

**SERVIDORES WEB DE ALTAS PRESTACIONES (2021-2022)**  
GRADO EN INGENIERÍA INFORMÁTICA  
UNIVERSIDAD DE GRANADA

---

## Memoria Práctica 2

---

Pedro Antonio Mayorgas Parejo

28 de marzo de 2022

# Índice

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Abstract</b>  | <b>3</b>  |
| <b>2</b> | <b>SSH y acceso remoto sin contraseña</b>  | <b>4</b>  |
| 2.1      | Configurando el servicio SSH . . . . .   | 4         |
| 2.2      | Creando una clave SSH . . . . .  | 6         |
| 2.2.1    | Generando clave RSA . . . . .  | 7         |
| 2.2.2    | Generando clave DSA . . . . .  | 7         |
| 2.3      | Proceso de copiado de clave asimétrica - proceso automático . . . . .                      | 8         |
| 2.4      | Proceso de copiado de clave asimétrica - proceso manual . . . . .                          | 9         |
| 2.5      | Accediendo con una clave o otra . . . . .  | 9         |
| <b>3</b> | <b>Copiar archivos locales con equipos remotos.</b>  | <b>10</b> |
| 3.1      | Uso de Tar para comprimir un directorio y mandarlo a un equipo remoto<br>por SSH . . . . . | 10        |
| 3.2      | Uso de SCP para enviar una copia de un directorio . . . . .                                | 10        |
| <b>4</b> | <b>Rsync</b>   | <b>12</b> |
| 4.1      | Instalación de Rsync . . . . .   | 12        |
| 4.2      | Realizando un backup full con Rsync de un árbol de directorios . . . . .                   | 12        |
| 4.3      | Modo Mirror . . . . .  | 13        |
| 4.4      | Modo Copia incremental . . . . .   | 14        |
| <b>5</b> | <b>Crontab</b>   | <b>17</b> |
| <b>6</b> | <b>Bibliografía</b>  | <b>20</b> |

## **1. Abstract**

En esta práctica he considerado conveniente empezar por la sección de SSH primero para poder facilitarme el resto de tareas a realizar como SCP y Rsync para no tener que utilizar la contraseña de manera frecuente. Así como encaminar la práctica rápidamente a la sección de crontab.

## 2. SSH y acceso remoto sin contraseña

SSH tiene dos formas de acceso directo, una es con contraseña y otra es sin contraseña. Para el acceso sin contraseña debemos tener en cuenta dos cosas:

- Debemos tener configurado que se permita el acceso sin contraseña, en el host destinatario, para ello debemos habilitar la opción **PubkeyAuthentication yes** y la opción **PasswordAuthentication Yes**<sup>1</sup>.
- Elección del tipo de algoritmo de cifrado asimétrico para el contacto entre el Host cliente y el Host servidor.

### 2.1. Configurando el servicio SSH

Las directivas han sido señaladas en la sección principal, en esta subsección se indican las rutas de los ficheros de configuración y reinicios de servicio para poder aplicar dichas directivas.

La ruta del fichero de configuración de SSH es: `/etc/ssh/sshd_config`

Las directivas cambiadas son:

- PasswordAuthentication Yes
- PubkeyAuthentication Yes

---

<sup>1</sup>¿Por qué debemos utilizar contraseña si no queremos usarlas? El comando **ssh-copy-id**. Precisa de la autenticación por contraseña para los servidores a los que no haya acceso físico donde permitan la copia de manera física con un medio. La clave pública del cliente a autenticar. Entonces precisa de tener la directiva de PasswordAuthentication temporalmente para poder realizar dicha la copia y instalación de la clave. Una vez finalizada la copia y si el usuario al que accede por SSH tiene permisos de administración o es administrador, **debe desactivar dicha directiva**.

```
GNU nano 5.4 /etc/ssh/sshd_config
PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no
```

