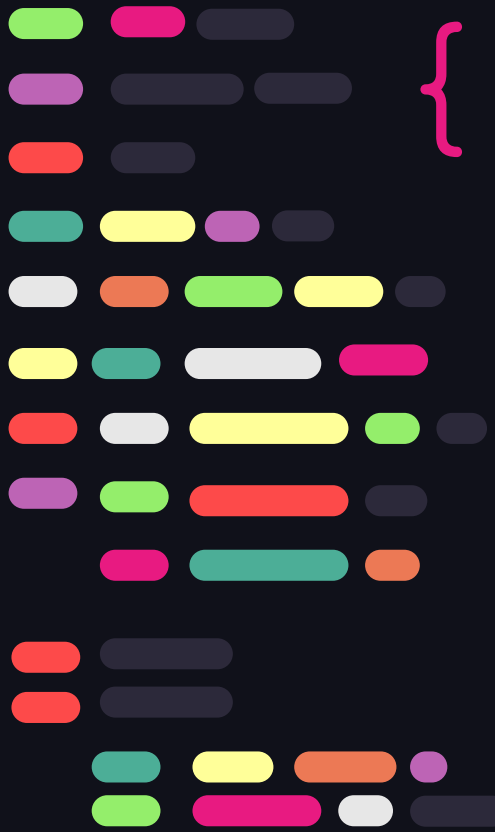
CENTRO UNIVERSITÁRIO
Integrado

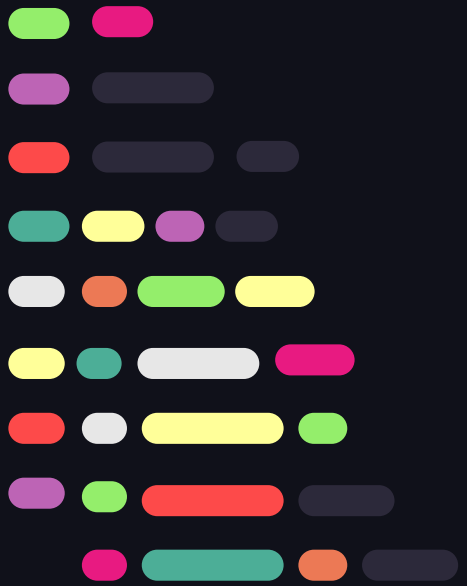
Aula 1 - 25/07



Desenvolvimento mobile com React Native

Profa: Claudia Poiet Sampedro

Conteúdo



01 { ..

Introdução

Primeiramente, vamos entender o que é o React Native

} ..

Introdução

React Native é um framework de desenvolvimento móvel que permite criar aplicativos nativos para iOS e Android usando JavaScript e a biblioteca React.

Ele foi desenvolvido pela empresa Meta (Facebook) e é amplamente utilizado na indústria para criar aplicativos móveis de alto desempenho.

Introdução

A principal característica do React Native é a capacidade de criar aplicativos nativos usando uma única base de código em JavaScript.

Isso significa que os desenvolvedores podem escrever a lógica do aplicativo em JavaScript e, em seguida, o React Native traduz esse código em componentes nativos específicos para cada plataforma. Isso permite reutilizar a maior parte do código entre as plataformas iOS e Android, economizando tempo e esforço de desenvolvimento.

Introdução

Ao contrário dos aplicativos híbridos, que são construídos em torno de uma WebView, os aplicativos React Native são construídos usando componentes nativos que são renderizados diretamente na interface do usuário.

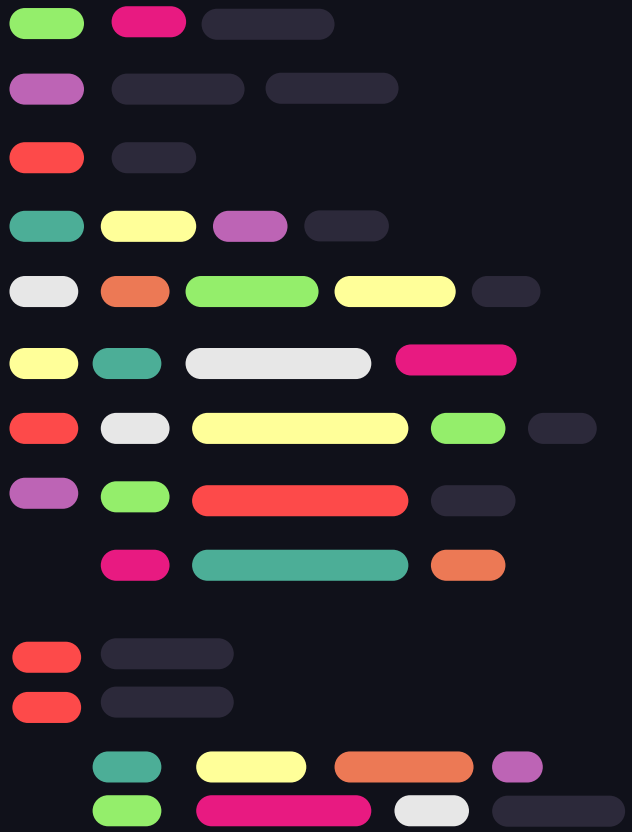
Isso resulta em um desempenho muito melhor, pois os aplicativos React Native têm acesso direto aos recursos nativos do dispositivo, como a câmera, sensores e notificações.

