

# Projeto Web Back-end

## Especificação do Sistema

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.

O projeto se trata da construção de uma API web back-end. Todavia, para aqueles que desejam recuperar a nota da avaliação 1 poderão desenvolver a interface front-end e promover sua integração com a API, conforme especificado a seguir. O projeto 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.
- Banco de dados: será de livre escolha de cada aluno, podendo ser um banco de dados relacional (ex. MySQL ou PostgreSQL) utilizando o sequelize ou um banco de dados NoSQL (ex. MongoDB).
- 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, Talend, etc.
- Os alunos que precisarem realizar a recuperação da avaliação 1 deverão utilizar as tecnologias HTML, CSS e Javascript para a criação da interface front-end e integração com a API desenvolvida. A critério do aluno, poderá ser utilizado algum framework como Bootstrap, React, Vue ou Angular (livre escolha).

## Usuários e sistema de autenticação (40%)

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 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 (valor: 30%)

Como requisito principal, o sistema deve permitir a realização de pelo menos 3 cadastros (operações de CRUD completa), tal que, estes itens apresentem entre si relacionamentos de um-para-muitos ou muitos-para-muitos, de acordo com a livre escolha de cada aluno. 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, de acordo com o tema proposto.

É necessário realizar a validação adequada dos dados fornecidos pelo usuário, e gerar mensagens de erros personalizadas de acordo com 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 de acordo com 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 e instalador (valor: 30%)

Deverá ser implementado alguma operação especial de livre escolha do aluno (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 criado uma rota **GET /install/** que realiza a instalação do banco de dados (criação das tabelas e inserção de dados no banco). Cada tabela deverá ser populada com pelo menos 5 registros.

## Recuperação da avaliação 1 (sistema front-end)

Para os alunos que precisam da recuperação de nota da avaliação 1, estes deverão implementar também um front-end para integração da API desenvolvida que deve contemplar os seguintes requisitos:

1. Autenticação (formulário de login)
2. Sistema de CRUD envolvendo uma das tabelas que possui relacionamento (um-para-muitos/muitos-para-muitos) com outra tabela e a paginação da listagem relativa a este cadastro (apenas para usuário logado no sistema)
3. Interface da lógica de negócio (apenas para usuário logado no sistema)

## Considerações finais

- O desenvolvimento da API é individual.
- 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 trabalho deverá ser apresentado ao professor e eventualmente submetido a alguma mudança em tempo real.
- O código deverá estar disponível em um repositório GIT, os commits devem ser incrementais.
- Deverá ser utilizado um arquivo (.env) com as configurações do banco de dados (pesquise sobre dotenv para mais detalhes).

- A organização da arquitetura do projeto e definição das rotas é de escolha individual e também será um critério avaliado, portanto, mantenha o código organizado.