

# Generalidades de XML

...y otras cosas relacionadas.

Mario A. Valdez-Ramírez.

# Agenda.

- Árboles de datos.
- HTML.
- Qué es el XML.
- Qué es el XSL.
- De XML al HTML.
- El intérprete de XML de Microsoft.
- B2B con XML (BizTalk).
- .NET (o “cómo le hago para hablar de algo que no viene al caso...”).

# ¿Qué es el XML?

Repasemos lo conocido...

# HTML: Lo bueno.

- El hipertexto funciona (es navegable).
- Es multiplataforma.
- Tiene una curva de aprendizaje muy plana.
- Barato (muchos editores, visores, verificadores, etc., gratuitos y comerciales).
- Base de información grande.
- Los navegadores son baratos, sencillos de construir y de usar y poderosos.

# HTML: Lo malo.

- Pobre herramienta de presentación.
  - » Poco control de espaciado.
  - » Problemas con el control de guiones, *kerning*, justificación y otras manipulaciones de texto.
  - » EL uso de columnas es problemático.
- Pobre herramienta de marcaje (*markup*).
  - » No se pueden agregar etiquetas nuevas.
  - » No es modular, poca oportunidad de reciclar.
  - » Hay demasiado código inválido publicado actualmente.
- Es una representación de un documento, no de información.

# HTML: Lo peor.

- No puede ser extendido elegantemente.
  - » Las etiquetas son fijas y rígidas.
  - » Las compañías y personas involucradas en hacer extensiones no saben de composición (*typesetting*) ni edición estructurada.
  - » Es campo de batalla comercial (Mozilla vs IE).

# HTML: Lo nuevo.

- Las hojas de estilo en cascada (*cascading style sheets, CSS*).
  - » Netscape 4.0 y superior.
  - » Internet Explorer 3.0 y superior.
  - » Opera y otros navegadores en sus últimas versiones.
- La versión 1, (CSS1) emitida como recomendación del W3C en 1996.
- Separa el contenido del formato.
- Mayor control sobre la apariencia y posición.

# Se requiere algo nuevo...

- Barato, veloz y sencillo:
  - » Para crear documentos.
  - » Para procesar documentos.
  - » Para presentar documentos.
- Extensible:
  - » Un conjunto de reglas, no un conjunto de etiquetas.
- Compatible con el HTML:
  - » Debe tener una manera sencilla de convertir en y desde HTML.
- Capaz de representar información en cualquier forma, no solo documentos.

# Nace el XML.

- En febrero de 1998, nace el XML como un borrador para una recomendación del W3C.

# Ahora... un ejemplo.

```
<Cliente ID="HVet950283">  
  <Nombre>Hospital Veterinario Kermit</Nombre>  
  <Direccion verificada="si">  
    <Calle>Padre Mier 1528</Calle>  
    <Ciudad>Monterrey</Ciudad>  
    <Estado>NL</Estado>  
    <CodigoPostal>64000</CodigoPostal>  
  </Direccion>  
</Cliente>
```

**Sintaxis simple**

**Legible por personas**

**Muy parecido al HTML**

# El XML es...

- El Lenguaje de Marcaje Extensible (*Extensible Markup Language, XML*).
  - » Un metalenguaje de marcaje.
  - » Una sintaxis utilizada para crear lenguajes declarativos.
- Una recomendación técnica del W3C.
  - » Es un estándar del W3C, no de alguna compañía.
- Multiplataforma, simple, fácil de aprender.
  - » Es fácil construir herramientas para XML.
  - » Optimizado para usarse en Internet.
- Libre (y gratuito).

# El XML **no** es...

- Un lenguaje de marcaje (*markup*).
  - » No. Es un estándar que especifica una sintaxis para crear lenguajes de marcaje.
- Solo para Web.
  - » No. Puede ser usado para describir y comunicar cualquier información estructurada.
- Un superconjunto del HTML.
  - » No. Aunque el HTML puede ser definido con sintaxis de XML.
- Un invento de [x compañía].
  - » No. XML es un estándar creado por el W3C y soportado por compañías e instituciones de todo el mundo.

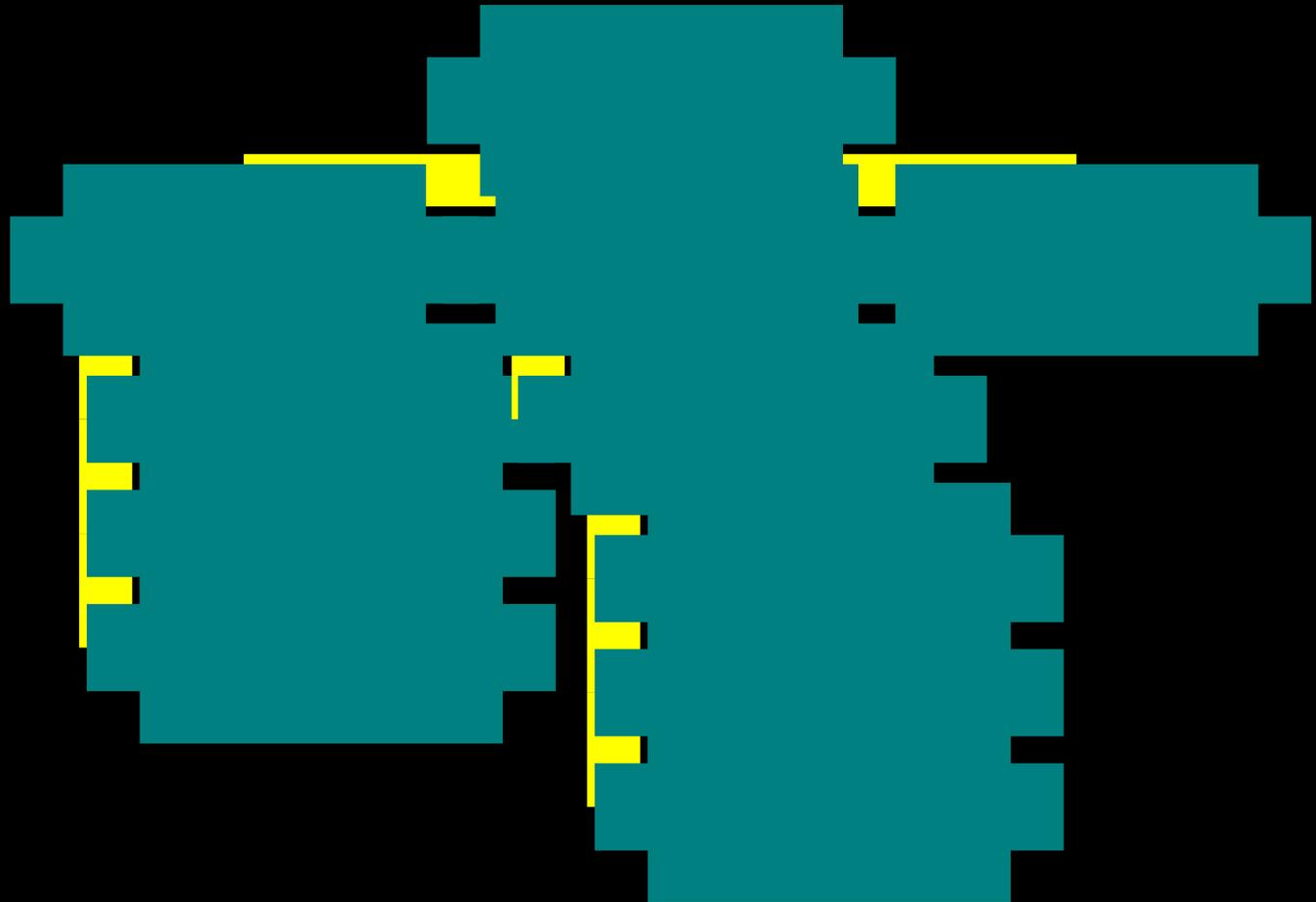
# El XML sirve para...

