

[Home](#) / [Artigos](#) / Flutter - Instalação e Configuração do Android no Windows

Flutter - Instalação e Configuração do Android no Windows

🕒 10 mins ⓘ Nível Fundamental

Para executar este tutorial você vai precisar do **Windows 10**. Você pode testar por conta e risco em versões anteriores, porém recomendamos a versão 10 por ser mais estável e recente.

Para executar os comandos, neste tutorial utilizamos o **Windows Terminal**, mas você pode utilizar o **Power Shell** que já vem nativo no Windows 10.

Para o download do Flutter, utilizaremos o **Git**, que está disponível pelo site oficial. Sua instalação é simples e rápida, só ir avançando.

Em adicional, nosso editor de código preferido é o **Visual Studio Code**, que é leve e simples de utilizar. Basta ir para seu site oficial e fazer o download dele também.

Android Studio

A maneira mais fácil de começar a instalação do Flutter é por meio do Download do Android Studio, uma IDE para desenvolvimento de aplicativos para o Android.

Embora não recomendamos os uso do Android Studio para a codificação em si, sua instalação trará a SDK do Android e Java já necessárias para instalação do Flutter.

Como as versões de SDK do Android dependem de versões específicas do Java, esta acaba sendo a melhor opção para começarmos.

Vamos então fazer o Download do Android Studio em sua página inicial e executar sua instalação.

A instalação é relativamente simples, basta ir prosseguindo sem muitos mistérios. Recomendamos **NÃO ALTERAR** nenhuma opção durante a instalação.

A única tela que você precisa prestar atenção é a Verify Settings, que conterà o caminho da SDK do Android e da JDK do Java, que utilizaremos em breve aqui.



Verify Settings

If you want to review or change any of your installation settings, click Previous.

Current Settings:

Setup Type:
Standard

SDK Folder:
C:\Users\balta\AppData\Local\Android\Sdk

JDK Location:
C:\Program Files\Android\Android Studio\jre

Previous

Next

Cancel

Finish

Anote estes caminhos ou utilize o Print Screen para capturar a tela.

Adicionando o Java ao ambiente

A maioria dos comandos do Android rodam via terminal, ou seja, pelo CMD ou Power Shell ou qualquer outro Shell que esteja utilizando.

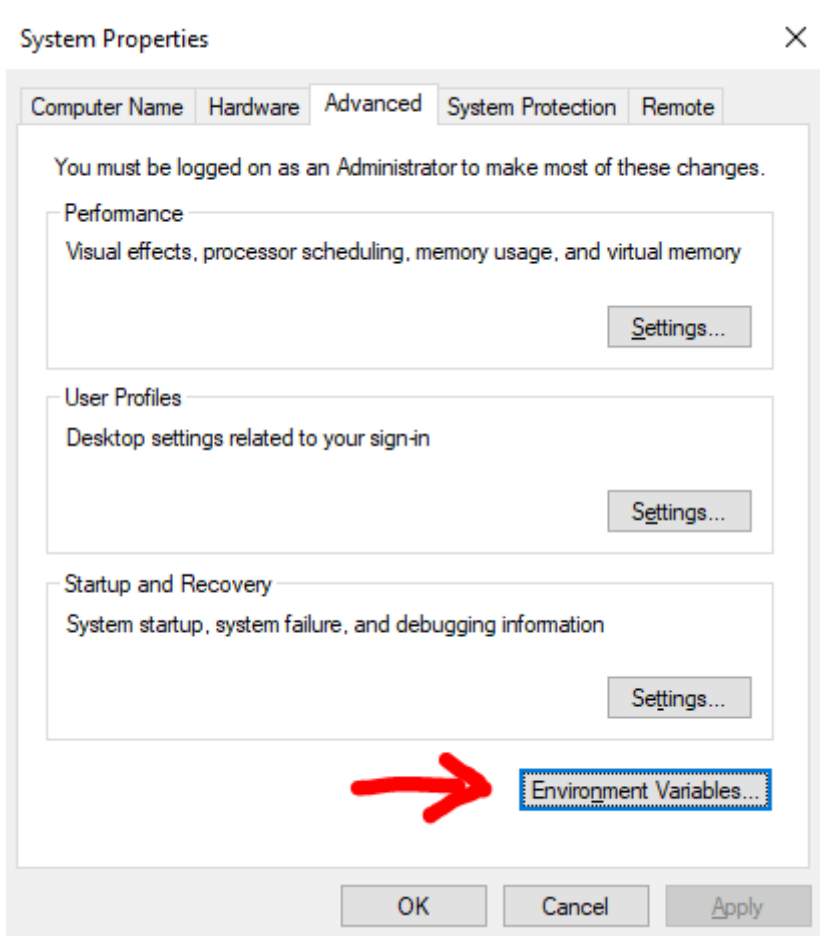
O Android também precisa do Java para executar seus comandos, e sempre que instalamos o Java (JDK ou JRE), temos o **java.exe** disponível na pasta da instalação.

