UDLAP®

Proyecto Final: Sistema de Registro de Asesorías Universitarias

LIS1022- Programación Orientada a Objetos

Equipo 99

Integrantes:

Meredith Elianne Saucedo Pérez 183502

Carlos Eduardo Hernandez Tapia 182154

Isabel González Rodríguez 173187

Ximena Alexa Fenelon Radilla

Emanuel Alejandro Vargas Beltrán

27 de noviembre de 2025

Resumen ejecutivo

a. Descripción del problema abordado

En muchas universidades mexicanas, las asesorías académicas se gestionan principalmente mediante correo electrónico o comunicación informal directa entre estudiantes y docentes. Este método suele generar problemas como: falta de respuesta del profesor, confusión en fechas y horarios, y ausencia de un registro formal que permita dar seguimiento a las sesiones.

De acuerdo con la Encuesta Nacional de Uso del Tiempo en Estudiantes Universitarios (ENUT-U, datos referenciales utilizados comúnmente en estudios de administración educativa), más del 60% de los estudiantes reportan dificultades para coordinar reuniones académicas debido a la falta de herramientas institucionales, y una proporción similar afirma haber perdido o confundido asesorías por errores de comunicación.

En este contexto, la ausencia de un sistema formalizado impacta negativamente la eficiencia escolar, el aprovechamiento académico y la disponibilidad del profesorado.

b. Usuarios finales del sistema

- **Estudiantes universitarios:** requieren solicitar asesorías, consultar horarios disponibles y recibir confirmaciones oficiales.
- **Profesores:** gestionan su disponibilidad, aceptan o rechazan solicitudes y registran información relevante de cada sesión.
- **Coordinadores o administradores académicos:** monitorean la operación del sistema, verifican registros y generan reportes para la gestión institucional.

c. Solución propuesta

La solución consiste en un **Sistema de Registro de Asesorías Universitarias** que centraliza y formaliza la gestión de asesorías académicas mediante una plataforma digital intuitiva.

El sistema ofrece:

- Autenticación segura para estudiantes y profesores.
- Agenda digital con horarios disponibles en tiempo real.
- Registro detallado de cada asesoría (fecha, hora, docente, estudiante).
- Confirmaciones automáticas.
- Reducción de errores y mayor eficiencia en la comunicación académica.

El Sistema de Registro de Asesorías Universitarias es una plataforma digital diseñada para mejorar la organización y comunicación entre estudiantes y profesores al

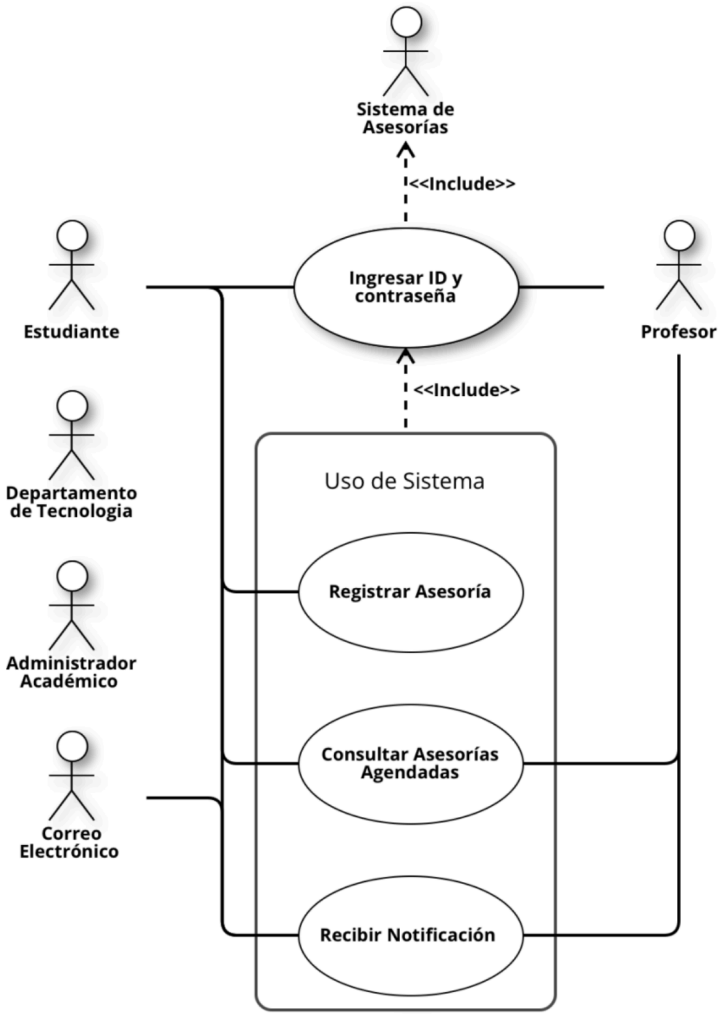
momento de solicitar y gestionar asesorías académicas. Este sistema permite registrar, consultar y administrar sesiones de manera eficiente, ofreciendo autenticación de usuarios, disponibilidad de horarios en tiempo real y confirmaciones formales de cada asesoría.

Con este proyecto se busca resolver problemas comunes derivados de la comunicación informal por correo electrónico, aumentando la eficiencia académica y reduciendo errores en la coordinación de asesorías.

Casos de Uso UML

A continuación se muestran 3 diferentes casos de uso con sus respectivos diagramas, donde los 3 escenarios muestran una alternativa distinta con la misma finalidad de abordar el problema principal.

1. **Caso 1: Sistema de Registro de Asesorías Universitarias**
    a. Diagrama de casos de uso



    b. Descripción

Este caso de uso permite a los estudiantes y profesores de una universidad registrar, consultar y gestionar asesorías académicas mediante una plataforma digital. El sistema

asegura la autenticación de los usuarios, la disponibilidad de horarios y el registro de información relevante sobre las sesiones realizadas.

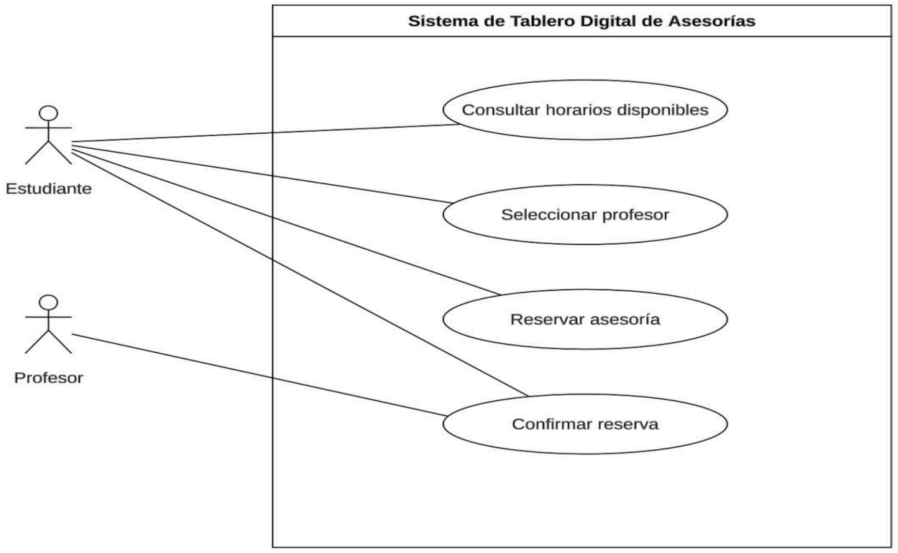
c. Flujo principal

- El usuario (estudiante o profesor) accede al sistema de asesorías universitarias.
- El sistema solicita autenticación mediante usuario (ID) y contraseña institucional.
- El usuario ingresa su ID y su contraseña.
- El sistema valida las credenciales y muestra el menú principal: 1. Registrar Asesoría,

  2. Consultar asesorías agendadas.
- El estudiante selecciona la opción “Registrar asesoría”.
- El sistema muestra los profesores disponibles por materia o área académica.
- El estudiante selecciona un profesor y consulta los horarios disponibles.
- El sistema muestra los horarios y permite seleccionar uno.
- El estudiante selecciona la fecha y hora de la asesoría.
- Se establece el lugar de la asesoría
- Si el profesor tiene oficina, su oficina es el lugar para la asesoría
- Si el profesor no tiene oficina, se agenda un salón.
- El sistema tiene un registro de los salones disponibles
- El sistema ofrece un salón automáticamente.
- El sistema solicita confirmar la solicitud.
- El estudiante confirma el registro.
- El sistema confirma que la información fue registrada correctamente.
- El sistema genera un número de registro y envía la notificación al profesor y al estudiante.
- El sistema actualiza la base de datos con la asesoría programada.
- La asesoría se agrega como un nuevo evento al calendario del correo electrónico y envía un recordatorio 1 día antes de la asesoría.
- El usuario (estudiante o profesor) puede ingresar al sistema para consultar las asesorías agendadas y/o cancelar asesorías.

