**Instituto Tecnológico de Los Mochis.**



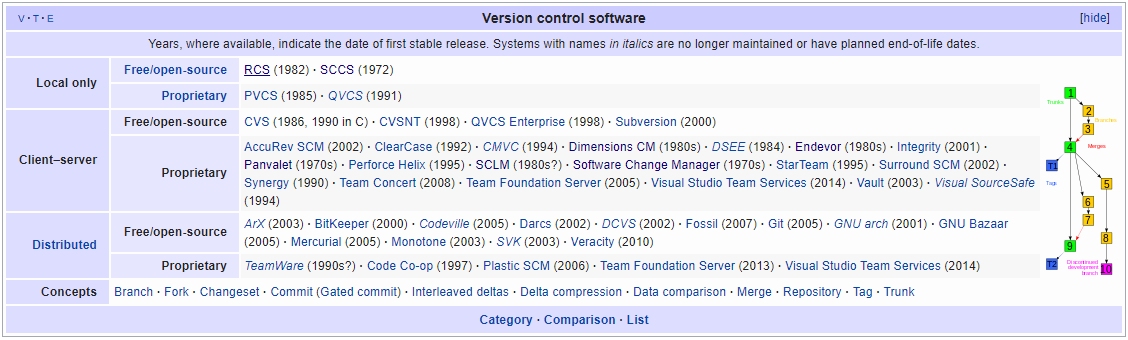
Ingeniería en Informática.

Herramientas de Apoyo para el Desarrollo de Software.

**Alumno:** Sánchez Zavala José Alejandro.

**Profesora:** Algara Norzagaray Juan Francisco.

Los Mochis, Sinaloa, a Febrero del 2018.



**1970s**

Computer Associates **Panvalet®** (también conocido como **CA-Panvalet**) es un control de revisión y un sistema de administración de código fuente para computadoras mainframe como IBM System z e IBM System / 370 que ejecutan los sistemas operativos [z / OS](https://en.wikipedia.org/wiki/Z/OS) y [z / VSE](https://en.wikipedia.org/wiki/VSE_(operating_system)) . A diferencia de las soluciones de código abierto como [CVS](https://en.wikipedia.org/wiki/Concurrent_Versions_System) , [SVN](https://en.wikipedia.org/wiki/Apache_Subversion) o [Mercurial](https://en.wikipedia.org/wiki/Mercurial) , Panvalet es un sistema patentado de código cerrado para el control de versiones y el código fuente como [Microsoft Visual SourceSafe](https://en.wikipedia.org/wiki/Microsoft_Visual_SourceSafe) en computadoras personales. Utiliza un [cliente-servidor](https://en.wikipedia.org/wiki/Client-server)modele donde los usuarios revisan los archivos para cambiarlos y revíselos en el depósito una vez finalizado. [[1]](https://en.wikipedia.org/wiki/Panvalet#cite_note-1)

Panvalet se puede usar para administrar el código fuente del programa, el [JCL](https://en.wikipedia.org/wiki/Job_Control_Language) y los [archivos del módulo de objetos](https://en.wikipedia.org/wiki/Object_file) . Admite controles de acceso granulares que incluyen el registro y la salida por ID de usuario de mainframe específicos. [[2]](https://en.wikipedia.org/wiki/Panvalet#cite_note-2)

En 1978, se informó que Panvalet, en ese momento un producto de Pansophic Systems, Inc, estaba en uso en más de 3.000 sitios.

|  |  |
| --- | --- |
| **CA Panvalet** | |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Computer Associates](https://en.wikipedia.org/wiki/CA_Technologies) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [z / OS](https://en.wikipedia.org/wiki/Z/OS) , [z / VSE](https://en.wikipedia.org/wiki/VSE_(operating_system)) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | Propiedad |
| **Sitio web** | [www .ca .com / us/ products / detail / ca-panvalet .aspx](http://www.ca.com/us/products/detail/ca-panvalet.aspx) |

**CA Harvest Software Change Manager** (originalmente conocido como **CCC / Harvest** ) es una herramienta de [software](https://en.wikipedia.org/wiki/Computer_software) para la [gestión de configuración](https://en.wikipedia.org/wiki/Configuration_management) ( [control de revisión](https://en.wikipedia.org/wiki/Revision_control) , [SCM](https://en.wikipedia.org/wiki/Software_configuration_management) , etc.) del [código fuente](https://en.wikipedia.org/wiki/Source_code) y otros activos de desarrollo de software.

El primer producto CCC (acrónimo de 'Cambio y control de configuración') se lanzó a principios de los años 70 y fue diseñado como un proyecto para un contratista del Departamento de Defensa en Santa Barbara CA. (La compañía en ese momento era [Hughes Aircraft](https://en.wikipedia.org/wiki/Hughes_Aircraft) , ahora el Centro de Investigación de Santa Bárbara para [Raytheon](https://en.wikipedia.org/wiki/Raytheon) ). Se convirtió en la primera herramienta de CM disponible comercialmente.

CCC fue diseñado para gestionar todos los componentes que entraron en un [motor de avión](https://en.wikipedia.org/wiki/Aircraft_engine) , y ya que el mismo motor fue utilizado tanto por la [Fuerza Aérea de Estados Unidos](https://en.wikipedia.org/wiki/United_States_Air_Force) y de [la marina de guerra de Estados Unidos](https://en.wikipedia.org/wiki/United_States_Navy) (para el [F-14 Tomcat](https://en.wikipedia.org/wiki/F-14_Tomcat) y [F-15 Eagle](https://en.wikipedia.org/wiki/F-15_Eagle) ) que requieren robusto y fiable desarrollo paralelo.

La primera versión de **CCC / Harvest** fue desarrollada comercialmente por Softool Corporation, una compañía de software enfocada en CM fundada en 1977 en Goleta, California. Otras herramientas de CCC incluyen CCC / Manager, CCC / DM Turnkey y CCC / QuickTrak.

Softool fue adquirida a fines de 1995 por [Platinum Technology](https://en.wikipedia.org/wiki/Platinum_Technology) , que luego fue adquirida en mayo de 1999 por Computer Associates (ahora conocida como [CA Technologies](https://en.wikipedia.org/wiki/CA_Technologies) ) que agregó CCC / Harvest a su suite AllFusion. En 2002, la parte 'CCC' del nombre se eliminó, y se agregó 'Change Manager', por lo que se hizo conocido como **AllFusion Harvest Change Manager** . Más tarde esto fue cambiado a **CA Harvest Software Change Manager** .

Funciones distintivas.

* **Cambie los paquetes:** Harvest puede proporcionar control de versiones y administración de cambios. El desarrollador realiza cambios en Harvest contra un paquete de cambio (creando un "conjunto de cambios"). Los paquetes de cambio consistirán inicialmente en una cantidad de archivos que el desarrollador ha creado o modificado. Este es el componente de control de versiones de Harvest.
* **Ciclos de vida: una** vez que el desarrollador está satisfecho con sus cambios, los cambios progresan a través de un ciclo de vida predefinido (es decir, en un número de etapas de PRUEBA secuenciales y finalmente en PRODUCCIÓN). En todas estas etapas de este "ciclo de vida", el paquete debe tener aprobaciones de los usuarios o grupos de usuarios apropiados. Estas aprobaciones se registran permanentemente para fines de auditoría. Por ejemplo, un gerente de pruebas puede tener que aprobar paquetes antes de pasar a la etapa de PRUEBA, y el equipo de administración de cambios de producción puede tener que aprobar paquetes antes de pasar al estado PROD.
* **Proyectos (Ambientes):** La filosofía de Central to Harvest es el concepto de un "proyecto" de cosecha. Los proyectos son totalmente personalizables según las necesidades de la aplicación, la organización o el equipo. El término proyecto se refiere a todo el marco de control en Harvest e incluye:
  + Una rama o línea de desarrollo separada donde los cambios pueden aislarse (el componente de control de versiones)
  + La definición de procesos y cómo los cambios progresan a través del ciclo de vida promocional
  + Control de acceso para procesos y archivos.

1972

**Source Code Control System** (**SCCS**) es un [sistema de control de versiones](https://en.wikipedia.org/wiki/Version_control) diseñado para rastrear los cambios en el [código fuente](https://en.wikipedia.org/wiki/Source_code) y otros archivos de texto durante el desarrollo de una pieza de software. Esto le permite al usuario recuperar cualquiera de las versiones anteriores del código fuente original y los cambios que se almacenan. Originalmente fue desarrollado en [Bell Labs](https://en.wikipedia.org/wiki/Bell_Labs) en 1972 por [Marc Rochkind](https://en.wikipedia.org/wiki/Marc_Rochkind) para una [computadora IBM System / 370](https://en.wikipedia.org/wiki/IBM_System/370) que ejecuta [OS / 360](https://en.wikipedia.org/wiki/OS/360) .

|  |  |
| --- | --- |
| **fuente** | |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | Marc J. Rochkind |
| **Versión inicial** | 1972 ; Hace 46 años |
| [**Repositorio**](https://en.wikipedia.org/wiki/Repository_(version_control)) | [Página web oficial](http://pubs.opengroup.org/) |
| **Escrito en** | [do](https://en.wikipedia.org/wiki/C_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Unix-like](https://en.wikipedia.org/wiki/Unix-like) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de versiones](https://en.wikipedia.org/wiki/Version_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | licencias de propiedad, licencia de[desarrollo común y distribución](https://en.wikipedia.org/wiki/Common_Development_and_Distribution_License) |
| **Sitio web** | [Las especificaciones de la base Open Group Edición 7, IEEE Std 1003.1-2008, edición de 2016](http://pubs.opengroup.org/onlinepubs/9699919799/utilities/sccs.html) |

Se considera el primer sistema de control de versiones oficial de la historia, ya que es la herramienta de control de versiones más antigua conocida. Sin rival hasta la llegada de RCS. SCCS basaba su gestión de las versiones en el procesamiento de ficheros individuales, además de cara a la concurencia, requería que cada uno de los desarrolladores del proyecto realizasen cambios en un momento determinado. Las versiones era protegidas mediante protecciones de seguridad a modo de bloqueo para el acceso del resto de desarrolladores, lo que originaba problemas de administración por olvido de retirar estos bloqueos. En 1972, [Marc Rochkind](https://en.wikipedia.org/wiki/Marc_Rochkind) desarrolló SCCS en [SNOBOL4](https://en.wikipedia.org/wiki/SNOBOL) en [Bell Labs](https://en.wikipedia.org/wiki/Bell_Labs) para una [computadora IBM System / 370](https://en.wikipedia.org/wiki/IBM_System/370) que ejecuta [OS / 360](https://en.wikipedia.org/wiki/OS/360)[MVT](https://en.wikipedia.org/wiki/Multiprogramming_with_a_Variable_number_of_Tasks) . [[1]](https://en.wikipedia.org/wiki/Source_Code_Control_System#cite_note-r1-1) Más tarde fue reescrito por él en [C](https://en.wikipedia.org/wiki/C_(programming_language))para [UNIX](https://en.wikipedia.org/wiki/UNIX) , y luego se ejecuta en un [PDP-11](https://en.wikipedia.org/wiki/PDP-11) .

La primera versión lanzada al público fue SCCS versión 4 del 18 de febrero de 1977. [[3]](https://en.wikipedia.org/wiki/Source_Code_Control_System#cite_note-3) Estaba disponible con la edición [Programmer's Workbench](https://en.wikipedia.org/wiki/PWB/UNIX) (PWB) del [sistema operativo](https://en.wikipedia.org/wiki/Operating_system) . La versión 4 de SCCS fue la primera versión que utilizó un formato de archivo de historial basado en texto, las versiones anteriores sí utilizaron formatos de archivo de historial binario. La versión 4 ya no fue escrita o mantenida por Mark Rochkind. Posteriormente, SCCS se incluyó en las distribuciones comerciales [System III](https://en.wikipedia.org/wiki/UNIX_System_III) y [System V de](https://en.wikipedia.org/wiki/UNIX_System_V)[AT & T.](https://en.wikipedia.org/wiki/AT%26T)No tenía licencia con [32V](https://en.wikipedia.org/wiki/UNIX/32V) , el antecesor de [BSD](https://en.wikipedia.org/wiki/Berkeley_Software_Distribution) . [[4]](https://en.wikipedia.org/wiki/Source_Code_Control_System#cite_note-4) El conjunto de comandos de SCCS ahora es parte de la [especificación de UNIX único](https://en.wikipedia.org/wiki/Single_UNIX_Specification) .

SCCS fue el sistema de control de versiones dominante para Unix hasta que [los](https://en.wikipedia.org/wiki/Version_control) sistemas de [control de versiones](https://en.wikipedia.org/wiki/Version_control) posteriores , especialmente el [RCS](https://en.wikipedia.org/wiki/Revision_Control_System) y [CVS](https://en.wikipedia.org/wiki/Concurrent_Versions_System) posterior , obtuvieron una adopción más generalizada. Hoy en día, estos primeros sistemas de control de versiones generalmente se consideran obsoletos, particularmente en la comunidad de [código abierto](https://en.wikipedia.org/wiki/Open_source) , que en gran medida ha adoptado sistemas de [control de versiones distribuidas](https://en.wikipedia.org/wiki/Distributed_version_control) . Sin embargo, el formato de archivo SCCS todavía se usa internamente en algunos programas de control de versiones más nuevos, incluidos [*BitKeeper*](https://en.wikipedia.org/wiki/BitKeeper) y [*TeamWare*](https://en.wikipedia.org/wiki/TeamWare) . Este último es un frontend de SCCS. [*Sablime*](http://sablime.alcatel-lucent.com/) se ha desarrollado a partir de una versión modificada de SCCS [[5]](https://en.wikipedia.org/wiki/Source_Code_Control_System#cite_note-5)pero usa un formato de archivo de historial que es incompatible con SCCS. El formato de archivo SCCS utiliza una técnica de almacenamiento llamada [deltas intercalados](https://en.wikipedia.org/wiki/Interleaved_deltas) (o el tejido [[6]](https://en.wikipedia.org/wiki/Source_Code_Control_System#cite_note-6) ). Esta técnica de almacenamiento ahora es considerada por muchos desarrolladores de sistemas de [control de](https://en.wikipedia.org/wiki/Version_control) versiones como fundamentales para las técnicas avanzadas de [fusión](https://en.wikipedia.org/wiki/Merge_(version_control)) y control de versiones, [[7]](https://en.wikipedia.org/wiki/Source_Code_Control_System#cite_note-7) como la [fusión](https://en.wikipedia.org/wiki/Merge_(version_control)) "Precise [Codeville](https://en.wikipedia.org/wiki/Codeville) " ("pcdv").

Además de solucionar algunos [problemas del año 2000](https://en.wikipedia.org/wiki/Year_2000_problem) en 1999, no existe un desarrollo activo en las diversas versiones SCCS específicas del proveedor de UNIX. [[8]](https://en.wikipedia.org/wiki/Source_Code_Control_System#cite_note-8) En 2006, [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) (hoy parte de [Oracle](https://en.wikipedia.org/wiki/Oracle_Corporation) ) lanzó su versión [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) de SCCS como [código abierto](https://en.wikipedia.org/wiki/Open_source) bajo la [licencia CDDL](https://en.wikipedia.org/wiki/Common_Development_and_Distribution_License) como parte de sus esfuerzos para abrir Solaris.

1980

**Dimensions CM** es un producto de [gestión de configuración y cambio de software](https://en.wikipedia.org/wiki/Software_configuration_management)[[4]](https://en.wikipedia.org/wiki/Dimensions_CM#cite_note-4) desarrollado por [Serena Software](https://en.wikipedia.org/wiki/Serena_Software) . Incluye [control de revisiones](https://en.wikipedia.org/wiki/Revision_control) , [cambio](https://en.wikipedia.org/wiki/Change_management) , [construcción](https://en.wikipedia.org/wiki/Build_management)[[5]](https://en.wikipedia.org/wiki/Dimensions_CM#cite_note-5) y capacidades de [gestión de versiones](https://en.wikipedia.org/wiki/Release_management) . Nombres de productos anteriores:

* Dimensiones PCMS (Software SQL) [[7]](https://en.wikipedia.org/wiki/Dimensions_CM#cite_note-7)
* Dimensiones PVCS (Merant, Intersolv)

|  |  |
| --- | --- |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Serena Software](https://en.wikipedia.org/wiki/Serena_Software) |
| **Versión inicial** | 1980 ; Hace 38 años [[*citación necesitada*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 14.2 [[1]](https://en.wikipedia.org/wiki/Dimensions_CM#cite_note-1) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Multiplataforma](https://en.wikipedia.org/wiki/Cross-platform)[[2]](https://en.wikipedia.org/wiki/Dimensions_CM#cite_note-2)[[3]](https://en.wikipedia.org/wiki/Dimensions_CM#cite_note-3) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Gestión de configuración de software](https://en.wikipedia.org/wiki/Software_configuration_management) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Serena Software](https://en.wikipedia.org/wiki/Serena_Software)[EULA](https://en.wikipedia.org/wiki/End-user_license_agreement) |
| **Sitio web** | [serena.com/index.php/en/products/featured-products/dimensions-cm](http://www.serena.com/index.php/en/products/featured-products/dimensions-cm/) |

**Endevor** es una herramienta de [administración de lanzamiento](https://en.wikipedia.org/wiki/Release_management) y [administración de código fuente](https://en.wikipedia.org/wiki/Source_code_management) para [computadoras mainframe que](https://en.wikipedia.org/wiki/Mainframe_computer) ejecutan [z / OS](https://en.wikipedia.org/wiki/Z/OS) . [[1]](https://en.wikipedia.org/wiki/Endevor#cite_note-1) Es parte de una familia de herramientas de administración de [CA Technologies](https://en.wikipedia.org/wiki/CA_Technologies) (anteriormente Computer Associates), que se utiliza para mantener aplicaciones de software y rastrear sus versiones. [[2]](https://en.wikipedia.org/wiki/Endevor#cite_note-2)

La palabra ENDEVOR es un acrónimo que originalmente significaba Medio ambiente para desarrolladores y operaciones, pero que ahora es el nombre formal del producto principal de control de fuente de [Application Lifecycle Management](https://en.wikipedia.org/wiki/Application_Lifecycle_Management) de CA mainframe . También compite contra otra herramienta de gestión de código fuente de CA, [Panvalet](https://en.wikipedia.org/wiki/Panvalet) .

**IBM Software Configuration and Library Manager** (SCLM) es un producto de software de [IBM](https://en.wikipedia.org/wiki/IBM) que es un componente de [ISPF](https://en.wikipedia.org/wiki/ISPF) .

Se introdujo por primera vez con la versión 3 de ISPF y fue una actualización de un componente anterior llamado LMF (Facilidad de administración de bibliotecas) que se introdujo en la versión 2 de ISPF. [[1]](https://en.wikipedia.org/wiki/IBM_Software_Configuration_and_Library_Manager#cite_note-1) Es un componente de [MVS](https://en.wikipedia.org/wiki/MVS) , [OS / 390](https://en.wikipedia.org/wiki/OS/390) y [z / OS](https://en.wikipedia.org/wiki/Z/OS)[sistemas operativos](https://en.wikipedia.org/wiki/Operating_system) / [plataformas](https://en.wikipedia.org/wiki/Computing_platform) . El SCLM consta de dos productos: un administrador de biblioteca y un [administrador de configuración](https://en.wikipedia.org/wiki/Software_configuration_management) y proporciona las siguientes funciones:

1. Administra los cambios en los datos de la aplicación
2. Realiza auditorías y [versiones](https://en.wikipedia.org/wiki/Version_control)
3. Controla el movimiento de los datos de la aplicación de un conjunto de bibliotecas de etapas al siguiente (conocido como Promover en SCLM)
4. Rastrea los componentes de la aplicación
5. Proporciona una función de compilación inteligente
6. Proporciona una promoción segura
7. Se integra con [Tivoli Information Management](https://en.wikipedia.org/w/index.php?title=IBM_Tivoli_Framework&action=edit&redlink=1) para z / OS y [WebSphere](https://en.wikipedia.org/wiki/WebSphere) Studio Asset Analyzer para z / OS

El recurso IBM Software Configuration and Library Manager (SCLM) for z/OS es un producto de gestión de configuración de software (SCM). SCLM for z/OS consta de dos productos: un gestor de bibliotecas y un gestor de configuración para la plataforma IBM z/OS. SCLM for z/OS es una característica de ISPF (Interactive System Productivity Facility), que se incluye con el sistema operativo z/OS.

SCLM for z/OS incluye:

* **Gestor de bibliotecas**: gestiona los cambios en los datos de las aplicaciones, lleva a cabo las auditorías y el mantenimiento de versiones y controla el traspaso de la aplicación de un conjunto de bibliotecas intermedias al siguiente.
* **Gestor de configuración**: sabe cómo encajan todas las piezas de una aplicación, incluidos los módulos de origen, de objetos y de carga. Permite especificar relaciones adicionales en SCLM para asociar otros datos, como casos de prueba, publicaciones y JCL, con la aplicación.
* **Gestión de cambios**: permite asociar un código de cambio con un miembro y, a continuación, crear, promover y crear informes según los códigos de cambio. También proporciona una interfaz de muestra para IBM Tivoli Information Management for z/OS, que realiza el seguimiento de los registros de problemas y cambios hasta la finalización.
* **Análisis del impacto**: lleva a cabo análisis hipotéticos de las posibles consecuencias de un cambio antes de que se realice una compilación o una promoción. Genera informes personalizados del contenido de las aplicaciones controladas por SCLM e informes que muestran la relación entre las partes de la aplicación.
* **Características de ISPF**: proporciona funciones de visualización y exploración de SCLM que son iguales a sus equivalentes de ISPF. La función de edición de SCLM se crea sobre la edición de ISPF y admite las macros y los mandatos de edición de ISPF, la edición sensible al idioma y la herramienta Edit Compare. Una opción de tabla de configuración de ISPF determina si se puede acceder a un miembro desde fuera de SCLM.

