

# Uso de Git y GitHub (Control de versiones)

## Instalar Git

<https://git-scm.com/>

## Crear cuenta en Github

Para el sitio publico Github

*Entrar a github.com*

*Crear una cuenta (sign up), para ingresar a la cuenta (Sign in)*

## Comandos Git

```
git --version  
git --help  
git --help bclearranch
```

### Para averiguar en qué directorio estamos

```
pwd
```

### Para limpiar pantalla

```
clear
```

## Para crear el repositorio local del proyecto

### Para crear el directorio del proyecto

```
mkdir proyecto  
cd proyecto  
ls
```

### Para crear el repositorio local

```
git init
```

Se crea una carpeta ".git" oculto

### Configurar la cuenta del git

```
git config --global user.email "tucorreo@tuservicio.com"  
git config --global user.name "tu_usuario"  
git commit
```

En la cuenta de correo y el nombre de usuario no use tildes ni ñ ni caracteres especiales.

Nos informa que estamos en el master y el checksum (hash).

### Para ver estado de los archivos

```
Git status
```

### Añadir archivos del working directory al staging área

```
git add <NombreArchivo> o git add .  
git status
```

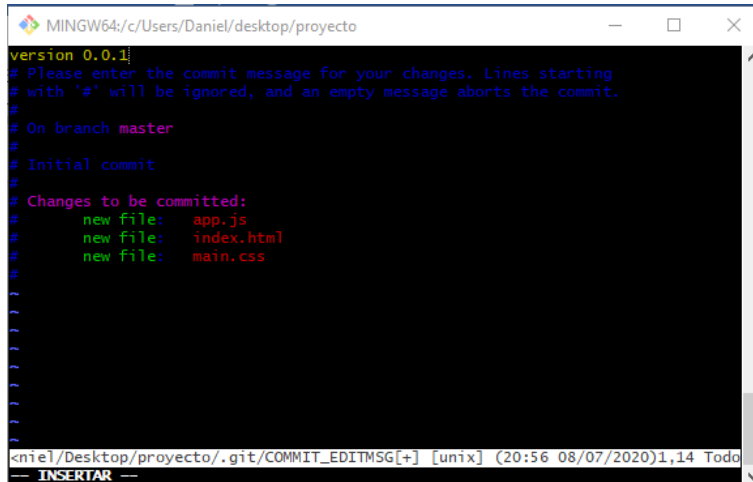
Muestra en verde los que están en el staging área y rojo los que no están.

Para seleccionar los comandos anteriores use la flecha hacia arriba o hacia abajo.

## Primer commit

```
git commit
```

Pulsar <i> para modificar, para salir de editor <esc> y para grabar y salir del editor <:wq><enter>



## Ver las diferencias

```
git add <NombreArchivo>
git status
git diff <NombreArchivo>
```

Las líneas verdes con + se añadieron y con – si se quitaron

## Regresar un cambio

```
git checkout -<NombreArchivo>
```

## Ver el estado

```
git status
```

## Siguientes commits

Si deseamos nuevamente hacer un commit con los cambios debemos añadir nuevamente el archivo add y luego commit.

```
git commit -m "mensaje"
```

No es necesario la fecha porque git se encarga de eso.

## Ver los commits

```
git log
```

## Ignorar los archivos a seguirlos

Crear el archivo “.gitignore” con el siguiente contenido:

```
Lib
```

Las librerías no se suben, sólo el trabajo que hacemos.

## Crear repositorio en Github

Para el sitio publico Github

Entrar a github.com

Crear una cuenta (sign up), para ingresar a la cuenta (Sign in)

Crea un nuevo repositorio

Clic en “New” o “Start project”



Ingresa el nombre del repositorio y la descripción, publico, clic “Create repository”

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

---

Owner \*      Repository name \*


 jdgamarram / proyectoPrueba 


Great repository names are short and memorable. Need inspiration? How about [symmetrical-octo-memory](#)?

Description (optional)

Es un proyecto para demostrar el uso de Git y Github

---


☒  **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

---

Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer.

Add .gitignore: **None**      Add a license: **None**      

---

**Create repository**

Copiar la URL git

Sincronizar git con github

```
git remote add origin <URL git.git>
git push -u origin master
```

Crear el README y clonar

Esquema:

```
# Título del Proyecto
_Acá va un párrafo que describa lo que es el proyecto_
## Subtitulo
### Subsubtitulo
_ _ _
Código
_ _ _
```

Crear el Readme

Ingresa al proyecto

Clic en el botón “Add README”

Ejemplo:

```
# proyecto-git
Proyecto git de prueba
```

```
## Introducción
```

```
```\php  
<?php  
phpInfo();  
?>  
```\
```

```
## Descripción
```

[Clic en preview](#)

[Clic en Commit new file](#)

Ya se añadió a la rama.

## Restaurar de Github a git

[Cerrar todos los archivos y git](#)

[Borrar el proyecto del escritorio](#)

[Abrir git](#)

[Cambiar al directorio donde se va a clonar](#)

[Clonar el repositorio de GitHub a Git](#)

## Clonar el repositorio de GitHub a Git

[Copie la dirección del repositorio GitHub](#)

```
git clone <URL git.git>  
cd proyecto-git
```

En el Github se tiene la opción de bajarlo como zip, pero también está la dirección (url).

Con el nombre que se le dio en github.

## Crear ramas (branch)

Para listar las ramas

[git branch](#)

Crear una rama "rama2"

[git branch rama2](#)

[git branch](#)

Muestra las dos ramas, en verde donde estamos ubicados.

## Cambiar a la rama rama2

[git checkout rama2](#)

[git Branch](#)

Los cambios afectan sólo a la rama activa.