



estudios de postgrado
en computación



Bases de datos avanzadas

Universidad de Los Andes

Postgrado en Computación

Prof. Isabel M. Besembel Carrera

Unidad II. Sesión 8. WEB.

Introducción ...

- WWW (World Wide Web, w3): concebido en 1989 por Tim Berners Lee del Laboratorio de Física (CERN-Suiza) para permitir a los investigadores y visitantes del CERN intercambiar informaciones científicas (artículos, reportes, etc) en física nuclear, luego de sus visitas
- Fundamentado en la extensión del concepto de hiperdocumento distribuido a escala mundial, donde los usuarios pueden navegar o surfear desde su terminal
- *W3C world wide web consortium* <http://www.w3.org>
- Web: permite establecer enlaces entre hiperdocumentos:
 - ✓ almacenados en cualquier servidor conectado a Internet
 - ✓ accesibles por diferentes protocolos (http, ftp, gopher, etc.)



Arquitectura 3 capas

- Acoplamiento de aplicaciones: bases de datos, aplicaciones e Internet para soportar sistemas de información web
 - ✓ Cliente-servidor Web: servidor de datos, servidor de aplicaciones y cliente web.
 - ✓ Generación de sitios web dinámicos: soporte necesario para la gestión de datos evolutivos y presentación homogénea en web.
 - ✓ Comercio electrónico: B2C contacto con el consumidor, B2B basado en el intercambio de datos semi-estructurados entre sistemas de información con transacciones largas y los servicios de pago seguro y seguimiento de la solicitud de compra
- Niveles: Java como lenguaje portable y robusto capaz de evolucionar en ellos
 - ✓ Servidor web o http: gestiona la interfaz de usuario en el navegador
 - ✓ Servidor de aplicaciones: gestiona la lógica del negocio
 - ✓ Servidor de datos: gestiona la BD dentro del SGBD



Herramientas

- Interfaz usuario-navegador:
 - ✓ HTML, formato MIME, HTTP, CGI
- Servidor Web-servidor de aplicación:
 - ✓ navegación dinámica: VBScript o JavaScript, ActiveX o Java Beans, applet Java, HTML dinámico, PHP, ASP, JSP.
- Gestión de los datos en el SGBD: cualquier SGBD con SQL
- XML como lenguaje unificador



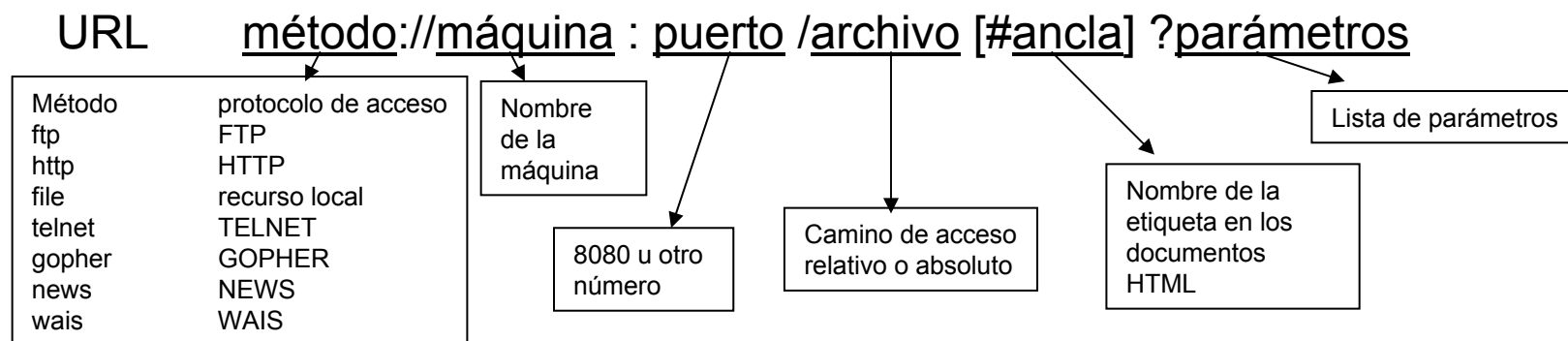
Aplicaciones Web

- Correo electrónico (email): primera aplicación que en 1994 ocupaba una porción importante del tráfico en la red
- Hoy día el tráfico está distribuido entre email, investigación, mercadeo, entretenimiento y comercio electrónico
- 2001:
 - ✓ 109.574.429 dominios,
 - ✓ 407,1 millones de usuarios (6,71% de la población mundial) y
 - ✓ 550 miles de millones de documentos en línea, de los cuales mil millones estaban indexados por los motores de búsqueda.



Tecnologías básicas

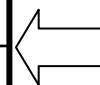
- Navegadores: Netscape, Explorer y servidores Domino, Apache, Mozilla, Information server y Enterprise Server
- Identificador Universal de Recursos (URI): nombre o dirección genérica que permite identificar un recurso, compuesto de una cadena de caracteres. <tipoDeConexión>://<servidor>/<recurso>/...



Tecnologías básicas

- Código MIME (Multipurpose Internet Mail Extensions): conjunto de reglas de codificación de documentos usados para codificar los mensajes electrónicos transmitidos en Internet.
- Lista de extensiones para identificar la naturaleza y finalidad de los datos transmitidos

http	Hypertext transfer protocol
https	Documentos seguros (cifrados)
pop	Post office protocol (recepción)
ftp	File transfer protocol
gopher	Material disponible en Internet
usenet	Grupos de debate (USErs NET)
telnet	Conexión remota
irc	Internet Relay Chat (mensajería en tiempo real)
smtp	Simple Mail Transport Protocol (envío)
mud	Jugar en tiempo real (Multiuser dimension)



agentes

mime

Aplicación postscript	ai
Audio	aiff
Vídeo microsoft	avi
Ejecutable binario	bin
Concha de aplicación	csh
Aplicación DVI-TeX	dvi
Encapsulado postscript	eps
Ejecutable binario	exe
Imagen compuServe	gif
Comprimido GNU	gz
Hypertext Markup L	html
Imagen JPEG	jpg
Vídeo QuickTime	mov
Vídeo MPEG	mpg
Postscript	ps



