

/dev/heartbeat.GUL

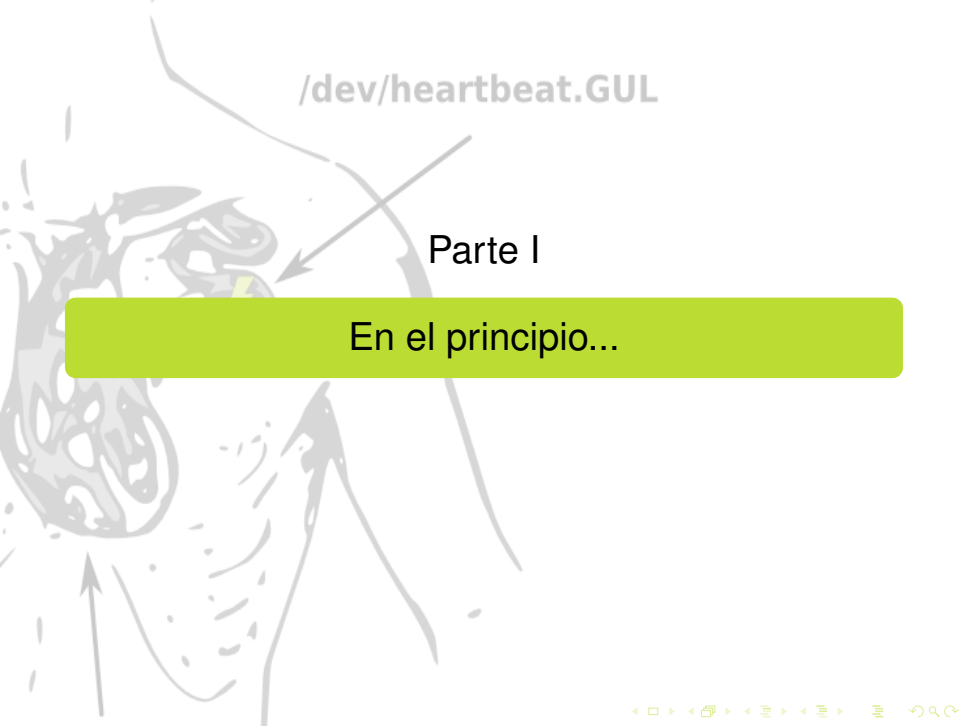
Los motores de la web 2.0

Un repaso por las tecnologías que hacen posible esta nueva experiencia

David Muñoz Díaz

Grupo de Usuarios de Linux de la UC3M - www.gul.es

6 de Marzo de 2008



`/dev/heartbeat.GUL`

Parte I

En el principio...

Web 1.0 en dos minutos

Los motores de la web 1.0

- Servidor web
- Páginas HTML estáticas
- HTTP
- Perl (la cinta aislante de la web)
- PHP
- plugins, applets, etc.

Web 1.0 en dos minutos

Los motores de la web 1.0

- Servidor web
- Páginas HTML estáticas
- HTTP
- Perl (la cinta aislante de la web)
- PHP
- plugins, applets, etc.

Web 1.0 en dos minutos

Los motores de la web 1.0

- Servidor web
- Páginas HTML estáticas
- HTTP
- Perl (la cinta aislante de la web)
- PHP
- plugins, applets, etc.

Todo esto se queda un poco corto

- La web es (prácticamente) sólo lectura
- Los usuarios quieren interactuar, a las empresas les puede convenir que así sea
- Giro heliocéntrico: del servidor al cliente

Todo esto se queda un poco corto

- La web es (prácticamente) sólo lectura
- Los usuarios quieren interactuar, a las empresas les puede convenir que así sea
- Giro heliocéntrico: del servidor al cliente

Evolución *tecnológica*

Todo esto se queda un poco corto

- Para hacerlo posible: nuevas tecnologías

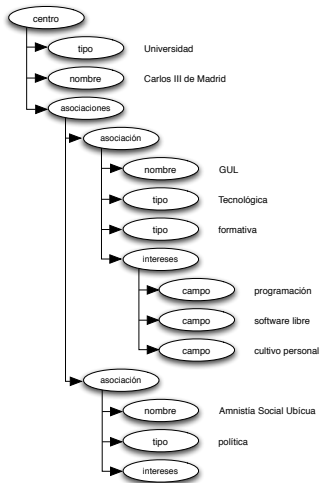
/dev/heartbeat.GUL

Parte II

XML

El origen de todo

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social UbícuA </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



¿Por qué XML?

