

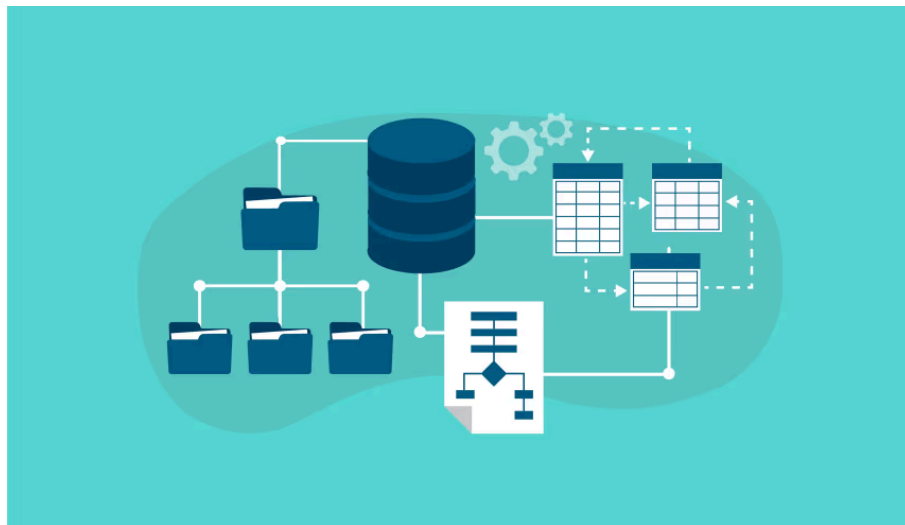
Avance Proyecto

Materia: Estructura de Datos

Maestro: Adalberto Emmanuel Rojas Perea

Alumna: Isabel Kyra Romero Borunda

TECMILENIO



Indice

Indice.....	2
Introducción.....	3
Implementación.....	3
Diagramas.....	3
Estructura.....	4
Usuarios.....	6
Resultados.....	8
Administrador.....	8
Cliente.....	10
Conclusión.....	11

Introducción

En esta actividad se hace un programa para una tienda en línea donde se implementan listas normales, pilas y colas de prioridad para gestionar los datos de la tienda.

Nota:

Solo se explicaran las cosas nuevas a detalle que se vieron esta semana, en este caso la Cola de prioridad.

