

Xamarin

Cross Platform Development Technologies

Gabriela Moreira Prata^[PG50390] and Marco António Rodrigues
Sampaio^[PG47447]

¹ Universidade do Minho

² UC: Tópicos de Desenvolvimento de Software

1 Introdução

A tecnologia móvel tornou-se uma parte crucial do nosso quotidiano, e o mercado de aplicações móveis cresce constantemente. Com isso, existe a necessidade de desenvolver aplicações para diferentes plataformas móveis, como iOS, Android e Windows.

No entanto, a criação de aplicações nativas para cada uma dessas plataformas pode ser um processo demorado e caro, pois envolve o uso de diferentes ferramentas e linguagens de programação. É aí que entra o Xamarin, uma plataforma de desenvolvimento de aplicações móveis que permite criar aplicações nativas para múltiplas plataformas usando uma única base de código em C#.

Neste projeto de investigação, vamos abordar os principais tópicos relacionados ao Xamarin, incluindo os conhecimentos necessários para desenvolver nessa tecnologia, a arquitetura e o fluxo de trabalho da plataforma, as suas vantagens e limitações, casos de uso e exemplos de sistemas conhecidos construídos com essa tecnologia. O objetivo deste relatório é fornecer uma compreensão básica de Xamarin e entender melhor todo o conceito destas plataformas de desenvolvimento de aplicações móveis.

2 Xamarin

Xamarin é uma tecnologia que permite criar aplicações para dispositivos móveis, como telefones e tablets, usando a linguagem de programação C#. Com o Xamarin, podemos criar aplicações para iOS, Android e Windows usando uma única base de código.

Uma das grandes vantagens do Xamarin é que quando usamos essa tecnologia para desenvolver aplicações, este tem um desempenho semelhante às aplicações nativas, ou seja, desenvolvidas especificamente para uma plataforma específica.

Há duas formas de realizar o desenvolvimento de aplicações móveis Xamarin: Xamarin.Native (Xamarin.iOS e Xamarin.Android) ou Xamarin.Forms. Normalmente, a principal diferença entre as duas plataformas é que o Xamarin.Forms permite reutilizar o mesmo código de interface do utilizador em várias plataformas, enquanto o Xamarin.Native é adaptado a APIs específicas da plataforma, como o Windows, iOS e Android.

O Xamarin consiste em várias partes, incluindo bibliotecas de classes para iOS e Android, compiladores para cada plataforma, um ambiente de desenvolvimento chamado Xamarin Studio e plug-ins para o Visual Studio.



3 Conhecimento adquirido anteriormente para desenvolvimento da tecnologia

Para desenvolver aplicações na plataforma Xamarin, é importante ter conhecimentos relevantes adquiridos anteriormente. Esses conhecimentos incluem uma boa compreensão da linguagem de programação C#, que é a linguagem principal usada no desenvolvimento de aplicações na plataforma. Além disso, é necessário ter um bom entendimento do framework .NET e dos seus diversos componentes.

Outro conhecimento relevante necessário é o desenvolvimento de aplicações móveis, incluindo princípios de design de interface do utilizador. O Xamarin.Forms é um conjunto de ferramentas de UI que permite a criação de uma única code-base de UI compartilhada para aplicações iOS e Android, e é importante ter conhecimento de como pode ser feita a sua utilização.

Muitas aplicações Xamarin dependem, também, de APIs e serviços web para acesso de dados e funcionalidades, sendo importante ter conhecimento sobre como trabalhar com APIs e serviços web em aplicações Xamarin. De modo mais abstrato, é de salientar também a importância de ter já alguma compreensão

sólida sobre do desenvolvimento de software em geral, incluindo o planeamento, desenvolvimento, teste e implantação de uma aplicação. É fundamental conhecer padrões de desenvolvimento de software e uma boa prática, como testes variados e recorrentes, controlo de versões, de modo a garantir a qualidade e eficácia da aplicação desenvolvida.

