
SISTEMAS WEB

CURSO 2023/2024

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

ASYNCHRONOUS JAVASCRIPT AND XML

- **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***, interface integrado en el navegador, se encarga de las comunicaciones asíncrona.
 - ***JavaScript*** y **HTML DOM (Document Object Model)**, para mostrar o usar los datos.
- Ejemplo: FechaAjax

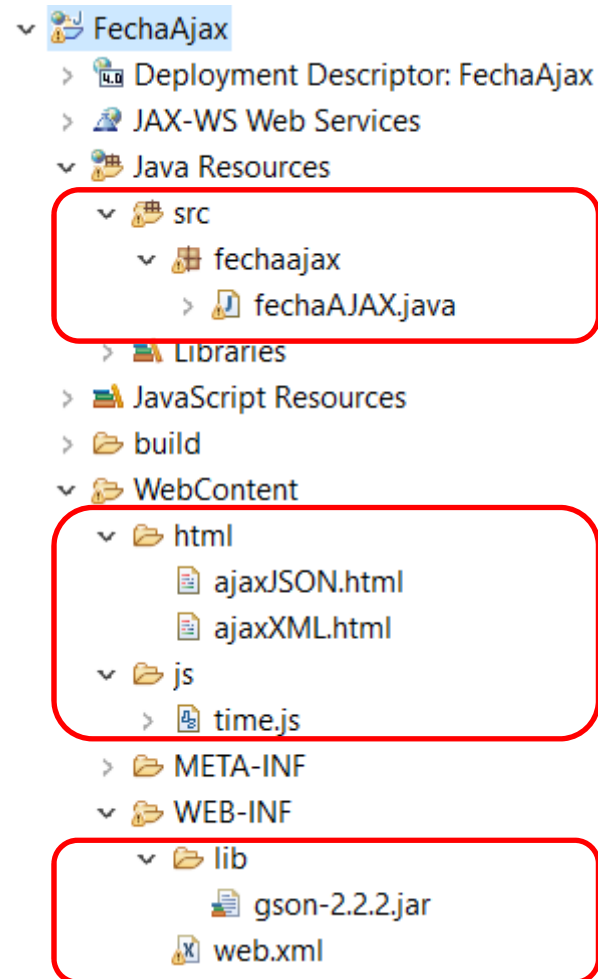
AJAX

FUNCIONAMIENTO

- Cuando ocurre un **evento** en una página web, por ejemplo, se carga la página, se hace clic en un botón, se ejecuta un código **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 código JavaScript lee la respuesta y realiza la acción adecuada, por ejemplo la actualización de la página.

