**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

**Versiones para “Manual de GitHub”**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 16/06/15 | 3.0 | Se añaden los detalles correspondientes al cliente Tortoise SVN para la gestión del repositorio. | Contreras Ortiz José Manuel |

INDICE

[1.- ¿Qué es GitHub? 4](#_Toc422253285)

[2 Comenzar a usar GitHub 5](#_Toc422253286)

[2.1 Registro de usuario 5](#_Toc422253287)

[2.2 Instalación del programa cliente Tortoise 6](#_Toc422253288)

[2.3 Elementos que conforman Tortoise 9](#_Toc422253289)

[3 Repositorios 11](#_Toc422253290)

[3.2 Crear un repositorio en la web 11](#_Toc422253291)

[3.3 Clonar un repositorio 12](#_Toc422253292)

[3.4 Actualizar un repositorio 14](#_Toc422253293)

[3.5 Eliminar un repositorio 15](#_Toc422253294)

[3.6 Añadir y eliminar un colaborador 16](#_Toc422253295)

[4 Estadísticas 17](#_Toc422253296)

[4.1 Cambios visibles desde el cliente. 17](#_Toc422253297)

[4.2 Historial de cambios 18](#_Toc422253298)

[Bibliografía web 20](#_Toc422253299)

# 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.

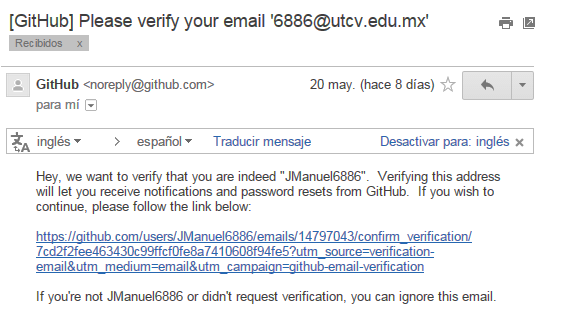
## 2.1 Registro de usuario

Para empezar a usar GitHub y sus herramientas es necesario registrarse en el portal oficial <https://github.com> proporcionando en primera instancia los siguientes datos.

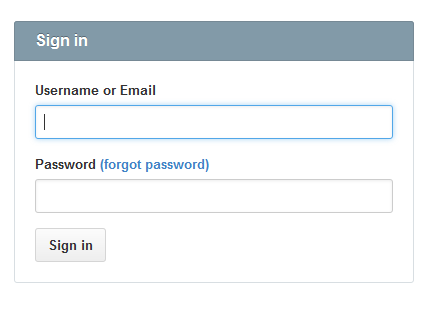
* Un nombre de usuario.
* Una dirección de correo electrónica válida.
* Una contraseña

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.

Una vez realizado el proceso de registro es importante verificar la bandeja de entrada de la dirección de correo electrónico ya que se enviará un vínculo para validar la cuenta recién creada.

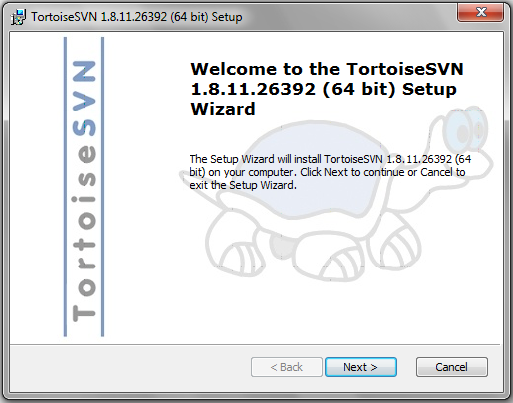


Una vez hecho esto es posible iniciar sesión desde el sitio web accediendo al apartado Sing in.

