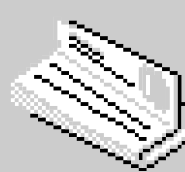
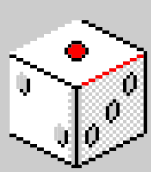


Ejercicio con el ADT

ColaConPrioridadA
cotada(Banco)



11:11PM

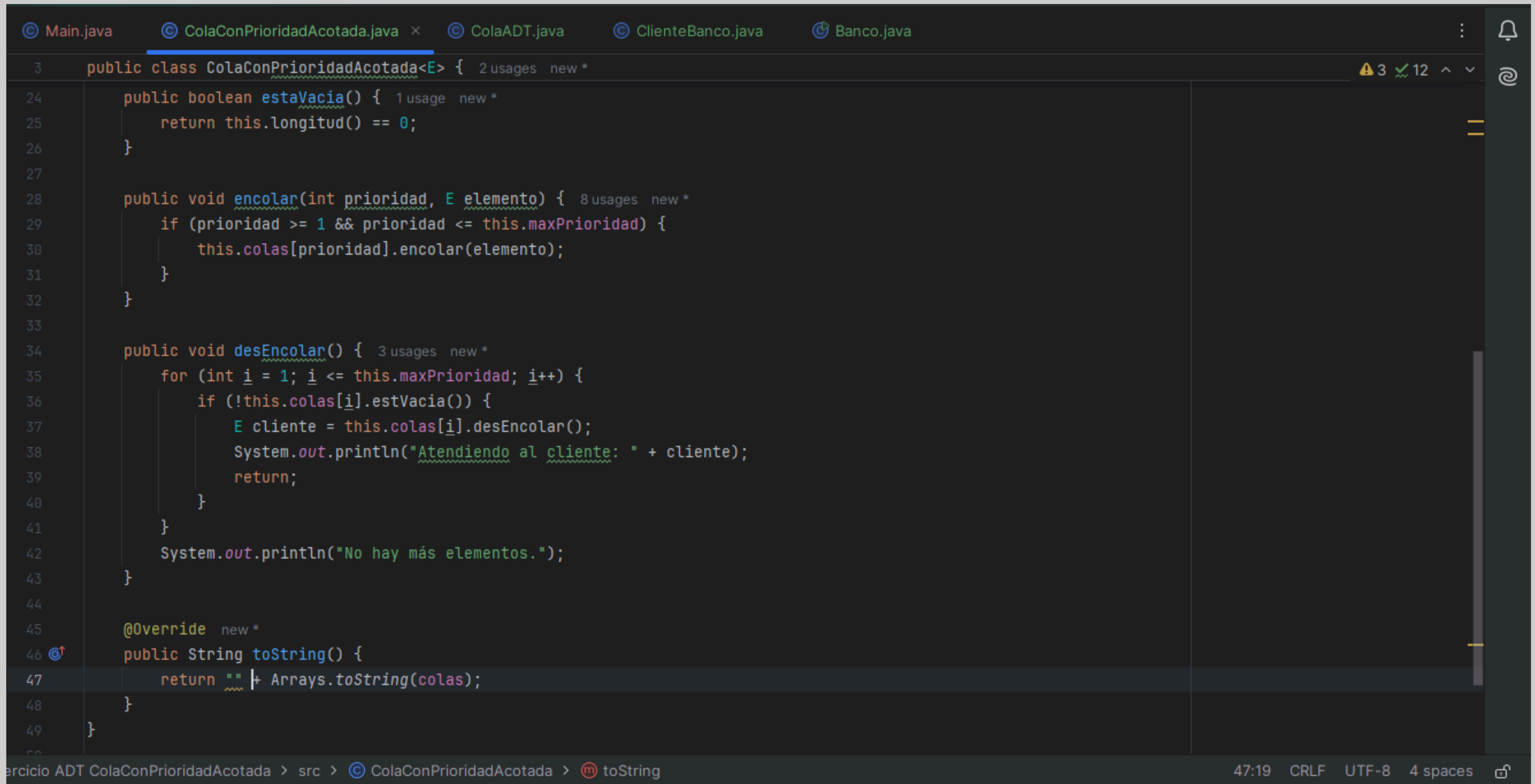
```

1  import java.util.Arrays;
2
3  public class ColaConPrioridadAcotada<E> { 2 usages new *
4      private ColaADT<E>[] colas; 7 usages
5      private int maxPrioridad; 5 usages
6
7      @SuppressWarnings("unchecked") 1 usage new *
8      public ColaConPrioridadAcotada(int maxPrioridad) {
9          this.maxPrioridad = maxPrioridad;
10         this.colas = new ColaADT[maxPrioridad + 1];
11         for (int i = 0; i < this.maxPrioridad + 1; i++) {
12             this.colas[i] = new ColaADT<>();
13         }
14     }
15
16     public int longitud() { 1 usage new *
17         int total = 0;
18         for (int i = 0; i < maxPrioridad + 1; i++) {
19             total += this.colas[i].longitud();
20         }
21         return total;
22     }
23
24     public boolean estaVacia() { 1 usage new *
25         return this.longitud() == 0;
26     }
27
28     public void encolar(int prioridad, E elemento) { 8 usages new *

