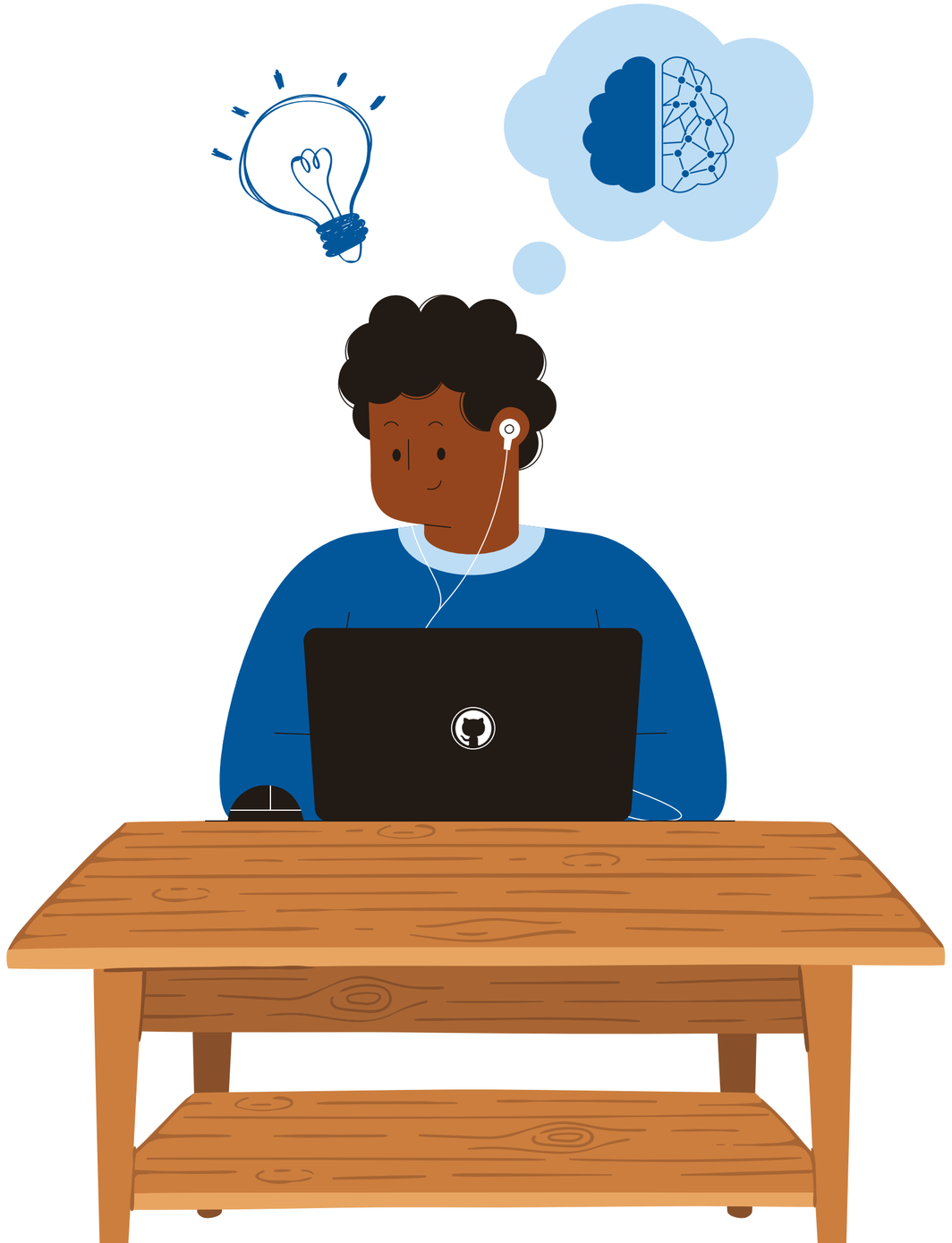
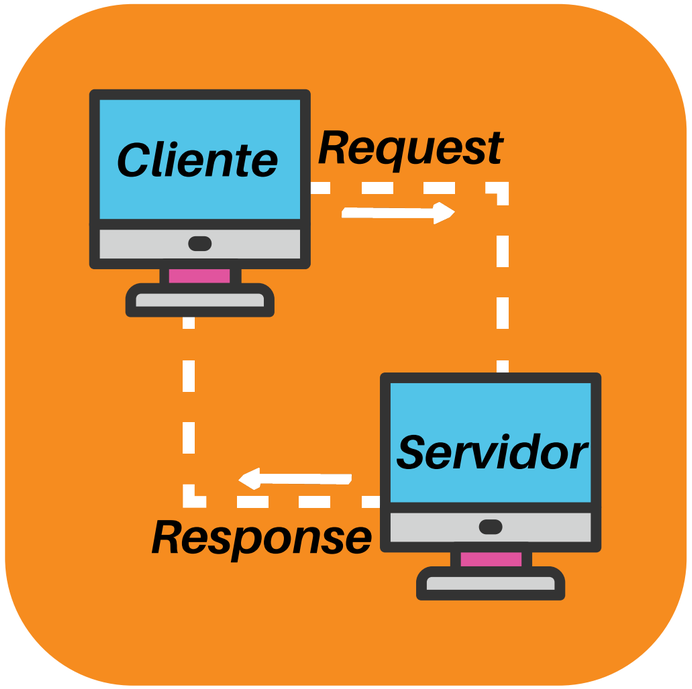
05

