

PROGRAMAÇÃO _

FRONT-END _

DATA SCIENCE _

INTELIGÊNCIA ARTIFICIAL _

DEVOPS _

UX & DESIGN _

MOBILE _

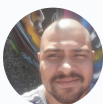
INOVAÇÃO & GESTÃO _

Artigos > [Front-end](#)

Utilizando export.modules no Node.js

```
app.js
app/aplicacaoEscola/app.js:5
ssores);

prime is not a function
s> (/home/felipe/Downloads/app/apl
```

**Felipe Nascimento**

16/05/2019

COMPARTILHE





MATRICULE-SE

No [artigo sobre arrow functions](#) criamos um módulo para imprimir o código da turma e o nome dos funcionários da escola. Agora o back-end precisa utilizar esse mesmo módulo, mas por enquanto o nosso código só funciona no lado do navegador.

O [back-end](#) possui um único arquivo chamado `app.js` e o que queremos é encontrar uma maneira de reutilizar nosso módulo dentro desse arquivo sem precisar reescrever a classe novamente. Como podemos fazer isso?

Importando nosso código

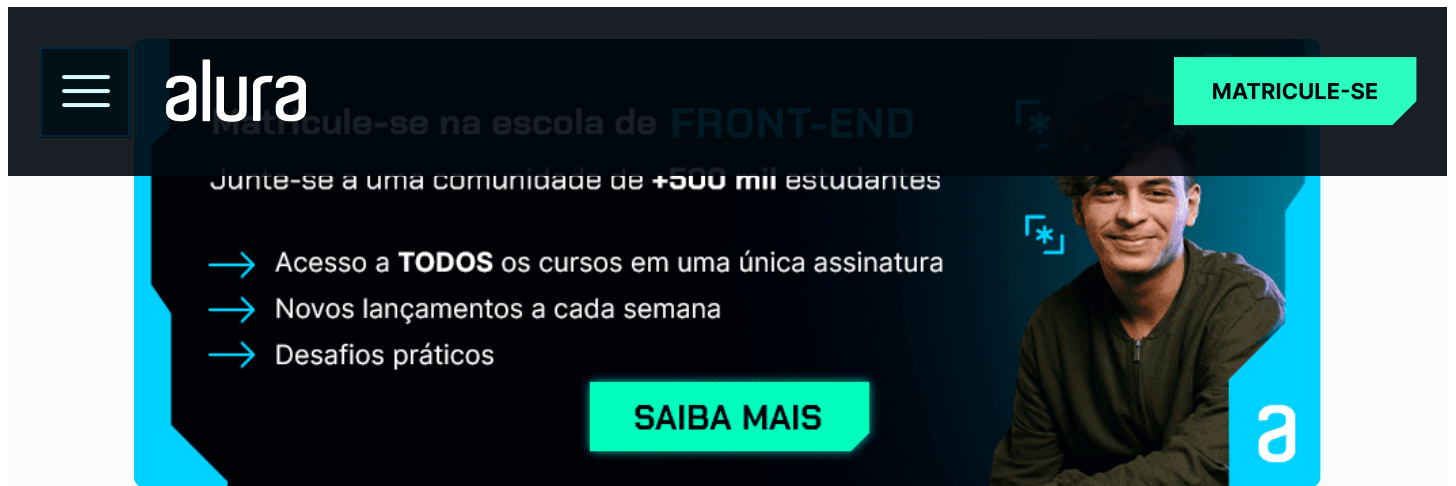
Como o back-end é escrito em [Node.JS](#), podemos fazer uso de uma função chamada **require** que vai ser responsável por chamar o nosso módulo `moduloDeImpressao.js` para dentro do nosso `app.js`. É essa função que vai ler e interpretar nosso módulo. Assim podemos reutilizar o nosso código sem precisar reescrever a nossa classe.



```
const impressao = require('./moduloDeImpressao');
```

```
→ aplicacaoEscola node app.js
/home/felipe/Downloads/app/aplicacaoEscola/app.js:5
impressao.imprime(professores);
      ^
TypeError: impressao.imprime is not a function
    at Object.<anonymous> (/home/felipe/Downloads/app/aplicacaoEscola/app.js:5:1
1)
```

Como podemos ver, algo de errado aconteceu. Se o **require** é o responsável por chamar o nosso módulo para a aplicação principal, por que então não conseguimos utilizar o método `imprime` para imprimir os nomes dos funcionários e o código da turma?

A banner for Alura, a front-end school. It features a dark background with a blue and green color scheme. On the left, there's a menu icon and the Alura logo. The text 'Matricule-se na escola de FRONT-END' is prominent. Below it, it says 'Junte-se a uma comunidade de +500 mil estudantes'. Three bullet points list benefits: 'Acesso a TODOS os cursos em uma única assinatura', 'Novos lançamentos a cada semana', and 'Desafios práticos'. A 'MATRICULE-SE' button is in the top right. A 'SAIBA MAIS' button is at the bottom center. On the right, there's a photo of a smiling man and a small 'a' logo in a blue circle.

alura

Matricule-se na escola de FRONT-END

Junte-se a uma comunidade de +500 mil estudantes

- Acesso a **TODOS** os cursos em uma única assinatura
- Novos lançamentos a cada semana
- Desafios práticos

MATRICULE-SE

SAIBA MAIS

a

Exportando nosso código

O **require** é responsável por interpretar e retornar o conteúdo do código para que possamos utilizar na aplicação principal, porém o que está acontecendo é que ele não está encontrando nada exportável dentro do arquivo, ou seja, nada está publico.

