



UNIVERSIDAD DE CARABOBO  
Facultad Experimental de Ciencias y Tecnología  
Departamento de Computación  
Fundamentos de Programación



Cod. Asignatura: TAO207

Proyecto (valor: 20%)  
Emisora de radio

### Normas de Entrega:

La entrega del proyecto se realizará durante la **última semana del semestre**, en la fecha y lugar que serán anunciados previamente por el profesor.

#### 1. Contenido

Se deberá entregar un archivo comprimido en formato **.zip** que contenga:

- Un único archivo fuente con extensión **.c**, que incluya **todo el desarrollo del proyecto**.
- Un informe cuyo contenido se describe en las páginas subsiguientes.
- Los **archivos de entrada correspondientes**.
- **No deben incluirse archivos de salida ni ejecutables generados por el compilador (gcc).**

#### 2. Codificación

- El **nombre del archivo .c** y el de los archivos de entrada debe coincidir exactamente con lo indicado en el enunciado del problema.
- El archivo fuente debe contener en su **cabecera**, mediante comentario de bloque, la siguiente información:
  - Fecha
  - Nombre y Apellido
  - Cédula
  - Número de Sección

#### 3. Defensa oral (Obligatoria)

- Cada equipo deberá realizar una **defensa oral del proyecto**, en la fecha y horario establecidos por el profesor.
- Durante la defensa, los integrantes deberán explicar el funcionamiento del programa, las decisiones de diseño tomadas, y responder preguntas sobre su implementación.

#### 4. Informe

Se debe entregar un **informe en formato digital**, incluido en el archivo comprimido, que contenga los siguientes apartados:

##### 1. Portada

- Título del proyecto
- Nombres y apellidos de los integrantes
- Cédulas de identidad
- Sección
- Fecha de entrega

##### 2. Descripción del problema

- Explicación general del enunciado y objetivos
- Contexto y propósito del software desarrollado

##### 3. Análisis de la solución

- Cómo se organizan y almacenan los datos
- Justificación de las decisiones tomadas

##### 4. Estructuras de datos

- Definición de los structs utilizados
- Arreglos u otras estructuras aplicadas
- Diagrama o representación opcional

##### 5. Algoritmos implementados

- Explicación en lenguaje pseudoformal
- Puede usarse pseudocódigo, texto estructurado o diagramas

##### 6. Pruebas realizadas

- Descripción de los casos de prueba
- Breve resumen del comportamiento observado

##### 7. Conclusiones

- Reflexión sobre la experiencia de desarrollo
- Dificultades encontradas y soluciones
- Posibles mejoras

El informe debe estar redactado en **letra Arial o Calibri (Word) o Roman (LaTeX)**, tamaño **12 puntos**, con márgenes de **3 cm a la izquierda y 2 cm en los demás bordes**.

#### 5. Importante

- El proyecto debe ser realizado en **equipos de 2 (dos) estudiantes** de la misma sección de laboratorio, **sin excepción**. No se aceptarán cambios de equipos una vez conformados.

- El programa debe desarrollarse **exclusivamente en lenguaje ANSI C**, utilizando las librerías estándar: *stdio.h*, *ctype.h*, *string.h*, *stdlib.h*, *assert.h*, *time.h* y *math.h*.
- La solución debe aplicar **únicamente los contenidos vistos en clase**, tales como: Condicionales, Ciclos, Acciones nominadas, Archivos, Arreglos y Registros.
- Es obligatorio el uso de **acciones nominadas (subrutinas con nombres significativos) y parámetros** en el código.
- El código debe estar **bien documentado y estructurado**, con comentarios, indentación clara y uso de nombres significativos. Considere seguir algún formato de documentación compatible con herramientas ampliamente utilizadas tales como **doxygen** (<https://www.doxygen.nl/>), o en su lugar, considere utilizar **sphinx** (<https://www.sphinx-doc.org/en/master/>).
- El informe será evaluado también en cuanto a su **redacción y ortografía**.
- Los **proyectos entregados después de la fecha tope** tendrán una penalización de **5 puntos por cada día de retraso**.
- Los **criterios de evaluación** serán publicados oportunamente por el profesor.

**No seguir las indicaciones antes mencionadas repercutirá en la nota final de su entrega.**

## 1. Enunciado

La emisora radial Éxitos FM los ha contratado para resolver una problemática de planificación de la programación. Actualmente, las emisoras se organizan a través de un software que planifica la programación diaria, permitiendo al operador asignar bloques en todos los horarios. Los operadores montan la "grilla" o "parrilla de programación" de forma manual, situando shows y canciones con mayor popularidad en los horarios estelares y el resto del tiempo los de menor rating.

