

19 de enero 2022

Mentoría: Introducción a GIT

Alumna: Jérica Edith Tapia Reyes

[ada_hack77](#)

¿Investigar qué es git?

¹Git es un software de [control de versiones](#) diseñado por [Linus Torvalds](#), pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de [código fuente](#).

¿Para qué sirve?

Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

Entre las características más relevantes se encuentran:

- Fuerte apoyo al desarrollo no lineal, por ende rapidez en la gestión de ramas y mezclado de diferentes versiones. Git incluye herramientas específicas para navegar y visualizar un historial de desarrollo no lineal. Una presunción fundamental en Git, es que un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente, conforme se pasa entre varios programadores que lo revisan.
- Gestión distribuida. Al igual que [Darcs](#), [BitKeeper](#), [Mercurial](#), [SVK](#), [Bazaar](#) y [Monotone](#), Git le da a cada programador una copia local del historial del desarrollo entero, y los cambios se propagan entre los [repositorios](#) locales. Los cambios se importan como ramas adicionales y pueden ser fusionados en la misma manera que se hace con la rama local.
- Los almacenes de información pueden publicarse por [HTTP](#), [FTP](#), [rsync](#) o mediante un protocolo nativo, ya sea a través de una conexión TCP/IP simple o a través de cifrado [SSH](#). Git también puede emular servidores [CVS](#), lo que habilita el uso de clientes CVS pre-existentes y módulos IDE para CVS pre-existentes en el acceso de repositorios Git.
- Los repositorios Subversion y svk se pueden usar directamente con git-svn.
- Gestión eficiente de proyectos grandes, dada la rapidez de gestión de diferencias entre archivos, entre otras mejoras de optimización de velocidad de ejecución.
- Todas las versiones previas a un cambio determinado, implican la notificación de un cambio posterior en cualquiera de ellas a ese cambio (denominado autenticación criptográfica de historial). Esto existía en [Monotone](#).
- Resulta algo más caro trabajar con ficheros concretos frente a proyectos, eso diferencia el trabajo frente a CVS, que trabaja con base en cambios de fichero, pero mejora el trabajo con afectaciones de código que concurren en operaciones similares en varios archivos.
- Los renombrados se trabajan basándose en similitudes entre ficheros, aparte de nombres de ficheros, pero no se hacen marcas explícitas de cambios de nombre con base en supuestos nombres únicos de nodos de

¹ Fuente: <https://es.wikipedia.org/wiki/Git>

sistema de ficheros, lo que evita posibles y desastrosas coincidencias de ficheros diferentes en un único nombre.

