



**CARRERA INGENIERIA DE
SISTEMAS**

Defensa de hito-4

Estudiante:

Michael Rigoberto Alcon Villca

EL ALTO-BOLIVIA



¿A qué se refiere cuando se habla de ESTRUCTURA DE DATOS?

En el ámbito de la informática, las estructuras de datos son aquellas que nos permiten, como desarrolladores, organizar la información de manera eficiente, y en definitiva diseñar la solución correcta para un determinado problema.

Ya sean las más utilizadas comúnmente -como las variables, arrays, conjuntos o clases- o las diseñadas para un propósito específico -árboles, grafos, tablas, etc.-, una estructura de datos nos permite trabajar en un algo nivel de abstracción almacenando información para luego acceder a ella, modificarla y manipularla.

2. ¿Qué significa FIFO?

El método de gestión de inventarios FIFO (First in, First out, en sus siglas en inglés) o PEPS (Primero en Entrar, Primero en Salir), es junto con el método LIFO (Last in, First out), una herramienta muy utilizada en la gestión del almacén.

- La definición y funcionamiento del método FIFO (o PEPS en español) en el almacenaje industrial tiene que ver con la forma en la que mueven las mercancías y es sencillo, primera en entrar (first in), primera en salir (first out). Es decir, la primera mercancía o unidad de carga en entrar en almacén, es la primera en salir de él.

Para ello se desarrollan sistemas de almacenaje industrial específicos que facilitan este tipo de proceso, tanto en términos operativos como de gestión.



3. ¿Muestra la diferencia entre LIFO y FIFO?

Diferencias entre los métodos FIFO y LIFO

Entre las principales diferencias entre estos dos sistemas se encuentran;

- En el método FIFO las primeras mercancías en entrar serán las primeras en salir, mientras que en el método LIFO, las últimas mercancías en entrar serán las primeras en salir.
- El método FIFO es más utilizado en la mayoría de empresas.
- El número de datos registrados es mucho mayor en el sistema LIFO.
- El método FIFO suele ser utilizado para incrementar el impuesto a la renta, mientras que el método LIFO se suele utilizar para diferir en los pagos de los impuestos sobre la renta.

4. ¿Qué es una COLA?

- Una cola es una estructura de datos que almacena elementos en una lista y permite acceder a los datos por uno de los dos extremos de la lista. Un elemento se inserta en la cola (parte final) de la lista y se suprime o elimina por la frente (parte inicial, cabeza) de la lista.

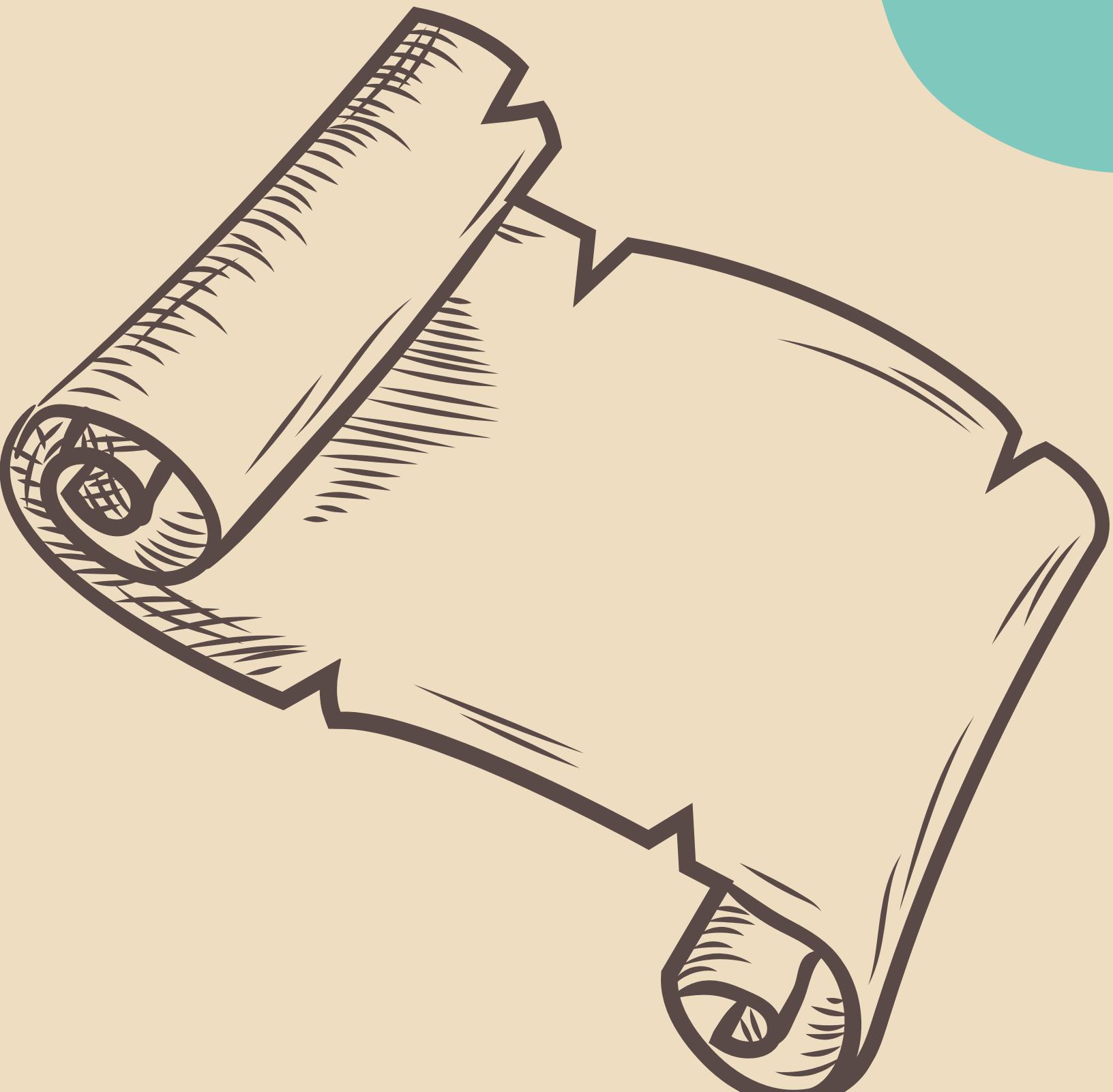
5. ¿Qué es QUEUE en JAVA. una QUEUE será lo mismo que una COLA?

