	Univ. Robin Largo Orcko	Materia: Seminario de Sistemas	N° Prac: 1
	Ci: 8574898	Sigla: SIS-719	Nota:
	Fecha de entrega: 6/4/2022	Aux. Olmedo Edson	
	Docente: Ing. Ditmar Castro Angulo		

TAREA 1

1.- ¿Que es NPM ?

De sus siglas NPM (Node Package Manager) es un gestor de paquetes desarrollado en su totalidad bajo el lenguaje JavaScript por Isaac Schlueter, a través del cual podemos obtener cualquier librería con tan solo una sencilla línea de código, lo cual nos permitirá agregar dependencias de forma simple, distribuir paquetes y administrar eficazmente tanto los módulos como el proyecto a desarrollar en general.
Esta herramienta funciona de dos formas:

- Como un repositorio ampliamente utilizado para la publicación de proyectos Node.js de código abierto. Lo que significa que es una plataforma en línea donde cualquiera puede publicar y compartir herramientas escritas en JavaScript.
- Como una herramienta de línea de comandos que ayuda a interactuar con plataformas en línea, como navegadores y servidores. Esto ayuda a instalar y desinstalar paquetes, gestión de versiones y gestión de dependencias necesarias para ejecutar un proyecto.

2.- ¿Para que sirve el comando npm init?

Este comando funciona como una herramienta para crear el archivo package.json de un proyecto. Una vez que ejecutes los pasos de npm init, se generará un archivo package.json y se guardará en el directorio actual.

3.- Si ejecutamos el comando npm init -y, que diferencia tiene frente al npm init

El indicador -y cuando se pasa a los comandos **npm** le dice al generador que use los valores predeterminados en lugar de hacer preguntas. Ejemplo:

npm inicializar -y

Simplemente generará un proyecto npm vacío sin pasar por un proceso interactivo.

La -y significa sí, mientras que npm init

4.- **Que es el http://registry.npmjs.org/**

Es un sitio web de registro que implementa la especificación del Registro de Paquetes CommonJS para leer la información de los paquetes que usa npm para resolver los paquetes por nombre y versión.

npm está configurado para usar el registro público de npm, Inc. en https://registry.npmjs.org de forma predeterminada. El uso del registro público de npm está sujeto a los términos de uso disponibles en https://www.npmjs.com/policies/terms

Puede configurar npm para utilizar cualquier registro compatible que desee,e incluso ejecutar su propio registro.El uso del registro de otra persona puede estar regido por sus condiciones de uso.

La implementación del registro de paquetes de npm soporta también varias APIs de escritura, para permitir la publicación de paquetes y la gestión de la información de las cuentas de los usuarios.

El registro público de npm funciona con una base de datos CouchDB, de la cual hay un espejo público en https://skimdb.npmjs.com/registry. El código para la couchapp está disponible en https://github.com/npm/npm-registry-couchapp.

La URL de registro utilizada está determinada por el alcance del paquete (consulte el scope . Si no se especifica ningún alcance, se usa el registro predeterminado, que es proporcionado por el parámetro de configuración del registry . Consulte npm config , npmrc y config para obtener más información sobre la administración de npm configuración.

5.- **¿Que son los módulos en Node?**

En Node. js, un módulo es un conjunto de funciones y objetos de JavaScript que las aplicaciones externas pueden usar. La descripción de un fragmento de código como módulo se refiere menos a lo que es el código que a lo que hace; cualquier archivo de Node.

6.- Cuáles son los dos tipos de funciones de API in node.j

a. Funciones asincrónicas, sin bloqueo
b. Funciones de bloqueo síncronas

Funciones asincrónicas, sin bloqueo: como su nombre indica, estas funciones operan de forma asincrónica. Lo que significa es que cuando Node.js haga una solicitud de datos a la API, no se bloqueará hasta que se reciban los datos. En cambio, continuará moviéndose a la siguiente API después de llamarla, y un mecanismo de notificación de un evento Node.js responderá al servidor para la llamada API anterior. Para decirlo en términos sencillos, estas funciones permiten seguir trabajando mientras se maneja la solicitud. Ejemplo: correos electrónicos, foros en línea.

Funciones de bloqueo síncronas: a diferencia de las funciones asíncronas, las funciones síncronas actúan como funciones de bloqueo. Lo que significa es que estas funciones harán que el sistema de llamada espere una respuesta. Por lo tanto, cuando un sistema usa API síncronas, espera obtener datos inmediatos cuando se realizan requests. Estos tipos de API se utilizan cuando la disponibilidad y la conectividad son altas y se espera una baja latencia. Para decirlo en términos simples, la aplicación solicitará y esperará una respuesta hasta que se devuelva el valor. Ejemplo: mensajería instantánea, videoconferencias

Ejemplo: tenemos un archivo JSON

```
{
   "name": "John",
   "age": 50,
   "gender": "male" }
```

```
// Requiring inbuilt module
const fs = require('fs');

// FUNCION ASINCRONA
fs.readFile('data.json', 'utf8', function (err,data) {
  if (err) {
    return console.log(err);
  }
  console.log("Below is the Data from Asynchronous function call")
  console.log(data);
});

// FUNCION SINCRONA
var data = fs.readFileSync('data.json','utf8');
console.log("Below is the Data from Synchronous function call")
console.log(data);
```

7.- Que es el archivos package.json

El archivo package. json contiene todos los metadatos acerca del proyecto tal como descripción, licencia, dependencias y scripts. En la mayoría de los casos es sencillo encontrar que módulos son requeridos y cuales son los comandos principales y archivos al solo ver la estructura del archivo package. Json

8.- Que es el archivos package.json.lock

El archivo package-lock.json es un archivo generado automáticamente cuando se instalan paquetes o dependencias en el proyecto. Su finalidad es mantener un historial de los paquetes instalados y optimizar la forma en que se generan las dependencias del proyecto y los contenidos de la carpeta node_modules/.

Este archivo debe conservarse e incluso versionarse para añadirlo al repositorio de control de versiones, puesto que es algo favorable para el trabajo con npm. Es por ello que no debe añadirse al fichero .gitignore.