

POO

Se te propone el siguiente “puzle” de clases y métodos:

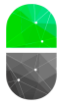


Debes crear una jerarquía de clases y herencia utilizando todos los elementos del puzle (clases, propiedades y métodos) ubicando cada elemento donde creas más conveniente. Puedes introducir elementos propios adicionales (clases, métodos o propiedades).

Este ejercicio se presta especialmente a que cada uno lo interprete de una forma diferente, así que las posibles soluciones son casi infinitas. Lo importante es que el diseño de la jerarquía de clases y herencia sea coherente.

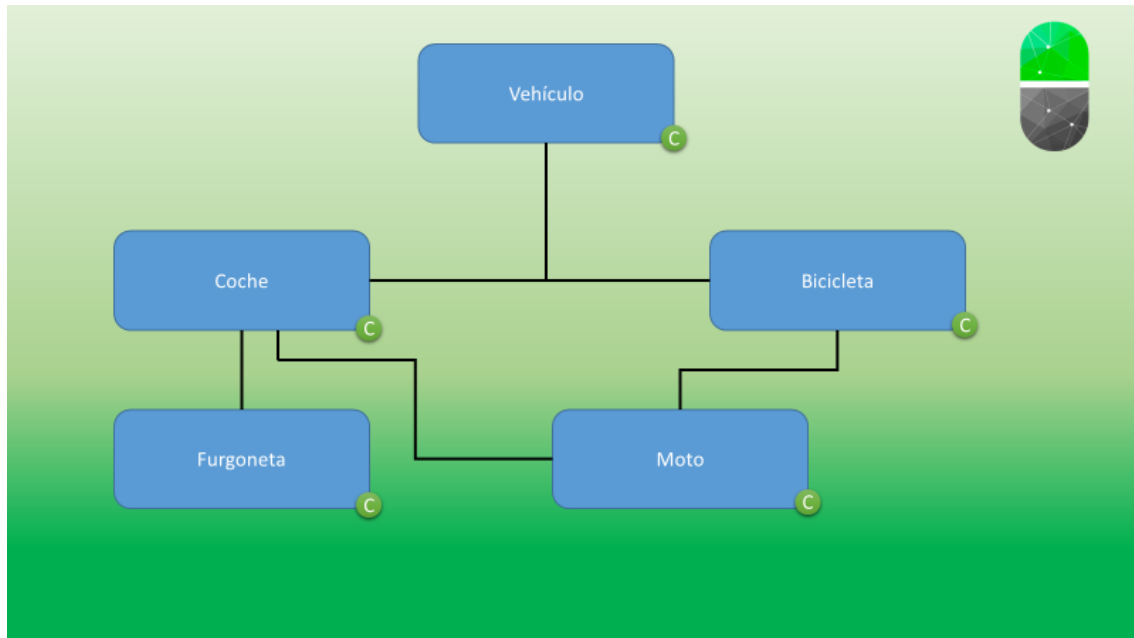
Si quieres ver una posible solución, ve a la siguiente página.





SOLUCIÓN

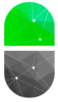
Jerarquía creada:



Código resultante:

```
1 class Vehiculo():
2
3     def __init__(self, ruedas, color, ancho, alto, marchas):
4
5         self.ruedas=ruedas
6         self.color=color
7         self.ancho=ancho
8         self.alto=alto
9         self.marchas=marchas
10        self.acelerando=False
11        self.frenando=False
12        self.girando=False
13
14    def Acelerar(self):
15
16        self.acelerando=True
17
18    def Frenar(self):
19
20        self.frenando=True
21
22    def Girando(self):
23
24        self.girando=True
25
```





```
class Coche(Vehiculo):
    def __init__(self, ruedas, color, ancho, alto, marchas, cilindrada, asientos, aire_acondicionado):
        super().__init__(ruedas, color, ancho, alto, marchas)
        self.cilindrada=cilindrada
        self.asientos=asientos
        self.aire_acondicionado=aire_acondicionado

    def Ir_Marcha_Atras(self):
        self.marcha_Atras=True

    def Arrancar(self):
        self.arrancar=True

class Furgoneta(Coche):
    def __init__(self, ruedas, color, ancho, alto, marchas, cilindrada, asientos, aire_acondicionado, carga):
        super().__init__(ruedas, color, ancho, alto, marchas, cilindrada, asientos, aire_acondicionado)
        self.carga=carga

    def Cargar(self):
        self.cargando=True
```

```
class Bicicleta(Vehiculo):
    def __init__(self, ruedas, color, ancho, alto, marchas):
        super().__init__(ruedas, color, ancho, alto, marchas)

    def Saltar(self):
        self.saltando=True

    def Derrapar(self):
        self.derrapando=True

class Moto(Coche, Bicicleta):
    def __init__(self, ruedas, color, ancho, alto, marchas, cilindrada, asientos):
        super().__init__(ruedas, color, ancho, alto, marchas, cilindrada, asientos)
```