

ENTREGA 1: ESTRUCTURAS LINEALES

Simulacion de un supermercado con
atencion prioritaria



OBJETIVOS DEL PROGRAMA



Simular un comportamiento real

Simular el flujo de atención de clientes en un supermercado, aplicando estructuras de datos lineales para modelar el comportamiento real.



Servicio prioritario

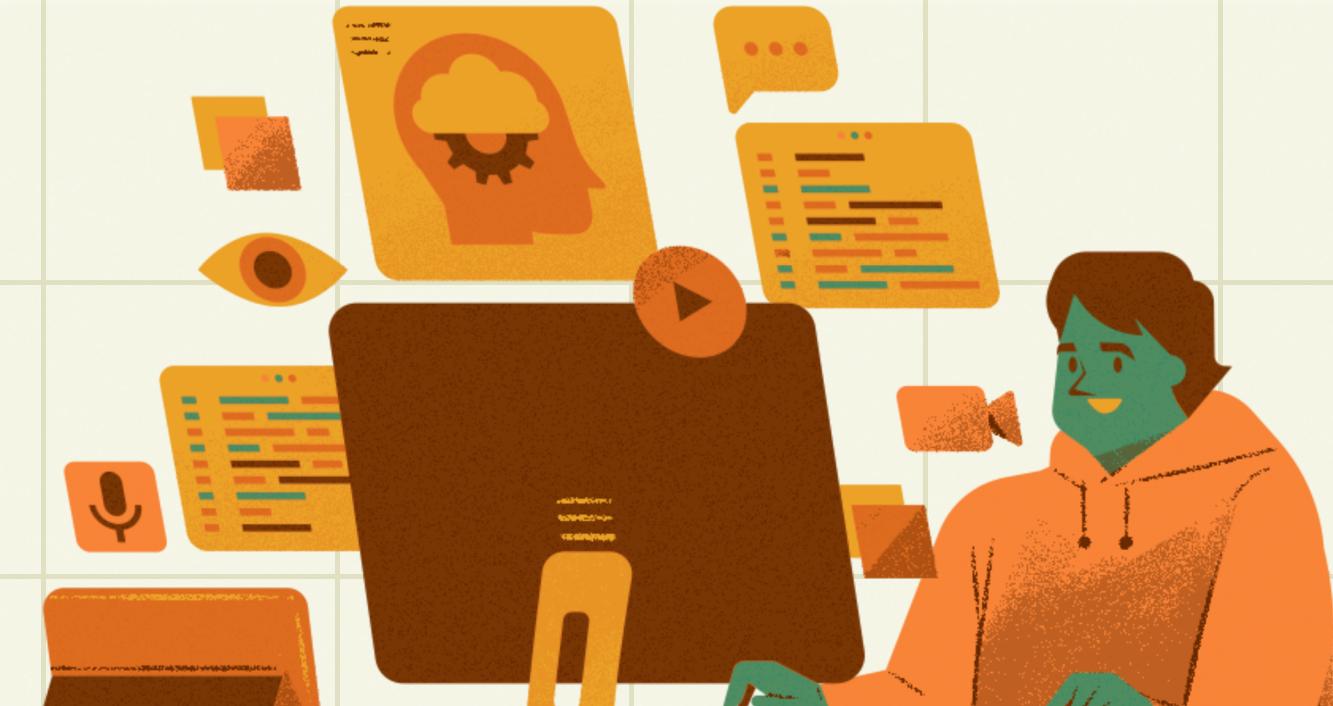
Otorgar prioridad a ciertos tipos de clientes en la fila (discapacidad, adultos mayores o embarazadas).