1982 **Revision Control System** **(RCS)** es un [sistema de control de versiones](https://en.wikipedia.org/wiki/Version_control_system) inicial ( [VCS](https://en.wikipedia.org/wiki/Version_control_system) ). Se puede considerar como un conjunto de comandos de [UNIX](https://en.wikipedia.org/wiki/Unix) que permiten que varios usuarios desarrollen y mantengan código de programa o documentos. Con RCS, los usuarios pueden hacer sus propias revisiones de un documento, realizar cambios y fusionarlos. RCS fue desarrollado originalmente para programas, pero también es útil para documentos de texto o archivos de configuración que se revisan con frecuencia.

|  |  |
| --- | --- |
| **Sistema de control de revisión** | |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Walter F. Tichy](https://en.wikipedia.org/wiki/Walter_F._Tichy) |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Proyecto GNU](https://en.wikipedia.org/wiki/GNU_Project) |
| **Versión inicial** | 1982 ; Hace 36 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 5.9.4 (22 de enero de 2015 ; hace 3 años [[1]](https://en.wikipedia.org/wiki/Revision_Control_System#cite_note-1) )[[±]](https://en.wikipedia.org/w/index.php?title=Template:Latest_stable_software_release/Revision_Control_System&action=edit) |
| [**Repositorio**](https://en.wikipedia.org/wiki/Repository_(version_control)) | * <http://git.savannah.gnu.org/cgit/rcs.git/?h=p>   [Edita esto en Wikidata](https://www.wikidata.org/wiki/Q705851#P1324) |
| **Escrito en** | [do](https://en.wikipedia.org/wiki/C_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Unix-like](https://en.wikipedia.org/wiki/Unix-like) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de versiones](https://en.wikipedia.org/wiki/Version_Control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Licencia pública general de GNU](https://en.wikipedia.org/wiki/GNU_General_Public_License) |
| **Sitio web** | [www .gnu .org / s / rcs /](https://www.gnu.org/s/rcs/) |

### Modo de operación

RCS opera solo en archivos individuales. No tiene forma de trabajar con un proyecto completo, por lo que no admite [confirmaciones atómicas que](https://en.wikipedia.org/wiki/Atomic_commit#Revision_control) afecten a múltiples archivos. Aunque proporciona ramificación para archivos individuales, la sintaxis de la versión es engorrosa. En lugar de utilizar ramas, muchos equipos simplemente usan el mecanismo de bloqueo incorporado y trabajan en una sola rama de la *cabeza*.

### Uso

RCS gira en torno al uso de "grupos de revisión" o conjuntos de archivos que se han registrado a través de los comandos "co" (pago) y "ci" (check-in). De forma predeterminada, un archivo registrado se elimina y se reemplaza con un archivo ", v" (por lo que foo.rb cuando se activa se convierte en foo.rb, v) que luego cualquier persona que tenga acceso al grupo de revisión puede revisar. Los archivos RCS (nuevamente, los archivos con la extensión ", v") reflejan el archivo principal con metadatos adicionales en sus primeras líneas. Una vez registrado, RCS almacena las revisiones en una estructura de árbol que se puede seguir para que un usuario pueda revertir un archivo a un formulario anterior si es necesario. [[2]](https://en.wikipedia.org/wiki/Revision_Control_System#cite_note-:1-2)

### Ventajas

### Estructura simple y fácil de trabajar con

* El ahorro de revisión no depende de un repositorio central

### Desventajas

* Hay poca seguridad, en el sentido de que el historial de versiones puede ser editado por los usuarios.
* Solo un usuario puede trabajar en un archivo a la vez.

1984 DSEE

1985 **PVCS Version Manager** (originalmente llamado **Polytron Version Control System** ) es un paquete de software de [Serena Software](https://en.wikipedia.org/wiki/Serena_Software)Inc., para el [control](https://en.wikipedia.org/wiki/Version_control) de [versiones](https://en.wikipedia.org/wiki/Version_control) de archivos de [códigos fuente](https://en.wikipedia.org/wiki/Source_code) .

PVCS sigue el enfoque de "bloqueo" para el control de concurrencia; no tiene un operador de [*fusión*](https://en.wikipedia.org/wiki/Merge_(version_control)) incorporado (pero, sin embargo, tiene un comando de fusión separado). Sin embargo, PVCS también se puede configurar para admitir a varios usuarios al mismo tiempo que intentan editar el archivo; en este caso, el segundo committer cronológico tendrá una rama creada para él / ella, de modo que ambas modificaciones, en lugar de estar en conflicto, aparecerán como historias paralelas para el mismo archivo. Esto es diferente de [Concurrent Versions System](https://en.wikipedia.org/wiki/Concurrent_Versions_System) (CVS) y [Subversion](https://en.wikipedia.org/wiki/Subversion_(software)) donde el segundo committer necesita *fusionar* primero los cambios a través del comando de *actualización* y luego resolver los conflictos (cuando existen) antes de comprometerse.

Originalmente desarrollado por Don Kinzer y publicado por [Polytron](https://en.wikipedia.org/wiki/Polytron_(software)) en 1985, a través de una historia de adquisiciones y fusiones, el producto a veces fue propiedad de Sage Software de Rockville (1989), Maryland (no relacionado con [Sage Software](https://en.wikipedia.org/wiki/Sage_Software) del Reino Unido), [Intersolv](https://en.wikipedia.org/w/index.php?title=Intersolv&action=edit&redlink=1) 1992, [Micro Focus International](https://en.wikipedia.org/wiki/Micro_Focus_International) 1998 y [Merant PLC](https://en.wikipedia.org/wiki/Merant_PLC) 2001. Este último fue adquirido por Serena Software en 2004, que luego fue adquirido por [Silver Lake Partners](https://en.wikipedia.org/wiki/Silver_Lake_Partners) en 2006.

[Synergex](https://en.wikipedia.org/w/index.php?title=Synergex&action=edit&redlink=1) portó tanto el Gestor de versiones de PVCS como el Constructor de configuraciones de PVCS (una [utilidad de](https://en.wikipedia.org/wiki/Make_(software)) creación extendida , que incluye una variante de la herramienta de línea de comandos que hace [[2]](https://en.wikipedia.org/wiki/PVCS#cite_note-2)[[*mejor fuente necesaria*](https://en.wikipedia.org/wiki/Wikipedia:NOTRS)] ) a varias plataformas [Unix](https://en.wikipedia.org/wiki/Unix) y [OpenVMS](https://en.wikipedia.org/wiki/OpenVMS) . [[3]](https://en.wikipedia.org/wiki/PVCS#cite_note-3)

En 2009, Serena Software aclaró que continuará invirtiendo en PVCS y brindará soporte a los clientes de PVCS en el futuro previsible.   
La versión de PVCS Version Manager 8.5 (2014) presenta tanto la nueva característica como el nuevo soporte de plataforma. [[1]](https://en.wikipedia.org/wiki/PVCS#cite_note-blog-1)

En 2016, [Micro Focus International](https://en.wikipedia.org/wiki/Micro_Focus_International) anunció la adquisición de Serena Software para volver a ser los custodios de PVCS.

|  |  |
| --- | --- |
| **Gerente de versión de PVCS** | |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Polytron](https://en.wikipedia.org/wiki/Polytron) |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Serena Software](https://en.wikipedia.org/wiki/Serena_Software) |
| **Versión inicial** | 1985 ; Hace 33 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 8.5 / 2 de enero de 2014 ; Hace 4 años [[1]](https://en.wikipedia.org/wiki/PVCS#cite_note-blog-1) |
| **Escrito en** | [C](https://en.wikipedia.org/wiki/C_(programming_language)) , [C ++](https://en.wikipedia.org/wiki/C%2B%2B) , [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [DOS](https://en.wikipedia.org/wiki/DOS) , [[*cita requerida*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)][Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) , tipo [Unix](https://en.wikipedia.org/wiki/Unix-like) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de versiones](https://en.wikipedia.org/wiki/Version_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Software propietario](https://en.wikipedia.org/wiki/Proprietary_software) |
| **Sitio web** | [www .microfocus.com / products / pvcs /](https://www.microfocus.com/products/pvcs/) |

1986 **Concurrent Versions System** (**CVS**) también conocido como el **Sistema Concurrent Versioning** , es un sistema de [control de revisiones](https://en.wikipedia.org/wiki/Revision_control)[cliente-servidor de](https://en.wikipedia.org/wiki/Client-server)[software libre](https://en.wikipedia.org/wiki/Free_software) en el campo del [desarrollo](https://en.wikipedia.org/wiki/Software_development) de [software](https://en.wikipedia.org/wiki/Software_development) . Un sistema de control de versiones realiza un seguimiento de todo el trabajo y todos los cambios en un conjunto de archivos, y permite que varios desarrolladores (potencialmente muy separados en el espacio y el tiempo) [colaboren](https://en.wikipedia.org/wiki/Collaboration) . [Dick Grune](https://en.wikipedia.org/wiki/Dick_Grune) desarrolló CVS como una serie de [guiones](https://en.wikipedia.org/wiki/Shell_scripts) de [shell](https://en.wikipedia.org/wiki/Shell_scripts) en julio de 1986. [[1]](https://en.wikipedia.org/wiki/Concurrent_Versions_System#cite_note-1)

Además de los desarrolladores de software comercial, CVS se hizo popular con el mundo del [software de código abierto](https://en.wikipedia.org/wiki/Open_source_software) y fue lanzado bajo la [Licencia Pública General de GNU](https://en.wikipedia.org/wiki/GNU_General_Public_License) . Si bien hubo un desarrollo regular para agregar funciones y corregir errores en el pasado, [[2]](https://en.wikipedia.org/wiki/Concurrent_Versions_System#cite_note-2) incluidas compilaciones regulares y resultados de pruebas, [[3]](https://en.wikipedia.org/wiki/Concurrent_Versions_System#cite_note-3) no ha habido nuevas versiones desde 2008.

|  |  |
| --- | --- |
| **Sistema de versiones concurrentes** | |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [El equipo de CVS](http://savannah.nongnu.org/project/memberlist.php?detailed=1&group=cvs) |
| **Versión inicial** | 19 de noviembre de 1990 ; Hace 27 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 1.11.23 / 8 de mayo de 2008 ; Hace 9 años |
| [**Lanzamiento de vista previa**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 1.12.13 / 26 de julio de 2006 ; Hace 11 años |
| [**Repositorio**](https://en.wikipedia.org/wiki/Repository_(version_control)) | * cvs: //anonymous@cvs.sv.gnu.org/sources/cvs   [Edita esto en Wikidata](https://www.wikidata.org/wiki/Q467252#P1324) |
| **Escrito en** | [do](https://en.wikipedia.org/wiki/C_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Unix](https://en.wikipedia.org/wiki/Unix-like) , [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Licencia pública general de GNU](https://en.wikipedia.org/wiki/GNU_General_Public_License) |
| **Sitio web** | [savannah .nongnu .org / projects/ cvs](http://savannah.nongnu.org/projects/cvs) |

CVS utiliza una arquitectura [cliente-servidor](https://en.wikipedia.org/wiki/Client%E2%80%93server) : un servidor almacena la (s) versión (es) actual (es) de un [proyecto](https://en.wikipedia.org/wiki/Project) y su historial, y los clientes se conectan al servidor para "verificar" una copia completa del proyecto, trabajar en esta copia y luego más tarde "registrar" sus cambios. Normalmente, el cliente y el servidor se conectan a través de una [LAN](https://en.wikipedia.org/wiki/Local_area_network) o de [Internet](https://en.wikipedia.org/wiki/Internet) , pero tanto el cliente como el servidor pueden ejecutarse en la misma máquina si CVS tiene la tarea de realizar un seguimiento del historial de versiones de un proyecto solo con desarrolladores locales. El software del servidor normalmente se ejecuta en [Unix](https://en.wikipedia.org/wiki/Unix) (aunque al menos el servidor [CVSNT](https://en.wikipedia.org/wiki/CVSNT) también es compatible con varios sabores de [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) ), mientras que los clientes de CVS pueden ejecutarse en cualquierplataforma del [sistema operativo](https://en.wikipedia.org/wiki/Operating_system) .

Varios desarrolladores pueden trabajar en el mismo proyecto al mismo tiempo, cada uno editando archivos dentro de su propia "copia de trabajo" del proyecto y enviando (o *registrando* ) sus modificaciones al servidor. Para evitar conflictos, el servidor solo acepta los cambios realizados en la versión más reciente de un archivo. Por lo tanto, se espera que los desarrolladores mantengan actualizada su copia de trabajo incorporando los cambios de otras personas de forma regular. El cliente de CVS maneja esta tarea en su mayoría automáticamente, y requiere intervención manual solo cuando surge un [conflicto de edición](https://en.wikipedia.org/wiki/Edit_conflict) entre una modificación registrada y la versión local aún no verificada de un archivo.

Si la operación de verificación se realiza correctamente, los números de versión de todos los archivos involucrados se incrementan automáticamente, y el servidor CVS escribe una línea de descripción proporcionada por el usuario, la fecha y el nombre del autor en sus archivos de [registro](https://en.wikipedia.org/wiki/Data_logging) . CVS también puede ejecutar scripts externos de procesamiento de registro especificados por el usuario después de cada confirmación. Estas secuencias de comandos se instalan mediante una entrada en el archivo de inicio de sesión de CVS , que puede desencadenar notificaciones por correo electrónico o convertir los datos de registro en un formato basado en web.

Los clientes también pueden comparar versiones, solicitar un historial completo de cambios o consultar una instantánea histórica del proyecto a partir de una fecha determinada o de un número de revisión.

Los servidores CVS pueden permitir el "acceso de lectura anónimo", [[5]](https://en.wikipedia.org/wiki/Concurrent_Versions_System#cite_note-anoncvs-5) en el que los clientes pueden verificar y comparar versiones con una contraseña en blanco o simple publicada (por ejemplo, "anoncvs"); solo el registro de cambios requiere una cuenta personal y contraseña en estos escenarios.

Los clientes también pueden usar el comando "actualizar" para actualizar sus copias locales con la versión más reciente en el servidor. Esto elimina la necesidad de descargas repetidas de todo el proyecto.

CVS también puede mantener diferentes "ramas" de un proyecto. Por ejemplo, una versión publicada del proyecto de software puede formar una rama, utilizada para corregir errores, mientras que una versión bajo desarrollo actual, con cambios importantes y nuevas características, puede formar una rama separada.

CVS usa [compresión delta](https://en.wikipedia.org/wiki/Delta_compression) para un almacenamiento eficiente de diferentes versiones del mismo archivo. Esto funciona bien con archivos de texto grandes con pocos cambios de una versión a la siguiente. Este suele ser el caso de los archivos de código fuente. Por otro lado, cuando se le dice a CVS que almacene un archivo como binario, mantendrá cada versión individual en el servidor. Almacenar archivos como binarios es importante para evitar la corrupción de archivos binarios.

En el mundo del software de código abierto, el Concurrent Version System (CVS) ha sido durante mucho tiempo la herramienta de elección para el control de versiones. Y con razón. CVS en sí mismo es software libre, y su *modus operandi* no restrictivo y soporte para el funcionamiento en red, que permite a docenas de programadores geográficamente dispersos compartir su trabajo, encaja muy bien con la naturaleza colaborativa del mundo de código abierto. CVS y su modelo de desarrollo semi-caótico se han convertido en piedras angulares de código abierto.

-  Collins-Sussman, Control de versiones con Subversion para Subversion 1.1, 2005

1990 **Rational Synergy**  es una herramienta de software que proporciona capacidades de [gestión de configuración de software](https://en.wikipedia.org/wiki/Software_configuration_management) (SCM) para todos los artefactos relacionados con el desarrollo de software, incluidos el [código fuente](https://en.wikipedia.org/wiki/Source_code) , los documentos y las imágenes, así como las bibliotecas ejecutables y el software final. Rational Synergy también proporciona el repositorio de la herramienta de [administración de cambios](https://en.wikipedia.org/wiki/Change_management) conocida como Cambio Racional. Juntas, estas dos herramientas forman un entorno de [gestión de configuración](https://en.wikipedia.org/wiki/Configuration_management) y [gestión de configuración](https://en.wikipedia.org/wiki/Configuration_management) integrado que se utiliza en organizaciones de desarrollo de software que necesitan procesos controlados de SCM y una comprensión de lo que hay en una compilación de su software.

El nombre *Synergy se* refiere a su integración a nivel de base de datos con Change Management que proporciona vistas de lo que está en una construcción en términos de defectos.

|  |  |
| --- | --- |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | CaseWare, Inc. |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Software racional](https://en.wikipedia.org/wiki/Rational_Software) |
| **Versión inicial** | 1990 ; Hace 28 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 7.2.1.5 [[1]](https://en.wikipedia.org/wiki/Rational_Synergy#cite_note-1) / 23 de marzo de 2016 ; Hace 21 meses |
| **Escrito en** | Acento, [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [AIX](https://en.wikipedia.org/wiki/IBM_AIX) , [HP-UX](https://en.wikipedia.org/wiki/HP-UX) , [Linux](https://en.wikipedia.org/wiki/Linux) ,[Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) , [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Gestión de configuración de software](https://en.wikipedia.org/wiki/Software_configuration_management) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | IBM [EULA](https://en.wikipedia.org/wiki/Software_license) |
| **Sitio web** | [www .ibm .com/ software / productos / es / ratisyne](http://www.ibm.com/software/products/en/ratisyne) |

1990 **Sun WorkShop TeamWare** más tarde **Forte TeamWare** , luego **Forte Code Management Software** ) es un sistema distribuido de [control de revisión de](https://en.wikipedia.org/wiki/Revision_control)[código fuente](https://en.wikipedia.org/wiki/Source_code) creado por [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) . Último disponible como parte del producto [Forte Developer 6 actualización 2](https://en.wikipedia.org/wiki/Oracle_Solaris_Studio) , TeamWare ya no se ofrece para la venta, [[1]](https://en.wikipedia.org/wiki/Sun_WorkShop_TeamWare#cite_note-1) y no es parte del producto [Sun Studio](https://en.wikipedia.org/wiki/Sun_Studio_(software)).

La implementación más grande de TeamWare se realizó dentro de Sun, donde (salvo algunas excepciones) en un momento dado fue el único [VCS](https://en.wikipedia.org/wiki/Version_Control_System) utilizado. TeamWare se utilizó para gestionar los árboles fuente más grandes de Sun, incluidos los de [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) y [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) , pero como parte del proceso de conversión de esas bases de [código a](https://en.wikipedia.org/wiki/Open_source) comunidades de [código abierto](https://en.wikipedia.org/wiki/Open_source) , se trasladaron a sistemas de control de revisiones más nuevos como [Mercurial](https://en.wikipedia.org/wiki/Mercurial_(software)) .

TeamWare presenta una serie de características avanzadas que no se encuentran en sistemas de control de versiones anteriores como [RCS](https://en.wikipedia.org/wiki/Revision_Control_System) y [CVS](https://en.wikipedia.org/wiki/Concurrent_Versions_System) . En particular, presenta una jerarquía de repositorios de origen y permite actualizaciones atómicas de múltiples archivos, características que se encuentran en sistemas de control de versiones posteriores, como [Subversion](https://en.wikipedia.org/wiki/Subversion_(software)) y [Perforce](https://en.wikipedia.org/wiki/Perforce) . TeamWare permite el [desarrollo distribuido](https://en.wikipedia.org/wiki/Distributed_development) copiando un repositorio a otro que podría residir en otra máquina o red. Los desarrolladores pueden luego realizar cambios en la copia local del repositorio, integrando periódicamente los cambios acumulados en el repositorio local en el repositorio original.

TeamWare se implementa como una capa sobre el [SCCS](https://en.wikipedia.org/wiki/Source_Code_Control_System) anterior , que se utiliza para rastrear los cambios en los archivos individuales. TeamWare solo funciona mediante un sistema de archivos a los que acceden [los programas cliente](https://en.wikipedia.org/wiki/Client_(computing)) (que interactúan sin un [servidor](https://en.wikipedia.org/wiki/Server_(computing)) ) y la mayoría de los usuarios distribuidos de un repositorio acceden a él mediante un [sistema de archivos en red](https://en.wikipedia.org/wiki/Distributed_file_system) montado como [NFS](https://en.wikipedia.org/wiki/Network_File_System_(protocol)) .

Evan Adams fue el líder arquitectónico de TeamWare. [[2]](https://en.wikipedia.org/wiki/Sun_WorkShop_TeamWare#cite_note-2) Glenn Skinner fue el inventor de *smoosh*[[3]](https://en.wikipedia.org/wiki/Sun_WorkShop_TeamWare#cite_note-3)[se[*necesita una fuente mejor*](https://en.wikipedia.org/wiki/Wikipedia:NOTRS)] [[*discutir*](https://en.wikipedia.org/wiki/Talk:Sun_WorkShop_TeamWare#smoosh?)] y [Larry McVoy](https://en.wikipedia.org/wiki/Larry_McVoy) escribió *smoosh* , una herramienta para fusionar archivos SCCS, [[4]](https://en.wikipedia.org/wiki/Sun_WorkShop_TeamWare#cite_note-4) que se dice que es un precursor de TeamWare. [[5]](https://en.wikipedia.org/wiki/Sun_WorkShop_TeamWare#cite_note-5) El sistema de control de versiones [BitKeeper](https://en.wikipedia.org/wiki/BitKeeper) , diseñado por McVoy, comparte una serie de conceptos de diseño con TeamWare anterior.

1991 **QVCS** (Quma Version Control System) es una familia de productos de [sistemas de control de versiones](https://en.wikipedia.org/wiki/Version_control_system) históricamente publicados por Quma Software, Inc.

QVCS se publicó por primera vez en 1991 como un conjunto de utilidades de línea de comandos para [Amiga](https://en.wikipedia.org/wiki/Amiga) .

Quma llevó ese producto original de Amiga a la plataforma de [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) en 1996. En 2000, se introdujo QVCS-Pro con características adicionales, incluida la compatibilidad con [entornos de desarrollo integrados que](https://en.wikipedia.org/wiki/Integrated_development_environment)[cumplen](https://en.wikipedia.org/wiki/SCC_compliant) con la norma [SCC de](https://en.wikipedia.org/wiki/SCC_compliant)Microsoft . [[4]](https://en.wikipedia.org/wiki/QVCS#cite_note-4) En 2004, se introdujo el producto QVCS-Enterprise [multiplataforma](https://en.wikipedia.org/wiki/Cross-platform) . [[5]](https://en.wikipedia.org/wiki/QVCS#cite_note-5) QVCS y QVCS-Pro están escritos en [C ++](https://en.wikipedia.org/wiki/C%2B%2B) . QVCS-Enterprise está escrito en [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) .