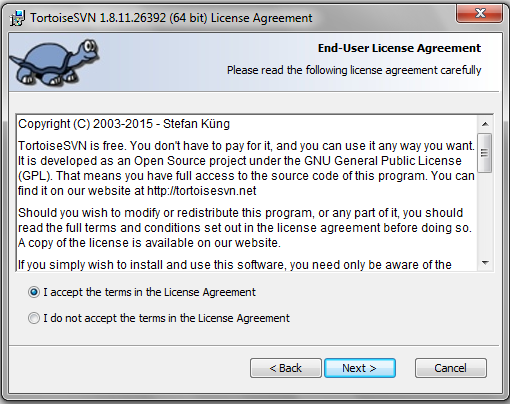


## 2.2 Instalación del programa cliente Tortoise

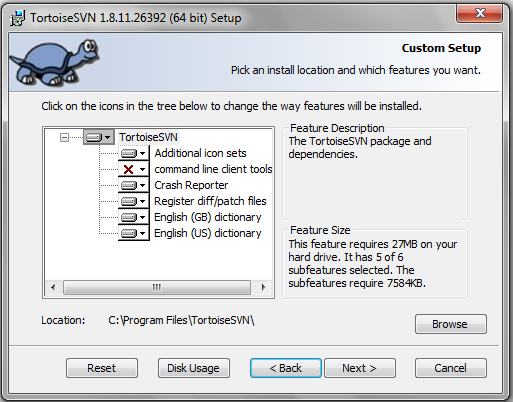
Para descargar el cliente es necesario ir su página oficial <http://tortoisesvn.net/> y descargar el instalador v.1.8.11 en la versión correspondiente a este manual, una vez descargado ejecutar el instalador, el sistema nos pedirá la siguiente confirmación ante lo cual le daremos en Ejecutar.

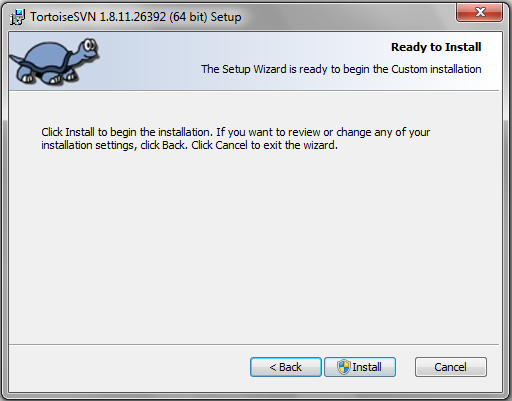


A continuación solamente daremos en el botón Next para continuar con la instalación.

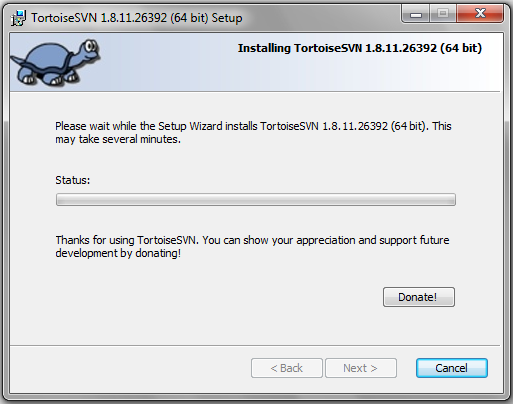


Aceptamos términos y condiciones de uso del programa.

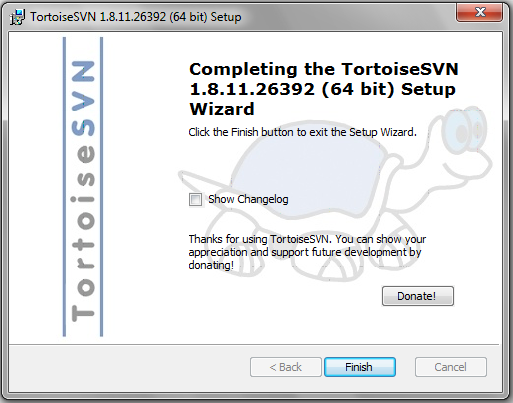


Si se desea personalizar la instalación se puede hacer este apartado y continuar con la inatalación.

Se presiona Install para confirmar la instalación.