Los **horarios estelares** son de **07:00 a 09:00**, de **12:00 a 14:00** y de **18:00 a 19:00**, ya que es cuando la mayoría de los oyentes están en el tráfico y eligen qué escuchar. Siempre habrá al menos 3 shows en la emisora que pueden cubrir estos horarios.

Debido a que el personal actual ha realizado un trabajo deficiente en la creación de la grilla diaria, el gerente de la emisora requiere sus servicios para crear un **software "inteligente"**. Este software debe permitir al operador ingresar la popularidad de cada canción y/o show radial, y **ser el software quien indique la mejor organización para aumentar el rating de la emisora**.

**Algunos aspectos a considerar son:**

- La emisora tiene **N patrocinantes**. Cada uno tiene un **número mínimo de cuñas por día**, sin importar el horario. Se debe cumplir con todos los patrocinantes, y cada uno puede tener una cantidad diferente de cuñas diarias. Cada patrocinante tiene un único modelo de cuña con una duración diferente a las de las demás empresas.
- Puede haber un número de canciones en cola para el día que no abarquen las 24 horas completas, incluso considerando las cuñas y shows radiales. Por lo tanto, las **canciones pueden repetirse**, pero no más de una vez cada 4 horas, para evitar saturación. Las canciones altamente populares tienden a repetirse en la programación.
- Cada show radial tiene una **duración variable en cada segmento**, y debe haber publicidad y reproducción de canciones entre segmentos. **Nunca se puede repetir el mismo show radial en el día**.

Los datos deben ser ingresados **únicamente a través de archivos**, y la salida debe ser **tanto en pantalla como en archivo**.

## 2. Actividades a realizar

Se solicita que, basándose en estos parámetros de exigencia, el software genere **una grilla para todo el día** (desde las 00:00:00 hasta las 23:59:59) **de forma automática**. El operador podrá **consultar por pantalla la programación para un momento determinado del día**, saber si una canción se reproducirá y a qué hora. La **programación completa del día** se debe mostrar

por pantalla y guardarse en un archivo, organizando los elementos (shows, canciones y cuñas publicitarias) en **bloques**.

Además, el software debe permitir la creación de una **grilla para toda la semana** (7 días). Cada día de la semana debe ser tratado como un bloque independiente, y el operador podrá consultar la programación para **cualquier día y hora específicos**. La programación de toda la semana debe guardarse en un archivo y poder ser consultada por día y hora.

Para garantizar la **calidad de la transmisión**, debe existir **1 segundo exacto de separación** entre cada elemento a reproducir (publicidad, show o canción). Esto significa que no puede haber más de **1 segundo sin que algo se esté reproduciendo** en la radio, asegurando así que siempre haya sonido.

### 3. Formato de entrada y salida

Se utilizarán **3 archivos de entrada**: *canciones.in*, *publicidad.in* y *shows.in*. Para la salida, se espera emitir una colección de archivos especificados en la sección 3.4.

#### 3.1. Archivo de entrada: canciones.in

La primera línea contendrá un entero  $1 \leq C \leq 1000$  que representa la cantidad de canciones que posee la emisora para ese día.

Las líneas siguientes se estructurarán como: **Nom, Min, Seg, y Punt**.

- **Nom** representa cada nombre de la canción (No mayor a 50 caracteres, nunca tendrán números).
- **Min** es un entero que representa la duración en minutos.
- **Seg** es un entero que complementa a Min en la duración de la canción, representando los segundos.
- **Punt** representa la puntuación ( $1 \leq \text{Punt} \leq 100$ , donde 1 es pésimo y 100 excelente).

Todos los parámetros están separados por un espacio en blanco (el nombre de la canción puede contener múltiples espacios en blanco).

**Ejemplo:** Se cargan 10 canciones de la siguiente manera:

```
10
Let it be 3 26 90
Complicated 3 38 75
Persiana Americana 4 55 80
Dar es dar 3 39 50
Bienvenidos al tren 3 14 6
Six Degrees of Inner Turbulence 42 04 99
You suffer 0 1 1
Cocoon 4 34 26
```

### 3.2. Archivo de entrada: publicidad.in

Se debe leer hasta el fin del archivo los siguientes parámetros: **Empresa, Segundos y Veces**, donde:

- **Empresa** define el nombre del patrocinante (No mayor a 30 caracteres).
- **Segundos** define la duración en segundos de cada anuncio de esta empresa, nunca será mayor a 60 segundos.
- **Veces** define la cantidad de repeticiones que debe tener la cuña durante el día.

