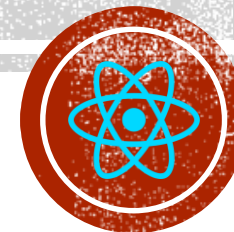


# INTRODUÇÃO AO REACT NATIVE



**Douglas Nassif Roma Junior**

 /douglasjunior

 /in/douglasjunior

 douglasjunior.me

 nassifrroma@gmail.com

Slides: <https://git.io/vbU3N>

# AGENDA

- Nativas vs Híbrido vs Multi-plataforma
- Introdução ao React Native
- Instalação
- Criando projetos
- Executando o projeto
- Explorando o ambiente
- Referências

# NATIVO

- Aplicativo desenvolvido, compilado e empacotado na plataforma nativa do sistema operacional.
  - Android: Java, Kotlin, C++, etc.
  - iOS: Objective-C, Swift, C, C++, etc.
  - Windows: C#, C++, etc.
- Todos os recursos de software e hardware estão disponíveis naturalmente.

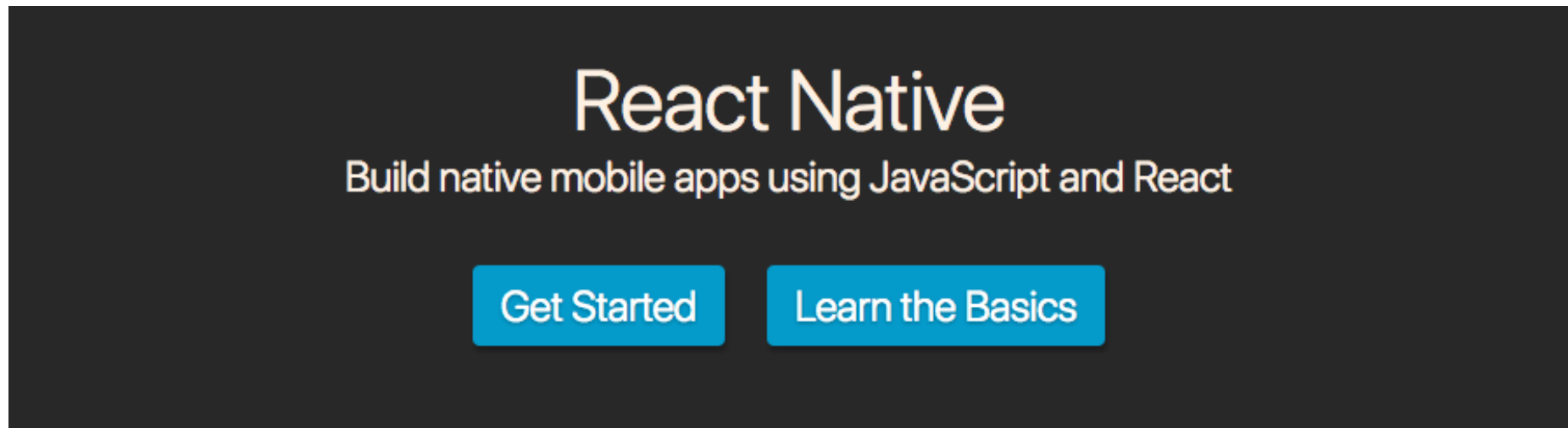
# HÍBRIDO

- Aplicativo desenvolvido em uma única linguagem e interpretado (Web View) em diversas plataformas.
  - Cordova/Phonegap: Html, CSS, JavaScript
  - Ionic: Html, CSS, JavaScript/TypeScript, Angular
  - Meteor: Html, CSS, JavaScript/TypeScript
- Desempenho inferior ao nativo.
- Para acessar recursos de hardware ou sistema operacional é preciso criar Plugins complexos e difíceis de manter.
- Sem controle ao projeto nativo.

# MULTI-PLATAFORMA

- Aplicativo desenvolvido em uma única linguagem, compilado e empacotado para a plataforma nativa.
  - React Native: JavaScript, TypeScript, React, etc.
  - Xamarin: C#.
  - Native Script: JavaScript, TypeScript, Angular, etc.
- Desempenho equivalente ao nativo.
- Para acessar recursos de hardware ou sistema operacional é possível criar módulos que, naturalmente são mais fáceis de manter devido a proximidade com a plataforma nativa.

# INTRODUÇÃO AO REACT NATIVE



Build native mobile apps using JavaScript and React

React Native lets you build mobile apps using only JavaScript. It uses the same design as React, letting you compose a rich mobile UI from declarative components.

# INTRODUÇÃO AO REACT NATIVE

- **Construa apps móveis nativos usando JavaScript e React.**
  - React Native permite construir apps utilizando apenas JavaScript. Ele utiliza o mesmo design do React, permitindo a composição de interfaces móveis ricas, utilizando componentes

```
import React, { Component } from 'react';
import { Text, View } from 'react-native';

class WhyReactNativeIsSoGreat extends Component {
  render() {
    return (
      <View>
        <Text>
          Se você gosta do React na Web, você vai gostar do React Native.
        </Text>
        <Text>
          Basta utilizar componentes nativos, como 'View' e 'Text',
          no lugar de componentes Web como 'div' ou 'span'.
        </Text>
      </View>
    );
  }
}
```

