

**1970s**

[Computer Associates](https://en.wikipedia.org/wiki/Computer_Associates)**Panvalet®** (también conocido como **CA-Panvalet** ) es un [control de revisión](https://en.wikipedia.org/wiki/Revision_control) y un sistema de administración de código fuente para computadoras mainframe como [IBM System z](https://en.wikipedia.org/wiki/IBM_System_z) e [IBM System / 370 que](https://en.wikipedia.org/wiki/IBM_System/370) 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" \o "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" \o "Tecnología Platinum) , 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)" \o "C (lenguaje de programación))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" \o "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" \o "Sistema UNIX III) y [System V de](https://en.wikipedia.org/wiki/UNIX_System_V" \o "Sistema UNIX 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" \o "BitKeeper)* y *[TeamWare](https://en.wikipedia.org/wiki/TeamWare" \o "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" \o "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" \o "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" \o "Gestión del ciclo de vida de la aplicación) de CA mainframe . También compite contra otra herramienta de gestión de código fuente de CA, [Panvalet](https://en.wikipedia.org/wiki/Panvalet" \o "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" \o "IBM Tivoli Framework (la página no existe)) para z / OS y [WebSphere](https://en.wikipedia.org/wiki/WebSphere" \o "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" \o "Sistema de versiones concurrentes) (CVS) y [Subversion](https://en.wikipedia.org/wiki/Subversion_(software)" \o "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)" \o "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" \o "Sage Software) del Reino Unido), [Intersolv](https://en.wikipedia.org/w/index.php?title=Intersolv&action=edit&redlink=1" \o "Intersolv (la página no existe)) 1992, [Micro Focus International](https://en.wikipedia.org/wiki/Micro_Focus_International) 1998 y [Merant PLC](https://en.wikipedia.org/wiki/Merant_PLC" \o "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" \o "Socios de Silver Lake) 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" \o "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" \o "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" \o "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)" \o "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)" \o "Subversion (software)) y [Perforce](https://en.wikipedia.org/wiki/Perforce" \o "Forzosamente) . 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" \o "BitKeeper) , diseñado por McVoy, comparte una serie de conceptos de diseño con TeamWare anterior.