Informe Proyecto SIA Final – Gestión de turnos de enfermeras

Asignatura: INF2236 – Programación Avanzada

Semestre: 2025-2

Profesores: Claudio Cubillos, Leonardo González Catalán

Integrantes del grupo: Joaquín Muñoz, Ignacio Reyes, Benjamín Pizarro

Lenguaje: Java (Oracle JDK 11)

# Introducción

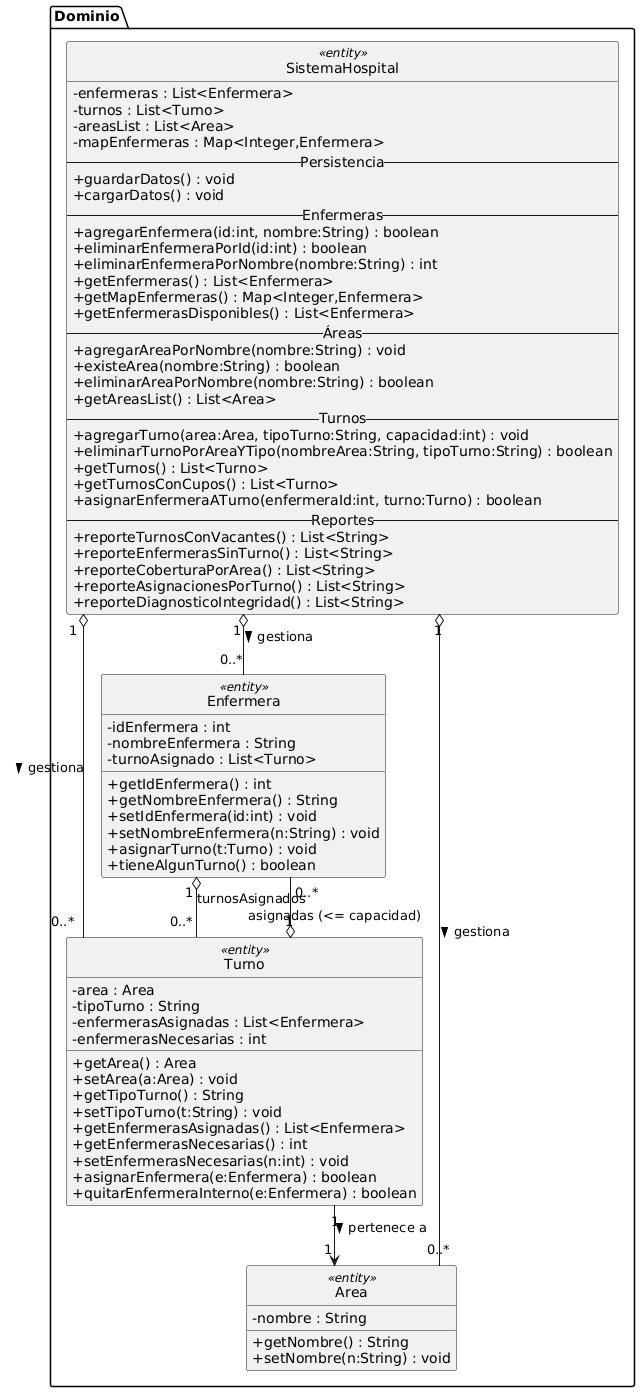
En este informe se presenta el desarrollo del Sistema de Información Final (SIA), creado en Java como parte del proyecto académico. El sistema tiene como objetivo facilitar la gestión hospitalaria, permitiendo registrar y organizar de manera sencilla las enfermeras, las áreas y los turnos, además de realizar asignaciones y generar reportes útiles para el control de la información.

Para asegurar que los datos no se pierdan, se implementó un mecanismo de persistencia en archivos CSV, que guarda la información al salir y la carga al iniciar la aplicación. También se desarrolló una interfaz gráfica con JavaFX, lo que hace que la interacción con el usuario sea más práctica e intuitiva, en contraste con los clásicos sistemas de consola.

Este proyecto refleja la aplicación de distintos contenidos aprendidos en el curso, como programación orientada a objetos, persistencia de datos, manejo de excepciones y diseño de interfaces, integrados en una solución que simula un escenario real de gestión hospitalaria.

