



Clase 12

Análisis de Sistemas

Materia:
Programación Web II

Docente contenidista: ROLDÁN, Hernán

Revisión: Coordinación

Contenido

AJAX: Asynchronous JavaScript and XML.....	3
¿Qué es AJAX?.....	3
Finalidad y ventajas de AJAX.....	4
Uso de AJAX	5
Ejemplo de uso de AJAX	6
1. Creación del Archivo HTML (index.html)	9
2. Creación del Archivo JavaScript (script.js)	9
3. Cargar la Lista de Países al Cargar la Página	10
4. Respuesta del Servidor y Actualización del Combo de Países	10
Bibliografía	13
Para ampliar la información	13

Clase 12



¡Te damos la bienvenida a la materia
Programación Web II!

En esta clase vamos a ver los siguientes temas:

- ¿Qué es AJAX?
- Finalidad y ventajas de AJAX.
- Uso de AJAX.
- Ejemplo de uso de AJAX.

AJAX: Asynchronous JavaScript and XML

¿Qué es AJAX?

AJAX es una técnica de programación que permite realizar comunicaciones asíncronas entre el navegador web y el servidor, sin tener que recargar toda la página.

La idea principal detrás de AJAX es mejorar la experiencia del usuario al proporcionar interacciones rápidas y dinámicas en una página web, evitando la necesidad de recargar completamente la página cada vez que se necesita actualizar o enviar datos al servidor.

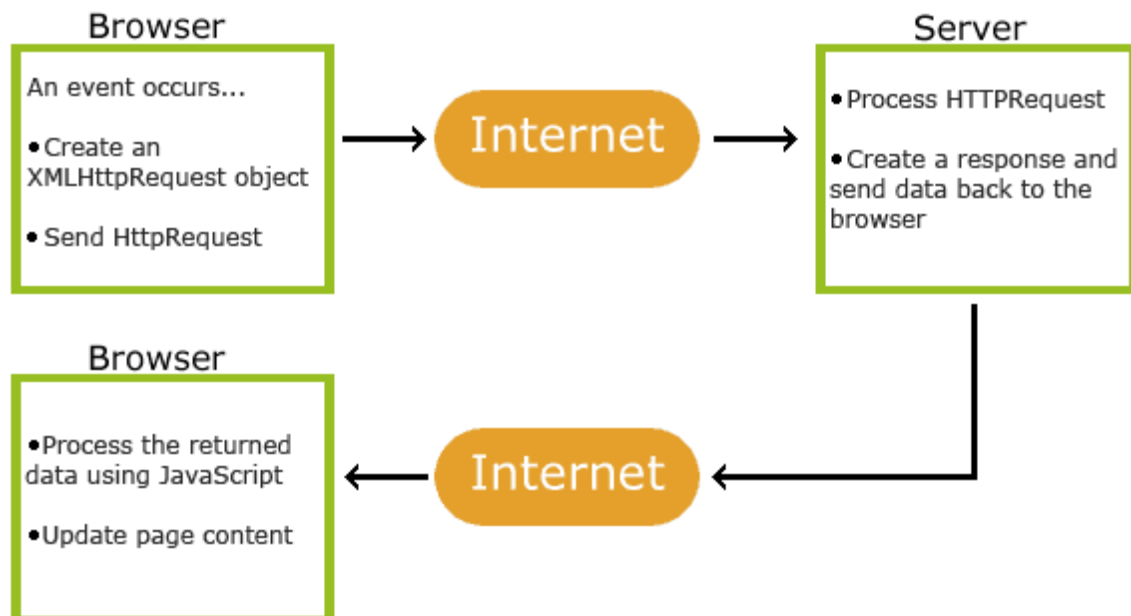


Imagen extraída de:

https://www.apuntesinformaticafp.com/cursos/uso_de_ajax.html

Finalidad y ventajas de AJAX

La finalidad principal de AJAX es mejorar la interacción del usuario con la aplicación web y proporcionar una experiencia más rápida y fluida.

Al utilizar AJAX, las páginas pueden actualizar de forma parcial, permitiendo cambios dinámicos sin necesidad de recargar la página completa.