# INTRODUÇÃO AO REACT NATIVE

- **Um app React Native é um app realmente nativo.**
  - Com React Native, você não constrói “web app”, ou “HTML5 app”, ou um “app híbrido”. Você constrói um app mobile real que é indistinguível de qualquer app criado com Java ou Objective-C.

```
<ScrollView>
  <Image
    source={{ uri: 'https://i.chzbgr.com/full/7345954048/h7E2C65F9/' }}
    style={{ width: 320, height: 180 }}
  />
  <Text>
    No iOS, um ScrollView do React Native usa o nativo UIScrollView.
    No Android, ele usa o nativo ScrollView.

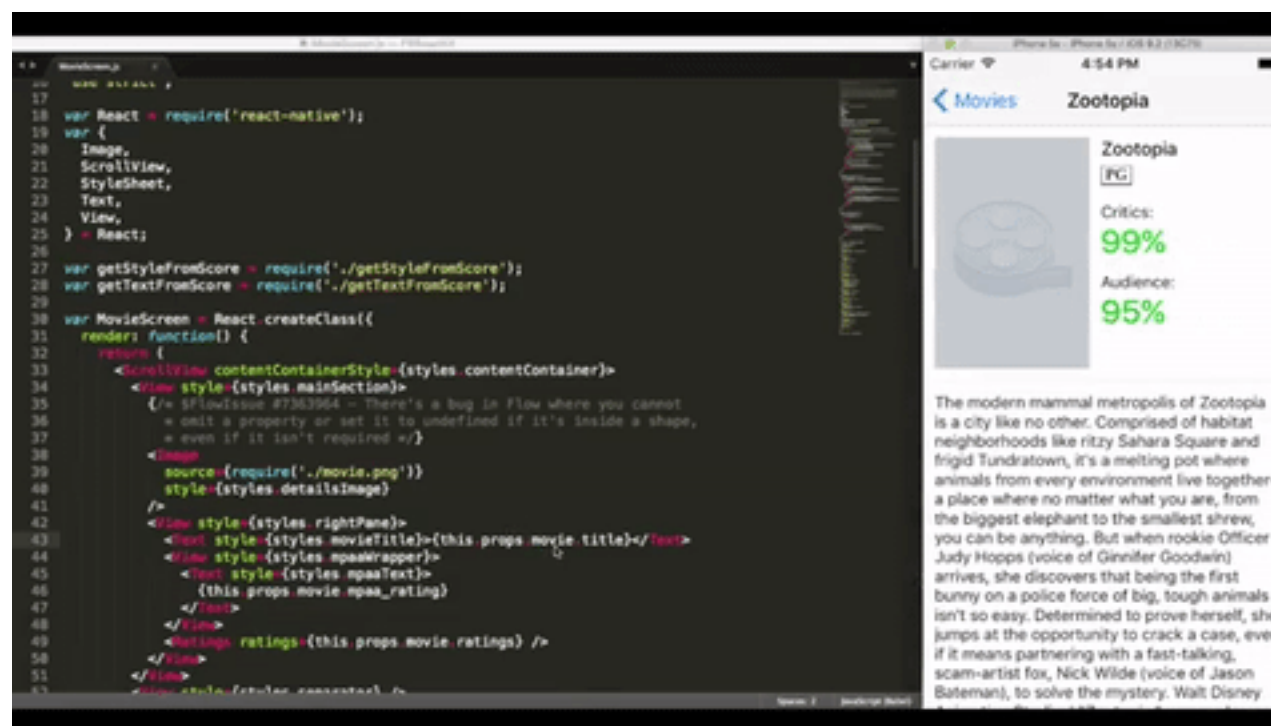
    No iOS, um Image do React Native usa o nativo UIImageView.
    No Android, ele usa o nativo ImageView.

    O React Native abstrai a maioria dos componentes fundamentais
    permitindo que você aproveite toda a performance nativa com React.
  </Text>
</ScrollView>
```



# INTRODUÇÃO AO REACT NATIVE

- **Não perca tempo recompilando**
  - O React Native permite construir apps rapidamente. Em vez de recompilar, você pode recarregar seu app instantaneamente.



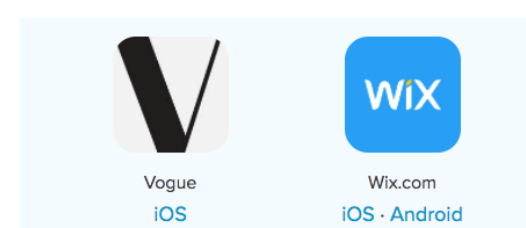
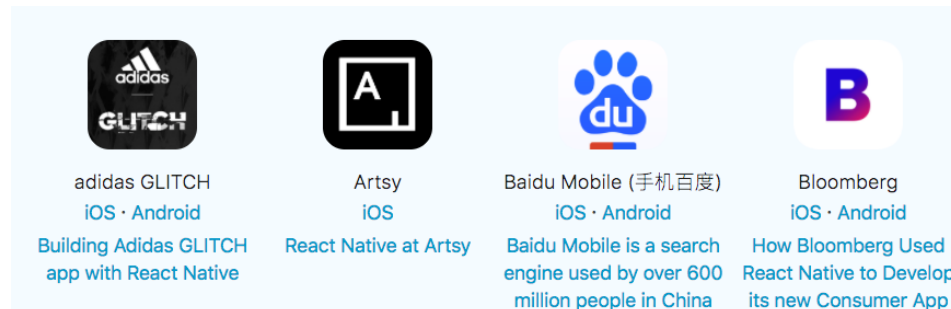
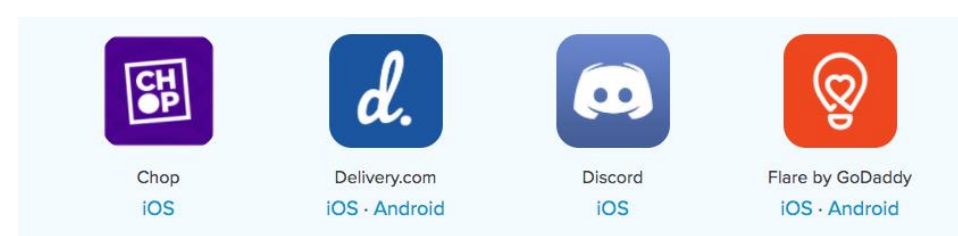
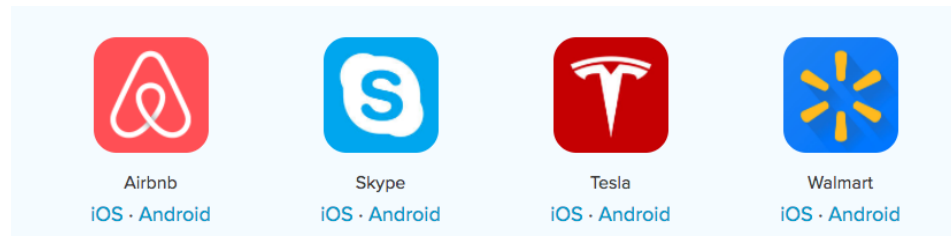
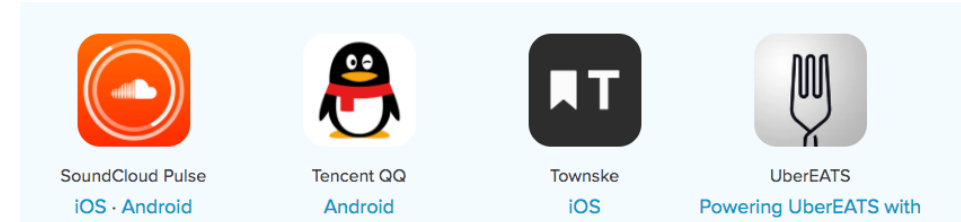
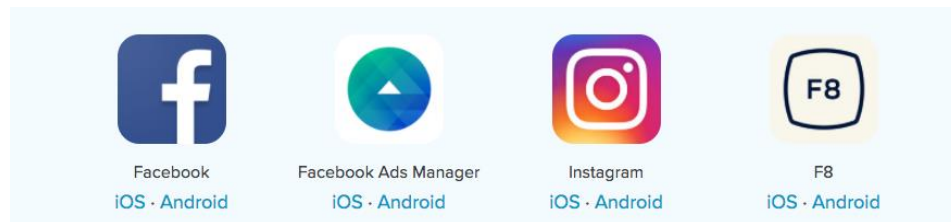
# INTRODUÇÃO AO REACT NATIVE

