

# GUÍAS BÁSICAS DE REDIRIS

## RECOMENDACIONES ACERCA DE ESTRUCTURA Y NOMBRES PARA LAS ENTRADAS EN SERVIDORES DE DIRECTORIO

Versión 1.0.4 - Febrero 2004

### 1.. Introducción

Hasta hace poco tiempo la tendencia habitual a la hora de estructurar el directorio de una organización consistía en reflejar en el mismo su estructura jerárquica (organización, facultades/departamentos, personas). Este tipo de estructuras implica una dependencia excesiva de aspectos no relacionados en absoluto con los fines del directorio y complica innecesariamente una serie de tareas enormemente comunes, como puede ser el cambio de personas de un departamento a otro. Para evitar este tipo de problemas, desde hace bastante tiempo, tanto fabricantes de software de directorio como operadores de los servicios vienen recomendando el cambio de estructura a una más plana donde la información corporativa quede almacenada en atributos de la propia entrada en lugar de en su DN.

Por otro lado, nos encontramos con el dilema de cómo nombrar estas entradas. Dado que la costumbre era que el directorio se empleara para almacenar el listado de centros y personas de la organización, los atributos usados para nombrar las entradas eran `o`, `ou` y `cn`. Además, los usuarios del directorio eran en su mayoría usuarios de los ordenadores centrales de la organización y por ello se solía usar además el atributo `uid`. Con el paso del tiempo el directorio ha ido evolucionando y actualmente se almacenan entidades electrónicas que no tienen por qué estar relacionadas con personas físicas. La mayor parte de usuarios del directorio son aplicaciones y por esta razón no tiene mucho sentido seguir usando únicamente `o`, `ou`, `cn` y `uid` para la construcción de estos DN. Es importante usar atributos que permitan a las aplicaciones poder encontrar las entradas en el directorio basándose únicamente en el DN de la misma. Un ejemplo de ello es el atributo `dc`, que permite relacionar la entrada con una localización dentro del sistema de nombres de dominio.

En muchos casos, tiene más sentido desacoplar el DN de una entrada de su contenido, lo que significa usar un atributo que no tenga un significado específico. Se ha propuesto la creación de un atributo llamado `dnComp`<sup>1</sup> (ó `dnComponent`) para este fin, de manera que podamos escribir DN de esta forma:

```
dnComp=a23b12,dc=rediris,dc=es
```

y no necesitemos conocer qué significa este valor `a23b12`. Todavía no está aceptada esta propuesta por lo que por el momento no se puede plantear su uso de manera estándar. En el IETF hay una controversia creada en relación a este atributo. Hay un sector que apuesta por el uso del atributo `cn` porque, de acuerdo al RFC 2256, es el atributo usado para contener el nombre de un objeto.

---

<sup>1</sup><http://www.georgetown.edu/giia/internet2/ldap-recipe/#RDNIssues>

En espera de que resuelvan esta controversia y decidan su definición, RedIRIS ha definido un atributo llamado `idnc`<sup>2</sup> (ó `irisDnComp`) para un uso similar. En el momento en el que `dnComp` sea un estándar nos plantearemos la migración de `irisDnComp` a `dnComp`. En caso de no llegar a serlo podremos valorar si migramos a `cn` o continuamos con `idnc`. Para usar el atributo `idnc` se ha definido la clase `irisObject`<sup>3</sup>.

## 2.. Estructura de un DN basado en componentes de dominio

Parece lógico que una identidad electrónica pueda especificarse (en términos de usuario) de la siguiente forma:

- Al estilo DNS:  
`componente.componente.componente`
- Al estilo dirección de correo:  
`usuario@componente.componente`

El rectorado de la Universidad de la Rioja podría identificarse como `"rectorado.unirioja.es"` (`dc=rectorado`, `dc=unirioja`, `dc=es`) y no perderíamos información. Hay quien opina que se pierde bastante información, por ejemplo, en el caso de Antonio López que trabaja en el Departamento de Física Atómica de la Universidad de Córdoba si su DN fuese similar a `uid=anlop`, `dc=uco`, `dc=es`. Esto no es del todo cierto ya que basta con ir al servidor LDAP correspondiente (identificado claramente por los componentes `dc` que aparecen en su DN) y acceder a sus atributos `ou`, `cn` y (por ejemplo, para saber si es un profesor, etc) `BusinessCategory` o `Role`.