```

ejercicio ADT ColaConPrioridadAcotada > src > ColaConPrioridadAcotada > toString

47:19 CRLF UTF-8 4 spaces



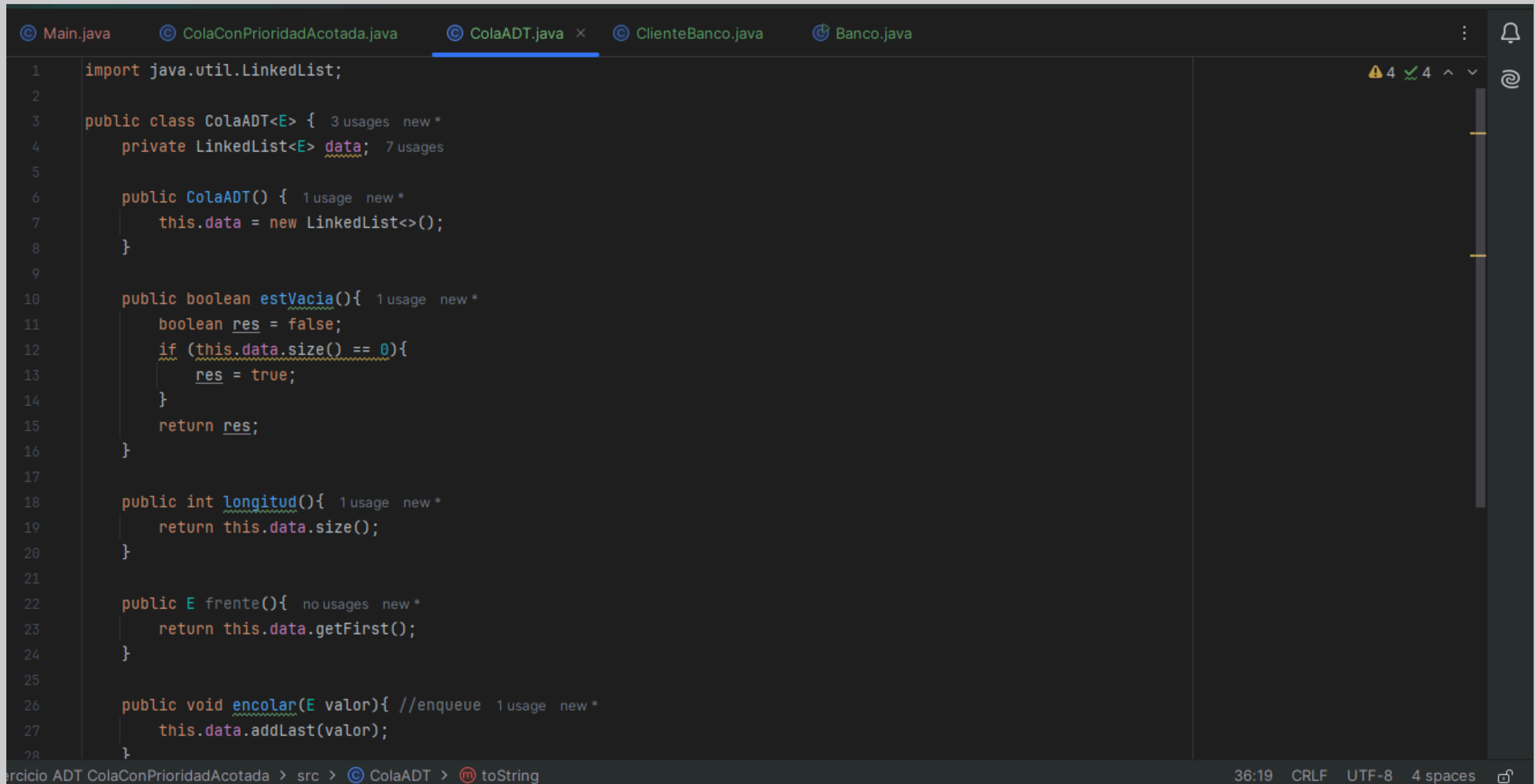
```

3  public class ColaConPrioridadAcotada<E> { 2 usages new *
24  public boolean estaVacia() { 1 usage new *
25      return this.longitud() == 0;
26  }
27
28  public void encolar(int prioridad, E elemento) { 8 usages new *
29      if (prioridad >= 1 && prioridad <= this.maxPrioridad) {
30          this.colas[prioridad].encolar(elemento);
31      }
32  }
33
34  public void desEncolar() { 3 usages new *
35      for (int i = 1; i <= this.maxPrioridad; i++) {
36          if (!this.colas[i].estVacia()) {
37              E cliente = this.colas[i].desEncolar();
38              System.out.println("Atendiendo al cliente: " + cliente);
39              return;
40          }
41      }
42      System.out.println("No hay más elementos.");
43  }
44
45  @Override new *
46  public String toString() {
47      return "" + Arrays.toString(colas);
48  }
49  }

