

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

1er. Examen

(Segundo Semestre 2022)

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`, `omanip` 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` de cada proyecto y el código contenido en él solo podrá estar conformado por tareas implementadas como funciones. 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**.
- 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 muestren 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

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_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.**

PREGUNTA (10 puntos)

PUNTEROS MÚLTIPLES

Elabore un proyecto denominado **"Examen01PunterosMultiples"** 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:

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.

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

```
#include "FuncionesConsultasMedicasPreg1.h"

int main(int argc, char** argv) {
    int *med_Codigo, ***consultas_int;
    double *med_Precio;
    char **med_Descripcion, ***consultas_char;

    CargarMedicamentos (med_Codigo, med_Descripcion, med_Precio, "Medicinas-Preg01.csv");
    PruebaDeCargaDeMedicamentos (med_Codigo, med_Descripcion, med_Precio, "PruebaMedicinas.txt");

    CargarConsultas (consultas_int, consultas_char, "Consultas-Preg01.csv");
    PruebaDeCargaDeConsultas (consultas_int, consultas_char, "PruebaConsultas.txt");

    ReporteDeGastos_e_IngresosMedicos (med_Codigo, med_Descripcion, med_Precio, consultas_int,
                                         consultas_char, "Reporte-Gastos-Ingresos.txt");

    return 0;
}
```

NO PUEDE CAMBIAR ESTE CÓDIGO

PARTE A (2 puntos) NO PUEDE UTILIZAR VARIABLES CON DOS O MÁS ÍNDICES

Implemente las funciones *CargarMedicamentos* y *PruebaDeCargaDeMedicamentos*, la primera debe leer los datos del archivo *Medicinas-Preg01.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 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.**

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

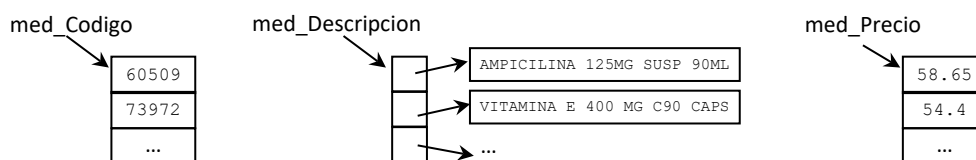


Figura No. 1

PARTE B (5 puntos) NO PUEDE UTILIZAR VARIABLES CON DOS O MÁS ÍNDICES

Implemente las funciones *CargarConsultas* y *PruebaDeCargaDeConsultas*, la primera debe leer los datos del archivo *Consultas-Preg01.csv*, y colocarlos en las estructuras como se muestra en la figura No. 2, según corresponda (el archivo solo puede leerse una vez en todo el programa). 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.**

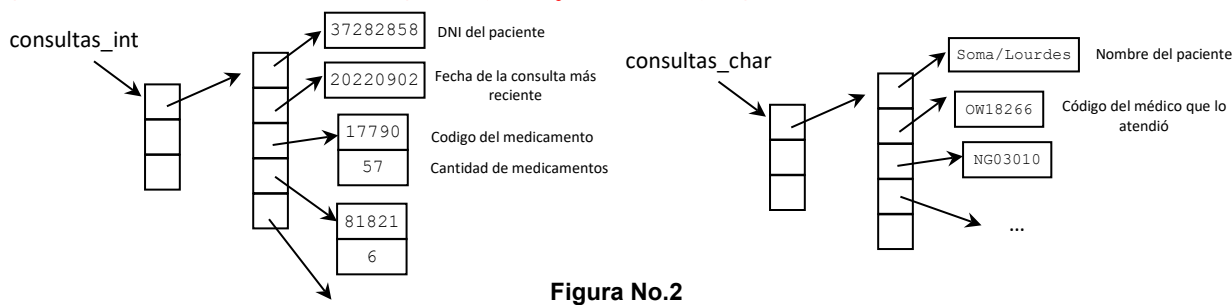


Figura No.2

Las estructuras muestran la lista de pacientes que fueron atendidos en un consulta médica. En ellos, no deben aparecer en ningún momento pacientes repetidos. Por cada paciente se guarda la información de sus consultas, en el arreglo *consultas_int* se guarda el DNI del paciente, la fecha de la consulta más

cercana y la relación de medicamentos de medicamentos que recibió en todas sus consultas (código y cantidad). Los medicamentos no pueden repetirse para un paciente dado, las cantidades se acumularán. En el arreglo `consultas_char` se debe colocar, por cada paciente, su nombre, y la lista de médicos (códigos) que lo atendieron. Los códigos de los médicos no deben repetirse para un paciente dado. La función `PruebaDeCargaDeConsultas` debe emitir un reporte que pruebe, de manera clara y bien alineada, la carga correcta de todos los datos.

