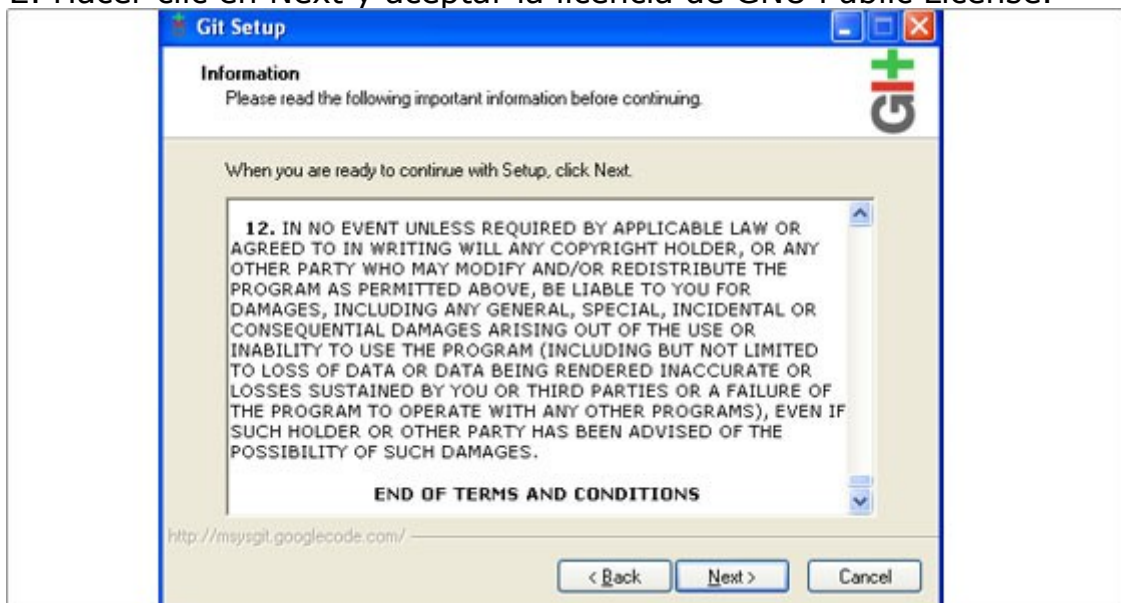


# 1. TUTORIAL DE INSTALACION DE GIT.

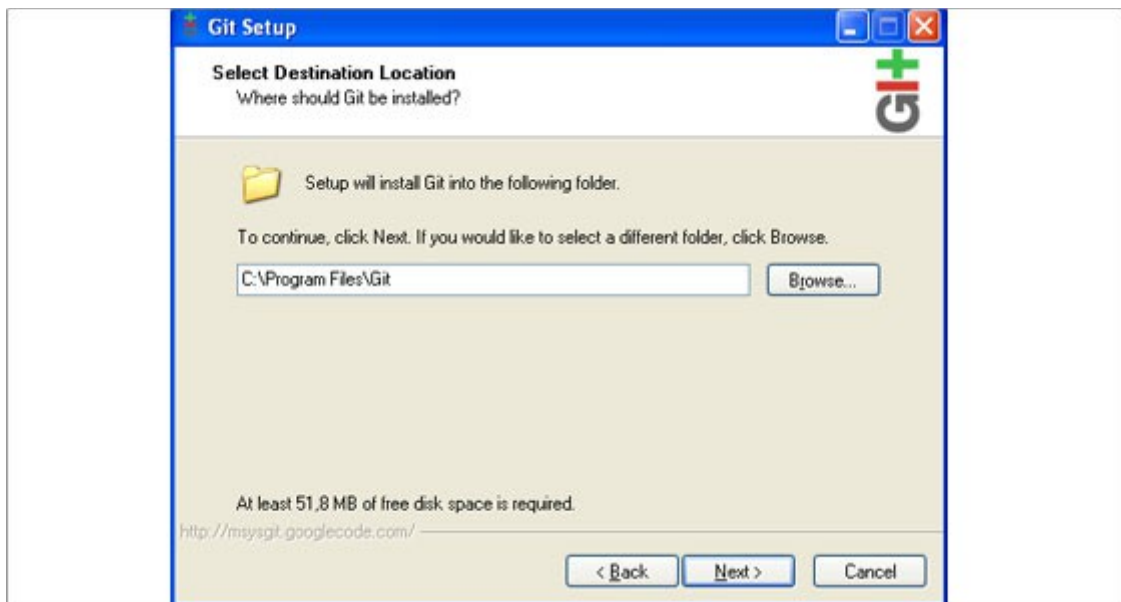
Paso 1. Ejecutar el archivo de instalación de Git.



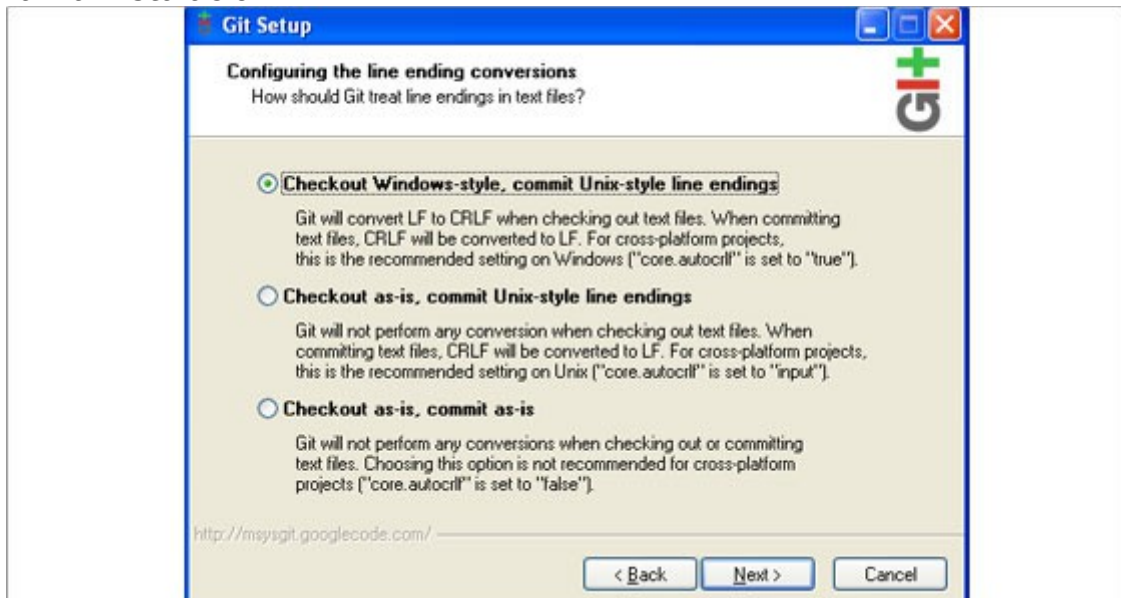
Paso 2. Hacer clic en Next y aceptar la licencia de GNU Public License.



Paso 3. Elegir carpeta destino para la instalación.

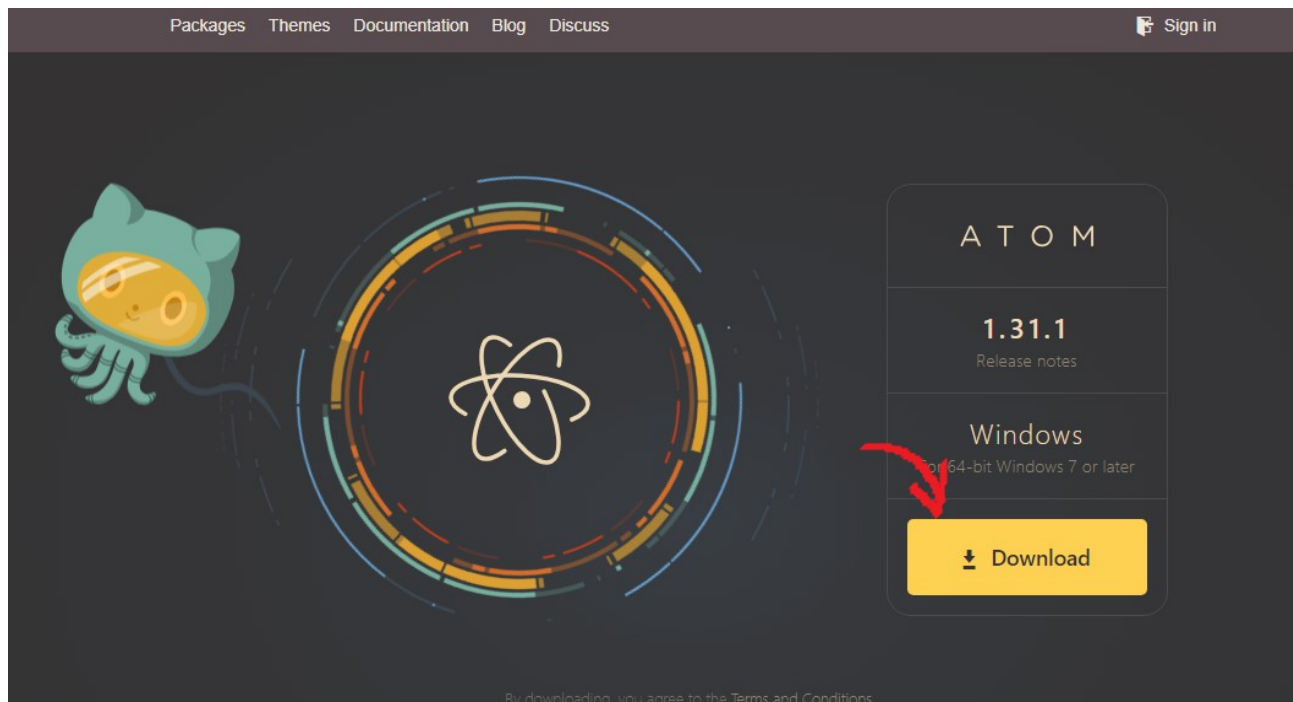


Paso 4. Dejar todas las opciones seleccionadas por defecto para Windows y finalizar la instalación.



