

**LIPI.**  
**Xbase, Xinclude**

# XBase

- ▶ HTML permite especificar un URI respecto a la cual se referencia el documento  
`<base href= "... ">`
- ▶ XML tiene el atributo reservado “base”

W3C<sup>®</sup> <http://www.w3.org/TR/xmlbase/>

# Ejemplo Xbase

<elemento xml:base="http://www.khyri.com">

.....

<contenido xml:base="/kim/">

.....

</contenido>

</elemento>



# XInclude

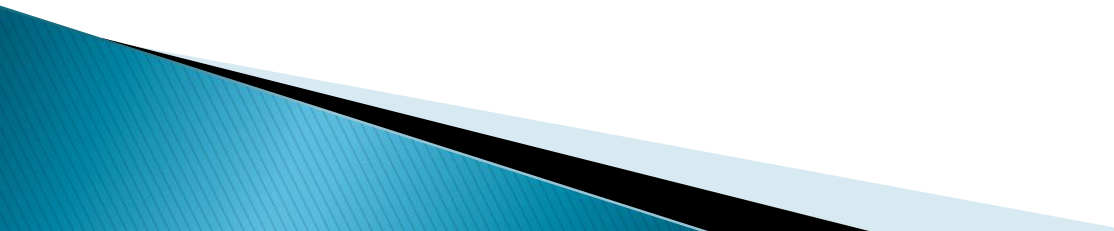
- ▶ Define como un documento puede ser construido con piezas de otros documentos.
- ▶ Similar a lo que podíamos hacer con las entidades externas procesables.
- ▶ La especificación XInclude tiene como Espacio de Nombres <http://www.w3.org/2001/xinclude> (para el que suele usarse el prefijo "xi:") reduciéndose al uso de dos elementos: xi:include y xi:fallback.



<http://www.w3.org/TR/xinclude>

# Xinclude. Sintaxis

```
<xi:include  
  xmlns:xi="http://www.w3.org/2001/XInclude"  
  href="URI "  
  parse="xml | text"  
  xpointer  
  encoding="tipo de codigo"/>
```



# Fallback

- ▶ Proporciona un mecanismo para recuperar recursos perdidos.

# DTD Xinclude

<!ELEMENT xi:include (xi:fallback?)>

<!ATTLIST xi:include

xmlns:xi CDATA #FIXED <http://www.w3.org/2001/XInclude>

href CDATA #IMPLIED

parse (xml|text) "xml"

xpointer CDATA #IMPLIED

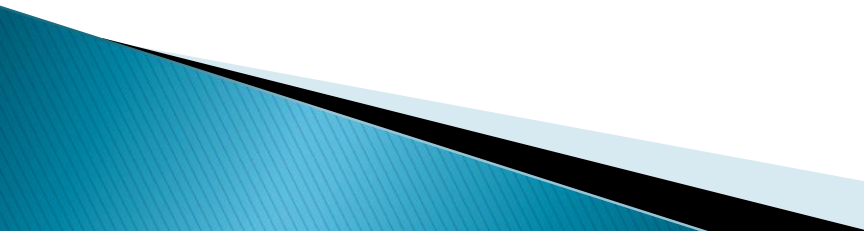
encoding CDATA #IMPLIED

accept CDATA #IMPLIED

accept-language CDATA #IMPLIED >

<!ELEMENT xi:fallback ANY>

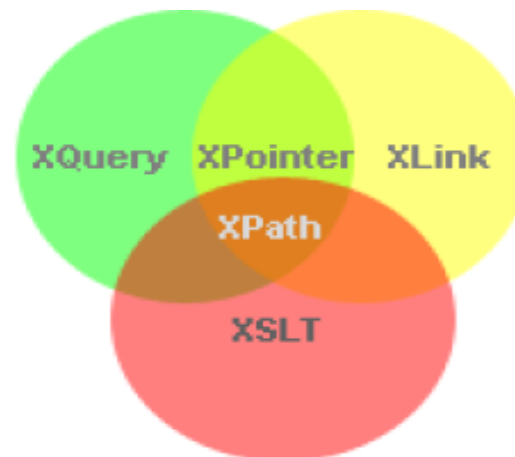
```
<?xml version='1.0'?>
  <div>
    <xi:include href="example.txt" parse="text"
      xmlns:xi="http://www.w3.org/2001/XInclude">
      <xi:fallback>
        <xi:include href="fallback-example.txt" parse="text">
          <xi:fallback>
            <a href="mailto:bob@example.org">
              Report error
            </a>
          </xi:fallback>
        </xi:include>
      </xi:fallback>
    </xi:include>
  </div>
```

A decorative graphic in the bottom-left corner consisting of overlapping blue and black geometric shapes, possibly representing a stylized wave or a modern design element.

