



# PlopJS: Criando recursos com rapidez e mais produtividade

Breno Nunes Tavares

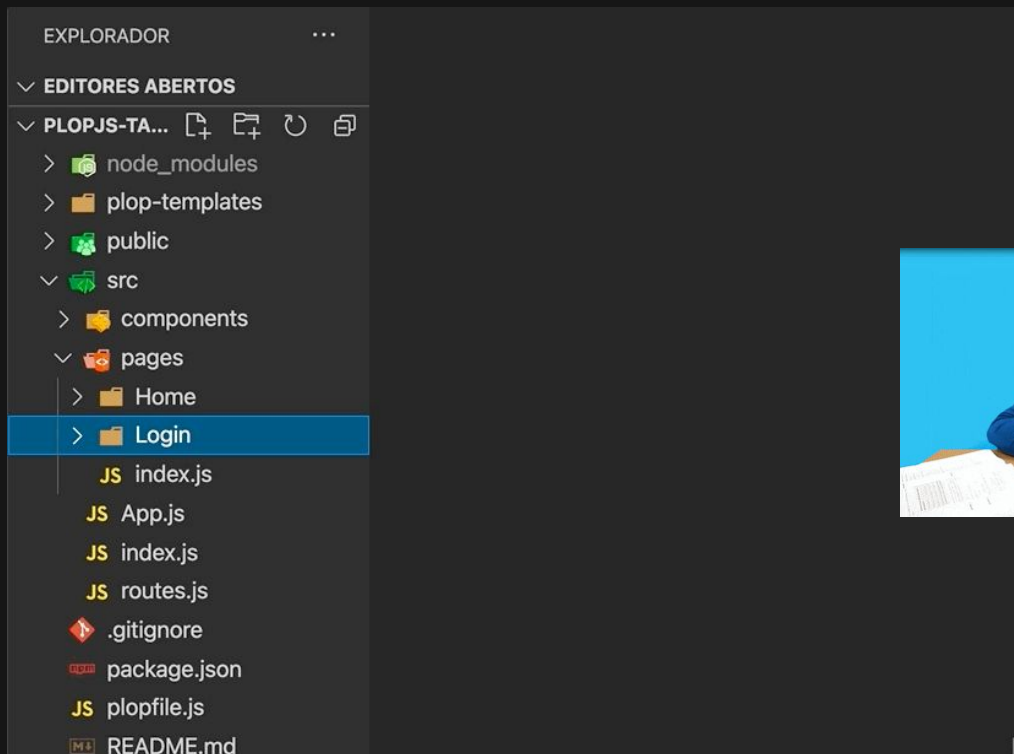


# Curiosidade: Quem é Larry Tesler?



Criador do comando ctrl + c / ctrl+v

# Qual o problema?



# Solução

> public

✓ src

> components

✓ pages

> Home

JS index.js

JS App.js

JS index.js

JS routes.js

.gitignore

package.json

JS plopfile.js M

README.md

yarn.lock

PROBLEMAS

OUTPUT

CONSOLE DE DEPURAÇÃO

TERMINAL

\$



# O que é o PlopJS?

Segundo a documentação oficial é um "micro-generator framework".

Basicamente com ele nós conseguimos gerar estruturas de código a partir de alguns comandos. Essas estruturas podem ser o padrão inicial para criação de um recurso.

Esse "recurso" no caso do front-end pode ser um componente, uma página, context ou definitivamente qualquer coisa 🤖

# Instalação do pacote plop



<# Using YARN #>

```
yarn add plop --dev
```

<# Using NPM #>

```
npm install plop --save-dev
```

## Iniciando o plopfile.js (na raiz do projeto)



```
module.exports = function (plop) {};
```

## Criando script no package.json



```
"scripts": {  
  "start": "react-scripts start",  
  "build": "react-scripts build",  
  "plop": "plop"  
},
```

# Configurando um gerador de recurso



```
// edite o arquivo plopfile.js
module.exports = function (plop) {
  // adicione os geradores com o método setGenerator
  plop.setGenerator('page', {
    description: 'Crie uma nova página',
    prompts: [], // array of prompts
    actions: []  // array of actions
  });
};
```



