

DEVinHouse

Módulo 1 - Projeto Avaliativo

SUMÁRIO

1 INTRODUÇÃO	1
2 ENTREGA	2
3 REQUISITOS DA APLICAÇÃO	2
4 ROTEIRO DA APLICAÇÃO	3
4.1 FORMATO DO SISTEMA	3
4.2 EXEMPLOS DE APLICAÇÃO	6
4.3 DOCUMENTAÇÃO NO README.MD	11
4.4 GRAVAÇÃO DE VÍDEO	11
5 CRITÉRIOS DE AVALIAÇÃO	11
6 PLANO DE PROJETO	15

1 INTRODUÇÃO

A **DEVinPharmacy LTDA**, uma renomada empresa do ramo farmacêutico, está expandindo a sua rede de lojas. Por conta da expansão, o time de gestão necessita da criação de um sistema online, intitulado **Medication Management**, para gerenciamento de medicamentos. O seu perfil chamou a atenção dos gestores para a criação do protótipo **front-end** do sistema que deverá ser codificado com o uso da biblioteca **React**.

Está animado para iniciar o desenvolvimento?

Leia **atentamente** os itens abaixo para ficar por dentro das regras de negócio e páginas 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 Avaliativo**, presente na semana 11 do AVA até o dia **27/08/2023** às **23h55**.

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

- Pedro Henrique Barroso da Silva - **pedrohbsilva**
- Operação DEVinHouse - **devinhouse-operacao**

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

Importante:

1. Lembre-se de não modificar o código até receber sua nota. Caso contrário, perderá pontos durante a correção.
2. Nos links, caso faça o deploy, disponibilize no AVA também.
3. Não esqueça de submeter o link no AVA.

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.
 - a. Caso julgue necessário, poderá usar Bootstrap, MUI (Material UI) e/ou outra ferramenta para estilização do *layout*.
4. Você deverá se preocupar com a **usabilidade** da aplicação.
 - a. Pense no contraste das cores para não atrapalhar na visualização.
 - b. Implemente sempre que possível responsividade para adaptação em telas de computadores de diferentes tamanhos.
5. Você deverá **gravar um vídeo** de apresentação do sistema.
6. Dica: Crie um board no Trello para organizar as tarefas a serem realizadas no projeto. Lembre-se que o desenvolvimento ocorre de forma mais fluída quando já se tem o planejamento prévio daquilo que será codificado.

4 ROTEIRO DA APLICAÇÃO

A **DEVinPharmacy LTDA**, deseja gerenciar seus medicamentos, através de um sistema de gestão.

4.1 FORMATO DO SISTEMA

A aplicação deverá possuir as seguintes páginas:

1. Uma página de **login** contendo um campo de **email** e **senha**.
 - a. O evento onSubmit deverá validar se os campos foram preenchidos, sendo:
 - i. E-Mail (obrigatório)
 - ii. Senha (obrigatório)
 - b. Ao validar os campos de input, o usuário é redirecionado para a página de **Mapa**.
 - i. Não é necessário guardar as informações, apenas valide se o campo de e-mail recebeu um e-mail verdadeiro, e se o campo de senha possui 8 ou mais caracteres com números e letras.
2. Uma página de **cadastro de nova farmácia**, contendo os campos:
 - a. Razão social (obrigatório)
 - b. CNPJ (obrigatório)
 - c. Nome Fantasia (obrigatório)
 - d. E-mail (obrigatório)
 - e. Telefone (opcional)
 - f. Celular (obrigatório)
 - g. Endereço
 - i. CEP (obrigatório)
 - ii. Logradouro/Endereço (obrigatório)
 - iii. Número (obrigatório)
 - iv. Bairro (obrigatório)
 - v. Cidade (obrigatório)
 - vi. Estado (obrigatório)
 - vii. Complemento (opcional)
 - viii. Geolocalização
 - Latitude (obrigatório)
 - Longitude (obrigatório)

