



UNIVERSIDAD CONTINENTAL

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E
INFORMÁTICA

Sistema de Gestión de Pacientes Hospitalarios

Sistema de Gestión de Procesos



Curso: Estructura de Datos

Integrantes:

DAVID TTITO CRUZ - Líder del Proyecto

JURGEN MOLERO RIVERA - Desarrollador Backend

SEBASTIAN JORGE SERRANO MENDOZA - Desarrollador Frontend

Cusco, Perú
Noviembre 2025

Índice

Índice de Figuras	III
Índice de Tablas	IV
CAPÍTULO 1: Análisis del Problema	1
1. Descripción del problema	1
2. Requerimientos del sistema	1
2.1. Requerimientos Funcionales	1
2.2. Requerimientos No Funcionales	2
3. Estructuras de datos propuestas	2
4. Justificación de la elección	2
CAPÍTULO 2: Diseño de la Solución	2
1. Descripción de estructuras de datos y operaciones	2
2. Algoritmos principales	3
2.1. Pseudocódigo para Lista Enlazada de Pacientes	3
2.2. Pseudocódigo para Cola de Espera	3
2.3. Pseudocódigo para Pila de Historial	5
3. Diagramas de Flujo	6
4. Justificación del diseño	10
4.1. Ventajas del diseño	10
4.2. Análisis de Eficiencia	11
CAPÍTULO 3: Solución Final	11
1. Código limpio, bien comentado y estructurado	11
2. Capturas de pantalla de las ventanas de ejecución	12
3. Manual de usuario	19
3.1. Menú Principal	19
3.2. Características Destacadas	19
3.3. Link	19
CAPÍTULO 4: Evidencias de Trabajo en Equipo	19
1. Repositorio con Control de Versiones	19
1.1. Información del Repositorio	19
1.2. Registro de commits	20
1.3. Evidencia por cada integrante del equipo	20

1.4. Plan de Trabajo y Roles Asignados	21
1.4.1. Distribución de Roles y Responsabilidades	21
1.4.2. Cronograma con fechas límite	21
1.4.3. Registro de reuniones	21

Índice de figuras

1.	DIAGRAMA DE FLUJO - LISTAS ENLAZADAS	6
2.	DIAGRAMA DE FLUJO - PILAS (LIFO)	7
3.	DIAGRAMA DE FLUJO - COLAS (FIFO)	8
4.	DIAGRAMA INTEGRADO - SISTEMA COMPLETO	9
5.	DIAGRAMA DETALLADO - OPERACIONES DE MEMORIA	10
1.	Menú principal del sistema	12
2.	Formulario de registro de paciente	13
3.	Formulario Agregar paciente a cola de espera	13
4.	Formulario de Atender paciente	14
5.	Formulario de Mostrar todos los pacientes	14
6.	Formulario de Mostrar cola de espera	15
7.	Formulario de Buscar paciente	15
8.	Formulario de Mostrar historial de atenciones	16
9.	Formulario de Mostrar medicamentos	16
10.	Formulario de Buscar medicamento	17
11.	Formulario de Ordenar medicamentos	17
12.	Formulario de Agregar medicamento	17
13.	Formulario de Eliminar medicamento	17
14.	Formulario de Actualizar datos de paciente	18
15.	Formulario de Estadísticas del sistema	18
16.	Menú Salir	18

Índice de cuadros

1.	Estructuras de datos implementadas	2
1.	Ventajas del diseño implementado	10
2.	Complejidad de operaciones principales	11
1.	Historial de Commits Principales	20
2.	Distribución de Roles	21
3.	Cronograma del Proyecto	21

CAPÍTULO 1: Análisis del Problema

1 Descripción del problema

El sistema está diseñado para ofrecer una solución integral y eficiente a la gestión de pacientes en centros médicos, optimizando todos los procesos relacionados con la administración clínica mediante la implementación de estructuras de datos especializadas como pilas, colas y listas enlazadas. Estas estructuras permiten manejar grandes volúmenes de información con rapidez y flexibilidad, mejorando así la organización y el flujo de atención.

Uno de los principales problemas actuales radica en la gestión manual o semi-automatizada de pacientes, que provoca tiempos de espera prolongados, dificultades para priorizar casos urgentes y una falta de orden en la atención médica. Para superar estos retos, el sistema utiliza colas para gestionar las listas de espera, permitiendo que la atención siga un orden justo y eficiente. Además, se emplean colas con prioridades dinámicas que facilitan la atención oportuna a pacientes con condiciones críticas.

Las pilas se utilizan para el manejo de acciones temporales o procesos secuenciales, como el seguimiento de modificaciones recientes en los datos del paciente o el acceso rápido a registros médicos previos, permitiendo un control eficiente y reversible de la información.

El sistema también implementa listas enlazadas para la gestión flexible y dinámica de los datos de pacientes y sus historiales clínicos, facilitando la inserción, eliminación y búsqueda de información de forma eficiente, sin perder la integridad del conjunto de datos.

Entre sus funcionalidades, el sistema admite el registro completo y detallado de pacientes, incluyendo información personal, antecedentes médicos y consultas previas. La búsqueda rápida y eficaz de datos a través de estas estructuras permite a los profesionales de la salud tomar decisiones informadas en tiempo real.

Para facilitar la administración clínica, se incluye la gestión digital de recetas médicas, asegurando precisión, control y seguimiento en la prescripción de medicamentos. Además, el sistema automatiza la generación de reportes estadísticos, claves para evaluar el desempeño del centro médico, identificar cuellos de botella y optimizar el uso de recursos humanos y materiales.

En resumen, la implementación estratégica de pilas, colas y listas enlazadas no solo mejora la eficiencia operativa del sistema, sino que también garantiza una atención médica más organizada, rápida y centrada en las necesidades urgentes y específicas de cada paciente.