- Hacer publicación electrónica independiente del medio.
- Crear protocolos para el intercambio de datos entre miembros de una industria.
- Facilitar el procesamiento de datos usando software barato.
- Permite a las personas visualizar la información de la manera que quieran.
- Proporcionar metadatos que mejoran la calidad de la búsqueda de información.

# Un documento XML es...

- Una colección de piezas llamadas "entidades".
- Texto y etiquetas en Unicode.
- Válido, o por lo menos bien formado.
  
- Representa una jerarquía de datos.

# Jerarquía de datos.



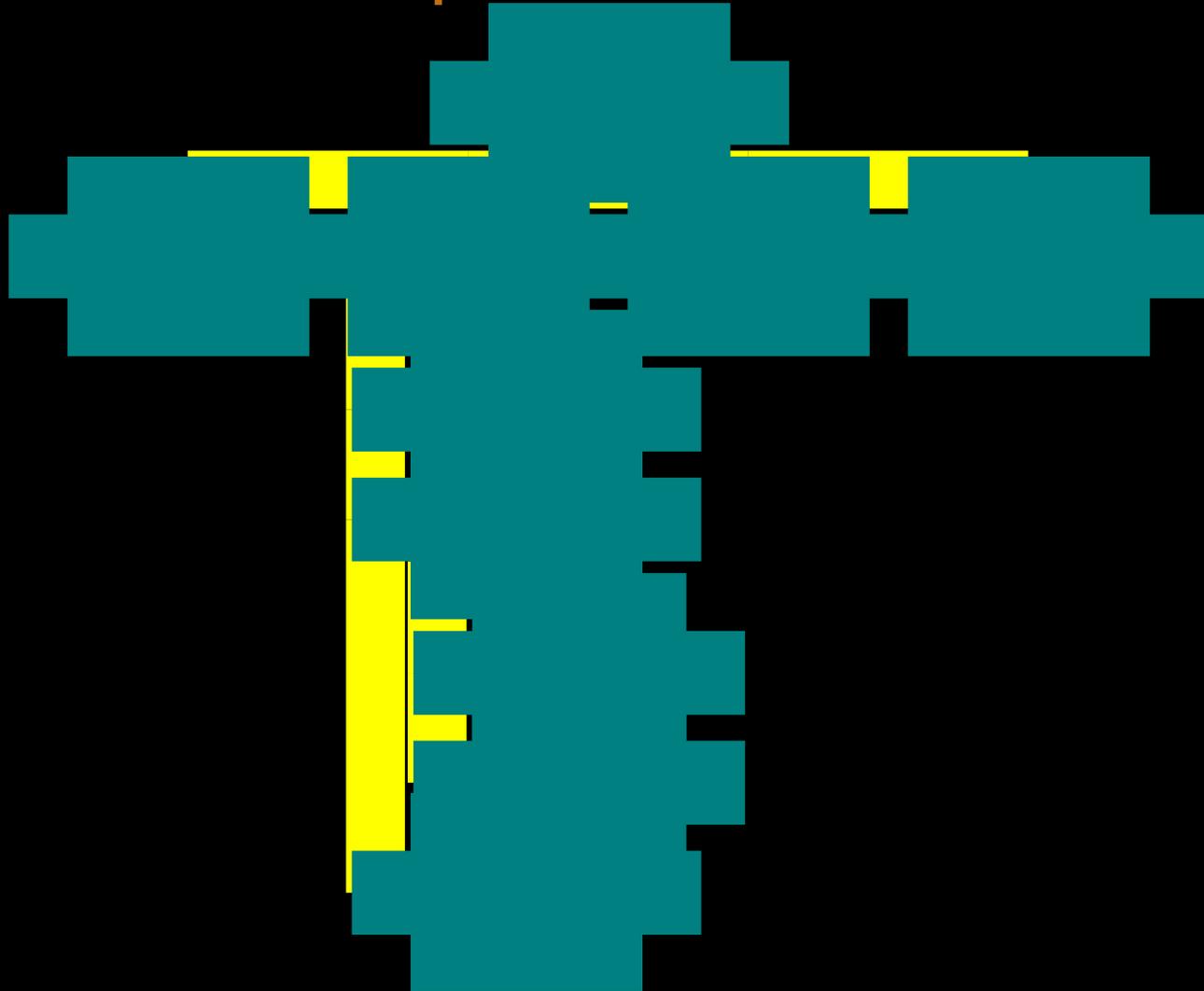
# Datos.

- Los datos estructurados pueden ser modelados como árboles o jerarquías.
- Un árbol de datos tiene uno o varios nodos.
- Cada nodo representa una unidad de información, y puede contener más nodos o unidades.
- La forma de modelar un árbol de datos es arbitraria y depende de las necesidades de quien modela.

# Forma y contenido.

- Un árbol o jerarquía de datos es representa solamente el contenido de ciertos datos.
- No incluye información sobre cómo presentar o utilizar esos datos.

# Jerarquía de datos.



**Ejemplo**

# El procesador XML (*parser*).

- Software que reconoce e interpreta las reglas del XML.
  - » También se le llama analizador o intérprete XML.
- Con XML bien formado:
  - » Revisa que el documento siga las reglas del XML para considerarse bien formado.
- Con XML válido:
  - » Revisa una DTD XML, luego
  - » revisa el documento XML con las reglas XML, luego
  - » revisa el documento XML con las reglas del DTD.

# Semántica XML.

# Semántica XML.

- El papel de la semántica.
  - » Agrega procesamiento (verbos) al documento XML (sustantivos y adjetivos).
- Semántica de visualización.
  - » Indica cómo debe de formatearse un elemento.
- Semántica de procesamiento.
  - » Indica cómo debe procesarse cada elemento.

