To na dúvida se salvei ele corretamente, pois o meu computador tá bugado e só aparece que deu certo quando ele quer, mas eu fiz em python e misturei os três que o senhor pediu;

Vou tentar explicar caso não abra kkkkk

**Misturei os 3 em um código só:**

**Classe pro cliente fazer o pedido**

class Pedido:

def \_\_init\_\_(self, id, cliente, valor):

self.id = id

self.cliente = cliente

self.valor = valor

**Classe que cria os pedidos, armazena em lista e avisa os observers sempre que um novo pedido for criado**

class PedidoFactory:

def \_\_init\_\_(self):

self.pedidos = []

self.observers = []

def criar\_pedido(self, cliente, valor):

id = len(self.pedidos) + 1

pedido = Pedido(id, cliente, valor)

self.pedidos.append(pedido)

self.notificar\_observers(pedido)

return pedido

def notificar\_observers(self, pedido):

for observer in self.observers:

observer.notificar(pedido)

def adicionar\_observer(self, observer):

self.observers.append(observer)

def remover\_observer(self, observer):

self.observers.remove(observer)

**Classe que valida o pedido antes de criar e recebe um objeto da classe PedidoFactory e os dados do pedido, ai ela chama o método criar\_pedido pra validar se o pedido é maior que zero.**

class PedidoProxy:

def \_\_init\_\_(self, factory, cliente, valor):

self.factory = factory

self.cliente = cliente

self.valor = valor

def criar\_pedido(self):

if self.valor <= 0:

raise ValueError("O valor do pedido deve ser maior do que zero")

return self.factory.criar\_pedido(self.cliente, self.valor)

**Classe que envia um e-mail sempre que um novo pedido for criado, ai informa os detalhes do pedido**

class EmailObserver:

def \_\_init\_\_(self, email):

self.email = email

def notificar(self, pedido):

print(f"Enviando e-mail para {self.email}: Novo pedido criado - ID: {pedido.id}, Cliente: {pedido.cliente}, Valor: {pedido.valor}")

**Colocando em ação o negócio aqui**

if \_\_name\_\_ == "\_\_main\_\_":

factory = PedidoFactory()

observer = EmailObserver("contato@lojaonline.com")

factory.adicionar\_observer(observer)

pedido\_proxy = PedidoProxy(factory, "Gabriel Dau", 150.35) <-- nome e valor

pedido = pedido\_proxy.criar\_pedido()

**Ai cria o rolê com o PedidoFactory e adiciona o EmailObserver, depois cria um PedidoProxy com o nome do cliente e o valor do pedido, e ai sim cria um novo pedido usando o método criar\_pedido lá do proxy.**