

Free Online CV Maker

Fast & Easy Resume in Minutes

Boost your Chances of Getting Interviews by 70% and Beat the Competition.

resumecoach.com

OPEN

Pro Git, el libro oficial de Git

2.6. Creando etiquetas

Como muchos VCSs, Git tiene la habilidad de etiquetar (*tag*) puntos específicos en la historia como importantes. Generalmente la gente usa esta funcionalidad para marcar puntos donde se ha lanzado alguna versión (v1.0, y así sucesivamente). En esta sección aprenderás cómo listar las etiquetas disponibles, crear nuevas etiquetas y qué tipos diferentes de etiquetas hay.

2.6.1. Listando tus etiquetas

Listar las etiquetas disponibles en Git es sencillo, Simplemente escribe `git tag`:

```
$ git tag
v0.1
v1.3
```

Este comando lista las etiquetas en orden alfabético; el orden en el que aparecen no es realmente importante.

También puedes buscar etiquetas de acuerdo a un patrón en particular. El repositorio fuente de Git, por ejemplo, contiene mas de 240 etiquetas. Si solo estas interesado en la serie 1.4.2, puedes ejecutar esto:

Free Resume Templates (2022)

Choose from 100s of Templates

Boost your Chances of Getting Interviews by 70% and Beat the Competition.

resumecoach.com

OPEN

```
$ git tag -l 'v1.4.2.*'
v1.4.2.1
v1.4.2.2
v1.4.2.3
v1.4.2.4
```

2.6.2. Creando etiquetas

Git usa dos tipos principales de etiquetas: ligeras y anotadas. Una etiqueta ligera es muy parecida a una rama que no cambia — un puntero a una confirmación específica —. Sin embargo, las etiquetas anotadas son almacenadas como objetos completos en la base de datos de Git. Tienen suma de comprobación; contienen el nombre del etiquetador, correo electrónico y fecha; tienen mensaje de etiquetado; y pueden estar firmadas y verificadas con GNU Privacy Guard (GPG). Generalmente se recomienda crear etiquetas anotadas para disponer de toda esta información; pero si por alguna razón quieres una etiqueta temporal y no quieres almacenar el resto de información, también tiene disponibles las etiquetas ligeras.

Indice de contenidos

1. Empezando

Capítulo 2. Fundamentos de Git

2.1. Obteniendo un repositorio Git

2.2. Guardando cambios en el repositorio

2.3. Viendo el histórico de confirmaciones

2.4. Deshaciendo cosas

2.5. Trabajando con repositorios remotos

2.6. Creando etiquetas

2.7. Consejos y trucos

2.8. Resumen

3. Trabajando con ramas en Git

4. Git en un servidor

5. Git en entornos distribuidos

6. Las herramientas de Git

7. Personalizando Git

8. Git y otros sistemas

9. Funcionamiento interno de Git

2.6.3. Etiquetas anotadas

Crear una etiqueta anotada en Git es simple. La forma más fácil es especificar `-a` al ejecutar el comando `tag`:

```
$ git tag -a v1.4 -m 'my version 1.4'
$ git tag
v0.1
v1.3
v1.4
```



El parámetro `-m` especifica el mensaje, el cual se almacena con la etiqueta. Si no se especifica un mensaje para la etiqueta anotada, Git lanza tu editor para poder escribirlo.

Puedes ver los datos de la etiqueta junto con la confirmación que fue etiquetada usando el comando `git show`:

```
$ git show v1.4
tag v1.4
Tagger: Scott Chacon <schacon@gee-mail.com>
Date: Mon Feb 9 14:45:11 2009 -0800

my version 1.4
commit 15027957951b64cf874c3557a0f3547bd83b3ff6
Merge: 4a447f7... a6b4c97...
Author: Scott Chacon <schacon@gee-mail.com>
Date: Sun Feb 8 19:02:46 2009 -0800

Merge branch 'experiment'
```

Esto muestra la información del autor de la etiqueta, la fecha en la que la confirmación fue etiquetada, y el mensaje de anotación antes de mostrar la información de la confirmación.

2.6.4. Etiquetas firmadas

También puedes firmar tus etiquetas con GPG, siempre que tengas una clave privada. Lo único que debes hacer es usar `-s` en vez de `-a`:

```
$ git tag -s v1.5 -m 'my signed 1.5 tag'
You need a passphrase to unlock the secret key for
user: "Scott Chacon <schacon@gee-mail.com>"
1024-bit DSA key, ID F721C45A, created 2009-02-09
```

Si ejecutas `git show` en esa etiqueta, puedes ver la firma GPG adjunta a ella:

```
$ git show v1.5
tag v1.5
Tagger: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Feb 9 15:22:20 2009 -0800

my signed 1.5 tag
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.8 (Darwin)

iEYEA BECAAYFAkmQurIACgkQON3DxfchxFr5cACeIMN+ZxLKggJQf0QYiQBwgySN
Ki0An2JeAVUCAiJ70x6ZEtK+NvZAJ82/
=WryJ
-----END PGP SIGNATURE-----
commit 15027957951b64cf874c3557a0f3547bd83b3ff6
Merge: 4a447f7... a6b4c97...
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sun Feb 8 19:02:46 2009 -0800

    Merge branch 'experiment'
```

Más tarde, aprenderás cómo verificar etiquetas firmadas.

