



Contenido de este documento:

- Trabajo con Git
- Procedimiento de trabajo con Git y GitHub en Visual Studio Code

## Trabajo con Git

### ¿Qué es Git?

Imagina que estás desarrollando un programa y escribiendo nuevo código, y en un momento dado decides que el camino que has emprendido no es el adecuado y quieres volver a la versión anterior; o imagina que estás trabajando en equipo con otros desarrolladores y tenéis que unir el código desarrollado por cada miembro del equipo.

Estas tareas son muy habituales y no son nada triviales, y aquí es donde Git acude en nuestra ayuda. Git es un sistema de control de versiones que permite llevar un registro de los archivos de un proyecto y coordinar el trabajo de todos los miembros de un equipo de desarrollo.

### ¿Para qué vamos a utilizar Git?

En nuestra asignatura te pedimos que desarrolles unos proyectos en las clases de laboratorio. Tu profesor puede pedirte que le envíes el trabajo que has realizado para revisarlo, o tú mismo querrás que él lo revise para ver si está todo correcto, o para que te ayude a resolver posibles dudas.

Puedes pensar, y no te falta razón, que enviar el proyecto a tu profesor y decirle cuáles han sido los últimos cambios que has hecho en él es una tarea complicada. Ahora, imagínate que existe un botón que al pulsarlo hace que tu profesor tenga acceso, de forma inmediata y desde su ordenador, a tu proyecto. Y todo ello sin salir del entorno de desarrollo. Pues bien, esto lo podemos hacer combinando Git con una herramienta denominada GitHub Classroom y el sitio web de GitHub, que usaremos para alojar una copia remota de nuestro proyecto. Y esto es solo una de las muchas cosas que podemos hacer con esta herramienta, pero de momento vamos a aprender a hacer bien esto, y el resto de las funciones ya las irás aprendiendo más adelante, y será algo que usarás, con toda seguridad, en tu futuro trabajo.

### ¿Cómo funciona Git?

El funcionamiento básico de Git lo puedes ver en el diagrama de la Figura 1. Como verás, hay cuatro elementos:

1. El directorio de trabajo, que es donde creas tus proyectos;
2. el índice, que es donde indicas los ficheros del proyecto a los que quieres realizar un seguimiento;
3. el repositorio local, que es una carpeta de tu disco donde se almacena una copia del proyecto que quieres gestionar con Git, junto con toda la información que necesita Git para registrar el historial de cambios, y
4. el repositorio remoto, que es una copia del proyecto que se almacena en un servidor remoto, y que te permite compartirlo con otros usuarios (tu profesor, por ejemplo).

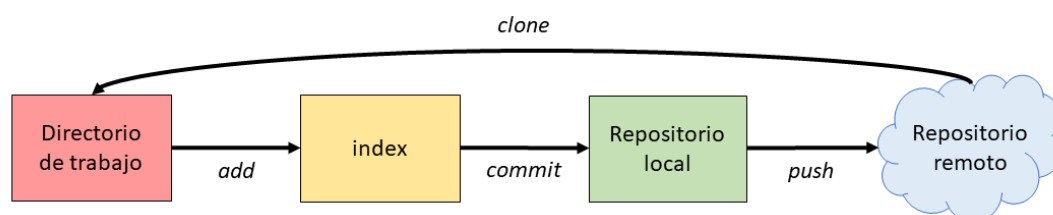




Figura 1. Esquema de funcionamiento de Git

Durante tu trabajo normal, estarás escribiendo código en tu espacio de trabajo, creando y modificando ficheros. Algunos de estos ficheros querrás que se transfieran al repositorio remoto, por lo cual a estos ficheros les harás un seguimiento; esta operación se conoce como *add*. Periódicamente, podrás transferir los cambios realizados en estos ficheros al repositorio local; esta operación se conoce como *commit*. Cuando hayas finalizado la tarea que estás haciendo y quieras compartirla, la subirás al repositorio remoto; esta operación se conoce como *push*.

Otra operación que puedes hacer es copiar el repositorio remoto en tu máquina local (por ejemplo, si has tenido un problema con tu ordenador, o te cambias a uno nuevo). Esta operación se conoce como *clone*.

### ¿Qué necesito para trabajar con Git y GitHub?

Para trabajar con Git y alojar tus proyectos en GitHub debes disponer de una cuenta en GitHub, que es gratuita y simple de crear. Quizás tengas ya una cuenta; en ese caso, puedes usarla y no necesitas crear otra. En caso contrario, debes crear una. Al crearla debes indicar un nombre de usuario y un correo electrónico. Es muy recomendable que utilices tu usuario virtual de la Universidad de Sevilla (*uvus*) -salvo que ese nombre ya esté cogido en GitHub- y tu correo electrónico de la Universidad de Sevilla (*usuario@alum.us.es*).

### ¿Cuál es el primer paso para comenzar el trabajo?

Tu profesor creará una clase a través de *GitHub Classroom*, en la que te incluirá junto al resto de tus compañeros de laboratorio. Dentro de esta clase creará tareas que asignará a todos los estudiantes. Habrá una tarea por cada proyecto. Cuando te asigne una tarea, recibirás un enlace a modo de invitación que deberás aceptar. Entonces, se creará un repositorio remoto en el servidor GitHub. Luego entrarás en tu IDE, y mediante la operación de clonado, se hará una copia de ese repositorio en tu espacio de trabajo.