Outro aspeto a ter em conta será a compreensão da plataforma de uso, ou seja, o sistema operativo onde a aplicação será executada. É necessário conhecer as particularidades do iOS e do Android, os diferentes dispositivos associados e resoluções de ecrã, além das diretrizes de design de cada plataforma.

Por fim, será importante manter-se atualizado com as novidades e atualizações do Xamarin e da plataforma .NET, aprendendo novos conceitos, bibliotecas, frameworks e ferramentas para melhorar o desenvolvimento e o desempenho da aplicação.

4 Arquiteturas e Diagrama de fluxo

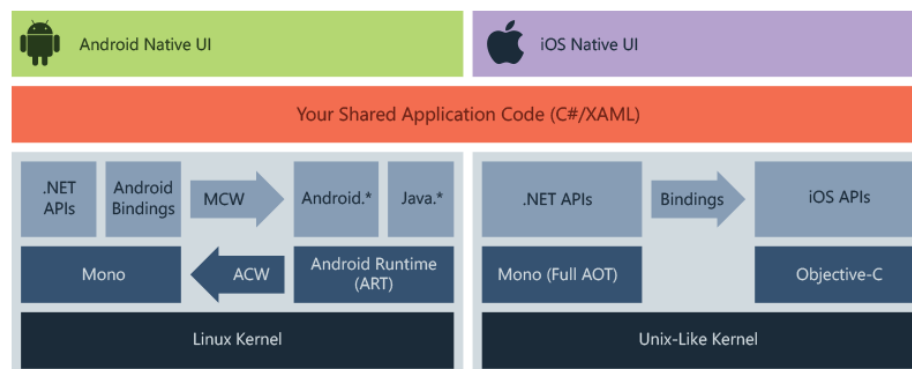
O Xamarin segue uma arquitetura model-view-controller (MVC) que separa a lógica da aplicação em três componentes: Modelo (dados), Visão (interface do usuário) e Controlador (gere a interação entre o utilizador, o Modelo e a Visão).

O fluxo de trabalho envolve escrever código em C# e usar as bibliotecas Xamarin.Forms ou Xamarin.iOS e Xamarin.Android.

4.1 Arquitetura Geral

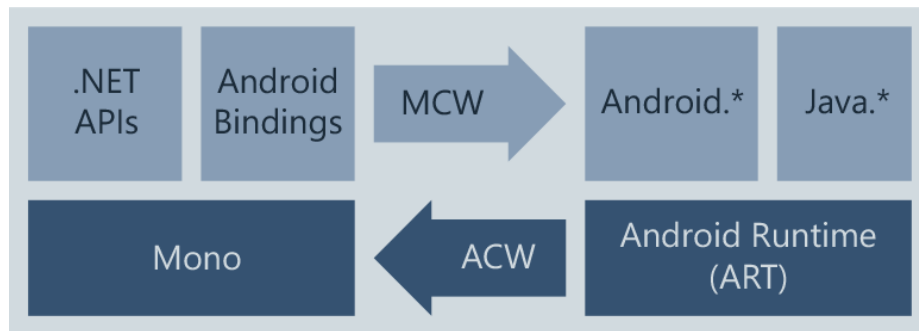
O diagrama seguinte ilustra a arquitetura global de uma aplicação cross-platform Xamarin. Xamarin permite criar uma interface UI nativa em cada plataforma e escrever lógica empresarial em C# partilhada entre plataformas. Geralmente, 80% do código da aplicação é partilhável.

Xamarin é construído sobre .NET, que trata automaticamente de tarefas tais como alocação de memória, recolha de lixo e interoperabilidade com plataformas subjacentes.



