

Gestión de grupos y usuarios en Windows.

Contents

1	Definición de cuentas y grupos	1
1.1	Cuentas de usuario locales	1
1.2	Grupos locales	1
2	Gestión de usuarios y grupos desde el entorno gráfico	1
2.1	Gestión de usuarios con la herramienta de configuración	1
2.1.1	Crear una nueva cuenta de usuario local	1
2.1.2	Cambiar el tipo de cuenta	2
2.1.3	Eliminar cuenta	3
2.2	Administración de usuarios y grupos con herramientas avanzadas	3
3	Gestión de usuarios y grupos desde el CMD	15
3.1	Gestión de usuarios	15
3.1.1	Administrar usuarios en Windows usando Net User	15
3.1.2	¿Qué es Net User y para qué se utiliza?	15
3.1.3	Funcionamiento de la herramienta Net User	16
3.2	Gestión de grupos	17
3.2.1	Gestión de grupos locales para Windows desde la línea de comandos.	17
3.2.2	Ayuda	17
3.2.3	Lista de grupos	17
3.2.4	Crear un grupo	17
3.2.5	Borrado de grupos	17
3.2.6	Lista de usuarios de grupos	17
3.2.7	Añadir usuario a un grupo	18
3.2.8	Añadir comentario al grupo	18
4	Administrar usuarios y grupos desde Powershell	18
4.1	Resumen de comandos	18
4.2	Gestión de usuarios	19
4.2.1	Obtener la lista de cuentas de usuario	19
4.2.2	Crear una cuenta de usuario local	20
4.2.3	Modificar una cuenta de usuario local	20
4.2.4	Renombrar una cuenta de usuario local	21
4.2.5	Activar una cuenta de usuario local	22
4.2.6	Desactivar una cuenta de usuario local	22
4.2.7	Eliminar una cuenta de usuario local	22
4.3	Gestión de grupos	23
4.3.1	Obtener la lista de grupos locales	23

- 4.3.2 Crear un grupo local 24
 - 4.3.3 Modificar las propiedades de un grupo local 25
 - 4.3.4 Renombrar un grupo local 25
 - 4.3.5 Eliminar un grupo local 25
- 4.4 Administrar los miembros de un grupo 26
 - 4.4.1 Agregar miembros en un grupo local 26
 - 4.4.2 Obtener los miembros presentes en un grupo local 27
 - 4.4.3 Eliminar miembros de un grupo local 27

1 Definición de cuentas y grupos

1.1 Cuentas de usuario locales

Una **cuenta de usuario** en un sistema operativo es la información que almacena el sistema para identificar a un usuario del mismo. Junto con la información de identificación de ese usuario, se suele guardar más información, como por ejemplo su contraseña que servirá para autenticar al usuario cuando inicie sesión en el sistema. Por supuesto, la contraseña no se guarda en «texto claro», es decir tal cual es, sino que se suele guardar cifrada para que un atacante no pueda descubrir la contraseña de un usuario.

De esta forma, cuando el usuario inserta una contraseña, el sistema la cifra y comprueba si la contraseña cifrada que ha insertado el usuario que quiere iniciar sesión coincide con la contraseña que tiene guardada.

Además como ya sabemos en sistemas operativos multiusuario como Windows, GNU/Linux o MacOS, cada usuario tiene un directorio propio o directorio personal. En Windows se denomina a este directorio perfil de usuario y por defecto se guarda dentro del directorio Users, que estará normalmente en el raíz de la unidad C:, con el nombre del usuario.

Con una cuenta de usuario, un usuario puede iniciar sesión, si tiene permiso para ello. Una vez inicia sesión, el sistema operativo conoce qué usuario está trabajando en el sistema en todo momento. Si intenta realizar cualquier operación, podrá darle permiso para hacerlo o no en función de los permisos que se hayan configurado.

1.2 Grupos locales

El problema de las cuentas de usuario surge en un sistema con muchos usuarios que comparten muchos permisos. La solución pasa por agruparlos utilizando grupos de usuarios. De hecho, en el mundo real, estamos acostumbrados a trabajar con agrupaciones de empleados, clientes, usuarios, etc.

Por ejemplo, todos los alumnos del curso de Sistemas Operativos comparten ciertos permisos. Podríamos establecer un permiso de acceso al aula de Sistemas operativos diciendo que: «Todos los alumnos del módulo de Sistemas Operativos pueden entrar en el aula 01». Ahora bastará con tener una hoja donde figuren qué alumnos pertenecen al módulo de Sistemas Operativos para controlar el acceso.

La otra opción hubiera pasado por nombrar uno a uno cada alumno matriculado en el módulo de Sistemas Operativos.

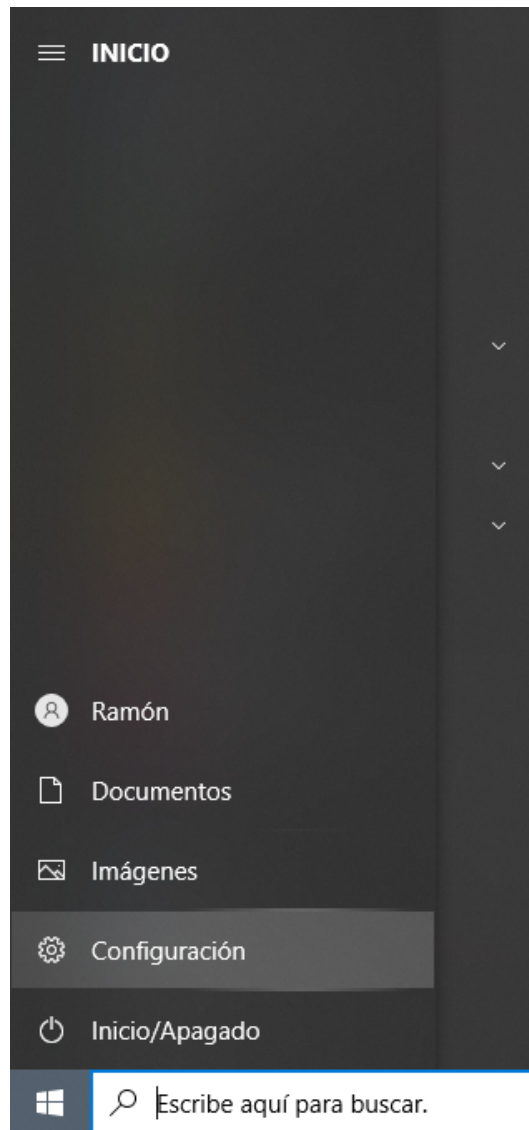
Al final, los grupos nos facilitan la administración de permisos en un sistema operativo multiusuarios. De esta forma, tan solo tenemos que gestionar qué usuarios pertenecen a cada grupo y después aplicar permisos sobre grupos en lugar de usuarios individuales.

2 Gestión de usuarios y grupos desde el entorno gráfico

2.1 Gestión de usuarios con la herramienta de configuración

2.1.1 Crear una nueva cuenta de usuario local

1. Pulsa Inicio y elige Configuración.
-



2. Se abre la ventana de Configuración. Pulsa el botón Cuentas.
3. Elige Familia y otros usuarios.
4. Pulsa en el botón Agregar otra persona a este equipo.
5. Selecciona No tengo los datos de inicio de sesión de esta persona, se abrirá un formulario para crear una cuenta de Microsoft, pero queremos una cuenta local.
6. Pulsa en Agregar un usuario sin cuenta Microsoft y pulsa Siguiente.
7. Llena el formulario y pulsa Siguiente.

2.1.2 Cambiar el tipo de cuenta

1. Pulsa Inicio y elige Configuración. Se abre la ventana de Configuración.
2. Pulsa el botón Cuentas.
3. Elige Familia y otros usuarios.
4. Haz clic en el nombre de usuario que creaste en el ejercicio anterior.

5. Pulsa en Cambiar el tipo de cuenta.
6. Despliega la lista de Tipo de cuenta y selecciona Administrador.
7. Pulsa Aceptar.

2.1.3 Eliminar cuenta

1. Pulsa Inicio y elige Configuración. Se abre la ventana de Configuración.
2. Pulsa el botón Cuentas.
3. Elige Familia y otros usuarios.
4. Haz clic en el nombre de usuario que creaste en el ejercicio anterior.
5. Pulsa en Quitar, se abre un cuadro de diálogo.
6. Pulsa en el botón Eliminar cuenta y datos.

2.2 Administración de usuarios y grupos con herramientas avanzadas

Vamos a utilizar las herramientas avanzadas de gestión de usuarios y grupos locales en un sistema Windows 10 Profesional.

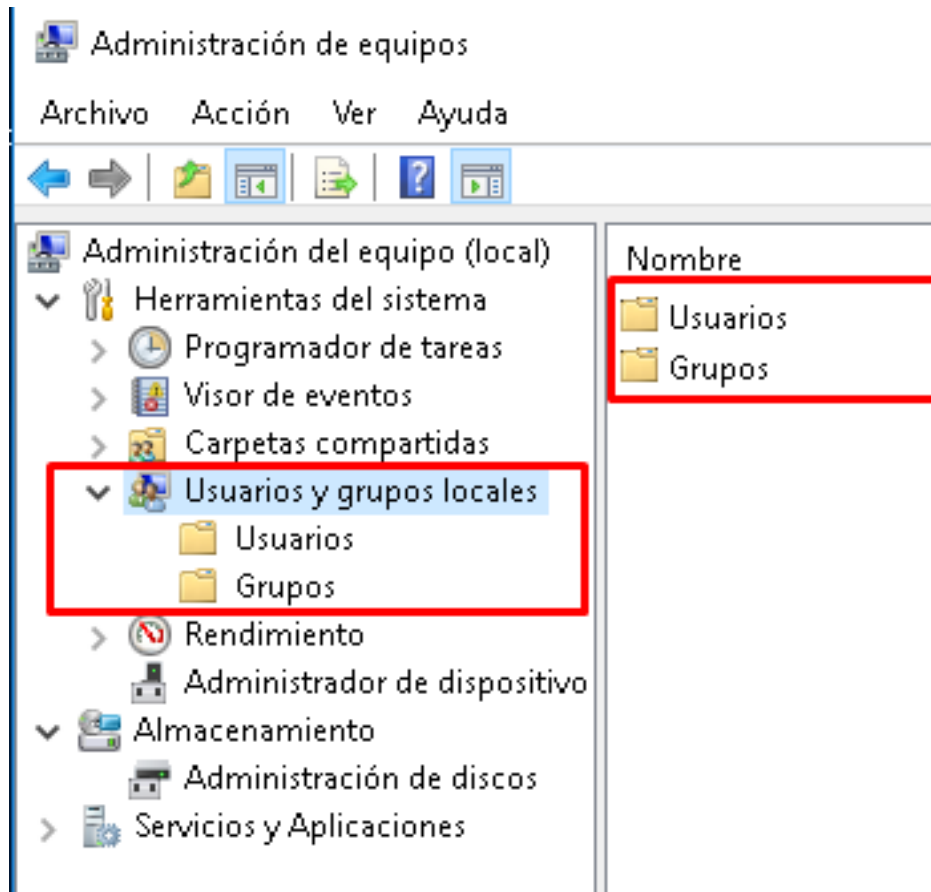
Para ello, vamos a realizar un pequeño ejemplo práctico. En nuestro sistema contamos con los siguientes usuarios y departamentos:

Usuario	Departamento
Marina Pérez Gómez	Contabilidad, Almacen
Alfredo Flores Fuentes	Contabilidad
Ramón Andrade Marín	Almacen

Todos estos usuarios utilizan un mismo sistema con Windows 10 Profesional. Queremos configurarlo de forma que, más adelante, cada tipo de usuario tenga distintos permisos o privilegios en función del departamento al que pertenezca.

