
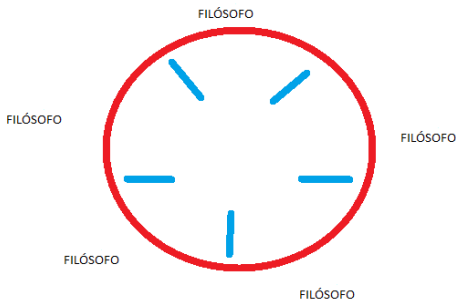

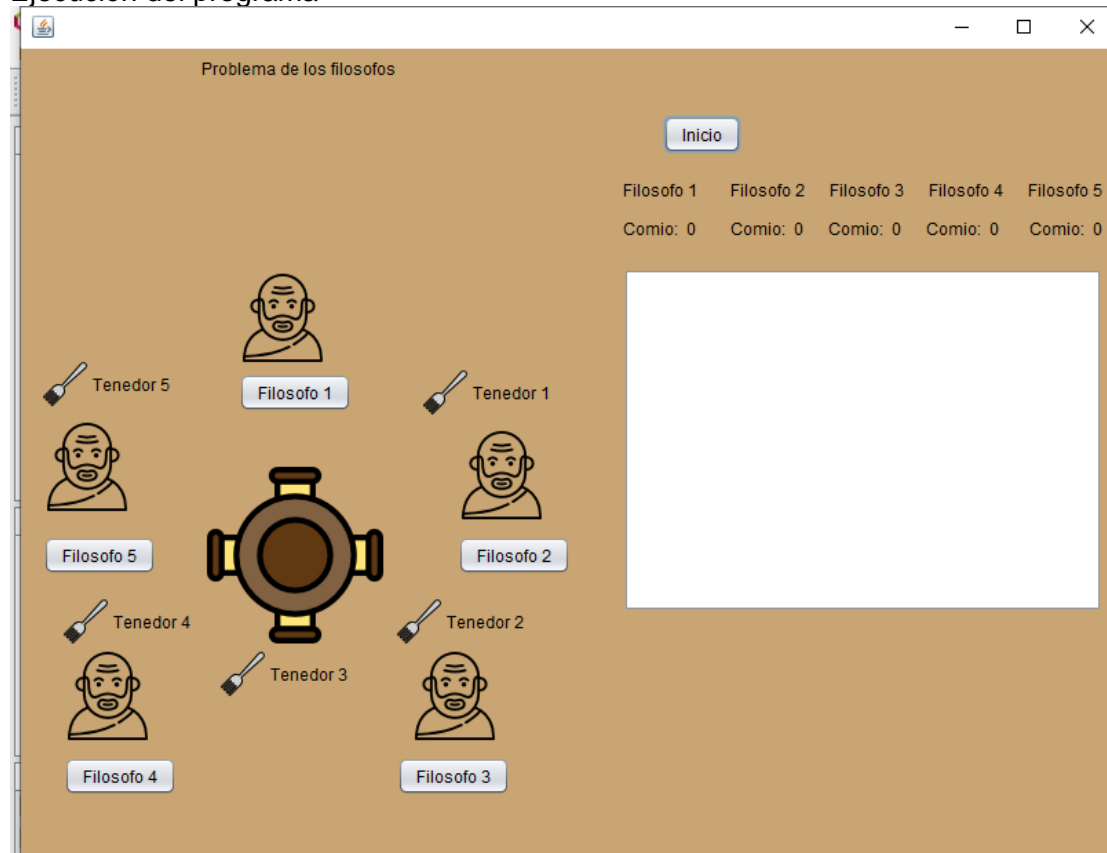
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<b>FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES</b>	
<b>CARRERA:</b> COMPUTACIÓN		<b>ASIGNATURA:</b> Programación Aplicada	
<b>NRO. PRÁCTICA:</b>	1	<b>TÍTULO PRÁCTICA:</b> Hilos en Java	
<b>OBJETIVO:</b> Identificar los cambios importantes de Java Diseñar e Implementar las nuevas técnicas de programación concurrente Entender cada una de las características de Thread en Java.			
<b>INSTRUCCIONES</b> (Detallar las instrucciones que se dará al estudiante):	1. Revisar los conceptos fundamentales de Thread en Java		
	2. Establecer como implementar Thread en Java		
	3. Implementar y diseñar los nuevos componentes de concurrencia		
	4. Realizar el informe respectivo según los datos solicitados.		
<b>ACTIVIDADES POR DESARROLLAR</b> (Anotar las actividades que deberá seguir el estudiante para el cumplimiento de la práctica)			
1. Revisar la teoría y conceptos de Thread en Java			
2. Diseñar e implementar las características de Java para generar una simulación 2D del siguiente enunciado:			
<b>Problema del Filósofo:</b>  En una mesa hay procesos que simulan el comportamiento de unos filósofos que intentan comer de un plato. Cada filósofo tiene un cubierto a su izquierda y uno a su derecha y para poder comer tiene que conseguir los dos. Si lo consigue, mostrará un mensaje en pantalla que indique «Filósofo 2 (numero) comiendo». Después de comer, soltará los cubiertos y esperará al azar un tiempo entre 1000 y 5000 milisegundos, indicando por pantalla «El filósofo 2 está pensando».			
En general todos los objetos de la clase Filósofo está en un bucle infinito dedicándose a comer y a pensar.			
Simular este problema en un programa Java que muestre el progreso de todos sin caer en problemas de sincronización a través de un método grafico.			
			

