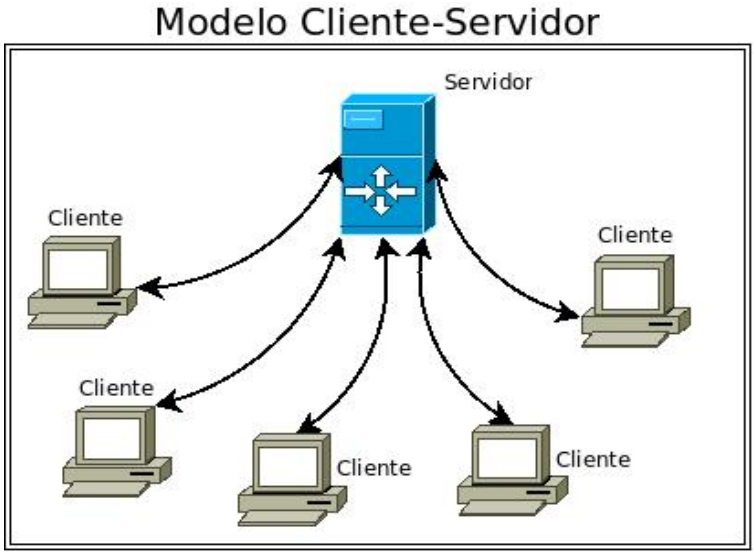
UD4. Implantación de dominios

- Estrutura cliente servidor.
- Protocolo LDAP.
- Concepto de dominio. Requisitos necesarios para montar un dominio.
- Servidores de dominio: estrutura e topoloxía. Bosques. Subdominios. Unidades organizativas.
- Controladores do dominio. Estacións e servidores membros dun dominio.
- Estratexias organizativas. Instalación dunha infraestrutura de dominio. Operacións sobre os servidores de dominio.
- Establecementos de relacións de confianza.
- Sitios e subredes. Replicación da información dos dominios.
- Servidores de catálogo global.
- Roles. Mestres de operacións.
- Delegación das funcións de administración.
- Realización de instalacións desatendidas de servidores de dominio.

1. Estrutura Cliente-Servidor

La **arquitectura Cliente/Servidor** es el procesamiento cooperativo de la información por medio de un conjunto de procesadores, en el cual múltiples clientes, distribuidos geográficamente, solicitan requerimientos a uno o más servidores centrales.

Se trata de la arquitectura más extendida en la realización de **Sistemas Distribuidos.**



Un sistema Cliente/Servidor presenta las siguientes **características:**

- **Servicio:** el servidor los proporciona y el cliente los utiliza.
- **Recursos compartidos:** Muchos clientes utilizan los mismos servidores y, a través

de ellos, comparten tanto recursos lógicos como físicos.

- **Protocolos asimétricos**: Los clientes inician "*conversaciones*". Los servidores esperan su establecimiento pasivamente.
- **Transparencia de localización física de los servidores y clientes:** El cliente no tiene por qué saber dónde se encuentra situado el recurso que desea utilizar.
- **Independencia de la plataforma HW y SW** que se emplee.
- Sistemas **débilmente acoplados**. Interacción basada en envío de mensajes.
- **Encapsulamiento de servicios.** Los detalles de la implementación de un servicio son transparentes al cliente.
- **Escalabilidad horizontal** (añadir clientes) **y vertica**l (ampliar potencia de los servidores).
- **Integridad:** Datos y programas centralizados en servidores facilitan su integridad y mantenimiento.

El Esquema de funcionamiento de un Sistema Cliente/Servidor sería:

1. El cliente solicita una información al servidor.
2. El servidor recibe la petición del cliente.
3. El servidor procesa dicha solicitud.
4. El servidor envía el resultado obtenido al cliente.
5. El cliente recibe el resultado y lo procesa.

Se deducen los tres elementos fundamentales sobre los cuales se desarrollan e implantan los sistemas Cliente/Servidor: **el proceso cliente** que es quien inicia el diálogo, **el proceso servidor** que pasivamente espera a que lleguen peticiones de servicio y el **middleware** que **corresponde a la interfaz** que provee la conectividad entre el cliente y el servidor para poder intercambiar mensajes.

Este modelo de descomposición en niveles, como se verá más adelante, permite introducir más claramente la discusión del desarrollo de aplicaciones en arquitecturas de hardware y software en planos.

Un **cliente** es todo proceso que reclama servicios de otro. Se lo conoce con el término ***front-end*.** Éste normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de la red y dar formato a los resultados.

El cliente se puede clasificar en:

- **Cliente basado en aplicación de usuario**. Si los datos son de baja interacción y están fuertemente relacionados con la actividad de los usuarios de esos clientes.
- **Cliente basado en lógica de negocio.** Toma datos suministrados por el usuario y/o la base de datos y los procesa según los requerimientos del usuario.

Un **servidor** es todo proceso que proporciona un servicio a otros. Se lo conoce con el término **back-end.** El servidor normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos y procesamientos de los datos

El **middleware** es un módulo intermedio que actúa como conductor entre sistemas permitiendo a cualquier usuario de sistemas de información comunicarse con varias fuentes de información que se encuentran conectadas por una red.

El **Network Operating System** es la parte central de toda la comunicación entre el cliente y el servidor. El NOS es el encargado de proporcionar una apariencia de sistema único a un sistema Cliente/Servidor. Se trata pues, de una extensión del Sistema Operativo. Los servicios que rinde son:

- **Servicios de transporte** comunicaciones Peer to Peer, Mensajes y Remote Procedure Call.
- **Servicios de aplicación del soporte de comunicaciones,** servicios de Directorio Glogal, Directorios LDAP, espacios de nombres (namespaces)...
- **Servicios de tiempo (fecha y hora**) en los clientes, mantenimiento consistente de la información...
- **Servicios de seguridad, confidencialidad, autentificación de usuarios, cifrados de datos...**

Se puede establecer un esquema de clasificación basado en los conceptos de:

- **Fat Client/Thin Server** donde el **peso** de la aplicación es ejecutada en el **cliente**
- **Fat Server/Thin Client** donde el **peso** de la aplicación es ejecutada en el **servidor**.

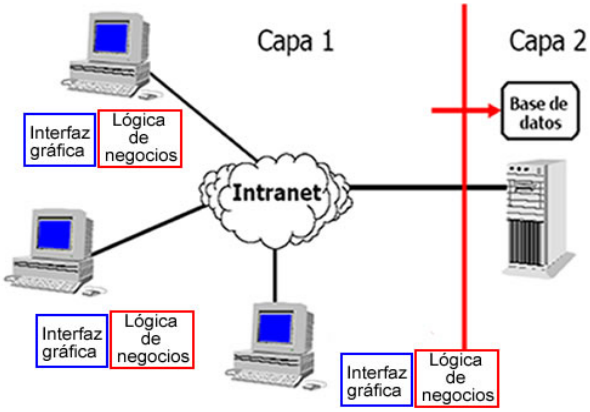
También los podemos clasificar por la **naturaleza del servicio:**

- **Servidor de archivos,** usan recursos compartidos sobre la red y son necesarios para crear repositorios de documentos, imágenes y archivos grandes sobre la red.
- **Servidores de Bases de Datos** asociada a la utilización de servidores de bases de datos relacionales SQL, y dependiendo de los requerimientos y restricciones se debe elegir entre una arquitectura dos o tres capas.
- **Servidores de transacciones** con un servidor de transacciones el proceso cliente llama a funciones, procedimientos o métodos que residen en el servidor, ya sea que se trate de un servidor de bases de datos o un servidor de aplicaciones.Estas aplicaciones denominadas OLTP (On Line Transaction Proccesing) están orientadas a dar soporte a los procedimientos y reglas de los sistemas de misión crítica. En la actualidad muchas aplicaciones tienen la necesidad de ser desarrolladas con la ayuda de transacciones, véase el ejemplo de los cajeros automáticos.
- **Servidores de Objetos** con un servidor de objetos, las aplicaciones Cliente/Servidor son escritas como un conjunto de objetos que se comunican. Los objetos cliente se comunican con los objetos servidores usando un Object Request Broker (ORB). El cliente invoca un método de un objeto remoto. El ORB localiza el método del objeto en el servidor, y lo ejecuta para devolver el resultado al objeto cliente.
- **Servidores Web** La primera aplicación cliente servidor que cubre todo el planeta es el World Wide Web. Este nuevo modelo consiste en clientes simples que hablan con servidores Web. Un servidor Web devuelve documentos cuando el cliente pregunta por el nombre de los mismos. Los clientes y los servidores se comunican usando un protocolo basado en RPC, llamado HTTP. Este protocolo define un conjunto simple de comandos, los parámetros son pasados como cadenas y no provee tipos de datos. La Web y los objetos distribuidos están comenzando a crear un conjunto muy interactivo de computación Cliente/Servidor.

