

Análisis y documentación  
Hoja de Trabajo #1

Para la realización de la hoja de trabajo número 1 de la clase de algoritmos y estructura de datos consistió en la realización de un simulador de una radio. Primero que todo, se definió durante el periodo de clase la interfaz que serviría de base para la realización del mismo. Posteriormente, como pareja se desarrollaron los métodos necesarios para el correcto funcionamiento.

Para llevar un control de versiones sobre el programa a desarrollar, se utilizó la plataforma GitHub, creando un repositorio público para que ambos miembros del grupo pudiéramos subir las distintas versiones del programa.

Los requerimientos establecidos en el pdf brindado son los siguientes: Diseñar una estructura de datos que funcione como un simulador de radio. Tomando en cuenta el estado apagado/encendido de la radio, así como también las distintas frecuencias AM o FM. Así mismo, se designó 12 botones modificables, además de permitir que se almacenen frecuencias en esos botones. Por último se designó como funcionan los rangos de frecuencias.

El programa se realizó con la correcta documentación generada con JavaDoc, mostrando al autor de cada método, los parámetros que recibe y aquellos valores que retorna. Se muestra el estilo de codificación correcto, con comentarios, indentación, nombres de variables correctos.

Se desarrolló correctamente el programa, todas las características a cumplir se realizaron. Por el siguiente enlace se puede observar el correcto funcionamiento del programa: <https://youtu.be/tWTyTD6nwW4>

Se desarrollaron las siguientes clases:

FuncionRadio: Esta clase se realizó con el propósito de correr los procesos a utilizar por la radio.

Boton: Esta clase se creó para los 12 botones a utilizar por la radio

Radio: Esta es la clase intercambiable, consiste en utilizar los métodos de la interfaz y reemplazarlos con las acciones correspondientes.

Adicional a la clase que se decidió durante el periodo de Estructura de Datos con un consenso entre todos los compañeros que se denominó como "Interface". En esta clase se crearon los 5 métodos a implementar en las distintas clases creadas por nosotros.

El método implementado de la interfaz llamado amFm() devuelve false, se tomó como que el estado de la radio está pagada. Por ende el método no es ejecutable en ese caso, de lo contrario devuelve un true y cambiará de frecuencia.

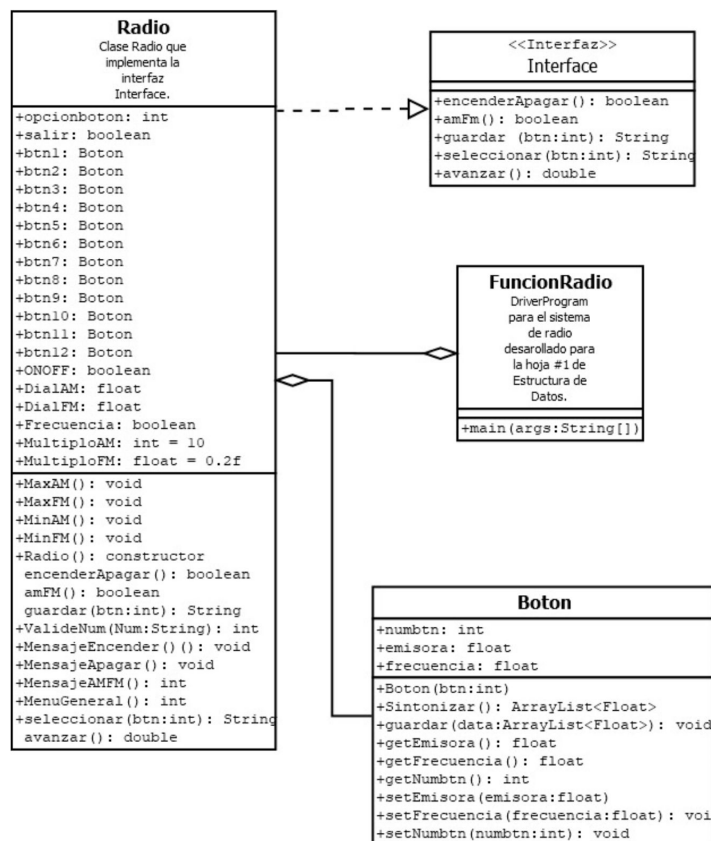
Para el desarrollo de este programa, también se exportó el proyecto en formato .jar; se procede a mostrar una fotografía mostrando el correcto funcionamiento del mismo.

```
D:\UVG\Segundo Año\Primer semestre\Estructura Datos\HDT 1\Jar>java -jar HDT1AlejandroMarco.jar
```

```
MENU PRINCIPAL:
1) Switch encender/apagar Radio
2) Cambiar frecuencia AM/FM
3) Avanzar en el dial de las emisoras
4) Guardar emisora
5) Seleccionar emisora
6) Salir del programa

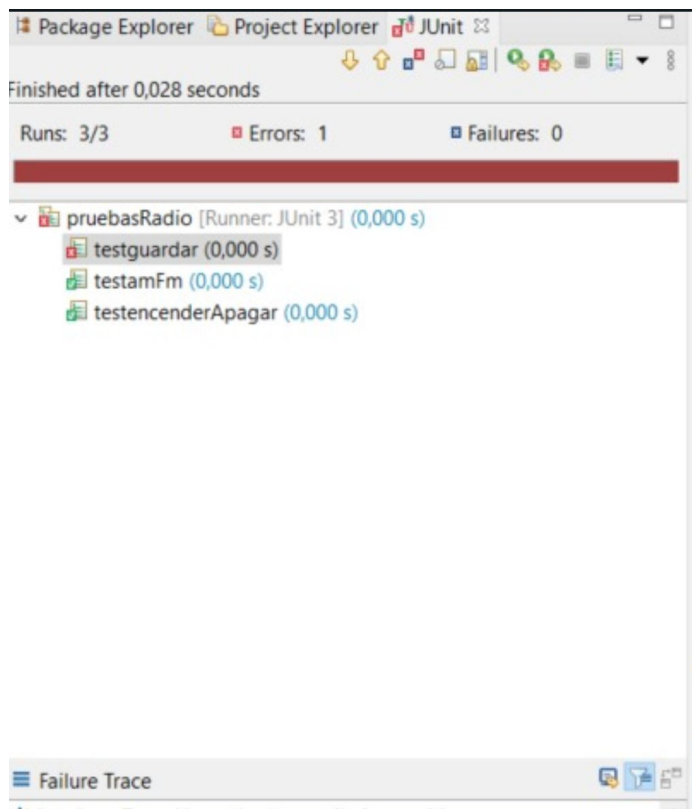
Ingrese numero de opcion:
```

