

Tema 2. Internet Básico

Jesús María Aransay Azofra

Sistemas Informáticos

Universidad de La Rioja 2010/2011

Índice

1. Breve historia de Internet
 2. ¿Qué es Internet?
 3. Organización de Internet
 4. Los estándares de Internet. Los RFC's
 5. Creación de páginas web
 1. HTML. Definición y Evolución
 2. Elementos fundamentales de un documento HTML
 3. Elementos de HTML de contenido. Separando contenido de estilo en páginas web
 4. CSS (Cascading Style Sheets)
 6. La pila de protocolos TCP/IP
 7. Capa de aplicación: protocolos de aplicación (http, ftp, smtp, imap, pop)
 8. Capa de transporte: TCP y UDP
 9. Capa de red o de Internet
 10. Sistema de nombres de dominios (DNS)
 11. URL's
 12. Seguridad en Internet: introducción
 13. Configuración de una red IP
-

2.1 Breve historia de Internet

- Internet nace a partir de ARPANET (Advanced Research Projects Agency Network) a iniciativa de DARPA (Defense Advanced Research Projects Agency) dependiente del Departamento de Defensa de los Estados Unidos.
 - En la década de los 60, el Pentágono crea DARPA para desarrollar técnicas de conexión entre ordenadores. Objetivo: construir una red amplia de ordenadores descentralizada y multidireccional. Se desarrollan unos protocolos de conexión y de comunicaciones de conmutación de paquetes para el intercambio de información.
 - Durante los años 70, ARPANET crece en número de ordenadores y actividades, desarrollando nuevos protocolos (intercambio de noticias, conexión remota, TCP/IP, etc.).
 - ARPANET sigue creciendo en número de ordenadores y se descubre su utilidad como medio de comunicación. Según crece el número de redes, los formatos se hacen cada vez más incompatibles.
-

2.1 Breve historia de Internet

- A principios de los años 80 ARPANET conecta unos 100 ordenadores que usan la familia de protocolos TCP/IP. Otras redes independientes como CSNET (Computer Science Nerwork) y MILNET (red militar del departamento de defensa) utilizan también protocolos TCP/IP para interconectar sus equipos.
- En 1983 se interconectan ARPANET, CSNET y MILNET naciendo INTERNET. Los protocolos TCP/IP permitieron comunicar ordenadores UNIX, MS-DOS o MacOS.
- En 1986 nace NSFnet (National Science Foundation) para facilitar el acceso a cinco grandes centros de supercomputerización. Esta red privada se convirtió en la espina dorsal de Internet.
- La gestión de Internet se refuerza en 1992 con la creación de la Internet Society (ISOC). Su objetivo será consensuar las acciones de extensión de Internet.
- http://www.elpais.com/articulo/Pantallas/Internet/agota/direcciones/elpepurtv/20101004elpepirtv_2/Tes

2.2 Qué es Internet

Siguiendo la definición en [RFC1462]
(<ftp://ftp.rfc-editor.org/in-notes/rfc1462.txt>):

- ❑ “Una red de redes basada en los protocolos TCP/IP”
 - ❑ “Una comunidad de gente que usa y desarrolla estas redes”
 - ❑ “Una colección de recursos que pueden ser alcanzados en esas redes”
-

Qué constituye Internet

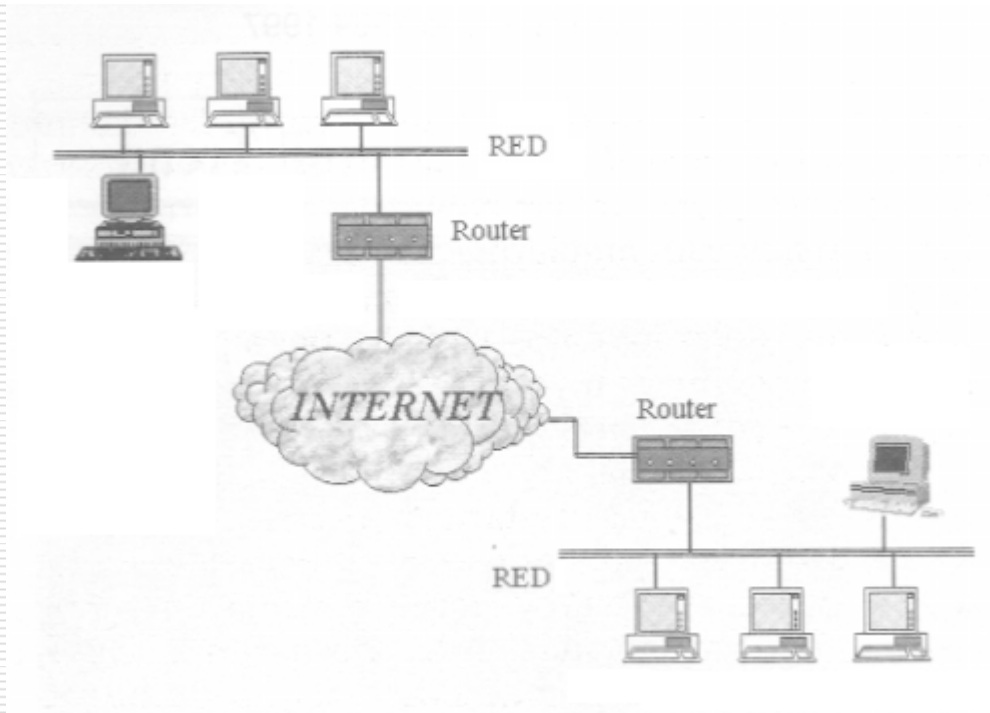
Siguiendo de nuevo [RFC1462]:

“Todas las redes, usando el protocolo IP, que cooperan para formar una red sin fisuras para los usuarios colectivos”

Máquinas que constituyen Internet:

- ❑ Routers: ordenadores que controlan y distribuyen la información a través de las redes
 - ❑ Servidores: ordenadores que ofrecen servicios (mail, http, ftp...)
 - ❑ Ordenadores de usuario: forman parte de LANs, MANs o WANs y permiten acceder a servicios (servidores) u otros equipos de la red
-

Qué constituye Internet



¿Qué falta en la imagen?

2.3 Organización de Internet

- Para poder acceder a Internet se requiere un ISP (Internet Service Provider). Cada ISP dispone de su red propia (o la alquila), y las redes intercambian información.
 - La red está descentralizada. Distintas redes forman parte de ella (por ejemplo, RedIRIS es una red a nivel nacional para instituciones de enseñanza e investigación). Más ejemplos en <http://www.rediris.es/lared/>.
 - La mayor o menor importancia de las redes crea entre ellas una cierta jerarquía (NSFNET, NSINET, ESNET...)
-

2.3 Organización de Internet

ISOC (Internet Society, www.isoc.org): organización sin ánimo de lucro para la cooperación y coordinación de Internet. Sus objetivos:

- Fomentar el crecimiento de Internet
- Desarrollar estándares
- Controlar la correcta administración de recursos de Internet

Iniciativas para 2008-2010:

- Facilitar el acceso a Internet
 - Internetworks: evolucionar infraestructuras y mejorar las direcciones globales (IPv4 hacia IPv6)
 - Confianza y fiabilidad
-

2.3 Organización de Internet

Dentro de ISOC encontramos:

- ❑ [IANA \(Internet Assigned Numbers Authority\)](#): responsable de protocolos, IPs, nombres de dominio, etc
- ❑ [IETF \(Internet Engineering Task Force\)](#): comunidad abierta preocupada por el diseño, arquitectura y evolución de Internet
- ❑ [El IAB \(Internet Architecture Board\)](#): grupo asesor técnico de ISOC responsable de la edición y publicación de RFCs (Request for Comments)

Los NIC (Network Information Center) se encargan en cada zona geográfica de:

- ❑ Asignación de IPs
 - ❑ Gestión de dominios, nombres de dominios y DNS
-

2.4 Los estándares de Internet. Los RFC's

RFC (Request for Comments): Definen los estándares de Internet sobre protocolos y servicios de red. Disponibles en <http://ietfreport.isoc.org/rfcindex.html>

Algunos de ellos:

2616: Hypertext Transfer Protocol – HTTP/1.1

5024: File Transfer Protocol 2.0

3502: IMAP

...

2.5 Creación de páginas web

1. HTML. Definición y Evolución
 2. Elementos fundamentales de un documento HTML
 3. Elementos de HTML de contenido. Separando contenido de estilo en páginas web
 4. CSS (Cascading Style Sheets)
-

HTML. Definición y Evolución

HTML (Hypertext Markup Language): lenguaje de marcado que permite describir la estructura y contenido en forma de texto de los documentos html

HTML se basa en tags o etiquetas, textos rodeados de los símbolos “<”, “>” que suelen contener atributos y datos

HTML permite la integración de contenidos que no son texto como imágenes, sonidos, vídeos

Los navegadores permiten entender y mostrar imágenes, scripts, sonidos, ficheros de texto, pdfs

HTML. Definición y Evolución

- 1980s: Nace de la mano de Tim Berners-Lee.
 - 1991: primer documento con descripción de HTML (22 tags), subconjunto de SGML.
 - 1991: primer navegador (modo texto) para Unix.
 - 1993: HTML 2 incorpora etiquetas para imágenes, formularios, tablas... (no consigue estandarización)
 - 1995: IETF publica el primer estándar oficial de HTML (3.0).
 - 1996: W3C (World Wide Web Consortium) asume la estandarización. 1997: HTML 3.2 incorpora applets y texto que fluye alrededor de las imágenes.
 - ...
-

HTML. Definición y Evolución

- 1998: HTML 4.0. Incorporación de hojas de estilos CSS y de scripts en las páginas web.
 - 1999: HTML 4.01.
 - 2004: Apple, Mozilla y Opera organizan WHATWG (Web Hypertext Application Technology Working Group) para elaborar el HTML 5.0
 - 2008: W3C retoma HTML 5.0
-

HTML y SGML

SGML (Standard Generalized Markup Language): lenguaje para la organización y etiquetado de documentos estandarizado en 1986.

HTML, XML, RTF o Postscript son lenguajes creados a partir de SGML. Hoy en día está siendo sustituido por XML.

XML

XML (Extensible Markup Language): un lenguaje de marcado para definir lenguajes. No posee elementos propios, cada grupo de usuarios crea lo suyos (a través de un DTD, document type definition).

Ejemplo de DTD (de nombre "ejemplo.dtd"):

```
<!ELEMENT lista_de_personas (persona*)>
<!ELEMENT persona (nombre, fechanacimiento?, sexo?, numeroseguridadsocial?)>
<!ELEMENT nombre (#PCDATA) >
<!ELEMENT fechanacimiento (#PCDATA) >
<!ELEMENT sexo (#PCDATA) >
<!ELEMENT numeroseguridadsocial (#PCDATA)>
```

Ejemplo de XML que usa el DTD anterior:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE lista_de_personas SYSTEM "ejemplo.dtd">
<lista_de_personas>
  <persona>
    <nombre>José García</nombre>
    <fechanacimiento>25/04/1984</fechanacimiento>
    <sexo>Varón</sexo>
  </persona>
</lista_de_personas>
```

XHTML

XHTML (1.0) es una versión de HTML 4.01 basada en XML (en lugar de en SGML)

Diferencias con respecto a HTML:

- ❑ Todas las etiquetas (con contenido) deben ir cerradas
 - ❑ Las etiquetas sin contenido se cierran como `<img ... />`, `<br />`
 - ❑ Todas las etiquetas y atributos deben ir en minúsculas
 - ❑ Permite la inclusión de fragmentos en diversos tipos XML, como MathML o SVG
-

HTML 5

HTML 5 (todavía en desarrollo) especifica las variantes HTML5 y XHTML5

Nuevas características:

- ☐ Incorpora soporte para vídeo y audio nativo
- ☐ Nuevos elementos: article, canvas, footer, header...
- ☐ Elementos rechazados: frame, font, center...

Quizá alcance el status de “Candidate Recommendation” en 2012, aunque varios navegadores (en particular Safari) soportan muchas de sus posibles características.

Elementos fundamentales de un documento HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
  <HEAD>
    <META http-equiv="Content-Type"
        content="text/html; charset=ISO-8859-5">
    <TITLE>Mi primer documento HTML</TITLE>
  </HEAD>
  <BODY>
    <P>¡Hola mundo!
  </BODY>
</HTML>
```

Elementos fundamentales de un documento HTML 4.01

1. Una línea que contiene información sobre la versión de HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

2. Una sección de cabecera declarativa (head):

```
<HEAD>
  <META http-equiv="Content-Type"
    content="text/html; charset=ISO-8859-5">
  <TITLE>Mi primer documento HTML</TITLE>
</HEAD>
```

3. Un cuerpo (body o frameset), con el contenido real del documento:

```
<BODY>
  <P>¡Hola mundo!
</BODY>
```

Información sobre la versión de HTML

Se debe especificar un DTD de entre los siguientes:

1. Strict: no permite ni elementos desaprobados (“deprecated”) ni marcos. En particular, excluye elementos y atributos de presentación
2. Transitional: es el Strict más los elementos desaprobados
3. Frameset: el Transitional más soporte para marcos

Generalmente trabajaremos con Strict:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

El DTD es la definición del tipo de documento. La de Strict puedes encontrarla en <http://html.conclase.net/w3c/html401-es/sgml/dtd.html>

Importancia de los estándares

Acogerse a un estándar (HTML 4.01) nos “asegura” que:

- ☐ Los tags aceptados van a tener similar significado en distintos navegadores (gracias al DTD)
 - ☐ En el caso de 4.01 Strict, el estándar elimina muchos de los elementos (de presentación) que no serán válidos en HTML 5 (compatibilidad hacia atrás)
 - ☐ El contenido de nuestra página está bien formado y estructurado (¿tiene sentido encontrar un celda fuera de una tabla?)
-

Algo sobre tags (o etiquetas) en HTML

- ❑ Los tags html suelen estar definidos por un tag de apertura y otro de cierre, de la forma `<tag>...</tag>`. Hay algunas excepciones como los tags `<br>`, `<img>`, `<meta>`, `<link>`...
 - ❑ Cada tag puede tener una serie de atributos que especifiquen propiedades:
`<tag atr1="value1" atr2="value2">...</tag>`
`<meta name="Author" content="J.Aransay">`
 - ❑ Cada tag puede contener dentro una serie de elementos (o tags) particulares (cuidado al validar).
`<ol>` sólo puede contener uno o más `<li>` dentro;
`<tr>` sólo puede contener uno o más `<td>` o `<th>`
-

Sección de cabecera declarativa (head):

`<HEAD>...</HEAD>` puede contener dentro los elementos:

- ❑ `<title>...</title>`: es obligatorio consignar un único título. El mismo se suele mostrar en la barra del navegador
 - ❑ `<meta ...>`
 - ❑ `<LINK ...>`
-

Uso de metas

Los metas contienen metainformación referente al autor de una página web, la codificación de la misma, descripción... La especificación de metas suele hacerse como:

```
<META name="keywords" lang="en" content="holiday,  
Greece, sunshine">
```

Una de las utilidades de los metas es ayudar a los robots a indexar nuestro sitio web. Puedes encontrar algunos consejos en <http://html.conclase.net/articulos/metadatos>:

- ☐ Proporcionar metas para keywords y description
 - ☐ Proporcionar instrucciones de indexado a robots (con metas o por robots.txt)
-

Uso de metas

Nos detenemos en un meta que permite especificar el tipo de documento y la codificación de caracteres del mismo:

```
<META http-equiv="Content-Type" content="text/html;  
      charset=ISO-8859-5">
```

El atributo content contiene el tipo MIME (Multipurpose Internet Mail Extensions) de la página web. El tipo MIME ayudará a los dispositivos a conocer cuál es el contenido de nuestro fichero. Puedes encontrar una lista exhaustiva de tipos MIME en

<http://www.htmlquick.com/es/reference/mime-types.html>.

