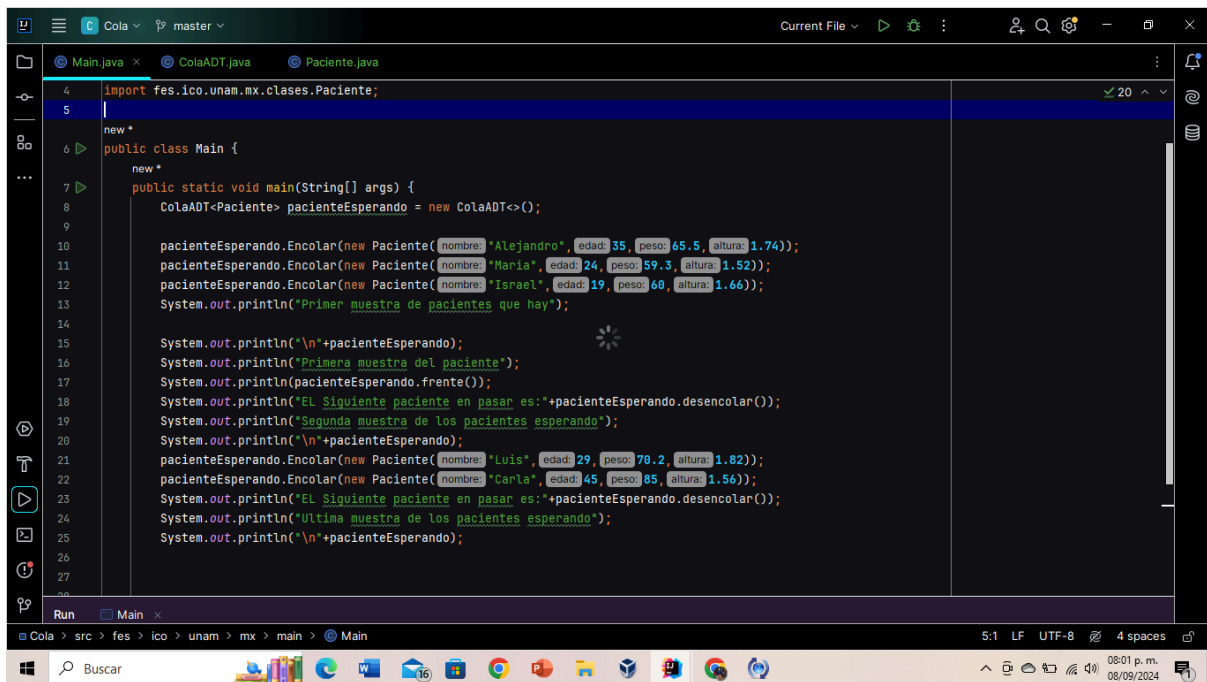


# Captura del código:



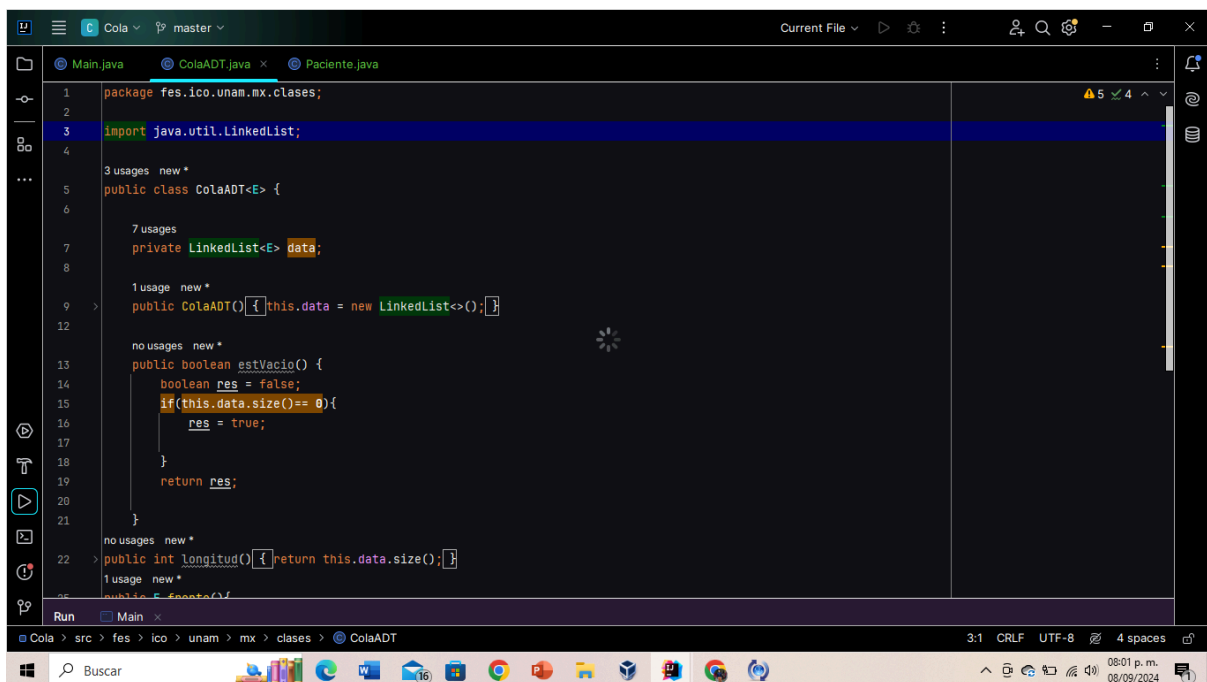
This screenshot shows the 'Main.java' file in an IDE. The code defines a 'Main' class with a 'main' method. It uses a 'ColaADT' (Queue ADT) to manage a list of 'Paciente' (Patient) objects. The 'main' method creates a 'ColaADT' instance, enqueues three patients (Alejandro, Maria, and Israel), prints the first patient, dequeues it, prints the next patient (Luis), dequeues him, prints the next patient (Carla), and finally dequeues her. The code is as follows:

```
import fes.ico.unam.mx.clases.Paciente;

new *
public class Main {
    new *
    public static void main(String[] args) {
        ColaADT<Paciente> pacienteEsperando = new ColaADT<>();

        pacienteEsperando.Encolar(new Paciente( nombres "Alejandro", edad: 35, peso: 65.5, altura: 1.74));
        pacienteEsperando.Encolar(new Paciente( nombres "Maria", edad: 24, peso: 59.3, altura: 1.52));
        pacienteEsperando.Encolar(new Paciente( nombres "Israel", edad: 19, peso: 60, altura: 1.66));
        System.out.println("Primer muestra de pacientes que hay");

        System.out.println("\n"+pacienteEsperando);
        System.out.println("Primera muestra del paciente");
        System.out.println(pacienteEsperando.frente());
        System.out.println("El siguiente paciente en pasar es:"+pacienteEsperando.desencolar());
        System.out.println("Segunda muestra de los pacientes esperando");
        System.out.println("\n"+pacienteEsperando);
        pacienteEsperando.Encolar(new Paciente( nombres "Luis", edad: 29, peso: 79.2, altura: 1.82));
        pacienteEsperando.Encolar(new Paciente( nombres "Carla", edad: 45, peso: 85, altura: 1.56));
        System.out.println("El siguiente paciente en pasar es:"+pacienteEsperando.desencolar());
