



UNIVERSIDADE VEIGA DE ALMEIDA –
UVA CIÊNCIA DA COMPUTAÇÃO

DESENVOLVIMENTO MOBILE

Daniel do Valle Ribeiro Santos – 1220102081

Daniel Rezende Marçal – 1220105970

Guilherme Suares de Freitas – 1220107251

Gustavo Passos Tascheri – 1220108204

Mauricio de Moraes Coutinho – 1220108478

Sabrina Alves Brito – 1220105863

Susanna Marapodi dos Passos – 1220108872

Vitor Ramon Lemos de Oliveira – 1220103381

RIO DE JANEIRO - RJ

2024

Sumário

1. Pesquisa sobre as Plataformas Mobile: React Native, Flutter, Ionic, Xamarim – (pág 3 a 7)
2. Pesquisa sobre as Linguagens: Swift e Objective- C – (pág 8 a 9)
3. Pesquisa sobre a Linguagem de Programação Kotlin – (10 a 11)
4. Bibliografia – (pág 12)

1. Pesquisa sobre plataformas de desenvolvimento Mobile:

React Native

React Native conhecida como um Framework de código aberto utilizada em diversos setores, um deles sendo aplicado em desenvolvimento mobile. React Native (RN) é uma estrutura de aplicativo móvel popular, baseada em JavaScript, permitindo que aplicativos móveis sejam renderizados nativamente para Android e iOS e outras plataformas. Lançada em 2015 pelo Facebook definida como um projeto de código aberto, e desde então vem crescendo no mercado atual, sendo utilizada em diversos aplicativos, como: Instagram, Facebook, Discord.

As principais razões para o seu sucesso, seria o fato de poder criar o código de forma rápida e eficaz, sem precisar escrever o código novamente, economizando tempo e custo. Segundo motivo seria a herança de ser construída com base em React, uma biblioteca popular do JavaScript e suas semelhanças. E terceira razão seria a nova capacidade de pessoas desenvolvedoras Front-end conseguirem criar aplicativos robustos e destinadas a somente plataforma móveis.

React Native possui como principais características: Desenvolvimento Multiplataforma, sendo uma prática de construção de software compatível a bem mais tipos de plataformas de hardware, ex: Funcionamento em sistemas operacionais diferentes, Windows, Linux, macOS. Outro ponto forte: Uso de recursos nativos, possibilidade de integrar ferramentas específicas da programação nativa em um ambiente híbrido de desenvolvimento. E por último, a Modularização em forma de Componentes, estrutura baseada em componentes, logo, ajuda na modularização do código, e ajudando na abstração de funções complexas do sistema.

Flutter

Flutter é uma ferramenta (Framework) com foco em desenvolvimento multiplataforma em dispositivos móveis, criada originalmente pela Google, é utilizada fortemente nos mercados atuais, geralmente para criações desktop, em Windows / Linux e Mac. Possui código único, logo sendo portado em diferentes plataformas, tendo assim como uma de seus principais benefícios a versatilidade, a baixa curva de aprendizado e sua agilidade.

Flutter utiliza o Dart, uma linguagem otimizada para dispositivos clientes multiplataforma, tendo como objetivo principal a produtividade e flexibilidade para servir como base para bons frameworks de desenvolvimento de aplicativos. Oferece facilidade de desenvolvimento: Hot reload (checagem na alteração feitas no código), consegue rodar templates no navegador, possui uma alta adaptabilidade de código e ainda é constituída por uma Documentação de qualidade.

Vantagens de utilizar Flutter: Flutter utiliza Dart, que é uma Linguagem Orientada a Objetos, abrangendo todo um cenário organizado e detalhado do paradigma da orientação a objetos. Outra vantagem seria uma linguagem não ligada a 100% a tipagem, podendo trabalhar com valores nulos (null safety), fazendo assim Flutter ser a combinação dos pontos das linguagens robustas, como Java, mais uma adaptabilidade milimétrica do JavaScript, possuindo uma boa liberdade do que deseja fazer.

Ionic

Famoso framework open source para desenvolvimento de aplicativos móveis multiplataforma, facilitando a implementação de tecnologias Front-end: Html, CSS e JavaScript nos aplicativos móveis, possuindo um diferencial de adotar como base o Apache Cordova, trazendo recursos simplificados de alto desempenho e elevando um aspecto mais profissional. Tais diferenciados gerados pelas linguagens / frameworks embutidos, provendo uma resolução de alto níveis em códigos e projetos bem-organizados.

Além da utilização do Cordova, deve-se dizer outras possíveis tecnologias utilizadas, como: Typescript, Angular e fortemente a Orientação a Objetos de JavaScript, construindo assim um framework capaz de suportar aplicações híbridas e multiplataforma.

Possuindo como vantagens: a baixa curva de aprendizado, tendo uma documentação simples e facilitada, com muitos componentes já pré-prontos, códigos podem ser feitos em várias plataformas, escolhendo qualquer forma entre Android, IOS a aplicação Web, por fim possuindo uma instalação leve e fácil de rodar nos navegadores, sem possuir a necessidade de instalar softwares externos.

