Com amor, styled-components

& typescript w/ react-native

hroad

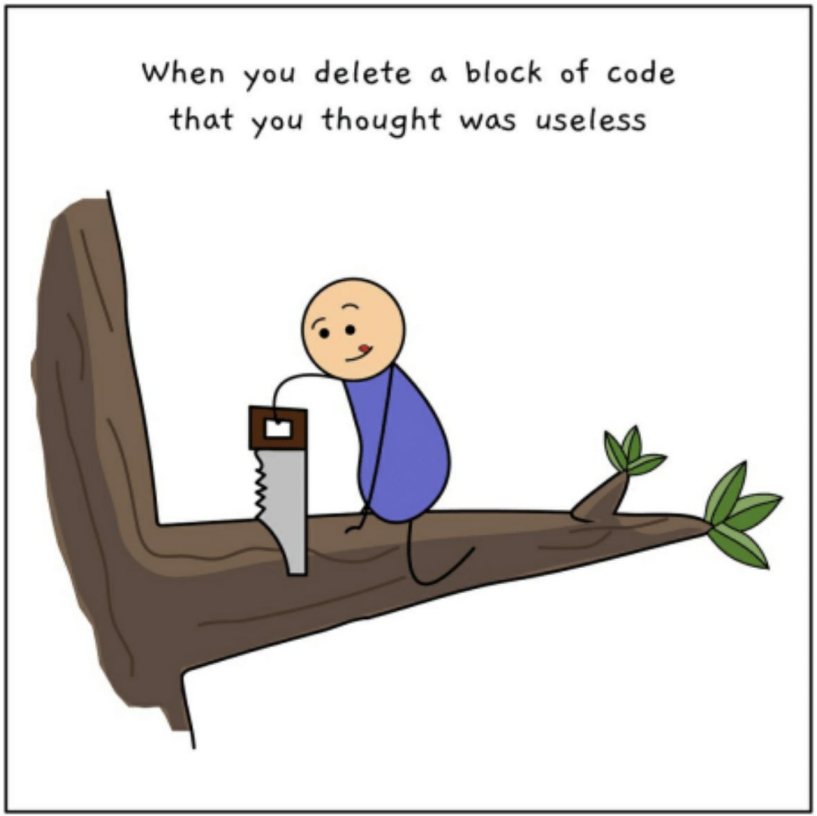
A dedicação é singular pra mim.

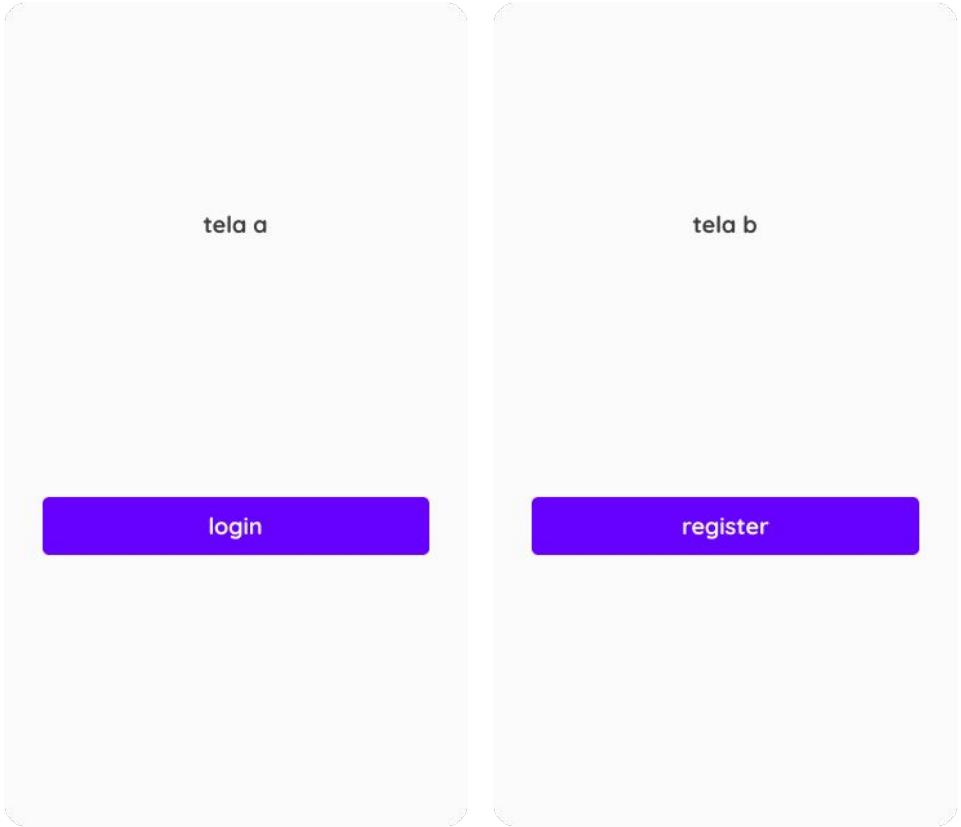
O que veremos

- ❏ Contexto
- ❏ Design
- ❏ Implementação
- ❏ Criando um componente com styled-components
- ❏ Dicas Extras
- ❏ Um pouco sobre o nosso desenvolvimento

