

# PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

## FACULTAD DE CIENCIAS E INGENIERÍA

### PROGRAMACIÓN 2 2da práctica (tipo b) (Segundo Semestre 2025)

Indicaciones Generales:

Duración: **110 minutos**.

- No se permite el uso de apuntes de clase, fotocopias ni material impreso.
- No se pueden emplear variables globales, estructuras (struct) ni objetos (con excepción de los elementos de **iostream**, **omanip** y **fstream**). No se puede utilizar la clase **string**. Tampoco se podrán emplear las funciones de C que gestionen memoria como **malloc**, **realloc**, **memset**, **strdup**, **strtok** o similares, igualmente no se puede emplear cualquier función contenida en las bibliotecas **stdio.h**, **cstdio** o similares y que puedan estar también definidas en otras bibliotecas. No podrá emplear plantillas en este laboratorio.
- Deberá modular correctamente el proyecto en archivos independientes. Las soluciones deberán desarrollarse bajo un estricto diseño descendente. Cada función no debe sobrepasar las 20 líneas de código aproximadamente. El archivo main.cpp solo podrá contener la función main de cada proyecto y el código contenido en él solo podrá estar conformado por tareas implementadas como funciones. En el archivo main.cpp deberá incluir un comentario en el que coloque claramente su nombre y código, de no hacerlo se le descontarán 0.5 puntos en la nota final.
- El código comentado no se calificará. De igual manera no se calificará el código de una función si esta función no es llamada en ninguna parte del proyecto o su llamado está comentado.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no muestren resultados o que estos no sean coherentes en base al 60%.
- Se tomará en cuenta en la calificación el uso de comentarios relevantes.
- Se les recuerda que, de acuerdo con el reglamento disciplinario de nuestra institución, constituye una falta grave copiar del trabajo realizado por otra persona o cometer plagio.
- No se harán excepciones ante cualquier trasgresión de las indicaciones dadas en la prueba.

Puntaje total: 20 puntos

- 
- La unidad de trabajo será t:\ (Si lo coloca en otra unidad, no se calificará su laboratorio y se le asignará como nota cero). En la unidad de trabajo t:\ colocará el(los) proyecto(s) solicitado(s).
  - Cree allí una carpeta con el nombre Lab02\_2025\_2\_CO\_PA\_PN donde: CO indica: Código del alumno, PA indica: Primer Apellido del alumno y PN indica: primer nombre. De no colocar este requerimiento se le descontarán 3 puntos de la nota final.
  - Cree un proyecto denominado PunterosMultiples\_Lab02\_2025\_2 y allí resuelva el problema que se indica a continuación. En el proyecto que desarrolle deberá crear una carpeta denominada "Bibliotecas" y allí coloque la biblioteca de funciones que necesite. 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. De no colocar esto se descontará un punto en cada proyecto.

### Cuestionario

- La finalidad principal de este laboratorio es la de reforzar los conceptos contenidos en el capítulo 3 del curso: Arreglos y punteros. En este laboratorio se trabajará con punteros a punteros y los métodos de asignación de memoria exacta y por incrementos.
- Al finalizar la práctica, comprima la carpeta dada en las indicaciones iniciales 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.

Para el diseño de su solución, considere que:

- No podrá emplear arreglos estáticos de más de una dimensión.
- No puede manipular un puntero con más de un índice.
- No puede emplear arreglos auxiliares, estáticos o dinámicos, para guardar los datos de los archivos.

Los archivos solo se pueden leer una vez.

## Descripción del caso

Se busca desarrollar un programa en C++ que permita gestionar de manera organizada la información de los streamers que participan en una plataforma de streaming. El sistema debe permitir **cargar los datos de los streamers**, incluyendo su categoría, así como **registrar los comentarios de los usuarios** realizados durante las transmisiones.

