

**EVIDENCIA DE APRENDIZAJE 1: CRUD DAO Y MANEJO DE EXCEPCIONES EN JAVA**

**PRESENTADO POR**

**ADRIANYS MARIA SAUMETH PADILLA**

**INSTITUCIÓN UNIVERSITARIA DIGITAL DE ANTIOQUIA**

**ASIGNATURA: DESARROLLO DE SOFTWARE SEGURO**

## **YEILER ALBERTO QUINTERO BARCO**

**05/06/2025**

# **INTRODUCCIÓN**

El presente documento tiene como finalidad exponer el desarrollo completo de una aplicación de escritorio implementada en Java, que realiza operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los datos de funcionarios, siguiendo el patrón DAO (Data Access Object) y empleando buenas prácticas en el manejo de excepciones. La actividad se fundamenta en un caso de estudio propuesto por el curso, centrado en la gestión del personal de la Institución Universitaria Digital de Antioquia.

# **Desarrollo**

## **Modelo relacional**

Se definió la tabla principal 'funcionarios', con los campos: id\_funcionario, nombres, apellidos, dirección y teléfono. Esta estructura permite almacenar la información básica del personal de manera normalizada.

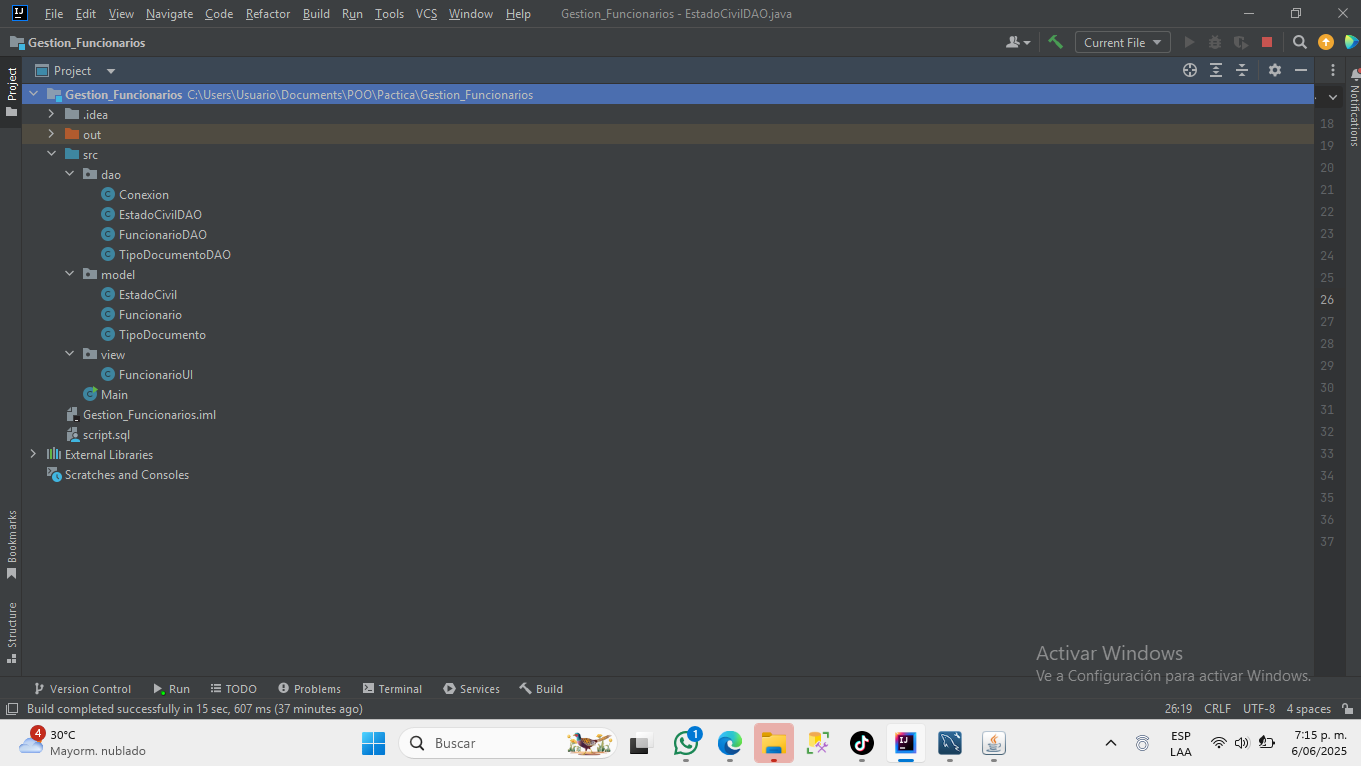
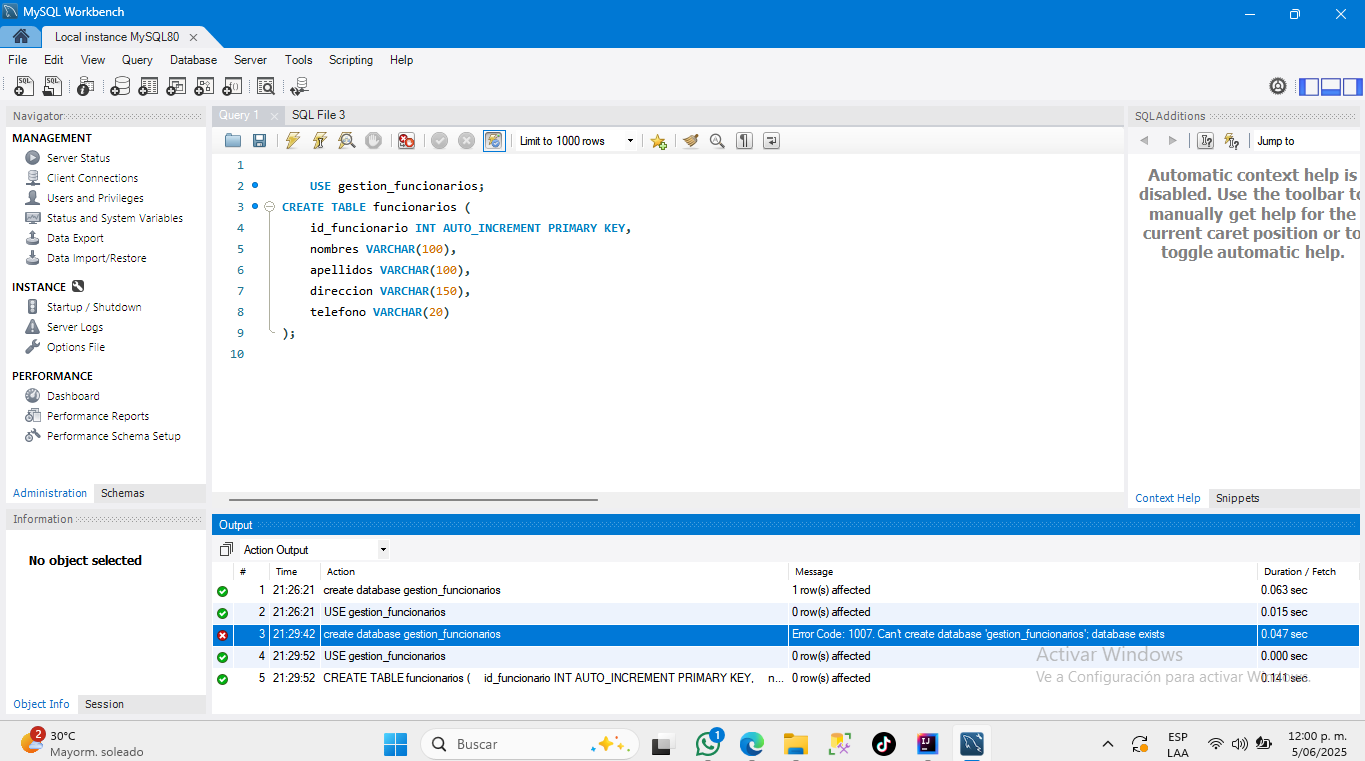
## **Script de base de datos**

