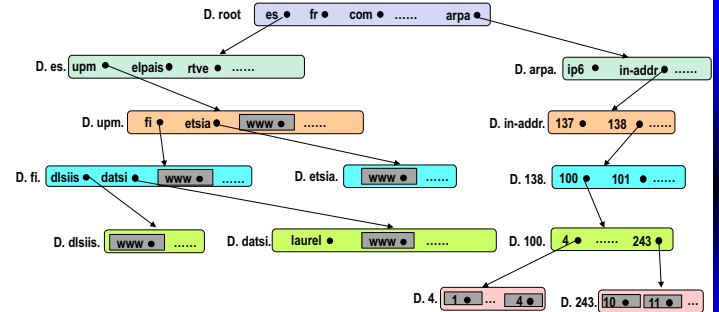


Domain Name System (DNS)

- Servicio de nombres de máquinas en Internet: nombre → IP
 - No es un serv. nombres general pero ilustrativo por escalabilidad
 - Diseño genérico: aunque uso habitual nombre de máquinas Internet
 - Inicios de Internet: fichero HOST que se actualizaba periódicamente
- Espacio de nombres de DNS jerárquico
 - Nombre: secuencia de dominios (≈directorios) de dcha. a izda.
 - `www.datsi.fi.upm.es.` → . + es + upm + fi + datsi
 - Dominio raíz: . → Caminos absolutos (FQDN) terminan con .
 - Dominios nivel superior (TLD)
 - gTLDs: genéricos (com, org, ...)
 - ccTLDs: por país (¿qué pasa con el de Tuvalu?)
 - De segundo nivel, de tercero, ...
- Implementación más usada BIND

Jerarquía de dominios DNS



Espacio de nombres distribuido: Zonas

- Zona DNS: partición del árbol global (zona ≠ dominio)
 - Información recursos de un dominio y sus subdominios no delegados
 - Delegación de dominios
 - Un subdominio puede tener su propia zona
 - Dominio padre incluye "punto de montaje" a esa zona subordinada
 - Diseño habitual: delegar todos los subdominios
 - Una zona para cada dominio (zona ≈ dominio)
 - Incluso a veces usando el mismo servidor primario que el dominio padre
 - Dominio y subdominio con mismo administrador
- Cada zona está replicada (comportamiento PAEL):
 - 1 servidor maestro/primario y N (al menos 1) esclavos/secundarios
 - Fiabilidad: mejor réplicas en distintas subredes
- Información contenida en una zona:
 - Colección de *Resource Records* (RR) que describen sus recursos

Resource Record

- Definición de un recurso: *Nombre TTL Clase Tipo Datos*
 - Clase *IN* para Internet (otros *HS*, para Hesiod, y *CH*, para Chaos)
 - `www.fi.upm.es. 86400 IN A 138.100.243.10`
- Fichero de zona:
 - Fichero de texto en serv. primario define RRs de una zona: 1 RR/línea
 - Aunque RRs se transmiten en binario
 - Incluye RRs de recursos del dominio y de subdominios no delegados
 - Sintaxis definida para facilitar introducción de datos en fichero de zona
 - Macros, caracteres especiales, caminos relativos, omisión de campos,...
- Diversos tipos de RRs
 - Nos centramos en SOA, A, AAAA, PTR, CNAME, MX, SRV, TXT y NS
 - No tratamos los RRs relacionados con la extensión DNSSEC
 - Proporciona autenticación e integridad en DNS

Ejemplo fichero de zona (Wikipedia)

```
example.com. 3600 IN SOA ns.example.com. username.example.com. ( 2007120710 86400 7200 2419200 3600)
example.com. 3600 IN NS ns.example.com. ; ns.example.com is a nameserver for example.com
example.com. 3600 IN NS ns.somewhere.example.com. ; backup nameserver for example.com
example.com. 3600 IN MX 10 mail.example.com. ; mail.example.com is the mailserver for example.com
example.com. 3600 IN MX 20 mail2.example.com. ;
example.com. 3600 IN MX 50 mail3.example.com. ;
example.com. 3600 IN A 192.0.2.1 ; IPv4 address for example.com
example.com. 3600 IN AAAA 2001:db8:10::1 ; IPv6 address for example.com
ns.example.com. 3600 IN A 192.0.2.2 ; IPv4 address for ns.example.com
ns.example.com. 3600 IN AAAA 2001:db8:10::2 ; IPv6 address for ns.example.com
www.example.com. 3600 IN CNAME example.com. ; www.example.com is an alias for example.com
wwwtest.example.com. 3600 IN CNAME www.example.com. ; another alias for www.example.com
mail.example.com. 3600 IN A 192.0.2.3 ; IPv4 address for mail.example.com
mail2.example.com. 3600 IN A 192.0.2.4 ; IPv4 address for mail2.example.com
mail3.example.com. 3600 IN A 192.0.2.5 ; IPv4 address for mail3.example.com
```

RR de tipo SOA (Start of Authority)

- Comienzo de definición de una zona
- Ejemplo de definición en fichero de zona (wikipedia)

```
example.com. IN SOA ns.example.com. username.example.com. (
2007120710 ; serial number of this zone file
1d ; slave refresh (1 day)
2h ; slave retry time in case of a problem (2 hours)
4w ; slave expiration time (4 weeks)
1h ; maximum caching time in case of failed lookups (1 hour)
)
```

- Ejemplo de consulta: *dig fi.upm.es. SOA*

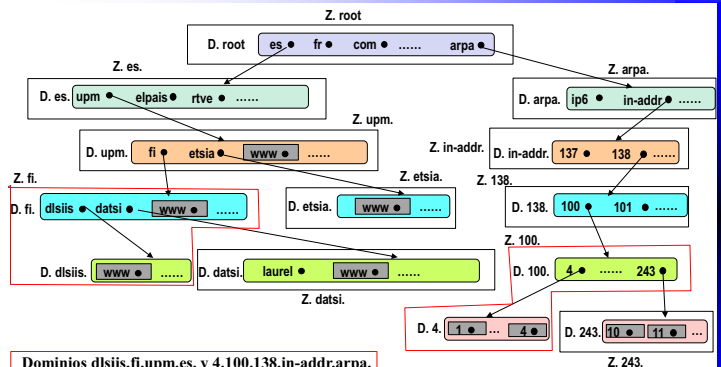
```
fi.upm.es. 86400 IN SOA chita.fi.upm.es. hostmaster.fi.upm.es. 2013102101 28800 7200 2419200 3600
Dominio; TTL; Clase Internet; Start Of Authority; S. maestro; responsable; n° serie (incrementar si cambio);
Periodo de actualización de secundario; Tiempo de reintento de secundario antes actualización fallida;
Tiempo de expiración de info. de secundario ante sincronización fallida; TTL para cache negativa (tiempo en cache de consultas erróneas)
```

- Si dominio está delegado → tiene SOA; tiene su propio fichero de zona

Registros SOA

- *dig . SOA:* . 44659 IN SOA a.root-servers.net. nstld.verisign-grs.com. 2016030901 1800 900 604800 86400
- *dig es. SOA:* es. 44796 IN SOA ns1.nic.es. hostmaster.nic.es. 2016030903 7200 7200 2592000 86400
- *dig upm.es. SOA:* upm.es. 44873 IN SOA einstein.ccupm.upm.es. hostmaster.upm.es. 2016022602 86400 7200 1209600 3600
- *dig etsia.upm.es. SOA:* etsia.upm.es. 45075 IN SOA einstein.ccupm.upm.es. hostmaster.upm.es. 2016022901 86400 7200 1209600 7200
- *dig fi.upm.es. SOA:* fi.upm.es. 86400 IN SOA chita.fi.upm.es. hostmaster.fi.upm.es. 2016021601 28800 7200 2419200 3600
- *dig datsi.fi.upm.es. SOA:* datsi.fi.upm.es. 86400 IN SOA chita.fi.upm.es. hostmaster.fi.upm.es. 2015120901 28800 7200 2592000 3600
- *dig dlsis.fi.upm.es. SOA:* No hay respuesta
- *dig arpa. SOA:* arpa. 45018 IN SOA a.root-servers.net. nstld.verisign-grs.com. 2016030900 1800 900 604800 86400
- *dig in-addr.arpa. SOA:* in-addr.arpa. 3600 IN SOA b.in-addr-servers.arpa. nstld.iana.org. 2015073098 1800 900 604800 3600
- *dig 138.in-addr.arpa. SOA:* 138.in-addr.arpa. 44938 IN SOA z.arin.net. dns-ops.arin.net. 2016013238 1800 900 691200 10800
- *dig 100.138.in-addr.arpa. SOA:* 100.138.in-addr.arpa. 44773 IN SOA einstein.ccupm.upm.es. hostmaster.upm.es. 2016022901 86400 7200 1209600 7200
- *dig 4.100.138.in-addr.arpa. SOA:* No hay respuesta
- *dig 243.100.138.in-addr.arpa. SOA:* 243.100.138.in-addr.arpa. 86400 IN SOA chita.fi.upm.es. hostmaster.fi.upm.es. 2014091801 28800 7200 2592000 3600

Jerarquía de zonas DNS



Dominios dlsis.fi.upm.es y 4.100.138.in-addr.arpa no están delegados. Están incluidos en la zona del dominio padre.

Zonas y servidores primarios

- zona . a.root-servers.net
- zona es ns1.nic.es
- zona upm.es einstein.ccupm.upm.es
- zona etsia.upm.es einstein.ccupm.upm.es.
 - Mismo servidor primario que su padre → mismo administrador
- zona fi.upm.es chita.fi.upm.es.
 - Incluye subdominio dlsis.fi.upm.es.
- zona datsi.fi.upm.es chita.fi.upm.es.
 - Mismo servidor primario que su padre → mismo administrador
- zona arpa a.root-servers.net.
 - Mismo servidor primario que su padre → mismo administrador
- zona in-addr.arpa b.in-addr-servers.arpa.
- zona 138.in-addr.arpa z.arin.net.
- zona 100.138.in-addr.arpa einstein.ccupm.upm.es.
 - Incluye subdominio 4.100.138.in-addr.arpa.
- zona 243.100.138.in-addr.arpa chita.fi.upm.es.

