Mercurial, sistema de control de versiones

LABI Cursos

Martín Mello Teggia, Patricio Tula

labi.fi.uba.ar

16 de septiembre de 2015

Repaso + changüí

¿Qué es Mercurial?

Mercurial es un sistema de control del versiones con repositorio.

¿Qué comandos aprendimos a usar?

clone, add, rm, commit, pull, push, update, st y merge.

¿Cuántas veces hay que hacer clone en un repositorio?

UNA!

¿Qué comandos nos quedaron en el tintero?

mv, cp, commit --amend, revert y log.

Changüí 1

Mover un archivo: *mv*

Move (mv) permite mover archivos en el repositorio. Ejecución: hg mv <origin> <destiny>

Copiar un archivo: *cp*

Copy (cp) permite copiar archivos en el repositorio. Ejecución: hg cp <origin> <copy>

Editar commits: *commit --amend*

La opción *amend* del commando *commit* nos permite editar el mensaje del "commit" en nuestra copia local. Si el commit ya fue pusheado esto no podrá realizarse. Ejecución: hg commit --amend

Changüí 1

Probémoslo!

Changüí 2

Reviertir cambios realizados: *revert*

Volver a la última versión de un archivo del repo que modificamos.
Ejecución: hg revert <file>

Listar revisiones: *log*

log lista todas las revisiones del repositorio.
Ejecución: hg log -l <n>

Changüí 2

Probémoslo!

Configuración local

¿Como configuramos a mercurial?

Para configurar hay que editar el archivo HOME/.hgrc (HOME se refiere a /home/username) con los parametros de configuración que deseamos.

¿Qué parametros podemos configurar?

En este archivo configuraremos el usuario, alguna extensiones, parametros de conexión y la configuración de colores en los estados de los archivos del repositorio.

¿Donde encontrar un ejemplo?

En el siguiente sitio pueden encontrar un ejemplo: http://labi.fi.uba.ar/chiliproject/projects/club-robotica/wiki/ABC_de_Mercurial.

Configuración local

.hgrc

```
[ui]
username = Firstname Lastname <mail@mailinator.com>
verbose = True
[extensions]
fetch =
color =
[hostfingerprints]
labi.fi.uba.ar = xxxx
[color]
status.modified = magenta bold
status.added = green bold
status.removed = red bold
status.deleted = cyan bold
status.unknown = blue bold
status.ignored = black bold
```

Configuración extensiones ignoraradas

.hgignore

```
syntax: glob
*.aux
*.log
*.toc
*.pdf
*.dvi
*~
*.swp
```

Configuraciones

Configurémoslo!

Kdiff3

¿Qué herramienta vamos a utilizar?

Vamos a utilizar un programa unificador de versiones cuyo objetivo es darnos una interfaz gráfica para la tarea de unir revisiones.

¿Cómo funciona Kdiff3?

Este programa recibe 3 versiones de la revisión a mergear. Estos serán llamados *buffers de entrada*. Entre estos 3 tendremos que elegir con que versión quedarnos. Al finalizar el proceso de unión de versiones se generará un archivo único con los cambios que le hemos realizado. Por último, habrá que realizar un "commit" explicando brevemente el criterio tomado para unificar.

Merge

Probémoslo!

Fin