- **Modelo Cliente-Servidor de 2 capas**

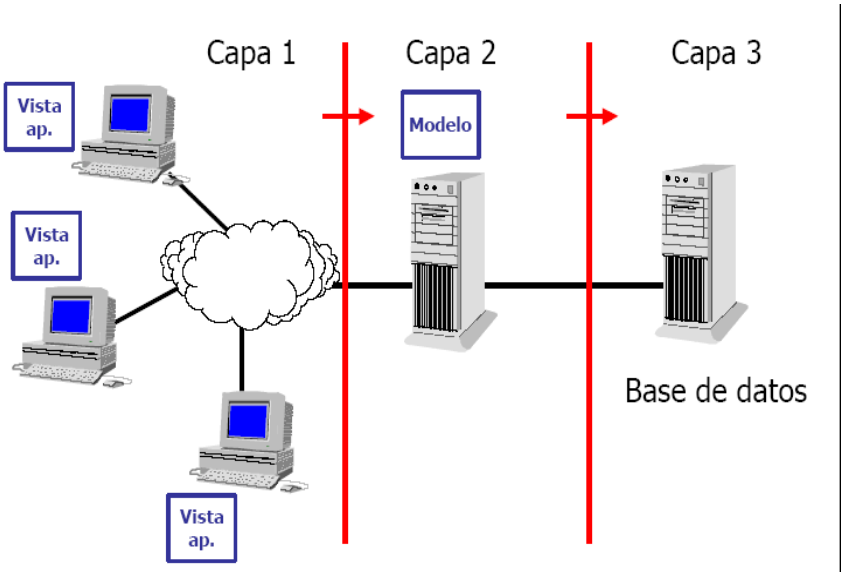
Esta estructura se caracteriza por la conexión directa entre el proceso cliente y un administrador de bases de datos. Es muy simple y barato, aunque tiene el inconveniente

que puede congestionar la red.



- **Modelo Cliente-Servidor 3 capas**

Esta estructura se caracteriza por elaborar la aplicación en base a dos capas principales de software, más la capa correspondiente al servidor de base de datos. Al igual que en la arquitectura dos capas, y según las decisiones de diseño que se tomen, se puede balancear la carga de trabajo entre el proceso cliente y el nuevo proceso correspondiente al servidor de aplicación. Los clientes son conectados vía LAN a un servidor de aplicaciones local, el cual a su vez se comunica con un servidor central de bases de datos. El **servidor local tiene un comportamiento dual,** dado que actúa como cliente o servidor en función de la dirección de la comunicación.



2. Protocolo LDAP

LDAP (Protocolo de Acceso Ligero a Directo) es un protocolo de acceso ligero a datos y archivos contenidos en un **directorio**. El servicio LDAP permite organizar toda la información del **directorio**. Está basado en el standard **X.500.**

Un **servicio de directorio** es un conjunto de objetos con atributos organizados en una manera **lógica y jerárquica,** por ejemplo, la estructura de RR.HH. de una empresa, o la estructura en sí misma de una organización. LDAP también es considerado una **base de datos** (aunque su sistema de almacenamiento puede ser diferente) a la que pueden realizarse consultas.

LDAP utiliza el **protocolo TCP/IP** para las comunicaciones a través de la red. El protocolo seguido por los clientes y servidores que se comunican es el siguiente:

- El cliente se conecta al servidor y **solicita autenticación (Bind).**
- El servidor regresa el código de **resultado de la operación de autenticación.**
- El cliente solicita una **operación de búsqueda o cualquier otra operación.**
- El servidor realiza la operación de obtención de los **resultados de la búsqueda.** Si no se encontró nada, no regresa nada.
- El servidor **envía el código del resultado de la operación de búsqueda al cliente.**
- El cliente solicita el **cierre de la conexión (Unbind).**
- El servidor **regresa el código del resultado de la operación de cierre de conexión** y cierra la conexión

La **estructura de un directorio** es:

- Un directorio es un **árbol de entradas**.
- Una entrada consta de un **conjunto de atributos**.
- Un **atributo tiene un nombre** (un *tipo de atributo* o *descripción de atributo*) **y uno o más valores**. Los atributos son definidos **en un *esquema*** *(véase luego)***.**
- Cada **entrada tiene un identificador único:** su ***Nombre distinguido*** **(*Distinguished Name*, DN**). Este consta de su ***Relative Distinguished Name*** **(RDN)** construido por algunos atributos en la entrada, **seguidos del DN de la entrada del padre**.

Por ejemplo, un árbol de directorio LDAP (sin incluir entradas individuales) podría parecerse a esto:

dc=example, dc=com
ou=clientes
ou=asia
ou=europe
ou=usa
ou=empleados

Una **entrada individual** puede ser como esta en formato LDAP:

dn: cn=John Doe, ou=empleados dc=example,dc=com

cn: John Doe

givenName: John

sn: Doe

telephoneNumber: +1 888 555 6789

telephoneNumber: +1 888 555 1232

mail: john@example.com

Donde:

- **dn es el nombre de la entrada; no es un atributo** ni tampoco parte de la entrada. **"cn=John Doe" es el nombre distinguido relativo, y "dc=example,dc=com" es el nombre distinguido de la entrada** del padre,
- **dc indica domain component (componente de dominio) y**
- **ou** es la **unidad organizativa**.
- Las otras líneas presentan los **atributos en la entrada.** Los nombres de atributos son generalmente cadenas mnemotécnicas, como "cn" para common name (nombre común), "dc" para domain component (componente de dominio), "mail" para dirección de e-mail y "sn" para surname (apellido).

Un **servidor aloja un subárbol comenzando por una entrada específica**, por ejemplo "dc=example,dc=com" y sus hijos. Los servidores también **pueden almacenar referencias a otros servidores**, con los cual un intento de acceso a "ou=department, dc=example, dc=com" puede retornar una *referencia* o *continuación de referencia* a un

servidor que aloja esa parte del árbol de directorio.

Un formato **URL** de LDAP puede ser:

ldap://host:port/DN?attributes?scope?filter?extensions

La mayoría de los componentes son opcionales.

- *host* es el nombre del servidor o dirección IP del servidor LDAP donde se realiza la consulta.
- *port* es el puerto de red del servidor LDAP.
- *DN* es el nombre distinguido a usar como base de búsqueda.
- *attributes* es una lista separada con comas de atributos a devolver.
- *scope* especifica el ámbito de búsqueda y puede ser "base" (por defecto), "one" o "sub".
- *filter* es un filtro de búsqueda.
- *extensions* son extensiones al formato URL de LDAP.

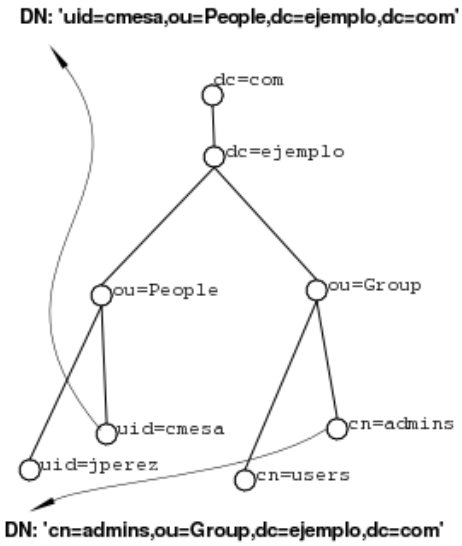
Por ejemplo,

"ldap://ldap.example.com/cn=John%20Doe,dc=example,dc=com"

refiere a todos los usuarios en la entrada de John Doe en ldap.example.com, mientras:

"ldap:///dc=example,dc=com??sub?(givenName=John)"

busca por la entrada en el servidor por defecto.



Los servidores LDAP pueden ser configurados para **replicar algunos o todos de sus datos**, basándose en enviar o recojer la información, utilizando autentificación simple o autetificación basada en certificados. Para ello necesitamos una implementación LDIF.

LDIF es un acrónimo para LDAP Data Interchange Format. Este formato permite representar entradas de un directorio LDAP o cambios hechos en el mismo, en un archivo con formato de texto. Esto nos da una gran flexibilidad en el momento en que queremos migrar datos entre directorios. LDIF es utilizado para varias funciones como son respaldos de un directorio, replicación, modificación de esquema y cuando mueves una gran cantidad de datos de un directorio a otro.

LDAP provee de una complejo nivel de instancias de **control de acceso,** o ***ACLs*** establecido del servidor que entre otras establece un nivel de control el acceso al directorio y sus recursos. Un ejemplo de ACL (acceso solo lectura a todos los del grupo People):

```
access to dn=".*,ou=People,dc=ejemplo,dc=net"
    by * read
```

Existe mucho software que nos ayuda a implementar un servicio de directorio LDAP. Podemos mencionar los siguientes:

- Microsoft Active Directory
- OpenLDAP
- phpLDAPadmin
- Novell eDirectory
- Samba4 (integra OpenLDAP y Samba)

3. Concepto de dominio. Requisitos necesarios para montar un dominio.

Desde el punto de vista de la administración de sistemas, suele denominarse **dominio a un conjunto de equipos interconectados que comparten información administrativa (usuarios, grupos, contraseñas,etc.) centralizada**. Ello requiere fundamentalmente la disponibilidad de (al menos) un ordenador que almacene físicamente dicha información y que la comunique al resto cuando sea necesario, típicamente mediante un **esquema cliente-servidor.** Por ejemplo, cuando un usuario desea iniciar una conexión interactiva en cualquiera

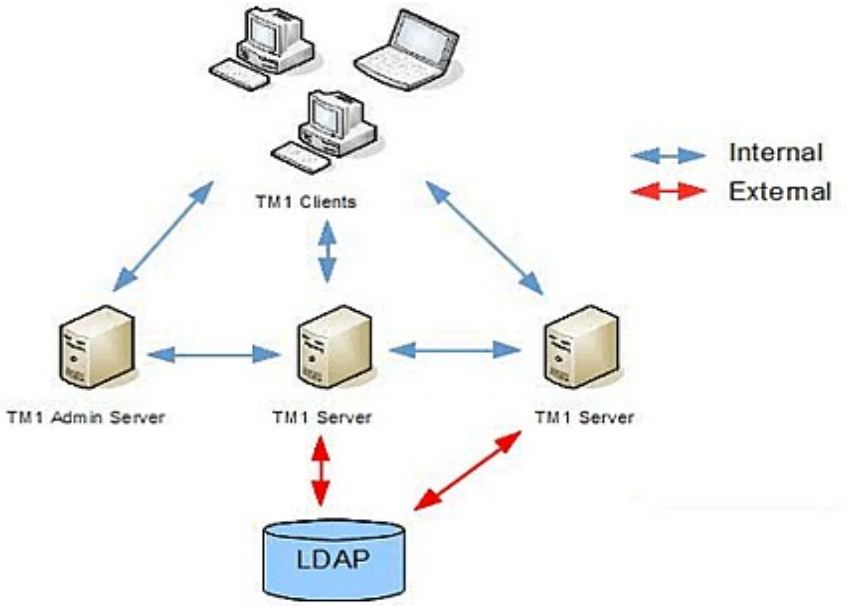
de los ordenadores (clientes) del dominio, dicho ordenador deberá validar las credenciales del usuario en el servidor, y obtener de éste todos los datos necesarios para poder crear el contexto inicial de trabajo para el usuario.

