



CICLO: DAW
MÓDULO DE DESPLIEGUE DE
APLICACIONES WEB.

Tarea Nº 6

Alumno:
Beatriz Gil Durán
76137032-V

Los documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos incluidos en este contenido pueden contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en el contenido. Fomento Ocupacional FOC SL puede realizar en cualquier momento, sin previo aviso, mejoras y/o cambios en el contenido.

Es responsabilidad del usuario el cumplimiento de todas las leyes de derechos de autor aplicables. Ningún elemento de este contenido (documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos asociados), ni parte de este contenido puede ser reproducida, almacenada o introducida en un sistema de recuperación, ni transmitida de ninguna forma ni por ningún medio (ya sea electrónico, mecánico, por fotocopia, grabación o de otra manera), ni con ningún propósito, sin la previa autorización por escrito de Fomento Ocupacional FOC SL.

Este contenido está protegido por la ley de propiedad intelectual e industrial. Pertenecen a Fomento Ocupacional FOC SL los derechos de autor y los demás derechos de propiedad intelectual e industrial sobre este contenido.

Sin perjuicio de los casos en que la ley aplicable prohíbe la exclusión de la responsabilidad por daños, Fomento Ocupacional FOC SL no se responsabiliza en ningún caso de daños indirectos, sean cuales fueren su naturaleza u origen, que se deriven o de otro modo estén relacionados con el uso de este contenido.

© 2022 Fomento Ocupacional FOC SL todos los derechos reservados.

Contenido

1. Documentos que se adjuntan a este informe.	2
2. (RA6_d) Se han utilizado herramientas colaborativas para la elaboración y mantenimiento de la documentación.....	2
3. (RA6_e) Se ha instalado, configurado y utilizado un sistema de control de versiones.....	3
4. (RA6_f) Se ha garantizado la accesibilidad y seguridad de la documentación almacenada por el sistema de control de versiones.	8
5. (RA6_g) Se ha documentado la instalación, configuración y uso del sistema de control de versiones utilizado.....	11

1. Documentos que se adjuntan a este informe.

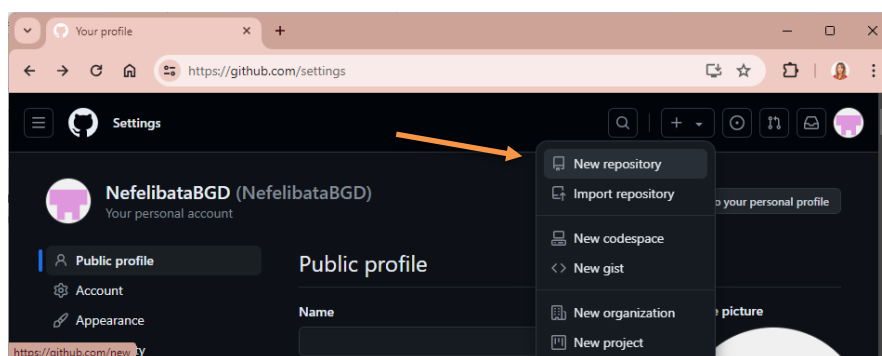
A continuación, se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea.

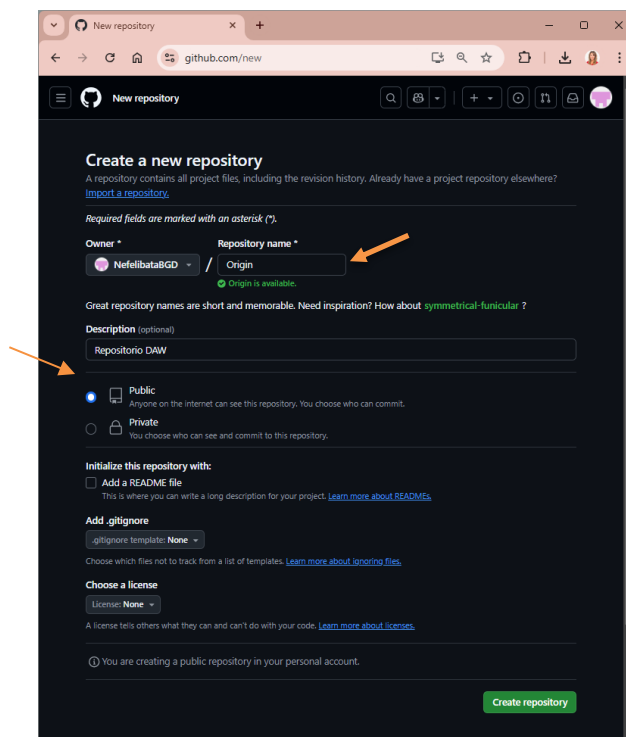
2. (RA6_d) Se han utilizado herramientas colaborativas para la elaboración y mantenimiento de la documentación.