Porém, mesmo com o Java instalado, o Windows não sabe onde ele se encontra. Na verdade, por padrão, o Windows só olha nas pastas System e System32 para execução de comandos.

Em suma, se você abrir um terminal e digitar o nome de qualquer executável que se encontra nestas pastas mencionadas, ele será chamado imediatamente.

Para os executáveis que ficam fora destas pastas, temos as **Variáveis de Ambiente**, que dizem para o Windows olhar também em outras pastas, além das Systems da vida.

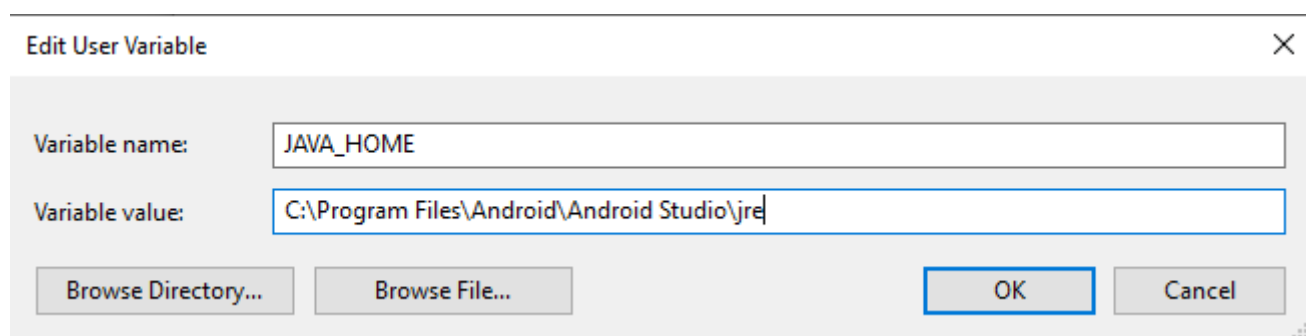
Para encontrar as variáveis de ambiente basta apertar a tecla **Janela do Windows** do seu teclado e digitar "Variaveis de Ambiente" ou "Environment Variables" caso esteja em inglês.



Na nova janela que se abrirá, vamos clicar em "Variáveis de Ambiente" ou "Environment Variables" caso esteja em inglês, e uma nova tela se abrirá.

Na sessão "Variáveis do usuário" ou "User Variables", adicione um novo valor utilizando o botão "Novo" ou "New" contendo a chave **JAVA_HOME** seguido pelo caminho onde o JDK foi instalado.

O caminho do JDK foi mostrado na imagem **Verify Settings** durante a instalação do Android Studio. Caso não tenha salvo o caminho, por padrão ele será **C:\Program Files\Android\Android Studio\jre**.



Clique em OK mas não feche as variáveis de ambiente ainda, precisamos agora configurar os caminhos que o Android utiliza.