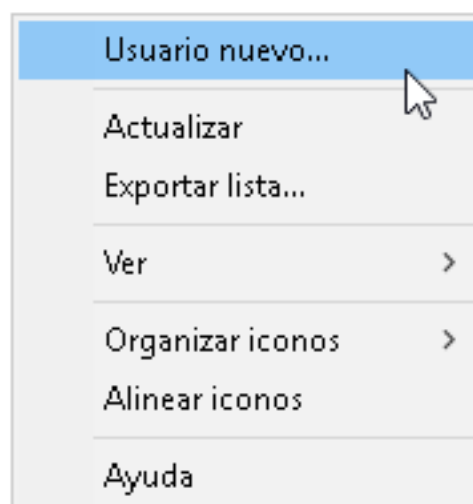
Es decir, se crearán una serie de directorios dentro del sistema de ficheros del equipo al que solo podrán acceder ciertos usuarios en función del departamento al que pertenezcan. Por ejemplo, un directorio Contabilidad al que solo podrán acceder los usuarios del departamento de Contabilidad.

Para crear usuarios y grupos locales utilizamos la herramienta de administración de equipos, a la que se puede acceder desde el menú contextual que aparece al pulsar con el botón derecho en el menú de Inicio.



Si nos fijamos en el panel izquierdo de la herramienta tenemos la opción de Usuarios y grupos locales. Vemos a trabajar con usuarios y grupos locales de este sistema. Es decir, todos los usuarios y grupos que creamos en este sistema serán propios de este sistema de forma local. Si tuviéramos dos sistemas con Windows 10 y quisiéramos contar en los dos con los mismos usuarios y grupos, tendríamos que crearlos en ambos sistemas.

Primero vamos a crear los tres usuarios.



Primero creamos el usuario de Marina Pérez Gómez. Como nombre de usuario utilizaremos el nombre de pila del usuario y la primera letra de cada uno de sus dos apellidos. De esta forma, el nombre de usuario será marinapg.

Además marcamos la opción de» El usuario debe cambiar la contraseña en el siguiente inicio de sesión» Esto lo hacemos así para que marinapg tenga que poner una contraseña propia que nosotros, los administradores, no conozcamos.

Usuario nuevo

Nombre de usuario: marinapg

Nombre completo:

Descripción:

Contraseña: ●●●●●●●●

Confirmar contraseña: ●●●●●●●●

☒ El usuario debe cambiar la contraseña en el siguiente inicio de sesión











☐ El usuario no puede cambiar la contraseña

☐ La contraseña nunca expira

☐ La cuenta está deshabilitada

Ayuda Crear Cerrar

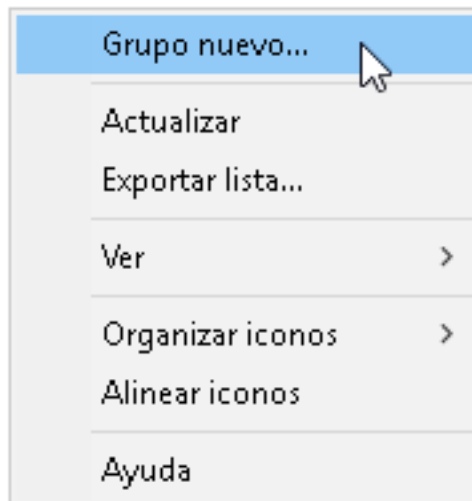
Ahora vamos a crear los grupos de usuarios locales propios de este sistema. En este caso, los grupos de usuario coincidirán con las agrupaciones presentes en la empresa real. Esto es muy habitual, puesto que si ya existen agrupaciones en la empresa es probable que los permisos tengan mucho que ver con estas agrupaciones.

Nombre	Nombre completo	Descripción
 Administrador		Cuenta integrada para la administ...
 alfredoff	alfredoff	
 alumno		
 DefaultAcco...		Cuenta de usuario administrada p...
 Invitado		Cuenta integrada para el acceso c...
 javier	javier	
 maria	Maria Antúnez	Jefa de Servicio de Contabilidad
 marinapg	marinapg	
 pepe	pepe	
 ramonam	ramonam	

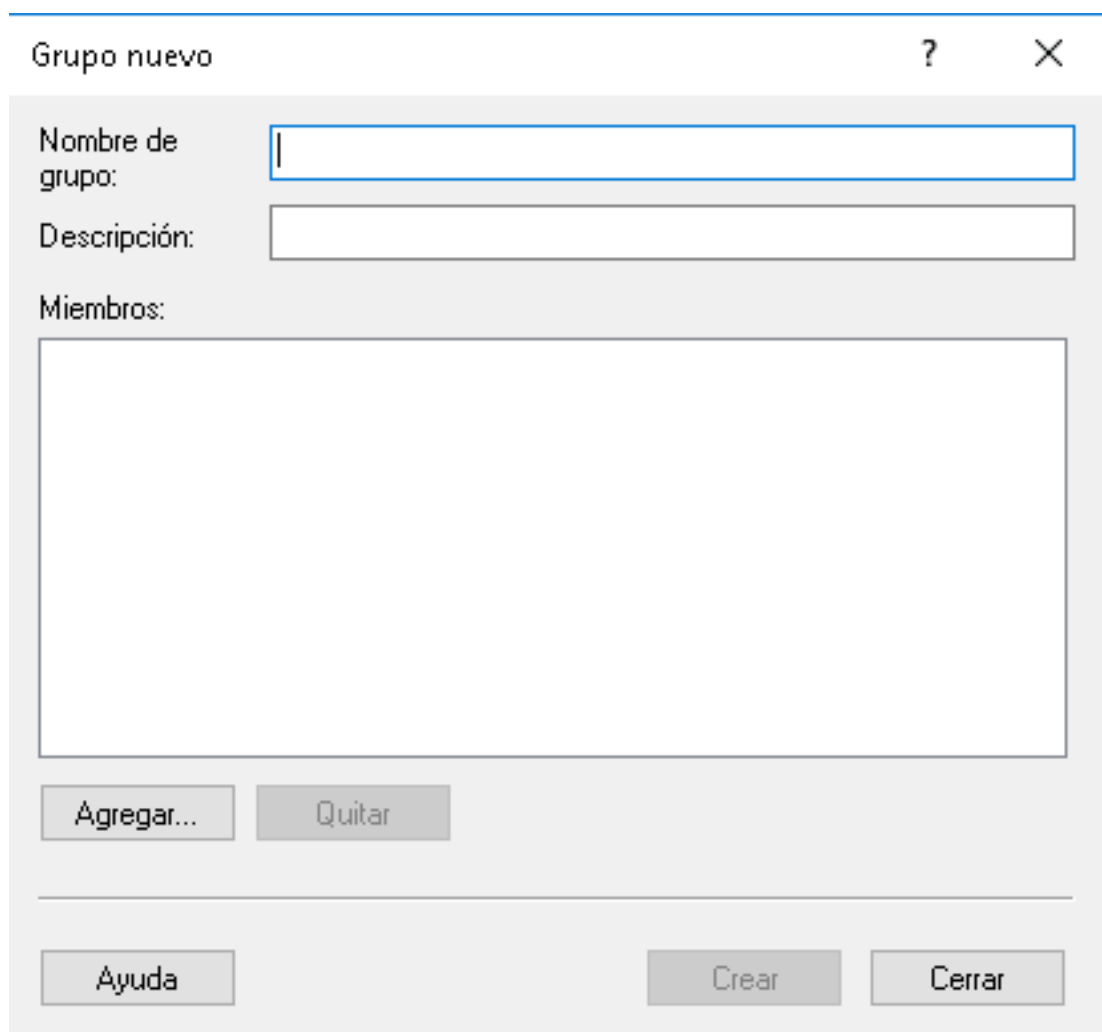
Seleccionamos la opción Grupos en el árbol de Usuarios y grupos locales.

Administración del equipo (local)		Nombre	Des
<ul style="list-style-type: none"> <ul style="list-style-type: none"> Herramientas del sistema Programador de tareas Visor de eventos Carpetas compartidas <ul style="list-style-type: none"> Usuarios y grupos locales <ul style="list-style-type: none"> Usuarios Grupos Rendimiento Administrador de dispositivo Almacenamiento <ul style="list-style-type: none"> Administración de discos Servicios y Aplicaciones 		<ul style="list-style-type: none"> Administradores Administradores de Hyper-V Duplicadores IIS_IUSRS Invitados Lectores del registro de eventos Operadores criptográficos Operadores de asistencia de control de acceso Operadores de configuración de red Operadores de copia de seguridad System Managed Accounts Group Usuarios Usuarios avanzados Usuarios COM distribuidos Usuarios de administración remota Usuarios de escritorio remoto Usuarios del monitor de sistema Usuarios del registro de rendimiento 	<ul style="list-style-type: none"> Los Los Pue Grup De f Los Los Los Los Los Los Los Los Los Los Los Los Los

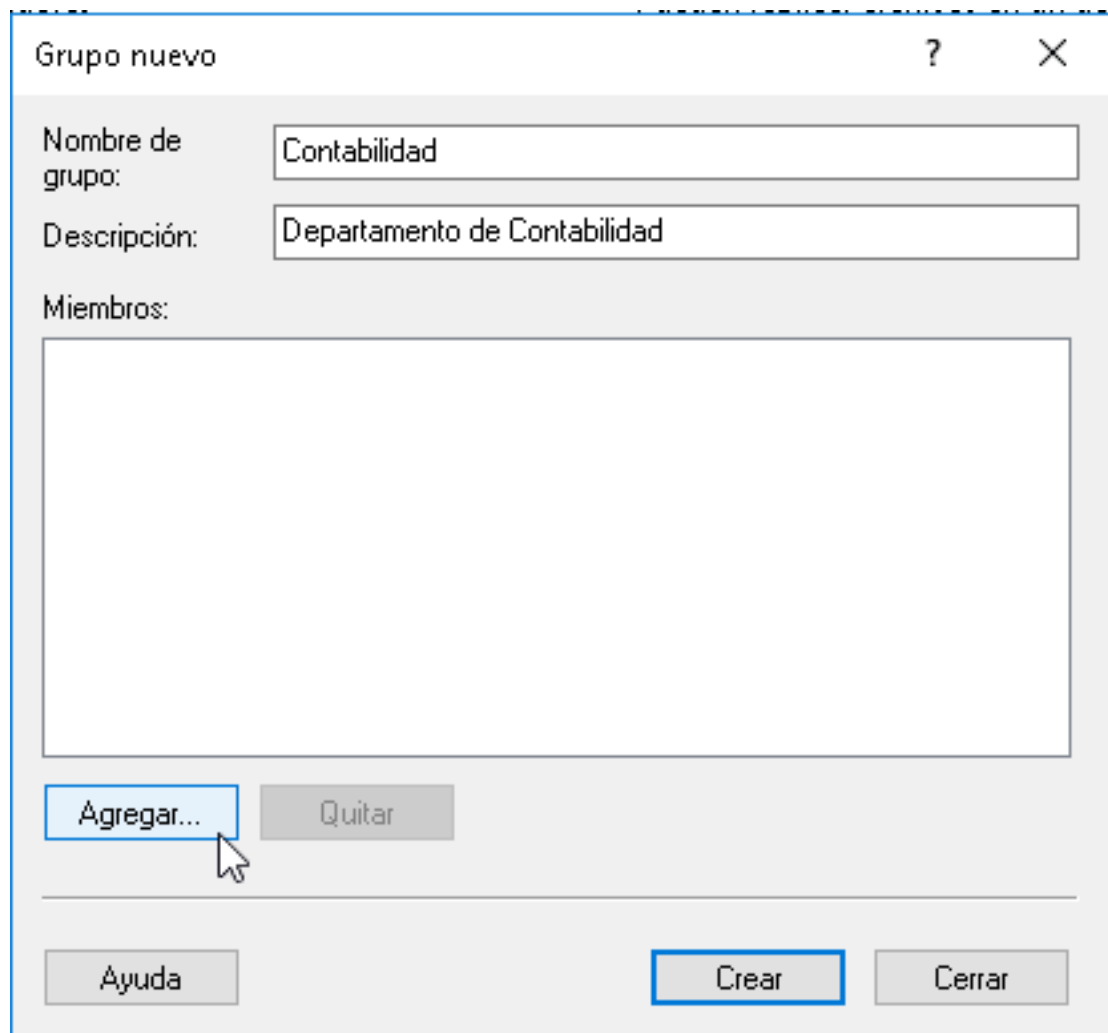
En los sistemas Windows hay una serie de grupos predeterminados que se crean automáticamente en la instalación del sistema. Estos grupos son utilizados por el propio sistema operativo.



