



Projeto Final
Prof.: Phyllipe Lima

Instruções Gerais:

- Atividade em dupla/trio. Não é permitido ser feita de forma individual.
- Haverá uma tarefa do SIGAA para entrar link de um git público com o código e slides de apresentação

Objetivo do Trabalho Final: Consolidar todos os fundamentos vistos em sala de aula na criação de uma aplicação web, e expandir conhecimentos de frameworks específicos escolhidos pela equipe. Podemos dividir o projeto desenvolvido em três blocos.

1 - Frontend

- HTML para definir os elementos
- CSS para estilizar os elementos
- JavaScript para adicionar comportamento aos elementos

2 – Backend

- NodeJS com Express para criar rotas (*endpoints*) e trabalhar requisições do tipo POST/GET
- NPM para gerenciar as dependências
- Manipulação de arquivos com NodeJS

3 – FullStack

- Utilização do Forms HTML para submeter dados para o servidor (NodeJS) com POST/GET
- Inserção em arquivos de dados do Forms (simulando Criação – **Create**)
- Leitura de arquivos a partir de elementos de busca do Forms (simulando Leitura - **Read**)

Para o trabalho final é necessário que o grupo aprofunde nesses temas para criar a sua aplicação.

Critérios e Requisitos Técnicos:

1. Ser uma aplicação web, isto é, precisa executar no navegador
2. Tela de Login e Cadastro de Usuário(a)
 - a. A Tela de Login deverá ter campo Usuário/Senha
 - b. Cadastro deverá ter Usuário/Senha/Confirmação Senha/Email
 - c. Tela de Login/Cadastro deverá ter as seguintes validações
 - i. Campo vazio => Todos os campos
 - ii. Checar email válido => Campo e-mail
 - iii. Chegar Senha/Confirmação Senha com no mínimo 4 dígitos => Campo Senha
 - iv. Idealmente utilizar alguma biblioteca para auxiliar o processo
 1. Yup como exemplo
3. Rotas Privadas
 - a. A aplicação não poderá ser acessada caso o login não tenha sucesso, de forma que se o(a) usuário(a) tentar acessar diretamente “forçando” a rota, haverá um erro.
 - b. Para isso é necessário utilizar algum sistema de Autenticação/Autorização com Tokens de validação.
 - i. Exemplo JSON Web Tokens
4. Utilizar framework *frontend* da escolha do grupo
 - a. Em sala de aula é apresentado uma introdução ao React, mas os grupos estão livres para utilizar/estudar/aprender qualquer outro e precisam aprofundar.
 - b. Exemplos: Vue, Angular.
5. Utilizar framework *backend* da escolha do grupo
 - a. Em sala de aula apresentei o NodeJS com Express, mas os grupos estão livres para utilizar/estudar/aprender qualquer outro em qualquer linguagem
 - b. Exemplos: SpringBoot, ASP.NET, Flask, outros.
6. Comunicação front-back através de verbos HTTP
 - a. Em sala de aula fizemos a comunicação com HTTP utilizando verbos GET e POST
 - b. No trabalho será necessário utilizar PUT e DELETE também.
7. Utilizar arquivo ou banco de dados para implementar um CRUD completo, isto é:
 - a. Criar, Ler, Atualizar, e Deletar => **C**reate **R**ead **U**pdate **D**elete
 - b. Em sala de aula fizemos leitura/escrita em arquivos utilizando NodeJS, mas os grupos devem seguir de acordo com a ferramenta de *backend* escolhida
 - c. Reafirmando que não é necessário utilizar banco de dados, sendo o uso de arquivos suficiente (como visto em aula).
8. A aplicação deverá permitir que o(a) usuário(a) atualize os dados, como trocar a senha e informar um novo e-mail.
9. A aplicação deverá certificar que o e-mail cadastrado é único, e não existe outro usuário na base com o mesmo e-mail.
10. A aplicação deverá conter as seguintes funcionalidades mínimas
 - a. Cadastrar algum recurso
 - i. Exemplo 1: Cadastrar avaliação de um jogo
 - ii. Exemplo 2: Cadastrar raça de um cachorro com a sua descrição e uma imagem
 - iii. Exemplo 3: Cadastrar benchmarks de placa de vídeo
 - b. Atualizar algum recurso (como os dos exemplos acima)
 - c. Remover Algum Recurso (como os do exemplo acima)
 - d. Ler algum recurso (como os do exemplo acima)

11. Versionamento com git e documentação

- a. O projeto deverá ficar disponível em algum repositório público git.
- b. Exemplos de hospedagem: GitHub, BitBucket, GitLab.
- c. Documentação com README
 - i. Descrição do projeto, apresentando o negócio e as tecnologias utilizadas
 - ii. Descrever o problema que o projeto tenta resolver.
 - iii. Descrever por que é um problema importante.

Nota: O projeto compõe 80% da N2

Avaliação Individual

Distribuição da Nota do Projeto:

1 – Apresentação 15% da nota

2 – Duas perguntas para cada membro, de teor técnico, que irá envolver conhecer o código e fazer eventuais mudanças pedidas durante a avaliação. Cada pergunta pode validar 40% da nota. As perguntas podem estar “Corretas” “Incorretas” ou “Meio-Correta”

3 – Entrega Parcial (01/11/2023) 5% da nota

Nota Projeto = Apresentação*0,15 + Entrega Parcial*0,05 + Pergunta1*0,4 + Pergunta2*0,4

Entrega Parcial – Até 05/11/2023

Entregar Via Tarefa do SIGAA um documento com os dados completos das equipes (nome e matrícula, e-mail) e quais frameworks front-end/back-end desejam utilizar.