

The background of the slide is a vibrant cosmic scene. It features a central galaxy with a bright yellow and orange core, surrounded by swirling purple and pink nebulae. Numerous bright stars of varying sizes are scattered across the dark space. The entire image is framed by a large, black, stylized 'X' shape that divides the slide into four triangular sections.

ESTRUCTURA DE DATOS

Hito 5

Introducción

Java es un lenguaje de programación ampliamente utilizado para codificar aplicaciones web. Ha sido una opción popular entre los desarrolladores durante más de dos décadas, con millones de aplicaciones Java en uso en la actualidad. Java es un lenguaje multiplataforma, orientado a objetos y centrado en la red que se puede utilizar como una plataforma en sí mismo. Es un lenguaje de programación rápido, seguro y confiable para codificarlo todo, desde aplicaciones móviles y software empresarial hasta aplicaciones de macrodatos y tecnologías del servidor.

¿Para qué se utiliza el lenguaje de programación Java?

Debido a que Java es un lenguaje versátil y de uso gratuito, crea software localizado y distribuido.

Algunos usos comunes de Java incluyen:

1. Desarrollo de videojuegos

Muchos videojuegos, así como juegos para móviles y computadoras, se crean con Java. Incluso los juegos modernos que integran tecnología avanzada, como el machine learning o la realidad virtual, se crean con la tecnología de Java.

2. Computación en la nube

Java a menudo se conoce como WORA: escribir una vez y ejecutar en cualquier lugar (por sus siglas en inglés "*Write Once and Run Anywhere*"), lo que lo hace perfecto para aplicaciones descentralizadas basadas en la nube. Los proveedores de la nube eligen el lenguaje Java para ejecutar programas en una amplia gama de plataformas subyacentes.

3. Macrodatos

Java se usa para motores de procesamiento de datos que pueden trabajar con conjuntos de datos complejos y cantidades masivas de datos en tiempo real.

4. Inteligencia artificial

Java es una fuente inagotable de bibliotecas de machine learning. Su estabilidad y velocidad lo hacen perfecto para el desarrollo de aplicaciones de inteligencia artificial como el procesamiento del lenguaje natural y el aprendizaje profundo.

5. Internet de las cosas

Java se ha utilizado para programar sensores y hardware en dispositivos de periferia que pueden conectarse de forma independiente a Internet.

Objetivos

El proyecto tiene de objetivo facilitar la atención en las tiendas de celulares
Mejorando la organización, administración de empleados y ventas.

Código del proyecto

Clase Detalles de la tienda

```
public class Detalle_De_Tienda {  
  
    private String Nombre_tienda;  
    private String Dueño;  
    private int nro_telefono;  
    private String ubicacion;  
  
    public Detalle_De_Tienda(String nombre_tienda, String dueño, int nro_telefono, String ubicacion) {  
        Nombre_tienda = nombre_tienda;  
        Dueño = dueño;  
        this.nro_telefono = nro_telefono;  
        this.ubicacion = ubicacion;  
    }  
  
    public void mostrar()  
    {  
        System.out.println(this.Nombre_tienda);  
        System.out.println(this.Dueño);  
        System.out.println(this.nro_telefono);  
        System.out.println(this.ubicacion);  
    }  
}
```

Clase celulares

```
public class Celulares {
    private int IMEI;
    private String modelo;
    private String marca;
    private String precio;
    private String estado;

    public Celulares(int IMEI, String modelo, String marca, String precio, String estado)
    {
        this.IMEI = IMEI;
        this.modelo = modelo;
        this.marca = marca;
        this.precio = precio;
        this.estado = estado;
    }

    public int getIMEI() { return IMEI; }

    public String getModelo() { return modelo; }

    public String getMarca() { return marca; }

    public String getPrecio() { return precio; }

    public String getEstado() { return estado; }

    public void setIMEI(int IMEI) { this.IMEI = IMEI; }

    public void setModelo(String modelo) { this.modelo = modelo; }

    public void setMarca(String marca) { this.marca = marca; }

    public void setPrecio(String precio) { this.precio = precio; }

    public void setEstado(String estado) { this.estado = estado; }

    public void mostrarCel() {
        System.out.println("\n ");
        System.out.println("IMEI: " + this.IMEI);
        System.out.println("MODELO: " + this.modelo);
        System.out.println("MARCA: " + this.marca);
        System.out.println("PRECIO: " + this.precio);
        System.out.println("ESTADO: " + this.estado);
    }
}
```

