**DSADD**

**Dsadd es una herramienta de línea de comandos que está integrado en Windows Server 2008/2012/2016/2019/2022. Está disponible si tiene los servicios de dominio de Active Directory (AD DS) función de servidor instalado.**

(DS de Directory Service )

DSADD –(ADD de **añadir**)

DSRM –(RM **Elimina Objetos**)

DSMOD—(MOD **Modifica objetos**)

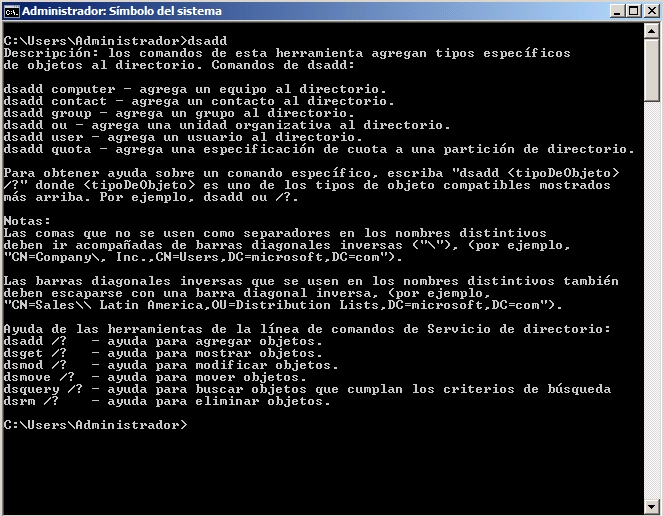
DSGET –(GET **Consulta u obtener atributos**)

DSMOVE –(MOVE **Mover objetos**)

**El comando dsadd nos permite agregar objetos al directorio desde la línea de comandos** lo que nos permite incluso llegar a automatizar la creación de una estructura de Grupos o Unidades Organizativas.

Para utilizar **dsadd** , debe ejecutar el **dsadd** comando desde un símbolo del sistema elevado. Para abrir un símbolo del sistema elevado, haga clic en **Inicio** , haga clic **en Símbolo del sistema**.

**Dsadd enter nos muestra la ayuda del comando dsadd.**



|  |  |
| --- | --- |
| [Dsadd user](http://technet.microsoft.com/en-us/library/cc731279.aspx) | Añade un único usuario en el directorio |
| [Dsadd group](http://technet.microsoft.com/en-us/library/cc754037.aspx) | Añade un único grupo en el directorio. |
| [Dsadd ou](http://technet.microsoft.com/en-us/library/cc770883.aspx) | Agrega una sola unidad organizativa al directorio. |
| [Dsadd computer](http://technet.microsoft.com/en-us/library/cc770883.aspx) | Añade un único equipo en el directorio.  [Dsadd HYPERLINK "http://technet.microsoft.com/en-us/library/cc754539.aspx" HYPERLINK "http://technet.microsoft.com/en-us/library/cc754539.aspx" HYPERLINK "http://technet.microsoft.com/en-us/library/cc754539.aspx"computer](http://technet.microsoft.com/en-us/library/cc754539.aspx) |
| [Dsadd quota](http://technet.microsoft.com/en-us/library/cc754339.aspx) | Añade una especificación de cuota a una partición de directorio. |

**Ayuda del comando dsadd para Unidades Organizativas**  
dsadd ou /?

**Ayuda del comando dsadd para Usuarios**  
dsadd user /?

**Ayuda del comando dsadd para Grupos**  
dsadd group /?

**Ayuda del comando dsadd para Equipos**  
dsadd computer /?

**Ayuda del comando dsadd para Contactos**  
dsadd contact /?

**Ayuda del comando dsadd para Cuotas de directorio**  
dsadd quota /?

Nombre de usuario: Laura  
Unidad Organizativa: Aulas,Segundo  
Dominio: ana.local

dsadd ou /?

**1º Primero creamos una Unidad Organizativa**

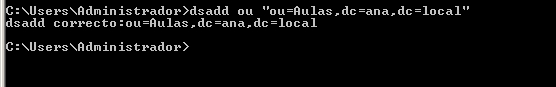
**Unidad Organizativa = OU**

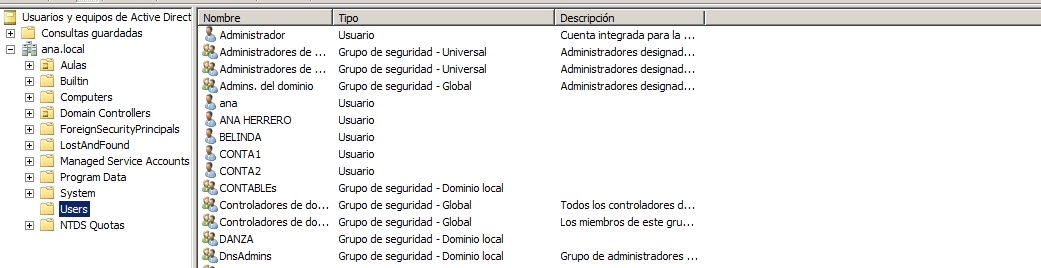
Para crear una unidad organizativa tenemos que utilizar **dsadd ou** y después hay que darle el nombre distintivo. El nombre distintivo consta de la unidad organizativa o unidades organizativas donde queremos crear nuestra unidad y el nombre de dominio. El nombre de dominio si tenemos un dominio que se llama **ana.local**  es **dc=ana,dc=local**

