

DEVinHouse

Módulo 1 - Projeto Avaliativo 2

SUMÁRIO

1 INTRODUÇÃO	1
2 REQUISITOS DA APLICAÇÃO	1
3 EXEMPLO DE APLICAÇÃO	4
4 CRITÉRIOS DE AVALIAÇÃO	9
5 ENTREGA	15
6 PLANO DE PROJETO	15

1 INTRODUÇÃO

A **LAB Things Corporation**, uma renomada empresa do ramo IOT, está expandindo sua operação. Por conta da expansão, novos produtos inovadores estão sendo criados, dentre eles, um sistema inovador para interface de dispositivos conectados, intitulado **Connect Lab**. O seu perfil chamou a atenção dos gestores para a criação do protótipo **front-end** do sistema que deverá ser codificado em **React**. Mãos à obra!

2 REQUISITOS DA APLICAÇÃO

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

1. Uma página de **cadastro de usuário**, contendo os campos:
 - a. Nome Completo (obrigatório)
 - b. URL de uma foto (opcional)
 - c. E-Mail (obrigatório)
 - d. Senha (obrigatório)
 - e. Validação de Senha (obrigatório)
 - f. Telefone (opcional)
 - g. Endereço
 - i. CEP (obrigatório)
 - ii. Logradouro/Endereço (obrigatório)

- iii. Número (obrigatório)
 - iv. Bairro (obrigatório)
 - v. Cidade (obrigatório)
 - vi. Estado (obrigatório)
 - vii. Complemento (opcional)
 - h. Os dados de endereço devem ser pré-preenchidos através da consulta do **CEP** pela **API [ViaCEP](#)** do IBGE Brasil.
 - i. Os campos obrigatórios e opcionais devem ser validados no método onSubmit.
 - j. Ao cadastrar o usuário, mostrar uma mensagem de feedback de usuário cadastrado com sucesso. Dica: Utilize o método onSubmit com Try/Catch.
 - k. Salve os dados utilizando a api disponibilizada API.
2. 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 **Início**.
 - i. Valide se o campo de e-mail recebeu um e-mail verdadeiro, e se o campo de senha possui 8 ou mais caracteres com letras, números e caracteres especiais.
3. Um **menu de navegação** a ser utilizado nas rotas para as páginas de:
- i. Início
 - ii. Adição de Dispositivos
 - iii. Perfil do usuário
4. Uma página de **Início**, contendo:
- a. Um componente de [previsão do tempo](#) mostrando temperatura, estado (ensolarado, nublado, chuvoso etc), umidade e demais informações que você achar relevante. Utilize os dados de endereço do usuário para trazer as informações climáticas.
 - b. Lista de dispositivos adicionados ao local (casa, escritório, fábrica etc) do usuário.
 - i. Adicione filtros para os agrupamentos que o usuário possui cadastrado.
Ex:
 - 1. Todos os dispositivos
 - 2. Quarto
 - 3. Sala

4. Cozinha
 5. Etc.
5. Uma forma de **detalhar os dispositivos** que deverá aparecer quando um dispositivo é clicado na página de listagem.
 - a. Você deve criar uma página de detalhes.
 - b. Todas as informações do dispositivo adicionado ao perfil de usuário devem ser exibidas, por exemplos:
 - i. Nome do Dispositivo
 1. Exemplos: Lâmpada da Sala, Fita Led do Escritório, Sensor de Temperatura, Ar Condicionado do Quarto e etc
 - ii. Tipo de Dispositivo
 1. Exemplos: Lâmpada, Fita Led, Sensor, Ar Condicionado e etc
 - iii. Fabricante do Dispositivo:
 1. Exemplo: Intelbras
 - iv. Estado do Dispositivo
 1. Estados: Ligado, Desligado, Desconectado
 - v. Informações do Dispositivo:
 1. Exemplos:
 - a. Lâmpada: 10 Watts
 - b. Fita Led: 5 Watts
 - c. Sensor de Temperatura: 23 °C
 - d. Ar Condicionado: Gelando (se ligado)
 - e. Etc
 - vi. IP do dispositivo
 - vii. Endereço MAC do dispositivo
 - c. Consuma os dados da API disponibilizada.
6. Uma página de **adição de dispositivos** onde todos os dispositivos cadastrados no sistema devem ser listados.
 - a. Na parte superior deve existir uma barra de busca para buscar por algum dispositivo.
 - b. Dica: Você também pode implementar um botão de filtro para filtrar por tipo.
 - c. Consuma os dados da API disponibilizada.
 - d. Ao clicar no dispositivo o usuário poderá através de um modal ou página vincular o dispositivo.
 - i. Dentre as informações deve haver um botão para vincular/adicionar o dispositivo ao perfil de usuário.

