

# **Facultad de Ingeniería y Ciencias Aplicadas**

## **Ingeniería de Software**

### **ISWZ2102 - Programación III**

#### **Exposición de Aplicaciones de Árboles y Grafos**

**Cristian López**

#### **Documento de Resumen**

#### **Redes de Comunicación Dinámica: Aplicaciones de Grafos en Entornos Cambiantes**

##### **Introducción**

Las redes de comunicación dinámica representan una evolución en los sistemas de transmisión de información, caracterizadas por la variabilidad de sus componentes a lo largo del tiempo. A diferencia de las redes estáticas, las redes dinámicas cambian su topología constantemente, lo que implica que los nodos y enlaces pueden aparecer o desaparecer en diferentes momentos. Estas redes se modelan de forma efectiva utilizando grafos dinámicos o temporales, lo que permite un análisis más detallado de su comportamiento. Son fundamentales en tecnologías modernas como vehículos autónomos, redes móviles, sensores inteligentes y plataformas digitales en tiempo real.

##### **Objetivos Específicos de Aprendizaje**

- Comprender el concepto de redes de comunicación dinámica.
- Analizar cómo se aplican los grafos para modelar estas redes.
- Identificar aplicaciones reales de redes dinámicas en la actualidad.
- Resolver problemas prácticos utilizando conceptos de grafos dinámicos.

##### **Contenido Detallado con Ejemplos**

###### **1. Definición de Redes de Comunicación Dinámica:**

Una red de comunicación dinámica es aquella en la que la estructura de conexiones (nodos y enlaces) cambia con el tiempo. En este contexto, cada estado de la red puede

representarse como un grafo  $G(t)$ , donde  $t$  representa el instante de tiempo. Estos grafos permiten analizar cómo fluye la información, cómo se forman caminos entre nodos y cómo se pueden optimizar las rutas de comunicación.

## 2. Modelado mediante grafos dinámicos:

Cada nodo representa una entidad (dispositivo, usuario, sensor), mientras que las aristas representan enlaces de comunicación activos en un momento dado. La secuencia de estos grafos en el tiempo permite construir un 'grafo temporal' que facilita el análisis del comportamiento dinámico.

## 3. Ejemplo ilustrativo:

Supongamos una red vehicular donde los autos (nodos) se conectan cuando están dentro del rango de comunicación. En  $t_1$ , A está conectado con B. En  $t_2$ , B se conecta con C. En  $t_3$ , A se desconecta pero C se conecta con D. Este comportamiento se modela como una sucesión de grafos  $G(t_1)$ ,  $G(t_2)$ ,  $G(t_3)$ ...

## 4. Aplicaciones reales:

- Redes de sensores móviles (Smart Cities).
- Comunicaciones entre drones o robots.
- Vehículos autónomos (VANETs).
- Aplicaciones móviles de transporte o mensajería en tiempo real.

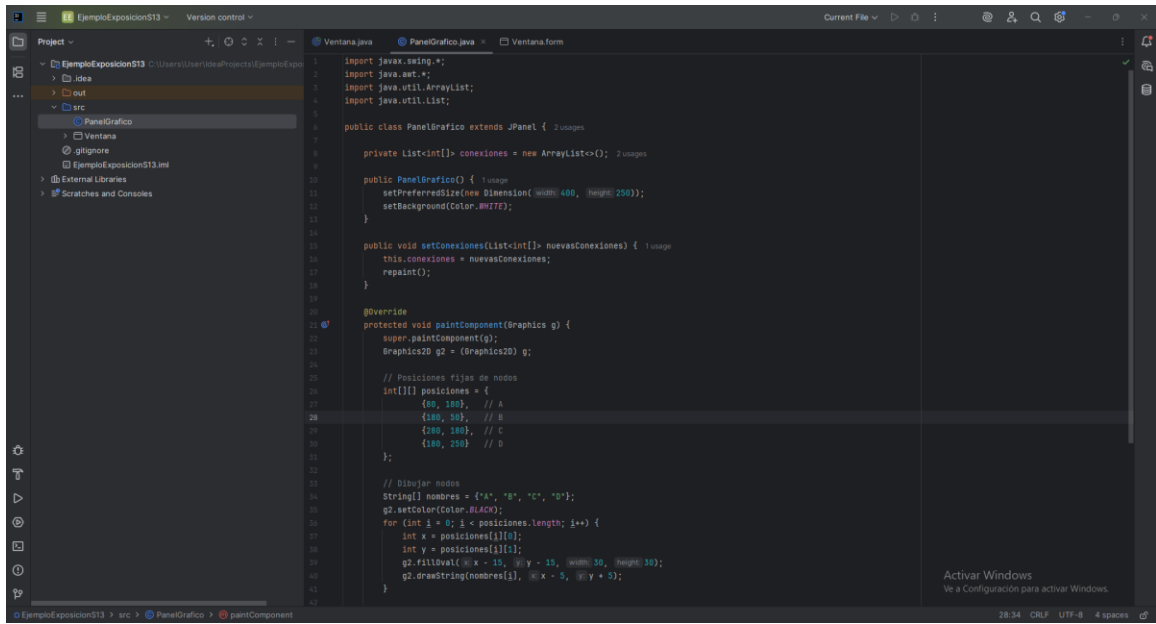
## 5. Resolución de problemas:

Problema: ¿Cuál es el camino más corto para enviar un mensaje de A a D en una red dinámica? Se analizan los posibles caminos a lo largo del tiempo, considerando que algunas conexiones no existen todo el tiempo. Aquí se utiliza el concepto de 'camino temporalmente válido'.

## Conclusión Reflexiva

El estudio de las redes de comunicación dinámica permite entender y diseñar sistemas más robustos y eficientes en entornos donde las condiciones cambian constantemente. Los grafos dinámicos son herramientas poderosas que ayudan a modelar, visualizar y analizar estos sistemas. Al aplicar estos conceptos, es posible mejorar la toma de decisiones en sistemas tecnológicos avanzados y optimizar recursos en redes móviles, inteligentes y conectadas.

Capturas del Código:



```
Ventana.java  PanelGrafico.java  Ventana.form
6  public class PanelGrafico extends JPanel { 2 usages
17      repaint();
18  }
19
20  @Override
21  protected void paintComponent(Graphics g) {
22      super.paintComponent(g);
23      Graphics2D g2 = (Graphics2D) g;
24
25      // Posiciones fijas de nodos
26      int[][] posiciones = {
27          {80, 180}, // A
28          {180, 50}, // B
29          {280, 180}, // C
30          {180, 250} // D
31      };
32
33      // Dibujar nodos
34      String[] nombres = {"A", "B", "C", "D"};
35      g2.setColor(Color.BLACK);
36      for (int i = 0; i < posiciones.length; i++) {
37          int x = posiciones[i][0];
38          int y = posiciones[i][1];
39          g2.fillOval(x - 15, y - 15, width: 30, height: 30);
40          g2.drawString(nombres[i], x - 5, y + 5);
41      }
42
43      // Dibujar conexiones
44      g2.setStroke(new BasicStroke(width: 2));
45      g2.setColor(Color.BLUE);
46      for (int[] c : conexiones) {
47          int x1 = posiciones[c[0]][0];
48          int y1 = posiciones[c[0]][1];
49          int x2 = posiciones[c[1]][0];
50          int y2 = posiciones[c[1]][1];
51          g2.drawLine(x1, y1, x2, y2);
52      }
53  }
54  }
55
```

```
Ventana.java x PanelGrafico.java Ventana.form
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.util.ArrayList;
6 import java.util.List;
7
8 public class Ventana extends JFrame {
9
10     private PanelGrafico panelGrafico; 3 usages
11
12     private JPanel contentPane; 3 usages
13     private JLabel labelEstado; 5 usages
14     private JLabel labelConexiones; 5 usages
15     private JButton btnSiguiente; 2 usages
16
17     private int estado = 0; 3 usages
18
19     public Ventana() { 1 usage
20         setTitle("Red de Comunicación Dinámica");
21         setContentPane(contentPane); // Asegúrate que el nombre esté correcto
22         setSize(400, 200);
23         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24         setLocationRelativeTo(null);
25         setVisible(true);
26         panelGrafico = new PanelGrafico();
27         getContentPane().add(panelGrafico, BorderLayout.SOUTH);
28
29         actualizarEstado();
30
31         btnSiguiente.addActionListener(new ActionListener() {
32             @Override
33             public void actionPerformed(ActionEvent e) {
34                 estado = (estado + 1) % 4;
35                 actualizarEstado();
36             }
37         });
38     }
39
40     private void actualizarEstado() { 2 usages
41         List<int[]> conexiones = new ArrayList<>();
42     }
```

```

39
40     private void actualizarEstado() { 2 usages
41         List<int[]> conexiones = new ArrayList<>();
42
43         switch (estado) {
44             case 0:
45                 labelEstado.setText("Estado: 61");
46                 labelConexiones.setText("Conexión: A - B");
47                 conexiones.add(new int[]{0, 1});
48                 break;
49             case 1:
50                 labelEstado.setText("Estado: 62");
51                 labelConexiones.setText("Conexión: B - C");
52                 conexiones.add(new int[]{1, 2});
53                 break;
54             case 2:
55                 labelEstado.setText("Estado: 63");
56                 labelConexiones.setText("Conexión: C - D");
57                 conexiones.add(new int[]{2, 3});
58                 break;
59             case 3:
60                 labelEstado.setText("Estado: 64");
61                 labelConexiones.setText("Conexión: A - D");
62                 conexiones.add(new int[]{0, 3});
63                 break;
64         }
65
66         panelGrafico.setConexiones(conexiones);
67     }
68
69     public static void main(String[] args) {
70         JFrame frame = new JFrame( title: "Ventana");
71         frame.setContentPane(new Ventana().contentPane);
72         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
73         frame.pack();
74         frame.setVisible(true);
75     }
76 }
77
78

