

# Guía uso de Pull Requests en la asignatura Sistemas de Información

## Objetivo de este documento

Este documento contiene una guía de uso de la funcionalidad **Pull Request** proporcionada por GitHub y que será usada en las prácticas de laboratorio de la asignatura de Sistemas de Información.

## Historia

V1.0, P. Suárez-Otero, Febrero de 2024. Inicial

## Contenido

1.	Uso de Pull Requests .....	2
1.1	Desarrollador: Solicitud pull request.....	2
1.2	Revisor: Revisión de la pull request.....	3
1.3	Desarrollador: Corrección de defectos detectados en revisión .....	6
1.4	Revisor: Aprobación de la Pull Request.....	8
2.	Guías externas Github .....	8

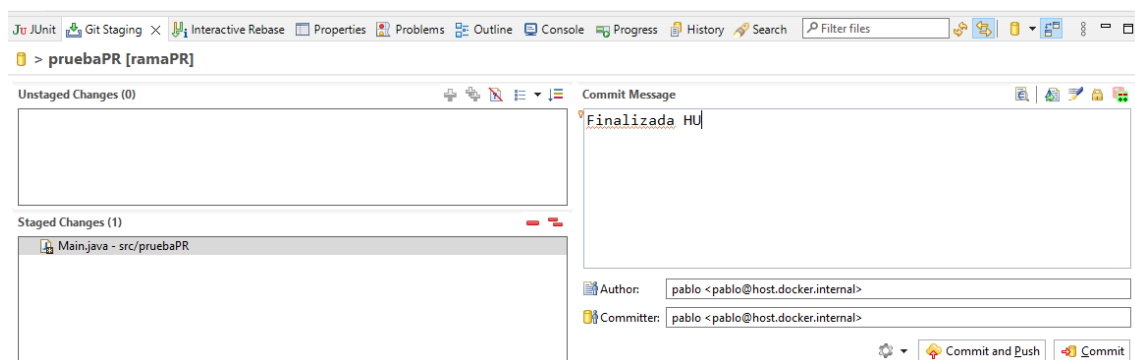
# 1. Uso de Pull Requests

Los Pull Requests consisten en la revisión y aceptación/rechazo por parte de un administrador o persona autorizada de código desarrollado en una rama del proyecto a incorporar a otra rama del proyecto. Normalmente, se realizan incorporaciones a la rama principal del proyecto, denominada por defecto en Github 'main' o 'master' desde Eclipse. Esta revisión permite la mejora de la calidad del código por parte de los desarrolladores antes de incorporarse a la rama principal del proyecto.

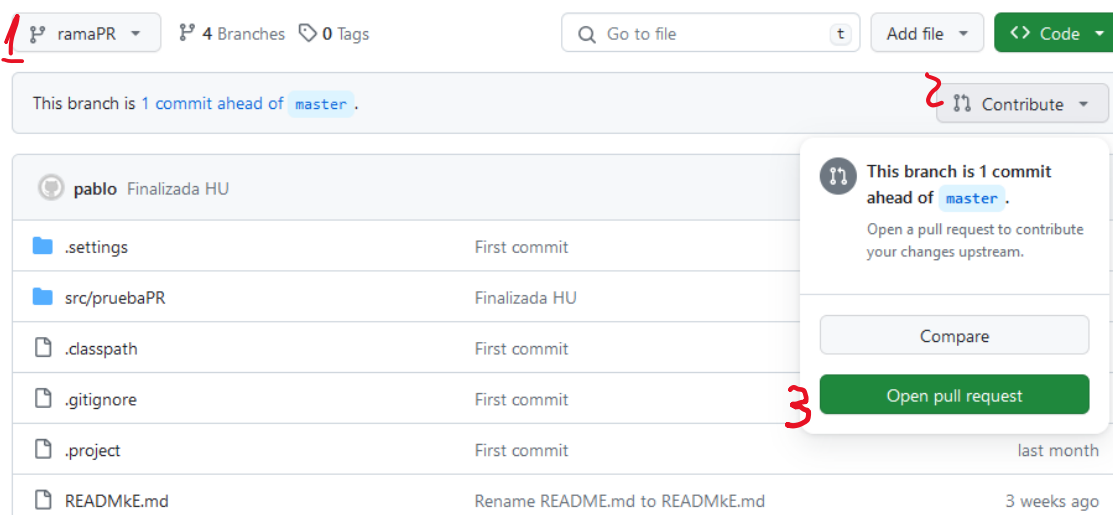
A continuación, se describe el proceso de un pull request a través de un ejemplo de uso, comenzando por la creación del mismo partiendo de un commit desde Eclipse, continuando con la revisión y mejora del código en base a esta revisión y finalizando con la aceptación de incorporación del código por parte del revisor. En el ejemplo usado en la guía el rol de desarrollador lo ejerce el usuario "pablogiis" mientras que el rol de "revisor" es ejercido por el usuario "suarezgpablo".

