Git

- Resumen basado en el libro 'Pro Git'
- Escrito por 'Scott Chacon' y por 'Ben Straub'
- Versión 2.1.22, 2021-02-09
- https://git-scm.com/book/es/v2

Acerca del Control de Versiones

Pg. 9

Un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.

Fundamentos de Git

Los Tres Estados

Pg. 16

Git tiene tres estados principales en los que se pueden encontrar tus archivos:

- confirmado (**committed**), significa que los datos están almacenados de manera segura en tu base de datos local
- modificado (**modified**), significa que has modificado el archivo, pero todavía no lo has confirmado a tu base de datos.
- preparado (**staged**), significa que has marcado un archivo modificado en su versión actual para que vaya en tu próxima confirmación.

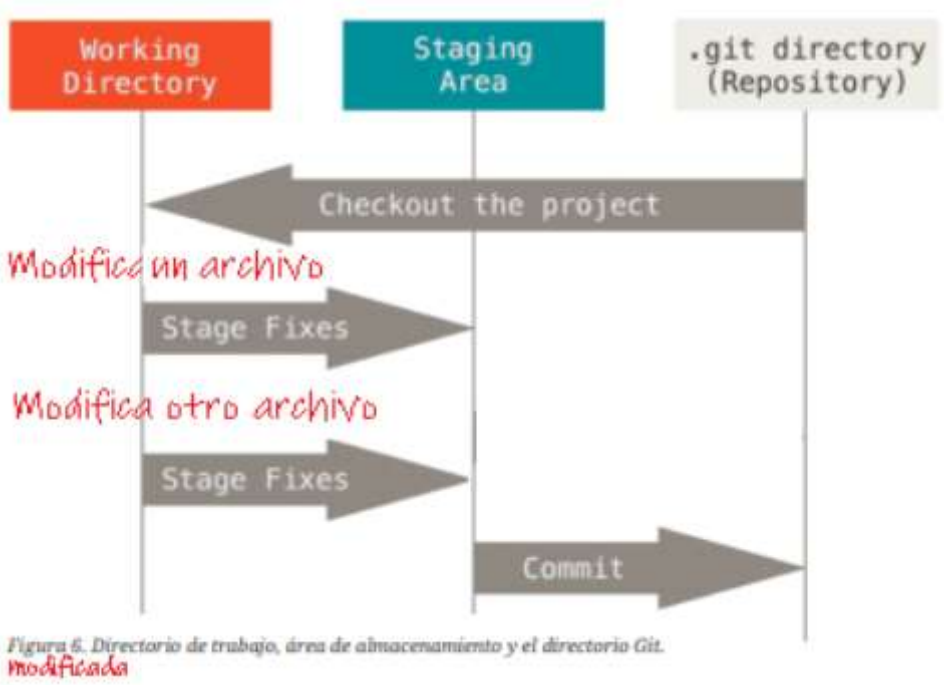


Figura 6. Directorio de trabajo, área de almacenamiento y el directorio Git.
modificada

Configurando Git por primera vez

Tu Identidad

Pg. 20

```
1 !git config --list
```

```
1 # no olvide colocar su nombre
2 !git config  --global user.name "cambiar-por-su-nombre"
```

```
1 # no olvide colocar su email
2 !git config  --global user.email cambiar.por.su.correo@gmail.com
```

```
1 !git config --list
```

```
user.name=cambiar-por-su-nombre
user.email=cambiar.por.su.correo@gmail.com
```

```
1 !git config user.name
```

```
cambiar-por-su-nombre
```

```
1 !git config user.email
```

```
cambiar.por.su.correo@gmail.com
```

Tu Editor

Pg. 21

Más adelante veremos que Git abre automáticamente el editor que este configurado cuando se confirman (`commit`) los cambios. El editor se configura de la siguiente forma

`git config --global core.editor` cambiar_por_su_editor

Desde Jupyter es preferible no abrir ningún editor ya que Jupyter se puede ejecutar en diferentes entornos. Para evitar abrir un editor al confirmar (`commit`) veremos que se puede usar la opción `-m`

```
commit -m "Aquí escribimos la documentación del cambio"
```

¿Cómo obtener ayuda?

