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

## LENGUAJES DE PROGRAMACIÓN 1
### 1er. Examen
### (Primer Semestre 2024)

**Indicaciones Generales:**

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

**NO SE PERMITE EL USO DE APUNTES DE CLASE, FOTOCOPIAS NI MATERIAL IMPRESO**

- No se pueden emplear **variables globales, NI OBJETOS** (con excepción de los elementos de iostream, iomanip y fstream). **NO PUEDE UTILIZAR LA CLASE string**. Tampoco se podrán emplear las funciones de C que gesten memoria como malloc, realloc, memset, strdup, strtok o similares, 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 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%.
- 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.

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 **"EX01_2024_1_CO_PA_PN"** 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 tres archivos del tipo CSV, los cuales se describen a continuación:

**Archivo de libros (.csv)**

```
ETZ8565,Confesiones de una mascara,Yukio Mishima
LQL0880,Contra el fascismo,Umberto Eco
…
```

Código del libro, nombre, autor

**Archivo de clientes (CSV)**

```
36542155,Perez Rodriguez Julio Alfonso
67128011,Quispe Huaman Luisa Hilda
…
```

DNI y nombre del cliente.

**Archivo de ventas (CSV)**

```
PVZ7181,26290971,28/8/2023,65
ICX1503,27912250,5/9/2023,23
AVN3710,54602211,27/8/2023,71
…
```

Código del libro, DNI del comprador, fecha del venta, puntaje del libro (1-100).

## PUNTEROS GENÉRICOS

**PREGUNTA 1 (7 puntos)**

Elabore un proyecto denominado "PunterosGenericos2Examen01Pregunta01" 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 "PunteroVoid.h"
#include "MuestraVoid.h"

int main(int argc, char** argv) {
   void *ventas;

