Ministério da Educação
Universidade Federal de Itajubá – *Itajubá*

XDES03– Programação Web

Projeto Final
Prof.: Phyllipe Lima Francisco

Instruções Gerais:

- Atividade em grupos com 4 integrantes ou trio. Não é permitido ser feita de forma individual ou dupla.
- Será ajustado de forma a obtermos no máximo 12 equipes.
- 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 – **C**reate)
- Leitura de arquivos a partir de elementos de busca do Forms (simulando Leitura - **R**ead)

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, NextJS.
5. Utilizar framework *backend* da escolha do grupo
    a. Em sala de aula vimos 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 => Create Read Update Delete
    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 certificar que o e-mail cadastrado é único, e não existe outro usuário na base com o mesmo e-mail.
9. 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)

10. 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.

11. O projeto deverá fazer um uso de alguma API externa. Em sala de aula vimos uma API pública do Star Wars (https://swapi.dev/). Os grupos estão livres para escolherem qualquer API.

Nota: O projeto compõe 80% da N2
Avaliação Individual

Distribuição da Nota do Projeto:

1. **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 50% da nota. As perguntas podem estar "Corretas" "Incorretas" ou "Meio-Correta".**

2. **As perguntas não serão restritas, necessariamente, as responsabilidades de cada um no projeto. Todo integrante deve estar apto a responder qualquer pergunta do sistema todo.**

Apresentação não valerá ponto, mas será obrigatória.

Nota Projeto = (Pergunta1*0,5 + Pergunta2*0,5)*(Critérios)[0-1]