

TÉCNICAS DE PROGRAMACIÓN
EXAMEN FINAL
SEMESTRE ACADÉMICO 2025-2

Horarios: Todos

Duración: 170 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.
- 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 PUEDE 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).
- 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 DESCONTARÁ 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 EL PROGRAMA. ESTA DESCRIPCIÓN NO DEBE SER GENÉRICA, DEBE SER ESPECÍFICA DE LO QUE HARÁ EL PROGRAMA. SE LE DESCONTARÁ 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.**
- CADA ESTRUCTURA DEBE DEFINIRSE EN UN ARCHIVO .h.hpp INDEPENDIENTE, DE NO RESPETAR ESTE REQUERIMIENTO, NO SE ASIGNARÁ PUNTAJE EN LA DECLARACIÓN DE LAS ESTRUCTURAS.
- **No se calificará el código puesto como comentario, ni aquellas funciones que no son invocadas en el proyecto o que su invocación esté comentada.**
- En la calificación se tomará en cuenta el buen uso de los nombres de los identificadores, y el eficaz uso de comentarios.
- **NO SE PERMITIRÁ LA LECTURA DE DATOS CARÁCTER POR CARÁCTER.**
- **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 su examen y se le asignará como nota cero**), cree allí una carpeta con el nombre “Examen2_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 2 puntos de la nota final). Allí colocará los proyectos solicitados en esta prueba. **NO SE HARÁN EXCEPCIONES SI NO ACATA ESTAS INDICACIONES.**

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

Para implementar un **nuevo plan de análisis y monitoreo** del comportamiento de los streamers de un reconocido servicio de transmisión en vivo (Twitch), se requiere desarrollar un **procesamiento integral de los comentarios** generados en diversos canales y sesiones de transmisión. El área de Gestión de Contenidos Digitales ha proporcionado varios archivos que serán utilizados en **dos aplicaciones independientes desarrolladas en lenguaje C++**.

Archivo: streamers.csv <pre>754,retromonk68,dota2,483748 214,incawizard70,apex,435046 125,quasarbard90,lol,24225 859,zenotter10,just_chatting,454486</pre>	Archivo: transmisiones.csv <pre>5002,194,03-06-2025,279 7054,323,29-07-2025,187 5669,303,01-06-2025,303 3584,854,09-04-2025,235</pre>
Archivo: comentarios.csv <pre>2232,sir_spamless,"La explicación estuvo clarísima, gracias." 1452,peru_player,Jajaja ese fail fue oro puro. 3925,chattycat,"Recién llego, ¿qué me perdi?" 5279,juancho,GG! tremendo final.</pre>	Archivo: stopwords.csv <pre>a actualmente acuerdo adelante ...</pre>

En el archivo “**streamers.csv**”, se ha colocado en cada línea la información de un streamer registrado. Por cada línea del archivo se cuenta con el código del streamer (un número entero), el nombre del canal, la categoría principal en la que realiza sus transmisiones (por ejemplo: mtg, valorant, just_chatting, etc.) y la cantidad de seguidores acumulados. Un streamer puede realizar varias transmisiones.

En el archivo “**transmisiones.csv**”, se ha colocado el registro de las transmisiones realizadas en la plataforma durante el periodo considerado. Cada línea contiene el código de la transmisión, el código del streamer que realizó dicha transmisión, la fecha de transmisión (en formato dd-mm-aaaa) y la duración de la transmisión en minutos. Este archivo incluye transmisiones asociadas a streamers existentes en streamers.csv.

En el archivo “**comentarios.csv**”, se ha colocado el detalle de los comentarios registrados en las transmisiones. Por cada línea se cuenta con el código de la transmisión a la que pertenece el comentario, el autor del comentario (usuario que comentó) y el texto del comentario. Dado que una misma transmisión puede recibir múltiples comentarios, el código de transmisión puede aparecer repetidas veces en el archivo, reflejando que una única transmisión tiene varios comentarios asociados.

En el archivo “**stopwords.csv**”, se incluye un listado de palabras en idioma español que son considerados **stopwords**, es decir, palabras o símbolos muy frecuentes que generalmente se excluyen durante análisis de texto. Cada línea del archivo contiene una única palabra. El archivo se encuentra ordenado alfabéticamente.

CONSIDERACIONES PARA AMBAS PREGUNTAS

- No puede desarrollar una tarea si no ha desarrollado por completo la tarea anterior, de hacerlo no se corregirá esa tarea.
- 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. No se podrá emplear el carácter de tabulación (“\t”) para la emisión del reporte.
- Toda operación de búsqueda debe realizarse en una función independiente. No se considerará en la calificación los procesos de búsqueda que estén contenidos en el código de otro proceso. Las funciones de búsqueda deben considerar la posibilidad que el dato buscado no se encuentre.