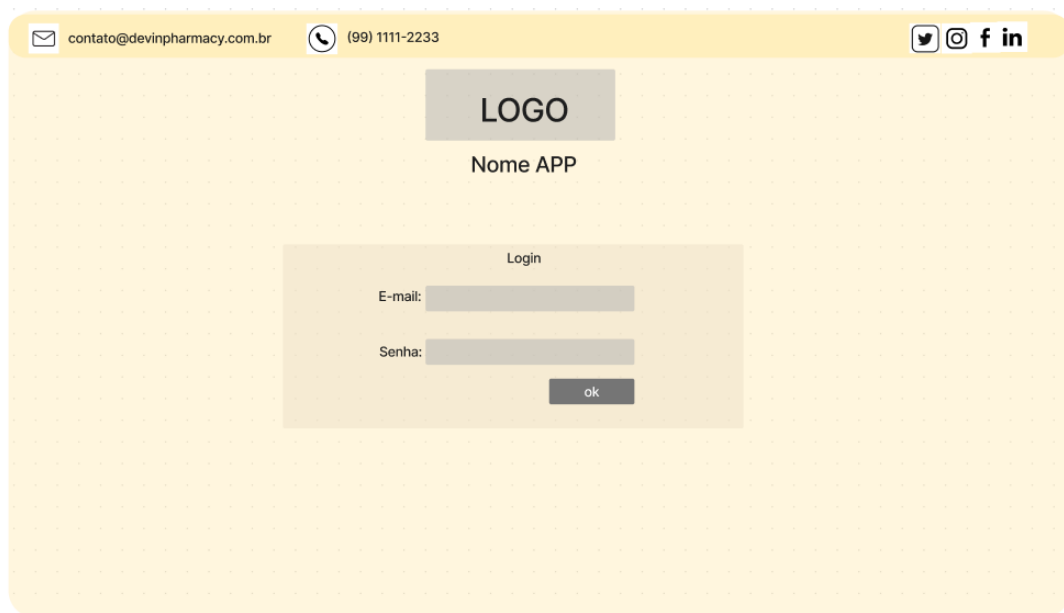
- h. Os dados de endereço devem ser pré-preenchidos através da consulta do **CEP** pela **API ViaCEP** do IBGE Brasil. Preencher com os campos disponíveis da API (logradouro, bairro, cidade e etc) e solicitar ao usuário os campos extras (número, complemento e etc)
 - i. Os campos obrigatórios e opcionais devem ser validados no método `onSubmit`.
 - j. Ao cadastrar uma nova farmácia, mostrar uma mensagem de feedback de empresa cadastrada com sucesso. Dica: Utilize o método `onSubmit` com `Try/Catch`.
 - k. Salve os dados em `localStorage` para simular uma API.
3. Uma página de **cadastro de medicamentos**, contendo os campos:
- a. Nome do medicamento (obrigatório)
 - b. Nome do laboratório (obrigatório)
 - c. Dosagem do medicamento (obrigatório)
 - d. Descrição do medicamento (opcional)
 - e. Preço unitário do medicamento (obrigatório)
 - f. Tipo do medicamento (obrigatório)
 - g. Realize o tratamento dos campos da seguinte forma:
 - i. Para a descrição, utilize `<textarea/>`
 - ii. Para o tipo, utilize `<select/>`
 - Medicamento controlado ou Medicamento comum
 - h. Ao cadastrar um novo medicamento, mostrar uma mensagem de feedback de produto cadastrado com sucesso. Dica: Utilize o método `onSubmit` com `Try/Catch`.
 - i. Salve os dados em `localStorage` para simular uma API.
4. Uma página de **mapa**, contendo um Mapa centralizado e com marcadores de todas as farmácias cadastradas no sistema.
- a. Utilize a **React Leaflet** para renderizar os mapas na tela
 - b. Utilize a latitude e longitude da farmácia para definir a localização no mapa
 - c. Ao clicar no marcador, apresentar uma listagem de todas as informações da farmácia que foram cadastradas, como Razão Social, CNPJ, Nome e etc.
 - i. Caso um item opcional não tenha sido cadastrado, a listagem não deverá mostrar o espaço destinado ao item.
 - ii. Dica: Você pode exibir o número de celular em formato de hiperlink que quando clicado direciona o usuário para a página de envio de mensagem do WhatsApp. Utilize **`https://wa.me/55DDNNNNNNNNNN`** sendo DD os dois dígitos do DDD e NNNNNNNNN os dígitos do número de celular.
5. Uma página de **listagem de medicamentos** onde todos os produtos cadastrados devem ser listados em formato de **card**.