Introdução

React Native também possui um recurso chamado "Hot Reloading", que permite atualizar o código em tempo real enquanto o aplicativo está sendo executado. Isso acelera o processo de desenvolvimento, permitindo que os desenvolvedores vejam imediatamente as alterações feitas no código, sem a necessidade de reiniciar o aplicativo.

Introdução

Além disso, React Native possui uma grande comunidade de desenvolvedores ativa, o que significa que há uma ampla variedade de bibliotecas e componentes disponíveis para acelerar o desenvolvimento e adicionar recursos extras aos aplicativos.



Mas... }

Qual a diferença do React Native pro Ionic, ou para o desenvolvimento Android/iOS nativo?

*

02 { ..

Arquitetura

Como funciona na prática?

} ..

Arquitetura

A arquitetura do React Native é composta por três principais componentes: JavaScript Thread, Bridge (ponte) e Camada de Abstração de Plataforma.

JavaScript Thread

A JavaScript Thread é onde o código JavaScript do aplicativo é executado. Ela contém a lógica de negócio e a renderização dos componentes da interface do usuário.

Essa thread é executada separadamente das threads nativas do dispositivo e é responsável por lidar com todas as operações relacionadas ao JavaScript, como manipulação de estados, atualizações de interface e comunicação com a Bridge.

Bridge (ponte)

A Bridge é responsável pela comunicação bidirecional entre o código JavaScript e o código nativo do dispositivo. Ela atua como uma camada de abstração que permite que o código JavaScript acesse os recursos nativos, como a câmera, GPS, notificações, entre outros.

A Bridge serializa as chamadas de função e os dados entre o JavaScript Thread e as threads nativas, garantindo a comunicação eficiente entre as duas partes.

Camada de Abstração de Plataforma

A Camada de Abstração de Plataforma é uma coleção de APIs JavaScript fornecidas pelo React Native. Ela oferece um conjunto de componentes e módulos que encapsulam os recursos nativos específicos da plataforma, como botões, imagens, mapas, entre outros.

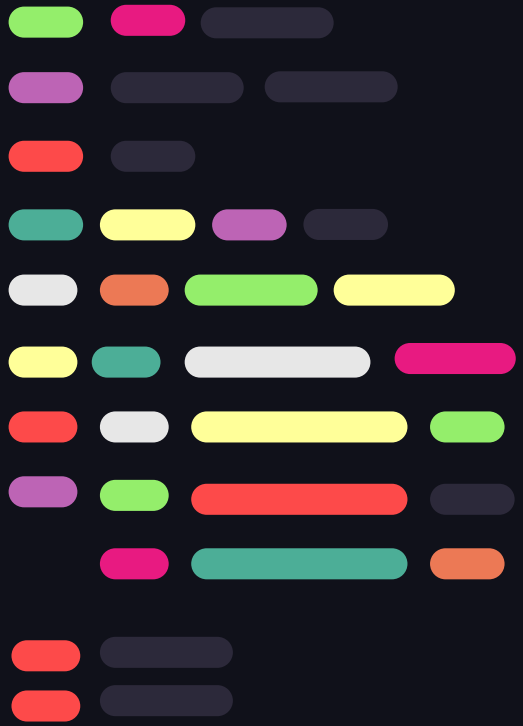
Esses componentes são renderizados como componentes nativos reais em cada plataforma, fornecendo uma experiência nativa para os usuários.

