

# UT9. Generación dinámica de páginas web interactivas

DAW - Desarrollo Web Entorno Servidor  
Fernando Galindo



1. Introducción
2. PHP y JavaScript
3. Integración de AJAX con PHP

El desarrollo de aplicaciones Web tiene dos grandes áreas de trabajo:

- Desarrollo en el lado de Servidor
- Desarrollo en el lado del Cliente

Las aplicaciones Web normalmente se desarrollan utilizando ambas técnicas, por lo que es el momento de integrar ambas áreas de trabajo en las aplicaciones que desarrollamos

El servidor web, deberá ser capaz de integrar ambos códigos

- Código será generado dinámicamente por el servidor (PHP)
- Código será generado dinámicamente por el cliente (JavaScript)

```
<?php
    //Generación de código servidor
    echo "Saludos desde PHP<br>";
?>
<script type="text/javascript">
    //Generación de código cliente
    document.write("Saludos desde JavaScript");
</script>
```

```
Saludos desde PHP
Saludos desde JavaScript
```

Como sabéis existen complementos y librerías adicionales de JavaScript que podemos incorporar para realizar programación en el lado del cliente

Con el fin de unificar las prácticas utilizaremos la misma librería de JQuery

Podemos incorporar la siguiente línea para importar la librería

```
<script type="text/javascript"  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jq  
uery.min.js"></script>
```

Utilizar ambos lenguajes de servidor y cliente nos permite elegir qué programamos en cada parte de la aplicación

Cada lenguaje aporta ventajas al cliente y a la navegación, los códigos en el cliente se ejecutan más rápidos y no se consumen recursos del servidor

Ejemplo

*Creamos una aplicación utilizando JavaScript, formulario, validación, etc.*

*Utilizamos la programación del lado de servidor para el almacenamiento de los datos*

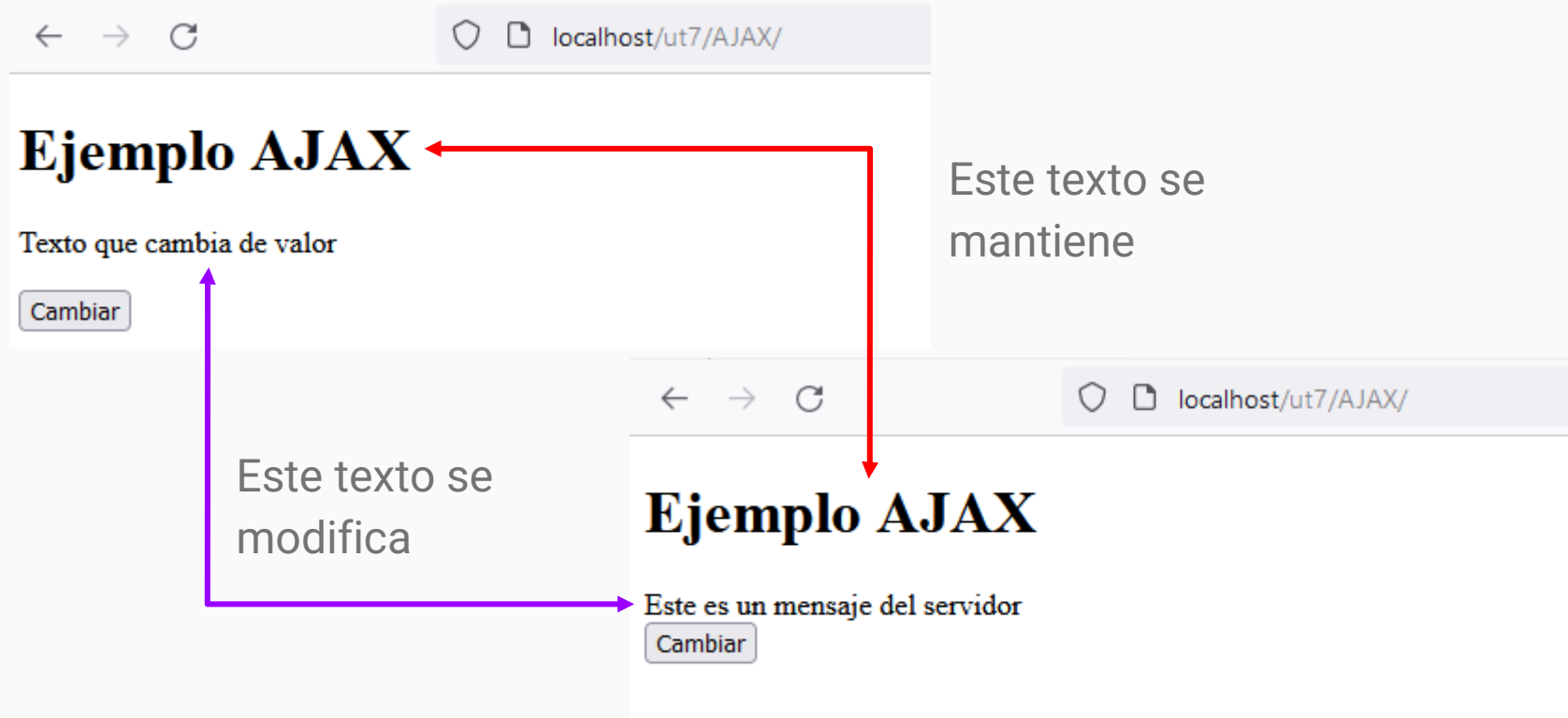
Pese a las ventajas de poder utilizar ambas técnicas (servidor y cliente) para la generación de código

