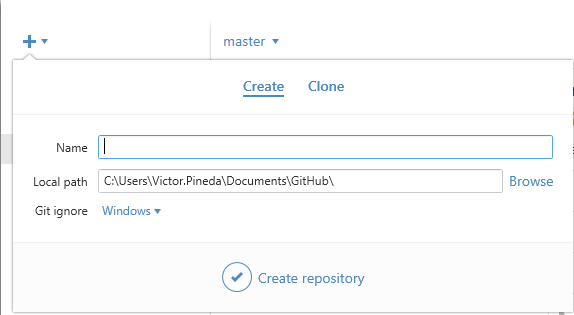
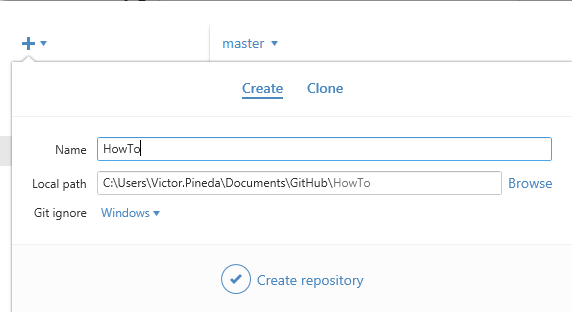
Agregar repositorio desde el cliente, clic en el signo +, parte superior

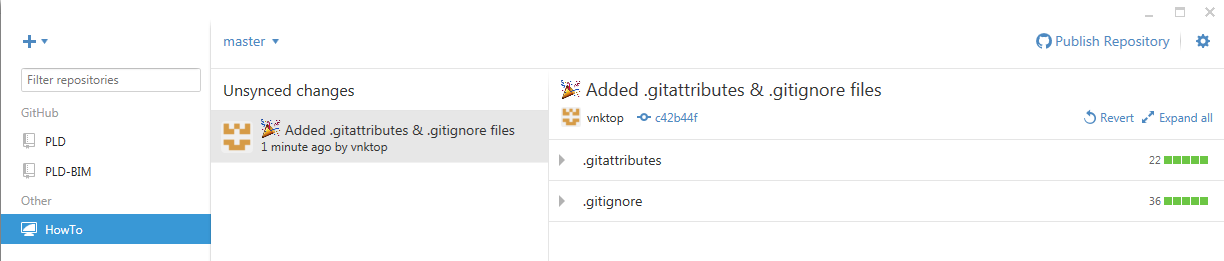


Asignar nombre de repositorio, observar cómo se genera la ruta local. Dar clic en Create repository

