**UNIVERSIDAD TECNOLÓGICA DEL CENTRO DE VERACRUZ**

TICSI 6° A

INGENIERÍA DE SOFTWARE



**MANUAL DE USO**

2015

Sistema de control colaborativo de revisión y desarrollo de software

Elaboró “Equipo 5”

Ana Karina Elotlán Hernández

Contreras Ortiz José Manuel

Celia Hernández Chávez

Estuardo Sánchez Gómez

Prof:

MMC. Honorato Aguilar Galicia

INDICE

[1.- ¿Qué es GitHub? 3](#_Toc419922762)

[2 Comenzar a usar GitHub 4](#_Toc419922763)

[2.1 Registro de usuario 4](#_Toc419922764)

[2.2 Instalación del cliente Git 4](#_Toc419922765)

[3 Repositorios 6](#_Toc419922766)

[3.1 Crear un repositorio local 6](#_Toc419922767)

[3.2 Crear un repositorio en la web 7](#_Toc419922768)

[3.3 Clonar un repositorio 8](#_Toc419922769)

[3.4 Actualizar un repositorio 9](#_Toc419922770)

[3.5 Eliminar un repositorio 10](#_Toc419922771)

[3.6 Añadir y eliminar un colaborador 10](#_Toc419922772)

[4 Estadísticas 12](#_Toc419922773)

[4.1 Cambios visibles desde el cliente. 12](#_Toc419922774)

[4.2 Historial de cambios 13](#_Toc419922775)

[Bibliografía web 14](#_Toc419922776)

# 1.- ¿Qué es GitHub?

GitHub aloja un repositorio de código y brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto.

Además de eso, se puede contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork.

Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en tu cuenta), para eliminar algún bug o modificar cosas de él, una vez realizadas las modificaciones se puede enviar un pull al dueño del proyecto, éste podrá analizar los cambios que se han realizado fácilmente, y si considera interesante la contribución, adjuntarlo con el repositorio original.

Herramientas

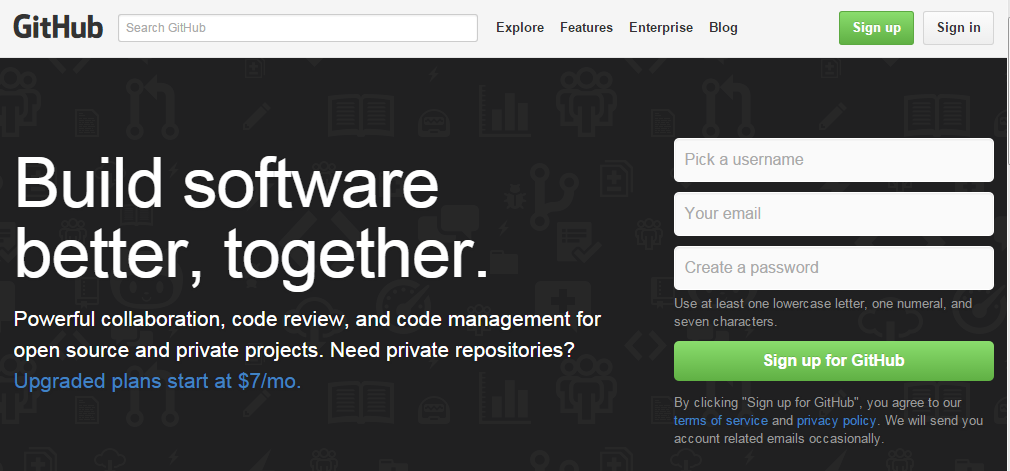
En la actualidad, GitHub es mucho más que un servicio de alojamiento de código. Además de éste, se ofrecen varias herramientas útiles para el trabajo en equipo. Entre ellas, cabe destacar:

* Una wiki para el mantenimiento de las distintas versiones de las páginas.
* Un sistema de seguimiento de problemas que permiten a los miembros de tu equipo detallar un problema con tu software o una sugerencia que deseen hacer.
* Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero y debatir sobre determinados cambios realizados en un Commit específico.
* Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

# 2 Comenzar a usar GitHub

Antes de comenzar cabe mencionar que el correspondiente manual está enfocado para implementarse bajo plataformas MS Windows, en caso de requerir su implementación bajo otro entorno recurrir a otras fuentes bibliográficas.

