

1. Realiza la instalación de git en linux

```
administrador@UBUNTUDAW:/$ sudo apt-get install git
[sudo] contraseña para administrador:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  git-man liberror-perl
Paquetes sugeridos:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
Se instalarán los siguientes paquetes NUEVOS:
  git git-man liberror-perl
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 84 no actualizados.
Se necesita descargar 4147 kB de archivos.
Se utilizarán 21,0 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
Des:1 http://es.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.17029-1 [26,5 kB]
Des:2 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git-man all 1:2.34.1-1ubuntu1.10 [954 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 git amd64 1:2.34.1-1ubuntu1.10 [3166 kB]
Descargados 4147 kB en 1s (6195 kB/s)
```

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

Se puede obtener Git a través del propio Git clonando el repositorio oficial. Para hacerlo, ejecutaremos el siguiente comando:

```
sudo git clone git://git.kernel.org/pub/scm/git/git.git
```

```
administrador@UBUNTUDAW:/$ sudo git clone git://git.kernel.org/pub/scm/git/git.git
Clonando en 'git'...
remote: Enumerating objects: 362291, done.
remote: Counting objects: 100% (6143/6143), done.
remote: Compressing objects: 100% (1048/1048), done.
remote: Total 362291 (delta 5540), reused 5395 (delta 5094), pack-reused 356148
Recibiendo objetos: 100% (362291/362291), 120.77 MiB | 14.18 MiB/s, listo.
Resolviendo deltas: 100% (272705/272705), listo.
```

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

```
administrador@UBUNTUDAW:/$ git --version
git version 2.34.1
administrador@UBUNTUDAW:/$
```

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

```
administrador@UBUNTUDAW:/$ git config --global user.name "Javier"
administrador@UBUNTUDAW:/$ git config --global user.email "javigonzaalez@outlook.es"
administrador@UBUNTUDAW:/$
```

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

```
git config --global core.editor "emacs"
```

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

Primero crearemos la carpeta “git” dentro de “cache”, y crearemos un repositorio dentro de nombre “tarea_git”:

```
sudo mkdir git
```

```
sudo mkdir tarea_git
```

Ahora, dentro de la carpeta “tarea_git”, ejecutaremos el siguiente comando:

```
administrador@UBUNTUDAW:/var/cache/git/tarea_git$ sudo git init
ayuda: Usando 'master' como el nombre de la rama inicial. Este nombre de rama predeterminado
ayuda: está sujeto a cambios. Para configurar el nombre de la rama inicial para usar en todos
ayuda: de sus nuevos repositorios, reprimiendo esta advertencia, llama a:
ayuda:
ayuda: git config --global init.defaultBranch <nombre>
ayuda:
ayuda: Los nombres comúnmente elegidos en lugar de 'master' son 'main', 'trunk' y
ayuda: 'development'. Se puede cambiar el nombre de la rama recién creada mediante este comando:
ayuda:
ayuda: git branch -m <nombre>
Inicializado repositorio Git vacío en /var/cache/git/tarea_git/.git/
```

7. Haz alguno de los ejercicios indicados en (como crear un fichero y hacer commit del cambio):

Vamos a crear un fichero dentro del repositorio:

```
sudo touch fichero.txt
```

```
sudo git add ejemplo.txt
```

```
administrador@UBUNTUDAW:/var/cache/git/tarea_git$ sudo git add fichero.txt
administrador@UBUNTUDAW:/var/cache/git/tarea_git$
```

```
sudo git commit -m "Añadir archivo fichero.txt"
```