

TAREA DAW06

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.

- Instalamos el paquete php-pear (entorno de desarrollo y sistema de distribución de componentes de código PHP) mediante `apt install php-pear`
- Antes de la instalación de phpDocumentor, debemos indicarle a pear el directorio en el que queremos que trabaje phpDocumentor mediante `pear config-set data_dir /var/www/html`
- Instalamos phpDocumentor y sus dependencias mediante `pear install -alldeps PhpDocumentor`
- Una vez instalado creamos el directorio de salida de phpDocumentor y cambiamos el propietario del directorio a www-data para que podamos trabajar en esta carpeta sin ninguna limitación; mediante `mkdir /var/www/html/docs` y `chown www-data /var/www/html/docs`
- Comprobamos en el navegador la correcta instalación accediendo a <http://localhost/PhpDocumentor/>

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 asociadas a palabras clave, que se encuentran definidas en Preferences > Java > Editor > Templates. Existen plantillas predefinidas aunque también podemos crear nuestras propias plantillas.

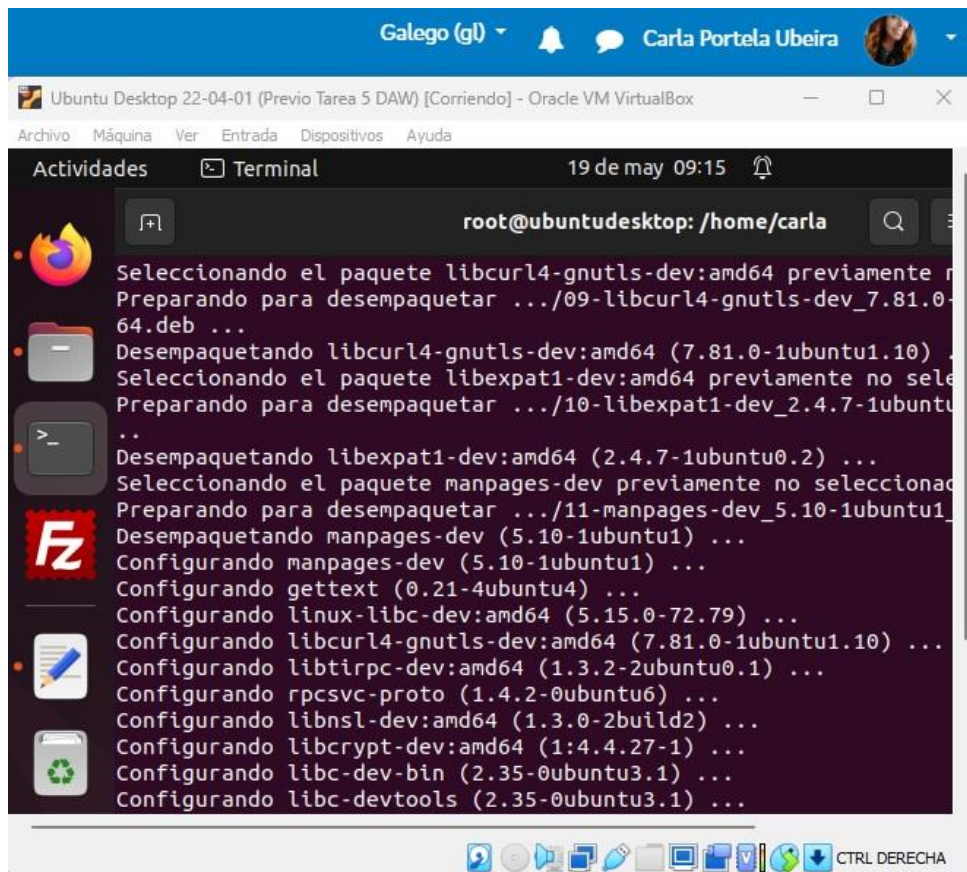
Se componen de:

- Nombre
- Descripción
- Contexto (en función del lenguaje)
- Pattern: es el código de la plantilla (texto fijo o variables predefinidas)

Estas plantillas se mostrarán como sugerencias en el código tras comenzar a escribir su nombre y pulsar CTRL+ espacio.

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:

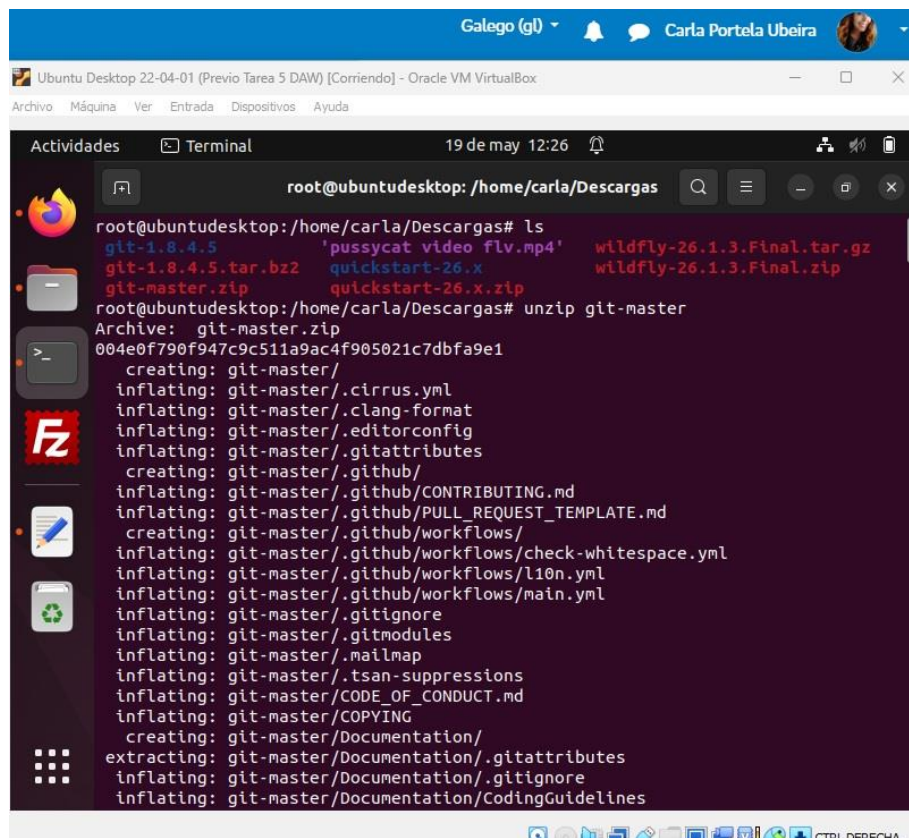
- Suponiendo que el sistema ya tiene instalado las siguientes librerías de las que Git depende: curl, zlib, openssl, expat, y libiconv, pasos a realizar la compilación e instalación de Git considerando que ya disponemos del paquete git-1.7.6.tar.bz2
Para instalar las librerías de las que depende Git ejecutamos en la consola de comandos `apt-get install libcurl4-gnutls-dev libexpat1-dev gettext \ libz-dev libssl-dev`



```
root@ubuntudesktop: /home/carla

Seleccionando el paquete libcurl4-gnutls-dev:amd64 previamente no seleccionado
Preparando para desempaquetar .../09-libcurl4-gnutls-dev_7.81.0-1ubuntu1.0.
64.deb ...
Desempaquetando libcurl4-gnutls-dev:amd64 (7.81.0-1ubuntu1.0) ...
Seleccionando el paquete libxpat1-dev:amd64 previamente no seleccionado
Preparando para desempaquetar .../10-libxpat1-dev_2.4.7-1ubuntu1.0.
..
Desempaquetando libxpat1-dev:amd64 (2.4.7-1ubuntu1.0) ...
Seleccionando el paquete manpages-dev previamente no seleccionado
Preparando para desempaquetar .../11-manpages-dev_5.10-1ubuntu1.0.
Desempaquetando manpages-dev (5.10-1ubuntu1) ...
Configurando manpages-dev (5.10-1ubuntu1) ...
Configurando gettext (0.21-4ubuntu4) ...
Configurando linux-libc-dev:amd64 (5.15.0-72.79) ...
Configurando libcurl4-gnutls-dev:amd64 (7.81.0-1ubuntu1.0) ...
Configurando libtirpc-dev:amd64 (1.3.2-2ubuntu0.1) ...
Configurando rpcsvc-proto (1.4.2-0ubuntu6) ...
Configurando libnsl-dev:amd64 (1.3.0-2build2) ...
Configurando libcrypt-dev:amd64 (1:4.4.27-1) ...
Configurando libc-dev-bin (2.35-0ubuntu3.1) ...
Configurando libc-devtools (2.35-0ubuntu3.1) ...
```

Obtenido el archivo comprimido para instalación de Git y alojado en el directorio **/Descargas/** procedemos a extraerlo, compilarlo y e instalar el paquete:
(Si el formato fuese .tar.bz2 extraeríamos mediante el comando `tar -xvjf git-1.7.6.tar.bz2`)
En este caso, al tratarse de un zip con una versión más actualizada lo descomprimimos con el comando `unzip git-master.zip`

