	<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		


# 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.


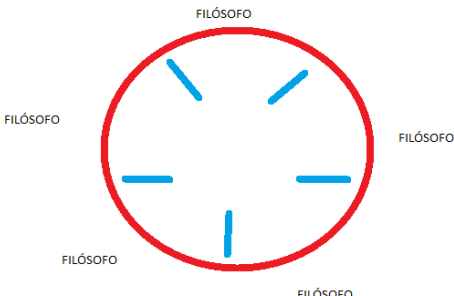
### Alcance


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		

		<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:			
<p><b>Problema del Filósofo:</b></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». 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 están en un bucle infinito dedicándose a comer y a pensar.</p> <p>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.</p>			
			

	<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		

3. Probar y modificar el método para que nos permita cambiar el número 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:** \_\_\_\_\_


		<b>FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES</b>	
<b>CARRERA:</b> COMPUTACIÓN		<b>ASIGNATURA:</b> Programación Aplicada	
<b>NRO. PRÁCTICA:</b>	1	<b>TÍTULO PRÁCTICA:</b> Hilos en Java	
<b>OBJETIVO ALCANZADO:</b>  <b>Se concreto la aplicación de concurrencia con el método run () en Java, mediante el problema clásico popular de los filósofos comensales.</b>			
<b>ACTIVIDADES DESARROLLADAS</b>			
1. Un hilo es un flujo de control dentro de un programa, el cual creando varios hilos podemos realizar varias tareas simultáneamente.			
2. Para resolver el problema se utilizó el siguiente análisis por cola de tenedores, esto quiere decir que cuando un filósofo cuando quiera comer se pone en la cola de los dos tenedores que necesita. Cuando un tenedor esta libre lo toma. Cuando toma los dos tenedores, come y deja los tenedores, por otro punto de vista, cada tenedor puede tener dos filósofos en cola, siempre los mismos.  Primero se crea la clase Filósofo la cual definimos los botones y label el cual ingresarán como parámetros desde la ventana, luego creamos los métodos comer () y pensar () los cuales se irán sincronizando mediante el método run () repitiendo así 5 veces para los 5 filósofos propuestos.			

### Clase Filosofo

```
package comenzales;
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
 */
public class Filosofo implements Runnable{
    int id,res;
    Thread t;
    JButton filosofo;
    JLabel derecho;
    JLabel izquierdo;
    JLabel resultado;
    String proceso;
    JTextArea textArea;
    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);
        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);
    }
}
```

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

```

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="Filosofo.= "+(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);


    try{

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

}

```

**Ventana**

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



### 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:

Se aplico partiendo del problema la sincronización de procesos de un sistema operativo con la ayuda del programa clásico de los filósofos comensales.

### RECOMENDACIONES:

Mostrar los resultados de los procesos mediante un textArea es muy amplio y sutil para mostrar información.

**Nombre de estudiante:** JUAN JOSE CORDOVA CALLE

**Firma de estudiante:**