

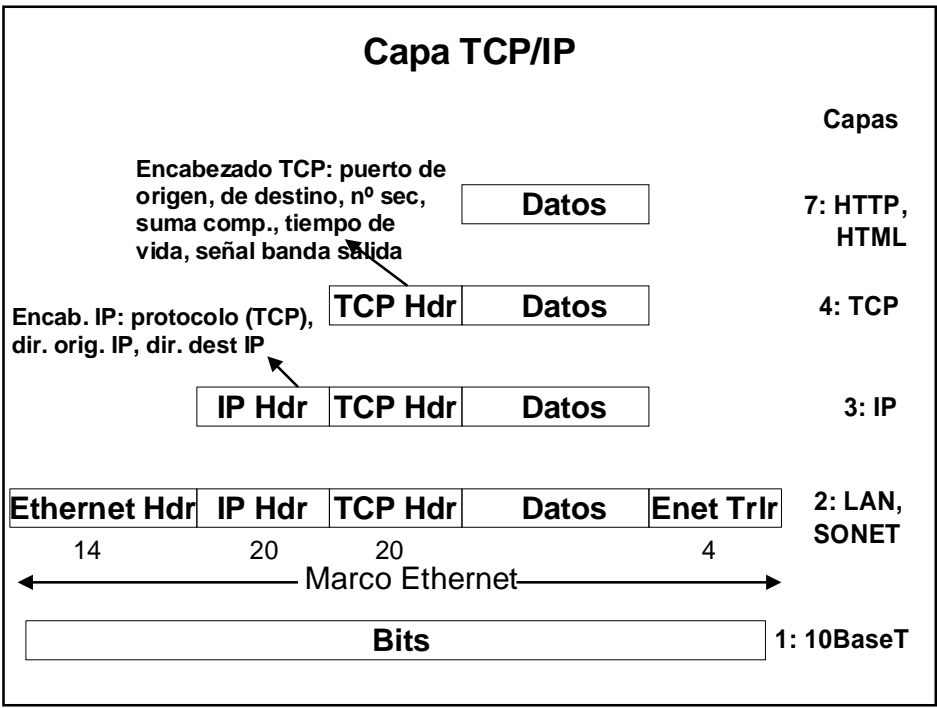
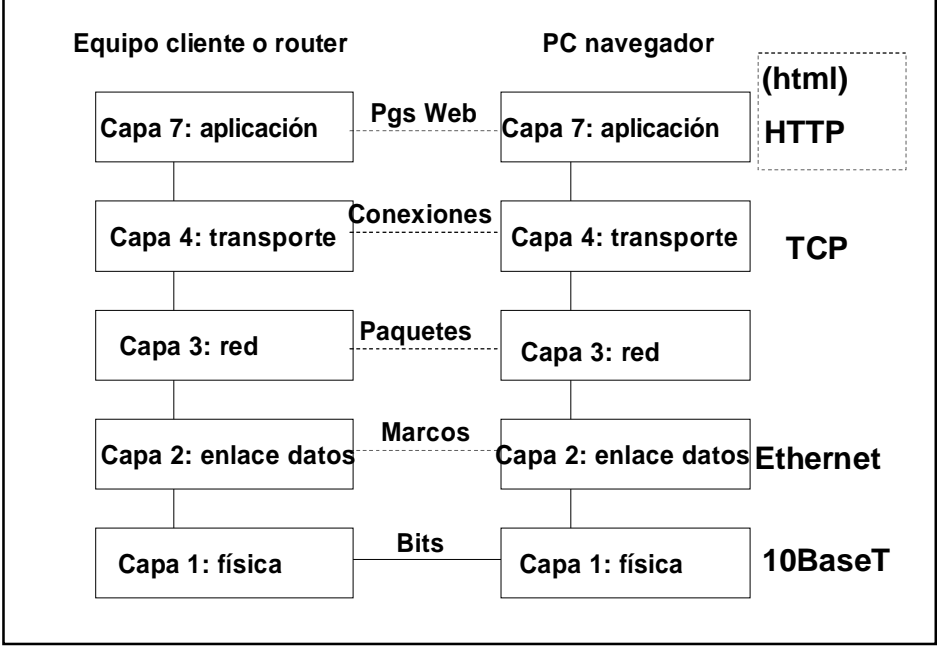
## 1.00 Clase 36

