Ifit - Aplicativo para atletas de crossfit acompanharem sua evolução

**Renan Campos Ribeiro**

Instituto Federal de Educação ciência e tecnologia de São Paulo (IFSP - Campus Hortolândia  
Hortolândia – SP – Brazil

***Abstract.*** *This article covers the React Native and Firebase technologies used to develop a hybrid application for IOS and Android platforms, an app for crossfit athletes who want to track their performance and manage their evolution, documenting their personal records in dozens of exercises of the modality. This article also covers the Extreme Programming development technique and some architecture design techniques and Firebase as Backend*

***Resumo.*** *Este artigo, aborda as tecnologias React Native e Firebase utilizadas para o desenvolvimento de um aplicativo híbrido para as plataformas IOS e Android, app esse destinado a atletas de crossfit que querem acompanhar seu desempenho e gerenciar sua evolução, documentando seus recordes pessoais em dezenas de exercícios da modalidade. Este artigo também aborda a técnica de desenvolvimento Extreme Programming e algumas técnicas de design arquitetura e Firebase as Backend.*

Ifit - Aplicativo para atletas de crossfit acompanharem sua evolução

**Renan Campos Ribeiro**

Instituto Federal de Educação ciência e tecnologia de São Paulo (IFSP - Campus Hortolândia  
Hortolândia – SP – Brazil

**Resumo**

**Palavras-chave:**

**1 Introdução:**

**2 Desenvolvimento :**

**2.1. Metodologia**

**2.2. Aplicativos Multiplataforma**

Aplicativos Multiplataforma são os aplicativos que estão acessíveis nos sistemas operacionais do mercado, esse projeto está disponível em nas plataformas líderes de mercado Android e IOS.  
 Vendas de smartphones no mundo:   
 Android 70,1%.  
 iOS 21%.  
 outros 8.9%

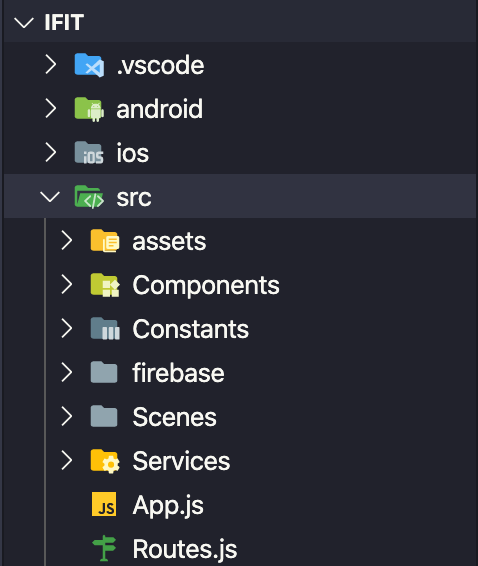
2.2.1 React Native:

React Native é uma biblioteca de desenvolvimento de aplicativos de forma nativa para IOS e Android, criada pelo Facebook e publicada em 26 de março de 2015 durante a ReactJS conf. Hoje essa biblioteca é mantida pelo Facebook e pela comunidade, sendo assim uma plataforma de software aberto com licença MIT.  
 O React Native realiza uma integração entre o JavaScript (ReactJS) e o Nativo (Objective-C e Java), essa integração é responsavel por toda a comunicação entre as plataformas e é conhecida como Bridge.  
 As aplicações rodam nativamente no dispositivo do usuário e contam com as seguintes arquiteturas:   
 Dispositivos Android:   
 - JSCore virtual machine: Destinado ao código em javascript.  
 - Android Runtime: Destinado ao código em Java.

Dispositivos IOS:   
 - JSCore virtual machine: Destinado ao código em javascript.  
 - Native Runtime: Destinado ao código em Objective-C e Swift.

Existem também três principais threads sendo executadas em uma aplicação React Native alem das que ja são executadas pelas aplicações nativas, sendo elas: Main thread, Exec Javascript e Shadow Queue.   
 Main Thread: é utilizada por qualquer aplicação nativa. Ela é responsável por tratar as requisições relacionadas a renderização de elementos na tela e também pelos gestos reproduzidos pelo usuário.  
 Exec Javascript: é exclusiva ao React Native, responsável por executar o código JavaScript. O JavaScript é o responsável por toda lógica de negócio da aplicação. Além de definir a estrutura e as funcionalidades da nossa UI.  
 Shadow Queue: que é a responsável pelos cálculos referentes ao layout.  
 A comunicação entre o Nativo e o Javascript se dá pela bridge da seguinte forma: A Main thread recebe uma informação de click na tela que executa determinada função, essa informação é levada a bridge onde a Thread de execução do Javascript recebe essa informação e devolve uma resposta para a Main thread com o que deve ser renderizado na tela, sendo assim uma comunicação N -> RN -> N. As Threads são independentes e nunca se cruzam, todas as comunicações são feitas através da Bridge.  
 Componentes Nativos e JSX, o ReactJS implementa componentes JSX, que são componentes escritos em Javascript, semelhante ao XML, são traduzidos para componentes Nativos através de um processo compilação transformando um código fonte em Javascript para um código fonte em código Java, Objective-C ou Swift.  
 Concluímos assim React Native é realmente nativa, uma vez que seus componentes são renderizados de forma nativa.

2.2.2 Estrutura do projeto



IFIT - Pasta Raiz do projeto, o projeto inteiro está dentro dela.

.vscode - Pasta onde estão as configurações do editor de texto, não é obrigatória.

Android - Código nativo em Java para os dispositivos Android.

IOS - Código nativo em Objective-C e Swift para os dispositivos IOS.

src - Código Javascript e as configurações do React native e onde a estrutura do código está definida.

assets - Onde estão as imagens e ícones da aplicação.

Components - Aqui estão os componentes do React Native, como botões, menus e listas.

Constants - Todos os objetos com listas de nomes de exercícios que se repetem em vários lugares do projeto, se precisarem ser alterados, basta alterar no arquivo de constantes que serão refletidas em todo o projeto.

firebase - Arquivos de configuração do firebase.

Scenes - Aqui estão as telas do aplicativo.

Services - Aqui está a lógica do aplicativo, todas as chamadas em bancos de dados do firebase e algumas funções utilitárias, como formatação de datas.

App.js - Arquivo de inicialização do aplicativo.

Routes.js - Arquivo onde é mapeada a navegação e os fluxos das telas do aplicativo.

2.2.3 - JSX

2.2.4 - Whitelabel

**2.3. Firebase**

2.3.1 Firebase as BackEnd

**2.4. Resultados**

**2.5. Próximos Passos**

**3. Considerações Finais**

**4. Agradecimentos**

**5. Referencias**

CORE COMPONENTS AND NATIVE COMPONENTS. React Native. Disponível em: https://reactnative.dev/docs/intro-react-native-components. Acesso em: 19 de Fevereiro de 2020.

RASMUSSEN, Ana, OS NÚMEROS NÃO MENTEN: ANDROID OU IOS, QUAL É O MELHOR?. Techno blog. Disponível em: https://tecnoblog.net/247956/referencia-site-abnt-artigos/. Acesso em: 19 de Fevereiro de 2020.