Backend x Frontend

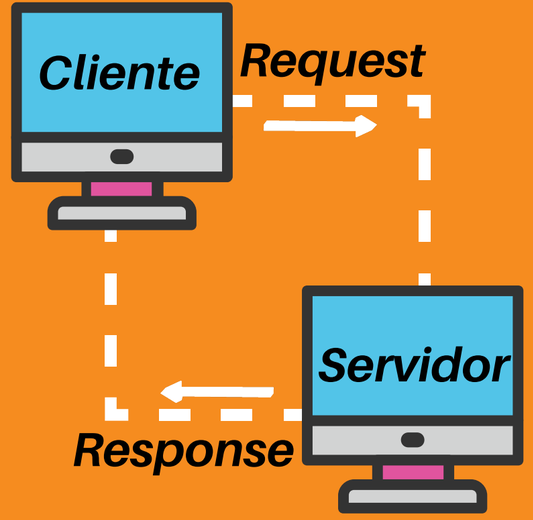


Visão cliente e servidor



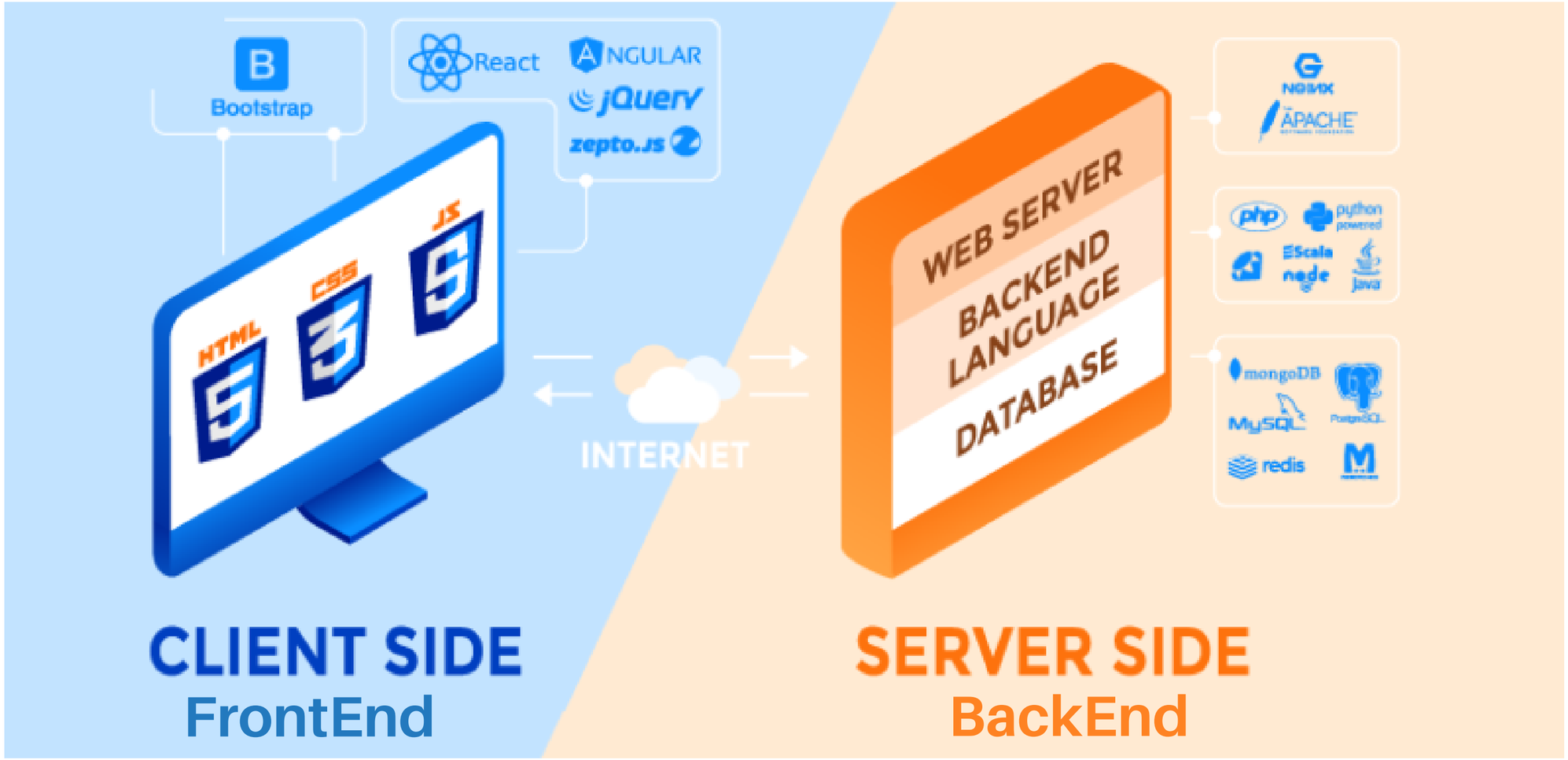
Cliente: *são os típicos dispositivos conectados à internet dos usuários da web (por exemplo, seu computador conectado ao seu Wi-Fi ou seu telefone conectado à sua rede móvel) e programas de acesso à Web disponíveis nesses dispositivos (geralmente um navegador como Firefox ou Chrome)*

