

# Mentoria – Vue básico

Proposta de uma tela única de formulário básica com suporte para mobile.

Ao entrar na tela, ela deverá estar em modo de visualização:

O endpoint a ser consultado para retornar as informações da tela é

<https://60c88bdfafc88600179f755e.mockapi.io/users/:id>

com :id sendo 1, 2, 3... até 20

Para chamar endpoints, podemos utilizar a biblioteca Axios.

O usuário pode ser premium ou não, dependendo da resposta da propriedade “isPremium”. Caso seja verdadeiro, exibirá o badge demonstrado na imagem acima. Caso contrário, exibirá o badge de user free:



O componente de dados pessoais irá exibir as informações descritas na imagem. O campo de CPF deve possuir uma máscara de CPF.

O componente de experiência profissional irá exibir a lista de experiências do usuário (propriedade experiences do endpoint). Cada experiência terá o nome da empresa e a função desempenhada pelo usuário

Caso o usuário não possua experiências profissionais, a seção 2 deve ser exibida como a seguir:

## 2. Experiência profissional

Não existem registros



No footer de botões, há um botão de editar. Ao clicar nele, a tela deve ser mostrada em modo edição:

Mentoria

Informações pessoais

Nome Sobrenome

Alterar avatar

Premium

1. Dados pessoais

Nome

Sobrenome

País

Cidade

CPF

2. Experiência profissional

Empresa

Nome empresa

Função

Função na empresa

Editar

Excluir

Empresa

empresa

io

Função na empresa

Empresa

Nome empresa

Função

Função na empresa

+ Adicionar experiência

Cancelar

Salvar

No modo edição da tela, o cabeçalho com nome e avatar do usuário terá um link para alterar o avatar do usuário. Ao clicá-lo, um dialog deve ser aberto com as opções de avatares disponíveis:



Os avatares estão disponíveis através do endpoint:

<https://60c88bdfafc88600179f755e.mockapi.io/avatars>

Ao selecionar um avatar e salvar, o avatar da tela inicial deve ser alterado. Caso o usuário somente selecione e saia do dialog, nada deve ser feito.

Os campos dos dados pessoais se tornarão inputs (preenchidos com os valores da visualização).

Os cards de experiências profissionais possuirão um menu com as opções de editar ou excluir cada experiência.

Ao clicar em editar a experiência, os valores de empresa e função devem se tornar inputs (preenchidos com os valores do modo visualização) e o menu deve dar lugar ao ícone de confirmação. O ícone de confirmação, ao ser clicado, deve voltar os campos de empresa e função para o modo de visualização com os novos valores e dar lugar novamente ao botão de menu.

No modo edição da tela, o botão “adicionar experiência” também irá aparecer. Ao clicá-lo, um novo card deve ser adicionado à lista de experiências, com os inputs vazios.

No footer, o botão de editar dará lugar a dois botões: cancelar e salvar.

Ao cancelar, todas as alterações feitas na tela devem ser ignoradas e as informações anteriores devem aparecer, com a tela novamente no modo de visualização.

Ao clicar em salvar, chamar o endpoint PUT (de mesma url do get) para salvar os dados. Os dados preenchidos devem ser mantidos, uma mensagem de sucesso deve ser exibida e a tela deve voltar ao modo visualização.

# Mobile

A tela deve ser responsiva aos diferentes breakpoints. Abaixo está um exemplo do mobile:



Mentoria

Nome Sobrenome

Premium

1. Dados pessoais

Nome

Nome usuário

Sobrenome

Sobrenome usuário

País

País do usuário

Cidade

Cidade do usuário

CPF

CPF do usuário

2. Experiência profissional

Empresa

Nome empresa

Função

Função na empresa

Empresa

Nome empresa

Função

Função na empresa

Empresa

Nome empresa

Função

Função na empresa

Editar

Links úteis:

[Vue básico – Figma](#)

[mockAPI](#)

[How To Configure Vue.js REST API Consumption with Axios](#)

[A Instância Vue — Vue.js \(vuejs.org\)](#)

[API — Vue.js \(vuejs.org\)](#)

[Custom Events — Vue.js \(vuejs.org\)](#)

[Interligações de Classe e Estilo — Vue.js \(vuejs.org\)](#)

[Renderização Condicional — Vue.js \(vuejs.org\)](#)

[Components Basics — Vue.js \(vuejs.org\)](#)

[Renderização de Listas — Vue.js \(vuejs.org\)](#)

[Dados Computados e Observadores — Vue.js \(vuejs.org\)](#)

[The Mask for Vue.js \(vuejs-tips.github.io\)](#)

[Avatar component — Vuetify \(vuetifyjs.com\)](#)

[Grid system — Vuetify \(vuetifyjs.com\)](#)

[Alert component — Vuetify \(vuetifyjs.com\)](#)

[Font Awesome](#)

[Navigation drawer component — Vuetify \(vuetifyjs.com\)](#)

[Menu component — Vuetify \(vuetifyjs.com\)](#)

[Grids - Learn web development | MDN \(mozilla.org\)](#)

Extras:

<https://cssgridgarden.com/> - Para praticar grid

<https://codepip.com/games/flexbox-froggy/> - Para praticar flexbox

[Web Skills \(andreasbm.github.io\)](#) – Para conhecer mais itens do frontend em geral