

## **Laboratorio 4 – Polimorfismo Análisis**

### **Requisitos funcionales**

- Encender/apagar
- Elegir modo
- Subir/bajar volumen
- Cambiar emisora
- Guardar emisora
- Cargar emisora
- Seleccionar lista de reproducción
- Cambiar canción
- Escuchar canción
- Conectar/desconectar
- Mostrar contacto
- Llamar contacto
- Finalizar llamada
- Cambiar a llamada en espera
- Ver pronostico del tiempo
- Mostrar el estado de la radio, con información relevante en el modo que se encuentre.

### **Clases por utilizar/atributos/métodos**

**La clase controladora:** Es la clase la cual va a ser la principal donde se van a crear los arraylist, para los contactos y las emisoras. Esta es la clase donde se implementa los ciclos para verificar las entradas de los usuarios y de esa forma mostrarle al usuario lo que está pidiendo, y responder a respuestas indebidas por parte del usuario.

**La clase Radio:** Esta clase es una clase abstracta con atributos protegidos, de tipo doble para el atributo emisora y de tipo entero para el volumen. Esta clase también tiene como métodos bajarle y subirle volumen, bajar y subir de estación, guardar una estación, poner una playlist, adelantar y retroceder una canción, conectarse y desconectarse a un teléfono, mostrar la lista de los contactos, y hacer una llamada.

**La clase RadioC:** La cual es la clase hija de Radio e implementa la interfaz Carc. En esta se visualiza el poner una llamada en espera, y mostrará el pronóstico del tiempo. Ambas a través de interfaces. Esta clase no tiene atributos solo métodos los cuales son imprimir los mensajes que se le muestran al usuario en caso de poner una llamada en espera o ver el pronóstico del tiempo.

**La clase Vista:** La clase vista es la que se usa para imprimir los menús de las opciones para que el usuario pueda verlos cuando se corre el código. Esta clase no tiene ningún atributo, por lo que solo cuenta con los métodos de SetRadioMode, que sirve para mostrar las opciones de que modo de radio desea acceder, ModoRadio sirve para mostrar solo las opciones para el modo Radio, ModoReproduccion, sirve para mostrar solo las opciones para el modo Reproducción, sirve para mostrar solo las opciones para el modo ModoTelefono y ModoTelefonoCarroS, sirve para mostrar todas las opciones que cuenta el Carro tipo C.

**La clase CarC:** Esta es la clase de la interfaz que tiene los métodos de poner la llamada en espera y ver el pronóstico del tiempo.

**La Clase canción:** Esta clase servirá para poder hacer posible el guardado de varias canciones en un playlist, por lo que contará con los atributos de Nombre, Género y Autor, que servirán para obtener el nombre, Género y autor de la canción, respectivamente, serán de un modificador privado y todos serán de tipo String. Asimismo, estará una variable Duración que servirá para almacenar la duración de la canción por lo que será de tipo Duration, y será un modificador privado. También, contará con una variable Numero que servirá para almacenar el número de canción que es, la cual será de tipo entero y será un modificador privado. En esta clase solamente se utilizará el método ToString, ya que servirá para mostrar al usuario todos estos datos anteriormente mencionados.

### **Link del Repositorio**

<https://github.com/gabrielpaz2003/Lab4.git>

### **Reflexión**

Los cambios que se tuvieron que hacer fueron varios ya que no nos funcionó con las interfaces que se habían previsto anteriormente así que hicimos nuestras propias interfaces para que el usuario pueda hacer todo lo que se pide en el programa. Asimismo, podemos confirmar que fuimos un buen grupo, que tuvimos muy buena comunicación y que pudimos hacer todo lo que se nos pedía en el programa para que funcionará correctamente.

Los tropiezos que tuvimos fue de que no sabíamos implementar las interfaces que propuso otro grupo y le comentamos que no lo hicieramos así y aún así insistió que las hicieramos así, y pues ese fue el único inconveniente que tuvimos como grupo.