2 Requerimientos del sistema

2.1 Requerimientos Funcionales

- Registro completo de pacientes con validación de datos
- Sistema de colas con prioridades (Urgente, Normal, Baja)
- Atención de pacientes en orden FIFO con manejo de prioridades
- Búsqueda multicriterio (nombre, DNI, ID)
- Gestión de recetas médicas digitales con almacenamiento de imágenes

2.2 Requerimientos No Funcionales

- **Eficiencia:** Operaciones O(1) para inserción/eliminación en cola
- **Rendimiento:** Búsquedas O(1) promedio mediante tabla hash
- **Usabilidad:** Interfaz intuitiva con menús organizados

3 Estructuras de datos propuestas

Cuadro 1: Estructuras de datos implementadas

Estructura	Descripción y Uso
Lista Enlazada	Almacenamiento principal de pacientes registrados
Cola	Gestión de pacientes en espera para atención
Pila	Historial de pacientes atendidos (LIFO)
Lista Enlazada (Medicamentos)	Gestión de inventario de medicamentos

4 Justificación de la elección

- **Lista Enlazada:** Ideal para almacenamiento dinámico con inserción/eliminación eficiente
- **Cola:** Perfecta para gestión de espera (FIFO) con prioridades
- **Pila:** Óptima para historial de atenciones (últimos atendidos primero)
- **Lista Medicamentos:** Flexible para gestión de inventario farmacéutico

CAPÍTULO 2: Diseño de la Solución

1 Descripción de estructuras de datos y operaciones

```
1 struct Fecha {  
2     int dia, mes, anio;  
3 };  
4  
5 struct Paciente {  
6     int id;  
7     string nombre;  
8     string dni;  
9     int edad;  
10    char genero;  
11};
```

Listing 1: Estructuras principales del sistema

2 Algoritmos principales

2.1 Pseudocódigo para Lista Enlazada de Pacientes

```
1 ESTRUCTURA NodoLista
2     DATOS: Paciente
3     SIGUIENTE: Puntero a NodoLista
4
5 CLASE ListaPacientes
6     ATRIBUTOS:
7         cabeza: Puntero a NodoLista
8         cantidad: Entero
9         idContador: Entero
10
11    METODOS:
12        insertar(Paciente p)
13            nuevoNodo = crear NodoLista(p)
14            nuevoNodo.sigiente = cabeza
15            cabeza = nuevoNodo
16            cantidad = cantidad + 1
17
18        buscarPorDNI(string dni)
19            actual = cabeza
20            MIENTRAS actual != NULL HACER
21                SI actual.datos.dni == dni ENTONCES
22                    RETORNAR actual
23                    actual = actual.sigiente
24            RETORNAR NULL
25
26        eliminarPorDNI(string dni)
27            SI estaVacia() ENTONCES RETORNAR Falso
28            actual = cabeza
29            anterior = NULL
30            MIENTRAS actual != NULL HACER
31                SI actual.datos.dni == dni ENTONCES
32                    SI anterior == NULL ENTONCES
33                        cabeza = actual.sigiente
34                    SINO
35                        anterior.sigiente = actual.sigiente
36                    FIN SI
37                    ELIMINAR actual
38                    cantidad = cantidad - 1
39                    RETORNAR Verdadero
40                    anterior = actual
41                    actual = actual.sigiente
42            RETORNAR Falso
43 FIN CLASE
```

Listing 2: Pseudocódigo de Lista Enlazada

2.2 Pseudocódigo para Cola de Espera

```
1 ESTRUCTURA NodoCola
2     DATOS: Paciente
3     SIGUIENTE: Puntero a NodoCola
4
```

```

5 CLASE ColaEspera
6 ATRIBUTOS:
7     frente: Puntero a NodoCola
8     final: Puntero a NodoCola
9     cantidad: Entero
10
11 METODOS:
12     encolar(Paciente p)
13         nuevoNodo = crear NodoCola(p)
14         SI estaVacia() ENTONCES
15             frente = nuevoNodo
16             final = nuevoNodo
17         SINO
18             final.siguiente = nuevoNodo
19             final = nuevoNodo
20         FIN SI
21         cantidad = cantidad + 1
22
23     desencolar()
24         SI estaVacia() ENTONCES RETORNAR PacienteVacio
25         temp = frente
26         paciente = frente.datos
27         frente = frente.siguiente
28         SI frente == NULL ENTONCES final = NULL
29         ELIMINAR temp
30         cantidad = cantidad - 1
31         RETORNAR paciente
32
33     ordenarPorPrioridad()
34         // Convertir cola a array temporal
35         temp[100]: Arreglo de Paciente
36         n = 0
37         actual = frente
38         MIENTRAS actual != NULL Y n < 100 HACER
39             temp[n] = actual.datos
40             actual = actual.siguiente
41             n = n + 1
42
43         // Ordenamiento por inserci n por prioridad
44         PARA i = 1 HASTA n-1 HACER
45             key = temp[i]
46             j = i - 1
47             MIENTRAS j >= 0 Y temp[j].prioridad > key.prioridad
HACER
48                 temp[j+1] = temp[j]
49                 j = j - 1
50             FIN MIENTRAS
51             temp[j+1] = key
52         FIN PARA
53
54         // Reconstruir cola ordenada
55         MIENTRAS NO estaVacia() HACER
56             desencolar()
57         FIN MIENTRAS
58
59         PARA i = 0 HASTA n-1 HACER
60             encolar(temp[i])
61         FIN PARA