Nuestra recomendación es que la afiliación de una persona o la pertenencia de cualquier otro objeto electrónico a una determinada unidad de la organización puede y debe reflejarse en la propia entrada, no en su localización en el directorio.

En esta cuestión de "ir al servidor LDAP correspondiente" es donde el empleo de los atributos `dc` cobra su importancia esencial. Usando este esquema es enormemente simple localizar el servidor LDAP que corresponde con una entrada determinada. Simplemente hemos de añadir registros SRV asociados al servicio LDAP en el DNS de nuestra organización para que apunten al servidor LDAP correspondiente. Por ejemplo en la organización con dominio `org.es` tenemos un servidor LDAP llamado `directorio.org.es`. Entonces si introducimos este registro SRV:

```
_ldap._tcp IN SRV 0 0 389 directorio.org.es.
```

Cuando preguntemos al DNS por `_ldap._tcp.org.es` obtendremos esto:

```
_ldap._tcp.org.es service = 0 0 389 directorio.org.es.
```

Podemos comprobarlo usando la pasarela web `web2ldap`<sup>4</sup>. Si necesitamos encontrar a una persona que trabaja en RedIRIS (`dc=rediris`, `dc=es`) que tiene una dirección de correo electrónico que empieza por `diego` (`mail=diego*`), y no conocemos el servidor LDAP al que hay que conectarse, basta con escribir el siguiente URL:

```
http://sites.inka.de:8002/web2ldap?ldap:///dc=rediris,dc=es??sub?(mail=diego*)
```

El sistema es capaz de encontrar la entrada correcta aunque no hayamos especificado en el URL la dirección del servidor LDAP que la mantiene. `Web2ldap` ha usado `dc=rediris`, `dc=es` para buscar en el DNS la dirección del servidor LDAP de RedIRIS que es `ldap.rediris.es:1389`.

Supongamos entonces el siguiente escenario: la Universidad de Córdoba y la de La Rioja firman un acuerdo por el cual los alumnos del "Master en Enología" de la primera pueden acceder a unas bases de datos sobre cepas de levaduras que tiene la segunda. Cuando alguien de la Universidad de Córdoba quiere acceder a esas bases de datos se conecta y envía

---

<sup>2</sup><http://www.rediris.es/si/oidregister/oidiris.es.html#idnc>

<sup>3</sup><http://www.rediris.es/ldap/esquemas/iris.schema>

<sup>4</sup><http://www.web2ldap.de/demo.html>

(supongamos) un certificado que autentifica que es "uid=alme5341,dc=uco,dc=es". Al servidor de La Rioja le resulta sumamente fácil conectarse al DNS, detectar el servidor LDAP correspondiente a la entrada y verificar si tiene el atributo adecuado que lo acredita como alumno del master. Si el certificado tiene la identidad "c=es, o=universidad de cordoba, ou=facultad de biologia, cn=perico el de los palotes" la detección del servidor LDAP es más problemática.

Cierto es que el mismo certificado enviado podría contener la condición de alumno del master, pero en ese caso es mucho más complicada la gestión: ¿Cuántos certificados debe tener una persona? ¿Cómo se gestionan las emisiones, revocaciones, etc cuando hablamos de un certificado para cada posible atribución?

Existen tecnologías para controlar el acceso a recursos restringidos que garantizan la movilidad del usuario y la independencia de gestión entre el dominio de origen del usuario y el dominio destino del servidor al que intenta acceder, como PAPI<sup>5</sup> o Shibboleth<sup>6</sup>. En estos sistemas, el usuario se identifica a sí mismo en un servidor local a su dominio de origen. Cuando intenta acceder al servidor en el dominio destino, éste y el servidor donde el usuario se ha autenticado intercambian datos sobre el mismo, de acuerdo con sus atributos y con las políticas de privacidad que siga el dominio origen. De esta manera, los datos privados del usuario (su identidad, contraseñas, etc.) no dejan nunca el dominio origen, mientras que el dominio destino mantiene un control total sobre los permisos de acceso en función de los atributos del usuario a los que tiene acceso. De nuevo, es evidente que el sistema funciona mejor con identidades digitales del estilo de "alme5341@uco.es" que "cn=perico el de los palotes, ou=facultad de biologia, o=universidad de cordoba, c=es". Este tipo de control de acceso basado en atributos puede extenderse, como es lógico, a los accesos a los propios datos del directorio, empleando el contenido de atributos en una entrada determinada para configurar los permisos de acceso. Un primer paso para ello ha sido la definición del atributo `redirisViewControl`<sup>7</sup>, que permite determinar si una entrada debe ser mostrada o no a través de un interface de consulta. RedIRIS colabora en la implementación de un sistema de control de acceso basado en atributos propios, tanto para interfaces Web como para accesos directos por medio de LDAP.

