

Mercurial, sistema de control de versiones

LABI Cursos

Martín Mello Teggia , Patricio Tula

labi.fi.uba.ar

10 de septiembre de 2015

Tabla de contenidos

- 1 **Introducción**
 - ¿Qué es Mercurial?
 - Repositorio y control de versiones
 - Bitbucket
- 2 **Actividad I: Bitbucket**
- 3 **Comandos básicos de hg**
- 4 **Actividad II: clonar repositorio y agregar archivo**
- 5 **Actividad III: Puller repositorio y hacerle modificaciones.**

¿Qué es Mercurial?



¿Qué es Mercurial?

Mercurial es un sistema de control de versiones multiplataforma, para desarrolladores de software. Está implementado principalmente haciendo uso del lenguaje de programación Python, pero incluye una implementación binaria de diff escrita en C. Mercurial fue escrito originalmente para funcionar sobre GNU/Linux pero ha sido adaptado para Windows, Mac OS X y la mayoría de otros sistemas tipo Unix.

Mercurial es, sobre todo, un programa para la línea de comandos. Todas las operaciones de Mercurial se invocan como opciones dadas a su programa motor, hg (cuyo nombre hace referencia al símbolo químico del mercurio).

Entre las principales metas de desarrollo de Mercurial se incluyen un gran rendimiento y escalabilidad. Es un desarrollo completamente distribuido, sin necesidad de un servidor central, provee una gestión robusta de archivos tanto de texto como binarios y capacidades avanzadas de ramificación e integración (todo ello manteniendo sencillez conceptual). También incluye una interfaz web integrada.

El creador y desarrollador principal de Mercurial es Matt Mackall. El código fuente se encuentra disponible bajo los términos de la licencia GNU GPL versión 2, lo que clasifica a Mercurial como software libre.

Repositorio y control de versiones

¿Qué es un repositorio?

Un repositorio, depósito o archivo es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.

¿Qué es un sistema de control de versiones?

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación.

¿Cual es la diferencia entre Mercurial y Dropbox?

- Mercurial permite el desarrollo distribuido de un proyecto mediante a *branches*.
- Mercurial provee un control de versiones robusto que permite volver a cualquier revisión subida al repositorio.
- Mercurial permite documentar cada uno de los cambios realizados en el repositorio.
- Mercurial ofrece un sistema de fusión de versiones mediante al programa *diff*.
- Dropbox ofrece una interfaz más amigable pero más limitada que la implementación básica de Mercurial.

¿Existen otros sistemas parecidos a Mercurial?

Sí existen, entre otros: CVS, Subversion, SourceSafe, ClearCase, Darcs, Bazaar, Plastic SCM, Git, Perforce, Fossil SCM.

Bitbucket

¿Qué es Bitbucket?

Bitbucket es un servicio de alojamiento web, para proyectos que utilizan el sistema de control de versiones Mercurial y Git.

¿Qué nos ofrece Bitbucket?

Bitbucket ofrece tanto planes comerciales como gratuitos para almacenar nuestros repositorios. Las cuentas gratuitas cuentan con un número ilimitado de repositorios privados, pero que solo pueden tener hasta 5 usuarios en cada uno. Si la cuenta de mail registrada es “@fi.uba.ar”, el número de colaboradores es ilimitado.

¿Dónde y cómo creo una cuenta?

El sitio web de Bitbucket es: bitbucket.org. Para crear una cuenta únicamente habrá que clicar en el botón *get started*, e ingresar los datos pedidos.

- 1 Crear cuenta en bitbucket.
- 2 Crear un repositorio de Mercurial.
- 3 Agregar un colaborador como administrador al repositorio.
- 4 Hacer un "README".
- 5 Mandar un "ISSUE" a algún colaborador.



Features Pricing Log in [Get started](#)

Your team, your code, connected

Git and Mercurial code management for teams

[Get started for free](#)

Free for small teams • Unlimited private repositories

More than 450,000 teams and 3 million developers love Bitbucket

salesforce

TESLA

El3n3rj3rk El3n3r

PayPal

DHL

clone, add, rm

Clonar una copia del repositorio: *clone*

Crea una copia de un repositorio existente en un nuevo directorio.

Ejecución: `hg clone <source>`

Agregar archivo al repositorio: *add*

Agrega al repositorio local los archivos `file_1` a `file_n` agregados.

Ejecución: `hg add <file_1> ... <file_n>`

Eliminar archivo del repositorio: *rm*

Elimina del repositorio local los archivos `file_1` a `file_n` agregados.

