

Migración de SVN a Git



git

Migración de SVN a Git

Desenvolvido por Jesús Amieiro Becerra

Este arquivo encóntrase baixo unha licencia [Creative Commons Recoñecemento-Compartir Igual \(CC BY-SA\)](#). Permítese o uso comercial da obra e das posibles obras derivadas, a distribución das cales se debe facer cunha licencia igual á que regula a obra orixinal.



Versión actual: 31 de xullo de 2017

Revisión: 1

Sumario

1	Introducción.....	4
1.1	Convencions.....	4
2	Requisitos previos.....	5
2.1	Instalación de paquetes.....	5
2.2	Configuración inicial de Git.....	5
3	Migración desde SVN a Git.....	6
3.1	Conversión do repositorio SVN remoto a un repositorio Git local.....	6
3.2	O novo repositorio.....	8
3.3	Conversión do repositorio SVN local a un repositorio Git local.....	8
3.4	Limpeza.....	11
3.5	Compartir o repositorio.....	14
4	Referencias.....	18

1 Introducción

Neste documento explícase como levar a cabo a migración dun sistema de control de versións baseado en Subversion (SVN) a un baseado en Git.

Este tutorial está redactado usando como sistema operativo de traballo unha Ubuntu 16.04.2.

O primeiro que explico son os paquetes que hai que instalar para poder seguir este tutorial.

A continuación realizo a configuración básica de Git e logo levo a cabo un par de exemplos de migración.

1.1 Convencións

– indica que o comando que vén a continuación ten que ser executado con permisos de root directamente co usuario root ou mediante o comando sudo.

\$ – indica que o comando que vén a continuación pode ser executado por un usuario normal sen privilexios de administración.

2 Requisitos previos

2.1 Instalación de paquetes

Este tutorial foi levado a cabo nunha máquina Ubuntu 16.04.2, concretamente nunha Vagrant Homestead

```
# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.2 LTS
Release:        16.04
Codename:       xenial
```

O primeiro que fixen foi instalar os paquetes necesarios, mediante os comandos:

```
# apt-get update
# apt-get install subversion git git-svn gzip default-jre
```

que instala:

- **Subversion**, o sistema de control de versións centralizado. Máis información [neste enlace](#).
- **Git**, o sistema de control de versións distribuído. Máis información [neste enlace](#).
- **git-svn**, conector que permite unha comunicación bidireccional ente Git e Suversion. Máis información [neste enlace](#).
- **gzip**, ferramentas de compresión e descompresión de arquivos. Máis información [neste enlace](#).
- **default-jre**, é o runtime de Java. Máis información [neste enlace](#).

2.2 Configuración inicial de Git

Para poder seguir este tutorial hai que realizar a configuración básica de Git, para que Git coñeza o noso nome de usuario e o correo electrónico.

Se estás seguindo este tutorial só tes que cambiar o meu nome e correo electrónico polo teu.

```
$ git config --global user.name "Jesús Amieiro"
$ git config --global user.email jesus@jesusamieiro.com
```

3 Migración desde SVN a Git

A migración dun repositorio SVN en produción a Git vou a levala a cabo en varios pasos:

- Clonado do repositorio SVN a un repositorio Git local.
- Conversión do repositorio SVN a un repositorio Git local.
- Publicación do repositorio Git local nun repositorio Git remoto.
- Paso do uso do servidor SVN ao servidor GIT.

3.1 Conversión do repositorio SVN remoto a un repositorio Git local

Para este fin vou a usar como repositorio de exemplo un dos moitos repositorios públicos SVN dispoñibles no servidor da EEA (European Environment Agency): <https://svn.eionet.europa.eu/>

En concreto, vou a empregar o proxecto “GeoNetwork Inspire”, cuxo repositorio SVN se encontra na URL <https://svn.eionet.europa.eu/repositories/GNInspire>

O primeiro que vou a facer vai a ser o clonado, que pode tardar un rato en levarse a cabo, dependendo da velocidade da liña de datos e da potencia do equipo onde se leva a cabo:

```
$ cd /tmp
$ git svn clone https://svn.eionet.europa.eu/repositories/GNInspire/ -T trunk -b
branches -t tags
```