## 2.1 Registro de usuario



Para empezar a usar GitHub y sus herramientas es necesario registrarse en el portal oficial <https://github.com> proporcionando en primera instancia solamente un nombre de usuario, un E-mail y la correspondiente contraseña.

Para poder añadir colaboradores al proyecto es necesario que cada uno de los integrantes del equipo cuente con una cuenta, el administrador del proyecto será quien añada a cada uno de esos usuarios como colaboradores y estos podrán realizar los cambios y modificaciones que les sean permitidos.

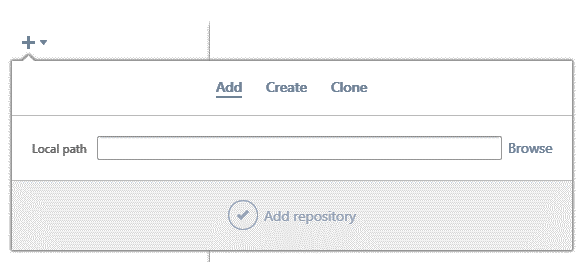
## 2.2 Instalación del cliente Git



Para descargar el cliente es necesario ir su página oficial <https://windows.github.com> y descargar el instalador v.2.13, su instalación no requiere configurar algún parámetro en especial, después de lo cual se tendrán accesos directos al cliente Git en modo gráfico y terminal, es necesario NetFramework 4.0 o superior.

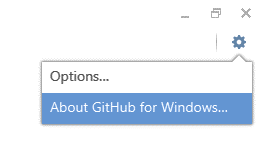


Una vez instalado se iniciará el proceso de configuración del cliente empezando por la autentificación por medio de una cuenta de usuario previamente establecida, el proceso es relativamente sencillo y se puede saltar cualquiera de los dos pasos siguientes como lo son la configuración del repositorio lo cual se verá más adelante.



Del lado superior izquierdo tendremos tres opciones para los repositorios.

* Add: Permite añadir un repositorio local a la cuenta On-line.
* Create: Permite crear un repositorio y almacenarlo tanto localmente.
* Clone: Clona o copia un repositorio web en forma local.

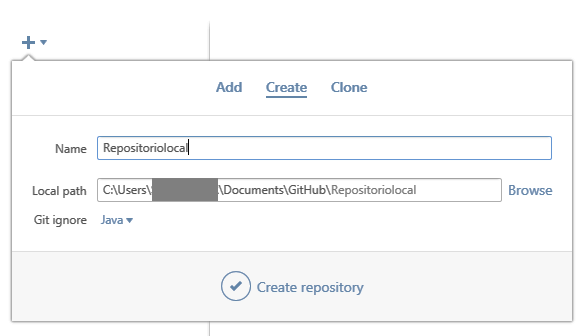


Del lado superior derecho tendremos las opciones de configuración del usuario y del programa cliente, entre ellas la de cerrar sesión o buscar o actualizar la lista de repositorios que no hayan sido sincronizados todavía.

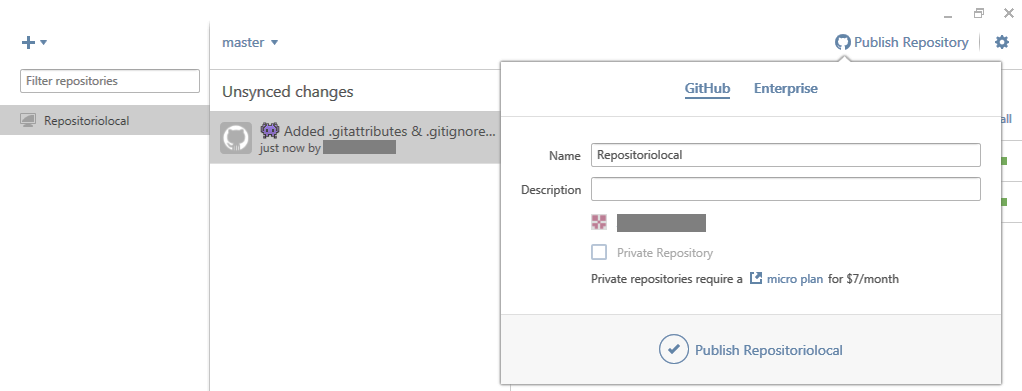
# 3 Repositorios

En el caso de los proyectos de software un repositorio es similar a una copia de seguridad en la cual se van generando copias y estadísticas controladas de los cambios realizados en el proyecto, esto permite más organización al momento de tener trabajando a muchas personas en un proyecto.