Pg. 22

```
1 #!git help
```

```
1 #!git help config
```

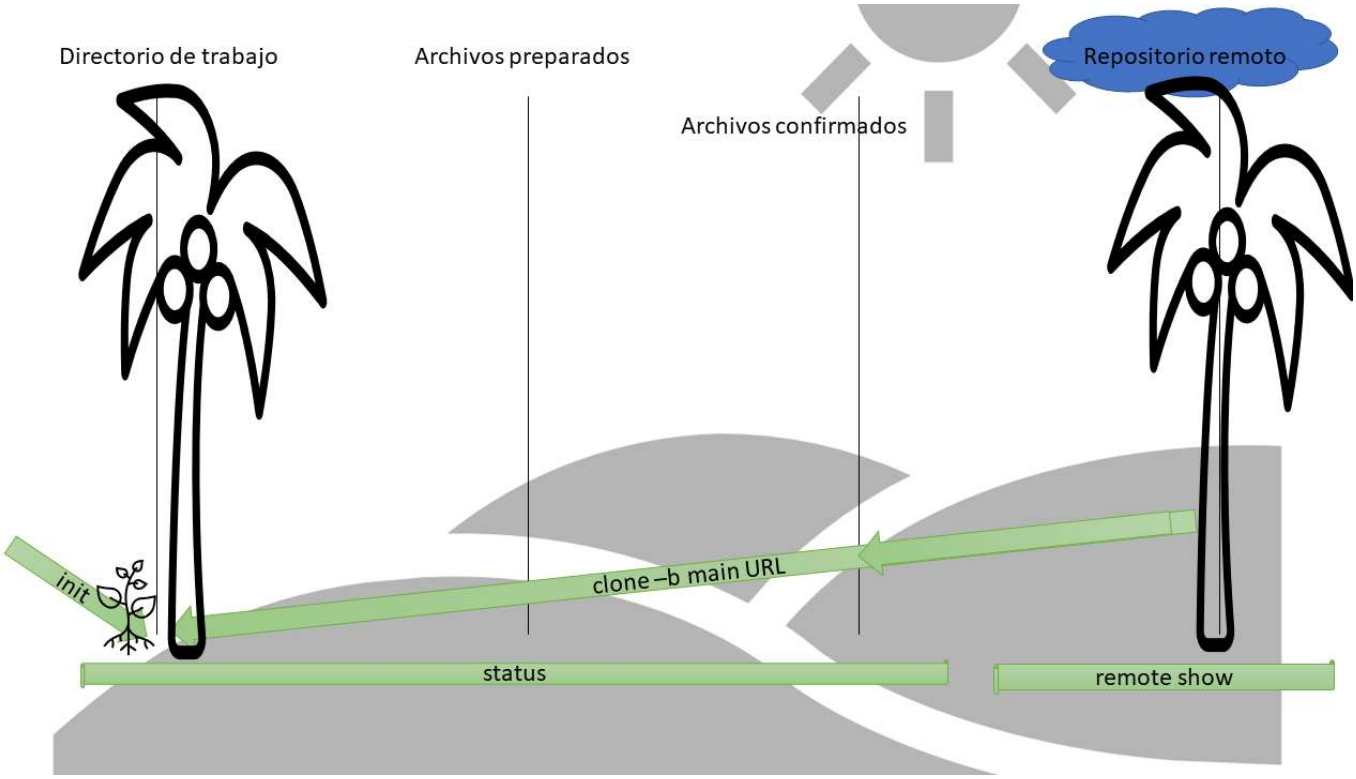
```
1
```

Fundamentos de Git

Pgs. 24 - 57

Obteniendo un repositorio Git

Pg. 23



Antes de comenzar vamos a listar los archivos en la carpeta actual, utilizando el comando de Linux `ls -l`

```
1 !ls -l
```

```
total 4
drwxr-xr-x 1 root root 4096 Apr  7 13:36 sample_data
```

La rama principal de GitHub por defecto se llama `main`. Sin embargo, en Git se llama `master`. Para evitar que al clonar se cree la rama `master`, es necesario indicarle el nombre de la rama con la opción `-b`.

```
1 !git clone -b main https://github.com/cambiar-por-su-nombre/TrabajosProgramacion
```

```
Cloning into 'TrabajosProgramacion'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

```
1 !ls -l
```

```
total 8
drwxr-xr-x 1 root root 4096 Apr  7 13:36 sample_data
drwxr-xr-x 3 root root 4096 Apr 15 03:11 TrabajosProgramacion
```

```
1 %cd TrabajosProgramacion
```

```
/content/TrabajosProgramacion
```

Guardando cambios en el Repositorio

Pg. 24

Revisando el Estado de tus Archivos

```
1 !git status
```

```
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

```
1 !ls -l
```

```
total 4
-rw-r--r-- 1 root root 23 Apr 15 03:11 README.md
```

```
1 !ls -al
```

