II. XML y Arquitectura de la Web Semántica

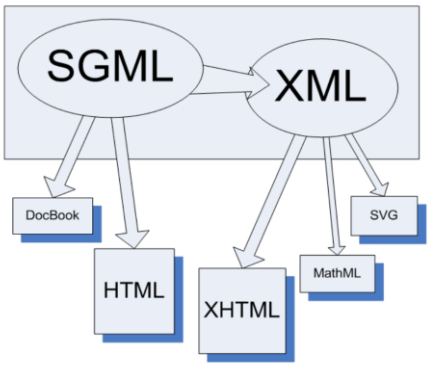
- ## XML

En la animación **De las Bibliotecas a la Web Semántica** se ilustra la necesidad de utilizar distintos modelos para organizar, recuperar e integrar información de las bibliotecas y la correspondencia de estos modelos con una arquitectura en capas de la Web Semántica.

En esa arquitectura la capa más inferior corresponde a la familia de **XML.** Comencemos entonces esta lección analizando **XML** con las palabras de uno de sus padres Tim Bray:

" XML will be the ASCII of the Web - basic, essential, unexciting."

De HTML A XML



HTML, el lenguaje para publicar páginas web, pasará a la historia como uno de los inventos más importantes de nuestro tiempo. Seguramente es tan importante para la creación y difusión de información como la imprenta. HTML y el navegador web transformaron Internet, que había existido durante dos décadas usado por científicos e ingenieros, en una plataforma de publicación ubicua utilizada por todos, desde niños de primaria hasta sus abuelas.
HTML despegó porque no era propietario y debido a su simplicidad técnica. Para los autores alcanza utilizar un editor de texto ordinario para "Marcar" un documento rodeando fragmentos de texto con "corchetes puntiagudos" y etiquetas cuyo nombre sugiere su rol estructural o formato, y el navegador hace el resto.

.....

Un problema fundamental con HTML surgió cuando la Web se transformó en una plataforma para el comercio. Hacer negocios en la Web requiere más que solo sitios web con marcas de formato para visualizar los atractivos catálogos de productos. Las empresas necesitan tener tanto la "Web para los ojos" que atrae a los clientes a sus sitios como una "Web para computadoras" que puede codificar información de productos, pedidos, facturas, pagos y otros documentos comerciales en formatos que puedan ser procesados por aplicaciones comerciales.
Para esta última tarea HTML era totalmeente inapropiado. No había etiquetas para marcar información como nombres de productos, números de artículos, precios, cantidades, etc. para darle sentido a un negocio.

Lo que el mundo necesitaba era un nuevo enfoque para usar etiquetas para marcar documentos.
En lugar de un conjunto fijo de tipos de elementos, necesitábamos una forma de definir cualquier conjunto de elementos.
Se requerían tipos de marcas para la aplicación comercial que los usaría. Se necesitaba un lenguaje de marcado extensible.

Aparece entonces **XML**, acrónimo de e**X**tensible **M**arkup **L**anguage.

[Tomado deGlushko, R. J., & McGrath, T. (2008). Document Engineering: Analyzing and

Designing Documents for Business Informatics and Web Services.*MIT Press Books1*.

Archivo accesible en

https://eva.fing.edu.uy/draftfile.php/3625/user/draft/309200238/Document-Engineering.pdf]

Algunas grandes ideas de XML que presentaremos en las siguientes secciones:

1. XML es extensible: permite la creación de nuevos conjuntos de etiquetas para contenido de dominios específicos

2. XML codifica el contenido, así como el formato de presentación de forma separada

3. Los esquemas XML definen modelos de tipos de documentos.

1. XML es extensible: permite la creación de nuevos conjuntos de etiquetas para contenido de dominios específicos

La posibilidad de creación de nuevas etiquetas es una gran ventaja de XML ya que permite agregar semántica a los elementos manejando un vocabulario controlado. Pero esto a menudo genera un conflicto al intentar utilizar en un mismo documentos XML términos homónimos de dominios diferentes. Consideremos por ejemplo el término "**capital** " del ejemplo de la Figura 1 dónde está siendo usado como una etiqueta para referenciar a un término geográfico y también para referencias a un término económico-financiero.

```
<inversión>
<país nombre="Uruguay">
<capital>Montevideo</capital>
</país>
<capital>$2000</capital>
</inversión>
```

Figura 1: Ejemplo del problema del uso de homónimos.

