



## Programación con Java

TAREA 22

<b>Índice.....</b>	<b>1</b>
<b>Introducción.....</b>	<b>2</b>
<b>Patrón MVC.....</b>	<b>3</b>
Ejercicio 01:.....	3
Ejercicio 02:.....	4
Ejercicio 03:.....	5
<b>Webgrafía.....</b>	<b>8</b>

En el proceso de esta práctica aprenderemos a utilizar el patrón de arquitectura Modelo-Vista-Controlador para la ejecución de los siguientes programas.

Estos programas consistirán en una aplicación para hacer las acciones CRUD en una base de datos, para ello en cada main del programa haré la conexión, crearé la base de datos del ejercicio y las tablas en las que tendremos que hacer las acciones.

## **Modelo-Vista-Controlador:**

Este patrón establece una arquitectura a seguir en el programa. Esta consiste en definir un mínimo de 3 archivos:

- El archivo del Modelo, solo contendrá valores/datos que necesitemos en los otros programas.
- En la Ventana tendremos que crear el frame con los componentes que tendrá, establecer los tamaños, colores, etc. **solo** la parte visual.
- Por último, el controlador, utilizará las otras dos partes para añadir las acciones a los componentes.

## Ejercicio 01:

En el primer ejercicio crearemos una única tabla llamada “cliente” con las columnas: “id, nombre, apellido, direccion, dni, fecha”.

### AppEj01

```
class AppEj01 {
    Run | Debug
    public static void main(String[] args) {
        Connection conec = null;
        Modelo modelo = new Modelo();
        // Hacer conexión con el localhost
        conec = BasesDeDatos.hacerConexionConBD(modelo.getConexion(), modelo.getHost(), modelo.getPort(),
            modelo.getDatabase(), modelo.getUser(), modelo.getPassword());
        modelo.setConexion(conec);
        // Crear base de datos para el ejercicio
        modelo.setDatabase(databaseName:"crud_ej01");
        conec = BasesDeDatos.crearDataBase(modelo.getConexion(), modelo.getDatabase());
        modelo.setConexion(conec);
        // Crear la tabla "cliente"
        modelo.setTable(tableName:"cliente");
        modelo.setValues(
            "(id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY, " +
            "nombre NVARCHAR(250) DEFAULT NULL, " +
            "apellido NVARCHAR(250) DEFAULT NULL, " +
            "direccion NVARCHAR(250) DEFAULT NULL, " +
            "dni int(11) DEFAULT NULL, " +
            "fecha date DEFAULT NULL)");
        conec = BasesDeDatos.crearTabla(modelo.getConexion(), modelo.getDatabase(), modelo.getTable(),
            modelo.setValues());
        modelo.setConexion(conec);

        String columnasAfectadas = "(nombre, apellido, direccion, dni, fecha)";
        String values = ("('Juan', 'Pérez', 'Calle Falsa 123', 12345678, '2020-01-01'), " +
            + "('Maria', 'López', 'Avenida Siempre 742', 87654321, '2021-02-02'), " +
            + "('Carlos', 'Gómez', 'Calle de la Paz 456', 45678912, '2022-03-03'), " +
            + "('Ana', 'Martínez', 'Puerta del Sol 789', 78912345, '2023-04-04')");

        conec = BasesDeDatos.insertarValores(conec, modelo.getDatabase(), modelo.getTable(), columnasAfectadas,
            values);

        int columnas = BasesDeDatos.cantidadColumnas(modelo.getConexion(), modelo.getTable());
        int filas = BasesDeDatos.cantidadFilas(modelo.getConexion(), modelo.getTable());
        ArrayList<String> nombres = BasesDeDatos.recogerNombresColumnas(modelo.getConexion(),
            modelo.getTable());
        Ventana ventana = new Ventana(modelo.getTable(), filas, columnas,
            new ArrayList<String>(Arrays.asList(...a:"cliente")), nombres);

        ArrayList<ArrayList<String>> datosTabla = BasesDeDatos.recogerValoresTabla(conec, modelo.getTable());
        PanelTabla.actualizarTabla(datosTabla);
        PanelConsola.setEntradaConsola(BasesDeDatos.getTextoConsola());

        new Controlador(modelo, ventana);
    }
}
```

## Ejercicio 02:

Para el segundo ejercicio crearemos la misma tabla del ejercicio anterior pero habrá otra tabla llamada “videos” con las columnas: “id, title, director, clid\_id” donde “clid\_id” será foránea de la primera tabla.

