

Práctica de Colas de Prioridad

Simulación de un Call Center con Cola de Prioridad

Descripción General

Los estudiantes deben implementar un **sistema de atención de llamadas** usando una **cola de prioridad**, donde los mensajes de los clientes son analizados para determinar su **prioridad de atención**.

Los **agentes** tienen diferentes niveles de experiencia y responderán los casos en la medida en que se desocupen.

El sistema funcionará de manera **continua a demanda**, leyendo nuevos mensajes desde una carpeta y encolándolos con una **prioridad dinámica basada en su contenido**.

Requisitos del Sistema

1 Procesamiento de mensajes de clientes

- Se leerán **mensajes aleatorios** desde archivos de una carpeta.
 - Cada mensaje tendrá un **nivel de prioridad** basado en su contenido.
 - La prioridad será un **valor continuo**, calculado con una heurística basada en la presencia de palabras clave.
-

2 Cálculo de Prioridad del Mensaje

- Se implementará una **heurística personalizada** que evalúe la urgencia de cada mensaje.
- Se usará un **diccionario de palabras clave**, donde cada término tiene un peso específico.
- La prioridad se calculará sumando los valores de las palabras encontradas en el mensaje.

Ejemplo de diccionario de palabras clave:

```
palabras_clave = {  
    "emergencia": 10, "urgente": 8, "fallo crítico": 9,
```

```
"problema": 5, "consulta": 2, "duda": 1
}
```

- Se debe recorrer el contenido del mensaje y **sumar los valores encontrados** para determinar la prioridad.
 - Este **valor continuo** determinará la posición del mensaje en la cola de prioridad.
-

3 Gestión de Agentes de Atención

Los agentes son los encargados de atender los mensajes en la medida en que se desocupen.

Cada agente tendrá características propias que afectarán el tiempo que tarda en atender un caso.

♦ Creación de Objetos de Tipo Agente

Cada agente debe ser representado como un **objeto con atributos específicos**, incluyendo:

- **id**: Identificador único del agente.
- **nivel_experiencia**: Puede ser **básico**, **intermedio** o **experto**.
- **estado**: Puede estar **ocupado** o **disponible**.
- **tiempo_de_respuesta**: Calculado con base en su experiencia y la dificultad del mensaje.

Ejemplo de niveles de experiencia y su impacto en el tiempo de respuesta:

Nivel	Reducción en tiempo de respuesta
Básico	Sin reducción (100%)
Intermedio	Reduce 25% del tiempo esperado
Experto	Reduce 50% del tiempo esperado

- Se debe calcular el tiempo estimado para cada agente basándose en:
 1. **Dificultad del caso** (longitud y palabras clave del mensaje).
 2. **Experiencia del agente** (afecta el tiempo final de resolución).

Ejemplo de fórmula de tiempo de atención:

$$\text{tiempo_estimado} = (\text{longitud_mensaje} / 10) + (\text{peso_palabras_clave} / 2)$$
$$\text{ajuste_por_experiencia} = \text{tiempo_estimado} * \text{factor_de_nivel}$$

Donde:

- `longitud_mensaje` es el número de palabras en el mensaje.
 - `peso_palabras_clave` es la suma de los valores del diccionario de palabras clave.
 - `factor_de_nivel` depende del nivel del agente (1.0 para básico, 0.75 para intermedio, 0.5 para experto).
-

4 Uso de Colas de Prioridad

- Se usará una **PriorityQueue** para organizar las llamadas.
 - Los mensajes con **mayor prioridad** serán atendidos primero.
 - Si hay **varios agentes disponibles**, el **más experimentado** tomará el siguiente caso.
-

5 Simulación de Atención de Agentes

♦ Opción 1 (Básica con `sleep`)

- Los agentes simplemente esperan `time.sleep(tiempo_estimado)` para simular el tiempo de atención.
- Una vez terminado, toman el siguiente caso en la cola.

♦ Opción 2 (Avanzada con Hilos)

- Cada agente funciona en **un hilo separado**, permitiendo que varios atiendan llamadas simultáneamente.
 - Se debe gestionar la sincronización para evitar que varios agentes intenten tomar el mismo caso.
 - ⚠ **Nota:** Quienes elijan esta opción estarán **eximidos** de la nota de seguimiento del bloque de Pilas y Colas.
-

6 Ciclo de Operación Continua

- Se seguirán **leyendo mensajes nuevos** de la carpeta en intervalos regulares.
 - Los mensajes serán **analizados, clasificados y encolados automáticamente**.
 - El sistema **no se detiene**, solo finaliza si el usuario lo indica.
-

⚠ **Restricción Importante**

❌ **No se permite el uso de librerías externas para la creación y gestión de la cola de prioridad.**

Los estudiantes deben **implementar su propia estructura de datos** para la cola de prioridad, asegurándose de manejar correctamente la inserción y extracción de elementos con base en su prioridad.

Pueden utilizar **listas enlazadas implementadas manualmente o cualquier otra estructura personalizada que respete el comportamiento de las colas de prioridad**, pero **no** se permite el uso de módulos como `queue.PriorityQueue` o `heapq`.

Entregables y Sustentación

- **Código fuente** del sistema implementado en Python con TypeHints.
- **Sustentación individual sobre la solución base y las modificaciones planteadas para el ejercicio de sustentación**, incluyendo:
 - **Explicación detallada** de la **heurística de prioridad** y la **heurística de tiempo de respuesta**.
- **Fecha de entrega y sustentación:** primera clase de semana 6

Grupo 63: 25 de marzo

Grupo 61: 27 de marzo

Grupo 64: 28 de marzo

Grupo 62: 28 de marzo