2. Caso 2: Tablero Digital de Asesorías

a. Diagrama



b. Descripción

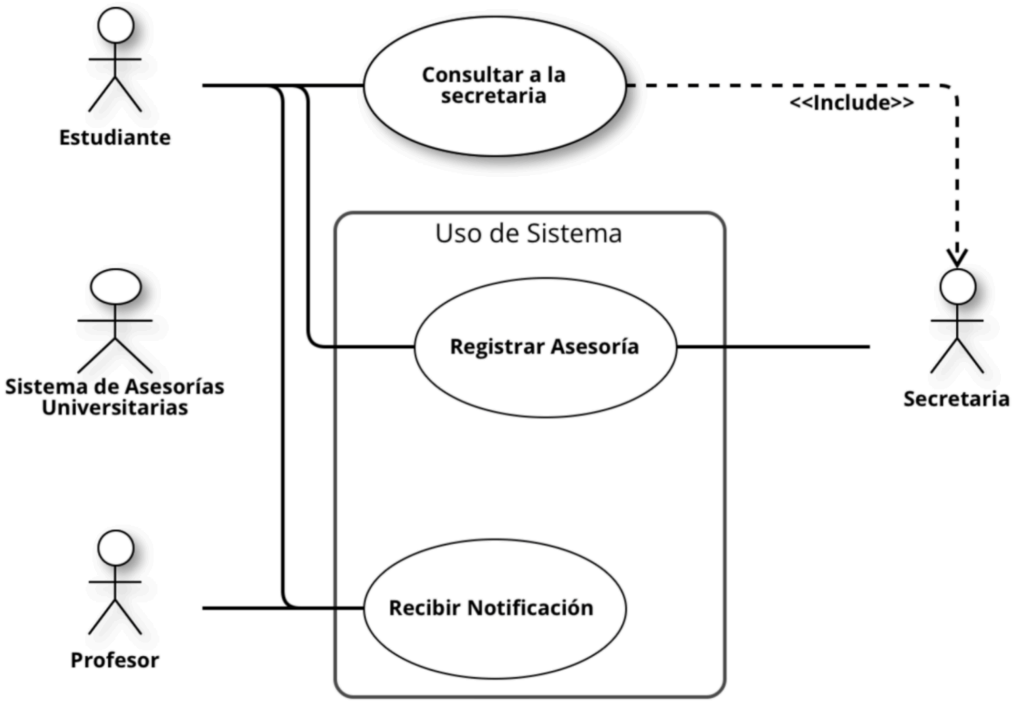
Este caso de uso describe un sistema digital ubicado fuera de las oficinas de los profesores, donde los estudiantes pueden consultar los horarios disponibles de asesorías y reservar un espacio. El tablero digital muestra la disponibilidad actualizada y permite que los estudiantes reserven sin necesidad de ingresar a un sistema en línea.

c. Flujo Principal

1. El estudiante se acerca al tablero digital fuera de las oficinas.
2. El sistema muestra la lista de profesores y sus horarios disponibles.
3. El estudiante selecciona un profesor y un horario disponible.
4. El sistema solicita confirmar la reserva.
5. El estudiante confirma la reserva.
6. El sistema actualiza la disponibilidad y muestra la confirmación.
7. El profesor y el estudiante recibe una notificación automática sobre la nueva asesoría agendada.

3. Caso 3: Secretaria de Asesorías

a. Diagrama



b. Descripción

Este caso de uso plantea que una secretaria, ubicada fuera de las oficinas de los profesores, será la encargada de registrar las asesorías solicitadas por los estudiantes. La secretaria consulta la disponibilidad del profesor, agenda la asesoría y registra los datos en el sistema institucional.

c. Flujo Principal

1. El estudiante se acerca a la secretaría para solicitar una asesoría.
2. La secretaria ingresa al sistema de asesorías institucionales.
3. La secretaria consulta los horarios disponibles del profesor.
4. El estudiante selecciona la fecha y hora deseada.

5. La secretaria registra la asesoría en el sistema.
6. El sistema genera la confirmación y envía la notificación al estudiante y al profesor.

Diagramas de Actividades

a. Escenario generado con Chat GPT

Al iniciar mi jornada como profesor universitario, lo primero que hago es entrar al Sistema de Registro de Asesorías. Abro el portal institucional, escribo mi ID y mi contraseña, y accede. A veces, por escribir rápido, cometo un error; cuando eso ocurre, el sistema me muestra un mensaje indicando que las credenciales son incorrectas y debo intentarlo nuevamente. Una vez que ingresó correctamente, aparece el menú principal con dos opciones: **Registrar Asesoría** y **Consultar Asesorías Agendadas**. Normalmente, comienzo revisando las asesorías que ya están programadas conmigo. Desde esa sección puedo ver quién me buscará, en qué fecha, a qué hora será la asesoría y en qué lugar se llevará a cabo.

Mientras reviso mi agenda, sé que los estudiantes también pueden programar asesorías conmigo. Ellos ingresan al sistema, seleccionan “Registrar Asesoría”, eligen la materia y luego pueden ver qué profesores están disponibles. Si ese día no hubiera profesores para la materia que necesitan, el sistema les muestra que no hay disponibilidad y les permite buscar otra opción. Cuando un estudiante me elige, el sistema le muestra mis horarios disponibles. Puede ocurrir que el estudiante seleccione un horario y, justo en ese momento, otro alumno lo haya tomado; de ser así, el sistema avisa que el espacio ya está ocupado y le pide que seleccione otro. Una vez que el horario queda asignado, el sistema determina el lugar: si tengo oficina registrada, se usa esa como sede; si no, el sistema busca un salón disponible y, en caso de no encontrarlo libre, ofrece otro. Cuando el estudiante confirma la solicitud, el sistema genera un número de registro, nos envía una notificación a ambos, guarda la asesoría en la base de datos y agrega el evento automáticamente a nuestro calendario, con un recordatorio un día antes.

Cuando llega el día de la asesoría, revisó nuevamente en el sistema la información registrada: quién es el estudiante, qué tema desea trabajar y el lugar asignado. Después de atender la sesión, ingreso al sistema para registrar mis observaciones sobre el desempeño del estudiante, acuerdos o tareas que deberá realizar. Si por alguna razón olvido hacerlo, con el paso de las horas el sistema me envía un recordatorio para no dejar la sesión sin comentarios registrados.

Si en algún momento necesito cancelar una asesoría —por ejemplo, por una reunión inesperada o una situación personal— puedo ingresar al sistema, ver mis asesorías programadas, seleccionar la que necesito cancelar y confirmar la acción. De inmediato, el sistema actualiza el estado de la asesoría, notifica al estudiante y a mí, y elimina el evento del calendario electrónico. Lo mismo ocurre cuando es el estudiante quien cancela: ambos recibimos la notificación y el horario queda disponible nuevamente.

No todo siempre funciona a la perfección. Hay días en los que el sistema presenta fallas: un estudiante me dice que intentó agendar una asesoría, pero no aparece registrada, o yo reviso la plataforma y no encuentro una sesión que supuestamente fue confirmada. Cuando eso sucede, lo primero que hago es verificar si el evento se añadió a mi calendario institucional, refrescar el sistema y revisar si aparece. Si el problema persiste, solicito al estudiante una captura de pantalla del error para tener evidencia y reporto el incidente al Departamento de Tecnología. También espero que el sistema explique claramente qué ocurrió, no duplique la asesoría y, sobre todo, permita liberar el horario si quedó bloqueado

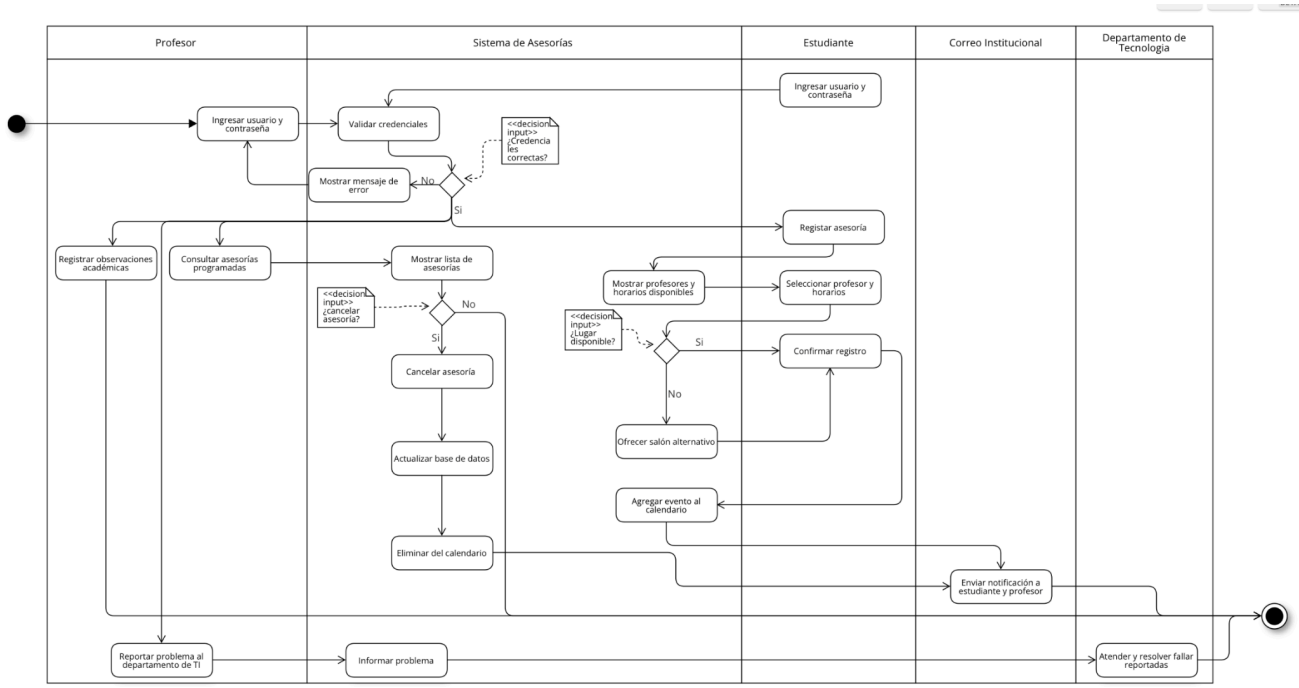
por el error. Mientras el problema se resuelve, suelo comunicarme directamente con el estudiante para confirmar la asesoría por correo o incluso reprogramar manualmente si es necesario.

