|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NOMBRE  Softwares de controles de versiones (Top 5) | CARACTERÍSTICAS | VENTAJAS | DESVENTAJAS | FABRICANTE | PRECIO |
| Git | 1. Distribuido y Descentralizado  2. Rastreo Preciso de Cambios  3. Ramificación Eficiente  4. Fusiones Simplificadas | 1. Historial Completo y Rastreable  2. Desarrollo Colaborativo sin Fricción  3. Reversión Sencilla y Segura  4. Flujo de Trabajo Flexible  5. Integración con Plataformas de Colaboración | 1.Es más complejo que los sistemas centralizados tradicionales  2.La curva de aprendizaje es empinada  3.Los comandos y algunos conceptos que usa pueden llegar a ser confusos | Linus Torvalds en 2005 | 0,06 USD por GB/mes. 0,001 USD por solicitud de Git. |
| CVS | Se basa en una arquitectura cliente-servidor como la de SVN, lleva presente mucho tiempo en el mundo del desarrollo y es una opción excelente para los proyectos de código abierto. | Muy buena comunidad para la resolución de dudas y búsqueda de soluciones.  Gran soporte multiplataforma.  Línea de comandos robusta y útil para el uso de scripts. | No soporta revisiones firmadas, como SVN.  Poco soporte para el control de código fuente distribuido.  No contempla checkouts ni commits atómicos. | CVS fue desarrollado por GNU | $753.64. |
| Apache Subversión  (SVN) | Se basa en una arquitectura cliente-servidor. Presenta algunas funcionalidades interesantes, como versiones de enlaces simbólicos, soporte para directorios vacíos y registros de operaciones de copiar, mover, borrar y renombrar. | Fácil de configurar y administrar.  Muy buen soporte para Windows.  Integración excelente con los principales IDE y las herramientas ágiles. | Al basarse en una arquitectura cliente-servidor, todas las versiones deben almacenarse en el mismo lugar.  Generación de ramas más compleja.  Repositorios más pesados.  Sin información sobre la hora de modificación de los archivos.  Sin soporte para versiones firmadas. | Desarrollado por el Ministerio de Economía | $1.17 |
| Mercurial | Es un sistema de control de versiones distribuido ideal para usuarios poco experimentados o perfiles poco técnicos. | Intuitivo y sencillo de manejar, con una curva de aprendizaje mucho más corta.  Veloz, potente y ligero.  Manejo sólido de archivos de texto plano y, también, binarios. | No funciona bien con extensiones.  Presenta menos funcionalidades que Git.  Solo soporta complementos programados en Python. | Matt Mackall | Gratis |
| Monotone | Es una herramienta de control de versiones distribuida de código abierto que puede integrarse en multitud de sistemas operativos. Prima la integridad ante el rendimiento, emplea técnicas criptográficas para rastrear las versiones y autenticaciones de archivos, puede importar proyectos CVS y utiliza el protocolo netsync. | Excelente funcionamiento de la ramificación.  Buena documentación oficial.  Requiere poco mantenimiento.  Sencillo, con una curva de aprendizaje corta.  Interfaz gráfica intuitiva. | Problemas de rendimiento en algunas operaciones.  No permite hacer checkout ni commit detrás del proxy. | Autor: Graydon Hoare  Desarrolladores: Nathaniel Smith, Graydon Hoare | Gratis |

CONCLUSIÓN

Con base a nuestra investigación de controles de software el que elegimos es Git porque te permite trabajar sin conexión o de forma remota con facilidad. Además de ser una forma segura de trabajar y permite el desarrollo colaborativo siendo así un flujo flexible de trabajo e integración.