

## Empacotando a aplicação

### Transcrição

Com a aplicação finalizada, como exportamos ela para executá-la, tanto no Windows, quanto no macOS, quanto no Linux? Ou seja, vamos gerar três executáveis, um para cada sistema operacional.

Para isso, utilizaremos o [electron-packager](https://github.com/electron-userland/electron-packager) (<https://github.com/electron-userland/electron-packager>), um pacote do npm que nos auxilia nesse processo. Ele é uma ferramenta **cli**, ou seja, é executado na linha de comando.

Para executá-lo, o comando é bem simples:

```
electron-packager <diretório da aplicação> <nome da aplicação> --platform=<sistema operacional>
```

Para instalá-lo, diferente das outras dependências, não precisa ser dentro do nosso projeto, pois não há necessidade da *flag* `--save`, já que como ele é uma dependência com a função de realizar o *deploy*, não faz sentido colocá-la dentro do próprio projeto, já que ele não depende dessa dependência para funcionar.

Então, executamos o seguinte comando para instalar o **electron-packager**:

```
npm install electron-packager -g
```

Dessa vez, utilizamos a *flag* `-g` para instalar a dependência globalmente, e podermos utilizá-la na linha de comando.

### Gerando um executável

Antes de gerar o executável, vamos remover a linha que abre o console de desenvolvedor automaticamente. No **main.js**, **removemos** a seguinte linha:

```
// main.js  
  
mainWindow.openDevTools();
```

Agora, para utilizar o **electron-packager** e gerar um executável para o macOS, podemos executar no terminal:

```
electron-packager alura-timer/ alura-timer --platform=darwin --arch=x64
```

Lembrando que, para o comando acima funcionar, devemos estar um nível acima do projeto **alura-timer**. Por exemplo, se o projeto está no Desktop, no terminal devemos estar dentro dessa pasta. E caso queiramos gerar um executável do Windows, basta colocar `win32` como valor de `--platform`, e do Linux é `linux`.

Na primeira vez que esse comando é executado para cada plataforma, será baixada a versão do Electron que estamos utilizando no nosso projeto para a plataforma específica, então o comando pode demorar um pouco.

Após o término do comando, é gerada uma pasta, e nela contém o executável da plataforma que selecionamos. Ao executá-lo, podemos ver finalmente que o nome do primeiro item do menu é o nome da nossa aplicação.

Mas quando damos **ALT+TAB**, o ícone que aparece é o do Electron, mas com o nome **alura-timer**. Veremos como mudar o ícone da nossa aplicação no próximo vídeo.