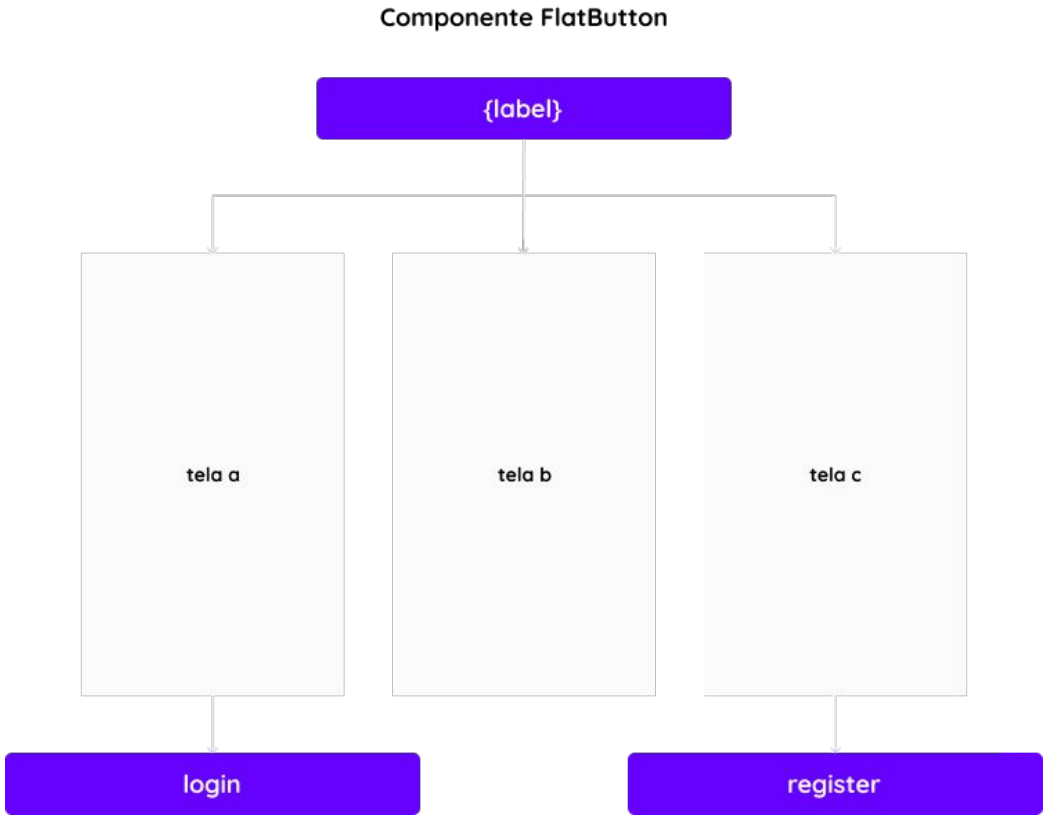
Dois pontos importantes

Manutenibilidade
Softwares Escaláveis





exemplo de uma aplicação



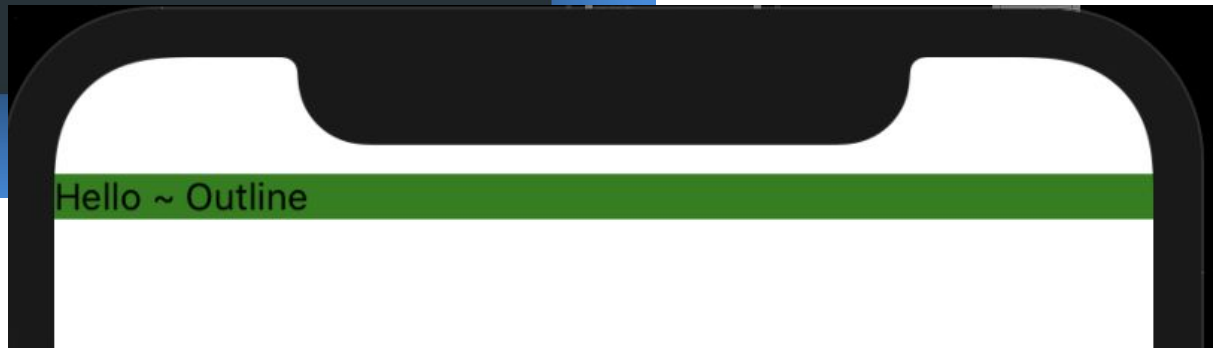
exemplo componentes

```tsx
import React from 'react';

import styled from 'styled-components/native';

import { Props, PropsStyled } from './interface';

const Container = styled.TouchableOpacity<PropsStyled>`
  background-color: ${(props) => (props.disabled ? 'green' : 'blue')};
`;

const Label = styled.Text``;

const OutlineButton = ({ text, disabled }: Props) => {
  return (
    <Container disabled={disabled}>
      <Label>{text}</Label>
    </Container>
  );
};

export default OutlineButton;
```

```tsx
export interface Props {
  text: string;
  disabled?: boolean;
