

m



**Ciclo:** Administración de Sist. Informáticos en Red **Curso:** 2022/23  
**Módulo:** IMPLANTACIÓN DE SISTEMAS OPERATIVOS  
**Grupo:** 1º ASIR

## PR0308: Creación automática de usuarios

**Nombre/s:** Luis Escobar

Tienes que contestar las preguntas en este mismo fichero después de cada pregunta. No te olvides de poner tu nombre en el recuadro superior.

Quando hayas acabado todas las prácticas renombas el fichero para que se llame **{Apellido1} {Apellido2}, {Nombre} – PR0307**. En el nombre y apellidos la primera mayúscula y el resto en minúsculas. El fichero tiene que estar en formato PDF. **Cualquier fichero que no siga esta nomenclatura o no esté en PDF no será corregido**. El fichero final lo tienes que subir a la plataforma.

### Ejercicio 1: Funciones avanzadas. Script de creación de grupos

Vamos a automatizar la tarea de creación de usuarios mediante un script.

**1.-** Implementa un script que lea un fichero y cree una serie de usuarios a partir de esos datos. El fichero tiene que tener los datos: nombre de usuario, nombre completo y descripción. Por defecto los usuarios creados no tendrán contraseña.

Verifica que funcione.

```
PS D:\iso\Prácticas\scripts> Import-Csv .\users.csv | ForEach-Object {New-LocalUser -name $_.nombreusuario -FullName $_.nombrecompleto -Description $_.descripcion -NoPassword}

Name Enabled Description
----
manuel True 9339
paco True 3343
saul True 3430

PS D:\iso\Prácticas\scripts>
```

**2.-** Modifica el script anterior para que todos los usuarios que se creen tengan como contraseña 1234.

```
PS D:\iso\Prácticas\scripts> Import-Csv .\users.csv | ForEach-Object {New-LocalUser -name $_.nombreusuario -FullName $_.nombrecompleto -Description $_.descripcion -Password (ConvertTo-SecureString "1234" -force)}

Name Enabled Description
----
manuel True 9339
paco True 3343
saul True 3430

PS D:\iso\Prácticas\scripts>
```

**3.-** Modifica el script para que la contraseña se pida por teclado una vez, utilizando la misma para todos los usuarios

```
PS D:\iso\Prácticas\scripts> $cont= Read-Host -Prompt "introduce la contraseña" -AsSecureString; Import-Csv ./users.csv | ForEach-Object {New-LocalUser $_.nombredeusuario -Password $cont}
introduce la contraseña: ****

Name Enabled Description
-----
mario True

PS D:\iso\Prácticas\scripts>
```

#### 4.- Modifica el script para que la contraseña se pida por teclado para cada usuario.

```
PS D:\iso\Prácticas\scripts> Import-Csv ./users.csv | foreach {New-LocalUser -name $_.nombredeusuario -Password (Read-Host -Prompt "Introduce la contraseña" -AsSecureString)}
Introduce la contraseña: ****

Introduce la contraseña: ****
Name Enabled Description
-----
sonia True
carla True

PS D:\iso\Prácticas\scripts>
```

#### 5.- Coge el script del punto 2 y modifícalo para que pida la descripción del usuario por teclado en lugar de leerlo del fichero CSV

```
PS D:\iso\Prácticas\scripts> Import-Csv ./users.csv | ForEach-Object {New-LocalUser -name $_.nombredeusuario -FullName $_.nombrecompleto -Description (Read-Host) -Password (ConvertTo-SecureString "1234" -force)}
asdkfk

Name Enabled Description
-----
asly True asdkfk

PS D:\iso\Prácticas\scripts>
```