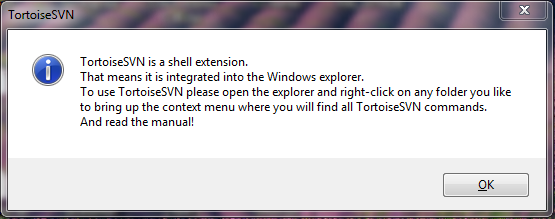


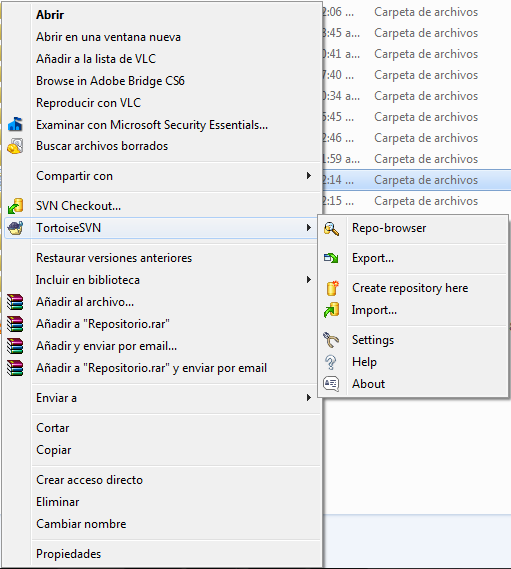
La instalación se llevará a cabo.



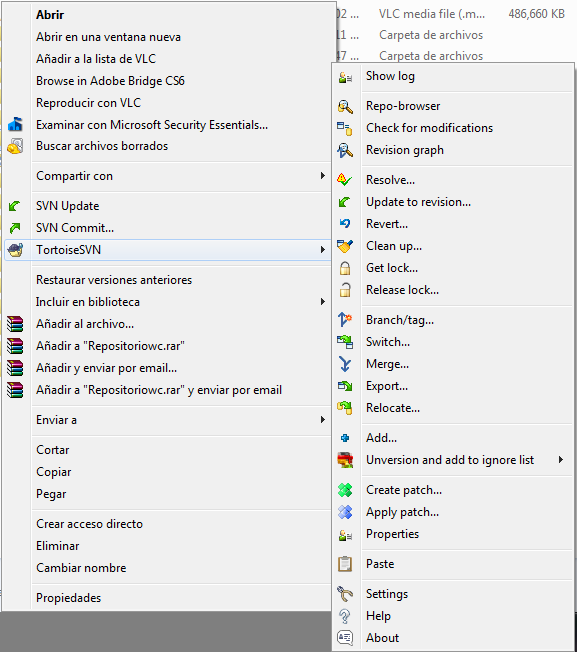
Una vez finalizada la instalación se presiona Finish para concluir y de esta manera el cliente quedará instalado y listo para usarse.

## 2.3 Elementos que conforman Tortoise



Tortoise SVN se instala como una extensión del Shell, es decir, no posee una interfaz concreta sino que consta de entradas en el menú contextual de Windows con las funciones correspondientes al elemento que se esté marcando y ventanas informativas, estos son algunos de los elementos que suelen aparecer.

Cuando aún no se ha creado el repositorio y también una vez hecho esto aparecerán las opciones para crearlo, exportarlo, importarlo (en caso de que exista) y opciones de búsqueda y configuración.



Tras la creación de un repositorio se tendrán nuevos elementos en el menú contextual mayormente relativos las acciones que se pueden realizar con los repositorios, cabe destacar los siguientes:

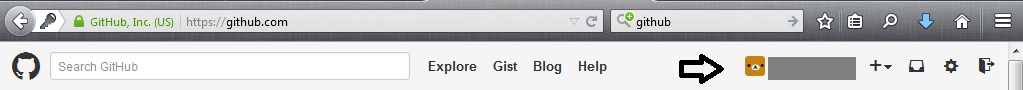
* SVN Update: Actualiza el contenido del repositorio.
* SVN Commit: Genera un nuevo commit para el repositorio.
* Show log: Muestra los cambios efectuados sobre un fichero o directorio del repositorio.
* Revision graph: Muestra los cambios de un fichero o directorio de una manera gráfica.

También se cuenta con opciones generales como son el borrado o el renombramiento, especificación de propiedades y configuraciones.

