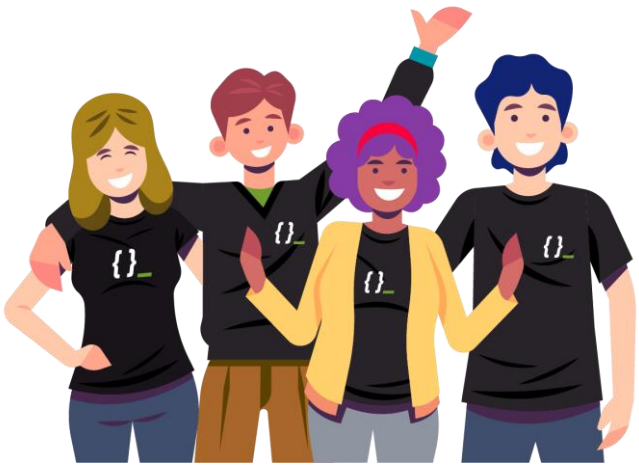
Conociendo paquetes en Node



¡Comencemos!

¿Qué aprenderás?

- A identificar los paquetes más utilizados en las aplicaciones hechas con Node para solucionar un problema planteado.
- A implementar el paquete nodemon para la automatización del levantamiento de aplicaciones Node.

Introducción

El desarrollo puro puede ser beneficioso en cuanto a rendimiento se refiere. No obstante, instalar dependencias y módulos convierten nuestro proceso de desarrollo, más ágil y óptimo, por todo el tiempo que nos ahorramos al no intentar reinventar la rueda.

Muchos de estos paquetes que usamos en el desarrollo son creados por la misma comunidad de desarrolladores a nivel mundial, personas u organizaciones que han dedicado una importante cantidad de tiempo en desarrollar una solución a un problema planteado y que disponen su código fuente de forma gratuita en NPM.

En este capítulo, identificarás varios paquetes conocidos de NPM y sus utilidades para conseguir resultados ideales en menos tiempo de desarrollo.

Una breve mirada a los paquetes más populares.

A continuación se presenta una lista con una breve descripción de los paquetes más utilizados en los proyectos para Node y que podemos gestionarlos a través de NPM.

- **Express:** Es un framework para aplicaciones web de servidor, famoso por la creación de infraestructuras web rápidas, minimalistas y flexibles. Su popularidad crece y se ha convertido en un estándar para el desarrollo Backend en Node.

- **Nodemon:** Es una librería de gran utilidad para el desarrollo, ya que permite ir reflejando los cambios que vamos haciendo en nuestro código a medida que vamos guardando nuestro archivo. Cada vez que guardamos nuestro código, el mismo se compila inmediatamente, lo cual evita que por cada cambio realizado tengas que volver a ejecutar las líneas de código correspondientes en la consola.

- **Yargs:** En nuestros desarrollos, hay ocasiones en que necesitaremos crear algún script que ejecute alguna tarea en específico, como por ejemplo, leer un archivo, automatizar una tarea, etc. Normalmente, estos script recibirán parámetros y se comportan de acuerdo a estos valores enviados. Yargs nos ayudará a formatear a objetos los parámetros que se envíen al ejecutar un script por consola.

- **Moment:** Es una librería que nos permitirá la manipulación de fechas y horas en nuestros desarrollos, trabajarlas en distintos formatos, acceder a la fecha y hora actual, realizar sumas, restas, comparaciones entre fechas, etc. Una gran gama de funcionalidades que todo desarrollo requiere en poca o gran medida.

- **Socket.io:** Libreria basada en WebSocket, que permite la comunicación bidireccional entre Cliente y Servidor. Con esto podrás realizar tareas en el servidor, por ejemplo, avisar a una aplicación de correos la llegada de un correo nuevo, o a una aplicación de chat indicarle que hay un mensaje nuevo por revisar.

- **Mocha:** Mocha es un framework de pruebas o test que nos entrega muchas características para la evaluación de código. Mocha puede ser implementado en Node o por medio de un navegador. Realiza sus pruebas en serie generando reportes flexibles y exactos. Es muy simple, flexible y divertido en su uso.

- **Underscore:** Es una librería de JavaScript que nos provee de un conjunto de métodos y funcionalidades que se pueden clasificar en manejo de colecciones, arreglos, funciones y objetos. Permite trabajar con funcionalidades avanzadas de búsqueda y filtros en arreglos.

