

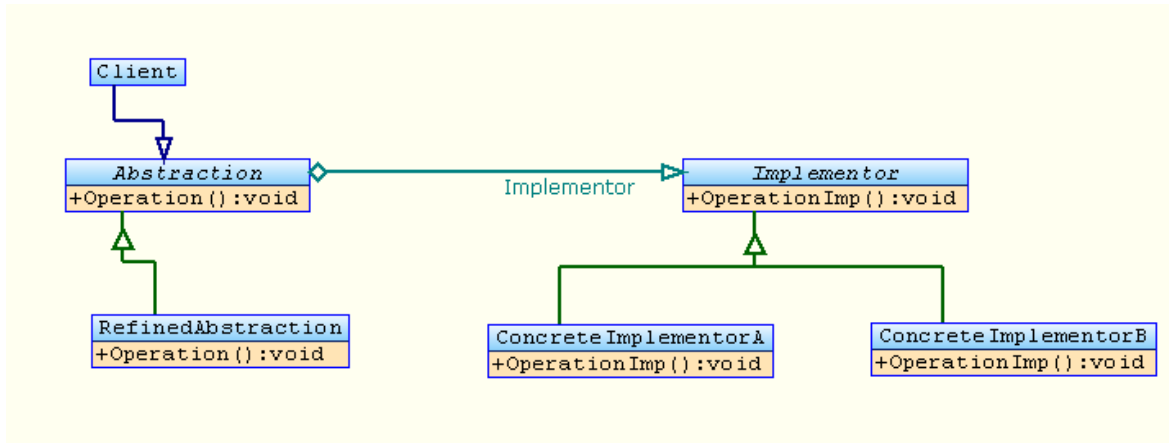
	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		<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>			
<b>1. Patrón Bridge</b> También conocido como Handle/Body, es una técnica usada para desacoplar una abstracción de su implementación, de manera que ambas puedan ser modificadas independiente sin la necesidad de alterar por ello la otra. Aumenta la flexibilidad de una jerarquía de clases y a la herencia acopla la abstracción e implementación, mejora la extensibilidad y esconde detalles de implementación a los usuarios también se usa cuando hay una nueva versión del código, pero la anterior aun es útil <ul style="list-style-type: none"> <li>Se usa cuando             <ul style="list-style-type: none"> <li>Se desea evitar un enlace permanente entre la abstracción y la implementación.</li> <li>Tanto las abstracciones como las implementaciones deben ser implementadas a subclases.</li> <li>El patrón bridge permite combinar abstracciones e implementaciones diferentes y extenderlas de manera independiente.</li> <li>La herencia acopla abstracción e implementación</li> <li>Los cambios en la implementación no deben impactar a los clientes, es decir, su código no debe ser recompilado</li> </ul> </li> </ul>			

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

## 2. Elementos



**Abstraction:** define una interfaz abstracta. Mantiene una referencia a un objeto de tipo Implementor.

**RefinedAbstraction:** Extiende la interfaz definida por Abstraction.

**Implementor:** Define la interfaz para la implementación de las clases. Esta interfaz no tiene que corresponder exactamente con la interfaz Abstraction, la interfaz Implementor provee solo operaciones primitivas, y Abstraction define operaciones de alto nivel que están basadas en estas primitivas.

**ImplementorConcreteA y B:** implementa la interfaz de Implementor y define su implementación concreta.

## 3. Ventajas y Desventajas

- Ventajas
  - Permite que la interfaz del cliente permanezca constante sin importar los cambios que tenga la abstracción.
  - No se limita permanentemente a una interface.
  - Oculta los detalles de implementación a los clientes.
  - Se pueden extender las jerarquías de Abstraction e Implementor sin que haya dependencias.
- Desventajas
  - Resulta incómodo ampliar la abstracción de la ventana para cubrir diversas clases de ventanas
  - Puede ser complicado de entender al principio
  - Complejo
  - Se tendrá problemas al no entender su funcionamiento

## 4. Ejemplo

Ejemplo se encuentra en el siguiente link;


[https://github.com/Bryambepz/Practica\\_03\\_Patron\\_Bridge/tree/master/src](https://github.com/Bryambepz/Practica_03_Patron_Bridge/tree/master/src)

## 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.

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

**CONCLUSIONES:**

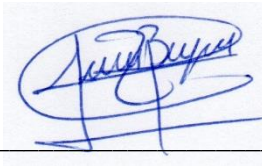
Aprenden a trabajar en grupo dentro de plazos de tiempo establecidos, manejando el lenguaje de programación de Java.

**RECOMENDACIONES:**

Realizar el trabajo dentro del tiempo establecido.

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

**Nombre de estudiante:** Bryam Parra



**Firma de estudiante:** \_\_\_\_\_