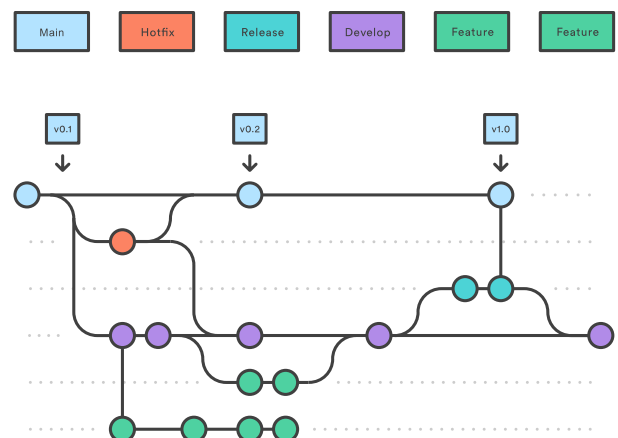


10-11-2021

# Flujo de Trabajo GitFlow

Nombre: Ricardo Espinosa Perez

Fuerza de Trabajo: DevSecOps



## Introducción

### Gitflow flujo de trabajo

Es un flujo de trabajo de git para gestionar ramas de git

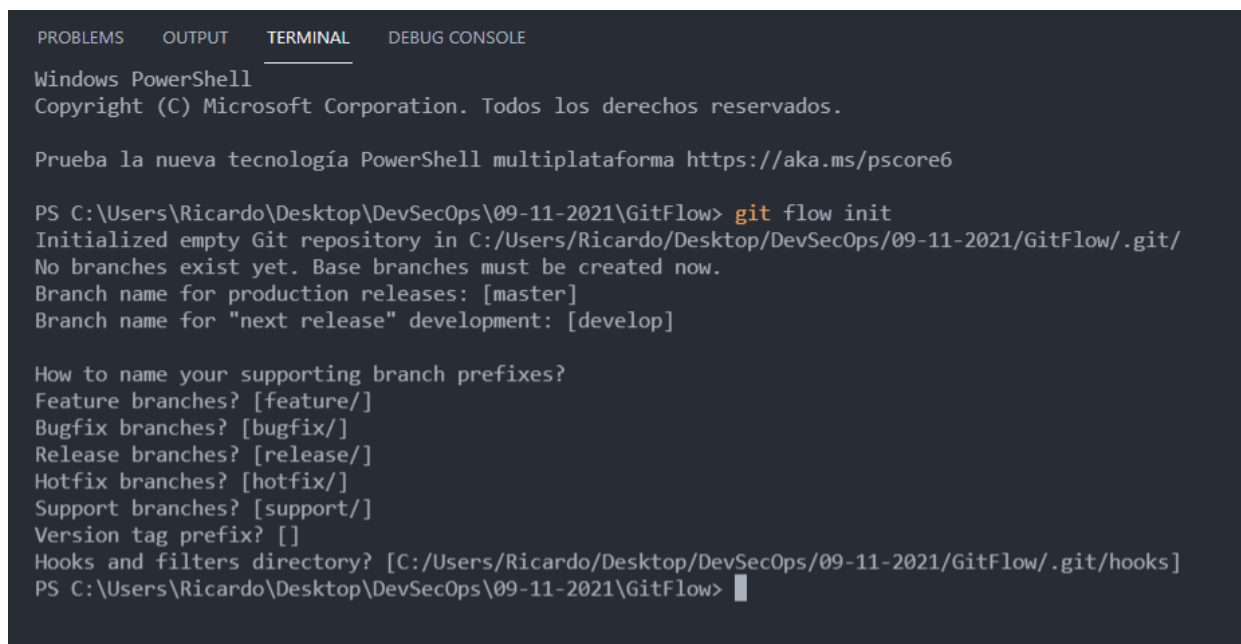
GitFlow: modelo alternativo de creación de ramas en git en la que se utiliza ramas de función y varias ramas principales.

Los desarrolladores crean una rama de función y retrasan su fusión con la rama principal del tronco hasta que la función este completado. Estas ramas de función de larga duración requieren más colaboración para la fusión y tienen mayor riesgo de desviarse de la rama troncal.

Este flujo ordena el tipo de ramas que se deben configurar.