Cada comentario de usuario puede mencionar o etiquetar a uno o varios streamers, lo que permite asociar palabras clave a cada streamer y llevar un control de la cantidad de menciones recibidas. La plataforma podrá así identificar cuáles son los streamers más comentados y qué etiquetas se les asignan con mayor frecuencia.

El programa también debe ser capaz de **generar reportes organizados por categoría**, mostrando los streamers de cada grupo junto con los comentarios en los que fueron mencionados. Esto permitirá a la plataforma analizar la interacción de los usuarios con los streamers y conocer la popularidad y relevancia de los distintos contenidos de manera clara y estructurada.

El programa deberá ser capaz de:

1. Leer y procesar la información de los streamers sus categorías y comentarios, desde archivos, y almacenarla en el sistema.
2. Registrar y asociar los comentarios de los usuarios realizados durante las transmisiones, incluyendo las menciones y etiquetas asignadas a los streamers.
3. Contabilizar la cantidad total de menciones de cada streamer y asociarlas a las etiquetas correspondientes, permitiendo identificar su popularidad y el tipo de interacción que generan.
4. Generar un reporte organizado por categorías, mostrando los streamers de cada grupo junto con los comentarios en los que fueron mencionados y el número total de menciones, para que la plataforma pueda analizar la participación de los usuarios y la relevancia de los contenidos.

Este sistema permitirá a la plataforma de streaming gestionar de manera eficiente la información de los streamers y la interacción de los usuarios, facilitando el análisis de la popularidad y relevancia de cada creador de contenido.

Para ello se cuenta con los siguientes archivos de texto: **streamers.csv** (cuenta, fecha de creación, fecha de la última transmisión, tiempo total reproducido – **en milisegundos**, promedio de espectadores y cantidad de seguidores), **categorias.csv** (código, nombre y descripción) y **comentarios.csv** (código del usuario y texto).

### streamers.csv

Cuenta	Fecha Creación	Fecha Última Transmisión	Tiempo Total Reproducido (ms)	Promedio Espectadores	Seguidores	Categoría
xQcOW	06/05/2023	13/08/2025	6196161750	27716	3246298	QA1080
summit1g	07/02/2022	16/08/2025	6091677300	211845	310998	MK1092

### categorias.csv

Código	Nombre	Descripción
QA1080	Magic: The Gathering	TCG; Arena/paper play; deck tech; and events.
MK1092	Mobile Legends: Bang Bang	Mobile MOBA; ranked grind and events.

### comentarios.csv

Código	Texto
ab7f2910	Can someone please help me understand <b>[Castro_1021 loltyler1]</b> how to do trading properly?
3d1a843f	<b>[summit1g]</b> Imagine doing this on a proper Road to Glory that's just a crazy thought.

Con esta información se solicita elaborar un proyecto en CLion cuya función “main” estará compuesta por el siguiente código:

```

int main() {
    char*** categorias;
    cargarCategorias("categorias.csv",
                    categorias);

    char*** streamers;
    int** fechasPromedios;
    long long** tiempoRepSegidores;
    cargarStreamers("streamers.csv",
                   streamers,
                   fechasPromedios,
                   tiempoRepSegidores);

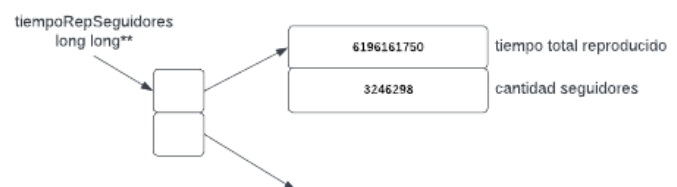
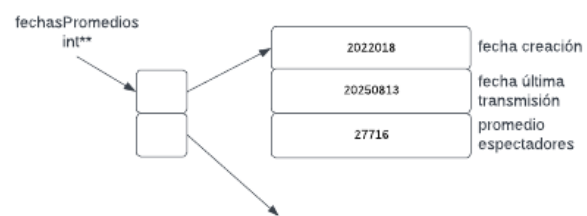
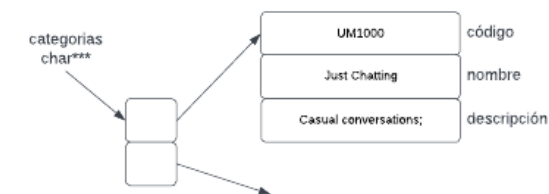
    char*** comentarios;
    char*** etiquetas;
    cargarComentarios("comentarios.csv",
                     comentarios,
                     etiquetas);

    imprimirReporte("reporte.txt",
                   categorias,
                   streamers,
                   fechasPromedios,
                   tiempoRepSegidores,
                   comentarios,
                   etiquetas);

    return 0;
}

