

Descargamos Git desde <https://git-for-windows.github.io/> Existen versiones para Windows, Mac y Linux.

Este Tutorial se realiza desde la versión de línea de comandos Git Bash.

Necesitamos tener una cuenta en Github, cuando la tengáis, decidme vuestro nombre y os añadiré como colaboradores del proyecto para que podáis hacer cambios.

El proyecto de Gestionator está ya en Github, en la dirección:

<https://github.com/Versec/Gestionator>

Una vez instalado Git, iniciamos Git Bash. Git Bash utiliza los mismos comandos que Linux (no los de Windows), pero no nos tenemos que preocupar por eso.

### **Configurar Git con nuestro email.**

Este paso nos lo podemos saltar si queremos. Lo único que pasará es que cada vez que queramos hacer un *push*, nos preguntará por el email y contraseña que introducimos en Github.

Escribimos en la línea de comandos:

```
git config --global user.name "TU NOMBRE DE GITHUB"
```

Después, escribimos la dirección de email que utilizamos en Github

```
git config --global user.email "TU DIRECCIÓN DE EMAIL"
```

Para confirmar que lo hemos hecho bien, podemos escribir en la línea de comandos

```
git config --global user.name
```

Y nos devolverá el nombre de usuario guardado. Lo mismo para el email:

```
git config --global user.email
```

### **Clonar el proyecto**

Viajamos hasta la carpeta donde queramos guardar la versión local de nuestro proyecto con el comando `cd`

```
cd Proyecto_IS
```

Una vez que estemos ahí, tenemos que clonar la versión del proyecto que se encuentra en el servidor remoto de Github a nuestra máquina local. Escribimos en Git Bash:

```
git clone https://github.com/Versec/Gestionator.git
```

Es muy importante que la dirección acabe en **.git**

Una vez clonado el proyecto, tenemos que darle un nombre a la versión del proyecto guardada en el servidor de Github.

```
git remote add origin https://github.com/Versec/Gestionator.git
```

A partir de ahora, nuestra versión remota se llama "*origin*"

Si queremos comprobar el estado de nuestra versión local, podemos hacerlo con el comando

```
git status
```

Nos informará si nuestra versión se encuentra delante o por detrás de la versión remota. Si estamos igual, nos pondrá un mensaje como este:

```
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
```

Hagamos un test sencillo. Crea un documento de texto con tu nombre dentro de la carpeta Gestionator, donde está el proyecto.

Volvemos a utilizar otra vez el comando `git status`, y nos pondrá el mensaje:

```
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    nombreDeFichero.txt
```

Git ha detectado un nuevo fichero (en rojo), pero le tenemos que decir que mantenga un control sobre ("track")

Utilizamos el comando `git add nombreDeFichero.txt` para que Git lo añada.

Si queremos añadir más de un archivo, lo mejor es utilizar el comando...

```
git add -all
```

O el comando

```
git add .
```

Veremos que al parecer, no pasa nada. Pero si utilizamos otra vez el comando `git status` nos aparecerá

```
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   nombredeFichero.txt
```

Si nos hemos equivocado y queremos deshacer el comando 'add', utilizamos

```
git reset HEAD
```

## **Hacer el commit en local**

Para hacer un nuevo commit en nuestra versión local, utilizamos el comando:

```
git commit -m '<Mensaje del commit>'
```

Es recomendable hacer un commit para cada cambio sustancial que se haga, y no mezclar nuevas características o bugfixes.

**Hay que poner siempre un mensaje en el commit. Si no lo ponemos, Git no ejecutará el comando.**

Si hacemos solo '`git commit`', se abrirá Vim en la línea de comandos, y podremos hacer un mensaje más largo.

Para escribir un mensaje en Vim:

1. Entramos en modo de edición presionando primero la tecla 'a'.
2. Escribimos nuestro mensaje. Las líneas que empiecen por una almohadilla, '#' no aparecerán en el mensaje final del commit, son como comentarios en código.
3. Presionamos la tecla de escape 'Esc' para salir del modo de edición.
4. Escribimos ':wq' (dos puntos, y después las letras 'w' y 'q'. Aparecerán en la parte de debajo de la ventana) para guardar el mensaje y salir (**W**rite and **Q**uit).

Aquí hay una lista completa y explicaciones de los comandos de Vim:

[http://www.radford.edu/~mhtay/CPSC120/VIM\\_Editor\\_Commands.htm](http://www.radford.edu/~mhtay/CPSC120/VIM_Editor_Commands.htm)

Si utilizamos el comando `git status` otra vez, nos dice que nuestra versión local se encuentra por delante de la remota:

```
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)
nothing to commit, working directory clean
```

Ahora vamos a subir los cambios para actualizar la versión del servidor

```
git push -u origin master
```

Si lo hemos hecho bien, Git nos informará de que ha comprimido los archivos, y los ha subido sin problemas. Si vamos a la página del proyecto en Github, veremos cómo aparecen los nuevos archivos con nuestro mensaje del commit

### **Actualizar nuestra versión local del proyecto**

Es posible que si alguien hizo algunos cambios antes de que nosotros subiésemos los nuestros, nuestra versión local se encuentre por detrás de la versión remota. Antes de empezar a hacer cambios en el proyecto, es bueno hacer un `pull` primero para asegurarnos de que tenemos la versión más reciente del proyecto:

```
git pull origin master
```

El orden que deberíamos seguir en una sesión de trabajo sería:

1. Pull para descargar la versión más nueva del proyecto a nuestra máquina local.
2. Hacemos los cambios que deseemos.
3. Guardamos y hacemos el commit en local. Se pueden hacer tantos commits como se quiera, pero es recomendable que tengan sentido (es decir, no hagas un commit individual si borras una línea vacía).
4. Hacemos pull otra vez por si alguien ha hecho otros cambios.
5. Hacemos push para subir los datos a la versión remota.