

---

**Codificación de módulos del software según requerimientos del proyecto GA7-  
220501096-AA2-EV01**

Aprendiz:

JOSE MANUEL MARROQUÍN NARVAEZ

Instructor:

JESUS ANDRES SILVA PLAZAS

Análisis y Desarrollo de Software

Centro Agroempresarial y Desarrollo Pecuario del Huila

Ficha 28349953

2024

---

**Contenido**

INTRODUCCIÓN .....	3
DESARROLLO .....	4
CONCLUSIONES .....	8

---

## INTRODUCCIÓN

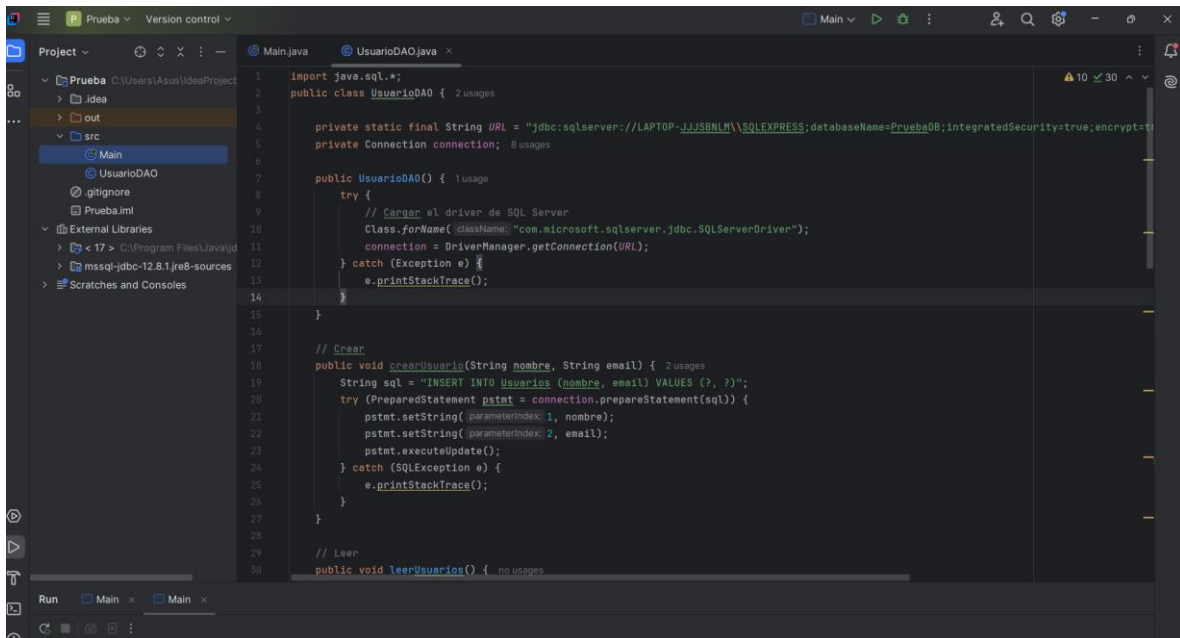
En la actualidad, la gestión de datos es un aspecto clave en el desarrollo de aplicaciones, pero para muchos, el mundo de las bases de datos puede parecer complicado y técnico. A menudo, las personas que no están familiarizadas con estos conceptos se sienten abrumadas por la terminología y los procesos involucrados. Este trabajo tiene como objetivo simplificar ese entendimiento al presentar un sistema CRUD (Crear, Leer, Actualizar y Eliminar) utilizando Java y SQL Server.

A través de un ejemplo práctico con una tabla de usuarios, se explorarán las operaciones básicas necesarias para interactuar con una base de datos de manera efectiva. Al desglosar estos conceptos de forma clara y accesible, se busca no solo ilustrar la importancia de la gestión de datos, sino también inspirar a aquellos que están comenzando su viaje en el desarrollo de software. Con este enfoque, esperamos que más personas puedan apreciar la relevancia y la aplicabilidad de estas herramientas en el mundo digital actual.

## DESARROLLO

Para el desarrollo de esta actividad se tuvo en cuenta descargar las librerías correspondientes al SGBD que para este caso se hizo relacionado a SQL Server, donde por medio de la página oficial se descargaron los drivers.

