

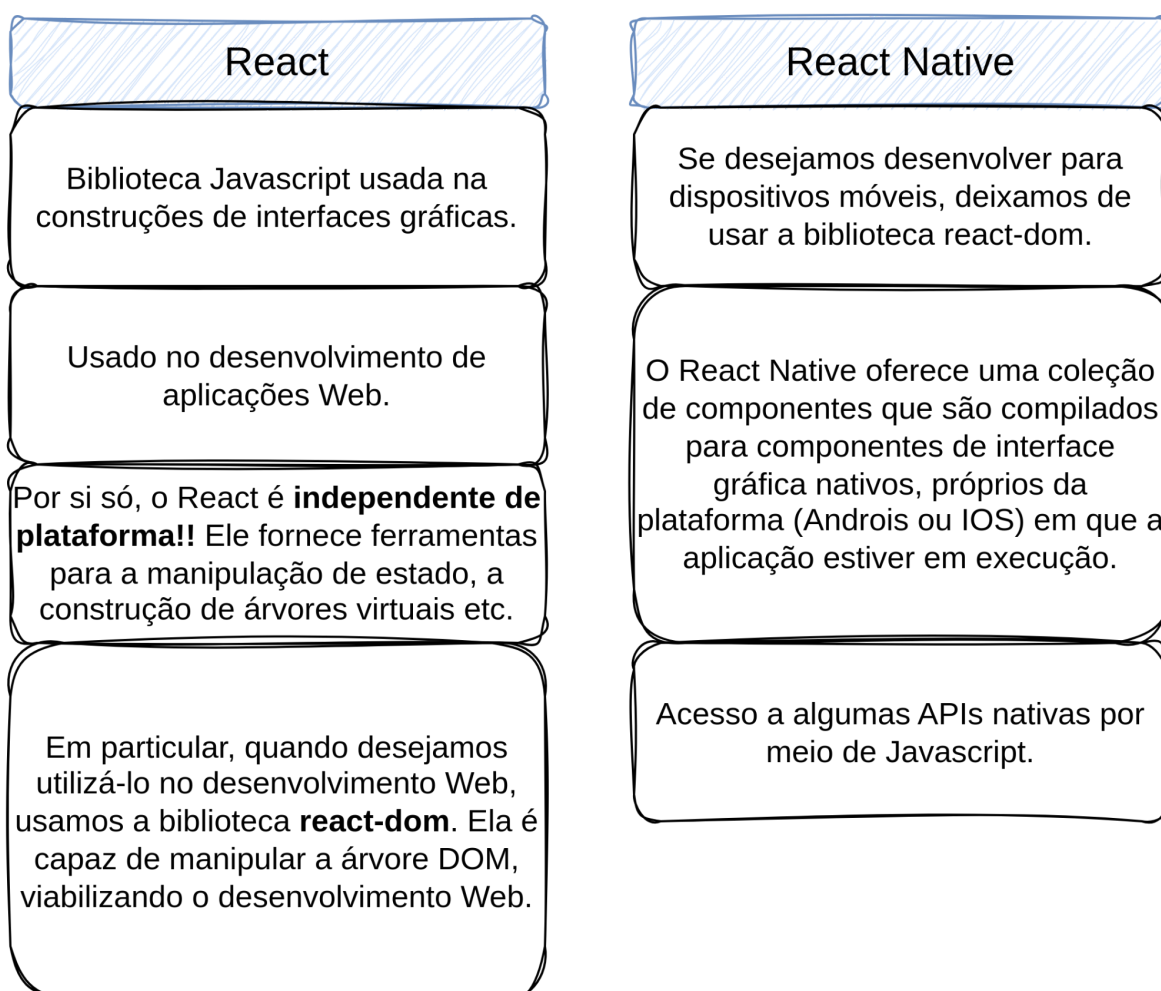
React Native

1 Introdução

React Native é um framework Open Source utilizado no desenvolvimento de aplicações para dispositivos móveis que podem ser executadas nas plataformas **Android** e **IOS**. Veja o seu site oficial.