XML se emplea en:

- Renderizado de páginas web: HTML → XHTML
 - Eliminar todo contenido relacionado con el aspecto: <italic>, , etc.
 - Accesibilidad
- Transmisión de información
 - Servicios web
 - AJAX
- Otras
 - Archivos de configuración (no del todo relacionado pero importante)

¿Por qué XML?

XML se emplea en:

- Renderizado de páginas web: HTML → XHTML
 - Eliminar todo contenido relacionado con el aspecto: `<italic>`, `<font>`, etc.
 - Accesibilidad
- Transmisión de información
 - Servicios web
 - AJAX
- Otras
 - Archivos de configuración (no del todo relacionado pero importante)

¿Por qué XML?

XML se emplea en:

- Renderizado de páginas web: HTML → XHTML
 - Eliminar todo contenido relacionado con el aspecto: <italic>, , etc.
 - Accesibilidad
- Transmisión de información
 - Servicios web
 - AJAX
- Otras
 - Archivos de configuración (no del todo relacionado pero importante)

/dev/heartbeat.GUL

Parte III

Servicios web: XML-RPC

Sólo lectura

- Es como una suscripción
- `http://cursos.gul.es/lectures/rss`
- `http://barrapunto.com/barrapunto.xml`

Se queda corto, también quiero pasar parámetros

Paso de parámetros

- Requiere saber cómo pasar estos parámetros
- En la misma URL:

`http://.....?param=...¶m2=....`

Se queda corto, los parámetros son demasiado simples

XML-RPC

- Encapsular la llamada y los parámetros en un esqueleto XML
- Lo vemos con un ejemplo: Wordpress

Servicios web propiamente dichos:

- XML-RPC es muy sencillo: XML-RPC → SOAP
- WSDL (*web service definition language*), UDDI (*universal description, discovery and integration*)

Servicio web = lógica del servicio + interfaces

/dev/heartbeat.GUL

Parte IV

Lógica de negocio: parseando XML

¿Cómo recuperar la información de un XML?

- SAX
- DOM
- XPath

¿Cómo recuperar la información de un XML?

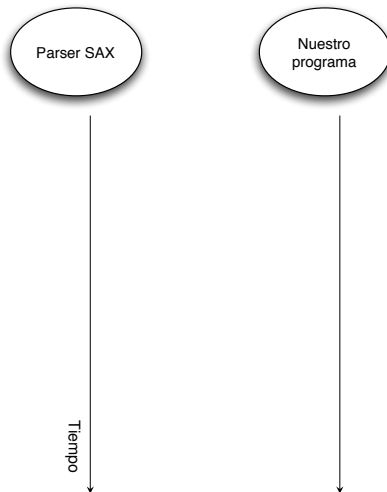
- SAX
- DOM
- XPath

Simple API for XML

- Parser a nivel léxico
- Basado en flujo de bytes
- Genera un conjunto de eventos

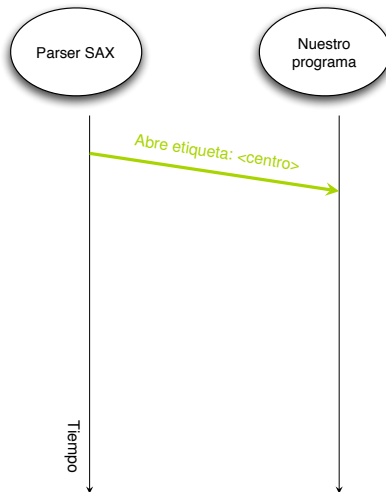
Parsear XML

`<centro><tipo>Universidad</tipo><nombre>Carlos III de Madrid</nombre><asociaciones><asociacion>...`



