

# Mini taller: Semana 2

Por: Sebastián Ávila Fica

A) Realice un listado resumen de las principales convenciones git utilizadas actualmente en el desarrollo de software.

## 1) Convenciones de Commit

Conventional Commits: Es un estándar que define un formato para los mensajes de commit, facilitando la generación de logs, el control de versiones semántico y la automatización de tareas.

- “feat”: Se usa para registrar la adición de una nueva funcionalidad en el programa.
- “fix”: Se usa para registrar correcciones de errores dentro del código del programa.
- “docs”: Se usa para registrar cambios en la documentación del programa (Usualmente en el README).
- “style”: Se usa para registrar cambios en el formato del código. Estos cambios no afectan la funcionalidad en la que se trabaja. (Ex: espacios en blancos, comas, etc.)
- “refactor”: Se usa para registrar refactorizaciones en el código que no añaden ni modifican funcionalidades en el programa.
- “test”: Se usa para registrar la adición o corrección de pruebas.
- “chore”: Se usa para registrar tareas de mantenimiento que no afectan las funcionalidades. Usualmente se le atribuye a la actualización de dependencias, configuraciones, etc.

B) Simule 5 commits, escribiendo su título y descripción en base a las convenciones estudiadas.

- **feat(auth): Añadir soporte para autenticación de dos factores.**

Implementa el flujo de autenticación de dos factores utilizando códigos enviados por correo electrónico. Este cambio mejora la seguridad de la cuenta de usuario.

- **fix(forms): Corregir error en la validación del formulario de registro.**

Se corrige un problema que impedía la validación adecuada del campo de correo electrónico en el formulario de registro. Ahora se muestran mensajes de error correctamente cuando se ingresan datos no válidos.

- **docs(readme): Actualizar la guía de instalación.**

Se ha revisado y actualizado la sección de instalación del archivo README para reflejar los nuevos requisitos del sistema y los pasos de instalación simplificados.

- **refactor(errors): Refactorizar funciones de manejo de errores en el módulo de pagos.**

Se mejoró la estructura de las funciones de manejo de errores para el módulo de pagos, haciendo el código más limpio y mantenible. No se realizaron cambios en la funcionalidad.

- **style(lint): Estilizar el código según las reglas de eslint.**

Se han realizado ajustes en el formato del código para cumplir con las reglas definidas en el archivo de configuración de eslint. Los cambios incluyen la corrección de la indentación, espacios en blanco y el estilo de las comillas.