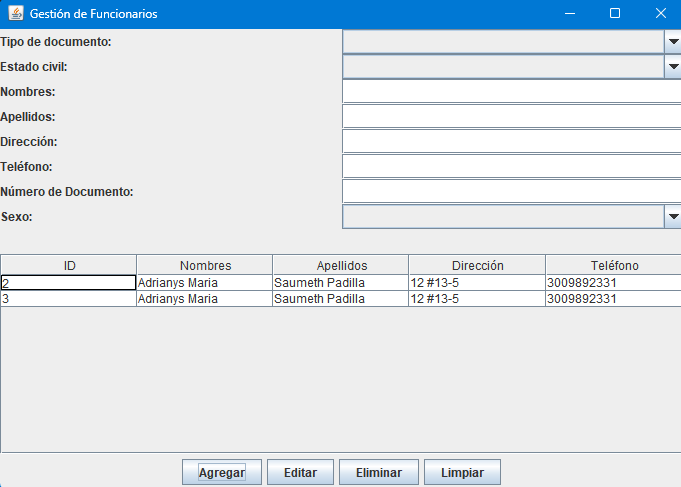
La base de datos fue creada en MySQL bajo el nombre 'gestion\_funcionarios'. A continuación, se muestra el script utilizado:  
  
CREATE DATABASE gestion\_funcionarios;  
USE gestion\_funcionarios;  
  
CREATE TABLE funcionarios (  
 id\_funcionario INT AUTO\_INCREMENT PRIMARY KEY,  
 nombres VARCHAR(100),  
 apellidos VARCHAR(100),  
 direccion VARCHAR(150),  
 telefono VARCHAR(20)  
);

