

# El eterno dilema del lector

Algoritmos y programación II - 2C2022

Trabajo práctico Nº1

Individual



## Introducción

---

A la hora de leer no todos tenemos nuestros gustos completamente definidos por lo que enfrentarnos a la temida pregunta ¿Cuál es tu género o libro favorito? Puede ser una verdadera pesadilla. Cansados de no saber responder esta pregunta los ayudantes de algoritmos 2 decidieron pedirle ayuda a los estudiantes de dicha materia.

## Enunciado

---

Para poder ayudar a salir a los ayudantes de este dilema y aprovechando que es una materia de programación se deberá hacer una aplicación en la que se ayudará a gestionar los libros leídos.

Primero se cargaron todos los libros leídos en un archivo al que se nombró **libros.csv** y se le designó el siguiente formato:

**título,género,puntaje**

Por ejemplo:

```
Salvajes,R,45
Narnia,A,5
Un mundo complejo,C,0
...
```

A la hora de modificar y leer el archivo es importante tener en cuenta que el título de un libro **puede tener espacios**, el puntaje asociado a cada libro es un **número positivo que va desde 0 hasta 100<sup>1</sup>** y los géneros respetan la siguiente nomenclatura:

Género	Representación
Aventura	A
Ciencia Ficción	C
Didáctica	D
Policíaca	P
Romance	R
Terror	T

Cualquier otro género que no esté en la tabla antes mencionada no estará en el archivo.

**NOTA:** no hay límite para la cantidad de libros, se deberá leer el archivo hasta el final y eso determinará cuantos existen<sup>2</sup>.

---

<sup>1</sup> Ambos inclusive.

<sup>2</sup> Para esto se utilizará lo visto sobre memoria dinámica.

Una vez leído el archivo armar un programa que cuente con el siguiente menú:

1. Listar libros leídos.
2. Agregar libro.
3. Editar puntaje de un libro por título.
4. Mostrar libro favorito.
5. Mostrar los 3 libros con menor puntaje.
6. Mostrar género más leído.
7. Mostrar género favorito.
8. Guardar y salir.

### *Listar libros leídos*

Se deberán listar todos los libros con su género en forma completa<sup>3</sup> y puntaje.

### *Agregar libro*

Se deberá verificar que no exista el libro en el vector, si existe se le deberá notificar al usuario que ese libro ya está agregado.

Si el libro no fue incluido aún se le deberá pedir el género y puntaje de este, validando que todos los campos ingresados sean válidos.

### *Editar puntaje de un libro por título*

Se deberá solicitar el título al usuario y validar que el mismo este entre los libros del archivo. Si el libro existe se le solicitará el nuevo puntaje, se validará que este en rango y se pedirá que lo reingrese en caso contrario.

En caso de que no exista un libro con el título ingresado se le deberá mostrar un error acorde al usuario.

### *Mostrar libro favorito*

Se deberá mostrar el título del libro con el mayor puntaje. Si hay un empate se deberán mostrar todos los que compartan este puntaje.

### *Mostrar los 3 libros con menor puntaje*

Se deberá mostrar los títulos de los 3 libros con los menores puntajes. Si hay un empate en el tercer puesto se deberá mostrar el título que aparecería primero si fuera en orden alfabético, es decir, si "Azul" y "Rusty" son libros válidos para ocupar el puesto tres solo se debería mostrar Azul.

### *Mostrar género más leído*

Se deberá mostrar el género que aparezca más veces. Si hay un empate se deberán mostrar todos los que tengan la máxima cantidad de apariciones.

### *Mostrar género favorito*

Se deberá mostrar el género con el mayor promedio. Si hay un empate se deberán mostrar todos los que compartan este promedio.

---

<sup>3</sup> Se debe poner la palabra completa en lugar del carácter.

## Guardar y salir

Al terminar de usar el programa se deberá actualizar<sup>4</sup> el archivo con los valores correspondientes.

## Aclaraciones

---

- ✚ El trabajo es de carácter **individual**.
- ✚ El trabajo no cuenta con reentrega.
- ✚ El **vector** de libros debe ser **dinámico**.
- ✚ Los archivos están bien formados.
- ✚ No recorrer más de una vez los archivos.
- ✚ Se deben validar los datos ingresados por el usuario y mostrar errores que sean fáciles de comprender por cualquier persona.
- ✚ No se deben subir archivos de configuración de los IDEs (**solo subir .cpp y .h**).
- ✚ El trabajo debe compilar con los flags -Wall -Werror -Wconversion.
- ✚ No se permiten usar bibliotecas de templates como por ejemplo STL.

## ¿Qué se evaluará?

---

- ✚ Compilación
- ✚ Funcionalidad
- ✚ Manejo de archivos
- ✚ Usabilidad (que tan sencillo es de utilizar para un usuario)
- ✚ Eficiencia espacial
- ✚ Eficiencia temporal
- ✚ Correcto uso de memoria dinámica.
- ✚ Buenas prácticas de programación (nombres descriptivos, indentación, etc.)
- ✚ Modularización.
- ✚ Precondiciones y postcondiciones.

---

<sup>4</sup> Si los valores cambiaron se reescribirá el archivo manteniendo el formato y nombre de este y notificando si surge algún error a la hora de abrir o escribir en el mismo.

## *Normas de entrega*

---

Se deberá subir al campus un único archivo comprimido (.zip o .rar) en la sección TPs. Este archivo deberá tener un nombre formado de la siguiente manera:

**Padron\_Apellido\_TP1**

Por ejemplo:

106204\_Perez\_TP1.zip

Deberá contener solo los archivos fuente. Es decir, solo .cpp y .h. NO subir los archivos de configuración de sus IDEs. (por ejemplo: CMakeList y cmake-build para Clion, .vscode para VisualStudioCode).

La fecha de **entrega** vence el **jueves 15/09/2022** a las 23.55hs.

**Puntaje:** 40 puntos.