Clase Accesorios

```
public class Accesorios {
    private int nro_de_accesorio;
    private String nombre;
    private String precios;
    private String estado;

    public Accesorios(int nro_de_accesorio, String nombre, String precios, String estado) {
        this.nro_de_accesorio = nro_de_accesorio;
        this.nombre = nombre;
        this.precios = precios;
        this.estado = estado;
    }

    public int getNro_de_accesorio() {return nro_de_accesorio;}

    public String getNombre() {return nombre;}

    public String getPrecios() {return precios;}

    public String getEstado() { return estado; }

    public void setNro_de_accesorio(int nro_de_accesorio) {this.nro_de_accesorio = nro_de_accesorio;}

    public void setNombre(String nombre) {this.nombre = nombre;}

    public void setPrecios(String precios) {this.precios = precios;}

    public void setEstado(String estado) { this.estado = estado; }

    public void mostrarAcces() {
        System.out.println("\n ");
        System.out.println("CODIGO DE ACCESORIO: " + this.nro_de_accesorio);
        System.out.println("NOMBRE: " + this.nombre);
        System.out.println("PRECIO: " + this.precios);
        System.out.println("ESTADO: " + this.estado);
    }
}
```


Clase Clientes

```
public class Clientes {  
  
    private int Id_cliente;  
    private String nombre;  
    private String apellidos;  
    private String CI;  
    private String compra;  
  
    public Clientes(int id_cliente, String nombre, String apellidos, String CI, String compra)  
    {  
        Id_cliente = id_cliente;  
        this.nombre = nombre;  
        this.apellidos = apellidos;  
        this.CI = CI;  
        this.compra = compra;  
    }  
  
    public int getId_cliente() {  
        return Id_cliente;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public String getApellidos() {  
        return apellidos;  
    }  
  
    public String getCI() {  
        return CI;  
    }  
  
    public String getCompra() {  
        return compra;  
    }  
}
```

```
    public void setId_cliente(int id_cliente) {  
        Id_cliente = id_cliente;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public void setApellidos(String apellidos) {  
        this.apellidos = apellidos;  
    }  
  
    public void setCI(String CI) {  
        this.CI = CI;  
    }  
  
    public void setCompra(String compra) {  
        this.compra = compra;  
    }  
  
    public void mostrarClient()  
    {  
        System.out.println("\n ");  
        System.out.println("ID: " + this.Id_cliente);  
        System.out.println("NOMBRE: " + this.nombre);  
        System.out.println("APELLIDOS: " + this.apellidos);  
        System.out.println("CI: " + this.CI);  
        System.out.println("Compro: " + this.compra);  
    }  
}
```

Clase Personal

```
public class Personal {

    private String Rol_personal;
    private String nombres;
    private String apellidos;
    private String CI;
    private int Nro_cel;

    public Personal(String rol_personal, String nombres, String apellidos, String CI, int nro_cel)
    {
        Rol_personal = rol_personal;
        this.nombres = nombres;
        this.apellidos = apellidos;
        this.CI = CI;
        Nro_cel = nro_cel;
    }

    public String getRol_personal() {
        return Rol_personal;
    }

    public String getNombres() {
        return nombres;
    }

    public String getApellidos() {
        return apellidos;
    }

    public String getCI() {
        return CI;
    }

    public int getNro_cel() {
        return Nro_cel;
    }
}
```

```
public void setRol_personal(String rol_personal) {
    Rol_personal = rol_personal;
}

public void setNombres(String nombres) {
    this.nombres = nombres;
}

public void setApellidos(String apellidos) {
    this.apellidos = apellidos;
}

public void setCI(String CI) {
    this.CI = CI;
}

public void setNro_cel(int nro_cel) {
    Nro_cel = nro_cel;
}

public void mostrar()
{
    System.out.println("\n ");
    System.out.println("ROL: " + this.Rol_personal);
    System.out.println("NOMBRE: " + this.nombres);
    System.out.println("APELLIDOS: " + this.apellidos);
    System.out.println("CI: " + this.CI);
    System.out.println("CELULAR: " + this.Nro_cel);
}
}
```