1. Quando o botão de vincular/adicionar for clicado, deverá simular um tempo de pareamento (dica: utilize setTimeout com tempos gerados randomicamente entre 5 e 20 segundos) e adicione o dispositivo ao perfil do usuário.
 2. O usuário deverá escolher o local e agrupamento (sala, cozinha, quarto e etc) do dispositivo na hora de vincular.
7. Uma página de **perfil de usuário**, contendo:
 - a. Foto do usuário
 - i. Usar a imagem da URL do cadastro caso exista, ou colocar um avatar genérico caso não exista
 - b. Nome do usuário
 - c. E-mail do usuário
 - d. Telefone do usuário
 - i. Colocar o telefone caso exista, ou esconder o campo caso não exista
 - e. Endereço do usuário
 - f. Botão/chave para habilitar a edição de todos os dados do usuário
 - i. Caso ativado, o usuário poderá editar suas informações.
 - g. Botão para realizar logout na aplicação.
8. Criação de uma **estilização (identidade visual) do sistema**. Na sequência do documento será apresentado um exemplo da aplicação proposta, conteúdo wireframes (Protótipo de baixa fidelidade) e links do figma (Protótipo de alta fidelidade). Utilize o exemplo apenas para entender o modelo a ser desenvolvido, porém fique livre para criar sua identidade visual única com cores e formatos que achar mais interessante e intuitivo.

3 EXEMPLO DE APLICAÇÃO

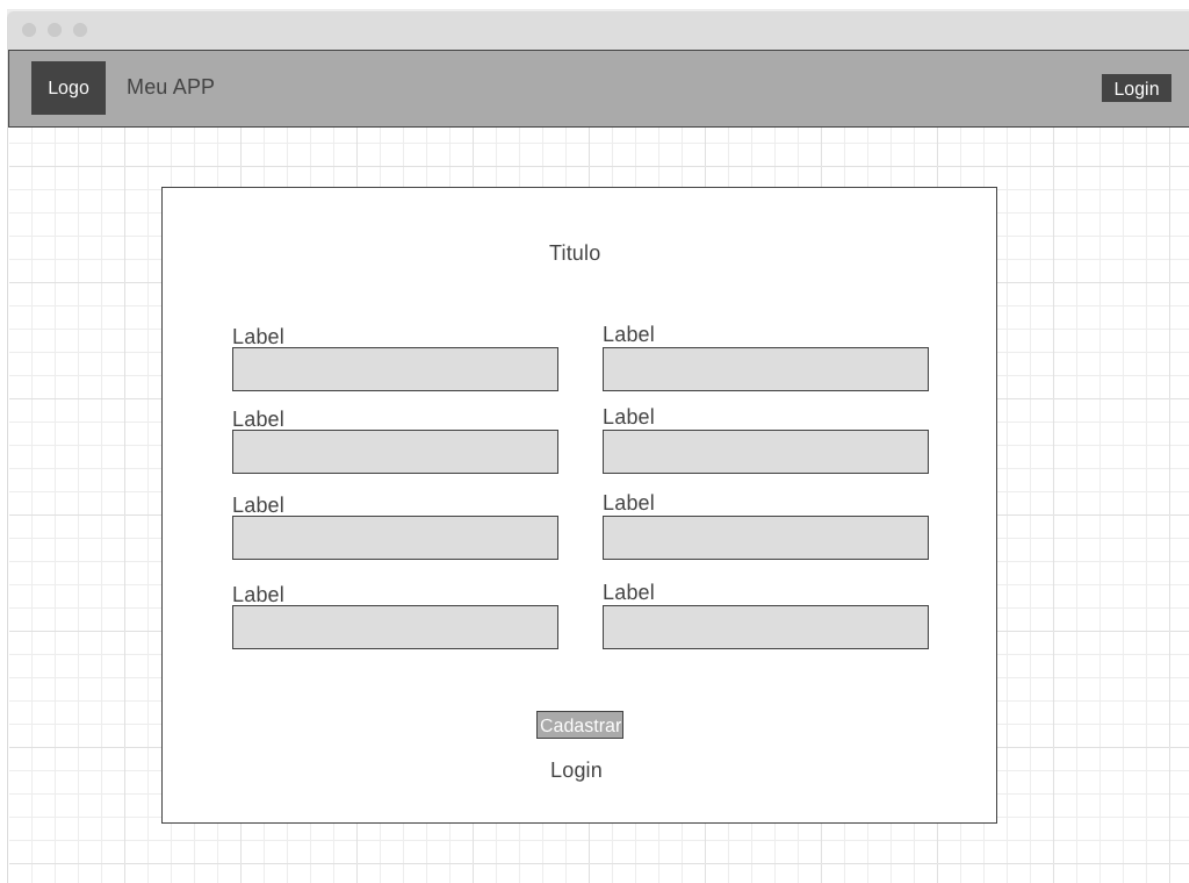
A aplicação deverá conter os requisitos apresentados anteriormente, sendo codificada utilizando a **Biblioteca React**. As imagens a seguir apresentam um exemplo de como a aplicação pode ser construída.

Figma sugestão de cores para criação do tema: [Link](#)

Figma com sugestão do layout das telas: [Link](#)

Figma com protótipo navegável: [Link](#)

Brandbook oficial da Intelbras: [Link](#)



The image shows a user registration form layout. At the top, there is a header bar with a 'Logo' placeholder, the text 'Meu APP', and a 'Login' button. The main content area is a white box with a title 'Titulo'. Inside this box, there are two columns of input fields, each preceded by a 'Label' text. Below the input fields, there is a 'Cadastrar' button and a 'Login' link.