Visão cliente e servidor



Servidores: *são computadores que armazenam páginas, sites ou aplicativos. Quando o dispositivo de um cliente quer acessar uma página, uma cópia dela é baixada do servidor para a máquina do cliente para ser apresentada no navegador web do usuário.*

Visão cliente e servidor



Iniciando NodeJS

```
#Verifique se você tem o node & npm instalados
node --version
npm --version
```

Caso não tenha instalado em sua máquina acesse nosso github e execute o script que está lá.

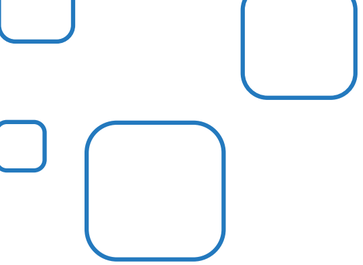
NPM

NPM: *é uma ferramenta do Node.js para o gerenciamento de pacotes. Ele permite instalar, desinstalar e atualizar dependências em uma aplicação por meio de uma simples instrução na linha de comando. Sempre que um projeto é criado por meio do gerenciador, é adicionado um arquivo chamado package.json, que contém a relação dos pacotes instalados no ambiente.*

NPM

npm

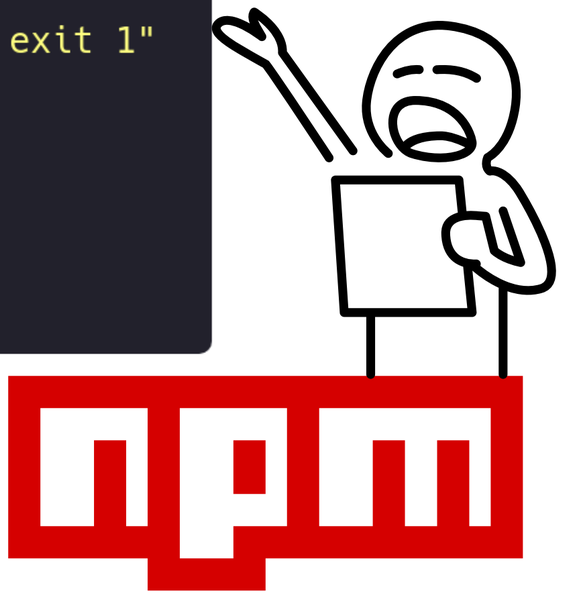
```
npm init
```



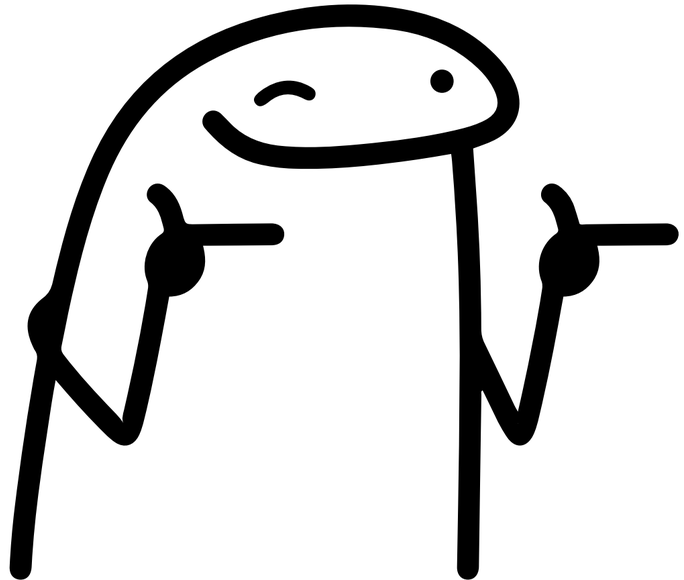
```
> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (teste)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to /home/gabriel/Documentos/teste/package.json:

{
  "name": "teste",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes)
```

SCRIPTS DO NPM

```
nano package.json
##################

{
  "name": "teste",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```



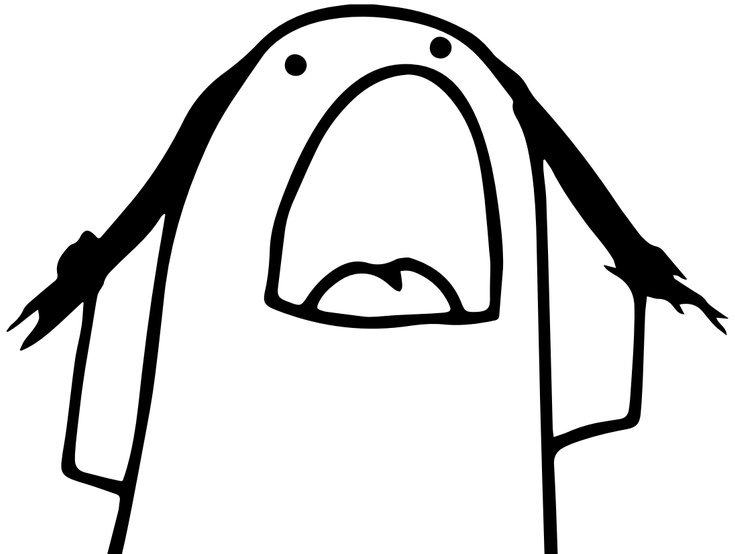
PRIMEIRO PROGRAMA



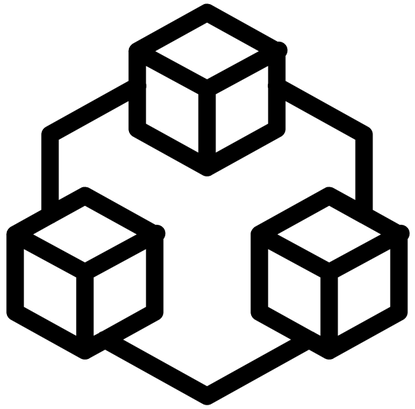
```
class Pessoa {
    nome;

    constructor(nome) {
        this.nome = nome;
    }
}

pessoa = new Pessoa("J da Silva");