Por medio del charset advertimos al navegador del tipo de codificación de caracteres en que ha sido guardada la página web.

Ejemplo de tipos MIME

Send Saved Discard Draft autosaved at 1:03 AM (0 minutes ago)

From: Jesus Aransay <jesus-maria.aransay@unirioja.es>

To: "Jesus Aransay" <jmaransay@gmail.com>

[Add Cc](#) | [Add Bcc](#)

Subject: tipos mime de archivos

Insert: [Invitation](#)

 ☒ [practica05_tablas.htm \(text/html\) 5K](#)

☒ [isar-ref.pdf \(application/pdf\) 1539K](#)

☒ [license.txt \(text/plain\) 2K](#)

☒ [luna1.jpg \(image/jpeg\) 1K](#)

☒ [lista-miller.doc \(application/msword\) 590K](#)

☒ [HxD.exe \(application/octet-stream\) 1630K](#)

[Attach another file](#)

[Rich formatting »](#)

--
Jesús María Aransay Azofra
Universidad de La Rioja
Dpto. de Matemáticas y Computación
tlf.: (+34) 941299438 fax: (+34) 941299460
mail: jesus-maria.aransay@unirioja.es ; web: <http://www.unirioja.es/cu/jearansa>
Edificio Luis Vives, c/ Luis de Ulloa s/n, 26004 Logroño, España

Uso del tag link

- A pesar de sus diversos usos, lo usaremos exclusivamente para “enlazar” hojas de estilo:

```
<link rel="stylesheet" href="miestilo.css"  
      type="text/css">
```

Atributos comunes a cualquier tag

Antes de especificar los distintos elementos que componen el cuerpo de una página html, vamos a enumerar una serie de atributos que serán comunes a todos los tags de la misma (salvo que especifiquemos lo contrario).

Algunos serán de utilidad a la hora de insertar estilos:

- ❑ `id="nombre"`: asigna un nombre a un elemento. El nombre debe ser único dentro del código HTML.
 - ❑ `class="nombre"`: asigna un nombre de clase a un elemento. Puede haber varios elementos que pertenezcan a una misma clase.
 - ❑ `style="estilo"`: permite especificar estilos en línea. Desaconsejado (no separa contenido de presentación).
 - ❑ `lang="es", "en"`: permite especificar el lenguaje de la página, de un párrafo, de un enlace...
 - ❑ `title="..."`: información consultiva sobre el elemento para el cual se establece
-

Cuerpo con el contenido real del documento

`<body>...</body>`:

- ❑ debe estar formado por uno o mas elementos de nivel de bloque (si incluimos directamente elementos de línea, como `span`, `a`, `em`, `strong`... no validará)
- ❑ puede contener cualquiera de los atributos `id`, `class`, `style`, `lang`, `title`

Nota: en general, los tags o elementos html se dividen en tags de nivel de bloque y tags de nivel de línea. Pasamos a ver su significado.

Elementos de nivel de bloque

Los elementos de nivel de bloque pueden contener otros elementos de nivel de bloque o elementos de nivel de línea.

Los elementos de nivel de bloque más usados son (veremos más adelante su utilidad):

- ❑ H1...H6: cabeceras
 - ❑ p: párrafo
 - ❑ dl, ol, ul: listas
 - ❑ table: tablas
 - ❑ form: formularios
 - ❑ div: contenedor genérico
 - ❑ address: dirección
 - ❑ blockquote: citas
-

Elementos de nivel de bloque

Los siguientes elementos también pueden contener elementos de nivel de bloque, y por tanto son considerados también de nivel de bloque:

- ❑ li: elemento de lista (ol, ul)
 - ❑ td: dato de tabla (table, tr)
 - ❑ th: cabecera de tabla (table, tr)
 - ❑ dt: término de definición (dl)
 - ❑ dd: descripción de definición (dl)
-

Elementos de nivel de línea

Los elementos de nivel de línea pueden contener datos u otros elementos de línea (pero no elementos de bloque).

Algunos de los más utilizados son:

- ☐ a: anclas
 - ☐ br: salto de línea
 - ☐ cite: citas
 - ☐ code: código fuente
 - ☐ em: enfatizado
 - ☐ img: imágenes
 - ☐ input: entrada para formularios
 - ☐ select: formularios
 - ☐ span: contenedor de línea
 - ☐ strong: énfasis fuerte
-

Contenedores genéricos

Cualquier tag da lugar a un contenedor, que podemos definir como los elementos que van dentro de ese tag. Por ejemplo, el tag `<p> Hola!</p>` sirve en este caso para:

1. Definir un párrafo
2. Generar un contenedor para el texto “Hola!”

HTML 4.01 define dos contenedores genéricos, cuya única misión es añadir estructura a una página web. Nos serán de gran utilidad a la hora de introducir estilos en la misma:

- ❑ `<div>...</div>`: contenedor de nivel de bloque
 - ❑ `<span>...</span>`: contenedor de nivel de línea
-

Cabeceras

H1...H6 (<H1>...</H1>): especifican cabeceras de tamaño 1 a 6 (mayor a menor).

- ❑ Exige tag de apertura y de cierre (ambos obligatorios)
 - ❑ Atributos: los ya vistos (id, class...)
 - ❑ Elementos: cualquiera de los de nivel de línea
 - ❑ Se suelen usar de forma jerárquica (no se usa H4 si no se van a usar también H1, H2 y H3)
-

Párrafos y saltos de línea

p (<p>...</p>): representa un párrafo.

- ❑ No es obligatorio el tag de cierre
- ❑ Puede contener elementos de línea, no elementos de bloque (ni siquiera otro p)

br (
): salto de línea

- ❑ Prohibido el tag de cierre
-

Anclas (origen o destino de vínculos)

a (<a>...):

- ☐ Obligatorio el tag de cierre
- ☐ Elementos: cualquiera de línea
- ☐ El contenido del tag se convierte en origen o destino del vínculo
- ☐ Atributos:
 - name o id: nombre del destino de un vínculo
 - href: URL del destino de un vínculo
 - type: MIME del destino
 - charset: codificación del destino

El atributo id="nombre" puede usarse en cualquier tag o elemento para crear un destino de vínculo

Ejemplos de enlaces

Estableciendo el origen de vínculos:

- `<H2><a name="seccion2">Sección 2</a></H2>`
- `<H2 id="seccion2">Sección 2</H2>`

Estableciendo el destino de vínculos:

- Vínculos internos (añadir “#” al nombre)
`<a href="#seccion2">enlace a la sección 2</a>`
 - Vínculos externos:
`<a href="http://otramaquina.com">enlace a otra máquina</a>`
 - Vínculos internos en documentos externos:
`<a href="http://otramaquina.com#seccion2"> enlace a otra máquina, sección 2</a>`
-

Enlaces relativos y absolutos

Imaginamos que nuestra página web es:

`http://belenus.unirioja.es/~usuario/index.htm`

Si queremos enlazar la página web de nombre SI.htm en la misma carpeta podemos usar

1. Enlace absoluto:

`<a href="http://belenus.unirioja.es/~usuario/SI.htm">Enl.</a>`

2. Enlace relativo:

`<a href="SI.htm">Enl.</a>`

El enlace relativo “infiere” el protocolo, la máquina y la ruta correspondiente y construye una URL válida.

Enlaces relativos y absolutos

Podemos especificar un elemento

```
<BASE href="http://mimaquina/index.htm">
```

dentro de `<head>...</head>` que se usa como base para las url relativas (por defecto, BASE toma el valor del fichero actual).

Ventajas del uso de URL relativas:

- ☐ Portabilidad de la página web

Desventajas del uso de URL relativas:

- ☐ Problemas con algunos robots al recorrer los enlaces

Listas

Dos tipos de listas:

1. `<ol>...</ol>`: ordered list (indexada por números, letras...)
 2. `<ul>...</ul>`: unordered list (indexada por cajas, puntos...)
- ☐ Elementos: uno o más `<li>`
 - ☐ Atributos: los conocidos

`<li>...</li>`: list item (puede contener elementos de línea o de bloque)

Ejemplos de listas

Lista ordenada simple

`<ol>`

`<li> Línea de texto 1`

`<li> Línea de texto 2`

`<li> Línea de texto 3`

`</ol>`

1. Línea de texto 1

2. Línea de texto 2

3. Línea de texto 3

Lista desordenada simple

`<ul>`

`<li> Línea de texto 1`

`<li> Línea de texto 2`

`<li> Línea de texto 3`

`</ul>`

□ Línea de texto 1

□ Línea de texto 2

□ Línea de texto 3

Nota: una lista puede contener en un elemento li otras listas

Listas de definiciones

`<dl>...</dl>`: como elemento contiene
pares de tags `<dt>...</dt>`,
`<dd>...</dd>`

- ☐ `<dt>...</dt>`: término definido
 - ☐ `<dd>...</dd>`: descripción de la definición
-

Ejemplos de listas de definiciones

<p>Diccionario Oxford:</p>

<dl>

<dt> footstep <dd> Seguirle los pasos a alguien

<dt> misplace<dd> Perder algo transitoriamente

</dl>

Resultado:

Diccionario Oxford

footstep

Seguirle los pasos a alguien

misplace

perder algo transitoriamente

Tablas

Pensadas para organizar datos en un formato de filas y columnas de celdas

Citando la especificación HTML 4.01:

“No deberían usarse tablas con la única finalidad de organizar la presentación de los contenidos de un documento (es decir, de crear el "layout"), ya que esto puede ocasionar problemas cuando se represente en un medio no visual. Además, al incluir gráficos, estas tablas pueden forzar a los usuarios a hacer desplazar horizontalmente la pantalla para ver una tabla diseñada en un sistema con una pantalla más grande. Para minimizar estos problemas, los autores deberían usar hojas de estilo en lugar de tablas para organizar la presentación.”

Definición de una tabla

`<TABLE>...</TABLE>`:

Atributos:

- ☐ `summary="texto"`: descripción de la tabla para dispositivos no visuales
- ☐ `width="longitud" ó "porcentaje%"`: anchura de la tabla para dispositivos visuales (mejor en CSS)
- ☐ `border="longitud"`: anchura del borde de la tabla y celdas (mejor en CSS)
- ☐ `cellspacing="longitud"`: distancia entre celdas y al marco de la tabla (mejor en CSS)
- ☐ `cellpadding="longitud"`: distancia entre el texto de una celda y los márgenes (mejor en CSS)

Nota: consignar siempre las unidades de la longitud (px)

Definición de una tabla

`<table>...</table>`

Elementos:

- ❑ `<caption>...</caption>`: título de la tabla.
Por defecto se sitúa encima de ella
 - ❑ `<colgroup>...</colgroup>`: grupos de columnas que van a compartir cierta propiedad (o atributo):
 - `span`="número de columnas"
 - `width`="anchura de las columnas" (longitud o porcentaje)
-

Conjuntos de filas en la tabla (opcionales)

`<THEAD>...</THEAD>, <TFOOT>...</TFOOT>:`
definen un bloque contenedor para el conjunto de filas de cabecera y pie de tabla
`<TBODY>...</TBODY>:` define un bloque para el conjunto de filas del cuerpo de la tabla

Elementos:

`<TR>...</TR>` (table row): cada uno de los bloques de la tabla (thead, tfoot y tbody) debe contener al menos una fila cada uno

Celdas en cada fila

Cada <TR> está formado por (al menos una):

- ❑ <TH>: celda de encabezado (negrita)

ó

- ❑ <TD>: celda de datos

Atributos (mejor dejarlos para CSS):

- ❑ align="left,center,right": alineación del texto en la celda

- ❑ valign="top,middle,bottom"

Estos atributos pueden ir especificados también en <TR> (afectando a toda una fila), <TBODY>, <TFOOT>, <THEAD> (afectando a todo el conjunto de filas) y en <colgroup> (afectando a todo el conjunto de columnas)

Celdas en cada fila

Atributos de `<th>` y `<td>`:

- ❑ `rowspan="4"`: hace que una celda se extienda 4 filas
 - ❑ `colspan="6"`: hace que una celda se expanda 6 columnas
-

Estructura básica de tabla

```
<TABLE border="..." summary="..." width="..." cellpadding="..."
  cellspacing="...">
  <CAPTION>...</CAPTION>
  <COLGROUP span="..." width="..."></COLGROUP>
  <COLGROUP span="..." width="..."></COLGROUP>
  <THEAD align="..." valign="...">
    <TR> <TD>...<TD>...<TD>...
  </THEAD>
  <TFOOT align="..." valign="...">
    <TR> <TD>...<TD>...<TD>...
  </TFOOT>
  <TBODY>
    <TR> <TH>...<TD>...<TD>...<TD>...
    <TR> <TH>...<TD align="...">...<TD valign="...">...<TD>...
  </TBODY>
</TABLE>
```

Algunos ejemplos de rowspan y colspan

```
<TABLE border="1">  
  <TR><TD>1<TD rowspan="2">2<TD>3  
  <TR><TD>4<TD>6  
  <TR><TD>7<TD>8<TD>9  
</TABLE>
```

1	2	3
4		6
7	8	9

```
<TABLE border="1">  
  <TR><TD>1<TD>2<TD>3  
  <TR><TD colspan="2">4<TD>6  
  <TR><TD>7<TD>8<TD>9  
</TABLE>
```

1	2	3
4		6
7	8	9

Formularios

Un formulario es una sección de un documento con contenido normal, código, elementos llamados controles y etiquetas en los controles.