- **Use código nativo quando precisar.**
  - É simples acessar recursos nativos se você precisar otimizar sua aplicação. Assim como é fácil de construir partes de seu app em React Native, e outros partes usando código nativo diretamente. É assim que o app do Facebook funciona.

```
import { TheGreatestComponentInTheWorld } from './your-native-code';  
  
<View>  
  <TheGreatestComponentInTheWorld />  
  <Text>  
    TheGreatestComponentInTheWorld permite utilizar Objective-C,  
    Java, ou Swift - o processo de desenvolvimento é o mesmo.  
  </Text>  
</View>
```

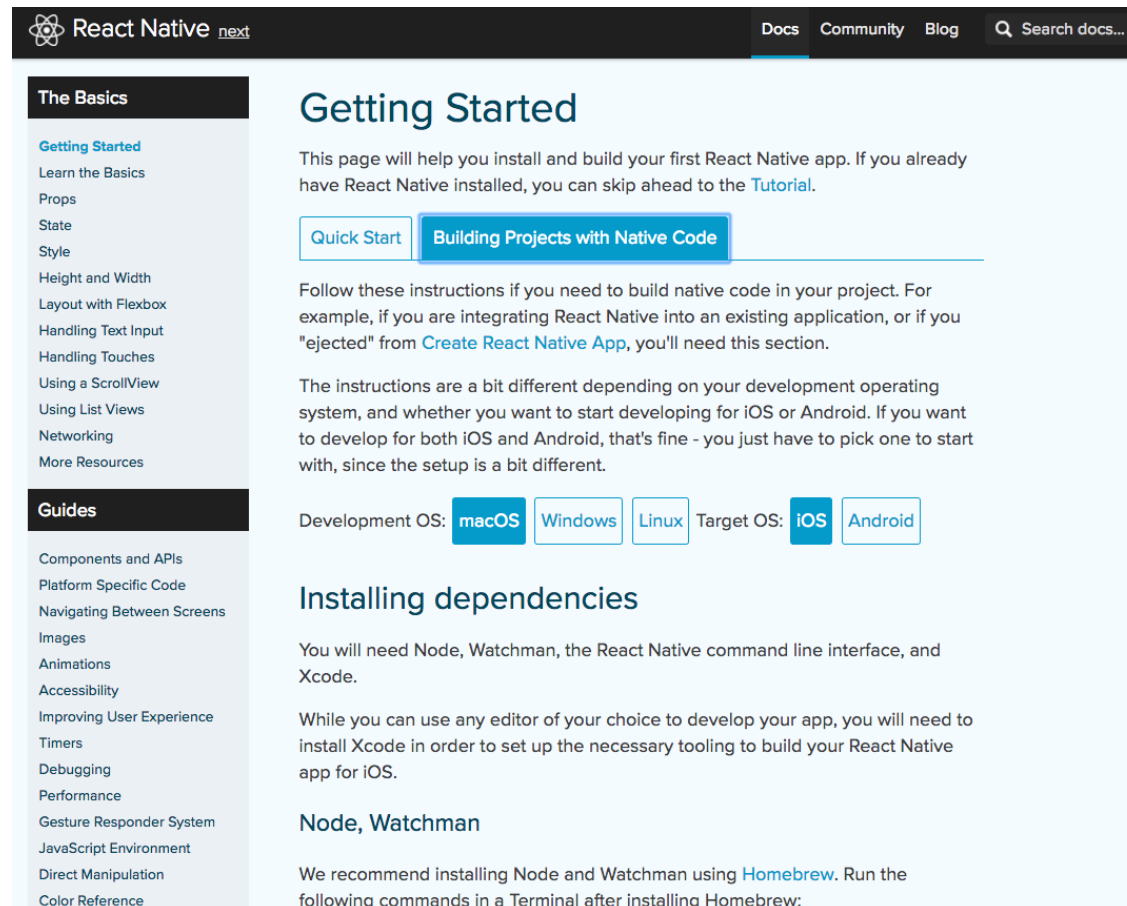
# INTRODUÇÃO AO REACT NATIVE

## ■ Quem utiliza?



# INTRODUÇÃO AO REACT NATIVE

- Documentação rica



The screenshot shows the React Native documentation website. The header includes the React Native logo, the word 'next', and navigation links for 'Docs', 'Community', 'Blog', and a search bar. The left sidebar has two main sections: 'The Basics' and 'Guides'. 'The Basics' includes links like 'Getting Started', 'Learn the Basics', 'Props', 'State', 'Style', 'Height and Width', 'Layout with Flexbox', 'Handling Text Input', 'Handling Touches', 'Using a ScrollView', 'Using List Views', 'Networking', and 'More Resources'. 'Guides' includes links like 'Components and APIs', 'Platform Specific Code', 'Navigating Between Screens', 'Images', 'Animations', 'Accessibility', 'Improving User Experience', 'Timers', 'Debugging', 'Performance', 'Gesture Responder System', 'JavaScript Environment', 'Direct Manipulation', and 'Color Reference'. The main content area is titled 'Getting Started' and contains text about installing and building a first React Native app. It has two buttons: 'Quick Start' and 'Building Projects with Native Code'. Below this, it explains that instructions are different depending on the development operating system (iOS or Android) and whether you want to develop for both. It includes a section for 'Installing dependencies' and a section for 'Node, Watchman'.

React Native next Docs Community Blog Search docs...

## The Basics

- Getting Started
- Learn the Basics
- Props
- State
- Style
- Height and Width
- Layout with Flexbox
- Handling Text Input
- Handling Touches
- Using a ScrollView
- Using List Views
- Networking
- More Resources

## Guides

- Components and APIs
- Platform Specific Code
- Navigating Between Screens
- Images
- Animations
- Accessibility
- Improving User Experience
- Timers
- Debugging
- Performance
- Gesture Responder System
- JavaScript Environment
