

# Desenvolvimento de Aplicativo Mobile

Material base desenvolvido pelo  
Prof. Raphael Barreto

---

[raphael.b.oliveira@docente.senai.br](mailto:raphael.b.oliveira@docente.senai.br)

# O Que É um Aplicativo?

## Aplicativos

Programas instalados diretamente no dispositivo móvel, baixados de lojas como App Store ou Google Play. Eles acessam recursos do celular e funcionam de forma independente.

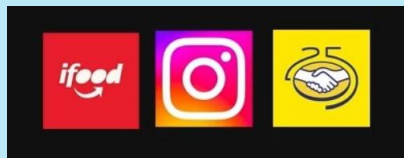
- Instalados no dispositivo
- Acesso direto aos recursos
- Funcionam offline (muitos)
- Performance otimizada

## Sites

Executam dentro de navegadores como Chrome, Safari ou Firefox. Você acessa digitando um endereço web, sem precisar instalar nada no dispositivo.

- Rodam no navegador
- Acesso via URL
- Requerem internet
- Multiplataforma por padrão

📄 Exemplos populares como iFood, Instagram e Mercado Livre têm tanto aplicativo quanto site, oferecendo experiências diferentes para cada plataforma.



# Dois Mundos, Dois Sistemas



Os dispositivos móveis funcionam com sistemas operacionais distintos, cada um com suas próprias regras e linguagens de desenvolvimento.

## iOS - Apple

Sistema operacional exclusivo da Apple, usado em iPhones e iPads. Conhecido por seu design elegante e ecossistema fechado.

- iPhone
- iPad
- Interface consistente

## Android - Google

Sistema operacional de código aberto usado por diversos fabricantes como Samsung, Motorola, LG e muitos outros ao redor do mundo.

- Samsung
- Motorola
- LG e outros

# Linguagens de Programação para Cada Sistema

Desenvolver aplicativos nativos significa usar linguagens específicas para cada sistema operacional. Isso garante máxima performance, mas duplica o trabalho.

## Desenvolvimento iOS

- **Objective-C:** Linguagem mais antiga, ainda presente em muitos projetos legados
- **Swift:** Linguagem moderna da Apple, mais intuitiva e segura
- **XML:** Para criar layouts e interfaces visuais

## Desenvolvimento Android

- **Java:** Linguagem tradicional do Android, usada desde o início
- **Kotlin:** Linguagem moderna, mais concisa e produtiva
- **XML:** Para estruturar a interface do usuário

# O Desafio do Desenvolvimento Nativo

Imagine que você está criando o Instagram. Para alcançar todos os usuários, você precisa desenvolver duas versões completamente separadas do mesmo aplicativo.



