**Práctico Nº 3 – Node.js**

1.- Realice una app que importe la librería os.

Averigue los métodos disponibles y muestre por consola la información del cpu, sistema operativo, cantidad de mamoria ram total y disponible de la PC donde ejecuta.

2.- Realice un programa que guarde en un archivo de texto los datos del cpu.

Note que la salida de cpus() es un json y debe pasarse a un string usando la función JSON.stringify().

Verifique el correcto funcionamiento.

3.- Realice un programa que lea el contenido de un archivo de texto y lo muestre en la consola. Al finalizar imprima el mensaje “Fin de contenido del archivo”.

4.- Considere el código que lista las películas de la API del sitio Studio Ghibli

const request = require('request');

request('https://ghibliapi.herokuapp.com/films', (error, response, body) => {

if (error) {

console.error(`Could not send request to API: ${error.message}`);

return;

}

if (response.statusCode != 200) {

console.error(`Expected status code 200 but received ${response.statusCode}.`);

return;

}

console.log('Processing our list of movies');

movies = JSON.parse(body);

movies.forEach(movie => {

console.log(`${movie['title']}, ${movie['release\_date']}`);

});

});

Ejecute el programa (recuerde instalar previamente el paquete request via NPM)

Contestar:

¿Cuáles són los párametros del callback?

¿Cual es el contenido de error y body si el request falla?

¿Dónde se encuentran almacenados los datos si el request es exitoso?

¿En que formato vienen los datos recuperados?

**Tarea**

a.- Modifique el programa para guardar las películas recuperadas en un archivo callbackMovies.csv.

b.- Modifique el programa para que este guarde en un archivo callbackMovies2011.cvs las películas del año 2011 (Nota: no elimine la operatoria de guardar todas las películas)

c.- Modifique el programa para que este guarde en un archivo callbackMoviesLocations.json los datos de las películas liberadas después del 2000 junto con los lugares donde fueron grababas las diferentes escenas. (Solo nombre de las películas y nombres de los lugares).

d.- Modifique el programa para que este sea realizado usando promesas.

e.- Modifique el programa para que este sea realizado utilizando async/await

5.- Desarrolle el módulo misUtilidades.js que exporte una variable y dos funciones.

Luego desarrolle una app que requiera y use las funciones exportadas.

6.- Averigue las funcionabilidades de la librería lodash.

Cree una app que inicialice una lista de objetos con algunas propiedades.

Ej:

Const listaCosas = [

{nombre:”mesa”, altura:”150”, peso:”23”},

{nombre:”silla”, altura:”130”, peso:”4”},

{nombre:”heladera”, altura:”180”, peso:”45”}

]

Utilice algunas de las utilidades de la librería para hacer operaciones con objetos.