

siLeDAP: Making Easier the Interaction with Directories

◆ C. Rodríguez

Resumen

Los servicios de directorios aun siendo, hoy en día, una de las bases más utilizadas en los entornos tecnológicos no proporcionan *toolkits* o *frameworks* significativos alrededor de ellos que faciliten su interoperabilidad con las aplicaciones desplegadas en dicho entorno. Además, la problemática aumenta en el momento en que se trata de aplicar tecnologías más complejas en el campo del directorio, como COPA. siLeDAP tiene como objetivo desarrollar no sólo un navegador/administrador de directorio avanzado, aprovechando las características de las aplicaciones web 2.0, sino también crear un conjunto de clases y servicios web que permitan que las aplicaciones puedan aprovechar toda la potencia que ofrecen los servicios de directorio.

Palabras claves: servicios de directorio, servicios web

Summary

At this time, although directory services are still one of the main components in our technologic environments, they do not provide any significant toolkit or framework around in order to make easier their interoperability with the applications deployed in those environments. In addition, at the time we want to apply more complex technologies, such as COPA, problems become more numerous. siLeDAP has the main goal of developing not only an advanced browser/administrator of directory services, taking advantage of web 2.0, but also of providing a set of object classes and web services which may allow us to take advantage of all power offered by directory services.

Keywords: directory services, web services

1. Introducción

Aunque se extiende cada vez más la implantación de servidores de directorio en los distintos entornos tecnológicos que nos podemos encontrar, el mayor *handicap* que existe para su crecimiento positivo es la inflexibilidad que ofrece este tipo de plataformas.

El objetivo del proyecto siLeDAP es desarrollar una aplicación web y un conjunto de servicios que facilite la administración de servidores LDAP, resolviendo la problemática anterior, teniendo claro que debe estar diseñado para facilitar la aplicación de diferentes líneas de investigación en este campo, como puedan ser las vistas virtuales (COPA) [1] sobre el directorio.

2. Arquitectura del sistema

En una de sus líneas de actuación, siLeDAP pretende ofrecer un entorno de comunicación y trabajo más transparente y ligero con el servidor de directorio, de manera que pueda integrarse con las aplicaciones distribuidas que conviven a su alrededor. Esto es posible gracias al desarrollo de un conjunto de APIs que pueden ser utilizados a través de servicios web de tipo REST [2]. De esta forma, tanto las operaciones más básicas, como puedan ser hacer consultas en un directorio o modificar una entrada de ésta, como las más complejas, por ejemplo la integración de COPA en el directorio, son alcanzables desde cualquier aplicación que realice peticiones HTTP al conjunto de servicios web.



Los servicios de directorios aún no proporcionan *toolkits* o *frameworks* significativos alrededor de ellos que faciliten su interoperabilidad con las aplicaciones desplegadas



El objetivo del proyecto siLeDAP es desarrollar una aplicación web y un conjunto de servicios que facilite la administración de servidores LDAP



◆
Cada uno de los servicios web tiene una actuación delimitada

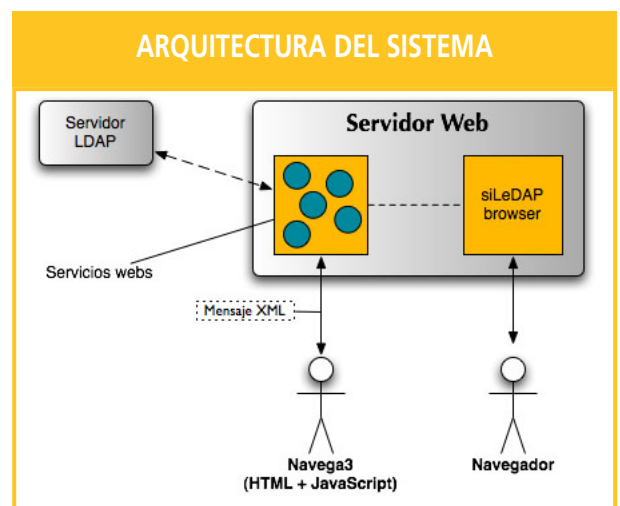
◆
Los servicios web en siLeDAP sólo admiten actualmente peticiones bajo el formato REST

