# Node.js

* ¿Qué es?
  + No es un lenguaje de programación
  + No es un framework
  + No es una web o un servidor

Es un entorno de ejecución de JavaScript, un sitio donde se puede ejecutar JS.

-Código abierto

-Multiplataforma

-Asíncrono (in, out de datos)

-Arquitectura orientada a eventos

Utiliza un V8, motor de JS que utiliza Chrome, mismo motor pero diferente entorno.

Event Loop // **ORIENTADO A EVENTOS** //

Tiene un bucle que va manejando las solicitudes que le van llegando en cada evento.

Va ejecutando tareas mientras espera respuestas de tareas en standby.

Y todo esto en un solo Thread (HILO).

***Es monohilo***, pero con webworkers se pueden tener más de un hilo.

***Resumen de Tipos de Sincronía***

* Síncrona:
* **fs.readFileSync**: Bloquea la ejecución hasta que la lectura del archivo se completa.
* Asíncrona:
* **Callbacks (fs.readFile con callback)**: Inicia la operación y continúa la ejecución, ejecutando el callback cuando la operación termina.
* **Promesas (fs.readFile con .then)**: Inicia la operación y continúa la ejecución, ejecutando el then cuando la promesa se resuelve.( ¡¡¡Varía según la velocidad de lectura!!!)
* **Promisify (promisify(fs.readFile))**: Convierte una función basada en callback a promesa, manejando la asincronía de manera similar a las promesas nativas. (¡¡¡Varía según la velocidad de lectura!!!)
* **async/await**: Sintaxis más legible para manejar promesas, esperando la resolución de la promesa de forma similar al código síncrono.
* **IIFE (Inmediatly Invoked Function Expression”Función autoinvocada”)** : Sintaxis de una función asíncrona, aunque puede ser una función corriente.
* **Promise.all**: Ejecuta múltiples promesas en paralelo y espera a que todas se resuelvan antes de continuar.