Instalação das ferramentas

Node e NPM

O primeiro passo para podermos utilizar a Omnistack (Node.js, ReactJS e React Native) é instalar o Node.js, que vem acompanhado do NPM.

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:

```bash
curl --version
```

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

```bash
sudo apt install curl
```

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

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

  - Debian (como root)

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

  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/install.sh)"
```

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 consequemente 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] Sim para Todos`

⚠️ Caso o comando acima apresente erro, tente usar:

`Set-ExecutionPolicy Bypass -Scope Process`

Verifique se alteração de permissão ocorreu com sucesso executando novamente o comando:

```
Get-ExecutionPolicy
```

Bash

Alterada a permissão, basta instalar o **Chocolatey** com o comando:

```
Set-ExecutionPolicy Bypass -Scope Process -Force;
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-
Object
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
```

Bash

⛔ Caso o comando acima apresente um erro, verifique se a sua máquina atende às requisições mínimas

`Windows 7+ / Windows Server 2003+`
`PowerShell v3+`
`.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:

```
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 abre 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.

Possíveis problemas

Ao usar o Yarn no Windows para instalar as dependências nos seus projetos, atente-se para que seu nome de usuário não possua espaços, pois nesse caso, alguns erros poderão ocorrer durante esse processo, como por exemplo: com o nome "Diego Fernandes", o caminho até a pasta do projeto (supondo que estivesse na pasta *Documents*) seria algo como `C:\Users\Diego Fernandes\Documents\NLW\Projeto` e nesse caso, uma solução seria criar o projeto já na raiz do **Disco C**. Dessa forma, o caminho até a pasta não passaria pelo nome do usuário, ficando `C:\NLW\Projeto`.

Expo

Atenção: Se você ainda não chegou na aula de desenvolvimento da aplicação mobile, **não** instale o Expo ainda, pois caso você passe por algum problema específico dessa etapa, as dúvidas poderão ser tiradas dentro da comunidade da NLW apenas quando a respectiva aula for liberada e junto dela, um novo canal na comunidade.

Se você acompanhou a instalação do Expo na aula mas ainda assim está com dificuldades ou problemas na hora de realizar essa configuração, aqui você poderá encontrar mais detalhadamente todo o passo a passo desse processo e tirar possíveis dúvidas em relação à essa etapa.

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:

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

ℹ️ 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 : O 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.

Se na instalação ou utilização do Expo você enfrentou algum erro não previsto nesse guia, no nosso GitHub existe um repositório que trata de problemas comuns e você pode procurar pela solução a partir do seguinte link:

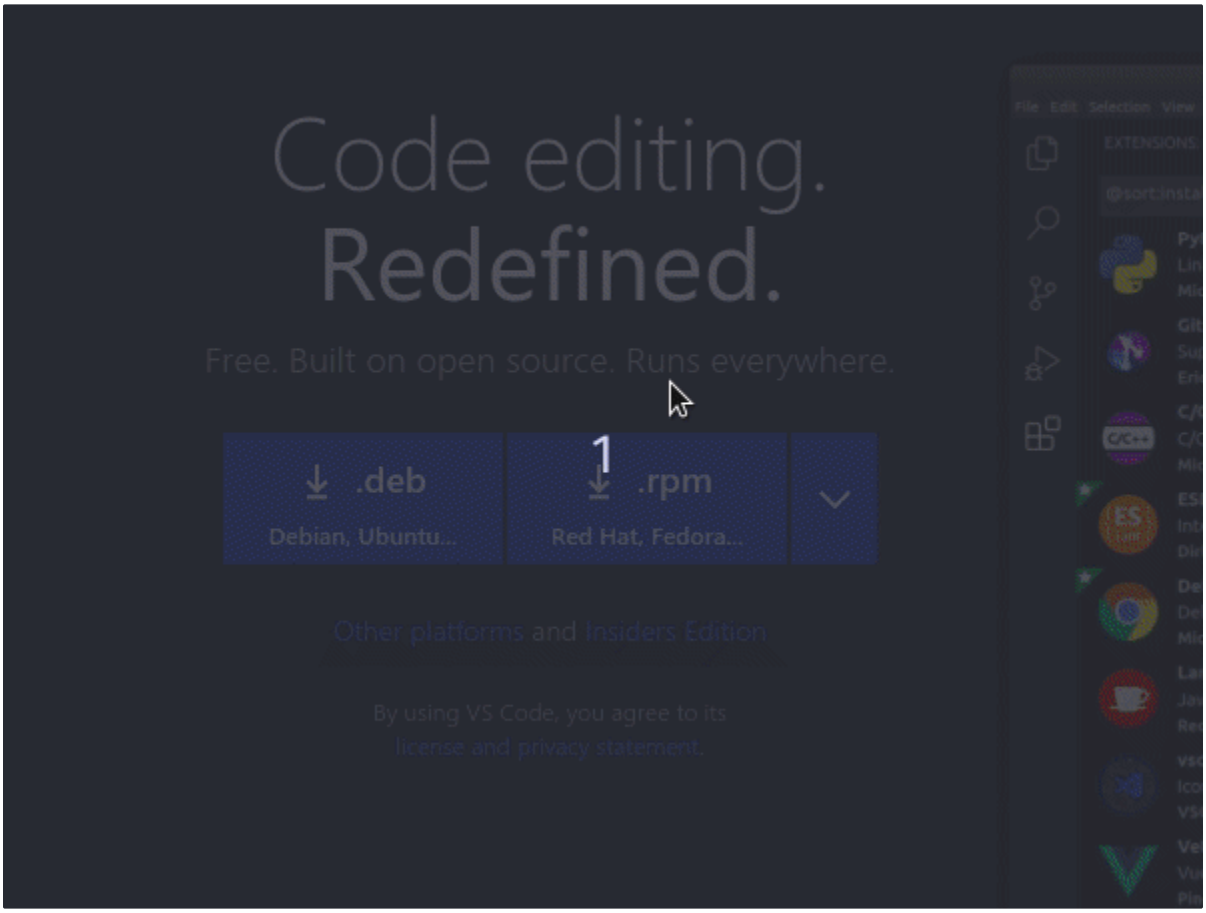
Rocketseat/expo-common-issues

Esse repositório contém uma série de erros (e suas soluções) que você pode ter com o Expo. Esse erro ocorre principalmente no

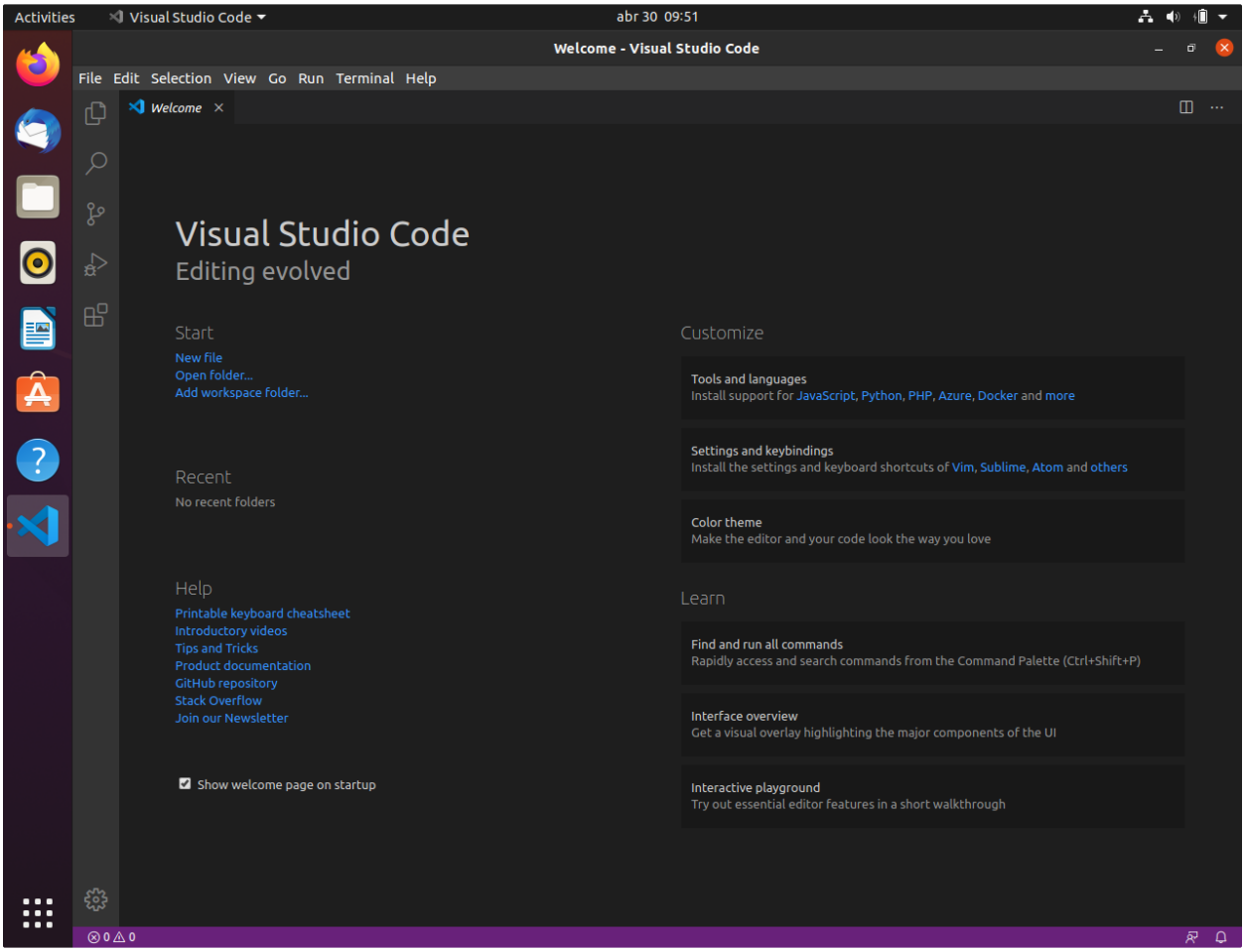
https://github.com/Rocketseat/expo-common-issues