# SIA2.1 – Diseño de diagrama de clases UML

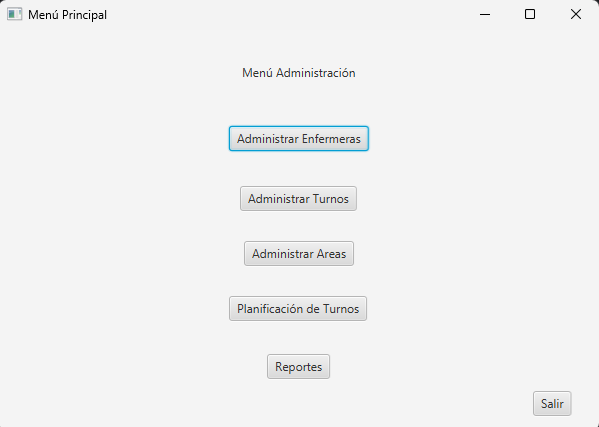
Se diseñó un diagrama de clases UML que modela las entidades principales:  
- Enfermera: contiene ID, nombre y turnos asignados.  
- Turno: pertenece a un área, define un horario y capacidad de enfermeras.  
- Área: representa la unidad hospitalaria.  
- SistemaHospital: clase central que administra colecciones de enfermeras, áreas y turnos.  
- Reporte: permite generar salidas textuales sobre turnos, coberturas y disponibilidad.



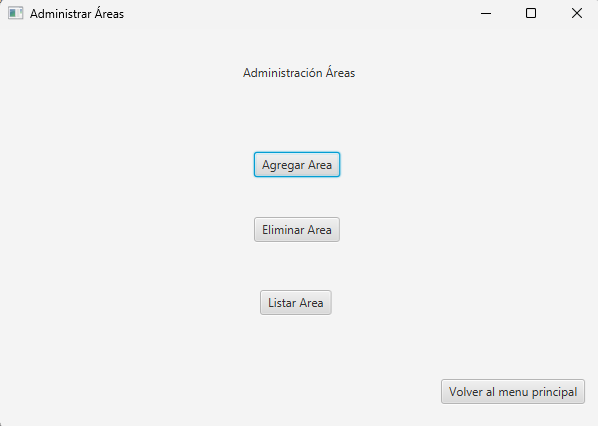
# SIA2.2 – Persistencia de datos (batch)

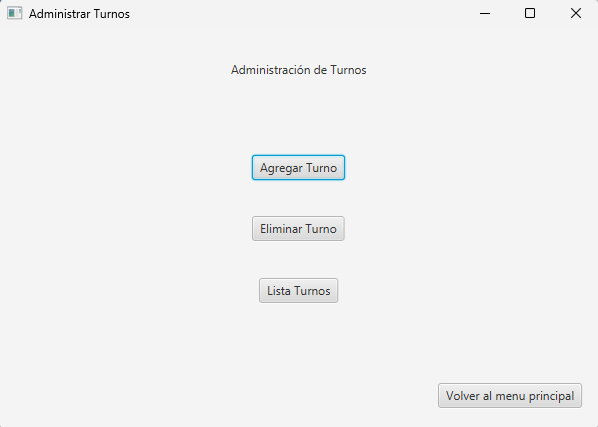
El sistema persiste en archivo hospital.csv:  
- Al iniciar: se ejecuta cargarDatos().  
- Al cerrar: se ejecuta guardarDatos() desde HelloApplication.stop() y en el botón 'Salir'.  
Esto asegura un manejo tipo batch: una carga al inicio y un guardado al final.

# SIA2.3 – Interfaces gráficas

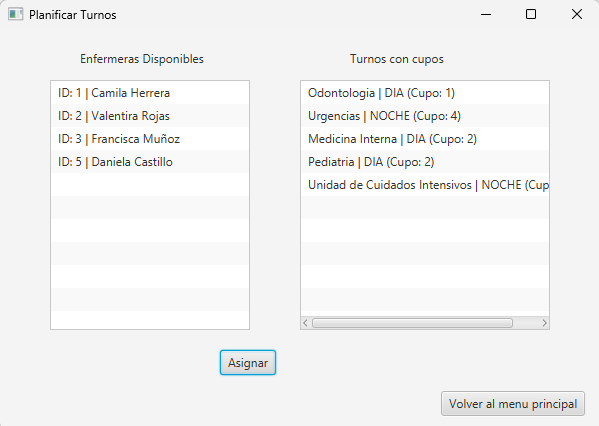
Se implementaron ventanas en JavaFX:  
- Menú principal.  
  