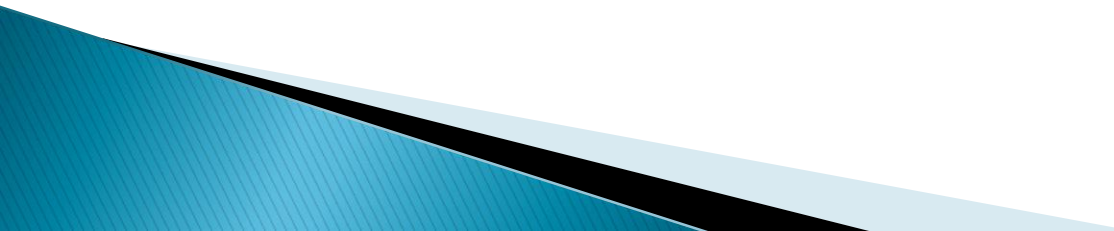


# ¿Que es XLink?

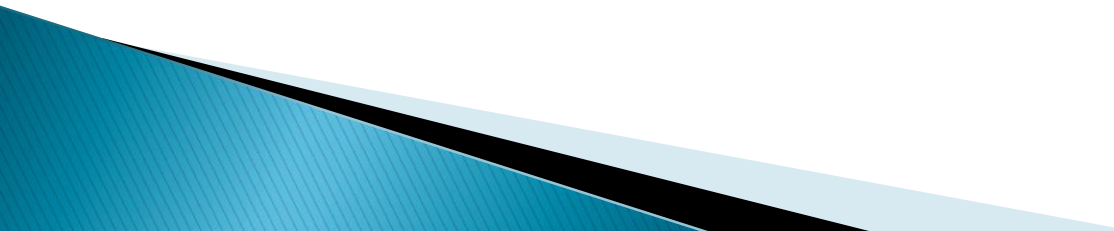
- ▶ XLink (XML Linking Language) es una tecnología que define un método estándar para la representación de enlaces entre recursos.
- ▶ A diferencia de las asociaciones simples, como sería la etiqueta HTML `<a>`, XML implementa un mecanismo para definir los enlaces entre recursos, XPointer describe la manera de dirigir estos recursos y XLink describe la manera en cual pueden ser asociados dos o más recursos.



# Objetivos de XLink

- ▶ El objetivo de la especificación XLink es satisfacer la necesidad de crear vínculos XML que tengan significado y que se puedan usar de maneras complejas.
  - ▶ XLink define un mecanismo para enlazar documentos entre sí.
  - ▶ Permite enlazar 2 o más recursos.
  - ▶ Facilita la inclusión de meta-información en los enlaces.
  - ▶ Se pueden expresar enlaces que residen en documentos externos.
- 

# Ventajas XLink

- ▶ Xlink permite crear vínculos bidireccionales, lo que implica la posibilidad de moverse en dos direcciones. Esto facilita la obtención de información remota como recursos en lugar de simplemente como paginas WEB.
  - ▶ Un XLink permite establecer una relación entre dos o más recursos en la Web, sin que necesariamente estos recursos sepan que están enlazados.
- 

# XLink: Elementos

- ▶ XLink no define elementos, define atributos y la manera como pueden ser utilizados los elementos contenidos.
- ▶ Un elemento de vinculación XLink define las relaciones entre los recursos. El primer paso para habilitar Xlink en un documento XML es definir su espacio de nombre identificado por la URI:

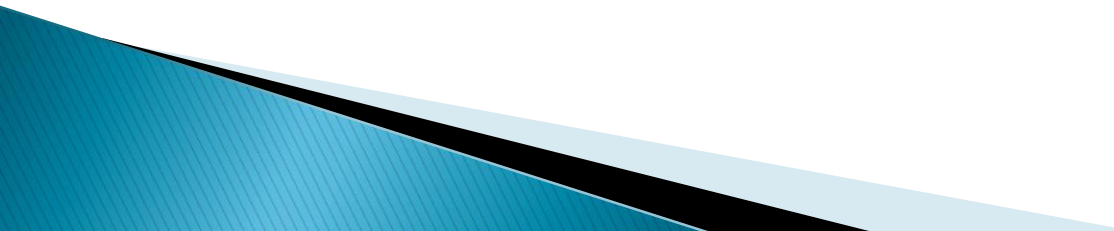
*<http://www.w3c.org/1999/xlink>*

# Tipos de elementos XLink

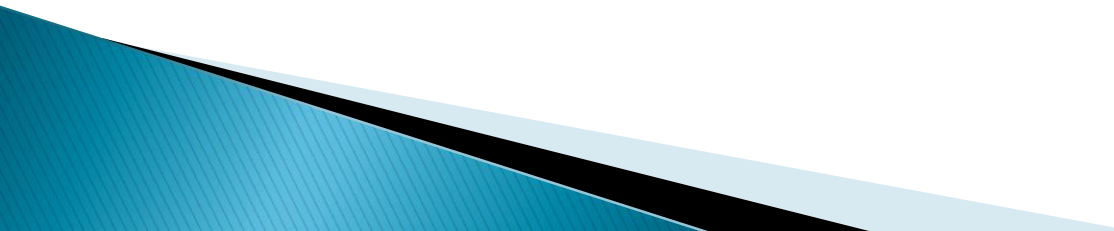
Existen dos tipos de vínculos:

- **Enlaces simples** (xlink: type = "simple") que son más fácil de utilizar que los vínculos extendidos. Puede tener hasta seis atributos adicionales de XLink namespace: href, title, actuate, show, role y arcrole.
- **Enlaces extendidos** (xlink:type="extended") que dan más control sobre sus vínculos. Elementos con el valor de xlink:type de "locator," "arc," "resource" y "title" son subelementos de los vínculos extendidos que dicen al procesador XLink los detalles del determinado vínculo.

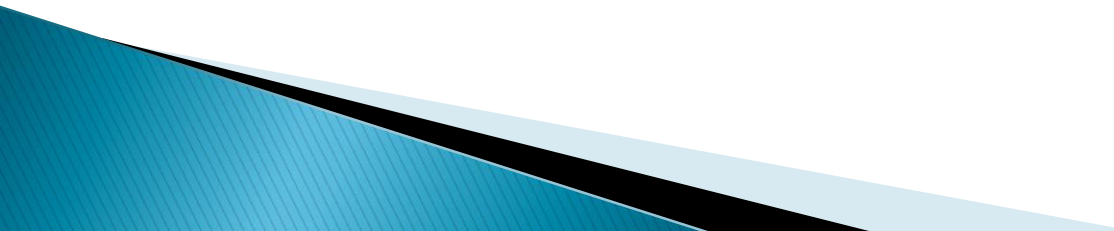
# XLink: Atributos

- type simple, extended, locator, arc, resource, title, none
  - href URI
  - role URI
  - arcrole URI
  - title es un string
  - show new, replace, embed, other, none
  - actuate onload, onrequest, other, none
  - label resource name
  - from resource name
  - to resource name
- 

# XLink: Atributos de Enlaces simples

- ▶ **xlink:type**: tipo del enlace.
  - ▶ **xlink:href**: URL del recurso.
  - ▶ **xlink:title**: título del enlace .
  - ▶ **xlink:role** rol del enlace.
  - ▶ **xlink:actuate**: define el momento de actuación. Puede ser: none, onLoad, onRequest ó other.
  - ▶ **xlink:show**: Define el comportamiento del enlace en el momento de actuar. Puede ser: embed, new, none, other, ó replace.
- 

# Xlink:show

- ▶ [XLink:show](#) define la representación del link
  - embed: poner al contenido del link en la misma ventana
  - new: abrir una nueva ventana
  - replace: sustituir la ventana con el contenido del link
  - none: no hace nada
  - other: especifico por cada aplicacion
- 



# Comportamiento del enlace

Mediante los atributos [show](#) y [actuate](#) se pueden sugerir acciones al software que utilice los enlaces.

```
<imagen xlink:type="simple"  
    xlink:actuate="onLoad" xlink:show="embed"  
    xlink:href="foto.jpg"  
    width="160" height="200" alt="Foto de Pepe"  
/>
```

# Semántica del enlace

Los atributos [title](#) y [role](#) facilitan la inclusión de descripciones sobre los vínculos. El atributo role debe ser una URI que puede representar propiedades de web semántica.

```
<novela xlink:type="simple"
  xlink:href="http://www.libros.com#clarin"
  xlink:title="Leopoldo Alas Clarin"
  xlink:role="http://purl.org/dc/elements/1.1/creator">
  La Regenta
</novela>
```

# Simple-Link. Ejemplo

```
<ingrediente xmlns:xlink="http://www.w3.org/1999/xlink"  
  cantidad="3"  
  unidades="piezas"  
  xlink:type="simple"  
  xlink:href="http://www.mexconnect.com/mex_/recipes/foodindexsp.html"  
  xlink:actuate="onRequest"  
  xlink:show="new"  
  xlink:title="Comprando ingredientes">  
/>
```

La URL con la que enlazar

Para mostrar el recurso tras su solicitud

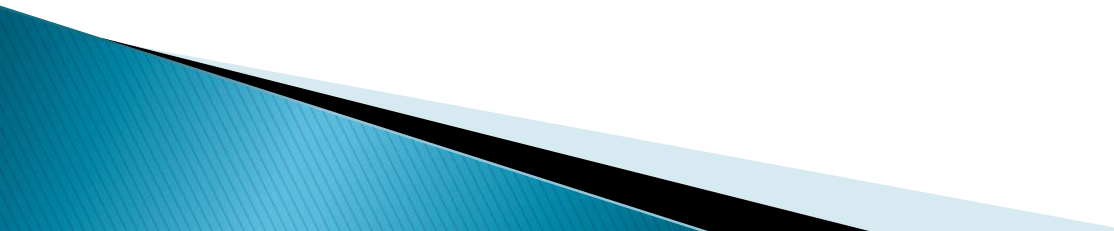
Dice al procesador XLink abrir el vinculo en una nueva ventana