<https://reactnative.dev/>

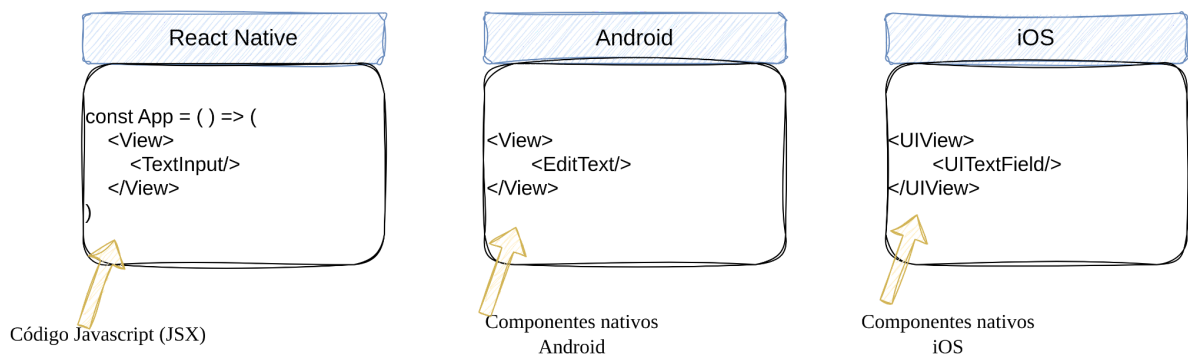
Como o nome sugere, o uso do React Native implica o uso do React, uma biblioteca Javascript amplamente utilizada no desenvolvimento de aplicações Front End, especialmente para a Web. Vejamos alguns aspectos importantes sobre React e React Native.



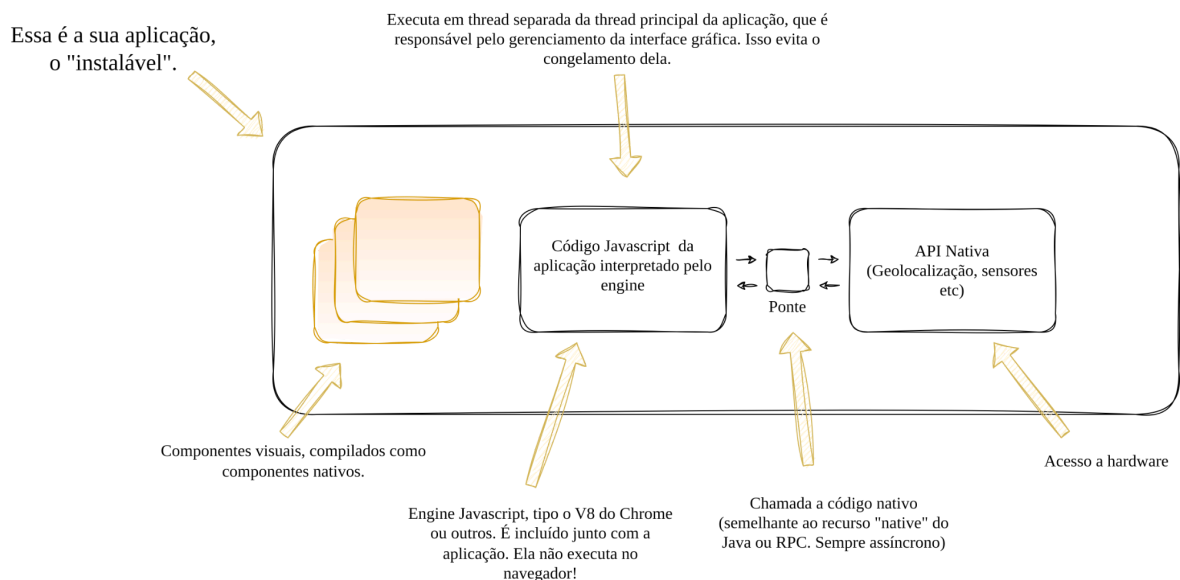
Quando uma aplicação desenvolvida com React Native é colocada em execução, ela possui

- componentes React que descrevem a interface gráfica
- código Javascript que implementa regras de negócio

Por um lado, **componentes visuais são compilados para componentes nativos da plataforma**, o que, em geral, provê bom desempenho à aplicação. No exemplo a seguir, temos um componente React Native cuja tela possui um gerenciador de leiaute que engloba um campo em que o usuário pode digitar texto. Observe como ele é compilado para código nativo dependente de plataforma.



Por outro lado, o código Javascript que implementa regras de negócio não é compilado para código nativo. Ele é executado por um engine Javascript que fica empacotado junto com a aplicação. Observe os detalhes mais importantes para este momento.



Aplicações React Native podem ser criadas de duas formas principais:

- usando o React Native CLI (Command Line Interface)
- utilizando o Expo

Vejamos os pontos mais importantes para agora.

