

## TÉCNICAS DE PROGRAMACIÓN

### CUARTA PRACTICA CALIFICADA SEMESTRE ACADÉMICO 2023-1

Horario: Todos

Duración: 110 minutos  
Elaborado por los profesores del curso.

#### ADVERTENCIAS:

- Todo dispositivo electrónico (teléfono, tableta, computadora u otro) deberá permanecer apagado durante la evaluación en su mochila.
- Coloque todo aquello que no sean útiles de uso autorizado durante la evaluación en la parte delantera del aula, por ejemplo, mochila, maletín, cartera o similar, y procure que contenga todas sus propiedades. La apropiada identificación de las pertenencias es su responsabilidad.
- Si se detecta omisión a los dos puntos anteriores, la evaluación será considerada nula y podrá conllevar el inicio de un procedimiento disciplinario en determinados casos.
- Es su responsabilidad tomar las precauciones necesarias para no requerir la utilización de servicios higiénicos: durante la evaluación, no podrá acceder a ellos, de tener alguna emergencia comunicárselo a su jefe de práctica.
- Quienes deseen retirarse del aula y dar por concluida su evaluación no lo podrán hacer dentro de la primera mitad del tiempo de duración destinado a ella.

#### INDICACIONES:

- No se pueden usar apuntes de clase ni calculadoras.
- LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN Estricto DISEÑO DESCENDENTE, por lo que NO SE CALIFICARÁN aquellos módulos que son llamados por otros que estén incompletos. Cada módulo no debe sobrepasar las 30 líneas de código aproximadamente.
- NO SE PUEDEN EMPLEAR ARCHIVOS DE DATOS AUXILIARES NI VARIABLES GLOBALES. NO podrá implementar funciones en el archivo main.cpp, las funciones se deberán implementar en archivos independientes (.h/.hpp y .cpp).
- No se calificará el código puesto como comentario.
- En la calificación se tomará en cuenta el buen uso de los nombres de los identificadores, y el eficaz uso de comentarios.
- **DEBE COLOCAR SU NOMBRE Y CÓDIGO EN CADA UNO DE LOS ARCHIVOS .h Y .cpp QUE EMPLEE EN SUS PROYECTOS, DE LO CONTRARIO SE LE DESCOTARÁ 0.5 PUNTOS POR CADA OMISIÓN. NO SE HARÁN EXCEPCIONES.**
- **TODA OPERACIÓN DE BÚSQUEDA DEBE REALIZARSE EN FUNCIONES INDEPENDIENTES Y DEBE CONSIDERAR QUE EL DATO BUSCADO NO SE ENCUENTRE.**

#### INDICACIONES INICIALES

Siga estrictamente las indicaciones que a continuación se detallan:

- En la unidad de trabajo indicada por los Jefes de práctica (si trabaja en otra unidad, no se calificará el laboratorio y se le asignará como nota cero) Cree allí una carpeta con el nombre **“CO\_PA\_PN\_2023\_1\_Practica04”** donde **CO** indica: Código del alumno, **PA** indica: Primer Apellido del alumno y **PN** primer nombre (de no colocar este requerimiento **se le descontarán 3 puntos de la nota final**). **Allí colocará el proyecto solicitado en la pregunta**

#### DEBE LEER TODA LA PRUEBA ANTES DE EMPEZAR A DESARROLLAR EL PROGRAMA

La biblioteca de una universidad desea contar con una aplicación que le permita llevar un control de sus libros y usuarios. Para ello cuenta con los siguientes archivos:

##### Libros.csv

```
NF53199,Suspense,Emilio Gonzales Aguirre,25,167.64
UW23132,House of Pomegranates,David Perez Antezano,44,122.89
... ..
```

En cada línea de este archivo se muestra la información de los libros. Primero aparece el **código del libro** y le sigue su **título**, **autor**, **cantidad de volúmenes** que posee en su almacén y su **precio**.

##### Usuarios.csv

```
81082011,Arroyo Gordillo Maria Heli,A,45.78
15129661,Wong Martel Madeleine,D,88.93
65071478,Mejia Diaz Blanca Milagros,V,15.23
... ..
```

En cada línea de este archivo se muestra el **DNI** del usuario, su **nombre**, su **categoría** (**A**: alumno, **D**: docente, **V**: visitante), finalmente viene un valor que define su calificación como **usuario** (valor de punto flotante entre 0 y 99) que define que tan buen cliente es.

**SolicitudesDePrestamo.txt**

45834351	SK15788
49516360	BB35881
45834351	BB35881
...	...

El archivo contiene de préstamos de libros que recibe la biblioteca en un día determinado. En cada línea aparece el pedido de un libro por un usuario, primero aparece el **DNI** del usuario seguido por el código del libro que solicita. Un usuario puede solicitar varios libros en distintos momentos por lo que puede aparecer varias veces en el archivo, de igual manera, como la biblioteca posee varios volúmenes de un libro determinado, un libro puede aparecer varias veces en el archivo.

La biblioteca tiene por política atender los pedidos por categoría, primero a los alumnos, luego a los docentes y finalmente a los visitantes. Además, dentro de una misma categoría atiende primero a los que tienen mayor calificación. Por otro lado, solo presta un máximo de 4 libros por alumno, 6 por profesor y 2 por visitante.

**Los archivos no están ordenados.**

---

La finalidad de esta práctica es la de reforzar los conceptos de estructuras.

**Cree un proyecto en NetBeans con el nombre: “Estructuras\_Arch\_Binarios\_2023\_1\_Pract04” (de no respetar este nombre se le descontarán dos puntos de su nota final – NO SE HARÁN EXCEPCIONES)** y en él desarrolle el programa que resuelva el problema que se describe a continuación. El proyecto debe estar dentro de la carpeta creada previamente.

La aplicación deberá realizar las siguientes tareas:

1. Definir las siguientes estructuras:  
**UsuarioConLibro:** debe poder contener el DNI, categoría y calificación de un usuario.  
**LibroPrestado:** debe poder contener el código y el precio de un libro.  
**Libros:** debe poder contener todos los datos de un libro del archivo **Libros.csv**, además, debe contener un campo **UsuariosQueTieneUnEjemplar** (arreglo dinámico del tipo **UsuarioConLibro**) y un campo con un valor entero que cuente cuántos libros ha prestado, el primero debe inicializarse con nullptr y el segundo con cero.  
**Usuarios:** debe poder contener todos los datos de un usuario del archivo **Usuarios.csv**, además debe contener un campo **LibrosEntregados** (arreglo dinámico del tipo **LibroPrestado**) y un campo con un valor entero que indique cuantos libros se le ha prestado, el primero debe inicializarse con nullptr y el segundo con cero. La categoría se debe considerar como un solo carácter, no como una cadena de caracteres.  
**Solicitudes:** debe poder contener todos los datos de una solicitud del archivo **SolicitudesDePrestamo.txt**  
**En esta parte todas las cadenas deben trabajarse de manera dinámica.**
2. Leer los datos de los tres archivos y colocar la información en arreglo de estructuras que contengan toda la información. Debe modular la lectura de cada archivo. No debe modificar aquí los campos adicionales.
3. Imprimir, en un solo archivo, los datos de los arreglos. Los datos deben presentarse bien tabulados y alineados, con encabezados adecuados encima de cada columna y un título que indique qué se está imprimiendo. Debe modular la impresión de cada arreglo.
4. Ordenar el arreglo de usuarios de modo que pueda atender los pedidos según la política de la biblioteca.
5. Atender las solicitudes de préstamo, según la política de la biblioteca. Por cada solicitud atendida debe colocar los datos del usuario al que se le entrega el libro en el arreglo dinámico del libro. De igual manera agregar los datos del libro al usuario que lo recibe. Para que una solicitud sea atendida, debe haber libros en el almacén, además, el usuario no debe haber superado su límite de libros prestados. Si no se cumple con los requisitos, debe descartar el pedido.
6. Volver a imprimir el reporte de la tarea 3, pero en otro archivo.

7. Cree un archivo binario denominado ***Libros.bin*** con la información del archivo Libros.csv.
8. Imprima el contenido del archivo ***Libros.bin*** en un archivo denominado “PruebaDeArchivo.txt”, los datos deben estar bien alineados y con encabezados que indiquen la naturaleza de los datos.

**Las tareas que no se desarrollen en el orden solicitado no se calificarán.**

**Al finalizar la práctica, comprima la carpeta que contiene el proyecto empleando el programa Zip que viene por defecto en el Windows, no se aceptarán los trabajos compactados con otros programas como RAR, WinRAR, 7zip o similares y súbalo a la tarea programada en PAIDEIA para esta práctica.**

CRITERIOS DE CALIFICACIÓN:
<ol style="list-style-type: none"><li>1. Si el programa no muestra los resultados o los muestren y no sean correctos, no podrán tener más del 75% de la nota.</li><li>2. Se descontará 15% de la nota si el programa define variables con nombres que no tengan sentido.</li><li>3.. Se descontará 15% de la nota si no se colocan comentarios relevantes.</li><li>4. No se calificará el código puesto como comentario.</li><li>5. No se calificarán aquellas funciones implementadas en el archivo main.cpp</li></ol>



San Miguel, 19 de junio del 2023