- a. Na parte superior deve existir uma barra de busca para buscar por algum medicamento.
 - b. Dica: Você também pode implementar um botão de filtro para filtrar por preço, tipo e etc.
6. Uma página de **detalhar os medicamentos** que deverá aparecer quando um medicamento é clicado na página de listagem.
- a. Você pode criar uma página de detalhes, ou simplesmente um modal.
 - b. Todas as informações cadastradas devem ser exibidas para o usuário.
 - i. Na página ou modal poderá existir alguma imagem que represente uma caixa de medicamento, já que o Ministério da Saúde Brasileira proíbe a divulgação da foto do medicamento.
7. Uma implementação de **página com funcionalidade extra** criada por você. Exemplo:
- a. Página de cadastro de usuário para ser utilizada no login
 - b. Página de esqueceu sua senha
 - c. Página de FAQ
 - d. Entre outras funcionalidades
8. Um **menu de navegação** a ser utilizado nas rotas para as páginas da aplicação.
9. Criação de uma **estilização (identidade visual) do sistema**. Na sequência do documento você será apresentado a um wireframe da aplicação proposta, entretanto será ponto avaliativo a criação de sua própria organização e estilização de páginas. Utilize o wireframe apenas para entender o modelo a ser desenvolvido, e crie uma identidade visual única com cores e formatos que achar mais interessante e intuitivo.

4.2 EXEMPLOS DE APLICAÇÃO

A aplicação deverá conter os requisitos apresentados anteriormente, sendo codificada em **html**, **css** e **javascript** através da biblioteca **React**, também deverá ser utilizado **markdown** para o README.md.

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



contato@devinpharmacy.com.br (99) 1111-2233

LOGO

Nome APP

Login

E-mail:

Senha:

ok

Imagem 1: Página de login.



LOGO Nome APP

Home

Farmácias

Medicamentos

Troca

Acompanhar troca

Cadastro de Farmácia

Razão Social CNPJ Nome Fantasia

E-mail Telefone Celular

CEP Logradouro Número

Bairro Cidade Estado

Complemento Latitude Longitude

LIMPAR SALVAR

Imagem 2: Página de cadastro de farmácia.

LOGO

Nome APP

Home

Farmácias

Medicamentos

Troca

Acompanhar troca

Cadastro de Medicamento

Medicamento

Laboratório

Dosagem

Tipo

Preço Unitário

Descrição

LIMPAR

SALVAR

Imagem 3: Página de cadastro de medicamentos.