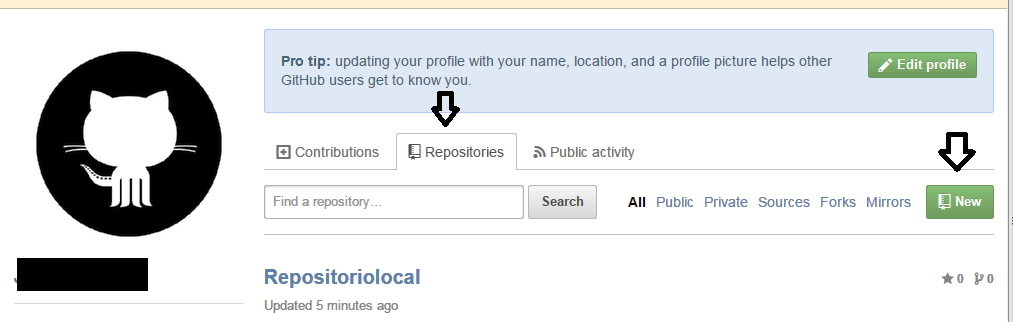
# 3 Repositorios

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.

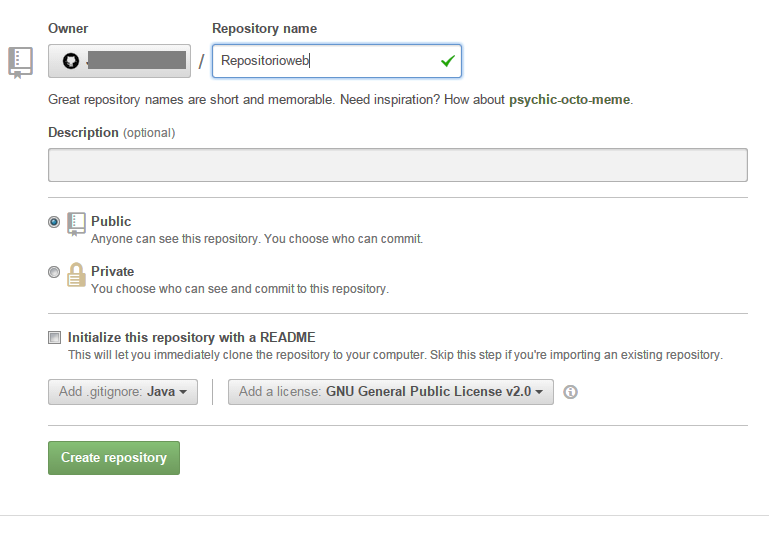
## 3.2 Crear un repositorio en la web



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

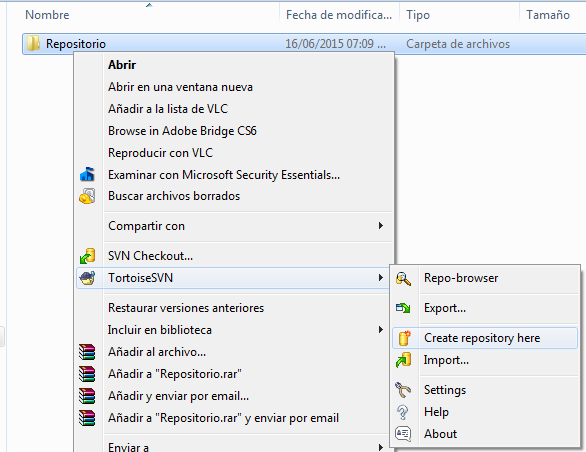


Después elegir la opción Repositories y dentro de esa opción New para crear un apartado para el nuevo proyecto, cabe mencionar que esta es la manera más recomendable.

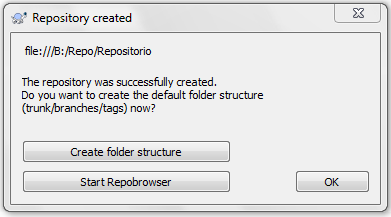


Algunas de las opciones que se permiten durante la creación del repositorio web son el tipo de lenguaje que se está utilizando para el proyecto, esto permite a GitHub omitir archivos que no sean necesarios tal es el caso de binarios compilados, también se puede establecer el tipo de licenciamiento bajo 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.