Logo Meu APP Login

Titulo

Label

Label

Label

Label

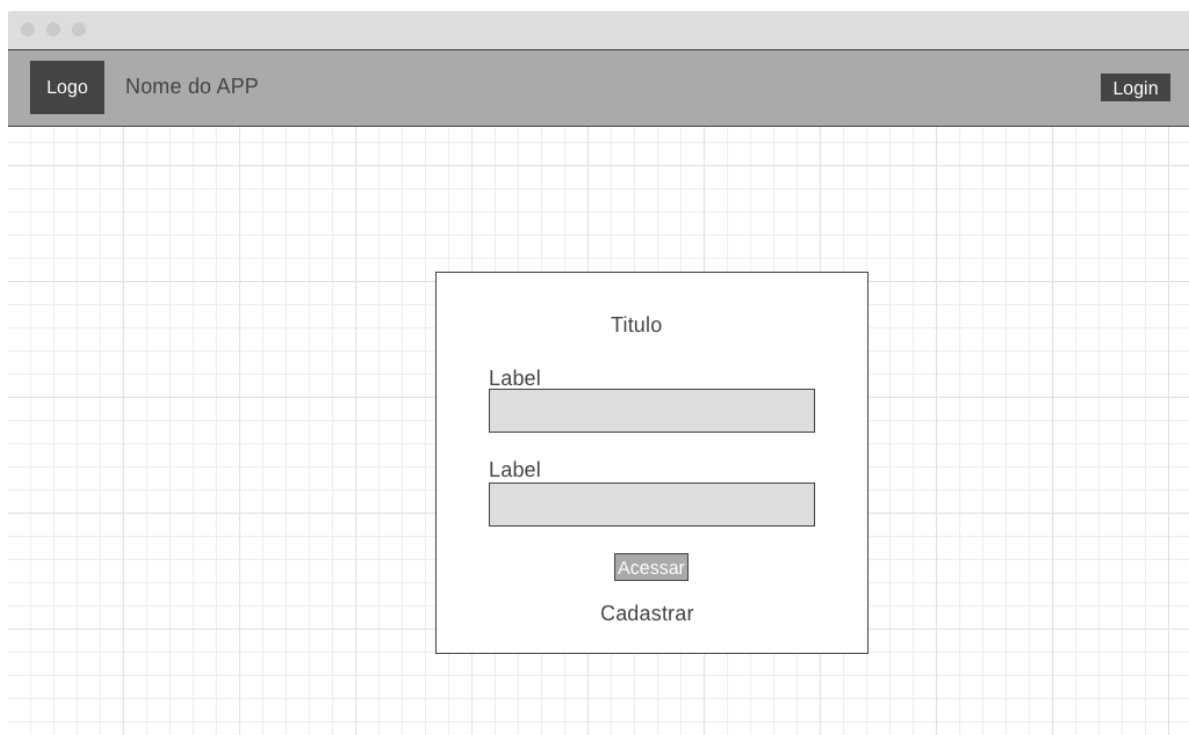
Label

Label

Cadastrar

Login

Imagem 1: Página de cadastro de usuário



The image shows a user login form layout. At the top, there is a header bar with a 'Logo' placeholder, the text 'Nome do APP', and a 'Login' button. The main content area is a white box with a title 'Titulo'. Inside this box, there are two input fields, each preceded by a 'Label' text. Below the input fields, there is an 'Acessar' button and a 'Cadastrar' link.

