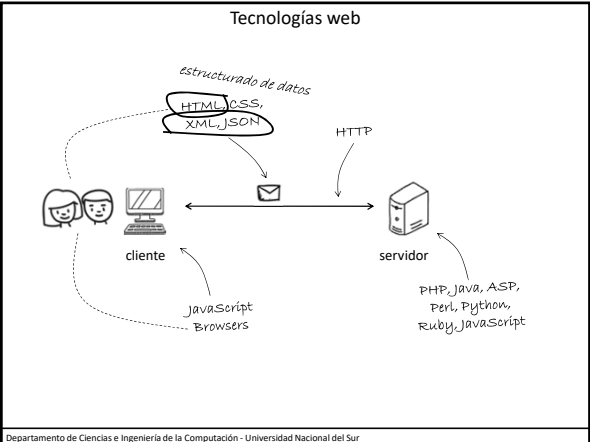


Ingeniería de Aplicaciones Web

Diego C. Martínez

Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur



Lenguajes de Marcas

HTML no es el primer lenguaje de marcado.

<p>GML IBM - 1973</p> <p>↓</p> <p>SGML ANSI - 1986</p> <p>⋮</p> <p>HTML W3C - 1990</p> <p>⋮</p> <p>XML W3C - 1998</p>	<p>Generalized Markup Language <i>Goldfarb, Mosher, Lorie</i></p> <p>Standard Generalized Markup Language <i>Un meta-lenguaje para definir lenguajes de marcas.</i></p> <p>Características distintivas:</p> <ul style="list-style-type: none"> • <i>Enfasis en marcado descriptivo, no procedural</i> • <i>Concepto de tipo de documento</i> • <i>Independencia del sistema</i> <p>XML - eXtensible Markup Language</p> <ul style="list-style-type: none"> • <i>Es la base de la interoperabilidad de muchos sistemas.</i> • <i>Es una recomendación de la W3C desde 1998.</i>
---	--

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

XML

La idea central detrás de XML es proveer una forma de estructurar información general, con independencia de la plataforma.
Facilita intercambio, procesamiento y claridad de datos

```

<matéria>
  <curricular/>
  <nombre>Historia</nombre>
  <profesor>
    Horacio Montesano
  </profesor>
  <horarios>
    <horario dia="lunes" desde="8" hasta="10"/>
    <horario dia="martes" desde="8" hasta="10"/>
  </horarios>
</matéria>

```

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

XML

Los objetivos principales son los siguientes:

1. XML debe usarse sin complicaciones sobre Internet.
2. XML debe soportar una gran variedad de aplicaciones.
3. XML debe ser compatible con SGML.
4. Debe ser fácil escribir programas que procesan documentos XML.
5. El número de características opcionales en XML debe mantenerse al mínimo.
6. Los documentos XML deben ser legibles por el humano y razonablemente claros.
7. El diseño XML debería ser preparado rápidamente.
8. El diseño de XML debe ser formal y conciso.
9. Los documentos XML deben ser fáciles de crear.
10. El laconismo es de importancia mínima.

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

XML - anatomía

```

<?xml version="1.0"?>
<matéria>
  <curricular/>
  <nombre>Historia</nombre>
  <profesor>
    Horacio Montesano
  </profesor>
  <horarios>
    <horario dia="lunes" desde="8" hasta="10"/>
    <horario dia="martes" desde="8" hasta="10"/>
  </horarios>
</matéria>

```

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

Estructurando documentos


HTML, XML y XHTML son estándares orientados principalmente a la representación y estructuración de información con algún fin particular

```
<producto>
<nombre>PS4</nombre>
<descrip>Buenisima</descrip>
<precio>17000</precio>
</producto>
```

```
<html>
<head>
<title>Hola</title>
</head>
<body>
<h2>Blog</h2>
<div>Bla bla bla</div>
</body>
</html>
```

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

JSON - una alternativa



JSON = JavaScript Object Notation
 Es un formato de intercambio de datos.
 Minimal y textual.
 Es una popular alternativa a XML.
 Es un **formato**, no un lenguaje

Basado en un subconjunto de JavaScript, Standard ECMA-262
 A veces denominados inicializadores en JavaScript


Usa convenciones de la familia de lenguajes C.
 C, C++, Java, C#, JavaScript, Perl, PHP, Python, ActionScript.

No es un formato de serialización de datos.
 No admite estructuras cíclicas ni la inclusión de funciones

json.org

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

JSON - una alternativa



JSON es un buen formato de intercambio de datos

Es fácilmente interpretable por el humano y por las computadoras

Poca decoración a los datos
 lo que facilita su transporte e interpretación algorítmica.

