

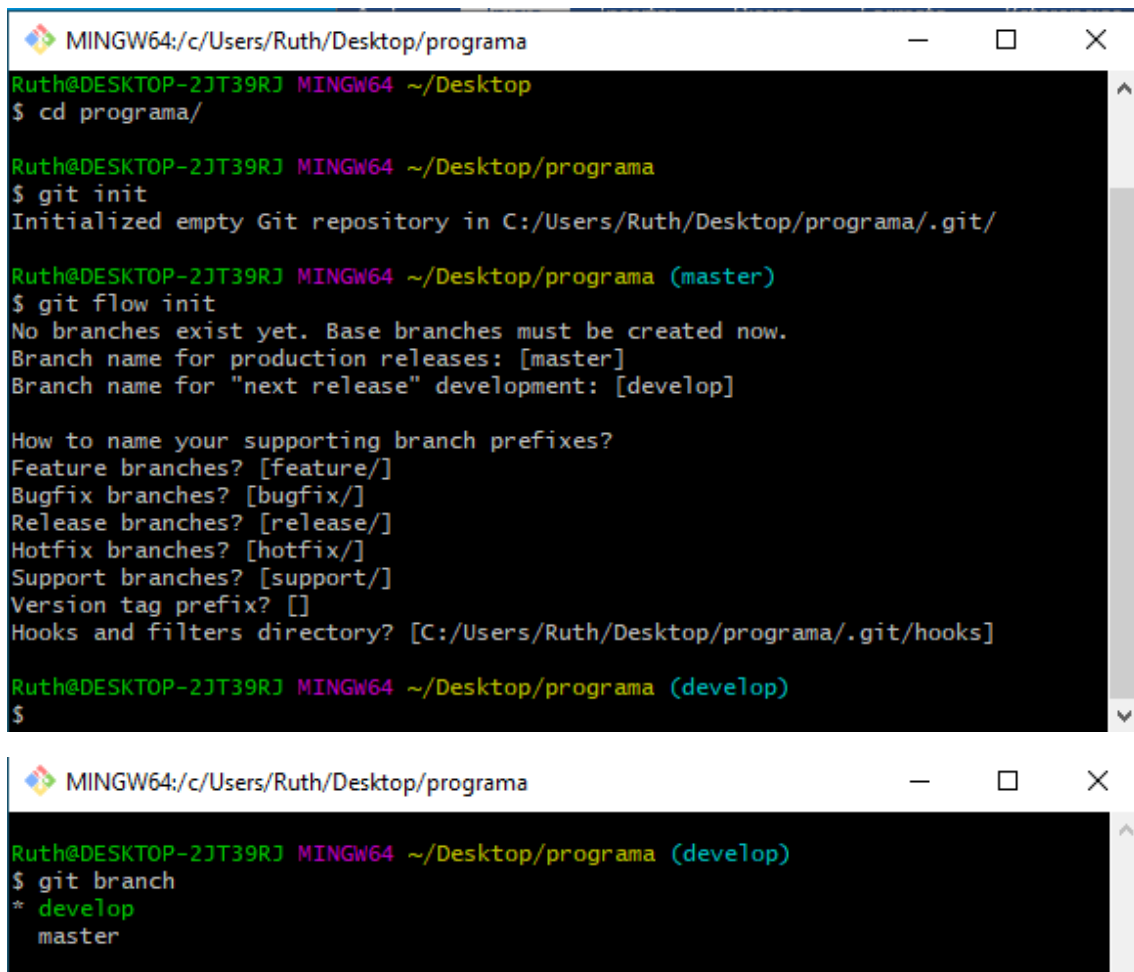
# SEMANA 04: GIT FLOW

## 1. Inicializando un proyecto

En nuestra consola bash se crea una carpeta y se escribe los siguientes comandos git:

```
$git init
$git flow init
$git branch
```

Git flow crea todas las *branches* necesarias.



```
MINGW64:/c/Users/Ruth/Desktop/programa
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop
$ cd programa/

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa
$ git init
Initialized empty Git repository in C:/Users/Ruth/Desktop/programa/.git/

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (master)
$ git flow init
No branches exist yet. Base branches must be created now.
Branch name for production releases: [master]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/Ruth/Desktop/programa/.git/hooks]

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (develop)
$

MINGW64:/c/Users/Ruth/Desktop/programa
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (develop)
$ git branch
* develop
  master
```

Luego se crea el repositorio en GitHub


## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

---

Owner

Repository name \*

 jenifer23 ▾

 / 

Git Flow ✓


Great repository names 

Your new repository will be created as Git-Flow. [Learn about upgraded-potato?](#)


Description (optional)

Semana 04

---

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

---


Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.


Add .gitignore: **None** ▾

 | 

Add a license: **None** ▾ 

---

Create repository



En nuestra consola bash se escribe los comandos:

```
$git remote add origin direccion
$git push -u origin master
$git branch -a
$git push -u origin develop
$git branch -a
```