Clase Cola de Celulares

```
public class Cola_Celulares {

    private int max;
    private int fin;
    private int ini;
    private Celulares Celular[];

    public Cola_Celulares(int max)
    {
        this.max = max;
        this.Celular = new Celulares[this.max + 1];
        this.ini = 0;
        this.fin = 0;
    }

    public boolean esVacio()
    {
        if(this.ini == 0 && this.fin == 0)
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public boolean esLleno()
    {
        if(this.fin == this.max)
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public int nroElementos()
    {
        return fin - ini;
    }
}
```

```
public void insertar(Celulares elemento)
{
    if(!esLleno())
    {
        fin++;
        Celular[fin] = elemento;
    }
    else
    {
        System.out.println("La cola esta llena");
    }
}

public Celulares eliminar()
{
    Celulares elemento = null;
    if(!esVacio())
    {
        ini++;
        elemento = Celular[ini];
        if (ini == fin)
        {
            ini = 0;
            fin = 0;
        }
    }
    else{
        System.out.println("La cola esta vacia");
    }
    return elemento;
}
```

```
public void mostrar()
{
    if(esVacio())
    {
        System.out.println("La cola esta vacia");
    }
    else
    {
        Celulares elementoEliminado = null;
        System.out.println("\n Celulares");
        Cola_Celulares aux = new Cola_Celulares(this.max) ;
        while(!esVacio()){
            elementoEliminado=eliminar();
            elementoEliminado.mostrarCel();
            aux.insertar(elementoEliminado);
        }
        vaciar(aux);
    }
}

public void vaciar(Cola_Celulares aux)
{
    Celulares elementoEliminado = null;
    while(!aux.esVacio())
    {
        elementoEliminado=aux.eliminar();
        insertar(elementoEliminado);
    }
}
}
```


Clase de Cola de Accesorios

```
public class Cola_Accesorios {
    private int max;
    private int fin;
    private int ini;
    private Accesorios Accesorio[];
    public Cola_Accesorios(int max)
    {
        this.max = max;
        this.Accesorio = new Accesorios[this.max+1];
        this.ini = 0;
        this.fin = 0;
    }
    public boolean esVacio()
    {
        if(this.ini == 0 && this.fin==0)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    public boolean esLleno()
    {
        if(this.fin==this.max)
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public int nroElementos()
    {
        return fin-ini;
    }
}
```

```
public void insertar(Accesorios elemento)
{
    if(!esLleno())
    {
        fin++;
        Accesorio[fin] = elemento;
    }
    else
    {
        System.out.println("La cola esta llena");
    }
}
public Accesorios eliminar()
{
    Accesorios elemento = null;
    if(!esVacio())
    {
        ini++;
        elemento = Accesorio[ini];
        if (ini==fin)
        {
            ini=0;
            fin=0;
        }
    }else{
        System.out.println("La cola esta vacia");
    }
    return elemento;
}
```

```
public void mostrar()
{
    if(esVacio())
    {
        System.out.println("La cola esta vacia");
    }
    else
    {
        Accesorios elementoEliminado = null;
        System.out.println("\n Accesorios");
        Cola_Accesorios aux = new Cola_Accesorios(this.max) ;
        while(!esVacio()){
            elementoEliminado = eliminar();
            elementoEliminado.mostrarAcces();
            aux.insertar(elementoEliminado);
        }
        vaciar(aux);
    }
}
public void vaciar(Cola_Accesorios aux)
{
    Accesorios elementoEliminado = null;
    while(!aux.esVacio())
    {
        elementoEliminado=aux.eliminar();
        insertar(elementoEliminado);
    }
}
}
```

Clase Pila de Clientes

```
public class Pila_Clientes {

    public int max;
    public int tope;
    public Clientes[] Clientes;

    public Pila_Clientes (int max)
    {
        this.tope = 0;
        this.max = max;
        Clientes = new Clientes[this.max + 1];
    }

    public boolean esVacio()
    {
        if ( tope == 0 )
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public boolean esLlena()
    {
        if ( tope == max )
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public int NroElementos()
    {
        return this.tope;
    }
}
```

```
public void adicionar(Clientes NuevoCliente)
{
    if (this.esLlena() == false)
    {
        this.tope = this.tope + 1;
        this.Clientes[this.tope] = NuevoCliente;
    }
    else
    {
        System.out.println("La pila de Clintes esta llena.");
    }
}

public Clientes eliminar()
{
    Clientes ElementosEliminados = null;

    if (this.esLlena() == false)
    {
        ElementosEliminados = this.Clientes[this.tope];
        this.tope = this.tope -1;
        return ElementosEliminados;
    }
    else
    {
        System.out.println("No se puede eliminar una pila vacia.");
    }
    return ElementosEliminados;
}

public void llenar() { }
```

```
public void mostrar()
{
    Clientes elemento = null;
    if(esVacio())
    {
        System.out.println("Pila vacia");
    }
    else
    {
        System.out.println("\n Clientes");
        Pila_Clientes aux = new Pila_Clientes(10) ;
        while (!esVacio())
        {
            elemento = eliminar();
            aux.adicionar(elemento);
            elemento.mostrarClient();
        }
        vaciar(aux);
    }
}

public void vaciar(Pila_Clientes a)
{
    while (!a.esVacio())
        adicionar(a.eliminar());
}
}
```

Clase Pila de Personal

```
public class Pila_personal {

    public int max;
    public int tope;
    public Personal[] personals;

    public Pila_personal (int max)
    {
        this.tope = 0;
        this.max = max;
        personals = new Personal[this.max + 1];
    }

    public boolean esVacio()
    {
        if ( tope == 0 )
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public boolean esLlena()
    {
        if ( tope == max )
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    public int NroElementos()
    {
        return this.tope;
    }
}
```

```
public void adicionar(Personal NuevoCliente)
{
    if (this.esLlena() == false)
    {
        this.tope = this.tope + 1;
        this.personals[this.tope] = NuevoCliente;
    }
    else
    {
        System.out.println("La pila de Clintes esta llena.");
    }
}

public Personal eliminar()
{
    Personal ElementosEliminados = null;

    if (this.esLlena() == false)
    {
        ElementosEliminados = this.personals[this.tope];
        this.tope = this.tope - 1;
        return ElementosEliminados;
    }
    else
    {
        System.out.println("No se puede eliminar una pila vacia.");
    }
    return ElementosEliminados;
}

public void llenar() { }
```

```
public void mostrar()
{
    Personal elemento = null;
    if(esVacio())
    {
        System.out.println("Pila vacia");
    }
    else
    {
        System.out.println("\n Personal");
        Pila_personal aux = new Pila_personal(10) ;
        while (!esVacio())
        {
            elemento = eliminar();
            aux.adicionar(elemento);
            elemento.mostrar();
        }
        vaciar(aux);
    }
}

public void vaciar(Pila_personal a)
{
    while (!a.esVacio())
        adicionar(a.eliminar());
}
}
```


