



---

# HITO INDIVIDUAL ACCESO A DATOS

---

Iván González Gordo DAM2



26 DE OCTUBRE DE 2022

CAMPUSFP  
GETAFE

## TECNOLOGÍAS UTILIZADAS

Las tecnologías usadas han sido Java 17 con Maven, MySQL y PostgreSQL para las bases de datos y como IDE IntelliJ de JetBrains.

### SCRIPT MYSQL

```
DROP TABLE IF EXISTS `facturas`;
```

```
CREATE TABLE `facturas` (
```

```
  `idfactura` int NOT NULL AUTO_INCREMENT,
```

```
  `idusUario` int DEFAULT NULL,
```

```
  `concepto` varchar(50) DEFAULT NULL,
```

```
  `fecha` varchar(25) DEFAULT NULL,
```

```
  PRIMARY KEY (`idfactura`),
```

```
  KEY `idusUario` (`idusUario`),
```

```
  CONSTRAINT `facturas_ibfk_1` FOREIGN KEY (`idusUario`) REFERENCES `usuarios` (`id`)
```

```
);
```

```
INSERT INTO `facturas` VALUES (1,1,'el mejor coche','17-12-2022'),(2,1,'la mejor moto','17-12-2022'),(3,2,'el mejor libro','17-12-2022'),(4,1,'TECLADOS PARA CLASE DAM 2','01-05-2021'),(5,1,'RATONES PARA CLASE DAM 2','01-05-2021'),(6,2,'PANTALLAS PARA CLASE DAM 2','01-05-2021'),(7,2,'ORDENADORES PARA CLASE DAM 2','01-05-2021');
```

```
DROP TABLE IF EXISTS `usuarios`;
```

```
CREATE TABLE `usuarios` (
```

```
  `id` int NOT NULL AUTO_INCREMENT,
```

```
  `nombre` varchar(25) DEFAULT NULL,
```

```
  `apellidos` varchar(100) DEFAULT NULL,
```

```
  `NIF` char(9) DEFAULT NULL,
```

```
  PRIMARY KEY (`id`)
```

```
);
```

```
INSERT INTO `usuarios` VALUES
```

```
(1,'carlos','gonzalez','49589604V'),(2,'javier','martin','49589605H'),(3,'javier','gutierrez','49589605A');
```

### SCRIPT POSRGRESQL

```
create table usuarios(
```

```
    id serial PRIMARY KEY,
```

```
    nombre varchar(25),
```

```
    apellidos varchar(100),
```

```
    NIF char(9)
```

```
);
```

```
create table facturas(
```

```
    idfactura serial,
```

```
    idusUario int,
```

```
    concepto varchar(50),
```

```
    fecha varchar(25),
```

```
    PRIMARY KEY (idfactura),
```

```
    FOREIGN KEY (idusuario) REFERENCES usuarios(id)
```

```
);
```

```
insert into usuarios(nombre, apellidos, NIF) values('Pepe', 'Perez Perez', '12345678A');
```

```
insert into usuarios(nombre, apellidos, NIF) values('Juan', 'Gomez Gomez',  
'12345678B');
```

```
insert into usuarios(nombre, apellidos, NIF) values('Maria', 'Garcia Garcia',  
'12345678C');
```

```
insert into usuarios(nombre, apellidos, NIF) values('Luis', 'Lopez Lopez',  
'12345678D');
```

```
insert into facturas (idusuario, concepto, fecha) values (1, 'TECLADOS PARA CLASE DAM 2', '01-  
05-2021');
```

```
insert into facturas (idusuario, concepto, fecha) values (1, 'RATONES PARA CLASE  
DAM 2', '01-05-2021');
```

```
insert into facturas (idusuario, concepto, fecha) values (2, 'PANTALLAS PARA CLASE  
DAM 2', '01-05-2021');
```

```
insert into facturas (idusuario, concepto, fecha) values (2, 'ORDENADORES PARA  
CLASE DAM 2', '01-05-2021');
```

