



**Avance Final**

**Curso: Estructura de datos**

**Karl Aase Blanco**

**Harold André Rodríguez Cortés**

**Katrina bou Mansour**

**Kenneth Carballo**

**Profesor: Fernando Salas**

**III Cuatrimestre 2019**

# Manual técnico del proyecto

## Clases:

- Lista Simple
- Lista Pedido
- Nodo Simple
- Árbol
- Cola
- Factura
- Ingresos
- Lista Dobike
- Nodo Doble
- Nodo Arbol
- Pedidos
- Main
- Mesa
- Balance

## Main

## Funcionalidad:

Esta clase es considerada como el "Main" esta establece en las mesas 3 usuarios predeterminados e inicializa todos los procesos, estructuras y Jforms

## Lista Simple y doble

### Variables:

NodoL primero;

NodoDoble cabeza, ultimo;

### Funcionalidad:

Esta clase nos permite distintas funciones para modificar la lista que se encuentra en la clase "Nodo L" las cuales son;

Métodos de Inserción de elementos hacia el nodo de una lista mediante las funciones:

- insertarCabezaLista
- insertarUltimo
- insertarUltimo (Producto, entrada)
- insertarLista (String charlie, Producto entrada)

Métodos de Búsqueda de elementos en el nodo de una lista mediante la función:

- buscarLista(String destino)

Métodos de Eliminación de elementos en el nodo de una lista mediante la función:

- eliminar(String entrada)

Métodos de visualización en consola de los elementos en el nodo de una lista mediante la función:

- visualizar()

## **Nodo Lista simple y doble**

### **Variables:**

Mesa dato;

NodoSimple next;

Factura dato;

NodoDoble next, back;

### **Funcionalidad:**

En esta clase es donde se crea la lista mediante los nodos y enlaces, esta cuenta con distintos métodos los cuales cumplen con los siguientes funcionamientos (La clase Nodo Lista simple se encargada de almacenar mesas y la clase nodo lista doble se encargara de almacenar las facturas):

- **NodoL(Producto x):** Aquí es donde se crea el enlace entre los datos (según el tipo el cuales son los productos) y se enlazan
- **Producto getDato():** Retorna el dato (tipo Producto)
- **getEnlace():** Retorna el enlace
- **setEnlace(NodoL enlace):** Establece el enlace

## **Pedidos**

### **Variables:**

- nombreComida= Producto que desea el cliente
- int precio= Cantidad que deberá pagar el cliente
- int cantidad = Cantidad del producto
- int total= total del pago de la entrega

### **Funcionalidad:**

La funcionalidad de esta clase se centra de almacenar los datos que son introducidos en la interfaz “frmListaMesasCliente” para redistribuirlo luego en la estructura del árbol y las listas correspondientemente, cuenta con los siguientes métodos:

- Obtener variables

toString()

- Establecer y obtener la variable

getNombreComida()

setNombreComida(String nombreComida)

- Establecer y obtener la variable

getCantidad()

setCantidad(int cantidad)

- Establecer y obtener la variable

getPrecio()

setPrecio(int precio)

- Establecer y obtener la variable

getTotal()

setTotal(int total)

## **Balance**

### **Variables:**

- String ID= ID de la mesa
- Int total= Producto que desea el cliente
- Cola cola= Declaracion de una cola

### **Funcionalidad:**

La funcionalidad de esta clase se centra de mostrar el balance de ganancias de la soda según los datos que son introducidos en la interfaz “frmListaMesasCliente”, cuenta con los siguientes métodos:

- Establecer y obtener la variable

getID()

setID(String ID)

- Establecer y obtener la variable

getTotal()

setTotal(int total)

- Establecer y obtener la variable

getCola()

setCola(Cola cola)

- Establecer y obtener la variable

getTotal()

setTotal(int total)

- Agregar un elemento

agregaPedido(String Comida, int cantidad, int precio)

## **Factura**

**Variables:**

String nombre, tarjeta, cvc, vencimiento, numeroMesa;  
int totalNeto;  
int personas, ID;  
Cola cola;

**Funcionalidad:**

La funcionalidad de esta clase se reportando y enviando datos para ser almacenados que son introducidos en la interfaz “frmListaMesasCliente” y mostrar los datos en los distintos Jforms para redistribuirlo luego en las estructuras distintas, cuenta con los siguientes métodos:

- Obtener variables

toString()

- Establecer y obtener la variable

getID()

setID(int ID)

- Establecer y obtener la variable

getCola()

setCola(Cola cola)

- Establecer y obtener la variable

getPrecio()



setPrecio(int precio)

- Establecer y obtener la variable

getPersonas()

setPersonas(int personas)

- Establecer y obtener la variable

getNombre()

setNombre(String nombre)

- Establecer y obtener la variable

getTarjeta()

setTarjeta(String tarjeta)

- Establecer y obtener la variable

getCvc()

setCvc(String cvc)

- Establecer y obtener la variable

getVencimiento()

setVencimiento(String vencimiento)

- Establecer y obtener la variable

getNumeroMesa()

setNumeroMesa(String numeroMesa)

- Establecer y obtener la variable

getTotalNeto()

setTotalNeto(int totalNeto)

- Agregar un elemento

agregaPedido(String Comida, int cantidad, int precio)

## Árbol

### Variables:

NodoArbol raíz

int cont

int suma

String msj

### Funcionalidad:

La funcionalidad de esta clase es la de generar métodos de inserción y eliminación de elementos del árbol, además de datos que facilitan la programación como la cantidad de nodos del árbol

- Métodos de inserción y eliminación:

insertar(int datos)

insertaRec(int datos, NodoArbol n)

borrarMenor()

- Métodos de control de programación:

cuentaNodos()

cuentaNodosRec(NodoArbol n)

sumaNodos()

sumaNodosRec(NodoArbol n)

- Métodos de impresión en consola:

inOrden()

inOrdenRec(NodoArbol n)

**Cola**

**Variables:**

NodoCola frente, ultimo;

### **Funcionalidad:**

La funcionalidad de esta clase es la de generar métodos de inserción, ordenamiento y visualización, y eliminación de elementos de la cola.

