

Desafio – Estrutura de Dados

Grupos: até 4 pessoas.

Objetivo: desenvolver um sistema para a coordenação de curso acompanhar a entrega e o andamento dos Trabalhos de Conclusão de Curso (TCC). Cada aluno precisa entregar versões do seu TCC, e cada versão é avaliada por orientadores em datas específicas.

O sistema deve permitir:

1. Cadastrar alunos e orientadores.
2. Registro de versões do TCC entregues.
3. Atribuição de notas por parte dos orientadores.
4. Verificação de pendências de entregas por aluno.
5. Relatórios para o orientador.

Estrutura de Aluno:

```
{  
    "nome": "João da Silva",  
    "matricula": "1234567",  
    "orientador": "Profa. Maria",  
    "entregas": [  
        ("v1", "2025-04-20", 8.5),  
        ("v2", "2025-05-15", None)  
    ]  
}
```

- Cada aluno é representado por um dicionário com quatro chaves obrigatórias: nome, matricula, orientador e entregas.

- As três primeiras têm os valores do tipo string.
- A última chave é do tipo lista de tuplas, onde cada tupla deve ter três elementos: o número da versão, a data no formato Y-m-d e a nota atribuída (opcional, pois o orientador pode não ter avaliado ainda).

Estrutura de Orientador:

```
{
    "Profa. Maria": ["João da Silva", "Fernanda Moraes"]
}
```

- Os orientadores são representados pela **chave** de um dicionário, e o seu valor são uma lista de strings, onde cada elemento deve ser o nome de um aluno.
- A chave do dicionário deve condizer com o valor da chave “orientador” na estrutura de Aluno.
- Os valores de cada elemento da lista devem estar presentes na chave “nome” na estrutura de Aluno.

Funcionalidades:

- Cadastrar alunos e seus respectivos orientadores.
- O aluno pode registrar uma nova versão do TCC, desde que não haja nenhuma versão pendente de correção por parte do orientador.
- O orientador pode atribuir a nota a uma versão do TCC.
- Apresentar as visões:
 - Listagem de alunos por orientador.
 - Versões entregues por alunos (número da versão e data).
 - Alunos com versões não avaliadas por orientador.
- O orientador deve ter visões específicas:
 - Média das notas dadas por aluno e média geral (considerando somente a última versão avaliada de cada aluno).

- Alunos com pendências (sem nota atribuída).

Orientações:

- Armazene os alunos em uma lista de dicionários.
- Armazene os orientadores em um dicionário.
- Crie funções para facilitar a manutenção do código de acordo com os requisitos solicitados nas funcionalidades acima. Exemplo: `cadastrar_orientador`, `cadastrar_aluno`, `registrar_entrega`, `registrar_nota` etc.
- O programa inicialmente deve pedir para pressionar 1 para cadastrar orientadores, 2 para cadastrar alunos ou 3 para realizar operações.
- Se a operação for 3, o programa deve listar as funcionalidades descritas na seção homônima. Sugestão:
 - Registrar nova entrega.
 - Registrar nota.
 - Listar alunos por orientador.
 - Listar versões entregues por aluno.
 - Listar pendências de avaliação.
 - Gerar relatório do orientador.
 - Voltar ao menu principal.
- Se a letra “q” for digitada ao invés dos números de opção, o programa deverá sair.
- Para verificar se há versões não avaliadas: procure na lista de entregas por tuplas onde o terceiro elemento (nota) seja `None`.
- Ao calcular médias, não se esqueça de:
 - Considerar apenas versões avaliadas.
 - Na média geral, considerar apenas a última versão avaliada de cada aluno.

Testes:

- Teste alunos com várias entregas, com e sem nota.

- Teste o cadastro de um aluno com um orientador que ainda não foi registrado (espera-se que seja impedido).
- Tente registrar uma nova entrega sem que a anterior tenha sido avaliada (deve ser bloqueado).