- Administración de enfermeras.  


- Administración de áreas.

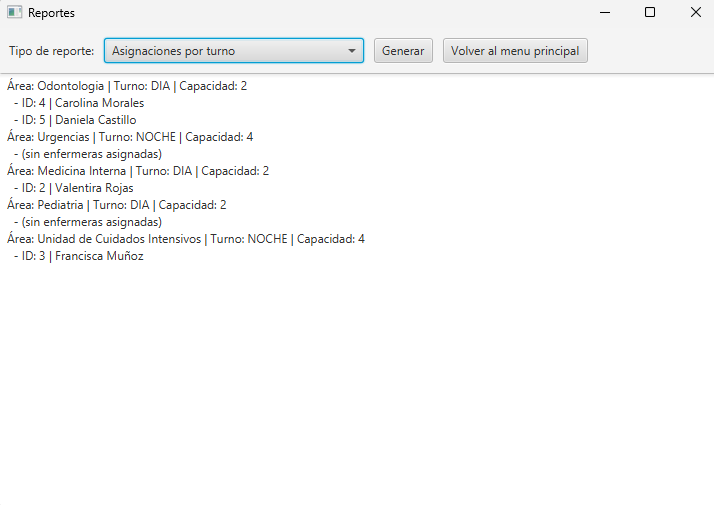
  
- Administración de turnos.



- Planificación de turnos.



- Reportes.



# SIA2.4 – Menú con edición/eliminación (2ª colección)

En este proyecto la colección anidada corresponde a las enfermeras que se asignan dentro de cada turno.

El sistema cuenta con un menú principal y varios submenús que permiten administrar enfermeras, áreas, turnos y también las asignaciones internas de los turnos.

**Colección interna (turnos con enfermeras):**

Se pueden asignar enfermeras disponibles a turnos que aún tengan cupos.

Al hacerlo, se actualizan las listas y se guarda en el archivo CSV.De forma opcional se podría permitir reemplazar enfermeras ya asignadas o ajustar la capacidad de un turno.

**Menú del sistema:**

**Enfermeras**: agregar (validando ID único), eliminar (por ID o nombre) y listar todas.

**Áreas**: agregar (sin duplicados), eliminar y listar.

**Turnos**: agregar (área + horario + capacidad), eliminar por datos ingresados y listar mostrando capacidad y enfermeras asignadas.

**Planificación**: ver enfermeras disponibles, turnos con vacantes y asignar enfermeras a turnos (con validaciones de cupos).

**Reportes**: incluye turnos con vacantes, enfermeras sin turno, cobertura por área, asignaciones detalladas por turno y un diagnóstico de integridad

**Persistencia:**

Los datos se cargan automáticamente al iniciar la aplicación desde hospital.csv

Se guardan en cada operación importante (agregar, eliminar, asignar).

Además, se asegura un guardado completo al salir de la aplicación, cumpliendo con la idea de persistencia “batch”.

Validaciones incluidas:

-IDs únicos para enfermeras.

-No se permiten áreas duplicadas por nombre.

-Las asignaciones respetan los cupos y evitan duplicados en los turnos.

# SIA2.5 – Funcionalidad propia

El sistema incorpora reportes y filtros que cumplen con la idea de obtener subconjuntos de datos según criterios específicos.  
Algunos ejemplos implementados:

Turnos con vacantes: muestra solamente aquellos turnos que aún no tienen completa su capacidad.  
Enfermeras sin turno: filtra a todas las enfermeras que todavía no están asignadas en ningún turno.  
Cobertura por área: agrupa los turnos por área y calcula el porcentaje de cobertura (asignadas/capacidad).

Estos reportes se generan directamente desde las colecciones principales (turnos, enfermeras y áreas) y sus relaciones.  
La idea es que no se muestren todos los datos sin más, sino que el usuario pueda enfocarse en casos de interés para la gestión del hospital.  
Con esto se cumple el requisito de ofrecer una funcionalidad de negocio distinta a las operaciones básicas de agregar, eliminar o listar.