Un objeto de la clase Queue es una cola. Permite almacenar objetos y luego recuperarlos en el orden en el cual se insertaron. Para insertar un objeto a la cola se invoca el método put.



6. ¿Qué es FIN o FRONT en una COLA?

- En una cola los datos entran por un extremo llamado final (rear) y se insertan por el otro extremo llamado frente(front). Una buena analogía de esta estructura de datos es un grupo de personas esperando en línea para entrar al cine.



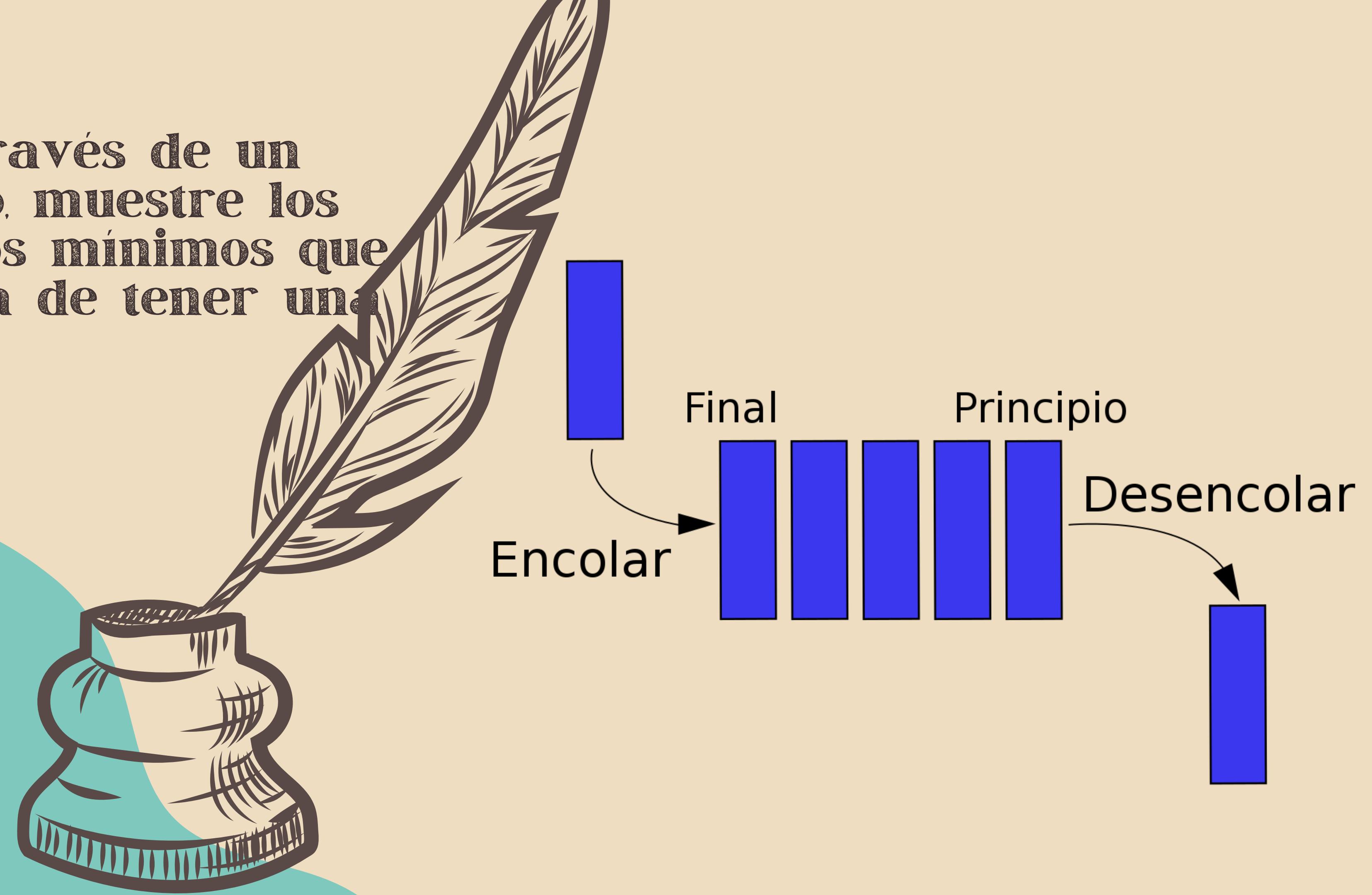
7. ¿A que se refiere los métodos esVacia() y esLlena() en una COLA?

- esVacia(): boolean
retorna verdad si la cola esta vacía es decir no tiene ningún elemento, para esto solo se pregunta si inicio es igual a fin.
- esLlena(): boolean
retorna verdad si es que la cola esta llena, pasa cuando se ha llenado todo el vector, la cantidad de elemento que permite la cola lo determina la variable MAXIMO.

8.¿Qué son los métodos estáticos en JAVA?

Un método estático es un método que tiene sentido invocarla sin crear previamente ningun objeto . Vamos a ver un ejemplo sencillo supongamos que necesitamos crear la clase Rectangulo que calcule el area y el perímetro

10. ¿A través de un gráfico, muestre los métodos mínimos que debería de tener una COLA?



