

## SUMÁRIO

<b>1. DESENVOLVIMENTO MOBILE CROSS-PLATAFORM.....</b>	<b>1</b>
1.1. DESENVOLVIMENTO NATIVO .....	1
1.2. EM DETALHES: APLICATIVO NATIVO, HÍBRIDO E CROSS PLATAFORM.....	1
1.2.1. SOBRE CAMADAS DE SOFTWARE – BAIXO NÍVEL E ALTO NÍVEL .....	1
1.2.2. APLICATIVOS NATIVOS: AO LADO DO SISTEMA OPERACIONAL .....	1
1.2.3. SISTEMAS HÍBRIDOS: SITES QUE FINGEM SER APLICATIVOS NATIVOS .....	2
1.2.4. APLICATIVOS CROSS-PLATAFORM.....	2
<b>2. LINGUAGEM DE PROGRAMAÇÃO DART .....</b>	<b>3</b>
2.1. DART NATIVE .....	3
2.2. DART WEB .....	3
<b>3. FRAMEWORK FLUTTER .....</b>	<b>4</b>
<b>4. POSSIBILIDADES COMBINAÇÃO DART-FLUTTER.....</b>	<b>4</b>
4.1. GERENCIADOR DE PACOTES PUB .....	4
4.2. APLICAÇÃO WEB .....	4
4.3. APLICATIVO ANDROID/IOS OFFLINE .....	4
4.4. APLICATIVO DUAL – ONLINE E OFFLINE.....	4
<b>5. FONTES.....</b>	<b>5</b>

## 1. DESENVOLVIMENTO MOBILE CROSS-PLATAFORM

Quando falamos em desenvolvimento multiplataforma ou cross-plataform, nos referimos a um único código-fonte compilado em código nativo de sistemas operacionais diferentes. Assim, quando adicionamos o termo “mobile”, antes de “cross-plataform”, queremos nos referir aos diferentes sistemas operacionais para dispositivos mobile.

O desenvolvimento mobile cross-plataform é, portanto, aquele cuja tecnologia utilizada entrega um produto que funciona nos diferentes sistemas operacionais dos dispositivos mobile - que seriam o Android e o IOS.

### 1.1. DESENVOLVIMENTO NATIVO

Se o cross-plataform é orientado a um aplicativo unitário multiplataforma, em contraste, o nativo é orientado ao desenvolvimento de uma solução para cada plataforma. Em outras palavras, a equipe de desenvolvimento cria um aplicativo para cada sistema operacional no qual o app será lançado.

### 1.2. EM DETALHES: APLICATIVO NATIVO, HÍBRIDO E CROSS PLATAFORM

Para complementar o que é dito nos tópicos a seguir, recomendo a leitura [desse](#) artigo, e também porque comenta sobre a tecnologia que será abordada posteriormente - o Flutter. Também [esse vídeo](#) irá auxiliar no entendimento dos tópicos, e desse [outro artigo](#).

#### 1.2.1. SOBRE CAMADAS DE SOFTWARE – BAIXO NÍVEL E ALTO NÍVEL

Um dispositivo computacional que possui linguagem binária, um kernel, um sistema operacional, enfim, é um que opera em **camadas de software**. Quanto mais próximo do hardware, dizemos que menor é o “**nível**” da camada, e quanto mais acima estiver, maior o “nível” dela. Como “nível”, pode-se entender o grau de distanciamento e abstração em relação ao funcionamento físico da máquina.

#### 1.2.2. APLICATIVOS NATIVOS: AO LADO DO SISTEMA OPERACIONAL

Para esta seção, **vamos considerar o sistema operacional como o nível mais baixo**, ou a base do fluxo das camadas. Pois bem: **diz-se que uma aplicação é nativa quando** o seu código é o preliminarmente usado para o dispositivo e seu sistema operacional. Ou seja, a aplicação nativa é a que conversa mais facilmente com o sistema operacional, e por isso, compreensivelmente, é mais rápida e performática.