En el cuadro de dialogo para crear grupos nos pide el nombre del grupo, que será su identificador y por tanto debe ser único; una descripción que es opcional y los miembros del grupo. Podemos aprovechar, en el momento de crear el grupo, para añadir los miembros que tenga dicho grupo.

A dialog box titled 'Grupo nuevo' with a question mark and close button in the title bar. The dialog has three main sections: 'Nombre de grupo:' with a text input field, 'Descripción:' with a text input field, and 'Miembros:' with a large empty rectangular area. Below the 'Miembros:' section are two buttons: 'Agregar...' and 'Quitar'. At the bottom of the dialog are three buttons: 'Ayuda', 'Crear', and 'Cerrar'.

Comencemos por el grupo Contabilidad.



Grupo nuevo

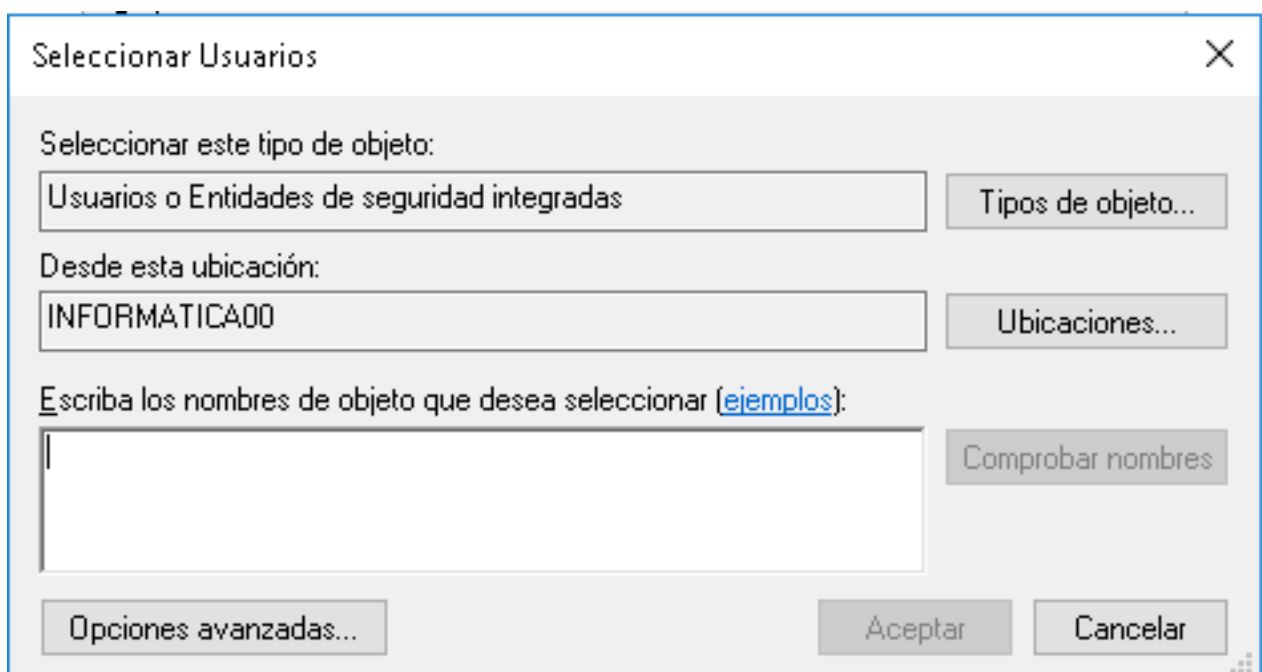
Nombre de grupo: Contabilidad

Descripción: Departamento de Contabilidad

Miembros:

Agregar... Quitar

Ayuda Crear Cerrar



Seleccionar Usuarios

Seleccionar este tipo de objeto: Usuarios o Entidades de seguridad integradas Tipos de objeto...

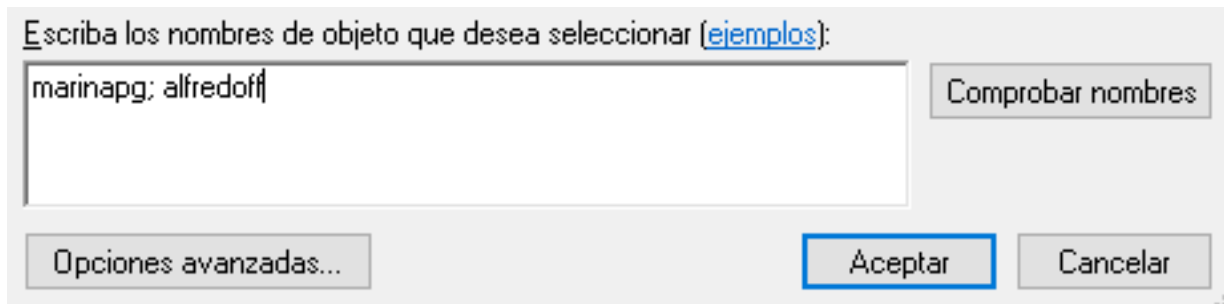
Desde esta ubicación: INFORMATICA00 Ubicaciones...

Escriba los nombres de objeto que desea seleccionar ([ejemplos](#)):

Comprobar nombres

Opciones avanzadas... Aceptar Cancelar

En este cuadro de diálogo seleccionamos los usuarios que queremos agregar al grupo de Contabilidad. Si conocemos los nombres de usuario y son pocos, como en este caso, podemos escribirlos directamente en una lista separados por «;».



Escriba los nombres de objeto que desea seleccionar ([ejemplos](#)):

marinapg; alfredoff

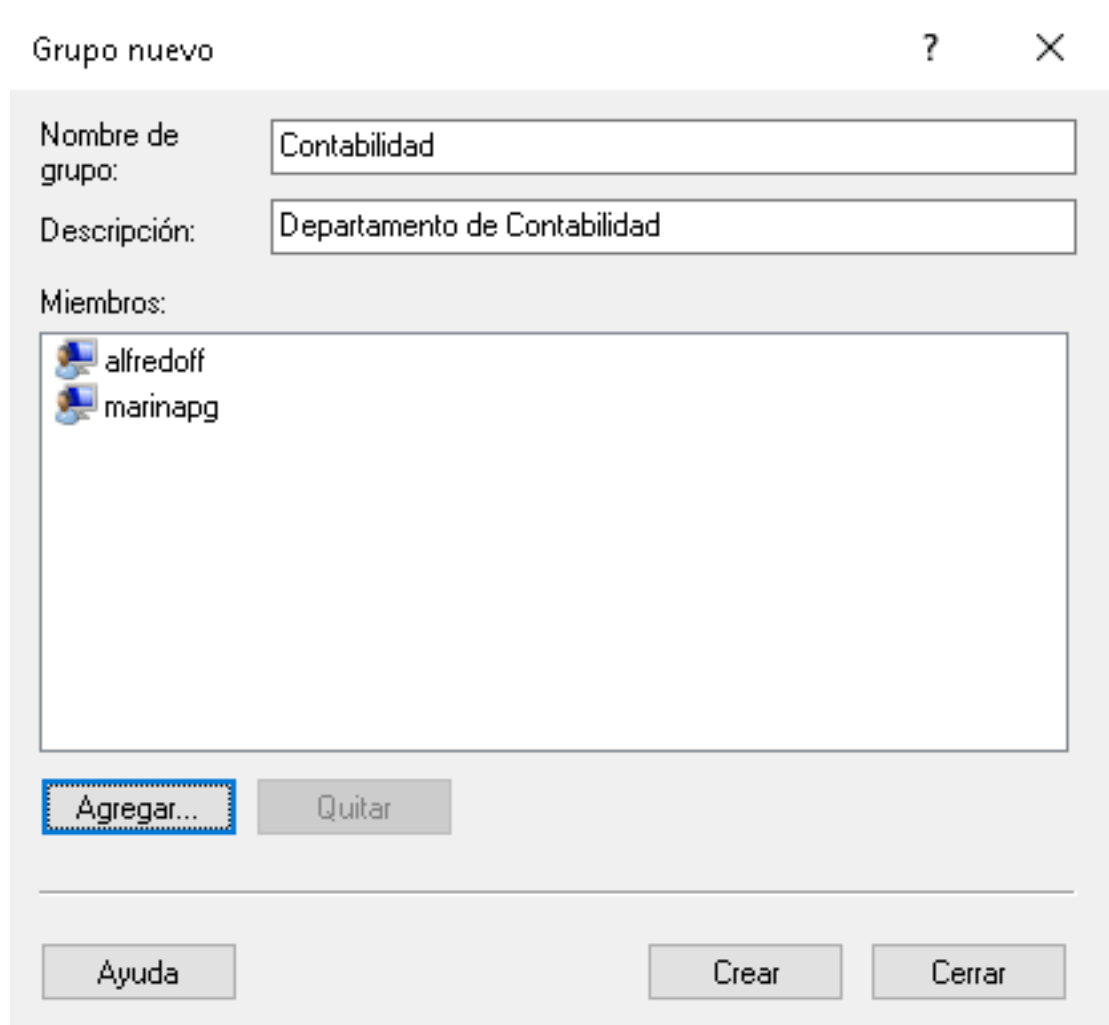
Comprobar nombres

Opciones avanzadas...

Aceptar

Cancelar

Al pulsar aceptar – Podemos pulsar Enter porque Aceptar esta preseleccionado – aparecerán los usuarios agregados.



Grupo nuevo

Nombre de grupo: Contabilidad

Descripción: Departamento de Contabilidad

Miembros:

- alfredoff
- marinapg

Agregar...

Quitar

Ayuda

Crear

Cerrar

Para crear el grupo tan solo tendremos que hacer clic en el botón Crear.

Si tenemos una lista de usuarios grande o no recordamos bien el nombre de los usuarios, podemos utilizar la opción de opciones avanzadas para seleccionar los usuarios a agregar en un grupo.

Seleccionar Usuarios ✕

Seleccionar este tipo de objeto:

Usuarios o Entidades de seguridad integradas Tipos de objeto...

Desde esta ubicación:

INFORMATICA00 Ubicaciones...

Escriba los nombres de objeto que desea seleccionar ([ejemplos](#)):

Comprobar nombres

Opciones avanzadas... Aceptar Cancelar

Seleccionar Usuarios ✕

Seleccionar este tipo de objeto:

Usuarios o Entidades de seguridad integradas Tipos de objeto...

Desde esta ubicación:

INFORMATICA00 Ubicaciones...

Consultas comunes

Nombre: Empieza con

Descripción: Empieza con

☐ Cuentas deshabilitadas


☐ Contraseñas que nunca expiran

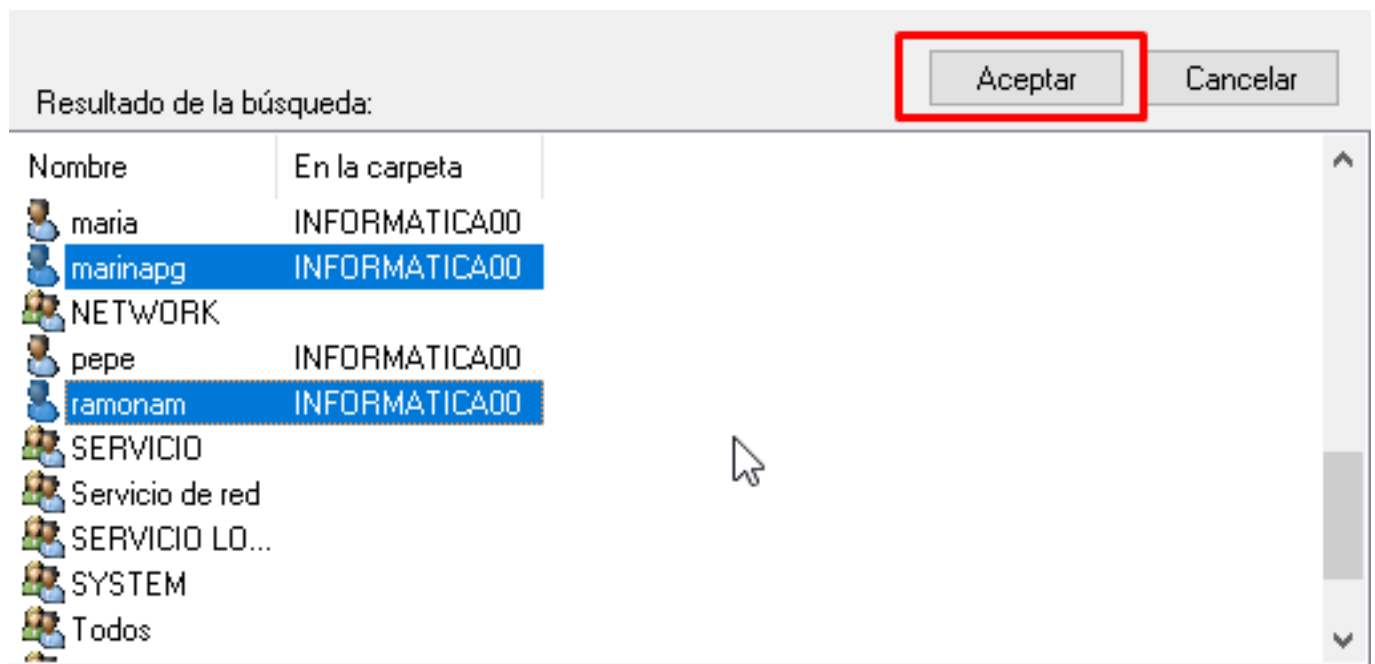
Número de días transcurridos desde el último inicio de sesión:

Columnas...

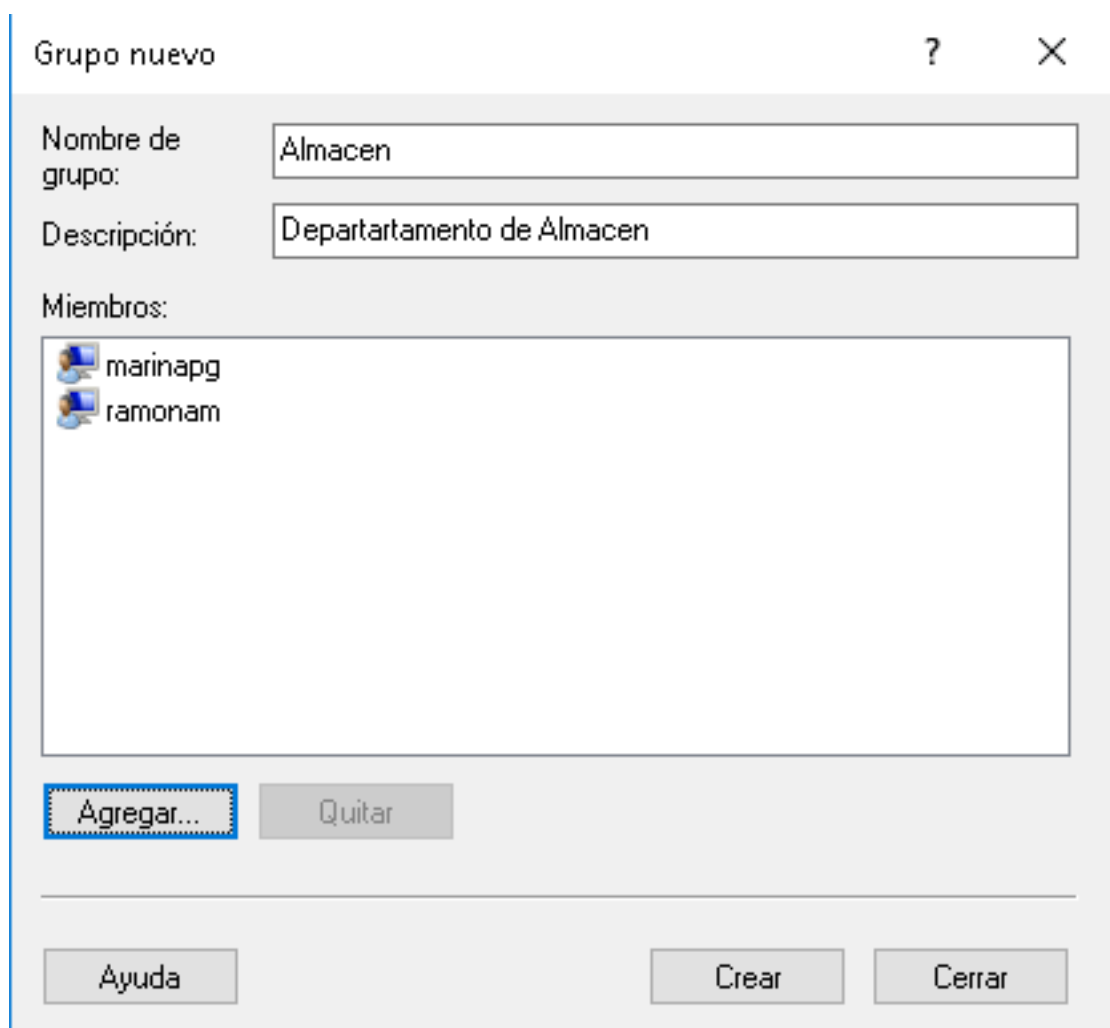
Buscar ahora

Detener

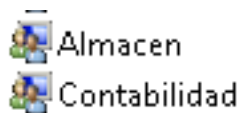




Podemos seleccionar varios usuarios utilizando la tecla Control para selecciones disjuntas, como es en este caso.

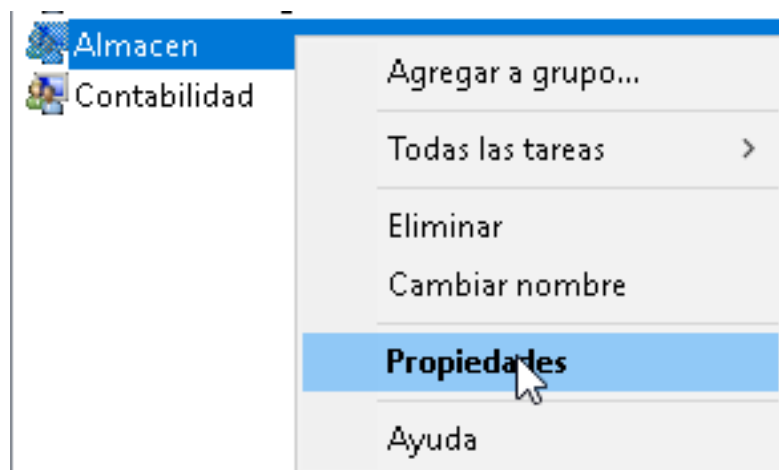


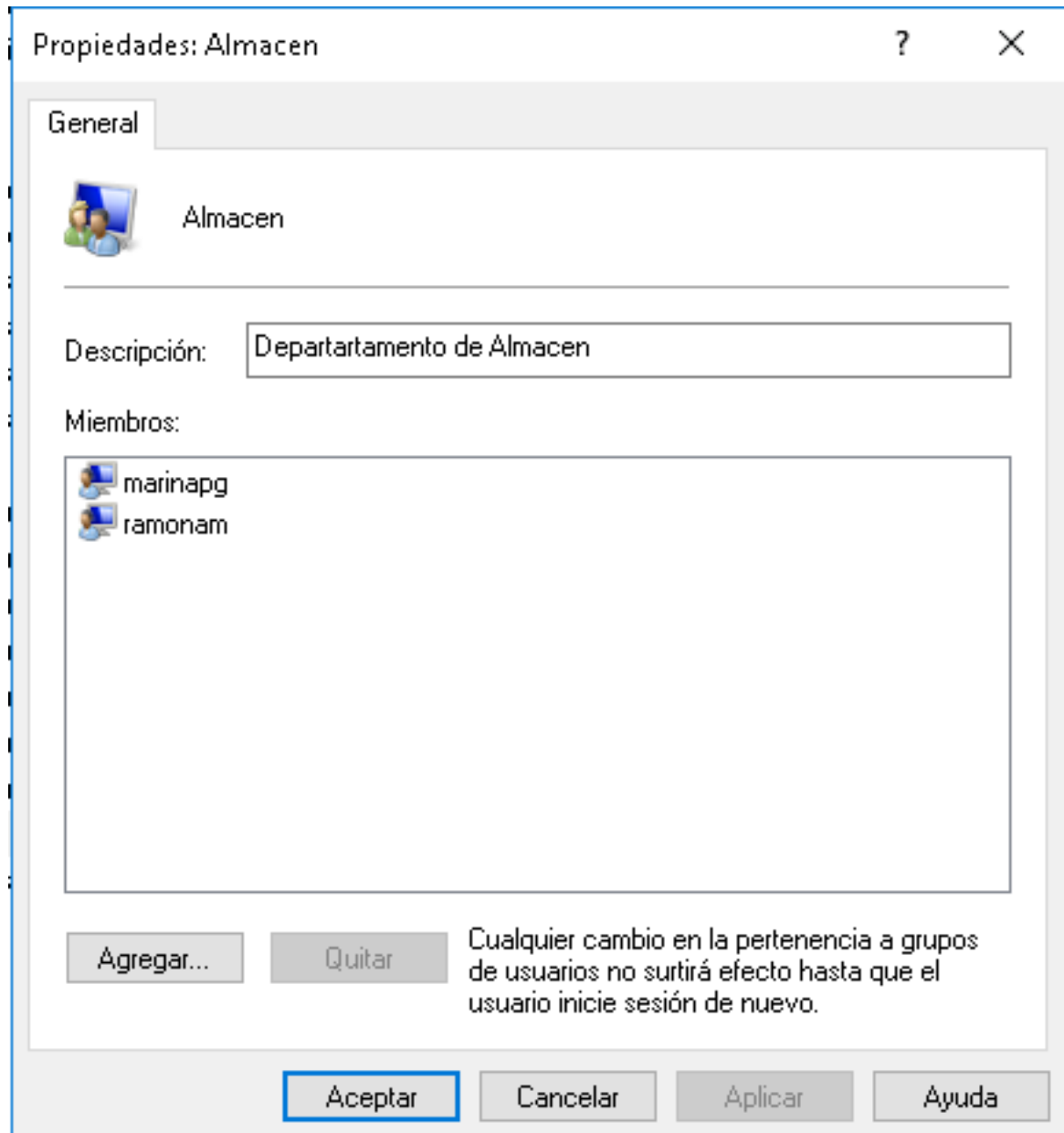
De esta forma ya hemos creado los grupos.