Clase Main

```
public class Main {

    public static void main (String[] args)
    {
        Detalle_De_Tienda c1 = new Detalle_De_Tienda(" CellStore", " Raul Alvares", 78965412, " Av. 6 de marzo");
        c1.mostrar();

        Celulares cel1 = new Celulares(654653213, "Samsung J7", "Samsung", "1200 bs", "Disponible");
        Celulares cel2 = new Celulares(645987946, "Tecno camon 16", "Tecno", "900 bs", "Disponible");
        Celulares cel3 = new Celulares(987984654, "Huawei p20 pro", "Huawei", "1000 bs", "Disponible");
        Celulares cel4 = new Celulares(546546548, "Samsung Galaxy S10", "Samsung", "2500 bs", "Disponible");
        Celulares cel5 = new Celulares(654987984, "Realme 8", "Realme", "1100 bs", "Disponible");

        Cola_Celulares Cels = new Cola_Celulares(100);
        Cels.insertar(cel1);
        Cels.insertar(cel2);
        Cels.insertar(cel3);
        Cels.insertar(cel4);
        Cels.insertar(cel5);

        //Cels.mostrar();

        //-----

        Accesorios Acc1 = new Accesorios(1, "Auriculares", "20 bs", "Disponible");
        Accesorios Acc2 = new Accesorios(2, "Microfonos", "15 bs", "Disponible");
        Accesorios Acc3 = new Accesorios(3, "Parlantes", "40 bs", "Disponible");
        Accesorios Acc4 = new Accesorios(4, "Fundas", "25 bs", "Disponible");
        Accesorios Acc5 = new Accesorios(5, "Memoria 64Gb", "40 bs", "Disponible");

        Cola_Accesorios obje = new Cola_Accesorios(100);
        obje.insertar(Acc1);
        obje.insertar(Acc2);
        obje.insertar(Acc3);
        obje.insertar(Acc4);
        obje.insertar(Acc5);

        //obje.mostrar();
```

Accesorios

ACCESORIO: 1
NOMBRE: Auriculares
PRECIO: 20 bs

ACCESORIO: 2
NOMBRE: Microfonos
PRECIO: 15 bs

ACCESORIO: 3
NOMBRE: Parlantes
PRECIO: 40 bs

ACCESORIO: 4
NOMBRE: Fundas
PRECIO: 25 bs

ACCESORIO: 5
NOMBRE: Memoria 64Gb
PRECIO: 40 bs

Cientes

ID: 5
NOMBRE: Alicia
APELLIDOS: Flores
CI: 64654986SC
Compro:

ID: 4
NOMBRE: Ruben
APELLIDOS: Mamani
CI: 3243546564LP
Compro:

ID: 3
NOMBRE: Jose
APELLIDOS: Quispe
CI: 6546651564SC
Compro:

ID: 2
NOMBRE: Carlos
APELLIDOS: Torrez
CI: 16654654LP
Compro:

ID: 1
NOMBRE: Juan
APELLIDOS: Mamani
CI: 14956854LP
Compro:

```
//-----
```

```
Cientes cli1 = new Cientes(1, "Juan", "Mamani", "14956854LP", "");  
Cientes cli2 = new Cientes(2, "Carlos", "Torrez", "16654654LP", "");  
Cientes cli3 = new Cientes(3, "Jose", "Quispe", "6546651564SC", "");  
Cientes cli4 = new Cientes(4, "Ruben", "Mamani", "3243546564LP", "");  
Cientes cli5 = new Cientes(5, "Alicia", "Flores", "64654986SC", "");
```

```
Pila_Cientes Clients = new Pila_Cientes(100);  
Clients.adicionar(cli1);  
Clients.adicionar(cli2);  
Clients.adicionar(cli3);  
Clients.adicionar(cli4);  
Clients.adicionar(cli5);
```

```
//Clients.mostrar();
```

```
//-----
```

```
Personal Emp1 = new Personal("Administrador", "Rodrigo", "Quispe", "65489566LP", 75695654);  
Personal Emp2 = new Personal("Empleado", "Julian", "Mamani", "656487954SC", 78964532);  
Personal Emp3 = new Personal("Empleado", "Jorge", "Nitale", "987945665SC", 69765465);  
Personal Emp4 = new Personal("Gerente", "Razor", "Tempest", "98798456LP", 79865546);  
Personal Emp5 = new Personal("Conserje", "Bruno", "Flores", "3646789LP", 64896521);
```

```
Pila_personal Perso = new Pila_personal(100);  
Perso.adicionar(Emp1);  
Perso.adicionar(Emp2);  
Perso.adicionar(Emp3);  
Perso.adicionar(Emp4);  
Perso.adicionar(Emp5);
```

```
//Perso.mostrar();
```

