

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

2do.Examen

(Segundo Semestre 2021)

Indicaciones Generales:

- Duración: 3 horas.

Obligatoriamente los alumnos deberán mantener en todo momento el AUDIO Y VIDEO de sus computadores abierto de modo que puedan recibir los comunicados que se hagan durante el examen y la revisión de los trabajos que estén desarrollando. De tener algún problema deberán hacérselo saber de inmediato al profesor de su horario por correo. De no hacerlo, no se aceptarán reclamos alegando que no oyeron las indicaciones.

- No se pueden emplear **variables globales**, **estructuras** ni la **clase String**. Tampoco se podrán emplear las funciones malloc, realloc, strdup o strtok, 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. **SOLO SE PODRÁ HACER USO DE PLANTILLAS Y STL EN AQUELLAS PREGUNTAS QUE ASÍ LO SOLICITEN, DE LO CONTRARIO SE ANULARÁ LA PREGUNTA.**
- 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 (una línea de código no es una línea de texto, es una instrucción que termina con un punto y coma). 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 métodos de clases**. En cada archivo que implemente en los proyectos (.h y .cpp) deberá colocar un comentario en el que coloque claramente su nombre y código, de no hacerlo se le **descontará 0.5 puntos por archivo. (NO SE HARÁN EXCEPCIONES)**.
- El código comentado **NO SE CALIFICARÁ y esto incluye el comentar el llamado a la función que lo contiene.**
- La cláusula **friend** solo se podrá emplear en el caso de clases auto referenciadas para ligar el nodo con la clase **inmediata** que encapsula la lista, **en ningún caso adicional**. No se considerará en la nota las clases que violen esto. Tampoco se podrá emplear la cláusula **protected**.
- Deberá mantener en todo momento el encapsulamiento de todos los atributos de las clases, así como guardar los estándares en la definición y uso de todas las clases desarrolladas.
- Salvo en la sobrecarga de los operadores >> y <<, no se podrán definir funciones (ni plantillas de funciones) independientes que no estén ligadas como métodos a alguna de las clases planteadas.
- Los proyectos deben obligatoriamente desarrollarse en NetBeans bajo el sistema operativo Windows. No se revisarán los proyectos desarrollados en otros sistemas operativos o IDEs.
- **Todos los métodos y funciones ligados a una clase deben estar desarrollados en el mismo archivo.**
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no den resultados 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. ESTO Y EL HECHO DE ENCONTRAR CUALQUIER ARCHIVO YA SEA .cpp O .h CON FECHA U HORA DE CREACIÓN ANTERIOR A LA EVALUACIÓN SERÁ CONSIDERADO UNA FALTA DE PROBIDAD Y POR LO TANTO AMERITARÁ LA ANULACIÓN DE LA PRUEBA.

**NO SE HARÁN EXCEPCIONES ANTE CUALQUIER TRASGRESIÓN DE LAS INDICACIONES  
DADAS EN LA PRUEBA**

Puntaje total: 20 puntos

**PARTE 1 (10puntos) STL y Polimorfismo**

Se solicita que desarrolle un proyecto denominado **"STL\_Polimorfico\_Pregunta01"** dentro de la carpeta correspondiente, con el fin de registrar las solicitudes de libros de los usuarios, **DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 2 PUNTOS DE LA NOTA FINAL.**

Para esta tarea debe implementar las siguientes clases:

- Para manejar los libros de la Biblioteca: La clase se denominará "Libro" y deberá contener lo siguiente: 1) un atributo denominado **codigo** (char\*) definido por una cadena de caracteres, 2) un atributo denominado **titulo** (char\*) definido por una cadena de caracteres, 3) un atributo denominado **autor**(char\*) definido por una cadena de caracteres, 4) un atributo denominado **cantidad** (int) con la cantidad de libros que posee la biblioteca, 5) un atributo denominado **stock** (int) con la cantidad de libros que quedan luego de los préstamos, 6) un atributo denominado **anho** (int) con el año del libro, 7) un atributo denominado **precio** (double), con el valor del libro. Brindada en la librería estática liblibro.a
- Para manejar a los Usuarios: La clase se denominará "Usuario" y deberá contener lo siguiente: 1) un atributo denominado **carne** (int) con la identificación del usuario, 2) un atributo denominado **tipo** (char) con el tipo de usuario, 3) un atributo denominado **nombre** (char\*) definido por una cadena de caracteres. Brindada en la librería estática libusuario.a
- Para manejar a los Administrativos: La clase se llamará "Administrativo" y deberá contener lo siguiente: 1) un atributo denominado **area** (char\*) definido por una cadena de caracteres, donde se almacena el área al que pertenece el administrativo. Esta clase posee datos heredados de la clase Usuario.
- Para manejar a los Docentes: La clase se llamará "Docente" y deberá contener lo siguiente: 1) un atributo denominado **anexo** (int) que almacenará el anexo del docente. Esta clase posee datos heredados de la clase Usuario.
- Para manejar a los Estudiantes: La clase se llamará "Estudiante" y deberá contener lo siguiente: 1) un atributo denominado **celular** (int) que almacena el número celular del alumno. Esta clase posee datos heredados de la clase Usuario.
- Para manejar los Nodos de Usuarios: La clase se denominará "NUusuario" y deberá contener lo siguiente: 1) un atributo denominado **pusuario**, este es un puntero de la clase Usuario.
- Para manejar los Nodos de Solicitudes de Préstamos: La clase se denominará "NodoCola" y deberá contener lo siguiente: 1) un atributo denominado **libsol** (char\*) definido por una cadena de caracteres, que guarda el código del libro que el usuario desea en préstamo, 2) un atributo denominado **carne** (int) con la identificación del usuario que solicita el libro, 3) un atributo denominado **sig**, este atributo es un puntero a la clase NodoCola (auto referenciado).
- Para manejar la cola de Solicitudes de Préstamos: La clase se denominará "ColaSol" y deberá contener lo siguiente: 1) un atributo denominado **inicola**, este atributo es un puntero de clase NodoCola, 2) un atributo denominado **fincola**, este atributo es un puntero de clase NodoCola, 3) un atributo denominado **n** (int) con la cantidad de solicitudes que tiene la cola. Esta lista de nodos auto referenciados deberá trabajar como una cola, en ningún momento se puede recorrer como una lista.
- Para manejar los procesos: La clase se denominará "Procesa" y deberá contener lo siguiente: 1) un atributo denominado **llibros**, este atributo es un STL-list de la clase Libro, 2) un atributo denominado **cola**, este atributo es de la clase ColaSol, 3) un atributo denominado **vusuarios**, este atributo es un STL-vector de la clase NUusuarios.

**"DEBE EMPLEAR OBLIGATORIAMENTE LOS NOMBRES DE LAS CLASES Y SUS ATRIBUTOS"**

Con las clases indicadas debe realizar las siguientes operaciones:

- En la clase **Procesa** debe implementar el método **carga**, que se encargará a su vez de los siguientes procesos:
  - Implementar el método **CargaLibros** que se encargará de la lectura del archivo "libros2.csv" y cargar cada registro en la STL-list denominado **llibros**.
  - Implementar el método **CargaUsuarios** que se encargará de la lectura del archivo "usuarios2.csv" y cargar cada registro en la STL-vector denominado **vusuarios**. Para la carga de los usuarios debe usar el método polimórfico **leer**, ya que los usuarios tienen campos propios, los

administrativos guardan el área, los estudiantes su número de celular y los docentes el anexo de su oficina.

- Implementar el método **LeeSolicitudes** que se encargará de la lectura del archivo "SolicitudDePrestamos2.csv" y cargar cada solicitud de préstamo de un libro en la cola denominada **cola**. Las solicitudes de préstamo están en el mismo orden que presenta el archivo.
- En la clase **Procesa** debe implementar el método **Atiende**, que se encargará de **desencolar** una solicitud de préstamo de la **cola** y verificar en la STL-list **llibros** si aún hay stock del libro solicitado. Si hay libros, se debe actualizar el stock. Si en caso ya no se cuenta con stock del libro, se debe **encolar** nuevamente la solicitud.
- Finalmente, en la clase **Procesa** debe implementar el método **Imprime**, que se encargará de los siguientes procesos:
  - Implementar el método **Imprimelibros** para realizar la impresión de un archivo con la información de los libros (**para esta operación debe usar el método imprimelibro de la clase Libro**). Puede usar este método para probar la carga correcta de libros en la STL-list correspondiente.
  - Implementar el método **ImprimeCola** para realizar la impresión de las solicitudes de libros que **no fueron atendidos por falta de stock**. Para el desarrollo de este proceso se debe **desencolar** cada solicitud e imprimir la información necesaria del usuario almacenada en el STL-vector utilizando el método polimórfico **imprime**, como se muestra en el siguiente cuadro:

| Solicitudes pendientes   |
|--|
| Libro:FW74175<br>Carne:33115577<br>Nombre:Chumpitaz_Iberos_Ronal<br>Celular:985526212<br><br>Libro:MG75057<br>Carne:83265244<br>Nombre:Pairazaman_Alamo_Moises_Miguel<br>Anexo:4312<br><br>Libro:IW10872<br>Carne:54393647<br>Nombre:Norabuena_Preciado_Cinthia<br>Area:Tesoreria<br>... |

### Consideraciones:

Se le recomienda revisar al detalle los archivos csv, indicados a continuación:

| Libros2.csv  |
|--|
| VG01047,LenguajedeprogramacionC++,Raul Disman,2010,12,325.75 |
| MG75319,La historia del mundo,Maria Lopez,2017,25,187.55     |
| .....  |

Código,Título,Autor,Año de edición,Cantidad,Precio

| Usuarios2.csv                         |
|---------------------------------------|
| 30018447D,FUENTES QUISPE CINTHIA,4455 |
| 32517791A,ZORRILLA LARA ARTURO,DTI    |
| ...                                   |

Carné/Tipo,Nombre, Celular|Anexo|Area

| SolicitudDePrestamos2.csv |
|---------------------------|
| 26873214A,XN48844         |
| 11223344D,VN17150         |
| ...                       |

Carné/Tipo,Código de libro

Para el desarrollo de la pregunta debe considerar el siguiente código:

```
#include "Procesa.h"
using namespace std;
int main(int argc, char** argv) {
    Procesa pro;

    pro.Carga();
    pro.Atiende();
    pro.Imprime();
    return 0;
}
```

No puede  
cambiar este  
código

## PARTE2 (10 puntos)

Elabore un proyecto denominado "BibliotecaConSTL\_Pregunta02" y en él desarrollará el programa que dé solución al problema planteado. **DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 2 PUNTOS DE LA NOTA FINAL.**

Se tienen cuatro archivos del tipo CSV, los cuales se describen a continuación:

| Libros.csv   |
|--|
| VG01047,LenguajedeprogramacionC++,Raul Disman,2010,12,325.75 |
| MG75319,La historia del mundo,Maria Lopez,2017,25,187.55     |
| .....  |
| Código,Título,Autor,Edición,Stock,Precio                     |

| Usuarios.csv                           |
|--|
| 30018447E,FUENTES QUISPE CINTHIA DELIA |
| 32517791D,ZORRILLA LARA ARTURO         |
| ...                                    |
| Carné/Tipo,Nombre                      |

| RegistroDePrestamos.csv                    |
|--|
| 26873214A,XN48844,22/6/2021                |
| 11223344D,VN17150,14/3/2021                |
| ...  |
| Carné/Tipo,Código de libro,Fecha de retiro |

Se le está proporcionando las interfaces (archivos .h) de tres clases denominadas **Libro**, **Usuario** y **RegistroPrestamo** y la biblioteca estática "libbibliotecaclases.a" con la implementación de las clases, deberá incorporar **OBLIGATORIAMENTE** estos archivos al proyecto y utilizarlos para solucionar el problema planteado a continuación, **DE LO CONTRARIO NO SE LE EVALUARÁ LA PREGUNTA.**

Se desea elaborar una aplicación orientada a objetos que permita controlar los libros y usuarios de la biblioteca, para esto el proyecto deberá definir dos clases como se muestran a continuación:

| RegistroUsuario                    |
|------------------------------------|
| class Usuario usuario;             |
| list <RegistroPrestamo> prestamos; |

| Biblioteca                       |
|----------------------------------|
| list<Libro>libros;               |
| vector<RegistroUsuario>usuarios; |

**NO PUEDE MODIFICAR,  
BORRAR O AGREGAR  
ATRIBUTOS.**

La clase biblioteca deberá implementar como mínimo los siguientes métodos:

**void Biblioteca::leeLibros(const char\*);** El método leer los datos del archivo "Libros.csv" y llena el contenedor list<Libro>libros. Recibe como argumento el nombre del archivo.

