



UNIVERSIDAD
NACIONAL
DE COLOMBIA

PRESENTADO POR:
TORRES JULIAN

FECHA
2025-10-20

Hospital Urgency System



```
=====
                        HOSPITAL WAITING SYSTEM
=====
1. Register New Patient
2. Attend Next Patient
3. View Waiting Queue
4. View Attended Patients
5. Undo Last Attendance
6. Exit
-----
Enter your choice (1-6): █
```

Problema y solución

PROBLEMA

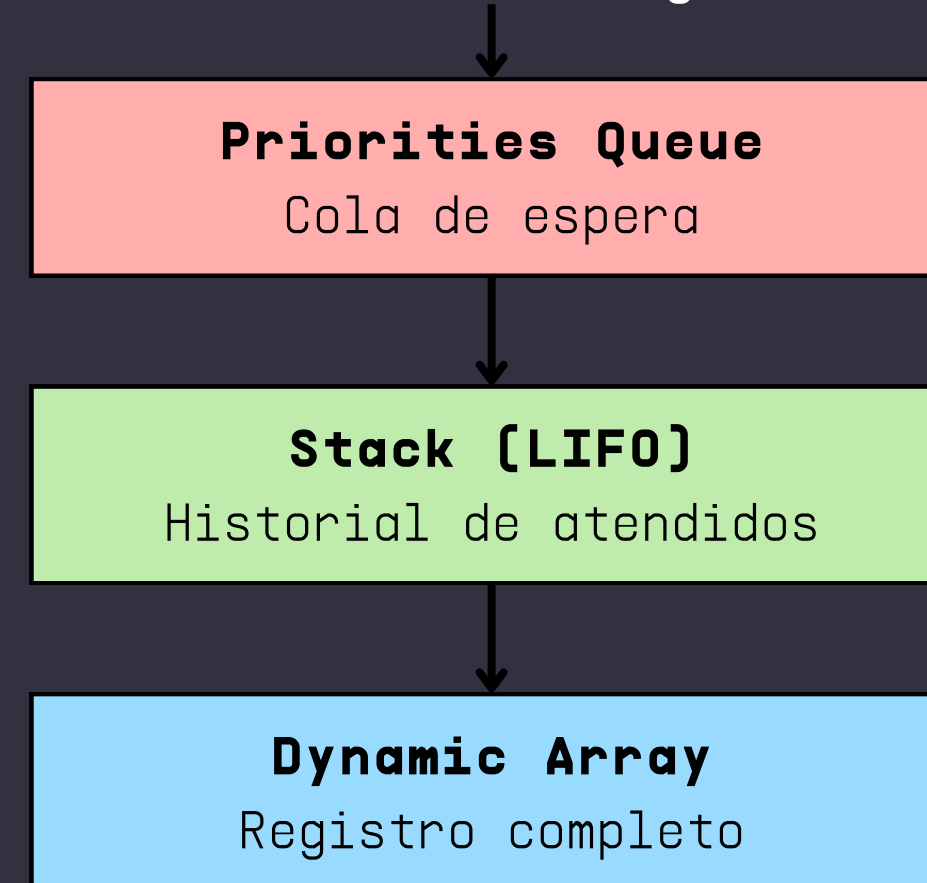
- Hospital atiente a aprox 100 pacientes
- Las llegadas son impredecibles
- Existen niveles dentro de las urgencias



