



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicaciones
Máster Oficial en Ingeniería Informática

Curso 2020/2021

GESTIÓN DE LA CONFIGURACIÓN Y SEGUIMIENTO DE PROYECTOS

Planificación y Gestión de Proyectos Informáticos

Breve descripción

Desarrollar un conjunto de documentos que permita llevar a cabo de forma organizada todo el proceso de gestión de la configuración del proyecto.

Autores

Álvaro de la Flor Bonilla y Antonio Manuel Salvat Pérez

Propiedad Intelectual

Universidad de Granada



ÍNDICE DEL PROYECTO

No se encuentran elementos de tabla de ilustraciones.¡Error! Marcador no definido.

1	Identificación de los CSCI.....	3
1.1	Ejecutables	3
1.2	Código Fuente	3
1.3	Modelos de datos	4
1.4	Modelos de procesos.....	4
1.5	Especificación de requisitos.....	5
1.6	Pruebas	6
2	Generación de la estructura de directorios del proyecto.....	6
3	tipo de sistema de control de versiones.....	7
4	GIT workflow	8
5	política de reuniones	9
6	Lista de aspectos a analizar	10



ÍNDICE DE ILUSTRACIONES

Ilustración 1 - Ramas principales GitFlow	8
Ilustración 2 - Ramas completas GitFlow	9

1 IDENTIFICACIÓN DE LOS CSCI

1.1 Ejecutables

1-

Nombre: Ejecutable para sistema móvil

Versión: 1.0.0.0

Estado: En desarrollo

Localización: Repositorio del proyecto

2-

Nombre: Ejecutable para sistema de comunicaciones

Versión: 1.0.0.0

Estado: En desarrollo

Localización: Repositorio del proyecto

3-

Nombre: Ejecutable para versión web

Versión: 1.0.0.0

Estado: En desarrollo

Localización: Repositorio del proyecto

1.2 Código Fuente

1-

Nombre: Código fuente de la versión móvil

Versión: 2.0.0.0

Estado: En desarrollo

Localización: Repositorio del proyecto

2-

Nombre: Código fuente de la versión web del personal sanitario

Versión: 2.0.0.0

Estado: En desarrollo

Localización: Repositorio del proyecto

3-

Nombre: Código fuente del sensor físico



Versión: 2.0.0.0

Estado: En desarrollo

Localización: Repositorio del proyecto

4-

Nombre: Código fuente del sistema de comunicaciones

Versión: 2.0.0.0

Estado: En desarrollo

Localización: Repositorio del proyecto

1.3 Modelos de datos

1-

Nombre: Paciente

Versión: 3.0.1.2

Estado: Finalizado

Localización: Documentación del proyecto

2-

Nombre: Personal Médico

Versión: 3.0.1.2

Estado: Finalizado

Localización: Documentación del proyecto

3-

Nombre: Sensor

Versión: 3.0.1.2

Estado: Finalizado

Localización: Documentación del proyecto

1.4 Modelos de procesos

1-

Nombre: Planificación

Versión: 3.0.1.2

Estado: Finalizado

Localización: Documentación del proyecto



2-

Nombre: Modelado

Versión: 3.0.1.2

Estado: Finalizado

Localización: Documentación del proyecto

3-

Nombre: Desarrollo

Versión: 3.0.1.2

Estado: Finalizado

Localización: Documentación del proyecto

4-

Nombre: Despliegue

Versión: 3.0.1.2

Estado: Finalizado

Localización: Documentación del proyecto

5-

Nombre: Mantenimiento

Versión: 3.0.1.2

Estado: Finalizado

Localización: Documentación del proyecto

1.5 Especificación de requisitos

1-

Nombre: Requisitos generales del sistema

Versión: 2.0.0.0

Estado: Finalizado

Localización: Documentación del proyecto

2-

Nombre: Casos de uso del sistema

Versión: 2.0.0.0

Estado: Finalizado

Localización: Documentación del proyecto



3-

Nombre: Requisitos funcionales

Versión: 2.0.0.0

Estado: Finalizado

Localización: Documentación del proyecto

4-

Nombre: Requisitos no funcionales

Versión: 2.0.0.0

Estado: Finalizado

Localización: Documentación del proyecto

1.6 Pruebas

1-

Nombre: Pruebas unitarias

Versión: 0.0.0.7

Estado: En desarrollo

Localización: Repositorio del proyecto

2-

Nombre: Pruebas de integración

Versión: 0.0.0.7

Estado: En desarrollo

Localización: Repositorio del proyecto

3-

Nombre: Pruebas de aceptación

Versión: 0.0.0.7

Estado: En desarrollo

Localización: Repositorio del proyecto

2 GENERACIÓN DE LA ESTRUCTURA DE DIRECTORIOS DEL PROYECTO

Proyecto:

Documentación:

Modelos de datos

Modelos de procesos

Especificación de requisitos

Código Fuente:

Desarrollo:

Sistema móvil

Sistema web

Sensor físico

Sistema de comunicaciones

Pruebas:

Pruebas unitarias

Sistema móvil

Sistema web

Sensor físico

Sistema de comunicaciones

Pruebas de integración

Sistema móvil

Sistema web

Sensor físico

Sistema de comunicaciones

Pruebas de aceptación

Sistema móvil

Sistema web

Sensor físico

Sistema de comunicaciones

3 TIPO DE SISTEMA DE CONTROL DE VERSIONES

Fuente: <http://www.juntadeandalucia.es/servicios/madeja/contenido/libro-pautas/148>

Para el sistema de control de versiones usaremos el software de control de versiones apoyados por la plataforma github.

Para la política de versionado usaremos la propuesta dada por el Marco de Desarrollo de la Junta de Andalucía el cual otorga a una versión 4 números por separado con el siguiente orden mayor-menor-revision-entrega que siguen la siguiente estructura:

- **major:** indica la versión principal del software, consiste en un conjunto de funcionalidades concretas que son cubiertas en dicha versión.
- **minor:** indican funcionalidad menor cubierta en la versión de software entregada.
- **revision:** se modifican cuando hay revisiones de código ante fallos de la aplicación.
- **entrega:** tiene el objetivo de llevar la cuenta del número de veces que una entrega se rechaza.

A su vez se variarán estos valores en los siguientes casos:

- **major:** nuevas funcionalidades claves de la aplicación respecto a la versión anterior debido a la inclusión de nuevos requerimientos para el sistema.
- **minor:** cambios significativos en la forma en la que se ofrece la funcionalidad existente, corrección de grandes fallos del sistema o nuevas versiones que modifican significativamente la funcionalidad ofrecida.
- **revision:** se modifica por cada entrega de software que se realice.
- **entrega:** al rechazarse una entrega se incrementa este contador en la siguiente. Cuando la entrega se aceptase se crearía un tag público que solo conservaría los tres primeros dígitos (mayor, minor, revision).

4 GIT WORKFLOW

Fuente: <https://cleventy.com/que-es-git-flow-y-como-funciona/>

Para el workflow nos basaremos en la propuesta dada por gitflow donde se cuentan con dos ramas principales las cuales son:

- **Master:** Contiene cada una de las versiones estables del proyecto. Cualquier commit que subamos en esta rama debe estar listo para que se pueda incluir en producción.
- **Develop:** Contiene el código de desarrollo de la siguiente versión planificada del proyecto. En ella se incluirán cada una de las nuevas características que se desarrollen.

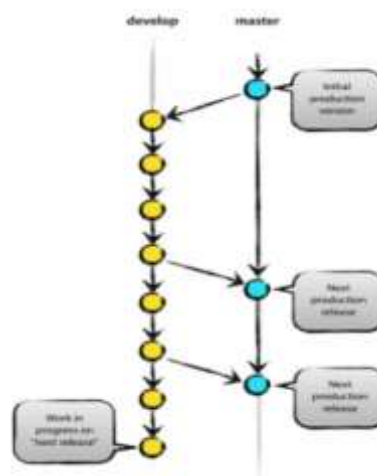


Ilustración 1 - Ramas principales GitFlow

A su vez gitflow también nos propone una serie de ramas de apoyo:

- Feature: Estas ramas tienen que surgir de la rama develop. Cada una de estas ramas almacenan código de desarrollo con nuevas características. Típicamente existen solamente en los repositorios locales de los desarrolladores y no en el repositorio origen. Una vez terminado su desarrollo, se incorporarán nuevamente a la rama develop.
- Release: Como las ramas feature, las ramas release también tienen que surgir de la rama develop. Contienen el código de la versión que se va a liberar próximamente. En ella se incluye todo el código de develop necesario para el lanzamiento. Una vez finalizada la rama, esta se debe incluir tanto en la rama develop como en la rama master.
- Hotfix: Estas ramas surgen de la rama master. Contienen una versión de producción con un error que se desea arreglar urgentemente. Una vez arreglado el error, se incluye el contenido de esta rama en las ramas master y develop para solventar el error.