Guardar registro

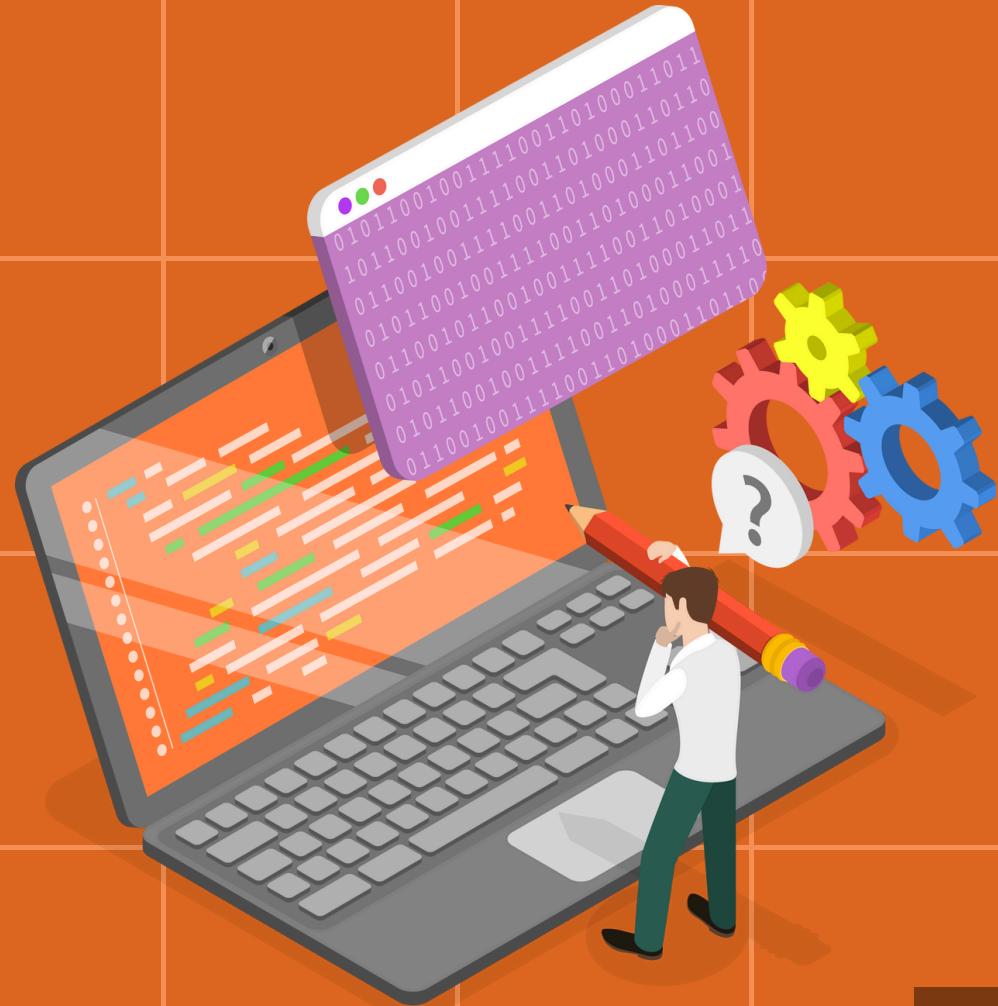
Incorporar estructuras lineales vistas en clase con el fin

LIBRERIAS USADAS



```
#include <iostream>      // Estandar, entrada y salida de datos mediante cin y cout
#include <stack>         // Estructura Pila
#include <queue>         // Estructura de cola y cola con prioridad
#include <string>         // Para usar los string
#include <vector>         // Permite usar vectores dinamicos
#include <cstdlib>        // rand(), srand() y otras funciones de utilidad
#include <ctime>          // Sirve para usar el time()
#include <thread>         // Permite generar hilos de ejecucion
#include <chrono>         // Se usa en conjunto con la libreria thread para pausas y tiempos
#include <cctype>          // Sirve para usar la funcion toupper y tolower
#include <limits>          // Sirve para usar la funcion numeric_limits para limpiar cin
#include <tuple>           // Para usar tuplas
```

ESTRUCTURAS DE DATOS USADAS



Cola prioritaria

Clase ColaPrioritariaD1
Gestiona el orden de atención de los clientes.
Usa el ComparadorPrioridad para decidir quién pasa primero.

Pila

Clase CarritoDeCompras
Almacena los productos del cliente (LIFO).
Métodos: push, pop, mostrarProductos.

Arreglo dinámico

Se usa para almacenar las facturas generadas.
Permite mostrar el historial de atención en orden.

LOGICA DE PRIORIDAD

Se define una estructura Cliente con atributos que definen su prioridad: discapacidad, adultoMayor, embarazada, ordenLlegada.

La cola de prioridad se define con el tipo de datos (Clientes), donde se almacenaran (Un vector) y el comparador de su prioridad.



CODIGO DEL COMPARADOR

El comparador (ComparadorPrioridad) asigna una puntuación a 2 clientes:

3 → prioridad alta (condiciones especiales)

2 → prioridad media (menos de 5 productos)

1 → prioridad normal

Si dos clientes tienen la misma prioridad, se atiende el que llegó primero.

```
struct ComparadorPrioridad {  
    bool operator()(const Cliente& a, const Cliente& b) const {  
        auto prioridad = [](const Cliente& c) {  
            if (c.discapacidad || c.adultoMayor || c.embarazada) return 3;  
            if (c.carrito.size() < 5) return 2;  
            return 1;  
        };
```



PROCESAMIENTO DEL CARRITO

Se le asigna un precio aleatorio (1000–20000) a cada producto, calcula el total de la compra, y muestra los productos y sus precios en consola.

Los resultados se almacenan como una Factura con: el nombre del cliente, sus productos, el total y fecha/hora.

Las facturas se guardan en una cola para mostrar en el historial final.

INTERFAZ VISUAL EN CONSOLA

H-----
BIENVENIDO AL D1 - PASILLO PRINCIPAL

[ESTANTE SUPERIOR] [Tomates] [Lechuga] [Manzanas] [Galletas] [Bebidas]

[ESTANTE MEDIO] [Arroz] [Aceite] [Carne] [Galletas] [Agua]

[ESTANTE INFERIOR] [Leche] [Huevos] [Pan] [Dulces] [Snacks]

Modo: Prueba rápida (clientes predefinidos) o crear tus propios clientes.

¿Deseas usar el modo de prueba (clientes predefinidos)? (S/N):



GRACIAS POR SU ATENCION

Juan Bohorquez
Julián Quintero
Santiago Herrera

