

## Commits

Los commits deben representar un objetivo específico, no se debe mezclar más de una funcionalidad en el mismo commit, si tributan a objetivos diferentes, son commits diferentes.

Para crear un nuevo commit se debe analizar la funcionalidad semánticamente, evitando hacer mención a elementos redundantes como los archivos modificados o las fechas en que se modificaron, ya que el control de versiones se encarga de identificar esos elementos. A continuación, se define la estructura que deben llevar los commits:

**Título:** El título debe ser breve (no más de 50 caracteres) y comenzar con un verbo en infinitivo que describa la acción realizada. Además, el título debe comenzar en mayúscula y no debe terminar con un punto.

El título debe estar prefijado con un código de representación en minúsculas envuelto en corchetes cuadrados “[ ]”, los prefijos se describen en la tabla a continuación de la estructura.

**Descripción:** Esta sección es opcional, pero se recomienda que esté siempre disponible. Proporciona una descripción más detallada de los cambios realizados, desde una perspectiva más conceptual de la funcionalidad. En esta sección, se pueden incluir detalles sobre la implementación, las decisiones de diseño tomadas, los problemas resueltos, entre otros. El cuerpo debe estar separado del título por una línea en blanco y no debe exceder 72 caracteres de ancho. El texto debe comenzar por mayúscula y terminar en punto.

**Observaciones:** Esta sección es opcional y se utiliza para agregar información adicional al commit, como el número de tarea asociado, los problemas cerrados, los cambios en la documentación, entre otros. Al igual que el cuerpo, el pie de página debe estar separado del cuerpo por una línea en blanco, el texto debe comenzar por mayúscula y terminar en punto.

### Ejemplo de un commit

[feat] Añadir autenticación en dos pasos

El código de verificación se puede enviar mediante diferentes canales, estos son: SMS, llamada telefónica, whatsapp o email. También se habilitan web services que permiten agregar, editar y eliminar canales de autenticación.

Responde a la característica solicitada en el issue #2000, la documentación de los webservices estará disponible en el siguiente commit.

**Tabla de prefijos**

<b>Prefijo</b>	<b>Descripción</b>
feat	Utilizado para indicar la implementación de una nueva característica o funcionalidad
fix	Utilizado para indicar la corrección de un error o problema existente.
docs	Utilizado para indicar cambios en la documentación del proyecto.
style	Utilizado para indicar cambios en el formato o estilo del código, como corrección de espacios en blanco, nombres de variables o comentarios.
refactor	Utilizado para indicar cambios en el código que mejoran la calidad o legibilidad del mismo, pero que no agregan nuevas características o corrigen errores.
test	Utilizado para indicar la implementación o mejora de pruebas en general.
config	Utilizado para indicar cambios en la configuración del proyecto.
dependencies	Utilizado para indicar cambios en las dependencias del proyecto.
perf	Utilizado para indicar mejoras de rendimiento.
ci	Utilizado para indicar cambios en la configuración de integración continua.
revert	Utilizado para indicar la reversión de un commit anterior.
wip	Utilizado para indicar que el commit está en progreso y aún no está completo.
security	Utilizado para indicar mejoras de seguridad.
merge	Utilizado para indicar la fusión de ramas o branches.
temp	Utilizado para indicar cambios temporales que se revertirán en una versión futura.
cleanup	Utilizado para indicar cambios que eliminan código o archivos no utilizados.

## Ramas

Cada proyecto deberá contar con las ramas Main y Develop.

La rama "Main" es la rama principal del proyecto, y es equivalente a la versión en producción. Esta rama solo debe contener el código que ha sido debidamente probado y aprobado para ser implementado en el software en producción.

Por otro lado, la rama "Develop" se utiliza para realizar integraciones entre los diferentes desarrolladores. En esta rama se mezclan los cambios y nuevas funcionalidades de los diferentes miembros del equipo, y se llevan a cabo las pruebas necesarias para asegurar que todo el código se integra sin problemas.