

A) Listado resumen de las principales convenciones (Reglas) Git utilizadas en el desarrollo de software:

- **Commits:**

- Es recomendable hacer commits pequeños y con frecuencia para facilitar la revisión del código y la identificación de errores.
- Estos a la vez deben de ser claros y descriptivos.
- Los mensajes del commit tienen que describir claramente los cambios realizados.
- La convención común es usar un verbo en tiempo presente, como "añade" o "corrige".

- **Uso de ramas (branches):**

- Se deben crear ramas separadas para cada nueva funcionalidad o corrección de errores.
- Utiliza main o master solo para código listo para producción.
- Antes de fusionar una rama, asegúrate de que está actualizada con la rama principal (main o master) y de que todos los conflictos se han resuelto.
- Usar convenciones claras para nombrar ramas, como feature/nueva-funcionalidad, fix/correccion-bug, o hotfix/urgente, para identificar fácilmente el propósito de cada rama.

- **Revisiones de código:**

- Realizar revisiones de código antes de hacer merge a la rama principal para asegurar la calidad del código.
- Evitar hacer push de commits que rompan la compilación o introduzcan errores en el código. Realiza pruebas locales antes de hacer push.

- **Etiquetas (tags):**

- Utilizar etiquetas (git tag) para marcar versiones específicas del software, como lanzamientos o hitos importantes.

B) Simulación de 5 commits:

1)

- **Nombre:** perf(cache): optimizar el uso de caché en el servidor.
- **Descripción:** Mejorar el rendimiento del servidor mediante la implementación de un sistema de caché más eficiente para reducir la carga en la base de datos y acelerar la respuesta del servidor.

2)

- **Nombre:** chore(deps): actualizar dependencias de seguridad.
- **Descripción:** Actualizar las dependencias del proyecto para resolver vulnerabilidades de seguridad detectadas en las versiones anteriores para así asegurar la estabilidad y seguridad del entorno de desarrollo.

3)

- **Nombre:** feat(notification): implementar sistema de notificaciones en tiempo real.
- **Descripción:** Añadir un sistema de notificaciones en tiempo real utilizando WebSockets para alertar a los usuarios de eventos importantes como mensajes nuevos o actualizaciones en sus cuentas.

4)

- **Nombre:** refactor(auth): reestructurar lógica de manejo de tokens.
- **Descripción:** Reestructurar el manejo de tokens en el sistema de autenticación para mejorar la claridad del código y facilitar el mantenimiento. Se separaron las funciones de generación y verificación de tokens.

5)

- **Nombre:** docs(API): actualizar la documentación del endpoint de búsqueda.
- **Descripción:** Actualizar la documentación de la API para incluir detalles sobre el nuevo comportamiento del endpoint de búsqueda, incluyendo nuevos parámetros opcionales y ejemplos de uso.

Bibliografía:

<https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell>

<https://cbea.ms/git-commit/>

<https://hackmd.io/@bytewareteam/rkFdKUURw>

<https://www.conventionalcommits.org/es/v1.0.0-beta.2/#:~:text=fix%3A%20un%20commit%20de%20tipo,PATCH%20en%20el%20versionado%20semántico>