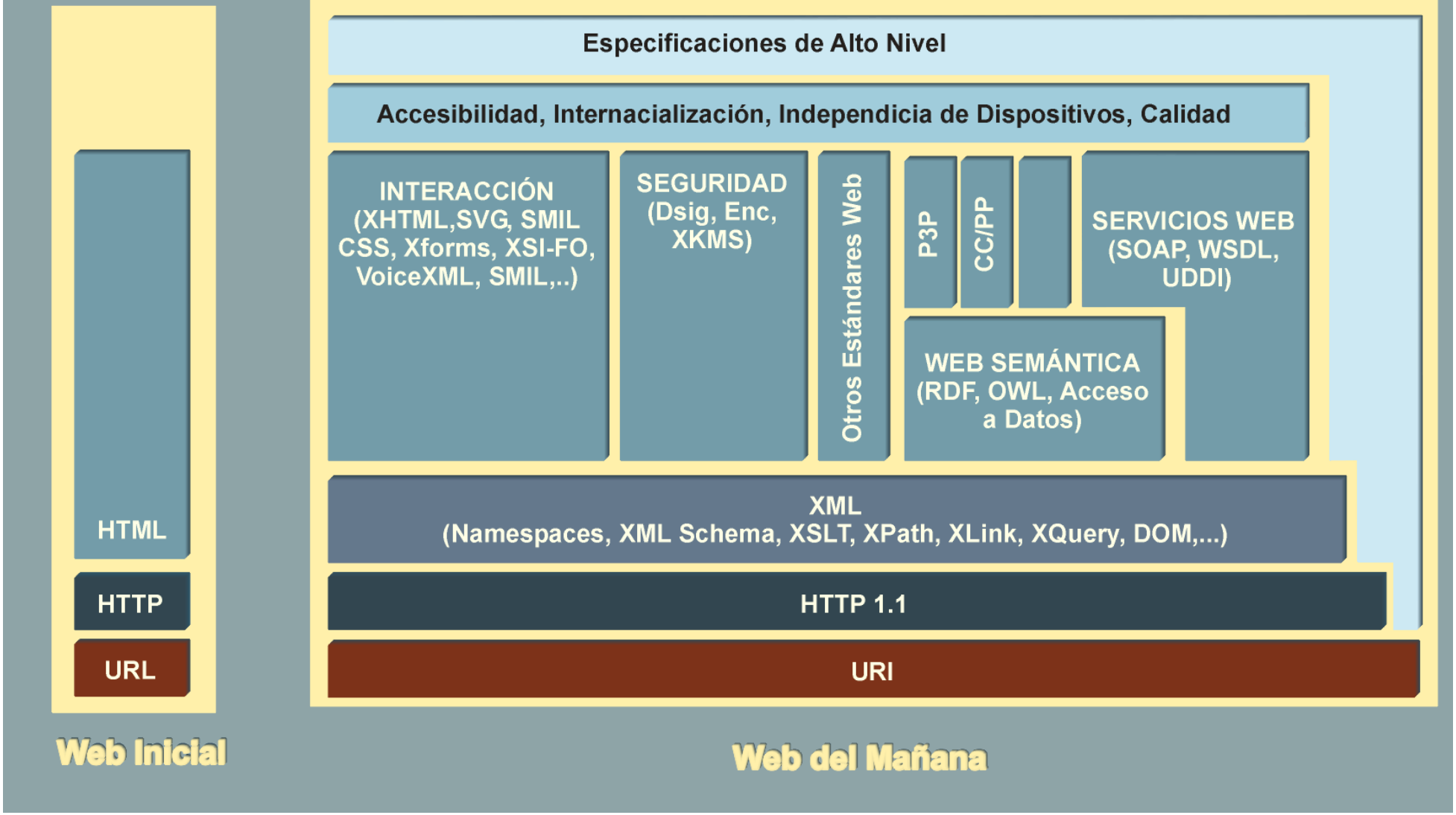
Tecnologías XML

- Web – Futuro
- XPath
- XSL/XSLT

Web - Futuro


Especificaciones de Alto Nivel
Accesibilidad, Internacialización, Independicia de Dispositivos, Calidad
INTERACCIÓN (XHTML,SVG, SMIL CSS, Xforms, XSI-FO, VoiceXML, SMIL,..)
SEGURIDAD (Dsig, Enc, XKMS)
Otros Estándares Web
P3P
CC/PP
SERVICIOS WEB (SOAP, WSDL, UDDI)
WEB SEMÁNTICA (RDF, OWL, Acceso a Datos)
XML (Namespaces, XML Schema, XSLT, XPath, XLink, XQuery, DOM,...)
HTTP 1.1
URI
HTML
HTTP
URL
Web Inicial
Web del Mañana

La Familia XML en la Web

- **Familia XML**: conjunto de tecnologías y lenguajes que permiten tanto la descripción estructurada de cualquier tipo de datos como la manipulación de documentos.
- Se encuentra formada por el conjunto de recomendaciones que proporcionan la base para soportar el uso de XML en la arquitectura de la Web.
- Algunas tecnologías XML:
  - **XPath** (**Lenguaje de Rutas XML**) utilizado para acceder a partes de un documento.
  - **XQuery** (**Lenguaje de Consulta XML**) facilita la extracción de datos/información desde documentos XML.
  - **XPointer** (**Lenguaje de Direccionamiento XML**) permite el acceso a la estructura interna de un documento XML (elementos, atributos y contenido).
  - **XLink (Lenguaje de Enlace XML**) permite establecer vínculos entre recursos XML.
  - **XSL** (**Lenguaje Extensible de Hojas de Estilo XML**) describe cómo deberían estar estructurados los contenidos de un documento, cómo deberían ser presentados y cómo deberían ser paginados.

XML Path Language: XPath

- Introducción
- Modelo de datos
- Expresiones XPath
- API Java

Bibliografía

- Yee Khun, F.; XSLT Working with XML and HTML. Addison Wesley.
- XSLT. Second Edition, Doug Tidwell O'Reilly. 2008.
