

[< VOLTAR](#)

Preparação do ambiente de desenvolvimento

Nesta aula vamos aprender como preparar o ambiente de desenvolvimento para podermos criar nossas aplicações com Flutter. Vamos aprender também como emular um smartphone em nosso próprio computador, para executar nossas aplicações.

NESTE TÓPICO

- › Ambiente de desenvolvimento
- › Preparação do ambiente Mac
- › Preparação do ambiente Linux
- › Emulação dos dispositivos
- › Referências



Ambiente de desenvolvimento

É muito importante termos o ambiente de desenvolvimento bem preparado para que não tenhamos nenhum tipo de problema durante o processo de criação dos aplicativos.

A preparação do ambiente não é difícil e não exige grandes habilidades em programação, entretanto é preciso ficar muito atento(a) aos passos pacotes necessários para cada etapa do processo de preparação.

O ambiente de desenvolvimento em Flutter possui diferentes processos de preparo, a depender de seu sistema operacional e, por isso, essa aula é dividida em três grandes grupos: Preparação no Windows, preparação no Linux e preparação no Mac.

É muito importante ressaltar que apesar do Flutter gerar, com um único código, um aplicativo para iOS e um aplicativo para Android, se você não possui um computador com macOS não será possível emular um dispositivo tipo iPhone e nem mesmo executar seus projetos em seu celular com iOS, pois é necessário ter o xCode para isso e essa ferramenta funciona exclusivamente no macOS. Mas não se preocupe, pois o mesmo código poderá ser emulado e executado em qualquer smartphone com Android, seja emulado ou físico (seu próprio celular).

É IMPORTANTE RESSALTAR

O Flutter é uma tecnologia bastante nova e está em constante atualização. O processo de instalação e preparação do ambiente pode mudar a qualquer momento em relação a este material e, por isso, recomenda-se fortemente que você confira o passo-a-passo do processo de instalação no próprio site do Flutter, [neste link](https://flutter.dev/docs/get-started/install) (<https://flutter.dev/docs/get-started/install>).

Outro ponto importante a ser considerado é que para emular um smartphone em seu computador é preciso atentar-se aos requisitos mínimos de memória, processador, tipo de sistema operacional (64 bits), espaço em disco (precisa de bastante espaço) entre outros fatores. É importante falarmos sobre isso pois existe um mito de que desenvolvimento mobile precisa de máquina muito forte, o que não é necessariamente verdade. Para desenvolvimento nativo sim, é preciso ter um computador com uma maior disponibilidade de recursos de memória, processador e disco. Para o desenvolvimento híbrido (independentemente da linguagem ou tecnologia utilizados) você precisará de um computador com menos recursos, mas ainda precisará atentar-se a isso na página oficial do Flutter.

Embora o processo de instalação do Flutter seja muito parecido em todos os sistemas operacionais, **você pode utilizar apenas a preparação do seu próprio sistema operacional**, ou seja, não precisa se preocupar com a preparação de ambientes em que você não possui.



Lembre-se, ainda, que o processo de preparação do ambiente pode variar muito a depender de configurações prévias em seu computador. Qualquer desvio ou erro no processo de preparação do ambiente deve ser consultado em fontes oficiais, fóruns e, naturalmente, sites de busca. Uma dica importante aqui é: se você obtiver qualquer erro durante a instalação, realize buscas em inglês pois o material e ajudas estão concentrados nessa linguagem. Bom, vamos começar a preparação do ambiente.

Preparação do ambiente Windows

Para a preparação do ambiente utilizando Windows, neste caso, fora utilizado o Windows 10 pra realização dos procedimentos. Se você possui qualquer versão anterior a esta, recomenda-se fortemente verificar no site oficial do Flutter as alternativas.

SE VOCÊ USA WINDOWS 7 OU 8!

Se você utiliza Windows 7 ou Windows 8 (qualquer subversão), antes de começar, faça o download e instalação do Powershell mais recente, disponível [neste link](https://docs.microsoft.com/pt-br/powershell/scripting/windows-powershell/install/installing-windows-powershell?view=powershell-7.1) (<https://docs.microsoft.com/pt-br/powershell/scripting/windows-powershell/install/installing-windows-powershell?view=powershell-7.1>).