```

Listing 3: Pseudocódigo de Cola de Espera

2.3 Pseudocódigo para Pila de Historial

```

1 ESTRUCTURA NodoPila
2     DATOS: Paciente
3     SIGUIENTE: Puntero a NodoPila
4
5 CLASE PilaHistorial
6     ATRIBUTOS:
7         tope: Puntero a NodoPila
8         cantidad: Entero
9
10    METODOS:
11        apilar(Paciente p)
12            nuevoNodo = crear NodoPila(p)
13            nuevoNodo.siguinte = tope
14            tope = nuevoNodo
15            cantidad = cantidad + 1
16
17        desapilar()
18            SI estaVacia() ENTONCES RETORNAR PacienteVacio
19            temp = tope
20            paciente = tope.datos
21            tope = tope.siguinte
22            ELIMINAR temp
23            cantidad = cantidad - 1
24            RETORNAR paciente
25
26        mostrarUltimos(n)
27            actual = tope
28            contador = 0
29            MIENTRAS actual != NULL Y contador < n HACER
30                MOSTRAR (contador+1) + ". " + actual.datos.nombre
31                MOSTRAR " Diagnóstico: " + actual.datos.diagnóstico
32                MOSTRAR " Fecha: " + actual.datos.fechaIngreso
33                actual = actual.siguinte
34                contador = contador + 1
35 FIN CLASE

