UNIVERSIDAD PRIVADA DEL NORTE

FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS COMPUTACIONALES

"Sistema de gestión de incidencias"
Informe Final

ARENAS NARBASTE, DANIEL ABELARDO (N00171207)

PERÚ

2025

CAPITULO 1: INTRODUCCIÓN

1. Objetivo

Uso eficiente de cada uno de los parámetros indicados (arrays, la recursividad, las funciones, las clases y el manejo de archivos) para desarrollar un sistema de gestión de incidencias utilizando como base el lenguaje Java con Programación Orientada a Objetos (POO) y la librería llamada "Swing" para la interfaz gráfica, de esta manera crear un software de calidad, organizado y fácil de comprender a fin de entender la importancia de uso de los parámetros solicitados.

Uso y presentación del trabajo mediante repositorio de GitHub, comandos git add y git commit, visualizar historial de commits, manejo de branches y merges; crear y cambiar entre ramas (branches) utilizando comandos got Branch y git checkout.

2. Características del trabajo

El presente trabajo consta del desarrollo de un software de gestión de incidencias, el cual permita tener en cuenta los problemas usados al usar el software ERP de una empresa.

3. Importancia

En la mayoría de software sea pequeño, mediana o gran escala, no siempre se implementa el uso de la recursividad o excepciones; siendo las listas y clases las más usadas al momento de programar. La consecuencia de no usar la recursividad o excepciones, aunque no parezca es muy notoria y a la larga, brinda dificultad al programador o equipo de programadores, los cuales podrían ser:

- **Código espagueti**: Sin recursividad, el código puede volverse más extenso y menos mantenible, aumentando el riesgo de errores, así como también el principal problema de no ser de fácil entendimiento.
- **Fallas abruptas**: Sin un manejo adecuado de excepciones, el programa puede fallar inesperadamente, lo que resulta en una mala experiencia para el usuario.
- **Difícil depuración**: Sin excepciones, puede ser complicado identificar y corregir errores, ya que el programa puede no proporcionar información útil sobre lo que salió mal.
- **Desorganización**: Sin clases, el código puede volverse desordenado y difícil de seguir, lo que complica su mantenimiento y evolución.

- **Repetición de código**: La falta de clases puede llevar a la duplicación del código, lo que aumenta el riesgo de errores y hace que el programa sea más difícil de actualizar.

Por ello, es muy recomendable el empleo de arrays, recursividad, funciones y clases al momento de programar, no solo en lenguaje Java, sino en cualquier lenguaje de programación; ya que esto no solo afecta el orden y la calidad del código, sino también la experiencia del usuario final al momento de usarlo, o en todo caso para su fácil entendimiento para la capacidad de respuesta ante errores y cambios futuros.

Recalcando que el uso de clases y funciones en programación es fundamental para estructurar y organizar el código de manera eficiente, modular y reutilizable. Estas herramientas permiten dividir un programa grande en partes más pequeñas y manejables, facilitando su mantenimiento, ampliación y depuración a futuro.

3.1. Convenciones de nombres

Las clases deben nombrarse utilizando el formato PascalCase, donde cada palabra comienza con una letra mayúscula (ejemplo: MiClaseEjemplo). Los métodos y las variables deben utilizar el formato camelCase, donde la primera palabra comienza con minúscula y las siguientes con mayúscula (ejemplo: miMetodoEjemplo, miVariableEjemplo). Los paquetes deben estar en minúsculas y, generalmente, se utilizan nombres de dominio invertidos (ejemplo: com.ejemplo.proyecto). Estas convenciones ayudan a los desarrolladores a entender rápidamente el propósito de cada elemento en el código.

3.2. Sobre los mensajes commit

Los mensajes de commit claros son esenciales para la colaboración efectiva en un proyecto. Proporcionan contexto sobre los cambios realizados, lo que facilita la revisión del historial del proyecto y la identificación de errores. Un buen mensaje de commit debe ser breve pero descriptivo, indicando qué se cambió y por qué.

3.3. Sobre el uso de ramas

El uso de ramas en el desarrollo de software permite a los equipos trabajar en diferentes características o correcciones de errores de manera aislada. Esto minimiza el riesgo de conflictos y permite una integración más fluida de nuevas funcionalidades.

3.4. Convenciones de nombres

- **git stash**: Este comando se utiliza para guardar temporalmente los cambios no confirmados en el área de trabajo. Por ejemplo, si un desarrollador necesita cambiar de rama, pero no está listo para hacer un commit, puede usar `git stash` para guardar su trabajo actual.
- **git revert**: Este comando se utiliza para deshacer cambios en el historial de commits. Por ejemplo, si un commit introdujo un error, se puede revertir con `git revert <commit_id>`, creando un nuevo commit que deshace los cambios del commit especificado.
- **git cherry-pick**: Este comando permite aplicar cambios de un commit específico en otra rama. Por ejemplo, si se desea aplicar un cambio de la rama `feature` a `develop`, se puede usar `git cherry-pick <commit_id>` para llevar ese cambio a la nueva rama.

CAPITULO 2: FUNDAMENTOS DE SOFTWARE

1. Parámetros a usar en el software

1.1. Calidad

- **La sobrecarga de métodos** es una característica de la programación orientada a objetos que permite definir múltiples métodos con el mismo nombre, pero con diferentes parámetros. Esto es útil para realizar la misma operación con diferentes tipos o números de argumentos.

- **El manejo de errores** es fundamental para garantizar que un programa se ejecute de manera robusta. En Java, se utilizan bloques `try-catch` para capturar excepciones y manejar errores de manera controlada.

- **Las colecciones en Java,** como `ArrayList` y `HashMap`, son estructuras de datos que permiten almacenar y manipular grupos de objetos de manera eficiente. Su uso es fundamental para gestionar datos dinámicos y realizar operaciones de búsqueda y manipulación de manera rápida.

1.2. Diagrama de Ishikawa

1.3. Identificación de problemas que afectaron el desarrollo

- El recurso económico limitado con lo cual se cuenta para implementar un software.

- El tiempo necesario para poder desarrollar el software, validando cada una de las partes de la misma llegue a funcionar sin dar algún error.

- Las técnicas de programación orientada a objetos (POO) se tiene que implementar en este proyecto, no solo eso, al ver que es un proyecto a escala, también se puede considerar el Modelo-Vista-Controlador

1.4. Alternativas de solución a estos problemas

- El uso de software libre como Eclipse, el lenguaje de programación Java y sus librerías, SQLite como herramienta para la base de datos.

- Con un poco de esfuerzo y extra de empeño para la programación se logra avanzar gran parte del proyecto.

- Cuestión de revisar los materiales brindados por la universidad, algunas páginas sobre la programación en esa modalidad e implementarla en el desarrollo.

1.5. Objetivos y soluciones

El objetivo principal del desarrollo de este software es crear un medio mas rápido y organizado en donde registrar las incidencias obtenidas con el uso de software ERP que vende la empresa, así como también a la vez ayuda a la empresa a mejorar el software en caso de detectar algunos problemas o errores por mal funcionamiento. Evitar las visitas constantes a las empresas de los clientes, para solución de problemas con el software, solo haciendo las visitas cuando sea necesario.

