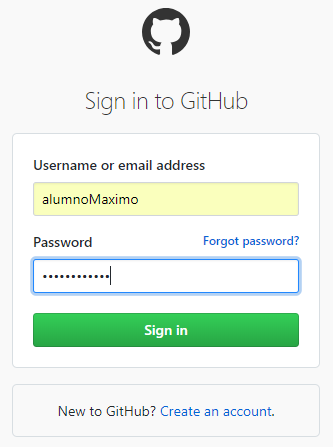
**Versionado de Proyectos: Ejemplo de uso de git Paso a Paso**

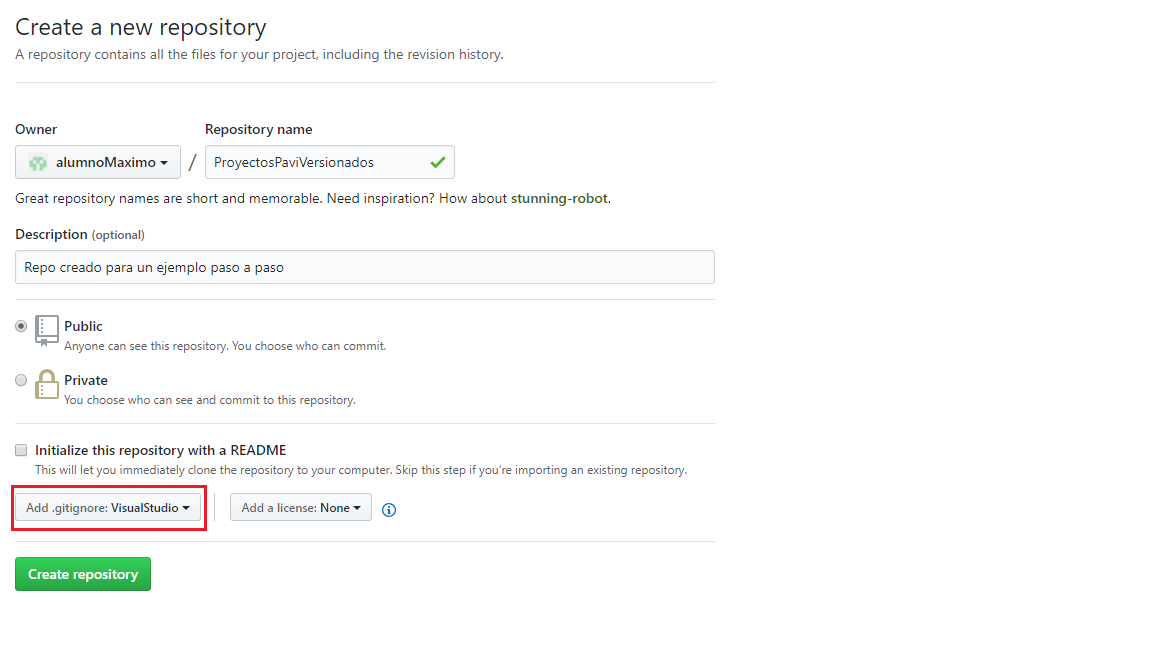
En el presente documento se detalla cómo compartir proyectos entre al menos dos colaboradores.

Podes seguir los pasos de los alumnos creados a modo de ejemplo: alumnoMaximo y alumnoMilena.

