



**Certified Tech
Developer**
The Ultimate Degree

Programación Imperativa

Modules

Objetivo

Vamos a realizar una serie de prácticas que nos van a ayudar a entender la funcionalidad y practicidad que traen los módulos a nuestros programas.



Creando módulos propios

En este primer paso vamos a crear nuestro primer módulo de forma local para acceder entre archivos.

1. En un archivo .js, crear un objeto que contenga un método para mostrar por consola sólo algunas propiedades de un objeto.

Por ejemplo en el caso de:

```
{
```

```
2.   "identificador": 0,  
3.   "estaHabilitado": false,  
4.   "honorarioConsulta": 2007.68,
```



```
5.   "edad": 29,  
6.   "nombre": "Huber Wilkins",  
7.   "sexo": "male",  
8.   "email": "huberwilkins@speedbolt.com",  
9.   "telefono": "+1 (926) 409-3726",  
10.  "direccion": "197 Stuart Street, Bowmansville, South Dakota, 4723",  
11.  "especialidad": "Neumonologia",  
12.  "cantidadConsultas": 0,  
13.  "puntuacion": 0  
14. }
```

15. Modificar el archivo de manera que sea un módulo que exporte la función que creaste en el paso anterior. Recordar usar la palabra reservada correcta.

16. En otro archivo .js, **dentro de la misma carpeta de trabajo**, requerir el módulo que exportamos.

17. Verificar que la función se ejecute correctamente.



¿Te imaginas en qué casos, o programas, la creación de módulos podría resolvernó y hasta ahorrarnos grandes problemas?

Si pensaste en programas grandes de muchas líneas de código, ¡estás en lo correcto! A medida que nuestros programas crecen, y empiezan a escalar, la cantidad de líneas de código también lo hace, y eso puede resultar en un problema de lectura y entendimiento para todos los que interactúen con tu código. Los módulos vienen a fraccionar nuestros archivos para que, si bien tendríamos mayor cantidad de archivos, serían menos pesados entre sí y contendrían una menor cantidad de líneas de código, haciéndolo más fácil de entender.



Instalando módulos externos

Ahora que ya sabemos como crear, exportar y requerir nuestros propios módulos, vamos a ver cómo podríamos utilizar módulos creados por la comunidad, y que ventajas tienen:



1. Siguiendo como ejemplo lo que vimos en la clase, buscar un módulo en Internet que te resulte interesante.
2. Una vez que lo encuentres, en una nueva carpeta de trabajo, crear un archivo .js e instalar el módulo correspondiente como lo vimos en la clase —usualmente se instalan mediante `npm install *NOMBRE DEL MÓDULO*`, pero es una buena práctica siempre leer la documentación propia de los módulos que vamos a utilizar—.
3. Siguiendo con la buena práctica de leer la documentación, requerir el módulo seleccionado como la misma te lo indique.
4. Verificar que funcione correctamente con algunos ejemplos —nunca vamos a dejar de insistir en que leas la documentación—.

Podemos ver que la comunidad de JavaScript, e Internet en general, es un mundo de información, por lo cual vamos a darnos cuenta de que hay miles de módulos de Node.js que pueden ayudarnos a que nuestro código sea mejor, y más potente.

Como última práctica te proponemos que realices una planilla en la que vas a poner los módulos de Node.js que encuentres, y que función cumplen en el código, es decir, que es lo que esos módulos hacen y para qué sirven. Luego compartilo con tus compañeros para que veas qué módulos encontraron ellos que a vos te falten, y viceversa, y por último, debatan sobre la practicidad de ellos y que tan fácil o difíciles son de usar.

¡Hasta la próxima!