**LOS ENTORNOS DE APRENDIZAJE A TRAVÉS DE CONTENDIDOS DIGITALES Y LA SELECCIÓN DE HERRAMIENTAS A UTILIZAR CON BASE EN LOS REQUISITOS DE LOS CASOS DE ESTUDIOS A DESARROLLAR.**

[[1]](#footnote-1)Un entorno de aprendizaje virtual es un espacio digital alojado en la web, el cual dispone de un conjunto de herramientas informáticas que posibilitan la interacción didáctica entre el docente y sus alumnos. De manera que un alumno puede llevar a cabo lectura de documentos, realizar ejercicios, formular preguntas al docente, trabajos en grupo…etc.

El entorno virtual de aprendizaje (EVA) más conocido en el mercado es Moodle el cual trata de un programa de código abierto (licencia gratuita). [[2]](#footnote-2)Existen diferentes grupos de entornos de formación según la finalidad, son los siguientes:

* Portales de distribución de contenidos
* Entornos de trabajo en grupo o de colaboración
* Sistemas de gestión de Contenidos (Content Management System, CMS).
* Sistemas de gestión del conocimiento (Learning Management System, LMS), también llamados Virtual Learning Enviroment (VLE) o Entornos Virtuales de aprendizaje (EVA).
* Sistemas de gestión de contenidos para el conocimiento o aprendizaje. (Learning Content Management System, LCMS)

[[3]](#footnote-3)Entre las herramientas para el e-learning se encuentran unas más sofisticadas para compartir información las cuales están orientadas a desarrollo de código o creación de textos en grupo y que pueden implementarse con fines de aprendizaje, además permiten llevar un registro de las actividades realizadas por los integrantes de un grupo para poder establecer la evolución de un proyecto con respecto al tiempo. Estas herramientas son repositorios alojados en la nube, donde los alumnos pueden trabajar en sus proyectos ya sean códigos o artículos. Git es un software libre tipo DVCS (Distributed Version Control Systems) diseñado para manejar proyectos grandes y pequeños con rapidez y eficiencia.

Con Git un usuario puede revisar diferentes versiones de un proyecto y llevar un seguimiento de las mejoras y correcciones que le realizo a ese proyecto en cada version, de esta forma si necesita regresar el proyecto a un estado anterior puede hacerlo sin comprometer la integridad de las otras versiones. En terminos de trabajo en equipo Git es un sistema distribuido lo que permite que varios desarrolladores puedan acceder a un proyecto que este en un servidor en la nube . En el caso en que 2 personas realicen un cambio de un archivo al mismo tiempo Git te ayuda a comparar y mostrar cual es el problema de manera que el desarrollador decida que parte se debe conservar.

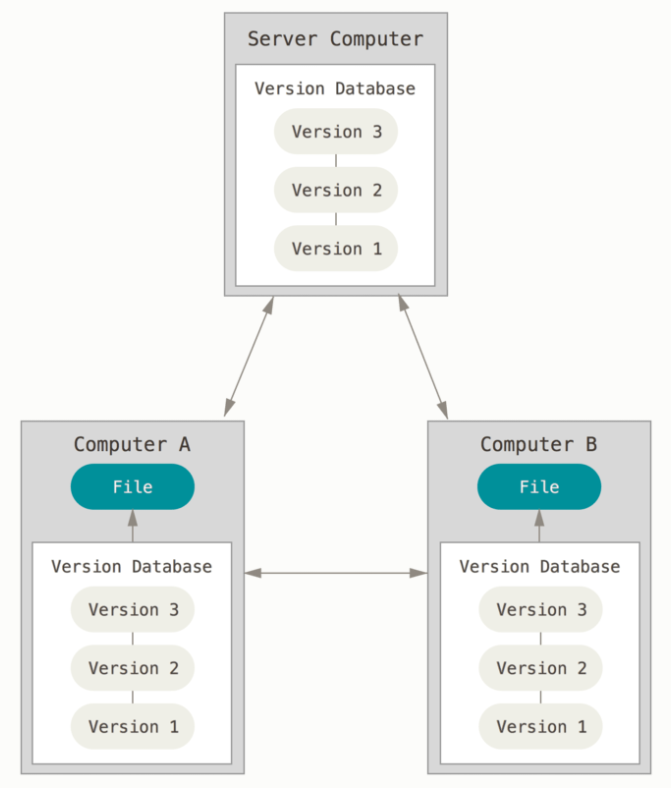


Fig1. Distributed Versión Control Systems

Github

Es uno de los más grandes host para repositorios de Git, y tiene una comunidad muy grande conformada por millones de personas compartiendo código entre otras cosas. La interfaz de usuario que proporciona Github es parecida a la de una red social donde el propietario de la cuenta posee su perfil de modo que puede obtener seguidores de su trabajo, puede seguir desarrolladores de código de la misma forma, también puede observar los repositorios que ha desarrollado y recibir “stars” o ayuda de otras personas que encontraron errores en su trabajo o simplemente quieren darle consejos para que mejore el proyecto. La plataforma permite que se pueda desarrollar una dinámica donde cualquier persona puede copiar tu repositorio si ve en él información útil para sus desarrollos, y aparte puede realizar mejoras y sugerírtelas, ya del autor del trabajo original depende de si considera que la mejora es válida para su proyecto; a este tipo de dinámica se le llama pull request y es una de las Herramientas de Github para realizar cambios en un repositorio cuando no se poseen los permisos para hacer un cambio directamente al repositorio.

