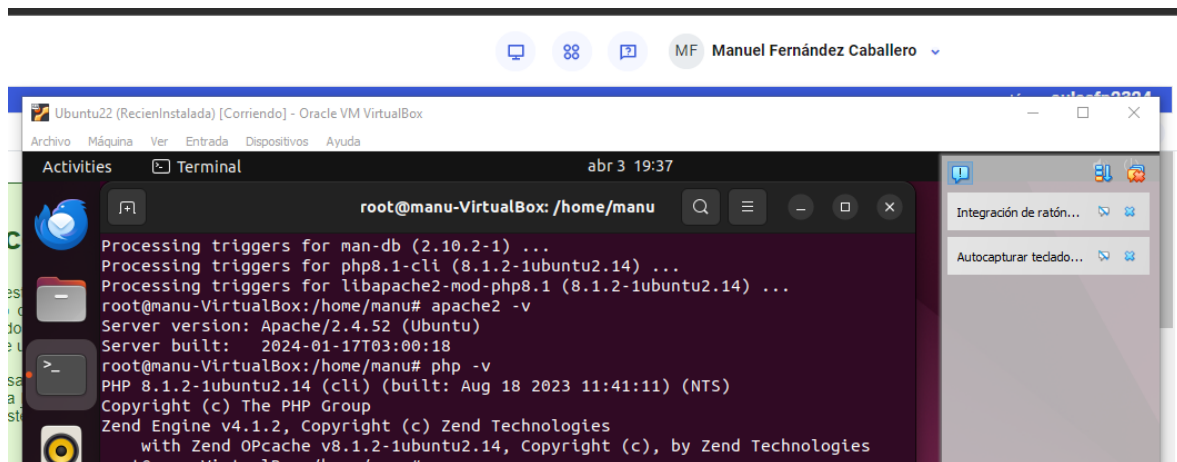


¿QUÉ TE PEDIMOS QUE HAGAS?

1. Indica cada uno de los pasos que deberías de dar para proceder a la instalación de **phpDocumentor**, suponiendo que vas a partir de una máquina en la que tienes instalado la distribución **Debian / Ubuntu** actual, y en la que ya están instalados y correctamente configurados **apache** y **php**.

Primero comprobamos que **apache** y **php** están instalados y correctamente configurados con **apache2 -v** y **php -v**



Ahora comenzamos con la guía de la instalación que viene en la unidad 6 de de **phpDocumentor**. Lo primero será instalar el paquete **php-pear**, que es un entorno de desarrollo y sistema de distribución para componentes de código PHP. Para instalarlo podemos hacerlo mediante el gestor de paquetes **apt**:

```
# apt-get install php-pear
```

```
root@manu-VirtualBox:/home/manu# apt-get install php-pear
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  php-xml php8.1-xml
The following NEW packages will be installed:
  php-pear php-xml php8.1-xml
0 upgraded, 3 newly installed, 0 to remove and 41 not upgraded.
3 of archives.
on, 2.630 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Por defecto **Apache** busca en el directorio **/var/www/html** para el contenido web, con lo cual antes de la instalación de **phpDocumentor** deberíamos decirle a **pear** que es aquí donde queremos que **phpDocumentor** debe trabajar, se puede configurar este u otro directorio, siempre y cuando sea accesible por el servidor web:

```
# pear config-set data_dir /var/www/html
```

ahora podemos instalar **phpDocumentor** y también sus dependencias:

```
root@manu-VirtualBox:/home/manu# pear config-set data_dir /var/www/html
config-set succeeded
root@manu-VirtualBox:/home/manu#
```

```
# pear install --alldeps PhpDocumentor
```

```
config-set succeeded
root@manu-VirtualBox:/home/manu# pear install --alldeps PhpDocumentor
WARNING: "pear/PhpDocumentor" is deprecated in favor of "channel://pear.phpdoc.org/phpdocumentor"
WARNING: channel "pear.php.net" has updated its protocols, use "pear channel-update pear.php.net" to update
WARNING: "pear/XML_Parser" is deprecated in favor of "pear/XML_Parser2"
downloading PhpDocumentor-1.4.4.tgz ...
Starting to download PhpDocumentor-1.4.4.tgz (1,554,970 bytes)
.....
.....done: 1,554,970 bytes
downloading XML_Beautifier-1.2.2.tgz ...
Starting to download XML_Beautifier-1.2.2.tgz (14,742 bytes)
...done: 14,742 bytes
downloading XML_Parser-1.3.8.tgz ...
Starting to download XML_Parser-1.3.8.tgz (13,534 bytes)
...done: 13,534 bytes
install ok: channel://pear.php.net/PhpDocumentor-1.4.4
install ok: channel://pear.php.net/XML_Parser-1.3.8
install ok: channel://pear.php.net/XML_Beautifier-1.2.2
root@manu-VirtualBox:/home/manu#
```

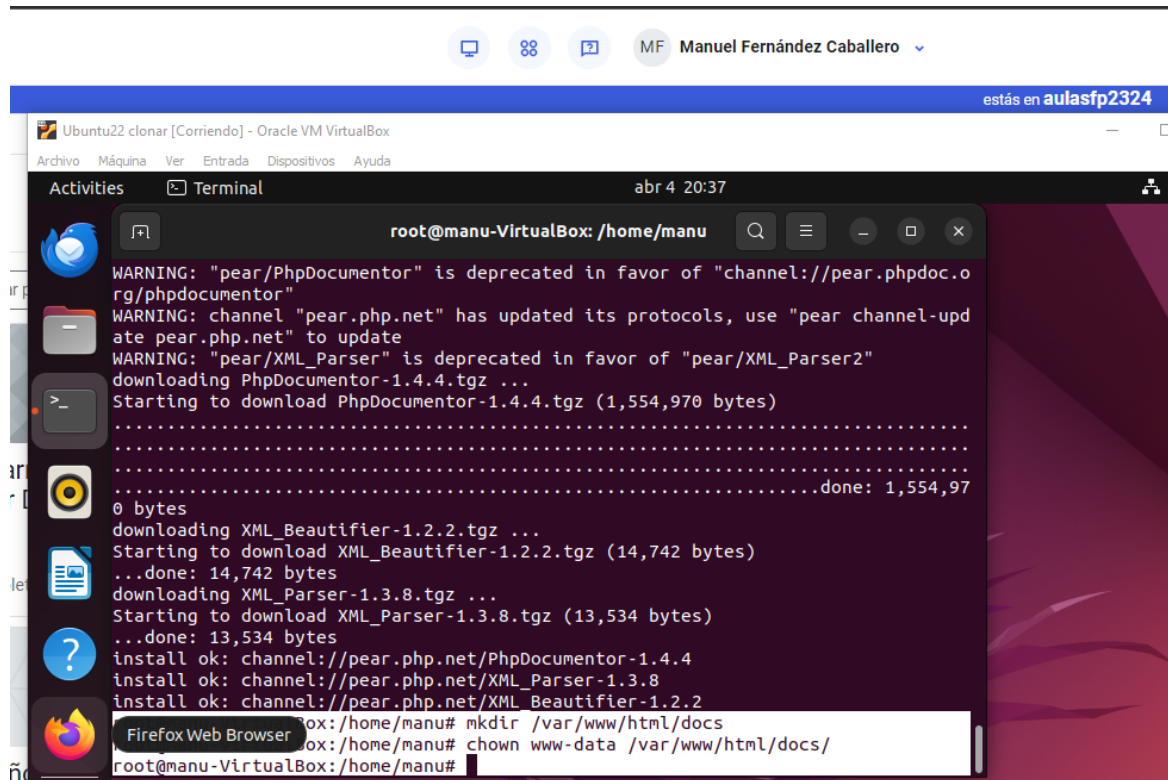
o también directamente podemos descargar el paquete mediante:

```
# wget sourceforge.net/projects/phpdocu/files/PhpDoc/phpDocumentor-1.4.3/PhpDocumentor-1.4.3.tgz
```

y luego descomprimirlo.

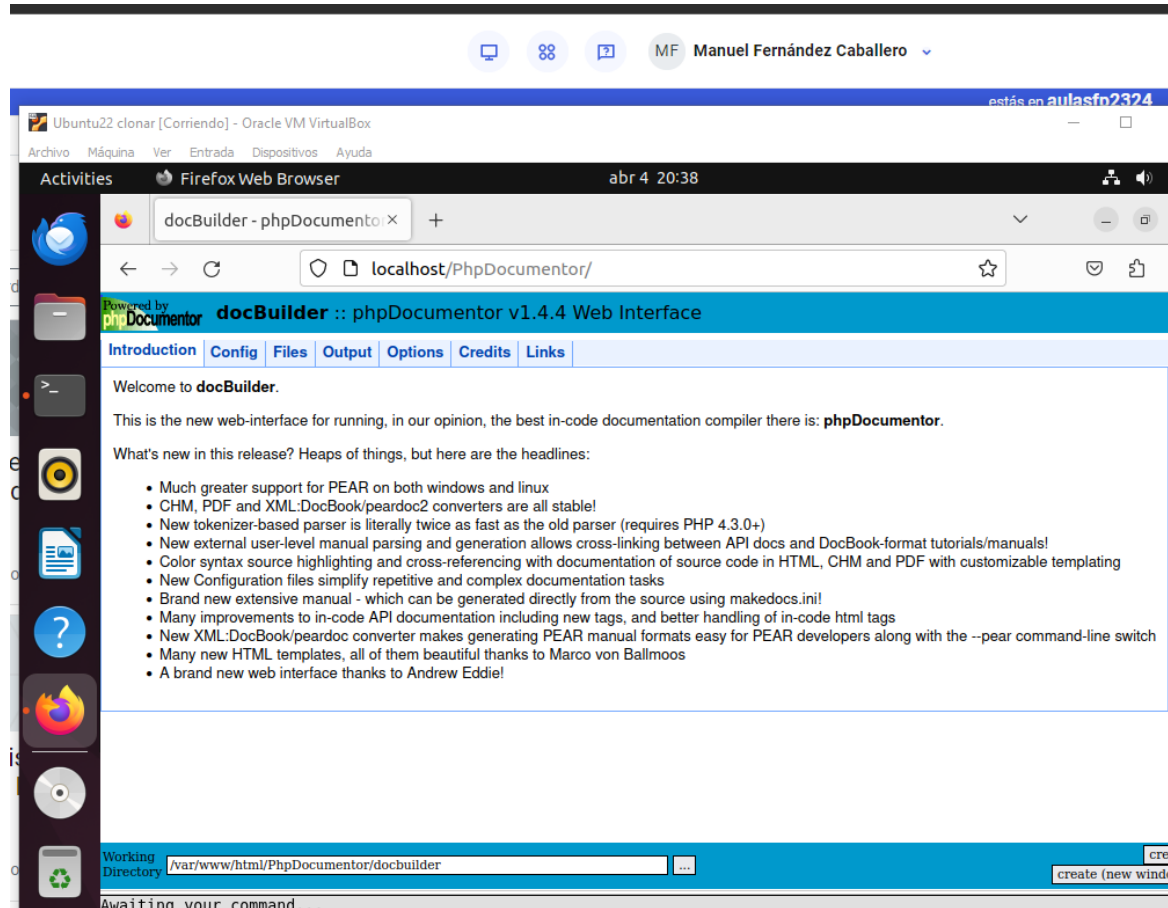
Una vez terminado el proceso de instalación necesitamos crear un directorio de salida para **phpDocumentor**, y cambiar el propietario de dicho directorio a **www-data**, de forma que pueda trabajar en esta carpeta sin ninguna limitación; por ejemplo podemos realizar lo siguiente:

```
# mkdir /var/www/html/docs
# chown www-data /var/www/html/docs/
```



```
root@manu-VirtualBox: /home/manu
WARNING: "pear/PhpDocumentor" is deprecated in favor of "channel://pear.phpdoc.org/phpdocumentor"
WARNING: channel "pear.php.net" has updated its protocols, use "pear channel-update pear.php.net" to update
WARNING: "pear/XML_Parser" is deprecated in favor of "pear/XML_Parser2"
downloading PhpDocumentor-1.4.4.tgz ...
Starting to download PhpDocumentor-1.4.4.tgz (1,554,970 bytes)
.....done: 1,554,970 bytes
downloading XML_Beautifier-1.2.2.tgz ...
Starting to download XML_Beautifier-1.2.2.tgz (14,742 bytes)
...done: 14,742 bytes
downloading XML_Parser-1.3.8.tgz ...
Starting to download XML_Parser-1.3.8.tgz (13,534 bytes)
...done: 13,534 bytes
install ok: channel://pear.php.net/PhpDocumentor-1.4.4
install ok: channel://pear.php.net/XML_Parser-1.3.8
install ok: channel://pear.php.net/XML_Beautifier-1.2.2
root@manu-VirtualBox: /home/manu# mkdir /var/www/html/docs
root@manu-VirtualBox: /home/manu# chown www-data /var/www/html/docs/
```

Si desde un navegador tecleamos **http://localhost/PhpDocumentor/** deberíamos comprobar que la instalación ha sido correcta y tenemos phpDocumentor listo para funcionar.



2. *Explica en qué consisten las plantillas de código en el caso de Javadoc y cada uno de sus componentes.*

Las plantillas de Javadoc son sugerencias de código, abreviaciones asociadas a **palabras clave** que facilitan **la generación de documentación** para el código Java.

Estas plantillas se definen en el menú "Preferences > Java > Editor > Templates" y están diseñadas para ahorrar tiempo al escribir documentación.

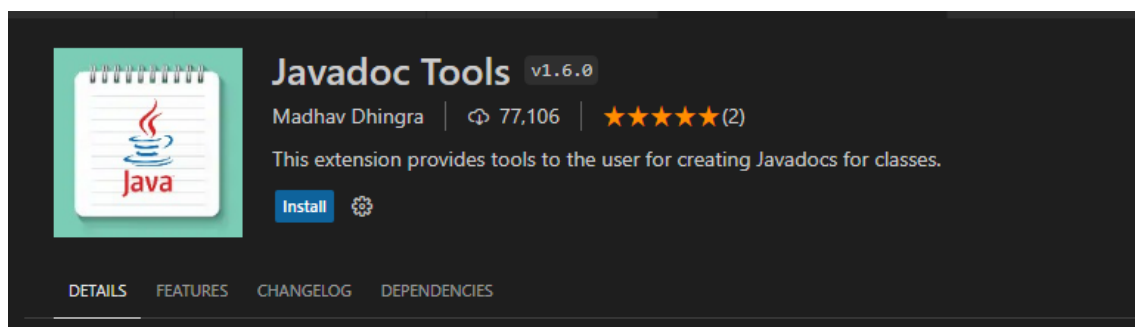
Cada plantilla tiene un nombre, una descripción y un contexto específico en función del lenguaje (como código Java o Javadoc). Además, está compuesta por un patrón de código que puede contener texto fijo y variables predefinidas.

Por ejemplo, algunas variables comunes son:

- **`${cursor}`** (la posición donde se colocará el cursor después de desplegar la plantilla)
- **`${enclosing_type}`** (el tipo de clase en la que se encuentra)
- **`${year}`** (el año actual)
- **`${enclosing_method}`**: nombre del método en el que nos encontramos.
- **`${year}`**: año en curso.
- **`${time}`**: hora en curso.

Estas plantillas se muestran como sugerencias cuando comienzas a escribir su nombre y presionas CTRL + espacio en el editor de código. También es posible crear y personalizar tus propias plantillas según tus necesidades específicas.

Si prefieres visual studio code, esta extensión te facilitará la generación de documentos Javadoc así como la generación de plantillas.



3. *Dispones de una máquina que cuenta con el sistema operativo Debian / Ubuntu recientemente actualizado, en la que está el entorno de red configurado y, además, dispones de conexión a Internet y estás trabajando con la cuenta del usuario root . Indica cada uno de los pasos y comandos implicados en ellos para conseguir hacer lo siguiente:*

1. Suponiendo que el sistema ya tiene instalado las siguientes librerías de las que Git depende: curl, zlib, openssl, expat, y libiconv, pasos a realizar la compilación e instalación de Git considerando que ya disponemos del paquete git-1.7.6.tar.bz2

Comprobamos que esta todo instalado:

Curl --version