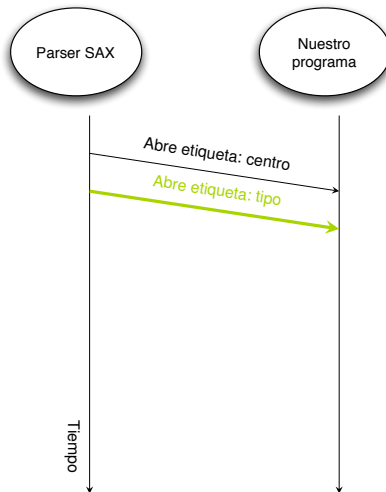
Parsear XML

`<centro><tipo>Universidad</tipo><nombre>Carlos III de Madrid</nombre><asociaciones><asociacion>...`



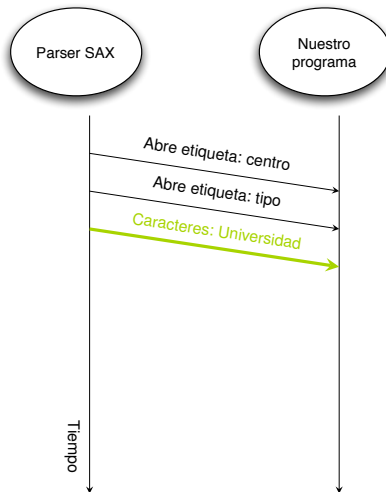
Parsear XML

`<centro><tipo>Universidad</tipo><nombre>Carlos III de Madrid</nombre><asociaciones><asociacion>...`



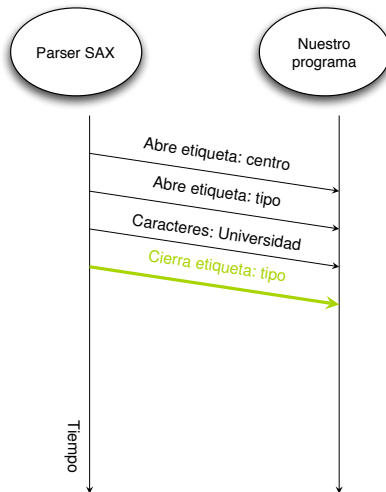
Parsear XML

<centro><tipo>Universidad</tipo><nombre>Carlos III de Madrid</nombre><asociaciones><asociacion>...



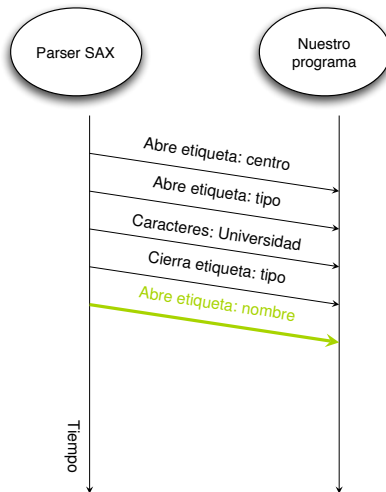
Parsear XML

`<centro><tipo>Universidad</tipo><nombre>Carlos III de Madrid</nombre><asociaciones><asociacion>...`



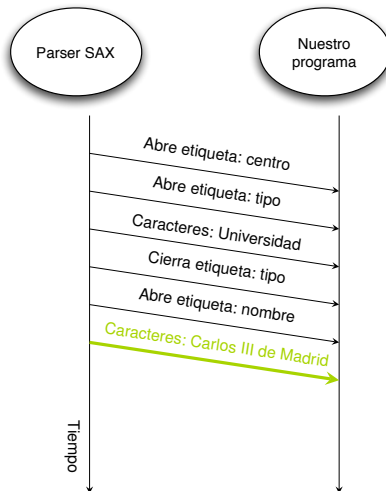
Parsear XML

`<centro><tipo>Universidad</tipo><nombre>Carlos III de Madrid</nombre><asociaciones><asociacion>...`



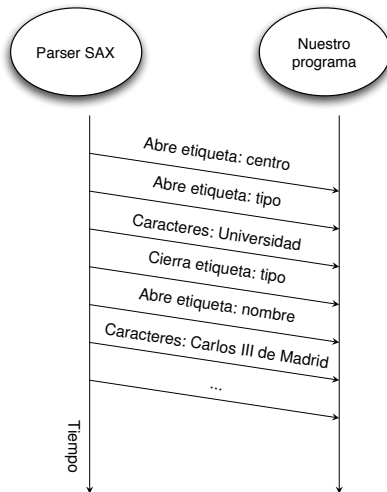
Parsear XML

`<centro><tipo>Universidad</tipo><nombre>Carlos III de Madrid</nombre><asociaciones><asociacion>...`



