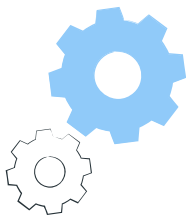


Lidando com o retorno da api

Como trabalhar melhor com o retorno do firebase



Evelise Vazquez Dibo



Baby steps



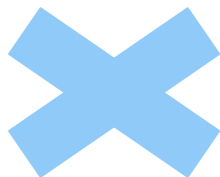
Antes de começar, precisamos entender a diferença de um site comum e uma SPA.



SPA

Single Page Application
Roda em uma única página
O conteúdo é carregado
uma única vez
O que chamamos de
componentes é carregado
por demanda

Exemplo:
<https://angular.io/docs>



Site comum

Na maioria das vezes cada
o página é carregada do
zero toda vez que
requisitada
Demora mais pra carregar

Exemplo:
[https://docctormed.com.br/
blog/](https://docctormed.com.br/blog/)

Single Page Applications



Multiple Page Applications





Isso significa que quanto menos você
“atualizar” a página toda, melhor!



https://material.io/components

Google is committed to advancing racial equity for Black communities. [See how.](#)

Components

App bars: bottom

App bars: top

Backdrop

Banners

Bottom navigation

Buttons

Buttons: floating action button

Cards

Checkboxes

Chips

Data tables

Date pickers

Dialogs

Dividers

Image lists

Lists

Components

Material Components are interactive building blocks for creating a user interface.

Browse all components or select a specific platform.

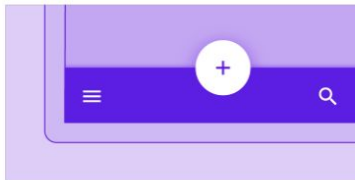
All Components

Android

Web

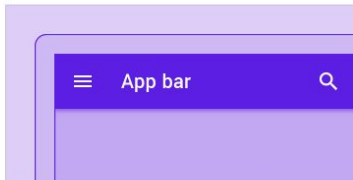
Flutter

iOS



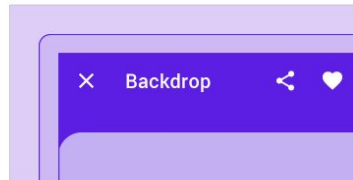
App bars: bottom

A bottom app bar displays navigation and key actions at the bottom of mobile screens



App bars: top

The top app bar displays information and actions relating to the current screen



Backdrop

A backdrop appears behind all other surfaces in an app, displaying contextual and actionable content

https://material.io/components

Google is committed to advancing racial equity for Black communities. [See how.](#)

Components

App bars: bottom

App bars: top

Backdrop

Banners

Bottom navigation

Buttons

Buttons: floating action button

Cards

Checkboxes

Chips

Data tables

Date pickers

Dialogs

Dividers

Image lists

Lists

Components

Material Components are interactive building blocks for creating a user interface.

Browse all components or select a specific platform.

All Components

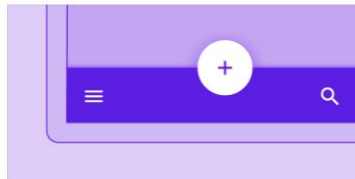
Android

Web

Flutter

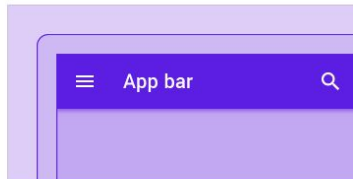
iOS

Clique do usuário



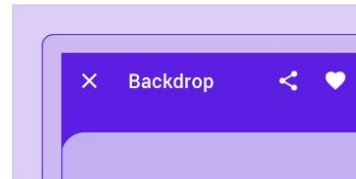
App bars: bottom

A bottom app bar displays navigation and key actions at the bottom of mobile screens



App bars: top

The top app bar displays information and actions relating to the current screen



Backdrop

A backdrop appears behind all other surfaces in an app, displaying contextual and actionable content

https://material.io/components

Google is committed to advancing racial equity for Black communities. [See how.](#)

Components

App bars: bottom

App bars: top

Backdrop

Banners

Bottom navigation

Buttons

Buttons: floating action button

Cards

Checkboxes

Chips

Data tables

Date pickers

Dialogs

Dividers

Image lists

Lists

Components

Material Components are interactive building blocks for creating a user interface.

Browse all components or select a specific platform.

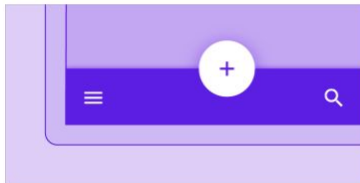
All Components

Android

Web

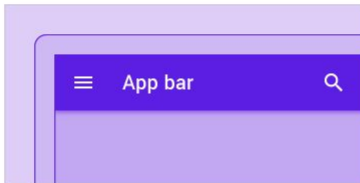
Flutter

iOS



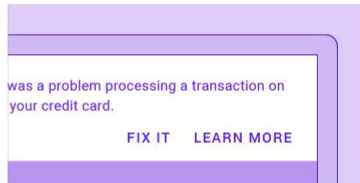
App bars: bottom

A bottom app bar displays navigation and key actions at the bottom of mobile screens.



App bars: top

The top app bar displays information and actions relating to the current screen.



Banners

A banner displays a prominent message and related optional actions.



Como fica no código?



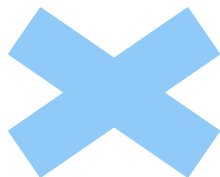
Manipular DOM

A ideia é fazer o processo de atualização dos dados na tela “manualmente” via manipulação do DOM.
Por quê?
Porque você não força a renderização da rota toda



Router change

Força a página toda renderizar novamente, mas na prática só o post, like, comentário foi atualizado



Rápido pro usuário



Devagar pro usuário



Manipulação do DOM



DOM (Document Object Model) serve é uma plataforma e interface neutra de linguagem que permite que programas e scripts acessem e atualizem dinamicamente o conteúdo, a estrutura e o estilo de um documento.

Dessa forma, você consegue navegar pela interface entre os usuários e “pegar”:

- Os elementos HTML como objetos
- As propriedades de todos os elementos HTML
- Os métodos para acessar todos os elementos HTML
- Os eventos para todos os elementos HTML



pushState



Adiciona uma nova entrada na pilha de histórico da sessão. É parecido com o `window.location`, só que a transição é mais sutil e imperceptível (quando é usada com o `PopStateEvent`).

Ou seja, a página é carregada novamente.



É trabalhoso?

Bem mais, mas a ideia é o ganho pro
usuário.
É assim que funciona no mercado.



Na prática:

<https://github.com/evelisee/SAP005-social-network>

Referências

Página de Angular - <https://angular.io/>

Material - <https://material.io/components>

O que é a Single Page Application - <https://blog.schoolofnet.com/o-que-e-uma-spa-single-page-application/>

PushState - <https://developer.mozilla.org/en-US/docs/Web/API/History/pushState>

O que é o DOM - https://www.w3schools.com/js/js_htmlDOM.asp

