
	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: Patrones en Java																					
OBJETIVO: Identificar los cambios importantes de Java Diseñar e Implementar las nuevas tecnicas de programación Entender los patrones de Java																							
INSTRUCCIONES (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.																					
ACTIVIDADES POR DESARROLLAR (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><u>NIXON ANDRES ALVARADO CALLE</u></td> <td>Factory Method</td> </tr> <tr> <td><u>ROMEL ANGEL AVILA FAICAN</u></td> <td>Builder</td> </tr> <tr> <td><u>JORGE SANTIAGO CABRERA ARIAS</u></td> <td>Abstract Factory</td> </tr> <tr> <td><u>EDITH ANAHI CABRERA BERMEO</u></td> <td>Prototype</td> </tr> <tr> <td><u>JUAN JOSE CORDOVA CALLE</u></td> <td>Chain of Responsibility</td> </tr> <tr> <td><u>DENYS ADRIAN DUTAN SANCHEZ</u></td> <td>Command</td> </tr> <tr> <td><u>JOHN XAVIER FAREZ VILLA</u></td> <td>Interpreter</td> </tr> <tr> <td><u>PAUL ALEXANDER GUAPUCAL CARDENAS</u></td> <td>Iterator</td> </tr> <tr> <td><u>PAUL SEBASTIAN IDROVO BERREZUETA</u></td> <td>Mediator</td> </tr> </tbody> </table>	Nombre	Patron	<u>NIXON ANDRES ALVARADO CALLE</u>	Factory Method	<u>ROMEL ANGEL AVILA FAICAN</u>	Builder	<u>JORGE SANTIAGO CABRERA ARIAS</u>	Abstract Factory	<u>EDITH ANAHI CABRERA BERMEO</u>	Prototype	<u>JUAN JOSE CORDOVA CALLE</u>	Chain of Responsibility	<u>DENYS ADRIAN DUTAN SANCHEZ</u>	Command	<u>JOHN XAVIER FAREZ VILLA</u>	Interpreter	<u>PAUL ALEXANDER GUAPUCAL CARDENAS</u>	Iterator	<u>PAUL SEBASTIAN IDROVO BERREZUETA</u>	Mediator	
Nombre	Patron																						
<u>NIXON ANDRES ALVARADO CALLE</u>	Factory Method																						
<u>ROMEL ANGEL AVILA FAICAN</u>	Builder																						
<u>JORGE SANTIAGO CABRERA ARIAS</u>	Abstract Factory																						
<u>EDITH ANAHI CABRERA BERMEO</u>	Prototype																						
<u>JUAN JOSE CORDOVA CALLE</u>	Chain of Responsibility																						
<u>DENYS ADRIAN DUTAN SANCHEZ</u>	Command																						
<u>JOHN XAVIER FAREZ VILLA</u>	Interpreter																						
<u>PAUL ALEXANDER GUAPUCAL CARDENAS</u>	Iterator																						
<u>PAUL SEBASTIAN IDROVO BERREZUETA</u>	Mediator																						


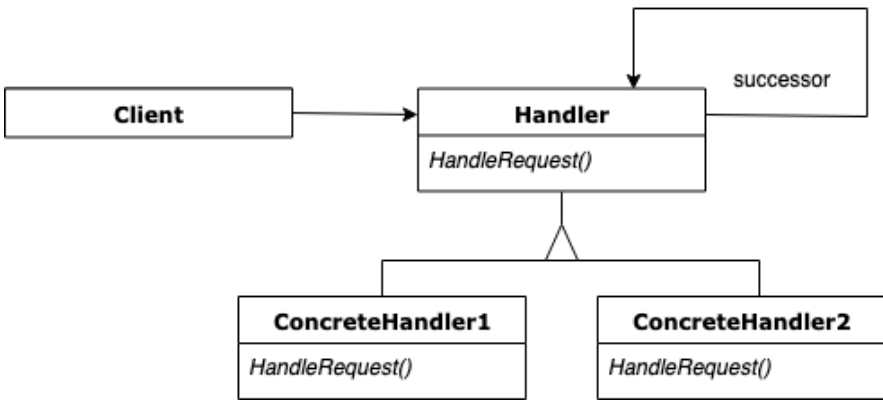
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		


	<u>ADOLFO SEBASTIAN JARA GAVILANES</u>	Observer
	<u>ADRIAN BERNARDO LOPEZ ARIZAGA</u>	State
	<u>ESTEBAN DANIEL LOPEZ GOMEZ</u>	Strategy
	<u>GEOVANNY NICOLAS ORELLANA JARAMILLO</u>	Visitor
	<u>NELSON PAUL ORTEGA SEGARRA</u>	Adapter
	<u>BRYAM EDUARDO PARRA ZAMBRANO</u>	Bridge
	<u>LISSETH CAROLINA REINOSO BAJAÑA</u>	Composite
	<u>MARTIN SEBASTIAN TOLEDO TORRES</u>	Decorator
	<u>SEBASTIAN ROBERTO UYAGUARI RAMON</u>	Flyweight
	<u>ARIEL RENATO VAZQUEZ CALLE</u>	Proxy
	CHRISTIAN ABEL JAPON CHAVEZ	Facade
3. Probar y modificar el patron de diseño a fin de generar cuales son las ventajas y desventajas.		
4. Realizar práctica codificando los codigos de los patrones y su estructura.		
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.		
RECOMENDACIONES: Realizar el trabajo dentro del tiempo establecido. Revisar el siguiente link: https://refactoring.guru/es/design-patterns/java		

