

Empacotando para os três S.O's

Transcrição

Conseguimos gerar o executável do **alura-timer** para um sistema operacional específico, mas uma das vantagens do Electron é que ele é multiplataforma, ou seja, podemos de uma vez gerar os três executáveis de cada sistema operacional. Mas como?

Continuaremos utilizando o **electron-packager**, mas devemos prestar atenção a alguns detalhes. Se o sistema operacional que irá gerar os executáveis for o Windows, não devemos nos preocupar, já que ele consegue gerar os executáveis dele mesmo, e tanto do Linux, quanto do macOS.

Mas quando estamos no Linux ou no macOS, por ter que gerar e manipular um executável para o Windows. E para poder manipular e rodar programas Windows tanto no Linux, quanto no macOS, precisamos instalar previamente o **Wine**.

Há dois exercícios neste capítulo para auxiliar na instalação do Wine. Para instalá-lo no macOS, há [esse exercício \(https://cursos.alura.com.br/course/electron/task/26651\)](https://cursos.alura.com.br/course/electron/task/26651), e no Linux, [esse exercício \(https://cursos.alura.com.br/course/electron/task/26665\)](https://cursos.alura.com.br/course/electron/task/26665).

Gerando de uma vez só um executável para cada plataforma

Com o Wine já instalado, para gerar os executáveis para as três plataformas, podemos passar o valor **all** para as *flags* **--platform** e **--arch**. Assim geraremos os executáveis para Windows, macOS e Linux, nas arquiteturas 32 e 64 bits de cada um.

Mas para gerar somente na arquitetura 64 bits, basta especificar isso, passando o valor **all** somente para a *flag* **--platform**:

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

Com o valor **all**, além de gerar o executável para as três plataformas, geraremos para uma quarta, para a Mac App Store, que é uma loja de aplicativos internos da Apple. Mas como não queremos gerar para essa plataforma, vamos especificar quais queremos:

```
electron-packager alura-timer/ alura-timer --platform=darwin,linux,win32 --arch=x64
```

Mas o comando está sem os ícones. Como cada plataforma possui um ícone específico, como especificamos isso no comando? O que fazemos é colocar os ícones com o mesmo nome, sendo distinguidos somente pela sua extensão, que o **electron-packager** é esperto o suficiente para saber qual ícone e extensão é de qual plataforma:

```
electron-packager alura-timer/ alura-timer --platform=darwin,linux,win32 --arch=x64 --icon=icon,
```

Na primeira vez, esse comando pode demorar bastante, já que é baixado o Electron específico de cada plataforma, para depois gerar os executáveis.

Assim, com o **electron-packager**, somos capazes de gerar um executável para cada sistema operacional, inclusive com ícone!