

# SEMANA 04: GIT

## 1. Instalación

Se descarga GIT del enlace <https://git-scm.com/>



## 2. Configuración Git por primera vez

Lo primero que se debe de hacer cuando se instala Git es establecer el nombre de usuario y dirección de correo electrónico, es importante porque los **commits** de Git usan esta información para introducirlas en los commits que envías.

Para ello abrimos **Git Bash** y escribimos los siguientes comandos

```
$git config --global user.name "Tu Nombre"
$git config --global user.email tucorreo@example.com
```

The image shows a screenshot of a Git Bash terminal window. The title bar indicates the path "MINGW64:/c:/Users/Ruth/Desktop/Proyecto". The terminal shows the following commands and output:

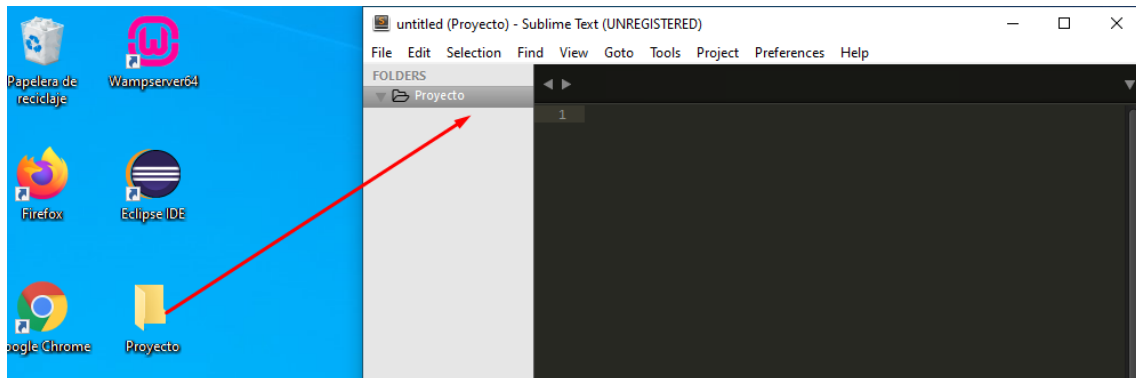
```
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git config --global user.name "Juan Carlos"

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git config --global user.email jtinoco@gmail.com

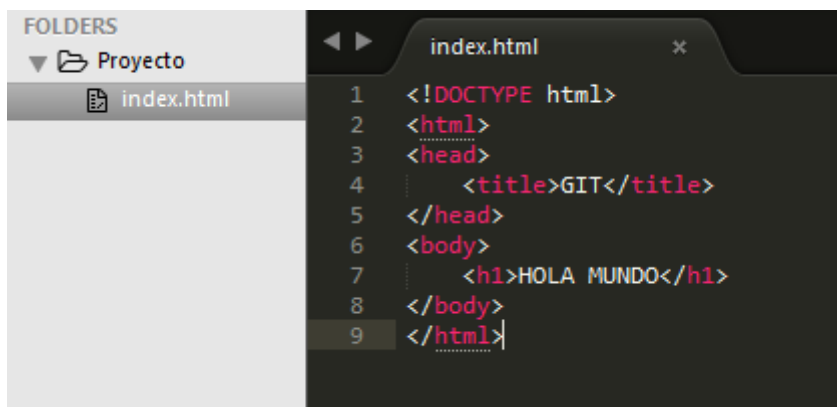
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$
```

### 3. Creando nuestro repositorio

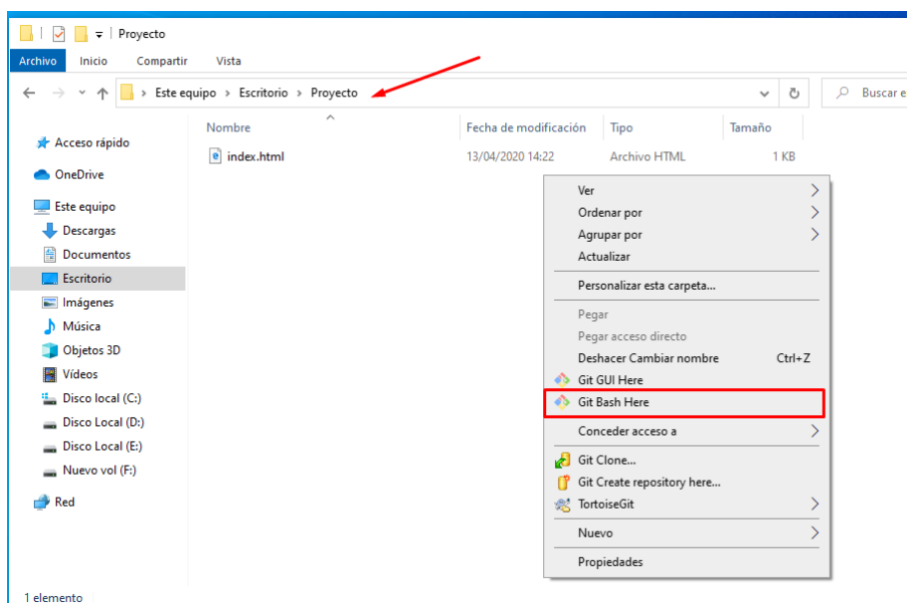
Primero se crea una carpeta llamada Proyecto y la abrimos en nuestro editor de texto



