

		<b>Primeira Avaliação Parcial Segundo Semestre de 2024</b>	
Curso: <b>Bacharelado em Engenharia de Software</b>			
Disciplina: <b>Backend development</b>		Turma: <b>A</b>	Período: <b>5/6</b>
Professor: <b>Paulo Sergio da Conceição Moreira</b>		Data: <b>10/10/2024</b>	Nota:
<b>Aluno (a):</b>			
<b>Instruções:</b> <ul style="list-style-type: none"><li>• A avaliação pode ser realizada <b>individualmente, em duplas ou em trios</b>. Quaisquer plágios ou cópias de colegas, se evidenciadas, resultarão na anulação da nota desta avaliação.</li><li>• A avaliação deverá ser respondida em forma de relatório, <b>contendo todas as informações necessárias para a replicação da solução</b>;</li><li>• Lembre-se: ao enviar para avaliação, <b>encaminhe os códigos e os arquivos necessários (Excel, JSON, links) em conjunto com o relatório principal</b>. Você pode compactar tudo em um arquivo (.zip, .rar);</li><li>• A entrega da avaliação deverá ser feita até o dia <b>10/10/2024, às 22:10, via AVA (Moodle)</b>.</li></ul>			

Nesta prova, você deverá criar uma API RESTful utilizando Flask como framework de backend e MongoDB como banco de dados. A avaliação será baseada na sua capacidade de implementar todos os métodos HTTP básicos, garantir uma boa organização do código, tratar erros de forma adequada e documentar claramente os endpoints.

A API será responsável por gerenciar eventos, permitindo operações como criação, leitura, atualização e exclusão.

Cada evento deverá conter as seguintes propriedades:

1. id: gerado automaticamente pelo MongoDB.
2. título: string, obrigatório.
3. descrição: string, obrigatório.
4. local: string, obrigatório.
5. capacidade: inteiro, obrigatório (deve ser maior que zero).

Sua API deve permitir:

1. Criar um novo evento;
2. Listar todos os eventos;

3. Buscar um evento a partir do ID ou por um conjunto de parâmetros;
4. Atualizar um evento a partir do ID ou por um conjunto de parâmetros;
5. Deletar um evento a partir do ID ou por um conjunto de parâmetros.

**Tarefas:**

1. Crie 10 eventos e armazene-os no MongoDB. **(1,0 ponto).**
2. Demonstre o correto funcionamento de todos os *endpoints*, gravando a execução. **(2,0 pontos).**
3. Acrescente validações e tratamentos de exceções para garantir que a API retorne erros apropriados e mensagens claras. **(1,0 ponto).**

**Observações:**

1. Para todas as questões, apresente o código-fonte e os arquivos necessários (JSON, arquivos de coleção do Insomnia/Postman) que possibilitem a sua correção. **Soluções sem códigos serão desconsideradas.**
2. O teste da API pode ser feito com qualquer ferramenta apropriada (e.g., Insomnia, Postman, Curl).
3. **Apenas soluções desenvolvidas com Flask e MongoDB serão aceitas.**
4. Caso a prova seja realizada por mais de uma pessoa, **inclua no relatório o nome de todos os integrantes.**