

Documentación Técnica

Aplicación de Solicitudes Académicas

Desarrollado por: Jose Felipe Carvajal Campos

Fecha: 12/06/2025

Tecnología usada: JavaFX + SQL server

Objetivo: Aplicación de escritorio para gestión de solicitudes de estudiantes

1. Descripción general del Sistema.

Esta aplicación de escritorio fue desarrollada en **JavaFX** con conexión a **SQL server** mediante **JDBC**. Su objetivo principal es permitir a estudiantes universitarios a registrar solicitudes administrativas, como peticiones a las áreas de registro, bienestar y biblioteca, entre otras.

El sistema permite:

- Inicio de sesión de estudiantes con validación en base de datos.
- Registro de estudiante.
- Registro de solicitudes con tipo y descripción.
- Visualización de solicitudes personales registradas.
- Visualización de grafica en forma de pastel del total de solicitudes por tipo.

La aplicación cuenta con un diseño limpio he intuitivo con el usuario para facilitar su uso.

2. Tecnologías usadas.

- Java17 (soporte para JavaFX).
- JavaFX SDK 17.0.2: Interfaz gráfica.
- SQL server 2022: Base de datos relacional.
JDBC (Java Database Connectivity): Conexión a la base de datos.
- NetBeans IDE 24: Entorno de desarrollo.
- Piechart: Gráficos para la visualización de estadísticas.

3. Modelo de datos.

El aplicativo trabaja con dos tablas principales en la base de datos:

Tabla 'usuarios'

Esta tabla almacena la información de cada estudiante que se registra en el sistema.

Campo	Tipo dato	Descripción
id	INT (PK)	Identificación de usuario
usuario	VARCHAR(50)	Nombre de usuario único
contrasena	VARCHAR(100)	Contraseña de usuario
nombre_completo	VARCHAR(100)	Nombre completo de usuario

Tabla 'solicitudes'

Esta tabla almacena las solicitudes realizadas por los usuarios registrados.

Campo	Tipo dato	Descripción
id	INT (PK)	Identificación de la solicitud
id_usuario	INT (FK)	Relación con el ID de la tabla 'usuarios'
tipo_solicitud	VARCHAR(50)	Tipo de solicitud (Registro, Bienestar, Biblioteca)
descripcion	TEXT	Detalles de la solicitud descrita por el estudiante
fecha	DATETIME	Fecha y hora en la que se realizó la solicitud

Relación entre las tablas

Un usuario puede tener muchas solicitudes

Cada solicitud pertenece a un solo usuario

Esta relación se consigue mediante la **llave foránea (FOREIGN KEY)** que está en la tabla solicitudes 'id_usuario', conectada a la **llave primaria (PRIMARY KEY)** en la tabla de usuarios 'id'.

4. Funcionalidades del sistema.

La aplicación está dividida en varias pantallas, cada una con funciones específicas. A continuación, se mostrarán sus principales funciones:

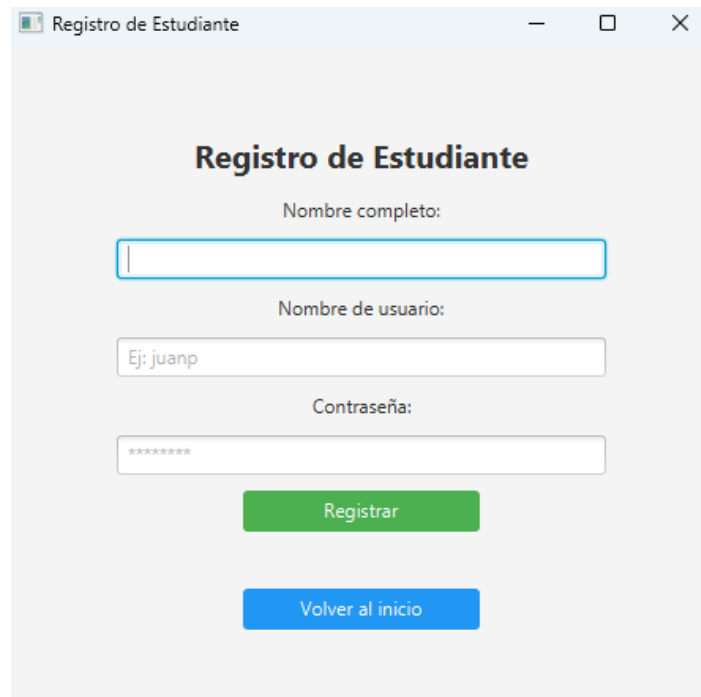
4.1 Registro de estudiante

Permite que cualquier persona se registre como estudiante ingresando:

- Nombre completo
- Nombre de usuario
- Contraseña

Los datos se guardan en la tabla 'usuarios'.

