

- Temos o código o teste da BridgePattern:

```
class BridgePattern:
    @staticmethod
    def test():
        veiculos = [
            Caminhao(TipoCombustao()),
            Automovel(TipoHibrido()),
            Bicicleta(TipoEletrico())
        ]
        for veiculo in veiculos:
            print("Tipo Veiculo:")
            veiculo.tipo_veiculo()
            print("Tipo estrada:")
            veiculo.run()
            print("Tipo Motor:")
            veiculo.show_motor()
            print("\n")

# ----- TESTES -----

BridgePattern.test()
```

A saída do programa:

```
Tipo Veiculo:
Caminhão
Tipo estrada:
Estrada para caminhos
Tipo Motor:
Motor a Combustão

Tipo Veiculo:
Automovel
Tipo estrada:
Estrada qualquer
Tipo Motor:
Motor Híbrido

Tipo Veiculo:
Bicicleta
Tipo estrada:
Ciclovia
Tipo Motor:
Motor Elétrico
```

- Para o exemplo de construtor de pizzas usando decorators temos o código de teste:

```
# ----- TESTES -----|

pizza1 = Calabresa(Queijo(Muzzarela(Plate()))))
pizza2 = Oregano(Lombo(MolhoTomate(Plate()))))

print(pizza1.getDescription() + ": $" + str(pizza1.getTotalCost()))
print(pizza2.getDescription() + ": $" + str(pizza2.getTotalCost()))
```

E a saída do programa:

```
Plate Muzzarela Queijo Calabresa: $2.9
Plate MolhoTomate Lombo Oregano: $3.7
```