

# TRABAJO PRÁCTICO N° 1

Ingeniería del Software

## PLAN DE MANEJO DE LAS CONFIGURACIONES

**INTEGRANTES:** Candotti, Enzo  
Flores, Mauricio  
Nicolaide, Christian

## HISTORIAL DE REVISIONES

Versión	Fecha	Observaciones	Autor
1.0.0	02/04/2016	Armado inicial del documento para trabajar.	Candotti, Enzo
1.1.0Draft	03/04/2016	Reestructuración del CM Plan con mayores detalles.	Flores, Mauricio
1.1.0	07/04	Versión final para producción.	Nicolaide, Christian

# **ÍNDICE DE CONTENIDOS**

## **1.0 Introducción**

- 1.1 Propósito y alcance
- 1.2 Propósito del Manejo de la Configuración del Software (SCM)
- 1.3 Referencias, Abreviaturas y Glosario
- 1.4 Herramientas de Gestión de Configuración
- 1.5 Conceptos Básicos

## **2.0 Roles de la Administración de la Configuración**

- 2.1 Administración de la Configuración del Proyecto
- 2.2 Esquema de Directorios
- 2.3 Normas de Etiquetado y Versionado de Archivos
- 2.4 Plan de Esquema de ramas
- 2.5 Plan de Fusión de Archivos (Merge)
- 2.6 Forma de Entrega de Código final Entregable (Release)
- 2.7 Comité de Control de Cambios (CCC)
- 2.8 Integrantes del Equipo, Roles y reuniones
- 2.9 Herramienta de seguimiento de BUGS

## 1.0 INTRODUCCIÓN

### 1.1 Propósito y alcance

Este documento trata sobre el Manejo del Plan de Configuración, la intención de este documento, es describir la forma de trabajo, los documentos, el hardware, software y las herramientas utilizadas en este proyecto.

Tratamos por sobre todas las cosas de utilizar herramientas OpenSource.

SCM es el proceso mediante el cual se identifican los métodos y herramientas para controlar el software a lo largo de su desarrollo y utilización.

### 1.2 Propósito del Manejo de la Configuración del Software (SCM)

- Asegurar la consistencia de la información poniendo en práctica la SCM.
- Definir las personas que le dan soporte a las prácticas de SCM.
- Mantener la integridad a lo largo de todo el ciclo de vida del producto.
- Informar a grupos y las personas el estado del proyecto.
- Crear un historial del estado anterior y actual del proyecto.
- Mejora de Procesos.

### 1.3 Referencias, Abreviaturas y Glosario

Abreviatura	Descripción
SCM	Gestión de la Configuración del Software (Software Configuration Management)
SVN	Sistema de control de versiones
CCC	Comité de Control de Cambios
SCI	Software Configuration item
SCMer	Rol encargado de realizar la tarea de gestión de la configuración (SCM).

### 1.4 Herramientas de Gestión de Configuración

Referencia	Propósito	Link
Netbeans IDE	Entorno de desarrollo	<a href="https://netbeans.org/">https://netbeans.org/</a>
Java	Lenguaje de programación	<a href="https://www.java.com">https://www.java.com</a>
Git	Software de Manejo de Versiones	<a href="https://git-scm.com/">https://git-scm.com/</a>

GitHub	Sistema de Control de Versiones	<a href="https://github.com/">https://github.com/</a>
Git Issues	Sistema de seguimiento de errores	<a href="https://github.com/issues">https://github.com/issues</a>
Travis	Herramienta de integración continua.	<a href="https://travis-ci.org/">https://travis-ci.org/</a>
Gradle	Herramienta de automatización	<a href="http://gradle.org/">http://gradle.org/</a>

## 1.5 Conceptos básicos:

**Repositorio:** Espacio físico (directorio y grupo de directorios), donde se almacenan, una vez terminados, todos los elementos generados durante el proceso de desarrollo en sus diferentes versiones. Estos elementos pueden estar en tres estados: Pendientes de aprobar por el área de SQA, Aprobados por SQA, y en Producción.

**SCI:** Software Configuration Item, Elemento de Configuración del Software. Son los elementos creados durante el proceso. Pueden ser de tres tipos:

- De Software: código fuente, recursos gráficos, bibliotecas, ejecutables
- Documentos: técnicos, administrativos y de usuario
- Estructuras de datos: estructura de base de datos, datos iniciales, archivos de configuración, etc.

**Check Out :**Tomar un SCI del repositorio y copiarlo en un área de trabajo, dejando bloqueado el SCI en el repositorio, para que nadie más pueda hacer un check out del mismo.

