**DISEÑO DE SOFTWARE**

**PARALELO 3**

**TAREA 3- SISTEMA ENVIVOTICKETS**

**INTEGRANTES:**

**BARRIOS URETA ROBERTO CARLOS**

**MACIAS MENDOZA CHRISTIAN JAVIER**

**TAPIA LOOR PAULO MARCELO**

**PAO II**

**2024-2025**

INDICE

[**Sección A: Diagramas UML de los patrones de diseño corregidos** 3](#_Toc188049136)

[1. Observer: 3](#_Toc188049137)

[2. Decorator: 3](#_Toc188049138)

[3. Facade: 4](#_Toc188049139)

[**Sección B: Evaluación de los patrones y reflexión sobre sus beneficios y limitaciones** 4](#_Toc188049140)

[1. Observer: 4](#_Toc188049141)

[2.Decorator: 4](#_Toc188049142)

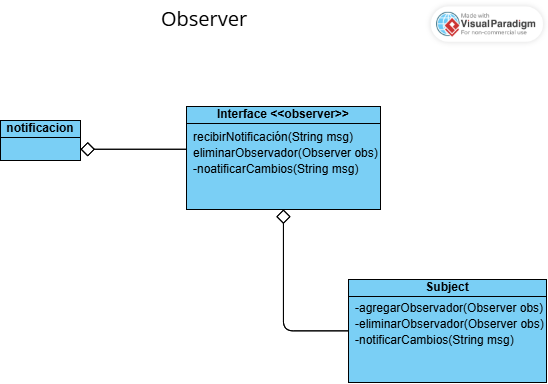
[3.Facade: 4](#_Toc188049143)

[4.Factory: 4](#_Toc188049144)

[**Sección C: Generación de Código en Java** 4](#_Toc188049145)

# **Sección A: Diagramas UML de los patrones de diseño corregidos**

## 1. Observer:



**LINK:** <https://online.visual-paradigm.com/share.jsp?id=333639363234322d36>

## 2. Decorator:

## 3. Facade:

## 4.Factory:

# **Sección B: Evaluación de los patrones y reflexión sobre sus beneficios y limitaciones**

## 1. Observer:

**Beneficios:**

* Desacoplamiento: El patrón desacopla al sujeto (Evento) de los observadores, permitiendo que estos se actualicen sin intervención directa del sujeto.
* Escalabilidad: Facilita la adición de nuevos observadores sin modificar el código existente, promoviendo la extensibilidad.
* Automatización: Automatiza la notificación de cambios, reduciendo errores humanos en la comunicación.

**Limitaciones:**

* Coste de notificación: Si hay muchos observadores, las notificaciones pueden volverse costosas en términos de rendimiento.
* Dependencia temporal: Los observadores dependen del momento en que se les notifique, lo que puede causar problemas si no están sincronizados adecuadamente.
* Complejidad adicional: Puede ser más complejo gestionar las relaciones entre sujeto y observadores, especialmente si hay múltiples jerarquías.

## 2.Decorator:

## 3.Facade:

Beneficio: Simplifica la interacción con el sistema, encapsulando operaciones complejas y ocultando detalles de implementación.

Limitación: Si crece demasiado, puede convertirse en un "dios objeto", acumulando demasiadas responsabilidades.

**2. Propuestas de mejora al diseño original**

1. **Agregar soporte para integración con servicios externos:**
   * **Propuesta:** Introducir un patrón de diseño como Adapter o Strategy para permitir la integración con diferentes servicios de pago, sistemas de envío de correos electrónicos o aplicaciones móviles.
   * **Beneficio:** Facilita la incorporación de nuevos servicios sin modificar el sistema existente.
2. **Refactorizar la clase FecadeSystem:**
   * **Propuesta:** Dividir FecadeSystem en subsistemas más específicos (como GestionEventos, GestionUsuarios, etc.).
   * **Beneficio:** Evita la acumulación de responsabilidades en una sola clase, mejorando la mantenibilidad.
3. **Implementar un sistema de gestión de errores más robusto:**
   * **Propuesta:** Introducir un patrón Chain of Responsibility para manejar errores e incidencias de soporte en diferentes niveles (usuario, administrador, soporte técnico).
   * **Beneficio:** Mejora la flexibilidad para gestionar errores complejos y permite escalar incidencias fácilmente.
4. **Ampliar la funcionalidad del patrón Decorator:**
   * **Propuesta:** Permitir que los decoradores sean configurables dinámicamente mediante archivos JSON o bases de datos.
   * **Beneficio:** Reduce la necesidad de modificar código para agregar nuevas características, haciéndolo más escalable.
5. **Implementar una base de datos para persistencia:**
   * **Propuesta:** Integrar una base de datos relacional o NoSQL para gestionar información de eventos, usuarios y tickets.
   * **Beneficio:** Mejora la escalabilidad del sistema y permite un acceso más eficiente a los datos.
6. **Soporte para múltiples idiomas:**
   * **Propuesta:** Usar un enfoque de internacionalización mediante archivos de recursos.
   * **Beneficio:** Facilita la adaptación del sistema a nuevos mercados o regiones.

# **Sección C: Generación de Código en Java**

**LINK:** [**https://github.com/RoberB1/Tarea2--EnVivoTickets**](https://github.com/RoberB1/Tarea2--EnVivoTickets)