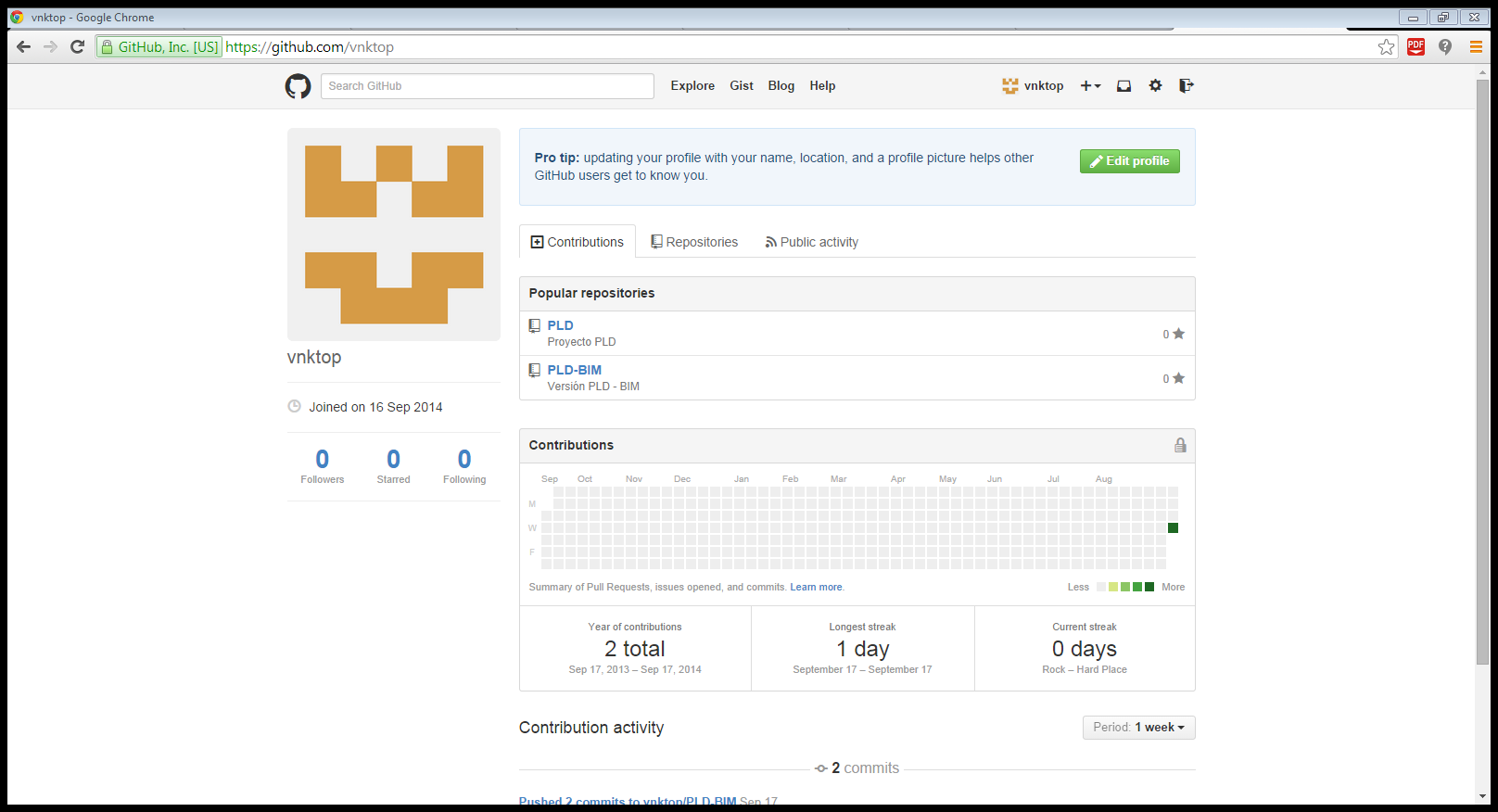


Repositorio sin commit. Existe el directorio local, pero no en la web

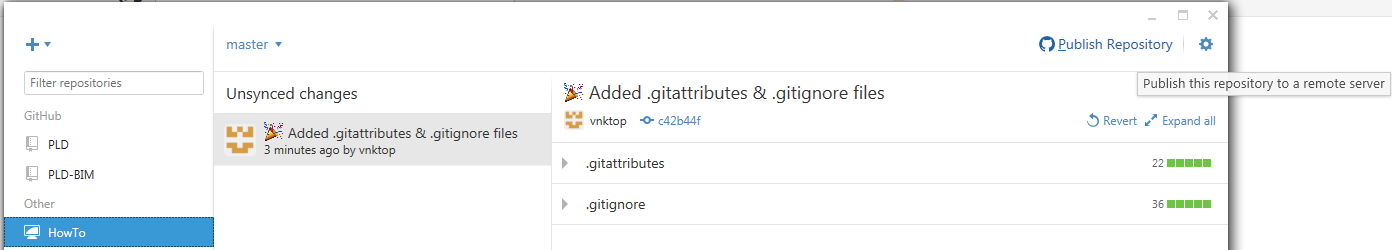
Local



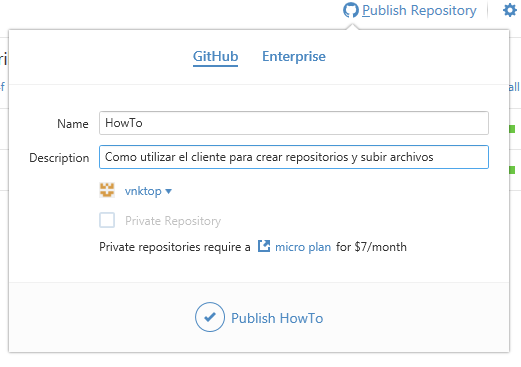
Web



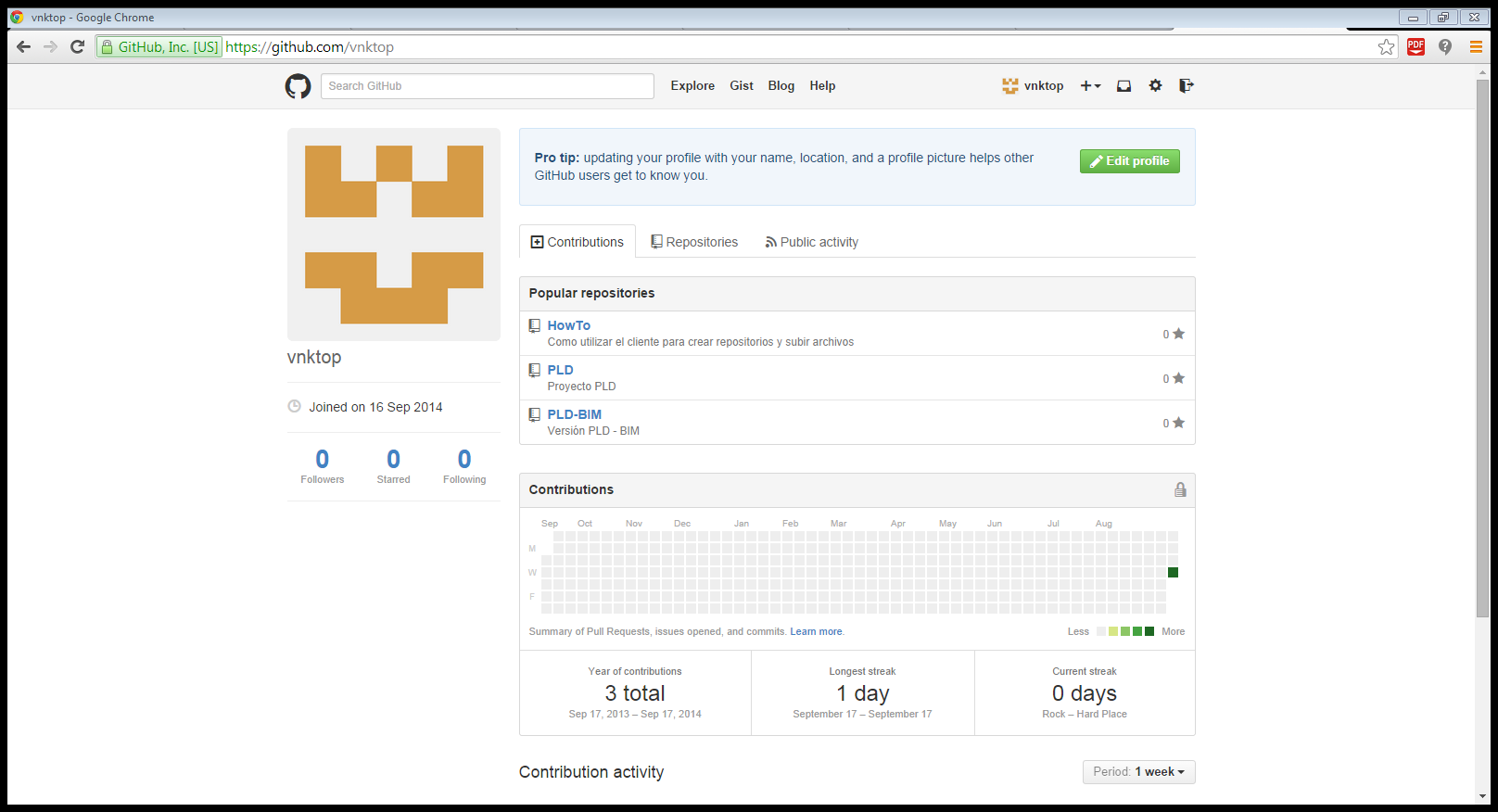
Dar clic en Publish Repository, para enviarlo al servidor de github



Se define el repositorio, dar clic en Publish HowTo

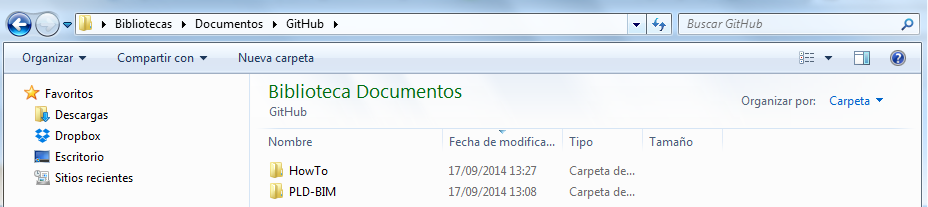


Ahora ya debe aparecer en el repositorio web

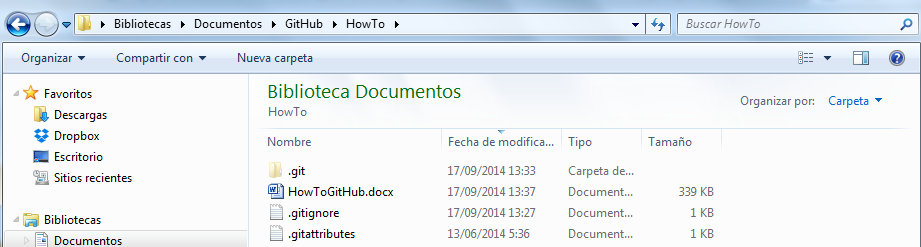


Se realiza la transferencia de un archivo, este mismo

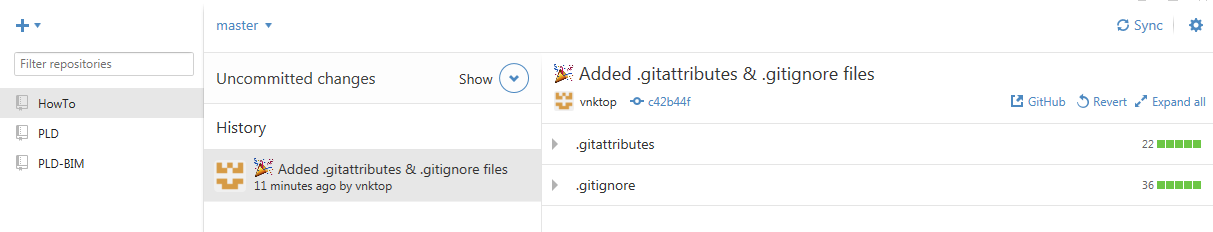
Primero se ubica la dirección del repositorio, que es la misma que se definió en la creación del repositorio

