

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

2do. Examen

(Segundo Semestre 2022)

Indicaciones Generales:

- Duración: 3 horas.

SOLO ESTÁ PERMITIDO EL USO DE APUNTES DE CLASE. NO PUEDE UTILIZAR FOTOCOPIAS NI MATERIAL IMPRESO, TAMPOCO PODRÁ EMPLEAR HOJAS SUELTAS.

- No se pueden emplear **variables globales, estructuras, ni la clase (o el tipo de datos) 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 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. (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.
- **Todos los métodos y funciones ligados a una clase deben estar desarrollados en el mismo archivo.**
- **NO PUEDE EMPLEAR ARREGLOS NI ARCHIVOS AUXILIARES PARA COLOCAR PARTE O TODOS LOS DATOS DE LOS ARCHIVOS DADOS**
- 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.
- **NO PUEDE UTILIZAR VARIABLES ESTÁTICAS.**

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

- 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 **"CO_PA_PN_EX_FINAL_2022_2"** 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.**

PARTE 1 (10 puntos)

Elabore un proyecto denominado "Pregunta01_ServicioDeSalud" y en él desarrollará el programa que dé solución al problema planteado. **DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ DOS (2) PUNTOS DE LA NOTA FINAL.**

Una institución de salud desea una aplicación orientada a objetos que permita manejar las consultas de sus pacientes, así como las medicinas que entrega. Para esto se tienen dos archivos del tipo CSV, los cuales se describen a continuación:

Medicinas-Preg01.csv	
60509,AMPICILINA 125MG SUSP 90ML,58.65	
73972,VITAMINA E 400 MG C90 CAPS,54.4	
.....	
Código, Descripción, Precio unitario	

Consultas-Preg01.csv	
9/02/2022,37282858,Soma/Lourdes,OW18266,17790,23,81821,8,96881,17,44253,12,34080,35,34027,9	
4/10/2022,81539774,Alfaro/Erick,VJ18711,19397,20,83801,35,72495,11,77203,1,52072,17	
...	
Fecha de la consulta,DNI del paciente, nombre del paciente, código del médico que lo atendió y la lista de medicamentos. Los medicamentos se identifican por su código y por la cantidad medicada.	

Se le está proporcionando las interfaces (archivos .h) de dos clases denominadas **Medicina** y **Paciente** y la biblioteca estática "libbibliotecadeclasses.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.**

El proyecto deberá definir tres clases como se muestran a continuación:

Medicina_Cantidad	Paciente_Medicina	ServicioDeSalud
int código; int cantidad;	class Paciente paciente; list<class Medicina_Cantidad>medicina_cantidad; double totalDeGastos;	map<int,Medicina>medicina; list<Paciente_Medicina>paciente_medicina;

NO PUEDE MODIFICAR, BORRAR O AGREGAR ATRIBUTOS

PARA LA LECTURA E IMPRESIÓN DE DATOS: NO DEBE REPETIR AQUÍ LOS PROCESOS DE LECTURA e IMPRESIÓN DADOS PARA LAS CLASES: "Medicina", "Paciente" ni "Medicina_Cantidad", DEBE EMPLEAR OBLIGATORIAMENTE LOS MÉTODOS DADOS PARA ESTE FIN.

En la clase **Medicina_Cantidad** debe definir todos los métodos que sean necesarios, pero será obligatorio que implemente el operador >> que solo lea un par de datos del archivo Consultas-Preg01.csv (código y cantidad) y los asigne a los atributos correspondientes. Además, debe implementar un método **void imprimeMedicina (ofstream &)** el método muestre ambos atributos en una sola línea y bien tabulados.

Para la clase **Paciente_Medicina** debe definir como mínimo el método **totalizar(map<int, Medicina>med)**; el método debe recibir como parámetro una variable de tipo **map** y con ella calcular el costo total de medicinas que recibió el paciente.

