# Hoja de ruta GIT

### Este documento ejemplifica el flujo de trabajo con git, es decir, aquellas tareas más usuales desde que comienzas a programar hasta que te sientas.

### Downloads

<https://git-scm.com/downloads>

## **Instalando Git**

<https://git-scm.com/book/es/v1/Empezando-Instalando-Git>

### Tutorial / curso git&github

<https://try.github.io/levels/1/challenges/1>

## 2.1 Fundamentos de Git - Obteniendo un repositorio Git

<https://git-scm.com/book/es/v1/Empezando-Configurando-Git-por-primera-vez>

Crear el repositorio

$ mkdir repo

$ cd repo/

**Inicializar git**

repo$ git init

Initialized empty Git repository in /home/david/Escritorio/cursos ED/GIT/repo/.git/

$ ls -a

. .. .git

Crear el fichero README

repo$ gedit README.md &

**Observar el estado del directorio de trabajo y el area de stage:**

repo$ git status

En la rama master

Commit inicial

Archivos sin seguimiento:

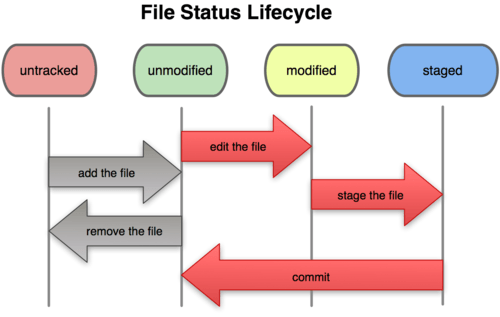
(use «git add <archivo>...» para incluir lo que se ha de ejecutar)

README.md

no se ha agregado nada al commit pero existen archivos sin seguimiento (use «git add» para darle seguimiento)

## 2.2 Git Basics - Recording Changes to the Repository

<https://git-scm.com/book/es-ni/v1/Git-Basics-Recording-Changes-to-the-Repository>



**Buscar comandos:**

$ git help

$ git help status

**o ayuda en línea:**

<https://git-scm.com/> e introducir el comando en el cuadro de búsqueda

git status:

<https://git-scm.com/docs/git-status>

repo$ git status -s

?? README.md

· ' ' = unmodified

· M = modified

· A = added

· D = deleted

· R = renamed

· C = copied

· U = updated but unmerged

X Y Meaning

-------------------------------------------------

[MD] not updated

M [ MD] updated in index

A [ MD] added to index

D [ M] deleted from index

R [ MD] renamed in index

C [ MD] copied in index

[MARC] index and work tree matches

[ MARC] M work tree changed since index

[ MARC] D deleted in work tree

-------------------------------------------------

D D unmerged, both deleted

A U unmerged, added by us

U D unmerged, deleted by them

U A unmerged, added by them

D U unmerged, deleted by us

A A unmerged, both added

U U unmerged, both modified

-------------------------------------------------

? ? untracked

! ! ignored

-------------------------------------------------

**Pasar ficheros al área de STAGE = comenzar el seguimiento de ficheros = grabar cambios**

repo$ git add README.md

repo$ git status

En la rama master

Commit inicial

Cambios para hacer commit:

(use «git rm --cached <archivo>...« para eliminar stage)

new file: README.md

repo$ git status -s

A README.md

----------------------------------------

**Modifico un fichero que ya está en el área staged:**

repo$ gedit README.md &

repo$ git status -s

AM README.md

repo$ git diff

diff --git a/README.md b/README.md

index 1f7c5d5..1187a07 100644

--- a/README.md

+++ b/README.md

@@ -1,4 +1,4 @@

-Código ejercicios Python

+Código ejercicios Python -modificado

========================

( Q )

**COMMIT changes:**

OJO: así sólo he hecho commit de la versión de README.md que está en el STAGE area.

repo$ git commit -m "Crear fichero README en markdown"