Tecnologías básicas

- HTML (Hypertext Markup Language): derivado del SGML (Standard Generalized Marked Language) que define un conjunto de marcas fijas para codificar y transmitir documentos y presentaciones integradas en Web.
- Lenguajes de marcas: **<marca> texto <fin de la marca>**
 - ✓ Descripción del contenido, del formato u ambas cosas simultáneamente
- Constante evolución
 - HTML-1 (1989) texto, algunos estilos y enlaces hipertexto
 - HTML-2 (1994) HTML-1 más imágenes y formularios interactivos
 - HTML-3 (1996) HTML-2 más gráficos vectoriales, sonido y applets
 - HTML-4, DHTML (1998) HTML-3 más vídeo, CSS y herramientas para intranet
<http://www.w3.org/pub/www/TR/>



HTML

- Contiene texto ASCII, por lo que puede ser escrito utilizando cualquier editor de texto

<! Comentario >

<html>

<head>

<title> título barra superior navegador </title>

<style> estilo del documento

<base href="URL">

<script> para versiones futuras

<isindex>

<meta>

<link>

</head>

<body> cuerpo del documento

</body>

</html>

Marcas del encabezado

<base href="URL"> documento con la dirección base

<isindex prompt="cadena"> para colocar una consulta simple a una BD

<meta http-equiv="cadena" name="nombre" content="valor"> incluye pares (nombre, valor) de descripción del documento

Ejm: <meta name="Autor" content="Yo">

<link href="URL" rel=relación rev=relación title="título">

relación: **top** documento raíz

contents documento tabla de contenido



UNIVERSIDAD
DE LOS ANDES

HTML

index documento índice del actual

glossary doc. glosario de términos

copyright doc. copyright

next próximo documento

previous documento previo

help documento de ayuda

search página de búsqueda

Ejm: `<link rel=glossary href="glo.html">`

Cuerpo del documento

```
<body bgcolor=color text=color link=color  
vlink=color alink=color>
```

Color = {white, black, blue, red, yellow, silver,
grey, marron, purple, fuschia, green, lime,
olive, navy, teal, aqua}

Colores: "#RRGGBB" en hexadecimal

Ejm: silver="#C0C0C0" olive="#808000"

bgcolor color de fondo

text color del texto

link color de los enlaces

vlink color de los enlaces visitados

alink color de las anclas

Títulos de secciones

```
<h1 align="left|right|center"> título </h1>
```

```
<h2> ... </h2> ... <h6> ... </h6>
```

h1 más grande y h6 el más pequeño, por omisión se asume left (subrayado)



`<address>` dirección del autor `</address>`

Párrafos

`<p align = {"left"|"right"|"center"|"justify"} >`
texto del párrafo
`</p>`

Viñetas pueden anidarse

`<ul type={"disc"|"circle"|"square"} compact>`
`<li type={"disc"|"circle"|"square"}>` elemento ``
.....
``

Numerados pueden anidarse

`<ol type = {"1"|"a"|"A"|"i"|"I"} start="#" compact>`
`<li type={"1"|"a"|"A"|"i"|"I"} value="#">` elemento ``
.....
``

Definiciones

`<dl compact >`
`<dt>` término `<dd>` definición
.....
`</dl>`

Ejemplo:

`<dl>`
`<dt>` Uno `<dd>` Primera definición
`<dt>` Dos `>dd>` Segunda definición
`</dl>`

Se desplegará en la página como

Uno
Primera definición
Dos
Segunda definición



UNIVERSIDAD
DE LOS ANDES

HTML

Directorio y menú igual a

```
<dir | menu compact> texto </dir>
```

Texto preformateado

```
<pre width="#">
```

texto que se desplegará tal cual aparece

```
</pre>
```

Ejemplo:

```
<pre>
```

```
for i = 1, 10
```

```
    A(i) = B(i)
```

```
end for
```

```
</pre>
```

Divisiones o tabulador

```
<div align={"left"|"right"|"center"}>
```

```
... </div>
```

Centrar

```
<center> elemento centrado </center>
```

Anotaciones hace las veces de tabulador

```
<blackquote> texto anotación </blackquote>
```

Líneas horizontales

```
<hr align={"left"|"right"|"center"} noshade  
size=#píxeles width="#">
```

Formas o formularios

```
<form action="URL" method=get|post enctype >  
.... </form>
```

Pueden encerrar botones, cajas de chequeo, menús, etc.

Estilos

Teletipo <tt> ... </tt>

Itálicas <i> ... </i>



HTML

Negritas	<code> ... </code>
Subrayado	<code><u> ... </u></code>
	<code><strike> ... </strike></code>
Enfatizado itálicas	<code> ... </code>
Enfatizado negritas	<code> ... </code>
Letra grande	<code><big> ... </big></code>
Letra pequeña	<code><small> ... </small></code>
Subíndice	<code><sub> ... </sub></code>
Superíndice	<code><sup> ... </sup></code>
Definición	<code><dfn> ... </dfn></code>
Código programas	<code><code> ... </code></code>
Salida de programas	<code><samp> ... </samp></code>
Variables o comandos	<code><var> ... </var></code>

Referencias o citas `<cite> ... </cite>`

Tablas

```
<table align={“left”|”right”|”center”}  
width={“#”|”#%”} border=[0, 1, ..., 4]  
cellspacing=“#píxeles” cellpadding=  
“#píxeles”> filas, encabezado y contenido de  
las columnas </table>
```

Filas

```
<tr align={“left”|”right”|”center”} valign=  
{“top”|”middle”|”bottom”}> ... </tr>
```

Encabezado de columnas

```
<th nowrap rowspan=“#” colspan=“#” align  
={“left”|”right”|”center”} valign = {“top”|”middle  
”|”bottom”} width={“#”|”#%”}  
height=“#píxeles”> ... </th>
```

Contenido de columna

```
<td nowrap rowspan="#" colspan="#" align
    ={"left"|"right"|"center"} valign
    ={"top"|"middle"|"bottom"} width={"#"|"#%"}
    height="#píxeles"> ... </td>
```

Áreas de texto

```
<textarea name=nombre rows=# cols=# >
... </textarea>
```

Anclas

```
<a name=etiqueta href="URL" rel=relación
    rev=relación title=título>
```

...

```
</a>
```

Tamaño de letra base

```
<basefont size=1,2,3, ...,7>
```

Cambio de tipo de letra

```
<font type="tipo de letra" size={"±1...4"|"1...7"}
    color={color|"#RRGGBB"}>
... </font>
```