Al finalizar mi día, entré una última vez al sistema para revisar mis asesorías futuras, asegurarse de haber añadido comentarios y cerrar sesión.

b. Análisis del escenario

- **Actores involucrados**:
- **Usuario principal:** Profesor.
- **Otros actores:** Estudiante (agenda asesorías), Sistema de Asesorías (gestiona información), Correo electrónico institucional (envía notificaciones), Departamento de Tecnología (atiende fallas).
- **Inicio de sesión y validación:**
  El profesor accede al sistema ingresando usuario y contraseña. El sistema valida credenciales; si son incorrectas, muestra error y solicita reintento.
- **Consulta y gestión de asesorías agendadas:**
  El profesor consulta asesorías programadas (fecha, hora, lugar, estudiante). Puede cancelar una asesoría, generando notificación automática a ambas partes, actualización en la base de datos y eliminación en calendario.
- **Proceso de registro de asesoría por el estudiante:**
  El estudiante elige “Registrar Asesoría”, selecciona profesor y horario disponible. El sistema asigna lugar (oficina o salón). Requiere confirmación y, una vez realizada, genera número de registro, notifica a ambos y agrega el evento al calendario.
- **Decisiones y bifurcaciones del sistema:**
- Credenciales incorrectas → mensaje de error.
- ¿Cancelar asesoría? → Eliminar del calendario
- Lugar ocupado → ofrecer alternativa.
- **Registro posterior a la asesoría:**
  Tras la sesión, el profesor debe registrar observaciones académicas.
- **Manejo de errores y excepciones del sistema:**
  Si la asesoría no aparece registrada o no se generan notificaciones, el profesor verifica el calendario y reporta al Departamento de Tecnología. Se espera que el sistema informe el problema, evite duplicados y libera horarios bloqueados.

c. Diagrama de actividades completo



d. Reflexión breve sobre el uso de IA para el diseño del flujo

La experiencia de utilizar Chat GPT para la generación de escenarios del caso de uso “Registro de Asesorías Universitarias” fue significativa, ya que permitió obtener descripciones detalladas sobre el comportamiento del sistema desde la perspectiva de los usuarios. La herramienta facilitó la redacción de situaciones realistas y coherentes, lo que ayudó a identificar con mayor precisión las acciones, decisiones y posibles excepciones que intervienen en el proceso.

Gracias a los escenarios generados, fue posible construir una base sólida para el análisis funcional y el diseño del diagrama de actividades, el cual se elaboró posteriormente de forma independiente. El uso de inteligencia artificial contribuyó a optimizar el tiempo destinado a la redacción, mejorar la comprensión del flujo de trabajo y fortalecer la capacidad de análisis.

En conjunto, la experiencia demostró que Chat GPT puede ser un recurso eficaz para apoyar el aprendizaje y la documentación de sistemas académicos complejos.

Modelo de Clases (POO)

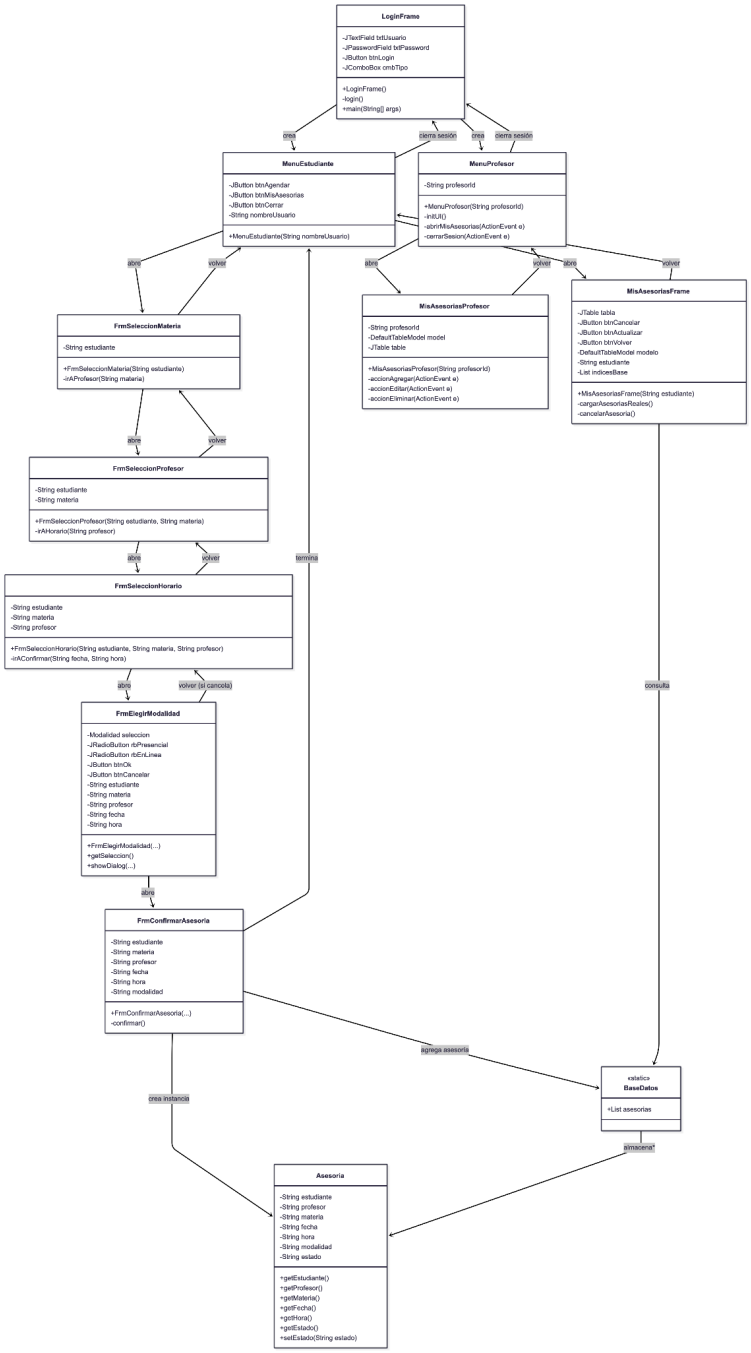
a. Tabla Organizada

Clase	Atributos	Métodos
LoginFrame	- JTextField txtUsuario - JPasswordField txtPassword - JButton btnLogin - JComboBox<String> cmbTipo	- LoginFrame() (constructor) - login() - main()
MenuEstudiante	- JButton btnAgendar - JButton btnMisAsesorias	- MenuEstudiante(String) - main()

	- JButton btnCerrar - String nombreUsuario	
FrmSeleccionMateria	- String estudiante	-FrmSeleccionMateria(String) - irAProfesor(String) - main()
FrmSeleccionProfesor	- String estudiante - String materia	-FrmSeleccionProfesor(String, String) - irAHorario(String) - main()
FrmSeleccionHorario	- String estudiante - String materia - String profesor	-FrmSeleccionHorario(String, String, String) - irAConfirmar(String, String) - main()
FrmElegirModalidad	- Modalidad seleccion -JRadioButton rbPresencial -JRadioButton rbEnLinea -JButton btnOk -JButton btnCancelar -String estudiante -String materia -String profesor -String fecha -String hora	-FrmElegirModalidad(Window parent) -FrmElegirModalidad(Window parent, String estudiante, String materia, String profesor, String fecha, String hora) -initComponents() -getSeleccion() -showDialog(Window parent) -showDialog(Window parent, String estudiante, -String materia, String profesor, String fecha, String hora)
FrmConfirmarAsesoria	- String estudiante - String materia - String profesor - String fecha - String hora - String modalidad	- FrmConfirmarAsesoria(...) - confirmar() - main()
MisAsesoriasFrame	- JTable tabla - JButton btnCancelar - JButton btnActualizar - JButton btnVolver - DefaultTableModel modelo - String estudiante - List<Integer> indicesBase	-MisAsesoriasFrame(String) - cargarAsesoriasReales() - cancelarAsesoria() - main()
MenuProfesor	- String profesorId	- MenuProfesor(String) - initUI() -abrirMisAsesorias(ActionEv