[master (root-commit) c0ad5fc] Crear fichero README en markdown

1 file changed, 22 insertions(+)

create mode 100644 README.md

repo$ git log

commit c0ad5fc72e67dfea3138084628a5fc5dc7c57a2d

Author: David <gelpiorama@gmail.com>

Date: Wed Nov 11 12:54:39 2015 +0100

repo$ git log --tags

commit c0ad5fc72e67dfea3138084628a5fc5dc7c57a2d

Author: David <gelpiorama@gmail.com>

Date: Wed Nov 11 12:54:39 2015 +0100

Crear fichero README en markdown

**Si quiero descartar todos los cambios hechos DESDE el último commit:**

$ git checkout --README.md

y ahora:

repo$ git status

En la rama master

nothing to commit, working directory clean

----------------------------------------------------------------

Modificar un fichero en seguimiento:

repo$ gedit README.md &

[1] 25838

david@HiPerX:~/Escritorio/cursos ED/GIT/repo$ git status -s

M README.md

[1]+ Hecho gedit README.md

repo$ git status

En la rama master

Cambios no preparados para el commit:

(use «git add <archivo>...» para actualizar lo que se ejecutará)

(use «git checkout -- <archivo>...« para descartar cambios en le directorio de trabajo)

modificado: README.md

no hay cambios agregados al commit (use «git add» o «git commit -a»)

**Diferencias entre los ficheros en el commit/staged y el directorio de trabajo:**

repo$ git diff

diff --git a/README.md b/README.md

index 1f7c5d5..04343ee 100644

--- a/README.md

+++ b/README.md

@@ -20,3 +20,4 @@ Los ejercicios no están resueltos de la forma más eficiente, s

4. [Code Complete, Second Edition.](http://cc2e.com/)

Si encuentras errores o deseas hacer sugerencias, ya sabes: commit!

+IES Francesc de Borja Moll - Palma

Diferencias entre los ficheros en el commit y el staged:

repo$ git diff --staged

nada que mostrar

Pero si añadimos al STAGE un fichero:

repo$ git add README.md

repo$ git status -s

M README.md

repo$ git status

En la rama master

Cambios para hacer commit:

(use «git reset HEAD <archivo>...«para eliminar stage)

modificado: README.md

una vez añadido el fichero al stage área:

repo$ git diff

nada que mostrar => no hay diferencias entre el staged area y el directorio de trabajo

pero ahora sí hay diferencias entre el commit y el staged area

repo$ git diff --staged

diff --git a/README.md b/README.md

index 1f7c5d5..04343ee 100644

--- a/README.md

+++ b/README.md

@@ -20,3 +20,4 @@ Los ejercicios no están resueltos de la forma más eficiente, s

4. [Code Complete, Second Edition.](http://cc2e.com/)

Si encuentras errores o deseas hacer sugerencias, ya sabes: commit!

+IES Francesc de Borja Moll - Palma

**Diferencias entre dos commits:**

$ git diff c2e338cf3beeb4bf057489e2b6cd84fc0f41f062 a3d4b05f658c3ef382a545106f0264c92e743401

**DESHACIENDO COSAS:**

<https://git-scm.com/book/es/v1/Fundamentos-de-Git-Deshaciendo-cosas>

Para sacar al fichero README.md del área de STAGE:

repo$ git status

En la rama master

Cambios para hacer commit:

(use «git reset HEAD <archivo>...«para eliminar stage)

modificado: README.md

repo$ git reset HEAD README.md

Unstaged changes after reset:

M README.md

y volvemos a tener README en el directorio de trabajo -con seguimiento- pero fuera del stage.

**Si quiero descartar todos los cambios hechos DESDE el último commit en README.md**

$ git checkout --README.md

**Crear un fichero, meterlo en el commit y luego eliminarlo del commit:**

<https://git-scm.com/book/es-ni/v1/Git-Basics-Recording-Changes-to-the-Repository>

Creo el fichero:

repo$ touch tocino.txt

repo$ ls

README.md tocino.txt

repo$ git status -s

M README.md => con seguimiento, en el directorio de trabajo, modificado respecto al commit

?? tocino.txt => sin seguimiento

Lo añado al stage:

repo$ git add tocino.txt

repo$ git status -s

M README.md

A tocino.txt

Los añado todos:

**repo$ git add .**

repo$ git status -s

M README.md

A tocino.txt

repo$ git diff

nada que mostrar: no hay diferencia entre el stage y el directorio de trabajo

repo$ git diff --staged => diferencia entre el commit y el staged

diff --git a/README.md b/README.md

index 1f7c5d5..04343ee 100644

--- a/README.md

+++ b/README.md

@@ -20,3 +20,4 @@ Los ejercicios no están resueltos de la forma más eficiente, s

4. [Code Complete, Second Edition.](http://cc2e.com/)

Si encuentras errores o deseas hacer sugerencias, ya sabes: commit!