Invocación de email

```
<a href="mailto: direccion"> ... </a>
```

Referencias a secciones de páginas

Dentro del documento

```
<a href="#etiqueta">
... </a>
```

Fuera del documento

```
<a href="URL#etiqueta"> ... </a>
```

Caracteres especiales deben ir anteceditos de
& y finalizados con ; o &#número

Ejemplo: á á ; á

Entrada de datos

```
<input type={"text"|"password"|  
"checkbox"|"radio"|"submit"|"reset"|  
"file"|"hidden"|"image"} name="nombre"  
value="valor" checked size="#"  
maxlength="#" scr="URL"  
align={"top"|"middle"|"bottom"|"left"|"right"}>
```

Text: texto de 1 línea

Password: contraseña

Checkbox: valores lógicos si/no

Radio: valor único entre varios disponibles

Submit: envío de los parámetros

Reset: limpiar la forma

File: asocia un archivo a la forma

Hidden: valor escondido

Image: botones gráficos de submit

Nombre: nombre de atributo

Valor: valor inicial para el atributo

Checked: iniciar atributo checkbox o radio

Size: tamaño visible

Maxlength: tamaño máximo

Scr: dirección del botón gráfico

Menús

```
<select name="nombre" size="#" multiple>  
  <option selected value="valor">  
  .... </select>
```

Imágenes

```

```

URL: dirección de la imagen jpg, gif, png

Cadena: texto desplegado

Mapa

```
<map name="etiqueta">
```

...

```
</map>
```

Áreas dentro del mapa

```
<area shape={"rect"|"circle"|"poly"}  
  coords=# href="URL" nohref  
  alt="cadena">
```

Rectángulos: coordenadas de la esquina superior izquierda y la inferior derecha

Círculos: coordenadas del centro y el radio

Polígonos: coordenadas X,Y de al menos 3 puntos



UNIVERSIDAD
DE LOS ANDES

Ejemplo HTML

```
<form method="POST" action="http://usr/people/isabel/WWW/correRT">
  Seleccione la <b>extensi&oacute;n</b> que usted desea
  <select name="rtv">
    <option> Extensi&oacute;n 1
    <option> Extensi&oacute;n 2
    <option> Extensi&oacute;n 3
    <option> Extensi&oacute;n 4
    <option selected> Extensi&oacute;n 5
    <option> &Aacute;rbol_O (6)    </select> <br>
  Cu&aacute;ntas entradas desea usted utilizar?
  <input textarea name="numEnt" 1 7 value="100"> </textarea> <br>
  Seleccione el operador topol&oacute;gico, si lo requiere.... <a href="table1.gif"> tabla 1 </a>
  <select name="sop" size=2>
    <option> disjunto
    <option> toca
    <option> igual
    <option> solapa
    <option> dentroDe
    <option> contiene
    <option> cubre
    <option> cubiertoPor          </select> <br>
  <b> y haga ejecutar su consulta..... </b>
  <input type="submit">          </form>
```



Tecnología básica

- Protocolo HTTP (Hypertext Transfer Protocol, 1990): protocolo genérico de nivel aplicación para intercambiar informaciones en sistemas distribuidos, usado para documentos hipermedia que permite la representación de los datos.
- Funciones:
 - ✓ **Get <URL>?<p1+p2+...>** para solicitar una URL que puede ser un programa accedido vía CGI
 - ✓ **Post <URL>?<p>** igual a get pero los parámetros pasan como un documento
 - ✓ **Head >URL>** recupera la información descriptiva del documento (fechas..)
 - ✓ **Put <URL>** envío de una copia de la URL a un servidor (poco utilizada)
 - ✓ **Delete <URL>** eliminar la URL (raramente autorizada)



Transacciones (Ti)

- Gestión de transacciones atómicas no está integrada en http por ser un protocolo sin estado (sin sesión, ni conexión, ni transacción)
- Simulación de transacciones:
 - ✓ Técnica de URL largas: anexar un identificador de Ti a la URL, el cual debe enviarse cada vez que haya intercambio de mensajes.
 - ✓ Cookies: identificador de Ti en la primera solicitud al servidor, envío del identificador en un campo especial (cookie) almacenado en un archivo texto en el cliente, luego se envía este archivo como un parámetro escondido cada vez que haya intercambio de mensajes entre el cliente y el servidor
- Protocolo SOAP: debe proveer la gestión de Ti
- Uso principal: comercio electrónico



Tecnologías dinámicas-cliente

- Desde los plugins al HTML dinámico (DHTML)
- **Scripts:** programa incluido en la página web interpretado por el navegador y activados por las acciones hechas sobre la página
- ✓ JavaScript `<script language="JavaScript"> ... </script>`
- ✓ VBScript `<script language="VBScript"> ... </script>` Solo en microsoft Explorer
- ✓ Lenguajes de programación con tipos de datos “suaves” (no se declaran las variables), con operadores clásicos de comparación y booleanos, instrucciones de asignación, repetición, decisión y manejo de subprogramas. Adicionalmente gestión de eventos (onClick, onChange, onMouseOver, onMouseOut, etc.)
- ✓ Especiales para programar clientes web
- ✓ Desventaja: falta de estándar para los fabricantes

HTML

Tecnologías dinámicas-cliente

- **Controles ActiveX:** tecnología microsoft para incluir objetos parametrizables y ejecutables dentro de la página web.
 - ✓ Cargados en binario desde el servidor, si no están, e instalados en el disco del cliente.
 - ✓ Con .NET se cargan en código intermedio ~ applets Java
 - ✓ Seguridad: controlada por un certificado del emisor
 - ✓ 1994: Windows 32 bits y OCX (OLE Control eXtension) objetos con varias interfaces pero sin herencia
 - ✓ 1996: extensión a ActiveX
- Objeto distribuido que sigue el modelo COM ejecutado en un contenedor
- ✓ `<object classid id align codebase data height width parameter>...</object>`