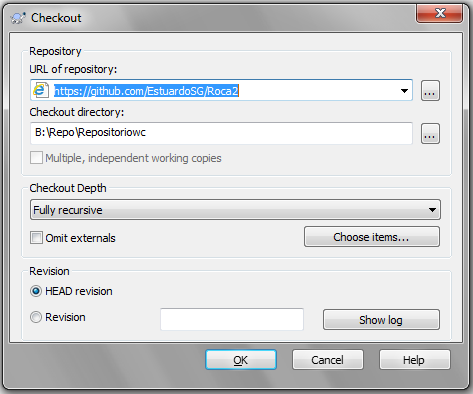
## 3.3 Clonar un repositorio



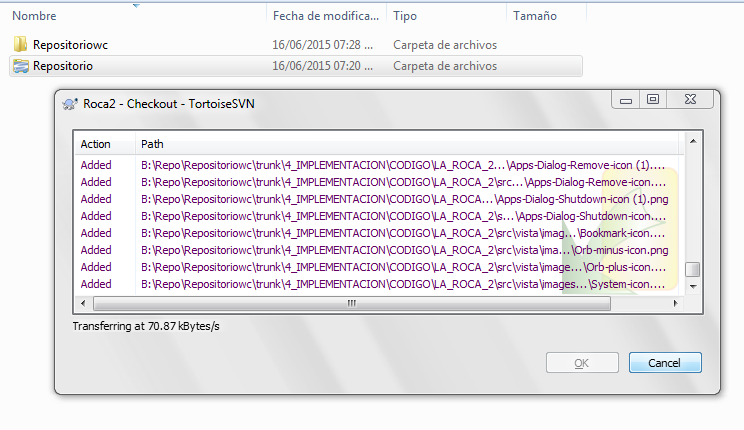
Clonar un repositorio no es otra cosa que generar una copia de un proyecto almacenado en GitHub de manera local o web para que de esta manera se pueda trabajar con él, para ello hay que crear una nueva carpeta vacía en la ruta donde se desee establecer el repositorio y con ayuda del menú contextual seleccionar TortoiseSVN>Create repository here.

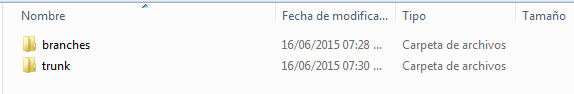


Acto seguido aparecerá una notificación a la que solo daremos OK para confirmar la creación de repositorio.

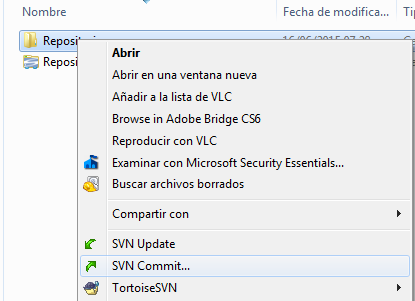


Una vez realizado lo anterior se procederá de manera similar a la anterior, pero esta vez seleccionando la opción SVN Checkout, de este modo se visualizará una ventana, se escribirá la url del repositorio en URL of repository, en caso de que el repositorio sea privado es posible que necesite ingresar el usuario y contraseña de su cuenta en caso de que sea propietario o colaborador.

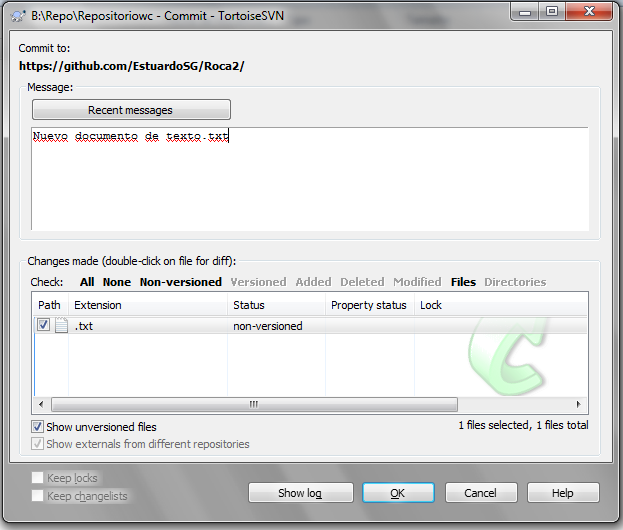


Si la url es válida se creará una nueva carpeta o directorio en donde se irá descargando todo el contenido solicitado del repositorio, la nueva copia se encontrará en el subdirectorio trunk.