        System.out.println("Ultima muestra de los pacientes esperando");
        System.out.println("\n"+pacienteEsperando);
    }
}
```



This screenshot shows the 'ColaADT.java' file in the same IDE. It defines the 'ColaADT' class, which uses a 'LinkedList' to store patient data. The class includes methods for enqueueing ('encolar'), dequeueing ('desencolar'), checking if the queue is empty ('estVacio'), and getting the front element ('frente'). The code is as follows:

```
package fes.ico.unam.mx.clases;

import java.util.LinkedList;

3 usages new *
public class ColaADT<E> {
    7 usages
    private LinkedList<E> data;

    1 usage new *
    public ColaADT() { this.data = new LinkedList<>(); }

    no usages new *
    public boolean estVacio() {
        boolean res = false;
        if (this.data.size() == 0) {
            res = true;
        }
        return res;
    }

    no usages new *
    public int longitud() { return this.data.size(); }

    1 usage new *
    public E frente() {
        if (this.estVacio()) {
            return null;
        }
        return this.data.getFirst();
    }
}
```

```
Cola > master > ColaADT.java
12
13 no usages new *
14 public boolean estVacio() {
15     boolean res = false;
16     if (this.data.size() == 0) {
17         res = true;
18     }
19     return res;
20 }
21
22 no usages new *
23 > public int longitud() { return this.data.size(); }
24
25 1 usage new *
26 public E frente() {
27     return this.data.getFirst();
28 }
29
30 5 usages new *
31 > public void Encolar(E valor) { this.data.addLast(valor); }
32
33 2 usages new *
34 > public E desencolar() { return this.data.removeFirst(); }
35
36 new *
37 @Override
38 public String toString() {
39     return "ColaADT{" +
40
Run Main
```