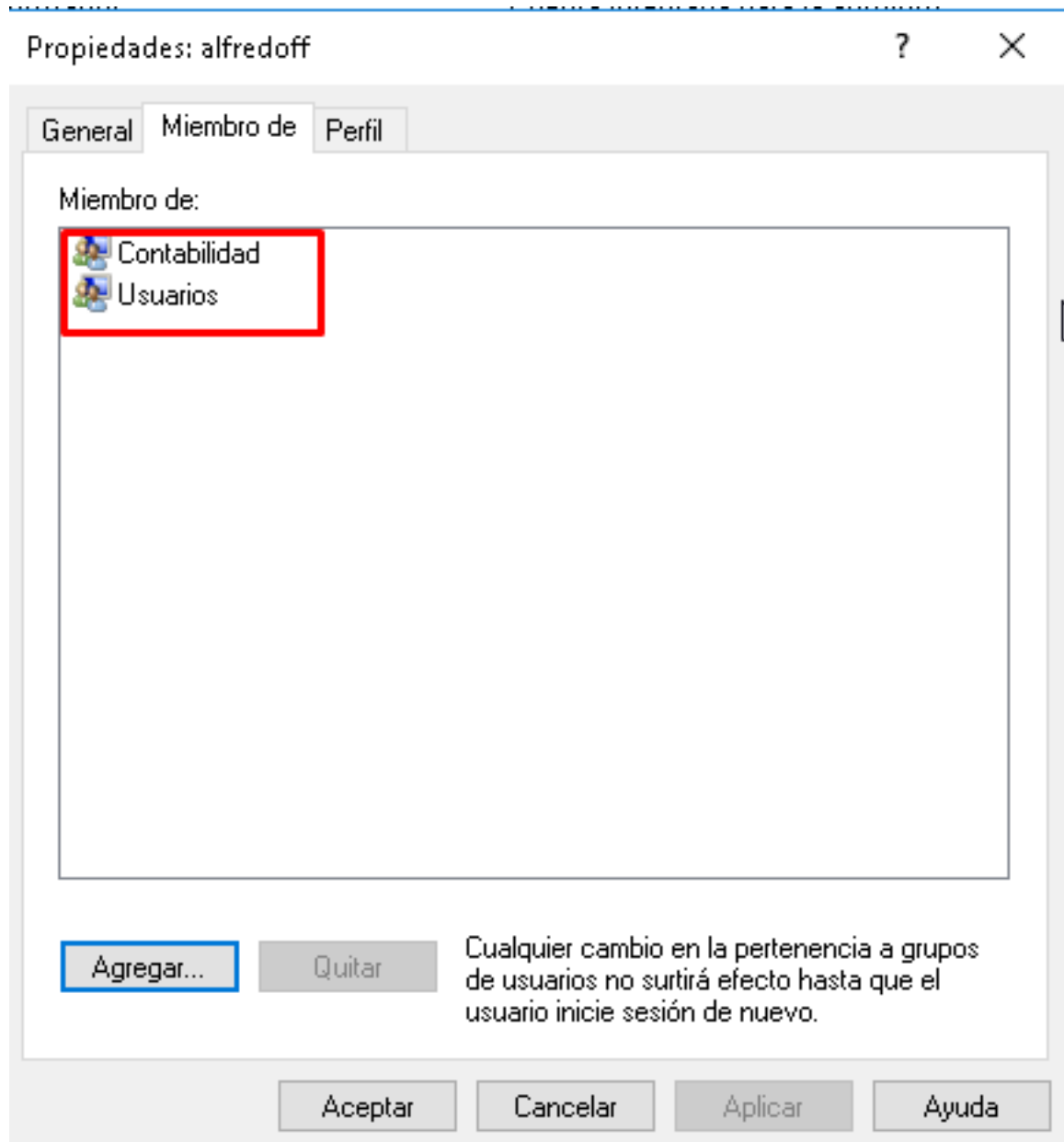
Departamento de Almacen
Departamento de Contabilidad

Vamos a ver que miembros tiene cada grupo.



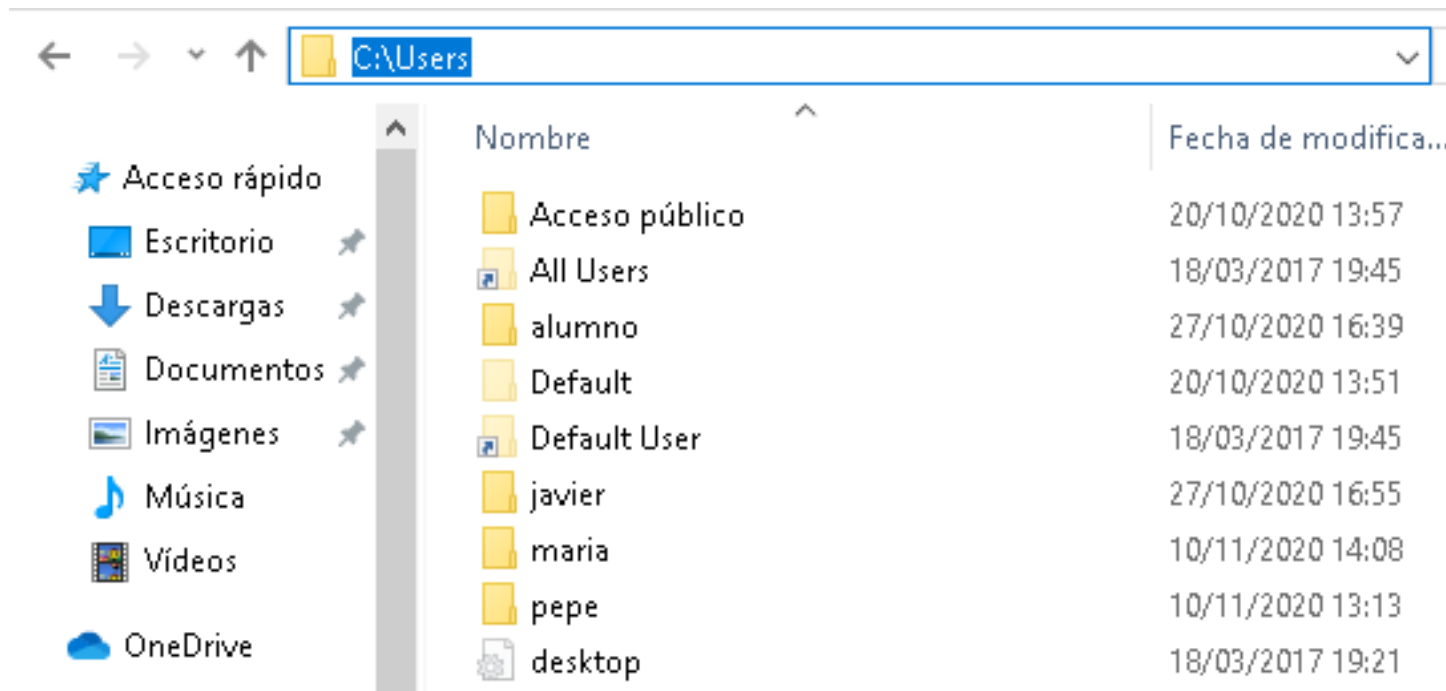


También podemos comprobar a qué grupo pertenece un usuario concreto e incluso agregarlo a uno o varios grupos desde propiedades de usuario. Vamos a comprobar las propiedades del usuario alfredoff



Si nos fijamos, este usuario es miembro de Contabilidad y del grupo Usuarios. Sin embargo, nosotros no hemos añadido este usuario al grupo Usuarios. El sistema operativo es el que ha añadido este usuario de forma automática al grupo Usuarios. El grupo Usuarios es el grupo al que pertenecen, por defecto, los usuarios del sistema.

Si nos fijamos, todavía no se han creado perfiles de usuario para los usuarios recién creados.



Esto es así porque todavía no han iniciado sesión en el sistema. Windows crea el perfil de un usuario nuevo solo cuando inicia sesión en el sistema por primera vez.

3 Gestión de usuarios y grupos desde el CMD

3.1 Gestión de usuarios

3.1.1 Administrar usuarios en Windows usando Net User

Como hemos visto, cuando administramos los usuarios de Windows, ya sean administradores o usuarios normales con menos permisos, solemos crearlos, modificarlos y eliminarlos directamente desde «**Panel de control / Usuarios**», aunque también lo podemos hacer desde «**Configuración / Cuentas**». Sin embargo, disponemos de un comando realmente interesante que es el de «**Net User**» que nos permitirá administrar los usuarios del sistema operativo Windows a través de consola con el símbolo del sistema.

3.1.2 ¿Qué es Net User y para qué se utiliza?

Net User es una aplicación de consola que permite gestionar los usuarios existentes en un sistema operativo de una forma rápida y sencilla, todo ello a través de comandos por consola. Se trata de una herramienta que está destinada a cumplir con las necesidades de los administradores de sistemas, los cuales suelen acceder al sistema operativo a través de SSH y ejecutar diferentes comandos en la consola. La principal finalidad de esta herramienta no es otra que activar las cuentas de administrador que permanecen inactivas en el sistema, desactivarlas y gestionar los diferentes usuarios que tenemos dados de alta en el equipo.

El único requisito para poder ejecutar este comando es lanzar la consola (CMD) o símbolo del sistema con permisos de administrador, por tanto, tendremos que pinchar en el icono de «**Inicio / Windows**», ponemos «**CMD**» y pinchamos click derecho sobre el símbolo del sistema y elegimos «**Ejecutar como administrador**», de lo contrario no podremos ejecutar los diferentes comandos que tenemos disponibles.

Una vez que hayamos abierto una sesión de símbolo del sistema o consola, vamos a ver todas las opciones que tenemos disponibles y os pondremos algunos ejemplos de utilización de este programa.