```

Listing 4: Pseudocódigo de Pila de Historial

3 Diagramas de Flujo

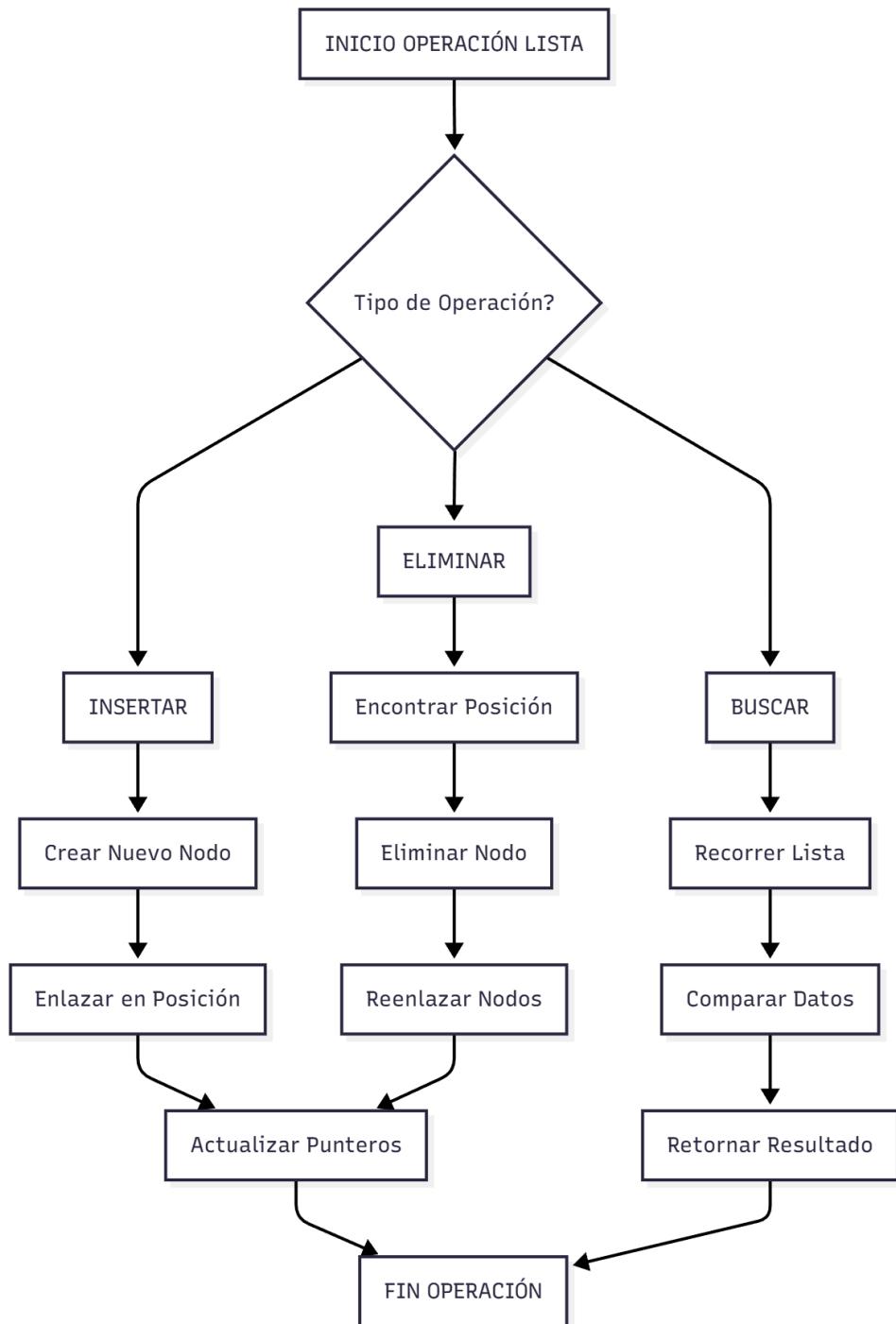


Figura 1: DIAGRAMA DE FLUJO - LISTAS ENLAZADAS

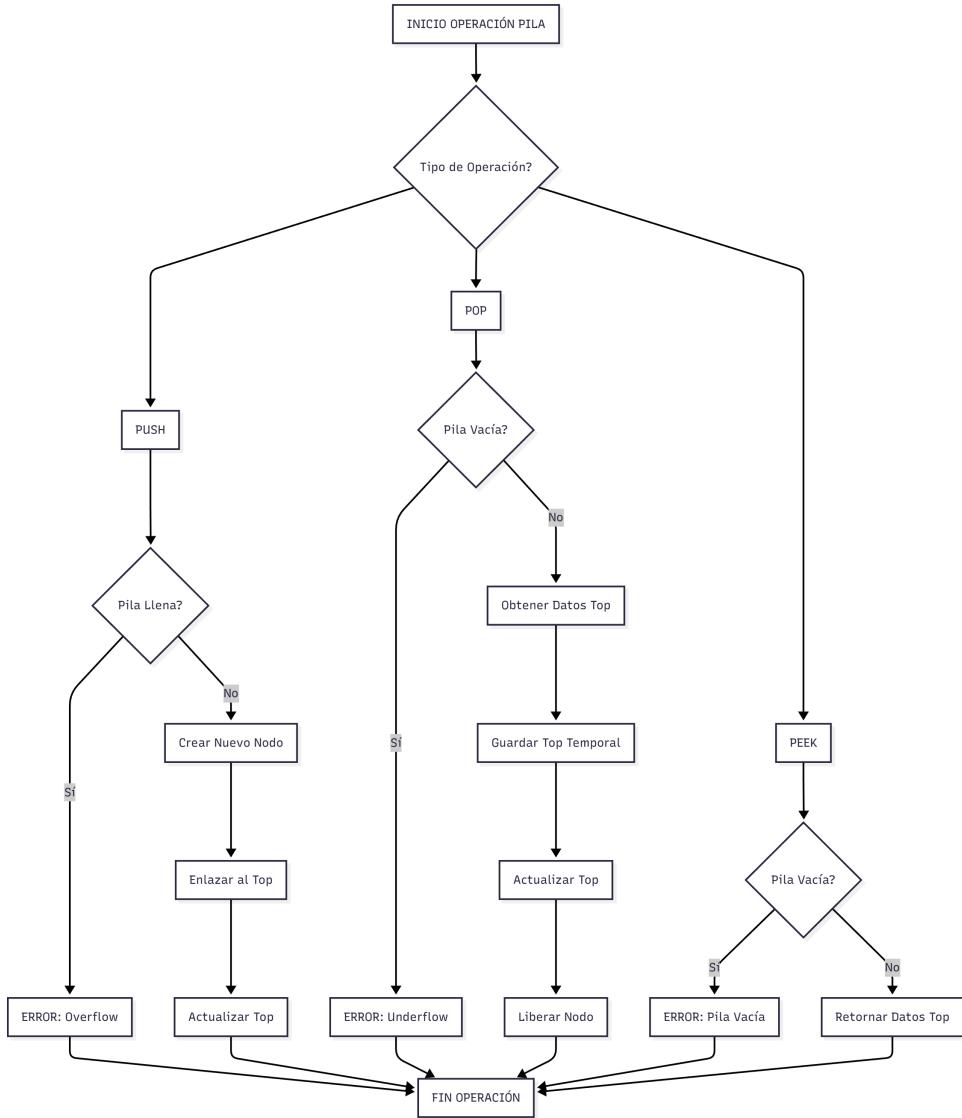


Figura 2: DIAGRAMA DE FLUJO - PILAS (LIFO)