Soporte *Unicode*, lo que lo hace universal en su contenido

Representación de estructuras de datos elementales.
 y valores simples universales

Facilidad de mapeo de estructuras nativas a notación JSON.

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

JSON

Consiste básicamente de dos estructuras de datos *universales*

Colección de pares nombre-valor
 En muchos lenguajes: objetos, records, structs, etc

Lista ordenada de valores
 En muchos lenguajes: listas, arreglos, vectores, etc

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

JSON

Un valor JSON puede ser

- string* Caracteres Unicode entre comillas dobles
 Caracteres especiales escapados con \
- number* enteros, reales
 notación científica
- object* colecciones de variables
- array* secuencia ordenada
 JSON no habla de indexado.
- true / false*
- null* dato nulo

```
var unObjeto = {"id": 1234, "nombre": "Juan Fulano", "alumnoRegular": true, "ultimosCursos": [7145, 7854, 7321], "origen": {"ciudad": "Bahia Blanca", "provincia": "Buenos Aires"}}
```

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

JSON - arrays

```
var arreglo = [ "Argentina", "Brasil", "Chile", "Uruguay", "Ecuador", "Colombia" ]

var horarios = [ ["martes", "jueves"], [14, 14], [18, 18] ]
```

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

JSON - Objetos

```
var unAlumno = {
  "id": 1234,
  "nombre": "Juan Fulano",
  "alumnoRegular": true,
  "ultimosCursos": [
    7145,
    7854,
    7321
  ],
  "origen": {
    "ciudad": "Bahia Blanca",
    "provincia": "Buenos Aires"
  }
}
```

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

JSON - Objetos

```
var unAlumno = {
  "id": 1234,
  "nombre": "Juan Fulano",
  "alumnoRegular": true,
  "ultimosCursos": [
    7145,
    7854,
    7321
  ],
  "origen": {
    "ciudad": "Bahia Blanca",
    "provincia": "Buenos Aires"
  }
}
```

Los campos de los objetos van entre comillas dobles

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

JSON - Objetos

```
var unAlumno = {
  "id": 1234,
  "nombre": "Juan Fulano",
  "alumnoRegular": true,
  "ultimosCursos": [
    7145,
    7854,
    7321
  ],
  "origen": {
    "ciudad": "Bahia Blanca",
    "provincia": "Buenos Aires"
  }
}
```

Los valores son marcados de acuerdo a su tipo

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

JSON - Objetos

```
var unAlumno = {
  "id": 1234,
  "nombre": "Juan Fulano",
  "alumnoRegular": true,
  "ultimosCursos": [
    7145,
    7854,
    7321
  ],
  "origen": {
    "ciudad": "Bahia Blanca",
    "provincia": "Buenos Aires"
  }
}
```

Un arreglo de enteros

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

JSON - Objetos

```
var unAlumno = {
  "id": 1234,
  "nombre": "Juan Fulano",
  "alumnoRegular": true,
  "ultimosCursos": [
    7145,
    7854,
    7321
  ],
  "origen": {
    "ciudad": "Bahia Blanca",
    "provincia": "Buenos Aires"
  }
}
```

Un objeto con dos campos

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

JSON - Objetos

```
var unAlumno = {
  "id": 1234,
  "nombre": "Juan Fulano",
  "alumnoRegular": true,
  "ultimosCursos": [
    7145,
    7854,
    7321
  ],
  "origen": {
    "ciudad": "Bahia Blanca",
    "provincia": "Buenos Aires"
  }
}
```

unAlumno.id
unAlumno.nombre
unAlumno.ultimosCursos[0]
unAlumno.origen.ciudad

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

Interpretando JSON

El lenguaje utilizado debe ser capaz de comprender y generar estructuras JSON

Los *decoders* aceptan descripciones JSON bien formadas y las traducen a estructuras de datos nativas del lenguaje.

Los *encoders* traducen estructuras nativas a expresiones JSON.

JavaScript es un *decoder* natural de JSON :)
No se emplea tiempo de parsing ni transformación

Otros lenguajes ofrecen librerías para operar con datos JSON

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

JSON

JSON no posee validadores, a diferencia de XML.

JSON permite una flexibilidad mayor que XML en algunos aspectos
Permite agregar campos a una estructura sin perturbar el código existente

JSON ha demostrado ser eficiente en
el intercambio de datos entre servidores y clientes, con tecnologías heterogéneas

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

Estructura vs. presentación

HTML tiene *algunas* guías de la W3C para la visualización de documentos, basada en la interpretación de algunos tags.

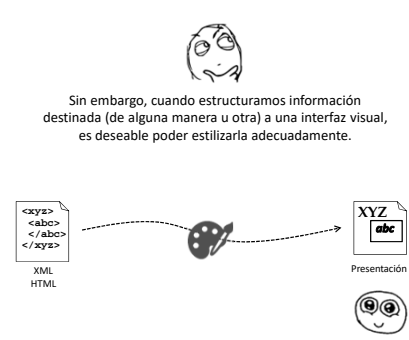
un elemento `strong` se verá en font **bold**
 un elemento `h1` se verá de mayor tamaño que uno de `h2`.

XML no incluye nada de aspectos de presentación, pues es un lenguaje de marcado de propósito general.
Tal vez no sea información destinada a la visualización.

JSON es una notación literal de estructuras de datos,
sin ninguna referencia a la presentación

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

Estructura vs. presentación



Sin embargo, cuando estructuramos información destinada (de alguna manera u otra) a una interfaz visual, es deseable poder estilizarla adecuadamente.

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

Estilos

CSS

Cascading Style Sheets

Nace en 1996, inicialmente como un reemplazo de aspectos visuales de HTML
Actualmente una herramienta avanzada de diseño de interfaces

Documento de texto con sintaxis simple.
Es aplicable a HTML y a XML.
Permite aplicar formato a elementos
Permite heredar formatos de elementos contenedores.

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

Cascading Style Sheets

Un documento CSS es simplemente una colección de reglas de la forma:

selector {property: value};

A qué tag HTML se le aplican las modificaciones indicadas.

El nombre de una cualidad del elemento elegido

El valor aplicable a esa propiedad

Por ejemplo

```

p {
color: blue;
text-align: right;
font-family:
courier;
}
hr {
color : #91B2C5;
height : 1px;
width : 100%;
}
td, tr, div {
font-family : Arial;
font-size : x-small;
color : #000;
}
    
```

Son características heredables.
Se puede indicar un font para un párrafo, y aparte un color para los links.
Luego, cada link usa sus propiedades y las del párrafo.

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

Cascading Style Sheets

font-family: "Times New Roman"

color: lightblue;

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

Cascading Style Sheets

Pueden categorizarse diferentes usos de un mismo elemento

```
p.bodytext {color: black;}
p.alert {color: red;}
```

Luego se indica, para ese elemento, de qué categoría es:

```
<p class="bodytext">
  Esto va de color negro
</p>
<p class="alert">
  Esto va de color rojo.
</p>
```

También pueden definirse elementos aplicables a cualquier tag. En este caso no se nombra el tag específico sino la clase de formato aplicada:

```
.highlight {color: blue;}
```

Esto se puede aplicar a cualquier elemento de HTML

```
<p class="highlight">Esto es texto azul</p>
<h2 class="highlight">Esto es un encabezado azul</h2>
```

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

Cascading Style Sheets

CSS se puede aplicar de formas diferentes

Inline style

```
<b style="color: blue;">Esto es bold y azul</b>
<span style="color: green;">Esto es verde</span>
```

En el mismo HTML

```
<style type="text/css">
  b { color: blue }
</style>
```

Como recurso aparte

```
<link rel="stylesheet" type="text/css" href="file.css">
```

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

Procesando XML/HTML - modelo web - scripting

Documento

```
<html>
  <sp>
</html>
```

Estructura del documento

Vista del documento

```
if()
then
else
while
for
```

simple, dinámico

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

JavaScript

JS

El más popular de los lenguajes de script para la programación del lado cliente.

JavaScript es en realidad una implementación de ECMAScript, que formaliza la creación original de Brendan Eich.

Brendan Eich

- Sintaxis simple, sin tipos de datos.
- Puede modificar dinámicamente el documento en el que se encuentra (via DOM).
- Puede reaccionar ante eventos ocurriendo en la página en la que se encuentra (onMouseOver, onlick, onLoad, etc).
- Puede ser utilizado para la validación de datos.
- Interacciones asíncronas con el servidor (AJAX).

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

JavaScript

```
<html>
<head>
  <title>Pagina</title>
</head>
<body>
  <h2>Bienvenido</h2>
  <div> etc </div>
  <script>
    //sentencias
  </script>
</body>
</html>
```

sentencias de alto nivel

El código será interpretado por el navegador.
El elemento tiene el código o una referencia a un recurso con código.

