

# Tarea 2 - Refactorización de Código

---

Trabajo individual

## Objetivo

---

Refactorizar el sistema de gestión de flota vehicular aplicando buenas prácticas de ingeniería de software vistas en el curso.

## Instrucciones

---

1. Analice el código provisto identificando al menos 5 problemas de diseño
2. Implemente las mejoras en un repositorio Git (opcional) o documento markdown para subir a Mediación Virtual.
3. Documente los cambios realizados explicando:
  - Problemas identificados
  - Soluciones implementadas
  - Buenas prácticas aplicadas
4. Incluya en dicho documento:
  - Explicación técnica de al menos 3 mejoras significativas
  - Enlace al repositorio si aplica o el código con los cambios respectivos.

## Código

---

```

1  class Vehiculo:
2      def __init__(self, tipo, color, peso, ruedas=4, es_electrico=False, capacidad_pasajeros=0):
3          self.tipo = tipo # 'auto', 'moto', 'camion'
4          self.color = color
5          self.peso = peso
6          self.ruedas = ruedas
7          self.es_electrico = es_electrico
8          self.capacidad_pasajeros = capacidad_pasajeros
9          self.estado = "nuevo"
10
11     def calcular_costo(self):
12         if self.tipo == 'auto':
13             base = 15000
14             extra = self.peso * 100
15             if self.es_electrico:
16                 extra += 5000
17         elif self.tipo == 'moto':
18             base = 8000
19             extra = self.peso * 50
20             if self.es_electrico:
21                 extra += 3000
22         elif self.tipo == 'camion':
23             base = 45000
24             extra = self.peso * 200
25         else:
26             base = 0
27             extra = 0
28
29         return base + extra
30
31     def necesita_inspeccion(self):
32         if self.tipo == 'auto' and self.peso > 2000:
33             return True
34         elif self.tipo == 'moto' and self.peso > 300:
35             return True
36         elif self.tipo == 'camion':
37             return True
38         else:
39             return False
40
41     def imprimir_datos(self):
42         print(f"Vehículo tipo: {self.tipo}")
43         print(f"Color: {self.color}")
44         print(f"Peso: {self.peso} kg")
45         print(f"Ruedas: {self.ruedas}")
46         print(f"Eléctrico: {'Sí' if self.es_electrico else 'No'}")
47         print(f"Capacidad: {self.capacidad_pasajeros} pasajeros")
48         print(f"Costo: ${self.calcular_costo()}")
49         print(f"Requiere inspección: {'Sí' if self.necesita_inspeccion() else 'No'}")
50         print("-----")
51
52

```

```

1  from vehiculo import Vehiculo
2
3  class Flota:
4      def __init__(self):
5          self.vehiculos = []
6
7      def agregar_vehiculo(self):
8          tipo = input("Tipo (auto/moto/camion): ").lower()
9          color = input("Color: ")
10         peso = float(input("Peso (kg): "))
11
12         if tipo == 'moto':
13             ruedas = 2
14             capacidad = 2
15         else:
16             ruedas = 4
17             capacidad = 5 if tipo == 'auto' else 2
18
19         electrico = input("Es eléctrico? (s/n): ").lower() == 's'
20
21         v = Vehiculo(tipo, color, peso, ruedas, electrico, capacidad)
22         self.vehiculos.append(v)
23         print("Vehículo agregado!")
24
25     def generar_reporte(self):
26         total = 0
27         electricos = 0
28         requiere_inspeccion = 0
29
30         for v in self.vehiculos:
31             v.imprimir_datos()
32             total += v.calcular_costo()
33             if v.es_electrico:
34                 electricos += 1
35             if v.necesita_inspeccion():
36                 requiere_inspeccion += 1
37
38         print(f"\nRESUMEN FLOTA:")
39         print(f"Total vehículos: {len(self.vehiculos)}")
40         print(f"Vehículos eléctricos: {electricos}")
41         print(f"Requieren inspección: {requiere_inspeccion}")
42         print(f"Valor total: ${total}")

```