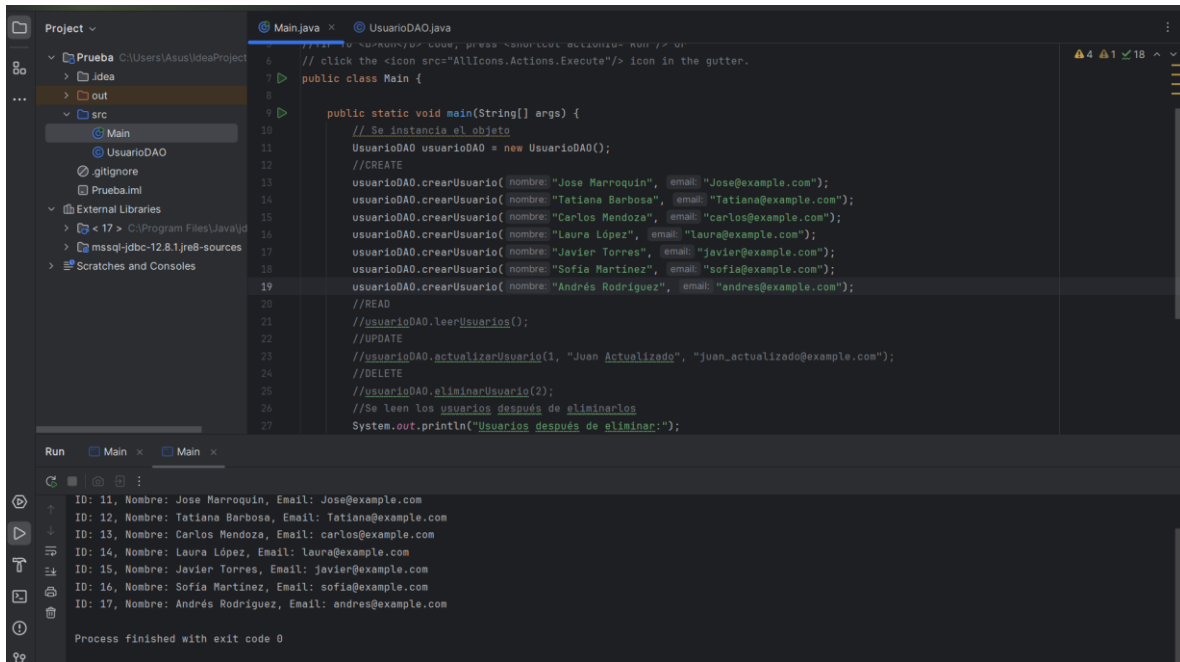
Después se proceden a crear las conexiones correspondientes en un clase de Java que se llama usuarioDAO, donde se debe configurar la URL de la conexión al servidor local, donde también por medio de los métodos se codifican los procesos CRUD (Create, Read, Update y Delete) cada método lleva nombre de crearUsuario(), leerUsuario(), actualizarUsuario() y eliminarUsuario() por último se crea un método para cerrar la conexión.



```
1 import java.sql.*;
2 public class UsuarioDAO { 2 usages
3
4     private static final String URL = "jdbc:sqlserver://LAPTOP-JJJSBNLM\\SQLSERVER;databaseName=PruebaDB;integratedSecurity=true;encrypt-t
5     private Connection connection; 8 usages
6
7     public UsuarioDAO() { 1 usage
8         try {
9             // Cargar el driver de SQL Server
10            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
11            connection = DriverManager.getConnection(URL);
12        } catch (Exception e) {
13            e.printStackTrace();
14        }
15    }
16
17    // Crear
18    public void crearUsuario(String nombre, String email) { 2 usages
19        String sql = "INSERT INTO Usuarios (nombre, email) VALUES (?, ?)";
20        try (PreparedStatement pstmt = connection.prepareStatement(sql)) {
21            pstmt.setString(1, nombre);
22            pstmt.setString(2, email);
23            pstmt.executeUpdate();
24        } catch (SQLException e) {
25            e.printStackTrace();
26        }
27    }
28
29    // Leer
30    public void leerUsuarios() { no usages
```

Después de haber configurado los métodos en el Main se ejecutan las acciones de los procesos CRUD de cada método de la clase usuarioDAO