Se mostro os committers vexo que non se adaptan ao formato de nome e correo electrónico que usa Git.

```
$ cd /tmp/GNInspire
$ git shortlog -sne
   62  picavvin <picavvin@6ea4d1b0-2fa9-4423-a4f2-244cad39d1e9>
   41  francpru <francpru@6ea4d1b0-2fa9-4423-a4f2-244cad39d1e9>
   25  chartben <chartben@6ea4d1b0-2fa9-4423-a4f2-244cad39d1e9>
    3  hasenpau <hasenpau@6ea4d1b0-2fa9-4423-a4f2-244cad39d1e9>
```

Polo tanto, o primeiro que vou a facer é establecer a referencia entre o usuario de SVN e o usuario e correo electrónico de Git.

Para isto vou a descargar unha ferramenta que publicou Atlassian

```
$ cd /tmp
$ wget https://bitbucket.org/atlassian/svn-migration-scripts/downloads/svn-
migration-scripts.jar
```

A continuación creo o arquivo de usuarios a partir do repositorio SVN

```
$ java -jar ./svn-migration-scripts.jar authors
https://svn.eionet.europa.eu/repositories/GNInspire/ > authors.txt
```

Se consulto o arquivo creado (authors.txt)

```
$ more authors.txt
chartben = chartben <chartben@mycompany.com>
francpru = francpru <francpru@mycompany.com>
hasenpau = hasenpau <hasenpau@mycompany.com>
picavvin = picavvin <picavvin@mycompany.com>
```

Teño na parte esquerda o usuario de SVN e na parte dereita o usuario e o correo electrónico cos que identificarei aos committers en Git.

O que vou a facer é cambiar a parte dereita, a que vén despois do símbolo “=” para adaptala ao formato que usa Git (nome completo e correo electrónico, os nomes e correos electrónicos usados neste exemplo son inventados), quedando da seguinte forma:

```
$ more authors.txt
chartben = Chart Ben <chartben@example.com>
francpru = Franc Prunus <francpru@example.com>
hasenpau = Hasen Paul <hasenpau@example.com>
picavvin = Picav Vinicius <picavvin@example.com>
```

Agora xa pode facer a conversión correcta.

Primeiro borro o repositorio Git creado previamente.

```
$ cd /tmp
$ rm GNInspire/ -rf
```

Volvo a facer a conversión, pero cun novo parámetro: --authors-file=authors.txt, que indica a relación entre os usuarios de SVN e os de Git

```
$ cd /tmp
$ git svn clone https://svn.eionet.europa.eu/repositories/GNInspire/ -T trunk -b
branches -t tags --authors-file=authors.txt
```

Se volvo a comprobar os committers, xa aparecen no formato empregado por Git

```
$ cd /tmp/GNInspire
$ git shortlog -sne
 62 Picav Vinicius <picavvin@example.com>
 41 Franc Prunus <francpru@example.com>
 25 Chart Ben <chartben@example.com>
  3 Hasen Paul <hasenpau@example.com>
```

Polo tanto, o primeiro que teño que facer é crear o arquivo de usuarios ou committers onde se vinculan os de SVN e os de Git antes de realizar o clonado a un repositorio Git.

3.2 O novo repositorio

Podo ver as ramas creadas, onde só teño unha rama local, executo

```
$ cd /tmp/GNInspire
$ git branch -a
* master
remotes/origin/trunk
```

Podo ver o número de commits

```
$ git rev-list --all --count
131
```

Podo consultar o último commit da rama master

```
$ git log -1
commit 58798088fe136d696d923533ccdf545620c97f22
Author: Hasen Paul <hasenpau@example.com>
Date: Thu Sep 25 11:58:08 2014 +0000

    git-svn-id: https://svn.eionet.europa.eu/repositories/GNInspire/trunk@131
    6ea4d1b0-2fa9-4423-a4f2-244cad39d1e9
```

Unha vez que teño o repositorio configurado en local, o seguinte que teño que facer é publicalo nun servidor Git, para que os participantes no proxecto poidan acceder a el e traballar, tal e como se explica no apartado 3.5.

