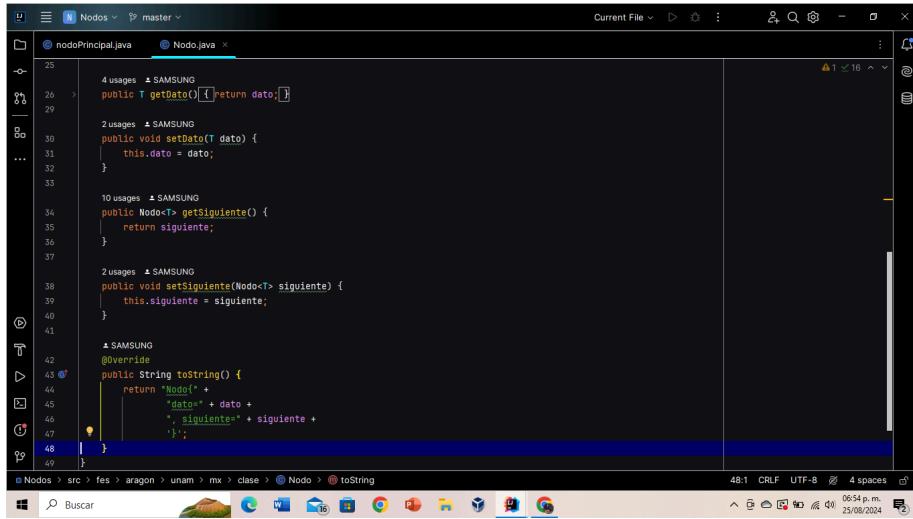
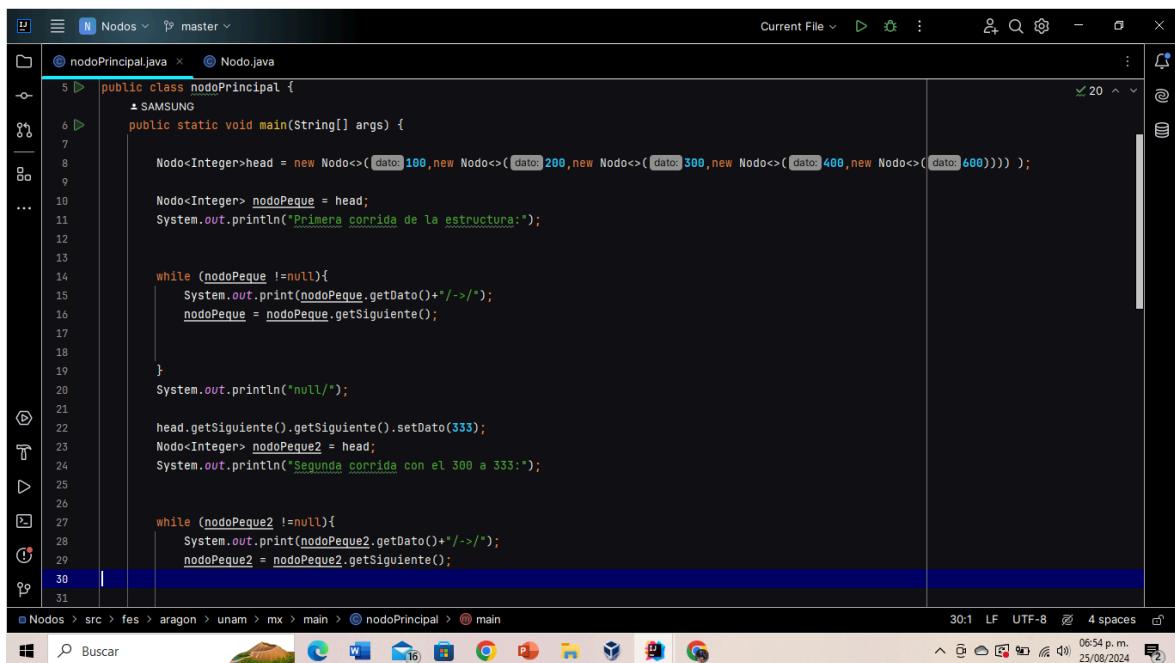


Captura del código:



```
nodoPrincipal.java
25     4 usages ± SAMSUNG
26     public T getData() { return dato; }
27
28     2 usages ± SAMSUNG
29     public void setData(T dato) {
30         this.dato = dato;
31     }
32
33
34     10 usages ± SAMSUNG
35     public Nodo<T> getSiguiente() {
36         return siguiente;
37     }
38
39     2 usages ± SAMSUNG
40     public void setSiguiente(Nodo<T> siguiente) {
41         this.siguiente = siguiente;
42     }
43
44     ± SAMSUNG
45     @Override
46     public String toString() {
47         return "Nodo{" +
48             "dato=" + dato +
49             "siguiente=" + siguiente +
50         '}';
51     }
52
53 }
```

```
Nodo.java
54 }
```



```
nodoPrincipal.java
5 > public class nodoPrincipal {
6     ± SAMSUNG
7     public static void main(String[] args) {
8
9         Nodo<Integer> head = new Nodo<>(dato:100,new Nodo<>(dato:200,new Nodo<>(dato:300,new Nodo<>(dato:400,new Nodo<>(dato:600)))) );
10
11         Nodo<Integer> _nodoPeque = head;
12         System.out.println("Primera corrida de la estructura:");
13
14         while (_nodoPeque !=null){
15             System.out.print(_nodoPeque.getData() +"/->");
16             _nodoPeque = _nodoPeque.getSiguiente();
17
18         }
19         System.out.println("null/*");
20
21         head.getSiguiente().setdato(333);
22         Nodo<Integer> _nodoPeque2 = head;
23         System.out.println("Segunda corrida con el 300 a 333:");
24
25
26
27         while (_nodoPeque2 !=null){
28             System.out.print(_nodoPeque2.getData() +"/->");
29             _nodoPeque2 = _nodoPeque2.getSiguiente();
30
31     }
```

