

# **React Native - Bibliotecas de UI**

# Introdução

# Introdução

- Uma das características do React Native é a componentização.
- Mas nem sempre criar nossos componentes é a melhor opção.
- Em um projeto grande ou até mesmo em uma entrega com prazo curto, o uso de bibliotecas de interface de usuário são a melhor escolha.

# Introdução

- Existem bibliotecas que fornecem componentes já prontos e que também podem ser customizados.
- É possível também, encontrar templates com tela já pré-prontas, auxiliando na agilidade da entrega do produto.
- Vamos conhecer as principais bibliotecas.

# Galio Framework

# Galio Framework

- <https://galio.io/>
- `npm install galio-framework`



# React Native Paper

# React Native Paper

- <https://callstack.github.io/react-native-paper/>
- npm install react-native-paper
- npm install react-native-safe-area-context
- No arquivo babel.config.js inserir:

```
module.exports = function(api) {  
  api.cache(true);  
  return {  
    presets: ['babel-preset-expo'],  
    env: {  
      production: {  
        plugins: ['react-native-paper/babel'],  
      },  
    },  
  };  
};
```



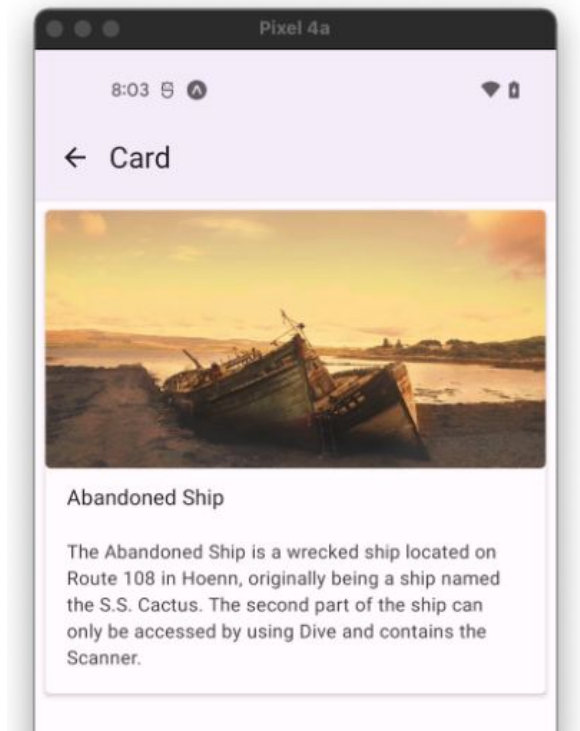


# React Native Paper

- No App.tsx inserir o provider:

```
<PaperProvider>  
  <App />  
</PaperProvider>
```

# React Native Paper



# Tamagui

# Tamagui

- <https://tamagui.dev/>
- `npm install tamagui`
- Configurar o provider:  
<https://tamagui.dev/docs/intro/installation>

# Tamagui

```
// some helpful reset styles for web:
import '@tamagui/core/reset.css'

import { TamaguiProvider, createTamagui } from 'tamagui'

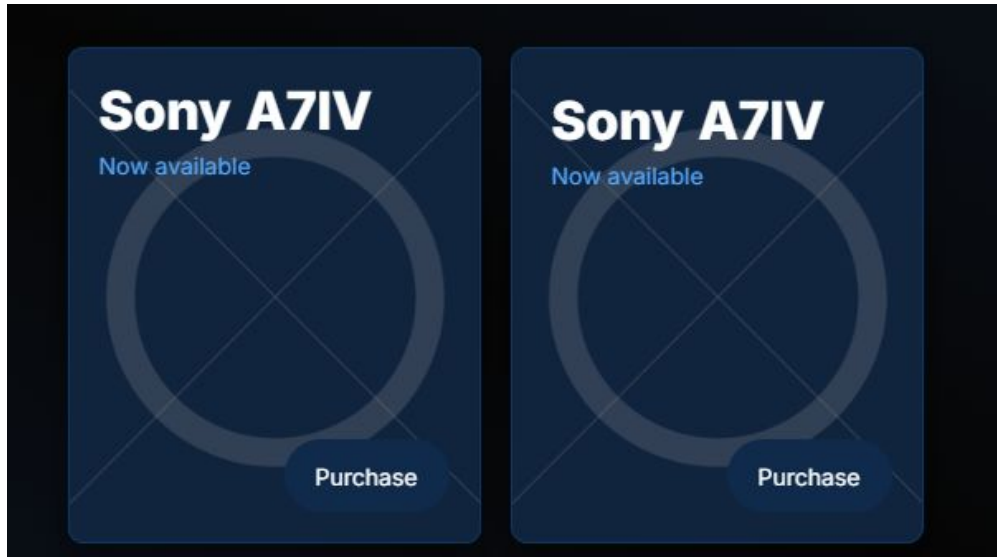
// some nice defaults:
import { config } from '@tamagui/config/v2'

// you usually export this from a tamagui.config.ts file:
// this can be as simple as an empty object
const tamaguiConfig = createTamagui(config)

// this makes typescript properly type everything based on the config
type Conf = typeof tamaguiConfig
declare module 'tamagui' {
  interface TamaguiCustomConfig extends Conf {}
}

export default () => {
  return (
    <TamaguiProvider config={tamaguiConfig}>
      { /* your app here */ }
    </TamaguiProvider>
  )
}
```