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

### 3. Programa

#### Ejecución del programa



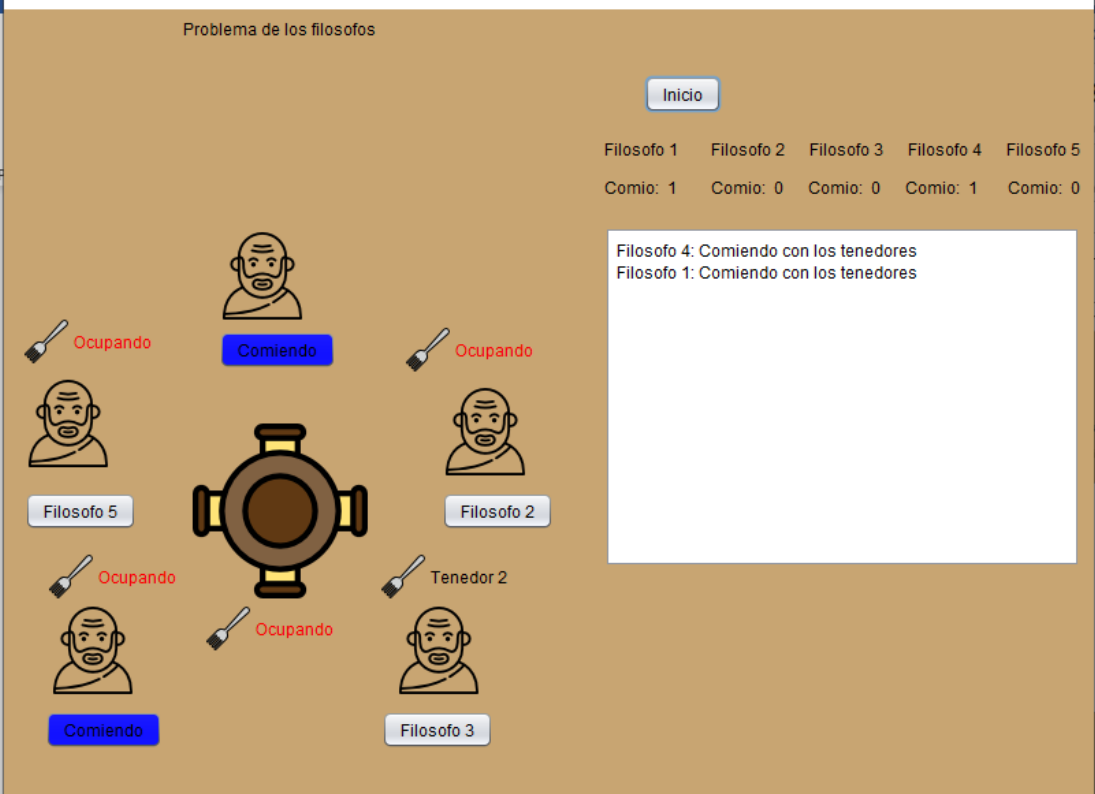
Inicio del programa

Problema de los filósofos

Inicio

Filosofo 1	Filosofo 2	Filosofo 3	Filosofo 4	Filosofo 5
Comio: 1	Comio: 0	Comio: 0	Comio: 1	Comio: 0

