



Node y el gestor de paquetes

NPM

Utilizar el gestor NPM para la administración de dependencia y paquetes reconociendo el rol que tiene dentro del ciclo de vida de un proyecto Node.js

- Unidad 1:
Introducción a Node
- Unidad 2:
Node y el gestor de paquetes
- Unidad 3:
Persistencia



Te encuentras aquí



¿Qué aprenderás en esta sesión?

- *Describir el rol del gestor de paquetes NPM en el ciclo de vida de un proyecto Node.*
- *Utilizar el gestor NPM para la instalación de paquetes en el entorno Node configurando sus versiones.*

¿Has oído hablar de
NPM antes? Si es así,
¿qué crees que es?



`/* ¿Qué es NPM? */`

¿Qué es NPM?

Node Package Manager

Es el sistema gestor de dependencias, paquetes o librerías predeterminado para proyectos que utilizan como base a Node y JavaScript.

Pero... ¿Qué es el gestor de dependencias?

Es una herramienta que gestiona nuestros proyectos, los paquetes o librerías que estemos utilizando.

{desafío}
latam_



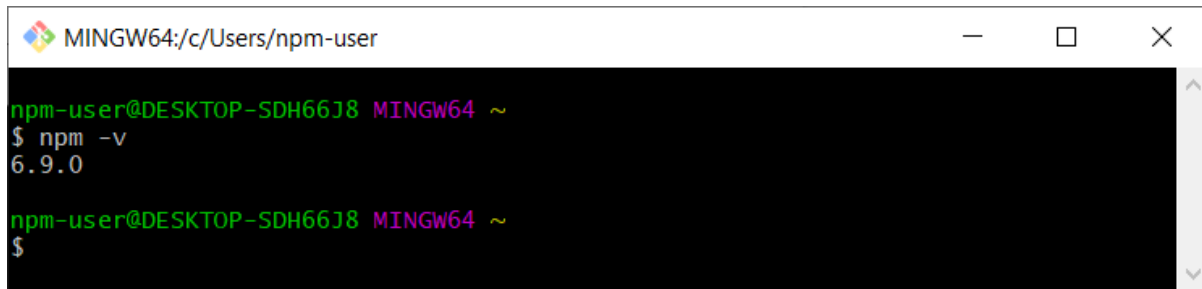
¿Qué es NPM?

Node Package Manager

- Podemos revisar qué versión de NPM tenemos instalado ejecutando la siguiente instrucción en un terminal de comandos (como Bash, CMD, Powershell, etc):

```
npm -v
```

- Presionando enter, la instrucción anterior arrojará el número de versión instalada, como muestra la siguiente imagen:



```
MINGW64:/c/Users/npm-user  
npm-user@DESKTOP-SDH66J8 MINGW64 ~  
$ npm -v  
6.9.0  
npm-user@DESKTOP-SDH66J8 MINGW64 ~  
$
```

