¿Qué son repositorios?

Un **repositorio**, **depósito** o **archivo** es un sitio centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.

**Características generales**

Los datos almacenados en un repositorio pueden distribuirse a través de una [red informática](http://es.wikipedia.org/wiki/Red_inform%C3%A1tica), como [Internet](http://es.wikipedia.org/wiki/Internet), o de un medio físico, como un [disco compacto](http://es.wikipedia.org/wiki/Disco_compacto). Pueden ser de acceso público o estar protegidos y necesitar de una autentificación previa. Los repositorios más conocidos son los de carácter académico e institucional. Los repositorios suelen contar con [sistemas de respaldo](http://es.wikipedia.org/wiki/Copia_de_seguridad) y mantenimiento preventivo y correctivo, lo que hace que la información se pueda recuperar en el caso que la máquina quede inutilizable. A esto se lo conoce como preservación digital,[1](http://es.wikipedia.org/wiki/Repositorio#cite_note-1) y requiere un exhaustivo trabajo de control de calidad e integridad para realizarse correctamente.

Depositar no debe confundirse con publicar. El depósito en los repositorios es una manera de comunicar públicamente los trabajos de los investigadores, aumentando su difusión: los autores ponen disponibles en acceso abierto una **versión** de los artículos que han publicado en revistas, tradicionales o de acceso abierto.Para ello, los sistemas de repositorios suelen integrarse e inter operar con otros sistemas y aplicaciones web. Asimismo, los repositorios cumplen un rol importante en la formación universitaria.

Algunas instituciones promueven el uso de sus repositorios como un servicio adicional para el investigador. Otras instituciones poseen mandatos propios que obligan a los autores o investigadores a depositar sus publicaciones (o determinados tipos, como por ej. tesis doctorales) en el repositorio institucional, con fines de visibilidad, impacto y preservación. En algunos países, como por ejemplo [Argentina](http://es.wikipedia.org/wiki/Argentina), se han promulgado leyes de acceso abierto que promueven la implementación y uso de los repositorios de instituciones sustentadas con fondos públicos, mientras que otros países están trabajando en la aprobación de leyes similares, como por ejemplo México.

**Software**

La elección del software es una cuestión crucial para la implementación de un depósito de objetos digitales. Existen distintos modelos de tecnología según su origen y forma de adquisición: gratuito o comercial, [software propietario](http://es.wikipedia.org/wiki/Software_propietario) o de [código abierto](http://es.wikipedia.org/wiki/C%C3%B3digo_abierto), [modelo de servicio de software](http://es.wikipedia.org/wiki/Software_como_servicio). En cualquier caso, deben cumplir con los siguientes requisitos:

* Apoyo a los diferentes formatos de archivo, escalabilidad, extensibilidad y mantenimiento del sistema.
* Aceptación de estándares de [metadatos](http://es.wikipedia.org/wiki/Metadatos), descriptivos, de conservación, administrativos.
* Interoperabilidad: cumplir con los principales protocolos de intercambio de registros de información ( [OAI-PMH](http://es.wikipedia.org/wiki/OAI-PMH), [Z39.50](http://es.wikipedia.org/wiki/Z39.50) ).
* Localización permanente de los documentos, mediante la incorporación de identificadores persistentes de objetos digitales como [DOI](http://es.wikipedia.org/wiki/DOI), [Handle](http://es.wikipedia.org/wiki/Handle).
* Aplicaciones de búsqueda y visualización de metadatos.
* Interfaz de búsqueda a texto completo.
* Autenticación y autorización de usuarios.
* Personalización del software ([API](http://es.wikipedia.org/wiki/API)).

Algunos de los productos más conocidos de software para repositorios institucionales son:

* Bepress (software comercial, pago de licencia y honorarios de suscripción).
* CONTENTdm (software comercial, desarrollado por la [OCLC](http://es.wikipedia.org/wiki/OCLC)).
* [DSpace](http://es.wikipedia.org/wiki/DSpace) (software gratuito, de código abierto desarrollado por el [MIT](http://es.wikipedia.org/wiki/MIT) y Hewlett Packard Labs).
* Eprints(gratuito, de código abierto desarrollado por la University of Southampton).
* [Greenstone](http://es.wikipedia.org/wiki/Greenstone_(software)) ( software gratuito y multilingüe de código abierto, bajo licencia según el [GNU](http://es.wikipedia.org/wiki/GNU) General Public Licence).
* Open Repository (software comercial, servicio de establecimiento y mantenimiento desarrollado por [BioMed Central](http://es.wikipedia.org/wiki/BioMed_Central)).