En nuestro repositorio se crea un archivo index.html y se da la estructura de un HOLA MUNDO



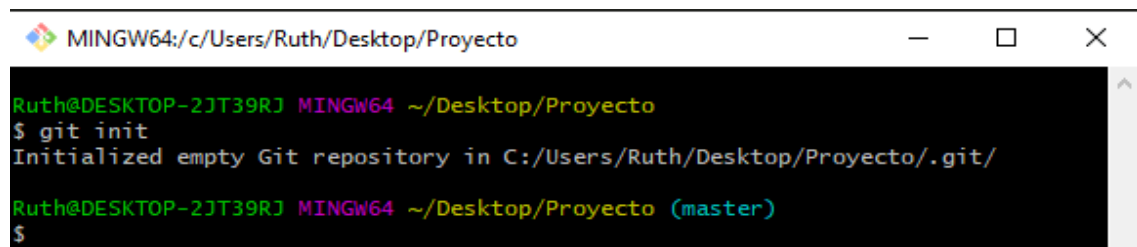
Luego se inicializa el proyecto, para ello nos dirigimos a nuestro proyecto y hacemos clic derecho seleccionando **Git Bash Here**



Escribimos los comandos :

```
$git init  
$git status  
$git add NombreArchivo  
$git commit -m 'Primera version'
```

Con el comando **init** se inicializa el proyecto y crea la carpeta oculta .git en nuestro repositorio



```
MINGW64:/c/Users/Ruth/Desktop/Proyecto  
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto  
$ git init  
Initialized empty Git repository in C:/Users/Ruth/Desktop/Proyecto/.git/  
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)  
$
```