## 1.1 Desarrollador: Solicitud pull request

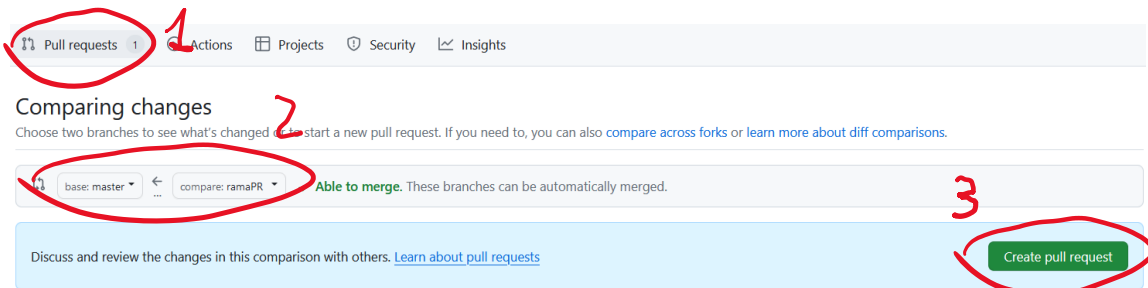
El desarrollar, tras finalizar una HU en su rama, debe realizar un push desde Eclipse de la rama en la que se está desarrollando. En esta guía, seguiremos el proceso de unir la rama "ramaPR" con la rama "master" (rama principal del proyecto creado en primera instancia desde Eclipse).



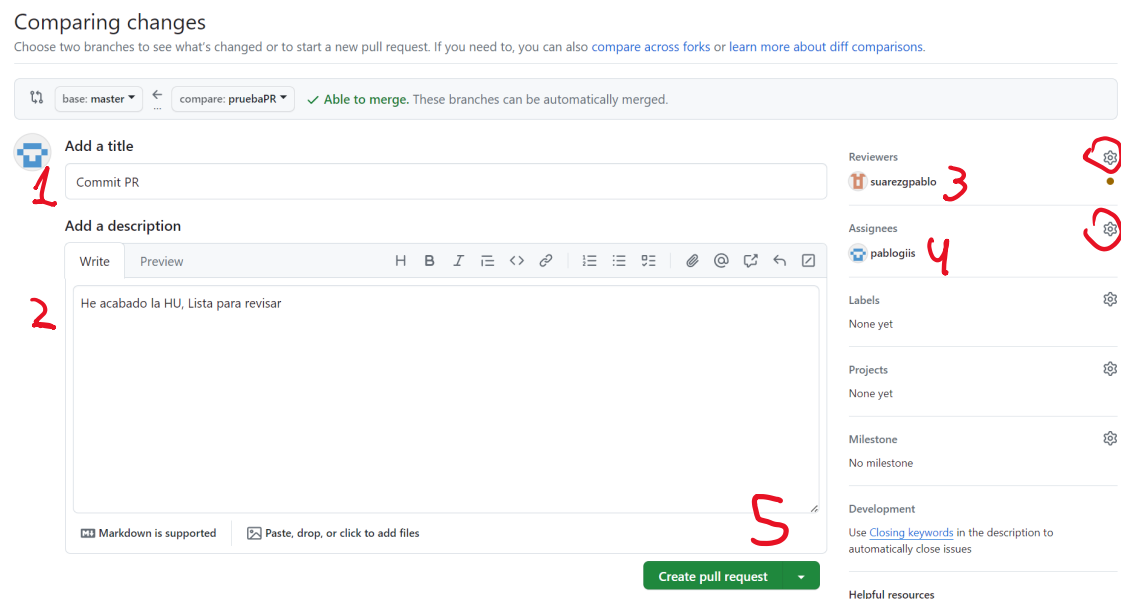
1) Tras realizar el push en Eclipse, desde Github el desarrollador deberá visualizar la rama, en este caso "ramaPR" (1), y pulsar en el botón "Contribute" (2), para posteriormente pulsar en "Open pull request" (3). Tras esto para pasar a la ventana de creación del pull request a la rama principal.



Otra opción para llegar a la ventana de creación del pull request sería la de pulsar en la opción 'Pull requests' en la parte superior del repositorio (1) y manualmente indicar qué rama se solicitar fusionar con la principal. Tras pulsar, deberá indicar que quiere realizar un 'merge' de la rama en la que se ha realizado la HU (en este caso 'ramaPR') a la rama 'master' (2) y posteriormente pulsar en el botón "Create pull request" (3).

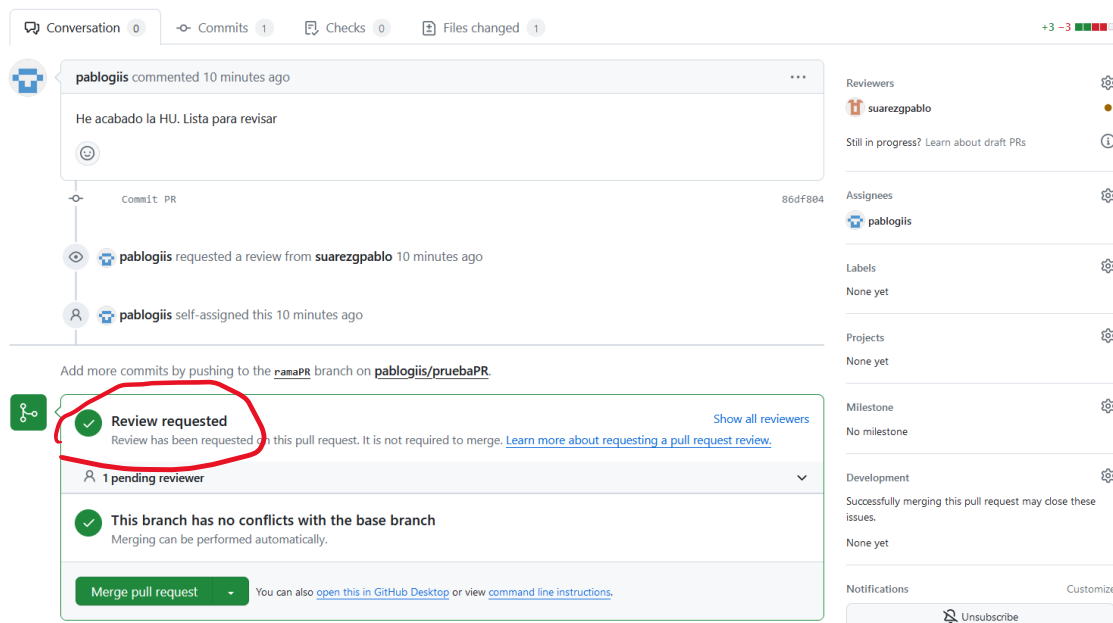


2) Ya en la ventana de creación del pull request, se debe escribir el título de la pull request (1) y la descripción de la misma (2). Esta descripción debe explicar qué se ha desarrollado y qué se quiere unir a la rama principal. A su vez, en la parte derecha de la pantalla, se debe asignar el revisor de la pull request ("Reviewer") (3) así como el desarrollador ("Assignees") (4) pulsando en el ícono de la rueda dentada. Finalmente, el desarrollador deberá pulsar en "Create pull request" (5) para crear la pull request.



## 1.2 Revisor: Revisión de la pull request

El revisor recibe tanto en su cuenta de Github como en su correo electrónico una notificación de Github sobre su asignación como revisor a la pull request recién creada. Al acceder a la ventana de la pull request creada vería la siguiente pantalla, informándole de que se le ha solicitado una revisión.



1) Para iniciar la revisión de la Pull Request, el revisor deberá en primer lugar leer su descripción. En una primera revisión solo existirá la descripción inicial introducida por el desarrollador, pero a partir de la segunda revisión podrá haber comentarios, aclaraciones y dudas por parte del desarrollador. Una vez que el revisor ha leído toda la información disponible, el revisor deberá revisar el código que se quiere unir a la rama principal. Para ello, el revisor deberá pulsar en la pestaña “Files changed” donde procederá a ver todos los cambios que se han producido entre el código de la rama principal y la rama a unir.

El revisor deberá poner todos los comentarios que considere oportunos, tanto del código eliminado como del añadido, con el objetivo de mejorar la calidad del código y/o eliminar defectos de este. Estos comentarios deben contener el suficiente detalle para que el desarrollador pueda efectuar los cambios solicitados por el revisor. Para introducir estos comentarios, el revisor deberá pulsar en el botón ‘+’ (1) que aparece al pasar el cursor por encima de una línea, mostrándose el campo de texto en el que se escribe el comentario (2). El revisor puede optar por poner un comentario (‘Add single comment’), que no exija una revisión por parte del desarrollador, o por un comentario que exija una corrección de defecto detectado en el código (‘Start a review’) (3). Normalmente, esta última opción será la más utilizada en una revisión.

Conversation 0 Commits 1 Checks 0 Files changed 1

Changes from all commits File filter Conversations Jump to

src/pruebaPR/Main.java

```
@@ -3,10 +3,10 @@
3 3 public class Main {
4 4
5 5     public static void main(String[] args) {
6 -         int numero = 9;
6 +         int numero = 16;
7 7     for (int i = 0; i<numero;i++) {
8 -         if (i==5) {
8 +         if (i==8) {
9 -             System.out.print("Iteracción 5");
9 +             System.out.print("Iteracción 8");
```

Write Preview

Existe un error en el código, ya que en realidad debería ser la iteración 9

Markdown is supported Paste, drop, or click to add files

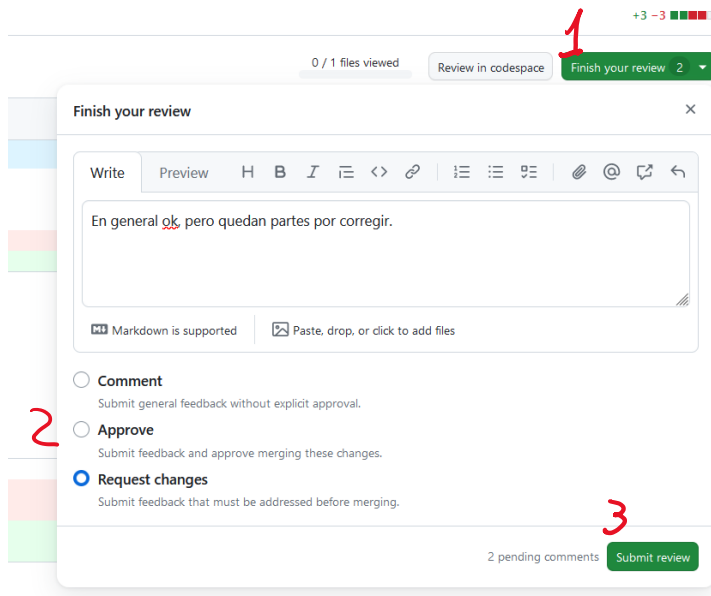
Cancel Add single comment Start a review

Tras crear el comentario, este quedará visible y asociada a la línea del código seleccionada. Este proceso deberá realizarse para todas las líneas del código para las que el revisor considere que hay que incluir comentarios.

2) Tras finalizar la revisión de todo el código, el revisor deberá pulsar en 'Finish your review' (1) para finalizar la revisión. En el panel que se despliega para tal propósito, se puede introducir un comentario opcional que resuma la revisión. **El revisor deberá también indicar su decisión, eligiendo una de las siguientes opciones (2):**

- **Comment:** Para dejar un comentario general sin aprobar el pull request ni solicitar cambios.
- **Approve:** Enviar los comentarios y aprobar el pull request. En el caso de no haber comentarios, aprobar el pull request sería opcional y se podría directamente realizar el pull request, entendiéndose aprobado de forma implícita.
- **Request changes:** Enviar los comentarios con los cambios que debe realizar el desarrollador antes de poder aceptar el pull request

Normalmente, en la asignatura, se elegirán las opciones 'Approve' o 'Request changes'. Finalmente se pulsa en "Submit review" (3) para finalizar la revisión.



Tras finalizar la revisión, se envía una notificación automática por parte de Github a la cuenta y al email del desarrollador informándole de ello.

### 1.3 Desarrollador: Corrección de defectos detectados en revisión

El desarrollador deberá entrar en la página de la pull request para visualizar los comentarios escritos por el revisor, indicándole los cambios que debe realizar para poder aceptarse la pull request.

1) El desarrollador deberá leer cuidadosamente los comentarios de revisión, pudiendo contestar a los mismos en el caso de tener dudas para que sean aclaradas por parte del revisor. Es muy importante que el desarrollador no tenga dudas sobre qué cambios debe realizar antes de comenzar a aplicar los cambios.

2) El desarrollador, desde Eclipse, deberá introducir todos los cambios solicitados en la revisión. Según se vayan solucionando los comentarios de los revisores, el desarrollador deberá pulsar en "Resolve conversation" en la página del pull request. Tras considerar que se han resuelto todos los comentarios, se deberá realizar, desde Eclipse, un commit (y su push al repositorio externo) a la rama con los cambios efectuados.

**Open** Commit PR #1  
pablogis wants to merge 1 commit into `master` from `ramaPR`

suarezgpablo left a comment

En general ok, pero quedan partes por corregir.

```
src/pruebaPR/Main.java
8 - if (i==5) {
9 -     System.out.print("Iteración 5");
8 + if (i==8) {
9 +     System.out.print("Iteración 8");
```

suarezgpablo 2 minutes ago

Este código sigue cometiendo el mismo error que antes. Al empezar en 0, sería la iteración 9.

Reply...

Resolve conversation

```
src/pruebaPR/Main.java
... @@ -3,10 +3,10 @@
3 3 public class Main {
4 4
5 5 public static void main(String[] args) {
6 - int numero = 9;
6 + int numero = 16;
```

suarezgpablo 1 minute ago

Debería ser 15

Reply...

Resolve conversation

Notifications

Unsubscribe

You're receiving notifications because you're watching this repository.

2 participants

Lock conversation

3) Tras el commit y push, desde Github, el desarrollador deberá solicitar una nueva revisión pulsando en “Re-request review”:

Add more commits by pushing to the `ramaPR` branch on `pablogis/pruebaPR`.

**Changes requested** [Hide all reviewers](#)  
1 review requesting changes [Learn more about pull request reviews.](#)

1 change requested

suarezgpablo requested changes

Require approval from specific reviewers before merging  
[Rulesets](#) ensure specific people approve pull requests before they're merged.

Continuous integration has not been set up  
[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch  
Merging can be performed automatically.

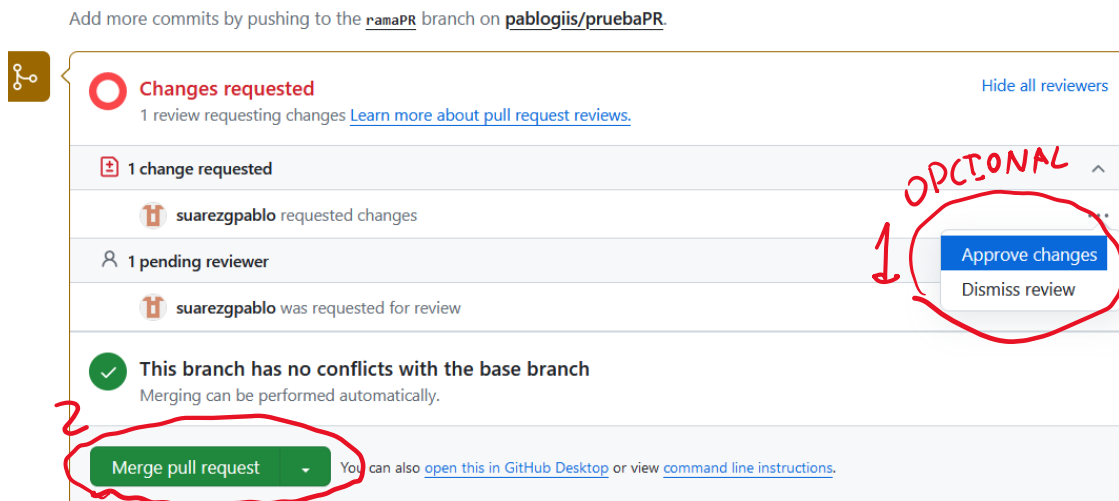
Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

See review  
Dismiss review  
**Re-request review**

Tras solicitar la revisión, el revisor volverá a recibir una notificación tanto en su cuenta de Github como por email avisándole de la petición de revisión del nuevo código. Este proceso de revisión y aplicación de cambios deberá ser repetido hasta que el revisor considere que se puede aceptar la pull request.

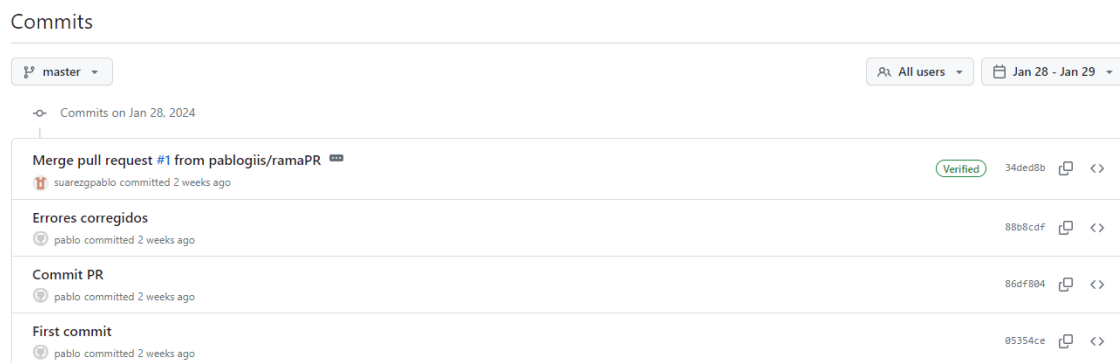
## 1.4 Revisor: Aprobación de la Pull Request

Cuando el revisor considere que se han corregido todos los defectos de forma exitosa, se considera que aprueba el pull request. Desde Github, el revisor podrá, de forma opcional, aprobar los cambios (1). A continuación, el revisor aceptará el pull request, pulsando en el botón “Merge pull request” (2), pudiendo elegir entre realizar un “merge commit”, “squash” o “rebase and merge”. Se aconseja elegir la opción “squash and merge”, combinando todos los commit que se hayan realizado durante la revisión en solo uno.



2) Opcionalmente, el revisor podrá cambiar el mensaje del commit correspondiente al pull request, mostrándose por defecto el mensaje introducido por el desarrollador en el commit que dio inicio a la pull request. Una vez confirmado el ‘merge’, el pull request se cerrará automáticamente y se podrá borrar la rama unida.

A continuación, se muestra el historial de la rama “master” en el que se puede visualizar como el último commit es el correspondiente al pull request desde la rama “ramaPR”.



## 2. Guías externas Github

En esta sección se proporciona el enlace a diferentes partes de la guía sobre pull requests proporcionada en la documentación de Github.

Acerca de las revisiones de solicitudes de extracción: <https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/reviewing-changes-in-pull-requests/about-pull-request-reviews>



Revisar los cambios propuestos en una solicitud de extracción:

<https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/reviewing-changes-in-pull-requests/reviewing-proposed-changes-in-a-pull-request>

Comentar en una solicitud de extracción: [https://docs.github.com/es/pull-](https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/reviewing-changes-in-pull-requests/commenting-on-a-pull-request)

[requests/collaborating-with-pull-requests/reviewing-changes-in-pull-requests/commenting-on-a-pull-request](https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/reviewing-changes-in-pull-requests/commenting-on-a-pull-request)

Visualización de una revisión: [https://docs.github.com/es/pull-requests/collaborating-with-](https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/reviewing-changes-in-pull-requests/viewing-a-pull-request-review)

[pull-requests/reviewing-changes-in-pull-requests/viewing-a-pull-request-review](https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/reviewing-changes-in-pull-requests/viewing-a-pull-request-review)