```
root@manu-VirtualBox:/home/manu# curl --version
curl 7.81.0 (x86_64-pc-linux-gnu) libcurl/7.81.0 OpenSSL/3.0.2 zlib/1.2.11 brotli/1.0.9 zstd/1.4.8 libidn2/2.3.2 libpsl/0.21.0 (+libidn2/2.3.2) libssh/0.9.6/openssl/zlib nghttp2/1.43.0 librtmp/2.3 OpenLDAP/2.5.17
Release-Date: 2022-01-05
Protocols: dict file ftp ftps gopher gophers http https imap imaps ldap ldaps mqtt pop3 pop3s rtmp rtsp scp sftp smb smbs smtp smtps telnet tftp
Features: alt-svc AsynchDNS brotli GSS-API HSTS HTTP2 HTTPS-proxy IDN IPv6 Kerberos Largefile libz NTLM NTLM_WB PSL SPNEGO SSL TLS-SRP UnixSockets zstd
```

dpkg -l | grep zlib1g-dev

```
root@manu-VirtualBox:/home/manu# dpkg -l | grep zlib1g-dev
ii  zlib1g-dev:amd64 1:1.2.11.dfsg-2ubuntu9.2
    amd64 compression library - development
```

openssl version

```
root@manu-VirtualBox:/home/manu# openssl version
OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)
```

dpkg -l | grep libiconv-hook-dev

```
root@manu-VirtualBox:/home/manu# dpkg -l | grep libiconv-hook-dev
ii  libiconv-hook-dev 0.0.20021209-12
    amd64 header files of libiconv-hook
```

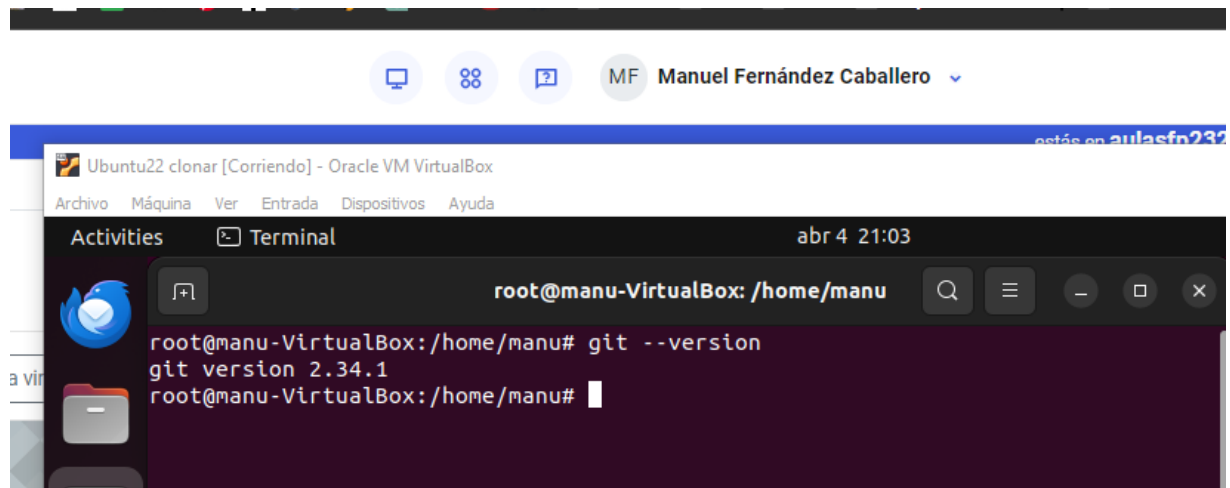
Ahora que tenemos las librerías instalamos git, pero no la versión mencionada que ya no existe, primero actualizamos los paquetes con:

sudo apt update

sudo apt upgrade

y después instalamos git con **sudo apt install git**

comprobamos la versión con **git --version**



2. *Cómo obtener Git a través del propio Git para futuras actualizaciones, de manera que descargaría automáticamente el código fuente desde su repositorio.*

Si quisiéramos obtener git a través de git podemos clonar el repositorio:

git clone <https://github.com/git/git>