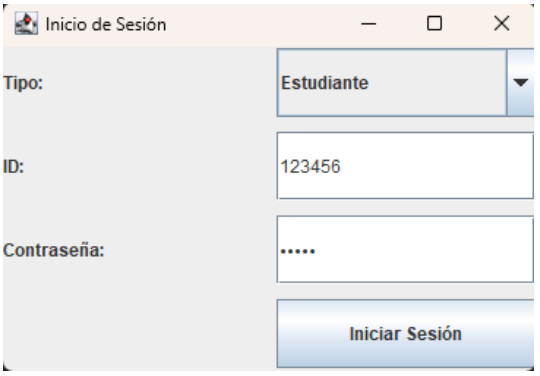
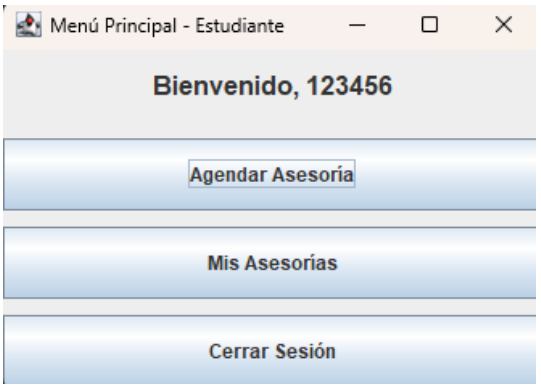
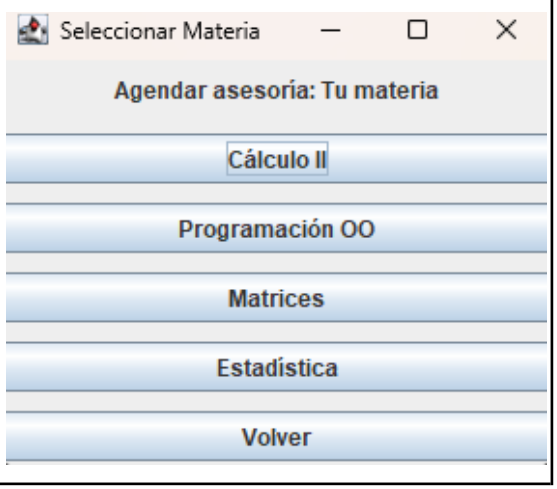
		ent) - cerrarSesion(ActionEvent)
MisAsesoriasProfesor	- String profesorId - DefaultTableModel model - JTable table	-MisAsesoriasProfesor(String) -accionAgregar(ActionEvent) - accionEditar(ActionEvent) -accionEliminar(ActionEvent)

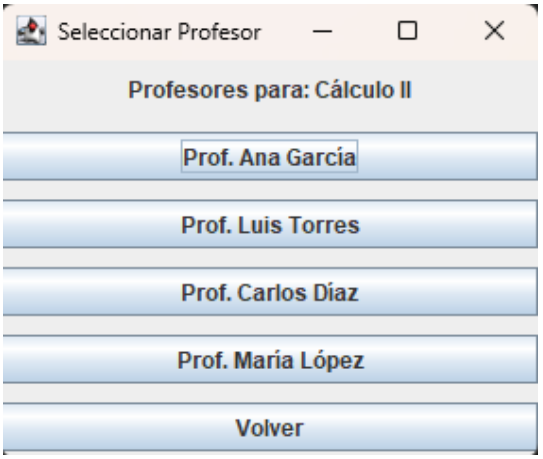
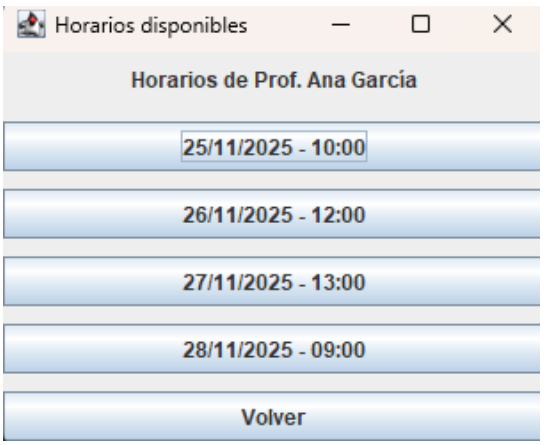
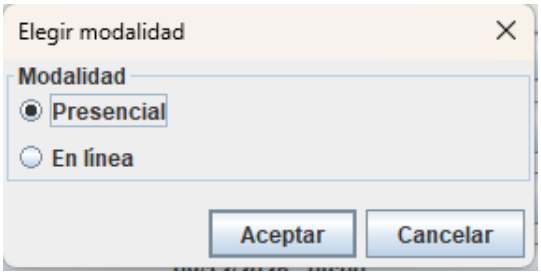
b. Diagrama de clases estructurado

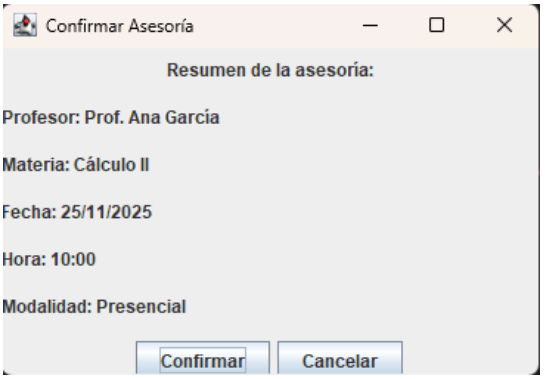
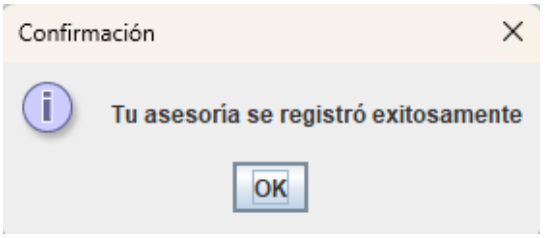
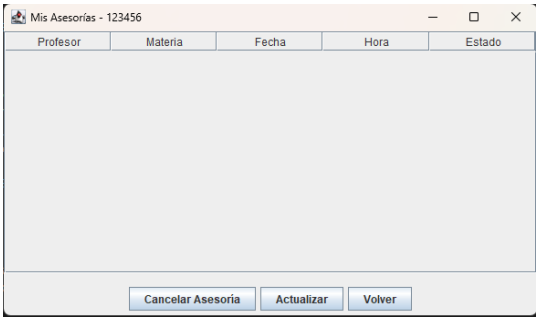
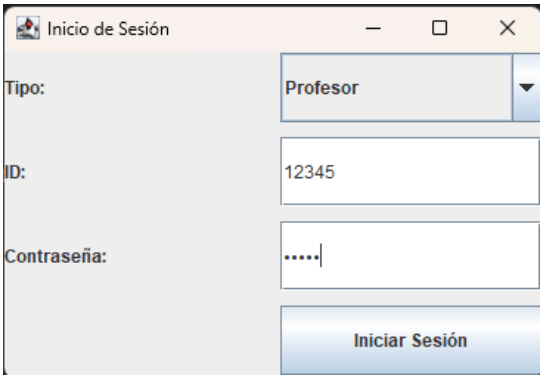


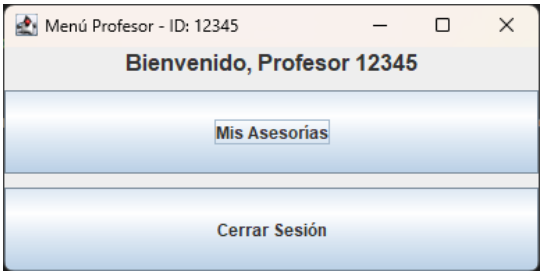
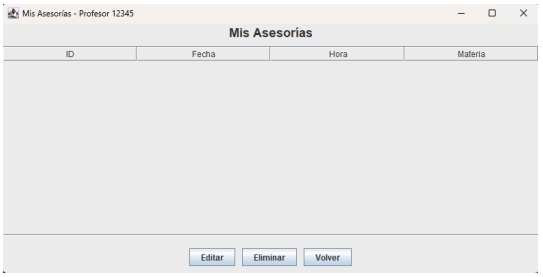
Interfaces Gráficas (Java Swing)

a. Capturas de pantalla de cada interfaz y Descripción funcional de cada pantalla

Pantalla	Descripción Funcional	Captura
Inicio Sesión Estudiante	Esta pantalla permite que el estudiante acceda al sistema ingresando sus credenciales. **Funciones principales:** ● Campo para ingresar ID del estudiante. ● Campo para ingresar contraseña. ● Botón *Iniciar sesión* que valida los datos en la base de datos SQLite. ● Mensajes de error en caso de credenciales incorrectas.	
Menú principal Estudiante	Es la interfaz principal disponible tras iniciar sesión. **Funciones principales:** ● Botón para *Solicitar asesoría*. ● Botón *Mis asesorías* para consultar asesorías registradas.Botón *Cerrar sesión*.	
Seleccionar materia	Aquí se muestra al estudiante una lista de materias disponibles. **Funciones principales:** ● Lista visual con todas las materias que tienen asesorías disponibles.	

