

Que es GIT

**GIT** es un Sistema de Control de Versiones distribuido libre y open source escrito en lenguaje C.

El objetivo de Git es la posibilidad de trabajar e interactuar todos los integrantes del grupo, teniendo acceso a todo el código , documentación y todos los archivos que se suban al repositorio.

Git se basa en que cada usuario almacena una copia completa del repositorio en su máquina de forma local, incluido el historial de cambios. Esto implica que muchas de las operaciones realizadas sobre el código no tienen lugar en la red, permitiendo que la velocidad de proceso dependa únicamente en los recursos locales.



**GitHub** es un hosting online para nuestros repositorios que utiliza Git para el mantenimiento y versionado del código fuente, brindando servicios extras para la gestión del proyecto y el código fuente. La parte gratuita de este hosting permite repositorios públicos, si queremos repositorios privados entramos en la parte “premium” del servicio.

Utilizaremos GitHub a traves de dos aplicaciones

Web : Control de Proyecto + cards + issues

<https://github.com>

Desktop : Desarrollo

La url para instalar GitHub desktop es <https://desktop.github.com/>

Los términos a utilizar en GitHub

**Repositorios** : Directorios donde se almacenan nuestros proyectos y toda la información sobre sus modificaciones .

Cada Repositorio tendrá un **Owner** (dueño) el creador del mismo y tendrá **Collaborators** (colaboradores) que serán los otros integrantes del grupo.

**Clone :** Mediante este comando, estaremos realizando una copia del Repositorio OnLine a nuestra PC.

**Commit:** Una vez que hemos realizado los cambios en nuestro proyecto, todos estos aparecen como modificaciones a subir.

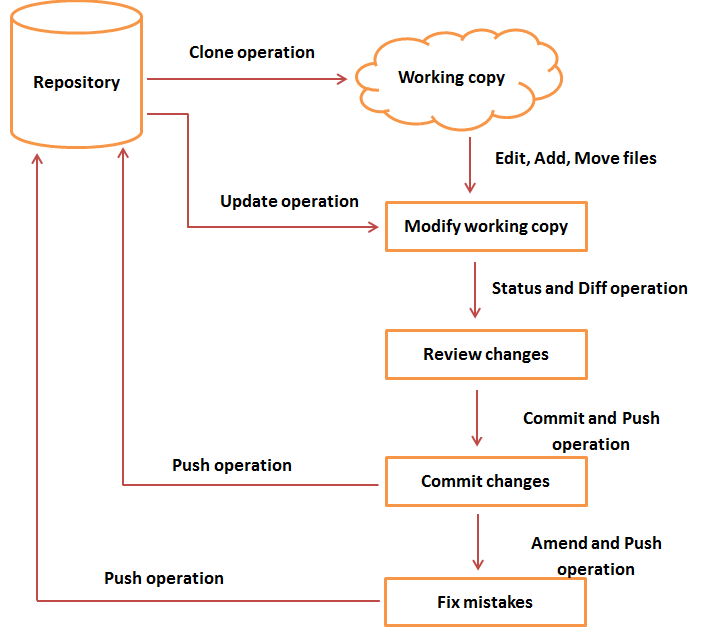
Para ello deberemos, SIEMPRE , agregar un comentario con el cual identificaremos estas modificación en caso de querer volver atrás la misma en caso de necesidad.

**Publish :** Todas las modificaciones estarán en StandBy en nuestro repositorio local , hasta que hagamos Publish, donde estas serán sincronizadas con nuestro repositorio OnLine.

## Flujo de trabajo general es el siguiente:

* Clonar el repositorio Git como una copia de trabajo.
* Modificar la copia de trabajo mediante la adición / edición de archivos.
* Si es necesario, también se actualiza la copia de trabajo mediante la adopción de cambios de otros desarrolladores.
* Revisar los cambios antes de subirlos.
* Confirma los cambios. Si todo está bien, a continuación, se presiona a los cambios en el repositorio.
* Después de commit, si se da cuenta de que algo está mal, entonces se corrige la última actualización y luego realizar un nuevo commit al repositorio.