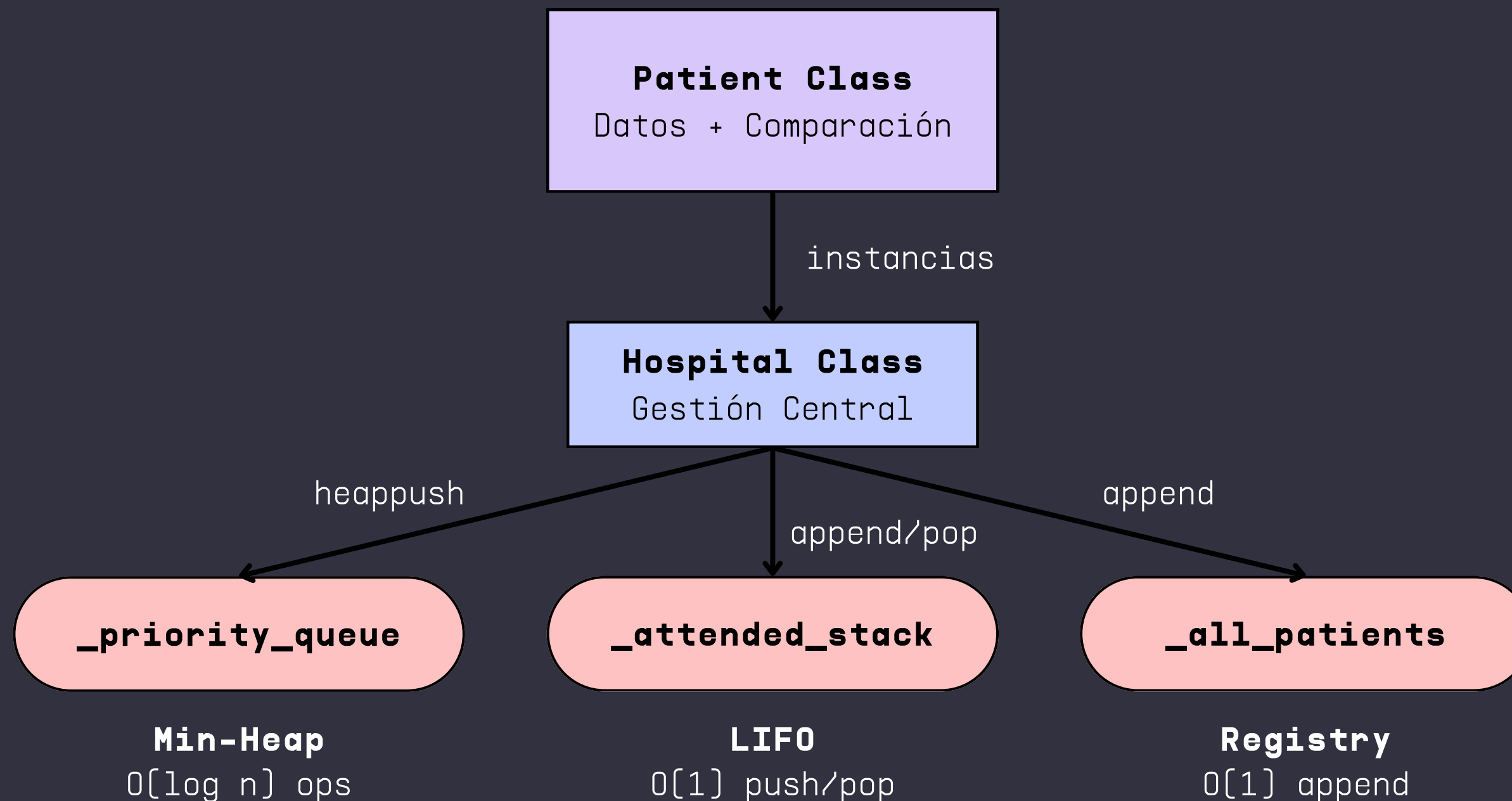
SOLUCION

Sistema de triaje

Paciente llega

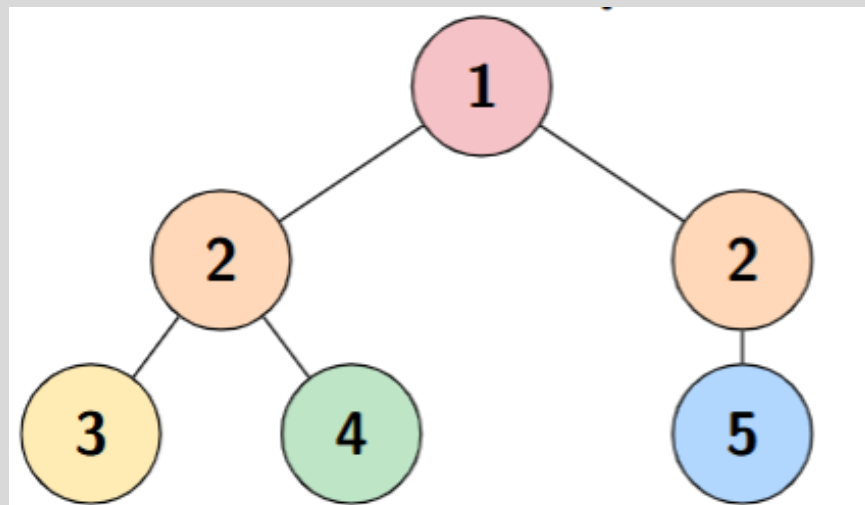


Arquitectura del Sistema



Priority Queue: Binary Min-Heap

ESTRUCTURA DEL HEAP



Invariante: $\text{parent} \leq \text{children}$

Representación: Array contiguo

¿Por qué $O(\log n)$?