## CREATE



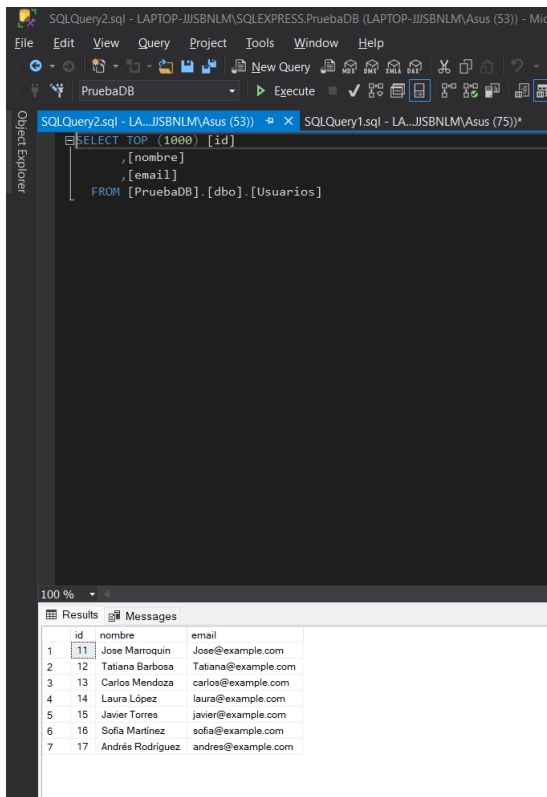
```
// click the <icon src="AllIcons.Actions.Execute"/> icon in the gutter.
public class Main {

    public static void main(String[] args) {
        // Se instancia el objeto
        UsuarioDAO usuarioDAO = new UsuarioDAO();
        //CREATE
        usuarioDAO.crearUsuario( nombre: "Jose Marroquin", email: "Jose@example.com");
        usuarioDAO.crearUsuario( nombre: "Tatiana Barbosa", email: "Tatiana@example.com");
        usuarioDAO.crearUsuario( nombre: "Carlos Mendoza", email: "carlos@example.com");
        usuarioDAO.crearUsuario( nombre: "Laura López", email: "laura@example.com");
        usuarioDAO.crearUsuario( nombre: "Javier Torres", email: "javier@example.com");
        usuarioDAO.crearUsuario( nombre: "Sofia Martinez", email: "sofia@example.com");
        usuarioDAO.crearUsuario( nombre: "Andrés Rodriguez", email: "andres@example.com");
        //READ
        //usuarioDAO.leerUsuarios();
        //UPDATE
        //usuarioDAO.actualizarUsuario(1, "Juan Actualizado", "juan_actualizado@example.com");
        //DELETE
        //usuarioDAO.eliminarUsuario(2);
        //Se leen los usuarios después de eliminarlos
        System.out.println("Usuarios después de eliminar:");
    }
}
```

Run Main Main

ID: 11, Nombre: Jose Marroquin, Email: Jose@example.com  
ID: 12, Nombre: Tatiana Barbosa, Email: Tatiana@example.com  
ID: 13, Nombre: Carlos Mendoza, Email: carlos@example.com  
ID: 14, Nombre: Laura López, Email: laura@example.com  
ID: 15, Nombre: Javier Torres, Email: javier@example.com  
ID: 16, Nombre: Sofia Martinez, Email: sofia@example.com  
ID: 17, Nombre: Andrés Rodriguez, Email: andres@example.com

Process finished with exit code 0



```
SELECT TOP (1000) [id]
      ,[nombre]
      ,[email]
FROM [PruebaDB].[dbo].[Usuarios]
```

Results Messages

	id	nombre	email
1	11	Jose Marroquin	Jose@example.com
2	12	Tatiana Barbosa	Tatiana@example.com
3	13	Carlos Mendoza	carlos@example.com
4	14	Laura López	laura@example.com
5	15	Javier Torres	javier@example.com
6	16	Sofia Martinez	sofia@example.com
7	17	Andrés Rodriguez	andres@example.com

## READ

```
//READ
usuarioDAO.leerUsuarios();
//UPDATE
```

```
31 }
32 }
33 }

Run Main x Main x

"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\In
ID: 11, Nombre: Jose Marroquin, Email: Jose@example.com
ID: 12, Nombre: Tatiana Barbosa, Email: Tatiana@example.com
ID: 13, Nombre: Carlos Mendoza, Email: carlos@example.com
ID: 14, Nombre: Laura López, Email: laura@example.com
ID: 15, Nombre: Javier Torres, Email: javier@example.com
ID: 16, Nombre: Sofía Martínez, Email: sofia@example.com
ID: 17, Nombre: Andrés Rodríguez, Email: andres@example.com

Process finished with exit code 0
```

## UPDATE

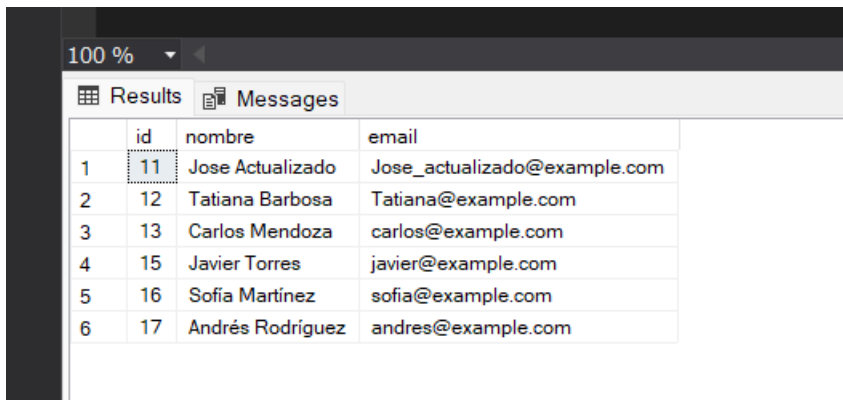
```
//UPDATE
usuarioDAO.actualizarUsuario( id: 11, nuevoNombre: "Jose Actualizado", nuevoEmail: "Jose.actualizado@example.com");
```

Results			
	id	nombre	email
1	11	Jose Actualizado	Jose_actualizado@example.com
2	12	Tatiana Barbosa	Tatiana@example.com
3	13	Carlos Mendoza	carlos@example.com
4	14	Laura López	laura@example.com
5	15	Javier Torres	javier@example.com
6	16	Sofia Martínez	sofia@example.com
7	17	Andrés Rodríguez	andres@example.com

## DELETE

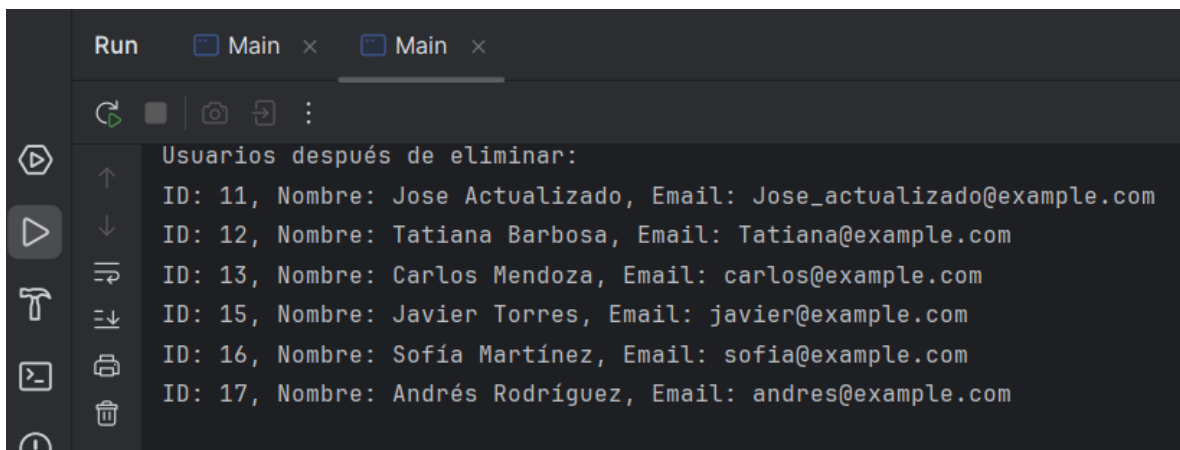
En este caso se elimina el id 14 que es Laura López registro 4

```
//DELETE
usuarioDAO.eliminarUsuario( id: 14);
```



	id	nombre	email
1	11	Jose Actualizado	Jose_actualizado@example.com
2	12	Tatiana Barbosa	Tatiana@example.com
3	13	Carlos Mendoza	carlos@example.com
4	15	Javier Torres	javier@example.com
5	16	Sofía Martínez	sofia@example.com
6	17	Andrés Rodríguez	andres@example.com

Se hace nuevamente la lectura de la base desde IntelliJ



```
Run Main x Main x
Usuarios después de eliminar:
ID: 11, Nombre: Jose Actualizado, Email: Jose_actualizado@example.com
ID: 12, Nombre: Tatiana Barbosa, Email: Tatiana@example.com
ID: 13, Nombre: Carlos Mendoza, Email: carlos@example.com
ID: 15, Nombre: Javier Torres, Email: javier@example.com
ID: 16, Nombre: Sofía Martínez, Email: sofia@example.com
ID: 17, Nombre: Andrés Rodríguez, Email: andres@example.com
```

---

## CONCLUSIONES

La implementación del sistema CRUD para la gestión de usuarios demuestra la eficacia de las bases de datos relacionales en la administración de información. A través del uso de JDBC, se logra una conexión robusta y eficiente entre la aplicación Java y SQL Server, permitiendo realizar operaciones fundamentales de manera sencilla y clara.

Este ejercicio no solo proporciona una base sólida para comprender los principios de la manipulación de datos, sino que también establece un punto de partida para el desarrollo de aplicaciones más complejas. En un entorno donde la información es un recurso valioso, dominar estas técnicas es esencial para cualquier desarrollador. Las lecciones aprendidas en este trabajo son aplicables a una amplia variedad de contextos, subrayando la relevancia del aprendizaje continuo en el campo del desarrollo de software.