

NODE.JS NO UBUNTU

MICROLEARNING

Getting started
Setup with NVM



01



AOBA!

Ricardo de Luna Galdino
Software Engineer



LINKEDIN:
[linkedin.com/in/ricardo-galdino](https://www.linkedin.com/in/ricardo-galdino)

GITHUB:
github.com/ricardogaldino

WHATSAPP GROUP:
[engsoft.org](https://www.whatsapp.com/group/engsoft.org)

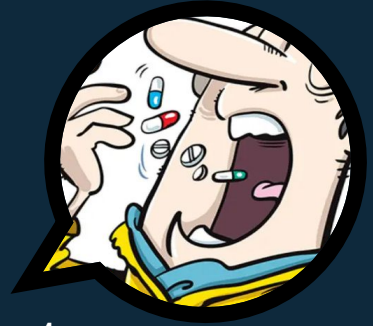


*"Sucesso é o acúmulo de pequenos esforços,
repetidos dia e noite."*

(Robert Collier)



Cultura ágil de aprendizado



Microlearning:

- É uma metodologia de ensino que subdivide um assunto em doses menores de conteúdo, com atividades rápidas, auxiliando na compreensão e retenção deste conteúdo.

Pílulas do Conhecimento:

- São pequenos conteúdos apresentados ao profissional para que consiga assimilar de forma mais focada e objetiva, melhorando a eficiência e potencializando os resultados obtidos.
- Microlearning é composto por diversas Pílulas do Conhecimento.

A decorative pattern of hexagons in various shades of blue and cyan on the left side of the slide. Some hexagons contain icons: a lightbulb, a thumbs up, a smartphone, a magnifying glass, and a gear. A network icon is also visible.

1

Glossário

Vamos rever alguns conceitos e termos...



NODE

- ◇ O Node.js é um ambiente de execução, multiplataforma, JavaScript, lançado em 2009 e criado para a programação no lado do servidor.
- ◇ Ele permite que os desenvolvedores criem funcionalidades de back-end escaláveis usando a linguagem de programação JavaScript sem a necessidade de um navegador.

nodejs.org

NVM

- ◇ O Node Version Manager (nvm), como o próprio nome já diz, é uma ferramenta que permite gerenciar diferentes versões do Node em um mesmo ambiente.

github.com/nvm-sh/nvm

NPM

- ◇ O Node Package Manager (npm) é o gerenciador de pacotes (bibliotecas) para o Node.js da Microsoft.

npmjs.com

A decorative pattern of hexagons in various shades of blue and cyan on the left side of the slide. Some hexagons contain icons: a lightbulb, a thumbs up, a network diagram, a smartphone, a magnifying glass, a gear, and a speech bubble.

2

Setup

Instalação do **NVM**, **Node.js** e **NPM** no Linux **Ubuntu**



Remoção de instalações anteriores!

Antes de instalarmos o “NVM” vamos remover qualquer outra instalação anterior do Node.js para evitarmos possíveis erros ou conflitos.

Abra o terminal e execute os comandos abaixo na sequência:

1. `$ sudo apt-get update && apt-get upgrade`
2. `$ sudo apt-get remove nodejs`
3. `$ sudo apt-get remove npm`
4. `$ sudo rm -rf ~/.nvm`
5. `$ sudo rm -rf ~/.npm`
6. `$ sudo rm -rf ~/.bower`





Instalação do NVM

(Node Version Manager)

Abra o terminal e execute os comandos abaixo na sequência:

1. **Instalar o pacote "build-essential":**

```
$ sudo apt install build-essential -y
```

2. **Criar o arquivo ".bashrc":**

- github.com/nvm-sh/nvm#troubleshooting-on-linux

```
$ source ~/.bashrc
```

3. **Instalar o "NVM":**

Obs: Visite o site do projeto para checar a versão mais atual:

- github.com/nvm-sh/nvm/releases
- github.com/nvm-sh/nvm#install--update-script

```
$ curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash
```






Comandos do NVM

(Node Version Manager)

Abra o terminal e execute os comandos abaixo na sequência:

1. **Verificar a versão do NVM:**
`$ nvm --version`
 2. **Verificar as versões do Node disponíveis para instalação:**
`$ nvm ls-remote node`
 3. **Instalar a versão mais recente do Node:**
`$ nvm install node`
 4. **Instalar uma versão específica do Node:**
`$ nvm install node v17.4.0`
 5. **Listar as versões instaladas:**
`$ nvm list node`
 6. **Alternar entre as versões instaladas:**
`$ nvm use node v17.4.0`
- 



Comandos do Node.js

Após instalado o Node.js, abra o terminal e execute os comandos abaixo na sequência:

1. **Verificar a versão corrente do Node.js:**

```
$ node -v
```

2. **Verificar o path da instalação do Node.js:**

```
$ which node
```

3. **Executar um arquivo JavaScript com Node.js:**

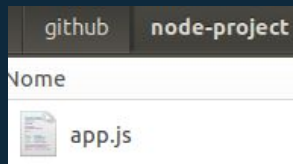
Crie um arquivo com nome e extensão **"app.js"** em uma pasta chamada "node-project", cole os comandos abaixo e salve o arquivo.

```
var msg = 'Hello World!';  
console.log(msg);
```

No terminal, vá até a pasta do arquivo e execute o comando:

```
$ node app.js
```

Você deverá ver a saída **"Hello World!"**





Comandos do NPM

(Node Package Manager)

Após instalado o Node.js, abra o terminal e execute os comandos abaixo na sequência:

1. **Verificar a versão corrente do NPM:**
`$ npm -v`
2. **Atualizar a versão do NPM:**
`$ npm install -g npm`
3. **Verificar o path da instalação do NPM:**
`$ which npm`
4. **Instalar um pacote NPM (ex: typescript):**
`$ npm install -g typescript`
5. **Desinstalar um pacote NPM (ex: typescript):**
`$ npm uninstall -g typescript`
6. **Listar pacotes NPM instalados:**
`$ npm list -g --depth=0`

Pesquise por pacotes
Node.js no site do
NPM:
[npmjs.com](https://www.npmjs.com)





package.json

(docs.npmjs.com/cli/v8/configuring-npm/package-json)

O "package.json" é um arquivo de configuração de um projeto Node.js, utilizado para definir dependências, scripts automatizados, dentre outras informações.

Após instalado o Node.js, abra o **terminal** e **execute os comandos** abaixo na sequência:

1. **Criar um arquivo "package.json" :**

```
$ npm init -y
```

2. **Salvar um pacote NPM como dependência (ex: typescript):**


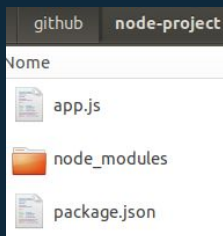
```
$ npm install -s typescript
```

ou

```
npm install --save typescript
```

3. **Instalar as dependências:**

```
$ npm install
```



```
{} package.json X
home > rluna > github > node-project > {} package.json > ...
1  {
2    "name": "node-project",
3    "version": "1.0.0",
4    "description": "",
5    "main": "app.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1"
8    },
9    "keywords": [],
10   "author": "",
11   "license": "ISC",
12   "dependencies": {
13     "typescript": "^4.5.5"
14   }
15 }
```



TO BE CONTINUED...



github.com/ricardogaldino/microlearning-node