Una solución es trabajar con NAMESPACES o espacios de nombres identificados de forma única.

Los espacios de nombres establecidos en la especificación del W3C *Namespaces in XML* (http://www.w3.org/TR/REC-xml-names/) sirven para proveer un método simple para calificar los nombres que se usan en los documentos XML por asociación con los *namespaces* identificados por una URI (Uniform Resource Identifier).

Este *Identificador de Recursos Uniforme* es un método que combina URNs y URLs, esto es, nombres/direcciones y que sirve para identificar de forma universal recursos de todo tipo que existen en la weeb.
En nuestro ejemplo, la solución estaría en asociar a cada etiqueta una URI que indicara a qué espacio de nombres pertenece, por ejemplo, de la siguiente forma:

[http:www.bolsa.es]:capital
[http:www.geograf.com]:capital

El ejemplo de la Figura 1 quedaría entonces así:

```
<[http://www.bolsa.es]:inversiones>
    <[http://www.geograf.com]:país [http://www.geograf.com]:nombre="Uruguay">
        <[http://www.geograf.com]:capital>Montevideo
        </[http://www.geograf.com]:capital>
    </[http://www.geograf.com]:país>
    <[http:www.bolsa.es]:capital>$2000</[http:www.bolsa.es]:capital>

</[http://www.bolsa.es]:inversiones>
```

Es muy importante tener en cuenta que los URIs sólo se utilizan para que el nombre sea único, no son necesariamente enlaces con contenido, sin embargo, también los URIs sirven para acceder a recursos. Los XML Namespaces describen cómo se puede asociar una URI con cada etiqueta y atributo en un documento XML, si bien, para que se utilice la URI depende de la aplicación que lea la URI. Por ejemplo, RDF (Resource Description Framework, https://www.w3.org/TR/rdf11-concepts/), el estándar del W3C para metadatos, lo usa para enlazar cada metadato a un archivo definiendo el tipo de ese metadato. Profundizaremos en la definición y manejo de URIs en la siguiente Unidad 4 donde trabajaremos sobre RDF.

Volviendo a nuestro ejemplo, XML permite asociar un alias en el alcance (*scope*) local, así por ejemplo el namespace de XML (*xmlns*) podemos asociarlo al nombre "bolsa" o "geograf" con sus correspondientes URIs como se muestra en la Figura 2.

Recordamos que no es el objetivo de esta Unidad estudiar la sintaxis de XML, quienes estén interesados pueden encontrarla en:
XML Essentials - W3C - World Wide Web Consortium https://www.w3.org/standards/xml/core

```
xmlns:bolsa="http://www.bolsa.es"
xmlns:geograf="http://www.geograf.com">


<bolsa:inversiones>
    <geograf:país geograf:nombre="Uruguay">
     <geograf:capital>Montevideo</geograf:capital>
    </geograf:país>
   <bolsa:capital>$2000</bolsa:capital>
</bolsa:inversiones>
```

Figura 2: Solución al problema del uso de homónimos.

***** Finalmente, el uso de NAMESPACES de XML no sólo soluciona el problema de los homónimos sino que también permite reutilizar un vocabulario controlado definido por una comunidad experta en el dominio.**

Referencias

W3C.*Namespaces in XML*. http://www.w3.org/TR/REC-xml-names/(traducción al castellano: POZO, Juan R. *Espacios de Nombres en XML*. http://html.conclase.net/w3c/xml-names-es/

Algunos ejemplos de XML en dominios específicos:

- **NewsML** define un lenguaje XML para expresar noticias. https://iptc.org/standards/newsml-g2/
- **HR-XML** Define un lenguaje XML para Recursos Humanos https://schemas.liquid-technologies.com/HR-XML/2007-04-15/
- **XBRL**(**eXtensible Business Reporting Language**) diseñado para intercambiar información de negocios y financiera. https://www.xbrl.org/the-standard/what/an-introduction-to-xbrl/
- **MathML** lenguaje XML para expresar contenido científic y matemático en la web. https://www.w3.org/Math/whatIsMathML.html
- **phyloXML** es un lenguaje XML para biología evolutiva y genómica comparativa http://www.phyloxml.org/
- **XML Metadata Interchange Specification** XML para intercambio de docuemntos. https://www.omg.org/spec/XMI/About-XMI/
- **SBML: The Systems Biology Markup Language** lenguaje XML para representar procesos biológicos. http://co.mbine.org/standards/sbml
- **Augmented Reality Markup Language (ARML)** lenguaje XML para describir escenas de realidad aumentada**.** https://www.ogc.org/standards/arml
- **Travel** https://schemas.liquid-technologies.com/OpenTravel/2008B/