# El XSL.

Lenguaje extensible de hojas de estilo  
(*extensible stylesheets language*).

# XSL.

- HTML: Formato sin estructura.
  - » Lenguaje de composición (*typesetting*).
  - » No extensible.
- CSS: Formato mejorado, sin estructura.
  - » Lenguaje de hojas de estilo.
  - » Ignora el formato predeterminado del HTML, pero no puede modificar la estructura básica del documento.
- XML: Estructura sin formato.
  - » Define los elementos.
  - » Crea una estructura jerárquica de un conjunto de información.

# Agregando formato a la estructura.

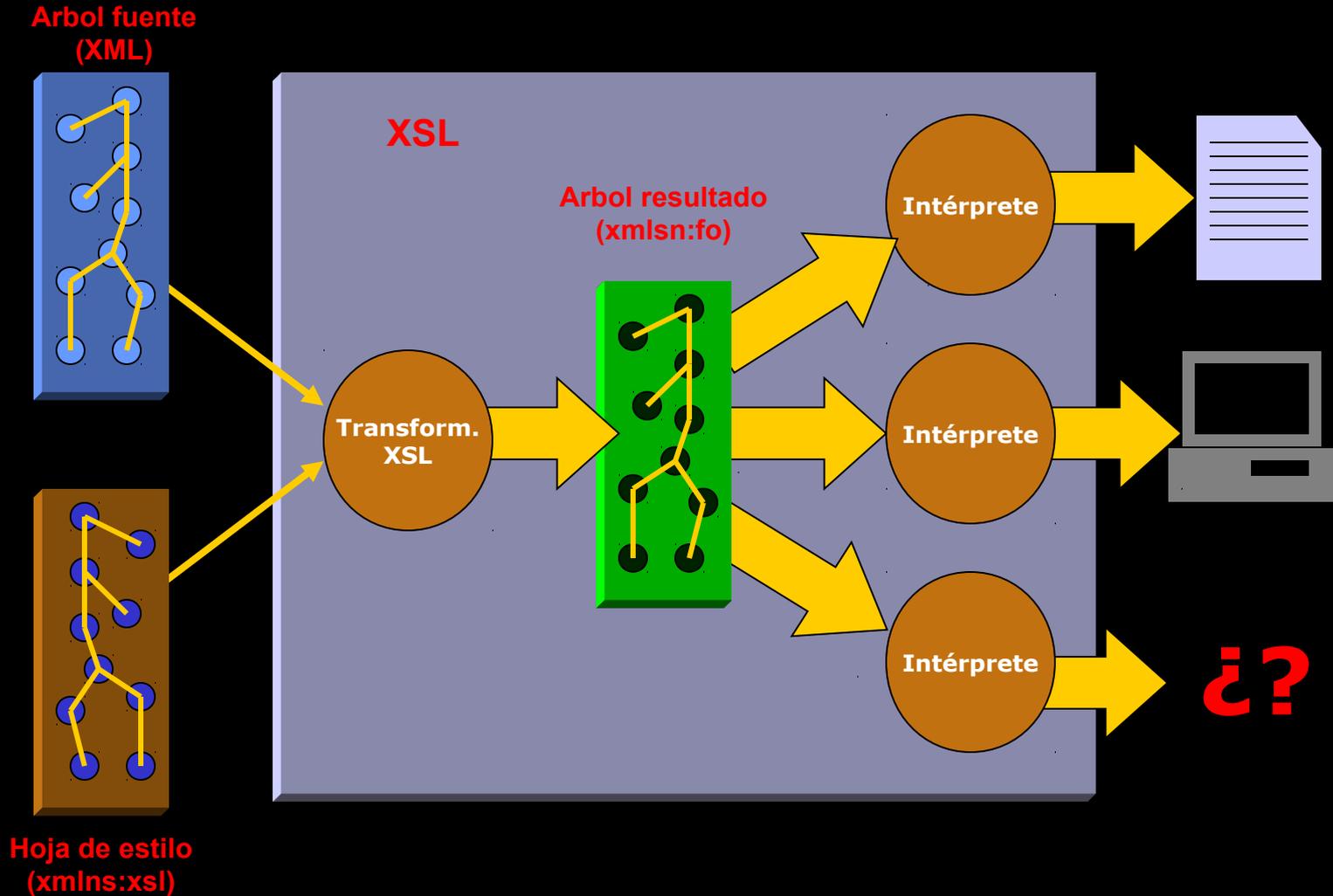
- Lenguaje de hojas de estilo.
- Proporciona definiciones semánticas (verbos o acciones) para los elementos.
- Consiste de dos partes:
  - » Lenguaje para transformar XML.
  - » Vocabulario para especificar semántica de formato.

