**Plan de Gestión de Configuración de Software**

**El documento describe el Plan de gestión de la configuración que se seguirá en el desarrollo del Sistema de Tramite Documentario para la escuela de Postgrado de la Universidad Ricardo Palma.**

Contenido

[1. Introducción 3](#_Toc366867442)

[a. Propósito 3](#_Toc366867443)

[b. Aplicabilidad 3](#_Toc366867444)

[c. Gobierno 3](#_Toc366867445)

[d. Alcance 3](#_Toc366867446)

[e. Definiciones 4](#_Toc366867447)

[2. Gestión de la SCM 5](#_Toc366867448)

[a. Organización 5](#_Toc366867449)

[b. Roles y responsabilidades 6](#_Toc366867450)

[c. Políticas, directrices y procedimientos 6](#_Toc366867451)

[d. Herramientas, entorno e infraestructura 7](#_Toc366867452)

# Introducción

La gestión de la configuración es uno de los componentes de éxito de los proyectos TIC. Sin una gestión de la configuración efectiva, la integridad de los ítems de configuración del proyecto y la capacidad de reportar el estado y configuración de aquellos ítems se pone en peligro.

El plan de gestión de la configuración permite establecer un método consistente para identificar y controlar formalmente los ítems de configuración del proyecto.

Los ítems de configuración del proyecto incluyen elementos de hardware y software, así como también información de gestión del proyecto como planes.

La gestión de la configuración es una función integral de la provisión de proyectos TIC porque facilita la protección de los ítems de configuración y comunica los cambios que se han hecho sobre ellos.

Una gestión de la configuración, planificada y ejecutada de manera efectiva, contribuye a la producción de productos TIC de alta calidad evitando el retrabajo. Esto aumenta el valor de los activos informáticos y ahorra costes, contribuyendo a la entrega de proyectos que satisfacen los costes, calendarios, calidad y requerimientos establecidos.

## Propósito

El objetivo de este Plan de Gestión de la Configuración (CM), es proporcionar una visión general de la organización, actividades, tareas en general, y los objetivos de Gestión de la Configuración.

Un plan de gestión de la configuración sirve como una herramienta medular de planificación que describe los esfuerzos de planificación para implementar y ejecutar la gestión de la configuración a lo largo del ciclo de vida del proyecto. Provee visibilidad y control del producto referente a su desempeño, funcionalidad y atributos físicos.

## Aplicabilidad

El plan de gestión de la configuración debe ser desarrollado para cualquier proyecto clasificado mediano y pequeño.

## Gobierno

El Líder de proyecto es responsable de asegurar que el plan de gestión de la configuración sea desarrollado en conjunto con el plan de gestión del proyecto. El Lider de proyecto asegurará su integración en la planificación general.

## Alcance

El plan de gestión de la configuración debe involucrar a todas las fases del ciclo de vida del software. El documento permitirá mostrar los estándares de etiquetación de los productos de trabajo.

Así mismo esclarecerá el tipo de nomenclatura utilizada para el control de las versiones de los documentos que se encuentran dentro de los elementos de la gestión.

Por otra parte cualquier stakeholder podrá presentar cualquiera de los siguientes tipos de peticiones de cambio sobre el sistema, para el control de cambios:

* Petición de cambios en los requerimientos (adiciones, supresiones, modificaciones, aplazamientos) en el software actualmente en desarrollo
* Informes de los problemas en la producción corriente o sistemas de pruebas beta
* Petición de mejoras en los sistemas actuales de producción
* Petición de nuevos proyectos de desarrollo.

