

---

# SISTEMAS WEB

## CURSO 2021/2022

Tomcat – ShareInfo IV - AJAX  
Aplicación web para compartir mensajes



Web Sistemak by [Oskar Casquero](#) & [María Luz Álvarez](#) is licensed under a [Creative Commons Reconocimiento 4.0 Internacional License](#).

# APLICACIÓN WEB DINÁMICA - SHAREINFO

## INTRODUCCIÓN

---

- **ShareInfo**, es una aplicación de ejemplo para compartir mensajes cortos.
- Utilizando **ShareInfo** se van a analizar, introducir y profundizar en diversos aspectos de las aplicaciones Web en un servidor:
  - Uso de bases de datos en una aplicación Web
  - Ciclo de vida de un servlet
  - Conceptos de sesión (un visitante)
  - Concepto de contexto de la aplicación (varios visitantes)
  - Compartir información entre servlets, en una sesiones y en diferentes sesiones
  - Redireccionamiento
  - Control de eventos (listeners)
  - JSP
  - **AJAX - Javascript y XML Asíncrono**

# AJAX

## JAVASCRIPT ASÍNCRONO + XML (AJAX)

---

- AJAX (*Asynchronous JavaScript And XML*) no es una tecnología por sí misma, es un término que describe un nuevo modo de utilizar conjuntamente varias tecnologías existentes.
- Se ejecutan en el cliente, se mantiene una comunicación asíncrona con el servidor en segundo plano, permitiendo realizar cambios sobre las páginas sin necesidad de recargarlas.
- AJAX puede:
  - Leer datos de un servidor web, después de que se haya cargado la página
  - Actualizar una página web sin recargar la página
  - Enviar datos a un servidor web - en segundo plano
- AJAX usa una combinación de:
  - Un objeto *XMLHttpRequest*, integrado en el navegador, se encarga de las comunicaciones asíncrona.
  - *JavaScript* y HTML DOM (para mostrar o usar los datos)
- Ejemplo: Reloj

# AJAX

## FUNCIONAMIENTO

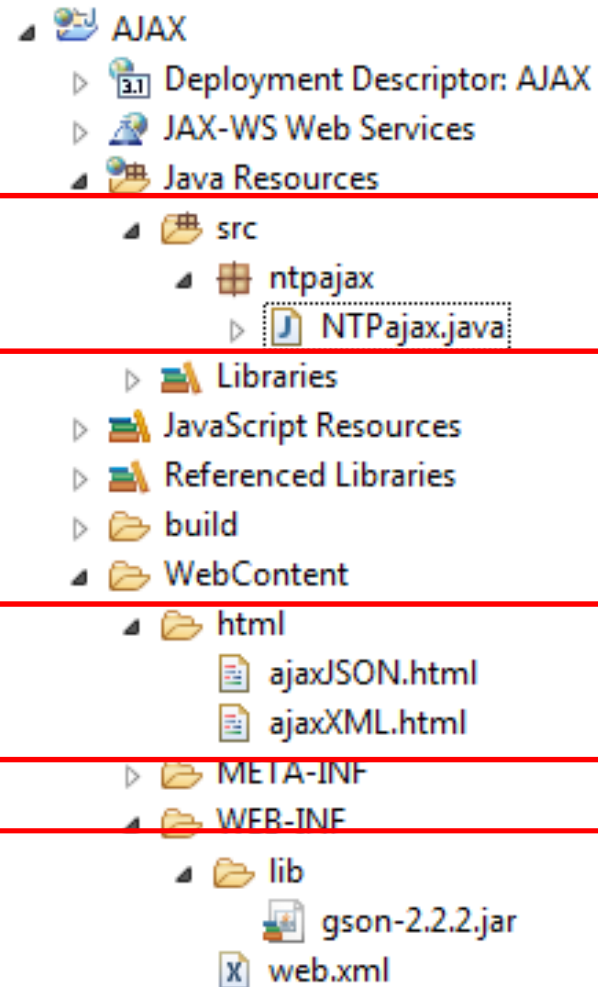
---

- Cuando se produce un **evento** en una página web, por ejemplo, se carga la página, se hace clic en un botón,...
- Se ejecuta un **JavaScript** que crea un objeto *XMLHttpRequest* y envía una solicitud a un servidor web.
- El servidor procesa la solicitud y envía una respuesta a la página web.
- El JavaScript lee la respuesta y realiza la acción adecuada, por ejemplo la actualización de la página.

