



Instalação

Node e NPM (Contextualização)

[Linux \(Ubuntu/Debian\)](#)

[macOS](#)

[Windows](#)

Yarn 1

[Linux \(Ubuntu/Debian\)](#)

[macOS](#)

[Windows](#)

Expo

Visual Studio Code

[Extensões](#)

[Omni](#)

[Material Icon Theme](#)

[Configurações](#)

Node e NPM (Contextualização)

O primeiro passo para podermos utilizar a Omnistack (Node.js, ReactJS e React Native) é instalar o Node.js, que vem acompanhado do NPM. Para visualizar o site do Node.js e suas versões [clique aqui](#).

Como você já deve ter visto, na página principal do Node são apresentadas duas versões: **LTS** e **Current**. A primeira se refere à versão do Node que possui **Long Term Support (LTS)**, são as mais confiáveis e é a que recomendamos utilizar na NLW. Já a segunda se refere à versão do Node mais atual e experimental, o que não é recomendada para desenvolvimento ainda.

Escolhida a versão LTS do Node, precisamos decidir o método de instalação. É importante ressaltar que apesar de na tela inicial do Node.js eles recomendarem a forma de instalação direta (famosa janela que só clicamos no **Next**), iremos utilizar nesse guia os gerenciadores de pacote (exceto Linux).

Não só pelo fato de facilitar possíveis desinstalações e atualizações do Node, mas também por serem muito úteis para trabalhar com diversos outros pacotes.

Pronto dev, agora que já sabemos que iremos instalar a versão LTS do Node.js utilizando um gerenciador de pacote, bora para o passo-a-passo de cada sistema operacional.

Linux (Ubuntu/Debian)

Para o Linux iremos utilizar o **NodeSource**, basta seguir esses passos:

- Verifique se você possui o [curl](#) instalado rodando no terminal o comando:

```
curl --version
```

Bash ▾

Caso ele retorne a versão, pode pular para o próximo passo. Caso não, basta rodar o comando:

```
sudo apt install curl
```

Bash ▾

- Com o **curl** instalado, execute o comando de instalação da versão LTS mais recente disponível:
 - Ubuntu

```
curl -sL https://deb.nodesource.com/setup_lts.x | sudo -E bash -
sudo apt-get install -y nodejs
```

Bash ▾

- Debian (como root)

```
curl -sL https://deb.nodesource.com/setup_lts.x | bash -
apt-get install -y nodejs
```

JSON ▾

Feche o terminal e abra novamente para as alterações fazerem efeito.

- Por fim, execute os seguintes comandos no terminal:

```
node -v
npm -v
```

Bash ▾

Caso retorne as versões do Node e npm, sua instalação foi um sucesso.

macOS

Para o macOS iremos utilizar o gerenciador de pacotes [Homebrew](#), que é instalado usando Ruby, que já vem instalado por padrão, execute o seguinte comando no terminal:

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master,...
```

Bash ▾

Para verificar se ele foi instalado com sucesso execute:

```
brew --version
```

Bash ▾

Com o [Homebrew](#) instalado, basta executar o comando para instalar a versão 12 (LTS) mais recente:

```
brew install node@12
```

Bash ▾

Como instalamos uma versão do Node diferente da default do Homebrew (o padrão é a current, nesse caso v14), é preciso adicionar manualmente o `path` do Node na nossa variável ambiente. Adicione a seguinte linha ao final do arquivo `~/.bashrc` (ou do arquivo `~/.zshrc` caso você utilize o shell ZSH):

```
export PATH="/usr/local/opt/node@12/bin:$PATH"
```

Bash ▾

Por fim, reinicie o terminal e execute os seguintes comandos:

```
node -v
npm -v
```

Bash ▾

Caso retorne as versões do Node e npm, sua instalação foi um sucesso.

Windows

Para o Windows utilizaremos o gerenciador de pacotes [Chocolatey](#), porém antes dos passos de instalação vamos falar brevemente sobre qual shell você deve usar.

- **CMD**: também conhecido como **Command Prompt**, ele é um dos shells mais antigos da atualidade (foi construído para ser compatível com o **MS-DOS**) e, apesar da sua fama, hoje em dia tem sido cada vez menos utilizado.
- **Powershell**: novo shell apresentado pela Microsoft por volta de 2005, ele apresenta diversas melhorias em relação ao **CMD**, tornando-o popular atualmente e consequentemente a nossa escolha para a NLW#02.

Escolhido o shell, vamos começar a instalação:

- Busque no campo de busca do Windows por **Windows Powershell**, clique com o botão direito em cima do programa e escolha a opção **Executar como administrador**.
- O Powershell trabalha com um esquema de autorizações (conhecido como **Execution Policy**) para execução de scripts e, por isso, precisamos verificar se o presente no sistema está compatível com o que o Chocolatey precisa. Execute o seguinte comando:

```
Get-ExecutionPolicy
```

Bash ▾

Caso ele retorne `Restricted`, execute o comando:

```
Set-ExecutionPolicy RemoteSigned
```

Bash ▾

E escolha a opção `[A]ll - Yes to All`. Todos os prompts de confirmação que surgirem durante a instalação devem ser respondidos com a opção `y`, e o processo de instalação deve ser concluído.

- **Powershell:** novo shell apresentado pela Microsoft por volta de 2005, ele apresenta diversas melhorias em relação ao **CMD**, tornando-o popular atualmente e consequentemente a nossa escolha para a NLW#02.

Escolhido o shell, vamos começar a instalação:

- Busque no campo de busca do Windows por **Windows Powershell**, clique com o botão direito em cima do programa e escolha a opção **Executar como administrador**.
- O Powershell trabalha com um esquema de autorizações (conhecido como `Execution Policy`) para execução de scripts e, por isso, precisamos verificar se o presente no sistema está compatível com o que o Chocolatey precisa. Execute o seguinte comando:

```
Get-ExecutionPolicy
```

Bash ▾

Caso ele retorne `Restricted`, execute o comando:

```
Set-ExecutionPolicy RemoteSigned
```

Bash ▾

E escolha a opção `[A]ll - Yes to All`.

`.NET Framework 4.5+`

Caso o erro apresentado seja `Exceção ao definir "SecurityProtocol": "Não é possível converter o valor "3312"`, siga [esse guia](#)

- Após o fim da instalação, feche e abra o powershell como administrador novamente e execute:

```
choco -v
```

Bash ▾

Caso ele retorne a versão do **Chocolatey**, a instalação foi um sucesso. Para finalizar, basta instalar a versão LTS mais recente do Node com o seguinte comando:

```
cinst nodejs-lts
```

Bash ▾

E escolha a opção `[A]ll - yes to all`.

Após o fim da instalação, feche e abra o powershell como administrador novamente e execute:

```
node -v  
npm -v
```

Bash ▾

Caso retorne as versões do Node e npm, sua instalação foi um sucesso.

Yarn 1

Linux (Ubuntu/Debian)

Para instalar o Yarn 1 no Linux vamos começar configurando o repositório do **Yarn** executando:

```
curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -  
echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee /etc/apt/sources.list.d/yarn.list
```

Bash ▾

Instale utilizando o seguinte comando:

```
sudo apt update && sudo apt install --no-install-recommends yarn
```

Bash ▾

Adicione ao arquivo `~/.bashrc` (ou `~/.zshrc` caso você utilize o shell Zsh) a seguinte linha:

Adicione ao arquivo `~/.bashrc` (ou `~/.zshrc` caso você utilize o shell Zsh) a seguinte linha:

```
export PATH="$PATH:'yarn global bin'"
```

Bash ▾

Feche e abra o terminal novamente, em seguida rode o comando:

```
yarn --version
```

Bash ▾

Caso retorne a versão do Yarn (acima de 1.0, abaixo de 2.0), a instalação ocorreu com sucesso.

macOS

Para instalar o Yarn 1 no macOS siga os seguintes passos, execute o comando:

```
brew install yarn
```

Bash ▾

Adicione ao arquivo `~/.bashrc` (ou `~/.zshrc` caso você utilize o shell Zsh) a seguinte linha:

```
export PATH="$PATH:'yarn global bin'"
```

Bash ▾

Feche e abra o terminal novamente. Em seguida, rode o comando:

```
yarn --version
```

Bash ▾

Caso retorne a versão do Yarn (acima de 1.0, abaixo de 2.0), a instalação ocorreu com sucesso.

Windows

Para instalar o Yarn 1 no Windows siga os seguintes passos, execute o comando no Powershell (como admin):

```
cinst yarn
```

Bash ▾

E escolha a opção `[A]ll - yes to all`.

Feche e abra o terminal novamente, em seguida rode o comando:

```
yarn --version
```

Bash ▾

Caso retorne a versão do Yarn (acima de 1.0, abaixo de 2.0), a instalação ocorreu com sucesso.

Expo

Para instalar o Expo é bem simples e o passo é o mesmo nos 3 sistemas operacionais.

- Com o Node e Yarn instalados, abra o terminal (no Windows, sem ser como admin) e execute:

```
yarn global add expo-cli
```

Bash ▾

i Caso você prefira utilizar o npm, basta executar:

```
npm install expo-cli --global
```

- Para verificar se a instalação ocorreu com sucesso, execute:

```
expo --version
```

Bash ▾



Caso o comando resulte no erro `expo : 0 arquivo`

`C:\Users\xxxx\AppData\Roaming\npm\expo.ps1 não pode ser carregado`, verifique se o **ExecutionPolicy** está configurado como `RemoteSigned`.

Se retornar a versão da cli do Expo, a instalação ocorreu com sucesso.

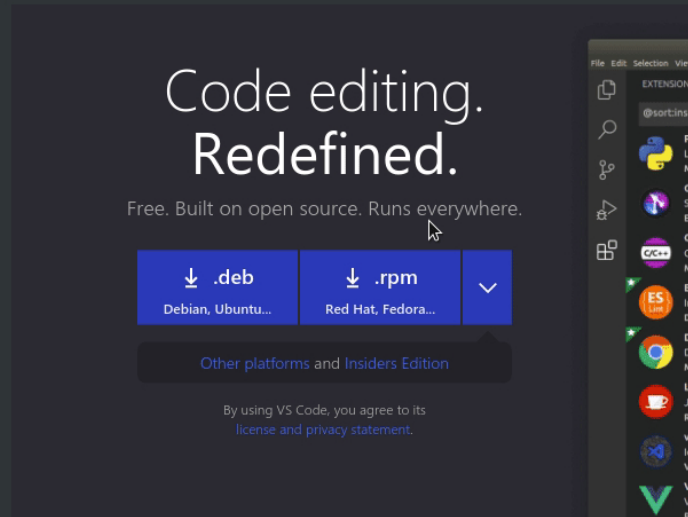


Caso a instalação da expo-cli como global no Yarn apareça que ocorreu com sucesso mas

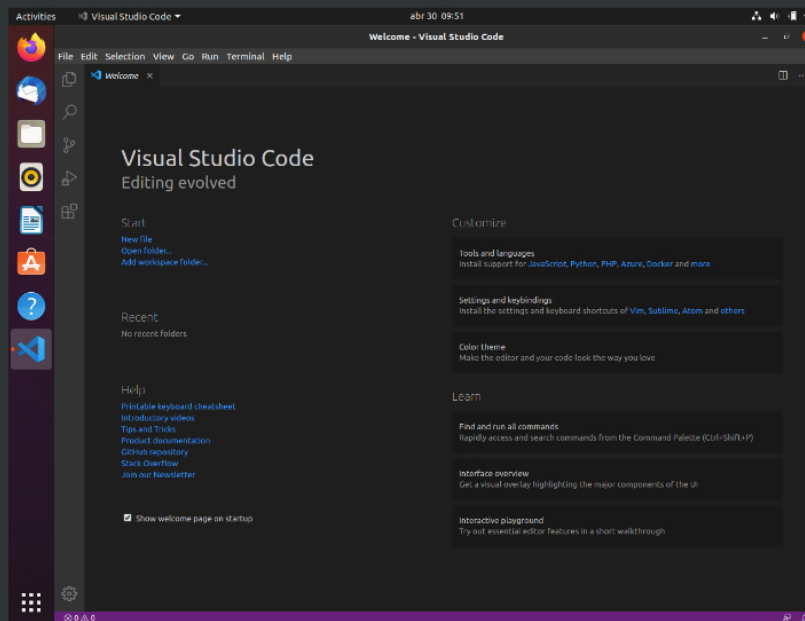
ao tentar utilizar o expo diz que o comando não existe, verifique no Linux e no macOS se você adicionou a linha `export PATH="$PATH:$yarn global bin"` ao arquivo de configuração do seu terminal.

Visual Studio Code

Para instalar o editor de texto Visual Studio Code em qualquer um dos 3 sistemas operacionais, basta [acessar o site](#), baixar e rodar o executável.



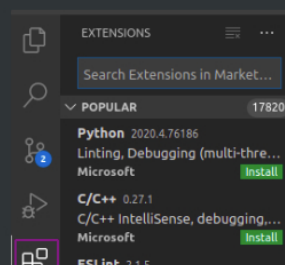
Com a instalação finalizada, abra o programa. Você deve se deparar com uma interface parecida com essa:

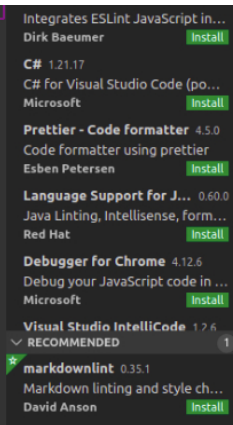


Fechamos a página **Welcome**. Para deixar a sua experiência ainda melhor, vamos passar para vocês algumas extensões e configurações especialmente escolhidas por ninguém menos que o Diego. Vamos lá

Extensões

Extensões são formas de adicionar ainda mais funcionalidades ao seu Visual Studio Code.



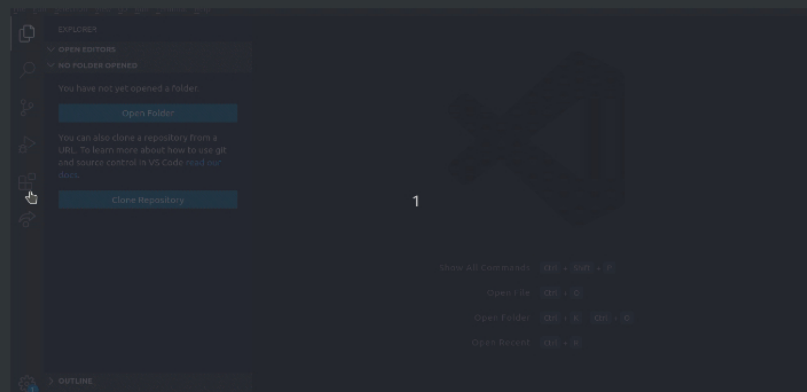


Vamos citar 2 aqui para vocês:

Omni

Nada melhor do que começar pelo tema do editor. Nós desenvolvedores trabalhamos diariamente, horas e horas, com o editor de código. Por isso, é muito importante escolher uma aparência para o Visual Studio Code que não canse demais os olhos e ao mesmo tempo realce bem o texto. É por isso que a Rocketseat decidiu criar (baseado no nosso querido Dracula) o seu próprio tema: Omni