Algunas de las ventajas de AJAX son:

- **Interactividad mejorada:** Los usuarios pueden realizar acciones sin experimentar interrupciones o tiempos de espera prolongados.
- **Respuestas más rápidas:** Como solo se actualizan partes específicas de la página, las respuestas del servidor son más rápidas y eficientes.
- **Reducción del consumo de datos:** Al no tener que cargar toda la página nuevamente, se reduce la cantidad de datos transferidos, lo que ahorra ancho de banda.
- **Mejora de la experiencia del usuario:** Las aplicaciones web se sienten más como aplicaciones de escritorio, lo que mejora la satisfacción del usuario.
- **Integración con API y servicios web:** **AJAX** permite interactuar con API y servicios web de forma sencilla, lo que amplía las capacidades de una aplicación.

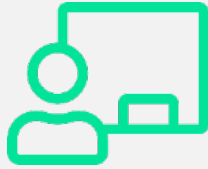
Uso de AJAX

El uso de **AJAX** implica la combinación de tecnologías como JavaScript, HTML, CSS y el uso de objetos XMLHttpRequest (XHR) o la API Fetch en navegadores modernos.

A continuación, se presenta el paso a paso de cómo se utiliza **AJAX**:

- 1. Crear un contenedor HTML para mostrar la respuesta:**
Definimos un elemento HTML (por ejemplo, un div) en nuestra página donde deseamos mostrar la respuesta del servidor.
- 2. Capturar eventos en JavaScript:** Utilizamos JavaScript para capturar eventos, como hacer clic en un botón, enviar un formulario o cualquier otra interacción del usuario que deseemos utilizar para desencadenar la acción AJAX.
- 3. Crear una función AJAX:** Creamos una función en JavaScript que realice la comunicación con el servidor utilizando objetos XMLHttpRequest o la API Fetch (navegadores modernos).
- 4. Configurar la solicitud AJAX:** Dentro de la función AJAX, configuramos la solicitud según lo que necesitemos enviar o recibir del servidor. Podemos especificar el método HTTP (GET o POST), la URL del servidor y los datos a enviar.
- 5. Enviar la solicitud al servidor:** Enviamos la solicitud al servidor utilizando la función send() en el objeto XHR o la función fetch().
- 6. Procesar la respuesta del servidor:** Manejamos la respuesta del servidor una vez que se haya completado la solicitud. Podemos actualizar el contenido del contenedor HTML definido en el paso 1 con la respuesta recibida.

Ejemplo de uso de AJAX



A continuación, crearemos el código que nos permita cargar los países en un combo (select) utilizando AJAX sin necesidad de utilizar una base de datos. Utilizaremos una API pública que proporciona información de países en formato JSON para obtener la lista de países (si no estás familiarizado con JSON, visitá la clase 9).

Creemos un archivo `index.html` que contenga el combo para seleccionar el país

```
<!DOCTYPE html>
<html>
<head>
  <title>Selección de País</title>
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.mi
n.css">
  <style>
    .cursor-pointer{
      cursor: pointer;
    }
  </style>
</head>
<body>
  <div class="container mt-5">
    <h1>Selección de País</h1>
    <select id="países" class="form-control cursor-pointer mt-3"
style="width: 200px;">
      <option value="" selected>Seleccionar país</option>
    </select>
  </div>

  <script src="script.js"></script>
</body>
</html>
```