## 3.1 Crear un repositorio local

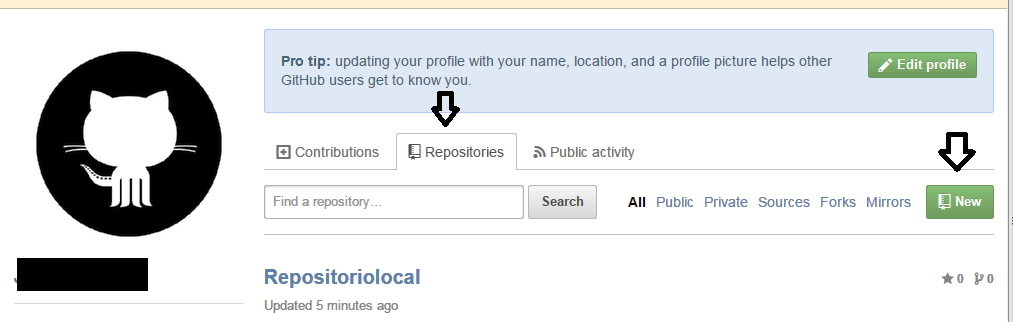


En las opciones del menú izquierdo de Git se escoge la opción Add, se establece el nombre del repositorio y la ruta donde este será almacenado de manera local, una vez hecho esto se creará un directorio con el nombre que se le haya asignado, dentro de este directorio se deberán copiar todos los archivos del proyecto con los cuales se desee trabajar.

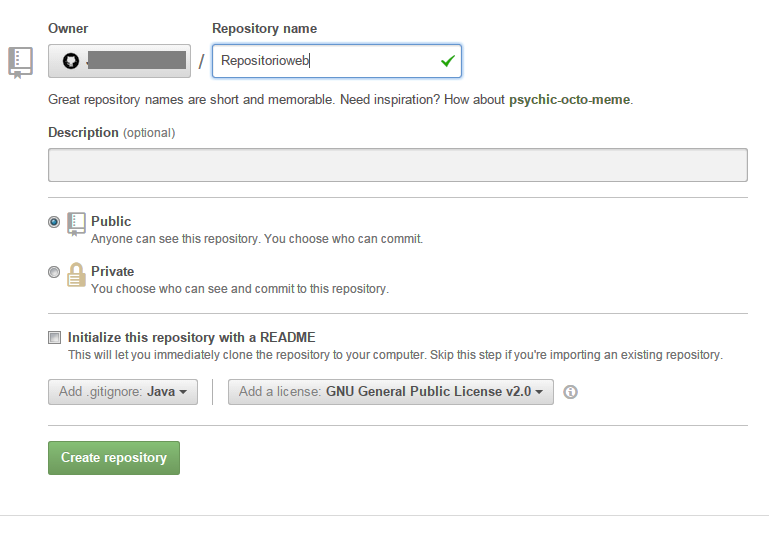


Una vez creado se puede optar por Publicar el repositorio para que este esté disponible de manera on-line y pueda ser modificado por otros usuarios colaboradores, se añade la opción de añadir una descripción, una vez esto el repositorio disponible.

## 3.2 Crear un repositorio en la web

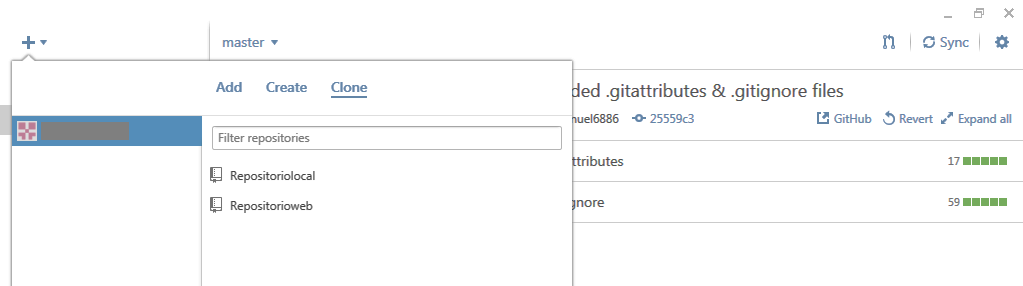


Para crear un repositorio web es necesario entrar a la página de GitHub y autentificarse con su cuenta de usuario, una vez dentro hay que dirigirse al perfil y elegir la opción Repositories y dentro de esa opción New para crear un apartado para el nuevo proyecto, cabe mencionar que esta es la manera más recomendable.