Si el nombre del usuario ya existe arrojará un mensaje de advertencia.



Registro de Estudiante

Nombre completo:

Nombre de usuario:

Contraseña:

Registrar

Volver al inicio

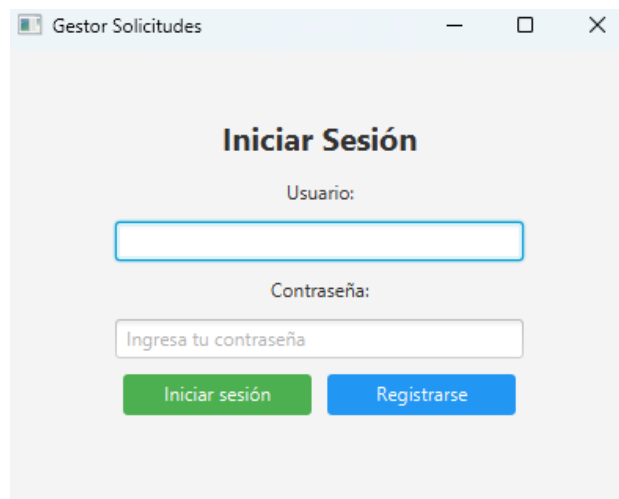
This is a screenshot of a web application window titled "Registro de Estudiante". It features a light gray background. At the top, the title "Registro de Estudiante" is centered in a bold black font. Below the title, there are three input fields: "Nombre completo:" (empty), "Nombre de usuario:" (containing "Ej: juanp"), and "Contraseña:" (containing "*****"). Below these fields are two buttons: a green "Registrar" button and a blue "Volver al inicio" button.

4.2 Inicio de sesión

Permite al estudiante iniciar sesión con su nombre de usuario y su contraseña.

El aplicativo valida la existencia en la base de datos.

Si las credenciales son correctas, el aplicativo le dará acceso al formulario de solicitudes.



Gestor Solicitudes

Iniciar Sesión

Usuario:

Contraseña:

Iniciar sesión Registrarse

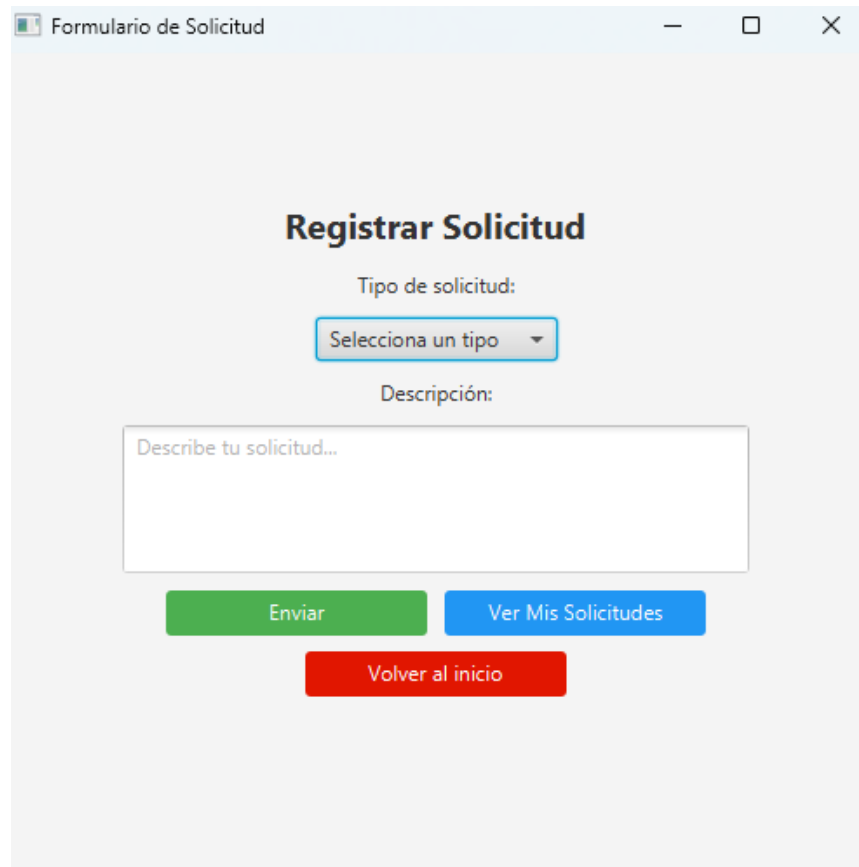
This is a screenshot of a web application window titled "Gestor Solicitudes". It features a light gray background. At the top, the title "Gestor Solicitudes" is centered in a bold black font. Below the title, there is a section titled "Iniciar Sesión" in bold black font. Underneath, there are two input fields: "Usuario:" (empty) and "Contraseña:" (containing "Ingresa tu contraseña"). Below these fields are two buttons: a green "Iniciar sesión" button and a blue "Registrarse" button.