11. Crear las clases necesarias para la PILA DE CLIENTES.



```
package ColaDeClientes;

public class Cliente {
    private String Nombres;
    private String Apellidos;
    private int Edad;
    private String Pais;
    private String Genero;
    private String tipo;

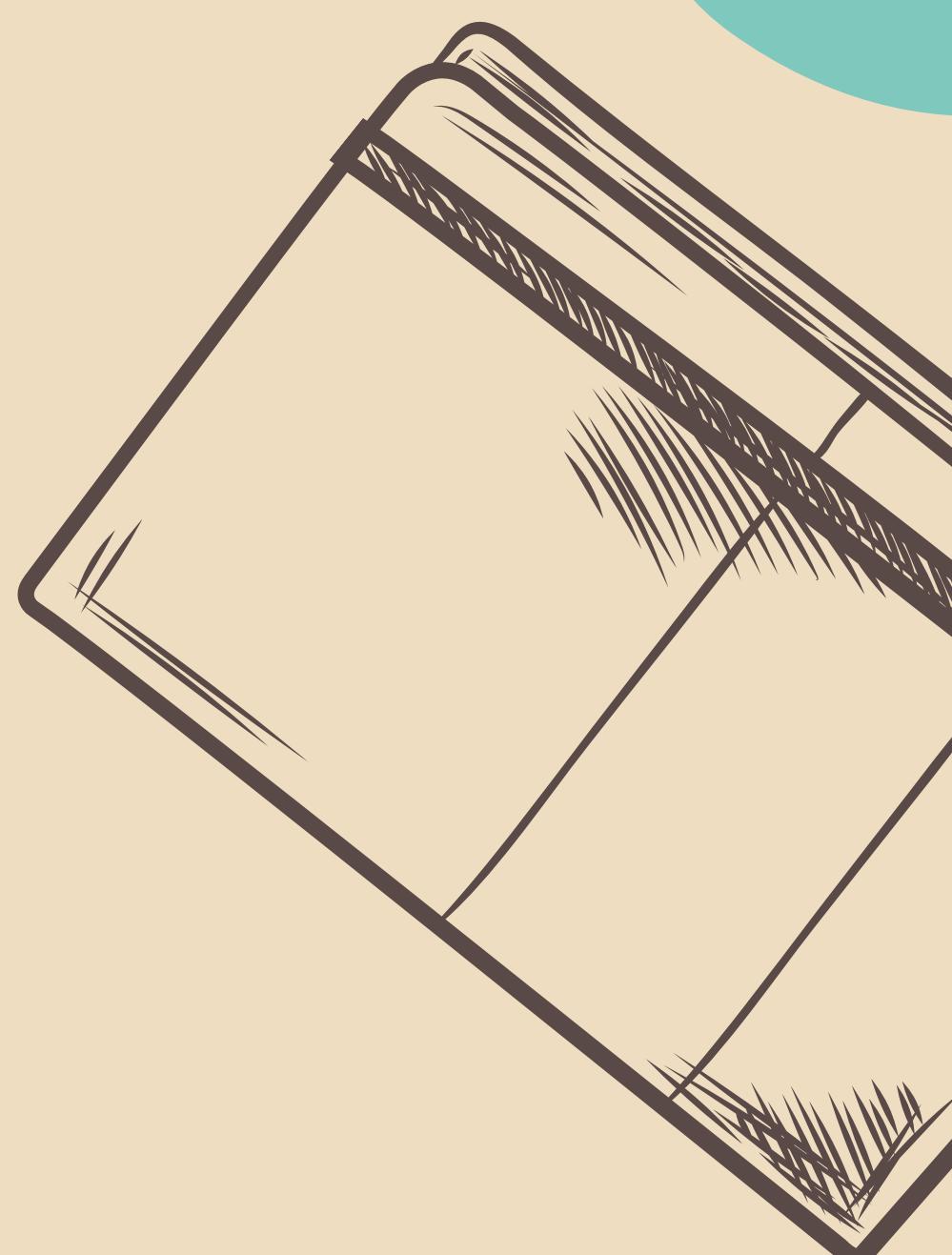
    public Cliente(String Nombres, String Apellidos, int Edad, String Pais, String Genero, String tipo) {
        this.Nombres = Nombres;
        this.Apellidos = Apellidos;
        this.Edad = Edad;
        this.Pais = Pais;
        this.Genero = Genero;
        this.tipo = tipo;
    }

    public String getNombres() {
        return Nombres;
    }

    public String getApellidos() {
        return Apellidos;
    }

    public int getEdad() {
        return Edad;
    }
}
```

11. Crear las clases necesarias para la PILA DE CLIENTES.



The image shows a Java code editor window with the following details:

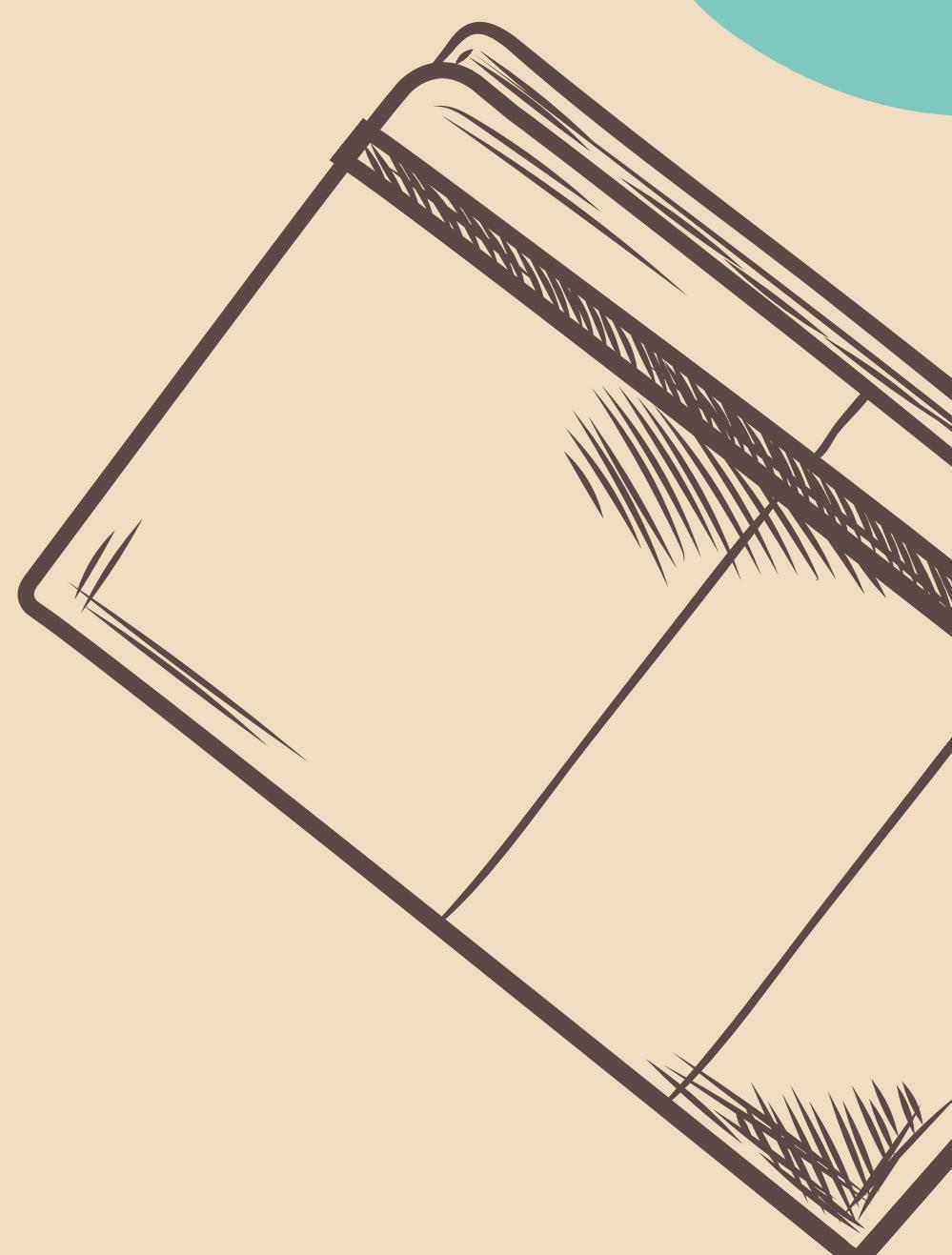
- Project Structure:** The project is named "Tarea_hito3". It contains a "src" folder with "calculadora" and "ColaDeClientes" packages. "calculadora" contains "Cliente.java". "ColaDeClientes" contains "ColaClientes.java", "Main.java", and "PilasDeClientes.java".
- Code Editor:** The "ColaClientes.java" file is open. It defines a class "ColaClientes" with the following methods:
 - getPais(): returns a String.
 - getGenero(): returns a String.
 - setNombres(String nombres): sets the attribute "Nombres".
 - setApellidos(String apellidos): sets the attribute "Apellidos".
 - setEdad(int edad): sets the attribute "Edad".
 - setPais(String pais): sets the attribute "Pais".
 - setGenero(String genero): sets the attribute "Genero".
 - setTipo(String tipo): sets the attribute "tipo".
- Toolbars and Status:** The top bar shows tabs for "Cliente.java", "ColaClientes.java", and "Main.java". The bottom status bar indicates "1 usage" for each method defined in the code.

