: nombre, correo, telefono y dirección todos ellos de texto y privados. Incluir sus métodos get y set para los atributos.

Crear el paquete DAO y dentro el fichero **Conexion_DB.java** en él importaremos la clase java.sql.* necesaria para el manejo de base de datos.

En nuestro **main** lo primero será llamar al método **abrirConexión** creado en Conexion_DB donde nos pedirá la base de datos y todo el código necesario, cuando tengamos conexión con la bdd nos aparecerá el siguiente menú:

Menú

1- Insertar contacto

2- Borrar contacto

3- Mostrar contacto

4- Mostrar Todos los contactos

5- Salir

Cada opción en un switch con su respectivo método.

Método insertarContacto:

Crearemos un objeto Contacto llamado por ejemplo `contactonuevo`, rellenaremos sus cuatro valores mediante `sout` y los `set` correspondientes.

Cuando ya este relleno pasamos a meter ese `contactonuevo` en la `bd`, primero la gestión de errores cuando trabajamos con ficheros o `bd` con la orden `try` dentro de esta comenzamos creando un *Statement* y ejecutando un `update` sobre la `bd` pasándole la `String` para insertar.

Método borrarContacto:

Guardamos en un `String` el nombre del contacto a borrar.

Dentro de un `try` creamos el *statement* y ejecutamos el `update` con la sentencia `SQL` de eliminación del contacto.

Método mostrarContacto:

Para `monstarContacto` el principio es igual pedimos un nombre de contacto lo guardamos, después creamos el *Statement* y ahora es diferente no queremos ni insertar ni borrar sino mostrar:

Hemos de crear un `ResultSet` que ejecute la consulta deseada y posteriormente en un bucle `while` mostrar cada una de las filas por pantalla recuperando los cuatro campos con `getString("nombrecolumna")`.

Método mostrarTodosContactos:

Más sencillo que el anterior simplemente crear el *Statement* y el *ResultSet* y mostrar en bucle por pantalla los datos de la `bd`.

Método cerrarConexion:

En un `try` ya que nos obliga al trabajar con `bd` cerrar la conexión llamando al método de `Conexion_DB` y gestionar su excepción.