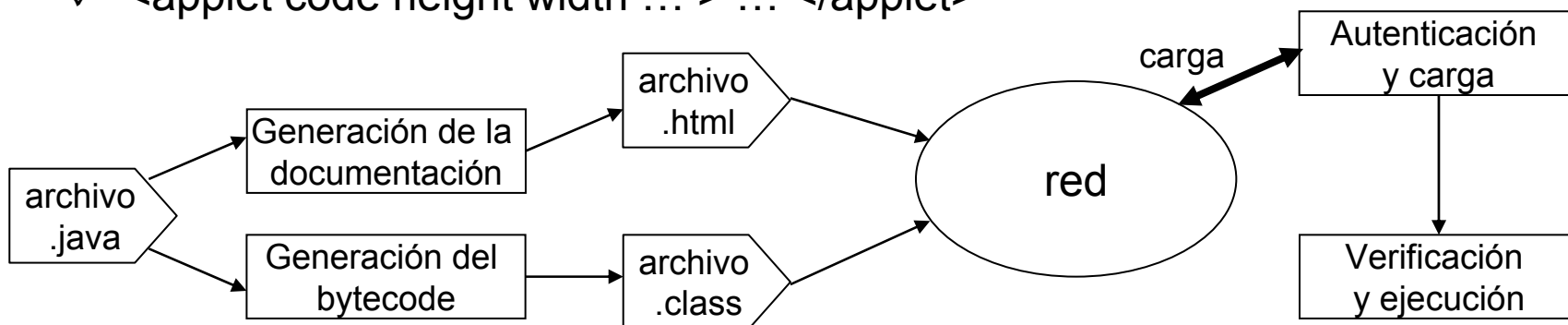
HTML

name=".." value="..."

Tecnologías dinámicas-cliente

- **Applets Java:** subclase de la clase ambiente de Java (`java.applet.Applet`) cuyo código es cargado por el navegador y ejecutado bajo su control en una ventana interna
 - ✓ La máquina virtual Java debe estar incluida en el navegador
 - ✓ Está dirigido por los eventos en la ventana (movimientos del ratón, etc.)
 - ✓ Gestión de interfaz gráfica con *swing* o *awt*
 - ✓ `<applet code height width ... > ... </applet>`

HTML

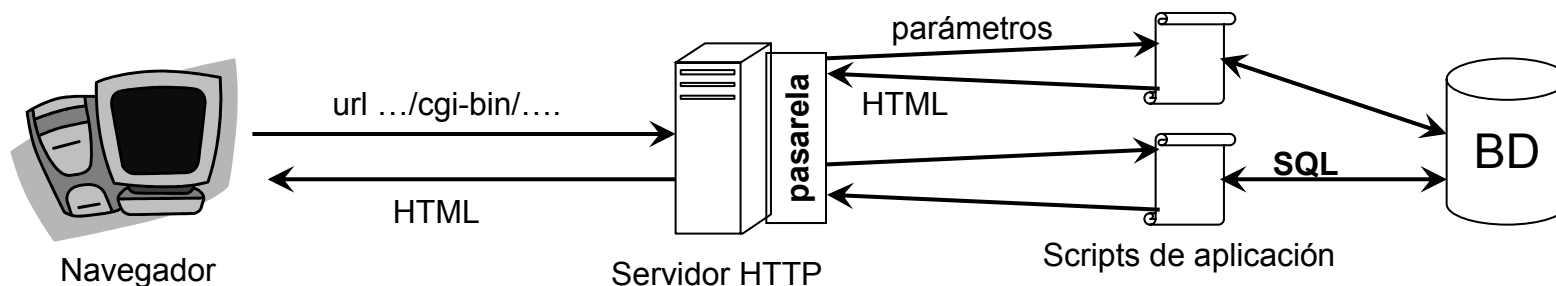


Tecnologías dinámicas-cliente

- **DHTML:** tecnologías HTML, CSS, DOM y JavaScript
 - ✓ Soportado por Explorer 5+ y Netscape 4+, no utiliza plugins
 - ✓ Cascading Style Sheet 1 (CSS1): hoja de estilo que estructura la página
 - Noción de cascada: permite definir varias veces las propiedades de un elemento, pero solo la última es la aplicada. Hay herencia entre ellas. Si está indefinida, se toma la del navegador.
 - H2 { font-family: Arial; font-style: italic; color: red }
 - ✓ Cascading Style Sheet Positioning (CSSP): extensión de CSS para controlar el posicionamiento en la ventana (píxeles)
 - ``
- Document Object Model (DOM): árbol de objetos que representa al documento usado por XML o HTML para la modificación dinámica

Tecnologías dinámicas-servidor

- CGI (Common Gateway Interface): Acoplamiento entre servidores web y programas de aplicación en el servidor según un estándar de pase de parámetros entre cliente y aplicación
- Script CGI: programa shell o Perl que ahora puede ser en cualquier lenguaje compilado o interpretado (VBScript, JavaScript)
 - ✓ Secuencia de comandos interpretados que pueden recibir parámetros desde los clientes web, procesarlos accediendo a fuentes de datos y devolver los resultados en forma de páginas web
- `<form method="POST" action="http://usr/people/isabel/WWW/correRT">` de html. Cada llamada crea un proceso servidor separado.
- Puede solicitar consultas en SQL a una BD





Tecnologías dinámicas-servidor

- API (Application Programming Interface): librería dinámica de programas compartidos en un servidor http que permite el llamado a programas de aplicación (PA), que pueden soportar multihilos
 - ✓ API HTTP: mecanismo propietario que permite ejecutar PA lanzados en el contexto de un servidor HTTP particular con pase de parámetros por una interfaz propietaria.
 - ✓ Lenguajes compilados (C) o de scripts (JScript)
 - ✓ ISAPI (Microsoft), NSAPI (Netscape) o API-Apache
 - ✓ Problema de compatibilidad y de versiones
 - ✓ Mejoran el rendimiento evitando recargar los binarios en cada llamada
 - ✓ Introducen una gestión de la sesión de usuario
 - Ejemplo: Netscape (NameTrans, PathCheck, Session, Request, Parameters, etc.)