Este proceso de control de cambio se aplica a los productos de línea base creados o gestionados por los miembros del sistema, incluyendo:

* El software que se ha lanzado a la producción o se encuentra en versión beta
* Requisitos de las especificaciones del sistema
* Grupo de procedimientos y procesos
* Usuarios y documentación técnica

Las siguientes clases de productos de trabajo están exentos de este proceso de control de cambios:

* Los productos de trabajo que están todavía en desarrollo, a excepción de cambios en los requerimientos solicitados en nuevos proyectos

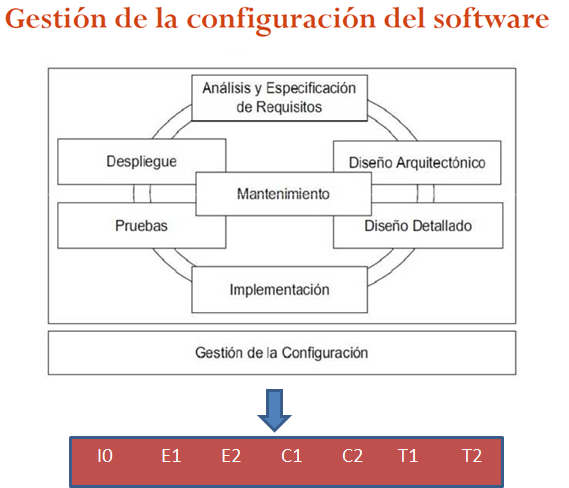
## Definiciones

* Línea Base. Un documento o producto oficial aprobado que sirve como punto de partida para futuras versiones.
* Petición de Cambio. Una solicitud que alguien ha presentado al sistema de control de cambio que describe un problema de software, una mejora solicitada, una propuesta de cambio en los requisitos de un producto en fase de desarrollo, o un nuevo proyecto que se propone.
* Stakeholder. Persona que directa o indirectamente se ve afectada por el sistema y que puede afectar el proyecto.
* Configuration Item. Los elementos que son puestos bajo el control de gestión de la configuración.
* Control de cambio. Proceso renovable de controlar todos los cambios.

# Gestión de la SCM

## Organización





## Roles y responsabilidades

El equipo de la Gestión de Configuración está conformado por los siguientes roles:

|  |  |
| --- | --- |
| **CM Rol** | **Responsabilidades** |
| Jefe del proyecto | * Definir los componentes de desarrollo. * Definir el control de accesos. * Definir políticas generales. * Definir la consecución de hitos. * Asignar actividades. |
| Administrador de la Configuración | * Crear los repositorios de administración de configuraciones. * Configurar entornos para los proyectos * Implementar las políticas definidas por el Jefe del proyecto . |
| Administrador de Releases | * Implementar los pasos de los elementos al entorno de producción a partir de las líneas base creadas por el Sistema de Administración de Configuraciones |
| Integrador del proyecto (Fábrica de Desarrollo) | * Identificar posibles construcciones en el entorno de desarrollo. * Entregar los requerimientos implementados de manera formal para su integración. * Sincronizar cuando sea necesario los cambios realizados por la fábrica con el punto de entrada al Sistema de Administración de Configuraciones. |
| Integrador del proyecto | * Identificar posibles construcciones para los entornos de pruebas y producción. * Planificar las liberaciones de las línea base del producto * Establecer liberaciones * Seleccionar la ubicación para artefactos liberados |

## Políticas, directrices y procedimientos

* Cada usuario tendrá derecho a solicitar un repositorio Git para su uso personal, el cual estará restringido por una quota y podrá distribuirla de la manera que más le acomode, pudiendo dividirla entre los repositorios que estime conveniente.
* Cada repositorio Git asignado a grupo de trabajo para los ramos impartidos por el departamento tendrá reglas diferentes a las de los personales y serán controlados por el Administrador de la configuración.
* Cada repositorio Git que sea asignado para un ramo solo durara desde el inicio del ramo hasta el final del ramo. Luego se dará un semestre para que lo respalden, luego de eso será eliminado.

## Herramientas, entorno e infraestructura

## Como herramienta se utilizara el sistema de control de versiones Git, para el entorno de desarrollo se utilizara Github.