}

export interface PropsStyled {
  disabled?: boolean;
}
```

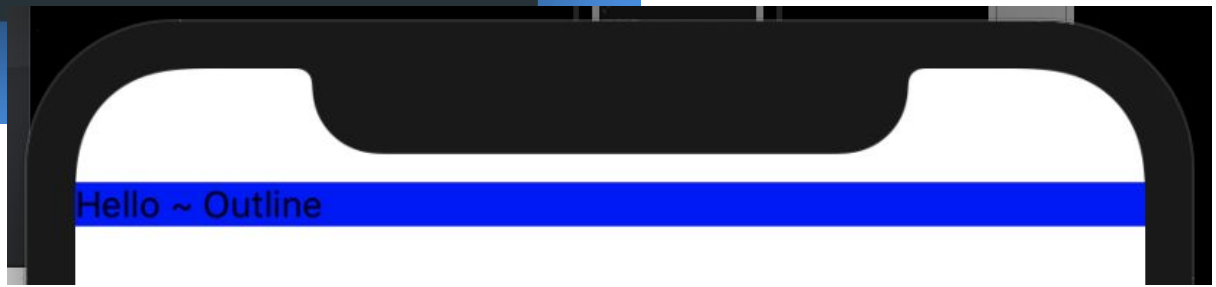
criando um componente

```
const App = () => {
  return (
    <SafeAreaView>
      <OutlineButton text="Hello ~ Flat" disabled={true} />
    </SafeAreaView>
  )
}
```



utilizando este componente

```
const App = () => {
  return (
    <SafeAreaView>
      <OutlineButton text="Hello ~ Flat" disabled={false} />
    </SafeAreaView>
  )
}
```



utilizando este componente

resultado e facilidade de manutenção de uma aplicação componentizada.

Organizando

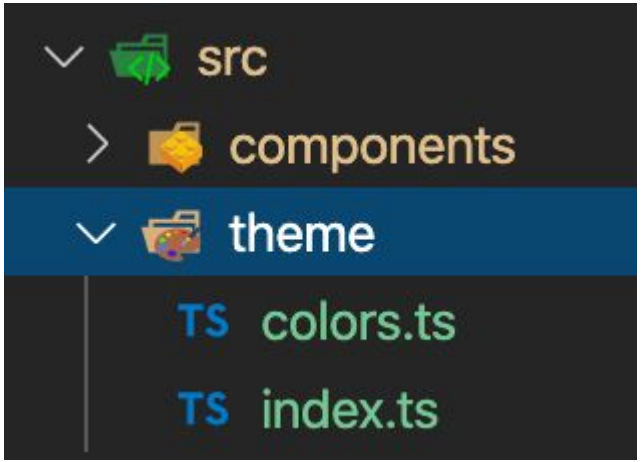
A. Instalar a dependência;
B. styled.[nome-react-native] ~ styled.Text``;
C. css-in-js;
D. Sem side-effects;
E. Sem nomes de classes;