Implementación

Diagramas

Diagrama de Clases

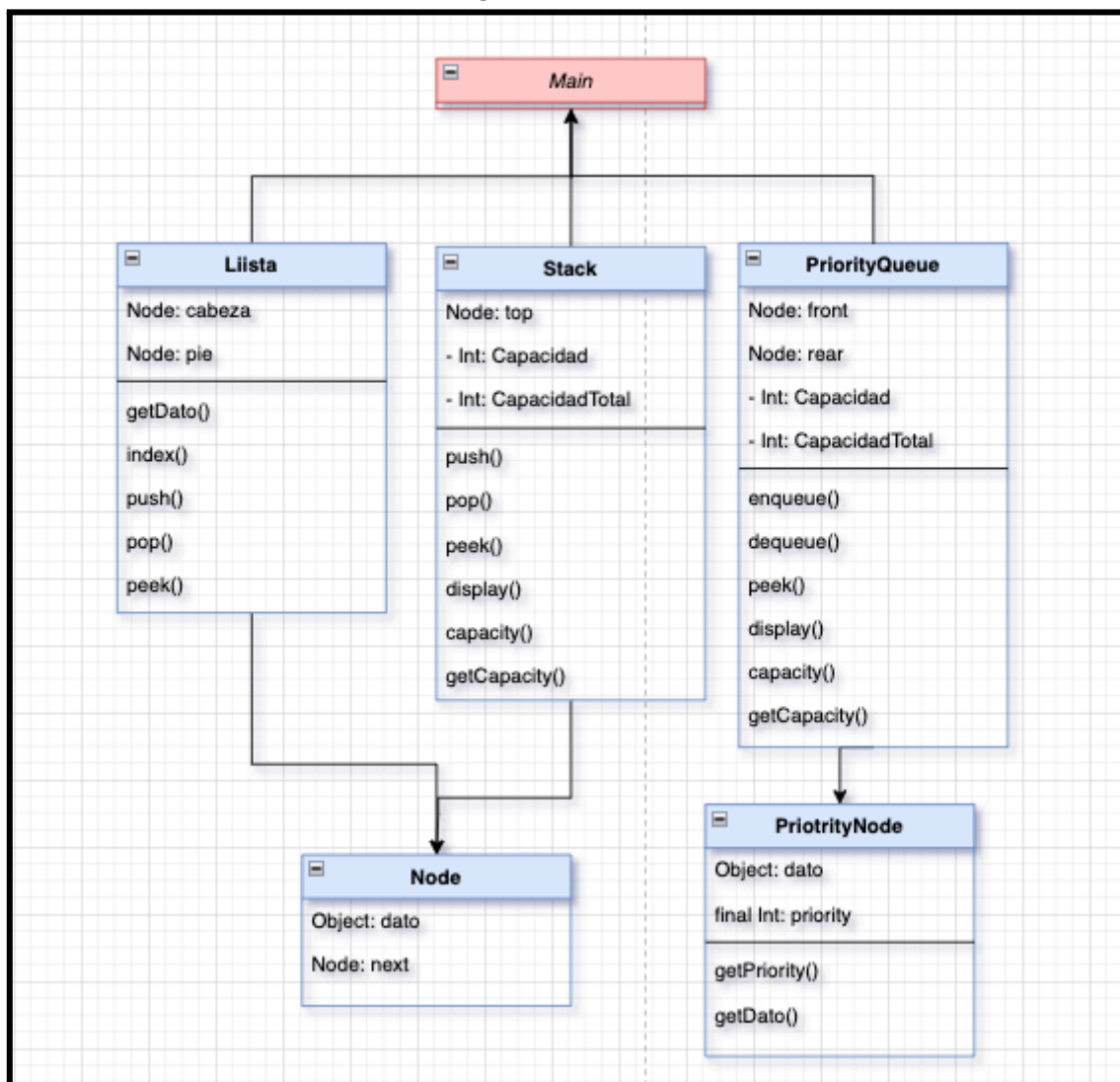
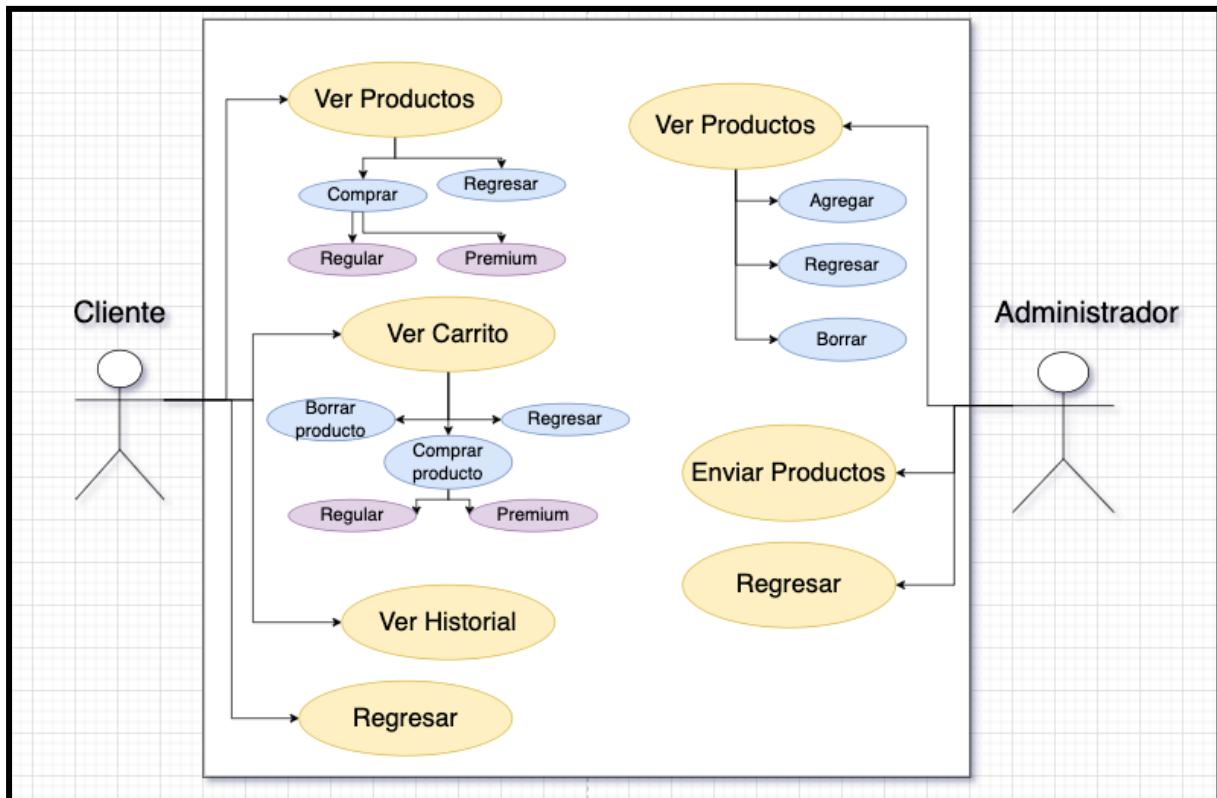


