
AGENDA 9

**UTILIZANDO
MICROSOFT
MAUI PARA
DESENVOLVIMENTO
MOBILE**

GEEaD - Grupo de Estudos de Educação a Distância

Centro de Educação Tecnológica Paula Souza

GOVERNO DO ESTADO DE SÃO PAULO

EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO

CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

PROGRAMAÇÃO MOBILE I

Expediente

Autores:

GUILHERME HENRIQUE GIROLI

TIAGO ANTONIO DA SILVA

Revisão Técnica:

Kelly Cristiane de Oliveira Dal Pozzo

São Paulo – SP, 2024

O QUE SÃO DISPOSITIVOS MOBILE?

Dispositivos móveis são aparelhos eletrônicos portáteis que oferecem funcionalidades de computação e comunicação sem fio. Projetados para serem transportados facilmente e utilizados em movimento, esses dispositivos incluem smartphones, tablets, laptops, smartwatches e dispositivos de realidade virtual.

A principal característica dos dispositivos móveis é a sua capacidade de se conectar a redes sem fio, como redes móveis de telefonia celular, Wi-Fi e Bluetooth. Isso permite acesso à internet, comunicação com outros dispositivos e transferência de dados de forma conveniente.

Esses dispositivos desempenham um papel fundamental na vida cotidiana, sendo usados para diversas atividades, incluindo comunicação, entretenimento, trabalho, organização pessoal, navegação na internet, mídia social, fotografia, jogos e muito mais. Sua versatilidade e praticidade os tornaram indispensáveis na era digital.

A história dos dispositivos móveis é fascinante tendo início no século XX.



Contexto Histórico dos Dispositivos Móveis

- ✓ **Telefones celulares analógicos (década de 1980):** Os primeiros telefones celulares analógicos eram grandes e pesados, com funcionalidade limitada. No entanto, permitiam comunicação sem fio de voz em uma ampla área de cobertura.
- ✓ **Telefones celulares digitais (década de 1990):** Com o avanço da tecnologia digital, os telefones celulares tornaram-se menores, mais leves e mais acessíveis. Além de chamadas de voz, passaram a oferecer recursos adicionais como mensagens de texto e jogos simples.
- ✓ **PDA's (Assistentes Pessoais Digitais) (década de 1990 - início dos anos 2000):** Os PDAs foram projetados para organização pessoal e produtividade. Com tela sensível ao toque e caneta stylus, permitiam entrada de dados e gerenciamento de informações. Exemplos notáveis incluem o PalmPilot e o Pocket PC.
- ✓ **Smartphones (meados dos anos 2000 até hoje):** A chegada dos smartphones revolucionou a tecnologia móvel, combinando funcionalidades de telefone celular com avançadas capacidades de computação. Eles oferecem acesso à internet, e-mail, aplicativos, câmeras de alta resolução e muito mais. Marcas como Apple, Samsung e Google competem para lançar dispositivos com inovações tecnológicas e recursos aprimorados.

Nos últimos anos, os smartphones se tornaram ainda mais poderosos, incorporando recursos como reconhecimento facial, inteligência artificial, realidade aumentada e conectividade 5G. Além disso, dispositivos vestíveis, como smartwatches e fones de ouvido inteligentes, têm ganhado popularidade, ampliando as possibilidades de interação móvel.



MERGULHANDO NO TEMA

Como todos já sabem, um dispositivo computacional, independentemente de seu tipo, é composto por duas partes principais: a parte física, denominada **Hardware**, e a parte lógica, conhecida como **Software**. Os dispositivos móveis, como smartphones e tablets, seguem a mesma estrutura. De nada adianta ter potência e design sofisticados sem um sistema operacional que faça o dispositivo ganhar vida!

Com a evolução dos equipamentos, a indústria passou a necessitar de um sistema operacional genérico, que funcionasse em diversos tipos de *hardwares* e estivesse em constante manutenção e evolução, assim como os dispositivos físicos.

Sistema Operacional Android

O sistema operacional Android é uma plataforma de software desenvolvida inicialmente pela Android Inc., que foi fundada em 2003. Em 2005, a Google adquiriu a Android Inc. e passou a liderar o desenvolvimento do sistema operacional.

