

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

**LENGUAJES DE PROGRAMACIÓN 1**

**1er. Examen**

**(Primer Semestre 2022)**

**Indicaciones Generales:**

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

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. 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. 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 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` 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. **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` y el código contenido en él solo podrá estar conformado por tareas implementadas como funciones. NO SE CALIFICARÁN AQUELLAS FUNCIONES DESARROLLADAS EN EL MISMO ARCHIVO QUE LA FUNCIÓN `main`.
- En cada archivo que implemente en los proyectos (`.h` y `.cpp`) deberá colocar un comentario en el que consigne claramente su nombre y código, de no hacerlo se le **descontará 0.5 puntos por 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 muestren resultados o que estos no sean coherentes en base al 60%.
- Se tomará en cuenta en la calificación el uso de comentarios relevantes.
- **NO SE PUEDEN EMPLEAR ARREGLOS ESTÁTICOS DE MÁS DE UNA DIMENSIÓN**
- **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, SI SE COLOCAN EN EL PROYECTO FUNCIONES QUE NADA TENGAN QUE VER CON LA SOLUCIÓN DEL PROBLEMA.

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 AL EXAMEN 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 "**LP1\_2022-1\_EXAMEN01**" 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.**

**PREGUNTA (10 puntos)**

**PUNTEROS MÚLTIPLES**

Elabore un proyecto denominado "**Pregunta01PunterosMultiples**" 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 dos archivos del tipo CSV, los cuales se describen a continuación:

ClientesPreg01.csv	
71984468,IPARRAGUIRRE VILLEGAS NICOLAS EDILBERTO B,935441620	
29847168,ALDAVE ZEVALLOS ROSARIO,6261108	
.....	
DNI, Nombre Categoría, Teléfono	
UN CLIENTE PUEDE NO TENER CATEGORÍA	

PedidosPreg01.csv	
412041,TORTILLAS DE MAIZ 1KG,7.91%,1.78,14.93,15.95,71378466,26/10/2020	
872819,GELATINA DE LECHE VAINILLA DANETTE 100GR,5.54,5.13,8.19,32821689,19/04/2020	
...	
Código,Descripción,Descuento(si lo tiene),cantidad,precio unitario,stock,DNI del cliente,Fecha del pedido.	

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

```
#include "ClientesPreg1.h"
#include "FuncionesExamen01Pregunta01.h"

int main(int argc, char** argv) {
    char ***cli_NombreCategoria, **pro_Descripcion;
    int ***cli_DniTelPed, **pro_Informacion, **ped_Todo;

    CargaDeClientes(cli_DniTelPed,cli_NombreCategoria,"ClientesPreg01.csv");
    PruebaDeClientes(cli_DniTelPed,cli_NombreCategoria,"PruebaClientes01.txt");

    CargarProductosPedidos(pro_Informacion,pro_Descripcion,ped_Todo,"PedidosPreg01.csv");
    PruebaProductosPedidos(pro_Informacion,pro_Descripcion,ped_Todo,"PruebaProductosPedidos01.txt");

    ordenarPedidos(ped_Todo);
    PruebaProductosPedidos(pro_Informacion,pro_Descripcion,ped_Todo,"PruebaProductosPedidos02.txt");

    asignarPedidos(cli_DniTelPed,pro_Informacion,ped_Todo);
    PruebaProductosPedidos(pro_Informacion,pro_Descripcion,ped_Todo,"PruebaProductosPedidos03.txt");
    PruebaDeClientes(cli_DniTelPed,cli_NombreCategoria,"PruebaClientes02.txt");
    return 0;
}
```

**NO PUEDE CAMBIAR ESTE CÓDIGO**

Se le ha colocado en Paideia la biblioteca "libclientespreg1.a", "ClientesPreg1.h", allí encontrará las funciones que se requieren para leer el archivo "Clientes.csv" y generar las estructuras que se muestran en la figura 1.