3.1.3 Funcionamiento de la herramienta Net User

El funcionamiento de este comando es realmente sencillo, sobre todo para visualizar información, si queremos modificar información entonces tendremos que poner comandos algo más complicados. Si ejecutamos el comando «**net user**» sin ningún argumento, podremos ver el nombre del equipo y también qué usuarios tenemos actualmente dados de alta en nuestro sistema operativo.

Existe un listado de argumentos que podemos utilizar para ver más información sobre una o varias cuentas de usuario, además, podremos fijar una contraseña, activar la cuenta de usuario, desactivarla, e incluso podremos eliminar el usuario por completo.

- **NET USER nombreUsuario:** esta instrucción devuelve toda la información asociada al nombre de usuario que se ha proporcionado. Es decir, se podrá visualizar la configuración de todos los parámetros asociados a la cuenta.
- **NET USER nombreUsuario contraseña:** asocia una la contraseña indicada al nombre de usuario facilitado, esto permite cambiar la clave de acceso con ese usuario en concreto.
- **NET USER nombreUsuario /active:yes|no:** permite activar o desactivar la cuenta del usuario. Es decir, permitir que esa cuenta haga login en el sistema o no lo haga aunque introduzca correctamente la contraseña.
- **NET USER nombreUsuario contraseña /add:** crea un nuevo usuario en el sistema, se incorpora el nombre de usuario nuevo y la contraseña que queremos que ese usuario tenga en concreto. Es muy parecida a la citada con anterioridad, con la única diferencia de que la anterior solo sirve para asociar una contraseña a un usuario ya existente en el sistema operativo, este comando sirve para crear un nuevo usuario.
- **NET USER nombreUsuario /delete:** elimina el usuario y todo lo relacionado con él en lo que se refiere a configuración. Los cambios no son reversibles.

Si escribimos el comando «**net user /help**» podremos lanzar la ayuda de net user con todos y cada uno de los comandos disponibles, además, también veremos una leyenda de lo que hace cada uno de los comandos que tenemos disponibles con este programa, por tanto, no tendremos demasiados problemas para utilizar este programa, además, tenemos diferentes ejemplos que podemos ejecutar para ir probando lo que hace cada uno de ellos.

Una función muy interesante es que nos permite también configurar estos usuarios directamente en un controlador de dominio del dominio actual, siempre que seamos administradores del propio controlador de dominio. Otras opciones interesantes que nos permite net user son las siguientes:

- **/COMMENT:** permite añadir un comentario en la cuenta de usuario que vamos a crear.
- **/COUNTRYCODE:** permite añadir el código del país o región del sistema operativo para usar los archivos de idioma.
- **/EXPIRES:** permite configurar cuándo expira una determinada cuenta que vamos a añadir o modificar, si no indicamos nada no expirará nunca.
- **/FULLNAME:** esta opción nos permite definir el nombre completo del usuario, y no solamente el nombre de usuario de inicio de sesión.
- **/HOMEDIR:** permite configurar la ruta del directorio principal del usuario.
- **/PASSWORDCHG:** permite configurar si se permite a los usuarios cambiar su propia contraseña de acceso.
- **/PASSWORDREQ:** permite configurar que los usuarios siempre tengan una contraseña de acceso.
- **/LOGONPASSWORDCHG:** fuerza al cambio de contraseña del usuario en cuestión en el siguiente inicio de sesión.
- **/PROFILEPATH:** establece la ruta para el perfil de inicio de sesión del usuario.
- **/SCRIPTPATH:** ubicación del script de inicio de sesión del usuario en cuestión.
- **/TIMES:** horas en las que un determinado usuario puede iniciar sesión.
- **/USERCOMMENT:** permite agregar o cambiar el comentario de un usuario para la cuenta.
- **/WORKSTATIONS:** muestra hasta 8 equipos en la red desde los que podrá iniciar sesión un usuario.

Este comando es realmente útil para gestionar los usuarios del sistema operativo. Podremos crear, modificarlos, eliminarlos, y también configurar sus contraseñas, políticas de contraseñas, si la cuenta puede expirar, el directorio raíz de un determinado usuario y muchos otros parámetros para gestionar correctamente todos los usuarios del sistema operativo haciendo uso de net user para Windows.

3.2 Gestión de grupos

3.2.1 Gestión de grupos locales para Windows desde la línea de comandos.

Los sistemas operativos modernos como Linux, Windows utiliza grupos para administrar los derechos de los usuarios de manera más flexible. Los usuarios generalmente asignados a los grupos para usar los privilegios relacionados con los grupos como Administración Local, Usuario de Escritorio Remoto, etc.

3.2.2 Ayuda

La información de ayuda y sintaxis de los grupos locales se puede imprimir con el siguiente comando.

```
$ net localgroup /?
```

3.2.3 Lista de grupos

Los grupos locales pueden ser listados con el comando net localgroup. Esto listará los grupos locales, no los grupos de un dominio.

```
$ net localgroup
```

Como podemos ver, los grupos se enumeran línea por línea. Estos grupos son los que vienen por defecto con el sistema operativo Windows y los que se han creado posteriormente.

3.2.4 Crear un grupo

Los grupos locales de Windows pueden ser creados con la opción /ADD fácilmente. En este ejemplo, crearemos un grupo local llamado Developers.

```
$ net localgroup Developers /ADD
```

3.2.5 Borrado de grupos

Los grupos locales pueden ser eliminados si no los necesitamos. Usaremos la opción /DELETE para eliminar el grupo.

```
$ net localgroup Developers /DELETE
```

3.2.6 Lista de usuarios de grupos

Una de las opciones más utilizadas es la de incluir a determinados grupos de usuarios. Necesitamos proveer el nombre del grupo local para sus usuarios. En este ejemplo listaremos el grupo local llamado Administradores.

```
$ net localgroup "Administradores"
```

Como podemos ver, el alias de los grupos locales y el comentario sobre el grupo también se muestra. No hay que confundir el usuario Administrador con el grupo local Administradores; son entidades diferentes.

3.2.7 Añadir usuario a un grupo

Podemos añadir un usuario existente al grupo local proporcionando el nombre del usuario y la opción /ADD. En este ejemplo añadiremos el usuario ismail al grupo de Developers. De este modo, el usuario ismail utilizará todos los recursos y privilegios proporcionados por el grupo de Developers.

```
$ net localgroup Developers ismail /ADD
```

3.2.8 Añadir comentario al grupo

En un entorno muy utilizado, manejaremos muchos grupos creados por los administradores. Necesitamos algunas pistas sobre estos grupos locales para recordar. O podemos poner algunas notas como cuando vamos a eliminar estos grupos. Toda esta información puede ser almacenada en los comentarios sobre el grupo. Vamos a poner una nota sobre el grupo de Developers.

```
$ net localgroup Developers /COMMENT: "Only 10.1.0.0/24 network"
```

El comentario puede ser visto con el GUI o mientras se listan los miembros del grupo.

4 Administrar usuarios y grupos desde Powershell

4.1 Resumen de comandos

Para que un usuario pueda iniciar una sesión en Windows o acceder a un recurso (carpeta compartida, archivo, impresora compartida, etc.), es necesario crear previamente una cuenta de usuario. Esta puede ser local, es decir, conocida únicamente en el puesto de trabajo en el que se creará, o bien pertenecer a un directorio Active Directory, donde tendrá la posibilidad de conectarse a cualquier equipo del dominio, siempre y cuando posea los permisos correspondientes.

En esta sección veremos la gestión de las cuentas de usuario y los grupos locales. He aquí un breve resumen de los cmdlets y su rol:

- Add-LocalGroupMember: agrega un miembro en un grupo local.
 - Disable-LocalUser: permite desactivar una cuenta de usuario.
 - Enable-LocalUser: permite activar una cuenta de usuario.
 - Get-LocalGroup: muestra el conjunto de grupos de locales presentes en el puesto de trabajo.
 - Get-LocalGroupMember: recupera los miembros presentes en un grupo local.
 - Get-LocalUser: muestra el conjunto de cuentas de usuario locales presentes en el puesto de trabajo.
 - New-LocalGroup: crea un nuevo grupo local.
 - New-LocalUser: crea una nueva cuenta de usuario local.
 - Remove-LocalGroup: elimina un grupo local.
 - Remove-LocalGroupMember: elimina un miembro de un grupo local.
 - Remove-LocalUser: elimina una cuenta de usuario local.
 - Rename-LocalGroup: renombra un grupo local.
 - Rename-LocalUser: renombra una cuenta de usuario local (y, por consiguiente, su identificador de login).
 - Set-LocalGroup: permite modificar las propiedades de un grupo local.
 - Set-LocalUser: permite modificar las propiedades de una cuenta de usuario local.
-

4.2 Gestión de usuarios

En esta sección veremos la administración de usuarios locales: cómo crear una nueva cuenta de usuario y también cómo eliminar, desactivar o incluso modificar una cuenta existente.

4.2.1 Obtener la lista de cuentas de usuario

Para obtener la lista de cuentas de usuario locales creadas en el puesto de trabajo, se utiliza el cmdlet `Get-LocalUser`. He aquí los parámetros:

Parámetro	Descripción
<code>-Name<String[]></code>	Designa el nombre de la cuenta o las cuentas de usuario que se devolverán.
<code>-SID<SecurityIdentifier[]></code>	Designa el identificador de seguridad (de tipo SID - Security Identifiers) de la cuenta o las cuentas de usuario que se devolverán.

Ejemplo: mostrar todas las cuentas de usuario locales

```
PS C:\WINDOWS\system32> Get-LocalUser~
~
Name~-----Enabled~Description~
-----
Administrator~False~~Built-in account for administering the computer/domain~
Alejandro~True~
Aurore Perrin~True~
DefaultAccount~False~~A user account managed by the system.~
Guest~False~~Built-in account for guest access to the computer/ ←
domain~
WDAGUtilityAccount~False~~A user account managed and used by the system for ←
Windows Defender Application Guard scen...
```

Ejemplo: recuperar toda la información correspondiente a una cuenta de usuario

```
PS C:\WINDOWS\system32> Get-LocalUser -Name Administrator | Format-List *~
~
AccountExpires~~~~~:~
Description~~~~~: Built-in account for administering the computer/domain~
Enabled~~~~~: False~
FullName~~~~~:~
PasswordChangeableDate~:~
PasswordExpires~~~~~:~
UserMayChangePassword~~: True~
PasswordRequired~~~~~: True~
PasswordLastSet~~~~~:~
LastLogon~~~~~:~
Name~~~~~: Administrator~
SID~~~~~: S-1-5-21-1090146567-1987342786-2075620693-500~
PrincipalSource~~~~~: Local~
ObjectClass~~~~~: UsuarioPS
```

De este modo, puede recuperar información como las fechas de expiración de la cuenta (`AccountExpires`) o de la contraseña (`PasswordExpires`), y también el identificador de seguridad de la cuenta (`SID`). También se obtiene información adicional, como la fecha de última conexión con esta cuenta (`LastLogon`).

4.2.2 Crear una cuenta de usuario local

El cmdlet `New-LocalUser` permite crear una nueva cuenta de usuario. Esta, una vez creada, no formará parte de ningún grupo, y por consiguiente no poseerá ningún permiso en el puesto de trabajo.