RR de tipo A o AAAA

- Dirección de máquina: A (IPv4) y AAAA (IPv6)

Ejemplo: *dig www.fi.upm.es. A*

```
www.fi.upm.es. 86400 IN A 138.100.243.10
```

Ej: *dig galileo.ccupm.upm.es. A*

```
galileo.ccupm.upm.es. 76103 IN A 138.100.4.4
```

Ej: *dig galileo.ccupm.upm.es. AAAA*

```
galileo.ccupm.upm.es. 76223 IN AAAA 2001:720:41c:40:12:100:4:4
```

- Múltiples recursos con mismo nombre (reparto de carga)

```
yahoo.es. 300 IN A 98.137.236.24
```

```
yahoo.es. 300 IN A 77.238.184.24
```

```
yahoo.es. 300 IN A 74.6.50.24
```

```
yahoo.es. 300 IN A 212.82.102.24
```

```
yahoo.es. 300 IN A 106.10.212.24
```

- Nótese TTL bajo en RR para favorecer el reparto de carga

RR de tipo CNAME (Canonical NAME)

- Alias: Nuevo nombre para mismo recurso (ejemplo real)

```
www.datsi.fi.upm.es. 86400 IN CNAME avellano.datsi.fi.upm.es.
```

```
avellano.datsi.fi.upm.es. 86400 IN A 138.100.9.22
```

- Frente a (ejemplo hipotético):

```
www.datsi.fi.upm.es. 86400 IN A 138.100.9.22
```

```
avellano.datsi.fi.upm.es. 86400 IN A 138.100.9.22
```

- Más flexibilidad ante cambios pero ineficiencia por indirección

- Pueden encadenarse:

```
www.elpais.com. 967 IN CNAME elpais.es.edgesuite.net.
```

```
elpais.es.edgesuite.net. 10467 IN CNAME a1749.g.akamai.net.
```

```
a1749.g.akamai.net. 20 IN A 130.206.192.24
```

```
a1749.g.akamai.net. 20 IN A 130.206.192.49
```

RR de tipo MX

- Servidores de correo para dominio con orden de preferencia

- Formato: *Nombre TTL clase MX prioridad servidor*

- Ejemplo de consulta: *dig upm.es. MX*

```
upm.es. 73788 IN MX 10 relay.upm.es.
```

```
upm.es. 73788 IN MX 30 relay4.upm.es.
```

```
upm.es. 73788 IN MX 50 correo.upm.es.
```

- Ejemplo de consulta: *dig fi.upm.es. MX*

```
fi.upm.es. 60 IN MX 10 relay.fi.upm.es.
```

```
fi.upm.es. 60 IN MX 100 relay.upm.es.
```

```
fi.upm.es. 60 IN MX 600 relay.fi.upm.es.
```

- Número indica orden de preferencia: ↓ prioridad → ↑ preferencia
- Remitente de correo debe contactar con servidor de menor n°
- Si caído con el siguiente, ...

RR de tipo SRV

- Permite especificar qué máquinas dan un servicio en el dominio
- Formato: `_servicio._protocolo.nombre TTL clase SRV prioridad peso puerto servidor`
- Permite especificar prioridades y reparto entre misma prioridad
- Ejemplo de wikipedia:

```
_sip._tcp.example.com. 86400 IN SRV 10 60 5060 bigbox.example.com.  
_sip._tcp.example.com. 86400 IN SRV 10 20 5060 smallbox1.example.com.  
_sip._tcp.example.com. 86400 IN SRV 10 10 5060 smallbox2.example.com.  
_sip._tcp.example.com. 86400 IN SRV 10 10 5066 smallbox2.example.com.  
_sip._tcp.example.com. 86400 IN SRV 20 0 5060 backupbox.example.com.
```

- Ejemplo real: dig SRV _xmpp-server._tcp.gmail.com.

```
_xmpp-server._tcp.gmail.com. 900 IN SRV 20 0 5269 alt2.xmpp-server.l.google.com.  
_xmpp-server._tcp.gmail.com. 900 IN SRV 20 0 5269 alt4.xmpp-server.l.google.com.  
_xmpp-server._tcp.gmail.com. 900 IN SRV 20 0 5269 alt1.xmpp-server.l.google.com.  
_xmpp-server._tcp.gmail.com. 900 IN SRV 5 0 5269 xmpp-server.l.google.com.  
_xmpp-server._tcp.gmail.com. 900 IN SRV 20 0 5269 alt3.xmpp-server.l.google.com.
```

- ¿Por qué se usan tan poco? ¿Por qué no se usan para la Web?

RR de tipo NS (Name Server)

- Primer uso: especificar servidores de nombres para un dominio
- Ejemplo: `dig fi.upm.es. NS`

```
fi.upm.es. 86400 IN NS chita.fi.upm.es.  
fi.upm.es. 86400 IN NS zape.fi.upm.es.  
fi.upm.es. 86400 IN NS tarzan.fi.upm.es.  
fi.upm.es. 86400 IN NS galileo.ccupm.upm.es.  
fi.upm.es. 86400 IN NS ns.fi.upm.es.
```

- Ejemplo: `dig 243.100.138.in-addr.arpa. NS`

```
243.100.138.in-addr.arpa. 86400 IN NS zape.fi.upm.es.  
243.100.138.in-addr.arpa. 86400 IN NS chita.fi.upm.es.  
243.100.138.in-addr.arpa. 86400 IN NS galileo.ccupm.upm.es.  
243.100.138.in-addr.arpa. 86400 IN NS tarzan.fi.upm.es.  
243.100.138.in-addr.arpa. 86400 IN NS ns.fi.upm.es.
```

Servidores de nombres en UPM

- Ejemplo: `dig upm.es. NS`

```
upm.es. 7054 IN NS galileo.ccupm.upm.es.  
upm.es. 7054 IN NS einstein.ccupm.upm.es.  
upm.es. 7054 IN NS sun.rediris.es.  
upm.es. 7054 IN NS chico.rediris.es.
```

- Ejemplo: `dig etsia.upm.es. NS`

```
etsia.upm.es. 42359 IN NS einstein.ccupm.upm.es.  
etsia.upm.es. 42359 IN NS galileo.ccupm.upm.es.
```

- Ejemplo: `dig fi.upm.es. NS`

```
fi.upm.es. 86400 IN NS chita.fi.upm.es.  
fi.upm.es. 86400 IN NS zape.fi.upm.es.  
fi.upm.es. 86400 IN NS tarzan.fi.upm.es.  
fi.upm.es. 86400 IN NS galileo.ccupm.upm.es.  
fi.upm.es. 86400 IN NS ns.fi.upm.es.
```

- Ejemplo: `dig datsi.fi.upm.es NS`

```
datsi.fi.upm.es. 86400 IN NS ns.fi.upm.es.  
datsi.fi.upm.es. 86400 IN NS chita.fi.upm.es.  
datsi.fi.upm.es. 86400 IN NS galileo.ccupm.upm.es.  
datsi.fi.upm.es. 86400 IN NS tarzan.fi.upm.es.  
datsi.fi.upm.es. 86400 IN NS zape.fi.upm.es.
```

RR de tipo NS para delegación

- Segundo uso: delegar subdominio a s.nombres (pto. montaje)
 - Aparece como nombre del RR el del subdominio
 - Lista subdominios delegados no se puede obtener mediante consulta
- Ejemplo: UPM delega administración de sus recursos DNS a FI:
 - fichero de zona de `upm.es.` debe incluir:

```
fi.upm.es. 86400 IN NS chita.fi.upm.es.  
fi.upm.es. 86400 IN NS zape.fi.upm.es.  
fi.upm.es. 86400 IN NS tarzan.fi.upm.es.  
fi.upm.es. 86400 IN NS galileo.ccupm.upm.es.  
fi.upm.es. 86400 IN NS ns.fi.upm.es.
```

- fichero de zona de `100.138.in-addr.arpa.` debe incluir:

```
243.100.138.in-addr.arpa. 86400 IN NS zape.fi.upm.es.  
243.100.138.in-addr.arpa. 86400 IN NS chita.fi.upm.es.  
243.100.138.in-addr.arpa. 86400 IN NS galileo.ccupm.upm.es.  
243.100.138.in-addr.arpa. 86400 IN NS tarzan.fi.upm.es.  
243.100.138.in-addr.arpa. 86400 IN NS ns.fi.upm.es.
```

Glue records

- Posibles círculos viciosos en la traducción de nombres
- Si s. nombres de subdominio (Ssub) pertenece a subdominio
 - P.e. en dominio `upm.` SN de `fi.`: `chita.fi.upm.es.`, `ns.fi.upm.es.`, ...
 - Para obtener IP de cualquier máq. subdominio → contactar con Ssub
 - ¿IP de `www.fi.upm.es.?` → contactar con `chita.fi.upm.es.`
 - Pero para hacerlo necesito IP de Ssub → contactar con Ssub
 - ¿IP de `chita.fi.upm.es.?` → contactar `chita.fi.upm.es.`
- **Glue record (GR)**
 - RR de tipo A/AAAA que se incluye en un dominio ajeno
 - Solución c. vicioso: padre debe incluir RR tipo A con dir. IP de Ssub
 - Aumenta problemas de coherencia
 - Cambios en IP de Ssub deben reflejarse también en dominio padre
 - No necesario *glue record* para servidor externo o en dominio padre
 - Siempre se puede obtener su traducción

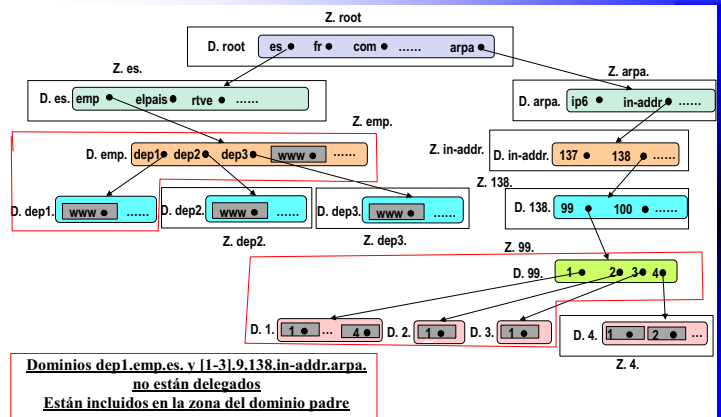
Delegaciones y Glue Records en UPM

```
• upm.es.  
etsia.upm.es. 42359 IN NS einstein.ccupm.upm.es.  
etsia.upm.es. 42359 IN NS galileo.ccupm.upm.es.  
fi.upm.es. 86400 IN NS chita.fi.upm.es.  
fi.upm.es. 86400 IN NS zape.fi.upm.es.  
fi.upm.es. 86400 IN NS tarzan.fi.upm.es.  
fi.upm.es. 86400 IN NS galileo.ccupm.upm.es.  
fi.upm.es. 86400 IN NS ns.fi.upm.es.  
chita.fi.upm.es. 86400 IN A 138.100.8.23 ; Glue Record  
tarzan.fi.upm.es. 86400 IN A 138.100.8.6 ; Glue Record  
zape.fi.upm.es. 86400 IN A 138.100.8.1 ; Glue Record  
ns.fi.upm.es. 86400 IN A 138.100.8.11 ; Glue Record  
• fi.upm.es  
datsi.fi.upm.es. 86400 IN NS ns.fi.upm.es.  
datsi.fi.upm.es. 86400 IN NS chita.fi.upm.es.  
datsi.fi.upm.es. 86400 IN NS galileo.ccupm.upm.es.  
datsi.fi.upm.es. 86400 IN NS tarzan.fi.upm.es.  
datsi.fi.upm.es. 86400 IN NS zape.fi.upm.es.
```

