

**UNIVERSIDAD PERUANA LOS ANDES**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE**  
**SISTEMAS Y COMPUTACIÓN**



---

**PROYECTO**

---

**CURSO: TALLER VII: DESARROLLO DE APLICACIONES I**

**DATOS DOCENTE: FERNANDEZ BEJARANO RAUL ENRIQUE**

**ESTUDIANTE: NONALAYA MAYTA LENIN ILDER**

**CICLO: IV**  
**SECCION: A1**

**2025**  
**HUANCAYO**

# Sistema de Gestión de Usuarios con Base de Datos SQLite

Proyecto: SistemaGestionUsuarios

Tecnología Principal: Java con Maven

Base de Datos: SQLite ( tareas.db)

Versión de Java: 22

Presentado por: (Tu Nombre/Equipo)

Fecha: Octubre 2025

## 1. Resumen Ejecutivo

El proyecto **SistemaGestionUsuarios** es una aplicación desarrollada en **Java** utilizando la herramienta de gestión **Maven**. Su objetivo principal es implementar un **módulo fundamental de gestión de usuarios** con persistencia de datos en una base de datos de llamada **SQLite** tareas.db. Este sistema establece un modelo básico de control de acceso, almacenando credenciales ( nombre\_usuario y contraseña) y un **rol** específico ( Estudiante o Docente) para cada usuario. La configuración base utiliza la versión **Java 22**.

## 2. Introducción

La gestión de usuarios y la diferenciación de funcionalidades son esenciales en cualquier plataforma moderna. Este proyecto establece un módulo robusto de **autenticación y autorización** como base. Al utilizar **Maven**, se asegura un entorno de desarrollo estandarizado y eficiente, mientras que la base de datos **SQLite** ( tareas.db) garantiza una solución de persistencia de datos ligera y portátil.

### 2.1. Objetivo del Proyecto

El objetivo es construir la **infraestructura de gestión de usuarios** para el proyecto SistemaGestionUsuarios, que se enfoca en:

1. Configurar el proyecto en **Maven**, especificando el artifactId y la versión de **Java 22**.
2. Crear una estructura de datos persistentes en **SQLite** para la gestión de usuarios.
3. Diseñar e implementar la tabla usuarios para manejar credenciales y el **control de acceso basado en roles** (Docente/Estudiante).

## 3. Desarrollo del Proyecto

### 3.1. Metodología

El desarrollo se centró inicialmente en la **configuración del entorno** y el **diseño del modelo de datos**, siguiendo estos pasos clave:

1. **Configuración del entorno** : Se definió el archivo `pom.xml` con el `artifactId SistemaGestionUsuarios` y se desarrolló la versión **Java 22** para la compilación. La clase principal de ejecución se encuentra en el paquete `com.mycompany.tarea_02`.
2. **Diseño de la Base de Datos** : Se creó la base de datos `tareas.db` en formato **SQLite** .
3. **Definición de esquema** : Se implementó la tabla `usuarios` con las columnas necesarias para soportar el inicio de sesión y la autorización.
4. **Carga de Datos Iniciales** : Se insertaron registros de prueba esenciales, incluyendo un usuario con rol `Estudiante` y otro con rol `Docente`, para validar el sistema de roles.

### 3.2. Estructura de La Base de Datos y Resultados

#### 3.2.1. Configuración de Maven y Paquetes

La configuración principal del proyecto es la siguiente:

- **ID de artefacto** : `SistemaGestionUsuarios`
- **Versión de Java** : 22
- **Paquetes (Inferidos)** : El punto de entrada está en `com.mycompany.tarea_02.Tarea_02`, indicando el uso de la estructura de paquetes de Java.

#### 3.2.2. Diseño de la Tabla de Usuarios

La tabla `usuarios` en `tareas.db` define la estructura del módulo de seguridad:

| Columna                      | Tipo de Dato | Restricciones                | Propósito                                  |
|------------------------------|--------------|------------------------------|--|
| <code>id</code>              | INTEGER      | PRIMARY KEY<br>AUTOINCREMENT | Identificador único.                       |
| <code>nombre_usuario</code>  | TEXT         | NOT NULL                     | Nombre utilizado para el inicio de sesión. |
| <code>contraseña</code>      | TEXT         | NOT NULL                     | Clave de acceso.                           |
| <code>nombre_completo</code> | TEXT         |                              | Nombre y apellido del usuario.             |
| <code>rol</code>             | TEXT         | NOT NULL                     | Defina el nivel de acceso (autorización) . |

#### 3.2.3. Datos Iniciales y Roles

La base de datos contiene datos precargados que confirman la implementación del sistema de roles:

- Un usuario con el rol de **Estudiante**.
- Un usuario con el rol de **Docente**.

La aplicación deberá utilizar estos roles para cargar los paneles y funcionalidades específicas para cada tipo de usuario tras el proceso de autenticación.

### 3.3. Paneles y elementos de interfaz (GUI)

El proyecto, al ser una aplicación de gestión, requiere una interfaz gráfica construida con Paneles (probablemente usando Swing o JavaFX). Los paneles y los botones funcionales asociados se definen para implementar el flujo del sistema:

#### Paneles Principales (Clases de Interfaz)

| Nombre de Panel<br>(Clase Inferida) | Propósito y Función   | Roles que Acceden                   |
|-------------------------------------|---|-------------------------------------|
| LoginPanel                          | Permite a los usuarios ingresar sus credenciales ( nombre_usuario y contraseña) para autenticarse contra la tabla usuarios. | Todos.                              |
| MainFrame                           | Es la ventana contenedora principal de la aplicación.   | Todos<br>(tras iniciar sesión)<br>. |
| PanelDocente                        | Contiene las herramientas y vistas específicas para la gestión de tareas, cursos o administración.                          | Docente<br>.                        |
| PanelEstudiante                     | Contiene las herramientas y vistas específicas para visualizar tareas asignadas, notas o enviar trabajos.                   | Estudiante.                         |

#### Botones Clave (Funcionalidades)

| Botón                   | Ubicación (Panel)                 | Función principal  |
|-------------------------|-----------------------------------|--|
| JButton_Ingresar        | LoginPanel                        | Inicia el proceso de autenticación, consultando la base de datos tareas.db.                          |
| JButton_Salir           | MainFrame                         | Cierra la aplicación o la sesión actual.   |
| JButton_CrearUsuario    | PanelDocente<br>(o Administrador) | Permite la inserción de nuevos registros en la tabla usuarios.                                       |
| JButton_GestionarTareas | PanelDocente                      | Acceso a las funcionalidades CRUD (Crear, Leer, Actualizar, Eliminar) para la gestión de las tareas. |
| JButton_VerTareas       | PanelEstudiante                   | Consulta las tareas asignadas al usuario Estudiante.   |

## 4. Conclusiones

El proyecto **SistemaGestionUsuarios** ha logrado establecer satisfactoriamente la arquitectura tecnológica y el modelo de datos para la gestión de usuarios, empleando **Java 22, Maven y SQLite**. La correcta definición de la tabla **usuarios**, en particular, del campo **rol**, es el resultado más importante, ya que asegura la capacidad del sistema para manejar la **autorización basada en roles**. La definición de los **Paneles y Botones** complementa esta estructura, delineando el flujo de la Interfaz Gráfica (GUI) desde el **LoginPanel** hasta la gestión específica por rol (**PanelDocente** y **PanelEstudiante**). Este nacimiento es robusto y permite avanzar inmediatamente al desarrollo de la capa de negocio y la implementación de la GUI funcional.