**Check In :**Si se trata de un SCI nuevo, ingresarlo en el repositorio y dejarlo disponible para posteriores modificaciones, mediante el mecanismo de check out y check in. Si se trata de un SCI existente al que se le hizo check out, el check in implica el ingreso al repositorio de una nueva versión del SCI, dejando al SCI disponible, liberando el bloqueo que registrado en el momento del check out.

## 2.0 ROLES DE LA ADMINISTRACIÓN DE LA CONFIGURACIÓN

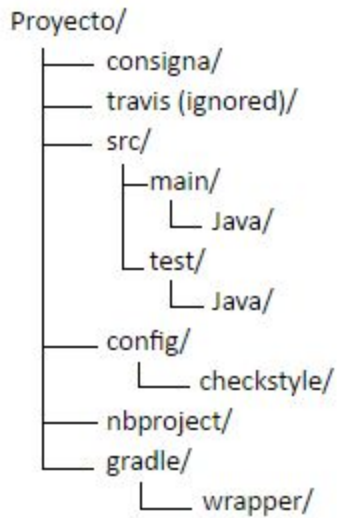
### 2.1 Administración de la Configuración del Proyecto

La Administración de la configuración del proyecto está a cargo del "Global PCM", él es el responsable de actividades como seguimiento de las herramientas, creación de los nuevos branches, creación del reléase, etc.

ROL	Responsable Principal	Responsable
GLOBAL PCM	Christian Nicolaide	Candotti Enzo

Rol	Responsabilidades
GPCM	<p>Posee toda responsabilidad sobre todos los CI.</p> <p>Responsabilidad en la creación de branches y administración de sus políticas.</p> <p>Responsabilidad y asistencia sobre el etiquetado y lanzamiento de branches.</p> <p>Coordinar actividades del CM en el proyecto.</p> <p>Asegurar la correcta ejecución del esquema del CM.</p> <p>Participación en auditorias.</p> <p>Analizar todas las novedades relacionadas al CM.</p>
TPCM	<p>Asistencia en la creación de tags y branches.</p> <p>Creación de actividades para el equipo específico en branches.</p> <p>Garantizar la integridad del producto y el seguimiento de los elementos de configuración propios del equipo.</p> <p>Participación en auditorias.</p> <p>Analizar todas las novedades relacionadas al CM.</p>
Team	<p>Ayudar a resolver conflictos durante la actividad de merge.</p> <p>Asegurarse que los criterios de calidad de los entregables a la rama principal se cumplan.</p> <p>Seguir todos los procesos asociados, políticas y prácticas definidas por sus roles asignados.</p>

## 2.2 Esquema de Directorios



Ruta	Propósito
.\	Directorio Principal, incluye todo el repositorio
.\consigna\	Contiene la consigna del presente trabajo a desarrollar
.\src\	Directorio del programa.
.\nbproject\	Carpeta creada al utilizar NetBeans.

## 2.3 Normas de Etiquetado y Versionado de Archivos

Para nombramiento de los archivos vamos a utilizar la siguiente notación:

**mayor.medio.menor**

Descripción

**Mayor:** Cambios mayores al código

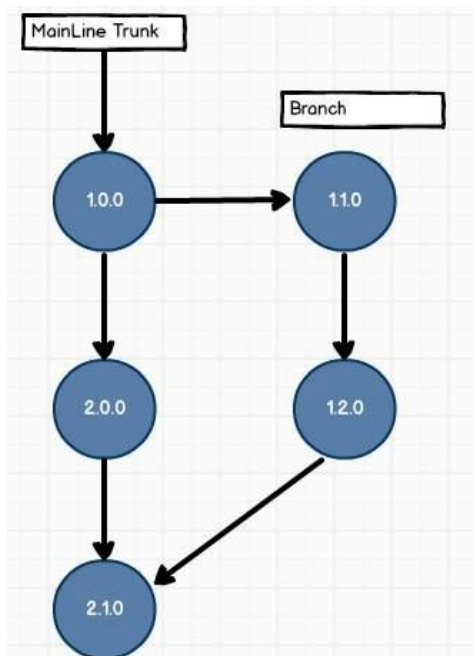
**Medio:** Cambios medios al código

**Menor:** Cambios menores al código

## 2.4 Plan de Esquema de ramas

Utilizamos una subdirectorio principal de desarrollo (trunk) en este lugar se encuentra el código principal, cada desarrollador debe solucionar sus bugs antes de subir a este subdirectorio su código.