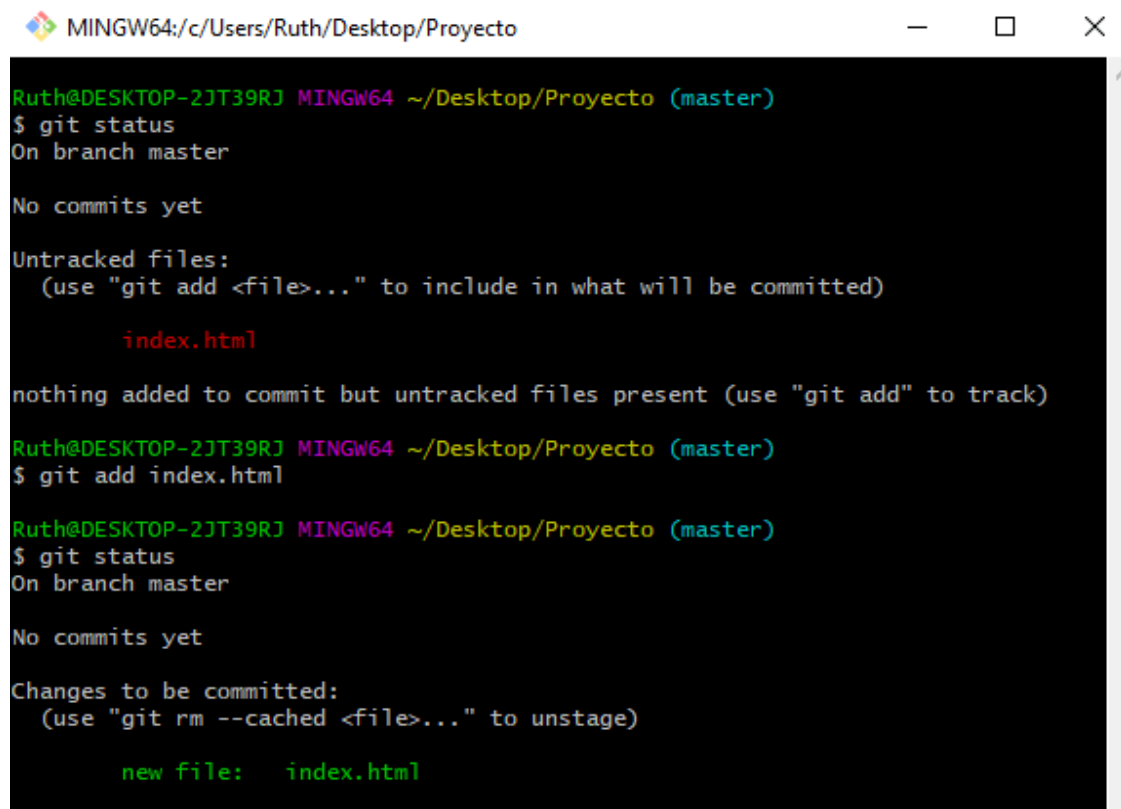
Proyecto

Compartir Vista

Este equipo > Escritorio > Proyecto

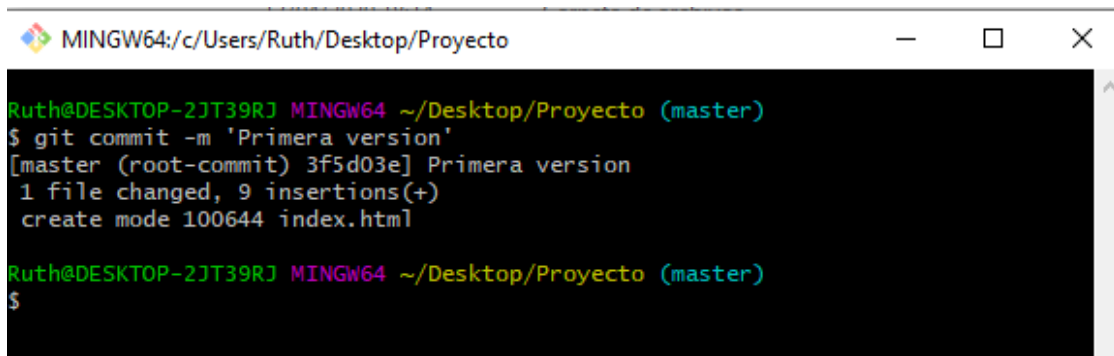
| Nombre     | Fecha de modificación | Tipo                | Tamaño |
|------------|-----------------------|---------------------|--------|
| .git       | 13/04/2020 16:58      | Carpeta de archivos |        |
| index.html | 13/04/2020 14:22      | Archivo HTML        | 1 KB   |

El comando **status** determina en qué estado se encuentran los archivos, el comando **add** comienza a trackear el archivo *index.html*



```
MINGW64:/c/Users/Ruth/Desktop/Proyecto  
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)  
$ git status  
On branch master  
  
No commits yet  
  
Untracked files:  
  (use "git add <file>..." to include in what will be committed)  
  
    index.html  
  
nothing added to commit but untracked files present (use "git add" to track)  
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)  
$ git add index.html  
  
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)  
$ git status  
On branch master  
  
No commits yet  
  
Changes to be committed:  
  (use "git rm --cached <file>..." to unstage)  
  
    new file:   index.html
```

El comando **commit** confirma los cambios realizados.



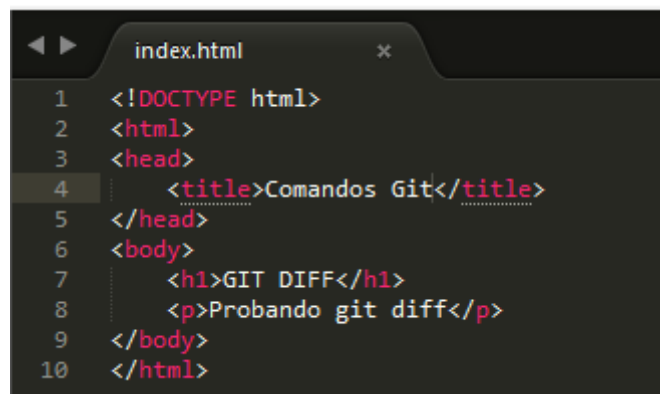
```
MINGW64:/c:/Users/Ruth/Desktop/Proyecto
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git commit -m 'Primera version'
[master (root-commit) 3f5d03e] Primera version
1 file changed, 9 insertions(+)
create mode 100644 index.html
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$
```

#### 4. Otros comandos:

- Git diff

Este comando permite ver los cambios realizados de un archivo.

Primero se realiza un cambio en nuestro archivo index.html

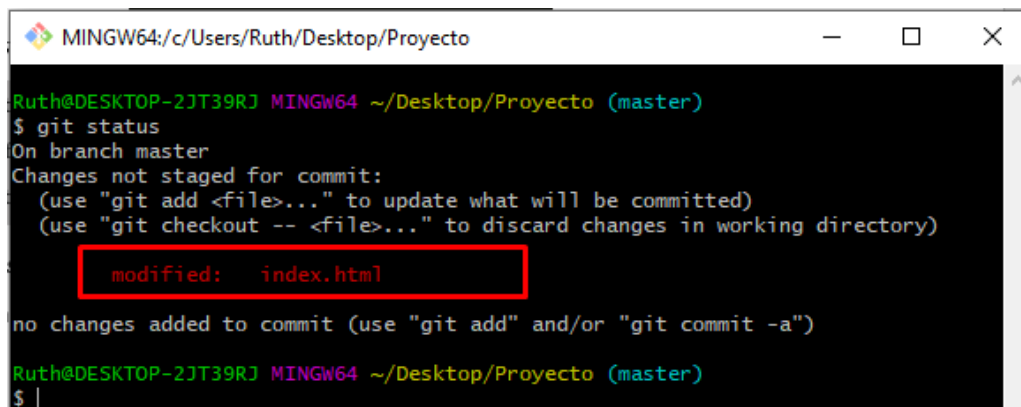


```
index.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Comandos Git</title>
5 </head>
6 <body>
7   <h1>GIT DIFF</h1>
8   <p>Probando git diff</p>
9 </body>
10 </html>
```