# Configurando um gerador de recurso - prompts

name: nome do dado inserido, é importante pois será usado nas actions e no template

```
module.exports = function (plop) {  
  plop.setGenerator('page', {  
    description: 'Crie uma nova página',  
    prompts: [  
      {  
        type: 'input',  
        name: 'name',  
        message: 'Qual será o nome da sua pagina?',  
      },  
    ],  
    actions: [], // array of actions  
  })  
}
```

type "input": você digitará o valor

# Configurando um gerador de recurso - actions

```
module.exports = function (plop) {  
  plop.setGenerator('page', {  
    description: 'Crie uma nova página',  
    prompts: [  
      {  
        type: 'input',  
        name: 'name',  
        message: 'Qual será o nome da sua página?',  
      },  
    ],  
    actions: [  
      {  
        type: 'add',  
        path: 'src/pages/{{pascalCase name}}/index.js',  
        templateFile: 'plop-templates/page/index.hbs',  
      },  
    ],  
  })  
}
```

path: caminho em que  
o arquivo será criado

pascalCase: case modifiers

templateFile: caminho para  
arquivo com template para  
criação da página

# Configurando um gerador de recurso - templateFile

plop-templates/page/index.hbs



```
import React from "react";

const {{pascalCase name}} = () => {
  return (
    <div>
      <h1>{{pascalCase name}}</h1>
    </div>
  );
};

export default {{pascalCase name}};
```

# Resultado

## prompt

PROBLEMS OUTPUT CONSOLE DE DEPURAÇÃO TERMINAL

```
$ yarn plop
yarn run v1.22.10
$ plop
? Qual será o nome da sua pagina? █
```

## actions

PROBLEMS OUTPUT CONSOLE DE DEPURAÇÃO TERMINAL

```
$ yarn plop
yarn run v1.22.10
$ plop
? Qual será o nome da sua pagina? Login
✓ ++ /src/pages/Login/index.js
🌟 Done in 3.47s.
$ █
```

```
✓ src
  ✓ pages
    ✓ Login
      JS index.js
```



## Criando ação do tipo "append"

```
plopfile.js

actions: [
  {
    type: 'add',
    path: 'src/pages/{{pascalCase name}}/index.js',
    templateFile: 'plop-templates/page/index.hbs',
  },
  {
    type: 'append',
    path: 'src/pages/index.js',
    pattern: `/* PLOP_PAGE_EXPORT */`,
    template: `export { default as {{pascalCase name}} } from "../{{pascalCase name}}",
  },
]
```

pattern: trecho de código onde será inserido o template

template: código que será inserido

## Adicionando pattern em pages/index.js



src/pages/index.js

```
/* PLOP_PAGE_EXPORT */  
export { default as Home } from "../Home";
```

# Resultado

EXPLORADOR

EDITORES ABERTOS

JS index.js src/pages

PLOPJS-TALK

node\_modules

plop-templates

public

src

components

pages

Home

Login

JS index.js

JS App.js

JS index.js

JS routes.js

.gitignore

package.json

JS plopfile.js

M

JS index.js

src > pages > JS index.js

You, 19 hours ago | 1 author (You)

1 /\* PLOP\_PAGE\_EXPORT \*/

2 export { default as Home } from "../Home";

3

PROBLEMAS

OUTPUT

CONSOLE DE DEPURAÇÃO

TERMINAL

\$



Show me the code





## Disclaimer

Os exemplos usados foram para implementação em um projeto React.js, porém é possível implementar o Plop em qualquer cenário do ecossistema Javascript.

# Isso não é tudo!

## Possibilidades:

- Adicionar trechos de código em arquivos já existentes
- Verificar se um arquivo já existe para executar ação ou não
- Diversos tipos de prompt além do "input"
- Muito mais +



# Links Úteis

[Documentação oficial](#)

[Case Modifiers](#)

[Projeto no Github](#)

## Fale Comigo



Se quiser falar sobre Plop, ou qualquer outra coisa você pode me encontrar no Twitter ou Instagram: [@brenontx](#).

Feedbacks são mais que bem-vindos!



# Obrigado!