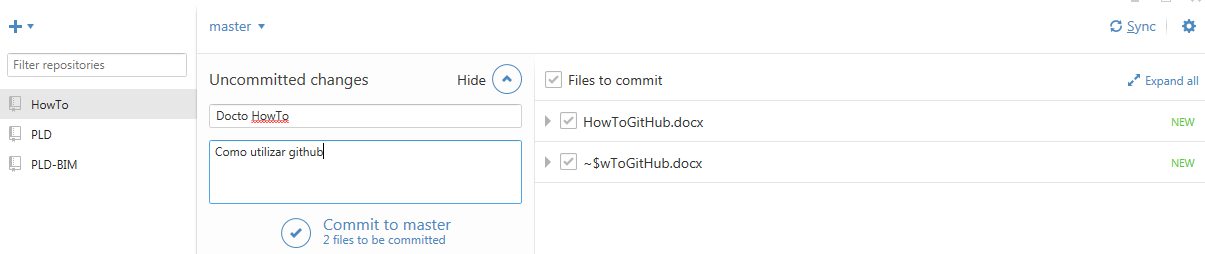


Dentro de esta ruta almacenamos el documento, esto es local

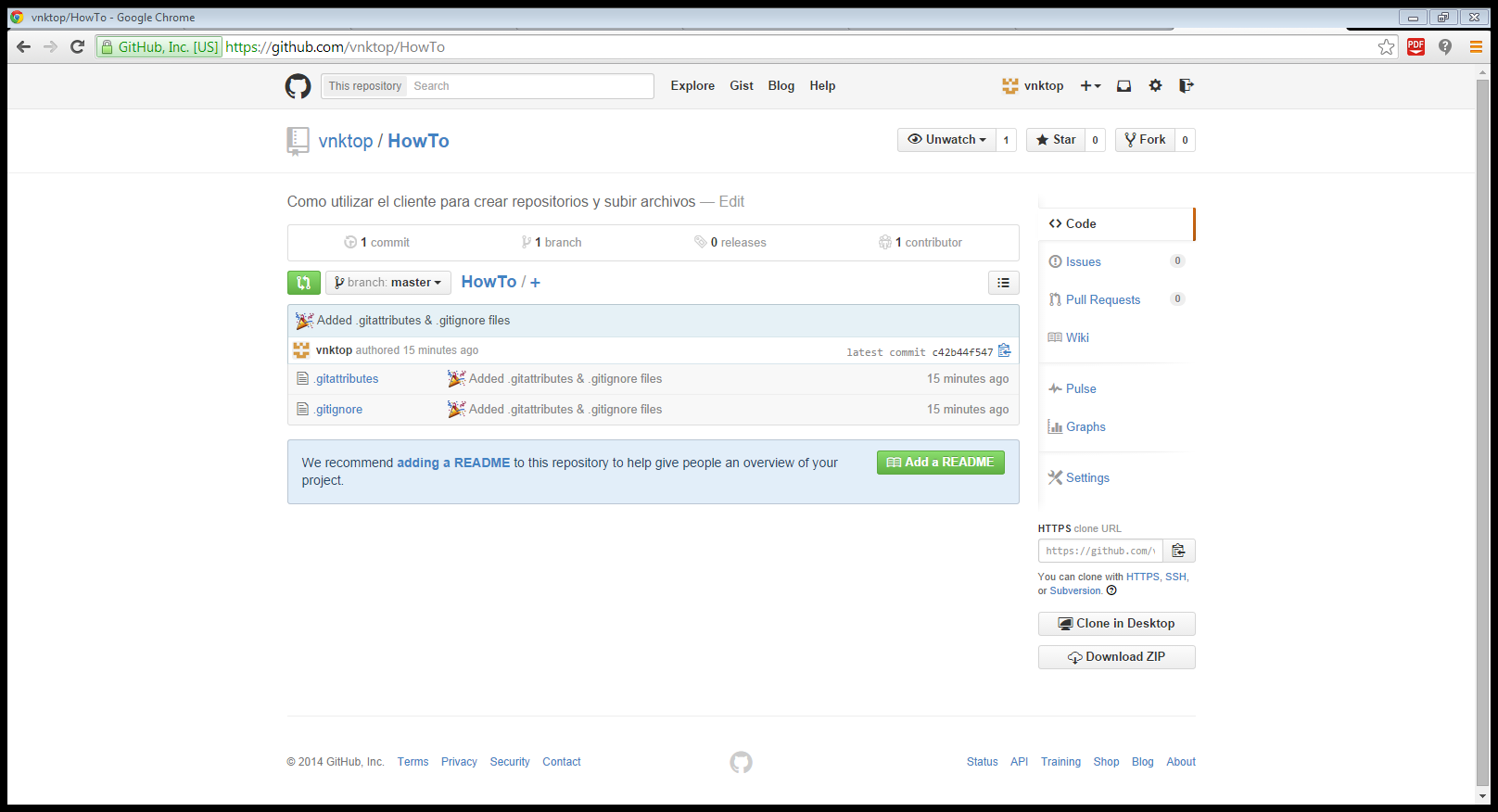


De regreso en el cliente se muestra que existen cambios

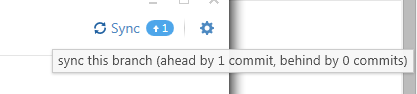
Dar clic en Show para parametrizar el cambio. Observar que aparece ahora el docto Word agregado en el directorio. Dar clic en commit to master



En el sitio remoto no aparece el docto, ya que falta sincronizar la asignación del cambio