Expo CLI	React Native CLI
É um serviço gratuito provido por terceiros.	Ferramenta React Native oficial, mantida pela equipe React Native e pela comunidade.
Pouca configuração para começar o desenvolvimento.	Em geral, mais configurações iniciais são necessárias.
Provê um ambiente de desenvolvimento "gerenciado". A criação de aplicações é simples, codificação em altíssimo nível de abstração, inúmeros pacotes para acesso a recursos às APIs nativas.	Em geral, o acesso a recursos de hardware é um pouco mais difícil, com menos opções ou em nível mais baixo de abstração quando comparado com o Expo.
Há um conceito conhecido como "ejeção". A ideia é que, ainda que optemos por utilizar o Expo, a qualquer momento podemos "ejetar" de seu ambiente gerenciado, passando a utilizar o ambiente React Native comum.	Viabiliza a integração com código nativo.
<p>Algumas razões para ejetar:</p> <ul style="list-style-type: none">- Necessidade de módulos nativos que o Expo não oferece- Escrever código nativo personalizado, específico, que nenhum módulo Expo possui- Integração com outras aplicações nativas	

Neste material, **optamos por utilizar o Expo CLI**.

2 Desenvolvimento

Agora, vamos criar nosso primeiro app React Native. Há um passo a passo no site oficial do Expo, que pode ser acessado a seguir, que vamos seguir - com eventuais variações - nesta seção.

<https://reactnative.dev/docs/environment-setup>

Comece criando uma pasta. No Windows, você pode querer algo como

```
C:\Usuários\seuUsuario\Documentos\react_native
```

E em sistemas Unix-like (Linux, Mac), você pode querer algo como

```
/home/usuario/react_native
```

Após criar um diretório, abra um terminal e vincule-o ao diretório com

```
cd path_do_diretorio
```

Utilizando o **npx**, executaremos o script **create-expo-app**, disponível na base **npm registry**. Observe.

```
npx create-expo-app primeiro-projeto
```

Observe que uma pasta foi criada. Vincule o terminal atual a ela com

```
cd primeiro-projeto
```

Abra o VS Code vinculado a ela com

```
code .
```

Feche o terminal atual. Abra um terminal interno do VS Code clicando Terminal >> New terminal. No terminal interno do VS Code, execute

```
npm start
```

para colocar - ou tentar colocar - a aplicação em execução. Veja o resultado.



Os detalhes principais são os seguintes:

- O Código QR exibido pode ser lido pela aplicação Expo, disponível no Google Play e na Apple Store. Quando fizer isso, a sua aplicação será executada no seu celular por intermédio do aplicativo Expo.
- A opção a, quando digitada direto no terminal, abre a aplicação num celular conectado à máquina ou num emulador disponível.
- A opção w coloca a aplicação para executar na web, ou seja, no seu navegador. Pode ser uma boa opção para testes iniciais.

No momento, desejamos testar na web. Entretanto, perceba que a opção w está desabilitada.



```

PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE

Starting Metro Bundler

[QR Code]

> Metro waiting on exp://10.70.212.18:8081
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

> Using Expo Go
> Press s | switch to development build

> Press a | open Android
> Press w | open web

> Press j | open debugger
> Press r | reload app
> Press m | toggle menu
> Press o | open project code in your editor

> Press ? | show all commands

```

Aperte w - com o foco no terminal mesmo - mesmo assim para ver o resultado.



```

PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE

[QR Code]

> Metro waiting on exp://10.70.212.18:8081
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

> Using Expo Go
> Press s | switch to development build

> Press a | open Android
> Press w | open web

> Press j | open debugger
> Press r | reload app
> Press m | toggle menu
> Press o | open project code in your editor

> Press ? | show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
It looks like you're trying to use web support but don't have the required dependencies installed.

Please install react-native-web@0.19.6, react-dom@18.2.0, @expo/metro-runtime@3.1.3 by running:
npx expo install react-native-web react-dom @expo/metro-runtime

If you're not using web, please ensure you remove the "web" string from the platforms array in the project
Expo config.

```

A explicação diz que há algumas dependências necessárias caso desejemos executar a aplicação na Web e que elas não estão instaladas no projeto no momento. Felizmente, a aplicação também mostra quais são elas e como instalá-las. Fazemos isso.

Encerre a execução momentaneamente com CTRL+ C. A seguir, use

`npx expo install react-native-web react-dom @expo/metro-runtime`

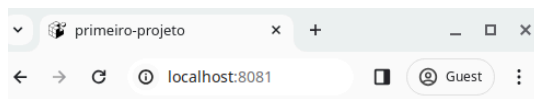
para instalar as dependências. Depois disso, execute o projeto novamente com

`npm start`

Observe que, agora, a opção w está habilitada.

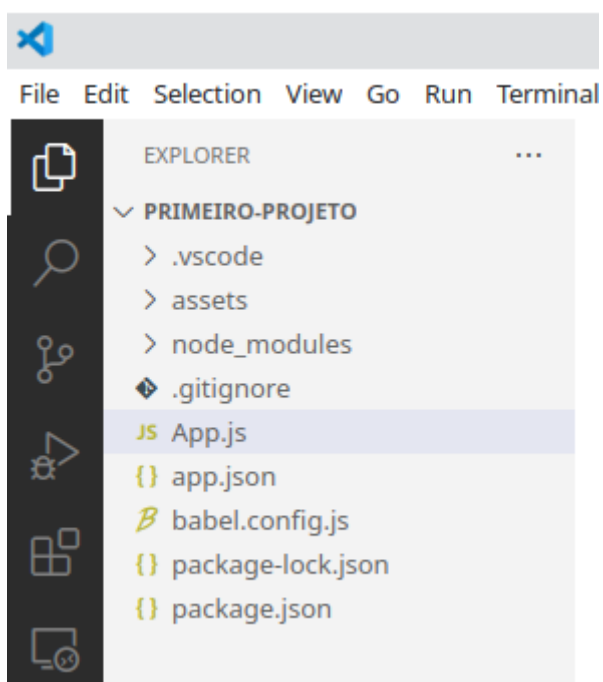


Aperte w no terminal e veja o resultado.



Open up App.js to start working on your app!

No VS Code, observe a estrutura básica da aplicação.



assets: recursos como figuras moram aqui.

node_modules: contém as dependências da aplicação, obtidas da base npm registry quando a aplicação foi criada.

.gitignore: arquivo tipicamente utilizado para instruir o Git a respeito dos arquivos que ele deve desconsiderar.

App.js: arquivo Javascript que contém a definição do componente principal da aplicação.

app.json: configurações da aplicação, como seu nome, modo (retrato ou paisagem) padrão etc.

babel.config.js: configurações do compilador Javascript Babel. Ele viabiliza o uso de código Javascript mais moderno, potencialmente ainda não disponível no engine que estivermos utilizando. Também compila expressões JSX. Também é capaz de aplicar transformações no código a fim de aprimorar o desempenho da aplicação.

package-lock.json: tem diversos propósitos, como

- descrever textualmente as dependências do projeto
- definir scripts de execução
- definir dados sobre a aplicação, como a versão, nome etc

package-lock.json: gerado automaticamente quando o npm modifica a pasta node_modules e/ou o arquivo package.json. Ele descreve a árvore de dependências, incluindo a versão exata de cada dependência. Isso permite que desenvolvedores diferentes utilizem as mesmas versões. Garante também que os ambientes de desenvolvimento e de produção, por exemplo, utilizem as mesmas versões.

Nota. Se você inspecionar o conteúdo do arquivo package.json, perceberá que a biblioteca **react-dom** está listada, muito embora não estejamos desenvolvendo para a web (apesar do teste inicial no navegador). Ocorre que, além dela, também consta a biblioteca react-native-web. Ela permite que aplicações React Native sejam compiladas para a Web e, por isso, usa a react-dom. A react-dom é, portanto, dependência da react-native-web.

Abra o arquivo **App.js** para fazermos uma pequena alteração.

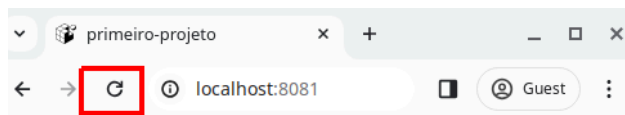
```
import { StatusBar } from 'expo-status-bar';
import { StyleSheet, Text, View } from 'react-native';

export default function App() {
  return (
    <View style={styles.container}>
      <Text>Hello, React Native!</Text>
      <StatusBar style="auto" />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

Nota. Pode ser boa ideia manter o VS Code em modo Auto save (File >> Auto save ou Arquivo >> Salvamento automático). Caso não faça isso, lembre-se de salvar seus arquivos de vez em quando, especialmente quando for testar a aplicação.

Clique no botão atualizar do navegador para ver o resultado.



Hello, React Native!

Referências

React Native. 2024. Disponível em <<https://reactnative.dev/>>. Acesso em 2024.