11. Crear las clases necesarias para la PILA DE CLIENTES.



```
60
61     public void setGenero(String genero) {
62         Genero = genero;
63     }
64
65     public void setTipo(String tipo) {
66         this.tipo = tipo;
67     }
68
69     public void mostrarCliente() {
70         System.out.println("\nMostrando datos del jugador");
71         System.out.println("Nombre: " + this.getNombres());
72         System.out.println("Apellidos: " + this.getApellidos());
73         System.out.println("Edad: " + this.getEdad());
74         System.out.println("Pais: " + this.getPais());
75         System.out.println("Genero: " + this.getGenero());
76         System.out.println("Tipo: " + this.getTipo());
77
78         System.out.println("\n");
79     }
80
81 }
82
```

11. Crear las clases necesarias para la PILA DE CLIENTES.



```
package ColaDeClientes;
public class ColaClientes {
    private int max;
    private int fin;
    private int ini;
    private Cliente[] Clientes;
    public ColaClientes(int max) {
        this.max = max;
        this.Clientes = new Cliente[this.max + 1];
        this.ini = 0;
        this.fin = 0;
    }
    public boolean esVacia() {
        if (this.ini == 0 & this.fin == 0) {
            return true;
        } else {
            return false;
        }
    }
    public boolean esLlena() {
        if (this.fin == this.max) {
            return true;
        } else {
            return false;
        }
    }
}
```

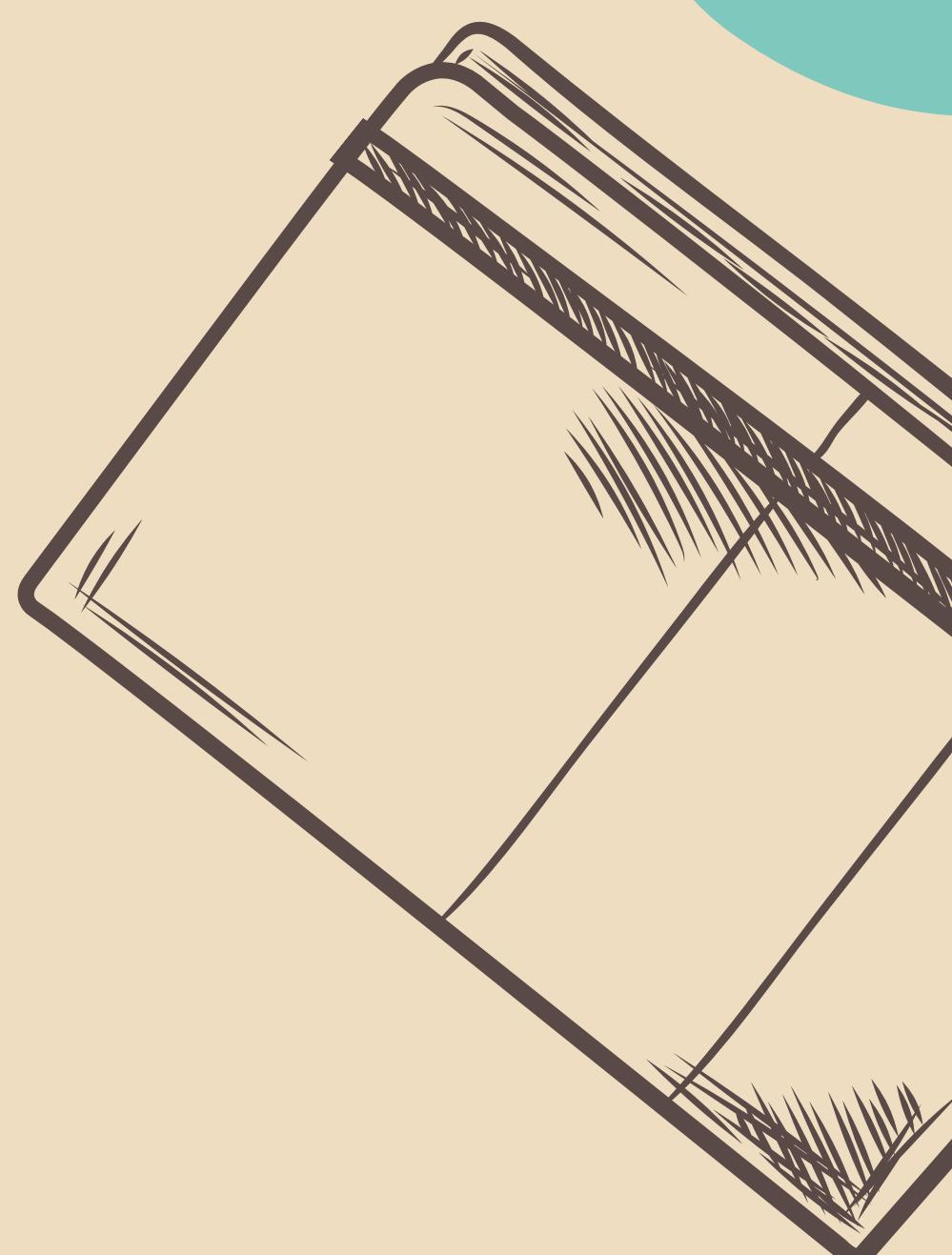
11. Crear las clases necesarias para la PILA DE CLIENTES.



