

# NPM 9 JEST

Naseddin Oufallah Jonathan Echeverria

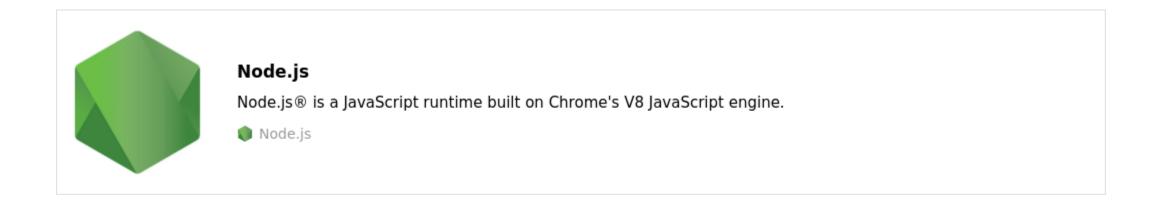






# Instalacion de Node.js

Descargar desde la pagina oficial de Node el ejecutable para instalarlo



node --version

npm --version

# ¿Que es Node.js?

**Node.js** es un entorno de ejecución para JavaScript construido con **V8**, motor de JavaScript de Chrome.

Node.js permite a los desarrolladores crear aplicaciones de **front-end y back-end** utilizando JavaScript. Fue lanzado en **2009 por Ryan Dahl.** 

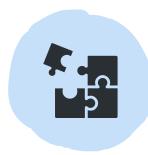




# "Node.js es un entorno de ejecución de JavaScript multiplataforma y de código abierto".



**Node.js es un entorno de ejecución de JavaScript:** cuando escribes código JavaScript en tu editor de texto, ese código no puede realizar ninguna tarea a menos que lo ejecutes. Y para ejecutar tu código, necesitas un entorno de ejecución Javascript.



**Node.js es de código abierto:** esto significa que el código fuente de Node.js está disponible públicamente. Y es mantenido por colaboradores de todo el mundo.



**Node.js es multiplataforma:** Node.js no depende de ningún software de sistema operativo. Puede funcionar en <u>Linux, macOS o Windows.</u>



















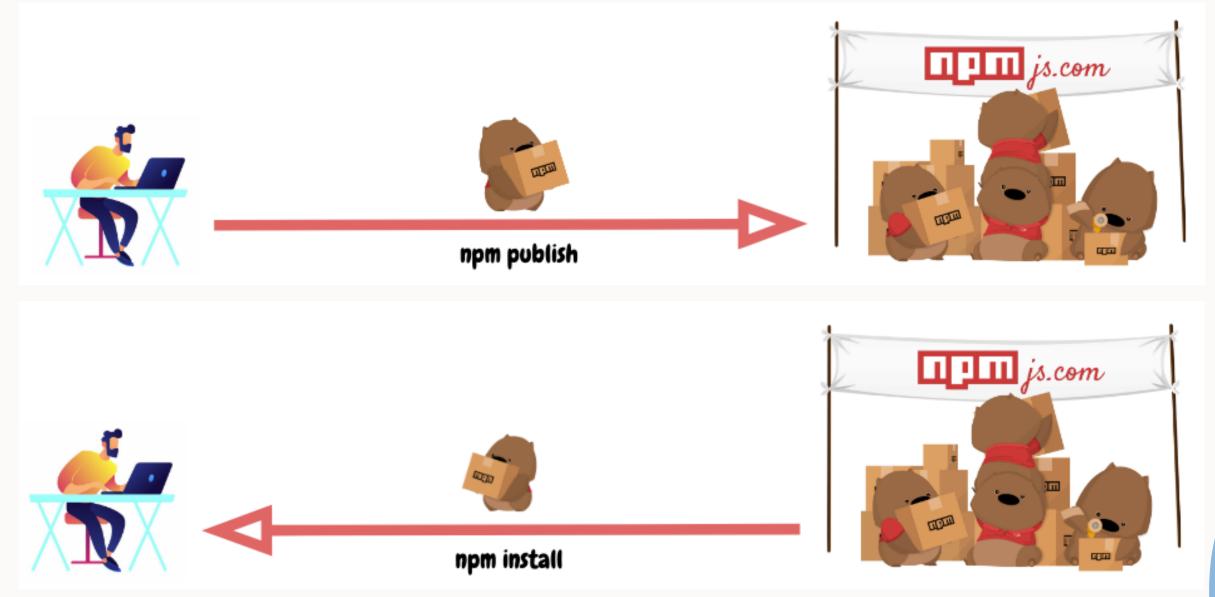
#### Node Package Manager (Administrador de paquetes de Node)

