

git-flow cheatsheet

creado por [Daniel Kummer](#)

[Tweet](#)

ramificaciones eficientes usando git-flow, por [Vincent Driessen](#)

traducciones: [English](#) - [Castellano](#) - [Brazilian Portugues](#) - [繁體中文\(Traditional Chinese\)](#) - [简体中文\(Simplified Chinese\)](#) - [日本語](#) - [한국어\(Korean\)](#) - [Français](#) - [Italiano](#) - [Nederlands](#) - [Русский\(Russian\)](#) - [Deutsch\(German\)](#) - [Català\(Catalan\)](#) - [Română\(Romanian\)](#) - [Ελληνικά\(Greek\)](#) - [Українська\(Ukrainian\)](#) - [Tiếng Việt\(Vietnamese\)](#) - [Polski](#) - [العربية](#) - [فارسی](#) - [Azərbaycanca\(Azerbaijani\)](#) - [Bahasa Indonesia](#)

Acerca de git-flow

git-flow es un conjunto de extensiones para git que proveen comandos de alto nivel para operar repositorios basados en el modelo de ramificaciones de Vincent Driessen. [más](#)

★ ★ ★

Esta hoja de referencia explica las operaciones básicas de git-flow, los comandos y sus efectos.

★ ★ ★

Consejos básicos

- ★ Git flow provee una excelente ayuda en la línea de comandos y e información. Léete con atención lo que sucede...
- ★ El cliente para OSX/Windows **Sourcetree** es una excelente GUI para git y tiene soporte para git-flow
- ★ - Git-flow funciona basándose en fusiones de ramas (merge). No reorganiza (branch rebase) las ramas de características (feature branches).

★ ★ ★

Configuración

- ★ Un prerequisite es una instalación de git en funcionamiento.
- ★ Git flow funciona en macOS, Linux y Windows

★ ★ ★

macOS

Homebrew

```
$ brew install git-flow-avh
```

Macports

```
$ port install git-flow-avh
```

Linux

```
$ apt-get install git-flow
```

Para instrucciones de instalación detalladas, por favor, visite la **[wiki de git flow](#)**



Windows (Cygwin)

```
$ wget -q -O - --no-check-certificate  
https://raw.githubusercontent.com/petervanderdoes/gitflow-  
avh/develop/contrib/gitflow-installer.sh install  
stable | bash
```

Necesitarás wget y util-linux para instalar git-flow.

Introducción

Git flow necesita ser inicializado para poder alterar la configuración del proyecto.

★ ★ ★

Inicialización

Comience a usar git-flow inicializándolo desde dentro de un repositorio git existente:

```
git flow init
```

Deberá contestar algunas preguntas relacionadas con las convenciones de nombres para las ramas.

Se recomienda utilizar los valores predeterminados.

Características

- ★ Desarrollar características para futuras versiones
- ★ Es típico que sólo se use en los repositorios para desarrollo

★ ★ ★

Comenzar una nueva característica

El desarrollo de nuevas características parte de la rama 'develop'.

Comience una nueva característica usando

```
git flow feature start  
MYFEATURE
```

Esta acción crea una nueva rama derivada de 'develop' y cambia a esta, estableciéndola como rama de trabajo actual.

Finalizar una característica

Finaliza el desarrollo de una característica. Esta acción realiza lo siguiente:

- ★ Fusiona MYFEATURE en 'develop'
- ★ Borra la rama MYFEATURE

★ **Cambia a la rama 'develop', estableciéndola como rama de trabajo actual**

```
git flow feature finish  
MYFEATURE
```

Publicar una característica

¿Estás trabajando colaborativamente?

Publica una característica a un servidor remoto para que así pueda ser vista por otros.

```
git flow feature publish  
MYFEATURE
```

Obteniendo características publicadas

Obten una característica publicada por otro.

```
git flow feature pull  
origin MYFEATURE
```

Puedes mantener un seguimiento de sus cambios usando.

```
git flow feature track  
MYFEATURE
```



Publicar una versión

★ **Prepara una versión para producción**

★ **Permite arreglos menores y la preparación de los metadatos para la publicación**

★ ★ ★

Comenzar una publicación

Para comenzar una publicación, usa el comando **git flow release**. Creará una rama de publicación derivada de la rama 'develop'.

```
git flow release start  
RELEASE [BASE]
```

Opcionalmente, puede usar **[BASE]** indicando el código sha-1 del cambio desde el cual comenzar la versión de publicación. El cambio debe ser parte de la rama 'develop'.