Além disso, aplicativos móveis nativos com capacidade **armazenam o software do aplicativo móvel e seus dados localmente** no dispositivo móvel, o que **dispensa a conexão com a Internet** para o seu uso.

**A comparação de desempenho entre o IOS e o Android dá uma luz adicional sobre o assunto.** Resumidamente, o Android é uma solução que visa a maior adaptação com a diversidade de hardwares, enquanto a Apple possui total controle sobre o ecossistema do IOS.

Ou seja: a solução Android necessita de uma camada adicional para sua adaptabilidade, e por isso, mais memória RAM. Clique [aqui](#) para ler mais.

### 1.2.3. SISTEMAS HÍBRIDOS: SITES QUE FINGEM SER APLICATIVOS NATIVOS

Os sistemas híbridos são uma combinação de aplicativos nativos e da web, um dos resultados da busca do mercado por interoperabilidade. O funcionamento interno de um aplicativo híbrido é semelhante ao de um aplicativo da web, mas é instalado como um aplicativo nativo. Os aplicativos híbridos têm acesso às APIs internas do dispositivo, o que significa que podem usar recursos como câmera, armazenamento, GPS, etc.

Os aplicativos híbridos são desenvolvidos com HTML e CSS, e normalmente executam um aplicativo da web por meio de um contêiner ou [WebView](#) , um navegador que pode estar contido em um aplicativo móvel. Por conseguinte, **essa camada da WebView será mais uma no fluxo do sistema**, e o efeito colateral será um adicional **custo de processamento**.

Os aplicativos híbridos são baseados em aplicativos da web e contêm os mesmos elementos de navegação que os aplicativos da web, como a **dependência de conexão com a Internet**. Exemplos de aplicativos híbridos são o Facebook e o Instagram.

**Ou seja**, e já evitando confusões com o cross-plataform, um aplicativo híbrido é “híbrido” não no sentido de que opera em diversos sistemas operacionais, mas tanto na web quanto como na forma de um aplicativo nativo. É um site fantasiado de aplicativo.

### 1.2.4. APLICATIVOS CROSS-PLATAFORM

**Muitas pessoas erram ao dizer que híbrido e plataforma cruzada são dois nomes para a mesma coisa.** Mas a única semelhança entre os dois tipos de aplicativo é a capacidade de utilização do código em diferentes plataformas.

“Ah, mas um híbrido é multiplataforma”. Sim, é verdade. Mas o ponto crucial é que não serve para os mesmos fins de um aplicativo classificado dessa forma. Um híbrido é um site que se passa por aplicativo nativo, e um cross-plataform está mais para um aplicativo que pode ser

instalado no dispositivo, pode funcionar offline, e ser utilizado em mais de um sistema operacional. Mais sobre isso pode ser encontrado [aqui](#).

O desenvolvimento de aplicativos cross-plataform é tida como a melhor abordagem para aplicativos personalizados de baixo custo com recursos seguros, estáveis, fáceis de manter e iterar, além de apresentarem uma performance melhor do que um sistema híbrido, ou até mesmo, dependendo da tecnologia, como o Flutter, comparável com a de um aplicativo nativo.

## 2. LINGUAGEM DE PROGRAMAÇÃO DART

Foi visando, no início, a substituição no frontend, do Javascript e do React Native para o desenvolvimento de aplicativos móveis de plataforma cruzada, que o Google criou a estrutura **Flutter**, que é um **kit de ferramentas UI**, e que **utiliza a linguagem de programação Dart**. E vale enfatizar que foi no início, porque hoje essa ideia foi abandonada, e o Google criou diversos mecanismos para que as tecnologias possam atuar juntas.

Pois bem: o Dart é **uma linguagem orientada a objetos**, e que pode ser utilizada em aplicações **desktop, mobile, back-end (algo recente) e para scripts**. Além disso, é bastante semelhante a outras linguagens de programação, de forma que o aprendizado dela, se posterior a essas outras, como Javascript, será facilitado.

Para atender a todas essas soluções, o **Dart trabalha com duas soluções diferentes**. Cada uma oferece os recursos especificamente necessários para o contexto. São elas: **Dart Native e o Dart Web**.