Parsear XML

```
<centro><tipo>Universidad</tipo><nombre>Carlos III de Madrid</nombre><asociaciones><asociacion>...
```



Simple API for XML

- Parser a nivel léxico
- Basado en flujo de bytes
- Genera un conjunto de eventos

Document Object Model

- Parser a nivel sintáctico
- Basado en la estructura en árbol del XML
- Obtener nodos, cambiar propiedades, eliminar, añadir...
- Un API muy potente, pero...
 - El API propuesto es genérico: complicado
 - Solución: un API específico para cada lenguaje

Document Object Model

- Parser a nivel sintáctico
- Basado en la estructura en árbol del XML
- Obtener nodos, cambiar propiedades, eliminar, añadir...
- Un API muy potente, pero...
 - El API propuesto es genérico: complicado
 - Solución: un API específico para cada lenguaje

Ejemplo: DOM vs JDOM (java)

```
import java.io.*;
import org.w3c.dom.*;
import org.apache.xerces.dom.*;
import org.apache.xml.serialize.*;

public class HelloDOM {
    public static void main(String[] args) {
        try {

            DOMImplementationImpl impl = (DOMImplementationImpl)
                DOMImplementationImpl.getDOMImplementation();

            DocumentType type = impl.createDocumentType("GREETING", null, null);
            DocumentImpl hello
                = (DocumentImpl) impl.createDocument(null, "GREETING", type);

            Element root = hello.createElement("GREETING");
            Text text = hello.createTextNode("Hello DOM!");
            root.appendChild(text);

            try {
                OutputFormat format = new OutputFormat(hello);
                XMLSerializer serializer = new XMLSerializer(System.out, format);
                serializer.serialize(root);
            }
            catch (IOException e) {
                System.err.println(e);
            }
        }
        catch (DOMException e) {
            e.printStackTrace();
        }
    }
}
```

Ejemplo: DOM vs JDOM (java)

```
import java.io.*;
import org.w3c.dom.*;
import org.apache.xerces.dom.*;
import org.apache.xml.serialize.*;

public class HelloDOM {
    public static void main(String[] args) {
        try {

            DOMImplementationImpl impl = (DOMImplementationImpl)
                DOMImplementationImpl.getDOMImplementation();

            DocumentType type = impl.createDocumentType("GREETING", null, null);
            DocumentImpl hello
                = (DocumentImpl) impl.createDocument(null, "GREETING", type);

            Element root = hello.createElement("GREETING");
            Text text = hello.createTextNode("Hello DOM!");
            root.appendChild(text);

            try {
                OutputFormat format = new OutputFormat(hello);
                XMLSerializer serializer = new XMLSerializer(System.out, format);
                serializer.serialize(root);
            }
            catch (IOException e) {
                System.err.println(e);
            }
        }
        catch (DOMException e) {
            e.printStackTrace();
        }
    }
}
```

```
import org.jdom.*;
import org.jdom.output.XMLOutputter;

public class HelloJDOM {
    public static void main(String[] args) {

        Element root = new Element("GREETING");
        root.addChild("Hello JDOM!");

        Document doc = new Document(root);
        XMLOutputter output = new XMLOutputter();
        try {
            output.output(doc, System.out);
        }
        catch (Exception e) {
            System.err.println(e);
        }
    }
}
```

Document Object Model

- Parser a nivel sintáctico
- Basado en la estructura en árbol del XML
- Obtener nodos, cambiar propiedades, eliminar, añadir...
- Un API muy potente, pero...
 - El API propuesto es genérico: complicado
 - Solución: un API específico para cada lenguaje
- Es más un problema de estructuras de datos que de XML :)

Document Object Model

- Parser a nivel sintáctico
- Basado en la estructura en árbol del XML
- Obtener nodos, cambiar propiedades, eliminar, añadir...
- Un API muy potente, pero...
 - El API propuesto es genérico: complicado
 - Solución: un API específico para cada lenguaje
- Es más un problema de estructuras de datos que de XML :)

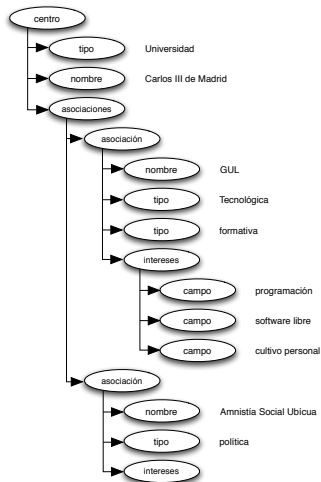
XPath

- Una forma harto sencilla de obtener los datos del documento XML
- Basado en consultas
- Semejante a una estructura de directorios de UNIX

```
while(! Volunteer.appears())  
{  
    sing("laralalaaaaa....");  
}
```

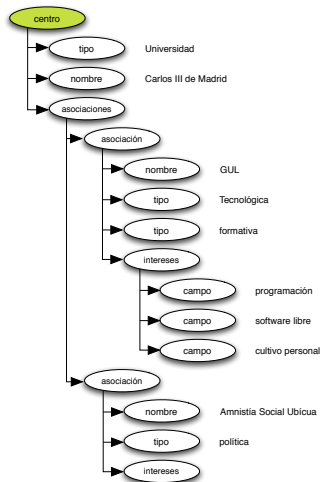
/centro

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícua </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



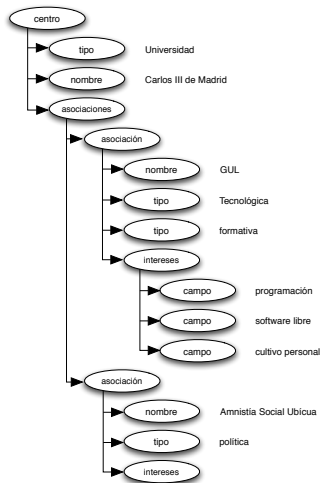
/centro

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícua </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



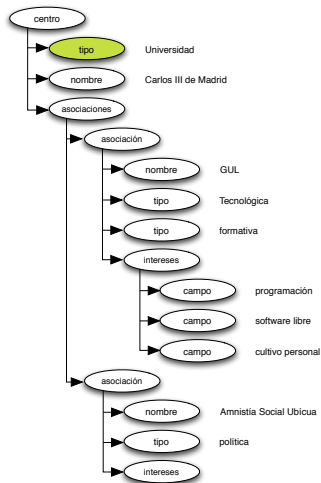
/centro/tipo

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícua </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



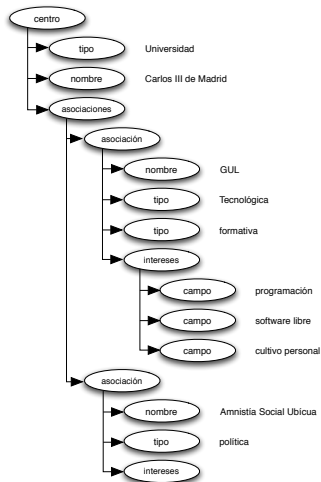
/centro/tipo

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícua </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



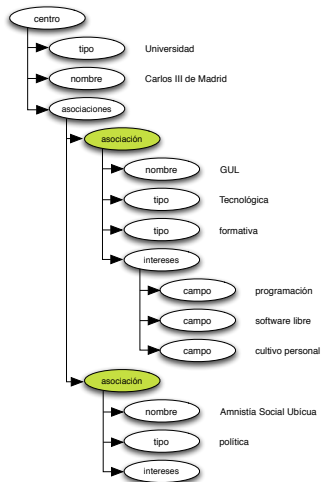
/centro/asociaciones/asociacion

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícua </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



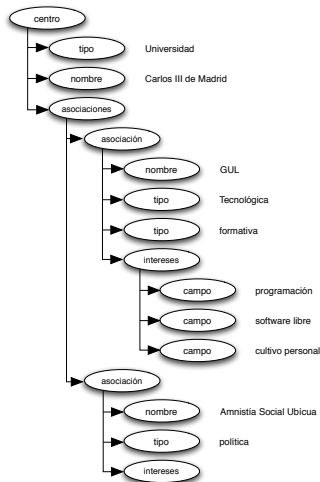
/centro/asociaciones/asociacion

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícua </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