Android Tools

A partir da versão 3.x do Android Studio, algumas mudanças na instalação da SDK ocorreram e o comando **"android"**, previamente utilizado no terminal foi descontinuado.

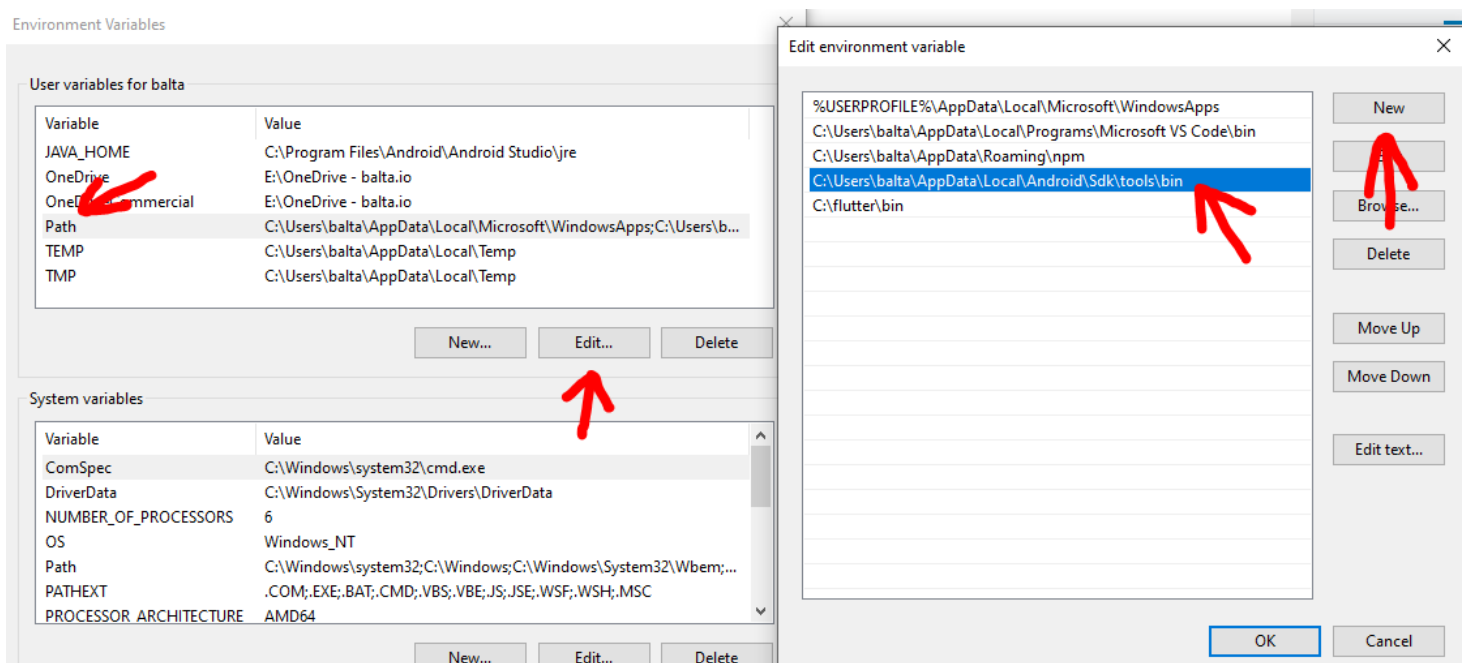
Desta forma, o Android agora traz um conjunto de ferramentas que podemos utilizar, como **SDK Manager**, que gerencia as SDKs do Android instaladas.

Se você não alterou nada durante a instalação, estas ferramentas ficam na pasta

```
C:\Users\USUARIO\AppData\Local\Android\Sdk\tools\bin
```

Este caminho também é mostrado na tela **Verify Settings** durante a instalação do Android. Note que a palavra **USUARIO** no caminho acima deve ser substituída pelo seu nome de usuário do Windows.