### Java y la Web

#### Navegadores web y servidores

- **Internet es “sólo” un conjunto de redes más o menos interconectadas**
  - Un conjunto de redes de área local conectadas mediante redes de área extensa
  - Sin control ni gestión centralizados
  - Los segmentos de red se interconectan con *routers*
  - Los *routers* son ordenadores dedicados que gestionan paquetes de datos
  - TCP/IP es el protocolo de datos universal de la red
  - El formato y contenido actual se deja a protocolos de mayor nivel, como la Web
- **Conexiones TCP/IP**
  - El cliente suele ser un consumidor de datos que envía peticiones breves
    - En la Web, el cliente es un navegador
  - El servidor suele ser un proveedor de datos que envía respuestas largas
    - Escucha peticiones y transmite los datos deseados (en el conocido puerto 80)
    - Puede enviar datos estáticos o dinámicos
  - La conexión a la Web está activa solamente durante el intercambio de datos
    - Se evita la sobrecarga de varios canales de comunicación, pero se pierde el estado

## Protocolo de control de transmisión/Protocolo Internet TCP/IP



## Organización de la Web: páginas web

- **Una página web es una unidad básica de organización**
  - Páginas simples recuperadas en una única operación
  - Las páginas compuestas (normalmente, texto y gráficos) requieren varios pasos (varias conexiones TCP/IP a no ser que esté definida la opción “Keep-Alive”)
  - Las páginas contienen hipertexto
    - Enlaces a otras páginas, implementados con URL incrustadas
  - Las páginas se describen con Lenguaje de marca de hipertexto (html)
    - Lenguaje de descripción de documentos de alto nivel
    - Especifica la estructura, pero no la apariencia del documento
    - Define las secciones, como “encabezado de nivel 1”, “lista”, “texto resaltado”
    - El navegador gestiona la visualización de la página en el equipo cliente

## Organización de la Web: tipos MIME

- **Cada documento de la Web tiene un tipo**
  - Tipos MIME (Multipurpose Internet Mail Extensions), en norma http
    - Algunos tipos, como html, texto y algunos gráficos, se muestran directamente en el navegador
    - Otros necesitan programas de ayuda o complementos externos
  - El sistema de tipos web es ampliable. Los nuevos tipos se incorporan fácilmente.
    - El nuevo tipo MIME se define en el servidor
    - El navegador solicita la aplicación de ayuda si no reconoce el tipo MIME
    - El navegador puede sugerir el sitio web desde el que descargar la ap.
    - Descarga e instala la ap. de ayuda y utiliza el nuevo tipo MIME
  - Ampliabilidad del servidor web: programas para generar documentos
    - Las URL pueden apuntar a programas y a páginas
    - Los programas pueden ser simples (mostrar la hora del día) o complejos (bases de datos, análisis)
    - Se utiliza Java, perl, C++, Java Server Pages, Visual Basic, Active Server Pages
    - Las aplicaciones pueden ser simples (mostrar la hora del día) o bases de datos y análisis complejos