Filosofo 4: Comiendo con los tenedores  
Filosofo 1: Comiendo con los tenedores



The diagram illustrates the Dining Philosophers problem. Five philosophers are seated around a circular table. Each philosopher is represented by a cartoon face and a label. The status of each philosopher is indicated by a label: 'Ocupando' (Occupied) or 'Comiendo' (Eating). The table has five forks, represented by icons. Fork 2 is labeled 'Tenedor 2'. The simulation shows Filosofo 4 and Filosofo 1 eating, while the others are occupied.

Problema de los filósofos

Inicio

Filosofo 1	Filosofo 2	Filosofo 3	Filosofo 4	Filosofo 5
Comio: 2	Comio: 2	Comio: 2	Comio: 2	Comio: 2

Filosofo 2: Comiendo con los tenedores  
 Filosofo 5: Comiendo con los tenedores  
 Filosofo 5: Dejo de usar los tenedores  
 Filosofo 2: Dejo de usar los tenedores  
 Filosofo 1: Comiendo con los tenedores  
 Filosofo 3: Comiendo con los tenedores  
 Filosofo 1: Dejo de usar los tenedores  
 Filosofo 3: Dejo de usar los tenedores  
 Filosofo 4: Comiendo con los tenedores  
 Filosofo 2: Comiendo con los tenedores  
 Filosofo 4: Dejo de usar los tenedores  
 Filosofo 2: Dejo de usar los tenedores  
 Filosofo 5: Comiendo con los tenedores  
 Filosofo 3: Comiendo con los tenedores

CONCLUSIONES:

Problema de los filósofos

Inicio

Filosofo 1	Filosofo 2	Filosofo 3	Filosofo 4	Filosofo 5
Comio: 4	Comio: 4	Comio: 3	Comio: 4	Comio: 3


Filosofo 5: Comiendo con los tenedores  
Filosofo 3: Comiendo con los tenedores  
Filosofo 5: Dejo de usar los tenedores  
Filosofo 1: Comiendo con los tenedores  
Filosofo 3: Dejo de usar los tenedores  
Filosofo 4: Comiendo con los tenedores  
Filosofo 1: Dejo de usar los tenedores  
Filosofo 2: Comiendo con los tenedores  
Filosofo 4: Dejo de usar los tenedores  
Filosofo 5: Comiendo con los tenedores  
Filosofo 2: Dejo de usar los tenedores  
Filosofo 3: Comiendo con los tenedores  
Filosofo 5: Dejo de usar los tenedores  
Filosofo 1: Comiendo con los tenedores  
Filosofo 3: Dejo de usar los tenedores

Termina de ejecutarse



Lo que presenta el JTextArea

Filosofo 4: Comiendo con los tenedores  
 Filosofo 1: Comiendo con los tenedores  
 Filosofo 1: Dejo de usar los tenedores  
 Filosofo 4: Dejo de usar los tenedores  
 Filosofo 2: Comiendo con los tenedores  
 Filosofo 5: Comiendo con los tenedores  
 Filosofo 5: Dejo de usar los tenedores  
 Filosofo 2: Dejo de usar los tenedores  
 Filosofo 1: Comiendo con los tenedores  
 Filosofo 3: Comiendo con los tenedores  
 Filosofo 1: Dejo de usar los tenedores  
 Filosofo 3: Dejo de usar los tenedores  
 Filosofo 4: Comiendo con los tenedores  
 Filosofo 2: Comiendo con los tenedores  
 Filosofo 4: Dejo de usar los tenedores  
 Filosofo 2: Dejo de usar los tenedores  
 Filosofo 5: Comiendo con los tenedores  
 Filosofo 3: Comiendo con los tenedores  
 Filosofo 5: Dejo de usar los tenedores  
 Filosofo 1: Comiendo con los tenedores  
 Filosofo 3: Dejo de usar los tenedores  
 Filosofo 4: Comiendo con los tenedores  
 Filosofo 1: Dejo de usar los tenedores

