



# UNIVERSIDAD DE CASTILLA-LA MANCHA ESCUELA SUPERIOR DE INFORMÁTICA

## Plan de Gestión de la Configuración

*Álvaro Ávila Cabello*  
*Daniel López Paredes*  
*Luis Miguel Barreiro Cabello*  
*Víctor Lay Gómez*

Asignatura: Ingeniería del Software II

Titulación: Grado en Ingeniería Informática

Fecha: 10 de octubre del 2023

# ÍNDICE

1. Introducción .....	1
• Propósito del plan.....	1
• Alcance del plan.....	1
• Relación con la organización y con otros proyectos .....	1
• Términos clave .....	2
• Referencias.....	3
2. Criterios para la identificación de los elementos de configuración a los cuales el CM será aplicado .....	3
3. Limitaciones y suposiciones que afectan al plan.....	4
4. Responsabilidades y autoridades del CM.....	4
5. Organización del proyecto .....	5
6. Responsabilidades de la GC .....	5
7. Políticas aplicables, directivas y procedimientos .....	6
8. Actividades planificadas, agenda y recursos.....	7
Agendas del CM.....	7
Recursos del CM .....	8
9. Mantenimiento del Plan de CM .....	8

# 1. Introducción

Este informe actúa como un manual para la gestión de la configuración durante el proceso de desarrollo de un sistema web para la gestión de una biblioteca. Aquí se proporcionan las pautas esenciales para supervisar las versiones de los diversos componentes, administrar las ramas de desarrollo y asignar responsabilidades y tareas a los diferentes participantes en el proyecto.

- **Propósito del plan**

El propósito principal de este Plan de Gestión de Configuración es:

1. Definir los procesos y procedimientos de gestión de configuración que se seguirán a lo largo del ciclo de vida del proyecto.
2. Establecer un marco claro para identificar, controlar y gestionar los elementos de configuración.
3. Asegurar la consistencia e integridad del software y su documentación asociada.
4. Facilitar la comunicación efectiva entre los miembros del equipo del proyecto y las partes interesadas con respecto a los cambios de configuración.
5. Mejorar la capacidad del proyecto para gestionar y mitigar los riesgos asociados con los cambios.

- **Alcance del plan**

Este plan abarca todos los aspectos de la gestión de configuración dentro del proyecto, incluyendo, pero no limitándose a:

1. **Identificación:** Definir e identificar de manera única todos los elementos de configuración (CI) dentro del proyecto.
2. **Control de Cambios:** Establecer procesos para presentar, revisar, aprobar e implementar cambios en los CIs.
3. **Contabilidad del Estado de Configuración:** Mantener registros precisos y actualizados del estado y la historia de los CIs.
4. **Auditorías de Configuración:** Realizar auditorías periódicas para verificar la integridad de los procesos de gestión de configuración.
5. **Gestión de Versiones:** Coordinar el lanzamiento e implementación de las versiones de software.

- **Relación con la organización y con otros proyectos**

El proyecto se lleva a cabo dentro de un entorno organizacional dinámico que promueve la colaboración entre equipos y proyectos. La gestión de la configuración del software se integra estrechamente con las políticas y prácticas de gestión de la configuración establecidas por la organización, asegurando la coherencia y la alineación con los estándares predefinidos.

La organización opera con una metodología ágil que fomenta la entrega iterativa y la colaboración entre equipos multifuncionales. La gestión de la configuración se alinea con las directrices establecidas en el marco de desarrollo ágil adoptado por la empresa, permitiendo la entrega continua y la adaptabilidad a cambios en los requisitos del cliente.

