

Projeto Avaliativo 2 - Sistema de Gestão Integrada

Peso: 50% da nota do módulo 1

DEVinHouse

Sumário

1. Introdução	1
2. Requisitos da Aplicação	1
3. Exemplo de aplicação	2
4. Entrega	3
5. Critérios de Avaliação	3
6. Plano de Projeto	5

1. Introdução

Você está prestes a entrar para o time de Desenvolvedores da **DEVin Corp**. Para concretizar a sua contratação, você deverá resolver um desafio utilizando React. O time de recrutamento necessita que você crie uma aplicação protótipo para o gerenciamento de empresas e produtos, chamada **Sistema de gestão integrada - SGI**.

2. Requisitos da Aplicação

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

- Uma página de login contendo um campo de email e senha. No evento de onSubmit, validar se os campos foram preenchidos: E-Mail (obrigatório) e Senha (obrigatório). Ao validar os campos de input, o usuário é redirecionado para a tela de Mapa;
- Uma página de cadastro de nova empresa, contendo os campos: razão social, nome fantasia, cnpj, e-mail, cep, endereço, número, bairro, cidade, complemento, latitude e longitude; Todos os campos são obrigatório, exceto o campo complemento (validar no método onSubmit). Ao cadastrar uma nova empresa, mostrar uma mensagem de feedback de **empresa cadastrada com sucesso** (tratar o método onSubmit com Try/Catch).
- Uma página de cadastro de produto, contendo os campos url do produto, nome do produto, preço unitário, descrição (<textarea/>) , fornecedor do produto (<select />) e grupo (<select />) . Ao cadastrar um novo produto, mostrar uma mensagem de feedback

de **produto cadastrado com sucesso** (tratar o método onSubmit com Try/Catch). As opções do **select de grupo** e **fornecedor** devem ser preenchidas de acordo com as rotas /categorias e /fornecedores

- Uma tela de mapa, contendo um Mapa centralizado e com marcadores de todas as empresas cadastradas no sistema (usar a latitude e longitude da empresa para definir a localização no mapa).
- Uma implementação extra por parte do aluno: Ex: uma tela de cadastro de fornecedores, uma tela de listagens de produtos, etc

3. Exemplo de aplicação

A aplicação deverá conter os requisitos apresentados anteriormente, sendo codificada utilizando a Biblioteca React.

As imagens a seguir apresentam um exemplo de como a aplicação pode ser construída.

Login

Email

Email é obrigatório

Senha

Senha é obrigatório

Sign in

Legenda: Tela de login

Nova empresa

Cancelar

Salvar

Razão social

Supermercado Paripassu Involves Ltda

Nome fantasia

Supermercado Paripasinho Involves

CNPJ

27.466.236/0001-74

Email

superparipassuinvolve@gmail.com

CEP

62870-000

Endereço

Rua Mario Coriolando Pontes

Número

8

Bairro

Buriti

Cidade

Pacajus

Complemento

Próximo a praça central

Latitude

-4.095229499999999

Longitude

-38.4550829

Legenda: Tela de cadastro de empresa

Novo produto

CancelarCadastrar

URL da imagem

Nome

Custo unitário

Descrição

Fornecedor

Grupo

Legenda: Tela de novo produto

Novo produto

Cancelar

Cadastrar



URL da imagem

https://static3.tcdn.com.br/img/img_prod/350075/muda_de_pitaya_branca_35cm_feita_de_estaca_76

Nome

Pitaya

Custo unitário

78,99

Descrição

Fruta rica em fibras, água, cálcio, ferro, fósforo e vitaminas B, C e E, além de ter poucas calorias, o que pode trazer diversos benefícios para o organismo.

