

**Laboratorio 8: Sitio Web con Express**

**Objetivos:**

Al finalizar el laboratorio el estudiante será capaz de:

* Entender el funcionamiento de los middlewares.
* Desarrollar aplicaciones web con el framework Express.
* Resolución de problemas en Javascript.

**Seguridad:**

* Ubicar maletines y/o mochilas en el gabinete del aula de Laboratorio.
* No ingresar con líquidos, ni comida al aula de Laboratorio.
* Al culminar la sesión de laboratorio apagar correctamente la computadora y la pantalla, y ordenar las sillas utilizadas.

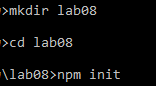
**Equipos y Materiales:**

* Una computadora con:
* Windows 7 o superior
* Conexión a la red del laboratorio
* Instalador de node.js

**Procedimiento:**

**Entendimiento de ejecución de funciones en JavaScript**

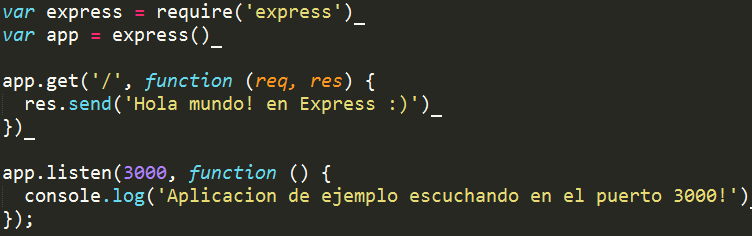
1. Iniciaremos la creación de nuestro laboratorio 8 creando la carpeta de nuestro proyecto en la raíz de la unidad C:
   1. Ingresamos los siguientes comandos y seguimos el asistente de NPM



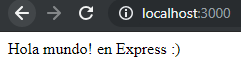
* 1. Instalamos la librería Express de modo local.



* 1. Creamos el archivo index.js con el siguiente contenido.







* 1. Como vemos, el código es distinto pero un poco más intuitivo. Esto es lo básico para nuestra aplicación en Express. Sin embargo, generaremos un esqueleto para conocer cuál es el estándar recomendado para Express.
  2. Instalaremos la herramienta express-generator, de manera global.



* 1. Mostramos todas las opciones disponibles de express con el argumento –-help o -h



* 1. Por ejemplo, crearemos la aplicación llamada **miapp**. Esta aplicación será creada en un folder llamado **miapp** en el directorio actual de trabajo.



* 1. Luego instalamos las dependencias (regeneramos el archivo npm)

****

* 1. Si estas usando MacOS o Linux, ejecuta el siguiente comando:

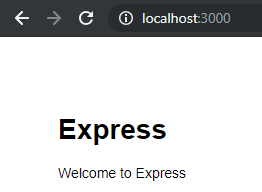
****

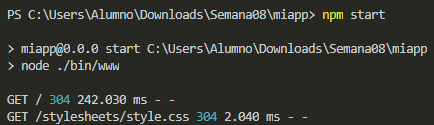
* 1. Si estas usando Windows, utiliza este:

****

* 1. Luego abre en el navegador la siguiente dirección:

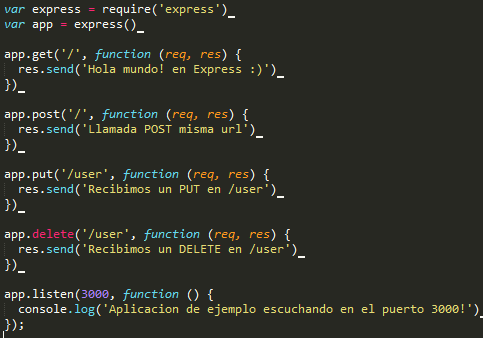
****





* 1. La estructura de la aplicación creada por el generador es una de las muchas formas que se puede estructurar una aplicación Express. Puedes usar este modelo o el que desees.

Ahora reemplaza el archive index.js que se creó en el inicio del laboratorio por el siguiente.



* 1. Explica los cambios que se buscan al utilizar middlewares para capturar la misma URL pero con diferente método HTTP.

**Al emplear middlewares se busca administrar la data que sale o entra de la aplicación. Se divide en los distintos métodos como POST o GET para redireccionar la data dependiendo del método.**

* 1. Instala el navegador Mozilla Firefox. Este siempre ha sido un navegador con muchos complementos para los desarrolladores web.
  2. Ingresa a la siguiente dirección:



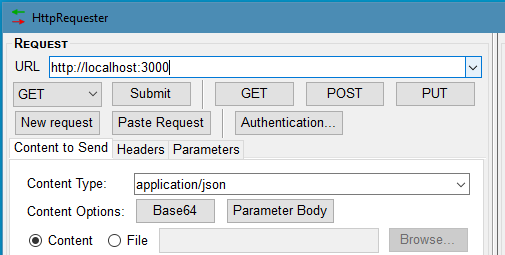
* 1. Estamos accediendo al complemento HttpRequester. Instalelo y reinicie el navegador cuando le pida la aplicación.