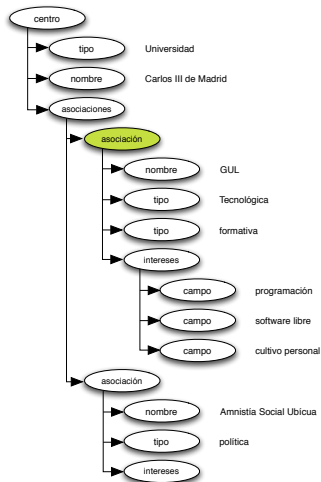
/centro/asociaciones/asociacion[1]

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícua </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



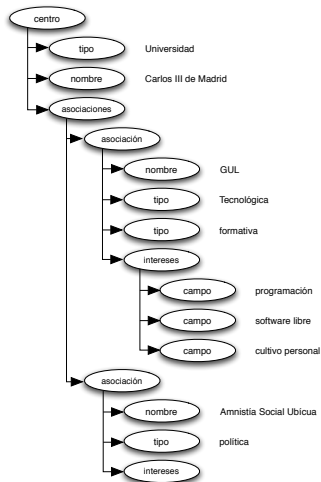
/centro/asociaciones/asociacion[1]

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícua </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



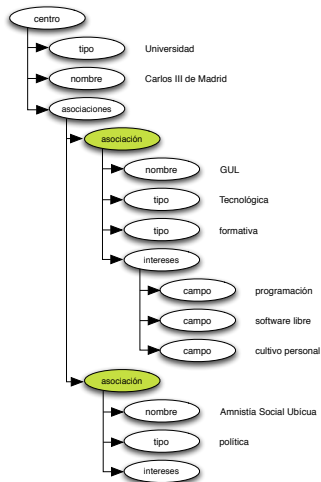
//asociacion

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícua </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



//asociacion

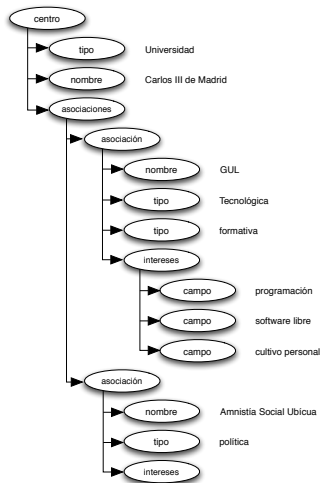
```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícu </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



XPath

//tipo

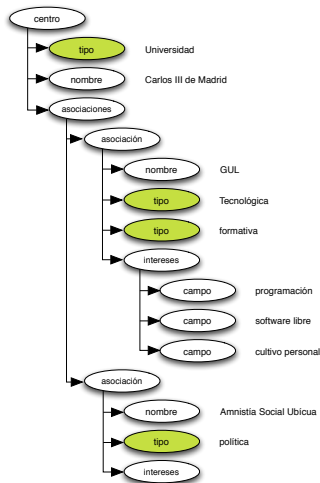
```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícua </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



XPath

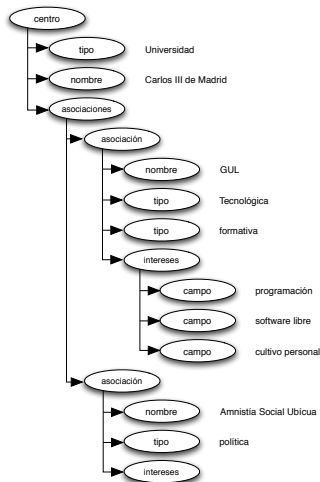
//tipo

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícua </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



