INFORME TÉCNICO

PSG2 G6-62

Índice de contenidos

1. INTRODUCCIÓN

En esta segunda parte del proyecto vamos a describir la metodología de gestión utilizada por el grupo. Usaremos GitHub para la gestión de tareas. En los siguientes puntos se explicará de forma detallada su estructura y la evolución que hemos tenido hasta ahora.

Figura 1 Icono GitHub

2. ESTRUCTURA DE REPOSITORIOS Y RAMAS POR DEFECTO

La primera estructura que utilizamos para nuestro repositorio fue la más simple, pero a su vez, la que puede dar origen a más problemas. Cada uno de los integrantes del grupo trabajamos sobre la misma rama 'master', lo que pudo ocasionar serios problemas de conflictos. Gracias a que algunos miembros del grupo ya usaron GitHub, pudimos resolver estos problemas de antemano. Para la prevención de estos conflictos decidimos no trabajar en paralelo. De esta forma, pudimos evitar que dos o más trabajos se pisen entre ellos.

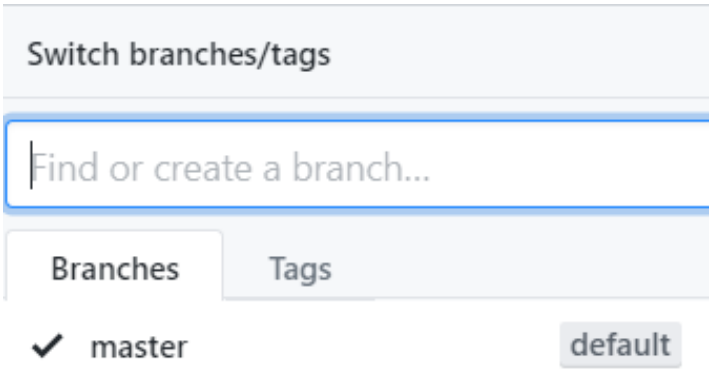

Figura 2 Rama máster

3. ESTRATEGIA DE RAMIFICACIÓN, BASADA EN GIT FLOW

La segunda estructura o metodología que utilizamos fue Git Flow. Esto es una estrategia de ramificación que permite una mejor organización del trabajo y la prevención de conflictos. Esta estructura parte de una rama principal 'master' en la que sale una rama secundaria 'develop'. Cada tarea del proyecto corresponde una rama 'task-#xx'. Estas ramas son creadas a partir de la rama secundaria. Cuando una tarea es acabada se hace un merge hacia la rama secundaria. Esto es, el trabajo de la task-#xx se sube a la rama develop. Así podemos ir uniendo todo el trabajo. Cuando todas las tareas están subidas a develop se comprueba que todo funcione a la perfección. Si todo va bien, entonces podemos proceder al último paso: hacemos un merge de develop a la master.

Con esta estructura podemos trabajar simultáneamente desde diversas ramas. Además, podemos evitar que se nos cuele algún error en la rama principal, el cual puede ser muy engorroso de resolver o que, en el peor de los casos, no se pueda ni resolver.

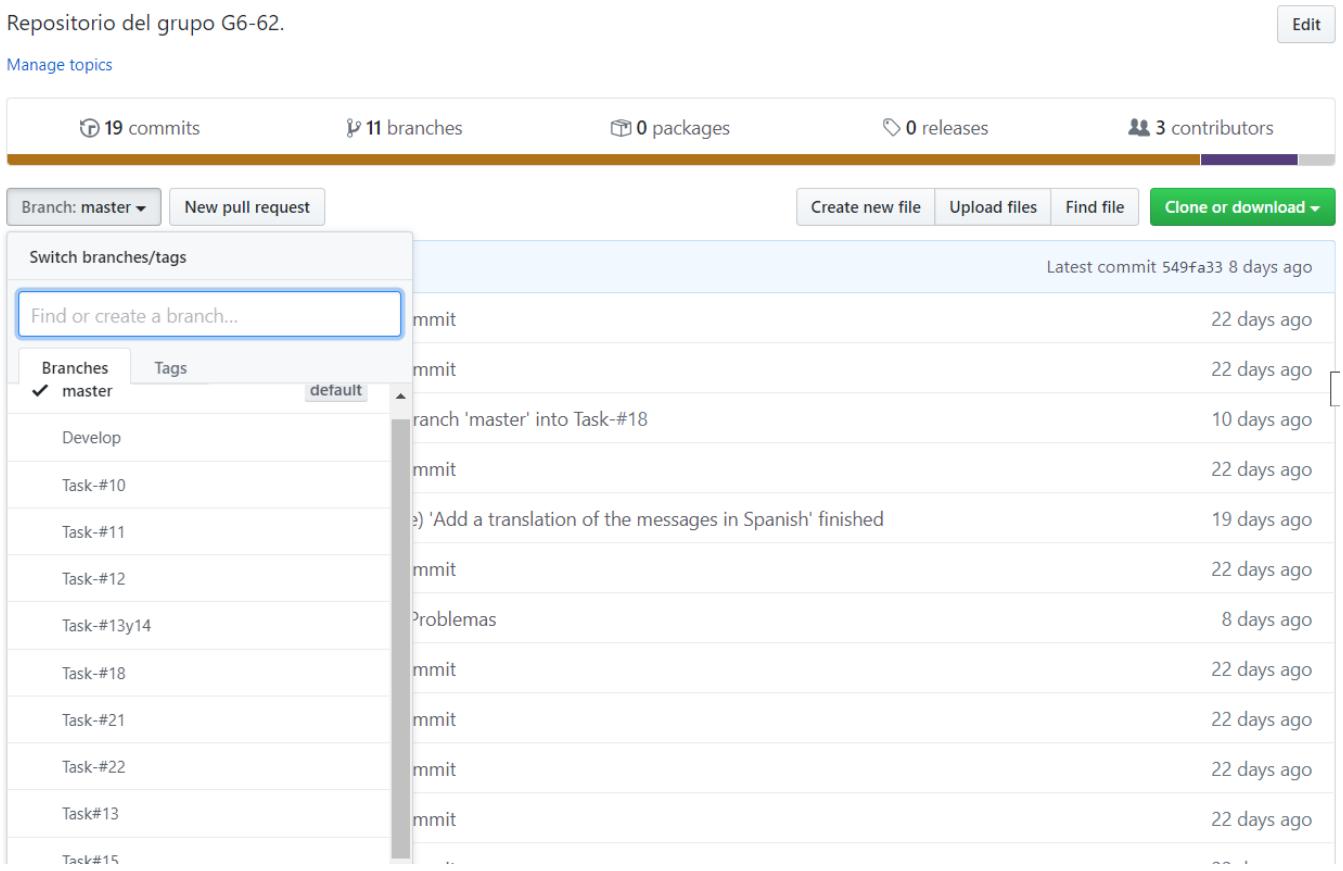

Figura 3 Conjunto de ramas

4. POLÍTICA DE VERSIONES

La primera versión del proyecto correspondería a la primera estructura del proyecto, en la cual usamos una única rama master. En cambio, para la segunda versión ya usaríamos la metodología Git Flow.

5. CONCLUSIONES

En vista a la metodología de gestión, el grupo pudo prevenir los primeros problemas comentados en el segundo punto. Otros problemas que surgieron es que algunos compañeros crearon ramas a partir de la master, en vez de develop. Estos problemas pudimos solventarlos creando otra rama a partir de develop y pasando el trabajo de la primera rama a ésta última. En general, ha habido una buena organización y madurez en la resolución de problemas.