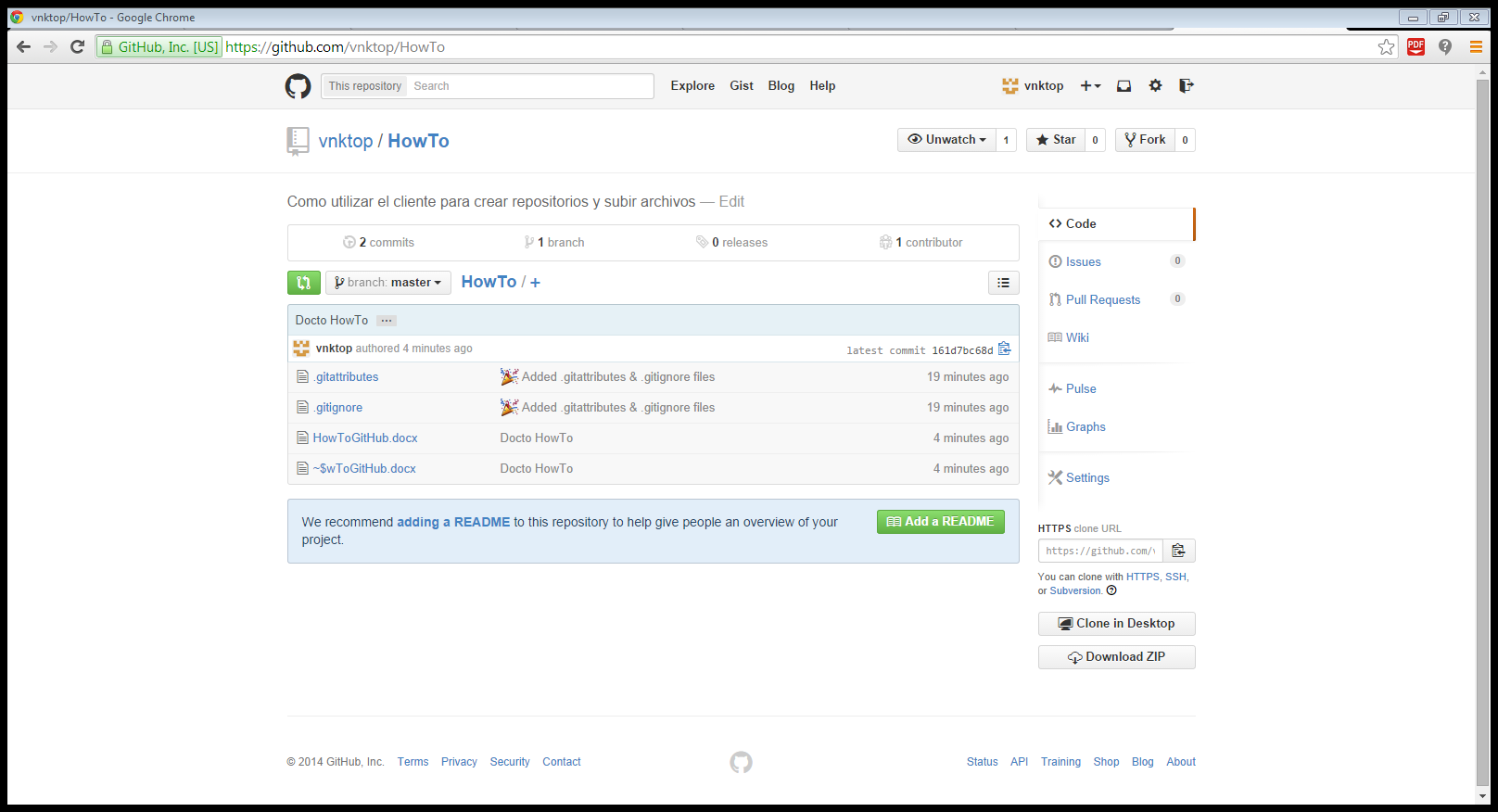


Dar clic en el cliente en el lado superior derecho en sync

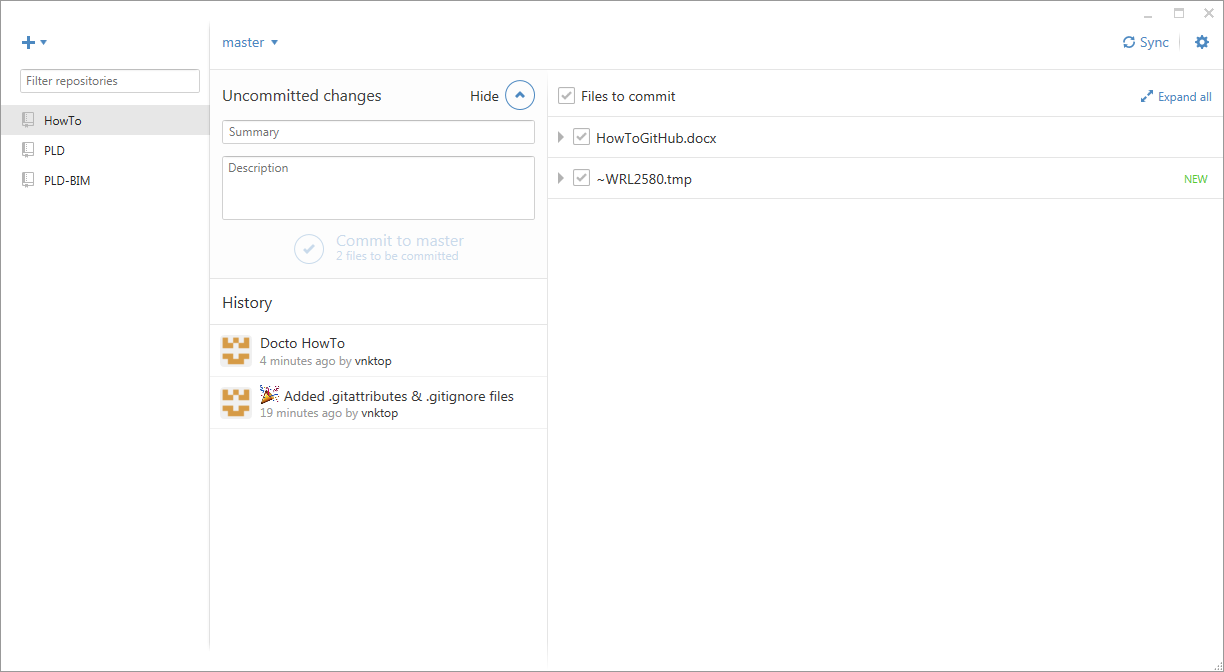


Almacenamos nuestro docto, para dar un primer cierre en este punto

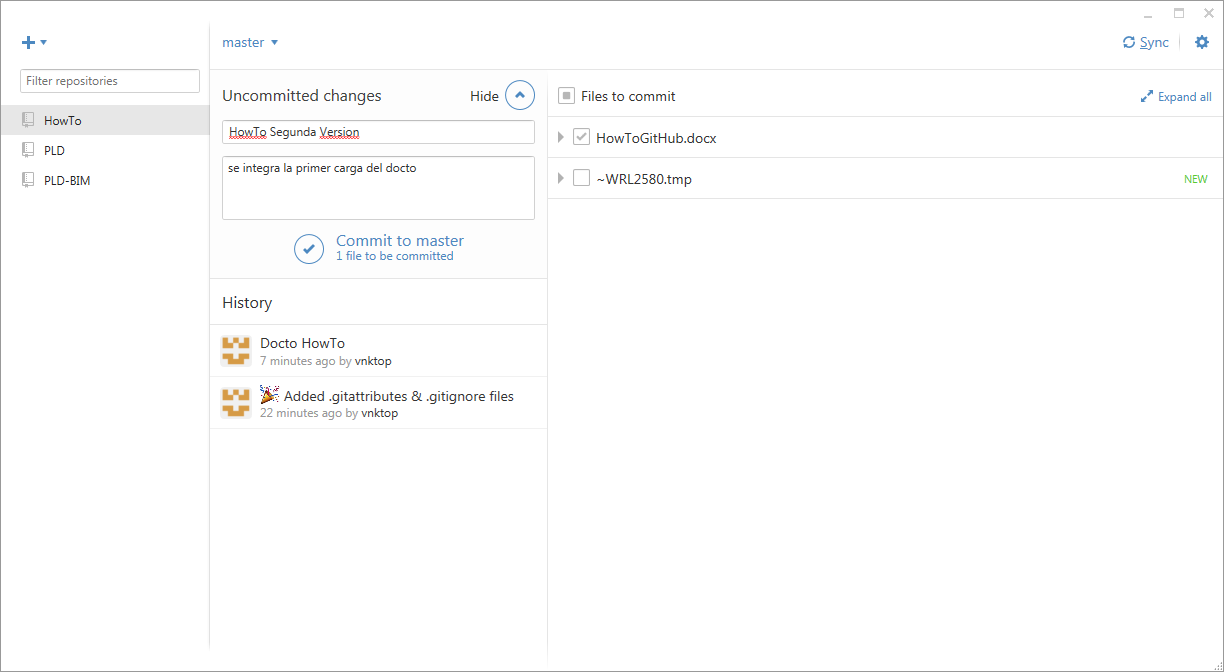
Iniciamos la segunda versión del docto. Ahora debemos observar que ya existe el docto en el sitio remoto



