

Projeto Web Back-end

API REST

O projeto web contemplado pela disciplina visa permitir que os alunos apliquem os conceitos e temas abordados em aula. O domínio do sistema a ser desenvolvido é de livre escolha de cada aluno, desde que atenda os requisitos definidos a seguir. O projeto pode contemplar um trabalho de outra disciplina, o início de um trabalho de TCC ou algo pessoal e pode ser desenvolvido preferencialmente em duplas (opcionalmente de modo individual). **O projeto se trata da construção de uma API REST, com persistência em arquivos e tratamento de exceções.** O sistema deve atender aos seguintes requisitos:

Tecnologias

Deverão ser empregadas as seguintes tecnologias na construção do projeto:

- Framework: **será utilizado o framework Express**, juntamente com os pacotes apresentados em sala de aula. A utilização de alguma ferramenta adicional deverá ser consultada, sob penalidade de invalidação do trabalho.
- Persistência: **A persistência dos dados deverá ser realizada em arquivos no formato JSON. Será necessário gerenciar um arquivo para cada coleção/modelo.**
- **Todas as funcionalidades deverão ser implementadas em formato de API REST não sendo necessário o desenvolvimento de uma interface. Portanto, os testes deverão ser realizados utilizando uma ferramenta específica para esta finalidade como Nodemon, Insomnia, Talend, etc.**

Usuários e sistema de autenticação

Os seguintes requisitos deverão ser implementados em relação ao gerenciamento e controle de usuários:

1. **O sistema deverá possuir uma rota que permite o cadastro de usuários.** A rota deve receber os dados pessoais e as credenciais (usuário e senha) para autenticação na API.
2. **O sistema deverá possuir um (ou mais) usuários administradores que possuem privilégios específicos como alterar e excluir outros usuários e criar outros administradores. A instalação do sistema deverá criar um usuário administrador por padrão na aplicação.**

3. Deverá haver uma rota para que os administradores possam criar outros administradores.
4. Deverá haver uma rota para que os administradores possam excluir um usuário não administrador.
5. A rota de login deverá receber o usuário e senha e gerar um token JWT que permitirá acesso às rotas protegidas da API
6. Um usuário poderá alterar seus dados pessoais por meio de uma rota específica. Os usuários comuns não poderão alterar dados de outros usuários, todavia os administradores poderão.

Sistema CRUD

Como requisito principal, o sistema deve permitir a realização de 2 cadastros (além dos usuários) com operações de CRUD completa, tal que, estes itens apresentem entre si algum relacionamento, conforme a livre escolha de cada projeto. Obrigatoriamente as operações de inserção, alteração e exclusão devem ser restritas para o usuário autenticado no sistema (que possuem um token válido). A restrição do acesso para as operações de listar e buscar pelo identificador único são de livre escolha do desenvolvedor, conforme o tema proposto.

É necessário realizar a validação adequada dos dados fornecidos pelo usuário, e gerar mensagens de erros personalizadas conforme o erro obtido. As mensagens de erros e sucessos deverão ser enviadas juntamente com as respostas. Os métodos HTTP GET, POST, PUT e DELETE devem ser empregados segundo a operação a ser executada.

Os métodos de listar deverão implementar a paginação dos dados, de tal forma que eles devem receber 2 parâmetros: **limite** e **página**. O atributo **limite** define quantos objetos devem ser retornados (os valores possíveis deverão ser 5, 10 e 30) na consulta. O atributo **página** define o ponto em que começa o retorno. Por exemplo, limite=5 e página=1, retorna os 5 primeiros registros; limite=5 e página=3, ignora os 10 primeiros registros e retorna do 11º ao 15º registro.

Lógica de negócio, instalador e documentação

Deverá ser implementado alguma operação especial de livre escolha do aluno ou dupla (disponível por uma ou mais rotas) implementando uma lógica de negócio que pode envolver inserção/alteração no banco de dados, geração de consultas elaboradas e/ou algum processamento dos dados sejam eles recebidos por parâmetros ou do próprio banco de dados.

Além disso, deverá ser criada uma rota **GET /install/** que cria um usuário administrador no sistema.

Deverá ser criada uma rota **GET /docs** contendo a documentação gerada pela ferramenta Swagger.

Por fim, deverá ser adotada uma estratégia de tratamento de erros, com utilização de middleware de erros e objetos de erros personalizados.

Prazo de entrega

A ser definido posterior a finalização da greve

Considerações finais

- Trabalhos similares ou plágio da internet serão zerados (se constatado alta similaridade com outros códigos), portanto, evite copiar códigos ou seguir a risca tutoriais/vídeo-aulas/chat-gpt.
- O código deverá estar disponível em um repositório GIT, os commits devem ser incrementais. Será considerado um percentual da nota para este requisito.
- Deverá ser utilizado um arquivo (.env) com as configurações do projeto (pesquise sobre dotenv para mais detalhes).
- A organização da arquitetura do projeto e definição das rotas é de livre escolha e também será um critério avaliado, portanto, mantenha o código organizado e comentado.
- Não utilizar pacotes e ferramentas não utilizados durante a aula sem o conhecimento e ciência prévia do professor.
- As demais dúvidas e questões que porventura surjam serão resolvidos em comum acordo entre alunos e professores, sendo a palavra final do professor :)