Para la clase **ServicioDeSalud** deberá implementar como mínimo los siguientes métodos:

void leerMedicinas (const char*); El método lee los datos del archivo "Medicinas-Preg01.csv" y llena el contenedor "map<int,Medicina>medicina". Recibe como argumento el nombre del archivo.

void imprimirMedicinas (const char*); El método imprime TODOS los datos almacenados en el contenedor "map<int,Medicina>medicina". Recibe como argumento el nombre del archivo del reporte. Debe colocar un título general y encabezados encima en cada columna. Debe alinear correctamente los encabezados.

void leerConsultas (const char*); El método lee los datos del archivo "Consultas-Preg01.csv", primero debe leer los datos del paciente y luego la lista de medicamentos al contenedor "paciente_medicina". En esta operación deberá verificar que **NO SE AGREGUEN PACIENTES REPETIDOS AL CONTENEDOR**, en este caso debe reemplazar la fecha y el código del médico por la última consulta, **TAMPOCO DEBE COLOCAR MEDICAMENTOS REPETIDOS**, en este caso solo debe acumular las cantidades.

void totalizar () el método debe calcular el total gastado por cada paciente.

void imprimePacientes (const char*); el método muestra en un archivo de textos todos los pacientes de la institución de salud, seguidos por las medicinas que le fueron entregadas en todas sus consultas, al final se colocará el total gastado por el cliente con una etiqueta adecuada. Debe colocar un título general y encabezados encima en cada columna. Debe alinear correctamente los encabezados.