Figura 1: Directiva PasswordAuthentication y PubkeyAuthentication

Finalmente ejecutamos el siguiente comando para establecer en el servicio las directivas nuevas y verificar que se ejecute correctamente:

```
systemctl restart sshd.service
&& systemctl status sshd.service
```

```

pedroamp@m1-pedroamp:~$ sudo systemctl restart ssh && sudo systemctl status sshd.service
sshd.service ssh.service ssh.socket
pedroamp@m1-pedroamp:~$ sudo systemctl restart sshd.service && sudo systemctl status sshd.service
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2022-03-27 18:02:52 CEST; 21ms ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 1177 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 1178 (sshd)
     Tasks: 1 (limit: 4650)
    Memory: 1.1M
       CPU: 44ms
   CGroup: /system.slice/ssh.service
           └─1178 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Mar 27 18:02:52 m1-pedroamp systemd[1]: ssh.service: Succeeded.
Mar 27 18:02:52 m1-pedroamp systemd[1]: Stopped OpenBSD Secure Shell server.
Mar 27 18:02:52 m1-pedroamp systemd[1]: Starting OpenBSD Secure Shell server...
Mar 27 18:02:52 m1-pedroamp sshd[1178]: Server listening on 0.0.0.0 port 22.
Mar 27 18:02:52 m1-pedroamp sshd[1178]: Server listening on :: port 22.
Mar 27 18:02:52 m1-pedroamp systemd[1]: Started OpenBSD Secure Shell server.
pedroamp@m1-pedroamp:~$

```

Figura 2: Reinicio de servicio y comprobación del estado nuevo.

## 2.2. Creando una clave SSH

Para el proceso de creación de una clave necesitamos decidir entre las siguientes claves asimétricas:

- **RSA:** Es usado para la transmisión de datos seguros, es muy rápido cifrando pero lento en el descifrado, pero es ideal para el envío de datos ya que el cifrado rápido permite enviar tanto como se pueda y el cliente tendrá un buffer donde almacenar temporalmente los datos cifrados. Donde los va descifrando y ofreciendo en texto plano.
- **DSA:** Este tipo de algoritmo es utilizado más para firmas digitales y su verificación. Es muy rápido descifrando aunque penaliza el proceso de cifrado. Esto quiere decir que merece la pena el uso original del algoritmo. La firma digital, donde cifra cierto documento plano una vez.

Los tamaños de las claves recomendado: *Es de  $\geq 2048$  bits en RSA* <sup>2</sup>. Luego para DSA el soportado de ssh-keygen es 1024 bits <sup>3</sup>

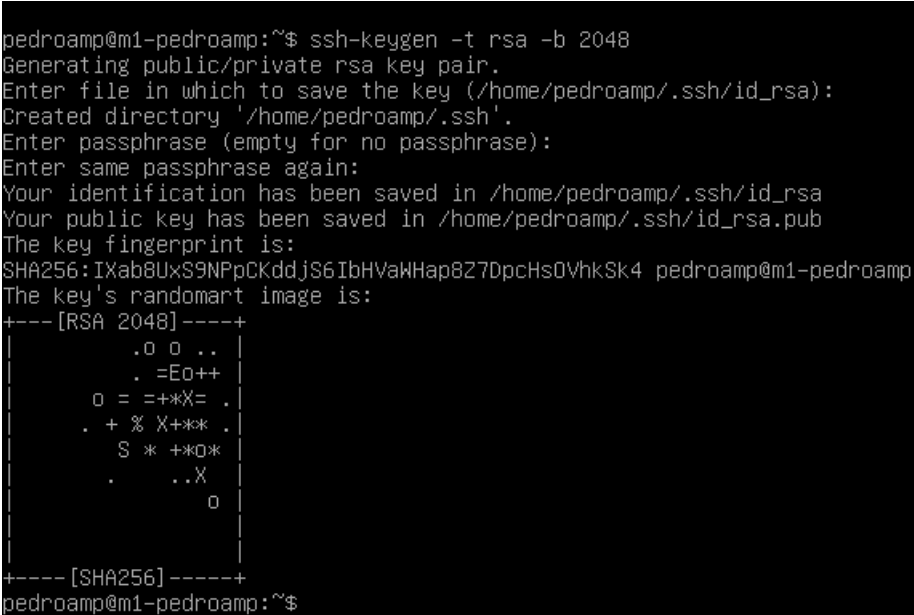
<sup>2</sup>En el manpage de ssh-keygen se indica que para RSA el mínimo es de 1024, por defecto se crea una clave asimétrica de tamaño de 3072.

