

# Práctica Parte I

# Git & GitHub

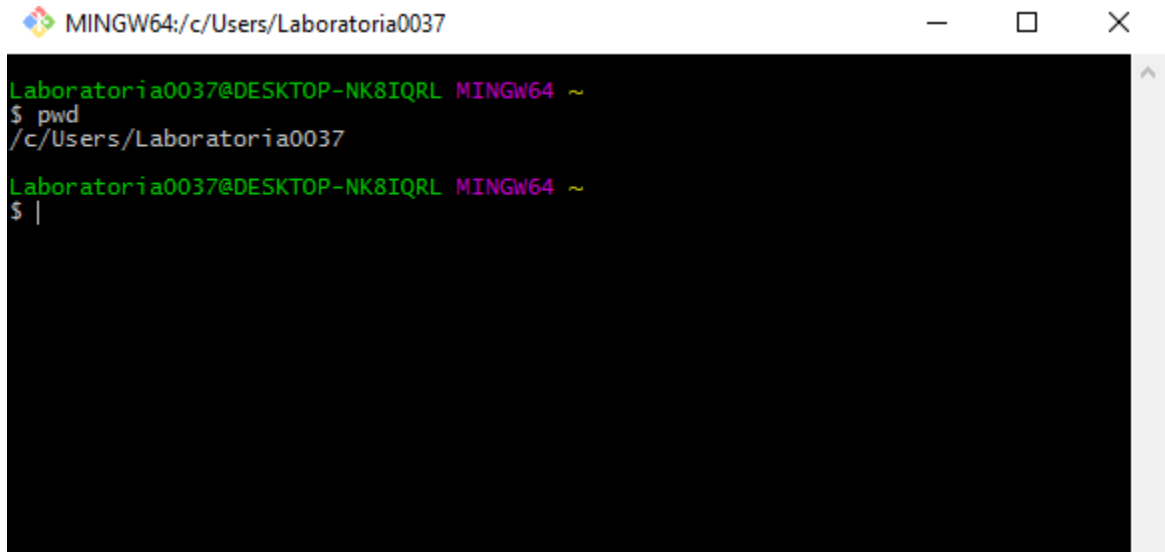
---



## Introducción

1. Empezaremos con los 6 comandos mas importantes en consola.

**pwd** : Para ubicarme en que directorio me encuentro



```
MINGW64:/c/Users/Laboratoria0037
Laboratoria0037@DESKTOP-NK8IQL MINGW64 ~
$ pwd
/c/Users/Laboratoria0037
Laboratoria0037@DESKTOP-NK8IQL MINGW64 ~
$ |
```

2. **cd** : Para situarse dentro del directorio.



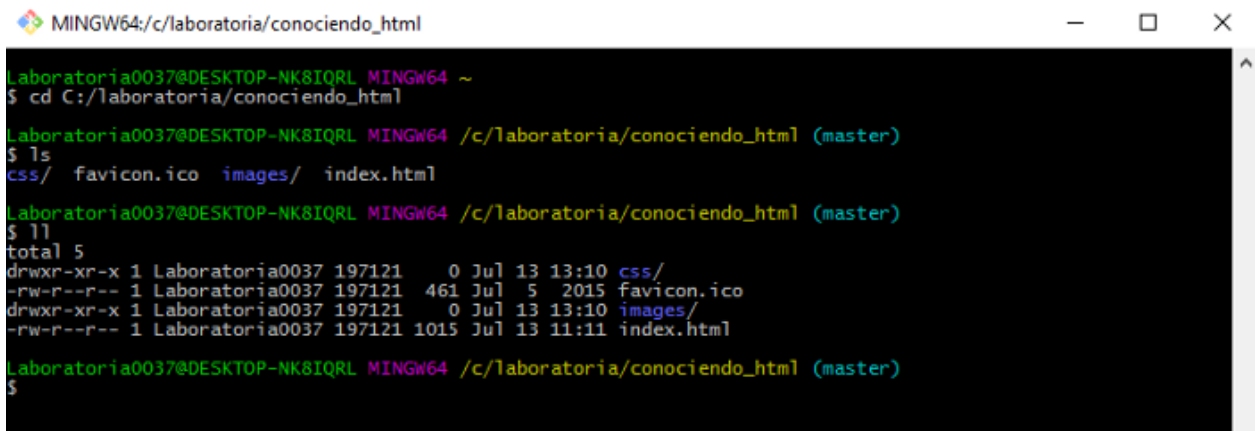
```
MINGW64:/c/laboratoria/conociendo_html
Laboratoria0037@DESKTOP-NK8IQL MINGW64 ~
$ cd C:/laboratoria/conociendo_html
Laboratoria0037@DESKTOP-NK8IQL MINGW64 /c/laboratoria/conociendo_html (master)
$ |
```