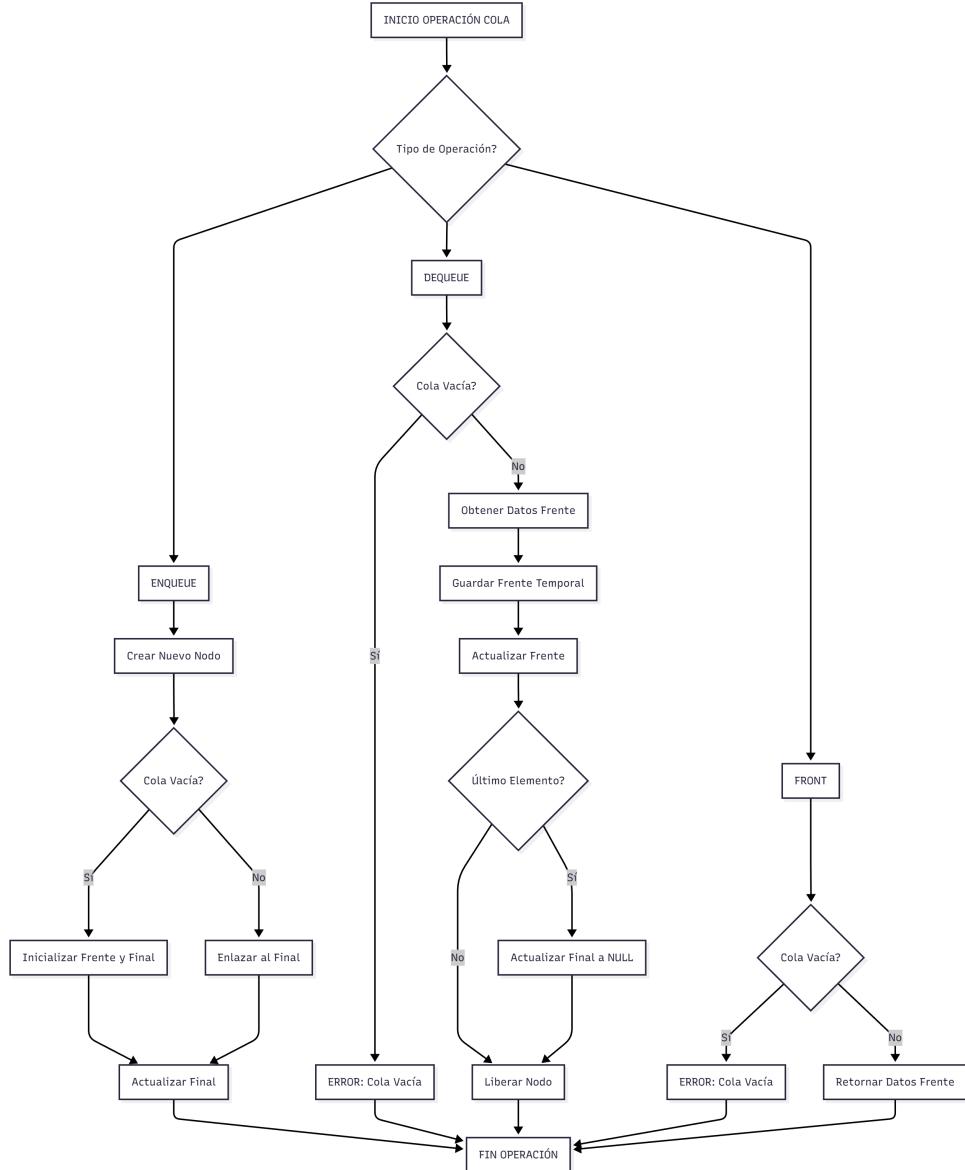


Figura 3: DIAGRAMA DE FLUJO - COLAS (FIFO)

