PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

2do.Examen

(Primer Semestre 2022)

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. Tanto los Profesores como los Jefes de práctica podrán solicitarles en control de sus computadoras y cuando esto suceda deberán darles el acceso inmediatamente.

- 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.**
- **SOLO PODRÁ TENER ABIERTO EN SU NetBeans LOS PROYECTOS QUE SE INDICAN EN LA PRUEBA. TAMPOCO PODRÁ EJECUTAR PROGRAMAS QUE NO TENGAN QUE VER CON EL LABORATORIO.**
- **SE DESCONTARÁ PUNTAJE, SEGÚN LA GRAVEDAD, DE SI SE COLOCAN EN EL PROYECTO FUNCIONES QUE NADA TENGAN QUE VER CON LA SOLUCIÓN DEL PROBLEMA.**
- **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.
- **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. 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

Cuestionario:

Deberá crear una carpeta denominada "EXAMEN_FINAL_2022_1 y en ella colocará los dos proyectos que den solución los problemas planteados. DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 3 PUNTOS DE LA NOTA FINAL.

PARTE 1 (10 puntos)

Elabore un proyecto denominado "Pregunta01_Producto_Clientes" 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.

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

Clientes.csv

```
71984468,IPARRAGUIRRE VILLEGAS NICOLAS EDILBERTO B,935441620
29847168,ALDAVE ZEVALLOS ROSARIO,6261108
……
```

DNI, Nombre Categoria, Teléfono
UN CLIENTE PUEDE NO TENER CATEGORÍA

Pedidos.csv

```
306824,FLAN LALA SABOR VAINILLA 100GR,5.88,8.13,71984468,29/03/2022
186026,YOGHURT NATURAL LIGHT LALA 1KG,%4.7,25.86,7.38,71378466,19/03/2020
…
```

Código del producto,Descripción,Descuento(si lo tiene), precio unitario,cantidad, DNI del cliente,Fecha del pedido.

Se le está proporcionando las interfaces (archivos .h) de dos clases denominadas **Cliente y Producto** y la biblioteca estática **"libbibliotecadeclases.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 manejar los clientes, productos y pedidos de una empresa comercializadora de abarrotes, para esto el proyecto deberá definir tres clases como se muestran a continuación:

Pedido
int codigo; double cantidad; int fecha; double subTotal;

RegCliente
class Cliente cliente; list <class Pedido> pedidos;

Empresa
list <class Producto> lstProductos; list <class RegCliente> lstClientes;

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: "Cliente", "Producto" ni "Pedido", DEBE EMPLEAR OBLIGATORIAMENTE LOS MÉTODOS DADOS PARA ESTE FIN.

En la **clase Pedido** debe definir todos los métodos que sean necesarios, pero será obligatorio que implemente un método **int leerDatos (ifstream &)** que lea solo los tres últimos datos de una línea del archivo Pedidos.csv y los asigne a los atributos correspondientes, el método recibe la variable de archivo y devuelve el dni del cliente. Además debe implementar un método **void imprimirDatos (ofstream &)** el método recibe la variable de archivo y que, en una línea y sin etiquetas, muestra todos los atributos en una sola línea y bien tabulados.

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

void leerClientes (const char*); El método leer los datos del archivo "Clientes.csv" y llena el contenedor "list <class RegCliente> lstClientes". Recibe como argumento el nombre del archivo. Esta operación debe asignar al atributo "descuento", de la clase Cliente, el valor de 23.5% a los clientes de tipo A, 16.8% a los clientes de tipo B, 8.3% a los de tipo C y 0% a los que no tienen tipo.

void imprimirClientes (const char*); El método imprime TODOS los datos almacenados en el contenedor "list <class RegCliente> lstClientes". Recibe como argumento el nombre del archivo del reporte. Debe colocar un título general y encabezados encima en cada columna. Debe alinear correctamente los datos. El orden de impresión para los clientes será: código, nombre, categoría, descuento y teléfono, para los pedidos: fecha (en la forma: dd/mm/aaaa), código, cantidad y subtotal.

void leerPedidos (const char*); El método lee los datos del archivo "Pedidos.csv" y, primero debe leer los datos del producto y agregar el producto al contenedor "list <class Producto> lstProductos", en esta operación deberá verificar que NO SE AGREGUEN PRODUCTOS REPETIDOS AL CONTENEDOR. Luego debe leer los tres últimos datos del archivo del archivo pedido.csv, asignar a los atributos correspondiente y finalmente completar los campos "codigo" y "subTotal para luego agregarlo al contenedor list <class Pedido> pedidos del cliente correspondiente. El atributo "subtotal" se calcula como: cantidad X precio X descuento del producto X el descuento por el tipo de cliente. Recibe como argumento el nombre del archivo.

void empresa.ordenarPedidos(); el método ordena los pedidos de todos los clientes por la fecha del pedido.

void imprimirProductos(const char*); el método muestra en un archivo de textos todos los productos de la empresa. Debe colocar un título general y encabezados encima en cada columna. Debe alinear correctamente los datos. El orden de impresión será: código, descripción, descuento y precio.