Personal

ROL: Conserje
NOMBRE: Bruno
APELLIDOS: Flores
CI: 3646789LP
CELULAR: 64896521

ROL: Gerente
NOMBRE: Razor
APELLIDOS: Tempest
CI: 98798456LP
CELULAR: 79865546

ROL: Empleado
NOMBRE: Jorge
APELLIDOS: Nitale
CI: 987945665SC
CELULAR: 69765465

ROL: Empleado
NOMBRE: Julian
APELLIDOS: Mamani
CI: 656487954SC
CELULAR: 78964532

ROL: Administrador
NOMBRE: Rodrigo
APELLIDOS: Quispe
CI: 65489566LP
CELULAR: 75695654

```
CelularesVendidos(Cels, "Samsung J7", "1200 bs");
```

```
public static void CelularesVendidos(Cola_Celulares Ncels, String modelo, String Venta)
{
    Cola_Celulares Aux = new Cola_Celulares(100);
    while (!Ncels.esVacio())
    {
        Celulares DatoMod = Ncels.eliminar();

        if (DatoMod.getMarca().equals(modelo) && DatoMod.getPrecio().equals(Venta) )
        {
            DatoMod.setPrecio("Vendido");
            Aux.insertar(DatoMod);
        }
        else
        {
            Aux.insertar(DatoMod);
        }
    }
    while (!Aux.esVacio())
    {
        Ncels.insertar(Aux.eliminar());
    }
    Ncels.mostrar();
}
```

Celulares

IMEI: 654653213
MARCA: Samsung J7
PRECIO: Vendido

IMEI: 546546548
MARCA: Samsung Galaxy S10
PRECIO: 2500 bs

IMEI: 987984654
MARCA: Huawei p20 pro
PRECIO: 1000 bs

IMEI: 645987946
MARCA: Tecno camon 16
PRECIO: 900 bs

IMEI: 654987984
MARCA: Realme 8
PRECIO: 1100 bs


```

public static void AccesoriosVendidos(Cola_Accesorios NAcc, String modelo, String Venta)
{
    Cola_Accesorios Aux = new Cola_Accesorios(100);
    while (!NAcc.esVacio())
    {
        Accesorios DatoMod = NAcc.eliminar();

        if (DatoMod.getNombre().equals(modelo) && DatoMod.getPrecio().equals(Venta))
        {
            DatoMod.setPrecio("Vendido");
            Aux.insertar(DatoMod);
        }
        else
        {
            Aux.insertar(DatoMod);
        }
    }
    while (!Aux.esVacio())
    {
        NAcc.insertar(Aux.eliminar());
    }
    NAcc.mostrar();
}

```

```
AccesoriosVendidos(obje, "Microfonos", "15 bs");
```

Accesorios

ACCESORIO: 1
NOMBRE: Auriculares
PRECIO: 20 bs

ACCESORIO: 2
NOMBRE: Microfonos
PRECIO: Vendido

ACCESORIO: 3
NOMBRE: Parlantes
PRECIO: 40 bs

ACCESORIO: 4
NOMBRE: Fundas
PRECIO: 25 bs

ACCESORIO: 5
NOMBRE: Memoria 64Gb
PRECIO: 40 bs

```

public static void BuscarCelular(Cola_Celulares NCel, String nombre)
{
    Cola_Celulares Aux = new Cola_Celulares(100);

    Celulares CelElim = null;

    while (!NCel.esVacio())
    {
        CelElim = NCel.eliminar();
        if (CelElim.getMarca().equals(nombre))
        {
            Aux.insertar(CelElim);
        }
    }
    NCel.vaciar(Aux);
    NCel.mostrar();
}

```

```

BuscarCelular(Cels, "Realme 8");

```

Celulares

IMEI: 654987984

MARCA: Realme 8

PRECIO: 1100 bs

```

public static void AñadirDatoDeCompra(Pila_Clientes pilaCl,String NomClient , String Añadircompra)
{
    Pila_Clientes aux = new Pila_Clientes(10) ;
    Clientes Comp = null;

    while (!pilaCl.esVacio())
    {
        Comp = pilaCl.eliminar();

        if (Comp.getNombre().equals(NomClient) )
        {
            Comp.setCompra(Añadircompra);
            aux.adicionar(Comp);
        }
    }
    pilaCl.vaciar(aux);
    pilaCl.mostrar();
}