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:



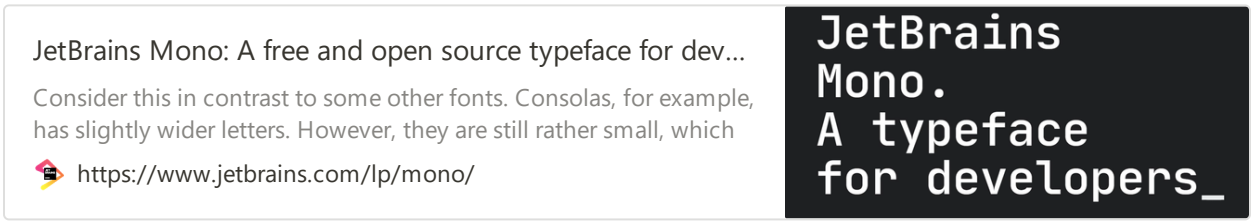
Feche a página **Welcome**. Para deixar a sua experiência ainda melhor, vamos passar para vocês algumas extensões e configurações especialmente escolhidas por ninguém menos que o Diego.

Fonte

Esse não é uma etapa obrigatória, mas se você quer usar a mesma fonte que o Diego usa nas aulas, com todos aqueles símbolos especiais quando combinamos alguns caracteres, basta seguir alguns passos:

JetBrains Mono

- A fonte utilizada nas aulas se chama JetBrains Mono e pode ser baixada gratuitamente no site da JetBrains através do seguinte link:



1 - Após baixar, extraia o arquivo e acesse a pasta `ttf`.

2 - Para instalar a fonte:

- **Windows**

  Selecione todos os arquivos que terminam com **.ttf**, clique com o botão direito do mouse e selecione **Instalar**.

- **Linux**

  Extraia as fontes para `~/.local/share/fonts` ou `/usr/share/fonts` para instalar as fontes no sistema e execute `fc-cache -f -v` no terminal.

- **Mac**

  Selecione todos os arquivos de fontes (aqueles que possuem a extensão **.ttf**), dê um duplo clique neles e clique no botão **Instalar Fonte**.

3 - O próximo passo é configurar a fonte para ser usada no Visual Studio Code. Após ter realizado as demais instalações, você poderá encontrar as configurações da fonte na seção Configurações.

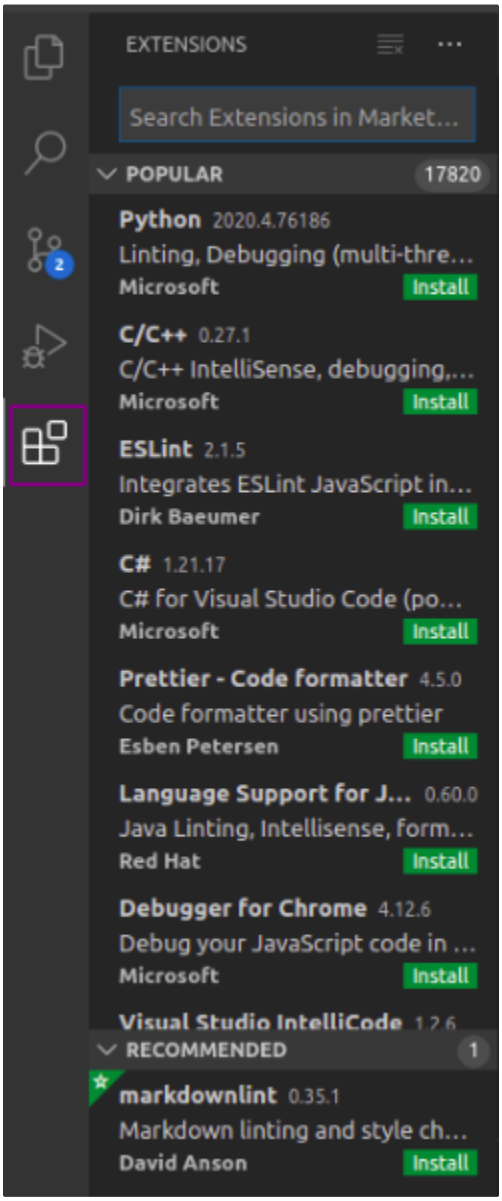
Usando as Font Ligatures

Uma das características dessa fonte é que ela possui suporte às font ligatures que é uma funcionalidade que permite combinarmos símbolos para formar um novo.

Aqui você verá como são formados os símbolos mais comuns usados durante a NLW para que não fique confuso quando ver algo diferente:

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.

```
// Configurações da fonte JetBrains Mono "editor.fontFamily": "JetBrains
Mono", "editor.fontLigatures": true, // Demais configurações
"workbench.colorTheme": "Omni", "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,
```

JSON