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

Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

Universidad Politécnica Salesiana

Vicerrectorado Docente

Código del Formato:	GUIA-PRL-001
Versión:	VF1.0
Elaborado por:	Directores de Área del Conocimiento Integrantes Consejo Académico
Fecha de elaboración:	2016/04/01
Revisado por:	Consejo Académico
Fecha de revisión:	2016/04/06
Aprobado por:	Lauro Fernando Pesántez Avilés Vicerrector Docente
Fecha de aprobación:	2016/14/06
Nivel de confidencialidad:	Interno

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

Descripción General

Propósito


El propósito del presente documento es definir un estándar para elaborar documentación de guías de práctica de laboratorio, talleres o centros de simulación de las Carreras de la Universidad Politécnica Salesiana, con la finalidad de lograr una homogenización en la presentación de la información por parte del personal académico y técnico docente.


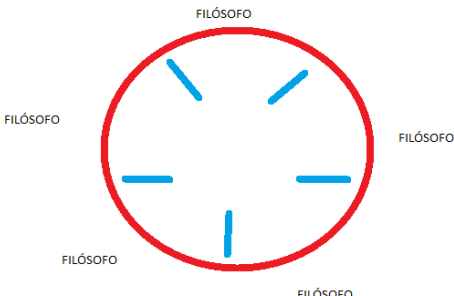
Alcance


El presente estándar será aplicado a toda la documentación referente a informes de prácticas de laboratorio, talleres o centros de simulación de las Carreras de la Universidad Politécnica Salesiana.

Formatos

- Formato de Guía de Práctica de Laboratorio / Talleres / Centros de Simulación – para Docentes
- Formato de Informe de Práctica de Laboratorio / Talleres / Centros de Simulación – para Estudiantes

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

		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES	
CARRERA: COMPUTACIÓN		ASIGNATURA: Programación Aplicada	
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Hilos en Java	
OBJETIVO: 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.			
INSTRUCCIONES (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.		
ACTIVIDADES POR DESARROLLAR (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:			
Problema del Filósofo: 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. Probar y modificar el método para que nos permita cambiar el numero de filósofos.
4. Realizar práctica codificando con las nuevas características de Java, patrones de diseño, Thread, etc.
5. **Fecha de Entrega:** 11 Enero del 2021 23:55

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.

Docente / Técnico Docente: _____

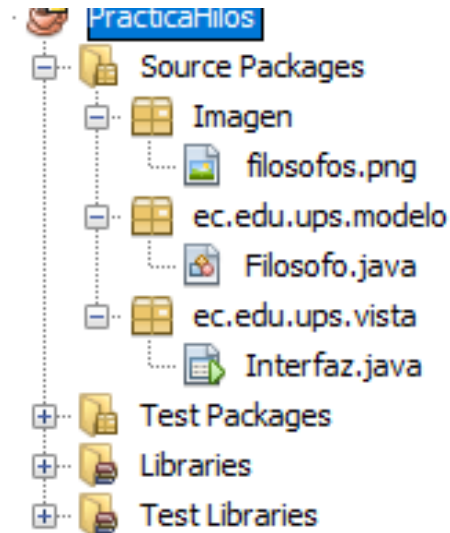
Firma: _____

		FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES	
CARRERA: COMPUTACIÓN		ASIGNATURA: Programación Aplicada	
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Hilos en Java	
OBJETIVO ALCANZADO: Diseñar e Implementar las nuevas técnicas de programación concurrente Entender cada una de las características de Thread en Java.			
ACTIVIDADES DESARROLLADAS			
<p>1. Diseñar e implementar las características de Java para generar una simulación 2D del siguiente enunciado:</p> <p>Problema del Filósofo:</p> <p>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».</p> <p>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».</p> <p>En general todos los objetos de la clase Filósofo está en un bucle infinito dedicándose a comer y a pensar.</p>			

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

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.


Paquetes:



Clase Filosofo

```
package ec.edu.ups.modelo;
import java.awt.Color;

import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JTextArea;
/**
 *
 * @author Anahi
 */
public class Filosofo implements Runnable{
int id,res;
Thread t;
JButton filosofo;
JLabel derecho;
JLabel izquierdo;
JLabel resultado;
String proceso;
JTextArea textArea;
public Filosofo(int id,JLabel izquierdo, JLabel derecho,JButton filosofo,JLabel
resultado, JTextArea textArea){
    this.id = id;
    this.derecho = derecho;
    this.izquierdo = izquierdo;
    this.filosofo=filosofo;
    this.resultado=resultado;
    this.textArea=textArea;
    t = new Thread(this);
```

