FuturoDEV [Eco]

Módulo 1 - Projeto de Recuperação

**SUMÁRIO**

[1 INTRODUÇÃO 1](#_p9e2s15k061r)

[2 ENTREGA 2](#_liz931fdtw82)

[3 REQUISITOS DA APLICAÇÃO 2](#_dg7uhywn42wb)

[4 ROTEIRO DA APLICAÇÃO 3](#_xeebetpgp6a)

[4.1 PÁGINAS 3](#_1hsaagur518j)

[4.2 DOCUMENTANÇÃO NO README.MD 4](#_bjmvgfnlqemk)

[4.3 GRAVAÇÃO DE VÍDEO 4](#_fwr5zlyjl1ub)

[5 CRITÉRIOS DE AVALIAÇÃO 5](#_o2xebzow95bl)

[6 PLANO DE PROJETO 8](#_r5h1whaoo55l)

# 1 INTRODUÇÃO

O **Aqua365** é uma plataforma que promove a conscientização e a gestão do uso da água. Através da plataforma, os usuários podem monitorar o consumo de água em suas residências, acessar dicas de economia, e reportar problemas relacionados ao desperdício ou má gestão da água. Os gestores do **Aqua365** convidaram você a criar o MVP (*Minimum Viable Product*) da aplicação **Front-End** do software, que deverá ser construída utilizando a biblioteca **React**.

Está animado para iniciar o desenvolvimento?

Leia **atentamente** os itens abaixo para ficar por dentro das regras de negócio e rotas/telas que devem ser criadas na aplicação. Lembre-se também de **ler atentamente as regras** de entrega do projeto.

# 2 ENTREGA

O código deverá ser inserido e versionado no **GitHub** em modo privado, e o vídeo deverá ser inserido no **Google Drive** em modo leitor para qualquer pessoa com o link. Ambos os links deverão ser disponibilizados na tarefa **Módulo 1 - Projeto de Recuperação**, presente na semana 14 do AVA até o dia **01/07/2024** às **22h**.

O repositório GitHub deverá ser **privado**, com as seguintes pessoas adicionadas:

* Bruno Costa Figueiredo - **Bruno-Costa-fig**
* Operação LAB 365 - **lab365-operacao**

Não serão aceitos projetos submetidos **após a data limite da atividade**, e, ou alterados depois de entregues.

**Importante**:

1. Não modifique o seu projeto antes de receber a nota e feedback. Caso contrário, o mesmo não será corrigido.
2. Não esqueça de **submeter os links no AVA**. Não serão aceitos projetos em que os links não tenham sido submetidos.
3. Se for utilizar o Trello, crie um quadro Trello **diferente do quadro de exercícios**. No quadro de exercícios, os cartões diferentes serão deletados.

# 3 REQUISITOS DA APLICAÇÃO

A aplicação que deverá ser realizada **individualmente** deve contemplar os seguintes requisitos:

1. O sistema deverá ser desenvolvido utilizando a biblioteca **React**.
2. O sistema deve seguir o **Roteiro da Aplicação**.
3. Você deverá modelar o layout da aplicação com os formatos, tipografias, cores e organização de layout que achar melhor.
   1. Caso julgue necessário, poderá usar a biblioteca *Material UI* para estilização do layout ou outra que já possua domínio de como utilizar.
4. Você deverá utilizar o GitHub como versionador de código.
5. Você deverá gravar um vídeo de apresentação do sistema.
6. **Ponto extra:** Você poderá trabalhar com a usabilidade da aplicação.
   1. Nesse caso, deverá implementar **responsividade** para adaptação em telas grandes (como monitores) e telas pequenas (como smartphones e tablets).
7. **Ponto extra:** Você poderá utilizar em seu projeto a biblioteca **Chart.js** para criar gráficos de consumo conforme descrito no **Roteiro da Aplicação**.

# 4 ROTEIRO DA APLICAÇÃO

A aplicação deverá conter os seguintes pontos:

**Regras:**

* O número de CPF sempre deverá ser único, não podendo cadastrar mais de uma pessoa com o mesmo CPF.
* Para endereços, deve ser utilizado a API do ViaCEP, disponível em <https://viacep.com.br/>
* Um usuário não poderá ser deletado se existir um ou mais relatórios de consumo vinculados a ele.

