



# Certified Tech Developer

The Ultimate Degree



## Programação Imperativa

### Módulos

### Objetivo

Vamos realizar uma série de práticas que nos ajudarão a entender a funcionalidade e praticidade que os módulos trazem aos nossos programas.



### Criando Módulos Próprios

Nesta primeira etapa, iremos criar nosso primeiro módulo localmente para acessá-lo entre nossos arquivos.

1. Em um **arquivo.js**, crie uma função (ou várias). Pode ser uma função que adicione dois parâmetros ou qualquer operação que você deseje.
2. Modifique seu arquivo para que seja um módulo que exporte a função criada na etapa anterior. Lembre-se de usar a palavra reservada correta.



3. Em outro arquivo JavaScript (**arquivo2.js**), dentro da mesma pasta de trabalho, solicite o módulo que você exportou.
4. Verifique se sua função funciona corretamente.

Você pode imaginar em quais casos, ou programas, a criação de módulos poderia nos resolver e até mesmo nos salvar de grandes problemas?

Se você pensou em programas grandes com muitas linhas de código, acertou! Conforme nossos programas crescem e começam a escalar, o número de linhas de código também aumenta, e isso pode resultar em um problema de leitura e compreensão para todos que interagem com o seu código. Os módulos vêm para dividir nossos arquivos de forma que, embora tenhamos um número maior de arquivos, eles seriam menos pesados entre si e conteriam menos linhas de código, tornando mais fácil de entender.



## Instalando Módulos externos

Agora que sabemos como criar, exportar e exigir nossos próprios módulos, vamos ver como poderíamos usar os módulos criados pela comunidade e quais as vantagens que eles têm:

1. Seguindo o que vimos na aula como exemplo, procure um módulo na internet que seja do seu interesse.
2. Depois de encontrá-lo, em uma nova pasta de trabalho, crie um **arquivo.js** e instale o módulo correspondente como vimos na aula (eles geralmente são instalados por npm `install *NOME DO MÓDULO*`, mas é uma boa prática sempre ler a própria documentação dos módulos que vamos usar).
3. Continuando com a boa prática de leitura da documentação, solicite o módulo selecionado conforme ele indica para você.

4. Verifique se funciona corretamente com alguns exemplos (nunca pararemos de insistir para que você leia a documentação).

Podemos ver que a comunidade JavaScript, e a Internet em geral, é um mundo de informações. Então vamos perceber que existem milhares de módulos Node.JS que podem nos ajudar a tornar nosso código melhor e mais poderoso.

Como última prática, sugerimos que você faça um modelo no qual colocará os módulos Node.JS que encontrar e que função eles cumprem no código, ou seja, o que esses módulos fazem e para que servem. Em seguida, compartilhe com seus colegas para que você possa ver quais módulos eles acharam que faltam e vice-versa e, por fim, discuta sua praticidade e como são fáceis ou difíceis de usar.



**Até a próxima!**