

# Configuración de contraseñas en *Microsoft Windows* y GNU/Linux

Ismael Macareno Chouikh

2024-11-15

## Índice

<b>1. Instrucciones</b>	<b>2</b>
<b>2. <i>Microsfot Windows</i></b>	<b>2</b>
2.1. Pruebas . . . . .	6
<b>3. GNU/Linux</b>	<b>7</b>
3.1. Pruebas . . . . .	9
3.2. Creación de usuarios con directorios separados . . . . .	9
3.3. Creación del grupo <b>proyecto</b> y asociación de usuarios al mismo . . . . .	10
3.4. Restricción de acceso a directorios no propios de los usuarios . . . . .	10
3.5. Cuota de disco para usuarios (Asociada a los 2 usuarios) . . . . .	10
<b>4. Valoración Personal</b>	<b>11</b>
<b>5. Bibliografía</b>	<b>12</b>

## 1. Instrucciones

Se adjunta la siguiente práctica guiada que se puede unir al ejemplo que se aportan al temario. [Adjunto fichero PDF](#)

Se aportan 2 webs de referencia:

- [Gestión de políticas de contraseñas en Linux](#)
- [Políticas de contraseñas en Debian](#)

Para Ubuntu 24.04 no aparecen `pam_cracklib`, usa `pam_passwdqc` o `pam_pwquality`

Se adjunta también fotografía de cada uno de los parámetros de políticas de contraseñas a configurar

Argumento	Descripción
<code>debug</code>	Escribe información de depuración en el log del sistema. Esta información nunca incluye la contraseña introducida.
<code>type=XXX</code>	Reemplaza la palabra UNIX que se muestra cuando se solicita la nueva contraseña por la palabra especificada.
<code>retry=N</code>	Número de reintentos para cambiar la contraseña. Por defecto el valor es 1.
<code>difork=N</code>	Número de caracteres que debe diferir la nueva contraseña de la anterior. Por defecto el valor es 10.
<code>minlen=N</code>	Número mínimo de caracteres que son aceptables para la contraseña. Es necesario tener en cuenta que algunos caracteres pueden contar como más de 1.
<code>dcredit=N</code>	Si N es mayor que 0 indica por cuantos caracteres cuenta un dígito, siendo el valor por defecto de 1. Si N es menor que 0 indica el número mínimo de dígitos que tiene que tener la contraseña.
<code>ucredit=N</code>	Igual que <code>dcredit</code> pero para las letras mayúsculas.
<code>lcredit=N</code>	Igual que <code>dcredit</code> pero para las letras minúsculas.
<code>ocredit=N</code>	Igual que <code>dcredit</code> pero para los caracteres que no son letras o números.
<code>use_authtok</code>	Fuerza a utilizar la contraseña solicitada por un módulo previo de tipo <code>password</code> .

Figura 1: Macareno, Ismael. (2024). Parámetros de Políticas de Contraseñas a configurar [PNG]. Propia

## 2. *Microsoft Windows*

### HAY QUE USAR UNA VERSIÓN DE *WINDOWS* QUE NO SEA LA *HOME*

Lo primero que tendremos que hacer será acceder a las directivas de seguridad local, para ello haremos lo siguiente

- Combinación de teclas Win + R
  - `secpol.msc`

Una vez estemos en el `secpol` lo que haremos será establecer una política de seguridad para las contraseñas. Para esto lo que haremos será ir a **Directivas de cuenta/Directivas de contraseñas**

Una vez ahí tendremos que configurar los siguientes parámetros:

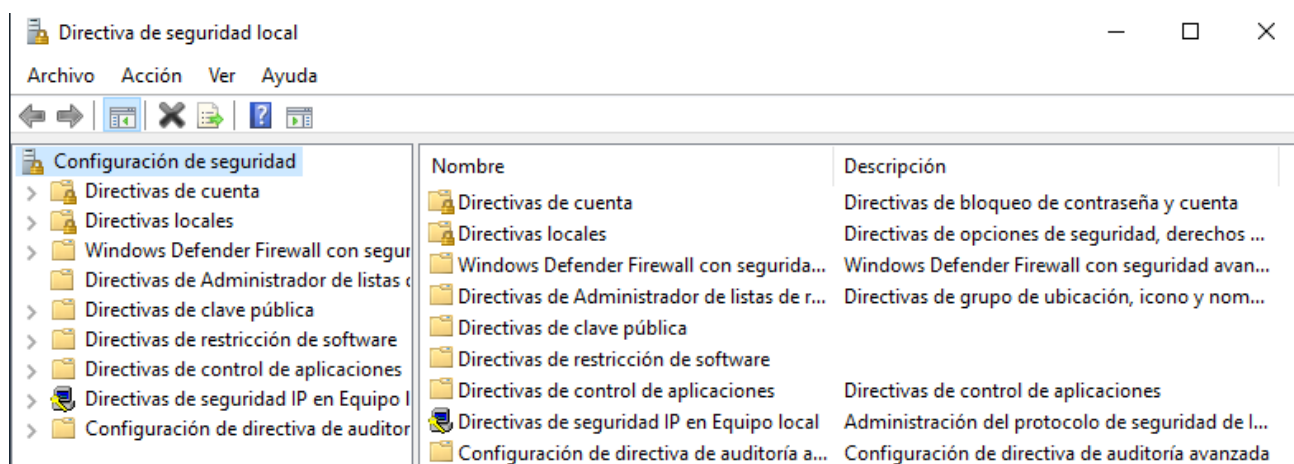
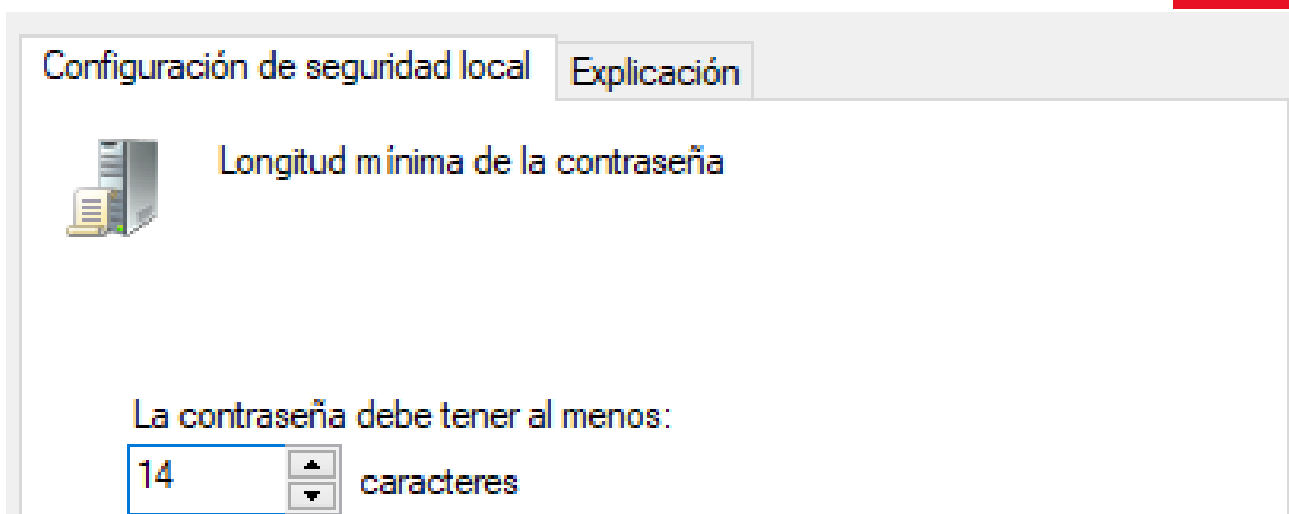


Figura 2: Macareno, Ismael. (2024). Directivas de Seguridad Local *Windows 10 LTSC* [PNG]. Propia

- Mínimo de 14 dígitos

### Propiedades: Longitud mínima de la contraseña




- La contraseña debe contener los requisitos mínimos de tipos de caracteres utilizados
  - Mayúsculas
  - Minúsculas
  - Dígitos
  - Carácter no alfanumérico