PREGUNTA 1 (10 puntos): EN ESTA PREGUNTA SERÁ OBLIGATORIO EL USO DE ESTRUCTURAS Y ARREGLOS

Cree un proyecto en CLion con el nombre: “**Pregunta1_2025_2**” (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 y debe contener todas las funciones necesarias para su correcto funcionamiento.

Esta aplicación debe procesar tres archivos: el listado de streamers (**streamers.csv**), la lista de transmisiones (**transmisiones.csv**) y el listado de comentarios asociados a cada transmisión (**comentarios.csv**). Cargar y organizar eficientemente los comentarios por streamer y por transmisión, aplicar un **pre-procesamiento básico** sobre el texto procesado y generar un reporte de transmisiones.

Recuerde que “NO DEBE HACER QUE DOS O MÁS PUNTEROS APUNTEM AL MISMO DATO.”

Las tareas por realizar son las siguientes, las cuales debe desarrollarlas en el orden que se indican y cada una en una función independiente:

a) (1 punto) Definir las siguientes estructuras:

Comentario: que debe contener los siguientes campos:

1. **id_transmision** (valor entero que identifica la transmisión a la que pertenece el comentario),
2. **comentador** (cadena de caracteres dinámica con el nombre/usuario del autor del comentario),
3. **comentario_original** (cadena de caracteres dinámica con el texto tal como fue escrito en la transmisión),
4. **comentario_pre_procesado** (cadena de caracteres dinámica con el texto luego de aplicar limpieza o preprocesamiento).

Streamer: que debe contener los siguientes campos:

1. **id_streamer** (valor entero identificador único del streamer),
2. **nombre_canal** (cadena de caracteres dinámica con el nombre del canal),
3. **categoria** (cadena de caracteres dinámica que indica la categoría principal del canal),
4. **cantidad_seguidores** (valor entero con el total de seguidores del canal).

Transmision: que debe contener los siguientes campos:

1. **id_transmision** (valor entero identificador único de la transmisión),
 2. **streamer** (estructura **Streamer** asociada a la transmisión, que indica quién realizó la emisión),
 3. **comentarios** (arreglo dinámico de estructuras **Comentario** correspondientes a esa transmisión), considere máximo 10 elementos en este arreglo,
 4. **fecha** (valor entero que representa la fecha de la transmisión),
 5. **duracion** (valor entero que representa la duración total de la transmisión en minutos),
 6. **cantidad_comentarios** (valor entero que almacena cuántos comentarios están asociados a la transmisión dentro del arreglo dinámico).
- b) (1.0 punto) Leer los datos del archivo transmisiones.csv y alojarlos en una variable llamada **transmisiones** que es un arreglo dinámico de estructura **Transmision**. Verifique, como considere conveniente, que el arreglo tenga los datos leídos. Debe existir evidencia de la verificación en el programa.
- c) (2.0 punto) Leer los datos del archivo streamers.csv y completar el arreglo de **transmisiones** con la información del streamer leído. Verifique, como considere conveniente, que el arreglo tenga los datos leídos. Debe existir evidencia de la verificación en el programa.
- d) (4.0 puntos) Leer los datos del archivo comentarios.csv y alojarlos en el arreglo de **transmisiones**. Debe guardar la cadena original en **comentario_original** y la cadena pre procesada en **comentario_pre_procesado**. Para que el comentario sea pre-procesado básicamente debe realizar lo siguiente:
- Pasar el comentario a minúsculas.
 - Eliminar cualquier carácter que no se encuentre entre ‘A-Z’ o ‘a-z’.
- e) (2.0 puntos) Generar el reporte que se muestra a continuación:

Reporte de Transmisiones						
ID TRANSMISION	FECHA	DURACION	ID STREAMER	NOMBRE CANAL	CATEGORIA	CANTIDAD SEGUIDORES
5002	2025-06-03	279m	194	pixelcaster65	lol	446548
COMENTARIOS DE LA TRANSMISION 5002:						
ORIGINAL				PRE-PROCESADO		
deltafox	"No entendí la regla, ¿repites?"				noentendilareglarepitese	
mana_addict	¿Puedes mostrar la lista/deck al final?				puedesmostrarlaralistadeckalfinal	
silent_viewer	Pregunta: ¿a qué hora sueles streamear?				preguntaaquehorasuelesstreamear	
silent_viewer	"Buen stream, me encantó la jugada #18."				buenstreammeencantolajugada	
...						

PREGUNTA 2 (10 puntos): EN ESTA PREGUNTA SERÁ OBLIGATORIO EL USO DE ESTRUCTURAS AUTOREFERENCIADAS.

Cree un proyecto en CLion con el nombre: “Pregunta2_2025_2” (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 y debe contener todas las funciones necesarias para su correcto funcionamiento.