3. **ls** : Para listar todo lo que está dentro de donde nos situamos



```
MINGW64:/c/laboratoria/conociendo_html
Laboratoria0037@DESKTOP-NK8IQL MINGW64 ~
$ cd C:/laboratoria/conociendo_html
Laboratoria0037@DESKTOP-NK8IQL MINGW64 /c/laboratoria/conociendo_html (master)
$ ls
css/  favicon.ico  images/  index.html
Laboratoria0037@DESKTOP-NK8IQL MINGW64 /c/laboratoria/conociendo_html (master)
$
```

4. **ll** : Para listar de una forma más grande



```
MINGW64:/c/laboratoria/conociendo_html
Laboratoria0037@DESKTOP-NK8IQL MINGW64 ~
$ cd C:/laboratoria/conociendo_html
Laboratoria0037@DESKTOP-NK8IQL MINGW64 /c/laboratoria/conociendo_html (master)
$ ls
css/  favicon.ico  images/  index.html
Laboratoria0037@DESKTOP-NK8IQL MINGW64 /c/laboratoria/conociendo_html (master)
$ ll
total 5
drwxr-xr-x 1 Laboratoria0037 197121  0 Jul 13 13:10 css/
-rw-r--r-- 1 Laboratoria0037 197121 461 Jul  5 2015 favicon.ico
drwxr-xr-x 1 Laboratoria0037 197121  0 Jul 13 13:10 images/
-rw-r--r-- 1 Laboratoria0037 197121 1015 Jul 13 11:11 index.html
Laboratoria0037@DESKTOP-NK8IQL MINGW64 /c/laboratoria/conociendo_html (master)
$
```

¿Qué entendemos por **-rw-r-r — 1**?

**READ (r)**

**WRITE (w)**


**EXECUTE(x)**

¡Son permisos para los directorios y archivos!

5. **mkdir** : Para crear directorios.

```
Laboratoria0037@DESKTOP-NK8IQRL MINGW64 /c/laboratoria/conociendo_html (master)  
$ mkdir prueba
```


**Resultado:**

 prueba	7/13/2016 1:31 PM	File folder
--	-------------------	-------------

6. **touch** : Para crear los ficheros archivos

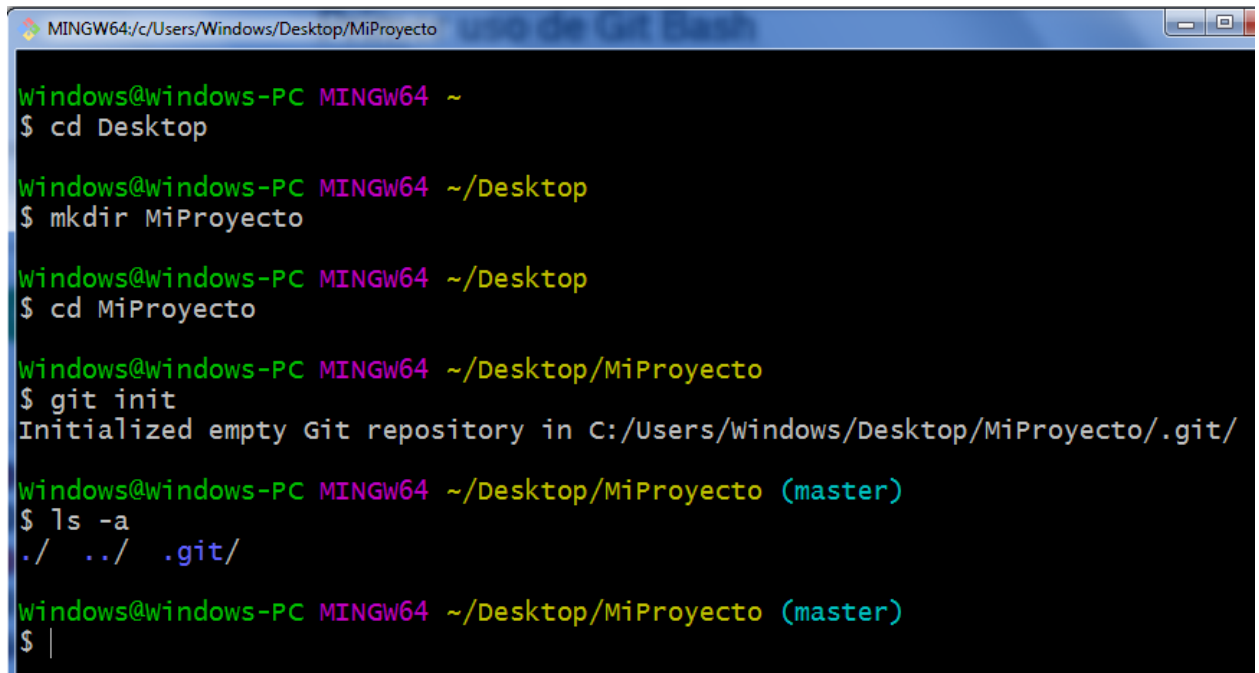
```
Laboratoria0037@DESKTOP-NK8IQRL MINGW64 /c/laboratoria/conociendo_html (master)  
$ touch prueba/main.css
```