```

ejercicio ADT ColaConPrioridadAcotada > src > ColaConPrioridadAcotada > toString

47:19 CRLF UTF-8 4 spaces

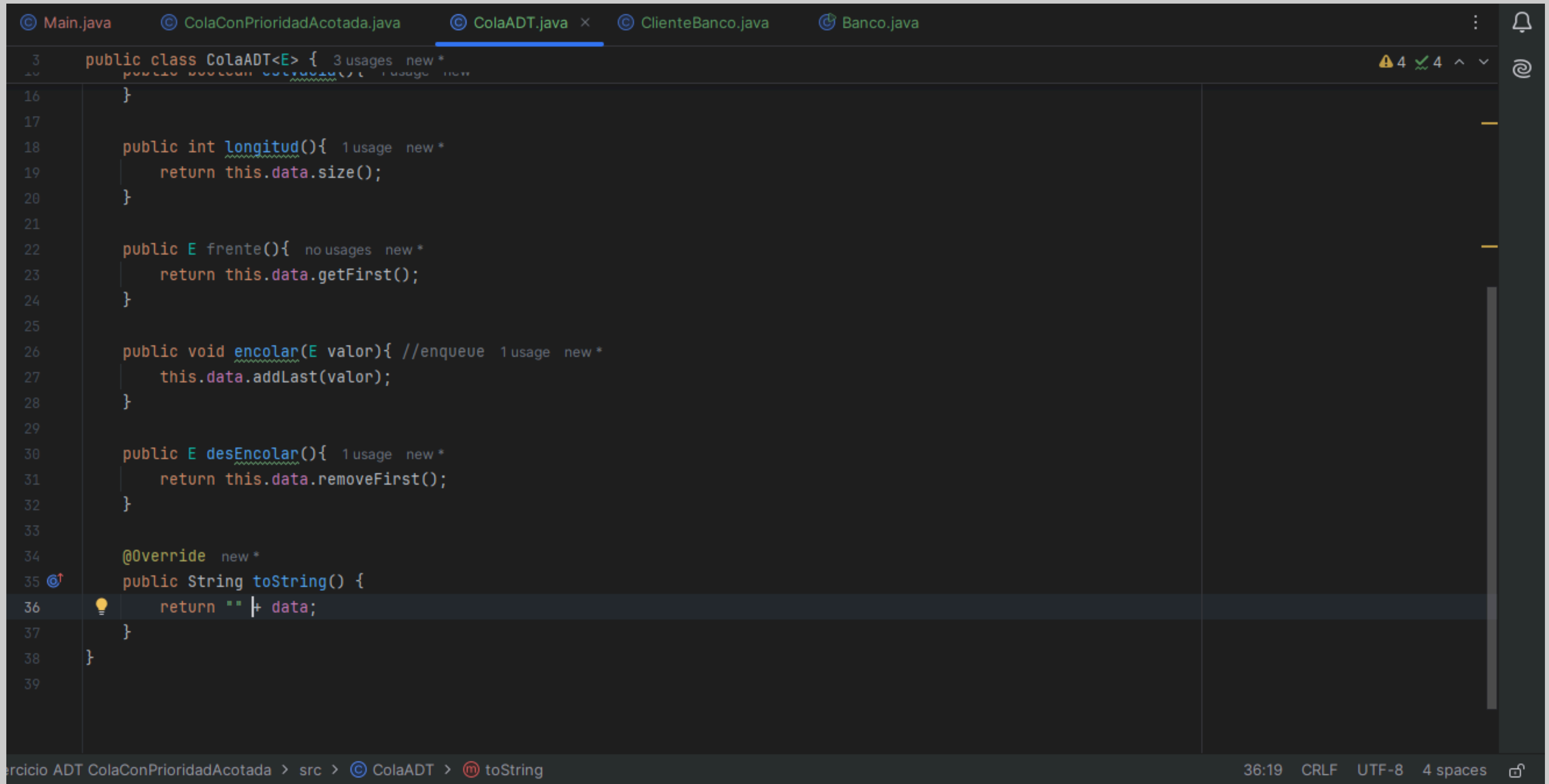


```

1  import java.util.LinkedList;
2
3  public class ColaADT<E> { 3 usages new *
4      private LinkedList<E> data; 7 usages
5
6      public ColaADT() { 1 usage new *
7          this.data = new LinkedList<>();
8      }
9
10     public boolean estVacia() { 1 usage new *
11         boolean res = false;
12         if (this.data.size() == 0) {
13             res = true;
14         }
15         return res;
16     }
17
18     public int longitud() { 1 usage new *
19         return this.data.size();
20     }
21
22     public E frente() { no usages new *
23         return this.data.getFirst();
24     }
25
26     public void encolar(E valor) { //enqueue 1 usage new *
27         this.data.addLast(valor);
28     }

