PSG2

PROCESO SOFTWARE Y GESTIÓN II

Informe técnico historial del proyecto II

product owner: JOSÉ ANTONIO PAREJO MAESTRE

Carmen Mª Muñoz Pérez (Scrum Master)

Enrique Salazar Márquez

Javier Martínez Fernández

José Carlos Morales Borreguero

Rafael Ángel Jiménez Fernández

Contenido

[Contenido 2](#_Toc67170109)

### Contenido

En la segunda parte del Sprint S2 se han realizado las tareas de la 11 a la 17. Estas tareas eran las siguientes a realizar según la prioridad que se había establecido en el Sprint Backlog. En primer lugar, se hizo la tarea número 11, ya que esta consistía principalmente en describir la estrategia de ramas que se va a seguir durante todo el proyecto. Esta tarea fue realizada por todos los miembros del grupo, ya que es la base de la forma de trabajar durante todo el proyecto.

Una vez realizada esa tarea, cada miembro del equipo de desarrollo se dedicó a realizar la tarea que le correspondía, para finalmente realizar una primera release del producto.

A continuación, se visualizará una tabla con las tareas que habían sido asignadas a cada desarrollador:

|  |  |
| --- | --- |
| **Desarrollador** | **Tareas** |
| Carmen Mª Muñoz Pérez | 13 |
| Enrique Salazar Márquez | 15 |
| Javier Martínez Fernández | 12 |
| José Carlos Morales Borreguero | 14 |
| Rafael Ángel Jiménez Fernández | 16 |
| Todos los miembros | 11,17 |

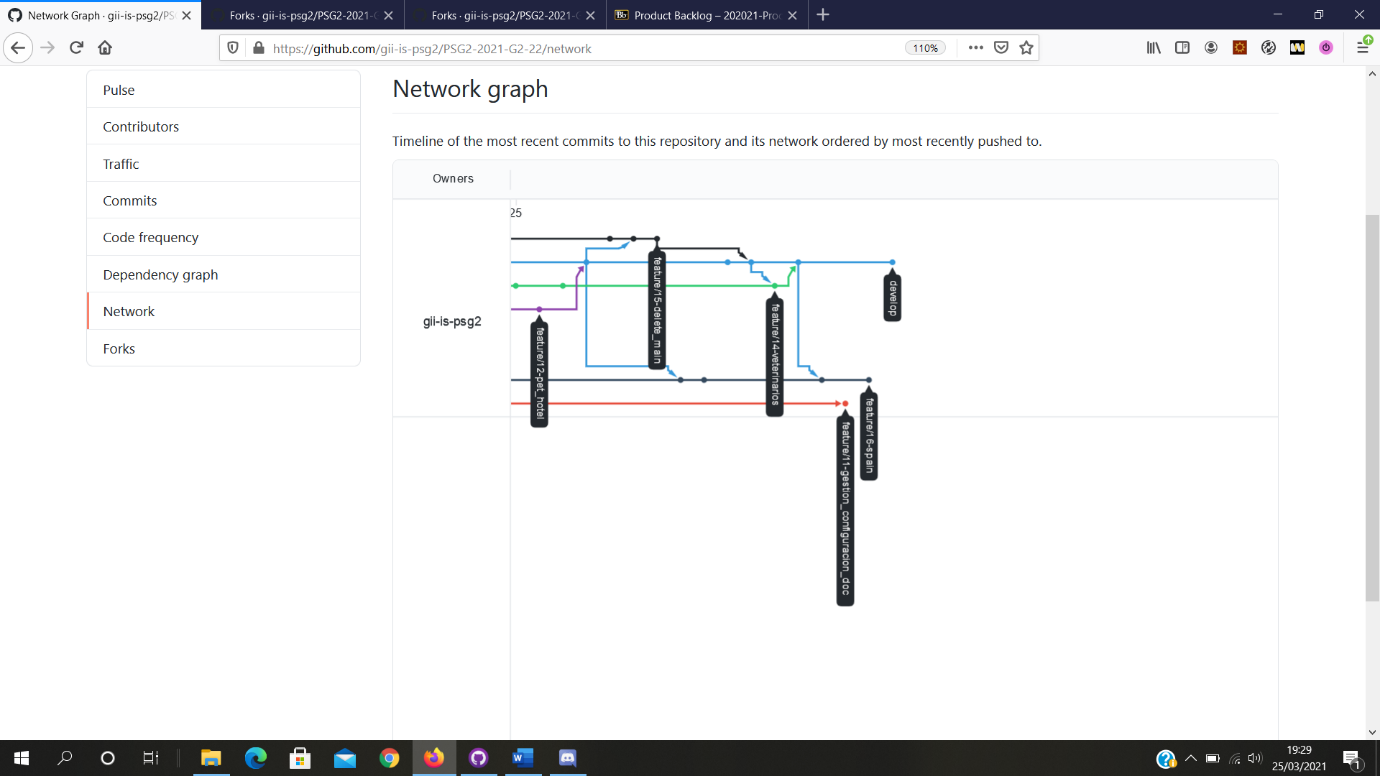
En cuanto a los conflictos, en nuestro caso se han tenido algunos conflictos cuando se ha realizado el merge entre una rama *“develop”* y una rama *“feature”.* Para solucionarlos, se han hecho los cambios pertinentes desde la interfaz de usuario de GitHub directamente desde el navegador o desde Eclipse. Esto se debe a que no eran conflictos graves. Cuando el conflicto se ha dado cuando se actualizaba la rama *“develop”,* los ha solucionado el Scrum Master junto con los implicados. En otro caso, lo ha solucionado la persona que haya actualizado la rama de su funcionalidad.

