

Mochila del viajero

Práctica individual obligatoria

Consigna

4. En la misma crear un archivo respondiendo las siguientes preguntas:

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

Git es un sistema de control de versiones distribuido que fue creado por Linus Torvalds en 2005 para ser utilizado en el desarrollo del kernel de Linux. Git es un software escrito en C, con interfaz por línea de comandos, disponible para Windows, MacOS X y GNU/Linux. En la actualidad, existen además multitud de clientes gráficos y herramientas basadas en Git.

Originalmente, la comunidad de desarrolladores de Linux utilizaba BitKeeper, un sistema de control de versiones distribuido y comercial, aunque gratuito para proyectos de código abierto. Como el propio Linus Torvalds explica, BitKeeper suponía un gran paso adelante comparado con otros SCMs como CVS y Subversion y, sin duda, inspiró muchos de los conceptos en los que se basa Git (especialmente el ser un sistema distribuido).

Sin embargo, el hecho de ser un software propietario y ciertas desavenencias con su creador, Larry McVoy, llevaron a Linus Torvalds a decantarse por buscar un nuevo sistema de control de versiones. Tras probar muchas de las soluciones existentes y descubrir que ninguna cumplía con sus requisitos, Torvalds decidió diseñar e implementar su propio sistema de control de versiones. Tan sólo un día después ya había desarrollado las bases de Git para poder auto-gestionar el propio código de Git en Git y, una semana después, ya tenía una primera versión. Resulta curioso que en la primera mención pública de Git, cuando llevaba un par de días de desarrollo, Torvalds describe Git no como un sistema de control de versiones sino simplemente como una herramienta de distribución y archivado de ficheros.

Git fue diseñado por Torvalds con el proyecto de Linux y el workflow de sus desarrolladores en mente y, obviamente, estos fueron los primeros en adoptarlo como sistema de control de versiones. El modo de funcionamiento distribuido, su eficiencia a la hora de trabajar con múltiples ramas y otras operaciones comunes como diffs, dan testimonio de cómo Git fue diseñado con la intención de gestionar proyectos grandes como el kernel sin problemas.

Tomado de: <https://www.conasa.es/blog/git-10-anos-controlando-versiones/>

b. ¿A quién pertenece actualmente Github y por qué?

Actualmente GitHub le pertenece a Microsoft, luego de la compra que se llevó acabo en el año 2018. Con el control de GitHub, Microsoft se hace también con la comunidad de desarrollo más nutrida del planeta. Con ese movimiento, Microsoft se hizo con una de las comunidades de desarrolladores más importantes de todo el mundo. Eso podría ser lo que más importa por encima de la tecnología de GitHub o del talento que encierra esta empresa.

De hecho, el segmento del desarrollo software está desde hace tiempo muy alineado con un segmento móvil en el que Microsoft apenas tiene participación. Esa es probablemente otra de las motivaciones de Microsoft, que con la adquisición de GitHub quiere mantener su relevancia en el desarrollo de todo tipo de plataformas, incluidas por supuesto las orientadas a dispositivos móviles. Muchos desarrolladores no confían en Microsoft, por lo que luego de la adquisición realizaron migraciones de proyectos a plataformas como GitLab y continúan allí sus desarrollos.

GitHub es referente incluso para proyectos que no son específicamente de código, y allí nos encontramos con proyectos que se encargan de gestionar versiones de canciones, recetas o tipografías, así como guías y listas colaborativas. No solo eso: aunque es célebre por su gestión de proyectos de código abierto, la propia Microsoft lo ha utilizado como sistema de control de versiones para su código propietario, y es que en GitHub también caben esos proyectos "no-OpenSource".

Tomado de: <https://www.xataka.com/aplicaciones/microsoft-punto-adquirir-github-desarrolladores-desarrolladores-desarrolladores>

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

Sí, existe un asistente visual para manejar GitHub: *GitHub Desktop*, disponible en <https://desktop.github.com/>

También existe un editor basado en web que se encuentra en fase beta: *github.dev editor*, información disponible en <https://docs.github.com/es/codespaces/the-githubdev-web-based-editor>