

UNIVERSIDAD DEL VALLE DE GUATEMALA

Introducción a la ingeniería en Ciencias de la Computación y tecnologías de la
información

Ing. Ludwing Cano



LABORATORIO 4

ANÁLISIS

Angela García #22869

Sergio Palacios #22808

Augusto Sanic #20717

GUATEMALA, noviembre de 2022

Introducción:

Este proyecto busca como objetivo realizar un software para la empresa “Mercedes-Benz”, la idea es realizar un sistema para los radios de los carros de esta marca. Existen tres clases de carro (A,B,C), en este caso este grupo se centrará en el desarrollo de la clase A, Implementado una “Interface”, que permita cumplir con todos los requerimientos que la marca exige para los radios, específicamente de la clase A. Después de desarrollar este programa, este será intercambiado con otro grupo de desarrolladores que también trabajan en el desarrollo de los radios para los carros clase A, para de esta forma poder experimentar si al modificar una línea de código, el software desarrollado sigue funcionando de manera óptima.

Clases del Modelo

1. Clase *Canciones*

- **Explicación de clase:**

Clase canciones, las cuales representarán al objeto de la canción presente en el modelo.

- **Atributos de la clase:**

nombre, duracion, Autor, genero del tipo string

- **Métodos de la clase:**

Métodos públicos set y get para obtener y settear los atributos de la clases canciones:

En el apartado de los getters, se establecen los métodos, public string, getNombre, getDuracion, getAutor y getGenero. Los cuales retornan nombre, duracion, autor y genero.

En los setters se tienen métodos public void (sin retornar valor). Las Cuales tienen en los parámetros del tipo string, nombre, duracion, autor y género.

Además se tiene el método constructor de la clase en el cual se mandan a llamar los atributos.

1. Clase *contactos*

- **Explicación de clase:**

la clase contactos, como el nombre indica representa la parte de los contactos, los cuales se identifican con el nombre y número

- **Atributos de la clase:**

La clase cuenta con dos atributos, uno del tipo string Nombre y otro del tipo int Numero.

- **Métodos de la clase:**

la clase presenta los getters publicos, uno del tipo string y otro del int, es decir getNombre, getNumero. Mientras que para los setters, se presentan los metodos public void (sin retorno) de setNombre (con el parametro string nombre) y el setNumero (con el parametro int numero).

2. Interfaz *IRadio*

- **Explicación de la interfaz:**

La interfaz de IRadio contiene una colección de métodos abstractos, los cuales representan las acciones u cambios de la radio. No presentando propiedades constantes.

- **Métodos abstractos de IRadio**

- public Boolean Estado();

Este método definirá el estado (modo radio) de la radio (dentro del controlador) y asimismo de la clase Radio.

- public void CambiarModulacionDeOnda();

Este método como el nombre indica cambiará el tipo de onda FM o AM desde el controlador y a partir de la modulación de onda de la clase *Radio*.

- public void CambiarEmisora(int opcion);

Dentro de este método, se relaciona la emisora sintonizada, la cual variará (en sumas o restas de frecuencia ± 0.5), dentro del parámetro opción se introduce la opción entera del usuario en función del cambio de frecuencia.

- public void GuardarEmisora();

Se refiere al método GuardarEmisora, que asigna el arreglo de emisoras a la lista.

- public void CargaEmisora(int indice);

Se refiere al método en el cual muestra la emisora sintonizada

- public void SeleccionarLista(int opcion);

Este método abstracto pretende seleccionar la lista de reproducción a partir de la función que ingrese el usuario.

- public void CambiarCancion (int opcion);

- public void EscucharCancion();

Ambos métodos vinculados con la reproducción de la canción, el primero a partir de la opción que ingrese el usuario cambiaba la canción, mientras que el segundo pretende mostrar la canción actual.

- public Boolean EstadoTelefono();

Método que muestra la parte del estado booleano del telefono.