Xamarin

Uma famosa plataforma para criação de aplicativos nativos nos grandes sistemas operacionais: IOS, Android e Windows, apenas utilizando uma linguagem e IDE sendo ela o C# como linguagem e Xamarin Studio como sua IDE. Possuindo uma vantagem em relações a outras ferramentas, sendo aplicativos serem nativos dos próprios Sistemas Operacionais, pelo fato das interfaces de usuário serem criadas para cada uma das plataformas possíveis e não se restringindo a uma UI padronizada. E como um diferencial ter a possibilidade de acessar recursos próprios dos dispositivos, utilizando C# com até certa facilidade.

Além do Xamarin Studio, presente no Mac, usuários de Windows podem optar por criar aplicações com Xamarin no Visual Studio, aproveitando que o Visual Studio é uma IDE consolidada, com diversas funções e recursos auxiliares para seu desenvolvimento do seu aplicativo.

Por fim, Xamarin oferece a capacidade de criar apps com apenas uma única base código, podendo ser aplicada em vários Sistemas Operacionais, permitindo até 90% do código compartilhado em código nativo C#, resultando em um aumento de performance e na rapidez do sistema. Possui uma biblioteca: Xamarin.Forms que facilita a criação de interfaces compartilhadas, destacando uma comunidade ativa e unida, declarando atualizações regularmente e poderosa.

Comparação das Plataformas

Comparando todas as plataformas populares de desenvolvimento mobile: React Native, Flutter, Ionic e Xamarin, deve-se destacar que todas possuem seus traços únicos, definindo em vantagens e desvantagens para possíveis mercados.

React Native, foi desenvolvida pelo Facebook, utiliza a junção de JavaScript com React, permitindo a criação de aplicativos com componentes nativos e de alta performance, gerados pela comunidade ativa e unida, oferecendo um suporte extensivo e familiar para pessoas que já usam ou conhecem JavaScript.

Flutter, criada pelo Google, utiliza a linguagem Dart, contribuindo uma excelente performance, comunicando diretamente ao código nativo do sistema, possui uma rica interface de usuário e tendo como ponto forte o Hot reload, sendo ideal para novos projetos que precisam de interfaces bonitas de alta performance.

Ionic, conhecida por utilizar: JavaScript, TypeScript, HTML, CSS, desenvolvida pela Drifty Co. e mantida pela Google. Permite aplicativos serem criados com uma aparência nativa, através da WebView. Idealizada para desenvolvedores web, devido a sua facilidade curva de aprendizado e elementos visuais estampados, entretanto, pode possuir problemas de performance se for trabalhado com trabalhos complexos.

Xamarin, criada pela Microsoft, utiliza a linguagem C#, oferecendo uma performance de alta desempenho, além de compilar códigos nativos. Destaca-se sua comunidade forte e unida da Microsoft, e que é recomendável para desenvolvedores familiarizados com sistemas .NET e outras tecnologias atuais.

Portanto, conclui-se que cada plataforma possui seus pontos fortes e fracos, ambientes diferentes, porém com a mesma área de atuação, sendo desenvolvimento mobile. A opção de escolha vai depender do que seu projeto precisar, levar sempre em conta as características de cada ferramenta: Performance, Agilidade, Empresa corporativa integrada, Comunidade ativa, baixa ou alta curva de

aprendizado, não existe resposta certa, e sim opções a serem escolhida.

2. Pesquisa das Linguagens de Programação Swift e Objective-C

Swift conhecida como uma linguagem de programação segura e moderna, em 2014 foi desenvolvida e introduzida pela Apple. Possui como características uma sintaxe simples e clara, facilitando a própria escrita do código e entendimento do projeto, tornando-se uma das linguagens mais acessíveis para todos os desenvolvedores. Tendo um diferencial no seu sistema de gerenciamento de memória automatizado, denominado “Automatic Reference Counting (ARC)”, utilizada para alta performance em suas diversas aplicações. Uma das grandes vantagens de Swift é a interoperabilidade própria, permitindo facilitar a migração gradual de projetos já existentes, refletindo que a Linguagem Swift receba atualizações regularmente pela Apple e tornando a Comunidade ativa feliz dos usuários e desenvolvedores.

Porém, Swift ainda enfrenta grandes desafios devido a sua relativa imaturidade, por ser considerada muito nova no mercado, causando um aumento de chance de erros acontecerem, sendo por exemplo de compatibilidade dos sistemas, resultando na necessidade de rescrever o mesmo código diversas vezes. Outro ponto que vale destacar, devido ao seu rápido desenvolvimento, a quantidade de recursos e bibliotecas disponíveis em Swift ainda não ultrapassam o mesmo nível de Objective-C.

Objective-C retratada como uma extensão da Linguagem C, introduzida na década de 1980, derivada do C, incorporando conceitos de orientação a objetos. Possui uma sintaxe complexa e detalhada, o que resulta numa curva de aprendizado demorada e íngreme para novos tipos de desenvolvedores. Todavia, Objective-C é considerada uma das linguagens mais estáveis e maduras, amplamente conhecida e utilizada no mercado atual. Permitindo manipulação direta entre ponteiros e outras funcionalidades de baixo nível, além de possuir uma base de código vasta, com muitas bibliotecas e recursos disponíveis, contribuindo para grande flexibilidade aos desenvolvedores finais.