<sup>3</sup>Para el algoritmo DSA el máximo indicado es de 2048 bits pero por razones que no controlo, no permite pasarse de 1024. Indicado en la bibliografía la documentación técnica de IBM.

### 2.2.1. Generando clave RSA

Para la creación de una clave RSA <sup>4</sup> debemos usar el siguiente comando:

```
ssh-keygen -t rsa -b 2048
```



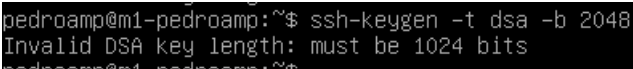
```
pedroamp@m1-pedroamp:~$ ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pedroamp/.ssh/id_rsa):
Created directory '/home/pedroamp/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/pedroamp/.ssh/id_rsa
Your public key has been saved in /home/pedroamp/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:IXab8UxS9NPpCKddjS6IbHVvWHap8Z7DpcHs0VhkSk4 pedroamp@m1-pedroamp
The key's randomart image is:
+---[RSA 2048]-----+
|
|  .o o ..
|  . =Eo++
|  o = =+*X= .
|  . + % X+*k .
|  S * +*o*
|  . ..X
|  o
|
+-----[SHA256]-----+
pedroamp@m1-pedroamp:~$
```

Figura 3: Creación de clave RSA

### 2.2.2. Generando clave DSA

Como he indicado en el pie de página anterior explicando los algoritmos de clave asimétrica, ssh-keygen no permite la creación de claves  $\geq 1024$  a pesar de encontrar información técnica que indique que hay soporte de 2048 bits.

```
ssh-keygen -t dsa -b 2048
```



```
pedroamp@m1-pedroamp:~$ ssh-keygen -t dsa -b 2048
Invalid DSA key length: must be 1024 bits
pedroamp@m1-pedroamp:~$
```

Figura 4: Error de 2048 bits de DSA

<sup>4</sup>El comando *ssh-keygen por defecto crea claves RSA*, esto es invocándolo sin ningún argumento, pero lo específico para ver el funcionamiento de los argumentos





## 2.4. Proceso de copiado de clave asimétrica - proceso manual

En este proceso para evitar duplicidad uso la clave DSA.

Para el proceso manual, que en resumen viene a ser lo mismo que en el proceso automático se tiene que utilizar el comando scp que permita la copia del fichero `id_dsa.pub` o `id_rsa.pub`.

Para luego a continuación poner al final (append) del fichero `/.ssh/authorized_keys` la clave.

```
pedroamp@m1-pedroamp:~$ scp ~/.ssh/id_dsa.pub pedroamp@m2-pedroamp:~/.ssh
id_dsa.pub                                100% 610   554.4KB/s   00:00
pedroamp@m1-pedroamp:~$ ssh pedroamp@m2-pedroamp
Linux m2-pedroamp 5.10.0-12-amd64 #1 SMP Debian 5.10.103-1 (2022-03-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Mar 27 19:45:59 2022 from 192.168.122.47
pedroamp@m2-pedroamp:~$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
pedroamp@m2-pedroamp:~$ rm ~/.ssh/id_dsa.pub
pedroamp@m2-pedroamp:~$ _
```

Figura 7: Proceso de copiado de clave asimétrica DSA manual

## 2.5. Accediendo con una clave o otra

Para el acceso de una clave o otra debemos tener las claves copiadas a través del anterior proceso y por supuesto se desactiva la directiva de *PasswordAuthentication*.

```
ssh -i ~/.ssh/id_dsa pedroamp@m2 -pedroamp
```

### 3. Copiar archivos locales con equipos remotos.

#### 3.1. Uso de Tar para comprimir un directorio y mandarlo a un equipo remoto por SSH

Se utiliza cauces para poder permitir el envío del stream de bits que son leídos por cat en el equipo remoto y se queda comprimido.

```
pedroamp@m1-pedroamp:~$ mkdir copia_seguridad_directorio
pedroamp@m1-pedroamp:~$ mkdir copia_seguridad_directorio/subdirectorio1
pedroamp@m1-pedroamp:~$ mkdir copia_seguridad_directorio/subdirectorio2
pedroamp@m1-pedroamp:~$ touch copia_seguridad_directorio/subdirectorio1/fichero_importante1
pedroamp@m1-pedroamp:~$ touch copia_seguridad_directorio/subdirectorio1/fichero_importante2
pedroamp@m1-pedroamp:~$ touch copia_seguridad_directorio/subdirectorio2/fichero_importante3
pedroamp@m1-pedroamp:~$ touch copia_seguridad_directorio/subdirectorio2/fichero_importante4
pedroamp@m1-pedroamp:~$
```

Figura 8: Directorio y su composición creada

```
tar -czf - copia\_seguridad\_directorio/
| ssh pedroamp@m2-pedroamp 'cat > ~/copia_1.tgz
&& tar -xzf ~/copia_1.tgz
&& ls -la copia_seguridad_directorio'
```

```
pedroamp@m1-pedroamp:~$ tar -czf - copia_seguridad_directorio/ | ssh pedroamp@m2-pedroamp 'cat > ~/copia_1.tgz && tar -xzf ~/copia_1.tgz && ls -la copia_seguridad_directorio/'
total 16
drwxr-xr-x 4 pedroamp pedroamp 4096 Mar 28 11:20 .
drwxr-xr-x 4 pedroamp pedroamp 4096 Mar 28 11:28 ..
drwxr-xr-x 2 pedroamp pedroamp 4096 Mar 28 11:20 subdirectorio1
drwxr-xr-x 2 pedroamp pedroamp 4096 Mar 28 11:20 subdirectorio2
pedroamp@m1-pedroamp:~$
```

Figura 9: Creando comprimido y mandándandolo al servidor remoto además de descomprimirlo.

#### 3.2. Uso de SCP para enviar una copia de un directorio

Partiendo del directorio de la subsección anterior, creamos directamente un tar que lo enviamos por scp y lo comprobamos el estado con ssh.

```
tar -czf enviar.tgz copia\_seguridad\_directorio/
&& scp enviar.tgz pedroamp@m2-pedroamp:~/enviar.tgz
```

```
pedroamp@m1-pedroamp:~$ tar -czf enviar.tgz copia_seguridad_directorio/ && scp enviar.tgz pedroamp@m2-pedroamp:~/enviar.tgz
enviar.tgz
pedroamp@m1-pedroamp:~$ ssh pedroamp@m2-pedroamp 'ls -la |
> ^C
pedroamp@m1-pedroamp:~$ ssh pedroamp@m2-pedroamp 'ls -la | grep enviar.tgz'
-rw-r--r-- 1 pedroamp pedroamp 259 Mar 28 11:32 enviar.tgz
pedroamp@m1-pedroamp:~$
```

Figura 10: Creando comprimido y mandándandolo al servidor remoto además de descomprimirlo.

Tambien podemos copiar directamente el directorio.

```
scp -r copia\_seguridad\_directorio/ \
pedroamp@m2 -pedroamp:~/copia\_seguridad\_directorio/
```

```
pedroamp@m1-pedroamp:~$ scp -r copia_seguridad_directorio/ pedroamp@m2-pedroamp:/home/pedroamp/copia_seguridad_directorio/
fichero_importante3 100% 0 0.0KB/s 00:00
fichero_importante4 100% 0 0.0KB/s 00:00
fichero_importante1 100% 0 0.0KB/s 00:00
fichero_importante2 100% 0 0.0KB/s 00:00
pedroamp@m1-pedroamp:~$ ssh pedroamp@m2-pedroamp 'ls -la copia_seguridad_directorio/'
total 20
drwxr-xr-x 5 pedroamp pedroamp 4096 Mar 28 11:35 .
drwxr-xr-x 4 pedroamp pedroamp 4096 Mar 28 11:32 ..
drwxr-xr-x 4 pedroamp pedroamp 4096 Mar 28 11:35 copia_seguridad_directorio
drwxr-xr-x 2 pedroamp pedroamp 4096 Mar 28 11:20 subdirectorio1
drwxr-xr-x 2 pedroamp pedroamp 4096 Mar 28 11:20 subdirectorio2
pedroamp@m1-pedroamp:~$
```

Figura 11: Copia recursiva con SCP.