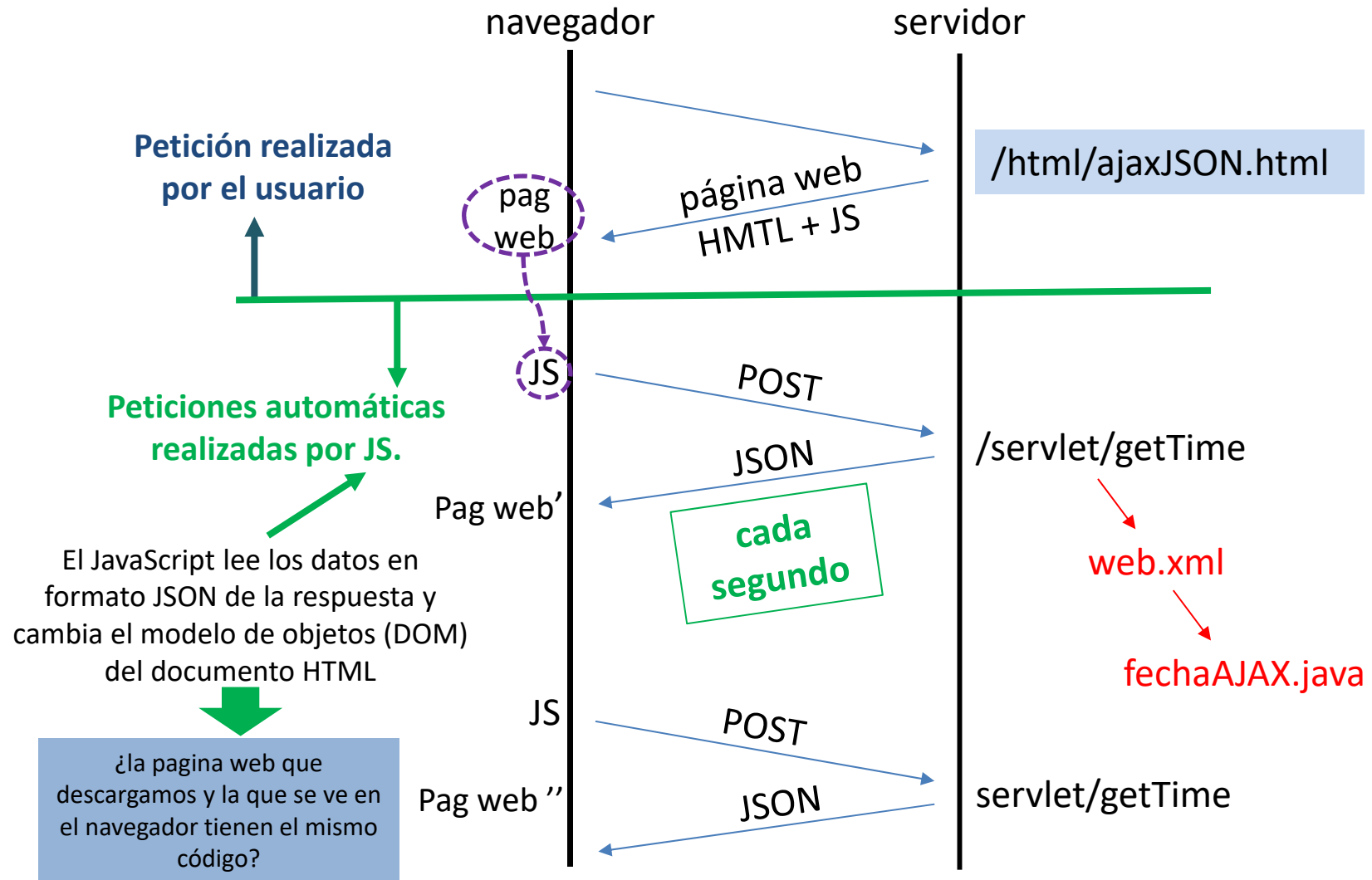
AJAX

EJEMPLO – FECHA AJAX.7z



AJAX

EJEMPLO - FUNCIONAMIENTO



AJAX

EJEMPLO - FUNCIONAMIENTO

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
  <head>
    <title>
      AJAX example
    </title>
    <script type="text/javascript">
  </head>
  <body onload="GetTimeIO()">
    <h3> JSON JavaScript Object Notation </h3>
    <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>
```

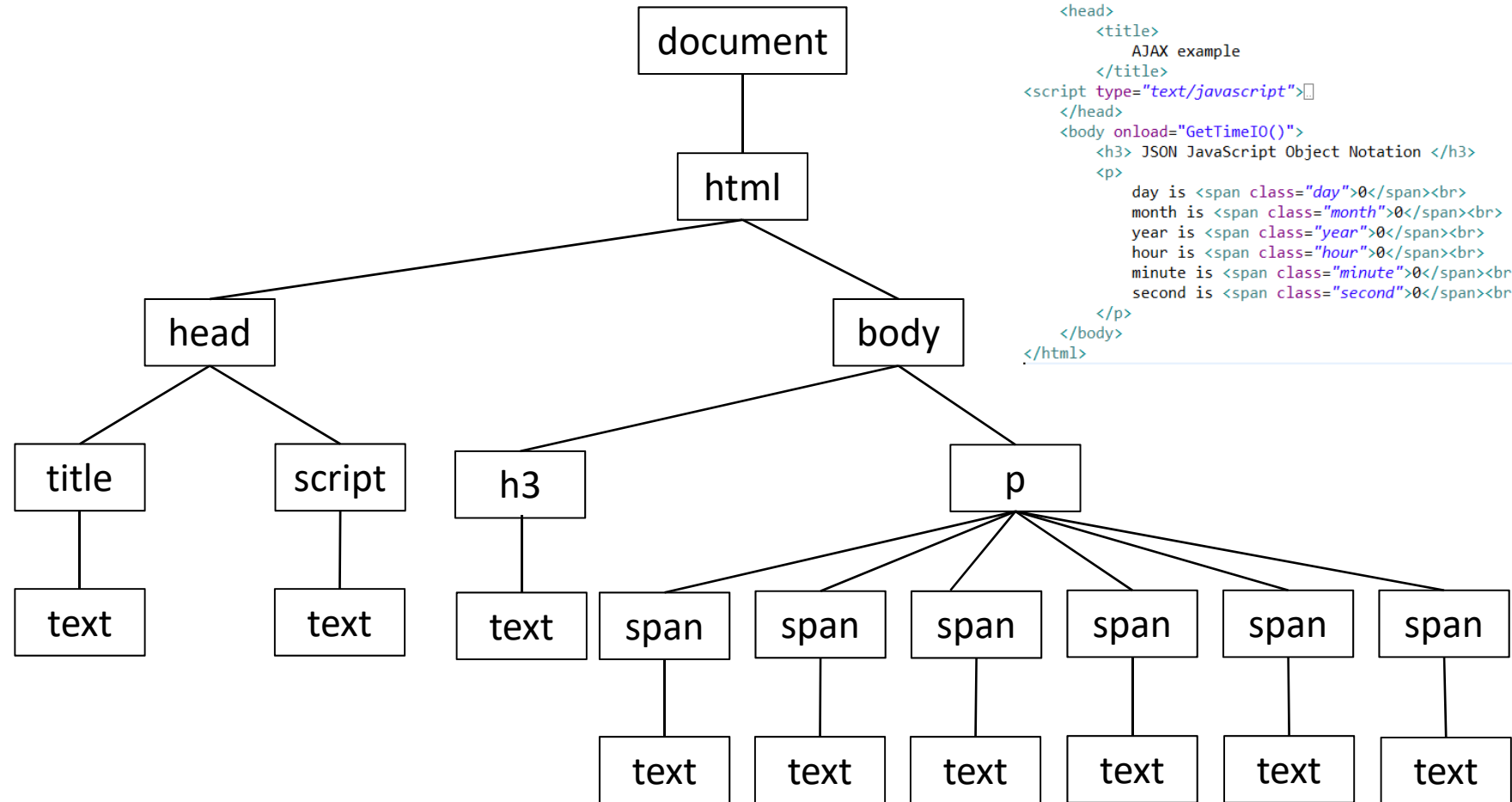


A red box highlights the `<script type="text/javascript">` tag in the head section. A red arrow points from this box to another red box that highlights the `onload="GetTimeIO()"` attribute in the `<body>` tag, indicating that the script is executed when the page loads.

</html/ajaxJSON.html>

AJAX

EJEMPLO - ARBOL DOM



AJAX

EJEMPLO - FUNCIÓN JS GETTIMEIO()

```
function GetTimeIO() {  
    nocache = "nocache=" + Math.random() * 1000000;  
    var request = new XMLHttpRequest();
```

Objeto 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;  
                }  
            }  
        }  
    };
```

Función que se llama cuando cambia la propiedad readyState

```
    request.open("POST", "/FechaAjax/servlet/getTime?" + nocache, true);  
    request.send(null);
```