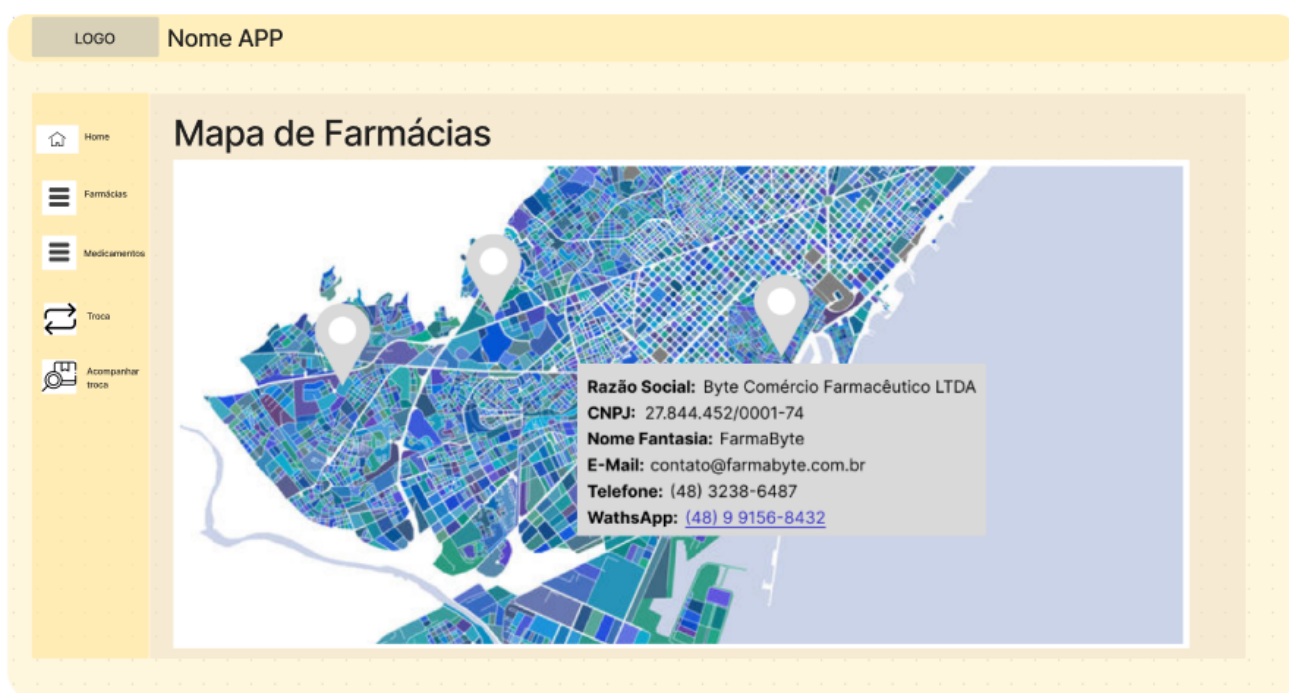


Imagem 4: Página de listagem de farmácias com mapa.

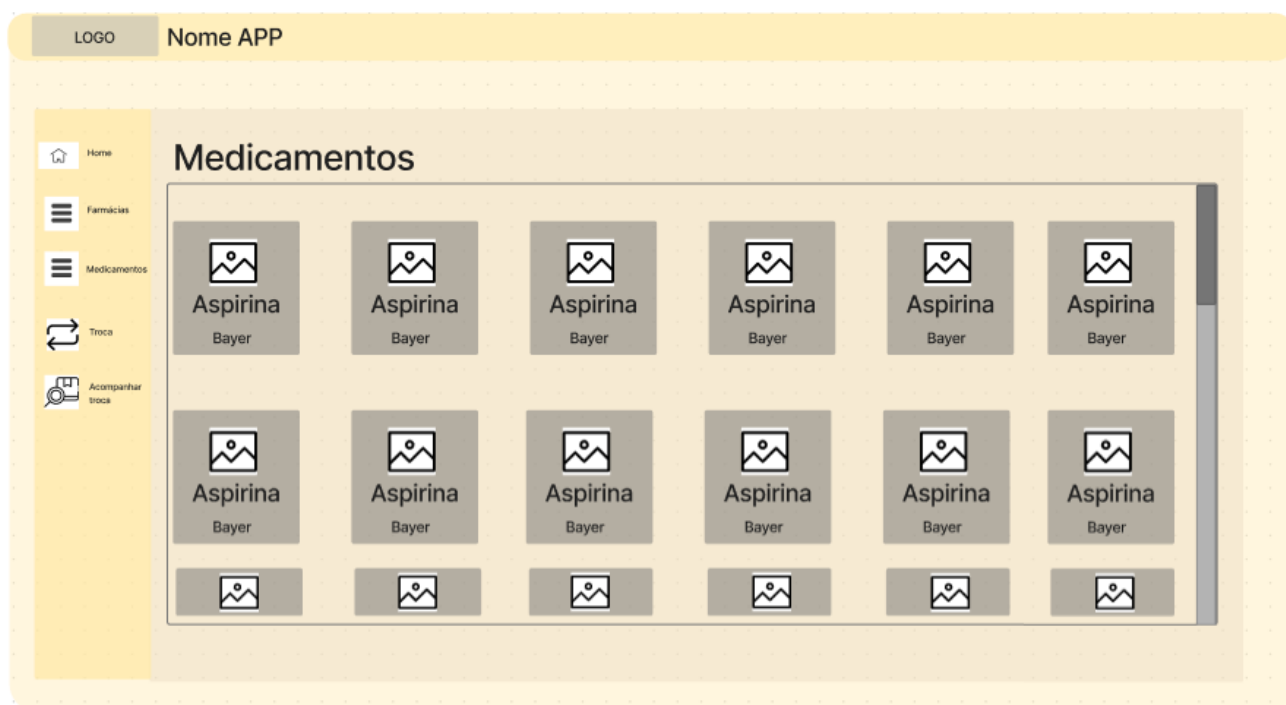


Imagem 5: Página de listagem de medicamentos.