### ¿Cuál será la forma de trabajo?


Trabajarás en tu proyecto de forma normal. Cuando implementes una funcionalidad nueva en tu proyecto, o realices una modificación importante, harás una anotación de los cambios en el repositorio local (operación *commit*). Esto te permitirá deshacer más tarde los cambios si lo necesitas. Cuando tengas que enviar el trabajo al profesor porque quieres que revise lo que has hecho, volcarás los cambios realizados en tu proyecto sobre el repositorio remoto (operación *push*). El profesor recibirá una notificación, podrá revisar el proyecto y enviarte sus comentarios (*feedback*).



## Procedimiento de trabajo con Git y GitHub en Visual Studio Code

A continuación, te mostramos la forma de trabajar con Git y GitHub desde Visual Studio Code. También puedes ver el procedimiento completo en el vídeo <https://www.kaltura.com/tiny/yv925>.

### Antes de comenzar

Para poder trabajar con GitHub desde Visual Studio Code debes instalar antes el software Git. Para ello, pulsa en el botón *Source Control* del panel lateral de Visual Studio Code (  ) y busca el enlace *install git*. Descarga la versión correspondiente a tu sistema operativo, ejecuta el archivo descargado y sigue las instrucciones de instalación. Deberás reiniciar Visual Studio Code para que detecte la configuración de Git.

Un paso más que debes hacer es configurar tu nombre de usuario y tu dirección de correo, ya que esa información se utilizará para sincronizar los cambios entre el repositorio local y el remoto. Para ello, debes abrir una ventana de terminal y teclear lo siguiente:

```
git config --global user.name "nombre_usuario"  
git config --global user.email "correo_electrónico"
```

donde *nombre\_usuario* y *correo\_electrónico* son el nombre de usuario y la dirección de correo, respectivamente, de tu cuenta de GitHub.

### Paso 1: aceptar la tarea

Tu profesor habrá creado una clase con todos los alumnos del laboratorio mediante la herramienta GitHub Classroom. Dentro de ella, creará una tarea para cada proyecto, y te enviará un enlace para que aceptes la tarea. Esta invitación es un enlace que debes copiar y pegar en tu navegador.

Cuando se abra el enlace, se te pedirá identificarte en GitHub si no lo has hecho antes. Una vez identificado, si es la primera invitación que recibes, te aparecerá una lista. Busca en ella tu nombre y selecciónalo. En la ventana que se abre, asegúrate bien de que has elegido tu nombre y no el de un compañero, y pulsa el botón Aceptar. Al hacerlo, te unirás a la clase de GitHub Classroom.

Una vez añadido a la clase, pulsa el botón para aceptar la tarea. Si ya estabas incluido en la clase, este botón te aparecerá directamente cuando abras el enlace con la invitación a la tarea. Al aceptar la tarea se crea un repositorio remoto en GitHub, y se muestra el enlace al repositorio (puede ser necesario refrescar la página para que aparezca). Copia este enlace porque te hará falta en el siguiente paso.

### Paso 2: crear el proyecto

Abre Visual Studio Code. Pulsa en el botón *Source Control* del panel lateral y pulsa en *Clone repository* y después en *Clone from GitHub* (también puedes irte a *Help > Show All Commands > Git clone > Clone from GitHub*). Pega el enlace a tu repositorio que copiaste en el paso anterior. Te pedirá una ubicación en el disco para crear el proyecto. Se copia el repositorio en esa carpeta, y te pregunta si quieres abrirlo. Pulsa en *Open*, y se abrirá el repositorio en Visual Studio Code.

La primera vez que te conectes desde Visual Studio Code a GitHub se te pedirá que te autentifiques. Para ello, tienes que pulsar el botón de abrir el navegador, introducir tu usuario y contraseña de GitHub si aún no estabas logueado, y pulsar en "autorizar a Visual Studio Code a usar GitHub".



### Paso 3: trabajar en el proyecto

Ya tienes tu proyecto en Visual Studio Code, y ahora puedes trabajar en él, escribiendo y probando tus módulos Python. Cuando hagas cambios, verás que aparece un número en el icono de *Source Control*. Si pulsas sobre él, verás una sección *Changes* con los ficheros que se han modificado. Si seleccionas los ficheros y pulsas el botón +, verás que los ficheros se añaden a la sección *Staged Changes*. Esto indica que están listos para ser copiados al repositorio remoto. Para hacerlo, escribe un mensaje breve en el cuadro de texto superior, describiendo los cambios que has realizado, y pulsa el botón *Commit*. Verás que el botón cambia ahora a *Sync Changes*. Si lo pulsas, los ficheros se publicarán en el repositorio remoto una vez que confirmes la operación. Puedes hacer la subida de una vez si en lugar de pulsar *Commit* abres el menú desplegable y seleccionas *Commit & Push*.

### Paso 4: ver los comentarios del profesor

Cuando tu profesor revise tu trabajo y te envíe sus comentarios, recibirás un correo electrónico. Pincha en el enlace del correo, o entra en GitHub y ve a tu repositorio. Pulsa en *Pull requests* y en *Feedback*, y observa los comentarios del profesor. Realiza los cambios propuestos y haz de nuevo *Commit & Push*. Este proceso se podrá repetir hasta que el proyecto esté finalizado. Entonces el profesor te pedirá que cierres la tarea pulsando *Merge pull request* y *Confirm merge*. Es importante que no hagas esto hasta que te lo pida.