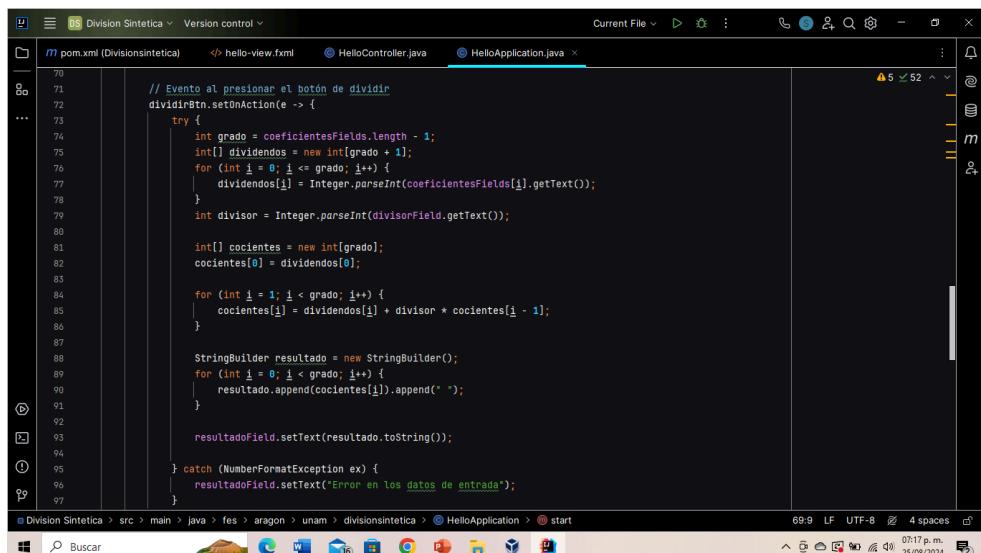
```
43
44     while (nodoPeque3 !=null){
45         System.out.print(nodoPeque3.getData()+"->"); 
46         nodoPeque3 = nodoPeque3.getSiguiente();
47     }
48
49 }
50 System.out.println("null/");
51 Nodo<Integer>head2 = new Nodo<>();
52
53 head2.setDatos(50);
54 head2.setSiguiente(head);
55 Nodo<Integer> nodoPeque4 = head2;
56 System.out.println("Cuarta corrida con 50 como primer dato:");
57
58
59 while (nodoPeque4 !=null){
60     System.out.print(nodoPeque4.getData()+"->"); 
61     nodoPeque4 = nodoPeque4.getSiguiente();
62 }
63
64
65 System.out.println("null/");
66
67
68
69
70
```

```
*C:\Program Files\Java\jdk-17\bin\java.exe* "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.4\lib\idea_rt.jar=53514:0:\Program Files\JetBrains
Primera corrida de la estructura:
100/->/200/->/300/->/400/->/600/->/null/
Segunda corrida con el 300 a 333:
100/->/200/->/333/->/400/->/600/->/null/
Tercera corrida con 700 como ultimo dato:
100/->/200/->/333/->/400/->/600/->/700/->/null/
Cuarta corrida con 50 como primer dato:
50/->/100/->/200/->/333/->/400/->/600/->/700/->/null/
Process finished with exit code 0
```

