

# Git and GitHub for Dummies

GitHub

# Sobre GitHub

- GitHub es una plataforma de desarrollo colaborativo (forja) para alojar proyectos utilizando el sistema de control de versiones git.
- GitHub se ha convertido en la plataforma de desarrollo colaborativo más popular en todo el mundo.
- Lanzamiento: 2007
- Alternativas: GitLab, Bitbucket, Azure Repos, etc.

# Por qué GitHub

- GitHub funciona como una comunidad de programadores que te permite interaccionar muy fácilmente con otros programadores.
- GitHub es una red social.
- La mayoría de los programadores de software libre usan GitHub.
- GitHub, gracias a sus métricas, características de gamificación, desarrollo continuo, conexión a plataformas de programación diversas y cliente móvil, es hoy en día la mejor opción para desarrollar software libre.

# Repositorios públicos y privados

Por omisión, los repositorios de GitHub son públicos. Para conseguir un repositorio privado hay que hacer una de estas dos cosas:

- Pagar para obtener una cuenta **Pro**. Por 7 dólares al mes puedes tener cinco repositorios privados.
  - <https://github.com/pricing>
- Conseguir una **cuenta para educación** que permite, generalmente de forma limitada, tener un número determinado de cuentas gratuitas.
  - <https://education.github.com/>

# Red social

Cada usuario tiene un perfil y en él puede contar cosas como dónde se encuentra, su web y poco más.

Se puede seguir la actividad de un usuario, pero también se puede ir un paso más allá y pulsar el botón de *Follow* con lo que, al entrar en la página principal de GitHub se te mostrará, junto con la actividad propia, la de esta persona.

Una persona puede ser también añadida a un repositorio, lo que le dará privilegios para realizar todo tipo de acciones sobre él. Conviene usar esto con moderación y sólo cuando se trate de una persona ya involucrada en el proyecto.

# Red social

El componente social también se ve en repositorios específicos: los repositorios se pueden vigilar (*Watch*) y también poner una especie de "Me gusta", *Star*.

# Red social

GitHub también permite comentar a diferentes niveles: se puede comentar un *commit*, se pueden comentar líneas de código y finalmente se pueden hacer solicitudes y comentarios a un repositorio completo usando *issues* o solicitudes.

Estas formas de interacción permiten aumentar el *karma* a base de las conexiones, estrellas que reciban sus repositorios y comentarios, así como los *issues* resueltos.

# Red social

Los *issues* se agrupan en *milestones* o hitos, que son simplemente grupos de temas que deben ser resueltos antes de pasar a otra fase del desarrollo.

Agrupar los *issues* en hitos permite ver el progreso del mismo, ya que te va mostrando cuál es el grado de terminación de dicho hito. Los hitos, además, pueden fecharse con lo que se puede ver si se ha pasado uno de fecha o no.



# Red social

Los usuarios se pueden agrupar en organizaciones.

Una organización es en muchos aspectos similar a un usuario; tiene las mismas limitaciones y las mismas ventajas, pero en una organización se definen equipos y los permisos para trabajar por repositorios se hacen usando estos equipos; cada repositorio puede tener uno o más equipos con diferentes niveles de privilegios y el repositorio en sí tendrá también un equipo que será el que pueda realizar ciertas acciones sobre el mismo.

Generalmente se crea un equipo por repositorio, pero puede organizarse de cualquier otra forma.

# GitHub pages

Una de las partes interesantes de GitHub es la posibilidad de publicar páginas estáticas relacionadas con el proyecto o, para el caso, sobre lo que uno quiera.



# Hooks

Los *hooks*, o ganchos, son eventos que se activan cuando se produce algún tipo de acción por parte de git.

En general, se usan para integrar el sistema de gestión de fuentes de git con otra serie de sistemas, principalmente de integración continua o entrega continua o, en general, cualquier tipo de sistema de notificaciones o de trabajo en grupo.

# Pull Request

Los *Pull Requests* permiten informar sobre cambios que se han realizado sobre una rama del proyecto.

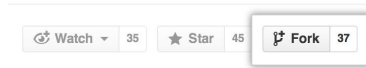
Una vez es abierto, se puede discutir y revisar los cambios potenciales con los colaboradores, pudiendo añadir comentarios antes de que los cambios sean integrados en la rama deseada.

Tutorial:

<https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/creating-a-pull-request>

# Forking projects

¿Qué es el *forking*?



Si se desea contribuir en un proyecto público o si se desea hacer una copia de un proyecto público para empezar un proyecto propio a partir de ese proyecto original, se denomina *forking* a dicho proceso.

Crear un *fork* es producir una copia personal del proyecto de otra persona. Los *forks* actúan como un puente entre el proyecto original y la copia personal. Se pueden generar *Pull Requests* para ofrecer tus cambios al proyecto original.

*Forking* es el “core” de la red social GitHub.

# Forking projects

Ejercicio: <https://guides.github.com/activities/forking/>

# Referencias

[1] GitHub - <https://es.wikipedia.org/wiki/GitHub>

[2] Usando git como los profesionales: GitHub -  
<https://github.com/JJ/aprende-git/blob/master/texto/github.md>

[3] GitHub - <https://github.com>