

Universidad Mariano Gálvez de Guatemala

Mazatenango Suchitepéquez

Programación Web

Ing. Axel Aguilar



Nombre: Yonatan Alexander Ixcoy Chuc

Carrera: Ingeniera en Sistemas

Carné: 3090-15-12540

Sección: "A"

Semestre: VIII

Taller No.1

Comandos para utilizar Git

Git es un sistema de control de versiones. Un sistema de control de versiones, registra los cambios realizados en un archivo o conjunto de archivos. Una de las ventajas de Git es que podemos regresar a ver las versiones anteriores de nuestro proyecto.

Paso 1:

```
YONATAN+GMG@YONATAN MINGW64 /c
$ cd git

YONATAN+GMG@YONATAN MINGW64 /c/git
$ cd master
```

- A través de la terminal buscamos el directorio donde se encuentran nuestros archivos o documentos con los cuales vamos a trabajar.

Paso 2:

```
YONATAN+GMG@YONATAN MINGW64 /c/git/master
$ git init
Initialized empty Git repository in C:/Git/master/.git/
```

- Crea un repositorio vacío. Con esto se inicializa el repositorio dentro de nuestro ordenador local. Este comando marca el inicio de nuestro proyecto, aquí le decimos a Git que empiece a monitorear todos los cambios que le realicemos a nuestros proyectos. Este comando siempre lo debemos al utilizar Git en nuestro proyecto.

Paso 3:

```
YONATAN+GMG@YONATAN MINGW64 /c/git/master (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)
```

- El comando **git status** nos va indicar el estado de nuestro Proyecto. Como podemos ver nos indica que el archivo existente no ha sido agregado.

Paso 4:

```
YONATAN+GMG@YONATAN MINGW64 /c/git/master (master)
$ git add index.html
```

- Con el comando **git add** decidimos que archivos ya están listos para crear un commit o dar el siguiente paso.

Paso 5:

```
YONATAN+GMG@YONATAN MINGW64 /c/git/master (master)
$ git status
On branch master

No commits yet

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

        new file:   index.html
```

- Si volvemos a ver el estado, nos va indicar que ya agregamos el archivo index.html

Paso 6:

 index.html	30/07/2018 13:15	Archivo HTML
 index2.html	30/07/2018 13:15	Archivo HTML
 index3.html	30/07/2018 13:15	Archivo HTML

- Creamos más archivos dentro de nuestra carpeta y miramos el estado de nuevo con el comando **git status**.

```
YONATAN+GMG@YONATAN MINGW64 /c/git/master (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        index2.html
        index3.html

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

- Nos damos cuenta que los archivos están en la carpeta pero no han sido agregados

Paso 7:

```
YONATAN+GMG@YONATAN MINGW64 /c/git/master (master)
$ git add -A
```

- El comando git add -A nos va agregar todos los archivos que tenemos en nuestra carpeta.

```
YONATAN+GMG@YONATAN MINGW64 /c/git/master (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   index2.html
        new file:   index3.html
```

- Al verificar el estado nos damos cuenta que todos los archivos pendientes han sido agregados.

Paso 8:

```
YONATAN+GMG@YONATAN MINGW64 /c/git/master (master)
$ git commit -m "Agregamos Titulo"
[master 539a3f0] Agregamos Titulo
1 file changed, 13 insertions(+)
```

Con el comando **git commit -m "Mensaje"**, guardamos los cambios que realizamos en nuestro proyecto. En este caso le agregamos el título a nuestra página. En el mensaje debemos de colocar algo muy específico para identificar cual fue la acción que realizamos.

Paso 9:

```
YONATAN+GMG@YONATAN MINGW64 /c/git/master (master)
$ git log
commit 539a3f0b700c382bc02c5cfdbac9e6c264da417d (HEAD -> master)
Author: yixcoy <yonatanixcoy_7@hotmail.com>
Date: Mon Jul 30 13:16:15 2018 -0600

    Agregamos Titulo

commit fb951dcd93a7b5e77bb4981793639687454501ae
Author: yixcoy <yonatanixcoy_7@hotmail.com>
Date: Mon Jul 30 13:12:25 2018 -0600

    Iniciamos Proyecto
```

- El comando **git log** nos da una lista de todos nuestros commits con su respectiva información.
- El código que nos da, nos sirve para poder viajar y ver las versiones anteriores.