Así mismo, se desarrolló un UML con el correcto uso de las flechas

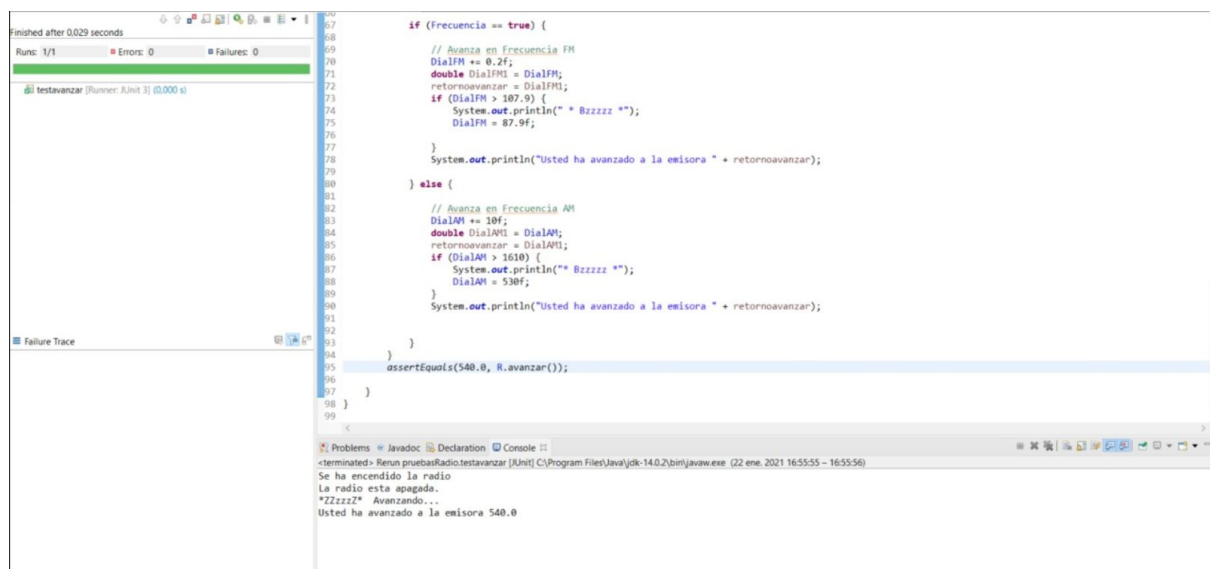


## JUnit test

Se adjunta evidencia de las pruebas realizadas de JUnit en el IDE de Eclipse.

