

practica6

Axel Valenzuela

```
# Observer (Notificaciones)
class Observer:
    # Interfaz general que define el método "update"
    def update(self, mensaje):
        pass

class Sujeto:
    # Clase base para los que serán observados
    def __init__(self):
        self._observadores = []

    def agregar_observador(self, observador):
        self._observadores.append(observador)

    def notificar(self, mensaje):
        # Cuando algo importante sucede, se avisa a todos los observadores
        for obs in self._observadores:
            obs.update(mensaje)

# Observadores concretos
class Tablero(Observer):
    def update(self, mensaje):
        print(f"[TABLERO] Advertencia: {mensaje}")

class Appmovil(Observer):
    def update(self, mensaje):
        print(f"[APP MÓVIL] Notificación enviada: {mensaje}")

# Factory creación de vehículos
class Vehiculo(Sujeto):
    def __init__(self, modelo):
        super().__init__()
        self.modelo = modelo
        self.gasolina = 100 # tanque lleno al inicio

    def consumir_gasolina(self, cantidad):
        self.gasolina -= cantidad
        print(f"{self.modelo} consumió {cantidad}%. Gasolina restante: {self.gasolina}%")
        # Si la gasolina está baja, se lanza una alerta
```

```

        if self.gasolina <= 20:
            self.notificar(f"{self.modelo} tiene gasolina baja
({self.gasolina}%)")

class Auto(Vehiculo):
    def __init__(self):
        super().__init__("Auto")

class Moto(Vehiculo):
    def __init__(self):
        super().__init__("Moto")

class Camion(Vehiculo):
    def __init__(self):
        super().__init__("Camión")

# Fábrica de vehículos
class VehiculoFactory:
    @staticmethod
    def crear_vehiculo(tipo):
        if tipo == "auto":
            return Auto()
        elif tipo == "moto":
            return Moto()
        elif tipo == "camion":
            return Camion()
        else:
            raise ValueError("Tipo de vehículo no válido")

# SIMULACIÓN
if __name__ == "__main__":
    # 1. Pedimos un vehículo a la fábrica
    vehiculo = VehiculoFactory.crear_vehiculo("auto")

    # 2. Le conectamos observadores (tablero y app móvil)
    tablero = Tablero()
    app = Appmovil()
    vehiculo.agregar_observador(tablero)
    vehiculo.agregar_observador(app)

    # 3. Simulamos que el auto consume gasolina
    vehiculo.consumir_gasolina(30) # queda 70%
    vehiculo.consumir_gasolina(40) # queda 30%
    vehiculo.consumir_gasolina(15) # queda 15% -> lanza alerta

```

Auto consumió 30%. Gasolina restante: 70%
Auto consumió 40%. Gasolina restante: 30%
Auto consumió 15%. Gasolina restante: 15%
[TABLER0] Advertencia: Auto tiene gasolina baja (15%)
[APP MÓVIL] Notificación enviada: Auto tiene gasolina baja (15%)