4.3 Registro de solicitud

El estudiante puede registrar una solicitud nueva cada vez que ingrese seleccionando:

- Tipo de solicitud (Lista desplegable con opción Bienestar, Registro, Biblioteca, otro).
- Descripción de solicitud.

La solicitud se almacena en la base de datos y se asocia al estudiante que la creo con fecha y hora.



Formulario de Solicitud

Registrar Solicitud

Tipo de solicitud:

Selecciona un tipo ▼

Descripción:

Describe tu solicitud...

Enviar Ver Mis Solicitudes

Volver al inicio

4.4 Visualización de solicitudes

El estudiante puede verificar en un listado las solicitudes que realizo.

Se muestra una tabla con:

- Tipo
- Descripción
- Fecha

4.6 Navegación de pantallas

Desde cualquier vista el usuario puede regresar a la ventana anterior con el botón **cerrar**.

Puede regresar al inicio de sesión mediante el botón **volver al inicio** en solicitud.

5. Estructura de proyecto

El proyecto está organizado en una estructura sencilla con las siguientes clases principales:

- ConexionDB.java – Maneja la conexión con SQL server
- LoginView.java – Pantalla de inicio de sesión
- RegistroEstudianteView.java – Registro de usuarios nuevos
- RegistroSolicitudView.java – Registro solicitudes
- ListadoSolicitudesView.java – Muestra solicitudes del usuario
- GraficoSolicitudesView.java – Muestra gráfico global
- Solicitud.java

Cada clase muestra una vista o un módulo funcional del sistema, respetando la responsabilidad de cada una.

6. Configuración de la base de datos.

Para ejecutar correctamente el sistema, se debe tener instalado la base de datos **SQL server** y crear una base de datos llamada **UniversidadDB**.

Tablas necesarias:

Tabla: usuarios

```
CREATE TABLE usuarios (  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    usuario VARCHAR(50) UNIQUE NOT NULL,  
    contrasena VARCHAR(100) NOT NULL,  
    nombre_completo VARCHAR(100)  
);
```

Tabla: solicitud

```
CREATE TABLE solicitudes (  
    id INT IDENTITY(1,1) PRIMARY KEY,  
    id_usuario INT,  
    tipo_solicitud VARCHAR(50),  
    descripcion TEXT,  
    fecha DATETIME DEFAULT GETDATE(),  
    FOREIGN KEY (id_usuario) REFERENCES usuarios(id)  
);
```

En la clase **ConexionDB**, se debe hacer la conexión así:

```
String url =  
"jdbc:sqlserver://localhost:1433;databaseName=UniversidadDB;encrypt=true;trustServerCertificate=true";  
  
String user = "admin";  
  
String password = "1234";
```