Vamos então selecionar o item **PATH** na tela de variáveis de ambiente e clicar em **EDITAR**. Na nova tela que se abre, adicione um novo item utilizando o botão **NOVO** com o **caminho do Android Tools**.



Pronto, isto é tudo que precisamos na parte de variáveis de ambiente por enquanto, mas deixe esta janela aberta pois precisaremos ainda configurar o Flutter.

Instalando o Flutter

A instalação do Flutter é relativamente simples, basta baixar o seu conteúdo para uma pasta em seu computador, e isto pode ser feito de duas formas.

A primeira é fazendo o Download de um ZIP e descompactando e a segunda é clonando o projeto pelo Git, que particularmente, recomendamos.

Um detalhe importante é o caminho onde vamos salvar estes arquivos. É muito importante que este caminho não tenha espaços ou caracteres especiais.

Aliás, uma dica valiosa é sempre criar seu usuário no Windows com um nome simples. No nosso caso nosso usuário chama apenas **balta**, ou seja, sem caracteres especiais ou espaços.

Evite nomes como "José Pereira" ou similares, pois além de caracteres especiais, eles possuem espaços, o que pode afetar a execução de alguns programas mais antigos.

Certifique-se que você possui o Git instalado e abra seu terminal favorito. Particularmente recomendamos o Windows Terminal ou Power Shell, sendo que este último já vem instalado no Windows.

Com o terminal aberto, navegue para uma pasta segura, no caso, recomendamos utilizar a raiz do **C:**, para não ter nenhum problema com diretórios.

Vamos então executar os seguintes comandos:

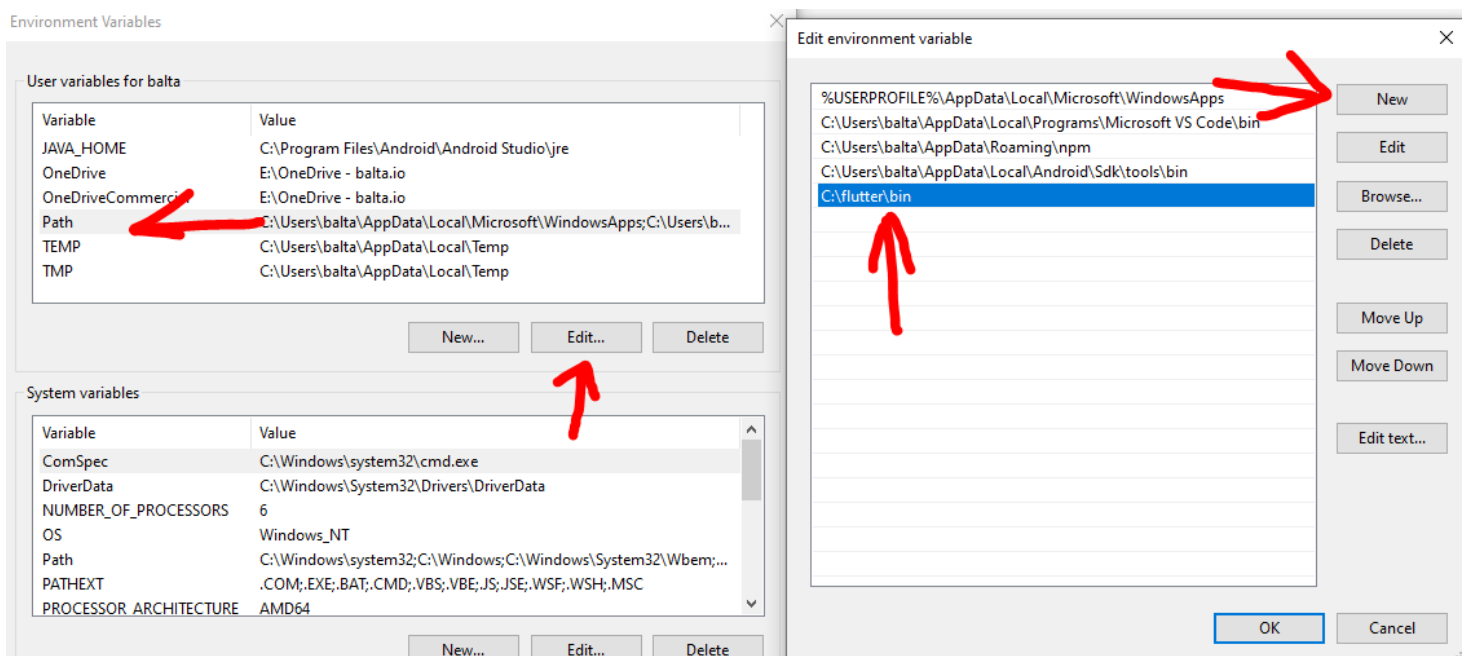
```
cd c:\\
git clone https://github.com/flutter/flutter.git -b stable
```

