	<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)			
<b>1. Revisar la teoría y conceptos de Patrones de Diseño de Java</b> <b>Definición de un patrón</b> Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno y describe también el núcleo de la solución al problema, de forma que puede utilizarse un millón de veces sin tener que hacer dos veces lo mismo. <b>Definición de un patrón de diseño</b> Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí adaptada para resolver un problema de diseño general en un contexto particular. <b>Introducción</b> <ul style="list-style-type: none"> <li>• Es un tema importante en el desarrollo de software actual: permite capturar la experiencia</li> <li>• Busca ayudar a la comunidad de desarrolladores de software a resolver problemas comunes, creando un cuerpo literario de base – Crea un lenguaje común para comunicar ideas y experiencia acerca de los problemas y sus soluciones</li> <li>• El uso de patrones ayuda a obtener un software de calidad (reutilización y extensibilidad)</li> </ul> <b>Elementos de un patrón</b> <ul style="list-style-type: none"> <li>• Nombre: describe el problema de diseño.</li> <li>• El problema: describe cuándo aplicar el patrón.</li> <li>• La solución: describe los elementos que componen el diseño, sus relaciones, responsabilidades y colaboración.</li> </ul>			
<b>2. Diseñar e implementa cada estudiante un patron de diseño y verificar su funcionamiento. A continuación se detalla el patron a implementar:</b>			
		<b>Nombre</b>	<b>Patron</b>

 <b>UNIVERSIDAD POLITÉCNICA SALESIANA</b> ECUADOR	<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		

<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
<a href="#"><u>ADOLFO SEBASTIAN JARA GAVILANES</u></a>	Observer
<a href="#"><u>ADRIAN BERNARDO LOPEZ ARIZAGA</u></a>	State
<a href="#"><u>ESTEBAN DANIEL LOPEZ GOMEZ</u></a>	Strategy
<a href="#"><u>GEOVANNY NICOLAS ORELLANA JARAMILLO</u></a>	Visitor
<a href="#"><u>NELSON PAUL ORTEGA SEGARRA</u></a>	Adapter
<a href="#"><u>BRYAM EDUARDO PARRA ZAMBRANO</u></a>	Bridge
<a href="#"><u>LISSETH CAROLINA REINOSO BAJAÑA</u></a>	Composite
<a href="#"><u>MARTIN SEBASTIAN TOLEDO TORRES</u></a>	Decorator
<a href="#"><u>SEBASTIAN ROBERTO UYAGUARI RAMON</u></a>	Flyweight
<a href="#"><u>ARIEL RENATO VAZQUEZ CALLE</u></a>	Proxy
CHRISTIAN ABEL JAPON CHAVEZ	Facade

2. Probar y modificar el patron de diseño a fin de generar cuales son las ventajas y desventajas.

	<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		

## Patron ITERATOR

Intención: – Proporcionar una forma de acceder a los elementos de una colección de objetos de manera secuencial sin revelar su representación interna. Define una interfaz que declara métodos para acceder secuencialmente a la colección.

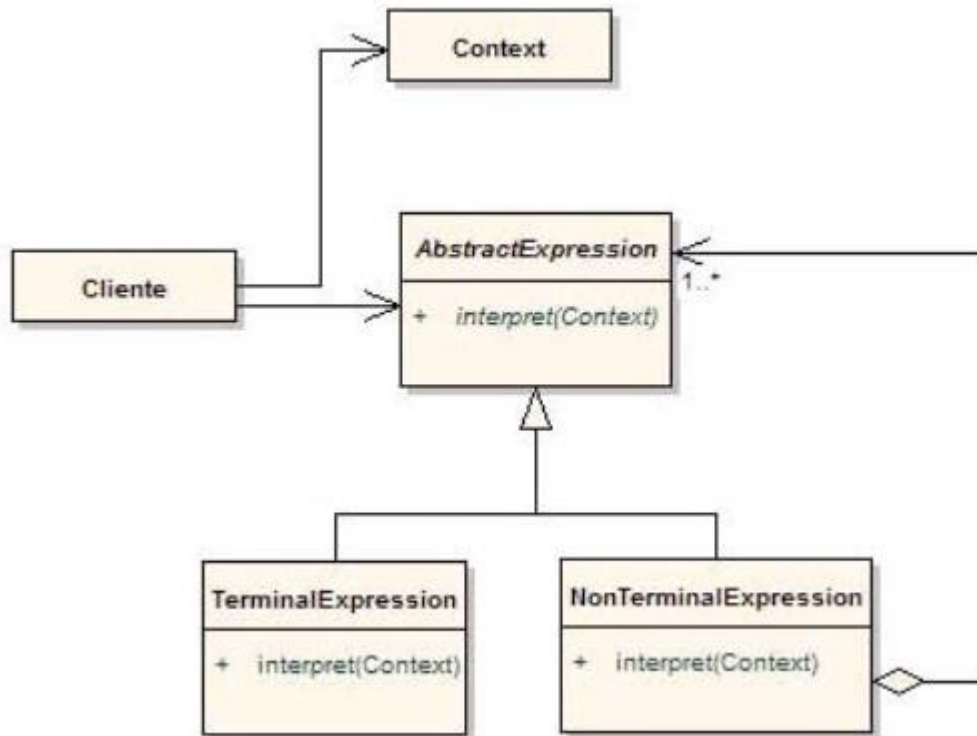
Este patrón busca representar un lenguaje mediante reglas gramáticas. Para ellos define reglas gramáticas y como interpretarlas. Se utiliza una clase para representar una regla gramática. Si un tipo partículas de problema se presentar frecuentemente, puede ser provechoso expresar los diferentes casos del problema que se resuelva el problema interpretado dichas sentencias. El patrón interprete no especifica como la estructura de objeto es creada, este procesos se conoce como parsing y compara la entrada contra la gramática para crear un pase tree(análisis de arbol) Gramática

