

# RESUMEN AJAX

## ¿Qué es AJAX?

JavaScript y XML asíncronos (AJAX) desarrollan aplicaciones web altamente interactivas y responsivas mediante el uso de un conjunto de tecnologías. Permite a los desarrolladores actualizar componentes individuales de la página web en tiempo real sin tener que recargar toda la página.

Antes de AJAX, cada vez que era necesario cambiar o modificar una parte del contenido de una página web, el proceso recargaba y actualizaba toda la página desde el servidor web. Como resultado, este método generaba grandes cantidades de datos duplicados en cada instancia. Este problema se resolvió con la introducción de AJAX, que permitió actualizar los componentes web de forma asíncrona sin recargar toda la página. Este método garantizaba que las actualizaciones se mostraran casi instantáneamente a los usuarios finales.

AJAX se basa en las siguientes tecnologías para realizar un intercambio de datos fluido entre un navegador y un servidor web:

- XHTML y CSS: Presentación de datos
- Modelo de objetos de documento HTML (DOM): permite la interacción dinámica del usuario mientras se muestra el contenido actualizado
- Objeto XMLHttpRequest: mover, manipular y actualizar datos hacia y desde el servidor
- XML, HTML, XSLT y JSON : Gestionar el intercambio de datos con el servidor
- JavaScript: Acopla las solicitudes de datos con la visualización del contenido en el lado del usuario

## Principales casos de uso de Ajax

### Carga dinámica de contenido:

Ajax se usa habitualmente para cargar contenido dinámico en una página web sin necesidad de recargarla por completo. Esto resulta especialmente útil para actualizar partes de una página, como comentarios, mensajes de chat o noticias.

### Envío y validación del formulario:

Ajax permite enviar datos de formularios al servidor sin recargar la página completa. Se utiliza frecuentemente para la validación de formularios y para proporcionar información en tiempo real a los usuarios.

### **Autocompletar y sugerencias:**

Las funciones de autocompletar en los cuadros de búsqueda o campos de entrada se benefician de Ajax. A medida que los usuarios escriben, se realizan solicitudes Ajax al servidor para obtener y mostrar sugerencias relevantes en tiempo real.

### **Desplazamiento infinito:**

Los sitios web que implementan el desplazamiento infinito, donde se carga contenido nuevo a medida que el usuario se desplaza hacia abajo, a menudo utilizan Ajax para obtener y agregar datos adicionales sin actualizar toda la página.

### **Actualización de interfaces de usuario:**

Ajax se utiliza para actualizar diversos elementos de una interfaz de usuario, como paneles de control, tablas y gráficos, con nuevos datos obtenidos del servidor.

### **Aplicaciones de página única (SPA):**

Las SPA dependen en gran medida de Ajax para cargar contenido dinámicamente a medida que los usuarios navegan por la aplicación. Esto se traduce en una experiencia de usuario más fluida y con mejor capacidad de respuesta.

### **Búsqueda y filtrado en vivo:**

Ajax se utiliza en las funciones de búsqueda en vivo, donde los resultados se actualizan en tiempo real a medida que el usuario escribe. También se emplea para filtrar y ordenar datos sin recargar la página completa.

### **Mapas interactivos:**

Los sitios web que incorporan mapas interactivos suelen utilizar Ajax para obtener datos cartográficos de forma dinámica en función de las interacciones del usuario, lo que permite una experiencia de mapas más dinámica y con mejor capacidad de respuesta.

### **Colaboración en tiempo real:**

Las aplicaciones colaborativas, como las herramientas de edición colaborativa de documentos o de gestión de proyectos, utilizan Ajax para actualizar el contenido en tiempo real para todos los usuarios involucrados.

### **Autenticación de usuario:**

Ajax se utiliza comúnmente en los procesos de autenticación de usuarios, permitiendo a los usuarios iniciar o cerrar sesión sin necesidad de actualizar toda la página.

### **Notificaciones y alertas:**

Ajax se utiliza para actualizar y mostrar notificaciones o alertas en tiempo real, informando a los usuarios sobre nuevos mensajes, eventos o actualizaciones.

## Actualizaciones meteorológicas y precios de las acciones:

Los sitios web que muestran datos en tiempo real, como actualizaciones meteorológicas, precios de acciones o resultados deportivos, a menudo utilizan Ajax para obtener y actualizar la información sin recargar toda la página.

Ajax mejora significativamente la experiencia del usuario en la web al proporcionar una interfaz más dinámica y con mayor capacidad de respuesta. Se ha convertido en una tecnología fundamental en el desarrollo web moderno, y sus principios se integran a menudo en frameworks y bibliotecas más avanzadas, como jQuery, React o Vue.js.

## Funcionamiento

### Interacción del usuario:

El usuario interactúa con un elemento de la página web, desencadenando un evento (por ejemplo, haciendo clic en un botón).

### Solicitud JavaScript:

El código JavaScript dentro de la página web crea un objeto XMLHttpRequest.

Este objeto envía una solicitud al servidor, especificando la URL, el método (por ejemplo, GET, POST) y cualquier dato que se vaya a enviar.

### Procesamiento del servidor:

El servidor recibe la solicitud y la procesa, normalmente utilizando un script del lado del servidor (por ejemplo, PHP, Python).

El servidor genera una respuesta, que puede ser XML, JSON o texto plano.

### Manejo de respuestas:

La respuesta se envía de vuelta al navegador.

El código JavaScript en el navegador recibe la respuesta y la analiza.

### Actualización dinámica:

El código JavaScript actualiza la página web en función de los datos recibidos, utilizando métodos como la manipulación del DOM.

Esto permite realizar actualizaciones parciales de la página sin recargar todo el contenido.

## Arquitectura

### Lado del cliente:

Código JavaScript: Inicia solicitudes, maneja respuestas y actualiza la interfaz de usuario.

Objeto XMLHttpRequest: Responsable de realizar solicitudes asíncronas al servidor.

## Lado del servidor:

Script del lado del servidor: Procesa solicitudes, genera respuestas e interactúa con bases de datos u otros recursos.

Servidor web: Recibe solicitudes, envía respuestas y se comunica con el cliente.

## Ejemplo

Página HTML

```
<!DOCTYPE html>
<html>
<body>

<div id="demo">
    <h2>Let AJAX change this text</h2>
    <button type="button" onclick="loadDoc()">Change
Content</button>
</div>

</body>
</html>
```

La página HTML contiene una sección `<div>` y un `<button>`.

La sección `<div>` se utiliza para mostrar información de un servidor.

El `<botón>` llama a una función (si se hace clic en él).

La función solicita datos a un servidor web y los muestra:

```
function loadDoc() {
    const xhttp = new XMLHttpRequest();
    xhttp.onload = function() {
        document.getElementById("demo").innerHTML = this.responseText;
    }
    xhttp.open("GET", "ajax_info.txt", true);
    xhttp.send();
}
```