Creemos un archivo JavaScript llamado `script.js` que manejará la carga de los países mediante AJAX:

```
// Obtener el combo de países
const comboPaíses = document.getElementById("países");

// Función para cargar los países mediante AJAX
function cargarPaíses() {
  // Crear objeto XMLHttpRequest
```

```

const xhr = new XMLHttpRequest();

// Configurar la solicitud AJAX
xhr.open("GET", "https://restcountries.com/v3.1/all", true);

// Manejar la respuesta del servidor
xhr.onreadystatechange = function () {
    if (xhr.readyState === XMLHttpRequest.DONE) {
        if (xhr.status === 200) {
            // Convertir la respuesta JSON en un array de objetos
            const paises = JSON.parse(xhr.responseText);

            // Recorrer el array de países y agregar cada país al
combo

            paises.forEach(function (pais) {
                const opcion = document.createElement("option");
                opcion.value = pais.name.common;
                opcion.textContent = pais.name.common;
                comboPaises.appendChild(opcion);
            });
        } else {
            console.error("Error al cargar la lista de países");
        }
    }
};

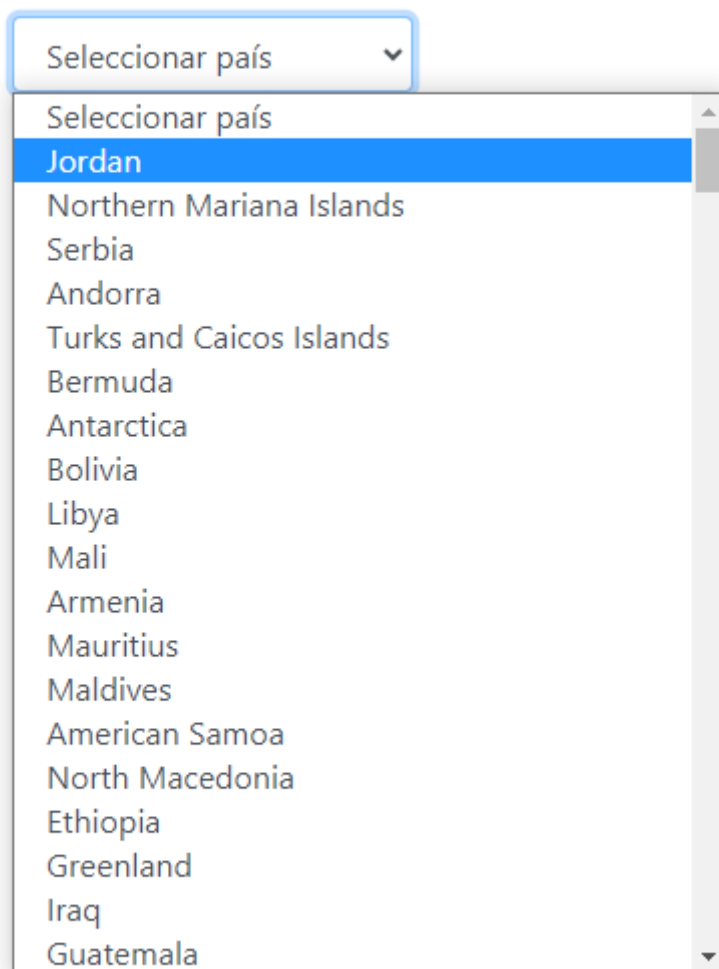
// Enviar la solicitud al servidor
xhr.send();
}

// Cargar la lista de países al cargar la página
cargarPaises();

```


Vista en el navegador:

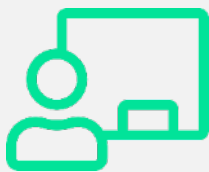
Selección de País



Selección de País

Seleccionar país ▼

- Seleccionar país
- Jordan**
- Northern Mariana Islands
- Serbia
- Andorra
- Turks and Caicos Islands
- Bermuda
- Antarctica
- Bolivia
- Libya
- Mali
- Armenia
- Mauritius
- Maldives
- American Samoa
- North Macedonia
- Ethiopia
- Greenland
- Iraq
- Guatemala



Explicación paso a paso...

1. Creación del Archivo HTML (index.html)

- En este paso, creamos un archivo HTML que contiene un elemento select (combo) donde se mostrará la lista de países. También incluimos un elemento script que enlaza al archivo JavaScript donde manejaremos la carga de los países mediante AJAX.

2. Creación del Archivo JavaScript (script.js)

- En este paso, creamos un archivo JavaScript que se encargará de manejar la carga de los países mediante AJAX.
- Obtenemos el elemento select del combo de países utilizando getElementById.
- Definimos una función llamada cargarPaises para realizar la solicitud AJAX y cargar la lista de países.
- Creamos un objeto XMLHttpRequest utilizando new XMLHttpRequest() para realizar la solicitud asíncrona al servidor.
- Configuramos la solicitud utilizando el método open, donde especificamos el método HTTP (GET, en este caso) y la URL de la API pública que proporciona la información de países (<https://restcountries.com/v3.1/all>).
- Añadimos un evento onreadystatechange para manejar la respuesta del servidor. Cuando el estado del objeto XMLHttpRequest cambia a DONE (4), verificamos que el estado de la respuesta sea 200, lo que indica que la solicitud se completó correctamente.
- Si la respuesta es satisfactoria, convertimos la respuesta JSON en un array de objetos mediante JSON.parse.
- Recorremos el array de países utilizando forEach y, por cada país, creamos un nuevo elemento option y lo agregamos al select con appendChild. Así, cada país se convierte en una opción dentro del combo de países.