Logo Nome do APP Login

Titulo

Label

Label

Acessar

Cadastrar

Imagem 2: Página de login

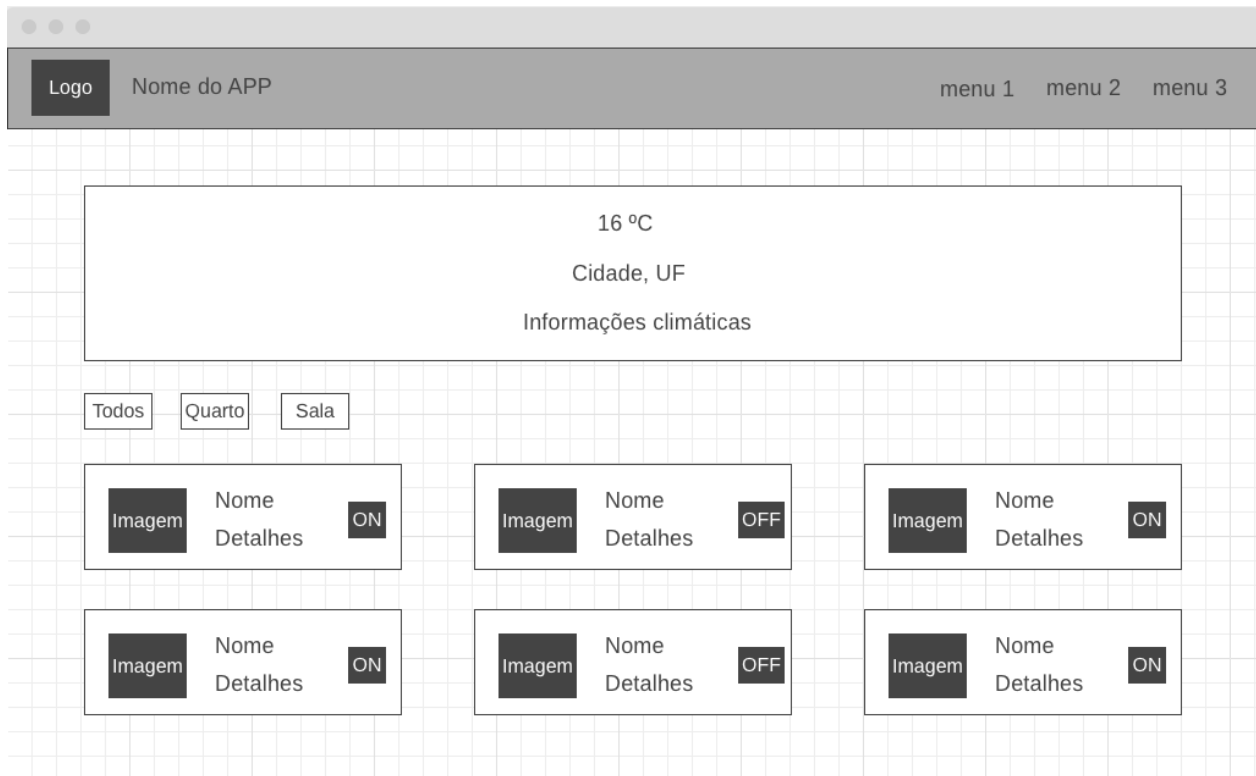


Imagem 3: Página de início

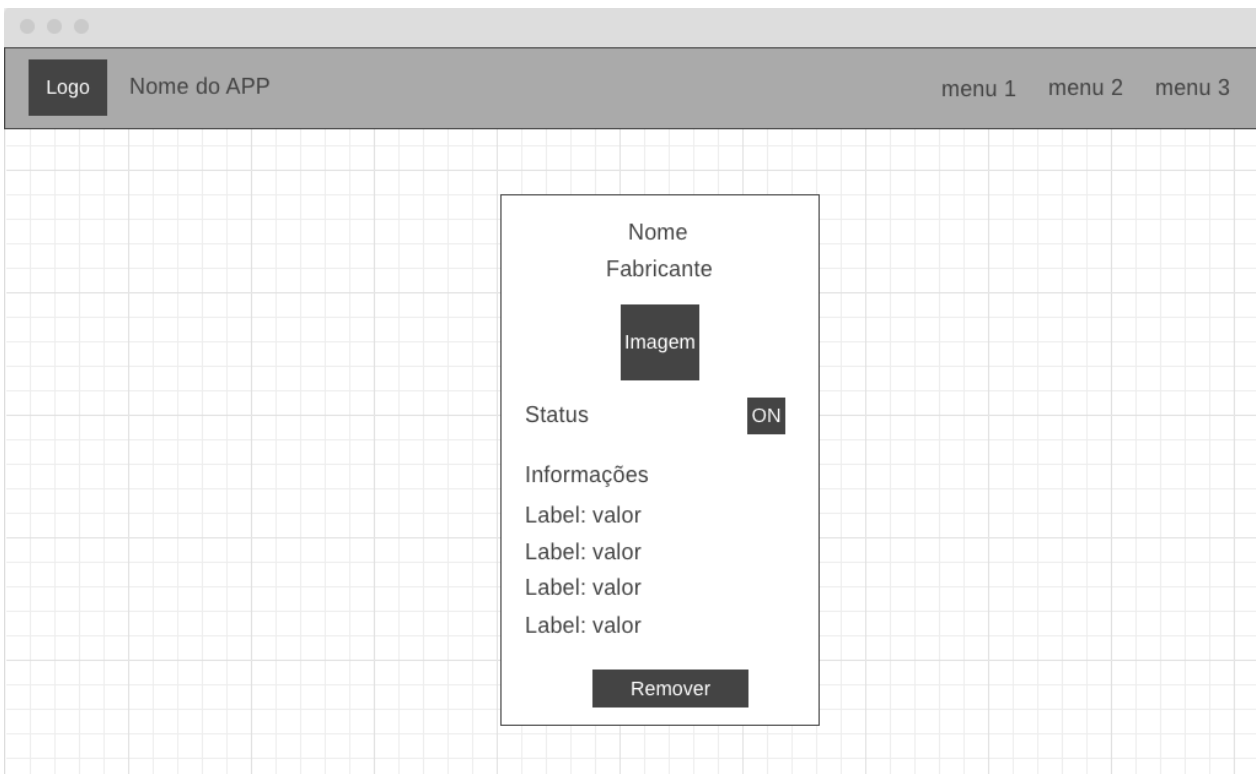


Imagem 4: Página de detalhe do dispositivo

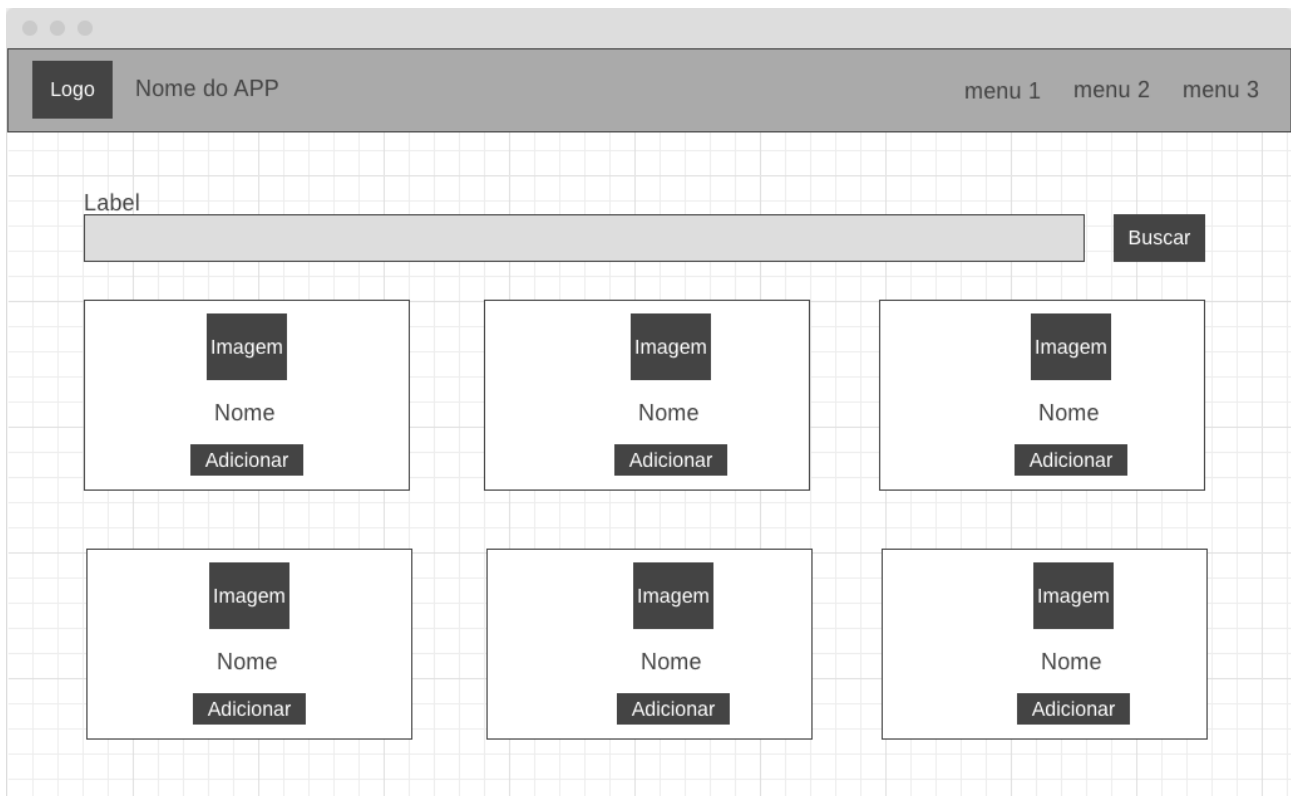


Imagem 5: Página de adição de dispositivos

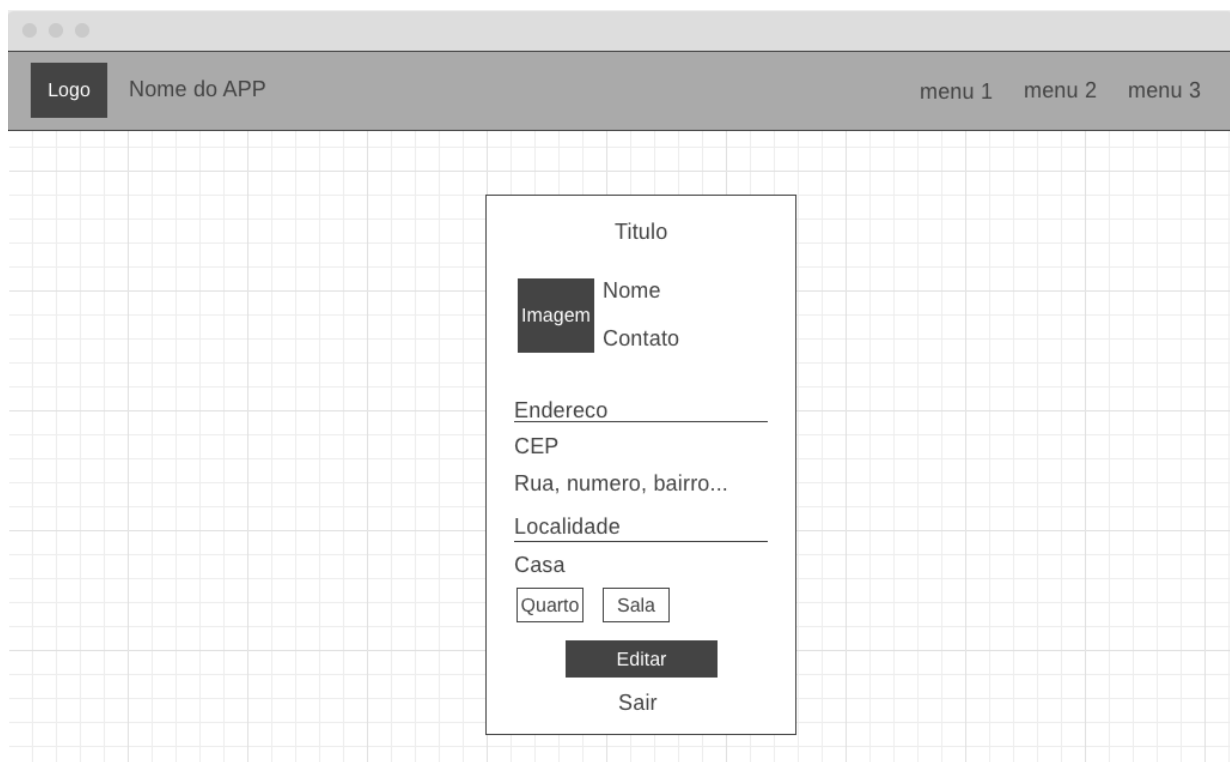


Imagem 6: Página de perfil do usuário - Visualização

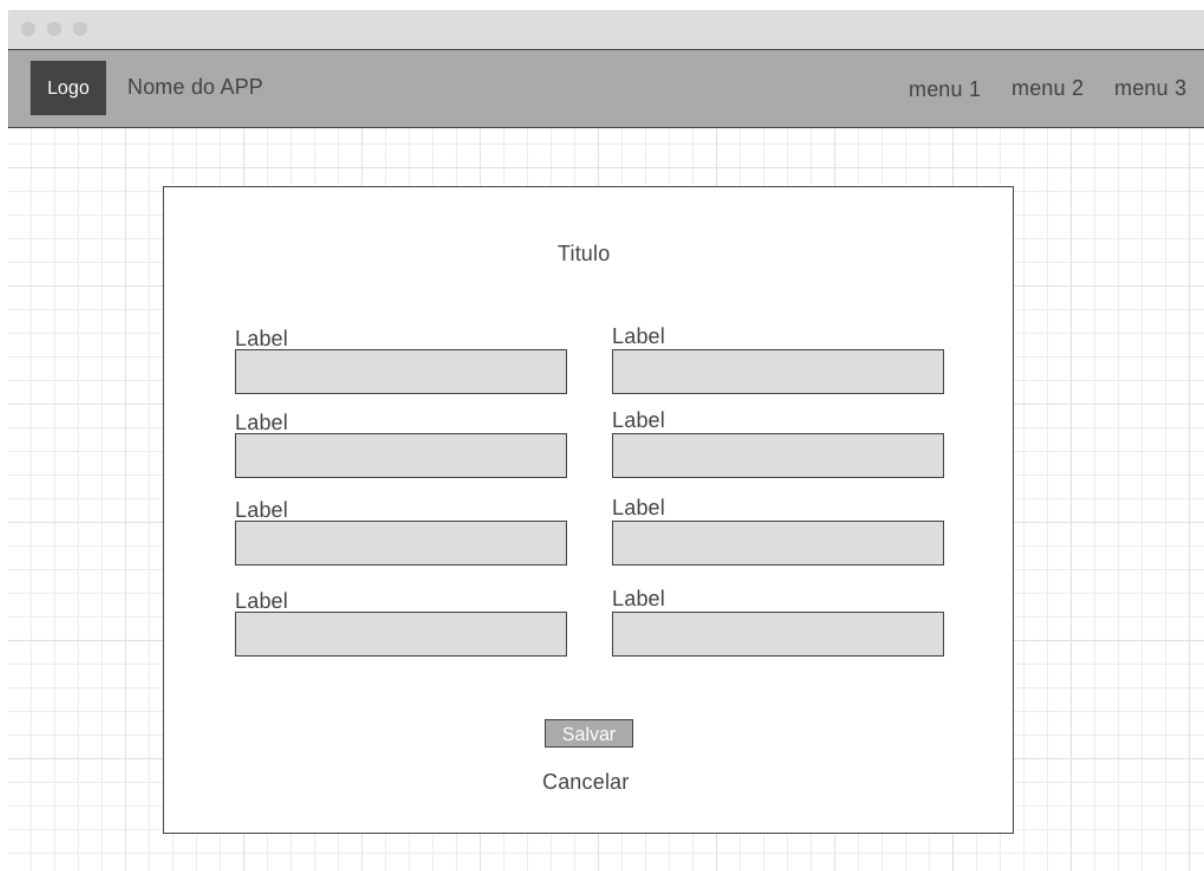


Imagem 7: Página de perfil do usuário - Edição

4 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**.

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 a 0,20	0,20 a 0,50	0,50 a 0,80
1	A estilização do projeto foi realizada com Styled Components criando pelo menos um tema contendo as cores	A estilização não foi feita utilizando Styled Components e/ou não foi criado um tema global. Não utilizou cores do	A estilização foi feita apenas em algumas partes utilizando styled components e/ou existem cores sendo utilizadas fora	A aplicação foi desenvolvida utilizando apenas styled components na estilização dos componentes. As cores