- Métodos de inserción y eliminación:

PedidosCola()

PedidosColaEditar(int pedido)

borrar(String name)

PedidosColaEliminar(int pedido)

- Métodos de control de orden:

encola(Pedidos p)

atiende()

setCantidadCola(int pedido, int cantidad)

pedidoValida(String name, int cant)

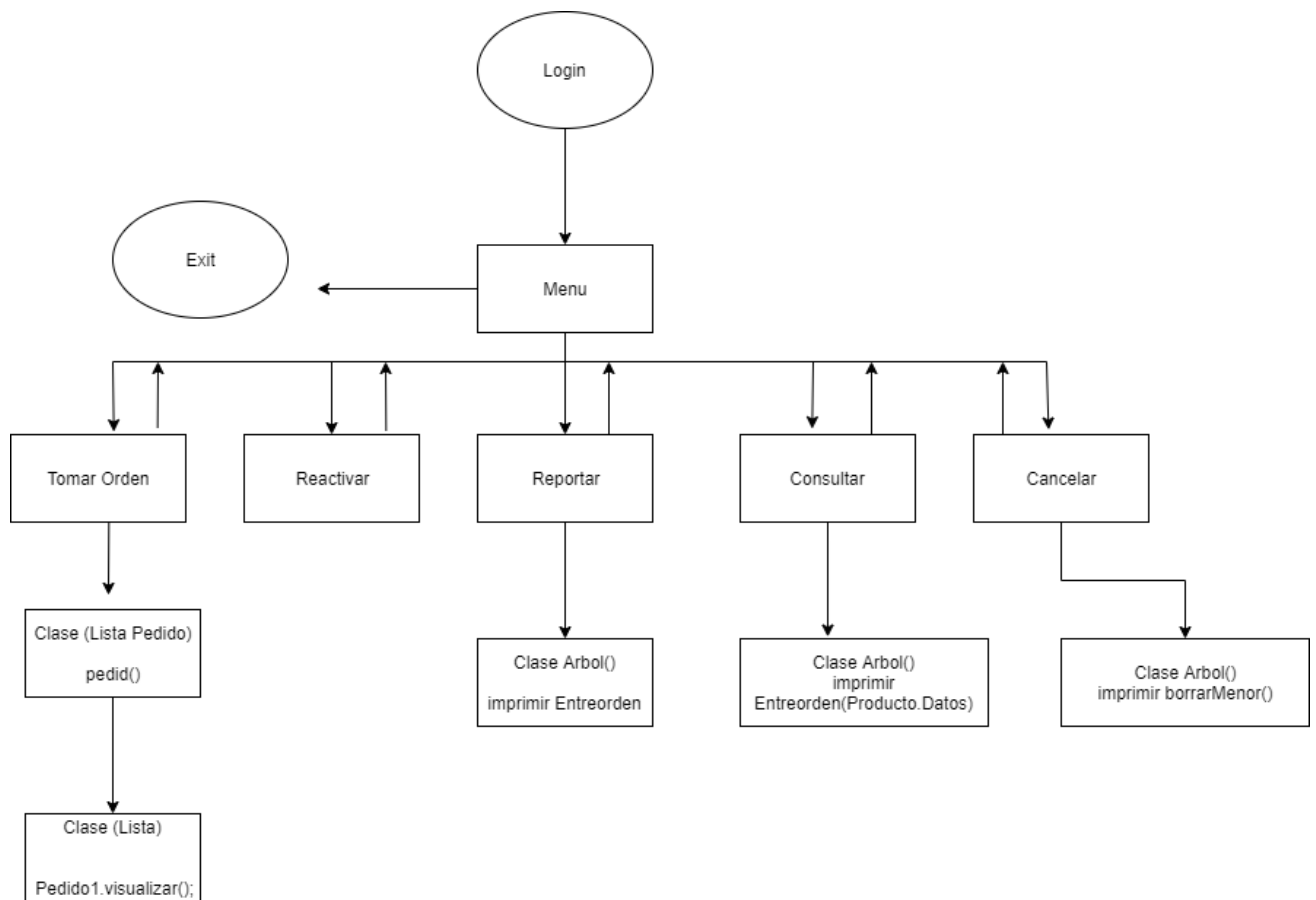
- Métodos de impresión en consola:

toString()

getNombreCola(int pedido)  
 getPrecioCola(int pedido)  
 getCantidadCola(int pedido)  
 print()  
 pedidoCompleto(int pedido)

## Diseños de Estructuras

- Diagrama de clases:



- Descripción de estructuras:

Se utilizaron las siguientes estructuras de datos para realizar la fase 2 del proyecto:

1. Listas

Las listas se encargan de ser métodos de pre visualización y que la persona que esté utilizando el programa apruebe o no este para luego enviar los datos del pedido al árbol

2. Árboles

Los árboles se encargan de almacenar la lista que trae los pedidos aprobados para luego tener funciones de reporte o consulta individual o general de todos los pedidos a la vez de poder eliminar

3. Colas

Se encarga de almacenar un balance de los ingresos y ganancias en un orden de las facturas para así brindar un reporte