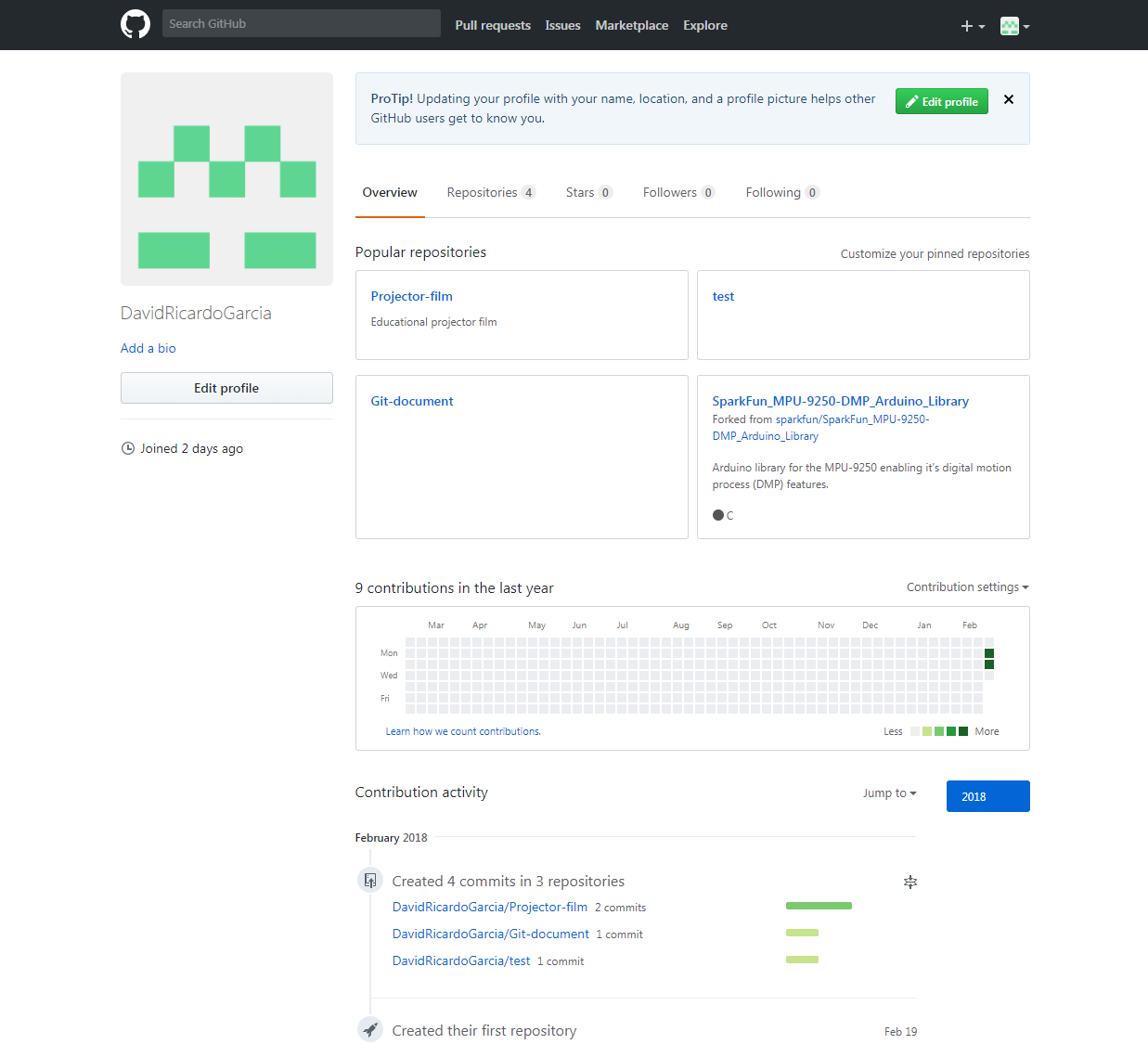


Fig2. Guardado local y subir al servidor.

