

## Práctico N° 5 – Node.js - Webapp

1.- Cree un servidor web que muestre por consola un mensaje “Servidor web iniciado” y que escuche peticiones en el puerto 8000 respondiendo el mensaje “bienvenido a su primera aplicación web”.

El servidor también debe mostrar en un div con formato la información del método http de la solicitud, el host, la ruta hacia el recurso y el recurso.

2.- Cree una aplicación web que atienda las rutas index, contacto y about.

La aplicación debe retornar el 404 para atender una ruta no considerada y un 500 en el caso de que exista un error al procesar el requerimiento.

El contenido de la respuesta del 404 debe ser una imagen que haga referencia al código de status.

a) Modifique su servidor. Use `console.log()` para mostrar en la consola todos los requerimientos que le llegan al servidor.

3.- Desarrolle una aplicación web que permita retornar todo el contenido estatico que se publique dentro de la carpeta /public de la aplicación.

4.- Implemente algún mecanismo de routing (sin utilizar una librería) que facilite la modularidad de la aplicación y unifique el lugar para atender los requerimientos.

5.- Agregue una funcionabilidad a la aplicación para que responda a la ruta /selects.

La respuesta deberá ser una página que retorne un select cargado con nombre de colores (en español) el cual una vez seleccionado/cambiado debe aplicar el color al fondo de la página.

El listado de colores debe obtenerse del archivo colores.dat que tendrá los nombres de colores y sus respectivos códigos hexadecimales.

6.- Utilice el módulo node-fetch para hacer un requerimiento get a la api <https://jsonplaceholder.typicode.com/users>

Esta api retorna un listado de usuarios fake.

Implemente una ruta en la aplicación que consuma la api y retorne una tabla html con la información de los usuarios recuperados.