- Ultima versión XPATH: https://www.w3.org/TR/xpath/
- Evaluador expresiones XPATH: https://www.freeformatter.com/xpath-tester.html
- Tutorial XPATH: https://www.mclibre.org/consultar/xml/lecciones/xml-xpath.html

Recomendaciones

- XPATH es un lenguaje de expresiones que permite seleccionar nodos de un documento XML.
- A partir del contenido de los nodos seleccionados se realizará la algorítmica del programa.
- Evolución de las recomendaciones XPATH del W3C:
  - XML Path Language (XPath) 1.0. Noviembre 1999.
  - XML Path Language (XPath) 2.0. Enero 2007.
  - XML Path Language (XPath) 2.0 (2º edición). Diciembre 2010.
  - XML Path Language (XPath) 3.0. Abril 2014.
  - XML Path Language (XPath) 3.1. Marzo 2017.

XPath – Modelo de Datos

- XPath construye internamente un árbol de nodos llamado **Árbol XPath**. A partir de la raíz, el árbol se diversifica a lo largo de los elementos hasta las hojas.
- El **valor de cadena** de cada nodo depende de su tipo, siendo ocasiones parte del nodo, y calculándose otras veces a partir del valor de cadena de los nodos descendientes.
- El **orden del documento** ordena los nodos elemento en el orden de aparición de las etiquetas XML. Los espacios de nombre y los atributos de un elemento aparecen antes que sus descendientes, y en este orden.
- El **orden inverso de documento** es el inverso del orden del documento.
- Los nodos raíz y los nodos elemento tienen una lista ordenada de **nodos hijo**.
- Todos los nodos excepto el raíz, tienen un único **padre** que es el nodo raíz o un nodo elemento.
- Los **descendientes** de un nodo son sus hijos y los descendientes de sus hijos.
- Los **ascendientes** de un nodo son su padre y los ascendientes de su padre.

