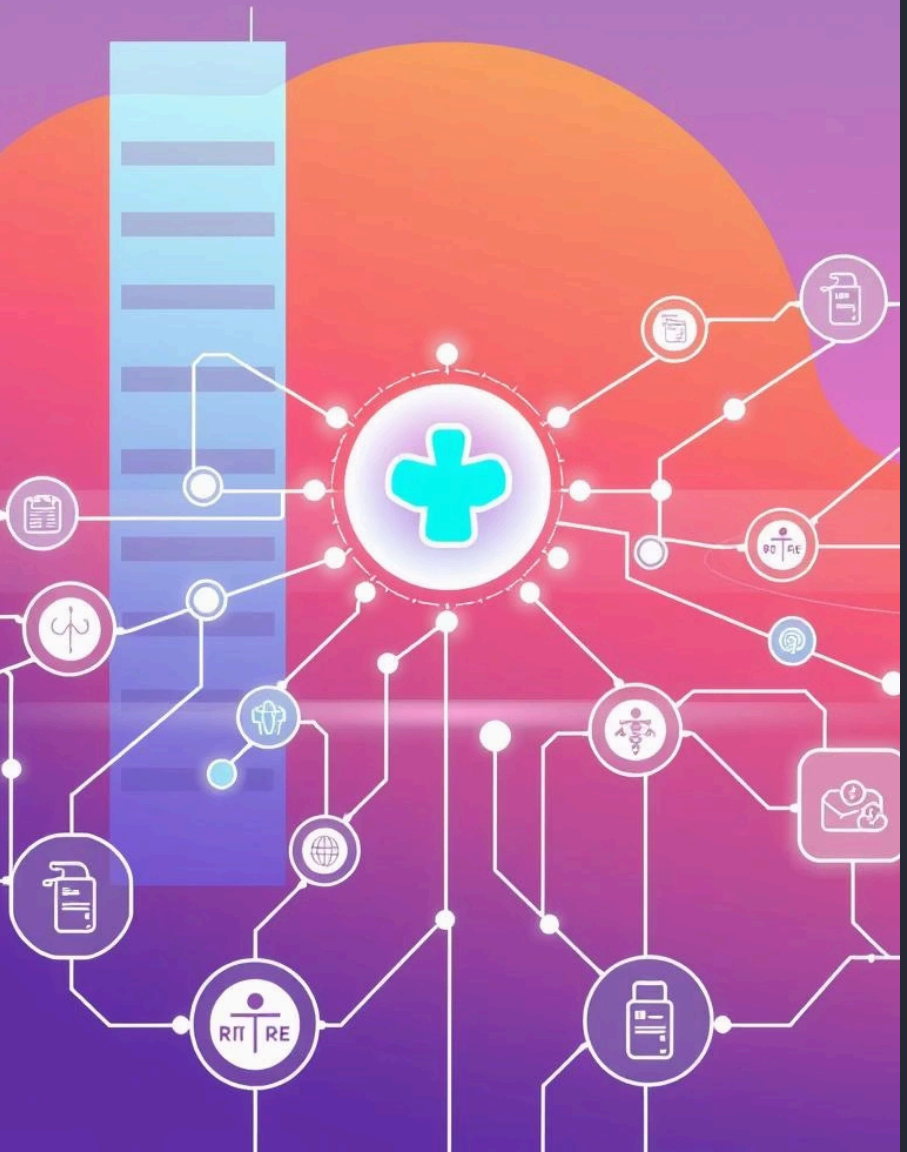


Estructuras de Datos Aplicadas en Sistemas Hospitalarios

Una aproximación para optimizar la gestión de pacientes y recursos en centros de salud mediante la informática.

Integrantes:

- DAVID TTITO CRUZ
- JURGEN MOLERO RIVERA
- SEBASTIAN JORGE SERRANO MENDOZA

Docente: Yesenia Concha Ramos

Desafíos Críticos en la Gestión Hospitalaria

La eficiencia operativa en entornos hospitalarios es fundamental para salvar vidas. Los sistemas actuales a menudo fallan debido a la falta de estructura digital robusta, generando cuellos de botella y errores críticos.

Colas de Espera Ineficientes

Desorganización en la atención que no prioriza la urgencia médica, llevando a tiempos de espera peligrosamente largos.

Registro Manual Propenso a Errores

La dependencia del papel y la entrada manual de datos incrementa las tasas de error, afectando la precisión del diagnóstico y tratamiento.

Historial Médico Disperso

Dificultad para acceder rápidamente al historial completo de atenciones del paciente, lo que ralentiza la toma de decisiones críticas.

Gestión Ineficiente de Recursos

Dificultad para asignar personal, salas y equipos de manera óptima según la demanda y prioridad en tiempo real.

Una necesidad crítica es desarrollar un sistema automatizado que garantice el flujo eficiente y la priorización médica correcta.