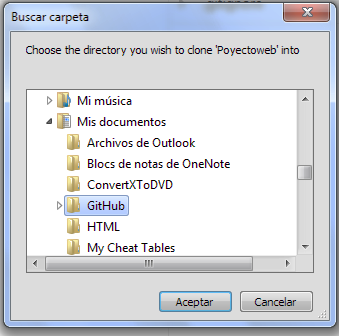


Algunas de las opciones que se permiten durante la creación del repositorio web son el tipo de lenguaje que se está utilizando para el proyecto, esto permite a GitHub omitir archivos que no sean necesarios tal es el caso de binarios compilados, también se puede establecer el tipo de licenciamiento bajo el cual se está desarrollando el producto de software, aunque en el caso de lenguaje este se puede establecer en el cliente al momento de publicarlo.

## 3.3 Clonar un repositorio

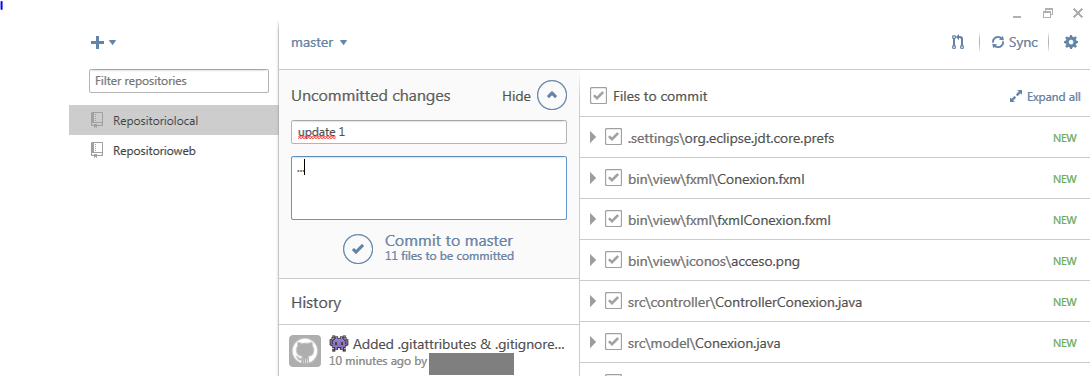


Clonar un repositorio no es otra cosa que generar una copia de un proyecto almacenado en GitHub de manera local o web para que de esta manera se pueda trabajar con él, para ello hay que ir al cliente Git y entre las opciones que se presentan seleccionar Clone lo cual cargará tanto los repositorios locales como los creados en el portal web, hay que seleccionar aquel del cual se quiera generar una copia.



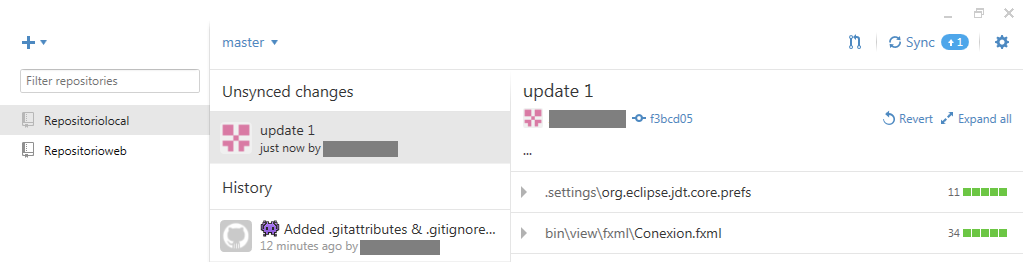
Acto seguido hay que seleccionar la ubicación donde se copiará el proyecto, por defecto es la misma ubicación que se haya establecido previamente, de preferencia es recomendable trabajar todos los proyectos en una misma ubicación ya que el programa cliente no permite configurar múltiples rutas.