XPath – Modelo de Datos

Tipos de Nodo

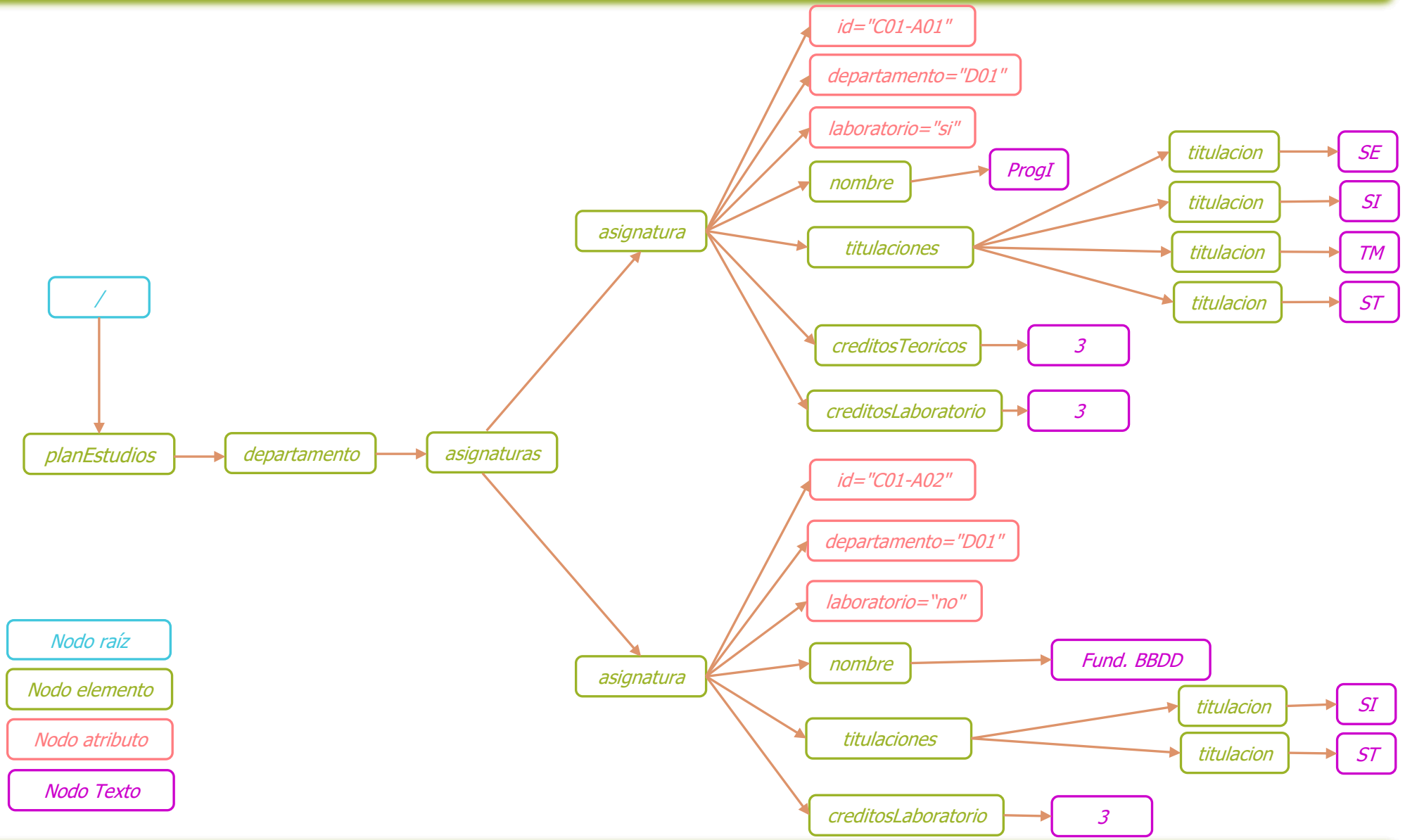
- **Raíz**: Contiene al elemento *root*, es decir, a todo el documento {/}.
  - **Valor de cadena**: concatenación del valor de cadena de todos los nodos descendientes en orden del documento.
- **Elemento**: Contiene un elemento del documento {*}.
  - **Valor de cadena**: concatenación del valor de cadena de todos los nodos descendientes en orden del documento.
- **Atributo**: Contiene el atributo de un elemento {@}.
  - **Valor de cadena**: valor del atributo.
- **Texto**: Contiene el texto de un elemento {text()}.
  - **Valor de cadena**: agrupación de los datos de carácter de un elemento.
- **Espacio de Nombres**: {namespace()}.
  - **Valor de cadena**: URI del espacio de nombres que se está asociando al prefijo.
- **Comentario**: Contiene un comentario {comment()}.
- **Instrucción de Procesamiento**: {processing-instruction()}.