Pero antes vou a mostrar un caso un pouco máis complexo, con ramas e etiquetas a partir dunha copia de seguridade dun repositorio SVN.

3.3 Conversión do repositorio SVN local a un repositorio Git local

Neste segundo exemplo vou a facer a conversión partindo dunha copia de seguridade dun repositorio SVN, polo que o primeiro que vou a facer será configurar un repositorio SVN. Creo o directorio onde vai a estar o servidor

```
$ mkdir -p /tmp/test-svn
```

Descargo a copia de seguridade do repositorio SVN de exemplo

```
$ cd /tmp/
$ wget https://storage.googleapis.com/google-code-archive-source/v2/code.google.com/progit-example/repo.svndump.gz
```

Descomprimo o arquivo


```
$ gunzip repo.svndump.gz
```

Acabo de obter o dump ou volcado, cun tamaño de 12 MB

```
$ ls
repo.svndump
$ du repo.svndump -sh
12M    repo.svndump
```

Creo o repositorio SVN

```
$ svnadmin create /tmp/test-svn
```

Cargo o dump no repositorio creado (pode tardar un rato en levarse a cabo)

```
$ svnadmin load /tmp/test-svn/ < repo.svndump
```

O seguinte que vou a facer é crear o arquivo de committers

```
$ java -jar ./svn-migration-scripts.jar authors file:///tmp/test-svn/ >
authors2.txt
About to create the authors file.
```

Comprobo o seu contido

```
$ more authors2.txt
kenton@google.com = Kenton <kenton@google.com>
lisafcarey = lisafcarey <lisafcarey@mycompany.com>
schacon = schacon <schacon@mycompany.com>
temporal = temporal <temporal@mycompany.com>
```

Edito o arquivo (algúns dos nomes e correos electrónicos son inventados)

```
kenton@google.com = Kenton Varda <kenton@google.com>
lisafcarey = Lisa Carey <lisafcarey@example.com>
schacon = Steve Chacon <schacon@example.com>
temporal = Usuario descoñecido <temporal@example.com>
```

Se intento facer o clonado con este arquivo vou a obter este erro:

```
$ git svn clone file:///tmp/test-svn/ -T trunk -b branches -t tags --authors-
file=/tmp/authors2.txt
Initialized empty Git repository in /tmp/test-svn/.git/
Author: (no author) not defined in /tmp/authors2.txt file
```

O que pasa é que non atopa ningunha referencia no caso de que non exista autor nalgún commit en SVN. Para poder continuar, teño que engadir a seguinte liña ao arquivo de autores (/tmp/authors2.txt):

```
(no author) = no_author <no_author@no_author>
```

Agora que teño o repositorio creado, o seguinte será clonalo en git. Para isto, creo un directorio

```
$ mkdir /tmp/test-git
$ cd /tmp/test-git
```

E clono o repositorio

```
$ git svn clone file:///tmp/test-svn/ -T trunk -b branches -t tags --authors-
file=/tmp/authors2.txt
```

Cando acaba o proceso vexo que crea unha nova carpeta

```
$ ls
test-svn
```

Accedo a este directorio, onde se encontra o repositorio

```
$ cd test-svn/
```

O primeiro que comprobo son as ramas creadas, onde só teño unha rama local. Ademais as etiquetas aparecen como ramas remotas.

```
$ git branch -a
* master
remotes/origin/my-calc-branch
remotes/origin/tags/2.0.2
remotes/origin/tags/release-2.0.1
remotes/origin/tags/release-2.0.2
remotes/origin/tags/release-2.0.2rc1
remotes/origin/trunk
```

Podo ver, por exemplo, o número de commits na rama master

```
$ git rev-list master --count
40
```

Podo consultar o último commit da rama master

```
$ git log -1
commit b7c63105d85a953b8ae14d2d6da1b4d4c9639d40
```

```
Author: kenton@google.com <kenton@google.com@aa94d90a-e0fd-4ac7-9478-8ef69023a4cb>
Date: Fri Nov 14 17:29:32 2008 +0000
```

```
Support HP C++ on Tru64.
```

```
Patch (mostly) by Vincent Choinière <Choiniere.Vincent@hydro.qc.ca>.
```

```
git-svn-id: file:///tmp/test-svn/trunk@75 aa94d90a-e0fd-4ac7-9478-8ef69023a4cb
```