	utilizadas?	tema.	dos estilos do tema	foram utilizadas através do tema.
2	O projeto apresenta uma boa estrutura de organização e o código da aplicação está organizado, indentado e é de fácil identificação?	<p>O código está mal indentado, sem espaço entre declaração de funções, variáveis e execução de funções.</p> <p>A estrutura de pastas e módulos não está bem organizada.</p> <p>Não instalou o ESLint ou existem erros de ESLint sendo exibidos no console.</p>	<p>O projeto está bem organizado, indentado e espaçado, e os módulos e componentes foram organizados utilizando separação por papel.</p> <p>Foi realizada a instalação e configuração do ESLint corretamente e não são exibidos erros de linting no console.</p>	<p>O projeto está bem organizado, indentado e espaçado, sem erros de ESLint, e os módulos foram organizados utilizando separação por papel.</p> <p>Os componentes foram separados de acordo com o padrão Atomic.</p> <p>As chamadas API foram separadas em serviços.</p>
3	A página de login foi desenvolvida com todos os campos solicitados e realiza a validação dos campos?	<p>Não foram adicionados todos os campos solicitados e/ou não foram realizadas as validações dos campos e/ou não foi utilizado React Hook Form para a implementação do formulário.</p>	<p>Foram adicionados todos os campos solicitados e/ou existem validações, mas não foram feitas utilizando o Yup e/ou nenhum feedback é exibido feedback ao usuário quando um campo está inválido.</p>	<p>Foram adicionados todos os campos solicitados, o formulário foi criado utilizando React hook form e a validação foi feita com Yup.</p> <p>É exibido feedback ao usuário que o campo está com valor inválido.</p> <p>O formulário é submetido corretamente em caso de tudo estar ok.</p>
4	A página de cadastro de usuário foi desenvolvida com todos os campos solicitados e realiza a validação dos	<p>Não foram adicionados todos os campos solicitados e/ou não foram realizadas as validações dos campos</p>	<p>Foram adicionados todos os campos solicitados e/ou existem validações, mas não foram feitas</p>	<p>Foram adicionados todos os campos solicitados, o formulário foi criado utilizando React hook form e a validação foi feita com</p>

