

**INFORME TÉCNICO DE PLAN DE TRABAJO PARA
CONSTRUCCIÓN DE SOFTWARE**

Laura Daniela Gómez González

**CENTRO DE FORMACIÓN PARA EL DESARROLLO RURAL Y MINERO
SENA**

2834894: ANALISIS Y DESARROLLO DE SOFTWARE

Instructora. LUZ KARIME CASTELLANOS JOYA.

FEBRERO 2025.

INTRODUCCIÓN

La Integración Continua es una práctica de desarrollo de software que se ha vuelto cada vez más popular en los últimos años. Esta práctica implica la automatización del proceso de construcción, prueba y despliegue de software, lo que permite a los equipos de desarrollo trabajar de manera más eficiente y rápida. En esencia, la Integración continua implica la integración de código frecuente y la ejecución de pruebas automatizadas para detectar errores temprano en el proceso de desarrollo.

OBJETIVO

El principal objetivo de esto es elegir una herramienta de versionamiento sobre la cual se va a trabajar a futuro, lo cual nos ayudará a futuro en el proyecto a desarrollar y la facilidad que nos proporciona a la hora de crear los distintos repositorios que nos permitirán desarrollar una solución práctica para futuras versiones y evitar alguna pérdida de información importante.

ELECCIÓN DE HERRAMIENTA DE VERSIONAMIENTO A UTILIZAR

De acuerdo con el material de estudio proporcionado, para el proyecto se elegirá Git y Github, teniendo en cuenta las siguientes características:

- ▶ **Versionar tu código:** Esto quiere decir guardar en determinado momento los cambios efectuados a un archivo o conjunto de archivos, con la oportunidad que tener acceso a ese historial de cambios, ya sea para regresar a una de esas versiones o para hacer comparaciones entre ellas. En este punto hay que destacar las ventajas frente a herramientas como Subversión y Perforce; que son usadas para este mismo fin, la principal es que no necesitarías montar, mantener, hacer copias de seguridad, etcétera a un repositorio o servidor para proteger el código que escribes, en este caso Github es la plataforma que cumple esta función. Lo único que necesitas es instalar Git en tu equipo, crear tu cuenta en Github y luego de realizar algunas configuraciones sencillas y ejecutar los comandos adecuados podrás enviar tu código a Github. Es importante mencionar que la visibilidad de los proyectos será público o privado según lo decidas.
- ▶ **Aprender y experimentar:** Como muchos desarrolladores tienen de forma pública sus proyectos en Github es posible acceder a su código, leerlo, estudiarlo y aprender de él, e incluso podrías hacer cambios y experimentar sin afectar el código original, a esto último se le conoce como: hacer fork.
- ▶ **Contribuir:** Si luego de copiar un proyecto (hacer fork) haces ajustes que arreglan bugs o introducen una nueva funcionalidad, puedes proponerle al dueño del proyecto que integre tus cambios en su código. Eso lo puedes hacer enviando un pull request con todas tus modificaciones o novedades. Mediante un pull request el administrador del repositorio

original tendrá la posibilidad de revisar, testear o escribir comentarios sobre algunas líneas de tu propuesta, y finalmente si considera interesante tu contribución podrá realizar la fusión de esta con el código original (a esto se le conoce como hacer merge) o en caso contrario rechazar los cambios propuestos.

- ▶ **Trabajo en equipo:** Github es el lugar perfecto para trabajar conjuntamente en una idea con un amigo o colega, después de crear tu proyecto podrás invitar a otros usuarios como colaboradores, permitiendo así que estos lean y escriban directamente sobre tu repositorio.
- ▶ **Compatibilidad:** Github es una plataforma web, por tanto, es independiente del sistema operativo que utilices, y además Git, que es la herramienta que, si requiere instalación, es compatible con todos los sistemas; Linux, OSX y Windows.

CONCLUSIÓN

Si unimos todo: las ramas de funciones, el desarrollo distribuido, las solicitudes de incorporación de cambios y una comunidad estable. Todas estas razones o características de Git facilitan un flujo de trabajo ágil y veloz en el que se anima a los desarrolladores a compartir cambios más pequeños con mayor frecuencia.