

## **Laboratorio 4 - Análisis**

### **Requisitos funcionales:**

- Lograr que ciertas funcionalidades solamente estén disponibles para cierto tipo de carro.
- Cambiar el volumen
- Tener distintos modos (radio, reproductor de música, etc)
- Cambiar el volumen
- Encender y apagar el radio
- Cambiar de FM a AM
- Cambiar emisora en intervalos de 0.5
- Guardar emisoras
- Cargar emisoras
- Seleccionar Lista de reproducción
- Cambiar canción
- Escuchar canción
- Conectar/desconectar teléfono
- Mostrar contactos
- Llamar contactos
- Finalizar una llamada
- Cambiar de speaker a auriculares
- Llamar al último contacto
- Cambiar a llamada de espera
- Planificar un viaje
- Ver tarjetas de presentación
- Ver el pronóstico del tiempo
- Mostrar el estado del radio

### **Análisis:**

### **Clases, métodos y atributos:**

- Cada modo (reproducción, radio, etc), tendrá su propia clase que implemente la GUI. Todas estas clases tendrán un método constructor el cuál contiene las componentes y acciones de la GUI.
- Las clases TurnONOFFGUI, ChooseCarGUI y MainMenuGUI sirven para algunos aspectos del radio cómo elegir la clase de carro, encender la radio y desplegar un menú el cuál permite seleccionar el modo del radio. La clase TurnONOFFGUI tiene el método main ya que es la primera en ejecutarse. Las demás clases solamente tienen un método constructor el cuál posee todas las componentes de la GUI.

- Cada funcionalidad (conectar teléfono, pronóstico del tiempo, etc) tiene su propia GUI. Cada una de estas clases tiene un método constructor el cuál posee todas las componentes de la respectiva GUI de la función.
- ICarC es una interfaz que posee los métodos que un carro tipo C debería implementar. Tiene los siguientes métodos: SwitchRadioONOFF, getVolume, setVolume, getRadioStatus, SwitchFrequency, setStation, getStation, saveStation, SelectSpecificSong, getAudioRepType, setAudioRepType, getAListOfSongs, nextSong, prevSong, Playsong, Connect\_disconnectPhone, getPhoneConnectionStatus, getListOfContacts, getCallStatus, makeCall, PlaceCallonHold, endCall y getWeatherForeCast.
- ICarB es una interfaz que posee los métodos que un carro tipo B debería implementar. Tiene los siguientes métodos: getLastContact, setLastContact, y getPresentationCards
- ICarA es una interfaz que posee los métodos que un carro tipo A debería implementar. Tiene los siguientes métodos: getSpeakerStatus, changeSpeakerStatus y TripPlanification.
- CarClass es una clase abstracta la cuál sirve para que todas las clases que sean una clase de un carro sean del mismo tipo. Esta clase provee las características fundamentales que un auto debería tener. Implementa ICarC, por lo que tiene los mismos métodos. Los atributos de esta clase son: volume, station, stations, frequency, isON, carClass, isPhoneConnected y contacts.
- CarC, CarA y CarB son clases las cuáles heredan de CarClass y sirven para almacenar propiedades del radio y llevar a cabo algunas acciones de este. CarA implementa ICarA, y CarB implementa ICarB, por lo que estas clases tienen los mismos métodos que las interfaces. Al heredar de CarClass, estas clases también implementan ICarC. Tienen los mismos atributos que la clase padre (CarClass).