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

```

t.start();

}
public void run() {
    for(int i =0;i<4;i++){
        synchronized(this.izquierdo){
            synchronized(this.derecho){
                comer();
            }
        }
        pensar();
    }
}

void comer () {
    derecho.setText("Ocupado");
    derecho.setForeground(Color.red);

    izquierdo.setText("Ocupado");
    izquierdo.setForeground(Color.red);

    filosofo.setText("Comiendo");
    filosofo.setBackground(Color.GREEN);

    res=Integer.parseInt(resultado.getText());
    res+=1;
    resultado.setText(String.valueOf(res));
    proceso= "Fil.= "+(id+1)+ " Comiendo usa sus tenedores\n";
    textArea.append(proceso);
    try{
        Thread.sleep(5000);
    }catch(InterruptedException e){
    }
    derecho.setText("Libre");
    derecho.setForeground(Color.black);

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


    filosofo.setText("Pensando");
    filosofo.setBackground(Color.DARK_GRAY);
    proceso="Fil.= "+(id+1)+ " Deja de comer y queda pensando, libera sus
tenedores\n";
    textArea.append(proceso);
}

void pensar(){
    derecho.setText("Libre");
    derecho.setForeground(Color.black);

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

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

```

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

```

try{

    Thread.sleep(5000);
}catch (InterruptedException e) {
}
}
}

```

Interfaz

```
package ec.edu.ups.vista;
```

```

import ec.edu.ups.modelo.Filosofo;
import java.awt.Graphics;
import java.awt.Image;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JPanel;
import ec.edu.ups.modelo.Filosofo;

```

```

/**
 *
 * @author Anahi
 */

```


```

public class Interfaz extends javax.swing.JFrame {

    Filosofo f;

    /**
     * Creates new form Interfaz
     */
    public Interfaz() {

```

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

```

initComponents();
inicio();


}

public void inicio(){
    labCom1.setText("0");
    labCom2.setText("0");
    labCom3.setText("0");
    labCom4.setText("0");
    labCom5.setText("0");
}

/**
 * This method is called from within the constructor to initialize the form.
 * regenerated by the Form Editor.
 * WARNING: Do NOT modify this code. The
 * content of this method is always r.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jPanel1 = new FondoPanel();
    butFil1 = new javax.swing.JButton();
    labTen1 = new javax.swing.JLabel();
    labTen5 = new javax.swing.JLabel();
    labTen3 = new javax.swing.JLabel();
    butFil2 = new javax.swing.JButton();
    labTen2 = new javax.swing.JLabel();
    labTen4 = new javax.swing.JLabel();
    butFil3 = new javax.swing.JButton();

```


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

```


butFil4 = new javax.swing.JButton();
butFil5 = new javax.swing.JButton();
jPanel2 = new javax.swing.JPanel();
jPanel3 = new javax.swing.JPanel();
jScrollPane1 = new javax.swing.JScrollPane();
tAreaProceso = new javax.swing.JTextArea();
jPanel4 = new javax.swing.JPanel();
labCom4 = new javax.swing.JLabel();
labCom5 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
jLabel7 = new javax.swing.JLabel();
jLabel8 = new javax.swing.JLabel();
jLabel9 = new javax.swing.JLabel();
labCom1 = new javax.swing.JLabel();
labCom2 = new javax.swing.JLabel();
labCom3 = new javax.swing.JLabel();
jButton1 = new javax.swing.JButton();
jLabel2 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setBackground(new java.awt.Color(0, 102, 102));

jPanel1.setBackground(new java.awt.Color(0, 153, 153));
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

butFil1.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
butFil1.setText("Filosofo1");
butFil1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {

```

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

```

        butFil1ActionPerformed(evt);
    }
});
jPanel1.add(butFil1, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 0, -1, -1));

labTen1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
labTen1.setText("Tenedor1");
jPanel1.add(labTen1, new org.netbeans.lib.awtextra.AbsoluteConstraints(310, 60, 80, 20));

labTen5.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
labTen5.setText("Tenedor5");
jPanel1.add(labTen5, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 70, -1, -1));

labTen3.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
labTen3.setText("Tenedor3");
jPanel1.add(labTen3, new org.netbeans.lib.awtextra.AbsoluteConstraints(190, 320, -1, -1));


butFil2.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
butFil2.setText("Filosofo2");
jPanel1.add(butFil2, new org.netbeans.lib.awtextra.AbsoluteConstraints(370, 160, -1, -1));

labTen2.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
labTen2.setText("Tenedor2");
jPanel1.add(labTen2, new org.netbeans.lib.awtextra.AbsoluteConstraints(360, 220, -1, -1));

labTen4.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
labTen4.setText("Tenedor4");
jPanel1.add(labTen4, new org.netbeans.lib.awtextra.AbsoluteConstraints(40, 220, -1, -1));

butFil3.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N

