

ARQUITECTURA CLIENTE - SERVIDOR

Introducción

El protocolo HTTP

El protocolo HTTP es el protocolo utilizado en Internet para realizar la “conversación” entre diferentes máquinas, denominadas cliente y servidor. El cliente realiza pedidos a un determinado servidor, que son transportados por la red a través de paquetes, utilizando precisamente este protocolo.

Sus siglas significan HyperText Transfer Protocol, y tiene como objetivo envolver el lenguaje HTML (HyperText Markup Language) y otras especificaciones para luego ser transportado hacia su destino.

El Web Server

En una arquitectura Cliente-Servidor, representa al servidor que se encarga de procesar los pedidos de los distintos clientes. Tiene la capacidad de recibir y administrar pedidos, para determinar la respuesta a enviar.

Además gestiona otros tipos de servicios tales como el de autenticación de usuario y la “comprensión” de las solicitudes del tipo seguras.

Tiene además la capacidad de colaborar con servicios configurados en la misma máquina, como scripts de programación que se ejecutan en coordinación y realizan algún tipo de procesamiento. Generalmente es necesario cierta configuración para disponer y poder utilizar dichos servicios.

El Navegador

En una arquitectura Cliente-Servidor, representa al cliente que solicita pedidos al servidor. Adicionalmente, tiene la responsabilidad de “comprender” las respuestas provenientes del servidor, interpretarlas y visualizarlas en pantalla de modo gráfico. Se conoce como un tipo de “cliente liviano” ya que prácticamente no tiene procesamiento, simplemente interpreta y visualiza.

Qué es un request

- Request = solicitud o pedido
- El request representa un pedido que hace un cliente al servidor
- En el pedido, el cliente le envía datos al servidor
- A partir de este pedido, el servidor obtiene los datos, los procesa y genera una respuesta
- Los datos pueden venir con el método GET o con el método POST, lo cual determinará la forma de procesamiento en el servidor

Qué es un response

- Response = respuesta
- El response representa la respuesta que realiza el servidor luego de recibir un pedido
- En general, retorna código HTML, aunque puede retornar otro tipo de información, como ser imágenes, XML, etc.

Páginas dinámicas vs. páginas estáticas

A la hora de trabajar con páginas Web, existen dos categorías de páginas bien definidas: las páginas

estáticas y las páginas dinámicas.

Las **páginas estáticas** son aquellas que un diseñador crea con el lenguaje HTML, puede incluir objetos incrustados de tipo Flash e imágenes, y luego sube al servidor. Son páginas que no poseen secciones de programación, es decir que ya están predefinidas, y su comportamiento/visualización es siempre el mismo, no importa quien realice un pedido por este tipo de página.

A diferencia de las páginas estáticas, las **páginas dinámicas** son aquellas que se generan cuando son solicitadas por un cliente, es decir, se generan en forma dinámica de acuerdo al pedido que se efectúe. Son páginas que poseen código de programación, que es “interpretado” en tiempo de ejecución para construir lenguaje HTML, pero distinto de acuerdo a la circunstancia. Las páginas dinámicas cambian su contenido dependiendo de la interacción del cliente. Un caso típico, es un carrito de compras.

Server-side processing

Creación de páginas dinámicas

Las páginas dinámicas tienen una extensión que no es html, generalmente suelen ser .php, .asp o .jsp, aunque también hay otras variantes. Dichas páginas se generan en forma dinámica por una aplicación instalada en el servidor, que se encarga de ejecutar un lenguaje de programación en cuestión, y generar el HTML necesario para la representación de la página Web.

Esta aplicación instalada “coordina y conversa” con el servidor Web para llevar a cabo la generación de las páginas dinámicas.

JSP / Servlets

Los servlets y las páginas JSP representan la propuesta de Oracle para la programación Web. Inicialmente fueron los **Servlets** los que aparecieron en el mercado, mejorando la propuesta que había anteriormente que eran los CGIs. Luego surge la tecnología **JSP (Java Server Pages)** que está basada en el uso de Servlets, pero con una productividad mucho mayor.

Los Servlets son unidades de código que presentan cierta funcionalidad, y se ejecutan del lado del servidor. Tiene como uno de sus objetivos generar el código HTML que se enviará al cliente. Técnicamente, se construyen a partir de clases que heredan de una superclase general, que forma parte de **JEE (Java Enterprise Edition)**

Los archivos JSP son la propuesta sencilla respecto de los Servlets, a diferencia de estos la programación se realiza a modo scripting. Se construyen a partir de archivos de texto con extensión .jsp

Alternativas: ASP y PHP

En el mercado, las tecnologías Web más reconocidas y utilizadas son ASP, PHP, JSP.

PHP es la propuesta más masiva, aunque no es utilizada para proyectos de gran envergadura ya que carece de respaldo empresarial.

ASP (Active Server Pages) es la propuesta de Microsoft, actualizada hace un tiempo por ASP.NET, y representa una tecnología productiva, aunque carece de los pilares que tiene la tecnología JAVA.

Client-side processing

Responsabilidades del navegador

El navegador se encarga de realizar el procesamiento del lado del cliente. En sus responsabilidades principales se destacan:

- Enviar los datos al servidor en “formato legible”
- Solicitar y recibir páginas del servidor
- Interpretar el lenguaje HTML para visualizarlo en forma gráfica
- Permitir realizar la validación de formularios

Javascript

Javascript es una de las tecnologías que poseen los navegadores, la cual permite realizar el procesamiento y manipular la información. No es parte de la tecnología Java, simplemente es un lenguaje de para realizar scripting.

Se utiliza principalmente para:

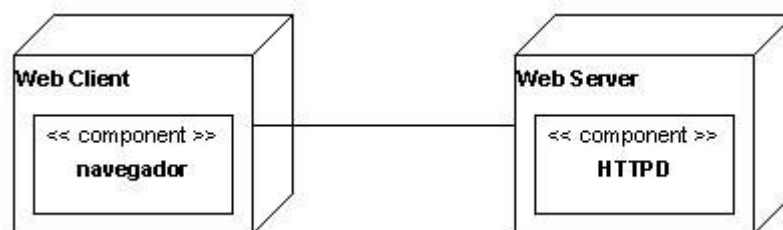
- Manipular componentes de HTML a través del DOM
- Realizar validaciones de formularios antes de ser enviados al servidor
- Controlar los eventos de una página HTML

Ajax

AJAX es una tecnología que tiene razonablemente poco tiempo en el mercado, es una de las más nuevas. Significa ***Asynchronous Javascript and XML***.

Es un mecanismo que permite, entre otras cosas, realizar una “conversación” entre el cliente y el servidor, pero con la ventaja de que este pedido, request en inglés, se realiza en el background, es decir de forma transparente para el usuario. Como consecuencia, el usuario no ve que la página se refresca, y se vuelva a dibujar.

Diagrama



HTML - HYPERTEXT MARKUP

LANGUAGE

Introducción

Definición

El lenguaje HTML es el lenguaje utilizado en Internet para la construcción de páginas Web. Se trata de texto en forma de etiquetas que permiten construir estructuras elaboradas a nivel diseño gráfico, y tienen la capacidad de agrupar información ingresada por el usuario para ser enviada posteriormente al servidor.

Las páginas Web son almacenadas en el servidor y los browser son los encargados de solicitarlas, interpretarlas y visualizarlas de formas gráfica.

Qué son los tags

Los tags, o etiquetas, es el mecanismo que tiene el lenguaje HTML para organizar la información, generalmente a través de la relación “contiene a”, es decir los tags se van conteniendo unos a otros para ir generando estructuras complejas. Cada tag tiene una semántica bien definida, con una representación gráfica asociada (los que se pueden visualizar).

Formularios

Definición

Los formularios representan el mecanismo para recolectar datos del cliente, es la forma de llevar a cabo el denominado user input. La información recolectada es enviada al servidor por medio del navegador, que se encarga de guardar dicha información bajo cierto formato estándar y la envía en forma de paquetes (respetando el protocolo HTTP) al servidor.

Los datos pueden ser enviados de dos formas distintas al servidor: a través del método GET y a través del método POST

El tag <FORM>

La etiqueta utilizada por HTML para modelar un formulario es <FORM>, donde todo el contenido del formulario se ubica posterior a esta etiqueta, y luego se cierra el bloque con la etiqueta </FORM>. Todos los tags que se utilizan para la captura de datos deberán quedar encerrados dentro del cuerpo de los tags <FORM> y </FORM>.

El tag tiene dos propiedades muy importantes: la propiedad **method** y la propiedad **action**. La propiedad “method” se utiliza para determinar si al enviar los datos se utiliza el método GET o el método POST, y la propiedad “action” se utiliza para determinar que recurso del servidor (generalmente una página web) procesará dichos datos. Un ejemplo de utilización podría ser: <FORM method=GET action=procesar.jsp>

El tag <INPUT>

El tag <INPUT> se utiliza para capturar los distintos datos que ingresa el usuario. Siempre deben quedar encerrados dentro de un tag <FORM>. Sus propiedades más importantes son **type** y **name**. La propiedad “name” se utiliza para identificar el componente que envía la información dentro de todos los componentes existentes, y la propiedad “type” se utiliza para determinar qué tipo de componente es. Entre las propiedades “type” más conocidas están button, submit, checkbox, radio, text, password, etc.

Envío de datos a través de HTTP

El método GET

- El método GET es la forma más sencilla de enviar información
- La información es enviada a través de la barra de direcciones
- Los datos a enviar son publicados en la barra de direcciones, y como consecuencia cualquiera que esté utilizando el browser puede ver el envío de dichos datos
- No es útil para el envío de contraseñas
- Tiene una limitación de 255 caracteres, si se envía una cadena de datos más larga entonces quedará truncada en el carácter 255
- Es ideal para el uso de links con pocos valores a ser procesados en el servidor

El método POST

- Es la alternativa al método GET
- La información es colocada en los headers de los paquetes enviados al Server, con lo cual no es visible para el cliente
- Se utiliza para el envío de contraseñas
- No tiene máximo en cuanto al envío de datos
- Es ideal para el envío de gran cantidad de datos al servidor, como ser un formulario de registración

SERVLETS

Introducción

Qué es un Servlet

Es parte de la tecnología Java junto con JSP, y pertenece a la Edición Empresarial (JEE).

Es una unidad de funcionalidad que se ejecuta del lado del servidor, y genera resultados que son enviados al cliente.

Debe ser desplegado dentro de un Servlet Container para su correcto funcionamiento.

La clase HttpServlet

Cuando es necesario trabajar dentro de un marco que utiliza el protocolo HTTP, es necesario construir una subclase de HttpServlet

Sus métodos más importantes son doGet(), doPost()

El método doGet()

Es el método utilizado para procesar los datos que vienen a través del método GET

Debe ser sobrescrito en la subclase de HttpServlet

El método doPost()

Es el método utilizado para procesar los datos que vienen a través del método POST

Debe ser sobrescrito en la subclase de HttpServlet

El método getParameter()

Utilizado para obtener el valor de un parámetro enviado por el cliente vía GET o POST

La firma del método es: `String getParameter(String s)`

Si el dato enviado está en una caja de texto denominada `txtNombre`, entonces la forma de obtener ese dato será:

```
String elDato = request.getParameter("txtNombre");
```