

1º ACTIVIDAD DE LA MOCHILA

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

R= Git es un proyecto de código abierto maduro y con un mantenimiento activo que desarrolló originalmente **Linus Torvalds**, el famoso creador del kernel del sistema operativo Linux, en 2005.

Git es un *software* de **control de versiones** = Se llama **control de versiones** a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación.

Diseñado por **Linus Torvalds**, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de **código fuente**= El **código fuente** de un **programa informático** (o **software**) es un conjunto de **líneas de texto** con los pasos que debe seguir la **computadora** para ejecutar un cargador.¹

El código fuente de un programa está escrito por un **programador** en algún **lenguaje de programación legible por humanos**, normalmente en forma de **texto plano**.² Sin embargo, el programa escrito en un lenguaje legible por humanos no es directamente ejecutable por la computadora en su primer estado, sino que debe ser traducido a otro lenguaje o **código binario**; así será más fácil para la máquina interpretarlo (**lenguaje máquina** o **código objeto** que sí pueda ser ejecutado por el **hardware** de la computadora). Para esta traducción se usan los llamados **compiladores**, **ensambladores**, **intérpretes** y otros sistemas de traducción.

Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o **front end** como **Cogito** o **StGIT**.² Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena.³ Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de **programación** del **núcleo Linux**.

El **mantenimiento del software** Git está actualmente (2009) supervisado por Junio Hamano, quien recibe contribuciones al código de alrededor de 280 programadores. En cuanto a derechos de autor Git es un **software libre** distribuible bajo los términos de la versión 2 de la **Licencia Pública General de GNU**.

¿A quién pertenece actualmente Github y por qué?

R= 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.

El software que opera GitHub fue escrito en Ruby on Rails.

Desde enero del 2010 opera el nombre de GitHub inc.

[Microsoft compra GitHub por 7.500 millones de dólares.](#)

Estamos hablando de una red social de facto, con más de 28 millones de usuarios y más de 85 millones de repositorios de código. En GitHub puedes seguir a otros desarrolladores, repositorios, colaborar en repositorios de terceros, discutir sobre bugs o proponer mejoras o nuevas características. Es realmente esta comunidad la que hace que no sea una forja más, y que migrar un repositorio (por ejemplo a [GitLab](#), que ha visto un considerable incremento de repositorios en estas últimas horas) sea más costoso de lo que parece a primera vista.

Pero GitHub no es sólo para simples programadores y [científicos](#): tal es el atractivo de su comunidad que, de esos 28 millones de perfiles, 1,8 millones corresponden a compañías, entre las que podemos destacar a la propia [Microsoft](#), [Apple](#), [Facebook](#) y [Amazon](#), que mantienen allí sus proyectos con licencias libres o, al menos, open source.

Desde el nombramiento de **Satya Nadella** como CEO de Microsoft (sustituyendo al infame Steve Ballmer), la compañía ha adoptado una estrategia nueva, mucho más abierta al mundo del software libre y a la integración con otras plataformas y servicios. Un ejemplo es la adquisición en 2016 de [Xamarin](#), una plataforma que permite crear aplicaciones móviles que pueden funcionar en distintos sistemas, como iOS, Android y (por supuesto) Windows. Y es, precisamente, el que fuera CEO de Xamarin, **Nat Friedman**, quien será nuevo CEO de GitHub. Así pues, esta

adquisición es, sin duda, un acercamiento a la comunidad más vibrante en el desarrollo de software libre y abierto.

Además, hay otro motivo de peso: **Microsoft Azure**. En plena explosión de *cloud services* (servicios en la nube), con un **crecimiento previsto para este año del 21.4%**, Microsoft tiene que competir con el líder indiscutible a día de hoy, la todopoderosa **Amazon con AWS**. Qué mejor forma de hacerlo que estar en contacto directo con los desarrolladores que trabajan en esos servicios, ver sus necesidades y ofrecerles herramientas que faciliten su integración con Azure.

La declaración de intenciones de Nat Friedman no puede ser más clara: **“GitHub seguirá siendo independiente, como comunidad, como plataforma y como negocio.”**

Hola, GitHub

El Octocat está fuera de la bolsa: GitHub se unirá a Microsoft y me convertiré en el nuevo CEO cuando se cierre el trato a finales de este año. Estoy seguro de que este cambio será excelente para GitHub y los desarrolladores de todo el mundo, y quería aprovechar la oportunidad para presentarme.

¿Quién soy? Mi nombre es Nat y he sido desarrollador desde que tenía seis años. He estado activo en código abierto desde los años 90 cuando descubrí Linux.

Me quedé impresionado al descubrir un sistema operativo gratuito que venía con un editor, un compilador y un depurador: ¡todas las herramientas que necesita para ser un desarrollador y todo el código fuente! Fue una oportunidad increíble para que un adolescente de un pequeño pueblo de Virginia aprendiera de los mejores desarrolladores del mundo. Pasé 15 años trabajando con Linux y fundé un par de empresas de código abierto. Mi primer compromiso con GitHub fue en 2009, cuando GitHub tenía un año.

Habrán cambios, sin duda. ¿Qué pasará con **Atom**, el editor creado por GitHub? ¿Será reemplazado por **Visual Studio Code**? ¿Reescribirán la plataforma? GitHub funciona con **Ruby on Rails**, un framework que no tiene precisamente la mejor integración con Windows. ¿Ofrecerán repositorios privados de forma gratuita y no solo de pago como hasta ahora? ¿Cómo incentivarán a los desarrolladores para que utilicen Azure? Será interesante ver qué sucede, pero no sólo en GitHub: tendremos que estar atentos a qué hacen el resto de jugadores.

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

R= SI EN VISIUAL STUDIO CODE Y LA MISA GITHUB