

## CLASE NODO:

src/clase2/NodoPacientes.java - Eclipse IDE

or Navigate Search Project Run Window Help

Type Hierarchy

JavaSE-21]

ociales.csv

ntos.java

igada.java

u.java

la2.java

la3.java

ada.java

va

ava

iva

s.java

shlo.java

```
1 package clase2;
2
3 class NodoPacientes<T> {
4     private T data;
5     private Nodo<T> siguiente;
6
7     public NodoPacientes(T data) {
8         this.data = data;
9         this.siguiente = null;
10    }
11
12    public T getData() {
13        return data;
14    }
15
16    public void setData(T data) {
17        this.data = data;
18    }
19
20    public Nodo<T> getSiguiente() {
21        return siguiente;
22    }
23
24    public void setSiguiente(Nodo<T> siguiente) {
25        this.siguiente = siguiente;
26    }
27
28    @Override
29    public String toString() {
30        return data.toString();
31    }
32 }
```

Task List

Find

Outline

- clase2
  - NodoPacientes<T>
    - data : T
    - siguiente : No
    - getData() : T
    - setData(T) : void
    - getSiguiente() : Nodo<T>
    - setSiguiente(Nodo<T>) : void
    - toString() : String

Problems @ Javadoc Declaration Console

<terminated> ListaLigadaP [Java Application] C:\Users\lizet\Downloads\eclipse-dsl-2025-06-R-win32-x86\_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.v

Writable Smart Insert 32 : 2 : 590 186M of 448M

Buscar

## CLASE PACIENTES:

src/clase2/Pacientes.java - Eclipse IDE

File | Edit | Format | Source | Run | Window | Help

Type Hierarchy

Package Hierarchy

src/clase2/Pacientes.java

```
1 package clase2;
2
3 class Pacientes {
4     private String nombre;
5     private int edad;
6     private double estatura;
7
8     public Pacientes(String nombre, int edad, double estatura) {
9         this.nombre = nombre;
10        this.edad = edad;
11        this.estatura = estatura;
12    }
13
14    @Override
15    public String toString() {
16        return "Paciente{" +
17            "nombre='" + nombre + '\'' +
18            ", edad=" + edad +
19            ", estatura=" + estatura + 'm' + '}';
20    }
21 }
22
23
```

Task List

Find

Outline

- clase2
  - Pacientes
    - nombre : String
    - edad : int
    - estatura : double
    - Pacientes(String nombre, int edad, double estatura)
    - toString() : String

Problems | Javadoc | Declaration | Console

<terminated> ListaLigadaP [Java Application] C:\Users\lizet\Downloads\eclipse-dsl-2025-06-R-win32-x86\_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.v

Writable | Smart Insert | 23 : 1 : 515 | 259M of 448M

Buscar

## CLASE LISTA LIGADA:

clase2/ListaLigadaP.java - Eclipse IDE

Navigate Search Project Run Window Help

Package Hierarchy

ent - C:\Users\lizet\Downlo  
- C:\Users\lizet\Downloa  
Jsers\lizet\Downloads\ec  
Jsers\lizet\Downloads\ec  
et\Downloads\eclipse-dsl  
C:\Users\lizet\Downloads\l  
s\lizet\Downloads\eclipse  
zet\Downloads\eclipse-ds  
Users\lizet\Downloads\ec  
Users\lizet\Downloads\ec  
ktop - C:\Users\lizet\Downr  
rs\lizet\Downloads\ecips  
zet\Downloads\eclipse-ds

SE-21]

ales.csv

s.java  
da.java  
ra  
java  
java  
u.java

va

visu

```
1 package clase2;  
2  
3 public class ListaLigadaP<T> {  
4     private Nodo<T> head;  
5     private Nodo<T> tail;  
6     private int tamanio = 0;  
7  
8     public ListaLigadaP() {  
9         head = null;  
10        tail = null;  
11    }  
12  
13    // Verificar si la cola está vacía  
14    public boolean estaVacia() {  
15        return head == null;  
16    }  
17  
18    public int getTamanio() {  
19        return tamanio;  
20    }  
21  
22    // Registrar llegada (agregar al final)  
23    public void registrarPaciente(T valor) {  
24        Nodo<T> nuevo = new Nodo<>(valor);  
25        if (estaVacia()) {  
26            head = nuevo;  
27            tail = nuevo;  
28        } else {  
29            tail.setSiguiente(nuevo);  
30            tail = nuevo;  
31        }  
32        tamanio++;  
33        System.out.println("\nPaciente registrado: " + valor );  
34    }  
35  
36    // Atender al paciente al frente  
37    public T atenderPaciente() {  
38        if (estaVacia()) {  
39            System.out.println("\n No hay pacientes en espera.");  
40            return null;  
41        }  
42        T valor = head.getData();  
43        head = head.getSiguiente();
```

Task List

Find

Outline

- clase2
  - ListaLigadaP<T>:
    - head : Nodo<T>
    - tail : Nodo<T>
    - tamanio : int
    - ListaLigadaP
    - estaVacia() : boolean
    - getTamanio() : int
    - registrarPaciente(T valor)
    - atenderPaciente() : T
    - mostrarCola()
    - verSiguiente()
    - main(String[] args)

Problems @ Javadoc Declaration Console

<terminated> ListaLigadaP [Java Application] C:\Users\lizet\Downloads\eclipse-dsl-2025-06-R-win32-x86\_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.v

Writable Smart Insert 93 : 41 : 2812 283M of 448M

Buscar

The screenshot shows the Eclipse IDE with the following components:

- Editor:** Displays the code for `ListaLigadaP.java`. The code implements a linked list with methods for adding, removing, and displaying nodes. It includes comments in Spanish and uses `System.out.println` for output.
- Left Sidebar:** Contains the 'Type Hierarchy' and 'Project Explorer' views. The 'Project Explorer' shows the project structure with files like `ListaLigadaP.java`, `ListaDobleLi...`, `PruebaListaD...`, `NodoPaciente...`, `*Pacientes.java`, and `ListaLigada...`.
- Right Sidebar:** Contains the 'Task List' and 'Outline' views. The 'Outline' view shows the class structure of `ListaLigadaP<T>` with methods like `head`, `tail`, `tamano`, `estaVacia()`, `getTamano()`, `registrarPacien`, `atenderPacien`, `mostrarCola()`, `verSiguiente()`, and `main(String[])`.
- Bottom Panel:** Contains the 'Problems', 'Javadoc', 'Declaration', and 'Console' views. The 'Console' view shows the output of the program, including the message `<terminated> ListaLigadaP [Java Application]`.

```
40         return null;
41     }
42     T valor = head.getData();
43     head = head.getSiguiente();
44     tamano--;
45     if (head == null) tail = null; // si la lista queda vacía
46     System.out.println("\n Atendiendo a: " + valor );
47     return valor;
48 }
49
50 // Mostrar estado de la cola
51 public void mostrarCola() {
52     if (estaVacia()) {
53         System.out.println("\n No hay pacientes esperando.");
54         return;
55     }
56     System.out.println("\n Pacientes en espera:");
57     Nodo<T> actual = head;
58     while (actual != null) {
59         System.out.println("  → " + actual.getData());
60         actual = actual.getSiguiente();
61     }
62 }
63
64 // Método extra para ver al siguiente en la fila sin eliminarlo
65 public T verSiguiente() {
66     if (estaVacia()) {
67         System.out.println("\n No hay pacientes en espera.");
68         return null;
69     }
70     return head.getData();
71 }
72
73 // Código en consola
74 public static void main(String[] args) {
75     ListaLigadaP<Pacientes> colaPacientes = new ListaLigadaP<>();
76
77     System.out.println("\n ¿Cola vacía? " + colaPacientes.estaVacia());
78
79     // Registrar pacientes
80     colaPacientes.registrarPaciente(new Pacientes("Fernanda Lara", 30, 1.65 ));
81     colaPacientes.registrarPaciente(new Pacientes("Carlos Ruiz", 45, 1.80 ));
82     colaPacientes.registrarPaciente(new Pacientes("Lucía Gómez", 25, 1.70 ));
```



The screenshot shows the Eclipse IDE with the following components:

- Left Panel (Project Explorer):** Displays the project structure with files like `agent.java`, `social.csv`, `Intos.java`, `ligada.java`, `a.java`, `la2.java`, `la3.java`, `ada.java`, `va`, `ava`, `es.java`, and `ble.java`.
- Editor:** Shows the code for `ListaLigadaP.java`. The code includes:
  - Line 60: `actual = actual.getSiguiente();`
  - Line 61: `}`
  - Line 62: `}`
  - Line 63: `// Método extra para ver al siguiente en la fila sin eliminarlo`
  - Line 64: `public T verSiguiente() {`
  - Line 65: `if (estaVacia()) {`
  - Line 66: `System.out.println("\n No hay pacientes en espera.");`
  - Line 67: `return null;`
  - Line 68: `}`
  - Line 69: `return head.getData();`
  - Line 70: `}`
  - Line 71: `// Código en consola`
  - Line 72: `public static void main(String[] args) {`
  - Line 73: `ListaLigadaP<Pacientes> colaPacientes = new ListaLigadaP<>();`
  - Line 74: `System.out.println("\n ¿Cola vacía? " + colaPacientes.estaVacia());`
  - Line 75: `// Registrar pacientes`
  - Line 76: `colaPacientes.registrarPaciente(new Pacientes("Fernanda Lara", 30, 1.65 ));`
  - Line 77: `colaPacientes.registrarPaciente(new Pacientes("Carlos Ruiz", 45, 1.80 ));`
  - Line 78: `colaPacientes.registrarPaciente(new Pacientes("Lucía Gómez", 25, 1.70 ));`
  - Line 79: `colaPacientes.mostrarCola();`
  - Line 80: `// Ver siguiente paciente`
  - Line 81: `System.out.println("\n Siguiente paciente: " + colaPacientes.verSiguiente());`
  - Line 82: `// Atender a los pacientes`
  - Line 83: `colaPacientes.atenderPaciente();`
  - Line 84: `colaPacientes.mostrarCola();`
  - Line 85: `colaPacientes.atenderPaciente();`
  - Line 86: `colaPacientes.mostrarCola();`
  - Line 87: `colaPacientes.atenderPaciente();`
  - Line 88: `colaPacientes.atenderPaciente(); // cuando ya no hay pacientes`
  - Line 89: `System.out.println("\n ¿Cola vacía? " + colaPacientes.estaVacia());`
  - Line 90: `}`
  - Line 91: `}`
  - Line 92: `}`
  - Line 93: `}`
  - Line 94: `}`
  - Line 95: `}`
  - Line 96: `}`
  - Line 97: `}`
  - Line 98: `}`
  - Line 99: `}`
  - Line 100: `}`
  - Line 101: `}`
  - Line 102: `}`
- Right Panel (Outline):** Shows the class structure with methods like `head : Nodo<`, `tail : Nodo<T:`, `tamano : int`, `ListaLigadaP()`, `estaVacia() : b`, `getTamano()`, `registrarPacien`, `atenderPacien`, `mostrarCola()`, `verSiguiente()`, and `main(String[])`.
- Bottom Panel (Console):** Shows the output of the program, including the message `<terminated> ListaLigadaP [Java Application] C:\Users\lizet\Downloads\eclipse-dsl-2025-06-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.v`.



## CONSOLA:

```
rc/clase2/ListaLigadaP.java - Eclipse IDE
or Navigate Search Project Run Window Help

Type Hierarchy
agent - C:\Users\lizet\Downl...
jfr - C:\Users\lizet\Downl...
C:\Users\lizet\Downloads\ec...
C:\Users\lizet\Downloads\ec...
lizet\Downloads\eclipse-ds...
C:\Users\lizet\Downloads\...
Users\lizet\Downloads\eclip...
\lizet\Downloads\eclipse-ds...
C:\Users\lizet\Downloads\ec...
C:\Users\lizet\Downloads\ec...
C:\Users\lizet\Downloads\ec...
esktop - C:\Users\lizet\Downr...
Users\lizet\Downloads\ecli...
s\lizet\Downloads\eclipse-d...

avaSE-21]

ociales.csv

ntos.java
gada.java
.java
a2.java
a3.java
da.java
ra
iva
va
s.java
blains

Problems @ Javadoc Declaration Console X
<terminated> ListaLigadaP [Java Application] C:\Users\lizet\Downloads\eclipse-dsl-2025-06-R-win32-x86_64\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.v...

¿Cola vacía? true
Paciente registrado: Paciente{nombre='Fernanda Lara', edad=30, estatura=1.65m}
Paciente registrado: Paciente{nombre='Carlos Ruiz', edad=45, estatura=1.8m}
Paciente registrado: Paciente{nombre='Lucía Gómez', edad=25, estatura=1.7m}

Pacientes en espera:
→ Paciente{nombre='Fernanda Lara', edad=30, estatura=1.65m}
→ Paciente{nombre='Carlos Ruiz', edad=45, estatura=1.8m}
→ Paciente{nombre='Lucía Gómez', edad=25, estatura=1.7m}

Siguiente paciente: Paciente{nombre='Fernanda Lara', edad=30, estatura=1.65m}
Atendiendo a: Paciente{nombre='Fernanda Lara', edad=30, estatura=1.65m}

Pacientes en espera:
→ Paciente{nombre='Carlos Ruiz', edad=45, estatura=1.8m}
→ Paciente{nombre='Lucía Gómez', edad=25, estatura=1.7m}

Atendiendo a: Paciente{nombre='Carlos Ruiz', edad=45, estatura=1.8m}

Pacientes en espera:
→ Paciente{nombre='Lucía Gómez', edad=25, estatura=1.7m}

Atendiendo a: Paciente{nombre='Lucía Gómez', edad=25, estatura=1.7m}

No hay pacientes en espera.
¿Cola vacía? true

Outline
clase2
  ListaLigadaP<T>
    head : Nodo<T>
    tail : Nodo<T>
    tamaño : int
    ListaLigadaP()
    estaVacia() : boolean
    getTamaño()
    registrarPaciente(Paciente)
    atenderPaciente()
    mostrarCola()
    verSiguiente()
    main(String[])
```

GitHub: <https://github.com/lizz-mond35/Estructura-de-Datos.git>