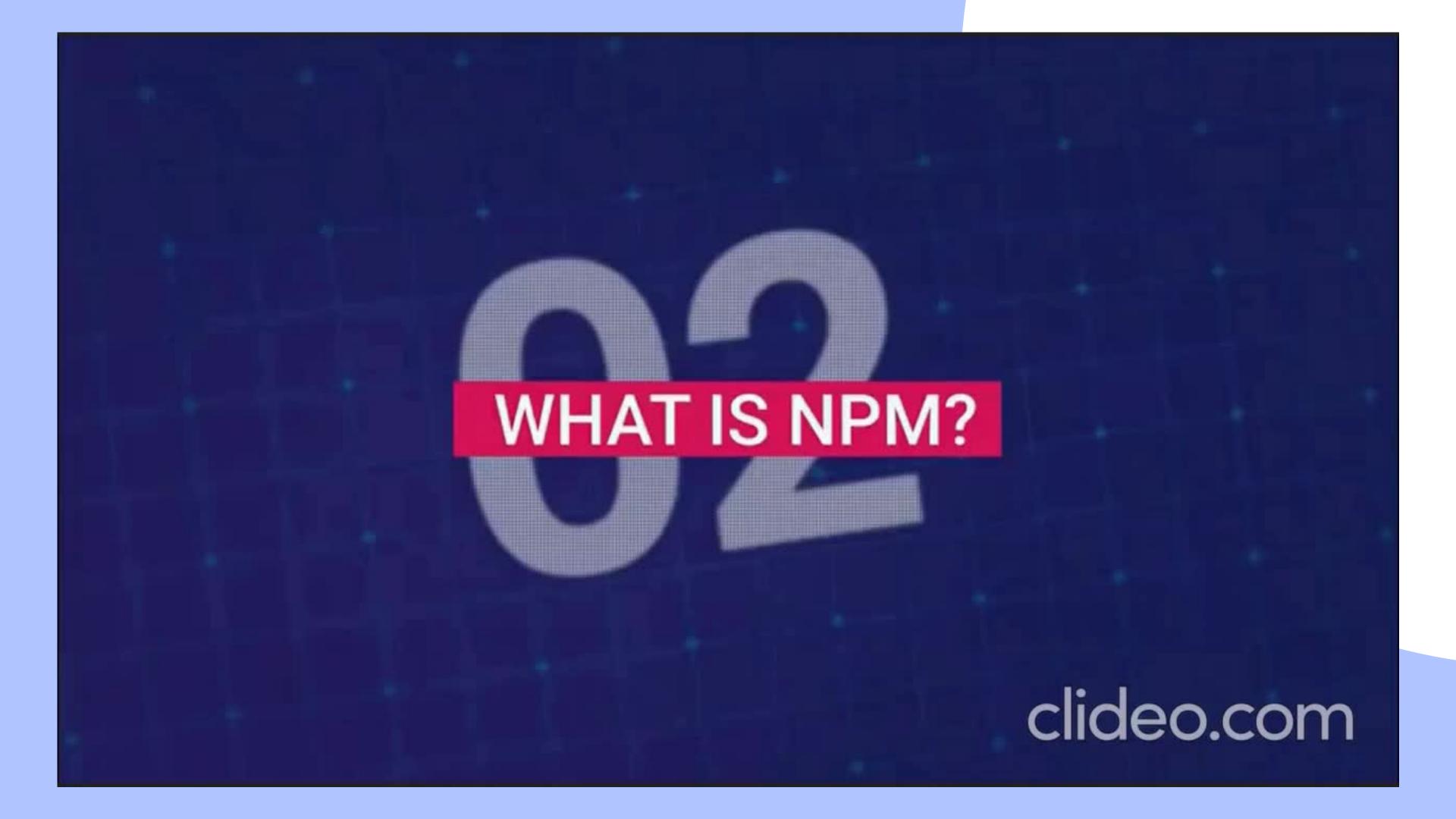
NPM es el registro de software más grande del mundo. Los desarrolladores de código abierto de todos los continentes usan npm para compartir y tomar prestados paquetes, y muchas organizaciones también usan npm para administrar el desarrollo privado.