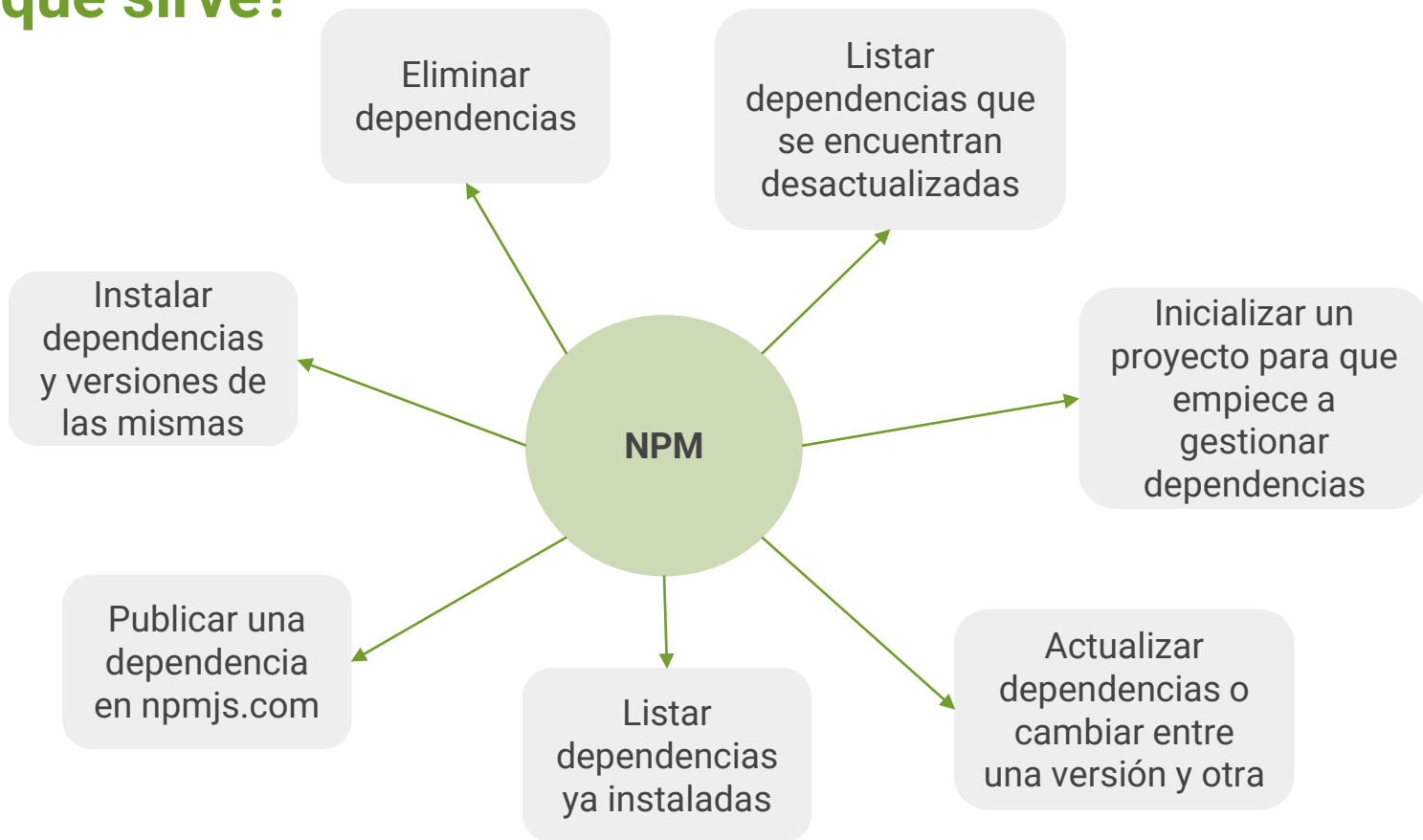
¿Qué es NPM?

Node Package Manager

- NPM también se le conoce como una plataforma de alojamiento en línea de paquetes, aplicaciones y/o proyectos Node de carácter open source y privado.
- Este repositorio de código se puede encontrar bajo la dirección <https://www.npmjs.com/> y alberga una cantidad importante de paquetes que aumentan día a día.
- Cada uno de los proyectos publicados en **npmjs.com** puede ser descargado e instalado en un proyecto local mediante la interfaz de comandos con la utilidad de NPM local.

`/* ¿Para qué sirve NPM? */`

¿Para qué sirve?



**/* Paquetes NPM que
no procesan datos */**

Chalk

¿Alguna vez has visto una terminal con colores? Pues Chalk probablemente fue el paquete usado para eso. Su API es bastante clara y directa, además de permitir concatenaciones de métodos y propiedades para construir toda la estética en una sola línea

Para instalar Chalk deberás ocupar el siguiente comando:

```
npm i chalk
```

```
bold dim italic underline inverse strikethrough black  
red green yellow blue magenta cyan white gray bgBlack  
bgRed bgGreen bgYellow bgBlue bgMagenta bgCyan bgWhite
```

Demostración

"Aplicando el paquete chalk"



Ejercicio guiado

Aplicando en paquete chalk

Se requiere implementar el paquete Chalk en una aplicación que imprima un mensaje por consola de color azul. Realiza los siguientes pasos para la creación de una mini aplicación que imprima un “Hola Mundo!” con el paquete Chalk:

- **Paso 1:** Importa el paquete Chalk en una constante.

```
const chalk = require('chalk')  
// Para versiones más recientes se debe importar chalk de la  
siguiente forma:  
import chalk from 'chalk'
```