## Herramientas: Git es un software de control de versiones, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente.

## Entre las características más relevantes se encuentran:

## Fuerte apoyo al desarrollo no-lineal, por ende rapidez en la gestión de ramas y mezclado de diferentes versiones. Git incluye herramientas específicas para navegar y visualizar un historial de desarrollo no-lineal. Una presunción fundamental en Git es que un cambio será fusionado mucho más frecuentemente de lo que se escribe originalmente, conforme se pasa entre varios programadores que lo revisan.

## Gestión distribuida. Git le da a cada programador una copia local del historial del desarrollo entero, y los cambios se propagan entre los repositorios locales. Los cambios se importan como ramas adicionales y pueden ser fusionados en la misma manera que se hace con la rama local.

## Los almacenes de información pueden publicarse por HTTP, FTP, rsync o mediante un protocolo nativo, ya sea a través de una conexión TCP/IP simple o a través de cifrado SSH. Git también puede emular servidores CVS.

## Gestión eficiente de proyectos grandes, dada la rapidez de gestión de diferencias entre archivos, entre otras mejoras de optimización de velocidad de ejecución.

## Todas las versiones previas a un cambio determinado, implican la notificación de un cambio posterior en cualquiera de ellas a ese cambio.

## Resulta algo más caro trabajar con ficheros concretos frente a proyectos, eso diferencia el trabajo frente a CVS, que trabaja con base en cambios de fichero, pero mejora el trabajo con afectaciones de código que concurren en operaciones similares en varios archivos.

## Los renombrados se trabajan basándose en similitudes entre ficheros, aparte de nombres de ficheros, pero no se hacen marcas explícitas de cambios de nombre con base en supuestos nombres únicos de nodos de sistema de ficheros, lo que evita posibles, y posiblemente desastrosas, coincidencias de ficheros diferentes en un único nombre.

**Entorno:** GitHub es un software web para alojar proyectos utilizando el sistema de control de versiones Git. Utiliza el framework Ruby on Rails por GitHub

Entre sus principales características se puede mencionar:

* Wiki para cada proyecto.
* Página web para cada proyecto.
* Gráfico para ver cómo los desarrolladores trabajan en sus repositorios y bifurcaciones del proyecto.
* Funcionalidades como si se tratase de una red social

**Infraestructura:**



Requisitos del sistema Cliente

* Sistema operativo
  + Microsoft® Windows XP, Windows Vista o Windows 7 (versiones de 32 y 64 bits).
* Especificaciones técnicas

|  |  |
| --- | --- |
| Procesador | Intel Core 2 Duo de 2 GHz (o equivalente) |
| Memoria | 2 GB RAM y 1 GB de espacio en el disco duro (sistemas operativos de 32 bits) 4 GB RAM y 1 GB de espacio en el disco duro (sistemas operativos de 64 bits) |
| Tarjeta gráfica | NVIDIA GeForce 5 o una versión más reciente |
| Otros | Unidad de DVD para la instalación del software |