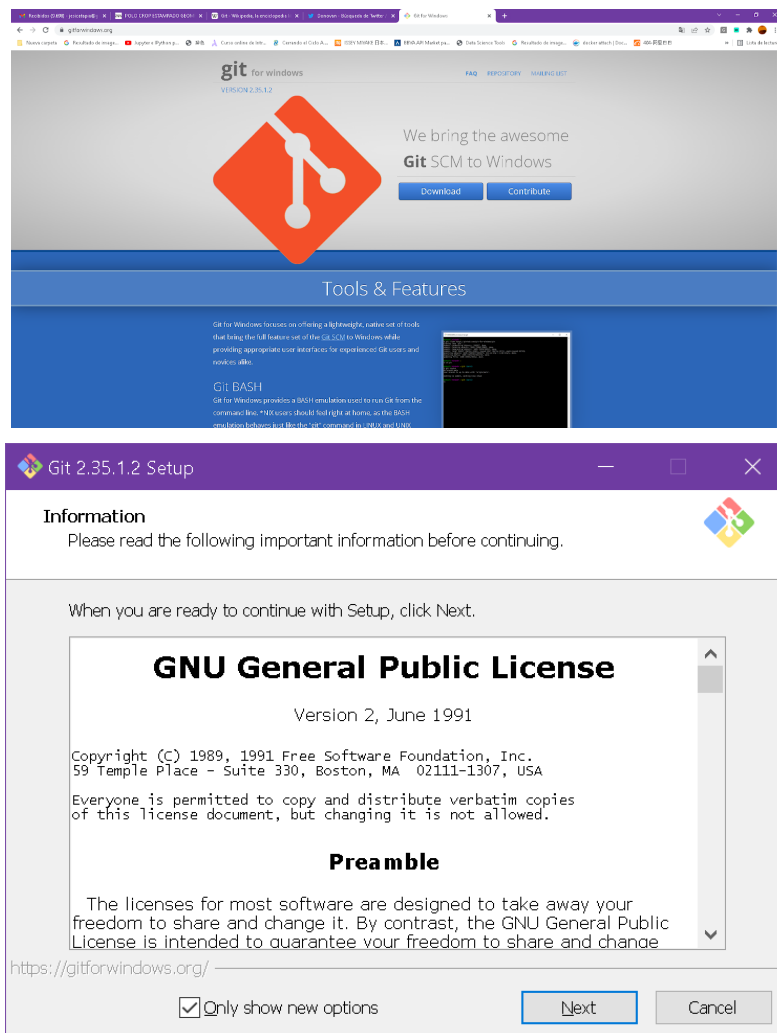
- Realmacenamiento periódico en paquetes (ficheros). Esto es relativamente eficiente para escritura de cambios y relativamente ineficiente para lectura si el reempaquetado (con base en diferencias) no ocurre cada cierto tiempo.
- Compatibilidad con Github y Microsoft Visual Studio Code

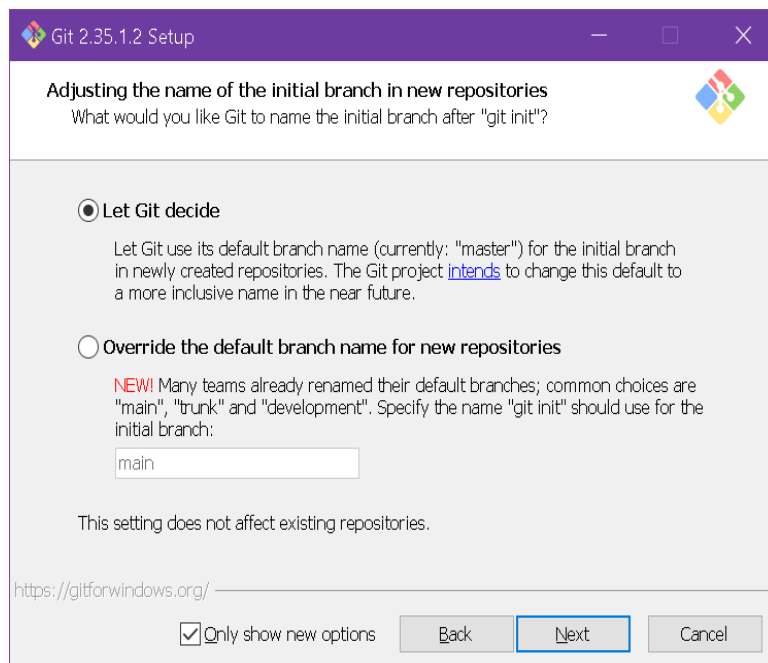
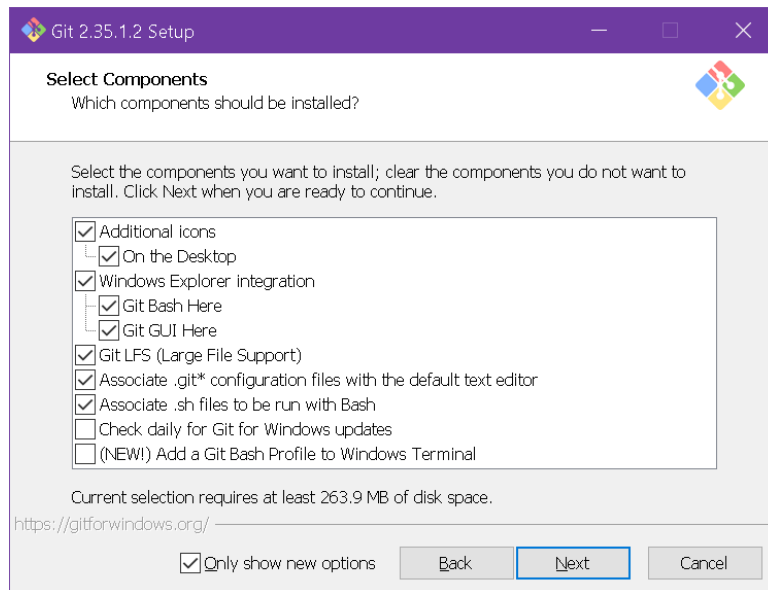
Instalar GIT

