

Patrón estructural: Adapter

Juan Pablo Dios Castro Vanegas

Código:1077842737

Emili García Bermúdez

Código:1091202597

Sarita Londoño Perdomo

Código: 1091884459



Profesor:Jose Wilson Capera

Universidad del Quindío

Facultad de ingeniería

Ingeniería en sistemas y computación

Sistemas de Información

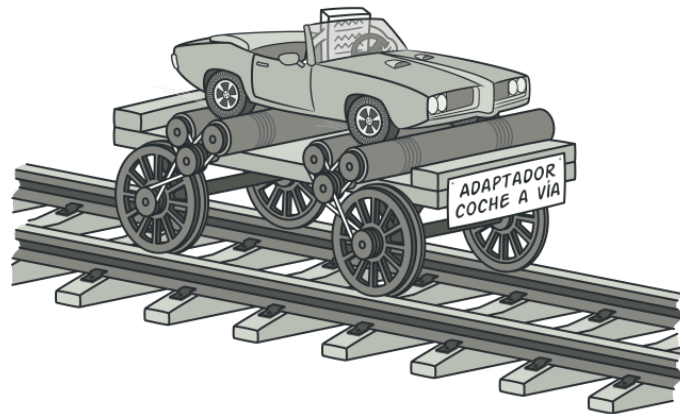
Armenia, Quindío 2024

Propósito:

El patrón Adapter se utiliza para permitir que interfaces incompatibles trabajen juntas. En otras palabras, convierte la interfaz de una clase en otra interfaz que el cliente espera. Esto se logra creando una clase intermediaria que conecta la interfaz del cliente con la interfaz de la clase existente.

En resumen, el patrón Adapter se usa para:

- Hacer que dos interfaces incompatibles sean compatibles.
- Convertir una interfaz de un tipo a otro.
- Permitir la reutilización de código antiguo en nuevos sistemas.

**Aplicaciones Típicas:**

1. Utiliza la clase adaptadora cuando quieras usar una clase existente, pero cuya interfaz no sea compatible con el resto del código: El patrón Adapter te permite crear una clase intermedia que sirva como traductora entre tu código y una clase heredada, una clase de un tercero o cualquier otra clase con una interfaz extraña.
2. Utiliza el patrón cuando quieras reutilizar varias subclases existentes que carezcan de alguna funcionalidad común que no pueda añadirse a la superclase: Puedes extender cada subclase y colocar la funcionalidad que falta, dentro de las nuevas clases hijas. No obstante, deberás duplicar el código en todas estas nuevas clases, lo cual huele muy mal.

Otras aplicaciones:

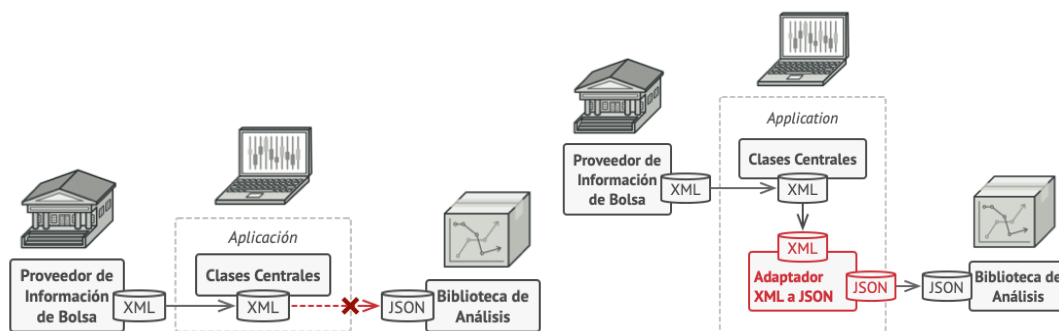
- Adaptación de interfaces de usuario: para que sean accesibles a usuarios con diferentes discapacidades.
- Adaptación de datos: para que sean compatibles con diferentes formatos o sistemas.
- Adaptación de protocolos de red: para que dos sistemas que usan diferentes protocolos puedan comunicarse entre sí.

Ventajas de usar el patrón Adapter:

- Aumenta la flexibilidad y la modularidad del código.
- Facilita la integración de diferentes componentes.
- Puede ayudar a evitar la duplicación de código.

Desventajas de usar el patrón Adapter:

- Puede agregar una capa de complejidad al código.
- Puede disminuir el rendimiento del sistema.
- Puede ser difícil de mantener a largo plazo.



Problema:

En Europa, los enchufes eléctricos tienen dos clavijas redondas, mientras que en América, los enchufes tienen dos clavijas planas y una clavija de tierra en forma de U. Un turista europeo que viaja a América no puede conectar sus dispositivos electrónicos a los enchufes americanos.

Abstracción:

1. Interfaces:

EnchufeEuropeo: Conectar un enchufe europeo a una toma de corriente.

EnchufeAmericano: Conectar un enchufe americano con tierra a una toma de corriente.

