

DIFERENTES ESTRATÉGIAS DE DESENVOLVIMENTO MOBILE

Ao desenvolver aplicações móveis (mobile) compete tanto ao programador quanto à equipe de programadores e gestores optar por uma estratégia de desenvolvimento que melhor se adeque às características do projeto em questão. As principais estratégias mercadológicas adotadas atualmente são: **I - Desenvolvimento Nativo, II – Desenvolvimento Híbrido e III – Desenvolvimento Cross-Platform Nativo** e são divididas de acordo com a linguagem e ferramentas utilizadas.

A escolha dessas estratégias pode ser baseada em diferentes características/necessidade como: **desempenho, custo de produção, experiência** do usuário, **manutenção** entre outras.

ESTRATÉGIAS

1. DESENVOLVIMENTO NATIVO

O desenvolvimento nativo é realizado com as linguagens **oficiais** de cada plataforma, por exemplo: **Swift** ou **Objective-C** para aplicações **iOS** (Apple). Linguagens como **Java** e **Kotlin** para **Android** (Google). Essa abordagem de desenvolvimento oferece **máximo desempenho** e **integração** pois são destinadas ao próprio sistema operacional. No entanto, por serem “**singulares**”, exigem o desenvolvimento de códigos distintos (um código para cada sistema) o que resulta em **maior** investimento de **tempo** e **custo** para criação e futuras manutenções.

2. DESENVOLVIMENTO HÍBRIDO

O desenvolvimento Híbrido utiliza tecnologias web (**HTML, CSS e JavaScript**) para criar aplicações que são executadas dentro de uma **WebView**. Para desenvolver esses aplicativos são usados Frameworks como **Ionic** e **Apache Cordova** que empacotam (reúnem) ferramentas para rodarem em dispositivos móveis. A grande vantagem dessa abordagem está na **velocidade** e **redução** de **custos** na hora de criar/reutilizar o código, porém o **desempenho** costuma ser **inferior** ao nativo e a **experiência** do usuário parecer menos integrada à plataforma, isso por que pode-se passar uma ideia de “**adaptado**”, ou até mesmo criar a seguinte ideia: “não foi feito para isso, mas pode resolver isso”.

3. DESENVOLVIMENTO CROSS-PLATFORM NATIVO

Essa abordagem tenta mesclar o **desempenho** da estratégia nativa com a **produtividade** e **flow** da estratégia híbrida permitindo criar um único código que gera aplicativos iOS e Android. Nela são utilizados Frameworks famosos como **React Native**, **Flutter** e **.NET MAUI** (evolução do Xamarin e foco desta disciplina). Essa abordagem objetiva simplificar o ciclo de desenvolvimento e oferecer uma base **única** de código, reduzindo o **tempo** e **custo** de manutenção e mantendo performance honesta e escalabilidade.

CONCLUSÃO

Cada tipo de desenvolvimento apresenta **vantagens** e **desvantagens** conforme às necessidades do projeto.

O **Desenvolvimento Nativo** oferece o melhor **desempenho** e **experiência do usuário**, mas também apresenta maior custo e manutenção duplicada, além de mão de obra especializada.

O **Desenvolvimento Híbrido** reduz o **custo** e o **tempo** de criação, porém sacrifica desempenho e responsividade (adequação ao mobile).

O **Desenvolvimento Cross-platform nativo**, especialmente com ferramentas modernas e versáteis como **.NET MAUI**, **Flutter** e **React Native**,

proporciona um bom equilíbrio: mantém **desempenho satisfatório**, **manutenção simplificada** e **baixo custo** em comparação às abordagens tradicionais sendo uma boa opção quando exposta ao ritmo selvagem de produção atual.

Segundo pesquisas de mercado (Stack Overflow e Statista), frameworks como **Flutter**, **React Native**, e linguagens como **Swift** e **Kotlin** são os mais utilizados atualmente, demonstrando a tendência de unificação de código e eficiência no ciclo de desenvolvimento.

Essas escolhas tecnológicas impactam diretamente na produtividade das equipes, a escalabilidade das soluções e a qualidade final da experiência entregue ao usuário.