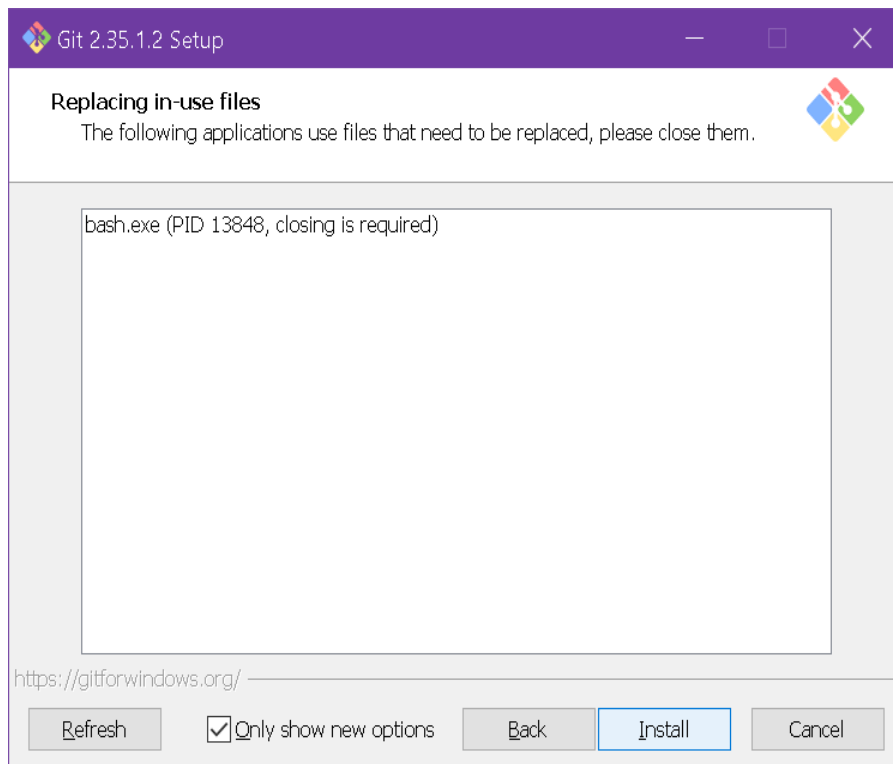
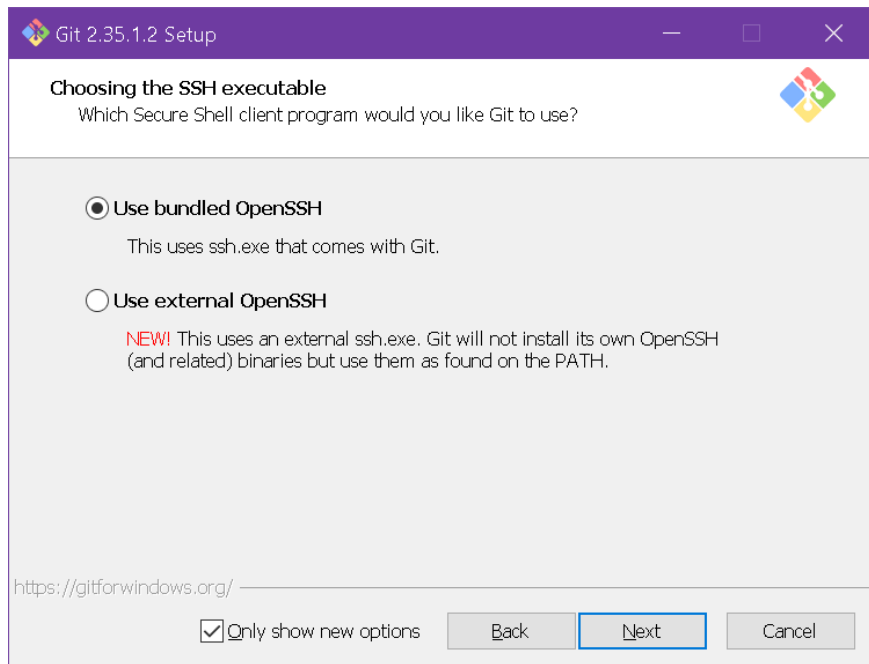
Una guía práctica de instalación