Cola > src > fes > ico > unam > mx > clases > ColaADT > toString 37:1 CRLF UTF-8 4 spaces

```
Cola > master > ColaADT.java
5 public class ColaADT<E> {
6     private List<E> data = new ArrayList<>();
7     private boolean res = true;
8
9     public ColaADT() {
10         res = true;
11     }
12
13     public boolean estVacio() {
14         return res;
15     }
16
17     public int longitud() {
18         return this.data.size();
19     }
20
21     public E frente() {
22         return this.data.getFirst();
23     }
24
25     public void Encolar(E valor) {
26         this.data.addLast(valor);
27     }
28
29     public E desencolar() {
30         return this.data.removeFirst();
31     }
32
33     @Override
34     public String toString() {
35         return "ColaADT{" +
36             "data=" + data +
37             "}";
38     }
39 }
40
41
42
43
Run Main
```

Cola > src > fes > ico > unam > mx > clases > ColaADT 42:1 CRLF UTF-8 4 spaces

The screenshot shows an IDE window with the file `Paciente.java` open. The code defines a `Paciente` class with the following structure:

```
1 package fes.ico.unam.mx.clases;
2
3 7 usages new *
4 public class Paciente {
5     4 usages
6     private String Nombre;
7     4 usages
8     private int Edad;
9     4 usages
10    private double Peso;
11    4 usages
12    private double Altura;
13
14    5 usages new *
15    public Paciente(String nombre, int edad, double peso, double altura) {
16        Nombre = nombre;
17        Edad = edad;
18        Peso = peso;
19        Altura = altura;
20    }
21
22    no usages new *
23    public String getNombre() {
24        return Nombre;
25    }
26 }
```

The status bar at the bottom indicates the file is at `Cola > src > fes > ico > unam > mx > clases > Paciente` with settings for 1:1 CRLF, UTF-8, and 4 spaces.

This screenshot shows the same IDE window with the `Paciente.java` file, displaying additional methods below the previous ones:

```
27
28
29    no usages new *
30    public void setNombre(String nombre) {
31        Nombre = nombre;
32    }
33
34    no usages new *
35    public int getEdad() {
36        return Edad;
37    }
38
39    no usages new *
40    public void setEdad(int edad) {
41        Edad = edad;
42    }
43 }
```

The status bar now shows the cursor is at `Cola > src > fes > ico > unam > mx > clases > Paciente > setEdad` with settings for 30:21 CRLF, UTF-8, and 4 spaces.

```
31  
32  
33 no usages new *  
34 public double getPeso() {  
35     return Peso;  
36 }  
37  
38 no usages new *  
39 public void setPeso(double peso) {  
40     Peso = peso;  
41 }  
42  
43 no usages new *  
44 public double getAltura() {  
45     return Altura;  
46 }  
47  
48 no usages new *  
49 public void setAltura(double altura) {  
50     Altura = altura;  
51 }  
52  
53 new *  
54 @Override  
55 public String toString() {  
56     return "Paciente{" +  
57         "Nombre=" + Nombre + '\n' +  
58         "Edad=" + Edad + '\n' +  
59         "Peso=" + Peso + '\n' +  
60         "Altura=" + Altura + '\n' +  
61         "}"  
62 }
```

Run Main

Cola > src > fes > ico > unam > mx > clases > Cola > Paciente > toString

51:22 CRLF UTF-8 4 spaces

```
"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.4\lib\idea_rt.jar=58350:C:\Program Files\JetBrains\JetBrains\IntelliJ IDEA 2023.3.4\bin" -Dfile.encoding=UTF-8  
Primer muestra de pacientes que hay  
ColaADT{data=[Paciente{Nombre='Alejandro', Edad=35, Peso=65.5, Altura=1.74}, Paciente{Nombre='Maria', Edad=24, Peso=59.3, Altura=1.52}, Paciente{Nombre='Israel', Edad=19, Peso=60.0, Altura=1.66}]}  
Primera muestra del paciente  
Paciente{Nombre='Alejandro', Edad=35, Peso=65.5, Altura=1.74}  
El Siguiente paciente en pasar es:Paciente{Nombre='Alejandro', Edad=35, Peso=65.5, Altura=1.74}  
Segunda muestra de los pacientes esperando  
ColaADT{data=[Paciente{Nombre='Maria', Edad=24, Peso=59.3, Altura=1.52}, Paciente{Nombre='Israel', Edad=19, Peso=60.0, Altura=1.66}]}  
El Siguiente paciente en pasar es:Paciente{Nombre='Maria', Edad=24, Peso=59.3, Altura=1.52}  
Ultima muestra de los pacientes esperando  
ColaADT{data=[Paciente{Nombre='Israel', Edad=19, Peso=60.0, Altura=1.66}, Paciente{Nombre='Luis', Edad=29, Peso=70.2, Altura=1.82}, Paciente{Nombre='Carla', Edad=28, Peso=68.0, Altura=1.78}]}  
Process finished with exit code 0
```

Cola > src > fes > ico > unam > mx > main > Main > main

21:48 LF UTF-8 4 spaces