Informe técnico   de   plan   de   trabajo   para construcción de software.

GA7-220501096AA1-EV01.

PRESENTADO POR:

LUIS EDUARDO CHALA SERRADA

**Ficha programa de formación: 2455277**



INSTRUCTOR:

[HENRY TORRES PERTUZ](https://sena.territorio.la/perfil.php?persona=1770103&grupo=1423502)

TECNOLOGÍA EN ANÁLISIS Y DESARROLLO DE SOFTWARE

SERVICIO NACIONAL DE APRENDIZAJE – SENA

CENTRO DE LOGÍSTICA Y PROMOCIÓN ECOTURÍSTICA DEL MAGDALENA​

2023

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 nosayudará a futuro en el proyecto a desarrollar y la facilidad que nosproporciona a la hora de crear los distintos repositorios que nospermitirán desarrollar una solución práctica para futuras versiones yevitar alguna pérdida de información importante.

SELECCIÓN 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:

GIT:

Flujo de trabajo en la rama de función

Las ramas brindan un entorno aislado para cada cambio en su código base. Cuando un desarrollador desea comenzar a trabajar en algo, sin importar el tamaño, crea una nueva rama en Git. Esto garantiza que la rama principal siempre contenga código de calidad de producción y al mismo tiempo evita que la rama principal sufra de cambios o fallos ya que

todo se realiza desde la nueva rama.

Desarrollo distribuido

Cada desarrollador obtiene su propio repositorio local, con un historial completo de confirmaciones. Tener un historial local completo logra que Git sea más rápido, ya que no se necesita una conexión de red para crear confirmaciones, inspeccionar versiones anteriores de un archivo o realizar diferencias entre confirmaciones ya realizadas.

Solicitudes de extracción

Una solicitud de extracción es una forma de solicitar a otro desarrollador que fusione una de sus ramas en su repositorio. Git facilita que los líderes del proyecto realicen un seguimiento de los cambios y permite a los desarrolladores iniciar debates sobre su trabajo antes de integrarlo con el resto de la base de código.

■Comunidad

Si tu equipo usa Git, lo más probable es que no tengas que capacitar a los nuevos empleados sobre flujo de trabajo, porque ya estarán familiarizados con el desarrollo distribuido.

Adicional a esto, Git es muy popular entre los proyectos de código abierto,

lo cual significa que es fácil aprovechar las bibliotecas de terceros y alentar a otros a crear su propio código fuente abierto gracias a las características de Git.

GITHUB

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 el 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 tú 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 últimose 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 equipoGithub 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 codesarrolladores a compartir cambios más pequeños con mayor frecuencia

CONCLUSIONES: El uso de GitLab puede ser muy beneficioso para los equipos de desarrollo de software, ya que ofrece una gran cantidad de funcionalidades que pueden facilitar el trabajo colaborativo y la automatización de tareas