The image shows a screenshot of an IDE (IntelliJ IDEA) interface. On the left is a file tree showing a project named 'Tarea_hito3' with various files and folders like '.idea', 'out', 'SRC', 'calculadora', and 'ColaDeClientes'. The 'ColaDeClientes' folder contains three files: 'Cliente.java', 'ColaClientes.java', and 'Main.java'. The 'ColaClientes.java' file is open in the main editor area, displaying Java code for a queue implementation. The code includes methods for adding new clients ('public void adicionar(Cliente NuevoCliente)'), removing clients ('public Cliente eliminar()'), and displaying the queue ('public void mostrar()'). The code uses an array 'Clientes' to store clients and tracks indices 'ini' and 'fin' to manage the queue. The code editor has syntax highlighting and shows code completion suggestions. A vertical navigation bar on the right lists 'External Libraries' and 'Scratches and Consoles'.

```
2 usages
33     public int nroElem() {
34         return fin - ini;
35     }
36
20 usages
37     public void adicionar(Cliente NuevoCliente) {
38         if (!esLlena()) {
39             fin++;
40             Clientes[fin] = NuevoCliente;
41         } else {
42             System.out.println("Cola de numeros llena");
43         }
44     }
6 usages
45     public Cliente eliminar() {
46         Cliente elementoEliminado = null;
47         if (!esVacia()) {
48             this.ini++;
49             elementoEliminado = this.Clientes[ini];
50             if (ini == fin) {
51                 ini = 0;
52                 fin = 0;
53             }
54             return elementoEliminado;
55         } else {
56             System.out.println("Cola de numeros vacia");
57         }
58         return elementoEliminado;
59     }
4 usages
62     public void mostrar () {
63         Cliente elem = null;
64         if (esVacia())
65             System.out.println("Cola Vacia");
66         else {
67             System.out.println("\nDatos de la Cola de clientes");
68             ColaClientes aux = new ColaClientes(fin);
69             while (!esVacia()) {
```

11. Crear las clases necesarias para la PILA DE CLIENTES.



The image shows a Java code editor interface with several tabs open. The tabs include 'Cliente.java', 'ColaClientes.java', and 'Main.java'. The code is related to a queue implementation for clients. The 'ColaClientes.java' tab contains the following code:

```
49     elementoEliminado = this.Clientes[ini];
50     if (ini == fin) {
51         ini = 0;
52         fin = 0;
53     }
54     return elementoEliminado;
55 } else {
56     System.out.println("Cola de numeros vacia");
57 }
58 return elementoEliminado;
59 }
60 }
61 }
62 }
63 public void mostrar () {
64     Cliente elem = null;
65     if (esVacia())
66         System.out.println("Cola Vacia");
67     else {
68         System.out.println("\nDatos de la Cola de clientes");
69         ColaClientes aux = new ColaClientes(fin);
70         while (!esVacia()) {
71             elem = this.eliminar();
72             aux.adicionar (elem);
73             elem.mostrarCliente();
74         }
75         vaciar(aux);
76     }
77 }
78 @
79 public void vaciar (ColaClientes cola) {
80     while (!cola.esVacia())
81         adicionar(cola.eliminar());
82 }
```