El 1 de enero de 2012, QVCS-Pro y QVCS se convirtieron en software gratuito. QVCS-Enterprise hizo lo mismo el 16 de junio de 2012, y luego fue abierta bajo la licencia Apache 2.0 el 2 de marzo de 2014.

En algún momento después del 23 de junio de 2013, el nombre de dominio qumasoft.com fue abandonado. Ahora (junio de 2015) es propiedad de una compañía de pérdida de peso. El contenido del sitio web cuando fue operado por el autor de QVCS está disponible en [Wayback Machine](https://en.wikipedia.org/wiki/Wayback_Machine) .

|  |  |
| --- | --- |
| **QVCS** | |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | Jim Voris |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | Software Quma |
| **Versión inicial** | 1991 ; Hace 27 años |
| [**Último lanzamiento**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 3.10.20 [[1]](https://en.wikipedia.org/wiki/QVCS#cite_note-1) / 5 de abril de 2010 ; hace 7 años |
| **Estado del desarrollo** | sitio web abandonado[[2]](https://en.wikipedia.org/wiki/QVCS#cite_note-2) |
| **Escrito en** | [C ++](https://en.wikipedia.org/wiki/C%2B%2B) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | Freeware |

QVCS y QVCS-Pro se dirigen a equipos de desarrollo de Windows más pequeños. QVCS-Enterprise es una aplicación [cliente-servidor](https://en.wikipedia.org/wiki/Client%E2%80%93server), es multiplataforma y está diseñada para ser utilizada por equipos de desarrollo distribuidos.

**1992 Rational ClearCase** es una familia de herramientas de [software](https://en.wikipedia.org/wiki/Computer_software) que admite [la administración de configuración de software](https://en.wikipedia.org/wiki/Software_configuration_management) (SCM) del [código fuente](https://en.wikipedia.org/wiki/Source_code) y otros activos de [desarrollo de software](https://en.wikipedia.org/wiki/Software_development) . También es compatible con la gestión de datos de diseño de artefactos de diseño electrónico, lo que permite el codesarrollo de hardware y software. ClearCase incluye [control de revisiones](https://en.wikipedia.org/wiki/Revision_control) y forma la base para la administración de la configuración en grandes y medianas empresas, acomodando proyectos con cientos o miles de desarrolladores. Está desarrollado por [IBM](https://en.wikipedia.org/wiki/IBM) .

ClearCase admite dos modelos de gestión de configuración: [UCM](https://en.wikipedia.org/wiki/Rational_ClearCase_UCM) (Unified Change Management) y [ClearCase base](https://en.wikipedia.org/w/index.php?title=Base_ClearCase&action=edit&redlink=1) . UCM proporciona un modelo listo para usar mientras que ClearCase base proporciona una infraestructura básica (UCM se basa en ClearCase base). Ambos se pueden configurar para admitir una amplia variedad de necesidades.

ClearCase puede acomodar grandes archivos binarios, grandes cantidades de archivos y grandes tamaños de repositorio. Admite ramificación, etiquetado y control de versiones de directorios.

El sistema de [base de datos](https://en.wikipedia.org/wiki/Database) que utiliza ClearCase es [RDM Embedded](https://en.wikipedia.org/wiki/RDM_Embedded) from [Raima](https://en.wikipedia.org/wiki/Raima) . En la terminología ClearCase, una base de datos individual se llama un *VOB* ( **V** ersioned **O**bject **B** ASE). En esta capa, el mantenimiento se lleva a cabo utilizando herramientas de Raima. Alrededor de esta capa, un conjunto de interfaces con las herramientas que lo acompañan se utilizan para administrar el sistema de base de datos física, que requiere habilidades específicas de [administrador de la base de datos](https://en.wikipedia.org/wiki/Database_administrator) .

El servicio más importante es la ubicación de Atria Broker Daemon (ALBD), que gestiona todas las comunicaciones (LAN) entre computadoras. A partir de la versión 7, la plataforma del servidor ejecuta Websphere Application Server con una aplicación de servidor llamada Change Management Server (CM Server), que presta servicios a los clientes de ClearCase a través del protocolo HTTP. (Antes de la versión 7 existía un servicio web mediante el cual los usuarios podían acceder a ClearCase a través de sus navegadores.) Desde entonces, el servidor CM ha sido reemplazado por el servidor de red de área amplia ClearCase Remote Client (servidor WAN CCRC), que sigue basándose en la aplicación Websphere. Servidor.

|  |  |
| --- | --- |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Atria Software](https://en.wikipedia.org/wiki/Atria_Software) |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [IBM](https://en.wikipedia.org/wiki/IBM) |
| **Versión inicial** | 1992 ; Hace 26 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 9.0.1.2 [[1]](https://en.wikipedia.org/wiki/Rational_ClearCase#cite_note-1) / 18 de diciembre de 2017 ; Hace 37 días |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [AIX](https://en.wikipedia.org/wiki/IBM_AIX) , [HP-UX](https://en.wikipedia.org/wiki/HP-UX) , [Linux](https://en.wikipedia.org/wiki/Linux) ,[Linux en z Systems](https://en.wikipedia.org/wiki/Linux_on_z_Systems) ,[Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) , [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) , [z / OS](https://en.wikipedia.org/wiki/Z/OS) (cliente limitado) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Gestión de configuración de software](https://en.wikipedia.org/wiki/Software_configuration_management) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | IBM [EULA](https://en.wikipedia.org/wiki/Software_license) |
| **Sitio web** | [www .ibm .com/ software / productos / en / clearcase](http://www.ibm.com/software/products/en/clearcase) |

1994 **Configuration Management Version Control (CMVC)** es un paquete de [software](https://en.wikipedia.org/wiki/Software) que sirve como depósito de objetos y realiza [control de versiones de](https://en.wikipedia.org/wiki/Version_control) software , administración de configuraciones y funciones de administración de cambios.

|  |  |
| --- | --- |
| **Configuration Management** | |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [IBM](https://en.wikipedia.org/wiki/IBM) |
| **Estado del desarrollo** | Inactivo |
| **Escrito en** | [C ++](https://en.wikipedia.org/wiki/C%2B%2B) , [C](https://en.wikipedia.org/wiki/C_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | Linux, Mac, Windows, z / OS, AIX ([multiplataforma](https://en.wikipedia.org/wiki/Cross-platform) ) |
| **Disponible en** | Plurilingüe |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Desarrollo de software](https://en.wikipedia.org/wiki/Software_development) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Propiedad](https://en.wikipedia.org/wiki/Proprietary_software) |
| **Sitio web** | [www .ibm .com](http://www.ibm.com/) |

* El seguimiento de defectos se integró en el sistema de control de origen.
* Posibilidad de configurar un repositorio para exigir que se asocien defectos con cada registro de archivo.
* Los niveles se utilizan para gestionar conjuntos de defectos, lo que permite a los administradores pensar en términos de conjuntos de cambios en lugar de conjuntos de archivos.
* La administración distribuida permite la delegación de autoridad. Esto es particularmente útil en equipos grandes.
* Tiene la capacidad de rastrear historiales de archivos incluso después de cambiar el nombre. Esto se debe a que el nombre del archivo en el disco era un número y el nombre del archivo en la base de datos podría cambiarse.
* Los archivos pueden ser compartidos entre el código fuente

1994 **Microsoft Visual SourceSafe** (**VSS**) es un [programa de](https://en.wikipedia.org/wiki/Computer_program)[control de fuente](https://en.wikipedia.org/wiki/Source_control) descontinuado , orientado a pequeños proyectos de desarrollo de software. Como la mayoría de los sistemas de control de fuente, SourceSafe crea una *biblioteca virtual* de archivos de computadora. Aunque se usa más comúnmente para el [código fuente](https://en.wikipedia.org/wiki/Source_code) , SourceSafe puede manejar cualquier tipo de archivo en su [base de datos](https://en.wikipedia.org/wiki/Database) , pero las versiones anteriores se mostraron inestables [[1]](https://en.wikipedia.org/wiki/Microsoft_Visual_SourceSafe#cite_note-1)[[2]](https://en.wikipedia.org/wiki/Microsoft_Visual_SourceSafe#cite_note-2) cuando se usaban para almacenar grandes cantidades de datos no textuales, como imágenes, y compilaban ejecutables.

|  |  |
| --- | --- |
| **Visual SourceSafe** | |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Microsoft](https://en.wikipedia.org/wiki/Microsoft) |
| **Versión inicial** | 1994 ; Hace 24 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 2005 (8.0.50727.42) / octubre de 2005 ; Hace 12 años |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Software comercial](https://en.wikipedia.org/wiki/Commercial_software)[propietario](https://en.wikipedia.org/wiki/Proprietary_software) |
| **Sitio web** | [www .microsoft .com](http://www.microsoft.com/) |

Originalmente, SourceSafe no era una [gestión de código fuente](https://en.wikipedia.org/wiki/Source_Code_Management) cliente / servidor , sino más bien un sistema SCM *local* único. Arquitectónicamente, esto sirve como una fortaleza y una debilidad del diseño, dependiendo del entorno en el que se utiliza. Permite configurar un solo sistema de usuario con menos configuración que la de otros sistemas [SCM](https://en.wikipedia.org/wiki/Source_Code_Management) . Además, el proceso de copia de seguridad puede ser tan simple como copiar todos los contenidos de un único árbol de directorios. Para [entornos multiusuario](https://en.wikipedia.org/wiki/Multi-user) , sin embargo, carece de muchas características importantes que se encuentran en otros productos [SCM](https://en.wikipedia.org/wiki/Source_Code_Management) , incluido el soporte para [confirmaciones atómicas](https://en.wikipedia.org/wiki/Atomic_commit) de múltiples archivos ( [CVS](https://en.wikipedia.org/wiki/Concurrent_Versions_System) tiene el mismo problema que el original).[RCS](https://en.wikipedia.org/wiki/Revision_Control_System) ). SourceSafe hereda su funcionalidad *compartida* mediante el acceso directo al sistema de archivos remoto a todos los archivos en el repositorio. Esto, junto con errores históricos en la base de código, ocasionalmente condujo a la corrupción de la base de datos de SourceSafe, un problema observado por Microsoft. [[9]](https://en.wikipedia.org/wiki/Microsoft_Visual_SourceSafe#cite_note-9)

A partir de VSS 2005, Microsoft agregó un modo [cliente-servidor](https://en.wikipedia.org/wiki/Client%E2%80%93server) . En este modo, los clientes no necesitan [acceso de escritura](https://en.wikipedia.org/wiki/Write_access) a un [recurso](https://en.wikipedia.org/wiki/Server_Message_Block) compartido [SMB](https://en.wikipedia.org/wiki/Server_Message_Block) donde pueden dañar la base de datos SS. En su lugar, se debe acceder a los archivos a través de las herramientas del cliente de VSS: el cliente de Windows VSS, la herramienta de [línea de comandos de](https://en.wikipedia.org/wiki/Command-line)VSS o alguna aplicación que se integre o emule con estas herramientas de cliente.

1995 **Perforce Software** es un desarrollador de software basado en [Minneapolis, Minnesota,](https://en.wikipedia.org/wiki/Minneapolis,_Minnesota) utilizado para el desarrollo de aplicaciones, que incluye software de [control](https://en.wikipedia.org/wiki/Version_control) de versiones, gestión de repositorios basada en la web, colaboración de desarrolladores, [gestión del ciclo de vida de las aplicaciones](https://en.wikipedia.org/wiki/Application_lifecycle_management) y software de [planificación ágil](https://en.wikipedia.org/wiki/Agile_software_development) . El software se vende bajo las marcas Helix y Hansoft. [[2]](https://en.wikipedia.org/wiki/Perforce_Software#cite_note-pocketgamer-2)

En enero de 2018, la empresa fue adquirida por la firma de capital privado Clearlake Capital.

|  |  |
| --- | --- |
| **Perforce Software** | |
| [Perforce-Software-Logo.png](https://en.wikipedia.org/wiki/File:Perforce-Software-Logo.png) | |
| [**Tipo**](https://en.wikipedia.org/wiki/List_of_business_entities) | Privado |
| **Industria** | [El control de versiones](https://en.wikipedia.org/wiki/Version_control) de software, [gestión del ciclo de vida de aplicaciones](https://en.wikipedia.org/wiki/Application_lifecycle_management) , [Git](https://en.wikipedia.org/wiki/Git)administración de repositorios, herramientas de colaboración desarrollador y [la planificación ágil](https://en.wikipedia.org/wiki/Agile_software_development) de software |
| **Fundado** | 1995 |
| **Sede** | [Minneapolis, MN](https://en.wikipedia.org/wiki/Minneapolis,_MN) |
| **Área servida** | En todo el mundo |
| **Gente clave** | Janet Dryer, CEO |
| **Productos** | Helix Core, Helix TeamHub, Helix Swarm, suite Helix ALM, Hansoft |
| **Número de empleados** | 250+ (2014) [[1]](https://en.wikipedia.org/wiki/Perforce_Software#cite_note-ashimel-1) |
| [**Padre**](https://en.wikipedia.org/wiki/Parent_company) | Clearlake Capital |
| **Sitio web** | [www .perforce .com](http://www.perforce.com/) |

**1995 StarTeam** es un sistema de [control de revisiones](https://en.wikipedia.org/wiki/Revision_control) utilizado en el desarrollo de software, especialmente cuando un proyecto involucra múltiples equipos en diferentes ubicaciones. StarTeam es una aplicación de software [SCM](https://en.wikipedia.org/wiki/Source_Code_Management) y [SDLC](https://en.wikipedia.org/wiki/Software_development_lifecycle) , creada por Starbase Corporation, que fue adquirida por [Borland](https://en.wikipedia.org/wiki/Borland) en enero de 2003 [[1]](https://en.wikipedia.org/wiki/StarTeam#cite_note-1) que fue adquirida por [Micro Focus](https://en.wikipedia.org/wiki/Micro_Focus_International) mientras tanto. La aplicación es cliente-servidor, respaldada por una base de datos relacional que conserva todos los cambios realizados en un proyecto durante su evolución, así como los requisitos del proyecto, las asignaciones de tareas, las discusiones y el seguimiento de errores. [Microsoft SQL Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server) y la [base de datos Oracle](https://en.wikipedia.org/wiki/Oracle_database) son servidores de bases de datos compatibles.

|  |  |
| --- | --- |
| **Borland StarTeam** | |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | Base estelar, [Borland](https://en.wikipedia.org/wiki/Borland) |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Micro Focus International](https://en.wikipedia.org/wiki/Micro_Focus_International) |
| **Versión inicial** | 1995 ; Hace 23 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 16.2 / 20 de noviembre de 2017 ; Hace 2 meses |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Microsoft Windows 10](https://en.wikipedia.org/wiki/Microsoft_Windows_10) , [Microsoft Windows 8](https://en.wikipedia.org/wiki/Microsoft_Windows_8) ,[Microsoft Windows 7](https://en.wikipedia.org/wiki/Microsoft_Windows_7), [Windows XP Professional](https://en.wikipedia.org/wiki/Windows_XP_Professional) SP3,[Windows Vista](https://en.wikipedia.org/wiki/Windows_Vista) SP1,[Red Hat Enterprise Linux](https://en.wikipedia.org/wiki/Red_Hat_Enterprise_Linux) (WS) 5.1 |
| **Disponible en** | Plurilingüe |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [SCM](https://en.wikipedia.org/wiki/Software_configuration_management) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | Opciones de licencia de usuario final denominadas y concurrentes |
| **Sitio web** | [www .microfocus.com / products / change-management/ starteam /](http://www.microfocus.com/products/change-management/starteam/) |

1997 **Code Co-op** es el sistema de [control de revisión](https://en.wikipedia.org/wiki/Revision_control)[punto a punto](https://en.wikipedia.org/wiki/Peer-to-peer) creado por [Reliable Software](https://en.wikipedia.org/w/index.php?title=Reliable_Software&action=edit&redlink=1) .

Code Co-op es un sistema de [control de revisión](https://en.wikipedia.org/wiki/Revision_control) distribuido del tipo replicado.

Utiliza [la](https://en.wikipedia.org/wiki/Peer-to-peer) arquitectura [punto a punto](https://en.wikipedia.org/wiki/Peer-to-peer) para compartir proyectos entre desarrolladores y controlar los cambios en los archivos. En lugar de utilizar una [base de datos centralizada](https://en.wikipedia.org/wiki/Centralized_database) (el repositorio), replica su propia base de datos en cada computadora involucrada en el proyecto.

Las réplicas se sincronizan mediante el intercambio de *scripts* (diferenciales) . El intercambio de scripts puede continuar usando diferentes transportes, incluidos el correo electrónico (soporte para [SMTP](https://en.wikipedia.org/wiki/SMTP) y [POP3](https://en.wikipedia.org/wiki/POP3) , integración con clientes [MAPI](https://en.wikipedia.org/wiki/MAPI) , [Gmail](https://en.wikipedia.org/wiki/Gmail) ) y LAN.

Code Co-op tiene incorporado un sistema [wiki punto a punto](https://en.wikipedia.org/wiki/Peer-to-peer_wiki) , que se puede usar para integrar la documentación con un proyecto de software. También es posible crear [bases de datos Wiki](https://en.wikipedia.org/w/index.php?title=Wiki_database&action=edit&redlink=1) basadas en texto , que pueden consultarse utilizando [SQL](https://en.wikipedia.org/wiki/SQL) simplificado directamente desde páginas wiki.

|  |  |
| --- | --- |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | Software confiable |
| **Versión inicial** | 1996 ; Hace 22 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 5.2a / 4 de junio de 2011 ; Hace 6 años [[1]](https://en.wikipedia.org/wiki/Code_Co-op#cite_note-1) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) |
| **Disponible en** | [Inglés](https://en.wikipedia.org/wiki/English_language) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | Propiedad |
| **Sitio web** | [relisoft .com / co\_op /](http://relisoft.com/co_op/) |

1. Soporte de desarrollo distribuido a través [de correo electrónico](https://en.wikipedia.org/wiki/E-mail) , [LAN](https://en.wikipedia.org/wiki/Local_Area_Network) o [VPN](https://en.wikipedia.org/wiki/VPN)
2. Modificaciones basadas en el modelo: las modificaciones en varios archivos se registran como una [transacción](https://en.wikipedia.org/wiki/Database_transaction)
3. Las adiciones, eliminaciones, renombrados y movimientos de archivos se tratan en el mismo nivel que las ediciones; se pueden agregar en cualquier combinación a un conjunto de cambios de check-in.
4. Los cambios de archivos pueden revisarse antes de un check-in utilizando un built-in o definido por el usuario
5. Los cambios de sincronización pueden ser revisados ​​de la misma manera por los destinatarios
6. Fusión visual de tres vías
7. El historial del proyecto se replica en cada máquina. La versión histórica puede ser revisada, comparada o restaurada
8. Integración con clientes Microsoft SCC, incluido [Visual Studio](https://en.wikipedia.org/wiki/Visual_Studio)

1998 El **CVSNT** Sistema de control de versiones implementa un [sistema de control de versiones](https://en.wikipedia.org/wiki/Version_control_system) : se realiza un seguimiento de todos los cambios en un conjunto de archivos, por lo general la implementación de un [software de](https://en.wikipedia.org/wiki/Software)[proyecto](https://en.wikipedia.org/wiki/Project) , y permite que varios (potencialmente) separadas geográficamente a los desarrolladores [colaboran](https://en.wikipedia.org/wiki/Collaboration) . Es compatible y originalmente se basa en el [Sistema de Versiones Concurrentes](https://en.wikipedia.org/wiki/Concurrent_Versions_System) (CVS), que se ha hecho popular en el mundo de [código abierto](https://en.wikipedia.org/wiki/Open-source) .

CVSNT realiza un seguimiento del historial de versiones de un proyecto (o conjunto de archivos).

CVSNT se basa en la misma arquitectura [cliente-servidor](https://en.wikipedia.org/wiki/Client%E2%80%93server) que el [Sistema de Versiones Simultáneas](https://en.wikipedia.org/wiki/Concurrent_Versions_System) : un servidor almacena la (s) versión (es) actual (es) del proyecto y su historial, y los clientes se conectan al servidor para verificar una copia completa del proyecto. , trabaje en esta copia y luego registre sus cambios. Un servidor puede ser un servidor caché o proxy (un servidor de solo lectura que transfiere solicitudes de escritura a otro servidor) o un servidor de lectura y escritura (normal).

Normalmente, el cliente y el servidor se conectan a través de una [WAN](https://en.wikipedia.org/wiki/Wide_area_network) o [Internet](https://en.wikipedia.org/wiki/Internet) , pero tanto el cliente como el servidor pueden ejecutarse en la misma [LAN](https://en.wikipedia.org/wiki/Local_area_network) o incluso en la misma máquina. El software del servidor funciona igual de bien en [Unix](https://en.wikipedia.org/wiki/Unix) y [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) . Es compatible con los clientes de CVS y es compatible con una gran cantidad de metodologías de [gestión](https://en.wikipedia.org/wiki/Configuration_management) de [configuración,](https://en.wikipedia.org/wiki/Configuration_management) incluido el método sin reservas compatible con CVS y el método reservado similar a [VSS](https://en.wikipedia.org/wiki/Microsoft_Visual_SourceSafe) . [[2]](https://en.wikipedia.org/wiki/CVSNT#cite_note-2)[[3]](https://en.wikipedia.org/wiki/CVSNT#cite_note-3)

La naturaleza modular de CVSNT permite a los administradores del servidor habilitar y deshabilitar protocolos de forma dinámica, así como cargar funciones adicionales a través de activadores como Auditoría, Integración de seguimiento de defectos, Automatic Checkout (para servidores web) y más.

