**Faça como eu fiz: refatorando o script**

[**PRÓXIMA ATIVIDADE**](https://cursos.alura.com.br/course/nodejs-criando-primeira-biblioteca/task/112063/next)

Finalizamos o curso com as seguintes opções para executar a biblioteca:

* Com o comando node ./src/cli.js ./arquivos/texto.md (ou com a opção de validação, node ./src/cli.js ./arquivos/texto.md --valida.
* Ou passando as instruções para um script do NPM:

"cli": "node ./src/cli.js ./arquivos/",

"cli:valida": "node ./src/cli.js ./arquivos/ --valida"COPIAR CÓDIGO

Porém, para a biblioteca funcionar como deve, deve receber qualquer caminho, não apenas o que usamos como teste durante o curso.

Vamos fazer um teste: modificar o comando npm run cli para passar o caminho do arquivo e a opção --valida em separado:

"cli": "node ./src/cli.js"COPIAR CÓDIGO

Se tentarmos executar este comando no terminal da seguinte forma:

npm run cli ./arquivos/COPIAR CÓDIGO

Tudo continua funcionando. Porém se fizermos um novo teste adicionando a opção --valida:

npm run cli ./arquivos/ --valida COPIAR CÓDIGO

A opção deixou de funcionar e o retorno continua sendo apenas a lista sem validação.

Um último teste: modificando o valor do argumento para apenas valida:

const valida = argumentos[3] === 'valida';COPIAR CÓDIGO

E executarmos novamente o comando como npm run cli ./arquivos/ valida, magicamente a validação volta a funcionar. Mas por que isso acontece?

Isso ocorre porque os hífens -- antes do argumento (ou a versão resumida de um comando com -) representam um padrão chamado de **flag**.

As *flags* servem justamente para especificar opções em um comando, que é a intenção da opção --valida. É o que acontece, por exemplo, quando executamos o comando node --version ou sua versão resumida node -v (chamamos esta versão “menor” do comando de *alias*, ou apelido).

As *flags* podem ser utilizadas tanto para “sinalizar” (daí o nome *flag*, ou “bandeira”) opções específicas em um comando de CLI, como também para indicar que o argumento seguinte é um parâmetro da opção, como por exemplo comando --flag opcao-da-flag

Se executarmos apenas node entramos no ambiente de execução do Node.js no terminal; se executarmos node arquivo.js usaremos o Node.js para executar um arquivo .js, mas ao passarmos a *flag* --version estamos pedindo ao Node.js que retorne a informação sobre a versão que está atualmente sendo utilizada para a execução.

Mas por que, afinal, a *flag* --valida não funciona com o script cli e o argumento normal valida funciona?

Isso acontece por causa da forma como o NPM interpreta e executa os scripts, que é um pouco diferente de passar os comandos “por extenso” no terminal. Neste caso, é necessário separar o que é um argumento comum das *flags* com um par extra de hífens --. Então, para a versão final, podemos deixar somente o script cli, passando o caminho e a opção --valida direto no comando:

"cli": "node ./src/cli.js",COPIAR CÓDIGO

Testando no console:

npm run cli ./caminho/caminho/arquivo.mdCOPIAR CÓDIGO

Ou, para usar a opção de validação:

npm run cli ./caminho/caminho/arquivo.md -- --validaCOPIAR CÓDIGO

O mesmo vale para um caminho de um diretório.