

 02

## (Obrigatório) Download do projeto e infraestrutura

Olá, sou Flávio Almeida (twitter: @flaviohalmeida). Neste exercício você encontrará o link de download do projeto do curso e toda infraestrutura necessária para poder dar continuidade com seus estudos.

### O arquivo do projeto

O projeto que utilizaremos no curso deve ser baixado [aqui](https://s3.amazonaws.com/caelum-online-public/772-js-padrao-funcional/stages/01-project.zip) (<https://s3.amazonaws.com/caelum-online-public/772-js-padrao-funcional/stages/01-project.zip>). É um arquivo zip que precisa ser descompactado em sua Área de Trabalho ou qualquer outro diretório que você esteja confortável em trabalhar. Porém, antes que você possa subir o servidor é importante verificar os requisitos mínimos de infraestrutura a seguir. No final desta atividade, você terá acesso aos comandos que serão utilizados para subir o servidor.

### Infraestrutura 1 - Node.js 8.0 ou superior (versões pares apenas)

Você pode baixar o Node.js da sua plataforma preferida (Windows, MAC ou Linux) em <https://nodejs.org> (<https://nodejs.org>). Depois de instalá-lo, para saber se ele está funcionando, basta abrir seu terminal preferido da sua plataforma e executar o comando (logo a seguir, veja as dicas caso o comando não funcione):

```
node --version
```

Este comando deve exibir a versão do Node instalada no terminal.

**ATENÇÃO USUÁRIOS LINUX:** algumas distribuições Linux já possuem um binário chamado **node**, que não tem nenhuma relação com o Node.js. Nestas distribuições, o binário passa a se chamar **nodejs**. Sendo assim, em todo lugar que eu referenciar o comando **node** ele deve ser trocado para **nodejs**.

### Infraestrutura 2 - Chrome!

Neste treinamento é necessário usar o **Google Chrome versão 63** ou superior. **É compulsória essa escolha do navegador.** Nós utilizaremos o sistema de módulos nativos do ES2015 suportado pelo Chrome, evitando assim que tenhamos que elaborar toda uma infraestrutura especial para isso. Outros navegadores do mercado ainda não suportam esse recurso na data de lançamento deste curso.

Caso você tenha interesse em saber como garantir que todos os módulos escritos em ES2015 sejam suportados em todos os outros navegadores, confira esses dois cursos abaixo.

- [Curso JavaScript Avançado III: ES6, orientação a objetos e padrões de projetos](https://cursos.alura.com.br/course/javascript-es6-orientacao-a-objetos-e-padrões-de-projetos) (<https://cursos.alura.com.br/course/javascript-es6-orientacao-a-objetos-e-padrões-de-projetos>)
- [Curso Webpack: Manipulando módulos na sua webapp](https://cursos.alura.com.br/course/webpack) (<https://cursos.alura.com.br/course/webpack>)

Novamente, é importante ficar claro que não nos preocuparemos com essa infraestrutura e pegaremos carona no sistema de módulos do Chrome. Aliás, em um futuro próximo ele será o padrão entre os navegadores.

### Subindo o servidor

Com os requerimentos de infraestrutura atendidos e com o zip do projeto descompactado, para levantá-lo, abra seu terminal favorito e entre na pasta `notas`. Dentro dela, execute o comando:

```
node server
```

Se você utiliza Linux e o Node.js foi instalado como `nodejs`, faça `nodejs server`.

A instrução fará com que o servidor fique de pé e exiba no console a seguinte mensagem:

```
Server is running at http://localhost:3000
```

Abra seu navegador favorito e accese o endereço `http://localhost:3000`. Deverá ser exibida uma página com um botão apenas.

Agora com tudo em seu devido lugar, podemos dar início ao treinamento. Vamos começar pelo sistema de módulos do ES2015 e seu suporte pelo Chrome.