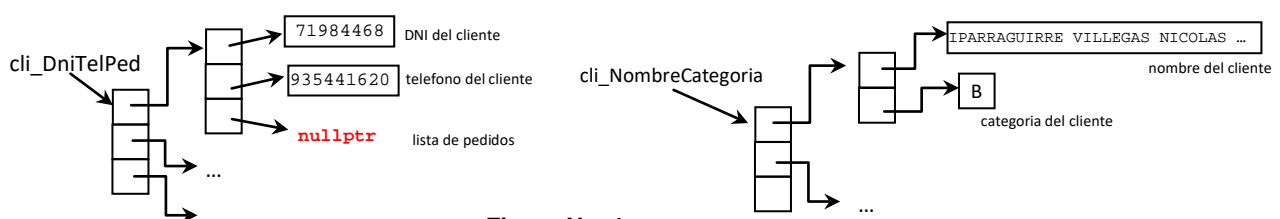


Figura No. 1

El proyecto deberá enlazar las bibliotecas de modo que su proyecto pueda ejecutar las funciones **CargaDeClientes** y **PruebaDeClientes**. Si no puede enlazar se descontarán 2 puntos de la nota final de la pregunta.

#### **PARTE A (4 puntos) NO PUEDE UTILIZAR VARIABLES CON DOS O MÁS ÍNDICES**

Implemente las funciones **CargarProductosPedidos** y **PruebaProductosPedidos**, la primera debe leer los datos del archivo **PedidosPreg01.csv**, y colocarlos en las estructuras como se muestra en la figura No. 2, según corresponda. Los productos **NO** deben estar repetidos los arreglos `pro_`, los pedidos se colocarán en el arreglo `ped_Todo` en el mismo orden en que aparecen en el archivo, los valores de punto flotante se colocarán como enteros multiplicados por 100. **El archivo CSV debe leerse una sola vez**, en todo el programa. Los espacios de memoria asignados para todos los datos deben ser **DINÁMICOS Y EXACTOS**.

La función **PruebaProductosPedidos** debe emitir un reporte que pruebe, de manera clara y bien alineada, la carga correcta de los datos.

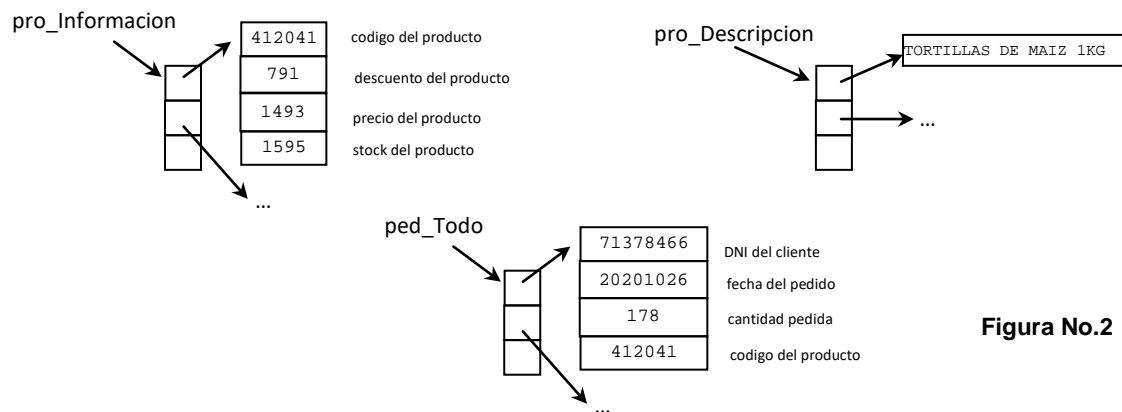


Figura No.2

## PARTE B (6 puntos) NO PUEDE UTILIZAR VARIABLES CON DOS O MÁS ÍNDICES

Implemente las funciones **ordenarPedidos** y **asignarPedidos**, la primera deberá ordenar ascendentemente el arreglo **ped\_Todo** por la fecha del pedido y si dos pedidos tienen la misma fecha, por el DNI. La ordenación deberá realizarse mediante QuickSort u otra similar con eficiencia  $n \log n$ . La función **asignarPedidos**, luego de ordenar los pedidos, deberá asignar los pedidos a cada cliente, descontando la cantidad pedida del stock de productos, si en el stock no hay suficiente para satisfacer la cantidad pedida no se asignará el pedido. El stock no puede ser negativo. La figura 3 muestra cómo quedará el arreglo **cli\_DniTelPed**. Los espacios de memoria asignados deberán ser **dinámicos y por incrementos de 5 en 5 pedidos**.