Se sube nuestro proyecto y sus nuevas ramas (**master y develop**) a un servidor remoto creado en Github

```
MINGW64:/c/Users/Ruth/Desktop/programa

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (develop)
$ git remote add origin https://github.com/jenifer23/Git-Flow.git

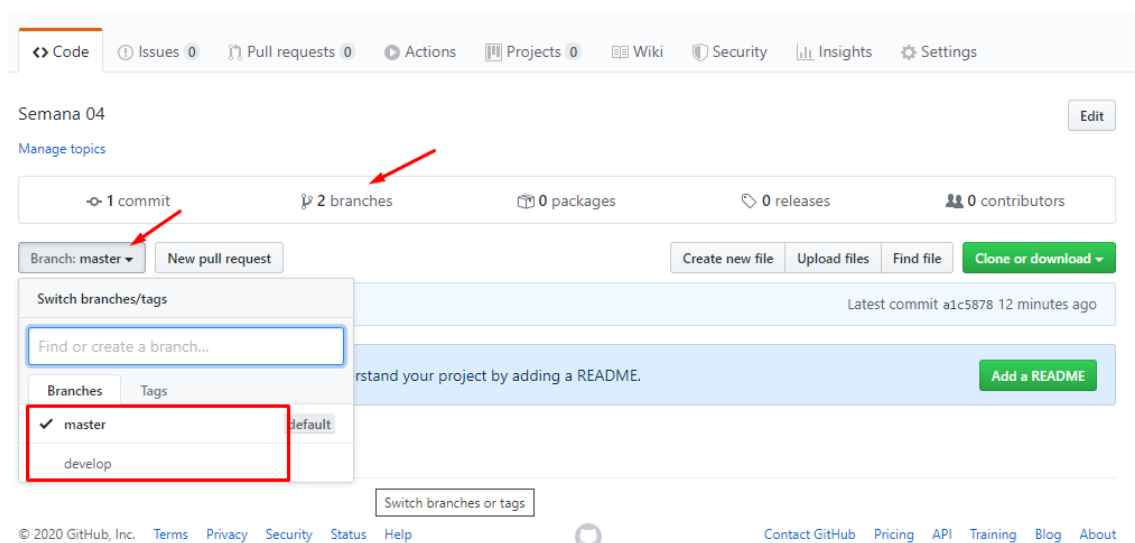
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (develop)
$ git push -u origin master
Enumerating objects: 2, done.
Counting objects: 100% (2/2), done.
Writing objects: 100% (2/2), 165 bytes | 165.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0)
To https://github.com/jenifer23/Git-Flow.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (develop)
$ git branch -a
* develop
  master
  remotes/origin/master

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (develop)
$ git push -u origin develop
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/jenifer23/Git-Flow/pull/new/develop
remote:
To https://github.com/jenifer23/Git-Flow.git
 * [new branch]      develop -> develop
Branch 'develop' set up to track remote branch 'develop' from 'origin'.

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (develop)
$ git branch -a
* develop
  master
  remotes/origin/develop
  remotes/origin/master
```

En nuestro repositorio remoto se mostraría así:



## 2. Creando una rama feature

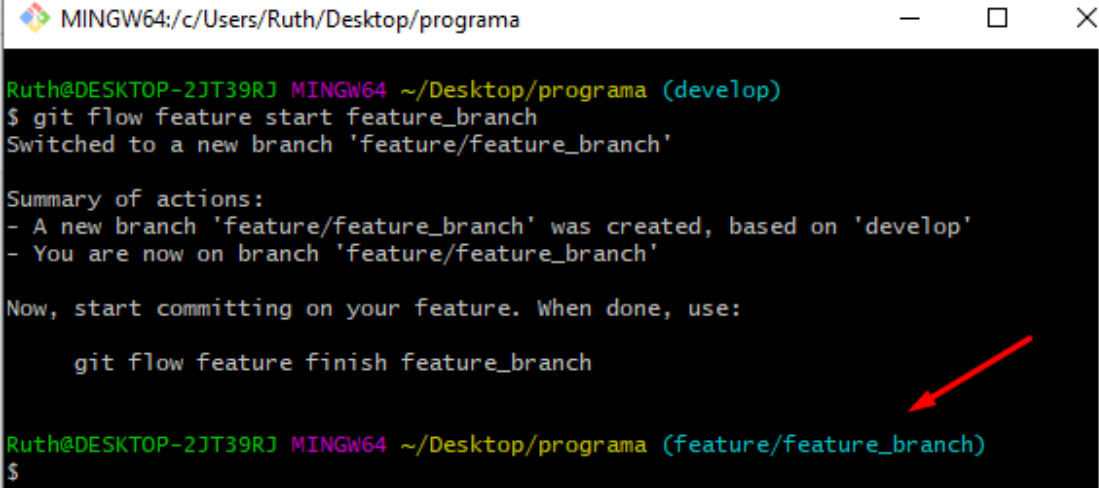
Con el uso de los comandos Git Flow en nuestra consola bash se escribe:

```
$git flow feature start feature_branch
```

Sin git flow el flujo sería el siguiente:

```
$git checkout develop  
$git checkout -b feature_branch
```

Git flow automáticamente crea la nueva rama e hizo un checkout a ésta.



The screenshot shows a terminal window titled 'MINGW64; c:/Users/Ruth/Desktop/programa'. The user is in the 'develop' branch. They run the command '\$ git flow feature start feature\_branch'. The output indicates that a new branch 'feature/feature\_branch' was created based on 'develop' and the user is now on that branch. A summary of actions is provided: '- A new branch 'feature/feature\_branch' was created, based on 'develop'' and '- You are now on branch 'feature/feature\_branch''. The user is then instructed to start committing on their feature and to use the command 'git flow feature finish feature\_branch' when done. A red arrow points to this command. The prompt changes to show the user is now in the 'feature/feature\_branch' branch.

```
MINGW64; c:/Users/Ruth/Desktop/programa  
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (develop)  
$ git flow feature start feature_branch  
Switched to a new branch 'feature/feature_branch'  
  
Summary of actions:  
- A new branch 'feature/feature_branch' was created, based on 'develop'  
- You are now on branch 'feature/feature_branch'  
  
Now, start committing on your feature. When done, use:  
  
    git flow feature finish feature_branch  
  
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (feature/feature_branch)  
$
```

