

PLANTILLA — PLAN / DOCUMENTO DE GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE (GCS)

Portada

- **Proyecto/Sistema:**
- **Versión del documento:**
- **Fecha:**
- **Responsable GCS:**
- **Organización/Equipo:**

Control de versiones

Versión	Fecha	Autor	Descripción del cambio
1.0			

1. Introducción

(Explicar qué es GCS y para qué se usará en el proyecto: control de versiones, control de cambios, trazabilidad, integridad.)

2. Objetivo

Establecer el proceso de Gestión de Configuración de Software para:

- controlar versiones y artefactos,
- mantener la integridad del producto,
- administrar cambios de forma formal,
- asegurar trazabilidad de entregables.

3. Alcance

Artefactos bajo control de configuración:

- Código fuente
- Documentación
- Casos de prueba
- Scripts de BD
- Archivos de despliegue (Docker, YAML, etc.)
- Manuales y reportes

4. Roles y responsabilidades GCS

Rol	Nombre	Responsabilidades
Administrador de configuración		Control de baselines y repositorio

Equipo de desarrollo		Cambios controlados por ramas
QA		Validación de cambios
Comité de cambios (CCB)		Aprueba cambios mayores

5. Elementos de configuración (EC)

ID EC	Elemento	Tipo	Ubicación	Responsable
EC-01	Código Backend	Código	repo/src/backend	
EC-02	Código Frontend	Código	repo/src/frontend	
EC-03	Documentación	Doc	repo/docs	

6. Identificación de configuración

- Convención de nombres:
 - Ramas: main, develop, feature/*, hotfix/*
 - Releases: v1.0.0, v1.0.1
 - Tags: release-YYYYMMDD
- Estructura del repositorio:
 - /src
 - /docs
 - /tests
 - /deploy

7. Control de versiones

- Herramienta: Git (GitHub/GitLab)
- Política de commits:
 - Mensajes claros: tipo: descripción
 - Ej.: fix: corregir validación de login
- Política de merge:
 - PR obligatorio
 - revisión por al menos 1 integrante

8. Control de cambios

8.1 Solicitud de cambio (RFC)

Información mínima:

- ID del cambio:
- Descripción del cambio:
- Impacto:
- Prioridad:
- Aprobación:

8.2 Flujo de aprobación (CCB)

1. Se registra el cambio
2. Se evalúa impacto y riesgos
3. Se aprueba/rechaza
4. Se implementa el cambio
5. Se prueba y documenta
6. Se actualiza baseline

9. Baselines (líneas base)

- **Baseline funcional:** requisitos y alcance aprobado
- **Baseline de desarrollo:** versión estable en develop
- **Baseline de release:** versión en main lista para despliegue

Baseline	Contenido	Fecha	Responsable
BL-01	Requisitos RF/RNF		
BL-02	Sprint 1 entregable		

10. Auditorías de configuración

- Frecuencia: semanal / por sprint
- Verificar:
 - integridad del repositorio,
 - trazabilidad,
 - existencia de documentación actualizada.

11. Reportes y métricas

- Cambios aprobados vs rechazados
- Bugs críticos por versión
- Frecuencia de releases

12. Herramientas de apoyo

- GitHub / GitLab
- Issues / Trello / Jira
- CI/CD (opcional)

13. Aprobación

Nombre Rol Firma Fecha