



Facultad de Ingeniería

Escuela de Computación

Asignatura: Lenguajes Interpretados en el Cliente.

Código: LIC104

Grupo: Lab 03

Ciclo: 02

Año: 2022

Docente: Ing. Aida Quintanilla

Correo electrónico: aida.quintanilla@udb.edu.sv

Semana de estudio: 11/12 **Fecha de realización:** Del 10 al 22 de Octubre

■ Competencia

- ✓ Dominar el amplio del Modelo de Objetos de Documento (DOM).
- ✓ Acceder, eliminar y modificar el contenido de una página web a través del acceso a los nodos que forman el documento.
- ✓ Emplear los métodos y propiedades del estándar DOM para tener un dominio completo sobre el documento HTML.
- ✓ Hacer uso de algunos de los elementos del Modelo de Objetos Tradicional para comparar su eficacia en comparación con el DOM.
- ✓ Realizar scripts en los que acceda a los elementos del documento web utilizando el Modelo de Objetos Tradicional.
- ✓ Adquirir dominio en la utilización de los principales métodos que ofrece JavaScript para trabajar con objetos JSON.
- ✓ Comprender las opciones para trabajar con almacenamiento local tanto con localStorage como con sessionStorage entendiendo sus diferencias.
- ✓ Utilizar las propiedades y métodos para el trabajo con los objetos localStorage y sessionStorage.

■ Presentación

Los modelos de objetos definen la interfaz para los diversos aspectos del navegador y del documento web que podrán controlarse haciendo uso de JavaScript. En JavaScript se emplean dos modelos de objetos fundamentales, que son: **Modelo de Objetos de Navegador (BOM)** y **Modelo de Objetos de Documento (DOM)**. **BOM (Browser Object Model)** proporciona acceso a las diversas características del navegador. Por ejemplo, la ventana del navegador, las características de la pantalla, el historial del navegador, etc. Entre tanto, **DOM (Document Object Model)**, proporciona acceso al contenido de la ventana del navegador; es decir, el documento web que se muestra (aunque en algunos casos puede ser el conjunto de marcos que forman la página) incluidos los diversos elementos HTML que estén presentes en él, como enlaces, imágenes, tablas, etc. Uno de los problemas principales es que la división entre el BOM y el DOM es poco clara. Además, las implementaciones de JavaScript de los diferentes navegadores varían significativamente.

JSON es una manera más simple de representar objetos de JavaScript como cadenas, es una alternativa más simple a XML para pasar datos entre el cliente y el servidor. En JSON, cada objeto se representa como una lista de nombres de propiedad y valores contenidos entre llaves, en el siguiente formato: `{"NombrePropiedad":valor1,"NombrePropiedad2":valor2}`. Los arreglos en json se representan en JSON mediante corchetes en el siguiente formato: `[valor1,valor2,valor3]`. Cada valor puede ser una cadena, un número, una representación de JSON de un objeto, true, false o null. Las aplicaciones Web pueden usar la propiedad **localStorage** del objeto Windows para almacenar hasta varios megabytes de datos de cadena de pares clave/valor en la computadora del usuario y pueden acceder a esos datos entre sesiones de navegación y pestañas del navegador. Las aplicaciones Web necesitan acceso a los datos solo para una sesión de navegación y que deben mantener esos datos separado entre múltiples pestañas pueden usar la propiedad **sessionStorage** del objeto Windows.

■ Situación problemática

Los estudiantes realizarán la guía desarrollando los ejemplos y la tarea complementaria.

■ Contenido

Para esta semana se comparten los siguientes materiales:

Material 1: Guía 09 [PDF].

Material 2: Guía 10 [PDF].

■ Actividad(es):

En la semana 11/12 se realizará la siguiente actividad:

1. Análisis de resultado – Guía 09.
2. Análisis de resultado – Guía 10.

■ Recursos

1. C #: Manual de Programación. Autores: Luis Joyanes Aguilar y Matilde Fernández Azuela Publisher: Madrid, España: McGraw-Hill, 2002 Clasificación: Libro 005.362 J88 2002.
2. Aprenda ya Microsoft Visual C#.NET. Autores: Sharp, John Jagger, Jon Coautor Publisher: Madrid, España: McGraw-Hill, 2002. Clasificación: Libro 005.362 S581 2002.
3. WARGO, J (2013). Apache Cordova 3 Programming, Addison-Wesley.

■ Horario de Consultas:

Para resolver dudas sobre los contenidos o las asignaciones a desarrollar, puedes contactarme al **correo institucional**: aida.quintanilla@udb.edu.sv o si no al número de **WhatsApp**: 7034 – 8514.

Pero si necesitan conectarse por medio de una videoconferencia lo podemos hacer en la plataforma **Google Meets** los días Lunes de 8:00 am a 5:00 pm, Martes de 10:00 am a 5:00 pm, Miércoles de 11:00 am a 5:00 pm y Sábados de 8:00 am a 1:00 pm. **Domingos no se dará consultas.**