Las características del servidor CVSNT incluyen: [[4]](https://en.wikipedia.org/wiki/CVSNT#cite_note-4)

* Control de acceso para asegurar proyectos y sucursales. [[5]](https://en.wikipedia.org/wiki/CVSNT#cite_note-5)
* Auditoría detallada y métricas registradas en una base de datos [SQL](https://en.wikipedia.org/wiki/SQL) . [[6]](https://en.wikipedia.org/wiki/CVSNT#cite_note-6)[[7]](https://en.wikipedia.org/wiki/CVSNT#cite_note-7)[[8]](https://en.wikipedia.org/wiki/CVSNT#cite_note-8)
* Autenticación con [Active Directory](https://en.wikipedia.org/wiki/Active_Directory) . [[9]](https://en.wikipedia.org/wiki/CVSNT#cite_note-9)
* El seguimiento de todo el cambio, incluso si se fusionó desde algún lugar, pertenece a un informe de problema o formaba parte de un conjunto de cambios.
* Un panel de control para administrar la notificación por correo electrónico de los cambios, la integración de seguimiento de defectos y más.
* Sincronización de repositorio integrada (para servidores de conmutación por error).
* Cambiar soporte de ajuste (cambios de grupo por número de defecto).
* Admite archivos Unicode [UTF-8](https://en.wikipedia.org/wiki/UTF-8) / [UCS-2](https://en.wikipedia.org/wiki/UCS-2) y nombres de archivos multilingües.
* Cuando se opera en modo UTF-8 (Unicode), puede traducir automáticamente la codificación del nombre de archivo para cualquier cliente.
* Complementos para notificaciones por correo electrónico.
* Ayuda a facilitar la fusión de sucursales con su función "Mergepoint".
* Servidores nativos disponibles para Mac OS X, Windows, Linux, Solaris, HP-UX.
* Admite metodologías de versiones reservadas y sin reservas.
* CVSAPI para la integración en productos de terceros.
* Interfaz Script, COM y 3GL para desencadenantes e integración en herramientas de terceros (como seguimiento de defectos)

1998 **QVCS Enterprise** (Quma Version Control System) es una familia de productos de [sistemas de control de versiones](https://en.wikipedia.org/wiki/Version_control_system) históricamente publicados por Quma Software, Inc.

QVCS se publicó por primera vez en 1991 como un conjunto de utilidades de línea de comandos para [Amiga](https://en.wikipedia.org/wiki/Amiga) .

Quma llevó ese producto original de Amiga a la plataforma de [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) en 1996. En 2000, se introdujo QVCS-Pro con características adicionales, incluida la compatibilidad con [entornos de desarrollo integrados que](https://en.wikipedia.org/wiki/Integrated_development_environment)[cumplen](https://en.wikipedia.org/wiki/SCC_compliant) con la norma [SCC de](https://en.wikipedia.org/wiki/SCC_compliant)Microsoft . [[4]](https://en.wikipedia.org/wiki/QVCS#cite_note-4) En 2004, se introdujo el producto QVCS-Enterprise [multiplataforma](https://en.wikipedia.org/wiki/Cross-platform) . [[5]](https://en.wikipedia.org/wiki/QVCS#cite_note-5) QVCS y QVCS-Pro están escritos en [C ++](https://en.wikipedia.org/wiki/C%2B%2B) . QVCS-Enterprise está escrito en [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) .

El 1 de enero de 2012, QVCS-Pro y QVCS se convirtieron en software gratuito. QVCS-Enterprise hizo lo mismo el 16 de junio de 2012, y luego fue abierta bajo la licencia Apache 2.0 el 2 de marzo de 2014.

En algún momento después del 23 de junio de 2013, el nombre de dominio qumasoft.com fue abandonado. Ahora (junio de 2015) es propiedad de una compañía de pérdida de peso. El contenido del sitio web cuando fue operado por el autor de QVCS está disponible en [Wayback Machine](https://en.wikipedia.org/wiki/Wayback_Machine) .

|  |  |
| --- | --- |
| **QVCS Enterprise** | |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | Jim Voris |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | Software Quma |
| **Versión inicial** | 1998 ; hace 20 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 3.0.9 [[3]](https://en.wikipedia.org/wiki/QVCS#cite_note-3) / 14 de abril de 2014 ; hace 3 años |
| **Escrito en** | [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) , [OS X](https://en.wikipedia.org/wiki/OS_X) ,[Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Licencia Apache 2.0](https://en.wikipedia.org/wiki/Apache_2.0_License) |
| **Sitio web** | [github .com / jimv39 / qvcsos](https://github.com/jimv39/qvcsos) |

2000 **BitKeeper** es una herramienta de [software](https://en.wikipedia.org/wiki/Software) para [control](https://en.wikipedia.org/wiki/Distributed_revision_control) de [revisión distribuida](https://en.wikipedia.org/wiki/Distributed_revision_control) del [código fuente](https://en.wikipedia.org/wiki/Source_code) de la computadora . Originalmente, [el software propietario](https://en.wikipedia.org/wiki/Proprietary_software) , fue lanzado como [software de código abierto](https://en.wikipedia.org/wiki/Open-source_software) bajo la [licencia Apache 2.0](https://en.wikipedia.org/wiki/Apache_License#Version_2.0) el 9 de mayo de 2016. [[2]](https://en.wikipedia.org/wiki/BitKeeper#cite_note-2) BitKeeper es producido por BitMover Inc., una empresa privada con sede en [Los Gatos, California](https://en.wikipedia.org/wiki/Los_Gatos,_California)[[3]](https://en.wikipedia.org/wiki/BitKeeper#cite_note-3) y propiedad de su [CEO](https://en.wikipedia.org/wiki/CEO) , [Larry McVoy](https://en.wikipedia.org/wiki/Larry_McVoy) , quien previamente había diseñado [TeamWare](https://en.wikipedia.org/wiki/TeamWare) .

|  |  |
| --- | --- |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | BitMover Inc. |
| **Versión inicial** | 4 de mayo de 2000 ; Hace 17 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 7.3.2 / 23 de septiembre de 2017 ; Hace 3 meses [[1]](https://en.wikipedia.org/wiki/BitKeeper#cite_note-1) |
| [**Repositorio**](https://en.wikipedia.org/wiki/Repository_(version_control)) | * <http://repos.bkbits.net/bk/dev/>   [Edita esto en Wikidata](https://www.wikidata.org/wiki/Q878697#P1324) |
| **Escrito en** | [do](https://en.wikipedia.org/wiki/C_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | AIX, FreeBSD, HP-UX, IRIX, Linux, Mac OS X, NetBSD, OpenBSD, Solaris, Windows |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión distribuido](https://en.wikipedia.org/wiki/Distributed_revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License#Version_2.0) |
| **Sitio web** | [www .bitkeeper .org](http://www.bitkeeper.org/) |

BitKeeper se mencionó por primera vez como una solución a algunos de los dolores de crecimiento que Linux estaba teniendo en septiembre de 1998. [[4]](https://en.wikipedia.org/wiki/BitKeeper#cite_note-4) Betas de acceso temprano estaban disponibles en mayo de 1999 [[5]](https://en.wikipedia.org/wiki/BitKeeper#cite_note-5) y el 4 de mayo de 2000 se puso a disposición el primer lanzamiento público de BitKeeper . [[6]](https://en.wikipedia.org/wiki/BitKeeper#cite_note-6)[[7]](https://en.wikipedia.org/wiki/BitKeeper#cite_note-7) BitMover solía proporcionar acceso al sistema para ciertos [proyectos de código abierto](https://en.wikipedia.org/wiki/Open_source) o [software libre](https://en.wikipedia.org/wiki/Free_software) , uno de los cuales era el código fuente del [kernel de Linux](https://en.wikipedia.org/wiki/Linux_kernel) . La licencia para la versión "comunitaria" de BitKeeper había permitido a los desarrolladores utilizar la herramienta sin costo alguno para proyectos de software libre o de código abierto, siempre que esos desarrolladores no participaran en el desarrollo de una herramienta competidora (como[Concurrent Versions System](https://en.wikipedia.org/wiki/Concurrent_Versions_System) , [GNU Arch](https://en.wikipedia.org/wiki/GNU_Arch) , [Subversion](https://en.wikipedia.org/wiki/Subversion_(software)) o [ClearCase](https://en.wikipedia.org/wiki/ClearCase) ) durante el uso de BitKeeper más un año. Esta restricción se aplicó independientemente de si la herramienta competidora era gratuita o patentada. Esta versión de BitKeeper también requería que cierta metainformación sobre cambios se almacenara en servidores de computadora operados por BitMover, una adición que imposibilitaba a los usuarios de la versión de la comunidad ejecutar proyectos de los cuales BitMover no era consciente.

2000 **Apache Subversion** (a menudo abreviado **SVN** , después de su nombre de comando *svn* ) es un [sistema de control de](https://en.wikipedia.org/wiki/Software_versioning)[revisiones](https://en.wikipedia.org/wiki/Revision_control) y [versiones de software](https://en.wikipedia.org/wiki/Software_versioning) distribuido como [código abierto](https://en.wikipedia.org/wiki/Open_source) bajo la [licencia de Apache](https://en.wikipedia.org/wiki/Apache_License) . [[2]](https://en.wikipedia.org/wiki/Apache_Subversion#cite_note-2) Los desarrolladores de software usan Subversion para mantener versiones actuales e históricas de archivos tales como [código fuente](https://en.wikipedia.org/wiki/Source_code) , páginas web y documentación. Su objetivo es ser un sucesor casi compatible del ampliamente utilizado [Sistema de Versiones Concurrentes](https://en.wikipedia.org/wiki/Concurrent_Versions_System) (CVS).

La comunidad de [código abierto](https://en.wikipedia.org/wiki/Open_source) ha utilizado ampliamente Subversion: por ejemplo, en proyectos como [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation) , [Free Pascal](https://en.wikipedia.org/wiki/Free_Pascal) , [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD) , [GCC](https://en.wikipedia.org/wiki/GNU_Compiler_Collection) y [SourceForge](https://en.wikipedia.org/wiki/SourceForge) . [CodePlex](https://en.wikipedia.org/wiki/CodePlex) ofrece acceso a Subversion, así como a otros tipos de clientes.

Subversion fue creada por [CollabNet](https://en.wikipedia.org/wiki/CollabNet) Inc. en 2000, y ahora es un proyecto de Apache de alto nivel que está siendo construido y utilizado por una comunidad global de colaboradores.

|  |  |
| --- | --- |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | [CollabNet](https://en.wikipedia.org/wiki/CollabNet) |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Apache Software Foundation](https://en.wikipedia.org/wiki/Apache_Software_Foundation) |
| **Versión inicial** | 20 de octubre de 2000 ; Hace 17 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 1.9.7 (9 de agosto de 2017 ; hace 5 meses [[1]](https://en.wikipedia.org/wiki/Apache_Subversion#cite_note-1) ) [[±]](https://en.wikipedia.org/w/index.php?title=Template:Latest_stable_software_release/Subversion&action=edit) |
| [**Lanzamiento de vista previa**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 1.10.0-alpha3 (26 de julio de 2017 ; 6 meses atrás ) [[±]](https://en.wikipedia.org/w/index.php?title=Template:Latest_preview_software_release/Subversion&action=edit) |
| [**Repositorio**](https://en.wikipedia.org/wiki/Repository_(version_control)) | * <https://svn.apache.org/viewvc/subversion/trunk/>   [Edita esto en Wikidata](https://www.wikidata.org/wiki/Q46794#P1324) |
| **Estado del desarrollo** | Activo |
| **Escrito en** | [do](https://en.wikipedia.org/wiki/C_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Multiplataforma](https://en.wikipedia.org/wiki/Cross-platform) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Licencia Apache](https://en.wikipedia.org/wiki/Apache_License) |
| **Sitio web** | [subversion .apache .org](https://subversion.apache.org/) |

* [Se compromete](https://en.wikipedia.org/wiki/Commit_(data_management)) como verdaderas [operaciones atómicas](https://en.wikipedia.org/wiki/Atomicity_(database_systems)) (las [operaciones de](https://en.wikipedia.org/wiki/Atomicity_(database_systems)) confirmación interrumpidas en CVS causarían incoherencia o corrupción del repositorio).
* Los archivos renombrados / copiados / movidos / eliminados retienen el historial completo de revisiones.
* El sistema mantiene el [control](https://en.wikipedia.org/wiki/Software_versioning) de [versiones](https://en.wikipedia.org/wiki/Software_versioning) para directorios, cambios de nombre y [metadatos de](https://en.wikipedia.org/wiki/Metadata) archivos (pero no para marcas de tiempo). Los usuarios pueden mover y / o copiar árboles de directorios completos muy rápidamente, conservando el historial completo de revisiones.
* Control de versiones de [enlaces simbólicos](https://en.wikipedia.org/wiki/Symbolic_link) .
* Soporte nativo para archivos binarios, con almacenamiento binario-diff con ahorro de espacio.
* [Apache HTTP Server](https://en.wikipedia.org/wiki/Apache_HTTP_Server) como servidor de red, [WebDAV](https://en.wikipedia.org/wiki/WebDAV) / [Delta-V](https://en.wikipedia.org/wiki/WebDAV#Extensions_and_derivatives) para [protocolo](https://en.wikipedia.org/wiki/Protocol_(computing)) . También hay un [proceso de](https://en.wikipedia.org/wiki/Process_(computing)) servidor independiente llamado svnserve que usa un protocolo personalizado sobre [TCP / IP](https://en.wikipedia.org/wiki/Internet_Protocol_Suite) .
* [La ramificación](https://en.wikipedia.org/wiki/Branching_(software)) es una operación económica, independiente del tamaño del archivo (aunque Subversion por sí misma no distingue entre una rama y un directorio)
* Nativamente [cliente-servidor](https://en.wikipedia.org/wiki/Client%E2%80%93server_model) , diseño de [biblioteca en](https://en.wikipedia.org/wiki/Library_(computing))[capas](https://en.wikipedia.org/wiki/Abstraction_layer) .
* El protocolo cliente / servidor envía [diffs](https://en.wikipedia.org/wiki/Diff) en ambas direcciones.
* Costos proporcionales al tamaño del cambio, no al tamaño de los datos.
* Salida [procesable](https://en.wikipedia.org/wiki/Parsing) , incluida la salida de registro [XML](https://en.wikipedia.org/wiki/XML) .
* licencia de [fuente abierta](https://en.wikipedia.org/wiki/Open_source) - [Apache License](https://en.wikipedia.org/wiki/Apache_License) desde la versión 1.7; las versiones anteriores usan una derivada de la Licencia de Software Apache 1.1.
* Mensajes de programas [internacionalizados](https://en.wikipedia.org/wiki/Internationalization_and_localization) .
* [Bloqueo](https://en.wikipedia.org/wiki/File_locking) de archivos para archivos no fusionables ("registros reservados").
* Autorización basada en ruta.
* [Enlaces de lenguaje](https://en.wikipedia.org/wiki/Language_binding) para [C #](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)) , [PHP](https://en.wikipedia.org/wiki/PHP) , [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) , [Perl](https://en.wikipedia.org/wiki/Perl) , [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)) y [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) .
* Soporte completo [MIME](https://en.wikipedia.org/wiki/MIME) : los usuarios pueden ver o cambiar el tipo MIME de cada archivo, y el software sabe qué tipos MIME pueden mostrar diferencias con respecto a las versiones anteriores.
* Seguimiento de fusión: se realizará un seguimiento de las fusiones entre sucursales, esto permite la fusión automática entre las sucursales sin decirle a Subversion qué necesita y qué no necesita fusionarse.
* Listas de cambios para organizar confirmaciones en grupos de compromiso.

2001 **PTC Integrity Lifecycle Manager** (anteriormente *MKS Integrity* ) es una plataforma de administración del ciclo de vida del sistema de software (SSLM) y de [administración del ciclo de vida de la aplicación](https://en.wikipedia.org/wiki/Application_lifecycle_management) (ALM) desarrollada por [MKS Inc.](https://en.wikipedia.org/wiki/MKS_Inc.) y lanzada por primera vez en 2001. El software es cliente / servidor, con ambas computadoras de escritorio ( [Java / swing](https://en.wikipedia.org/wiki/Swing_(java)) ) y las interfaces de cliente web. Proporciona a las organizaciones de desarrollo de software un entorno colaborativo en el que pueden gestionar los procesos de desarrollo de principio a fin, desde la [gestión de requisitos](https://en.wikipedia.org/wiki/Requirements_management) , la [gestión de cambios de ingeniería](https://en.wikipedia.org/wiki/Change_management_(engineering)) , el [control de revisiones](https://en.wikipedia.org/wiki/Revision_control) y la [gestión de compilación](https://en.wikipedia.org/wiki/Build_management) hasta la [gestión](https://en.wikipedia.org/wiki/Build_management) de [pruebas](https://en.wikipedia.org/wiki/Test_management) y[implementación de software](https://en.wikipedia.org/wiki/Software_deployment) , así como informes y métricas asociados.

|  |  |
| --- | --- |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [PTC, Inc.](https://en.wikipedia.org/wiki/PTC,_Inc.) |
| **Versión inicial** | Julio de 2001 ; Hace 16 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 11.1 / 30 de marzo de 2017 ; hace 9 meses |
| **Estado del desarrollo** | Activo |
| **Escrito en** | [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Unix](https://en.wikipedia.org/wiki/Unix-like) , [Windows](https://en.wikipedia.org/wiki/Windows) |
| **Disponible en** | Idiomas internacionales a través de IAN, localizaciones para inglés, alemán, japonés. |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [ALM](https://en.wikipedia.org/wiki/Application_Lifecycle_Management) , [RM](https://en.wikipedia.org/wiki/Requirements_Management) , [SCM](https://en.wikipedia.org/wiki/Software_Configuration_Management) ,[Calidad](https://en.wikipedia.org/wiki/Quality_Management) , [Agile](https://en.wikipedia.org/wiki/Agile_software_development) ,[SysEng](https://en.wikipedia.org/wiki/Systems_Engineering) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | PTC [EULA](https://en.wikipedia.org/wiki/Software_license) |
| **Sitio web** | [www .ptc .com / application-lifecycle-management](http://www.ptc.com/application-lifecycle-management) |

MKS Integrity ahora es un producto de [PTC](https://en.wikipedia.org/wiki/Parametric_Technology_Corporation) desde la adquisición de MKS Inc. que se completó el 31 de mayo de 2011 por PTC. [[1]](https://en.wikipedia.org/wiki/PTC_Integrity#cite_note-1)

PTC Integrity Lifecycle Manager (Integrity LM o ILM) permite a los equipos de desarrollo de software realizar un seguimiento de todos los aspectos de su trabajo, incluidos los elementos de trabajo, el control de origen, los informes y la administración de compilación, en un solo producto. El producto consta de dos componentes: Integrity Configuration Management e Integrity Workflow & Documents. La parte de Configuration Management de PTC ILM se usa para manejar versiones de código fuente, ramas, etc. Se basa en la arquitectura cliente-servidor. El cliente de Thick Java no almacena ningún dato de gestión en el sistema local, por lo tanto, cualquier tarea realizada en los archivos fuente requiere una conexión de red. Esto significa que, a diferencia de [los sistemas distribuidos](https://en.wikipedia.org/wiki/Distributed_revision_control), este sistema requiere una conexión de red confiable, suficiente ancho de banda de red y suficiente poder de procesamiento en el lado del servidor. El otro componente (Workflow y documentos) consiste en un [sistema de seguimiento de problemas](https://en.wikipedia.org/wiki/Issue_tracking_system) , así como una solución de gestión de requisitos y requisitos.

Uno de los puntos fuertes en comparación con otras soluciones similares es la flexibilidad de PTC Integrity en términos de flujo de trabajo, campos, diseño de presentación, validación y capacidades de automatización. PTC Integrity Lifecycle Manager se basa en Java y utiliza una extensión de JavaScript para la creación de informes. Cualquier interacción se puede realizar en línea, en la CLI (Interfaz de línea de comando) o utilizando la API de Java del servidor o del cliente.

PTC Integrity Lifecycle Manager se basa en un único repositorio. Esta solución de repositorio único admite los tres pilares de la gestión del ciclo de vida: trazabilidad, automatización de procesos, informes y análisis [[2]](https://en.wikipedia.org/wiki/PTC_Integrity#cite_note-2) y algunas empresas pueden ver un valor adicional en este enfoque. [[3]](https://en.wikipedia.org/wiki/PTC_Integrity#cite_note-3)

La integración de PTC Integrity Lifecycle Manager con [IDEs](https://en.wikipedia.org/wiki/Integrated_development_environment) y otras herramientas de desarrollo está, de fábrica, limitada a pocos productos. Los IDE admitidos incluyen [Eclipse](https://en.wikipedia.org/wiki/Eclipse_(software))[[4]](https://en.wikipedia.org/wiki/PTC_Integrity#cite_note-4) y [Visual Studio](https://en.wikipedia.org/wiki/Visual_Studio) . [[5]](https://en.wikipedia.org/wiki/PTC_Integrity#cite_note-5) También son compatibles [IBM i](https://en.wikipedia.org/wiki/IBM_i) y [Apache Maven](https://en.wikipedia.org/wiki/Apache_Maven) .

2001 **El**[software](https://en.wikipedia.org/wiki/Software)**GNU arch** es un sistema de [control de revisión distribuido](https://en.wikipedia.org/wiki/Distributed_revision_control) que es parte del [Proyecto GNU](https://en.wikipedia.org/wiki/GNU_Project) y está licenciado bajo la [Licencia Pública General de GNU](https://en.wikipedia.org/wiki/GNU_General_Public_License) . Se utiliza para realizar un seguimiento de los cambios realizados en un árbol fuente y para ayudar a los programadores a combinar y manipular los cambios realizados por varias personas o en diferentes momentos.