**dsadd ou “ou=Aulas,dc=ana,dc=local”**

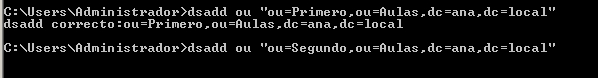
Al ejecutar esto veamos lo que estamos haciendo por partes:

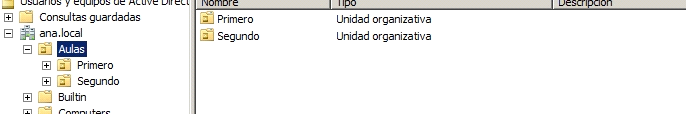
* **dsadd**  
  Este comando ordena la creación de un objeto en el directorio
* **ou**  
  Este modificador indica que el objeto que vamos a crear es una Unidad Organizativa
* **Después hay que indicar** el nombre distintivo entre “ “ de la unidad organizativa que queremos crear.
* **ou=Aulas,**  
  Aquí estamos indicando el nombre que tendrá la Unidad Organizativa
* **dc=ana,dc=local**  
  Aquí indicamos que el objeto será creado en el dominio ana.local





**Ahora vamos a crear dos unidades organizativas dentro llamas primero y segundo.**





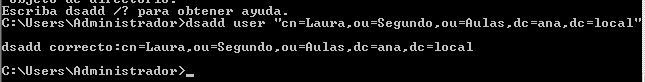
**Ahora utilizaremos dsadd para crear un usuario en la Unidad Organizativa Segundo que hemos creado antes.**

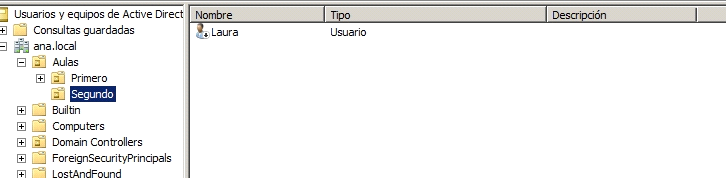
dsadd user /?

**Dsadd user: crea usuarios.**

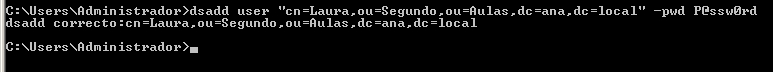
dsadd user “cn=Laura,ou=Aulas,ou=Segundo,dc=ana,dc=local”

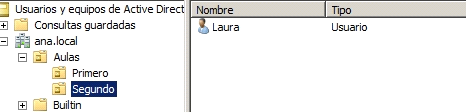
* **dsadd**  
  Este comando ordena la creación de un objeto en el directorio
* **user**  
  Este modificador indica que el objeto que vamos a crear es un Usuario
* **Después hay que indicar** el nombre distintivo entre “ “ del usuario a crear.
* **cn=Laura,**  
  Aquí estamos indicando el nombre que tendrá el Usuario
* **ou=Aulas,ou=Segundo**  
  Aquí estamos indicando en que Unidad Organizativa vamos a crear el usuario
* **dc=ana,dc=local**  
  Aquí indicamos que el objeto será creado en el dominio ana.local





Nos crea el usuario Laura en la unidad organizativa Segundo, pero la cuenta del usuario Laura la deja deshabilitada, gráficamente aparece con una flechita hacía abajo.

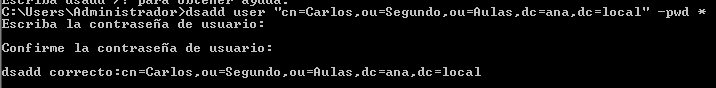


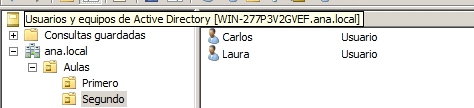


Nos crea el usuario Laura en la unidad organizativa Segundo, y ahora la cuenta del usuario Laura sí que está habilitada porque tiene contraseña.

Y pasamos a analizar este comando…

* **dsadd**  
  Este comando ordena la creación de un objeto en el directorio
* **user**  
  Este modificador indica que el objeto que vamos a crear es un Usuario
* **Después hay que indicar** el nombre distintivo entre “ “ del usuario a crear.
* **cn=Laura,**  
  Aquí estamos indicando el nombre que tendrá el Usuario
* **ou=Aulas,ou=Segundo**  
  Aquí estamos indicando en que Unidad Organizativa vamos a crear el usuario
* **dc=ana,dc=local**  
  Aquí indicamos que el objeto será creado en el dominio ana.local
* **-pwd \***  
  Este modificador lo que nos permite es especificar el password después de lanzar el comando, la principal ventaja es que no tendría que estar integrada en un fichero por lotes y no sería visible, aunque en un caso de crear muchos usuarios de forma masiva lo mejor es establecer que el usuario cambia la contraseña en el siguiente inicio de sesión.