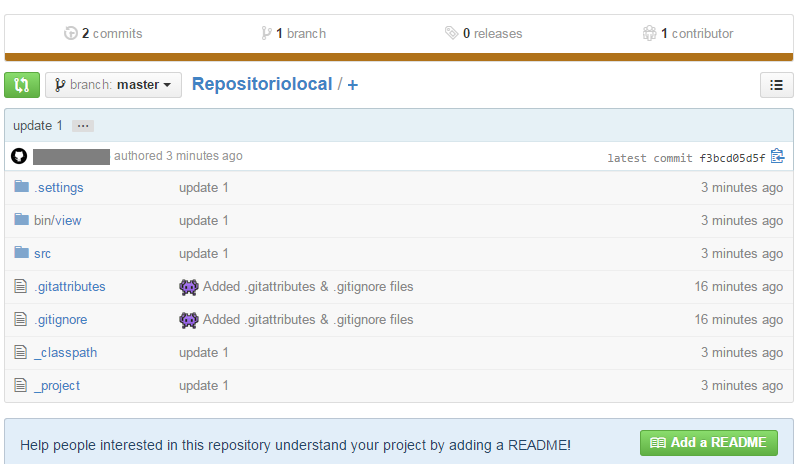
## 3.4 Actualizar un repositorio



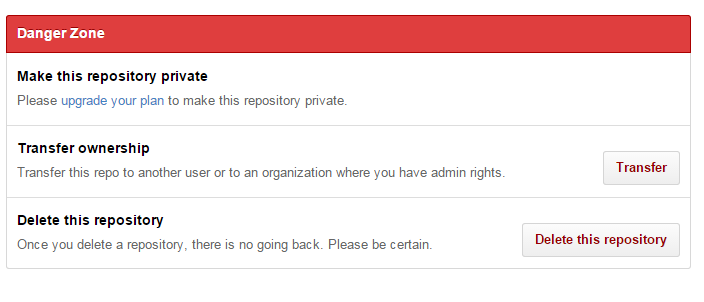
Cuando cada vez que se realice un cambio en los archivos el programa cliente detectará esas modificaciones como pueden ser la edición de un archivo, cambios en las líneas de código, borrado de elementos, hay ocasiones en que el cliente puede no detectar cambios como puede ser por ejemplo cuando se añade una librería la cual es un objeto precompilado y uno de los propósitos de GitHub es almacenar y gestionar únicamente código fuente.

Si se ha hecho una modificación esta es detectada en automático por el programa cliente, pero es necesario hacer un Commit para establecer una nueva copia y generar de este modo un listado de cambios.



Una vez hecho esto solo bastará con sincronizar (Sync) los nuevos cambios, esto también se tiene que hacer cuando se haya cargado el proyecto por primera vez y cada que se desee actualizar los cambios realizados.