É muito importante observar também os requisitos mínimos de sistema, disponíveis no site oficial do Flutter. No momento em que este material foi produzido, os requisitos são:

- **Sistema operacional mínimo:** Windows 7 SP1 or later (64-bit), x86-64 based
- **Espaço em disco:** Ao menos 1.32 GB (ainda precisará de mais espaço para os projetos e ferramentas de desenvolvimento)
- **Ferramentas necessárias (devem estar pré-instaladas):**
 - Powershell (apenas para Windows 7 ou 8, conforme destaque acima)
 - Git ? Não é necessariamente obrigatório, mas é importante tê-lo

Note acima que o sistema operacional mínimo precisa ser de 64 bits, ou seja, se seu sistema operacional for de 32 bits, o processo não funcionará, como praticamente qualquer linguagem de programação moderna, infelizmente.

Para deixar mais prático e intuitivo o processo, assista o vídeo abaixo que mostra como preparar o ambiente Windows para desenvolvimento com Flutter.

Preparação do ambiente Windows, Parte 1



Assista o vídeo pausadamente pois ele foi desenvolvido com as etapas que demoram um pouco mais capturadas de forma "acelerada". O vídeo anterior mostra como preparar o ambiente de desenvolvimento no Windows, mas ainda é necessário configurar o seu editor de código preferido, ou seja, é preciso escolher uma IDE (*Integrated Development Environment*, ambiente de desenvolvimento integrado).

O Google (proprietário do Flutter) recomenda fortemente que seja utilizado o próprio Android Studio para desenvolvimento com o Flutter, o que é realmente uma boa opção pois há muitas otimizações nessa IDE.

Por outro lado, existem algumas alternativas como a IntelliJ e o Visual Studio Code. A única desvantagem dessas IDEs é que você precisará, muitas vezes, realizar uma série de configurações no projeto manualmente e via terminal, coisa que o próprio Android Studio já está pronto para o fazer.

Para configurar sua IDE de preferência, assista o curto vídeo abaixo que mostra como configurar o Android Studio e Visual Studio Code, muito conhecido também como "VSCode".

Instalação e configuração do Visual Studio Code com Flutter



O vídeo acima mostra como configura o Android Studio e o VS Code em relação aos Plugins necessários. Lembre-se que você não precisa ter os dois instalados, ok? Apenas um. O vídeo abaixo mostra uma outra alternativa de IDE, o IntelliJ, que é bastante simples, mas ao mesmo tempo muito robusto, com recursos tão avançados quando o do Android Studio (na verdade o Android Studio é baseado nessa IDE).

Instalação e configuração da IntelliJ com Flutter



POR QUE EU PRECISO DO ANDROID STUDIO?

Apesar de você poder utilizar o VS Code, IntelliJ ou qualquer outra IDE de preferência, o Android Studio fornece ao Flutter recursos de compilação (criação do executável) específicos para o projeto Android. Quando formos entender melhor a arquitetura do Flutter você verá que seu código, na verdade, é convertido para uma aplicação Android e iOS e, por isso, você precisa das ferramentas de desenvolvimento específicas de cada plataforma instaladas.

Outro ponto importante que você deve estar se perguntando é: Professor, mas e o emulador de Android? Muito simples, como a emulação é realmente igual em qualquer sistema operacional, há uma sessão (lobo abaixo) sobre como criar um dispositivo virtual Android e iOS.

Preparação do ambiente Mac

Para a preparação deste ambiente com macOS, fora utilizado a versão Big Sur do sistema operacional da Apple. O processo embora seja muito parecido com o ambiente Windows, requer bastante atenção em algumas etapas, especialmente a edição do arquivo `?.bash-profile?` do usuário e, por isso, deve ser feita com muito cuidado.

Assista o vídeo abaixo para ver como preparar o ambiente de desenvolvimento em um macOS e não se esqueça de manter-se atualizado no site oficial do Flutter, caso o processo tenha mudado.



Preparação do ambiente macOS



Preparação do ambiente Linux

Para a realização destes procedimentos fora utilizado o Ubuntu, entretanto qualquer distribuição Linux será muito parecida com o procedimento, devido à sua arquitetura. É muito importante verificar no site oficial do Flutter o processo correto para sua versão de Linux.

A preparação de um ambiente Linux é bastante simples, mas requer muita atenção durante o processo e pode ser vista no vídeo abaixo.

Preparação do ambiente de desenvolvimento com Flutter no Linux (Ubuntu)



Emulação dos dispositivos

Agora que já temos o ambiente de desenvolvimento pronto, temos que saber como executar nossos projetos, não é mesmo? Para isso, essa sessão apresentará como isso pode ser feito de forma virtual, ou seja, como você pode criar um dispositivo virtual em seu computador para executar suas aplicações.

