



GITHUB

UNAI PINILLA Y JUAN CEBELLÁN

¿QUE ES?

GitHub es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador

¿Qué ventajas da GitHub?

- Guardar versiones del código.
- Colaborar con múltiples personas al mismo tiempo.
- Revisar cambios realizados por otros miembros del equipo.
- Integrar herramientas de automatización, documentación y pruebas.



GIT

Git es un sistema de control de versiones de código abierto muy popular y eficiente. Rastrea contenido como archivos y directorios. Permite registrar los cambios realizados en archivos a lo largo del tiempo, revertir a versiones anteriores y trabajar de forma paralela en ramas.

Características de Git:

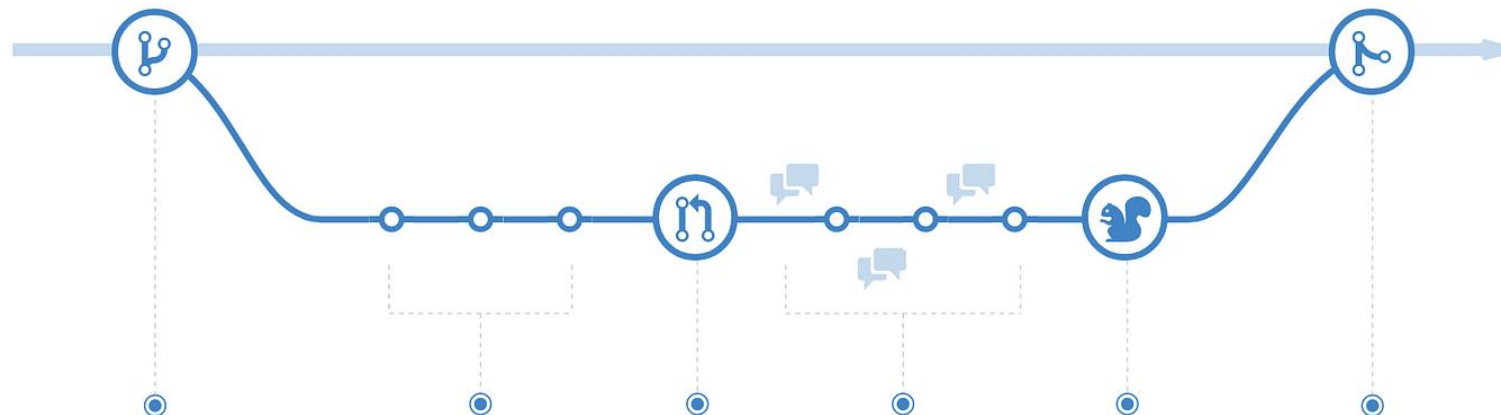
- Trabajo con ramas
- Combinar cambios con merge.
- Historial de cambios accesible y detallado
- Resolución de conflictos cuando se modifican las mismas líneas de código.



¿COMO FUNCIONA?

GitHub te permite trabajar en un repositorio privado y luego subirlo a la nube mediante:

- Fork: Copia de un repositorio en tu propia cuenta. Permite trabajar libremente sin afectar el proyecto original.
- Pull Request (PR): Propuesta de cambios al repositorio original. Los colaboradores pueden revisar, comentar y aprobar los cambios antes de fusionarlos.



FLUJO BÁSICO

■ Comandos:

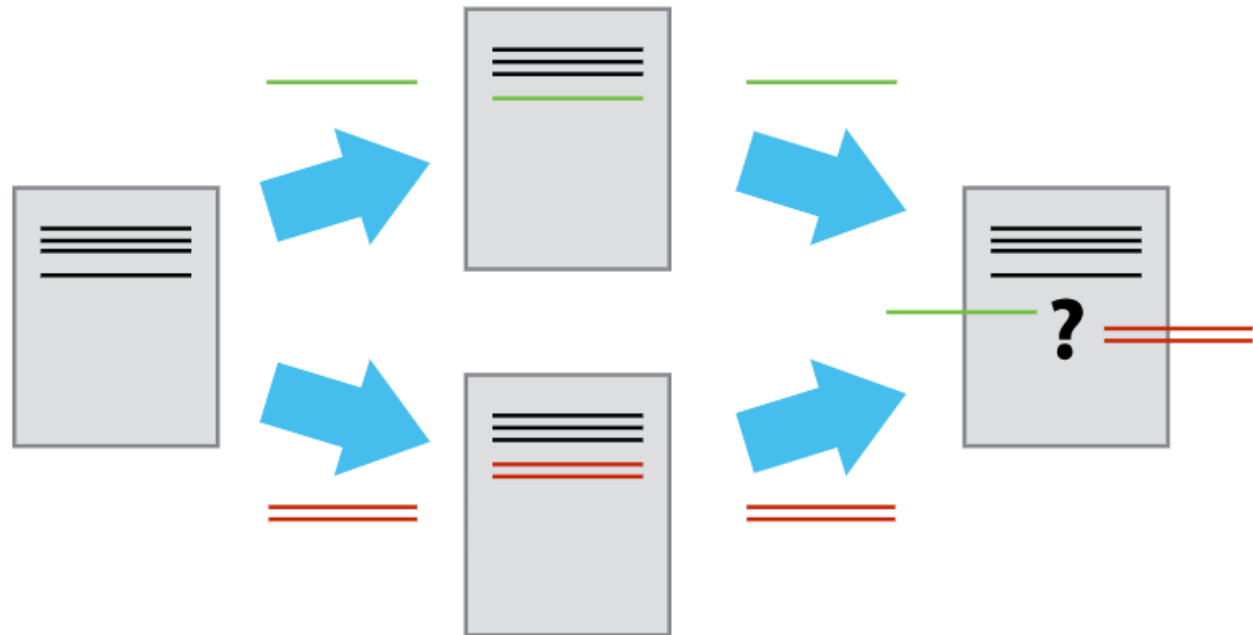
- git clone: El comando git clone se utiliza para clonar un repositorio remoto y trabajar en él de manera local.
- git Branch: sirve para gestionar las diferentes ramas
- git add: El comando git add se usa para preparar los archivos modificados o nuevos antes de confirmarlos con el commit.
- git push: El comando git push envía los commits al repositorio remoto en GitHub, dentro de la rama en la que estás trabajando.
- git pull: El comando git pull se utiliza para descargar los últimos cambios del repositorio remoto y fusionarlos con tu rama local.
- Pull Request: Esto es una solicitud para que tus cambios sean revisados y añadidos a la rama principal (main).

RESOLUCIÓN CONFLICTOS

En el caso de que varias personas trabajen en un mismo archivo, es normal que surjan conflictos, estos se deben a que Git no sabe cual de los dos criterios prevalece

Los conflictos más comunes aparecen al:

- Realizar un *merge* entre ramas con cambios en las mismas líneas.
- Hacer un *rebase* que reescribe el historial sobre archivos modificados.
- Aplicar *cherry-pick* o recuperar cambios con *stash*.



EJEMPLO DE RESOLUCIÓN DE CONFLICTO

```
```bash
```

```
git checkout -b rama-nueva
```

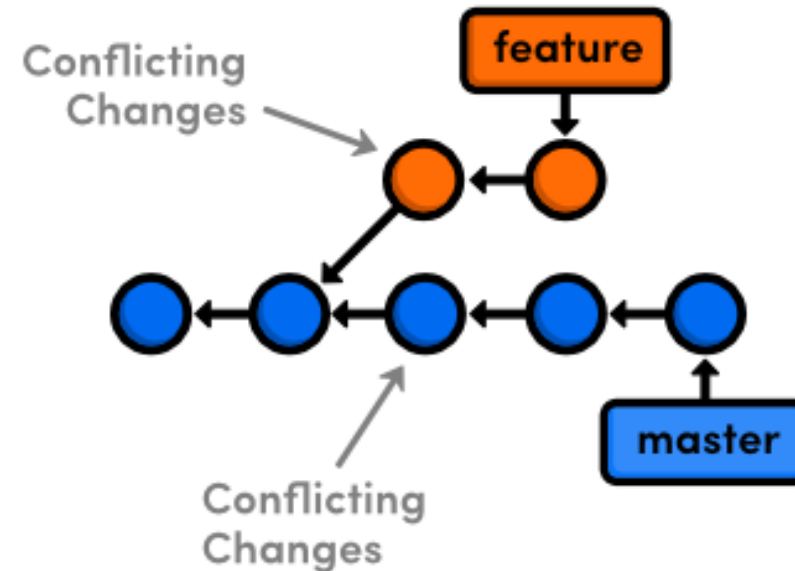
```
echo "Hola desde rama-nueva" >> archivo.txt
git add archivo.txt
git commit -m "Cambios en rama-nueva"
```

```
git checkout main
echo "Hola desde main" >> archivo.txt
git add archivo.txt
git commit -m "Cambios en main"
```

```
git merge rama-nueva
```

```
git add archivo.txt
git commit -m "Conflicto resuelto"
```

```
```
```



ISSUES

- Los Issues son tareas, ideas, errores o mejoras que se registran en un repositorio. Se utilizan para mantener organizado el trabajo dentro de un proyecto colaborativo
- Cada issue se puede:
 - Titular con una tarea clara.
 - Describir con más detalle lo que se necesita hacer.
 - Comentar y discutir con otros colaboradores.
 - Cerrar automáticamente con commits.