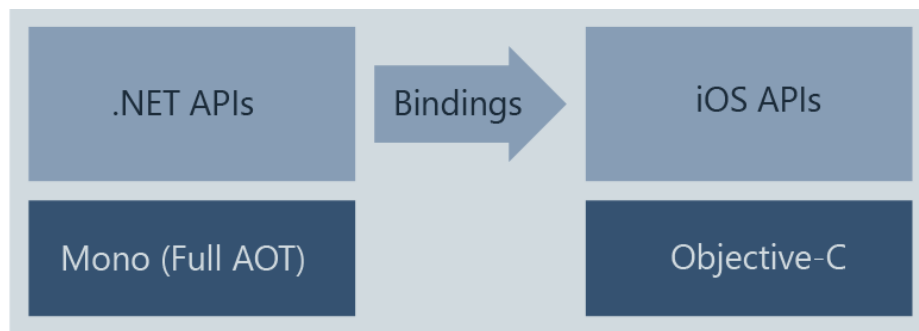
4.2 Arquitetura Xamarin.Android

Aplicações Xamarin.Android compilam de C# para Linguagem Intermédia (IL) que é depois compilada Just-in-Time (JIT) para um assembly nativo quando a aplicação é lançada. As aplicações Xamarin.Android correm dentro do ambiente de execução Mono, lado a lado com a máquina virtual Android Runtime (ART). Xamarin fornece ligações .NET para os namespaces Android.* e Java.*. O ambiente de execução Mono chama para estes namespaces via Managed Callable Wrappers (MCW) e fornece o Android Callable Wrappers (ACW) ao ART, permitindo a ambos os ambientes invocar código um no outro.



4.3 Arquitetura Xamarin.iOS

As aplicações Xamarin.iOS são compiladas em Ahead-of-Time (AOT) partindo de C# para código assembly ARM nativo. Xamarin utiliza Selectores para expor Objective-C a managed C# e Registrars para o inverso, estes dois denominam-se de bindings permitem a comunicação entre Objective-C o C#.



4.4 Diagrama de fluxo Xamarin.Forms

Xamarin.Forms é uma framework open-source de UI, permite aos programadores construir aplicações Xamarin.iOS, Xamarin.Android, e Windows a partir de um mesmo código de base partilhado. Permite também a criação de interfaces de utilizador em XAML com o code-behind em C#, que são renderizadas como sendo nativas de cada plataforma.

5 Vantagens e limitações

5.1 Vantagens

- **Partilha de código:** A plataforma Xamarin permite que os programadores escrevam o código da aplicação uma única vez e o utilizem em diferentes plataformas, como Windows, iOS e Android. Isso poupa tempo, esforço e dinheiro, pois o código não precisa ser reescrito para cada plataforma separadamente. Além disso, a partilha de código também ajuda a manter a consistência e uniformidade entre as diferentes aplicações desenvolvidas.
- **Suporte técnico:** A plataforma Xamarin é suportada pela Microsoft e, portanto, oferece excelente suporte técnico. Os programadores podem aceder a uma ampla gama de recursos e ferramentas, além de cursos para melhorar conhecimentos. Isso ajuda a garantir que os programadores possam resolver problemas rapidamente e manter as suas aplicações atualizadas e a funcionar sem problemas.
- **Manutenção simplificada:** O desenvolvimento de aplicações usando o Xamarin é mais simples do que se espera tendo em conta tudo o que já foi mencionado. Isso ajuda a reduzir a quantidade de tempo e esforço necessário para manter as aplicações. A comunidade Xamarin está ativamente envolvida em ajudar os programadores a fornecer documentação e opções de treino para facilitar a manutenção das aplicações.
- **Experiência do utilizador nativo:** O Xamarin oferece acesso completo a todas as APIs e kits de ferramentas nativos usados em diferentes plataformas, como Windows, iOS e Android. Isso garante que os programadores possam criar aplicações com desempenho e design nativos para cada plataforma. Além disso, o Xamarin oferece uma experiência de utilizador nativa em cada aplicação desenvolvida com ele, garantindo que os utilizadores finais tenham uma experiência perfeita ao usar a aplicação em diferentes dispositivos.
- **Xamarin.Forms:** O Xamarin sugere duas maneiras de criar aplicações móveis como já mencionado: usando Xamarin.iOS e Xamarin.Android como a abordagem principal ou usando a biblioteca Xamarin.Forms. O Xamarin.Forms é melhor para aplicações com mais foco na partilha de código do que na interface do utilizador. Esta ferramenta permite criar uma única interface e partilhá-la em várias plataformas, embora não seja tão eficiente em termos de aparência nativa.