## • Términos clave

1. **Configuración del Software:** Conjunto de componentes identificables en un sistema que se encuentran bajo control de gestión y se tratan como un conjunto coherente.
2. **Ítem de Configuración (CI):** Unidad identificable y manejable dentro de la configuración, como archivos de código fuente, documentos, scripts de configuración, etc.
3. **Gestión de la Configuración:** Proceso sistemático para identificar y controlar los ítems de configuración, gestionar los cambios y garantizar la integridad y la trazabilidad.
4. **Control de Versiones:** Sistema que administra cambios a los ítems de configuración, mantiene el historial de versiones y asegura la consistencia.
5. **Rama (Branch):** Variante de un conjunto de ítems de configuración que se deriva de la línea principal de desarrollo para implementar cambios separadamente.
6. **Etiqueta (Tag):** Marcador identificador asignado a una versión específica de los ítems de configuración para facilitar su identificación y seguimiento.
7. **Auditoría de Configuración:** Proceso de revisión sistemática de la configuración para garantizar que cumple con los requisitos especificados y los estándares definidos.
8. **Baseline:** Estado establecido y aprobado de un conjunto de ítems de configuración que sirve como referencia para medir los cambios.
9. **Integración Continua:** Práctica de fusionar el trabajo de múltiples desarrolladores en un repositorio compartido varias veces al día para detectar y solucionar problemas de integración temprano.
10. **Despliegue Continuo:** Automatización del proceso de implementación para llevar cambios de código del entorno de desarrollo al de producción de manera rápida y segura.
11. **Gerente de Proyecto:** Responsable de liderar y garantizar la entrega exitosa del proyecto.
12. **Configuration Manager:** Encargado de planificar, implementar y supervisar las actividades de gestión de configuración.
13. **Equipo de Desarrollo:** Incluye a los ingenieros, desarrolladores y otros miembros involucrados en la creación de software.
14. **Aseguramiento de la Calidad (QA):** Se encarga de garantizar que los productos y entregables cumplan con los estándares de calidad.
15. **Stakeholders:** Incluye a los patrocinadores del proyecto, usuarios finales y otras partes interesadas.
16. **Control de Cambios:** Establecimiento de procesos para presentar, revisar, aprobar e implementar cambios en los ítems de configuración.
17. **Gestión de librerías:** Mantener y gestionar repositorios que almacenan elementos de configuración, como código fuente, documentos, configuraciones, datos, etc.
18. **Gestión personal:** Proveer la infraestructura y el entorno de configuración para el proyecto, asegurándose de que todos los miembros entiendan y ejecuten las actividades de SCM asignadas.
19. **Política de Identificación de Configuración:** Establece la identificación única y la descripción clara de los elementos de configuración.
20. **Política de control de versiones:** Controla y registra versiones en un repositorio centralizado, asegurando que solo las versiones aprobadas estén disponibles para su implementación.
21. **Política de Ramificación y Fusión:** Define las ramas de desarrollo y cómo se fusionan cambios entre ellas.

22. **Directiva de nomenclatura de archivos:** Establece reglas para nombrar archivos de manera clara y concisa.
23. **Directiva de Gestión de Cambios:** Establece cómo se comunica y aprueba cualquier cambio en la configuración.
24. **Mantenimiento del Plan de CM:** Supervisa y actualiza el plan de configuración para adaptarse a los cambios en el proyecto.

## • Referencias

IEEE Std 828-2014 - IEEE Standard for Configuration Management in Systems and Software Engineering

## 2. Criterios para la identificación de los elementos de configuración a los cuales el CM será aplicado

Vamos a definir los diferentes elementos de configuración (Configuration Items - CIs) en función de una serie de criterios que dependerán de varios factores como los requisitos del cliente, impacto en el proyecto, importancia en las diferentes fases de desarrollo, etc.

En cuanto a elementos software. Trabajaremos con el lenguaje de programación java en el framework Spring junto con HTML y el motor de plantillas Thymeleaf en base a un modelo MVC (modelo, vista, controlador) para el desarrollo de nuestro proyecto.

Además, utilizaremos Maven como gestor de paquetes y dependencias.

Para gestionar la persistencia utilizaremos Java JPA una base de datos embebida Derby (Dbeaver Embedded).

Para el correcto funcionamiento necesitaremos estos requisitos software para la completa compatibilidad de todos los trabajadores durante el proyecto:

- Java JDK versión 21
- Maven versión 3.9.4
- Eclipse versión 2023-09
- Drivers Dbeaver versión 10.16.1.1

También para el correcto funcionamiento del framework Spring añadiremos las siguientes dependencias. En la siguiente lista salen los diferentes artefactos que utilizaremos:

- spring-boot-starter-data-jpa
- spring-boot-starter-thymeleaf
- spring-boot-starter-web
- spring-boot-devtools
- Derby
- Derbytools
- spring-boot-starter-tomcat
- spring-boot-starter-test

Además del plugin de Maven:

- spring-boot-maven-plugin

Por último, para el control de calidad (QA) de nuestro software utilizaremos la herramienta SonarCloud para monitorizar los posibles errores, bugs y mejoras de código. Esto irá acompañado de la realización de tests unitarios utilizando Unit Test de java

En cuanto al control de versiones utilizaremos Git junto con GitHub y GitHub Projects para mantener un control exhaustivo sobre las diferentes versiones y funcionalidades que se vayan desarrollando durante el proyecto.

### 3. Limitaciones y suposiciones que afectan al plan

Para evaluar las limitaciones y suposiciones que afectan a nuestro plan, es necesario establecer los detalles específicos de nuestro plan. Sin embargo, nos encontramos con una serie de limitaciones que pueden afectar a nuestro control de configuración.

