

## **BackEnd - Node.JS**

- 1 Mostrar HTML desde archivo
- 2 Formularios - Métodos GET y POST
- 3 NPM, el manejador de bibliotecas.
- 4 NPM Init
- 5 NPM -v
- 6 Archivo Json de configuración de NPM
- 7 Servidor en Node.Js
- 8 Concepto de Servicios Web
- 9 Consumir Web services desde JavaScript.

## Acerca de NPM

npm es el registro de software más grande del mundo. Los desarrolladores de código abierto de todos los continentes usan npm para compartir y tomar prestados paquetes, y muchas organizaciones también usan npm para administrar el desarrollo privado.

npm consta de tres componentes distintos:

- el sitio web
- la interfaz de línea de comandos (CLI)
- el registro

Utilizá el sitio web para descubrir paquetes, configurar perfiles y administrar otros aspectos de su experiencia npm. Por ejemplo, podés configurar organizaciones para administrar el acceso a paquetes públicos o privados.

La CLI se ejecuta desde una terminal y es la forma en que la mayoría de los desarrolladores interactúan con npm.

El registro es una gran base de datos pública de software JavaScript y la meta-información que la rodea.

## Utilizá npm para. . .

---

- Adaptar paquetes de código para tus aplicaciones o incorporar paquetes tal como están.
- Descargar herramientas independientes que podés usar de inmediato.
- Ejecutar paquetes sin descargar usando npx .
- Compartir código con cualquier usuario de npm, en cualquier lugar.
- Restringir el código a desarrolladores específicos.
- Crear organizaciones para coordinar el mantenimiento, la codificación y los desarrolladores de paquetes.
- Formar equipos virtuales utilizando organizaciones.
- Administrar múltiples versiones de código y dependencias de código.
- Actualizar las aplicaciones fácilmente cuando se actualice el código subyacente.

- Descubrir múltiples formas de resolver el mismo rompecabezas.
- Buscar otros desarrolladores que estén trabajando en problemas y proyectos similares.

## Empezando

---

Para comenzar con npm, podés crear una cuenta , que estará disponible en [http://www.npmjs.com/~\\*yourusername](http://www.npmjs.com/~*yourusername) \*.

Después de configurar una cuenta npm, el siguiente paso es usar la interfaz de línea de comandos (CLI) para instalar npm .

## Compartir paquetes y colaborar con otros

---

Si eligís compartir tus paquetes públicamente, no hay ningún costo. Para usar y compartir paquetes privados, debes actualizar tu cuenta. Para compartir con otros, crear organizaciones llamadas organizaciones npm , o invite a otros a trabajar con vos, de forma privada (por una tarifa) o públicamente (gratis). O podés registrarte para obtener una instancia privada de npm para tu empresa, llamada npm Enterprise , para que puedas desarrollar paquetes internamente que no se comparten públicamente.

Para obtener más información sobre npm como producto, las próximas funciones nuevas y los usos interesantes de npm, suscribite al boletín informativo en [npm-Weekly](#) o seguí a [@npmjs](#) en Twitter.

## Documentación de referencia de CLI

Si bien los comandos CLI relevantes se tratan a lo largo de esta cursada de Codo a Codo, también NPM pone a disposición su documentación del usuario, e incluye ayuda de línea de comandos, en su propia [sección de documentación y ayuda instantánea \(páginas de manual\)](#) .

# Crear una nueva cuenta de usuario en el registro público

Si no tenés una cuenta de usuario de npm, podés crear una cuenta para compartir y descargar paquetes de Javascript en el registro público.

Nota: Si estás utilizando un registro de npm Enterprise, debes iniciar sesión con tus credenciales de usuario en lugar de crear una cuenta.

## Crear una cuenta en el sitio web

---

1. Ir a la [página de registro de npm](#)
2. En el formulario de registro de usuario, escribí los campos:
  - Nombre de usuario: el nombre de usuario que se mostrará cuando publiques paquetes o interactúes con otros usuarios de npm en npmjs.com. Tu nombre de usuario debe estar en minúsculas y puede contener guiones y números.
  - Dirección de correo electrónico: tu dirección de correo electrónico pública se agregará a los metadatos de sus paquetes y será visible para cualquiera que descargue sus paquetes. También enviaremos un correo electrónico a esta cuenta cuando actualices paquetes, así como actualizaciones e información de productos ocasionales.
  - Contraseña : Tu contraseña debe cumplir con nuestras pautas de contraseña:
    - tiene más de 10 caracteres
3. no coincide o contiene significativamente su nombre de usuario, por ejemplo, no use 'username123'
4. no es una contraseña de uso común

## Sign Up

**Username**

monatheoctocat

**Email Address**

monatheoctocat@example.com

**Password**

.....