A partir de 2009, el estado oficial del arco de GNU es la desactivación, y solo se aplican correcciones de seguridad. [[1]](https://en.wikipedia.org/wiki/GNU_arch#cite_note-arch-status-2008-1) [Bazaar (o 'bzr')](https://en.wikipedia.org/wiki/GNU_bzr)también se ha convertido en un proyecto oficial de GNU y, por lo tanto, puede considerarse el reemplazo del arco de GNU. No es un tenedor de Arch.

|  |  |
| --- | --- |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | Thomas Lord |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | Andy Tai |
| [**Último lanzamiento**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 1.3.5 / 20 de julio de 2006 ; Hace 11 años |
| **Estado del desarrollo** | Solo correcciones de seguridad [[1]](https://en.wikipedia.org/wiki/GNU_arch#cite_note-arch-status-2008-1) |
| **Escrito en** | [do](https://en.wikipedia.org/wiki/C_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [GNU](https://en.wikipedia.org/wiki/GNU) / [Linux](https://en.wikipedia.org/wiki/Linux) ,[Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) , [Mac OS X](https://en.wikipedia.org/wiki/Mac_OS_X) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [GPL](https://en.wikipedia.org/wiki/GNU_General_Public_License) |
| **Sitio web** | [www .gnu .org/ software / gnu-arch /](https://www.gnu.org/software/gnu-arch/) |

Al tratarse de un sistema de control de versiones distribuido y descentralizado, cada revisión almacenada utilizando arch es única e identificable globalmente; dicho identificador se puede usar en una configuración [distribuida](https://en.wikipedia.org/wiki/Distributed_system) para fusionar fácilmente o "seleccionar cuidadosamente" cambios de fuentes completamente dispares.

Estar descentralizado significa que no hay necesidad de un servidor central para el cual los desarrolladores tengan que ser autorizados para contribuir. Al igual que con otros sistemas, se puede acceder a una copia completa de solo lectura de un proyecto en un repositorio "oficial" a través de [HTTP](https://en.wikipedia.org/wiki/HTTP) , [FTP](https://en.wikipedia.org/wiki/File_Transfer_Protocol) o [SFTP](https://en.wikipedia.org/wiki/SSH_file_transfer_protocol) ; pero luego, se alienta a los contribuyentes a realizar modificaciones y publicarlas en un archivo público (repositorio) propio, de modo que el desarrollador principal pueda fusionar manualmente los conjuntos de cambios en el repositorio oficial.

Para simular el comportamiento de los sistemas de control de revisiones centralizados, el desarrollador principal podría permitir el acceso al shell ( [SSH](https://en.wikipedia.org/wiki/Secure_Shell) ) o el acceso de escritura (FTP, SFTP, [WebDAV](https://en.wikipedia.org/wiki/WebDAV) ) a un servidor, permitiendo a los usuarios autorizados comprometerse con un servidor central. Más a menudo, los proyectos archivados de GNU tienen un [dictador benévolo](https://en.wikipedia.org/wiki/Benevolent_dictator) líder que fusiona los cambios con los contribuyentes.

GNU arch tiene otras muchas características:

[**Compromisos atómicos**](https://en.wikipedia.org/wiki/Atomic_commit)

Los compromisos son todo o nada. El árbol debe estar en buenas condiciones antes de que comience el compromiso, y las confirmaciones no son visibles para el mundo hasta que se completen. Si la confirmación se interrumpe antes de esto, permanece invisible y debe revertirse antes de la siguiente confirmación. Esto evita la corrupción del archivo y las copias extraídas de otros usuarios.

**Orientado a los cambios**

En lugar de rastrear archivos individuales (como en [CVS](https://en.wikipedia.org/wiki/Concurrent_Versions_System) ), GNU arch rastrea los conjuntos de cambios, que son similares a los parches. Cada conjunto de cambios es una descripción de la diferencia entre un árbol fuente y otro, por lo que un conjunto de cambios puede usarse para producir una revisión de otra revisión. Se alienta a los autores a usar una confirmación por función o corrección de errores.

**Fácil ramificación**

La ramificación es eficiente y puede abarcar archivos. Una rama (o 'etiqueta') simplemente declara la revisión del ancestro, y el desarrollo continúa desde allí.

**Fusión avanzada**

Debido al registro permanente de todos los antepasados ​​y las revisiones fusionadas, la fusión puede tener en cuenta qué rama contiene qué parche y puede realizar una combinación de tres vías basada en una revisión de antepasados ​​compartida.

[**Firmas criptográficas**](https://en.wikipedia.org/wiki/Cryptographic_signature)

Cada conjunto de cambios se almacena con un [hash](https://en.wikipedia.org/wiki/Cryptographic_hash_function) para evitar daños accidentales. Al usar un programa externo de firma de archivos (como [GnuPG](https://en.wikipedia.org/wiki/GnuPG) u otro cliente [PGP](https://en.wikipedia.org/wiki/Pretty_Good_Privacy) ), estos valores hash también se pueden firmar opcionalmente, evitando modificaciones no autorizadas si el archivo está en peligro.

**Renombrar**

Todos los archivos y directorios pueden renombrarse fácilmente. Estos se rastrean mediante un ID único en lugar de por su nombre, por lo que se conserva el historial y los parches de los archivos se fusionan correctamente, incluso si los nombres de archivo son diferentes en las sucursales.

**Seguimiento de metadatos**

Los [permisos](https://en.wikipedia.org/wiki/File_system_permissions) de todos los archivos son rastreados. [Los enlaces simbólicos](https://en.wikipedia.org/wiki/Symbolic_link) son compatibles y son rastreados de la misma manera que los archivos y directorios.

2002 **Darcs** es un sistema de [control de versiones distribuidas](https://en.wikipedia.org/wiki/Distributed_version_control) creado por [David Roundy](https://en.wikipedia.org/wiki/David_Roundy) . Las características clave incluyen la capacidad de elegir qué cambios aceptar de otros repositorios, la interacción con otros repositorios locales (en disco) o repositorios remotos a través de [SSH](https://en.wikipedia.org/wiki/Secure_Shell) , [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) o correo electrónico, y una interfaz inusualmente interactiva. Los desarrolladores también enfatizan el uso de herramientas de software avanzadas para verificar la corrección: el [sistema](https://en.wikipedia.org/wiki/Type_system) de [tipo](https://en.wikipedia.org/wiki/Type_system) expresivo del [lenguaje de programación funcional](https://en.wikipedia.org/wiki/Functional_programming)[Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)) impone algunas propiedades, y las pruebas aleatorias a través de [QuickCheck](https://en.wikipedia.org/wiki/QuickCheck) verifican muchas otras. [[2]](https://en.wikipedia.org/wiki/Darcs#cite_note-FOOTNOTERoundy20052-2) El nombre es un [acrónimo recursivo](https://en.wikipedia.org/wiki/Recursive_acronym)para el **Sistema de Control de Revisión Avanzado de Darcs** .

|  |  |
| --- | --- |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | [David Roundy](https://en.wikipedia.org/wiki/David_Roundy) |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | Guillaume Hoffmann, y col. |
| **Versión inicial** | 3 de marzo de 2003 ; Hace 14 años [[1]](https://en.wikipedia.org/wiki/Darcs#cite_note-darcs-changelog-1) |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 2.12.5 / 11 de enero de 2017 ; Hace 12 meses [[1]](https://en.wikipedia.org/wiki/Darcs#cite_note-darcs-changelog-1) |
| [**Repositorio**](https://en.wikipedia.org/wiki/Repository_(version_control)) | * <http://hub.darcs.net/darcs/darcs-reviewed>   [Edita esto en Wikidata](https://www.wikidata.org/wiki/Q204377#P1324) |
| **Escrito en** | [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Unix](https://en.wikipedia.org/wiki/Unix) , [Linux](https://en.wikipedia.org/wiki/Linux) , [BSD](https://en.wikipedia.org/wiki/BSD) , [Apple](https://en.wikipedia.org/wiki/Apple,_Inc.)[macOS](https://en.wikipedia.org/wiki/MacOS) , [MS](https://en.wikipedia.org/wiki/Microsoft)[Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de versiones](https://en.wikipedia.org/wiki/Version_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [GNU](https://en.wikipedia.org/wiki/GNU)[GPL](https://en.wikipedia.org/wiki/GPL_license) |
| **Sitio web** | [darcs .net](http://darcs.net/) |

Darcs trata parches como [ciudadanos de primera clase](https://en.wikipedia.org/wiki/First-class_citizen) . Para el usuario, un repositorio puede verse como un conjunto de parches, donde cada parche no está necesariamente ordenado con respecto a otros parches, es decir, el conjunto de parches es solo un [conjunto parcialmente ordenado](https://en.wikipedia.org/wiki/Partially_ordered_set) . En muchos casos, los parches se pueden transmitir de forma independiente entre varios repositorios.

Muchas operaciones de bifurcación, [fusión](https://en.wikipedia.org/wiki/Merge_(version_control)) y selección que requerirían comandos adicionales con sistemas basados ​​en instantáneas como [Git](https://en.wikipedia.org/wiki/Git_(software)) o [Mercurial se](https://en.wikipedia.org/wiki/Mercurial) pueden hacer directamente con Darcs con los comandos habituales "extraer" y "presionar". En términos de interfaz de usuario, esto significa que Darcs tiene menos comandos. Estos comandos son más interactivos: uno puede elegir con más precisión qué parches quieren intercambiar con repositorios remotos.

Los parches de un repositorio están ordenados linealmente. Darcs calcula automáticamente si los parches se pueden reordenar (una operación llamada conmutación) y cómo hacerlo. Estos cálculos implementan una llamada "teoría de parches".

Un parche de Darcs puede contener cambios de los siguientes tipos:

* cambios de línea,
* creación y eliminación de archivos y directorios,
* archivo y directorio en movimiento,
* sustitución de palabras (normalmente utilizada en la refactorización de códigos, por ejemplo, cambiar el nombre de todas las apariciones de "foo" a "bar" en un archivo determinado).

La noción de dependencia entre parches se define sintácticamente. Intuitivamente, un parche *B* depende de otro parche *A* si *A* proporciona el contenido que *B* modifica. Esto significa que los parches que modifican diferentes partes del código se consideran, por defecto, independientes. Para abordar casos cuando esto no es deseable, Darcs permite al usuario especificar dependencias explícitas entre parches.

Desde la versión 2.10, Darcs utiliza la [paciencia diff](https://en.wikipedia.org/w/index.php?title=Patience_diff&action=edit&redlink=1) por defecto.

2002 **Distributed Concurrent Versions System** (**DCVS**)  ( **DCVS** ) es un sistema de [control de revisiones distribuidas](https://en.wikipedia.org/wiki/Distributed_revision_control) que permite a los desarrolladores de software que trabajan en sitios distribuidos localmente colaborar de manera eficiente en un proyecto de software. DCVS se basa en el conocido sistema de control de [*versiones Sistema de versiones concurrentes*](https://en.wikipedia.org/wiki/Concurrent_Versions_System) . El código es libremente distribuible bajo las licencias de estilo GNU y BSD.

|  |  |
| --- | --- |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | elego Software Solutions GmbH |
| **Versión inicial** | Agosto de 2002 ; hace 15 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 1.0.3 / 25 de septiembre de 2006 ; Hace 11 años |
| **Estado del desarrollo** | Sin mantenimiento |
| **Escrito en** | [do](https://en.wikipedia.org/wiki/C_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Unix](https://en.wikipedia.org/wiki/Unix-like) , [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Licencia pública general de GNU](https://en.wikipedia.org/wiki/GNU_General_Public_License) |
| **Sitio web** | [dcvs .elegosoft .com](http://dcvs.elegosoft.com/) |

DCVS proporciona toda la funcionalidad de CVS. Pero a diferencia de CVS, un sistema DCVS puede comprender un número arbitrario de repositorios distribuidos geográficamente cuyos contenidos se mantienen iguales en segundo plano por una versión extendida de [*CVSup*](https://en.wikipedia.org/w/index.php?title=CVSup&action=edit&redlink=1) , un programa desarrollado por John D. Polstra. La combinación del *repositorio DCVS* , el *servidor CVSup* extendido y el *programa del servidor DCVS* se denominará *servidor DCVS* en los párrafos siguientes.

Todos los contenidos de todas las líneas de desarrollo se pueden *extraer* de cualquiera de los servidores DCVS en un *espacio de trabajo DCVS* propiedad de un desarrollador. Todas las operaciones que no modifican el repositorio, como [*diff*](https://en.wikipedia.org/wiki/Diff) , [*patch*](https://en.wikipedia.org/wiki/Patch_(Unix)) , *log* , *annotate*etc., funcionan igual que en CVS, pero siempre usan el repositorio local y, por lo tanto, son mucho más rápidos en un escenario distribuido. Para evitar colisiones y pérdida de datos, a cada servidor DCVS se le asigna un conjunto de líneas de desarrollo (ramas DCVS) de las que es responsable. Las modificaciones en una sucursal solo pueden registrarse en el servidor responsable de la sucursal. La separación de modificaciones por líneas de desarrollo hace posible transferir y distribuir automáticamente los cambios en la red DCVS. Una persona que desee realizar cambios para una línea de desarrollo de la que su servidor DCVS local no es responsable puede crear una nueva línea de desarrollo (sucursal) y comprometer los cambios a la misma. El servidor local es automáticamente responsable de cualquier línea de desarrollo recién creada.

Por otro lado, cada desarrollador puede fusionar los cambios de las líneas de desarrollo que su servidor DCVS local no es responsable de las líneas de desarrollo locales. Por lo tanto, todos los cambios realizados en cualquier sitio de trabajo pueden ser aplicados a la rama original por los desarrolladores en el servidor DCVS responsable que realiza una operación de [*fusión*](https://en.wikipedia.org/wiki/Merge_(revision_control)) .

La funcionalidad de los [*conjuntos*](https://en.wikipedia.org/wiki/Changeset) de [*cambios*](https://en.wikipedia.org/wiki/Changeset) permite a los desarrolladores producir pequeños conjuntos de cambios relacionados con una característica o un defecto, que luego pueden ser aplicados por otros.

Un esquema de numeración específico garantiza que las líneas de desarrollo y los deltas se puedan identificar como pertenecientes a un determinado servidor DCVS. DCVS asigna un [*rango*](https://en.wikipedia.org/wiki/Range_(statistics)) único de números de ramificación a cada par (servidor / colección). Todos los rangos para todos los servidores y colecciones deben ser mutuamente exclusivos. Las definiciones de servidores, colecciones y rangos se leen desde un único archivo de configuración. Al consultar el contenido de este archivo, cada servidor DCVS puede decidir si es responsable de una determinada rama o delta de un archivo determinado. Si es así, se permiten todas las operaciones de modificación; si no, las operaciones de modificación solo son posibles en el servidor remoto apropiado.

Además, los nombres reales de las configuraciones, las [*etiquetas*](https://en.wikipedia.org/wiki/Tag_(metadata)) , se asignan de manera única a exactamente un servidor DCVS por medio de un sufijo específico del servidor que extiende todas las *etiquetas* (es decir, \_at\_dcvs\_mydomain\_org). Por lo tanto, no pueden surgir conflictos en el espacio del nombre de la etiqueta.

2002 **Surround SCM** es una aplicación de [administración de configuración de software](https://en.wikipedia.org/wiki/Software_configuration_management) desarrollada por [Seapine Software](https://en.wikipedia.org/wiki/Seapine_Software) . En 2008, Surround SCM ganó un [Premio Jolt](https://en.wikipedia.org/wiki/Jolt_Awards) en la categoría Gestión de cambios y configuración.

|  |  |
| --- | --- |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Seapine Software](https://en.wikipedia.org/wiki/Seapine_Software) |
| **Versión inicial** | Septiembre de 2002 ; hace 15 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 2017.1.0 / 24 de abril de 2017 ; hace 8 meses |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) ,[Linux](https://en.wikipedia.org/wiki/Linux) , [Mac OS X](https://en.wikipedia.org/wiki/Mac_OS_X) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Propiedad](https://en.wikipedia.org/wiki/Proprietary_software) |
| **Sitio web** | [www .seapine .com](http://www.seapine.com/) |

2003

**Vault** es un comercial, propietaria [del sistema de control de versiones](https://en.wikipedia.org/wiki/Version_control_system) por **SourceGear LLC** , que comercializa su producto como un reemplazo para [Microsoft](https://en.wikipedia.org/wiki/Microsoft) 's [Visual Source Safe](https://en.wikipedia.org/wiki/Visual_Source_Safe) .

Vault utiliza [Microsoft SQL Server](https://en.wikipedia.org/wiki/Microsoft_SQL_Server) como una base de datos back-end y proporciona confirmaciones atómicas para el sistema de control de versiones.

La herramienta está construida sobre [Microsoft .NET](https://en.wikipedia.org/wiki/Microsoft_.NET) .

Fortress, originalmente un [producto de gestión del ciclo de vida de las aplicaciones](https://en.wikipedia.org/wiki/Application_lifecycle_management) (ALM) comercializado por separado para su uso con Vault, se fusionó más tarde en versiones de Vault. [[3]](https://en.wikipedia.org/wiki/Vault_(version_control_system)#cite_note-3)

Los productos de terceros se han diseñado para integrarse con Vault, como [OnTime](https://en.wikipedia.org/wiki/OnTime) , [[4]](https://en.wikipedia.org/wiki/Vault_(version_control_system)#cite_note-4) [FogBugz](https://en.wikipedia.org/wiki/FogBugz) , [[5]](https://en.wikipedia.org/wiki/Vault_(version_control_system)#cite_note-5) [TeamCity](https://en.wikipedia.org/wiki/TeamCity) , [[6]](https://en.wikipedia.org/wiki/Vault_(version_control_system)#cite_note-6) y [SmartBear](https://en.wikipedia.org/wiki/SmartBear_Software)CodeCollaborator.

|  |  |
| --- | --- |
| **Bóveda** | |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | SourceGear LLC |
| **Versión inicial** | 2003 ; hace 15 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 7.2 [[1]](https://en.wikipedia.org/wiki/Vault_(version_control_system)#cite_note-1)[[2]](https://en.wikipedia.org/wiki/Vault_(version_control_system)#cite_note-2) / 28 de mayo de 2014 ; hace 3 años |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | Windows XP, Windows Server 2003 (solo cliente), Vista, Windows 7, 8, Windows Server 2008, 2008R2, 2012 |
| [**Plataforma**](https://en.wikipedia.org/wiki/Computing_platform) | [CLI](https://en.wikipedia.org/wiki/Common_Language_Infrastructure) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Software comercial](https://en.wikipedia.org/wiki/Commercial_software)[propietario](https://en.wikipedia.org/wiki/Proprietary_software) |
| **Sitio web** | [www .sourcegear .com / bóveda /](http://www.sourcegear.com/vault/) |

2003

**ArX** es un sistema de [control de versiones distribuidas](https://en.wikipedia.org/wiki/Distributed_version_control) . ArX comenzó como una bifurcación del [arco](https://en.wikipedia.org/wiki/GNU_arch) de [GNU](https://en.wikipedia.org/wiki/GNU_arch) y está licenciado bajo la [GPL](https://en.wikipedia.org/wiki/GNU_General_Public_License) . Desde el tenedor, ArX ha sido ampliamente reescrito en C ++, con muchas características nuevas. El mantenedor del proyecto es Walter Landry. Landry fue, durante un corto tiempo, el mantenedor de Arch, y bifurcó a ArX cuando Tom Lord reasumió el mantenimiento de Arch y no aceptó algunas de las instrucciones de desarrollo de Landry. [[1]](https://en.wikipedia.org/wiki/ArX#cite_note-1)[[2]](https://en.wikipedia.org/wiki/ArX#cite_note-2) El tenedor se anunció en enero de 2003 y el primer código se lanzó en febrero. [[3]](https://en.wikipedia.org/wiki/ArX#cite_note-3)[[4]](https://en.wikipedia.org/wiki/ArX#cite_note-4) Durante un tiempo, ArX compartió una lista de correo y una comunidad con Arch, pero Landry fundó una nueva lista de correo en agosto de 2003 y la serie previa se convirtió en la serie de lanzamiento 1.0 en diciembre. [[5]](https://en.wikipedia.org/wiki/ArX#cite_note-5)[[6]](https://en.wikipedia.org/wiki/ArX#cite_note-6) La serie 2.0, que era incompatible con Arch, se hizo pública en octubre de 2004.

|  |  |
| --- | --- |
| **ArX** | |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | Walter Landry |
| **Versión inicial** | Febrero de 2003 ; hace 15 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 2.2.4 / 16 de noviembre de 2005 ; Hace 12 años |
| **Estado del desarrollo** | Sin mantenimiento |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Linux](https://en.wikipedia.org/wiki/Linux) , [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) , [Mac OS X](https://en.wikipedia.org/wiki/Mac_OS_X) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [GPL](https://en.wikipedia.org/wiki/GNU_General_Public_License) |
| **Sitio web** | [www .nongnu .org/ arx /](http://www.nongnu.org/arx/) |

2003 **Monotone** es una herramienta de [software de](https://en.wikipedia.org/wiki/Software)[código](https://en.wikipedia.org/wiki/Open_source)[abierto](https://en.wikipedia.org/wiki/Software) para el [control de revisiones distribuidas](https://en.wikipedia.org/wiki/Distributed_revision_control) .

Monotone realiza un seguimiento de las revisiones de los archivos, agrupa conjuntos de revisiones en conjuntos de cambios y rastrea el historial a través de los cambios de nombre. El objetivo del proyecto es la integridad sobre el rendimiento. [[1]](https://en.wikipedia.org/wiki/Monotone_(software)#cite_note-1) Monotone está diseñado para operaciones distribuidas, y hace un uso intensivo de [primitivas criptográficas](https://en.wikipedia.org/wiki/Cryptographic_primitive) para rastrear revisiones de archivos (a través del [hash](https://en.wikipedia.org/wiki/SHA-1)[seguro](https://en.wikipedia.org/wiki/Cryptographic_hash_function)[SHA-1](https://en.wikipedia.org/wiki/SHA-1) ) y para autenticar acciones del usuario (a través de [firmas criptográficas](https://en.wikipedia.org/wiki/Digital_signature)[RSA](https://en.wikipedia.org/wiki/RSA_(algorithm)) ).

|  |  |
| --- | --- |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | El equipo monótono |
| **Versión inicial** | 6 de abril de 2003 ; hace 14 años |
| **Escrito en** | C ++ |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Unix](https://en.wikipedia.org/wiki/Unix) , [Linux](https://en.wikipedia.org/wiki/Linux) , [BSD](https://en.wikipedia.org/wiki/BSD) , [Mac OS X](https://en.wikipedia.org/wiki/Mac_OS_X) , [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) |
| **Disponible en** | Inglés, italiano, sueco, portugués, alemán, español |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Licencia pública general de GNU](https://en.wikipedia.org/wiki/GNU_General_Public_License) |
| **Sitio web** | [www .monotone .ca](http://www.monotone.ca/) |

diseño. Al igual que [GNU arch](https://en.wikipedia.org/wiki/GNU_arch) , ya diferencia de [Subversion](https://en.wikipedia.org/wiki/Subversion_(software)) , Monotone adopta un enfoque distribuido para el control de versiones. Monotone usa hashes [SHA-1](https://en.wikipedia.org/wiki/SHA-1) para identificar archivos específicos o grupos de archivos, como con [Git](https://en.wikipedia.org/wiki/Git_(software)) y [Mercurial](https://en.wikipedia.org/wiki/Mercurial_(software)) , en lugar de números de revisión lineales. Cada participante mantiene su propio historial de revisión, almacenado en una [base de](https://en.wikipedia.org/wiki/SQLite) datos [SQLite](https://en.wikipedia.org/wiki/SQLite) local .