## 4. Rsync

### 4.1. Instalación de Rsync

Para la instalación de Rsync debemos poner lo siguiente.

```
apt-get install rsync
```

### 4.2. Realizando un backup full con Rsync de un árbol de directorios

Imaginemos que en una empresa se aloja por NFS un árbol de directorios crítico que en el caso de fallo de disco se produce una gran pérdida económica y de tiempo material o humano.

Entonces Rsync ofrece la copia recursiva de los archivos iniciales como una copia de seguridad completa.

```
rsync -avz -e ssh rsync/ pedroamp@m2 -pedroamp:~/rsync
```

-a implica todo lo siguiente:

1. -r recursive
2. -l links (preserva los enlaces simbólicos)
3. -p perms permisos
4. -t tiempo, conserva el tiempo de modificación creación etc...
5. -g preserva el grupo
6. -o preserva el owner o usuario dueño del archivo-

El resto de parámetros implica:

1. -z comprime los ficheros para usar menos ancho de banda pero más CPU y memoria.
2. -e ssh implica el uso del protocolo SSH como medio de transmisión, en el caso de no tener par de claves pediría una contraseña.

Aquí hemos copiado todo el árbol de directorios en el destino tal y como se muestra en el -v.

```

pedroamp@m1-pedroamp:~$ mkdir rsync
pedroamp@m1-pedroamp:~$ mkdir rsync/subfolder1
pedroamp@m1-pedroamp:~$ mkdir rsync/subfolder2
pedroamp@m1-pedroamp:~$ mkdir rsync/subfolder1/subsubfolder1
pedroamp@m1-pedroamp:~$ mkdir rsync/subfolder1/subsubfolder2
pedroamp@m1-pedroamp:~$ touch rsync/subfolder1/subsubfolder2/file.txt
pedroamp@m1-pedroamp:~$ mkdir rsync/subfolder1/subsubfolder1/another.c
pedroamp@m1-pedroamp:~$ touch rsync/subfolder2/keys.pem
pedroamp@m1-pedroamp:~$ tree rsync/
rsync/
├── subfolder1
│   ├── subsubfolder1
│   │   └── another.c
│   └── subsubfolder2
│       └── file.txt
└── subfolder2
    └── keys.pem

5 directories, 2 files
pedroamp@m1-pedroamp:~$ rsync -avz -e ssh rsync/ pedroamp@m2-pedroamp:~/
sending incremental file list
./
subfolder1/
subfolder1/subsubfolder1/
subfolder1/subsubfolder1/another.c/
subfolder1/subsubfolder2/
subfolder1/subsubfolder2/file.txt
subfolder2/
subfolder2/keys.pem

sent 372 bytes  received 81 bytes  906.00 bytes/sec
total size is 0  speedup is 0.00
pedroamp@m1-pedroamp:~$

```

Figura 12: Árbol de directorios y primera copia con Rsync full.

### 4.3. Modo Mirror

Es un modo en el que rsync sincroniza los ficheros modificados en el origen con el destino sin importar lo que haya en el destino y copia las diferencias del origen y las borra en el destino.

En el caso que voy a hacer es modificar un fichero poniendole un texto y borrar file.txt para sustituirlo por final.txt.

Luego el comando borrará el fichero file.txt en el destino.

```
rsync -avzp --delete -e ssh rsync/ pedroamp@m2-pedroamp:~/rsync
```