**Marco legal**[[editar](http://es.wikipedia.org/w/index.php?title=Repositorio&action=edit&section=4)]

Las obras incluidas en un repositorio deben cumplir con las leyes vigentes sobre los derechos que los creadores tienen de su obra. Se consideran dos tipos de derechos: morales y [patrimoniales](http://es.wikipedia.org/wiki/Derechos_patrimoniales). Los derechos morales son permanentes, irrenunciables, inexpropiables y no prescriben. Los derechos patrimoniales o de copyright son económicos, transferibles y de duración limitada en el tiempo. Los derechos patrimoniales, generalmente, son de explotación. Este derecho, suele cederse a terceros mediante la firma de un contrato. La cesión total o parcial de este derecho puede ser de cuatro tipos: reproducción, distribución, comunicación pública y transformación. Las instituciones académicas tienen que articular las condiciones legales de los repositorios contemplando los derechos de explotación, de depósito y de acceso a los contenidos. Estos aspectos pueden estar afectados por las cesiones de derechos de explotación que los investigadores han aceptado en los contratos editoriales.

Muchos autores ceden todos los derechos a las editoriales que publican sus trabajos. Esto significa que éste pierde la propiedad de explotación hasta la conclusión del contrato firmado (excepcionalmente el autor podría utilizarlo con fines didácticos o para uso personal). Las consecuencias de esta cesión exclusiva son múltiples y afectan de modo considerable a la publicación, distribución y usos de los trabajos. Hay opciones contractuales alternativas a la **cesión exclusiva**, como son la **cesión parcial** (en la que se establecen unos derechos para el autor y otros -como la publicación o la distribución-, para el editor) o la **no cesión** (en la que el autor retiene el copyright pero otorga al editor el permiso, mediante licencia, para publicar la obra).

Entre las diferentes alternativas a la cesión total y exclusiva de los derechos de explotación de una obra se encuentran las siguientes:

* Publicación en revistas de acceso abierto (un listado exhaustivo se encuentra en [DOAJ](http://es.wikipedia.org/wiki/DOAJ)).
* Publicación en revistas de suscripción que permiten el auto-archivo en repositorios abiertos.
* Publicación en revistas de suscripción que no exigen una cesión exclusiva.
* Enmienda de la licencia del editor.
* Optar por una licencia alternativa (por ejemplo [Creative Commons](http://es.wikipedia.org/wiki/Creative_Commons), Licencias abiertas Colorluris,Open Data Commons).

SUBVERSION

**Subversion (SVN)** es una herramienta de control de versiones open source basada en un repositorio cuyo funcionamiento se asemeja enormemente al de un sistema de ficheros. Es software libre bajo una licencia de tipo Apache/BSD.

Utiliza el concepto de revisión para guardar los cambios producidos en el repositorio. Entre dos revisiones sólo guarda el conjunto de modificaciones (delta), optimizando así al máximo el uso de espacio en disco. SVN permite al usuario crear, copiar y borrar carpetas con la misma flexibilidad con la que lo haría si estuviese en su disco duro local. Dada su flexibilidad, es necesaria la aplicación de buenas prácticas para llevar a cabo una correcta gestión de las versiones del software generado.

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas [computadoras](http://es.wikipedia.org/wiki/Computadora). A cierto nivel, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones. Y puesto que el trabajo se encuentra bajo el control de versiones, no hay razón para temer por que la calidad del mismo vaya a verse afectada —si se ha hecho un cambio incorrecto a los datos, simplemente deshaga ese cambio.

### Ventajas

* Se sigue la historia de los archivos y directorios a través de copias y renombrados..
* Las modificaciones (incluyendo cambios a varios archivos) son [atómicas](http://es.wikipedia.org/wiki/Atomicidad).
* La creación de ramas y etiquetas es una operación más eficiente. Tiene coste de complejidad constante (O(1)) y no lineal (O(n)) como en CVS.
* Se envían sólo las diferencias en ambas direcciones (en CVS siempre se envían al servidor archivos completos).
* Puede ser servido mediante [Apache](http://es.wikipedia.org/wiki/Servidor_HTTP_Apache), sobre [WebDAV](http://es.wikipedia.org/wiki/WebDAV)/[DeltaV](http://es.wikipedia.org/w/index.php?title=DeltaV&action=edit&redlink=1). Esto permite que clientes WebDAV utilicen Subversion de forma transparente.
* Maneja eficientemente archivos binarios (a diferencia de CVS que los trata internamente como si fueran de texto).
* Permite selectivamente el bloqueo de archivos. Se usa en archivos binarios que, al no poder fusionarse fácilmente, conviene que no sean editados por más de una persona a la vez.
* Cuando se usa integrado a Apache permite utilizar todas las opciones que este servidor provee a la hora de autentificar archivos ([SQL](http://es.wikipedia.org/wiki/SQL), [LDAP](http://es.wikipedia.org/wiki/LDAP), [PAM](http://es.wikipedia.org/wiki/Autenticaci%C3%B3n#PAM), etc.).