```

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

```

butFil3.setText("Filosofo3");
jPanel1.add(butFil3, new org.netbeans.lib.awtextra.AbsoluteConstraints(270, 310, -1, -1));

butFil4.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
butFil4.setText("Filosofo4");
jPanel1.add(butFil4, new org.netbeans.lib.awtextra.AbsoluteConstraints(70, 310, -1, -1));

butFil5.setFont(new java.awt.Font("Times New Roman", 0, 18)); // NOI18N
butFil5.setText("Filosofo5");
jPanel1.add(butFil5, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 160, -1, -1));

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 0, Short.MAX_VALUE)
);
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 113, Short.MAX_VALUE)
);

jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder("Resultados/Proceso"));


tAreaProceso.setColumns(20);
tAreaProceso.setRows(5);
jScrollPane1.setViewportView(tAreaProceso);

```

```

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(javax.swing.BorderFactory.createEmptyBorder(1,
1, 1, 1), "Numero de Veces en comer"));

```

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

```
labCom4.setText("0");
```

```
labCom5.setText("0");
```

```
jLabel5.setText("Filosofo 01");
```

```
jLabel6.setText("Filosofo 02");
```

```
jLabel7.setText("Filosofo 03");
```

```
jLabel8.setText("Filosofo 04");
```

```
jLabel9.setText("Filosofo 05");
```

```
labCom1.setText("0");
```

```
labCom2.setText("0");
```

```
labCom3.setText("0");
```

```
javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);
```

```
jPanel4.setLayout(jPanel4Layout);
```

```
jPanel4Layout.setHorizontalGroup(
```

```
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
```


```
        .addGroup(jPanel4Layout.createSequentialGroup()
```

```
            .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
```

```
                .addGroup(jPanel4Layout.createSequentialGroup()
```

```
                    .addGap(10, 10, 10)
```


```
                    .addComponent(jLabel5)
```

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

```

.addGap(36, 36, 36)
.addComponent(jLabel6)
.addGap(37, 37, 37))
.addGroup(jPanel4Layout.createSequentialGroup())
.addGap(37, 37, 37)
.addComponent(labCom1)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addComponent(labCom2)
.addGap(57, 57, 57)))
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel4Layout.createSequentialGroup())
.addComponent(jLabel7)
.addGap(34, 34, 34))
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel4Layout.createSequentialGroup())
.addComponent(labCom3, javax.swing.GroupLayout.PREFERRED_SIZE, 21,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(45, 45, 45)))
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabel8)
.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel4Layout.createSequentialGroup())
.addComponent(labCom4)
.addGap(22, 22, 22)))
.addGap(40, 40, 40)
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabel9)
.addGroup(jPanel4Layout.createSequentialGroup())
.addGap(22, 22, 22)
.addComponent(labCom5)))
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
);

```


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

```

jPanel4Layout.setVerticalGroup(
    jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel4Layout.createSequentialGroup())
            .addContainerGap()
            .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel5)
                .addComponent(jLabel6)
                .addComponent(jLabel7)
                .addComponent(jLabel8)
                .addComponent(jLabel9))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(labCom1)
                .addComponent(labCom5)
                .addComponent(labCom4)
                .addComponent(labCom3)
                .addComponent(labCom2))
            .addGap(0, 11, Short.MAX_VALUE))
);

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup())
            .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jPanel4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addGroup(jPanel3Layout.createSequentialGroup())
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 609,
javax.swing.GroupLayout.PREFERRED_SIZE)

```


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

```

        .addComponent(jPanel1,                                javax.swing.GroupLayout.PREFERRED_SIZE,      459,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

        .addComponent(jPanel3,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(122, 122, 122))

    .addGroup(layout.createSequentialGroup())

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addGap(297, 297, 297)

            .addComponent(jLabel2,                                javax.swing.GroupLayout.PREFERRED_SIZE,      320,
javax.swing.GroupLayout.PREFERRED_SIZE))

            .addGroup(layout.createSequentialGroup()

                .addGap(24, 24, 24)

                .addComponent(jButton1)))

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

            .addComponent(jPanel2,                                javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))

        .addContainerGap())

    );

    layout.setVerticalGroup(

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()

            .addContainerGap()

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

                .addGroup(layout.createSequentialGroup()


                    .addComponent(jPanel2,                                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

                    .addGap(27, 27, 27))

                    .addGroup(layout.createSequentialGroup()

                        .addComponent(jLabel2,                                javax.swing.GroupLayout.PREFERRED_SIZE,      19,
javax.swing.GroupLayout.PREFERRED_SIZE)