Nuestra recomendación es, en definitiva, usar componentes `dc` para nombrar cualquier entidad de la red que se corresponda "grosso modo" con un servidor/dominio en la red y usar un conjunto muy reducido de otros atributos<sup>8</sup> para entidades que correspondan a "algo dentro de un servidor" como:

- `cn` para nombres de servidores (virtuales) hospedados en el servidor, o elementos similares
- `uid` para personas y/o entes personalizables.
- `o` para organizaciones asociadas al dominio en cuestión que no dispongan de dominio propio.
- `ou` para grupos asociados con unidades de la organización que no dispongan de dominio propio.
- `idnc` para entradas en las que el valor del atributo en el DN no tenga un especial interés. Definida dentro de los atributos para la comunidad IRIS con OID 1.3.6.1.4.1.7547.4.3.2.1

De esta manera, se simplifica el uso del Directorio para lo que debe ser su objetivo fundamental: almacenar información para que las aplicaciones verifiquen propiedades sobre entidades electrónicas en la Red.

### 3. Por una estructura más plana del directorio

Hay que distinguir entre la estructura del directorio y la vista del mismo que deseemos dar hacia el exterior. Hasta ahora las aplicaciones que realizan una navegación por el directorio usan un sistema de navegación muy simple, se limitan a recorrer las entradas y la estructura que muestran es la que tiene el directorio.

Aquí proponemos una navegación inteligente basada en atributos, independientemente de la estructura del directorio. De esta forma podemos tener un directorio muy plano donde la jerarquía se implementa mediante los valores en ciertos atributos lo que permite mostrar diferentes jerarquías, es decir vistas virtuales.

<sup>5</sup><http://www.rediris.es/app/papi/>

<sup>6</sup><http://middleware.internet2.edu/shibboleth/>

<sup>7</sup><http://www.rediris.es/si/oidregister/oidiris.es.html#redirisViewControl>

<sup>8</sup>LDAP: Distinguished Names - <ftp://ftp.rediris.es/docs/drafts/draft-ietf-ldapbis-dn-09.txt>

Por ejemplo podemos tener un área `ou=prof, dc=univ, dc=es` donde se almacenen todos los profesores de la universidad y almacenamos en un campo de cada profesor el departamento al que pertenece. Lo mismo podemos hacer con las facultades, departamentos, alumnos, PAS,... Para hacer un interface que permita la navegación por departamentos solo hay que empezar mostrando todos los departamentos y realizar búsquedas por el campo donde se almacena el departamento cuando alguien pinche en uno en concreto.

Una ventaja muy importante es que el movimiento de entradas de una localización a otra no conlleva problemas. Se realiza cambiando el valor de un atributo de la entrada.

Otra posibilidad sería mostrar la universidad ordenada por áreas de conocimiento a tres niveles. Esta clasificación se puede crear introduciendo las áreas temáticas en el directorio bajo una misma localización al estilo `idnc=areas, dc=univ, dc=es` y asignandoles unos códigos especiales que indiquen la jerarquía. A cada recurso existente se le añade un atributo con uno o varios códigos que indiquen las áreas en las que ese recurso se encuentra clasificado. Ahora solo hay que hacer un pequeño programa que permita navegar por esas áreas temáticas hasta la que deseemos y que una vez en ella nos busque los profesores, departamentos o grupos de investigación que trabajan en temas relacionados. Es otra forma de mostrar el directorio sin tener que cambiar la estructura plana que pretendemos montar. Solo añadimos atributos.

Tenemos claro que poco a poco hay que cambiar la forma en la que usamos las aplicaciones de consulta del directorio ya que nos limitamos a realizar una navegación simple listando todas las entradas que existen bajo un determinado nodo y no usamos las facilidades de búsqueda mediante filtros de consulta LDAP.

