

O que é Node-Js

Node.js é um ambiente que permite executar JavaScript fora do navegador, no servidor. Ele usa o motor V8 do Google Chrome para interpretar o JavaScript, tornando possível criar aplicações web completas com JavaScript, tanto no front-end quanto no back-end. Com Node.js, você pode criar servidores, manipular arquivos, trabalhar com bancos de dados, e mais.

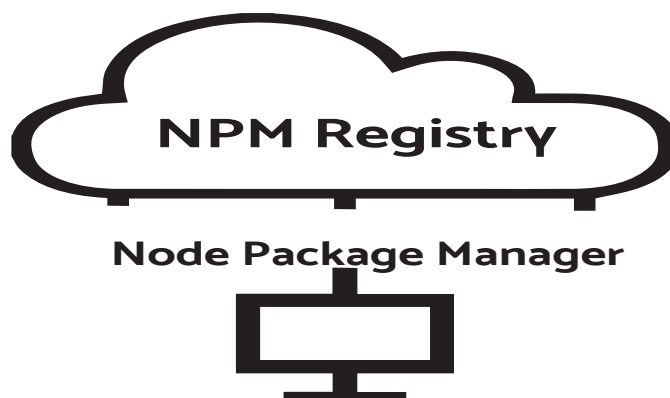


Como Funciona?

Node.js é baseado em eventos e usa um modelo assíncrono, o que significa que ele pode lidar com múltiplas conexões ao mesmo tempo sem bloquear a execução do código. Isso o torna ideal para aplicações de tempo real e de alto desempenho, como chats ou sistemas de streaming.

O que é NPM?

NPM (Node Package Manager) é o gerenciador de pacotes padrão para Node.js. Ele permite instalar, compartilhar e gerenciar bibliotecas e ferramentas de terceiros (chamados "pacotes") que podem ser usadas em seus projetos Node.js. Com o NPM, você pode adicionar funcionalidades ao seu projeto sem precisar escrever tudo do zero. O NPM baixa os pacotes de um repositório central chamado NPM Registry. Este repositório é uma enorme coleção online de pacotes (bibliotecas e ferramentas) que são disponibilizados pelos desenvolvedores. Quando você usa o comando `npm install [nome-do-pacote]`, o NPM acessa o NPM Registry, baixa o pacote solicitado e o instala no seu projeto.

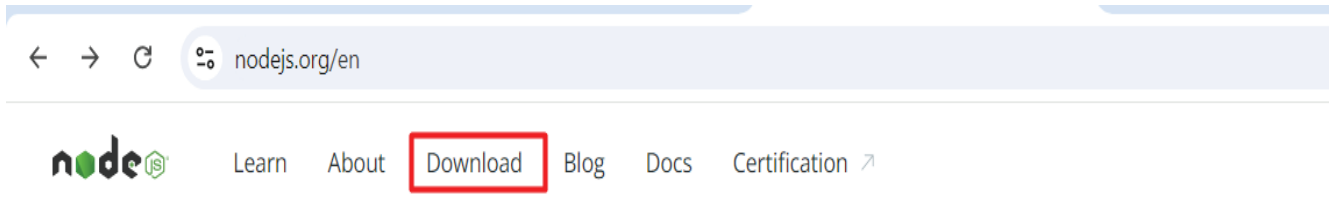


Instalando o Node.js.

