

## Final de Lenguajes de Programación III

### Enunciado:

Se desea modelar el siguiente requerimiento: “Un cliente, al **comprar** un producto, intenta pagarlo con su medio preferido, por ejemplo - tarjeta y, si no le alcanza el límite de la tarjeta, lo paga en efectivo. Cuando un producto está comprado, éste se registra como suyo.”

Años antes, se desarrollo la siguiente solución en Python 3 para dicho requerimiento:

```
class Cliente:
```

```
    def __init__(self, nombre_apellido, dni, tarjeta, efectivo, productos):
        self.nombre_apellido = nombre_apellido
        self.dni = dni
        self.tarjeta = Tarjeta()
        self.efectivo = Efectivo()
        self.productos = []
        self.preferencia = "tarjeta"

    def pagar(self, producto):
        if self.preferencia == "tarjeta":
            if self.tarjeta.limite_maximo - self.tarjeta.gastado > producto.precio:
                self.tarjeta.pagar(producto.precio)
            else:
                if self.efectivo.disponible > producto.precio:
                    self.efectivo.gastar(producto.precio)
        else:
            if self.efectivo.disponible > producto.precio:
                self.efectivo.gastar(producto.precio)
            else:
                if self.tarjeta.limite_maximo - self.tarjeta.gastado > producto.precio:
                    self.tarjeta.pagar(producto.precio)

    def comprar(self, producto):
        self.pagar(producto)
        self.productos.append(producto)
```

```
class Producto:
```

```
    def __init__(self, precio):
        self.precio = precio
```

```
class Efectivo:
    def __init__(self, disponible):
        self.disponible = disponible

    def gastar(self, plata):
        self.disponible -= plata

class Tarjeta(Efectivo):

    def __init__(self, limite_maximo, gastado):
        self.limite_maximo = limite_maximo
        self.gastado = gastado

    def pagar(self, plata):
        self.gastado += plata
```

### Tarea:

1. En base a la solución dada qué puede decir sobre: polimorfismo, herencia, declaratividad, expresividad, encapsulamiento, delegación? Están presentes? Se implementan bien dichos conceptos en el código? Justificar.
2. Escriba el código corregido en base a las críticas realizadas en el punto 1, y además, corrigiendo el **BUG conocido** que hace que se guarde el producto por más que no alcance ni tarjeta ni efectivo para pagarlo.