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

El código de la función main será el siguiente:

```
#include "ServicioDeSalud.h"
int main(int argc, char** argv) {
    class ServicioDeSalud servicioDeSalud;
    servicioDeSalud.leerMedicinas("Medicinas-Preg01.csv");
    servicioDeSalud.imprimirMedicinas("PruebaMedicinas.txt");
    servicioDeSalud.leerConsultas("Consultas-Preg01.csv");
    servicioDeSalud.imprimePacientes("PruebaConsultas.txt");
    servicioDeSalud.totalizar();
    servicioDeSalud.imprimePacientes("PruebaConsultasTotal.txt");
    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.**
- **EN LOS REPORTES NO PODRÁ EMPLEAR EL CARÁCTER '\t'.**
- **NO PUEDE COLOCAR LOS DATOS DEL ARCHIVO EN ESTRUCTURAS AUTOREFERENCIADAS AJENAS A LOS CONTENEDORES DADOS.**

PUNTAJES PREGUNTA 1:

- | | |
|--|-------------|
| a) Lectura, asignación y reporte de Medicinas: | 1.5 puntos. |
| b) Implementación de la clase Medicina_Cantidad: | 1.0 puntos. |
| c) Lectura y asignación correcta de los pacientes y sus medicinas: | 4.0 puntos. |
| d) Totalizar las medicinas de cada Paciente | 2.0 puntos. |
| e) Reporte completo, parcial y final, de los pacientes | 1.5 puntos. |

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

PARTE 2 (10 puntos) Autoreferencia y Polimorfismo

Se solicita que desarrolle un proyecto denominado **"Pregunta02_Lotes"** dentro de la carpeta correspondiente, con el fin de registrar los lotes de medicamentos que están activos o vencidos. Para esta tarea debe implementar las siguientes clases:

- **Para manejar los lotes de medicamentos:** La clase se denominará **"medicamento"** y deberá contener lo siguiente: 1) un atributo denominado **codigo** (**int**), 2) un atributo denominado **nombre** definido por una cadena de caracteres, 3) un atributo denominado **stock** (**int**), 4) un campo denominado **fechavenc** (**int**) en el formato **aaaammdd** que representa la fecha de vencimiento del lote de medicamentos, 5) un campo denominado **estado** (**int**) que almacena 1 cuando el medicamento está activo o 0 cuando el medicamento está vencido.
- **Para manejar los lotes de medicamentos activos:** La clase se denominará **"activo"** y deberá contener lo siguiente: 1) un atributo denominado **precio** (**double**), 2) un atributo denominado **vendedor** definido por una cadena de caracteres. Además, esta clase posee datos heredados de la clase **medicamento**.
- **Para manejar los lotes de medicamentos vencidos:** La clase se denominará **"vencido"** y deberá contener lo siguiente: 1) un atributo denominado **fechabaja** (**int**) que almacenará la fecha de revisión de los lotes y se pasan los medicamentos activos a vencidos, 2) un atributo denominado **inspector** definido por una cadena de caracteres, que almacenará el nombre del inspector que da de baja el medicamento. Además, esta clase posee datos heredados de la clase **medicamento**.
- **Para manejar los Nodos:** La clase se denominará **"Nodo"** y deberá contener lo siguiente: 1) un atributo denominado **med**, este es un puntero de la clase **medicamento**, 2) un atributo denominado **izq**, este

atributo es un puntero a la clase **Nodo** (autoreferenciado), 3) un atributo denominado **der**, este atributo es un puntero a la clase **Nodo** (autoreferenciado).

➤ **Para manejar el árbol:** La clase se denominará "**Arbol**" y deberá contener lo siguiente: 1) un atributo denominado **raiz**, este atributo es un puntero de clase **Nodo**. Esta estructura se trata de un ABB ordenado por la fecha del lote de medicamentos, en caso hay más de un lote con la misma fecha el orden estará determinado por el código del medicamento.

➤ **Para manejar el almacén:** La clase se denominará "**almacen**" y deberá contener lo siguiente: 1) un atributo denominado **arbolalma**, este atributo es un objeto de la clase **Arbol**, donde se guardarán todos los medicamentos que existen en el archivo a cargar.

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

Con las clases indicas debe realizar las siguientes operaciones:

- (4 puntos)** En la clase **almacen** debe implementar el método **carga**, que se encargará de la lectura del archivo "lotes.csv" y cargar la información en el árbol ABB denominada **arbolalma**. Para la lectura de los datos del pedido debe utilizar el método **polimórfico lee**. Además, la fecha recibida como parámetro se debe comparar con la fecha de vencimiento del material, si esta es menor se debe considerar la lectura de un material **vencido**, en caso contrario el material será **activo**. La cadena que recibe como parámetro el método **carga** será utilizado para almacenar el nombre del inspector.
- (4 puntos)** En la clase **almacen** debe implementar el método **actualiza** que, de acuerdo con la fecha recibida como parámetro, se debe comparar con la fecha de vencimiento del material, si esta es menor y el material es **activo** se debe modificar el nodo, pasando de un material **activo** a un material **vencido**. Para que esta pregunta sea válida no debe eliminar el nodo, pero si debe eliminar de forma correcta la memoria asignada a la clase **activo** y utilizar polimorfismo para registrar los datos que necesita una clase **vencido**. La cadena que recibe como parámetro el método **actualiza** será utilizado para almacenar el nombre del inspector.
- (2 puntos)** Finalmente, en la clase **almacen** debe implementar el método **imprime**, que se encargará de realizar la impresión de un archivo con la información de los productos (**para esta operación debe usar el método polimórfico imprime de la clase medicamentos**).

REPORTE DE LOTES DE MEDICAMENTOS					
Fecha	Codigo	Nombre del Medicamento	Stock	Vendedor/Inspector	Fecha Baja/Precio
2022/06/01	79851	CEFOTAXIMA 1GR IM	425	R. CUEVA	2022/07/01
2022/06/01	98261	ACICLOVIR 200 MG C/25 CPR	260	R. CUEVA	2022/07/01
2022/08/01	30430	PRUEBA DE EMBARAZO	200	K BORJA	22.10
2022/08/01	34027	DANAZOL 100MG 50T	365	K BORJA	79.05
2022/09/01	40997	ELECTROLIT PED FRESA 625ML	670	J. CARHUAS	41.65
2022/10/01	16055	AMBROXOL C/20 TAB	810	O. DUENAS	45.90
2022/10/01	27016	ACECLOFENACO 100MG 10T	940	ALUMNO	2022/11/29
2022/10/01	30220	AMOXICILINA+BROMEXINA C12	600	ALUMNO	2022/11/29
...					

Consideraciones:

- El main debe contar con la siguiente información:

```
#include "almacen.h"
using namespace std;

int main(int argc, char** argv) {
    almacen alma;
    alma.carga(20220701, "APELLIDO DE SU PROFESOR");
    alma.actualiza(20221129, "SU APELLIDO");
    alma.imprime();

    return 0;
}
```

**NO PUEDE
CAMBIAR
ESTE
CÓDIGO**

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

lotes.csv						
20221001,27016,ACECLOFENACO 100MG 10T,940,J. BACA,51						
20231001,27016,ACECLOFENACO 100MG 10T,940,O. DUENAS,51						
20220601,98261,ACICLOVIR 200 MG C/25 CPR,260,R. TOKUMORI,70.55						
20230601,72495,ACIDO GLUTAMICO 600 MG C/100,720,J. CARHUAS,43.35						
...						
Fecha Venc.,Codigo,Nombre del med.,Stock,Nombre del Vendedor,precio.						

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, 29 de noviembre del 2022.