

**FIE****FACULTAD DE INGENIERIA DEL EJERCITO**
Universidad de la Defensa Nacional

Final de Paradigmas de Programación V

Enunciado:

La aerolínea "Ruta al cielo S.A." nos pide realizar unas pequeñas modificaciones en su sistema de venta de pasajes que tiene desarrollado en Python.

Según lo relevado la empresa cuenta con dos categorías de pasajes:

- **Clase Turista:** su precio es el precio base de la butaca.
- **Clase Business:** su precio se calcula incrementando el precio base con un recargo de un 50%.

Además, el precio de pasaje cuenta con un descuento para los estudiantes que actualmente del 30%.

Se piden las siguientes modificaciones:

1. Agregar una nueva clase "Premium" cuyo precio se calcula de manera idéntica a la categoría *business*, sumándole además una propina que actualmente es de \$200.
2. agregar una bonificación para los vuelos de invierno que resta un 10% al precio total del pasaje.

Código legacy:

```
class Pasaje:

    def __init__(self, cliente):
        self.cl = cliente

    def precio(self):
        final = 0

        if self.cl.turista():
            final = self.cl.base()

        if self.cl.business():
            final = self.cl.base() * 1.5

        if self.cl.estudiante():
            final = final * self.cl.desc()

        return final
```

**FIE****FACULTAD DE INGENIERIA DEL EJERCITO**
Universidad de la Defensa Nacional

<pre>class Cliente: def base(self): return 500.0 def desc(self): return 0.70 def turista(self): return False def business(self): return False def estudiante(self): return False</pre>	<pre>class Turista(Cliente): def turista(self): return True def base(self): return super().base() def desc(self): return super().desc()</pre>	<pre>class TuristaEst(Cliente): def turista(self): return True def estudiante(self): return True def base(self): return super().base() def desc(self): return super().desc()</pre>
<pre>class Business(Cliente): def business(self): return True def base(self): return super().base() def desc(self): return super().desc()</pre>	<pre>class BusinessEst(Cliente): def business(self): return True def estudiante(self): return True def base(self): return super().base() def desc(self): return super().desc()</pre>	

Ejemplo de Uso

```
un_pasaje = Pasaje(TuristaEst())
print(un_pasaje.precio())
➤ 350.0
```

Tarea:

1. Explique que problemas tiene el código legacy en base al paradigma de Objetos. Mencíonelos y Justifique el porqué se trata de un problema.
2. Codifique su solución con las modificaciones pedidas, teniendo en cuenta punto anterior.