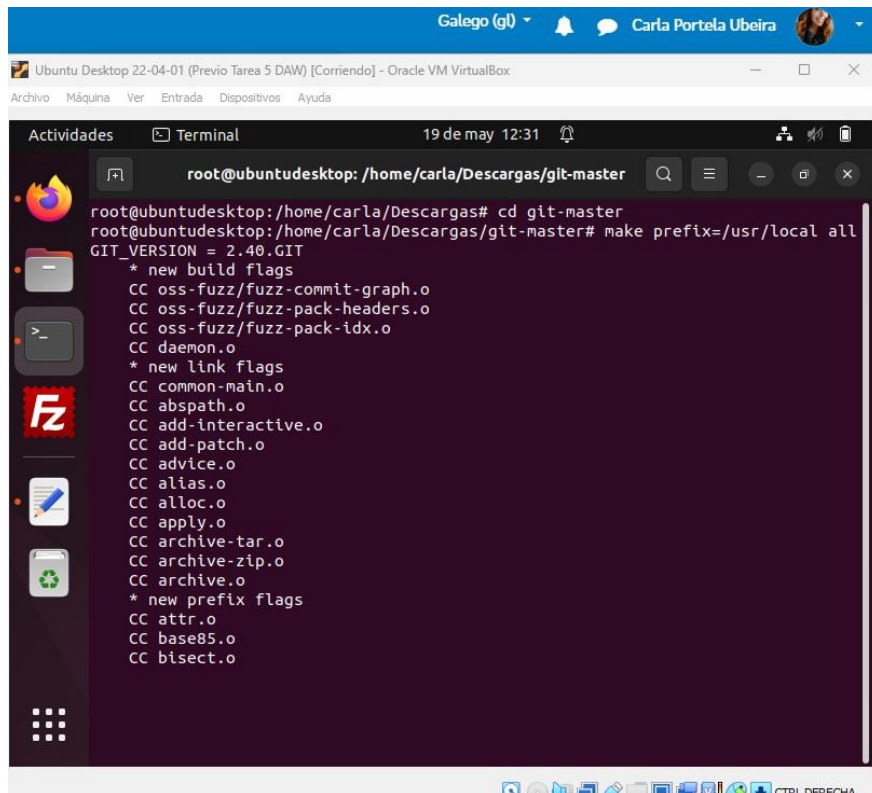


```
root@ubuntudesktop: /home/carla/Descargas

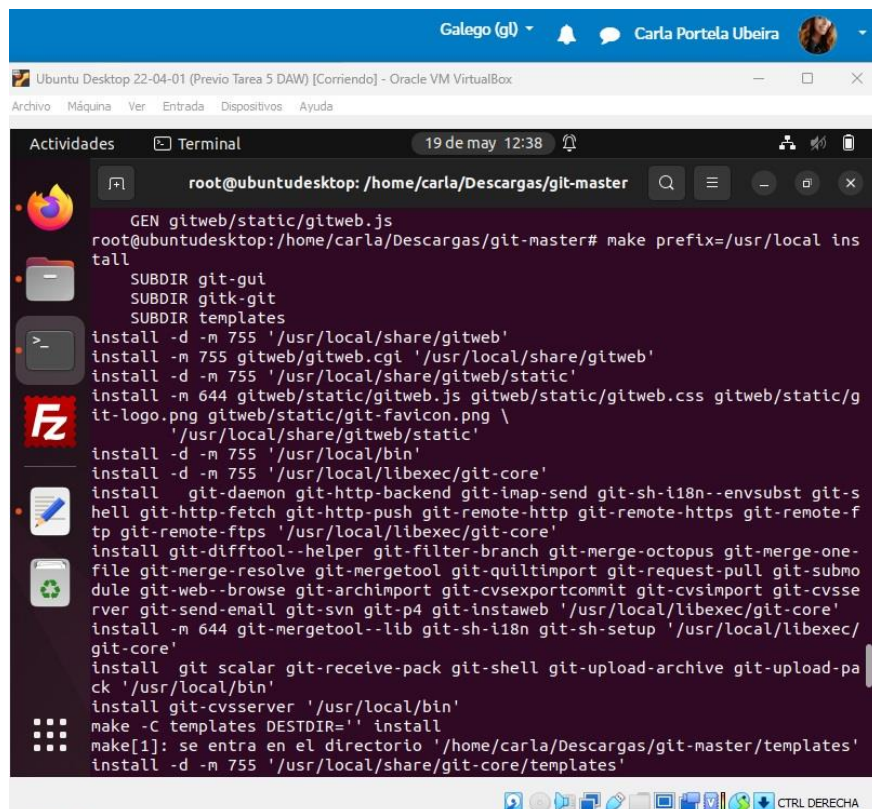
root@ubuntudesktop: /home/carla/Descargas# ls
git-1.8.4.5      'pussycat video flv.mp4'  wildfly-26.1.3.Final.tar.gz
git-1.8.4.5.tar.bz2  quickstart-26.x          wildfly-26.1.3.Final.zip
git-master.zip    quickstart-26.x.zip
root@ubuntudesktop: /home/carla/Descargas# unzip git-master
Archive: git-master.zip
004e0f790f947c9c511a9ac4f905021c7dbfa9e1
  creating: git-master/
  inflating: git-master/.cirrus.yml
  inflating: git-master/.clang-format
  inflating: git-master/.editorconfig
  inflating: git-master/.gitattributes
    creating: git-master/.github/
  inflating: git-master/.github/CONTRIBUTING.md
  inflating: git-master/.github/PULL_REQUEST_TEMPLATE.md
    creating: git-master/.github/workflows/
  inflating: git-master/.github/workflows/check-whitespace.yml
  inflating: git-master/.github/workflows/l10n.yml
  inflating: git-master/.github/workflows/main.yml
  inflating: git-master/.gitignore
  inflating: git-master/.gitmodules
  inflating: git-master/.mailmap
  inflating: git-master/.tsan-suppressions
  inflating: git-master/CODE_OF_CONDUCT.md
  inflating: git-master/COPYING
    creating: git-master/Documentation/
  extracting: git-master/Documentation/.gitattributes
  inflating: git-master/Documentation/.gitignore
  inflating: git-master/Documentation/CodingGuidelines
```

