

Gemini



2.5 Flash ▼

Fazer upgrade



```

        return f"{self.marca} {self.modelo} ligou."

class Moto(Veiculo): # Moto herda de Veiculo
    def __init__(self, marca, modelo, cilindrada):
        super().__init__(marca, modelo) # Chama o construtor
        self.cilindrada = cilindrada

    def empinar(self):
        return f"A {self.marca} {self.modelo} de {self.cilindrada} cilindrada."

minha_moto = Moto("Yamaha", "MT-07", 700)
print(minha_moto.ligar())
print(minha_moto.empinar())

```

3. **Polimorfismo:** Significa "muitas formas". Em POO, o polimorfismo permite que objetos de diferentes classes sejam tratados de forma semelhante se compartilharem uma interface comum (métodos com o mesmo nome). Isso é frequentemente alcançado através da herança e da sobrescrita de métodos.

Python



```

class Animal:
    def som(self):
        pass

class Cachorro(Animal):
    def som(self):
        return "Au au!"

class Gato(Animal):
    def som(self):
        return "Miau!"

def fazer_som(animal):
    animal.som()

```



Peça ao Gemini

Gemini



Deep Research



Canvas



Imagen



Fazer upgrade



2.5 Flash ▼

© Gemini made possible by Google. Gemini is a trademark of Google.