**Usuários:**

1. O sistema, quando iniciado, irá carregar do localStorage ou json-server uma lista de usuários já cadastrados que usam a plataforma (criar uma lista com no mínimo 5 usuários, use o 4Devs - Ferramentas Online Grátis para gerar os usuários).
2. Na tela de login terá o botão de login e o de cadastrar, que quando clicado permite se cadastrar como novo usuário. Os usuários precisam fornecer:
   1. Nome
   2. Sexo
   3. CPF
   4. Data de Nascimento
   5. E-mail
   6. Senha
   7. Endereço (usar ViaCEP)

**Relatórios de Consumo:**

1. Cada usuário poderá cadastrar um ou mais relatórios de consumo, fornecendo:
   1. Mês/Ano do relatório
   2. Volume de água consumido em metros cúbicos
   3. Identificador do Usuário
   4. Endereço (usar ViaCEP)
   5. Descrição opcional

**Menu:**

1. Menu de navegação entre as telas e botão de sair.

## 4.1 PÁGINAS

A aplicação deverá conter as seguintes páginas:

**Login:**

* Cadastro e entrada no sistema.

**Dashboard:**

* Página inicial, exibindo uma visão geral dos relatórios de consumo cadastrados.
* Uso de *cards* para mostrar a quantidade de usuários ativos, relatórios cadastrados e, se for usar gráficos, colocá-los nesta página.
* Listagem dos relatórios de consumo em forma de lista ou de *cards* sem as opções de editar e excluir.

**Cadastro de Relatórios de Consumo:**

* Tela para cadastrar e editar relatórios de consumo.

**Listagem de Relatórios de Consumo:**

* Tela para listar todos os relatórios, com botões para acessar, editar (vai para a tela de cadastro preenchida) e deletar um relatório.

## 4.2 DOCUMENTANÇÃO NO README.MD

Crie um arquivo readme.md no repositório do seu projeto no GitHub, para documentar a sua solução, bem como demonstrar as técnicas e linguagens utilizadas, além do escopo do projeto e como o usuário pode executar o seu sistema.

Algumas dicas interessantes para utilizar na criação do seu portfólio são:

* Criar um nome para o seu software;
* Descrever qual o problema ele resolve;
* Descrever quais técnicas e tecnologias utilizadas. Aqui você também pode inserir alguma imagem ou diagrama para melhor entendimento;
* Descrever como executar;
* Descrever quais melhorias podem ser aplicadas;
* Entre outras coisas.

## 4.3 GRAVAÇÃO DE VÍDEO

Além do desenvolvimento deste sistema você deverá gravar um vídeo, com tempo máximo de **5 minutos**, abordando os seguintes questionamentos:

* Qual o objetivo do sistema? E demonstração de funcionamento.
* O que deve ser realizado para executar o sistema?
* Como você organizou as tarefas antes de começar a desenvolver?
* Quais as branches você criou e quais os objetivos para cada uma?
* Você acha que faltou algo no seu código que você poderia melhorar?

Você poderá gravar na vertical ou na horizontal. É importante que apareça seu rosto e esteja em um local com boa iluminação. Para realizar a entrega do vídeo, coloque em uma pasta do **Google Drive** em modo leitor para qualquer pessoa com o link, e compartilhe o mesmo na submissão do projeto no AVA. Uma dica interessante é você inserir o vídeo no readme.md do seu projeto no repositório do GitHub.

# 5 CRITÉRIOS DE AVALIAÇÃO

A tabela abaixo apresenta os critérios que serão avaliados durante a correção do projeto. O mesmo possui variação de nota de 0 (zero) a 10 (dez) como nota mínima e máxima, e **substituirá a nota** do projeto do módulo.

Serão **desconsiderados e atribuída a nota 0 (zero)** os projetos que apresentarem plágio de soluções encontradas na internet ou de outros colegas. Lembre-se: Você está livre para utilizar outras soluções como base, mas **não é permitida** a cópia.

| **Apresentação da Solução** | | | | |
| --- | --- | --- | --- | --- |

| **Nº** | **Critério de Avaliação** | **0** | **1,50** | |
| --- | --- | --- | --- | --- |
| 1 | Realizou a gravação de um vídeo? | Não foi realizada a gravação do vídeo. | Gravou o vídeo e abordou todos os tópicos listados no item 4.3. | |