Antes de recomendar una estructura muy plana hemos querido demostrar que este nuevo planteamiento es factible. Para ello hemos creado un directorio de pruebas donde almacenamos áreas temáticas y recursos. Las áreas se crean todas al mismo nivel aunque mantienen una jerarquía mediante un código, en formato COPA<sup>9</sup>, que almacenamos en un campo concreto. Los recursos se almacenan todos juntos en otra localización y al mismo nivel y, mediante otro campo, indicamos las áreas en las que ese recurso puede ser clasificado. Esto nos permite clasificar un recurso en varias áreas sin necesidad de poner alias o duplicar la entrada bajo cada una de las áreas.

Para la navegación por esta estructura de pruebas hemos modificado el programa `navega`<sup>10</sup>, que usamos para navegar vía web por el directorio LDAP de la comunidad IRIS. El resultado ha sido satisfactorio y nos permite navegar sin problemas.

Partiendo de este planteamiento se ha definido el esquema `copa`<sup>11</sup> que nos va a permitir este tipo de navegación. En la versión 2 de `navega` se podrá recorrer el directorio por varias vistas virtuales del mismo, creadas usando varios atributos en formato `copa`. Podremos hacerlo por grupos de investigación, por departamentos, por áreas temáticas o por lo que deseemos. Incluso podremos mezclar las vistas virtuales en diferentes niveles. Por ejemplo, podremos navegar por áreas temáticas y dentro un área temática determinada navegar por los departamentos o grupos de investigación que están relacionados con esa área. Simplemente habrá que añadir tantos campos en formato `copa` como vistas virtuales deseemos tener.

## 4.. Esquema para la comunidad IRIS

El directorio actual sirve como base a diferentes servicios y esto ha ocasionado que se usen atributos particulares para almacenar la información necesaria para dar soporte a estos servicios. Cada centro ha definido unos esquemas particulares y creemos que es interesante hacerlos converger en la medida de lo posible.

La primera intención de RedIRIS ha sido la definición de `irisPerson`<sup>12</sup>, una clase para albergar todos aquellos atributos necesarios para entradas de tipo persona dentro de la comunidad IRIS. Pero antes de definir una clase propia hemos querido evaluar qué han hecho otras redes académicas para ver si podemos utilizar alguna clase estándar como `eduPerson`<sup>13</sup> (versión de octubre de 2002). Esta clase nos parece interesante aunque no contiene algo esencial para las personas de origen hispano, es decir, dos atributos que nos permitan guardar, de forma separada, nuestros dos apellidos. Así pues hemos definido

---

<sup>9</sup><http://www.rediris.es/ldap/copa/>

<sup>10</sup><http://www.rediris.es/ldap/software/navega/>

<sup>11</sup><http://www.rediris.es/ldap/esquemas/copa.schema>

<sup>12</sup><http://www.rediris.es/si/oidregister/oidiris.es.html#irisPerson>

<sup>13</sup><http://www.educause.edu/eduperson/>

estos dos atributos para que contengan estos valores. Hemos decidido llamarlos `sn1`<sup>14</sup> (1.3.6.1.4.1.7547.4.3.2.2) y `sn2`<sup>15</sup> (1.3.6.1.4.1.7547.4.3.2.3) y los hemos añadido a la clase `irisPerson`.

En `eduPerson-200210` se ha cambiado la antigua definición de `eduPerson 1.0`, y ha pasado de ser una clase estructural a auxiliar. Debido a esto hay que tener en cuenta que una entrada que sea de la clase `eduPerson` lo deberá ser además de las clases `person`, `organizationalPerson` e `inetOrgPerson`. De la misma forma sería interesante empezar a evaluar el uso de la clase `eduOrg`<sup>16</sup> para las organizaciones.

Además de estas clases hemos creído necesario definir otras que contengan una serie de atributos que permitan el intercambio de información entre todos los centros de la comunidad IRIS. Estas clases incluyen:

- Información para la clasificación de la entrada según el esquema de clasificación CATRE<sup>17</sup> propuesto por RedIRIS. En el caso de una persona pueden incluirse datos sobre las áreas en las que esa persona trabaja o es experta. Para ello basta con usar `catreObject`<sup>18</sup> y los atributos `catreCode`<sup>19</sup> y `catreVersion`<sup>20</sup>.

Por ejemplo, si una persona es experta en entomología basta con que introduzca el código de dicha área en el campo `catreCode`.

```
catreCode=a01b01c08
catreVersion=20030202-1.0.4
```

- Atributo `mail` para cualquier entrada a la cual necesite asociarse una dirección de correo electrónico y cuyas clases base no lo soporten. Para aglutinar atributos relacionados con la identidad electrónica hemos definido la clase `irisInetEntity`<sup>21</sup>.