O Android foi oficialmente anunciado em 2007, juntamente com a formação da Open Handset Alliance. Esta aliança, composta por várias empresas de tecnologia, incluindo fabricantes de dispositivos móveis, desenvolvedores de software e provedores de serviços de telefonia, tinha como objetivo desenvolver padrões abertos para dispositivos móveis e oferecer uma plataforma comum para fomentar a inovação.

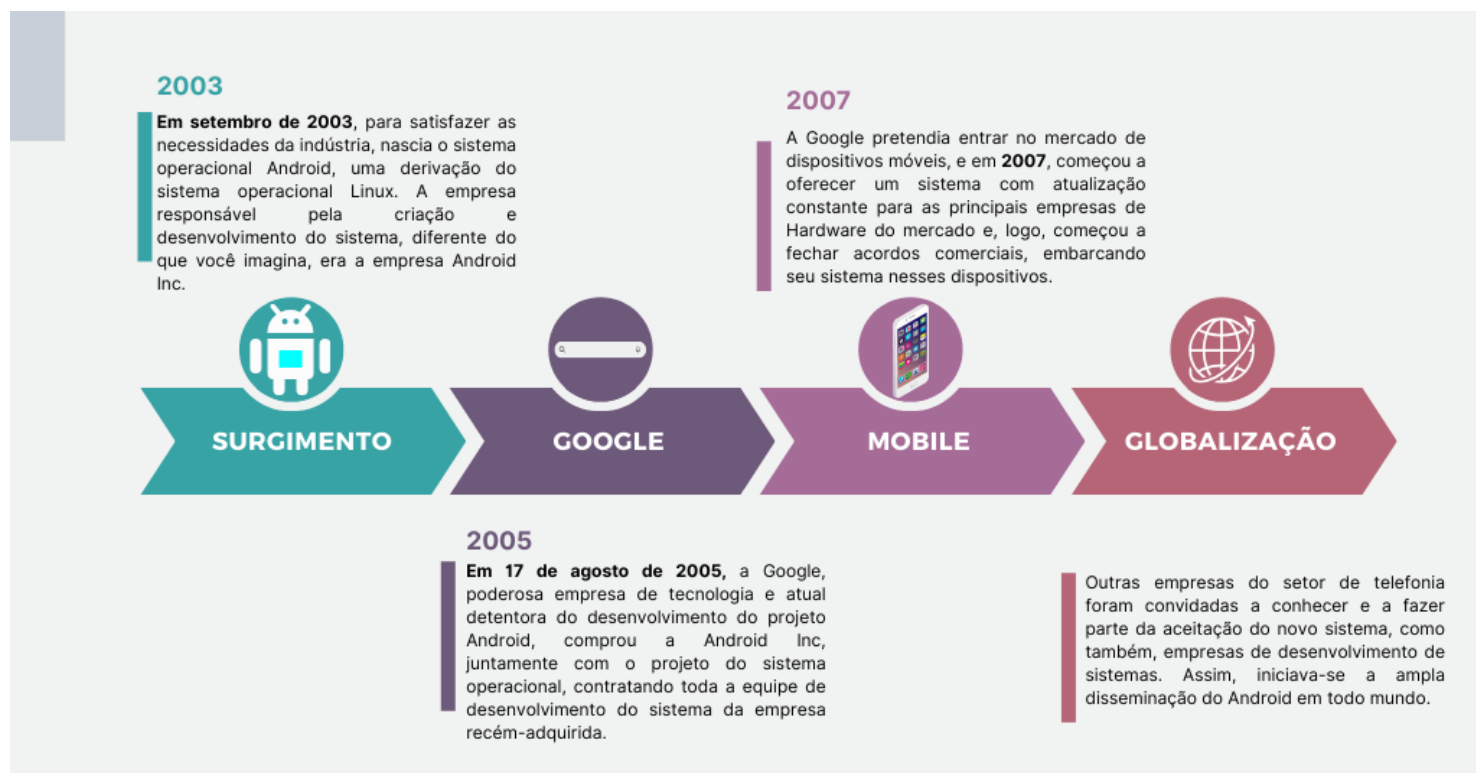


Figura 1 - Origem do sistema Operacional Android.



O sistema operacional Android, tem seu código aberto pela Google, e muitas empresas fabricantes de aparelhos, já trabalham alterando esse código, customizando e otimizando para se adequar aos seus produtos. Alguns dispositivos hoje, são lançados contendo uma versão do sistema que possui seu código aberto e parte sendo proprietário da empresa do aparelho, em virtude das customizações e otimizações.

