|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Nombre  Stefanny Rodríguez Jiménez  Elizabeth Castillo Castillo | | Concepto | Características | Ventajas | | Desventajas |
| Git | Es un modelo de repositorio distribuido compatible con sistemas y protocolos existentes como  HTTP, FTP, SSH y es capaz de manejar eficientemente proyectos pequeños a grandes. | | La peculiaridad de **Git**radica en soportar y gestionar de la mejor manera posible:  desarrollo distribuido  performance elevate para proyectos grandes  la preservación del historial de cambios | | **Sistema distribuido**  **Super rápido y ligero**,  **Crear ramas y mezclarlas**  **La integridad de la información está asegurada**  Permite flujos de trabajo muy flexibles.  **El concepto de área de preparación**  ¡Es gratis! y de código abierto. | **Es más complejo** porque entran en juego más repositorios, más operaciones y más posibilidades para trabajar en equipo,  **La curva de aprendizaje es empinada**  **Los comandos y algunos conceptos** que usa pueden llegar a ser **confusos**  **se lleva mal con archivos binarios muy grandes**, como vídeos o documentos gráficos muy pesados |
| CVS | Es un modelo de repositorio cliente-  servidor donde varios desarrolladores pueden trabajar en el mismo proyecto en paralelo. | | Este sistema nos proporciona  Almacenamiento de elementos a gestionar  Realizar cambios a los elementos almacenados  Registro histórico de los cambios realizados a los elementos | | 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. |
| Apache Subversion | Es un  modelo de repositorio cliente-servidor donde los directorios están versionados junto con las  operaciones de copia, eliminación, movimiento y cambio de nombre. | | Versionado de directorios  Verdadero historial de versiones  Envíos atómicos  Versionado de metadatos | | Se sigue la historia de los archivos y directorios a través de copias y renombrados.  Las modificaciones (incluyendo cambios a varios archivos) son atómicas. | La principal desventaja de Subversion es que es más lento que CVS y que una verificación local de Subversion requiere mayor espacio en disco.  El manejo de cambio de nombres de archivos no es completo. Lo maneja como la suma de una operación de copia y una de borrado.  El manejo de archivos binarios los trata internamente como si fuera de texto no como de Subversion. |
| Mercurial | control de versiones que está escrita en Python y  destinada a desarrolladores de software. | | Rápido, escalable y maneja proyectos de todos los tamaños de manera efectiva. | | 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. |
| Monotone | está escrito en C ++ y es una herramienta para el control de versiones distribuido. | | control de versiones distribuido, autenticación con cifrados, acceso de control, su pequeño tamaño, tiene pocas dependencias, el almacenamiento de los proyectos usa una base de datos SQLite comprimida, | | 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. |
|  |  | |  | |  |  |

El que nos pareció muy interesante fuel el Git ya que además de que es flexible se puede usar para todo tipo de proyectos no importando si son grandes o pequeños