### Integridad [ [editar](https://en.wikipedia.org/w/index.php?title=Monotone_(software)&action=edit&section=2)]

Antes de una gran optimización en la revisión 0.27, el énfasis de Monotone en la corrección sobre la optimización a menudo se atribuía a experiencias iniciales pobres. La primera acción de un nuevo usuario suele sincronizar (clonar) una gran base de datos Monotone existente, una acción que a menudo tomaba horas para grandes bases de datos, debido a la amplia validación y comprobación de integridad que realiza Monotone cuando las revisiones se mueven a través de la red. Una vez que se completa la base de datos inicial (clonar), las acciones posteriores generalmente se realizan más rápidamente. A partir de julio de 2010 , todavía hay lugar para una mayor optimización en algunas funciones más raras. [[*citación necesitada*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

### Flujo de trabajo [ [editar](https://en.wikipedia.org/w/index.php?title=Monotone_(software)&action=edit&section=3)]

Monotone es especialmente fuerte en su soporte para un flujo de trabajo de divergencia / fusión, que logra en parte permitiendo siempre la confirmación antes de la fusión. [[2]](https://en.wikipedia.org/wiki/Monotone_(software)#cite_note-2)

### Redes [ [editar](https://en.wikipedia.org/w/index.php?title=Monotone_(software)&action=edit&section=4)]

Aunque Monotone originalmente admitía una variedad de protocolos de red para sincronizar árboles, ahora utiliza exclusivamente un protocolo personalizado llamado *netsync* , que es más robusto y eficiente, y comparte algunas bases conceptuales con [rsync](https://en.wikipedia.org/wiki/Rsync) y [cvsup](https://en.wikipedia.org/w/index.php?title=Cvsup&action=edit&redlink=1) . (Sin embargo, a partir de la versión 0.27, es posible utilizar el protocolo netsync *sobre*cualquier transmisión, especialmente las conexiones ssh.) *Netsync* tiene su propio puerto asignado por la [IANA](https://en.wikipedia.org/wiki/Internet_Assigned_Numbers_Authority) (4691) y las versiones anteriores son compatibles con un complemento de [Wireshark](https://en.wikipedia.org/wiki/Wireshark) para el análisis del tráfico No hay un servidor Monotone separado porque cualquier cliente Monotone puede actuar como servidor.

Otras características de Monotone incluyen:

* Buen soporte para [internacionalización y localización](https://en.wikipedia.org/wiki/Internationalization_and_localization)
* Diseño portátil, implementado en [C ++](https://en.wikipedia.org/wiki/C%2B%2B)
* La alta integridad es un objetivo clave de diseño
* Monotone puede importar proyectos de [CVS](https://en.wikipedia.org/wiki/Concurrent_Versions_System) .
* [Firma](https://en.wikipedia.org/wiki/Digital_signature) de revisiones utilizando certificados [RSA](https://en.wikipedia.org/wiki/RSA_(algorithm))
* Fácil de aprender, debido a un conjunto de comandos similar al de [CVS](https://en.wikipedia.org/wiki/Concurrent_Versions_System)
* Muy bueno en la ramificación (ambas divergencias dentro de una rama y ramas nombradas) y fusión
* Buena documentación
* Muy bajo mantenimiento
* Existen [interfaces de usuario gráficas](https://en.wikipedia.org/wiki/Graphical_user_interface) estables :
  + [guitone](http://guitone.thomaskeller.biz/) , un [Qt](https://en.wikipedia.org/wiki/Qt_(toolkit)) -front para gestionar espacios de trabajo y bases de datos (compatible con MS-Windows y Unix / Linux / MacOS)
  + [mtn-browse](http://www.coosoft.plus.com/software.html) , un navegador gráfico [Gtk2](https://en.wikipedia.org/wiki/GTK2) que le permite explorar la base de datos, incluso de forma remota, sin la necesidad de un espacio de trabajo (compatible con Unix / Linux / MacOS)
  + [Monotone-Viz](http://oandrieu.nerim.net/monotone-viz) , un historial de revisión grapher (compatible con MS-Windows y Unix / Linux)
  + [TracMtn](http://tracmtn.1erlei.de/) , un plugin [Trac](https://en.wikipedia.org/wiki/Trac) para el historial y la búsqueda de repositorios
* Completa y completa biblioteca [Perl](https://en.wikipedia.org/wiki/Perl) que le permite controlar completamente Monotone desde un script Perl (mtn-browse hace uso de esto)

### Inconvenientes [ [editar](https://en.wikipedia.org/w/index.php?title=Monotone_(software)&action=edit&section=6)]

A partir de enero de 2008 , los posibles inconvenientes de Monotone incluyen:

* Los usuarios potenciales no pueden verificar (o comprometer) desde detrás de un proxy (muy común en entornos corporativos) debido al protocolo no http.
* Menos popular que [DVCS de](https://en.wikipedia.org/wiki/Distributed_Version_Control_System) código abierto [competidor](https://en.wikipedia.org/wiki/Distributed_Version_Control_System) , como [Bazar](https://en.wikipedia.org/wiki/Bazaar_(software)) , [Git](https://en.wikipedia.org/wiki/Git_(software)) o [Mercurial](https://en.wikipedia.org/wiki/Mercurial_(software))
* Problemas de rendimiento para ciertas operaciones (el tirón inicial más notable)

2003

**SVK** (también escrito **svk** ) es un [sistema de control de versiones](https://en.wikipedia.org/wiki/Version_control_system) descentralizado escrito en [Perl](https://en.wikipedia.org/wiki/Perl) , con un diseño distribuido jerárquico comparable a la implementación centralizada de [BitKeeper](https://en.wikipedia.org/wiki/BitKeeper) y [GNU arch](https://en.wikipedia.org/wiki/GNU_arch) . Se distribuye bajo la [Licencia artística](https://en.wikipedia.org/wiki/Artistic_License) y la [Licencia pública general de GNU](https://en.wikipedia.org/wiki/GNU_General_Public_License) . El autor principal de svk es Kao Chia-liang ( [chino](https://en.wikipedia.org/wiki/Chinese_language) : 高 嘉 良 ). El 5 de junio de 2006, Chia-liang Kao se unió a Best Practical Solutions, LLC, fabricantes de [Request Tracker](https://en.wikipedia.org/wiki/Request_Tracker) (y también usuarios importantes de SVK), y SVK se convirtió en un producto Best Practical. [[1]](https://en.wikipedia.org/wiki/SVK#cite_note-1)

El 28 de mayo de 2009, Chia-liang Kao anunció que Best Practical ya no estará desarrollando activamente SVK. [[2]](https://en.wikipedia.org/wiki/SVK#cite_note-2) Kao continuó desarrollando SVK hasta el último lanzamiento en marzo de 2010, cuando el desarrollo parece haberse detenido.

SVK usa el [sistema de](https://en.wikipedia.org/wiki/Apache_Subversion) archivos [Subversion](https://en.wikipedia.org/wiki/Apache_Subversion) pero proporciona características adicionales:

* Operaciones fuera de línea como *checkin* , *log* , *merge* .
* Ramas distribuidas.
* Ligera gestión de copia de pago (no *.svn* directorios).
* Algoritmos avanzados de [fusión](https://en.wikipedia.org/wiki/Merge_(revision_control)) , como *combinación de estrellas* y *selección de cerezas* .
* Cambio de firma y verificación.
* Puede duplicar y operar en repositorios [Subversion](https://en.wikipedia.org/wiki/Apache_Subversion) , [Perforce](https://en.wikipedia.org/wiki/Perforce) y [CVS](https://en.wikipedia.org/wiki/Concurrent_Versions_System) .

|  |  |
| --- | --- |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | Kao Chia-liang ([chino](https://en.wikipedia.org/wiki/Chinese_language) : 高 嘉 良 ) |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | Best Practical Solutions, LLC |
| **Versión inicial** | 19 de noviembre de 2003 ; hace 14 años |
| [**Último lanzamiento**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 2.2.3 / 21 de marzo de 2010 ; hace 7 años |
| **Estado del desarrollo** | Interrumpido |
| **Escrito en** | [Perl](https://en.wikipedia.org/wiki/Perl) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Licencia pública general de GNU](https://en.wikipedia.org/wiki/GNU_General_Public_License) o[licencia artística](https://en.wikipedia.org/wiki/Artistic_License) |
| **Sitio web** | [metacpan .org / module / SVK](https://metacpan.org/module/SVK) |

2005

**Mercurial** es una herramienta [distribuida de control de revisiones](https://en.wikipedia.org/wiki/Distributed_revision_control) para [desarrolladores de software](https://en.wikipedia.org/wiki/Software_developer) . Es compatible con [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) y [sistemas tipo](https://en.wikipedia.org/wiki/Unix-like) Unix, como [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD) , [macOS](https://en.wikipedia.org/wiki/MacOS) y [Linux](https://en.wikipedia.org/wiki/Linux) .

Los principales objetivos de diseño de Mercurial incluyen alto rendimiento y escalabilidad, desarrollo colaborativo descentralizado y completamente distribuido, manejo robusto tanto de [texto sin formato](https://en.wikipedia.org/wiki/Plain_text) como de [archivos binarios](https://en.wikipedia.org/wiki/Binary_file) , y funciones avanzadas de bifurcación y fusión, sin dejar de ser conceptualmente simples. [[3]](https://en.wikipedia.org/wiki/Mercurial#cite_note-3) Incluye una interfaz web integrada. Mercurial también ha tomado medidas para facilitar la transición para los usuarios de otros sistemas de control de versiones, particularmente [Subversion](https://en.wikipedia.org/wiki/Apache_Subversion) . Mercurial es principalmente un programa impulsado por [línea de comandos](https://en.wikipedia.org/wiki/Command_line_interface) , pero [las](https://en.wikipedia.org/wiki/Graphical_user_interface) extensiones de [interfaz gráfica de usuario](https://en.wikipedia.org/wiki/Graphical_user_interface) están disponibles, por ejemplo , [TortoiseHg](https://en.wikipedia.org/wiki/TortoiseHg) y varios [IDEs](https://en.wikipedia.org/wiki/Integrated_development_environment)ofrece soporte para el control de versiones con Mercurial. Todas las operaciones de Mercurial se invocan como argumentos para su programa controlador hg(una referencia a Hg, el [símbolo químico](https://en.wikipedia.org/wiki/Chemical_symbol) del elemento [mercurio](https://en.wikipedia.org/wiki/Mercury_(element)) ).

Matt Mackall originó Mercurial y sirve como su desarrollador principal. Mercurial se lanza como [software libre](https://en.wikipedia.org/wiki/Free_software) bajo los términos de [GNU GPL v2](https://en.wikipedia.org/wiki/GNU_General_Public_License#Version_2) (o cualquier versión posterior [[4]](https://en.wikipedia.org/wiki/Mercurial#cite_note-4) ). Se lleva a cabo principalmente mediante el [lenguaje de programación Python](https://en.wikipedia.org/wiki/Python_(programming_language)) , pero incluye un binario [diff](https://en.wikipedia.org/wiki/Diff) aplicación escrita en [C](https://en.wikipedia.org/wiki/C_(programming_language)) .

Mercurial usa hashes [SHA-1](https://en.wikipedia.org/wiki/SHA-1) para identificar revisiones. Para el acceso al repositorio a través de una red, Mercurial usa un [protocolo](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) basado en [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) que busca reducir las solicitudes de ida y vuelta, las nuevas conexiones y los datos transferidos. Mercurial también puede funcionar a través de [SSH,](https://en.wikipedia.org/wiki/Secure_Shell) donde el protocolo es muy similar al protocolo basado en HTTP. De forma predeterminada, utiliza una [combinación de 3 vías](https://en.wikipedia.org/wiki/3-way_merge) antes de llamar a las herramientas de combinación externas.

|  |  |
| --- | --- |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | Matt Mackall |
| **Versión inicial** | 19 de abril de 2005 ; Hace 12 años [[1]](https://en.wikipedia.org/wiki/Mercurial#cite_note-announced-1) |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 4.5 [[2]](https://en.wikipedia.org/wiki/Mercurial#cite_note-2) / 1 de febrero de 2018 ; hace 5 días |
| [**Repositorio**](https://en.wikipedia.org/wiki/Repository_(version_control)) | * <https://www.mercurial-scm.org/repo/hg-stable>   [Edita esto en Wikidata](https://www.wikidata.org/wiki/Q476543#P1324) |
| **Escrito en** | [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) y [C](https://en.wikipedia.org/wiki/C_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Unix-like](https://en.wikipedia.org/wiki/Unix-like) , [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) ,[macOS](https://en.wikipedia.org/wiki/MacOS) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de versiones](https://en.wikipedia.org/wiki/Version_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [GNU GPL v2 +](https://en.wikipedia.org/wiki/GNU_General_Public_License#Version_2) |
| **Sitio web** | [www .mercurial-scm.org](https://www.mercurial-scm.org/) |

2005. **GNU Bazaar** (anteriormente **Bazaar-NG** , herramienta de línea de comandos **bzr**) es un sistema de [control de revisiones](https://en.wikipedia.org/wiki/Revision_control)[distribuido](https://en.wikipedia.org/wiki/Distributed_version_control) y [cliente-servidor](https://en.wikipedia.org/wiki/Client%E2%80%93server) patrocinado por [Canonical](https://en.wikipedia.org/wiki/Canonical_Ltd.) .

Bazaar puede ser utilizado por un único desarrollador que trabaje en múltiples ramas de contenido local o por equipos que colaboran en una red.

Bazaar está escrito en el [lenguaje de programación Python](https://en.wikipedia.org/wiki/Python_programming_language) , con paquetes para las principales distribuciones de [GNU / Linux](https://en.wikipedia.org/wiki/Linux) , [Mac OS X](https://en.wikipedia.org/wiki/Mac_OS_X) y [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) . Bazaar es software libre y parte del [Proyecto GNU](https://en.wikipedia.org/wiki/GNU_Project) .

Los comandos de Bazar son similares a los encontrados en [CVS](https://en.wikipedia.org/wiki/Concurrent_Versions_System) o [Subversion](https://en.wikipedia.org/wiki/Subversion_(software)) . Se puede iniciar y mantener un nuevo proyecto sin un servidor de repositorio remoto invocando bzr initen un directorio que una persona desea versionar. [[6]](https://en.wikipedia.org/wiki/GNU_Bazaar#cite_note-bzr-man-page-6)

A diferencia de los sistemas de control de versiones puramente distribuidos que no usan un servidor central, Bazaar admite el trabajo con o sin un servidor central. [[*aclaración necesaria*](https://en.wikipedia.org/wiki/Wikipedia:Please_clarify)] Es posible utilizar ambos métodos al mismo tiempo con el mismo proyecto. Los sitios web [Launchpad](https://en.wikipedia.org/wiki/Launchpad_(website)) y [Sourceforge](https://en.wikipedia.org/wiki/Sourceforge) proporcionan un servicio de alojamiento gratuito para proyectos gestionados con Bazar.

Bazaar tiene soporte para trabajar con algunos otros sistemas de control de revisiones. [[7]](https://en.wikipedia.org/wiki/GNU_Bazaar#cite_note-7) Esto permite a los usuarios derivarse de otro sistema (como [Subversion](https://en.wikipedia.org/wiki/Subversion_(software))[[8]](https://en.wikipedia.org/wiki/GNU_Bazaar#cite_note-8) ), realizar cambios locales y enviarlos a una rama de Bazar, y luego fusionarlos de nuevo en el otro sistema. El acceso de solo lectura también está disponible para [Git](https://en.wikipedia.org/wiki/Git_(software))[[9]](https://en.wikipedia.org/wiki/GNU_Bazaar#cite_note-9) y [Mercurial](https://en.wikipedia.org/wiki/Mercurial_(software)) . [[10]](https://en.wikipedia.org/wiki/GNU_Bazaar#cite_note-10)Bazaar también permite la interoperación con muchos otros sistemas (incluidos [CVS](https://en.wikipedia.org/wiki/Concurrent_Versions_System) , [Darcs](https://en.wikipedia.org/wiki/Darcs) , [Git](https://en.wikipedia.org/wiki/Git_(software)) , [Perforce](https://en.wikipedia.org/wiki/Perforce) , [Mercurial](https://en.wikipedia.org/wiki/Mercurial) ) al permitir que uno importe / exporte el historial. [[11]](https://en.wikipedia.org/wiki/GNU_Bazaar#cite_note-11)

Bazaar admite archivos con nombres del conjunto [Unicode](https://en.wikipedia.org/wiki/Unicode) completo . También permite que los mensajes de confirmación, los nombres de los ejecutores, etc. estén en Unicode.

|  |  |
| --- | --- |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | Martin Pool |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Canonical](https://en.wikipedia.org/wiki/Canonical_Ltd.) y comunidad |
| **Versión inicial** | 26 de marzo de 2005 ; Hace 12 años [[1]](https://en.wikipedia.org/wiki/GNU_Bazaar#cite_note-1) |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 2.7.0 (15 de febrero de 2016 ; hace 23 meses ) [[±]](https://en.wikipedia.org/w/index.php?title=Template:Latest_stable_software_release/GNU_Bazaar&action=edit)[[2]](https://en.wikipedia.org/wiki/GNU_Bazaar#cite_note-2) |
| [**Lanzamiento de vista previa**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 2.6b2 (24 de julio de 2012 ; 5 años atrás ) [[±]](https://en.wikipedia.org/w/index.php?title=Template:Latest_preview_software_release/GNU_Bazaar&action=edit) |
| [**Repositorio**](https://en.wikipedia.org/wiki/Repository_(version_control)) | * <https://code.launchpad.net/bzr>   [Edita esto en Wikidata](https://www.wikidata.org/wiki/Q812656#P1324) |
| **Estado del desarrollo** | Stalled |
| **Escrito en** | [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) 2, [Pyrex](https://en.wikipedia.org/wiki/Pyrex_(programming_language)) (opcional), [C](https://en.wikipedia.org/wiki/C_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Multiplataforma](https://en.wikipedia.org/wiki/Cross-platform) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Distribuido](https://en.wikipedia.org/wiki/Distributed_version_control) y [cliente-servidor](https://en.wikipedia.org/wiki/Client%E2%80%93server)[de control de revisión](https://en.wikipedia.org/wiki/Revision_control) del sistema |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [GPLv2](https://en.wikipedia.org/wiki/GPLv2) o posterior [[3]](https://en.wikipedia.org/wiki/GNU_Bazaar#cite_note-3) |
| **Sitio web** | [bazar .canonical .com](http://bazaar.canonical.com/) |

2005. **Git** ( [/ ɡ ɪ t /](https://en.wikipedia.org/wiki/Help:IPA/English)[[7]](https://en.wikipedia.org/wiki/Git#cite_note-:0-7) ) es un [sistema de control de versiones](https://en.wikipedia.org/wiki/Version_control_system) para rastrear cambios en [archivos de computadora](https://en.wikipedia.org/wiki/Computer_file) y coordinar el trabajo en esos archivos entre varias personas. Se utiliza principalmente para la [gestión de código fuente](https://en.wikipedia.org/wiki/Source_code_management) en [el desarrollo de software](https://en.wikipedia.org/wiki/Software_development) , [[8]](https://en.wikipedia.org/wiki/Git#cite_note-effcomp-8) pero se puede utilizar para realizar un seguimiento de los cambios en cualquier conjunto de archivos. Como sistema de [control de revisión distribuido](https://en.wikipedia.org/wiki/Distributed_revision_control) , está orientado a la velocidad, [[9]](https://en.wikipedia.org/wiki/Git#cite_note-kernel_SCM_saga-9) integridad de datos, [[10]](https://en.wikipedia.org/wiki/Git#cite_note-integrity_goals-10) y soporte para flujos de trabajo distribuidos, no lineales. [[11]](https://en.wikipedia.org/wiki/Git#cite_note-linusGoogleTalk-11)

Git fue creado por [Linus Torvalds](https://en.wikipedia.org/wiki/Linus_Torvalds) en 2005 para el desarrollo del [kernel de Linux](https://en.wikipedia.org/wiki/Linux_kernel) , y otros desarrolladores del kernel contribuyeron a su desarrollo inicial. [[12]](https://en.wikipedia.org/wiki/Git#cite_note-pro-git-1.2-12) Su mantenedor actual desde 2005 es [Junio ​​Hamano](https://en.wikipedia.org/wiki/Junio_Hamano) .

Como con la mayoría de los otros sistemas de control de versiones distribuidas, ya diferencia de la mayoría de [los](https://en.wikipedia.org/wiki/Client%E2%80%93server) sistemas [cliente-servidor](https://en.wikipedia.org/wiki/Client%E2%80%93server) , cada [directorio de](https://en.wikipedia.org/wiki/Directory_(computing)) Git en cada [computadora](https://en.wikipedia.org/wiki/Node_(networking)) es un [repositorio](https://en.wikipedia.org/wiki/Repository_(version_control)) completo con historial completo y capacidad de seguimiento de versión completa, independientemente del acceso a la red o un servidor central. [[13]](https://en.wikipedia.org/wiki/Git#cite_note-13)

Git es [un software gratuito](https://en.wikipedia.org/wiki/Free_software) distribuido bajo los términos de la [Licencia Pública General GNU](https://en.wikipedia.org/wiki/GNU_General_Public_License) versión 2.