# AJAX

## EJEMPLO DEL RELOJ – ESTRUCTURA

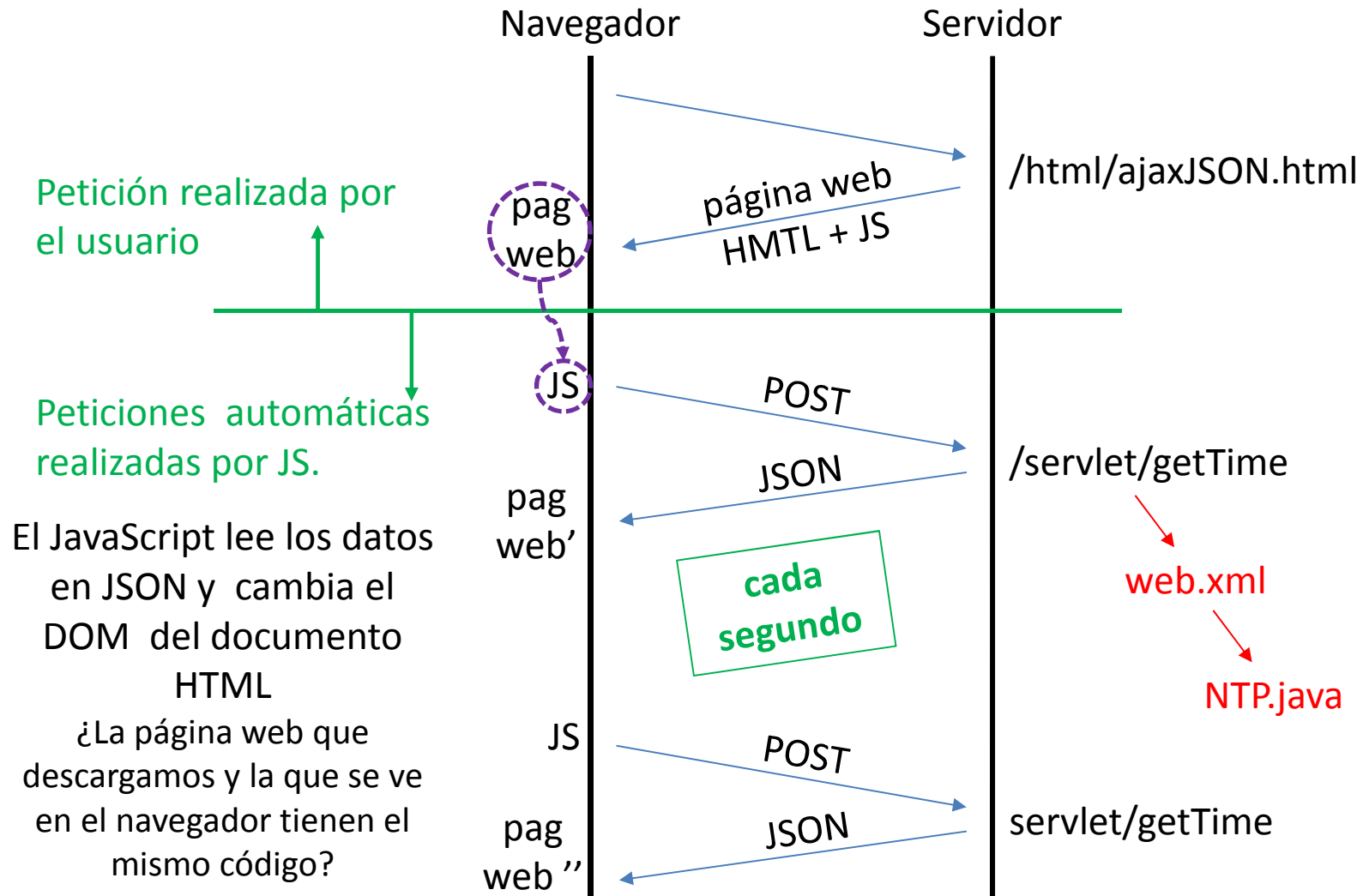
---



Ajax.7z

# AJAX

## EJEMPLO RELOJ – FUNCIONAMIENTO - JSON



# AJAX

## EJEMPLO RELOJ - FUNCIONAMIENTO

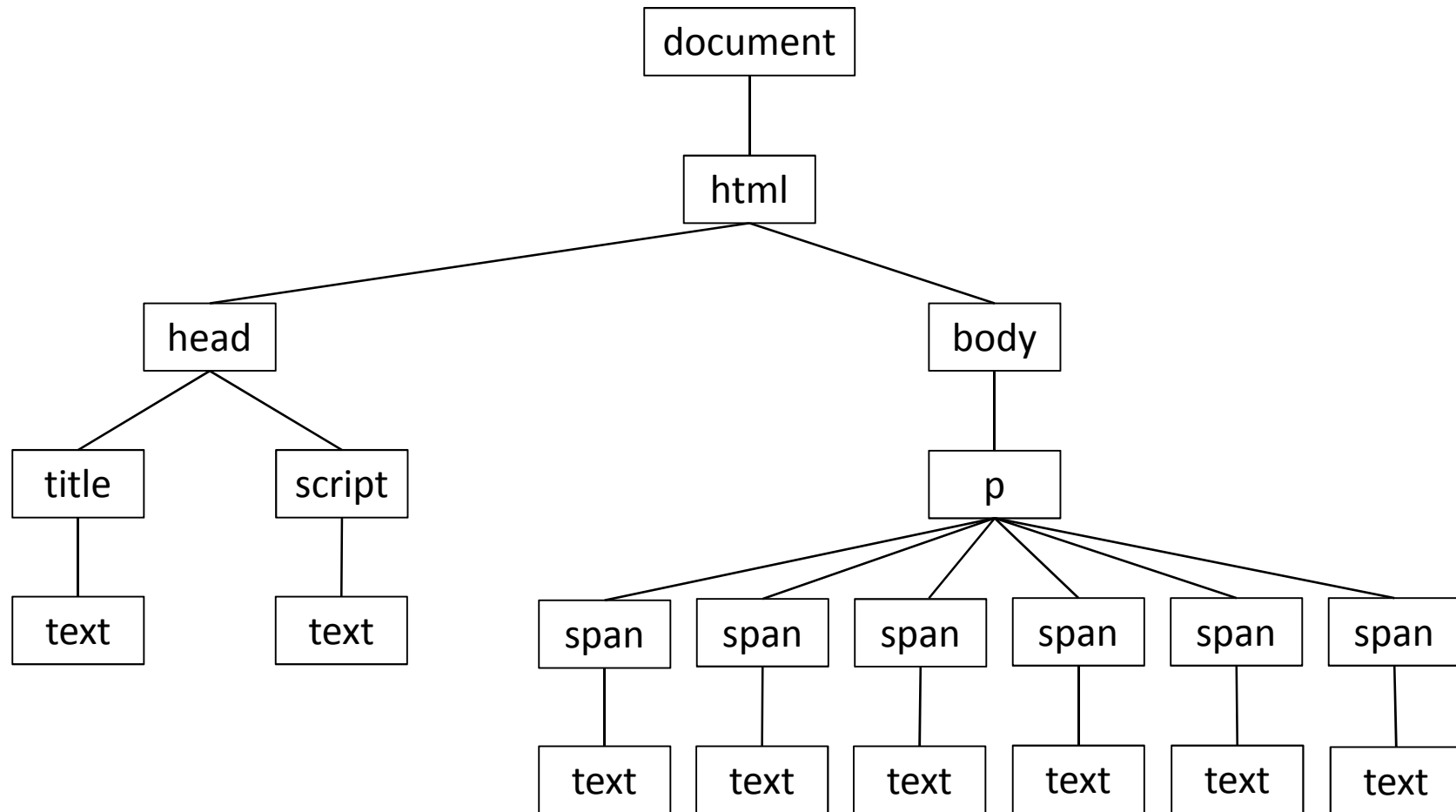
---

```
<html>
  <head>
    <title>AJAX example</title>
    <script type="text/javascript">
      function GetTimeIO() {}
    </script>
  </head>
  <body onload="GetTimeIO()">
    <p>
      day is <span class="day">0</span><br/>
      month is <span class="month">0</span><br/>
      year is <span class="year">0</span><br/>
      hour is <span class="hour">0</span><br/>
      minute is <span class="minute">0</span><br/>
      second is <span class="second">0</span><br/>
    </p>
  </body>
</html>
```

