**Laboratorio de Node JS**

Consideraciones

* El presente laboratorio considera que el servidor tiene el sistema operativo Windows recién instalado.
* Como editor de texto se usará Visual Studio Code (VSC).

Primeros pasos

* Instalación de Node JS
* Instalación de npm (Viene en la instalación de Node JS)

Procedimiento

* Crear una carpeta en donde pondremos nuestro proyecto (servidor).
* Abrir el forder creado desde VSC.
* En VSC, abrir la terminal (CTRL + ñ). En el terminal, debemos ubicarnos en la ruta del folder creado.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

* Instalamos nuestro servidor Node js con la sentencia “npm init -yes”.

Texto

Descripción generada automáticamente

Se creará nuestro archivo package.json.

Texto

Descripción generada automáticamente

* Instalamos la dependencia ***express*** con el comando “**npm install express”**.

Texto

Descripción generada automáticamente

* Al culminar la instalación, nos aparecerá la carpeta **node\_modules** y el archivo **package-lock.json**. En nuestro archivo original package.json podemos visualizar la nueva dependencia instalada.

Texto

Descripción generada automáticamente

* Creamos nuestro archivo de configuración con el nombre de app.js con la siguiente configuración.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

* Línea 1: Requerimos el módulo de **express** y la guardamos en una constante llamada **express** con el objetivo de usar sus funciones.
* Línea 2: Creamos el objeto app con la función **express()**.
* Línea 3: Indicamos el puerto por donde el servidor debe escuchar las comunicaciones. En este caso le ponemos 8000. En este punto nuestro servidor ya está funcionando.
* En el terminal, iniciamos nuestro servidor:

Texto

Descripción generada automáticamente

* Validamos en un navegador web con las siguientes rutas:
  + <http://localhost:8000>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* + <http://127.0.0.1:8000>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Para validar nuestro servidor en una red local, debemos de saber la ip de nuestra privada, en este caso es 192.168.18.14. El servicio web podrá ser accedido desde otro equipo local.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Ya podemos acceder a nuestro servidor por medio de otro dispositivo, como un celular móvil.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

* Por buenas prácticas, se recomienda poner un mensaje para conocer que el servidor está iniciando, esto lo podemos hacer mediante una función callback indicando la inicialización del servidor con el puerto asociado.

Texto

Descripción generada automáticamente

La forma actual de escritura de funciones con un solo parámetro nos permite escribirlos de la siguiente manera:

Captura de pantalla de un celular

Descripción generada automáticamente

* Al inicializar el servidor, obtenemos el siguiente mensaje.

Texto

Descripción generada automáticamente

NOTA: El mensaje “Cannot GET /” que aparece en el servidor es porque no está respondiendo con alguna especificación.

* Creamos el archivo index.html y dentro de ella ponemos un mensaje simple.

Captura de pantalla de un celular

Descripción generada automáticamente

* En nuestro archivo app.js especificamos que al iniciar el servidor debe de inicializar nuestra página indicándole la ruta en donde está ubicada.

Captura de pantalla con la imagen de una pantalla

Descripción generada automáticamente

* Inicializamos nuestro servidor nuevamente

Captura de pantalla de un celular

Descripción generada automáticamente

* Validamos nuestra página web.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Buenas prácticas:

Ruta variable:

* Se recomienda indicar la ruta del archivo index no como un valor fijo sino como una variable. Para ello debemos utilizar un módulo de Node JS llamado “path”y hacemos que la ruta raíz en donde está nuestro servidor sea una variable y desde ahí podemos convocar nuestra página web.

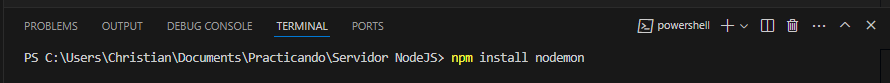
Captura de pantalla de computadora

Descripción generada automáticamente

* Línea 3, hemos invocado al módulo path.
* Línea 7: usamos el método join (unión) de path y usamos la constante “\_\_dirname” que representa nuestra ruta raíz del proyecto + la página web.

Reinicio automático:

Para evitar estar apagando el servicio y luego inicilizandolo, se recomienda usar nodemon, esto debemos de instalarlo.



En nuestro archivo package.json visualizamos que se realizó la instalación correctamente.

Texto

Descripción generada automáticamente

Indicamos que se ejecute nodemon con nuestro archivo app.js.

Texto

Descripción generada automáticamente

* Ahora, inicializamos nuestro servidor, pero con el comando “**npm run start**”. Cada vez que realicemos un cambio en nuestro servidor y guardemos, se actualiza de manera automática.

Texto

Descripción generada automáticamente

Carpeta Pública:

* Los archivos que forman parte de nuestra página web deben colocarse dentro de una carpeta llamada public (por convención). Al hacer esto, debemos de actualizar nuestra referencia de rutas.

Captura de pantalla de computadora

Descripción generada automáticamente