

# **Deno: O Guia Completo do Zero ao Domínio**

Aprenda DenoJS na prática.

Gabriel Mendes

# Deno: O Guia Completo do Zero ao Domínio

Aprenda DenoJS na prática.

Gabriel Mendes

Esse livro está à venda em <http://leanpub.com/deno-brasil>

Essa versão foi publicada em 2020-08-24



Esse é um livro [Leanpub](#). A Leanpub dá poderes aos autores e editores a partir do processo de Publicação Lean. [Publicação Lean](#) é a ação de publicar um ebook em desenvolvimento com ferramentas leves e muitas iterações para conseguir feedbacks dos leitores, pivotar até que você tenha o livro ideal e então conseguir tração.

© 2020 Gabriel Mendes

# Conteúdo

Agradecimentos . . . . .	1
Introdução . . . . .	2
<b>Fundamentos Deno</b> . . . . .	4
Por que Deno? . . . . .	4
Deno e a V8 Engine . . . . .	7
<b>Instalação do Deno</b> . . . . .	11
Configurando Variável de Ambiente . . . . .	11
Configurando Nosso Ambiente de DEV . . . . .	11
<b>Nosso Primeiro Aplicativo Deno</b> . . . . .	12
Deno por de baixo dos panos . . . . .	12
Métricas Deno . . . . .	12
<b>Deno vs Node</b> . . . . .	13
Mudanças do Deno . . . . .	13
Segurança . . . . .	13
Permissões Deno . . . . .	13
<b>Módulos e Ferramentas do Deno</b> . . . . .	14
Como os módulos funcionam no Deno . . . . .	14
Biblioteca Padrão . . . . .	14
Códigos de terceiro . . . . .	14
Cache do Deno . . . . .	15
CDN . . . . .	15

## CONTEÚDO

Controlando versões do módulo . . . . .	15
Arquivo de dependências do Deno . . . . .	15

# **Agradecimentos**

Olá, estou muito feliz de está iniciando este livro, é o primeiro de muitos que estão por vir e me sinto melhor ainda pelo fato de você está acompanhando como material de estudo.

Quando comecei este livro em 2020, a minha ideia era trazer para o mundo Deno um conteúdo em português que não tínhamos ainda, de forma simples, clara e barata.

Espero que o livro tenha um diferencial na sua carreira e abra portas para novas oportunidades!

# Introdução

Antes de iniciarmos com a codificação, iremos entender como é o processo para depois programar de fato, porque acredito que este é o caminho certo de aprender.

Logo abaixo, descrevo os passos que iremos seguir ao longo do livro.

Primeiro: Vamos iniciar pelos conceitos básicos e fundamentais do Deno, aprenderemos como ele foi construído, sua arquitetura e com isso você irá entender como funciona o runtime do Javascript.

Segundo: Iremos fazer uma comparação entre o Deno e o Node e quando tivermos realmente entendido o Deno, você irá entender algumas deficiências do Node que o Deno tenta melhorá-la.

Terceiro: Vamos aprender sobre os módulos e ferramentas, garantindo com que iremos utilizar os principais pacotes para construir um monte de aplicações.

Quarto: Iremos mergulhar através de algumas das bibliotecas básicas do Deno, veremos que podemos fazer um monte de coisas legais.

Quinto: Em seguida vamos ter uma rápida conversa sobre backend, APIS e aprender sobre como são construídos estes sistemas.

Sexto: Vamos construir uma divertida aplicação utilizando API do pokémon, com isto, iremos criar nossa própria pokédex.

Por fim o Sétimo: Iremos construir nosso segundo projeto utilizando API do Star Wars.

O livro utilizará Typescript, mas não tenha medo se você não está acostumado com ele porque no fim do livro tenho planos de incluir uma seção de Typescript nas seções de bônus.

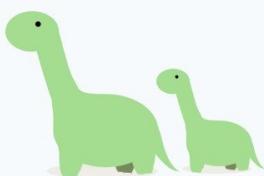
No fim deste livro você vai ter uma ótima noção de Deno porque passaremos por tópicos como segurança, deploy, desempenho e com

isso, você se sentirá confortável para fazer seus próprios projetos para por no github.

Prometo a você que você irá aprender bastante e se divertir fazendo seus próprios projetos, sairemos do Zero ao Domínio.

Agora bora deixar de enrolação e **VAMOS CODAR!**

# Fundamentos Deno



## Por que Deno?

Sabemos que o Deno é um bebê ainda comparado ao Node e também que ele é um runtime Javascript. Então por que usar Deno? O que deveríamos nos preocupar com ele? Se [acessarmos o site dele<sup>1</sup>](#), veremos um conjunto de palavras novas.

Deno is a simple, modern and secure runtime for JavaScript and TypeScript that uses V8 and is built in Rust.

Lá informa que o Deno é um simples, moderno e seguro runtime para Javascript e Typescript que usa a V8. Agora se você tiver um pouco de experiência como engenheiro sênior você pode saber o que elas significam, porém, se você não sabe, não se preocupe, vamos entrar profundamente nestes tópicos e entender por debaixo dos panos o Deno e de fato o que ele é, o que ele faz, a diferença em relação ao Node e o porquê devemos nos manter de olho nele.

Se você dar uma olhada nas palavras seguru runtime para Javascript, isso parece ser um bom sinal.

---

<sup>1</sup><https://deno.land/>

Você pegou este livro porque provavelmente já sabe que o Node é um Javascript runtime, como o Deno, então porque devemos ter outro Javascript runtime? Outro lugar para rodarmos nosso código Javascript? Isso é uma boa pergunta, por que precisamos do Deno?

Já temos muitas linguagens de programação, quero dizer, temos o Python, C++, PHP e também temos o NodeJS, todas são maneiras de escrever programas, certo? E, no fim de tudo, programas são apenas instruções que podemos escrever, são tarefas que o nosso computador pode fazer e todas essas linguagens são maneiras diferentes de fazer comunicação com nossa máquina ou computador, assim como temos linguagens humanas como Português ou Inglês para nos comunicarmos, também possuímos linguagens para comunicar com nosso computador, mas no fim de tudo, todas fazem ou significam a mesma coisa.

Então por que precisamos de mais uma? Temos um cara para explicar isso e seu nome é Rayan Dahl, ele é uma pessoa interessante, pois, além de criar o Deno, ele também foi o criador do NodeJS.



Em 2009, que agora parece ser um longo tempo atrás, Rayan Dahl criou o NodeJS e abriu o mundo para o Javascript, mas o que isso significa? Inicialmente Javascript poderia ser entendido apenas pelos navegadores, se você estivesse utilizando Chrome, Firefox ou

Safari, estes navegadores eram permitidos rodar Javascript através do lado do cliente por aplicações ou websites. Então, se abrir as ferramentas de desenvolvimento, ir para o console e escrever Javascript, o navegador consegue interpretar.



A screenshot of a browser's developer tools console. The tabs at the top are Elements, Console, Sources, Network, Performance, and Memory. The Console tab is selected. Below the tabs, there are buttons for 'top' and 'Filter', and dropdown menus for 'Default levels'. The main area shows a single line of code: > `console.log("Olá estou rodando Javascript através do navegador")`. To the right of the code, the output is shown: `Olá estou rodando Javascript através do navegador` with a link [entry-main-modern-ve...297c16f95402ea.js:1](#).

Podíamos escrever Javascript apenas nos navegadores, agora se você queria algo fora do navegador, deveríamos usar Python, C++, ou alguma outra linguagem para construir aplicativos desktops. Porém, Rayan foi além, pensou que poderia ser interessante usarmos Javascript fora do navegador e talvez construir servidores com Javascript, e com esta ideia ele criou o NodeJS.

Agora a pergunta que não quer calar, se já temos o NodeJS, por que ele criou o Deno?

Este tópico vai ser explorado através do livro, porque a ordem do verdadeiro entendimento do Deno é primeiro precisamos entender as diferenças sobre o Node, e também a razão do porque ele decidiu criar algo que essencialmente faz a mesma coisa que o Node, porém, mais moderno.

Para você ficar excitado porque certamente eu estou sobre o Deno, temos algumas coisas mais profundas sobre ele no curso como Rust(Mozilla), TypeScript(Microsoft) e V8 Engine(Google), porém, não precisa se desesperar se é algo novo para você, vamos realmente aprender como o Deno funciona internamente.

Como disse anteriormente, temos alguns nomes novos e logo abaixo irei explicar resumidamente o que é cada um deles.

Rust - É uma linguagem de programação multiparadigma que teve grandes investimentos e também foi desenvolvida pelo Mozilla.

TypeScript - É um superconjunto de Javascript desenvolvido pela Microsoft que adiciona tipagem e alguns outros recursos a lingua-

gem. O melhor benefício dela é a detecção de erros sem precisar rodar a aplicação em tempo de execução, porém, também possui outras vantagens, como uma manutenção mais fácil a longo prazo. O Typescript não é uma linguagem de programação, e sim um superset, logo, o Typescript depende do Javascript, e se torna uma ferramenta essencial ao longo do tempo que o projeto vai crescendo.

V8 Engine - É uma engine que basicamente consegue ler Javascript. A V8 Engine é mantida pelo Google, com isso, temos total segurança em utilizar ela em nossos navegadores e runtimes, porque possui uma equipe dedicada profissionalmente para termos melhorias e correções de erros ao tempo todo. Não seria nada interessante construir uma do zero, tendo perda de recursos e precisando reinventar a roda.

Agora vamos aprender como Deno trabalha com a V8 Engine.

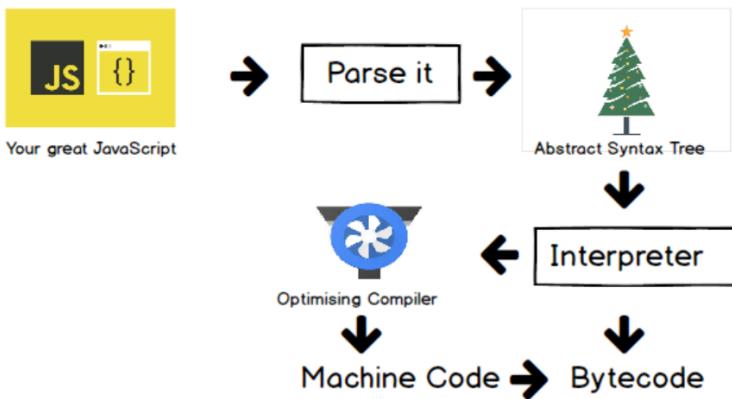
## Deno e a V8 Engine

Então sabemos que Deno é um Javascript e um Typescript runtime. Se você for para o site do NodeJS, lá diz que Node é um Javascript runtime construído através da V8 Engine.



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Então ambos foram construídos pela V8 Engine. Isso soa bem, mas o que é isto? A V8 Engine pega o código javascript escrito e transforma em instruções de máquina.



Por exemplo, o navegador Chrome utiliza a V8 Engine e quando ele recebe o código Javascript, acontece algo do tipo como o seguinte diálogo:

**Você envia o código javascript**

V8 Engine responde: Oh! Eu entendi isso, é Javascript, é o que o programador tentou nos dizer.

**Então o código é retornado ao navegador em instrução de máquina**

O código de máquina nada mais é que um conjunto de zeros e uns que dão instruções ao computador. E, no fim de tudo, tudo vira isto, certo? Escrevemos em diferentes linguagens e quando é compilado tudo vira zero e uns porque os computadores só conseguem entender assim.

As pessoas criam linguagens de programação porque é uma maneira mais fácil de escrever código sem precisar escrever em zeros e uns e eles constroem essas engines para serem capazes de traduzir de uma linguagem de programação lida por humanos para algo como isso para que o computador possa executar.

Então, por que precisamos utilizar a V8 Engine? Bem, precisamos da V8 Engine porque no fim de tudo, Deno é literalmente uma espécie de interface de usuário ou shell que consegue acessar serviços, e um

destes serviços engloba a V8 Engine.

Basicamente podemos escrever códigos Javascripts e a V8 Engine pega o código e devolve instruções para a máquina.

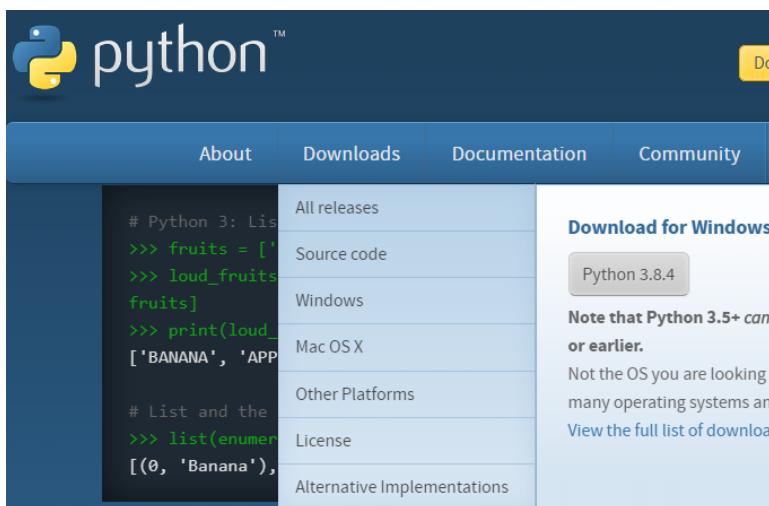
A V8 Engine foi construída pelo Google, e é realmente muito boa, performática, rápida e constantemente atualizada.

Só pelo fato de termos o Node ou Deno em nosso computador, podemos rodar Javascript, além disso, eles também podem oferecer ferramentas extras que a V8 Engine não oferece para gente, por exemplo, podemos usar Typescript no Deno porque ele compila nosso Typescript, converte para Javascript e então passa para a engine V8.

Este exemplo é claramente uma versão simplificada do que Deno pode fazer por nós, porém, dá para ter um entendimento melhor de como as coisas funcionam.

Agora se escrever códigos Javascripts em um bloco de notas e salvar no meu desktop, ele vai rodar sem o Node ou Deno instalado? E a resposta é não, porque meu computador não vai entender Javascript, ele vai basicamente dizer “Não entendo bulhufas do que está escrito aqui Gabriel”, grave isso **precisamos de um runtime para escrever códigos Javascripts**.

Além de Javascript, seu computador também não entende outras linguagens como Python, se você quer escrever em Python que é uma outra linguagem de programação, você precisa fazer download dele porque você precisa de um lugar para entender e ser capaz de rodar o seu código.



Se você for na [página de download do python](#)<sup>2</sup> e baixar a última versão dele, ele virá com um monte de coisas e uma delas a mais importante é o seu próprio interpretador, para ler e entender o seu código.

Acredito que já temos conhecimento suficiente para seguir em frente, vamos seguir fazendo o download do Deno e escrever nossa primeira aplicação.

---

<sup>2</sup><https://www.python.org/>

# **Instalação do Deno**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

## **Configurando Variável de Ambiente**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

## **Configurando Nosso Ambiente de DEV**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

# **Nosso Primeiro Aplicativo Deno**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

## **Deno por de baixo dos panos**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

## **Métricas Deno**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

# **Deno vs Node**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

## **Mudanças do Deno**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

## **Segurança**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

## **Permissões Deno**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

# **Módulos e Ferramentas do Deno**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

## **Como os módulos funcionam no Deno**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

## **Biblioteca Padrão**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

## **Códigos de terceiro**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

## **Cache do Deno**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

## **CDN**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

## **Controlando versões do módulo**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.

## **Arquivo de dependências do Deno**

This content is not available in the sample book. The book can be purchased on Leanpub at <http://leanpub.com/deno-brasil>.