Ejecución: `hg rm <file_1> ... <file_n>`

Diagrama del repositorio 1



local
repo



working
copy

st, commit, push

Listar estado de modificaciones en el copia local: *st*

Lista de archivos modificados, borrados y agregados en la copia local del repositorio. Ejecución: hg st

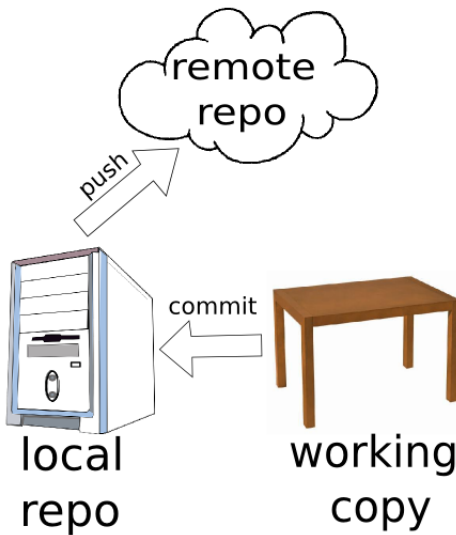
Guardar las modificaciones en el repositorio local: *commit*

Commit establece las modificaciones en los archivos dentro del repositorio local, generando una nueva revisión. Ejecución: hg commit -m 'Comments about changes'

Subir los cambios al repositorio original: *push*

Push sube las revisiones creadas sobre la copia local al repositorio original. Ejecución: hg push

Diagrama del repositorio 2



pull, update, merge

Descargar la copia original a la copia local: *pull*

Pull descarga el repositorio original nuestro repositorio local.

Ejecución: hg pull

Actualizar la copia local: *update*

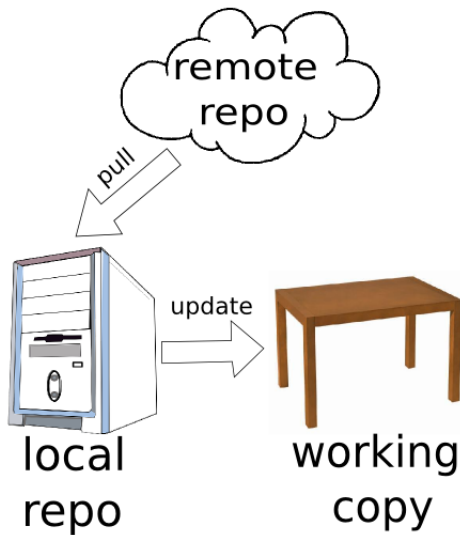
Una vez descargado el repositorio original, update actualiza nuestra copia local con su contenido. Ejecución: hg update

Fusionar copia local y copia original: *merge*

Si existen dos copias del mismo repositorio, con disntitas reivisiones, el comando merge se encargará de realizar una fusión de archivos y directorios para generar una copia unificada.

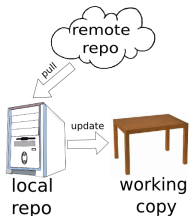
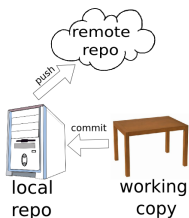
Ejecución: hg merge

Diagrama del repositorio 3



Actividad II: clonar repositorio y agregar archivo.

- 1 Clonar el repositorio creado.
- 2 Crear un archivo llamado `hola_repo_xxx.txt` con el siguiente texto "Hola Repo!".
- 3 Agregar el archivo al repo local.
- 4 Guardar los cambios del repo local.
- 5 Subir al repo original nuestra copia local.



Resolución actividad II

Comandos

```
$hg clone  
$echo Hola Repo! > hola_repo.txt  
$hg add hola_repo.txt  
$hg st  
$hg commit -m "subo archivo hola_repo.txt al repo."  
$hg push
```

Actividad III: Pullear repositorio y hacerle modificaciones.

- 1 Descargar repositorio original.
- 2 Modificar el archivo hola_repo.txt para que quede el siguiente texto: "Chau Repo!".
- 3 Guardar los cambios del repositorio local.
- 4 Subir al repo original nuestra copia local.

Resolución actividad II

Comandos

```
$hg pull
```

```
$hg update
```

```
$sed -i 's/Hola/Chau/g' hola_repo.txt
```

```
$hg commit -m 'Modifico archivo hola_repo.txt  
cambiando hola por chau.'
```

```
$hg push
```

Fin

Para la próxima

La siguiente clase veremos:

- Como hacer fusiones de copias.
- Como trabajar con ramas en un proyecto.