Con respecto a funcionalidades que suponían mayor trabajo de desarrollo se crearon en un branch etiquetado siguiendo nuestra política de versionado.



## 2.5 Plan de Fusión de Archivos (Merge)

En nuestro Trunk mantenemos el código principal, pero en el caso de alguna funcionalidad extra que tenía que ser desarrollada en simultáneo por otros desarrolladores, creamos una rama para su posterior Fusión.

Para fusionar esta rama con el código principal nuevamente, el administrador realiza la fusión haciendo un rebase del código **junto a los desarrolladores**, de esta forma rápidamente nos damos cuenta si hay errores y los desarrolladores pueden trabajar en resolverlos y el administrador está al tanto de la situación para coordinar otro eventual cambio en el código.

En caso de que la fusión se realice de forma correcta, se procede a un etiquetado de acuerdo a la política de versionados.

## 2.6 Forma de Entrega de Código final Entregable (Release)

Una vez terminado el código, corregidos todos los errores del mismo, el código es compilado y el compilador genera un archivo de extensión .jar, este archivo puede ser ejecutado ubicándolo en cualquier carpeta de la



computadora del usuario, con la aclaración que debe estar instalado en el equipo cliente el sistema Java runtime enviroment (JRE).

El software puede ser distribuido por medio de discos USB.

## 2.7 Comité de Control de Cambios (CCC)

El CCC es un comité que asegura que cada cambio está apropiadamente considerado por todas las partes y es autorizado antes de su implementación. El CCC es responsable de aprobar, monitorear y controlar cada solicitud de cambios para establecer una línea de trabajo.

El alcance del trabajo será aprobar o rechazar los cambios necesarios en planes, documentos y códigos. Las decisiones deberán ser tomadas respecto a las acciones que deberán estar basadas en la calidad del producto, asegurando el correcto estado del producto después de cada ciclo de prueba.

El CCC puede estar conformado por una o varias personas, pero es importante que estas tengan una visión global del proyecto.

La decisión del CCC y sus apreciaciones se registran en los apartados correspondientes del formulario del cambio.

Si el CCC aprueba el cambio, se genera la asignación de la tarea de llevar a cabo dicho cambio. Esta asignación y el formulario de cambio correspondientes, llegarán al SCMer .

El formulario de cambio debe incluir:

- Cambio a realizar.
- Productos a modificar.
- Restricciones.
- Criterios de revisión y auditoría.
- Tiempo estimado para realizar el cambio y dedicación real.
- Costo estimado y real.
- Persona responsable del cambio.
- Pruebas y reportes de pruebas.

Este último punto es importante, ya que si surge un cambio se deberán realizar nuevamente las mismas pruebas de manera de verificar que no se alteró la funcionalidad existente.

## 2.8 Integrantes del Equipo, Roles y reuniones

Los roles de los distintos miembros estarán dados por los siguientes cargos:

- Engineering Manager

- Release Manager - Coordinador de errores
- Uber Scrum Team
- Director de Ingeniería
- GPCM

Al ser 3 integrantes en el grupo, realizamos diversas tareas que engloban la mayoría de las responsabilidades de los roles nombrados anteriormente.

Integrantes	Contacto
Candotti, Enzo	enzocandotti93@gmail.com
Flores, Mauricio	elmauri229@gmail.com
Nicolaide, Christian	cnicolaide@gmail.com

Nuestra forma de trabajo con respecto al proyecto, fue con 2 reuniones reales donde nos organizamos y definimos la forma de trabajo a seguir, nos pusimos de acuerdo en las herramientas a utilizar y nivelamos los conocimientos en la utilización de las herramientas.

Siempre y en todo momento trabajamos todos los integrantes del grupo.

Fecha - Horas	Tipo de Reunión	Actividad
23/03/16 - 1,5 h	Real	Planificación y organización
28/03/16 - 2 h	Real	Nivelación de conocimiento sobre uso de herramientas.
31/03/16 2 h	Virtual	Creación de espacios virtuales

### Tipo de Reunión

Las reuniones reales las realizamos en los box de la Facultad, y para las reuniones virtuales y forma de contacto, utilizamos las siguientes Herramientas, con las que nos manteníamos al tanto.

Grupo de WhatsApp.

Grupo de Facebook del trabajo.

## 2.9 Herramienta de seguimiento de BUGS

Utilizamos la herramienta Issues proporcionada por Git para hacer seguimiento de los errores. Una vez creado el repositorio se tiene acceso a ella.

La dirección de la herramienta se encuentra en la sección 1.4.