```

ejercicio ADT ColaConPrioridadAcotada > src > ColaADT > toString
 36:19 CRLF UTF-8 4 spaces



```

3 public class ColaADT<E> { 3 usages new *
16 }
17
18 public int longitud(){ 1 usage new *
19     return this.data.size();
20 }
21
22 public E frente(){ no usages new *
23     return this.data.getFirst();
24 }
25
26 public void encolar(E valor){ //enqueue 1 usage new *
27     this.data.addLast(valor);
28 }
29
30 public E desEncolar(){ 1 usage new *
31     return this.data.removeFirst();
32 }
33
34 @Override new *
35 public String toString() {
36     return "" + data;
37 }
38 }
39

```

ercicio ADT ColaConPrioridadAcotada > src > ColaADT > toString 36:19 CRLF UTF-8 4 spaces

```

1  public class ClienteBanco { 17 usages new *
2      private String nombre; 5 usages
3      private String tipoCliente; 3 usages
4      private double saldo; 4 usages
5
6      public ClienteBanco(String nombre, String tipoCliente, double saldo) { 8 usages new *
7          this.nombre = nombre;
8          this.tipoCliente = tipoCliente;
9          this.saldo = saldo;
10     }
11
12     public String getNombre() { no usages new *
13         return nombre;
14     }
15
16     public String getTipoCliente() { no usages new *
17         return tipoCliente;
18     }
19
20     public void retirar(double cantidad) { 1 usage new *
21         if (saldo >= cantidad) {
22             saldo -= cantidad;
23             System.out.println(nombre + " retiró $" + cantidad);
24         } else {
25             System.out.println(nombre + " no tiene suficiente saldo.");
26         }
27     }
28

```

ejercicio ADT ColaConPrioridadAcotada > src > ClienteBanco > toString 31:60 CRLF UTF-8 4 spaces

```

Main.java ColaConPrioridadAcotada.java ColaADT.java ClienteBanco.java x Banco.java
1 public class ClienteBanco { 17 usages new *
6     public ClienteBanco(String nombre, String tipoCliente, double saldo) { 8 usages new *
9         this.saldo = saldo;
10    }
11
12    public String getNombre() { no usages new *
13        return nombre;
14    }
15
16    public String getTipoCliente() { no usages new *
17        return tipoCliente;
18    }
19
20    public void retirar(double cantidad) { 1 usage new *
21        if (saldo >= cantidad) {
22            saldo -= cantidad;
23            System.out.println(nombre + " retiró $" + cantidad);
24        } else {
25            System.out.println(nombre + " no tiene suficiente saldo.");
26        }
27    }
28
29    @Override new *
30    public String toString() {
31        return nombre + " (" + tipoCliente + ") " + "saldo: " + " (" + saldo + ")";
32    }
33 }
34