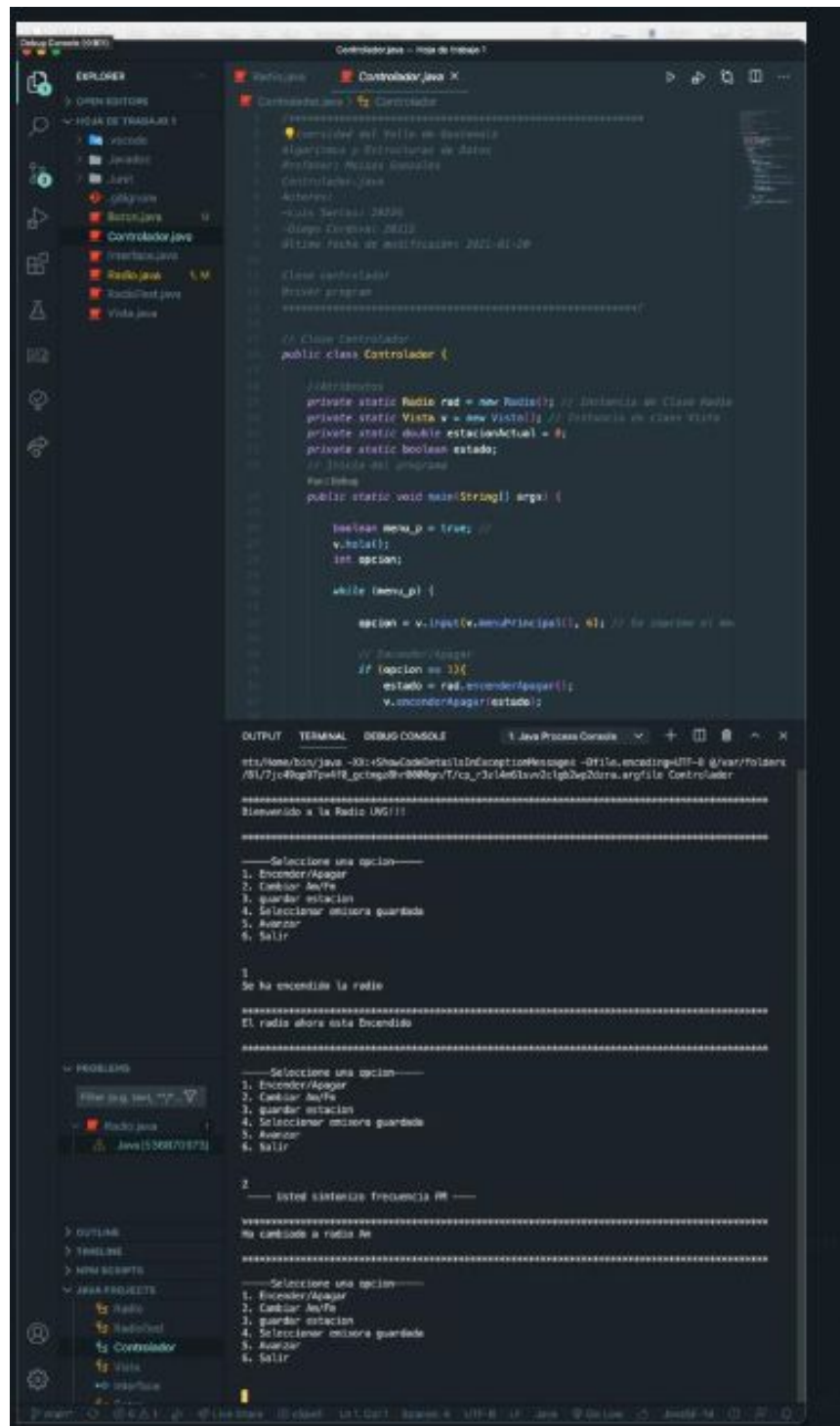


Se obtuvo un error al correrlo por primera vez.



Posteriormente se solucionó dicho error, resultando en 3 pruebas correctas.