O primeiro comando navegará para raiz e o segundo fará o download dos fontes do Flutter para a pasta **C:\flutter** que ele criará. Isto é tudo que precisamos no momento.

Adicionando Flutter ao Path

Assim como o Windows não reconhecia os comandos do Java, ele também não reconhece os do Flutter e precisamos novamente alterar nossas variáveis de ambiente.

Vamos agora editar a variável **PATH**, ainda nas variáveis de usuário, e adicionar o caminho para os binários, onde baixamos os fontes do Flutter, que no nosso caso foi **C:\flutter\bin**.



Note que adicionamos o **bin** ao caminho, não se esqueça deste detalhe. Neste momento você pode fechar a tela de variáveis de ambiente e todos os terminais em aberto.

Flutter Doctor

O Flutter possui uma ferramenta chamada doctor que nos permite verificar o estado da instalação dele, e se há algum erro.

Abra um novo terminal e vamos então executar o seguinte comando.

```
flutter doctor
```

Você notará que ele exibirá um erro relacionado a licenças no Android, Plugins e Dispositivos Conectados, não se preocupe pois vamos resolver isto.

Aceitando as Licenças

O primeiro passo é aceitar as licenças do Android e para isto vamos executar o comando.

```
flutter doctor --android-licenses
```

Ele exibirá várias licenças que precisamos aceitar, simplesmente pressione Y e enter para todas elas e pronto, pode rodar novamente um **flutter doctor** e verá que este item está OK.

Instalação dos Plugins

Abra o **Android Studio**, e na primeira tela, em **Configure**, selecione a opção **Plugins**. Na nova tela que se abre, pesquise por Flutter e realize a instalação do mesmo.

Após a instalação, ele pedirá para reiniciar a IDE, aceite e tudo pronto.

Abra o **Visual Studio Code**, navegue para as **extensões** no lado esquerdo da tela e na busca, procure por Flutter. Ele fará a instalação dos plugins do **Flutter e do Dart**.

Conectando seu Telefone

Você pode utilizar seu telefone para visualizar e testar suas aplicações, na verdade é recomendado que você **SEMPRE TESTE** em dispositivos físicos suas aplicações.

IMPORTANTE: Isto pode variar de acordo com cada fabricante, então caso este método não funcione para você, busque por "Habilitar modo desenvolvedor MODELO DO SEU TELEFONE".

Conecte seu telefone via USB em seu computador e nosso primeiro passo é habilitar o modo desenvolvedor do Android em seu telefone.

Isto **NORMALMENTE** é feito indo até a opção "Sobre este Telefone" nas Configurações e clicando várias vezes sobre o "Número da Versão (Build Number)".

Feito isto, uma nova opção no menu **Configurações** estará disponível, chamada **Opções do Desenvolvedor**. Clique sobre ela para visualizar os recursos.

Nesta nova tela, habilite a opção **Depuração via USB** e aceite ambas opções. Na segunda opção, caso seja um computador confiável, você pode marcar a opção "Sempre confiar neste computador".