npm, Inc. es una empresa **fundada en 2014** y fue adquirida por **GitHub en 2020**. npm es una parte fundamental de la comunidad de JavaScript y ayuda a respaldar uno de los ecosistemas de desarrolladores más grandes del mundo.

# NPM consta de tres partes:

- una herramienta CLI (interfaz de línea de comandos) para publicar y descargar paquetes
- el sitio web, un repositorio en línea que aloja paquetes de JavaScript
- el registro





# Existen distintos paquetes en NPM

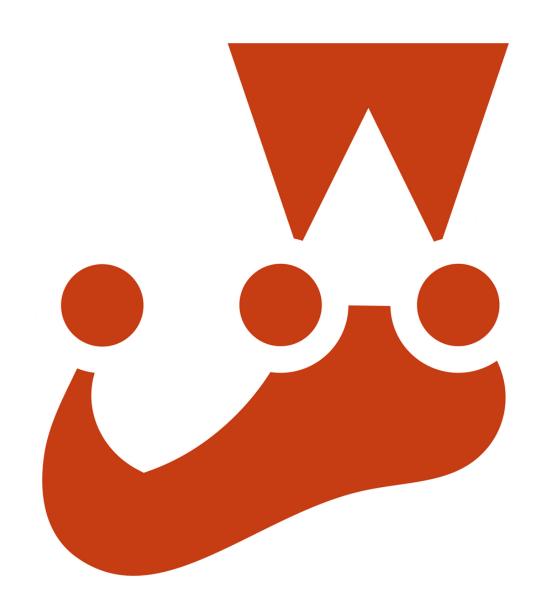


Pero nosotros nos centraremos en JEST

# ¿Qué es Jest?

Jest es un framework para realizar pruebas (test) y fue creado por Meta y mantenido por la comunidad con apoyo de Facebook. Es fácil de instalar y no requiere de una configuración muy compleja para poder añadirlo en nuestros proyectos.

Funciona con proyectos que usan: Babel, TypeScript, Node, React, Angular, Vue, Javascript y más!



# Rápido y confiable

Jest puede ejecutar pruebas en paralelo de manera confiable.

```
PASS packages/diff-sequences/src/__tests__/index.test.js
PASS packages/jest-diff/src/__tests__/diff.test.js
PASS packages/jest-mock/src/__tests__/jest_mock.test.js
PASS packages/jest-util/src/__tests__/fakeTimers.test.js
PASS packages/pretty-format/src/__tests__/prettyFormat.test.js
RUNS packages/jest-haste-map/src/__tests__/index.test.js
RUNS packages/pretty-format/src/__tests__/DOMElement.test.js
RUNS packages/jest-config/src/__tests__/normalize.test.js
RUNS packages/expect/src/__tests__/matchers.test.js
RUNS packages/pretty-format/src/__tests__/Immutable.test.js
RUNS packages/expect/src/__tests__/spyMatchers.test.js
RUNS packages/jest-cli/src/__tests__/SearchSource.test.js
RUNS packages/jest-runtime/src/__tests__/script_transformer.test.js
RUNS packages/jest-cli/src/__tests__/watch.test.js
RUNS packages/jest-haste-map/src/crawlers/__tests__/watchman.test.js
RUNS packages/pretty-format/src/__tests__/react.test.js
Test Suites: 5 passed, 5 of 303 total
            332 passed, 332 total
Tests:
Snapshots: 21 passed, 21 total
            4s
Time:
```

# Code coverage

Genere cobertura de código agregando la marca --coverage. Jest puede recopilar información de cobertura de código de proyectos completos, incluidos archivos no probados.