## 2. TUTORIAL INSTALACION ATOM

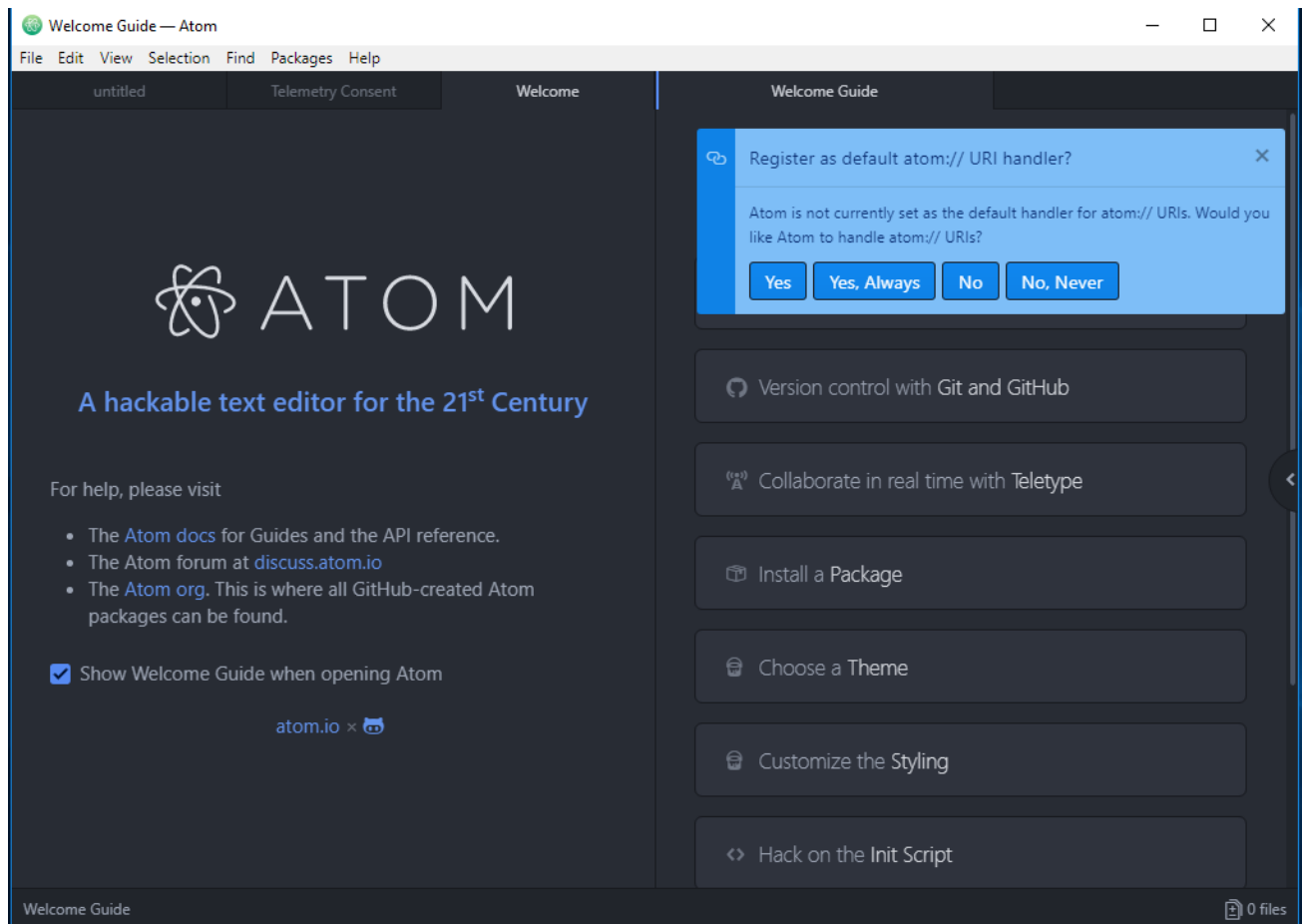
Paso 1. Descargar Atom de su página oficial.



Paso 2. Comenzar la instalación ejecutando el archivo descargado.

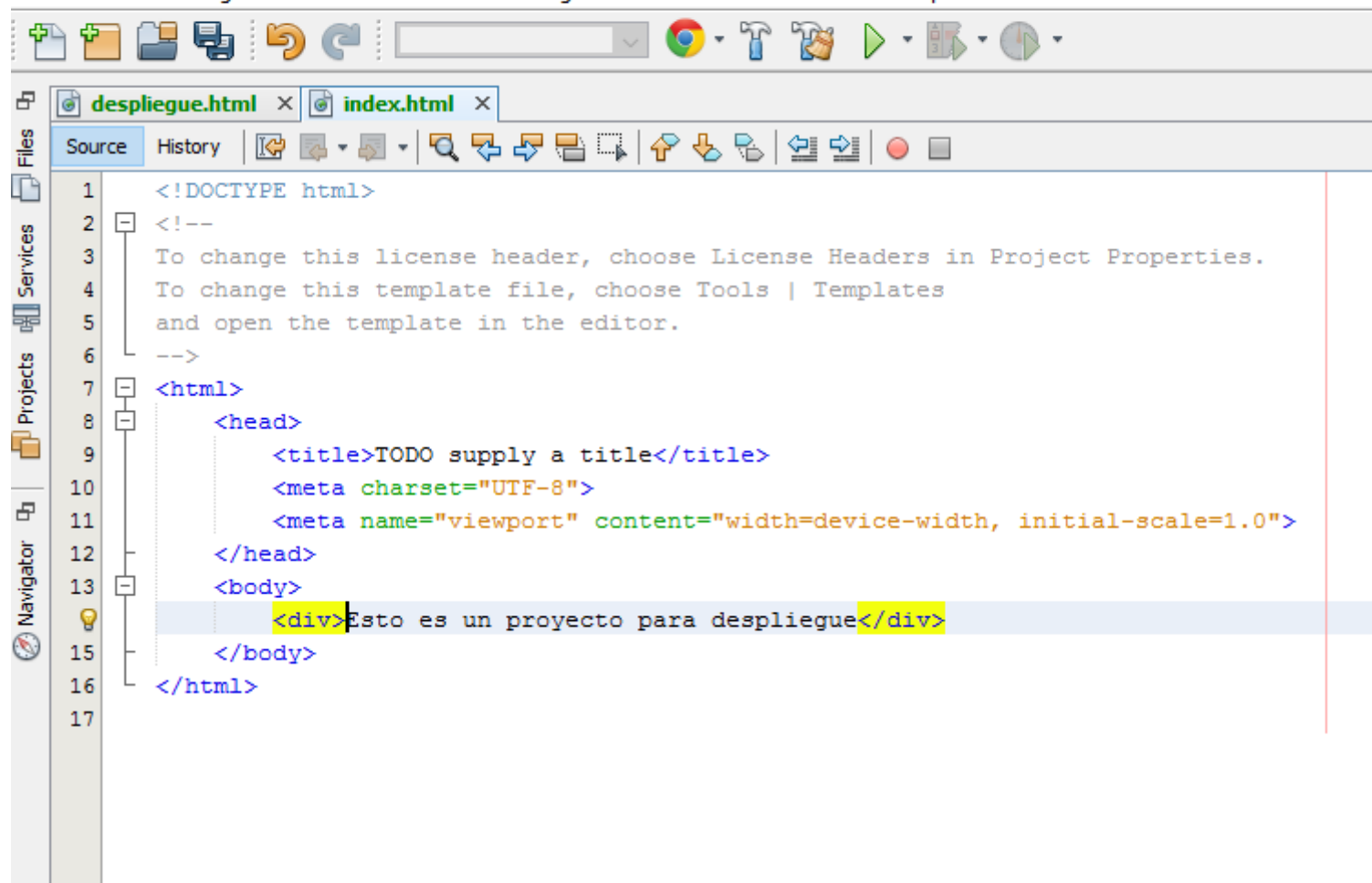


Paso 3. Ejecutar la aplicación para comprobar si se ha instalado correctamente y empezar a utilizarla.

