UNIVERSIDAD PRIVADA DE TACNA

INGENIERIA DE SISTEMAS

TITULO:

INFORME DE LABORATORIO No 02

CURSO:

BASE DE DATOS II

DOCENTE(ING):

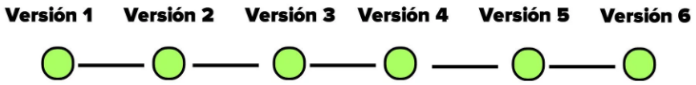
Patrick Cuadros Quiroga

Integrantes:

Lisbeth Espinoza Caso	(2011040667)
Fiorella Rosmery Salamanca Contreras	(2015053237)
Mireya Flavia Pilco Quispe	(2015053234)
Flor de Maria Condori Gutierrez	(2015053227)
Yerson Luis Coaquira Calizaya	(2015053225)

1. Actividad No 01 – PRIMER LATEX AGREGADO

¿Qué buscan los sistemas de control de versiones? Gestionar ágilmente proyectos. Parte de su principal propósito es que puedas regresar a un estado anterior del proyecto o conocer, incluso, toda su evolución en el tiempo. Desde sus inicios hasta donde se encuentra actualizado. Puedes ver a los SCV como máquinas del tiempo, que permiten regresar a cualquier momento que quieras de tu proyecto.



Imagina el proyecto de Mestros del Web. El primer lanzamiento es una versión funcional del sitio web. Pero muy tranquilo. Conforme han avanzado los años, se posiciona y ha llegado a una 6ª versión donde ha mejorado en todas sus perspectivas. Si un desarrollador nuevo quisiera ver el trabajo y la evolución durante todos estos años, si se utilizó Git desde el inicio, lo podrá ver sin ningún problema. Cada versión incluye mejoras en el código, imágenes, organización de carpetas, etc. El repositorio (historial) creado por Git, guarda TODO. Es como ver un libro de tu proyecto. Hagamos otra analogía, tenemos el famoso CTRL + Z. Cuando trabajas en Word, sabemos que hay momentos donde necesitas regresar a un momento anterior (porque te equivocaste, etc.). Su historial temporal permite manejarte ágilmente con los errores. Los SCV persiguen el mismo objetivo. La diferencia es que éstos tienen un ecosistema para que puedas gestionar cada cambio de la mejor manera.

2. Actividad nro 2 - SEGUNDO LATEX

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante. Te permite revertir archivos a un estado anterior, revertir el proyecto entero a un estado anterior, comparar cambios a lo largo del tiempo.