# AJAX

## EJEMPLO RELOJ- JSON.HTML: ARBOL DOM

---





# AJAX

## EJEMPLO RELOJ - FUNCIÓN GETTIMEIO()

---

```
function GetTimeIO() {  
    nocache = "nocache=" + Math.random() * 1000000;  
    var request = new XMLHttpRequest();  
  
    request.onreadystatechange = function() {  
        if (request.readyState == 4) {  
            if (request.status == 200) {  
                if (request.responseText != null) {  
                    var jsonObj = JSON.parse(request.responseText);  
                    document.getElementsByClassName("day")[0].innerHTML = jsonObj.day;  
                    document.getElementsByClassName("month")[0].innerHTML = jsonObj.month;  
                    document.getElementsByClassName("year")[0].innerHTML = jsonObj.year;  
                    document.getElementsByClassName("hour")[0].innerHTML = jsonObj.hour;  
                    document.getElementsByClassName("minute")[0].innerHTML = jsonObj.minute;  
                    document.getElementsByClassName("second")[0].innerHTML = jsonObj.second;  
                }  
            }  
        }  
    };  
  
    request.open("POST", "/AJAX/servlet/getTime?" + nocache, true);  
    request.send(null);  
    setTimeout("GetTimeIO()", 1000);  
}
```

# AJAX

## EJEMPLO RELOJ - REQUEST.ONREADystateCHANGE

---

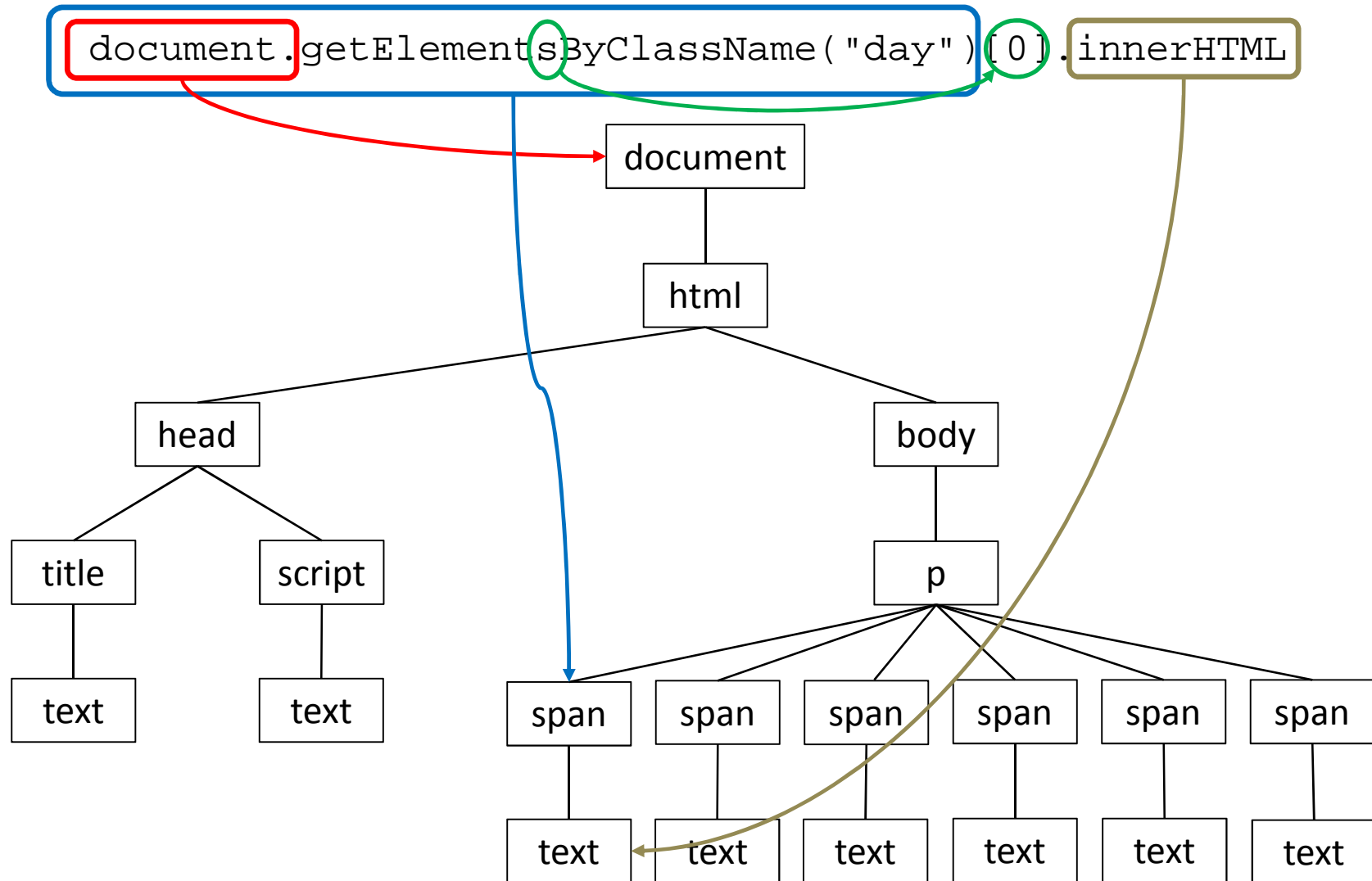
Table extracted from:

[http://www.w3schools.com/ajax/ajax\\_xmlhttprequest\\_onreadystatechange.asp](http://www.w3schools.com/ajax/ajax_xmlhttprequest_onreadystatechange.asp)

Property	Description
onreadystatechange	Stores a function (or the name of a function) to be called automatically each time the readyState property changes
readyState	Holds the status of the XMLHttpRequest. Changes from 0 to 4: 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
status	200: "OK" 404: Page not found

# AJAX

## EJEMPLO RELOJ - AJAX JSON.HTML: DOM ZUHAITZA



# AJAX

## EJEMPLO RELOJ - SERVLET

```
public void doPost(HttpServletRequest request, HttpServletResponse response)
    throws IOException, ServletException {
    Calendar cal = Calendar.getInstance();
    response.setContentType("application/json");
    PrintWriter out = response.getWriter();

    HashMap<String, Object> hashMap = new HashMap<String, Object>();
    hashMap.put("day", cal.get(Calendar.DAY_OF_MONTH));
    hashMap.put("month", cal.get(Calendar.MONTH));
    hashMap.put("year", cal.get(Calendar.YEAR));
    hashMap.put("hour", cal.get(Calendar.HOUR_OF_DAY));
    hashMap.put("minute", cal.get(Calendar.MINUTE));
    hashMap.put("second", cal.get(Calendar.SECOND));

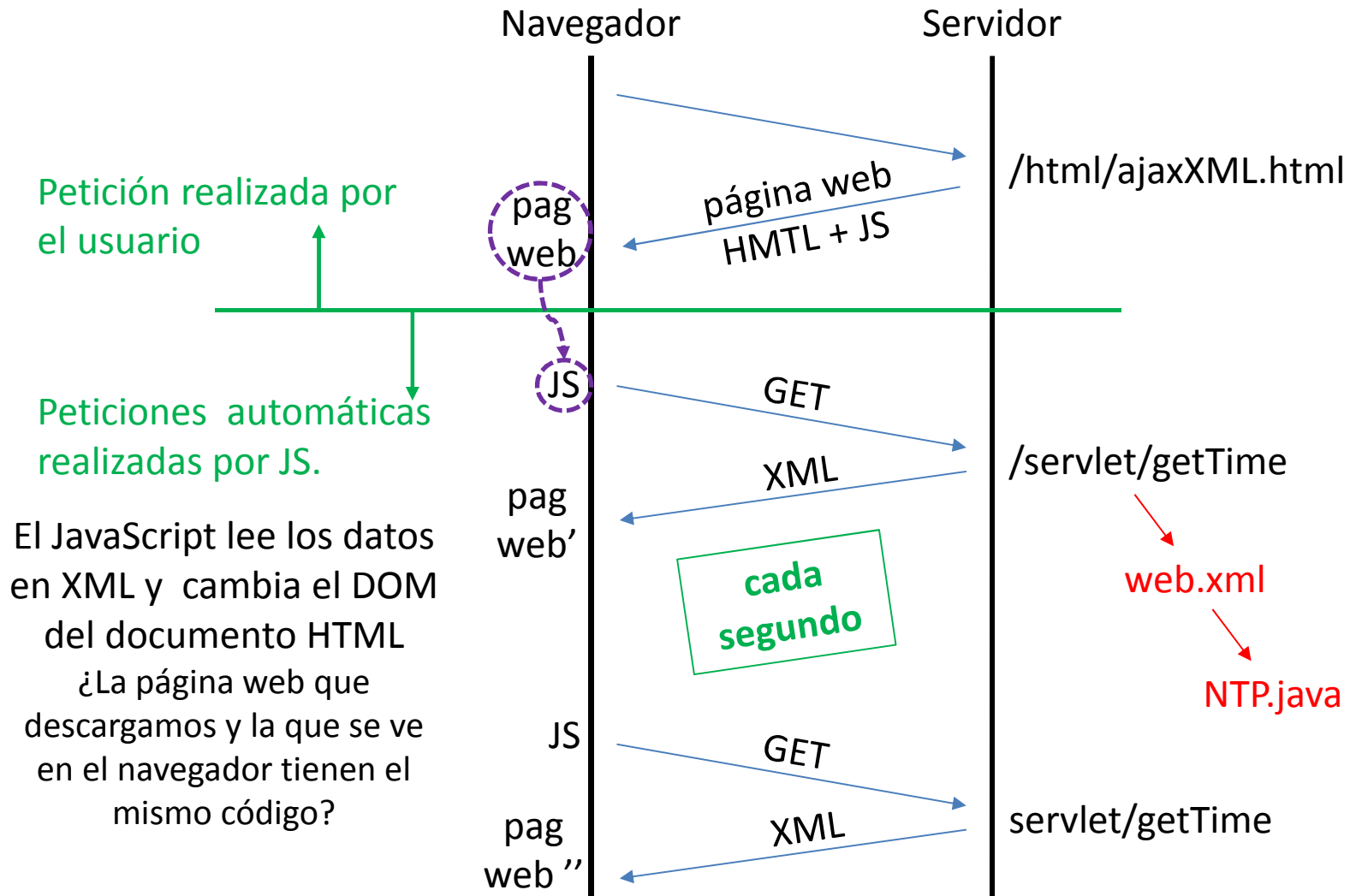
    Gson gson = new Gson();
    String json = gson.toJson(hashMap);

    out.println(json);
    out.flush();
    out.close();

    // Log
    System.out.println("JSON ntp servlet has been hit");
    System.out.println(json);
}
```