Cada repositorio cuenta con un historial para los “commit” estos son los cambios que se hacen a los documentos del repositorio y que se van guardando. En cada commit se puede observar los cambios al código detalladamente, además se puede saber la fecha y hora en que se realizó cada commit y el usuario que agrego el cambio. Un branch es una ramificación del proyecto , debe haber un branch principal por default el cual es master y de este se desprenden otros branch, usualmente los branch que se desprenden del master son versiones del proyecto original donde se quiere realizar pruebas de cambios al código sin alterar el código original, otra característica es que un branch se pueden combinar con el branch master; esto puede ocurrir en el caso en que una prueba realizada fue positiva y se quiere seguir trabajando en el proyecto desde este punto. Otra herramienta que proporciona Github es “releases” estas son las versiones del documento y se pueden ir guardando cada vez que el desarrollador considere que se alcanzó un punto importante del proyecto y se debe guardar una versión de este por si en un futuro necesita regresar a ese punto en específico. Por ultimo para cada repositorio se pueden dar permisos de escritura, esto quiere decir que el propietario del repositorio puede agregar colaboradores que modifiquen los archivos del repositorio, si se agrega un colaborador este realiza cambios sin necesidad de un pull request. Cada repositorio debe contar con un archivo README para describir de que trata el proyecto del repositorio, esto permite que los que accedan al repositorio sepan de qué se trata de una forma resumida además de que sirve para que los autores dejen datos de contacto en caso de que estén publicitando su trabajo, también se pueden agregar topics o palabras claves para que Gtihub pueda categorizar los repositorios y así sean más fáciles de encontrar en las búsquedas.

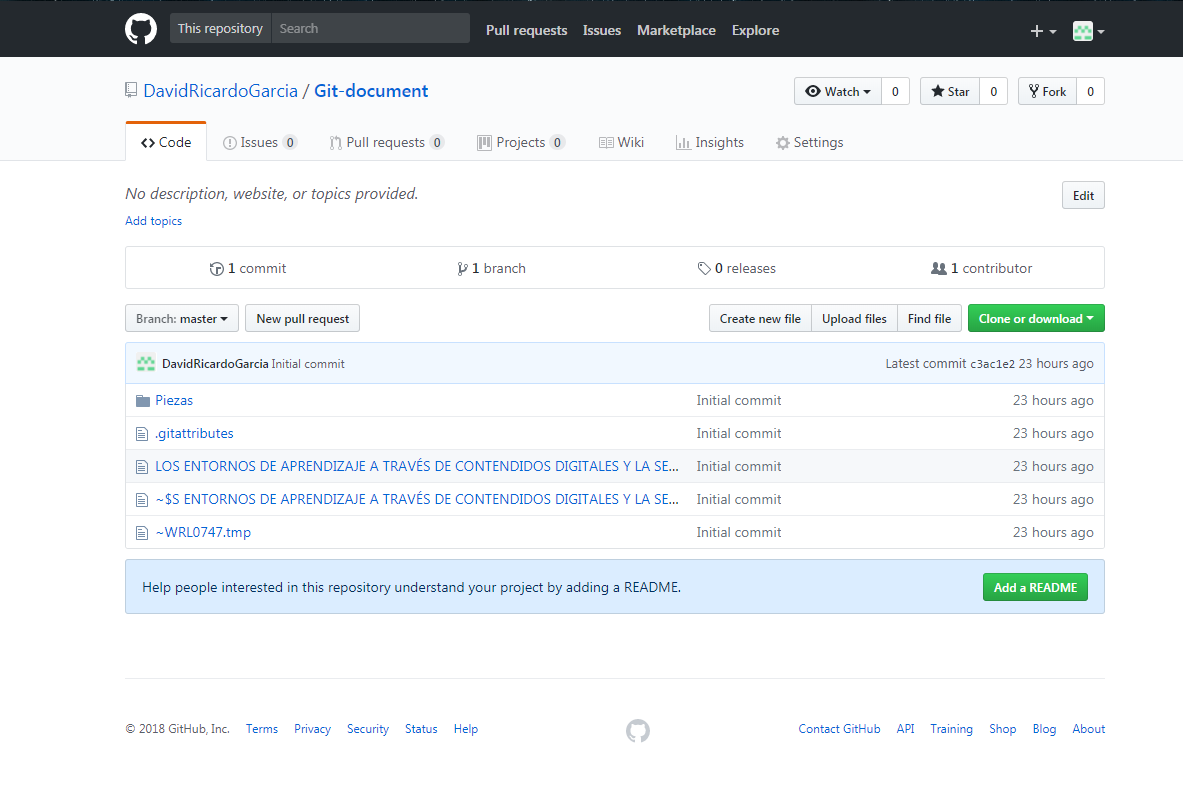


Fig2. Guardado local y subir al servidor.

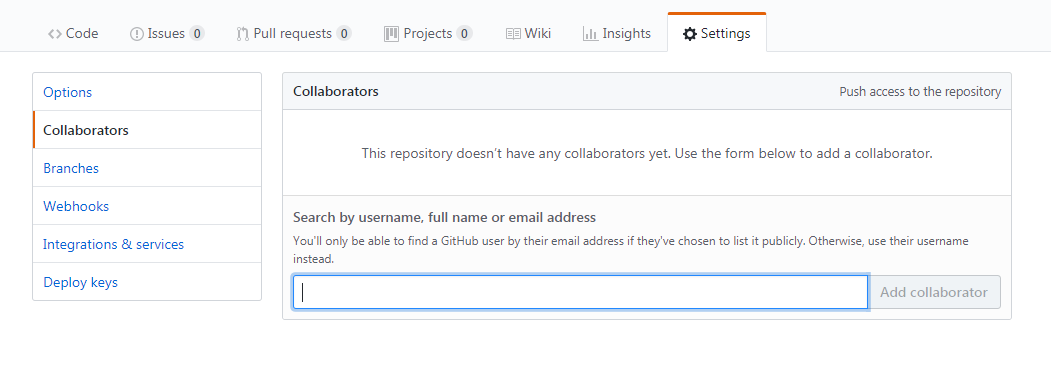


Fig2. Guardado local y subir al servidor.

Github Desktop

Es un software que permite agregar repositorios desde un computador local y estar subiendo los archivos a los repositorios en Github server, tambien permite realizar clones de los repositorios de

Github server a tu computador de manera que simplifica el trabajo con Github y ademas permite manejar muchas de las herramientas para el manejo de los archivos que la pagina de Github tiene por defecto como lo son el seguimiento de cada cambio que se realice a los archivos y la creacion de branches.

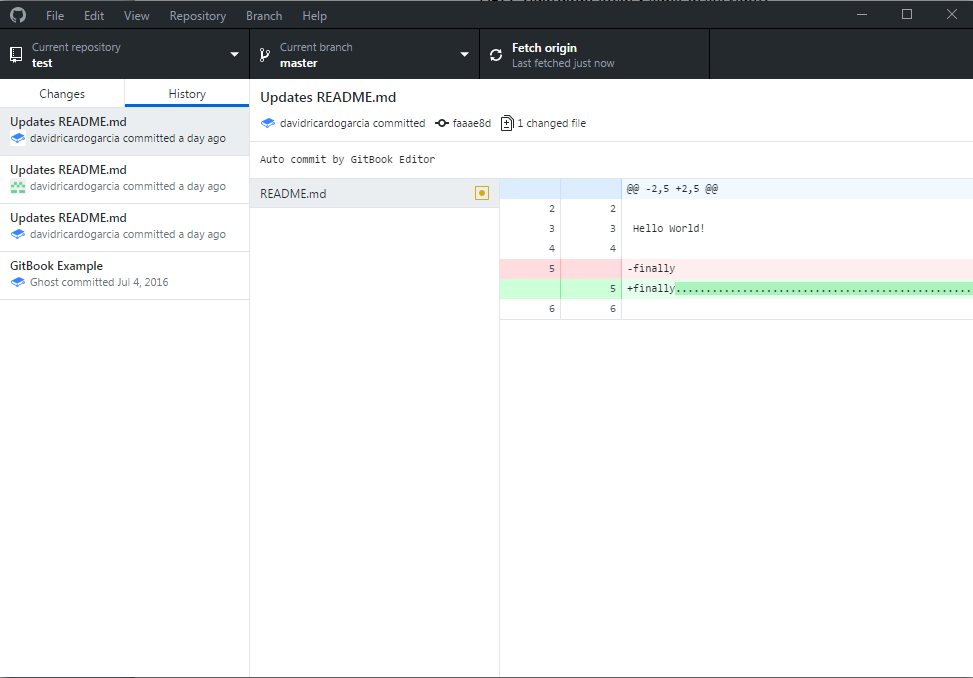


Fig2. Guardado local y subir al servidor.

La relacion entre Github y Gitbook

[[4]](#footnote-4)Son sistemas basados en Git , lo que los hace ser DVCS (Distributed Version Control Systems), y pueden agregar colaboradores a sus respectivos repositorios. Se diferencian en su enfoque, Gitbook se especializa para la compilación y publicación de libros o documentos escritos por lo que su función para el control de versiones es limitada, por otro lado Github es una plataforma profesional para el manejo de código con mejoras en su función para el control de versiones tales como pull ,request , issues and other functions.

Como crear un Book en Gitbook y relacionarlo con un repositorio en Github

1. Primero se debe crear una cuenta en  [http://github.com/](https://github.com/), e iniciar sesión.
2. Si ya tienes una cuenta en Github puedes acceder a [https://www.gitbook.com/](http://www.gitbook.com/) con esta misma cuenta. Al momento de iniciar sesión se debe dar la opción Sign in with Github.
3. Despues de iniciar en Gitbook , se crea un Book con cualquier nombre en este caso será llamado “test\_repository” y se da click en “créate book”.
4. Lo siguiente es conectarlo con Github de manera que se pueda desarrollar el Book de forma colaborativa. Entonces en la página de Gitbook se da click en “settings”, luego se copia la dirección del Book, esta dirección es usada para pasar la URL del Book a un repositorio en Github.