1. -p = -progress, muestra el progreso
2. -delete = borra los ficheros en el destino

```

pedroamp@m1-pedroamp:~$ rsync -avzp --delete -e ssh rsync/ pedroamp@m2-pedroamp:~/rsync
sending incremental file list
created directory /home/pedroamp/rsync
./
subfolder1/
subfolder1/subsubfolder1/
subfolder1/subsubfolder1/another.c/
subfolder1/subsubfolder2/
subfolder1/subsubfolder2/final.txt
subfolder2/
subfolder2/keys.pem

sent 376 bytes  received 124 bytes  333.33 bytes/sec
total size is 0  speedup is 0.00
pedroamp@m1-pedroamp:~$ rm
.bash_history          copia_seguridad_directorio/ .profile
.bash_logout           enviar.tgz               rsync/
.bashrc                exit                    .ssh/
pedroamp@m1-pedroamp:~$ rm rsync/subfolder1/subsubfolder2/final.txt
pedroamp@m1-pedroamp:~$ touch rsync/subfolder1/subsubfolder2/finalfinal.txt
pedroamp@m1-pedroamp:~$ rsync -avzp --delete -e ssh rsync/ pedroamp@m2-pedroamp:~/rsync
sending incremental file list
deleting subfolder1/subsubfolder2/final.txt
subfolder1/subsubfolder2/
subfolder1/subsubfolder2/finalfinal.txt

sent 327 bytes  received 81 bytes  816.00 bytes/sec
total size is 0  speedup is 0.00
pedroamp@m1-pedroamp:~$ _

```

Figura 13: Modo Mirror

#### 4.4. Modo Copia incremental

Rsync admite de modo nativo la copia incremental de ficheros para tener siempre un respaldo. El modo incremental es que copia todos los ficheros modificados o creados desde la última copia sea full o sea incremental.

El archivo modificado se guarda en un fichero .bakfile en el caso de que te hayas equivocado con alguna modificación este se guarda.

```

rsync -abvzp -e ssh --delete
--backup-dir='~/backup/backup_$(date +%d%m%Y%H%M)'
--suffix='.bak'
~/rsync pedroamp@m2-pedroamp:~/rsync

```

1. -b --backup -> modo backup necesario para los de abajo
2. --suffix='.bakfile' -> Sufijo de los ficheros de backup
3. --backup-dir='backup/backup\_\$(date +%d%m%Y%H%M)' -> Directorio de la copia incremental

He creado muchos ficheros para el objetivo de la práctica. Pero nos vamos a centrar en como detecta Rsync las diferencias para ponerlo en su modelo incremental.

```

pedroamp@m1-pedroamp:~$ rsync -abzvp -e ssh --delete --backup-dir='~/backup/backup_$(date +%d%m%Y%H%M)' --suffix='.bakfile' ~/rsync
pedroamp@m2-pedroamp:~/rsync
sending incremental file list
created directory /home/pedroamp/rsync
rsync/
rsync/subfolder1/
rsync/subfolder1/file1.txt
rsync/subfolder1/file10.txt
rsync/subfolder1/file2.txt
rsync/subfolder1/file3.txt
rsync/subfolder1/file4.txt
rsync/subfolder1/file5.txt
rsync/subfolder1/file6.txt
rsync/subfolder1/file7.txt
rsync/subfolder1/file8.txt
rsync/subfolder1/file9.txt
rsync/subfolder1/hola.txt
rsync/subfolder1/hola1.txt
rsync/subfolder1/hola1111.txt
rsync/subfolder1/hola1111222.txt
rsync/subfolder1/subsubfolder1/
rsync/subfolder1/subsubfolder1/another.c/
rsync/subfolder1/subsubfolder2/
rsync/subfolder1/subsubfolder2/finalfinal.txt
rsync/subfolder2/
rsync/subfolder2/keys.pem

sent 1,261 bytes  received 391 bytes  1,101.33 bytes/sec
total size is 5  speedup is 0.00
pedroamp@m1-pedroamp:~$ _

```

Figura 14: Realización de la copia inicial - Full Backup

```

pedroamp@m1-pedroamp:~$ echo "Primera Modificacion" > rsync/subfolder1/hola1.txt
pedroamp@m1-pedroamp:~$ rsync -abzvp -e ssh --delete --backup-dir='~/backup/backup_$(date +%d%m%Y%H%M)' --suffix='.bakfile' ~/rsync
pedroamp@m2-pedroamp:~/rsync
sending incremental file list
rsync/subfolder1/hola1.txt

sent 723 bytes  received 45 bytes  1,536.00 bytes/sec
total size is 26  speedup is 0.03
pedroamp@m1-pedroamp:~$ ssh pedroamp@m2-pedroamp 'tree backup'
backup
├── backup_280320221528
│   └── rsync
│       └── subfolder1
│           └── hola1.txt.bakfile
└── 3 directories, 1 file

