



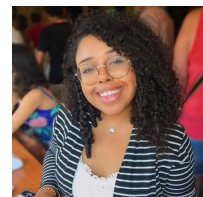
# Iniciando servidores com Node.js

Escrito por: Ana Luiza Sampaio para {Reprograma}

Me encontre no instagram: [@analui.io](https://www.instagram.com/analui.io)

no twitter: [@analui\\_io](https://twitter.com/analui_io)

ou me mande um e-mail: [sampaioanalui@gmail.com](mailto:sampaioanalui@gmail.com)



1. Garanta que você tem o node e o npm instalado no seu computador, no seu terminal escreva:

```
$ node --version
v12.18.3
$ npm --version
6.14.6
```

2. Caso você não tenha o node no seu computador, baixe-o aqui, o npm vem juntamente com o node:

## Download | Node.js

Versão LTS Mais Recente: 12.18.4 (inclui npm 6.14.6) Baixe o código fonte do Node.js ou um instalador pré-compilado para o seu sistema, e comece a desenvolver hoje. Plataformas adicionais

 <https://nodejs.org/pt-br/download/>



3. Crie a pasta em que ficará seu servidor

Dentro da sua pasta crie os arquivos referentes ao servidor, app, rotas, controller e model, como no exemplo:

```
\-- NOME-DO-SEU-SERVIDOR
|   server.js
|   \-- src
|       |   app.js
|       |   \-- controller
|       |       NOMEController.js
|       |   \-- model
|       |       NOME.json
|       |   \-- routes
|       |       NOMERoute.js
```

4. Pelo terminal entre em sua pasta referente ao Servidor, como no exemplo:

```
Ana Luiza Sampaio@DESKTOP MINGW64 ~  
$ cd Desktop/  
  
Ana Luiza Sampaio@DESKTOP MINGW64 ~/Desktop  
$ cd NOME-DO-MEU-SERVIDOR  
  
Ana Luiza Sampaio@DESKTOP MINGW64 ~/Desktop/NOME-DO-MEU-SERVIDOR  
$ ls  
server.js  src/
```

5. Inicie o git. (Se você preferir criar um repositório do git e depois clonar, você pode pular essa etapa)

```
Ana Luiza Sampaio@DESKTOP MINGW64 ~/Desktop/NOME-DO-MEU-SERVIDOR  
$ git init  
Initialized empty Git repository in  
C:/Users/Ana Luiza Sampaio/Desktop/NOME-DO-MEU-SERVIDOR/.git/  
  
Ana Luiza Sampaio@DESKTOP MINGW64 ~/Desktop/NOME-DO-MEU-SERVIDOR (master)  
$
```

6. Inicie o seu projeto com o comando **npm init**

```
Ana Luiza Sampaio@DESKTOP MINGW64 ~/Desktop/NOME-DO-MEU-SERVIDOR (master)  
$ 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: (NOME-DO-MEU-SERVIDOR) Escreva aqui o nome  
version: (1.0.0)  
description: escreva uma curta descrição  
entry point: (server.js)  
test command:  
git repository: se não vir automaticamente, coloque o link do repositório  
keywords:  
author: ana luiza sampaio  
license: (ISC)  
About to write to C:\Users\Analu\Desktop\NOME-DO-MEU-SERVIDOR\package.json:  
  
{  
  "name": "servidor",  
  "version": "1.0.0",  
  "description": "servidor para aula de backend da reprograma",  
  "main": "server.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "repository": {  
    "type": "git",  
    "url": "meugit"  
  },  
  "author": "ana luiza sampaio",  
  "license": "ISC"  
}  
  
Is this OK? (yes) yes
```

7. Repare que será criado um arquivo novo dentro do seu projeto, o package.json:

```
Ana Luiza Sampaio@DESKTOP MINGW64 ~/Desktop/NOME-DO-MEU-SERVIDOR (master)
$ ls
package.json server.js src/
```



O arquivo `package.json` é o ponto de partida de qualquer projeto NodeJS. Ele é responsável por descrever o seu projeto, informar a versão do node e do npm, url do repositório, versão do projeto, dependências de produção e de desenvolvimento dentre outras coisas.