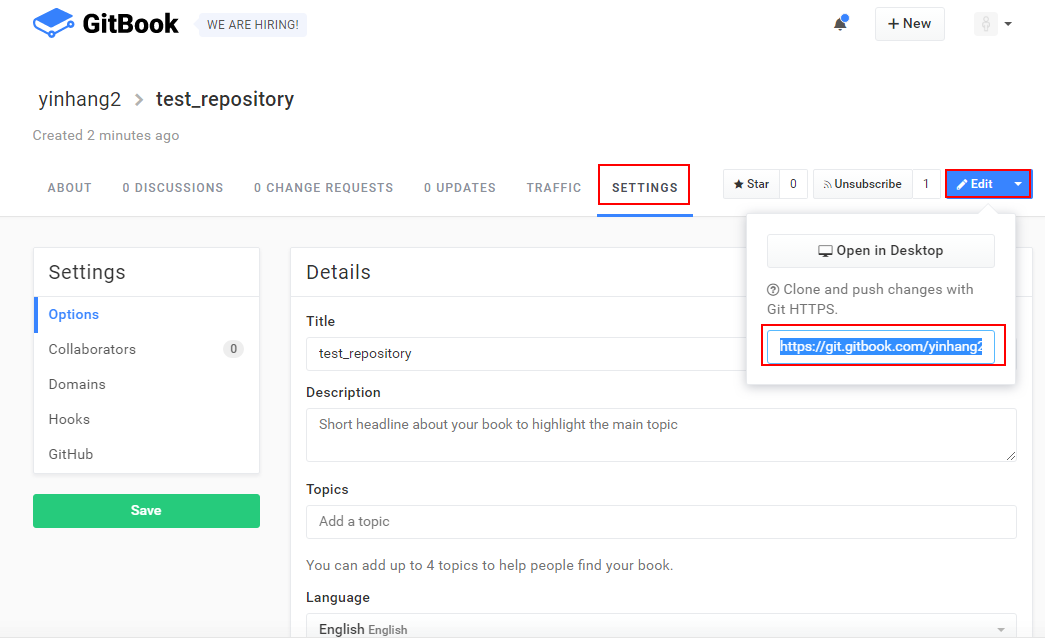
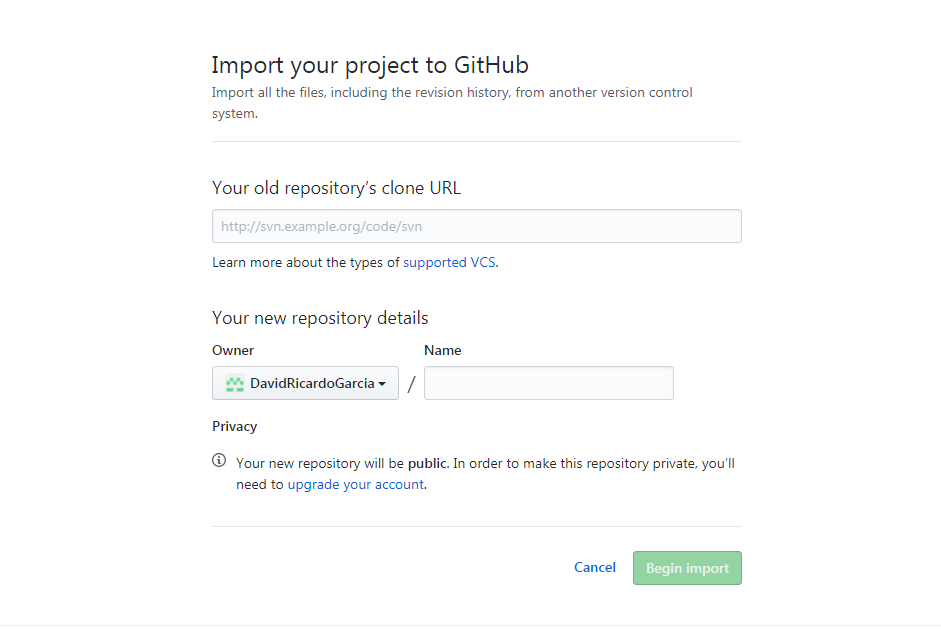


Fig1. Distributed Versión Control Systems.

1. En el link  <https://github.com/new/import>, se pone la dirección en la caja de URL y se crea un repositorio con cualquier nombre, en este caso será “test\_repository”. De esta forma Github puede sincronizarse con Gitbook atraves de la dirección.



Test\_repository

Fig1. Distributed Versión Control Systems.

1. Click en “Begin import”, e ingresa tu cuenta de Github y tu contraseña para dar permisos.
2. Ahora se debe regresar a Gitbook para darle la dirección del repositorio de Github. De esta manera queda establecida la comunicación bidireccional de las plataformas