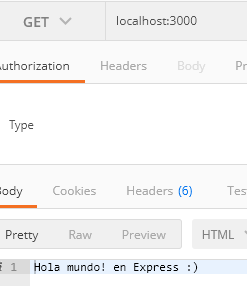
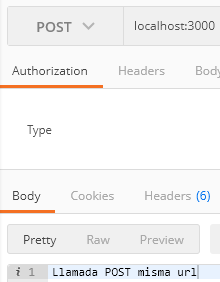


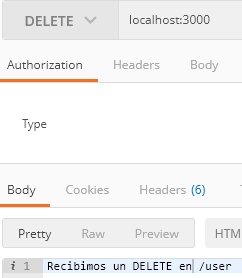
* 1. Se habrá añadido un ícono con el complemento en la parte superior derecha. Dele click para abrir la herramienta.



* 1. Para utilizar la herramienta, no basta más que con ingresar la URL deseada y hacer click en el método HTTP que deseemos probar. Adjunte capturas del uso de los middlewares declarados en el **index.js**

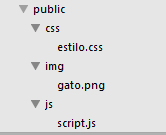




* 1. Para agregar archivo estáticos como imágenes, archivos de estilo CSS, y archivos JavaScript, use la función middleware **express.static**.

Para hacer esto, basta con pasar el nombre del directorio que contiene los archivos que deseamos publicar. Genere la siguiente estructura con archivos de ejemplo (pueden estar vacios o contener algo para demostrar el funcionamiento de la aplicación)

****

* 1. Declárelos con la función antes mencionada.

****

* 1. Pruebe la carga de estos archivos accediendo a sus URLs respectivas. Adjunte capturas de su verificación.

****

* 1. Express permite el uso de varios directorios. Basta con usar otro middleware parecido, no hay límite de uso.

****

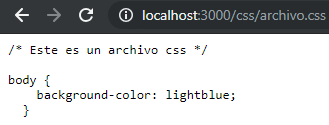
* 1. También se pueden crear prefijos de rutas, para el enmascaramiento de archivos. Basta con pasarle un argumento de prefijo para lograrlo.

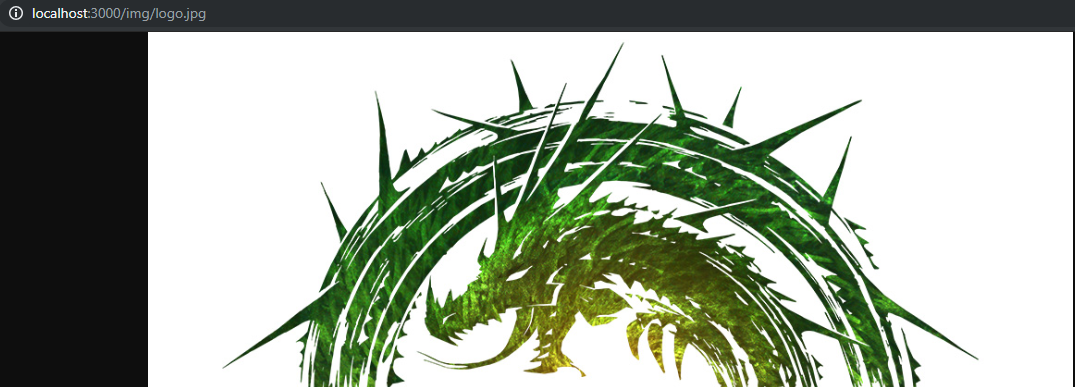
****

* 1. Adjunte capturas de la verificación de la existencia de los archivos.

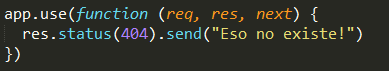






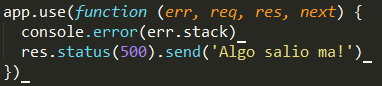


* 1. Para manejar errores necesitamos hacer uso de un middleware de errores. Primero lidiaremos con uno que devuelva algo cuando se accede a un recurso que no hemos declarado, es decir, “no existe”. Ya que los middlewares se ejecutan de manera progresiva, es importante decir que para el uso de un middleware de código 404 debe hallarse por lo general casi al final de su código.

******



* 1. La única diferencia será el middleware que capturará excepciones de código. Este recibirá un parámetro adicional, que es el error sucedido.



1. **Ejercicio propuesto**

Desarrolla un sistema utilizando estilos, jQuery y Express para la matrícula de un curso de capacitación. La matrícula se realizará en 3 pasos.

1. **Paso1:** Desde una lista desplegable debe seleccionarse un curso (Java, PHP y .NET). Luego se pulsará un botón siguiente.
2. **Paso2:** Luego deberá mostrarse una lista para marcar (con checkbokx) de los módulos que el participante llevará. Luego se pulsará el botón siguiente.
   1. Básico
   2. Intermedio
   3. Avanzado
3. **Paso3:** Luego deberá mostrarse para escoger (con radiobutton) si el pago será con tarjeta de crédito o en efectivo, finalmente se pulsará el botón siguiente.
4. Luego el sistema deberá mostrar el detalle de lo escogido y el total a pagar

***Nota.-*** Cada módulo del curso de Java esta S/ 1200, los de PHP S/ 800 y los de .NET S/ 1500. Si se paga en efectivo se aplicará un descuento del 10%.

**Conclusiones:**

Indicar las conclusiones que llegó después de los temas tratados de manera práctica en este laboratorio.

|  |
| --- |
|  |