

2. INTRODUÇÃO A DISPOSITIVOS MÓVEIS (2h)

1. Dispositivos Móveis

1.1 Definição e Tipos de Dispositivos

Dispositivos móveis são equipamentos eletrônicos portáteis que permitem a realização de diversas tarefas, como comunicação, acesso à internet, execução de aplicativos e consumo de mídia digital. Esses dispositivos contam com recursos de hardware e software que possibilitam sua versatilidade e mobilidade.

Tipos de Dispositivos:

- **Smartphones:** Dispositivos multifuncionais com capacidade de realizar chamadas, enviar mensagens, acessar a internet e executar aplicativos diversos.
- **Tablets:** Equipamentos com telas maiores, ideais para leitura, consumo de mídia e produtividade.
- **Wearables:** Dispositivos vestíveis, como smartwatches e pulseiras inteligentes, que monitoram atividades físicas, notificações e saúde.

Diferenças entre Android e iOS:

- **Android:** Sistema operacional de código aberto, com maior variedade de dispositivos e personalização.
- **iOS:** Sistema proprietário da Apple, com integração otimizada entre hardware e software, oferecendo maior controle de segurança e desempenho.

1.2 Histórico e Evolução

A evolução dos dispositivos móveis acompanha o avanço da tecnologia e das necessidades da sociedade:

- **Primeira Geração:** Telefones celulares básicos, limitados a chamadas de voz.
- **Segunda Geração (2G):** Introdução de mensagens de texto (SMS) e conexões digitais.
- **Terceira Geração (3G):** Acesso à internet móvel, videoconferências e maior velocidade de dados.
- **Quarta Geração (4G):** Streaming de vídeo em alta qualidade, aplicativos avançados e redes sociais.
- **Quinta Geração (5G):** Conexões ultra rápidas, suporte para IoT (Internet das Coisas) e realidade aumentada.

Impacto da Mobilidade:

- Transformação da comunicação instantânea.
- Facilidade de acesso à informação.
- Revolução em áreas como saúde, educação, transporte e entretenimento.

1.3 Características e Arquitetura

Os dispositivos móveis possuem uma arquitetura específica para otimizar o desempenho e a eficiência energética:

- **Processadores ARM:** Projetados para alta eficiência energética e desempenho em dispositivos móveis.
- **Memória:** Inclui RAM para execução de aplicativos e armazenamento interno para dados e arquivos.
- **Sensores:** Acelerômetro, giroscópio, GPS, sensores de proximidade e biometria, permitindo a interação com o ambiente.

Sistemas Operacionais Móveis:

- **Android:** Baseado em Linux, com suporte a uma ampla variedade de dispositivos.
- **iOS:** Sistema da Apple, otimizado para dispositivos como iPhone e iPad, com ênfase em segurança e desempenho.

Exemplo de Aplicação:

Atividade em Grupo:

- **Tema:** "Como os Aplicativos Móveis Transformaram o Dia a Dia"
- **Descrição:** Os alunos formarão grupos para discutir e apresentar casos reais de como os aplicativos móveis impactaram diferentes aspectos da sociedade, como saúde, transporte, educação e comunicação.
- **Objetivo:** Estimular o pensamento crítico sobre o papel da tecnologia móvel na vida cotidiana e sua evolução.

Anexo 1 – Desenvolvimento Mobile Multiplataforma X Nativo: Escolha a Melhor Abordagem

A Área de Desenvolvimento Mobile tem crescido cada dia mais e se expandido devido a uma série de fatores, mas em especial pelo aumento no uso de Dispositivos Móveis. Diversos estudos e indicadores apontam um aumento significativo na utilização em especial de celulares para diversas atividades diárias. Por esse motivo o trabalho de Desenvolvimento de Ferramentas, Softwares e Aplicativos Mobile vive um momento de aquecimento. E para ajudar você que possa estar em dúvidas sobre o Desenvolvimento Mobile Multiplataforma ou Nativo,

nós preparamos esse material para tirar todas as dúvidas e te ajudar na escolha da melhor abordagem.

Panorama do Mercado

Sabe-se que a utilização de Dispositivos móveis são extremamente populares. Só no Brasil, segundo Anatel de 2023, há aproximadamente 250 milhões de celulares em uso.

<https://urmobo.com.br/blog/brasil-possui-2516-milhoes-de-celulares-ativos/>

O desenvolvimento de aplicações e soluções com foco nesses dispositivos costuma ser abordado de duas maneiras hoje em dia: com desenvolvimento nativo ou híbrido.

Desenvolvimento Nativo

Desenvolvimento nativo é quando a aplicação mobile que desenvolvemos é específica de uma única plataforma ou sistema, e é feita com as linguagens e ferramentas específicas daquela plataforma.

As duas grandes plataformas mobile atualmente são o Android e o iOS. Para desenvolvimento nativo no Android, fazemos uso de uma SDK (Android SDK) e uma IDE (Android Studio).

A Google oferece ambas gratuitamente e um bundle com SDK + IDE que pode ser baixado

para Windows, Linux ou Mac. As linguagens para o desenvolvimento nativo são o Kotlin e o Java, mas o Kotlin é a linguagem preferível para o desenvolvimento, segundo a Google (não sabem apreciar a beleza e clareza da mais linda linguagem de todos os tempos).

Importante frisar que se seu computador possuir menos que 16Gb de RAM, usar emulador vai ser mais difícil que rodar GTA sem placa de vídeo. Temos algo muito parecido no desenvolvimento nativo no iOS, com uma IDE (XCode) e uma SDK (Cocoa Touch) disponíveis. A diferença, porém, é que você precisa ter um computador da Apple () para desenvolver nessa plataforma. Esses computadores podem valer o preço de um carro, mas são necessários para desenvolver e testar nessa plataforma.

As vantagens do desenvolvimento nativo são:

- Alta performance : Aplicativos nativos, na grande maioria dos casos, tem uma performance superior às alternativas híbridas. Se seu app faz uso intensivo de CPU ou GPU, ou necessita de um controle rígido de hardware por exemplo, considere o desenvolvimento nativo.
- Experiência de usuário: Há unificação da experiência de usuário, visto que usamos os recursos nativos da plataforma. Componentes e recursos, portanto, atendem melhor o padrão de experiência oferecido pela plataforma do que nas alternativas híbridas.
- Poder: Com o desenvolvimento nativo temos mais controle sobre o que acontece por “trás dos panos”, além de total compatibilidade com recursos do sistema.

...Já as desvantagens do desenvolvimento nativo

- Múltiplos aplicativos, mesma funcionalidade : Para atender o mercado de mobile, seu aplicativo possivelmente precisará ser disponibilizado no Android e no iOS. E talvez haja também o desejo de uma versão Web. Bom, para isso você vai precisar de desenvolvedores para o iOS, com conhecimento em Swift. Também precisará de desenvolvedores com conhecimento em Kotlin para o Android. E por fim os desenvolvedores de Web, o povo do Vue, React, Angular, etc... Será necessário que todos esses sistemas atendam as especificações visuais e comportamentais de cada plataforma e ao mesmo tempo possuam uma identidade compartilhada do negócio.

Resumindo: precisa ser o ‘mesmo app’ tanto no Android quanto no iOS e na Web.

Considere também que o tempo de desenvolvimento costuma ser maior e o planejamento de melhorias, mais delicado e trabalhoso, pois precisam ser aplicadas em 3 apps diferentes.

- Custo: Visto que, no desenvolvimento nativo, teremos múltiplos aplicativos, um para cada plataforma, o custo de desenvolvimento costuma ser muito maior. Haverá times de desenvolvimento para cada plataforma que se quer distribuir o app.
- Tempo de desenvolvimento: O que o time de Android desenvolver não poderá ser duplicado pelo time do iOS e vice-versa. As soluções precisam ser desenvolvidas para cada plataforma, demandando mais tempo.

Desenvolvimento Híbrido

Desenvolvimento híbrido é a utilização de algum framework ou ferramenta que permite uma mesma base de código. Ou seja, um único desenvolvimento sendo distribuído em múltiplas plataformas. No desenvolvimento híbrido, uma mesma equipe (possivelmente) vai desenvolver, numa única linguagem, e distribuir para múltiplas plataformas como Android, iOS, Desktop e Web. Existe, no entanto, o risco de se gastar muito tempo aplicando remendos no framework híbrido para atender algum recurso ou funcionalidade que teria um tratamento mais simples no caso do desenvolvimento nativo.

Dentre os maiores e mais populares frameworks de desenvolvimento híbrido temos:

React Native

Framework open source desenvolvido pelo Facebook onde os Desenvolvedores usam a linguagem Javascript para criação de seus apps. A distribuição de apps em React Native é possível em iOS, Android e Windows. Enquanto vantagens do React Native temos sua popularidade e também a base sólida de usuários, exemplos e código disponível. Já a desvantagem é a performance para operações de uso intensivo de processamento, por exemplo. A performance costuma ser pior em dispositivos Android do que em dispositivos iOS.

