

Programação Web 2

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 será organizado em 2 ciclos, sendo que a apresentação dos ciclos devem ser realizadas na ordem especificada.

Ciclo 1 - Site de apresentação (valor: 30%)

Neste primeiro ciclo, cada projeto deverá atender aos seguintes requisitos e realizar as seguintes atividades:

- Definir um nome (e um logo) para o sistema
- Criar um repositório na plataforma GitHub
- Definir um template e padrão visual para o site (cabeçalho e rodapé)
- Criar um arquivo para conter as configurações da aplicação web como senhas e os parâmetros necessários para o funcionamento da aplicação web (pesquisar sobre dotenv)
- Criar uma página inicial para o site com um breve resumo da proposta e os links para as demais páginas, a página inicial também deve possuir um formulário de login. Neste momento o mesmo não precisará funcionar
- Criar uma página com uma descrição detalhada sobre o que será desenvolvido
- Criar uma página para listar as tecnologias e ferramentas que serão utilizadas (todas as bibliotecas utilizadas deverão ser listadas nesta página)
- Criar uma página para descrever o desenvolvedor do projeto e demais pessoas que porventura estejam relacionados ao projeto
- Criar uma página de contato, no qual será possível enviar uma mensagem (por email) para os desenvolvedores da ferramenta. Os campos obrigatórios são nome, e-mail, assunto e mensagem (pesquisar sobre como enviar emails)

Para o desenvolvimento do projeto deverá ser obrigatoriamente utilizado o framework Express.js. Os demais pacotes e recursos empregados são de livre escolha de cada desenvolvedor. Neste ciclo, não deverá ser criada uma API para separação entre *back-end* e *front-end*, de tal forma, que o conteúdo e os dados deverão ser disponibilizados em um único servidor. Todavia, é permitido o uso de um *framework front-end* como o React, Vue ou Angular, desde que integrados em um único projeto.

Com relação ao conteúdo das páginas de **conteúdo**, **tecnologia** e **sobre**, estes deverão estar armazenados em arquivos de texto ou template, para que seja possível alterá-los sem a necessidade de manipular o código fonte do site.

Os valores recebidos pelo usuário deverão ser devidamente validados e mensagens de erros apropriadas devem ser apresentadas ao usuário.

Ciclo 2 - Sistema (valor: 40% back-end e 30% front-end)

Nesta segunda etapa do projeto, o objetivo é criar o sistema dinâmico integrado com um banco de dados (relacional ou NoSQL), por meio de uma API web (*back-end*) e um cliente web (*front-web*) que irá consumir a API desenvolvida.

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 um relacionamento de um-para-muitos ou muitos-para-muitos, de acordo com a livre escolha de cada aluno. Um destes cadastros deve ser do usuário, e as demais operações de inserção, alteração e exclusão devem ser restritas para o usuário autenticado no sistema. As operações de listar e buscar pelo identificador único são permitidos também para os usuários não logados no sistema. De acordo com o tema proposto.

Neste ciclo, será necessário fazer com que o *front-end* da aplicação consuma a API desenvolvida. E realize a validação dos dados fornecidos pelo usuário, além de realizar o tratamento dos erros e mensagens geradas pela API. Neste cenário, será permitido criar uma aplicação diferente para o *front-end* da aplicação, desde que a mesma seja adequadamente integrada com a solução desenvolvida no primeiro ciclo.

Com relação a API desenvolvida, a mesma também deverá realizar a validação dos campos, considerando que os dados podem vir de uma fonte diferente do cliente desenvolvido. Além disso, mensagens de erros e sucessos deverão ser enviados juntamente com as respostas. Os métodos HTTP GET, POST, PUT e DELETE devem ser empregados de acordo com a operação a ser executada. Quanto ao banco de dados, será de livre escolha de cada aluno, podendo ser um banco de dados relacional (ex. MySQL ou PostgreSQL) ou um banco de dados NoSQL (ex. MongoDB).

Em resumo, os seguintes recursos deverão ser desenvolvidos:

- Implementar uma API REST
- Consumir os dados da API de forma assíncrona
- Cadastro de usuário
- Implementar o controle de acesso de acordo com usuários logados e não logados utilizando JWT (API e cliente)
- Permitir o usuário alterar seus dados e excluir a conta
- Listar os dados de pelo menos uma das coleções gerenciadas pela aplicação, permitindo paginação e ordenação pelos diferentes atributos
- Mostrar os detalhes de um registro de cada coleção
- Inserir novos registros para cada coleção com a devida validação
- Alterar um registro para cada coleção
- Excluir um registro de cada coleção
- Desenvolver um relacionamento (1-N ou N-M) entre estas coleções, tal que em um dos formulários de cadastros tenha que se escolher um ou muitos itens relacionado ao outro tipo
- Fazer a validação de entradas de dados no cliente e no servidor
- Fazer tratamento de erros no cliente e no servidor
- Atualizar o arquivo de configuração com os detalhes de acesso ao banco de dados
- Implementar uma rota que realiza uma carga automática dos dados, tal que, cada coleção tenha ao menos 5 registros prévios
- Desenvolver uma rota que apresente um relatório no formato de gráfico e que pode ser exportado em PDF