Com sua ampla utilização, o sistema operacional Android abriu as portas para um novo mercado: o mercado de desenvolvimento mobile. O Android possui uma loja de aplicativos chamada Play Store, onde são comercializadas ferramentas para lazer, trabalho, estudo, entre outras finalidades. Com a popularização da loja virtual da Google, muitos desenvolvedores tiveram a oportunidade de publicar seus trabalhos e ferramentas para todo o mundo, permitindo que muitos saíssem do anonimato e se tornassem milionários.

Outra grande oportunidade de trabalho para o desenvolvedor é criar projetos para fins particulares, como desenvolver aplicativos para empresas, como bancos, lojas, entre outras.

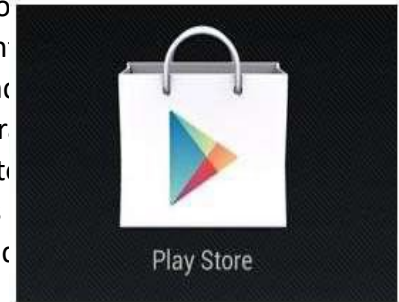


Figura 2 – Aplicativo de Compras da empresa Google, chamado PlayStore.

Sistema Operacional iOS

O iOS é o Sistema Operacional desenvolvido pela Apple especificamente para seus dispositivos móveis. Conhecido por sua interface de usuário intuitiva, segurança, e integração com o ecossistema de produtos e serviços da Apple, como o iCloud, o Apple Watch, o macOS e outros.

Histórico do iOS:



- ✓ **Anúncio e Lançamento (2007):** O iOS foi anunciado pela primeira vez em janeiro de 2007, durante uma apresentação histórica de Steve Jobs, então CEO da Apple, ao introduzir o primeiro iPhone. O sistema operacional foi lançado oficialmente em junho de 2007, junto com o lançamento do iPhone.
- ✓ **Evolução das Versões:** Ao longo dos anos, o iOS passou por diversas atualizações, cada uma trazendo novos recursos, melhorias de desempenho e correções de bugs. Versões marcantes incluem o iOS 2 (2008), que introduziu a App Store; o iOS 4 (2010), que trouxe suporte à multitarefa e o FaceTime; o iOS 7 (2013), que apresentou uma grande reformulação visual; e o iOS 11 (2017), que incorporou o ARKit e aprimorou a multitarefa no iPad.
- ✓ **Expansão para Outros Dispositivos:** Inicialmente desenvolvido exclusivamente para o iPhone, o iOS foi posteriormente expandido para outros dispositivos Apple, como o iPad e o iPod Touch. Essa expansão proporcionou uma experiência consistente entre os diferentes produtos da Apple, facilitando o desenvolvimento de aplicativos compatíveis com múltiplos dispositivos.

- ✓ **Ecossistema e Integração:** Um dos pontos fortes do iOS é sua integração com o ecossistema da Apple, que inclui serviços como iCloud, iTunes e Apple Music. Essa integração permite que os usuários acessem e sincronizem facilmente seus conteúdos em vários dispositivos Apple, proporcionando uma experiência contínua e fluida.
- ✓ **Segurança e Privacidade:** A Apple tem um foco destacado em segurança e privacidade em seus produtos, e o iOS é um exemplo disso. Recursos como Touch ID, Face ID e criptografia de ponta a ponta garantem que os dados dos usuários permaneçam seguros contra acessos não autorizados.
- ✓ **Atualizações Regulares:** A Apple mantém um ciclo contínuo de atualizações do iOS, visando aprimorar a estabilidade, a segurança e o desempenho do sistema. Além disso, essas atualizações trazem novos recursos e funcionalidades, garantindo que os usuários sempre tenham acesso às inovações mais recentes.

No geral, o iOS se tornou uma das plataformas de sistema operacional móvel mais populares e influentes do mundo, oferecendo uma combinação de design elegante, desempenho sólido e uma ampla variedade de aplicativos e serviços.

Agora que já conhecemos os principais sistemas operacionais móveis disponíveis no mercado, vamos explorar o mercado de desenvolvimento para essas plataformas.

Estratégias De Desenvolvimento De Software Para Dispositivos Móveis

Existem diversas alternativas e estratégias para o desenvolvimento de software multiplataforma para dispositivos móveis. Vamos conhecer algumas principais.