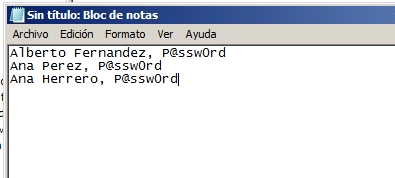


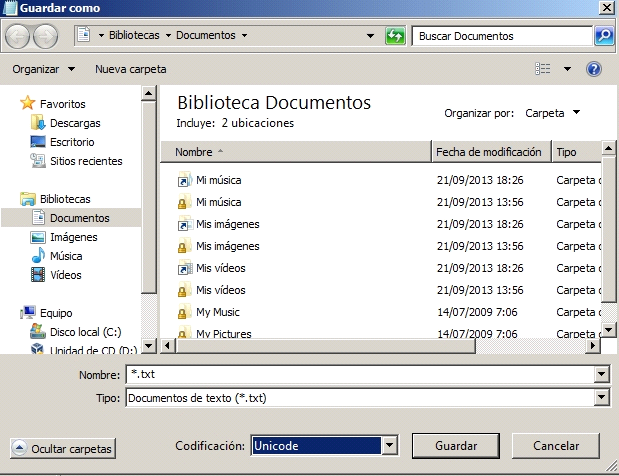


Nos crea el usuario Carlos en la unidad organizativa Segundo, y ahora la cuenta del usuario Carlos sí que está habilitada porque tiene contraseña, pero la contraseña al teclearla no se ve, permanece oculta. Esto se debe a que hemos indicado el parámetro **-pwd** con un asterisco (**\***).

Ahora que hemos aprendido a crear usuarios y Unidades organizativas desde línea de comando, vamos a Automatizar el proceso.

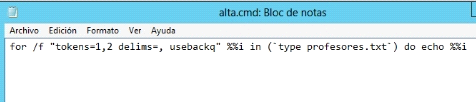
Para ello con el notepad creamos un fichero con nombres de personas.





Lo guardamos con codificación Unicode y con un nombre alumnos.txt.

Ahora creamos otro fichero llamado alta.cmd con el notepad con los siguientes datos. Tipo ANSI



UN BUCLE:

Sintaxis:

FOR **/F** ["*options*"] %%*parameter* IN (*filenameset*) DO *command*

FOR **/F** ["*options*"] %%*parameter* IN ("Text string to process") DO *command*

%%i es la variable I que va a ir tomando los distintos valores que se le van a pasar en el fichero profesores.txt

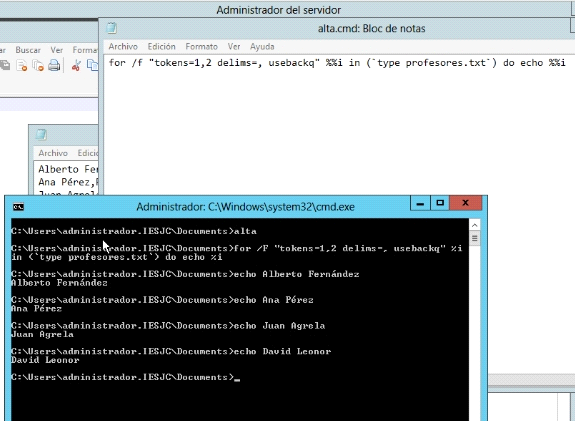
Do detrás del do se le dice lo que queremos que ejecute con cada uno de los valores de i.

Tokens= es como decirle que tenemos dos campos definidos en el fichero de datos de alumnos

Delims= le decimos que carácter vamos a utilizar para separar un campo de otro, es decir para separar el nombre del alumno de la contraseña utilizo una,

Usebackq= este delimitador es para decirle que en el type utilizo las comas invertidas.

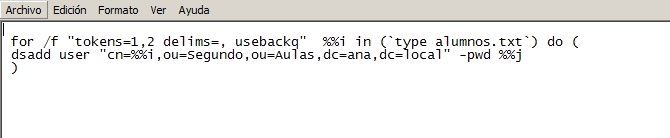
Type alumnos.txt = le decimos que nos visualice el contenido del fichero indicado.

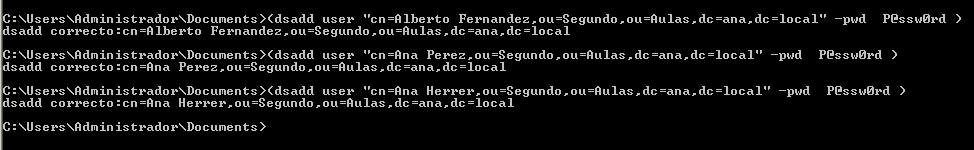


Para ejecutar el fichero ponemos alta.

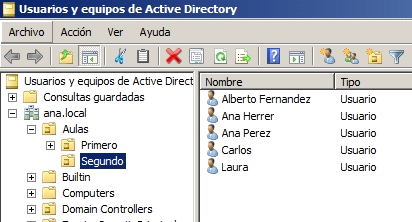
Ya vemos en la imagen como nos coge los nombres del fichero y nos los muestra, ahora solo nos queda decirle que en vez de mostrar los nombres cree las cuentas de cada uno de los usuarios en el sistema.

Esto se lo decimo con el do( dsadd user…y aquí la sintaxis del dsadd)





Si ahora nos vamos a usuarios, tendremos los usuarios creados



ESTRATEGIAS PARA NOMBRAR CUENTAS.