En nuestro repositorio creamos un archivo *inicio.html*



The screenshot shows a code editor with a file named 'inicio.html'. The code is a simple HTML document with a title 'Inicio' and a body containing the text 'Bienvenidos a Git Flow'.

```
1 <!DOCTYPE html>  
2 <html>  
3 <head>  
4     <title>Inicio</title>  
5     <meta charset="UTF-8">  
6 </head>  
7 <body>  
8     <h1>Bienvenidos a Git Flow</h1>  
9 </body>  
10 </html>
```

Guardamos los cambios realizados y hacemos un commit.

```
$git status  
$git add .  
$git status  
$git commit -m 'Agregando inicio.html'
```

```
MINGW64:/c/Users/Ruth/Desktop/programa

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (feature/feature_branch)
$ git status
On branch feature/feature_branch
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        inicio.html

nothing added to commit but untracked files present (use "git add" to track)
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (feature/feature_branch)
$ git add .
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (feature/feature_branch)
$ git status
On branch feature/feature_branch
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   inicio.html

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (feature/feature_branch)
$ git commit -m 'Agregando inicio.html'
[feature/feature_branch 4896330] Agregando inicio.html
1 file changed, 10 insertions(+)
create mode 100644 inicio.html
```

Luego publicaremos la nueva rama para su colaboración, para ello en nuestra consola bash escribimos:

```
$git flow feature publish feature_branch
```

```
MINGW64:/c/Users/Ruth/Desktop/programa

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (feature/feature_branch)
$ git flow feature publish feature_branch
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 362 bytes | 362.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'feature/feature_branch' on GitHub by visiting
:
remote:   https://github.com/jenifer23/Git-Flow/pull/new/feature/feature_bran
ch
remote:
To https://github.com/jenifer23/Git-Flow.git
* [new branch]      feature/feature_branch -> feature/feature_branch
Branch 'feature/feature_branch' set up to track remote branch 'feature/feature_b
ranch' from 'origin'.
Already on 'feature/feature_branch'
Your branch is up to date with 'origin/feature/feature_branch'.

Summary of actions:
- The remote branch 'feature/feature_branch' was created or updated
- The local branch 'feature/feature_branch' was configured to track the remote b
ranch
- You are now on branch 'feature/feature_branch'
```

Revisamos GitHub para ver los cambios:

Rama master y develop:

The screenshot shows the GitHub repository interface for a project. At the top, there's a navigation bar with links to Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below this, the repository name is 'Semana 04'. A summary bar shows 1 commit, 3 branches, 0 packages, 0 releases, and 0 contributors. The 'Your recently pushed branches' section shows two branches: 'feature/feature\_branch' (7 minutes ago) and 'develop' (9 minutes ago). For each branch, there's a 'Compare & pull request' button. Below the branches, there's a 'Branch: master' dropdown and a 'New pull request' button. The commit history shows 'Juan Carlos Initial commit' as the latest commit for both branches, 10 hours ago. Red arrows point to the 'Branch: master' dropdown, the 'New pull request' button, and the commit history entry.

Rama feature\_branch:

The screenshot shows the GitHub repository interface for the 'feature\_branch'. The 'Your recently pushed branches' section shows 'feature/feature\_branch' (10 minutes ago) with a 'Compare & pull request' button. Below this, there's a 'Branch: feature/featur...' dropdown and a 'New pull request' button. The commit history shows 'Juan Carlos Agregando inicio.html' as the latest commit, 16 minutes ago. The file list shows 'inicio.html' with a red arrow pointing to it. Red arrows point to the 'Branch: feature/featur...' dropdown, the commit history entry, and the file list entry.

Cuando se termine el trabajo de desarrollo de la característica, el siguiente paso es fusionar el **feature** con la rama **develop** escribimos en la consola el siguiente comando:

```
$git flow feature finish feature_branch
```

Nos abrirá el editor de texto de Git que por defecto es Vim, nos pedirá ingresar o modificar el commit, estos son los pasos para salir de Vim:

```
Primero se pulsa la Tecla ESC, para situarnos en el buffer inferior
:q → sale del archivo, si no tenemos cambios sin guardar sale sin
más.
:q! → sale del archivo descartando los cambios no guardados.
:w → guarda los cambios del archivo, pero no sale de Vim para seguir
editando el archivo.
:wq → guarda los cambios y sale de Vim.
:x → igual que el anterior.
```