Ambos han creado sus respectivas cuentas en https://github.com



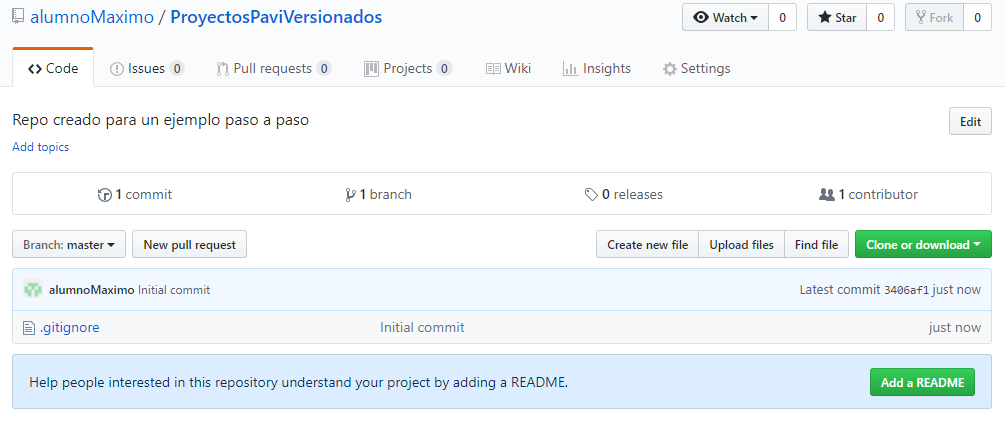
AlumnoMaximo crea un repositorio público ProyectosPaviVersionados



**Importante!** Es conveniente indicar que archivos no queremos que se intercambien entre el repo en la nube y la ubicación local. Para ello es conveniente crear un archivo .gitignore (la opción gitignore Visual Studio facilita esta tarea).

Presiona Crear repositorio.

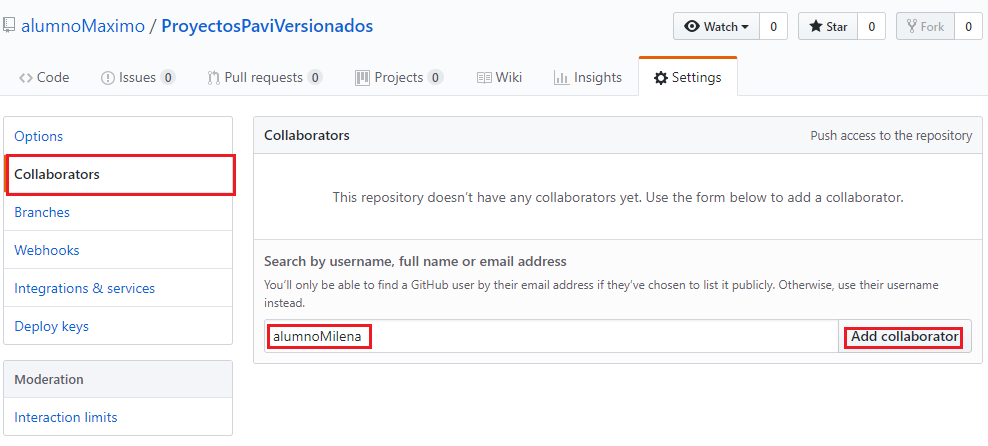
<https://github.com/alumnoMaximo/ProyectosPaviVersionados>



En el repo ya tenes un archivo versionado y un commit hecho (Initial commit).

Hasta aquí el repositorio solo está en la nube y no guarda relación con lo que tenemos en el equipo local.

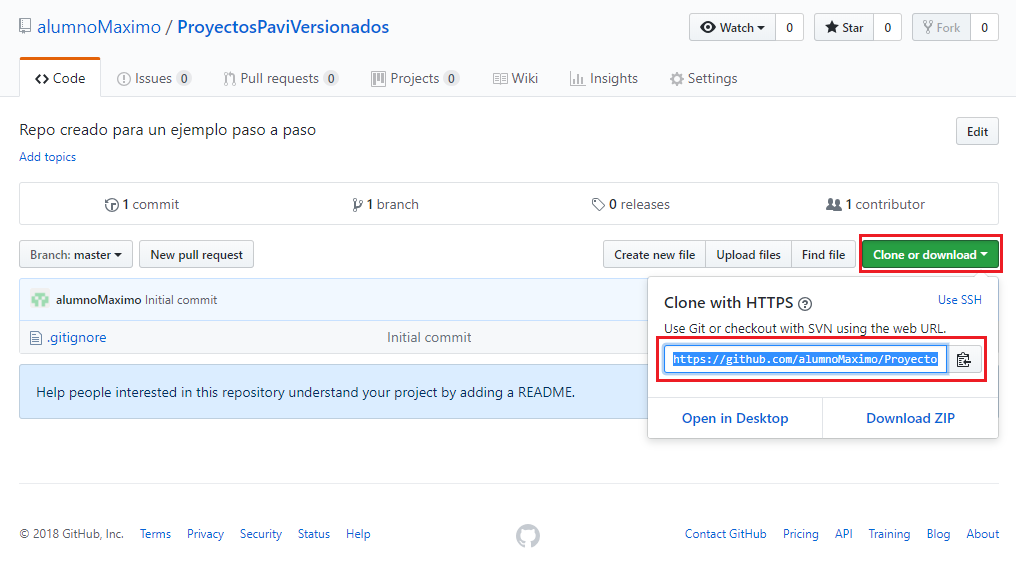
Maximo puede agregar a todos los colaboradores que desee. Para esto…