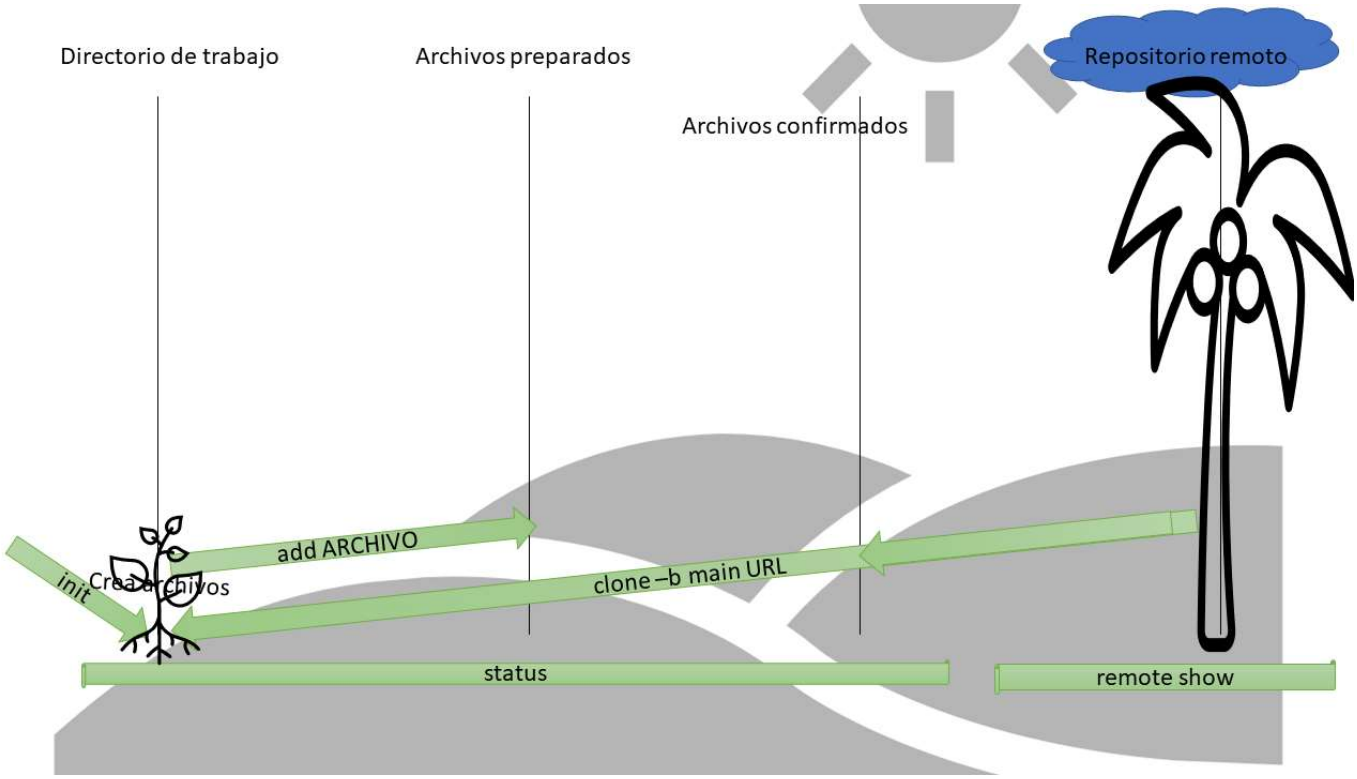
```
total 16
drwxr-xr-x 3 root root 4096 Apr 15 03:11 .
drwxr-xr-x 1 root root 4096 Apr 15 03:11 ..
drwxr-xr-x 8 root root 4096 Apr 15 03:11 .git
-rw-r--r-- 1 root root   23 Apr 15 03:11 README.md
```

```
1 !ls .git -al
2
```

```
total 52
drwxr-xr-x 8 root root 4096 Apr 15 03:11 .
drwxr-xr-x 3 root root 4096 Apr 15 03:11 ..
drwxr-xr-x 2 root root 4096 Apr 15 03:11 branches
-rw-r--r-- 1 root root  282 Apr 15 03:11 config
-rw-r--r-- 1 root root   73 Apr 15 03:11 description
-rw-r--r-- 1 root root   21 Apr 15 03:11 HEAD
drwxr-xr-x 2 root root 4096 Apr 15 03:11 hooks
-rw-r--r-- 1 root root  137 Apr 15 03:11 index
drwxr-xr-x 2 root root 4096 Apr 15 03:11 info
drwxr-xr-x 3 root root 4096 Apr 15 03:11 logs
drwxr-xr-x 7 root root 4096 Apr 15 03:11 objects
-rw-r--r-- 1 root root  112 Apr 15 03:11 packed-refs
drwxr-xr-x 5 root root 4096 Apr 15 03:11 refs
```

Rastrear Archivos Nuevos

Pg. 26



Creando un primer archivo (`concluciones.txt`)

Entra las muchísimas opciones que tenemos para crear un archivo, vamos a usar una que tiene Linux para crear archivos de texto. Esta opción consiste en el comando `echo`.

```
1 # `echo ... > ...` permite crear un archivo
2 !echo "Finalmente podemos concluir que ..." > conclusion.txt
```

```
1 !ls -l
```

```
total 8
-rw-r--r-- 1 root root 36 Apr 15 03:11 conclusion.txt
-rw-r--r-- 1 root root 23 Apr 15 03:11 README.md
```

```
1 # `cat ...` permite crear un archivo
2 !cat conclusion.txt
```

```
Finalmente podemos concluir que ...
```

```
1 !git status
```

```
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        conclusion.txt
```

```
nothing added to commit but untracked files present (use "git add" to track)
```

```
1 !git add conclusion.txt
```

```
1 !git status
```

```
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   conclusion.txt
```

Creando un segundo archivo (`resumen.txt`)

```
1 # Se puede copiar un archivo en la carpeta o crearlo desde cualquier programa
2 !echo "En este documento se presenta ..." > resumen.txt
```

```
1 !git status
```

```
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   conclusion.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        resumen.txt
```

```
1 !git add resumen.txt
```

```
1 !git status
```

