



Mensajes de confirmación

Curso de Fundamentos

Introducción

Esta lección explicará la importancia de los buenos mensajes de confirmación, cómo escribirlos, cuándo confirmarlos y por qué es tan importante tener un historial de buenos mensajes de confirmación.

Resumen de la lección

Esta sección contiene una descripción general de los temas que aprenderá en esta lección.

- Cómo escribir un mensaje de confirmación significativo.
- Por qué son importantes los mensajes de confirmación significativos.
- Cuándo comprometerse

¿Son los mensajes de confirmación tan importantes que merecen su propia lección?

¡Sí! Permítame darle una breve lista de razones:

- Cuando solicites un empleo, los empleadores revisarán tus proyectos en GitHub y tu historial de commits. Tener buenos commits como desarrollador novato te ayudará a destacar.

- Tener un buen historial de mensajes de confirmación le permitirá a usted (u otros desarrolladores que trabajen en su código) ver rápidamente qué cambios se realizaron y por qué. ¡Esto es útil si se encuentra un error en el código que necesita ser corregido!
- Tener un buen historial de mensajes de confirmación también será útil si vuelves a un proyecto en el que estabas trabajando después de haberlo dejado por un tiempo. Es probable que no recuerdes tu proceso de pensamiento y los cambios que realizaste cuando escribiste el código inicialmente.

Compromisos malos y buenos

A la hora de escribir commits, es fundamental saber cómo hacerlo de forma eficaz. A continuación, se muestra un ejemplo de un mensaje de commit incorrecto:

```
1 | fix a bug
```

Aunque describe lo que hiciste, el mensaje es demasiado vago, lo que deja a los demás desarrolladores de tu equipo confundidos. Un buen mensaje de confirmación explicará el **motivo** de tus cambios. En otras palabras, un mensaje de confirmación describe qué problema resuelven tus cambios y cómo lo resuelven.

Las confirmaciones efectivas constan de dos partes separadas: un asunto y un cuerpo:

Sujeto

Un breve resumen del cambio que realizó en el proyecto.

```
1 | This is the change I made to the codebase
```

GitHub tiene un límite de 72 caracteres, por lo que recomendamos mantener los asuntos de tus confirmaciones dentro de esta cantidad.

Cuerpo

Una descripción concisa pero clara de lo que hiciste.

1 | Describe the problem your commit solves and how.

Ahora que aprendimos el secreto para crear un buen mensaje de confirmación, intentemos arreglar el mensaje de confirmación anterior:

```
1 | Add missing link and alt text to the company's logo
2 |
3 | Screen readers won't read the images to users with disabilities
```

¡Ahh, eso está mejor! :) Ahora, los desarrolladores pueden comprender mejor este mensaje de confirmación porque hace lo siguiente:

- Proporciona un asunto que especifica la acción de su código (por ejemplo, "Agregar enlace faltante y texto alternativo al logotipo de la empresa").
- Contiene un cuerpo que proporciona una descripción concisa pero clara de por qué fue necesario realizar la confirmación (por ejemplo, "Los lectores de pantalla no leerán las imágenes a los usuarios con discapacidades sin esta información").
- Separa el asunto del cuerpo con una línea nueva o en blanco. Esta es una práctica recomendada que recomendamos seguir. Hace que los mensajes de confirmación sean más fáciles de leer para otros desarrolladores.

Cómo comprometerse con un asunto y cuerpo en el mensaje

Hasta ahora, se le ha indicado que realice la confirmación con `git commit -m <message>`. Para realizar una confirmación con asunto y cuerpo en el mensaje, la forma más sencilla es escribir `git commit` sin el `-m` indicador ni el argumento del mensaje.

Al hacerlo, se abrirá una nueva pestaña de Visual Studio Code si habías [configurado Visual Studio Code como editor de Git](#). Puedes eliminar cualquier comentario e ingresar mensajes de varias líneas. Cuando guardes y cierres la pestaña, se creará tu confirmación.

Cuándo comprometerse

Una buena forma de ver una confirmación es como una "instantánea" de tu código en el momento en que se creó. Esa versión de tu código hasta ese momento se guardará para que puedas volver a ella o mirarla.