Entretanto, apesar das vantagens registradas, Objective-C possui alguns pontos negativos. Sendo elas, a baixa segurança do sistema, e outro fato de possuir uma sintaxe complexa, demandando mais tempo que o necessário para se aprender uma ferramenta nova, deixando claro ser um desafio aos novos desenvolvedores. Com o passar do tempo vem recebendo menos atualizações e melhorias, mesmo sendo suportada pela Apple, deixa-se claro que a prioridade se passou a ser Swift.

Comparando as duas linguagens, percebe-se o destaque atual em cima de Swift, pela sua facilidade, acessibilidade, segurança e performance superior em várias operações, recebendo elogios pela sua sintaxe limpa e moderna, combinada com atualizações constantes, concluindo ser uma escolha ideal para novos desenvolvedores. Contudo, Objective-C oferece estabilidade e uma ampla base de recursos e bibliotecas no arsenal, sendo optada mais a projetos antigos ou já existentes que requerem uma manutenção, para no fim ter uma funcionalidade mais perto do nível baixo da máquina. Como escolha definitiva, deve-se levar em conta quais necessidades prioritárias os desenvolvedores desejam, pois com ambas as ferramentas é realizado o melhor dos dois mundos, apenas dependem do contexto embutido.

3. Pesquisa sobre a Linguagem de Programação Kotlin:

Kotlin desenvolvida pela JetBrains e suportada pelo Google e utilizada para desenvolvimento de aplicativos Android, lançada em meados de 2011, sendo uma linguagem de programação moderna e expressiva que se integra totalmente com JVM (Java Virtual Machine). Caracterizada por possuir uma sintaxe intuitiva e simples, tornando o aprendizado rápido e com baixos índices de erros. Baseada em ser interoperável com Java, permitindo que bibliotecas e frameworks em Java existentes sejam usados por novos desenvolvedores, facilitando a migração de projetos em Java para Kotlin, sem possuir a necessidade de rescrever o código novamente.

Kotlin possui como principais características, a ênfase na segurança e na simplicidade, introduzindo null safety, uma funcionalidade que ajuda a eliminar erros comuns relacionados a ponteiros nulos, demonstrando um dos problemas mais ocorridos em Java. Além disso, Kotlin suporta uma programação funcional, permitindo o uso de funções de ordem superior, a inclusão de extension functions, que permitem adicionar e alterar funcionalidades de classes, com intuito de melhorar a armazenagem de dados e coleções imutáveis, que proporciona o código ser mais legível e modular.

Exemplo de uma Aplicação:

Um exemplo notável de uma aplicação que utiliza Kotlin em suas tarefas é o Pinterest, uma plataforma de mídia social que permite usuários acessar, descobrir, salvar e compartilhar fotos, geralmente a respeito de ideias visuais, inspirações e designs e muito mais. Feita originalmente em Java, porém Pinterest começou a adotar Kotlin como uma forma de melhorar a eficiência, desenvolvimento de códigos e sua devida manutenção, melhorando a experiência do usuário sobre o aplicativo.

Definir Kotlin como uma escolha viável para o Pinterest, trouxe várias vantagens, sendo elas, o aumento da produtividade dos desenvolvedores, pela sintaxe clara e concisa do modelo Kotlin, reduzindo a quantidade de códigos complexos necessários antes em Java. Resultando numa facilidade de manutenção de códigos e promovendo uma melhoria no sistema de defesa mais robusto, através do null safety, eliminando possíveis ataques externos aos sistemas.

Em resumo, a adoção de Kotlin no Pinterest exemplifica como a linguagem pode beneficiar grandes aplicações móveis. Kotlin não apenas simplificou o processo de desenvolvimento, como também melhorou a performance e a confiabilidade do aplicativo, demonstrando porque muitas outras empresas estão interessadas em adotar Kotlin para seu desenvolvimento de aplicações Android.

Bibliografia

- 1 – React Native: (React Native o que é e tudo sobre o Framework) <https://www.alura.com.br/artigos/react-native>
- 2 – Flutter: (Flutter o que é e tudo sobre o Framework) <https://www.alura.com.br/artigos/flutter>
- 3 – Ionic: (Introdução ao Ionic)
<https://www.devmedia.com.br/guia/ionic/38372>
- 4 – Xamarim: (Desenvolvimento Multiplataforma com Xamarin)
<https://www.devmedia.com.br/desenvolvimento-multiplataforma-com-xamarin/33467>
- 5 – Swift: (Desenvolvimento IOS, conheça Swift)
<https://www.devmedia.com.br/desenvolvimento-ios-conheca-a-linguagem-swift/31860>
- 6 – Objective-C: (Introdução Objective-C)
<https://www.devmedia.com.br/objective-c-tutorial/19219>
- 7 – Kotlin: (Principais Startups usando Kotlin)
<https://blog.back4app.com/pt/startups-usando-kotlin/>