3.4 Limpeza

Agora teño que converter as ramas remotas en ramas locais, ao igual que as etiquetas (tags) e borrar as ramas de seguimento. Neste momento deixarei de ter vinculado o repositorio SVN, polo que non poderei volver a sincronizar o repositorio Git dende o repositorio SVN.

Isto debería de poder facelo de forma automática co mesmo script que estou usando (o facilitado por Atlassian), pero hai un bug neste script que o fai incompatible coa versión 2 de Git.

Para ver o que faría o comando executaría

```
$ cd /tmp/test-git/test-svn
$ java -Dfile.encoding=utf-8 -jar /tmp/svn-migration-scripts.jar clean-git
#####
#          This is a dry run, add --force to commit          #
#          No changes will be made to your repository        #
#####
# Creating annotated tags...
# Creating local branches...
Creating the local branch 'origin/my-calc-branch' for Subversion branch
'refs/remotes/origin/my-calc-branch'.
Creating the local branch 'origin/tags/2.0.2' for Subversion branch
'refs/remotes/origin/tags/2.0.2'.
Creating the local branch 'origin/tags/release-2.0.1' for Subversion branch
'refs/remotes/origin/tags/release-2.0.1'.
Creating the local branch 'origin/tags/release-2.0.2' for Subversion branch
'refs/remotes/origin/tags/release-2.0.2'.
Creating the local branch 'origin/tags/release-2.0.2rc1' for Subversion branch
'refs/remotes/origin/tags/release-2.0.2rc1'.
Creating the local branch 'origin/trunk' for Subversion branch
'refs/remotes/origin/trunk'.
# Checking for obsolete tags...
No obsolete tags to remove.
# Checking for obsolete branches...
```

```

No obsolete branches to remove.
# Cleaning tag names
# Cleaning branch names
#####
#           This is a dry run, add --force to commit           #
#           No changes will be made to your repository         #
#####

```

E para ejecutarlo, tendría que engadir o parámetro "--force".

```

$ java -Dfile.encoding=utf-8 -jar /tmp/svn-migration-scripts.jar clean-git --force
# Creating annotated tags...
# Creating local branches...
Creating the local branch 'origin/my-calc-branch' for Subversion branch
'refs/remotes/origin/my-calc-branch'.
Creating the local branch 'origin/tags/2.0.2' for Subversion branch
'refs/remotes/origin/tags/2.0.2'.
Creating the local branch 'origin/tags/release-2.0.1' for Subversion branch
'refs/remotes/origin/tags/release-2.0.1'.
Creating the local branch 'origin/tags/release-2.0.2' for Subversion branch
'refs/remotes/origin/tags/release-2.0.2'.
Creating the local branch 'origin/tags/release-2.0.2rc1' for Subversion branch
'refs/remotes/origin/tags/release-2.0.2rc1'.
Creating the local branch 'origin/trunk' for Subversion branch
'refs/remotes/origin/trunk'.
# Checking for obsolete tags...
No obsolete tags to remove.
# Checking for obsolete branches...
Deleting Git branch 'origin/tags/release-2.0.1' not in Subversion.
Deleted branch origin/tags/release-2.0.1 (was d87fcb6).
Deleting Git branch 'origin/tags/release-2.0.2' not in Subversion.
Deleted branch origin/tags/release-2.0.2 (was c0b8b6f).
Deleting Git branch 'origin/trunk' not in Subversion.
Deleted branch origin/trunk (was ced6191).
Deleting Git branch 'origin/tags/release-2.0.2rc1' not in Subversion.
Deleted branch origin/tags/release-2.0.2rc1 (was b546fe1).
Deleting Git branch 'origin/my-calc-branch' not in Subversion.
Deleted branch origin/my-calc-branch (was 448acae).
Deleting Git branch 'origin/tags/2.0.2' not in Subversion.
Deleted branch origin/tags/2.0.2 (was b37fe8b).
# Cleaning tag names