```
root@manu-VirtualBox:~# git clone https://github.com/git/git
Cloning into 'git'...
remote: Enumerating objects: 363918, done.
remote: Counting objects: 100% (497/497), done.
remote: Compressing objects: 100% (262/262), done.
remote: Total 363918 (delta 316), reused 350 (delta 235), pack-reused 363421
Receiving objects: 100% (363918/363918), 235.26 MiB | 6.17 MiB/s, done.
Resolving deltas: 38% (104029/273760)
```

nos cambiamos al directorio creado **con cd git** y luego compilamos e instalamos git desde la fuente con los siguientes comandos:

make prefix=/usr/local all

sudo make prefix=/usr/local install

(este comando lleva un tiempo para que termine...)

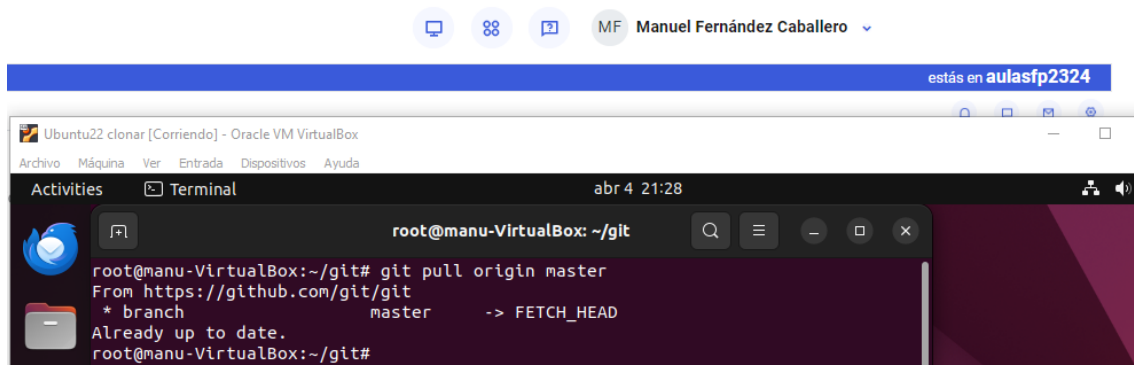
```
Preparing to unpack .../libcurl4-gnutls-dev_7.81.0-1ubuntu1.16_amd64.deb ...
Unpacking libcurl4-gnutls-dev:amd64 (7.81.0-1ubuntu1.16) ...
Setting up libcurl4-gnutls-dev:amd64 (7.81.0-1ubuntu1.16) ...
Processing triggers for man-db (2.10.2-1) ...
root@manu-VirtualBox:~/git# make prefix=/usr/local all
sudo make prefix=/usr/local install
* new build flags
CC oss-fuzz/dummy-cmd-main.o
CC oss-fuzz/fuzz-commit-graph.o
CC oss-fuzz/fuzz-config.o
CC oss-fuzz/fuzz-date.o
CC oss-fuzz/fuzz-pack-headers.o
Help CC oss-fuzz/fuzz-pack-idx.o
CC daemon.o
CC common-main.o
CC abspath.o
CC add-interactive.o
CC add-patch.o
```

Para permitir futuras actualizaciones desde el repositorio oficial de Git, podemos agregar el repositorio remoto origin al repositorio clonado con el siguiente comando:

git remote add origin <https://github.com/git/git>

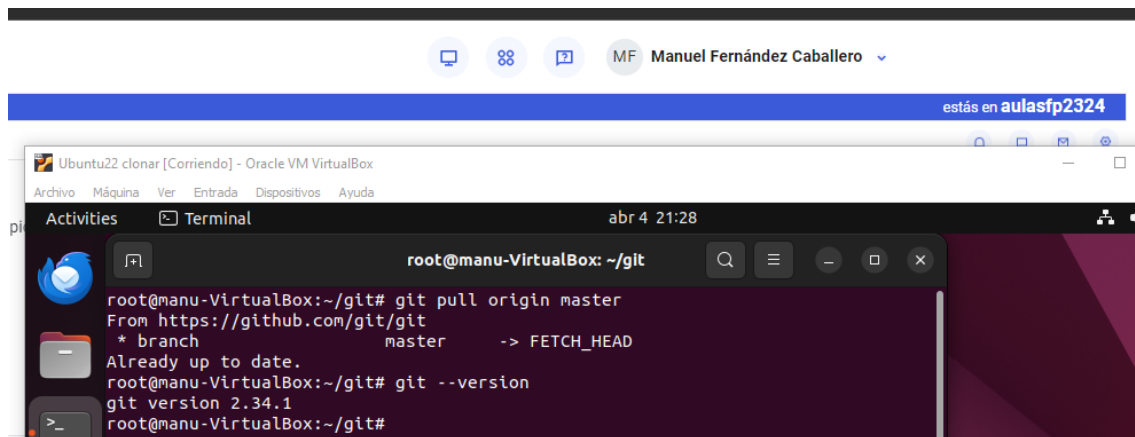
Y para actualizar tu instalación de Git en el futuro:

git pull origin master



3. Comprobar la versión que se ha instalado de Git.

Lo comprobamos con **git --version**



4. *Establecer el nombre de usuario y dirección de correo electrónico en la configuración de Git.*

Para establecer el nombre de usuario:

Git config --global user.name "manu"

