

**INSTITUTO TECNOLÓGICO DE CHIHUAHUA CAMPUS II.**

# 

**SVN.**

**UNIDAD I.**

**PERIODO:** ENERO-JUNIO 2022.

**CARRERA:** INGENIERÍA EN SISTEMAS COMPTACIONALES.

**ALUMNOS:** ARIANNA SARAHI MONTAÑO PÉREZ - 19550798

ANGELA JACKELINE ANDREW MARRUFO - 19550754

ÁNGEL OMAR VEGA PAEZ - C19550763

LUIS ALONSO ESCALANTE ANDUJO - 19550822

**GRUPO:** A

**MATERIA:** PROGRAMACIÓN WEB FRONT END.

**MAESTRO:** CARLOS HUMBERTO RUBIO RASCÓN.

**FECHA:** 22/FEBRERO/2022.

# **ÍNDICE.**

CONTENIDO.

[**ÍNDICE.** 1](#_Toc96465931)

[**INTRODUCCIÓN** 2](#_Toc96465932)

[**CONCEPTOS BÁSICOS.** 3](#_Toc96465933)

[EL REPOSITORIO. 3](#_Toc96465934)

[COPIA DE TRABAJO. 3](#_Toc96465935)

[BLOQUEAR-MODIFICAR-DESBLOQUEAR. 3](#_Toc96465936)

[COPIAR-MODIFICAR-FUSIONAR. 4](#_Toc96465937)

[**IMPLEMENTACIÓN.** 5](#_Toc96465938)

[IDENTIFICACIÓN DE IMLEMENTACIÓN DE CONTROL DE VERSIONES. 5](#_Toc96465939)

[**CONCLUSIÓN** 6](#_Toc96465940)

[**REFERENCIAS.** 7](#_Toc96465941)

[TRABAJOS CITADOS. 7](#_Toc96465942)

# **INTRODUCCIÓN**

Es una herramienta de control de versiones, que se basa en repositorios que funcionan prácticamente similar a lo que es un sistema de ficheros. Es software libre bajo una licencia como la de Apache.

Utiliza el concepto de revisión para guardar cambios producidos en un repositorio. Entre dos revisiones solo guarda el conjunto de modificaciones optimizando de forma máxima en el espacio en disco.

Es una herramienta de control de versiones, que se basa en repositorios que funcionan prácticamente similar a lo que es un sistema de ficheros. Es software libre bajo una licencia como la de Apache.

Utiliza el concepto de revisión para guardar cambios producidos en un repositorio. Entre dos revisiones solo guarda el conjunto de modificaciones optimizando de forma máxima en el espacio en disco.

Estos sitios brindan o brindaron almacenamiento usando SVC gratuito para proyectos que son de software libre:

* SourceForge
* Google Code
* CodePlex
* GitHub

# **CONCEPTOS BÁSICOS.**

## EL REPOSITORIO.

Subversion usa una base de datos central que contiene todos los archivos cuyas versiones se controlan y sus respectivas historias. Esta base de datos se conoce como el *repositorio*. El repositorio normalmente yace en un servidor de archivos ejecutando el servidor de Subversion, que provee a pedido el contenido a los clientes de Subversion (como TortoiseSVN, por ejemplo). Si solo puede hacer una copia de seguridad de una sola cosa, hágala del repositorio, ya que es la copia maestra de toda su información.

## COPIA DE TRABAJO.

Acá es donde se realiza el trabajo en serio. Cada desarrollador tiene su propia copia de trabajo, comúnmente conocida como caja de arena en su computador local. Usted puede obtener la última versión del repositorio, trabajar en ella localmente sin perjudicar a nadie, y cuando esté feliz con los cambios que ha realizado puede confirmar sus cambios en el repositorio.

Una copia de trabajo de Subversion no contiene la historia de un proyecto, pero sí contiene una copia de los archivos que existían en el repositorio antes que comience a hacer cambios. Esto significa que es fácil verificar qué cambios ha realizado.

## BLOQUEAR-MODIFICAR-DESBLOQUEAR.

Muchos sistemas de control de versiones utilizan un modelo bloquear-modificar-desbloquear para enfrentarse al problema, lo cual es una solución muy simple. En estos sistemas, el repositorio sólo permite que una persona cambie un archivo. Harry primero debe bloquear el archivo antes que pueda empezar a hacer cambios en él.

Bloquear un archivo se parece mucho a tomar prestado un libro de la biblioteca; si Harry ha bloqueado un archivo, entonces Sally no puede hacer ningún cambio en él. Si ella intenta bloquear el archivo, el repositorio le denegará la petición. Todo lo que ella puede hacer es leer el archivo, y esperar a que Harry termine sus cambios y libere su bloqueo. Después que Harry desbloquee el archivo, se acabó su turno, y ahora es el turno de Sally para bloquear y editar.