Para que o nosso código funcione na aplicação principal além de importar temos que lembrar de exportar também, ou seja, deixar o nosso código visível para a aplicação principal. Para fazermos isso utilizamos o [module.export](#), com isso nosso código fica da seguinte maneira:

```
class ModuloDeImpressao {  
  constructor() {  
    this._codigo = 10;  
  }  
  imprime(nomes) {  
    nomes.forEach(nome => {  
      console.log(`${this._codigo}: ${nome}`);  
    });  
  }  
}  
  
module.exports = ModuloDeImpressao;
```

Agora conseguimos ver a saída:

```
→ aplicacaoEscola node app.js
10: Elias
10: Yuri
10: Gabriel
10: Guilherme
10: Yan
```

Com a utilização do `module.export` conseguimos atingir nosso objetivo que era tornar nosso código reutilizável sem precisar reescrever dentro da aplicação principal, mas como o `module.exports` funciona?

Exports

O sistema de módulos do `Node.js` é responsável por criar o objeto `module.export` e o **export** aponta para esse objeto, podendo ser usado para **retornar funções e objetos** bastando somente adicioná-los ao export.



```
exports.imprimir = function( ) {
  console.log(' Meu nome é Yan ');
}
```

Como o exports é uma referência para o `module.exports` utilizamos essa sintaxe apenas para adicionar objetos e funções, se precisarmos exportar algo como uma classe vamos precisar utilizar o `module.exports`

Se ficou interessado em como o Node.js funciona e como você pode utilizá-lo melhor, aqui na **Alura** temos uma [formação desenvolvedor Node.js](#). Nela, você verá **como criar um servidor web, trabalhar com persistência de dados, utilizar middlewares**, dentre outras muitas coisas.

Confira neste artigo:

- [Importando nosso código](#)
- [Exportando nosso código](#)

**Felipe Nascimento**

Desenvolvedor e instrutor na Alura com foco em JavaScript.

[← Artigo Anterior](#)[Próximo Artigo →](#)

[Markdown: como trabalhar com essa linguagem de markup?](#)

[Como automatizar um code styleguide em projetos - Dicas de React #003](#)

Leia também:

- [Conhecendo Arrow Functions](#)
- [Alterando CSS com JavaScript](#)
- [Criando um autocomplete com JavaScript](#)
- [ES6 - Desestruturando objetos](#)
- [jQuery: DOM, AJAX e Máscaras](#)
- [Guia de JavaScript: o que é e como aprender a linguagem mais popular do mundo?](#)
- [Node.JS: o que é, como funciona esse ambiente de execução JavaScript e um](#)
- [Guia para iniciar](#)



Veja outros artigos sobre
[Front-end](#)

Quer mergulhar em tecnologia e aprendizagem?

Receba a newsletter que o nosso CEO escreve pessoalmente, com insights do mercado de trabalho, ciência e desenvolvimento de software

ME INSCREVA

alura

Nossas redes e apps



Institucional

[Sobre nós](#)

[Trabalhe conosco](#)

[Para Empresas](#)

[Para Escolas](#)

[Política de Privacidade](#)

A Alura

[Formações](#)

[Como Funciona](#)

[Todos os cursos](#)

[Depoimentos](#)

[Instrutores\(as\)](#)



Compromisso de Integridade

alura

Tecnologia em Uso

Dev em <T>

Luri by ChatGPT

MATRICULE-SE

Status

Conteúdos

[Alura Cases](#)[Imersões](#)[Artigos](#)[Podcasts](#)[Artigos de educação corporativa](#)

Fale Conosco

[Email e telefone](#)[Perguntas frequentes](#)

Novidades e Lançamentos

ENVIAR

CURSOS

Cursos de Programação

[Lógica](#) | [Python](#) | [PHP](#) | [Java](#) | [.NET](#) | [Node JS](#) | [C](#) | [Computação](#) | [Jogos](#) | [IoT](#)

Cursos de Front-end

[HTML](#), [CSS](#) | [React](#) | [Angular](#) | [JavaScript](#) | [jQuery](#)

Cursos de Data Science

[Ciência de dados](#) | [BI](#) | [SQL e Banco de Dados](#) | [Excel](#) | [Machine Learning](#) | [NoSQL](#) | [Estatística](#)

Cursos de Inteligência Artificial

[IA para Programação](#) | [IA para Dados](#)



Cursos de DevOps

AWX | Azure | Docker | Segurança | IaC | Linux

alura

MATRICULE-SE

Cursos de UX & Design

Usabilidade e UX | Video e Motion | 3D

Cursos de Mobile

React Native | Flutter | iOS e Swift | Android, Kotlin | Jogos

Cursos de Inovação & Gestão

Métodos Ágeis | Softskills | Liderança e Gestão | Startups | Vendas