## 5.. ¿Cómo escribir un nombre?

El atributo `cn`, de acuerdo al RFC 2256, es el atributo usado para contener el nombre de un objeto. Si el objeto corresponde a una persona se almacena el nombre completo de la misma.

Este hecho siempre nos ha dado quebraderos de cabeza debido, sobre todo, a que en las lenguas que se emplean en España se usan caracteres especiales como ñes, vocales con tilde, etc. Desde RedIRIS hemos intentado hacer una guía básica sobre la gestión del Servicio de Directorio en un centro que incluye unas recomendaciones<sup>22</sup> para la conversión de los caracteres especiales a un conjunto internacional de caracteres. Esto ha llevado a multiplicar el número de ocurrencias de ese atributo en la entrada (una vez con los caracteres especiales y otra con una notación aproximada). Además de todo esto hemos detectado que en muchos centros este atributo se escribe con muchas otras variaciones como anteponer los dos apellidos, una coma y luego el nombre, poner la inicial del nombre y luego los apellidos, se añaden apodos, etc.

Por ejemplo para Antonio García Marín se pueden encontrar los siguientes valores:

```
cn=Antonio García Marín          cn=Antonio Garcia
cn=Antonio García                cn=A. Garcia
cn=Antonio                       cn=Garcia Marin, Antonio
cn=García Marín, Antonio         cn=Garcia, Antonio
cn=García, Antonio               cn=Antonio
cn=Antonio Garcia Marin          cn=Toni
```

<sup>14</sup><http://www.rediris.es/si/oidregister/oidiris.es.html#sn1>

<sup>15</sup><http://www.rediris.es/si/oidregister/oidiris.es.html#sn2>

<sup>16</sup><http://www.educause.edu/eduperson/>

<sup>17</sup><http://www.rediris.es/ldap/esquemas/catre.schema>

<sup>18</sup><http://www.rediris.es/si/oidregister/oidiris.es.html#catreObject>

<sup>19</sup><http://www.rediris.es/si/oidregister/oidiris.es.html#catreCode>

<sup>20</sup><http://www.rediris.es/si/oidregister/oidiris.es.html#catreVersion>

<sup>21</sup><http://www.rediris.es/si/oidregister/oidiris.es.html#irisInetEntity>

<sup>22</sup><http://www.rediris.es/ldap/doc/gb/>

```
cn=Tonino
cn=Antonio G. Marin
```

Como vemos, este atributo, está a menudo sobrecargado debido a la necesidad de que diferentes aplicaciones puedan acceder a él de la forma más cómoda posible. Estas aplicaciones almacenan en el atributo `cn` el nombre en el formato en el que lo necesitan para trabajar y es prácticamente imposible dar una definición precisa de cómo debería almacenarse el contenido de este atributo.

Y ¿por qué ocurre todo esto?. Es obvio que sucede porque hay que facilitar la búsqueda de esta información a los posibles usuarios del sistema. Para solucionar este problema de sobrecarga se ha hecho que nuestras aplicaciones de consulta usen filtros de búsqueda configurados correctamente para el tipo de datos que tenemos almacenados o que modifiquen los datos de la consulta para ordenarlos de acuerdo a nuestro esquema de escritura del `cn`.

Por ejemplo, si almacenamos "`cn=Obdulio Quintanilla Bermejo`" y alguien pregunta por "Quintanilla, Obdulio" podremos hacer dos cosas:

- Tener un filtro de consulta que al encontrar un carácter coma le de la vuelta a las palabras encontradas o
- Que nuestro script de consulta detecte esa coma y lo haga él

Pero con la propuesta de definición de la clase `irisPerson` y de los atributos `sn1` y `sn2` hemos simplificado bastante este tipo de búsquedas. Además, en la especificación de `eduPerson` se usan una serie de atributos que nos pueden ayudar también a reducir la sobrecarga del atributo `cn`. Algunos de ellos son:

**eduPersonNickname** Apodo, nombre informal. Suele ser una palabra en lugar de un nombre completo.

**eduPersonPrincipalName** Es el identificador de red que la persona va a utilizar para autenticación inter-institucional. Debería ser de la forma `usuario@organizacion.es`.

**displayName** Según el RFC 2798 almacena el nombre que debería aparecer cuando busquemos en aplicaciones de páginas blancas o directorios de directorios.

