

Relatório A3 - Usabilidade, Desenvolvimento Web , Mobile e Jogos

Integrantes:

- Daniela Fernandes Santos de Britto - 1272215981
- Elias Neves Conceição - 12722127604
- Marcelo Victor Copis Sena - 12723110264
- Marlon Barreto Damasceno - 12722114193
- Rafael Almeida Rocha - 1272217730

Apresentando o projeto **Motion Games** da matéria de **Usabilidade, Desenvolvimento Web , Mobile e Jogos**, referente ao semestre 2023.2, onde seu objetivo é criar um aplicação que simule um site de jogos onde podemos cadastrar jogos de diversas plataformas (Steam,EpicGames,Battlenet,etc).

Requerimentos de software necessários para execução da aplicação:

- Node.js instalado (versão LTS);
- MySQL 8 (Workbench, Server, Router e Shell);

Observações:

1. Para utilizar o banco de dados é necessário importar o arquivo 'motiongames.sql' que consta na pasta *src* do repositório;
2. Se certificar-se que o serviço 'MySQL 80' está em execução;
3. Para fins de compatibilidade e prevenção de erros na conexão do servidor, executar o seguinte comando dentro do MySQL Command Line: "ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'admin';".

Instruções para execução da aplicação

1. Abra o Prompt de Comando dentro da pasta *src* do repositório;
2. Dentro dele execute o comando **http-server**. Caso ele mostre a mensagem "Available on" e o número da porta quer dizer que ele está funcionando.;

3. Abra outro Prompt de Comando, desta vez dentro da pasta *api* do repositório;
4. Dentro dele execute o comando **npm run dev**. Caso apareçam as mensagens "Servidor rodando!" e "Conectado ao Banco de Dados!" quer dizer que a API já está em funcionamento;
5. Para acessar as telas basta abrir o navegador (recomenda-se o Google Chrome) e digitar: `http://localhost:8080/inicio.html`

OBS: a porta ("8080" no caso acima) pode variar de acordo com a máquina;

Descrição do Front-end e Back-end:

No front-end, isto é, as telas visualizadas pelo, utilizamos os recursos HTML, CSS e JavaScript para a criação do nosso projeto.

No site apresenta-se uma tela de início com opção de **Login** e **Cadastrar**. Em relação ao que constava na primeira entrega, aqui praticamente nada mudou. Na tela de registro nada foi alterado pois achamos os dados solicitados para realizar o cadastro do usuário suficientemente satisfatórios. Já na tela de login, optamos por utilizar o e-mail e a senha como dados de entrada (antes usávamos o usuário e a senha).

Entrando no site teremos a aba principal **Home**. Aqui também nada foi alterado, e nela temos em exibição alguns dos jogos pré-registrados pelo usuário junto com a descrição do jogo e a nota dada pelos usuários.

Temos também outra aba chamada **Jogos**, onde terá as categorias de games (Terror, Ação, Guerra, etc) onde em cada categoria terá os games registrados onde podemos dar a nota para o jogo de 1 a 5 estrelas, pode-se observar também se o jogo já foi completo, incompleto ou jogando. Mais uma tela onde nada foi alterado

Por fim, inserimos a aba **Criar Jogo**, o local onde iria-se realizar todo o gerenciamento dos jogos da plataforma.

Já no back-end resolvemos utilizar como banco de dados relacional o MySQL 8, pela maior facilidade na manipulação de tabelas e códigos dentro do banco. Optamos por utilizar dentro da nossa API em Node.js a primeira versão do módulo referente ao MySQL, pela documentação ser mais clara de compreender e estabilidade do módulo. Utilizamos também alguns outros módulos, tais como: express, body-parser, cors, nodemon e bcrypt.

Tendo isso em mãos, conseguimos integrar a tela de cadastro e login com nosso servidor, tendo como diferencial nesta etapa a utilização do módulo bcrypt, que transforma a senha cadastrada pelo usuário em hash, enviando para armazenamento do banco uma versão criptografada da senha.

Já quando o login vai ser efetuado, ele compara a senha digitada pelo usuário com a senha armazenada no banco de dados. Novamente entra em ação o `bcrypt`, que possui um método que cuida de toda a comparação de forma segura e confiável.