# XSL.

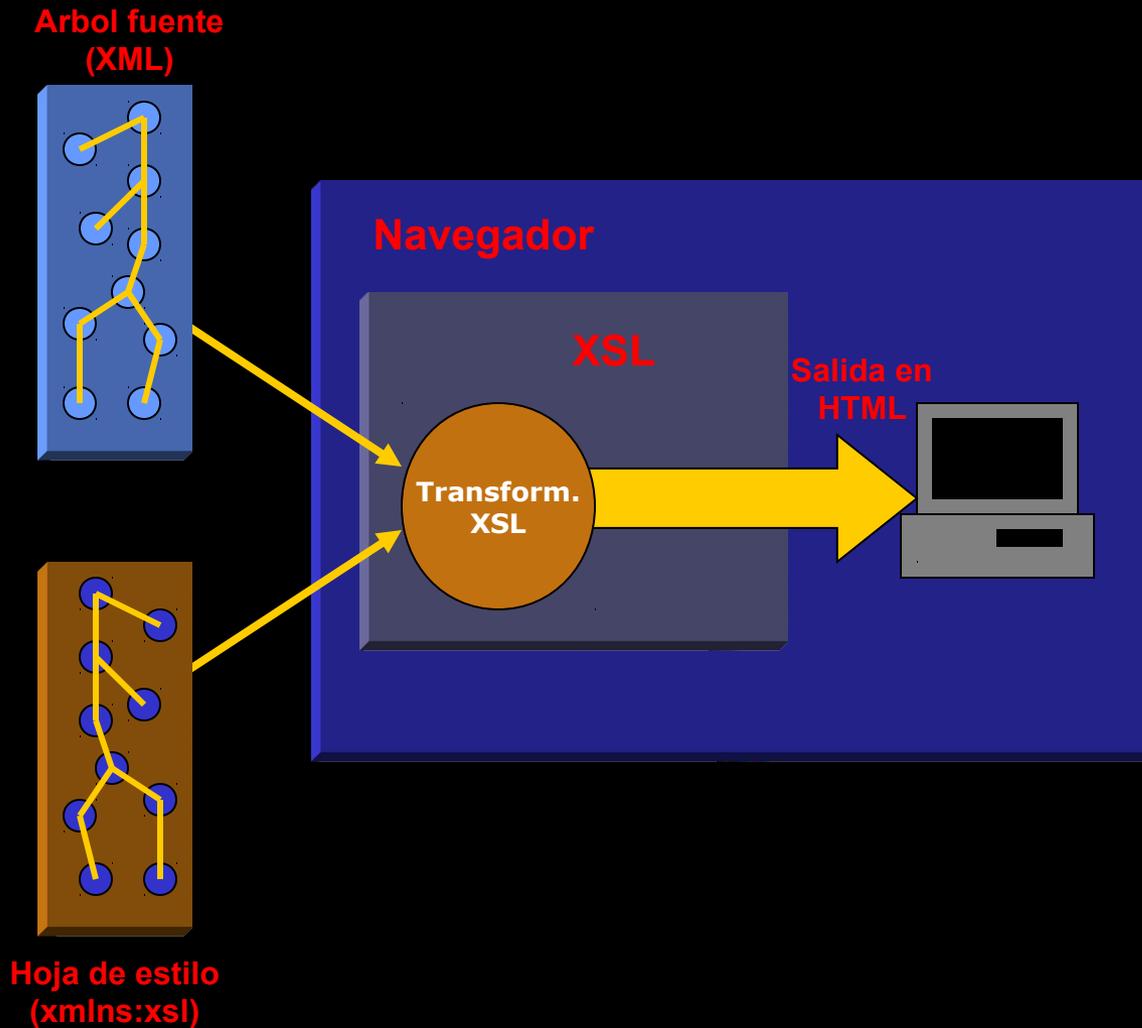
- Un lenguaje para expresar hojas de estilo.
- Proporciona semántica de visualización para el XML.
  - » Relaciona elementos XML con HTML o con otros lenguajes de formato (PDF, LaTeX, PostScript, etc).
- Soporte funcional para CSS.
  - » Simple, sintaxis conocida.
  - » Los principiantes pueden aprender rápido.