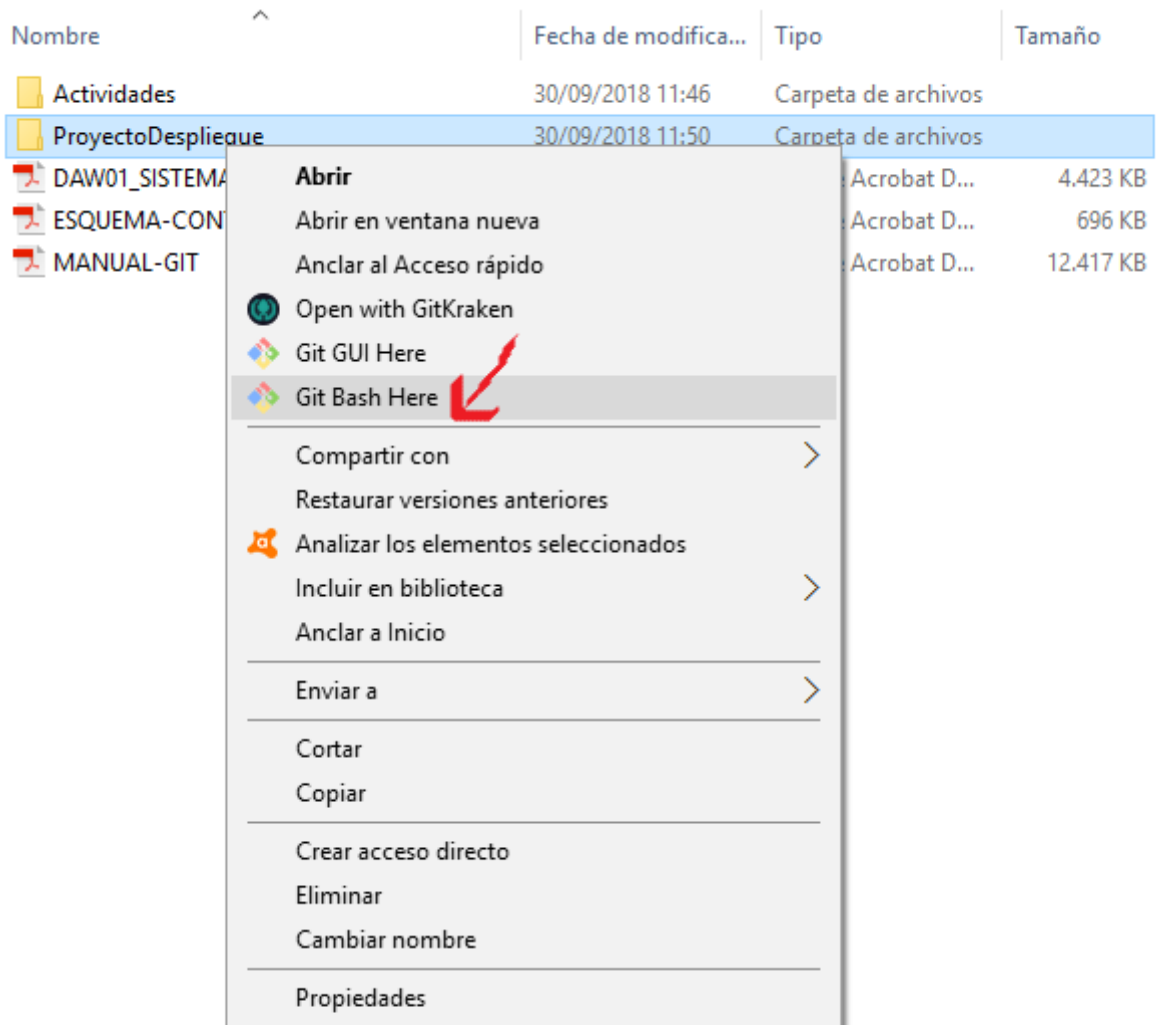


### 3. CREAR UN PROYECTO EN GIT

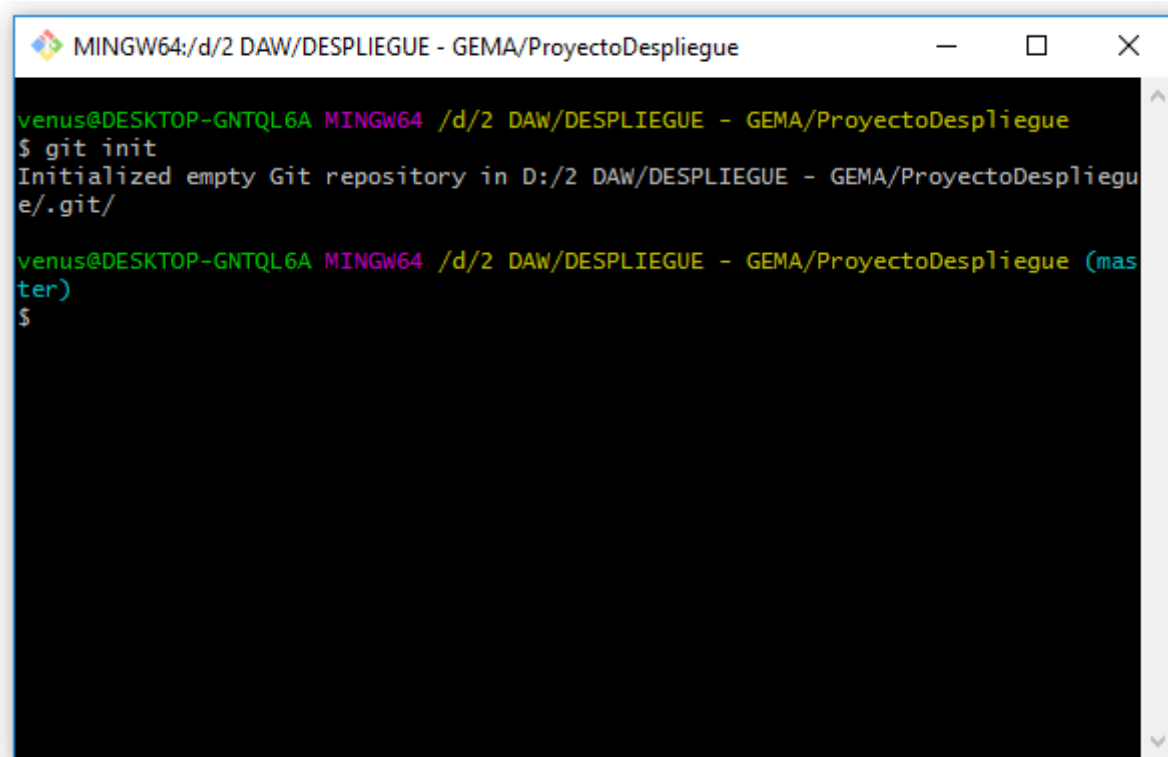
Paso 1. Creamos un nuevo proyecto en nuestro ordenador con el editor que estemos usando, en este caso NetBeans.



Paso 2. Iniciar la consola de Git sobre el proyecto creado.



Paso 3. Escribir la línea de comando "git init" en la consola para iniciar git sobre el proyecto.

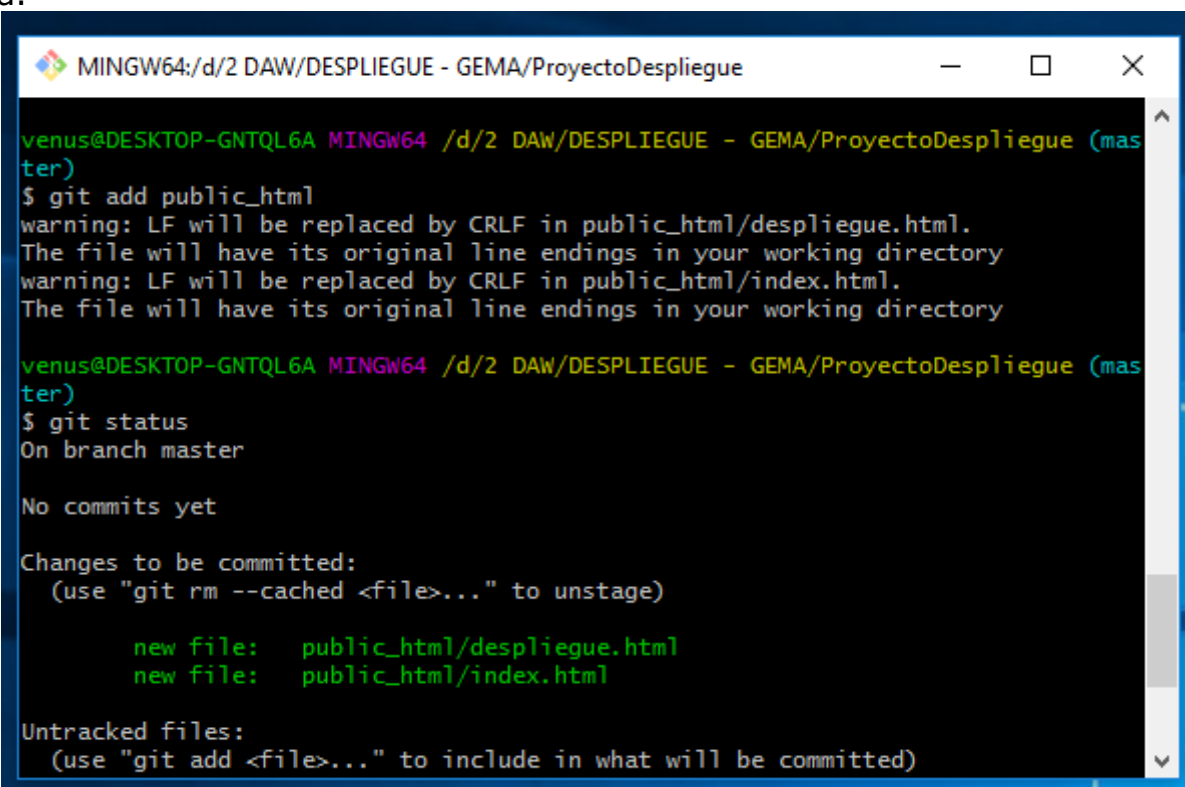


```
MINGW64:/d/2 DAW/DESPLIEGUE - GEMA/ProyectoDespliegue

venus@DESKTOP-GNTQL6A MINGW64 /d/2 DAW/DESPLIEGUE - GEMA/ProyectoDespliegue
$ git init
Initialized empty Git repository in D:/2 DAW/DESPLIEGUE - GEMA/ProyectoDespliegue/.git/

venus@DESKTOP-GNTQL6A MINGW64 /d/2 DAW/DESPLIEGUE - GEMA/ProyectoDespliegue (master)
$
```