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

El código de la función main será el siguiente:

```
#include "Empresa.h"

int main(int argc, char** argv) {

   class Empresa empresa;
   empresa.leerClientes("clientes.csv");
   empresa.imprimirClientes("PruebaClientes.txt");
   empresa.leerPedidos("Pedidos.csv");
   empresa.ordenarPedidos();
  empresa.imprimirProductos("ReporteDeProductos.txt");
   empresa.imprimirClientes("ReporteDeClientes.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 ARREGLOS O ESTRUCTURAS AUTO REFERENCIADAS AJENAS A LOS CONTENEDORES DADOS.**

PUNTAJES PREGUNTA 1:

a) Lectura y asignación de clientas y reporte parcial de clientes (sin pedidos):	1.5 puntos.
b) Implementación de la clase Pedidos	1.5 puntos.
c) Lectura y asignación de productos y pedidos y "ReporteDeProductos.txt";	4.0 puntos.
d) Ordenación de pedidos	2.0 puntos.
e) Reporte final de clientes (todo)	1.0 punto.

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

PARTE 2 (10 puntos) STL y Polimorfismo

Se solicita que desarrolle un proyecto denominado **"STL_Polimorfico_Pregunta02"** dentro de la carpeta correspondiente, con el fin de registrar las solicitudes de libros de los usuarios. Para esta tarea debe implementar las siguientes clases:

- **Para manejar los pedidos realizados**: La clase se denominará **"Pedido"** y deberá contener lo siguiente: 1) un atributo denominado **codigo** (**int**) que representa el código del producto, 2) un atributo denominado **cantidad** (**int**) que representa la cantidad de productos solicitado, 3) un atributo

denominado **dni** (int) que representa el dni del cliente que realiza el pedido, 4) un atributo denominado **fecha** (int) que representa la fecha donde se realiza el pedido con el formato **aaaammdd**, 5) un atributo denominado **total** (double) que representa el **monto del pedido**, 6) un atributo denominado **orden** (int) que representa la prioridad de atención del pedido (1=Si o 0=No).

- **Para manejar los productos:** La clase se denominará "**Producto**" y deberá contener lo siguiente: 1) un atributo denominado **codprod** (int), 2) un atributo denominado **nombre** (char *) definido por una cadena dinámica de caracteres, 3) un atributo denominado **stock** (int) que representa la cantidad de productos que se tienen en stock. **Además, se sabe que algunos productos tienen diferentes descuentos y si son solicitados, en los pedidos pueden cambiar su orden dentro de la lista de programación.**

- **Para manejar los productos de categoría 1:** La clase se denominará "**Categoria1**" y deberá contener lo siguiente: 1) un atributo denominado **prioridad** (int) que representa el orden del producto dentro de la programación de los pedidos, 2) un atributo denominado **minimo** (int) que representa la cantidad de productos mínima en stock para que un producto sea despachado. Además, esta clase posee datos heredados de la clase **Producto.**

- **Para manejar los productos de categoría 2:** La clase se denominará "**Categoria2**" y deberá contener lo siguiente: 1) un atributo denominado **prioridad** (int) que representa el orden del producto dentro de la programación de los pedidos, 2) un atributo denominado **descuento** (double) que representa el descuento que se le brinda a la venta a este producto. Además, esta clase posee datos heredados de la clase **Producto.**

- **Para manejar los productos de categoría 3:** La clase se denominará "**Categoria3**" y deberá contener lo siguiente: 1) un atributo denominado **prioridad** (int) que representa el orden del producto dentro de la programación de los pedidos, 2) un atributo denominado **descuento** (double) que representa el descuento que se le brinda a la venta a este producto. Además, esta clase posee datos heredados de la clase **Producto.**

- **Para manejar los Nodos de Productos**: La clase se denominará "**NProductos**" y deberá contener lo siguiente: 1) un atributo denominado **prod**, este es un puntero de la clase **Producto**.

- **Para manejar los Nodos**: La clase se denominará "**Nodo**" y deberá contener lo siguiente: 1) un atributo denominado **ped,** este es un puntero de la clase **Pedido**, 2) un atributo denominado **sig**, este atributo es un puntero a la clase **Nodo** (autoreferenciado).