## 3.4 Actualizar un repositorio



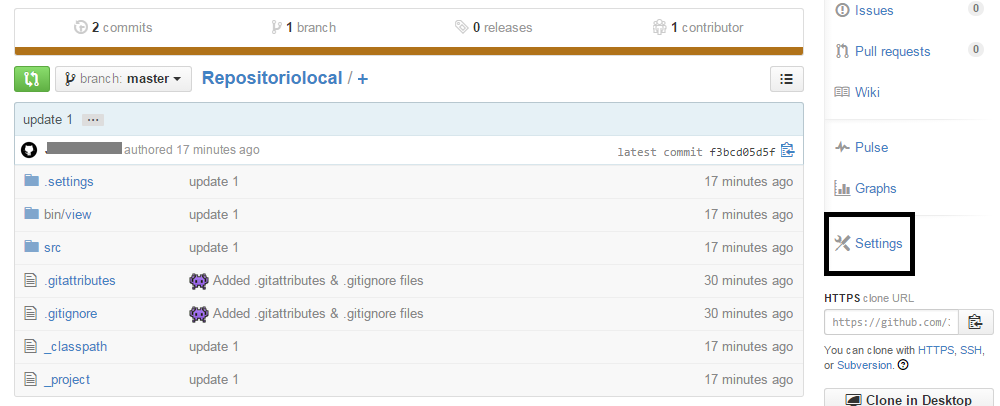
Cada vez que se desee realizar una actualización se repositorio en el directorio que contiene los ficheros clonados se selecciona con el menú contextual SVN Commit para generar un nuevo Commit.

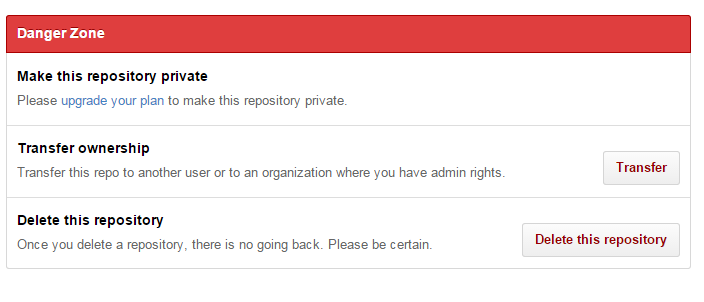


De haber cambios en los ficheros o directorios estos aparecerán en la parte inferior de una nueva ventana la en la cual se podrán marcar todos aquellos cambios que se deseen aplicar al repositorio así como los detalles anexos, posterior a esto será necesario ingresar el usuario y contraseña de una cuenta propietaria o colaboradora para poder aplicar los cambios.

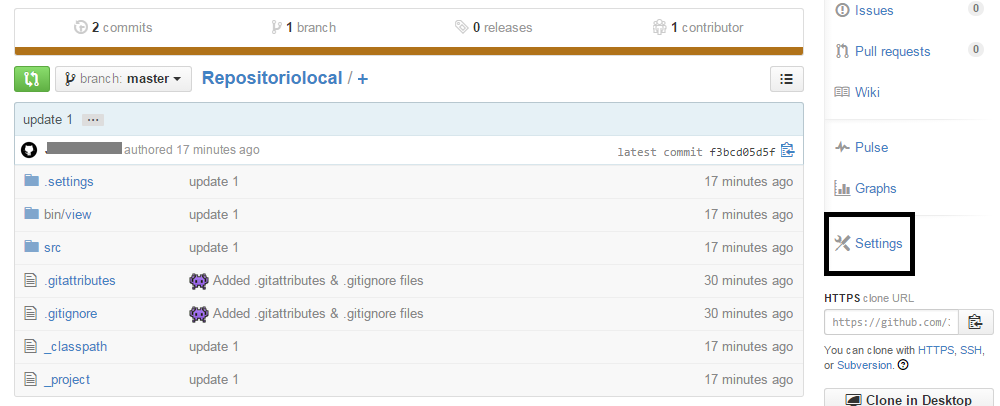
## 3.5 Eliminar un repositorio

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.