En **Windows *200/0/3/8/12* Server**, única herramienta que puede llevar a cabo la implementación del concepto de dominio se realiza mediante el denominado **Directorio Activo** , un servicio de directorio basado en diferentes estándares como **LDAP (Lightweight Directory Access Protocol) y DNS (Domain Name System).**

En el mundo Linux/Unix, los dominios solían implementarse mediante Network Information System (NIS), del que existían múltiples variantes. Sin embargo, la integración de servicios de directorio en Unix ha posibilitado la incorporación de esta tecnología, mucho más potente y escalable que NIS, en la implementación de dominios nos referimos a la implementación libre del protocolo LDAP para Unix, denominada **OpenLDAP** (www.openldap.org), puede utilizarse para implementar dominios en Linux.

Actualmente en Linux se utiliza Samba4 para implementar un dominio. Samba 4 introduce soporte la tecnología de Directorio Activo de Microsoft mediante la **combinación de un servidor LDAP, el servidor de autenticación Kerberos Heimdal, un servicio de DNS dinámico (BIND) lo que permite la implementación de un PDC (Controlador Primario de Dominio)**. Así que funciona como un controlador de dominio de Active Directory para todas las versiones Microsoft Windows. Esta implementación, permite directivas de grupo, perfiles móviles, etc. Curiosamente estas características fueron implementadas con el apoyo y colaboración de Microsoft.

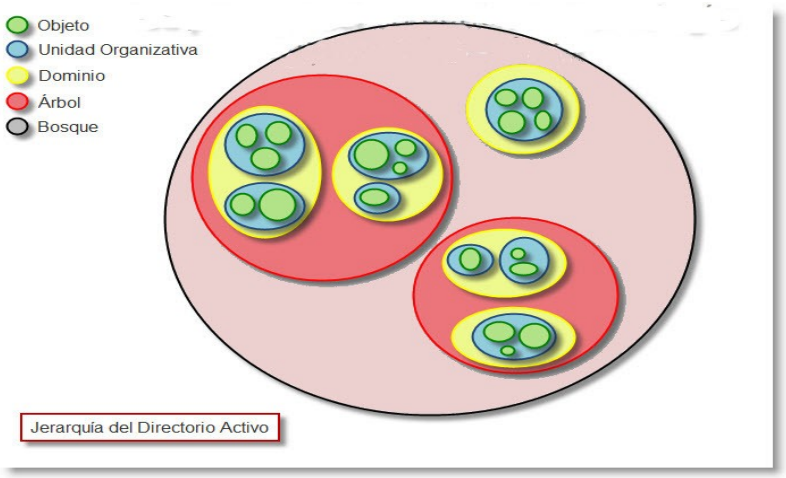
Para finalizar, en el contexto de las **redes de ordenadores**, se denomina **directorio a una base de datos especializada que almacena información sobre los recursos, u "objetos", presentes en la red** (tales como *usuarios, ordenadores, impresoras, etc.*) y que pone dicha información a disposición de los usuarios de la red. Por este motivo, esta base de datos suele estar optimizada para operaciones de búsqueda, filtrado y lectura más que para operaciones de inserción o transacciones complejas. Existen diferentes estándares que especifican servicios de directorio, siendo el denominado **X.500 tal vez el más conocido** y que ya vimos en el apartado anterior.



Los **requisitos hardware** ya los vimos en la UD1 para Windows Server que suele ser bastante exigente. En Linux depende del uso o no de entorno gráfico. Desde el punto de vista de la facilidad de uso, robustez, etc.... hay páginas enteras sobre ello a favor de un sistema u otro. En nuestro caso utilizaremos preferentemente la primera por su rapidez, aunque llevaremos a cabo también la implementación de un PDC en Linux.

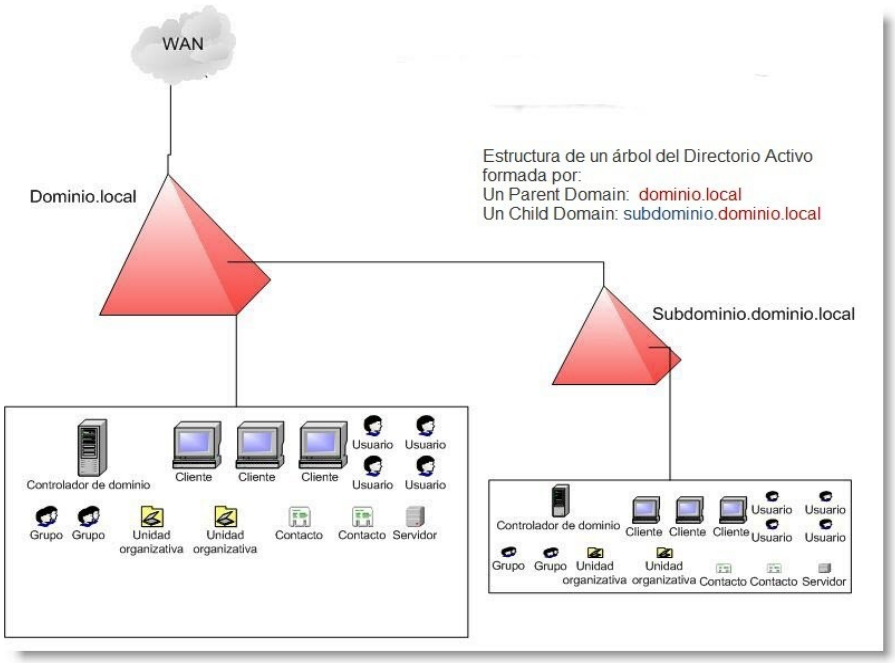
4. Servidores de dominio: estrutura e topoloxía. Bosques. Subdominios. Unidades organizativas.

En la instalación del Directorio Activo (en Windows Server) o LDAP (en Linux) para la administración de un dominio, **hay que darse un tiempo especial** para el diseño de la red, de los servicios, de las politicas de grup, etc...



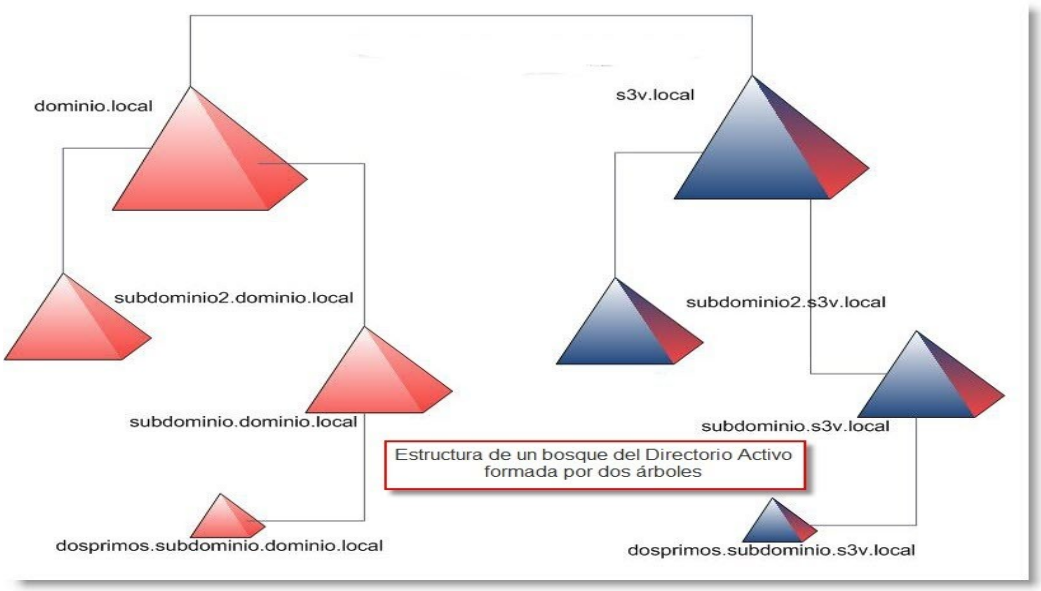
El Directorio Activo tiene un **funcionamiento similar a otras estructuras de LDAP** (*Lightweight Directory Access Protocol*), ya que este protocolo viene implementado de **forma similar a una base de datos,** la cual almacena en forma centralizada toda la información relativa a un dominio de autenticación. La ventaja de la centralización es la sincronización entre los distintos servidores de autenticación de todo el dominio.