El alumno/a se registrará en GitHub, creará un repositorio para la documentación.

Tras crearnos una cuenta en github desde el dashboard hacemos click en New repository:



Vamos a crear un repositorio al cual nos conectaremos desde nuestro servidor. Establecemos un nombre para el repositorio, le llamaremos “origin” por buenas prácticas al ser el repositorio principal. Seleccionamos la privacidad PUBLIC.



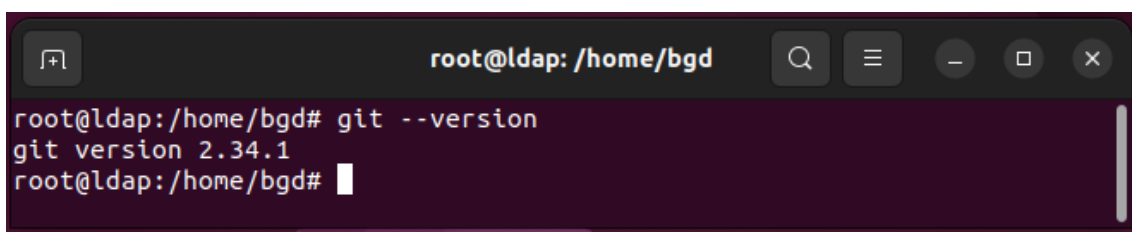
3. (RA6_e) Se ha instalado, configurado y utilizado un sistema de control de versiones.

El alumno/a instalará la aplicación git en su terminal y la sincronizará con su repositorio.

Instalación

Git viene instalado por defecto en nuestro Ubuntu 22.04. Vamos a comprobar su versión con el siguiente comando:

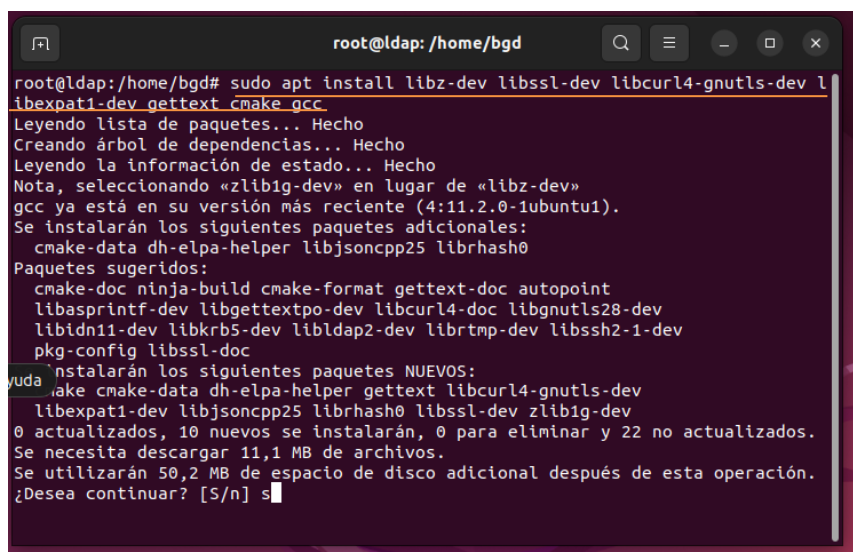
- ✓ “git --version”.



```
root@ldap: /home/bgd
root@ldap:/home/bgd# git --version
git version 2.34.1
root@ldap:/home/bgd#
```

Sin embargo, vamos a actualizar a la última versión. Para ello debemos instalar previamente unas librerías:

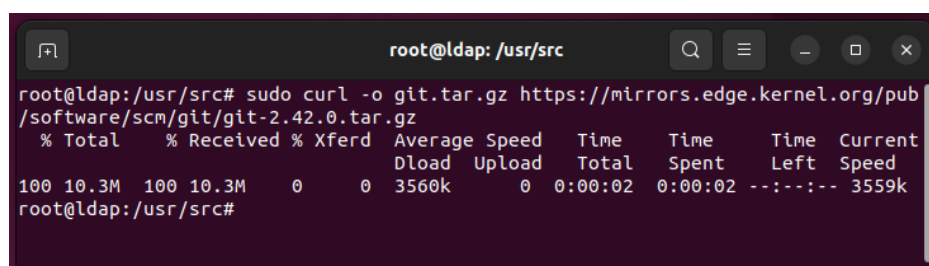
- ✓ “sudo apt install libz-dev libssl-dev libcurl4-gnutls-dev libexpat1-dev gettext cmake gcc”.



```
root@ldap: /home/bgd
root@ldap:/home/bgd# sudo apt install libz-dev libssl-dev libcurl4-gnutls-dev l
libexpat1-dev gettext cmake gcc
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Nota, seleccionando «zlib1g-dev» en lugar de «libz-dev»
gcc ya está en su versión más reciente (4:11.2.0-1ubuntu1).
Se instalarán los siguientes paquetes adicionales:
  cmake-data dh-elpa-helper libjsoncpp25 librhash0
Paquetes sugeridos:
  cmake-doc ninja-build cmake-format gettext-doc autopoint
  libasprintf-dev libgettextpo-dev libcurl4-doc libgnutls28-dev
  libidn11-dev libkrb5-dev libldap2-dev librtmp-dev libssh2-1-dev
  pkg-config libssl-doc
Se instalarán los siguientes paquetes NUEVOS:
  cmake cmake-data dh-elpa-helper gettext libcurl4-gnutls-dev
  libexpat1-dev libjsoncpp25 librhash0 libssl-dev zlib1g-dev
0 actualizados, 10 nuevos se instalarán, 0 para eliminar y 22 no actualizados.
Se necesita descargar 11,1 MB de archivos.
Se utilizarán 50,2 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

A continuación, descargamos desde el sitio oficial la versión actual en tarball.

- ✓ “sudo curl -o git.tar.gz https://mirrors.edge.kernel.org/pub/software/scm/git/git-2.42.0.tar.gz”.