2.6.5. Etiquetas ligeras

Otra forma de etiquetar confirmaciones es con una etiqueta ligera. Esto es básicamente la suma de comprobación de la confirmación almacenada en un archivo — ninguna otra información es guardada —. Para crear una etiqueta ligera no añadidas las opciones `-a`, `-s` o `-m`:

```
$ git tag v1.4-lw
$ git tag
v0.1
v1.3
v1.4
v1.4-lw
v1.5
```

Esta vez, si ejecutas el comando `git show` en la etiqueta, no verás ninguna información extra. El comando simplemente muestra la confirmación.

```
$ git show v1.4-lw
commit 15027957951b64cf874c3557a0f3547bd83b3ff6
Merge: 4a447f7... a6b4c97...
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sun Feb 8 19:02:46 2009 -0800

    Merge branch 'experiment'
```

2.6.6. Verificando etiquetas

Para verificar una etiqueta firmada, debes usar `git tag -v [tag-name]`. Este comando utiliza GPG para verificar la firma. Necesitas la clave pública del autor de la firma en tu llavero para que funcione correctamente.

```
$ git tag -v v1.4.2.1
object 883653bad8ee7ea23e6a5c392bb739348b1eb61
type commit
tag v1.4.2.1
tagger Junio C Hamano <junkio@cox.net> 1158138501 -0700

GIT 1.4.2.1

Minor fixes since 1.4.2, including git-mv and git-http with alternates.
gpg: Signature made Wed Sep 13 02:08:25 2006 PDT using DSA key ID F3119B9A
gpg: Good signature from "Junio C Hamano <junkio@cox.net>"
gpg:          aka "[jpeg image of size 1513]"
Primary key fingerprint: 3565 2A26 2040 E066 C9A7  4A7D C0C6 D9A4 F311 9B9A
```

Si no tienes la clave pública del autor de la firma, se obtiene algo parecido a:

```
gpg: Signature made Wed Sep 13 02:08:25 2006 PDT using DSA key ID F3119B9A
gpg: Can't check signature: public key not found
error: could not verify the tag 'v1.4.2.1'
```

2.6.7. Etiquetando más tarde

Puedes incluso etiquetar confirmaciones después de avanzar sobre ellos. Supón que tu historico de confirmaciones se parece a esto:

```
$ git log --pretty=oneline
15027957951b64cf874c3557a0f3547bd83b3ff6 Merge branch 'experiment'
a6b4c97498bd301d84096da251c98a07c7723e65 beginning write support
0d52aaab4479697da7686c15f77a3d64d9165190 one more thing
6d52a271eda8725415634dd79daabbc4d9b6008e Merge branch 'experiment'
0b7434d86859cc7b8c3d5e1dddfed66ff742fcfc added a commit function
4682c3261057305bdd616e23b64b0857d832627b added a todo file
166ae0c4d3f420721acbb115cc33848dfcc2121a started write support
9fceb02d0ae598e95dc970b74767f19372d61af8 updated rakefile
964f16d36dfccde844893cac5b347e7b3d44abbc commit the todo
8a5cbc430f1a9c3d00faaeffd07798508422908a updated readme
```

Ahora, supón que olvidaste etiquetar el proyecto en v1.2, que estaba en la confirmación "*updated rakefile*". Puedes hacerlo ahora. Para etiquetar esa confirmación especifica la suma de comprobación de la confirmación (o una parte de la misma) al final del comando:

```
$ git tag -a v1.2 9fceb02
```

Puedes ver que has etiquetado la confirmación:

```
$ git tag
v0.1
v1.2
v1.3
v1.4
v1.4-lw
v1.5

$ git show v1.2
tag v1.2
Tagger: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Feb 9 15:32:16 2009 -0800

version 1.2
commit 9fceb02d0ae598e95dc970b74767f19372d61af8
Author: Magnus Chacon <mchacon@gee-mail.com>
Date:   Sun Apr 27 20:43:35 2008 -0700

    updated rakefile
...
```

2.6.8. Compartiendo etiquetas

Por defecto, el comando `git push` no transfiere etiquetas a servidores remotos. Tienes que enviarlas explícitamente a un servidor compartido después de haberlas creado. Este proceso es igual a compartir ramas remotas, por lo que puedes ejecutar `git push origin [tagname]`.

```
$ git push origin v1.5
Counting objects: 50, done.
Compressing objects: 100% (38/38), done.
Writing objects: 100% (44/44), 4.56 KiB, done.
Total 44 (delta 18), reused 8 (delta 1)
To git@github.com:schacon/simplegit.git
* [new tag]          v1.5 -> v1.5
```

Si tienes un montón de etiquetas que quieres enviar a la vez, también puedes usar la opción `--tags` en el comando `git push`. Esto transfiere todas tus etiquetas que no estén ya en el servidor remoto.

```
$ git push origin --tags
Counting objects: 50, done.
Compressing objects: 100% (38/38), done.
Writing objects: 100% (44/44), 4.56 KiB, done.
Total 44 (delta 18), reused 8 (delta 1)
To git@github.com:schacon/simplegit.git
* [new tag]          v0.1 -> v0.1
* [new tag]          v1.2 -> v1.2
* [new tag]          v1.4 -> v1.4
* [new tag]          v1.4-lw -> v1.4-lw
* [new tag]          v1.5 -> v1.5
```

Ahora, cuando alguien clone o reciba de tu repositorio, obtendrá también todas tus etiquetas.

[Anterior](#)

2.5. Trabajando con repositorios remotos

[Siguiente](#)

2.7. Consejos y trucos