### Carencias

* El manejo de cambio de nombres de archivos no es completo. Lo maneja como la suma de una operación de copia y una de borrado.
* No resuelve el problema de aplicar repetidamente parches entre ramas, no facilita llevar la cuenta de qué cambios se han realizado. Esto se resuelve siendo cuidadoso con los mensajes de commit.

**Uso y reconocimiento**

Subversión es muy conocido en la comunidad de software libre y se utiliza en muchos proyectos, como:

* [Apache Software Foundation](http://es.wikipedia.org/wiki/Apache_Software_Foundation)
* [Django](http://es.wikipedia.org/wiki/Django_(framework)) (migrado a Git)
* [Free Pascal](http://es.wikipedia.org/wiki/Free_Pascal)
* [FreeBSD](http://es.wikipedia.org/wiki/FreeBSD)
* [GCC](http://es.wikipedia.org/wiki/GNU_Compiler_Collection)
* [KDE](http://es.wikipedia.org/wiki/KDE) (en proceso de migración a Git)
* [Ruby](http://es.wikipedia.org/wiki/Ruby)
* [Tigris.org](http://es.wikipedia.org/w/index.php?title=Tigris.org&action=edit&redlink=1)

Servicios que proporcionan almacenamiento usando Subversion gratuito para proyectos de software libre:

* [SourceForge](http://es.wikipedia.org/wiki/SourceForge)
* [Google Code](http://es.wikipedia.org/wiki/Google_Code)
* [Project Kenai](http://es.wikipedia.org/w/index.php?title=Project_Kenai&action=edit&redlink=1)
* [CodePlex](http://es.wikipedia.org/wiki/CodePlex)
* [Forja de Conocimiento Libre de la Comunidad RedIRIS](http://es.wikipedia.org/wiki/Forja_de_Conocimiento_Libre_de_la_Comunidad_RedIRIS) de [RedIRIS](http://es.wikipedia.org/wiki/RedIRIS)

Subversión también está siendo adoptado en el mundo corporativo. En un informe 2007 de Forrester Research, reconocía a Subversion como el líder destacado en la categoría de [sistema de control de versiones](http://es.wikipedia.org/wiki/Revision_Control_System).

**MERCURIAL**

**Mercurial** es un [sistema de control de versiones](http://es.wikipedia.org/wiki/Sistema_de_control_de_versiones) [multiplataforma](http://es.wikipedia.org/wiki/Multiplataforma), para [desarrolladores de software](http://es.wikipedia.org/wiki/Desarrollador_de_software). Está implementado principalmente haciendo uso del [lenguaje de programación](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [Python](http://es.wikipedia.org/wiki/Python), pero incluye una implementación binaria de [diff](http://es.wikipedia.org/wiki/Diff) escrita en [C](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_C). Mercurial fue escrito originalmente para funcionar sobre Linux. Ha sido adaptado para [Windows](http://es.wikipedia.org/wiki/Microsoft_Windows), [Mac OS X](http://es.wikipedia.org/wiki/Mac_OS_X) y la mayoría de otros sistemas [tipo Unix](http://es.wikipedia.org/wiki/Unix). Mercurial es, sobre todo, un programa para la [línea de comandos](http://es.wikipedia.org/wiki/L%C3%ADnea_de_comandos). Todas las operaciones de Mercurial se invocan como opciones dadas a su programa motor, hg (cuyo nombre hace referencia al [símbolo químico](http://es.wikipedia.org/wiki/S%C3%ADmbolo_qu%C3%ADmico) del [mercurio](http://es.wikipedia.org/wiki/Mercurio_(elemento))).

Las principales metas de desarrollo de Mercurial incluyen un gran rendimiento y escalabilidad; desarrollo completamente distribuido, sin necesidad de un servidor; gestión robusta de archivos tanto de [texto](http://es.wikipedia.org/wiki/Archivo_de_texto) como [binarios](http://es.wikipedia.org/wiki/Archivo_binario); y capacidades avanzadas de ramificación e integración, todo ello manteniendo sencillez conceptual.[1](http://es.wikipedia.org/wiki/Mercurial#cite_note-1) Incluye una interfaz web integrada.