```

ercicio ADT ColaConPrioridadAcotada > src > ClienteBanco > toString 31:60 CRLF UTF-8 4 spaces


```

Main.java ColaConPrioridadAcotada.java ColaADT.java ClienteBanco.java Banco.java x
1 public class Banco { new *
2     public static void main(String[] args) { new *
3         ColaConPrioridadAcotada<ClienteBanco> colaBanco = new ColaConPrioridadAcotada<>( maxPrioridad: 5);
4
5         System.out.println("Llegan 2 clientes nuevos (Prioridad baja):");
6         ClienteBanco cliente1 = new ClienteBanco( nombre: "Juan", tipoCliente: "Cliente nuevo", saldo: 5000);
7         ClienteBanco cliente2 = new ClienteBanco( nombre: "Ana", tipoCliente: "Cliente nuevo", saldo: 7000);
8         colaBanco.encolar( prioridad: 4, cliente1);
9         colaBanco.encolar( prioridad: 4, cliente2);
10        System.out.println(cliente1);
11        System.out.println(cliente2);
12        System.out.println("-----");
13
14        System.out.println("Llegan 3 personas que no son clientes (Prioridad muy baja):");
15        ClienteBanco cliente3 = new ClienteBanco( nombre: "Pedro", tipoCliente: "No es cliente", saldo: 0);
16        ClienteBanco cliente4 = new ClienteBanco( nombre: "Luis", tipoCliente: "No es cliente", saldo: 0);
17        ClienteBanco cliente5 = new ClienteBanco( nombre: "Mario", tipoCliente: "No es cliente", saldo: 0);
18        colaBanco.encolar( prioridad: 5, cliente3);
19        colaBanco.encolar( prioridad: 5, cliente4);
20        colaBanco.encolar( prioridad: 5, cliente5);
21        System.out.println(cliente3);
22        System.out.println(cliente4);
23        System.out.println(cliente5);
24        System.out.println("-----");
25
26        System.out.println("Llega una celebridad (Prioridad muy alta):");
27        ClienteBanco celebridad = new ClienteBanco( nombre: "Carlos", tipoCliente: "Celebridad", saldo: 100000);
28        colaBanco.encolar( prioridad: 1, celebridad);

```

ejercicio ADT ColaConPrioridadAcotada > src > Banco > main 64:29 CRLF UTF-8 4 spaces


```

© Main.java  © ColaConPrioridadAcotada.java  © ColaADT.java  © ClienteBanco.java  Banco.java x
1  public class Banco { new *
2      public static void main(String[] args) { new *
26      System.out.println("Llega una celebridad (Prioridad muy alta):");
27      ClienteBanco celebridad = new ClienteBanco( nombre: "Carlos", tipoCliente: "Celebridad", saldo: 100000);
28      colaBanco.encolar( prioridad: 1, celebridad);
29      System.out.println(celebridad);
30      System.out.println("-----");
31
32      System.out.println("Estado de la cola con prioridad acotada:");
33      System.out.println(colaBanco);
34      System.out.println("-----");
35
36      System.out.println("Atender al siguiente cliente (la celebridad) y retira $10,000:");
37      colaBanco.desEncolar();
38      celebridad.retirar( cantidad: 10000);
39      System.out.println("-----");
40
41      System.out.println("Llegan dos clientes más (uno frecuente y uno premium):");
42      ClienteBanco cliente6 = new ClienteBanco( nombre: "Sofia", tipoCliente: "Cliente frecuente", saldo: 20000);
43      ClienteBanco cliente7 = new ClienteBanco( nombre: "Miguel", tipoCliente: "Cliente premium", saldo: 50000);
44      colaBanco.encolar( prioridad: 3, cliente6);
45      colaBanco.encolar( prioridad: 2, cliente7);
46      System.out.println(cliente6);
47      System.out.println(cliente7);
48      System.out.println("-----");
49
50      System.out.println("Atender al siguiente cliente:");

```

ejercicio ADT ColaConPrioridadAcotada > src > Banco > main 50:50 CRLF UTF-8 4 spaces

```

© Main.java  © ColaConPrioridadAcotada.java  © ColaADT.java  © ClienteBanco.java  Banco.java x
1  public class Banco { new *
2      public static void main(String[] args) { new *
46      System.out.println(cliente6);
47      System.out.println(cliente7);
48      System.out.println("-----");
49
50      System.out.println("Atender al siguiente cliente:");
51      colaBanco.desEncolar();
52      System.out.println("-----");
53
54      System.out.println("Estado de la cola después de dos atenciones:");
55      System.out.println(colaBanco);
56      System.out.println("-----");
57
58      System.out.println("Atender a todos los clientes restantes:");
59      while (!colaBanco.estaVacia()) {
60          colaBanco.desEncolar();
61      }
62      System.out.println("-----");
63
64      System.out.println("Estado final de la cola:");
65      System.out.println(colaBanco);
66  }
67  }
68

```

pricio ADT ColaConPrioridadAcotada > src > Banco > main 50:50 CRLF UTF-8 4 spaces

The screenshot shows an IDE with several Java files open: Main.java, ColaConPrioridadAcotada.java, ColaADT.java, ClienteBanco.java, and Banco.java. The Banco.java file is active, showing a while loop that calls colaBanco.desEncolar() while colaBanco.estavacia() is true. Below the code editor is a Run console window showing the execution output. The output includes messages about new customers (Juan, Ana), non-customers (Pedro, Luis, Mario), a celebrity (Carlos), and the state of the queue with priority acotada. It also shows the service of Carlos, withdrawing \$10,000, and the arrival of two more customers (Sofia, Miguel).

```

59 while (!colaBanco.estavacia()) {
60     colaBanco.desEncolar();
61 }