Ahora el cliente detecta los cambios que hemos realizado



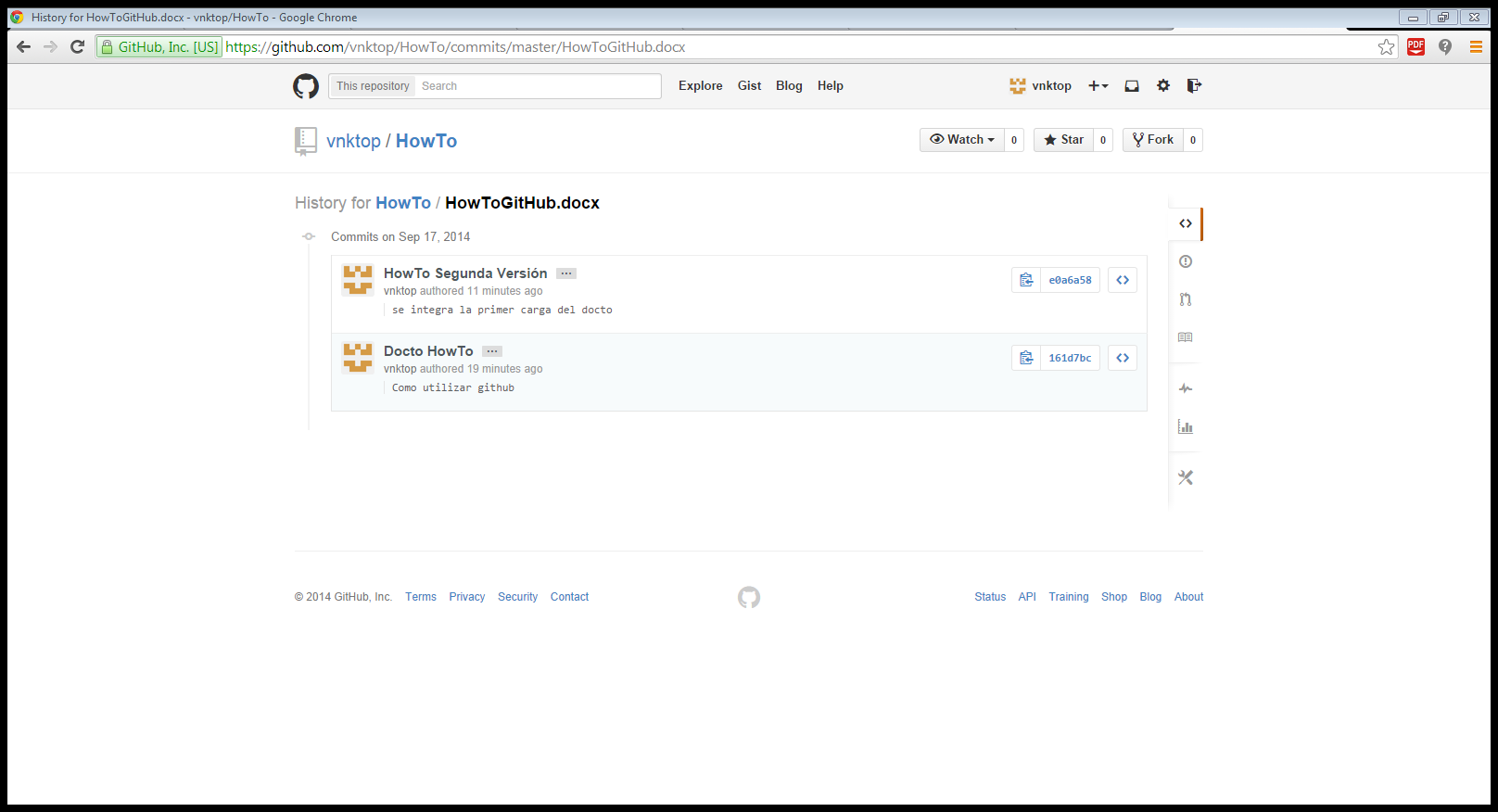
Parametrizar este nuevo cambio y observar que sucede en el sitio remoto para el control de los cambios, en este punto cerramos la segunda versión. Seleccionar los documentos a sincronizar



Sincronizar cambios



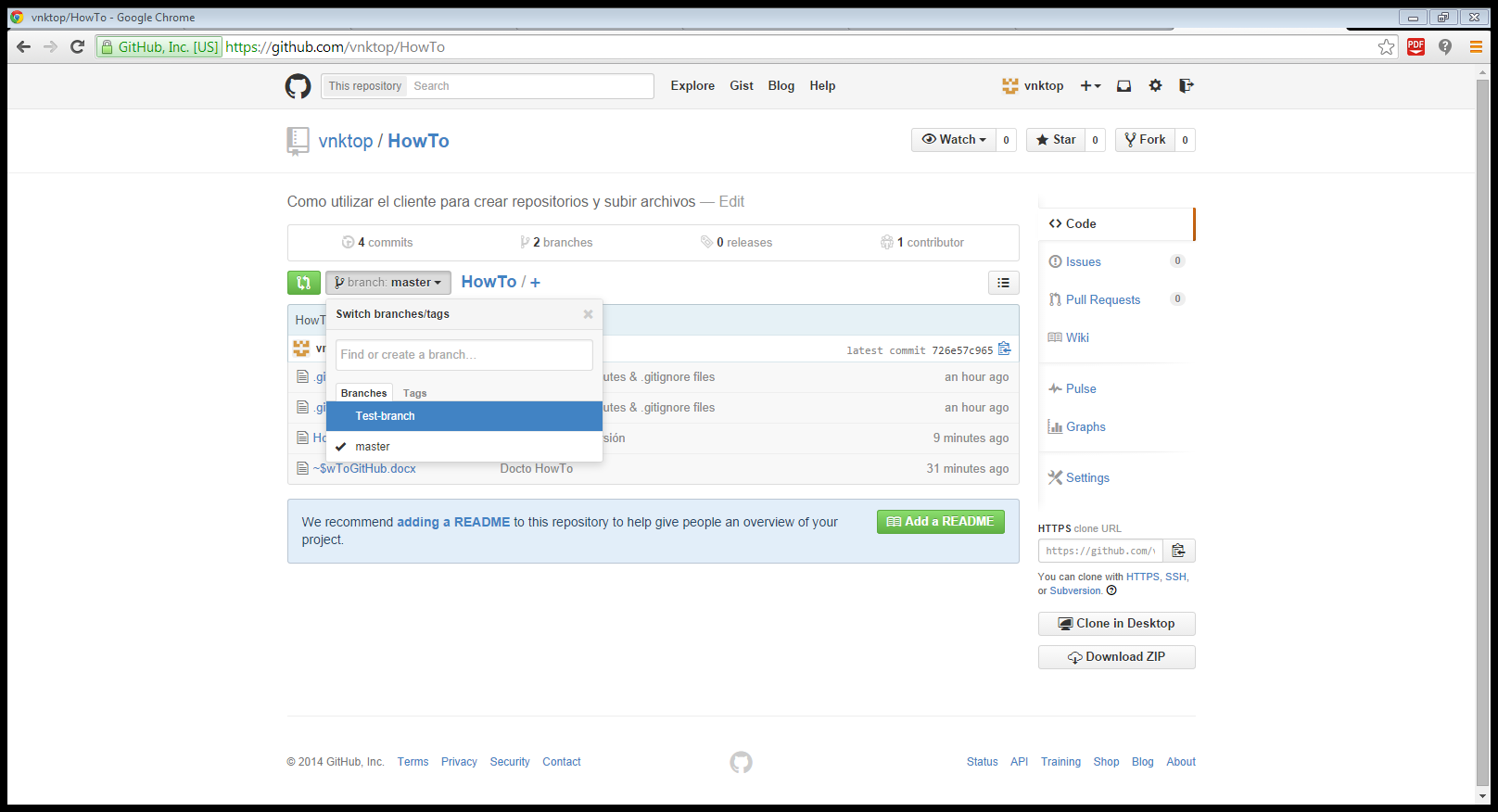
Inicio versión 3. El control de cambios se observa por el history del docto