Luego en nuestra consola bash escribimos los siguientes comandos:

```
$git status
$git diff index.html
$git status
```

Con el primer **status** nos muestra que el estado del archivo **index.html** ha sido modificado.



```
MINGW64:/c:/Users/Ruth/Desktop/Proyecto
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (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")
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$
```

Para ver los cambios realizados en el archivo se usa el comando **diff**

```
MINGW64:/c/Users/Ruth/Desktop/Proyecto

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git diff index.html
diff --git a/index.html b/index.html
index f2b25af..dd5f728 100644
--- a/index.html
+++ b/index.html
@@ -1,9 +1,10 @@
<!DOCTYPE html>
<html>
<head>
-   <title>GIT</title>
+   <title>Comandos Git</title>
</head>
<body>
-   <h1>HOLA MUNDO</h1>
+   <h1>GIT DIFF</h1>
+   <p>Probando git diff</p>
</body>
</html>
\ No newline at end of file
```

Agregamos y guardamos los cambios de nuestro archivo.

```
MINGW64:/c/Users/Ruth/Desktop/Proyecto

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git add index.html

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   index.html

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git commit -m 'Segunda version'
[master 3d110ce] Segunda version
1 file changed, 3 insertions(+), 2 deletions(-)
```

- **Git log**

Este comando permite ver el historial de confirmaciones de nuestro proyecto, nos muestra el autor y la fecha.

```
$git log
```

```
MINGW64:/c/Users/Ruth/Desktop/Proyecto

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git log
commit 3d110ce5d9823354e38a4cb7669811f1b73cb4e7 (HEAD -> master)
Author: Juan Carlos <jtinoco@gmail.com>
Date: Mon Apr 13 18:41:53 2020 -0500

    Segunda version

commit 3f5d03e6d1a0f436ea701b201c46e0ffd90de8e3
Author: Juan Carlos <angiejenifer16@gmail.com>
Date: Mon Apr 13 18:14:16 2020 -0500

    Primera version
```

- Git checkout - -

Para revertir los cambios de un archivo.

Primero realizamos cambios en nuestro *index.html*

```
index.html x
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Comandos Git</title>
5 </head>
6 <body>
7   <p>Probando git checkout</p>
8 </body>
9 </html>
```

Luego en nuestra consola bash escribimos los siguientes comandos:

```
$git status
$git checkout --index.html
```

Con **status** nos muestra que el estado del archivo ha sido modificado.

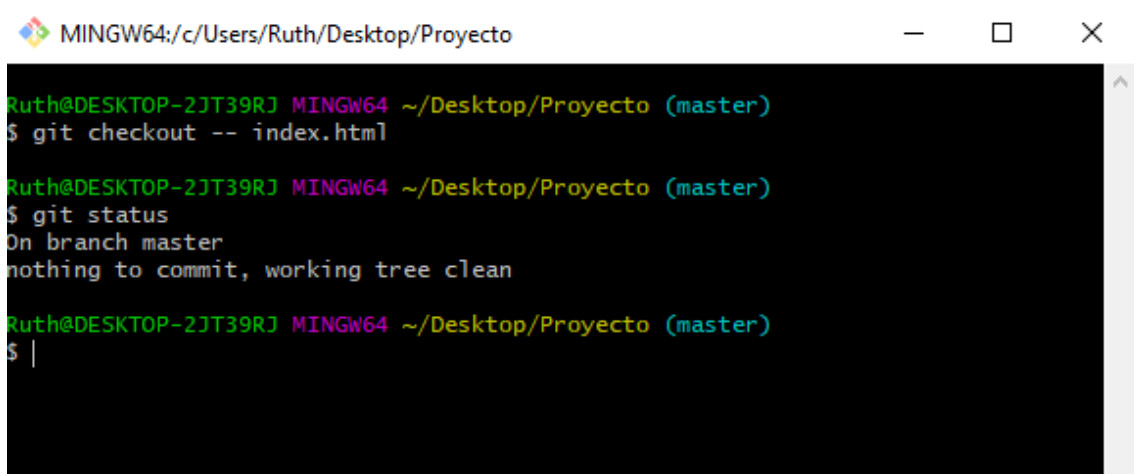
```
MINGW64:/c/Users/Ruth/Desktop/Proyecto

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (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")
```

Con **checkout** – se pueden descartar los cambios que se han hecho en el archivo, y luego el **status** nos dice que no hay ningún cambio.



```
MINGW64:/c/Users/Ruth/Desktop/Proyecto

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git checkout -- index.html

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git status
On branch master
nothing to commit, working tree clean

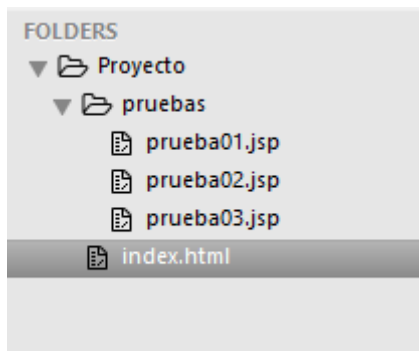
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ |
```

Y los cambios de nuestro archivo **index.html** han sido descartados.