El diseño de Git fue inspirado por [BitKeeper](https://en.wikipedia.org/wiki/BitKeeper) y [Monotone](https://en.wikipedia.org/wiki/Monotone_(software)) . [[27]](https://en.wikipedia.org/wiki/Git#cite_note-27)[[28]](https://en.wikipedia.org/wiki/Git#cite_note-a_quick_challenge-28) Git fue originalmente diseñado como un motor de sistema de control de versiones de bajo nivel además de que otros podían escribir [interfaces](https://en.wikipedia.org/w/index.php?title=StGIT&action=edit&redlink=1) , como [Cogito](https://en.wikipedia.org/wiki/Cogito_(software)) o [StGIT](https://en.wikipedia.org/w/index.php?title=StGIT&action=edit&redlink=1) . [[28]](https://en.wikipedia.org/wiki/Git#cite_note-a_quick_challenge-28) El proyecto core de Git se convirtió en un sistema completo de control de versiones que se puede usar directamente. [[29]](https://en.wikipedia.org/wiki/Git#cite_note-bare_url-29)Aunque fuertemente influenciado por BitKeeper, Torvalds evitó deliberadamente los enfoques convencionales, lo que condujo a un diseño único. [[30]](https://en.wikipedia.org/wiki/Git#cite_note-30)

**Características**[ [editar](https://en.wikipedia.org/w/index.php?title=Git&action=edit&section=4)]

El diseño de Git es una síntesis de la experiencia de Torvalds con Linux en el mantenimiento de un gran proyecto de desarrollo distribuido, junto con su conocimiento profundo del rendimiento del sistema de archivos obtenido del mismo proyecto y la necesidad urgente de producir un sistema operativo en poco tiempo. Estas influencias llevaron a las siguientes opciones de implementación [[*cita requerida*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] :

**Soporte fuerte para el desarrollo no lineal**

Git admite ramificaciones y fusiones rápidas, e incluye herramientas específicas para visualizar y navegar un historial de desarrollo no lineal. En Git, una suposición básica es que un cambio se fusionará con más frecuencia de lo que está escrito, ya que se transmite a varios revisores. En Git, las ramas son muy livianas: una rama es solo una referencia a una confirmación. Con sus confirmaciones parentales, se puede construir la estructura de rama completa.

**Desarrollo distribuido**

Al igual que [Darcs](https://en.wikipedia.org/wiki/Darcs) , [BitKeeper](https://en.wikipedia.org/wiki/BitKeeper) , [Mercurial](https://en.wikipedia.org/wiki/Mercurial) , [SVK](https://en.wikipedia.org/wiki/SVK) , [Bazaar](https://en.wikipedia.org/wiki/Bazaar_(software)) y [Monotone](https://en.wikipedia.org/wiki/Monotone_(software)) , Git proporciona a cada desarrollador una copia local del historial de desarrollo completo y los cambios se copian de uno de esos repositorios a otro. Estos cambios se importan como ramas de desarrollo agregadas y se pueden combinar de la misma manera que una rama desarrollada localmente.

**Compatibilidad con sistemas y protocolos existentes**

Los repositorios se pueden publicar a través de [Protocolo de transferencia de hipertexto](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) (HTTP), [Protocolo de transferencia de archivos](https://en.wikipedia.org/wiki/File_Transfer_Protocol) (FTP), [rsync](https://en.wikipedia.org/wiki/Rsync) (eliminado en Git 2.8.0 [[31]](https://en.wikipedia.org/wiki/Git#cite_note-31) ) o un protocolo Git sobre un socket simple o [Secure Shell](https://en.wikipedia.org/wiki/Secure_Shell) (ssh). Git también tiene una emulación de servidor CVS, que permite el uso de clientes CVS existentes y complementos IDE para acceder a repositorios Git. [Subversion](https://en.wikipedia.org/wiki/Apache_Subversion) y [svk](https://en.wikipedia.org/wiki/Svk) repositories se pueden usar directamente con git-svn.

**Manejo eficiente de grandes proyectos**

Torvalds describió a Git como muy rápido y escalable, [[32]](https://en.wikipedia.org/wiki/Git#cite_note-32) y las pruebas de rendimiento realizadas por Mozilla [[33]](https://en.wikipedia.org/wiki/Git#cite_note-33) mostraron que era un [orden de magnitud](https://en.wikipedia.org/wiki/Order_of_magnitude) más rápido que algunos sistemas de control de versiones, y recuperar el historial de versiones de un repositorio almacenado localmente puede ser cien veces más rápido que ir a buscarlo desde el servidor remoto. [[34]](https://en.wikipedia.org/wiki/Git#cite_note-34)

**Autenticación criptográfica de la historia**

El historial de Git se almacena de tal manera que el ID de una versión particular (una *confirmación* en términos de Git) depende del historial de desarrollo completo que conduzca a esa confirmación. Una vez que se publica, no es posible cambiar las versiones anteriores sin que se note. La estructura es similar a un [árbol de Merkle](https://en.wikipedia.org/wiki/Merkle_tree) , pero con datos agregados en los nodos y hojas. [[35]](https://en.wikipedia.org/wiki/Git#cite_note-35) ( [Mercurial](https://en.wikipedia.org/wiki/Mercurial) y [Monotone](https://en.wikipedia.org/wiki/Monotone_(software)) también tienen esta propiedad).

**Diseño basado en Toolkit**

Git fue diseñado como un conjunto de programas escritos en [C](https://en.wikipedia.org/wiki/C_(programming_language)) y varios scripts de shell que proporcionan envoltorios alrededor de esos programas. [[36]](https://en.wikipedia.org/wiki/Git#cite_note-36) Aunque la mayoría de esos scripts han sido reescritos en C para velocidad y portabilidad, el diseño permanece, y es fácil encadenar los componentes entre sí. [[37]](https://en.wikipedia.org/wiki/Git#cite_note-37)

**Estrategias de combinación conectables**

Como parte de su diseño de kit de herramientas, Git tiene un modelo bien definido de una fusión incompleta, y tiene múltiples algoritmos para completarlo, que culminan en decirle al usuario que no puede completar la fusión automáticamente y que es necesaria la edición manual.

[**La basura se**](https://en.wikipedia.org/wiki/Garbage_(computer_science))**acumula hasta ser recolectada**

Abortar operaciones o anular cambios dejará objetos colgantes inútiles en la base de datos. En general, son una pequeña fracción de la historia de objetos buscados que crece continuamente. Git realizará automáticamente la [recolección de basura](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)) cuando se hayan creado suficientes objetos sueltos en el repositorio. La recolección de basura se puede llamar usando explícitamente git gc --prune. [[38]](https://en.wikipedia.org/wiki/Git#cite_note-38)

**Embalaje de objeto explícito periódico**

Git almacena cada objeto recién creado como un archivo separado. Aunque está comprimido individualmente, ocupa mucho espacio y es ineficiente. Esto se resuelve mediante el uso de *paquetes* que almacenan una gran cantidad de objetos [delta-comprimidos](https://en.wikipedia.org/wiki/Delta_encoding) entre ellos en un archivo (o secuencia de bytes de red) llamado *packfile* . Los paquetes se comprimen usando la [heurística](https://en.wikipedia.org/wiki/Heuristic_(computer_science))Es probable que los archivos con el mismo nombre sean similares, pero no confíe en ello para que sean correctos. Se crea un archivo de índice correspondiente para cada archivo de paquete, que indica el desplazamiento de cada objeto en el archivo de paquete. Los objetos recién creados (con el historial recién agregado) aún se almacenan como objetos únicos y es necesario volver a empacar periódicamente para mantener la eficiencia del espacio. El proceso de empaquetar el repositorio puede ser muy costoso desde el punto de vista computacional. Al permitir que los objetos existan en el repositorio en un formato suelto pero generado rápidamente, Git permite postergar la costosa operación del paquete hasta más tarde, cuando el tiempo importa menos, por ejemplo, al final de un día de trabajo. Git realiza el reempaquetado periódico de forma automática, pero también es posible el reempaquetado manual con el comando git gc . Para la integridad de los datos, tanto el archivo de paquete como su índice tienen un [SHA-1](https://en.wikipedia.org/wiki/SHA-1)suma de comprobación dentro y el nombre del archivo packfile también contiene una suma de comprobación SHA-1. Para verificar la integridad de un repositorio, ejecute el comando git fsck .

Otra propiedad de Git es que snapshots directory trees of files. Los primeros sistemas para rastrear versiones de código fuente, [Sistema de control de código fuente](https://en.wikipedia.org/wiki/Source_Code_Control_System) (SCCS) y [Sistema de control de revisión](https://en.wikipedia.org/wiki/Revision_Control_System) (RCS), trabajaron en archivos individuales y enfatizaron el ahorro de espacio de los [deltas intercalados](https://en.wikipedia.org/wiki/Interleaved_deltas) (SCCS) o la [codificación delta](https://en.wikipedia.org/wiki/Delta_encoding) (RCS). (en su mayoría similares) versiones. Los sistemas de control de revisiones posteriores mantuvieron esta noción de que un archivo tiene una identidad en múltiples revisiones de un proyecto. Sin embargo, Torvalds rechazó este concepto. [[39]](https://en.wikipedia.org/wiki/Git#cite_note-39) En consecuencia, Git no registra explícitamente las relaciones de revisión de archivos en ningún nivel debajo del árbol de códigos fuente.

Estas relaciones de revisión implícitas tienen algunas consecuencias importantes:

* Es un poco más costoso examinar el historial de cambios de un archivo que el proyecto completo. [[40]](https://en.wikipedia.org/wiki/Git#cite_note-40) Para obtener un historial de cambios que afectan un archivo dado, Git debe recorrer el historial global y luego determinar si cada cambio modificó ese archivo. Sin embargo, este método de examen de historial permite que Git produzca con la misma eficacia un solo historial que muestre los cambios en un conjunto arbitrario de archivos. Por ejemplo, un subdirectorio del árbol fuente más un archivo de encabezado global asociado es un caso muy común.
* Los renombrados se manejan implícitamente en lugar de explícitamente. Una queja común con [CVS](https://en.wikipedia.org/wiki/Concurrent_Versions_System) es que usa el nombre de un archivo para identificar su historial de revisión, por lo que no es posible mover o renombrar un archivo sin interrumpir su historial o cambiar el nombre del historial y, por lo tanto, hacer que el historial sea inexacto. La mayoría de los sistemas de control de revisión posteriores al CVS lo resuelven dando a un archivo un nombre único de larga vida (análogo al número de [inodo](https://en.wikipedia.org/wiki/Inode) ) que sobrevive al cambio de nombre. Git no registra ese identificador, y esto se reclama como una ventaja. [[41]](https://en.wikipedia.org/wiki/Git#cite_note-41)[[42] Los](https://en.wikipedia.org/wiki/Git#cite_note-42) archivos de [código fuente a](https://en.wikipedia.org/wiki/Source_code) veces se dividen o combinan, o simplemente se renombran, [[43]](https://en.wikipedia.org/wiki/Git#cite_note-43)y registrar esto como un simple cambio de nombre congelaría una descripción inexacta de lo que sucedió en la historia (inmutable). Git aborda el problema detectando los cambios de nombre mientras navega por el historial de instantáneas en lugar de registrarlo al hacer la instantánea. [[44]](https://en.wikipedia.org/wiki/Git#cite_note-44) (En pocas palabras, dado un archivo en la revisión *N,* un archivo con el mismo nombre en la revisión *N-1* es su antecesor predeterminado. Sin embargo, cuando no hay un archivo con el mismo nombre en la revisión *N-1,*Git busca un archivo que existía solo en la revisión *N-1* y es muy similar al nuevo archivo.) Sin embargo, requiere más [CPU](https://en.wikipedia.org/wiki/Central_processing_unit)-intensivo trabajo cada vez que se revisa el historial, y varias opciones para ajustar la heurística están disponibles. Este mecanismo no siempre funciona; A veces, un archivo que se renombra con cambios en la misma confirmación se lee como una eliminación del archivo anterior y la creación de un nuevo archivo. Los desarrolladores pueden evitar esta limitación al comprometer el cambio de nombre y los cambios por separado.

Git implementa varias estrategias de fusión; se puede seleccionar una estrategia no predeterminada en el momento de la fusión: [[45]](https://en.wikipedia.org/wiki/Git#cite_note-45)

* *resolver* : el algoritmo de [fusión](https://en.wikipedia.org/wiki/Three-way_merge) tripartita tradicional .
* *recursivo* : este es el valor predeterminado al tirar o fusionar una rama, y ​​es una variante del algoritmo de fusión de tres vías.

Cuando hay más de un antepasado común que se puede utilizar para la combinación de tres vías, crea un árbol fusionado de los antepasados ​​comunes y lo utiliza como árbol de referencia para la combinación de tres vías. Se ha informado que esto provoca menos conflictos de fusión sin causar confusiones erróneas mediante pruebas realizadas en las asignaciones de fusión anteriores tomadas del historial de desarrollo del kernel de Linux 2.6. Además, esto puede detectar y manejar fusiones que implican cambios de nombre.

-  *Linus Torvalds*[[46]](https://en.wikipedia.org/wiki/Git#cite_note-46)

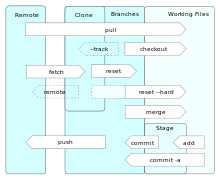
* *pulpo* : este es el valor predeterminado cuando se fusionan más de dos cabezas.

**Estructuras de datos**[ [editar](https://en.wikipedia.org/w/index.php?title=Git&action=edit&section=5)]

Las primitivas de Git no son intrínsecamente un sistema de [gestión de código fuente](https://en.wikipedia.org/wiki/Source_control_management) . Torvalds explica, [[47]](https://en.wikipedia.org/wiki/Git#cite_note-47)

En muchos sentidos, puedes ver a git como un sistema de archivos, es accesible al [contenido](https://en.wikipedia.org/wiki/Content-addressable_storage) , y tiene una noción de control de versiones, pero realmente lo diseñé viniendo al problema desde el punto de vista de una persona del *sistema* de *archivos* (hey, kernels es lo que hago ), y realmente no tengo *ningún* interés en crear un sistema SCM tradicional.

A partir de este enfoque de diseño inicial, Git ha desarrollado el conjunto completo de funciones que se espera de un SCM tradicional, [[29]](https://en.wikipedia.org/wiki/Git#cite_note-bare_url-29) con características que en su mayoría se crean según sea necesario, luego se refinan y amplían con el tiempo.

[](https://en.wikipedia.org/wiki/File:Git_operations.svg)

Algunos flujos de datos y niveles de almacenamiento en el sistema de control de revisiones de Git.

Git tiene dos [estructuras de datos](https://en.wikipedia.org/wiki/Data_structure) : un *índice* mutable (también llamado *etapa* o *caché* ) que almacena en caché la información sobre el directorio de trabajo y la próxima revisión que se comprometerá; y una *base de datos de objetos* anexa e inmutable *.*

El índice sirve como punto de conexión entre la base de datos de objetos y el árbol de trabajo.

La base de datos de objetos contiene cuatro tipos de objetos:

* Un *blob* ( [objeto binario grande](https://en.wikipedia.org/wiki/Binary_large_object) ) es el contenido de un [archivo](https://en.wikipedia.org/wiki/Computer_file) . Las blobs no tienen un nombre de archivo adecuado, marcas de tiempo u otros metadatos. (El nombre de un blob internamente es un hash de su contenido).
* Un objeto de *árbol* es el equivalente de un directorio. Contiene una lista de nombres de archivos, cada uno con algunos bits de tipo y una referencia a un objeto blob o árbol que es ese archivo, enlace simbólico o contenido del directorio. Estos objetos son una instantánea del árbol fuente. (En conjunto, esto comprende un [Merkle Tree](https://en.wikipedia.org/wiki/Merkle_Tree) , lo que significa que solo un solo hash para el árbol raíz es suficiente y realmente se utiliza en commits para identificar exactamente el estado exacto de las estructuras de árbol completo de cualquier cantidad de subdirectorios y archivos).
* Un objeto de *compromiso* vincula objetos de árbol en un historial. Contiene el nombre de un objeto de árbol (del directorio de origen de nivel superior), una marca de tiempo, un mensaje de registro y los nombres de cero o más objetos de confirmación primaria.
* Un objeto de *etiqueta* es un contenedor que contiene una referencia a otro objeto y puede contener metadatos agregados relacionados con otro objeto. Más comúnmente, se usa para almacenar una [firma digital](https://en.wikipedia.org/wiki/Digital_signature) de un objeto de confirmación correspondiente a una versión particular de los datos que rastrea Git.

Cada objeto se identifica por un SHA-1 [de hash](https://en.wikipedia.org/wiki/Cryptographic_hash_function) de su contenido. Git calcula el hash y usa este valor para el nombre del objeto. El objeto se coloca en un directorio que coincida con los dos primeros caracteres de su hash. El resto del hash se usa como el nombre de archivo para ese objeto.

Git almacena cada revisión de un archivo como un blob único. Las relaciones entre los blobs se pueden encontrar mediante el examen del árbol y la asignación de objetos. Los objetos recién agregados se almacenan en su totalidad utilizando compresión [zlib](https://en.wikipedia.org/wiki/Zlib) . Esto puede consumir una gran cantidad de espacio en el disco rápidamente, por lo que los objetos se pueden combinar en *paquetes* , que usan la [compresión delta](https://en.wikipedia.org/wiki/Delta_encoding) para ahorrar espacio y almacenar blobs como sus cambios en relación con otros blobs.

Los servidores Git normalmente escuchan en [el puerto TCP](https://en.wikipedia.org/wiki/TCP_and_UDP_port) 9418.

|  |  |
| --- | --- |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Linus Torvalds](https://en.wikipedia.org/wiki/Linus_Torvalds)[[1]](https://en.wikipedia.org/wiki/Git#cite_note-git-initial-commit-1) |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Junio ​​Hamano](https://en.wikipedia.org/wiki/Junio_Hamano) y otros [[2]](https://en.wikipedia.org/wiki/Git#cite_note-git-commits-2) |
| **Versión inicial** | 7 de abril de 2005 ; Hace 12 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 2.16.1 / 22 de enero de 2018 ; Hace 15 días [[3]](https://en.wikipedia.org/wiki/Git#cite_note-3) |
| [**Repositorio**](https://en.wikipedia.org/wiki/Repository_(version_control)) | [github .com / git / git](https://github.com/git/git) |
| **Estado del desarrollo** | Activo |
| **Escrito en** | [C](https://en.wikipedia.org/wiki/C_(programming_language)) , [Shell](https://en.wikipedia.org/wiki/Shell_(computing)) , [Perl](https://en.wikipedia.org/wiki/Perl) , [Tcl](https://en.wikipedia.org/wiki/Tcl) , [Python](https://en.wikipedia.org/wiki/Python_(programming_language))[[4]](https://en.wikipedia.org/wiki/Git#cite_note-4) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [POSIX](https://en.wikipedia.org/wiki/POSIX) : [Linux](https://en.wikipedia.org/wiki/Linux) , [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) ,[macOS](https://en.wikipedia.org/wiki/MacOS) |
| [**Plataforma**](https://en.wikipedia.org/wiki/Computing_platform) | [IA-32](https://en.wikipedia.org/wiki/IA-32) , [x86-64](https://en.wikipedia.org/wiki/X86-64) |
| **Disponible en** | Inglés |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de versiones](https://en.wikipedia.org/wiki/Version_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [GNU GPL](https://en.wikipedia.org/wiki/GNU_General_Public_License) v2 [[5]](https://en.wikipedia.org/wiki/Git#cite_note-license_gpl-5) y [GNU LGPL](https://en.wikipedia.org/wiki/GNU_LGPL) v2.1 [[6]](https://en.wikipedia.org/wiki/Git#cite_note-license_lgpl-6) |
| **Sitio web** | [git-scm .com](https://git-scm.com/) |

2005 **Codeville** es un sistema de [control de revisión distribuido](https://en.wikipedia.org/wiki/Distributed_revision_control) descontinuado . Fue escrito por [Ross Cohen](https://en.wikipedia.org/wiki/Ross_Cohen) usando [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) , con algunos trabajos de diseño realizados por su hermano [Bram Cohen](https://en.wikipedia.org/wiki/Bram_Cohen) . Utiliza un [algoritmo de](https://en.wikipedia.org/wiki/Algorithm)[fusión](https://en.wikipedia.org/wiki/Merge_(revision_control)) innovador llamado "fusión Codeville". Se estaba desarrollando un nuevo algoritmo de fusión llamado "Precise Codeville" o "pcvd" merge. El proyecto ha sido abandonado.