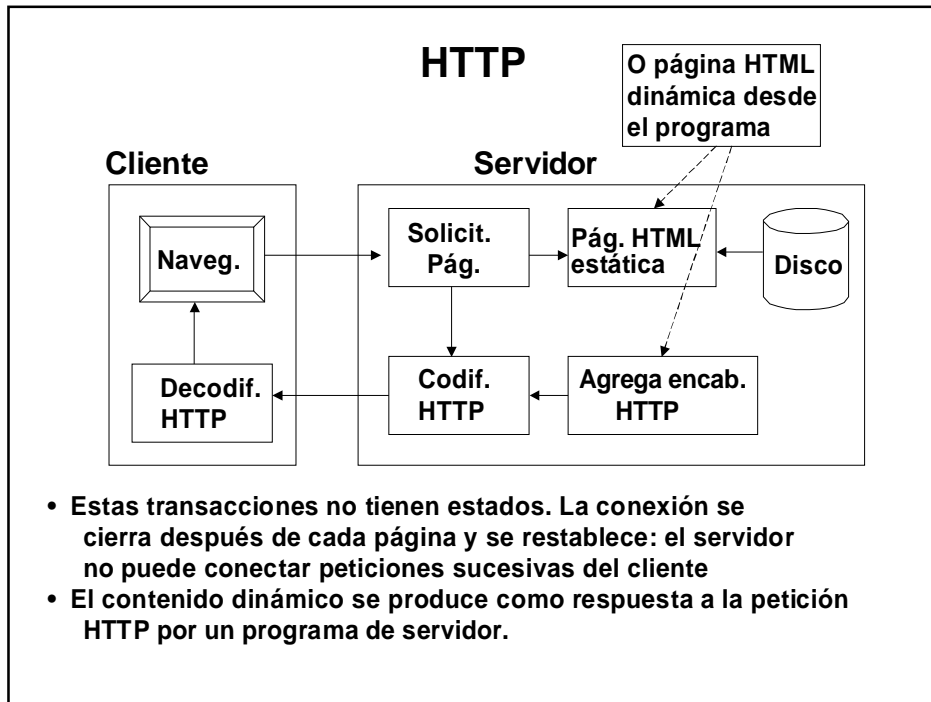
## Ejemplos de tipos MIME

- application/msword Word
- application/pdf Acrobat
- application/vnd.ms-excel Excel
- application/zip Archivo Zip
- audio/basic .au, .snd
- audio/x-wav MS audio
- image/gif GIF
- image/jpeg JPEG
- text/plain Texto sin formato
- text/html HTML
- video/mpeg Vídeo

Consulte <http://www.rfc-editor.org> para obtener una lista actualizada de HTTP, MIME y otras especificaciones de Internet

## Protocolo HTTP

- **HTTP 1.1 es el protocolo actual**
  - Basado en el alfabeto ISO Latin1 (ASCII con ampliaciones para lenguas europeas)
  - HTTP 1.1 agrega conexión persistente tcp/ip, transferencias parciales de documentos sobre HTTP 1.0
- **Cuatro fases:**
  - Conexión abierta: basada en URL
  - Petición: el navegador abre la conexión al servidor y envía:
    - Método de petición (y solicita datos al final si hay una petición POST o PUT)
    - URL,
    - Número de versión de HTTP
    - Información de encabezado (informativa, opcional), que termina con una línea en blanco
  - Respuesta: el servidor procesa la petición y envía:
    - Versión del protocolo HTTP y código de estado
    - Información de encabezado que termina con línea en blanco
    - Texto (datos)
  - Conexión cerrada



## Ejemplos de peticiones HTTP

**Petición típica de navegador: (Telnet de Unix web.mit.edu 80 para verlo)**

**GET /about-mit.html HTTP/1.1**

**Host: web.mit.edu** (requerido)

**Accept: text/html, text/plain, image/jpeg, image/gif, \*/\*** (opcional)

(línea en blanco)

**Respuesta típica del servidor:**

**HTTP/1.1 200 OK**

**Server: Apache/1.3.3 Ben-SSL/1.28 (Unix)**

**Content-Type: text/html**

**Content-Length: 8300**

(línea en blanco)

**<HTML>**

**<HEAD><TITLE>Acerca del MIT</TITLE></HEAD>**

**<BODY>La misión del MIT...</BODY>**

**</HTML>**

También (telnet de Unix):

euro.ecom.cmu.edu

GET /test.html HTTP/1.1

amazon.com

OPTIONS \* HTTP/1.1

## Fase de petición HTTP (naveg. a servidor)

### Descripción de comandos

<b>GET</b>	Solicita doc. especificado (también para 'post')
<b>HEAD</b>	Solicita sólo encabezado de doc. especificado
<b>POST</b>	Solicita que el servidor acepte datos del navegador y genera contenido dinámico
<b>OPTIONS</b>	Obtiene opciones de servidor y de acceso
<b>TRACE</b>	Se utiliza en la depuración
<b>PUT</b>	Sustituye el doc. del servidor con datos del navegador
<b>DELETE</b>	Elimina el doc. especificado en el servidor

- GET es la petición más frecuente de los navegadores
- POST se utiliza para procesar formularios de cumplimentación
- HEAD se usa en motores de búsqueda para buscar págs. activas
- No hay seguridad ni autenticación en HTTP ('Basic' envía la contraseña vacía).
  - Use *Secure Socket Layer* (SSL) para cifrar sus intercambios en la web

## HTML

- Subconjunto de lenguaje de marcado de estructura generalizada (SGML), un lenguaje de descripción de documentos
  - SGML es una norma ISO
- La versión actual de HTML es la 4.01
  - Consulte [www.w3.org](http://www.w3.org) para obtener los borradores, normas y debates más recientes
  - Nuevas funciones de la norma HTML 4.0
    - Hojas de estilos, internacionalización, acceso para usuarios con problemas, marcos, tablas y formularios mejorados

## Ejemplo de lenguaje de marcado de hipertexto (html)

```
<HTML>
<HEAD>
<TITLE> Bienvenido al sistema de piezas de aviones </TITLE>
</HEAD>
<BODY>
<H1> Bienvenido al sistema de piezas de aviones </H1>
Este sistema gestiona pedidos de piezas de aviones y globos.
  Trabajamos con piezas de aviones y de globos de todos los
  fabricantes. Cumplimos las últimas normativas
  estadounidenses.
<P>
La utilización de este sistema está sujeta a las <A HREF=
  MITRule.html> reglas y normativas del MIT. </A>
</BODY>
</HTML>
```

(Aircraft1.html)

## HTML

- Las etiquetas (p.ej. <ETIQ>) no se muestran: se reflejan directamente en el navegador
- Suelen ir en parejas (p.ej. <ETIQ> y </ETIQ>) para delimitar la sección
- Algunas etiquetas tienen atributos (<A HREF=abc.html> </A>)
- El documento HTML comienza con <HTML> y termina con </HTML>
  - Dos secciones dentro del documento: ENCABEZADO y CUERPO
  - El encabezado contiene información identificativa que no se muestra
  - El cuerpo sí se muestra y tiene formato:
    - Párrafo <P>
    - Niveles de encabezado del 1 al 6 <H1> a <H6>
    - Salto de línea y retorno de carro <BR>
    - Anclaje <A>, colocado en texto o gráficos; usado para hipervínculos

## Ejemplos de etiquetas

- Controlan el aspecto de la página
  - No tan precisas como MS Word u otros editores
  - Pensadas para que las páginas se puedan ver en equipos con distintas capacidades gráficas. Las etiquetas no apoyan asunciones específicas.

<u>Etiqu. apertura</u>	<u>Etiqu. cierre</u>	<u>Definición</u>
<EM>	</EM>	Resaltado (a menudo, cursiva)
<STRONG>	</STRONG>	Resaltado fuerte (a menudo, negrita)
<VAR>	</VAR>	Variable
<CITE>	</CITE>	Cita (de un artículo o libro)
<ADDRESS>	</ADDRESS>	Dirección (identifica info al final del doc)
<CODE>	</CODE>	Fragmento de código de equipo (ignora etiq.)
<i>	</i>	Cursiva (igual que otros estilos físicos)

(Aircraft2.html)

## Ejemplos de etiquetas (cont.)

- Tamaño y color de fuente: <FONT>, <FONT COLOR= name>
- Caracteres especiales: &gt; es >
- Formato de párrafo: <H2 ALIGN=CENTER> texto </H2>
- Texto preformateado: <PRE> (alineación de columnas con exactitud, etc.)
- Líneas horizontales: <HR WIDTH=80% ALIGN= LEFT>
- Listas: pueden ser de varios niveles, etc.
  - Ordenadas (con números) <OL>
  - No ordenadas (con viñetas) <UL>
  - Definición (con sangría) <DL>
- Tablas: <TABLE>, <CAPTION>, <TR> (fila), etc.

(Aircraft3.html)

## Imágenes

---

`<H1> Bienvenido al sistema de distribución de aviones </H1>`

`<A HREF=aircraft.gif>`

`<IMG SRC=thumbnail_aircraft.gif ALT="Servicios de piezas de aviones">`

`</A>`

Bienvenido al sistema de piezas de aviones

---

- Aquí se muestra una pequeña imagen de las piezas del avión, pero permite al usuario descargar una imagen más grande
- La HREF puede apuntar a otros documentos sobre aviones, transporte, etc.

(Aircraft4.html)

## HTTP y HTML

- **HTTP**
  - Sólo es una forma directa de interacción entre navegador y servidor
  - Fue una ampliación muy inteligente del correo electrónico y los protocolos ftp a cargo de Tim Berners-Lee para habilitar navegadores web
  - Paradigma de petición/respuesta
  - Conexión establecida para cada pareja petición/respuesta
  - El protocolo más popular de Internet, definición muy estable
- **HTML**
  - Lenguaje de descripción de texto basado en etiquetas
  - Descripción de alto nivel más que formato específico
  - HTML es estático y esto no basta para las aplicaciones web
  - Muchas ampliaciones y cambios (comandos, Java) para el contenido dinámico

## Formularios HTML

- **Utilizados como aplicaciones cliente para programas de servidor**
  - Java Server Pages, Active Server Pages, servlets, ...
- **Los formularios son controles de interfaz para recopilar datos del usuario y transmitirlos al programa de aplicación del servidor**
  - Son el método de interfaz de usuario más común
  - Se ubican en páginas web que también pueden contener otros elementos
  - JavaScript (no tiene nada que ver con el lenguaje Java) se puede utilizar en formularios HTML para mejorar su funcionamiento
  - Los *applets* de Java ofrecen una interfaz de usuario más completa, pero suponen mucho trabajo y tienen restricciones de seguridad
  - Todos ellos se ejecutan en el navegador y son componentes de la interfaz del usuario

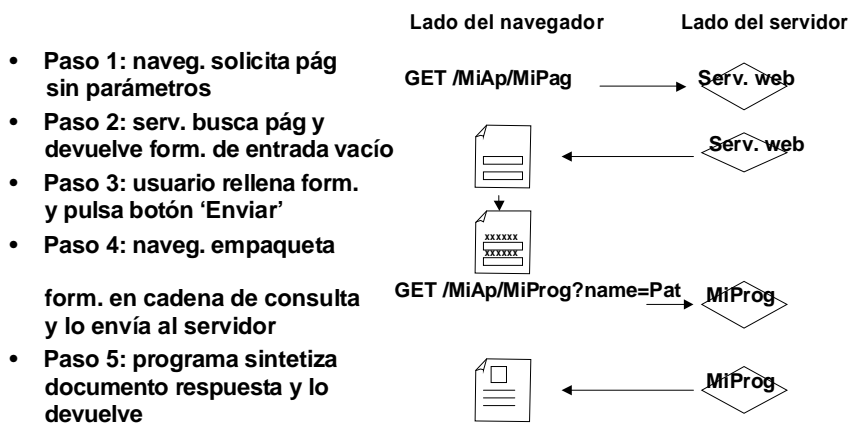
## Cómo transmiten datos los formularios HTML

- **Los formularios permiten colocar una serie de controles en la página**
  - Cada control tiene un nombre y un valor
  - Todo el formulario está asociado a la URL de un programa en el lado del servidor que procesa los datos de entrada
  - Los datos del formulario se envían cuando el usuario pulsa el Botón 'Enviar' (control)
  - Los datos se envían a la URL como una cadena de la forma:
    - Nombre1=Valor1&Nombre2=Valor2&...NombreN=ValorN
  - Si los datos se envían con el comando HTTP GET, se adjuntan al final de la cadena GET detrás de un ?:
    - GET /Index.html?Nombre1=Valor1...
  - Si los datos se envían con un comando HTTP PUT, se envían después de la línea en blanco, como la cadena original
  - Los programas de servidor (ASP, JSP, servlet, etc.) tienen métodos para extraer los datos de la cadena y usarlos en el programa

## Etiquetas de HTML 4.0 para formularios

<u>Etiqueta</u>	<u>Tipo</u>	<u>Definición</u>
<FORM>		Inicia un formulario
<INPUT TYPE=	text	Línea sencilla de entrada de texto
	password	Línea sencilla de entrada de contraseña
	file	Archivo para cargar con botón 'Examinar'
	checkbox	Casilla de verificación
	radio	Botón de opción (cuadro de opción)
	image	Imagen que actúa como botón
	hidden	seguim. usuario, almacena entrada predef.
	submit	Botón de envío para el formulario
	reset	Botón para restaurar valores predeter.
<SELECT>		Cuadro de lista o cuadro combinado
<OPTION>		Elem. de lista despleg. o menú emergente
<TEXTAREA>		Inicia campo de entrada de texto de varias líneas

## Interacción entre navegador y servidor



Cada validación dinámica y cumplimentación requiere la interv. del servidor.