Ejemplo hipotético

- Empresa con sede central y tres departamentos
 - Un administrador gestiona sede central, dep1 y dep2
 - Dep3 con administrador propio (y servidor de correo propio)
- Detalles de servidores de nombres de cada dominio:
 - Sede central de la empresa (emp.es.): 2 s. de nombres
 - maestro en dominio (ns.emp.es.); esclavo externo (ns.isp.com.)
 - Dep1 (dep1.emp.es.): subdominio no delegado
 - Dep2 (dep2.emp.es.): subdominio delegado a mismos servidores
 - Dep3 (dep3.emp.es.): 3 s. de nombres
 - Maestro (ns1.dep3.emp.es.); esclavo interno (ns2) y externo (ns.isp.com.)
- Empresa tiene asignada red clase B 138.99.0.0
 - 138.99.1 central; 138.99.2 dep1; 138.99.3 dep2; 138.99.4 dep3
 - Sólo está delegado subdominio de 138.99.4

Jerarquía de zonas ej. hipotético



F. zona emp.es. (ns.emp.es.)

emp.es.		IN	SOA	ns.emp.es.
emp.es.	86400	IN	NS	ns.emp.es.; servidor maestro del dominio
emp.es.	86400	IN	NS	ns.isp.com.; servidor esclavo externo
; dep2 delegado a mismos servidores				
dep2.emp.es.	86400	IN	NS	ns.emp.es.; servidor maestro en el dominio padre
dep2.emp.es.	86400	IN	NS	ns.isp.com.; servidor esclavo externo
; dep3 delegado a servidor maestro en el subdominio				
dep3.emp.es.	86400	IN	NS	ns1.dep3.emp.es.; servidor maestro en su propio subdominio
dep3.emp.es.	86400	IN	NS	ns2.dep3.emp.es.; servidor esclavo en su propio subdominio
dep3.emp.es.	86400	IN	NS	ns.isp.com.; servidor esclavo externo
; Máquinas en el dominio de la empresa				
emp.es.	86400	IN	MX	10 mail1.emp.es.; servidor de correo preferente para la empresa
emp.es.	86400	IN	MX	20 mail2.emp.es.; servidor de correo de reserva para la empresa
dep1.emp.es.	86400	IN	MX	10 mail1.emp.es.; servidor de correo preferente para dep1
dep1.emp.es.	86400	IN	MX	20 mail2.emp.es.; servidor de correo de reserva para dep1
; Máquinas en el dominio de la empresa				
ns.emp.es.	86400	IN	A	138.99.1.1
mail1.emp.es.	86400	IN	A	138.99.1.2
mail2.emp.es.	86400	IN	A	138.99.1.3
www.emp.es.	86400	IN	A	138.99.1.4
www.dep1.emp.es.	86400	IN	A	138.99.2.1; RR de subdominio no delegado en misma zona
; Glue records para subdominio dep3				
ns1.dep3.emp.es.	86400	IN	A	138.99.4.1
ns2.dep3.emp.es.	86400	IN	A	138.99.4.2

F. zona dep2.emp.es. (ns.emp.es.)

dep2.emp.es.		IN	SOA	ns.emp.es.
dep2.emp.es.	86400	IN	NS	ns.emp.es.; servidor maestro del dominio (=padre)
dep2.emp.es.	86400	IN	NS	ns.isp.com.; servidor esclavo externo
; Correo				
dep2.emp.es.	86400	IN	MX	10 mail1.emp.es.; s. correo preferente para dep2
dep2.emp.es.	86400	IN	MX	20 mail2.emp.es.; s. correo de reserva para dep2
; Máquinas en el dominio de dep2				
www.dep2.emp.es.	86400	IN	A	138.99.3.1
backup.dep2.emp.es.	86400	IN	CNAME	www.dep2.emp.es.

F. zona dep3.emp.es. (ns1.dep3.emp.es.)

dep3.emp.es.		IN	SOA	ns1.dep3.emp.es.
dep3.emp.es.	86400	IN	NS	ns1.dep3.emp.es.; servidor maestro del dominio (!=padre)
dep3.emp.es.	86400	IN	NS	ns2.dep3.emp.es.; servidor esclavo en el propio dominio
dep3.emp.es.	86400	IN	NS	ns.isp.com.; servidor esclavo externo
; Correo				
dep3.emp.es.	86400	IN	MX	10 mail.dep3.emp.es.; s. correo preferente para dep3
dep3.emp.es.	86400	IN	MX	20 mail2.emp.es.; s. correo de reserva para dep3
; Máquinas en el dominio de dep3				
ns1.dep3.emp.es.	86400	IN	A	138.99.4.1
ns2.dep3.emp.es.	86400	IN	A	138.99.4.2
mail.dep3.emp.es.	86400	IN	A	138.99.4.3
www.dep3.emp.es.	120	IN	A	138.99.4.4; reparto de carga en servicio web
www.dep3.emp.es.	120	IN	A	138.99.4.5; reparto de carga en servicio web

Servidores de nombres raíces

- Hay "13" servidores de dominio raíz (.) replicados
 - Desde *a.root-servers.net* hasta *m.root-servers.net*
 - "13" porque esa información cabe en paquete UDP
 - DNS usa UDP (53); y sólo TCP(53) cuando tamaño lo aconseja
 - ¿Problemas de escalabilidad?
 - Detrás de cada uno hay múltiples servidores (uso de *anycast*)
 - Incluyen NS y *glue records* de dominios de nivel 1º (TLDs)
 - Aunque también gestionan algunos dominios primer nivel → zona *arpa*.
 - Cada serv. DNS tiene dir. de servidores raíz (fichero *root.servers*)
 - Se debe actualizar periódicamente
- Lista y localización: <http://root-servers.org>

Servidores DNS

- Servidor gestiona (*authoritative*) $N (\geq 0)$ zonas directas/inversas
 - De algunas puede ser maestro de otras esclavo
- Servidor DNS puede tener doble rol (algo confuso):
 - Proporciona acceso a sus zonas
 - Puede actuar como cliente en navegación recursiva
- Servidor debe ofrecer navegación iterativa; recursiva opcional
 - Si no ofrece recursiva (no caché) → sólo acceso a sus zonas
 - Recursiva (caché) → por seguridad sólo peticiones de ciertas máquinas
- Ejemplos de diversos tipos de servidores de nombres:
 - Servidor sólo caché (*nonauthoritative*), recursivo para clientes internos
 - Sirve a clientes de una organización actuando de proxy
 - Servidor *authoritative* y recursivo sólo para clientes internos
 - Sirve a clientes de una organiz. y da acceso a sus zonas a todo el mundo
 - Servidor *authoritative* no recursivo → p.e. servidor raíz o de TLD

Resolver

- Parte cliente de DNS: da servicio a aplicaciones en un nodo
- Implementado habitualmente como biblioteca (\subset en *libc*)
- Proporciona API para traducción directa e inversa:
 - UNIX: *gethostbyname/getaddrinfo* y *gethostbyaddr/getnameinfo*
- Configurado con servidores de nombres a los que consulta
 - Requiere también las direcciones IP de todos esos s. de nombres
 - En UNIX: */etc/resolv.conf*
- Esos servidores de nombres deben ser recursivos
 - *Resolver* no sabe navegar
- Puede usar caché (las aplicaciones también)
- UNIX permite configurar mecanismo de traducción de *hosts*
 - */etc/hosts*, DNS, NIS, LDAP (*/etc/nsswitch.conf*)

Resolución de consultas

- Servidor *S* recibe una consulta *C* de *N*:
 - Compara con RRs de todas sus zonas y de su caché (si usa)
 - Selecciona mejor encaje → RR (*RRX*) que sea sufijo más largo de *C*
 - Si encaje completo → envía a *N* consulta resuelta
 - La marca como *authoritative* si no proviene de la caché
 - Si varios RRs satisfacen consulta, se envía a *N* lista con todos
 - Servidor rota la lista cada vez (*Round-robin DNS*) para reparto de carga
 - Se incluye información adicional para agilizar la operación
 - P.e. consulta MX puede retomar los RRs de tipo A de servidores de correo
 - Si encaje no completo, *RRX* → NSs de dominio por donde continuar
 - En el peor caso, los NSs de servidores raíz
 - Si op. recursiva: *S* envía consulta a uno de los NSs encontrados
 - Si op. no recursiva: *S* envía a *N* los RRs de los NSs encontrados
 - Si *S* conoce direcciones de NSs encontrados
 - Si no recursiva: las incluye como info. adicional en mens. de respuesta a *N*
 - Si recursiva: *S* las usa para contactar; sino tiene que obtenerlas

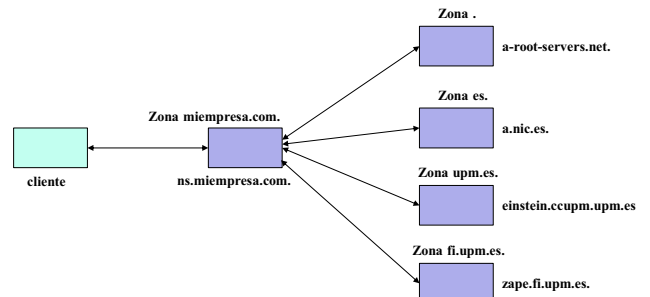
Ejemplos reales de traducción

- Operación de traducción directa: *www.fi.upm.es*.
- Operación de traducción inversa: *138.100.243.10*.
- *Resolver* tiene configurado como SN (3 opciones):
 - SN1: *ns.miempresa.com.*; IP 139.100.1.1
 - Zonas gestionadas: *miempresa.com.* y *100.139.in-addr.arpa.*, ...
 - SN2: *einstein.ccupm.upm.es.*; IP 138.100.4.8
 - Zonas gestionadas: *upm.es.* y *100.138.in-addr.arpa.*, ...
 - SN3: *zape.fi.upm.es.*; IP 138.100.8.1
 - Zonas gestionadas: *fi.upm.es.* y *243.100.138.in-addr.arpa.*, ...
- Supuestos:
 - traducción recursiva resolver-SN e iterativa desde SN
 - cachés vacías
- Test *online* de traducciones:
 - <https://dnsquery.org/>

Traducción directa usando SN1

- Aplicación llama a *gethostbyname("www.fi.upm.es.")* de *resolver*
- *Resolver* envía petición DNS de tipo A a dir. de SN1: 139.100.1.1
- SN1 mejor encaje: . → elige un s. raíz: *a.root-servers.net.* (198.41.0.4)
 - *a.root-servers.net.* mejor encaje: *es.* → envía a SN1 los NSs de *es.*
 - Y sus *glue records* como información adicional
- SN1 elige *a.nic.es.* (194.69.254.1)
 - *a.nic.es.* mejor encaje: *upm.es.* → envía a SN1 los NSs de *upm.es.*
 - Y sus *glue records* como información adicional
- SN1 elige *einstein.ccupm.upm.es.* (138.100.4.8)
 - *einstein.ccupm.upm.es.* mejor encaje: *fi.upm.es.*
 - envía a SN1 los NSs de *fi.upm.es.* y sus *glue records*
- SN1 elige *zape.fi.upm.es.* (138.100.8.1)
 - *zape.fi.upm.es.* Encaje completo: *www.fi.upm.es.*
 - envía a SN1 el NS de tipo A → *www.fi.upm.es.* | 138.100.243.10
 - SN1 se lo envía al *resolver* y éste retoma la IP a la aplicación