# Enlaces Extendidos

- ▶ Se indican con el valor del atributo: `xlink:type` igual a `extended`.
- ▶ Un enlace extendido puede tener más de dos recursos que enlazar. Dichos recursos se especifican con el elemento “locator”.
- ▶ Pueden contener enlaces de los siguientes tipos:
  - Localizadores (locator): que permiten localizar recursos.
  - Arcos (arc): que definen vínculos entre localizadores
  - Recursos (resource): definen recursos locales.

# Clasificación de arcos

Los arcos según lo que apuntan pueden ser:

- internos – dentro de un documento
  - inbound – de fuera  
from–recurso externo to– interno
  - outbound– a fuera  
from– recurso local to– externo
  - los dos son fuera
- 

# Definición arcos

Los arcos se definen utilizando los atributos xlink:from y xlink:to.

Ejemplo:

```
<siguiente xlink:type="arc" xlink:from="cd2" xlink:to="cd3">
```

```
<anterior xlink:type="arc" xlink:from="cd2" xlink:to="cd1">
```

Los valores de los atributos from-to corresponden al atributo xlink:label. Varios xlink elementos pueden tener el mismo label.

# Extended-Link

## Ejemplo 1

```
<novelas xlink:type="extended" xmlns:xlink="...">
  <novela xlink:type="locator" xlink:label="regenta"
    xlink:href="http://www.libros/regenta.html" />
  <novela xlink:type="locator" xlink:label="quijote"
    xlink:href="http://www.libros/quijote.html" />
  <siguiente xlink:type="arc"
    xlink:from="regenta" xlink:to="quijote" />
  <anterior xlink:type="arc"
    xlink:from="quijote" xlink:to="regenta" />
</novelas>
```

# Trabajo propuesto

Dado el siguiente fichero xml hacéis una transformación XSL para mostrar la utilización de vínculos simples.

```
<receta xmlns:xlink="http://www.w3.org/1999/xlink">  
  <autor xlink:href="http://www.guisando.org"  
    xlink:type="simple">Anita</autor>  
  <ingredientes>  
    <ingrediente xlink:href="http://www.guisando.org/torta-morena"  
      xlink:type="simple">harina</ingrediente>  
    <ingrediente xlink:href="http://www.guisando.org/torta-morena"  
      xlink:type="simple">azúcar</ingrediente>  
  </ingredientes>  
</receta>
```



# XLINK + XBASE

```
<?xml version="1.0"?>
<doc xml:base="http://example.org/today/"
     xmlns:xlink="http://www.w3.org/1999/xlink">
  <head>
    <title>Virtual Library</title>
  </head>
  <body>
    <paragraph>See <link xlink:type="simple" xlink:href="new.xml">what's
      new</link>!</paragraph>
    <paragraph>Check out the hot picks of the day!</paragraph>
    <olist xml:base="/hotpicks/">
      <item>
        <link xlink:type="simple" xlink:href="pick1.xml">Hot Pick #1</link>
      </item>
      <item>
        <link xlink:type="simple" xlink:href="pick2.xml">Hot Pick #2</link>
      </item>
      <item>
        <link xlink:type="simple" xlink:href="pick3.xml">Hot Pick #3</link>
      </item>
    </olist>
  </body>
</doc>
```

# XPointer



# Índice

1. Introducción
2. Conceptos
3. Árbol XPointer
4. Funciones
5. Ejercicios



# 1. Introducción

**XPointer** o Lenguaje de punteros XML es un estándar del W3C que permite identificar de forma única fragmentos de un documento XML.

**XPointer** ofrece un mecanismo para direccionamiento de documentos XML en función de su estructura interna. Permite direccionar elementos, valores de atributos, contenido de caracteres y posiciones relativas.

**XPointer** es una extensión de XPath.

**XPointer** permite a XPath:

- Seleccionar puntos, intervalos y nodos.
- Utilizar coincidencias de cadenas para buscar información.
- Utilizar expresiones de direccionamiento en referencias de URI como identificadores de fragmentos.

<http://www.w3.org/TR/xpath>

**Versiones anteriores:**

<http://www.w3.org/TR/1999/PR-xpath-19991008>

<http://www.w3.org/1999/08/WD-xpath-19990813>

<http://www.w3.org/1999/07/WD-xpath-19990709>

<http://www.w3.org/TR/1999/WD-xslt-19990421>

**Editores:**

James Clark <[jjc@jclark.com](mailto:jjc@jclark.com)>

Steve DeRose (Inso Corp. y Brown University) <[Steven\\_DeRose@Brown.edu](mailto:Steven_DeRose@Brown.edu)>

Copyright © 1999 W3C® (MIT, INRIA, Keio), Todos los Derechos Reservados. Son aplicables las reglas del W3C sobre [obligaciones](#), [marcas registradas](#), [utilización de documentos](#) y [licencias de software](#).