11. Crear las clases necesarias para la PILA DE CLIENTES.

```
package ColaDeClientes;

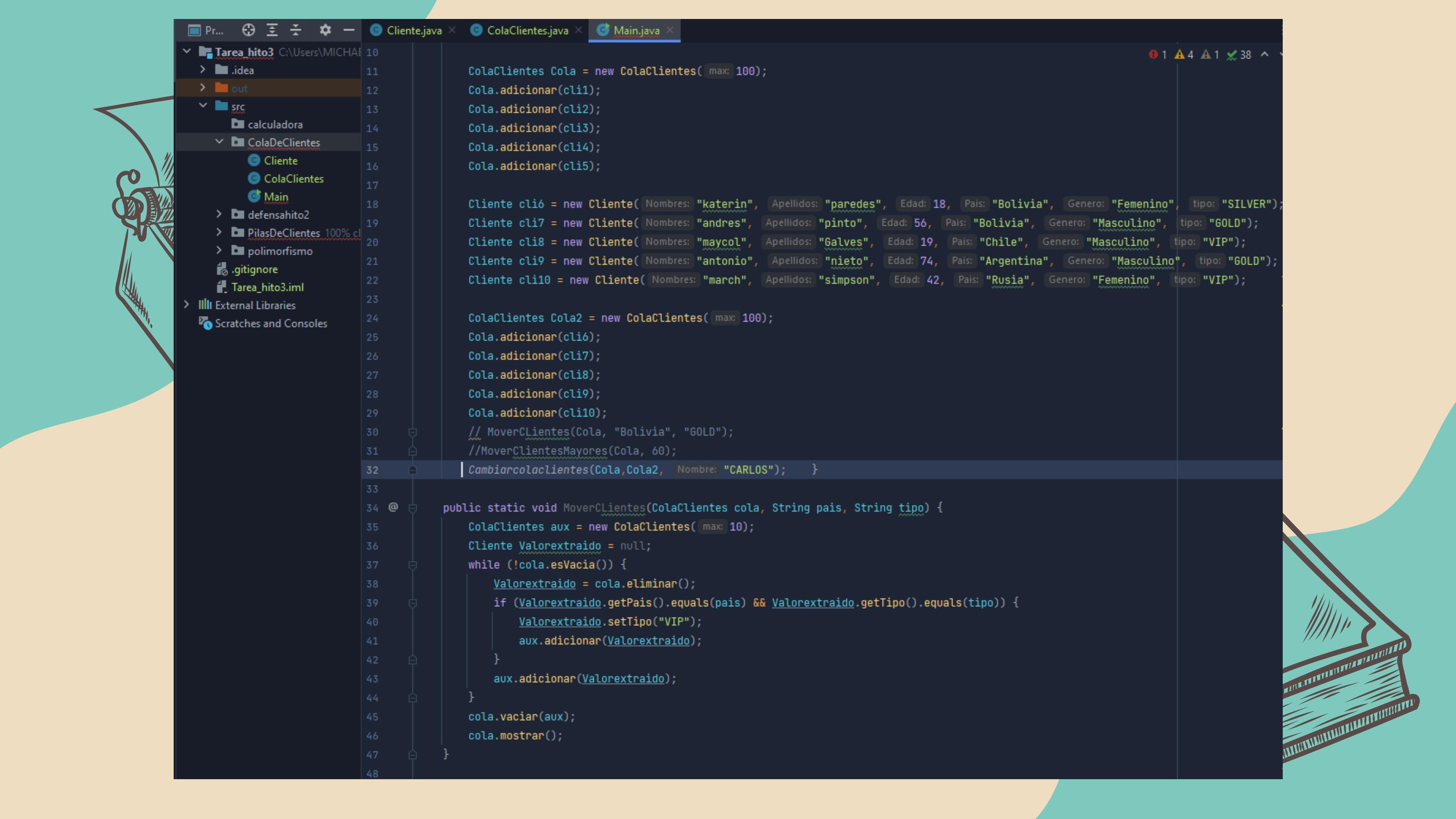
public class Main {
    public static void main(String[] args) {
        Cliente cli1 = new Cliente(Nombres: "Carla", Apellidos: "CALLE", Edad: 25, Pais: "canada", Genero: "Femenino", tipo: "SILVER");
        Cliente cli2 = new Cliente(Nombres: "DANIEL", Apellidos: "TANTANI", Edad: 32, Pais: "peru", Genero: "Masculino", tipo: "GOLD");
        Cliente cli3 = new Cliente(Nombres: "DAYANA", Apellidos: "Martines", Edad: 45, Pais: "venezuela", Genero: "Femenino", tipo: "VIP");
        Cliente cli4 = new Cliente(Nombres: "JOSE", Apellidos: "Soto", Edad: 56, Pais: "chile", Genero: "Masculino", tipo: "GOLD");
        Cliente cli5 = new Cliente(Nombres: "CARLOS", Apellidos: "Salsero", Edad: 32, Pais: "ucrania", Genero: "Masculino", tipo: "VIP");

        ColaClientes Cola = new ColaClientes(max: 100);
        Cola.adicionar(cli1);
        Cola.adicionar(cli2);
        Cola.adicionar(cli3);
        Cola.adicionar(cli4);
        Cola.adicionar(cli5);

        Cliente cli6 = new Cliente(Nombres: "katerin", Apellidos: "paredes", Edad: 18, Pais: "Bolivia", Genero: "Femenino", tipo: "SILVER");
        Cliente cli7 = new Cliente(Nombres: "andres", Apellidos: "pinto", Edad: 56, Pais: "Bolivia", Genero: "Masculino", tipo: "GOLD");
        Cliente cli8 = new Cliente(Nombres: "maycol", Apellidos: "Galves", Edad: 19, Pais: "Chile", Genero: "Masculino", tipo: "VIP");
        Cliente cli9 = new Cliente(Nombres: "antonio", Apellidos: "nieto", Edad: 74, Pais: "Argentina", Genero: "Masculino", tipo: "GOLD");
        Cliente cli10 = new Cliente(Nombres: "march", Apellidos: "simpson", Edad: 42, Pais: "Rusia", Genero: "Femenino", tipo: "VIP");

        ColaClientes Cola2 = new ColaClientes(max: 100);
        Cola.adicionar(cli6);
        Cola.adicionar(cli7);
        Cola.adicionar(cli8);
        Cola.adicionar(cli9);
        Cola.adicionar(cli10);
        // MoverCLientes(Cola, "Bolivia", "GOLD");
        //MoverClientesMayores(Cola, 60);
        Cambiarcolaclientes(Cola, Cola2, Nombre: "CARLOS"); }

    public static void MoverCLientes(ColaClientes cola, String pais, String tipo) {
        ColaClientes aux = new ColaClientes(max: 10);
        Cliente Valorextraido = null;
        while (!cola.esVacia()) {
            Valorextraido = cola.eliminar();
            if (Valorextraido.getPais().equals(pais) && Valorextraido.getTipo().equals(tipo)) {
                Valorextraido.setTipo("VIP");
                aux.adicionar(Valorextraido);
```



Project: Tarea_hito3 C:\Users\MICHAEL

Files:

- .idea
- out
- src
 - calculadora
 - ColaDeClientes
 - Cliente
 - ColaClientes
 - Main
- defensahito2
- PilasDeClientes 100% cl
- polimorfismo
- .gitignore
- Tarea_hito3.iml