Criando um AVD

AVD é a sigla para Android Virtual Device, que nada mais é do que uma máquina virtual rodando Android que simula um telefone para nós.

O modo mais fácil de criar um AVD é abrindo o **Android Studio**, clicando em **Configure** e selecionando a opção **AVD Manager**.

Se esta for sua primeira AVD, ele exibirá uma tela com um botão escrito "**Create Virtual Device**", clique neste botão para iniciar o assistente de criação.

Na tela seguinte, vamos selecionar o Hardware que queremos utilizar como máquina virtual. No caso, existem vários modelos de telefones e tablets e você pode criar quantos quiser.

Recomendamos que selecione a opção **Pixel 2** e clique em **Next** para prosseguir com a criação. O Pixel 2 possui uma tela intermediária que nos ajuda no desenvolvimento.

Na tela seguinte, vamos fazer o download da imagem da máquina virtual, e neste caso selecionamos a opção **Android 10 (Q)**, porém se você estiver lendo este artigo em um futuro, pode utilizar versões mais recentes se houver.

Na próxima tela basta selecionar a opção Finish para que o download seja iniciado e a máquina virtual seja criada. Neste momento, será necessário que sua máquina tenha suporte ao HAXM.

O HAXM é um recurso de aceleração em processadores Intel e precisa ser habilitado na BIOS em alguns casos. Você pode procurar como habilitar este recurso de acordo com sua placa mãe.

Nos computadores mais novos ele já vem habilitado, e alguns anti-vírus podem bloquear recursos de virtualização, impedindo a execução do seu emulador.

Com tudo pronto, basta executar o processo novamente (**Android Studio > Configure > AVD Manager**) para ver os emuladores instalados. Clique no **PLAY** para executar algum deles.

Testando seu primeiro App

Chegou a hora da verdade, vamos criar e testar um App para ver se tudo está funcionando corretamente.

Conecte seu telefone ou abra um Emulador e vamos até o Visual Studio Code. Nele, pressione **CTRL + SHIFT + P** e selecione a opção "Flutter new project"

CTRL+SHIFT+F e selecione a opção `Flutter new project`.

Caso esta opção não esteja aparecendo, comece a digitar o comando que ela aparecerá. Em seguida você será indagado a informar o nome do App.

Informe um nome sem espaços, caracteres especiais e tudo em minúsculo, e então selecione onde este projeto vai ficar. Particularmente, recomendamos cria-lo em um caminho seguro, como **C:\dev** no qual utilizamos aqui.

Aguarde a instalação dos pacotes, abra um novo terminal no Visual Studio Code (CTRL + `) e execute o seguinte comando.

```
flutter run
```

Caso esteja com o Emulador aberto e o telefone conectado, você receberá uma mensagem de erro, pois o Flutter não saberá em qual dispositivo deve rodar a aplicação.

Você pode remover o telefone ou fechar o emulador, mas pode também executar o comando **flutter devices** para listar os dispositivos conectados.

Em seguida você pode executar o comando **flutter run -d ID_DO_DEVICE** para executar o aplicativo no dispositivo desejado.

Pronto, se tudo deu certo você tem o Android e Flutter configurados no Windows e pronto para serem utilizados. Agora é só seguir nossa [Carreira Mobile Developer Flutter](#) e aprender a criar aplicativos incríveis.

Atualizando a versão

Recomendamos que sempre mantenha a versão do Flutter a mais atualizada possível, e para isto, basta abrir um terminal e executar o comando abaixo:

```
flutter upgrade
```

Deste modo ele fará o download dos novos pacotes e atualizará a versão atual do Flutter. Você pode executar um **flutter --version** para verificar a versão instalada.

Channels

Sempre que realizamos a instalação, por padrão o Flutter utiliza seu canal **Stable** (Estável), ou seja, apenas o que está testado e pronto para produção.