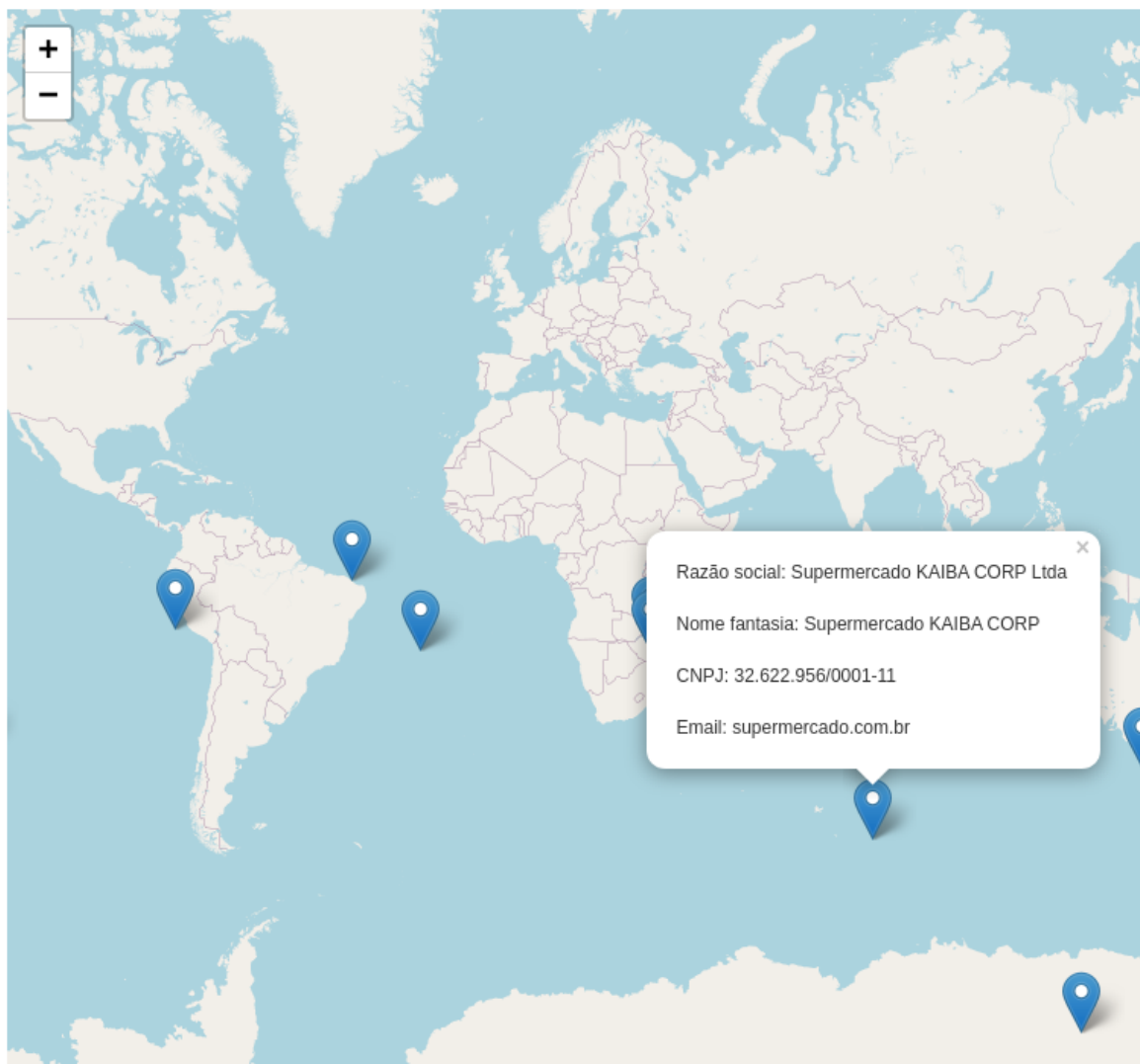
Fornecedor

Del valle

Grupo

Frutas

Legenda: Tela de novo produto



Legenda: Tela Mapa

4. Entrega

O código desenvolvido deverá ser submetido no GitHub, e o link deverá ser disponibilizado na tarefa [Submeter aqui a URL do projeto 2](#), presente na semana 12 do AVA até o dia 21/11/2021 às 23h55.

