

24_referencias

May 6, 2020

1 Referencias.

1.1 Preliminares.

Antes de empezar este capítulo es necesario descomprimir el archivo `src/prueba.zip`, el cual contiene una estructura similar al repositorio `demo` de capítulos previos.

```
[1]: rm -rf prueba
```

```
[2]: unzip src/prueba.zip
```

```
Archive:  src/prueba.zip
  creating: prueba/
 extracting: prueba/archivo-2
 extracting: prueba/archivo-1
   creating: prueba/.git/
 inflating: prueba/.git/index
   creating: prueba/.git/refs/
   creating: prueba/.git/refs/heads/
 inflating: prueba/.git/refs/heads/master
 extracting: prueba/.git/refs/heads/restituida
 extracting: prueba/.git/refs/heads/nueva
   creating: prueba/.git/refs/tags/
 inflating: prueba/.git/description
   creating: prueba/.git/info/
 inflating: prueba/.git/info/exclude
 extracting: prueba/.git/ORIG_HEAD
 extracting: prueba/.git/COMMIT_EDITMSG
   creating: prueba/.git/objects/
   creating: prueba/.git/objects/90/
 extracting: prueba/.git/objects/90/a8b80bbb845cb1187852777769937a9574e3ab
   creating: prueba/.git/objects/ed/
 extracting: prueba/.git/objects/ed/7d117fc8970b297653c23ebc41936cc4ab4472
   creating: prueba/.git/objects/pack/
   creating: prueba/.git/objects/cb/
 extracting: prueba/.git/objects/cb/0b6bb3a84bc6fcd0f6a0f87909bf68f298e21a
   creating: prueba/.git/objects/2c/
 extracting: prueba/.git/objects/2c/b76ef80c41c0037c6a82092737cba6061c7083
   creating: prueba/.git/objects/info/
```

```

creating: prueba/.git/objects/e6/
extracting: prueba/.git/objects/e6/9de29bb2d1d6434b8b29ae775ad8c2e48c5391
creating: prueba/.git/objects/a1/
extracting: prueba/.git/objects/a1/9abfea0f29b668c91c58c834b8965e6c37804f
creating: prueba/.git/objects/dc/
extracting: prueba/.git/objects/dc/d3ac8c60d63186964b45f87cc4416e6caefa5f
creating: prueba/.git/objects/78/
extracting: prueba/.git/objects/78/8564428c6076630b22efc89ad20c5f63ee9bf1
creating: prueba/.git/objects/ba/
extracting: prueba/.git/objects/ba/397e8f7c51f8dfdefa12250933811042df91a2
creating: prueba/.git/objects/ff/
extracting: prueba/.git/objects/ff/7ede1f3ce0201ae532684bbbee157247c3a88b7
creating: prueba/.git/objects/ac/
extracting: prueba/.git/objects/ac/9103394fb02ee49adb4c05e2125174707fbb7d
creating: prueba/.git/objects/1f/
extracting: prueba/.git/objects/1f/1f5adb5e6ff2e2ff651e352f21df0417defb2
creating: prueba/.git/objects/08/
extracting: prueba/.git/objects/08/34a0b252e02bc28bae34588e01f410ad771884
creating: prueba/.git/objects/d7/
extracting: prueba/.git/objects/d7/f0e7ba5cf079cdea1d7c8792e1432161f082c1
creating: prueba/.git/objects/46/
extracting: prueba/.git/objects/46/c8014e29a665bd0e883c9008ad4412f8933809
creating: prueba/.git/objects/df/
extracting: prueba/.git/objects/df/8a330f084618bdb823d99a6997d7be4b9efd15
creating: prueba/.git/objects/b3/
extracting: prueba/.git/objects/b3/890082fec489a5ecff76db4fd36cc10d4bc8d6
creating: prueba/.git/branches/
creating: prueba/.git/logs/
creating: prueba/.git/logs/refs/
creating: prueba/.git/logs/refs/heads/
inflating: prueba/.git/logs/refs/heads/master
inflating: prueba/.git/logs/refs/heads/restituida
inflating: prueba/.git/logs/refs/heads/nueva
inflating: prueba/.git/logs/HEAD
extracting: prueba/.git/HEAD
inflating: prueba/.git/config
creating: prueba/.git/hooks/
inflating: prueba/.git/hooks/post-update.sample
inflating: prueba/.git/hooks/commit-msg.sample
inflating: prueba/.git/hooks/pre-receive.sample
inflating: prueba/.git/hooks/update.sample
inflating: prueba/.git/hooks/pre-push.sample
inflating: prueba/.git/hooks/fsmonitor-watchman.sample
inflating: prueba/.git/hooks/prepare-commit-msg.sample
inflating: prueba/.git/hooks/pre-applypatch.sample
inflating: prueba/.git/hooks/pre-rebase.sample
inflating: prueba/.git/hooks/applypatch-msg.sample
inflating: prueba/.git/hooks/pre-commit.sample

