

# AVANCE DE PROYECTO

Integrantes: Nicolas Soto - Julian Toro - Vicente Robles

- 1.1:

Nuestro proyecto consiste en un programa basado en la gestión de atención al cliente, en la cual designamos las áreas a trabajar junto con los clientes de estos separados por área. Y para esto implementamos un menú funcionando por consola con la que se podrán organizar, ingresar, y mostrar de una manera eficiente las diferentes colecciones del programa.

Y trabajaremos con las siguientes funciones implementadas en el menú:

1. **Crear Área de Atención:** en esta función se crea una clave en un mapa con el nombre que se le dará al área de atención, donde se ingresaran los clientes
2. **Crear Área con Clientes Iniciales:** en esta función se crea una clave en un mapa con el nombre que se le dará al área de atención, y al momento de crearla, automáticamente se pedirá incluir una cantidad de clientes iniciales que el usuario estime conveniente
3. **Añadir Clientes a un Área:** en esta función se selecciona numéricamente una de las áreas ya creadas para añadir manualmente clientes, ingresando el nombre y la edad de estos
4. **Mostrar listado de Clientes:** en esta función se despliegan todas las listas junto con todos los clientes que se encuentren en cada área
5. **Mostrar Clientes de un Área Específica:** en esta función se despliegan todos los clientes que se encuentren en el área que se seleccione

## ● 1.2:

Para organizar los datos se usaron dos colecciones anidadas, un mapa para usar como clave de búsqueda el nombre del área, junto con un arraylist para almacenar la lista de clientes en cada clave del mapa

Este programa trabaja con las siguientes clases:

1. **Clase AtencionClientes:** Esta clase contiene el método main que controla el flujo del programa y la interacción con el usuario a través de un menú. El programa utiliza un hashmap de Java para almacenar las áreas de atención como claves (String) y sus respectivas listas de clientes (Sublista) como valores.

- **Atributos:**

- coleccionAnidada (Map): Estructura que almacena las áreas y los clientes dentro de cada una.

- **Métodos:**

- crearSublista(): Permite la creación de nuevas áreas de atención de dos formas diferentes, con o sin clientes iniciales.
- agregarClientesSublista(): Permite agregar clientes manualmente a un área específica.
- mostrarColeccion(): Muestra todos los clientes en cada área, o puede mostrar a todos los clientes de solo un area seleccionada.

2. **Clase Clientes:** Esta clase almacena a los clientes que serán ingresados en las diferentes áreas a trabajar, ingresados con los atributos nombre(string) y edad(int).

- **Atributos:**

- nombre (privado): String, nombre del cliente.
- edad (privado): Entero, edad del cliente.

- **Métodos:**

- getNombre(), setNombre(): Getter y setter para el nombre.
- getEdad(), setEdad(): Getter y setter para la edad.
- Sobrecarga del método toString(): Permite imprimir la información completa del cliente o sólo su nombre si se utiliza una versión abreviada del método

3. **Clase Sublista:** Representa una sublista de clientes asociada a un área específica de la empresa, Tiene dos atributos: área (String) y clientes (una lista de objetos Cliente).

- **Atributos:**

- área (privado): String, el nombre del área de servicio.
- clientes (privado): Lista de objetos Cliente que contiene todos los clientes de esa área.

- **Métodos:**

- getArea(), setArea(): Getter y setter para el nombre del área.
- getClientes(), setClientes(): Getter y setter para la lista de clientes.
- Sobrecarga del método agregarCliente: Permite añadir clientes a través de un objeto Cliente o directamente con nombre y edad.



## ● 1.4

Como se muestra en el archivo java se crearon dos sublistas nombradas "Soporte Técnico" y "Atención al Cliente" junto con 4 clientes asignados con su nombre y edad

```
// Datos iniciales
Sublista area1 = new Sublista(area:"Soporte Técnico");
area1.agregarCliente(new Cliente(nombre:"Juan Perez", edad:30));
area1.agregarCliente(new Cliente(nombre:"Ana Gomez", edad:25));

Sublista area2 = new Sublista(area:"Atención al Cliente");
area2.agregarCliente(new Cliente(nombre:"Carlos Diaz", edad:40));
area2.agregarCliente(new Cliente(nombre:"Maria Lopez", edad:35));
```

## ● 1.5

El programa contiene dos tipos principales de conexiones anidadas:

1. **Áreas con Clientes (Sublista → Cliente):** La clase Sublista tiene un atributo clientes, que es una lista de objetos Cliente. Cada área (instancia de Sublista) puede tener múltiples clientes (instancias de Cliente), lo que constituye una relación uno-a-muchos.
2. **Colección de Áreas (AtencionClientes → Sublista):** En la clase AtencionClientes, se utiliza un mapa llamado coleccionAnidada, que almacena múltiples áreas (Sublista) con su nombre como clave. Esto conecta el nombre de cada área (como clave en el mapa) con la lista de clientes que pertenece a esa área.

## ● 1.6

En la clase AtencionClientes se utilizan dos sobrecargas de métodos, las cuales son:

Crear Área de Atención  (Map<String, Sublista> coleccionAnidada, Scanner scanner)	Esta función crea el área con el nombre designado
Crear Área con Clientes Iniciales  (Map<String, Sublista> coleccionAnidada, Scanner scanner, List<Cliente> clientesIniciales)	Esta función crea el área con el nombre designado y te da la opción de agregar clientes inmediatamente

Mostrar listado de Clientes  (Map<String, Sublista> coleccionAnidada)	Esta función muestra todos los clientes de todas las áreas creadas
Mostrar Clientes de un Área Específica  (Map<String, Sublista> coleccionAnidada, Scanner scanner)	Esta función muestra todos los clientes de un área específica seleccionada previamente

Tanto en la clase Sublista como en Cliente, se usan métodos sobrecargados para permitir agregar clientes de distintas maneras y mostrar la información en distintos formatos.