

# m13-solution

February 18, 2024

## 1 Clases y Objetos

Juliho Castillo Colmenares

<https://github.com/julihocc/ebac-python-backend>

### 1.1 Instrucciones

- 1) Visualiza qué entidades/instancias/objetos podrían ser útiles en un eCommerce.
- 2) La idea es que declares una clase con por lo menos 3 funciones y 3 atributos.
- 3) Además deberás crear una segunda clase que herede estas funciones y atributos y debes agregar funciones y atributos a esta clase hija
- 4) Incluye un ejemplo de la instanciación de objetos de ambas clases, sus funciones y sus atributos

```
[ ]: class Product:
    def __init__(self, name, price, category):
        self.name = name
        self.price = price
        self.category = category

    def get_info(self):
        return f"Product: {self.name}, Category: {self.category}, Price: ${self.
↪price}"

    def apply_discount(self, percentage):
        self.price *= (1 - percentage)
```

```
[ ]: class Cart(Product):
    def __init__(self, owner):
        self.products = [] # Lista de instancias de Product
        self.owner = owner
        self.total = 0

    def add_product(self, product):
        self.products.append(product)

    def get_cart_total(self):
        self.total = sum(product.price for product in self.products)
        return self.total
```

```
def checkout(self):
    total = self.get_cart_total()
    print(f"{self.owner}, your total is: ${total}. Proceeding to checkout.")
    self.products = [] # Vaciar el carrito después de la compra
```

```
[ ]: # Ejemplo de instanciación y uso
product1 = Product("Laptop", 1200, "Electronics")
product2 = Product("Mouse", 25, "Accessories")

cart = Cart("John Doe")
cart.add_product(product1)
cart.add_product(product2)

print(cart.get_cart_total()) # Debería mostrar el total del carrito
cart.checkout() # Finalizar la compra y mostrar el total
```

1225

John Doe, your total is: \$1225. Proceeding to checkout.