# ¿Quién inventó el sistema de control de versión Git y por qué?

## Una breve historia de Git

El kernel de Linux es un proyecto de software de código abierto con un alcance bastante amplio. Durante la mayor parte del mantenimiento del kernel de Linux (1991-2002), los cambios en el software se realizaban a través de parches y archivos. En el 2002, el proyecto del kernel de Linux empezó a usar un DVCS propietario llamado BitKeeper.

En el 2005, la relación entre la comunidad que desarrollaba el kernel de Linux y la compañía que desarrollaba BitKeeper se vino abajo y la herramienta dejó de ser ofrecida de manera gratuita. Esto impulsó a la comunidad de desarrollo de Linux (y en particular a Linus Torvalds, el creador de Linux) a desarrollar su propia herramienta basada en algunas de las lecciones que aprendieron mientras usaban BitKeeper. Algunos de los objetivos del nuevo sistema fueron los siguientes:

* Velocidad
* Diseño sencillo
* Gran soporte para desarrollo no lineal (miles de ramas paralelas)
* Completamente distribuido
* Capaz de manejar grandes proyectos (como el kernel de Linux) eficientemente (velocidad y tamaño de los datos)

Desde su nacimiento en el 2005, Git ha evolucionado y madurado para ser fácil de usar y conservar sus características iniciales. Es tremendamente rápido, muy eficiente con grandes proyectos y tiene un increíble sistema de ramificación (branching) para desarrollo no lineal.

# ¿A quién pertenece actualmente Github y por qué?

En 2018 (4 de junio de 2018 - 26 de octubre 2018), Microsoft anunció y compró respectivamente la compañía por un importe de más de 7000 millones de dólares.

El amor —interesado— de Microsoft por el Open Source parece no conocer límites. Mucho ha llovido desde que Linux fuera el "cáncer" (Ballmer *dixit*) de nuestra civilización tecnológica, y con Satya Nadella a los mandos ahora Microsoft no hace más que **apostar más y más fuerte por el Open Source**.

# ¿Hay otra forma que no sea la terminal para trabajar con Github?

### Si, hay muchos entornos graficos que se pueden utilizar.

GitHub Desktop, SourceTree, SmartGit, etc.