2. XML codifica separado el contenido del formato de presentación

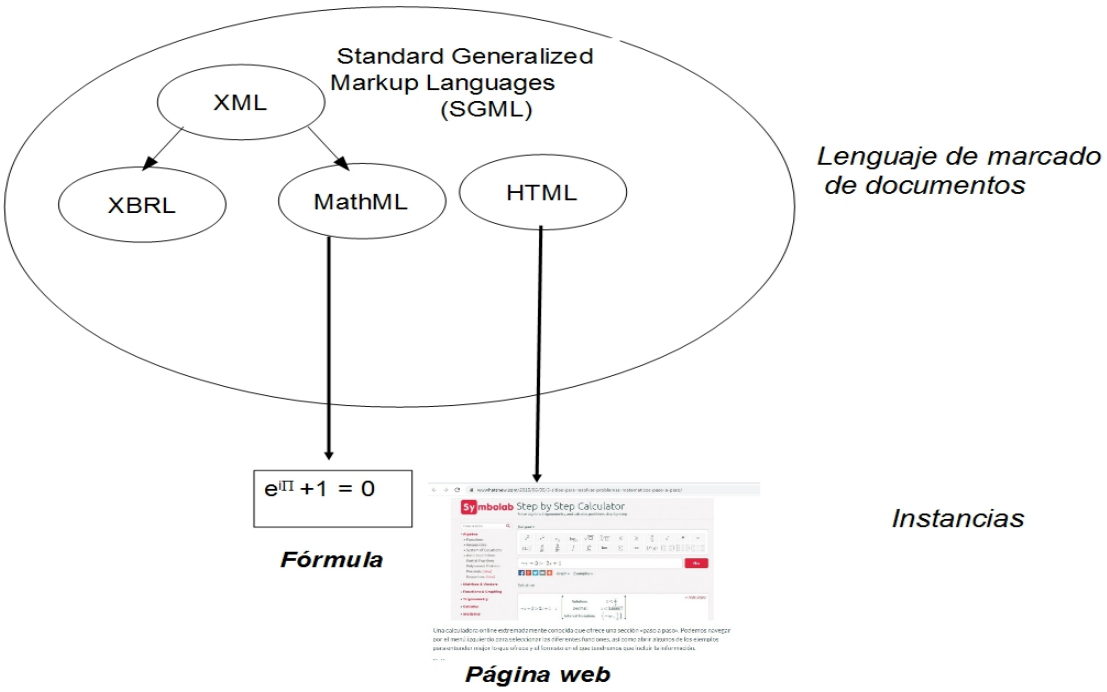


Figura 3. Diagrama de niveles de SGML.

La Figura 3 muestra que la instancia correspondiente a un documento HTML es una página web (donde se encuentran datos y presentación juntas) mientras que para un documento MathML (XML específico para matemáticas) una instancia es la fórmula matemática que describe el documento. La separación del contenido de la presentación del documento enfatiza la idea de que los elementos XML deben usarse para codificar distinciones conceptuales de una manera independiente de la presentación para permitir la reutilización y reutilización de información para diferentes contextos o implementaciones. Con el lenguaje XSLT se transforma un documento XML a HTML.

3. XML Schema define tipos de documentos

Es fácil distinguir un diccionario de una factura, un periódico de una novela o un menú del restaurante o de una colección de poemas, porque cada documento sigue un patrón estructural característico para organizar tipos de contenido que es poco probable que se encuentren en el otro.

XML Schema es un lenguaje XML que describe la estructura de un documento XML.
XML Schema es también denominado XML Schema Definition (XSD).

Un esquema es un modelo para describir la estructura de la información. Es un término tomado del mundo de la base de datos para describir la estructura de datos en tablas relacionales. En el contexto de XML, un esquema describe un modelo para toda una clase de documentos. El modelo describe la posible disposición de etiquetas y texto en un documento válido.

Cuando se envía un documento de un remitente a un receptor, es esencial que ambas partes tengan formas de validar que el documento tiene una estructura predefinida de acuerdo al modelo de documento y que además se comprenda de la misma forma su contenido.

Usando XML Schema se define la estructura del modelo de documento y se definen también tipos de datos.

