

Curso de Node

- **npm init** nos permitirá iniciar un proyecto node

Instalando módulos de npm.

- **npm install “nombre del modulo”** para la instalación de un modulo en node, utilizamos nuestro gestor de paquetes npm junto con el comando install y el nombre del paquete.
- **npm uninstall “nombre del modulo”**, podremos desinstalar un módulo de nuestro proyecto esto lo eliminará también del package.json

Automatización de tareas.

- **npm audit** nos permitirá realizar una auditoria sobre el proyecto, este analizará todos los módulos que hayamos agregado al proyecto y nos dará un informe sobre las vulnerabilidades que encontró.
- **node “nombre del archivo”** nos permitirá correr un archivo como ya sabemos.
- pero que pasa cuando queremos ejecutar dos o comando al mismo tiempo sobre nuestra app?

podemos agregar los comando que queremos ejecutar al mismo tiempo en la sección de scripts del package.json:

```
"scripts": {  
  "test": "echo \"Error: no test specified\" && exit 1",  
  "auditoria": "npm audit && node app.js"  
},
```

simplemente los separamos por un && “and” y lo corremos con el comando **npm run “nombre del scrip que creamos”**

- **Buffer()** nos permite leer datos binarios
- **process** nos permite poder leer información del sistema en el que se esta ejecutando node js, como por ejemplo la ruta en la que se esta ejecutando la aplicacion con **process.path** tambien podemos saber acerca de las variables del sistema de la maquina en la que se esta ejecutando la node js, con **process.env** si las variable de sistema no existe las podemos crear

Importacion de módulos.

- **require(“nombre del modulo”)**; podemos importar un modulo externo que hallamos descargado directamente con npm install, esto lo ira a buscar a la carpeta de node_modules
- **require(“ruta especifica mas nombre del modulo sin extensión”)** cuando hemos creado un modulo propio atravez de un archivo js, lo podemos importan de la misma manera solo que en nuestro modulo creado tenemos que indicar que sera exportado atraves de: **module.exports = “nombre del modulo”**.

modulos del core de node:

util “promueve métodos utilitarios como comprobación de tipos de datos”,
http “para el manejo de archivos externos y peticiones”, path “para el manejo
de archivos locales” fs “para el manejo de rutas locales”, etc.

nodemon: nodemon es un modelo que al instalarlo nos permitirá crear un módulo de manera automática cada vez que nosotros hagamos un cambio y guardemos.

- **npm install nodemon -g** instalamos de manera global nodemon
- **nodemon “nombre del modulo”** corremos un modulo con nodemon y una vez lo hagamos ya no es necesario el tener que volver hacerlo simplemente al guardar un cambio el automáticamente correrá de vuelta el modulo

Creación de un servidor basado en node js

para la creación de un servidor basico con nodes sin ningun mdulo externo instalado, lo que procedemos hacer es:

- importamos el modulo de http **const http = require(“http”);**
- creamos una variable para el servidor con lo que nos devuelve el metodo createServer del modulo http **const servidor = http.createServer()**
- procedemos a pasar como parámetro una función anónima a ese método
http.createServer(function () {
 escribo mi logica para ejecutarse cuando se reciba la peticion
})
- finalmente le decimos a nuestra variable servidor porque puerto va a escuchar la petición mediante el metodo listen **servidor.listen(8080);**

escribir en la cabecera de respuesta de una petición.

a la función anónima que creamos en el parametro del metodo createServer le podemos pasar como parámetros el req y el response para obtener los metadatos de la petición y responder a esta.

y con esto podemos escribir la respuesta para el req que se genere:

- el método writeHead nos permitirá escribir sobre el response el tipo de respuesta que queremos brindar a través del tipo mime y el código de estatus a responder, entre otros metadatos para el response. **response.writeHead(200, {“Content-type”: “text/html”})**
- el metodo write nos permitirá escribir el contenido de la respuesta **response.write(“<h1>hello</h1>”)**