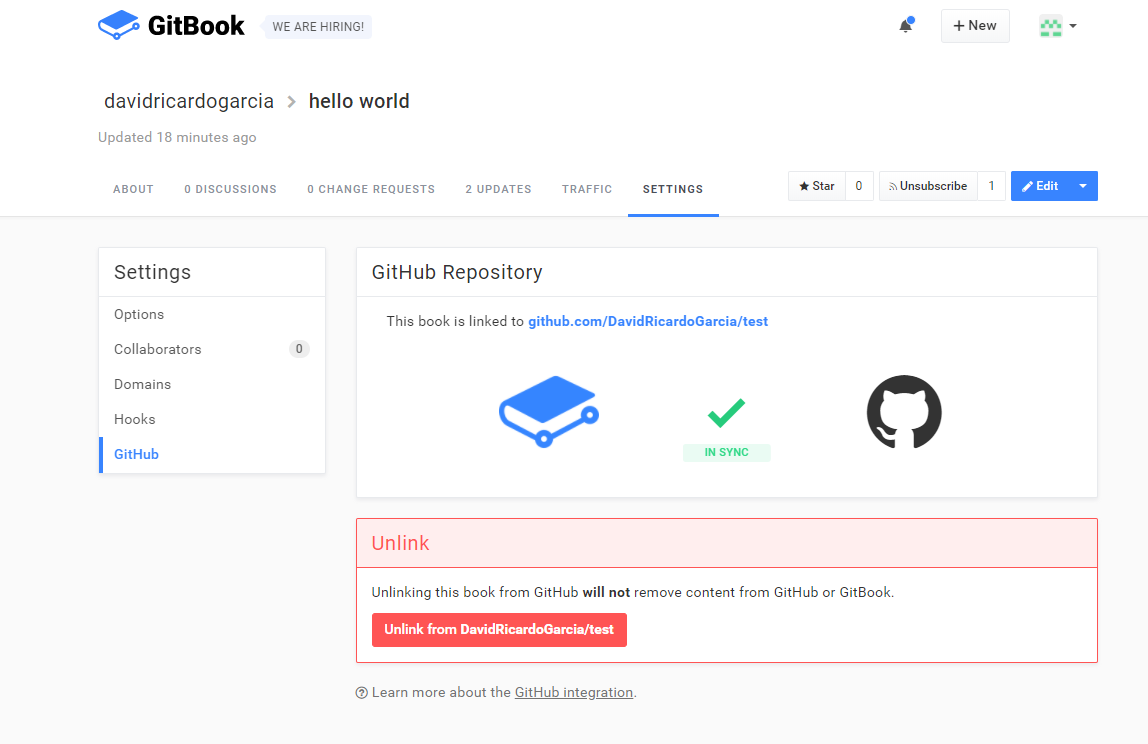


Fig1. Distributed Versión Control Systems.

Se establece la comunicación entre Gitbook y Github debido a que se va a crear un libro colaborativo, de manera que varios autores van a estar realizando modificaciones al documento. Github se encarga de saber quién, cuando, y donde se realizó un cambio al documento, además Github da los permisos de quien puede generar cambios en el documento.

Para empezar a trabajar el documento se puede descargar el Gitbook client, este programa posee una interfaz de fácil manejo que permite modificar los libros de tu cuenta de Gitbook de manera rápida y luego subir los cambios al servidor.

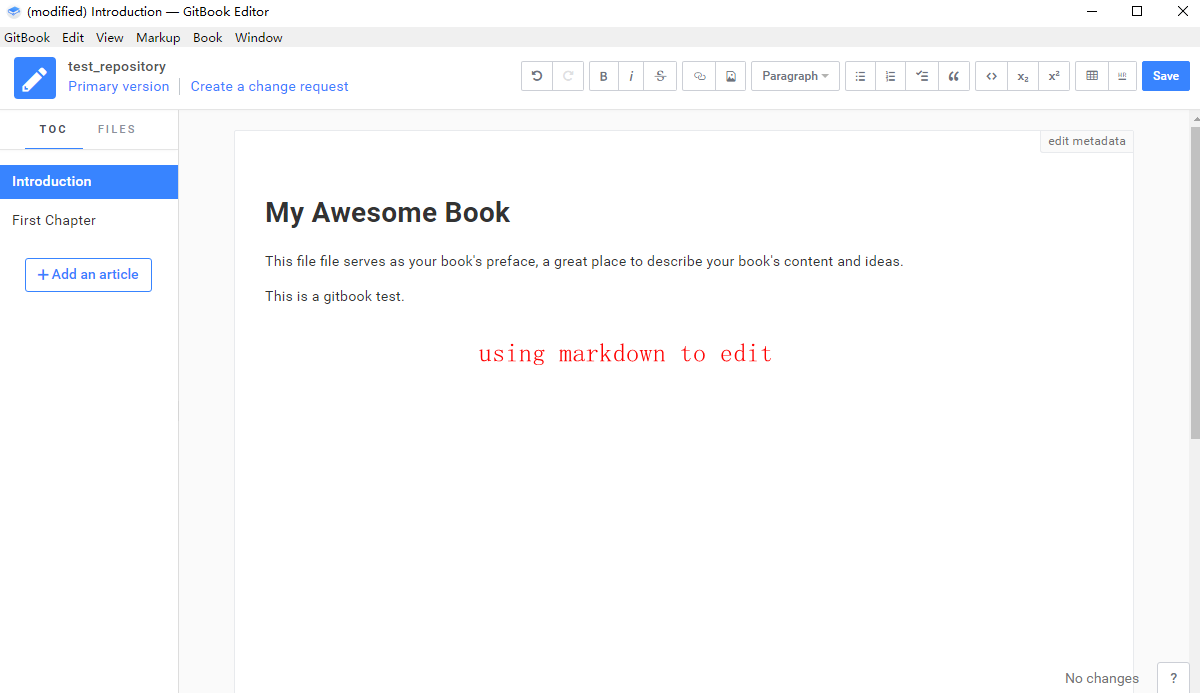


Fig2. Entorno de Gitbook

Cuando se termine un cambio se quieran guardar los cambios en el servidor en la nube se da click en el boton upload to cloud. Con esto los cambios que se hagan en el documento se realizaran en el sitio Web de Github y Gitbook.