Los distintos parámetros que ofrece este cmdlet permiten configurar la cuenta de usuario tras su creación:

Parámetro	Descripción
<code>-AccountExpires<DateTime></code>	Especifica la fecha y hora de expiración de la cuenta de usuario.
<code>-AccountNeverExpires</code>	Indica que la cuenta nunca expira.
<code>-Description<String></code>	Especifica una descripción para esta cuenta de usuario. La descripción no debe superar los 48 caracteres.
<code>-Disabled</code>	Indica que la cuenta de usuario creada estará desactivada.
<code>-FullName<String></code>	Especifica el nombre completo para esta cuenta de usuario.
<code>-Name<String></code>	Especifica el nombre para esta cuenta de usuario. Se trata del login.
<code>-NoPassword</code>	Indica que la cuenta de usuario no tiene contraseña.
<code>-Password<SecureString></code>	Especifica la contraseña asociada a la cuenta de usuario creada.
<code>-PasswordNeverExpires</code>	Indica que la contraseña de la cuenta de usuario nunca expira.
<code>-UserMayNotChangePassword</code>	Indica que el usuario no puede cambiar su contraseña.

Ejemplo: crear una cuenta de usuario

```
PS C:\Windows\system32> $pwd = ConvertTo-SecureString 'P@$$w0rd' ~~
-AsPlainText -Force~~
PS C:\Windows\system32> New-LocalUser -Name Julien.Musy -Password $pwd ~~
-PasswordNeverExpires
```

Se crea la cuenta de usuario local y estará visible desde la consola Usuarios y grupos locales.

Ejemplo: crear una cuenta de usuario con una fecha de expiración

```
PS C:\Windows\system32> $pwd = ConvertTo-SecureString 'P@$$w0rd' ~~
-AsPlainText -Force~~
PS C:\Windows\system32> $date = (Get-Date).AddDays(7)~~
PS C:\Windows\system32> New-LocalUser -Name Frederic -Password $pwd ~~
-PasswordNeverExpires -AccountExpires $date
```

La nueva cuenta de usuario creada tiene una validez de una semana. Pasado este tiempo, el usuario no podrá iniciar sesión en el puesto de trabajo.

4.2.3 Modificar una cuenta de usuario local

Para modificar una cuenta de usuario existente, disponemos del cmdlet `Set-LocalUser`. Los parámetros son similares a los presentados en la creación de una cuenta de usuario:

Parámetro	Descripción
<code>-AccountExpires<DateTime></code>	Especifica la fecha y hora de expiración de la cuenta de usuario.
<code>-AccountNeverExpires</code>	Indica que la cuenta nunca expira.
<code>-Description<String></code>	Especifica una descripción para esta cuenta de usuario. La descripción no debe superar los 48 caracteres.
<code>-FullName<String></code>	Especifica el nombre completo para esta cuenta de usuario.

Parámetro	Descripción
-Name<String>	Designa el nombre de la cuenta de usuario que se modificará.
-Password<SecureString>	Especifica la contraseña asociada a la cuenta de usuario creada.
-PasswordNeverExpires<Boolean>	Indica que la contraseña de la cuenta de usuario nunca expira (\$True) o bien si debe expirar (\$False).
-SID<SecurityIdentifier>	Designa el identificador de seguridad de la cuenta de usuario que se modificará.
-UserMayChangePassword<Boolean>	Indica si el usuario puede cambiar su contraseña (\$True) o no (\$False).

Ejemplo: modificar las propiedades de una cuenta de usuario

```
PS C:\Windows\system32> Set-LocalUser -Name Julien.Musy -FullName ~~
"Julien Musy" -Description "Workstation Administrator"
```

En este ejemplo, las propiedades Nombre completo (FullName) y Descripción (Description) de la cuenta de usuario Julien.Musy se modifican o agregan si no existe esta información.

Ejemplo: cambiar la contraseña de una cuenta de usuario

```
PS C:\Windows\system32> Set-LocalUser -Name Julien.Musy -Password ~~
(ConvertTo-SecureString -String 'm07d3p455' -AsPlainText -Force)
```

4.2.4 Renombrar una cuenta de usuario local

Rename-LocalUser# permite cambiar el nombre de la cuenta de usuario y, por consiguiente, su identificador de login. Este cmdlet se utiliza haciendo referencia a la cuenta de usuario que se renombrará mediante su nombre actual o SID, indicando a continuación el nuevo nombre con el parámetro -NewName.

Parámetro	Descripción
-Name<String>	Designa el nombre actual de la cuenta de usuario que se renombrará.
-NewName<String>	Especifica el nuevo nombre para la cuenta de usuario, y por consiguiente su nuevo identificador de login.
-SID<SecurityIdentifier>	Designa el identificador de seguridad de la cuenta de usuario que se renombrará.

Ejemplo: renombrar una cuenta de usuario

```
PS C:\Windows\system32> Rename-LocalUser -Name Julien.Musy ~~
-NewName ws.admin
```

La cuenta de usuario se renombra a ws.admin, y en lo sucesivo deberá utilizar este identificador para iniciar una sesión con esta cuenta. Sin embargo, observe que, si bien el nombre ha cambiado, la ruta de acceso a su perfil de usuario (si este ya se había creado) no lo ha hecho. He aquí las variables de entorno de la sesión de usuario tras renombrarla:

```
PS C:\Users\Julien.Musy> $env:USERNAME~~
ws.admin~~
PS C:\Users\Julien.Musy> $env:USERPROFILE~~
C:\Users\Julien.Musy
```

¡De ahí la importancia de utilizar correctamente las variables de entorno en los scripts!

4.2.5 Activar una cuenta de usuario local

En caso de tener cuentas de usuario desactivadas, por ejemplo tras la creación de una nueva cuenta con el parámetro `-Disabled` de `New-LocalUser#`, el cmdlet `Enable-LocalUser#` permite reactivarla. Las cuentas desactivadas poseen un indicador (flecha hacia abajo) en su icono. Recuerde que el cmdlet `Get-LocalUser#` también permite consultar rápidamente el conjunto de cuentas desactivadas:

```
PS C:\WINDOWS\system32> Get-LocalUser | ? {$_.Enabled -eq $false}~~
~~
Name~~~~~ Enabled Description~~
-----
Administrador~~ False~~ Cuenta de usuario de administración~~
DefaultAccount~ False~~ Cuenta de usuario administrada por el sistema.~~
Invitado~~~~~ False~~ Cuenta de usuario invitado~~
Ludovic.White~~ False
```

He aquí los parámetros de `Enable-LocalUser`:

Parámetro	Descripción
<code>-Name<String[]></code>	Designa el nombre de la cuenta o las cuentas de usuario que se activarán.
<code>-SID<SecurityIdentifier[]></code>	Designa el identificador de seguridad de la cuenta o las cuentas de usuario que se activarán.

Ejemplo: activar una cuenta de usuario

```
PS C:\Windows\system32> Enable-LocalUser -Name Ludovic.White
```

4.2.6 Desactivar una cuenta de usuario local

De manera inversa, puede desactivar en cualquier momento una cuenta de usuario. Esta operación se realiza mediante el cmdlet `Disable-LocalUser`. Los parámetros son idénticos:

Parámetro	Descripción
<code>-Name<String[]></code>	Designa el nombre de la cuenta o las cuentas de usuario que se desactivarán.
<code>-SID<SecurityIdentifier[]></code>	Designa el identificador de seguridad de la cuenta o las cuentas de usuario que se desactivarán.

Ejemplo: desactivar una cuenta de usuario

```
PS C:\Windows\system32> Disable-LocalUser -Name Ludovic.White
```

4.2.7 Eliminar una cuenta de usuario local

Para terminar, la última operación que es posible realizar sobre las cuentas de usuario locales es la eliminación. `Remove-LocalUser#` permite llevar a cabo esta tarea, indicando el nombre o el SID de la cuenta que se ha de eliminar.

Observe, sin embargo, que si bien la cuenta se ha eliminado, no ocurre así con el perfil de usuario y el conjunto de datos que contiene.

Parámetro	Descripción
<code>-Name<String[]></code>	Designa el nombre de la cuenta o las cuentas de usuario que se eliminarán.

Parámetro	Descripción
-SID<SecurityIdentifier[]>	Designa el identificador de seguridad de la cuenta o las cuentas de usuario que se eliminarán.

Ejemplo: eliminar una cuenta de usuario

```
PS C:\Windows\system32> Remove-LocalUser -Name Ludovic.White
```

Tenga precaución: no se pide confirmación durante la ejecución de este cmdlet. Puede agregar el parámetro -Confirm para solicitar una confirmación.

Del mismo modo, tenga en cuenta que, si recrea a continuación una nueva cuenta con el mismo nombre que la que acaba de eliminar, esta no recuperará el entorno del usuario de la cuenta anterior. En efecto, el SID será diferente y, por consiguiente, cuando el usuario se autentique con esta nueva cuenta, Windows lo identificará como una nueva persona y creará un nuevo entorno de usuario, de modo que la ruta de acceso al perfil del usuario será diferente (Windows agregará el nombre del equipo en el nombre de la carpeta):

```
PS C:\Users\Ludovic.White.W10> $env:USERNAME~~
Ludovic.White~~
PS C:\Users\Ludovic.White.W10> $env:USERPROFILE~~
C:\Users\Ludovic.White.W10
```

4.3 Gestión de grupos

Los grupos permiten agrupar un conjunto de objetos (cuentas de usuario, entidades de seguridad integradas, etc.), y facilitar así la gestión y la administración.

El mejor ejemplo es la agrupación de cuentas de usuario en un mismo grupo, que representa una actividad o entidad. Será más sencillo otorgar o retirar permisos sobre un recurso al conjunto de usuarios presentes en estos grupos.

Por último, debe saber que los grupos predefinidos durante una instalación de Windows (Administradores, Usuarios, etc.) no pueden eliminarse. En efecto, estos grupos poseen permisos a nivel de la máquina que asignan permisos específicos a las cuentas de usuario que serán miembros de estos grupos predefinidos. Por ejemplo, el grupo de Administradores dará un acceso completo a la máquina Windows a los miembros de este grupo.

4.3.1 Obtener la lista de grupos locales

El cmdlet Get-LocalGroup permite mostrar el conjunto de grupos locales presentes en el puesto de trabajo.

Parámetro	Descripción
-Name<String[]>	Designa el nombre del grupo o de los grupos que se devolverán.
-SID<SecurityIdentifier[]>	Designa el identificador de seguridad del grupo o de los grupos de seguridad que se devolverán.

Ejemplo: recuperar el conjunto de grupos

```
PS C:\Windows\system32> Get-LocalGroup~~
~~
Name~~~~~ Description~~
-----
Administradores~~~~~ Los miembros del grupo  ↔
    Administradores ...~~
Administradores de Hyper-V~~~~~ Los miembros de este grupo disponen  ↔
    de un...~~
```

```

Duplicadores~~~~~ Tiene en cuenta la replicación de  ←
  archi...~~
IIS_IUSRS~~~~~ Grupo integrado utilizado por los  ←
  servic...~
Invitados~~~~~ Los miembros del grupo Invitados  ←
  disponen...~~
Lectores de registros de eventos~~~~~ Los miembros de este grupo pueden  ←
  leer...~~
Operadores de asistencia de control de acceso Los miembros de este grupo pueden  ←
  leer...~~
Operadores criptográficos~~~~~ Los miembros están autorizados a  ←
  realizar...~~
Operadores de configuración de red~~~~~ Los miembros de este grupo disponen  ←
  de...~~
Operadores de copia de seguridad~~~~~ Los miembros del grupo Operadores de  ←
  su...~~
System Managed Accounts Group~~~~~ Los miembros de este grupo se  ←
  administran...~~
Usuarios~~~~~ Los usuarios no pueden realizar...~~
Usuarios avanzados~~~~~ Los usuarios avanzados están inc  ←
  ...~~
Usuarios de administración remota~~~~~ Los miembros de este grupo tienen  ←
  acceso...~~
Usuarios del Monitor de rendimiento~~~~~ Los miembros de este grupo puede  ←
  acceder...~~
Usuarios de Escritorio remoto~~~~~ Los miembros de este grupo disponen  ←
  de ...~~
Usuarios del registro de rendimiento~~~~~ Los miembros de este grupo pueden  ←
  plani...~~
Usuarios del modelo COM distribuido~~~~~ Los miembros tiene permisos para  ←
  ejecut...