Cuando implementamos un **DOMAIN CONTROLLER o CONTROLADOR DE DOMINIO**, este es por definición **una COLECCIÓN DE OBJETOS como Usuarios, Impresoras, y PCs,** y a este dominio le asignamos un nombre, este nombre **esta vinculado a la direccion DNS (de ahí la necesidad de instalar un servidor DNS cuando instalamos un dominio)**, para ser ubicado fácilmente en la red. Los **DNS** sirven para identificar a los dominios y subdominios por sus respectivos **namespaces (nombres de espacio)** estos nombres de espacios deben seguir un patrón al momento de implementarse que vimos en el apartado anterior. Si creamos un subdominio dentro de un dominio por ejemplo "***marca.com***", **este subdominio "*hereda*"** el nombre del dominio principal, de esta manera quedaría el nombre de **"*futbol.marca.com*"**, a consecuencia de esto, el dominio principal pasa a denominarse **PARENT DOMAIN** o Dominio Padre, asi mismo el subdominio pasa a denominarse **CHILD DOMAIN** o dominio hijo, por el mismo hecho de herencia de nombres. Si continuamos creando mas sub dominios estos heredaran los nombres.



A la coleccion de dominios y subdominios con nombres contiguos (herencia) se le da el nombre de <u>ARBOL DE DOMINIOS (Tree)</u> y esta compuesto por un **DOMAIN CONTROLLER** y uno o mas servidores en cada subdominio los cuales llevan el nombre de **MEMBER SERVER** (Servidor Miembro) quienes tienen el trabajo de "avisar" (replicar) de cualquier cambio en cada subdominio al **DOMAIN CONTROLLER**,se mantienen actualizados los datos en el **Directorio Activo**, así funcionan las sucursales bancarias, cuando se hace un retiro de un cajero automático, sus datos automáticamente son actualizados (replicados) en todas las agencias y oficinas casi al instante.

Sin embargo, un árbol que esta compuesto por un dominio y uno o mas subdominios no puede pertenecer a organizaciones diferentes, un arbol matiene un nombre principal y sus subdominios lo heredan. **Para el caso de varias organizaciones que desean interconectarse y compartir sus servicios de directorio**, deben existir por los menos **dos DOMAIN CONTROLLER**, uno para cada árbol. Supongamos dos árboles uno llamado **COMPAQ** y otro llamado **HP** ambos con sus subdominios correspondientes, **para enlazarse deberán confiar el uno en el otro solo de esa manera** pueden compartir sus recursos, porque Server 2003/8 esta pensado en SEGURIDAD y para ellos se deben establecer **RELACIONES DE CONFIANZA** entre sus servidores, crear un enlace para que ambos dominios puedan compartir sus servicios de directorio.



A esto se le conoce **con el nombre de BOSQUE DE DOMINIO (Forest)**, el cual está **compuesto por dos o mas Domain Controllers y dos o mas Member Serve**r.

Resumiendo, un dominio esta compuesto por objetos (PC, Usuarios, Impresoras, Escáner...) los cuales se pueden agrupar en OUs, la colección de dominios hacen un Árbol, la colección de arboles hacen un Bosque. Con lo que la Estructura Logica del Directorio Activo posee una jerarquía, y ésta esta dada por los siguientes componentes en estricto orden: **Objetos – Unidades Organizativas – Dominios – Árboles – Bosques.**

Queda por concretar el **concepto de OU (Unidad Organizativa) dentro del Active Directory** por ser desde el punto administrativo de un **Controlador de Dominio** fundamental**.** Las **OUs** se usan para organizar objetos en el directorio dentro de unidades administrables. Usamos las OUs para agrupar y organizar objetos con **propósitos administrativos**, como **delegar derechos administrativos** y asignar políticas para una colección de objetos como una unidad simple. En resumen:

- Permiten organizar objetos en un dominio como **cuentas de usuario**, **equipo** y **grupos, archivos e impresoras**
- Nos permiten delegar control administrativo, podemos especificar permisos en la propia OU y los objetos que contiene para uno o varios usuarios y grupos.

- Simplifican la administración de los recursos comúnmente agrupados.

Distinguimos tres tipos de nombres, **nombre completo relativo, nombrecompleto y nombre canónico. Es importante entender, una vez más, la sintaxis de LDAP para cuando usemos scripts para consulta y administración de AD.**

- El **nombre completo relativo** de LDAP **identifica unívocamente al objeto dentro su contenedor principa**l. Por ejemplo, el nombre completo relativo de LDAP correspondiente a una unidad organizativa llamada **miOU** sería **OU=miOU.**
- El **nombre completo** de LDAP **es globalmente único**. Por ejemplo, el nombre completo de una unidad organizativa llamada **MiUnidadOrganizativ**a del dominio microsoft.com sería **OU=MiUnidadOrganizativa, DC=microsoft, DC=com.**
- El **nombre canónico** se crea de la misma manera que el nombre completo, pero se representa con una notación diferente. El nombre canónico de la OU del ejemplo anterior sería **Microsoft.com/MiUnidadOrganizativa.**

5. Controladores do dominio. Estacións e servidores membros dun dominio.

Como se indicó en apartados anteriores se denomina **controlador de dominio DC** a una **entidad administrativa,** esto es, uno o más ordenadores agrupados que se ciñen a unas **reglas de seguridad y autenticación comunes**. Para regular un dominio, se precisa **al menos de un equipo que sea el controlador principal**, la fuente primera donde se almacenan las reglas del dominio, y donde serán consultadas esas reglas en última instancia. Un **controlador primario de dominio (PDC)** puede implementarse tanto bajo **Windows de la familia Server** como bajo **Linux** (actualmente con Samba4)**.**

Para que un ordenador con Linux pueda explorar una red Windows, acceder a sus recursos, compartir los propios, autenticarse o, ser el controlador de un dominio, tiene que ejecutar un conjunto de programas agrupados bajo el nombre ***Samba4*.** El nombre del protocolo que se utiliza para ello es **SMB,** pero al estar el acrónimo registrado por Microsoft, no se pudo utilizar y se sustituyó por *Samba.*

Samba4 superó una limitación importante gestionando dominios que tenía Samba, y es que en su versión estable actual **puede manejar políticas de grupo y soporta active directory.**

Se conoce como **servidor miembro** a un equipo en el que se ejecuta Windows Server y

pertenece al dominio, pero que **no actúa como controlador de dominio porque no contienen ninguna copia de los datos de Active Directory.** Puede realizar funciones de servidor de aplicaciones, de impresión, etc...

Normalmente el sistema operativo Windows guarda las contraseñas de los usuarios en un **archivo llamado SAM** en la carpeta **\WINDOWS\system32\config\ .** La **base de datos** del controlador de dominio cumple la misma función que **el archivo SAM**, **pero ahora almacena todas las contraseñas de la red.**

Finalmente tenemos las estaciones de trabajo que son los equipos desde los cuales los usuarios se autentifican antes el dominio. Asimismo los equipos son objetos del dominio.

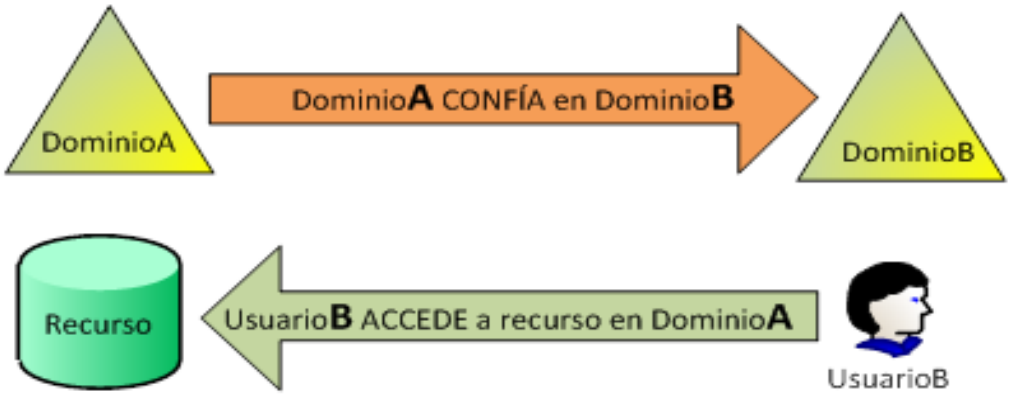
6. Estratexias organizativas. Instalación dunha infraestrutura de dominio. Operacións sobre os servidores de dominio.

Se desarrollará durante las actividades prácticas.

7. Establecementos de relacións de confianza.

Una relación de confianza (trust) es una relación establecida entre **dos dominios de forma que permite a los usuarios de un dominio ser reconocidos por los Controladores de Dominio de otro dominio.**

Estas relaciones permiten a los usuarios **acceder a los recursos de otro dominio**, y a los **administradores definir los permisos y derechos** de usuario para los usuarios del otro dominio. También permite establecer **comunicación entre varios controladores de dominio**, con el fin de poder **administrar desde un solo punto** de la red a todos los usuarios y recursos que tengas.



En una red que consista en dos o más dominios, cada dominio actúa como una red por separado **con su propia base de datos de cuentas**.