Paso 4. Agregar los archivos al área de trabajo de Git utilizando el comando add.



```
MINGW64:/d/2 DAW/DESPLIEGUE - GEMA/ProyectoDespliegue

venus@DESKTOP-GNTQL6A MINGW64 /d/2 DAW/DESPLIEGUE - GEMA/ProyectoDespliegue (master)
$ git add public_html
warning: LF will be replaced by CRLF in public_html/despliegue.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in public_html/index.html.
The file will have its original line endings in your working directory

venus@DESKTOP-GNTQL6A MINGW64 /d/2 DAW/DESPLIEGUE - GEMA/ProyectoDespliegue (master)
$ git status
On branch master

No commits yet

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

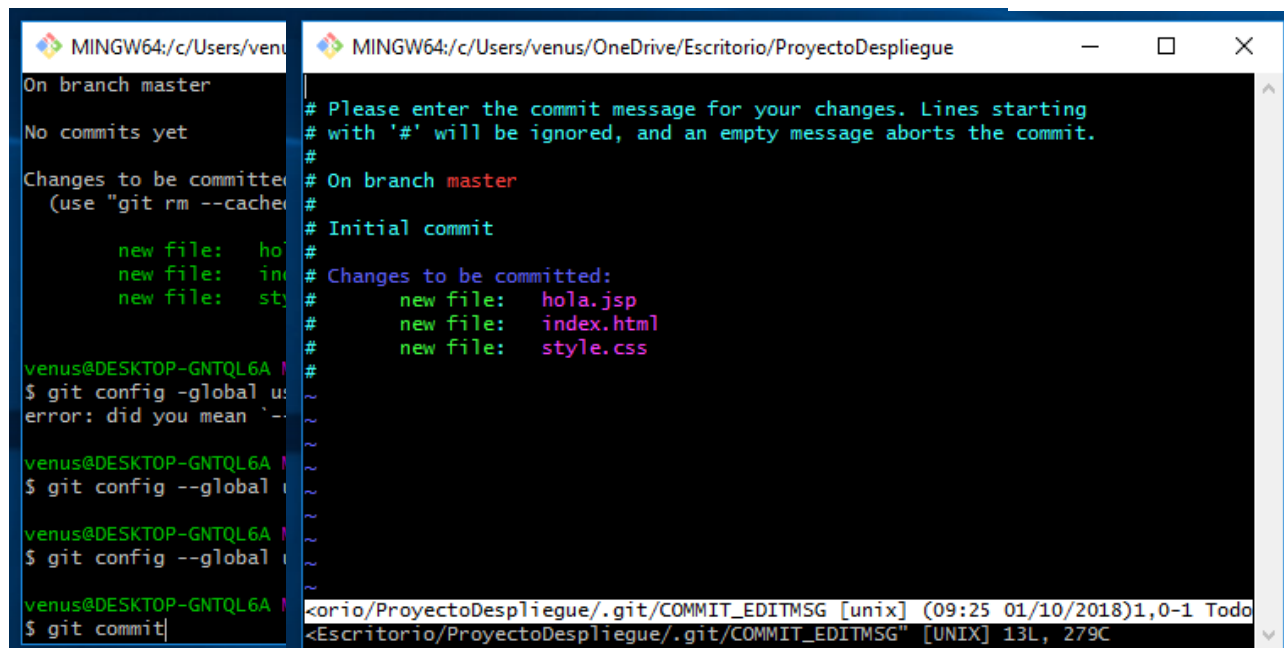
        new file:   public_html/despliegue.html
        new file:   public_html/index.html

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

Paso 5. Crear primer punto de partida. Para esto primero indicamos quien es el desarrollador del proyecto con los siguientes comandos:

- `git config --global user.email "pilardaw2@gmail.com"`. Este es el email que se va a asociar al usuario que ha hecho los cambios en el proyecto.
- `git config --global user.name "Pilar"`. Este es el nombre del usuario que

ha hecho los cambios en el proyecto.  
A continuación, escribimos el comando "git commit" y se abrirá un editor de código en el terminal.



```
MINGW64:/c/Users/venus/OneDrive/Escritorio/ProyectoDespliegue
On branch master
No commits yet
Changes to be committed:
  (use "git rm --cached" to unstage)
        new file:   hola.jsp
        new file:   index.html
        new file:   style.css

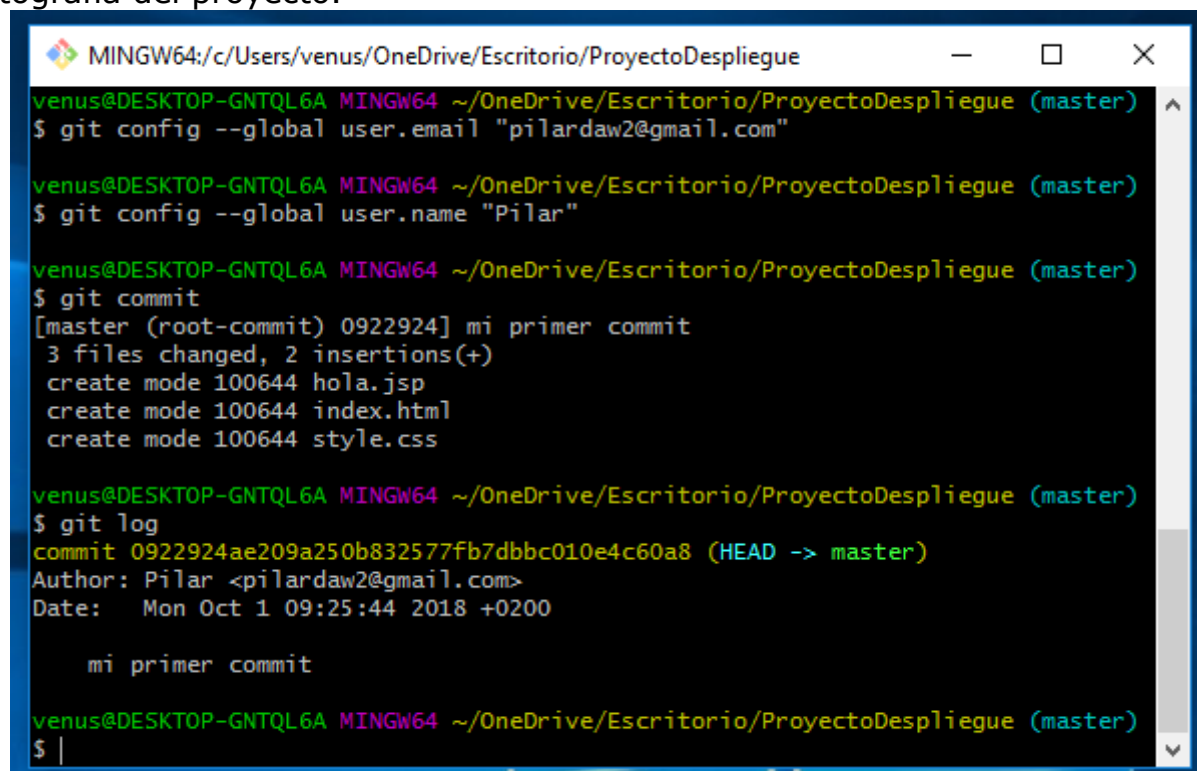
venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue
$ git config --global user.name "Pilar"
error: did you mean `--global`?