Crear nombres de cuentas de usuario parece una tarea trivial, y efectivamente lo es cuando tenemos que crear un par de usuarios, pero se transforma en una tarea mucho más complicada cuando queremos crear cientos, o incluso miles de cuentas de usuario.

Pongamos por ejemplo que queremos crear una cuenta de usuario para cada alumno del instituto

Francisco Romero Vargas, que cuenta con unos 900 alumnos. Evidentemente no podemos usar el nombre (sin apellidos) como nombre de cuenta, o acabaremos con cuentas como Jennifer32, dado que habrá muchos usuarios con el mismo nombre.

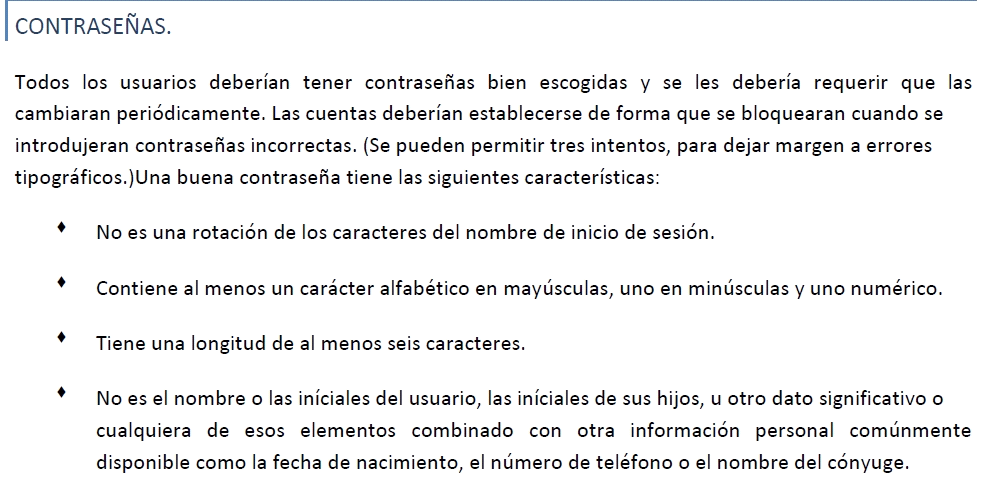
Es importante crear una estrategia de denominación de cuentas para nuestros bosques y dominios, creando unas convenciones para nombrar cuentas. Ejemplos validos de estrategias podrían ser, por ejemplo:

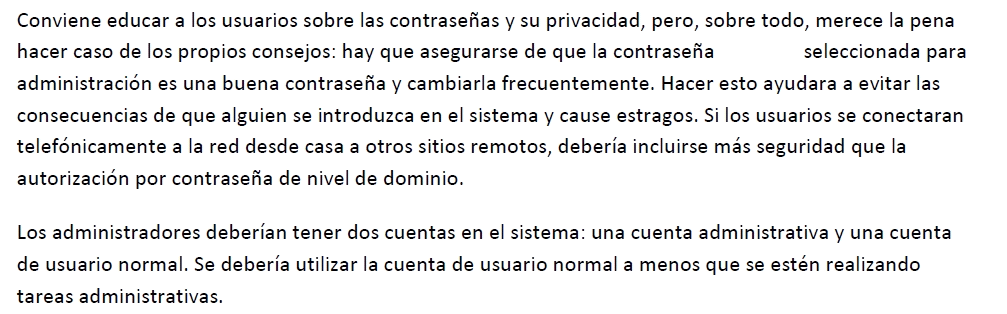
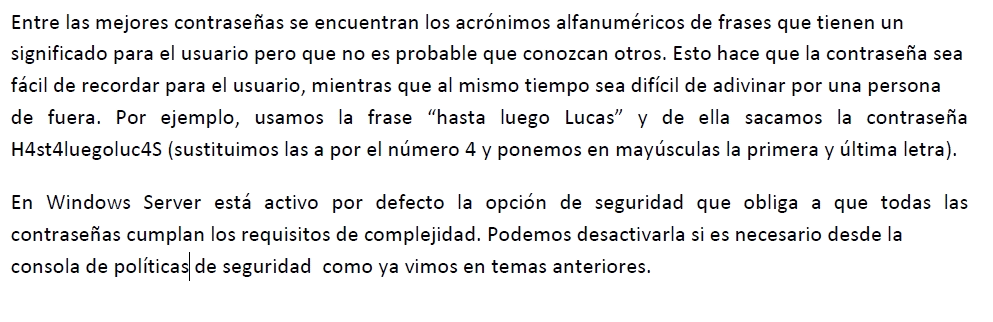
* Usar los 2 pfor /?rimeros caracteres del nombre, los 2 caracteres primeros del 1º apellido y los 2 caracteres primeros del 2º apellido.
* Usar 3 caracteres para indicar el curso del alumno, los 3 caracteres primeros del nombre y los 2 caracteres primeros del 1º apellido.
* Usar los 4 caracteres primeros del nombre del alumno, la inicial del 1º apellido, la inicial del 2º apellido y los 2 últimos números del DNI del alumno.

Fijaros como la 2ª estrategia tiene la ventaja de que la simple cuenta de usuario nos da información sobre el curso del alumno, por lo que tenemos más control que en los otros dos casos.