Puede pasar que un usuario de un dominio necesite utilizar algunos o todos los recursos del otro dominio. La solución usual para la configuración de niveles de acceso de usuario entre dominios es lo que se llama relación de confianza. Windows Server soporta varios tipos de relaciones de confianza.

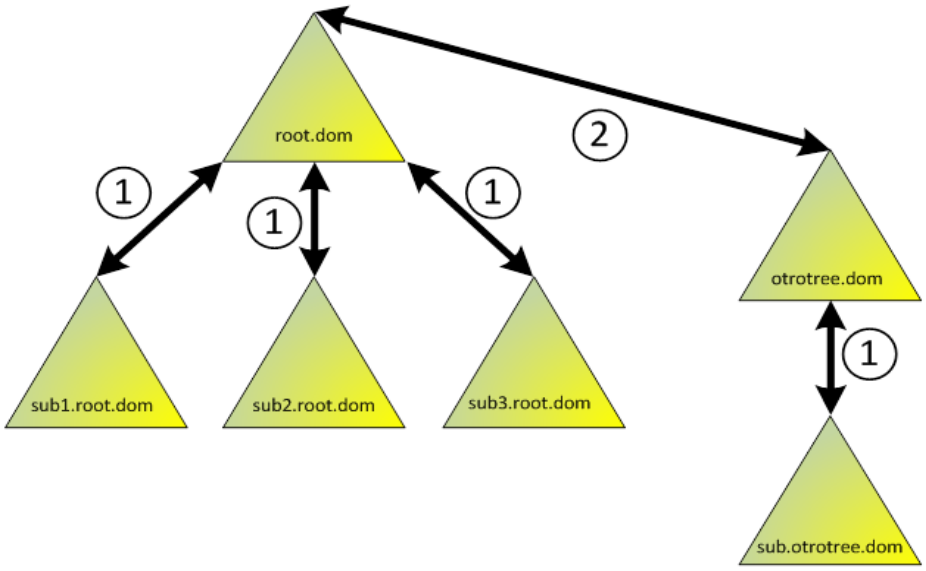
Los diferentes **tipos de relaciones** se diferencian en función de 3 rasgos característicos:

- **Método de creación.** Algunos tipos de relaciones de confianza se crean de forma automática (implícita) y otros de forma manual (explícita).
- **Dirección.** Si la relación de confianza es unidireccional, los usuarios del dominio A (de confianza) pueden utilizarlos recursos del dominio B (que confía), pero no al revés. En una relación bidireccional, ambas acciones son posibles.
- **Transitividad.** En una relación de confianza transitiva, si un dominio A confía en otro B, y este confía en un tercero C, entonces, de forma automática, es decir implícita, A confía en C. En las relaciones no transitivas, la confianza entre A y C tendría que añadirse explícitamente.

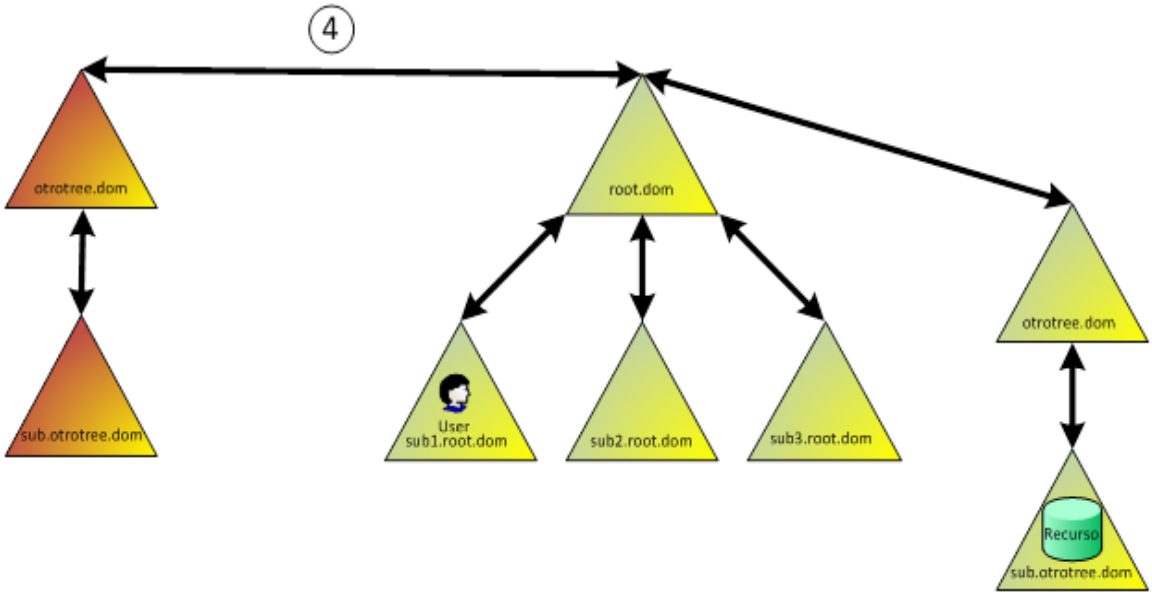
Los tipos de relaciones de confianza pueden ser:

- **Relaciones entre dominios sin confianza.** Por ejemplo, un usuario podría tener una cuenta separada para cada dominio en una red de múltiples dominios.
- **Relaciones de confianza unidireccionales.** Por ejemplo. un usuario que necesita acceder a ambos dominios desde una única cuenta. Es el trust más simple
- **Relaciones de confianza bidireccionales.** Se crea estableciendo 2 relaciones de confianza unidireccionales, una en cada dirección Un usuario conectado con éxito a uno de los dominios será considerado auténtico por el otro dominio. Permiten una mayor flexibilidad en el acceso de los usuarios a los recursos y hacen la administración de la red mucho más fácil.

Los dominios de Windows Server del **mismo bosque** comparten relaciones de confianza transitivas unos con otros. Hay una **confianza transitiva implícita** entre los dominios raíz de cada árbol del bosque de Windows Server. También existe una confianza transitiva implícita entre todos los dominios contiguos de un único árbol.



En cambios las **relaciones de confianza de bosques diferentes son explícitas.** Suelen ser habituales cuando existen programas de colaboración entre diferentes empresas u organizaciones o que desean acceder a recursos mutuos. Se establecen de forma manual.



Para configurar nuevas relaciones de confianza o ver las que están actualmente establecidas y con que características se puede realizar a través de la línea de comando con ***netdom*** o en entorno gráfico ***Dominios y confianzas de Active Directory.***

8. Sitios e subredes. Replicación da información dos dominios.

En Active Directory, la **estructura lógica que hemos visto hasta ahora está separada de la estructura física**. La estructura lógica se utiliza para organizar los recursos de red mientras que la estructura física se utiliza para configurar y administrar el tráfico de red. En concreto, la estructura física de Active Directory se compone de **sitios y controladores de dominio.**

La estructura física de Active Directory define **dónde y cuándo se producen el tráfico de replicación y de inicio de sesión**. Una buena comprensión de los componentes físicos de Active Directory permite optimizar el tráfico de red y el proceso de inicio de sesión, así como solventar problemas de replicación.

- **Sitios**

Un **sitio es una combinación de una o varias subredes IP** que están conectadas por un vínculo de alta velocidad. Definir sitios permite configurar la topología de replicación y acceso a Active Directory de forma que Windows Server utilice los vínculos y programas más efectivos para el tráfico de inicio de sesión y replicación.

Normalmente los sitios se crean por dos razones principalmente:

- Para optimizar el tráfico de replicación.
- Para permitir que los usuarios se conecten a un controlador de dominio mediante una conexión confiable de alta velocidad.

Es decir, **los sitios definen la estructura física de la red**, mientras que los dominios definen la estructura lógica de la organización.

- **Controladores de dominio**

Un controlador de dominio (*Domain Controller*, *DC*) es un equipo donde se ejecuta Windows Server y que almacena una replica del directorio. Los controladores de **dominio ejecutan el servicio KDC**, que es responsable de **autenticar inicios de sesión de usuario**.

La información almacenada en cada controlador de dominio se divide en tres categorías (particiones): **dominio, esquema y datos de configuración**. Estas particiones del directorio son las unidades de replicación:

- **Partición del directorio de esquema**: contiene todos los tipos de objetos y atributos que pueden ser creados en Active Directory. Estos datos son comunes a todos los dominios en el bosque. Por tanto los datos del esquema se replican a todos los controladores de dominio del bosque.

- **Partición de directorio de configuración**: contiene la estructura de los dominios y la topología de replicación. Estos datos son comunes a todos los dominios en el bosque, y se replican a todos los controladores de dominio en el bosque.
- **Partición de directorio de dominio**: contiene todos los objetos del directorio para este dominio. Dichos datos se replican a todos los controladores de ese dominio, pero no a otros dominios.
- **Partición de directorio de aplicaciones**: contiene datos específicos de aplicación. Estos datos pueden ser de cualquier tipo excepto *principales de seguridad* (usuarios, grupos y equipos). En este caso, se tiene un control fino sobre el ámbito de la replicación y la ubicación de las réplicas.

Además de estas cuatro particiones de directorio de escritura, existe una cuarta categoría de información almacenada en un controlador de dominio: **el catálogo global**. Un catálogo global **es un controlador de dominio que almacena las particiones de directorio** de escritura, así como copias parciales de sólo lectura de todas las demás particiones de directorio de dominio del bosque.