- **Lodash:** Este paquete nació como una bifurcación de Underscore por lo que está compuesto en gran parte por las mismas propiedades y métodos. No obstante, es demandado hoy en día mucho en el mercado laboral por sus optimizaciones de rendimiento.

- **Morgan:** Es un middleware de registro de sucesos o conocidos como logger, que nos permite escribir en la consola peticiones, errores o lo que necesitemos mostrar o monitorear. Es una gran herramienta para utilizarla en el registro de errores, tanto para el desarrollo como para su funcionamiento productivo.

- **Jimp:** Es una biblioteca de procesamiento de imágenes para Node escrita completamente en JavaScript sin dependencias nativas. Su API es muy cómoda de ocupar.

- **Nodemailer:** Nodemailer es un módulo para aplicaciones Node que permite enviar correos electrónicos de forma sencilla. El proyecto comenzó en 2010 cuando no había una opción sensata para enviar mensajes de correo electrónico. Hoy es la solución a la que recurren la mayoría de los usuarios de Node de forma predeterminada.

- **Axios:** Preferida por la mayoría de los programadores. Es una librería para consultas de API REST que te permite hacer peticiones bajo el protocolo HTTP especificando los verbos en forma de métodos, obteniendo en su ejecución una promesa que devuelve como parámetro la respuestas de la consulta.

- **Chalk:** Es un paquete de NPM que ofrece funciones para colorear y personalizar la estética de los mensajes por consola. Contiene una amplia gama de colores y formas de estilizar los mensajes y de esta manera hacer más intuitiva y cómoda la lectura de las reacciones que puedan tener nuestras aplicaciones.

- **UUID:** Por sus siglas en inglés Universally Unique Identifier (Identificador único universal). Este paquete es muy usado cuando necesitamos generar un campo identificador de un producto o recurso, su objetivo es generar identificadores para reconocer el objeto dentro de un sistema.

Todas estas dependencias puedes usarlas dentro de tus desarrollos y lo interesante sucede cuando mezclas sus poderes con los módulos integrados de Node.

Nodemon

Este paquete no puede faltar en el proceso de desarrollo de servidores con Node. Su objetivo es automatizar el levantamiento del servidor cada vez que salvas cambios en el código. De esta manera, no tendrás que preocuparte por cancelar la terminal dando de baja el servidor por cada nueva instrucción o cambio escrito.

Para instalar nodemon deberás ocupar el siguiente comando:

```
npm i nodemon
```

Este paquete tiene la peculiaridad de que no necesita ser importado, simplemente con el hecho de ser instalado podrás ocuparlo como un comando en la terminal. ¿Es posible usar una dependencia como un comando? Claro que sí, te lo muestro con el siguiente ejercicio:

Ejercicio guiado: ups, ¡me equivoqué!

Desarrollar una aplicación que imprima por consola un mensaje. La idea será voluntariamente escribir el mensaje que queremos enviar a la consola, al corregirlo y guardar cambios, notar que el servidor fue levantado de nuevo automáticamente arrojando por consola el mensaje corregido

En este ejercicio no serán necesario los pasos porque solo debes tener la siguiente línea de código escrita en tu index.js:

```
console.log('Desafio Laram')
```

Ahora abre la terminal y escribe entonces el siguiente comando:

```
nodemon index.js
```

Deberás obtener en la misma terminal lo que te mostramos en la siguiente imagen:

```
→  ejercicio lectura git:(master) ✗ nodemon index.js
[nodemon] 1.19.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Desafio Laram
[nodemon] clean exit - waiting for changes before restart
```

Imagen 1. Respuesta de nodemon al levantar la aplicación.
Fuente: Desafío Latam

Ahora la aplicación está preparada para volver a compilarse cada vez que se guarde un cambio en el código. Para esto cambia la letra "r" por una "t" en la palabra Laram y guarda el cambio. Deberás recibir lo que te mostramos en la siguiente imagen por la terminal:

```
→  ejercicio lectura git:(master) ✗ nodemon index.js
[nodemon] 1.19.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Desafio Laram
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
Desafio Latam
[nodemon] clean exit - waiting for changes before restart
```

Imagen 2. Respuesta de nodemon al salvar el documento editado.
Fuente: Desafío Latam

Como puedes notar, la aplicación se reseteó inmediatamente y ahora estamos recibiendo por consola el mensaje corregido.