Necesitamos recargar la página cuando queremos mostrar datos nuevos que vengan del servidor (Submit)

- Actualizar el contenido de una página Web sin recargar
- Enviar información a un servidor - después de cargar la página
- Recibir información del servidor - después de cargar la página
- Enviar datos al servidor

***AJAX (Asynchronous JavaScript And XML) Javascript y XML Asíncrono***

### 3. Integración de AJAX con PHP



The diagram illustrates an AJAX integration example using two browser windows. The top window shows the initial state of the page, and the bottom window shows the state after an AJAX call.

**Top Window (Initial State):**

- Address bar: localhost/ut7/AJAX/
- Title: **Ejemplo AJAX**
- Text: Texto que cambia de valor
- Button: Cambiar

**Bottom Window (After AJAX Call):**

- Address bar: localhost/ut7/AJAX/
- Title: **Ejemplo AJAX**
- Text: Este es un mensaje del servidor
- Button: Cambiar

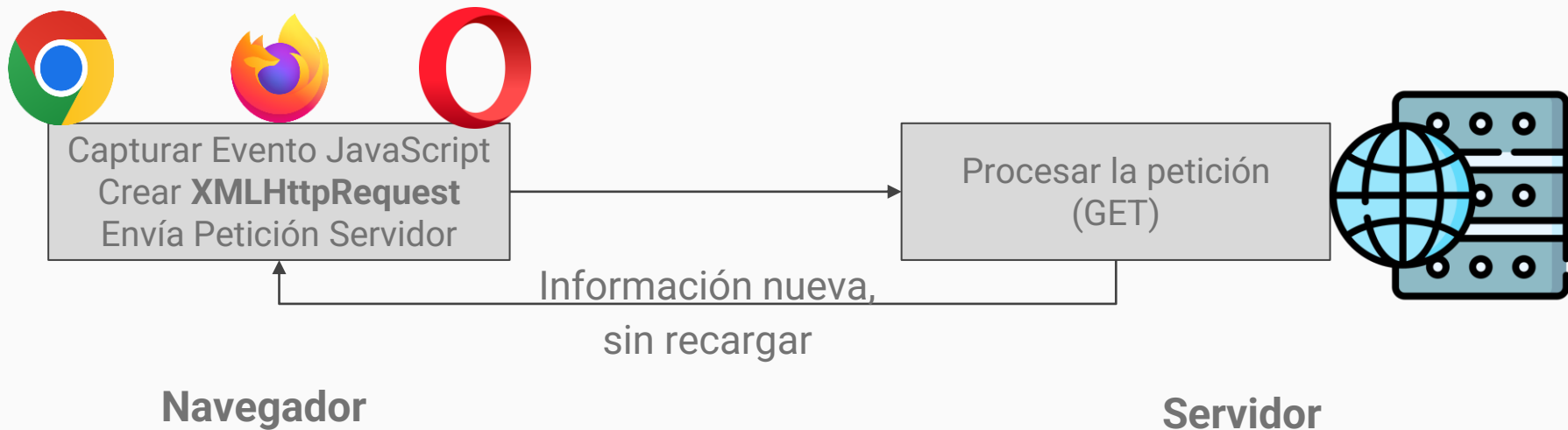
**Annotations:**

- Red Arrow:** Points from the title **Ejemplo AJAX** in the top window to the title **Ejemplo AJAX** in the bottom window, indicating that this text remains unchanged.
- Purple Arrow:** Points from the text **Este texto se modifica** to the new message **Este es un mensaje del servidor** in the bottom window, indicating that the content of the text area has been updated.