Realizar petición

```
    setTimeout("GetTimeIO()", 1000);
```

Repetir cada segundo

```
}
```



AJAX

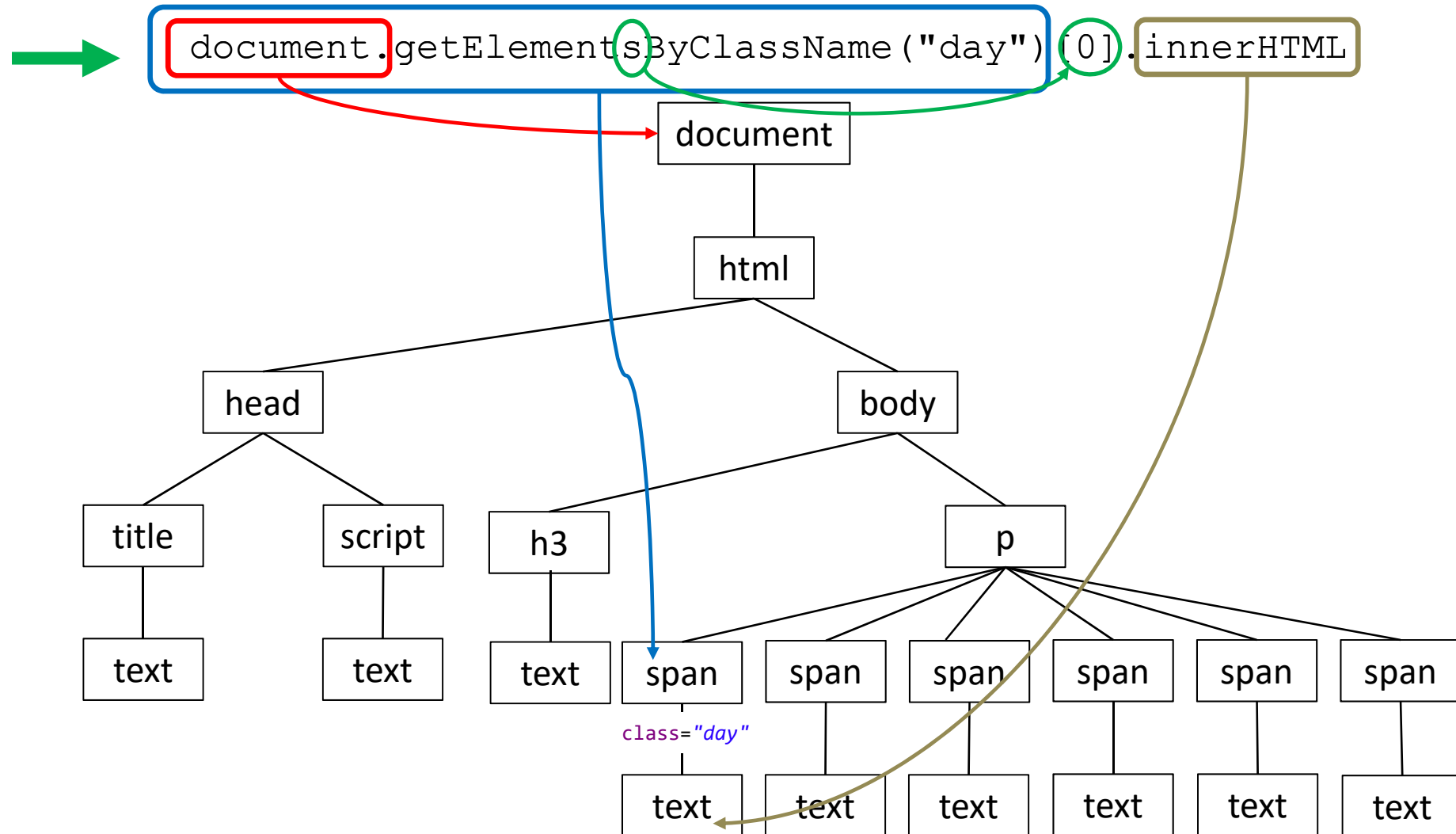
PROPIEDAD *ONREADYSTATECHANGE*

Propiedad	Descripción
onreadystatechange	Define una función que será llamada cuando la propiedad readyState cambie
readyState	Mantiene el estado de la XMLHttpRequest. 0: Solicitud no inicializada 1: Establecida conexión con servidor 2: Solicitud recibida 3: Procesando solicitud 4: Solicitud finalizada y respuestalista
status	200: "OK" 403: "Forbidden" 404: "Page not found"
statusText	Devuelve el texto del estado(p.ej. "OK" or "Not Found")