---

## Resumen

XPath es un lenguaje para direccionar partes de un documento XML, diseñado para ser utilizado tanto por XSL como por XPointer.

## Status de este documento

Este documento ha sido examinado por los miembros del W3C y otras partes interesadas y ha sido respaldado por el Director como una [Recomendación](#) del W3C. Es un documento estable y puede ser usado como material de referencia o citado como

**Latest version:**

<http://www.w3.org/TR/xptr-xpointer/>

**Previous version:**

<http://www.w3.org/TR/2002/WD-xptr-xpointer-20020710/>

**Editors:**

Steven DeRose, Brown University Scholarly Technology Group, Bible Technologies Group [<sderose@acm.org>](mailto:sderose@acm.org)

Eve Maler, Sun Microsystems [<eve.maler@sun.com>](mailto:eve.maler@sun.com)

Ron Daniel Jr., Taxonomy Strategies [<rdaniel@taxonomystrategies.com>](mailto:rdaniel@taxonomystrategies.com)

This document is also available in these non-normative formats: [XML](#), [\(its DTD\)](#), and [\(an XSL stylesheet for it\)](#).

[Copyright](#) © 2002 [W3C](#)® ([MIT](#), [INRIA](#), [Keio](#)), All Rights Reserved. W3C [liability](#), [trademark](#), [document use](#), and [software licensing](#) rules apply.

---

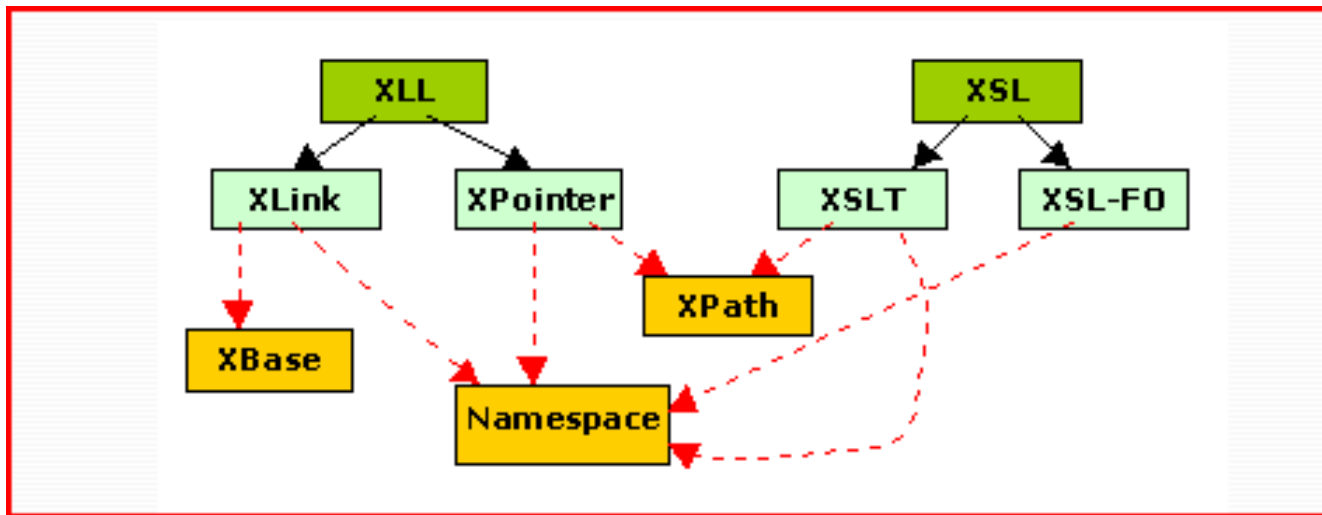
## Abstract

The XPointer `xpointer()` scheme is intended to be used with the XPointer Framework [\[XPtrFrame\]](#) to provide a high level of functionality for addressing portions of XML documents. It is based on XPath [\[XPath\]](#), and adds the ability to address strings, points, and ranges in accordance with definitions provided in DOM 2: Range. [\[DOM2\]](#)

## Status of this Document

This is a W3C Working Draft for review by W3C members and other interested parties. It is a draft document and may be updated,

- Evolución de la especificación XSL y XLL (enlaces entre documentos XML) y sus relaciones con otras especificaciones





## 2. Conceptos

- **Xpath**: Es un lenguaje que permite construir expresiones que recorren y procesan un documento XML
- **Point** (Punto): Es una localización en un conjunto de datos XML sin contenido o hijos.
- **Range** (Rango): Es una identificación de todo el contenido XML entre un par de puntos



## 2. Conceptos

- ▶ dada a una URIref como:  
<http://www.sitio.com/doc.xml>
- ▶ Se puede añadir un indicador de fragmento de la forma:  
#xpointer (XXX) , donde XXX sea una o varias expresiones XPath

Ejemplo:

....doc.xml#xpointer(/libro/capitulo[@public])xpointer(/libro/capitulo[@num="2"])

<Factura>

<Nombre>Juan Luna</Nombre>

<Direccion1>Cualquier Calle1 </Direccion1>

<Direccion2>Alguna localidad </Direccion2>

<Ciudad>Cualquier Ciudad</Ciudad>

<CodigoPostal>12345</CodigoPostal>

<Pais>España</Pais>

</Factura>

### **Puntos:**

xpointer(/Capitulo/Parrafo[2]/text())[0]).