### Detallado

Cuando las pruebas fallan Jest proporciona un contexto con detalles de la falla.

#### toBe

```
    toBe multi-line strings
    expect(received).toBe(expected) // Object.is equality
    Expected - 1
    + Received + 1
    Roses are red. Violets are blue.
    - Testing with Jest is good for you.
    + Testing your luck is bad for you.
```

#### toBeCloseTo

toBeCloseTo floating-point numbers
 expect(received).toBeCloseTo(expected)
 Expected: 3.141592653589793
 Received: 3.1
 Expected precision: 2
 Expected difference: < 0.005
 Received difference: 0.04159265358979303</li>

```
FAIL tests/q.test.js
 \times adds 1 + 2 to equal 3 (4ms)

    adds 1 + 2 to equal 3

   TypeError: sum is not a function
     2
         test('adds 1 + 2 to equal 3', () => {
           const result = sum(1, 3);
           expect(result).toBe(3);
     5
         });
     at Object.sum (tests/q.test.js:4:18)
```

# ¿Qué es testing en Programación?

El testing en programación es el proceso de verificar y validar el software para detectar errores y evaluar su calidad.

Se realiza mediante la ejecución de pruebas específicas que buscan identificar fallos y garantizar el correcto funcionamiento del programa. El objetivo final del testing es mejorar la confiabilidad y la eficiencia del software.

# Importancia de la fase de testing

- Nos permite identificar los errores que han ocurrido en la fase del desarrollo.
- Tener una buena calidad en el software desarrollado.
- Minimizamos el mantenimiento y los coste asociados cuando el desarrollo esta muy avanzado
- Garantizamos que el software es fiable.

# Tipos de pruebas

- Unit Testing: son pruebas unitarias y este tipo de testing consiste en probar de forma individual las funciones y/o métodos del código.
- Componente: Verifica que un componente se monte, se visualice adecuadamente, se pueda interactuar con él y se comporte como se esperaba.

## Matchers

Jest usa matchers para probar los casos de prueba unitaria de diferentes maneras, como hacer coincidir los valores iguales, la veracidad, los números, las cadenas, etc.

.toBe: Usado para comparar valores primitivos

.toEqual: compara recursivamente todas las propiedades de instancias de objetos

#### **Veracidad (Truthiness):**

.toBeTruthy: El valor es verdadero.

.toBeFalsy: El valor es falso.

.toBeDefined: El valor esta definido

.toBeUndefined: El valor es 'undefined'

.toBeNull: El valor es 'null'

#### Números

.toBeLessThan:

.toBeLessThanOrEqual:

.toBeGreaterThanOrEqual:

.toBeGreaterThan:

# Instalación de Jest

- 1. npm init -y (Esto creará un archivo package.json con la configuración básica).
- 2. npm install --save-dev jest (comando para instalar jest)
- 3. npm install --save-dev @babel/core @babel/cli @babel/preset-env
- 4. Para pruebas de test con jest hay que modificar package.json

```
"scripts": {
    "test": "jest --watchAll"
},
babel":{
    "presets": ["@babel/preset-env"]
},
```

# GRACIAS

#### Recursos

Jest: <a href="https://www.youtube.com/watch?v=tgWBQZNCOT0">https://www.youtube.com/watch?v=tgWBQZNCOT0</a>

Jest: <a href="https://www.youtube.com/watch?v=\_DzBez4qMi0">https://www.youtube.com/watch?v=\_DzBez4qMi0</a>

https://jestjs.io/es-ES/docs/getting-started

https://github.com/joseluisgs/testing-js-jest

NPM: <a href="https://www.youtube.com/watch?v=P3aKRdUyr0s">https://www.youtube.com/watch?v=P3aKRdUyr0s</a>