Git Flow automáticamente fusiona la rama **feature\_branch** a **develop** y lo elimina, también elimina la rama del servidor remoto y cambia a la rama **develop**.

```
MINGW64:/c/Users/Ruth/Desktop/programa

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (feature/feature_branch)
$ git flow feature finish feature_branch
Branches 'feature/feature_branch' and 'origin/feature/feature_branch' have diverged.
And local branch 'feature/feature_branch' is ahead of 'origin/feature/feature_branch'.
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Merge made by the 'recursive' strategy.
 inicio.html | 20 ++++++++
 1 file changed, 20 insertions(+)
 create mode 100644 inicio.html
To https://github.com/jenifer23/Git-Flow.git
 - [deleted]          feature/feature_branch
Deleted branch feature/feature_branch (was 20a6eb9).

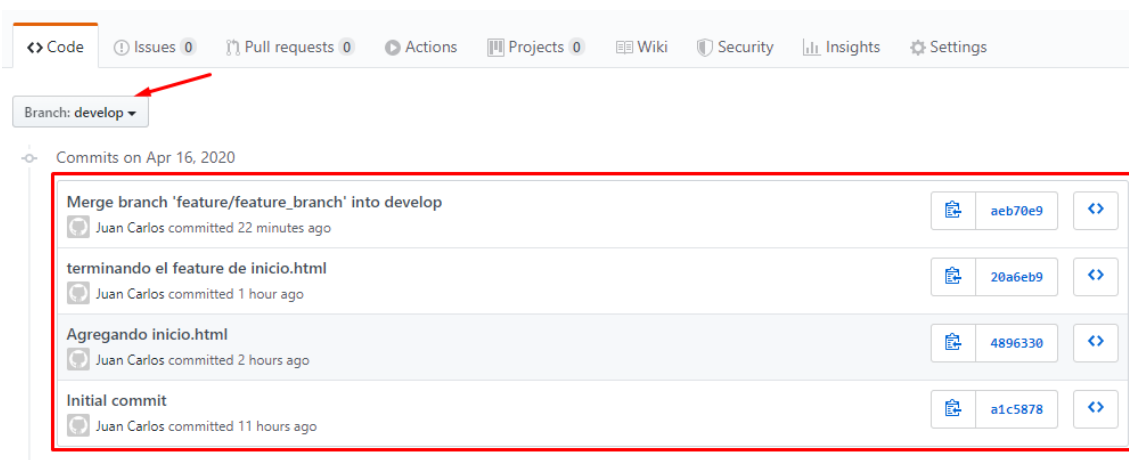
Summary of actions:
- The feature branch 'feature/feature_branch' was merged into 'develop'
- Feature branch 'feature/feature_branch' has been locally deleted; it has been remotely deleted from 'origin'
- You are now on branch 'develop'
```

Guardamos los cambios en el servidor remoto.

```
MINGW64:/c/Users/Ruth/Desktop/programa

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (develop)
$ git push origin develop
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 1.93 KiB | 989.00 KiB/s, done.
Total 7 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/jenifer23/Git-Flow.git
 a1c5878..aeb70e9  develop -> develop
```

En GitHub nos muestra la rama develop:

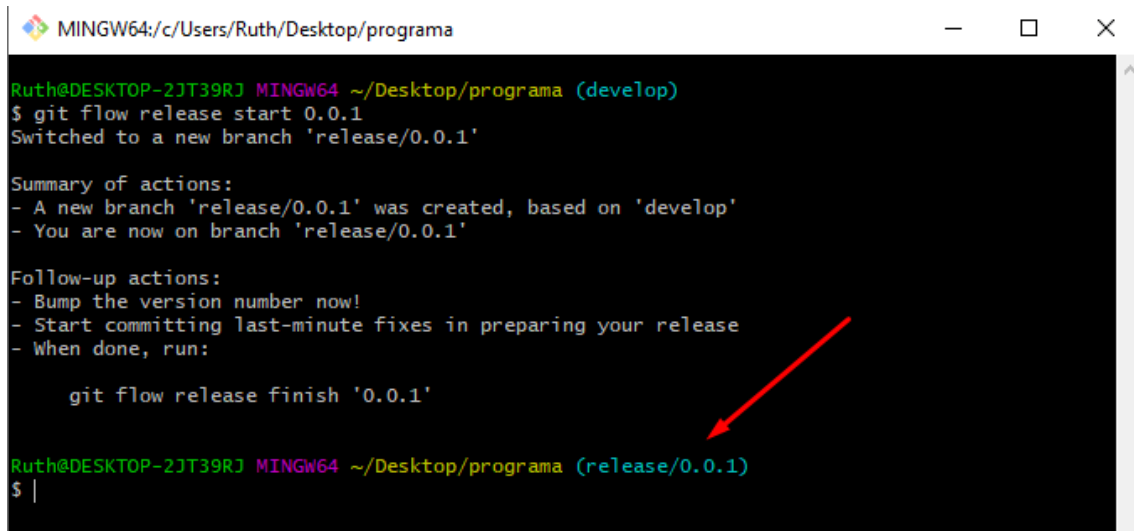


### 3. Creando una rama release

Con el uso de los comandos Git Flow en nuestra consola bash se escribe:

```
$git flow release start 0.0.1
```

Git flow automáticamente crea la nueva rama e hizo un checkout a ésta.



```
MINGW64:/c/Users/Ruth/Desktop/programa
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (develop)
$ git flow release start 0.0.1
Switched to a new branch 'release/0.0.1'

Summary of actions:
- A new branch 'release/0.0.1' was created, based on 'develop'
- You are now on branch 'release/0.0.1'

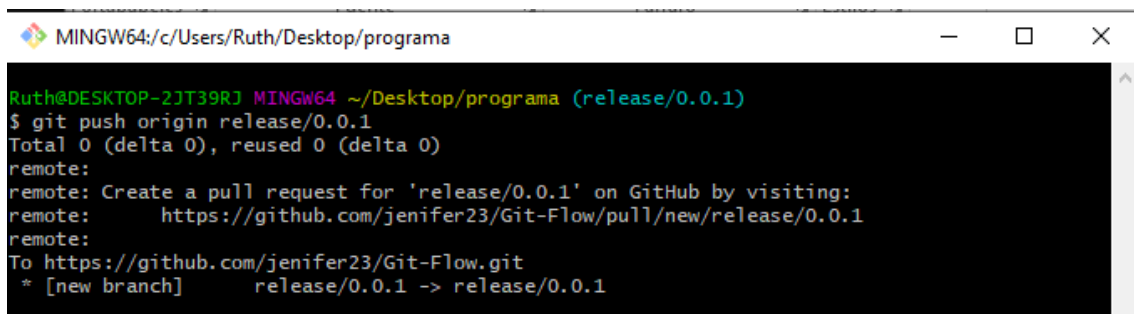
Follow-up actions:
- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

    git flow release finish '0.0.1'

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (release/0.0.1)
$ |
```