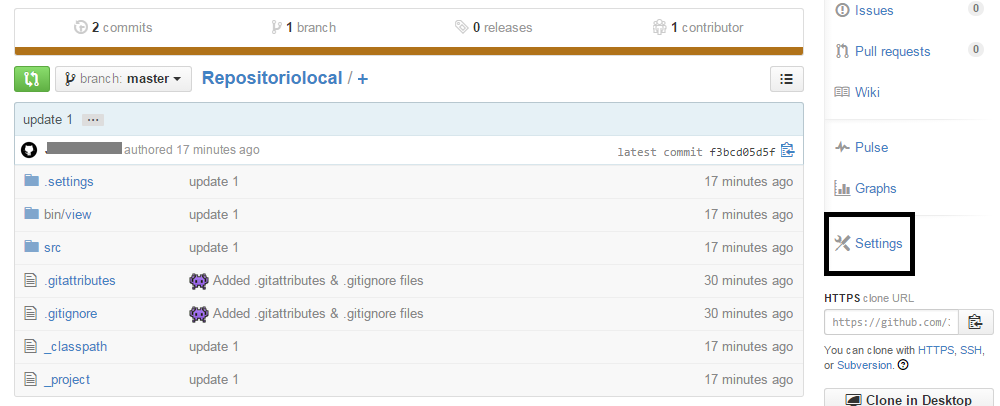
## 3.5 Eliminar un repositorio



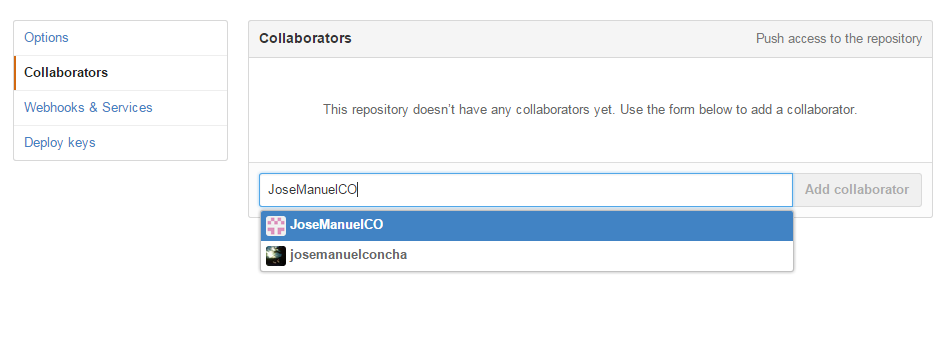
Por motivos de seguridad esta opción solo está disponible desde las opciones de configuración del propietario o administrador del proyecto, cuando se considere que el mismo ya no representa razón de ser se puede optar por eliminarlo completamente, también se puede transferir la propiedad a otro usuario o volverlo privado, esto último solo para una cuenta de pago.

## 3.6 Añadir y eliminar un colaborador

En el proyecto en cuestión seleccionar la opción Settings lo cual llevará a las opciones de configuración del proyecto.



Se podrá añadir solamente a aquellos usuarios registrados, una vez hecho esto los mismo recibirán una notificación vía correo electrónico y si tienen el cliente instalado verán tanto el usuario como el repositorio, para eliminar un usuario basta con ir a las mismas opciones y eliminar el o los usuarios que hayan sido invitados.

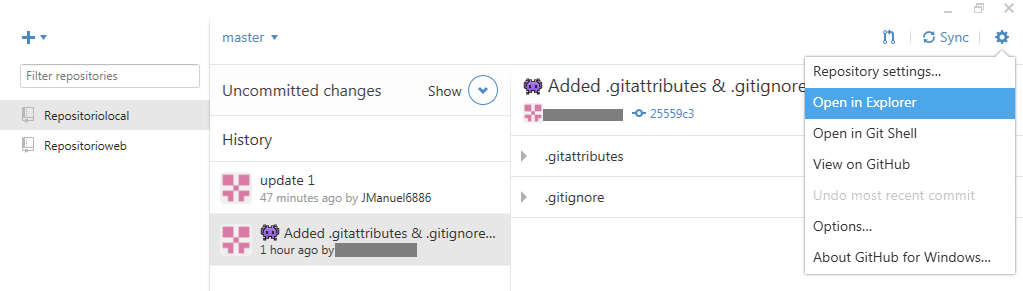


Los usuarios que coincidan irán apareciendo como recomendaciones, se debe estar completamente seguro del usuario al que se está invitando, tanto para evitar molestias o confusiones como para mantener la integridad del proyecto en cuestión.