**void Biblioteca::imprimeLibros(const char\*);** El método imprime todos los datos almacenados en el contenedor list<Libro>libros. Recibe como argumento el nombre del archivo del reporte.

**void Biblioteca::leeUsuarios(const char\*);** El método lee los datos del archivo "Usuarios.csv" y llena el contenedor vector<RegistroUsuario>usuarios, pero solo para el atributo "usuario" de la clase RegistroUsuario. Recibe como argumento el nombre del archivo.

**void Biblioteca::leePrestamos(const char\*);** El método lee los datos del archivo "RegistroDePrestamos.csv" y completa los datos de cada usuario con los libros que ha prestado, además deberá actualizar el atributo "prestados" de la clase Libro (ver interface de la clase). Recibe como argumento el nombre del archivo.

**void Biblioteca::imprimeUsuarios(const char\*);** El método imprime todos los datos almacenados en el contenedor vector<RegistroUsuario>usuarios (los usuarios y sus libros prestados). Recibe como argumento el nombre del archivo del reporte.

**void Biblioteca::imprimirSuspendidos(const char\*, int, int, int);** El método imprime solo los datos de los usuarios (no sus libros prestados) almacenados en el contenedor vector<RegistroUsuario>usuarios, que han quedado suspendidos por poseer algún libro con más de 7 días de retraso. Recibe como argumento el nombre del archivo del reporte y la fecha del proceso (día, mes y año).

Además, usted podrá definir todos los métodos que crea conveniente para las clases Biblioteca y RegistroUsuario de modo que pueda solucionar el problema.

El código de la función main será el siguiente:

```
#include "Biblioteca.h"

int main(int argc, char** argv) {
    class Biblioteca biblioteca;
    biblioteca.leeLibros("Libros.csv");
    biblioteca.imprimeLibros("ReporteLibrosInicial.txt");
    biblioteca.leerUsuarios("Usuarios.csv");
    biblioteca.imprimeUsuarios("ReporteUsuariosInicial.txt");
    biblioteca.leerPrestamos("RegistroDePrestamos.csv");
    biblioteca.imprimeUsuarios("ReporteUsuariosFinal.txt");
    biblioteca.imprimeLibros("ReporteLibrosFinal.txt");
    biblioteca.imprimirSuspendidos("ReporteSuspendidos.txt",19,10,2021);
    return 0;
}
```

**NO PUEDE CAMBIAR ESTE  
CÓDIGO, DE HACERLO  
RECIBIRÁ UN DESCUENTO  
DE 2 PUNTOS EN LA NOTA  
FINAL DE LA PREGUNTA**

**LOS ARCHIVOS CSV SOLO SE PUEDEN LEER UNA VEZ.**

**LOS DATOS DE LOS REPORTES DEBEN APARECER ALINEADOS CORRECTAMENTE CON  
ENCABEZADOS APROPIADOS QUE INDIQUEN LA NATURALEZA DE LOS DATOS (NO PODRÁ  
EMPLEAR EL CARÁCTER '\t')**

---

### **PUNTAJES:**

#### **PREGUNTA 1:**

- Implementar el método **CargaLibros** 0.5 puntos.
- Implementar el método **CargaUsuarios** 0.5 puntos.
- Implementar el método **LeeSolicitudes** 1.0 puntos.
- Implementar el método **Atiende** 4.0 puntos.
- Implementar el método **Imprimelibros** 1.0 puntos.
- Implementar el método **ImprimeCola** 3.0 puntos.

#### **PREGUNTA 2:**

- Lectura y asignación de libros y "ReporteLibrosInicial.txt" 1.5 puntos.
- Lectura y asignación de usuarios y "ReporteUsuariosInicial.txt" 1.5 puntos.
- Lectura y asignación de préstamos, actualización de libros, "ReporteUsuariosFinal.txt" y "ReporteLibrosFinal.txt" 4.0 puntos.
- Determinación e impresión de suspendidos 3.0 puntos.

**NO SE ASIGNARÁ PUNTAJE A LOS REPORTES SI NO SE IMPLEMENTAN DE FORMA CORRECTA LOS PROCESOS DE CARGA O ACTUALIZACIÓN.**

---

Al finalizar el examen, 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**. Luego súbalo a la tarea programa en Paideia para este examen.

Profesor del curso: Rony Cueva  
Miguel Guanira

San Miguel, 07 de diciembre del 2021.