- **Para manejar la lista**: La clase se denominará "**Lista**" y deberá contener lo siguiente: 1) un atributo denominado **lini**, este atributo es un puntero de clase **Nodo**, 2) un atributo denominado **lfin**, este atributo es un puntero de clase **Nodo**. La lista deberá esta ordenada por la **fecha** del pedido (ascendente). Esta estructura es una lista simplemente enlazada.

- **Para manejar el programa de pedidos**: La clase se denominará "**Programa**" y deberá contener lo siguiente: 1) un atributo denominado **lpedidos**, este atributo de la clase **Lista**, donde se guardarán todos los pedidos de los clientes, 2) un atributo denominado **vproductos**, este atributo es un STL-vector de la clase **NProductos.**

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

Con las clases indicas debe realizar las siguientes operaciones:

- (**3 puntos**) En la clase **Programa** debe implementar el método **carga,** que se encargará a su vez de los siguientes procesos:
  - Implementar el método **CargaProductos** que se encargará de la lectura del archivo "Productos4.csv" y cargar cada registro en la STL-vector denominado **vproductos**. Para la carga de los productos debe usar el método polimórfico **leer**, ya que los productos tienen campos propios de acuerdo con su categoría. Si al momento de la lectura del archivo el producto tiene como categoría el número 1 debe cargar la clase **Categoria1**, si el producto tiene como categoría

el número 2 debe cargar la clase **Categoria2** y si el producto tiene como categoría el número 3 debe cargar la clase Categoria3.

- Implementar el método **CargaLista** que se encargará de la lectura del archivo "Pedidos4.csv" y cargar cada pedido solicitado en la lista denominada **lpedidos**. Los pedidos deben cargarse ordenados por fecha, si hay pedidos para la misma fecha pueden colocarlas antes o después del pedido existente.

- (**4 puntos**) En la clase **Programa** debe implementar el método **actualiza,** que se encargará de **reordenar** la lista de acuerdo con la prioridad de atención que tiene el producto del pedido, este dato debe ser verificado en la STL-vector **vproductos**. Los pedidos que tienen prioridad 1 van primero, el resto van en su posición original. Si hay varios pedidos con prioridad 1 se colocan siempre al inicio de la lista. Durante el proceso se debe actualizar el atributo **orden** con el valor 1 si el pedido tiene prioridad o el valor 0 si no tiene prioridad, de acuerdo con el producto.
- (**3 puntos**) Finalmente, en la clase **Programa** debe implementar el método **muestra,** que se encargará de los siguientes procesos:
  - Implementar el método **ImprimeProductos** para realizar la impresión de un archivo con la información de los productos (**para esta operación debe usar el método polimórfico imprime de la clase Producto**). Puede usar este método para probar la carga correcta de productos en la STL-vector correspondiente.
  - Implementar el método **ImprimeLista** para realizar la impresión de los pedidos considerando el reordenamiento realizado como se muestra en el siguiente cuadro:

```
                          Programa de Pedidos

Fecha         Codigo del Producto    Cantidad     Monto      Prioridad
================================================================================
05/02/2021          422763                3        1.50          Si
26/01/2021          126218               11        5.50          Si
01/01/2021          285807                4       20.00          No
01/01/2021          580927                4       20.00          No
...
```

Consideraciones:

- No es necesario emplear o aplicar el atributo mínimo ni descuento, pero si debe cargar e imprimir los datos que leídos del archivo.
- Se le recomienda revisar al detalle los archivos csv, indicados a continuación:

Pedidos4.csv

```
178943,2,6,46462527,16/03/2021
179485,11,55,33397650,15/01/2021
182942,2,2,46462527,12/10/2021
…
```

Código, Cantidad solicitada, monto del pedido, DNI del cliente,Fecha del pedido.

Productos4.csv

```
2,0,25,882099,NESQUIK CHOCOLATE 200GR,15
2,0,25,375507,TE DE MANZANILLA McCORMICK 100 1.2GR,15
3,1,40,459032,GELATINA DANY LIMON 125GR,24
…
```

Categoría,Prioridad,Stock mínimo/descuento,Código, Nombre del producto,Stock.

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, 12 de julio del 2022.