```

Run Banco

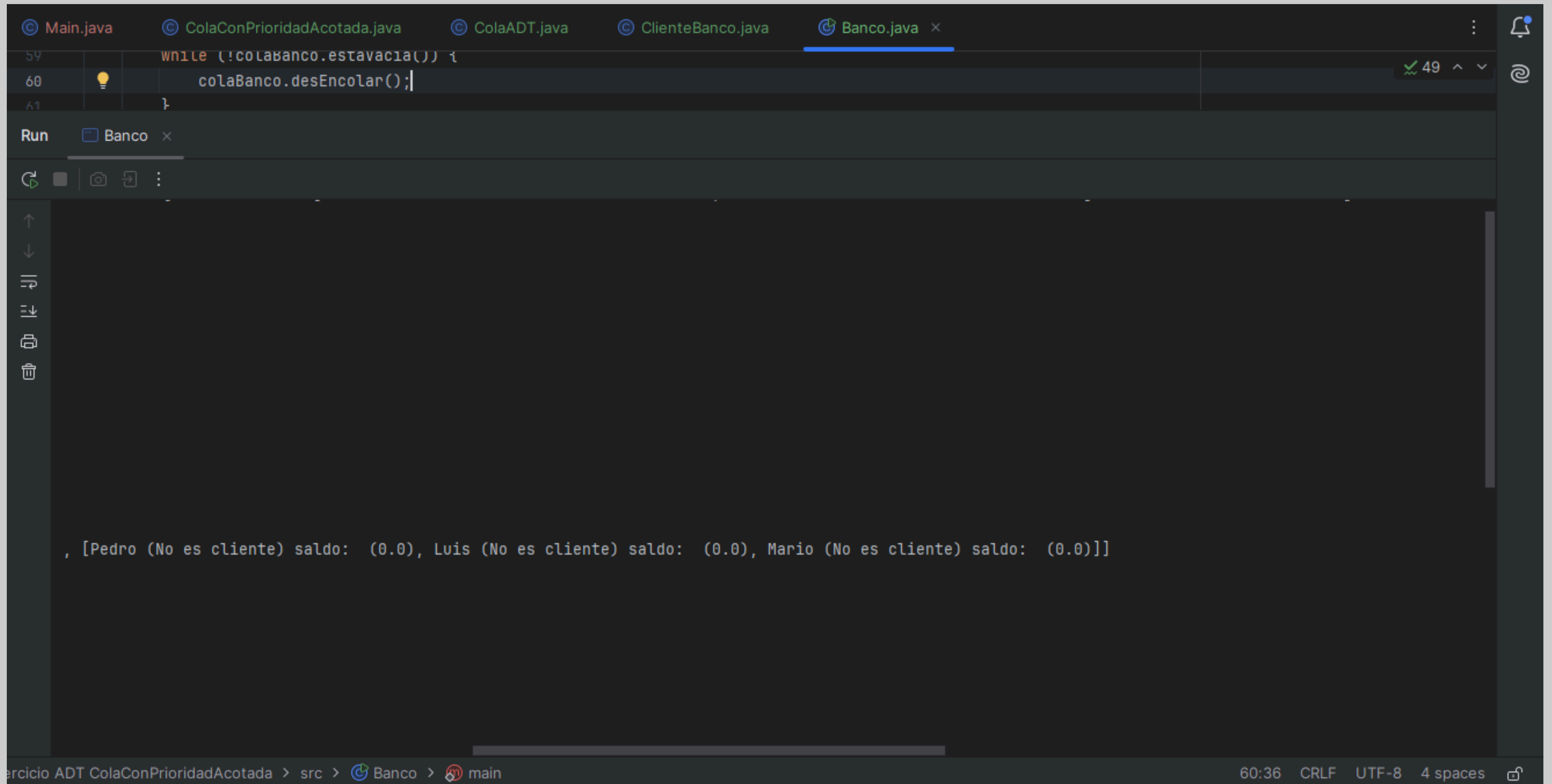
```

↑ Llegan 2 clientes nuevos (Prioridad baja):
↓ Juan (Cliente nuevo) saldo: (5000.0)
  Ana (Cliente nuevo) saldo: (7000.0)
  -----
  ↓ Llegan 3 personas que no son clientes (Prioridad muy baja):
    Pedro (No es cliente) saldo: (0.0)
    Luis (No es cliente) saldo: (0.0)
    Mario (No es cliente) saldo: (0.0)
    -----
    ↓ Llegan una celebridad (Prioridad muy alta):
      Carlos (Celebridad) saldo: (100000.0)
      -----
      ↓ Estado de la cola con prioridad acotada:
        [[], [Carlos (Celebridad) saldo: (100000.0)], [], [], [Juan (Cliente nuevo) saldo: (5000.0), Ana (Cliente nuevo) saldo: (7000.0)], [Pedro (No es cliente) saldo: (0.0), Luis (No es cliente) saldo: (0.0), Mario (No es cliente) saldo: (0.0)]]
        -----
        ↓ Atender al siguiente cliente (la celebridad) y retira $10,000:
          Atendiendo al cliente: Carlos (Celebridad) saldo: (100000.0)
          Carlos retiró $10000.0
          -----
          ↓ Llegan dos clientes más (uno frecuente y uno premium):
            Sofia (Cliente frecuente) saldo: (20000.0)
            Miguel (Cliente premium) saldo: (50000.0)