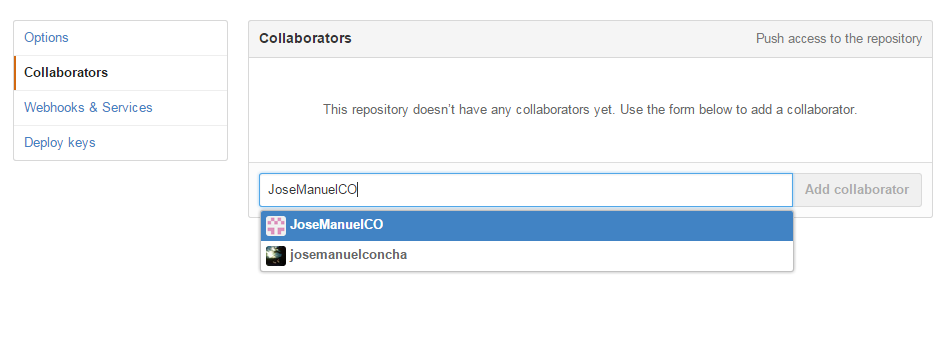




## 3.6 Añadir y eliminar un colaborador

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.

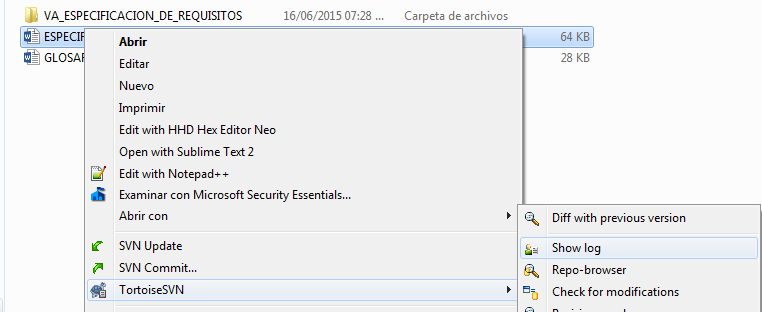


Los usuarios que coincidan irán apareciendo como recomendaciones, se debe estar completamente seguro del usuario al que se está invitando, tanto para evitar molestias o confusiones como para mantener la integridad del proyecto en cuestión.

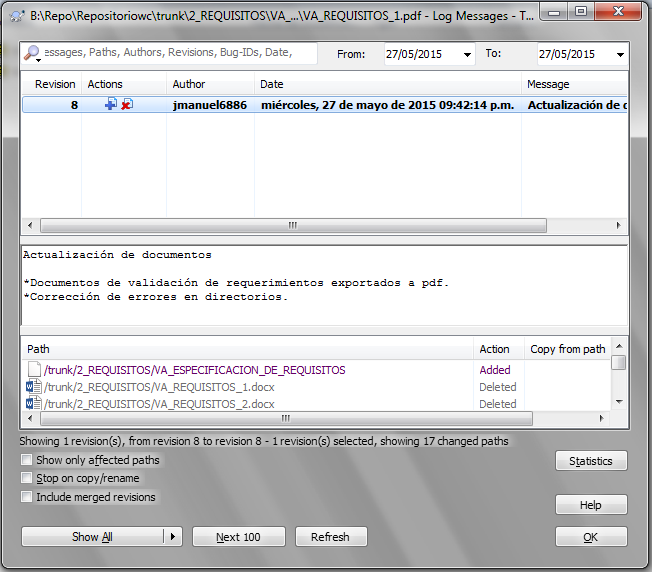
El repositorio será visible para el cliente y podrá clonarlo en el momento en que lo desee yendo al menú superior izquierdo del cliente Git y seleccionando Clone.

# 4 Estadísticas

## 4.1 Cambios visibles desde el cliente.



Cuando se desee ver los cambios realizados sobre un fichero o directorio se recurre al menú contextual en la opción Show log.