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue
$ git config --global user.email "pilardaw2@gmail.com"

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue
$ git commit

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   new file:   hola.jsp
#   new file:   index.html
#   new file:   style.css
#
<orio/ProyectoDespliegue/.git/COMMIT_EDITMSG [unix] (09:25 01/10/2018)1,0-1 Todo
<Escritorio/ProyectoDespliegue/.git/COMMIT_EDITMSG" [UNIX] 13L, 279C
```

Lo siguiente, para poder crear la primera fotografía del proyecto es pulsar la letra "i" para poder insertar texto y escribirle un título para hacer referencia a esa primera imagen (Ej: Mi primer commit). A continuación presionamos la letra ESC y escribimos :wq y nos muestra la información de la primera versión de nuestro proyecto. Si después de esto escribimos "git log" nos mostrara el registro de cambios que se han realizado, hasta ahora solamente la primera fotografía del proyecto.



```
MINGW64:/c/Users/venus/OneDrive/Escritorio/ProyectoDespliegue
venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$ git config --global user.email "pilardaw2@gmail.com"

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$ git config --global user.name "Pilar"

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$ git commit
[master (root-commit) 0922924] mi primer commit
3 files changed, 2 insertions(+)
create mode 100644 hola.jsp
create mode 100644 index.html
create mode 100644 style.css

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$ git log
commit 0922924ae209a250b832577fb7dbbc010e4c60a8 (HEAD -> master)
Author: Pilar <pilardaw2@gmail.com>
Date:   Mon Oct 1 09:25:44 2018 +0200

    mi primer commit

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$
```

Si después de todo esto escribimos en la consola "git status" nos mostrara que

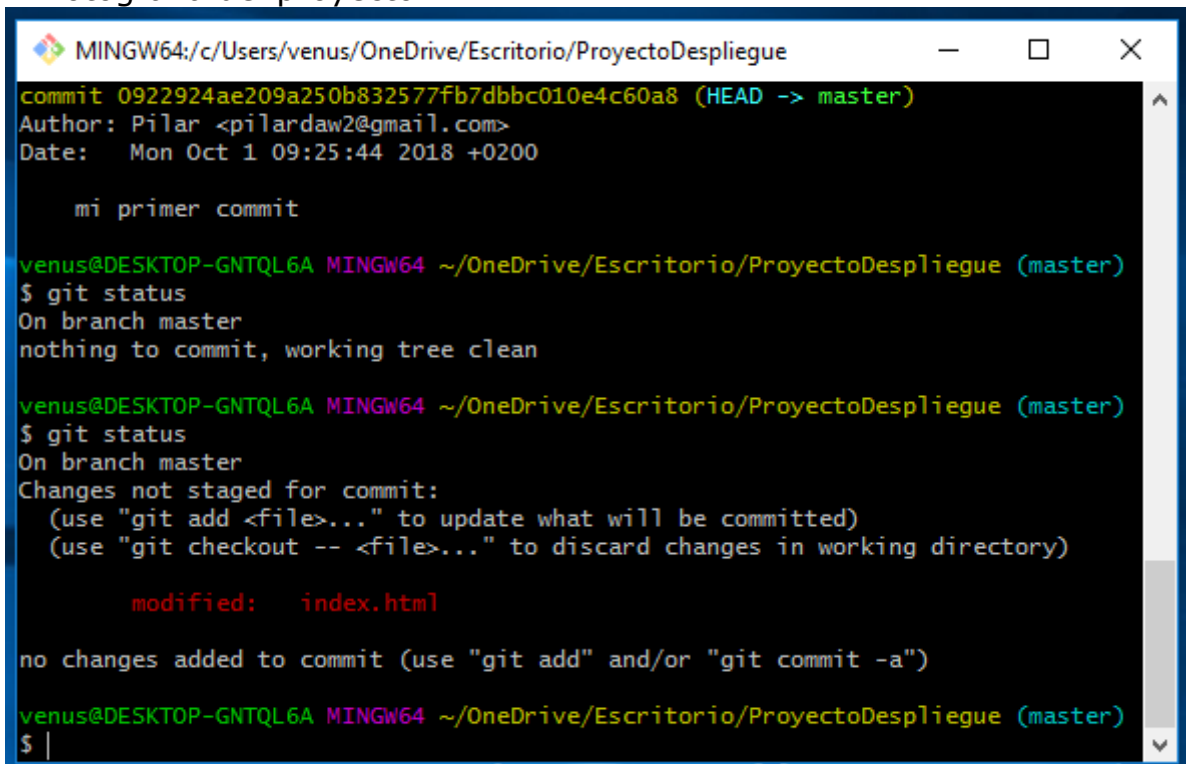


nuestra área de trabajo esta limpia, no hay archivos, es decir, la ultima versión de nuestro proyecto ha quedado registrada y al no haber realizado cambios no hay nada mas que registrar con Git.

#### 4. TRABAJAR CON GIT

- **Modificar un proyecto.**

Vamos a empezar a trabajar con Git realizando cambios en nuestro proyecto, por ejemplo, modificando el código html de uno de los archivos. Si después de haber realizado cambios en nuestro proyecto escribimos en la consola de comandos de Git el comando "git status" nos aparece que uno de los archivos de nuestro proyecto ha sido modificado y nos da la opción de descartar los cambios con el comando "git checkout --nombredelarchivo" y así volver a la ultima fotografía del proyecto. La otra opción es guardar los cambios creando una nueva fotografía del proyecto.



```
MINGW64:/c:/Users/venus/OneDrive/Escritorio/ProyectoDespliegue
commit 0922924ae209a250b832577fb7dbbc010e4c60a8 (HEAD -> master)
Author: Pilar <pilardaw2@gmail.com>
Date: Mon Oct 1 09:25:44 2018 +0200

    mi primer commit

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$ git status
On branch master
nothing to commit, working tree clean

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$ git status
On branch master
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:   index.html

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

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$ |
```

Antes de guardar una nueva fotografía del proyecto podemos ver las diferencias entre la ultima fotografía guardada y los nuevos cambios utilizando el comando "git diff nombre del archivo" y se mostraran los cambios que se ha producido. Si estamos conformes con los cambios añadimos el fichero con el comando add y volvemos a repetir la misma operación con git commit para guardar una nueva fotografía del proyecto. Si consultamos el log de ese proyecto nos aparecerán los cambios que se han ido registrando.

```
MINGW64:/c:/Users/venus/OneDrive/Escritorio/ProyectoDespliegue

modified: index.html

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$ git commit
[master c20878b] mi segundo commit
1 file changed, 7 insertions(+)

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$ git log
commit c20878b2756704d7cba6603f1def447edb21362e (HEAD -> master)
Author: Pilar <pilardaw2@gmail.com>
Date: Mon Oct 1 10:04:41 2018 +0200

    mi segundo commit

commit 0922924ae209a250b832577fb7dbbc010e4c60a8
Author: Pilar <pilardaw2@gmail.com>
Date: Mon Oct 1 09:25:44 2018 +0200

    mi primer commit

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$ |
```

- **Ignorar directorios y archivos.**

A veces, en nuestro proyecto tenemos directorios o archivos que no nos interesa guardar en nuestro entorno de trabajo de Git. Para que no nos aparezca en la consola que no han sido agregados cada vez que consultemos el status se pueden ignorar de la siguiente manera: creamos un archivo dentro de nuestro proyecto llamado ".gitignore" y en

```
MINGW64:/c:/Users/venus/OneDrive/Escritorio/ProyectoDespliegue
venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    .gitignore
    Ejercicios/
    test/