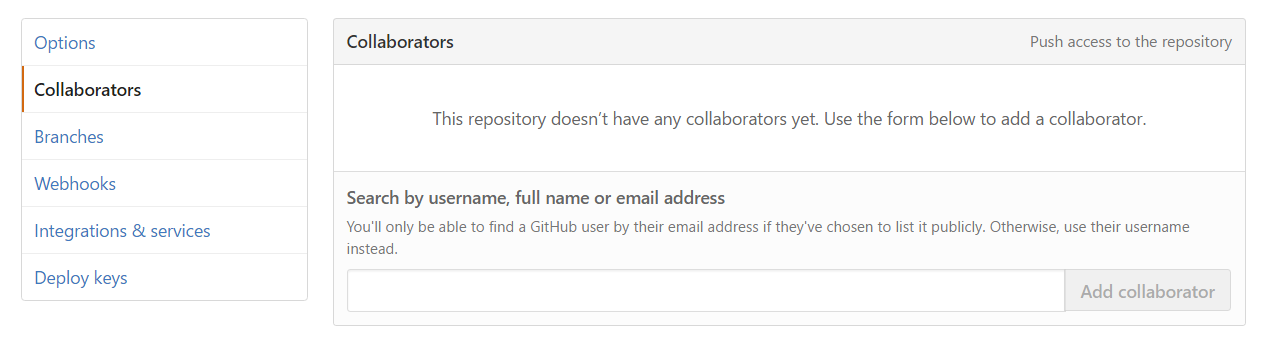
Representación gráfica del flujo de trabajo.



Forma de trabajo en Proyecto Final con Git + GitHub

Creación de Cuentas y Repositorio

1. Todos los integrantes del grupo deben crear un usuario en GitHub.com
2. Un integrante debe crear el repositorio del proyecto ( Uso de Organizaciones????? )
   1. Deberá agregar como colaboradores a los otros integrantes



1. *La información de los repositorios, será medida de evaluación del avance del grupo*

Trabajo Diario

1. Clonar el repositorio al iniciar el dia utilizando GitHub desktop o quienes utilizan Visual Studio, tendrán integrado en el framework el uso de GitHub
2. El repositorio local deberá estar en el disco D
3. Trabajar mucho ! Hacer modificaciones !
4. Hacer commits, poner comentarios claros y luego publish.
5. Intentar hacer commits de modificaciones pequeñas y no luego de un monton de desarrollo, ya que esto perjudica la posibilidad de volver a una versión anterior en caso de necesitarlo.
   1. Aca pueden surgir conflictos, que no permitirán el commit

Los conflictos aparecerán marcados en los archivos que tienen modificaciones en las mismas líneas. Se deberán modificar y definir cuál es el correcto y luego realizar el commit.

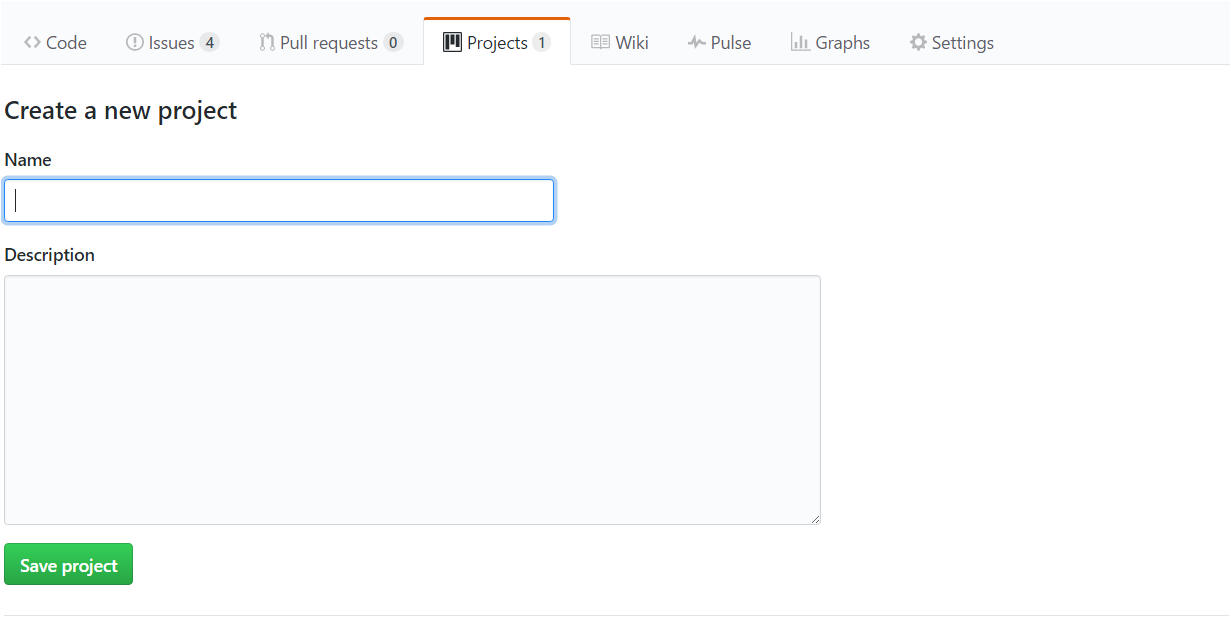
1. Verificar en GitHub.com , que hayamos realizado correctamente el procedimiento de Commit - Publish.