Dicas++

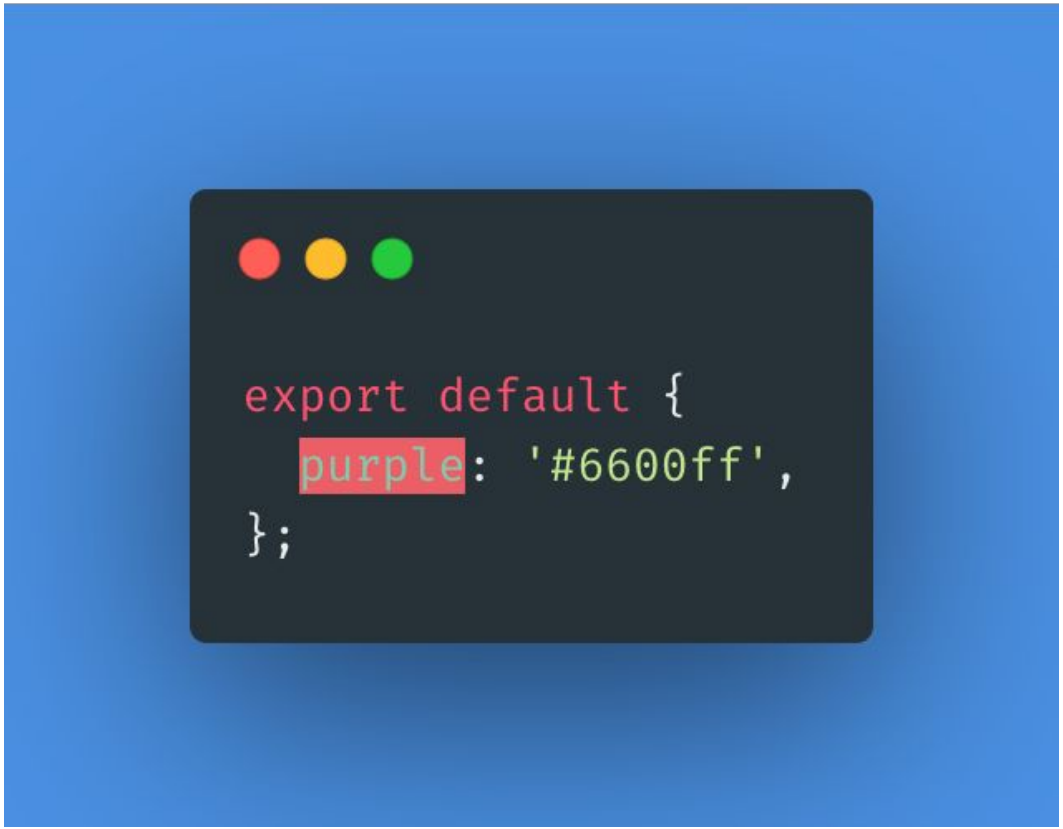
Declarando um tema;

Elegância nos componentes;

ThemeProvider



Você também pode utilizar para fontes e unidades, por exemplo.



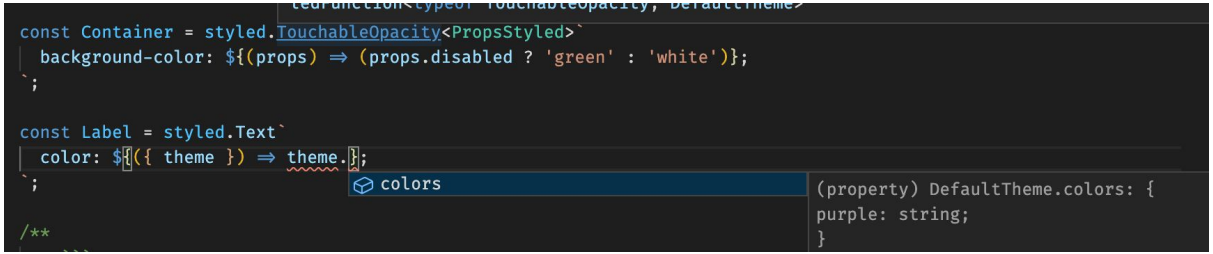
src/theme/colors.ts

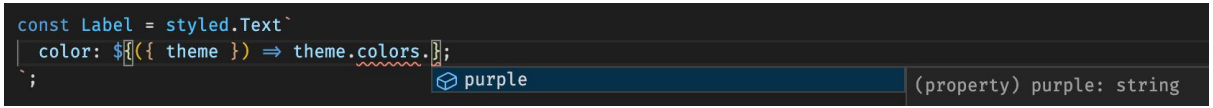
```ts
import { DefaultTheme } from 'styled-components';

import colors from './colors';

declare module 'styled-components' {
  export interface DefaultTheme {
    colors: typeof colors;
  }
}

export const theme: DefaultTheme = {
  colors,
};
```

src/theme/index.ts

incluir imagem aqui do themeprovider theme={theme}

ThemeProvider





utilizando as cores disponibilizadas pelo tema

E um pouco de elegância

Um pouco sobre nosso desenvolvimento

Tempo inicial maior na criação dos componentes;

Desafio de algo novo;

Robustez;

Genéricos;

Reutilização;

Escalabilidade;