```

Ventana.javaPanelGrafico.javaVentana.form x

Component Tree

Form (Ventana)

contentPane : JPanel

labelEstado : JLabel

labelConexiones : JLabel

btnSiguiente : JButton

Property	Value
field name	contentPane
Custom Create	<input type="checkbox"/>
Layout Manager	BorderLayout
border	None
Horizontal Gap	0
Vertical Gap	0
Client Properties	
background	[43,45,48]
enabled	<input checked="" type="checkbox"/>
font	<default>
foreground	[223,225,229]
toolTipText	

☐ Show expert properties

Estado:

Conexiones:

Siguiente Estado

Activar Windows  
Ve a Configuración pa

Ventana

Estado: G1

Siguiente Estado

Conexión: A - B

TPC

Algunos roles

Buscar

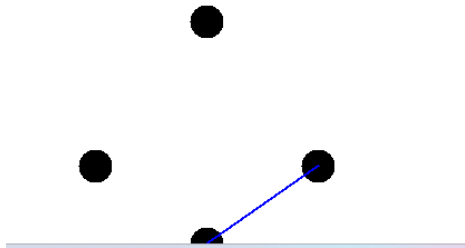
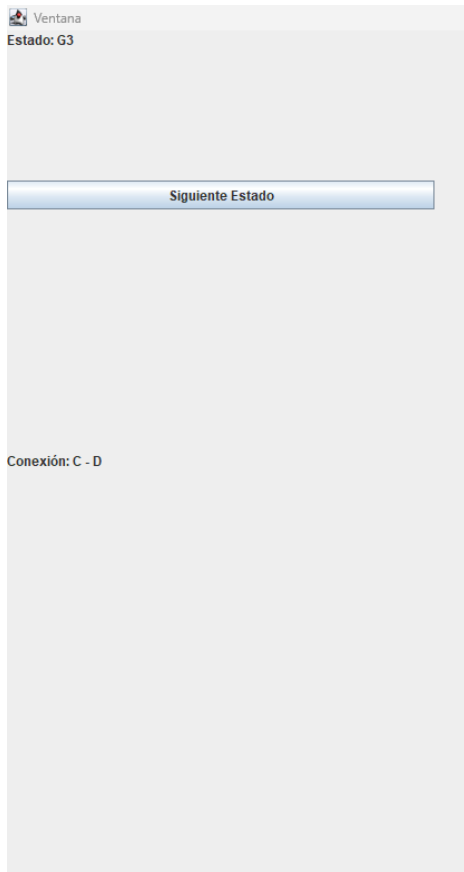
ESP

LAA

11:01

5/7/2023





### Referencias Académicas

- Holme, P., & Saramäki, J. (2012). Temporal networks. *Physics reports*, 519(3), 97-125.
- Nicosia, V., & Latora, V. (2015). Measuring and modeling correlations in multiplex networks. *Physical Review E*, 92(3), 032805.
- Casteigts, A., Flocchini, P., Quattrociocchi, W., & Santoro, N. (2012). Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5), 387-408.