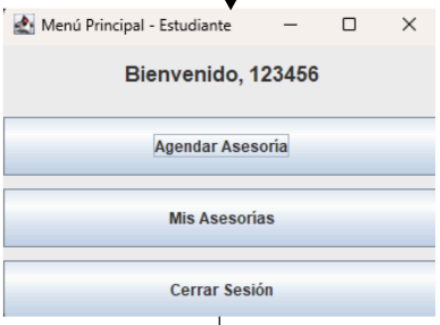
Seleccionar Profesor	Después de seleccionar la materia, esta pantalla muestra a los profesores disponibles. **Funciones principales:** ● Lista con profesores que imparten la materia seleccionada. ● Botón *Volver*.	
Horarios Disponibles	Permite elegir el horario en el que el profesor tiene disponibilidad. **Funciones principales:** ● Despliegue de horarios válidos y vigentes. ● Selección única de horario. ● Botón *volver* para cambiar de profesor	
Elegir Modalidad	Muestra una ventana para elegir la modalidad de la asesoría antes del registro final. **Funciones principales:** ● Permitir elegir entre modalidad **Presencial** o **En línea**. ● Botón **Aceptar** para confirmar la modalidad. ● Botón **Cancelar** para abortar la operación. ● Si se proporciona información completa, abre la ventana de confirmación final.	

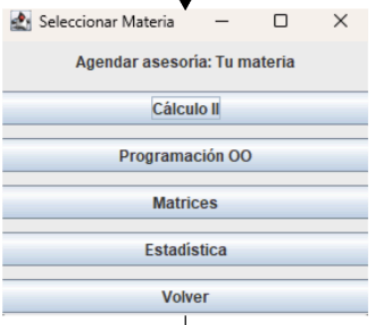
Confirmar Asesoría	Muestra un resumen previo al registro final. **Funciones principales:** ● Visualizar la información completa: ○ Materia ○ Profesor ○ Fecha ○ Hora ○ Modalidad Botón *Confirmar asesoría*. Botón *Cancelar*.	
Confirmación	Pantalla que indica que la asesoría se registró correctamente. **Funciones principales:** ● Mensaje de éxito: “Tu asesoría se registró exitosamente”. ● Botón *Ok al menú*.	
Mis Asesorías	El estudiante puede consultar sus asesorías ya registradas. **Funciones principales:** ● Tabla que muestra: ○ Materia ○ Profesor ○ Fecha ○ Hora ○ Estado (vigente, cancelada, concluida) ● Botón para cancelar asesoría. ● Botón *Volver al menú*.	
Inicio Sesión - Profesor	Pantalla para que el docente acceda al sistema. **Funciones principales:** ● Campo para ID de profesor. ● Campo de contraseña. ● Botón *Iniciar sesión*. ● Mensajes de error si las credenciales no coinciden con SQLite.	

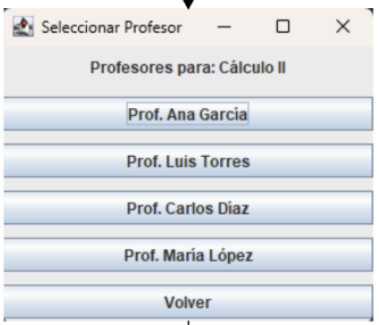
Menú Profesor	Interfaz que muestra las funciones disponibles para el profesor. **Funciones principales:** ● Botón *Mis asesorías*. ● Botón *Cerrar sesión*.	
Mis Asesorías Profesor	Muestra las asesorías asignadas al profesor. **Funciones principales:** ● Tabla con: ○ id ○ Fecha ○ Hora ○ Materia ● Botón para cancelar asesoría. ● Botón *Regresar al menú*.	

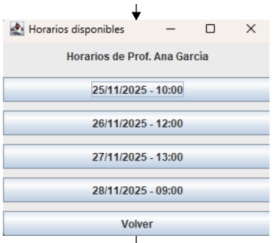
b. Diagrama de navegación entre interfaces


Inicio de Sesión
Tipo:
Estudiante
ID:
123456
Contraseña:
Iniciar Sesión

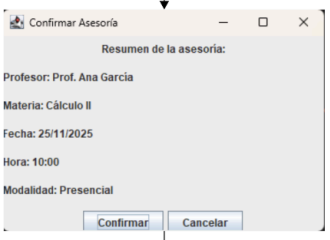

Menú Principal - Estudiante
Bienvenido, 123456
Agendar Asesoria
Mis Asesorías
Cerrar Sesión

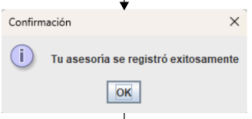

Seleccionar Materia
Agendar asesoría: Tu materia
Cálculo II
Programación OO
Matrices
Estadística
Volver

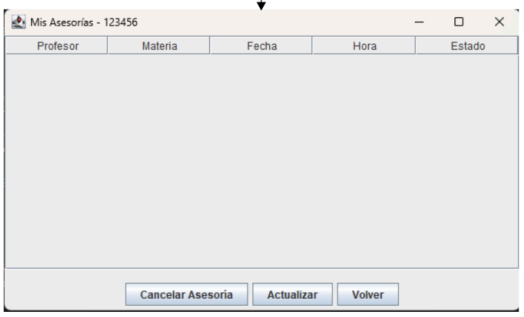

Seleccionar Profesor
Profesores para: Cálculo II
Prof. Ana Garcia
Prof. Luis Torres
Prof. Carlos Díaz
Prof. María López
Volver

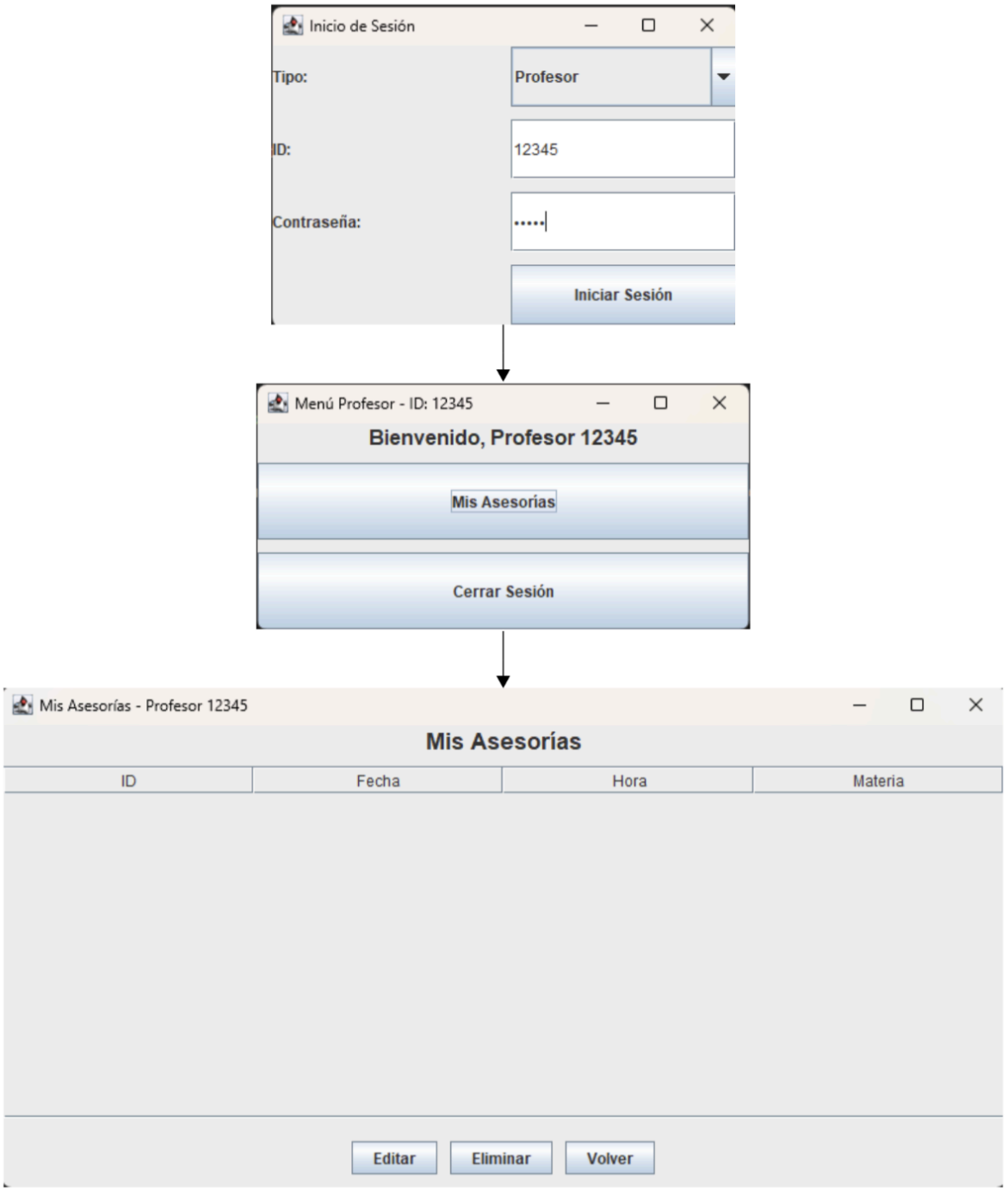

Horarios disponibles
Horarios de Prof. Ana Garcia
25/11/2025 - 10:00
26/11/2025 - 12:00
27/11/2025 - 13:00
28/11/2025 - 09:00
Volver