5. Lee el Acuerdo de licencia de usuario final y la Política de privacidad e indique que estás de acuerdo con ellos.

☒ Agree to the [End User License Agreement](#) and the [Privacy Policy](#).

6. Hacé clic en Crear una cuenta .

Create an Account



Nota: Después de registrarte para obtener una cuenta npm, recibirá un correo electrónico de verificación de cuenta. Debe verificar su dirección de correo electrónico para publicar paquetes en el registro.

## Probando su nueva cuenta con npm login

Utilizá el comando `npm login` para probar el inicio de sesión en su nueva cuenta.

Nota: Si escribís mal el nombre de usuario de tu cuenta existente cuando inicias sesión con el comando `npm login`, crearás una nueva cuenta con el nombre mal escrito. Para obtener ayuda con las cuentas creadas accidentalmente, comunícate con el Soporte de npm .

1. En la línea de comando, escribí el siguiente comando:

```
npm init sesion
```

2. Cuando se te solicite, ingrese su nombre de usuario, contraseña y dirección de correo electrónico.
3. Si tenés habilitada la [autenticación de dos factores](#) , cuando se te solicite, ingrese una contraseña de un solo uso.
4. Para comprobar que has iniciado sesión correctamente, escriba:

```
npm whoami
```

Debería mostrarte tu nombre de usuario npm.

# ¿Que es el registro público de npm?

El registro público de npm es una base de datos de paquetes de JavaScript, cada uno compuesto por software y metadatos. Los desarrolladores de código abierto y los desarrolladores de las empresas utilizan el registro npm para contribuir con paquetes a toda la comunidad o miembros de sus organizaciones, y descargar paquetes para utilizarlos en sus propios proyectos.

npm es el Node Package Manager que viene incluido y ayuda a cada desarrollo asociado a Node. Durante años, Node ha sido ampliamente utilizado por los desarrolladores de JavaScript para compartir herramientas, instalar varios módulos y administrar sus dependencias. Es por eso que es fundamental que las personas que trabajan con Node.js entiendan qué es npm.

Esta herramienta funciona de dos formas:

- Como un repositorio ampliamente utilizado para la publicación de proyectos Node.js de código abierto. Lo que significa que es una plataforma en línea donde cualquiera puede publicar y compartir herramientas escritas en JavaScript.
- Como una herramienta de línea de comandos que ayuda a interactuar con plataformas en línea, como navegadores y servidores. Esto ayuda a instalar y desinstalar paquetes, gestión de versiones y gestión de dependencias necesarias para ejecutar un proyecto.

Para usarlo, tenés que instalar node.js como te mostré en la unidad anterior de codo a codo Desarrollo Web, ya que están desarrollados y se instalan de forma agrupada.

# Acerca de paquetes y módulos

El registro npm contiene paquetes, muchos de los cuales también son módulos de nodo o contienen módulos de nodo.

## Acerca de los paquetes

---

Un paquete es un archivo o directorio descrito por un archivo `package.json`. Un paquete debe contener un archivo `package.json` para que se publique en el registro de npm.

Los paquetes pueden ser sin ámbito o con ámbito para un usuario u organización, y los paquetes con ámbito pueden ser privados o públicos.

## Formatos de paquete

Un paquete es cualquiera de los siguientes:

- a) Una carpeta que contiene un programa descrito por un archivo `package.json`.
- b) Un tarball comprimido con gzip que contiene (a).
- c) Una URL que se resuelve en (b).
- d) A `<name>@<version>` que se publica en el registro con (c).
- e) A `<name>@<tag>` que apunta a (d).
- f) A `<name>` que tiene una etiqueta `latest` que satisface (e).
- g) Una URL `git` que, cuando se clona, da como resultado (a).

## Formatos de URL de git del paquete npm

Las URL de Git utilizadas para los paquetes npm se pueden formatear de las siguientes formas:

- `git://github.com/user/project.git#commit-ish`
- `git+ssh://user@hostname:project.git#commit-ish`
- `git+http://user@hostname/project/blah.git#commit-ish`
- `git+https://user@hostname/project/blah.git#commit-ish`



El `commit-ish` puede ser cualquier etiqueta, sha, o una rama que se puede suministrar como un argumento a `git checkout`. El valor predeterminado `commit-ish` es `master`.

## Acerca de los módulos

---

Un módulo es cualquier archivo o directorio en el directorio `node_modules` que la función `require()` Node.js puede cargar .

Para que la función `require()` Node.js lo cargue , un módulo debe ser uno de los siguientes:

- Una carpeta con un archivo `package.json` que contiene un campo `"main"`.
- Un archivo JavaScript.