	campos?	e/ou não foi utilizado React Hook Form para a implementação do formulário.	utilizando o Yup e/ou nenhum feedback é exibido feedback ao usuário quando um campo está inválido.	Yup. É exibido feedback ao usuário que o campo está com valor inválido. O formulário é submetido corretamente em caso de tudo estar ok.
5	A aplicação possui um menu de navegação contendo os itens solicitados?	Não foi adicionado o menu de navegação. e/ou Os links não estão funcionando. e/ou Não foi utilizado React Router DOM para realizar a navegação entre as páginas.	Foi adicionado o menu de navegação, mas não foram utilizados os elementos HTML com a semântica correta.	Foi adicionado o menu de navegação utilizando React Router DOM para realizar os redirects das páginas. Foram utilizados os elementos html com a semântica correta.
Nº	Critério de Avaliação	0 a 0,25	0,25 a 0,75	0,75 a 1,00
6	A página de início foi implementada com todos os recursos solicitados e só é acessível se o usuário estiver logado?	A página não foi implementada ou foi implementada parcialmente. e/ou É possível acessar a página mesmo não estando logado e/ou os dados iniciais da página não são carregados via API corretamente	A página foi implementada por completo e só é acessível quando o usuário está logado. e/ou O carregamento dos dados foi implementado corretamente, porém não foi criado um custom hook para separação da lógica de busca e/ou não foi implementado o controle de estado de carregamento (loading, error, success, etc).	A página foi implementada por completo e só é acessível quando o usuário está logado. O carregamento dos dados foi implementado corretamente, tendo sido separada a lógica de de busca em um custom hook. Foi implementado o controle de estado de carregamento e utilizados para exibir feedback visual ao usuário.

7	A página de detalhes do dispositivo foi implementada corretamente e só é acessível se o usuário estiver logado?	<p>A página não foi implementada ou foi implementada parcialmente.</p> <p>e/ou</p> <p>É possível acessar a página mesmo não estando logado</p> <p>e/ou</p> <p>os dados iniciais da página não são carregados via API corretamente</p>	<p>A página foi implementada por completo e só é acessível quando o usuário está logado.</p> <p>e/ou</p> <p>O carregamento dos dados foi implementado corretamente, porém não foi criado um custom hook para separação da lógica de busca</p> <p>e/ou</p> <p>não foi implementado o controle de estado de carregamento (loading, error, success, etc).</p>	<p>A página foi implementada por completo e só é acessível quando o usuário está logado.</p> <p>O carregamento dos dados foi implementado corretamente, tendo sido separada a lógica de de busca em um custom hook.</p> <p>Foi implementado o controle de estado de carregamento e utilizados para exibir feedback visual ao usuário.</p>
8	A página de listagem dos dispositivos foi implementada corretamente e só é acessível se o usuário estiver logado?	<p>A página não foi implementada ou foi implementada parcialmente.</p> <p>e/ou</p> <p>É possível acessar a página mesmo não estando logado</p> <p>e/ou</p> <p>os dados iniciais da página não são carregados via API corretamente</p>	<p>A página foi implementada por completo e só é acessível quando o usuário está logado.</p> <p>e/ou</p> <p>O carregamento dos dados foi implementado corretamente, porém não foi criado um custom hook para separação da lógica de busca</p> <p>e/ou</p> <p>não foi implementado o controle de estado de carregamento (loading, error, success, etc).</p>	<p>A página foi implementada por completo e só é acessível quando o usuário está logado.</p> <p>O carregamento dos dados foi implementado corretamente, tendo sido separada a lógica de de busca em um custom hook.</p> <p>Foi implementado o controle de estado de carregamento e utilizados para exibir feedback visual ao usuário.</p>