🏗️ Estructuras de Datos Clave para la Arquitectura del Sistema

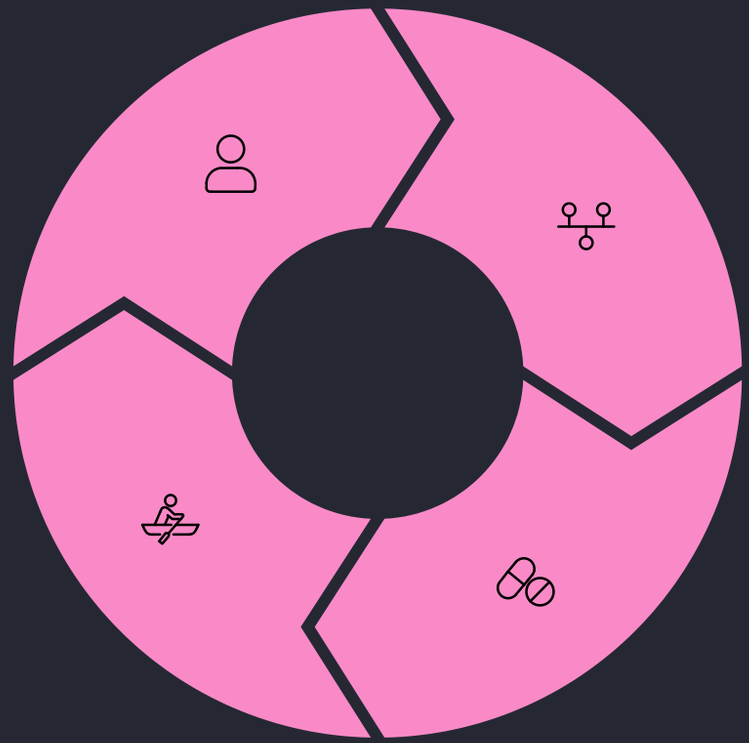
Hemos seleccionado estructuras de datos fundamentales para modelar y gestionar los distintos componentes del flujo hospitalario, asegurando eficiencia y accesibilidad.

Lista Enlazada (Pacientes)

Manejo de la **base de datos maestra** de pacientes registrados. Permite inserción rápida de nuevos registros (O(1) al inicio).

Cola (Espera y Triage)

Manejo de pacientes en espera. Permite **encolar** y **desencolar** en O(1), con un algoritmo de ordenamiento que aplica la priorización.



Pila (Historial Clínico)

Gestión del historial de atenciones de cada paciente. Las operaciones de **apilar** y **desapilar** son rápidas, reflejando el último evento primero (O(1)).

Lista Simple (Inventario)

Catálogo de **medicamentos y suministros**. Se prioriza el recorrido y la búsqueda para verificar la existencia (O(n)).

Complejidad de Operaciones

La elección de estructuras se basa en la complejidad temporal: inserción en lista O(1), manejo de cola O(1). La operación más costosa es **buscarPorDNI** o **ordenarPorPrioridad** (O(n) o $O(n^2)$), que requieren análisis cuidadoso para su optimización.

⚡ Algoritmos Centrales: Priorización y Gestión del Flujo

Dos algoritmos son cruciales para el sistema: el flujo de entrada de pacientes y el mecanismo de priorización de la cola de espera, este último basado en un ordenamiento por inserción para mantener la estabilidad.

Agregar Paciente a Cola

Este proceso garantiza que solo los pacientes registrados y validados sean admitidos a la cola de atención.

```
ALGORITMO agregarPacienteACola
ENTRADAS: dni (string)
INICIO
  paciente ← lista.buscarPorDNI(dni)
  SI paciente ≠ NULO ENTONCES
    paciente.estado ← "En espera"
    cola.encolar(paciente)
    MOSTRAR "Paciente agregado..."
  SINO
    MOSTRAR "Error: Paciente no encontrado"
  FIN SI
FIN
```

Ordenar por Prioridad (Algoritmo de Inserción)

El uso de un algoritmo $O(n^2)$ como la inserción es aceptable para colas pequeñas a medianas, comunes en una sala de emergencias, debido a su simplicidad e implementación directa.

```
ALGORITMO ordenarPorPrioridad
INICIO
  // Copia la cola a un arreglo temporal
  arregloTemp ← [cola.datos]
  // Ordenamiento por inserción
  PARA i ← 1 HASTA contador-1 HACER
    clave ← arregloTemp[i]
    j ← i - 1
    MIENTRAS j ≥ 0 Y arregloTemp[j].prioridad >
clave.prioridad HACER
      arregloTemp[j+1] ← arregloTemp[j]
      j ← j - 1
    FIN MIENTRAS
    arregloTemp[j+1] ← clave
  FIN PARA
  // Reconstruir cola ordenada
  cola.vaciar()
  PARA k ← 0 HASTA contador-1 HACER
    cola.encolar(arregloTemp[k])
  FIN PARA
FIN
```

La optimización del algoritmo de priorización es clave. Para colas muy grandes, se consideraría un Heap Binario, reduciendo la complejidad de las operaciones a O(log n).