```

```
extracting: prueba/archivo_nuevo
extracting: prueba/invisible
extracting: prueba/.gitignore
```

```
[3]: cd prueba
```

```
[4]: git log --oneline
```

```
ff7ede1 (HEAD -> master) commit
fusionado
0834a0b quinto commit
46c8014 (nueva) primer commit de la rama nueva
ba397e8 (restituida) cuarto commit
cb0b6bb segundo commit
ed7d117 primer commit
```

```
[5]: git branch -l
```

```
* master
  nueva
  restituida
```

```
[6]: ls
```

```
archivo-1  archivo-2  archivo_nuevo  invisible
```

```
[7]: git ls-files
```

```
.gitignore
archivo-1
archivo-2
archivo_nuevo
```

1.2 El almacén de datos de *Git*.

Git utiliza un “almacén de datos” que relaciona identificadores con objetos específicos.

Estos objetos corresponden a:

- Archivos.
- Ramas.
- Versiones.

Cada objeto es relacionado a un *hash* único que será usado como identificador.

Los hashes son difíciles de memorizar y es por ello por lo que se han desarrollado esquemas de referencias y asignación de etiquetas.

<https://git-scm.com/book/es/v2/Los-entresijos-internos-de-Git-Referencias-Git>

1.3 Referencias.

Las referencias se encuentran en el directorio `.git/refs` en una estructura con los directorios: `* .git/refs/heads`, el cual contienen las referencias a las “cabezas” de cada rama del repositorio. Las “cabezas” corresponden al commit más reciente en el que se encuentra una rama. `* git/refs/tags`, el cual contiene las referencias a etiquetas asignadas a ciertos objetos de Git.

Ejemplo:

- La siguiente celda mostrará la estructura del directorio `.git/refs` del repositorio en el que se encuentra esta notebook.
- El resultado para el repositorio `prueba` sería algo similar a:

```
.git/refs
  heads
    master
    nueva
    restituida
  tags
```

2 directories, 3 files

```
[8]: tree .git/refs
```

```
.git/refs
  heads
    master
    nueva
    restituida
  tags
```

2 directories, 3 files

- Como se puede apreciar, el directorio `.git/refs/heads/` contiene un archivo con el nombre de cada rama del directorio. Cada uno de estos archivos contiene el identificador del commit más reciente de cada rama.

Ejemplo:

- La siguiente celda regresará el contenido del archivo `.git/refs/heads/restituida`. El cual corresponde al identificador del commit con descripción `cuarto commit`.

```
[9]: cat .git/refs/heads/restituida
```

```
ba397e8f7c51f8dfdefa12250933811042df91a2
```

- La siguiente celda aplicará el comando `git show` al contenido de `.git/refs/heads/restituida`.

```
[10]: cat .git/refs/heads/restituida | xargs git show
```

```
commit ba397e8f7c51f8dfdefa12250933811042df91a2
```

```
(restituida)
```

```
Author: Jose Luis Chiquete <josech@gmail.com>
```

```
Date: Tue Dec 17 12:39:12 2019 -0600
```

cuarto commit

```
diff --git a/archivo-1 b/archivo-1
```

```
index e69de29..a19abfe 100644
```

```
--- a/archivo-1
```

```
+++ b/archivo-1
```

```
@@ -0,0 +1 @@
```

```
+Hola
```