Los **controladores de dominio admiten cambio**s, y estos cambios se replican a todos los controladores de dominio. Las operaciones de administración de usuarios, grupos y equipos son operaciones típicas de múltiples maestros. Sin embargo **no es práctico que algunos cambios se realicen en múltiples maestros debido al tráfico de replicación** y a los posibles conflictos en las operaciones básicas. Por estas razones, las funciones especiales, como la de servidor de catálogo global y operaciones de maestro único, se asignan sólo a determinados controladores de dominio por su seguridad y sobre todo por la comodidad de la administración.

Windows Server admite **la replicación con múltiples maestros: todos los controladores de un dominio pueden recibir los cambios efectuados en los objetos y pueden replicar esos cambios a todos los demás controladores de dicho dominio**. De manera predeterminada, **el primer controlador de dominio creado en un bosque es un servidor de catálogo global,** que contiene una réplica completa de todos los objetos del directorio para su dominio y una réplica parcial de todos los objetos almacenados en el directorio de todos los demás dominios del bosque.

Replicar datos de Active Directory entre controladores de dominio proporciona ventajas en cuanto a **disponibilidad de la información, tolerancia a errores, equilibrio de la carga y rendimiento.**

La práctica a realizar sobre la creación de un controlador adicional de dominio tiene que ver con esto último.

9. Servidores de catálogo global.

El **catálogo global es el almacén central de información sobre objetos (OU, grupos, usuarios, equipos,...) en un árbol del bosque**. De manera predeterminada, un catálogo global **se crea automáticamente en el controlador de dominio inicial del bosque**, **conocido como servidor de catálogo global.** Almacena una copia completa de todos los **atributos de los objetos del directorio** para su host de dominio y una copia parcial de todos los atributos de objetos que contiene el directorio de cada dominio en el bosque. La **copia parcial almacena los atributos usados con más frecuencia** en las operaciones de búsqueda (nombre y apellidos de usuario, nombre de inicio de sesión, etc.). Los atributos de los objetos que se copian en el catálogo global heredan los mismos permisos que tienen en los dominios origen, garantizando la seguridad de los datos almacenados en el catálogo global.

El **catálogo global realiza dos funciones clave** en el directorio:

- **Permite el inicio de sesión en red** proporcionando información universal sobre pertenencia al grupo de un controlador de dominio cuando se realiza un proceso de inicio de sesión.
- **Permite encontrar información de directorio** con independencia de qué dominio del bosque contenga los datos en ese momento.

Cuando un usuario inicia la sesión en red, el catálogo global proporciona información universal de pertenencia al grupo para la cuenta a los controladores de dominio que procesan la información de inicio de sesión.

Si sólo hay un controlador de dominio en el dominio, el controlador de dominio y el catálogo global son el mismo servidor. Si hay varios controladores de dominio en la red, el **catálogo global es el controlador de dominio que esté configurado como tal.** Si **un catálogo global no está disponible** cuando el usuario inicia el proceso de inicio de sesión, el usuario **sólo será capaz de iniciar la sesión en el equipo local.**

Si un usuario es **miembro de un grupo de administración de dominio,** será capaz de **iniciar la sesión en red incluso cuando el catálogo global no esté disponible.**

El catálogo global está diseñado para responder al usuario y a las preguntas de programa sobre los objetos situados en cualquier lugar del árbol de dominio o bosque a la máxima velocidad y generando un tráfico de red mínimo. Debido a que un único catálogo global contiene información sobre todos los objetos del dominio en un bosque, una pregunta sobre un objeto se puede resolver por el catálogo global del dominio donde se realiza la pregunta. De esta manera, encontrar información en el directorio no proporciona tráfico innecesario debido a las preguntas en los límites del dominio.

Opcionalmente, **se puede configurar cualquier controlador de dominio** o designar controladores de dominio **opcionales como servidores globales de catálogo.** A la hora de considerar qué controladores de dominio se designan como servidores globales de catálogo, es aconsejable basar la decisión **en la eficiencia de la estructura de red** para manejar tráfico de preguntas y replicación. Sin embargo, la disponibilidad de servidores adicionales puede proporcionar respuestas más rápidas a las preguntas de los usuarios, al igual que redundancia. Es recomendable que cada dominio principal de una empresa tenga al menos un servidor de catálogo global.

10. Roles. Mestres de operacións.

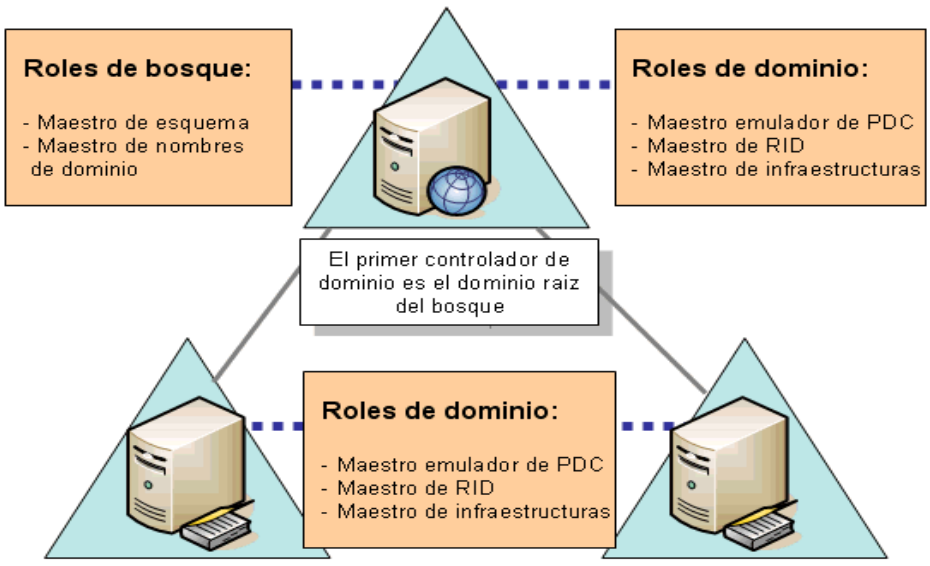
En el Directorio Activo, todos los controladores de dominio son iguales (a partir del Server 2000), a diferencia de NT4 en que un servidor era el principal (PDC) y el resto eran copias de sólo lectura de la base de datos del dominio (los BDC).

Dada la complejidad que puede adquirir un directorio, por ejemplo, varios **sitios** que pueden ser fábricas cercanas o complejas estructuras en distintos países debe haber por fuerza un **mecanismo que haga que toda la información sea coherente entre los sitios**. Es decir si un usuario ha bloqueado su cuenta en Barcelona yo debo, como administrador en Vigo, desbloqueársela. Para ello me iré al administrador de usuarios y lo haré. Si todo está bien configurado internamente l**as sincronizaciones entre los sitios dejarán la base de datos coherente,** al usuario le llegará ese desbloqueo inmediatamente y en el controlador de dominio de ese sitio estarán actualizados los cambios.

Por ello debe haber una serie de roles que **sólo pueden ser ejercidos por un único DC**, al tratarse de funciones que requieren ser únicas en el bosque o dominio. Por así decirlo, el **DC que ostenta en un momento dado un rol maestro** realiza funciones de moderador o director de esa función. **Un rol maestro puede transferirse de un DC a otro**, incluso forzarse el cambio caso de desastre del DC que lo ostenta, pero en ningún momento puede haber más de un DC con el mismo rol. A estos roles se les conoce como **FSMO (Flexible Single Master Operations**).

Los **tipos de roles maestros de operaciones son cinco**, **dos a nivel bosque y tres en cada dominio del mismo.**

- **Bosque:** A este nivel, y localizados siempre en algún DC del dominio raíz del bosque (**el primero que montas cuando lo creas)** hay dos, el **maestro de esquema** y el **maestro de nombres de dominio.**
    - **Maestro de Esquema**: es el DC que dirige todas las operaciones de cambio en el esquema del AD (la definición de clases de objeto, con sus atributos). El esquema **contiene la definición de clases de objetos y atributos** que se utilizan para crear todos los objetos de Active Directory, como usuarios, equipos, impresoras, Cuando se hace una modificación al esquema, siempre se realiza sobre el maestro de esquema (aunque la consola la lancemos desde otro DC), y a continuación se replica a todos los DCs del bosque. Esto permite asegurar que el esquema sea único para todo el AD.



- **Maestro de Nombres de Dominio**: El DC que ostenta este rol es el que **controla que los nombres propuestos para nuevos dominios** en el bosque no estén en uso y que la topología de nombres sea la correcta (p.e, si tenemos un árbol con un dominio de nombre "españa.es", no podremos crear otro árbol de nombre "sevilla.españa.es", sino que tendrá que ser un subdominio del anterior. Controla también la eliminación de dominios del bosque.