- Si las elecciones posteriores dependen de las anteriores, etc.
- JavaScript parecía ser la solución; ahora los son XML, XSLT

## Java en servidores

- **Servlets de Java. Implementan los pasos siguientes:**
  - **Leen datos enviados por el navegador**
    - Normalmente introducidos por el usuario, pero podrían ser de un *applet* o de JavaScript
  - **Busca otra información de la petición HTTP**
    - Capacidades del navegador, *cookies*, *host*, etc. de encabezados HTTP
  - **Genera resultados**
    - Acceso a base de datos, ejecución del programa (posiblemente a través de CORBA/COM), etc.
  - **Da formato a los resultados como HTML u otro documento**
    - Los *servlets* pueden generar GIF, gzip y muchos tipos MIME
  - **Define parámetros de respuestas HTTP en encabezados**
  - **Devuelve documentos al navegador**
- La máquina virtual de Java (JVM) ejecuta los *servlets* en el servidor
- Éstos han evolucionado desde los *applets* (uso limitado) hasta los *servlets* (muy útiles, seguros y normalizados)

## Ejemplo de servlet

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class Hello100 extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String docType =
            "<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 " +
            "Transitional//EN">\n";
        out.println(docType +
            "<HTML>\n" +
            "<HEAD><TITLE>Hola 1.00</TITLE></HEAD>\n" +
            "<BODY>\n" +
            "<H1>Hello 1.00</H1>\n" +
            "</BODY></HTML>");
    }
}
```

## Java Server Pages (JSP)

- **Nos permiten mezclar HTML estático con contenido generado dinámicamente desde *servlets***
  - Muchas páginas web son fundamentalmente estáticas
  - Las partes intercambiables se encuentran sólo en unos pocos lugares de la página
- **Un *servlet* requiere que generemos toda la página dinámicamente cada vez**
- **JSP permite que el diseñador web escriba HTML estático y deja los cabos para el contenido dinámico**
- **Los programadores de Java pueden escribir entonces los cabos**
- **Las Active Server Pages son muy parecidas (Microsoft)**

## Ejemplo de JSP

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
    Transitional//EN">
<HTML> <HEAD>
<TITLE>JSP Example</TITLE>
<LINK REL=STYLESHEET
    HREF="JSP-Styles.css"
    TYPE="text/css">
</HEAD>
<BODY>
<H2>JSP Expressions</H2>
<UL>
    <LI>Current time: <%= new java.util.Date() %>
    <LI>Your hostname: <%= request.getRemoteHost() %>
    <LI>Your session ID: <%= session.getId() %>
    <LI>The <CODE>testParam</CODE> form parameter:
        <%= request.getParameter("testParam") %>
</UL>
</BODY>
</HTML>
```

## Ejemplo de JSP 2

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD> <TITLE>Color Testing</TITLE> </HEAD>
<%
String bgColor = request.getParameter("bgColor");
boolean hasExplicitColor;
if (bgColor != null) {
    hasExplicitColor = true; }
else {
    hasExplicitColor = false;
    bgColor = "WHITE";}
%>
```

## Ejemplo de JSP 2 (cont.)

```
<BODY BGCOLOR="<%= bgColor %">
<H2 ALIGN="CENTER">Color Testing</H2>
<%
if (hasExplicitColor) {
    out.println("Ha facilitado un color de fondo explícito de " +
        bgColor + "."); }
else {
    out.println("Usando el color de fondo predeterminado de WHITE. " +
        "Facilite el atributo de petición bgColor para probar " +
        "un color estándar, un valor RRGGBB o para ver " +
        "si su navegador acepta nombres de colores X11.");
}
%>
</BODY>
</HTML>
```

## Resumen: Entorno de Java (J2SE, J2EE)