```
root@ldap: /usr/src
root@ldap:/usr/src# sudo curl -o git.tar.gz https://mirrors.edge.kernel.org/pub
/software/scm/git/git-2.42.0.tar.gz
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 10.3M  100 10.3M    0     0  3560k      0  0:00:02  0:00:02 --:--:-- 3559k
root@ldap:/usr/src#
```

Una vez descargado, descomprimiremos ejecutando la siguiente secuencia de comandos:

- ✓ “cd/tmp”.
- ✓ “sudo tar -xvzf git.tar.gz”.
- ✓ “cd git-2.42.0”.

```

root@ldap: /usr/src/git-2.42.0
git-2.42.0/xdiff/xmacros.h
git-2.42.0/xdiff/xmerge.c
git-2.42.0/xdiff/xpatience.c
git-2.42.0/xdiff/xprepare.c
git-2.42.0/xdiff/xprepare.h
git-2.42.0/xdiff/xtypes.h
git-2.42.0/xdiff/xutils.c
git-2.42.0/xdiff/xutils.h
git-2.42.0/configure
git-2.42.0/version
git-2.42.0/git-gui/version
root@ldap: /usr/src# cd git-2.42.0
root@ldap: /usr/src/git-2.42.0#
  
```

Una vez dentro del directorio con la fuente de git compilaremos la aplicación localmente con el siguiente comando:

- ✓ “sudo make prefix=/usr/local all”.

```

root@ldap: /usr/src/git-2.42.0
root@ldap: /usr/src/git-2.42.0# sudo make prefix=/usr/local all
GIT_VERSION = 2.42.0
* 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
  
```

Tras esto, instalaremos la aplicación con el siguiente comando:

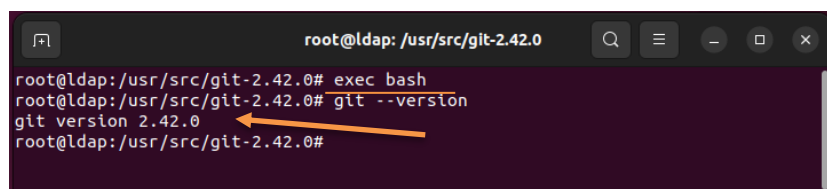
- ✓ “sudo make prefix=/usr/local install”.