Los cambios se pueden enviar al servidor remoto para que los colaboradores puedan descargar:

```
$git push origin release/0.0.1
```



```
MINGW64:/c/Users/Ruth/Desktop/programa
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (release/0.0.1)
$ git push origin release/0.0.1
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'release/0.0.1' on GitHub by visiting:
remote:   https://github.com/jenifer23/Git-Flow/pull/new/release/0.0.1
remote:
To https://github.com/jenifer23/Git-Flow.git
 * [new branch]   release/0.0.1 -> release/0.0.1
```

Se crea una copia de todo el contenido de la rama **develop** a la rama **0.0.1**



[Code](#)
[Issues 0](#)
[Pull requests 0](#)
[Actions](#)
[Projects 0](#)
[Wiki](#)
[Security](#)
[Insights](#)
[Settings](#)

Semana 04 [Edit](#)

[Manage topics](#)

[4 commits](#)
[3 branches](#)
[0 packages](#)
[0 releases](#)
[0 contributors](#)

Your recently pushed branches:

- [develop](#) (19 minutes ago) [Compare & pull request](#)
- [release/0.0.1](#) (1 minute ago) [Compare & pull request](#)

Branch: [release/0.0.1](#) [New pull request](#)
[Create new file](#)
[Upload files](#)
[Find file](#)
[Clone or download](#)

This branch is 3 commits ahead of master. [Pull request](#) [Compare](#)

[Juan Carlos](#) Merge branch 'feature/feature\_branch' into develop
 Latest commit aeb70e9 40 minutes ago

[inicio.html](#) terminando el feature de inicio.html 1 hour ago

Creamos la carpeta donde se guardará la documentación técnica del proyecto, luego guardamos los cambios.

```
$git status
$git add .
$git status
$git commit -m 'Agregando documentación técnica'
```

```

MINGW64:/c/Users/Ruth/Desktop/programa
On branch release/0.0.1
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    "Documentos T\303\251cnicos/"

nothing added to commit but untracked files present (use "git add" to track)
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (release/0.0.1)
$ git add .
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (release/0.0.1)
$ git status
On branch release/0.0.1
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   "Documentos T\303\251cnicos/Arquitectura de software.docx"
    new file:   "Documentos T\303\251cnicos/Plan de Trabajo.mpp"

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (release/0.0.1)
$ git commit -m 'Agregando documentación técnica'
[release/0.0.1 3370f9d] Agregando documentación técnica
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "Documentos T\303\251cnicos/Arquitectura de software.docx"
create mode 100644 "Documentos T\303\251cnicos/Plan de Trabajo.mpp"
  
```

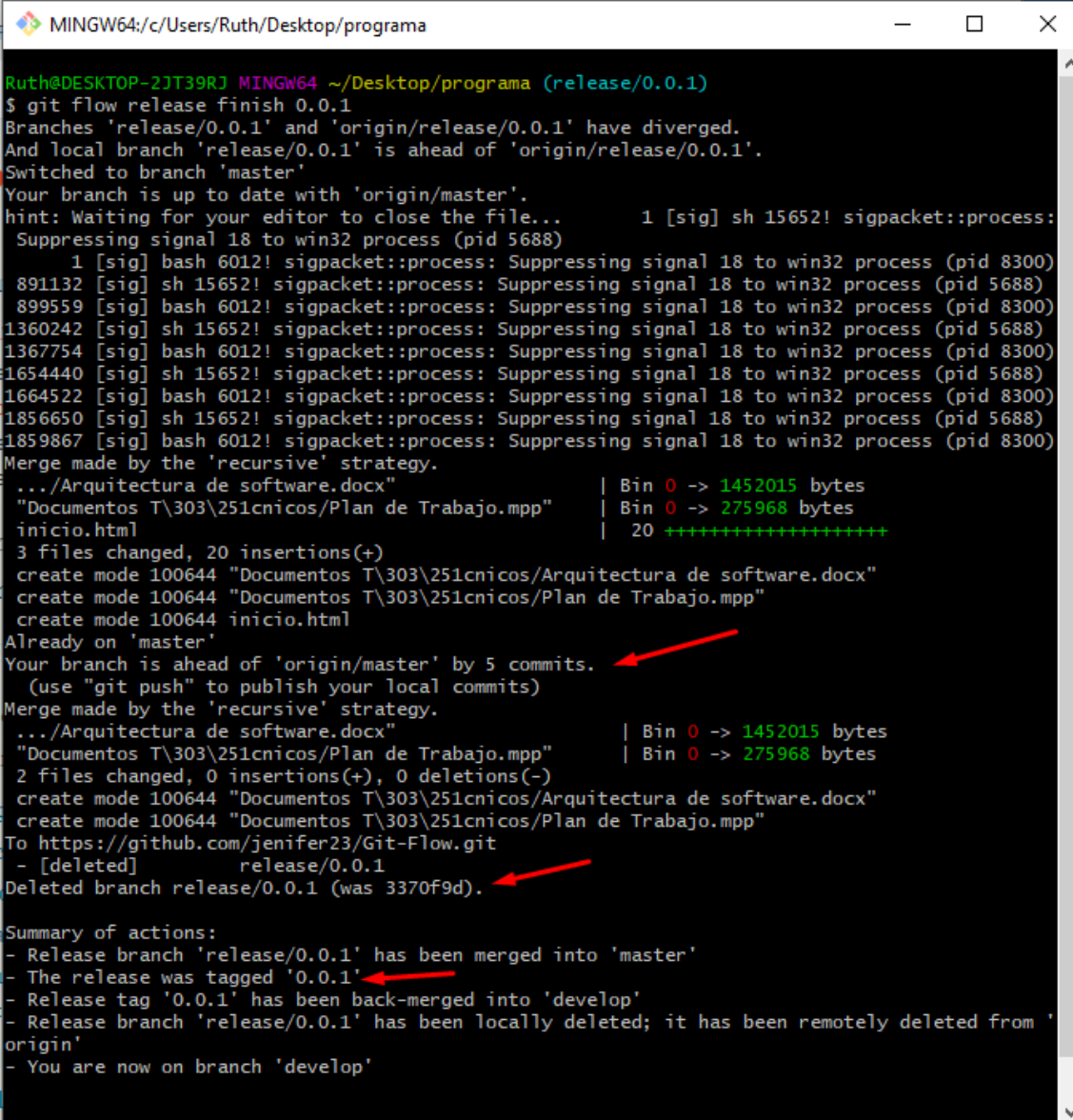
En ese punto se poder hacer Quality Assurance, todo lo que tenga que ver con testing y aseguramiento de calidad.