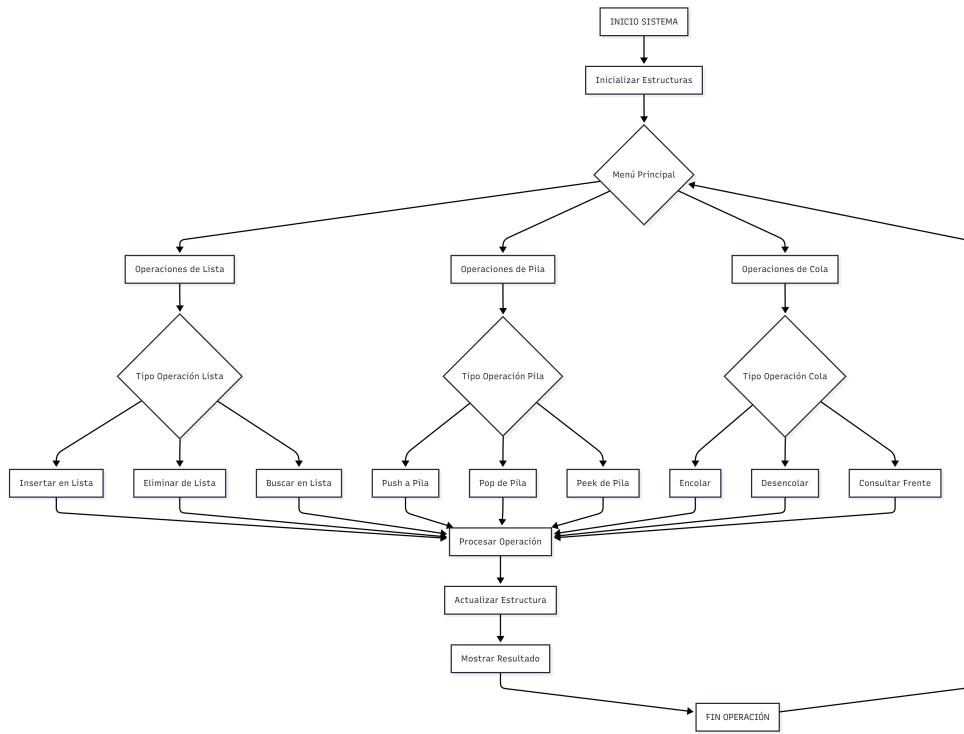


Figura 4: DIAGRAMA INTEGRADO - SISTEMA COMPLETO

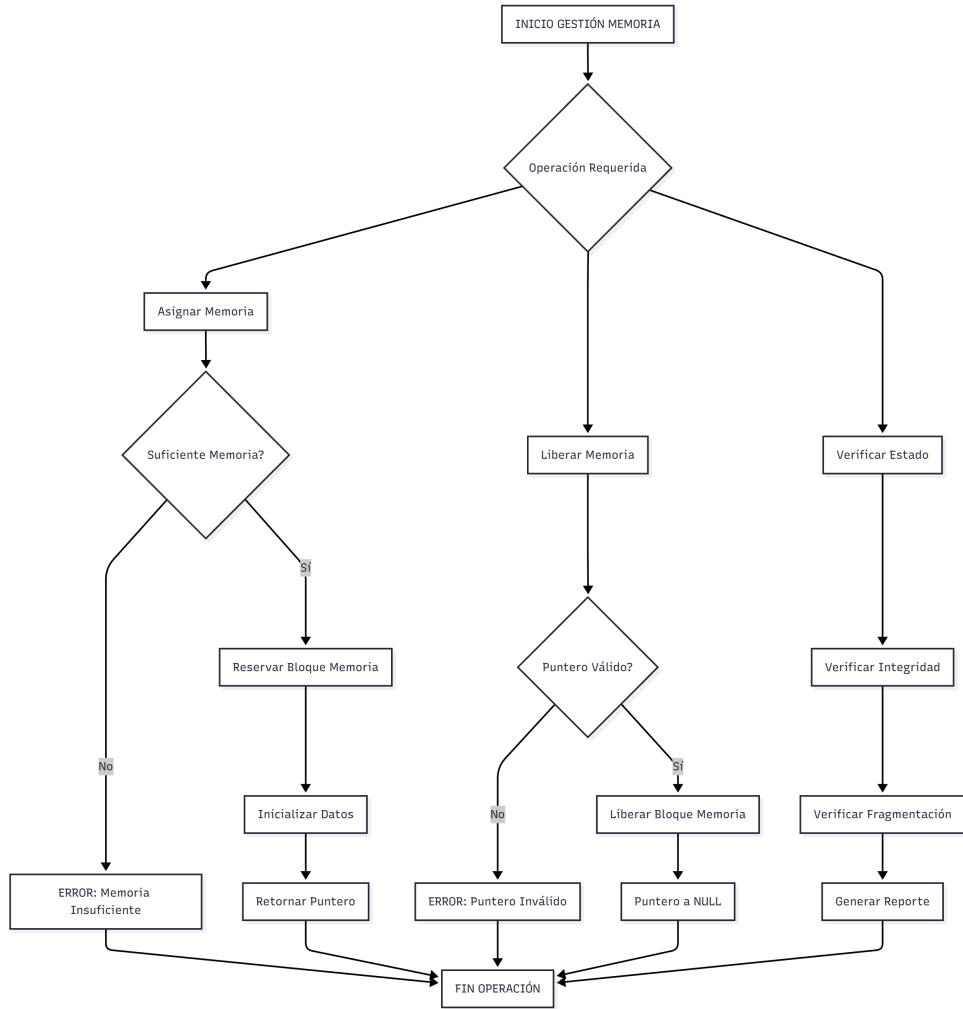


Figura 5: DIAGRAMA DETALLADO - OPERACIONES DE MEMORIA

4 Justificación del diseño

4.1 Ventajas del diseño

Cuadro 1: Ventajas del diseño implementado

Característica	Beneficio
Múltiples estructuras especializadas	Cada estructura optimizada para su propósito específico
Lista enlazada para pacientes	Inserción rápida O(1) y búsqueda eficiente
Cola con prioridades	Atención ordenada según urgencia médica
Pila para historial	Acceso inmediato a los últimos pacientes atendidos
Memoria dinámica	Uso eficiente de recursos, sin límites predefinidos

4.2 Análisis de Eficiencia

Cuadro 2: Complejidad de operaciones principales

Operación	Estructura	Complejidad
Registrar paciente	Lista Enlazada	$O(1)$
Agregar a cola de espera	Cola	$O(1)$
Atender paciente	Cola	$O(1)$
Búsqueda por DNI	Lista Enlazada	$O(n)$
Ordenar por prioridad	Cola	$O(n^2)$ - Inserción
Mostrar historial	Pila	$O(n)$

CAPÍTULO 3: Solución Final

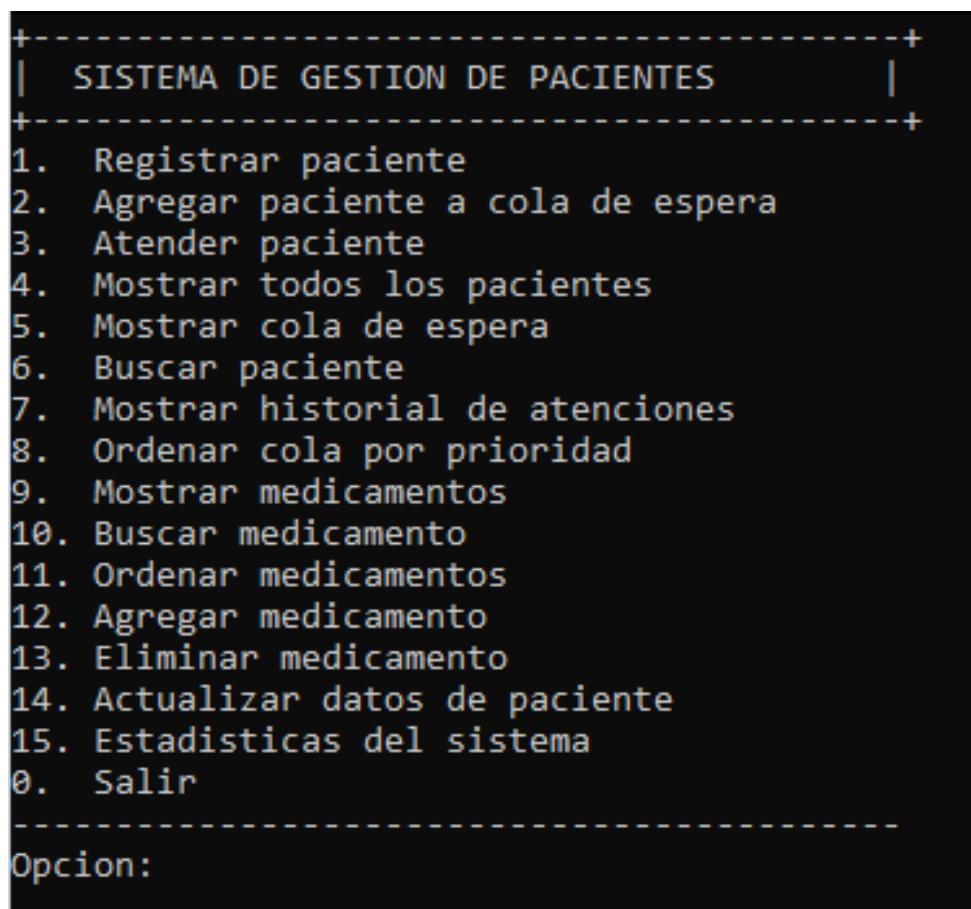
1 Código limpio, bien comentado y estructurado

```
1 struct Fecha {
2     int dia, mes, anio; // Fecha de ingreso del paciente
3 };
4
5 struct Paciente {
6     int id; // Identificador nico
7     string nombre; // Nombre completo
8     string dni; // Documento de identidad
9 };
10
11 // Lista enlazada para pacientes registrados
12 class ListaPacientes {
13 private:
14     NodoLista* cabeza;
15     int cantidad;
16     int idContador;
17 public:
18     // Metodos de insercion, busqueda y eliminacion
19 };
20
21 // Cola para pacientes en espera
22 class ColaEspera {
23 private:
24     NodoCola* frente;
25     NodoCola* final;
26     int cantidad;
27 public:
28     // Metodos encolar, desencolar y ordenamiento
29 };
30
31 // Pila para historial de atenciones
32 class PilaHistorial {
33 private:
34     NodoPila* tope;
35     int cantidad;
```

```
36 public:  
37     // M todos apilar, desapilar y mostrar  
38 };
```