Almacenamos esta versión como la última para el control de cambios y se debe observar la progresión en el history.

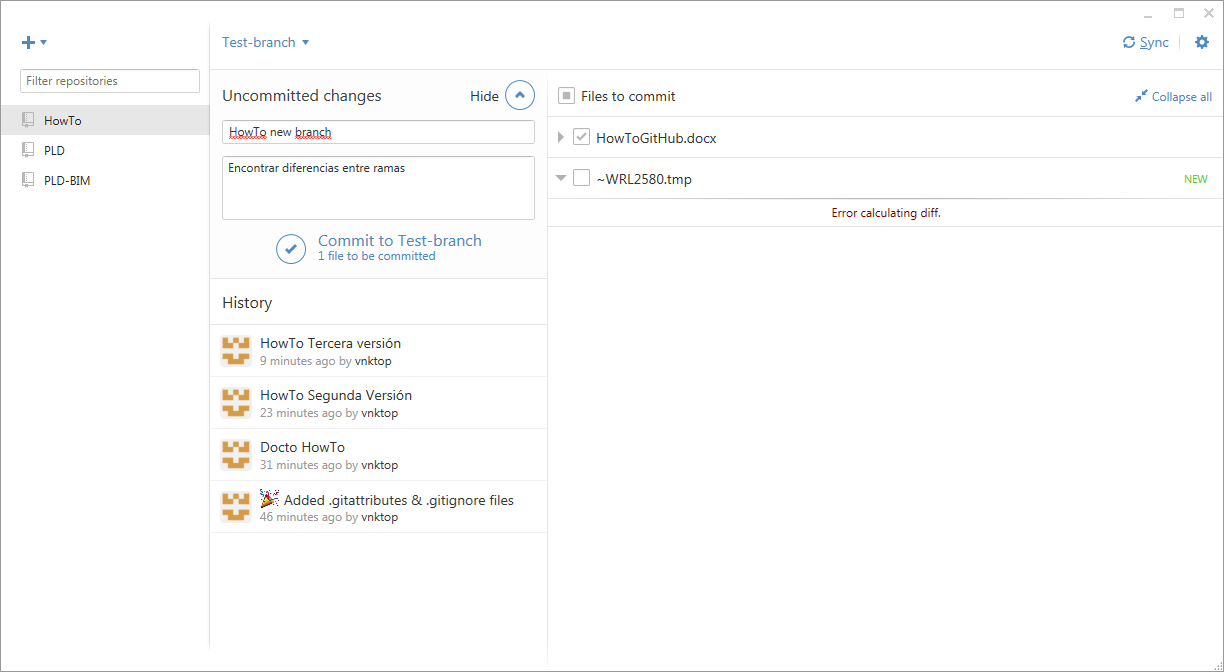
Entonces implementamos los pasos de parametrizar el cambio y sincronizar