- La siguiente celda aplicará el comando `git show` al contenido de `.git/refs/heads/master`.

```
[11]: cat .git/refs/heads/master | xargs git show
```

```
commit ff7ede1f3ce0201ae532684bbee157247c3a88b7 (HEAD ->
```

```
master)
```

```
Merge: 0834a0b 46c8014
```

```
Author: Jose Luis Chiquete <josech@gmail.com>
```

```
Date: Tue Dec 17 12:46:44 2019 -0600
```

commit fusionado

- La siguiente celda aplicará el comando `git show` al contenido de `.git/refs/heads/nuevaa`.

```
[12]: cat .git/refs/heads/nueva | xargs git show
```

```
commit 46c8014e29a665bd0e883c9008ad4412f8933809
```

```
(nueva)
```

```
Author: Jose Luis Chiquete <josech@gmail.com>
```

```
Date: Tue Dec 17 12:44:46 2019 -0600
```

primer commit de la rama nueva

```
diff --git a/archivo_nuevo b/archivo_nuevo
```

```
new file mode 100644
```

```
index 0000000..df8a330
```

```
--- /dev/null
```

```
+++ b/archivo_nuevo
```

```
@@ -0,0 +1 @@
```

```
+Archivo de la rama nueva.
```

1.3.1 La variable HEAD.

Git le asigna a la variable **HEAD** el valor de la cabeza de la rama en la que se encuentra un repositorio.

Ejemplo:

- El comando `git show` utilizado sin más argumentos devuelve la información de la posición en la que se encuentra actualmente el repositorio.

```
[13]: git show
```

```
commit ff7ede1f3ce0201ae532684bbee157247c3a88b7 (HEAD ->
master)
Merge: 0834a0b 46c8014
Author: Jose Luis Chiquete <josech@gmail.com>
Date: Tue Dec 17 12:46:44 2019 -0600

    commit fusionado
```

- La siguiente celda regresará el valor de **HEAD**.

```
[14]: git show HEAD
```

```
commit ff7ede1f3ce0201ae532684bbee157247c3a88b7 (HEAD ->
master)
Merge: 0834a0b 46c8014
Author: Jose Luis Chiquete <josech@gmail.com>
Date: Tue Dec 17 12:46:44 2019 -0600

    commit fusionado
```

1.3.2 El comando `git reflog`.

Este comando regresa la bitácora de todos los commits realizados desde cualquier rama, usando **HEAD** como referencia.

```
[15]: git reflog
```

```
ff7ede1 (HEAD -> master) HEAD@{0}:
checkout: moving from restituida to master
ba397e8 (restituida) HEAD@{1}: checkout:
moving from ba397e8f7c51f8dfdefa12250933811042df91a2 to restituida
ba397e8 (restituida) HEAD@{2}: checkout:
moving from master to ba397e8
ff7ede1 (HEAD -> master) HEAD@{3}:
merge nueva: Merge made by the 'recursive' strategy.
0834a0b HEAD@{4}: commit: quinto commit
ba397e8 (restituida) HEAD@{5}: checkout:
```

```

moving from nueva to master
46c8014 (nueva) HEAD@{6}: commit: primer
commit de la rama nueva
ba397e8 (restituida) HEAD@{7}: checkout:
moving from master to nueva
ba397e8 (restituida) HEAD@{8}: commit: cuarto
commit
cb0b6bb HEAD@{9}: commit: segundo commit
ed7d117 HEAD@{10}: commit (initial): primer commit

```

1.4 Operaciones con HEAD.

Es posible utilizar ciertos operadores con HEAD.

- @{n} para identificar a un commit de cualquier rama.
- ~ para regresar en la rama.
- ^ para ir a la rama de origen.

1.4.1 Operador de referencia general.

Este operador hace referencia a la secuencia lineal de commits tal como se enumeran con `git reflog`.