```

root@ldap: /usr/src/git-2.42.0
root@ldap: /usr/src/git-2.42.0# sudo make prefix=/usr/local install
SUBDIR git-gui
SUBDIR gitk-gui
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/gi
t-logo.png gitweb/static/git-favicon.png \
'/usr/local/share/gitweb/static'
install -d -m 755 '/usr/local/bin'
  
```

Finalmente actualizaremos las variables del sistema y comprobamos la versión final de Git.

- ✓ "exec bash".
- ✓ "git --version".



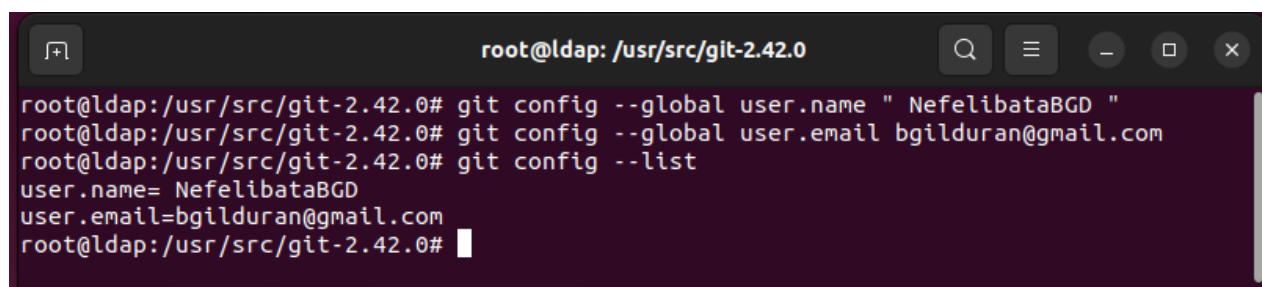
```
root@ldap: /usr/src/git-2.42.0
root@ldap:/usr/src/git-2.42.0# exec bash
root@ldap:/usr/src/git-2.42.0# git --version
git version 2.42.0
root@ldap:/usr/src/git-2.42.0#
```

Configuración:

Git trae una herramienta, git config, que permite obtener y establecer variables de configuración que controlan el aspecto y funcionamiento de git. Esta herramienta la emplearemos tras instalar git, ya que lo primero que se debe hacer 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.

Vamos a usar los siguientes comandos para establecer el nombre de usuario y correo electrónico:

- ✓ "git config --global user.name " NefelibataBGD ". Establecemos el usuario.
- ✓ "git config --global user.email bgilduran@gmail.com". Establecemos el correo electrónico.
- ✓ "git config --list". Comprobamos que se han actualizado los datos.



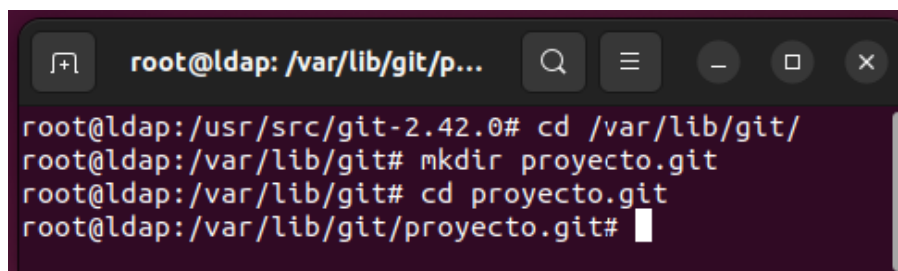
```
root@ldap: /usr/src/git-2.42.0
root@ldap:/usr/src/git-2.42.0# git config --global user.name " NefelibataBGD "
root@ldap:/usr/src/git-2.42.0# git config --global user.email bgilduran@gmail.com
root@ldap:/usr/src/git-2.42.0# git config --list
user.name= NefelibataBGD
user.email=bgilduran@gmail.com
root@ldap:/usr/src/git-2.42.0#
```

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.

Creación de un nuevo Proyecto:

Para trabajar con un proyecto en Git, lo primero que debemos hacer es crear la carpeta del mismo ejecutando la siguiente secuencia de comandos:

- ✓ "cd /var/lib/git/".
- ✓ "mkdir proyecto.git".
- ✓ "cd proyecto.git".



```

root@ldap: /var/lib/git/p...
root@ldap:/usr/src/git-2.42.0# cd /var/lib/git/
root@ldap:/var/lib/git# mkdir proyecto.git
root@ldap:/var/lib/git# cd proyecto.git
root@ldap:/var/lib/git/proyecto.git#

```

Una vez creada la carpeta, iniciamos un repositorio para nuestro nuevo proyecto y lo configuramos de acuerdo a nuestras necesidades con el comando:

- ✓ "sudo git init".



```

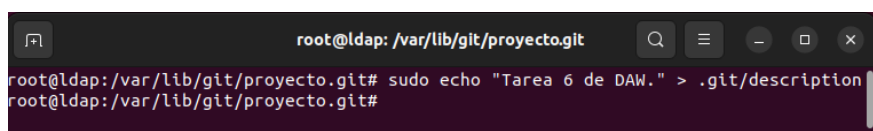
root@ldap: /var/lib/git/proyecto.git
root@ldap:/var/lib/git/proyecto.git# sudo git init
ayuda: Usando 'master' como el nombre de la rama inicial. Este nombre de rama
ayuda: predeterminado
ayuda: está sujeto a cambios. Para configurar el nombre de la rama inicial pa
ayuda: ra usar en todos
ayuda: de sus 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', 'trun
ayuda: k' y
ayuda: 'development'. Se puede cambiar el nombre de la rama recién creada med
ayuda: iante este comando:
ayuda:
ayuda: git branch -m <nombre>
Iniciado repositorio Git vacío en /var/lib/git/proyecto.git/.git/
root@ldap:/var/lib/git/proyecto.git#

```

Como podemos ver devuelve un mensaje indicando que el repositorio ha sido inicializado.

El siguiente paso es configurar el proyecto en el directorio .git ejecutando los siguientes comandos (aunque los dos últimos los obviaremos porque ya lo hemos configurado anteriormente):

- ✓ "sudo echo "Tarea 6 de DAW." > .git/description".
- ✓ "sudo git config --global user.name "Tu nombre"".
- ✓ "sudo git config --global user.email tu@correo.com".



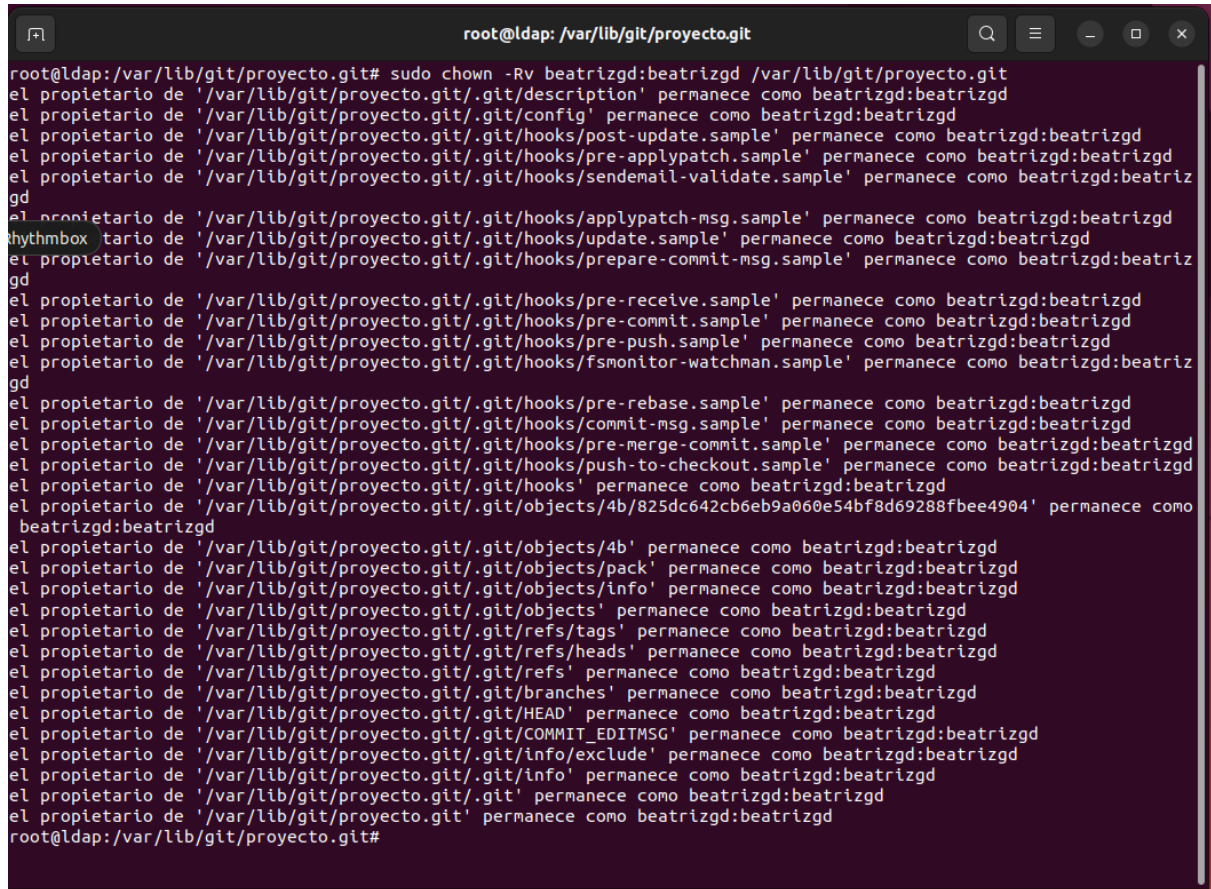
```

root@ldap: /var/lib/git/proyecto.git
root@ldap:/var/lib/git/proyecto.git# sudo echo "Tarea 6 de DAW." > .git/description
root@ldap:/var/lib/git/proyecto.git#

```


Para permitir que otro usuario pueda trabajar en el repositorio (sin ser root), vamos a darle los permisos:

- ✓ “`sudo chown -R beatrizgd:beatrizgd /var/lib/git/proyecto.git`”.



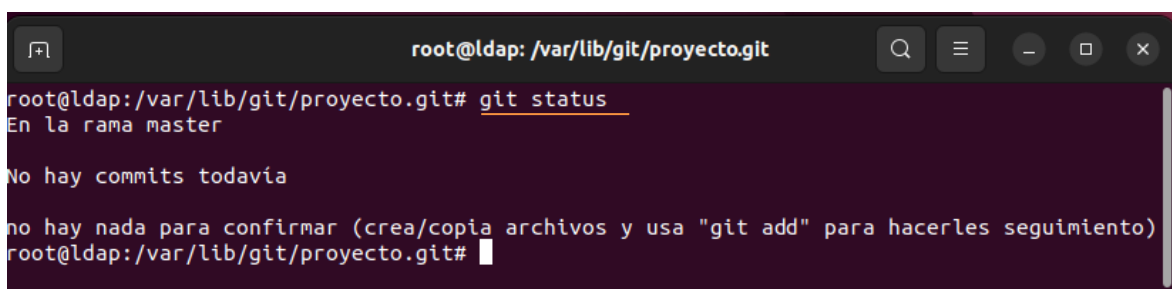
```

root@ldap: /var/lib/git/proyecto.git
root@ldap:/var/lib/git/proyecto.git# sudo chown -Rv beatrizgd:beatrizgd /var/lib/git/proyecto.git
el propietario de '/var/lib/git/proyecto.git/.git/description' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/config' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/post-update.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/pre-update.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/sendemail-validate.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/applypatch-msg.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/update.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/pre-applypatch.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/prepare-commit-msg.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/pre-receive.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/pre-commit.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/pre-push.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/fsmonitor-watchman.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/pre-rebase.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/commit-msg.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/pre-merge-commit.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks/push-to-checkout.sample' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/hooks' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/objects/4b/825dc642cb6eb9a060e54bf8d69288fbee4904' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/objects/4b' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/objects/pack' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/objects/info' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/objects' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/refs/tags' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/refs/heads' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/refs' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/branches' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/HEAD' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/COMMIT_EDITMSG' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/info/exclude' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git/info' permanece como beatrizgd:beatrizgd
el propietario de '/var/lib/git/proyecto.git/.git' permanece como beatrizgd:beatrizgd
root@ldap:/var/lib/git/proyecto.git#

```

Podemos comprobar el estatus del repositorio mediante el siguiente comando:

- ✓ “`git status`”.



```

root@ldap: /var/lib/git/proyecto.git
root@ldap:/var/lib/git/proyecto.git# git status
En la rama master

No hay commits todavía

no hay nada para confirmar (crea/copia archivos y usa "git add" para hacerles seguimiento)
root@ldap:/var/lib/git/proyecto.git#

```

Sincronización:

Nos aseguramos de estar en el directorio del repositorio local:

- ✓ “`cd /var/lib/git/proyecto.git`”.

Agregamos el repositorio remoto de GitHub como origen (origin):

- ✓ “git remote add origin https://github.com/NefelibataBGD/Origin.git”.

```
root@ldap: /var/lib/git/proyecto.git
root@ldap:/var/lib/git/proyecto.git# git remote add origin https://github.com/NefelibataBGD/Origin.git
root@ldap:/var/lib/git/proyecto.git#
```

Verificamos el estado de los remotos:

- ✓ “git remote -v”.

```
root@ldap: /var/lib/git/proyecto.git
root@ldap:/var/lib/git/proyecto.git# git remote -v
origin https://github.com/NefelibataBGD/Origin.git (fetch)
origin https://github.com/NefelibataBGD/Origin.git (push)
root@ldap:/var/lib/git/proyecto.git#
```

Para poder probarlo debemos incluir un archivo en el proyecto local y subirlo lo cual se documentará en el siguiente punto.

4. (RA6_f) Se ha garantizado la accesibilidad y seguridad de la documentación almacenada por el sistema de control de versiones.

El alumno/a realizará varias actualizaciones del repositorio, por ejemplo, subiendo en primer lugar los documentos HTML y posteriormente el PDF.

Vamos a incluir de primeras el index.html.

```
root@ldap: /var/lib/git/proyecto.git
root@ldap:/var/lib/git/proyecto.git# tree
└─ index.html ←
0 directories, 1 file
root@ldap:/var/lib/git/proyecto.git#
```

Hacemos un “**git status**” y comprobamos si aparece:



```
root@ldap: /var/lib/git/proyecto.git
root@ldap:/var/lib/git/proyecto.git# git status
En la rama master

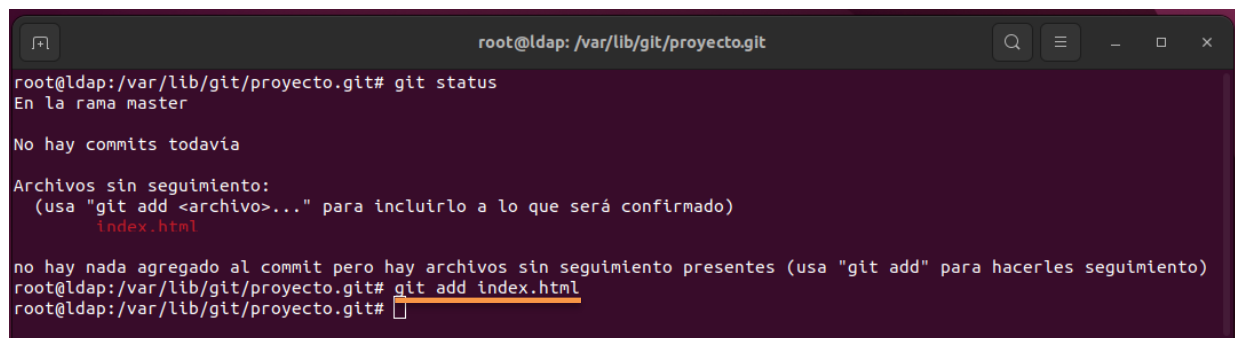
No hay commits todavía

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que será confirmado)
  index.html

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "
git add" para hacerles seguimiento)
root@ldap:/var/lib/git/proyecto.git#
```

Como podemos ver aparece, pero en archivos sin seguimiento. Ahora debemos utilizar el siguiente comando:

- ✓ “**git add index.html**”.



```
root@ldap: /var/lib/git/proyecto.git
root@ldap:/var/lib/git/proyecto.git# git status
En la rama master

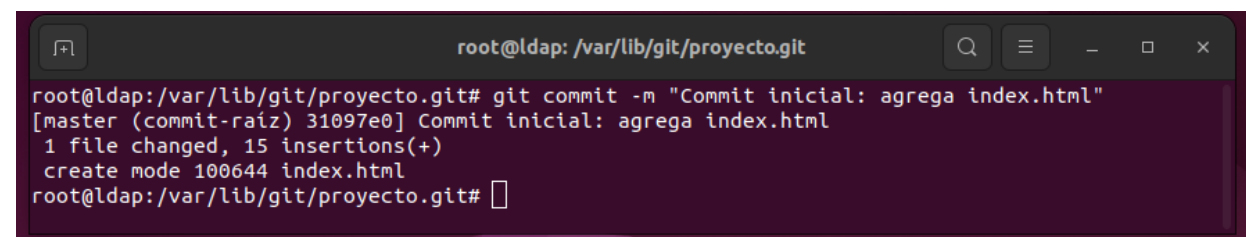
No hay commits todavía

Archivos sin seguimiento:
  (usa "git add <archivo>..." para incluirlo a lo que será confirmado)
  index.html

no hay nada agregado al commit pero hay archivos sin seguimiento presentes (usa "git add" para hacerles seguimiento)
root@ldap:/var/lib/git/proyecto.git# git add index.html
root@ldap:/var/lib/git/proyecto.git#
```

Ahora procedemos a hacer el commit inicial para guardar los cambios en el repositorio local:

- ✓ “**git commit -m "Commit inicial: agrega index.html"**”.

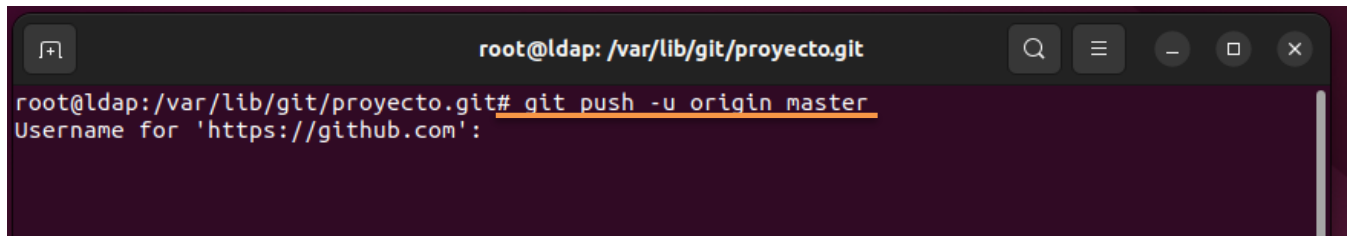