## 

## Project Management

La administración del proyecto, es la base fundamental para lograr el éxito del producto.

Para esto utilizaremos el método de tarjetas o kanban , a traves de github.com , que brinda esta posibilidad utilizando la seccion ***Projects ,*** alli crearemos nuestro proyecto, asignándole un nombre y la descripción del mismo.



El método de tarjetas , consiste en crear tarjetas que representan, tareas , funcionalidades o grupo de funcionalidades , que serán agrupadas en columnas mostrando su estado .

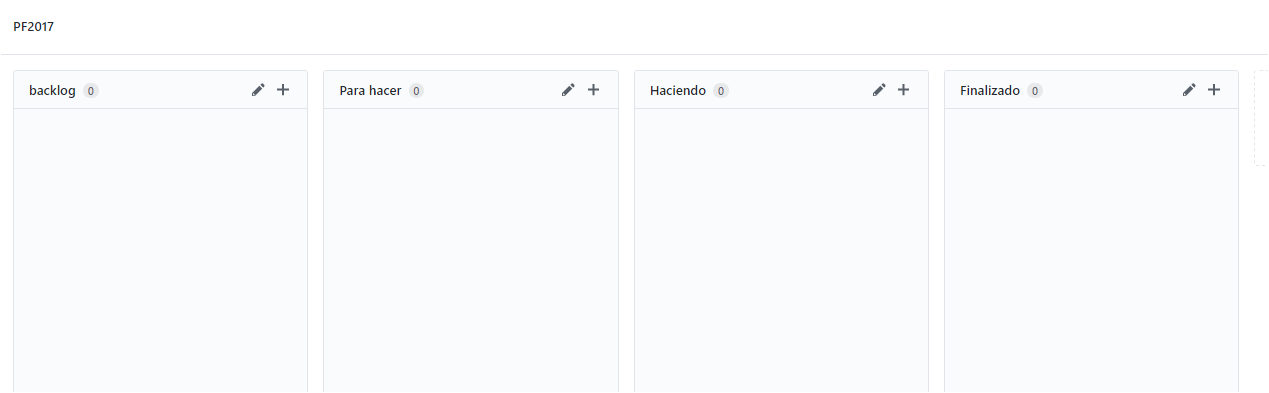
Utilizaremos 4 columnas para trabajo en cada iteración.

Backlog - Todas las tareas a realizar

Hacer - Tareas a hacer en la iteración

Haciendo y Finalizadas

Al finalizar cada iteración , crearemos una columna donde pondremos todas las tarjetas que finalizaron en la misma .



Para crear las tarjetas , github , posee un elemento llamado CARD, donde pondremos el título de la tarea, luego la convertimos en ISSUE ( Versión mejorada de la Card) , donde nos permitirá agregar mucho información a la misma, checklists , asignarla a un usuario y mucho mas.

Sites

<https://github.com>

<https://git-scm.com>

<https://desktop.github.com/>