## Santiago Mesa

## 17/02/2025

## Estructuras de datos – taller TAD: propuesta de implementación de soluciones

**Revisar y complementar en caso de ser necesario los TAD's propuestos**

TAD asesor: Todo en general está bien, pero para “fijarAtendidos(n\_turnos)” Se podría modificar para agregar turnos individualmente en lugar de reemplazar toda la lista.

TAD usuario: Todo ok

TAD CAU: “atenderUsuario()” Solo dice que toma la cola más larga, pero no menciona qué pasa si hay varias colas con la misma longitud, podría ser al azar o siempre uno de los 3 temas.

Además, atender usuario hace 3 cosas que se podrían dividir, ver que cola es mas larga, buscar un asesor disponible y atender, es mejor dividirlo en “obtenerAsesorDisponible()” y “colaMasLarga()” así no repetimos código

**Proponer los pasos para la implementación de los métodos**

Diseño: obtenerAsesorDisponible(tema)

Objetivo: Encontrar un asesor disponible que pueda atender un tema específico.

Pasos:

1. Recorrer la lista de asesores del CAU.
2. Para cada asesor, verificar:
3. Si su estado es "disponible".
4. Si su tema de atención coincide con el tema buscado.
5. Si se encuentra un asesor disponible, devolverlo.
6. Si ningún asesor está disponible, devolver nullptr**.**

obtenerEstado() en Asesor nos dice el estado de un asesor en particular.

obtenerAsesorDisponible(tema) en CAU busca dentro de todos los asesores del CAU aquel que pueda atender el tema y esté disponible.

Diseño: colaMasLarga()

Objetivo: Determinar cuál de las tres colas (ventas, servicio técnico o reclamos) tiene más usuarios en espera.

Pasos:

1. Obtener los tamaños de las tres colas.
2. Compararlos y encontrar el mayor.
3. En caso de empate, decidir un criterio (en este caso, prioridad fija: VE > ST > RE).
4. Retornar el índice de la cola con más usuarios (0 para VE, 1 para ST, 2 para RE).

**Diseño e implementación de operaciones**

**asignarTurno(id, prioridad, tema)**

Objetivo: Asignar un turno a un usuario y encolarlo en la lista de espera correspondiente.

Pasos a seguir:

1. Determinar a qué cola pertenece el usuario según su tema:
   * "VE" → Cola de ventas (índice 0).
   * "ST" → Cola de servicio técnico (índice 1).
   * "RE" → Cola de reclamos (índice 2).
2. Asignar número de turno:
   * Obtener el número consecutivo actual de la cola correspondiente.
   * Asignarlo al usuario.
   * Incrementar el contador de turnos para la próxima asignación.
3. Crear el objeto Usuario con los datos proporcionados.
4. Encolar al usuario en la estructura adecuada:
   * Si el usuario tiene prioridad (prioridad == true), insertarlo al frente de la cola.
   * Si no, encolarlo normalmente al final.
5. Retornar el número de turno asignado como string para confirmación.

**Diseño: Asignar un turno**

De acuerdo con la descripción del TAD CAU, tenemos:

* colasAtencion: Un **vector de colas** que almacena a los usuarios en espera, una para cada tema.
* consecutivos: Un **vector de enteros** que lleva el conteo de los turnos asignados para cada tema.

Por lo tanto, necesitamos:

1. **Determinar a qué índice del vector corresponde el tema** (VE, ST, RE).
2. **Obtener y actualizar el turno correspondiente** del vector consecutivos.
3. **Crear un objeto Usuario** y encolarlo correctamente en colasAtencion

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

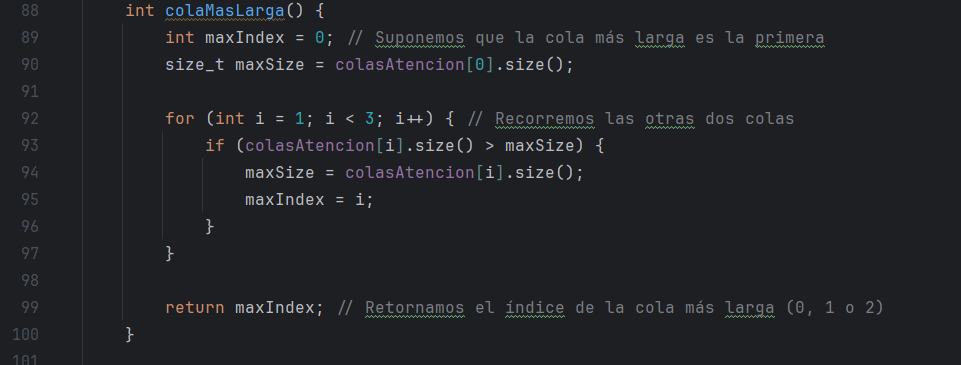
**Diseño: Atender un usuario**

**Objetivo:**  
Atender al usuario con mayor prioridad siguiendo estas reglas:

1. **Seleccionar la cola con más usuarios en espera**.
2. **Encontrar un asesor disponible** que pueda atender ese tema.
3. **Asignar el usuario al asesor**, retirándolo de la cola de espera.
4. **Registrar el turno del usuario en la lista de atendidos del asesor**.
5. **Retornar true si el usuario fue atendido exitosamente, o false si no fue posible**.

**Pasos detallados**

1. **Determinar cuál es la cola más larga**
   * Llamar al método colaMasLarga() para obtener el índice de la cola con más usuarios en espera.
   * Si todas las colas están vacías, retornar false.
2. **Verificar si hay un asesor disponible**
   * Llamar al método obtenerAsesorDisponible(tema) con el tema de la cola seleccionada.
   * Si no hay asesores disponibles, retornar false.
3. **Extraer al primer usuario de la cola seleccionada**
   * Obtener el usuario al frente de la cola.
   * Removerlo de la cola.
4. **Asignar el usuario al asesor**
   * Agregar el turno del usuario a la lista de turnos atendidos del asesor.
5. **Actualizar el estado del asesor**
   * Marcar al asesor como "ocupado" para indicar que está atendiendo.
6. **Retornar true para indicar que el usuario fue atendido exitosamente**.



Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

**Diseño: conteoAtendidos()**

Objetivo: Contar cuántos usuarios han sido atendidos por todos los asesores en el CAU.

Pasos a seguir:

1. Inicializar una variable totalAtendidos en 0.
2. Recorrer la lista de asesores (asesores).
3. Para cada asesor, obtener la cantidad de turnos atendidos (obtenerAtendidos()).
4. Sumar la cantidad de turnos atendidos al contador totalAtendidos.
5. Retornar el total de usuarios atendidos.

Texto

El contenido generado por IA puede ser incorrecto.