Para finalizar el reléase se escribe el siguiente comando:

```
$git flow release finish 0.0.1
```

Nos muestra el editor de Vim, primero pedirá la descripción del commit y luego el número del tag para la rama **master** y luego el commit para la rama **develop**.

Git flow, automáticamente fusiona la rama **0.0.1** a la rama **master**, le asigna un **tag** y elimina de forma local y remota la rama **0.0.1**.



```
MINGW64: c:/Users/Ruth/Desktop/programa
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (release/0.0.1)
$ git flow release finish 0.0.1
Branches 'release/0.0.1' and 'origin/release/0.0.1' have diverged.
And local branch 'release/0.0.1' is ahead of 'origin/release/0.0.1'.
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
hint: Waiting for your editor to close the file...      1 [sig] sh 15652! sigpacket::process:
Suppressing signal 18 to win32 process (pid 5688)
1 [sig] bash 6012! sigpacket::process: Suppressing signal 18 to win32 process (pid 8300)
891132 [sig] sh 15652! sigpacket::process: Suppressing signal 18 to win32 process (pid 5688)
899559 [sig] bash 6012! sigpacket::process: Suppressing signal 18 to win32 process (pid 8300)
1360242 [sig] sh 15652! sigpacket::process: Suppressing signal 18 to win32 process (pid 5688)
1367754 [sig] bash 6012! sigpacket::process: Suppressing signal 18 to win32 process (pid 8300)
1654440 [sig] sh 15652! sigpacket::process: Suppressing signal 18 to win32 process (pid 5688)
1664522 [sig] bash 6012! sigpacket::process: Suppressing signal 18 to win32 process (pid 8300)
1856650 [sig] sh 15652! sigpacket::process: Suppressing signal 18 to win32 process (pid 5688)
1859867 [sig] bash 6012! sigpacket::process: Suppressing signal 18 to win32 process (pid 8300)
Merge made by the 'recursive' strategy.
.../Arquitectura de software.docx" | Bin 0 -> 1452015 bytes
"Documentos T\303\251cnicos/Plan de Trabajo.mpp" | Bin 0 -> 275968 bytes
inicio.html | 20 ++++++
3 files changed, 20 insertions(+)
create mode 100644 "Documentos T\303\251cnicos/Arquitectura de software.docx"
create mode 100644 "Documentos T\303\251cnicos/Plan de Trabajo.mpp"
create mode 100644 inicio.html
Already on 'master'
Your branch is ahead of 'origin/master' by 5 commits.
(use "git push" to publish your local commits)
Merge made by the 'recursive' strategy.
.../Arquitectura de software.docx" | Bin 0 -> 1452015 bytes
"Documentos T\303\251cnicos/Plan de Trabajo.mpp" | Bin 0 -> 275968 bytes
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "Documentos T\303\251cnicos/Arquitectura de software.docx"
create mode 100644 "Documentos T\303\251cnicos/Plan de Trabajo.mpp"
To https://github.com/jenifer23/Git-Flow.git
- [deleted] release/0.0.1
Deleted branch release/0.0.1 (was 3370f9d).

Summary of actions:
- Release branch 'release/0.0.1' has been merged into 'master'
- The release was tagged '0.0.1'
- Release tag '0.0.1' has been back-merged into 'develop'
- Release branch 'release/0.0.1' has been locally deleted; it has been remotely deleted from 'origin'
- You are now on branch 'develop'
```

Para guardar los cambios en el servidor remoto:

```
$git push origin master
$git push origin develop
$git push --tags
```

