

Ciclo de desarrollo

El ciclo de desarrollo que describes es un proceso típico de desarrollo de software que sigue las mejores prácticas para garantizar la calidad, colaboración y entrega regular de nuevas características o mejoras en un proyecto.

Ciclo de desarrollo:

1. Desarrollo:

- *Trabajo por ramas (git flow schema):* En este enfoque, se utiliza un esquema de ramificación en Git, como el «git flow», que organiza el trabajo en ramas específicas para diferentes propósitos. Por ejemplo, hay ramas de desarrollo, ramas de características, ramas de correcciones de errores, etc. Esto ayuda a mantener un flujo de trabajo estructurado y aislado.
- *Código testable:* Se enfatiza la escritura de código que es fácil de probar mediante pruebas unitarias y pruebas de integración. El código testable es esencial para garantizar que las nuevas características o cambios no rompan funcionalidades existentes.
- *Código y plantillas reusables:* Se promueve la reutilización de código y plantillas cuando sea posible. Esto ahorra tiempo y esfuerzo, reduce la duplicación y mejora la consistencia en el código.

2. PR a rama de desarrollo:

- Después de completar el desarrollo de una característica o corrección de errores en una rama separada, se crea una solicitud de extracción (Pull Request o PR) para fusionar los cambios en la rama de desarrollo principal. Esto permite una revisión antes de la integración y ayuda a

mantener un historial limpio del repositorio.

3. CI: Test unitarios, funcionales y análisis estático de código:

- *Test unitarios:* Se ejecutan pruebas unitarias automatizadas en el código para verificar que las funciones y componentes individuales funcionen correctamente. Esto ayuda a identificar errores temprano.
- *Test funcionales:* Se realizan pruebas funcionales automatizadas para verificar que las características en su conjunto funcionen según lo esperado. Esto garantiza que el software cumple con los requisitos del usuario.
- *Análisis estático de código:* Se utilizan herramientas de análisis estático de código para buscar problemas potenciales, como convenciones de estilo de código, posibles errores y vulnerabilidades de seguridad.

4. Revisión de código por parte de miembros del equipo:

- Antes de que se fusionen los cambios en la rama de desarrollo, al menos dos miembros del equipo revisan el código propuesto en la PR. Esto permite detectar problemas, mejorar la calidad del código y asegurarse de que se sigan las pautas de codificación del proyecto.

5. Releases semanales:

- Se programa la liberación de nuevas versiones del software de manera regular, en este caso, semanalmente. Esto permite a los usuarios recibir nuevas características y correcciones de errores de forma predecible y constante.