# Tamagui



# Kitten UI

# Kitten UI

- <https://akveo.github.io/react-native-ui-kitten/>
- `npm i @ui-kitten/components @eva-design/eva expo`  
`install react-native-svg@9.13.6`
- É preciso envolver o conteúdo com o provider:

```
import React from 'react';
import * as eva from '@eva-design/eva';
import { ApplicationProvider, Layout, Text } from '@ui-kitten/components';

const HomeScreen = () => (
  <Layout style={{flex: 1, justifyContent: 'center', alignItems: 'center'}}>
    <Text category='h1'>HOME</Text>
  </Layout>
);

export default () => (
  <ApplicationProvider {...eva} theme={eva.light}>
    <HomeScreen />
  </ApplicationProvider>
);
```



# Kitten UI

- <https://akveo.github.io/react-native-ui-kitten/docs/components/components-overview>



# GlueStack

# GlueStack

- <https://gluestack.io/>
- `npm i @gluestack-ui/themed @gluestack-style/react react-native-svg@13.4.0`
- `npm i @gluestack-ui/config@latest`
- Envolver a aplicação com o provider:

```
import { GluestackUIProvider, Text } from "@gluestack-ui/themed"
import { config } from "@gluestack-ui/config" // Optional if you want to use default theme

export default function App() {
  return (
    <GluestackUIProvider config={config}>
      <Text>Hello World!</Text>
    </GluestackUIProvider>
  )
}
```

# GlueStack

- Extensão para VSCode:  
<https://marketplace.visualstudio.com/items?itemName=gluestack.gluestack-vscode>
- Design Kit no Figma:  
<https://www.figma.com/community/file/1284424195134275844>

# GlueStack



May 15, 2023

## **The Power of Positive Thinking**

[Read Blog →](#)

# NativeWind

# NativeWind

- <https://www.nativewind.dev/>
- npm install nativewind
- npm install --dev tailwindcss
- npx tailwindcss init para criar o arquivo tailwind.config.js

```
// tailwind.config.js
```

```
module.exports = {  
  - content: [],  
  + content: ["../App.{js,jsx,ts,tsx}", "./<custom directory>/**/*.{js,jsx,ts,tsx}"]  
  theme: {  
    extend: {},  
  },  
  plugins: [],  
}
```

# NativeWind

Modify your `babel.config.js`

```
// babel.config.js
module.exports = function (api) {
  api.cache(true);
  return {
    presets: ["babel-preset-expo"],
    + plugins: ["nativewind/babel"],
  };
};
```



# NativeWind

```
import React from 'react';
import { withExpoSnack } from 'nativewind';

import { Text, View } from 'react-native';
import { styled } from 'nativewind';

const StyledView = styled(View)
const StyledText = styled(Text)

const App = () => {
  return (
    <StyledView className="container h-12 justify-center bg-slate-300 items-center">
      <StyledText className="text-slate-800">Try resizing me! 🦄</StyledText>
    </StyledView>
  );
}
```

# NativeWind

- Utilização: <https://tailwindcss.com/>

**Dúvidas?**



# Tarefa

- Escolha um dos projetos anteriores e refatore utilizando uma das bibliotecas de UI vistas na aula.