Entregas realizadas após a data limite sofrerão **decréscimo na nota** de avaliação, sendo considerado **80%** da nota para tarefas submetidas até o dia 28/11/2021 às 23h55 e **50%** para tarefas submetidas até o dia 05/12/2021 às 23h55. **Não serão** avaliados projetos submetidos após o dia 05/12/2021.

Importante: Será considerado como data de entrega a **última atualização** no repositório do projeto no GitHub. Lembre-se de não modificar o código até receber sua nota.

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 possui peso de 40% sobre a avaliação do módulo 1.

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.

Nº	Critério de Avaliação	0	0,5	1
1	O aluno desenvolveu um componente de menu?	O aluno não desenvolveu o componente de menu.	O aluno desenvolveu o componente de menu e exibiu conforme o mockup da aplicação, mas não integrou com o react-router-dom.	O aluno desenvolveu o componente de menu e exibiu conforme o mockup. As rotas foram desenvolvidas corretamente e importadas no lugar correto da aplicação.
2	O aluno criou uma página com layout responsivo e agradável?	O aluno não desenvolveu uma tela intuitiva e agradável.	O aluno conseguiu desenvolver telas responsivas parcialmente responsivas e demonstrou noção de cores e posicionamento básicos.	O aluno conseguiu desenvolver telas responsivas e com cores agradáveis e elementos bem posicionados.
3	O aluno desenvolveu um código bem organizado?	O código do projeto não estava organizado ou sem implementações.	O aluno desenvolveu parcialmente um código bem organizado. Foi encontrado componentes não componetizados, variáveis e funções com nomes não objetivos.	O aluno desenvolveu um código bem organizado com pastas bem definidas e nome de variáveis e funções claras , além da separação de componentes.
4	O aluno implementou um recurso além do solicitado?	O aluno não inovou no projeto.	---	O aluno inovou no projeto, aplicando novas funcionalidades e libs além do solicitado.
Nº	Critério de Avaliação	0	0,75	1

5	O aluno desenvolveu a tela de cadastro de empresa ?	O aluno não desenvolveu a tela de cadastro de empresa	O aluno inseriu os campos conforme o mockup, porém não implementou o cadastro via POST na rota /empresas	O aluno desenvolveu o formulário completo, adicionando todos os campos com seus devidos estados e implementou o evento de onSubmit conforme especificado.
6	O aluno desenvolveu a tela de login ?	O aluno não desenvolveu a tela de login	O aluno inseriu os campos , mas não validou o campo de email e senha e não redirecionou para a tela de Mapa	O aluno inseriu os campos de email e senha, validou se foram preenchidos e redirecionou para a tela de Mapa.
Nº	Critério de Avaliação	0	1	2
7	O aluno desenvolveu a tela de mapas?	O aluno não desenvolveu a página de mapas.	O aluno inseriu o mapa, mas não mostrou os marcadores de acordo com a API.	O aluno inseriu o mapa e implementou os marcadores com base nos dados salvos na API. Seguiu o mockup solicitado.
8	O aluno desenvolveu a tela de cadastro de produto ?	O aluno não desenvolveu a tela de cadastro de produto	O aluno inseriu os campos conforme o mockup, porém não implementou o cadastro via POST na rota /produtos.	O aluno desenvolveu o formulário completo, adicionando todos os campos com seus devidos estados e implementou o evento de onSubmit conforme especificado.

6. Plano de Projeto

Ao construir a aplicação Sistema de gestão integrada - SGI contendo as páginas de Mapa, Empresas, Produtos e login, o aluno estará colocando em prática os aprendizados em:

- HTML: principais tags como head, title, body, div, h1, form, input, button, ul, li. Atributos de tags como class, id, type.
- CSS: estilizar a página, os botões, inputs, alterar atributos dos elementos da tela de acordo com a interação do usuário para uma melhor experiência do usuário (UX), Alinhamento de elementos com flex-box.
- Javascript: variáveis, arrays, funções .map(), .filter(), .reducer(), manipulação de eventos onChange, onSubmit, json-server.
- React: Componentes funcionais, Props, useEffect, useState, Estado do componente, Componentes de terceiros (react-leaflet), validação de formulários, chamadas para API usando API Fetch.