**NODE JS**

Node.js is an open source server environment.

Node.js allows you to run JavaScript on the server.

Node.js uses JavaScript on the server

**Node.js uses asynchronous programming!**

A common task for a web server can be to open a file on the server and return the content to the client.

Here is how PHP or ASP handles a file request:

1. Sends the task to the computer's file system.
2. Waits while the file system opens and reads the file.
3. Returns the content to the client.
4. Ready to handle the next request.

Here is how Node.js handles a file request:

1. Sends the task to the computer's file system.
2. Ready to handle the next request.
3. When the file system has opened and read the file, the server returns the content to the client.

Node.js eliminates the waiting, and simply continues with the next request.

Node.js runs single-threaded, non-blocking, asynchronous programming, which is very memory efficient.

What Can Node.js Do?

* Node.js can generate dynamic page content
* Node.js can create, open, read, write, delete, and close files on the server
* Node.js can collect form data
* Node.js can add, delete, modify data in your database

## What is a Node.js File?

* Node.js files contain tasks that will be executed on certain events
* A typical event is someone trying to access a port on the server
* Node.js files must be initiated on the server before having any effect
* Node.js files have extension ".js"

APUNTES RUAN: creo el Proyecto con [npm init], ejecuto un archivo node con [node nombredelarchivo.js] (ej node index.js). Si quiero que ese archivo se reinicie con cada alteración que hago automáticamente ( en vez de refreshear), puedo usar [node –watch nombredelarchivo.js]

**INTRODUCCION A PACOTES NODE.**

Modulo fs (filesystem) permite interagir con el sist de archivos del OS.

Basicamente: creamos una const fs = require(‘fs). [lo que hicimos es guardar en una variable fs ese comando require(‘fs), asi luego usaremos fs(la variable) para usar los métodos. Por ej:

const fs = require('fs')

const data = fs.readFileSync('arquivo.txt', 'utf-8')

console.log(data)

Aca lo que hicimos es ademas de usar fs.readFileSync(), ese resultado lo almacenamos en la const data y lo mostramos en un console.log

Por ej, para receber dados via terminal usamos el PACOTE:

**npm install readline-sync**

E para receber dados via terminal, usamos a seguinte sintaxe:

const readlineSync = require('readline-sync')

let nomeInserido = readlineSync.question("Qual é o seu nome?")

**Entendé que Node basicamente se maneja por MODULOS.**

## What is a Module in Node.js?

Consider modules to be the same as JavaScript libraries.

A set of functions you want to include in your application.

Node.js has a set of built-in modules which you can use without any further installation.

Look at our [Built-in Modules Reference](https://www.w3schools.com/nodejs/ref_modules.asp) for a complete list of modules.

To include a module, use the require() function with the name of the module:

var http = require('http'); //SIEMPRE ES BUENO ‘GUARDAR’ ESA function EN UNA

VARIABLE.

## Create Your Own Modules

You can create your own modules, and easily include them in your applications.

The following example creates a module that returns a date and time object:

exports.myDateTime = function () {  
  return Date();  
};

Use the exports keyword to make properties and methods available outside the module file.

## What is NPM?

NPM is a package manager for Node.js packages, or modules if you like.

[www.npmjs.com](https://www.npmjs.com/) hosts thousands of free packages to download and use.

The NPM program is installed on your computer when you install Node.js

## What is a Package?

A package in Node.js contains all the files you need for a module.

Modules are JavaScript libraries you can include in your project.

**PARA BAJAR UN PACKAGE, VAS A LA PAGINA DE NODE, BUSCAS EL QUE QUERES BAJAR Y LO INSTALAS CONpm install [nombre\_del\_package]** esto lo haces el el folder donde vas a trabajar.