# SIA2.6 – Modularización y buenas prácticas

El código está organizado en clases separadas según su responsabilidad:

-SistemaHospital: gestiona la lógica central, persistencia y colecciones principales.

-Enfermera, Area, Turno: representan las entidades del dominio.

-Controladores (AgregarEnfermeraController, EliminarTurnoController, PlanificacionController, etc.) → manejan la interacción de cada ventana con el modelo.

Esto permite mantener una clara separación entre la lógica de negocio y la interfaz gráfica.

Se utilizan métodos específicos y descriptivos, lo que facilita la lectura del código (ejemplo: eliminarTurnoPorAreaYTipo, reporteCoberturaPorArea).

Se manejan listas y mapas según la necesidad: mapEnfermeras para acceso rápido por ID, y listas para iteraciones generales.

La persistencia está encapsulada en los métodos guardarDatos() y cargarDatos(), evitando duplicación de código.  
En general, se sigue la idea de mantener el sistema modularizado, con responsabilidades claras y código legible.

# SIA2.7 – Sobreescritura de métodos

Se cumple el requisito de sobreescritura en varias partes del proyecto:

-En la clase Enfermera y en la clase Turno se redefinen métodos como toString() (si lo agregamos) para mostrar la información de manera más clara en listas o reportes.

-La clase Area también puede sobrescribir equals() y hashCode() para que las comparaciones se hagan en base al nombre del área, evitando duplicados en listas.

Ejemplos prácticos de sobreescritura:

-toString() en Turno: mostraría "Área: Urgencias | Turno: Noche | 2/3 enfermeras asignadas".

-equals() en Area: dos áreas con el mismo nombre se consideran iguales aunque sean objetos distintos.

Con esto se logra un comportamiento más coherente al listar, comparar y validar objetos, y se facilita el trabajo en colecciones (listas, mapas).

# SIA2.8 – Manejo de excepciones

El sistema maneja errores comunes mediante try–catch.

-En persistencia (guardar/cargar CSV) se capturan IOException para evitar que la app se detenga.

-En la ventana Agregar Enfermera se controla NumberFormatException cuando el ID no es numérico, mostrando un mensaje al usuario.  
En los controladores se valida que los campos no estén vacíos y que haya selecciones

-válidas, evitando errores como IndexOutOfBoundsException.  
En la carga de interfaces FXML se atrapan IOException y se notifica al usuario.

De esta forma se asegura estabilidad y una mejor experiencia frente a entradas o fallos inesperados.

SIA2.9 – Excepciones personalizadas  
- EnfermeraDuplicadaException.  
- TurnoNoEncontradoException.

# SIA2.10 – Reportes en archivo

El sistema genera reportes en consola y listas (ej: reporteTurnosConVacantes). Se puede agregar un método que exporte estos reportes a archivo TXT.

# SIA2.11 – Uso de GitHub

El proyecto fue versionado en GitHub con commits adicionales tras el avance. Aquí se puede incluir evidencia con capturas de los commits.

<https://github.com/deftonenjoyer/Gesti-n-de-turnos-de-enfermeras>

# SIA2.12 – CRUD en colección principal

El sistema permite realizar operaciones de CRUD sobre la colección principal de objetos:

-Agregar enfermeras, áreas y turnos mediante formularios con validación.

-Eliminar elementos de las mismas colecciones de manera controlada, con persistencia en archivo CSV.

-Modificar datos: por ejemplo, actualizar la capacidad de un turno o reasignar enfermeras entre turnos.

Estas operaciones garantizan la gestión completa de los elementos de primer nivel del sistema.

# SIA2.13 – Buscar elementos

Se implementaron métodos de búsqueda para localizar elementos específicos dentro de las colecciones:

-Búsqueda de enfermeras por ID o nombre.

-Verificación de existencia de un área por su nombre.

-Identificación de turnos disponibles (con cupos) o enfermeras libres (sin turno asignado).

Estas búsquedas permiten filtrar y recuperar información rápidamente, mejorando la usabilidad del sistema.

# Conclusiones

En conclusión, el proyecto permitió integrar lo aprendido durante el curso en un sistema funcional para la gestión de un hospital, con manejo de enfermeras, áreas y turnos, persistencia de datos y reportes útiles. Se cumplieron los objetivos planteados y se logró un programa práctico que puede seguir mejorándose en el futuro.