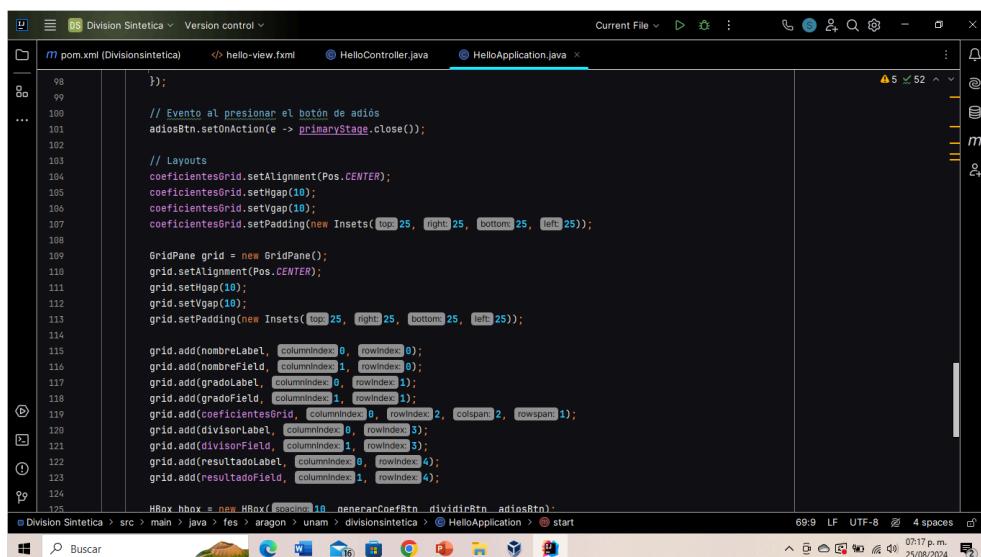
```
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.geometry.Insets;
6 import javafx.geometry.Pos;
7 import javafx.scene.Scene;
8 import javafx.scene.control.Button;
9 import javafx.scene.control.Label;
10 import javafx.scene.control.TextField;
11 import javafx.scene.layout.*;
12 import javafx.scene.paint.Color;
13 import javafx.stage.Stage;
14
15 import java.io.IOException;
16 import java.time.LocalDateTime;
17 import java.time.format.DateTimeFormatter;
18
19 public class HelloApplication extends Application {
20     private GridPane coeficientesGrid = new GridPane(); // Para los coeficientes del dividendo
21     private TextField[] coeficientesFields; // Campos para los coeficientes
22     private TextField divisionField = new TextField();
23     private TextField resultadoField = new TextField();
24
25     @Override
26     public void start(Stage primaryStage) throws IOException {
27         primaryStage.setTitle("División Sintética");
28
29         // Hora actual en la parte superior izquierda
30         LocalDateTime now = LocalDateTime.now();
31         DateTimeFormatter hourFormatter = DateTimeFormatter.ofPattern("HH:mm:ss");
32         Label horaLabel = new Label("Hora actual: " + now.format(hourFormatter));
33
34         // Fecha específica en la parte inferior derecha
35         Label fechaLabel = new Label("28 de agosto de 2024");
36
37         // Labels
38         Label nombreLabel = new Label("Nombre del usuario:");
39         Label gradoLabel = new Label("Grado de la función:");
40         Label divisorLabel = new Label("Divisor:");
41         Label resultadoLabel = new Label("Resultado:");
42
43         // TextFields
44         TextField nombreField = new TextField();
45         TextField gradoField = new TextField();
46
47         // Botones
48         Button generarCoefBtn = new Button("Generar Coeficientes");
49         Button dividirBtn = new Button("Dividir");
50         Button adiosBtn = new Button("Adiós");
51
52         // Evento al presionar el botón para generar los coeficientes según el grado
53         generarCoefBtn.setOnAction(e -> {
54             try {
55                 int grado = Integer.parseInt(gradoField.getText());
56                 coeficientesFields = new TextField[grado + 1];
57                 coeficientesGrid.getChildren().clear();
58
59                 for (int i = 0; i <= grado; i++) {
60                     Label coefLabel = new Label("Coeficiente x^" + (grado - i) + ":");
61                     coeficientesFields[i] = new TextField();
62                     coeficientesGrid.add(coefLabel, 0, i);
63                     coeficientesGrid.add(coeficientesFields[i], 1, i);
64                 }
65             } catch (NumberFormatException ex) {
66                 resultadoField.setText("Grado no válido");
67             }
68         });
69
70         // Evento al presionar el botón de dividir
71         dividirBtn.setOnAction(e -> {
72             try {
73                 int grado = coeficientesFields.length - 1;
74                 int[] dividendos = new int[grado + 1];
75                 for (int i = 0; i <= grado; i++) {
76                     dividendos[i] = Integer.parseInt(coeficientesFields[i].getText());
77                 }
78                 int divisor = Integer.parseInt(divisionField.getText());
79             } catch (NumberFormatException ex) {
80                 resultadoField.setText("Dividendo o divisor no válido");
81             }
82         });
83
84         // Configuración de la escena
85         Scene scene = new Scene(coeficientesGrid, 400, 300);
86         primaryStage.setScene(scene);
87         primaryStage.show();
88     }
89 }
```

```
25
26     @Override
27     public void start(Stage primaryStage) {
28         primaryStage.setTitle("División Sintética");
29
30         // Hora actual en la parte superior izquierda
31         LocalDateTime now = LocalDateTime.now();
32         DateTimeFormatter hourFormatter = DateTimeFormatter.ofPattern("HH:mm:ss");
33         Label horaLabel = new Label("Hora actual: " + now.format(hourFormatter));
34
35         // Fecha específica en la parte inferior derecha
36         Label fechaLabel = new Label("28 de agosto de 2024");
37
38         // Labels
39         Label nombreLabel = new Label("Nombre del usuario:");
40         Label gradoLabel = new Label("Grado de la función:");
41         Label divisorLabel = new Label("Divisor:");
42         Label resultadoLabel = new Label("Resultado:");
43
44         // TextFields
45         TextField nombreField = new TextField();
46         TextField gradoField = new TextField();
47
48         // Botones
49         Button generarCoefBtn = new Button("Generar Coeficientes");
50         Button dividirBtn = new Button("Dividir");
51         Button adiosBtn = new Button("Adiós");
52
53         // Evento al presionar el botón para generar los coeficientes según el grado
54         generarCoefBtn.setOnAction(e -> {
55             try {
56                 int grado = Integer.parseInt(gradoField.getText());
57                 coeficientesFields = new TextField[grado + 1];
58                 coeficientesGrid.getChildren().clear();
59
60                 for (int i = 0; i <= grado; i++) {
61                     Label coefLabel = new Label("Coeficiente x^" + (grado - i) + ":");
62                     coeficientesFields[i] = new TextField();
63                     coeficientesGrid.add(coefLabel, 0, i);
64                     coeficientesGrid.add(coeficientesFields[i], 1, i);
65                 }
66             } catch (NumberFormatException ex) {
67                 resultadoField.setText("Grado no válido");
68             }
69         });
70
71         // Evento al presionar el botón de dividir
72         dividirBtn.setOnAction(e -> {
73             try {
74                 int grado = coeficientesFields.length - 1;
75                 int[] dividendos = new int[grado + 1];
76                 for (int i = 0; i <= grado; i++) {
77                     dividendos[i] = Integer.parseInt(coeficientesFields[i].getText());
78                 }
79                 int divisor = Integer.parseInt(resultadoField.getText());
80             } catch (NumberFormatException ex) {
81                 resultadoField.setText("Dividendo o divisor no válido");
82             }
83         });
84
85         // Configuración de la escena
86         Scene scene = new Scene(coeficientesGrid, 400, 300);
87         primaryStage.setScene(scene);
88         primaryStage.show();
89     }
90 }
```

