



Módulos (Node.js modules)

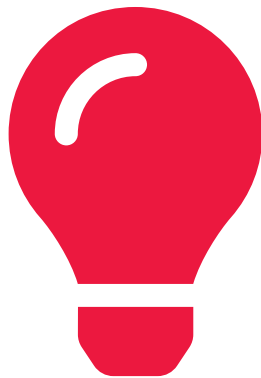


**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School



Um módulo é **um bloco de código reutilizável**, cuja existência ou não, não altera o comportamento de outros blocos de código. A partir disso, o Node.js se propõe a fragmentar nosso código em pequenos módulos, onde **cada um terá uma funcionalidade específica para atingir um objetivo definido.**





Tipos de módulos

Existem três tipos de módulos:

- Módulos **nativos**: já vêm instalados quando criamos um projeto de Node.js.
- Módulos **de terceiros**: podemos instalá-los usando o gerenciador de pacotes npm, que vamos conhecer mais adiante.
- Módulos **criados**: aqueles criados por nós mesmos.



Por convenção, o nome da variável que armazena o módulo que estamos solicitando deve receber **o mesmo nome do módulo**, ou uma abreviatura.



Módulos criados

Para utilizar um módulo criado por nós mesmos, primeiro devemos criar um arquivo com extensão **.js** e, dentro dele, digitar o script que precisamos.

Uma vez definido nosso código, temos que deixá-lo acessível para poder importá-lo dentro de nossa aplicação. Para isso, temos que fazer uso do objeto nativo **module** e de sua propriedade **.exports**. Atribuimos a ele o nome da variável que contenha a informação que queremos exportar.





{código}

```
let series = ['Friends', 'Mr. Robot',  
             'Breaking Bad'];
```

Definimos um **array** de séries, onde em cada posição há uma série.

```
module.exports = series;
```



{código}

```
let series = ['Friends', 'Mr. Robot',  
'Breaking Bad'];
```

```
module.exports = series;
```

Fazemos uso do objeto **module** e sua propriedade **.exports**, e atribuímos a ele, a variável que queremos exportar, neste caso, `series`.

Tenha em mente que sempre que quisermos exportar módulos de um script, teremos que digitar esta linha ao final do código do módulo.



Módulos criados

Uma vez que exportamos nosso módulo, vamos ao arquivo onde o queremos importar e usamos a função `require()`.

Neste caso, passamos como argumento, o caminho para o script onde o módulo que queremos requerer está localizado. Para isso, usamos `./`. Desta forma, estamos indicando ao Node.js que o caminho para chegar a esse módulo começa desde onde estamos parados (`app.js`), até o nome do arquivo que lhe passamos.



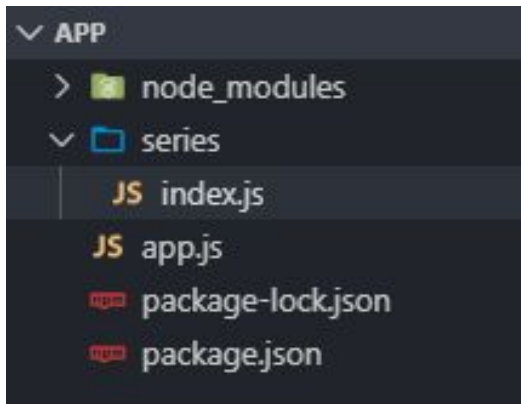
Como boa prática, os módulos que criamos são geralmente armazenados dentro de uma pasta com o mesmo nome do módulo que estamos prestes a criar.



Módulos criados

Quando nos referimos aos arquivos JavaScript, **não precisamos escrever a extensão do arquivo no caminho.**

```
JS  const series = require('./series/index');
```





Como requerer um módulo?

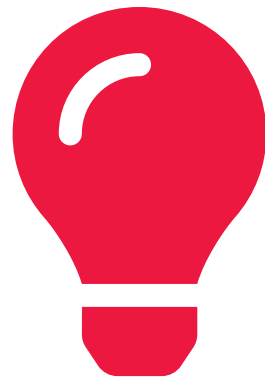
Para requerer um módulo, é necessário estar dentro do arquivo ao qual queremos incorporá-lo e fazer uso da função nativa do Node.js: `require()`. Esta recebe como parâmetro um string que será o nome do módulo.

Esta função devolve um objeto literal, portanto, é importante guardar a execução em uma variável para poder acessar, através de **dot notation**, todas as propriedades e funcionalidades do módulo.

```
js let modulo = require('nomeDoModulo');
```



Para poder ver **tudo** o que um módulo trás consigo, podemos fazer um **console.log()** da **variável** em que o armazenamos.



DigitalHouse>
Coding School