Finalmente, se mostrará la línea de tiempo de los commits realizados con las ramas correspondientes que se han creado para las tareas del Sprint S2:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Ilustración 2.1 - Flujo de commits**



**Ilustración 2.2 - Flujo de commits**

### ventajas e inconvenientes de la estrategia de ramas

Para el desarrollo del segundo bloque de tareas que se tenían que realizar en el Sprint S2, se ha puesto en práctica la propuesta de una nueva estrategia de ramas: Git Flow. Durante el desarrollo de estas tareas, todos los miembros del grupo se han adaptado a esta forma de trabajar, encontrando así tanto ventajas e inconvenientes.

En primer lugar, se pretenden definir varias de las ventajas que aporta esta estrategia. Esta evita que se produzcan errores muy graves que puedan afectar a todo el proyecto, ya que en esta estrategia si esto ocurre será en una rama secundaria. Así, si se produce un error grave e irreversible esa rama puede ser eliminada sin afectar al proyecto. Además, antes de obtener una rama con todo el sistema en producción, hay otras ramas para terminar de unificar las funcionalidades y solventar otros aspectos. Esto a su vez permite que los errores que se produzcan por una funcionalidad no afecten a las otras que están en desarrollo, evitando además el hecho de tener que estar solventando conflictos constantemente.

Por otro lado, también existen algunas desventajas al usar esta forma de trabajar. La mayor desventaja se produce cuando dos o más personas están trabajando en funcionalidades en las que tienen que tocar varios artefactos comunes, haciendo que los conflictos no se solucionen hasta el final. Estos conflictos pueden llegar a ser realmente graves si alguno hace algún cambio significativo en el artefacto que pueda afectar directamente a las otras funcionalidades.

En definitiva, utilizar una única rama en el desarrollo del proyecto no es buena idea, ya que puede acarrear serios problemas si hay algún error grave. Por otro lado, la estrategia de ramas utilizada tiene grandes ventajas, aunque tiene un inconveniente significativo que hay que tener muy presente para evitar conflictos. Para evitarlos, tiene que haber una buena comunicación entre los miembros del equipo de desarrollo.