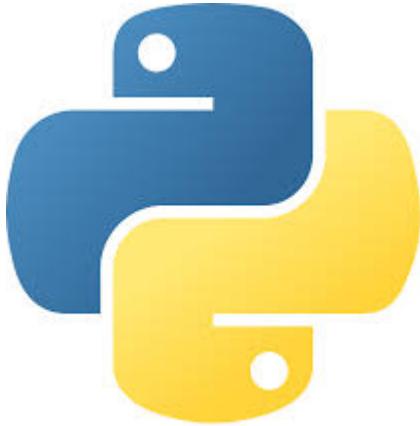


✓ **Nome:** Ruan Gabriel O. Santos

**Turma:** N2

**Disciplina:** ADS\_Tecnicas de desenvolvimento de algoritmos



```
1 # =====
2 # 2.1 Catálogo de Produtos
3 # =====
4
5 catalogo = [
6     ("Smartphone", 1500, "Eletrônicos"),
7     ("Tênis", 259, "Moda"),
8     ("Notebook", 3500, "Eletrônicos"),
9     ("Livro", 48, "Educação"),
10    ("Fone de ouvido", 209, "Eletrônicos")
11 ]
12
13 eletronicos = [produto for produto in catalogo if produto[2] == "Eletrônicos"]
14 assert len(eletronicos) == 3
15 print("Produtos eletrônicos:", eletronicos)
```

```
1 # =====
2 # 2.2 Carrinho de Compras
3 # =====
4
5 carrinho = {}
6 carrinho["Notebook"] = 1
7 carrinho["Fone de ouvido"] = 2
8 del carrinho["Notebook"]
9
10 assert "Notebook" not in carrinho
11 assert carrinho["Fone de ouvido"] == 2
```

```
12 print("Carrinho:", carrinho)
13
```

```
1 # =====
2 # 2.3 Cálculo de Total com Desconto
3 # =====
4
5 precos = [1500, 200, 3500]
6 precos_com_desconto = [p * 0.9 for p in precos]
7 total = sum(precos_com_desconto)
8
9 assert round(total, 2) == round(1500*0.9 + 200*0.9 + 3500*0.9, 2)
10 print(f"Total com desconto: R$ {total:.2f}")
11
```

```
1 # =====
2 # 2.4 Estoque por Categoria
3 # =====
4
5 estoque = {
6     "Eletrônicos": ["Smartphone", "Notebook", "Fone de ouvido"],
7     "Moda": ["Tênis", "Camisa"],
8     "Educação": ["Livro"]
9 }
10
11 categoria_maior = max(estoque, key=lambda c: len(estoque[c]))
12 assert categoria_maior == "Eletrônicos"
13 print("Categoria com mais produtos:", categoria_maior)
```

```
1 # =====
2 # 3.1 Ranking de Jogadores
3 # =====
4
5 ranking = [("Ana", 120), ("Bruno", 280), ("Carlos", 180)]
6 ranking.sort(key=lambda x: x[1], reverse=True)
7
8 assert ranking[0][0] == "Bruno"
9 print("Top 3 jogadores:", ranking[:3])
```

```

1 # =====
2 # 3.2 Inventário de Itens
3 # =====
4
5 inventario = {
6     "Espada": (1, "raro"),
7     "Poção": (5, "comum"),
8     "Armadura": (2, "épico")
9 }
10
11 raros = [item for item, (_, raridade) in inventario.items() if raridade
12 assert "Espada" in raros and "Armadura" in raros
13 print("Itens raros:", raros)

```

```

1 # =====
2 # 3.3 Histórico de Partidas
3 # =====
4
5 partidas = [
6     ("Ana", "Bruno", "Ana"),
7     ("Carlos", "Ana", "Carlos"),
8     ("Bruno", "Carlos", "Carlos")
9 ]
10
11 vitorias = {}
12 for _, _, vencedor in partidas:
13     vitorias[vencedor] = vitorias.get(vencedor, 0) + 1
14
15 assert vitorias["Carlos"] == 2
16 print("Vitórias:", vitorias)

```

```

1 # =====
2 # 3.4 Configurações do Jogo
3 # =====
4
5 config = {
6     "volume": 70,
7     "dificuldade": "médio",
8     "idioma": "português"
9 }
10
11 config["volume"] = 50
12 config["idioma"] = "inglês"
13
14 assert config["volume"] == 50
15 print("Configurações:", config)

```

```

1 # =====
2 # 4.1 Playlist de Músicas
3 # =====
4
5 playlist = [
6     {"nome": "Música A", "artista": "Artista 1", "duracao": 2.5},

```

```

7     {"nome": "Música B", "artista": "Artista 2", "duracao": 3.2},
8     {"nome": "Música C", "artista": "Artista 1", "duracao": 4.0}
9 ]
10
11 longas = [m["nome"] for m in playlist if m["duracao"] > 3]
12 assert "Música B" in longas and "Música C" in longas
13 print("Músicas com mais de 3 minutos:", longas)
14

```

```

1 # =====
2 # 4.2 Favoritar Músicas
3 # =====
4
5 favoritas = {
6     "Música A": False,
7     "Música B": True,
8     "Música C": True
9 }
10
11 favoritas_listadas = [nome for nome, fav in favoritas.items() if fav]
12 assert len(favoritas_listadas) == 2
13 print("Favoritas:", favoritas_listadas)

```

```

1 # =====
2 # 4.3 Estatísticas da Playlist
3 # =====
4
5 musicas = [("Música A", 2.5), ("Música B", 3.2), ("Música C", 4.0)]
6 total = sum([dur for _, dur in musicas])
7 media = total / len(musicas)
8
9 assert round(media, 2) == round(total / 3, 2)
10 print(f"Total: {total:.2f} min | Média: {media:.2f} min")

```

```

1 # =====
2 # 4.4 Organizar por Artista
3 # =====
4
5 por_artista = {
6     "Artista 1": ["Música A", "Música C"],
7     "Artista 2": ["Música B"]
8 }
9
10 artista_top = max(por_artista, key=lambda a: len(por_artista[a]))
11 assert artista_top == "Artista 1"
12 print("Artista com mais músicas:", artista_top)

```