En el equipo local de Maximo creo la carpeta donde voy a trabajar. En el ejemplo c:/proyectos\_git. En esta carpeta podes clonar el repositorio que solo contiene el archivo .gitignore

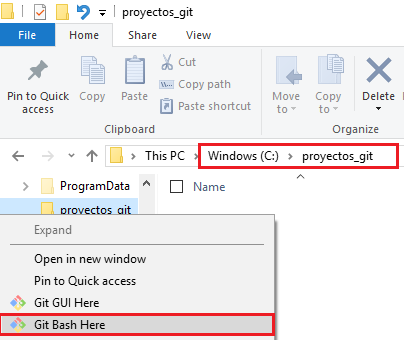
Accede a <https://github.com/alumnoMaximo/ProyectosPaviVersionados> y selecciona “Clonar o descargar” y copia la url del repo

|  |
| --- |
| **Importante:**   1. **git clone configurará automáticamente tu repositorio con un control remoto apuntando a la URL de Git desde la que lo clonó. Esto significa que una vez que realices cambios en un archivo y los comprometa, podes enviar esos cambios al repositorio remoto.** 2. **Si usaste git init para hacer un nuevo repositorio, no tendrás ningún repositorio remoto para realizar cambios. En el caso de usar git init, ademas es necesario usar git remote add origin https://github.com/alumnoMaximo/ProyectosPaviVersionados.git** |



url del repo: <https://github.com/alumnoMaximo/ProyectosPaviVersionados.git>

1. **Git clone (sin git init)**
2. Abrir gitbash en la carpeta c:/proyectos\_git

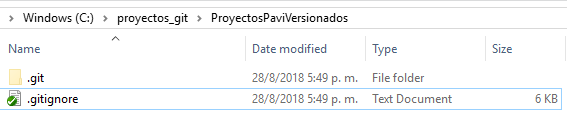
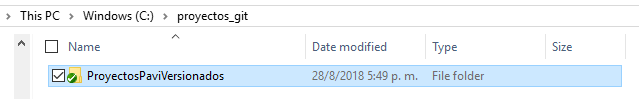


1. Ejecutar el comando git clone para traerse el repo a local.

$ git clone <https://github.com/alumnoMaximo/ProyectosPaviVersionados.git>

|  |
| --- |
|  |
| Comando git clone |

Como resultado se creará una carpeta nueva en local con el contenido del repo.



1. Moverse a la carpeta del repo clonado

$ cd ProyectosPaviVersionados

|  |
| --- |
|  |
| Carpeta del repo clonado |

1. Verificar que se conecta al repo remoto.

$ git pull

|  |
| --- |
|  |
| Comando git pull |

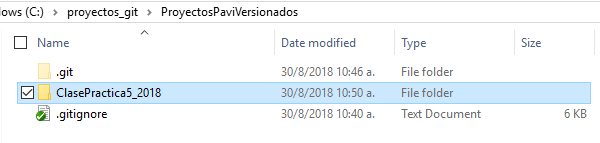
1. Setear localmente los valores de configuración con los comandos git config –local

$git config -- local user.name alumnoMaximo

$git config -- local user.email [alumnoMaximo@gmail.com](mailto:alumnoMaximo@gmail.com)

|  |
| --- |
|  |
| Comando git config –local/global – Comando git config --list |

1. Mover a la carpeta ProyectosPaviVersionados los archivos que queres versionar. En este caso la carpeta “ClasePractica5\_2018”.