External Libraries

Scratches and Consoles

```
10
11     ColaClientes Cola = new ColaClientes( max: 100);
12     Cola.adicionar(cli1);
13     Cola.adicionar(cli2);
14     Cola.adicionar(cli3);
15     Cola.adicionar(cli4);
16     Cola.adicionar(cli5);

17
18     Cliente cli6 = new Cliente( Nombres: "katerin", Apellidos: "paredes", Edad: 18, Pais: "Bolivia", Genero: "Femenino", tipo: "SILVER");
19     Cliente cli7 = new Cliente( Nombres: "andres", Apellidos: "pinto", Edad: 56, Pais: "Bolivia", Genero: "Masculino", tipo: "GOLD");
20     Cliente cli8 = new Cliente( Nombres: "maycol", Apellidos: "Galves", Edad: 19, Pais: "Chile", Genero: "Masculino", tipo: "VIP");
21     Cliente cli9 = new Cliente( Nombres: "antonio", Apellidos: "nieto", Edad: 74, Pais: "Argentina", Genero: "Masculino", tipo: "GOLD");
22     Cliente cli10 = new Cliente( Nombres: "march", Apellidos: "simpson", Edad: 42, Pais: "Rusia", Genero: "Femenino", tipo: "VIP");

23
24     ColaClientes Cola2 = new ColaClientes( max: 100);
25     Cola.adicionar(cli6);
26     Cola.adicionar(cli7);
27     Cola.adicionar(cli8);
28     Cola.adicionar(cli9);
29     Cola.adicionar(cli10);
30     // MoverCLientes(Cola, "Bolivia", "GOLD");
31     //MoverClientesMayores(Cola, 60);
32     Cambiarcolaclientes(Cola,Cola2, Nombre: "CARLOS"); }

33
34     @
35     public static void MoverCLientes(ColaClientes cola, String pais, String tipo) {
36         ColaClientes aux = new ColaClientes( max: 10);
37         Cliente Valorextraido = null;
38         while (!cola.esVacia()) {
39             Valorextraido = cola.eliminar();
40             if (Valorextraido.getPais().equals(pais) && Valorextraido.getTipo().equals(tipo)) {
41                 Valorextraido.setTipo("VIP");
42                 aux.adicionar(Valorextraido);
43             }
44             aux.adicionar(Valorextraido);
45         }
46         cola.vaciar(aux);
47         cola.mostrar();
48     }
```

12.Inicializar la cola de clientes.

- Crear una cola con 5 clientes.
 - En la clase MAIN deberán estar los 5 clientes.
 - Mostrar todos los datos de la cola de clientes
- Adjuntar los siguientes
 - El código del método que resuelve el problema.
 - Una imagen de la salida de la consola.
 - Link que me lleve a la clase main (GitHub)



The screenshot shows an IDE interface with three tabs open: Cliente.java, ColaClientes.java, and Main.java. The Main.java tab contains the following code:

```
Cola.adicionar(cli9);
Cola.adicionar(cli10);
Cola.mostrar();
//Cola2.mostrar();
// MoverClientes(Cola, "Bolivia", "GOLD");
//MoverClientesMayores(Cola, 60);
//Cambiarcolaclientes(Cola,Cola2, "CARLOS");

public static void MoverClientes(ColaClientes cola, String pais, String tipo)
    ColaClientes aux = new ColaClientes( max: 10);
```

The Run panel shows the output of the Main class:

```
Datos de la Cola de clientes
Mostrando datos del jugador
Nombre: Carla
Apellidos: CALLE
Edad: 25
Pais: canada
Genero: Femenino
Tipo: SILVER

Mostrando datos del jugador
Nombre: DANIEL
Apellidos: TANTANI
Edad: 32
Pais: peru
Genero: Masculino
Tipo: GOLD

Mostrando datos del jugador
Nombre: DAYANA
Apellidos: Martines
Edad: 45
Pais: venezuela
Genero: Femenino
```

13.Promoción para usuarios de Bolivia.

En el mes de diciembre a todos los clientes de Bolivia se les dará una promoción en cuanto a precios en viajes a nivel nacional.

■ A todos los clientes que sean de nacionalidad boliviana y además el tipo

de cliente GOLD convertir a estos clientes en VIP

■ Es decir si es de Bolivia y es GOLD deberá ser ahora un cliente VIP

○ El método estático dentro de la clase MAIN recibe 3 atributos

■ La cola de clientes

■ El tipo de cliente

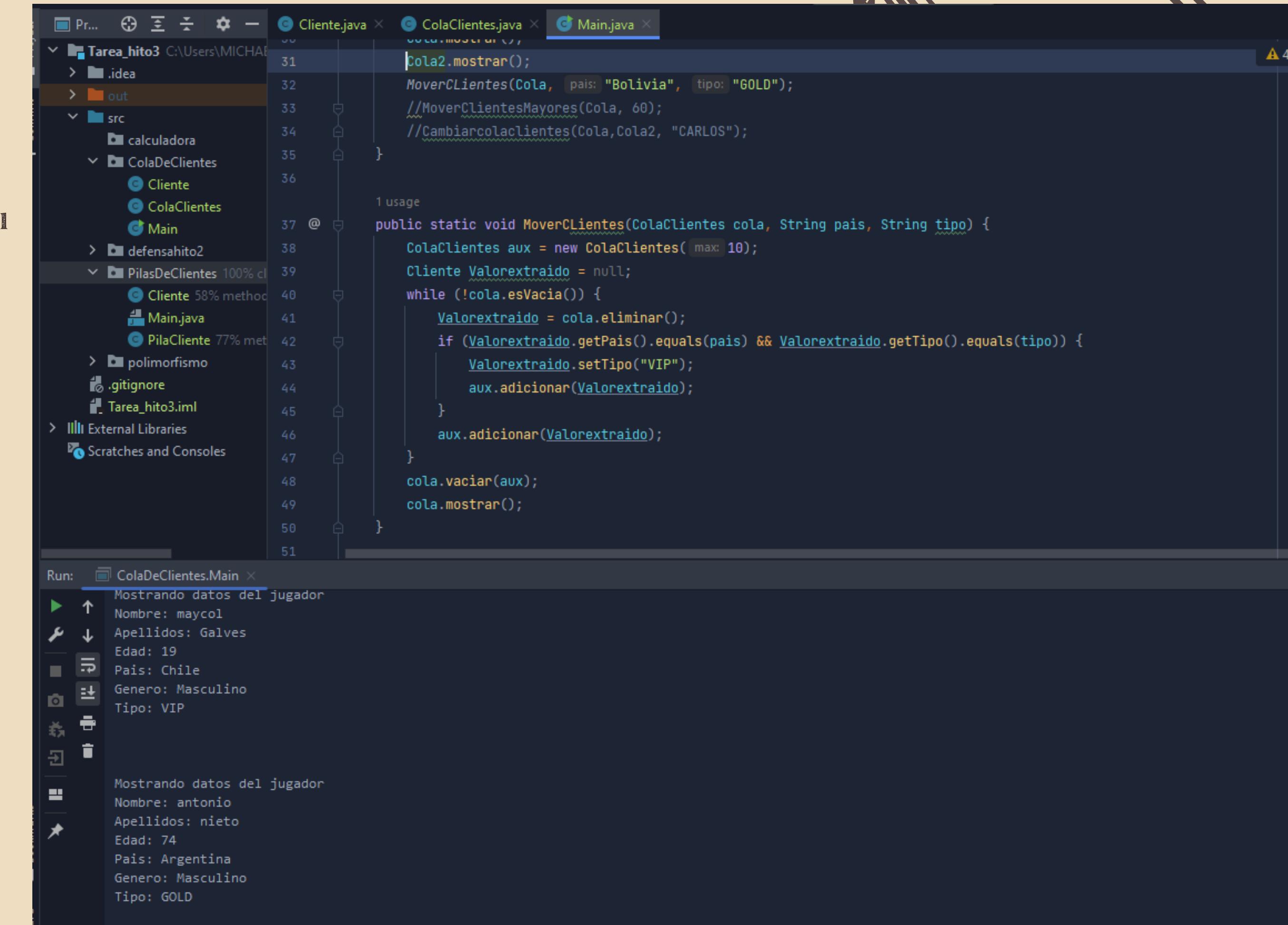
■ La nacionalidad del cliente.

