**Android Studio**

O Android Studio provê um ambiente de desenvolvimento, debug, testes e profile multiplataforma para Android.

Vantagem:

-A interface do Android Studio é bem atraente, tem um leque de *schema*considerável, podendo utilizar vários “*Look and Feel”*diferentes. Eu optei pelo tema *Darcula*, o mesmo usado na apresentação do Google I/O. Também é possível customizar os atalhos de teclado, de forma que sejam iguais a outras IDEs, como o Eclipse. Assim, o impacto ao usar a ferramenta tende a ser menor.

Desvantagem:

-Não é possível abrir mais de um projeto na mesma janela. Ao tentar abrir outro projeto, o Android Studio permite abri-lo em uma outra janela ou substituir a atual. Isso poderá incomodar alguns desenvolvedores que costumavam trabalhar com vários projetos e/ou workspaces no Eclipse.

**React Native**

React Native é uma das estruturas de JavaScript mais populares do mundo hoje. É famoso por sua utilidade para a criação de aplicativos nativos nas plataformas Android e iOS. O React Native encontra suas raízes no React; é uma biblioteca JavaScript que o Facebook criou para projetar a interface de aplicativos móveis. React Native pode ser combinado com JavaScript para construir aplicativos com recursos semelhantes aos nativos. Como os códigos React Native são compartilháveis, é uma excelente plataforma para o desenvolvimento de aplicativos de plataforma cruzada.

Vantagem:

-De acordo com o site Statista, JavaScript é a linguagem de programação mais usada no mundo. De acordo com um estudo da Statista, 68% dos desenvolvedores escrevem JavaScript. Isso torna mais fácil para os desenvolvedores JavaScript usarem o React Native, já que a estrutura é escrita em JavaScript.

Desvantagem:

-Apesar de todos os excelentes recursos, React Native ainda está na versão Beta! É por isso que ele ainda apresenta alguns problemas aparentes, como a complexidade de depurar aplicativos, entre outras limitações, incluindo problemas de compatibilidade. Lembre-se de que trabalhar com o React nativo pode ser complicado quando surge a necessidade de depuração.

-React Native é ótimo para criar aplicativos simples com visuais atraentes. No entanto, torna-se um desafio obter um desempenho excelente quando você precisa adicionar recursos complexos. Então, você pode ter que se contentar com um dos dois.

**Flutter**

Flutter é um SDK para desenvolvimento móvel multiplataforma (iOS e Android) lançado pela Google em 2017. Apesar de ter sido lançado recentemente, Flutter possui frameworks, ferramentas e widgets próprios que facilitam a criação de interfaces para ambas plataformas.

Flutter é baseado em Dart, uma linguagem orientada a objeto também desenvolvida pela Google, anunciada em 2011. Apesar de ser uma linguagem popularizada recentemente, é uma linguagem prática e fácil de aprender, principalmente porque possui uma sintaxe similar a Java e C#.

Vantagem:

O desenvolvimento é feito com um único código (no caso do Flutter, na linguagem Dart), poupando tempo e trabalho para os desenvolvedores. Esse, porém, é um benefício de todos os frameworks multiplataforma e um dos maiores pesos para desenvolvedores em grupos pequenos ou independentes, e para projetos pequenos.

Desvantagem:

Enquanto seu principal concorrente, o React Native, também é utilizado para desenvolvimento web, Flutter só é compatível com desenvolvimento mobile, então se for necessário o mínimo de esforço possível para desenvolver aplicações para várias plataformas (além de mobile), Flutter pode não ser a melhor solução. Entretanto, Flutter será utilizado para desenvolver aplicações na plataforma Fuchsia, um novo sistema operacional que está sendo desenvolvido pela Google desde 2016.

Xamarin

O Xamarin é uma ferramenta usada para o desenvolvimento de aplicativos móveis multiplataforma que permite aos engenheiros compartilhar cerca de 90% do código nas principais plataformas. Sendo uma ferramenta relativamente nova, ela é baseada na pilha de tecnologia da Microsoft e já possui uma comunidade de mais de 1,4 milhão de desenvolvedores. A plataforma foi construída pelos desenvolvedores por trás do Mono, uma plataforma de desenvolvimento de código aberto baseada no .NET Framework, liderada por Miguel de Icaza e introduzida pela primeira vez em 2001.

Vantagem:

-Ao contrário das soluções híbridas tradicionais, baseadas nas tecnologias da Web, um aplicativo de plataforma cruzada construído com o Xamarin ainda pode ser classificado como nativo. As métricas de desempenho são comparáveis ​​às do Java para Android (conforme explicado aqui) e do Objective-C ou Swift para o desenvolvimento de aplicativos iOS nativos. Além disso, o desempenho do Xamarin está sendo constantemente aprimorado para corresponder totalmente aos padrões de desenvolvimento nativo. O Visual Studio oferece uma solução completa para criar, testar e acompanhar o desempenho do aplicativo: o Visual Studio App Center permite que você execute testes de interface do usuário automatizados e identifique problemas de desempenho antes do lançamento. No entanto, este serviço é fornecido por uma taxa adicional.

Desvantagem:

-Pode haver alguns problemas ao integrar recursos de terceiros ao aplicativo Xamarin. Embora a maioria das ferramentas e bibliotecas ofereçam suporte completo para tecnologias nativas, a conformidade com Xamarin pode não ser fornecida pelo vedor. Embora o Xamarin tenha sua própria loja de componentes, sempre há uma chance de que você precise de um recurso específico ou integração no seu aplicativo que não seja fornecido pela plataforma.