



# Universidad Nacional Autónoma de México

Facultad de Estudios Superiores  
Aragón

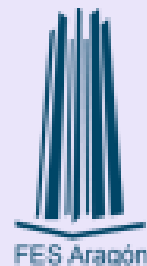
Sistemas de Información

## Primer SI

**Elaborado por:**

Juan Daniel Bermeo Espino

**Grupo: 1758 (2025-I)**



**Ejercicio. Elabore un sistema de información que reciba un nombre, y entregue un .txt donde almacene el nombre. El sistema debe de contar con una conexión a una DB y debe de poder guardar el numero de incidencias del nombre.**

Para realizar el ejercicio, decidí usar mariaDB y Java, porque son las tecnologías que tengo mas frescas en estos momentos y cuando escuché el problema lo primero que pensé fue como resolverlo con ellas.

El proyecto se realiza por medio de un proyecto en java que tiene dos archivos, la clase principal que será la que gestione todo el algoritmo y una clase conexión, que contendrá la información para conectarse a la base de datos y métodos para realizar ciertas acciones en la DB. Teniendo como resultado el siguiente código:

### Clase principal

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.SQLException;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        // Objeto scanner para leer la entrada del nombre
        Scanner sc = new Scanner(System.in);
        // Ciclo para poder introducir varios nombres de forma rapida
        while(true) {
            try {
                // Objeto conexion con el que trabajaremos la DB
                Conexion cn = new Conexion();
                // Interfaz del usuario
                System.out.println("Escribe tu nombre:");
                String nombre = sc.nextLine();
                // Variable que guardara el mensaje del txt
                String mensaje;

                // Caso 1: El nombre ya se encuentra en la DB
                if(cn.existe(nombre)) {
                    cn.aumentarIncidencia(nombre);
                    mensaje= "Hola " + nombre + "!!! por " +
cn.getIncidencias(nombre) + "° vez _-";

                    // Caso 2: El nombre es una entrada nueva
                }else {
```

```

        cn.crearRegistro(nombre);
        mensaje = "El nombre "+ nombre +" ha sido añadido a la
DB: Hola " + nombre + " !!! :3";
    }
    // Creacion del nombre del archivo asegurando irrepetibilidad
    String nombreArchivo = nombre + cn.getIncidencias(nombre) +
".txt";
    // Creacion del archivo en la carpeta del proyecto llamada
Archivos
    String pt ="Archivos/"+ nombreArchivo ;
    File archivo = new File(pt);
    // Creacion del objeto escritor
    FileWriter escribir = new FileWriter(archivo,true);
    // Se escribe el mensaje segun el caso
    escribir.write(mensaje);
    escribir.close();
    System.out.println("Su saludo ha sido guardado en: " + pt);
    cn.cerrarConexion();

} catch (ClassNotFoundException | SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

}

}
}

```

## Conexión

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Conexion {

    // Valores para hacer la conexion con mariaDB 10.11.2/ MySQL 15.1
    private String url =
"jdbc:mariadb://127.0.0.1:3306/nombres?serverTimezone = UTC";
    private String usuario = "root";
    private String pass = "1234";
    private Connection conexion;

    public Conexion() throws ClassNotFoundException, SQLException {

        Class.forName("org.mariadb.jdbc.Driver");
        conexion = DriverManager.getConnection(url,usuario,pass);
    }
}

```

```

    }

    // Devuelve verdadero si el nombre ya esta en la DB, o falso en
    caso de no estar
    public boolean existe(String nombre) throws SQLException {
        boolean existe = false;
        int indice = -1;
        String consulta = "SELECT id_nombre FROM datos WHERE nombre =
'" + nombre + "'";
        Statement st = conexion.createStatement();
        ResultSet rs = st.executeQuery(consulta);
        if(rs.next()) {
            indice = rs.getInt(1);
        }

        if(indice != -1) {
            existe = true;
        }

        return existe;
    }

    // Aumenta el valor INT que se encuentra contando el numero de
    veces en que el nombre es introducido
    public void aumentarIncidencia(String nombre) throws SQLException
    {
        String actualizar = "UPDATE datos SET incidencias = (SELECT
incidencias FROM datos WHERE nombre = ?) + 1 WHERE nombre = ?";
        PreparedStatement ps = conexion.prepareStatement(actualizar);
        ps.setString(1, nombre);
        ps.setString(2, nombre);
        ps.execute();
        ps.close();
    }

    // Crea un nuevo registro dentro de la tabla con el nombre y
    cuenta esta como su primer incidencia
    void crearRegistro(String nombre) throws SQLException {
        String insertar = "INSERT INTO datos(nombre,incidencias)
VALUES (?,1)";
        PreparedStatement ps =
conexion.prepareStatement(insertar);
        ps.setString(1, nombre);
        ps.execute();
        ps.close();
    }

    // Devuelve en INT el numero de veces que ha entrado el nombre
    public int getIncidencias(String nombre) throws SQLException {

        int incidencias = 0;
        String consulta = "SELECT incidencias FROM datos WHERE nombre
= '" + nombre + "'";
        Statement st = conexion.createStatement();
        ResultSet rs = st.executeQuery(consulta);