Aparte de facilitar la creación de las cuentas de usuario, estas estrategias tienen la gran ventaja de que nos permiten crear programas para crear automáticamente cuentas de usuario. Así, podemos llegar a una empresa que cuenta con 200 empleados a los que tenemos que crear una cuenta de usuario.

Podemos crear un script o programa que leyendo una lista de los nombres de los usuarios nos cree directamente las cuentas, usando nuestra estrategia definida para nombrar las cuentas.





Ya vimos anteriormente como mediante el comando dsadd podíamos crear cuentas de usuario. Pues conjuntamente con este comando, también podemos usar los siguientes comandos para la administración de usuarios:

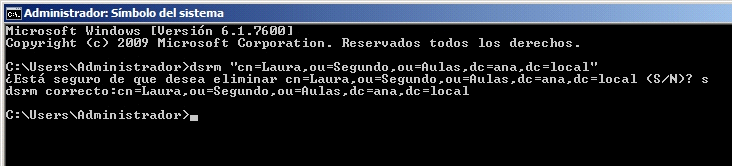
* Dsadd. Añade usuarios.
* Dsrm. Elimina usuarios.
* Dsmod. Modifica usuarios.
* Dsmove. Mueve usuarios.
* Dsquery. Busca equipos en el directorio que coincidan con los criterios de búsqueda
* Dsget. Muestra información sobre los usuarios.

Si escribimos dsrm user /? desde la línea de comandos nos aparecerá una ayuda sobre las opciones de dsrm. De igual modo podemos hacerlo con las restantes órdenes.

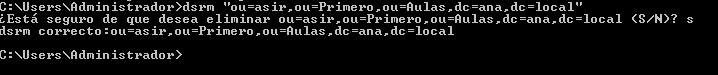
* **Eliminar una cuenta de usuario.**

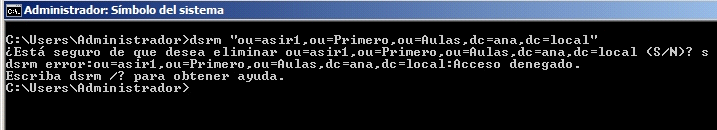
*dsrm “nombre distintivo del usuario”*

*Eliminamos la cuenta del usuario.*



Dsrm “Nombre distintivo de la unidad organizativa a borrar”

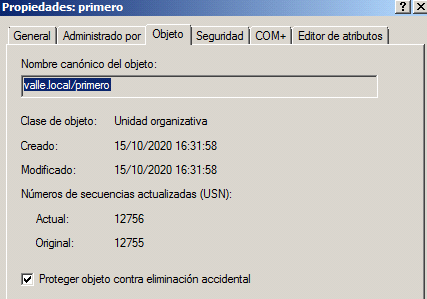


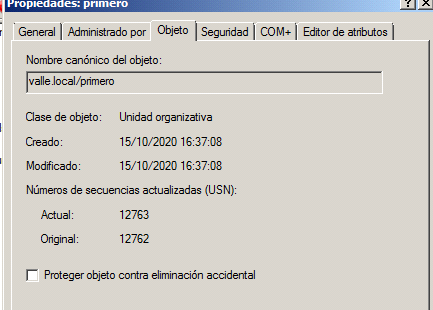


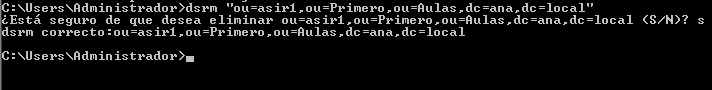
Ahora si nos da error puede ser porque esa unidad organizativa está protegida contra eliminación accidental.

Como eliminar la protección contra eliminación accidental: Nos vamos a la unidad organizativa y le quitamos la eliminación accidental y después nos dejara eliminarla.



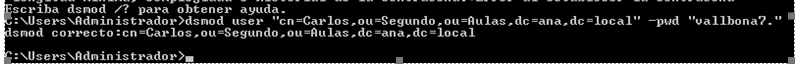






* Restablecer la contraseña de un usuario. El formato de la orden sería:

*dsmod user “nombre distintivo del usuario” –pwd “nueva contraseña”*



* Para restablecer(modificar) la contraseña y obligarle a cambiar su contraseña cuando a continuación inicie sesión en la red escriba.

Dsmod user “nombre distintivo “ –pwd P@ssword –mustchpwd yes

--Para que esto afecte a varios usuarios.

Dsmod user “nombre distintivo “ “nombre distintivo” –pwd P@ssword –mustchpwd yes

* Para que el usuario no pueda cambiar su contraseña.

Dsmod user “nombre distintivo “ -canchpwd (no/yes) -canchpwd no

* Para que la contraseña del usuario nunca expire.

Dsmod user “nombre distintivo “ -pwdneverexpires (yes/no) yes

* Para que la cuenta del usuario expire en un número de días a partir de hoy.

0--> expira hoy

+1--> expira dentro de un día

-1 --> expiro en el pasado

never--> no expira nunca

Dsmod user “nombre distintivo “ -acctexpires 0 /+1/-1 never.

* Deshabilitar una cuenta de usuario.

*dsmod user “nombre distintivo del usuario” –disabled yes*

--Deshabilitar varias cuentas al mismo tiempo.