## Equipe iOS

Desenvolvedores especializados em Swift/Objective-C criando a versão para Apple



## Equipe Android

Desenvolvedores especializados em Java/Kotlin criando a versão para Android

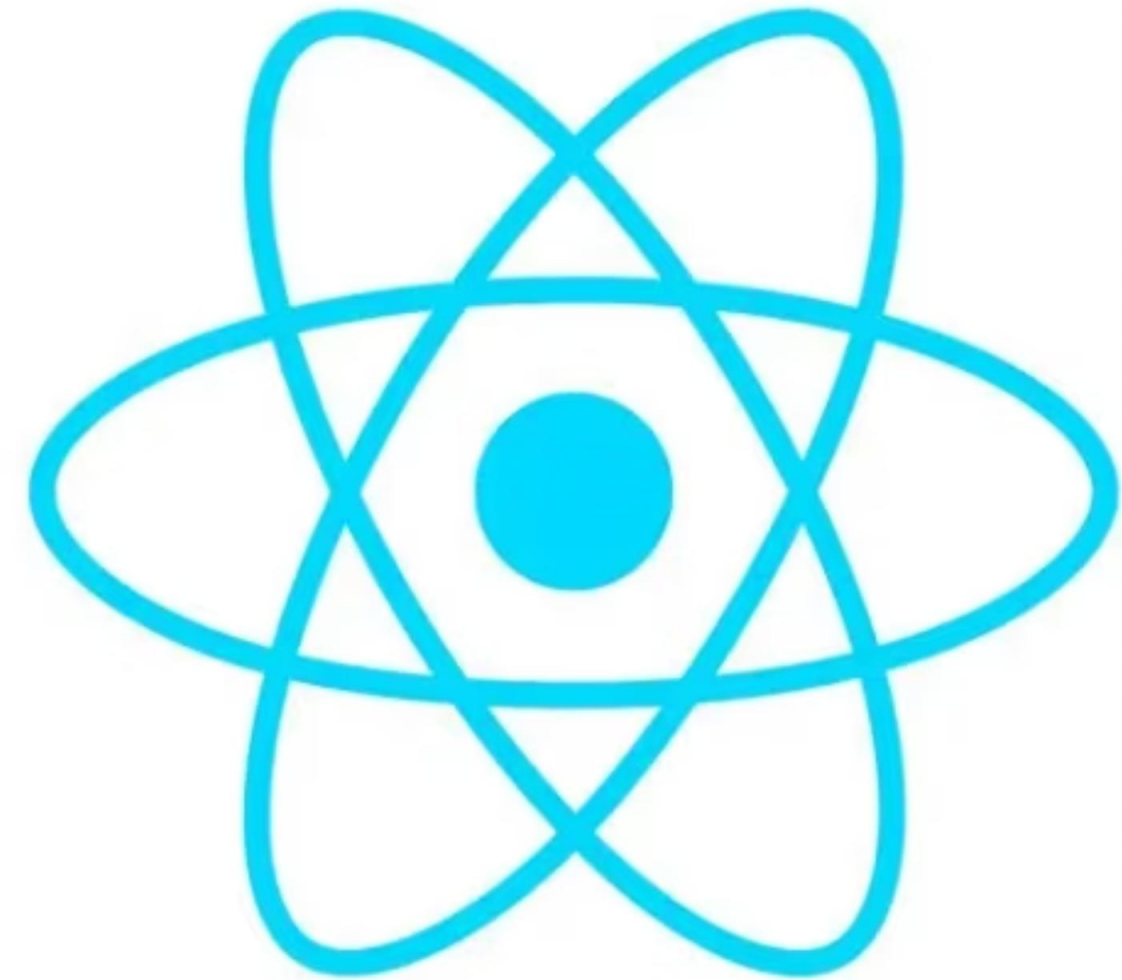


## Dois Lançamentos

Dobro do tempo, dobro do custo, duas bases de código para manter

- ❏ No desenvolvimento nativo, cada funcionalidade precisa ser implementada duas vezes, em linguagens diferentes, por equipes diferentes.





# A Solução: Desenvolvimento Híbrido

E se existisse uma forma de escrever o código uma única vez e ele funcionasse em ambas as plataformas? Essa é a promessa do desenvolvimento híbrido.

1

## O Problema

Duas equipes, duas linguagens, dobro de tempo e custo para manter os aplicativos

2

## A Ideia

Usar JavaScript, linguagem que desenvolvedores web já conhecem bem

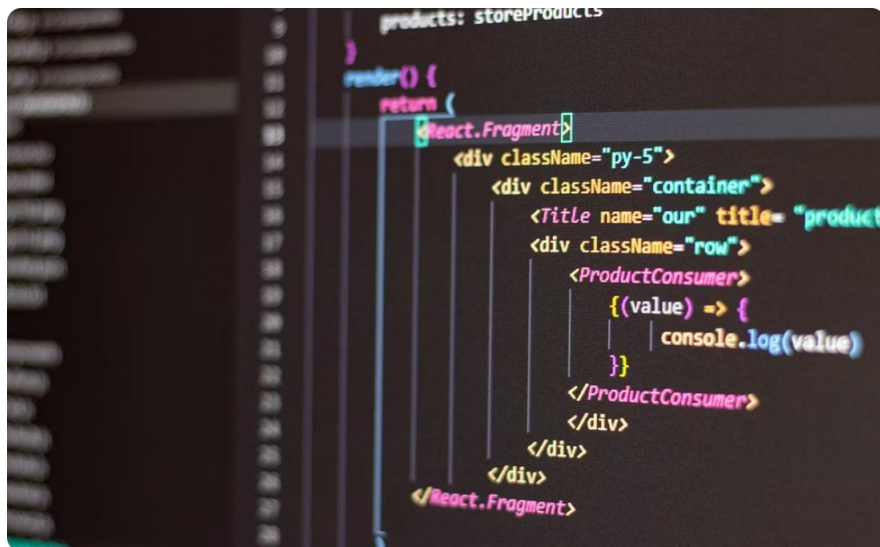
3

## O Benefício

Mesma equipe desenvolve web, back-end e mobile com uma base de código única

# React Native: O Melhor dos Dois Mundos

O React Native não é a única opção de desenvolvimento híbrido, mas se destaca por sua proximidade com React, framework já dominado por milhões de desenvolvedores.