### 2.1. DART NATIVE

É utilizado para o desenvolvimento mobile, desktop e de servidor, entre outros. Para realizar a produção de código, ele inclui a máquina virtual do Dart com compilação JIT (just-in-time) e compilador AOT (ahead-of-time). Ele reúne o código para Dart de forma nativa, seja em ARM ou X64.

### 2.2. DART WEB

Tem como foco o desenvolvimento web e traz o [dartdevc](#), que compila o código para o desenvolvimento, e o [dart2js](#), que é indicado para fazer o build para produção. Neste caso, o resultado final é um Javascript, que será facilmente interpretado por qualquer navegador – visto que essa linguagem é universal.

### 3. FRAMEWORK FLUTTER

Como dito inicialmente, [Flutter](#) é um framework que foi desenvolvido pelo Google, e utiliza a linguagem Dart. Ele é utilizado para desenvolvimento frontend de aplicativos em Android, IOS e aplicações web. Ele tem como bases fundamentais o desenvolvimento com mais agilidade, a produção de interfaces mais bonitas e a performance nativa.

É muito importante ressaltar que o suporte Flutter para web oferece as mesmas experiências para plataforma web e também no celular. Trata-se da possibilidade de ser uma mesma estrutura do Flutter em múltiplos ambientes.

### 4. POSSIBILIDADES COMBINAÇÃO DART-FLUTTER

#### 4.1. GERENCIADOR DE PACOTES PUB

Primeiramente, ao contrário de outras linguagens de programação, o Dart tem seu próprio gerenciador de pacotes chamado Pub. Clique [aqui](#) para acessar o site oficial da tecnologia.

#### 4.2. APLICAÇÃO WEB

Com Flutter é possível criar o frontend de uma aplicação web com o mesmo código que poderia ser utilizado para aplicativos mobile. Além disso, recentemente o Dart passou a ter [suporte para o back-end](#), sendo possível, então, criar um sistema completo com a combinação Flutter-Dart.

Também é possível combinar o desenvolvimento frontend do Flutter, com outra tecnologia backend, como PHP/Laravel.

#### 4.3. APLICATIVO ANDROID/IOS OFFLINE

Com Flutter é possível criar um aplicativo que funcione offline, utilizando o armazenamento interno/local do dispositivo para persistir os dados.

#### 4.4. APLICATIVO DUAL – ONLINE E OFFLINE

Com Flutter é possível criar um aplicativo DUAL em termos de conectividade, isto é, que funcione offline, persistindo os dados no armazenamento interno do dispositivo, que podem ser gerados durante a execução do programa, ou até mesmo lidos a partir de um dispositivo externo, como um [USB](#), e também online, por meio da [detecção](#) da existência de um ponto de acesso à internet.

## 5. FONTES

<https://www.opensoft.pt/diferencas-aplicacoes-nativas-web-hibridas/>

<https://bloomidea.com/blog/aplicacoes-nativas-vs-hibridas-qual-escolher-para-o-seu-projeto>

<https://www.leonardopalis.com.br/post/entendendo-de-vez-o-que-s%C3%A3o-aplica%C3%A7%C3%B5es-h%C3%ADbridas-e-cross-platform#:~:text=Sim%2C%20existe!,um%20app%20nativo%2C%20%C3%A9%20claro.>

<https://www.hostgator.com.br/blog/o-que-e-dart-na-programacao/>

<https://blog.cubos.io/por-que-desenvolver-sua-solucao-web-em-flutter/#:~:text=Como%20dito%20inicialmente%2C%20Flutter%20%C3%A9,web%2C%20foco%20maior%20deste%20artigo.>

<https://blog.codemagic.io/what-is-flutter-benefits-and-limitations/>

<https://www.iugu.com/blog/o-que-e-flutter#:~:text=Tendo%20como%20maior%20concorrente%20e,melhorando%20a%20experi%C3%Aancia%20do%20usu%C3%A1rio.>