```

Se obtiene el conjunto de grupos presentes en el puesto de trabajo. En el caso anterior, se trata del conjunto de grupos predefinidos por Windows 10 (los grupos predefinidos pueden variar en función de la versión de Windows).

Ejemplo: mostrar las propiedades de un grupo

```

PS C:\Windows\system32> Get-LocalGroup -Name Administradores | Format-List *~~
~~
~~
Description~~~~: Los miembros del grupo Administradores disponen de acceso~~
~~~~completo e ilimitado al equipo y al dominio~~
Name~~~~~: Administradores~~
SID~~~~~: S-1-5-32-544~~
PrincipalSource: Local~~
ObjectClass~~~~: Grupo

```

Como puede observar, también existe un SID para los grupos de seguridad. Windows detecta los grupos a través de este identificador y determina así los permisos asignados a los grupos.

4.3.2 Crear un grupo local

La creación de un grupo local se realiza mediante el cmdlet `New-LocalGroup`. Como ocurre con las cuentas de usuario, un grupo recién creado no poseerá ningún permiso en la máquina. Debe saber también que no es posible incluir un grupo local como miembro de otro grupo local. Los grupos locales creados pueden, sin embargo, configurarse para definir permisos NTFS sobre carpetas o archivos, también sobre recursos compartidos. De este modo, todos los miembros presentes en este grupo estarán afectados por la configuración que se haya definido.

He aquí los parámetros de New-LocalGroup:

Parámetro	Descripción
-Description<String>	Introduce una descripción para el grupo creado.
-Name<String>	Especifica el nombre del grupo que se va a crear.

Ejemplo: crear un grupo

```
PS C:\Windows\system32> New-LocalGroup -Name LabGroup_Users~~  
-Description "Los miembros del grupo LabGroup_Users pueden acceder"
```

Se ha creado el grupo LabGroup_Devices. Desde la interfaz gráfica de Windows, se verá aparecer el grupo.

4.3.3 Modificar las propiedades de un grupo local

Para modificar las propiedades de un grupo creado, puede utilizar el cmdlet Set-LocalGroup. Con este comando solo es posible modificar la descripción del grupo. He aquí los parámetros:

Parámetro	Descripción
-Description<String>	Define la nueva descripción del grupo indicado.
-Name<String>	Designa el nombre del grupo que se va a modificar.
-SID<SecurityIdentifier>	Designa el identificador de seguridad del grupo que se va a modificar.

Ejemplo: crear un grupo

```
PS C:\Windows\system32> Set-LocalGroup -Name LabGroup_Users~~  
-Description "Miembros con acceso al recurso compartido Lab"
```

La descripción del grupo se reemplaza por la nueva descripción indicada en el comando.

4.3.4 Renombrar un grupo local

Para editar el nombre de un grupo, se utiliza el cmdlet Rename-LocalGroup. Será necesario indicar el grupo que se va a renombrar, mediante su nombre actual o SID, y definir a continuación el nuevo nombre que se le asignará.

Parámetro	Descripción
-Name<String>	Designa el nombre del grupo que se va a renombrar.
-NewName<String>	Especifica el nuevo nombre que se asignará al grupo.
-SID<SecurityIdentifier>	Designa el identificador de seguridad del grupo que se va a renombrar.

Ejemplo: renombrar un grupo

```
PS C:\Windows\system32> Rename-LocalGroup -Name LabGroup_Users~~  
-NewName Group_LabFolderAccess
```

4.3.5 Eliminar un grupo local

La última acción posible sobre los grupos es la eliminación. Esta operación se realiza mediante el cmdlet Remove-LocalGroup. Observe que los objetos presentes en el grupo que se elimina no se verán afectados. He aquí los parámetros:

Parámetro	Descripción
-Name<String[]>	Designa el nombre del grupo o de los grupos que hay que eliminar.
-SID<SecurityIdentifier[]>	Designa el identificador de seguridad del grupo o de los grupos que hay que eliminar.

Ejemplo: eliminar un grupo

```
PS C:\Windows\system32> Remove-LocalGroup -Name
Group_LabFolderAccess
```

Se elimina el grupo Group_LabFolderAccess. Tenga precaución, pues no se pide confirmación; puede añadir una solicitud de confirmación de eliminación agregando el parámetro -Confirm.

Las referencias a este grupo eliminado definidas en la seguridad NTFS dejan de estar activas y, como ocurre con las cuentas de usuario, el hecho de recrear un grupo con el mismo nombre no es una solución reversible: el SID será necesariamente diferente. Habrá que volver a aplicar todos los permisos NTFS.

4.4 Administrar los miembros de un grupo

Una vez creado un grupo local, es preciso agregar los miembros, pues en caso contrario el grupo no tiene ningún interés. Veremos cómo agregar objetos (cuentas de usuarios locales o de dominio, grupos de dominio, etc.) y también cómo eliminarlos de los grupos.

4.4.1 Agregar miembros en un grupo local

Es posible agregar miembros en un grupo mediante el cmdlet Add-LocalGroupMember. A continuación, es preciso indicar el grupo, utilizando su nombre o SID, así como la cuenta o las cuentas de usuario que serán miembros de dicho grupo. He aquí los parámetros:

Parámetro	Descripción
-Group<LocalGroup>	Designa el grupo al que se agregarán los miembros.
-Member<LocalPrincipal[]>	Indica el objeto o los objetos que se agregarán al grupo indicado. Los objetos especificados pueden definirse por su nombre, SID o LocalPrincipal.
-Name<String>	Designa el nombre del grupo al que se agregarán los miembros.
-SID<SecurityIdentifier>	Designa el SID del grupo al que se agregarán los miembros.

Ejemplo: agregar un miembro al grupo Administradores

```
PS C:\Windows\system32> Add-LocalGroupMember -Group "Administrators" ~
-Member "Julien.Musy"
```

Puede definir el grupo sobre el que desea realizar la operación bien mediante el parámetro -Group o bien mediante -Name, o incluso -SID.

La cuenta de usuario local Julien.Musy se ha agregado al grupo predefinido Administradores. En lo sucesivo, el usuario puede iniciar una sesión en el puesto de trabajo con todos los permisos.

Ejemplo: agregar miembros a un grupo local

```
PS C:\Windows\system32> Add-LocalGroupMember -Group "LabGroup_Users" ~~
-Member "CONTOSO\Julien", "CONTOSO\Benjamin", "CONTOSO\Octavio", ~~
"CONTOSO\GG DSI"
```

El alcance de `Add-LocalGroupMember` no se limita únicamente a las cuentas locales. También es posible agregar cuentas de Microsoft, Azure Active Directory e incluso provenientes de otros dominios.

Ejemplo: agregar miembros a un grupo local con un objeto `LocalPrincipal`

```
PS C:\Windows\system32> Get-LocalGroupMember -Group LabGroup_Users ~~
-Member * | Add-LocalGroupMember -Group LabGroup_Access
```

El resultado devuelto por el cmdlet `Get-LocalGroupMember` es de tipo `LocalPrincipal`, de modo que puede transferirse al cmdlet `Add-LocalGroupMember` para agregar miembros a un grupo.

4.4.2 Obtener los miembros presentes en un grupo local

Para recuperar el conjunto de objetos que son miembros de un grupo, se utiliza `Get-LocalGroupMember`.

Parámetro	Descripción
-Group<LocalGroup>	Designa el grupo cuyos miembros se recuperarán.
-Member<String>	Indica el objeto miembro del grupo que se recuperará. El objeto especificado puede definirse por su nombre o SID, y se admite el uso del carácter comodín * (wildcard).
-Name<String>	Designa el nombre del grupo cuyos miembros se recuperarán.
-SID<SecurityIdentifier>	Designa el SID del grupo cuyos miembros se recuperarán.

Ejemplo: recuperar todos los miembros de un grupo local

```
PS C:\Windows\system32> Get-LocalGroupMember -Group "LabGroup_Users" ~~
~~
ObjectClass Name~~~~~ PrincipalSource~~
-----
Usuario~~~~~CONTOSO\Benjamin ActiveDirectory~~
Grupo~~~~~CONTOSO\GG DSI~~ ActiveDirectory~~
Usuario~~~~~CONTOSO\Julien~ ~ActiveDirectory~~
Usuario~~~~~CONTOSO\Octavio~ ActiveDirectory~~
Usuario~~~~~W10\Julien.Musy~ Local
```

El comando anterior devuelve el conjunto de miembros presentes en el grupo `LabGroup_Users`. El resultado del comando indica también claramente de qué tipo de objeto se trata, así como el origen del objeto (Local, Active Directory, etc.).

Ejemplo: recuperar los miembros (cuentas de usuario) de un grupo proveniente de un dominio específico

```
PS C:\Windows\system32> Get-LocalGroupMember -Group "LabGroup_Users" ~~
-Member CONTOSO\* | Where-Object {$_.ObjectClass -eq "Usuario"}~~
~~
ObjectClass Name~~~~~ PrincipalSource~~
-----
Usuario~~~~~CONTOSO\Benjamin ActiveDirectory~~
Usuario~~~~~CONTOSO\Julien~~ ActiveDirectory~~
Usuario~~~~~CONTOSO\Octavio~ ActiveDirectory
```

En este ejemplo, solo se muestran las cuentas de usuario provenientes del dominio `CONTOSO`.

4.4.3 Eliminar miembros de un grupo local

Para terminar, la última acción que es posible realizar es la eliminación de los miembros presentes en un grupo. Esta operación se lleva a cabo mediante `Remove-LocalGroupMember`. Como en los anteriores cmdlets de esta sección, hay que indicar el grupo, así como el miembro o los miembros que se eliminarán.

Parámetro	Descripción
-Group<LocalGroup>	Designa el grupo cuyos miembros se eliminarán.
-Member<LocalPrincipal[]>	Indica el objeto o los objetos que se eliminarán en el grupo indicado. Los objetos especificados pueden definirse mediante su nombre, SID o LocalPrincipal.
-Name<String>	Designa el nombre del grupo cuyos miembros se eliminarán.
-SID<SecurityIdentifier>	Designa el SID del grupo cuyos miembros se eliminarán.

Ejemplo: eliminar un miembro presente en un grupo

```
PS C:\Windows\system32> Remove-LocalGroupMember -Group LabGroup_Users ~~  
-Member "W10\Julien.Musy", "CONTOSO\Octavio"
```

Los miembros definidos en el comando se eliminan del grupo. Estas cuentas ya no poseen los permisos de acceso que tuvieran por ser miembros de este grupo.