Cuarta versión Cambio para verse solo en el branch: Test branch

Se crea branch en sitio remot

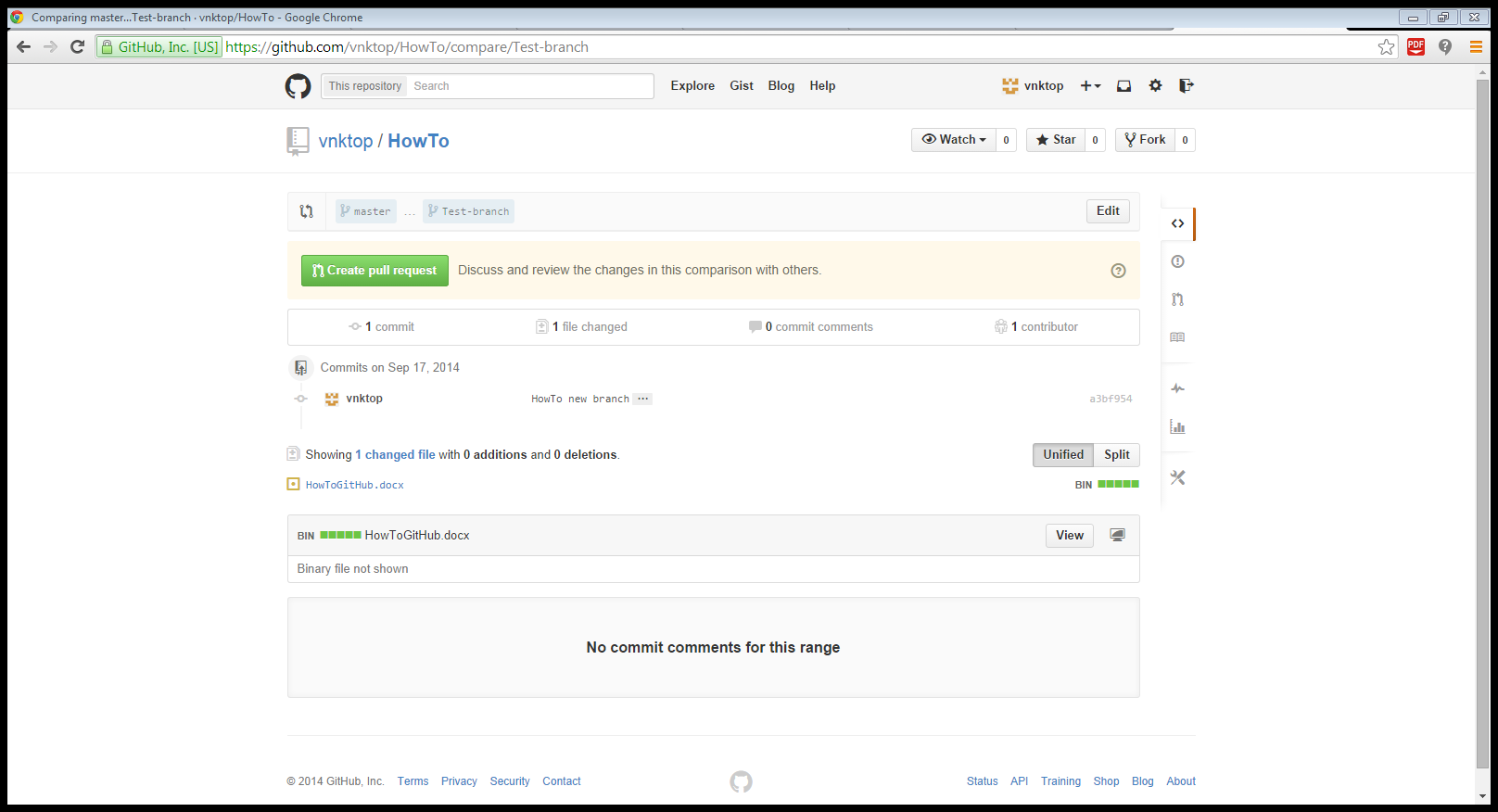
Esto genera el nuevo branch

Ahora se agrega el cambio en el cliente con el branch creado



Se espera encontrar las diferencias, después de asignar la parametrización del cambio y la sincronización, en el sitio remoto

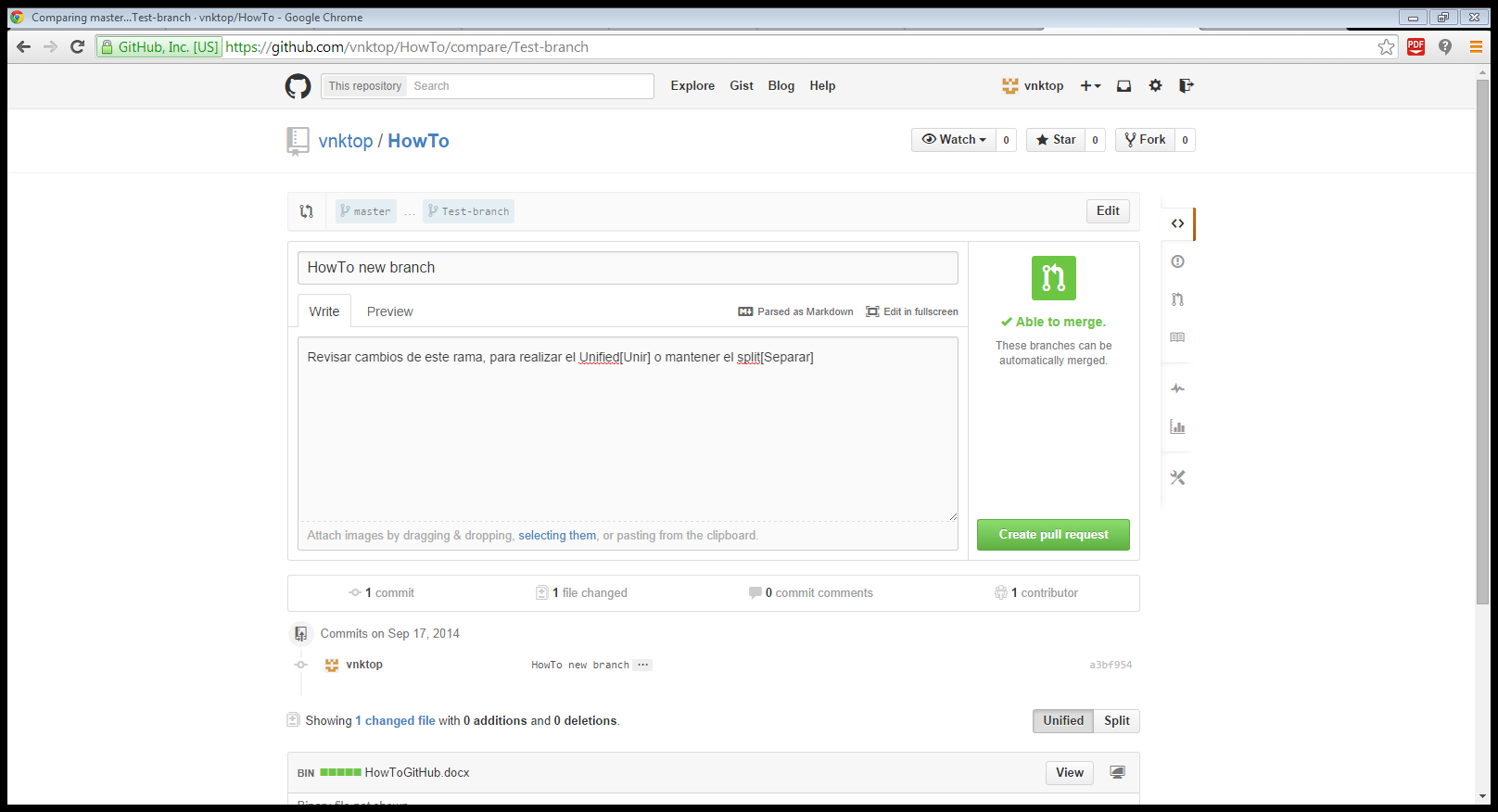
Versión 5, ahora se ven los cambios reportados entre ramas