```

```
# Cleaning branch names
```

Se volvo a comprobar as ramas, están igual

```
$ git branch -a
* master
remotes/origin/my-calc-branch
remotes/origin/tags/2.0.2
remotes/origin/tags/release-2.0.1
remotes/origin/tags/release-2.0.2
remotes/origin/tags/release-2.0.2rc1
remotes/origin/trunk
```

E non hai ningunha etiqueta

```
$ git tag
```

Polo tanto, teño que levar a cabo este traballo de forma manual.

Creo a rama local a partir da rama remota

```
$ git branch my-calc-branch origin/my-calc-branch
```

Creo as etiquetas de forma manual

```
$ git tag 2.0.2 origin/tags/2.0.2
$ git tag release-2.0.1 origin/tags/release-2.0.1
$ git tag release-2.0.2 origin/tags/release-2.0.2
$ git tag release-2.0.2rc1 origin/tags/release-2.0.2rc1
```

Se comprobo as etiquetas aparecen as etiquetas correctas

```
$ git tag
2.0.2
release-2.0.1
release-2.0.2
release-2.0.2rc1
```

Para finalizar teño que borrar as ramas de seguimento

```
$ git branch -rd origin/trunk
$ git branch -rd origin/my-calc-branch
$ git branch -rd origin/tags/2.0.2
$ git branch -rd origin/tags/release-2.0.1
```

```
$ git branch -rd origin/tags/release-2.0.2
$ git branch -rd origin/tags/release-2.0.2rc1
```

Se volvo a comprobar as ramas, só quedan as 2 ramas locais

```
$ git branch -a
* master
  my-calc-branch
```

3.5 Compartir o repositorio

Agora só me queda compartir o repositorio co meu equipo de traballo.

En SVN, os desenvolvedores comparten as contribucións ao realizar cambios desde unha copia de traballo no seu equipo local a un repositorio central. Posteriormente, outros desenvolvedores obteñen estas actualizacións do repositorio central nas súas copias locais de traballo para poder continuar o desenvolvemento. Este fluxo de traballo é así porque SVN é un sistema de control de versións centralizado.

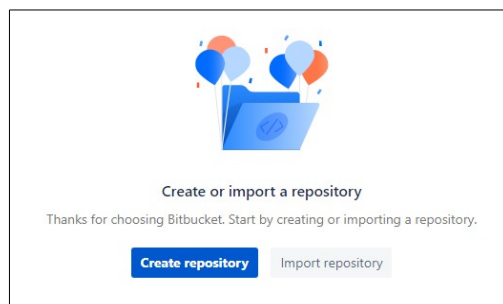
O fluxo de traballo de colaboración en Git é moi diferente. En lugar de diferenciar entre as copias de traballo e o repositorio central, Git proporciona a cada desenvolvedor a súa propia copia local de todo o repositorio. Os cambios almacénanse mediante commits no repositorio local en lugar dun repositorio central. Para compartir as actualizacións con outros desenvolvedores, cada desenvolvedor debe transferir estes cambios locais a un repositorio compartido de Git nun servidor. A continuación, os outros desenvolvedores poden extraer os commits do repositorio compartido nos seus propios repositorios locais.

Vou a explicar como compartir o repositorio en Bitbucket, co que vou a partir dunha conta creada neste servizo de repositorios de Git. O modo de traballo sería semellante en sistemas similares, tales como GitHub, GitLab,...

É importante ter en conta que calquera destes servizos limitan a colaboración a usuarios desa plataforma.

Tamén é importante que a partir deste momento o servidor SVN se poña en modo lectura, de tal forma que os commits se leven a cabo en local e se compartan a través deste novo servidor de Git, para evitar perder traballo con commits na plataforma incorrecta. Neste momento, a labor de comunicación por parte do encargado da migración é fundamental.

Tras acceder co meu usuario a BitBucket (<https://bitbucket.org/>), fago clic no botón “Create repository”.



Introduzo os datos do repositorio e fago clic no botón “Create repository”.