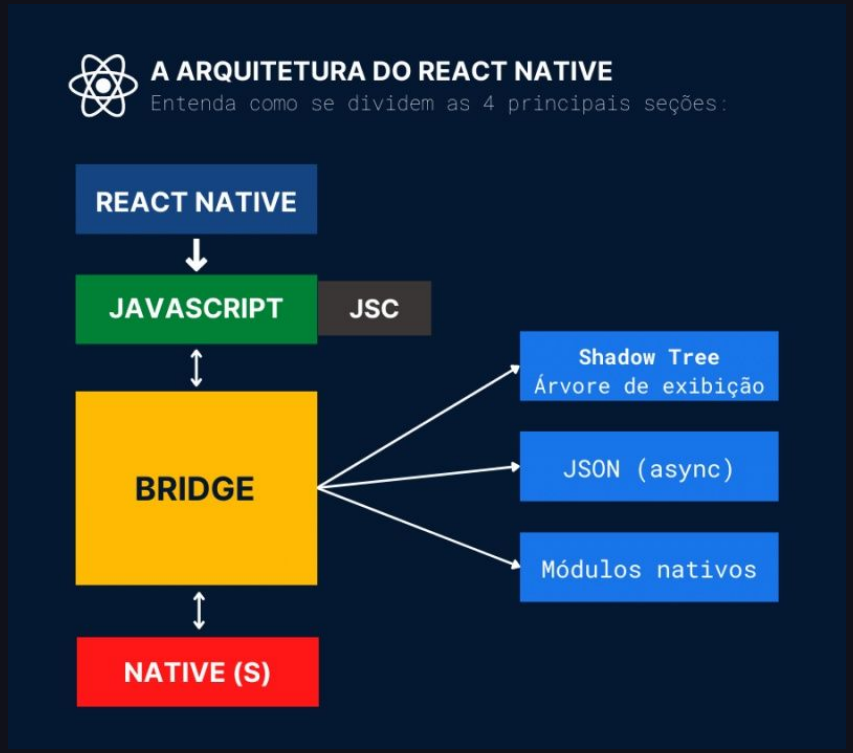
Camada de Abstração de Plataforma

Ao utilizar a Camada de Abstração de Plataforma, os desenvolvedores podem escrever uma única base de código em JavaScript que é executada em ambas as plataformas (iOS e Android), reutilizando grande parte do código entre elas.

A renderização dos componentes é realizada de forma nativa, garantindo um desempenho e experiência do usuário semelhantes aos aplicativos nativos tradicionais.

Arquitetura





03 { ..

Setup

} ..

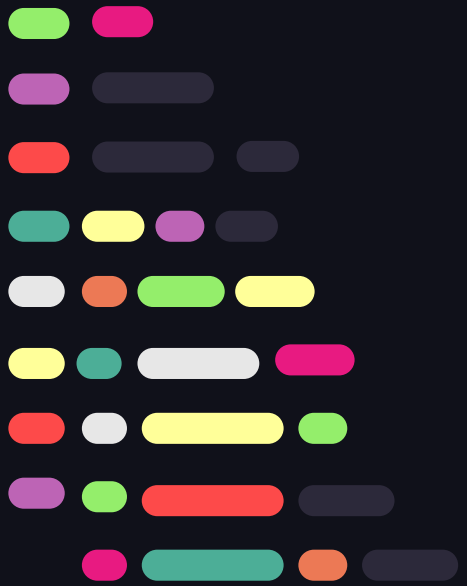
Setup

RN CLI

- Maior controle de módulos nativos
- Mais complexo
- Praticamente 100% costumizável
- Compatibilidade com mais ecossistemas]
- Build manual
- Questões com o iOS

Expo

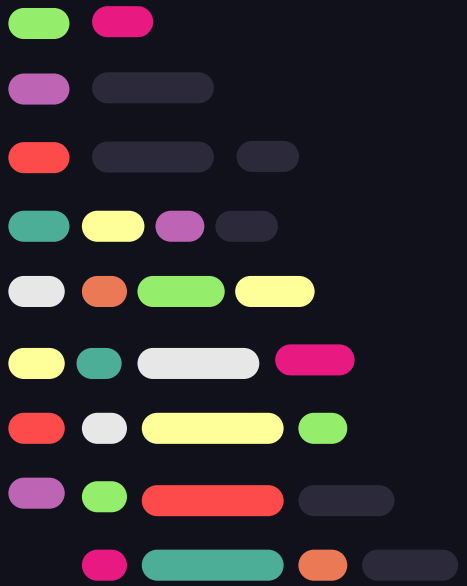
- Mais simples
- Over-the-Air Updates
- Bibliotecas pré-prontas
- Processo de build simplificado
- Módulos nativos limitados
- Menos controle da bridge
- Apps mais pesados



CLI:

https://reactnative.dev/docs/environment-setup?guide=native

https://reactnative.dev/docs/set-up-your-environment



Expo:

```
npx create-expo-app@latest
npx expo start
```

04 { ..

TypeScript

} ..

Conceito

TypeScript é uma linguagem de programação de código aberto desenvolvida pela Microsoft. É um superset do JavaScript, o que significa que estende o JavaScript adicionando recursos adicionais, como tipagem estática.

Isso permite que os desenvolvedores especifiquem explicitamente os tipos de variáveis, funções, e outros elementos, ajudando a identificar erros de tipo em tempo de compilação, em vez de tempo de execução.

Características

- Tipagem estática
- Compilação para o JS
- Ferramentas para desenvolvimento mais robustas
- Interfaces e Enums
- Verificação estrita
- Grande suporte da comunidade

Exemplos

- Tipagem básica

```
const isDone: boolean = false;
const decimal: number = 6;
const color: string = "blue";
const list: number[] = [1, 2, 3];
```

Exemplos

- Funções

```
function add(x: number, y: number): number {
    return x + y;
}

let result = add(5, 3);
```

Exemplos

- Interface/Type

```typescript
interface Person {
    firstName: string;
    lastName: string;
    age?: number;
}

function greet(person: Person): string {
    return `Hello, ${person.firstName} ${person.lastName}`;
}

let user = { firstName: "John", lastName: "Doe" };
```

Exemplos

- Genéricos

```
function identity<T>(arg: T): T {
    return arg;
}

let output1 = identity<string>("myString");
let output2 = identity<number>(42);
```

Exemplos

- Enums

```
enum Color {
    Red,
    Green,
    Blue
}

let c: Color = Color.Green;
```