

AC 5 – Sala de Aula (Servidor)

Arquivos

Arquivo para testar seu servidor:

sala_de_aula_unittests.py

Arquivo para iniciar a atividade:

sala_de_aula.py

Instruções

Para iniciar os testes, execute o arquivo sala_de_aula.py para iniciar o seu servidor.

```
$ python sala_de_aula.py
```

Execute os testes

```
$ python sala_de_aula_unittests.py
```

Quando seu servidor tiver implementado corretamente todas as funções seus testes irão passar.

Você deve implementar todas as funções ainda não implementadas!

Dicas

Você vai precisar manipular a lista de alunos e professores. Se tiver dúvidas sobre como incluir, alterar ou remover itens de uma lista: https://www.w3schools.com/python/python_lists.asp

Vai precisar consultar chaves de um dicionário. Exemplo: aluno['id']. Para mais instruções sobre como trabalhar com dicionários: https://www.w3schools.com/python/python_dictionaries.asp

O arquivo sala_de_aula.py já contém as referências às bibliotecas que você irá precisar (flask) e já contém a descrição das funções com uma sugestão de verbos e nome de métodos.

É boa prática nomear métodos como verbos. Cadastrar, Alterar, Remover, etc.

Os retornos são do tipo 500 pois a função não está implementada. Quando o usuário chamar um método de maneira errada, ele deverá receber 400. Exemplo: tentou cadastrar um aluno sem passar o id. Outro exemplo: Tentou alterar um aluno e não informou o nome a ser alterado, etc.

Utilize os retornos esperados os unittests para entender o que retornar quando uma ação inválida for proposta.

Exemplo:

No teste abaixo, o aluno solicitado não existirá:

```
r = requests.get('http://localhost:5002/alunos/15')
self.assertEqual(r.status_code, 400)
self.assertEqual(r.json()['erro'], 'aluno nao encontrado')
```

Então, quando for consultado um aluno que não existe, o retorno deverá ser algo assim:

```
return jsonify({'erro': 'aluno nao encontrado'}), 400
```