- Podemos crear una gramática propia
  - Indicamos cuales son los términos
  - Es necesario colocar reglas de interpretación
  - Evitar ambigüedades.
- Ventajas:
    - La clase que accede a la colección solamente a través de dicho interfaz permanece independiente de la clase que implementa la interfaz.
    - Limitaciones en el tipo de gramática: si no es simple, es casi imposible implementarlo
    - Facilidad para cambiar la gramática mediante herencia dado, que las diferentes reglas se representan como objetos;
    - Facilidad para implementar la gramática, dado que las implementaciones de las clases nodo del árbol sintáctico son similares, pudiendo usarse para ellos generadores automáticos de código.
    - Facilita para introducir nuevas formas de “Interpretar” las expresiones en gramática
    - Es fácil de cambiar y ampliar la gramática.
  - Desventajas
    - Limitaciones en el tipo de gramática: si no es simple, es casi imposible implementarlo.
    - No es conveniente utilizarla si la eficiencia es un punto clave.
    - No cubre gramáticas complejas.
    - No es muy eficiente
    - Las gramáticas complejas son difíciles de mantener.

### Aplicabilidad'

Cuando hay un lenguaje que interpretar y se puedan representar las sentencias del lenguaje como arboles sintácticos abstractos las sentencias del lenguaje como arboles sintácticos abstractos.(Instancia del composite) Funciona mejor cuando la gramática es simple, la eficiencia no es una preocupación critica.

### Estructura.



El cliente y el contexto son las herramientas para poder hacer la interpretación y el lenguaje se va a definir en un lenguaje se va a definir en un árbol sintáctico abstracto a partir de la expresión abstracta y las expresiones terminales y no terminales .

**AbstractExpression:** declara una interfaz para la ejecución de una operación.

**TerminalExpression :** implementa una operación de interpretación asociada con los símbolos no terminales de la gramática.

**Context:** contiene la información a usarse para el interpreter.

**Client :** la clase que construye una estructura que representa la gramática.


3. Realizar práctica codificando los códigos de los patrones y su estructura.  
[https://github.com/Jhon14DEA/P3\\_Patrones\\_Java/tree/master](https://github.com/Jhon14DEA/P3_Patrones_Java/tree/master)

### RESULTADO(S) OBTENIDO(S):

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.

### CONCLUSIONES:

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

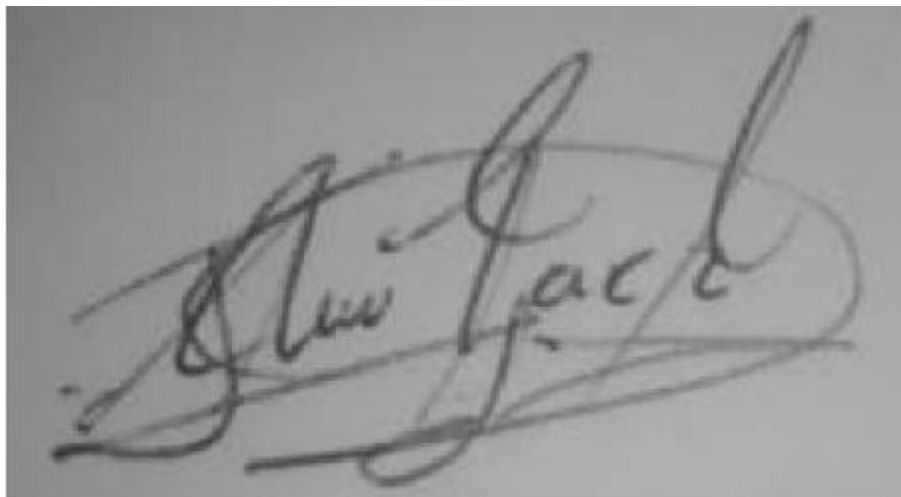
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

**RECOMENDACIONES:**

Realizar el trabajo dentro del tiempo establecido.

Revisar el siguiente link: <https://refactoring.guru/es/design-patterns/java>

**Nombre de estudiante:** Jhon xaier Farez Villa.



**Firma de estudiante**