



Stack tecnológico: o que é, exemplos e como construir seu stack

por **Cairo Noleto** · 12/04/2022

Um stack tecnológico é, resumidamente, uma pilha de tecnologias. **Stack é um** termo comum no dia a dia do desenvolvimento que vale a pena conhecer a fundo, afinal front end, back end e full stack são algumas expressões bastante populares. É a composição do stack tecnológico que vai determinar todas as dimensões do software que será desenvolvido.

Apesar de ser um termo muito popular, ele designa uma organização um tanto complexa. Pode ser considerado um conjunto de tecnologias mas que não são escolhidas aleatoriamente. Logo, um stack tecnológico tem uma estrutura cheia de elementos que pode assustar quem tenta entender e até construir o seu.

Dominar o tema vai auxiliar você a construir produtos mais eficazes, seja qual for a realidade da sua equipe. Veja a seguir todos os tópicos que vamos abordar para que você se familiarize e possa começar a aplicar seus conhecimentos sobre stack tecnológico.

- [O que é Stack tecnológico e como surgiu esse termo?](#)
- [Qual a importância do Stack tecnológico?](#)
- [Entenda os Stacks tecnológicos de desenvolvimento!](#)
- [Como é a estrutura básica de um Stack tecnológico?](#)
- [Exemplo de Stack tecnológico: entenda na prática!](#)
- [Quais as diferenças entre o Stack tecnológico de apps mobile e Web?](#)
- [Como escolher o Stack tecnológico correto para mim? Pontos a considerar!](#)



O que é Stack tecnológico e como surgiu esse termo?

Na tradução mais literal, um stack significa uma pilha. Nesse caso, um stack tecnológico seria **uma pilha de tecnologias**. De modo mais prático, podemos entender um stack tecnológico como um conjunto ou uma infraestrutura de tecnologias.

É importante frisar que não se trata apenas de um amontoado, mas uma organização lógica e funcional de ferramentas tendo em vista um objetivo, a aplicação em questão. O stack tecnológico são todas as tecnologias, incluindo frameworks, bibliotecas, linguagens e outras ferramentas usadas para determinado projeto.

O termo é bastante popular e usado no dia-a-dia de desenvolvimento, seja web ou mobile. Isso porque essas duas palavrinhas são capazes de traduzir tudo que engloba a construção e execução de uma aplicação.

Qual a importância do Stack tecnológico?

Com todo o avanço tecnológico que temos hoje em dia, existe um verdadeiro universo de possibilidades de ferramentas a se usar, seja no desenvolvimento web ou mobile.

Basicamente, **é a escolha do stack tecnológico quem dita o tipo de aplicação que será possível desenvolver**. Tudo depende então do que se pretende enquanto projeto. Cada stack tem suas características e ferramentas, sendo fundamental então uma boa noção de todas as possibilidades e limitações, para se fazer a escolha certa.

O stack tecnológico vai influenciar **o tipo de produto que você será capaz de produzir, a eficiência e até mesmo a equipe que estará contigo nesse projeto**. A construção de um stack adequado é também uma questão de planejamento e maturidade na governança.



tipo de profissional que será necessário.

O stack tecnológico de desenvolvimento se divide em dois principais eixos, o front end e o back end. Ainda há uma terceira possibilidade de stack, que engloba ambas as anteriores, o full stack.

Entenda os Stacks tecnológicos de desenvolvimento!

Front end

Resumidamente, o front end é a tecnologia que **roda no navegador da pessoa usuária**. Da tradução mais literal, front significa “frente”, ou seja, exatamente o que acontece na frente, o que o usuário ou usuária tem contato direto. Uma boa alegoria é a de um iceberg, onde a pontinha seria esse front end, ou seja, a pessoa usuária é capaz de “vê-lo”.

Para sermos mais específicos, o front end é então **tudo que a pessoa usuária é capaz de visualizar e interagir**, em contato com a interface. Inclui, obviamente, não só os elementos visuais em si, mas também suas funcionalidades. Botões, gráficos e caixas de entrada de texto são alguns exemplos de elementos que são objeto de interação, e que estão contidos no front end. O [front-end é, na verdade, tudo aquilo que acontece no cliente](#).

O trabalho de uma pessoa desenvolvedora front end é, basicamente, [tratar e tornar apresentáveis as informações contidas no back end](#). Quem se especializa nessa área costuma ter, então, familiaridade com a parte mais gráfica dos projetos. Alguns exemplos de tecnologias de front end são Javascript, HTML e CSS.

Back end

O Back end, por sua vez, diz respeito a tecnologias menos intuitivas (ou visuais). Por exemplo, o algoritmo que gere certa aplicação está determinado no back end dessa aplicação. O termo back, do inglês, quer dizer “trás” ou “fundo”. Ou seja, é aquilo que não é visível, que roda “atrás”, isto é, que é interno ao funcionamento da aplicação.



no cliente, o back-end e **aquilo que acontece no servidor**.

O back end é composto por servidor, banco de dados, linguagens de programação e um framework. [Python](#), Java, [Ruby](#), C# e PHP são alguns exemplos de linguagens que podem ser usadas no back end.

Full stack

Por último, o full stack engloba ambas áreas. Do inglês, full quer dizer cheio, ou completo. Ou seja, é um stack que contempla **tanto o front end, quanto o back end**. A pessoa desenvolvedora full stack é, logo, aquela que tem domínio das tecnologias tanto do lado do cliente, quanto do servidor.

Como é a estrutura básica de um Stack tecnológico?

De modo geral, um stack é uma combinação de ferramentas. A estrutura básica de um stack contém diversos tipos de tecnologias. Vejamos com atenção a seguir cada uma delas:

1. Framework de front end

O [framework](#) que será usado para o desenvolvimento do front end. Os frameworks são, basicamente, conjuntos de bibliotecas muito úteis no desenvolvimento.

Algumas possibilidades de framework para front end são o React, o JQuery, o AngularJS e outros.

2. Framework de back end

Existem algumas opções de frameworks de back end no mercado, Laravel, Django e Ruby são alguns deles.

3. Linguagens de programação e sistemas operacionais



4. **Servidores e balanceadores de carga**

Envolve servidores, redes de distribuição e serviços de cache.

5. **Armazenamento e consulta de dados**

Serve, basicamente, para que você possa armazenar dados no seu aplicativo. [Bancos de dados](#) podem ser relacionais ou não-relacionais.

6. **Ferramentas de monitoramento e performance**

São ferramentas para coleta e análise dos dados de desempenho técnico do software. Seu trabalho é rastrear a integridade dos sistemas internos.

7. **Serviços de API**

[API's](#) (Application Programming Interface) são interfaces que possibilitam o acesso a informações entre programas e aparelhos, seja com fins de integração de aplicativos, leitura e modificação de dados ou integração de hardware. Algumas plataformas de API são o Segment, o Zapier, o Google Apigee, o Tealium e o Mulesoft.

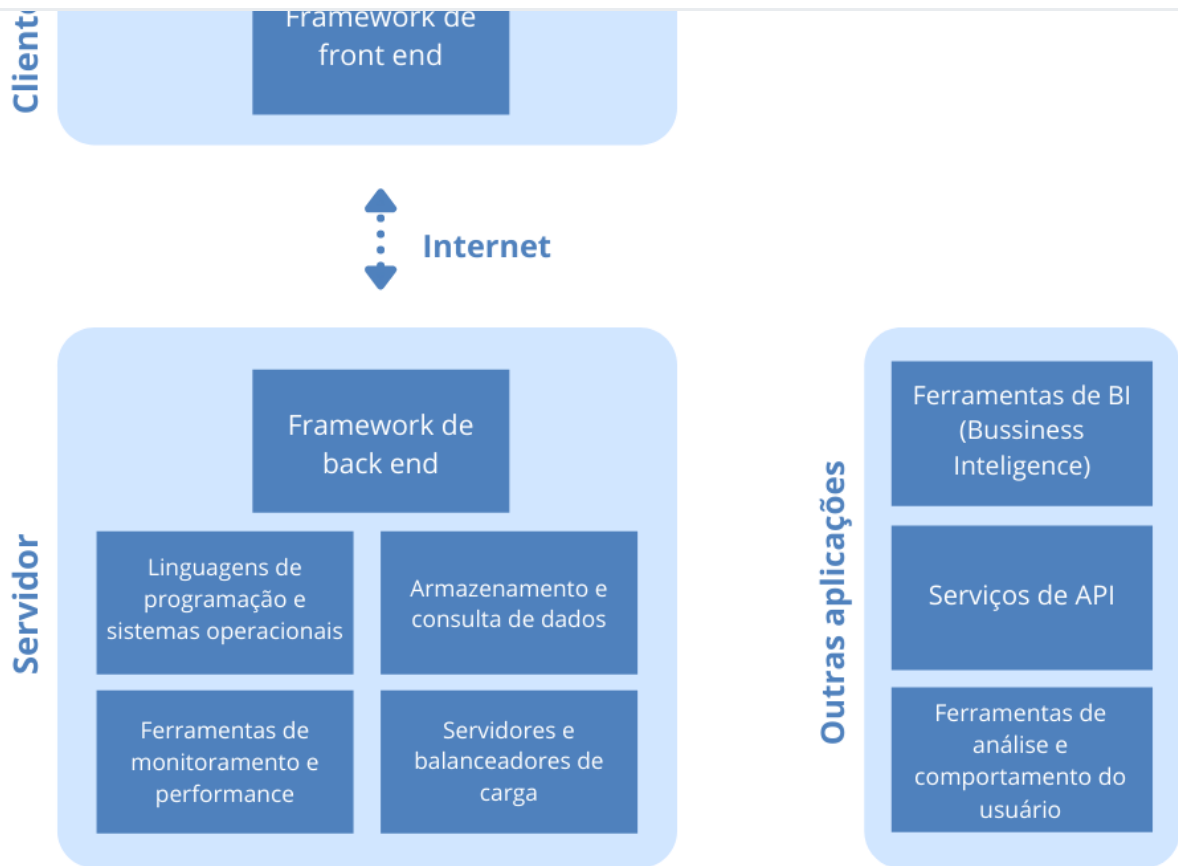
8. **Ferramentas de BI (Business Intelligence)**

São ferramentas voltadas ao desempenho da empresa e do mercado. Como exemplo de ferramentas com esse fim temos o Microsoft Power BI ou o Tableau, entre outros.

9. **Ferramentas de análise e comportamento do usuário**

São ferramentas voltadas para a análise do comportamento da pessoa usuária, tendo em vista a jornada do cliente.

A estrutura básica de um stack tecnológico se dá dessa forma:



No lado do cliente está o framework de front end e, no do servidor, outras cinco estruturas básicas: o framework de back end, as linguagens ou sistemas operacionais, as ferramentas de monitoramento e performance, o armazenamento e consulta de dados e os servidores e balanceadores de carga. Ainda, é possível que se utilize ferramentas de Business Intelligence, serviços de API e ferramentas de análise de comportamento do usuário.

Exemplo de Stack tecnológico: entenda na prática!

Agora que já vimos a estrutura básica de um stack tecnológico, podemos finalmente conhecê-lo de fato. Para compreendermos na prática como se estrutura um stack, usaremos como exemplo o stack do Airbnb. O Airbnb é uma plataforma de hospedagem que utiliza armazenamento em nuvem. A seguir, veremos um modelo simplificado de seu stack tecnológico.



Front end

Javascript

React

Back end

Java

MySQL

Redis

Nginx

- Javascript



- Nginx

O servidor.

- React

O framework Javascript.

- Java

A linguagem de programação do back end.

- MySQL

O banco de dados.

- Redis

Ferramenta para armazenamento de estrutura de dados.

- Rails

É o framework utilizado.

Você pode acessar e conhecer na íntegra o stack tecnológico do Airbnb e de outras aplicações gratuitamente na plataforma [stackshare](#).

Caso você esteja buscando se aprofundar enquanto [pessoa desenvolvedora](#) ou até buscando sua colocação no mercado de trabalho, vale a pena pesquisar as ferramentas mais usadas e, principalmente, as utilizadas pelas empresas que você mais almeja colaborar.

Quais as diferenças entre o Stack tecnológico de apps mobile e Web?



desenvolvimento web, enquanto, por sua vez, o mobile, ferramentas adequadas a ele.

A principal diferença é que, para se usar um aplicativo móvel, a pessoa usuária deve, necessariamente, baixá-lo em seu dispositivo. Isso não ocorre com aplicativos de desktop. **O uso de aplicativos web não depende de download, já que utilizam os recursos do navegador.**

Em comparação com um aplicativo mobile, um web demanda então menos versões, considerando que não é necessária uma construção específica para cada [sistema operacional](#). Porém, é importante também ter em mente que no caso de um stack web, a aplicação deve rodar em todos os navegadores! É importante lembrar que existem muitos navegadores no mercado.

Já no mobile, será necessário que a pessoa ou equipe desenvolvedora construa o software tendo em vista uma plataforma móvel específica. Isso porque a pessoa usuária vai ter acesso ao aplicativo a partir de determinada plataforma móvel, a depender do sistema operacional que usa. **No [desenvolvimento mobile](#), é preciso que se desenvolva uma versão para cada sistema operacional.**

Os sistemas operacionais mais comuns nos dispositivos móveis das pessoas brasileiras são o IOS e o Android. No desenvolvimento, precisam ser adicionadas ao stack tecnologias orientadas à construção para essas plataformas. Por exemplo, o uso do Swift para o IOS e do Kotlin para a integração com o Android.

No front end, o grupo mais básico de tecnologias para desenvolvimento web é HTML, CSS e Javascript. No desenvolvimento de apps mobile, por outro lado, temos uma variedade maior de possibilidades de tecnologias.

Como escolher o Stack tecnológico correto para mim? Pontos a considerar!

Parece simples, mas a escolha certa do stack tecnológico vai determinar muita coisa do seu projeto. É importante lembrar que a tecnologia, atualmente, é a ferramenta chave de qualquer processo em uma empresa. Para a escolha do stack tecnológico ideal, alguns pontos devem ser considerados.



necessárias no stack na muitos fatores a se considerar em relação ao funcionamento de um dispositivo móvel e como a pessoa usuária interage com ele. Além do tipo de projeto, é essencial ainda considerar o **tamanho e complexidade** do projeto.

Um outro ponto a se considerar é a **escalabilidade** do projeto. Isto é, deve-se considerar a capacidade de expansão do software para atendimento de novas pessoas usuárias. A escalabilidade pode ser de dois tipos, horizontal ou vertical. A **segurança** também é um fator importante. A depender do projeto, a segurança assume um lugar central no desenvolvimento. Logo, deve-se observar o que se pretende construir e, principalmente, o quão sensíveis são as informações com que se lidará.

Orçamento também é um fator importante. Os custos de um stack variam a depender do valor da hospedagem (hosting), licenças necessárias, remuneração da equipe ou valores para manutenção.

A própria **manutenção** também é, por si só, um ponto fundamental a se considerar na escolha do stack, já que é uma necessidade contínua, mesmo após o lançamento do produto. A manutenção vai então refletir tanto nos custos diretos, quanto no grau de dificuldade do trabalho, a depender do stack escolhido.

Além disso, é preciso considerar qual será o **prazo** para a entrega, o tempo necessário e disponível para que o produto vá para o mercado. Por último, um outro fator é o próprio grau de **experiência da equipe** e a curva de aprendizagem para se atingir determinados objetivos.

Como vimos, para escolher um stack tecnológico deve-se analisar quais e como estão dispostos os recursos disponíveis. Acima de tudo, é preciso ser realista nessa avaliação. Só assim poderá se mitigar possíveis transtornos e entregar um software com a melhor qualidade possível.

Escolher o Stack tecnológico errado pode matar meu projeto?

A escolha do stack determina exatamente todos os aspectos do que você será capaz de construir enquanto projeto. Por isso, **é essencial tomar uma decisão**



Tenha em mente que o stack é exatamente o conjunto de ferramentas que você terá disponível. Além disso, lembre-se que o ideal é que a decisão acerca do stack deva ocorrer bem no início do projeto, já que uma mudança de stack já no decorrer tende a acarretar em prejuízo, de tempo e trabalho, imensos.

Faça uma avaliação considerando todos os fatores, da complexidade do projeto ao orçamento. Com todos esses pontos na balança, as chances de se optar pelo stack errado serão menores.