```

```

        rs.next();
        incidencias = rs.getInt(1);

        return incidencias;
    }

    // Cierre de la conexion a la base de datos
    public void cerrarConexion() throws SQLException {
        conexion.close();
    }
}

```

La base de datos es sencilla, simplemente necesitamos una tabla que cuente con el nombre y el número de incidencias. Todo el proceso de guardar las incidencias y registrar un nuevo nombre ya lo hace el código en Java.

```

CREATE DATABASE IF NOT EXISTS nombres;
USE nombres;

CREATE TABLE datos (
    id_nombre      INTEGER NOT NULL AUTO_INCREMENT,
    nombre         VARCHAR(30) NOT NULL,
    incidencias    INTEGER NOT NULL,

    PRIMARY KEY(id_nombre)
);

```

Finalmente el funcionamiento es el siguiente:

Paso 1. Ingresar el nombre. Esta es la interfaz del usuario, al ingresar el nombre se genera un saludo en función de la información dentro de la base de datos y nos arroja un mensaje de confirmación de la tarea realizada.











```

Escribe tu nombre:
Daniel
Su saludo ha sido guardado en: Archivos/Daniel1.txt
Escribe tu nombre:
Daniel
Su saludo ha sido guardado en: Archivos/Daniel 2.txt
Escribe tu nombre:
Daniel Bermeo
Su saludo ha sido guardado en: Archivos/Daniel Bermeo1.txt
Escribe tu nombre:
Juan
Su saludo ha sido guardado en: Archivos/Juan1.txt

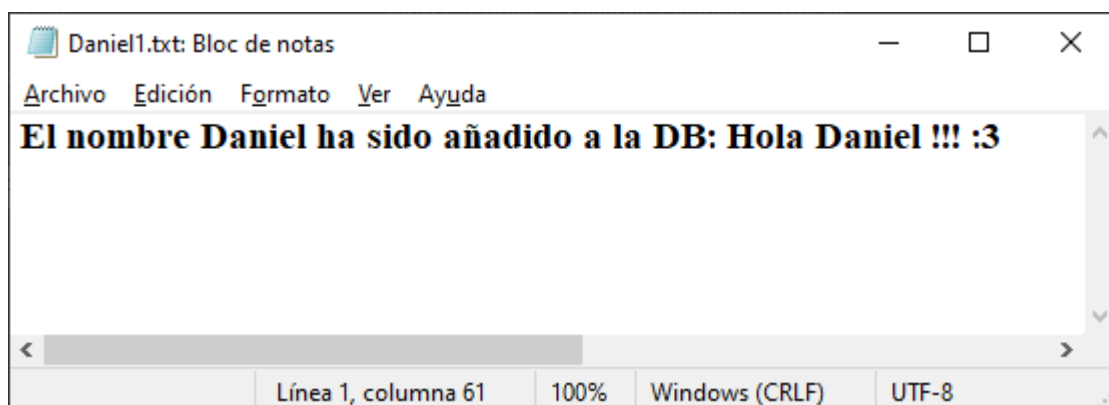
```