1. **Recursos financieros:** Nuestra empresa presenta unas limitaciones presupuestarias, una de ellas es el costo de herramientas de control de configuración, las cuales disponen de unas licencias de uso, para poder disponer de su uso completo. La empresa ha tenido que invertir en licencias de las diferentes herramientas.
2. **Recursos de personal:** La empresa tiene cuatro integrantes, todos desarrolladores y uno es scrum master, para el control de la configuración se ha visto limitado por el número de miembros, por lo que la eficacia de la administración del control de la configuración se ha visto afectada de manera significativa.
3. **Complejidad del sistema:** La complejidad del sistema presentado nos generó una serie de desafíos a la hora de determinar la gestión de la configuración
4. **Cambios de los requisitos:** Se han producido cambios en el sistema, pero los requisitos no se han visto modificados por parte del cliente.

### 4. Responsabilidades y autoridades del CM

El Configuration Manager según la norma IEEE 828 tiene la responsabilidad de establecer y mantener la integridad de la configuración del sistema. Esto implica identificar, auditar y controlar los elementos de configuración, así como gestionar cambios y asegurar la trazabilidad.

Las responsabilidades específicas incluyen la identificación de configuración, control de cambios, auditorías de configuración y gestión de la biblioteca de configuración. Además, el Configuration Manager tiene autoridad para aprobar o rechazar cambios propuestos, asegurando que se sigan los procesos definidos y manteniendo la documentación de configuración actualizada.

El Configuration Manager también desempeña un papel fundamental en la planificación y ejecución de revisiones y auditorías de configuración. Esto implica garantizar que la documentación de configuración esté completa, precisa y refleje el estado real del sistema.

Además, tiene la autoridad para coordinar con otros equipos y asegurar que todos los aspectos de la configuración, desde hardware hasta software, estén adecuadamente

documentados y versionados. También puede ser responsable de mantener y gestionar la línea base de configuración, que sirve como referencia estable para el sistema.

En resumen, el Configuration Manager no solo gestiona cambios, sino que también vela por la coherencia, integridad y trazabilidad de la configuración a lo largo del ciclo de vida del sistema.

## 5. Organización del proyecto

El proyecto se organiza atendiendo a la siguiente estructura:

- **Scrum Master:** Facilita el proceso Scrum, ayuda al equipo a comprender y seguir las prácticas ágiles. Elimina obstáculos y fomenta un entorno colaborativo y de mejora continua. Trabaja de la mano con el equipo de desarrollo y el Gerente de Proyecto para integrar las prácticas Scrum con la gestión de configuración.
- **Configuration Manager:** Encargado de planificar, implementar y supervisar las actividades de gestión de configuración. Colabora estrechamente con el Gerente de Proyecto para garantizar la alineación de la gestión de configuración con los objetivos generales del proyecto.
- **Equipo de Desarrollo:** Incluye a los ingenieros, desarrolladores y otros miembros directamente involucrados en la creación de software. Trabajan bajo la supervisión del Configuration Manager y contribuyen activamente a las actividades de gestión de configuración.
- **Aseguramiento de la Calidad (QA):** Se encarga de garantizar que los productos y entregables cumplan con los estándares de calidad establecidos. Colabora con el Configuration Manager para realizar revisiones y auditorías de configuración.
- **Stakeholders:** Incluye a los patrocinadores del proyecto, usuarios finales y otras partes interesadas. Participan en revisiones y proporcionan información clave para la gestión de configuración.

## 6. Responsabilidades de la GC

La gestión de la configuración en nuestro sistema de software es un elemento principal para controlar y gestionar los elementos de la configuración durante el ciclo de vida, algunas de estas responsabilidades se pueden representar en los siguientes elementos claves de la gestión de la configuración:

**Control de cambios:** para ello la empresa ha empleado una herramienta como git en el cual, mediante los *commits*, se notificaban las modificaciones pertinentes en el proyecto en las diferentes ramas de funcionalidades, una vez completadas las funcionalidades se hará *merge* en la rama de *develop* que servía para unir todas funcionalidades e integrarlas con las máximas garantías de una correcta funcionalidad entre las diferentes ramas.

**Control del versionado:** también, a la hora de realizar los cambios en el proyecto que afectaran al conjunto o parte de él se cambiaba la versión, en el caso de que fuera un cambio menor X.Z.Y. Cuando se producían cambios entre las diferentes funcionalidades se producía un cambio *minor* versión X.Y.Z > 0

Se produjo un Cambio *major* a la hora de introducir en la estructura del proyecto un cambio, la cual obligo a cambiar la versión.

**Auditorias de Configuración:** se realizan auditorias de la línea base antes de una liberación de esta o de una actualización de la versión de un componente prioritario.