- public void MostrarContactos();

Método abstracto que pretende mostrar los contactos

- public void LlamarAContacto (int opcion);

metodo que pretende establecer o no la llamada al contacto.

- public void FinalizarLlamada();

Este metodo abstracto pretende finalizar la llamada.

- public void CambiarModo();
Muestra y cambia, el modo en el que se encuentra el audio del coche
- public void PlanificarViajes (String fecha, String Pais, String Ciudad, int noBoletos)
Basicamente extrae apartir de los parametros los valores de los viajes planificados, con las características de fecha, el pais, la ciudad y el No. de boletos.

3. Clase *Radio*

- **Explicación de clase:**

- Esta clase representa a la radio, la cual tendrá las características del objeto desplegado como el estado, el volumen, las ondas, emisoras, estado del telefono, estado de la llamada, audio además de la invocación de las clase arraylist, para generar arreglos dinámicos de las EmisorasLista, canciones, Contactos y Viajes donde se almacenan los datos.

- **Atributos de la clase:**

- Boolean estado;
Este atributo es del tipo boolean y el cual representa el estado, verdadero o falso en el que se encuentra la radio.
- int volumen;
representa el volumen de la radio
- String ModulacionDeOnda;
representa la modulacion AM o FM posible de la Onda.
- Double emisoras;
Representa la modulacion de ondas Ej: 1.05.
- Int NoLista = 0;
Se refiere al número que tendrá la lista vinculada a la lista de reproducción de canciones, inicializada con el entero en 0.
- Int IcancionActual = 0;
Indica el número de la canción la cual se está reproduciendo de la lista de reproducción, encontrándose inicializada en cero.
- Boolean estadoTelefono;
Se refiere al modo del teléfono, verdadero o falso.
- Boolean EstadoLlamada;

- se refiere a dentro de una llamada si esta se encuentra dentro de una, true o en caso contrario no (false, de llamada finalizada).
- String Audio;
Se refiere al modo de auriculares o speaker.
- **Métodos de la clase:**
- el metodo public ArrayList<Viajes> getViajes(), es un metodo get, el cual retorna el arreglo de Viajes. Ademas que el metodo de void setViajes, permite ingresar insertar un arreglo de objetos de tipo Viajes.
- El metodo getEstadoLlamada devuelve el valor booleano que tenga la llamada. Ademas la parte del setEstadoLlamada devuelve en función de ser true o false, los valores contrarios de la misma.
- Los métodos setters y getters de, getIcancionActual, la cual devuelve el valor de la canción actual reproduciendo, ademas del setIcancionActual, el cual es void (no valores retornables) y permite establecer el valor del atributo.
- Los métodos setters y getters (de tipo entero) de, getNoLista, la cual devuelve el valor la lista de reproduccion, ademas del setNoLista, el cual es void (no valores retornables) y permite establecer el valor del atributo.
- Los métodos boolean getters y setters. de, getEstado, la cual devuelve el v o f de la radio. Ademas la parte del setEstado devuelve en función de ser true o false, los valores contrarios de la misma, retornando el estado.
- Los métodos setters y getters (de tipo entero) de, getVolumen, devuelve el valor del volumen. setVolumen como void y con el parametro (int tempVolumen) devuelve a partir de si el volumen es mayor o igual a cero y si la tempVolumen es 1 o 0, el aumento en (++) o la disminución (–) de la misma.
- Los métodos setters y getters (de tipo String) de, getModulacionDeOnda, devuelve el valor del string de la onda de radio en sintonización. En caso de setModulacionDeOnda, esta a partir del parámetro tempmodulacioDeOnda, cambiara o volvera valores reseteados contrarios de “FM” a “AM” o viceversa.
- Los métodos setters y getters de tipo double de las emisoras, devolverán tanto como establecen los valores de las emisoras (sin valor de retorno en set).
- El método public ArrayList<Double> getEmisorasLista(), es un método get, el cual retorna el arreglo de EmisorasLista. Además que el método de

void setemisorasLista, permite ingresar insertar un arreglo de objetos de tipo EmisorasLista.

- El método public ArrayList<Canciones> getCanciones(), es un método get, el cual retorna el arreglo de Canciones. Además que el método de void setCanciones, permite ingresar insertar un arreglo de objetos de tipo Canciones.
- El método public ArrayList<Contactos> getContactos(), es un método get, el cual retorna el arreglo de Canciones. Además que el método de void setContactos, permite ingresar insertar un arreglo de objetos de tipo Contactos.
- El método get y set Audio, permite no solo devolver el audio en el get, sino el set (ambos de tipo string) permite en función de si es “speaker” o “audífonos” devolver los valores en viceversa, en caso de ser llamado.
- El método get y set Estadodetelefono de tipo boolean, devuelve los valores true o false, además que en el set, establece dichos valores.
- Además presenta los métodos constructores donde se declaran los valores de los atributos.

4. Clase *Reproducir*

- **Explicación de clase:**

Esta clase representa la parte de reproducción de las listas de las canciones presentes, presentando un arraylist de canciones instanciada en la misma.

- **Atributo de la clase:**

String nombre;

- **Métodos de la clase:**

public String getNombre()

metodo que retorna el valor del nombre en reproduccion.

public void setNombre(String nombre)

se pone el valor asignado a nombre recibida en parámetro.

public ArrayList<Canciones> getCanciones()

retorna el arreglo de canciones y el public void setCanciones(ArrayList<Canciones> canciones) establece los valores del arreglo, además del constructor de la clase.

5. Clase *Viajes*

- **Explicación de clase:**

Esta clase representa la parte de asignación y características del viaje, por parte del usuario. Esta clase se implementa únicamente aplicable a los carros de clase “A”.

- **Atributo de la clase:**

Se presentan atributos de tipo string como Fecha, País, Ciudad que aluden a las características del viaje, además de boletos de tipo de int, el cual representa el número de boletos en el viaje.

- **Métodos de la clase:**

La clase presenta los getters y setters de tipo público, getFecha retorna el valor de Fecha, setFecha asigna el valor de Fecha a partir del valor que recibe del parámetro. Lo mismo ocurre con el país y ciudad, puesto que los anteriores son de tipo String. Finalmente setBoletos y getBoletos, son del tipo entero y devuelve o asigna la cantidad de boletos de la propiedad.

Paquete CVista

Clase: Main

- **Explicación de clase:**

Esta Clase es en la cual el programa procederá a ejecutarse.

- **Métodos de la Clase:**

MostrarPrograma(); Este es un método de la clase MostrarMenu, el cual se encuentra en el mismo paquete de CVista, este método tiene como función principal realizar una interacción amigable con el usuario para que este pueda navegar en el problema.

Clase: MostrarMenu

- **Explicación de clase:**

La función principal de esta clase es crucial, principalmente por el hecho de que esta es la que permite la interacción directa con el usuario, proporcionando todos los menús del programa, como por ejemplo, encender y apagar la radio, cambiar el volumen, Modo productividad etc.

- **Atributos de la Clase:**

int opcion = 0

int opcionRadio = 0


```
int opcionReproduccion = 0  
int opcionTelefono = 0
```

Todos estos atributos de tipo entero, tiene como objetivo poder utilizarse en los métodos de la clase para poder darle un valor a las diferentes funciones de esta clase.

- **Métodos de la Clase:**

Esta Clase cuenta con dos métodos el primero es MostrarMenu(), el cual es de tipo público y lo que hace es instanciar dos clases, vista y Controlador.

Mientras que el otro método también de tipo público, tiene como objetivo mostrar los diferentes menús del programa.

Clase: Teclado

- **Explicación de clase:**

Esta clase tiene como objetivo validar los espacios de los nombres al igual que el de los apellidos.

