

UNIVERSIDAD PRIVADA DE TACNA



INGENIERIA DE SISTEMAS

TITULO:

**INFORME DE SISTEMAS DE CONTROL DE
VERSIONES**

CURSO:

BASE DE DATOS II

DOCENTE(ING):

Patrick Cuadros Quiroga

Estudiante:

Mireya Flavia Pilco Quispe

(2015053234)

Índice

1. Actividad No 01 – Introduccion a los Sistemas de Control de Versiones	1
2. Actividad No 02 – Tipos de Sistemas de Control de Versiones	2
3. Actividad No 03 – Sistemas de Control de Versiones Libres	4

1. Actividad No 01 – Introduccion a los Sistemas de Control de Versiones

¿Que es un Sistema de Control de Versiones?

- Un Sistema de Control de Versiones (en adelante SCV), es un software que controla y organiza las distintas revisiones que se realizen sobre uno o varios documentos.

Una revisión es un cambio realizado sobre un documento, por ejemplo añadir un párrafo, borrar un fragmento o algo similar

EJEMPLO:

Supongamos que cargamos en un SCV el siguiente código fuente:

```
1 | #include <iostream>
2 |
3 | int main() {
4 |     cout << "Hola, mundo!" << endl;
5 | }
```

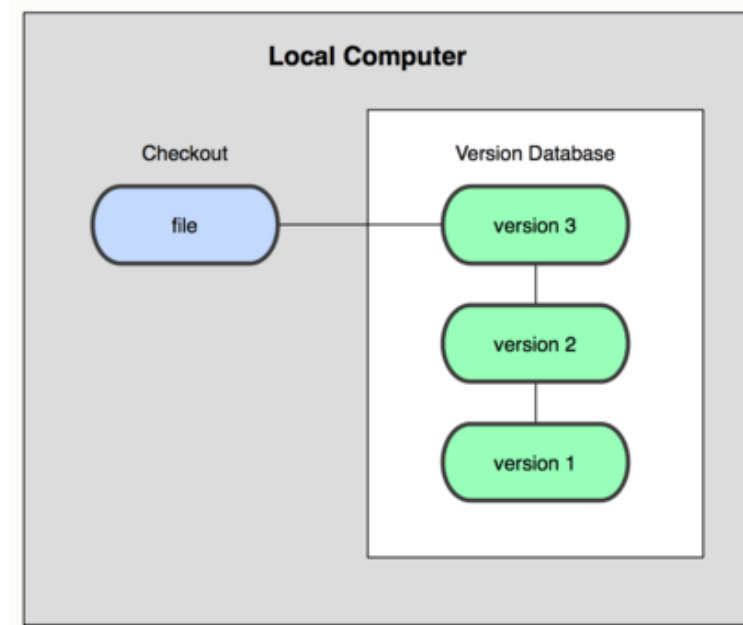
- Se añade al SCV como la revisión 1 del fichero. Una vez añadido, vemos que no compila, ya que nos falta incluir el uso del espacio de nombres, así que lo modificamos:

```
1 | #include <iostream>
2 |
3 | using namespace std;
4 |
5 | int main() {
6 |     cout << "Hola, mundo!" << endl;
7 | }
```

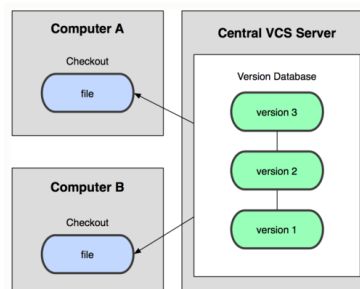
- Se vuelve a añadir al SCV, ahora como la revisión número 2. De esta forma, se guarda el historial de las distintas modificaciones sobre un fichero, por lo que en cualquier momento podemos restaurar la revisión que queramos de un fichero.

2. Actividad No 02 – Tipos de Sistemas de Control de Versiones

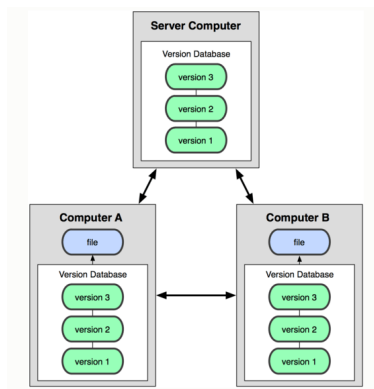
1. Sistemas de Control de versiones Locales



2. Sistemas de Control de Versiones Centralizados



3. Sistemas de Control de Versiones Distribuidos



3. Actividad No 03 – Sistemas de Control de Versiones Libres

1. CVS

CVS ha estado durante mucho tiempo, y muchos desarrolladores están ya familiarizados con él. En su día fue revolucionario: fue el primer sistema de control de versiones de código abierto con acceso a redes de área amplia para desarrolladores, y el primero que ofreció checkouts anónimos de sólo lectura, los que dieron a los desarrolladores una manera fácil de implicarse en los proyectos. CVS sólo versiona ficheros, no directorios; ofrece ramificaciones, etiquetado, y un buen rendimiento en la parte del cliente, pero no maneja muy bien ficheros grandes ni ficheros binarios. Tampoco soporta cambios atómicos.

2. SVK

Aunque se ha construido sobre Subversion, probablemente SVK[9] se parece más a algunos de los anteriores sistemas descentralizados que a Subversión. SVK soporta desarrollo distribuido, cambios locales, mezcla sofisticada de cambios, y la habilidad de reflejar/clonar árboles desde sistemas de control de versiones que no son SVK. Vea su sitio web para más detalles.

3. Mercurial

Mercurial es un sistema de control de versiones distribuido que ofrece, entre otras cosas, una completa indexación cruzada de ficheros y conjuntos de cambios; unos protocolos de sincronización SSH y HTTP eficientes respecto al uso de CPU y ancho de banda; una fusión arbitraria entre ramas de desarrolladores; una interfaz web autónoma integrada; [portabilidad a] UNIX, MacOS X, y Windows más (la anterior lista de características ha sido parafraseada del sitio web de Mercurial).

4. GIT

GIT es un proyecto empezado por Linus Torvalds para manejar el árbol fuente del kernel de Linux. Al principio GIT se enfocó bastante en las necesidades del desarrollo del kernel, pero se ha expandido más allá que eso y ahora es usado por otros proyectos aparte del kernel de Linux. Su página web dice que está "... diseñado para manejar proyectos muy grandes eficaz y velozmente; se usa sobre todo en varios proyectos de código abierto, entre los cuales el más notable es el kernel de Linux. GIT cae en la categoría de herramientas de administración de código abierto distribuido, similar al, por ejemplo, GNU Arch o Monotone (o bitKeeper en el mundo comercial). Cada directorio de trabajo de GIT es un repositorio completo con plenas capacidades de gestión de revisiones, sin depender del acceso a la red o de un servidor central."

5. Bazaar

Bazaar está todavía en desarrollo. Será una implementación del protocolo GNU Arch, mantendrá compatibilidad con el protocolo GNU Arch a medida que evolucione, y trabajará con el proceso de la comunidad GNU Arch para cualquier cambio de protocolo que fuera requerido a favor del agrado del usuario.