- **.gitignore**

Se puede crear el archivo **.gitignore** en el repositorio para decirle a Git que archivos y carpetas ignorar cuando se hace un commit.

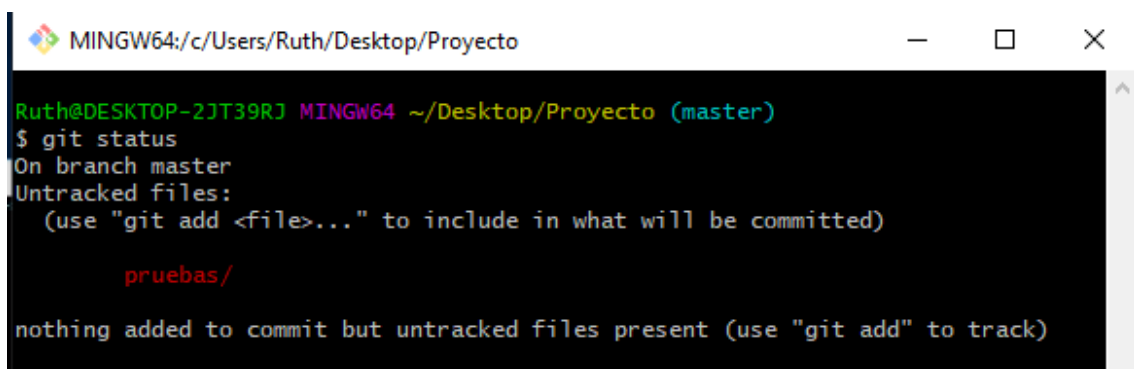
Para ello creamos la carpeta prueba con tres archivos *en* nuestro repositorio.



En la consola de bash escribimos

```
$git status
```

Nos muestra la carpeta **pruebas** para ser trackear nuestro repositorio.



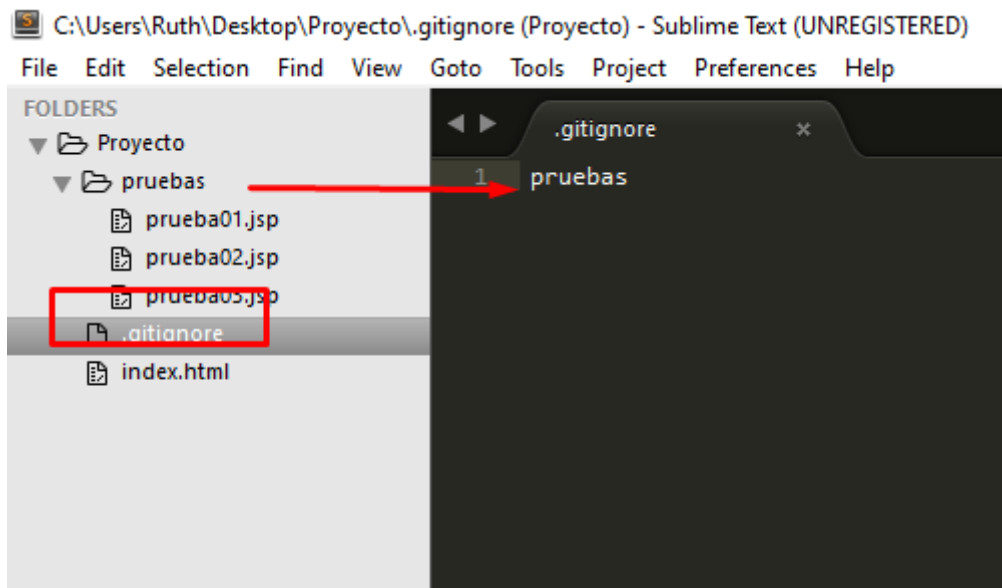
```
MINGW64:/c/Users/Ruth/Desktop/Proyecto

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    pruebas/

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

Creamos el archivo **.gitignore** en la raíz de nuestro repositorio.



En la consola de bash escribimos

```
$git status
```

Nos muestra solo el archivo **.gitignore** para ser trackear nuestro repositorio, esto es debido a que en nuestro archivo agregamos el nombre de la carpeta pruebas.

```
MINGW64:/c/Users/Ruth/Desktop/Proyecto

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        .gitignore

nothing added to commit but untracked files present (use "git add" to track)
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$
```

- **Git branch**

Este comando lista todas las ramas de nuestro repositorio, y también nos permite crear y borrar ramas.

