

# Code Reviews

Xurde García Fernández  
Enol Xesús González García

## Qué es

Una Code Review es el proceso por el cual los miembros de un equipo revisan el código en conjunto la implementación de otro miembro de una historia de usuario. Se trata de aportar ideas sobre mejoras en la implementación, posibles refactorizaciones, encontrar bugs o ampliar la cobertura de los tests. El objetivo es determinar qué se está haciendo bien para seguir haciéndolo y qué se puede mejorar.

## Qué mirar

Para realizar correctamente una Code Review hay que fijarse en estos aspectos:

- **Propósito:** saber cuál es la razón que dio lugar a los nuevos cambios y comprobar que estos cambios cumplen ese propósito.
- **Legibilidad y estilo:** comprobar que el código es fácil de entender y si cumple con las normas de estilo y convenciones adoptadas por el equipo.
- **Mantenibilidad:** revisar las pruebas para ver si cubren los casos más interesantes y si está actualizada la documentación
- **Seguridad:** comprobar que se haya implementado autorización, autenticación y una validación adecuada de los distintos formularios que el usuario tenga acceso.
- **Arquitectura Software:** mirar si los nuevos cambios cumplen con los principios y restricciones técnicas que se les hayan puesto al equipo.
- **Rendimiento:** revisar que el consumo de recursos y los tiempos de respuesta no sean excesivos.

## Como llevarlo a cabo

Ya que este es un proceso manual y humano hay que saber manejar algunas situaciones entre compañeros. Posibles enfoques para solucionar esto pueden ser hacer preguntas para comprender mejor el funcionamiento de una parte del código, discutir en persona las distintas correcciones ya que por escrito es bastante más complicado e impersonal, explicar razonadamente el porqué de las correcciones para que no parezca puro capricho y comentar al final posibles sugerencias para mejorar la dinámica de trabajo.

## Motivos para hacer una code review

Se incrementa el compromiso con el proyecto y aumenta la motivación del desarrollador al recibir buenas críticas por sus compañeros. Mejora la comunicación entre los miembros del equipo, que es uno de los mayores problemas en los equipos de desarrollo. Aumenta el nivel de calidad en el equipo ya que los desarrolladores más veteranos ayudan a los menos experimentados. Mejora la consistencia porque el código es propiedad y responsabilidad de todo el equipo.

## Pasos de una code review

Establecer objetivos. Echar un primer vistazo. Ejecutar pruebas. Probar correcciones propuestas. Revisión en profundidad. Hacer la evaluación.

## Buenas prácticas

La principal buena práctica para realizar Code Reviews es el uso de pull requests, ya que a través de ellos, el resto del equipo podrá revisar los cambios realizados y generar un espacio de debate entre los miembros del equipo.

Además estás son algunas buenas prácticas a la hora de dar feedback (Siempre tener en cuenta que Compasión > Precisión):

1. **Hacer preguntas y no directamente ordenar cambios**
2. **Centrarse en el código y no culpar directamente a la persona**
3. **Dejar claro que es tú perspectiva, no algo universal:**
4. **Evitar el uso de sarcasmos**
5. **Evitar ser condescendiente**
6. **Usar Emojis**
7. **Justificar tu comentario a un pull request**
8. **Guiar al autor del código durante una Code Review**
9. **Valorar al autor del código**

## Cómo actuar al feedback

La empresa Google da unos consejos a sus trabajadores para actuar ante el feedback:

1. **No te lo tomes como algo personal:** Piensa que el feedback es un intento de ayudarte. Nunca respondas de manera enfadada a los comentarios de retroalimentación, aunque haya veces que las personas expresen su frustración.
2. **Arregla el código:** La primera opción para responder al feedback es aclarar el código y si no comentar o el código o como última instancia utilizar la herramienta de Code Review.
3. **Piensa por ti mismo:** No te tomes el feedback como un impedimento y analiza siempre si la persona que te ha dado retroalimentación está en lo cierto o no.
4. **Siempre intenta resolver los conflictos:** El objetivo es llegar a un consenso.

## Posibles retos

En una Code Review las personas implicadas tienen diferentes retos con los que lidiar. Vamos a dividirlo en dos: Los autores de código y los revisores de código.

**Autores de código:** Uno de los retos a los que se enfrentan es el no recibir feedback en un plazo adecuado de tiempo, además de recibir feedback insignificante, ya que muchas veces el feedback se centra en detalles insignificantes y encontrar buenos revisores de código. Además los autores de código encuentran difícil el documentar los cambios para la revisión y también les resulta duro aceptar un rechazo de su trabajo y prefieren una explicación de por qué se han rechazado los cambios. Algunos autores dicen también que las herramientas pueden bajar la velocidad de trabajo.

**Revisores de código:** Los code reviewers o revisores de código encuentran difícil encontrar tiempo para realizar una revisión de código además de que no siempre es fácil entender el propósito del código y las motivaciones de los cambios. También es difícil encontrar documentación relevante de los cambios. Destacar también que algunos expresan que es difícil entender el historial de comentarios y que hay una muestra de falta de entrenamiento en el proceso de revisión y que estas actividades no se valoran lo suficiente.

## Herramientas

Estas son algunas de las herramientas para realizar Code Review. Estas herramientas por lo general permiten revisar código, analizar las diferencias en los cambios y generar un espacio de debate: Estas son 4 herramientas importantes.

1. **GitHub:** Permite analizar las diferencias, comentar en línea y comprobar el historial de cambios.
2. **GitLab:** Permite una revisión de código asíncrona. Puede automatizar, seguir y reportar Code Reviews.
3. **Bitbucket:** Permite revisión de código y hacer merge a pull requests.
4. **Azure DevOps:** Ofrece discusiones encadenadas e integración continua en las pull request.

Hay otras herramientas como Crucible, Gerrit, Upsource, JetBrains Space, Review Board, AWS Code Commit, Gogs/Gitea, Collaborator y Helix Swarm.

## Conclusión

Tras haber buscado información acerca de las Code Review, creemos que son importantes de realizar ya que fomentan el espíritu colaborativo en los trabajos en equipo, que tan importantes son hoy en día en el mundo del desarrollo siendo además muy frecuente que los equipos de trabajo se encuentren en diferentes lugares y no todos juntos en el mismo espacio de trabajo.

## Bibliografía

<https://www.michaelagreiler.com/respectful-constructive-code-review-feedback/>

<https://adrianalonso.es/buenas-practicas/buenas-practicas-para-hacer-code-review-en-tu-equipo-de-desarrollo/>

<https://www.michaelagreiler.com/wp-content/uploads/2019/03/Code-Reviewing-in-the-Trenches-Understanding-Challenges-Best-Practices-and-Tool-Needs.pdf>

<https://google.github.io/eng-practices/review/developer/handling-comments.html>

<https://kinsta.com/es/blog/herramientas-de-revision-de-codigo/#anlisis-de-12-poderosas-herramientas-de-revisin-de-codigo>

<https://blog.jetbrains.com/space/2021/12/15/best-code-review-tools/>

<https://slashmobility.com/blog/2017/01/code-review-todo-lo-que-debes-saber/>

<https://adrianalonso.es/buenas-practicas/buenas-practicas-para-hacer-code-review-en-tu-equipo-de-desarrollo/>