# PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
# FACULTAD DE CIENCIAS E INGENIERÍA

## LENGUAJES DE PROGRAMACIÓN 1

### 1er. Examen

### (Segundo Semestre 2023)

**Indicaciones Generales:**

- Duración: 170 minutos (2 horas con 50 minutos) .

**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 objetos** (con excepción de los elementos de iostream, iomanip y fstream). **No puede utilizar la clase (o el tipo de datos) string**. Tampoco se podrán emplear las funciones malloc, realloc, 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 NI MACROS EN ESTE EXAMEN.**
- 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 muestres resultados o que estos no sean coherentes en base al 60%.
- **EN NINGUNA PREGUNTA PODRÁ UTILIZAR VARIABLES CON DOS O MÁS ÍNDICES**
- **NO PODRÁ COLOCAR LOS DATOS EN ARREGLOS AUXILIARES A LOS INDICADOS EN LAS PREGUNTAS O LOS PROPIOS DE LOS MÉTODOS SOLICITADOS**
- 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 NetBeans 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 **"CO_PA_PN_EX01_2023_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.**

---

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

**Archivo de productos (CSV)**

```
YOT-530,Deshumedecedor DM-190H,941.73,13
NSR-955,Cocina a gas Blogna,2591.44,21
…
```

código, descripción, precio y stock final.

**Archivo de pedidos (CSV)**

```
PVZ-181,26290971,12,28/8/2023
ICX-503,27912250,5,5/9/2023
AVN-710,54602211,17,27/8/2023
…
```

Código del producto, DNI del solicitante, cantidad y fecha del pedido.

## PUNTEROS MÚLTIPLES

**PREGUNTA 1 (6 puntos)**

Elabore un proyecto denominado "Pregunta01Examen01PunterosMultiples" 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.

Con esta información, la función "main" del proyecto estará compuesto por el siguiente código:

```
#include " Pregunta01Examen01PunterosMultiples.h"

int main(int argc, char** argv) {
   char **codigoDelProducto;
   int **fechaClienteCantidad;

   cargarPedidos (fechaClienteCantidad, codigoDelProducto, "Pedidos.csv");
   pruebaDeCargaDePedidos (fechaClienteCantidad, codigoDelProducto, "PruebaPedidos.txt");

   return 0;
}
```

**NO PUEDE CAMBIAR ESTE CÓDIGO**

Implemente las funciones ***cargarPedidos y pruebaDeCargaDePedidos***, la función ***"cargarPedidos"*** debe leer los datos del archivo "***Pedidos.csv***" y colocarlos en las estructuras como se muestra en la figura No. 1, según corresponda. **El archivo CSV debe leerse una sola vez**, en todo el proyecto. Los espacios de memoria asignados para todos los datos deben ser **dinámicos y por incrementos de 5 en 5** (salvo para las cadenas de caracteres que deben ser exactas). **De emplearse otro método de asignación de memoria NO se asignará puntaje en esta pregunta.** Luego de colocar los datos en los arreglos deberá **ordenar los arreglos de manera ascendente por la fecha del pedido**, puede emplear cualquier método de ordenación, pero éste deberá tener obligatoriamente una eficiencia $n \cdot log\ n$ .

La función ***"pruebaDeCargaDePedidos"*** deben emitir un reporte que pruebe, de manera clara, bien alineada y con encabezados adecuados encima de cada columna, la carga correcta de los datos.

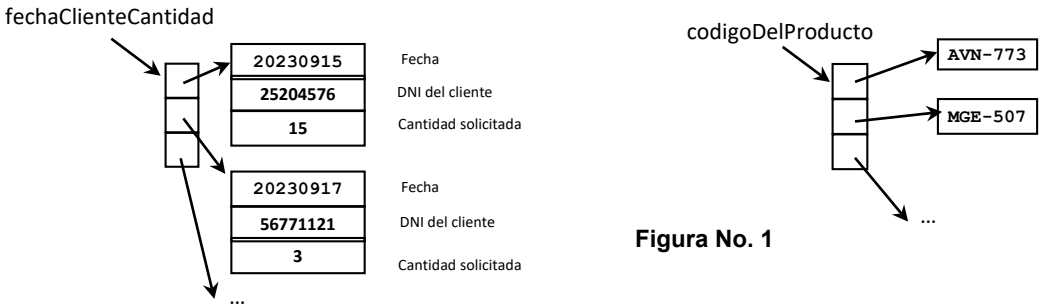


**Figura No. 1**

**NOTA A TENER EN CUENTA**:

Se le entregará una biblioteca estática con la solución de esta pregunta, usted tendrá la opción de desarrollar esta pregunta o simplemente emplear la solución dada para solucionar las siguientes preguntas del examen, en este caso no se le calificará esta pregunta.