Al escribir código, se considera que la mejor práctica es confirmar el código cada vez que se produce un cambio significativo en el código. Esto creará una línea de tiempo de su progreso y mostrará que el código terminado no apareció de la nada.

En otras palabras, haz una confirmación si logras que un fragmento de código en el que estás trabajando funcione como quieres, corrige un error tipográfico o corrige un error. A medida que ganes experiencia, desarrollarás una mejor idea de lo que debería confirmarse.

Llegará un momento en el que estés trabajando en un proyecto y FINALMENTE consigas algo perfecto (este sería un buen momento para confirmar el proyecto), y luego, tal vez 30 segundos o unos días después, se estropee. No tienes idea de qué has cambiado, todo *parece* estar igual y no recuerdas haber editado esa línea, pero, por desgracia, ya no funciona como quieres. Podrás volver a revisar tu historial de confirmaciones y revertir tu código a la última confirmación que hiciste cuando hiciste funcionar esa parte por primera vez o volver atrás y ver cómo se veía tu código en ese momento.

Asignación

1. Este artículo, [Cómo escribir un mensaje de confirmación de Git](#), cubre todos los aspectos básicos sobre cómo escribir buenos mensajes de confirmación. Todo el artículo es excelente e informativo, pero lo esencial del artículo son las "siete reglas de un gran mensaje de confirmación".

Consejos y cosas para recordar

- El uso de VSCode como editor de texto (debería haberlo configurado en la sección Conceptos básicos de Git) le permitirá crear fácilmente mensajes de confirmación de varias líneas, ver fácilmente la longitud de caracteres de cada línea y le permitirá usar [las extensiones de revisión ortográfica de VSCode](#) para asegurarse de que su ortografía sea correcta.

- Utilice una voz activa: "Arreglar generador de tarjetas".
- Evite utilizar mensajes de confirmación vagos como "guardado" o "actualizado".
- ¡Comprométete temprano y con frecuencia!

Comprobación de conocimientos

Las siguientes preguntas son una oportunidad para reflexionar sobre temas clave de esta lección. Si no puede responder una pregunta, haga clic en ella para revisar el material, pero tenga en cuenta que no se espera que memorice o domine este conocimiento.

- [¿Cuáles son dos beneficios de tener mensajes de confirmación bien escritos y un buen historial de confirmaciones?](#)
- [¿Cuántos caracteres debe tener la línea de asunto de tu mensaje de confirmación?](#)

Recursos adicionales

Esta sección contiene enlaces útiles a contenido relacionado. No es obligatoria, por lo que se la puede considerar complementaria.

- Una forma de formular mensajes de confirmación con mucha información es seguir una plantilla. [Las confirmaciones convencionales](#) son una de las muchas plantillas de mensajes de confirmación que puedes explorar.
- Explora este increíble video tutorial sobre confirmaciones convencionales → [Enlace al video completo](#) . El video muestra la plantilla de confirmaciones convencionales del recurso anterior. También menciona la creación de lanzamientos y muestra el uso de algo llamado "Yarn". Estas dos partes están fuera del alcance de esta parte del curso, así que no te preocupes por ellas y, en cambio, concéntrate en la plantilla de confirmación.

 [Mejorar en GitHub](#)  [Informar un problema](#)

[Ver registro de cambios de la lección](#)



Ver curso

[Marcar como completado](#)[Próxima lección](#)

¡Apóyanos!

El Proyecto Odin está financiado por la comunidad. ¡Únase a nosotros para ayudar a estudiantes de todo el mundo apoyando el Proyecto Odin!

[Más información](#)[Dona ahora](#)

THE ODIN PROJECT

Educación en codificación de alta calidad mantenida por una comunidad de código abierto.



Sobre nosotros

[Acerca de](#)[Equipo](#)[Blog](#)[Casos de éxito](#)

Apoyo

[Preguntas frecuentes](#)

Guías

[Guías de la comunidad](#)[Guías de instalación](#)

Legal

[Términos](#)[Privacidad](#)

[Contribuir](#)

[Contáctenos](#)

© 2025 El Proyecto Odin. Todos los derechos reservados.