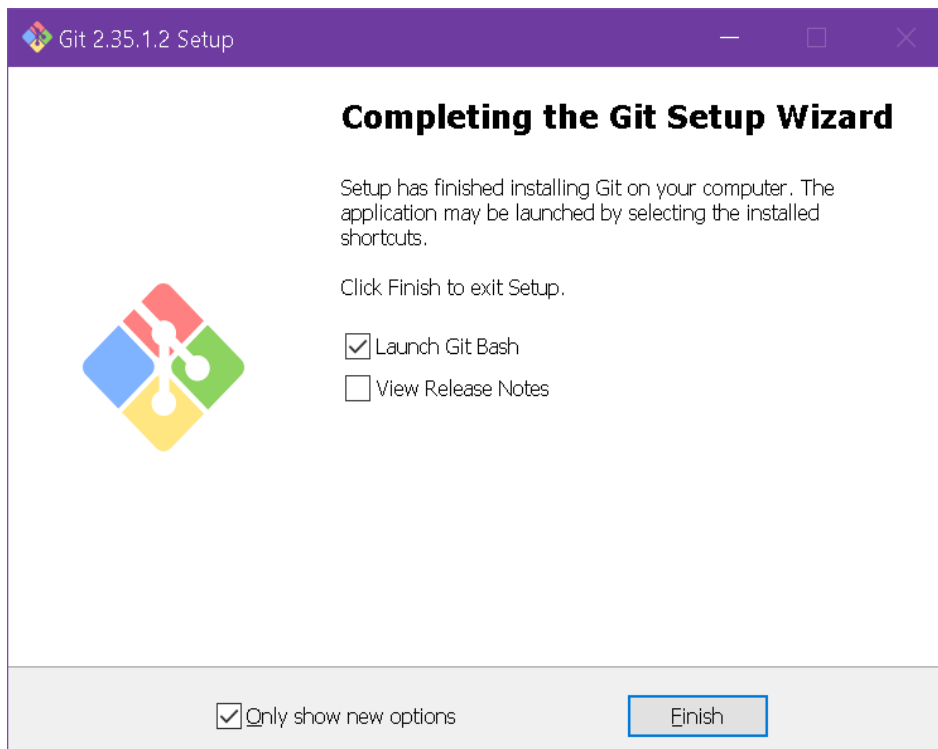
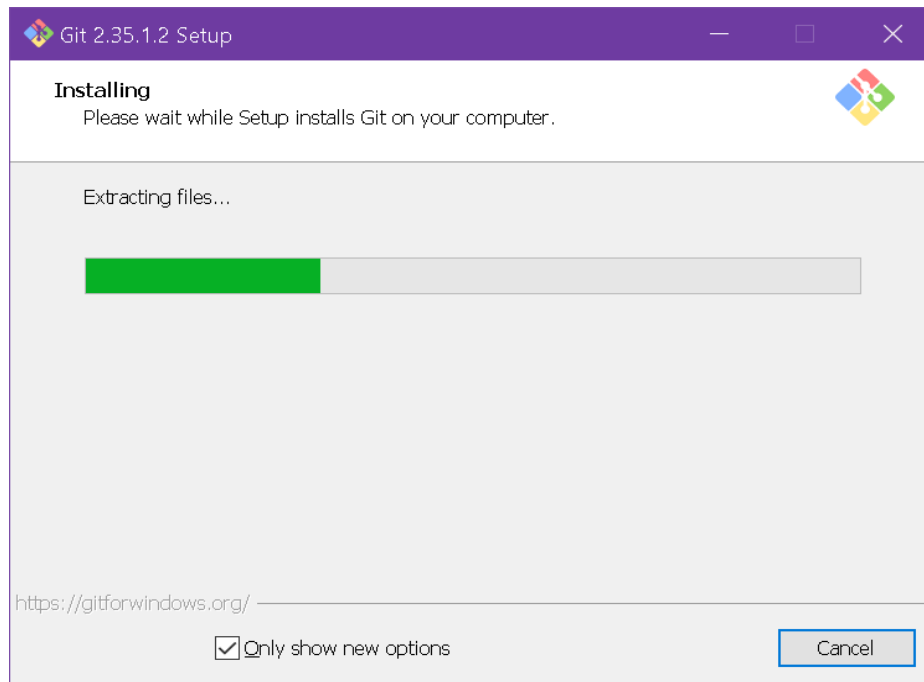
<https://rogerdudler.github.io/git-guide/index.es.html>

