

## Segunda Parte

### Servidor a la Express.js / Servidor estilo Express.js

#### Ingredientes:

- Sitio web al gusto
- Archivo index.js
- node.js
- npm
- express.js

#### Instrucciones:

1. Elige el lugar donde deseas poner tu proyecto. Te recomendamos crearle su propia carpeta en un lugar fácil de recordar.
2. Con ayuda de tu terminal instala node.js<sub>1</sub>. Un dato curioso de este ingrediente como recordará es que nos ayuda a correr Javascript fuera de un navegador. (Si quieres saber más sobre Javascript visita el apartado en la parte inferior de la página).
3. Verifica que la instalación haya sido exitosa y de ser así, ahora es momento de auxiliarnos de npm<sub>2</sub> para hacer la instalación de Express.js<sub>3</sub>. Una forma usual de realizar la instalación es a través de: npm<sub>4</sub> install express<sub>5</sub> --save  
De ser necesario ejecutar lo anterior como superusuario y espera hasta que llegué a su punto de cocción.
4. Una vez que ya tenemos Express.js<sub>6</sub> listo. Procedemos a incorporar nuestro archivo index.js<sub>7</sub>, a la carpeta del proyecto, este archivo contiene las directivas para levantar nuestro servidor. El cual luce de la siguiente manera:  
Imagen del archivo, podemos poner como ejercicio que identifiquen las partes del middleware, etc.
5. ¡Ya casi tenemos todo listo! solo falta agregar el sitio web al gusto, el primer paso para esto es crear la carpeta que albergará estos archivos dentro de nuestro proyecto. Ya que se trata de archivos que una vez que son lanzados no cambian (html y css), es decir son estáticos<sub>8</sub>, pero que además serán visibles para todos los usuarios llamaremos a nuestra carpeta public. (Un buen chef realiza el paso anterior y las posteriores creaciones de archivos y carpetas a través de la terminal)
6. Si quieres ser más ordenado y crees estar listo para dar un paso más para perfeccionar tus habilidades en proyectos web, separa los archivos de cada tipo en carpetas diferentes: una carpeta para imágenes llamada img, una carpeta para los archivos que sirven como estructura llamada html<sub>9</sub> y otra para los estilos llamada css<sub>10</sub>. (Si existieran archivos de otro tipo, por ejemplo : javascript. Tendrían su propia carpeta y hay algunas convenciones en cuanto a nombre, en este caso el nombre de la carpeta usualmente es js).

Nota: Si decides realizar este paso es posible que debas hacer algunas modificaciones en cuanto a rutas dentro de tus archivos html. Es importante familiarizarse con lo anterior ya que a medida que los proyectos crezcan tendrás más archivos y te será más fácil ubicarlos si se encuentran en carpetas separadas.

7. Estamos a unos pocos pasos de terminar, ya sólo debemos indicar que la carpeta que tiene nuestros archivos estáticos es public. Esto se realiza dentro de nuestro archivo index.js. A través de la siguiente línea:  
`app.use(express.static('public'));`  
Agrega la línea y guarda los cambios.
8. Ahora es momento de probar nuestra creación y verificar que tenga buen sabor. Corre el archivo index.js<sup>11</sup> en tu terminal a través del comando:  
node index.js<sup>12</sup>.  
La primera parte corresponde a lo que lo va a ejecutar (nuestro entorno de ejecución) y la segunda al archivo con terminación .js que tiene las pautas para levantar el servidor.
9. La terminal debe desplegar un mensaje donde indica que tu aplicación está corriendo en el puerto número 3000<sup>13</sup>, como lo indicamos en index.js. Ve al localhost<sup>14</sup>, que hace referencia a tu mismo ordenador en la dirección 127.0.0.1 de tu navegador ¡No olvides incluir el puerto! y visualiza tu sitio web desplegado.
10. Si llegaste hasta este paso es hora de que disfrutes y presumas tu creación con tu familia, amistades o con quien tu quieras. Con el pequeño detalle de que se encuentre observando la misma computadora que tú.
11. ¡El siguiente paso es conquistar el mundo! lo cual es posible a través de una plataforma de hosting que permita que todos en cualquier lugar vean tu sitio web.

Clasificación	Respuesta
1	node.js
2	npm
3	express.js
4	npm
5	express
6	express.js
7	archivo index.js
8	estáticos
9	html
10	css
11	index.js
12	node index.js
13	3000
14	localhost