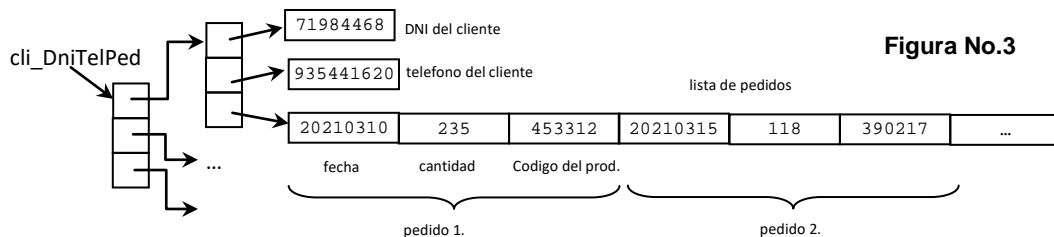


Figura No.3

## PREGUNTA 2 (10 puntos) PUNTEROS GENÉRICOS

Se pide que desarrolle un proyecto denominado **"Pregunta02PuntGenericos"** 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.** NO PUEDE UTILIZAR VARIABLES CON DOS O MÁS ÍNDICES.

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

Cientes2.csv
71984468,IPARRAGUIRRE VILLEGAS NICOLAS EDILBERTO
29847168,ALDAVE ZEVALLOS ROSARIO
.....
DNI,Nombre del cliente

Productos2.csv
459032,GELATINA DANY LIMON 125GR,5.38,24
422763,GALLETAS SUAVICREMAS FRESA 158GR,13.85,14
.....
Código,Descripción, precio, cantidad

Pedidos2.csv
422763,GALLETAS SUAVICREMAS FRESA 158GR,2,17913944,23/09/2019
675770,JARABE DE MAMEY 1.25KG,5,17913944,20/10/2019
...
Código,Descripción,cantidad solicitada,DNI del cliente,Fecha del pedido.

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

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

int main(int argc, char** argv) {
    void *productos,*pedidos,*clientes;