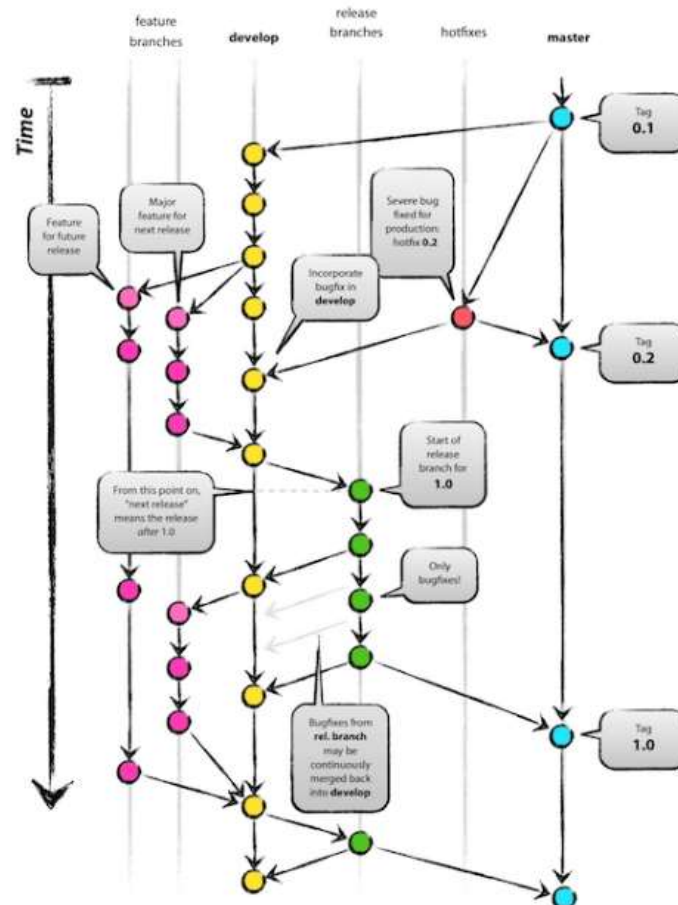


Ilustración 2 - Ramas completas GitFlow

5 POLÍTICA DE REUNIONES

Fuente: <https://www2.deloitte.com/es/es/pages/technology/articles/ceremonias-scrum.html>



Al usar la metodología SCRUM para el proyecto nos basaremos en su política de reuniones.

- **Sprint Planning:** El Sprint Planning es una reunión que se realiza al comienzo de cada Sprint donde participa el equipo Scrum al completo; sirve para inspeccionar el Backlog del Producto (Product Backlog) y que el equipo de desarrollo seleccione los Product Backlog Items en los que va a trabajar durante el siguiente Sprint. El product owner presenta el Product Backlog actualizado.
- **Daily Scrums:** Es una reunión diaria de 15 minutos en la que participa exclusivamente el Development Team. En esta reunión el equipo de desarrollo responderá a las siguientes cuestiones:
 - 1- ¿Qué hice ayer para contribuir al Sprint Goal?
 - 2- ¿Qué voy a hacer hoy para contribuir al Sprint Goal?
 - 3- ¿Tengo algún impedimento que me impida entregar?
- **Sprint Review:** El Sprint Review es la reunión que ocurre al final del Sprint, donde el product owner y el Development Team presentan a los stakeholders el incremento terminado para su inspección y adaptación correspondientes. En esta reunión organizada por el product owner se estudia cuál es la situación y se actualiza el Product Backlog con las nuevas condiciones que puedan afectar al negocio.
- **Sprint Retrospective:** La retrospectiva ocurre al final del Sprint, justo después del Sprint Review. En algunos casos y por comodidad de los equipos, se realiza conjuntamente con el Sprint Planning, siendo la retrospectiva la parte inicial de la reunión. El objetivo de la retrospectiva es hacer de reflexión sobre el último Sprint e identificar posibles mejoras para el próximo.

6 LISTA DE ASPECTOS A ANALIZAR

Una vez se realice el proyecto la principal tarea sería la del mantenimiento de un sistema, ya que un proyecto de desarrollo software no acaba cuando se termina el desarrollo, sino que hay que prolongar la tarea de mantenimiento ya que puede haber futuros errores que solventar o futuras actualizaciones.

Un aspecto importante es analizar el rendimiento del equipo, sobre todo a nivel individual que tal rinde cada miembro del equipo en tipos de tareas específicas.

Otro aspecto sería analizar que tal ha sido la comunicación entre los miembros del equipo, para hacer futuras mejoras como insertar ejercicios que mejoren el 'team-building'.