Nota: Dado que no se requiere que los módulos tengan un archivo

```
package.json
```

, no todos los módulos son paquetes. Solo los módulos que tienen un

```
package.json
```

El archivo también es parte del paquete.

En el contexto de un programa de nodo, `module` también es lo que se cargó desde un archivo. Por ejemplo, en el siguiente programa:

```
var req = require ('solicitud')
```

podríamos decir que "La variable se refiere `req` al módulo `request`".

# Acercas de los alcances

Nota: Se debe utilizar npm versión 2 o superior para utilizar ámbitos. Para actualizar a la última versión de npm, en la línea de comando, ejecutá

```
npm install npm @ latest -g
```

Cuando te registrás para obtener una cuenta de usuario de npm o creas una organización, se te otorga un alcance que coincide con el nombre de tu usuario u organización.

Un alcance te permite crear un paquete con el mismo nombre que un paquete creado por otro usuario u organización sin conflicto.

Cuando se enumeran como dependientes en un archivo `package.json`, los paquetes con ámbito están precedidos por su nombre de ámbito. El nombre del alcance es todo lo que se encuentra entre la barra `@` y la barra:

- Alcance "npm":

```
@ npm / nombre-paquete
```

- Alcance "npmcorp":

```
@ npmcorp / nombre-paquete
```

## Visibilidad de alcances y paquetes

- Los paquetes sin ámbito son siempre públicos.
- Los paquetes privados siempre tienen un alcance.
- Los paquetes con ámbito son privados de forma predeterminada; debe pasar una marca de línea de comandos al publicar para hacerlos públicos.

# Crear y publicar paquetes privados

Para compartir tu código con un conjunto limitado de usuarios o equipos, podés publicar paquetes privados con ámbito de usuario o con ámbito de organización en el registro de npm.

**Nota:** Antes de poder publicar paquetes npm privados con ámbito de usuario, debes registrarte para obtener una cuenta de usuario npm pagada.

## Creando un paquete privado

1. Si estás utilizando npmrc para administrar cuentas en múltiples registros, en la línea de comando, cambiá al perfil apropiado:

```
npmrc <nombre-perfil>
```

2. En la línea de comando, creá un directorio para su paquete:

```
mkdir mi-paquete-de-prueba
```

3. Navegá hasta el directorio raíz de su paquete:

```
cd mi-paquete-de-prueba
```

4. Si está utilizando git para administrar el código de tu paquete, en el directorio raíz del paquete, ejecute los siguientes comandos, reemplazándolos con la URL remota `git-remote-url` de git para tu paquete:

```
git init
```

```
git remote agregar origen git: // git-remote-url
```

5. En el directorio raíz del paquete, ejecutá el comando `npm init` y pase el alcance a la `scope` bandera:

- Para un paquete en el ámbito de la organización:

```
npm init --scope = @ my-org
```

- Para un paquete de ámbito de usuario:

```
npm init --scope = @ mi-nombre de usuario
```

## Revisar el contenido del paquete en busca de información sensible o innecesaria

La publicación de información confidencial en el registro puede dañar a tus usuarios, comprometer su infraestructura de desarrollo, ser costosa de arreglar y ponerlo en riesgo de acciones legales. Recomendamos encarecidamente eliminar la información confidencial, como claves privadas, contraseñas, [información de identificación personal] [pii] (PII) y datos de tarjetas de crédito antes de publicar su paquete en el registro. Incluso si su paquete es privado, la información confidencial puede quedar expuesta si el paquete se hace público o se descarga en una computadora a la que puedan acceder más usuarios de los previstos.

## Probando un paquete

Para reducir las posibilidades de publicar errores, recomendamos probar tu paquete antes de publicarlo en el registro de npm. Para probar su paquete, ejecute `npm install` con la ruta completa al directorio de su paquete:

```
npm instalar mi paquete
```

## Publicar paquetes privados

De forma predeterminada, los paquetes con ámbito se publican con visibilidad privada.

1. En la línea de comando, navegue hasta el directorio raíz de tu paquete.

```
cd / ruta / a / paquete
```

2. Para publicar tu paquete privado en el registro npm, ejecutá:

```
npm publicar
```

3. Para ver la página de tu paquete privado, visitá [https://npmjs.com/package/\\*package-name](https://npmjs.com/package/*package-name) , reemplazando package-name \* con el nombre de tu paquete. Los paquetes privados dirán `private` debajo del nombre del paquete en el sitio web de npm.

**@ne-test-org/hello-world**

1.0.0 • **Private** • Published 19 days ago