**Resultado:**

 main	7/13/2016 1:32 PM	Cascading Style S...	0 KB
--	-------------------	----------------------	------

## Primer uso de Git Bash

1. **git init** : Todo lo que se escriba dentro de la carpeta va a estar dentro del **.git**



```
MINGW64/c:/Users/Windows/Desktop/MiProyecto
windows@windows-PC MINGW64 ~
$ cd Desktop

windows@windows-PC MINGW64 ~/Desktop
$ mkdir MiProyecto

windows@windows-PC MINGW64 ~/Desktop
$ cd MiProyecto

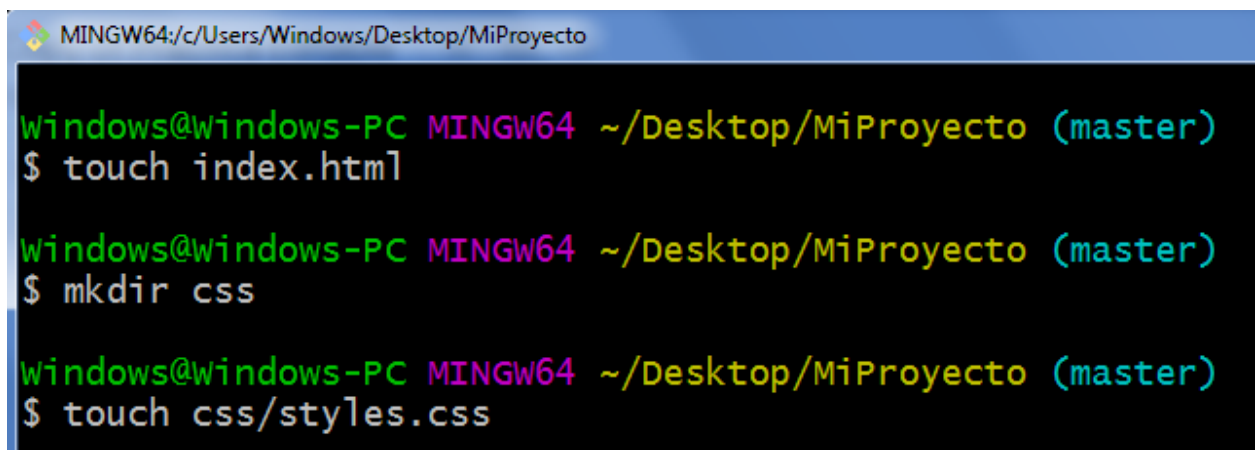
windows@windows-PC MINGW64 ~/Desktop/MiProyecto
$ git init
Initialized empty Git repository in C:/Users/Windows/Desktop/MiProyecto/.git/

windows@windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ ls -a
./ ../ .git/

windows@windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ |
```

2. **git add** : Lleva el control de los archivos que se agregan luego de escribir ese comando.

