

# Resumen acerca de GitHub

Escrito por: Daniela Martínez Madrid.

Este resumen es creado con la intención de recopilar información básica y relevante, fruto de un primer acercamiento a la plataforma, que además es un requerimiento del curso de programación avanzada del centro de investigación CINVESTAV, dirigido por el profesor Dr. Feliu Sagols.

Ciudad de México, marzo 2022.

GitHub es una plataforma de desarrollo colaborativo, en la cual se puede almacenar repositorios en la nube (que en su mayoría son proyectos de código), que pueden permanecer privados o ser compartidos con colaboradores. Esta plataforma está basada en el sistema GIT, un software de control de versiones distribuido, que permite a desarrolladores trabajar de forma simultánea y segura, en un proyecto en común sin compartir una misma red.

El objetivo principal de este sistema es llevar un registro de todas las modificaciones que se hacen en un proyecto, estas se guardarán en archivos y en caso de ser necesario, podrán ser recuperadas y rastreadas, evitando así: la sobreescritura de códigos, la pérdida de archivos y la aparición de errores que ya habían sido resueltos. Lo que hace de esta plataforma una herramienta fundamental en la gestión y organización de proyectos.

Adicionalmente, cuenta con un componente de red social, que permite conectar a desarrolladores de todo el mundo. Los proyectos pueden ser expuestos al público, brindando no solo un conocimiento a su población de interés, si no que ésta a su vez, propone mejoras o cambios de código. Las discusiones que se generan en torno a esto logran crear una comunidad de desarrolladores encaminados hacia la construcción de un conocimiento colectivo.

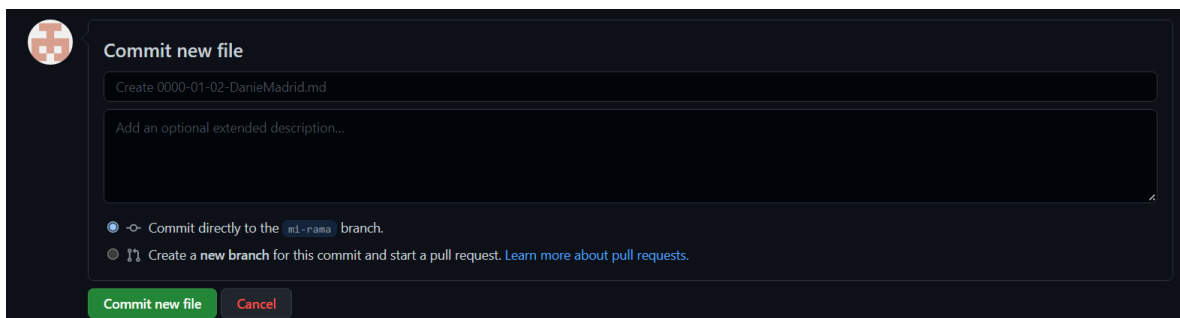
Si el interés del usuario es tener un repositorio para su proyecto, lo primero que debe de hacer es crearlo, subiendo un archivo o una colección de archivos. Previo a esto, tener conocimiento básico de los conceptos y de las herramientas que aparecen en la plataforma.

Las herramientas destacadas que ofrece GitHub son: issues, discusiones, pull request, notifications, labels, actions, forks y projects.

Issues es el lugar donde se concentra la comunicación de la comunidad del proyecto, ya sea con el (o los) colaborador(es) del repositorio (que son el (o los) administrador(es)), o en el caso de ser público, cualquier usuario de GitHub. Los issues pueden ser preguntas, tareas, ideas, errores o mejoras sugeridas dentro del proyecto. Estos pueden ser asignados a un desarrollador, el cual se hará responsable de su seguimiento. Como consecuencia a esto se obtiene un registro de la manera como surgen algunas ideas y de los argumentos a sus soluciones, además tiene la ventaja de que en este lugar se puede visualizar en tiempo real los issues que siguen abiertos o si han sido cerrados.

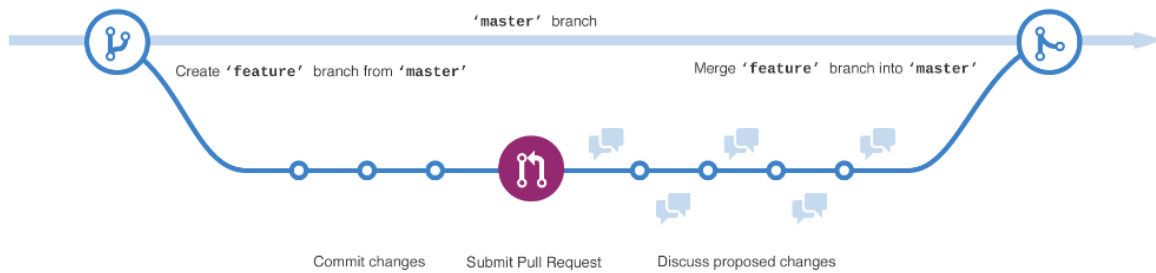
El flujo de trabajo en equipo en la plataforma GitHub consiste de una rama principal, denominada Main (en versiones anteriores conocida como Master) y las nuevas ramas, denominadas ramas de características (features Branch). Cuando iniciamos un proyecto, toda la información estará depositada en el main, el cual se irá nutriendo con los avances de los desarrolladores. Para lograr que las contribuciones se hagan de forma simultánea, sin interferir en el código fuente, se crean ramas características, que generan diferentes versiones de un repositorio.

Los cambios en el Main o en las ramas características son guardados en los llamados commits, los cuales se visualizarán con información relevante: asunto y una breve descripción. Lo que permite filtrar información de una manera más optima.



Una vez que se han considerado los cambios suficientes para una rama, se propone un pull request, una herramienta creada para integrar cambios a la fuente principal Main, sin embargo, antes de ser aceptados deben de pasar por el filtro del criterio de los desarrolladores, los cuales realizaron un feedback del trabajo realizado.

La dinámica antes de descrita se puede visualizar sintetizada en el siguiente gráfico:



Los labels son útiles para categorizar y organizar la información en los issues y pull request, los mas usados son: bug, doumentation, duplicate, help wanted, enhancement y question.

GitHub actions es una herramienta que sirve para automatizar tareas y workflows de los programadores, permite crear workflows que se puedan utilizar para compliar, testear y desplegar código, es decir, se puede usar para implementar la integración contiua (CI), desarrollo continuo y entrega continua (CD). Una ventaja que ofrece GitHub en Actions es el uso de plantillas, donde se usa paquetes de código que se ejecutan en los servidores de GitHub, y que son compatibles con cualquier lenguaje de programación.

Por último vamos a hablar de la herramienta Clone y Fork, que son una forma de poder copiar un proyecto. Si hacemos una copia con la herramienta Clone, se hará una copia del repositorio en la propia maquina local, en cambio el Fork hace una copia del repositorio en la la cuenta propia de GitHub. Esto tiene repercusiones en los permisos que se requieren para efectuar cambios en los diferentes repositorios que van a estar involucrados. Por parte del Clone, si se tiene acceso de escritura se puede generar directamente al repositorio original los cambios, sin embargo, en el Fork se crea una copia del repositorio remoto en la cuenta de GitHub, por lo que se requiere un pull request en el caso de querer contribuir en el repositorio original.

En conclusión, considero que es una herramienta muy poderosa, nada trivial de usar, la cual requiere un estudio más profundo para poder aprovechar todo su potencial. Así que este resumen es el inicio a un camino en la práctica de todas las herramientas que describí.