	<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


	<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		

		<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> Patrones en Java																					
<b>OBJETIVO:</b> Identificar los cambios importantes de Java Diseñar e Implementar las nuevas tecnicas de programación Entender los patrones de Java																							
<b>INSTRUCCIONES</b> (Detallar las instrucciones que se dará al estudiante):		1. Revisar los conceptos fundamentales de Java																					
		2. Establecer las características de Java basados en patrones de diseño																					
		3. Implementar y diseñar los nuevos patrones de Java																					
		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 Patrones de Diseño de Java																							
2. Diseñar e implementa cada estudiante un patron de diseño y verificar su funcionamiento. A continuación se detalla el patron a implementar:																							
		<table border="1"> <thead> <tr> <th>Nombre</th> <th>Patron</th> </tr> </thead> <tbody> <tr> <td><a href="#"><u>NIXON ANDRES ALVARADO CALLE</u></a></td> <td>Factory Method</td> </tr> <tr> <td><a href="#"><u>ROMEL ANGEL AVILA FAICAN</u></a></td> <td>Builder</td> </tr> <tr> <td><a href="#"><u>JORGE SANTIAGO CABRERA ARIAS</u></a></td> <td>Abstract Factory</td> </tr> <tr> <td><a href="#"><u>EDITH ANAHI CABRERA BERMEO</u></a></td> <td>Prototype</td> </tr> <tr> <td><a href="#"><u>JUAN JOSE CORDOVA CALLE</u></a></td> <td>Chain of Responsibility</td> </tr> <tr> <td><a href="#"><u>DENYS ADRIAN DUTAN SANCHEZ</u></a></td> <td>Command</td> </tr> <tr> <td><a href="#"><u>JOHN XAVIER FAREZ VILLA</u></a></td> <td>Interpreter</td> </tr> <tr> <td><a href="#"><u>PAUL ALEXANDER GUAPUCAL CARDENAS</u></a></td> <td>Iterator</td> </tr> <tr> <td><a href="#"><u>PAUL SEBASTIAN IDROVO BERREZUETA</u></a></td> <td>Mediator</td> </tr> </tbody> </table>	Nombre	Patron	<a href="#"><u>NIXON ANDRES ALVARADO CALLE</u></a>	Factory Method	<a href="#"><u>ROMEL ANGEL AVILA FAICAN</u></a>	Builder	<a href="#"><u>JORGE SANTIAGO CABRERA ARIAS</u></a>	Abstract Factory	<a href="#"><u>EDITH ANAHI CABRERA BERMEO</u></a>	Prototype	<a href="#"><u>JUAN JOSE CORDOVA CALLE</u></a>	Chain of Responsibility	<a href="#"><u>DENYS ADRIAN DUTAN SANCHEZ</u></a>	Command	<a href="#"><u>JOHN XAVIER FAREZ VILLA</u></a>	Interpreter	<a href="#"><u>PAUL ALEXANDER GUAPUCAL CARDENAS</u></a>	Iterator	<a href="#"><u>PAUL SEBASTIAN IDROVO BERREZUETA</u></a>	Mediator	
Nombre	Patron																						
<a href="#"><u>NIXON ANDRES ALVARADO CALLE</u></a>	Factory Method																						
<a href="#"><u>ROMEL ANGEL AVILA FAICAN</u></a>	Builder																						
<a href="#"><u>JORGE SANTIAGO CABRERA ARIAS</u></a>	Abstract Factory																						
<a href="#"><u>EDITH ANAHI CABRERA BERMEO</u></a>	Prototype																						
<a href="#"><u>JUAN JOSE CORDOVA CALLE</u></a>	Chain of Responsibility																						
<a href="#"><u>DENYS ADRIAN DUTAN SANCHEZ</u></a>	Command																						
<a href="#"><u>JOHN XAVIER FAREZ VILLA</u></a>	Interpreter																						
<a href="#"><u>PAUL ALEXANDER GUAPUCAL CARDENAS</u></a>	Iterator																						
<a href="#"><u>PAUL SEBASTIAN IDROVO BERREZUETA</u></a>	Mediator																						


