



Bacharelado em Engenharia de Software

Disciplina: Programação I - Python

Atividade para complemento da segunda nota

Estruturas de Repetição e Estruturas de Dados

Projeto 1 - Sistema de Cadastro de Produtos (Mini Controle de Estoque) - 2,0 pontos

Contexto:

Uma pequena loja deseja informatizar o controle de seus produtos. O proprietário precisa de um sistema simples que permita **cadastrar, listar, buscar, atualizar e excluir produtos** do estoque.

Você e sua dupla foram contratados para desenvolver esse sistema em Python.

Requisitos do sistema:

- O sistema deve apresentar um **menu interativo**, que permaneça em execução até o usuário escolher sair. Exemplo de opções:

1 - Cadastrar produto

2 - Listar produtos

3 - Buscar produto

4 - Atualizar produto

5 - Excluir produto

0 - Sair

- Cada produto deve ser armazenado em um **dicionário**, com as seguintes chaves:



```
{"codigo": 101, "nome": "Arroz", "preco": 7.99, "quantidade": 10}
```

- Todos os produtos devem ser guardados em uma **lista**.
- Utilize um **set** para controlar os códigos de produtos já cadastrados (evitando duplicatas).
- Crie uma **tupla** com as categorias disponíveis (por exemplo: ("Alimentos", "Limpeza", "Bebidas")).
- Utilize **estruturas de repetição** (`while`, `for`) para percorrer e manipular os dados.
- O sistema deve exibir mensagens amigáveis e validar entradas incorretas.

Projeto 2 – Sistema de Controle de Alunos e Notas - 2,0 pontos

Contexto:

Uma escola precisa de um sistema simples para **registrar alunos, armazenar notas e calcular médias**. O objetivo é ajudar o professor a consultar rapidamente o desempenho da turma. Vocês, como desenvolvedores, deverão criar esse sistema em Python.

Requisitos do sistema:

- O sistema deve apresentar um **menu principal**:

- 1 – Cadastrar aluno
- 2 – Registrar notas
- 3 – Listar alunos e médias
- 4 – Buscar aluno
- 5 – Mostrar aprovados e reprovados
- 6 – Relatórios
- 0 – Sair



- O usuário pode escolher entre estes tipos de relatórios:
 - Alunos cadastrados;
 - Médias individuais;
 - Aprovados (média ≥ 7) e Reprovados (média < 7)
 - Todos os alunos devem ser armazenados em um **dicionário principal**, onde:
 - A **chave** é a matrícula do aluno.
 - O **valor** é uma **tupla** com suas notas. Exemplo:
- ```
{
 "Ana": (8.0, 7.5, 9.0),
 "Carlos": (6.0, 5.0, 7.0)
}
```
- Utilize **listas temporárias** para registrar as notas antes de transformá-las em tuplas.
  - Use **sets** para armazenar o nome dos alunos que já foram cadastrados, evitando duplicatas.
  - O cálculo da média deve ser feito dentro de um **loop**, percorrendo as estruturas e exibindo os resultados.
  - Utilize **estruturas de repetição** (`for`, `while`) para exibir alunos, notas e médias.



## Entrega dos projetos:

- O projeto deve ser feito **em dupla** e **postado no GitHub**;
- Cada dupla deve criar um repositório com o nome:

**mini-projeto-python-aluno1-aluno2**

- Enviar o link do repositório através [deste formulário](#) até às 23h59 de **03/11/2025**.
- O repositório deve conter:
  - O **arquivo principal .py** (um para cada projeto);
  - Um **README.md** explicando o funcionamento dos sistemas (menu, opções e exemplos de uso)

## Observações finais:

- Você pode acrescentar novas funcionalidades se desejar, desde que mantenha os requisitos básicos.
- Evite copiar códigos prontos da internet — o objetivo é **praticar lógica e raciocínio** com base nos conceitos estudados em aula.