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Filosofo 2: Comiendo con los tenedores  
 Filosofo 4: Dejo de usar los tenedores  
 Filosofo 5: Comiendo con los tenedores  
 Filosofo 2: Dejo de usar los tenedores  
 Filosofo 3: Comiendo con los tenedores  
 Filosofo 5: Dejo de usar los tenedores  
 Filosofo 1: Comiendo con los tenedores  
 Filosofo 3: Dejo de usar los tenedores  
 Filosofo 4: Comiendo con los tenedores  
 Filosofo 1: Dejo de usar los tenedores  
 Filosofo 2: Comiendo con los tenedores  
 Filosofo 4: Dejo de usar los tenedores  
 Filosofo 5: Comiendo con los tenedores  
 Filosofo 2: Dejo de usar los tenedores  
 Filosofo 3: Comiendo con los tenedores  
 Filosofo 5: Dejo de usar los tenedores  
 Filosofo 3: Dejo de usar los tenedores  
 Hilos que se uso  
 public class Filosofo implements Runnable {

```

    private int id;
    private int resultado;
    private Thread t;
    private JButton filosofo;
    private JLabel derecha;
    private JLabel izquierda;
    private JLabel resultado1;
    private String process;
    private JTextArea area;
  
```

```

    public Filosofo(int id,JLabel derecha, JLabel izquierda,JButton filosofo, JLabel resultado1, JTextArea
area) {
        this.id = id;
        this.filosofo = filosofo;
        this.derecha = derecha;
        this.izquierda = izquierda;
        this.resultado1 = resultado1;
        this.area = area;
        t = new Thread(this);
        t.start();
    }

    public JLabel getDerecha() {
        return derecha;
    }

    public JLabel getIzquierda() {
        return izquierda;
    }
  
```

```
}

@Override
public void run() {
    for (int i = 0; i < 4; i++) { // se controla el numero de veces que van a comer
        synchronized (this.izquierda) {
            synchronized (this.derecha) {
                comer();
            }
        }
        pensar();
    }
}

private void comer() {
    derecha.setText("Ocupando");
    derecha.setForeground(Color.red);

    izquierda.setText("Ocupando");
    izquierda.setForeground(Color.red);
    filosofo.setText("Comiendo");
    filosofo.setBackground(Color.BLUE);

    resultado = Integer.parseInt(resultado1.getText());
    resultado += 1;
    resultado1.setText(String.valueOf(resultado));
    process = "Filosofo " + (id + 1) + ": " + "Comiendo con los tenedores\n";
    area.append(process);
    try {
        Thread.sleep(5000);
    } catch (InterruptedException e) {
    }
    derecha.setText("Libre");
    derecha.setForeground(Color.black);


    izquierda.setText("Libre");
    izquierda.setForeground(Color.black);

    filosofo.setText("Pensando");
    filosofo.setBackground(Color.DARK_GRAY);

    process = "Filosofo " + (id + 1) + ": " + "Dejo de usar los tenedores\n";
    area.append(process);
}

private void pensar() {
```



	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

```

derecha.setText("Libre");
derecha.setForeground(Color.black);

izquierda.setText("Libre");
izquierda.setForeground(Color.black);

filosofo.setText("Pensando");
filosofo.setBackground(Color.DARK_GRAY);

try {
    Thread.sleep(5000);
} catch (InterruptedException ex) {
}
}
}

```

**RESULTADO(S) OBTENIDO(S):**

Realizar procesos de Hilos en Java.  
Entender las aplicaciones de codificación de las nuevas características de concurrencia.  
Entender las funcionalidades de sincronización y manejo de grupo de Thread dentro de Java.

**CONCLUSIONES:**

Aprenden a trabajar en grupo dentro de plazos de tiempo establecidos, manejando el lenguaje de programación de Java.

**RECOMENDACIONES:**

Realizar el trabajo dentro del tiempo establecido.

**Nombre de estudiante:** Andres Alvarado

**Firma de estudiante:**