	<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		

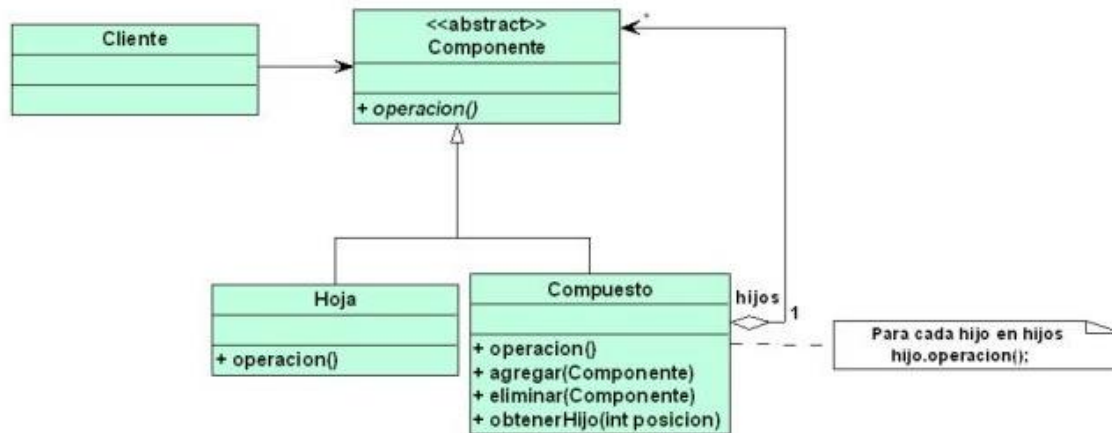
	<u><b>ADOLFO SEBASTIAN JARA GAVILANES</b></u>	Observer
	<u><b>ADRIAN BERNARDO LOPEZ ARIZAGA</b></u>	State
	<u><b>ESTEBAN DANIEL LOPEZ GOMEZ</b></u>	Strategy
	<u><b>GEOVANNY NICOLAS ORELLANA JARAMILLO</b></u>	Visitor
	<u><b>NELSON PAUL ORTEGA SEGARRA</b></u>	Adapter
	<u><b>BRYAM EDUARDO PARRA ZAMBRANO</b></u>	Bridge
	<u><b>LISSETH CAROLINA REINOSO BAJAÑA</b></u>	Composite
	<u><b>MARTIN SEBASTIAN TOLEDO TORRES</b></u>	Decorator
	<u><b>SEBASTIAN ROBERTO UYAGUARI RAMON</b></u>	Flyweight
	<u><b>ARIEL RENATO VAZQUEZ CALLE</b></u>	Proxy
	CHRISTIAN ABEL JAPON CHAVEZ	Facade
<b>3. Probar y modificar el patron de diseño a fin de generar cuales son las ventajas y desventajas.</b>		
<b>4. Realizar práctica codificando los codigos de los patrones y su estructura.</b>		
<b>RESULTADO(S) OBTENIDO(S):</b> Realizar procesos de investigación sobre los patrones de diseño de Java Entender los patrones y su utilización dentro de aplicaciones Java. Entender las funcionalidades basadas en patrones.		
<b>CONCLUSIONES:</b> Aprenden a trabajar en grupo dentro de plazos de tiempo establecidos, manejando el lenguaje de programación de Java.		
<b>RECOMENDACIONES:</b> Realizar el trabajo dentro del tiempo establecido. Revisar el siguiente link: <a href="https://refactoring.guru/es/design-patterns/java">https://refactoring.guru/es/design-patterns/java</a>		