Instalamos **make** mediante *apt make install*, que es una utilidad que permite determinar automáticamente qué piezas de un programa necesitan ser recompiladas, y lanzar las órdenes para recompilarlas; y el compilador **gcc** mediante *apt install cc*.

Accedemos al directorio donde se descomprimieron los archivos mediante *cd git-master* y ejecutamos *make prefix=/usr/local all* y *make prefix=/usr/local install*



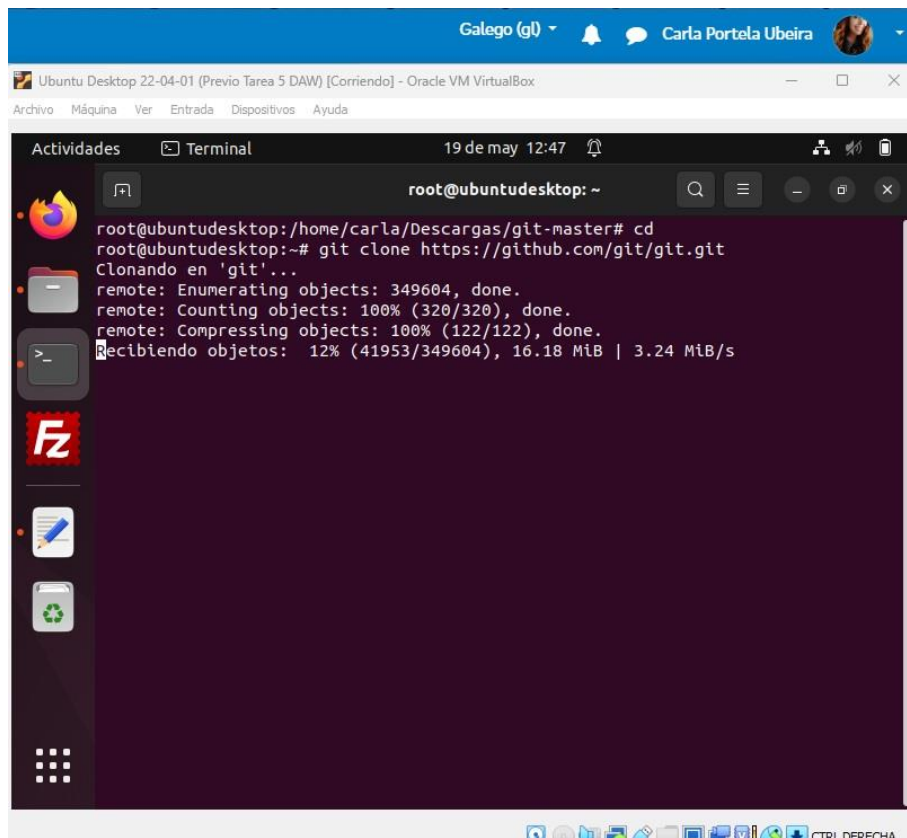
```
root@ubuntu: /home/carla/Descargas/git-master
root@ubuntu: /home/carla/Descargas/git-master# make prefix=/usr/local all
GIT_VERSION = 2.40.GIT
* new build flags
CC oss-fuzz/fuzz-commit-graph.o
CC oss-fuzz/fuzz-pack-headers.o
CC oss-fuzz/fuzz-pack-idx.o
CC daemon.o
* new link flags
CC common-main.o
CC abspath.o
CC add-interactive.o
CC add-patch.o
CC advice.o
CC alias.o
CC alloc.o
CC apply.o
CC archive-tar.o
CC archive-zip.o
CC archive.o
* new prefix flags
CC attr.o
CC base85.o
CC bisect.o
```



```
root@ubuntu: /home/carla/Descargas/git-master# make prefix=/usr/local install
GEN gitweb/static/gitweb.js
SUBDIR git-gui
SUBDIR gitk-git
SUBDIR templates
install -d -m 755 '/usr/local/share/gitweb'
install -m 755 gitweb/gitweb.cgi '/usr/local/share/gitweb'
install -d -m 755 '/usr/local/share/gitweb/static'
install -m 644 gitweb/static/gitweb.js gitweb/static/gitweb.css gitweb/static/g
it-logo.png gitweb/static/git-favicon.png \
'/usr/local/share/gitweb/static'
install -d -m 755 '/usr/local/bin'
install -d -m 755 '/usr/local/libexec/git-core'
install git-daemon git-http-backend git-imap-send git-sh-i18n--envsubst git-s
hell git-http-fetch git-http-push git-remote-http git-remote-https git-remote-f
tp git-remote-https '/usr/local/libexec/git-core'
install git-difftool--helper git-filter-branch git-merge-octopus git-merge-one-
file git-merge-resolve git-mergetool git-quiltimport git-request-pull git-submo
dule git-web--browse git-archimport git-cvsexportcommit git-cvsimport git-cvse
rver git-send-email git-svn git-p4 git-instaweb '/usr/local/libexec/git-core'
install -m 644 git-mergetool--lib git-sh-i18n git-sh-setup '/usr/local/libexec/
git-core'
install git-scalar git-receive-pack git-shell git-upload-archive git-upload-pa
ck '/usr/local/bin'
install git-cvsserver '/usr/local/bin'
make -C templates DESTDIR='' install
make[1]: se entra en el directorio '/home/carla/Descargas/git-master/templates'
install -d -m 755 '/usr/local/share/git-core/templates'
```