## Calendario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre de tarea | Duración | Comienzo | Fin | Predecesoras |
| **Plan de la gestión de la configuración** | **10.5 días?** | **sáb 31/08/13** | **lun 16/09/13** |  |
| Inicio | 0 días | sáb 31/08/13 | sáb 31/08/13 |  |
| **Definición de la estrategia de la configuración** | **1.5 días** | **lun 02/09/13** | **mar 03/09/13** |  |
| Establecer el propósito | 0.5 días | lun 02/09/13 | lun 02/09/13 | 2 |
| Definir la aplicabilidad | 0.5 días | lun 02/09/13 | lun 02/09/13 | 4 |
| Establecer el gobierno y el alcance | 0.5 días | mar 03/09/13 | mar 03/09/13 | 5 |
| **Definición de la gestión de la configuración** | **3 días** | **mar 03/09/13** | **vie 06/09/13** |  |
| Establecer organización | 0.5 días | mar 03/09/13 | mar 03/09/13 | 6 |
| Definir roles y responsabilidades | 0.5 días | mié 04/09/13 | mié 04/09/13 | 8 |
| Establecer políticas, directrices y procedimientos | 0.5 días | mié 04/09/13 | mié 04/09/13 | 9 |
| Establecer herramientas, entorno e infraestructura | 1 día | jue 05/09/13 | jue 05/09/13 | 10 |
| Generar calendario | 0.5 días | vie 06/09/13 | vie 06/09/13 | 11 |
| **Identificación de la configuración** | 1 día | vie 06/09/13 | lun 09/09/13 | 7 |
| **Establecer control de la configuración** | 2 días | lun 09/09/13 | mié 11/09/13 | 13 |
| **Obtener el estado de la configuración** | 1 día | mié 11/09/13 | jue 12/09/13 | 14 |
| **Obtener la auditoría de la configuración** | 1 día? | jue 12/09/13 | vie 13/09/13 | 15 |
| **Gestión y entrega de releases del software** | 1 día? | vie 13/09/13 | lun 16/09/13 | 16 |

# Actividades de la SCM:

3.1. Identificación de la configuración, Nomenclatura

3.1.1. Ítems de Configuración

Los ítems de configuración se clasifican en:

1. Ítems en evolución, tales como documentos, los que están sujetos a una o más revisiones y nuevas liberaciones durante el ciclo de vida del software;
2. Ítems fuente, generalmente código fuente y archivos objeto utilizados para compilar una aplicación de software para ambiente de producción, los cuales son generalmente numerosos y cambian frecuentemente;
3. Ítems de soporte, como sistemas operativos y software base, de los cuales el proyecto requiere ciertas versiones para su operación exitosa;
4. Ítems de archivo, tales como revisiones SQA las cuales generalmente sirvieron de soporte para la toma de decisiones durante el ciclo de vida del software, son almacenadas normalmente en formato electrónico para referencia futura;

**Nota.-** Para más detalle se incluye un anexo de ayuda para la clasificación de los ítems de configuración.

Los métodos para la identificación única de instancias de cada ítem de configuración son específicos de la clase de cada ítem. Los ítems en evolución se asignan identificadores únicos, mientras que la identificación de ítems fuente es gestionada por la herramienta de gestión de configuración seleccionada (o el entorno de desarrollo). Los ítems de soporte llevan su propia identificación y numeración de versiones asignados por sus desarrolladores, mientras que los ítems de archivo son identificados principalmente por el nombre y la fecha.

3.1.2. Convención de Nomenclatura de Ítems de Configuración

**Identificación de ítems en evolución**

Los ítems en evolución son de dos tipos: documentos, y archivos ejecutables o de soporte.

**Documentos**

Los ítems en evolución tipo documento se asignan identificadores únicos que permiten identificar el proyecto y componente (si es aplicable) con el cual está asociado, junto con el nivel de revisión actual. El identificador consiste de una a tres partes separadas por guión en el formato: PROYECTO-ACRÓNIMO, ó, PROYECTO-ACRÓNIMO-COMPONENTE.

Los ítems en evolución que son ítems no específicos a un proyecto único, tales como políticas, descripciones de procesos y guías, son identificados únicamente por su acrónimo, por ejemplo: SPMP (Software Project Management Policy).

Los ítems en evolución que son específicos de un proyecto, pero no asociados con un componente del proyecto, utilizan un identificador de dos partes: ACRÓNIMO\_PROYECTO y ACRÓNIMO derivado del tipo de artefacto. Por ejemplo, para identificar el plan de gestión de la configuración del proyecto ACME, tenemos: ACME-SCMP.