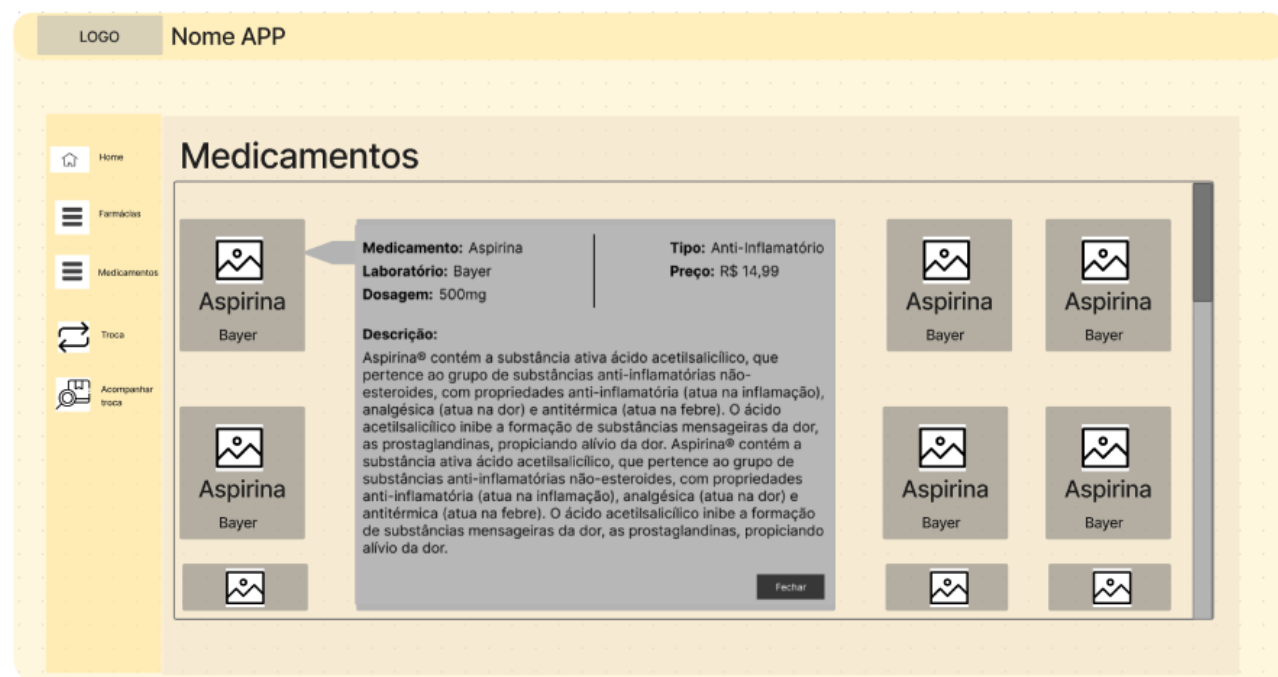


Imagem 6: Página de detalhes de medicamentos.

4.3 DOCUMENTAÇÃO 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.

4.4 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 possui peso de **60% sobre a avaliação 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	1,40
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 4.4.

Uso adequado do GitHub

Nº	Critério de Avaliação	0	0,25	0,70
2	Foi aberto um branch separado para cada etapa, com mensagens do commit claras e adequadas?	Os commits foram realizados diretamente na master/main, e não foi usado mensagens claras.	Foram abertas branches para cada funcionalidade, porém foram realizados commits de mesma funcionalidade em branches diferentes, ou mensagens inadequadas.	Foram abertas branches para cada funcionalidades, e os commits foram realizados de forma concisa com mensagens adequadas.
3	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 4.3

Deploy da Aplicação

Nº	Critério de Avaliação	0	0,80
4	Realizou o deploy?	Não foi realizado o deploy da aplicação.	Foi realizado o deploy da aplicação.

Desenvolvimento adequado da Aplicação

Nº	Critério de Avaliação	0	0,25 a 0,50	0,80
5	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 wireframe da aplicação, mas não integrou com o react-router-dom.	O aluno desenvolveu o componente de menu e exibiu conforme o wireframe. As rotas foram desenvolvidas corretamente e importadas no lugar correto da aplicação.
6	O aluno criou uma página com layout responsivo e agradável?	O aluno não desenvolveu uma página intuitiva e agradável.	O aluno conseguiu desenvolver páginas responsivas parcialmente responsivas e demonstrou noção de cores e posicionamento básicos.	O aluno conseguiu desenvolver páginas responsivas e com cores agradáveis e elementos bem posicionados.
7	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.
8	O aluno desenvolveu a página de login?	O aluno não desenvolveu a página de login.	O aluno inseriu os campos, mas não validou o campo de email e senha e não <u>redirecionou</u> para a página de Mapa.	O aluno inseriu os campos de email e senha, validou se foram preenchidos e redirecionou para a página do Mapa.
9	O aluno desenvolveu a página de cadastro de medicamentos?	O aluno não desenvolveu a página de cadastro de medicamento.	O aluno inseriu os campos conforme o mockup, porém não implementou o cadastro via LocalStorage	Desenvolveu o formulário completo, adicionando todos os campos com seus estados e fez o evento de onSubmit conforme especificado.

