

Instituto Politécnico Nacional

Unidad Profesional Interdisciplinaria de Ingeniería
Campus Tlaxcala (UPIIT).



Curso: Tecnologías para el desarrollo
de aplicaciones web.

Tarea 10.

AJAX

Alumno: Omar Emiliano León Castro.

Docente: Jesús Hernández Rojas.

Fecha de entrega: 16 de noviembre
del 2023.

Investigación

AJAX (Asynchronous JavaScript and XML) es una técnica de desarrollo web que permite actualizar partes de una página web sin necesidad de recargarla por completo. Fue popularizado por Google en 2005 con el lanzamiento de Gmail, y desde entonces se ha convertido en una parte integral de muchas aplicaciones web modernas.

JavaScript es un lenguaje de programación muy conocido. Entre otras funciones, gestiona el contenido dinámico de un sitio web y permite la interacción dinámica del usuario. XML es otra variante de un lenguaje de marcado como [HTML](#), como lo sugiere su nombre: eXtensible Markup Language. Mientras HTML está diseñado para mostrar datos, XML está diseñado para contener y transportar datos.

Principales aspectos de AJAX:

1. Asynchronous (Asíncrono):

- **Asincronía:** En el contexto de AJAX, asincronía significa que las solicitudes al servidor web pueden ocurrir en segundo plano sin afectar la visualización o el comportamiento de la página web. Esto permite que otras partes de la página sigan funcionando mientras se espera una respuesta del servidor.

2. JavaScript:

- **Lenguaje de Programación:** AJAX utiliza principalmente JavaScript para realizar solicitudes asíncronas al servidor y manipular el contenido de la página web.
- **Eventos:** Se basa en eventos del lado del cliente para gestionar las respuestas del servidor y actualizar dinámicamente la interfaz de usuario.

3. XML (Extensible Markup Language):

- **Formato de Datos Inicial:** En sus primeras implementaciones, AJAX utilizaba XML como formato de intercambio de datos entre el cliente y el servidor. Sin embargo, en la actualidad, se utilizan formatos de datos más ligeros y eficientes, como JSON (JavaScript Object Notation), en lugar de XML.

4. Solicitud y Respuesta Asíncrona:

- **Solicitud Asíncrona:** Se envían solicitudes al servidor web desde el cliente sin recargar la página completa.
- **Respuesta Asíncrona:** El servidor responde con datos que pueden ser procesados por el cliente para actualizar la interfaz de usuario.

5. Objeto XMLHttpRequest:

- **API Fundamental:** AJAX se implementa principalmente a través del objeto XMLHttpRequest en JavaScript, que proporciona la funcionalidad para realizar solicitudes HTTP asíncronas.

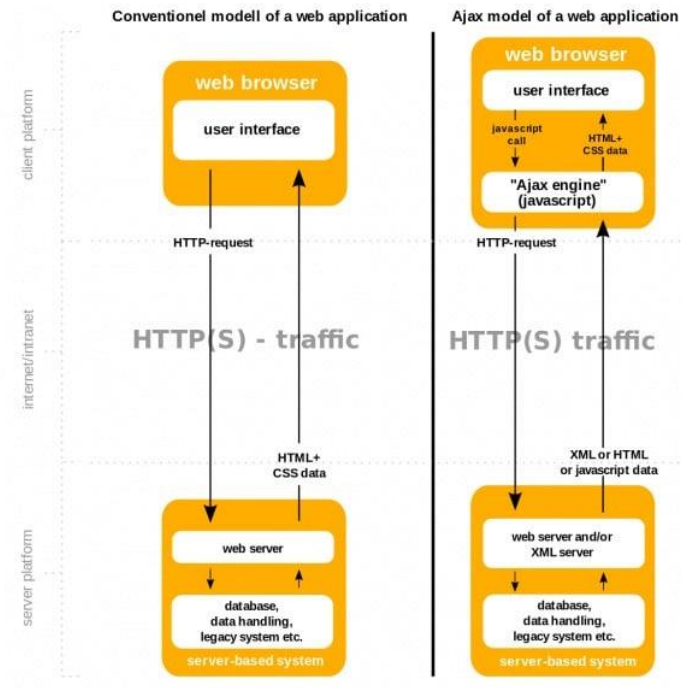
- **Manipulación de Datos:** Permite enviar datos al servidor y recibir respuestas sin recargar la página.
- 6. **Ejemplo de Uso:**
 - **Evento Onreadystatechange:** Un ejemplo sencillo podría ser una solicitud para verificar la disponibilidad de un nombre de usuario en un formulario de registro. El evento onreadystatechange del objeto XMLHttpRequest se activa cuando cambia el estado de la solicitud, y el contenido de la página puede actualizarse en consecuencia.
- 7. **Frameworks y Bibliotecas:**
 - **jQuery:** Antes, el uso de AJAX podía ser complicado debido a las diferencias en la implementación de los navegadores. Bibliotecas como jQuery simplificaron el proceso y se convirtieron en herramientas populares para trabajar con AJAX.
- 8. **Ventajas de AJAX:**
 - **Experiencia del Usuario Mejorada:** Al evitar recargas completas de la página, se mejora la velocidad y la experiencia del usuario.
 - **Eficiencia en la Transmisión de Datos:** Al enviar solo datos necesarios en lugar de la página completa, se reduce el ancho de banda necesario.

