



Ambiente React Native

Desenvolva apps para iOS e Android
com interface nativa.

1. Instale o Node.js

- Baixe a versão LTS (recomendada): <https://nodejs.org>
- Após instalar, verifique:

```
node -v
```

```
npm -v
```

2. Instale o Expo CLI

```
npm install -g expo-cli
```

Extensões recomendadas no VS Code

Instale estas extensões:

- **React Native Tools** (da Microsoft)

O que faz:

- Permite **depurar** apps React Native direto no VS Code.
- Roda e controla o app com comandos integrados (start, build, attach debugger etc).
- Mostra logs e ajuda com mensagens do Metro bundler.

Exemplo:

Você consegue **iniciar o app ou inspecionar erros** direto pelo menu do VS Code, sem precisar alternar com o terminal.

- **ESLint**

O que faz:

- Analisa seu código em tempo real e **avisa quando tem erro ou código "feio"**.
- Fornece sugestões baseadas em regras de boas práticas de código JavaScript/React.

Exemplo:

Esqueceu de colocar ;, usou uma variável sem declarar, nomeou algo fora do padrão? Ele já te dá um toque — direto no código.

• Prettier - Code formatter

O que faz:

- **Formata automaticamente** seu código (indentação, aspas, quebras de linha, etc).
- Trabalha junto com o ESLint para manter o código bonito e padronizado.

💡 Exemplo:

Você digita o código meio bagunçado, salva o arquivo... e boom 💣, o código se arruma sozinho.

```
const a="Hello"
```

Vira:

```
const a = "Hello";
```

• React Native Snippets (atalhos úteis)

O que faz:

- Fornece **atalhos de código prontos (snippets)** pra você digitar mais rápido.
- Inclui estruturas prontas para componentes, hooks, estilos, etc.

💡 Exemplo:

Você digita rnc e ele gera:

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

const ComponentName = () => {
  return (
    <View>
      <Text>ComponentName</Text>
    </View>
  );
};

export default ComponentName;
```

Dica:

Alguns snippets úteis:

- rnc: componente funcional básico
- usf: useState + função
- rnstyle: estrutura de StyleSheet

Criando o Projeto React Native com Expo

Abra o terminal e execute:

```
npx create-expo-app MeuPrimeiroApp
```

Espere instalar os pacotes. Depois:

```
npx expo start
```

O que você vai ver na tela (por padrão)

Na tela do celular vai aparecer algo como:

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

Ou seja, o “Hello World” já está lá — meio genérico, mas funcional.

Hello World no React Native

Abra o projeto no VS Code. No arquivo App.js, troque o conteúdo por:

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

export default function App() {
  return (
    <View style={styles.container}>
      <Text style={styles.text}>Olá mundo! 🚀</Text>
    </View>
  );
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
  text: {
    fontSize: 24,
  },
});
```

Modo via USB (com Debug e adb)

1. Ative o "**modo desenvolvedor**" no celular.
2. Ative "**Depuração USB**" nas opções de desenvolvedor.
3. Instale o **Android SDK + ADB** corretamente no PC.
4. Conecte o celular via USB e aceite a permissão no celular.
5. Rode:

`npx react-native run-android`