Ejemplo

Documento XML

```
<?xml version="1.0" encoding="windows-1252"?>
<PlanEstudios>
        <departamento>
                <asignaturas>
                        <asignatura id="C01-A01" departamento="D01" tipo="T" laboratorio="si">
                                <nombre>Programación I</nombre>
                                <titulaciones>
                                        <titulacion> SE</titulacion>
                                        <titulacion> SI"</titulacion>
                                        <titulacion> TE </titulacion>
                                        <titulacion> ST<t/itulacion>
                                </titulaciones>
                                <creditosTeoricos>3</creditosTeoricos>
                                <creditosLaboratorio>3</creditosLaboratorio>
                        </asignatura>
                        <asignatura id="C01-A02" departamento="D01" tipo="L" laboratorio="no">
                                <nombre>Fundamentos de las Bases de Datos</nombre>
                                <titulaciones>
                                        <titulacion> SI"</titulacion>
                                        <titulacion> ST<t/itulacion>
                                </titulaciones>
                                <creditosTeoricos>3</creditosTeoricos>
                        </asignatura>
                </asignaturas>
        </departamento>
</PlanEstudios>
```

Árbol XPATH


/
planEstudios
departamento
asignaturas
asignatura
id="C01-A01"
departamento="D01"
laboratorio="si"
nombre
ProgI
titulaciones
titulacion
SE
titulacion
SI
titulacion
TM
titulacion
ST
creditosTeoricos
3
creditosLaboratorio
3
asignatura
id="C01-A02"
departamento="D01"
laboratorio="no"
nombre
Fund. BBDD
titulaciones
titulacion
SI
titulacion
ST
creditosLaboratorio
3
Nodo raíz
Nodo elemento
Nodo atributo
Nodo Texto

XPath – Rutas de Localización

- Una **Ruta de localización** (*location path*) es una expresión que describe la ubicación de un nodo en un árbol XPath.

- Las rutas de localización se encuentran formados por una secuencia de uno o más **pasos de localización**, separados por **/**, que se evalúan de izquierda a derecha.

  [/]paso1**/**paso2**/**paso3**/**….**/**paso*n*

- **Tipos de Rutas de Localización**:
  - **Rutas Absolutas**: señalan una ubicación específica respecto a la raíz del árbol **/**.
  - **Rutas Relativas**: operan a partir de un *Nodo de Contexto*, establecido previamente mediante un camino de localización absoluto.

- **Sintaxis**: sintaxis **no abreviada** y **sintaxis abreviada**.

XPath – Rutas de Localización

- **Elementos de un Paso de Localización**:

Eje :: Nodo [Predicado]

- **Eje**: especifica la relación jerárquica entre los nodos seleccionados por el paso de localización y el nodo de contexto.
- **Nodo de comprobación**: especifica el tipo de los nodos seleccionados por el paso de localización.
- **Predicado**: una o más expresiones que permiten refinar el conjunto de nodos seleccionados.

