

Proyecto Final Programación I

Integrantes:

- Paula Delgadillo Arguedas.
- Axel Mauricio Rodríguez.
- Joseph Jesús Aguilar.

Programa en C++.

Desarrollado en Visual Studio Code.

Profesor: Alexander Simon Benjamin.

Universidad: UTC.

Entrega: sábado 18 de diciembre.

- **Inventario:** ¿Qué estructura permite almacenar y organizar productos para búsquedas rápidas?

Para el inventario, el uso de la lista (lista doblemente enlazada) es adecuado, ya que proporciona flexibilidad y eficiencia en un inventario dinámico. Esta estructura permite realizar inserciones y eliminaciones en cualquier posición sin necesidad de mover otros elementos, lo cual es ventajoso frente a estructuras como vectores. Además, aunque la lista no organiza automáticamente los elementos alfabéticamente, facilita recorrer los productos para ordenarlos o realizar búsquedas específicas si se implementan mecanismos adicionales.

Por otro lado, para búsquedas rápidas en un inventario grande, una estructura como un árbol binario balanceado o un mapa (hash map) sería más eficiente. Sin embargo, dado que en este programa no se requieren búsquedas complejas, la lista resulta suficiente para las necesidades del programa y es la opción más óptima según las instrucciones dadas por el profesor sobre el uso de colas, pilas y listas.

- **Solicitudes de Compra:** ¿Qué estructura permite atender solicitudes en el mismo orden en que llegan?

Para las solicitudes de compra, el uso de una cola (std::queue) es la opción ideal, ya que permite atenderlas en el mismo orden en que se registraron, siguiendo el principio FIFO (Primero en Entrar, Primero en Salir). En este programa, la cola garantiza que las solicitudes se procesen en el orden en que llegaron, cumpliendo con el requisito de prioridad temporal.

Su interfaz sencilla permite agregar nuevas solicitudes al final de la cola con el método push y atenderlas desde el frente con el método pop. Este flujo lineal (registro → atención) se adapta perfectamente a la dinámica de las solicitudes de compra, asegurando que se respete el orden de llegada sin necesidad de reordenamientos adicionales.

- Clientes en Espera: ¿Qué estructura es ideal para manejar una fila de espera?

En este caso también se utilizó una cola, ya que sigue la misma lógica que para las solicitudes de compra: los clientes en espera deben atenderse en el orden en que llegan, siguiendo un esquema FIFO (Primero en Entrar, Primero en Salir). Una cola cumple con este requisito de manera natural. La cola proporciona una interfaz sencilla para agregar clientes (al final) y procesarlos (desde el frente), siguiendo su flujo lineal (registro → atención).

Aunque se utiliza la misma estructura (cola) que para las solicitudes de compra, ambas están separadas conceptualmente y en implementación, lo que evita confusiones y errores en el programa. En la implementación, se utiliza el método push para registrar clientes en la cola y pop para atenderlos, asegurando el cumplimiento del esquema FIFO.

- Registro de Cambios: ¿Qué estructura permite registrar cada cambio y deshacer la última modificación?

Para el registro de cambios se usó la pila, aprovechando la lógica del LIFO (Último en Entrar, Primero en Salir), lo que la hace ideal para registrar y deshacer cambios de manera eficiente. Cada cambio realizado se guarda como una acción en la pila. Si es necesario revertirlo, se usa el método pop para eliminar el último cambio y deshacer la acción. En la implementación, se utilizó push para registrar cada cambio, asegurando un historial lineal y reversible, que procesa los cambios desde el más reciente hacia el más antiguo. La pila es fácil de implementar y manejar, lo que permite un control claro y eficiente de los cambios. Además, proporciona una forma sencilla de deshacer acciones rápidamente, restaurando el estado anterior del sistema sin complicaciones. Esto asegura que las modificaciones sean reversibles y que el sistema mantenga su consistencia, sin tantos errores.