### Capturas de pantalla

Elección entre acceder un mysql o postgresql:

```
Elige una opción:  
1. Mysql  
2. Postgresql  
3. Salir  
|
```

Dentro de cada base de datos puedes acceder a las dos tablas usuarios o facturas:

```
¿Desea acceder a los usuarios o a las facturas?  
1. Usuarios  
2. Facturas  
|
```

Si accedes a cualquiera de las dos tablas te indica el nombre de las tablas que vas a usar y un menú con las opciones disponibles:

```
Tabla de usuarios  
¿Que accion desea realizar?  
1. Añadir.  
2. Listado.  
3. Modificar.  
4. Exportar CSV  
5. Importar CSV  
6. Salir  
|
```

Si eliges añadir:

```
1  
Introduzca el nombre, apellidos y NIF separados por comas  
Recuerda que el concepto ni ningun campo puede ir separado por espacios  
|
```

Si eliges listado:

Luego tienes dos opciones: una o todas

```
2
¿Desea ver una o todas?
todas
NUM NOMBRE      PESO    ALTURA
1   ivan      gonzalez 49589604V
2   javier    martin  49589605H
3   javier    gutierrez 49589605A
```

Si deseas ver una tienes que escribir los parámetros de búsqueda:

```
¿Desea ver una o todas?
una
Escriba la condicion separadas por coma. EJ: nombre,ivan
Recuerde, puede buscar nombre, apellidos, NIF
nombre,javier
nombre
javier
NUM NOMBRE      APELLIDOS  NIF
2   javier    martin  49589605H
3   javier    gutierrez 49589605A
```

Si eliges la opción de modificar:

```
3
Escriba la condicion separadas por coma. EJ: nombre,ivan
nombre,ivan
Escriba que quiera cambiar separado por coma. EJ: nombre,ivan
nombre,carlos
Cambio realizado con exito!
```

Si eliges la opción de exportar CSV:

```
Introduzca la ruta absoluta
C:\Users\Campus FP\Downloads\archivos
ruta: C:\Users\Campus FP\Downloads\archivos
Tabla: usuarios
id
nombre
apellidos
NIF
CSV File is created successfully.Tabla de usuarios
```

Si eliges la opción de importar CSV:

```
5
Introduzca la ruta absoluta
C:\Users\Campus FP\Downloads\archivos\usuarios.csv
Desea borrar los datos anteriores o sobreescribirlos? (sobreescibir/añadir)
Recuerda que si introduce un dato que ya existe, se sobreescibirá
sobreescibir
id igual 1
Update 1: carlos
carlos
Update 2: gonzalez
gonzalez
Update 3: 49589604V
49589604V
Repe valor: 1
id igual 2
Update 1: javier
javier
Update 2: martin
martin
Update 3: 49589605H
49589605H
Repe valor: 2
id igual 3
Update 1: javier
javier
Update 2: gutierrez
gutierrez
Update 3: 49589605A
49589605A
```

```
Repe valor: 3
Data Successfully Uploaded
```

Si eliges la opción de Salir vuelves al menú del principio:

```
8. Salir
6
Elige una opción:
1. Mysql
2. Postgresql
3. Salir
|
```