**givenName** Según el RFC 2256 este atributo es usado para almacenar el nombre de la persona sin los apellidos.

**initials** Según el RFC 2256 sirve para almacenar las iniciales del nombre de una persona.

**uid** Según el RFC 1274 este atributo especifica un login en un ordenador.

**uniqueIdentifier** Según el RFC 1274 este atributo especifica un identificador único para una entrada en el directorio.

## 5.1.. Recomendaciones para el correcto almacenamiento del nombre de una persona

Usando `eduPerson` e `irisPerson` el nombre de una persona puede escribirse de la siguiente forma:

```
cn= Antonio García Marín
sn= García Marín
givenName= Antonio
sn1= García
sn2= Marín
eduPersonNickname= Toni
displayName= Antonio G. Marin
```

Seguimos con el mismo problema de los caracteres especiales. Si algún atributo tiene un valor que lleva caracteres especiales recomendamos multivaluar ese atributo con un conjunto de caracteres internacionales.

## 6.. Recomendaciones generales

### 6.1.. Estructura

- Hacer que el espacio donde se almacenen las entradas sea lo más plano posible.
- No intentar reflejar la estructura jerárquica de la organización en el directorio ya que la afiliación de una persona o la pertenencia de cualquier otro objeto electrónico a una determinada unidad de la organización puede y debe reflejarse en la propia entrada, no en su localización en el directorio.
- Usar una clase interna del tipo `campusPerson` para almacenar información relativa a la Universidad como permisos de aparcamiento, bono del comedor, créditos de la biblioteca,... que no necesite ser accedida desde el exterior de la organización.
- Uso de nomenclatura DC por lo menos para la raíz de la organización.
- Uso de DN's o URLs-LDAP para relacionar entradas. Por ejemplo cuando creamos una entrada de un cargo como `cn=rector,dc=uca,dc=es` sería interesante que apuntase a la entrada de la persona correspondiente. No recomendamos crear un alias ya que si tenemos un certificado para el rector es interesante que se encuentre separado del certificado de la persona que en estos momentos es el rector. Esto implica dos entradas diferentes y posiblemente dos claves de usuario.

### 6.2.. Nombrado

- Para una entrada de tipo persona el identificador usado como RDN debe ser escogido con mucho cuidado. Quizá sea preferible escoger algún valor diferente del nombre de la persona `cn` como por ejemplo `uid`.
- El valor que se haya tomado para el RDN deberá existir posteriormente en un atributo de la entrada. Si el DN es `cn=Antonio Garcia Marin,dc=rediris,dc=es` tendrá que existir un `cn=Antonio Garcia Marin`.
- Recomendamos el uso de componentes `dc` para nombrar cualquier entidad electrónica de la red que se corresponda "grosso modo" con un servidor/dominio en la red y usar un conjunto muy reducido de otros atributos para entidades que correspondan a "algo dentro de un servidor" (como `uid, o, ou`).
- Desacoplar, en la medida de lo posible, el DN de los atributos de la entrada. Para ello el empleo de `irisDnComp` puede resultar muy útil.

### 6.3.. Atributos

- Uso de los nuevos atributos `sn1` y `sn2` de la clase `irisPerson`
- Cuidado con los atributos multivaluados. Algunos clientes tratan de diferente forma los atributos multivaluados.
- Indexar por los atributos que sean más accedidos.
- No utilizar un atributo para un uso diferente para el que fue definido. Es preferible definir un atributo nuevo.
- Activar el chequeo de esquema en la entrada de datos para evitar introducir datos que no se ajusten al esquema.

## 7.. Agradecimientos

Queremos agradecer a Nacho Díaz Asenjo (uc3m), Víctor Hernández Gómez (upo) y Pablo Fernández Baladrón (uvigo) sus comentarios como colaboración en la realización de este documento.

## 8.. Revisiones del documento

Versión	Fecha	Revisión
1.0.4	20040226	Eliminar el sufijo .txt de los enlaces a los ficheros de esquema
1.0.3	20040130	Añadir una recomendación sobre el RDN en la sección 6.2
1.0.2	20030506	Incorporar definiciones de los esquemas iris, copa y catre. Hacer mención de la controversia surgida en el IETF por la posible definición del atributo dnComp. Un sector prefiere usar el atributo cn
1.0.1	20030415	Actualización del enlace al <code>draft-ietf-ldapbis-dn-09.txt</code> - LDAP: String Representation of Distinguished Names
1.0.0	20030410	Versión inicial