1. Con el comando git status, visualizar el estado de los cambios en local (archivos nuevos).

$ git status

|  |
| --- |
|  |
| Comando git status |

Como podes ver la carpeta “ClasePractica5\_2018” está entre los archivos no trackeados.

1. Agregar los archivos que se quieren versionar y compartir con el comando git add.

$git add .

$git add “nombre del archivo”

|  |
| --- |
|  |
| Comando git add . |

f) Con el comando git commit –m se comprometen los cambios.

$ git commit –m “clase 5”

|  |
| --- |
|  |
| Comando git commit –m |

1. Con el comando git push envias las cambios comiteados a la rama master del repo remoto.

|  |
| --- |
|  |
| Comando git push |

En la ventana de comandos o en una ventana se te solicitarán las credenciales github para acceder al remoto (en este caso las de alumnoMaximo).

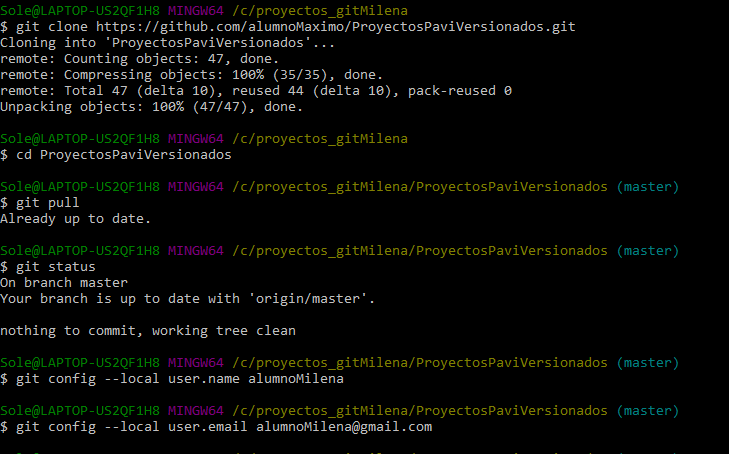
|  |
| --- |
| **Importante:** Si ya tuvieras otro usuario de github seteado a nivel global intentará usar esas credenciales para conectarse, para borrarlas en Window vas a Panel de Control – Cuentas de Usuario – Administrador de Credenciales – Credenciales Genéricas – Llaves Github - Remover |

El usuario alumnoMilena repite las mismas operaciones:

git clone

git pull

git config – local



Realiza cambios en cualquier archivo y repite el circuito

git pull (traer cambios del repo remoto)

git status

git add

git commit

git push (enviar cambios al repo remoto)

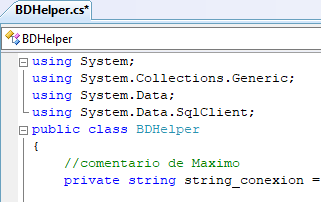
**Resolución de conflictos**

Puede ocurrir que al menos dos usuarios (colaboradores) modifiquen el mismo archivo, en este caso el DBHelper.cs

|  |
| --- |
| Todo aquello que sea conflictivo y no se haya podido resolver, se marca como "sin fusionar" (unmerged). Git añade a los archivos conflictivos unos marcadores especiales de resolución de conflictos. Marcadores que te guiarán cuando abras manualmente los archivos implicados y los edites para corregirlos. El archivo conflictivo contendrá algo como: |

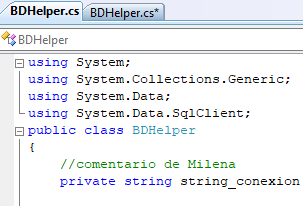
Ejemplo:

Maximo y Milena comentan la misma línea



|  |
| --- |
|  |
| Trabajo de Maximo |

|  |
| --- |
|  |
| Trabajo de Milena – se denota un conflicto y git trata de resolverlo |



Para resolverlo hay que quitar las marcas dejando lo que se desea y luego agregar, comitear, enviar el cambio.

|  |
| --- |
|  |
| Conflicto |

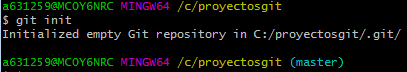
Milena ingresa al archivo, acuerda con Máximo que dejar. Lo agrega, comitea y lo envía al repositorio remoto.

|  |
| --- |
|  |
| Edición del archivo para resolver el conflicto. |

|  |
| --- |
|  |
| Milena envía el conflicto resuelto. |

1. **Git clone (con git init)**

Comando git init, por defecto la rama en local es master.



$git init

Para inicializar el repositorio

$git config --list

Configuración de todos los repositorios a nivel local.

$git clone https://github.com/alumnoMaximo/ProyectosPaviVersionados.git

Cloning into 'ProyectosPaviVersionados'...

remote: Counting objects: 3, done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

Sole@LAPTOP-US2QF1H8 MINGW64 /c/proyectos\_git (master)

$ git pull

There is no tracking information for the current branch.

Please specify which branch you want to merge with.

See git-pull(1) for details.

git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

git branch --set-upstream-to=<remote>/<branch> master

Usar el commando git remote add origin

Sole@LAPTOP-US2QF1H8 MINGW64 /c/proyectos\_git (master)

$ git remote add origin <https://github.com/alumnoMaximo/ProyectosPaviVersionados.git>

Sole@LAPTOP-US2QF1H8 MINGW64 /c/proyectos\_git (master)

$ cd ProyectosPaviVersionados

Sole@LAPTOP-US2QF1H8 MINGW64 /c/proyectos\_git/ProyectosPaviVersionados (master)

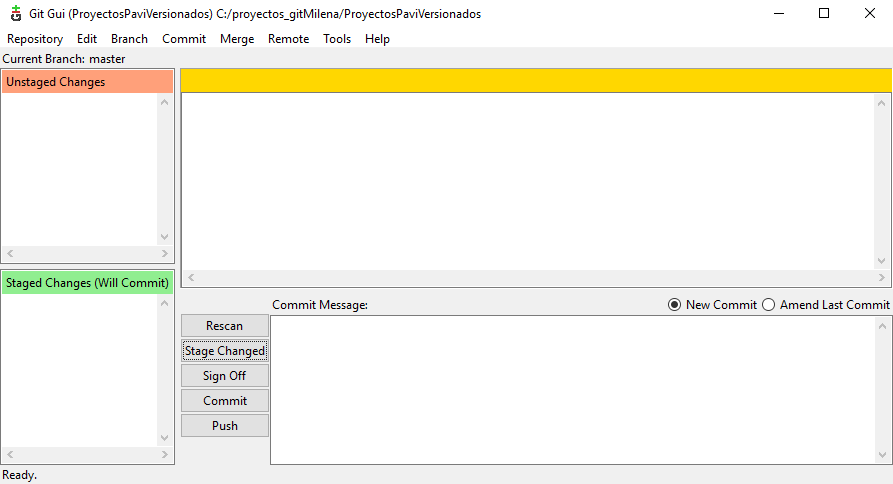
$ git pull

Already up to date.

Otras herramientas

También podes usar herramientas gráficas:

* Git GUI
* Complementos instalados al instalar Visual Studio



* TortoiseGit (https://tortoisegit.org/)
* SourceTree (https://www.sourcetreeapp.com/)

Referencias:

https://git-scm.com/docs

<https://git-scm.com/book/es/v1/Fundamentos-de-Git-Trabajando-con-repositorios-remotos>

https://git-scm.com/book/es/v1/Ramificaciones-en-Git-Procedimientos-b%C3%A1sicos-para-ramificar-y-fusionar