# Comparación de Software de Control de Versiones

A continuación se presenta un cuadro comparativo con las ventajas, desventajas, precios y si el software es de código abierto.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Software | Ventajas | Desventajas | Precio | Código Abierto |
| Git | - Distribuido, permite trabajo sin conexión - Amplio soporte y comunidad - Integración con múltiples plataformas y servicios | - Curva de aprendizaje pronunciada para principiantes - Complejidad en la gestión de grandes repositorios | Gratuito | Sí |
| Apache Subversion | - Sistema centralizado, fácil de entender - Buen manejo de archivos grandes - Soporte robusto para control de versiones | - Menos eficiente en operaciones de bifurcación y fusión - Requiere conexión constante al servidor central | Gratuito | Sí |
| Mercurial | - Interfaz de usuario sencilla - Buen rendimiento con repositorios grandes - Manejo eficiente de bifurcaciones y fusiones | - Menor adopción en la industria comparado con Git - Comunidad y soporte más pequeños | Gratuito | Sí |
| Perforce Helix Core | - Alto rendimiento para proyectos a gran escala - Control de versiones centralizado y distribuido - Herramientas avanzadas de gestión de activos digitales | - Modelo de precios basado en usuarios, puede ser costoso para equipos grandes - Curva de aprendizaje para configuraciones avanzadas | Modelo de precios basado en usuarios | No |
| Plastic SCM | - Soporte para flujos de trabajo centralizados y distribuidos - Interfaz gráfica de usuario intuitiva - Manejo eficiente de grandes archivos y proyectos | - Menor adopción en la industria - Integraciones limitadas en comparación con Git | Versión gratuita para equipos pequeños; suscripción para equipos grandes | No |