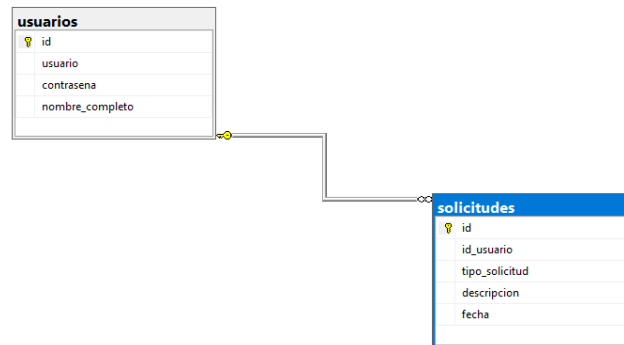
El usuario y contraseña debe ser con la que se inicia sesión en la base de datos SQL server.

7. Instrucciones de ejecución.

1. Abrir el proyecto en NetBeans.
2. Asegurarse de tener SQL server corriendo y la base de datos creada.
3. Verificar la clase '**ConexionDB.java**' tenga el usuario y la contraseña correcta.
4. Ejecutar el proyecto desde la clase principal '**MainApp.java**'.

Sugerencia: Asegurarse de tener añadido el driver JDBC de SQL server en las bibliotecas del proyecto.

8. Diagrama simple de entidades.

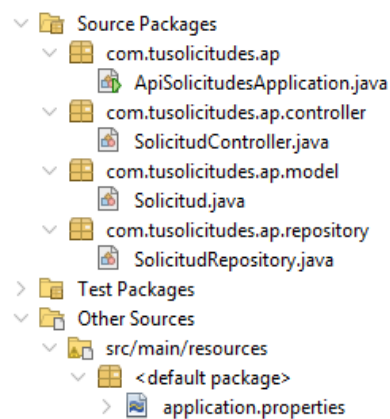


9. Descripción del código .

- **LoginView.java**: muestra el login, valida usuario/contraseña y redirige.
- **RegistroEstudianteView.java**: permite registrar nuevos usuarios en la tabla `usuarios`.
- **RegistroSolicitudView.java**: muestra un formulario para registrar solicitudes con tipo y descripción.
- **ListadoSolicitudesView.java**: presenta las solicitudes del usuario en una tabla.
- **GraficoSolicitudesView.java**: genera un gráfico circular que muestra cuántas solicitudes hay por tipo.
- **ConexionDB.java**: centraliza la conexión a la base de datos SQL Server.
- **Solicitud.java**: clase modelo que representa cada solicitud con tipo, descripción y fecha.

10. API REST de solicitudes.

1. Estructura del proyecto.



2. Configuración con la Base de Datos.

'Application.properties'

```
spring.datasource.url=jdbc:sqlserver://localhost:1433;databaseName=UniversidadDB;encrypt=true;trustServerCertificate=true
spring.datasource.username=sa
spring.datasource.password=tu_contrasena

spring.jpa.hibernate.ddl-auto=none
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.SQLServerDialect
```

Usa SQL server como gestor de base de datos.

No genera ni modifica tablas automáticamente con (**ddl-auto=none**).

3. Instrucciones de ejecución.

- **Abrir el proyecto en NetBeans.**
- Verificar que la base de datos **UniversidadDB** este activa.
- Ejecuta la aplicación con clic derecho en la clase principal -> run file.
- Acceder a los endpoints mediante el navegador o postman.

4. Endpoints disponibles.

GET /api/solicitudes/{usuario}

Devuelve un JSON con todas las solicitudes que le pertenecen al usuario.

Ejemplo:

GET <http://localhost:8080/api/solicitudes/admin>

Respuesta:

```
[  
  {  
    "id": 1,  
    "idUser": 2,  
    "tipoSolicitud": "Registro",  
    "descripcion": "Actualizar datos de matrícula",  
    "fecha": "2025-06-10T14:00:00"  
  },  
  ...  
]
```

5. Descripción del código.

Solicitud.java – Representa la tabla solicitud en la base de datos

SolicitudRepository.java – Permite buscar solicitudes mediante el Id del usuario

SolicitudController.java – Expone el endpoint REST y consulta la base de datos para obtener el id del usuario