3. Cargar la Lista de Países al Cargar la Página

- Al final del archivo JavaScript, llamamos a la función cargar Países. Esto se hace para que, al cargar la página, se realice automáticamente la solicitud AJAX para obtener la lista de países y mostrarla en el combo.

4. Respuesta del Servidor y Actualización del Combo de Países

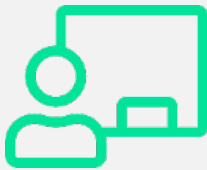
- Cuando la solicitud AJAX se completa, el navegador recibe la respuesta del servidor y ejecuta el código definido en la función onreadystatechange.
- El código convierte la respuesta JSON en un array de objetos que contiene los países y sus detalles.
- Luego, mediante el uso de forEach, recorremos el array y creamos un nuevo elemento option por cada país recibido.
- Finalmente, agregamos cada opción al combo de países mediante appendChild.

Con estos pasos, hemos logrado cargar dinámicamente la lista de países en el combo utilizando **AJAX**.

Cuando el usuario carga la página, la lista de países se obtiene del servidor y se muestra en el combo, lo que mejora la interactividad y experiencia del usuario, ya que la página no necesita recargarse por completo para obtener esta información.

En este ejemplo, estamos utilizando la API pública <https://restcountries.com/v3.1/all> , que proporciona información de todos los países en formato JSON.

Al cargar la página, el archivo JavaScript script.js realizará una solicitud AJAX para obtener la lista de países y los agregará como opciones al combo (select) en el archivo index.html.



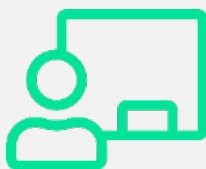
Tené en cuenta que este ejemplo utiliza una API pública para obtener la información de los países. En aplicaciones reales, es posible que desees utilizar una API diferente o una base de datos para obtener datos específicos de países, pero el enfoque de utilizar AJAX para cargar la información dinámicamente en la página es similar.

En resumen, AJAX (Asynchronous JavaScript and XML) es una potente técnica de programación que nos permite mejorar la interactividad y experiencia del usuario en aplicaciones web. Al utilizar AJAX, podemos realizar comunicaciones asíncronas entre el navegador y el servidor, lo que nos permite actualizar contenido en la página sin tener que recargarla por completo.



Hemos llegado así al final de esta clase en la que vimos:

- ¿Qué es AJAX?
 - Finalidad y ventajas de AJAX.
 - Uso de AJAX.
- Ejemplo de uso de AJAX.



Te esperamos en la **clase en vivo** de esta semana.
No olvides realizar el **desafío semanal**.

¡Hasta la próxima clase!

Bibliografía

[Documentación Oficial de PHP](#)

[W3Schools: PHP Tutorial](#)

Cabezas Granado, Luis. González Lozano, F. (2018) Desarrollo web con PHP y MySQL. Anaya Multimedia.

Heurtel, O. (2016) Desarrolle un sitio web dinámico e interactivo. Eni Ediciones.

Welling, Luke Thompson, L. (2017) Desarrollo Web con PHP y MySQL. Quinta Edición (2017, Editorial Anaya).

Para ampliar la información

[PHP: AJAX Introduction](#)

[Tutorial AJAX 1 - introducción a AJAX | ¿Qué es AJAX? | Ajax desde Cero](#)

[PHP: The Right Way](#)

[Todo sobre PHP](#)

[PHP Programming Language Tutorial - Full Course](#)

[PHP Full Course for non-haters](#)

[CURSO de php desde cero](#)

[MDN: Introducción al lado Servidor](#)

[Guía de HTML](#)