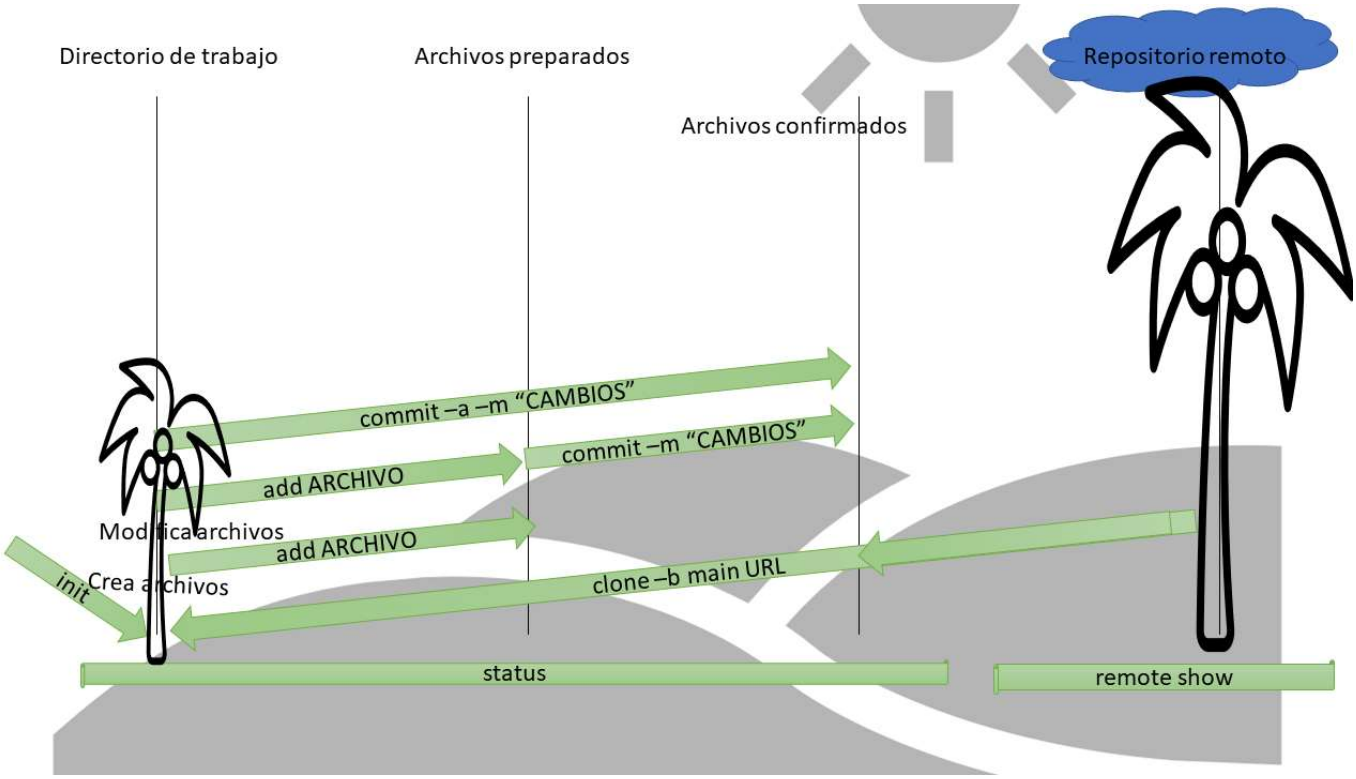
```
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   conclusion.txt
        new file:   resumen.txt
```

Confirmar tus Cambios (`commit`)

Pg. 33




```
1 !git commit -m "Versión inicial del proyecto"
2
```

```
[main 16941cb] Versión inicial del proyecto
 2 files changed, 2 insertions(+)
 create mode 100644 conclusion.txt
 create mode 100644 resumen.txt
```

```
1 !git status
```

```
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Modificando y preparando archivos

Pg. 26

```
1 !echo "... en este aspecto mejoró 3% respecto ..." >> conclusion.txt
```

```
1 !type conclusion.txt
```

```
/bin/bash: line 0: type: conclusion.txt: not found
```

```
1 !git status
```

```
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   conclusion.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

```
1 !echo "... mejores resultados que los reportados." >> resumen.txt
```

```
1 !cat resumen.txt
```

```
En este documento se presenta ...
... mejores resultados que los reportados.
```

```
1 !git status
```

```
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   conclusion.txt
        modified:   resumen.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

```
1 !echo "Desde los inicios de ..." >> introduccion.txt
```

```
1 !git status
```

```
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   conclusion.txt
        modified:   resumen.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        introduccion.txt
```

```
    no changes added to commit (use "git add" and/or "git commit -a")
```

```
1 !echo "... se presentará en las conclusiones." >> introduccion.txt
```

```
1 !git status
```

```
    On branch main
    Your branch is ahead of 'origin/main' by 1 commit.
      (use "git push" to publish your local commits)