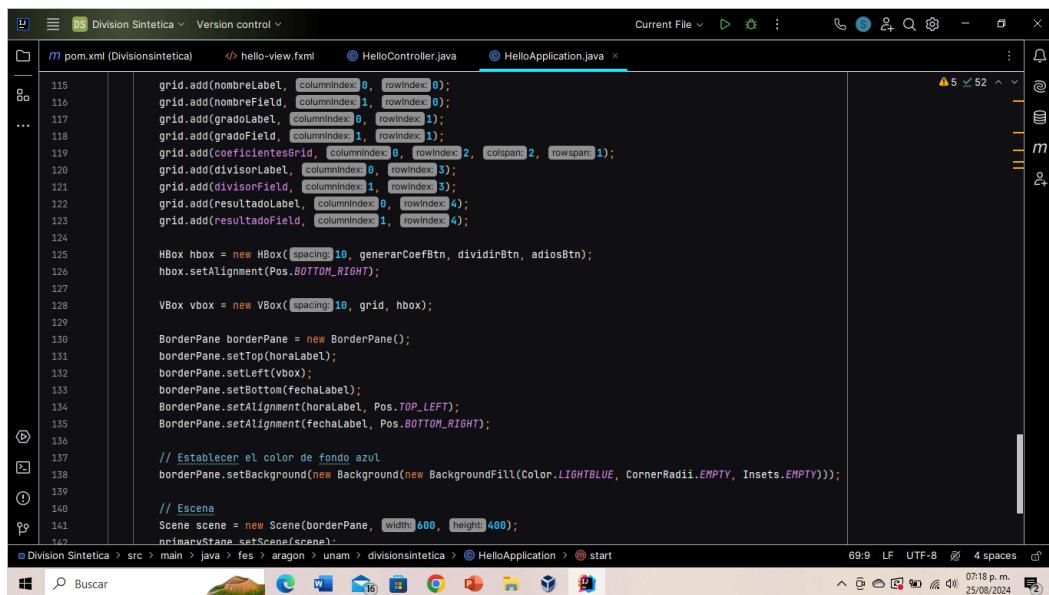
```
52         // Evento al presionar el botón para generar los coeficientes según el grado
53         generarCoefBtn.setOnAction(e -> {
54             try {
55                 int grado = Integer.parseInt(gradoField.getText());
56                 coeficientesFields = new TextField[grado + 1];
57                 coeficientesGrid.getChildren().clear();
58
59                 for (int i = 0; i <= grado; i++) {
60                     Label coefLabel = new Label("Coeficiente x^" + (grado - i) + ":");
61                     coeficientesFields[i] = new TextField();
62                     coeficientesGrid.add(coefLabel, 0, i);
63                     coeficientesGrid.add(coeficientesFields[i], 1, i);
64                 }
65             } catch (NumberFormatException ex) {
66                 resultadoField.setText("Grado no válido");
67             }
68         });
69
70         // Evento al presionar el botón de dividir
71         dividirBtn.setOnAction(e -> {
72             try {
73                 int grado = coeficientesFields.length - 1;
74                 int[] dividendos = new int[grado + 1];
75                 for (int i = 0; i <= grado; i++) {
76                     dividendos[i] = Integer.parseInt(coeficientesFields[i].getText());
77                 }
78                 int divisor = Integer.parseInt(resultadoField.getText());
79             } catch (NumberFormatException ex) {
80                 resultadoField.setText("Dividendo o divisor no válido");
81             }
82         });
83
84         // Configuración de la escena
85         Scene scene = new Scene(coeficientesGrid, 400, 300);
86         primaryStage.setScene(scene);
87         primaryStage.show();
88     }
89 }
```



```
71     // Evento al presionar el botón de dividir
72     dividirBtn.setOnAction(e -> {
73         try {
74             int grado = coeficientesFields.length - 1;
75             int[] dividendos = new int[grado + 1];
76             for (int i = 0; i <= grado; i++) {
77                 dividendos[i] = Integer.parseInt(coeficientesFields[i].getText());
78             }
79             int divisor = Integer.parseInt(divisorField.getText());
80
81             int[] cocientes = new int[grado];
82             cocientes[0] = dividendos[0];
83
84             for (int i = 1; i < grado; i++) {
85                 cocientes[i] = dividendos[i] + divisor * cocientes[i - 1];
86             }
87
88             StringBuilder resultado = new StringBuilder();
89             for (int i = 0; i < grado; i++) {
90                 resultado.append(cocientes[i]).append(" ");
91             }
92
93             resultadoField.setText(resultado.toString());
94
95         } catch (NumberFormatException ex) {
96             resultadoField.setText("Error en los datos de entrada");
97         }
98     });
99 }
```



```
98     });
99
100    // Evento al presionar el botón de adiós
101    adiosBtn.setOnAction(e -> primaryStage.close());
102
103    // Layouts
104    coeficientesGrid.setAlignment(Pos.CENTER);
105    coeficientesGrid.setHgap(10);
106    coeficientesGrid.setVgap(10);
107    coeficientesGrid.setPadding(new Insets(25, 25, 25, 25));
108
109    GridPane grid = new GridPane();
110    grid.setAlignment(Pos.CENTER);
111    grid.setHgap(10);
112    grid.setVgap(10);
113    grid.setPadding(new Insets(25, 25, 25, 25));
114
115    grid.add(nombreLabel, columnIndex: 0, rowIndex: 0);
116    grid.add(nombreField, columnIndex: 1, rowIndex: 0);
117    grid.add(gradoLabel, columnIndex: 0, rowIndex: 1);
118    grid.add(gradoField, columnIndex: 1, rowIndex: 1);
119    grid.add(coeficientesGrid, columnIndex: 0, rowIndex: 2, colspan: 2, rowspan: 1);
120    grid.add(divisorLabel, columnIndex: 0, rowIndex: 3);
121    grid.add(divisorField, columnIndex: 1, rowIndex: 3);
122    grid.add(resultadolabel, columnIndex: 0, rowIndex: 4);
123    grid.add(resultadofield, columnIndex: 1, rowIndex: 4);
124
125    HBox hbox = new HBox(spacing: 10, generarCoefBtn, dividirBtn, adiosBtn);
126    hbox.setAlignment(Pos.BOTTOM_RIGHT);
127
128    VBox vbox = new VBox(spacing: 10, grid, hbox);
129
130    BorderPane borderPane = new BorderPane();
131    borderPane.setTop(horaLabel);
132    borderPane.setLeft(vbox);
133    borderPane.setBottom(fechaLabel);
134    borderPane.setAlignment(horaLabel, Pos.TOP_LEFT);
135    borderPane.setAlignment(fechaLabel, Pos.BOTTOM_RIGHT);
136
137    // Establecer el color de fondo azul
138    borderPane.setBackground(new Background(new BackgroundFill(Color.LIGHTBLUE, CornerRadii.EMPTY, Insets.EMPTY)));
139
140    // Escena
141    Scene scene = new Scene(borderPane, width: 600, height: 400);
142    primaryStage.setScene(scene);
143 }
```



```
115    grid.add(nombreLabel, columnIndex: 0, rowIndex: 0);
116    grid.add(nombreField, columnIndex: 1, rowIndex: 0);