Desenvolvimento Nativo

O desenvolvimento nativo envolve a criação de aplicativos específicos para cada plataforma (iOS e Android) utilizando as respectivas linguagens de programação e kits de desenvolvimento (SDKs). No iOS, é comum o uso de Objective-C ou Swift, enquanto no Android, usa-se Java ou Kotlin. Essa abordagem oferece o melhor desempenho e permite o acesso completo aos recursos da plataforma, mas pode ser mais demorada e exigir habilidades específicas para cada ambiente.

Desenvolvimento Híbrido

O desenvolvimento híbrido utiliza tecnologias web como HTML, CSS e JavaScript para criar aplicativos que são encapsulados em contêineres nativos, podendo ser distribuídos através das lojas de aplicativos. Esta estratégia permite que os desenvolvedores reutilizem suas habilidades em web e atinjam várias plataformas com um único código-base. No entanto, podem surgir desafios relacionados ao desempenho e ao acesso limitado aos recursos nativos do dispositivo.

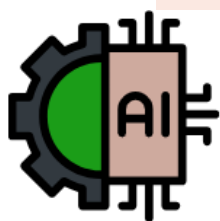
Desenvolvimento Cross-Platform Nativo

Essa abordagem combina os benefícios do desenvolvimento nativo e híbrido. Frameworks como Flutter

(Google), SwiftUI (Apple) e .NET MAUI (Microsoft) permitem que os desenvolvedores escrevam código em uma linguagem comum, compartilhando uma boa parte do código entre as plataformas iOS e Android. Isso resulta em um equilíbrio entre desempenho nativo e eficiência no desenvolvimento, já que reduz a necessidade de escrever código duplicado.

Progressive Web Apps (PWAs)

PWAs são aplicativos web que funcionam de maneira semelhante aos aplicativos nativos, mas são acessados por meio de navegadores. Eles são desenvolvidos com tecnologias web padrão, como HTML, CSS e JavaScript, e podem ser instalados diretamente nos dispositivos dos usuários. Embora os PWAs ainda não ofereçam acesso total a todos os recursos nativos do dispositivo, estão se tornando uma alternativa atraente pela facilidade de desenvolvimento e pela capacidade de alcançar uma variedade de dispositivos.



Utilize ferramentas de Inteligência Artificial para pesquisar mais sobre as diferenças entre desenvolvimento nativo, híbrido, multiplataforma e PWAs. A IA pode ajudar a encontrar rapidamente informações atualizadas e exemplos práticos.

Essas são apenas algumas das principais alternativas e estratégias disponíveis para o desenvolvimento multiplataforma de software para dispositivos móveis. A escolha da abordagem ideal deve considerar fatores como as necessidades específicas do projeto, os recursos técnicos e financeiros disponíveis, além das preferências e habilidades da equipe de desenvolvimento. Avaliar cuidadosamente esses aspectos é fundamental para garantir o equilíbrio entre desempenho, custo e eficiência no desenvolvimento de aplicativos móveis.

Desenvolvimento Cross Platform com .NET MAUI

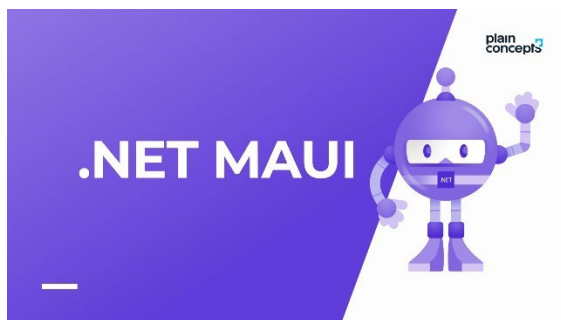


Figura 1 – Apresentação .NET MAUI

Fonte: <https://www.plainconcepts.com/net-maui/>

Utilizando a estratégia de desenvolvimento cross-platform, o .NET MAUI (Multi-platform App UI) é uma estrutura de desenvolvimento gratuita e de código aberto da Microsoft que permite a criação de aplicativos

multiplataforma com um único código-base em C# e XAML. Considerado a evolução do Xamarin.Forms, o MAUI faz parte do ecossistema .NET, distribuído sob a licença MIT. Ele simplifica o desenvolvimento de interfaces de usuário (UI) e oferece suporte a recursos nativos de cada plataforma, sendo ideal para projetos comerciais e não comerciais.