- **Dominio**: En cada dominio del bosque hay **tres roles maestros,** que pueden ser ejercidos por el mismo o por distintos DCs del dominio. Son los siguientes:
  - **Emulador de PDC**: Entre otras, realiza todas aquellas tareas que los equipos anteriores a Windows 2000 esperaban que se realizasen en un PDC de NT4. Cuando un DC recibe una modificación de la contraseña de un usuario, al primero que se lo replica es al PDC, quien además **ejerce de árbitro** cuando se produce una autenticación incorrecta de la contraseña de un usuario (antes de generar el mensaje de error, el DC en que se valida la contraseña errónea le pregunta al PDC por si éste ya hubiera recibido un cambio de la contraseña). El PDC de un dominio **es la cabeza jerárquica** en el mismo para la sincronización de tiempo (los clientes sincronizan con el DC con que se validan, y éstos con el PDC de su dominio). También, cuando editamos una GPO (Política de Grupo) desde cualquier equipo, por defecto se hace contra la copia almacenada en el PDC del dominio y se guardan los cambios en el mismo, tras lo cual se replican al resto de DCs.
  - **RID Master (Relative Identifier Master)**: Al ser todos iguales, en cualquier DC del dominio se pueden crear objetos del AD. Al crear un objeto de tipo usuario, grupo o equipo **se le asigna un identificador único de seguridad en el dominio (SID)**. Este identificador consta de una parte única para todo el dominio y de otra variable dentro del mismo, que le asigna el DC en que se crea el objeto. Para evitar que dos DCs distintos generen el mismo SID para un objeto, el DC que hace de RID master asigna al resto de DCs del dominio un número de IDs (un RID Pool), de tal forma que son distintos en cada DC. Cuando a un DC se le está acabando el número de IDs disponibles, solicita al RID Master que le asigne un nuevo pool de RIDs. Si el RID Master cayese y no forzásemos que otro DC pasase a llevar este rol, llegaría un momento en que

no se podrían crear más objetos en el dominio por falta de IDs.

- **Maestro de Infraestructura**: Es el DC **responsable de actualizar en otros dominios de su mismo bosque aquellos objetos del dominio propio que son referenciados por objetos de otros dominios.** Por poner un ejemplo claro que lo explique, podemos tener un grupo de usuarios en un dominio, al que pertenecen cuentas de usuario de otros dominios. Si en un momendo dado cambiamos el nombre al grupo, el Maestro de Infraestructura es el encargado de notificar a los de otros dominios de este cambio. En un dominio, el Maestro de Infraestructura no puede ser al mismo tiempo Catálogo Global, salvo en el caso de un bosque de dominio único, debido al modo como ese DC consulta a los de otros dominios sobre los cambios de este tipo.

Los cambios de servidor para cada rol maestro se pueden hacer de dos formas: con las distintas herramientas gráficas de administración del AD (Usuarios y Equipos de AD, Sitios y Servicios de AD, Dominios y Confianzas de AD y cargando en una consola mmc el complemento de esquema), o bien con la herramienta de linea de comandos "ntdsutil". Para que te aparezca esta última tienes que instalar las support tools del CD del servidor. Además, con las herramientas gráficas se puede cambiar un rol de servidor siempre que tanto el de origen como el de destino estén en linea. Si el original hubiese caído, el forzamiento del cambio (seize) sólo se puede hacer con ntdsutil.

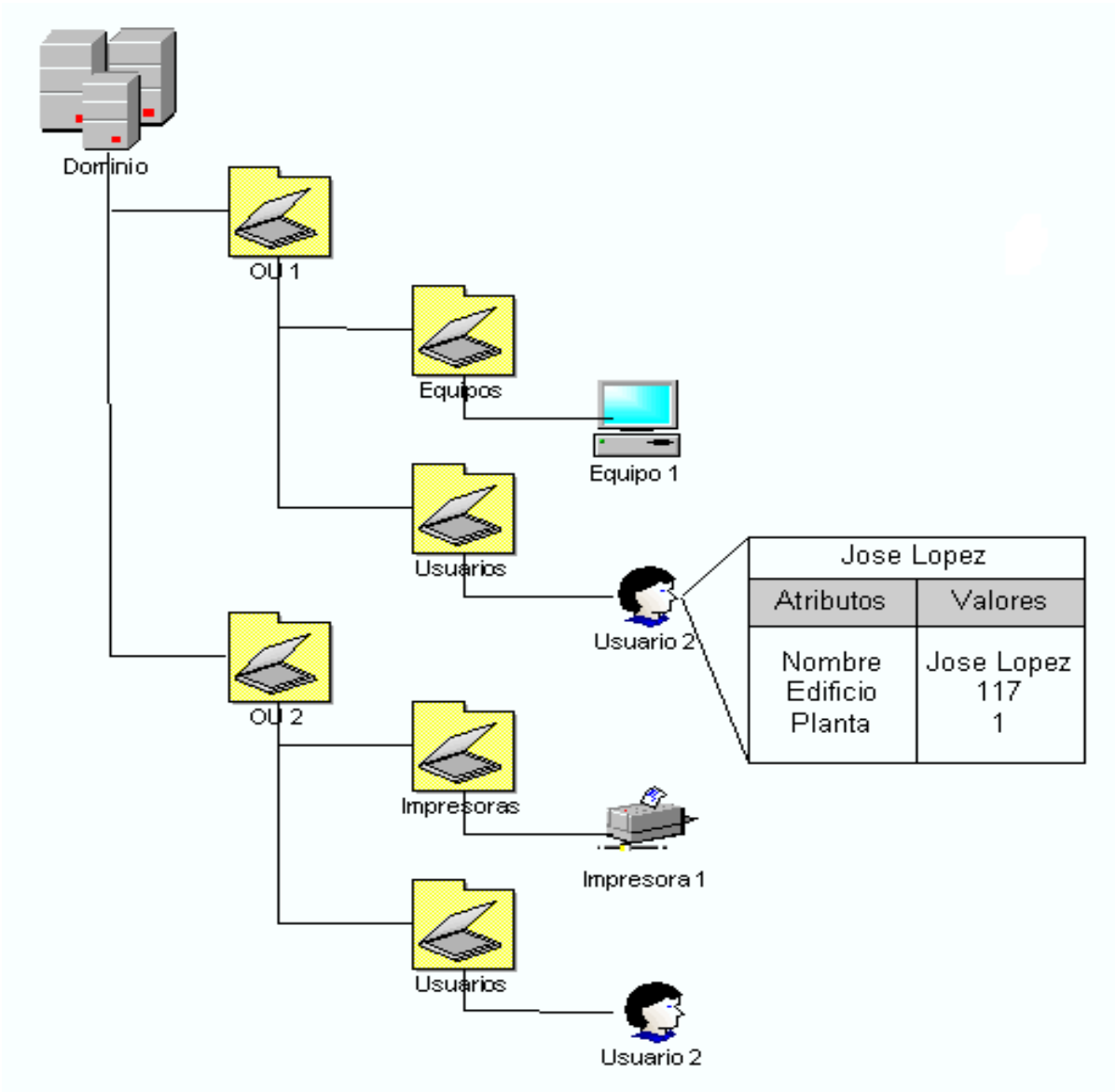
Por fin, hay otro rol que no se define como tal como **Maestro de Operaciones**, que es el de Catálogo Global y del que hablamos en el apartado anterior. Es un DC del dominio que además de tener toda la información de los atributos de todos los objetos de su propio dominio, tiene un subconjunto de los atributos de todos los objetos de todos los dominios del bosque.

11. Delegación das funcións de administración.

El administrador de AD (Active Directory o Directorio Activo), podrá asignar una serie de **tareas administrativas a los usuarios o grupos apropiados**. Puede asignar tareas administrativas básicas a grupos o usuarios normales, y dejar la administración de todo el bosque o de todo el dominio a cargo de los miembros de los grupos Administradores de dominio y Administradores de organización. Al delegar la administración, puede permitir a grupos de su organización ejercer un mayor control sobre los recursos de la red local.

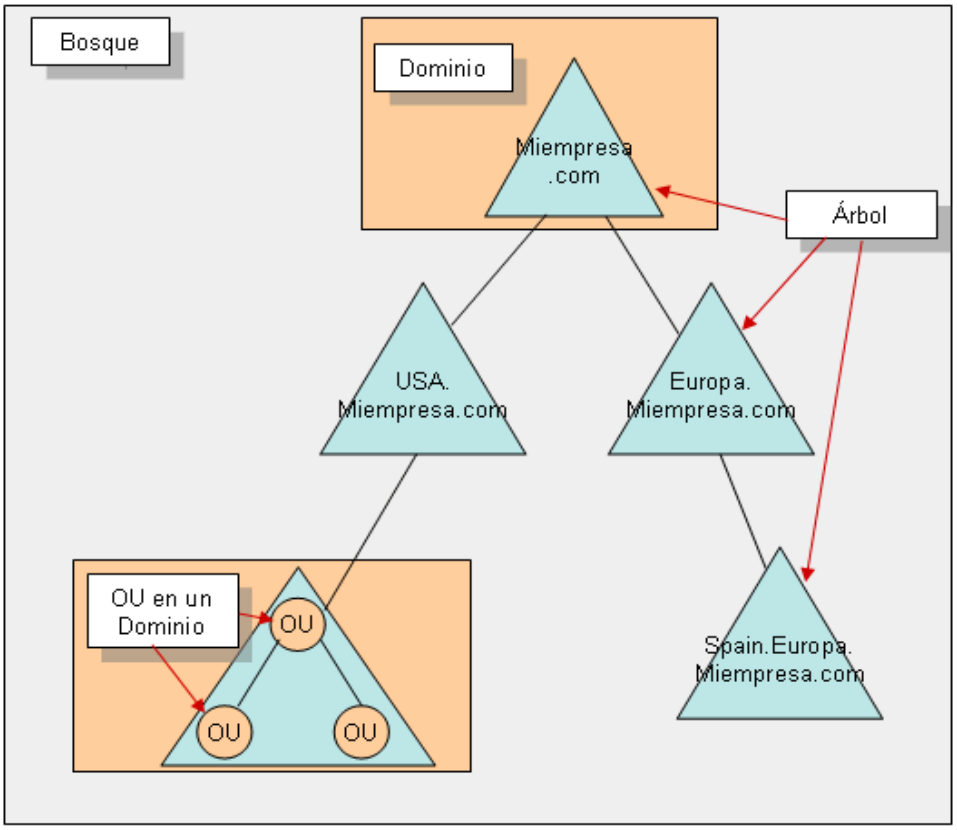
También puede ayudar a proteger la red de daños malintencionados o accidentales limitando la pertenencia de los grupos de administradores.