117    grid.add(gradoLabel, columnIndex: 0, rowIndex: 1);
118    grid.add(gradoField, columnIndex: 1, rowIndex: 1);
119    grid.add(coeficientesGrid, columnIndex: 0, rowIndex: 2, colspan: 2, rowspan: 1);
120    grid.add(divisorLabel, columnIndex: 0, rowIndex: 3);
121    grid.add(divisorField, columnIndex: 1, rowIndex: 3);
122    grid.add(resultadolabel, columnIndex: 0, rowIndex: 4);
123    grid.add(resultadofield, columnIndex: 1, rowIndex: 4);
124
125    HBox hbox = new HBox(spacing: 10, generarCoefBtn, dividirBtn, adiosBtn);
126    hbox.setAlignment(Pos.BOTTOM_RIGHT);
127
128    VBox vbox = new VBox(spacing: 10, grid, hbox);
129
130    BorderPane borderPane = new BorderPane();
131    borderPane.setTop(horaLabel);
132    borderPane.setLeft(vbox);
133    borderPane.setBottom(fechaLabel);
134    borderPane.setAlignment(horaLabel, Pos.TOP_LEFT);
135    borderPane.setAlignment(fechaLabel, Pos.BOTTOM_RIGHT);
136
137    // Establecer el color de fondo azul
138    borderPane.setBackground(new Background(new BackgroundFill(Color.LIGHTBLUE, CornerRadii.EMPTY, Insets.EMPTY)));
139
140    // Escena
141    Scene scene = new Scene(borderPane, width: 600, height: 400);
142    primaryStage.setScene(scene);
143 }
```

```
pom.xml (DivisionSintetica)  hello-view.fxml  HelloController.java  HelloApplication.java
125     HBox hbox = new HBox();
126     hbox.setSpacing(10);
127     hbox.setAlignment(Pos.BOTTOM_RIGHT);
128
129     VBox vbox = new VBox();
130     vbox.setSpacing(10);
131     vbox.setTop(horaLabel);
132     vbox.setLeft(vbox);
133     vbox.setBottom(fechaLabel);
134     BorderPane.setAlignment(horaLabel, Pos.TOP_LEFT);
135     BorderPane.setAlignment(fechaLabel, Pos.BOTTOM_RIGHT);
136
137     // Establecer el color de fondo azul
138     borderPane.setBackground(new Background(new BackgroundFill(Color.LIGHTBLUE, CornerRadii.EMPTY, Insets.EMPTY)));
139
140     // Escena
141     Scene scene = new Scene(borderPane, 600, 400);
142     primaryStage.setScene(scene);
143     primaryStage.show();
144 }
145
146 public static void main(String[] args) {
147     launch(args);
148 }
```

```
Nodos  master
nodoPrincipal.java  Nodo.java
1 package fes.aragon.unam.mx.clase;
2
3 usages new *
4 public class Nodo<T> {
5
6     usages
7     private T dato;
8     usages
9     private Nodo<T> siguiente;
10
11     no usages new *
12     public Nodo(T dato) {
13         this.dato = dato;
14     }
15
16     no usages new *
17     public Nodo(Nodo<T> siguiente) {
18         this.siguiente = siguiente;
19     }
20
21     no usages new *
22     public Nodo(T dato, Nodo<T> siguiente) {
23
24         this.dato = dato;
25         this.siguiente = siguiente;
26     }
27 }
```

```
ListaLigadas  Version control
Main.java  smartphone.java  ListaLigada.java  Nodo.java
18pring('IntelliJIdea 2023.3.4@bin' -Dfile.encoding=UTF-8 -classpath C:\Users\SAMSUNG\IdeaProjects\listaligadas\out\production\listaligadas;iso\fes\arago
scamiento>128, Color='Rosa') --> smartphone(Marca='LG', Anio=2016, memoriaRam=2, Almacenamiento=32, Color='Blanco') --> smartphone(Marca='Huawei', Anio>
lento>32, Color='Blanco') --> smartphone(Marca='Huawei', Anio=2017, memoriaRam=4, Almacenamiento=64, Color='Amarillo') --> smartphone(Marca='Xiaomi', Ani
miento>256, Color='Rojo') --> smartphone(Marca='Huawei', Anio=2017, memoriaRam=4, Almacenamiento=64, Color='Amarillo') --> smartphone(Marca='Xiaomi', Ani
miento>256, Color='Rojo') --> smartphone(Marca='Huawei', Anio=2017, memoriaRam=4, Almacenamiento=64, Color='Amarillo') --> smartphone(Marca='Xiaomi', Ani
miento>256, Color='Rojo')
OneDrive
Screenshot saved
The screenshot was added to your OneDrive.
```

```
listaligadas ico.fes.aragon.unam.mx.main.Main
hone{Marca='Huawei', Anio=2017, memoriaRam=4, Almacenamiento=64, Color='Amarillo'} --> smartPhone{Marca='Xiaomi', Anio=2022, memoriaRam=8, Almacenamiento=256, Color='Cafe'}
tPhone{Marca='Xiaomi', Anio=2022, memoriaRam=8, Almacenamiento=256, Color='Cafe'} -->
e{Marca='Huawei', Anio=2017, memoriaRam=4, Almacenamiento=64, Color='Amarillo'} --> smartPhone{Marca='Xiaomi', Anio=2022, memoriaRam=8, Almacenamiento=256}
tPhone{Marca='Xiaomi', Anio=2022, memoriaRam=8, Almacenamiento=256, Color='Cafe'} --> smartPhone{Marca='Realme', Anio=2019, memoriaRam=4, Almacenamiento=128}
```

```
--> smartPhone{Marca='Xiaomi', Anio=2022, memoriaRam=8, Almacenamiento=256, Color='Cafe'} -->
--> ...
--> ...
--> smartPhone{Marca='Xiaomi', Anio=2022, memoriaRam=8, Almacenamiento=256, Color='Cafe'} --> smartPhone{Marca='Realme', Anio=2019, memoriaRam=4, Almacenamiento=128, Color='Rosa'} -->
```

```
, Anio=2022, memoriaRam=8, Almacenamiento=256, Color='Cafe'} -->
Anio=2022, memoriaRam=8, Almacenamiento=256, Color='Cafe'} --> smartPhone{Marca='Realme', Anio=2019, memoriaRam=4, Almacenamiento=128, Color='Rosa'} -->
Anio=2019, memoriaRam=4, Almacenamiento=128, Color='Rosa'} -->
```

