Guía de examen de la primera unidad de Programación Web Avanzada

Autor: Antonio Emiko Ochoa Adame

Disclaimer

La finalidad de este documento es servir como apoyo de estudio. El autor de la versión original de este documento no se hace responsable del uso indebido del mismo.

1.1 DOM (Document Object Model)

Es una API (Application Programming Interface) para documentos HTML válidos.

Qué permite hacer el DOM?

- Construir documentos
- Navegar por su estructura
- Modificar
- Eliminar

Cómo funciona?

- Cada etiqueta HTML es un objeto que puede tener hijos y/o padres
- Todo el DOM es accesible desde JavaScript
- Es una herramienta que te permite la modificación dinámica del contenido

Tipos de nodos

Existen **12** pero se usan **5** principalmente:

- Documento nodo raíz
- Element: Etiquetas HTML que pueden ser padres e hijos
- Attr: Nodos de atributos para etiquetas Element
- Text: Texto dentro de etiquetas Element
- Comment: Nodos de comentarios en HTML

Operaciones básicas

Obtener

- getElementsBytagName(): Obtiene todos los elementos de una cierta etiqueta en el DOM
- getElementsByClassNmae(): Obtener todos los elementos con una clase específica
- getElementById(): Obtener un elemento con id particular

Cambiar

- element.innerHTML = nuevo HTML: Cambiar interior de un elemento
- element.attribute = nuevo valor: Cambiar el valor del atributo
- element.style.property = nuevo estilo: Cambiar estilo
- element.setAttribute(atributo, valor): Colocar nuevo atributo

Crear y eliminar

- document.createElement(element): Crear elemento de document
- document.removeChild(element): Eliminar hijo
- document.appendChild(element): Añadir hijo
- document.replaceChild(nuevo, viejo): Remplazar HTML

Eventos

Un **evento** es un suceso en un programa que ejecuta alguna acción pre programada al ser detectado.

Ejemplos:

- Usuario da click en un botón
- Usuario cambia el contenido de un input
- Usuario muevo el puntero

Uso de eventos

Incrustado en HTML:

```
<button onclick="nombreFuncion()">
    Presioname
</button>
```

Directo en el DOM:

```
document.getElementById("boton").onclick = nombreFuncion;
```

Tipos de eventos

Eventos de teclado:

- Onblur
- Onchange
- OnFocus
- Onkeydown
- Onkeyup
- Onkeypress

Eventos de mouse:

- Onmouseover
- Onmousedown
- Onmouseup

Eventos de mouse:

- Onclick
- Ondblclick

Eventos de carga

1.2 MVC(Modelo Vista Controlador)

Patrón de diseño: Solución probada, expresiva y fácil de mantener para resolver problemas comunes en el desarrollo de software.

Arquitectura de software: se compone de las esctructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos.

Qués es MVC?

Es un patrón de diseño que separa la lógica de la aplicación, la vista y los datos.

Ventajas:

- Mantenible a largo plazo
- Pauta para desarrollo colaborativo
- Estadarización de código
- Cambios independientes de acuerdo al problema

Modelo

Representación de los datos con los que la aplicación va a interactuar.

Implica:

- Consulta a la base de datos
- Consulta a APIs externas
- CRUD de datos

Vista

Representación gráfica de los datos al usuario

Implica:

- Mostrar los datos al usuario final
- Otorgarle una intefaz al usuario para ingresar datos
- Comunicación directa con el controlador

Controlador

Gestiona el flujo de información entre el modelo y la vista.

Implica:

- Contestar peticiones del usuario (vista)
- Responde información estructurada por el modelo

Ejemplo de petición en un MVC

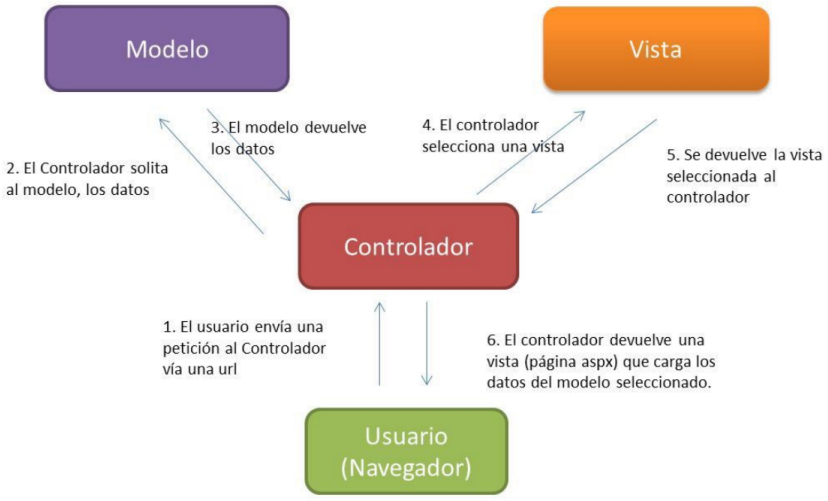

Figure 1: Ejemplo de peticion

Fundamentos de MVC

- Convención sobre configuración: Precuparse por programar en lugar de configurar
- No repetir (DRY): Mantener código centralizado

Patrones derivados de MVC

- MVP (Modelo Vista Presentador)
- MVA (Modelos Vista Adaptador)
- MVVM (Modelo Vista VistaModelo)

Qué es MVVM? (Modelo Vista VistaModelo)

Derivado de MVC que separa la UI de todo lo demás

Características:

- Facilita Unit Tests
- Separa la lógica de negocio de la vista

1.3 Webs

Hacen referencia a la evolución de la forma de interacción y formas de transmisión de información del usuario con la red (web)

Web 1.0

- Primera versión de la web
- Surge en los 60s con HTML
- Netscape

Características:

- Contenido estática
- Solo consumo de información
- Sin interacción del usuario final
- Diseño pobre
- Unidireccional (el web master era el único con intereacción directa)

Ejemplo: La página de la commodore amiga

Web 2.0

Es una segunda generación de modelos de páginas web, así como cambios en la forma de compartir información.

Características:

- Contenido dinámico
- El usuario crea contenido (aparte de consumir)
- Diseño más amable para el usuario
- Personas conectando con personas
- Redes sociales

Ejemplo: Metroflog

Web 3.0

Es la wbe 2.0 con características extra intentando se más intuitiva y fluida.

Características:

- Sitios multiservidor
- Web multiplataforma (responsive)
- Web semántica (con significado)
- AI
- IOT
- Realidad aumentada

Ejemplo: facebook actual?

Web semántica:

- Búsquedas más humanas
- Respuestas variables
- Capacidad de "razonar"

Contenido accesible sin tanta navegación:

- Centralizado
- Rápido
- Concreto

Ejemplos de contenido accesible:

- Marcadores
- Evernote

Web semántica en HTML5

Es darle significado a la maquetación.

Ventajas:

- Ayuda al SEO-
- Estructura que se entienda por el proggramador y los navegadores
- Mantenible

- Lectores de pantalla mejoran su función

Ejemplos de etiquetas semánticas:

- main
- header
- footer
- section
- article
- nav
- aside
- hgroup
- figure
- figcaption
- video
- audio
- canvas