## Conceitos Familiares

Componentes, estados, props, ciclo de vida



## Multiplataforma

Escreva uma vez, rode em iOS e Android sem duplicação de código



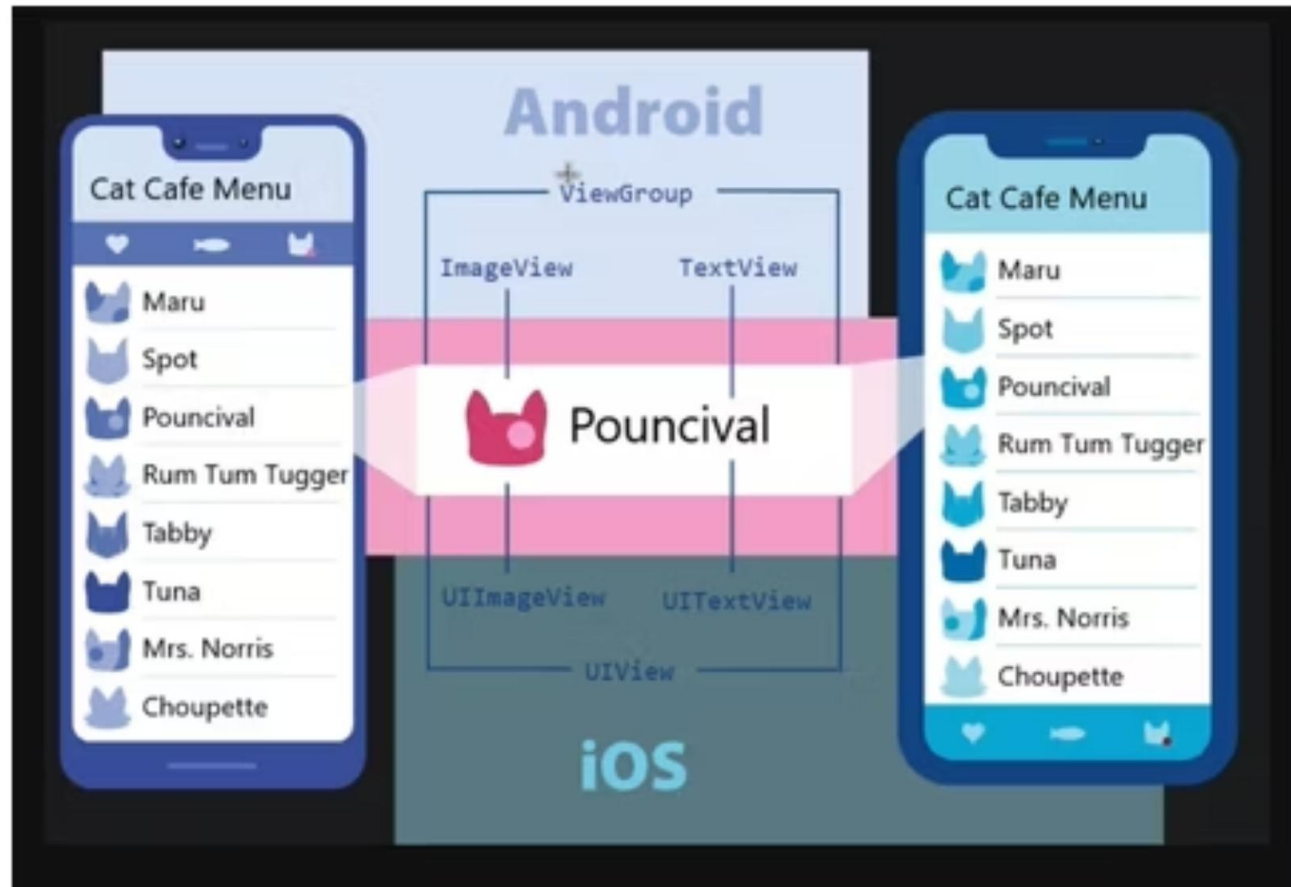
## Equipe Unificada

Desenvolvedores front-end e back-end podem trabalhar no mobile também

Além do React Native, outras opções como Flutter também permitem desenvolvimento híbrido, mas a familiaridade com React torna o React Native especialmente atraente.