A continuación volvo á liña de comandos e engado o servidor mediante o comando

```
$ git remote add origin https://JesusAmieiro@bitbucket.org/JesusAmieiro/test-svn.git
```

Obviamente, a URL cambiará en función do usuario de BitBucket e do nome do repositorio.

A continuación realizo o push de todas as ramas

```
$ git push -u origin --all
Password for 'https://JesusAmieiro@bitbucket.org':
Counting objects: 939, done.
Compressing objects: 100% (327/327), done.
Writing objects: 100% (939/939), 1.00 MiB | 0 bytes/s, done.
Total 939 (delta 560), reused 939 (delta 560)
To https://JesusAmieiro@bitbucket.org/JesusAmieiro/test-svn.git
 * [new branch]      master -> master
 * [new branch]      my-calc-branch -> my-calc-branch
Branch master set up to track remote branch master from origin.
Branch my-calc-branch set up to track remote branch my-calc-branch from origin.
```

E tamén de todas as etiquetas

```

$ git push --tags
Password for 'https://JesusAmieiro@bitbucket.org':
Counting objects: 18, done.
Compressing objects: 100% (15/15), done.
Writing objects: 100% (18/18), 2.05 KiB | 0 bytes/s, done.
Total 18 (delta 11), reused 10 (delta 3)
To https://JesusAmieiro@bitbucket.org/JesusAmieiro/test-svn.git
* [new tag]          2.0.2 -> 2.0.2
* [new tag]          release-2.0.1 -> release-2.0.1
* [new tag]          release-2.0.2 -> release-2.0.2
* [new tag]          release-2.0.2rc1 -> release-2.0.2rc1

```

Se accedo ao historial de commits de BitBucket podó ver as ramas e as etiquetas que acabo de enviar dende a consola ao servidor de BitBucket

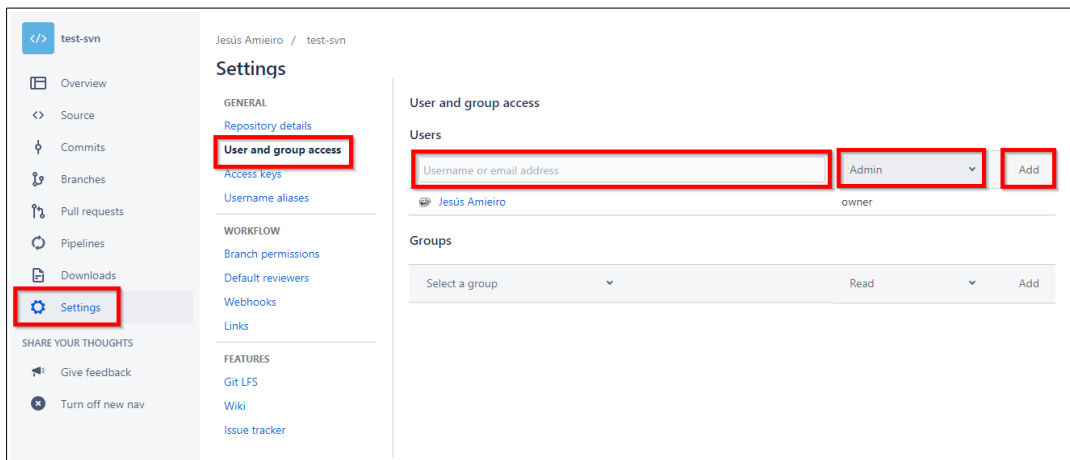
The screenshot shows the 'Commits' page in BitBucket. The table lists commits with columns for Author, Commit ID, Message, Date, and Builds. Several commits are highlighted with red boxes, indicating tags and branches created during the migration process.