E dentro do seu projeto o `package.json` vai estar mais ou menos assim:

```
{
  "name": "servidor",
  "version": "1.0.0",
  "description": "servidor para aula de backend da reprograma",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "https://github.com/reprograma/meu-servidor.git"
  },
  "author": "ana luiza sampaio",
  "license": "ISC"
}
```

#### 8. Instale alguns pacotes de dependências importantes para o projeto.

O primeiro é o [nodemon](#), que nos ajuda restartando nosso servidor automaticamente toda vez que fizermos uma modificação no nosso projeto, para instala-la coloque no seu terminal o comando `npm install nodemon` a resposta será a seguinte:

```
Ana Luiza Sampaio@DESKTOP MINGW64 ~/Desktop/NOME-DO-MEU-SERVIDOR (master)
$ npm install nodemon

> nodemon@2.0.4 postinstall C:\Users\Analu\Desktop\NOME-DO-MEU-SERVIDOR\node_modules\nodemon
> node bin/postinstall || exit 0

npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@~2.1.2 (node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ nodemon@2.0.4
added 120 packages from 54 contributors and audited 121 packages in 53.443s

10 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Em seguida instale o [express](#), ele é nos ajuda a criar e gerenciar as chamadas HTTP e montar o servidor com mais facilidade. Para instala-lo você deve colocar no seu terminal o comando `npm install express` a resposta será a seguinte:

```
Ana Luiza Sampaio@DESKTOP MINGW64 ~/Desktop/NOME-DO-MEU-SERVIDOR (master)
$ npm install express

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ express@4.17.1
added 56 packages from 37 contributors and audited 177 packages in 19.035s

10 packages are looking for funding
  run `npm fund` for details
```

```
run `npm fund` for details
found 0 vulnerabilities
```

Perceba que agora no seu package.json tem algumas novas linhas que indicam as dependências usadas e necessárias para o seu projeto:

```
{
  "name": "servidor",
  "version": "1.0.0",
  "description": "servidor para aula de backend da reprograma",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "repository": {
    "type": "git",
    "url": "https://github.com/reprograma/meu-servidor.git"
  },
  "author": "ana luiza sampaio",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.1",
    "nodemon": "^2.0.4"
  }
}
```

Perceba também que um novo arquivo foi criado o package-lock.json e uma nova pasta a node\_modules:

```
Ana Luiza Sampaio@DESKTOP MINGW64 ~/Desktop/NOME-DO-MEU-SERVIDOR (master)
$ ls
node_modules/ package.json package-lock.json server.js src/
```



Sempre que você instalar um pacote do npm, ele será referenciado no package-lock.json e será instalado na pasta node\_modules.

Na node\_modules estarão baixadas as dependências que o seus pacotes precisarão pra funcionar e o package-lock especifica a versão e suas dependências próprias, assim, a instalação criada será sempre a mesma, toda vez.

9. Dentro do projeto, crie o arquivo **.gitignore** e adicione a node\_modules. Dentro do .gitignore ficará assim:

```
node_modules/
```

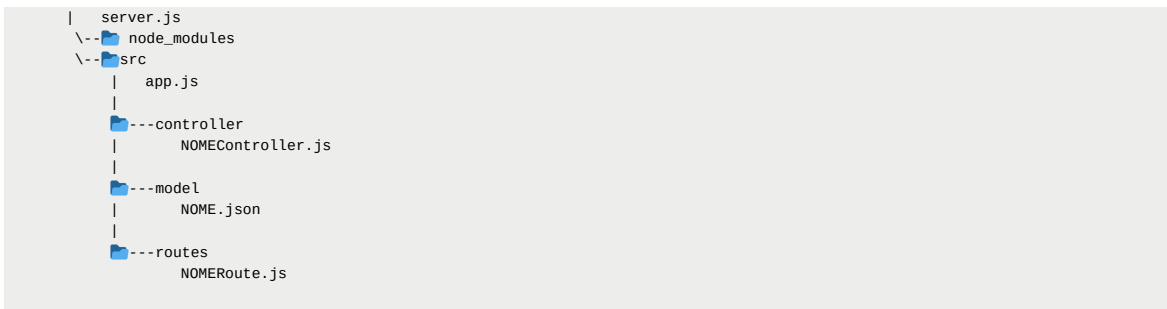


Nós ignoramos a node\_modules pois nela estão todos os downloads de todas as dependências do projeto, se apagarmos ela só precisamos dar o comando **npm install** que as dependências serão baixadas de novo e pasta node\_modules reaparecerá.

O npm sabe quais dependências baixar pois elas estão referenciadas no package.json e no package-lock.json.

10. Agora você terá essa organização de pastas:

```
\-- NOME-DO-SEU-SERVIDOR
|  .gitignore
|  package-lock.json
|  package.json
```



11. Procure dentro do seu `package.json` a chave **"scripts"**, nela são colocados os comando e suas ações. Verifique se existe o comando **"start"**, caso não, adicione o start do server pelo nodemon, dessa forma:

```
{
  "name": "servidor",
  "version": "1.0.0",
  "description": "servidor para aula de backend da reprograma",
  "main": "server.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "nodemon server.js"
  },
  "repository": {
    "type": "git",
    "url": "https://github.com/reprograma/meu-servidor.git"
  },
  "author": "ana luiza sampaio",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.1",
    "nodemon": "^2.0.4"
  }
}
```

12. Agora pode começar a codar seu servidor!