XPath – Rutas de Localización – Ejes –

Eje

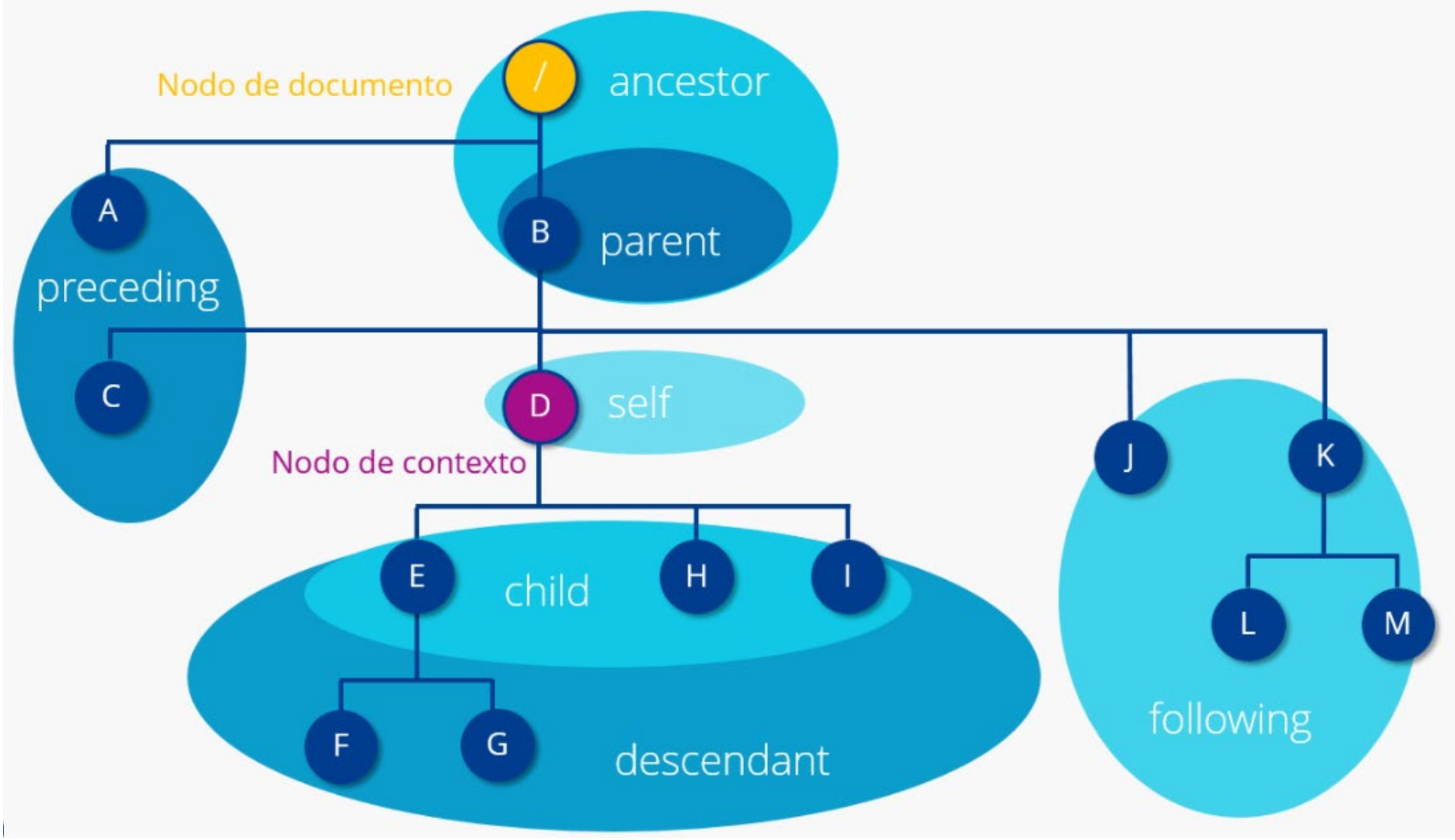
Los Ejes (**axes**) son los elementos encargados de especificar la relación existente entre los distintos nodos que quieren seleccionarse en un camino de localización, y el nodo de contexto de donde se parte.

- **ancestor**: Contiene los nodos elemento que contienen al nodo de contexto, hasta el *root*. El *ancestor* de *root* es un conjunto de nodos vacío.
- **ancestor-or-self**: Igual que *ancestor* con el añadido del nodo de contexto. El nodo *root* siempre está incluido.
- **attribute**: Contiene los nodos atributo del nodo de contexto {@}.
- **child**: Contiene los nodos secundarios del nodo de contexto { }. Los nodos atributo y los espacios de nombre no están incluidos.
- **descendant**: Contiene los nodos de contenido del nodo de contexto (hijos, nietos, …). Los nodos atributos y espacios de nombre se consideran descendientes de los nodos elemento.
- **descendant-or-self**: Igual que *descendant* con el añadido del nodo de contexto{//}.