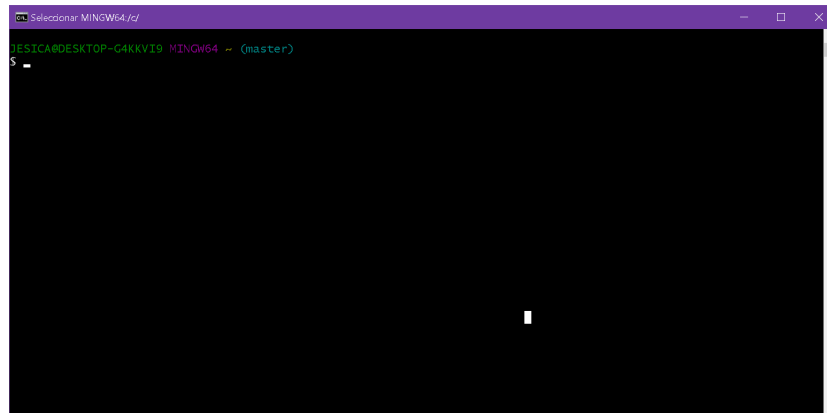
1.-Descargar Git para tu sistema operativo





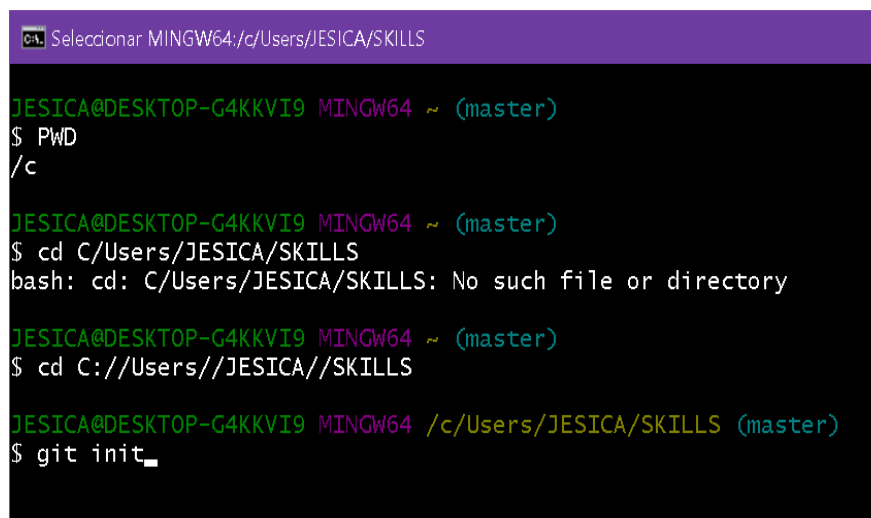




A terminal window titled 'Selecionar MINGW64/q' with a dark background. The prompt shows the user 'JESICA@DESKTOP-G4KKVI9' in a 'MINGW64' environment on the '(master)' branch. The prompt is '\$ _'.

Moverle a GIT

git init: Crear tu primer repositorio

A terminal window titled 'C:\ Selecionar MINGW64:/c/Users/JESICA/SKILLS' with a dark background. The prompt shows the user 'JESICA@DESKTOP-G4KKVI9' in a 'MINGW64' environment on the '(master)' branch. The user enters '\$ PWD' and the output is '/c'. Then the user enters '\$ cd C/Users/JESICA/SKILLS' and the output is 'bash: cd: C/Users/JESICA/SKILLS: No such file or directory'. Then the user enters '\$ cd C://Users//JESICA//SKILLS' and the prompt changes to 'JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)'. Finally, the user enters '\$ git init'.

```
C:\> Selecionar MINGW64:/c/Users/JESICA/SKILLS

JESICA@DESKTOP-G4KKVI9 MINGW64 ~ (master)
$ PWD
/c

JESICA@DESKTOP-G4KKVI9 MINGW64 ~ (master)
$ cd C:/Users/JESICA/SKILLS
bash: cd: C:/Users/JESICA/SKILLS: No such file or directory

JESICA@DESKTOP-G4KKVI9 MINGW64 ~ (master)
$ cd C://Users//JESICA//SKILLS

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ git init
Initialized empty Git repository in C:/Users/JESICA/SKILLS/.git/

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$
```

Con **mkdir** creamos una nueva carpeta y con **touch** creamos el archivo **README.md**

```
C:\> MINGW64:/c/Users/JESICA/SKILLS/PROYECTO1

JESICA@DESKTOP-G4KKVI9 MINGW64 ~ (master)
$ cd C:/Users/JESICA/SKILLS
bash: cd: C:/Users/JESICA/SKILLS: No such file or directory

JESICA@DESKTOP-G4KKVI9 MINGW64 ~ (master)
$ cd C://Users//JESICA//SKILLS

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ git init
Initialized empty Git repository in C:/Users/JESICA/SKILLS/.git/

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ mkdir PROYECTO1

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ LS
NUEVONOMBRE.txt  NUEVONOMBRE.txtEXIT  PROYECTO1

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ cd C://Users//JESICA//SKILLS//PROYECTO01
bash: cd: C://Users//JESICA//SKILLS//PROYECTO01: No such file or directory

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ cd C://Users//JESICA//SKILLS//PROYECTO1

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ touch README.md
```

- podemos modificar README.md con nano

```
MINGW64:/c/Users/JESICA/SKILLS/PROYECTO1
JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ git init
Initialized empty Git repository in C:/Users/JESICA/SKILLS/.git/

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$
JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ mkdir PROYECTO1

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ ls
NUEVONOMBRE.txt  NUEVONOMBRE.txtEXIT  PROYECTO1

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ cd C://Users//JESICA//SKILLS//PROYECTO01
bash: cd: C://Users//JESICA//SKILLS//PROYECTO01: No such file or directory

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ cd C://Users//JESICA//SKILLS//PROYECTO1

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ touch README.md

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ ls
README.md

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ nano README.md
```