PARTE B (3 puntos) NO PUEDE UTILIZAR VARIABLES CON DOS O MÁS ÍNDICES

Implemente la función `ReporteDeGastos_e_IngresosMedicos`, la cual deberá mostrar un reporte similar al que se muestra a continuación:

```
=====
GASTOS MEDICOS POR PACIENTE
=====
DNI          NOMBRE          ULTIMA CONSULTA    MEDICOS QUE LO ATENDIERON
372828589    Soma/Lourdes      02/09/2023        OW18266  NG03010 ...
-----
MEDICAMENTOS QUE RECIBIO:
-----
CODIGO      DESCRIPCION          PRECIO      CANTIDAD      TOTAL
17790      DICLOFENACO 100MG 20T    17.00        57          969.00
81821      AMIODARONA 200MG T20    61.20         6          367.20
...        ...                    ...          ...          ...
-----
Total gastado:                                6783.56
=====
DNI          NOMBRE          ULTIMA CONSULTA    MEDICOS QUE LO ATENDIERON
...        ...                    ...          ...          ...
=====
TOTAL DE INGRESOS POR MEDICINAS:                876554.98
=====
```

PREGUNTA 2 (10 puntos)

PUNTEROS GENÉRICOS

Se pide que desarrolle un proyecto denominado `"EX01_PREG02_PuntGenericos"` 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 dos archivos del tipo CSV, los cuales se describen a continuación:

Medicinas.csv
60509,AMPICILINA 125MG SUSP 90ML,58.65,37
73972,VITAMINA E 400 MG C90 CAPS,54.4,45
.....
Código, Descripción, precio, stock

Consultas.csv
LZ53556,32538825,Hijar/Martin,1/09/2022,4,92560,11,16681,35,47697,7,26506,14
LZ53557,85395219,Fuentes/Patricia,29/12/2022,5,30428,31,33311,15,57505,23,52072,15,22232,30
.....
Código med.,DNI,Nom Paciente,Fecha,Cant. de medicinas, Cód Medicina, Cant. Solicitada
Los dos últimos campos se pueden repetir de acuerdo a la cantidad de medicinas.

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

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

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

    cargamedicinas(medicinas);
    cargaconsultas(citas);
    actualizareservas(medicinas,citas);
    imprimerepfinal(citas);
    return 0;
}
```

**NO PUEDE
CAMBIAR
ESTE CÓDIGO**

- a. (2 puntos) Implemente las funciones **cargamedicinas** que lea los medicamentos, desde el archivo **medicinas.csv**, y coloque los datos en la estructura **medicinas**, representadas en la figura No. 3 según corresponda. Inicialmente la cantidad de **reserva** es 0. Los espacios de memoria asignados deberán ser **dinámicos y exactos**. Utilice la función **imprimemedicinas**, para validar el contenido de las estructuras implementadas.

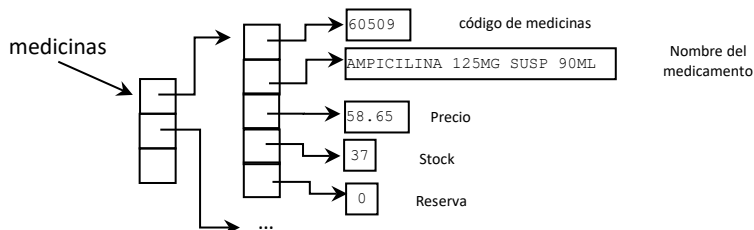


Figura No. 3

- b. (3 puntos) Implemente las funciones **cargaconsultas**, que lea las citas, desde el archivo **Consultas.csv**, y coloque los datos en la estructura **citas**, representadas en la figura No. 4 según corresponda. Inicialmente el **estado** de todas las consultas está en "S". Los espacios de memoria asignados deberán ser **dinámicos y exactos**. Utilice la función **imprimeconsultas** el contenido de las estructuras implementadas.

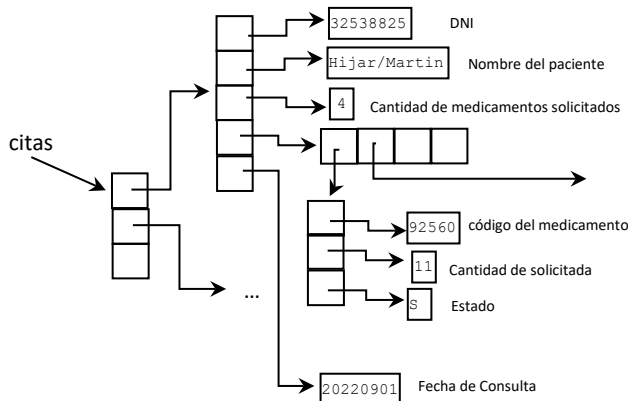


Figura No. 4

- c. (5 puntos)) Implemente la función **actualizareservas**, que se encargue de recorrer la estructura de **citas** y actualice los medicamentos recetados en una consulta que pueden ser reservados de acuerdo con el stock y la cantidad reservada que se muestra en la estructura **medicinas**, si es posible se debe cambiar el estado de "S" a "R". Además, la cantidad reservada debe actualizarse utilizando el campo Reserva de la estructura **medicinas**. No pueden reservarse cantidades parciales. Utilice la función **imprimerepfinal** para validar el contenido de la estructura implementada.

Además, se cuenta con una biblioteca estática "**FuncionesAuxiliares**" donde están desarrollados los reportes de validación, el reporte final y una función de ordenamiento, que puede utilizar si lo considera necesario.

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.