Flutter

Não tão popular quanto o React Native o Flutter é um framework desenvolvido pela Google. Ele utiliza a obscura linguagem Dart e é capaz de distribuir a mesma base de desenvolvimento para as plataformas iOS, Android, Desktop, Web e até para dispositivos embedded, como geladeiras, torradeiras, massageadores, etc. Costuma ter um desempenho melhor que o React Native e Ionic, sendo essa sua principal vantagem. Apesar de ser jovem, Flutter está tornando-se popular rapidamente. A linguagem Dart, segundo pesquisa do stackoverflow de 2020, é mais querida entre desenvolvedores do que C#, Swift ou JavaScript (inclusive, dessas linguagens aqui citadas, JavaScript está em último lugar. Pobrezinha). Como desvantagem há o atraso em algumas adaptações realizadas pelos sistemas operacionais – é necessário um tempo para que essas adaptações sejam implementadas e disponibilizadas no framework.

Ionic

O mais velho dos três, o Ionic é um SDK open source que permite uso de frameworks famosos no desenvolvimento web para a criação de apps mobile: React, Angular e Vue. Ele faz uso do runtime Capacitor (a recomendação moderna) para disponibilizar recursos de hardware e software do sistema. Sua principal vantagem é permitir diversos frameworks modernos e populares no desenvolvimento.

A desvantagem é o baixo desempenho em operações com alto processamento e a necessidade de múltiplos plugins e pacotes de terceiros.

Qual Ambiente de Desenvolvimento Mobile utilizar, então?

Vai depender do que é o seu projeto, do conhecimento que sua equipe tem e do tempo disponível para atender sua entrega. Não existe uma solução matadora para todo caso, ou uma receita pronta do sucesso e melhor resultado.

Porque o Flutter

O Flutter vem ganhando um status considerável como Framework de desenvolvimento mobile e com ele muitas pessoas estão buscando conhecer melhor como é a criação de aplicativos utilizando essa ferramenta.

Como se trata de uma tecnologia Google, não é de se estranhar que os Dev's que estão vindo do desenvolvimento nativo com o Kotlin/Java e utilizavam o Android Studio, iniciem seu contato com o Flutter aproveitando o conhecimento na mesma IDE já que o suporte é excelente também.

A todo momento surgem soluções do tipo “um código, dois aplicativos” entre as comunidades de desenvolvimento mobile. Isso é natural porque a ideia de escrever um único código que pode ser reutilizado entre aplicações e em diversas plataformas é tentadora tanto

para quem financia o projeto, pela possibilidade de manter equipes menores, quanto para o desenvolvedor, que muitas vezes pode utilizar suas habilidades como ponto de partida no estudo de uma tecnologia, reduzindo assim a sua curva de aprendizado. Entretanto, muitas dessas soluções, citadas no fim, tendem a sacrificar a performance para alcançar certa onipresença.

Por exemplo, o Ionic, por executar em uma Webview, pode entregar um aplicativo multiplataforma, mas a um custo de desempenho altíssimo. O React Native, embora utilize código nativo de cada plataforma, ainda precisa de “pontes” que façam a ligação entre as mesmas e a lógica JavaScript da sua aplicação, o que também podem ser um causador de gargalos.

E se pudéssemos gerar código nativo de alto desempenho tanto para Android quanto para iOS, utilizando uma linguagem de alto nível? E se existisse uma biblioteca que nos desse todos os elementos visuais necessários para uma aplicação, seja seguindo o Material Design do Android ou o padrão de design do iOS, que permitisse a criação de aplicativos extremamente elegantes, com animações fluídas? Esse mundo existe e se chama Flutter.

O Flutter foi apresentado pela primeira vez em meados de 2017 pelo Google, mas foi em Dezembro de 2018, que a primeira versão estável foi liberada para o público. Com o Flutter podemos construir aplicativos nativos e de alta qualidade tanto para Android quanto para iOS, respeitando os padrões de cada uma dessas plataformas, a partir de um mesmo código escrito com a linguagem Dart.

<https://medium.com/@leonardopaim/utilizando-o-vs-code-para-criar-aplicativos-flutter-windows-5074962ecfce>

<https://www.devmedia.com.br/hello-world-com-flutter/40321>

<https://blog.accurate.com.br/desenvolvimento-mobile/>