Codeville fue alojado y también utilizado por [BitTorrent, Inc](https://en.wikipedia.org/wiki/BitTorrent_(company)) y Mosuki.

|  |  |
| --- | --- |
| **Codeville** | |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Ross Cohen](https://en.wikipedia.org/wiki/Ross_Cohen) |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 0.8.0 / 13 de julio de 2007 ; Hace 10 años |
| **Estado del desarrollo** | Abandonado |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Multiplataforma](https://en.wikipedia.org/wiki/Cross-platform) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [BSD](https://en.wikipedia.org/wiki/BSD_License) |
| **Sitio web** | [codeville .org](http://codeville.org/) |

**Team Foundation Server** (comúnmente abreviado como **TFS** ) es un producto de [Microsoft](https://en.wikipedia.org/wiki/Microsoft) que proporciona [administración de código fuente](https://en.wikipedia.org/wiki/Revision_control) (con Team Foundation Version Control o [Git](https://en.wikipedia.org/wiki/Git_(software)) ), informes, [gestión de requisitos](https://en.wikipedia.org/wiki/Requirements_management) , [gestión de proyectos](https://en.wikipedia.org/wiki/Project_management) (para [desarrollo de software ágil](https://en.wikipedia.org/wiki/Agile_software_development) y [equipos de cascada](https://en.wikipedia.org/wiki/Waterfall_model) ), compilaciones automatizadas, capacidades de [administración de](https://en.wikipedia.org/wiki/Release_management) laboratorio, [pruebas](https://en.wikipedia.org/wiki/Software_testing) y [liberación](https://en.wikipedia.org/wiki/Release_management) . Cubre todo el [ciclo de vida de la aplicación](https://en.wikipedia.org/wiki/Application_Lifecycle_Management) y habilita [las](https://en.wikipedia.org/wiki/DevOps) capacidades de [DevOps](https://en.wikipedia.org/wiki/DevOps) . [[1]](https://en.wikipedia.org/wiki/Team_Foundation_Server#cite_note-1) TFS puede usarse como back-end para numerosos[entornos de desarrollo integrados](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE), pero está diseñado para [Microsoft Visual Studio](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio) y [Eclipse](https://en.wikipedia.org/wiki/Eclipse_(software)) en todas las plataformas.

|  |  |
| --- | --- |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Microsoft](https://en.wikipedia.org/wiki/Microsoft) |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 2018 |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de versiones](https://en.wikipedia.org/wiki/Version_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Software de prueba](https://en.wikipedia.org/wiki/Trialware) |
| **Sitio web** | [visualstudio .com / tfs](https://visualstudio.com/tfs) |

2006 **Plastic SCM** es una herramienta de [control de versiones distribuidas](https://en.wikipedia.org/wiki/Distributed_version_control) y multiplataforma desarrollada por Códice Software Inc. Está disponible para [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) , [Mac OS X](https://en.wikipedia.org/wiki/Mac_OS_X) , [Linux](https://en.wikipedia.org/wiki/Linux) y otros sistemas operativos. Incluye una herramienta de línea de comandos, una GUI e integración con varios [IDEs](https://en.wikipedia.org/wiki/Integrated_development_environment) .

* Historial completo de archivos e metadatos
* Historial de revisiones para archivos bifurcados, renombrados, movidos, copiados y eliminados
* Renombrar soporte, también a través de sucursales
* Representación gráfica del historial de archivos y directorios a través del árbol de versiones 2D
* Representación gráfica de la evolución del repositorio a través del *explorador de ramificación*[[2]](https://en.wikipedia.org/wiki/Plastic_SCM#cite_note-2)
* Combinación de archivos de texto de tres vías; fusionar el seguimiento y la prevención de la fusión; detección común de ancestros
* [Herramientas de conciliación](https://en.wikipedia.org/wiki/Diff) gráfica de [diferencias](https://en.wikipedia.org/wiki/Diff) , fusiones y fuera de línea / en línea
* Historial de contenido de archivos gráficos e historial de sucursales
* Repositorio centralizado y de acceso controlado con soporte para control de revisión distribuido (ver a continuación)
* [Compromisos atómicos:](https://en.wikipedia.org/wiki/Atomic_commit) el servidor asegura que los conjuntos de cambios se comprometan por completo en el repositorio
* *Estanterías: los* usuarios pueden guardar y restaurar trabajos en curso para cambiar tareas
* Admite archivos ASCII, [Unicode](https://en.wikipedia.org/wiki/Unicode) , binarios, de [enlace simbólico](https://en.wikipedia.org/wiki/Symbolic_link) (en Unix), específicos de Mac y [UTF-16](https://en.wikipedia.org/wiki/UTF-16)
* Soporte para plataformas Mac OS X, Windows y Linux
* Desencadenantes de eventos del lado del servidor y del lado del cliente

|  |  |
| --- | --- |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | Codice Software |
| **Versión inicial** | 3 de julio de 2007 ; Hace 10 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 5.4 / 22 de marzo de 2017 ; Hace 10 meses |
| [**Lanzamiento de vista previa**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 6.0 / 19 de mayo de 2017 |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) ,[Linux](https://en.wikipedia.org/wiki/Linux) , [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) ,[Mac OS X](https://en.wikipedia.org/wiki/Mac_OS_X) |
| [**Plataforma**](https://en.wikipedia.org/wiki/Computing_platform) | [.NET](https://en.wikipedia.org/wiki/.NET_Framework) / [Mono](https://en.wikipedia.org/wiki/Mono_(software)) |
| [**tamaño**](https://en.wikipedia.org/wiki/File_size) | 93 MB (instalador de Windows) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | Propiedad |
| **Sitio web** | [www .plasticscm .com](http://www.plasticscm.com/) |

2007. **Fossil** es un [sistema de control de versiones distribuidas](https://en.wikipedia.org/wiki/Distributed_version_control_system) , [un sistema de seguimiento de errores](https://en.wikipedia.org/wiki/Bug_tracking_system) y [un](https://en.wikipedia.org/wiki/Wiki_software) servidor de [software wiki](https://en.wikipedia.org/wiki/Wiki_software) para usar en [el desarrollo de software](https://en.wikipedia.org/wiki/Software_development) creado por [D. Richard Hipp](https://en.wikipedia.org/wiki/D._Richard_Hipp) . Fossil es un servidor [multiplataforma](https://en.wikipedia.org/wiki/Cross-platform) que se ejecuta en [Linux](https://en.wikipedia.org/wiki/Linux) , derivados de [BSD](https://en.wikipedia.org/wiki/Berkeley_Software_Distribution) , [Mac](https://en.wikipedia.org/wiki/Mac_OS_X) y [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) . Es capaz de realizar control de versiones distribuidas, seguimiento de errores, servicios [wiki](https://en.wikipedia.org/wiki/Wiki) y blogs. El software tiene una [interfaz web](https://en.wikipedia.org/wiki/Web_interface) incorporada , que reduce la complejidad de seguimiento del [proyecto](https://en.wikipedia.org/wiki/Project) y promueve la [conciencia situacional](https://en.wikipedia.org/wiki/Situational_awareness) . Un usuario puede simplemente escribir "fósil ui" desde cualquier extracción y Fossil abre automáticamente el [navegador web](https://en.wikipedia.org/wiki/Web_browser) del usuario en una página que proporciona información detallada sobre el estado y el historial de ese proyecto.

Al ser distribuido, Fossil no requiere un servidor central, aunque la colaboración es más fácil mediante el uso de uno.

El contenido se almacena utilizando una [base de](https://en.wikipedia.org/wiki/SQLite) datos [SQLite](https://en.wikipedia.org/wiki/SQLite) para que las [transacciones](https://en.wikipedia.org/wiki/Database_transaction) sean [atómicas](https://en.wikipedia.org/wiki/Atomic_transaction) incluso si se interrumpe por una pérdida de energía o un bloqueo del sistema. [[3]](https://en.wikipedia.org/wiki/Fossil_(software)#cite_note-3)

Fossil es [un software gratuito](https://en.wikipedia.org/wiki/Free_software) publicado bajo [licencia BSD](https://en.wikipedia.org/wiki/BSD_license)[[4]](https://en.wikipedia.org/wiki/Fossil_(software)#cite_note-4) (con [licencia](https://en.wikipedia.org/wiki/Software_relicensing) de [GPL](https://en.wikipedia.org/wiki/GNU_General_Public_License) anteriormente ).

|  |  |
| --- | --- |
| [**Autor (es) original (es)**](https://en.wikipedia.org/wiki/Software_developer) | [D. Richard Hipp](https://en.wikipedia.org/wiki/D._Richard_Hipp) |
| **Versión inicial** | 2006 ; Hace 12 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 2.3 / 21 de julio de 2017 ; Hace 6 meses [[1]](https://en.wikipedia.org/wiki/Fossil_(software)#cite_note-fossil-changes-1) |
| [**Repositorio**](https://en.wikipedia.org/wiki/Repository_(version_control)) | * <http://www.fossil-scm.org/index.html/dir?ci=tip>   [Edita esto en Wikidata](https://www.wikidata.org/wiki/Q1439431#P1324) |
| **Escrito en** | [C](https://en.wikipedia.org/wiki/C_(programming_language)) , [SQL](https://en.wikipedia.org/wiki/SQL) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Multiplataforma](https://en.wikipedia.org/wiki/Cross-platform) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Sistema de control de versiones distribuidas](https://en.wikipedia.org/wiki/Distributed_version_control_system) ,[sistema de seguimiento de errores](https://en.wikipedia.org/wiki/Bug_tracking_system), [software wiki](https://en.wikipedia.org/wiki/Wiki_software) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Licencia BSD de 2 cláusulas](https://en.wikipedia.org/wiki/BSD_license)[[2]](https://en.wikipedia.org/wiki/Fossil_(software)#cite_note-2) |
| **Sitio web** | [www .fossil-scm .org](https://www.fossil-scm.org/) |

2008. **Rational Team Concert** es una herramienta de colaboración de equipos de desarrollo de software desarrollada por la marca [Rational Software](https://en.wikipedia.org/wiki/Rational_Software) de IBM, que la lanzó por primera vez en 2008. El software está disponible en versiones de cliente, versión web y en la nube. Proporciona un entorno de colaboración que los equipos de desarrollo de software usan para administrar todos los aspectos de su trabajo, como planes, tareas, [control de revisiones](https://en.wikipedia.org/wiki/Revision_control) , [administración de compilación](https://en.wikipedia.org/wiki/Build_management) e informes.

|  |  |
| --- | --- |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Software racional](https://en.wikipedia.org/wiki/Rational_Software) |
| **Versión inicial** | Junio ​​de 2008 ; Hace 9 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 6.0.2 (25 de abril de 2016 ; hace 21 meses ) [[±]](https://en.wikipedia.org/w/index.php?title=Template:Latest_stable_software_release/Rational_Team_Concert&action=edit) |
| [**Lanzamiento de vista previa**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | Non [[±]](https://en.wikipedia.org/w/index.php?title=Template:Latest_preview_software_release/Rational_Team_Concert&action=edit) |
| **Escrito en** | [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) /[JavaScript](https://en.wikipedia.org/wiki/JavaScript) |
| **Disponible en** | Inglés, alemán, español, francés, italiano, japonés, coreano, portugués brasileño, chino tradicional, chino simplificado, checo, húngaro, polaco y ruso. |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Entorno de desarrollo integrado](https://en.wikipedia.org/wiki/Integrated_development_environment) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | IBM [EULA](https://en.wikipedia.org/wiki/Software_license) |
| **Sitio web** | [jazz .net / projects / rational-team-concert](http://jazz.net/projects/rational-team-concert) |

Rational Team Concert se basa en [IBM Jazz](https://en.wikipedia.org/wiki/IBM_Jazz) , una plataforma de tecnología extensible que ayuda a los equipos a integrar tareas en todo el ciclo de vida del software.

Rational Team Concert se basa en una [arquitectura cliente-servidor](https://en.wikipedia.org/wiki/Client%E2%80%93server_model) . Los equipos de desarrollo de software lo usan para rastrear aspectos de su trabajo, como elementos de trabajo, control de fuente, informes y [administración de compilación](https://en.wikipedia.org/wiki/Software_build) en un solo producto. Rational Team Concert se integra con varios otros productos, que incluyen:

* [Subversion](https://en.wikipedia.org/wiki/Apache_Subversion)[[1]](https://en.wikipedia.org/wiki/Rational_Team_Concert#cite_note-1)
* [Git](https://en.wikipedia.org/wiki/Git_(software))[[2]](https://en.wikipedia.org/wiki/Rational_Team_Concert#cite_note-2)
* [Rational ClearCase](https://en.wikipedia.org/wiki/Rational_ClearCase)
* [Rational ClearQuest](https://en.wikipedia.org/wiki/Rational_ClearQuest)
* [JIRA](https://en.wikipedia.org/wiki/Jira_(software))
* [Rational Quality Manager](https://en.wikipedia.org/w/index.php?title=Rational_Quality_Manager&action=edit&redlink=1)
* [Punto Focal Racional](https://en.wikipedia.org/wiki/Rational_Focal_Point)
* [Rational System Architect](https://en.wikipedia.org/wiki/Rational_System_Architect)
* [Centro de calidad de HP](https://en.wikipedia.org/wiki/HP_Quality_Center)
* Rational Build Forge
* [Hudson](https://en.wikipedia.org/wiki/Hudson_(software))
* [Jenkins](https://en.wikipedia.org/wiki/Jenkins_(software))
* [CruiseControl](https://en.wikipedia.org/wiki/CruiseControl)
* [Maven](https://en.wikipedia.org/wiki/Apache_Maven)

Rational Team Concert presenta una interfaz de cliente basada en Eclipse, una interfaz de cliente de Microsoft Visual Studio y una interfaz web. Además, desde la versión 4.0 proporciona una integración de shell de Windows dentro del Explorador de Windows para el control de origen de los archivos dentro del repositorio de Rational Team Concert. Las interfaces de cliente proporcionan un entorno de desarrollo integrado que los desarrolladores utilizan para crear y entregar artefactos. Los usuarios pueden acceder a la interfaz web para administrar servidores y proyectos, acceder a las áreas del proyecto, buscar información del repositorio, actualizar tareas o leer sobre eventos recientes.

2010. **Veracity** es un [sistema de control de versiones distribuidas de](https://en.wikipedia.org/wiki/Distributed_version_control_system)[código abierto](https://en.wikipedia.org/wiki/Open_source) escrito principalmente por **SourceGear LLC** que no solo almacena los artefactos colocados bajo control de versiones en el repositorio, sino también datos asociados para funciones como el [sistema](https://en.wikipedia.org/wiki/Bug_tracking_system) integrado de [seguimiento de errores](https://en.wikipedia.org/wiki/Bug_tracking_system) y la herramienta de [administración de compilación ágil](https://en.wikipedia.org/wiki/Agile_software_development) . [[1]](https://en.wikipedia.org/wiki/Veracity_(software)#cite_note-1)[[2]](https://en.wikipedia.org/wiki/Veracity_(software)#cite_note-vcbe-2) Escrito en C y JavaScript, Veracity se lanzó bajo la [Licencia Apache](https://en.wikipedia.org/wiki/Apache_License) y tiene un repositorio de código disponible públicamente, [[3]](https://en.wikipedia.org/wiki/Veracity_(software)#cite_note-infoq-3) sin embargo, aún está desarrollado en su mayoría por SourceGear con una participación limitada de la comunidad.

|  |  |
| --- | --- |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | SourceGear LLC |
| **Versión inicial** | 18 de octubre de 2010; hace 7 años |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 2.5 / 5 de marzo de 2013 ; Hace 4 años |
| **Escrito en** | [do](https://en.wikipedia.org/wiki/C_(programming_language)) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | [Multiplataforma](https://en.wikipedia.org/wiki/Cross-platform) |
| [**Plataforma**](https://en.wikipedia.org/wiki/Computing_platform) | [CLI](https://en.wikipedia.org/wiki/Common_Language_Infrastructure) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Control de revisión](https://en.wikipedia.org/wiki/Revision_control) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Licencia Apache](https://en.wikipedia.org/wiki/Apache_License) v2 |
| **Sitio web** | [veracity-scm .com](http://veracity-scm.com/) |

2013. A mediados de 2013, Microsoft compró un producto llamado InRelease de InCycle Software. [[22]](https://en.wikipedia.org/wiki/Team_Foundation_Server#cite_note-22) InRelease se incorporó por completo en Team Foundation Server 2013. Esta capacidad complementó los procesos automatizados de compilación y prueba al permitir una verdadera solución de [implementación continua](https://en.wikipedia.org/wiki/Continuous_delivery) . Las herramientas fueron renombradas como "Release Management" para TFS 2013. Las capacidades de Release Management brindan a los equipos la capacidad de realizar un flujo de trabajo controlado (proporcionado por [Windows Workflow Foundation](https://en.wikipedia.org/wiki/Windows_Workflow_Foundation) ) a Dev, Test and Production environments y proporciona paneles para monitorear el progreso de uno o más lanzamientos.

Microsoft ha reconstruido Release Management para Visual Studio Team Services y la versión local de Team Foundation Server con los nuevos cambios en 2015 Update 2. La nueva versión de Release Management aprovecha el navegador web como cliente y se basa en la misma arquitectura de agente que Team Foundation Build. Release Management habilita [las](https://en.wikipedia.org/wiki/DevOps) capacidades de [DevOps](https://en.wikipedia.org/wiki/DevOps) para Team Foundation Server.

2014. **Microsoft Visual Studio** es un [entorno de desarrollo integrado](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) de [Microsoft](https://en.wikipedia.org/wiki/Microsoft) . Se utiliza para desarrollar [programas informáticos](https://en.wikipedia.org/wiki/Computer_program) , así como [sitios web](https://en.wikipedia.org/wiki/Web_site) , [aplicaciones web](https://en.wikipedia.org/wiki/Web_app) , [servicios web](https://en.wikipedia.org/wiki/Web_service) y [aplicaciones móviles](https://en.wikipedia.org/wiki/Mobile_app) . Visual Studio utiliza plataformas de desarrollo de software de Microsoft como [Windows API](https://en.wikipedia.org/wiki/Windows_API) , [Windows Forms](https://en.wikipedia.org/wiki/Windows_Forms) , [Windows Presentation Foundation](https://en.wikipedia.org/wiki/Windows_Presentation_Foundation) , [Windows Store](https://en.wikipedia.org/wiki/Windows_Store) y [Microsoft Silverlight](https://en.wikipedia.org/wiki/Microsoft_Silverlight) . Puede producir [código nativo](https://en.wikipedia.org/wiki/Native_code) y [código administrado](https://en.wikipedia.org/wiki/Managed_code) .

Visual Studio incluye un [editor de código que](https://en.wikipedia.org/wiki/Code_editor) admite [IntelliSense](https://en.wikipedia.org/wiki/IntelliSense) (el componente de [finalización de código](https://en.wikipedia.org/wiki/Code_completion) ), así como la [refactorización de código](https://en.wikipedia.org/wiki/Code_refactoring) . [El depurador integrado](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio_Debugger) funciona como un depurador de nivel de origen y un depurador de nivel de máquina. Otras herramientas incorporadas incluyen un [generador de perfiles de código](https://en.wikipedia.org/wiki/Profiling_(computer_programming)) , diseñador de formularios para crear aplicaciones [GUI](https://en.wikipedia.org/wiki/GUI) , [diseñador web](https://en.wikipedia.org/wiki/Web_designer) , diseñador de [clases](https://en.wikipedia.org/wiki/Class_(computing)) y diseñador de [esquema de bases de datos](https://en.wikipedia.org/wiki/Database_schema) . Acepta plug-ins que mejoran la funcionalidad en casi todos los niveles, incluida la adición de soporte para sistemas de [control de origen](https://en.wikipedia.org/wiki/Source_control) (como [Subversion](https://en.wikipedia.org/wiki/Subversion_(software))) y agregar nuevos conjuntos de herramientas como editores y diseñadores visuales para [idiomas específicos](https://en.wikipedia.org/wiki/Domain-specific_language) o conjuntos de herramientas para otros aspectos del [ciclo de vida de desarrollo de software](https://en.wikipedia.org/wiki/Software_development_lifecycle) (como el cliente [Team Foundation Server](https://en.wikipedia.org/wiki/Team_Foundation_Server) : Team Explorer).

Visual Studio admite 36 [lenguajes de programación](https://en.wikipedia.org/wiki/Programming_language) diferentes y permite que el editor de código y el depurador admitan (en diversos grados) casi cualquier lenguaje de programación, siempre que exista un servicio específico para el idioma. Los lenguajes incorporados incluyen [C](https://en.wikipedia.org/wiki/C_(programming_language)) , [[6]](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-6) [C ++](https://en.wikipedia.org/wiki/C%2B%2B) , [C ++ / CLI](https://en.wikipedia.org/wiki/C%2B%2B/CLI) , [Visual Basic .NET](https://en.wikipedia.org/wiki/Visual_Basic_.NET) , [C #](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)) , [F #](https://en.wikipedia.org/wiki/F_Sharp_(programming_language)) , [[7]](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-7) [JavaScript](https://en.wikipedia.org/wiki/JavaScript) , [TypeScript](https://en.wikipedia.org/wiki/TypeScript) , [XML](https://en.wikipedia.org/wiki/XML) , [XSLT](https://en.wikipedia.org/wiki/XSLT), [HTML](https://en.wikipedia.org/wiki/HTML) y [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) . Soporte para otros lenguajes como [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) , [[8]](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-8) [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)) , [Node.js](https://en.wikipedia.org/wiki/Node.js), y [M](https://en.wikipedia.org/wiki/MUMPS) entre otros está disponible a través [de complementos](https://en.wikipedia.org/wiki/Plug-in_(computing)) . [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) (y [J #](https://en.wikipedia.org/wiki/J_Sharp) ) fueron compatibles en el pasado.

La edición más básica de Visual Studio, la edición de la comunidad, está disponible de forma gratuita.

|  |  |
| --- | --- |
| [**Desarrollador (es)**](https://en.wikipedia.org/wiki/Software_developer) | [Microsoft](https://en.wikipedia.org/wiki/Microsoft) |
| [**Lanzamiento estable**](https://en.wikipedia.org/wiki/Software_release_life_cycle) | 2017 (7 de marzo de 2017 , hace 10 meses ) [[±]](https://en.wikipedia.org/w/index.php?title=Template:Latest_stable_software_release/Microsoft_Visual_Studio&action=edit)[[1]](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-1) |
| **Escrito en** | [C ++](https://en.wikipedia.org/wiki/C%2B%2B) y [C #](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))[[2]](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-2) |
| [**Sistema operativo**](https://en.wikipedia.org/wiki/Operating_system) | * [Windows 7 SP1](https://en.wikipedia.org/wiki/Windows_7_SP1) y posterior * [Windows Server 2008 R2 SP1](https://en.wikipedia.org/wiki/Windows_Server_2008_R2_SP1)y posterior [[3]](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-3) |
| **Disponible en** | Chino, checo, inglés, francés, alemán, italiano, japonés, coreano, polaco, portugués (Brasil), ruso, español y turco[[4]](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-4) |
| [**Tipo**](https://en.wikipedia.org/wiki/Software_categories#Broad_categories) | [Entorno de desarrollo integrado](https://en.wikipedia.org/wiki/Integrated_development_environment) |
| [**Licencia**](https://en.wikipedia.org/wiki/Software_license) | [Freemium](https://en.wikipedia.org/wiki/Freemium)[[5]](https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-5) |
| **Sitio web** | [visualstudio .com](https://visualstudio.com/) |