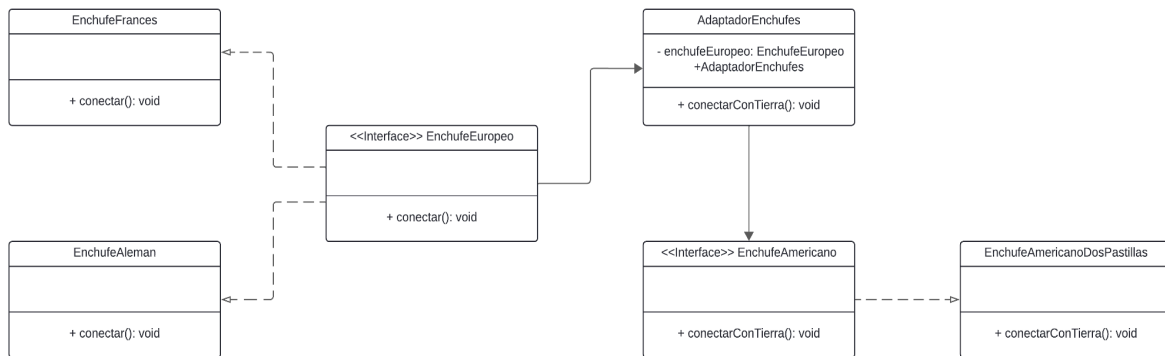
2. Clases:

EnchufeAleman, EnchufeFrances: Implementan EnchufeEuropeo para enchufes específicos.

EnchufeAmericanoDosPatillas: Implementa EnchufeAmericano para enchufes americanos de dos patillas.

AdaptadorEnchufeEuropeoAmericano: Adapta un enchufe europeo a un enchufe americano con tierra.

Diagrama de clases:



Implementacion del codigo:

```
package co.edu.uniquindio.poo;
```

```
//La interfaz EnchufeEuropeo define un método conectar().
```

```
public interface EnchufeEuropeo {
    void conectar();
}
```

```
//La clase EnchufeAleman implementa la interfaz EnchufeEuropeo y proporciona una
implementación para el método conectar().
```

```
public class EnchufeAleman implements EnchufeEuropeo{
```

```
    @Override
```

```
    public void conectar() {
        System.out.println("Enchufe aleman conectado");
    }
```

```
}
```

```
//La clase EnchufeFrances implementa la interfaz EnchufeEuropeo y proporciona una
implementación para el método conectar().
```

```
public class EnchufeFrances implements EnchufeEuropeo {
```

```
    @Override
```

```
    public void conectar() {
        System.out.println("Enchufe frances conectado");
    }
```

```
}
```

```
}
```

```
//La interfaz EnchufeAmericano define un método conectarConTierra().
```

```
public interface EnchufeAmericano {
    void conectarConTierra();
}
```

```
}
```

```
//La clase EnchufeAmericanoDosPatillas implementa la interfaz EnchufeAmericano y
proporciona una implementación para el método conectarConTierra().
```

```

public class EnchufeAmericanoDosPastillas implements EnchufeAmericano{

    @Override
    public void conectarConTierra() {
        System.out.println("Enchufe americano de dos pastillas conectado con tierra");
    }
}

public class AdaptadorEnchufes {
    private EnchufeEuropeo enchufeEuropeo;
    //Esta clase tiene un constructor que toma un objeto EnchufeEuropeo como argumento.
    public AdaptadorEnchufes(EnchufeEuropeo enchufeEuropeo) {
        this.enchufeEuropeo = enchufeEuropeo;
    }
    //Este método primero imprime un mensaje que indica que se está adaptando el enchufe europeo. Luego, llama al método conectar() del objeto EnchufeEuropeo. Finalmente, imprime un mensaje que indica que el enchufe europeo se ha conectado al enchufe americano con tierra.
    public void conectarConTierra() {
        System.out.println("Adaptando enchufe europeo...");
        enchufeEuropeo.conectar();
        System.out.println("Enchufe europeo conectado a enchufe americano con tierra.");
    }
}

//La clase App crea instancias de EnchufeAleman, EnchufeFrances y AdaptadorEnchufes. Luego, llama al método conectarConTierra() del adaptador para cada tipo de enchufe europeo.
public class App {
    public static void main(String[] args) {
        EnchufeAleman enchufeAleman = new EnchufeAleman();
        AdaptadorEnchufes adaptador = new AdaptadorEnchufes (enchufeAleman);
        adaptador.conectarConTierra();

        EnchufeFrances enchufeFrances = new EnchufeFrances();
        adaptador = new AdaptadorEnchufes(enchufeFrances);
        adaptador.conectarConTierra();
    }
}

```

Conclusión: El adaptador AdaptadorEnchufes permite que un objeto EnchufeEuropeo se use con un objeto EnchufeAmericano, El adaptador convierte la forma y el tamaño del enchufe europeo para que coincidan con las del enchufe americano, Este es un ejemplo simple de cómo se puede usar el patrón Adapter en Java

Bibliografía:

Web Content Accessibility Guidelines (WCAG) 2.1. (2023b, septiembre 21).

<https://www.w3.org/TR/WCAG21/>

Initiative, W. W. A. (s. f.). *Home.* Web Accessibility Initiative (WAI).

<https://www.w3.org/WAI/>

Adapter. (s. f.). <https://refactoring.guru/es/design-patterns/adapter>

Fundación ONCE. (s. f.). Fundación ONCE. <https://www.fundaciononce.es/>