## Propiedades: La contraseña debe cumplir los requisitos de...



Configuración de seguridad local Explicación

 La contraseña debe cumplir los requisitos de complejidad

☒ Habilitada


☐ Deshabilitada

- Vigencia máxima de 30 días

## Propiedades: Vigencia máxima de la contraseña



Configuración de seguridad local Explicación

 Vigencia máxima de la contraseña

La contraseña expirará en:


días

- Vigencia mínima de 7 días

## Propiedades: Vigencia mínima de la contraseña



Configuración de seguridad local Explicación

 Vigencia mínima de la contraseña

La contraseña se puede cambiar después de:


días

- Tener un historial de 10 contraseñas

## Propiedades: Exigir historial de contraseñas



Configuración de seguridad local Explicación

 Exigir historial de contraseñas

Guardar el historial de contraseñas durante:

contraseñas recordadas


Estos dos últimos apartados se deben realizar en /Directivas de cuenta/Directiva de bloqueo de cuenta

- Se bloquee después del quinto intento de acceso fallido

## Propiedades: Umbral de bloqueo de cuenta



Configuración de seguridad local Explicación

 Umbral de bloqueo de cuenta

La cuenta se bloqueará después de:


intentos de inicio de sesión no válidos

- Se desbloquea a los 2 minutos

## Propiedades: Duración del bloqueo de cuenta



Configuración de seguridad local Explicación

 Duración del bloqueo de cuenta

La cuenta se bloqueará durante:

minutos

### 2.1. Pruebas

La primera prueba que voy a realizar es crear un usuario y establecer una contraseña débil (Ej. 1234)

Usuario nuevo

?

✕

Nombre de usuario:

ejemplopass

Nombre completo:

ejemplopass

Descripción:

Usuarios y grupos locales

✕



Ocurrió el siguiente error al intentar crear el usuario ejemplopass en el equipo WIN10-LTSC:

La contraseña no cumple con los requisitos de la directiva de contraseñas. Compruebe los requisitos de longitud mínima, complejidad e historial de la contraseña.

Aceptar

Si creásemos un usuario con una contraseña aceptada por los criterios establecidos no nos daría ningún tipo de problema

### 3. GNU/Linux

El control sobre complejidad y cifrado en contraseñas se realiza en GNU/Linux mediante el servicio PAM (*Pluggable Authentication Module*). Mediante PAM podemos comunicar a nuestras aplicaciones con los métodos de autenticación que deseemos de una forma transparente, lo que permite integrar las utilidades de un sistema UNIX clásico (*login*, *ftp*, *telnet*) con esquemas diferentes del habitual *password*: claves de un solo uso, biométricos, tarjetas inteligentes...

El módulo `pam_cracklib` está hecho específicamente para determinar si es suficientemente fuerte una contraseña que se va a crear o modificar con el comando `passwd`

Hay otro paquete que es `pam_pwquality` que sustituye de alguna manera a `pam_cracklib`, y cuya configuración está dentro del archivo `pwquality.conf` (necesario su uso en la versión Ubuntu 24.04)

```
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
```

```
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password      requisite                                pam_pwquality.so retry=3
password      [success=2 default=ignore]              pam_unix.so obscure use_authtok try_first_pass yescryp
password      sufficient                              pam_sss.so use_authtok
# here's the fallback if no module succeeds
password      requisite                                pam_deny.so
```

Hay que tener en cuenta que la práctica nos pide que pongamos las siguientes normas para las contraseñas:

- Mínimo 1 dígito
- 2 minúsculas
- 1 mayúscula
- 1 carácter no alfanumérico
- no coincidir con al menos 3 letras de la anterior
- Longitud mínima de 12 caracteres
- Vigencia máxima de 30 días
- Vigencia mínima de 7 días
- Historial de 10 contraseñas
- Bloqueo de cuenta a los 5 fallos
- Desbloqueo de cuenta pasados 2 minutos

Para modificar la exigencia de las contraseñas en Ubuntu 24.04 lo que haremos será editar el fichero `/etc/security/pwquality.conf`. En este fichero tendremos que modificar los siguiente parámetros:

- `minlen : 12` #Para que la longitud mínima de la contraseña sea de 12
- `dcredit: 1` #Para que tenga mínimo 1 dígito
- `lcredit: 2` #Para que tenga mínimos 2 minúsculas
- `ucredit: 1` #Para que tenga mínimo 1 mayúscula
- `ocredit: 1` #Para que tenga mínimo 1 carácter no alfanumérico
- `maxrepeat: 3` #Para que no se pueda poner el mismo carácter más de tres veces seguidas
- `remember: 10` #Para que recuerde las últimas 10 contraseñas

Obviamente nos quedan parámetros que la práctica que terminar como por ejemplo las vigencias de las contraseña, el bloqueo de cuenta y el tiempo que tarda el sistema en desbloquear la cuenta después de los fallos.

Para establecer la vigencia de las contraseñas usaremos el comando `chage` de la siguiente manera:

```
root@makaSAD:~# chage -m 7 -M 30 maka
```

Ejecutando el comando que se puede ver arriba lo que estamos haciendo es poner una vigencia mínima de 7 días y una vigencia máxima de 30 días.



Para los últimos puntos que nos faltan (bloquear el acceso a los 5 fallos y que se desbloqué a los 2 minutos) lo que tendremos que hacer será modificar el fichero `/etc/pam.d/common-auth` de la siguiente manera:

```
# since the modules above will each just jump around
auth      required      pam_faillock.so preauth audit deny=5 unlock_time=120
```

### 3.1. Pruebas

Para probar las configuraciones establecidas lo que haré será crear dos usuarios llamados `alumnoFCT1` y `alumnoFCT2` con sus respectivos directorios en `/home` separados. Para crear estos usuarios lo que hago es ejecutar el siguiente comando con privilegios de superusuario

```
root@makaSAD:~# adduser alumnofct1
```

Al usar el comando `adduser` en vez del comando `useradd` se nos pide que le pongamos la contraseña al usuario en el mismo proceso. Si intentamos poner una contraseña débil (Ej. 12345) nos saltaran errores del tipo

```
BAD PASSWORD: The password is shorter than 5 characters
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password fails the dictionary check - it is too short
Retype new password:
Sorry, passwords do not match.
passwd: Have exhausted maximum number of retries for service
passwd: password unchanged
Try again? [y/N] y
New password:
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password:
```

### 3.2. Creación de usuarios con directorios separados

Para crear dos usuarios se ejecutará el comando `adduser NOMBREUSUARIO`

```
root@makasad:~# ls -la /home/
total 24
drwxr-xr-x  6 root      root      4096 Nov 14 21:18 .
drwxr-xr-x 23 root      root      4096 Sep 23 17:46 ..
drwxr-x---  2 alumnofct1 alumnofct1 4096 Nov 14 21:17 alumnofct1
drwxr-x---  2 alumnofct2 alumnofct2 4096 Nov 14 21:18 alumnofct2
drwxr-x--- 14 fary      fary      4096 Sep 23 18:38 fary
drwxr-x--- 16 maka      maka      4096 Sep 30 17:20 maka
```

Como se puede apreciar en el bloque de código de arriba se ve que hay dos nuevos directorios en `/home`, cada uno para cada usuario creado

```
root@makasad:~# cat /etc/passwd | grep -E 'alumnofct1|alumnofct2'
alumnofct1:x:1002:1002:,,,:/home/alumnofct1:/bin/bash
alumnofct2:x:1003:1003:,,,:/home/alumnofct2:/bin/bash
```

Como se puede apreciar en el bloque de código de arriba hay dos usuarios en el fichero `/etc/passwd` con el mismo nombre que los creados

### 3.3. Creación del grupo proyecto y asociación de usuarios al mismo

Se nos indica crear el grupo `proyecto` y asociar a esté los usuarios

- `alumnoft1`
- `alumnoft2`

Para ello ejecutaremos los siguientes comandos

```
root@makasad:~# addgroup proyecto
info: Selecting GID from range 1000 to 59999 ...
info: Adding group `proyecto' (GID 1004) ...
root@makasad:~# usermod -aG proyecto alumnoft1
root@makasad:~# usermod -aG proyecto alumnoft2
```

### 3.4. Restricción de acceso a directorios no propios de los usuarios

Para restringir el acceso del usuario `alumnoft1` al directorio de `alumnoft2` y viceversa se modificarán los permisos de los directorios `/home/alumnoft[1,2]`.