```


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

```

        .addGap(29, 29, 29)

        .addComponent(jButton1,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)))

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addComponent(jPanel3,                javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

        .addGap(0, 0, Short.MAX_VALUE))

        .addComponent(jPanel1,                javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

        .addContainerGap()

    );

    pack();
} // </editor-fold>


private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    JButton filosofo[];
    filosofo = new JButton[5];
    filosofo[0] = butFil1;
    filosofo[1] = butFil2;
    filosofo[2] = butFil3;
    filosofo[3] = butFil4;
    filosofo[4] = butFil5;

    Filosofo com;

    JLabel tenedor[];

```

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

```

tenedor = new JLabel[5];
tenedor[0] = labTen1;
tenedor[1] = labTen2;
tenedor[2] = labTen3;
tenedor[3] = labTen4;
tenedor[4] = labTen5;

JLabel resultado[];
resultado = new JLabel[5];
    resultado[0] = labCom1;
    resultado[1] = labCom2;
    resultado[2] = labCom3;
    resultado[3] = labCom4;
    resultado[4] = labCom5;

int i, izq, der = 0;


for (i = 0; i < 5; i++) {

    izq = i - 1;

    if (izq < 0) {
        izq = 4;
    }
    der = i;

    com = new Filosofo(i, tenedor[izq], tenedor[der], filosofo[i], resultado[i], tAreaProceso);
}
}

```


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

```

private void butFil1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
     * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
     */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Interfaz.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Interfaz.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Interfaz.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Interfaz.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
}

```

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


//</editor-fold>

/* Create and display the form */

```
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new Interfaz().setVisible(true);
    }
});
}
```

// Variables declaration - do not modify

```
private javax.swing.JButton butFil1;
private javax.swing.JButton butFil2;
private javax.swing.JButton butFil3;
private javax.swing.JButton butFil4;
private javax.swing.JButton butFil5;
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JLabel labCom1;
private javax.swing.JLabel labCom2;
```

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

```

private javax.swing.JLabel labCom3;
private javax.swing.JLabel labCom4;
private javax.swing.JLabel labCom5;
private javax.swing.JLabel labTen1;
private javax.swing.JLabel labTen2;
private javax.swing.JLabel labTen3;
private javax.swing.JLabel labTen4;
private javax.swing.JLabel labTen5;
private javax.swing.JTextArea tAreaProceso;
// End of variables declaration

```

```

class FondoPanel extends JPanel {

    private Image imagen;

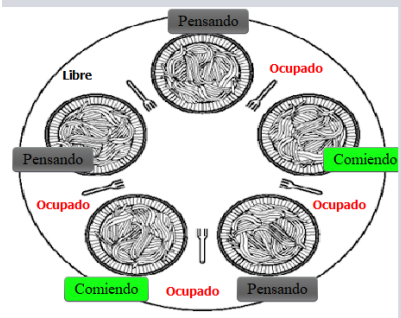
    @Override
    public void paint(Graphics g) {
        imagen = new ImageIcon(getClass().getResource("/Imagen/filosofos.png")).getImage();
        g.drawImage(imagen, 0, 0, 465, 361, this);
        setOpaque(false);
        super.paint(g);
    }
}

```

RESULTADO(S) OBTENIDO(S):

PROBLEMA DE LOS FILOSOFOS

Inicio



Resultados/Proceso

Numero de Veces en comer				
Filosofo 01	Filosofo 02	Filosofo 03	Filosofo 04	Filosofo 05
2	2	2	2	2

Fil = 3 Comiendo usa sus tenedores
Fil = 1 Comiendo usa sus tenedores
Fil = 1 Deja de comer y queda pensando, libera sus tenedores
Fil = 3 Deja de comer y queda pensando, libera sus tenedores
Fil = 5 Comiendo usa sus tenedores
Fil = 2 Comiendo usa sus tenedores
Fil = 2 Deja de comer y queda pensando, libera sus tenedores
Fil = 5 Deja de comer y queda pensando, libera sus tenedores
Fil = 1 Comiendo usa sus tenedores
Fil = 4 Comiendo usa sus tenedores
Fil = 1 Deja de comer y queda pensando, libera sus tenedores
Fil = 4 Deja de comer y queda pensando, libera sus tenedores
Fil = 5 Comiendo usa sus tenedores
Fil = 3 Comiendo usa sus tenedores
Fil = 5 Deja de comer y queda pensando, libera sus tenedores

CONCLUSIONES:
Se aplicó los conceptos sobre programación concurrente y sincronización en java.

RECOMENDACIONES:
Revisar contenido sobre hilos

Nombre de estudiante: Edith Anahí Cabrera Bermeo