```

ejercicio ADT ColaConPrioridadAcotada > src > Banco > main 60:36 CRLF UTF-8 4 spaces

on the featured statistic • Elaborate on the featured statistic • Elaborate on the featured statistic • Elaborate on the featur



The screenshot shows an IDE with several Java files open: Main.java, ColaConPrioridadAcotada.java, ColaADT.java, ClienteBanco.java, and Banco.java. The Banco.java file is active, showing a while loop that calls colaBanco.desEncolar(). The console output displays the state of a queue: [Pedro (No es cliente) saldo: (0.0), Luis (No es cliente) saldo: (0.0), Mario (No es cliente) saldo: (0.0)].

```
59 while (!colaBanco.estavacia()) {
60     colaBanco.desEncolar();
61 }
```

Run Banco

[Pedro (No es cliente) saldo: (0.0), Luis (No es cliente) saldo: (0.0), Mario (No es cliente) saldo: (0.0)]

ejercicio ADT ColaConPrioridadAcotada > src > Banco > main 60:36 CRLF UTF-8 4 spaces

on the featured statistic • Elaborate on the featured statistic • Elaborate on the featured statistic • Elaborate on the featur

The screenshot shows an IDE with several Java files open: Main.java, ColaConPrioridadAcotada.java, ColaADT.java, ClienteBanco.java, and Banco.java. The Banco.java file is active, showing a while loop that calls colaBanco.desEncolar(). Below the code editor is a Run console window displaying the execution output. The output shows the arrival of two clients (Sofia and Miguel), the service of Miguel, the state of the queue after two services, the service of all remaining clients (Sofia, Juan, Ana, Pedro, Luis, Mario), and the final state of the queue.

```

59 while (!colaBanco.estavacia()) {
60     colaBanco.desEncolar();
61 }