Diagrama de Usos



Estructura

Contamos con **5 clases**, 2 de Nodos y 3 de listas(lista simple, pila y cola de prioridad).

- Nodo de Prioridad / PriorityNode

El nodo de prioridad cuenta con 2 atributos, 'Object: dato' y 'final int: priority'. Donde el dato representa lo que se va a guardar en el nodo y la prioridad nos indica que tan importante va a ser este dato al momento de guardarlo en la cola de prioridad.

También cuenta con dos métodos getters, que nos permite acceder a la prioridad y al dato guardado en este.

```

class PriorityNode { // Nodo colas de prioridad
    Object dato;
    final int priority;

    public PriorityNode(Object dato, int priority) {
        this.dato = dato;
        this.priority = priority;
    }

    // Getters
    public int getPriority() {
        return priority;
    }
    public Object getDato() {
        return dato;
    }
}
  
```

- Nodo / Node

```

// Clases de Nodos
class Node { // Nodo normal
    Object dato;
    Node next;
    public Node(Object dato) {
        this.dato = dato;
        next = null;
    }
}
  
```

Este nodo es normal, contiene 2 atributos, 'Object: dato' y 'Node: next'. El dato representa la información guardada y el atributo next nos señala al siguiente nodo.

- Lista simple / Lista

Esta lista se usa para gestionar los productos de la tienda. Nos permite obtener un producto(getDato), así como el índice de la lista para seleccionar un dato(index), también nos deja ingresar nuevos productos(push), eliminar productos(pop) y nos muestra todos los productos guardados en la lista(mostrar).

```
class Lista {  
    Node cabeza;  
    Node pie;  
    public Lista() {  
        cabeza = null;  
        pie = null;  
    }  
}
```

```
// Getters  
> public Object getDato(int index) { ...  
  
// Indice de la lista  
> public void index() { ...  
// Push  
> public void push(Object dato) { ...  
// Pop  
> public Object pop(int index) { ...  
// Mostrar  
> public void mostrar() { ...  
}
```