    Changes not staged for commit:
      (use "git add <file>..." to update what will be committed)
      (use "git checkout -- <file>..." to discard changes in working directory)

            modified:   conclusion.txt
            modified:   resumen.txt

    Untracked files:
      (use "git add <file>..." to include in what will be committed)

            introduccion.txt

    no changes added to commit (use "git add" and/or "git commit -a")
```

```
1 !git add conclusion.txt
```

```
1 !git status
```

```
    On branch main
    Your branch is ahead of 'origin/main' by 1 commit.
      (use "git push" to publish your local commits)

    Changes to be committed:
      (use "git reset HEAD <file>..." to unstage)

            modified:   conclusion.txt

    Changes not staged for commit:
      (use "git add <file>..." to update what will be committed)
      (use "git checkout -- <file>..." to discard changes in working directory)

            modified:   resumen.txt

    Untracked files:
      (use "git add <file>..." to include in what will be committed)

            introduccion.txt
```

```
1 !echo "... superando en tres de los cinco ..." >> conclusion.txt
```

```
1 !type conclusion.txt
```

```
    /bin/bash: line 0: type: conclusion.txt: not found
```

```
1 !git status
```

```
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   conclusion.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   conclusion.txt
        modified:   resumen.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        introduccion.txt
```

```
1 !git status -s
2 # ??  No rastreado (Untracked)
3 # A  Preparado (Staged)
4 # M  Modificado (Modified)
5     # Columna izquierda es el estado preparado
6     # Columna derecha es el estado sin preparar
```

```
MM conclusion.txt
 M resumen.txt
?? introduccion.txt
```

```
1 !git add introduccion.txt
```

```
1 !git status -s
```

```
MM conclusion.txt
A  introduccion.txt
 M resumen.txt
```

```
1 !git add conclusion.txt
```

```
1 !git status -s
```

```
M  conclusion.txt
A  introduccion.txt
 M resumen.txt
```

```
1 !git add resumen.txt
```

```
1 !git status -s
```

```
 M  conclusion.txt
 A  introduccion.txt
 M  resumen.txt
```

```
1 !git commit -m "Se actualizó la conclusión y el resumen con los resultados y se comenzó
```

```
[main 4ff40d5] Se actualizó la conclusión y el resumen con los resultados y se comen
 3 files changed, 5 insertions(+)
 create mode 100644 introduccion.txt
```