```

Se deberán gestionar las siguientes estructuras



## Pregunta 01 (7 puntos)

Implementación de las funciones `cargarCategorias` y `cargarStreamers`

### Cargar Categorías:

Esta función deberá cargar la información contenida en el archivo **categorias.csv** sobre categorías de streamers en el arreglo `char*** categorias` usando asignación de memoria por incrementos con una capacidad incrementada en 5 por cada incremento. Cada fila en el archivo debe corresponder a un puntero de tipo `char**`.

### Cargar Streamers:

Esta función deberá cargar la información contenida en el archivo **streamers.csv** en los arreglos

`char*** streamers`, `int** fechasPromedios` y `long long** tiempoRepSeguidores` usando asignación de memoria por incrementos con una capacidad incrementada en 5 por cada incremento.

- El arreglo `char*** streamers` debe contener la cuenta y el código de la categoría.
- El arreglo `int** fechasPromedios` debe contener la fecha de creación, fecha de la última transmisión y el promedio de seguidores. Las fechas deben ser almacenadas como enteros donde 6/5/2023 debe convertirse a 20230506
- El arreglo `long long** tiempoRepSeguidores` debe contener el tiempo total reproducido y la cantidad de seguidores.

Cada fila en el archivo corresponde a un `char** streamer`, `int* fechaPromedio` y a un `long long* tiempoRepSeg`

Al implementar estas funcionalidades, el estudiante deberá desarrollar todas las funciones necesarias para garantizar su correcto funcionamiento, aplicando las mejores prácticas de modularización, claridad y reutilización de código.

## Pregunta 02 (7 puntos)

Implementar la función `cargarComentarios`. Esta función debe cargar la información contenida en el archivo `comentarios.csv` en los arreglos `char*** comentarios` y `char*** etiquetas`, para esto se deben seguir las siguientes indicaciones.

1. Todos los comentarios tendrán etiquetas en sus textos. Es decir, siempre existirá una sección en el texto que inicie con [ y termine con ]. Esta sección contendrá cuentas de streamers etiquetadas las cuales estarán separadas por un espacio en blanco en caso se haya etiquetado a más de un streamer.
2. **Los comentarios se cargarán usando asignación de memoria por incrementos.**
3. **Las etiquetas se cargarán usando asignación de memoria exacta.**
4. Cada fila en el archivo corresponde a un `char** comentario` almacenando el id del usuario y el texto del comentario.
  - a. Al leer el texto del comentario **se debe** extraer del archivo una cadena hasta encontrar [
  - b. Cargar las etiquetas en un comentario en un `char** etiqueta` de forma exacta. Para eso deberá manejar el espacio en blanco y el carácter ] que determina el final de la sección de etiquetas en el texto.
  - c. Extraer del archivo el resto del texto hasta llegar al final de la fila.
  - d. Concatenar el texto extraído antes de las etiquetas con el texto extraído luego de las etiquetas y almacenar la cadena resultante en el segundo elemento del `char** comentario` - Tenga en cuenta que al concatenar debe asegurarse que la cadena final debe tener el espacio suficiente antes de realizar la concatenación

## Pregunta 03 (6 puntos)

Implemente un reporte por categoría que liste todos los usuarios pertenecientes a una categoría incluyendo la siguiente información.