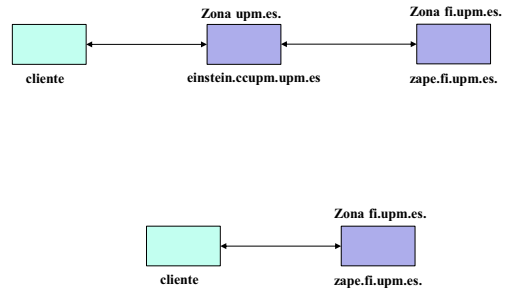
Traducción directa con SN1



Traducción directa usando SN2 y SN3

- Aplicación llama a `gethostbyname("www.fi.upm.es.")` de `resolver`
- `Resolver` envía petición DNS de tipo A a dir. de SN2: 139.100.4.8
- SN2 mejor encaje: `fi.upm.es.` → elige `zape.fi.upm.es.` (138.100.8.1)
 - `zape.fi.upm.es.` encaje completo: `www.fi.upm.es.`
 - envía a SN2 el NS de tipo A → `www.fi.upm.es.` (138.100.243.10)
 - SN2 se lo envía al `resolver` y éste retorna la IP a la aplicación
- Aplicación llama a `gethostbyname("www.fi.upm.es.")` de `resolver`.
- `Resolver` envía petición DNS de tipo A a dir. de SN3: 138.100.8.1
- SN3 encaje completo: `www.fi.upm.es.` (138.100.243.10)
 - SN3 se lo envía al `resolver` y éste retorna la IP a la aplicación
- ¿Y si hubiera elegido `galileo` en vez de `einstein`?
- Un paso menos en la traducción ya que es secundario de zona `fi`
- ¿Y si hay que traducir `www.datsi.fi.upm.es.`?
- Mismos pasos ya que `zape` gestiona también zona `datsi`

Traducción directa con SN2 y SN3



Mantenimiento de réplicas

- Sincronización de secundario con primario
 - Sigue esquema *pull*: esclavo pide información de zona a maestro
 - Periódicamente (tal como lo especifica SOA)
 - O cuando maestro avisa de cambios (*NOTIFY*)
 - Si cambio: transferencia zona completa (*AXFR*) o incremental (*IXFR*)
 - Sólo se debe permitir transferencia de zona entre maestro y esclavos
- PAEL: no asegura consistencia en ningún escenario
 - Red partida, secundario aislado sigue sirviendo peticiones
 - Valores obsoletos durante un tiempo limitado (tal como especifica SOA)
 - Red normal, actualización primario pero secundario sirve peticiones
 - Valores obsoletos hasta sincronización de secundario con primario
 - Además, cachés de clientes y de SNs sin consistencia
 - Valores obsoletos durante tiempo limitado (tal como especifica TTL)

Actualización de DNS

- Por defecto (y en RFC original), administración manual local
 - Editar fichero zona
 - Incrementando nº en SOA
 - Avisando a proceso maestro para que relea fichero de zona
 - p.e. enviándole una señal
- *Dynamic DNS*
 - Protocolo DNS incluye ops. para actualizar zona
 - Añadir, modificar y borrar RR pero no crear nuevas zonas
 - Mucho más flexible pero menos seguro
 - Algunas aplicaciones:
 - Permitir que máquinas mantengan mismo nombre en sistemas DHCP
 - Servidor elige cualquier puerto y usa SRV (requerido *Active Directory*)

LDAP: Objetos y clases

- Entidad → Objeto (entrada) en LDAP
 - Orientado a objetos: Objeto ∈ Clase (atributo *objectClass*)
- Clase define conjunto de atributos del objeto
 - Tipo del atributo | obligatorio(ob) u optativo(op) | valor único o múltiple
- Herencia: clases forman una jerarquía (*top* raíz de jerarquía)
 - Clase derivada hereda atributos de superclases
- Tipos de clases:
 - Abstracta (AB): no pueden definirse objetos de esa clase (p.e. *top*)
 - Estructural (ES): Objeto ∈ Una y solo una clase estructural
 - No puede cambiar la clase estructural de un objeto
 - Auxiliar (AU): Objeto puede estar asociado a varias clases auxiliares
 - Pueden añadirse dinámicamente: Facilitan extensión de objetos
 - Superclase(ES)=ES|AB; Superclase(AU)=AU|AB

Ejemplos de clases

- *top*: raíz; AB; ob: *objectClass*
- *person*: ↓*top*; ES; ob: *cn, sn*; op: *telephoneNumber, ...*
- *residentialPerson*: ↓*person*; ES; ob: *l*; op: *postalAddress, ...*
- *organization*: ↓*top*; ES; ob: *o*; op: *postalAddress, ...*
- *organizationalUnit*: ↓*top*; ES; ob: *ou*; op: *postalAddress, ...*
- *dcObject*: ↓*top*; AU; ob: *dc* (valor único)
- *device*: ↓*top*; ES; ob: *cn*; op: *serialNumber, o, ou, owner, ...*
- *groupOfNames*: ↓*top*; ES; ob: *cn, member*; op: *o, ou, ...*
- *alias*: ↓*top*; ES; ob: *aliasedObjectName*
- *referral*: ↓*top*; ES; ob: *ref*

Extracto de mi entrada en LDAP de FI

Formato de texto LDIF (LDAP Data Interchange Format): protocolo LDAP es binario

```

objectClass: inetOrgPerson ← estructural (top→person→organizationalPerson→inetOrgPerson)
objectClass: posixAccount ← auxiliar (top→posixAccount)
objectClass: fiEmployee ← auxiliar (top→irisPerson→fiPerson→fiEmployee)
objectClass: sambaSamAccount ← auxiliar (top→sambaSamAccount)
cn: Fernando Perez Costoya
cn: F. P. Costoya
sn: Perez Costoya
telephoneNumber: 913367377
mail: fperez@fi.upm.es
uid: fperez
-----
uidNumber: .....
gidNumber: .....
irisUserStatus: Activo
fiRelationship: pdi
fiTeaching: .....
sambaSID: .....
  
```

Diagram showing inheritance: *inetOrgPerson* (parent) includes *person* and *organizationalPerson* (children), and *inetOrgPerson*, *posixAccount*, and *fiEmployee* (children) include *fiPerson*. *fiPerson* includes *irisPerson* and *sambaSamAccount*.

Extracto de entrada FI en LDAP de FI

```

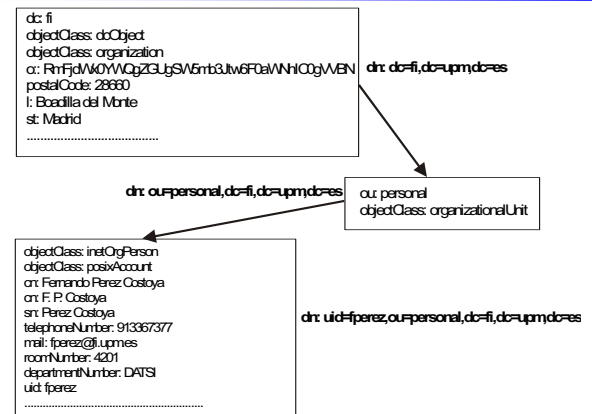
objectClass: dcObject ← auxiliar (top → dcObject)
objectClass: organization ← estructural (top → organization)
objectClass: labeledURIObject ← auxiliar (top → labeledURIObject)
dc: fi
o: RmFjdWx0YWQgZGUgSW5mb3Jt6F0aWNhC0gVVEN
postalCode: 28660
l: Boadilla del Monte
st: Madrid
labeledURI: http://www.fi.upm.es ← atributo específico de labeledURIObject
telephoneNumber: +34 913367399
  
```

Decodificación de base 64
o: Facultad de Informática – UPM

Modelo de nombres

- Entrada tiene un nombre: *Relative Distinguished Name (RDN)*
 - 1 o más atributos de la entrada que la hacen única entre "hermanos"
- *uid=fperez* (ej. múltiples: *cn=Fernando Perez Costoya+dni=76543210*)
- Jerarquía de nombres (*Directory Information Tree, DIT*)
 - Nombre completo (*path*): *Distinguished Name (DN)*
 - RDN de la entrada + DN del padre (separados por comas)
 - *dn: uid=fperez,ou=personal,dc=fi,dc=upm,dc=es*
 - No confundir con jerarquía de clases
 - Similar a SF pero directorios también tienen información asociada
 - Nombre del objeto raíz (sufijo o base): a discreción
 - Convenio: a partir de dominio DNS usando clase auxiliar *dcObject*
 - Dominio: *fi.upm.es* → *dn: dc=fi,dc=upm,dc=es*
 - Servidor LDAP gestiona 1 ó más DIT
 - Servidor devuelve metainformación en objetos/atrib. operacionales
 - DIT gestionados por el servidor, esquemas soportados, ...