Ajax se compone de las siguientes tecnologías:

- XHTML y CSS para presentar información.
- DOM (Document Object Model - modelo de objetos de documento) para visualizar e interactuar de forma dinámica la información presentada.
- El objeto XMLHttpRequest para manipular los datos de forma asíncrona con el servidor web.
- XML, HTML y XSLT para el intercambio y la manipulación de datos.
- Se visualiza JavaScript para enlazar solicitudes e información de datos.

¿Cómo funciona AJAX?

Modelo convencional	Modelo AJAX
1. Se envía una solicitud HTTP desde el navegador web al servidor.	1. El navegador crea una llamada de JavaScript que luego activará XMLHttpRequest.
2. El servidor recibe y, posteriormente, recupera los datos.	2. En segundo plano, el navegador web crea una solicitud HTTP al servidor.
3. El servidor envía los datos solicitados al navegador web.	3. El servidor recibe, recupera y envía los datos al navegador web.
4. El navegador web recibe los datos y vuelve a cargar la página para que aparezcan los datos.	4. El navegador web recibe los datos solicitados que aparecerán directamente en la página. No se necesita recargar.



Desarrollo de ejemplo

Tenemos un archivo HTML con un botón y un contenedor donde mostramos los datos:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo AJAX</title>
  <script src="script.js" defer></script>
</head>
<body>

  <h1>Contenido Dinámico con AJAX</h1>

  <button onclick="cargarDatos()">Cargar Datos</button>

  <div id="datos-container">
    <!-- Aquí se mostrarán los datos cargados -->
  </div>

</body>
</html>
```

```
function cargarDatos() {  
    // Crear una instancia del objeto XMLHttpRequest  
    var xhttp = new XMLHttpRequest();  
  
    // Configurar la solicitud con el método HTTP y la URL del servidor  
    xhttp.open("GET", "datos.txt", true);  
  
    // Definir la función que se ejecutará cuando la solicitud se complete  
    xhttp.onreadystatechange = function () {  
        // Verificar si la solicitud se completó con éxito (estado 4) y el  
        // código de estado HTTP es 200 (OK)  
        if (this.readyState == 4 && this.status == 200) {  
            // Actualizar el contenido del contenedor con los datos  
            // recibidos del servidor  
            document.getElementById("datos-container").innerHTML =  
this.responseText;  
        }  
    };  
  
    // Enviar la solicitud al servidor  
    xhttp.send();  
}
```

Cuando haces clic en el botón, se ejecuta la función `cargarDatos()`, que utiliza AJAX para obtener los datos del archivo y los muestra en el contenedor especificado.

Resultados





Conclusión

En resumen, AJAX ha revolucionado el desarrollo web al proporcionar una manera eficiente y asincrónica de actualizar contenido en una página sin recargarla por completo. Surgió como una respuesta a la necesidad de mejorar la experiencia del usuario en aplicaciones web, permitiendo la actualización dinámica de datos sin interrupciones perceptibles.

El corazón de AJAX radica en su capacidad para realizar solicitudes asíncronas al servidor mediante el objeto XMLHttpRequest de JavaScript. Aunque inicialmente utilizaba XML como formato de intercambio de datos, en la actualidad, se prefieren formatos más livianos como JSON.

El ejemplo proporcionado ilustra cómo implementar AJAX en una página HTML básica. Al hacer clic en un botón, se activa una solicitud al servidor, y los datos recuperados se actualizan dinámicamente en la página sin necesidad de recargarla. Este enfoque ha mejorado significativamente la eficiencia y la velocidad de las aplicaciones web, proporcionando una experiencia del usuario más fluida.