Desarrollo para poder iniciar con git Flow deberemos contar con git (se puede descargar del siguiente enlace <https://git-scm.com/download/win>)

Para poder empezar a trabajar con git Flow tenemos que poner “git Flow init” en la carpeta del proyecto local como se muestra en la siguiente captura:



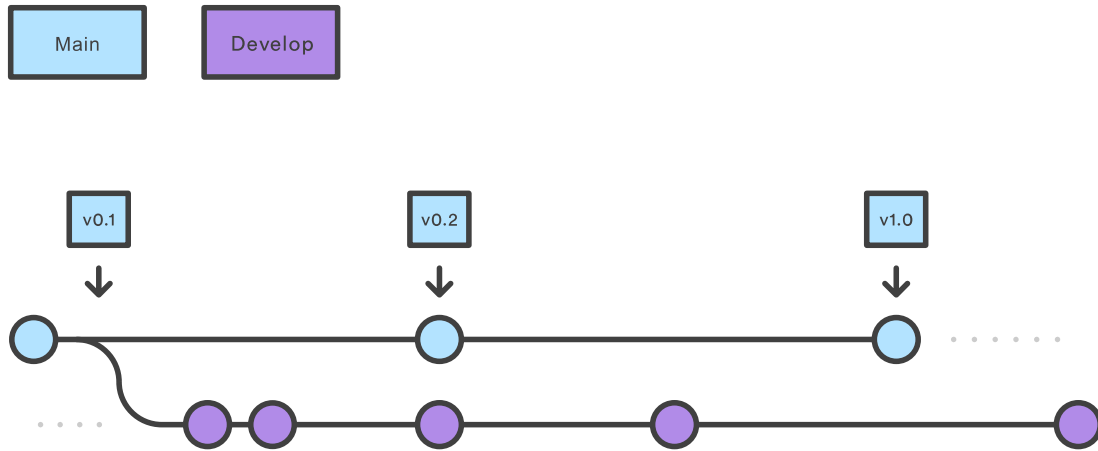
```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git flow init
Initialized empty Git repository in C:/Users/Ricardo/Desktop/DevSecOps/09-11-2021/GitFlow/.git/
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/Ricardo/Desktop/DevSecOps/09-11-2021/GitFlow/.git/hooks]
PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow>
```

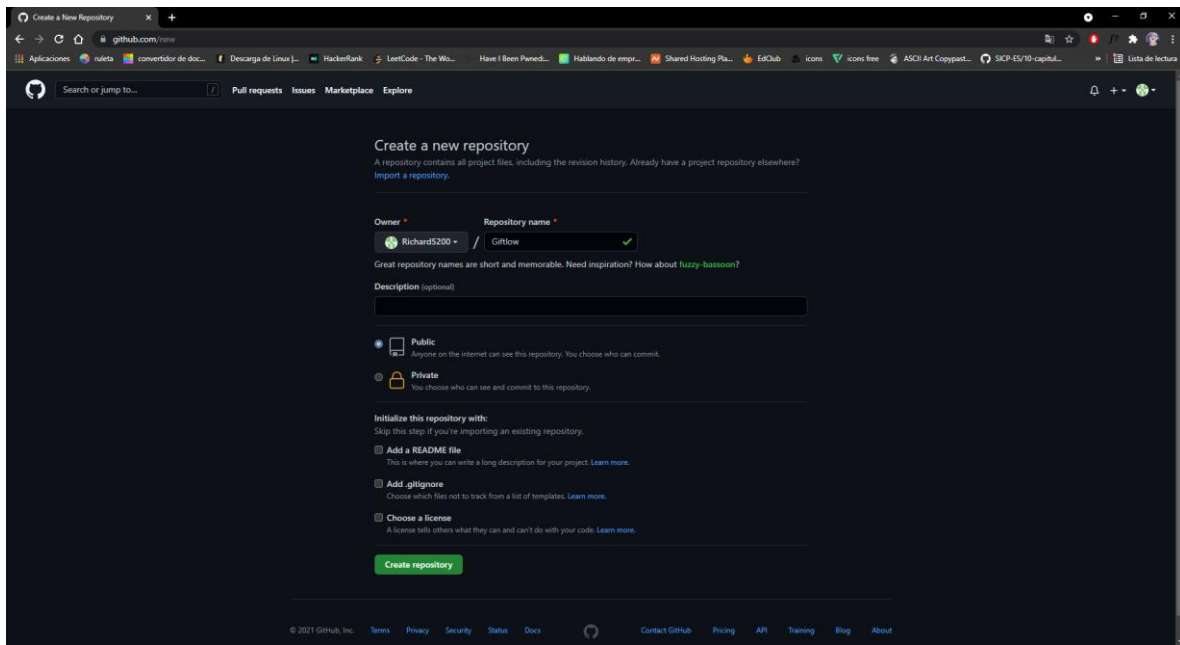
Al ejecutar el comando dirá que no existe ramas en el proyecto al no haber ramas creará dos la primera con el nombre de “master” y la otra con el nombre de “develop” en un diagrama se vera de la siguiente forma:



### Funcionamiento Ramas Principales y de Desarrollo

Con el flujo de trabajo gitflow trabaja utilizando dos ramas para registrar la historia del proyecto. La rama main almacena la historia de la publicación oficial o mas bien dicho almacena el proyecto central y la rama develop sirve para la integración de funciones.

Con el comando de git Flow init se nos creo una rama de develop debemos subirla a un repositorio de git en mi caso cree uno con el nombre de GitFlow

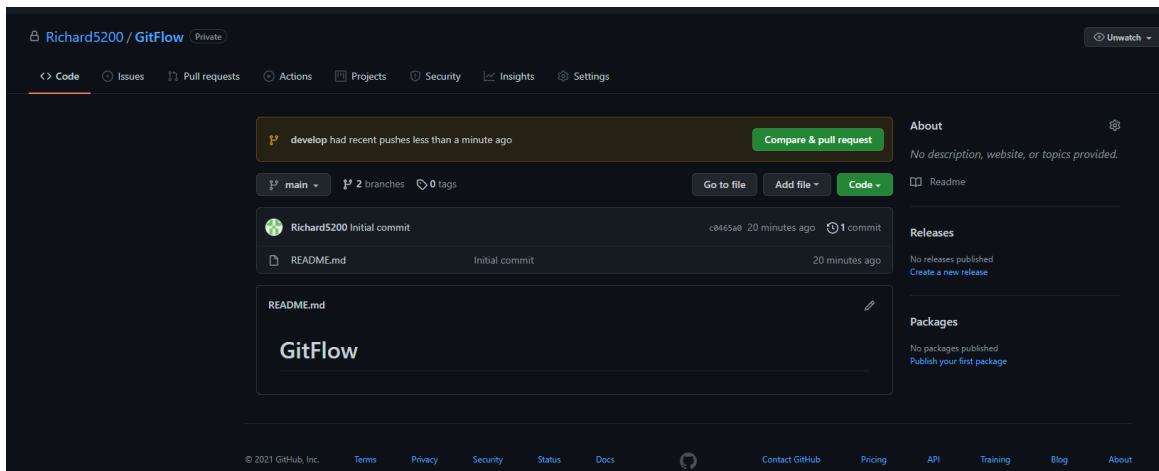


Una vez creado el repositorio en la consola de git colocamos la url del repositorio con el comando de git remote add origin y la url como se muestra en la siguiente imagen.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git remote add origin https://github.com/Richard5200/GitFlow.git
PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> |
```

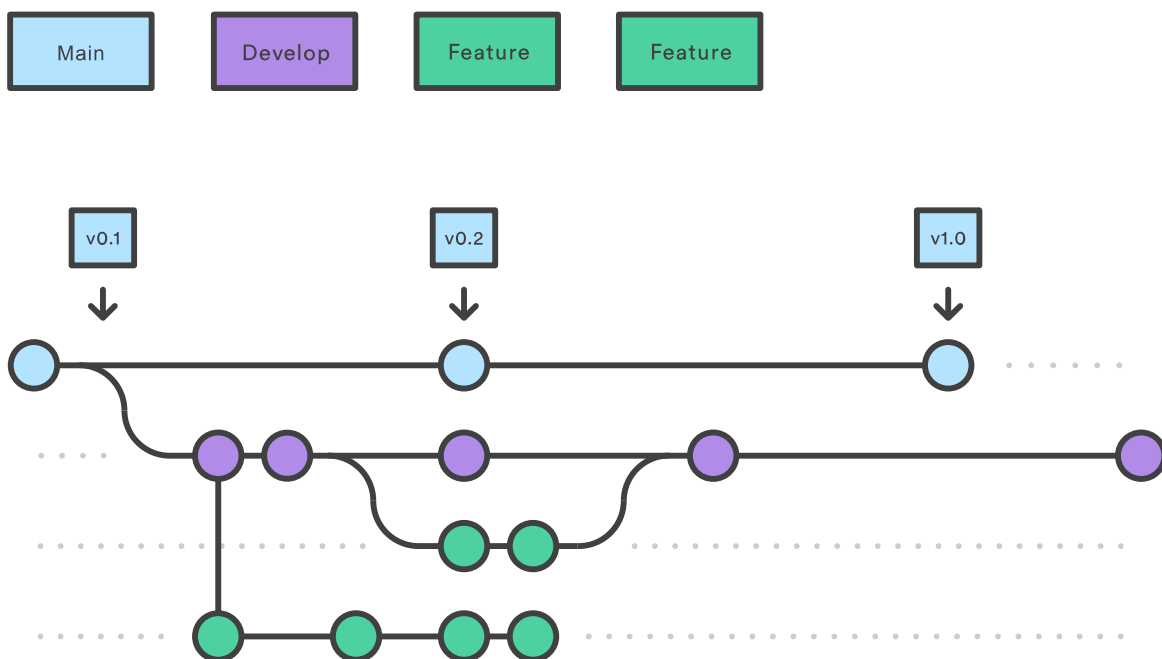
Después subimos la rama al repositorio.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git push -u origin develop
Enumerating objects: 2, done.
Counting objects: 100% (2/2), done.
Writing objects: 100% (2/2), 169 bytes | 169.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/Richard5200/GitFlow/pull/new/develop
remote:
To https://github.com/Richard5200/GitFlow.git
 * [new branch]      develop -> develop
Branch 'develop' set up to track remote branch 'develop' from 'origin'.
PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> |
```



## Ramas de función

Las funciones que colocamos en las de develop se debe crear otra rama que son las funciones en lugar de ramificarse en la main , las ramas feature utilizan la rama develop como rama primaria. Cuando una función esta terminada, se vuelve a fusionar en develop. Las funciones no deben interactuar nunca con la main como se ve en la siguiente figura.



## Creación de una rama de función

Para poder crear la rama feature tenemos que verificar que estemos en la rama de develop como aparece en la anterior imagen podemos verificar con el comando “git Branch” como se ve en la siguiente imagen:

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git branch
* develop
  master
PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> 
```

En caso que no estemos solo nos pasamos con el comando de “git checkout mas el nombre de la rama”

Para crear ocupamos el comando de git Flow feature start feature\_branch

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> 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

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> 
```

```
python - C:\Users\Ricardo\Desktop\areas pendientes

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> 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

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git branch
  develop
* feature/feature_branch
  master
PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> 
```

Nos dará un resumen de lo que se creo que fue la rama feature con la rama feature\_branch, al iniciarla podemos trabajar en las nuevas funciones modificando y creando nuevas funciones.

## Finalización de una rama de función

Cuando se termine de trabajar y se quiera funcionar la rama feature con la develop se ocupa el siguiente comando para finalizarla “git Flow feature finish feature\_branch”