```
MINGW64:/c/Users/Ruth/Desktop/Proyecto

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git branch
* master

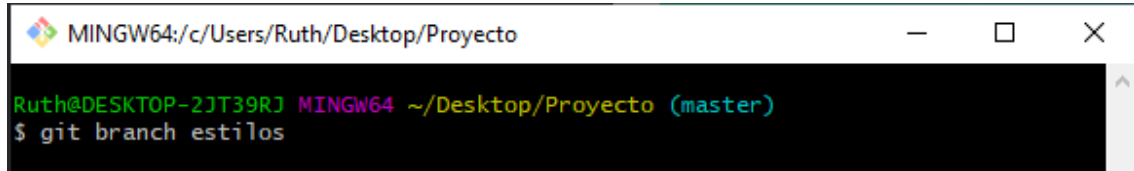
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$
```



Para crear una rama se agregan los siguientes comandos:

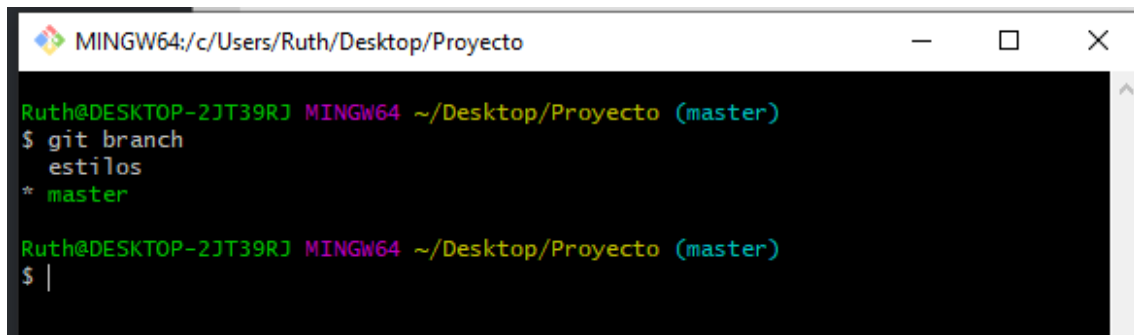
```
$git branch NombreRama
```

Agregamos la rama estilos



```
MINGW64:/c/Users/Ruth/Desktop/Proyecto
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git branch estilos
```

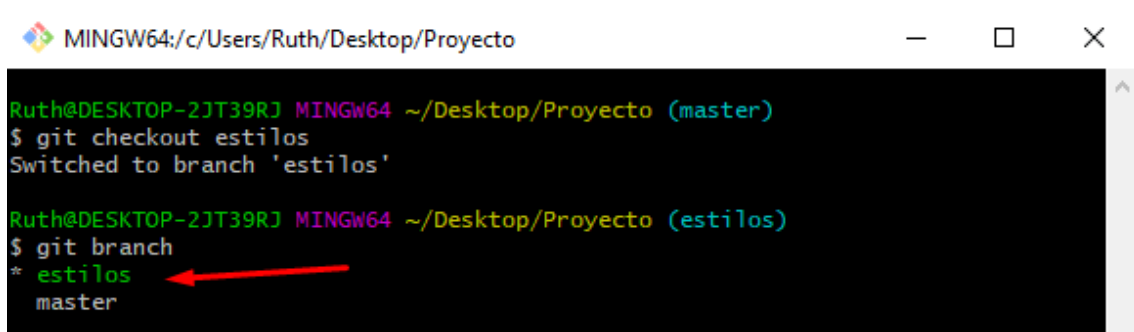
Listamos de nuevo, y nos muestra que estamos en la rama **master**



```
MINGW64:/c/Users/Ruth/Desktop/Proyecto
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git branch
estilos
* master
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ |
```

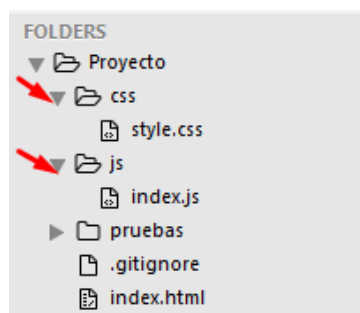
Para cambiar de rama se escribe los siguientes comandos:

```
$git checkout estilos
$git brach
```



```
MINGW64:/c/Users/Ruth/Desktop/Proyecto
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git checkout estilos
Switched to branch 'estilos'
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (estilos)
$ git branch
* estilos
master
```

Creamos la carpeta css en la raíz de nuestro repositorio y agregamos el archivo style.css, también creamos la carpeta js y agregamos el archivo index.js y hacemos un commit.



```
MINGW64:/c/Users/Ruth/Desktop/Proyecto

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (estilos)
$ git status
On branch estilos
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        css/
        js/

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

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (estilos)
$ git add .

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (estilos)
$ git commit -m 'Agregando las hojas de estilos'
[estilos 52e7fb3] Agregando las hojas de estilos
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 css/style.css
create mode 100644 js/index.js
```

Revisamos el historial de confirmaciones, y vemos que el commit anterior se guardó en la rama **estilos**.

```
MINGW64:/c/Users/Ruth/Desktop/Proyecto

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (estilos)
$ git log
commit 52e7fb3e2767b166a555ce78d909b7d9496919e7 (HEAD -> estilos)
Author: Juan Carlos <jtinoco@gmail.com>
Date:   Mon Apr 13 20:54:00 2020 -0500

    Agregando las hojas de estilos

commit f1ad89e4df03df18112c2e38230c1d8d1d78d252 (master)
Author: Juan Carlos <jtinoco@gmail.com>
Date:   Mon Apr 13 20:12:28 2020 -0500

    agregando gitignore

commit 3d110ce5d9823354e38a4cb7669811f1b73cb4e7
Author: Juan Carlos <jtinoco@gmail.com>
Date:   Mon Apr 13 18:41:53 2020 -0500

    Segunda version

commit 3f5d03e6d1a0f436ea701b201c46e0ffd90de8e3
Author: Juan Carlos <angiejenifer16@gmail.com>
Date:   Mon Apr 13 18:14:16 2020 -0500

    Primera version
```

Cambiamos de rama a **master** y revisamos el historial de confirmaciones en donde no se encontrará los cambios realizados en la rama de estilos.