### 1. Equipo Luis Diego Santos y Diego Cordova



## 2. Equipo Paola de León y Paola Contreras:

```
MENU PRINCIPAL:
1. Encender radio
2. Seleccionar una frecuencia de radio
3. Cambiar estacion
4. Guardar la estacion de radio actual
5. Acceder a una estacion de radio previamente guardada
6. Apagar radio
7. Salir

Ingrese la opcion a la cual desea acceder:
4

Has ingresado a la opción 4: Guardar la estacion de radio actual
Ingresa el numero boton en el que deseas guardar tu estacion actual (1-12):
1
-----> Se ha guardado exitosamente la emisora 88.1 de la frecuencia FM en el boton 1

MENU PRINCIPAL:
1. Encender radio
2. Seleccionar una frecuencia de radio
3. Cambiar estacion
4. Guardar la estacion de radio actual
5. Acceder a una estacion de radio previamente guardada
6. Apagar radio
7. Salir

Ingrese la opcion a la cual desea acceder:
5

Has ingresado a la opción 5: Acceder a una estación de radio previamente guardada
Ingresa el numero boton que desees presionar (1-12):
1

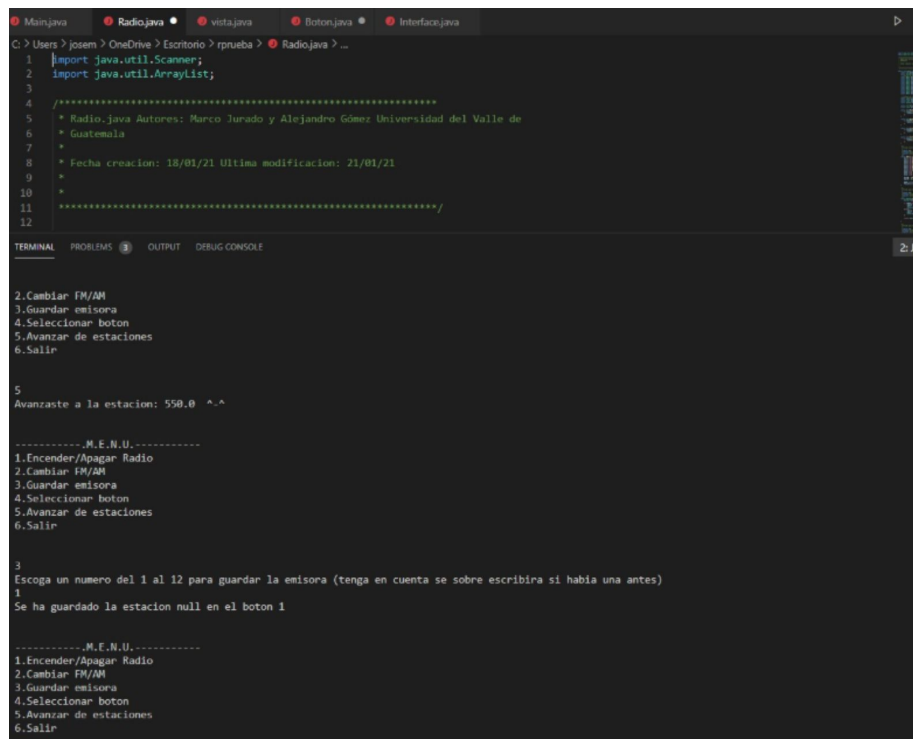
Sintonizando la emisora 88.1 en la frecuencia FM

MENU PRINCIPAL:
1. Encender radio
2. Seleccionar una frecuencia de radio
3. Cambiar estacion
4. Guardar la estacion de radio actual
5. Acceder a una estacion de radio previamente guardada
6. Apagar radio
7. Salir

Ingrese la opcion a la cual desea acceder:
6

Has ingresado a la opción 1: Apagar radio
Se ha apagado la radio
- RADIO: APAGADA
```

## 3. Equipo Jose Miguel Gonzalez y Carol Arevalo:



The screenshot shows a Java IDE with several files open: Menu.java, Radio.java, vista.java, Boton.java, and Interface.java. The main editor displays the code for Menu.java, which includes imports for Scanner and ArrayList, and a menu structure with options 1 through 6. The terminal output shows the execution of the program, including the menu display, user input for selecting an option (4), and the resulting action (guarding the station 88.1 on FM frequency 1).

```
C:\Users\josem> java -cp . Menu.java
1  import java.util.Scanner;
2  import java.util.ArrayList;
3
4  /*****
5   * Radio.java Autores: Murco Jurado y Alejandro Gómez Universidad del Valle de
6   * Guatemala
7   *
8   * Fecha creacion: 18/01/21 Ultima modificacion: 21/01/21
9   *
10  */
11  *****/
12

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

2.Cambiar FM/AM
3.Guardar emisora
4.Seleccionar boton
5.Avanzar de estaciones
6.Salir

5
Avanzaste a la estacion: 550.0 ^.^

-----M.E.N.U-----
1.Encender/Apagar Radio
2.Cambiar FM/AM
3.Guardar emisora
4.Seleccionar boton
5.Avanzar de estaciones
6.Salir

3
Escoga un numero del 1 al 12 para guardar la emisora (tenga en cuenta se sobre escribira si habia una antes)
1
Se ha guardado la estacion null en el boton 1

-----M.E.N.U-----
1.Encender/Apagar Radio
2.Cambiar FM/AM
3.Guardar emisora
4.Seleccionar boton
5.Avanzar de estaciones
6.Salir
```

\*Nota - Se tuvo que cambiar el método amFM pues algunas parejas tenían el método con el nombre de amFm. Así como Double y double.