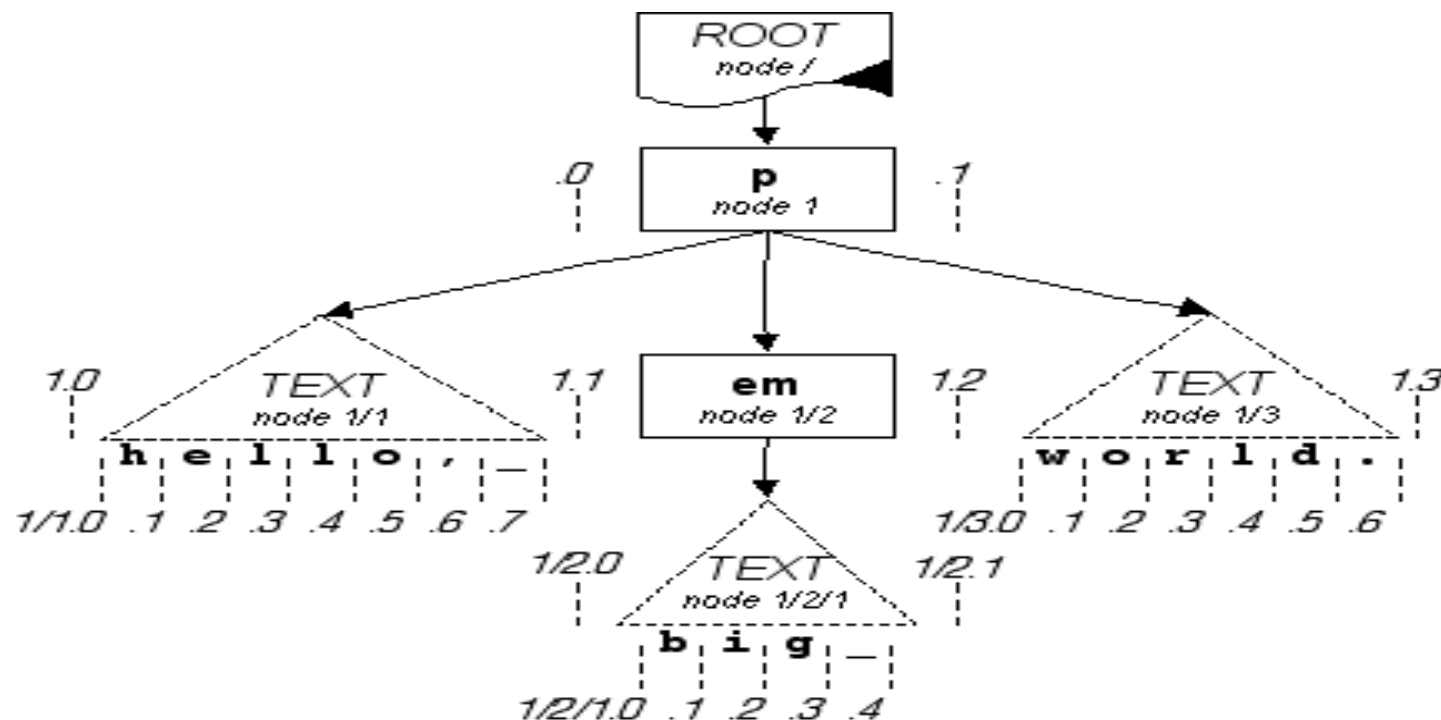
xpointer(/Capitulo/ Parrafo[2]/text())[8]).