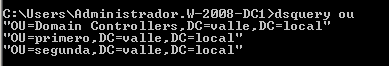
*dsmod user “nombre distintivo del usuario” “nombre distintivo del usuario” –disabled yes*

*Para cambiar la descripción de varias unidades organizativas al mismo tiempo -. Dsmod Ou “OU=…” “OU=……” “OU=…” -desc “esto es una prueba de una unidad organizativa”*

*DSQUERY*

*Consulta al AD mediante el uso de criterios de búsqueda. Cada uno de los comandos dsquery encuentra objetos de un tipo de objeto especifico, con la excepción de dsquery \* que puede consultar cualquier tipo de objeto.*

*Dsquery ou Consulta ou nos retorna todas las unidades organizativas.*



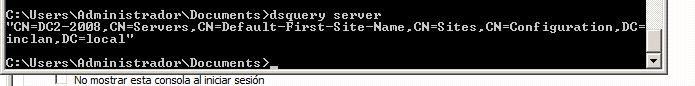
*Dsquery computer Consulta pc. Busca equipos en el directorio que coincidan con los criterios de búsqueda*

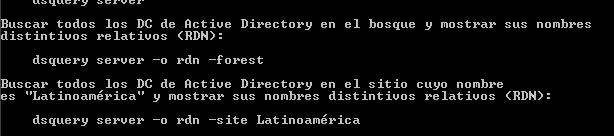


*Dsquery computer –name DC-200\* Busca los equipos cuyo nombre comience por DC-200*

*DSQUERY computer “*ou=Aulas,dc=ana,dc=local” nos retorna los ordenador que tengamos en esa unidad organizativa.

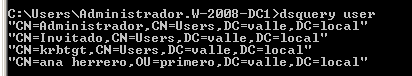
*Dsquery server*







*Dsquery user Localiza los usuarios.*

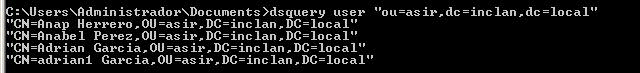


*Dsquery user –name “Sara\_\*” Localiza los usuarios cuyo nombre sea Sara\_ y después cualquier cosa.*



*Dsquery user –name “Sara \*” Localiza los usuarios cuyo nombre sea Sara luego un espacio en blanco y a partir de ahí cualquier caracter.*

* *Dsquery user “OU=USUARIOS,dc=inclan,dc=local” Nos visualiza a todos los usuarios que hay en esta unidad organizativa.*

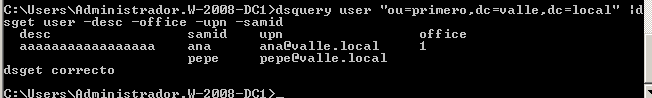


* *Dsquery user “OU=USUARIOS,dc=inclan,dc=local” | dsget user –desc*

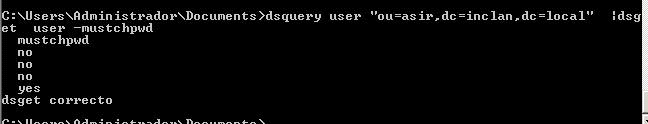
*Con dsget obtenemos las propiedades en este caso desc.*

Texto

Descripción generada automáticamente



* *Dsquery user “OU=USUARIOS,dc=inclan,dc=local” | dsget user – mustchpwd*



*Dsquery user -name “pepe” |dsmod user -office “SANTANDER”*





*Dsquery user –name “Sacra \*” | dsmod user –office “Alicante” –disabled yes. Utilizamos el dsmod para modificar los atributos office y disabled de los usuarios que nos devuelve el dsquery.*

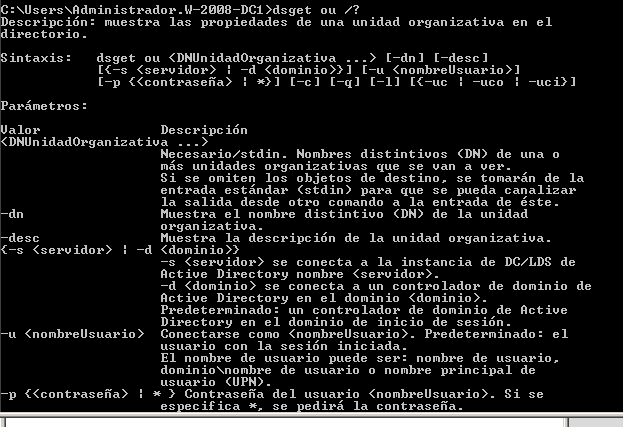
*Modifica la Password para todos los usuarios de una Unidad Organizativa.*

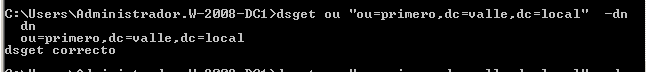
*Dsquery user “OU=USUARIOS,dc=inclan,dc=local” |dsmod user -pwd Password1 –mustchpwd yes.*

*DSGET /? Consultas al LADP desde línea de comando*

Dsget es un comando que nos permitirá obtener información de los objetos del Directorio Activo.

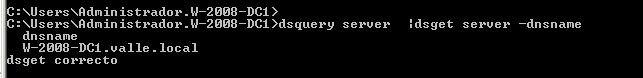
* *Dsget OU Muestra las diversas propiedades de una U.O*

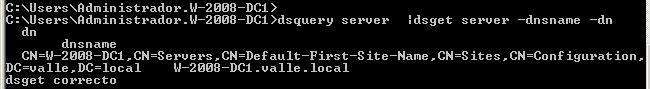
Podemos ver los atributos de -dn y -desc de las unidades organizativas.