Además, a la hora de modificar nuestra aplicación para que se integre con el directorio esto es independiente del lenguaje de programación utilizado, lo cual permite que sea cual sea la tecnología utilizada en la aplicación pueda aprovechar estas APIs. Cada uno de los servicios web tiene una actuación delimitada, casi atómica, como añadir una entrada o eliminarla, las cuales especifican en su documentación si en la petición HTTP se admite que se pasen parámetros por POST o por GET, así como la especificación que deberán tener.

La respuesta que generarán éstas será siempre en XML, facilitando la interacción con cualquier tipo de aplicación y permitiendo que aplicaciones web utilicen tecnología AJAX para comunicarse con el servidor de directorio.

Como vemos en la figura, tanto los servicios web como el navegador/administrador son instalados en un servidor web. La única condición que debe cumplir es que tenga compatibilidad con PHP 4, puesto que ambos están programados con dicho lenguaje.

Como hemos comentado, al realizar la aplicación todas las operaciones con el directorio se realizan a través de los servicios web, esto permite independizar el lenguaje de programación utilizado.



Para configurar correctamente dichos servicios, necesitaremos especificar en un archivo la siguiente información:

- **IP/Nombre** del servidor LDAP y su **puerto**.
- Si es una conexión **anónima** o en caso contrario el **DN** y **contraseña** del usuario.
- **DN base** al realizar las peticiones.

3. Servicios web proporcionados

Los servicios web en siLeDAP sólo admiten actualmente peticiones bajo el formato REST [2]. Se ha optado inicialmente por este formato, ya que es el tipo más simple que existe en la actualidad y el que se utiliza en las aplicaciones web que trabajan con tecnología AJAX. De cualquier forma, se espera que en un futuro admita peticiones tipo XML-RPC [3] o SOAP [4]. La estructura de una petición REST es básicamente una petición GET con parámetros bajo HTTP. Cada uno de los servicios web especifica qué parámetros admite y los valores válidos. Por ejemplo, para solicitar una entrada en concreto al directorio, utilizaríamos la siguiente URL, que como vemos, le pasamos un parámetro, dn, con el DN de la entrada codificado en base 64.

`servidor_web/siledap/API/getEntry.php?dn=ZGM9cmVkaXJpcyxkYz1lcw==`

Por otro lado, toda petición REST tiene una respuesta REST, siendo ésta un mensaje XML que contiene una serie de elementos particulares en cada servicio web. Es decir, cada uno de los servicios especifica en su documentación la estructura del mensaje y su contenido. En el caso de nuestro ejemplo anterior, la respuesta que obtendremos es:

```
<?xml version='1.0' encoding='UTF-8'?>
<Entries>
<Entry dn="uid=persona,dc=rediris,dc=es">
<Attribute name="cn">
<AttributeValue value="Manolito perez"/>
</Attribute>
<Attribute name="description">
<AttributeValue value="Currante"/>
</Attribute>
</Entry>
</Entries>
```

Actualmente se está trabajando en proveer más tipos de mensajes de respuesta como JSON [5], SOAP o XML-RPC.