- Pila / Stack

La pila se utiliza como un historial de compras del cliente(solo el cliente utiliza el historial, al inicio no vi necesario que el administrador pudiera ver este), donde puede ver los productos anteriores que ha comprado. Aquí los únicos métodos que el cliente puede usar son 'peek' y 'mostrar', las compras se agregan automáticamente al historial y el cliente no puede modificarlo.

```
class Stack {  
    Node top;  
    private int capacidad;  
    private int capacidadTotal;  
  
    public Stack() {  
        top = null;  
        capacidadTotal = 100;  
        capacidad = capacidadTotal;  
    }  
}
```

```
class Stack {  
    Node top;  
    private int capacidad;  
    private int capacidadTotal;  
  
    public Stack() {  
        top = null;  
        capacidadTotal = 100;  
        capacidad = capacidadTotal;  
    }  
}
```

- Cola de prioridad / PriorityQueue

La cola de prioridad se usa para los pedidos que hay que enviar, aquí los pedidos entran a la hora de que el cliente compre algo. Este decide si quiere el paquete premium para darle más prioridad al pedido o si quiere el regular, donde el pedido no va a tener tanta prioridad.

Aquí la prioridad va de menor a mayor, donde 1 es el que tiene la mayor prioridad. El administrador tiene acceso a esta lista donde solo se le permite enviar el pedido que se encuentra primero en la cola, fuera de eso no puede modificar la cola.

```

class PriorityQueue {
    PriorityNode[] data;
    int tamaño;
    int limite;

    public PriorityQueue() {
        limite = 50;
        data = new PriorityNode[limite + 1];
        tamaño = 0;
    }

    // Enqueue
    > public void enqueue(Object dato, int priority) { ...
    // Dequeue
    > public Object dequeue() { ...
    // Peek
    > public Object peek() { ...
    // Mostrar
    > public void mostrar() { ...
    }
}

```

Usuarios

En total tenemos 4 listas creadas para diferentes propósitos, no todos los usuarios pueden modificar o acceder a todas las listas.

```

Lista productos = new Lista();
Lista carrito = new Lista();
PriorityQueue porEnviar = new PriorityQueue();
Stack pedidosAnteriores = new Stack();

```

- **productos**

Aquí se guardan los productos para que los clientes puedan verlos y comprar el que quieran.

- **carrito**

Aquí se guardan los productos para guardarlos más tarde.

- **porEnviar**

Se forman los productos ya comprados para enviarlos a los clientes.

- **pedidosAnteriores**

Es la lista que representa el historial de compras del cliente.

Tenemos dos tipos de usuarios para la tienda:

- **Administrador**

El que gestiona los productos de la tienda. Solo tiene permitido acceder a las siguientes listas de manera limitada:

- **productos (lista simple)**

- [Agregar productos / push](#)

- Le permite agregar productos a la lista.

- [Borrar productos / pop](#)

- Le permite borrar productos de la lista.

- **porEnviar (cola de prioridad)**

- [Enviar productos comprados / dequeue](#)

- Le permite enviar pedidos en la cola según la prioridad de este.

- **Cliente**

Los que compran en línea. Estos pueden acceder a las siguientes listas.

- **productos (cola de prioridad)**

- [Ver productos / mostrar](#)

- Permite al cliente ver los productos que hay en la tienda.

- **carrito (lista simple)**

- [Agregar productos / push](#)

- Puede agregar productos al carrito para comprarlos después, como una wishlist.

- [Borrar productos del carrito / pop](#)

- Puede quitar productos del carrito que ya no quiere.

- [Ver productos del carrito / peek y mostrar](#)

- Le permite ver los productos que tiene guardados en su carrito.

- **pedidosAnteriores (pila)**

- [Ver historial de compras / peek y display](#)

- Le permite ver su historial de compras anteriores

Resultados

Nota:

Me di cuenta de unas fallas en la lógica del código para el menú, por lo que se ingresaron unos datos desde el código en las listas para poder demostrar cómo funciona.

