Módulo 8: Backend com C# e Python

Dia 02: O Choque de Realidade - Coleções e Decisões

João Quentino

29 de dezembro de 2025

Roteiro do Dia 02

Objetivo da Aula

Ao final desta aula, você será capaz de:

- Manipular coleções em **Python** (listas e dicionários).
- Manipular coleções em **C#** (arrays, `List<T>` e `Dictionary<K,V>`).
- Comparar abordagens **dinâmicas** (Python) vs **estáticas** (C#) na prática.
- Utilizar estruturas de decisão `if/else` para controlar o fluxo.
- Entender, conceitualmente, o uso de **LINQ** para filtrar coleções.

Aquecimento: Revisão Rápida

Do Dia 01 para o Dia 02

- Variáveis, tipos básicos e entrada de dados.
- Exibir informações no terminal.
- Agora vamos **organizar muitos dados** (produtos) usando coleções.

Motivação:

- Um mercado não tem **1** produto, tem **centenas**.
- Precisamos de estruturas que guardem listas de informações.

Listas em Python

Lista de Produtos

```
produtos = []  # Lista vazia

produto_1 = {"nome": "Teclado", "preco": 150.0, "
    quantidade": 10}
produto_2 = {"nome": "Mouse", "preco": 80.0, "
    quantidade": 25}

produtos.append(produto_1)
produtos.append(produto_2)

for p in produtos:
    print(p["nome"], p["quantidade"])
```

Pontos-chave:

- [] cria uma lista.
- append adiciona itens.

Dicionários em Python

Representando um Produto

```
produto = {
    "nome": "Monitor 24",
    "preco": 899.90,
    "quantidade": 5
}

print(produto["nome"])
produto["quantidade"] = 8  # Atualizando estoque
```

Ideia: um `dict` é como uma ficha de cadastro: chave -> valor. **Na prática da aula:** teremos **lista de dicionários** para montar o estoque.

Arrays e List<T> em C#

De Array para List

```
string[] nomes = new string[3];
nomes[0] = "Teclado";
nomes[1] = "Mouse";
nomes[2] = "Monitor";

foreach (var n in nomes)
{
    Console.WriteLine(n);
}

var produtos = new List<string>();
produtos.Add("Teclado");
produtos.Add("Mouse");
```

Resumo:

- **Array**: tamanho fixo.

Dictionary<K,V> em C#

Mapa de Produtos

```
var estoque = new Dictionary<string, int>();

estoque["Teclado"] = 10;
estoque["Mouse"] = 25;

Console.WriteLine(estoque["Teclado"]);

if (estoque.ContainsKey("Monitor"))
{
    Console.WriteLine("Tem monitor no estoque");
}
```

Idéia: chave é o nome do produto, valor é a quantidade. Mais à frente, podemos guardar um `Produto` inteiro como valor.

LINQ: Filtrando Coleções em C#

Where e Select

```
var produtos = new List<Produto>
{
    new Produto { Nome = "Teclado", Preco = 150,
        Quantidade = 10 },
    new Produto { Nome = "Mouse", Preco = 80,
        Quantidade = 2  },
    new Produto { Nome = "Monitor", Preco = 900,
        Quantidade = 0 }
};

// Filtrar produtos com estoque abaixo de 5
var baixoEstoque = produtos
    .Where(p => p.Quantidade < 5)
    .Select(p => p.Nome);
```

Analogia com Python: list comprehension.

Regras de Negócio com if/else (Python)

Verificando Estoque

```python
def status_estoque(quantidade: int) -> str:
    if quantidade == 0:
        return "Sem estoque"
    elif quantidade < 5:
        return "Estoque baixo"
    else:
        return "Estoque ok"
```

Conceito: regras de negócio em cima dos dados da coleção.

Regras de Negócio com if/else (C#)

Verificando Estoque

```
string StatusEstoque(int quantidade)
{
    if (quantidade == 0)
        return "Sem estoque";
    else if (quantidade < 5)
        return "Estoque baixo";
    else
        return "Estoque ok";
}
```

Observe: mesma lógica, sintaxe diferente.

Desafio Guiado: CRUD em Memória

Requisitos do Sistema

- Armazenar produtos em memória (sem banco de dados).
- Operações:
  - Criar produto.
  - Listar produtos.
  - Atualizar quantidade e preço.
  - Remover produto.
- Interface via linha de comando (menu textual).

Versões: uma em Python (`estoque.py`) e outra em C# (`Program.cs`).

Menu em Python (Esqueleto)

```python
while True:
    print("1 - Criar produto")
    print("2 - Listar produtos")
    print("3 - Atualizar produto")
    print("4 - Remover produto")
    print("0 - Sair")
    opcao = input("Opcao: ")

    if opcao == "1":
        # chamar funcao de criar
        ...
    elif opcao == "2":
        # listar
        ...
    elif opcao == "0":
        break
    else:
        print("Opcao invalida")
```

Menu em C# (Esqueleto)

```
while (true)
{
    Console.WriteLine("1 - Criar produto");
    Console.WriteLine("2 - Listar produtos");
    Console.WriteLine("3 - Atualizar produto");
    Console.WriteLine("4 - Remover produto");
    Console.WriteLine("0 - Sair");
    var opcao = Console.ReadLine();

    if (opcao == "0") break;

    // if/else ou switch para chamar as operacoes
}
```

Objetivo: perceber que a mesma ideia de fluxo se repete nas duas linguagens.

Resumo do Dia 02

- Coleções em Python e C#: listas, dicionários, arrays, `List<T>`, `Dictionary<K,V>`.
- Diferenças práticas entre tipagem dinâmica e estática.
- Primeiros passos com **LINQ** e comparação com list comprehensions.
- Uso de `if/else` para implementar regras de negócio.
- Sistema de estoque em memória como laboratório de tudo isso.

Próximos Passos

Vamos codar o estoque!

- Implementar o CRUD completo em Python e C#.
- Adicionar regras extras: alerta de estoque baixo, cálculo de valor total em estoque.
- Preparar o código para, no futuro, conectar com banco de dados e APIs.