ESTÁNDARES DE SISTEMA DE CONTROL DE VERSIONES GIT

SOSP

Julian David Lasso Aguilar

Juan Pablo Coronado

Oscar David Rodriguez

Ivan Felipe Torres

Abril 2017.

Servicio Nacional de Aprendizaje.

Electricidad Electrónica y Telecomunicaciones.

Análisis y Desarrollo de Sistemas de Información

**Tabla de Contenidos**

[Objetivo 1](#_Toc481356443)

[Manejo de branchs 1](#_Toc481356444)

[Manejo de carpetas 1](#_Toc481356445)

[Sintaxis commits 1](#_Toc481356446)

[Manejo de tags (Versiones) 2](#_Toc481356447)

# Objetivo

Establecer las pautas y normas para aplicar en el manejo del SCV de nuestro Proyecto formativo

# Manejo de branchs

Los nombres de las branchs serán en inglés, se harán con mayúscula inicial

El proyecto se va a trabajar sobre una rama principal llamada “Master”, y sobre esta se ubicarán las versiones funcionales del proyecto; habrá una sub-rama llamada “Development” sobre la cual se harán ramificaciones por cada sprint del SCRU, y sobre estas, las demás ramificaciones necesarias, en las cuáles se llevarán a cabo las tareas y modificaciones del proyecto

# Manejo de carpetas

Los nombres de las carpetas serán en inglés, se harán con mayúscula inicial.

Las carpetas estarán numeradas en orden ascendente

* Ejemplo carpetas: “01\_Analysis” , “02\_Design”

# Sintaxis commits

El mensaje del commit estará compuesto de 3 partes separadas por una línea en blanco: el título, el cuerpo (opcional) y un pie.

Type / Tipo

El tipo es contenido en el título y puede ser de alguno de los siguientes casos:

feat: Una nueva característica.

fix: Se solucionó un bug.

docs: Se realizaron cambios en la documentación.

style: Se aplicó formato, comas y puntos faltantes, etc; Sin cambios en el código.

refactor: Refactorización del código en producción.

test: Se añadieron pruebas, refactorización de pruebas; Sin cambios en el código.

chore: Actualización de tareas de build, configuración del admin. de paquetes; Sin cambios en el código.

**Subject / Asunto**

El asunto no debe contener más de 50 caracteres, debe iniciar con una letra mayúscula y no terminar con un punto. Debemos ser imperativos al momento de redactar nuestro commit, es decir hay que ser objetivos y muy importante tenemos que acostumbrarnos a escribirlos en **Ingles** esto es una de las mejores prácticas que podemos tener en nuestros commits.

Cuando hablo de ser imperativos hago referencia a este sencillo ejemplo: usar **change** en lugar de "changed" o "changes".

**Body / Cuerpo**

No todos los commits son lo suficientemente complejos como para necesitar de un cuerpo, sin embargo es opcional y se usan en caso de que el commit requiera una explicación y contexto. Utilizamos el cuerpo para explicar el ¿Qué y Porque? de un commit y no el ¿Cómo? Al escribir el cuerpo, requerimos de una línea en blanco entre el título y el cuerpo, además debemos limitar la longitud de cada línea a no más de 72 caracteres.

**Footer / Pie**

El pie es opcional al igual que el cuerpo, pero este es usado para el seguimiento de los IDs de las tareas y las historias de usuario con incidencias.

Ejemplo:

* Resolve: #110 (Este indicará la tarea o tareas que se cierran con el commit)
* See also: #105 (Este indicará si hay historias de usuario padres de las tareas cerradas en el commit)

# Manejo de tags (Versiones)

Los tags se harán sobre la rama “master”, se harán cada vez que haya una versión estable y funcional del aplicativo, tendrán una numeración ascendente y estarán compuestos de 3 partes separadas por puntos como se muestra a continuación: V 0.0.1, donde la numeración de color verde indicará “revisiones” que son modificaciones o correcciones de fallos de la aplicación; la numeración de color azul indica cambios sobre la forma en que se presentan las funcionalidades “menores” , correcciones significativas o nuevas funcionalidades a presentar; y la numeración de color rojo hará referencia a la versión principal o “mayor” del software, que será el conjunto de las funcionalidades cubiertas en tal versión