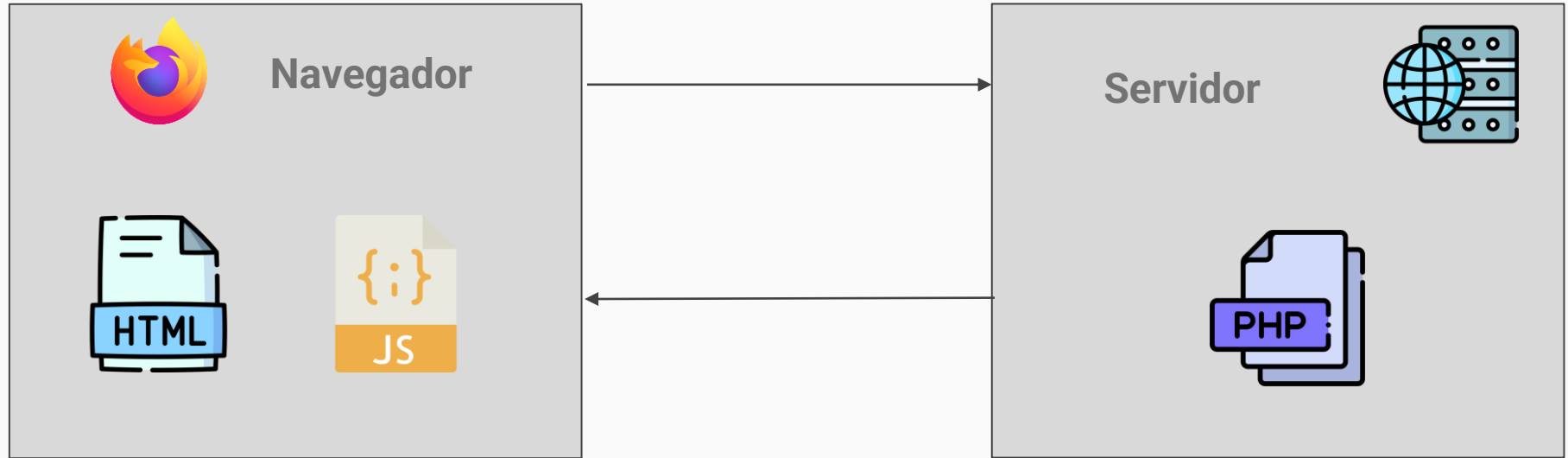


### 3. Integración de AJAX con PHP

Veamos el funcionamiento



### 3. Integración de AJAX con PHP



El funcionamiento de AJAX se apoya en el objeto **XMLHttpRequest**

<https://developer.mozilla.org/es/docs/Web/API/XMLHttpRequest>

Método	Descripción
new XMLHttpRequest()	Creación del objeto
open(method,url,async,user,psw)	Indica los parámetros para realizar la petición
send()	Envía la petición

El funcionamiento de AJAX se apoya en el objeto **XMLHttpRequest**

Propiedad	Descripción
readyState	0 todavía no se llamó a open() 1 todavía no se llamó a send() 2 send() ya fue invocado, y los encabezados y el estado están disponibles 3 Descargando; responseText contiene información parcial 4 La operación está terminada.
status	Estado de la petición HTTP normalmente 200 (Exitoso)

El funcionamiento de AJAX se apoya en el objeto **XMLHttpRequest**

Propiedad	Descripción
onreadystatechange	Define la función para procesar la información recibida
responseText	Información recibida como cadena
responseXML	Información recibida como XML

El funcionamiento de AJAX se apoya en el objeto **XMLHttpRequest**

```
function modificar() {  
    var xhttp = new XMLHttpRequest();  
    xhttp.onreadystatechange = function() {  
        if (this.readyState == 4 && this.status == 200) {  
            document.getElementById("texto").innerHTML = this.responseText;  
        }  
    };  
    xhttp.open("GET", "server.php", true);  
    xhttp.send();  
}
```

Podemos enviar peticiones por POST a nuestro servidor modificando las propiedades del objeto **XMLHttpRequest**

```
xhttp.open("POST", "servidor.php", true);  
xhttp.setRequestHeader("Content-type", "application/x-www-form-  
urlencoded");  
xhttp.send("user=Fernando&pass=Patata");
```

#### Aplicaciones prácticas:

- Listas desplegables enlazadas: Provincia -> Población
- Formularios autocompletar
- Votaciones y registro de respuesta