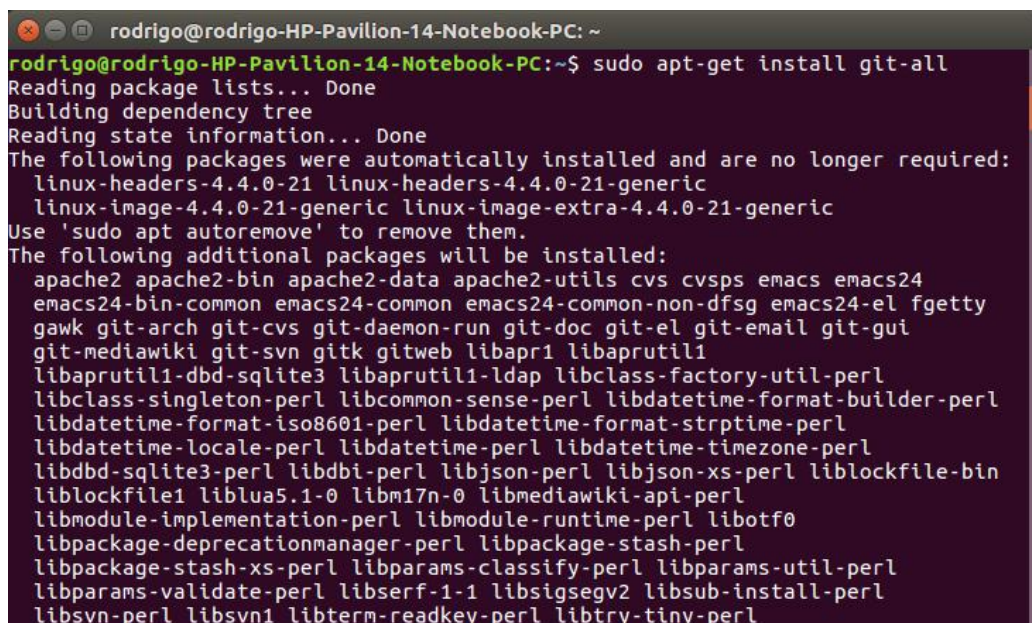


Manual para subir archivos a GitHub

Antes de comenzar a subir archivos, debemos de contar con la creación de algún repositorio, por lo que es necesario instalar Git en nuestra computadora. Para eso, debemos abrir la consola y ejecutar el siguiente comando (Para Ubuntu):

```
sudo apt-get install git-all
```

Con ese comando empezarán a descargarse todos los paquetes y aparecerá una pantalla como la siguiente:



```
rodrigo@rodrigo-HP-Pavilion-14-Notebook-PC: ~  
rodrigo@rodrigo-HP-Pavilion-14-Notebook-PC:~$ sudo apt-get install git-all  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
  linux-headers-4.4.0-21 linux-headers-4.4.0-21-generic  
  linux-image-4.4.0-21-generic linux-image-extra-4.4.0-21-generic  
Use 'sudo apt autoremove' to remove them.  
The following additional packages will be installed:  
  apache2 apache2-bin apache2-data apache2-utils cvs cvsps emacs emacs24  
  emacs24-bin-common emacs24-common emacs24-common-non-dfsg emacs24-el fgetty  
  gawk git-arch git-cvs git-daemon-run git-doc git-el git-email git-gui  
  git-mediawiki git-svn gitk gitweb libapr1 libaprutil1  
  libaprutil1-dbd-sqlite3 libaprutil1-ldap libclass-factory-util-perl  
  libclass-singleton-perl libcommon-sense-perl libdatetime-format-builder-perl  
  libdatetime-format-iso8601-perl libdatetime-format-strptime-perl  
  libdatetime-locale-perl libdatetime-perl libdatetime-timezone-perl  
  libdbd-sqlite3-perl libdbi-perl libjson-perl libjson-xs-perl liblockfile-bin  
  liblockfile1 liblua5.1-0 libm17n-0 libmediawiki-api-perl  
  libmodule-implementation-perl libmodule-runtime-perl libotf0  
  libpackage-deprecationmanager-perl libpackage-stash-perl  
  libpackage-stash-xs-perl libparams-classify-perl libparams-util-perl  
  libparams-validate-perl libserf-1-1 libsigsegv2 libsub-install-perl  
  libsvn-perl libsvn1 libterm-readkey-perl libtry-tiny-perl
```

Si no se tiene alguna cuenta de GitHub es necesario que creamos alguna para poder utilizar la plataforma y poder trabajar con los archivos que queramos. Debemos de dar un nombre de usuario y un correo electrónico. Los comandos son los siguientes:

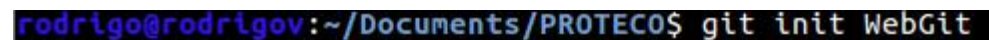
```
git config -g user.name "nombre_usuario"  
git config -g user.email "email_id"
```

Creación de un repositorio

Antes de crear el repositorio en sí, debemos de crear una carpeta en nuestra computadora, el cual nos servirá como un repositorio local. Para ello, es necesario poner el siguiente comando:

```
git init Carpeta
```

El nombre “Carpeta” puede ser sustituido por cualquier otro como por ejemplo “WebGit” así como se muestra en la siguiente imagen:



```
rodrigo@rodrigo:~/Documents/PROTECO$ git init WebGit
```

Después de ejecutar este comando, aparecerá un mensaje de que el repositorio se inició con éxito.

Después sigue entrar nuestra carpeta, en este caso “WebGit” con el comando:

```
cd WebGit
```

Ya tenemos nuestra carpeta donde se guardarán todos nuestros archivos para nuestro repositorio, lo que sigue entonces es la adición de los archivos del repositorio a un índice. Es un paso importante que se debe de hacer antes de poder subir los cambios a Github; hay que indexar todos los archivos contenidos en el repositorio local. Este índice contendrá los archivos nuevos así como los cambios a los archivos existentes en el repositorio local.

Para agregar archivos al índice se hará con el comando add, de la siguiente manera:

```
git add archivo
```

El comando “git add” se utiliza para agregar cualquier número de archivos y carpetas al índice.

Es necesario poner un git add . (con punto final) para agregar todos los cambios, sin especificar el nombre de los archivos. O bien podemos usar un git add seguido del nombre de un archivo en específico.



```
rodrigo@rodrigo:~/Documents/PROTECO/WebGit$ git add Test.java
```

Una vez que se añadan todos los archivos, es posible dejar un registro de los cambios que se hicieron, y esto se logra con un “commit”. Eso es para cuando ya se han terminado de agregar o modificar los archivos y que los cambios pueden ser subidos al repositorio de Github. El comando es el siguiente:

`git commit -m 'mensaje'`

'mensaje' puede ser cualquier mensaje que nosotros queramos y que describa las acciones que realizamos como "Agregué una modificación".

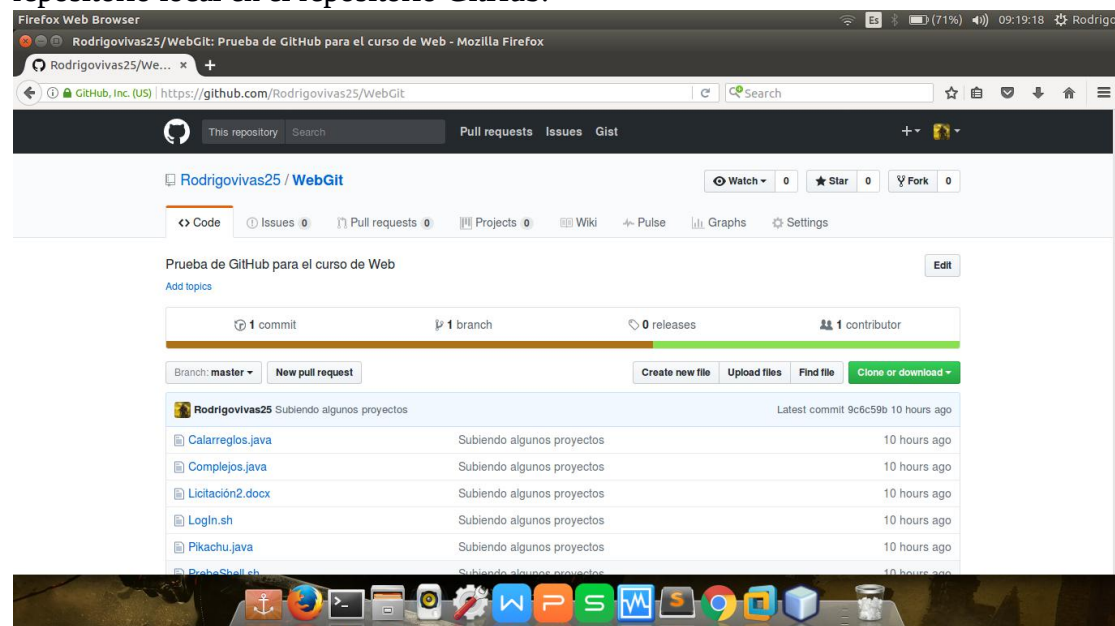
```
rodrigo@rodrigov:~/Documents/PROTECO/WebGit$ git add Factorial.java
rodrigo@rodrigov:~/Documents/PROTECO/WebGit$ git commit -m 'Un archivo mas'
[master 0812611] Un archivo mas
1 file changed, 35 insertions(+)
create mode 100644 Factorial.java
```

En la imagen nos muestra un mensaje que nos dice que un cambio ha ocurrido en el repositorio, y eso es porque agregamos un archivo a él; en este caso "Factorial.java".

- Creación de un repositorio en GitHub

Ya que se siguieron los pasos anteriores, tenemos que iniciar sesión en la página de GitHub y cuando estemos ahí, debemos crear un nuevo repositorio.

Una vez hecho esto se creará el repositorio y será posible subir el contenido del repositorio local en el repositorio GitHub.



Para conectarse al repositorio remoto en GitHub hay que ejecutar el comando:

`git remote add origin https://github.com/user_name/Carpeta.git`

```
rodrigo@rodrigov:~/Documents/PROTECO/WebGit$ git remote add origin https://github.com/Rodrigovivas25/WebGit.git
```

El paso final es empujar el contenido del repositorio local hacia el repositorio remoto, y eso se hace con el comando:

```
git push origin master
```

Esto subirá todo el contenido de la carpeta “Carpeta” (repositorio local) a GitHub (repositorio externo).

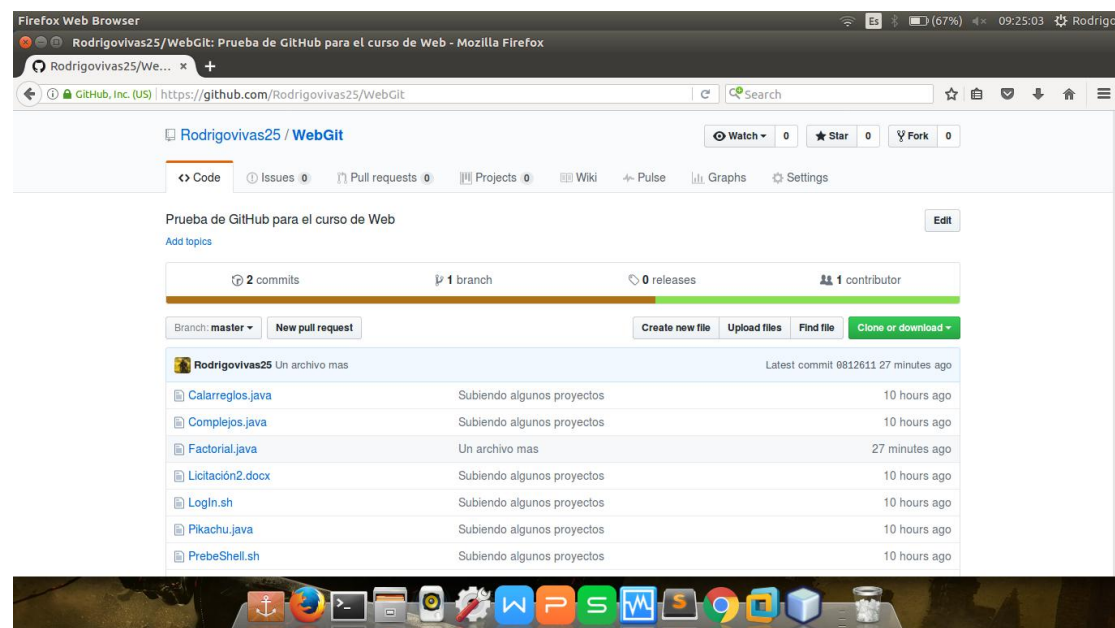
Para los archivos que queramos subir después, no será necesario hacer todo lo anterior, sino que basta con empezar desde el paso de creación de archivos.

Para que funcione lo anterior, es necesario hacer algún tipo de archivo, ya que de no hacerlo, nos mandará un error, debido a que no estamos subiendo nada. Si se tiene, por ejemplo, un archivo llamado Factorial.java, se verá lo siguiente:

```
rodrigo@rodrigo:~/Documents/PROTECO/WebGit$ git push origin master
Username for 'https://github.com': Rodrigovivas25
Password for 'https://Rodrigovivas25@github.com':

Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 686 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local objects.
To https://github.com/Rodrigovivas25/WebGit.git
9c6c59b..0812611 master -> master
```

Como se puede ver, al momento de hacer el push, pedirá el nombre y la contraseña de tu usuario de GitHub para poder mandarlo. Al finalizar, se verá lo siguiente en la página de GitHub:



Con esto, la creación de nuestro repositorio, quedaría terminada, al igual que nuestros archivos subidos a él.