```

```

AñadirDatoDeCompra(Clients, "Jose", "Samsung J7");

```

Cientes

ID: 3

NOMBRE: Jose

APELLIDOS: Quispe

CI: 6546651564SC

Compro: Samsung J7

```
public static void CambioDeRol(Pila_personal pilaPer,String NomPersonal , String roles)
{
    Pila_personal aux = new Pila_personal(10) ;
    Personal CRol = null;

    while (!pilaPer.esVacio())
    {
        CRol = pilaPer.eliminar();

        if (CRol.getNombres().equals(NomPersonal) )
        {
            CRol.setRol_personal(roles) ;
            aux.adicionar(CRol);
        }
        else
        {
            aux.adicionar(CRol);
        }
    }
    pilaPer.vaciar(aux);
    pilaPer.mostrar();
}
```

```
CambioDeRol(Perso, "Jorge", "Gerente");
```

Personal
ROL: Conserje NOMBRE: Bruno APELLIDOS: Flores CI: 3646789LP CELULAR: 64896521
ROL: Gerente NOMBRE: Razor APELLIDOS: Tempest CI: 98798456LP CELULAR: 79865546
ROL: Empleado NOMBRE: Jorge APELLIDOS: Nitale CI: 987945665SC CELULAR: 69765465
ROL: Empleado NOMBRE: Julian APELLIDOS: Mamani CI: 656487954SC CELULAR: 78964532
ROL: Administrador NOMBRE: Rodrigo APELLIDOS: Quispe CI: 65489566LP CELULAR: 75695654

Personal
ROL: Conserje NOMBRE: Bruno APELLIDOS: Flores CI: 3646789LP CELULAR: 64896521
ROL: Gerente NOMBRE: Razor APELLIDOS: Tempest CI: 98798456LP CELULAR: 79865546
ROL: Gerente NOMBRE: Jorge APELLIDOS: Nitale CI: 987945665SC CELULAR: 69765465
ROL: Empleado NOMBRE: Julian APELLIDOS: Mamani CI: 656487954SC CELULAR: 78964532
ROL: Administrador NOMBRE: Rodrigo APELLIDOS: Quispe CI: 65489566LP CELULAR: 75695654


```
public static void moverCelularesDeMarca(Cola_Celulares Cels, String marca)
{
    Cola_Celulares Aux = new Cola_Celulares(100);
    Cola_Celulares Aux1 = new Cola_Celulares(100);

    Celulares Cel = null;

    while (!Cels.esVacio())
    {
        Cel = Cels.eliminar();
        if (Cel.getMarca().equals(marca))
        {
            Aux.insertar(Cel);
        }
        else
        {
            Aux1.insertar(Cel);
        }
    }
    Cels.vaciar(Aux);
    Cels.vaciar(Aux1);
    Cels.mostrar();
}
```

```
moverCelularesDeMarca(Cels, "Samsung");
```

Celulares

IMEI: 654653213
MODELO: Samsung J7
MARCA: Samsung
PRECIO: 1200 bs
ESTADO:

IMEI: 546546548
MODELO: Samsung Galaxy S10
MARCA: Samsung
PRECIO: 2500 bs
ESTADO:

IMEI: 645987946
MODELO: Tecno camon 16
MARCA: Tecno
PRECIO: 900 bs
ESTADO:

IMEI: 987984654
MODELO: Huawei p20 pro
MARCA: Huawei
PRECIO: 1000 bs
ESTADO:

IMEI: 654987984
MODELO: Realme 8
MARCA: Realme
PRECIO: 1100 bs
ESTADO:

Conclusión

1. El proyecto en términos básicos cumple con lo que se pide tanto en búsqueda como en mover o cambiar datos.
2. El proyecto se puede mejorar y llevarlo mas alla por ejemplo implementar los form que es del método gráfico.
3. El proyecto también puede mejorar con la implementación de una base de datos para el almacenaje de la información.



FIN