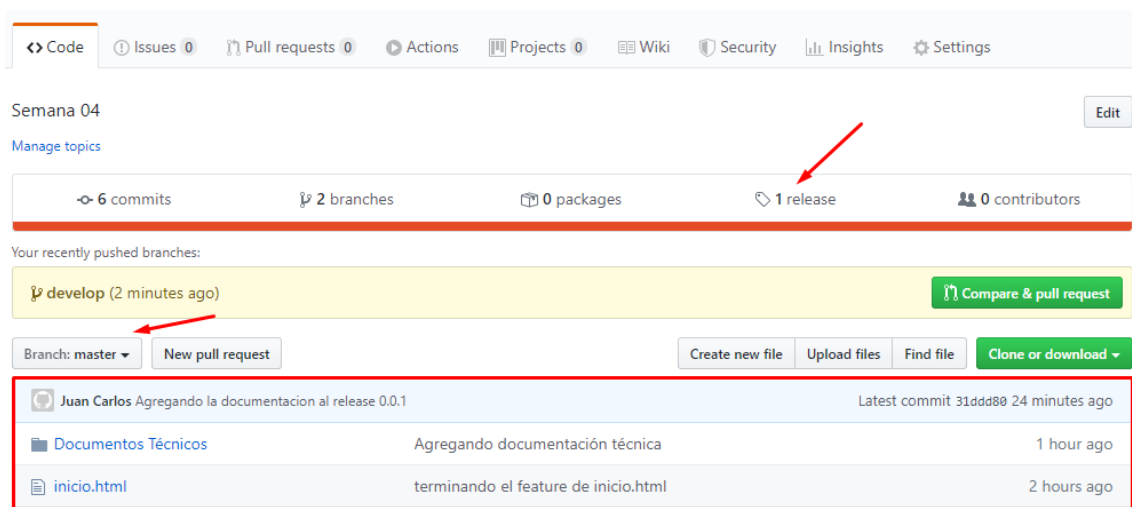
```
MINGW64:/c/Users/Ruth/Desktop/programa

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (develop)
$ git push origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.32 MiB | 470.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0)
To https://github.com/jenifer23/Git-Flow.git
   a1c5878..31ddd80  master -> master

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (develop)
$ git push origin develop
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 231 bytes | 231.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://github.com/jenifer23/Git-Flow.git
   aeb70e9..afd9652  develop -> develop

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (develop)
$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 158 bytes | 158.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://github.com/jenifer23/Git-Flow.git
 * [new tag]         0.0.1 -> 0.0.1
```

En GitHub, nos mostraría lo siguiente:



#### 4. Creando una rama hotfix

Con el uso de los comandos Git Flow en nuestra consola bash desde la rama **master** se escribe:

```
$git flow hotfix start hotfix_branch
```

Git flow automáticamente crea la nueva rama e hizo un checkout a ésta.

```
MINGW64:/c/Users/Ruth/Desktop/programa
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (master)
$ git flow hotfix start hotfix_branch
Switched to a new branch 'hotfix/hotfix_branch'

Summary of actions:
- A new branch 'hotfix/hotfix_branch' was created, based on 'master'
- You are now on branch 'hotfix/hotfix_branch'

Follow-up actions:
- Start committing your hot fixes
- Bump the version number now!
- When done, run:

    git flow hotfix finish 'hotfix_branch'

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (hotfix/hotfix_branch)
$ |
```

Los cambios se pueden enviar al servidor remoto para que los colaboradores puedan descargar:

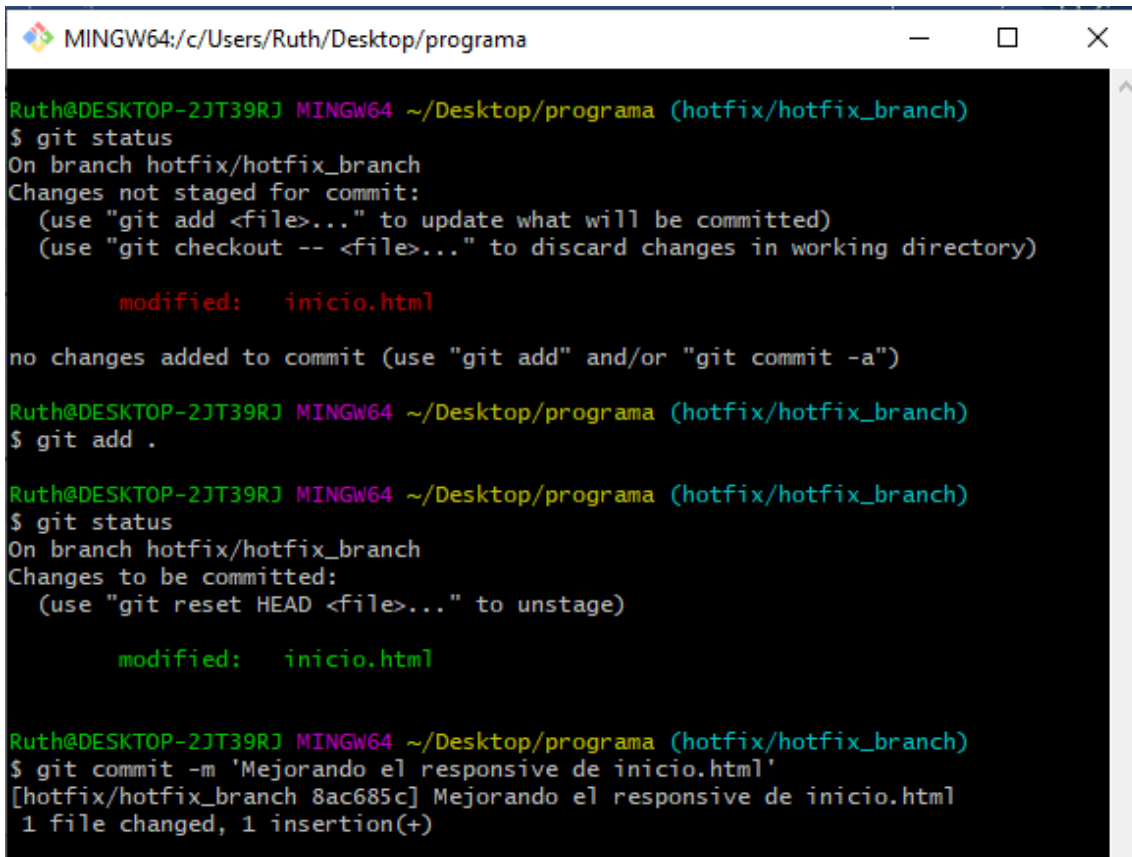
```
$git push origin hotfix/hotfix_branch
```

```
MINGW64:/c/Users/Ruth/Desktop/programa
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (hotfix/hotfix_branch)
$ git push origin hotfix/hotfix_branch
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'hotfix/hotfix_branch' on GitHub by visiting:
remote:   https://github.com/jenifer23/Git-Flow/pull/new/hotfix/hotfix_branch
remote:
To https://github.com/jenifer23/Git-Flow.git
 * [new branch]      hotfix/hotfix_branch -> hotfix/hotfix_branch
```