Elegir modalidad
Modalidad
Presencial
En línea
Aceptar
Cancelar


Confirmar Asesoría
Resumen de la asesoría:
Profesor: Prof. Ana Garcia
Materia: Cálculo II
Fecha: 25/11/2025
Hora: 10:00
Modalidad: Presencial
Confirmar
Cancelar

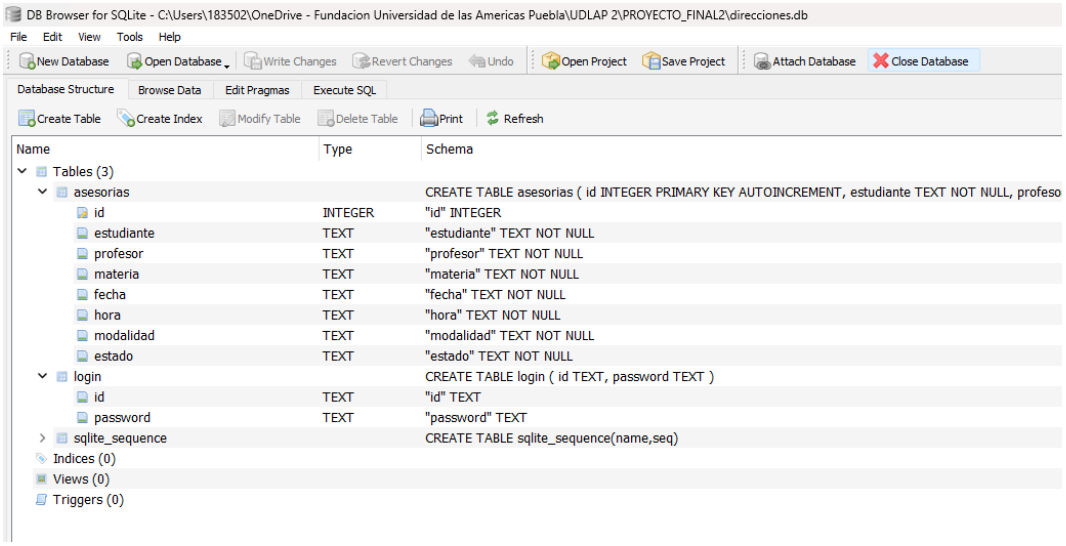

Confirmación
Tu asesoría se registró exitosamente
OK


Mis Asesorías - 123456
Profesor
Materia
Fecha
Hora
Estado
Cancelar Asesoría
Actualizar
Volver