Creamos la siguiente estructura de carpetas y archivos



```
MINGW64/c:/Users/Windows/Desktop/MiProyecto
windows@windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ touch index.html

windows@windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ mkdir css

windows@windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ touch css/styles.css
```

Añadimos el archivo **index.html** a nuestro escenario (**stage**)

```
MINGW64/c/Users/Windows/Desktop/MiProyecto

windows@Windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ git add index.html

windows@Windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ git status
On branch master

No commits yet

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

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

Como podemos ver al usar **git status** (ver en qué estado se encuentran nuestros archivos) podemos ver que la carpeta **css** se encuentra aún fuera del **stage**

```
MINGW64/c/Users/Windows/Desktop/MiProyecto

windows@Windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ git add css

windows@Windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ git status
On branch master

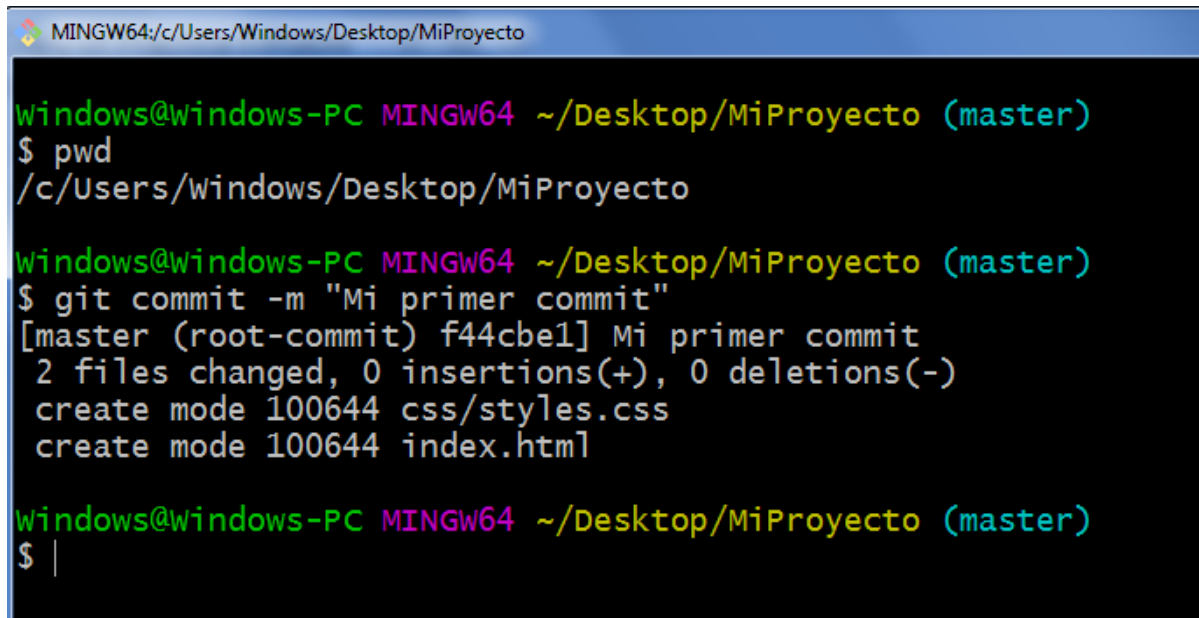
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   css/styles.css
    new file:   index.html
```

Una vez añadida la carpeta **css** a nuestro escenario (**stage**), vemos el estado del repositorio (**git status**) vemos que todos los archivos ya se encuentran en el **stage**

3. **git commit** : Comando que indica que esta lista alguna funcionalidad para que sea una versión del código.

Usamos el comando **pwd** para asegurarnos que nos encontramos en la raíz de nuestro proyecto, luego de ello ingresamos el comando **git commit -m “Mi primer commit”**

A screenshot of a terminal window with a blue title bar that reads "MINGW64:/c/Users/Windows/Desktop/MiProyecto". The terminal has a black background with green and white text. It shows the following commands and output:

```
windows@Windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ pwd
/c/Users/windows/Desktop/MiProyecto

windows@Windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ git commit -m "Mi primer commit"
[master (root-commit) f44cbe1] Mi primer commit
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 css/styles.css
create mode 100644 index.html