**Ejemplo**

# Cómo funciona el XSL.

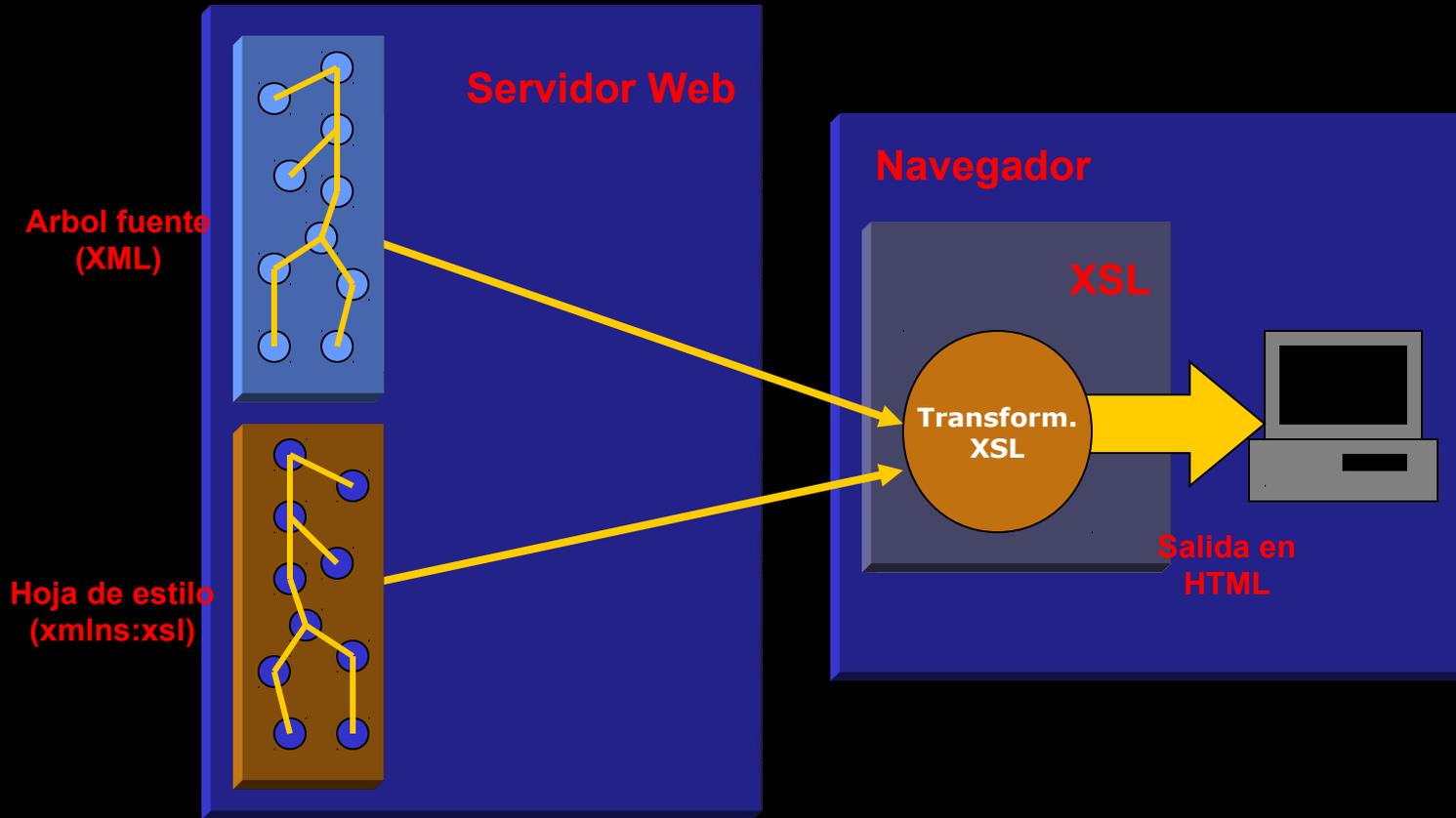


# Cómo funciona el XSL en el navegador.

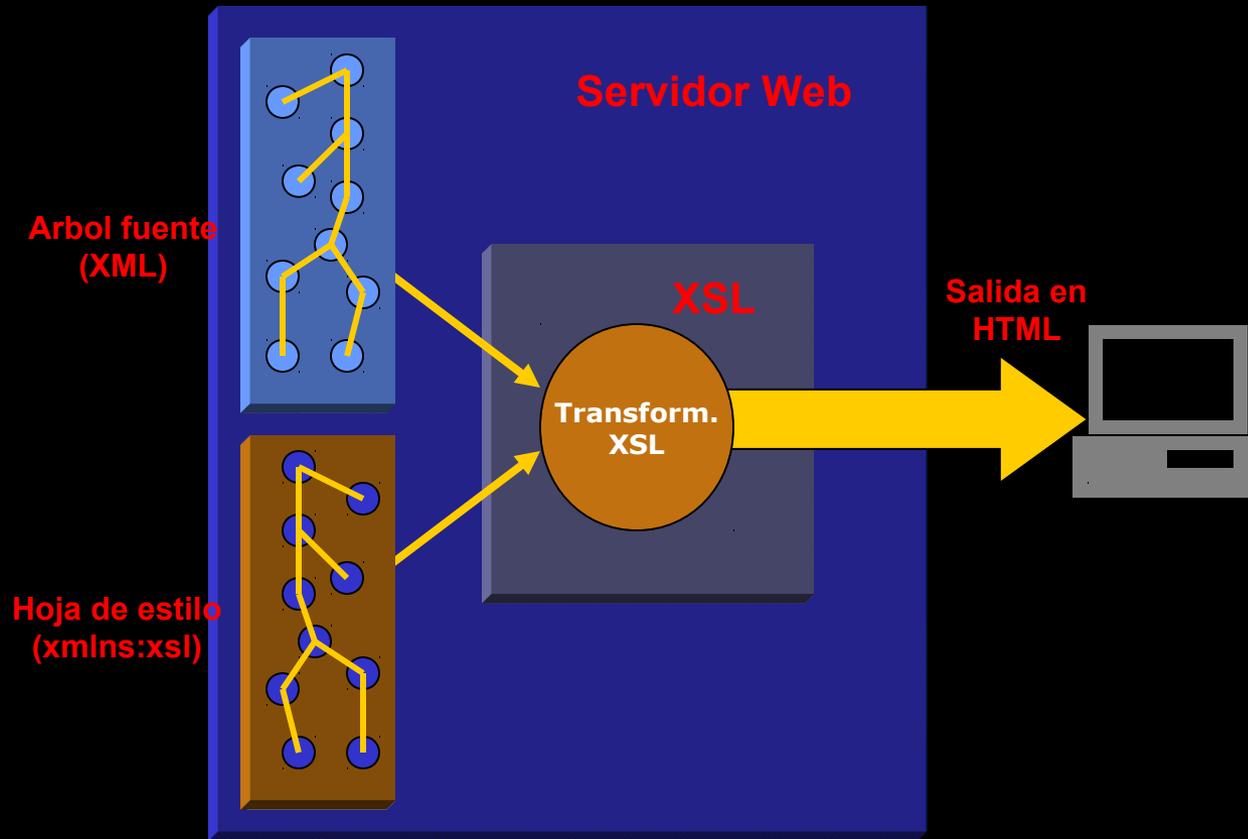


**Demo**

# Cómo funciona el XSL en el servidor sin conversión HTML.



# Cómo funciona el XSL en el servidor.



**Demo**

# Plantillas XSL.

- Un documento XSL aplica una o varias plantillas (*templates*) al código fuente XML.
- Un archivo XSL es una secuencia de **plantillas** que se aplican a una o más etiquetas XML de acuerdo a un **patrón**.

**Ejemplo**

Islas de datos XML.

# Islas XML.

- XML dentro de una página HTML.
- Invoca una instancia del procesador XML del cliente.
- Puede ser identificada por un ID.
- Puede controlarse y modificarse con scripts en el cliente.

# Ejemplos de islas XML.

```
<html>
```

```
<head>
```

```
  <title>Islas XML</title>
```

**Isla XML**

```
<XML id="info-clima">
```

```
  <clima><ciudad>
```

```
    <nombre>Mexico DF</nombre>
```

```
    <reporte>
```

```
      <alta>27</alta><baja>18</baja>
```

```
      <precip total_dia="0" tipo="lluvia"
        fuerza="ligera"/>
```

```
    </reporte>
```

```
  </ciudad></clima>
```

```
</XML>
```

```
</head>
```

```
<body>
```

```
...
```

# Ejemplos de islas XML.

```
<html>
```

```
<head>
```

```
  <title>Islas XML</title>
```

**Isla XML**

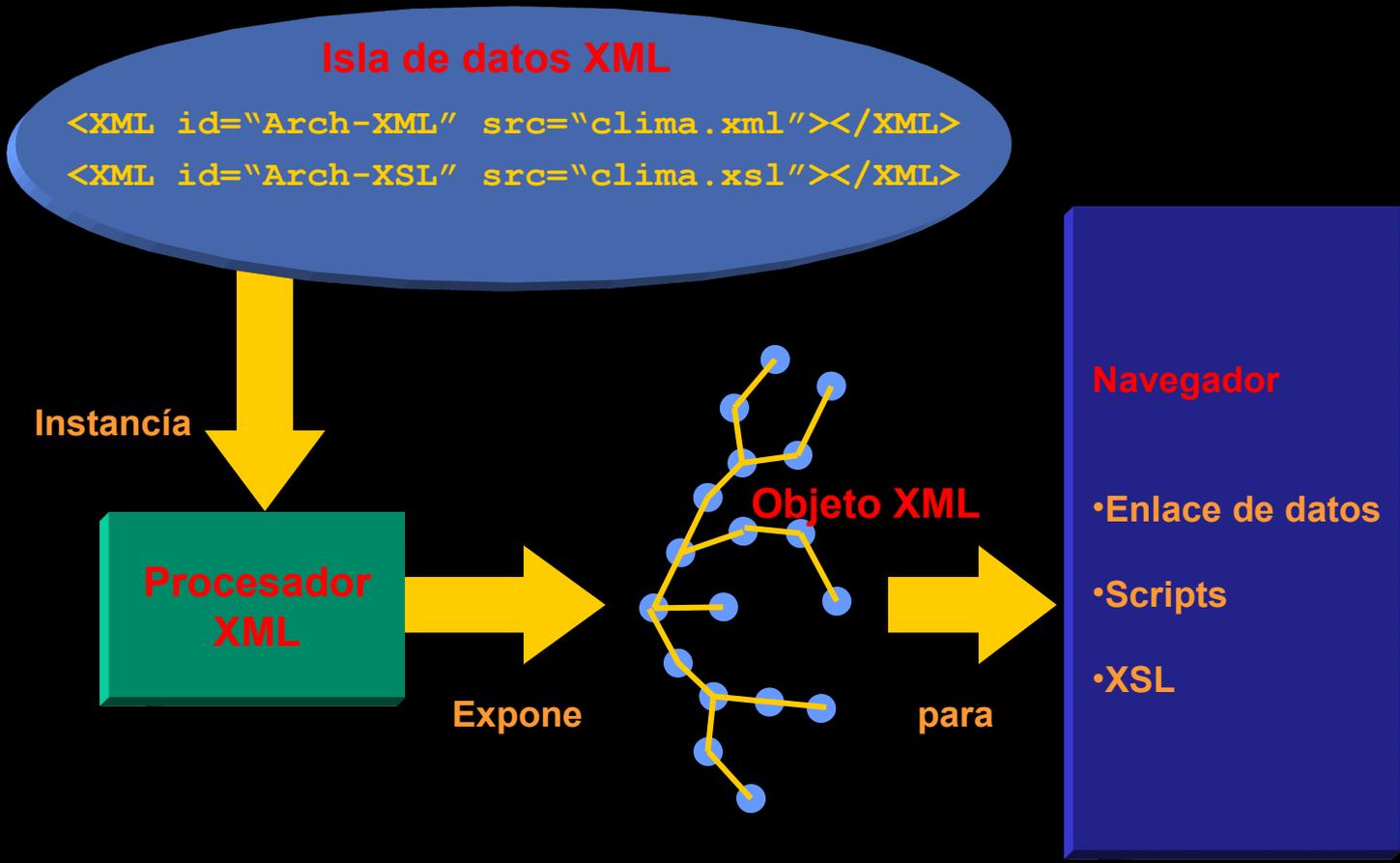
```
  <XML id="Arch-XML" src="clima.xml"></XML>
```

```
  <XML id="Arch-XSL" src="clima.xsl"></XML>
```

```
</head>
```

```
<body> ...
```