+IES Francesc de Borja Moll - Palma

diff --git a/tocino.txt b/tocino.txt

new file mode 100644

index 0000000..e69de29

**Commit: -a => todos los cambios de todos los ficheros en seguimento (en el stage)**

repo$ git commit -a -m "Añadir a README instituto y añadir fichero tocino"

[master a7ccb77] Añadir a README instituto y añadir fichero tocino

2 files changed, 1 insertion(+)

create mode 100644 tocino.txt

repo$ git status

En la rama master

nothing to commit, working directory clean

repo$ git log

commit a7ccb77fb28db6f9197f0482f27eba680214bc2c

Author: David <gelpiorama@gmail.com>

Date: Wed Nov 11 13:44:01 2015 +0100

Añadir a README instituto y añadir fichero tocino

commit c0ad5fc72e67dfea3138084628a5fc5dc7c57a2d

Author: David <gelpiorama@gmail.com>

Date: Wed Nov 11 12:54:39 2015 +0100

Crear fichero README en markdown

Me arrepiento y quiero eliminar el fichero tocino del commit.

**1º) Elimina el archivo del directorio de trabajo:**

repo$ rm tocino.txt

Ahora, git informa de que hemos eliminado un fichero y que debemos llevar este cambio al stage area y así prepararlo para el siguiente commit:

repo$ git status

En la rama master

Cambios no preparados para el commit:

(use «git add/rm <archivo>...» para actualizar lo que se ejecutará)

(use «git checkout -- <archivo>...« para descartar cambios en el directorio de trabajo)

deleted: tocino.txt

no hay cambios agregados al commit (use «git add» o «git commit -a»)

repo$ git status -s

D tocino.txt

**2º) Así que ahora llevamos este cambio -la eliminación de tocino- al stage area:**

repo$ git rm tocino.txt

rm 'tocino.txt'

repo$ git status

En la rama master

Cambios para hacer commit:

(use «git reset HEAD <archivo>...«para eliminar stage)

deleted: tocino.txt

repo$ git status -s

D tocino.txt

**este es el único cambio a llevar al commit:**

repo$ git commit -a -m "eliminar tocino.txt"

[master ff9104b] eliminar tocino.txt

1 file changed, 0 insertions(+), 0 deletions(-)

delete mode 100644 tocino.txt

**Veamos el histórico de commits con sus diferencias:**

<https://git-scm.com/book/es/v1/Fundamentos-de-Git-Viendo-el-hist%C3%B3rico-de-confirmaciones>

$ git log -p -2 => compara los dos últimos commits

**Listar los ficheros que forman parte de un commit:**

$ git ls-tree 5d093dfe1785bbaa72b65a02c1ac2283fc93f957

100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391 panceta.txt

100644 blob 9183da3f30eb50b4d5103b90462e97f69e7fde07 tocino.txt

Poner los ficheros en el estado en que estaban en un commit anterior:

$ git checkout a3d4b05f658c3ef382a545106f0264c92e743401

Note: checking out 'a3d4b05f658c3ef382a545106f0264c92e743401'.

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -b with the checkout command again. Example:

git checkout -b <new-branch-name>

HEAD se encuentra en a3d4b05... tocino una linea

Para volver a poner el HEAD en el último commit:

$ git checkout master

Previous HEAD position was a3d4b05... tocino una linea

Switched to branch 'master'

**Deshaciendo cosas**

<https://git-scm.com/book/en/v2/Git-Basics-Undoing-Things>

modifico un fichero

lo añado al staged

git add fichero

amend del commit

$ git commit -m 'initial commit'  
$ git add forgotten\_file  
$ git commit --amend

se abre el editor pico para permitirte editar el mensaje del último commit

you’re not so much fixing it as *replacing* it entirely with a new, improved commit that pushes the old commit out of the way

The obvious value to amending commits is to make minor improvements to your last commit, without cluttering your repository history with commit messages of the form, “Oops, forgot to add a file” or “Darn, fixing a typo in last commit”.

**Renombrar un fichero en seguimiento**

$ git mv README.txt README

esto es equivalente a ejecutar algo así:

$ mv README.txt README  
$ git rm README.txt  
$ git add README

La única diferencia real es que mv es un comando en vez de tres