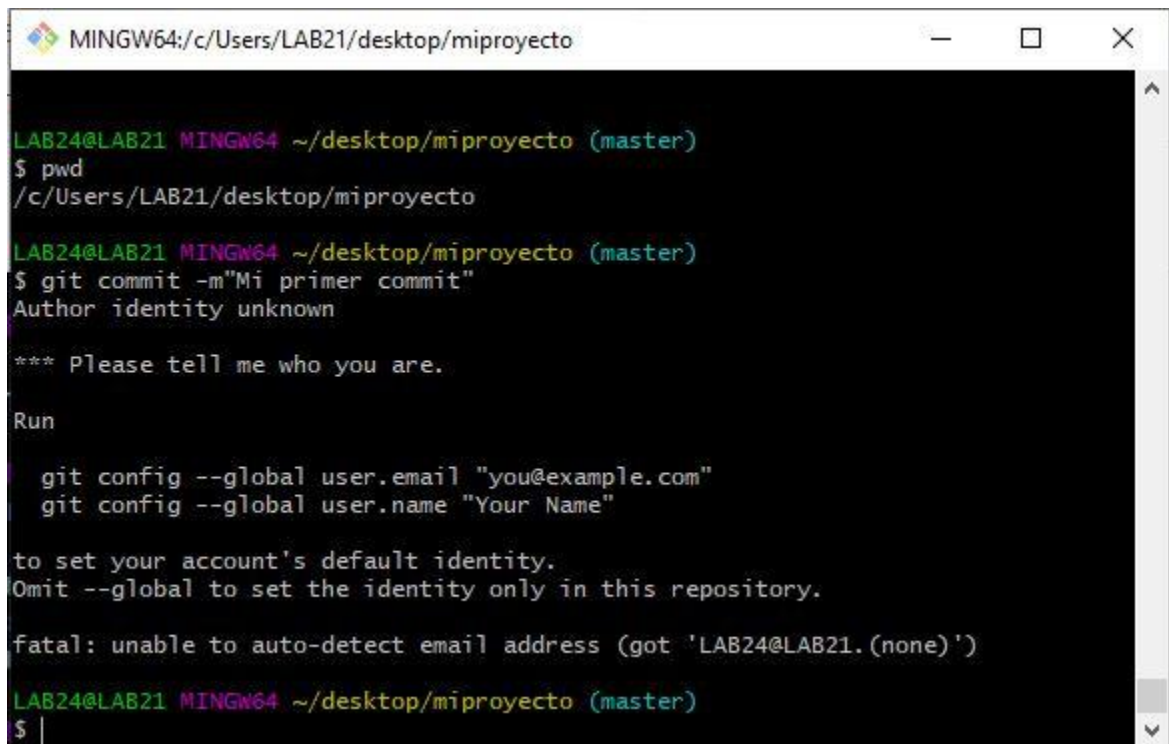
windows@Windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ |
```

El comando **git commit** envía todos los archivos que se encuentran en el escenario o **stage** (añadidos en el paso anterior) al **repositorio local** (almacén)

La opción **-m “Mensaje”** nos permite ingresar un mensaje para indicar que estamos enviando, por lo general se envía un incremento de nuestro proyecto como por ejemplo : “Pagina de login añadida”, “Pagina de registro realizada”, “Corrección de errores en en formulario de contacto”

**Consejo** : *Se tiene que detallar muy bien lo que se pone en el comentario es una buena práctica para que nuestro equipo colaborativo nos pueda entender y solucionar errores en caso sucediera.*

En caso que no nos hayamos autenticado en git nos puede salir el siguiente error

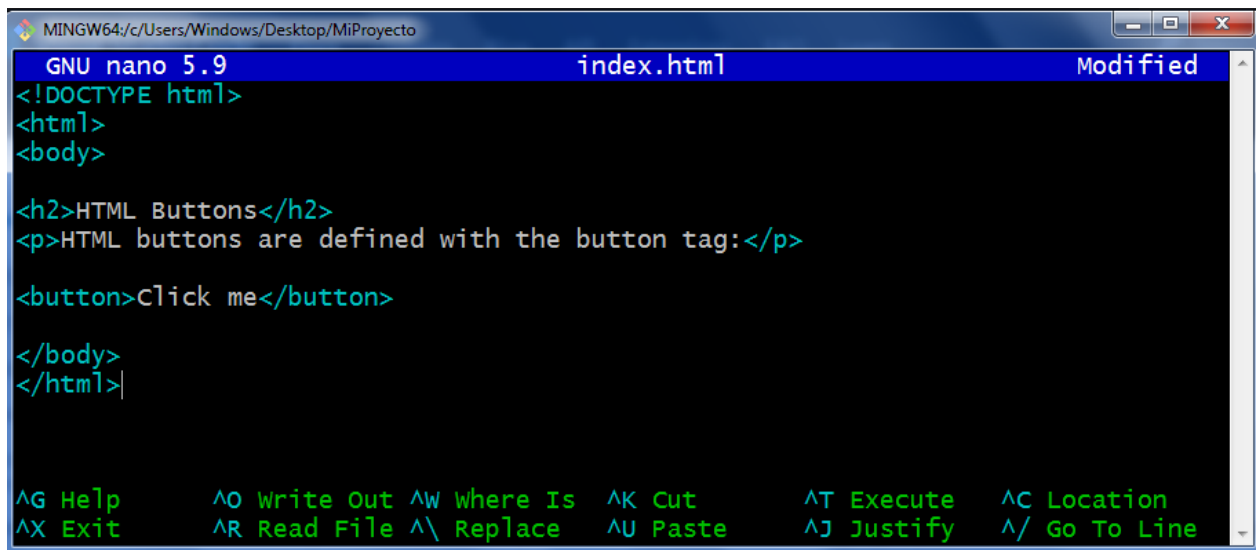
A screenshot of a terminal window titled 'MINGW64:/c/Users/LAB21/desktop/miproyecto'. The prompt is 'LAB24@LAB21 MINGW64 ~/desktop/miproyecto (master)'. The user enters '\$ pwd' and the output is '/c/Users/LAB21/desktop/miproyecto'. Then the user enters '\$ git commit -m"Mi primer commit"'. The output is 'Author identity unknown', followed by '\*\*\* Please tell me who you are.' and 'Run' on a new line. Then two lines of code are shown: 'git config --global user.email "you@example.com"' and 'git config --global user.name "Your Name"'. Below these is the text 'to set your account's default identity. Omit --global to set the identity only in this repository.' followed by the error message 'fatal: unable to auto-detect email address (got 'LAB24@LAB21.(none)')'. The prompt returns to '\$ |'.

Modifiquemos el archivo **index.html** utilizando el comando **nano**.

A screenshot of a terminal window titled 'MINGW64:/c/Users/Windows/Desktop/MiProyecto'. The prompt is 'windows@windows-PC MINGW64 ~/Desktop/MiProyecto (master)'. The user enters '\$ ls' and the output is 'css/ index.html'. Then the user enters '\$ nano index.html|'.

Ingresamos el siguiente contenido al documento **index.html**