Author	Commit	Message	Date	Builds
Steve Chacon	448acae	created a branch git-svn-id: file:///tmp... my-calc-branch	2009-05-02	
Kenton Varda	ced6191	Support HP C++ on Tru64. Patch (mostly) by Vincent Cho...	2008-11-14	
Kenton Varda	fc7a97f	Fix bugs in zero_copy_stream_impl. Patch from Frank Chu...	2008-11-07	
Kenton Varda	c5501d9	Only build tests when running "make check", not "make". ...	2008-11-07	
Kenton Varda	707da0d	Fixed alignment issue that caused bus errors on platform...	2008-11-07	
Kenton Varda	4a146f1	Simplify some template usage so that older compilers ca...	2008-11-06	
Kenton Varda	f3f3acc	Work around GCC 4.3.0 x86_64 compiler bug (seen on Fe...	2008-11-06	
Kenton Varda	71515c0	Update MSVC installation instructions for static linking ch...	2008-10-21	
Kenton Varda	d40d596	Avoid an "unused parameter" warning when using high w...	2008-10-17	
Kenton Varda	5d3b516	Update version number to 2.0.3-SNAPSHOT. git-svn-id: fil...	2008-10-17	
Kenton Varda	c60ed6e	Make Python example output identical to C++ and Java ...	2008-10-07	
Kenton Varda	c08b6f	Set version to 2.0.2 in release branch. git-svn-id: file:///tmp/test-svn/t...	2008-10-03	
Kenton Varda	b578e0c	Tag release 2.0.2 (take 2). git-svn-id: file:///tmp/test-svn/t...	2008-10-03	
Kenton Varda	b37fe8b	Tag release 2.0.2. git-svn-id: file:///tmp/test-svn...	2008-10-03	
Kenton Varda	17308de	* Fixed template specialization syntax that MSVC 2005 di...	2008-10-03	
Kenton Varda	84afcbb	Improvements to configure.ac from Jeff Bailey. git-svn-id:...	2008-10-01	
Kenton Varda	a9b80d0	Pass -Wall,no-obsolete to autoreconf. git-svn-id: file:///t...	2008-10-01	
Kenton Varda	96dc4ad	Improve autogen.sh. git-svn-id: file:///tmp/test-svn/trunk...	2008-09-30	
Kenton Varda	4bbfaF4	Added clarification to license that generated code is own...	2008-09-30	
Kenton Varda	b546fe1	Update version number in 2.0.2rc1 r... release-2.0.2rc1	2008-09-30	

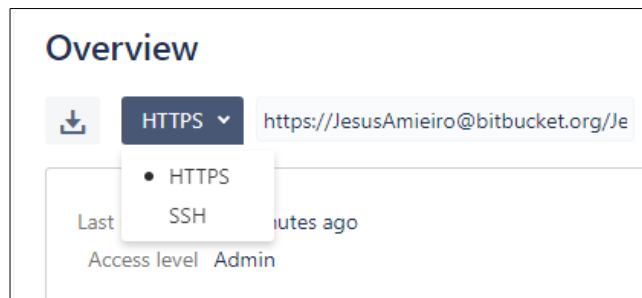
Tan só me queda compartir o proxecto cos meus compañeiros de equipo ou coa xente que quero que teña acceso ao repositorio.

Para isto, teño que ir á configuración do repositorio (Settings), logo a “User and group access”, buscar o usuario en BitBucket, o permiso que lle quero asignar (lectura, escritura ou administración) e facer clic

no botón “Add”.



O novo usuario, tras acceder á súa conta de BitBucket, ten que seleccionar o protocolo de comunicación que quere empregar (HTTPS ou SSH)



e realizar o clonado do repositorio

```
$ git clone https://JesusAmieiro@bitbucket.org/JesusAmieiro/test-svn.git
```

A partir de aquí, o traballo enfócase como un repositorio Git normal, polo que o máis adecuado sería poñer o repositorio SVN e modo só lectura e realizar os commits en Git.

4 Referencias

<https://git-scm.com/book/es/v2/Git-and-Other-Systems-Git-as-a-Client>

<https://git-scm.com/book/es/v2/Git-and-Other-Systems-Migrating-to-Git>

<https://www.atlassian.com/git/tutorials/migrating-overview>

<https://github.com/danieleasoni/git-svn-example>

<https://dominikdorn.com/2016/05/how-to-recover-a-google-code-svn-project-and-migrate-to-github/>

<https://svn.eionet.europa.eu/>