



Nombre: Chicaiza Andrea

Fecha: 09/11/2025

Control de versiones distribuido para trabajo colaborativo e historial de cambios.

Git

Control de versiones centralizado con historial completo y reversión fácil.

Subversion (SVN)

Automatización para gestionar configuraciones de servidores y entornos consistentes.

Ansible

Gestión de configuraciones mediante contenedores para desarrollo y producción.

Docker

Automatización de integración que soporta construcción y despliegue continuo.

Jenkins

Previene pérdida de información

Eleva calidad mediante auditorías

Facilita trabajo paralelo

Soporta recuperación de errores

Incrementa agilidad en despliegues

Beneficios

Gestión de Configuración de Software (SCM)

Definición

Proceso sistemático de gestión

Asegura integridad de productos

Garantiza trazabilidad completa

Evita conflictos entre desarrolladores

Objetivos

Control de cambios

Aprobar, documentar y hacer rastreable cada modificación.

Rastreo continuo

Monitorear la evolución desde el inicio hasta el presente.

Gestión de versiones

Manejar distintas versiones y permitir reversiones cuando sea necesario.

Proteger integridad

Verificar que cambios sean coherentes con requisitos del sistema.

Reducir errores

Disminuir fallas por configuraciones erróneas o inconsistentes.

Mejorar comunicación

Promover coordinación estructurada entre desarrolladores y equipos.

Componentes

Define ítems a gestionar

Identificación de Configuración

Autoriza y registra cambios formalmente

Control de Configuración

Historial para restauración

Gestión de Versiones y Lanzamientos

Valida cumplimiento de estándares

Auditoría y Validación

Consistencia dev/test/prod

Gestión de Entornos