* *Dsget server Muestra las propiedades de un controlador de dominio.*

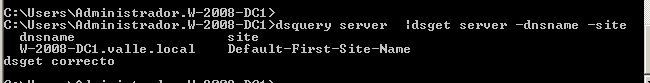


 Texto

Descripción generada automáticamente

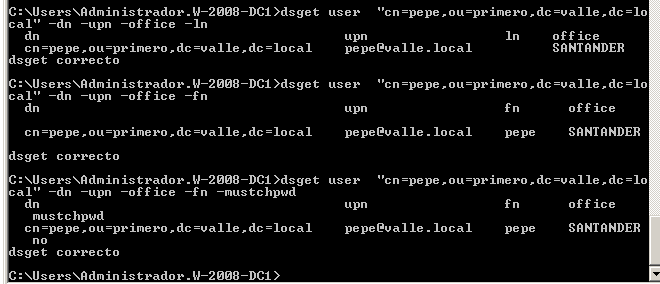
Texto

Descripción generada automáticamente



* *Dsget user Muestra las propiedades de un usuario*





*Ejemplo segundo de creación de usuarios.*

*Vamos a crear un fichero distinto llamado alumnos.txt y vamos a poner comentarios dentro del fichero y que luego el script no tenga en cuenta los comentarios que se establece con #.*

*# alumnos nuevos*

*# formato*

*# 1.nombre; 2. Apellido; 3. Nombre upn;*

*#*

*# separador;*

Nombre 1º Apellido 2º Apellido UPN Contraseña

*Carlos;Alcazar;alcazar.carlos*

*Miguel;Bautista;bautista.miguel*

*Juan Carlos;Bohoyo;bohoyo.jcarlos*

*Ahora creamos el fichero alta-alumnos.cmd*

*For /f “eol=# tokens=1,2,3 delims=; skip=1 usebackq” %%a in (`type alumnos.txt`) do (*

*Dsadd user “cn=%%a %%b,ou=alumnos,dc=inclan,dc=local” –pwd P@ssw0rd –upn* [*%%c@inclan.local*](mailto:%25%25c@inclan.local)*)*

*Nombre upn Nombre principal único en todo el bosque* [*apellido.nombre@inclan.local*](mailto:apellido.nombre@inclan.local)*.*

*Eol=# con esto le decimos que desde que aparezca este carácter hasta final de línea son comentarios.*

*Tokens=1,2,3 que hay tres campos*

*Skip=numero, siendo el numero de fila que no queremos que no ejecute.*

*CREACIÓN DE USUARIOS*

*ORGANIZACIÓN CON UNIDADES ORGANIZATIVAS.*

*En un centro educativo podríamos trabajar con UO profesores y alumnos, y equipos.*

*Administrar los equipos organizándolos por Aulas. No existe una solución ideal para todos los casos, sino según cada caso.*

*Cuando veamos que a una U.O le podemos asociar una directiva de seguridad, pues dependiendo de nuestras necesidades de seguridad nos llevara a crear unas U.O y ordenar los objetos dentro de esas U.O para facilitarnos luego aplicar las directivas de grupo.*

*Estas directivas que se aplican a una U.O pues entonces todas las U.O que hay debajo heredan esas directivas.*

*Depende un poco de como vayamos a aplicar luego la seguridad si para mí los profesores independientemente del departamento donde se encuentre va a tener las mismas necesidades, lo lógico va a ser complicarnos lo menos posible y tenemos una sola U.O donde se encuentran todos los usuarios profesores. Si los Alumnos si todos van a tener las mismas necesidades, misma seguridad las mismas características los ponemos en la misma U.O*

*Que queremos separar por ciclos formaticos por departamentos pues crearemos esas U.O colgando de Alumnos*

**PowerShell**

Desde un primer momento, Microsoft ha incluido una línea de comandos que ha evolucionado a lo largo de los diferentes sistemas operativos que ha ido desarrollando. Actualmente, está incluido en todos los sistemas operativos de Microsoft.

Las órdenes en PowerShell (PS) son muchas, ya que su entorno ha crecido de tal forma que se puede administrar casi cualquier característica de windows y de otras aplicaciones sobre él.

PS tiene todas las características de cmd.exe, pero, además, realiza otras tareas que permiten el acceso a elementos internos más profundos de Windows. Otra particularidad que lo diferencia de la línea de comandos y los sistemas de tipo Unix es que está basado en objetos, de manera que se obtiene la información en forma de objetos en lugar de texto.

PS no se define como una línea de comandos exclusivamente, sino que también actúa como entorno basado en scripts, donde se pueden desarrollar comandos propios o combinar varios para facilitar la gestión del sistema.

Existen dos opciones a la hora de ejecutar PS:

* Con entorno gráfico llamado PowerShell ISE (Integrated Scripting Enviroment)
* Con una consola de texto que se parece a la línea de comandos tradicional.