### AppEj02

```
class AppEj02 {
    Run | Debug
    public static void main(String[] args) {
        Connection conec = null; Modelo modelo = new Modelo(); String columnasAfectadas = ""; String values = "";
        ArrayList<String> tablas = new ArrayList<>(Arrays.asList(...:"cliente", "videos"));

        // Hacer conexión con el localhost
        conec = BasesDeDatos.hacerConexionConBD(modelo.getConexion(), modelo.getHost(), modelo.getPort(),
            modelo.getDatabase(), modelo.getUser(), modelo.getPassword());
        modelo.setConexion(conec);
        // Crear base de datos para el ejercicio
        modelo.setDatabase(databaseName:"crud_ej02");
        conec = BasesDeDatos.crearDataBase(modelo.getConexion(), modelo.getDatabase());
        modelo.setConexion(conec);
        // Crear la tabla "cliente"
        modelo.setTable(tablas.get(index:0));
        modelo.setValues("(id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY, " +
            "nombre NVARCHAR(250) DEFAULT NULL, " +
            "apellido NVARCHAR(250) DEFAULT NULL, " +
            "direccion NVARCHAR(250) DEFAULT NULL, " +
            "dni INT(11) DEFAULT NULL, " +
            "fecha date DEFAULT NULL)");
        conec = BasesDeDatos.crearTabla(modelo.getConexion(), modelo.getDatabase(), modelo.getTable(), modelo.getValues());
        modelo.setConexion(conec);

        columnasAfectadas = "(nombre, apellido, direccion, dni, fecha)";
        values = ("('Pedro', 'Garcia', 'Calle Mayor 456', 98765432, '2019-05-15'), "
            + "('Laura', 'Sánchez', 'Avenida del Sol 789', 65432198, '2020-06-20'), "
            + "('Antonio', 'Fernández', 'Calle del Río 321', 74185296, '2021-07-25'), "
            + "('Elena', 'Díaz', 'Plaza España 654', 85296374, '2022-08-30')");

        conec = BasesDeDatos.insertarValores(conec, modelo.getDatabase(), modelo.getTable(), columnasAfectadas, values);

        // Crear la tabla "videos"
        modelo.setTable(tablas.get(index:1));
        modelo.setValues("(id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY, " +
            "title NVARCHAR(250) DEFAULT NULL, " +
            "director NVARCHAR(250) DEFAULT NULL, " +
            "clid_id INT(11) DEFAULT NULL, " +
            "CONSTRAINT videos_fk FOREIGN KEY (clid_id) REFERENCES cliente (id))");
        conec = BasesDeDatos.crearTabla(modelo.getConexion(), modelo.getDatabase(), modelo.getTable(), modelo.getValues());
        modelo.setConexion(conec);

        columnasAfectadas = "(title, director, clid_id)";
        values = ("('El Secreto de sus Ojos', 'Juan José Campanella', 1), "
            + "('Relatos Salvajes', 'Damián Szifron', 2), "
            + "('Nueve Reinas', 'Fabián Bielinsky', 3), "
            + "('El Hijo de la Novia', 'Juan José Campanella', 4)");

        conec = BasesDeDatos.insertarValores(conec, modelo.getDatabase(), modelo.getTable(), columnasAfectadas,
            values);

        int columnas = BasesDeDatos.cantidadColumnas(modelo.getConexion(), modelo.getTable());
        int filas = BasesDeDatos.cantidadFilas(modelo.getConexion(), modelo.getTable());
        ArrayList<String> nombres = BasesDeDatos.recogerNombresColumnas(modelo.getConexion(),
            modelo.getTable());
        Ventana ventana = new Ventana(modelo.getTable(), filas, columnas, tablas, nombres);

        ArrayList<ArrayList<String>> datosTabla = BasesDeDatos.recogerValoresTabla(conec, modelo.getTable());
        PanelTabla.actualizarTabla(datosTabla);
        PanelConsola.setEntradaConsola(BasesDeDatos.getTextoConsola());

        new Controlador(modelo, ventana);
    }
}
```

## Ejercicio 03:

Por último, en el tercer programa, crearemos tres tablas:

Tabla 1: “cientificos” con las columnas: “dni, NomApels”

Tabla 2: “proyecto” con las columnas: “id, nombre, horas”

Tabla 3: “asignado\_a” con las columnas: “cientifico, proyecto” donde “cientifico” y “proyecto” son foráneas de las anteriores.