# XML en el DOM.



**Demo**

Otras "Xs".

# XHTML.

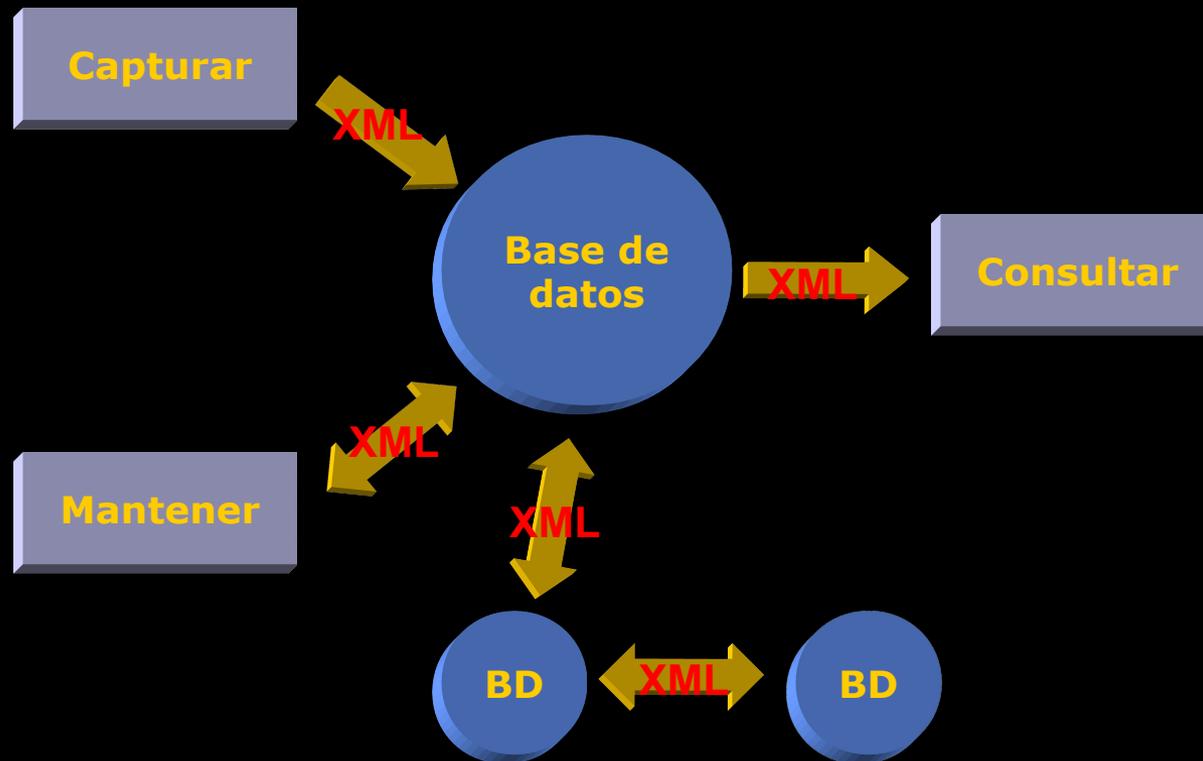
- Es simplemente la representación del HTML 4 en forma de XML.
- Los documentos en XHTML son realmente XML.
- Funcionan con los navegadores que cumplen con el estándar HTML 4.
- Es la siguiente fase en la evolución de las páginas de Web.

# XSLT.

- Transformaciones XSL.
- Son un tipo de XSLs específicamente para transformar un árbol XML en otro árbol XML.
- Sirven para reacondicionar la estructura de los datos.
- Si una XSL transforma XML en XHTML también es una XSLT.

El XML y las bases de datos.

# ¿Dónde coinciden el XML y las BD?



# XML y las BDs.

- Capturar en XML.
  - » Publicar de una fuente XML hacia la BD.
- Consultar en XML.
  - » Crear salidas desde la BD a un formato de presentación (como el HTML).
- Exportar en XML.
  - » Crear vistas lógicas de la base de datos.
- XML como protocolo entre BDs.
  - » Operaciones entre bases de datos usando XML.

Publicando sitios con XML.

# Publicación Web de XML a HTML.

1. El XML funciona como BLOB persistente en el sistema de archivos.
  - » Requiere asistencia para encontrar cada documento.
1. Almacenar y consultar los documentos XML desde una base de datos.
  - Conversión a HTML usando XSL en el servidor.
  - Enviar al navegador.
- Ventajas:
  - » XML para manejo de documentos.
  - » HTML para máxima compatibilidad con los navegadores.

B2B con XML.

## B2B.

- Las transacciones B2B (Business-to-business) existen desde antes de la Web.
- Anteriormente (antes de 1998) se hacían casi exclusivamente con soluciones EDI (Electronic Data Intechange) propietarias.
- El modelo era un consumidor grande imponiendo su solución a sus proveedores.
- La Web ha facilitado la conversión a mercados electrónicos (e-market) abiertos, con soluciones abiertas.

# XML como lenguaje B2B.

- Las transacciones B2B requieren intercambio de información estructurada.
- Ahora, además, debe ser usando soluciones abiertas y estándares.
- El XML es ideal.
- Requiere estandarizar el árbol de información a intercambiar.

# Perfecto, pero...

- Para intercambiar información se requiere estar de acuerdo en qué árbol o jerarquía de datos se utilizarán.

# La necesidad de guías de uso y estilo.

- Si queremos que todos en nuestro proceso (personas y máquinas) usen las mismas etiquetas de la misma manera, entonces requerimos...
- Guías de uso y estilo para las estructuras de datos.
  - » Listas de etiquetas válidas.
  - » Relaciones entre etiquetas.
  - » Valores predeterminados para los atributos.
  - » Tipos de datos que no sean XML estandarizados.

# Guías de uso y estilo.

- Nada de esto es nuevo.
  - » Las guías de uso y estilo se han usado por décadas para intercambiar información entre personas.
- Pero el XML no es solo para personas.
  - » Se requieren guías legibles por máquinas.

# Reglas: La DTD del XML.

- Una Definición de Tipo de Documento (Document Type Definition, DTD) permite:
  - » Definir un conjunto específico de etiquetas con relaciones específicas.
  - » Definir valores predeterminados para los atributos.
  - » Definir entidades de texto y binarias adicionales junto con sus notaciones.
  - » Indicar el elemento raíz.

# El control de la DTD.

- La DTD proporciona:
  - » Una sintaxis formal que sirva de guía a un intérprete/analizador (*parser*).
  - » La habilidad de definir valores predeterminados para los atributos.
  - » Especificaciones para la estructura.
- Una DTD es una buena manera (pero no la única) de controlar la creación de datos.

# Ejemplo completo (XML+DTD).

```
<?xml version="1.0"?>
<!DOCTYPE clima [
<!--      Nombre      Modelo contenido      -->
<!ELEMENT clima      (ciudad+)              >
<!ELEMENT ciudad     (nombre, reporte)      >
<!ELEMENT nombre     (#PCDATA)             >
<!ELEMENT reporte    (alta, baja, precip?)  >
<!ELEMENT alta       (#PCDATA)             >
<!ELEMENT baja       (#PCDATA)             >
<!ELEMENT precip     EMPTY                  >
<!ATTLIST precip     total_dia      CDATA      #REQUIRED
                    tipo          (lluvia | nieve) "lluvia"
                    fuerza       (ligera | fuerte) #IMPLIED >
]>
```

**DTD incluida con el XML.**

```
<clima>
  <ciudad>
    <nombre>Mexico DF</nombre>
    <reporte>
      <alta>27</alta>
      <baja>18</baja>
      <precip total_dia="0" tipo="lluvia" fuerza="ligera"/>
    </reporte>
  </ciudad>
  <ciudad>
    <nombre>Monterrey</nombre>
    <reporte>
      <alta>42</alta>
      <baja>36</baja>
      <precip total_dia="0" tipo="lluvia" fuerza="fuerte"/>
    </reporte>
  </ciudad>
</clima>
```

**Código XML.**

2 elementos de segundo nivel de ejemplo.

# Alternativas a las DTDs.

- Para datos generados automáticamente:
  - » Scripts.
  - » Programas.
- Para datos generados por personas:
  - » Formularios.
  - » Scripts de conversión.
  - » Editores restringidos.
  - » “Guías de uso y estilo”.
- Esquemas W3C.
  - » Nueva especificación del W3C.

# La iniciativa BizTalk.

- Grupo de compañías que están creando esquemas XML para su propia industria.
- Principalmente esquemas para interacciones B2B.
- Iniciada por Microsoft.
- No son un cuerpo de estandarización oficial.
- Apoyada demasiado en su propio Biztalk Server.
- Los esquemas XML son libres, gratuitos y utilizables con soluciones de otras compañías.

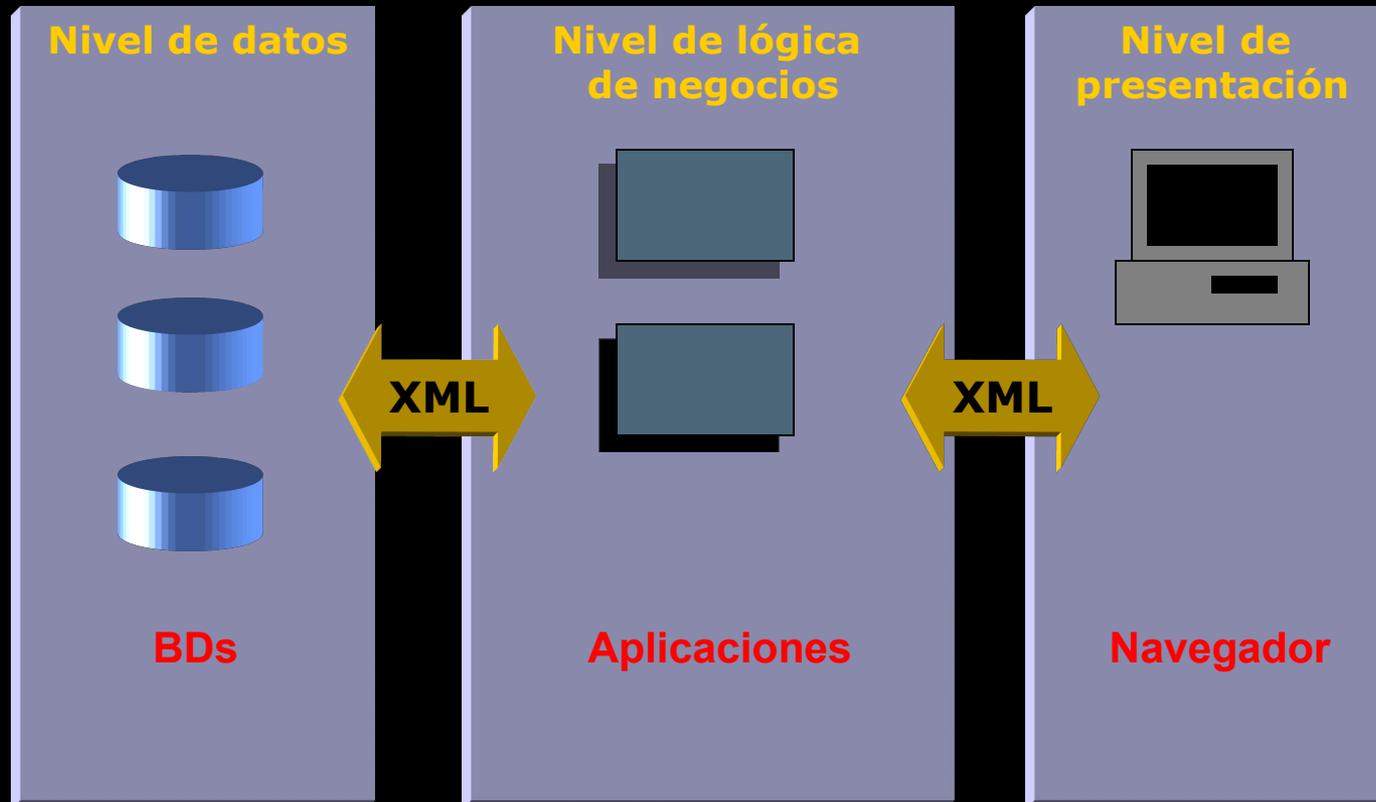
# XML y el software del B2B.

- Los mercados electrónicos requieren software extensible, distribuido, eficiente y con un TOC reducido.
- Las compañías de software se mueven de un modelo de venta a uno de suscripción.
- Parte de la "Webificación" de los negocios (y de la vida como la conocemos).
- Nacimiento del Proveedor de Software de Aplicación (ASP).
- Mercado estimado en >US\$20,000 millones dls. hacia el 2003.

# Aplicaciones Web con XML.

- Modelo de 3 niveles (3-tier).
  - » Nivel de presentación o de usuario.
    - Código para presentación.
  - » Nivel de lógica de negocios.
    - Código donde se ejecutan las decisiones de la aplicación, se aplican políticas y casi toda la lógica que rige a la aplicación.
  - » Nivel de datos.
    - Código para acceder y transformar básicamente el contenido de bases de datos.
- Mejor escalabilidad y flexibilidad.

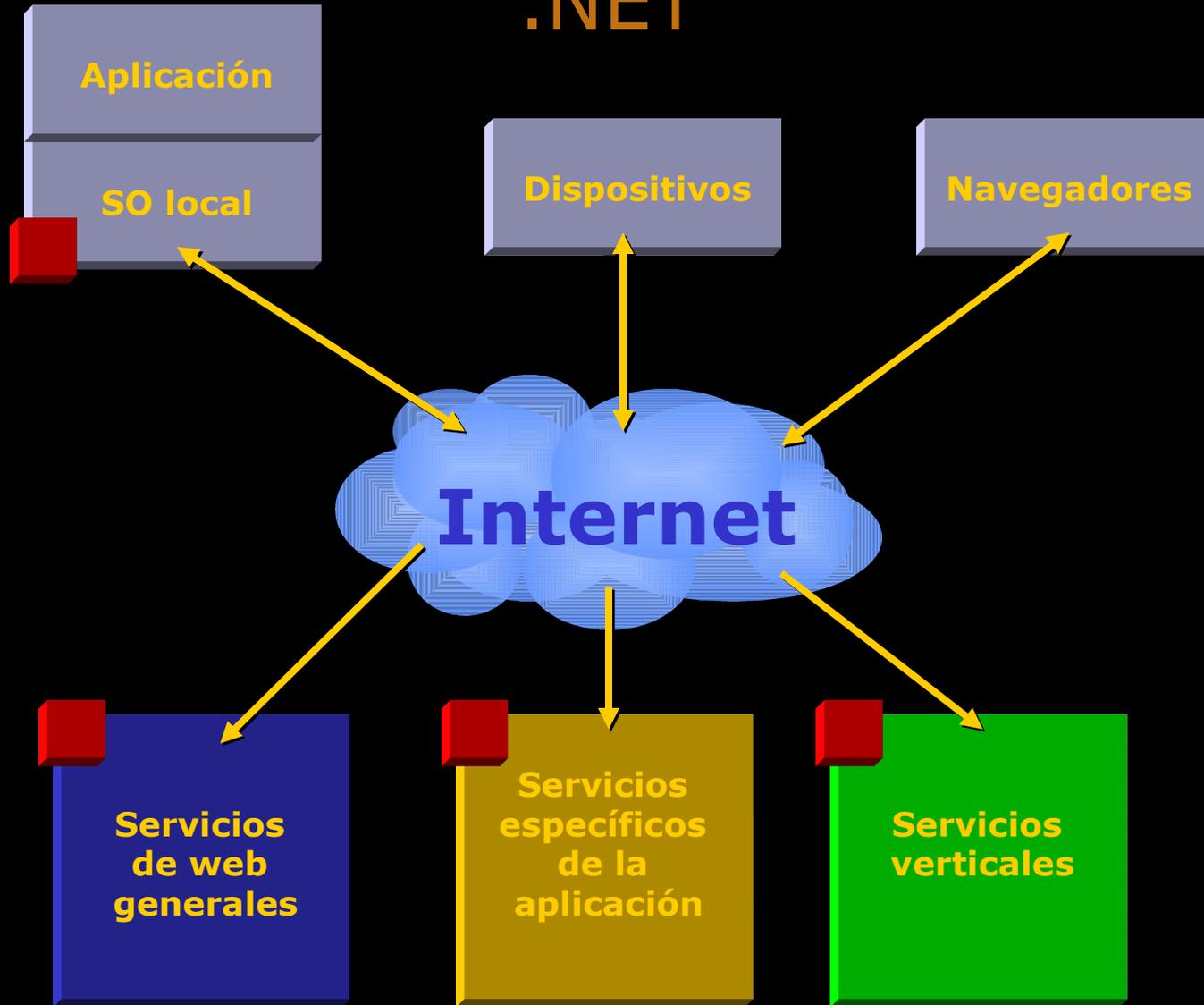
# XML y el modelo de 3 niveles.



# .NET

- Es el modelo de n-niveles, para Internet, en versión Microsoft.
- Las funciones del software son servicios de Web que se rentan, no se venden.
- Lo nuevo no es el concepto sino la integración en sus productos y herramientas.
- El XML es el transporte de datos entre los diferentes segmentos del software.

# .NET



# .NET

- ¿El futuro del software?
- Problemas para penetrar el mercado latinoamericano y europeo.
- Problemas de seguridad.
- Problemas de desempeño.
- Resistencia al cambio de modelo.
- ...etc., etc., etc.

# XML está en todas partes.

- Todas las “suites” importantes graban sus archivos en XML (o formatos mixtos).
- Muchos sitios Web están usando XML como formato de almacenamiento.
- Oracle, MS y otras compañías de BDs ya almacenan usando XML.
- Microsoft Latam va a migrar a XML ya.
- Etc.

# La octava maravilla.

- No tienes nada de mágico excepto que es muy flexible.
- Le llaman el “ASCII del futuro”.
- No es una panacea.
- Es sencillo de usar.
- No crean todo lo que oigan.
- Estudien.

# Más información.

- MSDN Latinoamérica,
  - » <http://www.microsoft.com/latam/msdn/>
- MSDN Web Workshop,
  - » <http://msdn.microsoft.com/workshop/>
- W3C,
  - » <http://www.w3.org/xml/>
- Architag,
  - » <http://architag.com/xmlu/>
- XML en DevX,
  - » <http://www.xml-zone.com/>

...para empezar.