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: _____

		FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES	
CARRERA: Computación		ASIGNATURA: Programación Aplicada	
NRO. PRÁCTICA:	4	TÍTULO PRÁCTICA: Patrones en Java	
OBJETIVO ALCANZADO: Aplicar nuevos conceptos investigativos acerca del patrón de diseño Chain Of Responsibility			
ACTIVIDADES DESARROLLADAS			
<p>1. Como todos los patrones de diseño, nos ayuda a solucionar problemas típicos y recurrentes que nos podemos encontrar al momento de desarrollar una aplicación por lo que nos ayudan uniformar el código haciendo que el diseño sea más comprensible para otros programadores, en este caso se dará a conocer el patrón de diseño chain of responsibility.</p> <p>El patrón de diseño Chain Of Responsibility es un patrón de comportamiento que permite mandar solicitudes a lo largo de una cadena de manejadores, por lo que, al recibir una solicitud, el manejador decide procesarla o ir al siguiente controlador de la cadena.</p> <p>La estructura genérica del Patrón es la siguiente:</p> <div style="text-align: center;">  <pre> graph TD Client[Client] --> Handler[Handler
HandleRequest()] Handler -- successor --> Handler Handler < -- ConcreteHandler1[ConcreteHandler1
HandleRequest()] Handler < -- ConcreteHandler2[ConcreteHandler2
HandleRequest()] </pre> </div>			

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

Aplicación del principio Open/Closed Principle: Permite ampliar con nuevos manejadores sin romper código existente, mejorando así la mantenibilidad.

Desventajas:

Presenta el inconveniente de no garantizar la recepción. Dado que las peticiones no tienen un receptor explícito, no hay garantías de que sean manejadas. La petición puede alcanzar el final de la cadena sin haber sido procesada.

2. Ejemplo en Java

Para este ejemplo se utilizó una sencilla aplicación que verifique si un numero es positivo y negativo y si el numero positivo esta dentro del rango 1-50

Primero se creo el manejador principal el cual se lo definió como clase abstracta

```
package manejadoresJava;

/**
 *
 * @author user
 */
public abstract class Manejador {

    protected Manejador siguienteManejador;

    public Manejador getSiguiente() {
        return this.siguienteManejador;
    }

    public void setSiguiente(Manejador m) {
        this.siguienteManejador = m;
    }

    public abstract void comprobar(int numero);
}
```

Luego la clase ManejadorPositivo

```
package manejadoresJava;

/**
 *
 * @author user
 */
public class ManejadorPositivo extends Manejador {

    public ManejadorPositivo() {
    }
}
```

```
@Override
public void comprobar(int numero)
{
    if( numero>0 ) {
        System.out.println("El número es positivo");
    }
    if( this.getSiguiente() != null ) {
        this.getSiguiente().comprobar( numero );
    }
}
}
```

Luego la clase ManejadorNegativo

```
package manejadoresJava;

/**
 *
 * @author user
 */
public class ManejadorNegativo extends Manejador {

    public ManejadorNegativo() {

    }

    @Override
    public void comprobar(int numero)
    {
        if( numero<0 ) {
            System.out.println("El número es negativo");
        }
        if( this.getSiguiente() != null ) {
            this.getSiguiente().comprobar( numero );
        }
    }
}
```


Y por último la clase ManejadorRangoPositivo

```
package manejadoresJava;

/**
 *
 * @author user
 */
public class ManejadorRangoPositivo extends Manejador {

    public ManejadorRangoPositivo() {

    }
}
```

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

```
@Override
public void comprobar(int numero)
{
    if( numero>0 && numero<=50 ) {
        System.out.println("El número está entre 1 y 50");
    }
    if( this.getSiguiente() != null ) {
        this.getSiguiente().comprobar( numero );
    }
}
}
```

Y la clase Main es la siguiente

```
package manejadoresJava;

/**
 *
 * @author user
 */
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {


        Manejador m1 = new ManejadorPositivo();
        Manejador m2 = new ManejadorNegativo();
        Manejador m3 = new ManejadorRangoPositivo();
        m1.setSiguiente( m2 );
        m2.setSiguiente( m3 );

        m1.comprobar(60); // Verificamos con el numero 60 si es positivo
        m1.comprobar(-30); // Verificamos con el numero -30 si es negativo
        m1.comprobar(40); // Verificamos con el número 40 en rango
    }

}
```

Y como Resultado tendríamos lo siguiente:

```
run:
El número es positivo
El número es negativo
El número es positivo
El número está entre 1 y 50
BUILD SUCCESSFUL (total time: 0 seconds)
|
```


	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):

Se analizó y comprobó el uso del patrón para aplicaciones dentro del ámbito de la programación genérica, se lo puede definir como un método genérico ya que esta centrado en un algoritmo de datos

CONCLUSIONES:

Se puede llegar a concluir que el uso de este Patrón puede reducir el acoplamiento, añade flexibilidad para asignar responsabilidades a objetos, pero no se garantiza la recepción.

RECOMENDACIONES:

Para aplicaciones que se requieran el uso de proceso de solicitudes es muy recomendable utilizar este patrón de diseño, sin embargo, tener en cuenta que no se puede garantizar la recepción.

Nombre de estudiante: JUAN JOSE CORDOVA CALLE

Firma de estudiante:

