



Desenvolvimento Backend

Aula 01 - Entendendo o Node.js



Material Did tico do Instituto Metr pole Digital - IMD

Termo de uso

Os materiais did ticos aqui disponibilizados est o licenciados atrav s de Creative Commons **Atribui  o-SemDeriva  es-SemDerivados CC BY-NC-ND**. Voc  possui a permiss o para realizar o download e compartilhar, desde que atribua os cr ditos do autor. N o poder  alter -los e nem utiliza-los para fins comerciais.

Atribui  o-SemDeriva  es-SemDerivados

CC BY-NC-ND



<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Apresentação

Nesta primeira aula, você aprenderá um pouco sobre programação backend com Node.js. Conhecerá um pouco sobre essa tecnologia, sobre como instalar e executar programas.

Objetivos

- Entender do que se trata o termo Backend
- Conhecer o Node.js
- Aprender um pouco da história do Node.js
- Aprender como instalar o Node.js e a criar um primeiro programa nessa tecnologia

O que é backend?

Link do video da aula: <https://youtu.be/mlfFpS3q0kM>

O que significa *backend*? Normalmente, uma aplicação web tem duas grandes partes: *frontend* e *backend*.

Frontend é a parte da aplicação que executa no próprio navegador do usuário. Ou seja, código HTML, CSS e Javascript que manipula a página com objetivo de apresentá-la ao usuário para que ele possa interagir com o seu sistema.

No entanto, é muito comum ter uma outra parte que executa em servidores. Essa outra parte é chamada de *backend*. O *backend* tem esse nome porque executa por trás em servidores e não na máquina do usuário.

Essas duas partes estão a todo tempo se comunicando, trocando mensagens. O *backend* geralmente é a parte que manipula dados mais sensíveis da aplicação. É ela que interage normalmente com o banco de dados e mantém os dados salvos e seguros.

É dessa forma que uma aplicação web é normalmente dividida. Neste curso, iremos focar na parte do *backend*. Preparados? Vamos lá!

Um pouco do Node.js

Link do video da aula: <https://youtu.be/YB-IKjzhzXA>

Neste curso, nós utilizaremos a **tecnologia** [Node.js](#) para executar e desenvolver

código de *backend* com **Javascript**.

Antes do Node.js, o Javascript era uma linguagem utilizada apenas para manipular páginas nos navegadores web. Como o Node.js, foi possível aproveitar essa mesma linguagem para criar código de *backend*. Com isso, milhares de programadores de *frontend* que já estão habituados com a linguagem Javascript podem aproveitar esse conhecimento para criar todo o sistema. Essa é a grande inovação do Node.js.

Como dito, o Node.js não é uma linguagem de programação e sim uma tecnologia que utiliza o motor de execução Javascript do Google Chrome (V8) para execução de código Javascript fora do navegador. O motor de execução Javascript do Google Chrome é bastante eficiente e rápido, trazendo todo esse benefício à tecnologia.

Uma outra grande vantagem do Node.js é a simplicidade para atender a múltiplos usuários em um programa web. Diferentemente das soluções mais clássicas, o Node.js utiliza apenas uma única linha de execução (thread) para atendimento de múltiplas requisições. No entanto, mesmo assim, consegue atender a múltiplos usuários através de uma técnica chamada event loop. Essa técnica traz bastante simplicidade para os desenvolvedores com baixo custo computacional. Neste curso, estudaremos esses diferenciais em mais detalhes nas próximas aulas.

Atualmente, o Node.js conta com um grande ecossistema com milhões de bibliotecas disponíveis que você pode usar gratuitamente. Empolgado? Vamos lá!

Um pouco da História do Node.js

Link do video da aula: <https://youtu.be/hcd9dNk9mto>

Agora que você já conhece um pouco do Node.js, vamos entender sua história. O Node.js foi criado no fim de 2009 e é relativamente novo, é verdade. Apesar de relativamente recente, já pode ser considerado maduro, principalmente por se ancorar numa tecnologia ainda mais antiga: o Javascript, que foi criado em 1995 pela Netscape Navigator.

A Netscape na época era uma grande empresa que tinha o principal navegador: Netscape. Nesse momento, o Javascript foi criado para permitir maior interatividade nas páginas. A própria Netscape tentou utilizar o Javascript fora do navegador com a tecnologia chamada Netscape LiveWire, mas na época não vingou.

No entanto, hoje em dia o Javascript se tornou muito mais moderno e muito mais rápido e o Node.js se beneficiou desse cenário.

De lá para cá, houve uma forte competição para tornar o Javascript cada vez mais rápido, principalmente com a guerra dos browsers.

Nesse cenário, o Google contribuiu bastante criando uma *engine* muito poderosa de execução de Javascript. O Node.js utiliza tal engine aproveitando todo o trabalho realizado pelo Google. Pode-se dizer então que o Node.js foi criado no local certo e na hora certa.

Além disso, dada as limitações de paralelismo do Javascript, o Node.js trouxe novas abordagens para tratamento de vários usuários simultaneamente.

Essa é uma breve história do Node.js. Agora, vamos começar a colocar a mão na massa, aprender a instalar e criar nosso primeiro programa. Até lá!

Instalando o Node.js (windows)

Link do video da aula: https://youtu.be/JeVp6_d_i_o

Acompanhe o passo a passo de instalação e teste do Node.js no ambiente Windows, pela videoaula. Para fazer o download do instalador, acesse <https://nodejs.org>.

Qualquer dificuldade, poste sua dúvida nos nossos fóruns.

Executando o primeiro programa

Link do video da aula: <https://youtu.be/noo32vKfeVk>

Agora que você já instalou o Node.js, vamos construir um primeiro programa.

O programa mais básico em Node.js exibe uma mensagem no console. Acompanhe o código abaixo:

```
console.log('Hello World')
```

Salve o arquivo com o nome `index.js` e execute pelo *prompt* de comando, digitando:

```
$ node index.js
```

Agora, vamos incrementar um pouco mais o exemplo, realizando leitura de dados do teclado. Acompanhe a videoaula para maiores detalhes.

```
const readline = require('readline')

const leitor = readline.createInterface({
```

```
    input: process.stdin,  
    output: process.stdout  
  });  
  
leitor.question('Digite um número de 1 a 10: ', function (resposta)  
{  
    console.log(resposta)  
    leitor.close()  
})
```

A documentação completa da biblioteca readline pode ser encontrada no seguinte link: <https://nodejs.org/api/readline.html>.

Resumo

Nesta aula, você viu o que significa o termo *backend* e começou a estudar a tecnologia Node.js. Viu um pouco de sua história, como instalar e executar programas escritos em Javascript fora do navegador. Agora, você está pronto para avançar e criar programas em Node.js cada vez mais completos.