# Comparando Componentes Nativos

Cada sistema operacional tem seus próprios componentes para construir interfaces. Veja como uma lista simples é implementada de formas diferentes no Android e iOS.



## Android

- **ViewPager:** Agrupa elementos (como uma `div` no HTML)
- **ImageView:** Exibe imagens
- **TextView:** Mostra texto
- Menu superior na interface

## iOS

- **UIView:** Container básico de elementos
- **UIImageView:** Componente de imagem
- **UITextView:** Exibição de texto
- Menu inferior na interface

❏ Sistemas operacionais diferentes têm padrões de design distintos. Por exemplo, Android geralmente usa menus superiores enquanto iOS prefere menus inferiores.



# Como o React Native Funciona?

Por trás da magia de escrever JavaScript e executar código nativo, existe uma arquitetura sofisticada com várias camadas trabalhando em conjunto.

## 1 Metro Bundler + Babel

Empacota seu código JavaScript, similar ao Webpack no React web. Permite hot reload para ver mudanças instantaneamente.

Babel Converte TypeScript para JavaScript e transforma JSX em código válido. Garante compatibilidade entre diferentes versões.

## 2 JavaScript Core

Engine que interpreta JavaScript no dispositivo. Já vem no iOS (do Safari) e é incluído nos apps Android.

## 3 Bridge (Ponte)

Traduz comandos JavaScript para APIs nativas do iOS e Android, convertendo componentes automaticamente.

# Metro Bundler: O Empacotador

## 1 Empacotamento

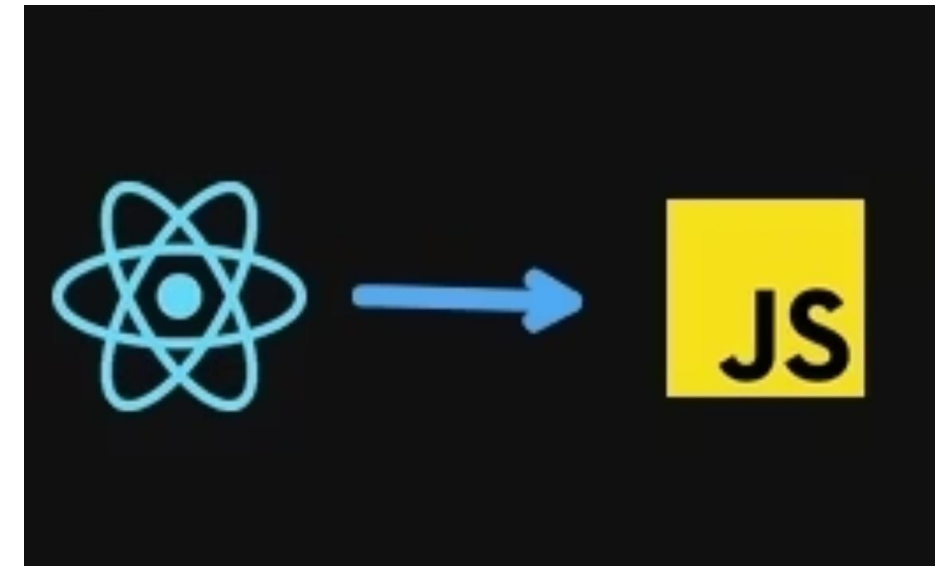
Reúne todos os seus arquivos JavaScript em um único pacote otimizado para o dispositivo móvel.

## 2 Minificação

Reduz o tamanho do código removendo espaços e otimizando a estrutura para melhor performance.

## 3 Hot Reload

Atualiza o aplicativo instantaneamente enquanto você programa, sem perder o estado atual da aplicação.



# JavaScript Core: A Engine de Execução

Para que o código JavaScript funcione nos dispositivos móveis, é necessária uma engine que interprete a linguagem. O JavaScript Core cumpre esse papel fundamental.

## No iOS

Já vem instalado por padrão, pois é a mesma engine usada pelo navegador Safari. Isso torna os aplicativos iOS naturalmente mais leves.