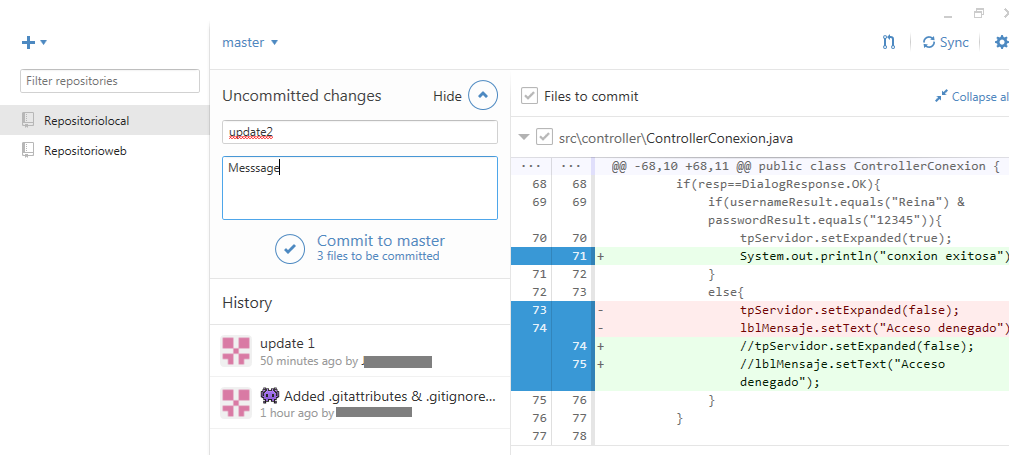


# 4 Estadísticas

## 4.1 Cambios visibles desde el cliente.

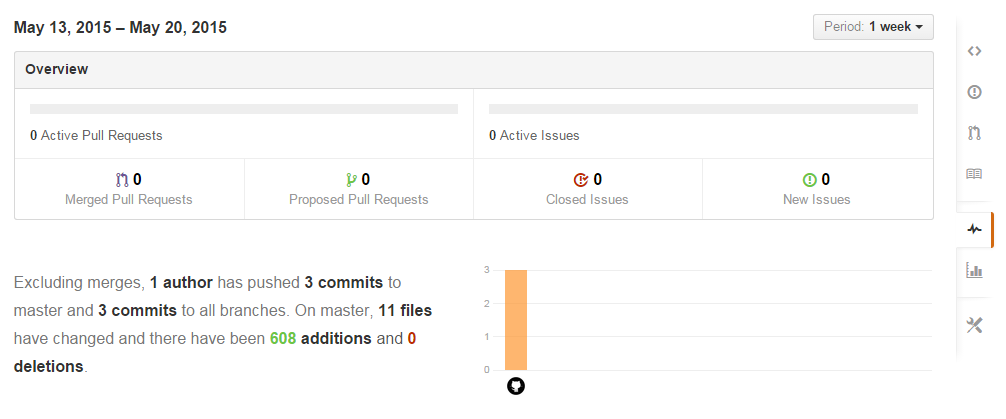


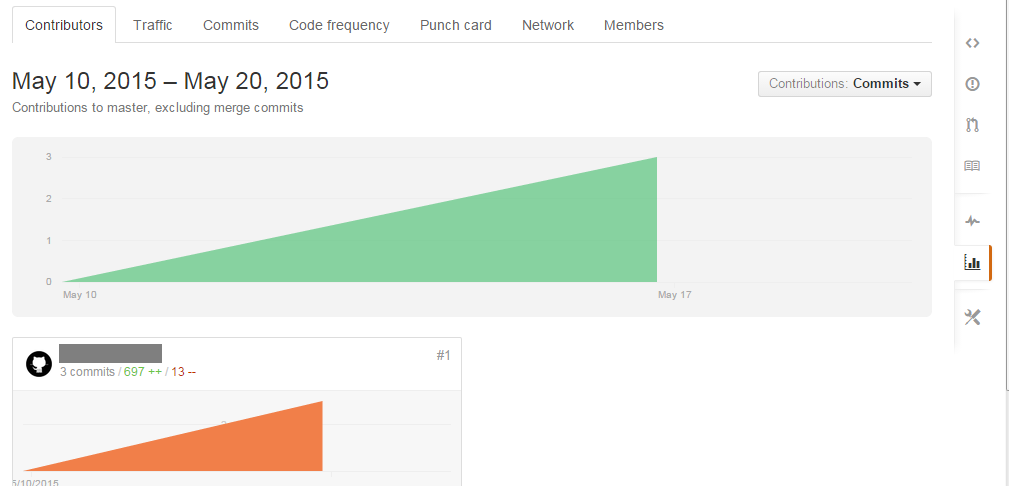
Tanto si los cambios son realizados de manera manual, por algún IDE o editor de código los mismos son detectados automáticamente, mientras el programa cliente esté en ejecución estará monitorizando el directorio donde se ubiquen los repositorios configurados.



Antes de realizar la sincronización y por tanto autorizando los cambios en la interfaz de Git se podrán ver todos los cambios realizados, no solamente de los ficheros y directorios, también los realizados al código fuente, dará la posibilidad de revertir los cambios a un estado anterior si estos no son autorizados, de los contrario se puede proceder a hacer la actualización de los mismos añadiendo un nombre y comentario.

## 4.2 Historial de cambios





En las opciones Pulse y Graph de cada repositorio se pueden visualizar todas las estadísticas relativas a los cambios realizados, al ser un apartado por demás extenso solo se hace mención de algunas de las opciones que se tienen presente en estas secciones.

* Commit realizados.
* Archivos o directorios modificados o eliminados.
* Código fuente modificado y autorizado.
* Nivel de actividad de cada uno de los colaboradores.
* Clones realizados y usuario que los realizó.
* Frecuencia de actualización el código fuente.
* Miembros activos.

# Bibliografía web

Documentación-Conociendo GitHub. (Luciano Castillo).

Recuperado de https://conociendogithub.readthedocs.org/en/latest/

GitHub help. (GitHub team).

Recuperado de https://help.github.com

Usar GitHub como repositorio de código. (Mario Chávez).

Recuperado de http://www.decisionesinteligentes.com/guides/github-como-repositorio