- **Atributos de la clase:**

Scanner leer;

Este atributo tiene como objetivo utilizar la función de la clase java.util.Scanner; para así poder ingresar distintos datos al programa.

String string1;

Este atributo tiene como objetivo colocar ingresar datos de tipo string.

- **Métodos de la clase:**

Public Teclado() {},

Este atributo tiene como objetivo ingresar datos que luego el programa se encargará de leer.

Public String getString() {},

Este método tiene como función, que el usuario ingrese su nombre y apellido y este método lo guarde utilizando java.util.scanner.

Clase: Vista

- **Explicación de clase:**

Esta clase tiene como objetivo, controlar lo que el usuario introduce en el programa para así poder saber si la información es correcta o no.

- **Atributos de la clase:**

Scanner leer; Este tiene la función de validar los espacios en el texto ingresado.

int Opcion; tiene como función devolver un entero en el método que será utilizado como menú del programa.

String respuesta; Tiene como función devolver un valor de tipo String, el cual servirá dentro del programa para poder entender la acción que el usuario desea implementar.

- **Métodos de la clase:**

public Vista() {}, Está tiene como objetivo recibir y guardar datos que luego el programa leerá.

public Mensaje() {}, Éste tiene como objetivo que el usuario escoja su acción dentro del programa.

public int publicMensaje() {}, Este método busca, que el usuario pueda escoger de forma sencilla el menú que desea emplear dentro del sistema.

public void validación () {}, muestra un mensaje en la consola, indicando que el dato que ingresó es incorrecto.

Public void Validacion2() {}, muestra un mensaje en la consola, indicando que el dato que ingresó es incorrecto.

Paquete MostrarMenu

ClaseControlador:

- **Explicación de clase:**

Esta clase tiene como objetivo principal unir y controlar todos los datos que en el programa se manejan, tanto los que se ingresan como las que ya se encuentran establecidos en el programa .

- **Métodos de clase:**

`ArrayList<Reproducir>`

Este tiene como función principal reproducir las diferentes canciones del radio, a través de Spotify.

`ArrayList<Double>`

Este método es para poder realizar listas de canciones próximas en la radio.

`Boolean Estado(){}`

Tiene como objetivo indicar si el radio está apagado o por el contrario encendido.

`void Cambiar volumen(int volumen){}`

Su función es reducir o aumentar el volumen del radio.

`voidCambiarModulacionDeOnda(){}`

Este tiene como función cambiar la modulación de radio de FM a AM o viceversa.

`voidCambiarEmisora(int opcion){}`

Este método permite que a través de valores enteros el usuario pueda seleccionar la emisora de radio que desee escuchar.

`void GuardarEmisora(){}`

Su funcionalidad es permitir que el usuario pueda guardar las emisoras de radio que desee.

`void MostrarEmisoras(){}`

Este método se encarga de mostrar en forma de lista todas las emisoras disponibles, para que el usuario las sintonice.

```
void CargaEmisora(int indice){}
```

Este método muestra en una lista, cual emisora el usuario está escuchando en ese momento.

```
Void SeleccionarLista(int opcion){}
```

El método de SeleccionarLista, busca que el usuario pueda escoger a través de un número la lista de reproducción que desea escuchar.

```
voidCambiarCancion(int opcion){}
```

Esté método es para cuando el usuario está utilizando spotify y desea cambiar la canción.

```
voidEscucharCancion(){}
```

El método indica que canción se está reproduciendo.

```
voidEstadoTelefono(){}
```

Indica cuando el teléfono está funcionando en la radio del carro.

```
voidMostrarContactos(){}
```

Muestra los contactos que el usuario tiene en su teléfono.

```
voidLlamarAContactos(int opcion){}
```

Realiza una llamada a la persona que el usuario desee.

```
voidFinalizarLlamada(){}
```

Finaliza la llamada en curso.