Como construir seu Stack tecnológico? O passo a passo!

Passo 1 – Analise dados

Esse passo permite que você reconheça os principais processos e ineficiências.

Passo 2 – Tenha sempre em vista o objetivo da aplicação

Deixe que o objetivo da aplicação determine a linguagem a ser utilizada. Apesar da sua experiência e da equipe desenvolvedora serem fatores muito importantes, o projeto em questão deve assumir centralidade nas decisões acerca do stack.

Passo 3 – Planeje-se para o futuro

Nunca se esqueça que o planejamento deve ser a longo prazo. Nesse tema, você deve buscar, acima de tudo, flexibilidade e adaptabilidade. Alguns critérios a serem utilizados nessa etapa são: escalabilidade, integração, customização e viabilidade.

Passo 4 – Comece a decidir pelas ferramentas

Já sabendo todos os pormenores acerca dos recursos que você dispõe atualmente e tendo em vista o que almeja construir em termos de software, você já pode decidir as ferramentas. Na hora de escolher, pense em todos os pontos a se



Passo 5 – Confie na comunidade

O mundo da tecnologia tem um grande diferencial: sua comunidade. [Código aberto, ou open source](#), são códigos sem restrições de acesso e uso. Códigos abertos fazem muita diferença para quem está começando a construir seu stack. Além disso, sempre vale a pena conversar com outros colegas sobre ferramentas e avaliações. Você também pode buscar por versões de teste e demonstrações.

Passo 6 – Lista final

Agora que você já considerou todos os pontos e ordenou seu processo de escolha em um passo a passo detalhado, deve ter uma noção muito mais aprofundada de que ferramentas usar em seu stack. Agora resta avançar.

Como você viu, um stack tecnológico é fundamental para a execução de um bom projeto resultando em boas aplicações, isso porque busca e permite a construção do **melhor software possível**, dadas as ferramentas e condições que você dispõe. Um bom stack tecnológico reflete organização e maturidade da equipe.

Agora que você está familiarizado com o que é e como se constrói na prática um stack tecnológico, [conheça mais sobre a pessoa profissional desenvolvedora full stack e como se tornar uma](#). Veja também [aqui](#) um glossário completo de termos de tecnologia.

Cairo Noleto

Nascido na terra distante e quente de Teresina - PI em 1986, é quase Bacharel em Sistemas de Informação e trabalha com Desenvolvimento de Software desde 2007. Começou sua carreira na linguagem PHP e depois, em 2009, mudou para Ruby, na qual teve a felicidade de trabalhar no projeto de tradução da documentação para português (do Brasil) do framework Ruby on Rails. Já trabalhou em empresas como Jus Navigandi, e Enjoei, marketplace. Além de ter lidado com as linguagens Ruby e GoLang (2013-2016), e por fim impactou mais de 2 milhões de estudantes em todo o Brasil no AppProva, com as linguagens Ruby, PHP, JavaScript, GoLang e Elixir. Atualmente na Trybe é responsável pelo time de tecnologia de Experiência de Estudantes.



Meta testa tentativas de monetização na Horizon Worlds

Próximo artigo — >

Pesquisadores descobrem novo malware após ataque cibernético à Ucrânia

VOCÊ TAMBÉM PODE GOSTAR

T — Tecnologia

Tecnologias do futuro: Supervisão, Internet 6g, Carros autônomos e muito mais!

por **Cairo Noleto** · 27/03/2020



Séries sobre tecnologia: 8 séries imperdivéis para os amantes de Tecnologia!

por **Cairo Noleto** · 31/03/2020

T — Tecnologia



5 Filmes sobre tecnologia imperdíveis, para quem se interessa pela área!

por **Cairo Noleto** · 05/04/2020

[Currículo](#)[Dúvidas](#)[Trabalhe Conosco](#)[Gerador de CPF](#)[Pague só quando trabalhar](#)[Guia HTML](#)[Guia Javascript](#)[Guia Soft skills](#)[Carreira](#)[Tecnologia](#)[Desenvolvimento Web](#)[Linguagens de Programação](#)[Framework de Programação](#)[TXN](#)[Ferramentas](#)[Desabilitar cookies](#)[Política de Privacidade](#)