Nº	Critério de Avaliação	0	0,25	0,50	0,80
10	O aluno desenvolveu a página de listagem de medicamentos?	O aluno não desenvolveu a página de listagem de medicamentos.	O aluno desenvolveu a página de listagem de medicamentos, mas não implementou a barra de busca de medicamento.	O aluno implementou a listagem de medicamentos com a barra de busca de maneira funcional, mas não implementou uma forma de detalhar as informações do medicamento.	O aluno implementou a listagem de medicamentos, sua barra de busca e também o detalhamento das informações de cada medicamento.
11	O aluno desenvolveu a página de cadastro de farmácia?	O aluno não desenvolveu a página de cadastro de farmácia	O aluno inseriu os campos conforme o mockup, porém não implementou o cadastro via salvamento no localStorage	O aluno desenvolveu o formulário completo, adicionando todos os campos com seus devidos estados e implementou o evento de onSubmit conforme especificado, mas não implementou o consumo de dados do endereço através da API ViaCEP.	O aluno desenvolveu corretamente o formulário e consumiu os dados da API para preencher os campos do endereço.
12	O aluno desenvolveu a página de mapas?	O aluno não desenvolveu a página de mapas.	O aluno inseriu o mapa, mas não <u>mostrou os marcadores</u> de acordo com a API.	O aluno inseriu o mapa, <u>mostrou os marcadores</u> , porém sem informações da API.	O aluno inseriu o mapa e implementou os marcadores com base nos dados salvos.

Ponto Extra				
Nº	Critério Extra	0	0,10 à 0,50	1,00
E1	O aluno implementou um recurso além do solicitado?	O aluno não inovou no projeto.	O aluno trouxe algumas novidades, mas com falhas.	O aluno inovou no projeto, aplicando novas funcionalidades e libs funcionais além do solicitado.
E2	O aluno entregou todos os exercícios da semana 11 e 12 e está conforme solicitado?	O aluno não entregou	O aluno entregou parcialmente os exercícios e /ou entregou todos porém com falhas.	O aluno entregou todos os exercícios com sucesso e conforme o solicitado.

Observações importantes:

1. A nota do projeto possui variação de **zero (0)** a **dez (10)**. Caso o aluno tire 10,00 no projeto e alcance um ou mais ponto extra, a nota ficará fixada em 10,00.
2. O ponto extra só será atribuído se a funcionalidade for implementada de forma correta, **funcionando perfeitamente**.
3. Evite adicionar bibliotecas extras e não utilizar elas. Adicione somente as bibliotecas necessárias para o seu projeto.

6 PLANO DE PROJETO

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

- **Fundamentos da Programação Web:**
 - HTML: Principais elementos (versão 4) e Elementos semânticos (versão 5)
 - CSS: Seletores, Principais Estilos, Layouts e Layout com Flexbox
 - Organização: Trello e Versionamento com GitHub
- **JavaScript:**
 - Variáveis, Tipos de dados, Array, Estrutura de Controle de Fluxo (condicional e repetição) e Operadores (aritméticos, lógicos e relacionais)
 - Funções, Manipulação do DOM, Utilização de Seletores, Eventos, JSON, LocalStorage, Interval e Timeout
 - Apresentação do ES6+, Escopo (let, var e const), Classes, Atributos, Encapsulamento (closure) e Funções de Arrays (forEach, map, filter, find, reduce e every)
 - Arrow Functions e Módulos (export e import)
 - Funções Assíncronas (Promises, Async e Await), Operadores Rest e Spread, e Consumo de API com Fetch

- **React:**

- Introdução ao React, Primeiros passos, Preparação do ambiente, Componentes no React, Sintaxe JSX (tipos de retorno, renderização de listas e condicionais), Props e PropTypes, e Eventos.
- Componentes funcionais, Introdução à Hooks e Estado, Componentes controlados vs não controlados, Efeito colateral, Hooks Customizadas, Outras hooks úteis (useReducer, useContext e etc).
- Organização de projeto e Atomic Design, CSS Modules, Bibliotecas Externas, Bibliotecas de componentes do Material UI.
- Rotas com React Router Dom, Estado global e ContextAPI, Formulários com React Hook Form, Estilização com Styled Components, Técnicas avançadas, Build e Deploy de um projeto
- Componentes de terceiros (react-leaflet)