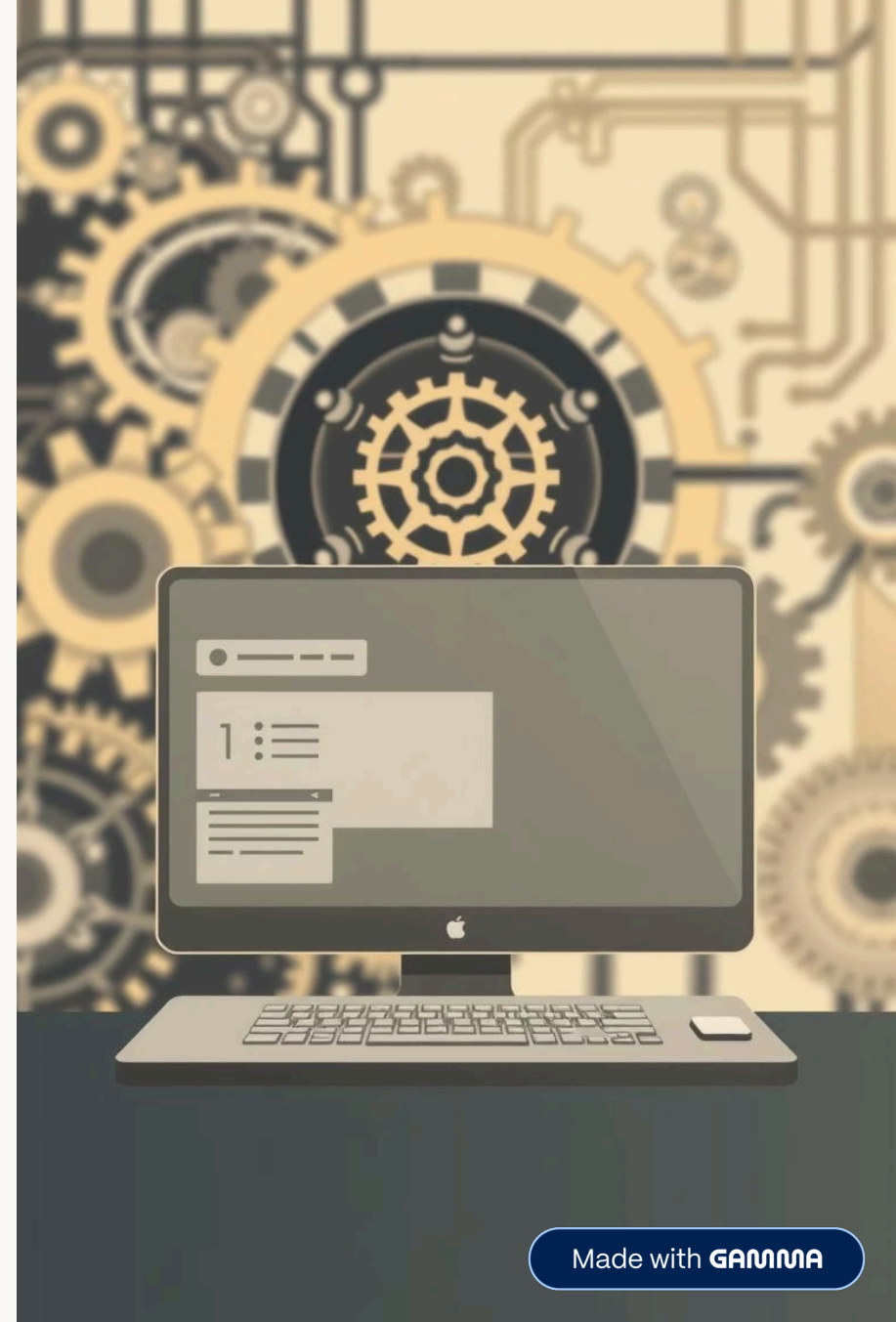


Patrón de Diseño Facade: Simplificando Sistemas Complejos

El patrón Facade, una joya en el mundo del diseño de software, nos permite domesticar la bestia de la complejidad. Su esencia reside en proporcionar una interfaz unificada y sencilla que oculta las complejidades internas de un subsistema. Esto no solo facilita la interacción con componentes sofisticados, sino que también mejora la mantenibilidad y reduce el acoplamiento, haciendo que nuestro código sea más robusto y fácil de entender. Es como tener un control remoto universal para un sistema de entretenimiento complejo.



¿Cómo Funciona el Patrón Facade?