Los ítems en evolución que son específicos de un proyecto y están asociados con un componente específico, utilizan un identificador de tres partes: ACRÓNIMO\_PROYECTO, ACRÓNIMO\_COMPONENTE, y, ACRÓNIMO derivado del tipo de artefacto. Por ejemplo, para identificar el documento de especificación de requerimientos, del componente B2B (Business to Business) del proyecto ACME, tenemos: ACME-B2B-SRD.

El nivel de versión de cada ítem se mantiene como un identificador separado. Esto permite que el identificador principal sea utilizado como parte del nombre de archivo o URL para acceder a la versión más actualizada sin necesidad de requerir cambios a todos los ítems referenciados. El nivel de versión se mantiene como un identificador numérico con dos componentes: Versión.Revisión. Ejemplo: 1.1; Versión 1, Revisión 1.

**Número de versión**

El número de versión cambia únicamente cuando la arquitectura principal del ítem ha cambiado, o cuando el ítem es completamente reconstruido, con cambios internos sustanciales. En este caso la versión 1.1 se convertirá en versión 2.0.

**Número de revisión**

El número de revisión cambia cuando el contenido ha cambiado, pero la estructura principal y el flujo del ítem se mantiene igual. La secuencia normal de las revisiones es: 1.0, 1.1, 1.2, etc.

**Archivos ejecutables y de soporte**

Los ejecutables del software y los archivos de soporte son identificados generalmente por el nombre y el número de versión, tales como “Main DB v1.1a"

La convención para los nombres para cada ítem de software en evolución es definida por el equipo de desarrollo. El esquema de numeración de versiones consiste de tres componentes: Versión.RevisiónActualización. Ej: 1.1a.

**Número de versión**

El número de versión cambia únicamente cuando la arquitectura principal del ítem de software cambia, cuando migramos de un nivel de herramienta de desarrollo a otro, cuando una aplicación es totalmente reconstruida, o cuando se producen cambios sustanciales en la GUI. En este caso, la versión 1.1a se convierte en la versión 2.0

**Número de revisión**

El número de revisión es actualizado cuando se añade nuevas características, funcionalidad y otro contenido, o estas han cambiado significativamente. Normalmente la arquitectura principal o la GUI ha sido extendida o limitada de alguna manera. La razón más común de cambiar el número de revisión es cuando añadimos un nuevo módulo u otra funcionalidad al ítem de software. La secuencia normal de revisión es 1.0, 1.1, 1.2, etc.

**Carácter de actualización**

El carácter de actualización se incrementa cuando el único cambio al ítem de software es corregir uno o más defectos, sin añadir ninguna nueva funcionalidad. Las actualizaciones evolucionan 1.1a, 1.1b, etc. Esta actualización se sobrescribe cuando una revisión combinada, que incluye arreglar defectos y añadir nuevas características, se lleva a cabo. En tal caso, se incrementa el número de revisión y se descarta el carácter, es decir: 1.1b a 1.2.

**Identificación de ítems fuente**

Esto se maneja en base a la herramienta de gestión de la configuración utilizada, o de las utilidades del entorno de desarrollo para versionamiento.

**Identificación de ítems de soporte**

Son identificados por su nombre y el número de versión necesario para soportar el entorno de producción o desarrollo. Por ejemplo, si un editor se actualiza de la versión 2.1 a 2.2a, el rango de versión del ítem de configuración será 2.1 – 2.2a. Esto es importante para la recuperación posterior de la información archivada del proyecto; la documentación y los ítems fuente se manejan mejor si conocemos las versiones compatibles de sus ítems de soporte.

**Identificación de ítems de archivo**

Los ítems de archivo son documentos misceláneos de soporte y registros de comunicaciones que son almacenadas para referencias futuras. Estos ítems se almacenan según lo describan el plan de gestión de la configuración. Cada ítem se identifica por el nombre de archivo y la fecha de modificación. En el caso de que se tenga que mantener el mismo nombre en el subdirectorio correspondiente, se añadirá un número secuencial para prevenir conflictos.

3.1.3. Librerías Controladas