### AppEj03

```
class AppEj03 {
    Run | Debug
    public static void main(String[] args) {
        Connection conec = null;
        Modelo modelo = new Modelo();
        String columnasAfectadas = "";
        String values = "";
        ArrayList<String> tablas = new ArrayList<>(Arrays.asList(...a:"cientificos", "proyecto", "asignado_a"));

        conec = BasesDeDatos.hacerConexionConBD(modelo.getConexion(), modelo.getHost(), modelo.getPort(),
            modelo.getDatabase(), modelo.getUser(), modelo.getPassword());
        modelo.setConexion(conec);
        modelo.setDatabase(databaseName:"crud_ej03");
        conec = BasesDeDatos.crearDataBase(modelo.getConexion(), modelo.getDatabase());
        modelo.setConexion(conec);
        // Crear la tabla "cientificos"
        modelo.setTable(tablas.get(index:0));
        modelo.setValues(newValues:"(DNI varchar(8) NOT NULL PRIMARY KEY, NOMAPELS nvarchar(255) NOT NULL)");
        conec = BasesDeDatos.crearTabla(modelo.getConexion(), modelo.getDatabase(), modelo.getTable(),
            modelo.getValues());
        modelo.setConexion(conec);
        columnasAfectadas = "(DNI, NOMAPELS)";
        values = ("('12345678', 'Juan Pérez'), ('87654321', 'Ana López'), " +
            "('98765432', 'Pedro García'), ('23456789', 'Maria González');");
        conec = BasesDeDatos.insertarValores(conec, modelo.getDatabase(), modelo.getTable(), columnasAfectadas, values);

        // Crear la tabla "proyecto"
        modelo.setTable(tablas.get(index:1));
        modelo.setValues(newValues:"(ID char(4) PRIMARY KEY, NOMBRE nvarchar(255) NOT NULL, HORAS int NOT NULL)");
        conec = BasesDeDatos.crearTabla(modelo.getConexion(), modelo.getDatabase(), modelo.getTable(), modelo.getValues());
        modelo.setConexion(conec);
        columnasAfectadas = "(ID, NOMBRE, HORAS)";
        values = ("('PR01', 'Desarrollo de un nuevo medicamento', 120), ('PR02', 'Análisis del cambio climático', 80), " +
            "('PR03', 'Investigación de la biodiversidad', 150), ('PR04', 'Diseño de una nueva tecnología', 100)");
        conec = BasesDeDatos.insertarValores(conec, modelo.getDatabase(), modelo.getTable(), columnasAfectadas, values);

        // Crear la tabla "asignado_a"
        modelo.setTable(tablas.get(index:2));
        modelo.setValues("(PROYECTO char(4), CIENTIFICO varchar(8), PRIMARY KEY(PROYECTO, CIENTIFICO), " +
            "FOREIGN KEY (CIENTIFICO) REFERENCES CIENTIFICOS(DNI), FOREIGN KEY (PROYECTO) REFERENCES PROYECTO(ID))");
        conec = BasesDeDatos.crearTabla(modelo.getConexion(), modelo.getDatabase(), modelo.getTable(), modelo.getValues());
        modelo.setConexion(conec);
        columnasAfectadas = "(PROYECTO, CIENTIFICO)";
        values = ("('PR01', '12345678'), ('PR01', '87654321'), ('PR02', '98765432'), ('PR02', '23456789'), " +
            "('PR03', '12345678'), ('PR03', '87654321'), ('PR04', '98765432'), ('PR04', '23456789');");
        conec = BasesDeDatos.insertarValores(conec, modelo.getDatabase(), modelo.getTable(), columnasAfectadas, values);

        int columnas = BasesDeDatos.cantidadColumnas(modelo.getConexion(), modelo.getTable());
        int filas = BasesDeDatos.cantidadFilas(modelo.getConexion(), modelo.getTable());
        ArrayList<String> nombres = BasesDeDatos.recogerNombresColumnas(modelo.getConexion(), modelo.getTable());
        Ventana ventana = new Ventana(modelo.getTable(), filas, columnas, tablas, nombres);
        ArrayList<ArrayList<String>> datosTabla = BasesDeDatos.recogerValoresTabla(conec, modelo.getTable());
        PanelTabla.actualizarTabla(datosTabla); PanelConsola.setEntradaConsola(BasesDeDatos.getTextoConsola());

        new Controlador(modelo, ventana);
    }
}
```



## Paneles para cada acción

The image displays four distinct panels for database operations, each with a title bar and standard window controls (minimize, maximize, close).

- Insertar datos:** Contains input fields for ID, NOMBRE, APELLIDO, DIRECCION, DNI, and FECHA (YYYY-MM-DD). A blue button labeled "Insertar" is at the bottom.
- Seleccionar datos:** Contains input fields for SELECT, FROM, WHERE, GROUP BY, HAVING, and ORDER BY. A blue button labeled "Select" is at the bottom.
- Actualizar datos (bottom left):** Contains input fields for UPDATE, SET, WHERE, ORDER BY, and LIMIT. A blue button labeled "Update" is at the bottom.
- Actualizar datos (bottom right):** Contains input fields for DELETE FROM, WHERE, ORDER BY, and LIMIT. A blue button labeled "Delete" is at the bottom.



- Para ver API's de java:  
[Java API](#)
- El copiloto de confianza:  
[ChatGPT](#)