**Ejercicios prácticos Node.js**

Ejercicio práctico con ToDoList.

Creamos la carpeta y le corremos **npm** para que creé el archivo package.json y creamos el archivo **app.js** para el proyecto:

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

**Dependencias para trabajar.**

<https://nodejs.org/docs/latest/api/>

**Redline**: Crea una forma de interactuar con el flujo input y output de una interfaz de línea de comandos permitiendo la lectura de datos desde la consola.

Librería de **npm** (<https://www.npmjs.com/package/chalk>) **Chalk**: Servirá para proporcionar colores a terminal identificando código.

Ambas son dependías del proyecto, no de desarrollo. Por lo tanto, se crean en la sección “dependences” (--save). **Como son parte del core de Node.js no es necesario instalarlas, podemos utilizarlas de esta manera:** Se intala Chalk con “npm i chalk” para tener su dependencia, y se colocan en el archivo app.js.

Texto

Descripción generada automáticamente **Recordemos que al hacer esa importación de módulos dentro del archivo main, tenemos que especificar “type: “module”” dentro del archivo json para que puedan ser detectados.**

Ahora se crea la interfaz readline para procesar entradas y salidas de datos.

Texto

Descripción generada automáticamente

Y posteriormente creamos el menú de la app basándonos en una función.

Texto

Descripción generada automáticamente

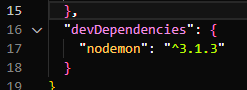
Damos “node app.js” en la terminal y obtenemos el resultado.

Texto

Descripción generada automáticamente

**Nodemon.**

Esto es una herramienta de desarrollo utilizada con aplicaciones Node.js. Ayuda a reiniciar automáticamente las aplicaciones al detectar cambios en el archivo(s) del proyecto. Quitando la necesidad de reiniciar de forma natural el servidor siendo mas eficiente durante el desarrollo. Ahora hacemos su instalación como dependencia de desarrollo de la siguiente manera “npm i –save-dev nodemon” y también de forma global “npm install -g nodemon”.



En la consola podemos utilizar el comando **nodemon -h** para visualizar las características y los archivos que nos da nodemon para utilizarlo correctamente.

Pantalla negra con letras blancas

Descripción generada automáticamente con confianza media

Una vez consiguiendo esto ejecutamos un script personalizado dentro del package.json de raíz para que nodemon pueda ser ejecutado en la aplicación permitiendo que haga el reinicio automático al detectar un cambio.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente De esta forma quedaría el script.

Ahora simplemente echamos a andar la aplicación únicamente con el comando **npm start**.

Texto

Descripción generada automáticamente

**Programando la comunicación con la CLI (Interfaz de línea de comando).**

Creamos una función que ayuda a elegir una opción de las cuales ya están en el menú. Lo realizamos con una callback de la siguiente manera:

Texto

Descripción generada automáticamente

Esta función de callback (con la función de dysplayMenu()) contiene el argumento **“question” de redline**, el cual contiene un parámetro de respuesta del usuario y el mismo parámetro “choice” se tomará de un **switch**, el cual contiene las **opciones indicadas en el menú**.

En caso de que no sea una de esas opciones, el **default** se tomará como opción no valida y se aplica una **recursividad** mandando llamar de nuevo la función del menú y el **choiceOption()** (aplicando la función de la callback).

**coiceOption** **solo se accionará en caso de elegir una opción del manu**.

**Programación de cada tarea (menu).**

**Creamos la función para agregar tarea (opción 1).**

Captura de pantalla de computadora

Descripción generada automáticamente

El **rl.question** que solicita al usuario ingresar su tarea, misma que se detecta con el valor de task, dentro del array **tasks** se almacenarán (en memoria) las tareas ingresadas con ayuda del **push**.

Dichas tareas se agregan como objeto al array **tasks** con valores de “task” para referirse a si misma como tarea, y “comlpete” indicando su estado inicial de la misma tarea que aún no está completada (false).

Al terminar de escribir y agregar la tarea, se devuelve el mensaje de la línea 37 y se vuelve a continuar mostrando el menú de opciones junto con su acción de elegir (líneas 39 y 40).

**Creando el listado de tareas (opción 2).**