

Facultad de matemáticas

Plan de administración de la configuración

Versión 1.0

Autores:

Aaron Isaac Graniel Arzat

Fernando Joachín Prieto

David Peña Muñoz

Jose Luis Pooc Moo

Andrea Isabel Torres Perez

**CONTROL DE DOCUMENTACIÓN**

**Control de Configuración**

| **Título:** | Plan de administración de la configuración |
| --- | --- |
| **Referencia:** | PLN\_PlanAdmistracionConfiguracion\_ProyectoConteoLineas\_v1.0\_2025-02-19 |
| **Autor(es):** | David Muñoz Peña, Aaron Isaac Graniel Arzat |
| **Fecha:** | 19/02/2025 |

**Histórico de versiones**

| **Versión** | **Fecha** | **Estado** | **Responsable** | **Nombre de archivo** |
| --- | --- | --- | --- | --- |
| 1.0 | 19/02/2025 | A | David Muñoz Peña, Aaron Isaac Graniel Arzat | PLN\_PlanAdmistracionConfiguracion\_ProyectoConteoLineas\_v1.0\_2025-02-19 |

**Estado:** (B)orrador, (R)evisión, (A)probado

**Histórico de cambios**

| **Versión** | **Fecha** | **Cambios** |
| --- | --- | --- |
| 1.0 | 19/02/2025 | Primera versión. |

**Índice**

[**1. Introducción 4**](#_wsqrngq5caka)

[**2. Propósito 4**](#_typfdik6m7un)

[**3. Alcance 4**](#_3pzyz12e4aqk)

[4.1. Código Fuente 4](#_ya2y1e1lcdwm)

[4.2. Documentación 5](#_ejle6dbdz7f)

[4.3. Tickets y Registros en Jira 5](#_xv4nbjg9sl95)

[**5. Control de Configuración 5**](#_250cy33pc3b7)

[5.1. Solicitud y Aprobación de Cambios 5](#_x56en9x3ulfc)

[5.2. Reuniones de Seguimiento 6](#_487oune256z2)

[**6. Estado de Configuración 6**](#_3j6y717399fb)

[**7. Auditorías y Revisiones 7**](#_xf1t1nubaqqm)

[**8. Liberación del Software 7**](#_gyeaipg4ksnw)

[**9. Control de Cambios 7**](#_qkedr4az6r0s)

[**10. Herramientas 8**](#_y67q1jb8sjs0)

# 

# **1. Introducción**

Este documento establece los procedimientos y herramientas para la administración de la configuración del proyecto contador de líneas de código, con el objetivo de garantizar la trazabilidad, control de cambios y mantenimiento adecuado del software durante su ciclo de vida.

# **2. Propósito**

El plan de gestión de configuración tiene como finalidad:

* Definir los lineamientos para identificar, versionar y controlar los cambios en los artefactos del proyecto.
* Mantener la integridad y coherencia del producto a lo largo de su ciclo de vida.
* Proveer visibilidad y trazabilidad de todos los cambios, facilitando auditorías y revisiones internas.

Las referencias utilizadas son las normas IEEE 1042-1987 y ISO 10007.

# **3. Alcance**

El plan cubre la gestión de configuración de todos los artefactos del software, incluyendo:

* Código fuente desarrollado en Java, encargado de realizar el análisis de líneas de código en archivos fuente.
* Documentos de soporte, que incluyen:
* Especificación de Requisitos de Software (ERS).
* Documento de Casos de Prueba.
* Manual de Usuario.
* Estándares de Codificación y Conteo.
* Configuraciones del entorno de desarrollo, incluyendo dependencias y herramientas utilizadas.
* Control de versiones y cambios, asegurando la trazabilidad y gestión eficiente de modificaciones en el código y documentación.

**4. Identificación de Configuración**

## **4.1. Código Fuente**

* **Repositorio:** GitHub.
* **Nomenclatura de Ramas:**
  + Para cada ticket (ej. “SCRUM #15 – Implementar login de usuario”), se crea una rama cuyo nombre reemplaza espacios por guiones bajos: SCRUM-15\_Implementar\_login\_de\_usuario.

