LABORATORIO NODE.JS / MONGODB
CLASE 1

Vamos a crear una estructura de trabajo organizada para nuestro proyecto de curso. En el mismo iremos explorando las distintas posibilidades que nos brinda Node.js como entorno de desarrollo y MongoDB como sistema de organización de información. Para ello crearemos servidores de distintos tipos ,serviremos archivos estáticos a través de la web y podemos interactuar con nuestra información.

Ejercicio 1

Vamos a trabajar sobre un archivo llamado servidor.js el cual será el punto inicial para nuestro programa general.

Ejercicio 2

Dentro del archivo creado vamos a crear un simple servidor de HTTP que nos permita abrir el puerto 8080 y nos muestre un mensaje de texto simple en pantalla si ingresamos a nuestra web a través de http://localhost:8080 (En caso de querer usar el puerto 80 node debe correr con permisos de administrador)

Ejercicio 3

Vamos a crear un archivo adicional llamado template.js en donde tendremos constantes que guarden un string literal con un template de HTML. Arranquemos con una estructura básica de html+head+body+h1 . La constante deberá ser exportada como contenido default del archivo template.js

Ejercicio 4

Importemos nuestro archivo previamente creado template.js dentro de servidor.js . A continuación modifiquemos la respuesta del servidor por una respuesta con header cuyo content-type corresponda al contenido que queremos servir, en este caso HTML, y enviamos el contenido de template.js como respuesta del servidor.

BONUS

Modifiquemos el código de nuestro servidor anterior para que el mismo pueda admitir varias rutas y que sea capaz de resolver varias respuestas. Para eso tendremos que hacer uso de la variable url que se encuentra dentro del objeto respuesta en cada solicitud del servidor y crear distintos templates para cada caso.

Ejercicio 5

Vamos a crear los archivos home.js , usuarios.js , mensajes.js y perfil.js . Cada uno debe tener una plantilla básica de HTML tal y como lo hicimos en el archivo template.js

Ejercicio 6

Debemos crear una estructura de condicional múltiple en donde para cada caso de url podamos responder con algo distinto.

Ejercicio 7

Vamos además a dejar montada una ruta específica para una API REST en donde podemos consumir servicios JSON desde la base de datos que vayamos a tener más adelante en el curso. Para ello vamos a crear una ruta que apunte a nuestro mismo dominio o host (EJ.: localhost) con el prefijo api para cada llamada. Ej.: Si tuviéramos un recurso para traer usuarios la url podria ser http://localhost/api/usuarios . Vamos a necesitar las rutas usuarios , posts, comentarios y mensajes y cada uno tiene que devolver a cambio un array vacío en formato JSON