Screenshot of a GitHub commit interface showing code analysis results.

Commit screen for file `smartPhone.java`.

Code snippet:

```
1 package ico.fes.aragon.unam.mx.clases;
2
3 10 usages ▲ SAMSUNG
4 public class smartPhone {
5
6     4 usages
7     private String Marca;
8
9     4 usages
10    private int Anio;
11
12    4 usages
13    private int memoriaRam;
14
15    4 usages
16    private int Almacenamiento;
17
18    4 usages
19    private String Color;
20
21    no usages ▲ SAMSUNG
22    public smartPhone(){
23
24    }
25
26    8 usages ▲ SAMSUNG
27    public smartPhone(String marca, int anio, int mem
28        Marca = marca;
29
30    }
31
32    8 usages ▲ SAMSUNG
33    public smartPhone(String marca, int anio, int memoriaRam, int almacenamiento, String color) {
34        Marca = marca;
35        Anio = anio;
36        this.memoriaRam = memoriaRam;
37        Almacenamiento = almacenamiento;
38        Color = color;
39
40    }
41
42    no usages ▲ SAMSUNG
43    public void setMarca(String marca){ Marca = marca;}
44
45    no usages ▲ SAMSUNG
46    public int getAnio(){ return Anio;}
```

A blue callout box at the bottom right indicates: **Commit contains problems**, 20 warnings, Review code analysis, Ignore.

Screenshot of a GitHub commit interface showing code analysis results.

Commit screen for file `smartPhone.java`.

Code snippet:

```
17 }
18
19 8 usages ▲ SAMSUNG
20 public smartPhone(String marca, int anio, int memoriaRam, int almacenamiento, String color) {
21     Marca = marca;
22     Anio = anio;
23     this.memoriaRam = memoriaRam;
24     Almacenamiento = almacenamiento;
25     Color = color;
26
27     no usages ▲ SAMSUNG
28     public String getMarca() {
29         return Marca;
30     }
31
32     no usages ▲ SAMSUNG
33     public void setMarca(String marca){ Marca = marca;}
34
35     no usages ▲ SAMSUNG
36     public int getAnio(){ return Anio;}
```

A blue callout box at the bottom right indicates: **Commit contains problems**, 20 warnings, Review code analysis, Ignore.

Screenshot of a GitHub commit interface showing code analysis results.

Commit screen for file `smartPhone.java`.