1.6. Requerimientos funcionales

Código	Detalle	Estado
RF01	Crear un sistema de logeo con autenticación por correo y clave	Realizado
RF02	Validar credenciales desde la base de datos SQLite	Realizado
RF03	Redirigir al usuario al menú correspondiente según su rol	Realizado
RF04	Crear menú principal por rol (Admin, Consultor, Cliente)	Realizado
RF05	Mostrar nombre y rol del usuario en el menú principal	Realizado
RF06	Crear panel para que el cliente registre incidencias	Realizado
RF07	Registrar incidencia con estado inicial "Pendiente"	Realizado
RF08	Crear panel para que el administrador asigne incidencias a consultores	Realizado
RF09	Cambiar estado de la incidencia a "Asignado" al asignarla	Realizado
RF10	Crear panel para que el consultor vea incidencias asignadas	Realizado
RF11	Consultor responde incidencia y cambia estado a "Resuelto"	Realizado
RF12	Cliente puede ver el estado de sus incidencias	Realizado
RF13	Cliente puede ver la respuesta de una incidencia resuelta	Realizado
RF14	Crear panel CRUD de usuarios (Administrador)	Realizado
RF15	Agregar nuevo usuario desde interfaz	Realizado
RF16	Editar usuario desde interfaz	Realizado
RF17	Eliminar usuario lógicamente (activo = 0)	Realizado
RF18	Ver lista de usuarios activos en una tabla	Realizado
RF19	Agregar campo "respuesta" a la tabla incidencia	Realizado

RF20	Mostrar respuesta en área separada para el cliente	Realizado
RF21	Mostrar respuesta en área separada para el consultor	Realizado
RF22	Empaquetar navegación con JFrame principal (`MenuPrincipal`)	Realizado
RF23	Mostrar tabla de incidencias filtradas por usuario logueado	Realizado
RF24	Aplicar eliminación lógica a incidencias	Realizado

Mejoras de consulta y visualización

Código	Detalle	Estado
RF25	Buscar incidencias por título o ID	Realizado
RF26	Filtrar incidencias por estado (Pendiente / Asignado / Resuelto)	Realizado
RF27	Mostrar incidencias ordenadas por fecha de creación	Realizado
RF28	Ver cantidad total de incidencias por estado	Realizado
RF29	Mostrar solo incidencias activas (ocultar eliminadas)	Realizado

Seguridad y control

Código	Detalle	Estado
RF30	Validar correo duplicado al registrar usuario	Realizado
RF31	Encriptar contraseñas al guardar en la base de datos	Realizado
RF32	Validar campos vacíos en formularios con mensajes amigables	Realizado
RF33	Confirmar antes de cerrar sesión con mensaje de advertencia	Realizado
RF34	Validar cambio de estado de incidencia solo por usuarios autorizados	Realizado

Campos adicionales

Código	Detalle	Estado

RF35	Agregar campo de fecha de creación a las incidencias	Realizado
RF36	Agregar campo de fecha de resolución cuando se marca como resuelta	Realizado
RF37	Mostrar fecha de creación y resolución en tablas	Realizado

Exportación y respaldo

Código	Detalle	Estado
RF38	Exportar incidencias a archivo CSV	Realizado
RF39	Exportar usuarios a archivo CSV	Realizado

Roles y configuración

Código	Detalle	Estado
RF40	Cambiar rol de un usuario existente desde PanelUsuario	Realizado

1.7. Historias de usuario

- Como cliente, quiero registrar una incidencia para reportar un problema con el sistema ERP.
- Como cliente, quiero ver el estado de mis incidencias para saber si han sido atendidas.
- Como cliente, quiero leer la respuesta del consultor para entender cómo resolver el problema.
- Como consultor, quiero ver solo las incidencias asignadas a mí para trabajar de forma ordenada.
- Como consultor, quiero responder una incidencia para ayudar al cliente a resolver su problema.
- Como administrador, quiero asignar incidencias a consultores para garantizar su atención.
- Como administrador, quiero registrar nuevos usuarios con diferentes roles para gestionar accesos.
- Como administrador, quiero editar los datos de los usuarios para mantener la información actualizada.
- Como administrador, quiero eliminar usuarios lógicamente para conservar el historial sin perder datos.
- Como administrador, quiero ver una lista de usuarios activos para tener control sobre el sistema.
- Como administrador, quiero tener acceso a todas las incidencias para poder gestionarlas si es necesario.

- Como cualquier usuario, quiero iniciar sesión con mis credenciales para acceder al sistema.
- Como administrador, quiero filtrar incidencias por estado para revisar solo las que requieren atención.
- Como consultor, quiero registrar la fecha de resolución para tener trazabilidad de mi trabajo.
- Como cliente, quiero buscar una incidencia por título para encontrarla rápidamente.
- Como administrador, quiero ver la estadística de soluciones, para poder premiar a mis consultores.
- Como administrador, quiero tener la chance de cambiar de rol a un cliente o consultor, en caso me equivoque al momento de registrarlos.
- Como administrador, necesito ver una advertencia antes de eliminar un cliente o incidencia, para poder asegurarme de que lo estoy haciendo bien.
- Como administrador, quiero ver las estadísticas de todas las incidencias (incluidas las eliminadas) para analizar el avance de los consultores.
- Como consultor, necesito ver solo las incidencias asignadas a mi persona, para poder solucionarlas sin interferir en el trabajo de otros compañeros.

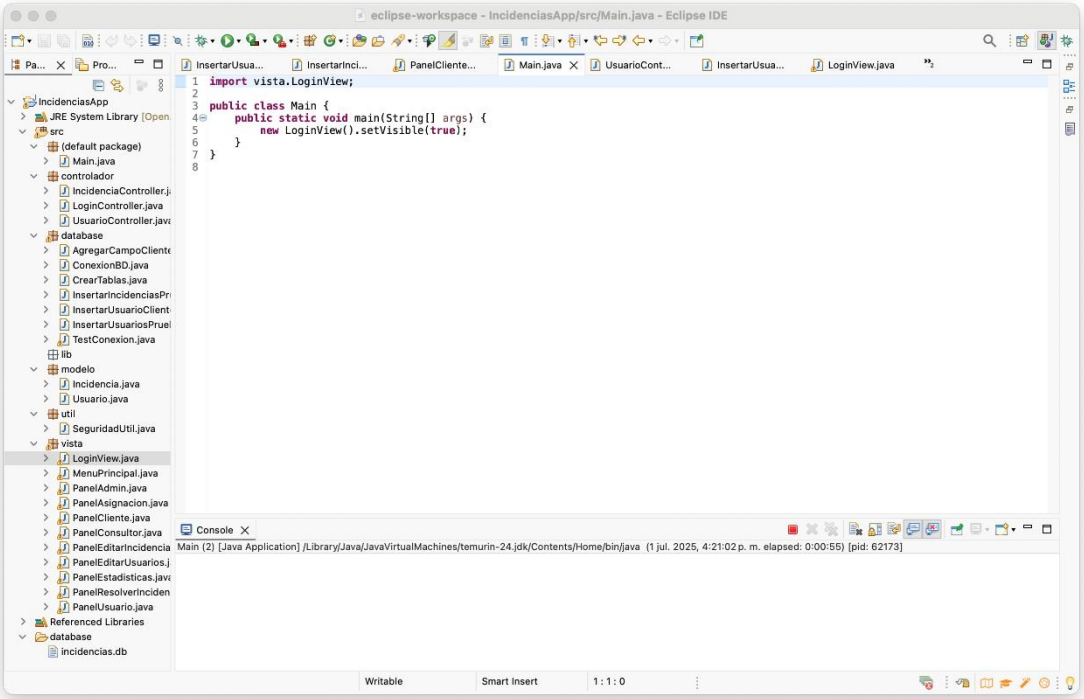
1.8. Descripción técnica del software realizado

