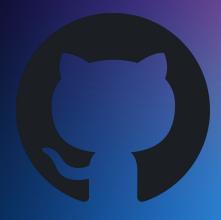
Clase N° 9: Git & Github

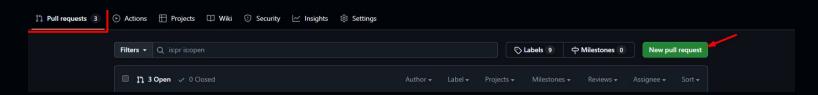






Pull request

- → La PR nos permite proponer cambios en una rama para que sean revisados e integrados en otra (main)
- → Para crear una PR:
 - Hacer cambios en una feature branch.
 - Sube los cambios al repositorio remoto con git push origin [branch-name].
 - En GitHub, ve a la pestaña Pull Requests y haz clic en New Pull Request.
 - Selecciona las ramas involucradas y proporciona un título/descripción clara de los cambios.





Code Review

- → Proceso en el que tu equipo revisa el código propuesto en un PR.
- → Objetivo: Detectar errores, mejorar la calidad del código, y garantizar que sigue los estándares del proyecto.
- → Buenas prácticas para revisar código:
 - Comentar de manera constructiva y educada.
 - Validar el impacto de los cambios en otras partes del sistema.
 - Usar herramientas como GitHub Reviews para aprobar, comentar o pedir cambios.



Tipos de merge

	Squash		Rebase	
Ventajas		ntiene el historial pio y organizado.	→	Facilita entender el historial del proyecto.
Usos ideales:		ta commits ermedios irrelevantes.	→	ldeal para sincronizar tu rama con main antes de abrir un PR.



Squash vs Rebase

```
# Antes:
main ---o---o
\
feature/login ---a---b---c

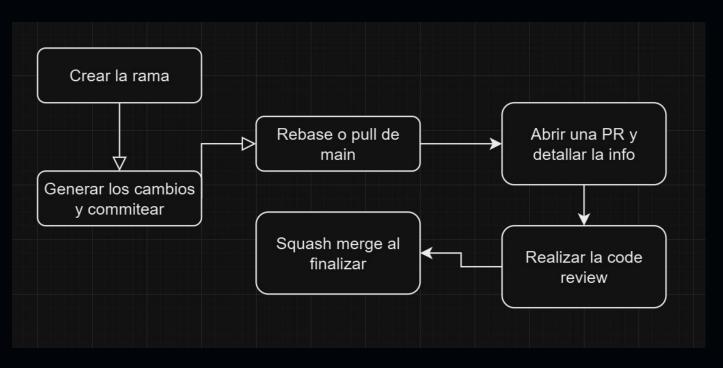
# Después del Squash:
main ---o---o---d [1 commit squash]
```

```
#Antes
main ---o---o---o
\ \ \
feature ---a---b---c

#Despues del rebase
main ---o---o---a---b---c
```



Flujo ideal





Práctica

- → Seleccionar una feature del listado enviado por Slack
- → Hacer git pull, para actualizar la rama local
- → Crear una rama que empiece con feature/*
- → Comenzar el desarrollo (1 hora)
- → Pushear la rama al repositorio
- Crear una pull request que contenga
 - Descripción de la feature
 - Explicar técnicamente qué hiciste (para el resto de los devs)
 - Agregar imágenes
 - Asignar 3 revisores de código
- → Realizar code review de las otras PRs
 - Sugerir cambios
 - Proponer mejoras
 - Cuestionar cómo resolvió el problema
 - Aprobar/Rechazar la PR