Tecnologías dinámicas-servidor

- FastCGI: mejora del CGI de Open Market que permite gestionar varios procesos asociados a un servidor
 - ✓ Trata las llamadas vía una conexión, procesa y devuelve los resultados por la conexión
 - ✓ PA multihilos y pueden jugar diferentes roles, como un:
 - CGI normal,
 - Filtro de archivos html antes del envío, o
 - Controlador de autorizaciones
 - ✓ Operaciones:
 - FCGI_ACCEPT() para aceptar una solicitud
 - FCGI_PARAMS() para tratar las variables de ambiente
 - ✓ No tienen mejor rendimiento que los servidores de aplicaciones



Tecnologías dinámicas-servidor

- Servlets Java: programa Java en el servidor que se ejecuta luego de una llamada http bajo el control de un motor de ejecución que extiende las funcionalidades de un servidor web con API extensibles
 - ✓ Integrados en la arquitectura J2EE (Java 2 Enterprise Edition)
 - ✓ Independientes de la plataforma
 - ✓ Desarrollados por Sun, <http://java.sun.com/products/servlet>
 - ✓ GenericServlet (javax.servlet.GenericServlet) clase abstracta
 - Public void init (ServletConfig config) inicia el servlet luego de su creación
 - Public void service(ServletRequest llamada, ServletResponse respuesta) trata la solicitud pasada en llamada y genera el resultado en respuesta (implementación obligatoria)
 - Public void destroy() destruye el servlet
 - Public ServletConfig getServletConfig() recupera los parámetros de configuración del servlet
 - Public String getServletInfo() regresa una cadena que describe el servlet

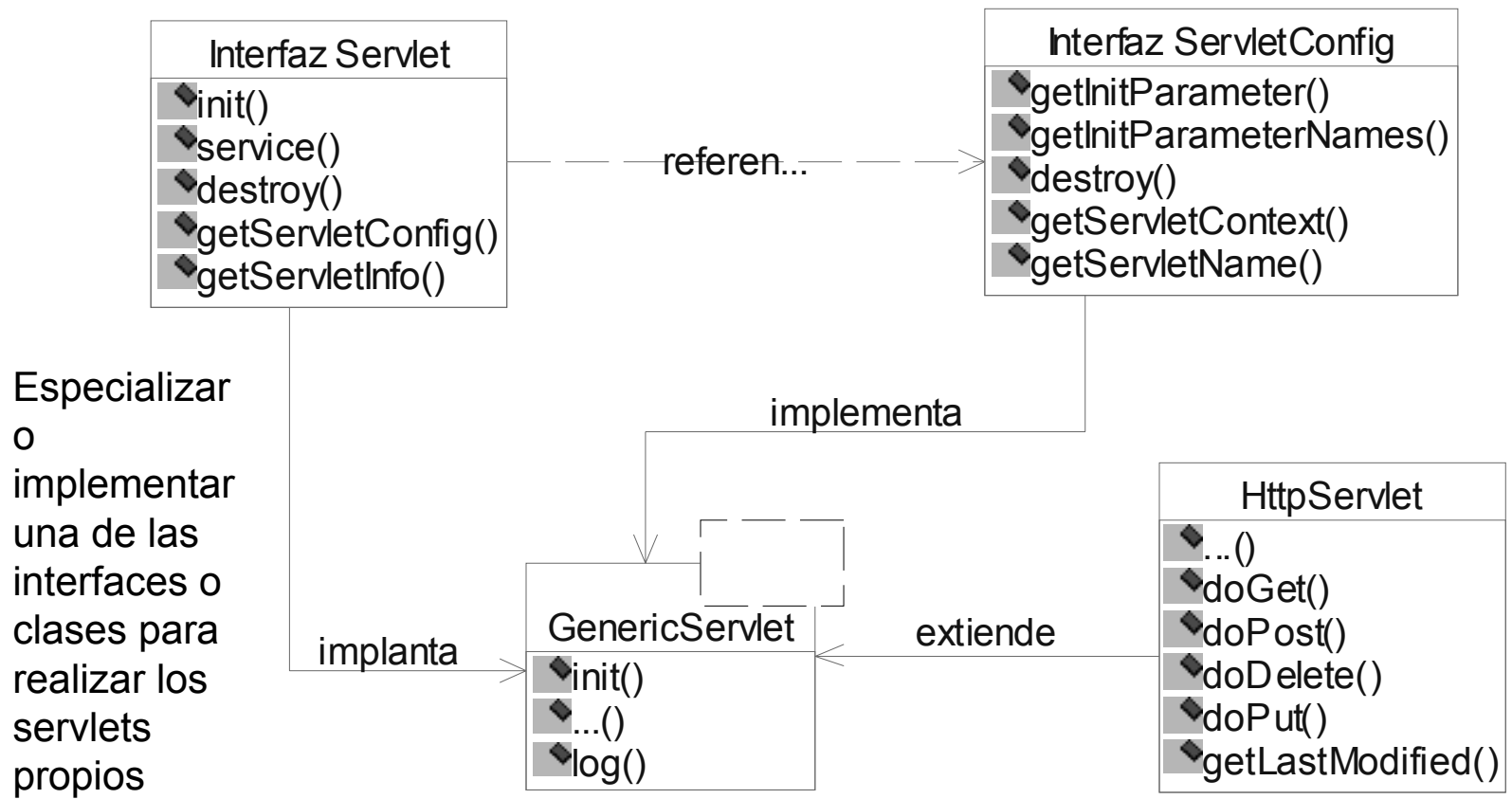


Tecnologías dinámicas-servidor

- log(String mensaje) y log(String mensaje, Throwable t) bitácora del servlet
- ✓ Http-Servlet (javax,servlet.http.HttpServlet) extensión de la anterior para http
 - Public void do???(HttpServletRequest llamada, HttpServletResponse respuesta) throws ServletException, IOException
 - Métodos posibles: doGet(), doPost(), delete() o doPut()
- Configuración: del objeto ServletConfig
 - ✓ Archivo de propiedades del servlet
 - ✓ J2EE archivo XML con el nombre del servlet, su clase y parámetros
 - Public String getInitParameter(String nombre) regresa el valor del parámetro nombre
 - Public Enumeration getInitParameterNames() regresa los nombres de los parámetros
 - Public ServletContext getServletContext() regresa el objeto que define el contexto
 - Public String getServletName() regresa el nombre del servlet

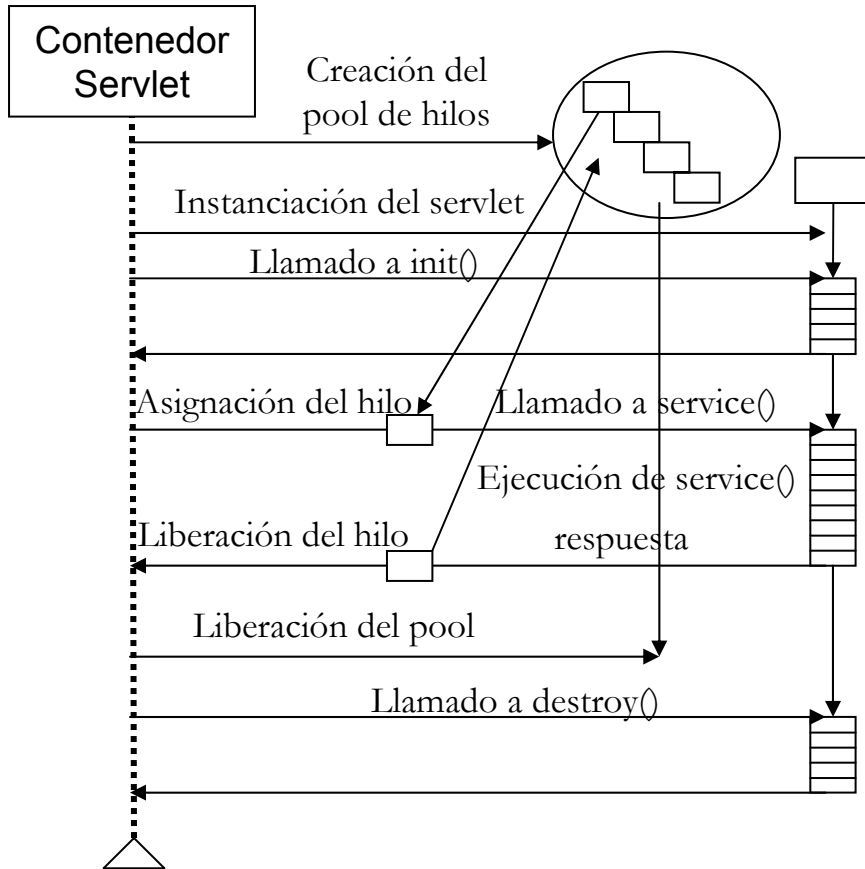


Tecnologías dinámicas-servidor



Especializar o implementar una de las interfaces o clases para realizar los servlets propios

Ciclo de vida de un servlet



- Creación del pool de hilos
- Instanciación del servlet
- Inicialización
- Por cada solicitud http se usa un hilo para ejecutar el servicio invocado
- Elaboración de la respuesta
- Liberación del hilo
- Cierre del servlet
- Liberación del pool de hilos

Sesiones en servlet

➤ API para gestión de sesiones:

- ✓ Interfaz HttpSession (javax.servlet.http.HttpSession) debe ser implementada por el motor servlet
- ✓ La implementación de la sesión puede ser por:
 - Cookies
 - URL largas con identificador de sesión pasado por parámetros
 - Campos en caché con el identificador de sesión, etc.
- ✓ Métodos:
 - getId(), invalidate(), isNew(): identificadores de sesión
 - getCreationTime(), getLastAccessTime(), setMaxInactiveInterval(), getMaxInactiveInterval(): duración de la sesión
 - getAttribute(), getAttributeNames(), removeAttribute(), setAttribute(): atributos del estado de una sesión



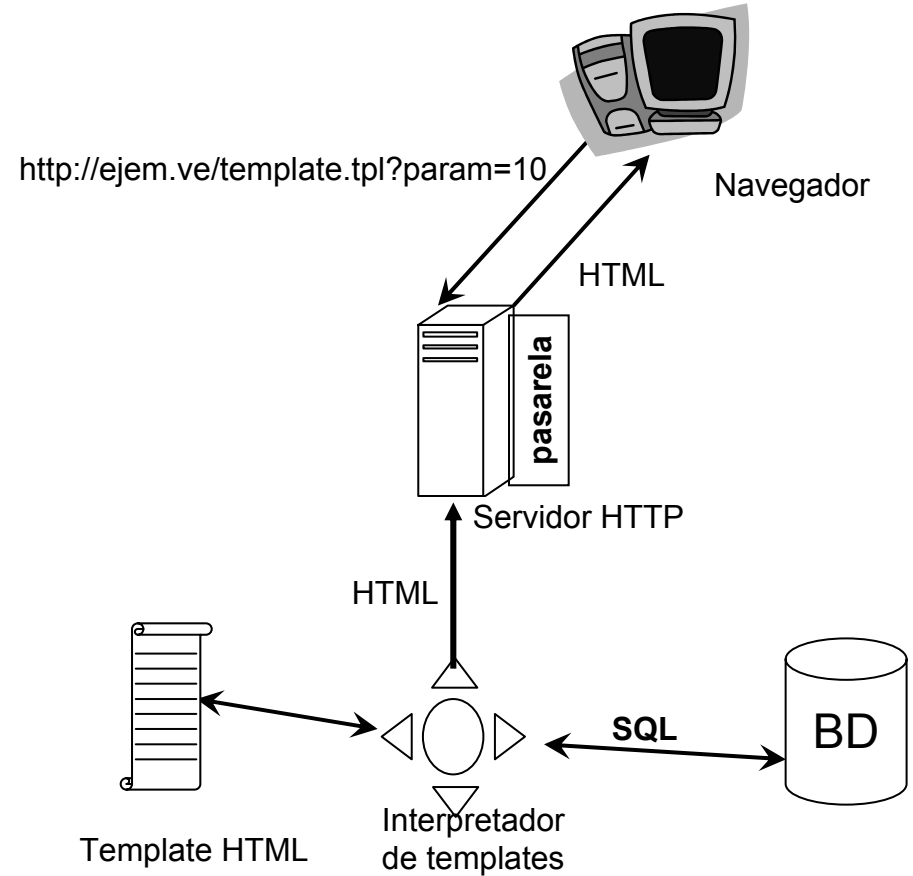
Páginas activas-servidor

- Combinan HTML o XML, scripts y componentes en el servidor en un único archivo representando la página ASP, JSP o PHP
- **Template html:** página html que incluye variables y scripts, los cuales generan los datos que reemplazan las variables luego de ejecución
 - ✓ Es un archivo html almacenado en el servidor que contiene marcas especiales para invocar los scripts servidores escritos en: Perl, php, VBScript o JScript
 - ✓ Uso: preparar los parámetros para consultas SQL o formatear resultados

```
<html>
  <body> <h1> titulo </h1>
        <%select Q1 = "select ..." %>
        <table>
          <tr> <td> "A1 " </td> <td> "A2 " </td> </tr>
          <%iterate Q1 %>
            <tr> <td> %Q1.A1% </td> <td> %Q1.A2 % </td> </tr>
          <%end-iterate%>
        </table>
      </body>
</html>
```


Páginas activas-servidor

- Select envía la consulta al SGBD haciendo que se ejecute
- Iterate selecciona cada tupla resultado, desde la primera a la última
- Fetch lee la tupla resultado actual y recupera sus atributos en las variables A1, A2, etc.
- La solicitud a http permite cargar el archivo template para su interpretación
- Esquema de ejecución



- Personal Home Page o Preprocessor HyPertext (PHP) creado por R. Lerdorf en 1995, lenguaje de programación para realizar templates HTML
- Tecnología “open source” propuesta por Apache para incluir templates html escritos en un lenguaje simple, ejecutados por invocación del template y cuyos resultados se integran en un código html
- Aplicaciones con BD como: MySQL, Postgres, Oracle, etc.
- Interpretado
- Código php en la página html:
 - ✓ **<?php códigoPHP ; ?>**
 - ✓ **Echo “ texto ”; o print(“ texto ”);** incluyen la salida en la página html
 - ✓ **<script language=“php”> códigoPHP; </script>**

```
<html> <head>
    <title> Título de la página </title>
</head> <body>
    <?php echo “Hola .... ”; ?>
</body>
</html>
```




- Variables que comienzan con \$ que reciben el tipo del valor que resulta (débilmente tipeado)
- Especialmente adaptado para BD, uso de variables con el nombre del parámetro para tener el valor en ellas
- Tres tipos de datos: int, double y string
- Definición de constantes: define("constante", "valor")
- Operadores: aritméticos, de relación, cadenas y lógicos
- Estructuras de decisión: if, if-else, swith
- Estructuras de repetición: while, do-while, for
- Evaluación de código: require(), include()
- Salida: exit
- Pase de parámetros: por valor o referencia

```
<?php mysql_connect("host", "raíz", ..);  
$conexionExitosa=mysql_selectdb("p1");  
if (!$conexionExitosa)  
    echo "Falla de conexión ";  
mysql_query(  
    "insert into persona (ci, nom, ape, feNac)  
    values('$c', '$n', '$a', '$fn') "  
    or die("Falla de inserción");  
echo "Gracias... "; ?>
```

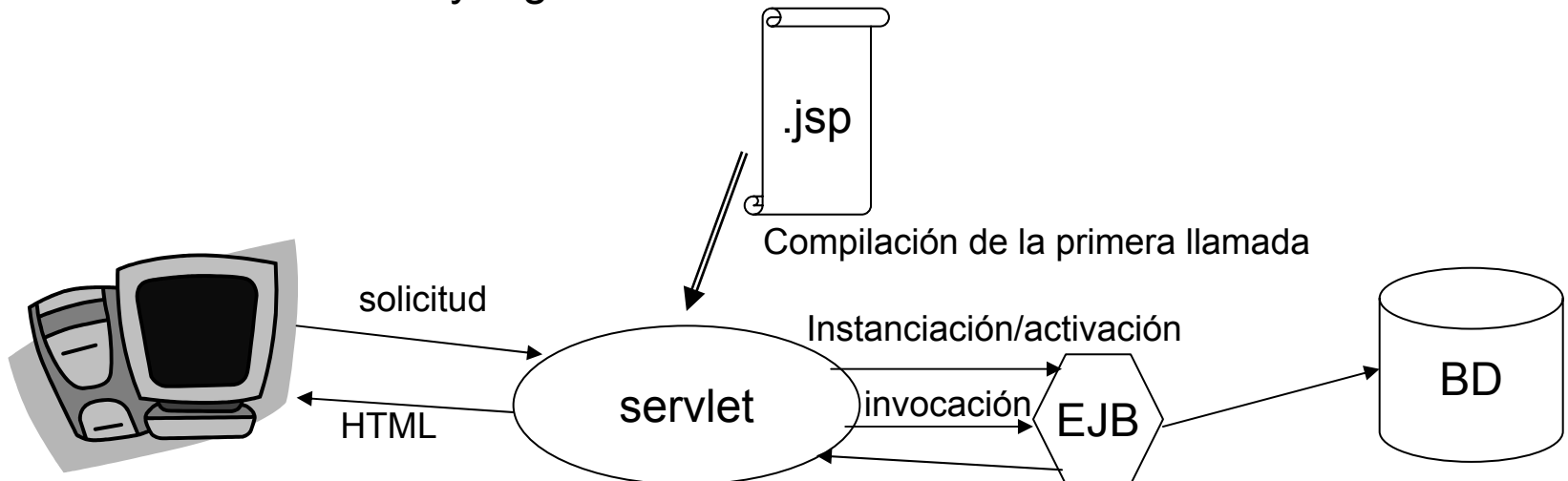
- Tablas: \$tab[]=array("v1", "v2", ...)
 - ✓ \$tab[\$i]
 - ✓ Funciones: list(), each(), next(), prev()
 - ✓ Indexación por cadenas posible
- PHP4: orientado por objetos
 - ✓ Clases con atributos y métodos
 - ✓ Constructor genérico new
 - ✓ \$variable->propiedad
 - ✓ Herencia: class C1 extends C2 {...}

- Active Server Page-Microsoft: tecnología que permite incluir scripts en los templates html ejecutados por el servidor antes de enviar la página al cliente
- Páginas activas interpretadas por Internet Information Server disponible también en .NET
- Versión básica: objetos DCOM
 - ✓ Request: representa la solicitud CGI
 - ✓ Response: la respuesta generada
 - ✓ Application: permite tener los parámetros de la aplicación
 - ✓ Session: da las características de la sesión generada por cookies
 - ✓ Server: da las características del servidor
- Acceso a BD vía Active Data Object (ADO)

```
<html>
  <% emp=Request.QueryString("ci")
    Set NObj=Server.CreateObject("Emp.ci")
    if NObj.GetEmpleado(emp)=False then Server.Redirect("Cl/entryform.htm")
  %>
  <h1> Empleado seleccionado: <%=NObj.Empleado%> </h1>
  <table>
    <tr> <td>Nombre </td> <td>Apellido </td> <td>Fecha de nacimiento </td> </tr>
    <tr> <td><%=NObj.nombre%> </td> <td><%=NObj.apellido%> </td> <td>
  <%=NObj.fechaNacimiento%> </td> </tr>
  </table>
  <h2> Fecha de la consulta: <%=time()%> <%=date()%> </h2>
</html>
```

- Java Server Page-Sun: tecnología propuesta por Java que permite incluir en los templates html los programas java ejecutados luego de la invocación del template y cuyos resultados se integran en html
- Posibilidad de tener documentos mixtos html-java o xml-java
- Scriptlet: porción de código java incluida en una página jsp que permite generar una parte variable de html resultado de la ejecución de la página
- Compilados en servlets que implementan tres interfaces de la clase `javax.servlet.jsp.JspPage`
 - ✓ `jspInit()`: inicio de la página 
 - ✓ `jspDestroy()`: destruye la página luego de una parada del servidor
 - ✓ `jspService()`: código java del servlet resultado de la compilación de la pág.

- Actualizaciones provocan recompilación de la página
- Es posible agrupar el código java en componentes Java Beans o EJB e invocarlos en páginas JSP
- Directivas: useBean y taglib



- Directivas: precisan el tipo de servlet que resultará luego de la compilación de la página (JSP directives). Para declaración de variables globales entre el motor JSP y las clases JSP.
 - ✓ Sintaxis: `<%@ nomDirectiva atributo="valor" at2="v2" ...%>`
 - ✓ Sintaxis XML: `<jsp :directive nomDirectiva atributo="valor" at2="v2" .../>`
 - ✓ Principales directivas: page, include, taglib
- Scriptlet: código java
 - ✓ Tres partes: `<%! Declaración de variables y métodos %>`, `<% instrucciones Java %>` y `<%= resultado o expresión Java %>`
- Acciones: permite parametrizar, incluir e invocar el código adicional al servlet que resultará luego de la compilación
- Marcas personalizadas: mecanismo para invocar funciones para las marcas



```
<html>
  <head>
    <title> Ejemplo en JSP </title>
  </head>
  <body>
    <!-- fijar los parámetros de la página -->
    <%@ page language="java" session="false" %>
    <!-- declaración de una variable tipo carácter -->
    <% char c=0; %>
    <%
      for(int i=0; i < 26; i++)
      {
        c= (char) (0x41 + i);
      }
    %>
    <!-- resultado -->
    <%= c %> <% } %>
  </body>
</html>
```

<jsp:include> inclusión de un recurso

<jsp:forward> pasar una solicitud http a otro recurso

<jsp:plugin> activar un plugin en el cliente

<jsp:param
name="nombre"
value="valor"> puede incluirse para colocar parámetros dentro de las acciones

<jsp:useBean id="#"
scope=
"page|request|session
| application" class=
"nombreClase">
invocación de un bean



- URN (Universal Resource Names) dirección lógica de un recurso
- URL (Universal Resource Locator) dirección física
- URI (Universal Resource Identifier) identificador del recurso
- DNS (Domain Name Server) <http://www.w3.org/TR/REC-xml-names>
Identificación por el atributo `< elemento xmlns:prefijo='URI'/>`
- XLink (XML Linking Language) <http://www.w3.org/TR/xlink> norma encargada de controlar los enlaces externos entre los documentos XML usando xPointer para gestionar las partes del mismo
- XPointer (XML Pointer Language) norma que controla los enlaces internos entre las partes del documento(s) XML usando XPath para su gestión

Espacio de nombres ...

- Conjunto de nombres que se utilizan como tipos de elementos o nombres de atributos
- Permiten clasificar elementos como miembros de un espacio de nombres determinado y así
 - eliminar ambigüedades y
 - permitir a las aplicaciones el uso de espacios de nombres propios

Espacio de nombres:
comentario

E
J
E
M
P
L
O

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes">
<comentario:comentarioCD xmlns:comentario="http://www.cd.def/revcd" xmlns:cd="http://www.cd.def/cd">
<comentario:autor> Isabel Besembel </comentario:autor>
<comentario:puntaje="8" de="10" />
  <cd:autor> Los Beatles </cd:autor>
  <cd:titulo xml:space="preserve"> Yellow submarine </cd:titulo>
  <cd:fecha>1967</cd:fecha>
  <cd:pista numero="1"> ...</cd:pista>
  <cd:pista numero="2"> ...</cd:pista>
  .....
```

Espacio de
nombres: cd



Espacio de nombres ...

- Nombre del espacio debe ser un URI válido y único
- Nombre calificado:
 - Cadena con prefijo del tipo prefijo:nombre
 - Nombre
- Alcance:
 - ✓ se aplica al elemento en que está especificada y a todos los elementos dentro de su contenido
 - ✓ se vuelve a definir mediante otra declaración que tenga exactamente el mismo prefijo
- Predeterminado: todos los elementos no calificados pertenecen a él
 - ✓ se elimina cuando se establece una cadena vacía

Espacio de nombres ...

➤ Ejemplo: Predeterminado

```
<xsl:transform xmlns="http://www.algo.com/algo" version="1.0">  
<history vmlns="">  
</history>  
</xsl:transform>
```

Anulación del
predeterminado

➤ Atributos: no están unidos a ningún espacio de nombre (opción predeterminada)

```
<X xmlns="http://www.algo.com/algo" xmlns:history="http://www.algo.com/historia"  
  version="1.0">  
<A history:B="foo" C="bar"/>
```

A

B

```
<X xmlns="http://www.algo.com/historia" xmlns:history="http://www.algo.com/historia"  
  version="1.0">  
<A history:B="foo"/>
```

A y B pertenecen al mismo espacio
de nombres

Espacio de nombres

- Atributos únicos: ninguna marca puede contener atributos con el mismo nombre o nombres equivalentes calificados

```
<X xmlns:n1="http://www.algo.com/algo" xmlns:n2="http://www.algo.com/historia">
```

```
<x a="1" a="2"/>
```

```
<y n1:a="1" n2:a="2" />
```

```
</X>
```

x e y NO son
válidos

```
<X xmlns:n1="http://www.algo.com/algo" xmlns="http://www.algo.com/historia">
```

```
<x a="1" b="2"/>
```

```
<y n1:a="1" a="2" />
```

```
</X>
```

x e y correctos

- Un URI de espacio de nombre NO tiene que apuntar a nada
- No existe un espacio de nombres nulo o básico