

# Documento de Descripción del Programa de Agenda Telefónica

Este es un programa para gestionar una agenda telefónica. El programa permite agregar, listar, buscar, eliminar contactos y realizar otras operaciones relacionadas con la gestión de una agenda.

El programa consta de tres clases principales:

## 1. **\*\*Clase Contacto:\*\***

- Esta clase representa un contacto dentro de la agenda.
- Tiene dos atributos principales: ``nombre`` y ``telefono``.
- Cuenta con métodos como ``equals`` para comparar contactos por nombre, y ``toString`` para mostrar la información de un contacto.

## 2. **\*\*Clase Agenda:\*\***

- Esta clase maneja un arreglo de contactos.
- Tiene varios métodos para interactuar con los contactos, tales como:
  - **`**aniadirContacto(Contacto c):**`** Añade un contacto a la agenda si no existe ya, y si hay espacio.
  - **`**existeContacto(Contacto c):**`** Verifica si un contacto ya está en la agenda.
  - **`**listarContactos():**`** Muestra todos los contactos presentes en la agenda.
  - **`**buscarPorNombre(String nombre):**`** Busca un contacto por su nombre y muestra su teléfono.
  - **`**agendaLlena():**`** Verifica si la agenda está llena o si hay espacio para más contactos.
  - **`**eliminarContacto(Contacto c):**`** Elimina un contacto de la agenda.
  - **`**huecosLibre():**`** Indica cuántos huecos vacíos hay en la agenda.

## 3. **\*\*Clase AgendaT (Interfaz de Usuario):\*\***

- Esta clase contiene el menú de opciones y permite al usuario interactuar con la agenda.
- Permite agregar, listar, buscar, eliminar y verificar la disponibilidad de la agenda.
- También maneja la persistencia de los datos de la agenda en un archivo de texto llamado "agenda.txt".

A continuación se describen los métodos de cada clase:

### ### Clase Contacto:

#### - \*\*Constructores:\*\*

- `public Contacto(String nombre, int telefono)` : Constructor que crea un contacto con nombre y teléfono.
- `public Contacto(String nombre)` : Constructor que crea un contacto con nombre solo.

#### - \*\*Métodos:\*\*

- `getNombre()` : Devuelve el nombre del contacto.
- `setNombre(String nombre)` : Establece el nombre del contacto.
- `getTelefono()` : Devuelve el teléfono del contacto.
- `setTelefono(int telefono)` : Establece el teléfono del contacto.
- `equals(Contacto c)` : Compara el nombre del contacto actual con otro contacto.
- `toString()` : Devuelve la información del contacto en formato de texto.

### ### Clase Agenda:

#### - \*\*Constructores:\*\*

- `public Agenda()` : Crea una agenda con 10 contactos.
- `public Agenda(int tamaño)` : Crea una agenda con el tamaño especificado.

#### - \*\*Métodos:\*\*

- `aniadirContacto(Contacto c)` : Añade un contacto a la agenda si no existe y hay espacio.
- `existeContacto(Contacto c)` : Verifica si un contacto existe en la agenda.

- `listarContactos()`: Muestra todos los contactos en la agenda.
- `buscarPorNombre(String nombre)`: Busca un contacto por su nombre.
- `agendaLlena()`: Verifica si la agenda está llena.
- `eliminarContacto(Contacto c)`: Elimina un contacto de la agenda.
- `huecosLibre()`: Devuelve la cantidad de espacios vacíos disponibles en la agenda.

### ### Clase AgendaT:

#### - **Métodos:**

##### - **Interacción con el usuario:**

- Permite al usuario elegir una opción del menú (Añadir, Listar, Buscar, etc.).
- Permite al usuario introducir datos como nombre y teléfono.
- Maneja la persistencia de los datos mediante un archivo de texto "agenda.txt".

##### - **Flujo general:**

- El usuario puede elegir entre las siguientes opciones:

1. Añadir contacto
2. Listar contactos
3. Buscar contacto
4. Verificar si existe un contacto
5. Eliminar un contacto
6. Ver contactos disponibles
7. Verificar si la agenda está llena
8. Salir

En cada opción, el programa interactúa con la clase **Agenda** para realizar las acciones correspondientes, y guarda o actualiza los cambios en el archivo "agenda.txt".

### ### Comentarios en el Código:

#### #### Clase Contacto:

- Los constructores permiten crear objetos Contacto con nombre y teléfono. Si solo se proporciona el nombre, el teléfono se establece a 0.
- El método `equals` compara los contactos por nombre sin considerar mayúsculas/minúsculas ni espacios.
- El método `toString` permite mostrar la información del contacto de manera legible.

#### #### Clase Agenda:

- La agenda se implementa como un arreglo de objetos Contacto.
- El método `añadirContacto` añade un contacto si no existe y si hay espacio disponible. Si la agenda está llena o el contacto ya existe, muestra un mensaje correspondiente.
- El método `listarContactos` muestra todos los contactos que no son nulos.
- El método `buscarPorNombre` busca un contacto por nombre e imprime su teléfono si lo encuentra.
- El método `eliminarContacto` elimina un contacto comparando nombres.
- El método `agendaLlena` verifica si la agenda está llena.
- El método `huecosLibre` devuelve el número de espacios disponibles en la agenda.

#### #### Clase AgendaT:

- La clase maneja la interfaz con el usuario a través de un menú.
- Permite al usuario realizar las operaciones de la agenda y actualizar el archivo "agenda.txt" con los cambios.