

Práctica individual obligatoria

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

Sr. Torvalds en 2005 anunció un nuevo protocolo en el Control de Versiones. Torvalds era experto en el desarrollo del kernel de Linux y conocía al dedillo todos los problemas que implicaba la administración y manejo del Control de Versiones: todo un trabajo equivalente a programar el mismo kernel.

En ese entonces él venía trabajando en una empresa llamada BitMover que utilizaba "BitKeeper" como software de versionado (VCS) que en ese momento era pago (recién en 2016 se ofreció como software libre).

En el año 2000 BitMover solía proporcionar acceso al sistema para ciertos proyectos de código abierto o software libre, uno de los cuales era el código fuente del kernel de Linux. **La licencia para la versión "comunitaria" de BitKeeper había permitido a los desarrolladores usar la herramienta sin costo para proyectos de código abierto o software libre, siempre que esos desarrolladores no participaran en el desarrollo de una herramienta de la competencia (como Concurrent Versions System, GNU arch, Subversion o ClearCase) durante la duración de su uso de BitKeeper más un año. Esta restricción se aplicó independientemente de si la herramienta de la competencia era gratuita o propietaria.** Esta versión de BitKeeper también requería que cierta metainformación sobre los cambios se almacenara en servidores informáticos operados por BitMover, una adición que hacía imposible que los usuarios de la versión comunitaria ejecutaran proyectos que BitMover desconocía.

En abril de 2005, BitMover anunció que dejaría de proporcionar una versión de BitKeeper sin cargo a la comunidad, dando como razón los esfuerzos de Andrew Tridgell, un desarrollador empleado por OSDL en un proyecto no relacionado, para desarrollar un cliente que mostraría el metadatos (datos sobre revisiones, posiblemente incluyendo diferencias entre versiones) en lugar de solo la versión más reciente. Poder ver metadatos y comparar versiones pasadas es una de las características principales de todos los sistemas de control de versiones, pero no estaba disponible para nadie sin una licencia comercial de BitKeeper, lo que incomoda significativamente a la mayoría de los desarrolladores del kernel de Linux. Aunque BitMover decidió proporcionar licencias comerciales gratuitas de BitKeeper a algunos desarrolladores de kernel, se negó a otorgar o vender licencias a cualquier persona empleada por OSDL, incluidos Linus Torvalds y Andrew Morton, colocando a los desarrolladores de OSDL en la misma posición que otros desarrolladores de kernel. El proyecto Git se lanzó con la intención de convertirse en el software de gestión de código fuente del kernel de Linux y, finalmente, fue adoptado por los desarrolladores de Linux.

El fin de la compatibilidad con la versión de "uso gratuito" de BitKeeper fue oficialmente el 1 de julio de 2005, y para entonces los usuarios debían cambiar a la versión comercial o cambiar el sistema de control de versiones. Los usuarios comerciales también debían no producir ninguna herramienta de la competencia.

Durante el lanzamiento de la versión 7.2ce el 9 de mayo de 2016, BitKeeper anunció que está comenzando a pasar de una licencia propietaria a una de código abierto.

Fuentes: <https://pandorafms.com/blog/es/control-de-versiones/>
<https://en.wikipedia.org/wiki/BitKeeper>

A quién pertenece actualmente GitHub y por qué?

El 4 de junio de 2018 Microsoft compró GitHub por la cantidad de 7500 millones de dólares.

Github fue desarrollado por Chris Wanstrath, P. J. Hyett, Tom Preston-Werner y Scott Chacon usando Ruby on Rails, y empezó en 2008. Aunque la compañía, Github, Inc, existía desde 2007. Crecieron de manera exponencial hasta que el 28 de febrero de 2018, GitHub fue víctima del segundo mayor ataque de denegación de servicio (DDoS) de la historia. Llegando a recibir un tráfico de 1.35 terabits por segundo.

Los ataques distribuidos de denegación de servicio (DDoS) sobrecargan los recursos computacionales del sistema atacado hasta dejarlo inaccesible. Saturación mediante grandes flujos de información desde varios puntos hasta dejar los servidores fuera de servicio.

Explicación de cómo lo solucionaron: <https://geekytheory.com/la-original-forma-en-que-github-responde-a-un-ataque-ddos>

¿Por qué Microsoft?

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.

Fuente: <https://naukas.com/2018/06/06/microsoft-compra-github/>

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

Sí, se puede trabajar desde github.com.