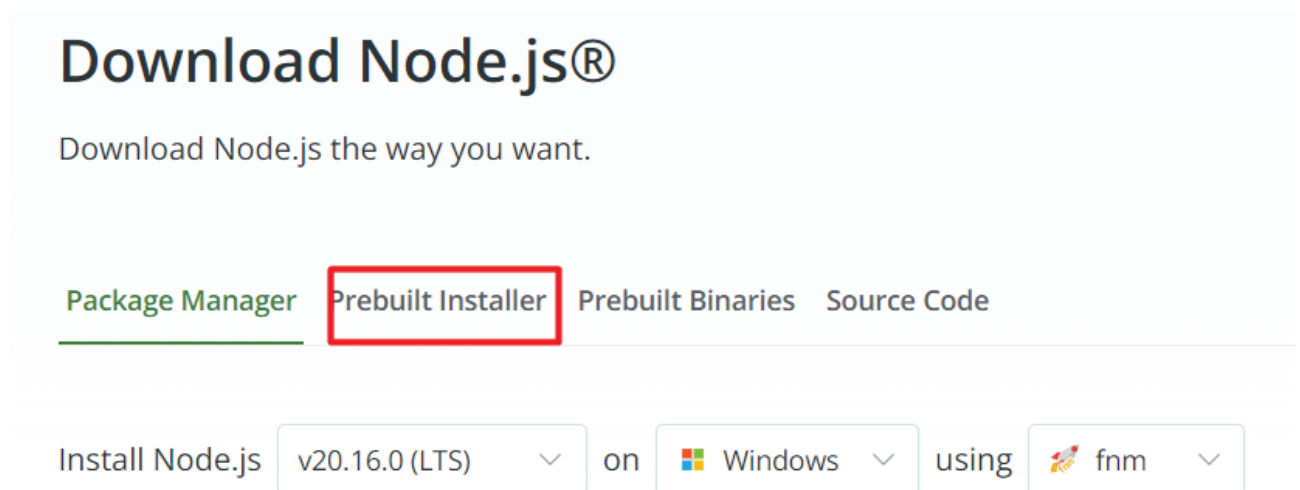
Vamos aprender a baixar o node.
primeiro da forma tradicional.

Acesse o site do node: <https://nodejs.org/en>

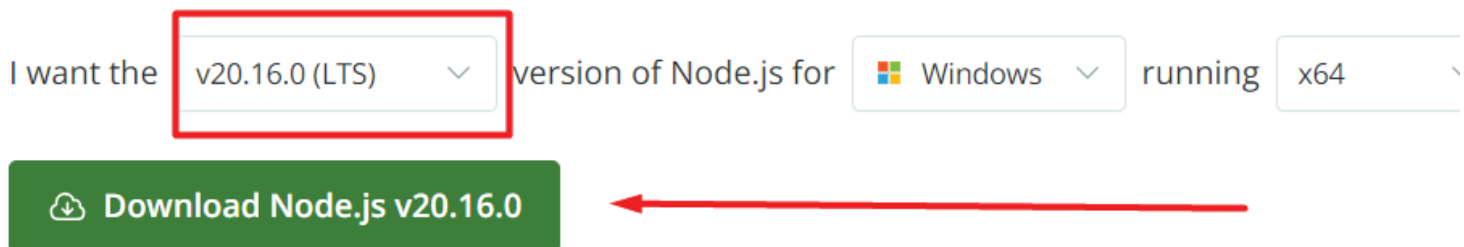
1



2



3



Para saber se o node foi instalado tem duas formas:
1 - Verificar na barra de “pesquisar” pela palavra node.