Code snippet:

```
35 >     public int getAnio(){ return Anio; }
36
37     no usages ▲ SAMSUNG
38     public void setAnio(int anio){ Anio = anio; }
39
40     no usages ▲ SAMSUNG
41     public int getMemoriaRam(){ return memoriaRam; }
42
43     no usages ▲ SAMSUNG
44     public void setMemoriaRam(int memoriaRam){ this.memoriaRam = memoriaRam; }
45
46     no usages ▲ SAMSUNG
47     public int getAlmacenamiento(){ return Almacenamiento; }
48
49     no usages ▲ SAMSUNG
50     public void setAlmacenamiento(int almacenamiento){ Almacenamiento = almacenamiento; }
51
52     no usages ▲ SAMSUNG
53     public String getColor() {
54         return Color;
55     }
56
57     no usages ▲ SAMSUNG
58     public void setColor(String color){ Color = color; }
```

A blue callout box at the bottom right indicates: **Commit contains problems**, 20 warnings, Review code analysis, Ignore.

Screenshot of a GitHub commit interface showing code analysis results for a Java file.

The commit message is "Tarea 4".

The code editor shows a Java class named `smartPhone`:

```
public class smartPhone {  
    no usages ▲ SAMSUNG  
    public int getAlmacenamiento() { return Almacenamiento; }  
  
    no usages ▲ SAMSUNG  
    public void setAlmacenamiento(int almacenamiento) { Almacenamiento = almacenamiento; }  
  
    no usages ▲ SAMSUNG  
    public String getColor() {  
        return Color;  
    }  
  
    no usages ▲ SAMSUNG  
    public void setColor(String color) { color = color; }  
  
    ▲ SAMSUNG  
    @Override  
    public String toString() {  
        return "smartPhone" +  
            " " + Marca + '\'' +  
            ", Anios" + Anio +  
            ", memoriaRam" + memoriaRam +  
            ", Almacenamiento" + Almacenamiento +  
            ", Color" + Color + '\'';  
    }  
}
```

A tooltip indicates: "Commit contains problems 20 warnings".

File navigation bar: `smartPhone.java`, `ListaLigada.java`, `Nodo1.java`.

Screenshot of a GitHub commit interface showing code analysis results for a Java file.

The commit message is "Tarea 4".

The code editor shows a Java class named `Main.java`:

```
package ico.fes.aragon.unam.mx.clases;  
  
3 usages  
public class ListaLigada<T> {  
    17 usages  
    private Nodo1<T> head;  
    no usages  
    private int tamano;  
  
    1 usage  
    public ListaLigada() {  
    }  
  
    5 usages  
    public void agregarAlFinal(T dato) {  
        Nodo1<T> nuevo = new Nodo1<(dato);  
        if (head == null) {  
            head = nuevo;  
        } else {  
            Nodo1<T> aux = this.head;  
            while (aux.getSiguiente() != null) {  
                aux = aux.getSiguiente();  
            }  
            aux.setSiguiente(nuevo);  
        }  
    }  
}
```

File navigation bar: `Main.java`, `smartPhone.java`, `ListaLigada.java`, `Nodo1.java`.

Screenshot of a GitHub commit interface showing code analysis results for a Java file.

The commit message is "Tarea 4".

The code editor shows a Java class named `Main.java`:

```
8 }  
9  
5 usages ▲ SAMSUNG  
public void agregarAlFinal(T dato) {  
    Nodo1<T> nuevo = new Nodo1<(dato);  
    if (head == null) {  
        head = nuevo;  
    } else {  
        Nodo1<T> aux = this.head;  
        while (aux.getSiguiente() != null) {  
            aux = aux.getSiguiente();  
        }  
        aux.setSiguiente(nuevo);  
    }  
}  
  
5 usages ▲ SAMSUNG  
public void transversal() {  
    Nodo1<T> aux = this.head;  
    while (aux != null) {  
        System.out.print(aux.getDato() + " --> ");  
        aux = aux.getSiguiente();  
    }  
    System.out.println("");  
}
```

File navigation bar: `Main.java`, `smartPhone.java`, `ListaLigada.java`, `Nodo1.java`.

Commit

Main.java smartPhone.java ListaLigada.java Nodo1.java

```
no usages ▲ SAMSUNG
public boolean estaVacia() {
    if (this.head == null) {
        System.out.println("esta vacia");
        return true;
    }
    System.out.println("no esta vacia");
    return false;
}

2 usages ▲ SAMSUNG
public void insertarAlInicio(T dato) {
    this.head = new Nodo1<T>(dato, this.head);
}

1 usage ▲ SAMSUNG
public int getTamanio() {
    Nodo1 aux = this.head;
    int contador = 0;
    while (aux != null) {
        contador++;
        aux = aux.getSiguiente();
    }
    return contador;
}
```

Select files to commit

Commit Commit and Push...

listaligadas > src > ico > fes > aragon > unam > mx > clases > ListaLigada > getTamanio

57:1 LF UTF-8 4 spaces 01:21 a.m. 30/08/2024

Commit

Main.java smartPhone.java ListaLigada.java Nodo1.java

```
1 usage ▲ SAMSUNG
public int getTamanio() {
    Nodo1 aux = this.head;
    int contador = 0;
    while (aux != null) {
        contador++;
        aux = aux.getSiguiente();
    }
    return contador;
}

no usages ▲ SAMSUNG
public void agregarDespuesDe(Nodo1<T> referencia, T valor) {
    if (referencia != null) {
        Nodo1<T> nuevo = new Nodo1<T>(valor);
        nuevo.setSiguiente(referencia.getSiguiente());
        referencia.setSiguiente(nuevo);
    }
}
```

Select files to commit

Commit Commit and Push...

listaligadas > src > ico > fes > aragon > unam > mx > clases > ListaLigada > agregarDespuesDe

79:1 LF UTF-8 4 spaces 01:21 a.m. 30/08/2024

Commit

Main.java smartPhone.java ListaLigada.java Nodo1.java

```
82
no usages ▲ SAMSUNG
public void eliminarElPrimero() {
    if (head != null) {
        head = head.getSiguiente();
    }
}

no usages ▲ SAMSUNG
public int buscar(T valor) {
    Nodo1<T> nuevo = head;
    int posicion = 0;
    while (nuevo != null) {
        if (nuevo.getDato().equals(valor)) {
            return posicion;
        }
        nuevo = nuevo.getSiguiente();
        posicion++;
    }
    return -1;
}

1 usage ▲ SAMSUNG
public void actualizar(int posicion, T valor) {
    if (posicion < 0 || posicion >= getTamanio()) {
}
```

Select files to commit

Commit Commit and Push...

listaligadas > src > ico > fes > aragon > unam > mx > clases > ListaLigada > actualizar

105:1 LF UTF-8 4 spaces 01:21 a.m. 30/08/2024

Screenshot of a GitHub commit interface showing changes in Main.java:

```
1 usage  ↳ SAMSUNG
public void actualizar(int posicion, T valor) {
    if (posicion < 0 || posicion >= getTamanio()) {
        System.out.println("Posicion invalida");
        return;
    }
    Nodo1<T> nuevo = head;
    for(int i = 0; i<posicion; i++){
        nuevo= nuevo.getSiguiente();
    }
    nuevo.setdato(valor);
}

2 usages  ↳ SAMSUNG
public void eliminar(int posicion) {
    if (head == null || posicion < 0) {
        return;
    }
    if (posicion == 0) {
        head = head.getSiguiente();
        return;
    }
    Nodo1<T> actual = head;
    for (int i = 0; i < posicion - 1 && actual != null; i++) {
        actual = actual.getSiguiente();
    }

    if (actual == null || actual.getSiguiente() == null) {
        return;
    }

    actual.setSiguiente(actual.getSiguiente().getSiguiente());
}
```

Screenshot of a GitHub commit interface showing changes in Main.java:

```
117
118

2 usages  ↳ SAMSUNG
public void eliminar(int posicion) {
    if (head == null || posicion < 0) {
        return;
    }
    if (posicion == 0) {
        head = head.getSiguiente();
        return;
    }

    Nodo1<T> actual = head;
    for (int i = 0; i < posicion - 1 && actual != null; i++) {
        actual = actual.getSiguiente();
    }

    if (actual == null || actual.getSiguiente() == null) {
        return;
    }

    actual.setSiguiente(actual.getSiguiente().getSiguiente());
}
```

Screenshot of a GitHub commit interface showing changes in Nod01.java:

```
1 package ico.fes.aragon.unam.mx.clases;
2
3 18 usages
4 public class Nod01<T> {
5     5 usages
6     private T dato;
7     4 usages
8     private Nod01<T> siguiente;
9
10    no usages
11    public Nod01() {
12    }
13
14    2 usages
15    public Nod01(T dato){this.dato = dato;}
16
17    1 usage
18    public Nod01(T dato, Nod01<T> siguiente) {
19        this.dato = dato;
20        this.siguiente = siguiente;
21    }
22
23    2 usages
24    public T getdato(){return dato;}
25
26    1 usage
27    public void setdato(T dato){this.dato = dato;}
```

Screenshot of a GitHub commit interface for a Java project named "listaligadas". The commit message is "Tarea 4". The code editor shows Main.java with the following content:

```
10 2 usages ▲ SAMSUNG
11 > public Nodo1<T dato> {this.dato = dato;}
12
13 usage ▲ SAMSUNG
14 public Nodo1<T dato, Nodo1<T> siguiente> {
15     this.dato = dato;
16     this.siguiente = siguiente;
17 }
18
19
20 > 2 usages ▲ SAMSUNG
21 public T getdato() {return dato;}
22
23 1 usage ▲ SAMSUNG
24 > public void setdato(T dato) {this.dato = dato;}
25
26 13 usages ▲ SAMSUNG
27 public Nodo1<T> getSiguiente() {return siguiente;}
28
29 4 usages ▲ SAMSUNG
30 public void setsiguiente(Nodo1<T> siguiente) {this.sigiente = siguiente;}
31
32 ▲ SAMSUNG
33 @Override
34 public String toString() {
35     return "Nodo1<" + dato +
36         ", siguiente=" + siguiente + ">";
37 }
38
39 }
```

The status bar at the bottom indicates the file is 39/2, LF, UTF-8, 4 spaces, and was last modified at 01:22 a.m. on 30/08/2024.

Screenshot of the same GitHub commit interface, showing the code editor with the cursor moved to the end of the toString() method's body. The code now looks like this:

```
15 public Nodo1<T dato, Nodo1<T> siguiente> {
16     this.dato = dato;
17     this.siguiente = siguiente;
18 }
19
20 > 2 usages ▲ SAMSUNG
21 public T getdato() {return dato;}
22
23 1 usage ▲ SAMSUNG
24 > public void setdato(T dato) {this.dato = dato;}
25
26 13 usages ▲ SAMSUNG
27 public Nodo1<T> getSiguiente() {return siguiente;}
28
29 4 usages ▲ SAMSUNG
30 public void setsiguiente(Nodo1<T> siguiente) {this.sigiente = siguiente;}
31
32 ▲ SAMSUNG
33 @Override
34 public String toString() {
35     return "Nodo1<" + dato +
36         ", siguiente=" + siguiente + ">";
37 }
38
39 }
```

The status bar at the bottom indicates the file is 44/1, LF, UTF-8, 4 spaces, and was last modified at 01:22 a.m. on 30/08/2024.