



Universidad del Valle de Guatemala
Facultad de Ingeniería
Carrera, Ciencias de la computación

“Laboratorio 4”
Análisis y diseño
Grupo 4

Autores:

Nelson García, 22434
Joaquín Campos, 22155
Oscar Fuentes 22763

Docente:

Lynette García Perez

Auxiliares:

Rodrigo Barrera
Gabriela Contreras

Link del repositorio: <https://github.com/nel-eleven11/Laboratorio-4-Grupo-4.git>

Análisis:

- **Se identifican correctamente las clases que se necesitan para resolver el problema. El número de clases identificadas es suficiente para darle solución a la situación planteada.**
 - principal
 - cancion
 - contacto
 - radio
- **Se identifica correctamente la interfaz que debe diseñarse para resolver el problema planteado.**
 - modo_estado
 - modo_volumen
 - modo_radio
 - modo_reproducción
 - modo_teléfono
 - modo_productividad
- **Se explica correctamente y de forma lógica el propósito de cada una de las clases.**
 - principal
 - Esta es la clase que se encargará de ejecutar el programa entero.
 - cancion
 - Sirve para poder obtener los datos de la canción y luego desplegarlos en la consola
 - contacto
 - Sirve para poder hacer un toString y poderlo mostrar en consola
 - radio
 - Es la clase en donde irán los menús, la general, radio es la clase que permitirá al usuario usar todas las interfaces e interactuar con su dispositivo.
 - modo_estado
 - Es la clase que por medio de un método definirá si la radio está encendida o apagada.
 - modo_volumen
 - La clase tendrá el método para controlar el volumen siendo este un booleano que si se le desea subir volumen será true o un false será para bajar volumen.
 - modo_radio
 - Será la clase que permitirá controlar el área de emisoras, como por ejemplo, subir o bajar de frecuencia, cambiará entre FM y AM dar opción de guardar en la lista de emisoras que se quieran guardar.

- modo_reproducción
 - Será la clase que permite que el usuario interactúe con los métodos para las canciones que no son de radio o sea las playlist que se creen, este podrá adelantar, atrasar, seleccionar lista de reproducción, entre otros.
- modo_teléfono
 - Es la clase que informa y permite que el usuario pueda acceder a las opciones de teléfono como poder hacer llamadas, finalizarlas, tener el audio en audífonos o en speaker, entre otros.
- modo_productividad
 - Es la clase que le permite al usuario poder planear sus viajes.
- **Se identifican correctamente todos los atributos de cada una de las clases seleccionadas. Son los necesarios para resolver el problema planteado.**
 - principal:
 -
 - canción:
 - nombre: String, privado
 - duracion: String, privado
 - autor : String, privado
 - genero: String, privado
 - contacto:
 - nombre: String, privado
 - telefono: String, privado
 - radio:
 - emisoras: ArrayList<double[]>, privado
 - cancion: ArrayList<cancion>, privado
 - contactos: ArrayList<contacto> , privado
 - modulacion : String, privado
 - estado : String, privado
 - conectar: boolean, privado
 - emisora: double, privado
 - volumen : int, privado
 - posicion : int, privado
 - modo_audio : String, privado
- **Se identifican correctamente los atributos y/o variables polimórficos.**
 - estado(boolean)
 - volumen(int)
 - cambiar(boolean)
 - cambiar_e(boolean)
 - cambiar_c(boolean)
 - conexion(boolean)
 - contactos (String)
 - llamar_c (String)
 - finalizar (String)
 - audio (boolean)
- **Se explica correctamente y de forma lógica el propósito de cada uno de los atributos.**
 - estado(boolean)

- Su propósito es determinar por medio de un boolean si la radio está encendida o apagada.
- cambiar(boolean)
 - Su propósito es determinar por medio de un boolean si la radio está en FM o AM.
- cambiar_e(boolean)
 - Su propósito es determinar por medio de un boolean si la frecuencia debe cambiar .5 hacia arriba (true) o hacia abajo (false)
- cambiar_c(boolean)
 - Su propósito es determinar por medio de un boolean si la cancion en reproduccion debe saltarse o retroceder.
- conexion(boolean)
 - variable polimórfica que permite determinar si el telefono está o no conectado.
- llamar_c (String)
 - variable polimórfica que permite llamar al contacto seleccionado.
- finalizar (String)
 - permite finalizar la llamada empezada
- audio (boolean)
 - determina si el audio es en auriculares o en speaker
- **Se identifican correctamente los métodos necesarios para resolver la situación planteada. Explica correctamente y de forma lógica el propósito de cada uno de los métodos de las clases.**
 - guardar()
 - Se encargará de almacenar las estaciones que el usuario desee guardar en la radio, evaluando que no pase el máximo de 50 estaciones.
 - volumen(boolean)
 - cargar()
 - se encargará de mostrar todas las estaciones posibles y permitir al usuario escoger una que será la primera en cargar
 - seleccionar()
 - Es el método que mostrará la lista de reproducción y le permitirá al usuario escoger la lista que quiera que se reproduzca.
 - contacto()
 - método por el cual se crea un contacto para agregarlo a la lista de contactos
 - viajes()
 - El método en el que se generará un plan de viaje para el usuario.