```
1 !git status -s
```

```
1 !git status
```

```
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Ignorar archivos

Pg. 29

En el archivo `.gitignore` se colocan los archivos a ignorar

Ver los Cambios Preparados y No Preparados

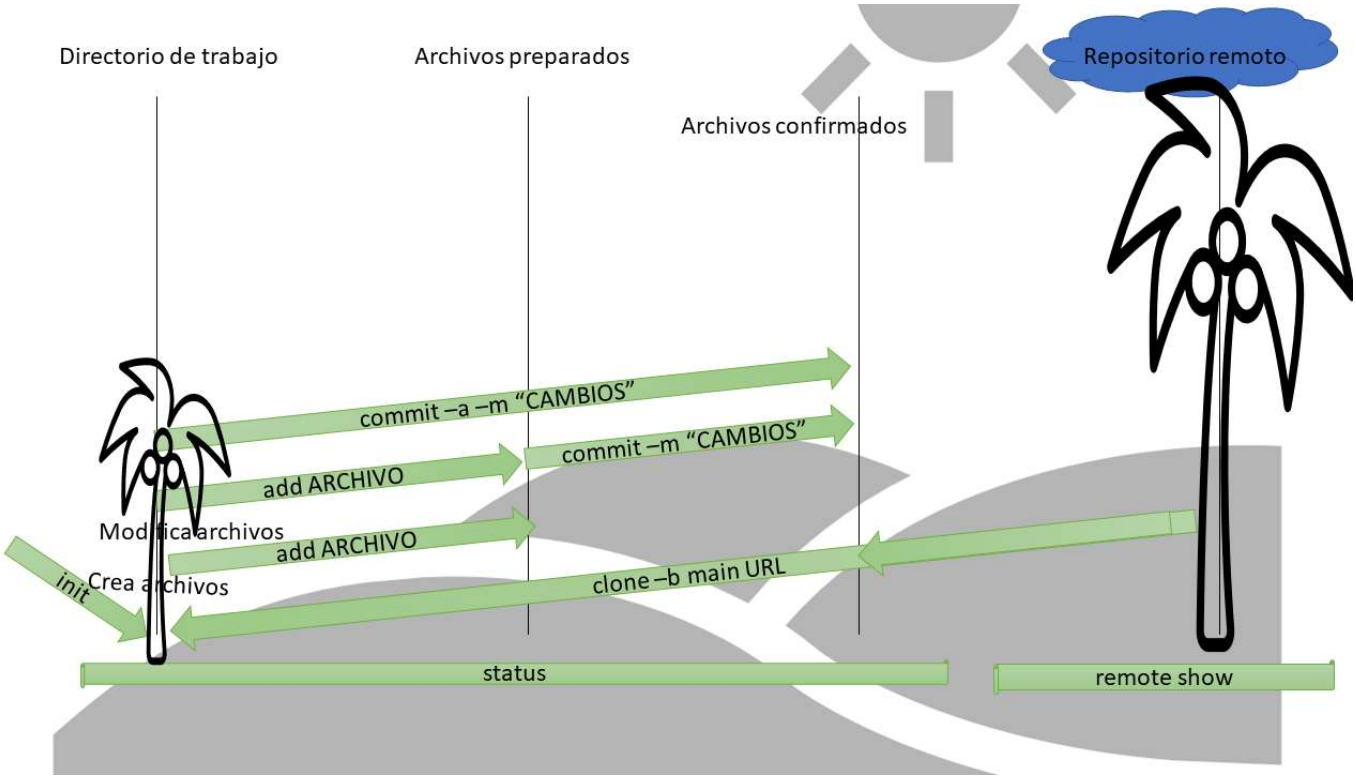
Pg. 30

```
git diff --staged
```

Saltar el Área de Preparación

Pg. 34

```
git commit -a -m "added new benchmarks"
```



Eliminar Archivos

Pg. 35

```
git rm un_archivo
```

Cambiar el Nombre de los Archivos

Pg. 36

```
git mv file_from file_to
```

Ver el Historial de Confirmaciones

Pg. 37

```
1 !dir
```

```
conclusion.txt  introduccion.txt  README.md  resumen.txt
```

```
1 !git log
```

```
commit 4ff40d5cf8d522323ddb4183376c058217a214e0 (HEAD -> main)
Author: cambiar-por-su-nombre <cambiar.por.su.correo@gmail.com>
Date:   Thu Apr 15 03:11:12 2021 +0000

    Se actualizó la conclusión y el resumen con los resultados y se comenzó la introc
```

```
commit 16941cb648a2aa9b97ac927c8b8555f2a219bc4e
Author: cambiar-por-su-nombre <cambiar.por.su.correo@gmail.com>
Date:   Thu Apr 15 03:11:09 2021 +0000

    Versión inicial del proyecto

commit 112b9d9ea1a1b0b20411a9757833d3b7e6531188 (origin/main, origin/HEAD)
Author: cambiar-por-su-nombre <81787129+cambiar-por-su-nombre@users.noreply.github.c
Date:   Sun Apr 11 19:26:51 2021 -0500

    Create README.md
```