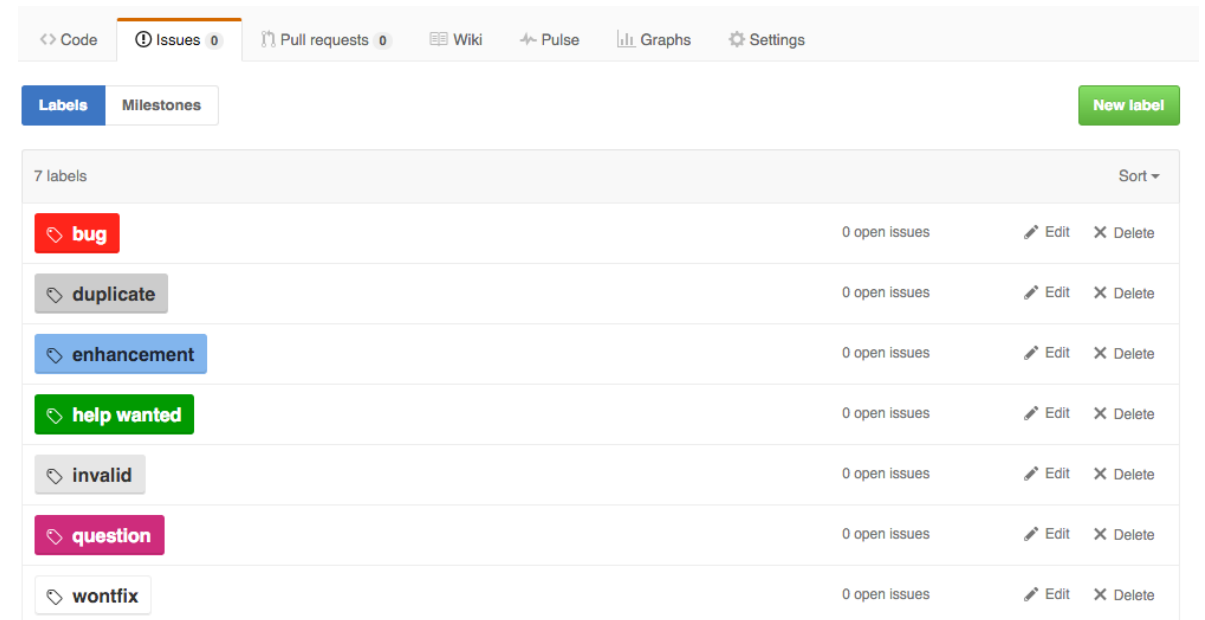
Issues

ISSUES

- Las etiquetas o labels ayudan a categorizar los Issues.

Algunas etiquetas comunes son:

- *bug*: indica un error.
- *enhancement*: indica una mejora o nueva funcionalidad.
- *question*: para dudas o aclaraciones.
- *documentación, urgente, fácil...* (se pueden personalizar).















The screenshot shows the GitHub interface for a repository. At the top, there are tabs for Code, Issues (0), Pull requests (0), Wiki, Pulse, Graphs, and Settings. Below these, there are tabs for Labels and Milestones, and a 'New label' button. The 'Labels' tab is active, showing a list of 7 labels. Each label has a name, a count of open issues, and options to edit or delete it.

| Label | Open Issues | Actions |
|-------------|---------------|-------------|
| bug | 0 open issues | Edit Delete |
| duplicate | 0 open issues | Edit Delete |
| enhancement | 0 open issues | Edit Delete |
| help wanted | 0 open issues | Edit Delete |
| invalid | 0 open issues | Edit Delete |
| question | 0 open issues | Edit Delete |
| wontfix | 0 open issues | Edit Delete |

PROJECTS

Los Projects permiten organizar los Issues en forma de un tablero Kanban.

Sirve para visualizar el estado del proyecto y ver quién está haciendo cada cosa.

| Title | ... | Assignees | ... | Status | ... | Fecha estimada | ... | + |
|-------|-----|---|---|--------|-------------|----------------|--------------|---|
| 1 | ✓ | Redactar introducción general a GitHub #14 |  juancebellan | ▼ | Done | ▼ | Jun 13, 2025 | |
| 2 | ✓ | Redactar el archivo flujo-basico.md #15 |  unaipini | ▼ | Done | ▼ | Jun 13, 2025 | |
| 3 | ✓ | Redactar guía sobre resolución de conflictos #16 |  juancebellan | ▼ | Done | ▼ | Jun 15, 2025 | |
| 4 | ✓ | Simular un conflicto y resolverlo (práctica) #17 |  unaipini | ▼ | Done | ▼ | Jun 15, 2025 | |
| 5 | ✓ | Explicar uso de Issues, Proyectos y Milestones #18 |  unaipini | ▼ | Done | ▼ | Jun 16, 2025 | |
| 6 | ✓ | Explicar convenciones de commits y ramas #19 |  juancebellan | ▼ | Done | ▼ | Jun 15, 2025 | |
| 7 | 🟢 | Revisión final y unión de todas las secciones #20 |  unaipini | ▼ | In Progress | ▼ | Jun 15, 2025 | |
| 8 | 🟢 | Crear presentación en PowerPoint #21 |  juancebellan | ▼ | In Progress | ▼ | Jun 16, 2025 | |
| 9 | 🟢 | Grabar el vídeo explicativo #22 |  juancebellan and u... | ▼ | Todo | ▼ | Jun 17, 2025 | |
| 10 | 🟢 | Subir capturas del tablero Kanban en 3 momentos distintos #23 |  unaipini | ▼ | Todo | ▼ | Jun 17, 2025 | |
| 11 | 🟢 | Preparar entrega final (ZIP) #24 |  juancebellan | ▼ | Todo | ▼ | Jun 18, 2025 | |
| 12 | 🟢 | Revisión final de la entrega y segundo OK #25 |  unaipini | ▼ | Todo | ▼ | Jun 18, 2025 | |