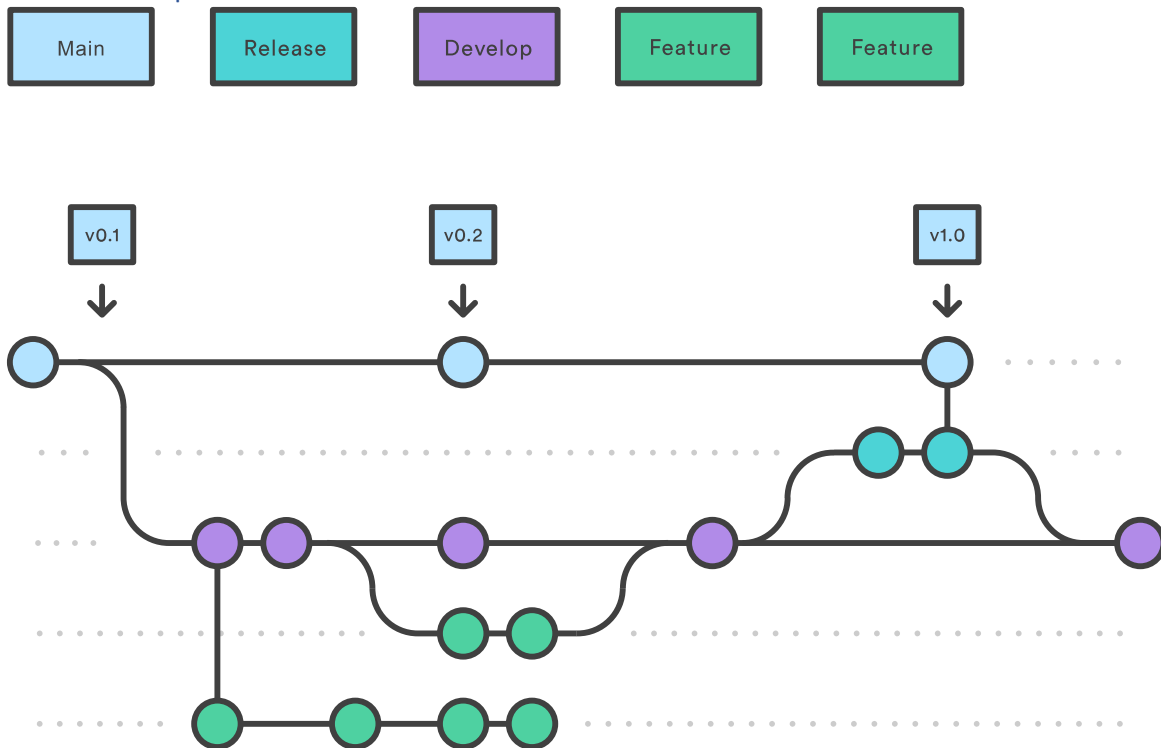
```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git flow feature finish feature_branch
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Already up to date.
Deleted branch feature/feature_branch (was 734caed).

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

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> 
```

## Ramas de publicación



Cuando develop hay adquirido suficientes funciones para una publicación, se debe bifurcar una rama release. Al crear esta rama, se inicia el ciclo de publicación, cuando esta lista para el lanzamiento, la rama release se fusiona en main y se etiqueta con un numero de versión.

Esta rama de feature, se basan en la rama develop:

Para crear esta rama nos deberemos ir a la rama de develop y poner el siguiente comando “git Flow release start 0.1.0 switched to a new Branch ‘release/0.1.0’ ”

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git flow release start 1.1.0
Switched to a new branch 'release/1.1.0'

Summary of actions:
- A new branch 'release/1.1.0' was created, based on 'develop'
- You are now on branch 'release/1.1.0'

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

    git flow release finish '1.1.0'
```



```
PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git branch
develop
master
* release/1.1.0
PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> █
```

Para finalizar la rama se pone el siguiente comando “git Flow release finish ‘numero\_version’”

```
EXPLORER    ...    Get Started    TAG_EDITMSG ●

GITFLOW
└─ .git
   └─ TAG_EDITMSG

.git > TAG_EDITMSG
1  se creo la version 1.1.0
2  #
3  # Write a message for tag:
4  # 1.1.0
5  # Lines starting with '#' will be ignored.
6

PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git branch
develop
master
* release/1.1.0
PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git flow release finish '1.1.0'
Switched to branch 'master'
hint: Waiting for your editor to close the file...
█
```

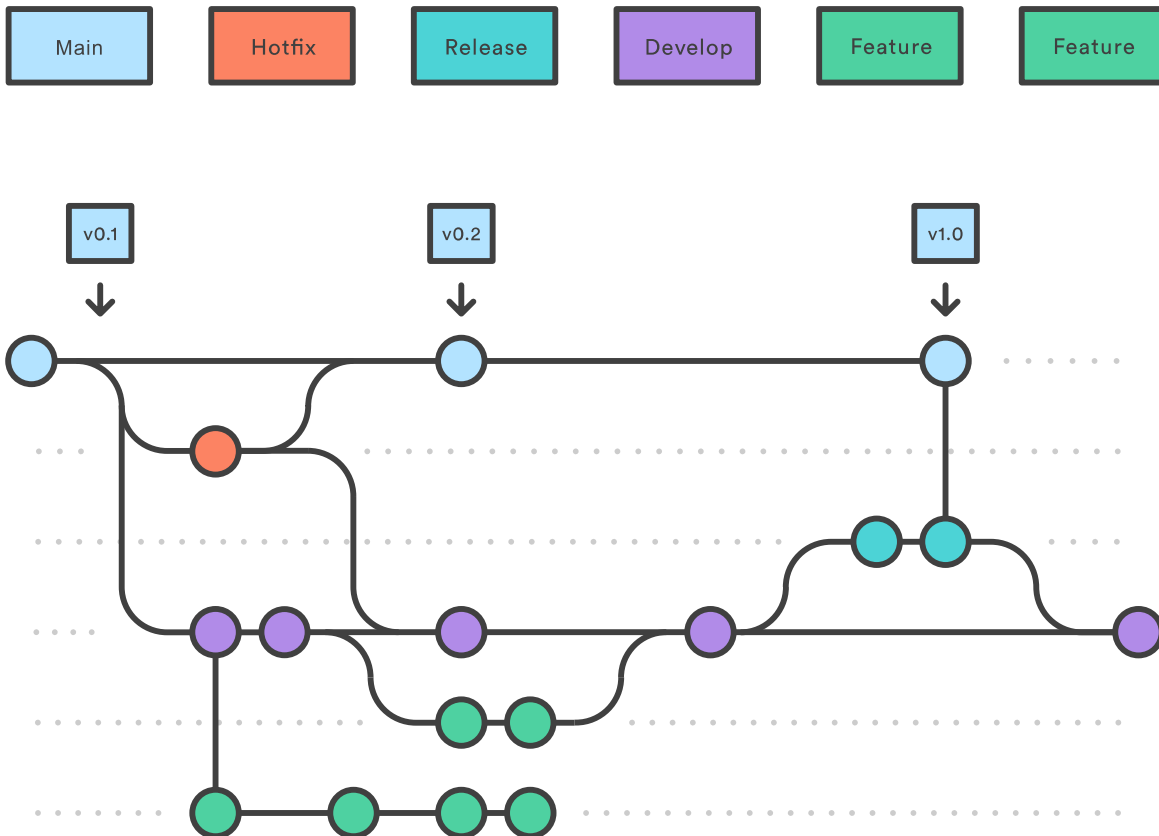
```
PROBLEMS    OUTPUT    TERMINAL    DEBUG CONSOLE

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git branch
develop
master
* release/1.1.0
PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git flow release finish '1.1.0'
Switched to branch 'master'
hint: Waiting for your editor to close the file...
Deleted branch release/1.1.0 (was 734caed).

Summary of actions:
- Release branch 'release/1.1.0' has been merged into 'master'
- The release was tagged '1.1.0'
- Release branch 'release/1.1.0' has been locally deleted
- You are now on branch 'master'

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> █
```

## Ramas de Corrección



Las ramas de mantenimiento, o hotfix sirven para reparar rápidamente las publicaciones de producción. Son muy similares a las ramas de reléase y feature, salvo por el hecho de que se basan en la rama main y no en la develop.

Tener una línea de desarrollo específica para la corrección de errores que permite que el equipo aborde incidencias sin interrumpir el flujo de trabajo.

Para iniciar la rama hotfix se debe ubicarse en la rama main y poner la siguiente línea como se muestra a continuación (el comando es "git Flow hotfix start hotfix\_branch"):

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git branch
develop
* master
PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> 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'

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> █
```

```
PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git branch
develop
* hotfix/hotfix_branch
master
PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> █
```

Para finalizar la rama y fusionarla se pone el siguiente comando “git Flow hotfix finish hotfix\_branch” escribimos algo para la etiqueta un comentario

```
File  Edit  Selection  View  Go  Run  Terminal  Help  • TAG_EDITMSG - GitFlow - Visual Studio Code

EXPLORER  ...  Get Started  TAG_EDITMSG  •
└─ GITFLOW
  └─ .git
    └─ TAG_EDITMSG

.git > TAG_EDITMSG
1  etiqueta para ramas
2  #
3  # Write a message for tag:
4  #   hotfix_branch
5  # Lines starting with '#' will be ignored.
6

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git flow hotfix finish hotfix_branch
hint: Waiting for your editor to close the file...
█
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> git flow hotfix finish hotfix_branch
hint: Waiting for your editor to close the file...
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
Deleted branch hotfix/hotfix_branch (was 734caed).

Summary of actions:
- Hotfix branch 'hotfix/hotfix_branch' has been merged into 'master'
- The hotfix was tagged 'hotfix_branch'
- Hotfix branch 'hotfix/hotfix_branch' has been locally deleted
- You are now on branch 'develop'

PS C:\Users\Ricardo\Desktop\DevSecOps\09-11-2021\GitFlow> █
```

Ahora nos dirá que se eliminó la rama hotfix y que nos encontramos en la rama develop