

Tema: II git Pràctiques.

Herramientas Avanzadas para el Desarrollo de Aplicaciones

Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant

Curs 2014-2015 , Copyleft © 2011-2015 .

Reproducció permesa sota els termes de la llicència de documentació
lliure GNU.

Contingut

- 1 Primers passos amb git
- 2 Modificació d'arxius (I)
- 3 Modificació d'arxius (II)
- 4 Afegit d'arxius
- 5 Log del repositori
- 6 Esborrat d'arxius
- 7 Desfer accions (I)
- 8 Desfer accions (II)
- 9 Renombrar fitxers
- 10 Branques: Actual, crear, canviar de branca, diferències
- 11 Branques: Deferències a nivell de commits
- 12 Exercici amb branques
- 13 Branques: barrejar, depassar
- 14 Clonat de repositoris
- 15 Objectius...
- 16 Lliurament...

Primers passos amb git

```
1  # Comproveu la version instal·lada
   git --version
3
   # Partim del codigo de la practica 1 del joc d'
   #   endevinar un numero
5  # Creguem el repositori inicial per posar sota
   #   control de versions
   # els arxius d'aquest projecte.
7  cd Juego; git init

9  # Anyadimos al repositori els arxius del directori
   #   actual i subdirectoris
   git add .
11 # Comprovem l'estat despues de anyadirlos
   git status
13 # Confirmem que els hem anyadido (commit)
   git commit -m 'Primer commit.' -m "Descripcion
   #   detallada." -a
```

Modificació d'arxius (I)

```
2      # Modifiquem algun arxiu
      gedit main.vala Makefile

4      # Veiem les diferències de la còpia de treball amb
      el repositori
      git diff

6      # acceptem els canvis, podem fer-ho de diverses
      maneres:

8      # 1- Arxiu a arxiu
10     git commit -m "Correccion de errores en el prog.
      ppal." main.vala
      git commit -m "Anyadido objetivo de depuracion."
      Makefile
```

Modificació d'arxius (II)

```
1  # 2- Preparem el 'escenari' o 'stage' que formes el  
    commit  
    # A  
3  git add Makefile  
    git commit -m "Modificado Makefile."  
5  # B  
    git add Makefile main.vala  
7  git commit -m "Un solo commit con varios archivos."
```

Afegit d'arxius

```
1  # Creguem l'arxiu
   gedit autores.txt
3
   # El anyadimos...
5  git add autores.txt

7  # Confirmem el anyadido...
   git commit -m "fichero autores.txt" autores.txt
9
   # Que arxius hi ha baix control de versions?
11 git ls-files
```

Log del repositori

```
1  # Que operacions hem fet sobre el repositori?
   git log
3
   commit dbafcc771eb504a541db32d7a7b5287a470516aa    # HEAD
5   Author: hada <hada@dlsi.ua.es>
   Date:   Fri Jan 20 18:26:22 2012 +0100
7
   fichero autores.txt
9
   commit 18442c5369b1f6f7920a4ebea39ec38bf9b62555    # HEAD~1
11  Author: hada <hada@dlsi.ua.es>
   Date:   Fri Jan 20 17:54:48 2012 +0100
13
   Canvis al makefile.
15
   commit e3911a1778c405ecce14bae1c1a97ec81832242b    # HEAD~2
17  Author: hada <hada@dlsi.ua.es>
   Date:   Fri Jan 20 17:54:12 2012 +0100
19
   # Cada commit te el seu numero SHA-1 propi
21  # Els numeros SHA-1 es poden resumir fins que siguin distingibles
   # Podem referir-nos a l'ultim commit fet (el primer de la llista) per l'alies:
   HEAD
23
   # Etiquetem HEAD amb v1.0 i HEAD1 amb v0.9
25  git tag v1.0
   git tag v0.9 HEAD~1
27  git tag -l
   # Ho visualitzem graficament amb gitk
29  gitk
```

Esborrat d'arxius

```
1  # Esborrem l'arxiu
   git rm autores.txt
3
   # Veiem l'estat del repositori
5  git status

7  # Confirmem l'esborrat...
   git commit -m "fichero autores.txt borrado"
9
   # Que diu el log del repositori?
11 git log

13 commit 5a369acdefc3c1d28c7d1c9561f7bb26d9daead3
   Author: hada <hada@dlsi.ua.es>
15 Date:   Fri Jan 20 18:41:56 2012 +0100

17 Archivo autores.txt borrado.
```


Desfer accions (I)

```
# Esborrem l'arxiu
2  git rm autores.txt
  git commit ...

4  # NOOO!!! es una errada!!!, ho puc recuperar?

6  # Que diu el log del repositori?
8  git log

10 commit 5a369acdefc3c1d28c7d1c9561f7bb26d9daead3 # HEAD
   Author: hada <hada@dlsi.ua.es>
12   Date:   Fri Jan 20 18:41:56 2012 +0100

14   Archivo autores.txt borrado.

16 commit dbafcc771eb504a541db32d7a7b5287a470516aa # HEAD~1
   Author: hada <hada@dlsi.ua.es>
18   Date:   Fri Jan 20 18:26:22 2012 +0100

20   Correccio d'errors.
```

Desfer accions (II)

Hi ha diverses formes de fer-ho, usarem 'git revert':

```
1  # revert desfa un commit creant un commit 'invers'
   # amb la opcion '-n' fa tot menys crear el commit 'invers'
3  # ho fem así per veure pas a pas l'estat del repositori
   git revert -n HEAD
5
   git status
7
   # On branch master
   # Changes to be committed:
   #   (use "git reset HEAD <file>..." to unstage)
11  #
   #       new file:   autores.txt
13  #
15  # Que diu ara el log del repositori?
   git log
17
   commit 5a369acdefc3c1d28c7d1c9561f7bb26d9daead3    # HEAD
19  Author: hada <hada@dlsi.ua.es>
   Date:   Fri Jan 20 18:41:56 2012 +0100
21
   Archivo autores.txt borrado.
23
   # PREGUNTA: Que hauria que fer ara?
25
   # Una altra forma de fer-ho: git reset
27  # Investiga que fa i tracta d'usar-ho per resoldre aquesta situaci.
```

Renombrar fitxers

```
1  git mv autores.txt AUTHORS
   git status
3  # On branch master
   # Changes to be committed:
5  #   (use "git reset HEAD <file >..." to unstage)
   #
7  #       renamed:    autores.txt -> AUTHORS
9  # PREGUNTA: Que hauria de fer ara?
11 # Solament un commit que abasti l'operacio de renombrat
   git commit -m "Renombrado archivo autores.txt a AUTHORS."
```

Branques: Actual, crear, canviar de branca, diferències

```
2  # Comprovar l'actual, remotes o totes
   git branch [-r] [-a]
   * master
4
   # Crear una branca anomenada 'devel' basada en l'actual
6   git branch devel
   # Branques existents, en la qual estem porta un ''
8   git branch
   devel
10  * master
   # Canviar a 'devel'
12  git checkout devel
   Switched to branch 'devel'
14
   # Comprova que aquestes en la branca 'devel'
16  # Fes canvis en AUTHORS aci i guardals (commit)
   # canviat de nou a 'master'
18  git checkout master
   Switched to branch 'master'
20
   # Diferències entre 'devel' i 'master' (estem en master)
22  git diff devel
   # o tambien
24  git diff master devel
   git diff devel master
```

Branques: Deferències a nivell de commits.

Ho fem amb 'git log [-p]'

```
1  # Podem veure-les de diverses maneres
   git log master..devel # branca origen es master, branca destino es devel
3  git log devel..master
   git log master..      # branca origen es master, branca destino l'actual
5  git log ..master      # branca origen l'actual, branca destino master

7  # tambe podem emprar show-branch, p.e.:
   git show-branch master devel
9  ! [master] Renombrat arxiu.
   * [devel] Canvis en README.
11 —
   * [devel] Canvis en README.
13 +* [master] Renombrat arxiu.
```

- A hores d'ara has d'haver creat la branca 'devel'.
- Has d'estar treballant en ella i no en 'master'.
- Modifica els arxius necessaris perquè en el joc es permeti un nombre màxim d'intents per endevinar el nombre, per exemple 3.
- En finalitzar aquesta part has de tenir tots els canvis incorporats a la branca 'devel' (haver fet l'o els commits necessaris)

Branques: barrejar, depassar

- Suposem que volem passar els nostres canvis de la branca 'devel' a 'master'.
- Tenim dues opcions: **barrejar-les** o **rebassa-les**

```
1  # Opcio Barreja
   git merge devel
3  Updating ddeebdc..646eba9
   Fast-forward
5  README |      1 +
   1 files changed, 1 insertions(+), 0 deletions(-)
7
   # Opcio Rebase
9  git rebase devel
   First, rewinding head to replay your work on top of it...
11 Fast-forwarded master to devel.
```

Clonat de repositoris

- Ens permet 'copiar' un repositori local o remotament (http/s, ssh, git, git+ssh).
- S'estableix un enllaç entre tots dos que permet realitzar operacions **pull**, **fetch**, **merge** i **push**.

```
1  # clona el repositori amb el codi de la practical
   # ho fem en un directori anomenat practicalb
3  git clone practical practicalb
   Cloning into 'practicalb'...
5  done.

7  cd practicalb; ls  # debe estar todo copia-trabajo + repositorio (.git)

9  # Si intentem fer un push obtindrem un error. El repositori origen
   # conte copia de treball (no es bare -sol directori .git-)
```

El clonat de repositoris és molt util quan es clona un repositori sol amb dades (bare) i no amb còpia de treball.

Objectius...

L'alumne sap:

- ☐ Crear un repositori amb git i afegir els arxius que posarà baix control de versions.
- ☐ Fer commits de les accions que porti acabo (modificacions en arxius, afegit de nous arxius, esborrat d'arxius, renombrado d'arxius, etc. . .)
- ☐ Veure el 'log' de les accions dutes a terme.
- ☐ Posar etiquetes a una determinada versió dels arxius.
- ☐ Desfer accions, p.i. recuperar un arxiu esborrat.
- ☐ Crear branques, canviar de branca, veure diferències entre les branques.
- ☐ Importar canvis d'una branca a una altra.
- ☐ Clonar repositoris.

- El lliurament d'aquesta pràctica consisteix en el directori de treball (Juego) comprimit en un fitxer anomenat `juego.tgz`.
- Recorda que aquest directori ja conté la còpia de treball i el directori `' .git '`.