

**TÉCNICAS DE PROGRAMACIÓN**  
OCTAVO LABORATORIO CALIFICADO  
SEMESTRE ACADÉMICO 2024-2

Horarios: Todos

Duración: 110 minutos  
Elaborado por los profesores del curso.

**ADVERTENCIAS:**

- Todo dispositivo electrónico (teléfono, tableta, computadora u otro) deberá permanecer apagado durante la evaluación **en su mochila**.
- Coloque todo aquello que no sean útiles de uso autorizado durante la evaluación en la parte delantera del aula, por ejemplo, mochila, maletín, cartera o similar, y procure que contenga todas sus propiedades. La apropiada identificación de las pertenencias es su responsabilidad.
- Si se detecta omisión a los dos puntos anteriores, la evaluación será considerada nula y podrá conllevar el inicio de un procedimiento disciplinario en determinados casos.
- Es su responsabilidad tomar las precauciones necesarias para no requerir la utilización de servicios higiénicos: durante la evaluación, no podrá acceder a ellos, de tener alguna emergencia comunicárselo a su jefe de práctica.
- Quienes deseen retirarse del aula y dar por concluida su evaluación no lo podrán hacer dentro de la primera mitad del tiempo de duración destinado a ella.

**INDICACIONES:**

- No se pueden usar apuntes de clase ni calculadoras.
- Está prohibido el acceso a Internet y a correo electrónico hasta que lo indiquen los jefes de práctica, tampoco podrá emplear dispositivos USB.
- Si se detecta omisión al punto anterior, la evaluación será considerada nula y podrá conllevar el inicio de un procedimiento disciplinario en determinados casos.
- NO SE PUEDEN EMPLEAR ARCHIVOS DE DATOS AUXILIARES NI VARIABLES GLOBALES. NO podrá implementar funciones en el archivo main.cpp, las funciones se deberán implementar en archivos independientes (.h y .cpp).
- En la calificación se tomará en cuenta el buen uso de los nombres de los identificadores, y el eficaz uso de comentarios.
- **DEBE COLOCAR SU NOMBRE Y CÓDIGO EN EL ARCHIVO main.cpp DE SU PROYECTO, DE LO CONTRARIO SE LE DESCOTARÁ 0.5 PUNTOS EN SU NOTA FINAL. NO SE HARÁN EXCEPCIONES.**
- **DEBE COLOCAR UN COMENTARIO AL INICIO DEL ARCHIVO main.cpp UNA DESCRIPCIÓN DE LO QUE HACE EL PROGRAMA. ESTA DESCRIPCIÓN NO DEBE SER GENÉRICA, DEBE SER ESPECÍFICA DE LO QUE HARÁ EL PROGRAMA. SE LE DESCOTARÁ 0.5 PUNTOS EN SU NOTA FINAL SI NO SE COLOCA ESTE COMENTARIO O LO QUE SE EXPRESE EN ÉL NO SEA ESPECÍFICO. NO SE HARÁN EXCEPCIONES.**
- NO PUEDE TENER EN SU NetBeans OTROS PROYECTOS ABIERTOS AL INDICADO EN ESTE LABORATORIO
- NO PODRÁ EMPLEAR LAS BIBLIOTECAS <stdio.h>, <cstdlib> <string> ni <string.h>. NO CONFUNDIR CON <cstring>

La finalidad de este laboratorio es la de reforzar los conceptos de estructuras

Cree un proyecto en NetBeans con el nombre: “Laboratorio\_08-2024-2 y en él desarrolle el programa que resuelva el problema que se describe a continuación.

**DEBE LEER TODA LA PRUEBA ANTES DE EMPEZAR A DESARROLLAR EL PROGRAMA  
INCLUYENDO LAS INDICACIONES Y ADVERTENCIAS**

Una empresa de servicios de entrega a domicilio de platos de comida desea contar con una aplicación que le permita procesar los platos enviados a sus clientes. La empresa cuenta con varios archivos de textos, los cuales son similares a los que se muestran a continuación:

**Archivo: PlatosOfrecidos.csv**

```
AP-73428,TEQUENOS,19.50,APERITIVO
PO-11283,2 BOLAS DE HELADO,14.90,POSTRES
...f...f...f...
```

En el archivo “PlatosOfrecidos.csv” se ha colocado en cada línea la descripción de un plato que ofrece la empresa. Un plato estará dado por su código, nombre, precio y categoría.

**Archivo: DistritosDeLima.csv**

```
LN3085,Ancon,Norte
LE3053,Ate Vitarte,Este
...f...f...
```

En el archivo “DistritosDeLima.csv” se ha colocado en cada línea la información de un distrito de la ciudad de Lima. En cada línea aparece el código, nombre y ubicación de un distrito.

**Archivo: Clientes.csv**

```
63640178,CABRERA CANALES GUILLERMO EDRIC,LO4439
66039166,LAURA WONG MELVIN HENRRY,LO3562
...f...f...f...
```

En el archivo “**Clientes.csv**” se ha colocado en cada línea la descripción de un cliente registrado en la empresa. Un cliente estará dado por su DNI, nombre y el código del distrito donde vive.