É muito importante nos atentarmos a alguns pontos cruciais para o sucesso do processo. Não podemos negar que o processo de emulação do dispositivo requer uma elevada disponibilidade de recursos no computador, especialmente memória RAM, entretanto não há nenhum consenso sobre a quantidade mínima necessária, se não a exigida pelo próprio Android Studio (4GB mínimo, 8GB recomendado, atualmente), mas normalmente é preciso mais.

Para computadores com Windows, Linux e Mac você poderá emular o Android diretamente pelo Android Studio e para isso basta seguir os passos do vídeo abaixo que, embora desenvolvido em um macOS, o processo em qualquer sistema operacional é igual, pois é interno ao Android Studio.

Mas antes de começar é muito importante saber que para que a emulação funcione corretamente é preciso que seu computador tenha a opção de virtualização ativada na BIOS (configurações iniciais de hardware). Para isso, cada fabricante de chipset coloca a configuração de uma forma e, neste

caso, você precisa verificar exatamente como ativar na sua, especificamente. Não é difícil, normalmente as opções são nomeadas como ?hardware virtualization? ou ?hyper-v?, ou ?hardware emulation?. E sim, infelizmente existem algumas BIOS que não suportam a virtualização e, portanto, você não conseguirá emular um celular no computador, mas poderá utilizar seu próprio celular para rodar seus aplicativos.

Outro ponto muito importante a ser considerado é que emular um dispositivo tipo iPhone é exclusivo para hardwares da Apple (com o macOS). O mesmo vale para celulares físicos da Apple: Você somente conseguirá instalar suas próprias aplicações se tiver um macOS com XCode instalado.

Setup de uma AVD



E se você quiser executar seus projetos em um celular físico (real), não deixe de assistir o vídeo abaixo que mostra como habilitar o Android para executar as aplicações.

Como habilitar o modo desenvolvedor no Android



Bom, esperamos que você tenha conseguido realizar todos os passos para a preparação do ambiente. Lembre-se que não faltam fóruns de discussão e, certamente, você encontrará uma boa solução para qualquer desvio no processo. E vamos em frente, que agora é hora de começar a programar.

Quiz

Exercício Final

Preparação do ambiente de desenvolvimento

INICIAR ➤



Referências

BRACH, Gilard, LARS, Bak. Dart: a new programming language for structured web programming. **GOTO Conference**, 10 out. 2011. Disponível em: https://gotocon.com/dl/goto-aarhus-2011/slides/GiladBracha_and_LarsBak_OpeningKeynoteDartANewProgrammingLanguageForStructuredWebProgramming.pdf. Acesso em: 12 nov. 2020.

DART. **Dart documentation**. *Site*. Disponível em: <https://dart.dev/>. Acesso em: 12 nov. 2020.

FELIX, Rafael. **Programação orientada a objetos**. São Paulo: Pearson, Ed. Pearson, 2017. *E-book*. Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Loader/128217/pdf>. Acesso em 19 nov. 2020.

OKEDIRAN, O. O. *et al.* Mobile operating systems and application development platforms: a survey. **Int. J. Advanced Networking and Applications**. v.6, n.1, p. 2195-2201, july-aug. 2014. Disponível em: <https://www.ijana.in/download%206-1-9.php?file=V6I1-9.pdf>. Acesso em: 12 nov. 2020.

SCHWARZMÜLLER, Maximilian. **Learn Flutter and Dart to Build iOS and Android Apps 2020**. Oreilly, Packt Publishing, 2020. *Vídeo*. Disponível em: <https://learning.oreilly.com/videos/learn-flutter-and/9781789951998/>. Acesso em 19 nov. 2020.

SINHA, Sanjib . **Quick start guide to Dart programming: create high performance applications for the web and mobile**. Lompoc, CA, EUA: Apress, 2019. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/quick-start-guide/9781484255629/>. Acesso em: 12 nov. 2020.

WINDMILL, Eric. **Flutter in action**. Nova Iorque: Manning publications, 2020. *E-book*. Disponível em: <https://learning.oreilly.com/library/view/flutter-in-action/9781617296147/>, Acesso em: 12 nov. 2020.



Avalie este tópico



ANTERIOR

Coleções com Mapas e Listas em DART

Biblioteca

([https://www.uninove.br/conheca-](https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/)

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site



Índice

Ajuda?

([https://ava.un](https://ava.uninove.br/)

Introdução ao Flutter

IdCurso=)



© Todos os direitos reservados

