

TÉCNICAS DE PROGRAMACIÓN
NOVENO LABORATORIO CALIFICADO
SEMESTRE ACADÉMICO 2025-2

Horarios: 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.
- Está prohibido el acceso a Internet y a correo electrónico hasta que lo indiquen los jefes de práctica, tampoco podrá emplear dispositivos USB.
- Si se detecta omisión al punto anterior, la evaluación será considerada nula y podrá conllevar el inicio de un procedimiento disciplinario en determinados casos.
- 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 y .cpp).
- 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 EL ARCHIVO main.cpp DE SU PROYECTO, DE LO CONTRARIO SE LE DESCOTARÁ 0.5 PUNTOS EN SU NOTA FINAL. NO SE HARÁN EXCEPCIONES.**
- **DEBE COLOCAR UN COMENTARIO AL INICIO DEL ARCHIVO main.cpp CON UNA DESCRIPCIÓN DE LO QUE HACE PROGRAMA. ESTA DESCRIPCIÓN NO DEBE SER GENÉRICA, DEBE SER ESPECÍFICA DE LO QUE HARÁ EL PROGRAMA. SE LE DESCOTARÁ 0.5 PUNTOS EN SU NOTA FINAL SI NO SE COLOCA ESTE COMENTARIO O LO QUE SE EXPRESE EN ÉL NO SEA ESPECÍFICO. NO SE HARÁN EXCEPCIONES.**
- **TODA OPERACIÓN DE BÚSQUEDA DEBE REALIZARSE EN FUNCIONES INDEPENDIENTES Y DEBE CONSIDERAR QUE EL DATO BUSCADO NO SE ENCUENTRE.**
- **SOLO PUEDE TENER ABIERTO EL PROYECTO INDICADO EN ESTE LABORATORIO, NINGÚN OTRO.**
- **NO PODRÁ EMPLEAR LAS BIBLIOTECAS <stdio.h>, <csdtdio> <string> ni <string.h>. NO CONFUNDIR CON <cstring> QUE SÍ LA PUEDE EMPLEAR.**

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 su laboratorio y se le asignará como nota cero**), cree allí una carpeta con el nombre “Lab09_2025_2_CO_PA_PN” donde **CO** indica: Código del alumno, **PA** indica: Primer Apellido del alumno y **PN** primer nombre (de no colocar este requerimiento se le descontará 3 puntos de la nota final). **Allí colocará el proyecto solicitado en la prueba. NO SE HARÁN EXCEPCIONES SI NO ACATA ESTAS INDICACIONES.**

La finalidad de este laboratorio es la de reforzar los conceptos de estructuras autoreferenciadas.

(20 puntos) Cree un proyecto en CLion con el nombre: “Lab09_2025_2_ListasLigadas” (**de no respetar este nombre se le descontarán 2 puntos de su nota final – NO SE HARÁN EXCEPCIONES**) y en él desarrolle el programa que resuelva el problema que se describe a continuación. En la carpeta de su proyecto cree una carpeta denominada “Bibliotecas” y allí coloque la biblioteca de funciones que necesite (archivo .h y .cpp). Dentro de la carpeta “cmake-build-debug” cree las carpetas “ArchivosDeDatos” y “ArchivosDeReporte” y allí respectivamente coloque los archivos de datos que se le proporcionará y el archivo del reporte solicitado en la prueba.

**DEBE LEER TODA LA PRUEBA ANTES DE EMPEZAR A DESARROLLAR EL PROGRAMA
INCLUYENDO LAS INDICACIONES Y ADVERTENCIAS**

Una plataforma de Streaming lleva un registro de los canales afiliados a la plataforma, la información de los canales se encuentra almacenada en diferentes archivos, algunos de ellos se muestran a continuación:

Categorias.csv

```
CW154A34,marbles on stream,Casual conversations; reactions; and hangouts without a main game.
OZ151I97,haunted room 205,5v5 MOBA; ranked climbs; pro play analysis; and coaching.
...
```

Según lo que se ve en el archivo, en cada línea aparece una categoría registrada, se registra el código, nombre y la descripción de la categoría.

StreamsReproducidos.csv

```
VD139J89,Lord_Rebun,1.389,1:51:10
NN179A87,riotgames,1.477,1:47:25
...
```

Según lo que se puede ver en cada línea del archivo se ha colocado la información de un Stream. Primero se ve el código de la categoría asignada al Stream, luego vienen el nombre del canal donde se reprodujo, el drop-off y la duración del Stream.

CONSIDERACIONES IMPORTANTES A TENER EN CUENTA PARA EL PROYECTO:

En el proyecto deberá definir, crear y llenar una lista ligada ordenada con la información de las categorías de los Streams. La lista ligada contará con las siguientes estructuras (2 puntos):

Estructura Categoría: la cual debe contener los siguientes campos: 1) denominado **codigo** cadena dinámica de tipo **char***, 2) denominado **nombre** cadena dinámica de tipo **char***, 3) denominado **descripcion** cadena dinámica, 4) denominado **canalConDuracionMaxima** cadena dinámica de tipo **char***, 5) denominado **duracionMaxima** de tipo **int**, 6) denominado **duracionTotal** de tipo **int**, 7) denominado **arrDropOff** arreglo estático de tipo **double** con un máximo de 20 elementos, 8) denominado **cantidadDeReproducciones** de tipo **int** y 9) denominado **promedioDropOff** de tipo **double**.

Estructura Nodo: la cual debe contener los siguientes campos: 1) denominado **categoria** de tipo **struct Categoria** y 2) denominado **siguiente** de tipo **struct Nodo ***.

La función main debe estar compuesta por las siguientes tareas, **las cuales debe desarrollarlas en el orden que se indican y cada una en una función independiente.**

- (4 puntos) Leer los datos del archivo “**Categorias.csv**”, y con ellos llenar una lista ligada ordenada de manera ascendente por el código de la categoría. Los campos que no se llenen deben ser correctamente inicializados.
- (2 puntos) Verificar, utilizando el modelo indicado más adelante en el reporte (**zona roja**) la correcta asignación de los datos en el arreglo. Al archivo colóquelo por nombre “**ReporteDePruebaDeCategorias.txt**”.
- (4 puntos) Leer los datos del archivo “**StreamsReproducidos.csv**”, y con ellos registrar los datos de los campos correspondientes, el campo **promedioDropOff** no debe ser modificado aquí.
- (2 puntos) Calcule los promedios de los drop-off de cada categoría.
- (1.5 puntos) Emitir el reporte completo que se describe más adelante con el nombre de “**ReporteDeCategoriasCompleto.txt**”.
- (4 puntos) Elimine de la lista ligada todos los nodos que tienen un drop-off promedio menor a 3.5.
- (0.5 punto) Vuelva a emitir el mismo reporte con el nombre “**ReporteDeCategoriasEliminados.txt**”.

El reporte solicitado debe ser similar al siguiente:

Archivo: **ReportePorCategorias.txt**

```

=====
REPORTE POR CATEGORIAS
=====
Codigo:      CW1089
Nombre:      marbles on stream
Descripcion: Casual conversations; reactions; and hangouts without a main game.
-----
INFORMACION DE LAS REPRODUCCIONES:
DURACION MAXIMA:
NOMBRE DEL CANAL: auronplay
DURACION:    00:20:15
-----
LISTA DE DROP-OFF:  2.78  5.02  3.88  ...
PROMEDIO DE DROP-OFF:      3.27
TIEMPO DE DURACION TOTAL: 02:10:15
=====
Codigo: OZ151I97
Nombre: haunted room 205
...

```

CONSIDERACIONES IMPORTANTES A TENER EN CUENTA PARA LOS REPORTES:

- En este laboratorio se tomará muy en cuenta en la nota: la correcta lectura de los datos, en este sentido será obligatorio leer los números y cadenas como tal.
- Las cadenas de caracteres deben leerse de manera exacta cuando se requieran.
- **CADA ESTRUCTURA DEBE SER DEFINIDA EN UN ARCHIVO .h/.hpp INDEPENDIENTE.**
- Será parte importante de la nota el formato del reporte, éste deberá ser lo más parecido a la muestra dada. En este sentido, todos los valores deben estar correctamente alineados y formateados. No se podrá emplear el carácter de tabulación ('\\t') para la emisión del reporte.

Al finalizar el laboratorio, comprima la carpeta de su 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 programa en Paideia para este laboratorio.

ADVERTENCIAS:
Obligatoriamente debe desarrollar su proyecto bajo NetBeans en Windows, no podrá desarrollarlo empleando otro IDE ni otro sistema operativo.

CRITERIOS DE CALIFICACIÓN:
1. Si el programa entregado presenta más de tres errores de sintaxis serán calificados sobre la mitad del puntaje. 2. Si el programa no muestra los resultados o los muestren y no sean correctos, no podrán tener más del 75% de la nota. 3. Se descontará 15% de la nota si el programa define variables con nombres que no tengan sentido. Las variables deben empezar con una minúscula, se emplearán mayúsculas para separar las palabras compuestas (p. e.: baseInferior). 4. No se calificará el código puesto como comentario, ni aquellas funciones que no son invocadas en el proyecto. 5. No se calificarán aquellas funciones implementadas en el archivo main.cpp

San Miguel, 24 de noviembre del 2025