```
MINGW64/c/Users/Windows/Desktop/MiProyecto
GNU nano 5.9 index.html Modified
<!DOCTYPE html>
<html>
<body>

<h2>HTML Buttons</h2>
<p>HTML buttons are defined with the button tag:</p>

<button>Click me</button>

</body>
</html>|

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

Utilizamos las teclas **Ctrl + X** , a continuación presionamos **Enter** y luego presionamos **la tecla Y** y finalmente un **Enter** para sobrescribir los cambios.

Ingresamos el comando **git status** nuevamente para ver el estado de nuestro proyecto.

```

Windows@Windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   index.html

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

Windows@Windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ git add index.html
warning: LF will be replaced by CRLF in index.html.
The file will have its original line endings in your working directory

```

Como podemos notar git detecto los cambios realizados en el archivo **index.html** por lo que ahora debemos añadirlo a nuestro **stage** y subirlo a nuestro repositorio

```

MINGW64:/c:/Users/Windows/Desktop/MiProyecto

Windows@Windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   index.html

Windows@Windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ git commit -m "Modificando el index.html"
[master 6da7e43] Modificando el index.html
1 file changed, 11 insertions(+)

```

Finalmente ingresamos el comando **git log** para ver todos los envíos (commits) registrados

```
MINGW64:/c/Users/Windows/Desktop/MiProyecto

windows@windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ git log
commit 6da7e439c54c2e2deaa612e69c56f885ac10fd91 (HEAD -> master)
Author: Cristhianrg <cristhianr.gomez@gmail.com>
Date: Sat Nov 27 10:09:19 2021 -0500

    Modificando el index.html

commit f44cbe1933f3896595dc4f802f34889eaf3dc890
Author: Cristhianrg <cristhianr.gomez@gmail.com>
Date: Sat Nov 27 09:48:05 2021 -0500

    Mi primer commit
```