```
$git checkout master
$git log
```

```
MINGW64:/c/Users/Ruth/Desktop/Proyecto

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (estilos)
$ git checkout master
Switched to branch 'master'

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git log
commit f1ad89e4df03df18112c2e38230c1d8d1d78d252 (HEAD -> master)
Author: Juan Carlos <jtinoco@gmail.com>
Date: Mon Apr 13 20:12:28 2020 -0500

    agregando gitignore

commit 3d110ce5d9823354e38a4cb7669811f1b73cb4e7
Author: Juan Carlos <jtinoco@gmail.com>
Date: Mon Apr 13 18:41:53 2020 -0500

    Segunda version

commit 3f5d03e6d1a0f436ea701b201c46e0ffd90de8e3
Author: Juan Carlos <angiejenifer16@gmail.com>
Date: Mon Apr 13 18:14:16 2020 -0500

    Primera version
```

Y tampoco se verán en nuestro repositorio los archivos creados en la rama estilos.

Este equipo > Escritorio > Proyecto

Buscar en Proyecto

| Nombre     | Fecha de modificación | Tipo                | Tamaño |
|------------|-----------------------|---------------------|--------|
| .git       | 13/04/2020 22:08      | Carpeta de archivos |        |
| pruebas    | 13/04/2020 19:41      | Carpeta de archivos |        |
| .gitignore | 13/04/2020 20:08      | Documento de te...  | 1 KB   |
| index.html | 13/04/2020 19:02      | Archivo HTML        | 1 KB   |

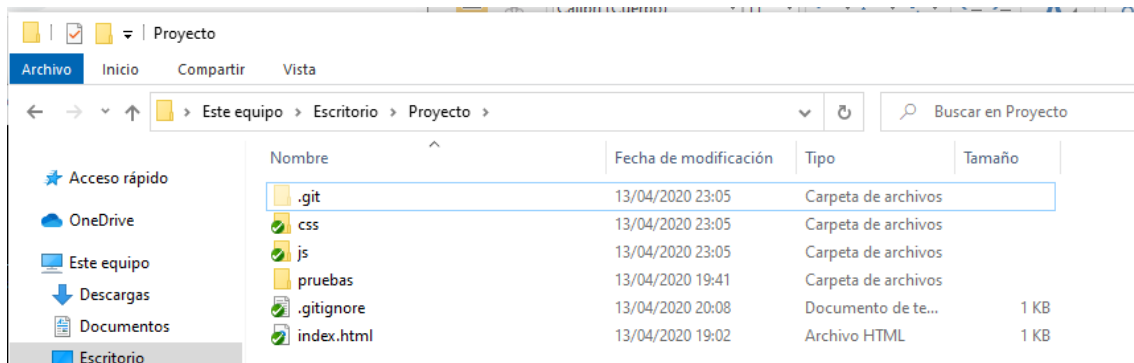
## Git merge

Este comando fusiona las ramas. En nuestra consola bash escribimos los siguientes comandos

```
$git merge estilos
```

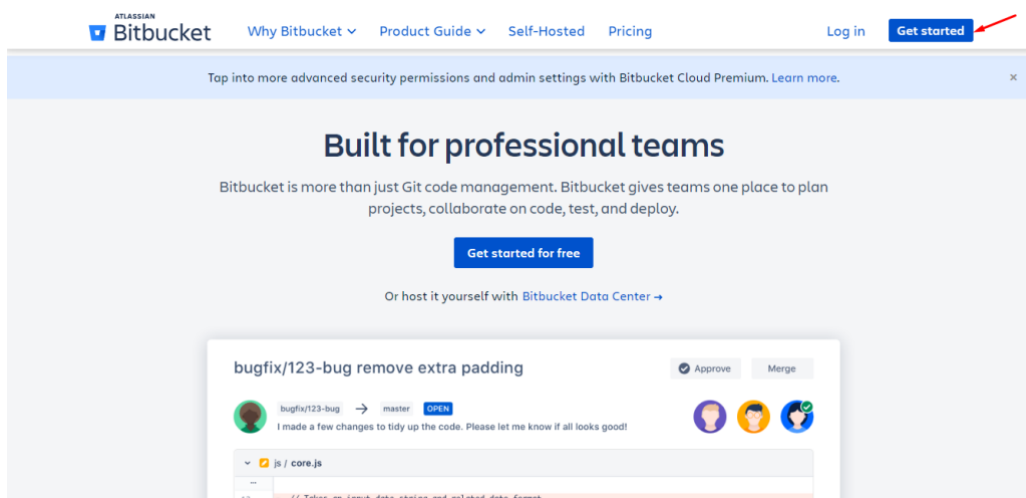
```
MINGW64:/c/Users/Ruth/Desktop/Proyecto

