



Añadiendo GIT a nuestro proyecto de angular

El primer paso que debemos realizar es el de comprobar que tenemos Git instalado en nuestro equipo. Para ello podemos ejecutar el siguiente comando en la consola del sistema (node command prompt en Windows) o si lo preferís, en el terminal de Visual Studio Code.

```
git -versión
```

En caso de no tenerlo los descargamos de la página oficial (revisar la instalación para los distintos sistemas)

Ahora tienes que abrir el símbolo de sistema y escribir los siguientes comandos en la terminal:

```
git config --global user.name "Tu nombre"
```

```
git config --global user.email ejemplo@email.com
```

Recuerda que debes de cambiar Tu Nombre y ejemplo@email.com por tu información.

Ahora que tenemos nuestro proyecto de Angular listo, nos posicionamos en el terminal y escribimos el siguiente comando:

```
git init
```

Este comando inicializará Git en nuestro proyecto, creando además una carpeta oculta `.git` y un fichero `.gitignore` donde incluiremos los ficheros que no deseamos que se suban al repositorio. Si ya teníamos Git instalado y en función de su configuración, es probable que al crear el proyecto de Angular ya nos haya creado automáticamente la carpeta y el fichero `.gitignore`.

```
git status
```

Nos indica que ficheros son nuevos o han sido modificados y aun no se han confirmado (committed) al repositorio.

```
git add .
```

Añade todos los ficheros y carpetas al commit, es decir, estamos indicando que estamos de acuerdo con los cambios que hemos realizado y que hemos visto en el `git status` y los preparamos para su posterior subida.

```
git commit -m «Initial commit»
```

Confirmamos los cambios añadiéndoles un mensaje explicativo.

Al ejecutar estos parámetros y confirmar los cambios, veremos como en el Visual Studio Code desaparece cualquier indicativo visual de cambios, y que, si volvemos a lanzar el comando `git`



Taller de programación

status, no aparecen cambios. Esto es así porque al hacer el commit, hemos indicado que, desde ese momento, ese es el código base, y todo lo que hagamos después, serán nuevos cambios que deberán de subirse con otro commit.

Es importante recalcar que todo este proceso confirma los cambios en tu rama local, en el código que te has descargado, no en código del proyecto ubicado en el repositorio remoto. De hecho, durante este post aún no hemos vinculado nuestro proyecto a un repositorio como GitHub.

```
git remote add origin https://...
```

Con este comando, estamos indicando el repositorio al que queremos subir nuestro proyecto.

```
git branch -M main
```

Creamos una nueva rama con el código que tenemos llamada “main” y nos movemos a ella.

```
git push -u origin main
```

Subimos el código. El comando push sigue el esquema «git push -u [remote name] [branch name]». El parámetro -u se usa la primera vez para referenciar el repositorio al que vamos a realizar las subidas posteriores de código o siguientes push.

Nota: Si es la primera vez que realizamos este proceso, el terminal nos solicitará que ingresemos el usuario y la contraseña de GitHub para verificar que tenemos acceso. Una vez introducidas, se realizará la subida de código de forma automática.