

 05

## Baixando e configurando o Electron

### Transcrição

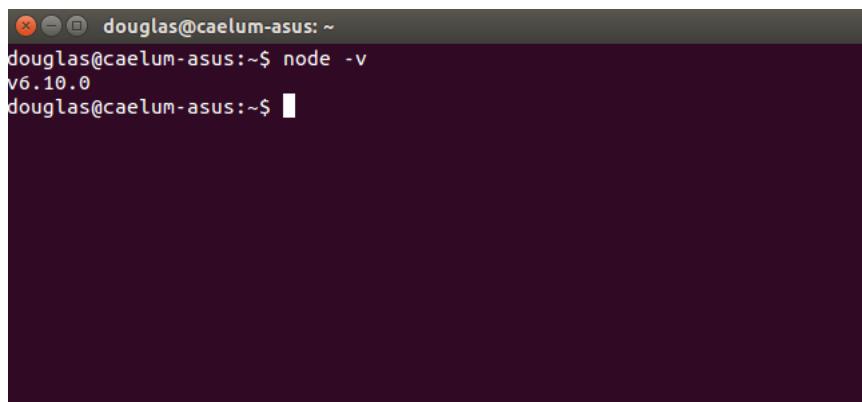
Para dar ínicio ao nosso projeto com o Electron precisamos antes garantir que temos todos os pré requisitos para construir nossa aplicação. Como vimos, o Electron depende do NodeJs para funcionar, então precisamos ter o Node em nossa máquina, pelo menos na versão 6.10.

Se você **ainda não tem** o NodeJs instalado, basta seguir o [exercício](https://cursos.alura.com.br/course/electron/task/26078) (<https://cursos.alura.com.br/course/electron/task/26078>) de instalação do mesmo.

Se você já tem o NodeJs instalado, você consegue ver a versão do mesmo utilizando o comando:

```
node -v
```

em seu terminal.



```
douglas@caelum-asus: ~
douglas@caelum-asus:~$ node -v
v6.10.0
douglas@caelum-asus:~$ █
```

Também precisaremos de um editor de textos, e já que o tema deste curso é o Electron, nada mais justo do que utilizar um editor de textos que foi feito com essa tecnologia, por isto neste curso o editor escolhido é o [Atom](https://atom.io/) (<https://atom.io/>).

### Começando o projeto: Instalando as dependências

Crie uma pasta chamada `alura-timer` que será a pasta responsável por conter nossa aplicação e lá dentro vamos começar o processo de importar as dependências com o NodeJs.

Como todo bom projeto em NodeJs, precisamos primeiro iniciá-lo com o comando:

```
npm init
```

Responda as questões com um simples `ENTER` e o comando criará o arquivo `package.json` para nós.

Com o `package.json` criado podemos começar a importar as nossas dependências e obviamente a principal delas é o próprio Electron. Use o comando abaixo para instalar o Electron:

```
npm install electron@1.6.1 --save
```

Você verá que agora o Electron aparece como uma dependency em seu package.json .

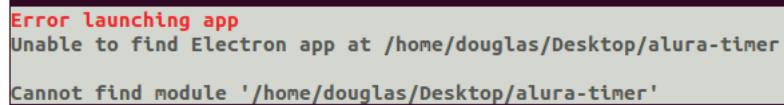
## Utilizando o Electron

Como o Electron foi instalado como um módulo do Node podemos executá-lo de um modo fácil pelo terminal, basta adicionar um novo script no package.json para isto.

Em seu package.json , no campo scripts adicione:

```
//Restante do `package.json`
...
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1",
  "start": "electron ."
},
...
```

Se você tentar rodar o comando `npm start` neste você receberá um erro dizendo:

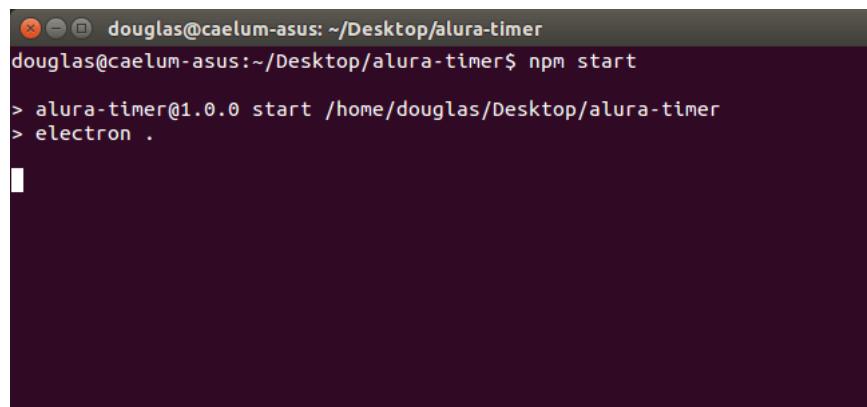


```
Error launching app
Unable to find Electron app at /home/douglas/Desktop/alura-timer
Cannot find module '/home/douglas/Desktop/alura-timer'
```

Isto acontece por que ainda não criamos um arquivo que será o ponto de partida de nossa aplicação. Crie um arquivo chamado `main.js` na raiz de seu projeto e altere o seu `package.json` para que a entrada "`main`" aponte para o arquivo recém criado:

```
//Restante do `package.json`
...
  "main": "main.js",
...
```

Agora execute novamente o comando `npm start` e veja que o terminal ficará travado:

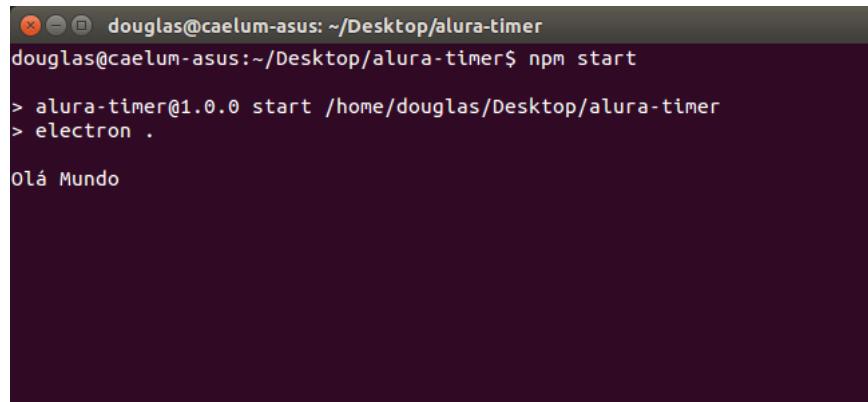


Isso significa que o Electron conseguiu achar nosso arquivo `main.js` e executou-o com sucesso, porém como ele não tem nada dentro dele, o terminal nada exibe.

Altere o `main.js` e coloque uma pequena mensagem de Olá Mundo:

```
//main.js  
console.log("Olá Mundo");
```

E execute novamente o `npm start`:



```
douglas@caelum-asus: ~/Desktop/alura-timer  
douglas@caelum-asus:~/Desktop/alura-timer$ npm start  
> alura-timer@1.0.0 start /home/douglas/Desktop/alura-timer  
> electron .  
  
Olá Mundo
```

Veja que agora nossa mensagem é exibida. O setup inicial está completo, agora vamos mergulhar de cabeça em tudo que o Electron pode nos oferecer!