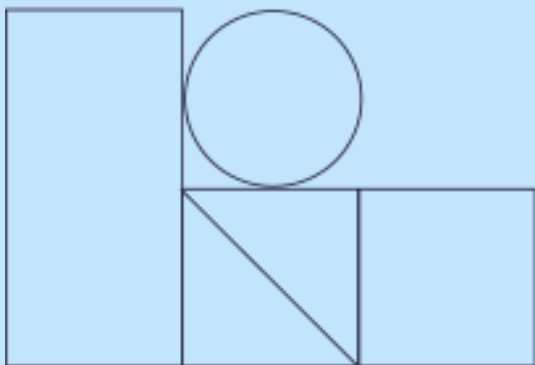


# Git y Github

Puesta en marcha de un repositorio



---

# Índice

Introducción	3
Seguimiento de archivos	3
Copia de respaldo	3

---

# Introducción

Para comenzar un proyecto con *GIT*, abriremos el *Windows Power Shell* o cualquier consola de comandos con la que trabajemos, nos situaremos en directorio de trabajo, es decir, el directorio en el que tenemos nuestro proyecto y desde allí, ejecutaremos el comando:

```
git init
```

Este comando sólo se ejecuta una vez, al principio de nuestro proyecto o cuando queremos que *GIT* comience a hacer un seguimiento de nuestro proyecto.

Introduciendo este comando por consola lo que conseguiremos es que *GIT* cree, de forma transparente al usuario, dos áreas donde irá almacenando los archivos. Estas son el área de ensayo (o *staging area*) y el área de repositorio local.

El **área de ensayo**, como su propio nombre indica es un área donde se van a almacenar los archivos de forma temporal. Este paso es muy útil para que podamos ver qué archivos tienen seguimiento por parte de *GIT*, en qué estado se encuentran...etc.

En el **área de repositorio local** es donde se almacenan esas instantáneas que va haciendo *GIT* a nuestros archivos y es aquí desde donde las podremos rescatar cuando nosotros queramos.

## Seguimiento de archivos

El siguiente paso sería decirle a *GIT* que haga un seguimiento de todos los archivos de mi proyecto (o sólo de alguno en concreto).

Para hacer un seguimiento de todos los archivos existentes en el directorio del proyecto, usaremos el comando:

```
git add .
```

Para añadir solamente un archivo en concreto, añadiremos el nombre del archivo:

```
git add nombreDelArchivo
```

Con esto conseguimos que *GIT* lleve estos archivos desde el directorio de trabajo al área de ensayo y haga un seguimiento de los cambios que se produzcan. No es que los mueva físicamente, los archivos siguen estando en el directorio de trabajo, pero internamente *GIT* los trata como archivos listos para trabajar con ellos en el *staging area*.

A continuación, haremos un:

```
git commit
```

Con lo que nuestros archivos pasarán (de nuevo de forma transparente para nosotros) al área del repositorio local.

Es aquí donde *GIT* crea la instantánea del estado actual de nuestros archivos. Es decir, donde se crea el respaldo (la copia) de nuestro proyecto.

## Copia de respaldo

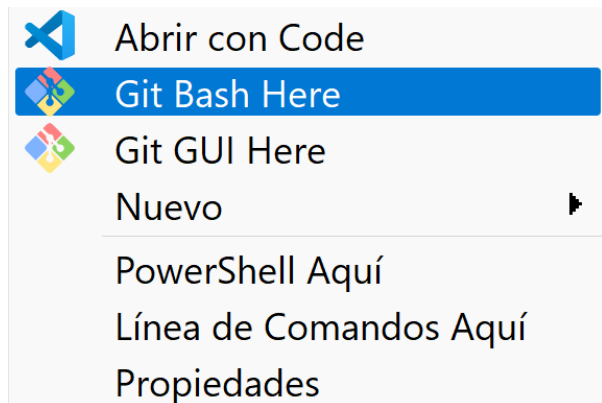
A partir de este momento, podemos obtener una copia de respaldo de cómo se encontraban esos archivos al momento de hacer dicho respaldo.

Veamos un ejemplo de lo dicho anteriormente:

Crearemos un directorio de trabajo para nuestro proyecto al que llamaremos por ejemplo "*Proyecto GIT*".

Crearemos o copiaremos los archivos de nuestro proyecto. Para nuestro ejemplo hemos creado tres archivos, un .html, un .css y un .js

Desde dentro de nuestro directorio de trabajo pulsaremos el botón derecho del ratón y, si hemos hecho la instalación de *GIT* correctamente, no debería salir en el menú la opción “*Git Bash Here*”.



Al pulsar sobre esta opción se nos abrirá la consola de *Git Bash* con la ruta de nuestro proyecto ya predefinida:

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT
```

Ejecutamos el comando `git init` para que *GIT* comience a hacer un seguimiento de nuestro proyecto.

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git init

Initialized empty Git repository in
G:/WORKSPACE/011 Github/Proyecto GIT.git
```

En ese momento *GIT* nos habrá creado una carpeta oculta en nuestro directorio de trabajo con el nombre de “.git”.

Nombre	Tamaño	Modificación	Ancho	Tipo
.git		Hoy 3:20		Archivo carpeta
estilo.css	0 bytes	Hoy 3:35		Documento de hoja de estilos en cascada
index.html	0 bytes	Hoy 3:35		Documento HTML
main.js	0 bytes	Hoy 3:35		Archivo JavaScript

Es la carpeta que usará *GIT* para crear las áreas de ensayo local y de repositorio.

Ahora debemos especificarle a *GIT* sobre qué archivos tiene que hacer el seguimiento:

Para ver el estado en el que se encuentran nuestros archivos usaremos el comando:

```
git status -s
```

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git status -s

?? estilo.css
?? index.html
?? main.js
```

Como podemos ver, nos da un listado de todos los archivos que forman parte de nuestro proyecto.

Los interrogantes ?? nos indican que ninguno de nuestros archivos está siendo sometido a seguimiento por parte de *GIT*.

Para ir practicando, vamos a decirle a *GIT* que le haga un seguimiento sólo al archivo `index.html`

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git add index.html
```

Si volvemos a hacer un “`git status -s`”:

```
miPC@miPC MINGW64 /g/WORKSPACE/011
GitHub/Proyecto GIT (master)

$ git status -s

A index.html
?? estilo.css
?? main.js
```

Veremos que ahora sale una **A** que nos indica que está haciendo el seguimiento del archivo `index.html`. Dicho archivo ahora está en el *staging área*, el área de ensayo.

Una vez que nuestro archivo está en el *staging área*, para hacer una instantánea de su contenido usaremos el comando “*commit*” que veremos en el siguiente tema.