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

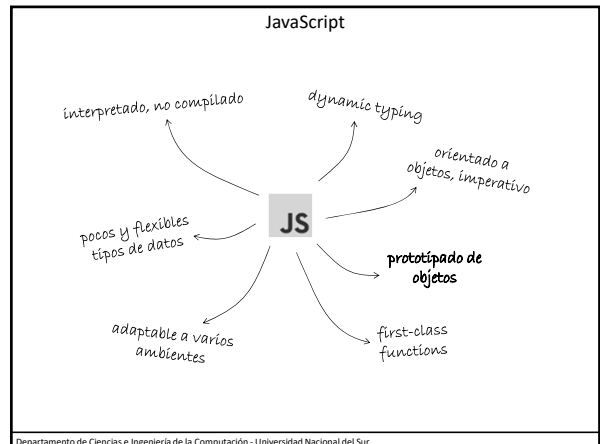
JavaScript - ejemplo

```

<html>
<head>
  <title>Ejemplo JS</title>
</head>
<body>
  <script>
    var mostrar=true
  </script>
  <p>Bienvenidos!</p>
  <script language="JavaScript">
    if(mostrar==true){ document.write("Hola! "); }
  </script>
  <p>Fin.</p>
</body>
</html>
    
```

continuidad

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur



JavaScript

first-class functions

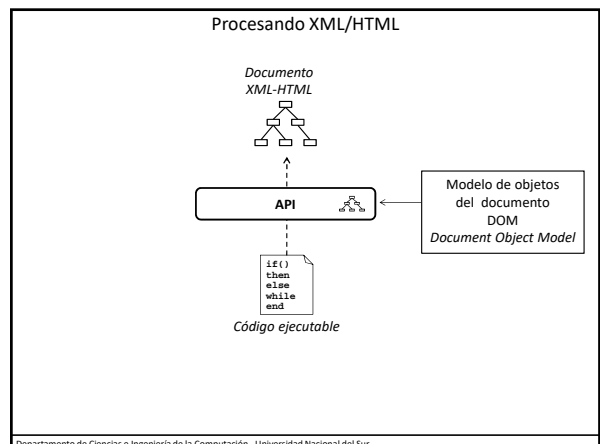
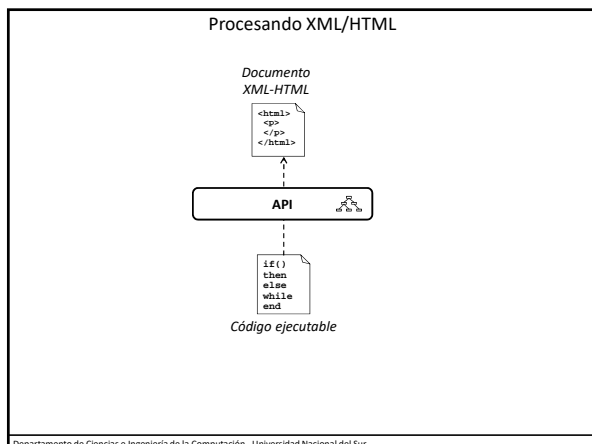
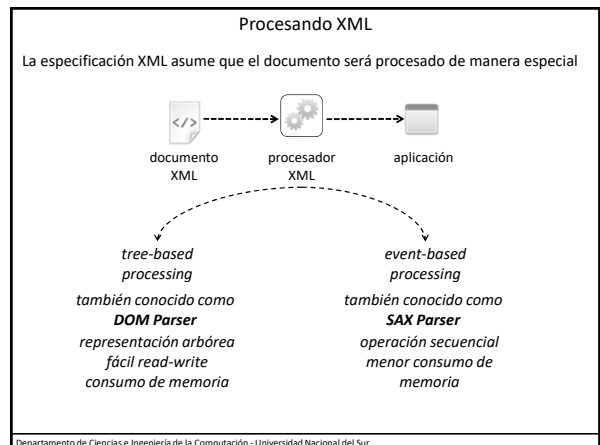
```

<script>
  var a = [ function f(){ return "Hola ";},
            function g(){ return "Mundo ";},
            "!!! " ];

  document.write(a[0]());
  document.write(a[1]());
  document.write(a[2]);

</script>
    
```

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur



Document Object Model - DOM

DOM es una interfaz de acceso y manipulación de documentos XML y HTML bien formados
Es independiente de la plataforma, del navegador y del lenguaje que lo utiliza.
Es un estándar de la W3C desde 1998

```
<table>
<tbody>
<tr>
<td>San Fierro</td>
<td>Los Santos</td>
</tr>
<tr>
<td>Las Venturas</td>
<td>San Andreas</td>
</tr>
</tbody>
</table>
```

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

Document Object Model - DOM

El estándar DOM identifica:

- Las interfaces y objetos usados para representar y manipular un documento.
- La semántica de estas interfaces y objetos (comportamiento y atributos)
- La relación y las colaboraciones entre estas interfaces y objetos

El estándar DOM no es una especificación binaria de implementación.
No es una forma de persistir objetos a XML.
No define la semántica interna de los XML.
El estándar DOM no es una estructura de datos

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

Document Object Model - ejemplo

```
<html>
<head>
<title></title>
</head>
<body>
<p>Este es un párrafo</p>
</body>
</html>
```

La siguiente expresión es equivalente al nombre "P"

```
document.documentElement.lastChild.firstChild.tagName
```

Tag HTML de la página	Tag BODY	1er. Elemento del tag BODY	Nombre del 1er. elemento del tag BODY

Un acceso más simple puede conseguirse ante la posibilidad de nombrar los elementos de un documento HTML...

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

Document Object Model - ejemplo

```
...
<p id="miParrafo">Este es un párrafo.</p>
...
```

En este caso la expresión para acceder al nombre del tag es:

```
document.getElementById("miParrafo").tagName
```

También pueden accederse a todos los tags de un mismo tipo....

```
var nodeList = document.getElementsByTagName("A");
for (var i = 0; i < nodeList.length; i++)
    nodeList[i].style.color = "#ff0000";
```

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

DOM – Propiedades de nodos

Propiedad	Descripción
nodeName	Nombre de nodo. Usualmente el nombre del tag
nodeValue	Valor del nodo. Dependiente del tipo del nodo
nodeType	Tipo del nodo, indicado como un entero.
childNodes	Un arreglo de los nodos hijos de un nodo
parentNode	El ancestro inmediato de un nodo
firstChild, lastChild	El primer hijo de un nodo El último hijo de un nodo
nextSibling, previousSibling	Hermanos de un nodo
attributes	Un arreglo de atributos del nodo, si es un elemento Si no, vacío.

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

DOM – Propiedades de nodos

Propiedad	Descripción
documentElement	Referencia al nodo raíz del documento.
createElement()	Creación de un nuevo elemento que puede insertarse en el documento. Solo creación.
createTextNode()	Creación de un nuevo elemento de tipo texto que puede insertarse en el documento.
createCDATASection()	Creación de una nueva sección CDATA que puede insertarse en el documento
getElementsByTagName()	Devuelve un arreglo de elementos que coinciden con el nombre indicado.
getElementById()	Devuelve una referencia a un nodo elemento en el documento, identificado por un Id particular.
innerHTML	El contenido HTML dentro de un dado elemento

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

Procesando XML - SAX

SAX significa *Simple Access for XML*.
Es un parser para XML basado en eventos.

SAX opera sobre el documento en forma secuencial.

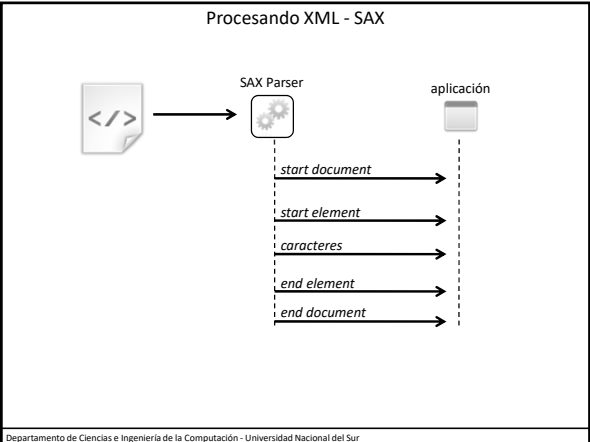
No existe una especificación formal, pero la implementación Java es referencial.

Notifica a la aplicación de eventos durante el procesado del XML.

Es más natural para aplicaciones no muy centradas en XML
Es más eficiente en recursos que la aproximación DOM
La aplicación puede comenzar a procesar el XML mientras el parser lo lee

No puede navegarse por el documento con la misma libertad que DOM
La aplicación debe registrar los eventos en los que está interesada.

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur



Procesando XML - SAX

```
<?xml version="1.0"?>
<saludo>
  <texto>
    Hola Mundo!
  </texto>
</saludo>
```

1. start document
2. start element: saludo
3. start element: texto
4. characters: Hola Mundo!
5. end element: texto
6. end element: saludo
7. end document

La aplicación debe registrar funciones *call-back* para atender estos eventos.

```
public void startElement(String uri, String localName, String qName,
  Attributes attributes)
public void endElement(String uri, String localName, String qName)
public void characters(char[] ch, int start, int length)
```

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur