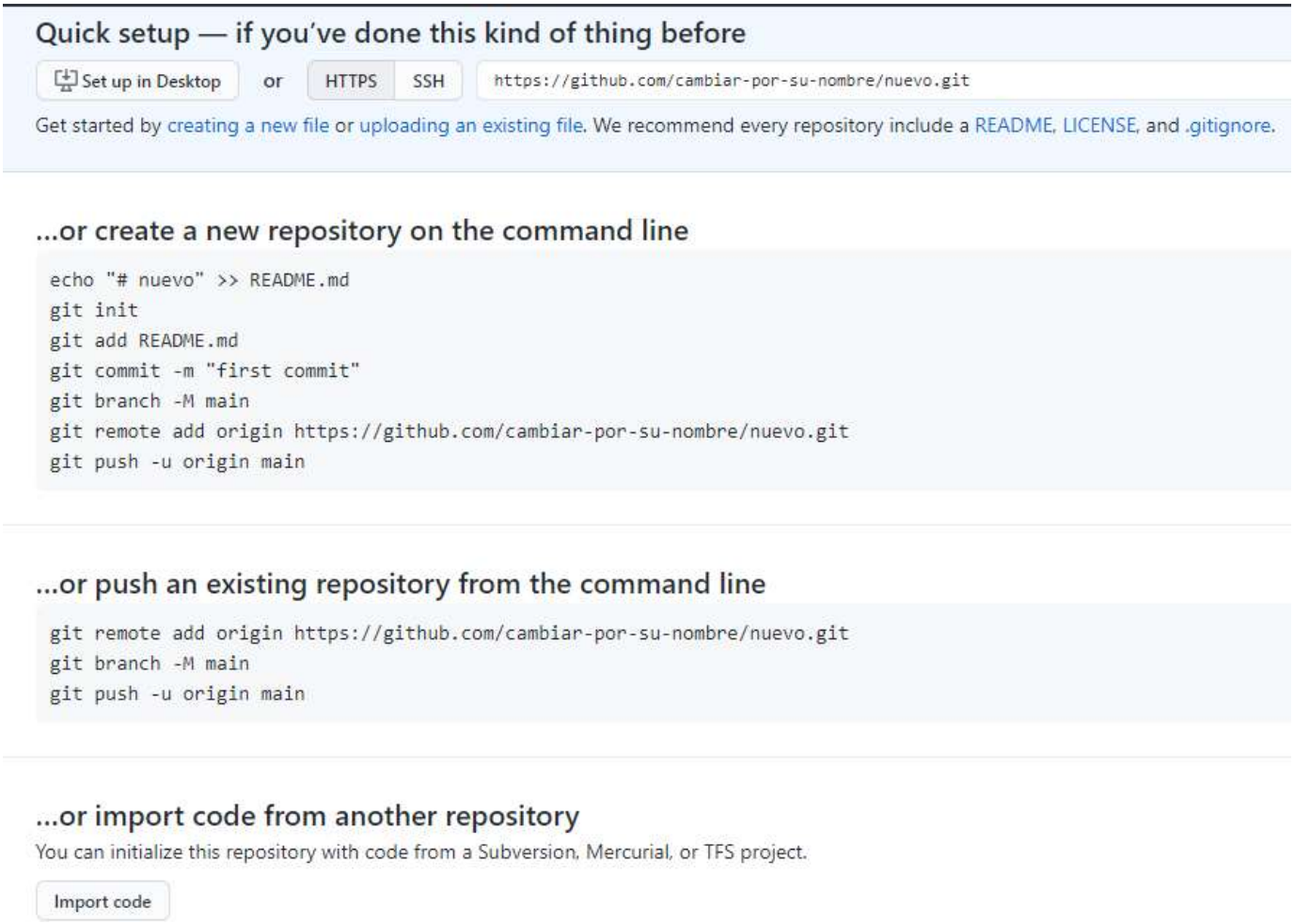
Deshacer Cosas

- `git commit --amend` permite sobrescribir la anterior confirmación.
- `git reset HEAD <file>` permite deshacer la preparación.
- `git checkout -- [archivo]` Se pierden los cambios que se han hecho en un archivo después de la última confirmación.

Trabajar con Remotos

Pg. 46

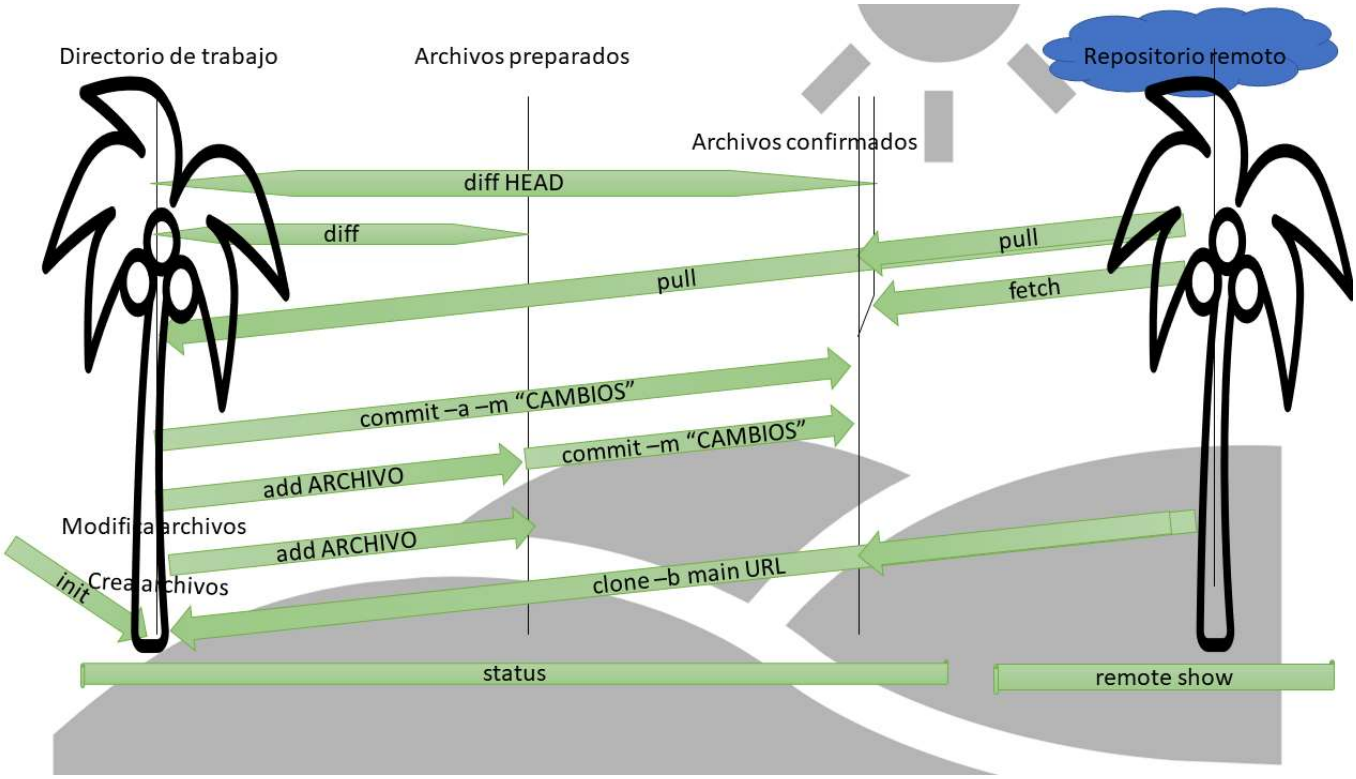
Al crear un nuevo repositorio remoto en GitHub, este ofrece una serie de posibilidades para conectarse con el repositorio local. En particular no ofrece clone ya que no hay nada en el remoto.



La sugerencia para este curso consiste en crear al menos el README.md en el repositorio remoto (GitHub) y luego clonarlo. Esto permite que el repositorio local quede configurado de una vez con el remoto.

