

# Exercícios Práticos em Python

---

## Exercício 1: Gerenciamento de Alunos

Crie uma classe `Aluno` com os atributos `nome`, `idade` e `notas` (que será uma lista de notas). A classe deve ter os seguintes métodos:

- `adicionar\_nota(nota)`: adiciona uma nova nota à lista de notas do aluno.
- `media\_notas()`: retorna a média das notas do aluno.
- `exibir\_dados()`: exibe o nome, idade e a média das notas do aluno.

Em seguida, crie um programa que peça ao usuário para cadastrar três alunos. O programa deve permitir ao usuário adicionar notas e exibir as informações dos alunos.

## Exercício 2: Catálogo de Livros

Implemente uma classe `Livro` que tenha os atributos `titulo`, `autor`, `ano\_publicacao` e `genero`. Crie também uma classe `Biblioteca`, que terá um dicionário onde a chave será o gênero do livro e o valor será um conjunto de títulos.

Implemente os seguintes métodos na classe `Biblioteca`:

- `adicionar\_livro(livro)`: adiciona um livro ao dicionário, classificando-o pelo gênero.
- `remover\_livro(titulo)`: remove um livro pelo título.
- `listar\_livros\_por\_genero(genero)`: exibe todos os livros de um determinado gênero.

Crie um programa que permita ao usuário adicionar livros à biblioteca e listar livros por gênero.

## Exercício 3: Cadastro de Funcionários

Implemente uma classe `Funcionario` que possui os atributos `nome`, `departamento` e `salario`. O programa deve ter uma lista para armazenar objetos `Funcionario`.

Implemente as seguintes funcionalidades:

- `adicionar\_funcionario(funcionario)`: adiciona um funcionário à lista.
- `excluir\_funcionario(nome)`: remove um funcionário da lista pelo nome.
- `listar\_funcionarios\_por\_departamento(departamento)`: exibe todos os funcionários de um determinado departamento.

Crie um programa que permita cadastrar funcionários, listar por departamento e remover um funcionário pelo nome.

#### Exercício 4: Controle de Estoque de Produtos

Crie uma classe `Produto` com os atributos `codigo`, `nome` e `quantidade`. Implemente uma classe `Estoque` que utiliza um dicionário para controlar os produtos, onde a chave será o `codigo` do produto e o valor será o objeto `Produto`.

Implemente os seguintes métodos:

- `adicionar\_produto(produto)`: adiciona ou atualiza um produto no estoque.
- `remover\_produto(codigo)`: remove um produto pelo código.
- `listar\_produtos()`: exibe todos os produtos no estoque.
- `verificar\_estoque()`: retorna um conjunto com os nomes de todos os produtos que possuem mais de 10 unidades no estoque.

Crie um programa que permita ao usuário adicionar, remover e listar produtos no estoque.