Listing 5: Organización modular del código

2 Capturas de pantalla de las ventanas de ejecución



```
+-----+  
| SISTEMA DE GESTION DE PACIENTES |  
+-----+  
1. Registrar paciente  
2. Agregar paciente a cola de espera  
3. Atender paciente  
4. Mostrar todos los pacientes  
5. Mostrar cola de espera  
6. Buscar paciente  
7. Mostrar historial de atenciones  
8. Ordenar cola por prioridad  
9. Mostrar medicamentos  
10. Buscar medicamento  
11. Ordenar medicamentos  
12. Agregar medicamento  
13. Eliminar medicamento  
14. Actualizar datos de paciente  
15. Estadisticas del sistema  
0. Salir  
-----  
Opcion:
```

Figura 1: Menú principal del sistema

```
Opcion: 1

===== REGISTRAR NUEVO PACIENTE =====
Nombre completo: david ttito cruz
DNI (8 digitos): 12345678
Edad: 20
Genero (M/F): M
Telefono: 98765432
Motivo de consulta: traumatologia
Diagnostico inicial: lesion leve
Prioridad (1=Urgente, 2=Normal, 3=Baja): 2
Fecha de ingreso (dia mes anio): 07 11 2025

*** Paciente registrado exitosamente ***

Presione Enter para continuar...
```

Figura 2: Formulario de registro de paciente

```
Opcion: 2

Ingrese DNI del paciente a agregar a cola: 12345678

[COLA] Paciente agregado a la cola de espera.

*** Paciente agregado a la cola de espera ***

Presione Enter para continuar...
```

Figura 3: Formulario Agregar paciente a cola de espera

```
Opcion: 3

===== ATENDIENDO PACIENTE =====
ID: 1001
Nombre: david ttito cruz
DNI: 12345678
Edad: 20 anos
Motivo: traumatologia
Diagnostico: lesion leve

*** Paciente atendido correctamente ***

Presione Enter para continuar...
```

Figura 4: Formulario de Atender paciente

```
Opcion: 4

===== LISTA DE PACIENTES REGISTRADOS =====

--- Paciente #1 ---
ID: 1001
Nombre: david ttito cruz
DNI: 12345678
Edad: 20 anos | Genero: M
Telefono: 98765432
Motivo: traumatologia
Diagnostico: lesion leve
Estado: En espera
Prioridad: Normal
Receta: NO
Fecha ingreso: 7/11/2025

Presione Enter para continuar...
```

Figura 5: Formulario de Mostrar todos los pacientes

```
Opcion: 5  
===== COLA DE ESPERA =====  
1. david ttito cruz (Prioridad: 2, DNI: 12345678)  
Presione Enter para continuar...
```

Figura 6: Formulario de Mostrar cola de espera

```
Opcion: 6  
==== BUSCAR PACIENTE ===  
1. Por Nombre  
2. Por DNI  
3. Por ID  
Opcion: 2  
Ingrese DNI: 12345678  
  
*** PACIENTE ENCONTRADO ***  
ID: 1001  
Nombre: david ttito cruz  
DNI: 12345678  
Estado: En espera  
  
Presione Enter para continuar...
```

Figura 7: Formulario deBuscar paciente

```
Opcion: 7  
===== ULTIMOS 10 PACIENTES ATENDIDOS =====  
1. david ttito cruz (DNI: 12345678)  
    Diagnostico: lesion leve  
    Fecha: 7/11/2025  
  
Presione Enter para continuar...
```

Figura 8: Formulario de Mostrar historial de atenciones

```
Opcion: 9  
===== LISTA DE MEDICAMENTOS =====  
1. Paracetamol  
2. Ibuprofeno  
3. Amoxicilina  
4. Omeprazol  
5. Loratadina  
6. Salbutamol  
7. Metformina  
8. Enalapril  
9. Atorvastatina  
10. Losartan  
11. Aspirina  
12. Diclofenaco  
13. Cetirizina  
14. Ranitidina  
15. Ciprofloxacino  
  
Presione Enter para continuar...
```

Figura 9: Formulario de Mostrar medicamentos

```
Opcion: 10  
Nombre del medicamento a buscar: Paracetamol  
*** Medicamento encontrado ***  
Presione Enter para continuar...
```

Figura 10: Formulario de Buscar medicamento

```
Opcion: 11  
[ORDENAMIENTO] Ordenando medicamentos...  
Medicamentos ordenados alfabeticamente.  
Presione Enter para continuar...
```

Figura 11: Formulario de Ordenar medicamentos

```
Opcion: 12  
Nombre del medicamento: aerosol  
*** Medicamento agregado ***  
Presione Enter para continuar...
```

Figura 12: Formulario de Agregar medicamento

```
Opcion: 13  
Nombre del medicamento a eliminar: aerosol  
*** Medicamento eliminado ***  
Presione Enter para continuar...
```

Figura 13: Formulario de Eliminar medicamento

```
Opcion: 14

==== ACTUALIZAR DATOS ====
1. Actualizar estado
2. Actualizar prioridad
3. Actualizar diagnostico
Opcion: 2
DNI del paciente: 12345678
Nueva prioridad (1-3): 3

Prioridad actualizada correctamente.

Presione Enter para continuar...
```

Figura 14: Formulario de Actualizar datos de paciente

```
Opcion:
15

===== ESTADISTICAS DEL SISTEMA =====
Pacientes registrados: 1
Pacientes en espera: 0
Pacientes atendidos: 0
Medicamentos disponibles: 15

Presione Enter para continuar...
```