Estas auditorias incluyen:

- Objetivo: Verificar que la línea base esté compuesta por una colección definida y coherente.
- Agenda de las auditorias: previo a liberación o actualización del proyecto
- Dirigidas: las auditorias estarán a cargo de SCMR
- Participantes: SCMR y los autores de la configuración a resaltar en la auditoria.
- Reportes de deficiencias y acciones correctivas: estas estarán determinada por los participantes.
- Aprobación: lo determinara el SCMR

**Gestión de librerías:** Mantener y gestionar librerías o repositorios de configuración, nos permiten poder almacenar diversos elementos de configuración, incluyendo código fuente, documentos, configuración, datos y los demás componentes relacionados con el software.

**Gestión personal:** el SCMR debe proveer la infraestructura y el entorno de configuración para el proyecto. Debe preocuparse porque todos los integrantes del grupo entiendan y puedan ejecutar las actividades de SCM que el plan le asigna, así como conseguir que estas sean llevadas a cabo. Seguir la línea base, controlando e informando al resto de componentes sobre la manera de usarlo.

## 7. Políticas aplicables, directivas y procedimientos

Durante el proyecto se aplicarán una serie de políticas y directivas que serán clave para el buen funcionamiento y desarrollo de las diferentes actividades que realizará cada desarrollador. Estas políticas y directivas son las siguientes:

- Política de Identificación de Configuración: los elementos de configuración deben identificarse con un nombre único y una descripción que refleje su función y propósito (Spring- Framework Java). Dichos elementos deberán ser supervisados por el Configuration Manager antes de su debida formalización.
- Política de control de versiones: todas las versiones de los elementos de configuración deben ser controladas y registradas en un repositorio centralizado. Solo las versiones aprobadas deben estar disponibles para su implementación. Dicho versionado se aplicará cuando el software pase a pre-producción (rama *release*) y un versionado final cuando pase a producción (rama *main*). Dicho versionado será supervisado por el Configuration Manager y por el Gerente del proyecto en caso de ser necesario
- Política de Ramificación y Fusión (*Branching and Merging*): durante el desarrollo del proyecto tendremos 4 ramas principales:
  - Rama *Main*: donde el sistema pasará a fase de producción. En esta rama solo se realizará un *commit* por lanzamiento. Los pull-request en dicha rama estarán siempre supervisados por el Configuration Manager y por el Gerente del proyecto



- Rama *Hotfix*: en esta rama se solucionarán problemas graves que suceda en el sistema cuando pase a producción. Dichos cambios se volverán a pasar a producción y se devolverán a la rama *develop* también.
- Rama *Release*: lugar donde se llevarán a cabo distintos tests donde comprobar si el sistema está preparado para pasar a producción
- Rama *Develop*: aquí se subirán todas las funcionalidades desarrolladas en las ramas *feature*.
- Rama *Feature*: en esta rama nombrada anteriormente se desarrollarán todas las funcionalidades que requiera el sistema. Una vez finalizada la funcionalidad se creará un pull-request hacia la rama *develop*.

Los conflictos que puedan dar lugar en el *merge* serán comentados con el grupo y consultados con el Configuration Manager en caso de ser necesario

- Directiva de nomenclatura de archivos: en el momento de la creación y nombrado de archivos se nombrarán a dichos archivos de la forma más clara posible definiendo de forma concisa su finalidad (i.e. formListarTitulos.html)
- Directiva de Gestión de Cambios: en el momento de algún cambio en la configuración, en la estructura del proyecto o cualquier *Configuration Item* será obligatorio comunicar de manera clara y concisa la propuesta de cambio al *Configuration Manager* y si fuese necesario al Gerente del proyecto

## 8. Actividades planificadas, agenda y recursos

### Agendas del CM

Se describen en el tiempo las actividades de SCM que serán realizadas en el proyecto,

Actividad	Entrega	Tiempo estimado
Control de versiones	Código fuente y versionado	Continuo en cada ciclo de vida del proyecto
Gestión de cambios	Cambios implementados y verificados	Varía según la implementación y aprobación
Auditorías de control de gestión	Documentación de la gestión de control	Programadas periódicamente, al final de cada sprint
Gestión de la baseline	Documentación de la línea base	Según los hitos del proyecto
Integración continua y despliegue	Entregables entregados y desplegados automáticamente	Continua en cada ciclo del proyecto

## Recursos del CM

- **Herramientas**  
IDE eclipse  
Apache Derby
- **Lenguaje:**  
HTML  
CSS  
Java  
XML

## 9. Mantenimiento del Plan de CM

Para mantener el plan, el SCMR es el responsable de monitorear el plan de configuración, por cada interacción nueva se revisará el plan de configuración, si se producen modificaciones, se hará constar mediante comunicación directa (correo) a todos los integrantes.