El creador y desarrollador principal de Mercurial es [Matt Mackall](http://es.wikipedia.org/w/index.php?title=Matt_Mackall&action=edit&redlink=1). El [código fuente](http://es.wikipedia.org/wiki/C%C3%B3digo_fuente) se encuentra disponible bajo los términos de la licencia [GNU GPL](http://es.wikipedia.org/wiki/GNU_General_Public_License) versión 2, lo que clasifica a Mercurial como [software libre](http://es.wikipedia.org/wiki/Software_libre).

**Información Técnica**

Al igual que [git](http://es.wikipedia.org/wiki/Git) y [Monotone](http://es.wikipedia.org/wiki/Monotone), Mercurial usa resúmenes [SHA-1](http://es.wikipedia.org/wiki/Secure_Hash_Algorithm) para identificar revisiones.

A diferencia de [git](http://es.wikipedia.org/wiki/Git) es muy fácil romper el repo, solo hace falta apretar ctrl+c en un commit y realizar un push para romper todos los repositorios.[[*cita requerida*](http://es.wikipedia.org/wiki/Wikipedia:Verificabilidad)]

Para el acceso a repositorios mediante red, Mercurial usa un protocolo eficiente, basado en [HTTP](http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol), que persigue reducir el tamaño de los datos a transferir, así como la proliferación de peticiones y conexiones nuevas. Mercurial puede funcionar también sobre [ssh](http://es.wikipedia.org/wiki/Secure_Shell), siendo el protocolo muy similar al basado en HTTP.

**Historia**

Mackall hizo pública la existencia de Mercurial el [19 de abril](http://es.wikipedia.org/wiki/19_de_abril) de [2005](http://es.wikipedia.org/wiki/2005). El estímulo que llevó a esto fue el anuncio de [Bitmover](http://es.wikipedia.org/w/index.php?title=Bitmover&action=edit&redlink=1), publicado anteriormente aquel mismo mes, informando que retirarían la versión gratuita de [BitKeeper](http://es.wikipedia.org/wiki/BitKeeper).

Se había estado usando BitKeeper debido a los requisitos de control de versiones del proyecto del núcleo [Linux](http://es.wikipedia.org/wiki/Linux_(n%C3%BAcleo)). Mackall decidió escribir un sistema de control distribuido de versiones como sustituto para usarlo con el núcleo Linux. Este proyecto comenzó aproximadamente al mismo tiempo que otro denominado [git](http://es.wikipedia.org/wiki/Git), iniciado por el propio [Linus Torvalds](http://es.wikipedia.org/wiki/Linus_Torvalds) con objetivos similares.

El proyecto Linux decidió usar [Git](http://es.wikipedia.org/wiki/Git) en lugar de Mercurial. Sin embargo, muchos otros proyectos usan este último.

Algunos de los proyectos que usan el RCS distribuido Mercurial:[4](http://es.wikipedia.org/wiki/Mercurial#cite_note-4)

* [Python](http://es.wikipedia.org/wiki/Python)
* [OpenOffice.org](http://es.wikipedia.org/wiki/OpenOffice.org)
* [Google Code](http://es.wikipedia.org/wiki/Google_Code) Project Hosting on Google Code permite crear repositorios mercurial (además de Subversion)
* [Aldrin](http://es.wikipedia.org/wiki/Aldrin)
* [Audacious](http://es.wikipedia.org/wiki/Audacious_Media_Player)
* [CubicWeb](http://es.wikipedia.org/wiki/CubicWeb)
* [Dovecot](http://es.wikipedia.org/wiki/Dovecot) [IMAP](http://es.wikipedia.org/wiki/IMAP) server
* [GNU Health](http://meta.wikimedia.org/wiki/w:en:GNU_Health)
* [GNU Octave](http://es.wikipedia.org/wiki/GNU_Octave)
* [Growl](http://es.wikipedia.org/wiki/Growl_(software))
* [MoinMoin](http://es.wikipedia.org/wiki/MoinMoin) wiki software
* [Mozilla](http://es.wikipedia.org/wiki/Mozilla_Firefox)
* [Netbeans](http://es.wikipedia.org/wiki/Netbeans)
* [OpenJDK](http://es.wikipedia.org/wiki/OpenJDK)
* [SAGE](http://es.wikipedia.org/wiki/Sage)
* [OpenSolaris](http://es.wikipedia.org/wiki/OpenSolaris) de Sun
* Los desarrolladores de [Xen](http://es.wikipedia.org/wiki/Xen)
* Software Opensource de [Oracle](http://es.wikipedia.org/wiki/Oracle) como [Btrfs](http://es.wikipedia.org/wiki/Btrfs)
* [Tuenti](http://es.wikipedia.org/wiki/Tuenti)