O .NET MAUI traz uma abordagem moderna permitindo a criação de aplicativos nativos para múltiplas plataformas, como iOS, Android, Windows e MacOS, utilizando uma base de código comum. Como evolução do Xamarin.Forms, o MAUI se consolida como um framework robusto para o desenvolvimento de soluções nativas de forma eficiente e escalável.

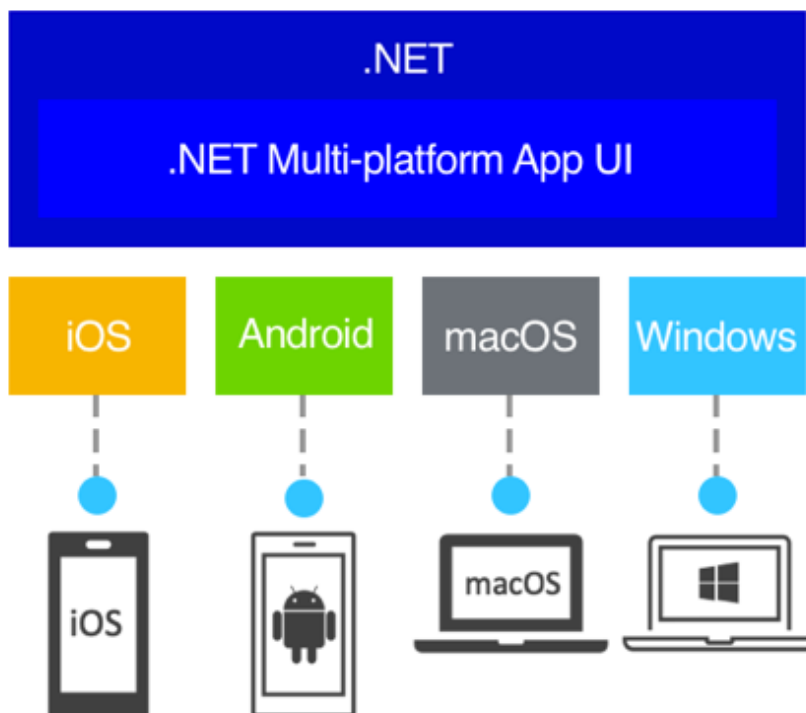


Figura 2 - Diagrama exemplificando o desenvolvimento com MAUI para várias plataformas.
Fonte: <https://learn.microsoft.com/pt-br/dotnet/maui/what-is-maui?view=net-maui-8>.

Saiba mais sobre o .NET acessando o site:

[https:// www.plainconcepts.com/net-maui/](https://www.plainconcepts.com/net-maui/)



Vejamos os pontos-chave sobre a estratégia de desenvolvimento cross-platform com .NET MAUI:

- 1. Base de Código Comum:** Com o MAUI, os desenvolvedores podem utilizar uma única base de código, escrita em C# e com o framework .NET, compartilhando grande parte desse código entre diferentes plataformas. Isso reduz significativamente o esforço de desenvolvimento, permitindo que as equipes criem aplicativos nativos para iOS, Android, Windows e macOS a partir de uma única base de código.
- 2. UI Flexível e Adaptável:** O MAUI oferece uma abordagem flexível para a interface de usuário (UI), garantindo que os aplicativos mantenham consistência visual em todas as plataformas, ao mesmo tempo em que aproveitam os padrões de design nativos de cada uma. Isso é possível por meio de recursos como layouts flexíveis, estilos adaptáveis e controles personalizáveis.
- 3. Integração com o Ecossistema .NET:** Como parte do ecossistema .NET, o MAUI se beneficia de uma vasta gama de ferramentas, bibliotecas e recursos disponíveis para desenvolvedores, incluindo suporte robusto para depuração, testes, controle de versão e integração contínua.
- 4. Suporte a Novas Plataformas:** Além das plataformas tradicionais, como iOS, Android, Windows e macOS, o MAUI está expandindo seu suporte para outras, como Linux e WebAssembly (WASM), ampliando ainda mais as possibilidades de desenvolvimento cross-platform.
- 5. Compatibilidade com Xamarin.Forms:** O MAUI é uma evolução do Xamarin.Forms e foi projetado para ser altamente compatível com aplicativos já desenvolvidos com o Xamarin.Forms. Isso facilita a migração gradual desses aplicativos para o MAUI, permitindo que as equipes aproveitem novas funcionalidades e melhorias da plataforma.