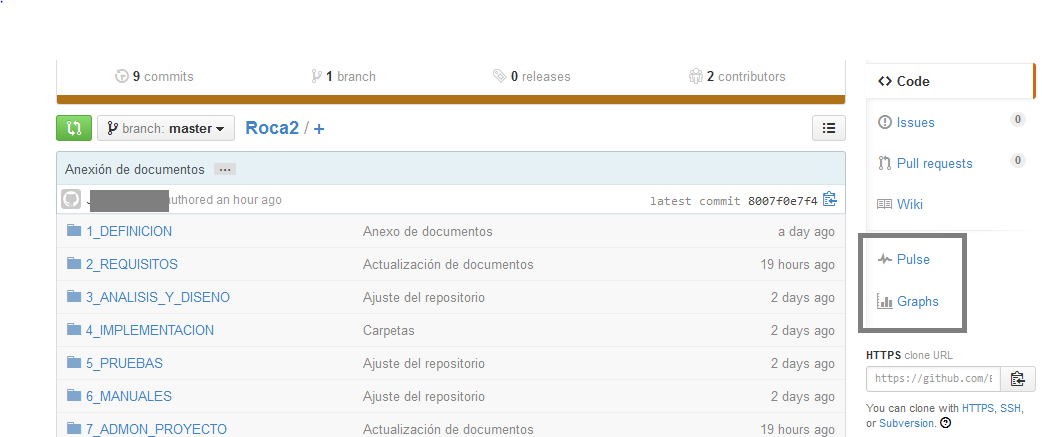


De este modo se podrán visualizar los commits generados, se visualiza el historial se cambios, los usuarios involucrados así como la clase de modificación realizada.

## 4.2 Historial de cambios

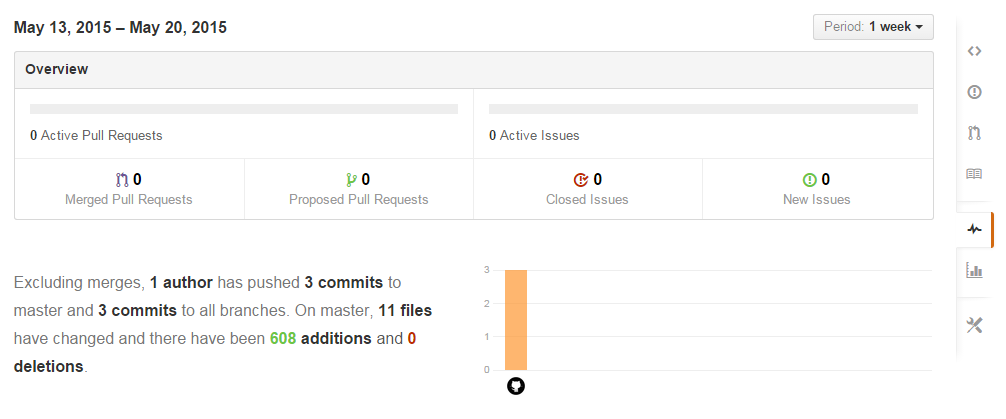
Para visualizar el historial basta con acceder a la url del repositorio, dado que para cuentas gratuitas el mismo es siempre público no es necesario siquiera contar con un registro previo para ello.

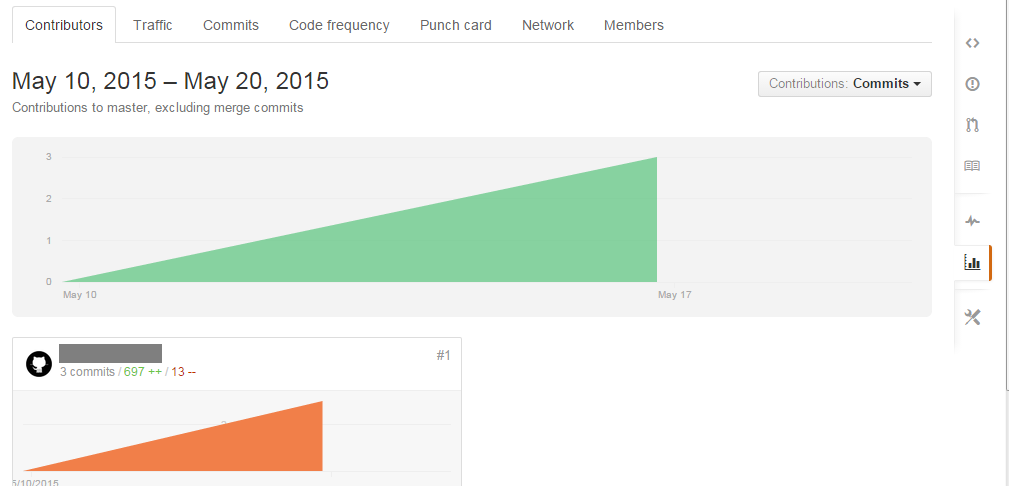
Una vez dentro se pueden visualizar los cambios accediendo a los apartados Pulse y Graph.



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. (L. 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. (M. Chávez).

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