```

Run Banco x

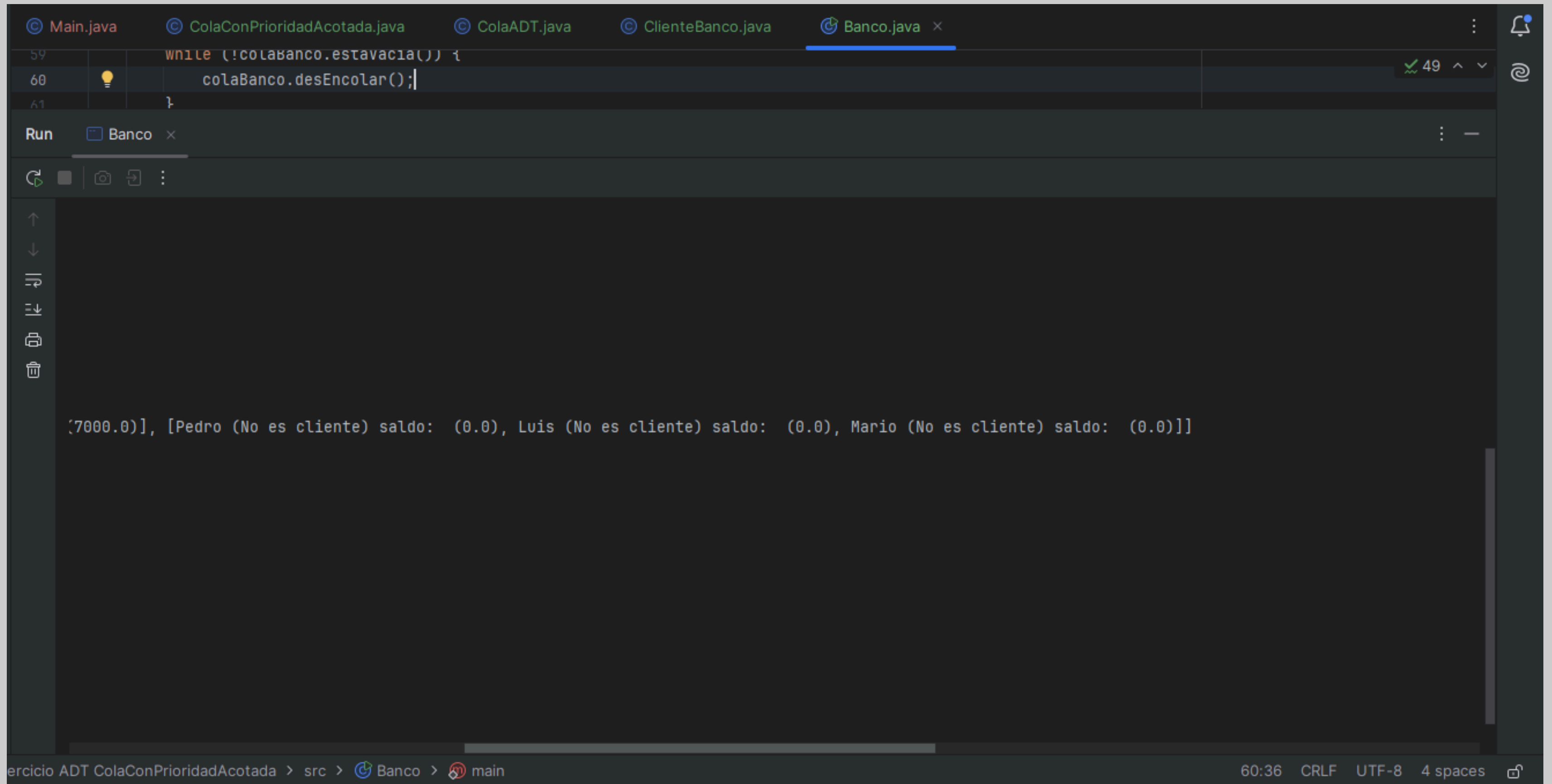
```

Llegan dos clientes más (uno frecuente y uno premium):
Sofia (Cliente frecuente) saldo: (20000.0)
Miguel (Cliente premium) saldo: (50000.0)
-----
Atender al siguiente cliente:
Atendiendo al cliente: Miguel (Cliente premium) saldo: (50000.0)
-----
Estado de la cola después de dos atenciones:
[[], [], [], [Sofia (Cliente frecuente) saldo: (20000.0)], [Juan (Cliente nuevo) saldo: (5000.0), Ana (Cliente nuevo) saldo: (7000.0)], [Pedro (No es c
-----
Atender a todos los clientes restantes:
Atendiendo al cliente: Sofia (Cliente frecuente) saldo: (20000.0)
Atendiendo al cliente: Juan (Cliente nuevo) saldo: (5000.0)
Atendiendo al cliente: Ana (Cliente nuevo) saldo: (7000.0)
Atendiendo al cliente: Pedro (No es cliente) saldo: (0.0)
Atendiendo al cliente: Luis (No es cliente) saldo: (0.0)
Atendiendo al cliente: Mario (No es cliente) saldo: (0.0)
-----
Estado final de la cola:
[[], [], [], [], [], []]

```

ejercicio ADT ColaConPrioridadAcotada > src > Banco > main 60:36 CRLF UTF-8 4 spaces

on the featured statistic • Elaborate on the featured statistic • Elaborate on the featured statistic • Elaborate on the featur



The screenshot shows an IDE with several Java files open: Main.java, ColaConPrioridadAcotada.java, ColaADT.java, ClienteBanco.java, and Banco.java. The Banco.java file is active, showing a while loop that checks if the bank is not empty and then calls desEncolar(). The console output shows the state of the bank and the status of three clients: Pedro, Luis, and Mario, all of whom are not clients and have a balance of 0.0. The bottom status bar indicates the file path, line number (60:36), and encoding (CRLF, UTF-8, 4 spaces).

```
59 while (!colaBanco.estavacia()) {  
60     colaBanco.desEncolar();  
61 }
```

```
[7000.0]], [Pedro (No es cliente) saldo: (0.0), Luis (No es cliente) saldo: (0.0), Mario (No es cliente) saldo: (0.0)]
```

ejercicio ADT ColaConPrioridadAcotada > src > Banco > main 60:36 CRLF UTF-8 4 spaces

on the featured statistic • Elaborate on the featured statistic • Elaborate on the featured statistic • Elaborate on the featur