Subsistemas Complejos

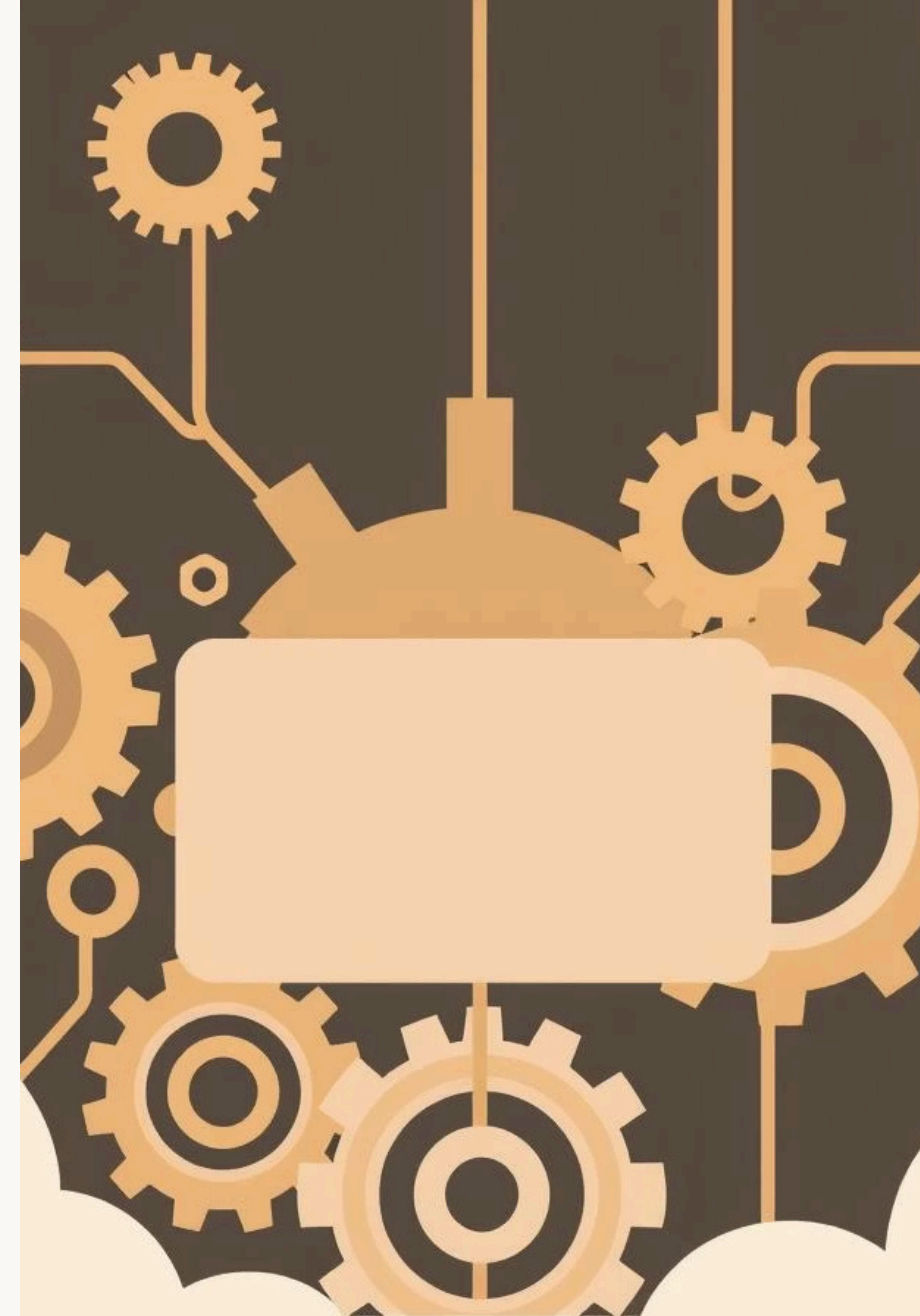
Conjunto de clases con funcionalidades específicas y múltiples dependencias. Imagina cada una como un engranaje intrincado con su propia función dentro de una maquinaria mayor.

Clase Facade

Actúa como un intermediario o "fachada". Expone solo métodos sencillos y de alto nivel que, internamente, coordinan las llamadas necesarias a los subsistemas. Es la cara amigable que oculta la maquinaria.

Cliente

Interactúa exclusivamente con la Facade. No necesita conocer ni preocuparse por la intrincada lógica interna o los componentes individuales del subsistema. Su vida se simplifica enormemente.



Ejemplo: Facade para Procesar Pedidos

Imagina un sistema de comercio electrónico donde el procesamiento de un pedido implica varias operaciones complejas: verificación de stock, procesamiento de pagos, actualización de inventario y envío de notificaciones. Un patrón Facade simplifica todo esto para el cliente.

```
class Inventario:
    def verificar_stock(self, producto_id, cantidad):
        print(f"Verificando stock para {producto_id}, cantidad {cantidad}")
        return True
```

```
class Pago:
    def procesar_pago(self, monto):
        print(f"Procesando pago de {monto}€")
        return True
```

```
class Notificacion:
    def enviar_confirmacion(self, pedido_id):
        print(f"Enviando confirmación para el pedido {pedido_id}")
```

```
class PedidoFacade:
    def __init__(self):
        self._inventario = Inventario()
        self._pago = Pago()
        self._notificacion = Notificacion()

    def realizar_pedido(self, producto_id, cantidad, monto):
        print("\n--- Iniciando procesamiento de pedido ---")
        if self._inventario.verificar_stock(producto_id, cantidad):
            if self._pago.procesar_pago(monto):
                print("Pedido realizado con éxito.")
                pedido_id = "ORD-12345"
                self._notificacion.enviar_confirmacion(pedido_id)
                print("--- Pedido procesado ---")
                return True
            print("--- Fallo al procesar el pedido ---")
            return False
```

```
# Uso del cliente
facade = PedidoFacade()
facade.realizar_pedido("LAPTOP-X", 1, 1200.00)
facade.realizar_pedido("MOUSE-GAMER", 5, 25.00)
```