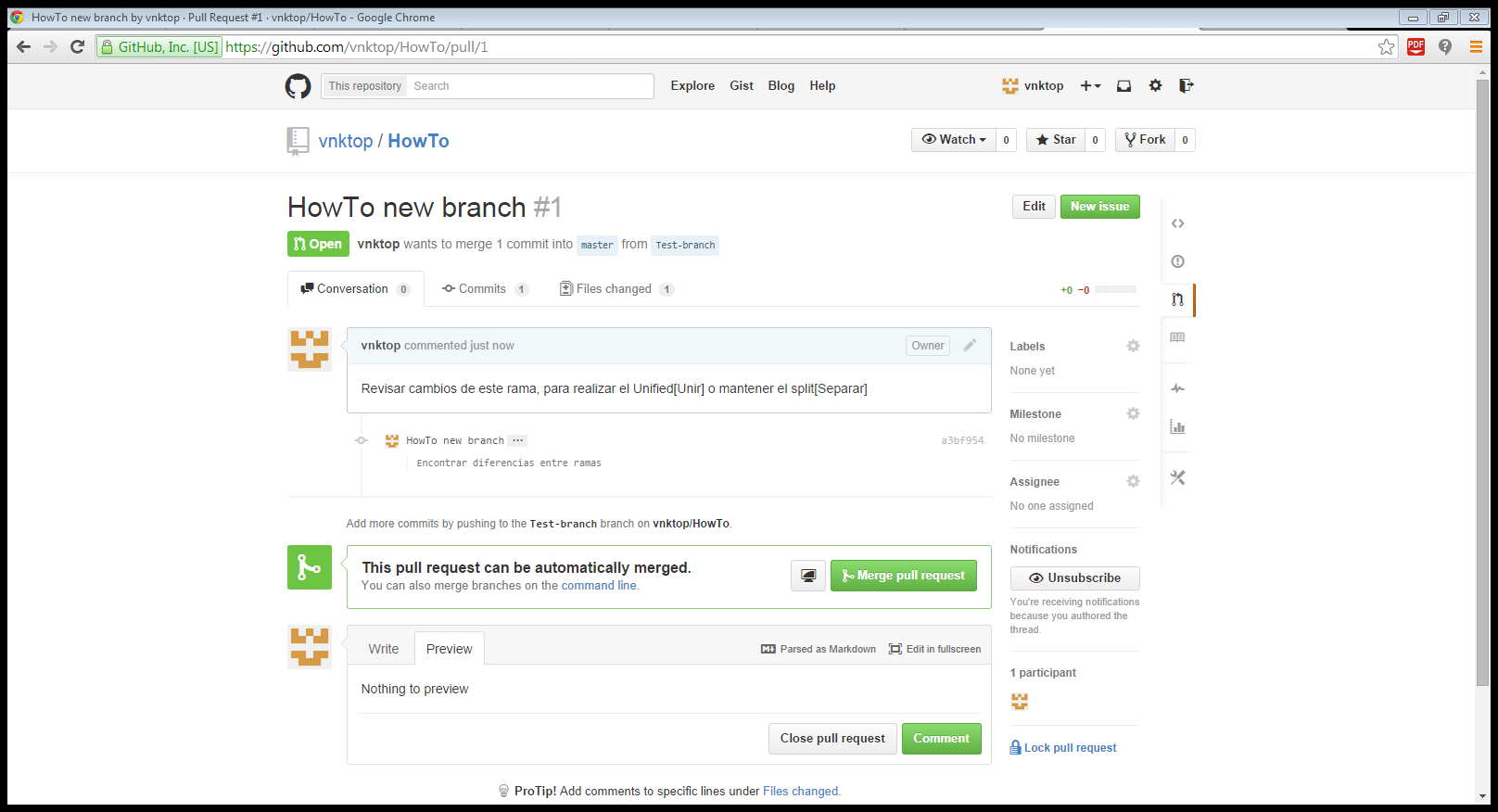
Los documentos se muestran como sigue, manteniedo los cambios entre versiones y ramas

|  |  |
| --- | --- |
| master | New |
|  |  |

Existe en el sitio remoto al opción de Create a pull request, que debe ser algo como un foro de discusión de la diferencia entre ramas

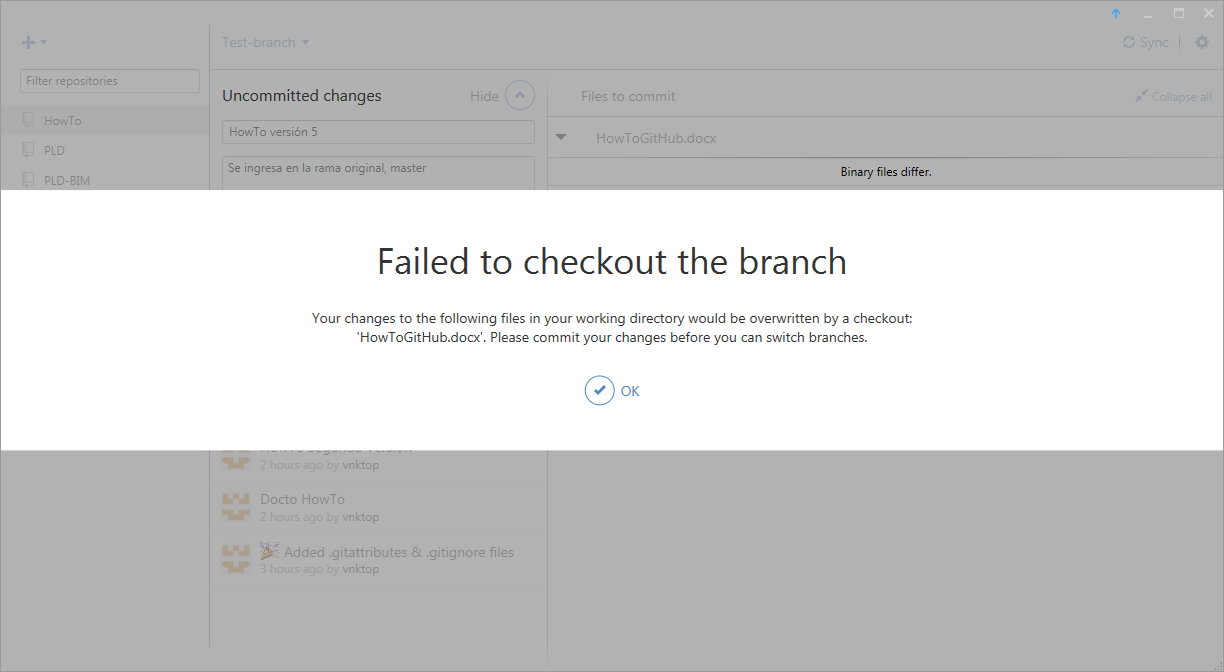


En el sitio remoto se muestra el pull request



Ahora se cierra esta versión, con el branch master. Al final se debe tener una versión completa, realizando el Unified

Interesante, no se permite cambiar entre branches



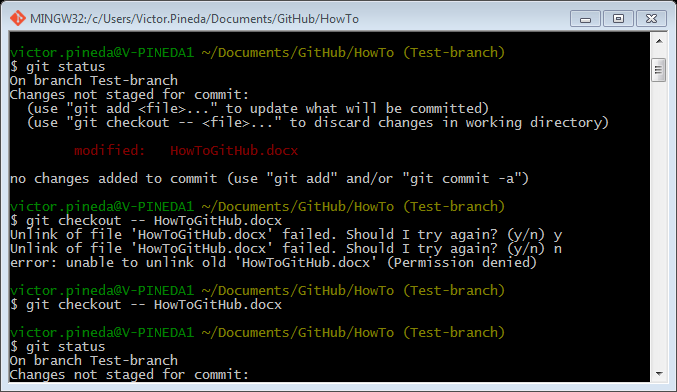
Entonces se realiza el cambio en la rama actual; es decir, Test-branch

Versión 6. Se tienen problemas para poder enlazar el cambio en el cliente, se elimina la relación hecha al branch eliminado, pero también se pierde los cambios hechos, por ello se re escribe ahora

El cambio se realiza en el Shell con los comandos

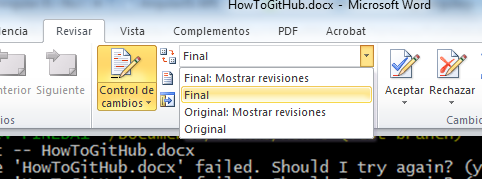
git status, nos informa el error, se dio el comando branch, etc. Se perdió en la versión. Pero posteriormente se da el comando

git chekout – HowToGitHub.docx, pero como no esta cerrado el archivo solicita confirmación, se cierra el archivo y entonces se pierde la versión creada, 6, pero se recupera la 5 y es en la que se documenta ahora



Ahora ya se puede asignar este cambio al branch master

Versión final, se elimina el control de versiones, Revisar-final



Final versión 7