9	A página de perfil do usuário foi implementada corretamente e só é acessível se o usuário estiver logado?	<p>Não foi implementado ou foi implementado parcialmente</p> <p>e/ou</p> <p>Não foi implementada a função de vincular o dispositivo ao usuário.</p> <p>e/ou</p> <p>os dados iniciais da página não são carregados via API corretamente</p>	<p>A página foi implementada por completo e só é acessível quando o usuário está logado.</p> <p>e/ou</p> <p>O carregamento dos dados foi implementado corretamente, porém não foi criado um custom hook para separação da lógica de busca</p> <p>e/ou</p> <p>não foi implementado o controle de estado de carregamento (loading, error, success, etc).</p>	<p>A página foi implementada por completo e só é acessível quando o usuário está logado.</p> <p>O carregamento dos dados foi implementado corretamente, tendo sido separada a lógica de de busca em um custom hook.</p> <p>Foi implementado o controle de estado de carregamento e utilizados para exibir feedback visual ao usuário.</p>
10	Na página de listagem foi adicionado a função de clicar no dispositivo e abrir um modal/página para vincular o dispositivo ao usuário.	<p>Não foi implementado ou foi implementado parcialmente</p> <p>e/ou</p> <p>Não foi implementada a função de vincular o dispositivo ao usuário.</p> <p>e/ou</p> <p>os dados iniciais da página não são carregados via API corretamente</p>	<p>Foi implementado a parte visual corretamente, mas utilizou outro recurso que não seja modal</p> <p>e/ou</p> <p>O carregamento dos dados foi implementado corretamente, porém não foi criado um custom hook para separação da lógica de busca</p> <p>e/ou</p> <p>não foi implementado o controle de estado de carregamento (loading, error, success, etc).</p>	<p>Foi implementada corretamente tanto a parte visual quanto a função de vincular o dispositivo ao usuário.</p> <p>Foi separada a lógica de fetch dos dados em arquivos de serviço e custom hook.</p> <p>Foi implementado o controle dos estados adicionando loading, status de erro... ao usuário.</p>

11	A busca dos dados de endereço na página de cadastro do usuário foi realizada utilizando a API ViaCEP?	Não foi implementada a busca dos dados através da API ViaCEP	A busca dos dados foi implementada, porém não foi criado um custom hook para separação da lógica de busca e/ou não foi implementado o controle de estado de carregamento (loading, error, success, etc).	A busca foi implementada corretamente, sendo separada a lógica de busca em um custom hook. e/ou Foi implementado o controle de estado de carregamento e utilizados para exibir feedback visual ao usuário.
Nº	Critério Extra	0	0,25 a 0,50	
12	A aplicação possui a possibilidade de escolha do tema pelo usuário (Ex: Dark/Light) e é responsiva em celular, tablet e desktop?	Não foi implementada a opção de troca de tema. Não foi implementado responsividade.	Foi implementado corretamente a troca de tema. O tema foi persistindo no local storage e é recuperado corretamente, desta forma, ao atualizar a página o tema permanece o escolhido pelo usuário. A aplicação está responsiva em celular tablet e desktop.	
13	Utilizou a API do Open Weather para criar um componente de exibição da previsão do tempo?	Não foi implementada a busca dos dados através da API Open Weather e/ou não criou um componente para exibição dos dados de previsão do tempo exibindo as informações mínimas (cidade, temperatura atual, previsão mínima e previsão máxima) com base na cidade cadastrada para o usuário.	Criou o componente de exibição da previsão do tempo com as informações mínimas (cidade, temperatura atual, previsão mínima e previsão máxima) com base na cidade cadastrada para o usuário. Utilizou corretamente a api do Open Weather para buscar as informações.	

14	A aplicação foi publicada e o link para acesso foi adicionado no README.md do projeto?	<p>Não foi feito o deploy do projeto.</p> <p>Não foi adicionado o README.md no projeto contendo as informações do projeto e o Link para acesso do projeto publicado.</p>	<p>Foi realizado o deploy da aplicação.</p> <p>Foi criado um README.md contendo as informações do projeto e o link para acesso ao projeto publicado.</p>
----	-----------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------

5 ENTREGA

O código desenvolvido deverá ser submetido no **GitHub**, e o link deverá ser disponibilizado na tarefa **Módulo 1 - Projeto Avaliativo 2**, presente na semana 12 do AVA até o dia **25/09/2022 às 23h55**.

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

- Michael Nascimento - **mikansc**
- Thais Bertoldo - **thaiscristinabertoldo**
- Operação DEVinHouse - **devinhouse-operacao**

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

Importante:

1. Será considerado como data final de entrega a **última atualização** no repositório do projeto no GitHub. Lembre-se de não modificar o código até receber sua nota.
2. Não esqueça de submeter o link no AVA.

6 PLANO DE PROJETO

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

- **Fundamentos da Programação Web:**
 - Arquitetura cliente-servidor, introdução ao HTML e CSS

- HTML e CSS: HTML5 (elementos semânticos) e CSS (seletores, principais estilos, layouts e flexbox)
- Organização: Kanban Board com Trello e Versionamento com GitHub
- **JavaScript:**
 - Variáveis e tipos de dados, Operadores (aritméticos, lógicos e relacionais) e Manipulação do DOM (utilização de seletores)
 - Objetos, Array, Funções, Arrow Functions, Funções de Arrays (forEach, map, filter, find, reduce e every), JSON e LocalStorage
 - Estrutura de Controle de Fluxo (condicional e repetição), Eventos, Apresentação do ES6+, Operadores Rest e Spread, Escopo (let, var e const)
 - Módulos (export e import), Funções Assíncronas (Promises, Async e Await), Interval, Timeout e fetch
- **React:**
 - Renderização de componentes, Renderização de listas, Props e Proptypes
 - Hooks Principais (useState e useEffect), eventos e lifecycles
 - Context API, Hooks Avançados (useReducer e useContext), React Router e Prop Drilling
 - Composition, Formulários (React Hook Form), Yup,
 - Organização de projeto e melhores práticas com React
 - Estilos (Styled Components), Developer Tools e Deploy