

Informe técnico de plan de trabajo para construcción de software GA7-220501096-AA1-
EV01.

Fabian Rodríguez Álzate

Nicolas Tamayo Jimenez

Instructor: José Carmelo Álzate

Servicio nacional de aprendizaje SENA

Tecnología en análisis y diseño de software ADSO

Ficha: 2977373

Agosto de 2025

Introducción

Este documento evidencia el informe del plan de trabajo para la construcción del software, donde se exponen las herramientas, tecnologías y prácticas específicas a los procesos de control de versiones e integración continua, tales como aquellas necesarias para propiciar un desarrollo de calidad mediante el trabajo en colaboración.

En la actualidad del desarrollo del software, la integración continua permite realizar con frecuencia la construcción, la prueba y la validación del código. De esta forma, los errores que aparecen de manera temprana pueden ser minimizados y se pueden acelerar las entregas. En este sentido, este proyecto usará herramientas modernas para garantizar que existe una colaboración efectiva entre las personas miembros del equipo de desarrollo, trazabilidad con respecto a los cambios realizados en el código fuente y control de la calidad del software para con sus respectivas iteraciones.

Esta estrategia establece el uso de Git como sistema de control de versiones distribuido, y de GitHub como plataforma para hospedar repositorios y colaborar. Por lo cual, permite establecer una gestión adecuada del código mediante ramas (branches), control de cambios, revisiones (pull requests/merge requests) y seguimiento de incidencias.

Objetivo

El diseño y documentación de un plan de trabajo técnico para la construcción del software, usando herramientas modernas de control de versiones e integración continua que aseguren un desarrollo colaborativo, eficiente y con máxima calidad del software.

Diferencia versionamiento locales y remotos

Local: los cambios se manejan en la propia maquina donde se pueden crear ramas, hacer commits.

Remotos: Es el repositorio que se encuentra alojado en una plataforma como GitHub, GitLab o Bitbucket permitiendo subir cambios locales, bajar cambio de otros colaboradores y la colaboración entre equipos.

Versión Local	Versión remota
Esta en el computador personal, repositorio local.	Esta en un servidor remoto (GitHub, GitLab)
No requiere internet	Necesita internet para subir o bajar cambios
Solo se ven los cambios hasta que se suban	Los demás pueden ver los cambios cuando los suben al remoto.

Sistema control de versiones

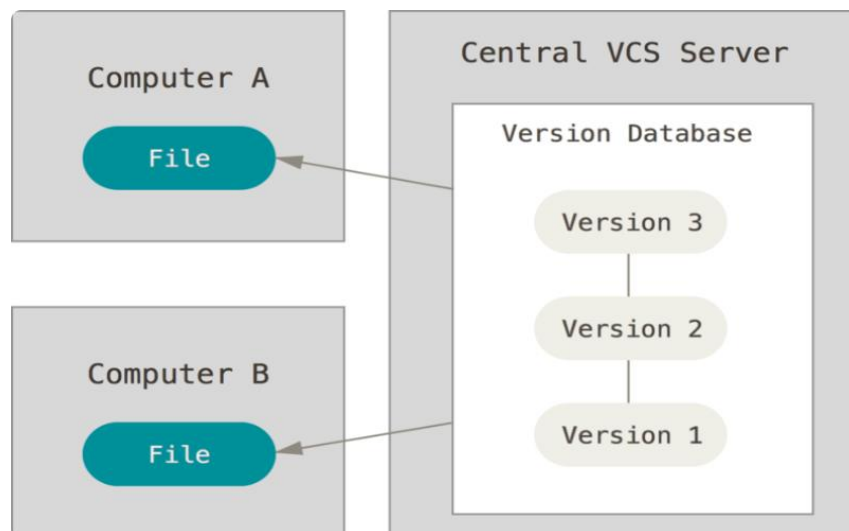
Un sistema de control de versiones es una herramienta que permite a los desarrolladores registrar los cambios realizados en el código fuente de un proyecto a lo largo del tiempo. Estos sistemas permiten llevar un registro de todas las modificaciones realizadas en los archivos, así como gestionar colaboraciones entre múltiples personas sin riesgo de sobrescribir el trabajo de otros.

Tipos de Sistemas de Control de Versiones

1. **Control de versiones local:** El control de versiones se maneja de manera local en la máquina del usuario. Los cambios no se comparten con otras personas, lo que limita la colaboración. Aunque puede ser una práctica común y fácil de implementar, resultaría muy riesgosa, ya que la posibilidad de introducir errores,

olvidar la ubicación del directorio, sobrescribir por accidente un archivo o nombrar de forma incorrecta un archivo es muy alta.

2. **Control de versiones centralizado:** Utiliza un repositorio central en el que todos los desarrolladores interactúan, pero los cambios no se almacenan localmente en los equipos. En los sistemas de control de versiones centralizados se dispone de un servidor donde se almacenan todos archivos y a partir del cual varios clientes pueden realizar descargas.



3. **Control de versiones distribuido:** Cada desarrollador tiene una copia completa del repositorio, lo que les permite trabajar de manera más independiente. Posteriormente, pueden sincronizar sus cambios con otros repositorios.

GIT

Git es un sistema de control de versiones distribuido que tiene la finalidad de registrar los cambios que se realizan en uno o varios archivos de código fuente durante un periodo de tiempo. Permite que varios desarrolladores puedan trabajar a la vez en un proyecto con la máxima eficiencia, seguridad y organización. Git fue creado en 2005 por las necesidades que afrontaba la comunidad del kernel de Linux, hoy por hoy es una herramienta que se utiliza en el desarrollo de software por su Velocidad, diseño sencillo, soporte para múltiples desarrollos no lineales, debía ser un sistema totalmente distribuido, debía ser capaz de manejar eficientemente proyectos de gran envergadura.

El sistema control de versiones GIT maneja tres estados:

Confirmado: Inicia con los datos se encuentra almacenados de forma segura en la base local se considera una versión concreta.

Preparado: Indica un archivo modificado que ha sido marcado en su versión actual para una próxima confirmación.

Modificado: Indica un que el archivo ha sufrido alguna modificación, pero esta aún no se ha preparado ni confirmado en la base de datos.