★ ★ ★

Es apropiado publicar remotamente la rama de publicación después de crearla para permitir que otros desarrolladores envíen cambios para esta versión. Hazlo de forma similar a publicar características:

```
git flow release publish  
RELEASE
```



(Puede establecer el seguimiento de los cambios de la publicación remota utilizando el siguiente comando)

```
git flow release track  
RELEASE
```

Concluir una publicación

Dar cierre a una publicación es un gran paso. Realiza varias acciones:

- ★ **Fusiona la rama de la publicación con la rama 'master'**
- ★ **Etiqueta el cambio con su nombre**
- ★ **Vuelve a fusionar la publicación con la rama 'develop'**
- ★ **Borra la rama de la publicación**

```
git flow release finish  
RELEASE
```

No olvides añadir las tags con

```
git push --tags
```

Revisiones

- ★ Las revisiones surgen de la necesidad de actuar inmediatamente cuando la versión ejecutándose en producción se encuentra en un estado que no deseamos
- ★ Puede ramificarse desde la versión correspondiente etiquetada en la rama 'master' que corresponda a la versión en producción.

★ ★ ★

git flow hotfix start

Como otros comandos de git flow, una revisión se abre con

```
git flow hotfix start  
VERSION [BASENAME]
```

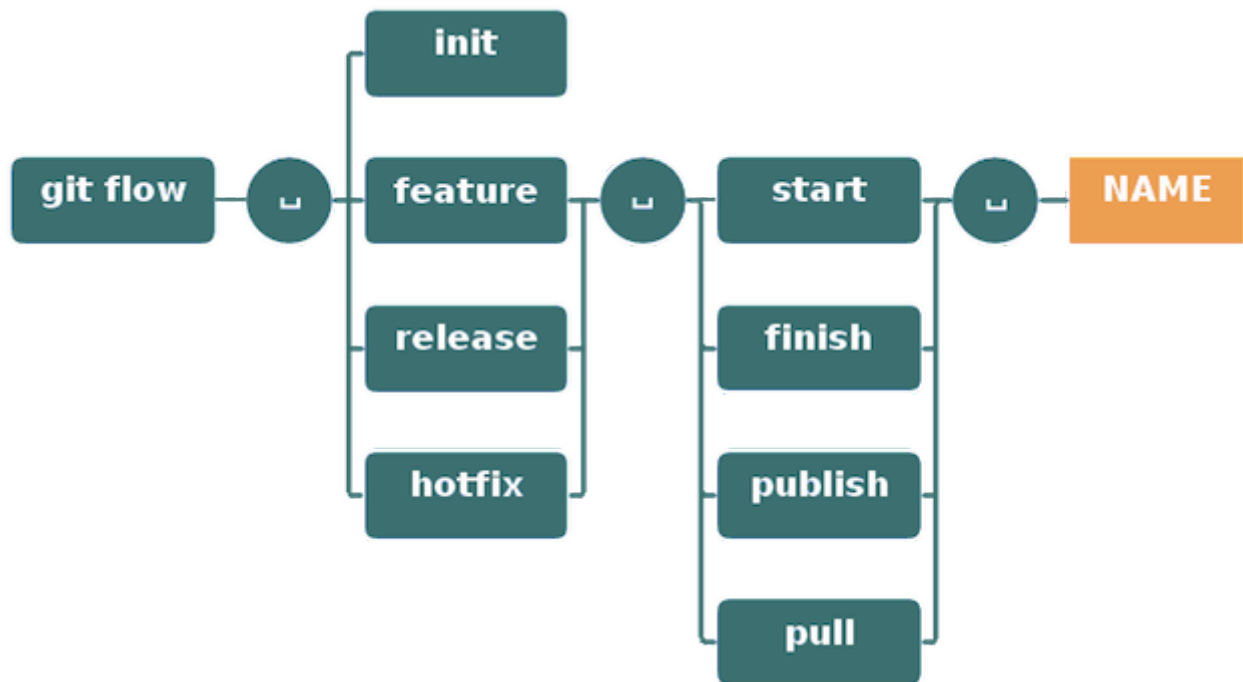
El argumento de la versión determina el nombre de la revisión. Opcionalmente, puede agregar un nombre para la base desde la cual comenzar.

Cierra una revisión

Al cerrar una revisión, esta se fusiona en las ramas 'develop' y 'master'. Luego, el cambio en 'master' es etiquetado con el nombre de la revisión.

```
git flow hotfix finish  
VERSION
```


Comandos



Tareas pendientes

★ ★ ★

- ★ Aquí no se han cubierto todos los comandos disponibles, sólo los más importantes
- ★ Aún puede usar todos los comandos de git normales tal y como los conoces, git-flow sólo es una herramienta más que puedes agregar a tu colección.
- ★ La característica 'support' aún está en fase beta y no se recomienda usarla
- ★ Si quiere proveer traducciones, serán bien recibidas.

★ ★ ★

Comentarios

*comments powered by **Disqus***