Una de estas fallas fue que al momento de elegir al usuario ya no se puede cambiar después, solo saliendo y volviendo a correr el programa se puede elegir a otro, entonces cómo se tiene que reiniciar cada vez que queremos cambiar de usuario agregue unos datos directamente en el código para poder ver el funcionamiento de unas funciones.

Otra falla fue con el carrito, como tenía la idea de que iba a tener productos guardados no puse una opción para poder agregar productos al carrito.

```
// Datos para casos
productos.push(dato:"Camisa");
productos.push(dato:"Pantalon");
productos.push(dato:"Zapatos");
productos.push(dato:"Collar");
productos.push(dato:"Lentes de sol");

porEnviar.enqueue(dato:"Pedido 1", priority:2);
porEnviar.enqueue(dato:"Pedido 2", priority:1);
porEnviar.enqueue(dato:"Camisa rosa", priority:3);

carrito.push(dato:"Reloj");
carrito.push(dato:"Pantalon");
carrito.push(dato:"Bolsa roja");
```

Administrador

Elegimos el usuario de Administrador y nos muestra varias opciones, seleccionamos la primera para ver los productos que se encuentran en la tienda actualmente.

```
1-. Usuario
2-. Administrador
3-. Salir
>>> 2

Bienvenido Administrador

Que quiere hacer?
  1-. Ver Productos
  2-. Enviar Pedidos
  3-. Salir
>>> 1
```

Eliminar Producto

Luego de seleccionar la opción 1, nos muestra que podemos hacer con la lista. Podemos agregar y borrar un producto.

Seleccionamos la segunda opción para borrar y nos despliega otro menú

```
PRODUCTOS:
// - Camisa
// - Pantalon
// - Zapatos
// - Collar
// - Lentes de sol

1-. Agregar producto  2-. Borrar producto  3-. Regresar
>>> 2

Ingrese el index del producto a borrar
0: Camisa
1: Pantalon
2: Zapatos
3: Collar
4: Lentes de sol
1
// Producto 'Pantalon' Borrado
```


para seleccionar el producto que deseemos, en este caso se borró el pantalón de la tienda.

Agregar Producto

Le damos a la opción 1 para agregar un producto, y luego de agregarlo seleccionamos de nuevo 'Ver productos' para ver que si se agregó. En la segunda imagen podemos ver que si se agregó el producto 'Bolsa'.

```
PRODUCTOS:
// - Camisa
// - Zapatos
// - Collar
// - Lentes de sol

1-. Agregar producto  2-. Borrar producto  3-. Regresar
>>> 1

Ingrese un producto: Bolsa
Producto 'Bolsa' Agregado
```

```
Que quiere hacer?
1-. Ver Productos
2-. Enviar Pedidos
3-. Salir
>>> 1

PRODUCTOS:
// - Camisa
// - Zapatos
// - Collar
// - Lentes de sol
// - Bolsa
```

Enviar Producto

Si queremos enviar un producto solo seleccionamos la segunda opción, donde nos muestran los pedidos a enviar por prioridad y automáticamente se envía el que tiene más prioridad que fue el Pedido con una prioridad de 1.

```
Que quiere hacer?
1-. Ver Productos
2-. Enviar Pedidos
3-. Salir
>>> 2

Pedidos por enviar:
Pedido 2 (prioridad 1)
Pedido 1 (prioridad 2)
Camisa rosa (prioridad 3)

Pedido 'Pedido 2' Enviado
```

Salir del programa

Si queremos salir solo nos vamos al menú principal y seleccionamos la opción 3.

```
PRODUCTOS:
// - Camisa
// - Zapatos
// - Collar
// - Lentes de sol
// - Bolsa

1-. Agregar producto  2-. Borrar producto  3-. Regresar
>>> 3

Que quiere hacer?
1-. Ver Productos
2-. Enviar Pedidos
3-. Salir
>>> 3
>>>>
dfgh@MacBook-Air-de-DFGH Actividades %
```