Altura del árbol = $\lceil \log_2(n) \rceil$

Operaciones \leq altura

$= O(\log n)$

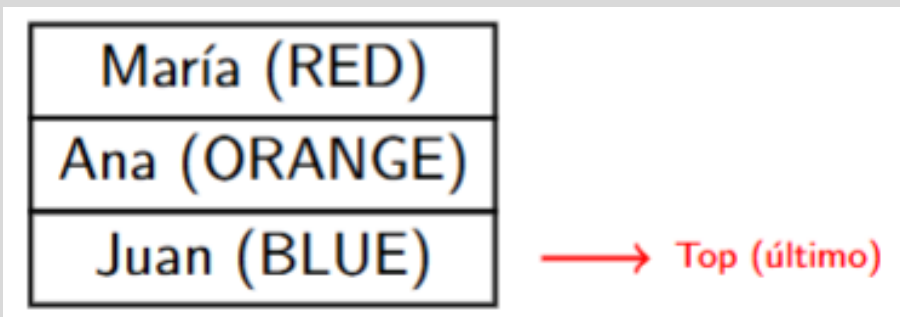
OPERACIONES CLAVE

```
1  # Inserción -  $O(\log n)$ 
2  heapq.heappush(queue, patient)
3  # 1. Add to end
4  # 2. Bubble up ( compare with parent )
5  # 3. Stop when  $\text{parent} \leq \text{current}$ 
6
7  # Extracción -  $O(\log n)$ 
8  patient = heapq.heappop(queue)
9  # 1. Remove root ( minimum )
10 # 2. Move last to root
11 # 3. Bubble down ( compare with children )
```

Stack and Dynamic Array

STACK:

Historial de atendidos



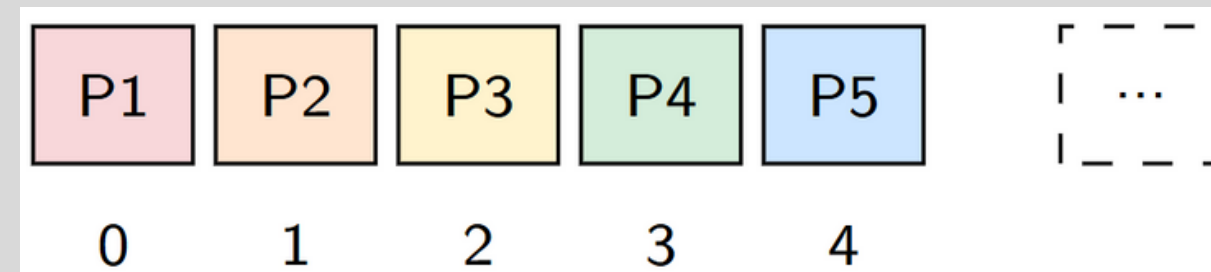
Push: $O(1)$

Pop: $O(1)$

Peek: $O(1)$

DYNAMIC ARRAY:

Registro total



Append: $O(1)$ amortizado

Access: $O(1)$

Search: $O(n)$

Crecimiento Geométrico

Nueva capacidad = actual * 1,25

Permite $O(1)$ amortizado en append.

Método de Elementos Finitos

COMPARACIÓN DE ALTERNATIVAS

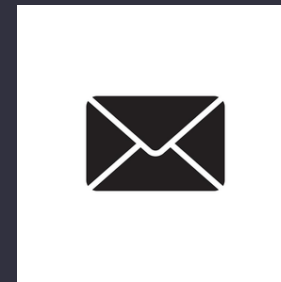
Operación	Heap	Array Ordenado	Array Desordenado
Insertar Paciente	$O(\log n)$	$O(n)$	$O(1)$
Atender Próximo	$O(\log n)$	$O(1)$	$O(n)$
Ver Mínimo	$O(1)$	$O(1)$	$O(n)$
Caso Mixto*	$O(\log n)$	$O(n)$	$O(n)$
*Workload típico: inserciones y extracciones intercaladas			

FLUJO DE OPERACIÓN

Demostración del sistema

1. Registrar: Mara [RED]
2. Registrar: Juan [BLUE]
3. Registrar: Ana [ORAN GE]
4. Ver cola de espera
5. Atender [RED] Mara
6. Ver cola de espera

Muchas gracias



Correos:

jutorresz@una1.edu.co



Repositorio:

https://github.com/JulianT/Proj1_Urgency-System