## **4.2. Documentación**

* **Nomenclatura de Archivos:** Formato: **Categoría\_Descripción\_Versión\_Fecha.ext** Donde:
  + **Categoría**:
    - INF\_ → Informes
    - REP\_ → Reportes
    - ESP\_ → Especificaciones
    - MAN\_ → Manuales
    - PLN\_ → Planes
  + **Descripción**: Nombre en PascalCase que describa brevemente el contenido.
  + **Versión**: Número de versión (p. ej. v1.0, v2.1).
  + **Fecha**: Formato YYYY-MM-DD.
  + *Ejemplo:* INF\_ConteoLineasCodigo\_v1.0\_2024-02-13.docx

## **4.3. Tickets y Registros en Jira**

* **Formato de Tickets:** SCRUM #Número – Descripción.
* **Estados Principales:** *En Proceso*, *Revisión*, *Aprobado*, entre otros.
* **Asignación de Tareas:** Cada ticket se asigna a un responsable y se realiza un seguimiento en las reuniones periódicas.

# **5. Control de Configuración**

## **5.1. Solicitud y Aprobación de Cambios**

* **Código Fuente:**
  + Creación de una rama para cada cambio.
  + Generación de pull request (PR) revisado por al menos un integrante antes del merge a la rama principal.
* **Documentación:**
  + Creación de nuevos tickets en Jira para los cambios de documentos.
  + Uso de la convención de nombres para las nuevas versiones.
  + Mantenimiento de un historial de versiones y cambios (ver Sección 6.2).

## **5.2. Reuniones de Seguimiento**

Se realizan reuniones (scrums) para asignar tickets y actualizar su estado en Jira, asegurando la correcta priorización y revisión de cada ítem de configuración.

# **6. Estado de Configuración**

Para cada documento o artefacto considerado ítem de configuración, se elabora una **ficha de control** que incluye:

1. **Control de Configuración**
   * Título del documento.
   * Referencia (nombre del archivo, siguiendo la convención).
   * Autor(es).
   * Fecha de creación o última modificación.
   * Estado actual (Borrador, Revisión, Aprobado, etc.).
2. **Histórico de versiones** Se utiliza una tabla donde se detalla cada versión liberada:

*Tabla 1 - Historial de Versiones*

| **Versión** | **Fecha** | **Responsable** | **Nombre del archivo** |
| --- | --- | --- | --- |
| 1.0 | 2025-02-17 | *Nombre del responsable* | ESP\_EstandarDocumentacion-ProyectoConteoLineas\_v1.0\_2025-02-17.docx |

1. **Histórico de cambios** Una tabla que refleja los cambios más significativos en cada versión:

*Tabla 2 - Historial de Cambios*

| **Versión** | **Fecha** | **Cambios realizados** |
| --- | --- | --- |
| 1.0 | 2025-02-17 | Primera versión |

Esta estructura permite llevar un seguimiento claro y ordenado de la evolución de cada documento o artefacto, facilitando su auditoría y revisión.

# **7. Auditorías y Revisiones**

Para asegurar la efectividad del plan de gestión de configuración:

* **Auditorías Periódicas:** Verifican la consistencia entre el repositorio de código, la documentación y los registros de Jira.
* **Revisiones de Pull Requests:** Cada cambio en el código fuente pasa por una revisión formal en GitHub, exigiendo al menos una aprobación antes del merge.
* **Control de Documentos:** Validación de la nomenclatura, versiones y el uso correcto de las fichas de control (ver Sección 6).

# **8. Liberación del Software**

Los procedimientos de liberación incluyen:

* **Pruebas Previas a la Liberación:** Se ejecutan pruebas unitarias, de integración y de aceptación.
* **Aprobación en Jira:** El ticket asociado al release debe estar en estado *Aprobado* tras la revisión final.
* **Merge en Rama Principal:** Tras la aprobación, se integra la rama de desarrollo en la rama principal.
* **Distribución:** La versión final se almacena en el repositorio central de GitHub y se comparten los documentos actualizados.

# **9. Control de Cambios**

Todo cambio (en código o documentación) debe seguir el proceso:

1. **Creación de Ticket en Jira**: Se registra la solicitud de cambio con su respectivo impacto.
2. **Asignación y Desarrollo**: El responsable crea una rama en GitHub (para código) o trabaja en la actualización del documento (para documentación).
3. **Revisión**:
   * Código: Se realiza un pull request revisado por al menos una persona.
   * Documentos: Se verifica la nueva versión y se actualiza la tabla de histórico de cambios.
4. **Aprobación y Merge**:
   * Una vez aprobado el pull request, se integra a la rama principal.
   * El documento se considera “Aprobado” y se publica con la nomenclatura y versión actualizados.

# **10. Herramientas**

* **Jira:** Para la gestión de tickets, sprints y control de estados (En Proceso, Revisión, Aprobado, etc.).
* **GitHub:** Repositorio central de código, manejo de ramas (branching), pull requests y versionado.
* **Plataforma de Documentación (Drive):** Para almacenar los documentos con la nomenclatura acordada y su historial de versiones.