- Guardemos el estado del repositorio utilizando **git add**
El proceso de una nueva versión se divide en tres partes
 - 1) Repositorio Local
 - 2) Índice -stage area - área de preparación, allí seleccionas las cosas que si vas a guardar
 - 3) Versión Guardada
- Para añadir todos los archivos usamos **git add nombre archivo1/nombearchivo 2**

Una forma más rápida
git add .

```
MINGW64:/c/Users/JESICA/SKILLS/PROYECTO1
JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ git init
Initialized empty Git repository in C:/Users/JESICA/SKILLS/.git/

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$
JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ mkdir PROYECTO1

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ ls
NUEVONOMBRE.txt  NUEVONOMBRE.txtEXIT  PROYECTO1

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ cd C://Users//JESICA//SKILLS//PROYECTO01
bash: cd: C://Users//JESICA//SKILLS//PROYECTO01: No such file or directory

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS (master)
$ cd C://Users//JESICA//SKILLS//PROYECTO1

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ touch README.md

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ ls
README.md

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ git add .
```


- queremos saber el estado de nuestro repositorio
git status

```

C:\Users\JESICA\SKILLS\PROYECTO1
$ cd C://Users//JESICA//SKILLS//PROYECTO1

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ touch README.md

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ ls
README.md

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ git add .

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md

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

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$

```

- **git commit** recibe algunos parámetros
- **-m** escribimos mensaje que nos ayuda a identificar de qué se trata el cambio

```

C:\Users\JESICA\SKILLS\PROYECTO1
$ cd C://Users//JESICA//SKILLS//PROYECTO1

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ touch README.md

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ ls
README.md

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ git add .

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md

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

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ git commit -m incluí solo el README.m

```

```
C:\. Selecionar MINGW64:/c/Users/JESICA/SKILLS/PROYECTO1

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ git commit -m "primer commit solo incluí el README.MD"
[master (root-commit) d69fa06] primer commit solo incluí el README.MD
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 PROYECTO1/README.md

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ _
```

El comando me regresa información sobre la rama en la que se hizo el commit (master), el hash **d69fa06**, el mensaje que puse y cuántos archivos cambiaron.

¿Cómo puedo ver mis commits en terminal?
utilizamos **git log**

```
C:\. MINGW64:/c/Users/JESICA/SKILLS/PROYECTO1

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ git commit -m "primer commit solo incluí el README.MD"
[master (root-commit) d69fa06] primer commit solo incluí el README.MD
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 PROYECTO1/README.md

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ git log
commit d69fa06dac7c23ae792bef90e8a28ed510a426d6 (HEAD -> master)
Author: unknown <jesicatapia@gmail.com>
Date:   Wed Feb 9 21:53:11 2022 -0600

    primer commit solo incluí el README.MD

JESICA@DESKTOP-G4KKVI9 MINGW64 /c/Users/JESICA/SKILLS/PROYECTO1 (master)
$ _
```

git restore

nos regresa a la versión anterior del repositorio

git commit -a

Es un atajo para no ejecutar commit y add por separado, hacemos commit y añadimos todos los archivos

Recibidos

DreamHost

Compartir

Document

Docker can

The docker

circunflejo

JesTapia/Ho

+

github.com/JesTapia/Hola-Mundo-SFWT

Nueva carpetaResultado de image...Jupyter e iPython p...紫色Curso online de Intr...Cerrando el Ciclo A...ISSEY MIYAKE 日本...BBVA API Market pa...Lista de lectura

JesTapia Initial commit

9dd1c66 18 days ago 1 commit

README.md

Initial commit 18 days ago

README.md

Hola-Mundo-SFWT

Primer repositorio Skills for Women in Tech

Primer repositorio Skills for Women in Tech

Readme

0 stars

1 watching

0 forks

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

© 2022 GitHub, Inc.

[Terms](#)[Privacy](#)[Security](#)[Status](#)[Docs](#)[Contact GitHub](#)[Pricing](#)[API](#)[Training](#)[Blog](#)[About](#)