- Direct Manipulation
- Color Reference

## Getting Started

This page will help you install and build your first React Native app. If you already have React Native installed, you can skip ahead to the [Tutorial](#).

[Quick Start](#) [Building Projects with Native Code](#)

Follow these instructions if you need to build native code in your project. For example, if you are integrating React Native into an existing application, or if you "ejected" from [Create React Native App](#), you'll need this section.

The instructions are a bit different depending on your development operating system, and whether you want to start developing for iOS or Android. If you want to develop for both iOS and Android, that's fine - you just have to pick one to start with, since the setup is a bit different.

Development OS: [macOS](#) [Windows](#) [Linux](#) Target OS: [iOS](#) [Android](#)

## Installing dependencies

You will need Node, Watchman, the React Native command line interface, and Xcode.

While you can use any editor of your choice to develop your app, you will need to install Xcode in order to set up the necessary tooling to build your React Native app for iOS.

## Node, Watchman

We recommend installing Node and Watchman using [Homebrew](#). Run the following commands in a Terminal after installing Homebrew:

# INTRODUÇÃO AO REACT NATIVE

- Comunidade ativa

## Where To Get Support

**React Native** is worked on full-time by Facebook's product infrastructure engineering team. But there are far more people in the community who make key contributions and fix things. So if you need help with your React Native app, the right place to go depends on the type of help that you need.

### Stack Overflow

Many members of the community use Stack Overflow to ask questions. Read through the [existing questions](#) tagged with **react-native** or [ask your own](#)!

### Discussion Forum

For longer-form conversations about React Native, we've set up a [discussion forum at `discuss.reactjs.org`](#). This forum is a great place for discussion about best practices and application architecture as well as the future of React Native. If you have an answerable code-level question, please post it to Stack Overflow instead.

### Reactiflux Chat

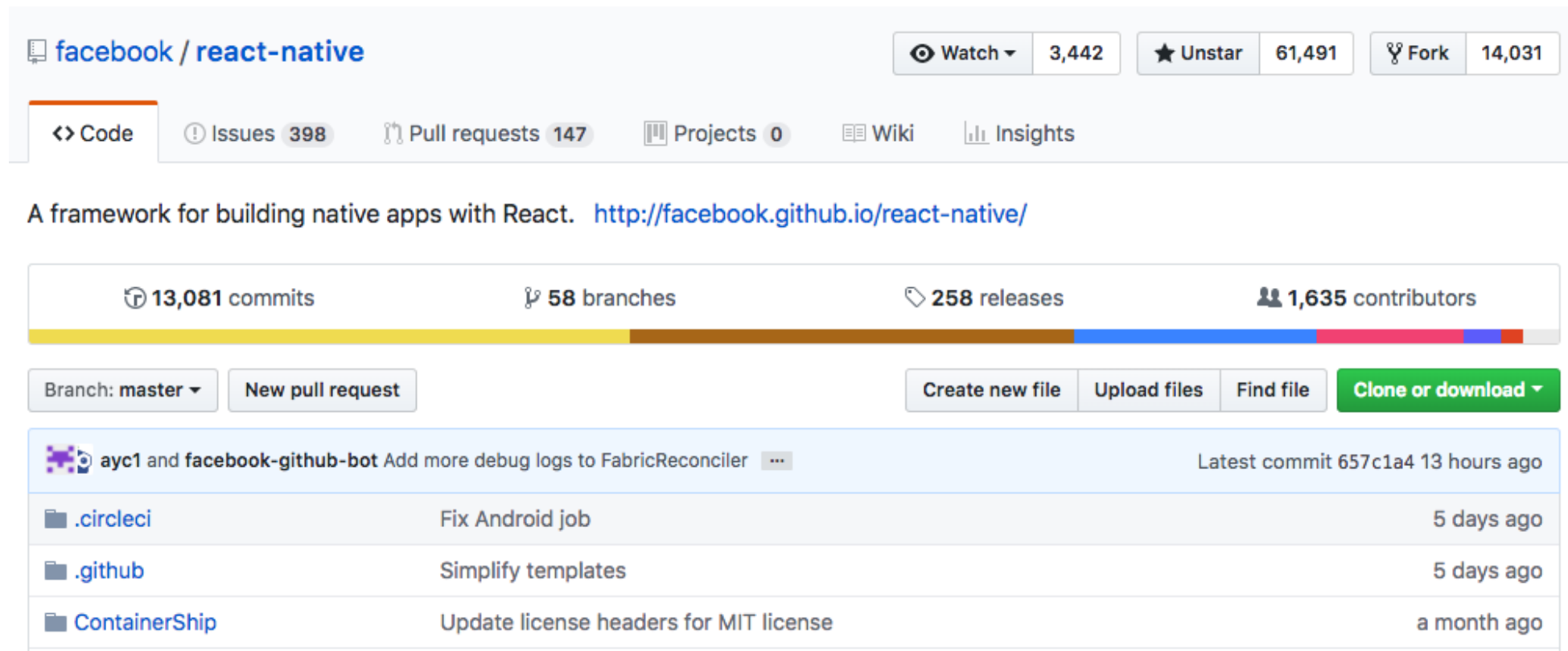
If you need an answer right away, check out the [Reactiflux Discord](#) community. There are usually a number of React Native experts there who can help out or point you to somewhere you might want to look.