**Archivo: PedidosRealizados.txt**

```
211001      86988155      BE-39695      6
261051      36471787      BR-15665      5
...          ...          ...          ...
```

En el archivo “**PedidosRealizados.txt**”, se describen todos los pedidos que se deben atender durante la jornada, en cada línea del archivo se ha colocado solo el código de uno de los platos que conforman el pedido, por esta razón un pedido completo aparecerá en varias líneas del archivo. El archivo no está ordenado, por lo que un pedido completo no está colocado en el archivo en líneas consecutivas. En cada línea se ha colocado el código del pedido, el DNI del cliente que hizo el pedido, el código del plato pedido y la cantidad pedida. Debido al formato del archivo, el código y DNI del pedido pueden aparecer varias veces en el archivo.

Se le pide que desarrolle un proyecto en C++, en el que, empleando los archivos, **genere un reporte en un archivo de textos** similar a los que se muestran a continuación:

Reportel: **Pedidos\_Atendidos.txt**

EMPRESA DE REPARTOS A DOMICILIO TP S. A.  
RELACION DE PEDIDOS POR CLIENTE

```
=====
CIENTE No. 001
DNI      NOMBRE      DISTRITO      PAGO TOTAL
72330000 MEDINA VELASQUEZ MILENA JENNY LN4803 Carabayllo 1764.38
=====
PEDIDOS REALIZADOS:
NO. PEDIDO CODIGO DESCRIPCION CATEGORIA PRECIO CANTIDAD SUBTOTAL
001 181777 AP-73428 TEQUEÑOS APERITIVO 19.50 3 58.50
...      ...      ...      ...      ...      ...      ...
=====
CIENTE No. 002
DNI      NOMBRE      DISTRITO      PAGO TOTAL
...      ...      ...      ...
=====
```

En el proyecto y en la función main se deben determinar por las siguientes tareas, **las cuales debe desarrollarlas en el orden que se indican y cada una en una función o archivo .h independientes**:

- Definir la siguientes estructuras:
  - **Distrito**: que debe contener los siguiente campos: 1) **codigo** (cadena de caracteres dinámica - char \*) 2) **nombre** (cadena de caracteres dinámica - char \*) y **ubicacion** (cadena de caracteres dinámica - char \*).
  - **Plato**: que debe contener los siguientes campos: 1) **codigo** (cadena de caracteres dinámica - char \*), 2) **descripcion** (cadena de caracteres dinámica - char \*), 3) **precio** (valor de punto flotante), 4) **categoria** (cadena de caracteres dinámica - char \*) y 5) **cantidad** (valor entero).
  - **PlatoPedido** que debe contener los siguientes campos: 1) **codigoPedido** (valor entero), 2) **codigoPlato** (cadena de caracteres dinámica - char \*), 3) **descripcion** (cadena de caracteres dinámica - char \*), 4) **cantidad** (valor entero), 5) **precio** (valor de punto flotante) y 6) **subtotal** (valor de punto flotante).
  - **Cliente** que debe contener los siguientes campos: 1) **dni** (valor entero), 2) **nombre** (cadena de caracteres dinámica - char \*), 3) **distrito** (estructura de tipo **Distrito**), 4) **pedidos** (arreglo estático de estructuras de tipo **PlatoPedido**), 5) **cantidadDePedidos** (valor entero) y 6) **montoTotal** ((valor de punto flotante)).
- Definir e inicializar tres arreglos estáticos para las estructuras **distrito**, **plato** y **cliente**, considerar que los platos pedidos en total de un cliente no sobrepasan de 30.
- En una función por archivo, lea los datos de los archivos *PlatosOfrecidos.csv*, *DistritosDeLima.csv* y *Clientes.csv* y colóquelos en los arreglos definidos. **Los archivos solo pueden leerse una vez**. Los campos que no se llenen deben quedar inicializados convenientemente. En cada función, el arreglo se manejará independiente, no debe llenar los campos con valores de los otros arreglos.
- Verifique el llenado de los arreglos en reportes independientes por archivo.

- Empleando el archivo ***PedidosRealizados.txt***, complete los platos pedidos de cada cliente. Aquí debe completar todos los campos faltantes de la estructura cliente empleando también los otros arreglos.
- Ordene los clientes por su nombre de manera ascendente.
- Ordene los platos pedidos de cada cliente de manera ascendente por el código del pedido y descendientemente por el código del plato.
- Emita el reporte solicitado.
- **Toda operación de búsqueda debe realizarse en una función independiente. No se considerará en la calificación los procesos de búsqueda que estén contenidos en el código de otro proceso. Las funciones de búsqueda deben considerar la posibilidad que el dato buscado no se encuentre. LAS CADENAS DEBEN COMPARARSE COMO TAL,**
- No debe repetir código innecesariamente, tareas como la lectura y escritura de los textos deben ser realizadas en una única función que se adapte a todas las situaciones del problema.
- **Debe usar obligatoriamente los elementos de entrada y salida proporcionados por la biblioteca fstream para resolver el problema.**

**Si así lo desea, al finalizar su trabajo 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 y súbalo a la tarea programada en Paideia.**

San Miguel, 18 de noviembre del 2024