## COPIAR-MODIFICAR-FUSIONAR.

Subversion, CVS y otros sistemas de control de versiones usan un modelocopia-modificación-fusión como una alternativa de bloqueo. En este modelo, cada cliente de los usuarios lee el repositorio y crea una copia de trabajo personal del fichero o proyecto. Los usuarios trabajan en paralelo, modificando sus copias privadas. Finalmente, las copias privadas son unificadas conjuntamente en una nueva, versión final. El sistema de control de versiones normalmente asesora con la unificación, pero en última instancia un humano es el responsable de hacer esta acción de manera correcta.

Aquí hay un ejemplo. Digamos que tanto Harry como Sally crean copias de trabajo del mismo proyecto, copiado del repositorio. Ellos trabajan simultáneamente, y hacen cambios al mismo archivo A dentro de sus copias. Sally es la primera en grabar sus cambios en el repositorio. Cuando Harry intenta grabar sus cambios más tarde, el repositorio le informa que su archivo A está desactualizado. En otras palabras, que el archivo A en el repositorio ha cambiado de alguna forma desde la última vez que lo copió. Por lo que Harry le pide a su cliente que fusione cualquier nuevo cambio del repositorio dentro de su copia de trabajo del archivo A. Lo más seguro es que los cambios de Sally no se superpongan a los suyos; por lo que una vez que ambos conjuntos de cambios se han integrado, él graba su copia de trabajo de nuevo en el repositorio.

# **IMPLEMENTACIÓN.**

Las empresas con procesos de fabricación que implementan gran volumen de maquinaria tienen hoy en día que enfrentar varios desafíos, incluidos: producción de lotes pequeños, alta variabilidad en la producción de distintos tipos de productos y una cartera de productos cambiante durante el ciclo de vida de una máquina o planta. Por lo tanto, los sistemas de producción deben admitir cambios no solo en su diseño físico, sino también en actualizaciones técnicas exhaustivas ya que los ciclos de vida de un sistema o estación de producción pueden durar hasta 50 años.

En la mayoría de los casos las empresas no tienen suficiente conocimiento sobre las variantes y versiones de software existentes dentro de su propia empresa. Además, por cuestiones de presiones de tiempo de liberación de productos en la manufactura se recurre al típico: “Si ya jala, no le muevas”. Es imprescindible llevar un control de versiones de software de todas las estaciones de prueba y medición de las líneas de manufactura de tu empresa.

## IDENTIFICACIÓN DE IMPLEMENTACIÓN DE CONTROL DE VERSIONES.

¿Como identificar cuando se necesita implementar un control de versiones?

1. Número de desarrolladores. Si tu equipo de desarrollo cuenta con más de dos Ingenieros involucrados en el desarrollo simultaneo del software.
2. Desarrollo concurrente: Se plantea el desarrollo de varias estaciones con enfoque y/o implementación similar.
3. Mantenimiento de software: El propósito del desarrollo es un software que requerirá mantenimiento y liberación de diferentes versiones.
4. Diferentes plataformas para desarrollo de software: LabVIEW, LabWindows/CVI, TestStand, C#, Python, etcétera.
5. Adaptaciones y anexos al código: Se requiere coordinación en cambios y mejoras.
6. Manejo de errores: Se ha depurado código debido a presencia de errores.

# **CONCLUSIÓN.**

Esta herramienta informática es un Software Libre bajo una licencia de tipo Apache/BSD que es muy semejante a un sistema de ficheros nos sirve para el control de versiones, es una herramienta muy útil e interesante ya que nos ayuda a recuperar versiones antiguos y ver nuestro historial de cambios también sirve para guardar y compartir entre varios usuarios múltiples copias de un conjunto de directorios y ficheros en un depósito central , su principal característica como ya mencionamos antes es que mantiene la historia de los cambios y modificaciones que se han realizado sobre ellos a lo largo del tiempo, en este trabajo también nos damos una idea de cómo se implementa o nos ayuda en el ámbito laboral ya que muchas veces no conocemos el software de nuestra empresa y es bueno saber poder darle ese mantenimiento al software.

# **REFERENCIAS.**

# TRABAJOS CITADOS.

LabStudio. (22 de Febrero de 2022). *LabStudio*. Obtenido de LabStudio: https://labitstudio.com/control-de-versiones-de-software-en-manufactura/

Modelos de versionado. (s. f.). *Modelos de versionado.* https://tortoisesvn.net/docs/nightly/TortoiseSVN\_es/tsvn-basics-versioning.html

Conceptos básicos de control de versiones. (s. f.). *Conceptos basicos.* https://tortoisesvn.net/docs/release/TortoiseSVN\_es/tsvn-qs-basics.html

colaboradores de Wikipedia. (2021, 13 febrero). *Subversion* (software). Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Subversion\_(software)

EcuRed. (s. f.). Subversion (software) - *EcuRed.* <https://www.ecured.cu/Subversion_(software)>

*Universidad Carlos III de Madrid. (20017). El controlador de versiones Subversion. svc.*