### **Rango:**

xpointer(/Capitulo/Parrafo[2]/text())[0] to /Capitulo/Parrafo[2]/text())[8])



### 3. Árbol Xpointer (I)

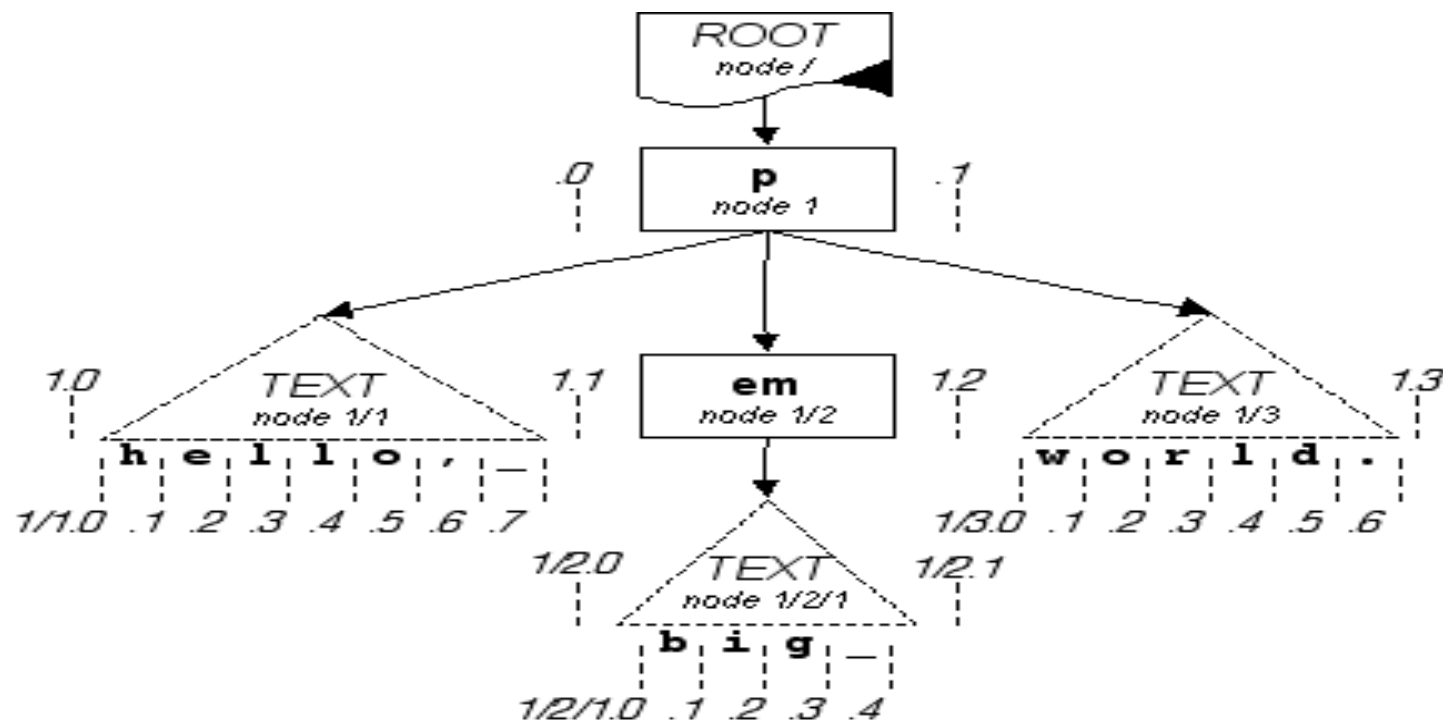


Ejemplo: **point(.0)** precede inmediatamente al nodo root

Ejemplo: **point(1.2)** está entre el final del elemento 'em' y el nodo de texto que contiene 'world.'



### 3. Árbol Xpointer (II)



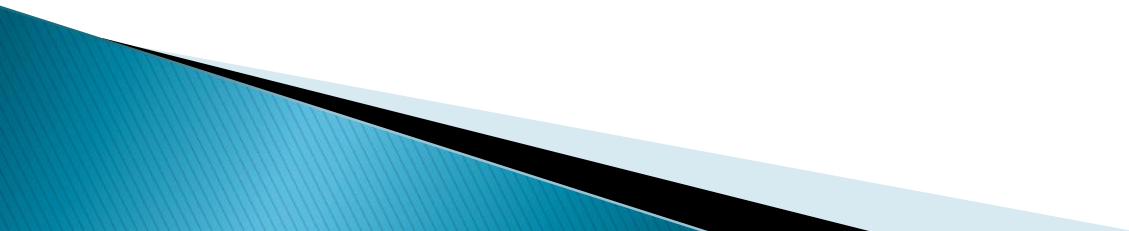
Ejemplo: **point(1/2/1.1)** Inmediatamente después de la 'b' en el medio del nodo texto ('big').

# ¿Cuántos puntos hay??

<documento>

<saludo> ¡Hola! </saludo>

</documento>





## 4. Funciones (I)

Funciones clasificadas según lo que devuelven:

- Nodos
  - **Id()**
  - **Here()**
  - **Origin()**
- Puntos
  - **Start-Point()**
  - **End-Point()**
- Rangos
  - **Range()**
  - **Range-Inside()**
  - **Range-To()**
- Rangos String (String matching)
  - **String-Range()**



## 4. Funciones (II)

### Id()

Sirve para seleccionar el elemento con el id especificado.  
Por ejemplo:

Xpointer(id("p12"))

<CITY>

<PERSON ID="p12">

<NAME>A NAME</NAME>

</PERSON>

....