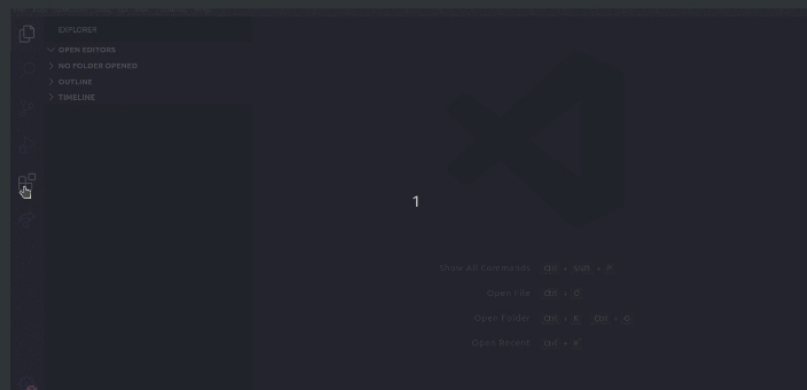
Como instalar? ➕



Material Icon Theme

O **Material Icon Theme** é uma extensão que permite a customização dos ícones das pastas por extensões de arquivos, por exemplo, com ele conseguimos customizar um ícone para arquivos **typescript**, outro para **javascript**, outro para **html** e assim por diante.

Como instalar? ➕



Configurações

Para finalizar, vamos adicionar algumas configurações no Visual Studio Code especialmente escolhidas pelo Diego. Para isso, basta pressionar **Ctrl + Shift + P** e escolher a opção **Open Settings (JSON)**. Na janela que foi aberta, adicione as configurações abaixo:

⚠ É preciso tomar alguns cuidados ao realizar essas alterações. Verifique se a configuração adicionada já não existe no arquivo. Se sim, apenas atualize o valor.

Verifique também se a todas as linhas de configuração **exceto a última** terminam com vírgula, para não gerar erro.

Por fim, caso queira substituir completamente a sua configuração pela abaixo, envolva com chaves `{ }` todo o código disponibilizado.

```
"workbench.iconTheme": "material-icon-theme",
"workbench.startupEditor": "newUntitledFile",

"explorer.compactFolders": false,
"editor.renderLineHighlight": "gutter",
"workbench.editor.labelFormat": "short",
"extensions.ignoreRecommendations": true,

"javascript.updateImportsOnFileMove.enabled": "never",
"typescript.updateImportsOnFileMove.enabled": "never",

"breadcrumbs.enabled": true,
"editor.parameterHints.enabled": false,
"explorer.confirmDragAndDrop": false,
"explorer.confirmDelete": false,

"emmet.syntaxProfiles": { "javascript": "jsx" },
"emmet.includeLanguages": { "javascript": "javascriptreact" },

"javascript.suggest.autoImports": true,
"typescript.suggest.autoImports": true,

"workbench.colorTheme": "Omni"
```

JSON ▾