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

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue
$ git add .gitignore

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   .gitignore

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

    Ejercicios/

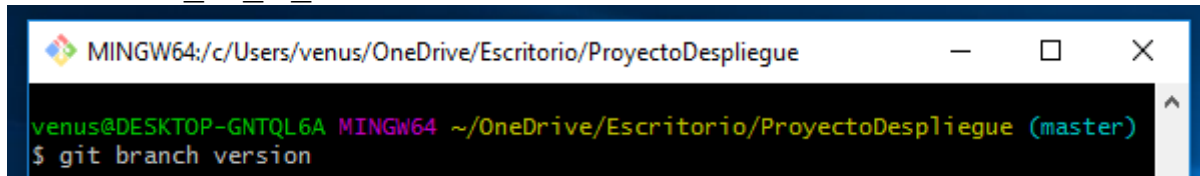
venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$
```

este archivo se almacenaran todos los archivos y carpetas que queremos que Git no refleje al consultar el status y que no están agregados a nuestro entorno de trabajo. Para finalizar agregamos el archivo ".gitignore" a nuestro entorno de trabajo y Git ignorará los archivos y carpetas especificados en ese archivo.

- **Crear versiones del proyecto.**

Por defecto en nuestro entorno de trabajo se crea una versión de nuestro proyecto y se almacena en una rama llamada master. Podemos crear diferentes versiones de nuestro proyecto creando diferentes ramas. Para

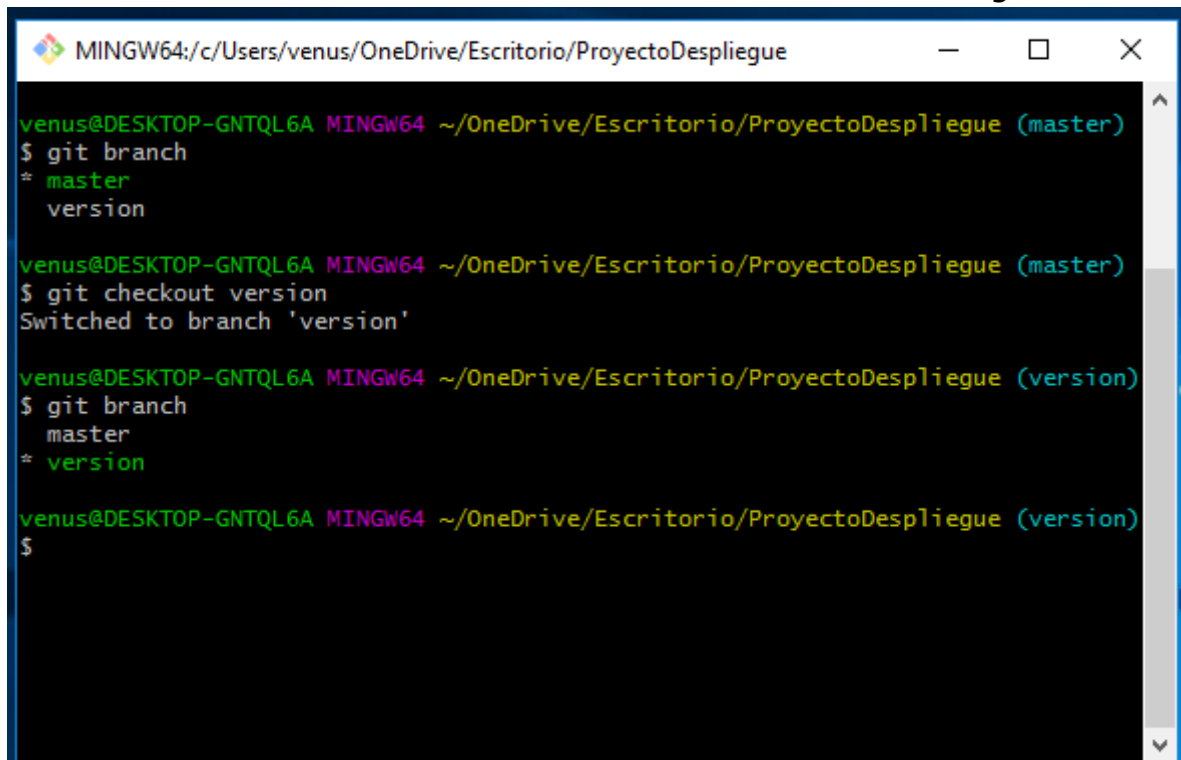
crear una nueva rama utilizaremos el comando "git branch nombre\_de\_la\_rama".



```
MINGW64:/c:/Users/venus/OneDrive/Escritorio/ProyectoDespliegue
venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$ git branch version
```

Si consultamos las ramas que hay en nuestro proyecto con el comando "git branch" nos aparecerá la que se crea por defecto "master" y la que nosotros hemos creado, en este caso "version".

Para acceder a esta nueva rama se hace con el comando "git checkout



```
MINGW64:/c:/Users/venus/OneDrive/Escritorio/ProyectoDespliegue
venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$ git branch
* master
  version

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (master)
$ git checkout version
Switched to branch 'version'

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (version)
$ git branch
* master
  version

venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDespliegue (version)
$
```

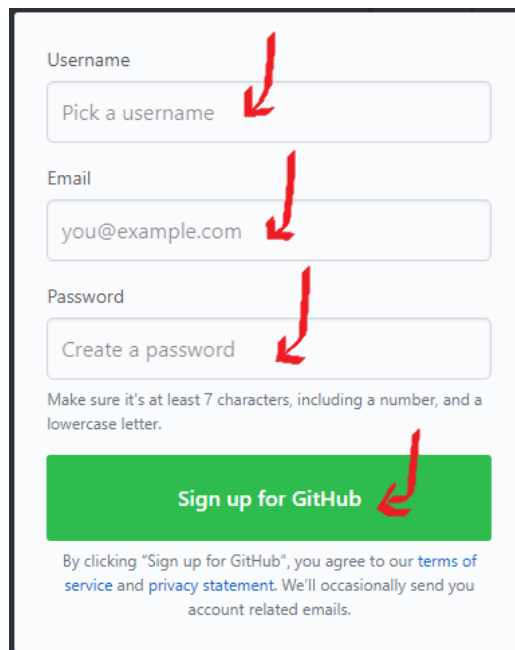
nombre\_de\_la\_rama" y si volvemos a consultar con "git branch" nos indicara en que rama nos encontramos coloreando dicha rama en verde.

Si a partir de aquí comenzamos a modificar nuestro proyecto, se guardaría en la nueva rama como una versión alternativa a nuestro proyecto original. Para volver al proyecto original habría que cambiar de rama. Con esto nos facilita el trabajo de realizar cambios teniendo varias versiones del mismo proyecto y no perder información.

## 5. GitHub

- **Registro.**

Accedemos a la pagina <https://github.com/> rellenamos el formulario con nuestros datos y hacemos clic sobre "Sign up for GitHub". Completamos todos los datos requeridos y finalizamos el proceso de registro.



Username

Pick a username

Email

you@example.com

Password

Create a password

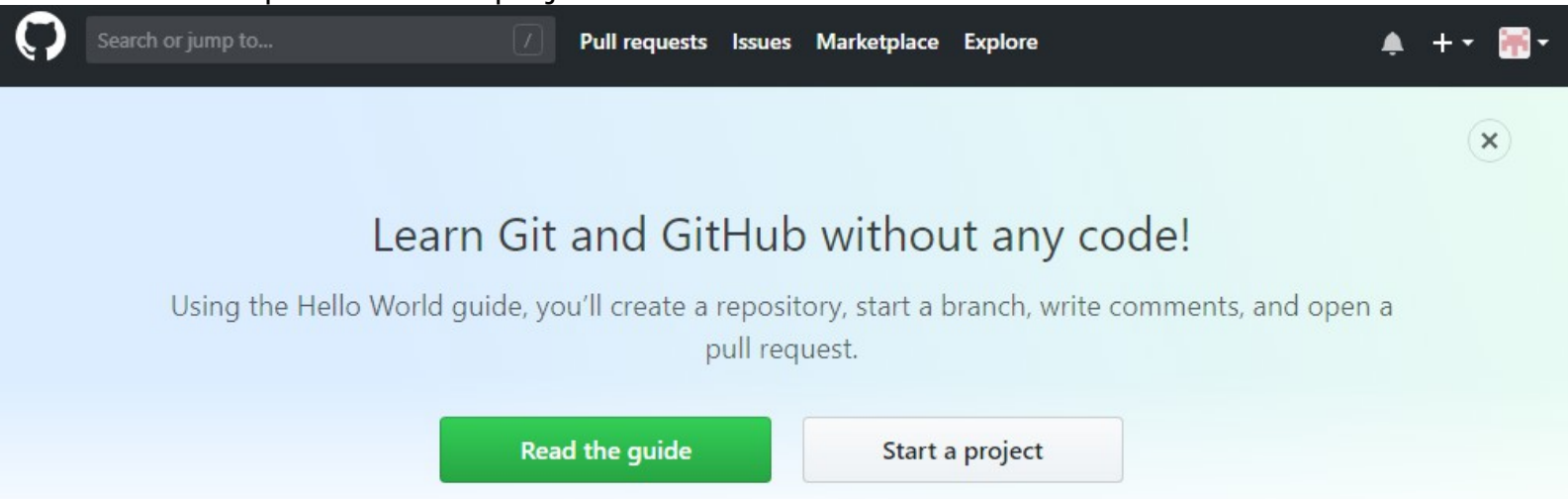
Make sure it's at least 7 characters, including a number, and a lowercase letter.

**Sign up for GitHub**

By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy statement](#). We'll occasionally send you account related emails.

- **Crear un repositorio.**

En la misma pagina de <https://github.com/> una vez finalizado el registro, nos identificamos y comenzamos a crear un nuevo repositorio con la opción "Start a project".



Search or jump to...

Pull requests Issues Marketplace Explore

Learn Git and GitHub without any code!



Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

**Read the guide** Start a project

A continuación, nos aparece una pagina en la que debemos introducir el nombre de nuestro proyecto y una breve descripción. Seleccionamos la opción "public" y hacemos clic en "Create repository".

Una vez hemos creado nuestro repositorio, en la pagina de nuestro repositorio nos aparece una dirección que es donde estará alojado y una serie de comandos para trabajar con el.


### Quick setup — if you've done this kind of thing before

 Set up in Desktop or **HTTPS** **SSH** <https://github.com/venusXV/ProyectoDesplieguePilar.git> 

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


### ...or create a new repository on the command line

```
echo "# ProyectoDesplieguePilar" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/venusXV/ProyectoDesplieguePilar.git
git push -u origin master
```



### ...or push an existing repository from the command line

