

## at\_S18\_A3\_SL07\_desenvolvimento\_software

### Roteiro de Atividade Prática

Nome: \_\_\_\_\_ Turma: \_\_\_\_\_

## **Atividade 1 - Sistema de controle de marcas de carros com estoque**

### **Objetivos**

A partir do código da aula passada, faça melhorias nele para ter um controle de estoque, criando a classe Estoque e os métodos adicionar\_carro. Faça testes e mostre os resultados.

Dica:

```
class Estoque:
```

```
    def __init__(self):
```

```
        self.__carros = [] # Inicializa a lista de carros vazia
```

```
    def adicionar_carro(self, carro):
```

```
        self.__carros.append(carro)
```

```
        print(f"Carro {carro.get_marca()} {carro.get_modelo()} adicionado ao  
estoque.")
```

Descrição do exercício:

Utilizando a linguagem de programação Python, podemos seguir o mesmo padrão utilizado para o atributo marca. Adicionaremos métodos *getters* e *setters* para cada um desses atributos.

Tempo estimado: 25 minutos

### **Lista de materiais**

- Computador com internet;
- Caderno para anotações;

- 1 caneta.

1. Anote o código desenvolvido nas linhas a seguir:

```
atividades_praticas_S18 > at_S18_A3_SL07_desenvolvimento_software_pt_1.py > Estoque > adicionar_carro
1  class Carro:
2      ...def __init__(self, marca, modelo):
3          ...self.__marca = marca
4          ...self.__modelo = modelo
5
6      ...def get_marca(self):
7          ...return self.__marca
8
9      ...def get_modelo(self):
10         ...return self.__modelo
11
12
13  class Estoque:
14      ...def __init__(self):
15          ...self.__carros = []
16
17      ...def adicionar_carro(self, carro):
18          ...self.__carros.append(carro)
19      ...def print(f"Carro {carro.get_marca()} {carro.get_modelo()} adicionado ao estoque.")
20
21
22  concessionaria_estoque = Estoque()
23
24  carro_1 = Carro("Toyota", "Corolla")
25  carro_2 = Carro("Volkswagen", "Gol")
26
27  concessionaria_estoque.adicionar_carro(carro_1)
28  concessionaria_estoque.adicionar_carro(carro_2)
29
```