## No Android

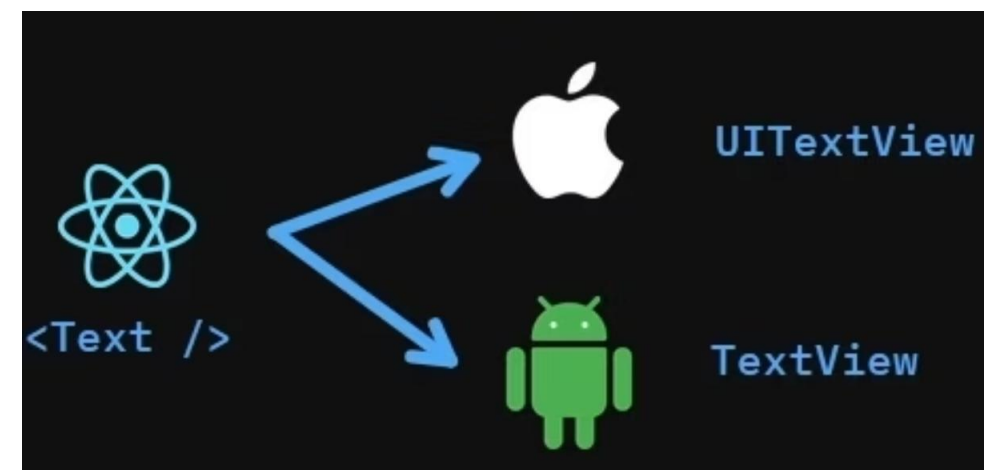
Precisa ser incluído no pacote do aplicativo, já que Android não tem JavaScript Core nativo. Isso explica por que apps React Native no Android costumam ser ligeiramente maiores.



📌 Lembre-se: Java e JavaScript são linguagens completamente diferentes, apesar da semelhança no nome!

# Bridge: Conectando JavaScript e Código Nativo

A Bridge (ponte) é o coração do React Native. Ela traduz os componentes que você escreve em JavaScript para os componentes nativos correspondentes de cada plataforma.



Esse processo acontece automaticamente para todos os componentes, permitindo que você escreva código uma única vez e ele seja renderizado nativamente em ambas as plataformas.




# Vantagens do React Native

 **Desenvolvimento Mais Rápido**  
Uma única base de código para iOS e Android reduz drasticamente o tempo de desenvolvimento e permite lançamentos mais ágeis.

 **Equipe Unificada**  
Desenvolvedores que já conhecem React podem trabalhar no mobile imediatamente, sem precisar aprender linguagens totalmente novas.

 **Redução de Custos**  
Menor necessidade de especialistas em múltiplas plataformas significa economia significativa em contratações e manutenção.

 **Hot Reload**  
Veja suas alterações instantaneamente durante o desenvolvimento, acelerando ciclos de feedback e iteração.

 **Comunidade Forte**  
Grande ecossistema de bibliotecas, ferramentas e suporte da comunidade React, facilitando resolução de problemas.

 **Reutilização de Código**  
Componentes e lógica de negócio podem ser compartilhados entre projetos web e mobile, maximizando eficiência.

# O Fluxo Completo de Desenvolvimento

Agora que você entende todos os conceitos fundamentais, vamos visualizar o processo completo de como seu código JavaScript se transforma em um aplicativo nativo funcional.



## Você Escreve JSX

Código React Native com componentes familiares



## Metro Bundler + Babel

Empacotam e convertem seu código



## JavaScript Core

Interpreta o JavaScript no dispositivo



## Bridge Traduz

Converte para componentes nativos



## Renderização Nativa

App funciona nativamente em iOS e Android

developers === people



Chris Heilmann ©

🚀 PRÓXIMOS PASSOS

# Pronto Para Começar a Desenvolver?

Agora você compreende os fundamentos do desenvolvimento de aplicativos e como o React Native funciona por baixo dos panos. Na próxima aula, vamos colocar a mão na massa!

## 1 Setup do Ambiente

Instalação de todas as ferramentas necessárias: Node.js, npm, React Native CLI e emuladores

## 2 Primeiro Projeto

Criação do seu primeiro aplicativo React Native e compreensão da estrutura de pastas

## 3 Rodando o App

Como executar seu aplicativo no emulador e em dispositivos físicos para testar



# Perguntas?

Entre em contato:

[raphael.b.oliveira@docente.senai.br](mailto:raphael.b.oliveira@docente.senai.br)

Obrigado pela atenção! Estou à disposição para esclarecer dúvidas e ajudar no seu aprendizado.