```
root@manu-VirtualBox:~/git# git config --global user.name "manu"
```

Para establecer el correo:

git config --global user.email "manu@gmail.com"

Para comprobarlo:

Git config --list

```
root@manu-VirtualBox:~/git# git config --global user.email "manu@gmail.com"
error: key does not contain a section: --global
root@manu-VirtualBox:~/git# git config --global user.email "manu@gmail.com"
root@manu-VirtualBox:~/git# git congif --list
git: 'congif' is not a git command. See 'git --help'.

The most similar command is
  config
root@manu-VirtualBox:~/git# git config --list
user.name="manu"
user.email=manu@gmail.com
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=https://github.com/git/git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.master.remote=origin
branch.master.merge=refs/heads/master
root@manu-VirtualBox:~/git#
```

5. *Cambiar el editor de texto que trae por defecto Git al editor emacs.*

Para cambiar el editor de texto usamos:

Git config --global core.editor Emacs

Si abrimos el archive de configuración de git, Emacs estará como editor predeterminado

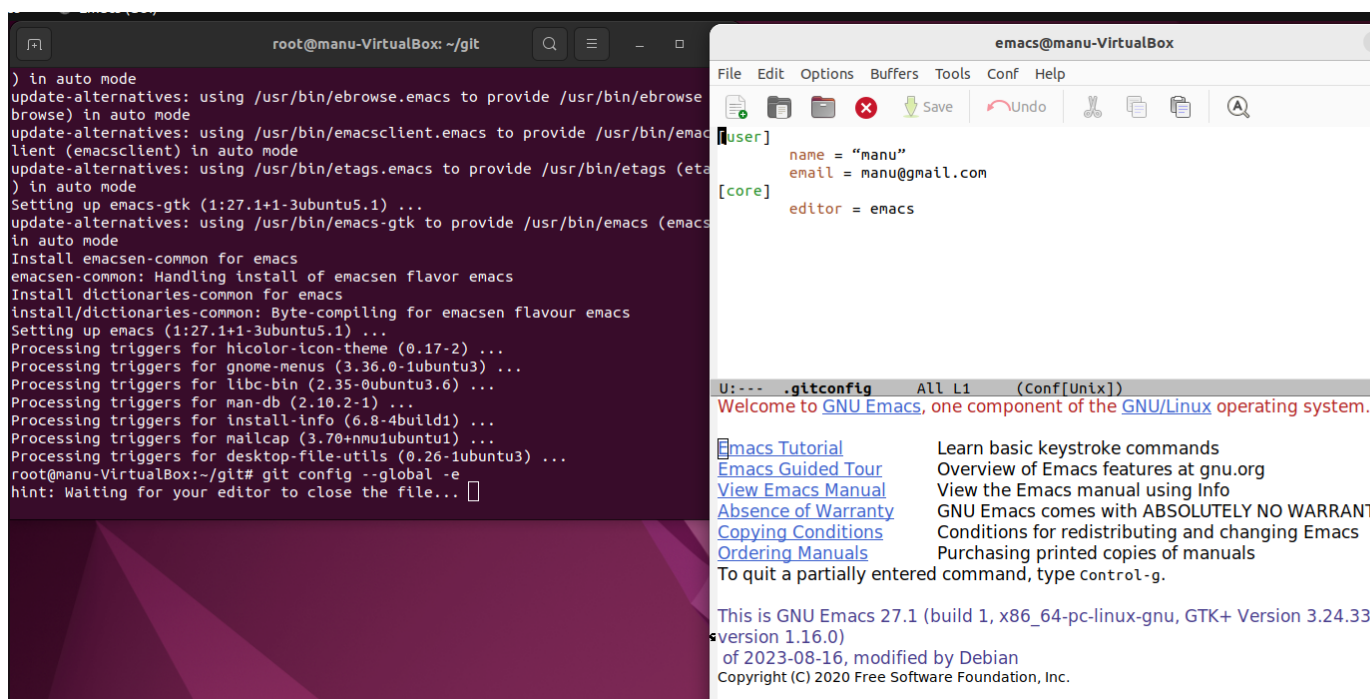
git config --global -e