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

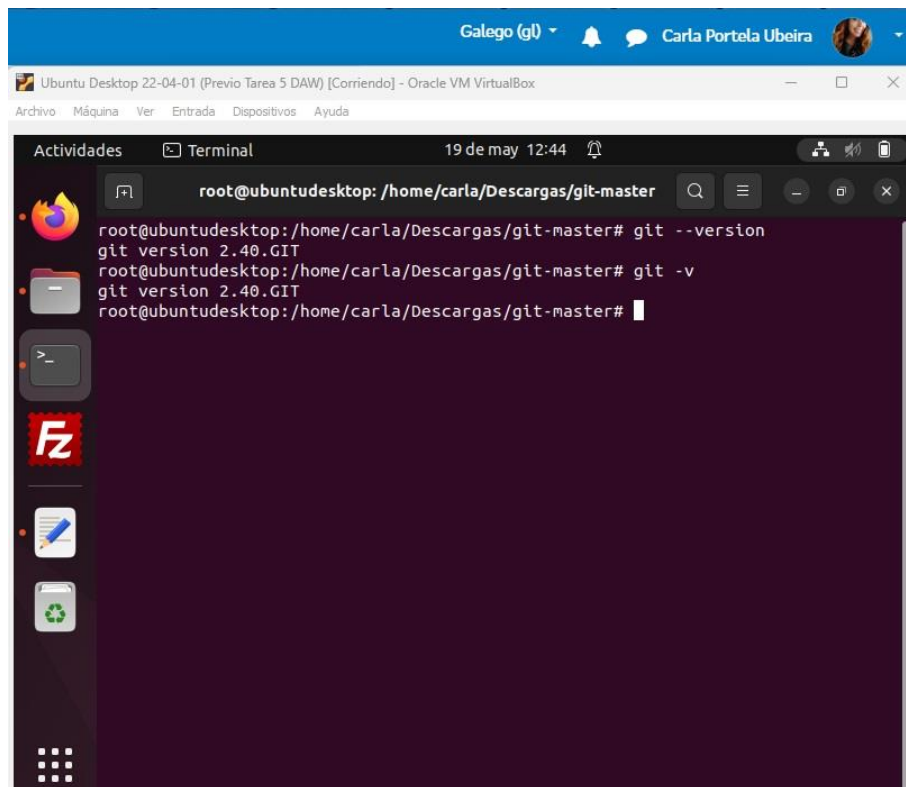
Empleando el comando *git clone* <https://github.com/git/git.git> por el cual se descargaría el código fuente desde su repositorio automáticamente.



The screenshot shows a terminal window titled "Terminal" with the date and time "19 de may 12:47". The prompt is "root@ubuntudesktop: ~". The user has navigated to the directory "/home/carla/Descargas/git-master" and executed the command "git clone https://github.com/git/git.git". The output shows the cloning process: "Clonando en 'git'...", "remote: Enumerating objects: 349604, done.", "remote: Counting objects: 100% (320/320), done.", "remote: Compressing objects: 100% (122/122), done.", and "Recibiendo objetos: 12% (41953/349604), 16.18 MiB | 3.24 MiB/s". The terminal window is part of an Ubuntu Desktop environment running in Oracle VM VirtualBox.

```
root@ubuntudesktop:/home/carla/Descargas/git-master# cd
root@ubuntudesktop:~# git clone https://github.com/git/git.git
Clonando en 'git'...
remote: Enumerating objects: 349604, done.
remote: Counting objects: 100% (320/320), done.
remote: Compressing objects: 100% (122/122), done.
Recibiendo objetos: 12% (41953/349604), 16.18 MiB | 3.24 MiB/s
```