**Ejemplo:** Se cargan 3 patrocinantes de la siguiente manera:

```
Zapatos comod0s de piedra C.A 25 31
Pollos de goma S.R.L 15 7
Empresas La Arepa Rellena 59 158
```

### 3.3. Archivo de entrada: shows.in

Este archivo está definido por **N shows radiales**, donde  $3 \leq N \leq 15$ .

Es importante recordar que, aunque un show radial ocupe un horario específico, el mismo tiene una cantidad de segmentos que son las que definen su duración.

Se deben leer en las siguientes N líneas los parámetros: **Nombre Minutos Segundos Segmentos Preferencia**.

- **Nombre** el nombre del show (No mayor a 100 caracteres).
- **Minutos** define la duración en minutos de cada segmento del show.
- **Segundos** define la duración en segundos que complementa a Minutos en la definición de la duración del segmento, nunca será mayor a 60 segundos.
- **Segmentos** define la cantidad de oportunidades que salen al aire durante el horario.
- **Preferencia** define en una escala de valores enteros del 1 al 10 el nivel de rating que tiene el show, donde 1 es bajo y 10 es alto.

**Ejemplo:** Se cargan 5 shows de la siguiente manera:

```
5
Aprendiendo a escuchar 5 30 4 5
El anuario de los mejores 1 20 10 9
Ciencia Critica 10 00 4 8
El programa con el nombre mas largo y
aburrido en la historia de la radio 30
30 2 1
Partidos de damas chinas 60 05 1 3
```

### 3.4. Archivos de salida: grilla\_{dia}.out

El software deberá generar una **grilla de programación para cada día de la semana**, comprendida entre las **00:05:00 y las 23:59:59**, utilizando los mismos archivos de entrada (canciones.in, publicidad.in, shows.in) para todos los días.

Se deben generar **7 archivos de salida**, uno por cada día, bajo el formato:

- grilla\_lunes.out
- grilla\_martes.out
- grilla\_miercoles.out
- grilla\_jueves.out
- grilla\_viernes.out
- grilla\_sabado.out
- grilla\_domingo.out

Cada archivo debe mostrar el listado organizado de la programación del día correspondiente, considerando la duración exacta de cada elemento.

La información se debe presentar en el formato estricto de **HH MM SS T Nombre**, donde:

- **HH** representa las horas (dos dígitos)
- **MM** representa los minutos (dos dígitos)
- **SS** representa los segundos (dos dígitos)
- **T** es una letra mayúscula que indica el tipo de contenido:
  - **S** para Show
  - **P** para Publicidad
  - **C** para Canción
- **Nombre** es el nombre del show, canción o cuña publicitaria según corresponda

Debe existir **una separación de exactamente 1 segundo** entre cada elemento a reproducir. Esto significa que **no puede haber más de un segundo sin sonido en la emisora**, y ese segundo está implícito como el tiempo de transición entre el final de un elemento y el inicio del siguiente.

Cada grilla diaria debe ser única. El software deberá aplicar mecanismos para **evitar repeticiones exactas de contenido entre los diferentes días de la semana**, variando el orden de reproducción, las canciones seleccionadas, o la combinación de elementos.

**Ejemplo** (muestra solo una parte del archivo, el software debe colocar toda la planificación del día):

```
00 05 00 C Let it be
00 08 27 C Bienvenidos al tren
00 11 42 C Dar es dar
00 15 22 C Six Degrees of Inner Turbulence
00 57 27 C You suffer
00 57 29 P Empresas La Arepa Rellena
00 58 29 C Complicated
01 02 08 P Zapatos comodios de piedra C.A
01 02 34 C Persiana Americana
01 07 30 S El programa con el nombre mas largo y aburrido en la historia de la radio
01 38 01 C Wont get fooled again
01 46 38 S El programa con el nombre mas largo y aburrido en la historia de la radio
.
.
.
```

Como toda estrategia bien ejecutada, cada segundo cuenta.

*"La sabiduría comienza en la reflexión. Nada es verdad, todo está permitido."*

— E. A., AC II