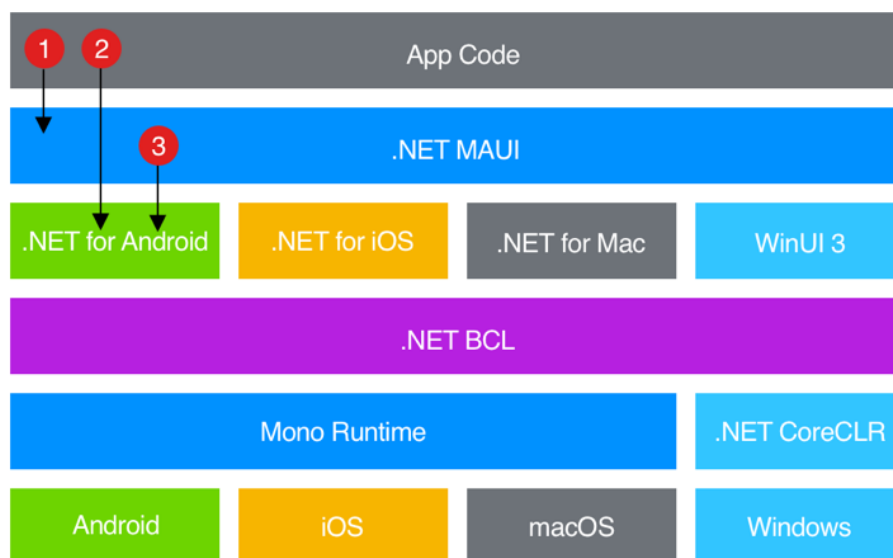


Figura 3 - Arquitetura de Funcionamento do MAUI.

Fonte: O que é o .NET MAUI? - .NET MAUI | Microsoft Learn.

No geral, a estratégia de desenvolvimento cross-platform com MAUI oferece uma solução poderosa e eficiente para a criação de aplicativos nativos em várias plataformas, maximizando a produtividade, reduzindo custos e permitindo que as equipes alcancem um público mais amplo.

Visual Studio 2022 e Desenvolvimento Cross Plataforma com Maui

O desenvolvimento de aplicativos cresce a cada ano, gerando à falta de mão de obra qualificada.

O Visual Studio 2022 é a versão mais recente do popular Ambiente de Desenvolvimento Integrado (IDE) da Microsoft, oferecendo uma ampla gama de ferramentas e recursos que permitem aos desenvolvedores criar aplicativos para diversas plataformas, como desktop, web, mobile e nuvem.

No contexto do desenvolvimento Cross-Platform com .NET MAUI, o Visual Studio 2022 desempenha um papel crucial ao fornecer suporte completo e integrado para a criação de aplicativos nativos em iOS, Android, Windows e MacOS, a partir de uma única base de código. Esse suporte simplifica o processo de desenvolvimento, permitindo que os desenvolvedores maximizem sua produtividade ao reutilizar código entre diferentes plataformas, sem comprometer a experiência nativa de cada uma delas.

Alguns recursos e funcionalidades do Visual Studio 2022 relacionados ao desenvolvimento Cross-Platform com MAUI:



Suporte Nativo para MAUI: O Visual Studio oferece suporte nativo para o desenvolvimento de aplicativos com .NET MAUI, permitindo que os desenvolvedores criem, editem e gerenciem projetos diretamente no IDE. Isso simplifica a configuração do ambiente de desenvolvimento e agiliza o fluxo de trabalho, facilitando o desenvolvimento de aplicativos multiplataforma.



Templates e Projetos MAUI: O Visual Studio oferece uma variedade de templates e projetos pré-configurados para iniciar rapidamente o desenvolvimento de aplicativos MAUI. Esses templates fornecem estruturas iniciais para diversos tipos de aplicativos, como listas, formulários, entre outros, economizando tempo na configuração do projeto e acelerando o início do desenvolvimento.



Depuração e Teste: O Visual Studio oferece ferramentas de depuração e teste para aplicativos MAUI, permitindo que os desenvolvedores identifiquem e corrijam problemas de código de forma eficiente. A IDE também oferece suporte para testes unitários, de interface do usuário e de integração, garantindo a qualidade e confiabilidade dos aplicativos desenvolvidos.