| **Uso adequado do GitHub** | | | | |
| --- | --- | --- | --- | --- |

| **Nº** | **Critério de Avaliação** | **0** | **0,25** | **0,50** |
| --- | --- | --- | --- | --- |
| 2 | As mensagens dos commits estão claras e adequadas? | Os commits do aluno não tiveram nenhuma mensagem | Os alunos fizeram commits com comentários, mas esses estavam inadequados ou incompletos com relação ao conteúdo das alterações. | As mensagens do commit estão claras e adequadas em relação ao conteúdo das alterações. |
| 3 | Foi aberto um branch separado para cada etapa, funcionalidade? | Os commits foram realizados diretamente na master/main. | Foram abertas branchs para cada funcionalidade, porém foram realizados commits de mesma funcionalidade em branchs diferentes. | Foram abertas branchs para cada funcionalidades, e os commits foram realizados de forma concisa nas mesmas. |
| 4 | Criou uma documentação com readme.md? | Não criou a documentação. | Criou uma documentação, porém de forma muito simplificada. | Criou uma documentação completa com ao menos todos os tópicos sugeridos no item 4.2 |

| **Desenvolvimento adequado da Aplicação** | | | | |
| --- | --- | --- | --- | --- |

| **Nº** | **Critério de Avaliação** | **0** | **0,50 a 0,75** | **1,00** |
| --- | --- | --- | --- | --- |
| 5 | Criou a página de cadastro de usuários e Login? | Não criou a página. | Criou a página mas a função de cadastrar um novo usuário e/ou Login não funciona. | Criou a página e a função de cadastrar um usuário e realizar login corretamente. |
| 6 | Utilizou Context API para gerenciar os dados globais da aplicação? | Não utilizou Context para gerenciar os dados globais. | Utilizou a Context API, mas de maneira superficial. | Utilizou a Context API de maneira correta. |
| 7 | Utilizou os inputs de formulário de cadastro de maneira correta? | Usou somente inputs do tipo ‘text’ | Usou somente alguns inputs de maneira correta. | Usou todos os inputs de maneira correta e validou a obrigatoriedade dos campos. |
| 8 | Criou as páginas de Dashboard e Listagem de Relatórios de Consumo? | Não criou a página. | Criou apenas uma das páginas, e as informações estão imprecisas. | Criou ambas as páginas funcionais com as informações corretas. |
| 9 | Utilizou localStorage ou JSON Server de maneira correta? | Não utilizou nenhuma solução. | Apenas um dos cadastros está integrado com a localStorage ou JSON Server. | Utilizou a localStorage ou JSON Server de maneira correta. |
| 10 | Criou a página para cadastrar e/ou editar um relatório de consumo? | Não criou a página. | Criou a página, mas as funções de cadastrar e/ou editar não funcionam. | Criou a página e as funções de cadastro e/ou edição corretamente. |
| 11 | Usou a API do ViaCEP? | Não usou a API. | A função foi criada, porém a consulta executa com erro e/ou não são usados os dados recebidos da API. | Criou a função de consultar a API corretamente e usou os dados obtidos. |

| **Ponto Extra** | | | | |
| --- | --- | --- | --- | --- |

| **Nº** | **Critério de Avaliação** | **0** | **0,50** | |
| --- | --- | --- | --- | --- |
| E1 | Implementou responsividade para diferentes tipos de tela? | Não foi implementado responsividade para diferentes tipos de tela. Ou foi implementado em apenas algumas partes do sistema. | Todas as páginas do sistema receberam a implementação de responsividade para diferentes tamanhos de tela. | |
| E2 | Implementou o uso de gráficos no lugar da lista de relatórios de consumo? | Não foi implementado o uso de gráficos na aplicação, ou a implementação não funciona adequadamente. | Foi implementado o uso de gráficos, e o mesmo está funcionando adequadamente. | |

Importante lembrar que a nota de projeto é saturada em 10,00 e caso o aluno tira nota máxima no projeto e implemente os pontos E1 e E2, ficará com nota 10,00 e não 11,00.

# 6 PLANO DE PROJETO

Ao construir a aplicação proposta, o aluno estará colocando em prática os aprendizados em:

* **Versionamento**: Uso do GitHub para versionamento de código.
* **HTML**: Tags e Elementos semânticos
* **CSS**: Seletores, Principais estilos, Layouts e Flexbox
* **JavaScript**: Variáveis, Tipos de dados, Operadores, Manipulação do DOM, Estrutura de Controle de Fluxo, Funções, Eventos, JSON, LocalStorage, Interval, Timeout, Operadores Rest e Spread, Módulos, Arrow Functions, Funções de Arrays, Funções Assíncronas e Fetch.
* **React**: Renderização de componentes, Props, Proptypes, Hooks, Eventos, Renderização de listas, React Router, Formulários, Prop Drilling, Composition, Estilos (Material UI), Developer Tools e Deploy.
* **Skills**: Organização, criação de documentação e apresentação de solução.