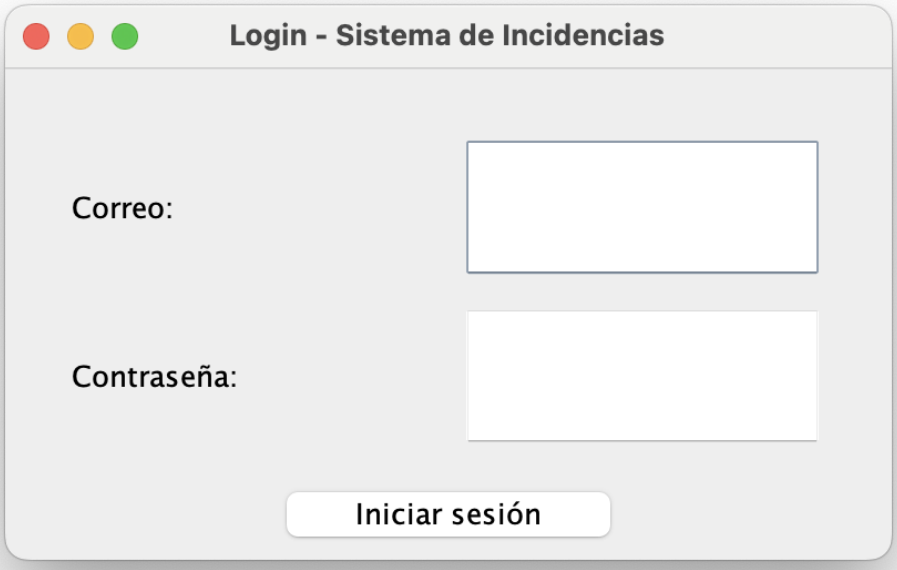
La empresa donde trabajo desarrollo un software ERP que ha comercializado a las empresas locales, las cuales a pesar de haber sido capacitados y habérsele proporcionado el manual, los mismos siguen teniendo inconvenientes de uso, en su mayoría por mal uso de software y pocas veces por errores netamente de código. Por lo cual, los clientes como tienen comunicación constante con los gerentes se comunican mediante llamadas o correos, indicando los mismos inconvenientes a la espera de ser solucionados, por lo que este software de gestión de incidencias se implementa con la finalidad de tener registrado cada una de las incidencias e inconvenientes de los clientes al momento de usar el software incluidas las dudas sobre el uso de las mismas.

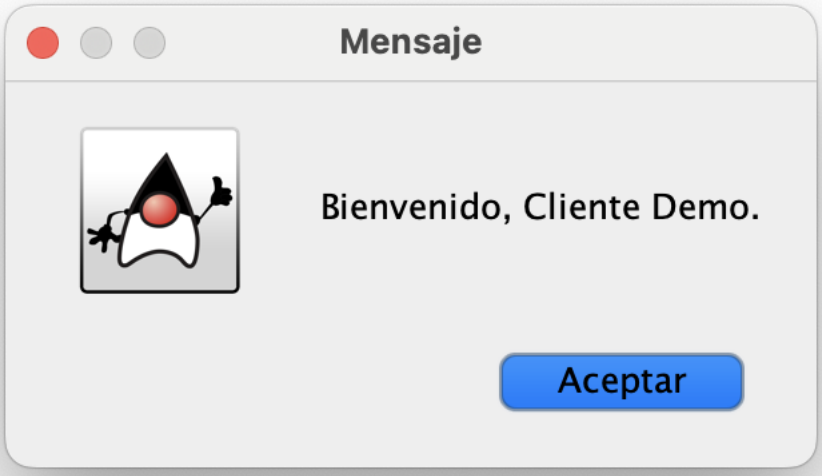
1.9. Pruebas y validaciones

Se realizaron pruebas a nivel de usuario con tipo de rol: administrador, consultor y cliente para asegurar que el software, la gestión de palabras en el arreglo, incluyendo el registro, las listas y las condicionales, funcionaran correctamente. Los resultados confirmaron que cada funcionalidad cumple con su propósito de manera eficiente, garantizando el manejo adecuado del software y los requerimientos solicitados.

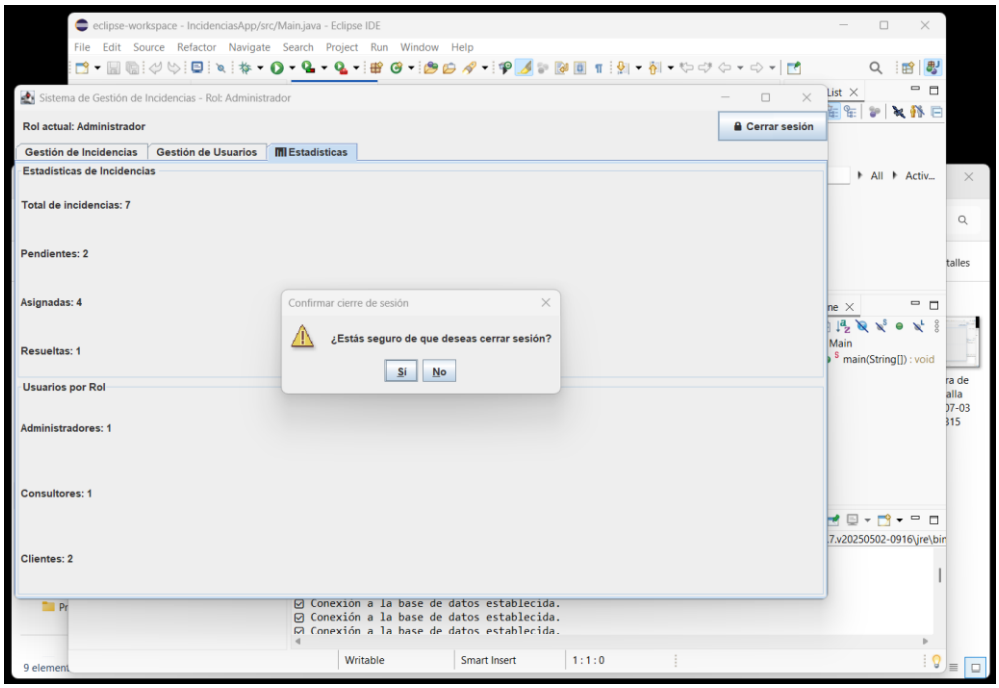
1.10. Evidencias del software en funcionamiento