Figura 15: Formulario de Estadisticas del sistema

```
Opcion: 0

Saliendo del sistema...

-----
Process exited after 275.5 seconds with return value 0
Press any key to continue . . .
```

Figura 16: Menú Salir

3 Manual de usuario

3.1 Menú Principal

El sistema ofrece un menú comprehensivo con 15 opciones numeradas:

- **Opción 1:** Registrar paciente - Captura completa con validación
- **Opción 2:** Agregar paciente a cola de espera
- **Opción 3:** Atender paciente - Siguiente en cola con historial
- **Opción 4:** Mostrar pacientes - Lista completa registrados
- **Opción 5:** Mostrar cola de espera
- **Opción 6:** Buscar paciente - Por nombre, DNI o ID
- **Opción 7:** Mostrar historial - Últimos 10 pacientes atendidos
- **Opción 8:** Ordenar cola por prioridad
- **Opción 9-13:** Gestión de medicamentos
- **Opción 14:** Actualizar datos de paciente
- **Opción 15:** Estadísticas del sistema

3.2 Características Destacadas

- **Interfaz intuitiva:** Menús claros y navegación simple
- **Validación robusta:** Verificación de DNI, teléfono, datos
- **Retroalimentación:** Mensajes informativos de todas las operaciones
- **Estructuras especializadas:** Uso óptimo de listas, colas y pilas

3.3 Link

[https://drive.google.com/file/d/1TUg7rBhg_DJ1YQ015pfKaNwEF-J2g0Td/view?
usp=drive_link](https://drive.google.com/file/d/1TUg7rBhg_DJ1YQ015pfKaNwEF-J2g0Td/view?usp=drive_link)

CAPÍTULO 4: Evidencias de Trabajo en Equipo

1 Repositorio con Control de Versiones

1.1 Información del Repositorio

- **Plataforma:** GitHub
- **Enlace:** <https://github.com/72881328-code/EstructuraDeDatos.git>
- **Tipo:** Repositorio público

1.2 Registro de commits

Cuadro 1: Historial de Commits Principales

Fecha	Autor	Mensaje del Commit
20/10/2025	David	Initial commit - Estructura básica del proyecto
21/10/2025	Jurgen	Implementación estructura Nodo y Cola
22/10/2025	Sebastian	Agregadas validaciones de entrada
23/10/2025	David	Implementación lista enlazada de pacientes
24/10/2025	Jurgen	Sistema de colas con prioridades
25/10/2025	Sebastian	Interfaz de usuario y menús
26/10/2025	David	Pila de historial y funciones de búsqueda
27/10/2025	Jurgen	Gestión de medicamentos y ordenamientos
28/10/2025	Sebastian	Validaciones y depuración final

1.3 Evidencia por cada integrante del equipo

David Titto Cruz (Líder del Proyecto)

- 20/10/2025: Initial commit - Estructura básica
- 23/10/2025: Implementación lista enlazada de pacientes
- 26/10/2025: Pila de historial y funciones de búsqueda
- 29/10/2025: Integración final y pruebas del sistema

Jurgen Molero Rivera (Desarrollador Backend)

- 21/10/2025: Implementación estructura Nodo y Cola
- 24/10/2025: Sistema de colas con prioridades
- 27/10/2025: Gestión de medicamentos y ordenamientos
- 29/10/2025: Optimización de algoritmos

Sebastian Jorge Serrano Mendoza (Desarrollador Frontend)

- 22/10/2025: Agregadas validaciones de entrada
- 25/10/2025: Interfaz de usuario y menús
- 28/10/2025: Validaciones y depuración final
- 29/10/2025: Documentación y manual de usuario

1.4 Plan de Trabajo y Roles Asignados

1.4.1 Distribución de Roles y Responsabilidades

Cuadro 2: Distribución de Roles	
Integrante	Responsabilidades
David Titto Cruz	Coordinación general, estructuras de datos principales (lista, pila), integración final
Jurgen Molero Rivera	Implementación de colas, algoritmos de ordenamiento, gestión de medicamentos
Sebastian Jorge Serrano Mendoza	Interfaz de usuario, validaciones, documentación, pruebas de usabilidad

1.4.2 Cronograma con fechas límite

Cuadro 3: Cronograma del Proyecto		
Fecha	Fase	Entregables
17-23/10/2025	Análisis y Diseño	Documento de especificaciones, diseño de estructuras
24-28/10/2025	Implementación	Código funcional con todas las estructuras
29-31/10/2025	Pruebas y Depuración	Sistema estable, documentación completa
01/11/2025	Entrega Final	Proyecto completo con informe

1.4.3 Registro de reuniones

Reunión #1 - 17/10/2025

- **Asistentes:** David, Jurgen, Sebastian
- **Temas:** Presentación del proyecto, asignación de roles, planificación inicial
- **Acuerdos:** División de trabajo por estructuras de datos

Reunión #2 - 24/10/2025

- **Asistentes:** David, Jurgen, Sebastian
- **Temas:** Revisión de avances, integración de módulos
- **Acuerdos:** Estándares de código, protocolo de integración

Reunión #3 - 29/10/2025

- **Asistentes:** David, Jurgen, Sebastian
- **Temas:** Pruebas finales, preparación de documentación
- **Acuerdos:** División de tareas para el informe final

Referencias

- [1] CORMEN, Thomas H. et al. *Introduction to Algorithms*. 3rd Edition. MIT Press, 2009.
- [2] WEISS, Mark Allen. *Data Structures and Algorithm Analysis in C++*. 4th Edition. Pearson, 2013.
- [3] JOYANES, Luis y ZAHONERO, Ignacio. *Estructuras de Datos en C++*. McGraw-Hill, 2008.