Los usuarios modifican los controles (introducen texto, seleccionan opciones...) antes de enviar el formulario (a un servidor web, de correo)

Formularios

`<form>...</form>`:

Atributos más usados:

- ☐ `action="url"` (obligatorio): especifica la acción a ser realizada
 - ☐ `method="get|post"`: método http para enviar formulario
 - ☐ `accept="tipos MIME"`: lista de tipos MIME para subir ficheros
 - ☐ `accept-charset="charset"`: codificación de caracteres aceptada por el servidor
-

Elementos dentro de `<form>...</form>`

- Texto, tablas, listas...

- Controles:

 - Tipo input

 - Tipo select

 - Tipo textarea

Controles tipo input

`<input>` (etiqueta final prohibida):

Atributos:

- ☐ `type="text|password|checkbox|radio|submit|reset|file|hidden|image|button"`
- ☐ `name="nombre":` nombre del control

Los demás atributos son dependientes del tipo de control

Ejemplos de uso

<http://www.webexperto.com/articulos/art/111/todo-sobre-formularios-en-html/>

Controles tipo select

`<select>...</select>`:

Atributos:

- ❑ `size="número"`: filas visibles
- ❑ `multiple`: permite selecciones múltiples

Elementos:

❑ `<option>...</option>`:

Atributos:

- `selected`: opción por defecto
 - `value`: valor que se envía con el formulario (por defecto es el contenido del formulario)
-

Ejemplos de uso

<http://www.webexperto.com/articulos/art/111/todo-sobre-formularios-en-html/>

Controles tipo textarea

`<textarea>...</textarea>`: especifica un cuadro de texto de múltiples líneas

Atributos:

- ☐ `name="nombre"`: nombre del control
 - ☐ `rows="número"`: número de filas del área
 - ☐ `cols="número"`: número de columnas del área
-

Ejemplos de uso

<http://www.webexperto.com/articulos/art/111/todo-sobre-formularios-en-html/>

CSS

CSS (Cascading Style Sheets): lenguaje (distinto de HTML) usado para definir la presentación de un documento estructurado en HTML o XML

w3c es el encargado de desarrollar las especificaciones formales (1.0, 2.0, 2.1, 3.0)

Objetivo: separar la estructura del documento (por etiquetas HTML) de su presentación (código CSS)

Breve historia de CSS

- ❑ CSS 1.0: publicado en 1996 (hasta el año 2000 no tuvo soporte completo en ningún navegador)
- ❑ CSS 2.0: publicado en 1998, CSS 2.1 fija algunos de los errores en CSS 2.0 (a día de hoy – 2010 -, todavía no tiene soporte completo en ningún navegador, [http://en.wikipedia.org/wiki/Comparison_of_layout_engines_\(CSS\)](http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(CSS))). El estándar todavía no está aprobado
- ❑ CSS 3.0: trabajo en progreso. Soporte pobre en la mayor parte de los navegadores. Está formado por diversas recomendaciones (para selectores, tipografías, matemáticas...)

Nota: por su aceptación en la mayor parte de los navegadores y sus mayores capacidades que CSS 1.0, nosotros trabajaremos con CSS 2.1.

Ejemplo de hoja CSS

```
H1 {font-size: 2em;}
strong {font-weight: bolder;}
table {border-spacing: 2px;}
center {text-align: center;}
:link, :visited {text-decoration:
    underline;}
```

Antes de introducir CSS...

Antes de crear nuestras páginas de estilo, es importante definir correctamente la estructura de nuestro documento HTML

Algunas tácticas:

- ☐ usar elementos div, span...
 - ☐ usar atributos id y class para etiquetar elementos en HTML
-

Cómo introducir CSS en nuestra página HTML

Podemos encontrar tres formas fundamentales de adjuntar estilos a código HTML:

1. Estilo en línea (atributo “style”)
 2. Estilo en cabecera (tag “STYLE”)
 3. Hojas de estilo enlazadas (tag “LINK”)
-

Estilo en línea

Dado cualquier tag html le asignamos un atributo “style” con uno o varios pares “propiedad: valor;”

Ejemplo: `<p style="border-color: blue; border-style: solid; border-width: 2px"> Un párrafo con un borde azul alrededor.</p>`

Estilo en línea. Desventajas:

- ❑ Tiene escasa portabilidad de los estilos entre diferentes páginas web, o incluso entre etiquetas html (mantenimiento muy costoso)
 - ❑ La estructura del código HTML y la presentación CSS siguen estando mezclados
-

Uso del tag “STYLE”:

Dentro de la cabecera html podemos hacer uso del tag “STYLE”, dentro del cual situaremos ternas “selector {propiedad: valor;}”

Atributos del tag “STYLE”:

- ❑ type=“tipo MIME”: obligatorio. Especifica el lenguaje de las hojas de estilo utilizadas
 - ❑ media=“media-descriptor”: opcional. Descriptor del tipo de medio en el que pretendemos que se muestre el estilo elegido
-

Ejemplo de uso del tag “STYLE”

Ejemplo:

```
<head>
```

```
...
```

```
<STYLE type="text/css">
```

```
h1, h2 {font-size: 2em;}
```

```
p {color: red;}
```

```
a:link{text-decoration:none;}
```

```
a:hover{text-decoration:italic;}
```

```
li {display:inline}
```

```
</STYLE>
```

```
</head>
```

Desventajas de uso:

- ☐ Falta de portabilidad del estilo definido entre las diferentes páginas html (mantenimiento costoso)
-

Hojas de estilo enlazadas (LINK ...):

Dentro de la cabecera html hacemos uso del tag `<link>`.

Atributos:

- ❑ `href="URI"`: dirección del recurso enlazado (en nuestro caso, del css)
 - ❑ `type="tipo MIME"`: Especifica el lenguaje de las hojas de estilo utilizadas. Si el navegador no lo soporta, no las descargará
 - ❑ `rel="stylesheet", "alternate stylesheet"`: especificamos cuál de nuestras hojas de estilo debe ser usada, y cuáles quedan disponibles para los usuarios como alternativas
-

Ejemplos de uso:

```
<head>
```

```
...
```

```
<LINK rel="alternate stylesheet"  
      title="letra_grande" href="fuente_grande.css"  
      type="text/css">
```

```
<LINK rel="stylesheet" title="letra normal"  
      href="fuente_normal.css" type="text/css">
```

```
</head>
```

La hoja "fuente_normal.css" sería el estilo por defecto que tomaría la página.

La hoja "fuente_grande.css" sería un estilo alternativo que el usuario podría elegir desde su navegador (en Mozilla, por ejemplo, en Ver -> Estilo de página -> ...)

Ventajas de uso de LINK:

- ☐ Portabilidad: una misma página css se puede aplicar a varias páginas web simultáneamente
 - ☐ Disponibilidad: podemos hacer uso de cualquier página css disponible a través de una url.
 - ☐ Separación entre la presentación (css) y el contenido (html)
 - ☐ Disponemos de la propiedad rel="alternate stylesheet" que permite introducir diversos estilos en una página html (o eliminarlos)
-

Sintaxis básica en CSS

CSS se basa en la definición de reglas de estilo.

Cada regla de estilo se especifica como:

```
selector {propiedad: valor;}
```

También se admite la sintaxis:

```
Selector1, selector2 {  
    propiedad1: valor1;  
    propiedad2: valor2;  
    ...}
```

Selectores genéricos

Ver <http://www.w3.org/TR/CSS2/selector.html#pattern-matching> para una lista completa.

Casos más comunes:

- ❑ *: selector universal. coincide con todos los elementos de HTML
 - ❑ h1: selector de tipo. coincide con todos los elementos de tipo h1
 - ❑ ol > li: selector de hijo. Coincide con todos los elementos de lista (li) que desciendan directamente de una lista ordenada (ol).
 - ❑ p a: selector de descendiente. Coincide con todos los elementos “a” (enlaces) que aparezcan dentro de un elemento “p” (párrafos).
-

Diferencia entre selector de tipo, selector de hijo y selector de descendiente:

```
<ol>
  <li> Un elemento
  <li>
    <ul> <li> Un elemento de la sublista
    </ul>
  <li> Otro elemento
</ol>
```

Tenemos ahora los selectores:

```
li {propiedad: valor;}: coincide con todos los li's de las listas
(dentro de ol y ul)
ol li {propiedad: valor;}: coincide con "Un elemento", "Un
elemento de la sublista" y "Otro elemento"
ol>li {propiedad: valor;}: coincide con "Un elemento",
"Otro elemento"
ul>li {propiedad: valor;}: coincide con "Un elemento de la
sublista"
```

Selectores particulares

Para un id="nombre":

- ❑ `h1#nombre{propiedad:valor;}`: selecciona el h1 etiquetado con el id "nombre"
- ❑ `#nombre h1{propiedad:valor;}`: selecciona todos los h1 dentro del contexto "nombre"

Para una clase class="nombre":

- ❑ `p.nombre{propiedad:valor;}`: todos los párrafos (p) de la clase "nombre"
 - ❑ `.nombre{propiedad:valor;}`: cualquier elemento de la clase "nombre"
-

Pseudo-clases:

- ❑ `h1:link{propiedad:valor;}`: cualquier elemento `h1` que sea un enlace no visitado
 - ❑ `img:visited{propiedad:valor;}`: cualquier elemento `img` que sea un enlace ya visitado
 - ❑ `a:hover{propiedad:valor;}`: cualquier elemento `a` cuando pasamos el puntero del ratón por encima
-

Propiedades CSS

Lista de propiedades:

<http://www.w3.org/TR/CSS2/propidx.html>

Algunos ejemplos: background, background-color, background-image, position, top, right, height, width, font, font-family, font-size, z-index, text-align, text-transform, vertical-align, empty-cells, display...

Nota: muchas no coinciden con su “equivalente” en HTML (vertical-align por valign, text-align por align...). Otras muchas no tienen equivalente HTML

Valores de propiedades

Puedes encontrar en

[http://www.w3.org/TR/CSS2/propidx.](http://www.w3.org/TR/CSS2/propidx.html)

[html](http://www.w3.org/TR/CSS2/propidx.html) la lista completa de valores admisibles y la lista de valores por defecto

Importante: especificar siempre unidades

p {font-size:2;} ¿2 píxeles, 2 em...?

¿Por qué lo de cascada?

Imaginemos la siguiente hoja CSS:

Hoja1.css:

```
em{color: blue;}
```

```
#indice em {color: red;}
```

Hoja2.css:

```
#indice em {color: red;}
```

```
em{color: blue;}
```

¿De qué color serían los fragmentos “em” dentro de “indice” con la “Hoja1”?

¿Y con la “Hoja2”?

¿Por qué lo de cascada?

```
Hoja3.css:  
em{color: blue;}  
em {color: red;}
```

¿Y ahora?

Las reglas más específicas (`#indice em{}`) tienen más peso que las más genéricas (`em{}`)

A igualdad de especificidad (`em{}` y `em{}`), la última regla en el orden que aparezcan en el CSS se aplica

(Más ejemplos y detalles en http://codexexempla.org/curso/curso_3_6.php#cascada)

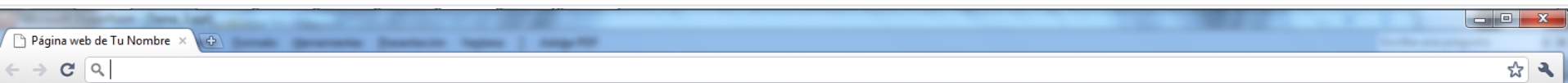
Algunas webs interesantes sobre CSS

- ❑ <http://www.w3.org/TR/CSS2/>: página de la especificación CSS
 - ❑ <http://jigsaw.w3.org/css-validator/>: validador de código CSS
 - ❑ <http://www.ignside.net/man/css/index.php>: tutorial de CSS en castellano
 - ❑ <http://www.camaleoncss.com/>: página en castellano dedicada a alojar estilos css
 - ❑ <http://www.csszengarden.com/>: página en inglés dedicada a alojar estilos css
 - ❑ <http://www.csszengarden.com/tr/espanol/>: traducción al castellano de la anterior web
-

Consejos para insertar CSS en un sitio web (una posibilidad):

(Recordar los pasos de la práctica 7, sobre todo pensando en el trabajo)

1. Partimos del código html con el contenido deseado (pero limpio de estilos) y la estructura necesaria (`<div>...</div>`, `id="..."`):
-



Bienvenido a la página web para la asignatura Sistemas Informáticos de Tu Nombre.

A continuación puedes ver un listado con enlaces a los informes de las prácticas:

- [Práctica 00](#)
- [Práctica 01](#)
- [Práctica 02](#)
- [Práctica 03](#)
- [Práctica 04](#)
- Práctica 5
 1. [Práctica 05 Calendario](#)
 2. [Práctica 05 Horario](#)
 3. [Práctica 05 Formulario](#)
- Añade tantas entradas como necesites

Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario.

*El autor de esta página ha sido [Jesús María Aransay](#)
La última modificación tuvo lugar el 13 de Octubre de 2010.*

2. Definimos posicionamientos de las cajas
(repasar práctica 7: por defecto serán static,
habrá que elegir entre absolutos y relativos)

Conveniente: usar tamaños y coordenadas en
porcentaje, porque así la página se
redimensiona convenientemente.

Importante:

- ☐ Tener cuidado de que unas cajas no pisen a otras
 - ☐ No usar nunca tablas para posicionar elementos en las páginas web
-

3. Insertar imágenes y flotar textos (propiedad float, valores left, right)

Importante: definir el tamaño (en css)
de las imágenes o sus cajas
contenedoras (relativo al del div
donde se incluyan)

Bienvenido a la página web para la asignatura Sistemas Informáticos de Tu Nombre.