Para ello no tendremos que ejecutar ningún tipo de comando debido a que si realizamos algún tipo de prueba podremos apreciar que los usuarios `alumnoft1` y `alumnoft2` no pueden acceder al directorio `/home` del otro.

```
root@makasad:~# ls -la /home/
total 24
drwxr-xr-x  6 root      root      4096 Nov 14 21:18 .
drwxr-xr-x 23 root      root      4096 Sep 23 17:46 ..
drwxr-x---  2 alumnoft1 alumnoft1 4096 Nov 14 21:17 alumnoft1
drwxr-x---  2 alumnoft2 alumnoft2 4096 Nov 14 21:18 alumnoft2
drwxr-x--- 14 fary      fary      4096 Sep 23 18:38 fary
drwxr-x--- 16 maka      maka      4096 Sep 30 17:20 maka
root@makasad:~# su - alumnoft2
alumnoft2@makasad:~$ cd /home/alumnoft1
bash: cd: /home/alumnoft1: Permission denied
```

Listing 1: Prueba de acceso desde `alumnoft2` a `alumnoft1`

```
root@makasad:~# su - alumnoft1
alumnoft1@makasad:~$ cd /home/alumnoft2
-bash: cd: /home/alumnoft2: Permission denied
```

Listing 2: Prueba de acceso desde `alumnoft1` a `alumnoft2`

### 3.5. Cuota de disco para usuarios (Asociada a los 2 usuarios)

Crear una cuota de disco para `alumnoFCT1` y `alumnoFCT2`.

Está cuota limitará el uso de espacio a 500 MiBi.

Para realizar esto habrá que editar el fichero `/etc/fstab` añadiendo lo siguiente

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/aea78c2d-3a20-4722-9811-6b5c20abd332 / ext4 defaults 0 1
/swap.img none swap sw 0 0
# quota para alumnoFCT1 y alumnoFCT2
/dev/sda2 / ext4 defaults,usrquota 0 1
```

Luego habrá que ejecutar los siguientes comandos para definir una cuota de 500 MiBi para ambos usuarios creados

```
root@makasad:~# vi /etc/fstab
root@makasad:~# mount -o remount /dev/sda2
root@makasad:~# quotacheck -cum /dev/sda2
root@makasad:~# quotaon /dev/sda2
root@makasad:~# setquota -u alumnofct1 500000 500000 0 0 /dev/sda2
root@makasad:~# setquota -u alumnofct2 500000 500000 0 0 /dev/sda2
```

Para verificar que la cuota esta establecida podremos ejecutar los siguientes comandos

- Comando `repquota`

```
root@makasad:~# repquota /dev/sda2 | grep -E 'alumnofct1|alumnofct2'
alumnofct1 --      16 500000 500000          4    0    0
alumnofct2 --      20 500000 500000          5    0    0
```

- Comando `quota`

```
root@makasad:~# quota -u alumnofct1
Disk quotas for user alumnofct1 (uid 1002):
    Filesystem blocks quota limit grace files quota limit grace
    /dev/sda2   16 500000 500000          4    0    0
root@makasad:~# quota -u alumnofct2
Disk quotas for user alumnofct2 (uid 1003):
    Filesystem blocks quota limit grace files quota limit grace
    /dev/sda2   20 500000 500000          5    0    0
```

## 4. Valoración Personal

Práctica muy interesante, sobre todo la parte de GNU/Linux.

En *Microsoft Windows* no es muy difícil establecer restricciones de contraseñas ya que lo hemos realizado mil veces previamente pero en GNU/Linux llevaba desde mayo del año pasado sin hacerlo y encima ahora ha cambiado porque yo no usaba el `pwquality.conf`

Muy interesante :)

## 5. Bibliografía

- Ubuntu, pwquality.conf
  - <https://manpages.ubuntu.com/manpages/focal/en/man5/pwquality.conf.5.html>
- Comando chage
  - <https://man7.org/linux/man-pages/man1/chage.1.html>