    cargaproductos(productos);
    cargapedidos(pedidos);
    procesaclientes(productos,pedidos,clientes);
    imprimereporte(clientes);
    return 0;
}
```

NO PUEDE  
CAMBIAR  
ESTE CÓDIGO

## PARTE A (2 puntos)

Implemente las funciones **cargaproductos** y **cargapedidos**, que lea los productos y los pedidos, desde el archivo **productos2.csv**, y coloque los datos en la estructura **productos**, luego debe leer el archivo **pedidos2.csv** y coloque los datos en la estructura **pedidos**, representadas en la figura No. 2 según corresponda. Inicialmente el **estado** de todos los pedidos es "N". Los espacios de memoria asignados deberán ser **dinámicos y exactos**. Utilice la función **imprimeproductos** e **imprimepedidos**, para validar el contenido de las estructuras implementadas.

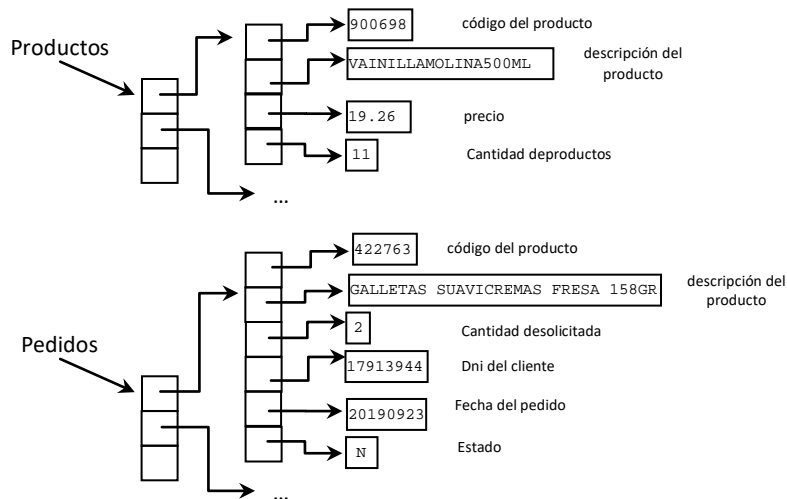


Figura No. 2

## PARTE B (6 puntos)

Implemente la función **procesaclientes**. Como primera operación actualice los estados de los pedidos de la estructura **pedidos**, (colocando "A" si se lograron atender) que es posible atender de acuerdo con la **cantidad de productos** de la estructura **productos**, **solo se pueden atender pedidos completos no parciales**. Como segunda operación lea el dni y el nombre del cliente, desde el archivo **clientes2.csv**, y coloque los datos en la estructura **clientes**, además empleando la estructura **pedidos**, cargue todos los **pedidos que es posible atender para el cliente** (puede recorrer muchas veces la estructura **pedidos**), de acuerdo con lo representado en la figura No. 3. Los espacios de memoria asignados deberán ser **dinámicos y exactos**. Utilice la función **imprimerepfinal** para validar el contenido de la estructura implementada.

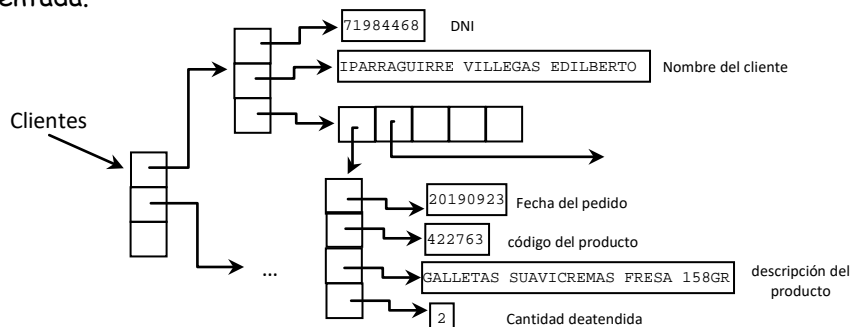


Figura No. 3

### PARTE C (2 puntos)

Esta pregunta no tendrá sentido si no implementa las preguntas a y b. Implemente la función **imprimereporte**. Esta función trabajará únicamente el arreglo. La función deberá emitir un reporte denominado **ReporteDePedidos.txt** como el que se indica a continuación:

```
=====
PEDIDOS ATENDIDOS POR CLIENTE:
=====
DNI          Nombre
71984468     IPARRAGUIRRE VILLEGAS NICOLAS EDILBERTO
=====
Pedidos atendidos:
=====
Fecha      Codigo  Descripcion del Producto      Cantidad
-----
07/08/2019  874503  GALLETAS LORS SNACKS MARINELA 75GR      1
06/10/2019  997564  MANGO ECUATORIANO                1
26/11/2020  330512  YOGHURT LALA LIGHT 240GR          6
05/03/2021  283781  FRIJOLES REFRITOS ISADORA 430GR      7
11/02/2022  651034  MANGO EN REBANADAS LA MORENA 800GR     8
=====
DNI          Nombre
29847168     ALDAVE ZEVALLOS ROSARIO
=====
Pedidos atendidos:
=====
Fecha      Codigo  Descripcion del Producto      Cantidad
-----
04/03/2020  308982  YOGHURT P BEBER FRUGGIS FRESA 150GR     11
26/06/2020  961802  ACHIOTE LA EXTRA 1KG              1
01/01/2021  580927  TORTILLAS DE HARINA DEL HOGAR 22P. 840GR  4
23/04/2021  620036  SABRITAS ORIGINAL SAL 490GR         8
02/08/2021  422763  GALLETAS SUAVICREMAS FRESA 158GR        4
02/11/2021  979865  GALLETAS MARIAS CUETARA 160GR          9
...
=====
```

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**. Luego súbalo a la tarea programa en Paideia para este laboratorio.

Profesor del curso:            Rony Cueva  
                                 Miguel Guanira

San Miguel, 17 de mayo del 2022.