**Docente / Técnico Docente:** \_\_\_\_\_

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

Firma: \_\_\_\_\_

		<b>FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES</b>	
<b>CARRERA:</b> Ingeniería en Computación		<b>ASIGNATURA:</b> Programación Aplicada	
<b>NRO. PRÁCTICA:</b>	3	<b>TÍTULO PRÁCTICA:</b> Patrones de diseño: Composite.	
<b>OBJETIVO ALCANZADO:</b> Identificar los cambios importantes de Java Diseñar e Implementar las nuevas técnicas de programación Entender los patrones de Java			
<b>ACTIVIDADES DESARROLLADAS</b>			
<ul style="list-style-type: none"> <li>Revisar los conceptos fundamentales de Java:</li> </ul> <b>COMPOSITE:</b> <ul style="list-style-type: none"> <li><b>¿Cuál es su propósito?</b> Es un patrón de diseño estructural que permite componer objetos en estructuras de un árbol y poder trabajar como si fueran objetos individuales.</li> </ul> <b>Partes:</b> <ul style="list-style-type: none"> <li><b>-Componente (Composite )</b> :Interfaz o clase abstracta donde se implementan las clases.</li> <li><b>-Hoja(Leaf):</b> Objeto que fue implementado a partir del componente.</li> <li><b>-Composición:</b> Componente con hojas.</li> <li><b>-Cliente:</b> Clase que maneja el composite a través de la interfaz del Componente</li> </ul> <b>ESTRUCTURA:</b> <ul style="list-style-type: none"> <li><b>1. Interfaz Componente:</b> Describe las operaciones que son comunes a elementos simples y complejos del árbol.</li> <li><b>2. Hoja:</b> Elemento básico de el árbol que no tiene subelementos.</li> <li><b>3. Contenedor:</b> También llamado <i>compuesto</i> o <i>nodo</i> es un elemento que tiene subelementos. No conoce las clases concretas de sus hijos. Funciona con todos los subelementos únicamente a través de la interfaz componente.</li> <li><b>4. Cliente:</b> Funciona con todos los elementos a través de la interfaz componente. El cliente puede funcionar de la misma manera tanto con elementos simples como complejos del árbol.</li> </ul>			



#### APLICABILIDAD:

- Se utiliza el patrón Composite cuando se tenga que implementar una estructura de objetos en forma de árbol.

-Composite proporciona dos elementos básicos que comparten una interfaz común: hojas simples y contenedores complejos. Un contenedor puede estar compuesto por hojas y otros contenedores.

- Se utiliza el patrón cuando se quiera que el código cliente trate elementos simples y complejos de la misma forma.

-Todos los elementos definidos por Composite comparten una interfaz común. Utilizando esta interfaz el cliente no tiene que preocuparse por la clase concreta de los objetos con los que funciona.

#### PROS Y CONTRAS:

- PROS:

-Se puede trabajar con estructuras de árboles complejas, utiliza el polimorfismo y la recursión a su favor.


-Principio de abierto/cerrado. Se puede introducir nuevos tipos de elementos en la aplicación sin descomponer el código existente.

- CONTRAS:

- Resulta difícil proporcionar una interfaz común para las clases cuya funcionalidad difiere demasiado. En algunos casos se tiene que generalizar en exceso la interfaz componente causando que sea complejo de comprender.

#### RELACIONES CON OTROS PATRONES:

- Builder:** Se lo puede utilizar al crear árboles Composite complejos porque se puede programar sus pasos de construcción para que funcionen de forma recursiva.
- Chain of Responsibility:** A menudo es utilizado con Composite ya que cuando un componente hoja recibe una solicitud puede pasarla a lo largo de la cadena de todos los componentes padre hasta la raíz del árbol objetos.
- Iteradores:** Se lo utiliza para recorrer el árbol Composite,
- Visitor:** Sirve para ejecutar una operación sobre un árbol Composite entero.
- Flyweights:** Se puede implementar nodos de hoja compartidos del árbol para ahorrar memoria RAM.
- Decorator:** Tiene un diagrama de estructura similar al de Composite ya que ambos se basan en la composición recursiva para organizar un número indefinido de objetos.

	<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		

**BIBLIOGRAFÍA:**

- <https://refactoring.guru/es/design-patterns/composite>
- <https://experto.dev/patron-de-diseno-java-composite/>
- <https://informaticapc.com/patrones-de-diseno/composite.php>

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

Realizar procesos de investigación sobre los patrones de diseño de Java específicamente el Patrón de diseño Composite.

Entender el patrón de diseño Composite y su utilización dentro de aplicaciones Java.

Entender las funcionalidades basadas en patrones.

**CONCLUSIONES:**

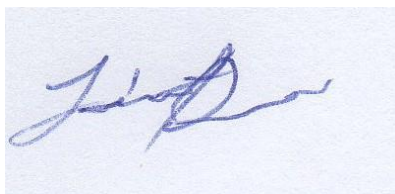
Aprenden a trabajar en grupo dentro de plazos de tiempo establecidos, manejando el lenguaje de programación de Java.

Aprender el funcionamiento de el Patrón de diseño establecido en la lista.

**RECOMENDACIONES:**

No tengo ninguna recomendación, el docente fue claro al dar la clase y la tarea.

**Nombre de estudiante:**     Lisseth Reinoso    



**Firma de estudiante:**