

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

PROGRAMACIÓN 2
1ra 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**, **NI OBJETOS** (con excepción de los elementos de `iostream`, `omanip` y `fstream`). **NO PUEDE UTILIZAR LA CLASE `string`**. Tampoco se podrán emplear las funciones `malloc`, `realloc`, `memset`, `strtok` o `strdup`, 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 colocar un comentario en el que coloque claramente su nombre y código, **de no hacerlo se le descontará 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 AL 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.

INDICACIONES INICIALES

Cree un proyecto de C++ en CLION siguiendo estrictamente las indicaciones que a continuación se detallan:

- La unidad de trabajo será **t:** (Si lo coloca en otra unidad, no se calificará su laboratorio y se le asignará como nota cero)
- Cree allí una carpeta con el nombre **"Lab01_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 los proyectos solicitados en la prueba.**
- En cada proyecto que desarrolle deberá crear una carpeta denominada "Bibliotecas" y allí coloque la biblioteca de funciones que necesite. 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 1 del curso: "Funciones y alcance de variables". En este laboratorio se desarrollará una **biblioteca estática de funciones** en la que se implementen sobrecargas de operadores y funciones que le permitan solucionar el problema planteado.

En la carpeta solicitada anteriormente, cree **dos carpetas** denominadas **"CrearBiblioteca_Parte1"** y **"UsarBiblioteca_Parte2"**, allí colocará los proyectos solicitados en las partes 1 y 2 respectivamente de este laboratorio. **DE NO COLOCAR ALGUNO DE ESTOS REQUERIMIENTO SE LE DESCONTARÁ 3 PUNTOS DE LA NOTA FINAL. NO SE HARÁN EXCEPCIONES.**

TAMBIÉN SE LE PROPORCIONARÁ UN ARCHIVO DE CABECERA (.h) CON LAS ESTRUCTURAS DE DATOS QUE UTILIZARÁ PARA DESARROLLAR ESTE LABORATORIO, NO PUEDE MODIFICAR ESTE ARCHIVO, SI MODIFICA DE ALGUNA FORMA ESTE ARCHIVO SE LE DESCONTARÁ 5 PUNTOS EN SU NOTA FINAL.

PARTE 01 (15 puntos): **CREACIÓN DE LA BIBLIOTECA ESTÁTICA**

Se solicita que desarrolle una biblioteca estática en la cual se defina una serie de operadores sobrecargados que permitirá manejar las estructuras de datos que se le está proporcionando. Estas estructuras de datos están relacionadas con la matricula de alumnos a diversos cursos.

Las operaciones que la biblioteca estática permitirá realizar a través de sobrecargas de operadores se definen a continuación:

➤ **Lectura:**

- Sobrecargando el operador >> de modo que permita leer los datos de un curso de un archivo de tipo CSV. La operación (**arch >> curso;**) involucrará una variable de archivo y una variable de tipo "**struct Cursos**". La sobrecarga no deberá devolver valor alguno. Una línea de archivo tendrá la siguiente forma:

```
INF263,Algoritmia,3.75,35030611,INGA_FLORES_CESAR_ADOLFO
(código, nombre, créditos, código del docente y nombre del docente)
```

- Sobrecargando el operador >> de modo que permita leer los datos de un alumno de un archivo de tipo CSV. La operación (**arch >> alumno;**) involucrará una variable de archivo y una variable de tipo "**struct Alumnos**". La sobrecarga no deberá devolver valor alguno. Una línea de archivo tendrá la siguiente forma:

```
202111260,GORDILLO/CASTRO/RONAL,P
(código del alumno, nombre y modalidad de estudio)
```

➤ **Operaciones:**

- Sobrecargando el operador > de modo que permita determinar si el código de un alumno corresponde a la estructura Alumnos. La operación: **Alumno > codigoalumno;** tomará el código del alumno y verificará si es el mismo código de "Alumno" (**struct Alumnos**). Si es igual debe devolver el valor **true**, si no es igual devolverá un valor **false**.
- Sobrecargando el operador > de modo que permita determinar si el código de un curso corresponde a la estructura Cursos. La operación: **Curso > codigocurso;** tomará el código del curso y verificará si es el mismo código de "Curso" (**struct Cursos**). Si es igual debe devolver el valor **true**, si no es igual devolverá un valor **false**.
- Sobrecargando el operador += de modo que permita agregar un alumno a un curso. La operación **Curso += Alumno;** agregará el alumno (**struct Alumnos**) al curso correspondiente (**struct Cursos**). El curso puede tener varios tipos de alumnos matriculados: presenciales, semipresenciales y virtuales. Cada vez que se añade un alumno al curso, debe colocarse en el arreglo que le corresponde. El número de alumnos para el tipo asignado debe incrementarse.
- Sobrecargando el operador <= de modo que permita buscar un alumno en un arreglo de alumnos. La operación **ArrAlumnos <= Alumno;** buscará el alumno (**struct Alumnos**) en el arreglo de alumnos, esta función retorna el índice del alumno buscado dentro del arreglo. Como no se puede trabajar con la cantidad de datos en el arreglo, deberá tener como último dato un registro con el código con valor 0.
- Sobrecargando el operador >= de modo que permita buscar un curso en un arreglo de cursos. La operación **ArrCursos >= Curso;** buscará el curso (**struct Cursos**) en el arreglo de cursos, esta función retorna el índice del curso buscado dentro del arreglo. Como no se puede trabajar con la cantidad de datos en el arreglo, deberá tener asignado **nullptr** en el código del último curso.
- Sobrecargando el operador "++" de modo que se permita calcular el porcentaje de alumnos matriculados de forma presencial. La operación **Curso++** actualizará el campo **tipofinal** del curso