#### 6.- Modifica el script anterior para que al principio cree un grupo llamado GRUPO\_{iniciales}, donde iniciales son las tuyas, y que agregue a todos los usuarios creados a dicho grupo.

```
PS D:\iso\prácticas\scripts> New-LocalGroup GRUPO_LoaeA; Import-Csv ./users.csv | ForEach-Object {New-LocalUser -name $_.nombredeusuario -FullName $_.nombrecompleto -Description (Read-Host) -Password (ConvertTo-SecureString -AsPlainText '1234' -force); Add-LocalGroupMember GRUPO_LEEA $_.nombre}

Name Description
-----
GRUPO_LoaeA

AccountExpires :
Description :
Enabled : True
FullName : gonzalez
PasswordChangeableDate : 21/11/2022 13:19:52
PasswordExpires : 02/01/2023 13:19:52
UserMayChangePassword : True
PasswordRequired : False
PasswordLastSet : 21/11/2022 13:19:52
LastLogon :
Name : david
SID : S-1-5-21-4057441978-157600961-3064114078-1031
PrincipalSource : Local
ObjectClass : Usuario
```

#### 7.- Volvemos al script del punto 2 y modifícalo para que al principio lea otro fichero CSV que contendrá un listado de grupos que creará antes de crear los usuarios.

8.- Modifica el script anterior añadiendo una columna más en el fichero CSV con los usuarios que contendrá un nombre de grupo. Primero creará los grupos, luego los usuarios, y para cada usuario lo añadirá al grupo correspondiente.

```
Write-Host "añadiendo usuario" "$_.name" al grupo $_.group_name
Add-LocalGroupMember
    -group $_.group_name `
    -Member $_.name
    | Out-Null
Add-LocalGroupMember `
    -group "usuarios" `
    -member $_.name `
    | Out-Null
```

```
scripts > groups.csv
1 group_name
2 grupo_1
3 grupo_2
4 grupo_3
```

## Ejercicio 2: Complicamos las cosas (opcional)

Si has acabado pronto puedes intentar hacer esta tarea opcional. El objetivo consiste en que tendremos dos ficheros CSV:

- Uno contiene datos de grupos, con el nombre y descripción de cada uno
- El otro contiene datos de alumnos con, por lo menos, el nombre de usuario, nombre completo y la descripción, así como el nombre de un grupo de los incluidos en el fichero anterior.

El objetivo será crear en primer lugar todos los grupos del primer fichero y luego iterar sobre el segundo fichero creando los usuarios y añadiéndolos al grupo que se indica.

## Ejercicio 5: ¿Ganas de más? (opcional)

Con Powershell podemos administrar todo lo relativo a Microsoft, no solo Windows 10. Por ejemplo, podemos trabajar con Azure o con las herramientas de Office 365 como puede ser Microsoft Teams, lo cual será el objetivo de este ejercicio.

Para poder hacerlo primero tienes que instalar un módulo específico que incluye los *cmdlets* necesarios para operar con Teams. Las instrucciones para hacerlo las tienes en <https://docs.microsoft.com/es-es/microsoftteams/teams-powershell-install>. Ahí también te explica cómo tienes que conectar con Teams antes de lanzar cualquier otro comando.

La referencia de los *cmdlets* la tienes en <https://docs.microsoft.com/es-es/powershell/module/teams/?view=teams-ps> con detalle de cada uno de los comandos. Necesitarás los siguientes:

- Add-TeamUser: añade un usuario a un equipo
- New-Team: crea un equipo nuevo. Ten en cuenta que este comando devuelve un identificador que necesitarás para añadir miembros posteriormente al equipo (en el ejemplo 2 de la referencia se ve como recoger este identificador en una variable).

Vas a hacer un script que:

- Recoja un nombre de equipo por teclado.
- Conecte al *Teams* (aquí te pedirá tus credenciales).
- Crea un equipo con el nombre que leyó por teclado en el primer paso.
- Lea una lista de usuarios de un fichero CSV. Estos usuarios estarán identificados por su cuenta @educa.jcyl.es, así que pídeles la cuenta a 2 o 3 de tus compañeros.
- Añade cada uno de esos usuarios al equipo que creaste.
- Desconecta del Teams.