

PROGRAMACIÓN I (PUI)**TRABAJO PRÁCTICO 5: ESTRUCTURAS DE DATOS (PILAS Y COLAS)**

Al finalizar este práctico se espera que los alumnos logren:

- Relacionar conceptos.
- Realizar la búsqueda creativa de soluciones algorítmicas, si es que existen, y seleccionar la alternativa más adecuada.
- Emplear, según la situación problemática, las estructuras de datos Pilas y Colas.
- Verificar la solución algorítmica.

Actividades:

Definir en pseudocódigo las estructuras de datos y variables necesarias.

Diseñar el algoritmo de solución expresado en diagrama de flujo utilizando los módulos básicos del ejercicio1, el diagrama de jerarquía de módulos y verificar si se obtienen el/ los resultado/s esperado/s.

PARTE A**1. Realizar los métodos básicos para el manejo de la Pila que permitan:**

- Crear Pila. (llamar **CrearPila()**)
- Apilar un elemento a la Pila (llamar **Apilar()**)
- Desapilar un elemento de la Pila (llamar **Desapilar()**)
- Indicar si la Pila está vacía (llamar **PilaVacía()**)
- Indicar si la Pila está llena (llamar **PilaLlena()**)

2. Dados los siguientes problemas, se pide:

- Dada una pila A de tipo carácter, verificar si todos sus elementos están contenidos en la pila B (de igual tipo). Dejar al final la Pila A en su estado original.
- Realizar un menú que permita realizar las siguientes tareas sobre la Pila Z de números enteros.
 - ✓ Mostrar el promedio de los elementos cargados en la Pila.
 - ✓ Mostrar los elementos mayores a un número ingresado.
 - ✓ Informar si un elemento determinado está en la pila.
 - ✓ Determinar el menor elemento de la pila.

Dejar al final la Pila A en su estado original.

- Un alumno guarda las diferentes fotocopias correspondientes a las tres asignaturas que está cursando, en un cajón; y en un momento determinado sólo puede acceder al apunte que está situado en la parte superior del cajón. Para cada apunte se conoce la siguiente información: *asignatura, tema, número de páginas y precio*. Se pide:
 - ✓ Clasificar los apuntes por asignaturas en cajas separadas.
 - ✓ Una vez clasificados los apuntes mostrar los montos gastados por cada asignatura.
 - ✓ Comparando los totales de cada asignatura informar cuál es el apunte que le ocasionó el mayor gasto.

PARTE B**1- Realizar los métodos básicos para el manejo de la Cola que permitan:**

- Crear Cola (llamar **CrearCola()**)
- Agregar un elemento a la Cola (llamar **Añadir()**)
- Tomar un elemento de la Cola (llamar **Cabeza()**)
- Eliminar un elemento de la Cola (llamar **Eliminar()**)
- Indicar si la cola está vacía (llamar **ColaVacía()**)
- Indicar si la cola está llena (llamar **ColaLlena()**)

2- Dados los siguientes problemas, se pide:

- Dada una Cola **C** de N elementos, leer un número, si existe en la cola eliminarlo, caso contrario mostrar el mensaje *"Número Inexistente"*.
- Dada una Cola **A** de N elementos y la Cola **B** de M elementos (ambas colas se encuentran ordenadas), intercalar los elementos de la Cola **A** y de la Cola **B**, para generar la Cola **C** que debe quedar ordenada.
- Un Banco tiene una cola **B** con una capacidad máxima de 50 personas con los siguientes datos: *código de cliente y tipo de transacción*. Se requiere conocer el tiempo que le lleva atender a la 5ta persona, la que se encuentra en la mitad y la que se encuentra al último. Si los tiempos estimados de cada transacción son:

Tipo	Minutos	Descripción
1	5	Extracción
2	6	Deposito
3	8	Pagos

PARTE C

Ejercicio 1

Un Negocio que se dedica a la entrega de regalos empresariales desea organizar la mercadería para las cajas navideñas de fin de año. Estas se almacenan una encima de otra y tienen los datos: descripción del contenido, destinatario y cantidad de unidades pedidas. Por otro lado se cuenta con una planilla que contiene el destinatario, su dirección y teléfono.

Se pide:

- a) Definir en pseudocódigo las estructuras de datos adecuadas para almacenar la información de las estructuras **Caja, Destinatario, Stock, Planilla Entrega, y Nuevos Pedidos**
- b) Realizar el Diagrama de Flujo utilizando los principios de programación modular y estructura de los módulos que permitan:
 - 1- Confeccionar la Planilla Entrega de pedidos con todos los datos de cajas y destinatarios. Esta se genera por orden de llegada.
 - 2- Ante una consulta informar si ese destinatario se encuentra en Planilla de entrega. Caso contrario incorporarla a Nuevos pedidos con todos los datos de la Planilla (contenido y destinatario).
 - 3- Controlar si el stock es suficiente para satisfacer la planilla de entrega. Para ello Stock posee los datos Tipo de regalos (1- Genérico, 2-Caja navideña, 3-artículo Vip) y cantidad disponible. En caso contrario informar mediante un mensaje la cantidad faltante.

Ejercicio 2

Una distribuidora de bebidas recibe los pedidos en las planillas de los preventistas según los siguientes datos: código de cliente, código de producto y cantidad pedida. A medida que llegan, se depositan uno encima de otro. Cuando se realiza el procesamiento se toma el que se encuentra mas arriba y se confecciona la orden de carga para el camión de reparto según la zona del cliente y la información del producto. Por cada cliente se tienen los siguientes datos: Código de cliente. Nombre. Dirección. Zona. En productos se guarda el código, precio, descripción y cantidad existente.

Se pide:

- c) Definir en pseudocódigo las estructuras de datos adecuadas para almacenar la información de las estructuras de datos
- d) Realizar el Diagrama de Flujo utilizando los principios de programación modular y estructura de los módulos que permitan:
 - 1- Dados los pedidos de los preventistas confeccionar las órdenes de carga para el camión, cada una contiene: zona, nombre del cliente, dirección, descripción del producto y precios.
 - 2- Realice la actualización que refleje la disminución en cantidad de productos en stock. Tenga en cuenta que si la cantidad existente no alcanza a cubrir la cantidad necesaria se agrega a la estructura Pedidos insatisfechos (por estricto orden de llegada, según los siguientes datos: Código del producto, Cantidad requerida y no se carga en el camión).