- **Comprobar la versión que se ha instalado de Git.**
Verificamos la versión instalada de Git mediante *git--version*

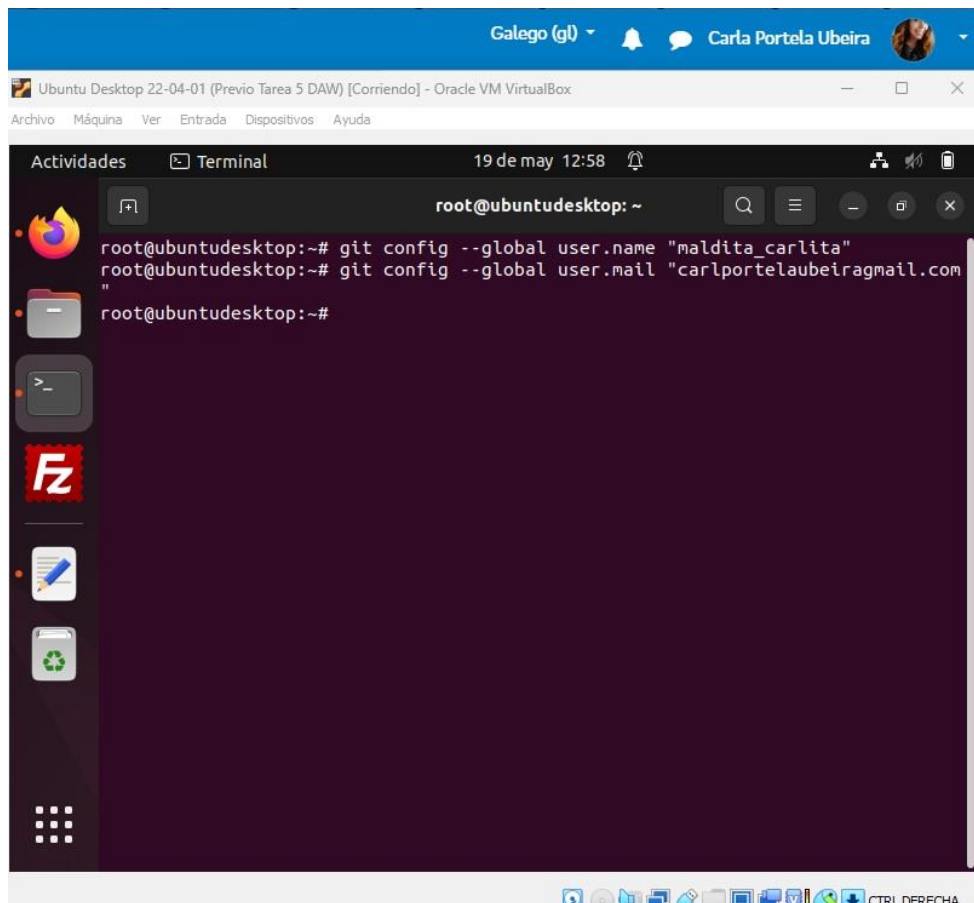


- Establecer

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

Git trae una herramienta llamada **git config** que permite obtener y establecer variables de configuración que controlan el aspecto y funcionamiento de Git.

Lo primero que se debe hacer cuando se instala Git es establecer el nombre de usuario y dirección de correo electrónico. Esto es importante porque las confirmaciones de cambios (commits) en Git usan esta información, y es introducida de manera inmutable en los commits que el usuario va a enviar. Para ello usamos el comando *git config --global user.name* “(nombre)” y *git config --global user.email* “(dirección de correo)”