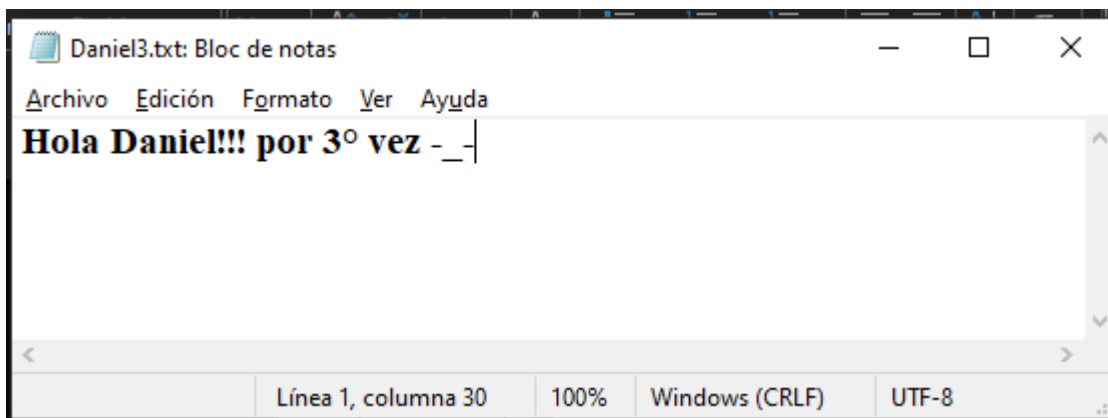
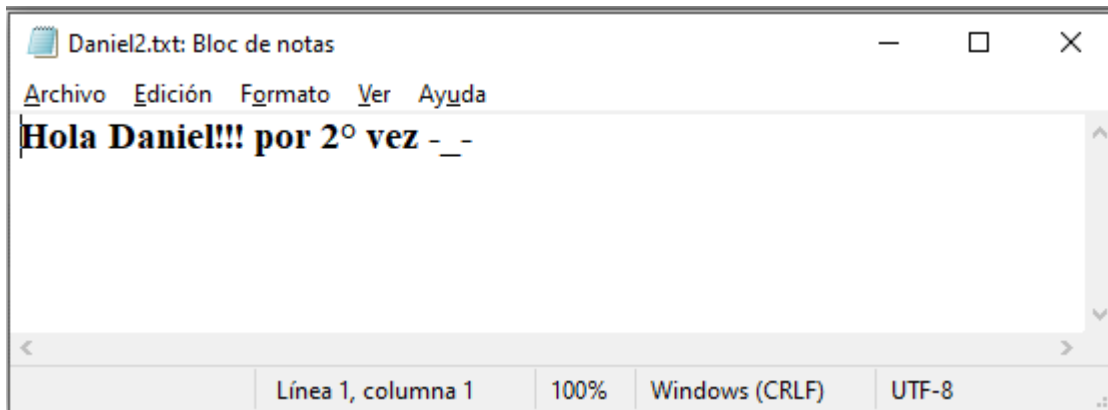
```
Escribe tu nombre:
Daniel
Su saludo ha sido guardado en: Archivos/Daniel3.txt
Escribe tu nombre:
Bermeo
Su saludo ha sido guardado en: Archivos/Bermeo1.txt
Escribe tu nombre:
Daniel
Su saludo ha sido guardado en: Archivos/Daniel4.txt
```

Finalmente, para corroborar cual es la salida de nuestro sistema vamos a abrir esos documentos que se encuentran en una carpeta dentro de nuestro proyecto, esto lo logramos usando una ruta relativa en la programación:

	Bermeo1.txt	31/08/2024 02:27 p. m.	Documento de te...	1 KB
	Daniel 2.txt	31/08/2024 02:27 p. m.	Documento de te...	1 KB
	Daniel Bermeo1.txt	31/08/2024 02:27 p. m.	Documento de te...	1 KB
	Daniel1.txt	31/08/2024 02:27 p. m.	Documento de te...	1 KB
	Daniel2.txt	31/08/2024 02:22 p. m.	Documento de te...	1 KB
	Daniel3.txt	31/08/2024 02:27 p. m.	Documento de te...	1 KB
	Daniel4.txt	31/08/2024 02:27 p. m.	Documento de te...	1 KB
	Juan Daniel1.txt	31/08/2024 02:22 p. m.	Documento de te...	1 KB
	Juan1.txt	31/08/2024 02:27 p. m.	Documento de te...	1 KB
	Juan2.txt	31/08/2024 02:22 p. m.	Documento de te...	1 KB

Podemos reconocer que el programa hace diferenciación de incidencias comparando los numero 1 al 3 de un nombre, por ejemplo, Daniel:





Y podemos definir al sistema como un sistema obediente pero apático, cual trabajador del seguro social.