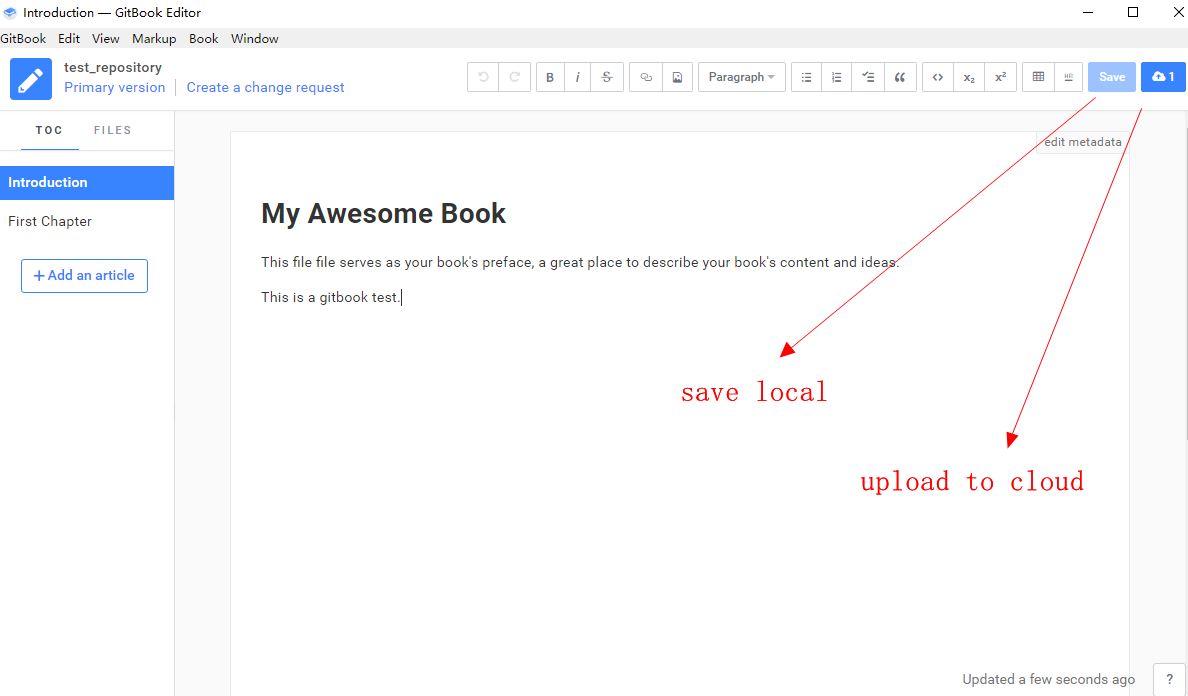


Fig2. Guardado local y subir al servidor.

Para agregar colaboradores a tu documento en tu cuenta de Gitbook vas a settings y agregas el correo o nombre de la cuenta de Github del colaborador.

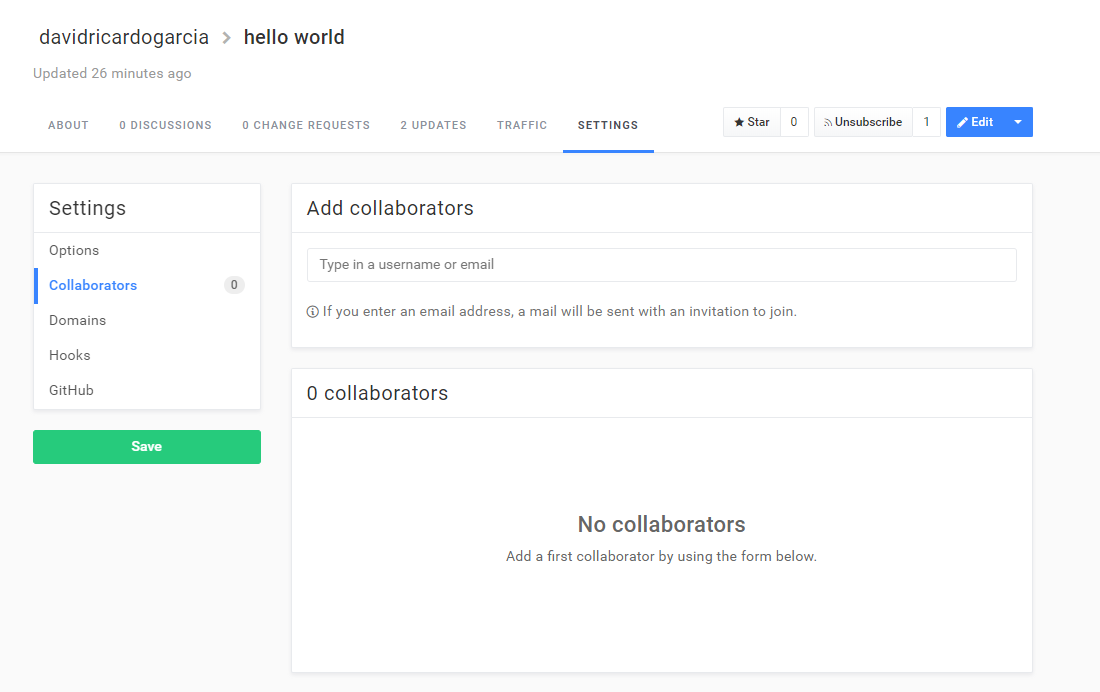


Fig2. Guardado local y subir al servidor.