```

Figura 15: Modificando un archivo, haciendo copia incremental y visualizando directorio de backups

Ahora hacemos una segunda modificación para poder ver si se guarda en el backup la primera modificación.

```

pedroamp@m1-pedroamp:~$ echo "Segunda Modificacion" >> rsync/subfolder1/hola1.txt
pedroamp@m1-pedroamp:~$ rsync -abzvp -e ssh --delete --backup-dir="/backup/backup_$(date +%d%m%Y%H%M)" --suffix='.bakfile' ~/rsync
pedroamp@m2-pedroamp:~/rsync
Sending incremental file list
rsync/subfolder1/hola1.txt

sent 735 bytes  received 51 bytes  1,572.00 bytes/sec
total size is 47  speedup is 0.06
pedroamp@m1-pedroamp:~$ ssh pedroamp@m2-pedroamp 'tree backup'
backup
├── backup_280320221528
│   └── rsync
│       └── subfolder1
│           └── hola1.txt.bakfile
└── backup_280320221531
    └── rsync
        └── subfolder1
            └── hola1.txt.bakfile

6 directories, 2 files
pedroamp@m1-pedroamp:~$ ssh pedroamp@m2-pedroamp 'cat backup/backup_280320221531/rsync/subfolder1/hola1.txt.bakfile'
Primera Modificacion
pedroamp@m1-pedroamp:~$ _

```

Figura 16: Segunda modificación del archivo y haciendo copia incremental

Podemos ver que las copias se realizan de la versión anterior y se guardan con sus respectivas fechas de realización de backup.



## 5. Crontab

Crontab es la tabla sobre la que el demonio cron lee las tareas que tiene que lanzar en el tiempo determinado por el usuario. Los ficheros cron se instalan en la ruta `/var/spool/cron/crontabs/$USER`. **No deberían ser modificados directamente, deben usarse su comando requerido**

```
pedroamp@m1-pedroamp:~$ sudo ls /var/spool/cron/crontabs/
[sudo] password for pedroamp:
pedroamp
pedroamp@m1-pedroamp:~$ sudo cat /var/spool/cron/crontabs/pedroamp
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.ahJipE/crontab installed on Mon Mar 28 21:57:27 2022)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * * bash ~/backups.sh
pedroamp@m1-pedroamp:~$ _
```

Figura 17: Crontab ruta de localización de los ficheros de edición

Existen muchas opciones de como utilizar crontab, pero me voy a centrar en hacer copias diarias de las diferencias.

Cada asterisco corresponde con lo siguiente: Min Hour Day Mon Weekday

En mi caso programaría tareas diarias a las 7 de la Mañana cada día. Para editar nuestro crontab usamos el siguiente comando.

```
crontab -e
```

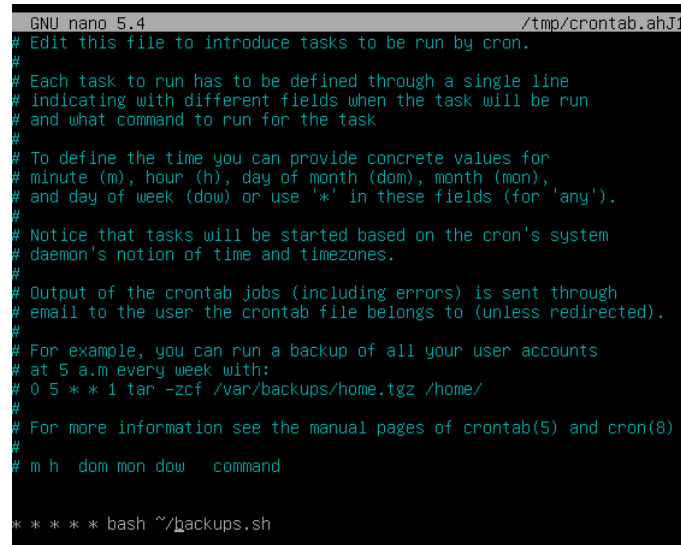
```
pedroamp@m1-pedroamp:~$ crontab -e
no crontab for pedroamp - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.tiny

Choose 1-2 [1]: 1_
```

Figura 18: Programación de Crontab - Modo de edición

*Nota:* la entrada debería de redirigirse hacia un fichero de salida para verificar el estado correcto.