○ Adjuntar los siguientes

■ El código del método que resuelve el problema.

Una imagen de la salida de la consola

LIMA que me lleve a la clase



The screenshot shows an IDE interface with three tabs: Cliente.java, ColaClientes.java, and Main.java. The Main.java tab is active, displaying the following code:

```
Cola2.mostrar();
MoverClientes(Cola, pais: "Bolivia", tipo: "GOLD");
//MoverClientesMayores(Cola, 60);
//Cambiarcolaclientes(Cola,Cola2, "CARLOS");
}

public static void MoverClientes(ColaClientes cola, String pais, String tipo) {
    ColaClientes aux = new ColaClientes( max: 10 );
    Cliente Valorextraido = null;
    while (!cola.esVacia()) {
        Valorextraido = cola.eliminar();
        if (Valorextraido.getPais().equals(pais) && Valorextraido.getTipo().equals(tipo)) {
            Valorextraido.setTipo("VIP");
            aux.adicionar(Valorextraido);
        }
        aux.adicionar(Valorextraido);
    }
    cola.vaciar(aux);
    cola.mostrar();
}
```

The Run panel at the bottom shows two executions of the Main class:

- Mostrando datos del jugador
Nombre: maycol
Apellidos: Galves
Edad: 19
Pais: Chile
Genero: Masculino
Tipo: VIP
- Mostrando datos del jugador
Nombre: antonio
Apellidos: nieto
Edad: 74
Pais: Argentina
Genero: Masculino
Tipo: GOLD

14 Moviendo clientes en la cola



```
Cliente      49     |     cola.mostrar();
ColaClientes 50     |   }
Main          51
defensahito2 52     //Determinar cuantos clientes son mayores a 60 años
PilasDeClientes 100% cl 53
  Cliente 58% method
  Main.java
  PilaCliente 77% met
polimorfismo
.gitignore
Tarea_hito3.iml
External Libraries
Scratches and Consoles

        54 @     public static void MoverClientesMayores(ColaClientes cola, int edadMayor) {
        55     ColaClientes aux = new ColaClientes( max: 10);
        56     ColaClientes aux2 = new ColaClientes( max: 10);
        57     Cliente Valorextraido = null;
        58     while (!cola.esVacia()) {
        59         Valorextraido = cola.eliminar();
        60         if (Valorextraido.getEdad() >= edadMayor) {
        61             cola.adicionar(Valorextraido);
        62         }
        63         aux.adicionar(Valorextraido);
        64     }
        65     cola.vaciar(aux2);
        66     cola.vaciar(aux);
        67     cola.mostrar();
        68 }

        69
        70
        71 @     public static void Cambiarcolaclientes(ColaClientes colaA, ColaClientes colaB, String Nombre) {
        72     int nroElemColaA = colaA.nroElem();
        73     int nroElemColaB = colaB.nroElem();
        74     ColaClientes aux = new ColaClientes( max: 100);
        75     ColaClientes aux2 = new ColaClientes( max: 100);

        76
        77     Cliente valorEliminado;
```

Run: ColaDeClientes.Main

- ▶ Cola de numeros llena
- ▶ Cola de numeros llena
- ▶ Cola de numeros llena

15.Moviendo clientes entre 2 colas



The image shows a hand holding a pencil, writing on a large, stylized question mark that spans across the bottom of the slide. The question mark is filled with a light blue color and has a dark blue outline.

Project: Tarea_hito3 C:\Users\MICHAEL

Client.java ColaClientes.java Main.java

```
public static void Cambiarcolaclientes(ColaClientes colaA, ColaClientes colaB, String Nombre) {
    int nroElemColaA = colaA.nroElem();
    int nroElemColaB = colaB.nroElem();
    ColaClientes aux = new ColaClientes(max: 100);
    ColaClientes aux2 = new ColaClientes(max: 100);

    Cliente valorEliminado;
    for (int i = 1; i <= nroElemColaA; i++) {
        valorEliminado = colaA.eliminar();
        if (valorEliminado.getNombres().equals(Nombre)) {
            colaB.adicionar(valorEliminado);
        } else {
            aux.adicionar(valorEliminado);
        }
    }
    for (int i = 1; i <= nroElemColaB; i++) {
        valorEliminado = colaB.eliminar();
        if (valorEliminado.getNombres().equals(Nombre)) {
            colaA.adicionar(valorEliminado);
        } else {
            aux2.adicionar(valorEliminado);
        }
    }
    colaA.vaciar(aux);
    colaB.vaciar(aux2);
    colaA.mostrar();
    colaB.mostrar();
}
```

Run: ColaDeClientes.Main

Mostrando datos del jugador
Nombre: CARLOS
Apellidos: Salsero
Edad: 32
Pais: ucrania
Genero: Masculino
Tipo: VIP

Process finished with exit code 0



Gracias