Ejercicio guiado

Aplicando en paquete chalk

- **Paso 2:** Envía por consola un “Hola Mundo!” pero este texto deberá ser el argumento del método “blue” del objeto importado.

```
console.log(chalk.blue('Hola Mundo!'))
```

- **Paso 3:** Correr la aplicación por la terminal y deberás recibir lo que se muestra en la siguiente imagen:

```
→ ejercicio lectura git:(master) x node index.js  
Hola Mundo!
```

Ese método “blue”, también puede ser usado como propiedad y ser concatenada con otras propiedades. Sustituye el console.log escrito anteriormente por la siguiente línea de código.

```
console.log(chalk.blue.bgRed.bold('Hola Mundo!'))
```

Si vuelves a correr el archivo index.js obtendrás lo que se muestra en la siguiente imagen

```
→ ejercicio lectura git:(master) x node index.js  
Hola Mundo!
```


UUID

El UUID es un paquete de NPM que nos permite crear identificadores únicos universales, ¿Y como se ven estos identificadores? Son cadenas de texto conformadas por 32 dígitos divididos en cinco grupos que obedecen a la siguiente forma: 8-4-4-4-12. Es decir, 36 caracteres y cuatro guiones que los separan.

Para instalar UUID deberás ocupar el siguiente comando:

```
npm i uuid
```

UUID Version-1 Record Layout

123e4567-e89b-12d3-a456-426655440000
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

- Low Time
- Mid Time
- High time and version
- Clock sequence and variant
- Node

Fuente: [uuidtools](https://uuidtools.com/)

Demostración "Aplicando el paquete UUID"



Ejercicio guiado

Aplicando el paquete UUID

Realizar una pequeña aplicación que utilice UUID para enviar por consola un identificador único universal. Para esto sigue los siguientes pasos:

- **Paso 1:** Importar el paquete UUID en una constante especificando que usaremos la versión 4.

```
const { v4: uuidv4 } = require('uuid')
```



Ejercicio guiado

Aplicando el paquete UUID

- **Paso 2:** Enviar por consola la llamada al método uuidv4.

```
console.log(uuidv4())
```

Ahora corre la aplicación 3 veces por la terminal y deberás recibir lo que se muestra en la siguiente imagen.

```
→ ejercicio lectura git:(master) x node index.js  
6f2471bb-8abf-419c-8765-57afca70a4e1  
→ ejercicio lectura git:(master) x node index.js  
30c7b0f7-5051-4a9d-80ae-545ab16e82c6  
→ ejercicio lectura git:(master) x node index.js  
affad006-b066-4764-afb2-e65db5e67590
```



Ejercicio guiado

Aplicando el paquete UUID

No obstante, no estamos obligados a usar los 36 dígitos, podríamos ocupar el método **splice** o el **slice** para tomar solo una parte de estos. En el siguiente código se muestra cómo usar solo una parte de estos identificadores:

```
"670b9562-b30d-52d5-b827-655787665500".slice(0,6)
```

Esto devolvería el String que te mostramos a continuación:

```
"670b95"
```



Si necesitas repasar los métodos para recortar cadenas de texto y arreglos te dejo las documentaciones de [splice](#) y [slice](#).



Resumen

- NPM es una herramienta que administra paquetes y librerías en proyectos, asegurando fácil control y automatización.
- NPM ofrece una variedad de comandos para tareas como instalación, actualización y eliminación de paquetes, así como para administrar proyectos y compartir repositorios.
- NPM también es una plataforma de alojamiento en línea para paquetes, aplicaciones y proyectos Node open source y privados.
- Se presentan ejemplos de uso con dos paquetes. El paquete "Chalk" agrega colores a la terminal. El paquete "UUID" genera identificadores únicos.

¿Cuál es el propósito principal de NPM y cómo puede facilitar la gestión de dependencias en proyectos de desarrollo de software?





Próxima sesión...

- *Utilizar el gestor NPM para la instalación de paquetes en el entorno Node configurando sus versiones.*
- *Utilizar un módulo dentro de un programa Node mediante sentencias de importación para el uso de sus funcionalidades.*

{desafío}
latam_

*Academia de
talentos digitales*