```
root@ldap: /var/lib/git/proyecto.git
root@ldap:/var/lib/git/proyecto.git# git commit -m "Commit inicial: agrega index.html"
[master (commit-raiz) 31097e0] Commit inicial: agrega index.html
1 file changed, 15 insertions(+)
create mode 100644 index.html
root@ldap:/var/lib/git/proyecto.git#
```

Ya lo tenemos guardado en el repositorio local.

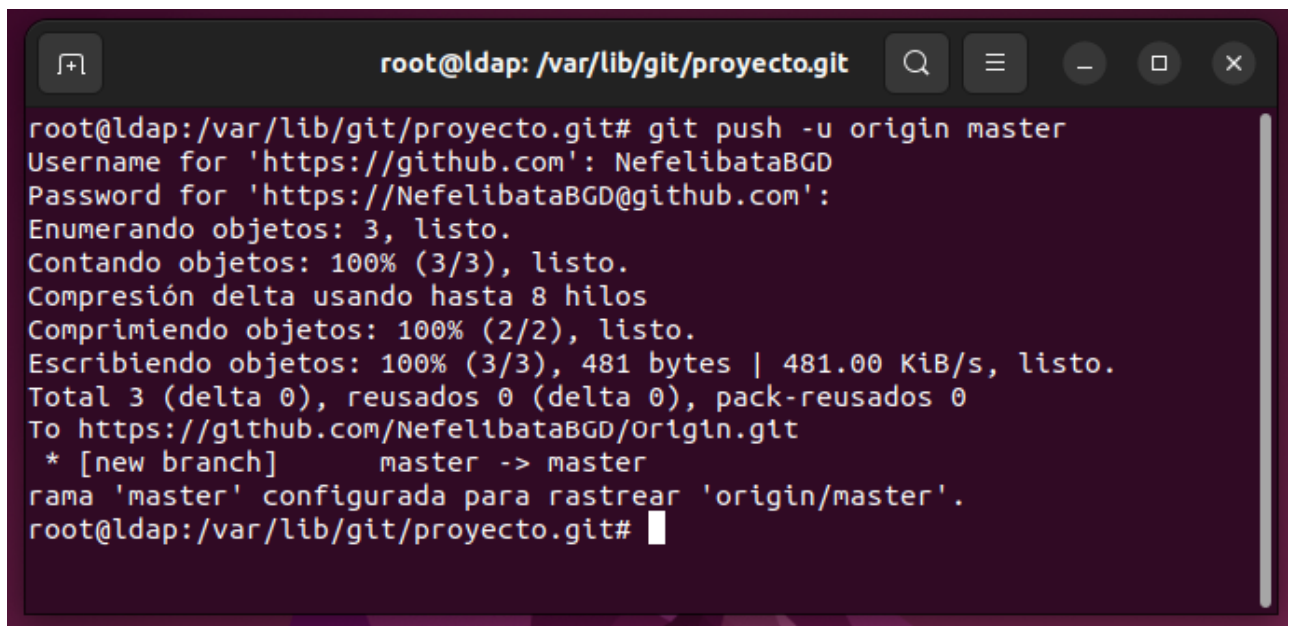
Ahora vamos a sincronizarlo con el remoto.

✓ “`git push -u origin master`”.



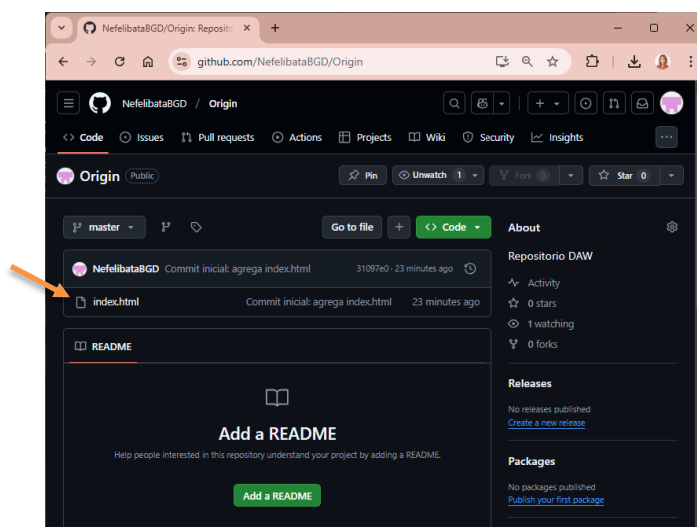
```
root@ldap: /var/lib/git/proyecto.git
root@ldap:/var/lib/git/proyecto.git# git push -u origin master
Username for 'https://github.com':
```

Como Podemos observar nos pide usuario de Git. Introduciremos el usuario y como pass un Token que tenemos creados previamente.



```
root@ldap: /var/lib/git/proyecto.git
root@ldap:/var/lib/git/proyecto.git# git push -u origin master
Username for 'https://github.com': NefelibataBGD
Password for 'https://NefelibataBGD@github.com':
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Compresión delta usando hasta 8 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 481 bytes | 481.00 KiB/s, listo.
Total 3 (delta 0), reusados 0 (delta 0), pack-reusados 0
To https://github.com/NefelibataBGD/Origin.git
 * [new branch]      master -> master
rama 'master' configurada para rastrear 'origin/master'.
root@ldap:/var/lib/git/proyecto.git#
```

El push se ha realizado correctamente, y el archivo ha sido subido al repositorio en GitHub. Vamos a ir a comprobarlo:



5. (RA6_g) Se ha documentado la instalación, configuración y uso del sistema de control de versiones utilizado.

El alumno/a debe desarrollar y recoger en el informe los pasos necesarios para satisfacer los RA6_d, RA6_e y RA6_f.

Se muestran los vínculos a los apartados donde han sido ampliamente detallados y explicados paso a paso:

- [\(RA6_d\) Se han utilizado herramientas colaborativas para la elaboración y mantenimiento de la documentación.](#)
- [\(RA6_e\) Se ha instalado, configurado y utilizado un sistema de control de versiones.](#)
- [\(RA6_f\) Se ha garantizado la accesibilidad y seguridad de la documentación almacenada por el sistema de control de versiones.](#)