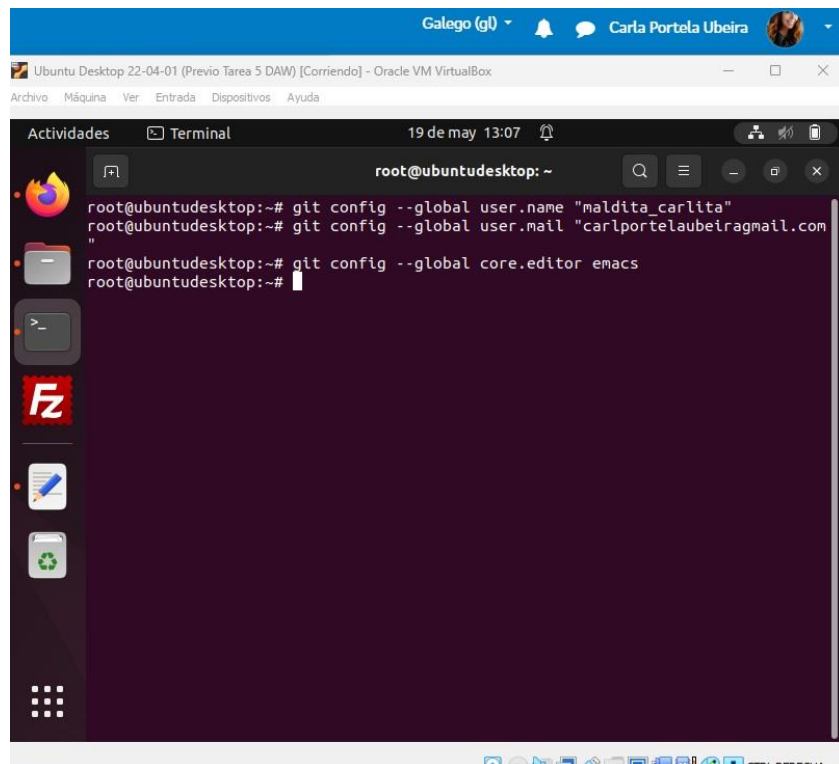


Solamente se necesita hacer esto una vez si se especifica la opción `--global`, ya que Git siempre usará esta información para todo lo que se haga en ese sistema.

En el caso de querer sobrescribir esta información con otro nombre o dirección de correo para proyectos específicos, se ejecuta el mismo comando sin la opción `--global` cuando estemos en el proyecto concreto.

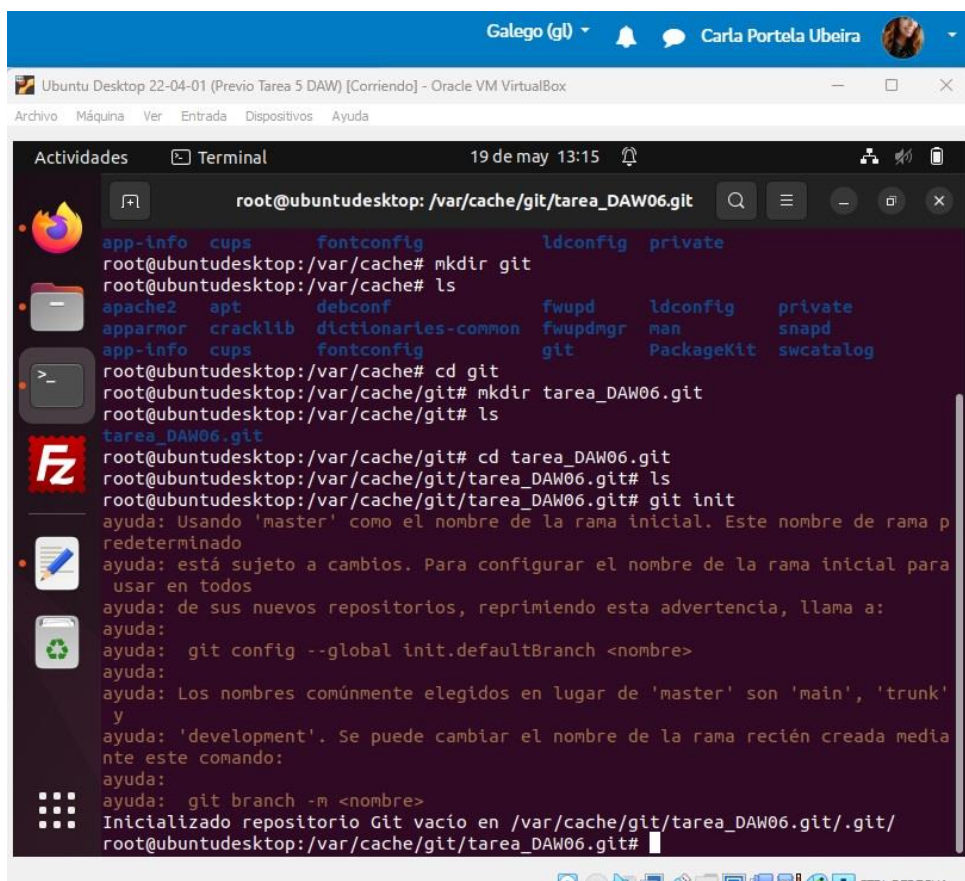
- **Cambiar el editor de texto que trae por defecto Git al editor emacs.**

Si no se indica nada, Git usa el editor por defecto del sistema que, generalmente, es Vi o Vim. En el caso de querer usar otro editor de texto usamos `git config --global core.editor` (nombre del editor de texto)

