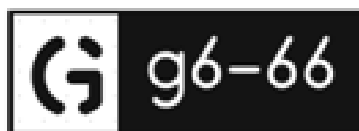


Gestión de la Configuración del proyecto.



Proceso software de Gestión II. Grado en Ingeniería

Informática – Ingeniería del Software

Curso 2019 - 2020

Grupo de prácticas	66
Autores	Rol
Andrés Aguilar Alhama	Scrum Master
Alejandro González Martín	Director de proyecto
Francisco José Perejón Barrios	Equipo de desarrollo
Francisco José Quintela Vela	Equipo de desarrollo
Ramón Javier Morales Pacheco	Equipo de desarrollo

Contenido

Introducción.....	3
Estructura del Repositorio.	3
Ramas Principales.	3
Rama Master.....	3
Ramas de Funcionalidad.	3
Ramas de Mantenimiento.....	4
Ramas de Lanzamiento.	4
Política de Versiones.....	4
Estrategia de Lanzamiento.....	4

Introducción.

En el contexto de la asignatura Proceso y Sistemas de Gestión II, se requiere definir la estrategia a seguir a la hora de manejar el código fuente del proyecto asignado a nuestro grupo (G6-66). Dicha estrategia deberá componerse al menos de:

- Estructura del repositorio y sus ramas principales.
- Estrategia de Branching, basada en GitFlow.
- Política de Versiones.

Estructura del Repositorio.

El repositorio del grupo G6-66 emplea la tecnología de repositorios Git y se encuentra alojado en Github, dentro de la organización “gii-is-psg2”: <https://github.com/gii-is-psg2/PSG2-1920-G6-66>

Este es el único repositorio “canónico”, sin embargo, no negamos la posibilidad de realizar un fork a cada desarrollador del grupo si lo viese necesario (Aunque seguramente no proceda para este proyecto, por tanto, no se especificarán las instrucciones para hacerlo). El repositorio adjunto al enlace del párrafo anterior es un duplicado del proyecto de la misma organización: “spring-petclinic” con acceso limitado a los usuarios del ya citado grupo, añadir que todos los desarrolladores del grupo poseen derechos de Management.

Ramas Principales.

Rama Master.

La rama por defecto y que contendrá los últimos incrementos realizados a las funciones del proyecto será “**master**”, siendo esta la **única rama que será continuamente actualizada durante el desarrollo**. Dicha rama también será directamente actualizada con las actas de reuniones u otros documentos relevantes directamente, sin tener que recurrir a una rama. Esta rama será el “padre” de las ramas de funcionalidad.

Ramas de Funcionalidad.

Estas ramas **corresponderán a las tareas principales asignadas (A.X.X.X) siempre y cuando dichas tareas impliquen la implementación de nuevas funciones en el proyecto**. Estas ramas también se emplearán para el bug-fixing. Se nombrarán siguiendo la siguiente formula: **A.X.X.X-[placeholder]**; donde **[placeholder]** corresponde a como máximo dos palabras (Separadas por un guión) que estén relacionadas con la rama a implementar.

Una excepción a esta convención de nombrado son las ramas que denominaremos personales: En caso de que un desarrollador comparta una tarea principal con otro, podrá elegir crear una rama para trabajar en su parte, cuando esté lista dicha rama será mezclada a la rama de funcionalidad principal. No se establece ninguna convención de nombrado para estas ramas, y la responsabilidad de mezclarlas corresponde a la persona que creó la rama.

Estás ramas siempre procederán de “master” y serán mezcladas en las ramas de lanzamiento correspondiente.

Ramas de Mantenimiento.

Las **Ramas de Mantenimiento (L.X.X)** contendrán **una porción de un sprint, y se emplearán a modo de backup**. Dichas ramas **no serán modificadas una vez finalizadas** y serán una copia de master al final de esa porción del sprint. En caso de que sea necesario hacer un bug-fix en la release de un sprint anterior, se recurrirá a estas ramas (Aunque sea prácticamente irrelevante en este proyecto).

Ramas de Lanzamiento.

Las **Ramas de Lanzamiento (LX ó X.X.X)** corresponderán al **estado del proyecto al final de un Sprint**. Estas ramas sólo son el paso previo a añadir una nueva versión a master, el propósito de estas ramas es la realización de pruebas antes del lanzamiento de la nueva versión.

Una vez revisada, esta rama será mezclada con Master para realizar el incremento del Sprint.

Política de Versiones.

Corresponderán al **Sprint en el que se encuentre el proyecto**, por ejemplo: Si nos encontramos en el Sprint L2, la versión será 2.0.0. Cada **nuevo sprint incrementará la versión principal del producto**.

El solventado de bugs podrá ser nombrado con un incremento de X.0.{+1}.

Estrategia de Lanzamiento.

Un nuevo lanzamiento consiste en **crear una nueva Rama de Lanzamiento a partir de Master**. Esta rama es temporal y su versión no recibirá un incremento salvo en caso de que un bug crítico sea arreglado.

Las **Ramas de Funcionalidad que correspondan a funcionalidades del lanzamiento serán mezcladas en la rama de la versión correspondiente**. Una vez añadidas todas las Ramas de Funcionalidad correspondientes al incremento de la versión en la que se está trabajando, se **procederá a probar dichas funcionalidades** y, en caso de que los resultados sean satisfactorios se mezclarán en master; en caso de que sea necesario arreglar un bug se creará una **Rama de Funcionalidad** para ello y se repetirá el ciclo. Si el bug se considera lo suficientemente crítico, se realizará un pequeño incremento en la versión.