

Practica 6. Factory y Observer

Ximena Valdez

```
# FACTORY
# La fábrica crea objetos de un tipo de mensaje (Email o SMS)
class Mensaje:
    def enviar(self):
        pass

class Email(Mensaje):
    def enviar(self):
        return "✉ Enviando Email..."

class SMS(Mensaje):
    def enviar(self):
        return "📱 Enviando SMS..."

class MensajeFactory:
    # La fábrica decide qué tipo de mensaje crear
    def crear_mensaje(self, tipo):
        if tipo == "email":
            return Email()
        elif tipo == "sms":
            return SMS()
        else:
            raise ValueError("Tipo no válido")

# OBSERVER
# Un Sujeto avisa a los Observadores cuando hay un nuevo mensaje
class Sujeto:
    def __init__(self):
        self.observadores = []

    def agregar(self, obs):
        self.observadores.append(obs)

    def notificar(self, mensaje):
        for obs in self.observadores:
            obs.actualizar(mensaje)

class Usuario:
    def __init__(self, nombre):
        self.nombre = nombre
```

```

    def actualizar(self, mensaje):
        print(f"{self.nombre} recibió notificación: {mensaje}")

# Creamos la fábrica y el sujeto
fabrica = MensajeFactory()
sistema = Sujeto()

# Creamos usuarios que observarán (recibirán notificaciones)
u1 = Usuario("Ana")
u2 = Usuario("Luis")
sistema.agregar(u1)
sistema.agregar(u2)

# Creamos y enviamos mensajes
msg1 = fabrica.crear_mensaje("email")
msg2 = fabrica.crear_mensaje("sms")

# El sistema notifica a los usuarios cada vez que se envía un mensaje
sistema.notificar(msg1.enviar())
sistema.notificar(msg2.enviar())

```

```

Ana recibió notificación: 📧 Enviando Email...
Luis recibió notificación: 📧 Enviando Email...
Ana recibió notificación: 📱 Enviando SMS...
Luis recibió notificación: 📱 Enviando SMS...

```