### Twitter

For the latest news about React Native, [follow @reactnative](#) on Twitter.

# INTRODUÇÃO AO REACT NATIVE

- Desenvolvimento constante



The screenshot shows the GitHub repository page for `facebook/react-native`. At the top, it displays the repository name and navigation tabs: `<> Code`, `Issues 398`, `Pull requests 147`, `Projects 0`, `Wiki`, and `Insights`. On the right, there are buttons for `Watch 3,442`, `Unstar 61,491`, and `Fork 14,031`.

Below the navigation tabs, a description reads: "A framework for building native apps with React. <http://facebook.github.io/react-native/>".

A horizontal bar displays repository statistics: `13,081 commits`, `58 branches`, `258 releases`, and `1,635 contributors`.

Below the statistics bar, there are buttons for `Branch: master`, `New pull request`, `Create new file`, `Upload files`, `Find file`, and a green `Clone or download` button.

The commit history section shows the latest commit by `ayc1 and facebook-github-bot` with the message "Add more debug logs to FabricReconciler" and a timestamp of "Latest commit 657c1a4 13 hours ago". Below this, a table lists recent commits:

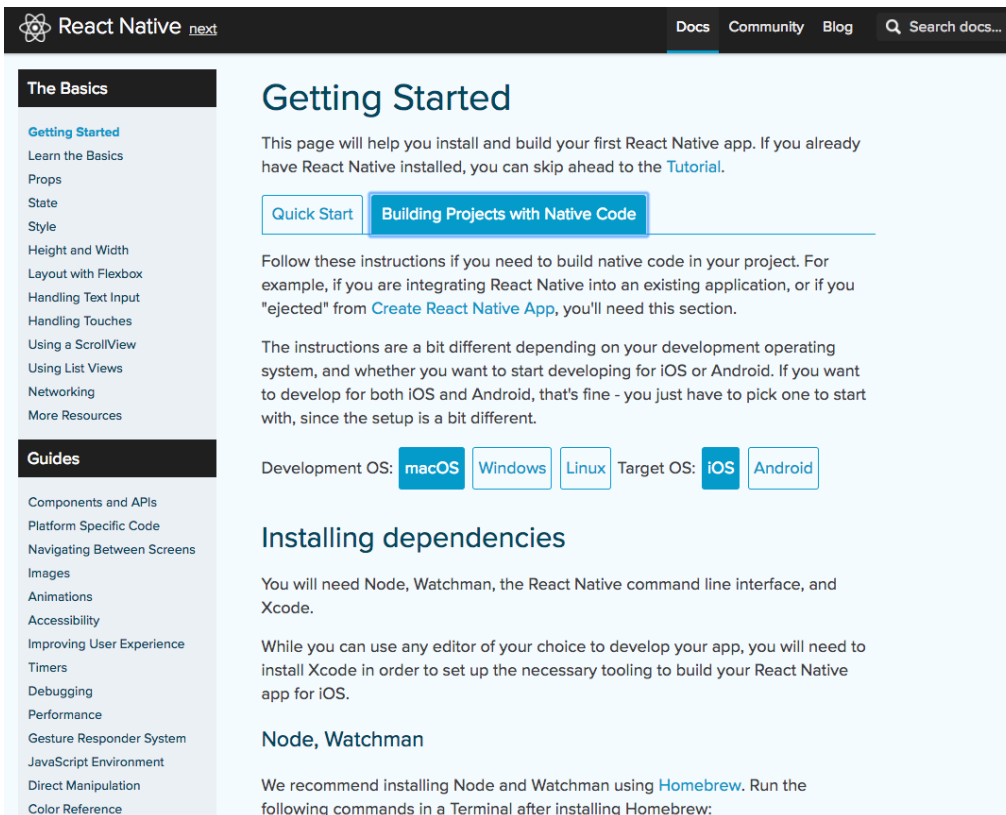
Commit Message	Time Ago
<code>.circleci</code> Fix Android job	5 days ago
<code>.github</code> Simplify templates	5 days ago
<code>ContainerShip</code> Update license headers for MIT license	a month ago

# INSTALAÇÃO E CONFIGURAÇÃO

- Pré-requisitos
  - Node 6.x
  - Python 2.x
  - Java JDK (Open JDK)  $\geq 1.7$
  - Git
  - Android SDK 23.0.1
  - Xcode
  - Windows, Linux ou Mac



# INSTALAÇÃO E CONFIGURAÇÃO



The screenshot shows the React Native documentation website. The header includes the React Native logo, the word 'next', and navigation links for 'Docs', 'Community', and 'Blog'. A search bar is also present. The left sidebar is divided into 'The Basics' and 'Guides' sections. 'The Basics' includes links like 'Getting Started', 'Learn the Basics', 'Props', 'State', 'Style', 'Height and Width', 'Layout with Flexbox', 'Handling Text Input', 'Handling Touches', 'Using a ScrollView', 'Using List Views', 'Networking', and 'More Resources'. 'Guides' includes links like 'Components and APIs', 'Platform Specific Code', 'Navigating Between Screens', 'Images', 'Animations', 'Accessibility', 'Improving User Experience', 'Timers', 'Debugging', 'Performance', 'Gesture Responder System', 'JavaScript Environment', 'Direct Manipulation', and 'Color Reference'. The main content area is titled 'Getting Started' and contains text about installing and building a first React Native app. It includes a 'Quick Start' button and a 'Building Projects with Native Code' button. Below this, there are instructions for development and target OSes, with buttons for 'macOS', 'Windows', 'Linux', 'iOS', and 'Android'. The section 'Installing dependencies' follows, discussing the need for Node, Watchman, and Xcode. The 'Node, Watchman' section recommends using Homebrew for installation.