Una fecha como: "03-11-2019", en algunos países, se interpretará como 3.Noviembre y en otros países como 11.Marzo.

Sin embargo, un elemento XML con un tipo de datos como este:

<date type = "date"> 2019-03-11 </date>

garantiza una comprensión mutua del contenido, porque el tipo de datos XML "fecha" requiere el formato "AAAA-MM-DD".

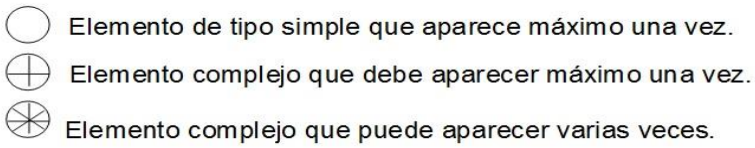
Los tipos de datos simples que define XML Schema sobre los elementos y los atributos son:

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

XML Schema permite también definir restricciones en los valores de los elementos y atributos. **Consulta en este sitio varios ejemplos: https://www.w3schools.com/xml/schema_facets.asp

Para validar el modelo de un documento en base a su estructura, XML Schema permite construir un árbol de los tipos de elementos del modelo y resultan válidos para ese modelo los documentos compatibles con ese árbol.

Por ejemplo el documento XML Schema simplificado de una Factura se puede representar con el árbol de la Figura 4 donde los nodos se interpretan según la siguiente notación:




```
<xsd:element name="factura" type="DatosFactura"/>
<xsd:complexType name="DatosFactura">
    <xsd:sequence>
        <xsd:element name="Comprador" type="DatosComprador"/>
        <xsd:element name="Productos" type="DatosProductos"/>
        <xsd:element name="Total" type="xsd:integer"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DatosComprador">
    <xsd:sequence>
        <xsd:element name="Nombre" type="xsd:string"/>
        <xsd:element name="Rut" type="xsd:integer"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DatosProductos">
    <xsd:sequence>
        <xsd:element name="Producto" minOccurs="0" maxOccurs="unbounded">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name="Item"  type="xsd:string"/>
                    <xsd:element name="Precio" type="xsd:decimal"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
```

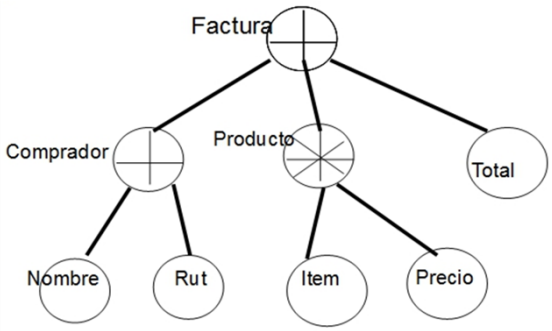


Figura 4. Ejemplo del modelo de un documento simplificado de Factura

Pueden encontrar un ejemplo de herramienta para validar un doc xml respecto a un xml schema en: https://www.liquid-technologies.com/online-xsd-validator
El lenguaje XML DOM define la forma de acceder y manipular documentos XML. Con el XML DOM se representan los documentos XML con estructura de árbol.

Observen que si volvemos a la Figura 3 y ubicamos allí XML Schema nos queda el diagrama de la Figura 5 que muestra que las instancias de un XML Schema son modelos de documentos.

Es importante notar la diferencia entre un documento XML "**Bien Formado**" y un documento XML "**Válido**". Un documentoXML puede estar **Bien-Formado** y no ser **Válido.** Está **Bien-Formado** cuando cumple las reglas de sintaxis, gramática y estructura de XML (debe tener un único elemento raíz, los elementos deben estar correctamente anidados, los nombres de etiquetas no pueden comenzar con un número o contener ciertos caracteres, etc.), mientras para ser un documento **Válido** necesita que su estructura y restricciones de valores conformen con un documento XML Schema.

