

PL1 – parte 1- Estructura de datos

[Subtítulo del documento]



23 de octubre de 2023

Pablo manglano redondo 03492382Q y alberto durán vadillo 09100943G

**1. Implementar las Clases:**

* **Pedido:** Representa un pedido con un ID único, número de seguimiento, DNI del cliente y urgencia.
* **NodoPila:** Nodo para la estructura de datos de Pila.
* **NodoCola:** Nodo para la estructura de datos de Cola.
* **NodoLista:** Nodo para la estructura de datos de Lista.
* **Pila:** Implementa la Pila LIFO para almacenar pedidos.
* **Cola:** Implementa la Cola FIFO para las estaciones de empaquetado.
* **Lista:** Implementa la Lista para almacenar pedidos ordenados por prioridad.

**2. Implementar la Clase Gestor:**

* **Gestor:** Contiene métodos para gestionar la simulación del control de pedidos, como generar, almacenar, procesar y mostrar pedidos según las opciones del menú.

**3. Implementar la Función Principal (main):**

* **main:** Muestra un menú interactivo para que el usuario seleccione las opciones A-L según lo requerido.

**4. Generar Pedidos Aleatorios (Opción A):**

* Al introducir la opción A hacemos una llamada al método “gestor.generar12Pedidos();” de la clase Gestor. Este método emplea funciones de generación de números aleatorios para crear 12 pedidos y almacénalos en la Pila. Cada pedido es una estructura en la que guardamos las variables idPedido, dniCliente, numeroSeguimiento y urgente. Se generarán de 12 en 12 y serán almacenados hasta un máximo de 48 pedidos en la pilaPedidos.
* Al introducir la opción B hacemos una llamada al método “gestor.muestraPedidos();” de la clase Gestor. Este método llama al método de la clase Pila “mostrar();” el cual se encarga de ir imprimiendo por pantalla los datos de cada pedido hasta que haya mostrado todos mientras la pilaPedidos no esté vacía.
* Al introducir la opción C hacemos una llamada al método de la clase Gestor encargado de vaciar la pilaPedidos: “gestor.borrarPedidosPila();”. Este método hace una llamada al método “vaciarPila();” de la clase Pila. Este método va eliminando pedido a pedido hasta que no queda ninguno dejando la pilaPedidos vacía.
* Al introducir la opción D hacemos una llamada al método “gestor.encolarPedidos();” de la clase Gestor. Este método nos vacía la pilaPedidos y los almacena en 4 colas: estaciónA, estaciónB, estaciónC y estaciónD. Si el pedido no es urgente, comparará las longitudes de las colas A y B con el método “getLongitud()” e introducirá el pedido no urgente en la cola que menos pedidos tenga. En caso de tener la misma longitud lo almacenará en la estaciónA. Si el pedido es urgente sucede lo mismo, pero con las estaciones C y D.
* La opción E hace una llamada al método “gestor.muestraPedidosSalasAyB();” de la clase Gestor el cual hace dos llamadas (una para estacionA y otra para estacionB) al método de la clase Cola “mostrar();”. Este método muestra mientras no esté vacía todos los pedidos uno por uno de cada una de las colas para los pedidos no urgentes (A y B).
* La opción F hace una llamada al método “gestor.muestraPedidosSalasCyD();” de la clase Gestor el cual hace dos llamadas (una para estacionC y otra para estacionD) al método de la clase Cola “mostrar();”. Este método muestra mientras no esté vacía todos los pedidos uno por uno de cada una de las colas para los pedidos urgentes (C y D).

**6. Reiniciar el Programa (Opción L):**

* Implementa la opción L para reiniciar el programa a su estado inicial. Puedes lograr esto creando un método de reinicio en la clase Gestor que restablezca todas las estructuras de datos a su estado inicial.

**7. Pruebas y Debugging:**

* Después de implementar cada parte, realiza pruebas exhaustivas para asegurarte de que cada funcionalidad esté trabajando según lo esperado. Utiliza la depuración para corregir cualquier error que encuentres.

**8. Documentación y Comentarios:**

* Asegúrate de documentar tu código correctamente, incluyendo comentarios descriptivos para cada clase y método que expliquen su propósito y funcionamiento.

Dificultades encontradas:  
  
- Error “does not name a type”.