A continuación puedes ver un listado con enlaces a los informes de las prácticas:

- [Práctica 00](#)
- [Práctica 01](#)
- [Práctica 02](#)
- [Práctica 03](#)
- [Práctica 04](#)
- Práctica 5
 1. [Práctica 05 Calendario](#)
 2. [Práctica 05 Horario](#)
 3. [Práctica 05 Formulario](#)
- Añade tantas entradas como necesites

- [Práctica 00](#)
- [Práctica 01](#)
- [Práctica 02](#)
- [Práctica 03](#)
- [Práctica 04](#)
- [Práctica 5](#)
 1. [Práctica 05 Calendario](#)
 2. [Práctica 05 Horario](#)
 3. [Práctica 05 Formulario](#)
- Añade tantas entradas como necesites

Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario.

A cartoon illustration of a young boy with brown hair, wearing glasses, a yellow and black striped shirt, and blue pants. He is standing with his hands at his sides.

El autor de esta página ha sido [Jesús María Aransay](#)
La última modificación tuvo lugar el 13 de Octubre de 2010.

4. Modificar presentación de listas,
tablas y otros elementos de bloque
que se usen

Bienvenido a la página web para la asignatura Sistemas Informáticos de Tu Nombre.

A continuación puedes ver un listado con enlaces a los informes de las prácticas:

- [Práctica 00](#)
- [Práctica 01](#)
- [Práctica 02](#)
- [Práctica 03](#)
- [Práctica 04](#)
- [Práctica 5](#)
 - [Práctica 05 Calendario](#)
 - [Práctica 05 Horario](#)
 - [Práctica 05 Fomulario](#)

Añade tantas entradas como necesites

Añade tantas entradas como necesites

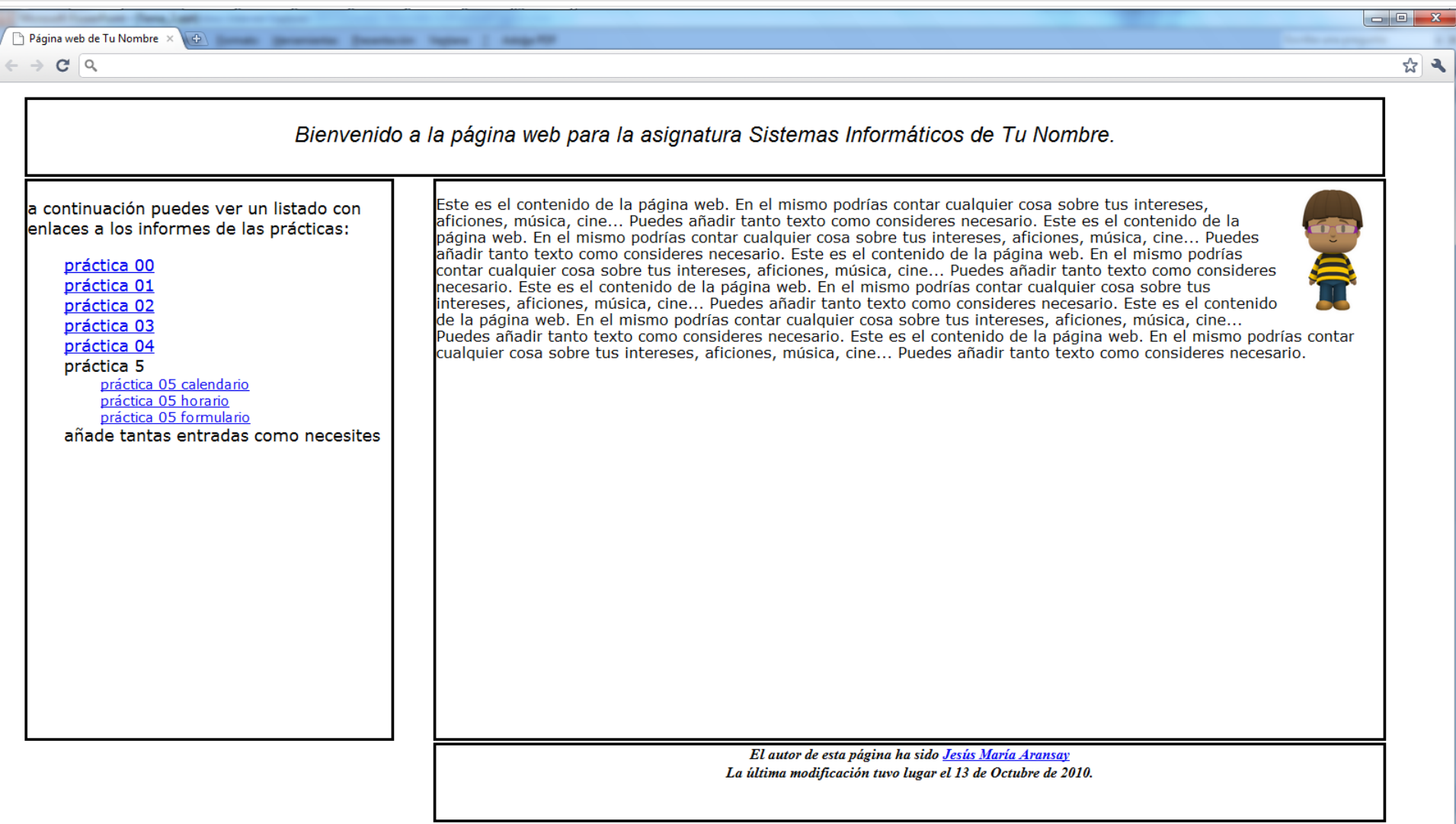
Este es el contenido de la página web. En el mismo podrias contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrias contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrias contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrias contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrias contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrias contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario.

A cartoon illustration of a young boy with brown hair, wearing glasses, a yellow and black striped shirt, and blue pants. He is standing with his hands at his sides.

El autor de esta página ha sido [Jesús María Aransay](#)
La última modificación tuvo lugar el 13 de Octubre de 2010.

5. Modificar fuentes (definir tipos, detallar tamaños, formatos, estilos)

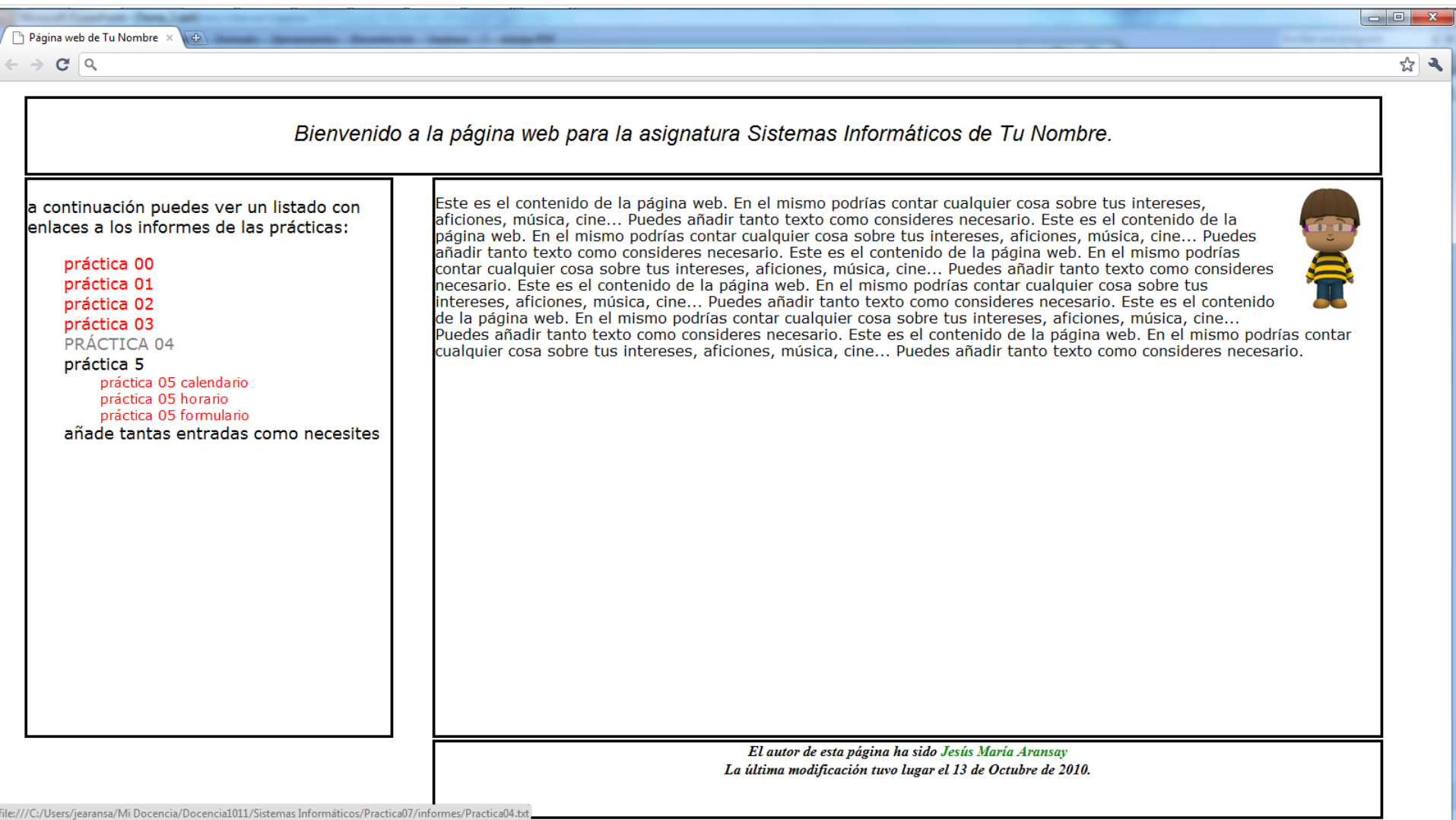
Conveniente: hacer uso de elementos propios de html para estructurar textos (p, em, strong, code, samp, var, cite, abbr, acronym, blockquote, sub, sup, a)



6. Definir estilos propios para los enlaces

Conveniente:

- ☐ modificar sobre todo las pseudo-clases :link, :visited y :hover (en orden, si ponemos :hover antes de :link y :visited su estilo no se asignará a ningún elemento)
 - ☐ resaltar el enlace que está bajo el puntero del ratón (:hover), sobre todo en páginas con mucho texto
-



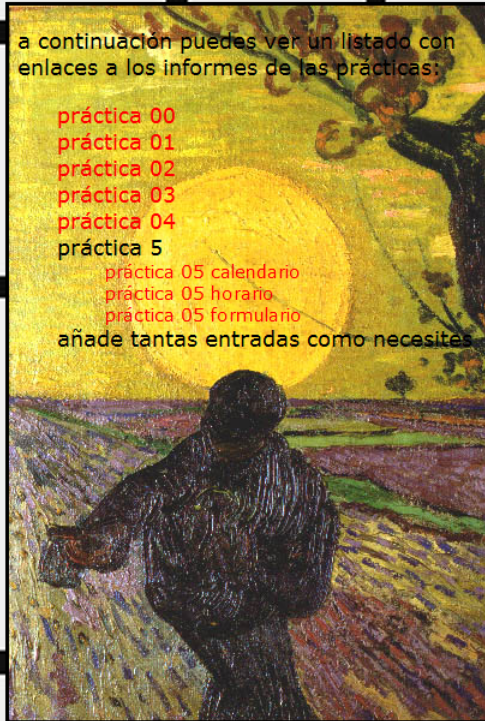
7. Definir fondos de las cajas (background-image, background-color...)

Conveniente: usar imágenes o degradados que permitan la legibilidad del código (buscar siempre contraste entre fondos y texto)



8. Prestar atención a los pequeños detalles (márgenes del texto en las cajas, redimensionamiento de la ventana del navegador...)

Bienvenido a la página web para la asignatura Sistemas Informáticos de Tu Nombre.



a continuación puedes ver un listado con
enlaces a los informes de las prácticas:

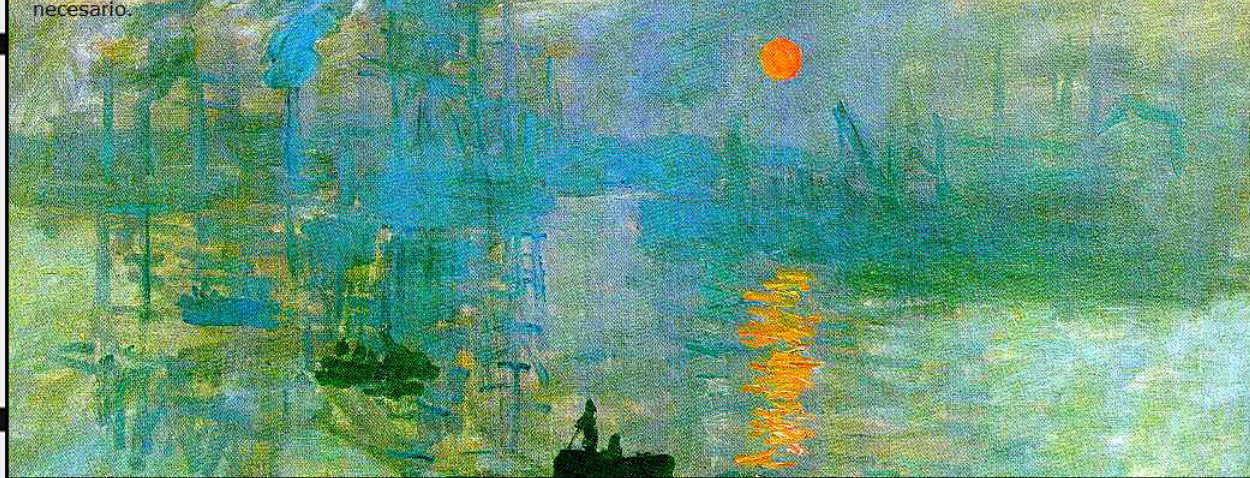
- práctica 00
- práctica 01
- práctica 02
- práctica 03
- práctica 04
- práctica 5
- práctica 05 calendario
- práctica 05 horario
- práctica 05 formulario

añade tantas entradas como necesites

práctica 05 calendario
práctica 05 horario
práctica 05 formulario

añade tantas entradas como necesites

Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario. Este es el contenido de la página web. En el mismo podrías contar cualquier cosa sobre tus intereses, aficiones, música, cine... Puedes añadir tanto texto como consideres necesario.