//asociacion[nombre="GUL"]/intereses/campo

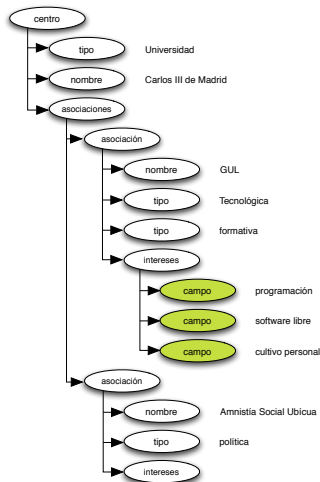
```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícu </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



XPath

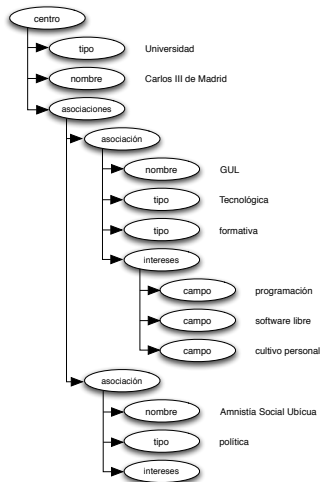
//asociacion[nombre="GUL"]/intereses/campo

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícu </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



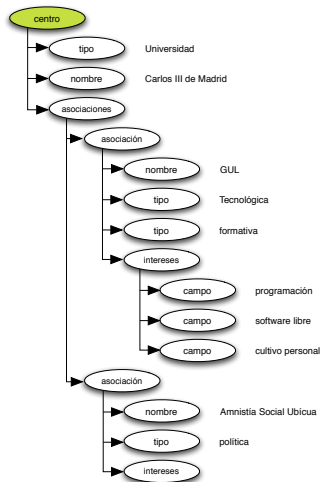
//asociaciones/..

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícua </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



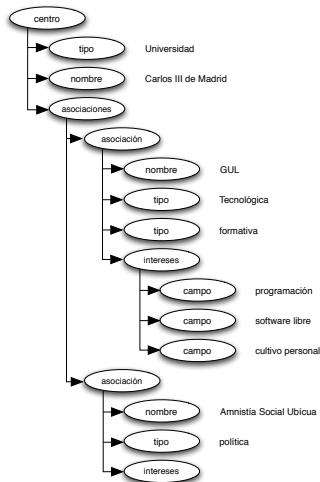
//asociaciones/..

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícua </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



//intereses/..

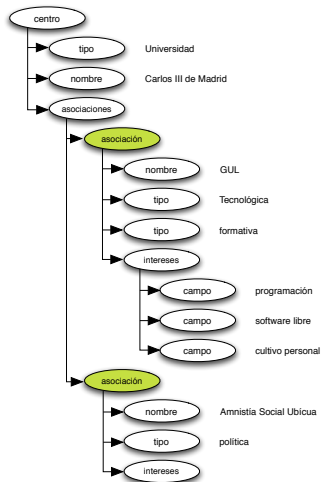
```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social UbícuA </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



XPath

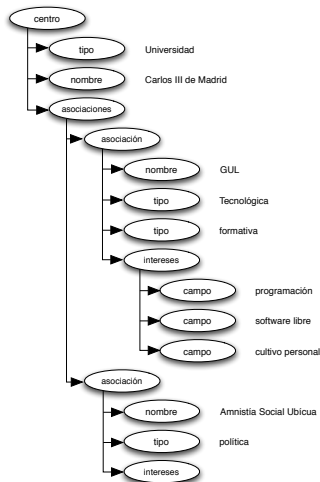
//intereses/..

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social UbícuA </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



`//campo[../../nombre="GUL"]`

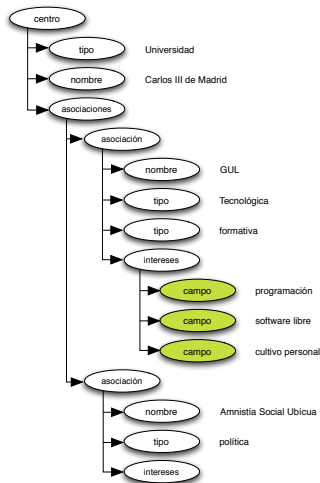
```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícua </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



XPath

//campo[../../nombre="GUL"]