Obtener un repositorio y solicitud de extracción.

Debido a que Gitub es una página de código abierto al crear un documento en ella muchas personas que visitan el sitio pueden tener acceso a la información. En caso de que alguien encuentre problemas en el código o información que se halla compartido en un repositorio o tal vez quiera ser parte del proyecto y agregar información útil este puede hacerlo mediante un pull request. Para hacer esto se deben seguir los siguientes pasos:

1. Primero se debe obtener el repositorio, esto se hace clonando el repositorio sobre el que se quiere trabajar en tu propia cuenta. Para ello se debe descargar el cliente de Github [https://desktop.github.com](https://desktop.github.com/) e instalarlo.
2. Luego se debe iniciar sesión con tu cuenta de Github y proceder a clonar el repositorio, haciendo click en el símbolo”+” en la esquina izquierda, se busca el repositorio deseado y se da click en clone.

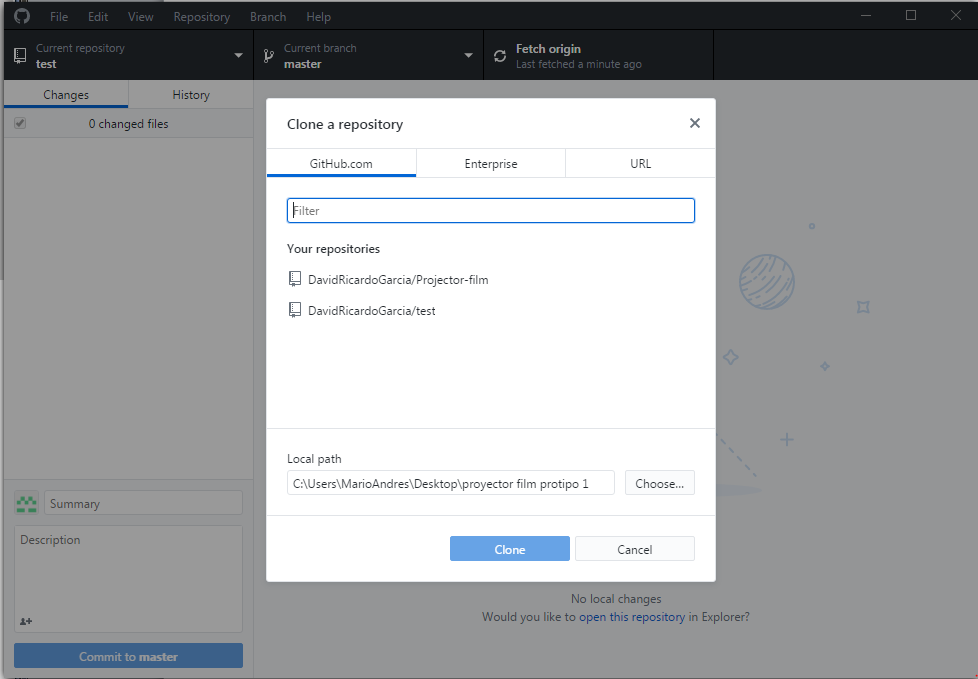


Fig2. Guardado local y subir al servidor.

1. Lo siguiente es elegir la localización donde el repositorio se va a guardar en tu pc.
2. Cuando se termine la operación de clonado de repositorio, se debe abrir Gitbook cliente y dar click en Gitbookeditor->“import”, luego se selecciona el repositorio que se acabó de clonar en Github client y se abre.
3. Desde este momento el repositorio que se ha clonado aparecerá en la página de inicio de Gitbook. Despues de terminar el trabajo sobre el repositorio recuerda guardar “save” y dar click en “upload to cloud”.
4. Despues en el Github da click en “Sync” y en ese momento tu trabajo se sincronizara en tu repositorio en el servidor de Github.
5. Finalmente tú necesitas hacer un “pull request” desde tu repositorio de github de manera que el cambio que se hizo a la versión clonada pueda ser ejecutado en el repositorio maestro, esto dependiendo de si el propietario considera que es adecuado.

1. http://www.aula1.com/entorno-virtual-aprendizaje-eva/ [↑](#footnote-ref-1)
2. https://www.uv.es/bellochc/pedagogia/EVA3.pdf [↑](#footnote-ref-2)
3. https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control [↑](#footnote-ref-3)
4. https://cooc-china.gitbooks.io/cooc-howto-book/content/english-version/create-the-book-and-make-connection-with-github-repository.html [↑](#footnote-ref-4)