CLASE 5 –PRACTICO- 30-08-21

* **Clousure:** encapuslua un contexto y podemos utilizar ese contexto en otro momento, xq queda viviendo en ese punto del código donde lo creamos olo generamos. NO SE USAN TANTO EN EL OBLIGATORIO
* THIS: prop q se resuelve en tiempo de ejecución. En función de donde se esta llamando al this, el valor que asume. Dependiendo del contexto de donde esta corriendo, (CLOSURE) el valor que se va a mostrar del this.
* nodeJS:
  + Single thread, un call stack o una cosa a la vez
* ARQUITECTURA DE NODE JS
  + Api de node:toma esas op del callstack q no tienen q ser resueltas en ese momento y cuando esta pronto para ser ejecutadas las coloca en el event queue.
  + Event queue: recibe de la node api las tareas a ser ejecutadas
  + Callstack: cuando pasa delevent queue al callstack? Cuando el callstack este vacio
  + Event loop: se encarga de pushear cosas al event queue cuando la api de node asi lo dice
  + Reglas del event loop con la api – ver foto

A picture containing graphical user interface

Description automatically generated

<https://nodejs.org/en/docs/guides/event-loop-timers-and-nexttick/> 🡪 LEER

🡪 practico

* + - const http = require(‘http’) 🡪 http modulo core de nodejs
    - diferencia entre filesystem, modulo core/local o de la comunicada
    - nodejs modules
    - ver express npm module
    - init app nodejs : npm init -y
    - npm install <la dependenecia q quiera> ej express