Proyecto de Aplicaciones Web  
Plataforma: CANVAS

Profesor: Ing. Jorge Luis Flores  
jorge.flores@tec.mx

NOMBRE:   
HEROKU LINK :

Node + Express

En este laboratorio pondrás en práctica lo visto respecto a Node y Express.

El objetivo es desarrollar una aplicación web sencilla para la administración de reservaciones de un restaurante. El restaurante solo tiene 5 mesas disponibles. Los primeros 5 en reservar obtienen reservación, cada petición de reserva posterior deberá enviarse a una lista de espera. Simularemos la BD por medio de arreglos que almacenarán las reservaciones (similar a los ejercicios de Star Wars).

Antes de Comenzar

La aplicación tendrá 2 partes:

1. Front-end HTML/JS para ingresar y visualizar datos. (La parte visual se te proporcione con 3 archivos “home/reserve/tables”, pero eres libre de crearlos desde 0 si así lo deseas).
2. Back-end que incluirá Node/Express para almacenar y actualizar datos de reservación.

Te recomiendo primeramente hacer un diagrama para visualizar mejor la aplicación. Considera los conceptos vistos en clase hasta el momento:

* Servidores
* Routing
* APIs
* AJAX (GET y POST)

Te ayudará mucho utilizar el ejercicio de Stars Wars que vimos en clase, particularmente la versión final del mismo. Puedes utilizar la siguiente URL como referencia de la funcionalidad:   
<https://mi-restaurante-cool.herokuapp.com/>

Es un laboratorio extenso, pero sencillo. Te ayudará a practicar nuevamente los conceptos previos y los nuevos.

La siguiente semana desmenuzaremos este laboratorio paso a paso. Anota todas las dudas que tengas durante tu desarrollo para revisarlas en clase.

**Instrucciones**

**Fases de Desarrollo**

**Fase I: Para esta primera fase, desarrolla las “piezas” que deberán programarse para crear la funcionalidad de la aplicación. Procura dividirlo en 6-7 partes:**

* El front end ya se te proporciona, pero eres libre de crearlo a tu gusto o modificar los colores/diseño/textos.
* Creación de un servidor básico de Express.JS (Recuerda que el puerto sea dinámico).
* Creación de 2 arreglos para almacenar los datos y simular una BD.
* Creación de rutas “GET” y “POST” para regresar y enviar información. GET= Enviarle al usuario las páginas html que solicite. POST = Manejar las peticiones con datos.
* Utiliza jQuery para ejecutar llamadas Ajax de GET y POST que se ejecutarán desde el front-end, puedes probarlas con POSTMAN antes de implementarlas en tu front-end.

**Fase II: Para esta segunda fase, completa lo siguiente:**

**Backend**

Al encender el servidor este deberá mostrar al usuario las páginas “home.html”, “reserve.html”, “tables.html”.

**FrontEnd**

Revisa los archivos: home.html tables.html reserve.html

Identifica la manera en que los datos, el API, y las rutas se comunicarán entre sí.

**Fase III: Para esta tercera fase, completa lo siguiente:**

**Backend**

Crea los arreglos para almacenar los datos de reservación y de lista de espera.

Crea las rutas para desplegar los datos de los arreglos en formato JSON, este vendría siendo el API. El usuario deberá poder visualizar los datos en la página correspondiente. (Por ejemplo, si el usuario visita <http://localhost:3000/api/tables> deberán visualizar los datos en formato JSON). Necesitas 2 rutas de APIS que solamente retornan el contenido de los arreglos.

/api/tables  
/api/waitlist

**FrontEnd**

Comienza a trabajar el código necesario para convertir los datos del formulario a objetos JSON. Y enviarlos por medio de un $.post o $.ajax hacia el servidor.

**Fase IV: Para esta tercera fase, completa lo siguiente:**

**Backend**

Desarrolla la lógica que maneje la administración de mesas en el restaurante. Tu código deberá poder manejar solicitudes de tipo POST que contengan objetos JSON, validar si hay espacio disponible, agregar los objetos JSON al arreglo de reservación o al de la lista de espera. Tu ruta POST también deberá regresar una confirmación (true / false) para saber si la reservación se hizo exitosamente (TRUE) o si está en la lista de espera (FALSE). Utiliza Postman para todas las pruebas en este punto. La respuesta puede ser utilizando el siguiente comando: *res.json(true)*

**FrontEnd**

Desarrolla el código necesario para que tus páginas tables.html puedan obtener los datos del backend. En esencia estarás creando peticiones de tipo GET para obtener los datos desde que la página se carga.

Desarrolla el código necesario en reserve.html para enviar datos, en esencia estarás creando peticiones de Tipo POST para enviar los datos.

**Fase V: Para esta quinta y última fase, completa lo siguiente:**

Prueba tu aplicación para asegurar que no existan bugs. Valida si hay alguna manera de hacer que la aplicación “truene”. De ser posible súbela a heroku y sube este documento con la liga a tu app.

(En caso de problemas sube tu código en un zip a Canvas e incluye errores o problemas que hayas tenido).

Este laboratorio se calificará como entregado / no entregado. Pero es importante que puedan identificar dudas que te surgan durante el desarrollo.

**Una vez que termines el desarrollo deberás de publicar este laboratorio en un nuevo**

**repositorio (TIENE QUE SER DIFERENTE A LOS REPOSITORIOS DE LOS LABORATORIOS**

**ANTERIORES DE LO CONTRARIO SE TE DESCONTARÁN 15 PUNTOS.**

“If you try and Fail, Congratulations. Most People won’t even try”