XPath – Rutas de Localización – Ejes –

- **following**: Contiene todos aquellos elementos del documento que aparecen después del nodo de contexto. Los descendientes del nodo de contexto no están incluidos.
- **following-sibling**: Contiene los nodos posteriores (que aparecen después en el documento) al nodo de contexto que tienen su mismo padre.
- **namespace**: Contiene los nodos del espacio de nombres al que pertenece al nodo de contexto.
- **parent**: Contiene el nodo padre del nodo de contexto {..}, si existe un padre.
- **preceding**: Contiene los nodos anteriores al nodo contexto, excepto los nodos ascendientes, atributo y espacio de nombres.
- **preceding-sibling**: Contiene los nodos anteriores (que aparecen antes en el documento) al nodo contexto y que tienen el mismo padre. Este eje se encuentra vacío si el nodo contexto es un nodo atributo o espacio de nombres.
- **self**: Contiene el propio nodo contexto {.}.

XPath – Rutas de Localización – Ejes –



Fuente: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/tutorial-de-xpath/

XPath – Rutas de Localización – Nodos –

Nodos

Los Nodos de Comprobación o Búsqueda (**node-test**) son los encargados de identificar un nodo o un conjunto de nodos dentro de un eje.

- **node()**: devuelve todos los nodos de cualquier tipo.
- *: selecciona los nodos principales de cada trayecto, a excepción de los tipo texto, comentario e instrucción de procesamiento.
- **text()**: devuelve cualquier nodo de tipo texto.

- Todos los elementos *departamento* del documento:
  - /child::planEstudios/child::departamento
  - /descendant::departamento
- Todos los elementos del documento
  - /descendant::node()
- Nodos de tipos texto descendientes de *departamento*:
  - /descendant::departamento/descendant::text()
- Nombre de las asignaturas impartidas por el departamento 'D01':
  - /descendant::asignatura[@departamento=string('D01')]/child::nombre

XPath – Sintaxis abreviada

- **//** => eje **descendant**.
- **.** => eje **self**.
- **..** => eje **parent**.
- **@** => eje **attribute**
- **/** => eje **child**.

- Todos los elementos *departamento* del documento:
  - /planEstudios/departamento
  - //departamento
- Todos los elementos del documento
  - /
- Nodos de tipos texto descendientes de *departamento*:
  - //departamento//text()
- Nombre de las asignaturas impartidas por el departamento 'D01':
  - //asignatura[@departamento=string('D01')]/nombre

Operadores XPath

- Existen cuatro tipos de operadores: **relacionales**, **booleanos**, **matemáticos** y de **expresión**.

Matemáticos	
+	Suma
-	Resta
*	Multiplicación
div	División
idiv	División entera
mod	Resto de la división

Booleanos	
and	Y
or	O
not	NO

Relacionales	
=	Igual
<	Menor
<=	Menor o Igual
>	Mayor
>=	Mayor o Igual
!=	Diferente

Operadores XPath

- Si una expresión XPath aparece en un documento XML, los operadores <,<=, > y >= deben ser *escapados* de acuerdo con las reglas de XML 1.0.

Operadores de Expresión

- **|**: Permite unir varias expresiones XPath.
- ***:** Hace referencia a cualquier nodo.
- **$**: Precede a un nombre de variable.
- **[]**: Permite especificar un predicado en la expresión XPath.

- Nombre de las asignaturas impartidas por el departamento 'D01' o 'D04':
  - //asignatura[@departamento="D01"]/nombre | //asignatura[@departamento="D04"]/nombre
  - descendant::asignatura[@departamento="D01"]/child::nombre | descendant::asignatura[@departamento="D04"]/child::nombre
  - //asignatura[@departamento="D01" or @departamento="D04")]/nombre