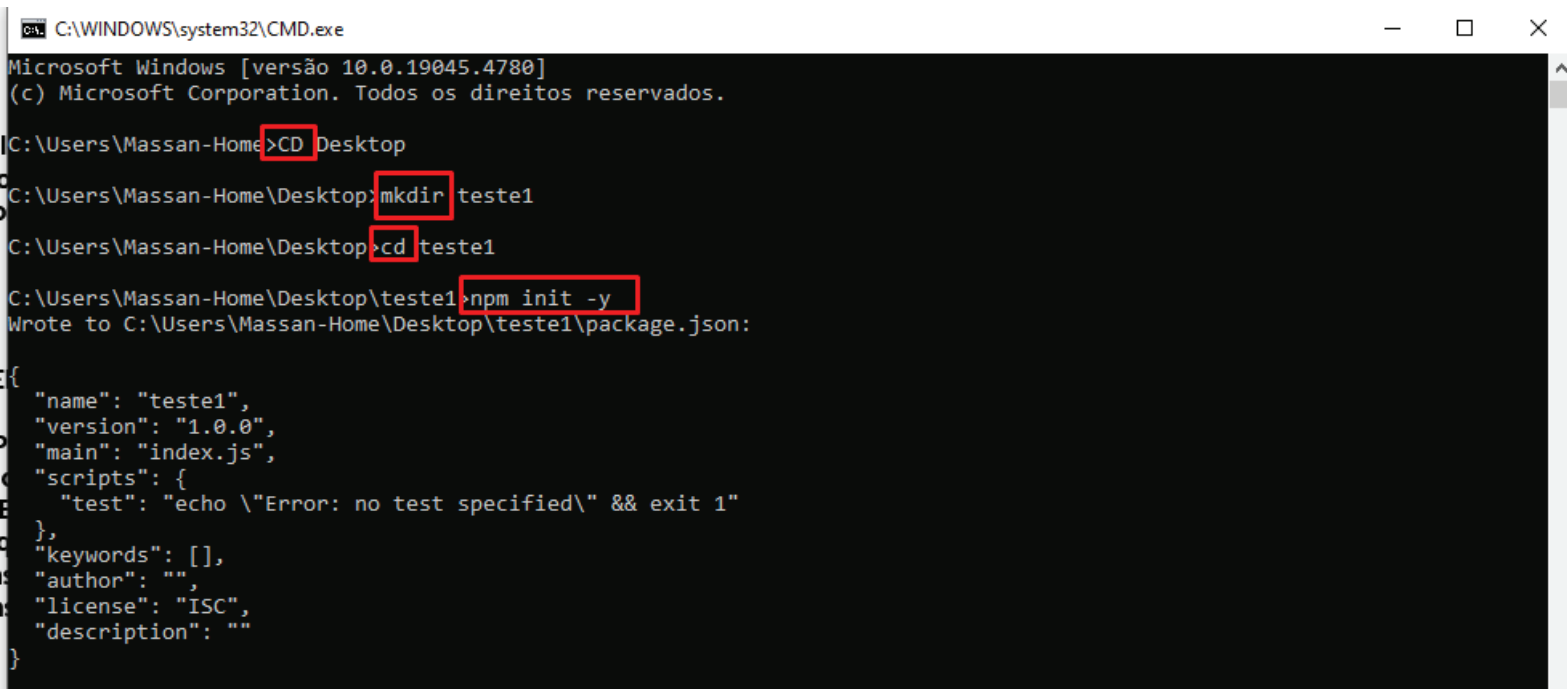


Caso exista algum erro,
é recomendavel reiniciar
o computador para completar
a instalação.

Exercício de Classe: Instalação e Uso de Dependências em Node.js.

Vamos Aprender a instalar, configurar e utilizar dependências externas em um projeto Node.js.

Crie uma pasta para o projeto e dentro dela, inicialize um novo projeto Node.js com `npm init -y`.
UTILIZE OS COMANDOS DO CMD NO PDF PASSADO.



```
C:\WINDOWS\system32\CMD.exe
Microsoft Windows [versão 10.0.19045.4780]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Massan-Home>CD Desktop
C:\Users\Massan-Home\Desktop>mkdir teste1
C:\Users\Massan-Home\Desktop>cd teste1
C:\Users\Massan-Home\Desktop\teste1>npm init -y
Wrote to C:\Users\Massan-Home\Desktop\teste1\package.json:

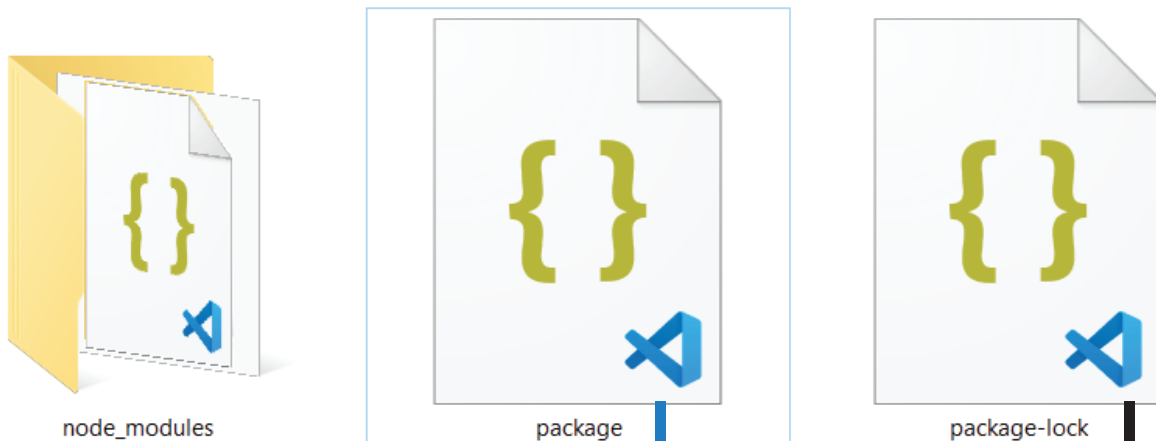
{
  "name": "teste1",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

Instalar uma Dependência:
Neste exemplo,
vamos usar a dependência `axios`,
que é uma biblioteca popular para fazer requisições HTTP
(Requisições HTTP são pedidos que um cliente
(como um navegador ou um aplicativo)
faz a um servidor para obter ou enviar dados, usando a internet.)