```
git remote add origin https://github.com/venusXV/ProyectoDesplieguePilar.git
git push -u origin master
```

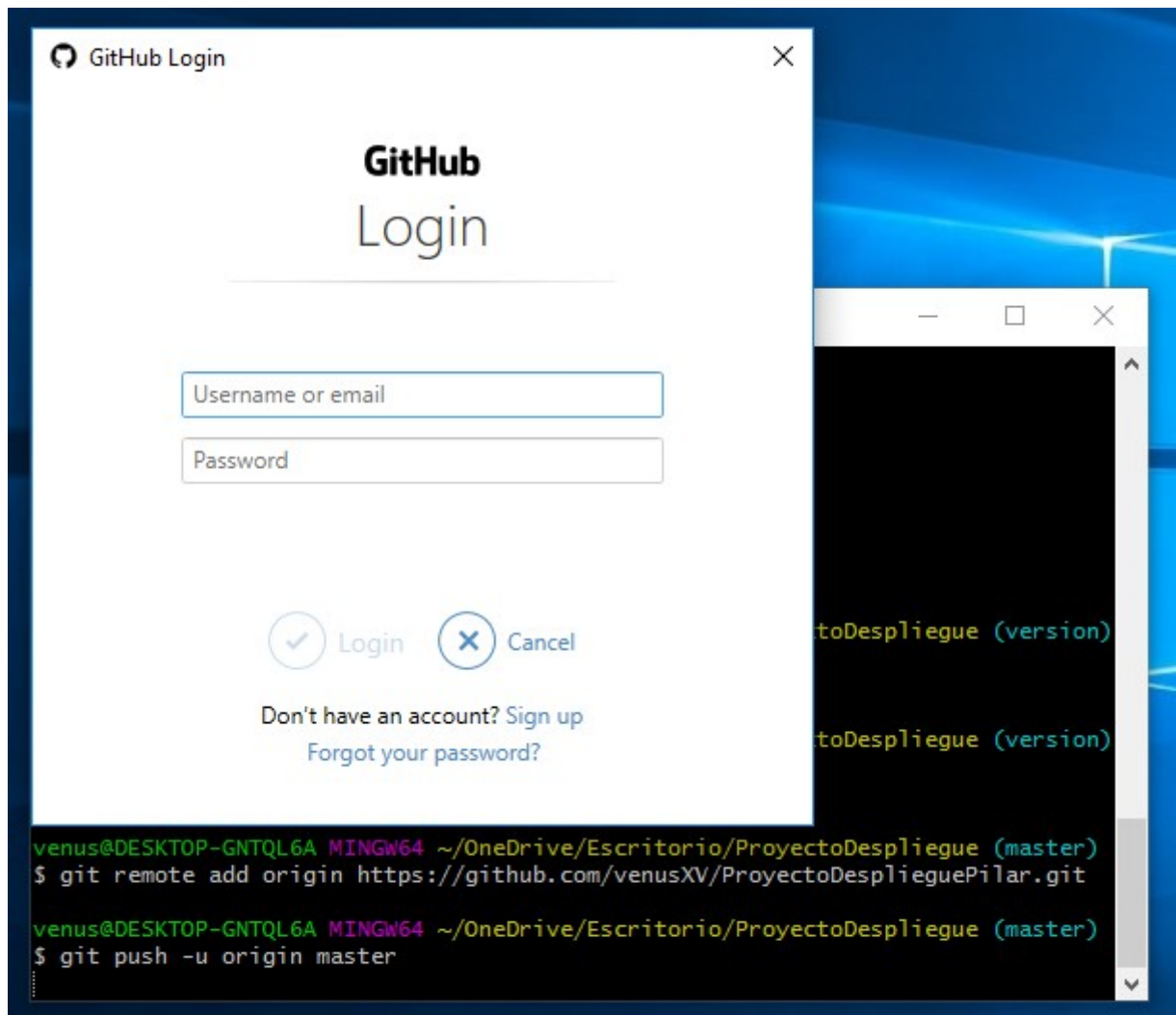


### ...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

Con "git remote add origin <https://github.com/venusXV/ProyectoDesplieguePilar.git>" le indicamos a nuestro proyecto cual es el origen y donde vamos a almacenar el código. Con "git push -u origin master" damos la orden a git de ingresar los archivos al repositorio. Ejecutamos estos comandos en la consola de Git y si se realiza correctamente se abre una ventana que nos pide ingresar nuestro usuario y contraseña a GitHub.



Una vez nos identificamos en la consola aparece que se esta subiendo nuestro proyecto al repositorio de GitHub. Y en la pagina donde se esta alojando aparecerá todo nuestro proyecto.

Switched to branch 'master'

```
venus@DESKTOP-GNTQL6A MINGW64 ~/OneDrive/Escritorio/ProyectoDesplieguePilar
$ git remote add origin https://github.com/venusXV/ProyectoDesplieguePilar.git
$ git push -u origin master
```

venusXV / ProyectoDesplieguePilar

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Mi primer proyecto de practica. Edit

Manage topics

3 commits 1 branch 0 releases 0 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

File	Commit Message	Time
.gitignore	He agregado un gitignore	an hour ago
hola.jsp	mi primer commit	8 hours ago
index.html	mi segundo commit	8 hours ago
style.css	mi primer commit	8 hours ago

Help people interested in this repository understand your project by adding a README. Add a README

Podemos crear un archivo README desde la pagina de GitHub de nuestro proyecto en el que incluir una breve descripción de nuestro proyecto para las personas que estén interesadas sepan de que va nuestro proyecto.

- **Clonar un repositorio.**

Para clonar un repositorio propio o de otra persona, primero debemos acceder a dicho repositorio a través del enlace de GitHub. Una vez estamos dentro del repositorio tenemos dos opciones, descargarlo automáticamente como un fichero comprimido o descargarlo mediante la consola de Git con el comando "git clone enlace\_al\_repositorio".

The screenshot shows the GitHub interface for the repository 'SergioRV87 / DESPAWManual-git'. At the top, there are buttons for 'Watch', 'Star', and 'Fork', each with a count of 0. Below this is a navigation bar with links for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', and 'Insights'. The main content area shows the repository's commit history with 7 commits, 1 branch, 0 releases, and 2 contributors. A 'Clone or download' button is visible. A modal window is open, showing the 'Clone with HTTPS' option, which provides the URL 'https://github.com/SergioRV87/DESPAManual-git.git' and buttons for 'Open in Desktop' and 'Download ZIP'. The repository's file list is also visible, showing files like 'Actualizacion.txt', 'Index.html', 'Manual Git Kraken.docx', 'Manual git.docx', and 'compañero-apuntes.txt'.