Operadores XPath

- Nombre de las asignaturas impartidas a más de dos titulaciones:
  - Expresión xPath
    - //asignatura[count(.//titulacion)>2]/nombre
    - /descendant::asignatura[count(descendant::titulacion)>2]/nombre
  - Expresión xPath en documento XML
    - //asignatura[count(.//titulacion)>2]/nombre
    - /descendant::asignatura[count(descendant::titulacion)>2]/nombre
- Número asignaturas con laboratorio impartidas a más de dos titulaciones:
  - Expresión xPath
    - count(//asignatura[count(.//titulacion)>2 and ./creditosLaboratorio])
    - count(descendant::asignatura[count(descendant::titulacion)>2 and child::creditosLaboratorio])
  - Expresión xPath en documento XML
    - count(//asignatura[count(.//titulacion)>2 and ./creditosLaboratorio])
    - count(descendant::asignatura[count(descendant::titulacion)>2 and child::creditosLaboratorio])

Tipos de datos en XPath

En XPATH 1.0:

- *Node-set:* conjunto de nodos.
- *Boolean*: true o false.
- *Number*: valor real.
- *String*: cadena de caracteres

Desde XPATH 2.0:

- *Todos los tipos de datos definidos en XMLSchemas* (es el cambio más significativo en XPATH 1.0 a 2.0).

Funciones XPath

Funciones sobre Nodos

- *Number* **count**(*node-set*)*:* devuelve el número total de nodos.
- *Number* **position()**: devuelve un número igual a la **posición contextual** del contexto de evaluación de la expresión.
- *Number* **last()**: devuelve un número igual al **tamaño** contextual del contexto de evaluación de la expresión.
- *Node-set* **id(***expresion***)***:* devuelve un conjunto de nodos con el valor del identificador.
- *String* **name(***nodo-set?***)**: devuelve una cadena de caracteres con el nombre expandido (incluyendo el prefijo *namespace*) de conjunto de nodos. Si se omite el argumento el conjunto estará formado por el nodo de contexto.
- *String* **local-name(***nodo-set?***)**: devuelve una cadena de caracteres con la parte local del nombre expandido del nodo.
- **namespace-uri(***nodo***)**: devuelve una cadena de caracteres con la URI asociada al *namespace* del nodo.

Funciones XPath

Funciones sobre Cadenas

- *String* **concat(***cad1, cad2,…***)**: devuelve la concatenación de todos los argumentos.
- *Boolean* **contains(***cad, subcad***)**: devuelve verdadero o falso en función de si la cadena de caracteres subcad se encuentra o no contenida en cad.
- *Boolean* **starts-with(***cad,subcad***)**: devuelve verdadero o falso en función de si la cadena cad empieza o no por subcad.
- *String* **string(***valor***)**: convierte el valor del argumento en una cadena de caracteres.
- *String* **substring(***cad,ini,l[ong]***)**: devuelve una subcadena de cad utilizando ini como inicio y long (si existe) como longitud.
- *String* **substring-before(***cad1, cad2***)**: devuelve la subcadena anterior a cad2 encontrada en cad1.
- *String* **substring-after(***cad1, cad2***)**: devuelve la subcadena posterior a cad2 encontrada en cad1.
- *Number* **string-length(***cad***)**: devuelve un número con la longitud de cad.
- *String* **normalize-space(***cad***)**: elimina los espacios anteriores y posteriores de cad y sustituye las sucesiones de espacios por un solo espacio.

Funciones XPath

Funciones sobre Números

- *Number* **ceiling(***número***)**: devuelve la menor cota entera superior de número.
- *Number* **floor(***número***)**: devuelve la mayor cota entera inferior de número.
- *Number* **number(***object***)**: devuelve el valor numérico de object.
- *Number* **round(***número***)**: devuelve el valor de número redondeado al entero más próximo.
- *Number* **sum(***node-set***)**: devuelve la suma el resultado de la conversión de los valores de cadena de los nodos del conjunto, a número.

Funciones booleanas

- *Boolean* **true()**: siempre devuelve verdadero.
- *Boolean* **false()**: siempre devuelve false.
- *Boolean* **not(***bool-expresion***)**: devuelve la negación de la *bool-expresion*.
- *Boolean* **boolean(***valor***)**: convierte el valor del argumento valor a verdadero a falso.