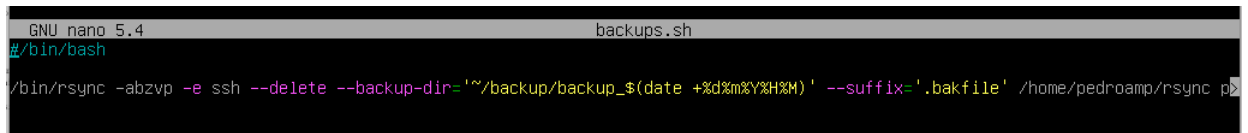
```
GNU nano 5.4 /tmp/crontab.ahJ1
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * * bash ~/backups.sh
```

Figura 19: Programación de Crontab - Entrada programada

Para ver que efectivamente se ha instalado en el sistema tiene que salir. *crontab: installing new crontab*

*Nota:* Muchas veces los cron no cogen las variables de entorno de los PATH de los /bin por eso le he puesto el /bin/rsync porque si no bash no puede localizar el binario al buscarlo en el directorio mismo.

*Nota:* Cron necesita tener los permisos de ejecución puestos. O de otra manera el script no hace ninguna acción.



```
GNU nano 5.4 backups.sh
#!/bin/bash
/bin/rsync -abzvp -e ssh --delete --backup-dir='~/backup/backup_$(date +%d%m%Y%H%M)' --suffix='.bakfile' /home/pedroamp/rsync p>
```

Figura 20: Programación de Crontab - Script

En el log aparece como que *cron usa nuestro usuario (o en otros casos el de root - para las tareas programadas de este) para la ejecución de tareas.*

```
The job identifier is 432.
Mar 28 21:42:46 m1-pedroamp crontab[598]: (pedroamp) BEGIN EDIT (pedroamp)
Mar 28 21:42:52 m1-pedroamp crontab[598]: (pedroamp) REPLACE (pedroamp)
Mar 28 21:42:52 m1-pedroamp crontab[598]: (pedroamp) END EDIT (pedroamp)
Mar 28 21:43:01 m1-pedroamp cron[489]: (pedroamp) RELOAD (crontabs/pedroamp)
Mar 28 21:43:01 m1-pedroamp CRON[606]: pam_unix(cron:session): session opened for user pedroamp(uid=1000) by (uid=0)
Mar 28 21:43:01 m1-pedroamp CRON[607]: (pedroamp) CMD (rsync -avzp -e ssh --delete --backup-dir=~/.backup/backup_$(date +%Y%m%d%H%M%S) /home/pedroamp /home/pedroamp)
Mar 28 21:43:01 m1-pedroamp CRON[606]: (CRON) info (No MTA installed, discarding output)
Mar 28 21:43:01 m1-pedroamp CRON[606]: pam_unix(cron:session): session closed for user pedroamp
Mar 28 21:43:03 m1-pedroamp sudo[605]: pedroamp : TTY=ttty1 ; PWD=/home/pedroamp ; USER=root ; COMMAND=/usr/bin/journalctl -xe
Mar 28 21:43:03 m1-pedroamp sudo[605]: pam_unix(sudo:session): session opened for user root(uid=0) by pedroamp(uid=1000)
lines 1785-1831/1831 (END)
```

Figura 21: Log de ejecución

## 6. Bibliografía

<sup>5</sup> <https://manpages.debian.org/stretch/openssh-client/ssh-copy-id.1.en.html>

<sup>6</sup> <https://www.ibm.com/docs/en/zos/2.2.0?topic=certificates-size-considerations-public-private-keys>

<sup>7</sup> [https://wiki.archlinux.org/title/SSH\\_keys\\_\(Espa%C3%B1ol\)](https://wiki.archlinux.org/title/SSH_keys_(Espa%C3%B1ol))

---

<sup>5</sup> Manpage de ssh-copy-id

<sup>6</sup> Información técnica de las claves asimétricas y recomendaciones.

<sup>7</sup> SSH keys y procesos de copiado