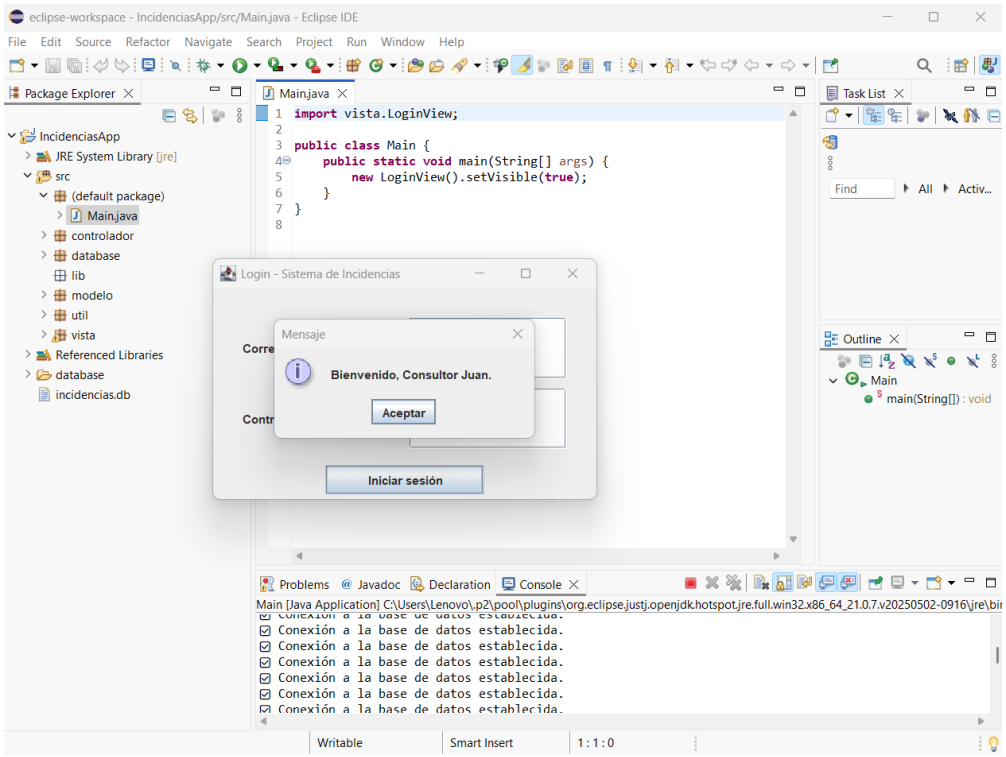


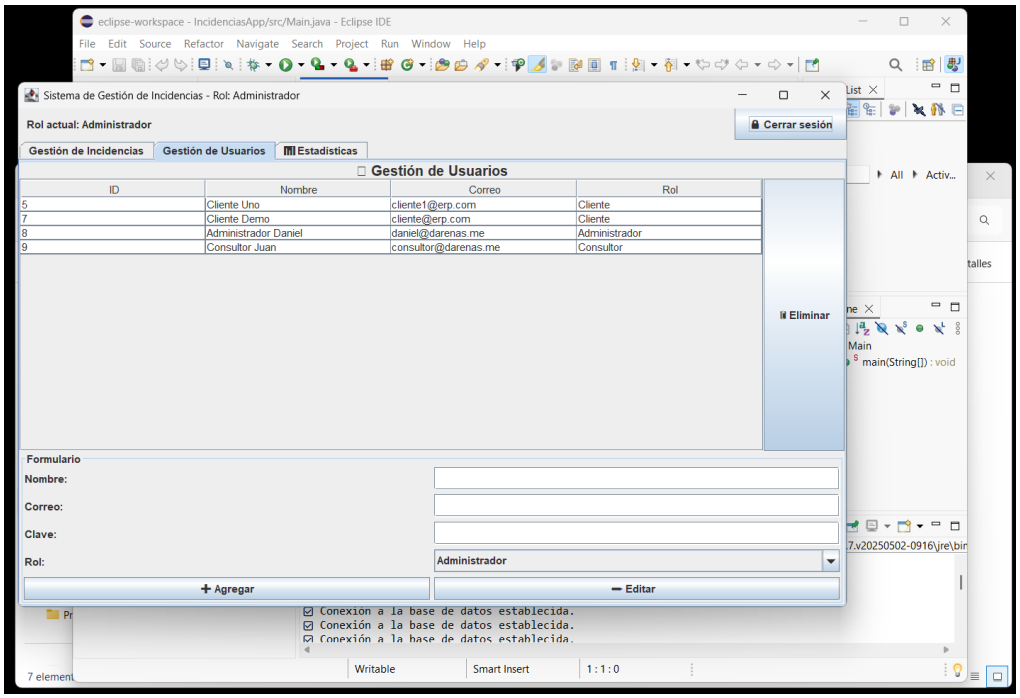


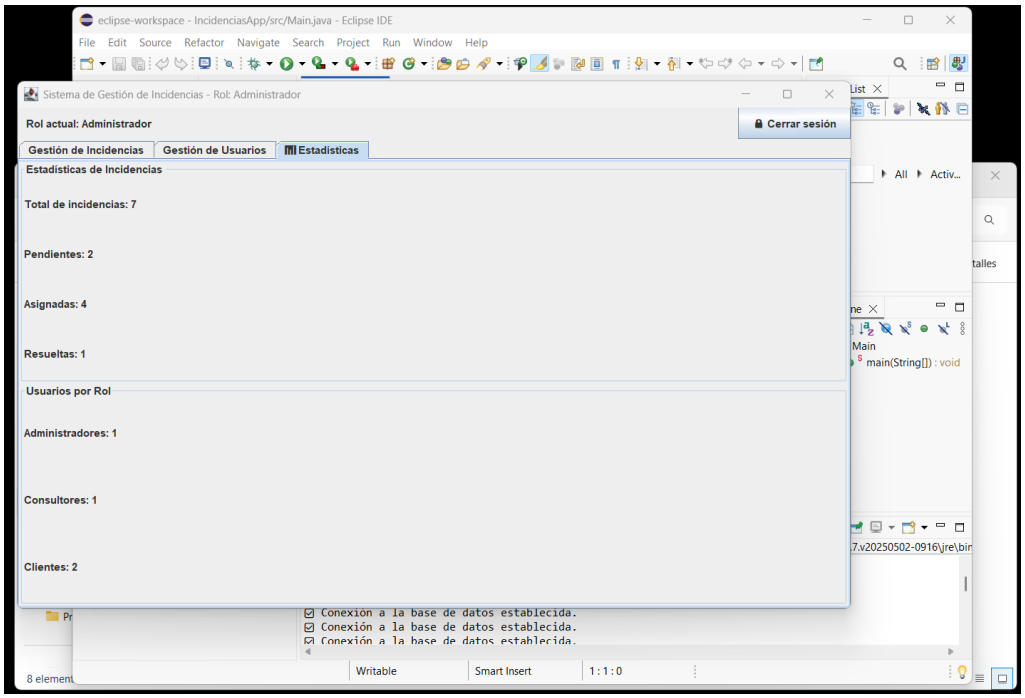

Mensaje
Bienvenido, Cliente Demo.
Aceptar

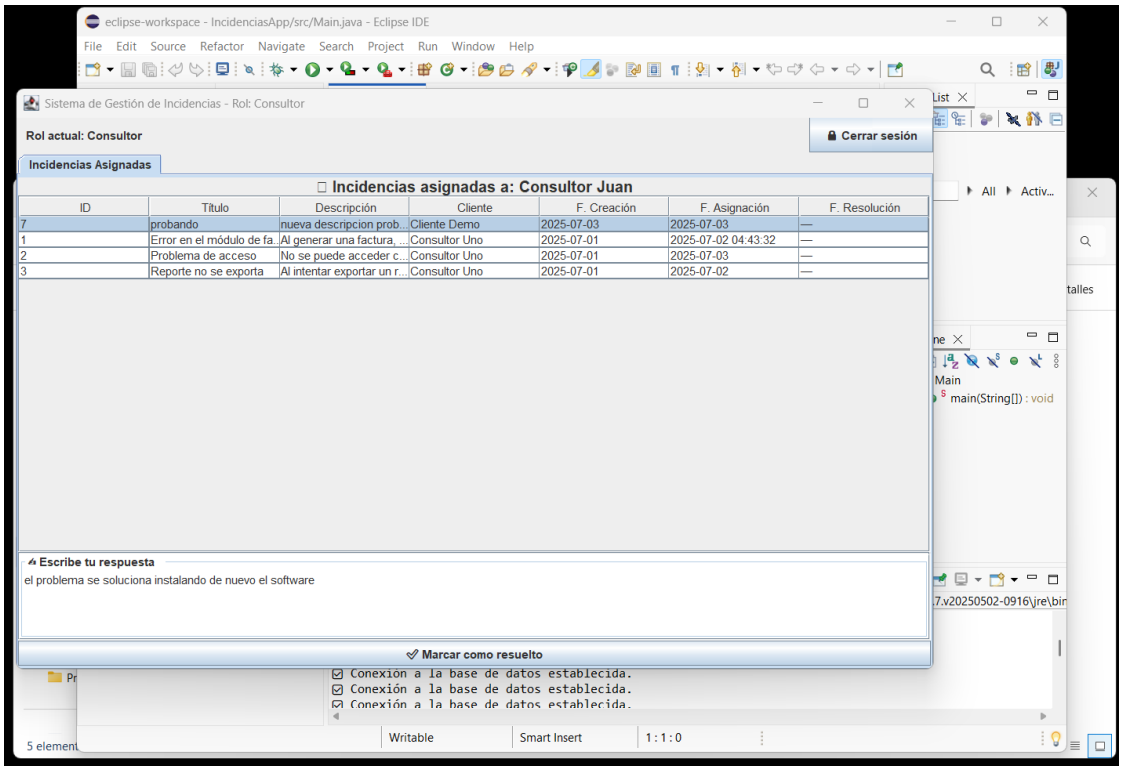

Sistema de Gestión de Incidencias - Rol: Cliente
Rol actual: Cliente
Cerrar sesión
Mis Incidencias
ID
Título
Descripción
Estado
F. Creación
F. Asignación
F. Resolución
+ Registrar nueva incidencia

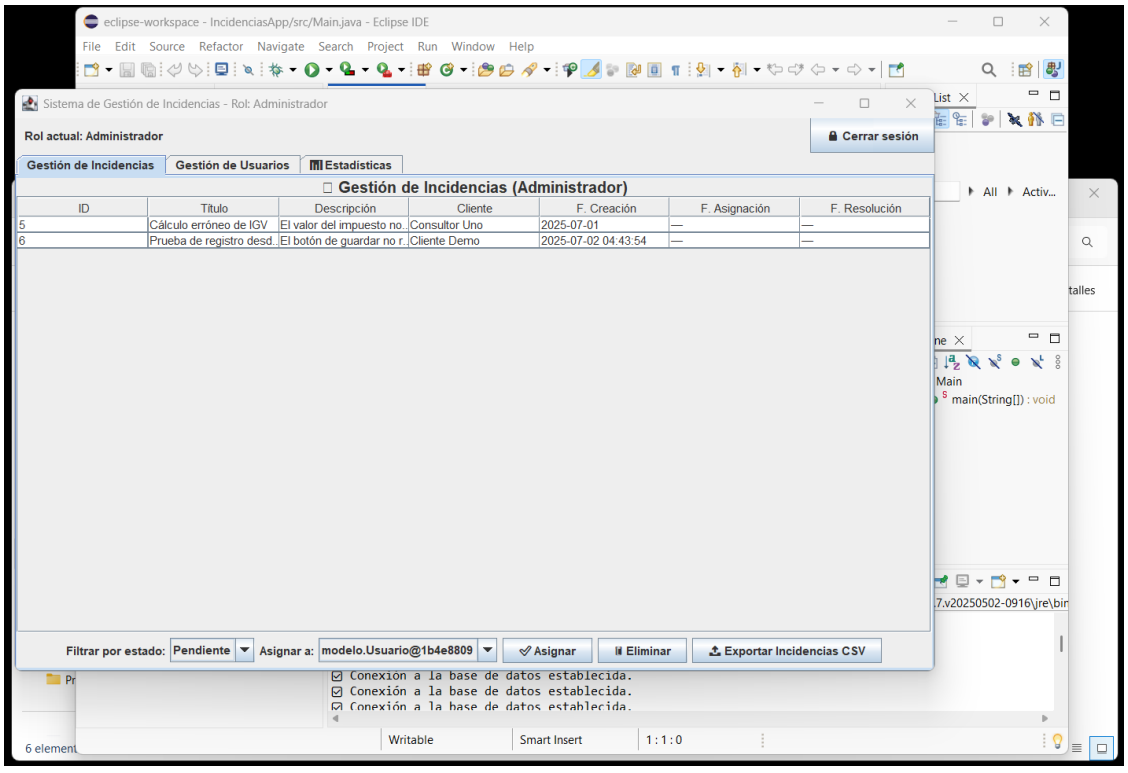

eclipse-workspace - IncidenciasApp/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Sistema de Gestión de Incidencias - Rol: Administrador
Rol actual: Administrador
Cerrar sesión
Gestión de Incidencias
Gestión de Usuarios
Estadísticas
Estadísticas de Incidencias
Total de incidencias: 7
Pendientes: 2
Asignadas: 4
Resueltas: 1
Usuarios por Rol
Administradores: 1
Consultores: 1
Clientes: 2
Confirmar cierre de sesión
¿Estás seguro de que deseas cerrar sesión?
Si
No
Conexión a la base de datos establecida.
Conexión a la base de datos establecida.
Writable
Smart Insert
1 : 1 : 0

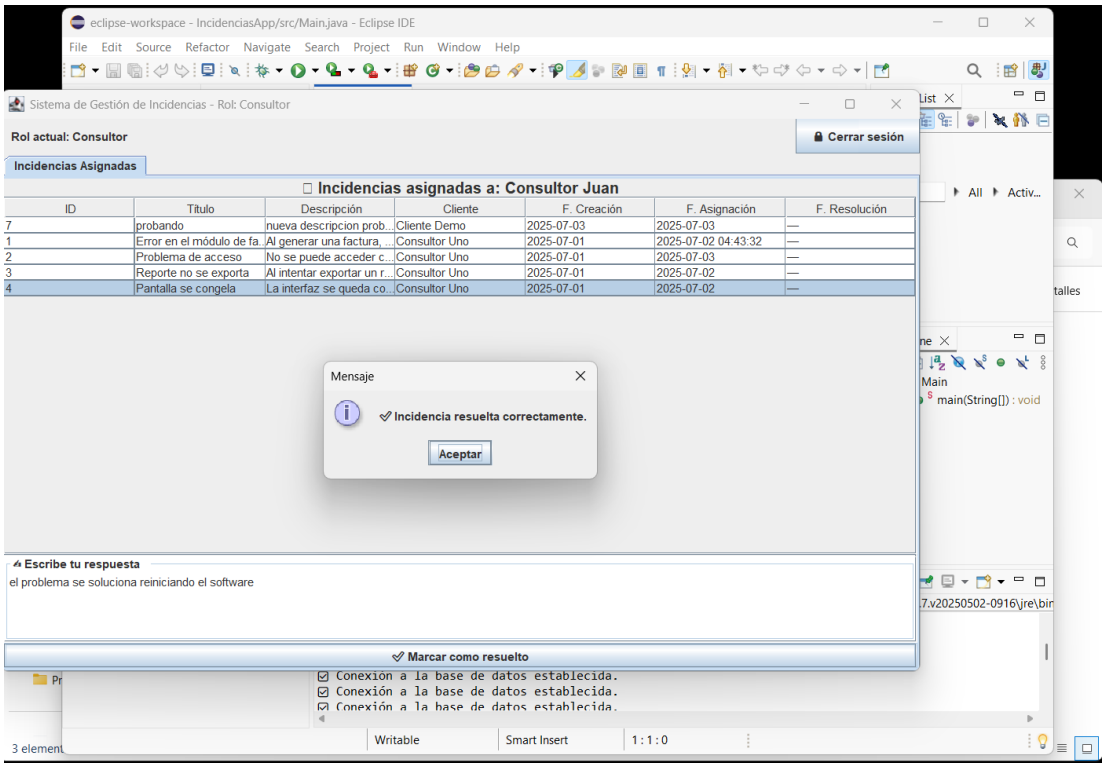

eclipse-workspace - IncidenciasApp/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
IncidenciasApp
JRE System Library [jre]
src
(default package)
Main.java
controlador
database
lib
modelo
util
vista
Referenced Libraries
database
incidencias.db
Main.java
1 import vista.LoginView;
2
3 public class Main {
4 public static void main(String[] args) {
5 new LoginView().setVisible(true);
6 }
7 }
8
Login - Sistema de Incidencias
Mensaje
Bienvenido, Consultor Juan.
Aceptar
Iniciar sesión
Task List
Outline
Main
main(String[]) : void
Problems
Javadoc
Declaration
Console
Main [Java Application] C:\Users\Lenovo\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0916\jre\bi
Conexión a la base de datos establecida.
Conexión a la base de datos establecida.
Conexión a la base de datos establecida.
Conexión a la base de datos establecida.
Conexión a la base de datos establecida.
Writable
Smart Insert
1 : 1 : 0

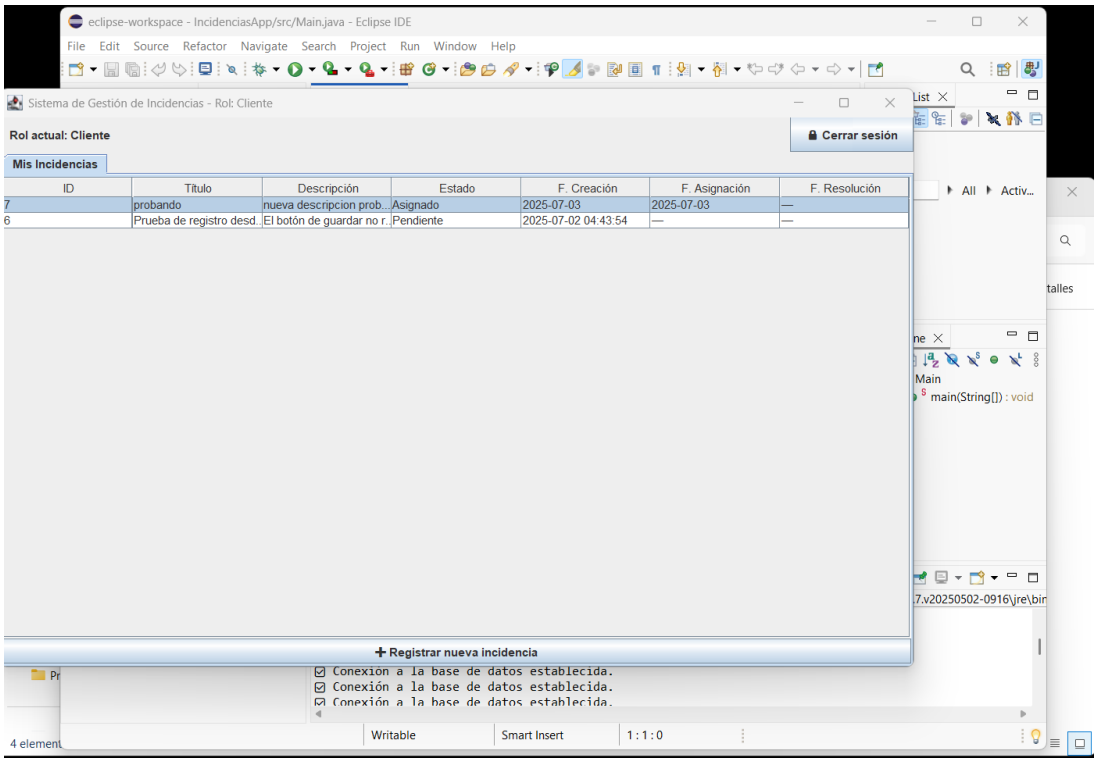

eclipse-workspace - IncidenciasApp/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Sistema de Gestión de Incidencias - Rol: Administrador
Rol actual: Administrador
Cerrar sesión
Gestión de Incidencias
Gestión de Usuarios
Estadísticas
Gestión de Usuarios
ID
Nombre
Correo
Rol
5
Cliente Uno
cliente1@erp.com
Cliente
7
Cliente Demo
cliente@erp.com
Cliente
8
Administrador Daniel
daniel@darenas.me
Administrador
9
Consultor Juan
consultor@darenas.me
Consultor
Eliminar
Formulario
Nombre:
Correo:
Clave:
Rol:
Administrador
Agregar
Editar
Conexion a la base de datos establecida.
Conexión a la base de datos establecida.
Conexión a la base de datos establecida.
Writable
Smart Insert
1 : 1 : 0
7 element

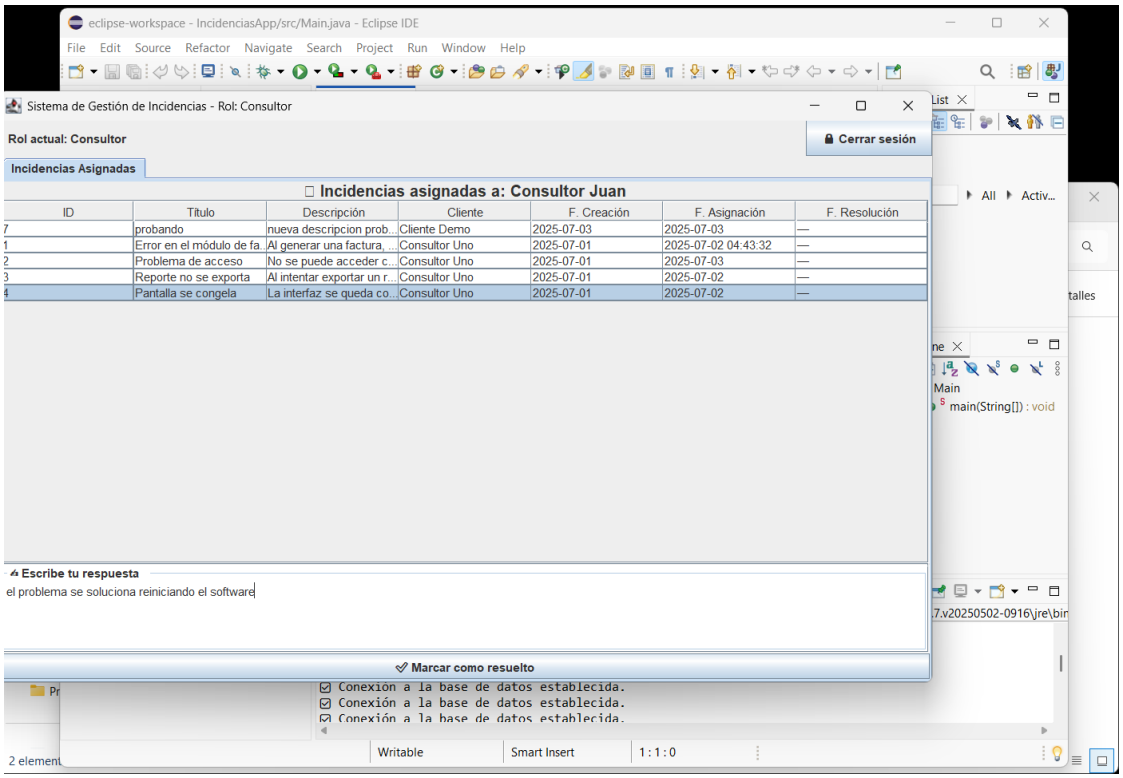

eclipse-workspace - IncidenciasApp/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Sistema de Gestión de Incidencias - Rol: Administrador
Rol actual: Administrador
Cerrar sesión
Gestión de Incidencias
Gestión de Usuarios
Estadísticas
Estadísticas de Incidencias
Total de incidencias: 7
Pendientes: 2
Asignadas: 4
Resueltas: 1
Usuarios por Rol
Administradores: 1
Consultores: 1
Clientes: 2
Conexion a la base de datos establecida.
Conexión a la base de datos establecida.
Conexión a la base de datos establecida.
Writable
Smart Insert
1 : 1 : 0
8 element

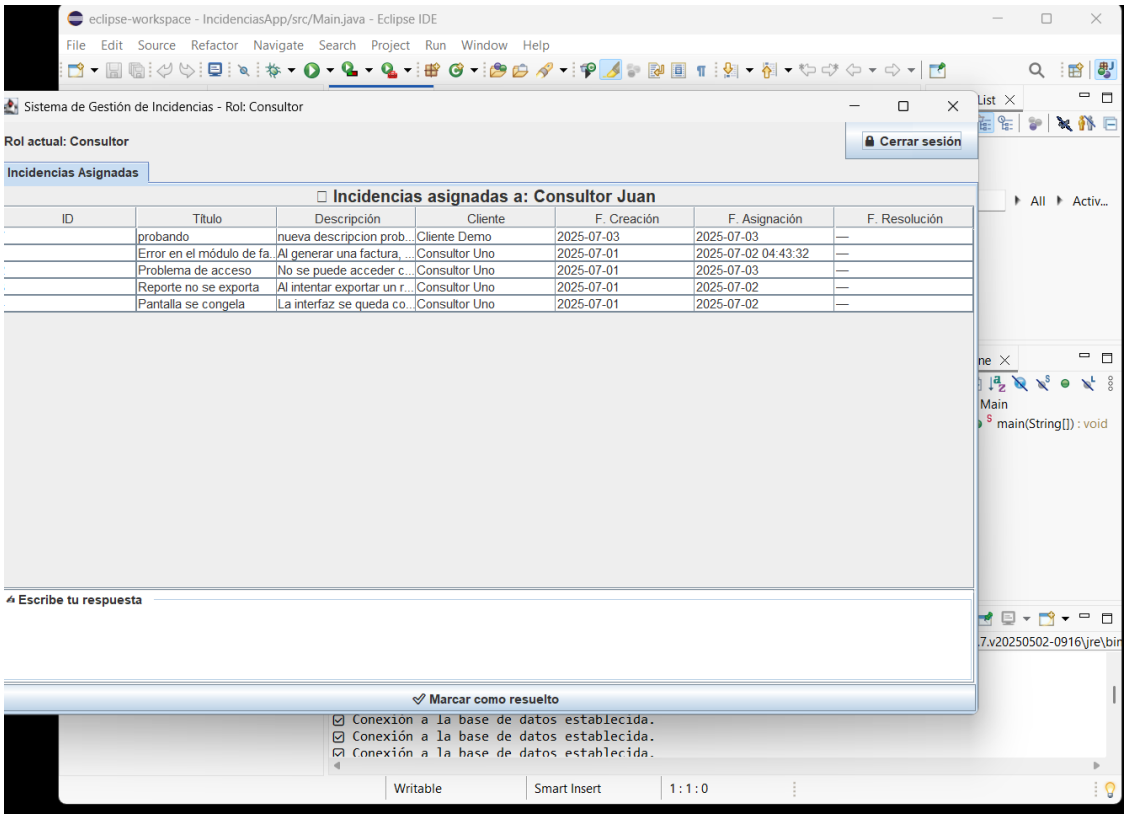

eclipse-workspace - IncidenciasApp/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Sistema de Gestión de Incidencias - Rol: Consultor
Rol actual: Consultor
Cerrar sesión
Incidencias Asignadas
Incidencias asignadas a: Consultor Juan
ID | Título | Descripción | Cliente | F. Creación | F. Asignación | F. Resolución
7 | probando | nueva descripcion prob... | Cliente Demo | 2025-07-03 | 2025-07-03 | —
1 | Error en el módulo de fa... | Al generar una factura, ... | Consultor Uno | 2025-07-01 | 2025-07-02 04:43:32 | —
2 | Problema de acceso | No se puede acceder c... | Consultor Uno | 2025-07-01 | 2025-07-03 | —
3 | Reporte no se exporta | Al intentar exportar un r... | Consultor Uno | 2025-07-01 | 2025-07-02 | —
Escribe tu respuesta
el problema se soluciona instalando de nuevo el software
Marcar como resuelto
Conexion a la base de datos establecida.
Conexión a la base de datos establecida.
Conexión a la base de datos establecida.
Writable
Smart Insert
1 : 1 : 0


eclipse-workspace - IncidenciasApp/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Sistema de Gestión de Incidencias - Rol: Administrador
Rol actual: Administrador
Cerrar sesión
Gestión de Incidencias
Gestión de Usuarios
Estadísticas
Gestión de Incidencias (Administrador)
ID | Título | Descripción | Cliente | F. Creación | F. Asignación | F. Resolución
5 | Cálculo erróneo de IGV | El valor del impuesto no... | Consultor Uno | 2025-07-01 | — | —
6 | Prueba de registro desd... | El botón de guardar no r... | Cliente Demo | 2025-07-02 04:43:54 | — | —
Filtrar por estado: Pendiente
Asignar a: modelo.Usuario@1b4e8809
Asignar
Eliminar
Exportar Incidencias CSV
Conexion a la base de datos establecida.
Conexión a la base de datos establecida.
Conexión a la base de datos establecida.
Writable
Smart Insert
1 : 1 : 0


