

GUÍA PARA LA IMPLEMENTACIÓN Y ESTRUCTURACIÓN DE MENSAJES DE COMMITS.

Esta guía establece las normas y mejores prácticas para estandarizar los mensajes de commit en el proyecto con el fin de facilitar la colaboración, el rastreo de cambios, y el análisis histórico del código.

Formato Estándar del Mensaje de Commit

Cada mensaje de commit debe seguir este formato:

<tipo>(WI), breve descripción

Desglose del formato:

1. **Tipo:** Describe la naturaleza del cambio. Los tipos permitidos son:
 - feat: Introducción de una nueva funcionalidad.
 - fix: Corrección de errores en el código.
 - docs: Cambios o adiciones a la documentación.
 - test: Adiciones o modificaciones relacionadas con pruebas.
2. **WI:** Nombre del Work Item asociado en español.
3. **Breve descripción:** Resumen claro y conciso del cambio realizado.

Ejemplos Prácticos de Mensajes

- feat: Community, agregar funcionalidades básicas del módulo.
- fix: Fakenodo, errores de codacy solucionados.
- docs: Documentación para la gestión de issues (Los cambios en documentación no tienen WI asociado)
- test: Validar registro, adición de pruebas unitarias.

Pautas Generales

1. **Consistencia:** Siempre usar el formato definido.
2. **Claridad:** Ser específico y directo al describir los cambios.
3. **Concisión:** Evitar mensajes genéricos como "update" o "fix bug".
4. **Revisión:** Antes de hacer un commit, verificar que el mensaje refleja los cambios realizados y el Work Item correcto.

Implementación

1. **Ubicación de la Guía:** La guía se encuentra en el archivo `GUÍA PARA LA IMPLEMENTACIÓN Y ESTRUCTURACIÓN DE MENSAJES DE COMMITS.pdf` en la carpeta de documentación del repositorio.
2. **Validación del Equipo:** Esta guía será revisada y aprobada por el equipo para garantizar su aplicabilidad.

Beneficios

- Historial de commits más legible y estructurado.
- Mayor facilidad para la colaboración entre desarrolladores.
- Mejora en la trazabilidad y vinculación con tareas.