

Atención de pacientes en orden:

En una aplicación de envíos y mensajería, el personal de atención al cliente necesita llevar un registro rápido del estado de cada paquete (por ejemplo: “En camino”, “Entregado”, “Retrasado”) usando su número de seguimiento. La tabla hash debe lidiar con colisiones cuando dos IDs diferentes caigan en la misma cubeta.

Historia de usuario Nro.	1	Título:	Atención de pacientes por orden de llegada
Descripción	COMO:	Administrador del sistema de turnos del hospital	
	QUIERO:	Gestionar la atención de pacientes en el orden en que llegan.	
	PARA:	Garantizar justicia y orden en la atención médica.	
Criterios de aceptación	<ul style="list-style-type: none">▪ Se puede agregar un paciente (ENQUEUE <nombre>).▪ Se puede atender al paciente más antiguo (DEQUEUE).▪ Si no hay pacientes para atender, se debe imprimir "COLA VACIA".▪ Cada DEQUEUE debe imprimir el nombre del paciente atendido.		

Entrada	Varias líneas con comandos: <ul style="list-style-type: none">▪ ENQUEUE <nombre>▪ DEQUEUE▪ END (para finalizar)
Salida	Por cada DEQUEUE, imprimir el nombre del paciente, o "COLA VACIA" si la cola está vacía. No imprimir nada por ENQUEUE.

Instrucciones para la calificación automática

Antes de enviar la solución del reto, por favor tenga en cuenta los siguientes aspectos:

- La clase principal debe llamarse RetoCola.
- El método que procesa debe llamarse run.
- Solo debe imprimirse salida por los comandos DEQUEUE.
- Se leerá desde la entrada estándar y se escribirá a la salida estándar.
- Únicamente debe imprimir los resultados finales. No imprima valores con cálculos intermedios

Casos de prueba (visibles)

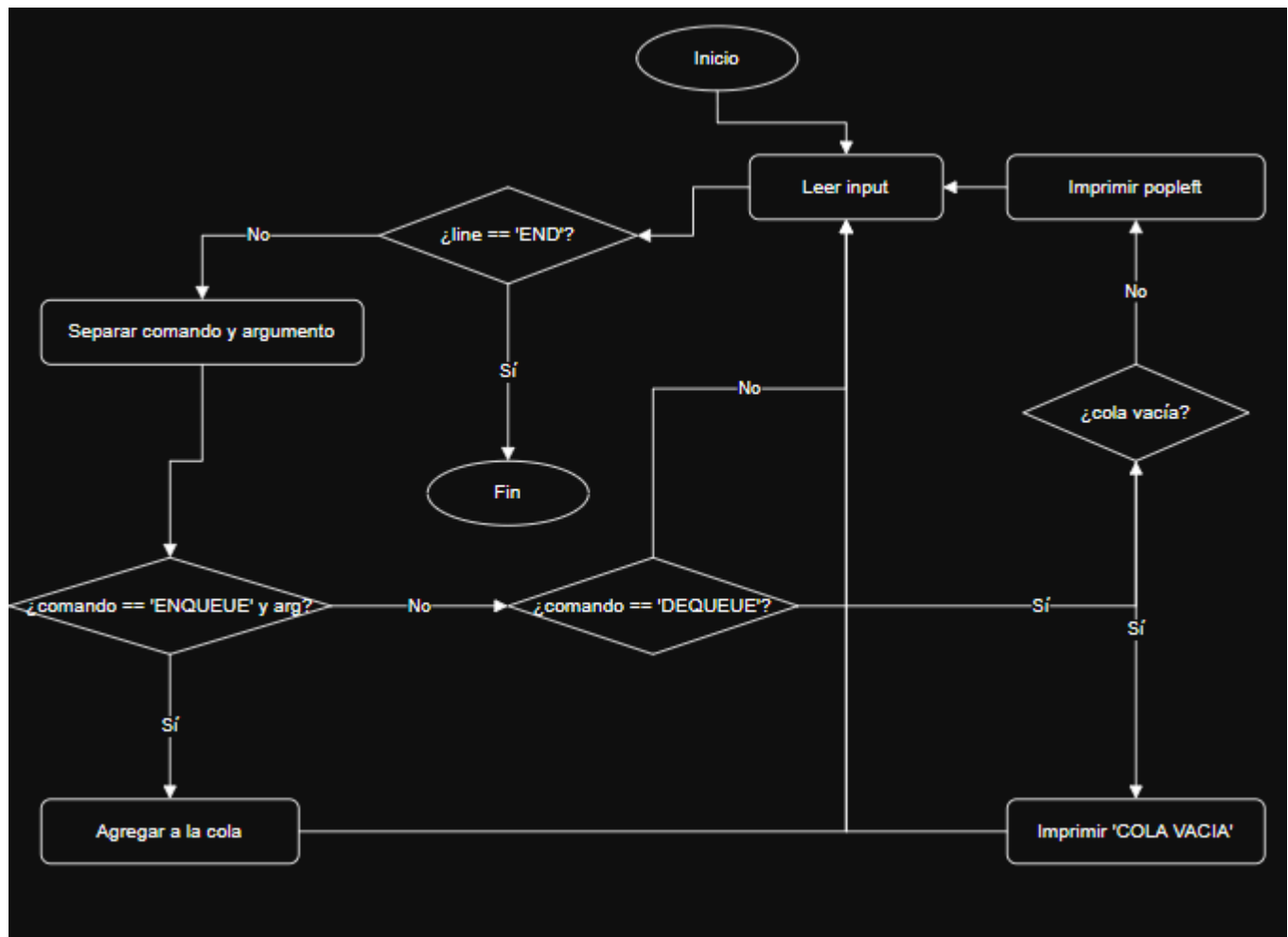
Entradas de ejemplo	ENQUEUE Juan ENQUEUE María
---------------------	-------------------------------

	DEQUEUE DEQUEUE DEQUEUE END
Salida de ejemplo	Juan María COLA VACIA

Casos de prueba (ocultos)

Caso prueba	Entrada	Salida esperada
1	DEQUEUE ENQUEUE Laura DEQUEUE END	COLA VACIA Laura
2	ENQUEUE Ana ENQUEUE Luis ENQUEUE Tomás DEQUEUE DEQUEUE END	Ana Luis

Diagrama de flujo:



Código Python :

```

from collections import deque

class RetoCola:

    def run(self):

        queue = deque()

        while True:

            try:

                line = input()

                if line.strip() == "END":

```

```
        break

    parts = line.strip().split(maxsplit=1)

    command = parts[0]

    if command == "ENQUEUE" and len(parts) == 2:

        queue.append(parts[1])

    elif command == "DEQUEUE":

        if queue:

            print(queue.popleft())

        else:

            print("COLA VACIA")

    except Exception:

        continue

if __name__ == "__main__":

    RetoCola().run()
```

```
        continue
    if __name__ == "__main__":
        RetoCola().run()
```

```
... ENQUEUE Juan
    ENQUEUE María
    DEQUEUE
    Juan
    DEQUEUE
    María
    DEQUEUE
    COLA VACIA
```

```
RetoCola().run()
```

```
⇒ DEQUEUE
   COLA VACIA
   ENQUEUE Laura
   DEQUEUE
   Laura
   END
```

```
RetoCola().run()
```

```
⇒ ENQUEUE Ana
   ENQUEUE Luis
   ENQUEUE Tomás
   DEQUEUE
   Ana
   DEQUEUE
   Luis
   END
```