### Explicación de las partes más relevantes del código:

El constructor de la clase BDatos permite conectar con una base de datos mysql o postgresql y realizar las conexiones para introducirlas como propiedad de una clase. Para ello se utiliza un condicional y en cualquiera de las dos clases utiliza Class.forName para cambiar el driver de mysql a postgresql y viceversa.

```
public BDatos(String baseDatos, String user, String contra, String tabla, String bd) {
    this.cadenaConexion = this.cadenaConexion + baseDatos;
    this.cadenaConexionposgre = this.cadenaConexion + baseDatos;
    this.user = user;
    this.pass = contra;
    this.tabla = tabla;

    if (bd.equals("mysql")){
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            this.con = DriverManager.getConnection(cadenaConexion, user, pass);
            this.con.setAutoCommit(false);
            //Statement sentencia = con.createStatement();
        } catch (SQLException e) {
            System.out.println("No se ha podido establecer la conexión con la BD");
            System.out.println(e.getMessage());
        } catch (ClassNotFoundException e) {
            System.out.println("No se ha encontrado el driver para MySQL");
            return;
        } catch (Exception e){System.out.println("ERROR GENERAL");}
    } else if (bd.equals("postgres")) {
        //Conexión a PostgreSQL
        try {
            Class.forName("org.postgresql.Driver");
            this.con = DriverManager.getConnection("jdbc:postgresql://localhost/prueba?user="+user+"&password="+pass);
            //this.con = DriverManager.getConnection(cadenaConexionposgre, user, pass);
            this.con.setAutoCommit(false);
            //Statement sentencia = con.createStatement();
        } catch (SQLException e) {
            System.out.println("No se ha podido establecer la conexión con la BD");
            System.out.println(e.getMessage());
        } catch (ClassNotFoundException e) {
            System.out.println("No se ha encontrado el driver para PostgreSQL");
            return;
        } catch (Exception e){System.out.println("ERROR GENERAL");}
    } else {
        System.out.println("ERROR EN RECONOCER EL TIPO DE BASE DE DATOS");
    }
}
```

```
while(true){
    Scanner sc0pc = new Scanner(System.in);
    System.out.println("Elige una opción:");
    System.out.println("1. Mysql");
    System.out.println("2. Postgresql");
    System.out.println("3. Salir");
    String sc44 = sc0pc.next();
    BDatos bdUsuarios = null;
    BDatos bdFacturas = null;
    if (sc44.equals("1")){
        bdUsuarios = new BDatos("PRUEBA", "root", "curso", "usuarios", "mysql");
        bdFacturas = new BDatos("PRUEBA", "root", "curso", "facturas", "mysql");
    } else if (sc44.equals("2")) {
        bdUsuarios = new BDatos("prueba", "ivi", "ivi", "usuarios", "postgre");
        bdFacturas = new BDatos("prueba", "ivi", "ivi", "facturas", "postgre");
    } else if (sc44.equals("3")) {
        System.exit(0);
    }
    else{
        System.out.println("Opción incorrecta");
    }
}
```

Para sobrescribir lo que ya exista y añadir el nuevo contenido actualizando se realiza este código:

```
String repe = "0";
while((rowData = reader.readNext()) != null)
{
    repe = "0";
    PreparedStatement pstmt = con.prepareStatement(insertQuery);
    PreparedStatement update = con.prepareStatement(updateQuery);
    for (String data : rowData)
    {
        if (data.equals("id")){
            flag = true;
            break;
        }
        if (i == 5){
            break;
        }
        if (repe.equals("0")){
            while (rs.next()){
                if (rs.getString(columnLabel "id").equals(data)){
                    repe = rs.getString(columnLabel "id");
                    System.out.println("id igual " + repe);
                    break;
                }
            }
        }

        if (!repe.equals(data)){
            System.out.println("Update " + i + ": " + data);
            System.out.println(data);
            update.setString(i++, data);
            flag = false;
        }else if(repe.equals("0")){
            System.out.println(data);
            pstmt.setString(i++, data);
            flag = false;
        }
    } // Cierra for
    if (!flag){
        i = 1;
        System.out.println("Repe valor: " + repe);
        if (repe.equals("0")){
            pstmt.executeUpdate();
            con.commit();
        }else{
            update.setString(parameterIndex 4, repe);
            update.executeUpdate();
            con.commit();
        }
    }
}
```

Donde se realizan una serie de comprobaciones y dependiendo de si necesita actualizar o simplemente añadir se añade a la sentencia necesaria y se realiza el commit. A través de varios System.out.println podemos comprobar que los valores repetidos se actualizan.