Diagramas de flujos

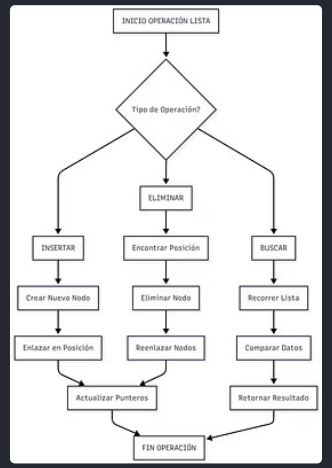


DIAGRAMA DE FLUJO - LISTAS ENLAZADAS

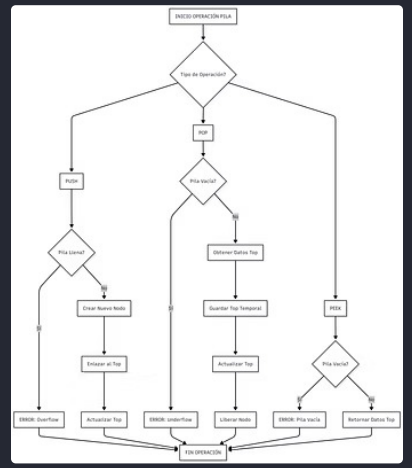


DIAGRAMA DE FLUJO - PILAS

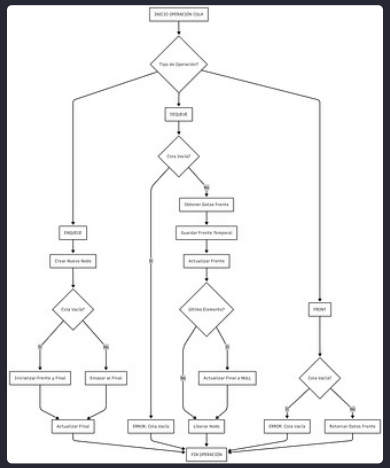
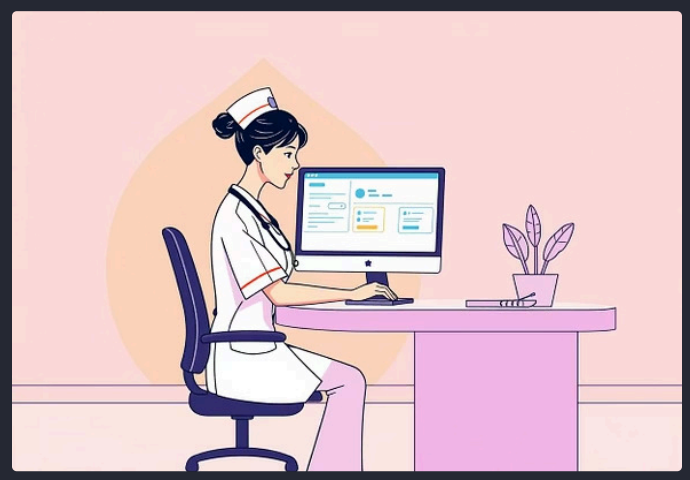


DIAGRAMA DE FLUJO - COLAS

🔄 Visualización del Flujo: Del Registro a la Atención

La correcta implementación de los algoritmos se traduce en flujos de trabajo claros y automatizados, minimizando los puntos de error y maximizando la velocidad de respuesta.



Flujo de Atención (Extracción de la Cola)

- El proceso de atención comienza con la verificación de la **Cola de Espera**.
- Si la cola no está vacía, se ejecuta la operación **Desencolar** (O(1)).
- El estado del paciente es actualizado a "Atendido" en la Lista Principal.
- Finalmente, un registro de la atención se **Apila** en el Historial Clínico para su consulta futura.

👥 Planificación y Equipo: Un Proyecto de Impacto Académico

Este proyecto es un esfuerzo colaborativo que aplica los conocimientos de estructuras de datos y algoritmia para construir una solución tangible en el ámbito de la salud.

Liderazgo y Arquitectura

David Titto Cruz: Definición de la arquitectura del sistema y gestión integral del proyecto.

Desarrollo Backend

Jurgen Molero Rivera: Implementación y optimización de las estructuras de datos (Lista, Cola, Pila) y sus algoritmos.

Interfaz y Documentación

Sebastian Serrano Mendoza: Diseño de la Interfaz de Usuario, validaciones de entrada y elaboración del informe final.

Cronograma y Metodología

Adoptamos una metodología de **Desarrollo Iterativo** con entregas semanales claras para asegurar el avance continuo y la calidad del código.

Fase	Duración	Entregables Clave
Análisis y Diseño	17-23 Oct	Especificaciones funcionales y Diseño de Estructuras de Datos
Implementación	24-28 Oct	Módulos de Lista, Cola y Pila funcionales y testeados
Integración	29-31 Oct	Sistema integrado y pruebas de extremo a extremo
Documentación	01 Nov	Informe final y Presentación ejecutiva