**Patrón de diseño**

**(Observer)**

1. **Definición del patrón de diseño.**

Observer es un patrón de diseño de comportamiento que te permite definir un mecanismo de suscripción para notificar a varios objetos sobre cualquier evento que le suceda al objeto que están observando. De igual manera, define una dependencia uno a muchos entre objetos, de modo que cuando un objeto cambia su estado, todos sus dependientes son notificados y actualizados automáticamente.

1. **Justificación**

Este patrón ha sido seleccionado como base para el desarrollo de dicho sistema, ya que debido a la necesidad de automatizar múltiples procesos interrelacionados que deben reaccionar ante ciertos eventos del sistema, como una venta, una actualización de inventario o el registro de un cliente. Implementa una arquitectura flexible, en donde el evento principal puede notificar a varios módulos sin que estén conectados directamente entre sí.

1. **Características**

* Desacopla los componentes

Los diferentes módulos del sistema trabajan sin depender directamente unos de otros, lo que facilita cambiar o mejorar partes sin afectar todo.

* Reacciona automáticamente a eventos

Cuando pasa algo importante, el sistema ejecuta varias tareas al mismo tiempo sin que el código esté mezclado.

* Facilita agregar nuevas funciones

Se pueden añadir acciones sin modificar la parte central del sistema.

* Mantiene el código organizado y limpio

Cada tarea que responde a un evento está en un lugar separado, lo que hace más fácil entender y mantener el código.

* Está integrado en Laravel

Laravel tiene herramientas listas para usar que facilitan aplicar este patrón sin complicaciones extra.

* Mejora el rendimiento y las pruebas

Permite ejecutar tareas en segundo plano y hacer pruebas de cada parte por separado, ayudando a que el sistema funcione mejor y sea más confiable.