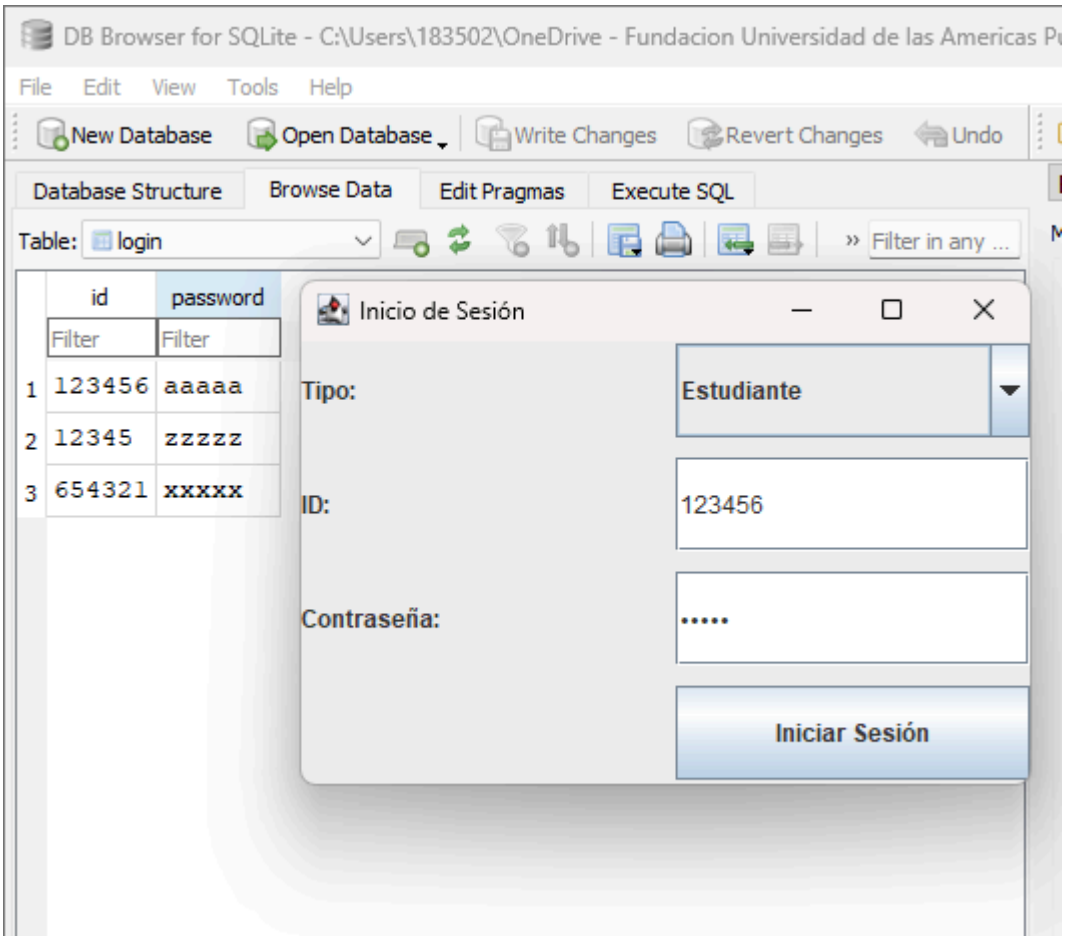


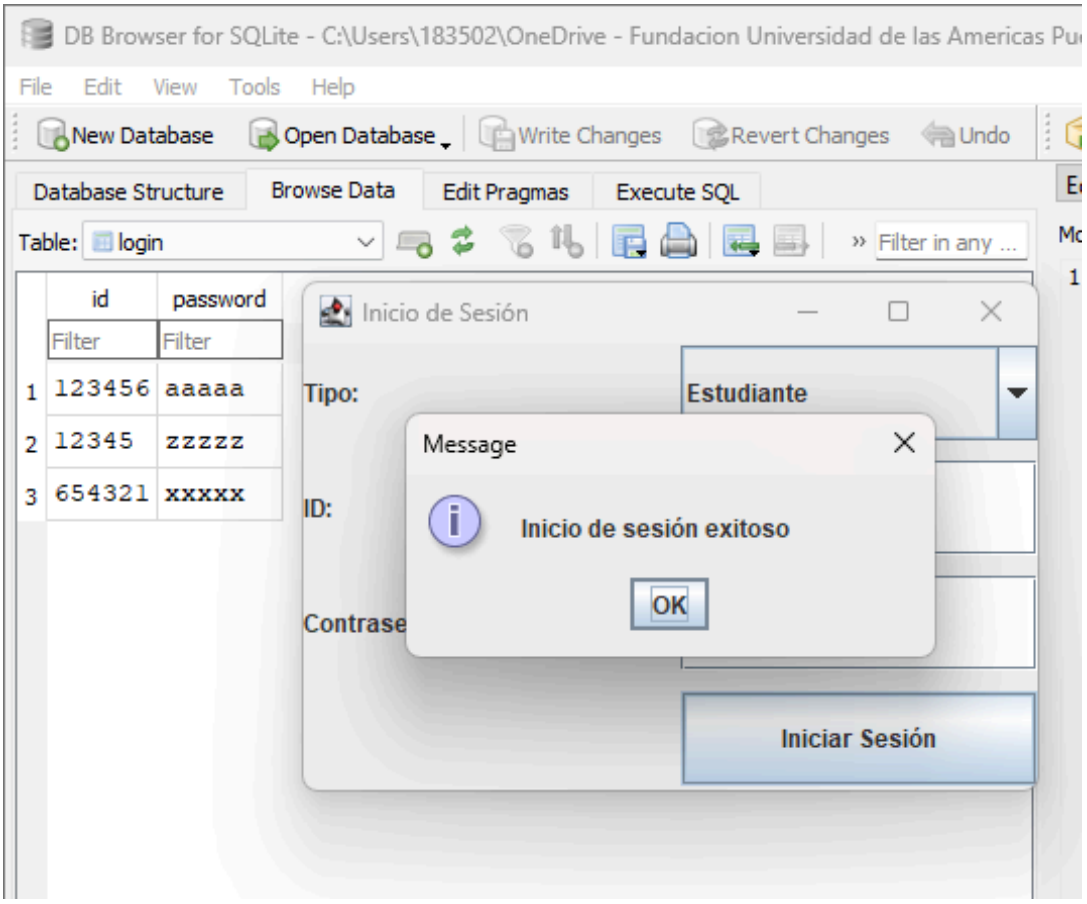
Integración con Base de Datos SQLite

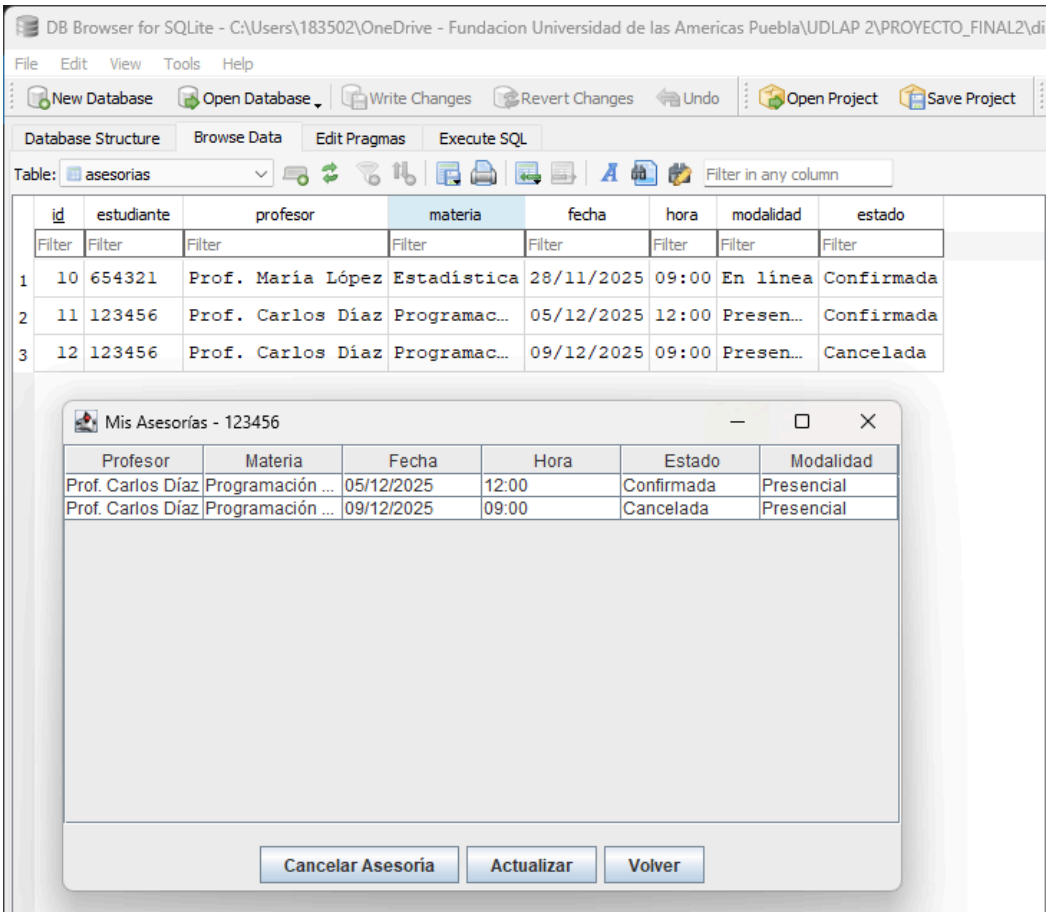
a. Esquema de la base de datos (tablas y relaciones)



b. Capturas de formularios interactuando con la base




DB Browser for SQLite - C:\Users\183502\OneDrive - Fundacion Universidad de las Americas Pu
File Edit View Tools Help
New Database
Open Database
Write Changes
Revert Changes
Undo
Database Structure
Browse Data
Edit Pragmas
Execute SQL
Table: login
Filter in any ...
id
password
Filter
Filter
1 123456 aaaaa
2 12345 zzzzz
3 654321 xxxxx
Inicio de Sesión
Tipo:
Estudiante
ID:
Contrase
Message
Inicio de sesión exitoso
OK
Iniciar Sesión


DB Browser for SQLite - C:\Users\183502\OneDrive - Fundacion Universidad de las Americas Puebla\UDLAP 2\PROYECTO_FINAL2\di
File Edit View Tools Help
New Database
Open Database
Write Changes
Revert Changes
Undo
Open Project
Save Project
Database Structure
Browse Data
Edit Pragmas
Execute SQL
Table: asesorias
Filter in any column
id estudiante profesor materia fecha hora modalidad estado
1 10 654321 Prof. María López Estadística 28/11/2025 09:00 En línea Confirmada
2 11 123456 Prof. Carlos Díaz Programac... 05/12/2025 12:00 Presen... Confirmada
3 12 123456 Prof. Carlos Díaz Programac... 09/12/2025 09:00 Presen... Cancelada
Mis Asesorías - 123456
Profesor Materia Fecha Hora Estado Modalidad
Prof. Carlos Díaz Programación ... 05/12/2025 12:00 Confirmada Presencial
Prof. Carlos Díaz Programación ... 09/12/2025 09:00 Cancelada Presencial
Cancelar Asesoría
Actualizar
Volver

c. Código de conexión

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class ConexionSQLite {
    private static final String DB_URL =
"jdbc:sqlite:direcciones.db";



    @SuppressWarnings("CallToPrintStackTrace")
    public static Connection conectar() {
        Connection conexion = null;
        try {
            // Cargar el driver JDBC para SQLite
            Class.forName("org.sqlite.JDBC");

            // Obtener la conexión
            conexion = DriverManager.getConnection(DB_URL);
            System.out.println("Conexión exitosa a SQLite");
        } catch (ClassNotFoundException e) {
            System.err.println("Error: No se encontró el driver
SQLite");
            e.printStackTrace();
        } catch (SQLException e) {
            System.err.println("Error de conexión a la base de
datos");
            e.printStackTrace();
        }
        return conexion;
    }