5.2 Desvantagens

- **Atrasos nas atualizações:** Embora o Xamarin forneça suporte para o desenvolvimento de aplicações móveis em várias plataformas, pode levar algum tempo para que as atualizações sejam lançadas para cada uma delas. Isto

ocorre porque, para cada atualização, é necessário integrar os novos recursos em todo o ecossistema Xamarin, o que pode levar tempo. Isto pode ser uma desvantagem em comparação com outras plataformas, que podem ter atualizações imediatamente disponíveis após o lançamento.

- **Tamanho da aplicação:** O tamanho das aplicações desenvolvidas com o Xamarin pode ser um pouco maior do que as aplicações nativas. Isto ocorre porque o Xamarin usa uma camada intermédia para traduzir o código C# em código nativo para iOS e Android. Isto pode levar a um aumento no tamanho da aplicação e pode causar problemas de gestão de tamanho, principalmente se estiver a desenvolver aplicações com muitas funcionalidades e bibliotecas centralizadas.
- **Comunidade menor:** Embora o Xamarin tenha uma boa comunidade, esta é menor do que outras plataformas de desenvolvimento móvel. Isto pode dificultar encontrar ajuda ou recursos adicionais para resolver problemas específicos. No entanto, a comunidade do Xamarin ainda é ativa e crescente, com muitos recursos disponíveis online, incluindo documentação oficial, fóruns de discussão e grupos de utilizadores.
- **Dificuldades em gráficos complexos:** O Xamarin pode não ser a melhor escolha para aplicações que exigem gráficos complexos e visualização de dados avançada. Embora o Xamarin forneça suporte para bibliotecas gráficas, como o SkiaSharp, pode ser mais difícil alcançar os mesmos níveis de desempenho e funcionalidade em comparação com outras plataformas. No entanto, isto pode mudar à medida que o Xamarin continua a evoluir e a melhorar as suas capacidades de desenvolvimento gráfico.

6 Aplicabilidade, Casos de Uso e Sistemas conhecidos construídos com Xamarin

Aplicações criadas com Xamarin podem usufruir de todas as funcionalidades nativas dos dispositivos móveis, como a câmara, o GPS, o armazenamento interno, entre outros.

Além disso, o Xamarin permite que as empresas criem aplicações personalizadas para as suas necessidades específicas, independentemente do tamanho ou setor em que atuam. Por exemplo, um banco pode desenvolver uma aplicação móvel para os seus clientes acederem às suas contas bancárias e realizar transações financeiras de forma segura e conveniente. Já uma empresa de educação pode criar uma aplicação para os alunos acederem a materiais didáticos e atividades online.

Como o Xamarin é uma plataforma de desenvolvimento de aplicações móveis que permite a criação de aplicações nativas para várias plataformas utilizando uma única base de código as empresas podem poupar tempo e recursos, desenvolvendo aplicações personalizadas para as suas necessidades específicas, tornando

o Xamarin uma escolha popular. Algumas das áreas em que o Xamarin é mais frequentemente usado incluem:

- **Aplicações de negócios:** O Xamarin é frequentemente usado para desenvolver aplicações de negócios para empresas de diversos setores. Essas aplicações podem incluir recursos como gestão de tarefas, relatórios de despesas, gestão de projetos, gestão de inventário e muito mais. Essas aplicações podem ajudar as empresas a aumentar a produtividade, melhorar a eficiência e fornecer melhores serviços aos clientes.
- **Aplicações de jogos:** O Xamarin é frequentemente usado para desenvolver jogos para dispositivos móveis iOS e Android. O Xamarin pode ajudar a acelerar o processo de desenvolvimento, permitindo que os programadores criem jogos de alta qualidade usando uma única base de código. Os jogos criados com Xamarin incluem jogos de ação, puzzles, jogos de simulação e muito mais.
- **Aplicações Práticas:** O Xamarin também é frequentemente usado para desenvolver aplicações utilitárias para iOS e Android, como aplicações de tempo, calendários, calculadoras, aplicações de câmara e muito mais. Essas aplicações podem ajudar as pessoas a gerir as suas vidas diárias de maneira mais eficiente e são muito populares entre os utilizadores de dispositivos móveis.
- **Aplicações educacionais:** O Xamarin é frequentemente usado para desenvolver aplicações educacionais para iOS e Android, incluindo aplicações de aprendizagem de línguas, aplicações de matemática e ciência, aplicações de leitura e muito mais. Essas aplicações podem ajudar os utilizadores a adquirir novas habilidades e conhecimentos de uma maneira fácil e divertida.

De acordo com o StackShare, aqui estão algumas das melhores startups que utilizam Xamarin.

- **Olo:** A Olo é uma das empresas de software mais popular no setor da alimentação e bebida. Fundada em 2005, a Olo está envolvida no desenvolvimento de interfaces digitais para restaurantes, que podem ser usados pelos clientes para fazer pedidos através da aplicação. Alguns dos restaurantes que usam Olo: Denny's, The Cheesecake Factory, Wing Stop, El Poco Loco, etc.
- **Alaska Airlines:** A Alaska Airlines é a quinta maior companhia aérea dos EUA em termos de tráfego de passageiros. Após a fusão com a Virgin America, a empresa opera várias rotas transcontinentais da Costa Leste e da Costa Oeste, além de serviços internacionais para o México, Canadá e Costa Rica.
- **BBVA:** Fundado em 1857, o Banco Bilbao Vizcaya Argentaria (BBVA) é uma empresa multinacional de soluções financeiras de origem espanhola e

uma das instituições financeiras mais proeminentes do mundo.

- **UPS:** Fundada em 1907, a United Parcel Service (UPS) é uma das mais antigas e conhecidas empresas de logística, transporte e gestão da cadeia de suplementos do mundo. É agora a maior empresa de entrega do mundo em termos de receita e opera em mais de 220 países por todo o mundo. A UPS oferece principalmente serviços de transporte aéreo e terrestre, mas também é popular pela sua cadeia de distribuição UPS Store, que fornece serviços a pequenas empresas.
- **FreshDirect:** A FreshDirect é um serviço americano de entrega de supermercado que fornece o seu serviço através de uma aplicação mobile. É uma empresa que produz produtos alimentares e refeições personalizadas, utilizando o método de produção "lean manufacturing" para reduzir o desperdício e manter a qualidade.

7 Xamarin vs. Outras Tecnologias

Ultimamente o Xamarin tem perdido popularidade comparativamente com outras tecnologias de desenvolvimento cross-plaform.

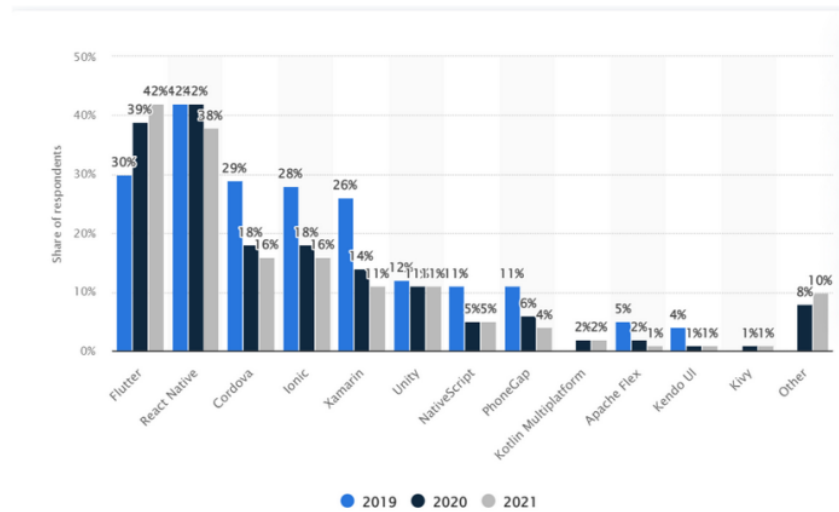
O facto de ser uma tecnologia que requer C# e .NET torna para muitos um começo mais lento comparando com por exemplo JavaScript.