   cargalibros(ventas);
   cargaventas(ventas);
   muestraventas(ventas);
   return 0;
}
```

**NO PUEDE CAMBIAR ESTE CÓDIGO**

Implemente las funciones ***cargalibros y cargaventas,*** la función ***"cargalibros"*** debe llenar la estructura **ventas** de la figura 1 con los datos del archivo ***"Libros.csv"***, luego la función ***"cargaventas"*** debe leer el archivo ***"Ventas.csv"*** y actualizar la estructura **ventas**, incrementando los campos de libros vendidos cada vez que se lea un libro cargado, de la misma forma actualizando los campos de suma y promedio de puntaje, el campo ranking debe calcularse de la siguiente forma: si el puntaje es menor a 30 se considera ranking 1, si el puntaje es mayor igual a 30 y menor que 70 se considera ranking 2, si el puntaje es mayor o igual a 70 se considera ranking 3. La estructura cuenta con un arreglo con los DNI de los clientes que han comprado el libro, la misma puede crearse con un tamaño fijo sin necesidad de recortarlo. **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 exactos. **De emplearse otro método de asignación de memoria NO se asignará puntaje en esta pregunta.**

La función ***"muestraventas"*** implementada en la biblioteca MuestraVoid debe ser empleada para la impresión de prueba de la estructura. Los libros deben ser ordenados por el código del libro empleando el qsort de cstdlib.

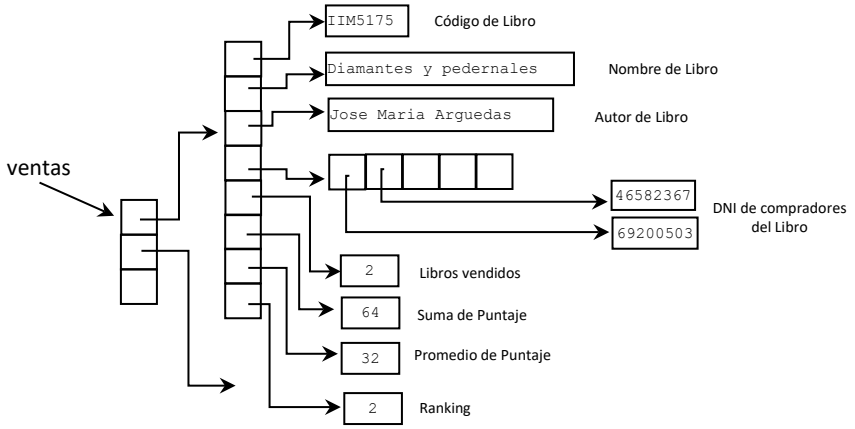


**Figura No. 1**

**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 último caso no se le calificará esta pregunta.

**PREGUNTA 2 (7 puntos)** **PUNTEROS A FUNCIÓN**

Se pide que desarrolle un proyecto denominado "PunterosAFuncion2Examen01Pregunta02", en él incorpore las bibliotecas empleadas en las preguntas 1 **(recuerde que si desarrolló la pregunta 1 no**

**debe emplear la solución dada para esta 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.

Desarrolle la biblioteca **ColaDoblementeEnlazadaGenerica**, que brinde soporte al **ranking** 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 cuántos elementos tiene la cola. Esta estructura de cola, deberá ser creada con las características de una lista doblemente enlazada, es decir que cada nodo deberá contener: un puntero al dato, un puntero al siguiente nodo y un puntero al nodo anterior. Para completar el proceso, debe desarrollar funciones como ***insertainicio*** e ***insertaenposicion***, esta última podrá recorrer la cola para insertar un nodo.

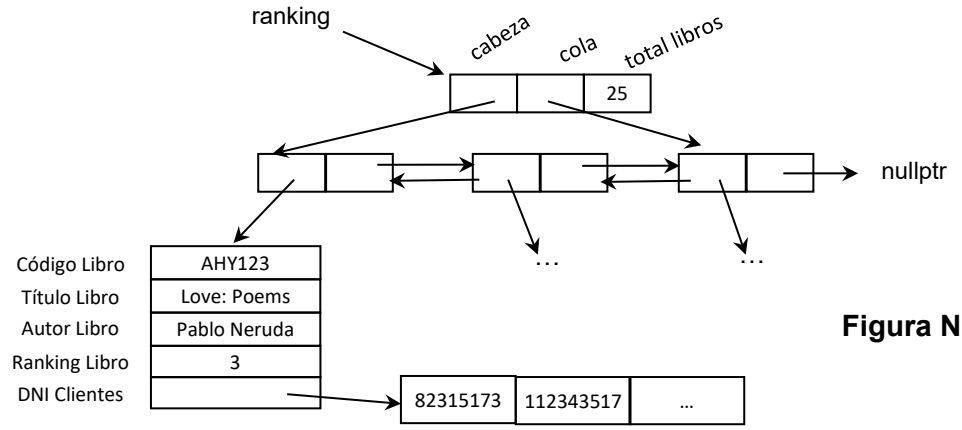


**Figura No. 2**

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

```
#include "PunteroVoid.h"
#include "MuestraVoid.h"
#include "PunterosFuncion2Examen01Pregunta03.h"
#include "ColaDoblementeEnlazada.h"
#include "Registros.h"
int main(int argc, char** argv) {
   void *ventas;
   void *ranking;

   cargalibros(ventas);
   cargaventas(ventas);
   muestraventas(ventas);
   cargarranking(ranking, crearegistro, ventas);
   muestraranking(ranking, imprimeregistro, "rankings.txt");

   return 0;
}
```

**NO PUEDE CAMBIAR ESTE CÓDIGO**

Implemente en "PunterosFuncion2Examen01Pregunta03.h":

1. **cargaranking** (5ptos): **Para este proceso deberá ubicar en la cola los libros por orden de ranking (3, 2, 1)**. Use las propiedades de la lista doblemente enlazada y la cola para poder hacer las inserciones de las prioridades. En la cabeza de la cola deberá estar ubicado el libro con mejor ranking (ranking 3). Insertar al final los libros con prioridad 1 y buscar la posición adecuada para los libros de prioridad 2. Para los libros de prioridad 2 puede recorrer la cola. Para esta pregunta deberá gestar nueva memoria para los nodos y datos
2. ***muestraranking*** (2ptos) La función ***muestraranking*** recibe el nombre del archivo donde realizará la impresión de la cola.

**PREGUNTA 3 (6 puntos)** **PUNTEROS MÚLTIPLES**

Elabore un proyecto denominado "PunterosMultiples2Examen01Pregunta03" 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.

Deberá incluir en este proyecto las bibliotecas de las preguntas anteriores. La función "main" para este proyecto estará compuesto por el siguiente código:

```
#include "PunteroVoid.h"
#include "MuestraVoid.h"
#include "PunterosFuncion2Examen01Pregunta03.h"
#include "ColaDoblementeEnlazada.h"
#include "Registros.h"
#include " Examen01PunterosMultiples1Pregunta02.h"

int main(int argc, char** argv) {
   void *ventas;
   void *ranking;
   char **clienteNombre, ***libroStr;
   int *clienteDNI, **libroInt;

   cargalibros(ventas);
   cargaventas(ventas);
   muestraventas(ventas);
   cargarranking(ranking, crearegistro, ventas);
   muestraranking(ranking, imprimeregistro, "rankings.txt");
   cargarClientas (clienteDNI, clienteNombre, "Clientes.csv");
   pruebaDeCargaDeClientes(clienteDNI, clienteNombre, "PruebaClienteas.txt");
   cargaVentas (ranking, libroStr libroInt);
   reporteDeLibros (clienteDNI, clienteNombre,libroStr,libroInt,"ReporteLibros.txt");

   return 0;
}
```

**NO PUEDE CAMBIAR ESTE CÓDIGO**

Implemente las funciones ***cargarClientes, cargarVentas, pruebaDeCargaDeClientes y pruebaDeCargaDeVentas***, la función ***"cargarClientes"*** deben leer los datos del archivo ***"Clientes.csv"*** y la función ***"cargarVentas"*** debe tomar los datos de la **ranking** (desencolando los datos), implementada en la preguntas anterior. En ambos casos deben colocar los datos en las estructuras como se muestra en la figura No. 3, 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.** Para esta pregunta deberá gestar nueva memoria para los datos

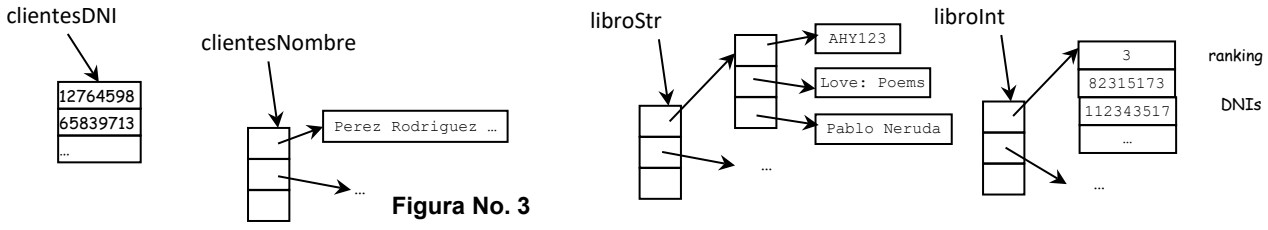


**Figura No. 3**

La función ***"pruebaDeCargaDeClientes"*** 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. Los datos de un cliente deben aparecer en una línea, no se permite que aparezcan como en una ficha. La función ***"reporteDeLibros"*** debe generar un reporte similar a que se muestra a continuación:

```
No.  CODIGO   TITULO                    AUTOR           RANKING
01)  AHY1234  Love: Poems                Pablo Neruda       3
     Compradores:  Rojas Jimenez Juan Pedro
                   Turpan Chala Maria
                   …
02)  XRQ1234  …                         …                  …
```

Los datos deben aparecer muy bien alineados.

**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.
- NO PODRÁ EMPLEAR ARREGLOS DE MÁS DE UNA DIMENSIÓN.
- NO PUEDE MANIPULAR UN PUNTERO CON MÁS DE UN ÍNDICE.
-

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**.

Profesores del curso: Rony Cueva, Erick Huiza, Miguel Guanira, Erasmo Gómez, Heider Sánchez

San Miguel, 14 de mayo del 2024.