+

You can use

Control + Space

 to add an item

MILESTONES

Un Milestone es una meta concreta que agrupa varios Issues relacionados. Se puede usar, por ejemplo, para agrupar todas las tareas necesarias para una entrega del proyecto o una versión.

Permiten llevar control de cuánto se ha completado y qué falta para alcanzar un objetivo mayor.



CONVENCIONES DE COMMITS Y NOMBRES DE RAMAS

Es importante nombrar a las cosas correctamente y explicar bien todas las cosas que vas a añadir, ya que de esta manera se puede saber bien qué cosas se han modificado, añadido y/o quitado

Algunos mensajes genéricos que se pueden poner son los siguientes:

- *feat*: Añadir una nueva funcionalidad
- *fix*: corregir un error
- *docs*: Cambios en la documentación
- *style*: Cambios de estilo
- *refactor*: Reestructuración del código sin cambiar su comportamiento
- *test*: Añadir o modificar tests
- *chore*: Tareas menores que no afectan directamente el código (actualizar dependencias, configuración, etc.)

CONVENCIONES DE COMMITS Y NOMBRES DE RAMAS

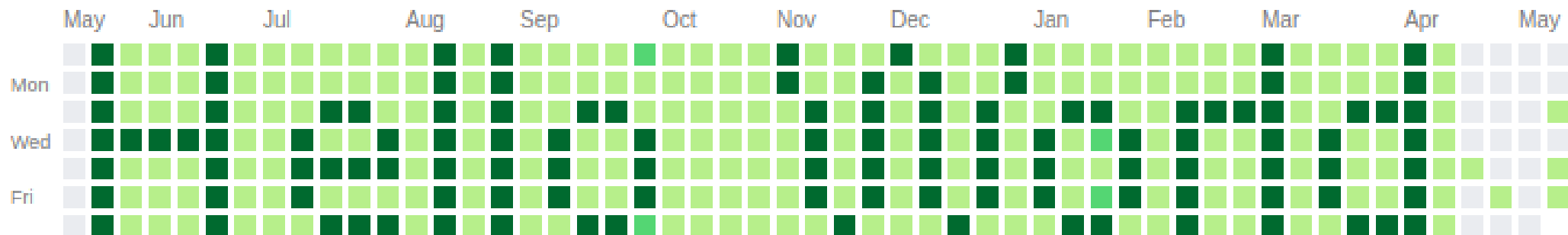
Recomendaciones Generales:

Escribir los mensajes de commit en modo infinitivo

Limitar la descripción a máximo 72 caracteres

Utilizar el cuerpo del commit (si es necesario) para explicar el "por qué" del cambio.

Evitar mensajes genéricos como *update*, *cambiar*, *eliminar*, *agregar*, *cosas*.

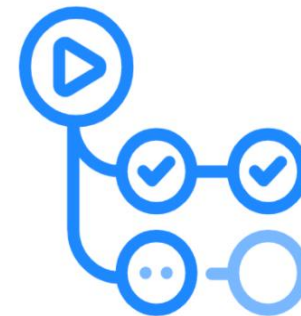


[Learn how we count contributions.](#)

Less More

GIT HUB ACTIONS

- GitHub Actions es una funcionalidad que permite automatizar tareas dentro de un repositorio de GitHub. Esto significa que puedes configurar procesos automáticos que se ejecuten en respuesta a eventos, como cuando alguien hace un commit o crea un pull request.



GitHub Actions

CI/CD

CI/CD es una práctica de desarrollo que automatiza la integración y entrega del código:

- **CI (Integración Continua):** verifica automáticamente los cambios mediante pruebas cada vez que se sube código.
- **CD (Entrega/Despliegue Continuo)** automatiza el proceso de llevar ese código a producción o entornos de prueba de forma segura y rápida.