```
root@manu-VirtualBox:~/git# git config --global core.editor emacs
root@manu-VirtualBox:~/git# git config --global -e
hint: Waiting for your editor to close the file... error: cannot run emacs: No such file or directory
error: unable to start editor 'emacs'
```

Tenemos que instalar el editor primero con

sudo apt install Emacs

Y ahora al ejecutar **git config --global -e** se abre con el editor Emacs que acabamos de instalar:



6. Dentro de la carpeta `/var/cache/git/` crear una carpeta para un nuevo proyecto denominado `tarea_DAW06` e iniciar un repositorio el nuevo proyecto.

Git en mi sistema esta localilazdo en otra ruta, reamos la carpeta solicitada con el comando

```
root@manu-VirtualBox:/var/cache# which git
/usr/bin/git
root@manu-VirtualBox:/var/cache#
```

```
sudo mkdir /var/cache/tarea_DAW06
```

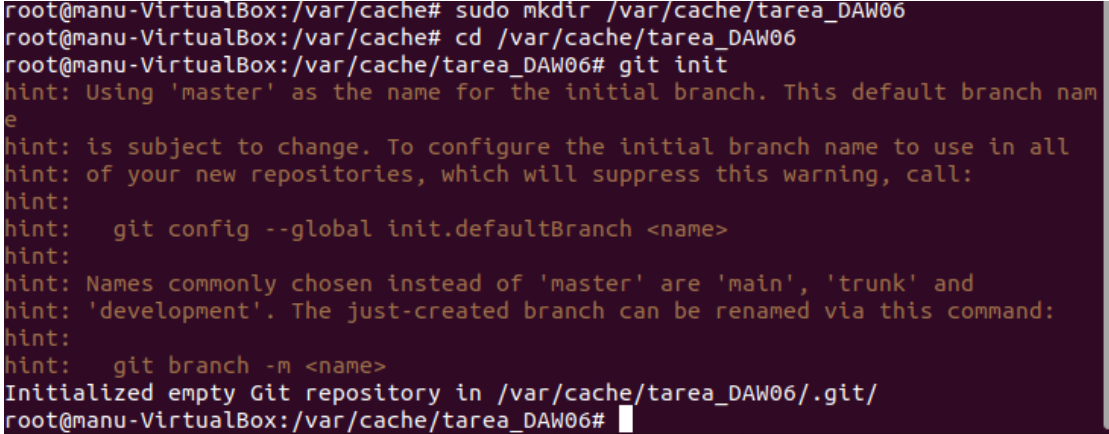
cambiamos de directorio a la carpeta

```
cd /var/cache/tarea_DAW06
```

e iniciamos un repositorio

```
sudo git init
```

ahora añadimos un par de archivos de



```
root@manu-VirtualBox:/var/cache# sudo mkdir /var/cache/tarea_DAW06
root@manu-VirtualBox:/var/cache# cd /var/cache/tarea_DAW06
root@manu-VirtualBox:/var/cache/tarea_DAW06# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /var/cache/tarea_DAW06/.git/
root@manu-VirtualBox:/var/cache/tarea_DAW06#
```

Con esto ya hemos inicializado un nuevo repositorio Git en el directorio tarea_DAW06 dentro de /var/cache/.

NOTA IMPORTANTE

Para el apartado 3 es necesario entregar las capturas de pantalla de los principales pasos realizados, explicando el proceso seguido en cada uno de ellos. Las capturas de pantalla realizadas deben tener como fondo de pantalla la plataforma con tu usuario mostrando claramente la foto de tu perfil. Aquellos apartados/subapartados que no cumplan esta condición no serán corregidos.