La API de siLeDAP proporciona a día de hoy los siguientes servicios web:

- *Búsquedas en LDAP*: este servicio tiene como objetivo realizar las operaciones necesarias para realizar búsquedas, a las que podemos especificar el filtro de búsqueda y el alcance de las mismas.
- *Actualizaciones en LDAP*: se encarga de realizar operaciones de escritura como añadir, reemplazar o eliminar entradas en el LDAP.
- *Información del esquema*: tiene como objetivo proporcionar la información relativa al esquema almacenado en el directorio y obtener la definición de una clase de objeto (*objectClass*) o de un atributo.
- *Obtener una entrada*: obtiene una entrada de todos los pares atributos/valores del DN solicitado. De aquellos atributos que sean de tipo binario se dará su valor codificado en base 64.
- *Importar/exportar*: este servicio web es capaz de importar o exportar en formato LDIF.
- *Integración COPA*: se encarga de comunicarse con el servidor LDAP aplicando la tecnología COPA y pudiendo de esta manera navegar utilizando vistas virtuales.

La decisión de soportar inicialmente sólo las peticiones y respuestas de tipo REST es debido a que es la tecnología base que nos permite facilitar la integración de un servidor de directorio en las nuevas aplicaciones que utilizan la tecnología AJAX.

Como ejemplo, vamos a programar en nuestra aplicación web que solicite por AJAX la lista de entradas que hay a partir de un DN base determinado. Esta acción se realizará a través de JavaScript con el siguiente método:

```
function updateChildsInfo() {
    var url=urlWebServices + "searchldap.php?dnbase=" +
    encode64(dnbase) + "&scope=one";
    var request = getXMLHttpRequest();
    request.open('GET', url, true);
    request.onreadystatechange = function() {
        if (request.readyState == 4) {
            var res="";
            var xml=request.responseXML;
            var entries=xml.getElementsByTagName("Entry");
            for (var i=0; i<entries.length; i++) {
                var dnEntry=entries[i].getAttribute("dn");
                processEntry(dnEntry);
            }
        }
    }
    request.send(null);
}
```

El que las peticiones y respuestas sean sólo de tipo REST es debido a que es la tecnología base que nos facilita la integración de un servidor de directorio en las nuevas aplicaciones con tecnología AJAX



◆
siLeDAP
proporcionará una
herramienta
moderna y
avanzada para
navegar y
administrar un
servicio de
directorío

5. Conclusiones

Aun cuando estamos todavía en las etapas iniciales de desarrollo de siLeDAP, prevemos un futuro prometedor para esta iniciativa, puesto que no sólo proporcionará una herramienta moderna y avanzada para navegar y administrar un servicio de directorio, sino que su conjunto de APIs implementados a través de servicios web va a facilitar la aparición de aplicaciones que se integren con tecnologías tipo COPA, ya que no va a ser necesario que aquéllas tengan que implementarlas. Éste está siendo el principal problema a día de hoy para que las aplicaciones desarrolladas y utilizadas por los distintos organismos pertenecientes a la red académica y científica española puedan utilizarla.

Además, la realización en colaboración con RedIRIS de la versión 3 de su sencillo navegador vía web, Navega [6], ha permitido realizar el primer navegador de LDAP íntegramente con HTML y JavaScript, lo cual permite demostrar el potencial que ofrecen los servicios web proporcionados con el *software*.

Recientemente se ha dado de alta el proyecto en la Forja de RedIRIS [7], y se espera en un futuro que se publique su página web en la que se está preparando un Wiki sobre LDAP en castellano.

Referencias

- [1] *Esquema COPA*.
www.rediris.es/ldap/esquemas/copa.schema
- [2] R. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*.
www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- [3] *XML-RPC Specification*.
www.xmlrpc.com/spec
- [4] M. Gudgin, M. Hadley, N. Mendelsohn, J. J. Moreau, H. F. Nielsen. *Simple Object Access Protocol (SOAP) 1.2*.
www.w3.org/TR/soap12
- [5] *The application/json Media Type for JavaScript Object Notation (JSON)*.
<http://tools.ietf.org/html/rfc4627>
- [6] *Navega*
www.rediris.es/ldap/software/navega
- [7] *siLeDAP*
<https://forja.rediris.es/projects/siledap>

Cándido Rodríguez
(kan@cica.es)
Centro Informático Científico
de Andalucía (CICA)