Se crea una copia de todo el contenido de la rama **master** a la rama **hotfix\_branch**

Modificamos nuestro **inicio.html** para que pueda ser responsive.

```
$git status
$git add .
$git status
$git commit -m 'Mejorando el responsive de inicio.html'
```



```
MINGW64:/c/Users/Ruth/Desktop/programa
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (hotfix/hotfix_branch)
$ git status
On branch hotfix/hotfix_branch
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   inicio.html

no changes added to commit (use "git add" and/or "git commit -a")
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (hotfix/hotfix_branch)
$ git add .
Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (hotfix/hotfix_branch)
$ git status
On branch hotfix/hotfix_branch
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   inicio.html

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (hotfix/hotfix_branch)
$ git commit -m 'Mejorando el responsive de inicio.html'
[hotfix/hotfix_branch 8ac685c] Mejorando el responsive de inicio.html
1 file changed, 1 insertion(+)
```

Para finalizar el hotfix se escribe el siguiente comando:

```
$git flow hotfix finish hotfix_branch
```

Nos muestra el editor de Vim, primero pedirá la descripción del commit y luego el número del tag para la rama **master** y luego el commit para la rama **develop**, esto con el fin de no perder en **develop** los arreglos hechos en **master**.

Git flow, automáticamente fusiona la rama **hotfix\_branch** a la rama **master**, le asigna un **tag** y elimina de forma local y remota la rama **hotfix\_branch**.

```
MINGW64:/c/Users/Ruth/Desktop/programa

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (hotfix/hotfix_branch)
$ git flow hotfix finish hotfix_branch
Branches 'hotfix/hotfix_branch' and 'origin/hotfix/hotfix_branch' have diverged.
And local branch 'hotfix/hotfix_branch' is ahead of 'origin/hotfix/hotfix_branch'.
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
Merge made by the 'recursive' strategy.
 inicio.html | 1 +
 1 file changed, 1 insertion(+)
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Merge made by the 'recursive' strategy.
 inicio.html | 1 +
 1 file changed, 1 insertion(+)
To https://github.com/jenifer23/Git-Flow.git
 - [deleted] hotfix/hotfix_branch
Deleted branch hotfix/hotfix_branch (was 8ac685c).

Summary of actions:
- Hotfix branch 'hotfix/hotfix_branch' has been merged into 'master'
- The hotfix was tagged 'hotfix_branch'
- Hotfix tag 'hotfix_branch' has been back-merged into 'develop'
- Hotfix branch 'hotfix/hotfix_branch' has been locally deleted; it has been remotely deleted from 'origin'
- You are now on branch 'develop'
```

Para guardar los cambios en el servidor remoto desde la rama **master**:

```
$git push origin master
$git push origin develop
$git push --tags
```

```
MINGW64:/c/Users/Ruth/Desktop/programa

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (develop)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (master)
$ git push origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 514 bytes | 514.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/jenifer23/Git-Flow.git
 31ddd80..fca8255 master -> master

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (master)
$ git push origin develop
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 228 bytes | 228.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://github.com/jenifer23/Git-Flow.git
 afd9652..6d6e8c1 develop -> develop

Ruth@DESKTOP-2JT39RJ MINGW64 ~/Desktop/programa (master)
$ git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 165 bytes | 165.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://github.com/jenifer23/Git-Flow.git
 * [new tag] hotfix_branch -> hotfix_branch
```

En GitHub, nos mostraría lo siguiente:

CodeIssues 0Pull requests 0ActionsProjects 0WikiSecurityInsightsSettings

Semana 04Edit

Manage topics

8 commits2 branches0 packages2 releases0 contributors

Your recently pushed branches:

develop (2 minutes ago)Compare & pull request

Branch: masterNew pull requestCreate new fileUpload filesFind fileClone or download

Juan Carlos fusionando branch 'hotfix/hotfix\_branch'Latest commit fca8255 9 minutes ago

Documentos TécnicosAgregando documentación técnica1 hour ago

inicio.htmlMejorando el responsive de inicio.html11 minutes ago