

# **GIT DEVELOPMENT MODEL**

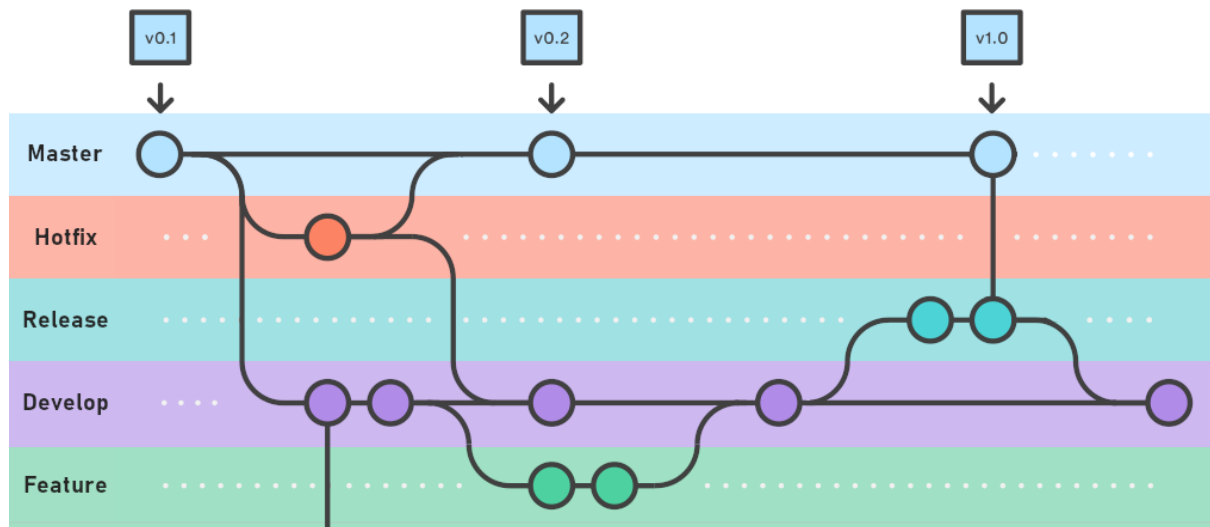
Jorge Aparicio, Alex Duran e Iker Zambrano

# ÍNDICE

|  |   |
|--|---|
| 1.- Esquema de ramas para el desarrollo.....                         | 3 |
| 2.- Donde se encontrará el repositorio principal.....                | 3 |
| 3.- Ramas de los proyectos.....                                      | 4 |
| 4.- Día a día del desarrollo.....                                    | 5 |
| 5.- Cómo marcamos que es una versión nueva, los commits y ramas..... | 5 |

## 1.- Esquema de ramas para el desarrollo.

Seguiremos el esquema de Git Flow.



## 2.- Donde se encontrará el repositorio principal.

El repositorio principal estará en la organización que crearemos en github, se llamará DevGurus, y cada miembro del equipo trabajará sobre el repositorio del proyecto.



### 3.- Ramas de los proyectos

Las ramas consistirán en:

- **La rama develop:**
  - Se crea al inicio del proyecto y se mantendrá durante la totalidad del proceso de desarrollo.
  - Desde esta rama se crearán las ramas para las features, así mismo, estas deberán luego fusionarse con la rama develop.
  - Solo habrá una rama develop.
  - Se pueden hacer pequeños commits a la hora de juntar el trabajo si todos los integrantes están en el mismo equipo
  - Esta es la rama que contiene toda la historia del desarrollo.
- **La rama feature:**
  - Esta rama se creará cuando haya que desarrollar una determinada función / característica.
  - Se creará desde la rama develop y se fusiona también con la rama develop
  - Estas ramas tendrán un inicio y un fin.
  - Habrá tantas ramas como se necesiten o funcionalidades se están desarrollando.
- **La rama master:**
  - Se crea al inicio del proyecto.
  - Solo mantendrá versiones estables de la aplicación.
  - Solo podrán fusionarse a esta rama las ramas hotfixes y releases.
  - Se etiquetará con un número de versión.
  - Solo habrá una rama master.
- **La rama hotfixes:**
  - Se crea a partir de la rama master.
  - Se puede fusionar con la rama master, develop y release.
  - Se utilizará para resolver problemas de producción.
  - Generará una nueva versión en release y master.
  - Habrá las ramas que se necesiten.
- **La rama release:**
  - Se crea a partir de la rama develop.
  - Es la encargada de tener el código limpio y funcional listo para el próximo lanzamiento a producción.
  - Develop se fusiona a esta rama cuando se haya finalizado una funcionalidad.
  - Se fusiona con la rama master y después con la rama develop.
  - En releases solo se pueden generar commits para solucionar problemas de producción y temas relacionados con la documentación.
  - Estas ramas deberán etiquetarse con un número de versión.
  - Solo habrá una rama release.

## 4.- Día a día del desarrollo

Si es un **proyecto nuevo** tendremos que crear en repositorio en la organización con las ramas de **master**, **release** y **develop**. Cuando se vaya avanzando crear las demás ramas.

Si ya estamos **avanzados** en el **desarrollo** tendremos que hacer los siguientes pasos:

- Al empezar el día hacer un **pull** para tener todos los cambios del día anterior.
- Ir desarrollando las funcionalidades o características en las ramas de features.
- Los commit ir **pusheandolos** hacia github.
- Si una funcionalidad ha sido terminada solicitar el pull request para que se revise y apruebe, para su posterior fusión con la rama develop y/o release.
- Si el punto anterior sucede hacer un pull para tener los cambios en los repositorios.
- Al final del día se juntaran todos los cambios.

El encargado de revisar el código y aprobar los pull request seremos cada vez uno.

## 5.- Cómo marcamos que es una versión nueva, los commits y ramas

Hay diferentes casos a la hora de marcar una nueva versión:

- Que se haya realizado un **hotfix**, será una nueva versión para la rama master y release.
- Que se haya terminado el frontend, funcionalidad o añadido un nuevo requisito para la aplicación.
- Se marcaran las versiones con v0.1, v1.0, v1.1, etc...

En el caso de los commits:

- Deberá tener un mensaje descriptivo como "**Cambiado el color para el encabezado y terminado menú de navegación**" o "**Terminado la función de conexión con la base de datos**".
- Se deberán hacer los commits cuando creamos que un cambio merezca la pena cambiarse (serán unos 2 o 3 commits al día).
- Si se quiere volver atrás sabremos por el mensaje del commit hasta donde volver.

En el caso de las ramas:

- Ramas feature: serán nombradas con el nombre de la característica como "**feature-css**" o "**feature-conexionBD**"