```
Git CMD
C:\Users\venus>cd Desktop
C:\Users\venus\Desktop>git clone https://github.com/SergioRV87/DESPAManual-git.git
Cloning into 'DESPAManual-git'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (16/16), done.
Unpacking objectsremote: Total 21 (delta 6), reused 18 (delta 3), pack-reused 0: 80% (17/21)
Unpacking objects: 100% (21/21), done.
C:\Users\venus\Desktop>
```

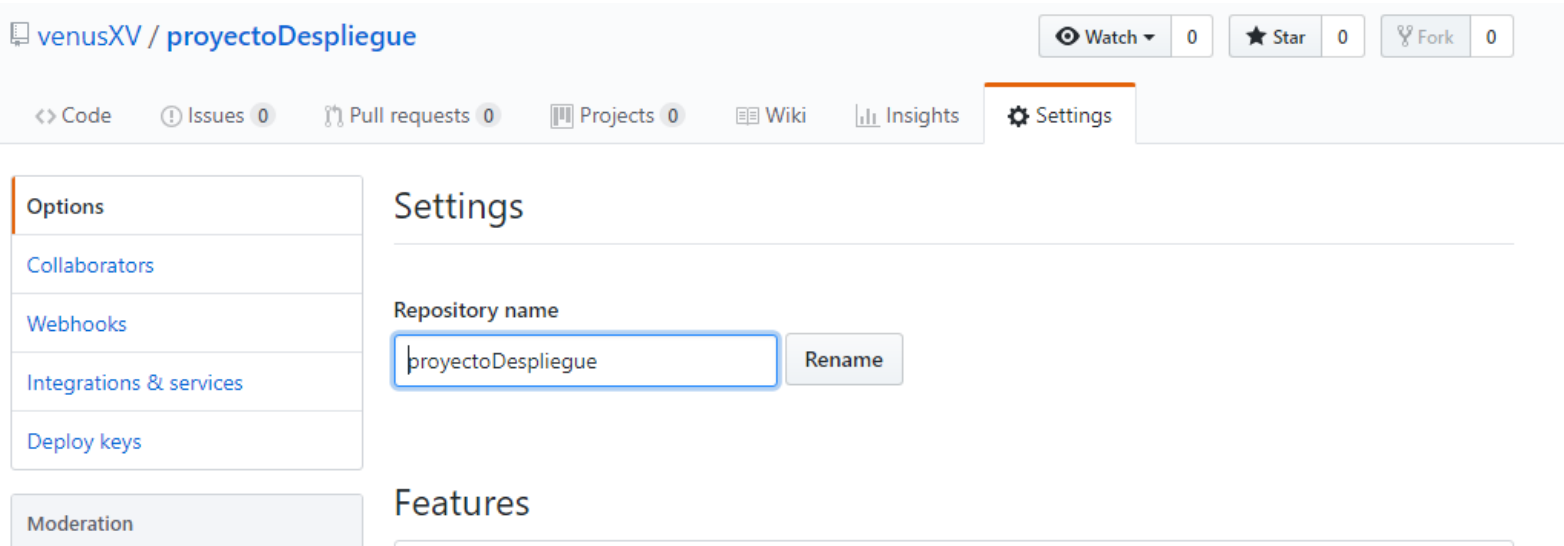
Al clonar un proyecto en nuestro ordenador, podemos trabajar con Git realizando cambios y guardando las fotografías de esos cambios. Una vez hayamos finalizado el trabajo con ese proyecto y este todo registrado en



Git, podemos eliminar el proyecto y seguirá guardado en el repositorio de GitHub junto con los cambios que hayamos realizado.

- **Eliminar un repositorio.**

Podemos eliminar un repositorio de GitHub a través de la página <https://github.com/>. Accedemos al proyecto que queremos borrar y



hacemos clic sobre "Settings".

En esta pagina abajo del todo encontramos la opción para eliminar el repositorio de GitHub en la llamada Danger Zone. Hacemos clic sobre "Delete this repository". Por seguridad habrá que introducir el nombre del repositorio y nuestra contraseña de GitHub para confirmar su eliminación.

## Danger Zone

**Make this repository private**  
Please [upgrade your plan](#) to make this repository private.

**Transfer ownership**  
Transfer this repository to another user or to an organization where you have the ability to create repositories.

Transfer

**Archive this repository**  
Mark this repository as archived and read-only.

Archive this repository

**Delete this repository**  
Once you delete a repository, there is no going back. Please be certain.

Delete this repository

## 6. GitKraken

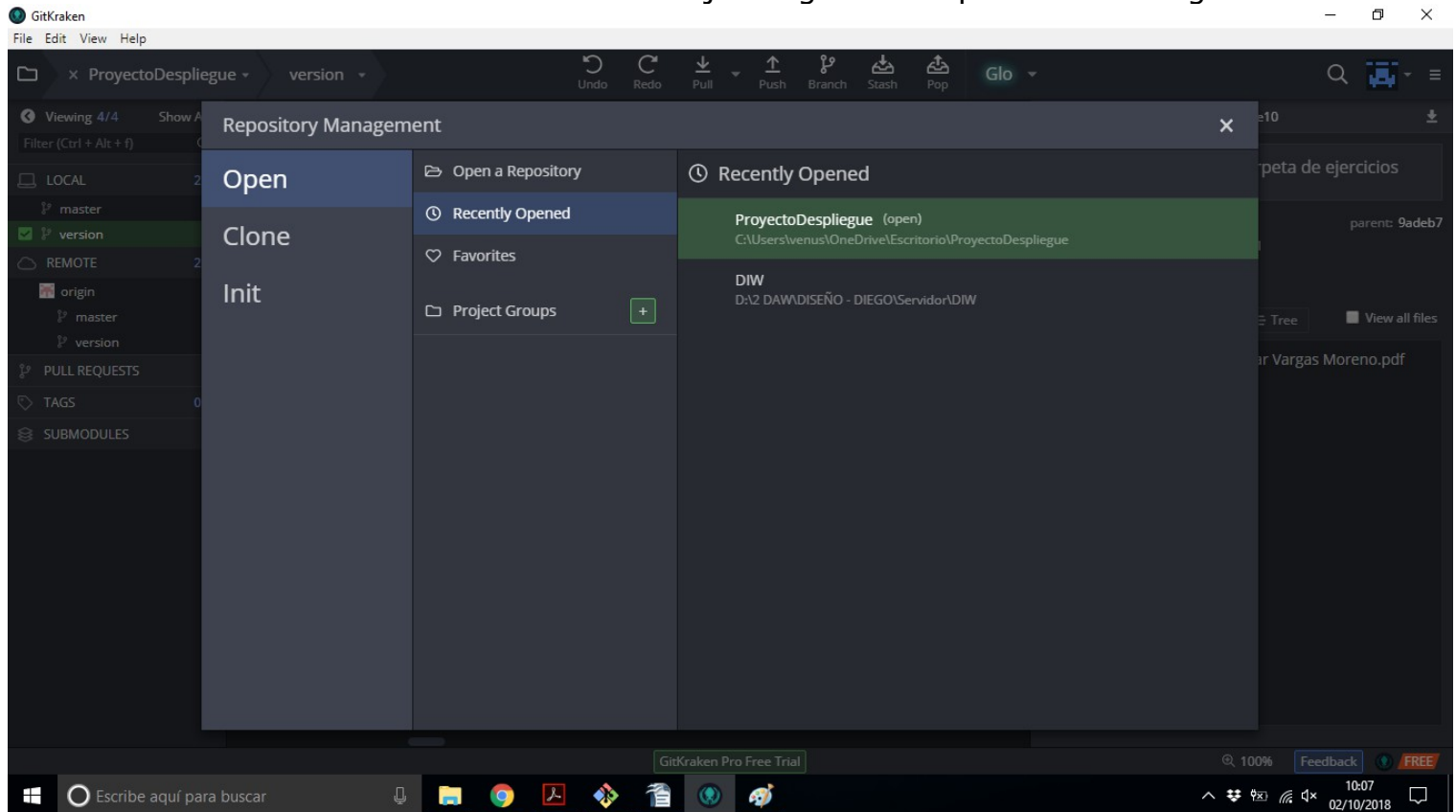


- **Instalación.**

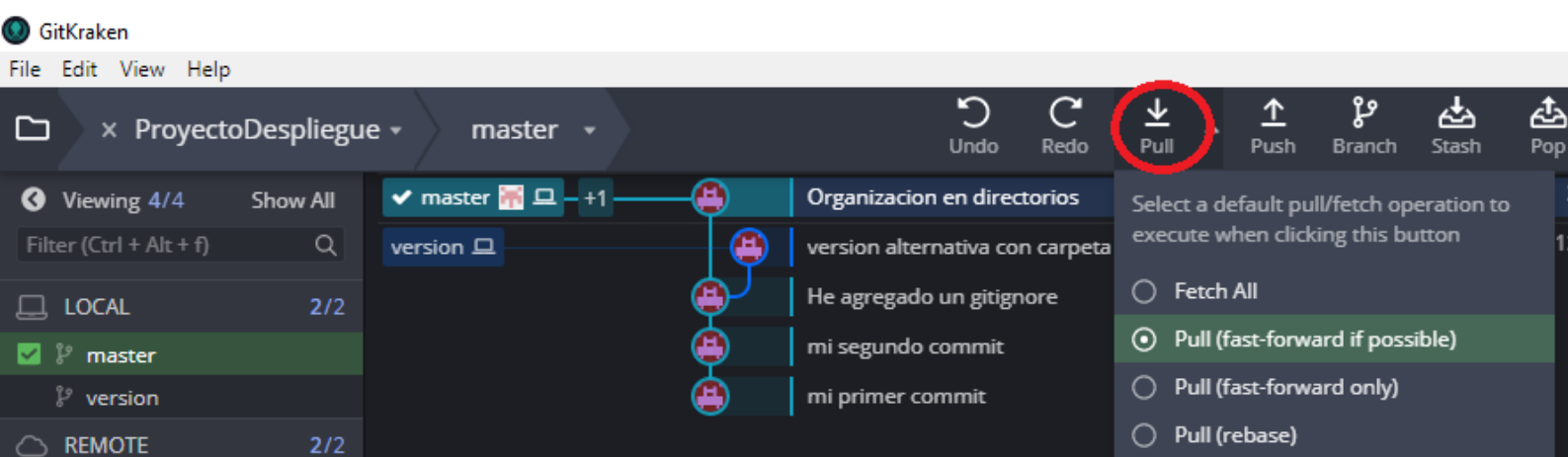
Descargamos GitKraken de la pagina <https://www.gitkraken.com/>. Una vez descargado lo ejecutamos y seguimos los pasos para instalarlo. Nos identificamos con nuestra cuenta de GitHub para poder acceder a nuestros repositorios y comenzar a trabajar.

- **Trabajar con GitKraken.**

Abrimos nuestro proyecto en GitKraken haciendo clic en File > Open Repo. Se abrirá nuestro proyecto y ya podemos trabajar con el a través de GitKraken. La forma de trabajo es igual a Git pero de forma gráfica.



Para descargar o hacer un pull en nuestro proyecto a través de GitKraken hacemos clic en el botón Pull y seleccionamos el tipo de pull que queremos realizar.



Para subir o realizar un push al repositorio de nuestro proyecto, al igual que para hacer un pull, hacemos clic en el botón push y se subirán al repositorio los nuevos cambios que hayamos realizado.