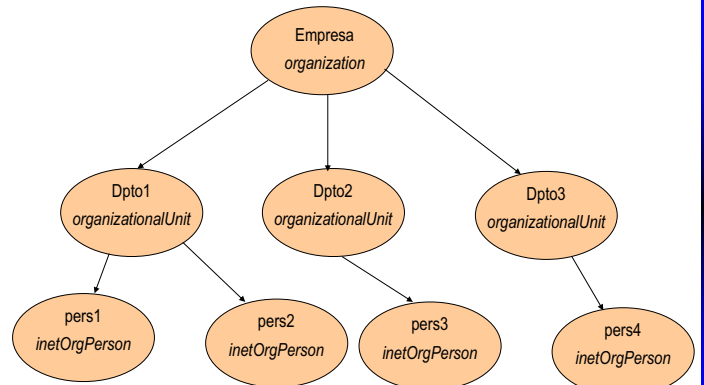
Extracto de rama del DIT del LDAP de FI



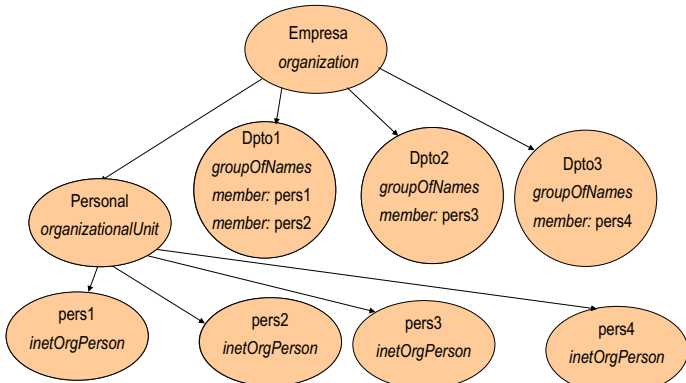
Diseño del DIT

- No trivial: requiere experiencia
- Análisis previo de info. del SD y cómo evolucionará
 - Diseño debería evitar que cambios previstos en info. modifiquen DIT
 - Cambio debería afectar a atributos en vez de a estructura de DIT
 - Mejor árbol poco profundo
- Ej.: empresa donde personal cambia de dpto. con frecuencia
 - Diseño 1
 - 1 *organizationalUnit*/dpto. + 1 *inetOrgPerson*/persona
 - Entrada de persona hija de entrada de su departamento
 - Diseño 2
 - 1 *organizationalUnit* para todo el personal + 1 *inetOrgPerson*/persona
 - 1 *groupOfNames*/dpto. con 1 atributo *member*/persona
 - Persona cambia de departamento: cambio atributos, no cambio DIT
 - Aunque ciertas búsquedas pueden ralentizarse

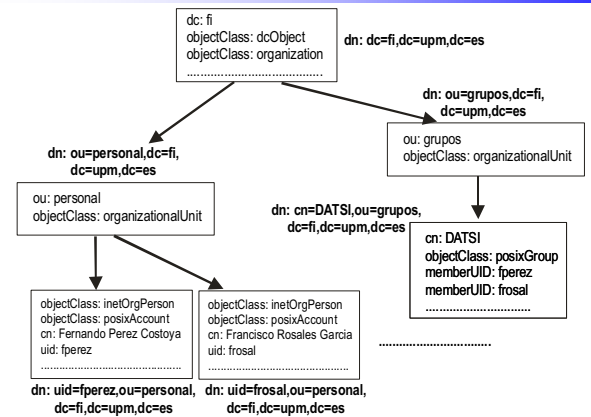
Diseño 1



Diseño 2



Extracto de jerarquía de LDAP de FI



Operaciones de LDAP

- **Bind/Unbind:** conecta y autentica/desconecta
- **Search:** realiza una búsqueda basada en los parámetros:
 - DN base de la búsqueda
 - Ámbito: Sólo la entrada base, sólo hijos o todo el sub-árbol
 - Filtro de búsqueda
 - Atributos que se devuelven (además, si valores o sólo tipos)
 - Si se siguen los alias o no durante la búsqueda
 - Límite de tiempo y máximo nº de entradas retornadas
- **Compare:** comprueba si DN dado tiene un valor en atributo
- **Add/Delete:** Añade/Elimina la entrada del DN dado
- **Modify:** Modifica atributos (añade, elimina o cambia) de un DN
- **Modify DN:** Cambia DN de una entrada
 - Renombra si sólo cambia RDN final; mueve en DIT en caso contrario

Linda: Operaciones sobre tuplas

- out(tupla):** Añade *tupla* al espacio de tuplas; no bloqueante
- in(patrn):** Busca tupla que encaje en *patrón* y la extrae si varias → cualquiera; si ninguna → se bloquea
Ejemplo: *in("hola", ?, ?, "adios")* → ("hola", 7, 3.14, "adios")
inp: versión no bloqueante
- rd(patrn):** Igual que *in* pero la tupla no se extrae
rdp: versión no bloqueante
- eval(tupla):** Añade *tupla* de proceso al espacio de tuplas
Al insertar se activa proceso que la convertirá en tupla de datos
- Extensión a propuesta original:**
 - Múltiples espacio de tuplas
 - Con operaciones para mover/copiar tuplas entre espacios

Linda: Ejemplos de uso

Semáforo:

Bajar semáforo: *in("sem")*
Subir semáforo: *out("sem")*

Cliente/servidor:

```

servidor() {
  int indice=1;
  out("turno", indice);
  while(1) {
    in("peticion", indice, ?pet);
    .....
    out("respuesta", indice++, res);
  }
}
  
```

```

cliente(){
  int indice;
  .....
  in("turno", ?indice);
  out("turno", indice+1);
  out("peticion", indice, pet);
  in("respuesta", indice, ?res);
}
  
```

Ejemplos vectores y matrices en Linda

Iniciar matriz "M" (MxN) con diagonal a 1 y resto a 0

```

for (i=1; i<=M; i++)
  for (j=1; j<=N; j++)
    out("M", i, j, (i=j));
  
```

Cálculo vector "V" suma de filas de matriz "M" (MxN)

```

for (i=1; i<=M; i++) {
  suma = 0;
  for (j=1; j<=N; j++) {
    rd("M", i, j, ?v);
    suma += v;
  }
  out("V", i, suma);
}
  
```


Ejercicio de servicio de nombres y DSM

Abril del 2016. Grupo de mañana.

Ejercicio de DNS

Considere el servicio DNS de una nueva compañía (dominio `co.es.`) organizada en sucursales y departamentos dentro de cada sucursal. En esta etapa inicial la compañía sólo tiene una sucursal (dominio `su.co.es.`) con un único departamento (dominio `de.su.co.es.`). En la figura que aparece a continuación se muestran extractos de los ficheros de zona de `co.es.` y de `su.co.es.`, no apareciendo el correspondiente a `de.su.co.es.`, que usará los mismos servidores de nombres que el dominio `su.co.es.`. Téngase en cuenta que en las cuestiones sobre búsquedas que se plantean en el ejercicio se supondrá que la búsqueda es recursiva entre cliente y servidor pero iterativa entre servidores, así como que todas las cachés de DNS están vacías.

```
co.es. SOA n1.co.es. ... su.co.es. SOA n1.su.co.es. ...
co.es. NS n1.co.es. su.co.es. NS n1.su.co.es.
co.es. NS n2.co.es. su.co.es. NS n2.su.co.es.
co.es. NS n3.co.es. su.co.es. NS n3.co.es.
co.es. NS n4.co.es. n1.su.co.es. A ...
co.es. NS n1.co.es. n2.su.co.es. A ...
n1.co.es. A ... su.co.es. MX 20 n1.su.co.es.
n2.co.es. A ... su.co.es. MX 30 n2.su.co.es.
n3.co.es. A ... _sip._tcp.su.co.es SRV 1 X 8000 n1.su.co.es.
n4.co.es. A ... _sip._tcp.su.co.es SRV 1 Y 8000 n2.su.co.es.
                _sip._tcp.su.co.es SRV 2 Z 8000 n3.co.es.
```

1. ¿Cuántos RR de tipo NS adicionales hay que incluir en la zona `co.es.`?

- a. 3
- b. 0
- c. 2
- d. 4

Explicación

Tantos como servidores de nombres tenga el dominio hijo (`su.co.es.`):

```
su.co.es. NS n1.su.co.es.
su.co.es. NS n2.su.co.es.
su.co.es. NS n3.co.es.
```

2. ¿Cuántos *glue records* hay que incluir en la zona `co.es.`?

- a. 2
- b. 3
- c. 0
- d. 4

Explicación

Hay que incluir un *glue record* en el dominio padre por cada servidor de nombres del dominio hijo que pertenezca a ese subdominio:

```
n1.su.co.es. A ...
n2.su.co.es. A ...
```

3. ¿Cuántos *glue records* hay que incluir en la zona `su.co.es.`?

- a. 0
- b. 3
- c. 2
- d. 4

Explicación

Ninguno, puesto que el dominio hijo usa los mismos servidores de nombres (no utiliza un servidor de nombres que esté dentro del propio subdominio).

4. ¿Cuántos *glue records* hay que incluir en la zona `de.su.co.es.`?

- a. 0
- b. 3
- c. 2
- d. 4

Explicación

Ninguno, puesto que no tiene dominios hijos.

5. Suponiendo que se tiene configurado como servidor de nombres (SN) `n1.su.co.es.`, ¿con cuántos servidores tendría que contactar SN para traducir `www.co.es.`?

- a. 3

- b. 2
- c. 1
- d. 0

Explicación

Paso a paso:

1. SN compara la dirección a traducir con sus RR y no encuentra ningún encaje por lo que tiene que contactar con un servidor raíz, que le retorna las direcciones de los servidores de `.es.`.
2. SN contacta con un servidor de `.es.`, que le envía las direcciones de los servidores de `co.es.`.
3. SN contacta con un servidor de `co.es.` que completa la traducción devolviendo la dirección de `www.co.es.`.

6. Suponiendo que se tiene configurado como servidor de nombres (SN) `chita.fi.upm.es.`, ¿con cuántos servidores tendría que contactar SN para traducir `www.de.su.co.es.` si siempre se elige el servidor primario de cada dominio?

- a. 4
- b. 5
- c. 3
- d. 2

Explicación

Paso a paso:

1. El servidor `chita.fi.upm.es` gestiona el dominio de la FI. Al recibir la petición de `www.de.su.co.es.`, no encuentra ningún encaje teniendo que contactar directamente con el servidor primario raíz, que le retorna las direcciones de los servidores de `.es.`.
2. SN contacta con el servidor primario de `.es.`, que le envía las direcciones de los servidores de `co.es.`.
3. SN contacta con el servidor primario de `co.es.` (`n1.co.es.`), que le retorna las direcciones de los servidores de `su.co.es.`.
4. SN contacta con el servidor primario de `su.co.es.` (`n1.su.co.es.`), que también gestiona la zona asociada a `de.su.co.es.`, con lo que completa la traducción devolviendo la dirección de `www.de.su.co.es.`.

7. Suponiendo que se tiene configurado como servidor de nombres (SN) `chita.fi.upm.es.`, ¿con cuántos servidores tendría que contactar SN para traducir `www.su.co.es.` si siempre se elige el tercer servidor de nombres encontrado?

- a. 3
- b. 2
- c. 4
- d. 1

Explicación

Paso a paso:

1. El servidor `chita.fi.upm.es` gestiona el dominio de la FI. Al recibir la petición de `www.su.co.es.`, no encuentra ningún encaje teniendo que contactar directamente con el tercer servidor raíz, que le retorna las direcciones de los servidores de `.es.`.
2. SN contacta con el tercer servidor de `.es.`, que le envía las direcciones de los servidores de `co.es.`.
3. SN contacta con el tercer servidor de `co.es.` (`n3.co.es.`), que también gestiona el dominio `d1.co.es.`, con lo que completa la traducción devolviendo la dirección de `www.su.co.es.`.

8. Suponiendo que se tiene configurado como servidor de nombres (SN) `n1.co.es.`, ¿con cuántos servidores tendría que contactar SN para traducir `www.de.su.co.es.` si siempre se elige el servidor primario de cada dominio?

- a. 1
- b. 0
- c. 5
- d. 2

Explicación

Paso a paso:

1. SN compara la dirección a traducir con sus RR y encuentra como mejor encaje los NS `su.co.es.` y contacta con el servidor primario de ese dominio `n1.su.co.es.`. Este servidor gestiona también la zona asociada al dominio `de.su.co.es.`, por lo que completa la traducción devolviendo la dirección de `www.de.su.co.es.`.