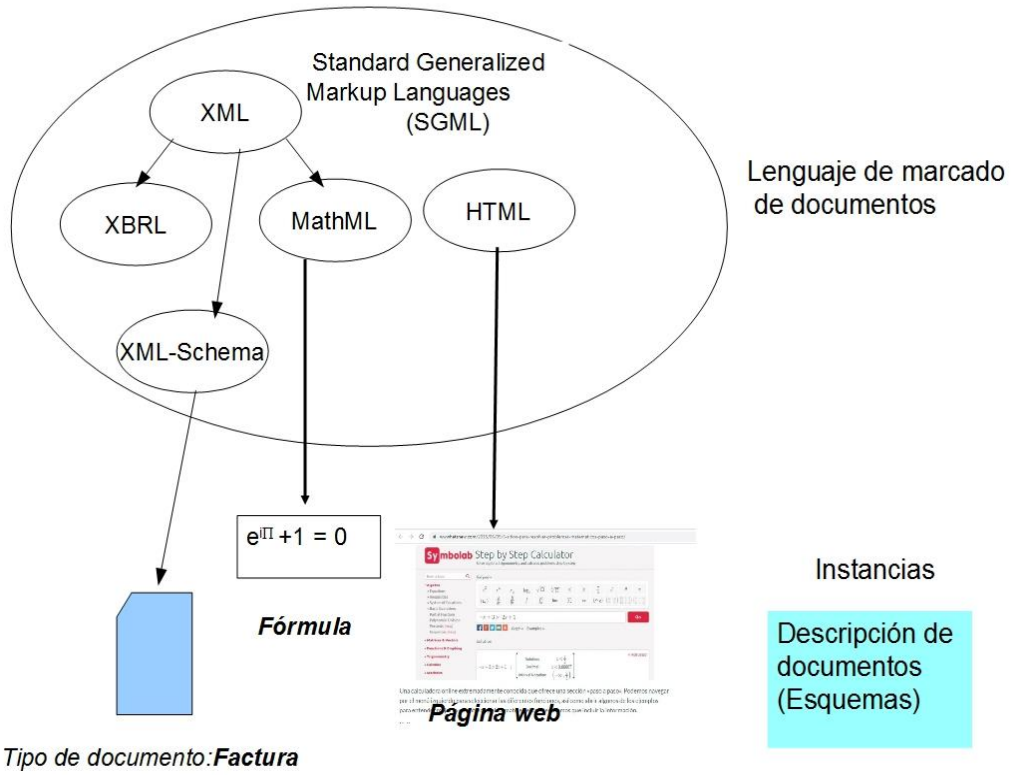


Figura 5. Diagrama de niveles de XML Schema.

Un documento XML puede hacer referencia a un documento XML Schema

El siguiente es un ejemplo de un documento XML Schema llamado "note.xsd":

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="https://www.w3schools.com"
xmlns="https://www.w3schools.com"
elementFormDefault="qualified">

<xs:element name="note">
 <xs:complexType>
  <xs:sequence>
   <xs:element name="to" type="xs:string"/>
   <xs:element name="from" type="xs:string"/>
   <xs:element name="heading" type="xs:string"/>
   <xs:element name="body" type="xs:string"/>
  </xs:sequence>
 </xs:complexType>
</xs:element>

</xs:schema>
```

El siguiente documento XML hace referencia al XML Schema note.xsd:

```
<?xml version="1.0"?>

<note
xmlns="https://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://www.w3schools.com/xml note.xsd">
 <to>Tove</to>
 <from>Jani</from>
 <heading>Reminder</heading>
 <body>Don't forget me this weekend!</body>
</note>
```

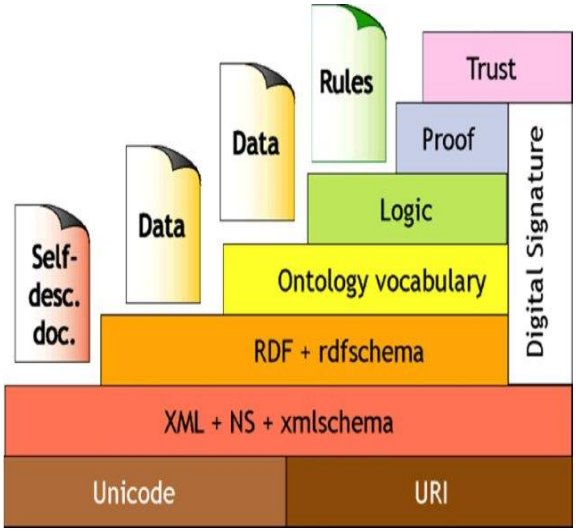
Referencias XML Schema tutorial
https://www.w3schools.com/xml/schema_intro.asp
https://www.liquid-technologies.com/xml-schema-tutorial/xsd-elements-attributes
https://www.liquid-technologies.com/xml-demonstration-videos

Un video de 2 minutos de cómo XML es usado en las reservas de hoteles. https://youtu.be/RjlhsmpFY3Y

• Sobre la Arquitectura de la Web Semántica

Presentación de las capas de lenguajes de la web semántica

La Figura 6a muestra la versión inicial con la cual se ha presentado la arquitectura de la web semántica y en la Figura 6b la versión más actual y extendida de lo que se ha dado en llamar la arquitectura de la web semántica. Varios autores entre ellos [1] han discutido el nombre de *Arquitectura* dado a estos diseños debido a que en realidad lo que describen son las capas de los distintos lenguajes que forman la web semántica.



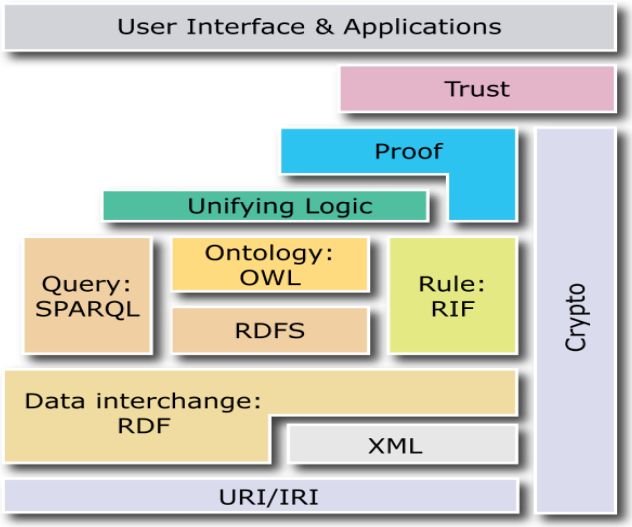


Figura 6a Figura 6b
[https://www.w3.org/2007/03/layerCake.png **]**
Figura 6: Capas de la web semántica.

Generalmente, en una arquitectura en capas para los sistemas de información, una capa representa una agrupación de elementos que proporcionan servicios relacionados. Una capa superior solo puede usar varios servicios definidos por la capa inferior inmediata (arquitectura cerrada) o servicios de todas las capas inferiores (arquitectura abierta). Sin embargo, las capas inferiores desconocen las capas superiores, y no pueden acceder a las funciones proporcionadas por las capas superiores. Esto implica un estricto orden de acceso a las funcionalidades proporcionadas por los componentes en una arquitectura en capas en una única

dirección. Sin embargo en la capa más inferior de la figura en capas de la web semántica (Figura 6) aparece Unicode como alfbeto y URIs como identificadores únicos de recursos en contacto directo con XML y sobre la capa de XML los XML Namespaces y XML Schemas que también hacen uso de las URIs y de de UNICODE. Sobre éstos aparece la capa de RDF y RDF Schema que también hacen uso extenso de las URIs. Por esta razón se nombra actualmente a estos diseños simplemente como *capas* de la web semántica o *lenguajes* de la web semántica.

En este curso nos limitaremos al estudio de los lenguajes XML Schema, RDF, RDF Schema y OWL analizando las limitaciones de cada capa y la necesidad de contar entonces con la capa superior.

En esta unidad hemos analizado las funcionalidades que brindan las URI, los Namespace y XML Schema. Si tienes cualquier duda sobre estos temas puedes enviar un mensaje al foro o ponerte en contacto directamente con tu docente.

Para poner en práctica la utilidad de XML Schema les proponemos realizar la parte de la actividad grupal que corresponde a esta unidad.

Referencias

[1] Harth, Andreas, Maciej Janik, and Steffen Staab. "Semantic web architecture." *Handbook of Semantic Web Technologies*. 2011. Está disponible el archivo pdf en la sección Materiales Complementarios de la Unidad 3 en el EVA.

[2] Gerber, Aurona, Alta Van der Merwe, and Andries Barnard. "A functional semantic web architecture." *European Semantic Web Conference*. Springer, Berlin, Heidelberg, 2008. Accesible en: https://www.researchgate.net/publication/220854050_A_Functional_Semantic_Web_Architecture

[3] Horrocks, Ian, et al. "Semantic web architecture: Stack or two towers?." *International Workshop on Principles and Practice of Semantic Web Reasoning*. Springer, Berlin, Heidelberg, 2005. Accesible en: https://www.researchgate.net/publication/220988307_Semantic_Web_Architecture_Stack_or_Two_Towers