API Java

javax.xml.xpath https://docs.oracle.com/javase/8/docs/api/javax/xml/xpath/package-summary.html

- API para la evaluación de expresión XPath que satisfagan la especificación: XML Path Language (XPath) Version 3.1.

public abstract class XPathFactory

- Permite configurar y crear una instancia de un objeto **XPath**.

public abstract class XPath

- Permite evaluar expresiones XPath.
- El tipo del resultado de la evaluación de una expresión XPath, deberá especificarse mediante la utilización de uno de los siguientes **QName**:
  - XPathConstants.NODESET
  - XPathConstants.NODE
  - XPathConstants.STRING
  - XPathConstants.BOOLEAN
  - XPathConstants.NUMBER

Ejemplo XPath

- Ejemplo sencillo de utilización de XPath para la evaluación de una expresión en un documento XML: Número de asignaturas, del departamento D01, con créditos teóricos.

```
import javax.xml.xpath.*;
import org.xml.sax.InputSource;
public class XPath_EjemploSencillo {
    public static void main(String[] args){
        try{
            XPath xpath = XPathFactory.newInstance().newXPath();
            InputSource inputSource = new InputSource("plan_estudios.xml");

            String sPath="count(//asignatura[@departamento='D01']/creditosTeoricos)";
            System.out.println("\n\ExpresionXPATH:: "+ sPath);
            System.out.println("\nResultado:: "+
                                xpath.evaluate(     sPath,
                                                    inputSource,
                                                    XPathConstants.NUMBER) );
        } catch (Throwable t) {  t.printStackTrace();  }
    }
}
```

Ejmplo XPath

- Dado una documento XML válido con respecto al XMLSchema *plan_estudios.xsd,* generar un parser que, combinando Xpath con DOM, obtenga la siguiente información para cada uno de los elementos con **qname** asignatura:
  - Listado de sus atributos indicando **nombre del atributo** y **valor**.
  - Valor de cadena asociado al elemento *nombre* asociado al asignatura.
  - **Número de elementos asignatura**.

```
import java.io.*;
import javax.xml.parsers.*;
import org.w3c.dom.*;
import javax.xml.xpath.*;

public class XPath_PlanEstudios {
    //propiedad para referenciar al documento
      private Document doc;
```

Ejemplo XPath

```
public XPath_PlanEstudios(File f){
   DocumentBuilderFactory domFactory = DocumentBuilderFactory.newInstance();
   domFactory.setNamespaceAware(true);
   domFactory.setIgnoringElementContentWhitespace(true);
   try{
      DocumentBuilder builder = domFactory.newDocumentBuilder();
      this.doc = builder.parse(f.getCanonicalPath());
   } catch (Throwable t) {
      t.printStackTrace();
   }
}
public void getInformacion(){
   try{
      XPath xpath = XPathFactory.newInstance().newXPath();
      NodeList lAsignaturas= (NodeList) xpath.evaluate("//asignatura",
                                                this.doc,
                                                XPathConstants.NODESET);
      System.out.println("\nAsignaturas:: " + lAsignaturas.getLength() + "\n");
```

Ejemplo XPath

```
for(int i=0;i< lAsignaturas.getLength(); i++) {
   Element asignatura=(Element)lAsignaturas.item(i);
   /* hijo con tagname: nombre */
   System.out.println("\n- " + xpath.evaluate("./nombre", asignatura, XpathConstants.STRING) );
   /* atributos, se podría hacer con la expresión  XPATH :  'attribute::*'  */
   NamedNodeMap attList=lAsignatura.item(i).getAttributes();
    for(int j=0;j<attList.getLength();j++) {
      Node nodeAtt=attList.item(j);
       System.out.println("\n\t"+ nodeAtt.getNodeName() + ":: " +
                                   nodeAtt.getNodeValue());
   }
    /* numero de titulaciones en las que se imparte */
   System.out.println("\n\tTitulaciones:  " +
                       xpath.evaluate("count(./titulaciones/titulacion)",
                                      asignatura,
                                      XPathConstants.NUMBER)  );
      }
  } catch (Throwable t) { t.printStackTrace(); }
}
```