El autor de esta página ha sido *Jesús María Aransay*
La última modificación tuvo lugar el 13 de Octubre de 2010.

9. Comprobar compatibilidad entre navegadores (no es necesario que la página se vea igual en todos, pero sí que sea visible)

10. Validar con respecto a la especificación CSS usada (2.1, 3.0)

Recuperamos el índice:

1. Breve historia de Internet
 2. ¿Qué es Internet?
 3. Organización de Internet
 4. Los estándares de Internet. Los RFC's
 5. Creación de páginas web
 1. HTML. Definición y Evolución
 2. Elementos fundamentales de un documento HTML
 3. Elementos de HTML de contenido. Separando contenido de estilo en páginas web
 4. CSS (Cascading Style Sheets)
 6. La pila de protocolos TCP/IP
 7. Capa de aplicación: protocolos de aplicación (http, ftp, smtp, imap, pop)
 8. Capa de transporte: TCP y UDP
 9. Capa de red o de Internet
 10. Sistema de nombres de dominios (DNS)
 11. URL's
 12. Seguridad en Internet: introducción
 13. Configuración de una red IP
-

Introducción

En la primera parte del capítulo hemos visto algo sobre la historia y organización de Internet, así como uno de los lenguajes de mayor uso en la misma (html) y el lenguaje de hojas de estilo (css)

En lo que queda del tema nos centraremos en el funcionamiento de la red

2.6 Pila de protocolos TCP/IP

Internet se basa en la conexión entre redes (inter - net) cada una de las cuales pueden tener características muy distintas (redes LAN, WAN, Ethernet, Wireless, Token Ring)

Para permitir la comunicación entre estas redes, hace falta una serie de protocolos que las redes respeten y que les permitan interactuar

Dicho conjunto de protocolos se resume en la pila de protocolos TCP/IP (que reúne más de 100 protocolos distintos)

2.6 Pila de protocolos TCP/IP

La pila de protocolos TCP/IP recibe su nombre de dos de los protocolos fundamentales de la misma (TCP, Transmission Control Protocol, IP, Internet Protocol)

Está basada en el modelo OSI de Interconexión de Sistemas Abiertos, pero permite reducir el número de 7 capas de OSI a simplemente 4

2.6 Pila de protocolos TCP/IP

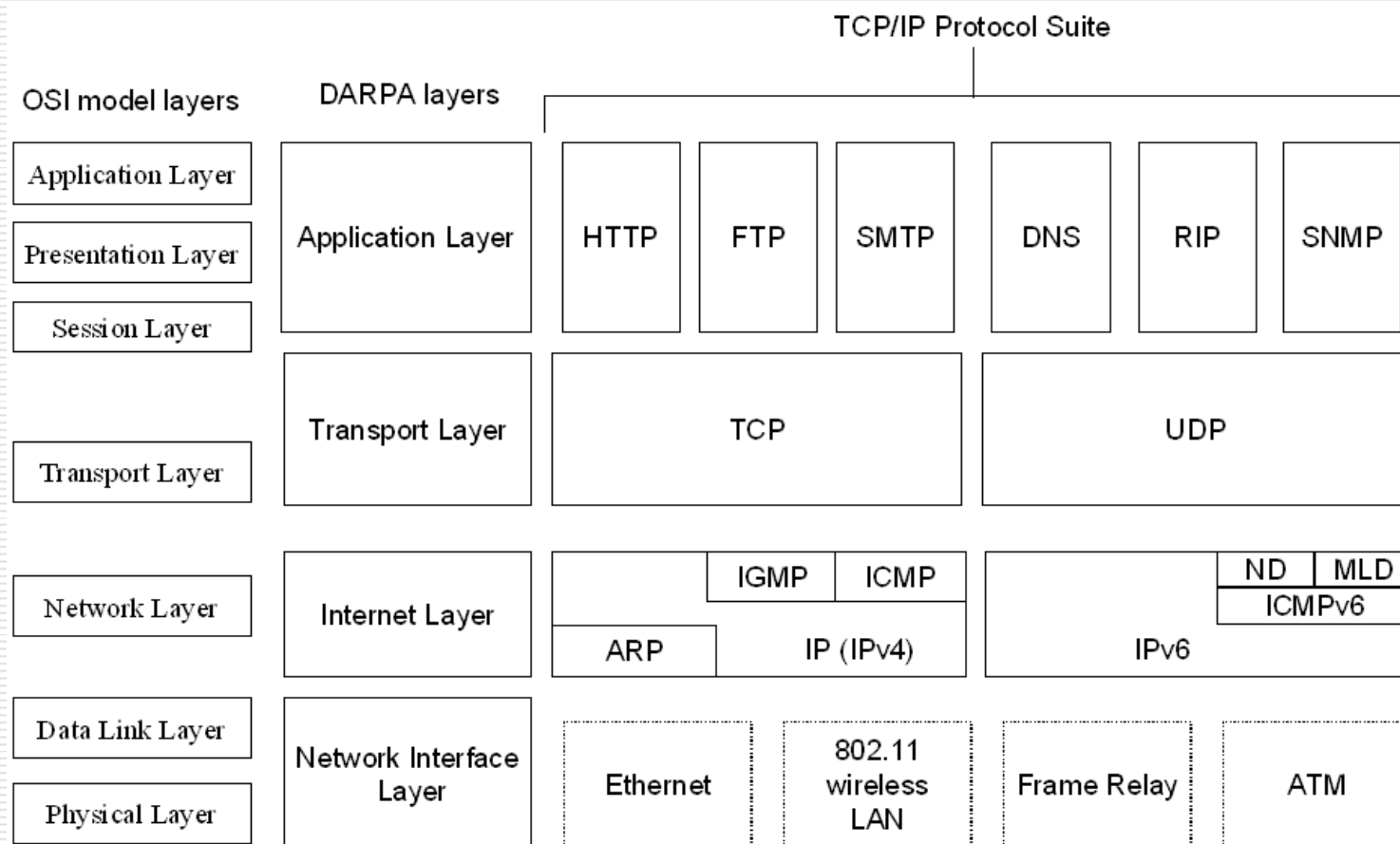
El origen de los protocolos está en un proyecto del DARPA (Agencia de Investigación de Proyectos Avanzados de Defensa) a principios de los 70

Robert Kahn y Vint Cerf desarrollaron un protocolo de comunicaciones que ocultaba las diferencias entre redes (1973); en el mismo, cada ordenador era responsable de sus comunicaciones

Cada red dispone de un router o gateway que oculta las características particulares de su red y se encarga de las comunicaciones con el resto de redes

Primera especificación e implementación de TCP en 1973

2.6 Pila de protocolos TCP/IP

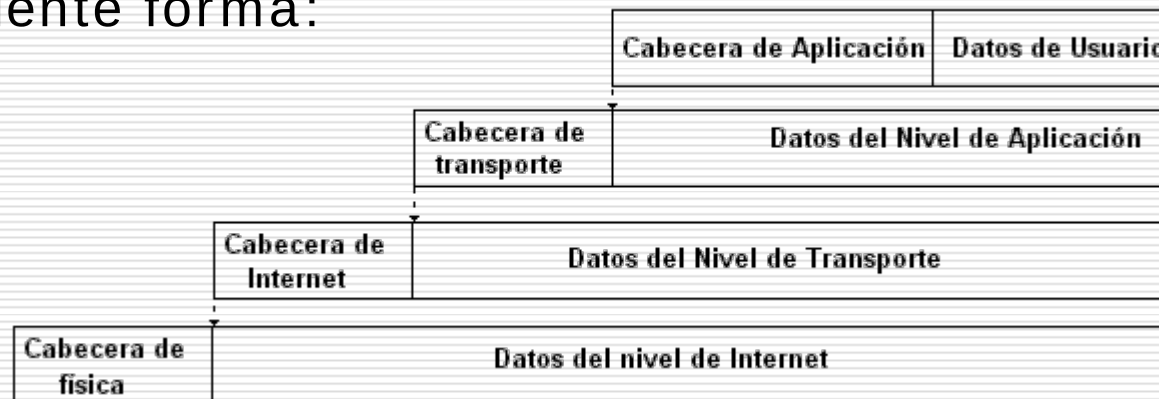


2.6 Pila de protocolos TCP/IP

- La estructuración por niveles de TCP/IP se basa en un conjunto de protocolos que se organizan de manera escalonada. Cualquier mensaje que se origine en un nivel superior (p. ej. capa de aplicación) tendrá que hacer uso de al menos un protocolo de cada una de las capas inferiores (capa de transporte, red, física)
-

2.6 Pila de protocolos TCP/IP

- La información a través de la pila de protocolos TCP/IP (y a través de Internet) se hace a través de paquetes, datagramas o tramas. Cada paquete se construye de la siguiente forma:



- En “datos de usuario” puedes poner una petición a un sitio web, a un servidor dns, a un servidor ftp, mail...
- Es importante notar que no todos los mensajes se originan en la capa de aplicación; puede haber mensajes que se originen en la de transporte (TCP), en la de Internet (ARP)...

Aspecto de un paquete de petición http

411 26.518523 192.168.0.10 193.144.2.30 HTTP GET / HTTP/1.1	
+	Frame 411: 616 bytes on wire (4928 bits), 616 bytes captured (4928 bits)
+	Ethernet II, Src: IntelCor_16:0d:fc (00:24:d7:16:0d:fc), Dst: Thomson_d4:87:4d (00:11:e3:d4:87:4d)
+	Destination: Thomson_d4:87:4d (00:11:e3:d4:87:4d)
+	Source: IntelCor_16:0d:fc (00:24:d7:16:0d:fc)
	Type: IP (0x0800)
+	Internet Protocol, Src: 192.168.0.10 (192.168.0.10), Dst: 193.144.2.30 (193.144.2.30)
	Version: 4
	Header length: 20 bytes
+	Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
	Total Length: 602
	Identification: 0x1e20 (7712)
+	Flags: 0x02 (Don't Fragment)
	Fragment offset: 0
	Time to live: 128
	Protocol: TCP (6)
+	Header checksum: 0x561d [correct]
	Source: 192.168.0.10 (192.168.0.10)
	Destination: 193.144.2.30 (193.144.2.30)
+	Transmission Control Protocol, Src Port: 51313 (51313), Dst Port: http (80), Seq: 1, Ack: 1, Len: 562
	Source port: 51313 (51313)
	Destination port: http (80)
	[Stream index: 37]
	Sequence number: 1 (relative sequence number)
	[Next sequence number: 563 (relative sequence number)]
	Acknowledgement number: 1 (relative ack number)
	Header length: 20 bytes
+	Flags: 0x18 (PSH, ACK)
	window size: 17520
+	Checksum: 0xf23f [validation disabled]
+	[SEQ/ACK analysis]
+	Hypertext Transfer Protocol
+	GET / HTTP/1.1\r\n
	Host: www.unirioja.es\r\n
	Connection: keep-alive\r\n
	Accept: application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5\r\n
	User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US) AppleWebKit/534.7 (KHTML, like Gecko) Chrome/7.0.517.41 Safari/534.7\r\n
	Accept-Encoding: gzip,deflate,sdch\r\n
	Accept-Language: es-ES,es;q=0.8\r\n
	Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3\r\n
	Cookie: __utma=130107851.303738447.1287310227.1287310227.1287310227.1; __utmz=130107851.1287310227.1.1.utmccn=(direct) utmcsr=(direct) utmcmd=(none)\r\n\r\n

Aspecto de un paquete TCP

```
507 27.470275 192.168.0.10 193.144.2.30 TCP 51317 > http [ACK] Seq=531 Ack=8739 Win=17520 Len=0
Frame 507: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
Ethernet II, Src: IntelCor_16:0d:fc (00:24:d7:16:0d:fc), Dst: Thomson_d4:87:4d (00:11:e3:d4:87:4d)
  Destination: Thomson_d4:87:4d (00:11:e3:d4:87:4d)
  Source: IntelCor_16:0d:fc (00:24:d7:16:0d:fc)
  Type: IP (0x0800)
Internet Protocol, Src: 192.168.0.10 (192.168.0.10), Dst: 193.144.2.30 (193.144.2.30)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  Total Length: 40
  Identification: 0x1e47 (7751)
  Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 128
  Protocol: TCP (6)
  Header checksum: 0x5828 [correct]
  Source: 192.168.0.10 (192.168.0.10)
  Destination: 193.144.2.30 (193.144.2.30)
Transmission Control Protocol, Src Port: 51317 (51317), Dst Port: http (80), Seq: 531, Ack: 8739, Len: 0
  Source port: 51317 (51317)
  Destination port: http (80)
  [Stream index: 50]
  Sequence number: 531 (relative sequence number)
  Acknowledgement number: 8739 (relative ack number)
  Header length: 20 bytes
  Flags: 0x10 (ACK)
  window size: 17520
  Checksum: 0xdd57 [validation disabled]
  [SEQ/ACK analysis]
```

Aspecto de un paquete ARP

232 7.905313 Thomson_d4:87:4d IntelCor_16:0d:fc ARP 192.168.0.1 is at 00:11:e3:d4:87:4d

- + Frame 232: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
- Ethernet II, Src: Thomson_d4:87:4d (00:11:e3:d4:87:4d), Dst: IntelCor_16:0d:fc (00:24:d7:16:0d:fc)
 - + Destination: IntelCor_16:0d:fc (00:24:d7:16:0d:fc)
 - + Source: Thomson_d4:87:4d (00:11:e3:d4:87:4d)
Type: ARP (0x0806)
- Address Resolution Protocol (reply)
 - Hardware type: Ethernet (0x0001)
 - Protocol type: IP (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - Opcode: reply (0x0002)
 - [Is gratuitous: False]
 - Sender MAC address: Thomson_d4:87:4d (00:11:e3:d4:87:4d)
 - Sender IP address: 192.168.0.1 (192.168.0.1)
 - Target MAC address: IntelCor_16:0d:fc (00:24:d7:16:0d:fc)
 - Target IP address: 192.168.0.10 (192.168.0.10)

2.7 Capa de aplicación

- ❑ La capa de aplicación está formada por una serie de protocolos llamados Protocolos de Aplicación
 - ❑ Reciben este nombre porque la mayor parte de ellos se usan en aplicaciones que interactúan con servidores
 - ❑ Se preocupan en particular de cómo intercambiar la información entre cliente y servidor (y no tanto de la información intercambiada, cosa que hacen los protocolos de transporte como TCP,UDP)
 - ❑ Algunos ejemplos: http, ftp, dns, smtp...
-

Protocolos de aplicación: http

Http (Hypertext Transfer Protocol), protocolo de transferencia de hipertexto

Cabeceras habituales:

```

Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
  [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
    [Message: GET / HTTP/1.1\r\n]
    [Severity level: Chat]
    [Group: Sequence]
  Request Method: GET
  Request URI: /
  Request Version: HTTP/1.1
Host: www.rediris.es\r\n
Connection: keep-alive\r\n
Accept: application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5\r\n
User-Agent: Mozilla/5.0 (windows; U; windows NT 6.1; en-US) AppleWebKit/534.7 (KHTML, like Gecko) Chrome/7.0.517.41 safari/534.7\r\n
Accept-Encoding: gzip,deflate,sdch\r\n
Accept-Language: es-ES,es;q=0.8\r\n
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3\r\n
\r\n

```

Protocolos de aplicación: http

Las cabeceras contienen el mensaje al servidor (GET), el nombre del recurso (www.rediris.es) e información sobre nuestro cliente (o navegador); en particular, observa que no contienen ninguna información sobre a dónde y cómo nos conectaremos (puertos, IPs...)

