

20 de Octubre de 2025

Documento explicativo del código

Estructuras de datos

Juan Manuel Bohorquez Saavedra

Julian Humberto Quintero Caicedo

Santiago Herrera Parra

## Introducción

Para el proyecto de estructuras de datos, nuestro grupo escogió la idea de representar el funcionamiento de una tienda D1, simulando la selección de productos de clientes y las compras de los mismos en las cajas. Con esta premisa es posible implementar estructuras lineales ya vistas tales como arreglos y listas, incluyendo colas, pilas y listas con prioridades.

Para el proyecto, se usó alrededor de un 40% de herramientas de Inteligencia Artificial, principalmente para crear la interfaz de usuario.

## Estructuras Usadas

Pila:

Se creó una pila de strings implementada con la biblioteca `<stack>` para representar el carro de compra donde normalmente los elementos se apilan. Usualmente el último objeto que se pone en el carro de compras es el primero en salir, ya sea para ser para cancelar su compra o para ser comprado. La clase de *"CarritoDeCompras"* presenta las propiedades de una pila, siguiendo la regla de que el último dato en entrar es el primero en salir, de manera que se ingresa por delante con el atributo *"push()"* y se elimina por delante también con el atributo *".pop()"*. A parte de estos 2 atributos se implementaron otros para conocer el estado de la pila, siendo *".empty()"* un booleano que ayuda a saber si la pila está vacía devolviendo 1 si lo está y 0 si tiene algún valor dentro, *".size()"* para conocer el tamaño de la fila y *".top()"* que devuelve el último valor añadido a la pila.

Cola con prioridad:

La cola con prioridad se implementó en la clase *"ColaPrioritariaD1"*, que representa la fila de clientes en el D1, esto con el fin de escoger quien va a estar de primeras en la fila y realizar sus compras primero. Lo curioso de esta cola es que como dice su nombre, algunos clientes van a tener más prioridad que otros en ser atendidos aunque hayan llegado después.

La prioridad de cada cliente se mide mediante el nivel de prioridad, el cual es dado por algunos atributos de los clientes, siendo que cada cliente "normal" tiene nivel 1. Si el cliente tiene pocos objetos para comprar (menos de 5), va a recibir un 2 en su nivel de prioridad. Y si el cliente es discapacitado, adulto mayor o mujer embarazada, recibe un 3 en su nivel de prioridad sin importar si tiene menos de 5 objetos a comprar.

Como la cola escoge a los clientes con mayor prioridad para ponerlos en primer lugar, el comparador *"ComparadorPrioridad"* se encarga de definir este criterio. En caso de que haya un empate, el factor determinante para desempatar es la hora de llegada, pues el cliente que haya llegado antes va a ser atendido primero.

Arreglo dinámico:

En la estructura Factura, se define un vector (que es lo mismo que un arreglo dinámico) de pares.

Este vector dinámico permite crecer o reducirse según se inserten elementos dentro de sí, y, como es de pares, también permite guardar los datos que están relacionados (Siendo este caso el nombre y precio de un producto) y se necesitan para mostrar en la factura al final del programa.