



INGENIERIA EN CIENCIAS DE LA COMPUTACIÓN

NOMBRE: OSCAR JIMÉNEZ **CARRERA:** COMPUTACIÓN **ASIGNATURA:** PLATAFORMASWEB

PRACTICA: 02 **TÍTULO:** FUNDAMENTOS DE NODE.JS

OBJETIVO:

- Entender los fundamentos del lenguaje de programación Node.js

PRE REQUISITOS:

- a) Computador con Microsoft Windows o GNU/Linux
- b) Interprete de Node.js versión 12 o superior
- c) Editor de código fuente Visual Studio Code
- d) Repositorio de software Git

- INSTRUCCIONES:**
1. Lea detenidamente cada uno de los enunciados propuestos
 2. Plantee una solución a cada uno de los ejercicios
 3. Programe una solución utilizando el lenguaje de programación Node.js
 4. Elabore un informe con la solución de los ejercicios

ACTIVIDADES A DESARROLLAR:

1. Documentar scripts

Estudiar, ejecutar y documentar (colocar comentarios) los scripts revisados en clase. Por cada uno de los ejercicios, colocar una descripción en sus propias palabras de los conceptos más importantes. El informe debe contener los siguientes ejercicios:

- REQUERIR PAQUETES

```
C:\Users\PERSONAL\Desktop\holaNode\bases>nmp i yargs --save_
```

```
C:\Users\PERSONAL\Desktop\holaNode\bases>nmp init
```

```
C:\Users\PERSONAL\Desktop\holaNode\bases>nmp i colors
```

Con este comando podemos instalar los paquetes que requiere yargs. Una vez demos entre se creará una carpeta llamada node_modules. Con el segundo comando tendremos acceso a dos archivos .json que serán los encargados en definir ciertos parámetros de nuestro proyecto. Y por último tenemos el i colors que este será la librería para poner colores a los resultados impresos en consola.

- **IMPORTAR ARCHIVOS AL PROYECTO**

```
16
17     fs.writeFile(`tablas/tabla-${base}.txt`, data, (err) => {
18         if (err)
19             reject(err);
20         else
21             resolve(`el archivo de la tabal ${base} a ${limite} sido creado!`.green);
22     });
23 }
24
```

Con la línea de código 17 podemos encontrar la función para crear archivos. Fs es una librería Importar previamente para su uso y con writeFile lo que hacemos es poner el nombre de nuestro Archivo, el dato que vamos a guardar y de igual manera ejecutar un error por si lo hubiera.

Resultado:

```
C:\Users\PERSONAL\Desktop\holaNode\bases>node app crear --lim 10 -b 8
el archivo de la tabal 8 a 10 sido creado!
```

- **RECIBIR INFORMACIÓN DE LA LÍNEA DE COMANDOS**

```
1  //definimos los comandos para ejecutarlo en el terminal
2  let opc = {
3      base: {
4          demand: true,
5          alias: 'b'
6      },
7      limite: {
8          alias: 'l',
9          default: 10
10     }
11 }
12
13 //Creamos descripciones para que el usuario tenga un ayuda por si no sabe la funcion
14 const argv = require('yargs')
15     .command('listar', 'Imprime en consoloa la tabla de multiplicar ', opc)
16     .command('crear', 'Creado ', opc)
17     .help()
18     .argv;
19
20 module.exports = {
21     argv
22 }..
```

Creamos un objeto llamado base y otro limite en el cual definimos comandos para ejecutar el programa según sea el criterio establecido.

Con la constate yargs lo que hacemos es crear pequeñas frases con las cuales el usuario se podrá guiar si es que no conoce el programa y quiere ejecutarlo

Resultado:

```
C:\Users\PERSONAL\Desktop\holaNode\bases>node app listar --limite 8 -b 8
=====TABLA DEL 8=====
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
=====
```

- **COLORES EN LA CONSOLA**

Con la librería colors podemos pintar las letras que aparecerán en el terminal. Esto se puede hacer de dos formas la primera es invocando a la librería colors como en el ejemplo 1 y la segunda simplemente es agregar una extensión con un punto y el nombre del color que queramos al final de un console.log como en el ejemplo 2

EJEMPLO 1

```
console.log(colors.red(`=====TABLA DEL ${base}=====`));
let data = '';
for (let i = 1; i <= limite; i++) {
  data += `${base} x ${i} = ${base*i}\n`;
}
```

EJEMPLO 2

```
console.log(data.yellow);
console.log("=====".green);
});
```

Resultado:

```
C:\Users\PERSONAL\Desktop\holaNode\bases>node app listar --limite 9 -b 17
=====TABLA DEL 17=====
17 x 1 = 17
17 x 2 = 34
17 x 3 = 51
17 x 4 = 68
17 x 5 = 85
17 x 6 = 102
17 x 7 = 119
17 x 8 = 136
17 x 9 = 153
=====
```