- Acesse a página de documentação, selecione seu sistema operacional e a plataforma desejada.



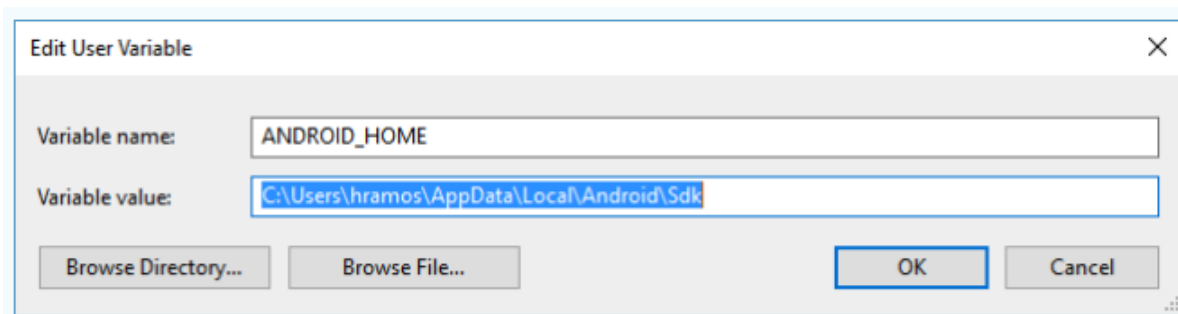
# INSTALAÇÃO E CONFIGURAÇÃO

- Em todas as plataformas é preciso criar as variáveis de ambiente que apontam para o diretório do Android SDK, e em seguida adicionar ao PATH.

- Linux ou Mac

```
export ANDROID_HOME=$HOME/Library/Android/sdk
export PATH=$PATH:$ANDROID_HOME/tools
export PATH=$PATH:$ANDROID_HOME/platform-tools
```

- Windows

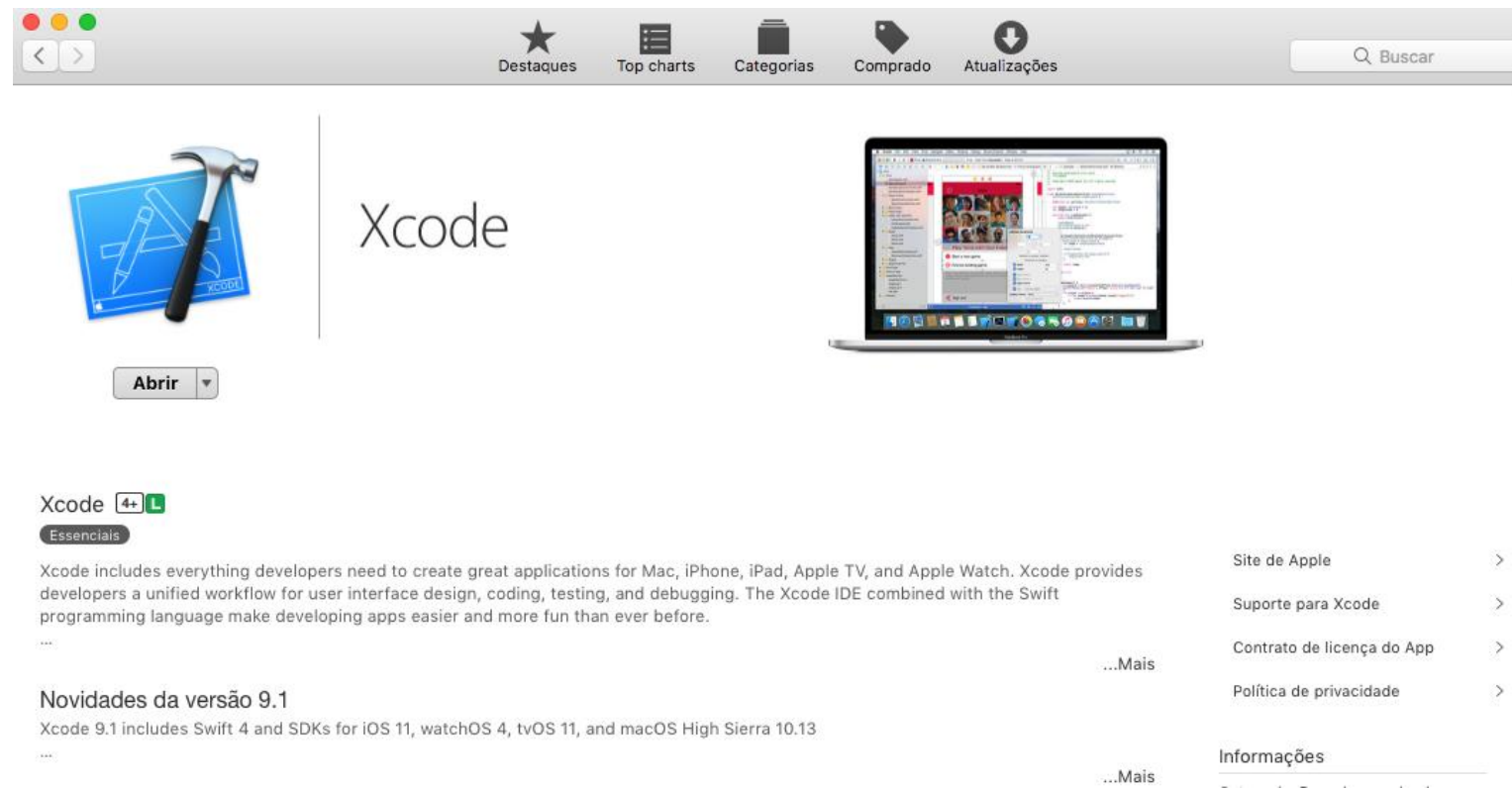


# INSTALAÇÃO E CONFIGURAÇÃO

- No Linux pode ser necessário instalar alguns pacotes 32 bits para compilar para Android.
- Instruções do site do Android Studio.
  - <https://developer.android.com/studio/install.html>
- Stack Overflow
  - <https://stackoverflow.com/questions/2710499/android-sdk-on-a-64-bit-linux-machine>