Esta aplicación debe procesar dos archivos: el listado de comentarios (**comentarios.csv**) y el listado de stopwords (**stopwords.csv**). Cargar ambos archivos en listas simplemente ligadas, eliminar los stopwords en la lista de palabras pertenecientes a los comentarios, construir otra lista simplemente ligada que permita contabilizar la frecuencia de cada palabra significativa y generar un ranking de las palabras más frecuentes.

Las tareas por realizar son las siguientes, las cuales debe desarrollarlas en el orden que se indican y cada una en una función independiente:

- a) (1.0 punto) Definir las siguientes estructuras:

Palabra: que debe contener los siguientes campos: 1) **original** (cadena de caracteres dinámica) y 2) **preproc** (cadena de caracteres dinámica) 3) **cantidad** (número entero que lleva la cantidad de veces que la palabra se ha repetido en los comentarios).

Nodo: que debe contener los siguientes campos: 1) **palabra** (estructura **Palabra**) y 2) **siguiente** (puntero a **Nodo**). Esta es una estructura auto-referenciada.

Para implementar la lista de **Nodo** puede decidir entre uno de los siguientes esquemas:

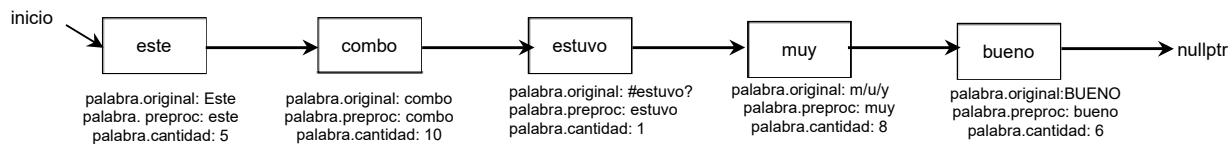
Esquema 1: Crear la estructura **ListaPalabra**: que debe contener el siguiente campo: 1) **inicio** (puntero a **Nodo**). Este es un **TAD**.

Esquema 2: Definir la lista usando **inicio** (puntero a **Nodo**)

- b) (1.0 punto) Crear la lista de **stopwords** usando el archivo **stopwords.csv**. En este archivo, las palabras se encuentran ordenadas alfabéticamente, no se requiere que la lista quede en orden, coloque el **stopword** en el campo **preproc**, el campo **original quedará vacío**, la cantidad de repeticiones para cada palabra en este archivo es 1.

- c) (0.5 puntos) Verificar en un reporte simple “**reporteStopwords.txt**” que la lista esté creada correctamente. El reporte debe tener un título, encabezados en las columnas y los datos con el formato apropiado.

- d) (2.5 puntos) Crear la lista de **palabras** usando el archivo **comentarios.csv**, las palabras no deben repetirse en la lista.



Recuerde que para que el proceso sea adecuado, debe pre-procesar cada palabra, siguiendo los siguientes pasos:

- Pasar a minúsculas cada palabra.
- Eliminar caracteres que no se encuentren dentro de ‘a-z’ o ‘A-Z’.

Se recomienda leer los comentarios palabra por palabra para asegurar una recuperación correcta de los datos. La lista no debe contener palabras repetidas en el campo preproc.

Utilice la función de la parte c) para emitir un reporte con el nombre “**reporteComentarios.txt**”. Si utiliza otra función no se le asignará puntaje a la parte c).

- e) (2.0 puntos) Remover de la lista de **palabras**, aquellas que se encuentren en la lista de **stopwords**. Con la finalidad de quedarse SOLO con las palabras más relevantes.

- f) (1.0 punto) Generar un archivo llamado **palabras_frecuencia.txt**, donde coloque por cada línea: la palabra y su cantidad de la lista de **palabras**.

g) (1.0 punto) Crear una lista de palabras llamada **frecuencias**, usando el archivo **palabras_frecuencia.txt**, generado por usted e inserte las palabras **ordenadas de mayor a menor por cantidad**. Si hay empate, ordene **alfabéticamente**.

h) (1.0 punto) Generar el reporte de Palabras Más Frecuentes que se muestra a continuación:

reporte_palabras_frecuentes.txt

PALABRAS FRECUENTES TRANSMISIONES NOVIEMBRE			
	#	Palabra significativa	Frecuencia
1	brutal		40
2	combo		40
3	ranked		39
4	sale		39
5	listadeck		38
6	mostrar		38
7	puedes		38
8	streamear		36
9	sueles		36
...			

Al finalizar el examen, comprima la carpeta que contiene los dos proyectos 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 este examen.

ADVERTENCIAS:

Obligatoriomente debe desarrollar sus proyectos bajo CLion 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 o guiones para separar las palabras compuestas (p. e.: baseInferior).
4. Se descontará 15% de la nota si no se colocan comentarios relevantes,
5. No se calificarán aquellas funciones implementadas en el archivo main.cpp

San Miguel, 01 de diciembre del 2025