console.log(pessoa);
```

VAMOS PRATICAR

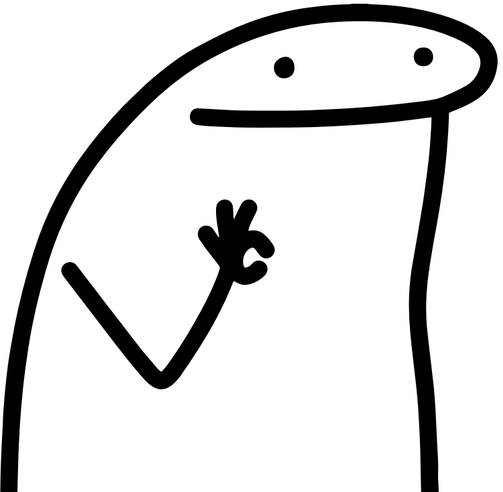
```
> Crie um sistema onde deverá ter o cadastro de 2 alunos, deve conter nome, matricula e 3 notas.
> Uma função quer irá calcular as notas e no final do sistema deve informar qual aluno teve a maior média
```



Módulos do NodeJS



Um module é uma coleção de funções e objetos do JavaScript que podem ser utilizados por aplicativos externos.
Qualquer arquivo Node.js pode ser considerado um módulo caso suas funções e dados sejam feitos para programas externos.



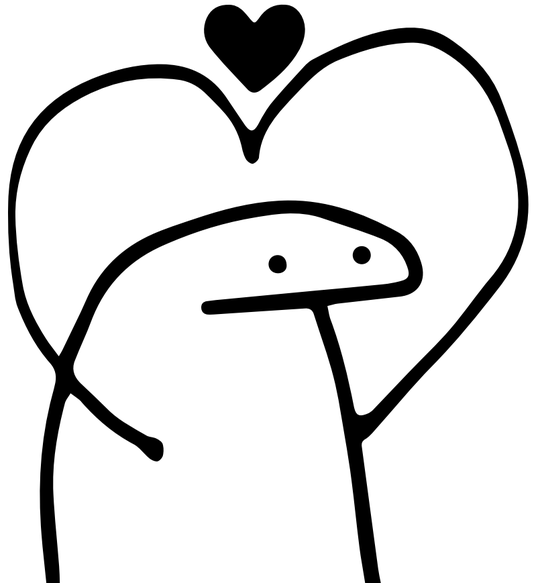
Módulos do NodeJS

Criando nosso próprio módulo

```
> Crie um arquivo chamado modulo .js
> Crie um arquivo chamado index.js
```

Módulos do NodeJS

Transforme o código de média em um módulo



Utilizando módulos nativos

*O primeiro módulo nativo que iremos estudar é o **path**, que é um módulo que tem funcionalidades que ajudam no tratamento de diretórios no NodeJS.*

https://nodejs.org/api/path.html

Utilizando módulos nativos

Utilize o mesmo site de documentação para criar um programa que identifica o sistema operacional do servidor node.

https://nodejs.org/api/path.html

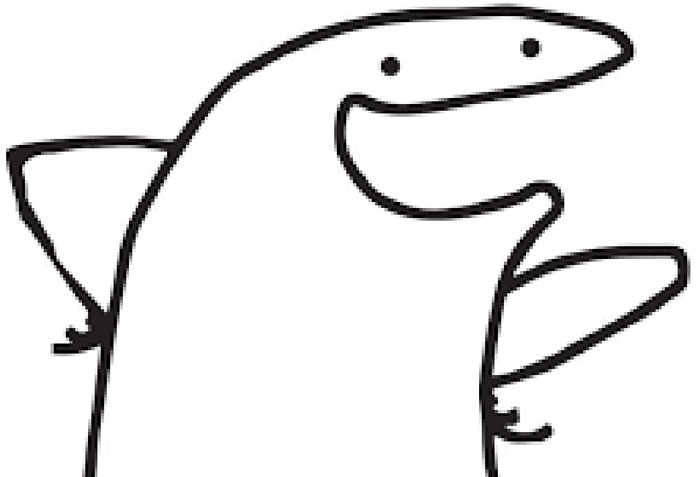
Utilizando módulos externos

Os módulos Node de terceiros desenvolvidos por desenvolvedores do Node que são disponibilizados por meio do ecossistema do Node. Mas precisamos de um gerenciador de pacotes que mantenha todos os módulos para que possam ser acessados com facilidade.
É aqui que o NPM entra em cena.

Utilizando módulos externos

https://www.npmjs.com/package/minimist

Vamos fazer uso do módulo **minimist** disponível no ***npmjs.com***. Este módulo permite que a gente passe parâmetros na chamada do node.



Utilizando módulos externos

- Faça uso de outro módulo externo chamado **chalk** - e tente mudar as cores do terminal

- https://www.npmjs.com/package/chalk/v/4.1.2

npm install chalk@4.1.2