https://www.w3schools.com/js/js_ajax_http.asp

AJAX

EJEMPLO - ARBOL DOM



AJAX

EJEMPLO - 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);
}
```

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

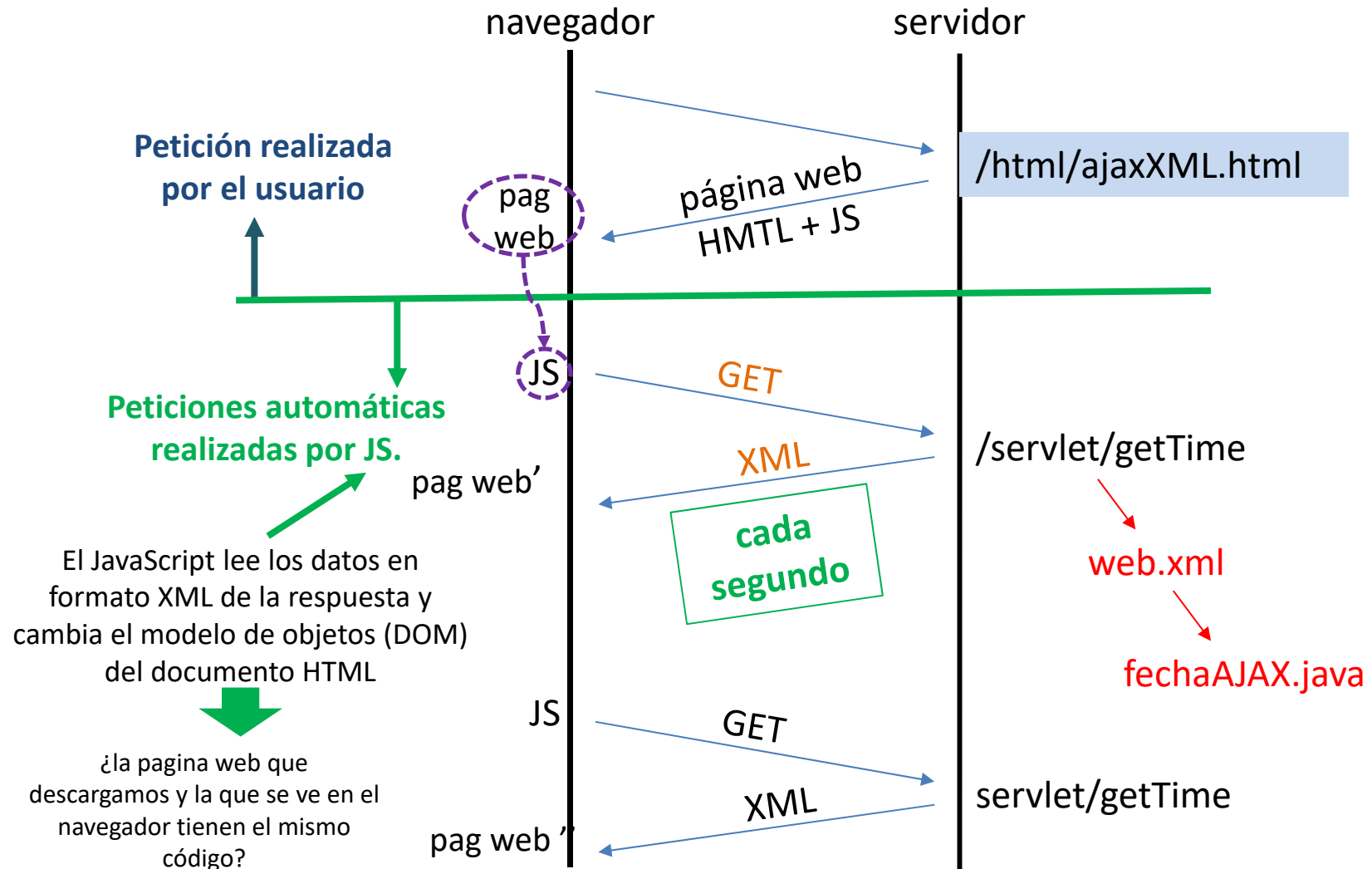
AJAX

EJEMPLO - SERVLET

- Analiza del mismo modo la pagina **ajaxXML.html**.
- Analiza las diferencias.

AJAX

EJEMPLO - FUNCIONAMIENTO



APLICACIÓN WEB DINÁMICA - SHAREINFO

HORA /MENSAJE/USUARIOS ACTIVOS

Web para Compartir Mensajes Cortos
Vista de Mensajes

Actualizar Exit

Tu usuario mla;

Usuarios Activos: mla;

Mensaje:

Enviar

Usuario	Mensaje
kaske	kaixol
mariluz	hola
mla	"Cuando creíamos que teníamos todas las respuestas, de pronto, cambiaron todas las preguntas"

Hora Servidor: 20:51:35
Hora Cliente: 20:51:35

Sistemas Web - Escuela de Ingeniería de Bilbao

APLICACIÓN WEB DINÁMICA - SHAREINFO

ACTUALIZAR CON AJAX MENSAJES

```
public class MsgAjax extends HttpServlet {  
  
    private MySQLdb mySQLdb;  
  
    public void init(ServletConfig config) {  
        System.out.println("---> Entering init() MsgAjax");  
        mySQLdb = new MySQLdb();  
        System.out.println("---> Exiting init() MsgAjax");  
    }  
  
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws  
        IOException, ServletException {  
        System.out.println("---> Entering doGet() MsgAjax");  
  
        ArrayList<MessageInfo> messageList= mySQLdb.getAllMessages();  
  
        Gson gson = new Gson();  
        String json = gson.toJson(messageList);  
        System.out.println(json);  
  
        response.setContentType("application/json");  
        PrintWriter out = response.getWriter();  
        out.println(json);  
        out.flush();  
        out.close();  
  
        System.out.println(json);  
        System.out.println("---> Exiting doGet() MsgAjax");  
    }  
}
```

1.- Creamos un servlet que lea los mensajes de la base de datos y envíe los datos en formato json
<http://localhost:8080/ShareInfo/servlet/MsgAjax>

lista de mensajes

Creamos Json

Lo escribimos en
cuerpo de la
respuesta

APLICACIÓN WEB DINÁMICA - SHAREINFO

ACTUALIZAR CON AJAX MENSAJES

```
<html>
<head>
<title>View Messages</title>
<link href="/ShareInfoV2/css/styleSheet.css" rel="stylesheet"
<script type="text/javascript">
function GetMsg() {
    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);
                    var table = "<tr><th>Username</th> <th>Message</th></tr>";
                    for (i = 0; i < jsonObj.length; i++) {
                        table += "<tr><td>" + jsonObj[i].username + "</td>";
                        table += "<td>" + jsonObj[i].message + "</td></tr>";
                    }
                    document.getElementsByClassName("msgtable")[0].innerHTML = table;
                }
            }
        }
    };
    request.open("GET", "/ShareInfo/servlet/MsgAjax", true);
    request.send(null);
    setTimeout("GetMsg()", 1000);
};
</script>
</head>
```

2.- Modificamos **viewMessages.jsp** para que modifique la página html con AJAX

A.- crear el js

B.-Modificar el html para que se ejecute el js al inicio

Con el Json creamos una nueva tabla

Sustituimos la tabla en el árbol DOM del html

APLICACIÓN WEB DINÁMICA - SHAREINFO

ACTUALIZAR CON AJAX MENSAJES

```
<body onload="GetMsg()">
  <header>
    <h1>A webapp to share short messages</h1>
    <h3>View Messages</h3>
  </header>
  -.....
  <section>
    <table class="msgtable">
      <tr>
        <th>Username</th>
        <th>Message</th> </tr>
        <% for (int i = 0; i < messageList.size(); i++) {
          MessageInfo messageInfo = messageList.get(i); %>
          <tr><td><%=messageInfo.getUsername()%></td>
          <td><%=messageInfo.getMessage()%></td>
          </tr>
        <% } %>
      </table>
    </section>
    <footer>Web Systems - EUITI Bilbao</footer>
  </body>
</html>
```

2.- Modificamos **viewMessages.jsp** para que genere la página html con AJAX

A.- crear el javascript

B.-Modificar el html para que se ejecute el javascript al inicio

Ejecutar js al cargar el html

Identificar la tabla se le pone un nombre

APLICACIÓN WEB DINÁMICA - SHAREINFO

HORA /MENSAJE/USUARIOS ACTIVOS

Web para Compartir Mensajes Cortos
Vista de Mensajes

Actualizar Exit

Tu usuario mla;

Usuarios Activos: mla;

Mensaje:

Enviar

Usuario	Mensaje
kaske	kaixol
mariluz	hola
mla	"Cuando creíamos que teníamos todas las respuestas, de pronto, cambiaron todas las preguntas"

Hora Servidor: 20:51:35

Hora Cliente: 20:51:35

Sistemas Web - Escuela de Ingeniería de Bilbao

ACTIVIDAD EVALUABLE 1 – SHAREINFO – APLICACIÓN WEB

- Añadir las siguientes mejoras.
 - 1.- Actualizar con Ajax la lista de usuarios activos y horas del cliente y servidor.
 - 2.- Permitir que el usuario elimine todos **sus** mensajes de la base de datos.
 - 3.- Permitir cambiar de contraseña al usuario.
 - 4.- Caduca la sesión. Como mejorar.