9. Suponiendo que todas las máquinas están funcionando, ¿qué valores deberían tener X, Y y Z para que `n1.su.co.es.` se lleve aproximadamente un tercio del tráfico del servicio SIP especificado mediante SRVs?

- a. 1 2 3
- b. 1 1 1
- c. 3 2 1
- d. 33 33 34

Explicación

Al estar todas las máquinas en funcionamiento, las dos que tienen mayor prioridad (1), reciben todo el tráfico repartiéndolo según su peso. Por tanto, para conseguir que la primera máquina reciba un tercio, su peso debe ser la mitad que el de la segunda (por ejemplo: $X=1$ e $Y=2$), siendo intrascendente el valor de Z.

10. ¿A qué máquina se enviaría el correo dirigido al dominio de la empresa (`usuario@co.es`): `n1.su.co.es.` o `n2.co.es.`?

- a. Daría error
- b. A la primera.
- c. A la segunda.
- d. A ambas.

Explicación

Dado que no hay registros MX en el dominio `co.es` (están incluidos en `su.co.es`), no se podrán enviar mensajes de correo a ese dominio.

Los extractos actualizados de las tres zonas planteadas serían:

Zona `co.es`

```
co.es. SOA n1.co.es. ...
co.es. NS n1.co.es.
co.es. NS n2.co.es.
co.es. NS n3.co.es.
co.es. NS n4.co.es.
n1.co.es. A ...
n2.co.es. A ...
n3.co.es. A ...
n4.co.es. A ...
su.co.es. NS n1.su.co.es.
su.co.es. NS n2.su.co.es.
su.co.es. NS n3.co.es.
n1.su.co.es. A ...
n2.su.co.es. A ...
```

Zona `su.co.es`

```
su.co.es. SOA n1.su.co.es. ...
su.co.es. NS n1.su.co.es.
su.co.es. NS n2.su.co.es.
su.co.es. NS n3.co.es.
n1.su.co.es. A ...
n2.su.co.es. A ...
su.co.es. MX 20 n1.su.co.es.
su.co.es. MX 30 n2.co.es.
_sip._tcp.su.co.es. SRV 1 X 8000 n1.su.co.es.
_sip._tcp.su.co.es. SRV 1 Y 8000 n2.su.co.es.
_sip._tcp.su.co.es. SRV 2 Z 8000 n3.co.es.
de.su.co.es. NS n1.su.co.es.
de.su.co.es. NS n2.su.co.es.
de.su.co.es. NS n3.co.es.
```

Zona `de.su.co.es`

```
de.su.co.es. SOA n1.su.co.es. ...
de.su.co.es. NS n1.su.co.es.
de.su.co.es. NS n2.su.co.es.
de.su.co.es. NS n3.co.es.
```

Ejercicio de LDAP

El servicio LDAP de una escuela está organizado de manera que los alumnos cuelgan de un único nodo de tipo OU, tal que cada alumno es un `inetOrgPerson` al que se le ha añadido mediante una clase auxiliar la información de qué titulaciones cursa (un alumno cursa al menos una). Además, se usa un `groupOfNames` por cada titulación para saber qué alumnos la cursan.

11. ¿Qué operaciones LDAP requiere el ingreso de un nuevo alumno que se matricula en una cierta titulación?
- Add + Modify**
 - Add
 - Add + ModifyDN
 - Modify

Explicación

En primer lugar, hay que usar un operación `Add` para añadirle al nodo del que dependen todos los alumnos. A continuación, hay que usar operaciones `Modify` para incluirle como miembro del `groupOfNames` correspondiente a la titulación, así como para añadir a su propio nodo (`inetOrgPerson`) esta información usando una clase auxiliar.

12. ¿Qué operaciones LDAP requiere la matriculación de un alumno en una titulación adicional?
- Modify**
 - Add
 - Add + ModifyDN
 - Add + Modify

Explicación

Hay que usar operaciones `Modify` para incluirle como miembro del `groupOfNames` correspondiente a esa titulación, así como para añadir a su propio nodo esta información.

13. ¿Qué operaciones LDAP requiere la baja de un alumno en una titulación manteniéndose en las restantes?
- Modify**
 - Delete
 - ModifyDN
 - Modify + Delete

Explicación

Hay que usar operaciones `Modify` para eliminarle como miembro del `groupOfNames` correspondiente a esa titulación, así como para quitar de su propio nodo esta información.

14. Se pretende buscar toda la información de los alumnos de una determinada titulación pero sin usar los grupos y utilizando una única operación de búsqueda. ¿Qué ámbito mínimo debería usar esa búsqueda? ¿Necesitaría utilizar un filtro?
- one; sí**
 - one; no
 - default; sí
 - default; no

Explicación

Esta búsqueda debe establecer como base el nodo del que dependen todos los alumnos y visitar sólo los hijos (`one`) de ese nodo especificando como filtro que el alumno curse esa titulación.

15. Se pretende buscar el número de teléfono de los alumnos de una determinada titulación usando los grupos y utilizando múltiples operaciones de búsqueda para minimizar el número de nodos visitados. ¿Qué ámbito mínimo debería usar cada una de esas operaciones? ¿Necesitaría utilizar un filtro?
- base; no**

- base; sí
- one; sí
- one; no

Explicación

Habrà, en primer lugar, una búsqueda de tipo `base` sin filtro para obtener del `groupOfNames` de esa titulación el atributo donde se almacenan todos los miembros. A continuación, por cada miembro, hay que realizar búsqueda de tipo `base` sin filtro para obtener del `inetOrgPerson` de ese alumno el atributo donde se almacenan sus teléfonos. Cada búsqueda debe establecer como base el nodo del que dependen todos los alumnos y visitar sólo los hijos (`one`) de ese nodo especificando como filtro que el alumno curse esa titulación.

Ejercicio de espacios de tuplas

Se pretende implementar en Linda un esquema cliente-servidor con múltiples clientes y servidores, tal que los servidores procesan en paralelo pero de forma ordenada las peticiones.

```
void iniciar() {
    out("turno", 1);
    out("trabajo", 1);
}

void cliente_i () {
    while (1) {
        int p; D dato=genera(); R resul;
        OP1
        out("turno", p+1);
        out("pet", p, dato);
        OP2; consume(resul); } }

void servidor_j () {
    while (1) {
        int p; D dato; R resul;
        OP3
        out("trabajo", p+1);
        OP4; resul=procesar(dato);
        out("resp", p, resul); } }
```

16. ¿Con qué operación de Linda se corresponde OP1?
- in("turno", ?p)**
 - in("turno", p)
 - rd("turno", p)
 - rd("turno", ?p)

Explicación

Esta operación obtiene el valor actual del `turno` para incrementarlo justo en la siguiente sentencia. Es necesaria, por tanto, una operación de extracción usando `?`.

17. ¿Con qué operación de Linda se corresponde OP3?
- in("trabajo", ?p)**
 - in("trabajo", p)
 - rd("trabajo", p)
 - rd("trabajo", ?p)

Explicación

Esta operación obtiene el valor actual del número de `trabajo` para incrementarlo justo en la siguiente sentencia. Es necesaria, por tanto, una operación de extracción usando `?`.

18. ¿Con qué operación de Linda se corresponde OP2?
- in("resp", p, ?resul)**
 - in("resp", ?p, ?resul)
 - rd("resp", p, ?resul)
 - rd("resp", ?p, ?resul)

Explicación

Esta operación intenta obtener el resultado de una determinada petición. Por tanto, debe usar `in` para extraerlo del espacio de tuplas y `p` para identificar la petición.

19. ¿Con qué operación de Linda se corresponde OP4?
- in("pet", p, ?dato)**
 - in("pet", ?p, ?dato)
 - rd("pet", p, ?dato)
 - rd("pet", ?p, ?dato)

Explicación

Esta operación intenta obtener la siguiente petición pendiente de procesar según determina el valor de `trabajo`. Por tanto, debe usar `in` para extraerla del espacio de tuplas y `p` para identificar esa petición.

20. Suponga que se han procesado 2 peticiones y hay otras 2 pendientes de procesar. ¿Qué valor estaría asociado a `turno`?
- 5**
 - 4
 - 3
 - 2

Explicación

El valor de `turno` establece cuál será el identificador de la próxima petición (inicialmente, 1). Si ya se han procesado 2 peticiones y hay otras dos pendientes, el valor debe ser igual a 5.

Ejercicio de servicio de nombres y DSM

Abril del 2016. Grupo de tarde.

Ejercicio de DNS

Considere el servicio DNS de una compañía (dominio `co.es.`) que tiene dos departamentos (dominios `d1.co.es.` y `d2.co.es.`). En la figura que aparece a continuación se muestran extractos de los ficheros de zona de `co.es.` y de `d1.co.es.`, no apareciendo el correspondiente a `d2.co.es.`, que usará los mismos servidores de nombres que el dominio `co.es.`. Téngase en cuenta que en las cuestiones sobre búsquedas que se plantean en el ejercicio se supondrá que la búsqueda es recursiva entre cliente y servidor pero iterativa entre servidores, así como que todas las cachés de DNS están vacías.

```
co.es. SOA n1.co.es. ...
co.es. NS n1.co.es.
co.es. NS n2.co.es.
co.es. NS n3.co.es.
n1.co.es. A ...
n2.co.es. A ...
n3.co.es. A ...
_sip._tcp.co.es SRV 1 1 8000 n1.co.es.
_sip._tcp.co.es SRV 2 1 8000 n2.co.es.
_sip._tcp.co.es SRV 2 1 8000 n3.co.es.
_sip._tcp.co.es SRV 3 1 8000 n3.d1.co.es.

d1.co.es. SOA n1.d1.co.es. ...
d1.co.es. NS n1.d1.co.es.
d1.co.es. NS n2.co.es.
d1.co.es. NS n3.d1.co.es.
n1.d1.co.es. A ...
n3.d1.co.es. A ...
n4.d1.co.es. A ...
```

1. ¿Cuántos RR de tipo NS adicionales hay que incluir en la zona `co.es.`?

- a. 7
- b. 3
- c. 4
- d. 6

Explicación

Tantos como servidores de nombres tengan sus dominios hijos (`d1.co.es.` y `d2.co.es.`):

```
d1.co.es. NS n1.d1.co.es.
d1.co.es. NS n2.co.es.
d1.co.es. NS n3.d1.co.es.
d1.co.es. NS n4.d1.co.es.
d2.co.es. NS n1.co.es.
d2.co.es. NS n2.co.es.
d2.co.es. NS n3.co.es.
```

2. ¿Cuántos *glue records* hay que incluir en la zona `co.es.`?

- a. 3
- b. 2
- c. 0
- d. 4

Explicación

Hay que incluir un *glue record* en el dominio padre por cada servidor de nombres de un dominio hijo que pertenezca a ese subdominio. Dado que el segundo departamento usa los mismos servidores que el padre no requiere ningún *glue record*. El primer departamento, sin embargo, requiere tres:

```
n1.d1.co.es. A ...
n3.d1.co.es. A ...
n4.d1.co.es. A ...
```

3. ¿Cuántos *glue records* hay que incluir en la zona `d1.co.es.`?

- a. 0
- b. 3
- c. 2
- d. 4

Explicación

Ninguno, puesto que no tiene dominios hijos.

4. ¿Cuántos *glue records* hay que incluir en la zona `d2.co.es.`?

- a. 0
- b. 3
- c. 2
- d. 4

Explicación

Ninguno, puesto que no tiene dominios hijos.

5. Suponiendo que se tiene configurado como servidor de nombres (SN) `n1.d1.co.es.`, ¿con cuántos servidores tendría que contactar SN para traducir `www.co.es.`?

- a. 3
- b. 2
- c. 1
- d. 0

Explicación

Paso a paso:

1. SN compara la dirección a traducir con sus RR y no encuentra ningún encaje por lo que tiene que contactar con un servidor raíz, que le retorna las direcciones de los servidores de `.es.`.
2. SN contacta con un servidor de `.es.`, que le envía las direcciones de los servidores de `co.es.`.
3. SN contacta con un servidor de `co.es.` que completa la traducción devolviendo la dirección de `www.co.es.`

6. Suponiendo que se tiene configurado como servidor de nombres (SN) `chita.fi.upm.es.`, ¿con cuántos servidores tendría que contactar SN para traducir `www.d2.co.es.` si siempre se elige el servidor primario de cada dominio?

- a. 3
- b. 4
- c. 1
- d. 2

Explicación

Paso a paso:

1. El servidor `chita.fi.upm.es` gestiona el dominio de la FI. Al recibir la petición de `www.d2.co.es.`, no encuentra ningún encaje teniendo que contactar directamente con el servidor primario raíz, que le retorna las direcciones de los servidores de `.es.`.
2. SN contacta con el servidor primario de `.es.`, que le envía las direcciones de los servidores de `co.es.`.
3. SN contacta con el servidor primario de `co.es.` (`n1.co.es.`), que completa la traducción puesto que también gestiona la zona del subdominio asociado a `d2`, devolviendo la dirección de `www.d2.co.es.`

7. Suponiendo que se tiene configurado como servidor de nombres (SN) `chita.fi.upm.es.`, ¿con cuántos servidores tendría que contactar SN para traducir `www.d1.co.es.` si siempre se elige el segundo servidor de nombres encontrado?

- a. 3
- b. 2
- c. 4
- d. 1

Explicación

Paso a paso:

1. El servidor `chita.fi.upm.es` gestiona el dominio de la FI. Al recibir la petición de `www.d1.co.es.`, no encuentra ningún encaje teniendo que contactar directamente con el segundo servidor raíz, que le retorna las direcciones de los servidores de `.es.`.
2. SN contacta con el segundo servidor de `.es.`, que le envía las direcciones de los servidores de `co.es.`.
3. SN contacta con el segundo servidor de `co.es.` (`n2.co.es.`), que también gestiona el dominio `d1.co.es.`, con lo que completa la traducción devolviendo la dirección de `www.d1.co.es.`

8. Suponiendo que se tiene configurado como servidor de nombres (SN) `n1.co.es.`, ¿con cuántos servidores tendría que contactar SN para traducir `www.d1.co.es.` si siempre se elige el servidor primario de cada dominio?

- a. 1
- b. 0
- c. 4
- d. 2

Explicación

Paso a paso:

1. SN compara la dirección a traducir con sus RR y encuentra como mejor encaje los NS `d1.co.es.` y contacta con el servidor primario de ese dominio `n1.d1.co.es.`, que completa la traducción devolviendo la dirección de `www.d1.co.es.`

9. Si `n1.co.es.` está caído, ¿qué porcentaje del tráfico del servicio SIP especificado mediante SRVs recibirá `n2.co.es.`?

- a. 50%
- b. 100%
- c. 25%
- d. (100/3)%

Explicación

Al estar caído el servidor de mayor prioridad (1), entran en juego los dos servidores que tienen una prioridad intermedia (2) y, dado que ambos tienen el mismo peso (1), se reparten el tráfico a partes iguales.

10. Suponga que sólo se dispone de correo en el dominio de la compañía y en el del primer departamento (`usuario@co.es` y `usuario@d1.co.es`) y que se usarán tres máquinas de `d1` para recibir el correo de ambos dominios. ¿Cuál es el número total de registros MX que hay que incluir en el sistema?

- a. 6
- b. 3
- c. 2
- d. 4

Explicación

Los registros MX hay que incluirlos en las zonas correspondientes a los dominios donde se quiere recibir correo. Por tanto, habrá que incluir tres registros MX (uno por cada máquina que gestione el correo) en el dominio `co.es` y otros tres en `d1.co.es`.

Zona <code>co.es</code>	Zona <code>d1.co.es</code>	Zona <code>d2.co.es</code>
<code>co.es. SOA n1.co.es. ...</code>	<code>d1.co.es. SOA n1.d1.co.es. ...</code>	<code>d2.co.es. SOA n1.co.es. ...</code>
<code>co.es. NS n1.co.es.</code>	<code>d1.co.es. NS n1.d1.co.es.</code>	<code>d2.co.es. NS n1.co.es.</code>
<code>co.es. NS n2.co.es.</code>	<code>d1.co.es. NS n2.co.es.</code>	<code>d2.co.es. NS n2.co.es.</code>
<code>co.es. NS n3.co.es.</code>	<code>d1.co.es. NS n3.d1.co.es.</code>	<code>d2.co.es. NS n3.co.es.</code>
<code>n1.co.es. A ...</code>	<code>d1.co.es. NS n4.d1.co.es.</code>	
<code>n2.co.es. A ...</code>	<code>n1.d1.co.es. A ...</code>	
<code>n3.co.es. A ...</code>	<code>n3.d1.co.es. A ...</code>	
<code>_sip_tcp.co.es SRV 1 1 8000 n1.co.es.</code>	<code>n4.d1.co.es. A ...</code>	
<code>_sip_tcp.co.es SRV 2 1 8000 n2.co.es.</code>		
<code>_sip_tcp.co.es SRV 2 1 8000 n3.co.es.</code>		
<code>_sip_tcp.co.es SRV 3 1 8000 n3.d1.co.es.</code>		
<code>d1.co.es. NS n1.d1.co.es.</code>		
<code>d1.co.es. NS n2.co.es.</code>		
<code>d1.co.es. NS n3.d1.co.es.</code>		
<code>d1.co.es. NS n4.d1.co.es.</code>		
<code>d2.co.es. NS n1.co.es.</code>		
<code>d2.co.es. NS n2.co.es.</code>		
<code>d2.co.es. NS n3.co.es.</code>		
<code>n1.d1.co.es. A ...</code>		
<code>n3.d1.co.es. A ...</code>		
<code>n4.d1.co.es. A ...</code>		

Ejercicio de LDAP

En el servicio LDAP de un centro de investigación hay un nodo de tipo OU para los investigadores y otro para los becarios. Además, se usan `groupOfNames` para agrupar todos los becarios que tienen el mismo tipo de beca (todo becario tiene un tipo de beca asignado) y todos los investigadores del mismo grupo de investigación (todo investigador pertenece a uno).

11. ¿Qué tipo de operaciones requiere la incorporación al centro de un becario de un cierto tipo?

- Add y Modify
- Modify y ModifyDN
- Modify
- Add y ModifyDN

Explicación

En primer lugar, hay que usar una operación `Add` para añadirle al nodo del que dependen todos los becarios. A continuación, hay que usar una operación `Modify` para incluirle como miembro del `groupOfNames` correspondiente a ese tipo de beca.

12. ¿Qué tipos de operaciones requiere el cambio de grupo de investigación?

- Modify
- Add y Modify
- Modify y ModifyDN
- Add y ModifyDN

Explicación

Hay que usar operaciones `Modify` para eliminarle como miembro del `groupOfNames` del grupo de investigación previo y, a continuación, para añadirle como miembro del `groupOfNames` de su nuevo grupo de investigación.

13. ¿Qué tipos de operaciones requiere la conversión de un becario en investigador?

- Modify y ModifyDN
- Add y Modify
- Modify
- Add y ModifyDN

Explicación

Hay que usar una operación `Modify` para eliminarle como miembro del `groupOfNames` correspondiente a su tipo de beca. Acto seguido, hay que realizar una operación `ModifyDN` para moverlo del nodo de becarios al de investigadores. Por último, hay que usar una operación `Modify` para añadirle como miembro del `groupOfNames` de su nuevo grupo de investigación.

14. Se pretende buscar la dirección de correo de todos los miembros del centro. ¿Qué ámbito mínimo debería usar esa búsqueda? ¿Necesitaría utilizar un filtro?

- default; no
- one; no
- one; sí
- default; sí

Explicación

Esa búsqueda debe establecer como base el nodo raíz de toda la organización y aplicar un ámbito que recorra todo el sub-árbol (`default`), puesto que hay que acceder a los nodos tanto de los becarios como de los investigadores para recuperar su dirección de correo. No hay que usar ninguna función de filtro ya que hay que recuperar información de todos los nodos.

15. Se pretende buscar la información de todos los investigadores de un grupo usando el `groupOfNames` correspondiente. ¿Cuál es el número mínimo de nodos que hay que acceder si, de los X investigadores existentes, Y están en ese grupo?

- Y+1
- Y
- X
- X+1

Explicación

Habrà, en primer lugar, un acceso, mediante una búsqueda de tipo `base`, al `groupOfNames` de ese grupo de investigación para obtener el atributo donde se almacenan todos los miembros (1 nodo). A continuación, por cada miembro, hay que acceder, mediante una búsqueda de tipo `base`, al `inetOrgPerson` de cada investigador de ese grupo (Y nodos).

Ejercicio de espacios de tuplas

Se pretende implementar en Linda un esquema maestro-trabajador tal que los maestros generan trabajos de tamaño variable y los trabajadores los van procesando. Cada trabajo tiene una tupla de cabecera, con el número de datos, y las tuplas con los datos, y queda identificado por el PID del maestro y el número de trabajo que ha generado ese maestro hasta el momento.

```
void maestro i () {
    while (1) {
        int nt=0; D d[]=genera();
        out("T", getpid(), ++nt, d.size);
        for (int i=0; i<d.size; i++)
            out("D", getpid(), nt, d[i]);
    }
}

void trabajador_j () {
    while (1) {
        int pid, nt, tam; D d[];
        OP1("T", p1, p2, p3);
        for (int i=0; i<tam; i++)
            OP2("D", p4, p5, p6);
        procesar(d);
    }
}
```

16. ¿Con qué operación de Linda se corresponde OP1 y qué valor tendrá p1?