## PREGUNTA 2 (6 puntos) PUNTEROS GENÉRICOS

Elabore un proyecto denominado "Pregunta02Examen01PunterosGenericos" y desarrolle el programa que dé solución al problema planteado. DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 2 PUNTOS DE LA NOTA FINAL. **Esta pregunta no se calificará si el desarrollo lo realiza en el proyecto de la pregunta 1.**

Con esta información, la función "main" del proyecto estará compuesto por el siguiente código:

```
#include " Pregunta02Examen01PunterosGenericos.h"

int main(int argc, char** argv) {
void *productos;
  cargarProductos (productos, "Productos.csv");
   pruebaDeCargaDeProductos (productos);

   return 0;
}
```

**NO PUEDE CAMBIAR ESTE CÓDIGO**

Implemente las funciones ***cargarProductos*** y ***pruebaDeCargaDeProductos,*** la primera debe leer los datos del archivo ***"Productos.csv"***, y colocarlos en las estructuras como se muestra en la figura No. 2, el archivo solo puede leerse una vez en todo el programa. Los espacios de memoria asignados para todos los datos deben ser **dinámicos y por incrementos de 5 en 5** (salvo para las cadenas de caracteres que deben ser exactas). **De emplearse otro método de asignación de memoria NO se asignará puntaje en esta pregunta.** Luego de colocar los datos en los arreglos deberá **ordenar el arreglo de manera ascendente por el código del producto**, debe emplear la función **qsort** de **cstdlib**.

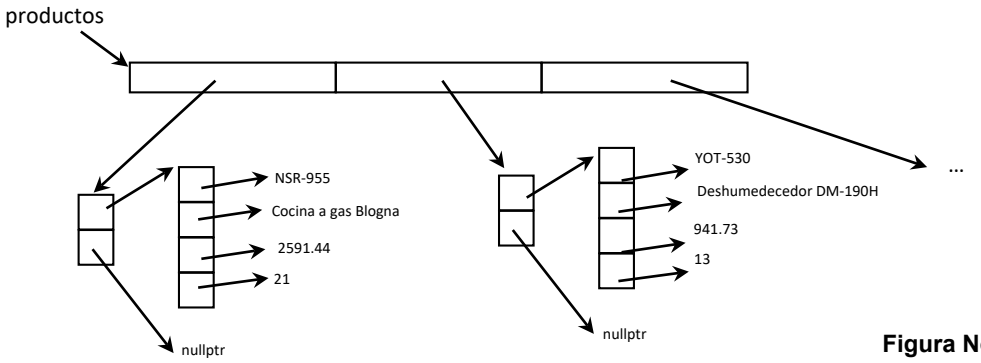


**Figura No. 2**

La función ***pruebaDeCargaDeProductos*** deben emitir un reporte que pruebe, de manera clara, bien alineada y con encabezados adecuados encima de cada columna, la carga correcta de los datos.

---

**PREGUNTA 3 (8 puntos)** **PUNTEROS A FUNCIÓN**

Se pide que desarrolle un proyecto denominado "Pregunta03Examen01PunterosAFuncion", en él incorpore las bibliotecas empleadas en las preguntas 1 y 2 **(recuerde que si desarrolló la pregunta 1 no debe emplear la solución dada para esa pregunta)**. Desarrolle en el proyecto el programa que dé solución al problema planteado. DE NO COLOCAR ESTOS REQUERIMIENTOS SE LE DESCONTARÁ 2 PUNTOS DE LA NOTA FINAL.

Se tienen los archivos que se describen a continuación:

| Numeros.txt |
|---|
| 3 |
| 4 |
| 1 |
| 9 |
| ... |

La función "main" del proyecto estará compuesto por el siguiente código:

```
#include "Pregunta01_Funciones_de_los_pedidos.h"
#include "Pregunta02_Funciones_de_los_productos.h"
#include "ColaConEnteros.h"
#include "ColaConProductos.h"
#include "ColaGenerica.h"
int main(int argc, char** argv) {
   char **codigoDelProducto;
   int **fechaClienteCantidad;
   void *productos;
   void *cola;
   cargarPedidos (fechaClienteCantidad, codigoDelProducto, "Pedidos.csv");
   pruebaDeCargaDePedidos (fechaClienteCantidad, codigoDelProducto, "PruebaPedidos.txt");
   cargarProductos (productos, "Productos.csv");
   pruebaDeCargaDeProductos (productos);
   cargaEnteros(cola, leenumero,"numeros.txt");
   mostrarEnteros(cola,imprimenumero,"reporte.txt");
   procesarPedidos(fechaClienteCantidad, codigoDelProducto, productos);
   reporteDePedidos(productos,"reportefinal.txt");
return 0;
}
```

**NO PUEDE CAMBIAR ESTE CÓDIGO**

Desarrolle la biblioteca **ColaGenerica**, que brinde soporte a la **cola** que aparece en el main, con las funciones necesarias para su soporte, como son las funciones ***generacola***, ***encola***, ***desencola*** y ***colavacia***. La función ***generacola*** es una función que se encarga de crear la estructura que representa a la cola. Como se ve en la imagen, la cola está formada por 3 direcciones de memoria, la primera apunta a la cabeza, la segunda a la cola y la tercera lleva el conteo de cuantos elementos tiene la cola.