```
<centro>
  <tipo> Universidad </tipo>
  <nombre> Carlos III de Madrid </nombre>
  <asociaciones>
    <asociacion>
      <nombre> GUL </nombre>
      <tipo> tecnológica </tipo>
      <tipo> formativa </tipo>
      <intereses>
        <campo> programación </campo>
        <campo> software libre </campo>
        <campo> cultivo personal </campo>
      </intereses>
    </asociacion>
    <asociacion>
      <nombre> Amnistía Social Ubícu </nombre>
      <tipo> política </tipo>
      <intereses />
    </asociacion>
  </asociaciones>
</centro>
```



/dev/heartbeat.GUL

Parte V

Transformaciones XML

XSLT

- Toma un XML y aplica transformaciones para obtener otro XML diferente
- Es, a su vez, un documento XML
- Hace uso de XPATH

XSLT

- Veamos un ejemplo

AJAX

AJAX

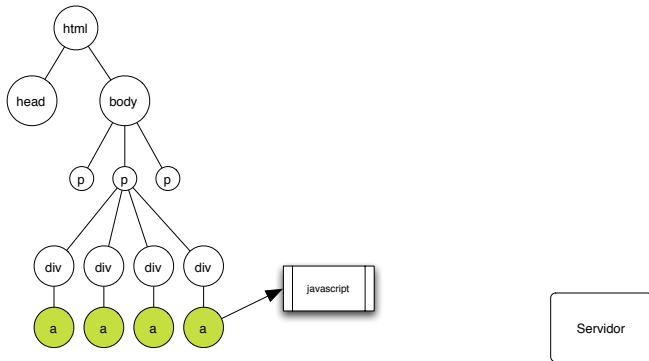
- Asynchronous Javascript And XML
- Llamada asíncrona al servidor: es una tarea *de fondo*
- Al recibir la respuesta, nos notifica
- Cambiamos el árbol DOM del documento HTML con:
 - La respuesta del servidor (lo más sencillo)
 - Transformación de la respuesta del servidor (no es recomendable)
 - Otra operación sobre la respuesta del servidor (sólo si no se puede hacer lo primero)

AJAX

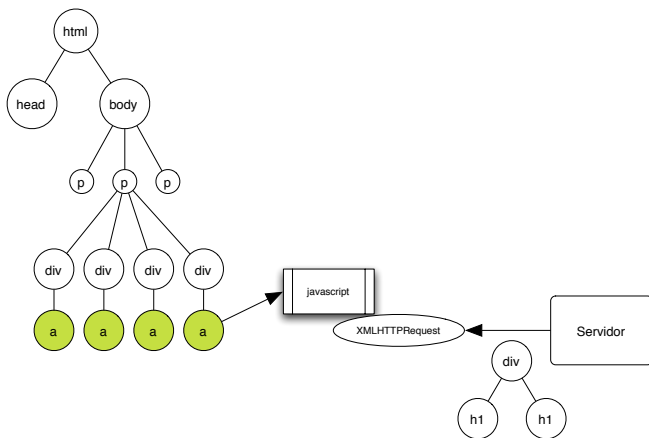
AJAX

- Lo vemos con un ejemplo

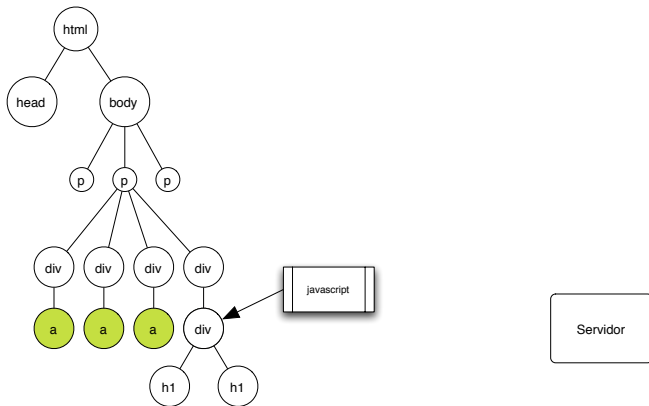
AJAX: ejemplo



AJAX: ejemplo



AJAX: ejemplo



```
self.stop 1.minute do
  self.think responsibility
  self.think accesibility
  self.think society
  self.think freedom
done
```

/dev/heartbeat.GUL

¡Fin!

- Gracias a todos por asistir
- A vuestra disposición en gul@gul.uc3m.es
- Os dejo con Sergio Gil y Ruby On Rails. ¡No os lo podéis perder!



<http://creativecommons.org/licenses/by-nc-sa/2.5/es/>