Guía de Ejercicios - Laboratorio III (AJAX - TypeScript) - 2020

Profesor: *Neiner Maximiliano*

Parte 1 - Ejercicios Simples

Aplicación Nº 1 (Mostrar números impares)
Confeccionar un programa que solicite el ingreso de un número positivo (validar) y que muestre en un <input type="text"> la cantidad de números impares que hay entre ese número y el cero. Realizarlo utilizando el objeto ***XMLHttpRequest***.

Aplicación Nº 2 (Mostrar contenido de archivo)
Generar una aplicación Web, que permita al usuario ingresar el path de un archivo de texto. Al pulsar un botón, y utilizando el objeto ***XMLHttpRequest***, mostrar el contenido del archivo en un . En caso de que el archivo no exista, mostrar un mensaje de error por ***alert()***.

Aplicación Nº 3 (Determinar si una palabra existe)
A partir del ejercicio anterior, agregarle un <input type="text"> que le permita al usuario ingresar un palabra. Por medio del el objeto ***XMLHttpRequest***, buscar el archivo, y al encontrarlo, buscar dentro del archivo si existe la palabra que ingreso el usuario dentro del mismo.
Indicar, por medio de mensajes:

- Si el archivo existe o no.
- Si la palabra existe o no.

Aplicación Nº 4 (Calculadora)
Escribir un programa que posea dos <input type="text"> (uno para cada operando) y un <select> con los símbolos matemáticos: '**+**', '**-**', '**/**' y '*****'. De acuerdo al símbolo, se deberá realizarse la operación indicada y mostrarse el resultado por pantalla (en un). Realizar dicha operatoria con el objeto ***XMLHttpRequest***.

Parte 2 - Ejercicios Complejos

Aplicación Nº 6 (Verificar nombre de usuario)
Realizar una aplicación web que verifique la disponibilidad de un nombre de usuario. Para ello crear una página sencilla, que posea un <input type="text"> para ingresar el nombre de usuario y un <input type="button"> (con la leyenda: Verificar) que se comunicará (utilizando AJAX) al servidor de nombres (página *comprobarDisponibilad.php*) y obtendrá como respuesta un "SI" o un "NO", dependiendo si ese nombre de usuario está disponible o no.
Para simular la búsqueda sobre la base de datos de usuarios, en *comprobarDisponibilidad.php* se realizará lo siguiente:
Generara un falso retardo ocasionado por la búsqueda y el tráfico de la red de entre 0 y 6 segundos (utilizar función ***rand()*** y la función ***sleep()***. Buscar información). Además, retornará aleatoriamente, "SI" o "NO". A partir de la respuesta del servidor, mostrar un mensaje al usuario indicando el resultado de la comprobación.

Aplicación Nº 7(Sugerir nombres de usuario)
Normalmente, cuando se valida la disponibilidad de un nombre de usuario, se muestra una lista de valores alternativos en el caso de que el nombre elegido no esté disponible. Modificar el ejercicio anterior para que permita mostrar una serie de valores alternativos devueltos por el servidor.
Si el nombre de usuario está ocupado, los nombres de usuario alternativos se deben mostrar en forma de lista de elementos (
). Para ello generar un archivo de texto con varios nombres.
Modificar la lista anterior para que muestre enlaces (<a>) para cada uno de los nombres alternativos. Al pinchar sobre el enlace de un nombre alternativo, se copia en el cuadro de texto del login del usuario.

Aplicación Nº 8 (Mostrar noticias)
Diseñar una página HTML que incluya una zona llamada ***noticias***, en la que se deben mostrar noticias generadas por el servidor web. Añadir el código TypeScript necesario para:

- De forma periódica cada cierto tiempo (por ejemplo cada cinco segundos) se realiza una petición al servidor mediante AJAX y se muestra el contenido de la respuesta en la zona reservada para las noticias.
- Además del contenido enviado por el servidor, se debe mostrar la hora en la que se ha recibido la respuesta.
- Cuando se pulse el botón "Detener", la aplicación detiene las peticiones periódicas al servidor. Si se vuelve a pulsar sobre ese botón, se reanudan las peticiones periódicas.
- Añadir la lógica de los botones "Anterior" y "Siguiente", que detienen las peticiones al servidor y permiten mostrar los contenidos anteriores o posteriores al que se muestra en ese momento.
- Cuando se recibe una respuesta del servidor, se resalta visualmente la zona llamada ***noticias***.

Aplicación 9 (Mostrar las provincias y ciudades)

Crear un script que cargue de forma dinámica mediante AJAX la lista de provincias de nuestro país y las ciudades más importantes de cada provincia seleccionada.

Definir el código HTML de las dos listas desplegables vacías.

Cuando se cargue la página, cargar la lista de provincias en la primera lista desplegable.

Añadir de forma semántica el evento adecuado a la lista de provincias para que cuando se seleccione una provincia, se cargue automáticamente sus ciudades más importantes en la otra l ista.

Cuando se seleccione una determinada provincia, se carga mediante AJAX la lista completa de ciudades en la otra lista desplegable. El parámetro que se debe enviar al servidor es el código de la provincia.