PUBLICAR PROYECTO EN GITHUB

Iniciamos nuestro repositorio con el comando `git init`

```
C:\Users\PERSONAL\Desktop\holaNode\bases>git init
Reinitialized existing Git repository in C:/Users/PERSONAL/Desktop/holaNode/bases/.git/
```

Luego realizamos un commit para que las carpetas pasen al lugar preliminar para ingresar a nuestro repositorio

```
C:\Users\PERSONAL\Desktop\holaNode\bases>git commit
On branch master
Changes not staged for commit:
  modified:   app.js
  modified:   config/yargs.js
  modified:   multiplicar/multiplicar.js
```

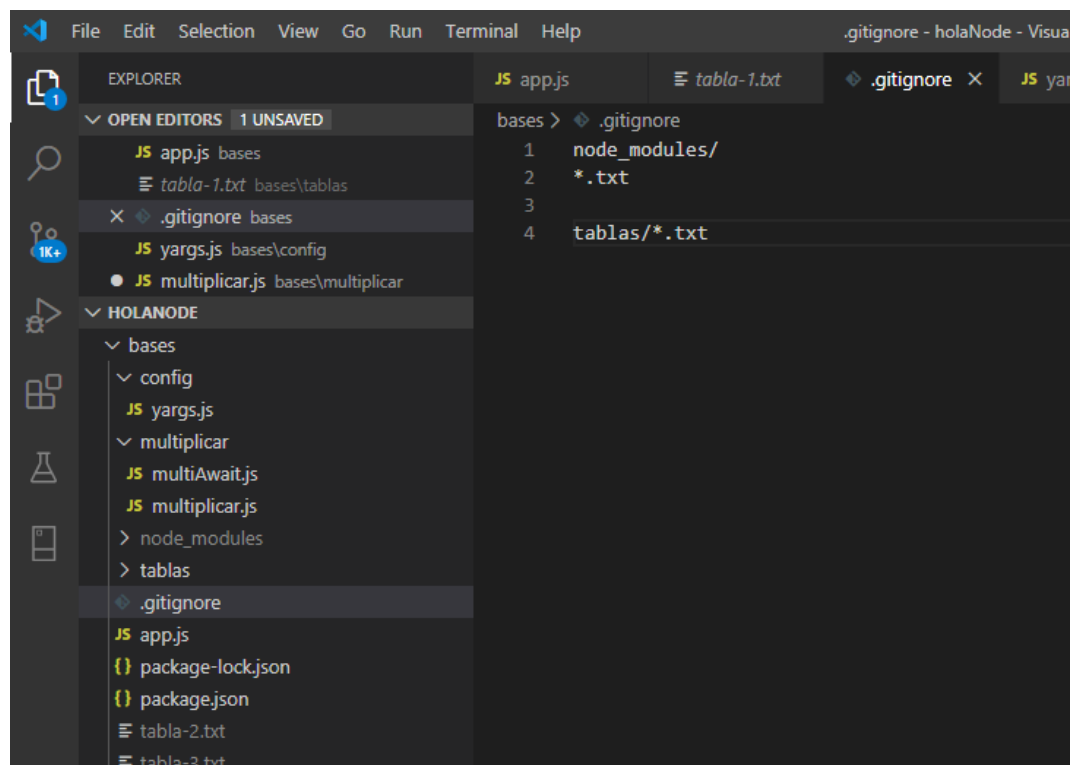
Ingresamos el comando `git add` con nuestra url del repositorio

```
C:\Users\PERSONAL\Desktop\holaNode\bases>git remote add origin https://github.com/ojimenezl/PlataformasWeb7.git
```

Y finalizamos con el comando `git push -u origin master`

```
C:\Users\PERSONAL\Desktop\holaNode\bases>git push u-origin master_
```

Recordemos que va a existir persona que talvez quieran utilizar nuestros proyectos, por lo que debemos previamente omitir la carga de ciertos archivos como la carpeta `node_modules` para ello creamos un archivo llamado `.gitignore` y ponemos dentro de este lo que no queremos que se suba a github.



Resultado:

2. GitHub

Subir el código revisado y documentado a su cuenta de GitHub. Colocar el enlace del repositorio en el informe de la práctica.

REPOSITORIO GITHUB:

<https://github.com/ojimenezl/PlataformasWeb7>

RESULTADOS OBTENIDOS:

1. El estudiante está familiarizado con la sintaxis y los fundamentos del lenguaje Node.js

CONCLUSIONES:

- Con esta nueva implementación de poder introducir datos por consola, la enseñanza cada vez es más completa y más entendible, de igual forma nos permite realizar proyectos pequeños que eficaces y aún más atractivos para el usuario con la librería "colors". Esta práctica nos deja con un mejor aprendizaje y con funciones nuevas para ponerlas en práctica ya que son esenciales tanto en un proyecto universitario como en el ámbito laboral

REFERENCIAS:

- [1] F. OpenJS, "Node.js," nodejs.org. [Online]. Available: <https://nodejs.org/en/>

