

# TRUNK-BASED DEVELOPMENT

## INTRODUCCIÓN

¿Alguna vez te has preguntado sobre los beneficios de trabajar directamente en la rama principal? En este blog, exploraremos el mundo del desarrollo basado en tronco y cómo puede agilizar tu proceso de desarrollo.

## SIMPLICIDAD Y COLABORACIÓN

El desarrollo basado en tronco simplifica el proceso eliminando ramas separadas. Esto fomenta la colaboración continua y agiliza el flujo de trabajo. Imagina a todos trabajando juntos en una única rama principal compartida, asegurando que todos estén en la misma página y promoviendo el trabajo en equipo.

## MEJORES PRÁCTICAS

El desarrollo basado en tronco enfatiza dividir el trabajo en unidades manejables, habilitando revisiones de código regulares e implementando pruebas automatizadas. Estas prácticas aseguran la calidad y estabilidad de la base de código, contribuyendo a un producto más confiable.

## INTEGRACIÓN CONTINUA Y FUTURE FLAGS

La integración continua y las Future Flags son fundamentales para el desarrollo basado en tronco. La integración continua asegura que los cambios sean continuamente probados e integrados, evitando problemas de integración. Las Future Flags facilitan el control de la exposición y lanzamiento de características, haciendo el desarrollo más adaptable y eficiente.

## RAMAS DE CORTA DURACIÓN Y PULL-REQUESTS

El desarrollo basado en tronco promueve ramas de corta duración y pull-requests para revisión de código. Esta práctica asegura que los cambios sean seguros y bien probados antes de su integración en la rama principal, manteniendo la estabilidad de la base de código.

## COLABORACIÓN EFICIENTE

Trabajando en una única rama principal compartida, el desarrollo basado en tronco permite una colaboración más rápida y eficiente. Este enfoque mitiga conflictos al fusionar el sistema y proporciona una mejor visibilidad sobre el proyecto, facilitando el trabajo en equipo sin problemas.

## **DESAFÍOS Y CONSIDERACIONES**

Mientras que el desarrollo basado en tronco ofrece numerosos beneficios, también trae complejidad y requiere consideración cuidadosa. Trabajar directamente en la rama principal puede impactar la estabilidad y aislamiento de pruebas, lo que puede llevar a problemas de calidad y complejidad adicional en el proceso de desarrollo.

## **METODOLOGÍA ÁGIL Y EFICIENTE**

El desarrollo basado en tronco ofrece una metodología ágil y eficiente, especialmente adecuada para ciclos de entrega rápidos. Minimiza los esfuerzos de coordinación para equipos de desarrollo pequeños y promueve un enfoque simplificado y efectivo para la entrega de proyectos.

## **CONCLUSIONES**

En conclusión, el desarrollo basado en tronco proporciona un enfoque simplificado y colaborativo, ofreciendo numerosos beneficios como la integración continua, colaboración eficiente y metodologías ágiles. Aunque presenta desafíos, sus ventajas lo hacen una opción convincente para equipos de desarrollo modernos.

## **RECURSOS BIBLIOGRÁFICOS**

[https://www.youtube.com/watch?v=PtprqRH7EDQ&t=193s&ab\\_channel=Feregrino](https://www.youtube.com/watch?v=PtprqRH7EDQ&t=193s&ab_channel=Feregrino)

<https://www.abtasty.com/blog/git-branching-strategies/>

<https://dev.to/marianocodes/por-que-trunk-based-development-i5n>

<https://blog.logrocket.com/product-management/a-guide-to-trunk-based-development/>

<https://se-radio.net/2023/05/se-radio-564-paul-hammant-on-trunk-based-development/>