eclipse-workspace - IncidenciasApp/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Sistema de Gestión de Incidencias - Rol: Consultor
Rol actual: Consultor
Cerrar sesión
Incidencias Asignadas
Incidencias asignadas a: Consultor Juan
ID | Título | Descripción | Cliente | F. Creación | F. Asignación | F. Resolución
7 | probando | nueva descripcion prob... | Cliente Demo | 2025-07-03 | 2025-07-03 | —
1 | Error en el módulo de fa... | Al generar una factura, ... | Consultor Uno | 2025-07-01 | 2025-07-02 04:43:32 | —
2 | Problema de acceso | No se puede acceder c... | Consultor Uno | 2025-07-01 | 2025-07-03 | —
3 | Reporte no se exporta | Al intentar exportar un r... | Consultor Uno | 2025-07-01 | 2025-07-02 | —
4 | Pantalla se congela | La interfaz se queda co... | Consultor Uno | 2025-07-01 | 2025-07-02 | —
Mensaje
Incidencia resuelta correctamente.
Aceptar
Escribe tu respuesta
el problema se soluciona reiniciando el software
Marcar como resuelto
Conexión a la base de datos establecida.
Conexión a la base de datos establecida.
Writable
Smart Insert
1 : 1 : 0


eclipse-workspace - IncidenciasApp/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Sistema de Gestión de Incidencias - Rol: Cliente
Rol actual: Cliente
Cerrar sesión
Mis Incidencias
ID | Título | Descripción | Estado | F. Creación | F. Asignación | F. Resolución
7 | probando | nueva descripcion prob... | Asignado | 2025-07-03 | 2025-07-03 | —
6 | Prueba de registro desd... | El botón de guardar no r... | Pendiente | 2025-07-02 04:43:54 | — | —
Registrar nueva incidencia
Conexión a la base de datos establecida.
Conexión a la base de datos establecida.
Writable
Smart Insert
1 : 1 : 0


eclipse-workspace - IncidenciasApp/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Sistema de Gestión de Incidencias - Rol: Consultor
Rol actual: Consultor
Cerrar sesión
Incidencias Asignadas
Incidencias asignadas a: Consultor Juan
ID | Título | Descripción | Cliente | F. Creación | F. Asignación | F. Resolución
7 | probando | nueva descripcion prob... | Cliente Demo | 2025-07-03 | 2025-07-03 | —
1 | Error en el módulo de fa... | Al generar una factura, ... | Consultor Uno | 2025-07-01 | 2025-07-02 04:43:32 | —
2 | Problema de acceso | No se puede acceder c... | Consultor Uno | 2025-07-01 | 2025-07-03 | —
3 | Reporte no se exporta | Al intentar exportar un r... | Consultor Uno | 2025-07-01 | 2025-07-02 | —
4 | Pantalla se congela | La interfaz se queda co... | Consultor Uno | 2025-07-01 | 2025-07-02 | —
Escribe tu respuesta
el problema se soluciona reiniciando el software
Marcar como resuelto
Conexión a la base de datos establecida.
Conexión a la base de datos establecida.
Writable
Smart Insert
1 : 1 : 0


eclipse-workspace - IncidenciasApp/src/Main.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Sistema de Gestión de Incidencias - Rol: Consultor
Rol actual: Consultor
Cerrar sesión
Incidencias Asignadas
Incidencias asignadas a: Consultor Juan
ID | Título | Descripción | Cliente | F. Creación | F. Asignación | F. Resolución
probando | nueva descripcion prob... | Cliente Demo | 2025-07-03 | 2025-07-03 | —
Error en el módulo de fa... | Al generar una factura, ... | Consultor Uno | 2025-07-01 | 2025-07-02 04:43:32 | —
Problema de acceso | No se puede acceder c... | Consultor Uno | 2025-07-01 | 2025-07-03 | —
Reporte no se exporta | Al intentar exportar un r... | Consultor Uno | 2025-07-01 | 2025-07-02 | —
Pantalla se congela | La interfaz se queda co... | Consultor Uno | 2025-07-01 | 2025-07-02 | —
Escribe tu respuesta
Marcar como resuelto
Conexión a la base de datos establecida.
Conexión a la base de datos establecida.
Writable
Smart Insert
1 : 1 : 0

CAPITULO 3: CONCLUSIONES Y RECOMENDACIONES

1. Conclusiones

1.1. La utilidad de las clases

Las clases son fundamentales en la programación orientada a objetos, ya que permiten organizar y encapsular datos y comportamientos relacionados. Al definir clases, se promueve un enfoque más estructurado y comprensible en el desarrollo del software.

1.2. La utilidad de funciones

Las funciones son una piedra angular de la programación. Ayudan a crear código limpio, reutilizable, fácil de mantener y escalar. Al usarlas correctamente, los programadores pueden producir software más eficiente y confiable.

1.3. Cuando utilizar el manejo de excepciones

Implementar un manejo de excepciones adecuado permite anticipar fallos y ofrecer una experiencia de usuario más fluida al manejar errores de manera controlada.

1.4. Utilidad de la recursividad

La recursividad es una técnica poderosa que permite resolver problemas complejos de manera elegante y concisa. En este trabajo, se ha utilizado para implementar algoritmos de búsqueda y ordenación, demostrando su capacidad para simplificar el código y mejorar la legibilidad. Sin embargo, es importante ser consciente de su uso, ya que puede llevar a un mayor consumo de memoria si no se maneja adecuadamente.

1.5. Utilidad de array unidimensionales

Los arrays unidimensionales (o vectores en algunos lenguajes) son estructuras de datos fundamentales en la programación. Su utilidad se extiende a muchos aspectos del desarrollo de software, ya que permiten almacenar colecciones de datos de forma eficiente y organizada.

2. Recomendaciones

Adoptar Buenas Prácticas de Programación: Se recomienda seguir las buenas prácticas de programación, como la escritura de código limpio y bien comentado, para facilitar su mantenimiento y comprensión a largo plazo.

Implementar Pruebas: Incluir pruebas unitarias y funcionales en el ciclo de desarrollo para identificar y corregir errores de manera temprana, asegurando que el software cumpla con los requisitos establecidos.

Fomentar la Colaboración: Continuar promoviendo un ambiente de trabajo colaborativo donde se valore el intercambio de ideas y se aprovechen las fortalezas de cada miembro del equipo.

3. Referencias

Java YA. *Recursos para aprender JAVA en ECLIPSE*. https://www.tutorialesprogramacionya.com/javaya/

NetMentor. *Recursividad en programación*. https://www.netmentor.es/entrada/Recursividad-programacion

UPN BlackBoard. *Métodos de sobrecarga, Static, Excepciones y Colecciones.* https://upn.blackboard.com/ultra/courses/_1602368_1/outline/file/_53724816_1

El patrón modelo-vista-controlador: Arquitectura y frameworks explicados https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/

Modelo-Vista-Controlador (MVC) https://uetitc.github.io/ProgrammingII-2024-2/Lessons/05-MVC.html

Programación Orientada a Objetos: Conceptos y Fundamentos https://saberpunto.com/programacion/programacion-orientada-a-objetos-que-es/

LINK a repositorio en GitHub: https://github.com/arenas1802/Practicadecampo3