    public static void main(String[] args) {
        conectar();
    }
}
```

Pruebas y Validación

a. Evidencia de pruebas funcionales y de validación

Para garantizar el correcto funcionamiento del sistema de gestión de asesorías, se realizaron pruebas funcionales enfocadas en validar los módulos principales: agendar asesorías, cancelar asesorías, listar asesorías por estudiante y validar el ID del usuario. Estas pruebas se implementaron utilizando **JUnit**, lo que permitió verificar el comportamiento esperado de cada método y asegurar la integridad de los datos dentro de la aplicación.

b. Resultados de pruebas JUnit

Prueba 1: Agendar Asesoría

En esta prueba se evaluó el funcionamiento del proceso de registro de una asesoría nueva dentro del sistema. Para ello, se creó un objeto de tipo *Asesoría* con datos válidos y se añadió a la lista de asesorías almacenada en la clase *BaseDatos*. El resultado fue satisfactorio: la lista incrementó su tamaño a uno, confirmando que la asesoría se agregó correctamente. Además, se verificó que el ID del estudiante almacenado coincidiera con el valor ingresado inicialmente, lo que garantiza que la información no sufre alteraciones durante el proceso. Por lo tanto, esta prueba demuestra que la funcionalidad de agendar asesorías opera de manera adecuada.

Prueba 2: Cancelar Asesoría

En esta prueba se buscó validar que el sistema permitiera modificar correctamente el estado de una asesoría existente. Se generó una asesoría en estado “Activa”, se añadió a la base de datos y posteriormente se cambió su estado a “Cancelada”. La verificación mediante JUnit confirmó que el nuevo estado se actualizó correctamente en el registro correspondiente. Esto comprueba que el proceso de cancelación funciona como se espera, permitiendo al usuario gestionar el estado de sus asesorías sin inconsistencias.

Prueba 3: Listar Asesorías por Estudiante

El propósito de esta prueba fue asegurar que el sistema pudiera identificar y filtrar correctamente las asesorías pertenecientes a un estudiante específico. Para ello, se agregaron tres asesorías a la base de datos, dos de ellas asociadas al estudiante con ID “123456” y una perteneciente a otro estudiante. Después, mediante una operación de filtrado, se contó cuántas asesorías correspondían a ese ID. Los resultados mostraron que el sistema detectó exactamente dos asesorías, confirmando que no se mezclan registros entre estudiantes distintos y que el filtrado se realiza de manera precisa. Esta prueba valida la correcta implementación del listado personalizado de asesorías.

Prueba 4: Validación de ID de Estudiante

La última prueba evaluó la función encargada de comprobar que el ID de un estudiante cumpliera con el formato requerido: exactamente seis dígitos numéricos. Para ello, se utilizó un ID de ejemplo (“123456”) y se comprobó mediante una expresión regular si cumplía con las reglas establecidas. El resultado fue positivo, lo que demuestra que el sistema es capaz de validar correctamente los IDs ingresados por los usuarios, reduciendo la posibilidad de errores por datos inválidos.

Prueba 5: Validación de ID de Profesor

Esta prueba tuvo como objetivo confirmar que los IDs asignados a los profesores cumplieran con el formato establecido dentro del sistema: cinco dígitos numéricos. Para ello, se evaluó un ID de ejemplo y se aplicó una expresión regular que valida su estructura. La prueba concluyó de manera exitosa, demostrando que el sistema distingue correctamente el formato de los identificadores de los profesores y permite detectar entradas inválidas antes de que causen errores en los procesos asociados.

Prueba 6: Verificación de Existencia de Estudiantes en la Base de Datos

Con esta prueba se buscó garantizar que el sistema sea capaz de validar la existencia de un estudiante registrado en la base de datos antes de realizar operaciones relacionadas. Se seleccionó un ID previamente almacenado en la tabla `login` y se ejecutó una consulta directa a la base. El resultado mostró que el registro efectivamente existe, lo que confirma que el sistema puede identificar correctamente a los usuarios válidos y prevenir operaciones sobre datos inexistentes.

Prueba 7: Verificación de Existencia de Profesores en la Base de Datos

De forma similar a la prueba anterior, en esta validación se comprobó que un profesor registrado pudiera ser localizado correctamente en la base de datos. Se utilizó un ID real perteneciente a un profesor y se realizó una consulta SQL mediante JUnit. El sistema devolvió un resultado positivo, indicando que el registro existe en la tabla y que el manejo de credenciales es consistente y confiable.

Prueba 8: Disponibilidad de Horarios para Asesorías

Esta prueba evaluó la capacidad del sistema para verificar si un horario determinado se encuentra disponible antes de agendar una asesoría. Se proporcionaron datos específicos del profesor, fecha y hora, y se comprobó si existía o no un registro previo en la base de datos que coincidiera con esos criterios. Los resultados indicaron que el horario estaba libre, lo que demuestra que el sistema es capaz de prevenir conflictos entre asesorías y asegurar una correcta gestión de disponibilidad.

Prueba 9: Prevención de Doble Agendado

El objetivo de esta prueba fue validar que el sistema impida agendar más de una asesoría en el mismo horario con el mismo profesor. Para lograrlo, primero se limpió cualquier registro previo con esos datos y posteriormente se insertó una asesoría manualmente. Luego, se realizó una verificación para asegurarse de que el sistema detectara exactamente un registro en esa combinación de fecha, hora y profesor. La prueba fue satisfactoria y confirma que el sistema evita la duplicación de horarios, garantizando la integridad de la programación de asesorías.

Prueba 10: Verificación de Login Válido

En esta prueba se comprobó que el sistema reconozca correctamente las credenciales válidas de un usuario. Se proporcionó un ID real junto con su contraseña correspondiente y

se ejecutó una consulta en la tabla `login`. El resultado confirmó que el par ID-contraseña coincide con un registro válido, lo que garantiza que el mecanismo de autenticación funciona adecuadamente.

Prueba 11: Login con Contraseña Incorrecta

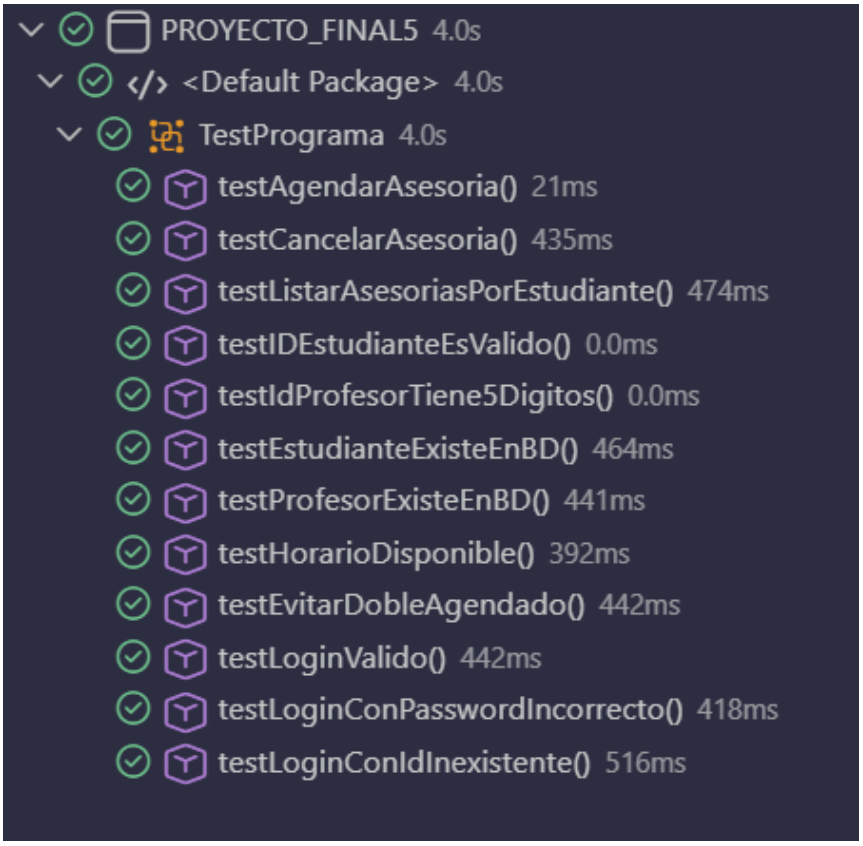
Esta prueba verificó que el sistema sea capaz de detectar contraseñas incorrectas. Se utilizó un ID válido acompañado de una contraseña errónea y se verificó que el sistema no encontrara coincidencias en la base de datos. El resultado fue el esperado, demostrando que el sistema rechaza intentos de acceso con credenciales inválidas y refuerza la seguridad del proceso de autenticación.

Prueba 12: Login con ID Inexistente

Finalmente, se evaluó la respuesta del sistema ante un intento de inicio de sesión con un ID que no se encuentra registrado. Se proporcionó un identificador inexistente y una contraseña válida. Tal como se esperaba, la consulta no arrojó resultados, lo que indica que el sistema distingue correctamente entre usuarios reales y no registrados, manteniendo la integridad del control de acceso

c. Checklists, reportes o capturas de salida

Se realizó la ejecución de pruebas JUnit implementada para el proyecto. En la siguiente evidencia se muestran los resultados obtenidos durante la ejecución de cada caso de prueba, donde se puede observar que todas las pruebas fueron exitosas. Esto confirma que las funcionalidades principales del sistema operan correctamente y que no se encontraron errores en los módulos evaluados.



Conclusiones del equipo

a. Retos enfrentados

El desarrollo del Sistema de Registro de Asesorías presentó varios desafíos importantes. Uno de ellos fue diseñar un flujo que contempla todas las variantes reales del proceso, como credenciales incorrectas, cancelaciones y manejo de excepciones cuando el sistema no lograba registrar una asesoría correctamente. También representó un reto integrar SQLite como base de datos, garantizando consultas eficientes, manejo correcto de conexiones, validación de datos y registro seguro de cada asesoría.

b. Aprendizajes obtenidos

El desarrollo del sistema permitió comprender la importancia de modelar correctamente los procesos académicos y traducirlos a casos de uso claros y funcionales. Se aprendió a identificar puntos críticos como la validación de credenciales y gestión de horarios. También se reforzó el conocimiento sobre la creación y diseño de bases de datos SQLite, especialmente en lo relacionado con la integridad de la información, relaciones entre entidades y manejo de errores. Además, se entendió el valor de documentar escenarios completos del usuario, lo que facilitó prever situaciones reales dentro del sistema y mejorar la precisión del diseño.

c. Posibles mejoras o extensiones futuras del sistema

Entre las mejoras futuras se podría añadir un historial completo de asesorías por estudiante y por profesor, con métricas que permitan evaluar la demanda académica y el rendimiento.

También podría integrarse una aplicación móvil complementaria para facilitar la reserva y consulta desde cualquier dispositivo. Finalmente, una mejora importante sería

optimizar la detección y resolución automática de errores, permitiendo que el sistema libere horarios bloqueados y brinde mensajes más detallados cuando ocurra una falla.

Aprendizajes obtenidos

Como equipo, entendemos mejor cómo un problema cotidiano dentro de la universidad puede convertirse en un proyecto de software completo. Aprendimos a descomponer el proceso de asesorías en actores, casos de uso y flujos específicos, lo que nos ayudó a ordenar la lógica del sistema. Igualmente, fortalecimos nuestras habilidades en manejo de bases de datos con SQLite, desde el diseño de la estructura hasta el manejo de errores al ejecutar consultas.

Otro aprendizaje importante fue la necesidad de pensar siempre en la experiencia del usuario, documentando escenarios de uso y revisando qué tan claro y usable es el sistema para estudiantes, profesores y personal administrativo.