Protocolos de aplicación: http

Mensaje de respuesta (también http):

```
[-] Hypertext Transfer Protocol
  [-] HTTP/1.1 200 OK\r\n
    [-] [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      [Message: HTTP/1.1 200 OK\r\n]
      [Severity level: Chat]
      [Group: Sequence]
      Request Version: HTTP/1.1
      Response Code: 200
      Date: Sun, 24 Oct 2010 22:54:59 GMT\r\n
      Server: Apache/2.2.3 (Red Hat)\r\n
      Content-Location: index.php.es\r\n
      Vary: negotiate,accept-language\r\n
      TCN: choice\r\n
      X-Powered-By: PHP/5.2.10\r\n
      Set-Cookie: language=es; path=/; domain=rediris.es\r\n
      Keep-Alive: timeout=15, max=100\r\n
      Connection: Keep-Alive\r\n
      Transfer-Encoding: chunked\r\n
      Content-Type: text/html; charset=UTF-8\r\n
      Content-Language: es\r\n
      \r\n
    [+ ] HTTP chunked response
  [+ ] Line-based text data: text/html
```

Información referente al resultado de la petición (200 OK), al tipo MIME de la respuesta, a la fecha de la operación...

Otros mensajes típicos de respuesta:

404: Recurso no encontrado

403: Acceso prohibido

304: Recurso no modificado

Protocolos de aplicación: dns

- ❑ DNS (Domain Name Server) es el protocolo que se usa para convertir nombres de dominios (www.rediris.es) en direcciones numéricas (130.206.1.46)
 - ❑ Las peticiones van a dirigidas a un servidor DNS, que contiene información de nombres de dominios y sus direcciones IP
 - ❑ Observa la importancia de este protocolo, ya que siempre que nos conectamos a cualquier dominio de Internet (por http, ftp o cualquier otro protocolo) será necesario previamente resolver su IP en el servidor DNS, por medio de este protocolo
-

Protocolos de aplicación: dns

- ❑ Los mensajes del protocolo DNS suelen usar UDP como protocolo de transporte
- ❑ Ejemplo de mensaje de solicitud DNS:

```
Domain Name System (query)
  [Response In: 3798]
  Transaction ID: 0x9fb0
  Flags: 0x0100 (Standard query)
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  Queries
    www.rediris.es: type A, class IN
      Name: www.rediris.es
      Type: A (Host address)
      Class: IN (0x0001)
```

Contiene información sobre el dominio a ser buscado, el tipo de mensaje que es (Standard Query)...

Protocolos de aplicación: dns

Mensaje de respuesta dns

```
[-] Domain Name System (response)
    [Request In: 3792]
    [Time: 0.050139000 seconds]
    Transaction ID: 0x9fb0
    [-] Flags: 0x8180 (standard query response, No error)
        Questions: 1
        Answer RRs: 1
        Authority RRs: 0
        Additional RRs: 0
    [-] Queries
        [-] www.rediris.es: type A, class IN
            Name: www.rediris.es
            Type: A (Host address)
            Class: IN (0x0001)
    [-] Answers
        [-] www.rediris.es: type A, class IN, addr 130.206.1.46
            Name: www.rediris.es
            Type: A (Host address)
            Class: IN (0x0001)
            Time to live: 13 minutes, 5 seconds
            Data length: 4
            Addr: 130.206.1.46 (130.206.1.46)
```

Contiene información sobre la búsqueda a realizar (www.rediris.es) y sobre la respuesta (130.206.1.46); esto permite que la comunicación no se bloquee. También contiene información sobre el tiempo de vida de la respuesta, el tipo de mensaje (Standard query response, No error)

Protocolos de aplicación: ftp

FTP (File transfer protocol) permite enviar y recibir ficheros de nuestro equipo a un servidor

Permite envío de ficheros en modo ASCII o en binario

Ejemplo de cabeceras ftp:

```
File Transfer Protocol (FTP)
150 Here comes the directory listing.\r\n
    Response code: File status okay; about to open data connection (150)
    Response arg: Here comes the directory listing.
```

La sencillez de sus cabeceras lo hace especialmente práctico por la poca información extra que genera

Protocolos de aplicación: smtp

- ❑ SMTP (Simple Mail Transfer Protocol): protocolo para el envío de correos electrónicos
 - ❑ El ordenador interactúa con el servidor proporcionándole información sobre los destinatarios, asunto, cuerpo del mensaje (y el servidor le devuelve códigos numéricos)
 - ❑ El transporte del contenido del mensaje, adjuntos... se hace por algún protocolo de la capa de transporte (TCP, encriptado con TLSv1...)
-

Protocolos de aplicación: pop3

- ❑ POP3 (Post Office Protocol 3) es un protocolo para la transferencia de correos.
 - ❑ Permite coger los mensajes de un servidor de correo y transferirlos a nuestro equipo
 - ❑ Los mensajes dejan de estar disponibles en el servidor (salvo que configuremos la cuenta adecuadamente)
 - ❑ Apto para grandes (varios GB) cuentas de correo
-

Protocolos de aplicación: imap

- ❑ IMAP (Internet Mail Access Protocol) es un protocolo de acceso a mensajes de correo
 - ❑ Permite sincronizar en nuestro equipo una carpeta con la carpeta de correo en el servidor
 - ❑ Los mensajes siempre residen en el servidor
 - ❑ El correo WEB suele ser una interfaz a una cuenta de correo por protocolo IMAP
-

2.8 Capa de transporte: TCP y UDP

Todos los protocolos de aplicación que hemos visto dependen de un protocolo de transporte

Los protocolos de transporte se encargan de:

- ❑ Empaquetar la información en paquetes razonables para que se puedan enviar (TCP y UDP) y prestar atención a su recepción en el destinatario (TCP)
 - ❑ Garantizar la seguridad en el envío e informar del estado del mismo (TCP)
 - ❑ Asignar puertos a la comunicación (toda comunicación entre máquinas tiene lugar entre dos puertos que se deben mantener abiertos mientras dure la misma) (TCP y UDP)
-

Protocolos de Transporte: TCP

- ❑ TCP (Transmission Control Protocol) es un protocolo orientado a conexión. Verifica la ruta que van a seguir los paquetes y respeta la ordenación entre los mismos
 - ❑ Es el protocolo usado por la mayor parte de los protocolos de aplicación (http, smtp, imap, pop...) cuando éstos deben fragmentar los datos de un envío
 - ❑ Sus cabeceras incluyen información sobre los puertos de origen y envío de la comunicación, la secuencia de envíos y recepciones de los mismos, si la comunicación ya ha terminado, información de validación...
-

Protocolos de transporte: TCP

```
Transmission Control Protocol, Src Port: 64081 (64081), Dst Port: https (443), Seq: 1, Ack: 84, Len: 0
  Source port: 64081 (64081)
  Destination port: https (443)
  [Stream index: 6]
  Sequence number: 1      (relative sequence number)
  Acknowledgement number: 84    (relative ack number)
  Header length: 20 bytes
  Flags: 0x10 (ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion Window Reduced (CWR): Not set
    .... .0.. = ECN-Echo: Not set
    .... ..0. = Urgent: Not set
    .... ...1 .... = Acknowledgement: Set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    .... .... ...0 = Fin: Not set
  Window size: 64214
  Checksum: 0x971d [validation disabled]
    [Good Checksum: False]
    [Bad Checksum: False]
  [SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 16]
    [The RTT to ACK the segment was: 0.000096000 seconds]
```

Protocolos de transporte: TCP

- ❑ Algunos protocolos de aplicación que hacen uso de TCP para enviar la información (y sus puertos por defecto) son http (80), ftp (20 para los datos y 21 para los mensajes), smtp (25), pop3 (110), imap (143)
 - ❑ El mayor inconveniente que tiene es la gran cantidad de información que añade en las cabeceras de los mensajes y el número excesivo de mensajes que genera en toda comunicación
-

Ejemplo de comunicación TCP

No.	Time	Source	Destination	Protocol	Info
15084	5040.532874	192.168.0.10	128.30.52.168	TCP	61116 > http [SYN] Seq=0 win=8192 Len=0 MSS=1460 SACK_PERM=1
15087	5040.672216	128.30.52.168	192.168.0.10	TCP	http > 61116 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=1460
15088	5040.673178	192.168.0.10	128.30.52.168	TCP	61116 > http [ACK] Seq=1 Ack=1 win=17520 Len=0
15089	5040.677455	192.168.0.10	128.30.52.168	HTTP	GET /Consortium/Member/Testimonial/Logo/874 HTTP/1.1
15092	5040.802768	128.30.52.168	192.168.0.10	TCP	http > 61116 [ACK] Seq=1 Ack=388 win=6432 Len=0
15093	5040.811774	128.30.52.168	192.168.0.10	TCP	[TCP segment of a reassembled PDU]
15094	5040.812017	128.30.52.168	192.168.0.10	HTTP	HTTP/1.1 200 OK (PNG)
15095	5040.812070	192.168.0.10	128.30.52.168	TCP	61116 > http [ACK] Seq=388 Ack=2605 win=17520 Len=0
15096	5040.812173	128.30.52.168	192.168.0.10	TCP	http > 61116 [FIN, ACK] Seq=2605 Ack=388 win=6432 Len=0
15097	5040.812208	192.168.0.10	128.30.52.168	TCP	61116 > http [ACK] Seq=388 Ack=2606 win=17520 Len=0
15098	5040.812608	192.168.0.10	128.30.52.168	TCP	61116 > http [FIN, ACK] Seq=388 Ack=2606 win=17520 Len=0
15099	5040.938401	128.30.52.168	192.168.0.10	TCP	http > 61116 [ACK] Seq=2606 Ack=389 win=6432 Len=0

1. La comunicación empieza con “nuestra” solicitud de SYN (sincronización) al servidor (por el puerto 61116). El servidor responde que está preparado ([SYN, ACK]) y nosotros respondemos con [ACK] para reconocer que seguimos esperando la comunicación.
2. Nuestra máquina ya está preparada para enviar la solicitud HTTP.
3. La comunicación de datos se hace por TCP, y cuando el servidor nos manda la señal “200 OK”, se cierra la comunicación ([FIN, ACK], [ACK])

Protocolos de transporte: UDP

- ❑ UDP (User Datagram Protocol) es un protocolo no orientado a conexión. No solicita reconocimiento de conexión ([SYN]) ni de recepción de paquetes ([ACK]) ni de fin de comunicación ([FIN])
 - ❑ Sus cabeceras son mucho más simples que las de TCP (aunque, por supuesto, incluyen los puertos)
-

Protocolos de transporte: UDP

Ejemplo de cabeceras UDP:

```
[-] User Datagram Protocol, Src Port: 54616 (54616), Dst Port: domain (53)
    Source port: 54616 (54616)
    Destination port: domain (53)
    Length: 41
    [-] Checksum: 0xad0c [validation disabled]
        [Good Checksum: False]
        [Bad Checksum: False]
```

Ejemplo de comunicación a través de UDP (DNS)

15544	5352.891264	192.168.0.10	62.42.230.24	DNS	standard query A mail.google.com
15545	5352.939010	62.42.230.24	192.168.0.10	DNS	standard query response CNAME googlegmail.l.google.com A 72.14.235.83

Su mayor ventaja es que genera poca información adicional en los mensajes (sólo los puertos), y pocos mensajes en cada comunicación. Eso lo hace ideal para mensajes DNS (puerto 53), o algunos otros protocolos que usan comunicaciones breves

2.9 Capa de red o de Internet

- ❑ Base de la interconexión entre distintas redes que pueden tener estructuras e implementaciones distintas
 - ❑ Tanto TCP como UDP hacen uso de protocolos de la capa de red (en particular, de IPv4)
 - ❑ Principales funciones:
 - Gestión de direcciones de máquinas (IPs)
 - Enrutamiento de la información
 - Control de errores, tiempo de vida de los paquetes, control del flujo
 - Traducción de direcciones IP a direcciones físicas (por ejemplo MAC) con protocolos como ARP
 - Gestión del tamaño de los paquetes que se entregan a la capa física
-

2.9 Protocolo IP

- ❑ IP es un protocolo no orientado a conexión (TCP sobre IP sí lo es, pero no así UDP sobre IP). No presta atención al orden de envío o recepción de paquetes (aunque sí comprueba su integridad).
-

2.9 Protocolo IP

Ejemplo de cabeceras IP:

```
Internet Protocol, Src: 62.42.63.52 (62.42.63.52), Dst: 192.168.0.10 (192.168.0.10)
  Version: 4
  Header length: 20 bytes
  + Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 123
    Identification: 0x0000 (0)
  + Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 55
    Protocol: UDP (17)
  + Header checksum: 0x0562 [correct]
    Source: 62.42.63.52 (62.42.63.52)
    Destination: 192.168.0.10 (192.168.0.10)
```

Contiene información sobre:

1. Las IPs de origen y destino
2. El tiempo de vida del datagrama (impidiendo que vague por la red indefinidamente)
3. Comprobación de la integridad de los datos (header checksum)
4. El protocolo de transporte del paquete que contiene (TCP o UDP)

2.9 Protocolo IP: direcciones IP

- ❑ En una red IP cada equipo se identifica por una dirección numérica (única) IP
 - ❑ Estas direcciones se basan en el protocolo IPv4 (Ipv6...)
 - ❑ Cada dirección ocupa 4 bytes, 32 bits (desde 0.0.0.0 a 255.255.255.255)
 - ❑ Cada ordenador suele formar parte de su propia red (clases de direcciones IP) y por tanto tiene una IP privada (que podría coincidir con la de otros equipos en otras redes privadas)
-

Protocolo IP: clases de direcciones IP

	0	1	8	16	24	31
Clase A	0	N° de red		número de host		
Clase B	1	0	Número de red		número de host	
Clase C	1	1	0	Número de red		número de host
Clase D	1	1	1	0	Direcciones multicast	
Clase E	1	1	1	1	Reservadas (no se han usado)	

Protocolo IP: clase A de direcciones

- ❑ El primer bit es igual a 0
 - ❑ Quedan 7 bits para identificar la red ($2^7 = 128$ posibles redes de clase A)
 - ❑ Quedan 24 bits para identificar cada máquina (o host) de la red ($2^{24} = 16777216$ posibles máquinas por red)
-

Protocolo IP: clase A de direcciones

- El rango de redes de tipo A es:
 - Desde 1.xxx.xxx.xxx hasta 127.xxx.xxx.xxx
 - Por tanto, contamos 126 posibles redes de tipo A
 - El rango de ordenadores en cada red es:
 - Desde xxx.0.0.0 hasta xxx.255.255.255
 - Por tanto, cada red de tipo A dispone de $256^3 = 16777216$ direcciones posibles
-

Protocolo IP: clases de direcciones IP

	0	1	8	16	24	31
Clase A	0	N° de red		número de host		
Clase B	1	0	Número de red		número de host	
Clase C	1	1	0	Número de red		número de host
Clase D	1	1	1	0	Direcciones multicast	
Clase E	1	1	1	1	Reservadas (no se han usado)	

Protocolo IP: clase B de direcciones

- ❑ Los dos primeros bits son 10
 - ❑ Quedan 14 bits para identificar la red ($2^{14} = 16384$ posibles redes de clase B)
 - ❑ Quedan 16 bits para identificar hosts o máquinas ($2^{16} = 65536$ posibles máquinas)
-

Protocolo IP: clase B de direcciones

- El rango de redes de tipo B es:
 - Desde 128.0.xxx.xxx hasta 192.255.xxx.xxx
 - Por tanto, contamos $64 * 256 = 16384$ posibles redes de tipo B
 - El rango de ordenadores en cada red es:
 - Desde xxx.xxx.0.0 hasta xxx.xxx.255.255
 - Por tanto, cada red de tipo B dispone de $256^2 = 65536$ direcciones posibles
-

Protocolo IP: clases de direcciones IP

	0	1	8	16	24	31
Clase A	0	N° de red		número de host		
Clase B	1	0	Número de red		número de host	
Clase C	1	1	0	Número de red		número de host
Clase D	1	1	1	0	Direcciones multicast	
Clase E	1	1	1	1	Reservadas (no se han usado)	

Protocolo IP: clase C de direcciones

- ❑ Los tres primeros bits son 110
 - ❑ Quedan 21 bits para identificar la red (2^{21} posibles redes de clase B)
 - ❑ Quedan 8 bits para identificar hosts o máquinas ($2^8=256$ posibles máquinas, al menos con salida al exterior)
-

Protocolo IP: clase C de direcciones

- El rango de redes de tipo C es:
 - Desde 192.0.0.xxx hasta 223.255.255.xxx
 - Por tanto, contamos $32 * 256^2 = 2.097.152$ posibles redes de tipo C
 - El rango de ordenadores en cada red es:
 - Desde xxx.xxx.xxx.0 hasta xxx.xxx.xxx.255
 - Por tanto, cada red de tipo C dispone de 256 direcciones posibles
-

Clases de direcciones:

- ❑ Las direcciones de clase A están entre 1 y 127 (su primer octeto)
- ❑ Las direcciones de clase B están entre 128 y 191 (su primer octeto)
- ❑ Las direcciones de clase C están entre 192 y 224 (su primer octeto)

¿Qué tipo de direcciones tiene la UR?

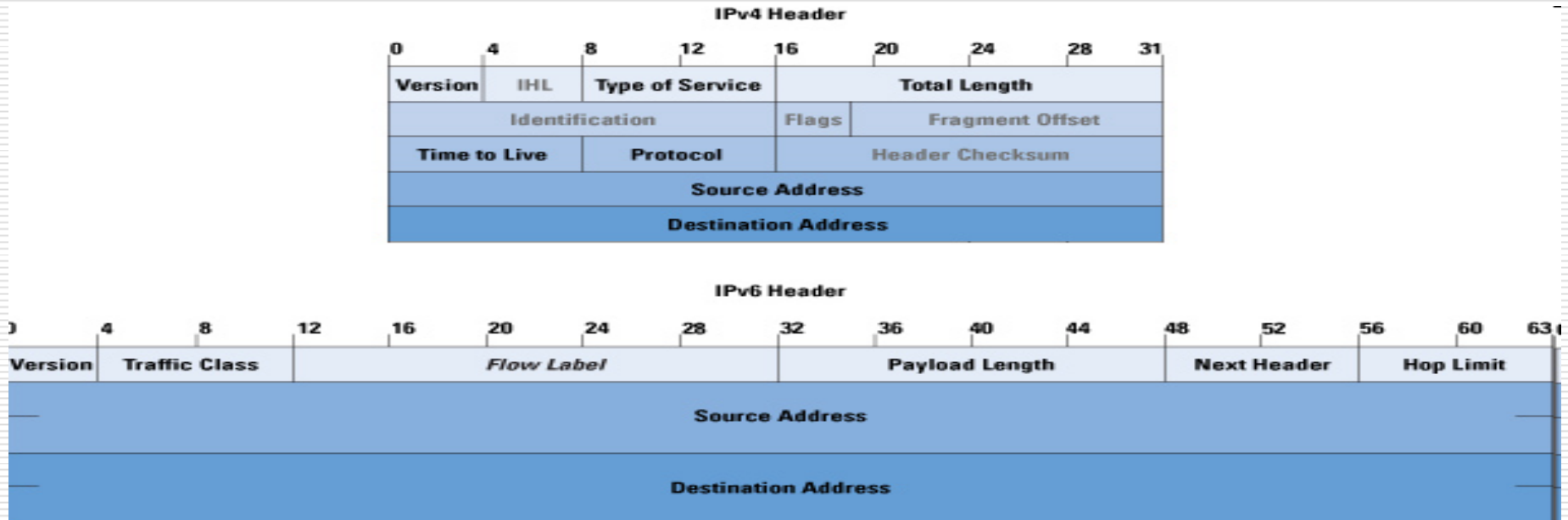
IPv4 vs IPv6

- La escasez de direcciones IPv4 empieza a ser un problema serio
 - Algunas soluciones:
 - Asignación de IPs no fijas (protocolo DHCP, IPs dinámicas)
 - Fragmentación de redes en subredes
 - Uso en redes privadas de IPs reservadas
-

IPv4 vs IPv6

- Solución a medio plazo: uso del protocolo IPv6
 - Algunas características:
 - 2^{128} direcciones posibles (por 2^{32} de IPv4)
 - Aceptado por la mayor parte de SO
 - Se han simplificado (o eliminado) algunas de las cabeceras menos usadas de IPv4
-

IPv4 vs IPv6 (cabeceras):



Las direcciones ocupan 4 veces más en IPv6 que en IPv4 (128 bits contra 32 bits)

Las comprobaciones de integridad del mensaje (Header Checksum) se omiten y se deja a los protocolos de transporte (TCP, UDP)

Protocolo IP: Enrutamiento

- ❑ Cada vez que un paquete abandona nuestra red, nuestra puerta de enlace lo adapta y lo redirige hacia el servidor de destino
 - ❑ El mensaje puede llegar a destino en un paso (bastante raro), o ir pasando por routers que lo van redirigiendo hacia su destino (prueba a ejecutar “tracert”)
 - ❑ Esto es posible porque cada paquete lleva su dirección de origen (IP, MAC) y destino (IP, MAC)
 - ❑ Cada paquete lleva un parámetro adicional (“Time to live”) que indica por cuántos routers puede pasar el paquete antes de eliminarse (p. ej, en “tracert” el primer paquete es 1, el segundo 2, hasta el máximo que es 30). Así evitamos paquetes “vagando” por Internet
-

Protocolo IP: Control de errores

- El protocolo ICMP (Internet Control Message Protocol) se utiliza para comprobar si los dispositivos que deben enviar y recibir paquetes están disponibles
Dentro del protocolo ICMP, las dos aplicaciones más conocidas son:
 - ping “nombre_de_dominio o IP”: determina si la máquina está disponible (quizá el servidor no acepte mensajes ping)
 - tracert “nombre_de_dominio o IP”: envía mensajes ping con “Time to live” igual a 1, 2, 3, 4... para ver hasta qué máquinas llegan, y así verificar la ruta que han seguido los paquetes hasta llegar a la máquina de destino
 - Además, en las cabeceras IP existe un campo “Header Checksum” que comprueba la integridad de los paquetes (como un bit de paridad que se comprueba en cada paso del paquete); si el “Header Checksum” se comprueba correcto, se descarta el paquete
-

Protocolo IP: paso a nivel físico

Dependiendo del tipo de red (Ethernet, Wireless, Token Ring) las direcciones IP se convierten a direcciones de dispositivos físicos (MAC...). El protocolo ARP (Address Resolution Protocol) permite convertir IPs a MACs

También puede darse el caso de que los paquetes sean demasiado grandes para una red, y entonces la propia red se deberá encargar de segmentarlo y recomponerlo

2.9 Capa de red: redes privadas

☐ Direcciones IP reservadas:

- Clase A: la red 10 (¿Qué IP tenía www.unirioja.es desde la red local? ¿Y nuestras máquinas de prácticas?) Admiten 256^3 máquinas locales
 - Clase B: las redes 172.16 y 172.31. Admiten 256^2 máquinas locales
 - Clase C: las redes entre 192.168.0 y 192.168.255 (¿Qué IP tiene tu ordenador particular?) Admiten 256 máquinas locales
-

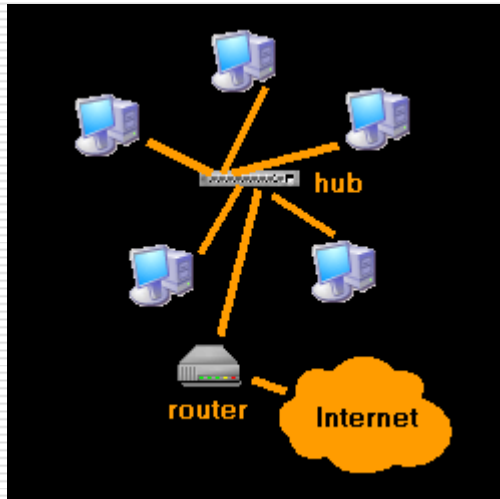
2.9 Capa de red: redes privadas

- ❑ Redes privadas: generalmente, las anteriores IPs reservadas se utilizan en redes privadas o particulares (si tiene “muchos ordenadores”, como la de la Universidad, usará el “10.....”; si tiene “pocos”, como las redes particulares, usará “192.168.....”)
 - ❑ Estas redes privadas se configuran de modo que un gran número de máquinas se conectan a la red exterior (Internet) a través de un solo dispositivo (puerta de enlace o “gateway”)
-

2.9 Capa de red: redes privadas

La red queda de la siguiente forma:

- Todos los dispositivos de nuestra red disponen sólo de dirección privada
- El dispositivo de enlace tendrá una dirección privada para comunicarse con las máquinas de la red privada, y una dirección pública para conectarse con las del exterior



2.9 Capa de red: redes privadas

- El dispositivo enrutador tiene ahora una doble tarea:
 - Enviar peticiones al exterior y recibir las del exterior para encaminarlas al interior
 - Mediante el protocolo NAT (Network Address Translation) convierte todas las peticiones del interior en peticiones que lleven una IP pública (la suya) pero “recordando” que cuando se responda a las peticiones deberá enviarlas a cada máquina de la red privada
-

2.9 Capa de red: Subredes

Si una red tiene demasiadas direcciones, podemos pensar en dividirla en subredes para:

- Reducir el tamaño de los dominios de broadcast (mensajes que se envían a todos los ordenadores de la red)
 - Permitir que la red sea más manejable (poder explorar una de las subredes...)
-

2.9 Capa de red: Subredes

- Para definir subredes debemos definir una máscara de subred;

Dada una red de tipo C (256 direcciones) con dirección 221.33.45....:

- si definimos su máscara de subred como “255.255.255.0” dividiremos la misma en subredes de 256 máquinas, por tanto no queda dividida
 - si definimos su máscara de subred como “255.255.255.224” dividiremos la misma en subredes de 32 máquinas
 - si definimos su máscara de subred como “255.255.255.240” dividiremos la misma en subredes de 16 máquinas
 - ...(las subdivisiones deben ser siempre de potencias de 2)
-

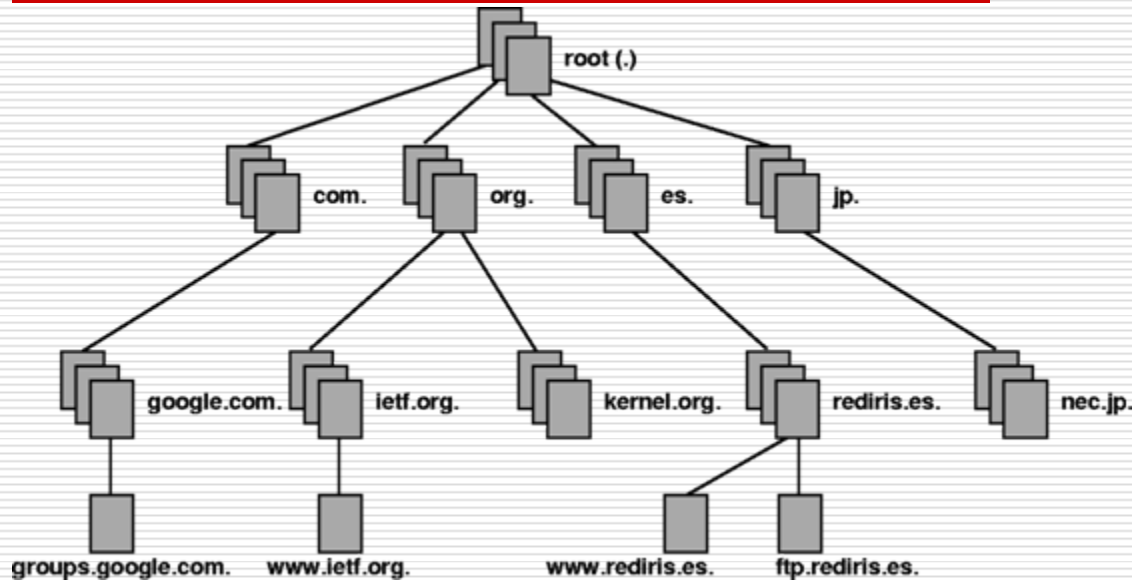
2.10 Sistemas de nombres de dominios

- ❑ Las direcciones de dominios se corresponden con direcciones IP de servidores que cada máquina debe conocer para poder conectarse a dichos servidores
 - ❑ Al principio de Internet, cada ordenador disponía de un fichero de texto en que los usuarios apuntaban los nombres de dominios (www.unirioja.es) y sus IPs correspondientes (193.146.250.30)
 - ❑ Ante la imposibilidad de mantener ese sistema, se creó el servicio o protocolo DNS (Domain Name Server)
-

2.10 Sistemas de nombres de dominios

- ❑ DNS consiste en una base de datos jerárquica distribuida por Internet y consultada por las aplicaciones de usuario
 - ❑ La jerarquía se basa en dominios, cada uno de los cuales es responsable de la asociación entre IPs y urls de su “dominio”
-

2.10 Sistemas de nombres de dominios



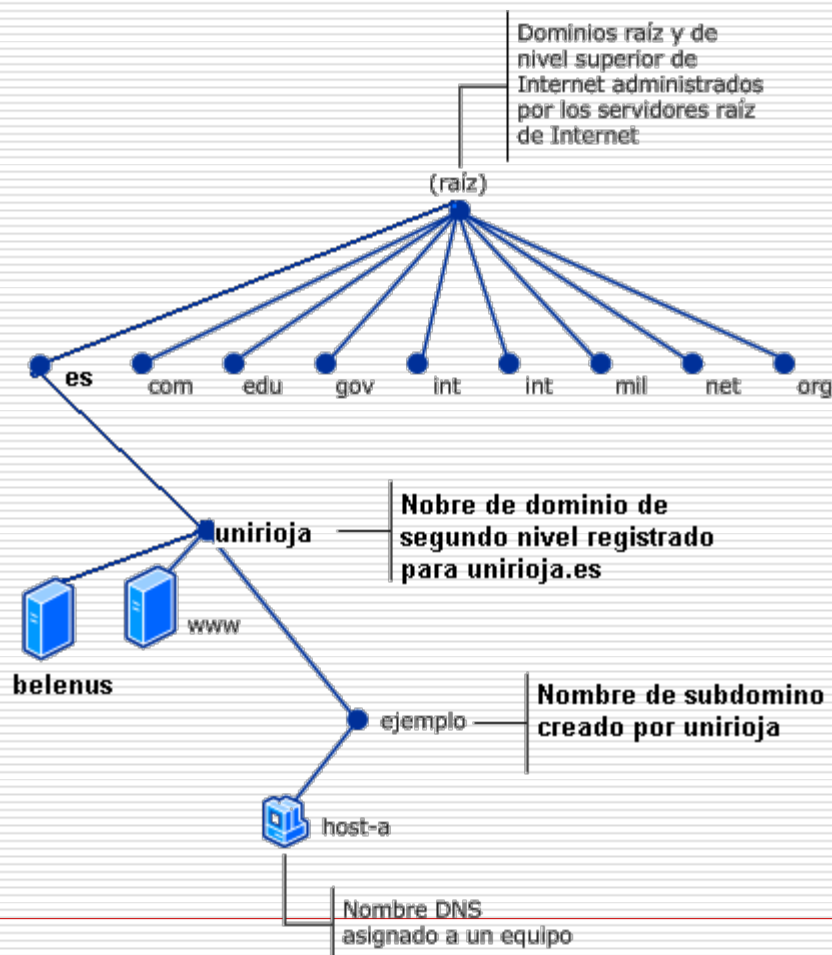
En el nivel superior se encuentran los dominios de países (.es) o agrupaciones supranacionales (.eu), y también algunos especiales como “.org, .com, .edu, .net...”

En el segundo nivel, el NIC de cada zona se hace responsable de los dominios que ha asignado (wikipedia.org, google.com, rae.es, rediris.es)

2.10 Sistemas de nombres de dominios

- ❑ Cuando reservamos un nombre de dominio a través de un registrador (prueba en www.nic.es si algún dominio está libre y con quién lo puedes contratar), el mismo no facilita unos servidores DNS autoritativos, en los cuales está explícita la información de nuestra IP y nuestro nombre de dominio
 - ❑ Otros servidores DNS también podrán buscar esa información, pero no serán autoritativos para nuestro dominio
 - ❑ La página www.nic.cl/doctorDNS/ te da información sobre servidores autoritativos de distintas páginas web (puedes probar con “nslookup” si lo anterior es cierto)
-

2.10 Sistemas de nombres de dominios



2.11 URLs

- ❑ Los URL's (Uniform Resource Locator) surgen en 1991 para facilitar el acceso a enlaces en la web
 - ❑ A partir de 1994 se generaliza el concepto URL a URI (Uniform Resource Identifier), incluyendo también los URN (Uniform Resource Names)
-

Estructura de una URL

protocolo://maquina.dominio:puerto/camino/
fichero?parámetros

- ❑ El protocolo es alguno de los que ofrece Internet: http, https, ftp, ftps, news, telnet, mailto
 - ❑ La maquina.dominio indica el servidor que nos ofrece el recurso (para nuestra universidad el servidor web es www.unirioja.es, el servidor de correo Web es correo.unirioja.es)
 - ❑ El puerto es opcional y no es necesario indicarlo si usamos el habitual del servicio (http=80, ftp=20, ftp-data=21...)
-

Estructura de una URL

protocolo://maquina.dominio:puerto/camino/fichero?parámetros

- ❑ El camino es la ruta de directorios a seguir para encontrar el fichero deseado. Para separar subdirectorios utilizaremos la barra de UNIX /, se utiliza por convenio por ser el sistema de la mayoría de los servidores
 - ❑ La extensión de los ficheros es importante porque indica al navegador cómo tiene que tratarlos. txt fichero de texto plano, htm o html documento HTML, gif jpg o jpeg formato de imagen... Los servidores WEB pueden estar configurados para que, al poner la ruta, si no indicamos nombre de fichero, un determinado fichero (o varios en orden de preferencia) se asuman por defecto. Nuestro servidor belenus carga por defecto la página index.htm
 - ❑ Los parámetros sirven para pasar a un determinados ficheros (asp, php,...) valores de variables. El formato es variable1=valor1&variable2=valor2...
-

Estructura de una URL

Podemos encontrar algunas variantes como:

- ❑ `protocolo://usuario:contraseña@máquina:puerto/directorio/archivo`

No muy recomendable su uso porque la contraseña aparecerá visible para todo aquel que esté mirando, pero que puedes probar, por ejemplo, para acceder a tu directorio FTP en belenus:

`ftp://cuasi:contraseña@belenus.unirioja.es`

- ❑ Los URL's usados para http ó https que tienen algunas particularidades como el uso de anclas (#) para crear enlaces a fragmentos o puntos concretos de una página web
-

Seguridad en Internet: introducción

Internet, como toda comunicación entre pares, es potencialmente insegura

Las comunicaciones en muchos casos no van encriptadas, y es fácil capturar paquetes que contienen mensajes, usuarios, contraseñas...

La criptografía y los protocolos tratan de ayudar a aumentar la seguridad de comunicaciones

Principios básicos del intercambio seguro de información

Principios básicos que toda comunicación debe cumplir para considerarse segura:

☐ Confidencialidad o Privacidad:

- Objetivo: hacer posible el intercambio de mensajes de manera que sólo puedan ser leídos por las personas a quienes van dirigidos
- Metodología: Se hace uso de la Criptografía (ciencia de cifrar y descifrar información utilizando técnicas matemáticas)
- Ejemplo: Encriptación por clave pública y clave privada

☐ Integridad:

- Objetivo: Garantizar que la información que ha llegado a destino no ha sido modificada o alterada por el camino
 - Metodología: combinación de Criptografía, funciones hash (huellas digitales) y firmas digitales
-

Principios básicos del intercambio seguro de información (y 2)

Principios básicos que toda comunicación debe cumplir para considerarse segura:

- ❑ Autenticidad:
 - Objetivo: garantizar que la persona con la que nos comunicamos es, realmente, quien dice ser
 - Metodología: mediante certificados digitales
 - ❑ No Repudio:
 - Objetivo: Garantizar que, una vez efectuada una determinada comunicación o transacción, ninguno de los intervinientes en la misma pueda negar su participación en la misma
 - Metodología: mediante certificados digitales y firmas digitales
-

Algunos protocolos para la seguridad: SSL (Secure Socket Layer) y TLS (Transport Layer Security)

- ❑ Se ubican entre la capa de aplicación y la capa de transporte.
- ❑ Se basan en criptografía asimétrica (clave pública-clave privada)
- ❑ Comienzan con
 - el envío desde el servidor al cliente de su certificado X.509 (como cuando nos conectamos a <https://mail.google.com>)
 - o el intercambio bidireccional de certificados desde el servidor al cliente y desde el cliente al servidor
- ❑ Posteriormente se negocia entre ambos una clave de sesión para cifrar los datos sensibles que se intercambian o todos los datos del intercambio
- ❑ La información va por la red encriptada

(Recuerda lo que pasaba con los mensajes SMTP en la práctica 10, usando TLSv1)

Algunos protocolos para la seguridad: HTTPS y FTPS

- ❑ HTTPS: utiliza un cifrado basado en TLS. Permite utilizar de forma simultánea HTTP y HTTPS. Utiliza, por defecto, el puerto 443
 - ❑ FTPS: en general corresponde con diversas formas de añadir a FTP una capa de SSL/TLS
 - ❑ Protocolos de correo seguros:
<http://www.unirioja.es/servicios/si/seguridad/correoseguro.shtml>
-

2.12 Configuración de una red IP

Algunos ordenadores al arrancar no disponen de una configuración IP correcta (sólo la capa física funciona).

Existen algunos modos (protocolos) de “recuperar” o “conseguir” la configuración IP

Protocolos para obtener direcciones IP

- ❑ RARP (Reverse Address Resolution Protocol):
Partiendo de la dirección de la capa física, la dirección del hardware, se conecta con un servidor RARP que le proporciona una dirección IP
 - ❑ BOOTP (Bootstrap Protocol):
Es utilizado por los clientes de red para obtener su dirección IP automáticamente desde un servidor adecuado. Normalmente se realiza en el proceso de arranque de los ordenadores o del sistema operativo. Se ha usado mucho en estaciones basadas en UNIX.
Ventajas sobre RARP:
 - Se basa en UDP y puede atravesar la barrera física de la propia red para alcanzar un servidor fuera de ella
 - Integrado en las BIOS de tarjetas de red y BIOS placas base de ordenador
 - Se puede obtener toda la configuración TCP/IP, incluso imágenes de arranque completas gracias a estar integrado en las BIOS
 - ❑ DHCP (Dynamic Host Configuration Protocol):
Protocolo basado en BOOTP que permite obtener diferentes parámetros entre los que están dirección IP, máscara de red, DNS y puerta de enlace para crear la configuración IP de un equipo
-

Comandos para operaciones TCP/IP en la consola de MSDOS

- ❑ > arp
Muestra y modifica entradas en la caché del Protocolo de resolución de direcciones (ARP), que contiene una o varias tablas utilizadas para almacenar direcciones IP y sus direcciones físicas Ethernet o Token Ring resueltas. Existe una tabla independiente para cada adaptador de red Ethernet o Token Ring instalados en el equipo
 - ❑ > ftp
Transfiere archivos en equipos que ejecutan un servicio del servidor del Protocolo de transferencia de ficheros (FTP, File Transfer Protocol)
 - ❑ > ipconfig
Muestra los valores actuales de la configuración de la red TCP/IP y actualiza la configuración de DHCP (Protocolo de configuración dinámica de host) y DNS (Sistema de nombres de dominio)
 - ❑ > nslookup
Muestra información que puede usar para diagnosticar la infraestructura de DNS (Sistema de nombres de dominio). Para utilizar esta herramienta hay que conocer el funcionamiento de DNS
-

-
- ❑ > ping
Comprueba la conectividad de nivel IP en otro equipo TCP/IP al enviar mensajes de solicitud de eco de ICMP
 - ❑ > route
Muestra y modifica las entradas de la tabla de rutas IP local
 - ❑ > tracert
Determina la ruta tomada por un paquete hacia un destino mediante el envío por ping de paquetes al destino con valores de de tiempo de vida (TTL) incrementales (1, 2, 3...)
 - ❑ > netstat
Muestra las conexiones de TCP activas, los puertos en que el equipo escucha, las estadísticas de Ethernet, la tabla de enrutamiento IP, las estadísticas de IPv4 (para los protocolos IP, ICMP, TCP y UDP) y las estadísticas de IPv6 (para los protocolos IPv6, ICMPv6, TCP sobre IPv6 y UDP sobre IPv6). Cuando se utiliza sin parámetros, netstat muestra las conexiones de TCP activas
-