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git merge estilos
Updating f1ad89e..52e7fb3
Fast-forward
 css/style.css | 0
 js/index.js   | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 css/style.css
 create mode 100644 js/index.js
```

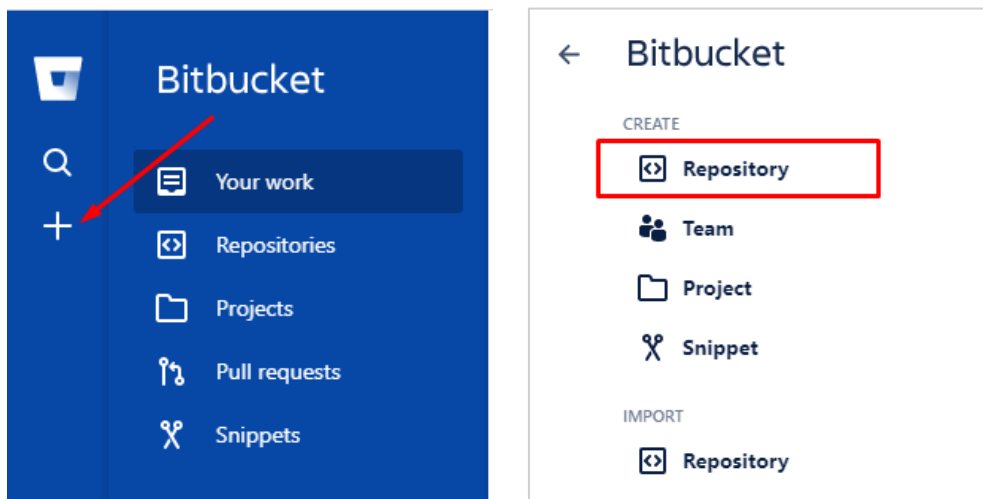


## 5. Subir nuestro repositorio

Se crea una cuenta en Bitbucket <https://bitbucket.org/product/>



Creamos un repositorio en Bitbucket



**Create a new repository** [Import repository](#)

Repository name

Access level ☒ This is a private repository  
 Uncheck to make this repository public. Public repositories typically contain open-source code and can be viewed by anyone.

Include a README?

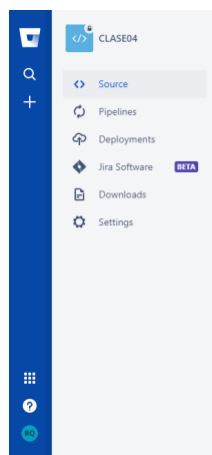
[Advanced settings](#)

Description

Forking

Language

[Create repository](#) [Cancel](#)



**Let's put some bits in your bucket**

HTTPS

**Get started quickly**  
 Creating a README or a .gitignore is a quick and easy way to get something into your repository.  
[Create a README](#) [Create a .gitignore](#)

**Get your local Git repository on Bitbucket**

Step 1: Switch to your repository's directory

```
1 cd /path/to/your/repo
```

Step 2: Connect your existing repository to Bitbucket

```
1 git remote add origin https://rramirezq@bitbucket.org/rramirezq/clase04.git
```

En nuestra consola bash escribimos los siguientes comandos:

```
$git remote add origin direccion
$git push -u origin master
```

Copiamos los comandos de nuestro repositorio de Bitbucket al bash.

## Get your local Git repository on Bitbucket

Step 1: Switch to your repository's directory

```
1 cd /path/to/your/repo
```

Step 2: Connect your existing repository to Bitbucket

```
1 git remote add origin https://rramirezq@bitbucket.org/rramirezq/clase04.git
2 git push -u origin master
```

Need more information? [Learn more](#)

Con **remote add origin** se conecta el repositorio local a un repositorio remoto, y con **push -u origin master** se envían los cambios al repositorio remoto.

```
MINGW64:/c/Users/Ruth/Desktop/Proyecto
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git remote add origin https://rramirezq@bitbucket.org/rramirezq/clase04.git

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/Proyecto (master)
$ git push -u origin master
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 4 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (14/14), 1.25 KiB | 428.00 KiB/s, done.
Total 14 (delta 0), reused 0 (delta 0)
To https://bitbucket.org/rramirezq/clase04.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Podemos verificar nuestro repositorio en Bitbucket

CLASE04

Source

Commits

Branches

Pull requests

Pipelines

Deployments

Jira Software BETA

Downloads

Settings

Ruth Jenifer Ramirez Quispe

CLASE04

Esta es la prueba de la semana 04

master

Filter files

| Name       | Size  | Last commit | Message                        |
|------------|-------|-------------|--------------------------------|
| css        |       | 2 hours ago | Agregando las hojas de estilos |
| js         |       | 2 hours ago | Agregando las hojas de estilos |
| .gitignore | 7 B   | 3 hours ago | agregando gitignore            |
| index.html | 134 B | 5 hours ago | Segunda version                |