# INSTALAÇÃO E CONFIGURAÇÃO

- Para instalar o Xcode, basta acessar a loja de aplicativos do Mac OSX.



# CRIANDO PROJETOS

- Antes de criar o projeto, recomenda-se, utilizar o Yarn no lugar do NPM.

```
$ npm install -g yarn
```

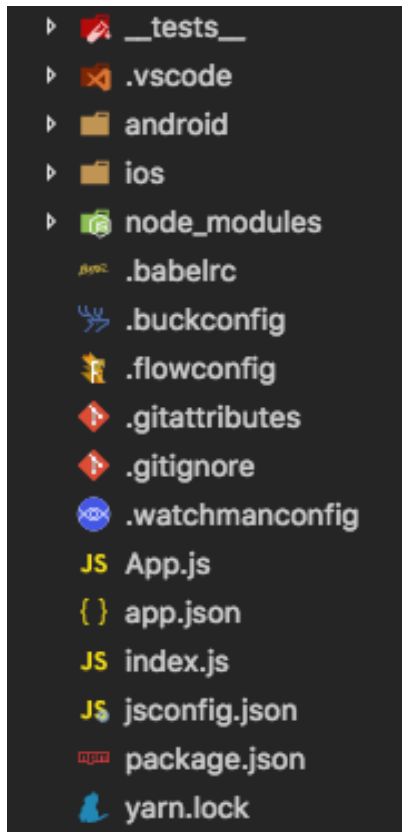
- Para facilitar a criação dos projetos, você pode usar o cliente de linha de comando **react-native-cli**.

```
$ npm install -g react-native-cli
```

- Após todo ambiente instalado e configurado, você pode criar seu projeto utilizando o comando.

```
$ react-native init PrimeiroProjeto
```

# CRIANDO PROJETOS



- /android – projeto nativo do Android Studio
- /ios – projeto nativo do Xcode
- index.js – ponto de partida da aplicação
- App.js – componente inicial da aplicação

# EXECUTANDO O PROJETO ANDROID

- Antes de executar o projeto a primeira vez, recomenda-se abri-lo com o Android Studio. Este processo vai garantir que todas as dependências necessárias estão instaladas, e se não estiver, o Android Studio irá lhe guiar para a instalação.
- Para executar o projeto Android, basta ter aberto um emulador ou dispositivo plugado ao USB, e então:

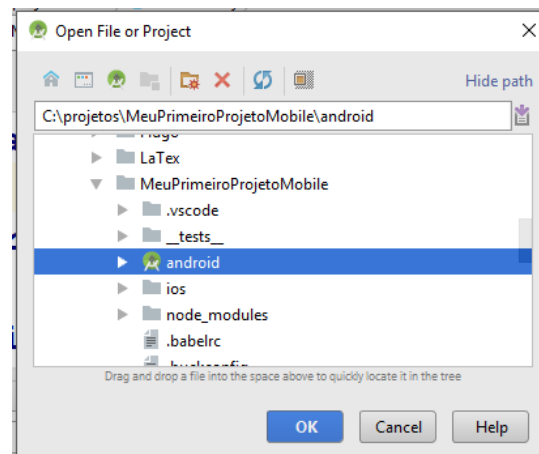
```
$ react-native run-android
```

- **Caso o packager não inicie automaticamente, execute:**

```
$ yarn start
```

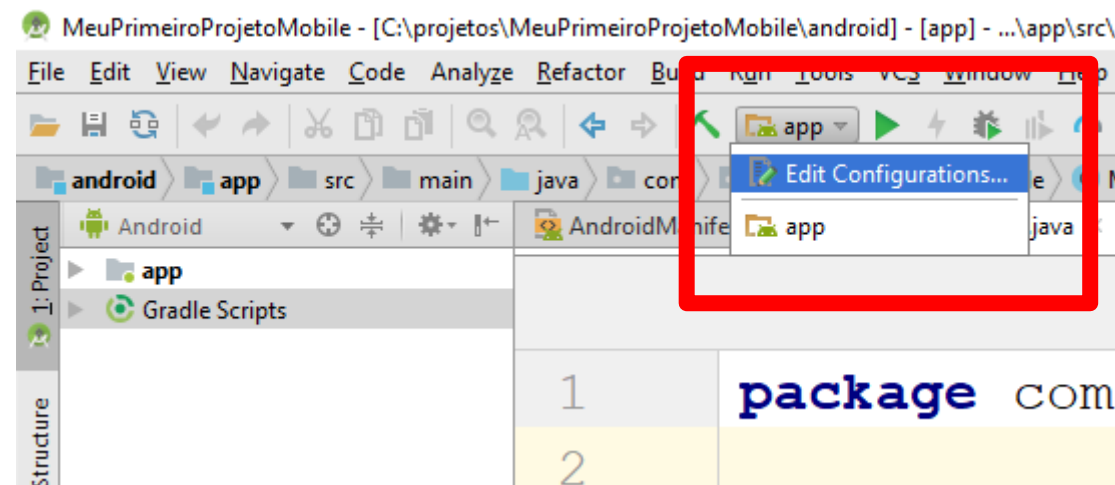
# EXECUTANDO O PROJETO ANDROID

- Para executar o app direto do **Android Studio** em um dispositivo real, é preciso adicionar um comando extra do **ADB** para que o app consiga se conectar ao packager.
- Primeiro é preciso abrir o projeto com o Android Studio, para isso você deve ir no menu **File -> Open** e selecionar apenas a pasta '**android**' que está dentro do projeto que você criou.