Ver Tus Remotos

Pg. 47




```
1 !git remote
```

```
origin
```

```
1 !git remote -v
```

```
origin  https://github.com/cambiar-por-su-nombre/TrabajosProgramacion (fetch)
origin  https://github.com/cambiar-por-su-nombre/TrabajosProgramacion (push)
```

Traer y Combinar Remotos

Pg. 49

> `git fetch` solo trae datos a tu repositorio local - ni lo combina automáticamente con tu trabajo ni modifica el trabajo que llevas hecho. La combinación con tu trabajo debes hacerla manualmente cuando estés listo.

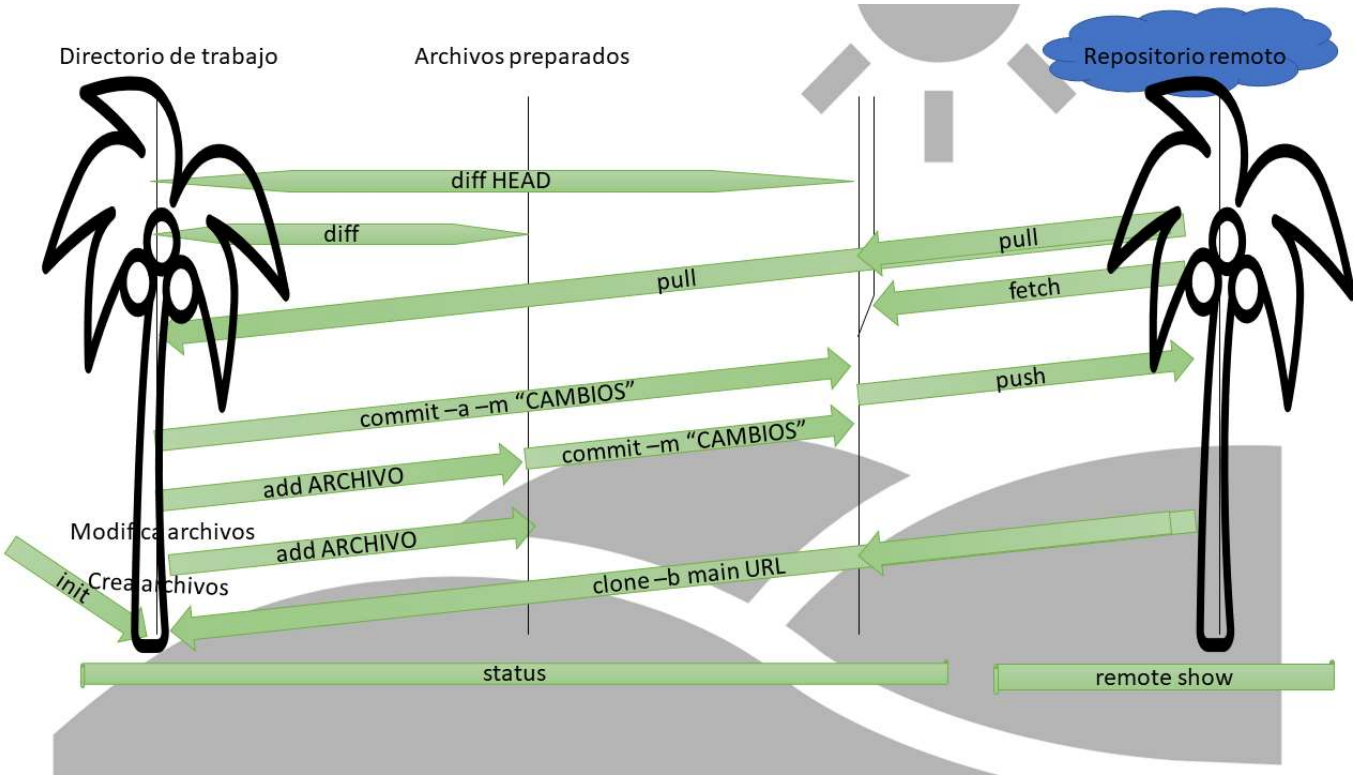
> Generalmente, al ejecutar `git pull` traerás datos del servidor del que clonaste originalmente y se intentará combinar automáticamente la información con el código en el que estás trabajando.

```
1 !git pull
```

```
Already up to date.
```

Enviar a Tus Remotos

Pg. 49




```
1 !git push
```

```
fatal: could not read Username for 'https://github.com': No such device or address
```

Inspeccionar un Remoto

Pg. 50

```
1 !git remote show
```

```
origin
```

```
1 !git remote show origin
```

```
* remote origin
  Fetch URL: https://github.com/cambiar-por-su-nombre/TrabajosProgramacion
  Push  URL: https://github.com/cambiar-por-su-nombre/TrabajosProgramacion
  HEAD branch: main
  Remote branch:
    main tracked
  Local branch configured for 'git pull':
    main merges with remote main
  Local ref configured for 'git push':
    main pushes to main (fast-forwardable)
```

Etiquetado

Pg 52

`git tag` Permite resaltar algunas confirmaciones verdaderamente importantes.

Alias de Git

Pg. 56

Permite dar nombres cortos a comandos de git.

```
1
```