# AJAX

## EJEMPLO RELOJ – FUNCIONAMIENTO HTML



# FORMATOS DE INTERCAMBIO

## XML vs JSON --- HTML

Utilizados en Aplicaciones WEB como formatos de intercambio datos

```
{ "month":2,  
  "hour":19,  
  "year":2016,  
  "day":6,  
  "minute":24,  
  "second":14 }
```

**JSON**

```
<?xml version="1.0"?>  
<time>  
  <day>6</day>  
  <month>2</month>  
  <year>2016</year>  
  <hour>19</hour>  
  <minute>23</minute>  
  <second>7</second>  
</time>
```

**XML**

Formato utilizado para la presentación de los datos

```
<html>  
<head>  
  <title>AJAX example</title>  
  
</head>  
<body onload="GetTimeIO()">  
  <p>  
    day is <span class="day">0</span><br>  
    month is <span class="month">0</span><br>  
    year is <span class="year">0</span><br>  
    hour is <span class="hour">0</span><br>  
    minute is <span class="minute">0</span><br>  
    second is <span class="second">0</span><br>  
  </p>  
</body>  
</html>
```

**HTML**

# APLICACIÓN WEB DINÁMICA - SHAREINFO

## ACTUALIZAR CON AJAX MENSAJES

---

**A webapp to share short messages**

**View Messages**

Refresh

You are logged in as: kaske

Active users: mariluz kaske

Message:

Send

Username	Message
kaske	Kaixo guztioil
mariluz	Hola a todos!

Web Systems - EUITI Bilbao

# APLICACIÓN WEB DINÁMICA - SHAREINFO

## ACTUALIZAR CON AJAX ACTIVE USERS

---

**A webapp to share short messages**  
**View Messages**

Refresh

You are logged in as: kaske

Active users: mariluz kaske

Message:

Send

Username	Message
kaske	Kaixo guztioil
mariluz	Hola a todos!

Web Systems - EUITI Bilbao



# ACTIVIDAD EVALUABLE 1 – SHAREINFO – APLICACIÓN WEB

---

- Añadir las siguientes mejoras.
  - 1.- Actualizar con Ajax la lista de usuarios activos
  - 2.- Permitir que el usuario elimine todos **sus** mensajes de la base de datos.
  - 3.- Permitir cambiar de contraseña al usuario.