(**struct Cursos**), si en caso el porcentaje es mayor o igual al 50% debe grabar el texto "La clase se dictará en aula" en caso contrario se grabará el texto "La clase se dictará 100% virtual".

➤ **Impresión:**

- Sobrecargando el operador << de modo que permita imprimir la información de **un** Curso. La operación (**arch << Curso;**) permitirá imprimir en un archivo de textos los datos contenidos en una variable de tipo "**struct Cursos**". El formato será el siguiente:

```
INF263   Algoritmia
INGA_FLORES_CESAR_ADOLFO
Alumnos presenciales:    7
Alumnos semipresenciales: 5
Alumnos virtuales:      5
La clase se dictará 100% virtual
```

Los datos deben aparecer con un formato adecuado y alineados correctamente, tanto para los enteros (hacia la derecha), valores de punto flotante (a la derecha y al punto decimal) y cadenas de caracteres (a la izquierda). **No se dará puntaje a esta parte si no respeta esta condición. NO SE HARÁN EXCEPCIONES.**

Consideraciones:

La solución debe contemplar la elaboración de:

- 1) un proyecto de implementación y prueba de las sobrecargas, denominado "**Fuentes_Biblioteca_2025_2**", en este mismo proyecto genere la biblioteca estática (.a)
- 2) un proyecto donde se pruebe la biblioteca ya compilada "**Prueba_Biblioteca_2025_2**". La prueba de las sobrecargas puede ser hechas lo más simple posible pero que muestre claramente que son correctas. Los dos proyectos deberán colocarse en la carpeta "**Parte1_CrearBiblioteca**".

EN LAS PRUEBAS PARA LOS PROYECTOS 1 Y 2 SE PUEDE HACER DIRECTAMENTE EN LA FUNCIÓN MAIN, SE PUEDE HACER USO DE "HARD CODE" Y SE PUEDE SOBREPASAR LA CANTIDAD EL NUMERO DE LÍNEAS, SOLO EN ESTA PARTE DE LA PRUEBA.

EN LA BIBLIOTECA ESTÁTICA, NO PUEDE DESARROLLAR OTRAS FUNCIONES O SOBRECARGAS ADICIONALES A MENOS QUE ESTÉN DIRECTAMENTE RELACIONADAS A LAS SOBRECARGAS SOLICITADAS.

PARTE 2 (5 puntos): REUTILIZACIÓN DE LA BIBLIOTECA ESTÁTICA.

Desarrolle un proyecto denominado "**UnaAplicacionConBibEstatica**" (dentro de la carpeta "**Parte2_UsarBiblioteca**") en el cual se utilizarán obligatoriamente las sobrecargas de la biblioteca estática (**compilada**).

LA PREGUNTA NO SE EVALUARÁ SI EN ESTE PROYECTO NO SE USAN (O NO SE VE QUE USAN) LAS SOBRECARGAS IMPLEMENTADAS EN LA BIBLIOTECA COMPILADA (.A) Y LA BIBLIOTECA Y ESTRUCTURAS PROPORCIONADAS.

El proyecto ejecutará las tareas descritas a continuación utilizando las sobrecargas definidas en la biblioteca:

- a) Leer los datos de los alumnos contenidos en el archivo "Alumnos.csv" y los coloque en un arreglo de estructuras del tipo "**struct Alumnos**".
- b) Leer los datos de los cursos contenidos en el archivo "Cursos.csv" y los coloque en un arreglo de estructuras del tipo "**struct Cursos**".
- c) Leer los datos de cada matricula contenidos en un archivo de CSV como se muestra a continuación:

```
INF263,202318320
MEC289,202312000
...
```

Código del curso y Código del alumno

Con la información de cada línea deberá actualizar el arreglo de cursos colocando los alumnos matriculados en el curso correspondiente, recuerde que la información completa debe obtenerse del arreglo de alumnos. DEBE EMPLEAR DE MANERA OBLIGATORIA LOS OPERADORES SOBRECARGADOS.

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.

Profesores del curso: Rony Cueva
Erasmus Gómez
Andrés Melgar
Erick Huiza
Miguel Guanira

San Miguel, 5 de septiembre del 2025.