



# git

## **GIT DEVELOPMENT MODEL**

Documentación de Control de Versiones Git



### **DESARROLLO DE APLICACIONES WEB** **RETO nº1 - EQUIPO nº3**

*Integrantes: Adrián Pisabarro, Mario Zatón, Tania García.*

---

## **Índice**

1.- Dónde se encontrará el repositorio principal:	2
2.- Qué ramas va a tener y qué normas posee cada rama:	2
3.- Nomenclatura de cara al nombre de las ramas: ¿En que rama trabajo? ¿Qué rama actualizo al final del día de trabajo?	2
4.- Nomenclatura de cara a los commits: ¿Cómo se a cual volver en caso de necesidad?	3
5.- Cómo marcamos qué momentos son una versión nueva.	3
6.- Quién se encarga de ser el administrador: ¿Todos, cada vez uno?	3
7.- Esquema de las ramas para el desarrollo	4

## 1.- Dónde se encontrará el repositorio principal:

El repositorio principal se encontrará en GitHub. Todos tendremos el mismo acceso como colaboradores a ese repositorio para que los commit se hagan cada uno con su cuenta y no haya conflictos.

El repositorio será creado por el coordinador del proyecto y todos seremos contribuidores al repositorio haciendo *commits*, aunque no todos podrán navegar por todas las ramas. Por ejemplo, un contribuidor normal no pasará directamente el proyecto de Develop a Master, lo hará un responsable de rama. Lo mismo si tenemos que hacer algún tipo de *rollback*.

Como programa para gestionar Git, cada uno puede ayudarse de las herramientas que tenga en su ordenador: "GitKraken", "GitHub Desktop", ... Sin embargo, como grupo apoyaremos siempre que se haga con la terminal, para ir cogiendo soltura en comandos Git, que ya hemos practicado en la asignatura: "Despliegue de Aplicaciones Web".

## 2.- Qué ramas va a tener y qué normas posee cada rama:

- Master: rama para las versiones definitivas.
- Develop: rama secundaria para almacenar los cambios diarios. De la rama develop saldrán las siguientes ramas.
  - HTML: rama en la que solo se trabajará el HTML.
  - CSS: a partir del primer guardado del archivo html trabajar con el CSS.
  - JavaScript: se trabajará en el JS a partir de una versión avanzada del proyecto para evitar problemas.
- Características: ramas nombradas a partir de la característica/parte en la que se esté trabajando.
- Hotfix: rama para revisar errores de la rama Master. al corregir el error se actualizará tanto la rama Master como la Develop.

## 3.- Nomenclatura de cara al nombre de las ramas: ¿En que rama trabajo? ¿Qué rama actualizo al final del día de trabajo?

La rama en la que vamos a trabajar normalmente es la rama de características (*features*) y en la rama que vamos a actualizar va a ser la rama secundaria (*develop*) para incluir todo el código terminado.

Para los nombres de las ramas utilizaremos unos nombres descriptivos. Los nombres de las ramas son los indicados en el apartado de [esquema de ramas para desarrollo](#). Si debajo de HTML, CSS, o JavaScript (rama características, *features*) tenemos

que incluir un nombre sería de la siguiente forma: Menú (para identificar que estoy haciendo el css del menú descenderá de la rama css con lo cual sabremos a que parte de “Develop” se refiere).

#### 4.- Nomenclatura de cara a los commits: ¿Cómo se a cual volver en caso de necesidad?

**Feature** -> La nomenclatura a seguir será:

-Utilizar un comentario descriptivo+nombre de la persona que lo ha realizado

Ejemplo: menudesplegable-tania

**Develop** -> La nomenclatura a seguir será:

-En caso de que sea unos cambios pequeños pasaremos del 1.0+descripción al 1.1+descripción

-En caso de realizar cambios grandes pasaremos del 1.0+descripción al 2.0+descripción

Ejemplo: 1.0-html

**Master y Hotfix** -> La nomenclatura a seguir será:

-En caso de cambios pequeños pasaremos del 1.0 al 1.1 y así sucesivamente(1.1.1)

-En caso de que realizaremos cambios grandes pasaremos del 1.0 al 2.0

Ejemplo: 1.0

#### 5.- Cómo marcamos qué momentos son una versión nueva.

Los momentos irán definidos en dos características diferentes:

- Finalización de una tarea específica. Si alguien termina un menú o una característica específica del menú hará un commit en la rama de desarrollo en la que nos encontremos. Cuando terminemos la tarea específica entera la pasará el responsable de esa rama a la rama develop o a la rama feature (css, html, js,...).
- Finalización de las horas de trabajo. Todos haremos un commit el día de trabajo a las 14h con nuestro trabajo tal y como este a nuestra rama en la que estemos desarrollando. Si seguimos en casa al final de nuestro trabajo esté completado o no haremos un commit también a nuestra rama en la que estemos desarrollando.

#### 6.- Quién se encarga de ser el administrador: ¿Todos, cada vez uno?

Cada rama va a tener un responsable, cada uno va a actuar como administrador de su rama correspondiente. En el momento de repartirse cada trabajo, se nombrará a un responsable de su rama. Si dos personas trabajan en una misma rama, tendrá que coordinarse con su responsable de rama.

Únicamente una persona hará commit a la rama “Master”, si en algún caso hay un *bug* en la “Master” se designará a una persona para arreglar ese bug y será automáticamente la responsable del commit y merge de esa rama: “Hotfix” .

## 7.- Esquema de las ramas para el desarrollo

