

PR0308: Creación automática de usuarios

Nombre/s: Diego Fernández Pita

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

Cuando hayas acabado todas las prácticas renombra 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.

```
Import-Csv .\ejer3_8_1.csv | ForEach-Object {
    New-LocalUser -Name $_.nombre -FullName $_.nombreCompleto -Description $_.descripcion -NoPassword `
}

PS Microsoft.PowerShell.Core\FileSystem::\\tsclient\D\ISO\practicas\SCRIPTS> .\script3_8_1.ps1

Name      Enabled Description
-----
prueba1 True      Este es el usuario prueba 1
prueba2 True      Este es el usuario prueba 2
```

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

```
Import-Csv .\ejer3_8_1.csv | ForEach-Object {
    New-LocalUser -Name $_.nombre -FullName $_.nombreCompleto -Description $_.descripcion -
    -Password (ConvertTo-SecureString "1234" -AsPlainText -Force)
}

PS Microsoft.PowerShell.Core\FileSystem::\\tsclient\D\ISO\practicas\SCRIPTS> .\script3_8_1.ps1

Name      Enabled Description
-----
prueba3 True      Este es el usuario prueba 3
prueba4 True      Este es el usuario prueba 4
```

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

```
$pass = Read-Host -Prompt "Introduce la contraseña para los usuarios" -AsSecureString
Import-Csv .\ejer3_8_1.csv | ForEach-Object {
    New-LocalUser -Name $_.nombre -FullName $_.nombreCompleto -Description $_.descripcion `
    -Password $pass
}
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\tsclient\D\ISO\practicas\SCRIPTS> .\script3_8_1.ps1
Introduce la contraseña para los usuarios: ****

Name      Enabled Description
----      -
prueba7 True      Este es el usuario prueba 7
prueba8 True      Este es el usuario prueba 8
```

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

```
Import-Csv .\ejer3_8_1.csv | ForEach-Object {
    $pass = Read-Host -Prompt "Contraseña" -AsSecureString
    New-LocalUser -Name $_.nombre -FullName $_.nombreCompleto -Description $_.descripcion `
    -Password $pass
}
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\tsclient\D\ISO\practicas\SCRIPTS> .\script3_8_1.ps1
Contraseña: ****

Contraseña: ****
Name      Enabled Description
----      -
prueba9 True      Este es el usuario prueba 9
prueba10 True      Este es el usuario prueba 10
```

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

```
Import-Csv .\ejer3_8_1.csv | ForEach-Object {
    $desc = Read-Host -Prompt "Descripción"
    New-LocalUser -Name $_.nombre -FullName $_.nombreCompleto -Description $desc `
    -Password (ConvertTo-SecureString -AsPlainText "1234" -Force)
}
```

```
$desc = Read-Host -Prompt "Indica la descripción para" $_.name
```

```
PS Microsoft.PowerShell.Core\FileSystem::\\tsclient\D\ISO\practicas\SCRIPTS> .\script3_8_1.ps1
Descripción: Esta es la descripción del usuario de prueba

Descripción: Esta es otra descripción
Name      Enabled Description
----      -
prueba11 True      Esta es la descripción del usuario de prueba
prueba12 True      Esta es otra descripción
```

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.

```
New-LocalGroup -Name "GRUPO_DFP" | Out-Null

Import-Csv .\ejer3_8_1.csv | ForEach-Object {
    $desc = Read-Host -Prompt "Descripción"
    New-LocalUser -Name $_.nombre -FullName $_.nombreCompleto -Description $desc `
    -Password (ConvertTo-SecureString -AsPlainText "1234" -Force)

    Add-LocalGroupMember -Group GRUPO_DFP -Member $_.nombre | Out-Null
}
```

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

```
Import-Csv .\groups.csv | ForEach-Object { `
    New-LocalGroup -Name $_.nombre_grupos | Out-Null
    Write-Host "Se ha creado el grupo" $_.nombre_grupos
}
```

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.

```
Import-Csv .\groups.csv | ForEach-Object { `
    New-LocalGroup -Name $_.nombre_grupos | Out-Null
    Write-Host "Se ha creado el grupo" $_.nombre_grupos
}

Import-Csv .\ejer3_8_1.csv | ForEach-Object { `
    $pass = Read-Host -Prompt 'Indica la contraseña para '$_.name
    `
    `
    `
    -AsSecureString
    $desc = Read-Host -Prompt "Indica la descripción para "$_.name
    New-LocalUser `
    `
    -Name $_.nombre `
    -FullName $_.nombreCompleto `
    -Description $desc `
    -Password $pass `
    `
    Add-LocalGroupMember -Group $_.nombre_grupos -Member $_.nombre | Out-Null
}
```

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.