<CITY><PERSON><ID>p12</ID>...</PERSON>...

Equivale a:

- XPointer("p12")
- XPointer(//\*[ @ID="p12"])



## 4. Funciones (III)

### **Start-Point()**

Devuelve el punto inicial de un rango.

Ejemplo:

```
Xpointer(start-point(string-range (id("p1")/NAME, "Laura")))
```

### **End-Point()**

Devuelve el punto inicial de un rango.

Ejemplo:

```
Xpointer(end-point(string-range (id("p1")/NAME, "Laura")))
```





## 4. Funciones (IV)

### Range()

Devuelve un rango.

Ejemplo:

```
xpointer(range(//NAME))
```

```
<PERSON ID="p12">
```

```
<NAME>A NAME</NAME>
```

```
</PERSON>
```

### Range-Inside()

Devuelve un rango, excepto los elementos de tipo nodo.

Ejemplo:

```
xpointer(range-inside(//NAME))
```

```
<PERSON ID="p12">
```

```
<NAME>A NAME</NAME>
```

```
</PERSON>
```



## 4. Funciones (V)

### Range-to()

Devuelve un rango entre dos puntos.

Ejemplo:

```
xpointer (/NAME/range-to(EDAD))
```

```
<PERSON>
```

```
<NAME>A Name</NAME>
```

```
<EDAD>19</EDAD>
```

```
</PERSON>
```

```
xpointer (/NAME/range-to(/NAME/text()))
```

```
<PERSON>
```

```
<NAME>A Name</NAME>
```

```
<EDAD>19</EDAD>
```

```
</PERSON>
```



## 4. Funciones (VI)

### String-Range()

-String-range (expresión, cadena, desplazamiento?, longitud?):  
devuelve un rango de strings que coincidan con la cadena (patrón).  
Es posible indicar un desplazamiento y una longitud de las cadenas resultantes.

Ejemplo: xpointer(string-range(//NAME, "Laura",6,4))

```
<PERSON>
  <NAME>Laura Oran</NAME>
  <EDAD>XX</EDAD>
</PERSON>
```

xpointer(string-range(//NAME,"Laura"))

```
<PERSON>
  <NAME>Laura Oran</NAME>
  <NAME>Laura Dos</NAME>
</PERSON>
```



## 5. Ejemplo de uso

- Se puede utilizar conjuntamente con XLink para referenciar a partes de un documento XML:
  - Encontrar el primer elemento *name* en el documento:  
<http://www.ibiblio.org/xml/people.xml>
  - Solución:  
[http://www.ibiblio.org/xml/people.xml#xpointer\(//name\[position\(\)=1\]\)](http://www.ibiblio.org/xml/people.xml#xpointer(//name[position()=1]))



## 5. Ejercicios (I)

1. Identifica todos los puntos del siguiente XML igual que en la transparencia 5:

```
<PERSONA>  
  <NAME>Eugenia</NAME>  
  <EDAD>38</EDAD>  
</PERSONA>
```

2. A partir del siguiente XML escribe la expresión XPointer que devolvería el rango seleccionado:

```
<CIUDAD ID="VALENCIA">  
  <PERSONA ID="a1">  
    <NOMBRE>Javier Beteta</NOMBRE>  
  </PERSONA>  
  <PERSONA ID="a2">  
    <NOMBRE>Javier Dos</NOMBRE>  
  </PERSONA>  
</CIUDAD>
```



## 5. Ejercicios (II)

3. A partir del siguiente XML escribe la expresión XPointer que devolvería el rango seleccionado:

```
<CIUDAD ID="VALENCIA">  
  <PERSONA ID="a1">  
    <NOMBRE>Javier Beteta</NOMBRE>  
  </PERSONA>  
  <PERSONA ID="a2">  
    <NOMBRE>Javier Dos</NOMBRE>  
  </PERSONA>  
</CIUDAD>
```



## 5. Ejercicios (III)

4. A partir del siguiente XML escribe la expresión XPointer que devolvería el rango seleccionado:

```
<CIUDAD ID="VALENCIA">  
  <PERSONA ID="a1">  
    <NOMBRE>Javier Beteta</NOMBRE>  
  </PERSONA>  
  <PERSONA ID="a2">  
    <NOMBRE>Javier Dos</NOMBRE>  
  </PERSONA>  
</CIUDAD>
```



## 5. Ejercicios (IV)

5. A partir del siguiente XML escribe la expresión XPointer que devolvería el punto justo anterior al primer nodo NOMBRE:

```
<CIUDAD ID="VALENCIA">  
  <PERSONA ID="a1">  
    <NOMBRE>Javier Beteta</NOMBRE>  
  </PERSONA>  
  <PERSONA ID="a2">  
    <NOMBRE>Javier Dos</NOMBRE>  
  </PERSONA>  
</CIUDAD>
```