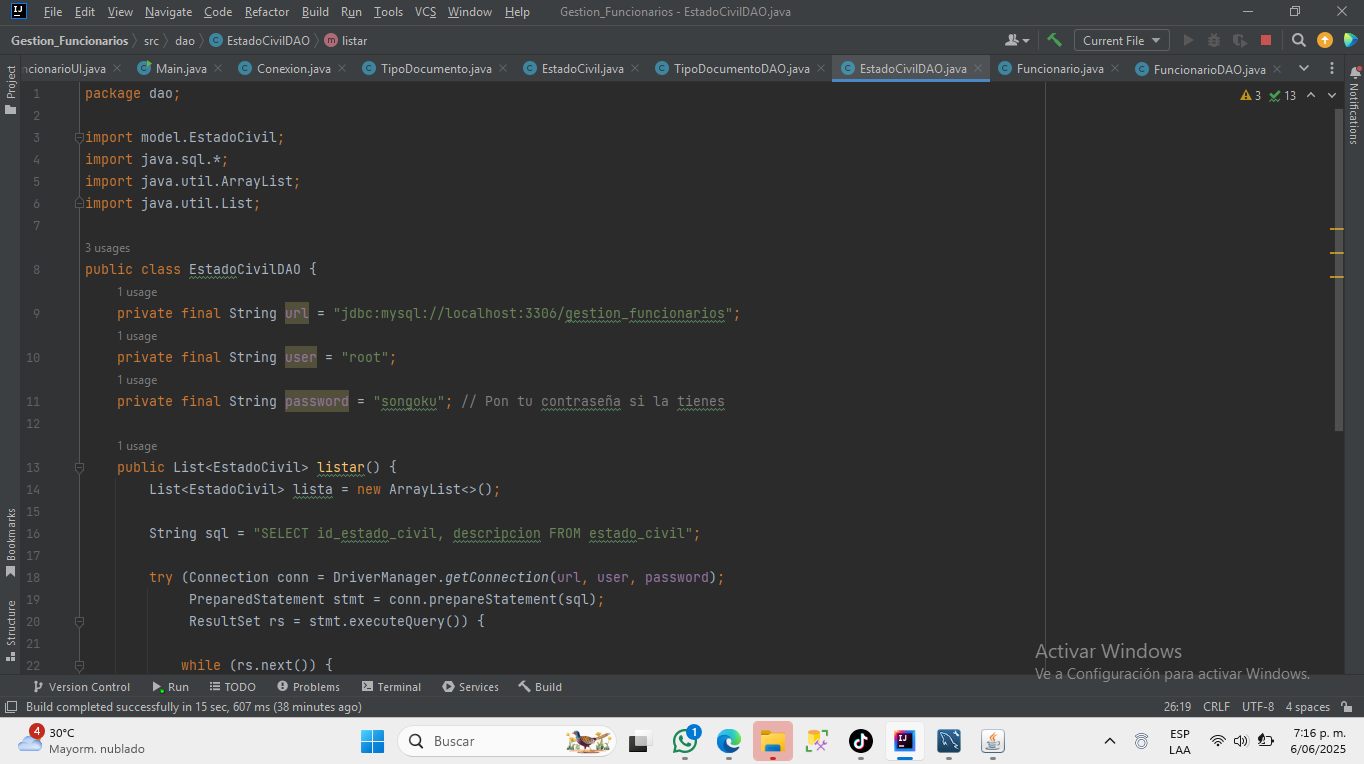
## **Estructura del proyecto Java**

El proyecto fue desarrollado en IntelliJ IDEA utilizando Java 17. La estructura incluye los siguientes paquetes:  
- model: contiene la clase Funcionario.java  
- dao: contiene Conexion.java y FuncionarioDAO.java  
- view: contiene la interfaz gráfica FuncionarioUI.java  
- Main.java: clase principal que lanza la aplicación

## **Capturas de pantalla**

A continuación, se ilustran algunas pantallas clave del sistema en funcionamiento:





## **Manejo de excepciones**

Se emplean bloques try-catch en todas las operaciones de base de datos para capturar y gestionar excepciones del tipo SQLException. Esto garantiza robustez en la ejecución, evitando que errores de conexión o sentencias mal formuladas interrumpan el flujo de la aplicación.

# **Conclusión**

La actividad permitió aplicar de forma práctica conceptos fundamentales de programación orientada a objetos en Java, así como el patrón DAO y la gestión de errores mediante excepciones. La implementación del CRUD sobre la tabla de funcionarios evidencia la correcta estructuración del código, el uso de JDBC y el diseño modular del sistema. Esta experiencia refuerza la capacidad de construir aplicaciones escalables, seguras y funcionales en entornos reales.