```
npm install axios
```

Exercício de Classe: Instalação e Uso de Dependências em Node.js.

Confira a pasta do seu projeto.



Pasta node_modules: É a pasta onde o Node.js armazena todas as dependências instaladas no seu projeto. Cada pacote que você instala com npm install é colocado nessa pasta, e ela contém o código que seu projeto precisa para funcionar.

Arquivo package.json: Este arquivo contém metadados sobre o projeto, como seu nome, versão, autor, scripts, e, o mais importante, as dependências que o projeto utiliza. Ele serve como um manual do projeto, informando ao Node.js quais pacotes devem ser instalados.

Arquivo package-lock.json: Esse arquivo é gerado automaticamente pelo npm para registrar as versões exatas das dependências instaladas. Ele garante que, quando o projeto for instalado em outro ambiente, todas as dependências sejam baixadas exatamente nas mesmas versões, garantindo consistência.

Exercício de Classe: Instalação e Uso de Dependências em Node.js.

Verifique se a dependência foi adicionada ao **package.json** na seção dependencies.

Escrever o Script Node.js:

1 - Abra a pasta no editor de texto

Utilize o comando “code .” para abrir a pasta no VsCode.

2 - Crie um arquivo chamado index.js na pasta do projeto.

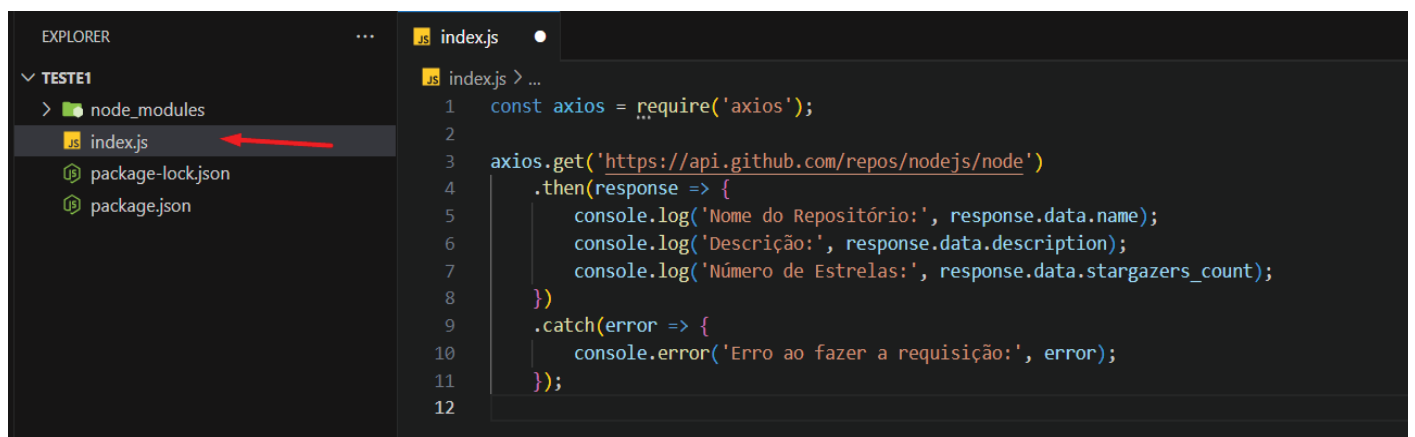
Neste arquivo, importe a dependência axios e faça uma requisição GET para uma API pública, com o código abaixo:

copie no arquivo index.js. utilizando editor de texto ou no vscode o seguinte código.
(você não precisa entender como ele funciona, só entender que ele está utilizando a dependência que instalamos.)

```
const axios = require('axios');

axios.get('https://api.github.com/repos/nodejs/node')
  .then(response => {
    console.log('Nome do Repositório:', response.data.name);
    console.log('Descrição:', response.data.description);
    console.log('Número de Estrelas:', response.data.stargazers_count);
  })
  .catch(error => {
    console.error('Erro ao fazer a requisição:', error);
  });
```

Finalizando:



Aperta “CTRL+J”

No terminal, execute o script com o comando **node index.js**.

Verifique a saída no console para ver as informações retornadas pela API do GitHub.

Agora vamos analisar a estrutura de pastas do projeto gameplay.