Cliente

Al seleccionar la opción 1 nos muestra un mensaje de bienvenida de la tienda y luego nos da opciones de lo que podemos hacer.

```
1-. Usuario
2-. Administrador
3-. Salir
>>> 1

Bienvenido a la Tienda Online JDGATE
```

Comprar un Producto

Seleccionamos 'Ver productos' y nos imprime los productos que hay en la tienda. Nos pregunta si queremos comprar un producto y nos da a elegir. Le decimos que sí y elegimos el producto que queremos comprar.

```
Que quiere hacer?
1-. Ver productos
2-. Ver carrito
3-. Ver historial de compras
4-. Salir
>>> 1

PRODUCTOS:
// - Camisa
// - Pantalon
// - Zapatos
// - Collar
// - Lentes de sol

Desea comprar un producto?
1-. SI 2-. NO
>>> 1
```

```
Ingrese el index del producto deseado
0: Camisa
1: Pantalon
2: Zapatos
3: Collar
4: Lentes de sol
>>> 3
```

Nos pregunta si queremos el envío premium o regular, y nos dice que si elegimos premium se le dará más prioridad al producto para enviarlo. Al final nos da un mensaje por la compra hecha.

```
Que tipo de entrega desea?:
1-. Premium
2-. Regular
(Si elige premium se le dara más prioridad a su pedido al enviar)
>>> 1
Su pedido tomara prioridad sobre otros.

Gracias por su compra!
```

Carrito

Si queremos ver nuestro carrito seleccionamos 'Ver carrito' donde nos muestra las cosas que guardamos en el carrito para comprar después, luego nos da la opción de comprar o borrar un producto del carrito.

Podemos ver 3 productos guardados.

```
Que quiere hacer?
  1-. Ver productos
  2-. Ver carrito
  3-. Ver historial de compras
  4-. Salir
>>> 2

CARRITO
// - Reloj
// - Pantalon
// - Bolsa roja
```

Eliminar un Producto

Si queremos eliminar algo le damos a la opción 2, donde nos pregunta cuál producto queremos borrar. En este caso borramos el pantalón del carrito

```
Desea comprar o borrar algo de su carrito?
1-. Comprar 2-. Borrar 3-. Salir
>>> 2
0: Reloj
1: Pantalon
2: Bolsa roja

Ingrese el index del producto a borrar
>>> 1
Producto 'Pantalon' Borrado del Carrito
```

Comprar producto

Podemos comprar algo del carrito dándole a la opción 1, nos pregunta cuál producto queremos comprar y después nos da la opción de elegir el paquete premium o regular.

```
Desea comprar o borrar algo de su carrito?
1-. Comprar 2-. Borrar 3-. Salir
>>> 1

Ingrese el index del producto a comprar
0: Reloj
1: Bolsa roja
>>> 0

Que tipo de entrega desea?:
1-. Premium
2-. Regular
(Si elige premium se le dara más prioridad a su pedido al enviar)
2
Entendido

Gracias por su compra!
```

Historial compras

Podemos ver nuestro historial de compras si le damos a la tercera opción y nos muestra los productos que compramos. En la imagen podemos ver que compramos un reloj y una bolsa.

```
Que quiere hacer?  
1-. Ver productos  
2-. Ver carrito  
3-. Ver historial de compras  
4-. Salir  
>>> 3  
  
HISTORIAL DE COMPRAS:  
  
Ultimo producto comprado:  
  
Historial completo:  
Reloj -> Collar ->
```

Conclusión

Como conclusión de este avance de proyecto hay áreas de mejora, como es al momento de pensar bien en la estructura del menú así como a la hora de tener que reiniciar el programa para poder cambiar de usuario. Pudimos ver que me falló la lógica del código con el carrito y hubo métodos que no se usaron.