

“Sistema de atención al cliente”

Nuestro proyecto busca implementar un sistema de atención al cliente, enfocado principalmente a áreas de servicio. El sistema cuenta con un sistema de inserción y visualización de los datos de los clientes a través de la consola, colecciones anidadas de objetos, las sobrecargas de métodos, el uso de colecciones del Java Collections Framework y el diseño de clases.

Datos:

En este sistema se manejan dos tipos principales de datos:

1. **Cliente:** Representa a una persona que requiere atención en un área determinada. Los atributos son:
 - nombre: String que almacena el nombre del cliente.
 - edad: Entero que indica la edad del cliente.
2. **Sublista (Área):** Representa una colección de clientes atendidos en un área específica de atención al cliente. Los atributos son:
 - area: String que define el nombre del área.
 - clientes: Lista de objetos de tipo Cliente que contiene a las personas que están siendo atendidas en esa área.

Además, se utiliza una estructura de tipo **Map**, que asocia cada área con su respectiva sublista de clientes.

Funcionalidad:

El sistema permite realizar las siguientes operaciones principales:

1. **Creación de áreas de atención (Sublista):** El usuario puede crear nuevas áreas donde se atenderán clientes. Cada área está representada por un nombre único y una lista de clientes.
2. **Añadir clientes a un área específica:** El sistema permite agregar clientes manualmente a un área determinada. Cada cliente está identificado por su nombre y su edad.
3. **Mostrar listado de clientes:** El sistema ofrece la posibilidad de mostrar todos los clientes que pertenecen a cada área registrada.
4. **Sobrecarga de métodos:** Se implementa la sobrecarga de métodos en dos clases (Cliente y Sublista). Por ejemplo, en la clase Sublista, el método agregarCliente permite añadir clientes ya sea a través de un objeto Cliente o mediante el nombre y la edad directamente.

Clases:

El diseño conceptual se basa en tres clases principales:

La clase `Cliente` representa a un individuo que será atendido en una de las áreas de servicio.

- **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.

La clase `Sublista` representa una colección de clientes agrupados por un área de servicio.

- **Atributos:**
 - `area` (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.

La clase “`AtencionClientes`” maneja la lógica principal del sistema y el menú de usuario. El sistema se basa en un `Map` para gestionar las áreas y los clientes asociados a cada una de ellas.

- **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.
 - `agregarClientesSublista()`: Permite agregar clientes manualmente a un área específica.
 - `mostrarColeccion()`: Muestra todos los clientes en cada área.

