Proyecto de Aplicaciones Web  
Plataforma: CANVAS

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

NOMBRE:   
GITHUB REPOSITORY : [ ]

BONUS?: [ ]

Node + Express

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

El objetivo es desarrollar una aplicación web 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.

Antes de Comenzar

La aplicación tendrá 2 partes:

1. Front-end HTML/CSS/JS para ingresar y visualizar datos. (La parte visual se te proporcionará, pero eres libre de crearlos desde 0 si así lo deseas).
2. Back-end que incluirá Node/Express y javascript 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://hot-restaurantmx.herokuapp.com/>

Es un laboratorio extenso, pero te ayudará a practicar nuevamente los conceptos previos y los nuevos. Pide ayuda en caso de que te sientas perdido.

La siguiente semana desmenuzaremos este laboratorio paso a paso. Anota todas las dudas que tengas durante la actividad.

**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.
* Creación de un servidor básico de Express.JS
* Creación de algunos arreglos para almacenar los datos y simular una BD.
* Creación de rutas “GET” y “POST” para regresar y enviar información.
* Creación de rutas para desplegar las páginas html.
* 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**

Crea un servidor Express con lo mínimo (Que escuche en el puerto de tu preferencia).

**FrontEnd**

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

Trata de identificar la manera en que los datos, el API, y las rutas se verán y se comunicarán entre sí.

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

**Backend**

Crea variables (arreglo de objetos) para almacenar los datos de reservación y de lista de espera.

Crea las rutas para desplegar los datos 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).

**FrontEnd**

Crea las rutas de Express que muestren las páginas HTML cuando el usuario visite la ruta correspondiente. (Ejemplo, si el usuario visita <http://localhost:3000/tables> debería desplegarse el html de table.html)

Comienza a crear el código necesario para convertir los datos del formulario a objetos JSON.

**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.

**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.

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”. Si llegaste hasta este punto sin ninguna dificultad, intenta hacer cualquiera del siguiente bonus, y si lo logras asegúrate de marcar la casilla de BONUS al inicio de este archivo:

* Agrega Botones para hacer “check off” de la lista de reservación. Básicamente esto deberá eliminar a las personas de la lista de reservación (y su arreglo correspondiente) y mover el primer objeto de la lista de espera al último del arreglo de reservaciones.
* Agrega código en tu servidor para contabilizar cada vez que una persona visita cualquiera de las páginas. Muestra un contador en cualquier parte de la página.
* Agrega un botón para enviar correos a los usuarios de la lista de espera para avisarles cuando haya disponibilidad para ellos. Necesitarás investigar sobre **node-mailer** (es un paquete de npm).

**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”