# EXECUTANDO O PROJETO ANDROID

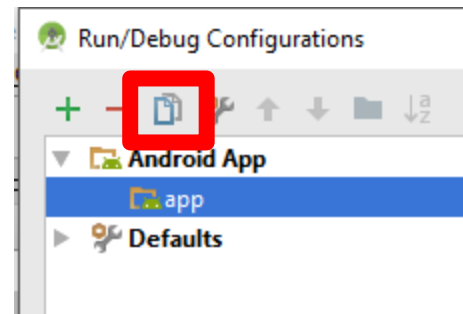
- Depois que o projeto foi aberto e o Gradle terminou a indexação dos arquivos, você deve criar um novo **'profile'** de execução do projeto.
- Para isso clique no menu suspenso **'app'** -> **Edit Configurations.**





# EXECUTANDO O PROJETO ANDROID

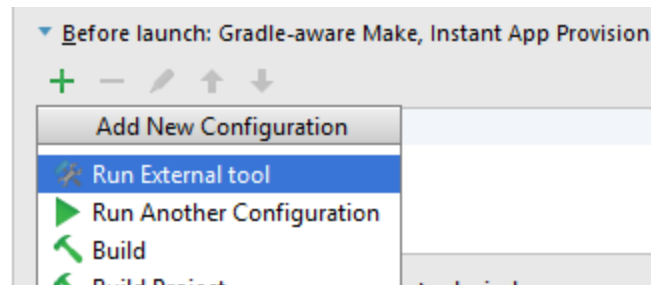
- Você pode duplicar o perfil já existente clicando no ícone **duplicate**.



- O novo perfil você pode chamar de **'react-native'**

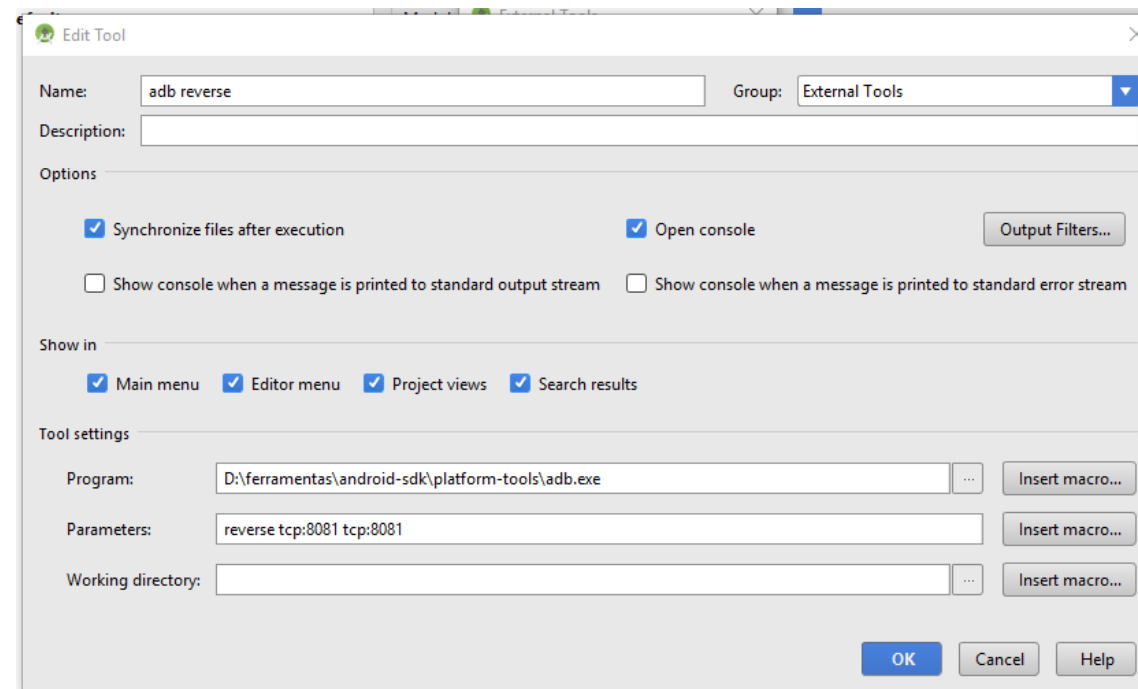
# EXECUTANDO O PROJETO ANDROID

- No perfil 'react-native' você deve adicionar mais uma etapa na seção “**Before launch**”, que são os scripts executados pelo Android Studio antes do aplicativo rodar. **Clique em + para adicionar um novo script -> Run External Tools.**



# EXECUTANDO O PROJETO ANDROID

- Preencha o **Name**, **Program** e **Parameters** do script conforme a imagem abaixo:



- Certifique-se de que **Program** aponta para o caminho correto do executável do ADB.

# EXECUTANDO O PROJETO IOS

- O projeto iOS pode ser executado diretamente pelo Xcode, ou através do comando:

```
$ react-native run-ios
```

- **Caso o packager não inicie automaticamente, execute:**

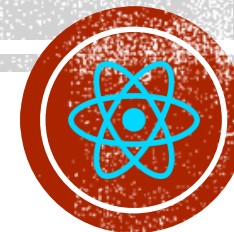
```
$ yarn start
```

**VAMOS EXPLORAR O AMBIENTE**

# REFERÊNCIAS

- React Native - <https://facebook.github.io/react-native/>
- Showcase - <https://facebook.github.io/react-native/showcase.html>
- Documentação - <https://facebook.github.io/react-native/docs/getting-started.html>

# DÚVIDAS?



**Douglas Nassif Roma Junior**

 /douglasjunior

 /in/douglasjunior

 douglasjunior.me

 nassifroma@gmail.com

Slides: <https://git.io/vbU3N>