Integração com Git e Serviços de DevOps: O Visual Studio inclui integração nativa com sistemas de controle de versão, como Git, e com serviços de gerenciamento de ciclo de vida de software, como Azure DevOps. Isso facilita a colaboração entre as equipes de desenvolvimento, o gerenciamento do código-fonte e a automação de processos de build e deploy.

Com seu conjunto abrangente de recursos e suporte integrado para MAUI, o Visual Studio simplifica o desenvolvimento de aplicativos nativos para diversas plataformas a partir de uma única base de código, aumentando a produtividade e acelerando o tempo de lançamento no mercado.

Vamos iniciar realizando o download da IDE Visual Studio 2022.

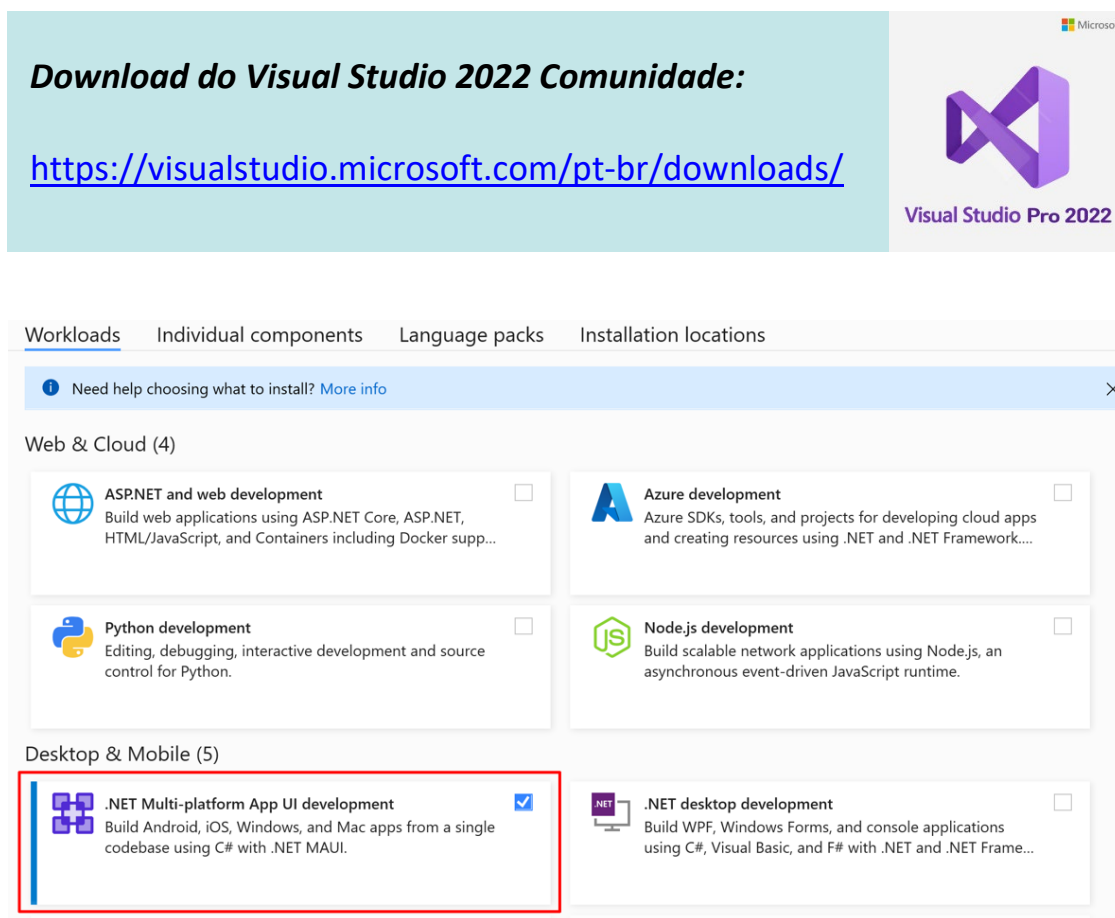


Figura 4 - Pacotes Necessários na Instalação do Visual Studio 2022 para utilização do MAUI.

Fonte: <https://learn.microsoft.com/pt-br/dotnet/maui/get-started/installation?view=net-maui-8.0&tabs=vswin>

Assista o vídeo apresentando o passo a passo para instalação do Visual Studio 2022:



Disponível em: <https://youtu.be/xqHyY0G11ek?si=63GqNx4hZ4FAjQjG>