`HEAD@{<n>}`.

Donde: * <n> es un número que va de 0 para el commit mas reciente y va regresando de uno en uno.

Ejemplos:

```
[16]: git branch -l
```

```

* master
  nueva
  restituida

```

```
[17]: git show
```

```

commit ff7ede1f3ce0201ae532684bbee157247c3a88b7 (HEAD ->
master)
Merge: 0834a0b 46c8014
Author: Jose Luis Chiquete <josech@gmail.com>
Date: Tue Dec 17 12:46:44 2019 -0600

    commit fusionado

```

```
[18]: git show HEAD@{8} --oneline
```

```
ba397e8 (restituida) cuarto commit
diff --git a/archivo-1 b/archivo-1
index e69de29..a19abfe 100644
--- a/archivo-1
+++ b/archivo-1
@@ -0,0 +1 @@
+Hola
```

[19]: `git show HEAD@{3} --oneline`

```
ff7ede1 (HEAD -> master) commit
fusionado
```

[20]: `git diff HEAD@{8} HEAD`

```
diff --git a/archivo-1 b/archivo-1
index a19abfe..d7f0e7b 100644
--- a/archivo-1
+++ b/archivo-1
@@ -1 +1,2 @@
  Hola
+Otra linea
diff --git a/archivo_nuevo b/archivo_nuevo
new file mode 100644
index 0000000..df8a330
--- /dev/null
+++ b/archivo_nuevo
@@ -0,0 +1 @@
+Archivo de la rama nueva.
```

1.4.2 Operador de referencia dentro de la rama.

Este operador se usa para regresar en los commits de una rama.

HEAD~<n>.

- <n> es un número que va de 1 para el commit mas reciente y va regresando de uno en uno. El valor por defecto es 1.

[21]: `git show HEAD~ --oneline`

```
0834a0b quinto commit
diff --git a/archivo-1 b/archivo-1
index a19abfe..d7f0e7b 100644
--- a/archivo-1
+++ b/archivo-1
@@ -1 +1,2 @@
  Hola
+Otra linea
```



```
[22]: git show HEAD~1 --oneline
```

```
0834a0b quinto commit
diff --git a/archivo-1 b/archivo-1
index a19abfe..d7f0e7b 100644
--- a/archivo-1
+++ b/archivo-1
@@ -1,2 @@
    Hola
+Otra linea
```

```
[23]: git show HEAD~4 --oneline
```

```
ed7d117 primer commit
diff --git a/.gitignore b/.gitignore
new file mode 100644
index 0000000..90a8b80
--- /dev/null
+++ b/.gitignore
@@ -0,0 +1 @@
+invisible
diff --git a/archivo-1 b/archivo-1
new file mode 100644
index 0000000..e69de29
diff --git a/archivo-2 b/archivo-2
new file mode 100644
index 0000000..e69de29
diff --git a/archivo-3 b/archivo-3
new file mode 100644
index 0000000..e69de29
```

1.4.3 Operador de referencia de la rama de origen.

Este operador se usa para regresar en los commits de una rama que dio origen a la actual.

HEAD~<n>.

- <n> es un número que va de 1 para el commit mas reciente y va regresando de uno en uno. El valor por defecto es 1.

```
[24]: git show HEAD^ --oneline
```

```
0834a0b quinto commit
diff --git a/archivo-1 b/archivo-1
index a19abfe..d7f0e7b 100644
--- a/archivo-1
+++ b/archivo-1
@@ -1,2 @@
    Hola
```

+Otra linea

```
[25]: git show HEAD^2 --oneline
```

```
46c8014 (nueva) primer commit de la rama nueva
diff --git a/archivo_nuevo b/archivo_nuevo
new file mode 100644
index 0000000..df8a330
--- /dev/null
+++ b/archivo_nuevo
@@ -0,0 +1 @@
+Archivo de la rama nueva.
```

Esta obra está bajo una Licencia Creative Commons Atribución 4.0 Internacional.

© José Luis Chiquete Valdivieso. 2020.