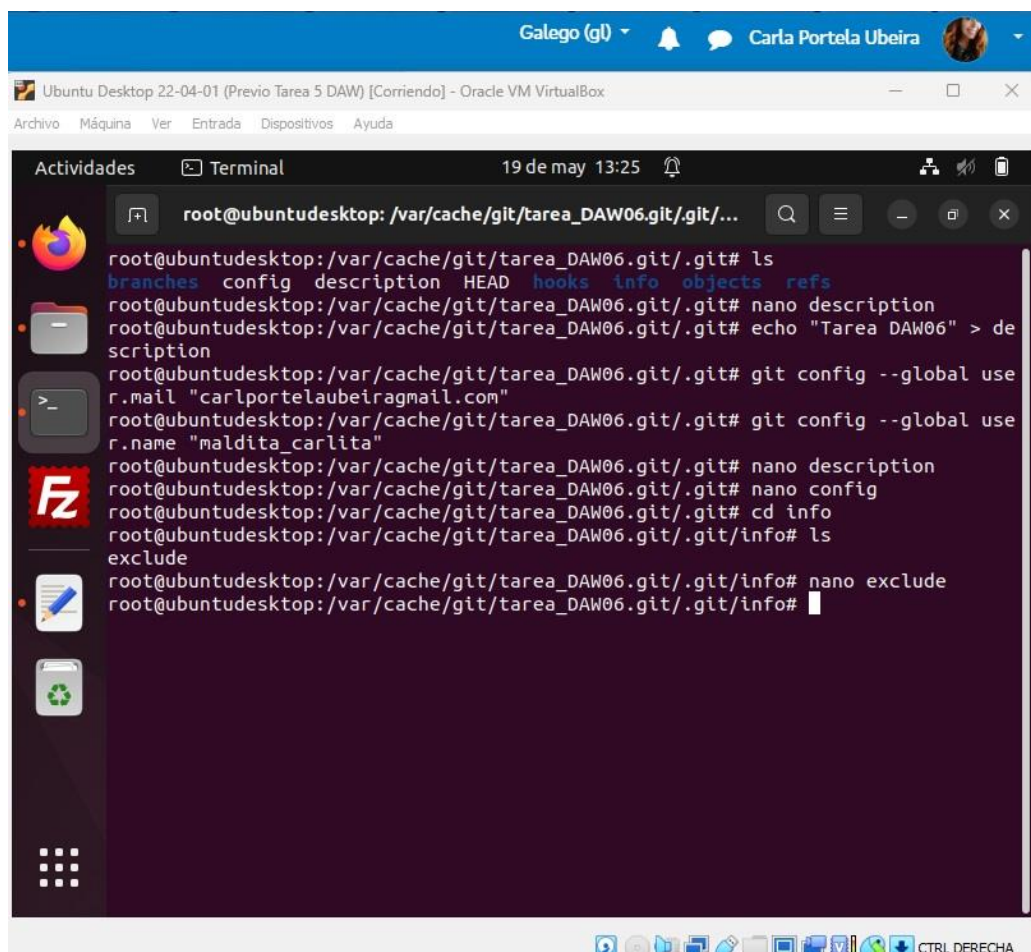


- Dentro de la carpeta `/var/cache/git/` crear una carpeta para un nuevo proyecto denominado `tarea_DAW06` e iniciar un repositorio el nuevo proyecto.

Creamos la carpeta del proyecto moviéndonos al directorio `/var/cache/git` mediante `cd /var/cache/git`, creamos el proyecto con el comando `mkdir` (nombre del proyecto).`git` y nos movemos a dicho directorio mediante `cd tarea_DAW06.git`
E iniciamos un repositorio del nuevo proyecto mediante el comando `git init`



Comprobamos que en el directorio .git hay una serie de archivos asociados al proyecto que se ha creado automáticamente.



```
root@ubuntudesktop: /var/cache/git/tarea_DAW06.git/.git/...
root@ubuntudesktop:/var/cache/git/tarea_DAW06.git/.git# ls
branches  config  description  HEAD  hooks  info  objects  refs
root@ubuntudesktop:/var/cache/git/tarea_DAW06.git/.git# nano description
root@ubuntudesktop:/var/cache/git/tarea_DAW06.git/.git# echo "Tarea DAW06" > de
scription
root@ubuntudesktop:/var/cache/git/tarea_DAW06.git/.git# git config --global use
r.mail "carlportelaubeiragmail.com"
root@ubuntudesktop:/var/cache/git/tarea_DAW06.git/.git# git config --global use
r.name "maldita_carlita"
root@ubuntudesktop:/var/cache/git/tarea_DAW06.git/.git# nano description
root@ubuntudesktop:/var/cache/git/tarea_DAW06.git/.git# nano config
root@ubuntudesktop:/var/cache/git/tarea_DAW06.git/.git# cd info
root@ubuntudesktop:/var/cache/git/tarea_DAW06.git/.git/info# ls
exclude
root@ubuntudesktop:/var/cache/git/tarea_DAW06.git/.git/info# nano exclude
root@ubuntudesktop:/var/cache/git/tarea_DAW06.git/.git/info#
```

Y
añadimos
una breve

descripción en el archivo **description** y configuramos nuestro nombre y el correo para ese proyecto (sin la opción --global)