Puede delegar el control administrativo en cualquier nivel de un árbol de dominios **mediante la creación de unidades organizativas OU dentro de un dominio** y la delegación del control administrativo de determinadas unidades organizativas a usuarios o grupos específicos.



Por ejemplo, puede ser aconsejable crear una unidad organizativa que permita conceder a un usuario el control administrativo sobre todas las cuentas de usuario y de equipo en

todas las ramas de un departamento, como el de Recursos humanos. O bien, puede que desee asignar a un usuario el control administrativo de sólo algunos recursos dentro de un departamento, por ejemplo, las cuentas de equipo o las impresoras.



Active Directory **define permisos y derechos de usuario específicos** que se pueden usar **para delegar o limitar el control administrativo**. Mediante una combinación de unidades organizativas, grupos y permisos, puede definir el ámbito administrativo más adecuado para una determinada persona, que podría ser un dominio completo, todas las unidades organizativas de un dominio o una sola unidad organizativa

El control administrativo **se puede asignar a un usuario o grupo mediante el uso del Asistente para delegación de control o a través de la consola del Administrador de autorización**. Ambas herramientas le permiten asignar derechos o permisos a determinados usuarios o grupos.

Las tareas administrativas deben **delegarse con sumo cuidado y documentando** todas las asignaciones delegadas. Antes de delegar ninguna tarea, hay que asegurarse de que los usuarios a los que se les va a asignar el control de objetos cuentan con la formación adecuada.

Delegar funciones es una capacidad muy importante de Active Directory siempre y cuando se lleve a cabo de **forma muy granular y solo en algunas tareas administrativas a usuarios que no pertenezcan a alguno del los grupos con poderes administrativos** superiores.
En una organización pequeña probablemente no se haga delegación de tareas, pero es algo casi imprescindible en grandes organizaciones por razones obvias.

Se permite delegar definiendo:

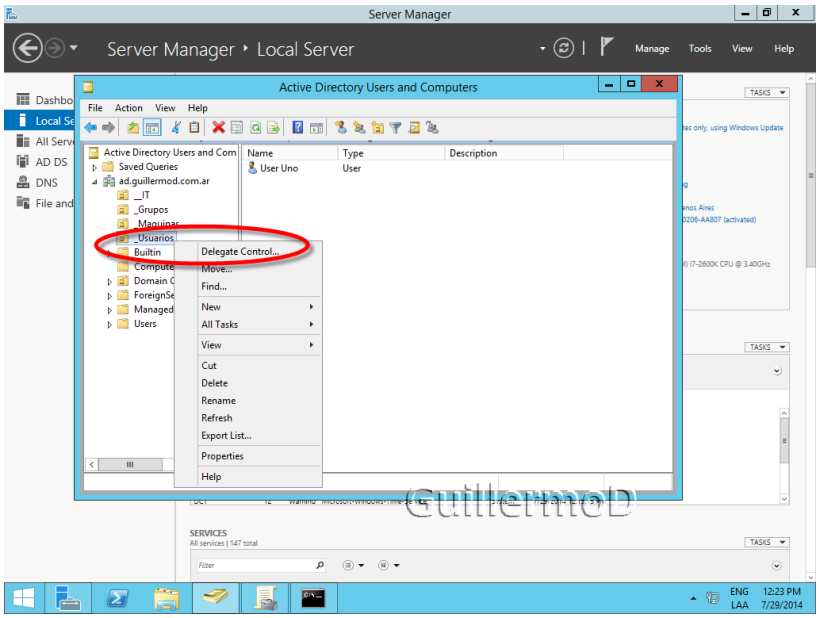
- Qué **tarea/s se delegará** o delegarán
- Sobre **qué tipo de objetos** (usuarios, grupos, GPOs, unidades organizativas, etc.)
- Sobre **qué partes del Dominio** (unidades organizativas, el Dominio, etc.)

Esto también responde a una consulta que se escucha con frecuencia: “que sea administrador, pero que no pueda …”. Lo cual es imposible, ya que si pertence al grupo de Administradores no hay forma de quitarle privilegios sobre la máquina o el Dominio

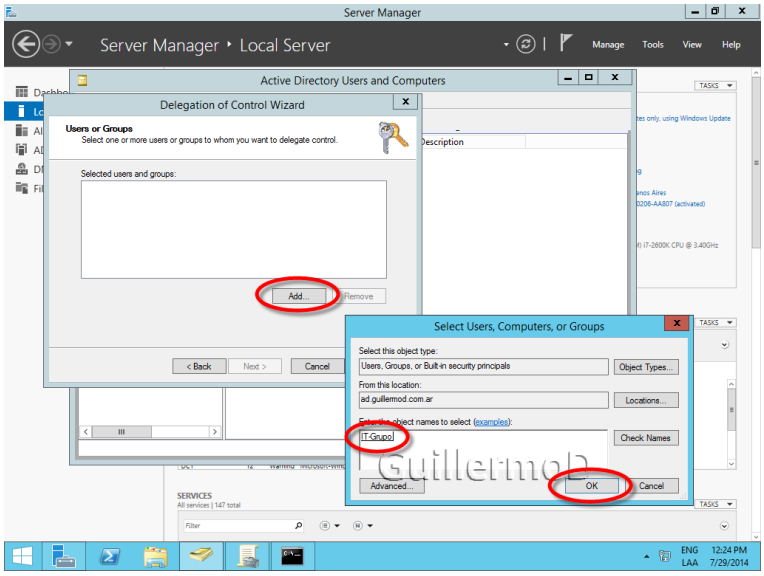
Veremos una demostración muy simple. En un ambiente de Active Directory, cada objeto, y en realidad cada propiedad de cada objeto tiene una **Lista de Control de Acceso** (*ACL = Access Control List)* **que determina quién y qué puede hacer con cada elemento**

Lo anterior está definido por omisión cuando se crea un Dominio, y está establecido en el Esquema (“Schema”) pero no es algo que no pueda cambiarse, aunque no absolutamente todos los objetos, para preservar la integridad.

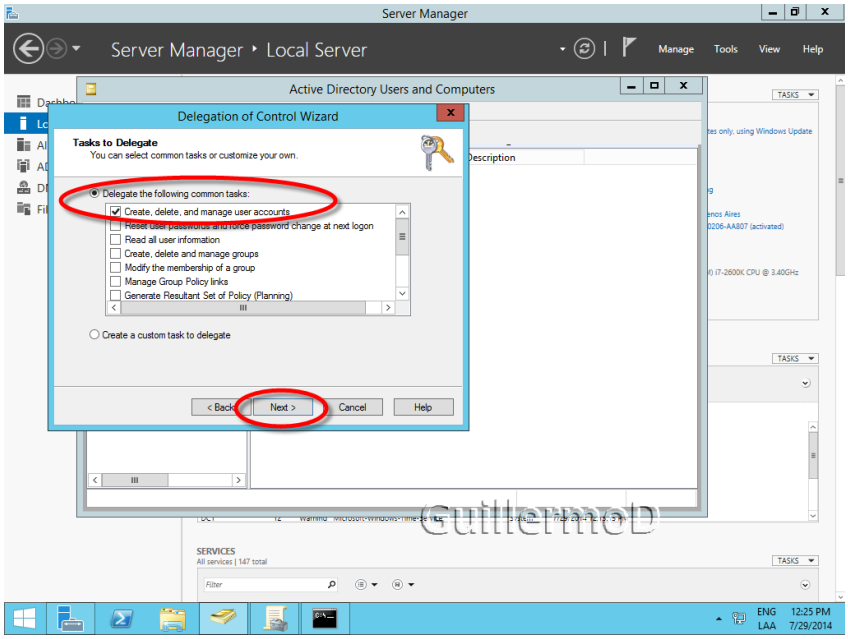
El sistema, por omisión no muestra la ficha de Seguridad de cada objeto, pero si por ejemplo en Usuarios y Equipos de Active Directory (“Active Directory Users and Computers”) habilitamos la opción *“View / Advanced Features”*, e ingresamos a las propiedades de un objeto, veremos quien tiene, y qué permisos tiene sobre el objeto



Al iniciar el asistente te solicita en que usuarios o grupos se delegarán las funciones.



Y elegimos la tarea que delegamos. En esta caso el alta, modificación y baja de usuarios.



Para finalizar, no se deben hacer cambios si uno no conoce exactamente cuáles serán las consecuencias de un cambio.

12. Realización de instalaciones desatendidas de servidores de dominio

Se realizará una práctica relacionada con este apartado en la medida de los posible.

Referencias

- *Página oficial de Microsoft MSDN*
- *Implantación de Sistemas Operativos RA-MA*
- *Sistema Operativos. Apuntes UPC*
- *Wikipedia*
- *windowsserver.wordpress.com*
- *Centro Recursos Formativos del Ministerio Educación*
- *www.msmvps.com*