Todos los comandos reciben el nombre de cmdlets (de command-let) Para conocer todos los cmdlets que hay disponibles en el sistema, basta

get

Get-Command

Get-Command -Commandtype cmdlet | Measure-object

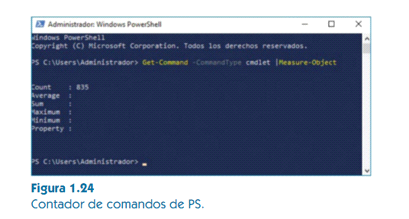


Imagen que contiene Tabla

Descripción generada automáticamente

Aplicación

Descripción generada automáticamente con confianza baja

**El Get-ChildItem cmdlet obtiene los elementos en una o más ubicaciones especificadas**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Set-location establece la ubicación actual en la unidad c:\**

Tabla

Descripción generada automáticamente

Imagen que contiene Texto

Descripción generada automáticamente

**h---ocultos**

**s – archivo de sistema**

**d- directorio**

**a—archivo**

**r-- Lectura**

**Bootmgr --- gestor de arranque de windows.**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

Tabla

Descripción generada automáticamente

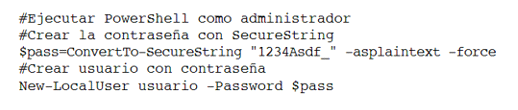
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Administrar usuarios**

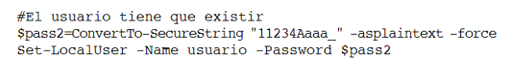
Una cuenta de usuario consiste en un nombre y una contraseña de forma que estos dos elementos forman un conjunto de credenciales que sirven para identificar a una persona en el sistema.

* Para crear un usuario es necesario hacerlo como administrador:

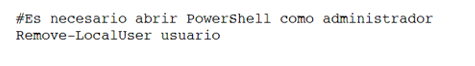


New-ADuser usuario -Password $pass

* Para Cambiar la contraseña de un usuario:



* Para eliminar un usuario:



Remove-ADUser usuario

**Administrar procesos desde PowerShell**

Igual que en el administrador de tareas de Windows, con PowerShell podremos gestionar los procesos prescindiendo de la interfaz gráfica del sistema.

* Consultar procesos: Para obtener la lista con los procesos que se están ejecutando en el equipo. Se puede emplear la opción |more para pausar y procesar la información poco a poco en caso de obtener una salida demasiado larga.

**Get-Process |more**

**Get-Process -Name win\*** Con este comando filtramos todos aquellos que empiecen por “Win”

Si queremos obtener una información detallada sobre un proceso o grupo de procesos, enviamos la salida del Get-Process a format-List, obteniendo una salida donde las propiedades de un objeto aparecen en una nueva lista.

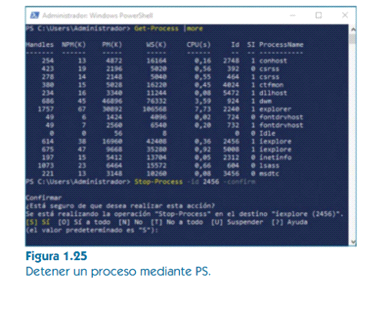
**Get-Process -Name explorer | Format-List \***

* Detener un proceso: para forzar la detención de un proceso recurrimos a Stop-Process como por ejemplo Stop-Process -name Firefox\* donde se detendrán todos los procesos cuyo nombre comience por “Firefox”.

Otra opción para obtener un proceso es basarse en su número de proceso o id. Por ejemplo, si partimos de un proceso con el id 2456, el comando que se ha de ejecutar sería:

**Stop-Process -id 2456 -confirm**

Con el argumento -confirm, forzamos a que nos pida confirmación anted de detener el proceso.

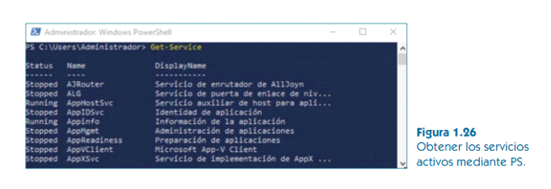


* Iniciar un nuevo proceso: para iniciar uno o más procesos utilizamos **Start-Process**, incluyendo, a continuación, el nombre de un programa o de un script. Por ejemplo, para ejecutar el bloc de notas, debemos escribir la siguiente orden:

**Start-Process Notepad.exe**

**Administrar Servicios desde PowerShell**

Obtener los servicios en un equipo local o remoto es muy sencillo. Para ello, se debe emplear **Cmdlet Get-Service**. Del mismo modo que ocurre con Get-process.



Se puede filtrar por nombre, incluso empleando el asterisco como carácter comodín, tal y como se empleó al administrar procesos:

**Get-Service -Name se\***

El parámetro ComputerName del Get-Service sirve para obtener los servicios en equipos remotos. Acepta varios valores y caracteres comodín, por lo que es posible obtener los servicios en varios equipos de una sola vez. Por ejemplo, el comando Get-Service -ComputerName ServerPPal obtiene los servicios en el quipo remoto ServerPPal.

Detener, iniciar, suspender y reiniciar los servicios es sencillo y muy intuitivo. Todos los cmdlet asociados a servicios tienen el mismo formato general, donde se especifica el nombre común o el nombre para mostrar y caracteres comodín como valores. Por ejemplo, si queremos gestionar el administrador de trabajos de impresión, podremos usar los siguientes comandos:

