

# Introducción a github

---

## Referencias

- [Ayuda de GitHub] (<https://help.github.com/>)
- [Sintaxis de Markdown](#)
- [Autenticación en GitHub mediante tokens](#)
- [Solicitud de descuentos para uso educativo](#)

## Funcionalidades

GitHub ofrece funcionalidades muy interesantes para trabajar con prácticas relacionadas con código o archivos de texto. Tenemos las siguientes ventajas:

- Posibilidad de ver el código en el explorador web sin necesidad de descargarlo
- Podemos crear conversaciones para realizar comentarios sobre la tarea
- Posibilidad de hacer comentarios directamente sobre el código, haciendo referencia a líneas concretas, otros comentarios, commits, archivos, etc.
- Colaboración entre distintas personas para tareas en grupo. Visibilización de las aportaciones de cada persona al trabajo común.

## Autenticación mediante tokens

GitHub ofrece la posibilidad de utilizar tokens en lugar de la contraseña de la cuenta para interactuar con su servicio a través de línea de comandos, gestor gráfico o API. Los tokens ofrecen una alternativa a la contraseña de la cuenta y proporcionan las siguientes ventajas:

Permiten especificar permisos de acceso en lugar de proporcionar acceso completo

Pueden generarse tantos como se necesiten para proporcionar distintos perfiles de acceso

En este [enlace](#) tienes toda la información para crear un token en tu cuenta. Una vez creado, solo se podrá visualizar una vez, por lo que deberá copiarse para poder ser utilizado. Recuerda que el token puede utilizarse para sustituir a la contraseña en el acceso a GitHub desde línea de comando o mediante un gestor gráfico

## Colaboración

GitHub ofrece dos mecanismos básicos para la colaboración en un determinado repositorio:

- **Forks y Pull Requests** -Mecanismo utilizado por defecto. Permite que personas que no tienen acceso de escritura al repositorio puedan hacer una copia del mismo en su propia cuenta y enviar los cambios para que la persona dueña del repositorio original decida si quiere o no integrarlos. Muy útil en proyectos de Código Abierto, donde las personas colaboradoras no se conocen entre sí.
- **Permisos de colaboradores** Se pueden agregar **colaboradores** a un repositorio para que puedan realizar cambios. Este método es útil si tenemos claro que determinadas personas van a colaborar en el repositorio. Este método de trabajo puede extenderse mediante el uso de organizaciones, que permiten crear equipos de personas y asignarles permisos para cada uno de los repositorios de la organización.

## Issues y milestones

Los Issues (propuestas) permiten crear notas para, por ejemplo:

- Identificar fallos en el código y hacer su seguimiento
- Proponer mejoras o nuevas características
- Hacer un seguimiento de las tareas que hay que realizar en el repositorio

En el texto de los Issues se pueden hacer menciones a otros usuarios de GitHub, hacer referencia a líneas de código o crear listas de tareas que pueden marcarse como completadas. También pueden adjuntarse fragmentos de código.

Los Issues pueden asignarse a personas responsables u organizarse en hitos (milestones) para agruparlos y hacer un seguimiento de los mismos.

Por último, es posible cerrar los issues a través de palabras clave en el texto de los commits. De esta manera se asociará el commit con el issue resuelto o cerrado.