Alguns outros problemas que puderam estar relacionados com a sua perda de popularidade são o pay-wall por detrás de algumas versões no passado e também as diferenças de features e updates em diferentes plataformas, tornando o código mais difícil de manter.

Em comparação com o React Native ambos os frameworks partilham muitas semelhanças. Ambos permitem que os programadores construam aplicações móveis multiplataforma com desempenho próximo do nativo, no entanto, o Xamarin executa o código mais rápido no Android e iOS e possui uma interface do utilizador (UI) para usar ferramentas nativas. Em termos de popularidade, como o React Native é open-source e o Xamarin é da Microsoft, o React Native é mais popular na comunidade de desenvolvimento de software.

Pelos mesmos motivos mencionados em cima, pode esperar-se que o React Native tenha uma comunidade maior e mais solidária do que o Xamarin.

Quando o React Native era recente, a sua comunidade era limitada. No entanto, à medida que o framework cresce e os frameworks JavaScript em geral ganham mais atenção, torna-se cada vez mais fácil identificar outros utilizadores de React Native. O Xamarin e React Native são ambos grátis, embora o Xamarin seja apenas grátis para empresas pequenas. Ao comparar lado a lado estes os dois frameworks, é claro que o React Native ganha ao Xamarin no que toca a comunidade e disponibilidade. No entanto, uma empresa que priorize o desempenho e os tempos de compilação em detrimento de outros fatores, poderá optar pelo Xamarin.



8 Conclusão

Para concluir, é de notar que Xamarin ainda é uma tecnologia poderosa, útil, eficiente e versátil para o desenvolvimento de aplicações cross-platform, especialmente para programadores já familiarizados com a C# e .NET, a Microsoft continua a investir nesta tecnologia e integrando-a também na sua plataforma .NET, podendo ajudar o aumento e a sua adoção no futuro.

No entanto, o Xamarin também tem algumas limitações, como apresentamos anteriormente.

Em comparação com outras frameworks populares, como o Flutter e o React Native, o Xamarin pode não ser a melhor opção para programadores que desejam uma solução mais leve ou para grandes empresas, mas ainda é uma plataforma com algum poder.

References

1. <https://codificar.com.br/o-melhor-e-o-pior-do-xamarin/>
2. <https://www.devmedia.com.br/desenvolvimento-multiplataforma-com-xamarin/33467>
3. <https://www.sam-solutions.com/blog/xamarin-cross-platform-development/>
4. <https://jelvix.com/blog/xamarin-forms-vs-xamarin-native/>
5. <https://www.altexsoft.com/blog/mobile/13-apps-made-with-xamarin-cross-platform-development-in-practice/>
6. <https://blog.back4app.com/pt/xamarin-vs-react-native-segredos-desvendados/>
7. <https://www.manchesterdigital.com/post/foresight-mobile/is-xamarin-dead>
8. <https://blog.back4app.com/pt/startups-e-empresas-que-usam-o-xamarin/>
9. <https://www.trio.dev/blog/xamarin-vs-react-native>
10. <https://nix-united.com/blog/the-complete-guide-to-xamarin-mobile-app-development-in-2023/>