Caso queira testar algo que está em beta, você pode trocar o canal principal utilizando o comando:


```
flutter channel beta
```

Lembre-se que não é legal utilizar itens em beta em produção. Para voltar para o canal principal, basta executar o comando:

```
flutter channel master
```

Populares

Clean Code - Guia e exemplos

Como manter seu código limpo (Clean Code) seguindo algumas práticas sugeridas pelo Robert C. Mart...

Orientação a objetos: Introdução e guia de estudos

Orientação a objetos é um paradigma aplicado na programação que consiste na interação entre diver...

Perguntas para entrevista de C# e .NET

Está realizando ou participando de um processo seletivo que envolve C#? Então você precisa respon...

Dependency Injection

Dependency Injection (Injeção de dependência) ou DI é a técnica que implementa o padrão IoC que v...

ASP.NET Compressão

ASP.NET Compressão otimiza a performance das aplicações reduzindo tráfego de rede entre cliente e...

Conheça o autor



André Baltieri
Microsoft MVP

Me dedico ao desenvolvimento de software desde 2003, sendo minha maior especialidade o Desenvolvimento Web. Durante esta jornada pude trabalhar presencialmente aqui no Brasil e Estados Unidos, atender remotamente times da Índia, Inglaterra e Holanda, receber **8x Microsoft MVP** e realizar diversas consultorias em empresas e projetos de todos os tamanhos.

Comece de graça agora mesmo!

Temos mais de 16 cursos totalmente de graça e todos com certificado de conclusão.

Começar

Prefere algo mais Premium?

Conheça nossos planos

Premium anual

Compra única, parcelada em até
12x no cartão de crédito

12x
R\$ **64,90**
=R\$ 778,80

- ✓ Economize R\$ 180,00 (+ de 2 meses)!
- ✓ 1 ano de acesso
- ✓ Acesso à todo conteúdo
- ✓ Emissão de Certificado
- ✓ Tira Dúvidas Online
- ✓ 53 cursos disponíveis
- ✓ 4 carreiras disponíveis
- ✓ 161 temas de tecnologia
- ✓ Conteúdo novo todo mês

- ✓ [Encontros Premium](#)
- ✓ 7 dias para arrependimento

[Começar agora](#)

[Política de privacidade](#)

Premium mensal

Cobrado mensalmente via
cartão de crédito

R\$ **79,90**
por mês

- ✓ Acesso à todo conteúdo
- ✓ Emissão de Certificado
- ✓ 53 cursos disponíveis
- ✓ 4 carreiras disponíveis
- ✓ 161 temas de tecnologia
- ✓ Conteúdo novo todo mês
- ✓ Cobrado mensalmente

[Começar agora](#)

[Política de privacidade](#)

1,626

Aulas disponíveis

+500

horas de conteúdo

horas de conteúdo

+20.000

Alunos matriculados

+30.000

Certificados emitidos

Precisa de ajuda?

Dúvidas frequentes

Posso começar de graça?

Vou ter que pagar algo?

Por onde devo começar?

Os cursos ensinam tudo que preciso?

O que eu devo estudar?

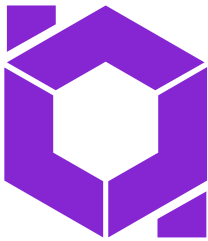
Estou pronto para estudar no balta.io?

Ainda tem dúvidas?

Assine nosso Newsletter

Receba em primeira mão todas as nossas novidades.

Cadastrar



Sobre

[Como funciona?](#)

[Seja Premium](#)

[Agenda](#)

[Blog](#)

[Todos os cursos](#)

Cursos

[Frontend](#)

[Backend](#)

[Mobile](#)

[Fullstack](#)

[Todos](#)

Suporte

[Termos de uso](#)

[Privacidade](#)

[Cancelamento](#)

[Central de ajuda](#)

Redes Sociais

[Telegram](#)

[Facebook](#)

[Instagram](#)

[YouTube](#)