- in; ?pid
- in, pid
- rd, ?pid
- rd, pid

Explicación

Esta operación extrae cualquier tupla que corresponda a la cabecera de un trabajo. Por tanto, debe usar `in` y `?pid`.

17. ¿Con qué operación de Linda se corresponde OP2 y qué valor tendrá p4?

- in; pid
- in; ?pid
- rd; ?pid
- rd; pid

Explicación

Esta operación extrae una tupla de datos de un determinado trabajo. Por tanto, debe usar `in` y `pid`.

18. ¿Qué valor tendrá p2 y p3?

- ?nt; ?tam
- nt; ?tam
- ?nt; tam
- nt; tam

Explicación

Esos parámetros corresponden a la operación que extrae cualquier tupla de cabecera de un trabajo. Por tanto, debe usar `?nt` y `?tam`.

19. ¿Qué valor tendrá p5 y p6?

- nt; ?d[i]
- ?nt; ?d[i]
- ?nt; d[i]
- nt; d[i]

Explicación

Esos parámetros corresponden a la operación que extrae una tupla de datos de un determinado trabajo. Por tanto, debe usar `nt` (con el valor correspondiente a ese trabajo) y `?d[i]` (para obtener el dato).

20. ¿En qué orden se procesan los trabajos en el sistema: (F)IFO o (I)mpredecible?, ¿Y los generados por el mismo maestro?

- I; I
- F; F
- I; F
- F; I

Explicación

Cuando hay múltiples tuplas que encajan en un patrón, Linda devuelve cualquiera de ellas. Por tanto, OP1 puede obtener cualquier trabajo de cualquier maestro, siendo el orden impredecible en ambos casos.

Ejemplos de operaciones sobre el LDAP de la FI

En este documento se muestran diversos ejemplos ([código de los ejemplos](#)) de operaciones LDAP sobre el servicio de directorio de la Facultad soportado en la máquina `info.fi.upm.es`. De las diversas opciones para acceder a este servicio, se ha optado por utilizar operaciones de línea de mandatos.

Acceso al nodo raíz de la Facultad

Empecemos *visitando* el nodo superior de la organización, que es de tipo `organization` y cuyo DN es `dc=fi,dc=upm,dc=es`. Para ello, se usará el siguiente mandato:

```
ldapsearch -LLL -x -w contraseña -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' \
-b 'dc=fi,dc=upm,dc=es' -s base
```

Dado que es la primera vez que aparece este mandato, vamos a explicar las distintas opciones especificadas en el mismo (nótese que las cuatro primeras opciones especifican aspectos generales de la operación y serán igual en todos los ejemplos):

- `-LLL`: Especifica que use el formato de salida más conciso.
- `-x`: Solicita un mecanismo de autenticación simple...
- `-w`: especificando la contraseña en la línea de mandatos...
- `-D`: y siendo éste el usuario.
- `-b`: Especifica el nodo base de la búsqueda; en este caso, el nodo raíz.
- `-s`: Especifica el ámbito de búsqueda: `base` sólo el nodo; `one` sólo los nodos hijos del nodo base; `sub` todo el sub-árbol, que es la opción por defecto.
- Nótese que no se ha especificado ninguna función de filtro ni una selección de los atributos que se pretende recuperar de cada objeto por lo que se retornarán todos los atributos de todos los objetos visitados.

A continuación, se muestra la salida del mandato:

```
dn: dc=fi,dc=upm,dc=es
dc: fi
objectClass: dcObject
objectClass: organization
objectClass: labeledURIObject
o: RmFjdWx0YWQgZGUgSW5mb3Jtw6F0aWNhIC0gVVBN
postalCode: 28660
l: Boadilla del Monte
st: Madrid
description: RmFjdWx0YWQgZGUgSW5mb3Jtw6F0aWNhICVbml2ZXJzaWRhZCBQb2xpdMOpY25p
Y2EgZGUgTWFKcmklkKQ==
labeledURI: http://www.fi.upm.es
telephoneNumber: +34 913367399
facsimileTelephoneNumber: +34 913367412
postalAddress: RmFjdWx0YWQgZGUgSW5mb3Jtw6F0aWNhICVlLlAuTS4pDQpDYWlwdXMgZGUgTW
9udGVnYW5jZWVudDQpCb2FkaWxsYSBkZWwgdW9udGUNCjI4NjYwIC0gTWFKcmklkDQpTcGFpbG==
```

Nótese que, en primer lugar, aparece el DN del objeto accedido y, a continuación, los atributos del objeto recuperado.

Si cambiamos el ámbito para que sea de tipo `-s one`, se obtendrán los nodos de segundo nivel de la organización, que son de tipo `organizationalUnit`, entre los que se encuentran el correspondiente al personal de la facultad (`personal`), a los alumnos de la misma (`alumnos`), así como un nodo a partir del cual se despliegan las distintas agrupaciones que han definido los administradores de este servicio de directorio (`grupo`).

Si especificamos `-s sub`, u omitimos ese valor, muestra todo el árbol de objetos de la Facultad.

Acceso a mi entrada y a la de un alumno *anónimo*

Los empleados de este centro, entre los que me encuentro, están incluidos en el nodo `personal`, son de tipo `inetOrgPerson` y, en mi caso, el DN es `uid=fperez,ou=personal,dc=fi,dc=upm,dc=es`. Para ello, se usará el siguiente mandato:

```
ldapsearch -LLL -x -w contraseña -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' \
-b 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' -s base
```

Los alumnos están incluidos en el nodo `alumnos`, son de tipo `inetOrgPerson`. Se va a acceder a un alumno de prueba cuyo DN es `uid=cprueba,ou=alumnos,dc=fi,dc=upm,dc=es`, pero sólo vamos a mostrar algunos de los atributos del objeto. Para ello, se usará el siguiente mandato, donde se especifica al final del mandato los atributos que se quiere recuperar:

```
ldapsearch -LLL -x -w contraseña -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' \
-b 'uid=cprueba,ou=alumnos,dc=fi,dc=upm,dc=es' -s base objectClass cn sn uid uidNumber mail homeDirectory
```

Búsquedas en el árbol

En este apartado se van a mostrar distintas búsquedas en el DIT de la Facultad.

El primer ejemplo muestra todos los miembros del personal que comparten un determinado despacho. Nótese que es suficiente con un ámbito que abarque a los nodos hijos para visitar todo el personal del centro. Observe que el filtro se especifica en notación prefija (polaca):

```
ldapsearch -LLL -x -w contraseña -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' \
-b 'ou=personal,dc=fi,dc=upm,dc=es' -s one '(&(objectClass=inetOrgPerson)(roomNumber=4201))' cn sn
```

El siguiente ejemplo, que también usa un ámbito que afecta sólo a los hijos, obtiene los números de teléfono de los miembros del personal de nombre Fernando pero que no son del DATSI:

```
ldapsearch -LLL -x -w contraseinha -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' \
```

```
-b 'ou=personal,dc=fi,dc=upm,dc=es' -s one '(&(objectClass=inetOrgPerson)(!(departmentNumber=DATSI))(cn=*Fernando*)(telephone
```

El tercer ejemplo usa el ámbito por defecto (-s sub, que afecta a todo el sub-árbol) para enviar un correo a todas las personas vinculadas con el centro (profesores, alumnos, personal de servicio,...) que se llamen Fernando:

```
LISTA=`ldapsearch -LLL -x -w micontraseña -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' \
-b 'dc=fi,dc=upm,dc=es' '(&(objectClass=inetOrgPerson)(cn=*Fernando*))' mail | grep -v ^dn | sed 's/^mail:/'`
mail -s Felicitades `echo $LISTA` << EOF
Felicitades....
EOF
```

Uso de grupos

Es frecuente el uso de grupos dentro del DIT de una institución para poder tratar de forma un conjunto de nodos. En esta sección se plantean ejemplos que hacen uso de esa funcionalidad.

El primer ejemplo obtiene un listado con todos los números de teléfono del personal del DATSI accediendo primero al grupo asociado con ese departamento y, a continuación, por cada miembro de ese grupo, obtiene su número de teléfono. Nótese que son todos accesos de ámbito base.

```
LISTA=`ldapsearch -LLL -x -w micontraseña -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' \
-b cn=datsi,ou=dinamicos,ou=grupos,dc=fi,dc=upm,dc=es -s base memberUID | grep -v ^dn | sed 's/memberUid:/'`
for U in `echo $LISTA`
do
  ldapsearch -LLL -x -w micontraseña -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' \
  -b 'uid=$U',ou=personal,dc=fi,dc=upm,dc=es' -s base telephoneNumber | grep -v ^dn | sed 's/telephoneNumber:/'
done
```

Este segundo ejemplo crea una página web con todas las fotografías de los profesores del departamento de matemáticas. Observe el uso de la opción -t para que la información de las fotografías quede almacenada en ficheros externos.

```
LISTA=`ldapsearch -LLL -x -w micontraseña -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' \
-b cn=dma,ou=dinamicos,ou=grupos,dc=fi,dc=upm,dc=es -s base memberUID | grep -v ^dn | sed 's/memberUid:/'`
rm -rf fotos_dma*; mkdir fotos_dma
echo "<html><body>" >> fotos_dma.html
for U in `echo $LISTA`
do
  IMG=`ldapsearch -LLL -x -w micontraseña -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' \
  -b 'uid=$U',ou=personal,dc=fi,dc=upm,dc=es' -t -T fotos_dma -s base jpegPhoto | grep -v ^dn |
  sed 's/jpegPhoto:< file:\\/\//\'`
  test $IMG || continue
  mv $IMG ${IMG}.jpg
  echo "<img src=${IMG}.jpg><p>" >> fotos_dma.html
done
echo "</body></html>" >> fotos_dma.html
```

Operación compare

LDAP ofrece una operación para comparar si un determinado atributo de un objeto tiene un valor dado, pudiendo devolver, entre otros, los valores verdadero, si ese atributo tiene dicho valor, falso, si tiene asociado otro valor, o indefinido, si no tiene ese atributo.

```
ldapcompare -x -w micontraseña -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' \
'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' 'roomNumber: 4201'
```

Operaciones de modificación de una entrada

En esta sección se muestran ejemplos de operaciones que modifican mi entrada del DIT del centro.

El primer ejemplo me asigna un número de despacho:

```
ldapmodify -x -w micontraseña -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' << EOF
dn: uid=fperez,ou=personal,dc=fi,dc=upm,dc=es
add: roomNumber
roomNumber: 4201
-
EOF
```

El segundo me lo cambia:

```
ldapmodify -x -w micontraseña -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' << EOF
dn: uid=fperez,ou=personal,dc=fi,dc=upm,dc=es
changetype: modify
replace: roomNumber
roomNumber: 4202
-
EOF
```

El tercero me lo elimina:

```
ldapmodify -x -w micontraseña -D 'uid=fperez,ou=personal,dc=fi,dc=upm,dc=es' << EOF
dn: uid=fperez,ou=personal,dc=fi,dc=upm,dc=es
delete: roomNumber
-
EOF
```