1. Cuenta
2. Fecha de creación en formato DD-MM-YYY
3. Fecha de la última transmisión en formato DD-MM-YYY
4. Tiempo total reproducido en días (debe convertir milisegundos a días)
5. Cantidad de seguidores
6. Etiquetas
  - a. Usuario que etiquetó al Streamer
  - b. Texto del comentario

Para esto debe seguir las siguientes indicaciones:

1. Iterando sobre el arreglo de categorías encontrar los streamers que pertenecen a esa categoría.
2. Imprimir el nombre de la categoría como encabezado.
3. Imprimir la información del streamer.
4. Iterando sobre el arreglo de `streamers` y los arreglos de `comentarios` y `etiquetas` encontrar los comentarios donde los streamers fueron etiquetados.
5. Imprimir los comentarios y los usuarios que etiquetaron al streamer. Si el streamer fue etiquetado más de una vez separar por un espacio en blanco.

A continuación, se muestra una referencia del reporte solicitado.

/*****League of Legends*****/					
CUENTA	FECHA CREACION	FECHA ULT. STREAM.	TIEMPO REP.	CANT. SEGUID.	ETIQUETAS
chocoTaco	09-03-2022	15-08-2025	7.18 (días)	1134153	[373221e6] 'Everyone knows Jake is the goat he's the greatest of all time.'
Yogscast	04-12-2023	16-08-2025	7.46 (días)	961860	[605d8716] 'Hahaha that is just too funny laughing out loud .'
Amaz	13-12-2021	14-08-2025	2.72 (días)	915023	
olyashaa	23-04-2022	16-08-2025	2.17 (días)	559877	
Smurfdomuca	10-04-2022	14-08-2025	3.41 (días)	267948	
erobb221	02-11-2022	13-08-2025	1.88 (días)	244207	
Broekil	16-04-2023	22-08-2025	2.16 (días)	150944	
Lapi	16-08-2023	14-08-2025	1.59 (días)	130323	
chistor_	06-06-2021	21-08-2025	2.85 (días)	89608	
Steelmage	17-11-2023	13-08-2025	3.60 (días)	60987	[3d3e843f] 'You seem to be way gone from reality my friend. '

/*****Grand Theft Auto V*****/					
CUENTA	FECHA CREACION	FECHA ULT. STREAM.	TIEMPO REP.	CANT. SEGUID.	ETIQUETAS
DrLupo	08-02-2023	16-08-2025	17.56 (días)	4115083	[62e841f3] 'Two million coins spent on fitness cards? Holy crap that's unbelievable.'
RobertoCein	17-03-2022	14-08-2025	1.73 (días)	730326	
GMHikaru	07-12-2022	20-08-2025	6.41 (días)	505361	[964cc88c] 'Please fix the kit numbers on the player jerseys. '
m0xyy	01-11-2022	17-08-2025	4.45 (días)	461534	[22c07609] 'Will you try TOTY Messi this FIFA or can you not afford him? '
RATIRL	13-08-2023	25-08-2025	7.52 (días)	423002	[373221e6] 'Jake is the goat we all know that's true .'
Becca	23-12-2022	18-08-2025	2.02 (días)	357844	
Harmi	24-06-2023	21-08-2025	1.62 (días)	333858	
Ponce	11-03-2022	17-08-2025	4.11 (días)	242969	
Calebhart42	14-04-2022	17-08-2025	2.86 (días)	177595	[48711904] 'Okay that's definitely a professional player or something like that. '
Spraggy	20-08-2023	25-08-2025	1.62 (días)	99807	
primeleague	12-06-2022	24-08-2025	1.68 (días)	73409	
Destructoid	17-07-2021	20-08-2025	1.42 (días)	48007	

Profesores del curso: Miguel Guanira  
Rony Cueva  
Erasmus Gómez

Andrés Melgar  
Eric Huiza

Pando, 12 de septiembre de 2025