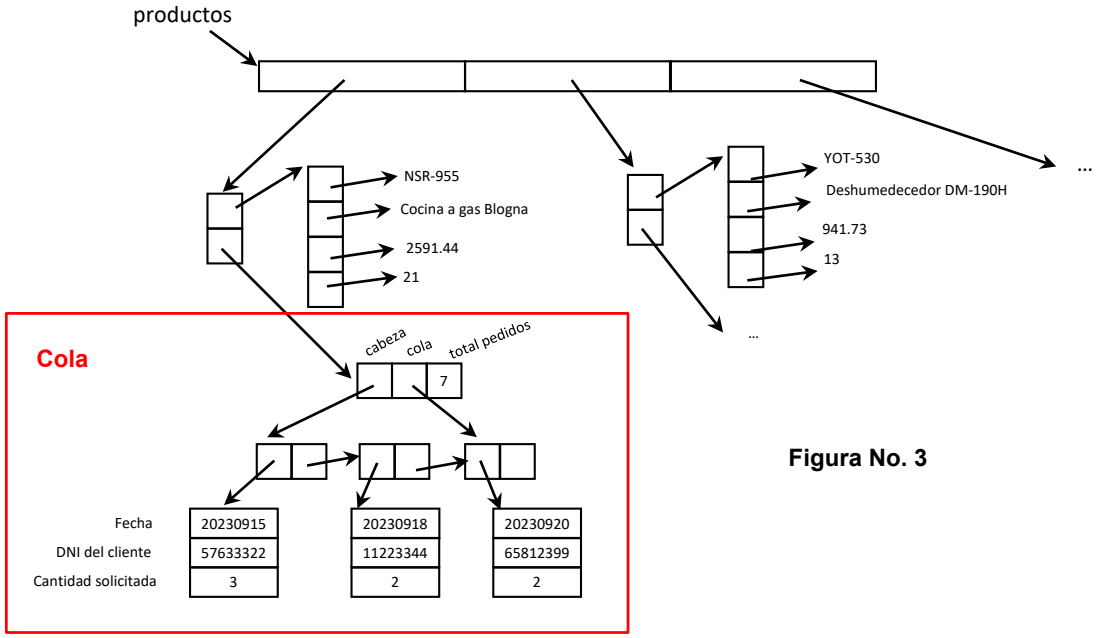


**Figura No. 3**

a. **(3 puntos)** Implemente la función ***cargaEnteros***, que lea un archivo de texto y utilizando las funciones de la biblioteca **ColaGenerica** construya la cola genérica, en este caso compuesta por números enteros. Para las operaciones con tipo debe usar la biblioteca **ColaConEnteros**. Para probar la implementación de la cola desarrolle la función ***mostrarEnteros***, esta función desencola un elemento de la cola y lo imprime en un archivo de texto.

b. **(3 puntos)** Implemente la función ***procesarPedidos***, que se encargará de leer un pedido contenido en las estructuras de la pregunta 1, para luego buscar el producto que le corresponde en el arreglo **productos**. Con la posición determinada verificar si el pedido se puede atender con el stock existente. Si en caso se puede atender, se debe disminuir el stock del producto y encolar el pedido realizado. Si no se puede atender, no considerar el pedido realizado y pasar al siguiente. Recuerde que solo se puede atender pedidos completos.

c. **(2 puntos)** Implemente la función ***reporteDePedidos***, que se encargará de leer el arreglo **productos** y mostrar su contenido, para realizar esta operación debe desencolar cada pedido de la cola e imprimirlo en un archivo de texto. El diseño se muestra a continuación:

```
Producto 1:
ADX-669   Refrigeradora cap. 220 litros
Precio: 1239.38        Stock:11
Pedidos aceptados:
Fecha      DNI          Cantidad
20230901   96659352         4

Producto 2:
AFL-718   Vinera
Precio:  1664.97       Stock:0
Pedidos aceptados:
Fecha      DNI          Cantidad
20230824   26976877         5
20230825   49087435         1
20230826   78922174         5
20230826   77484838         1
20230830   50365593         5
…
```

**NOTAS IMPORTANTES**:

- En ningún caso se permitirá desarrollar dos preguntas en el mismo proyecto. De hacerlo no se calificará la segunda y/o tercera pregunta.
- Las marcas de fin de datos solo podrán ser cero o nulo.
- Toda tarea de búsqueda debe ser desarrollada en una función independiente. Toda función de búsqueda debe prever que el dato buscado no se encuentre en la estructura empleada.

Al finalizar la práctica, 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**.

Profesor del curso: Rony Cueva
Erasmo Gómez
Miguel Guanira

San Miguel, 10 de octubre del 2023.