

Práctica 1. Git flow

Enunciado

Se dispone de una aplicación que gestiona una librería online. Esta librería ofrece un interfaz web y una API REST para la gestión de libros, con sus correspondientes pruebas unitarias, de API REST y Selenium. Se proporciona el código de dicha aplicación como un zip en el aula virtual.

Se desea introducir una funcionalidad adicional utilizando el modelo de desarrollo git flow. Concretamente, los dueños de la librería nos han pedido que las descripciones de los libros se guarden en la base de datos separada por líneas con un número de caracteres máximo en cada línea. Es decir, como ocurre con un procesador de textos cuando se llega al final de la línea, que se empieza una nueva. Las palabras no se parten, si no cabe la última palabra de la línea, pasa a la línea siguiente. No obstante, si la línea es tan pequeña que una palabra no cabe, esa palabra se partirá. Para que se sepa que la palabra se ha partido, se deberá poner un guión como último carácter de cada línea donde se parte una palabra larga. Además, nos han pedido que hagamos un procesamiento de los espacios de la siguiente forma:

- Sustituir varios espacios consecutivos entre dos palabras por un único espacio.
- No permitir espacios al final de una línea.
- No permitir espacios al principio de una línea.

El cliente no tiene claro todavía qué longitud de línea establecer, así que nos han pedido que la aplicación esté preparada para cualquier tamaño de línea, aunque de momento será de 10. Nunca será menor que 2.

Este servicio de partición de líneas y sus correspondientes tests se proporcionan como dos clases separadas que pueden descargarse del aula virtual. Es decir, no hay que implementarlo. Lo único que hay que hacer es usar este servicio desde el BookService de la aplicación para separar en líneas la descripción del libro antes de ser guardado en la base de datos.

La práctica consiste en **introducir estos cambios en la aplicación utilizando** de forma escrupulosa el modelo de desarrollo **git flow**, teniendo en cuenta las siguientes consideraciones:

- En un primer paso deben crearse las ramas main y develop, y el código inicial tiene que estar en develop, ser liberado, tal como se indica en git flow, y acabar en la rama master. En este momento estamos listos para comenzar la inclusión de la funcionalidad que nos han pedido.
- La funcionalidad deberá incluirse como requiere git flow, y debe seguirse todo el proceso hasta tener una versión liberada en la rama main.
- La funcionalidad se implementará en los siguientes commits:
 1. Inclusión del fichero de tests de linebreaker en el proyecto (estos tests no

- pasarán, dado que no está el código que implementa la funcionalidad, todos los demás tests siguen pasando)
2. Inclusión de los ficheros con la funcionalidad en el proyecto (los tests de linebreaker pasan)
 3. Modificación de BookService para llamar al servicio de linebreaker antes de guardar el objeto. Todos los tests pasan.
 4. Abrir pull request contra la rama que corresponda y mezclar con la rama que corresponda
- Las ramas deben fusionarse utilizando Pull Requests, y cuando estos se fusionan con la rama correspondiente debe usarse la estrategia merge con commit. El comentario del commit indicará qué rama se mezcla con la rama destino.
 - El código de desarrollo, una vez la nueva funcionalidad ha pasado a formar parte de una nueva versión, debe ver incrementado su minor version (1.1.0 → 1.2.0) y la versión liberada no debe contener “SNAPSHOT” en su número de versión en el pom.
 - Al abrir los pull requests todos los tests deben pasar, así como en los commits resultado de mezclar ramas.
 - No se borrará ninguna rama.

Formato de entrega

Toda la práctica se debe desarrollar en un repositorio GitHub privado, cuyo nombre será el del identificador del correo del alumno (sin @urjc.es), al que se invitará al profesor (usuario GitHub: gortazar). Concretamente, este repositorio debe tener las ramas exigidas por el modelo de desarrollo git flow, los cambios realizados deben hacerse siguiendo este modelo, y los profesores evaluarán que el modelo se haya seguido concienzudamente. Por ello, no se deberá borrar ninguna rama.

Adicionalmente, la práctica se entregará por el aula virtual teniendo en cuenta los siguientes aspectos:

- Las prácticas se podrán realizar de forma individual o por parejas. En caso de que la práctica se haga por parejas sólo será entregada por uno de los alumnos.
- La práctica se entregará como un fichero .zip del repositorio GitHub. El nombre del fichero .zip será el usuario URJC del alumno. En caso de dos alumnos, el nombre del zip será el usuario URJC separado por guión (p.perezf-z.gonzalez.zip)
- En el fichero pom.xml se deberá incluir el siguiente nombre del proyecto (donde nombre.alumno corresponde con el identificador del alumno o los alumnos):

`<artifactId>nombre.alumno</artifactId>`