- PERÚ
  Parte del mundial

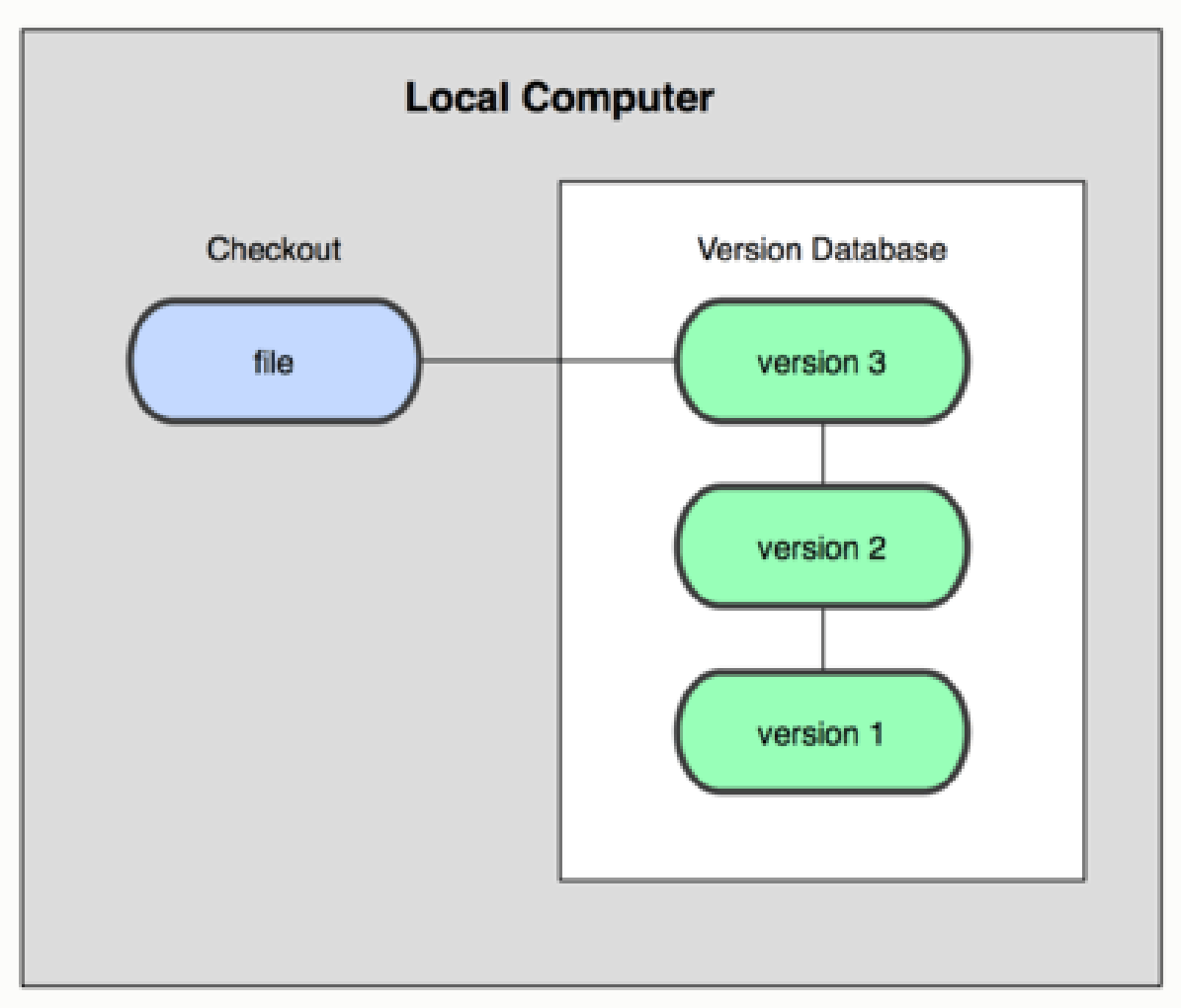


Figura 1: Sistemas de Control de Versiones Locales

3. Actividad No 03 – Documento importantes

1. **Introducción**
   Este es un pequeño trabajo de investigacion para el curso de Base de Datos II. Basicamente el trabajo consta de tres partes: Los objetivos, el Marco Teorico y las Conclusiones.

2. **Marco Teórico**
   El SCV
   Un sistema de control de versiones es una herramienta que registra todos los cambios hechos en uno o más proyectos, guardando así versiones del producto en todas sus fases del desarrollo. Las versiones son como fotografías que registran su estado en ese momento del tiempo y se van guardando a medida que se hacen modificaciones al código fuente.
   Un sistema de control de versiones debe proporcionar:

   * Mecanismo de almacenamiento de los elementos que deba gestionar.
   * Posibilidad de realizar cambios sobre los elementos almacenados (ej. modificaciones parciales, añadir, borrar, renombrar o mover elementos).
   * Registro histórico de las acciones realizadas con cada elemento o conjunto de elementos (normalmente pudiendo volver o extraer un estado anterior del producto).

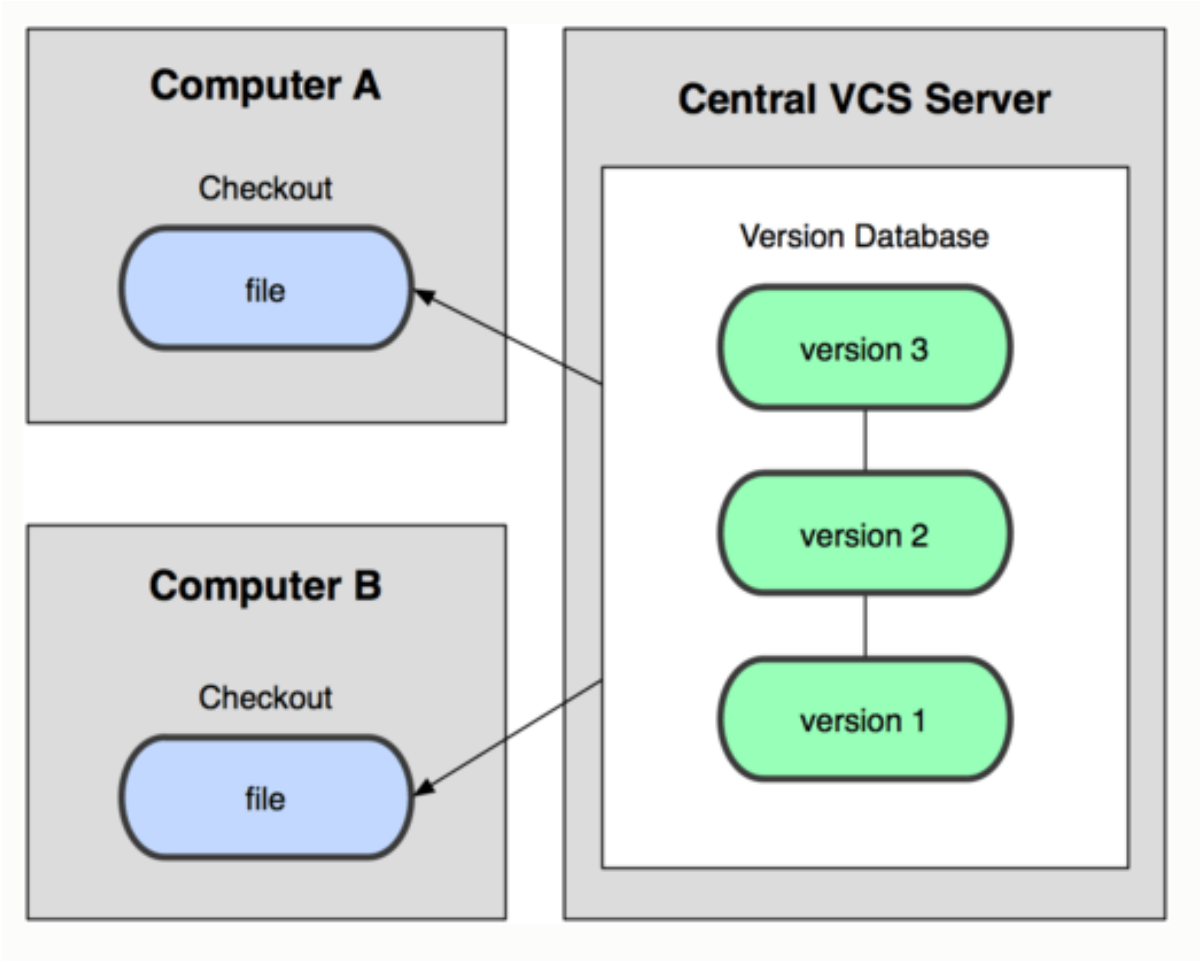


Figura 2: Sistemas de Control de Versiones Centralizados

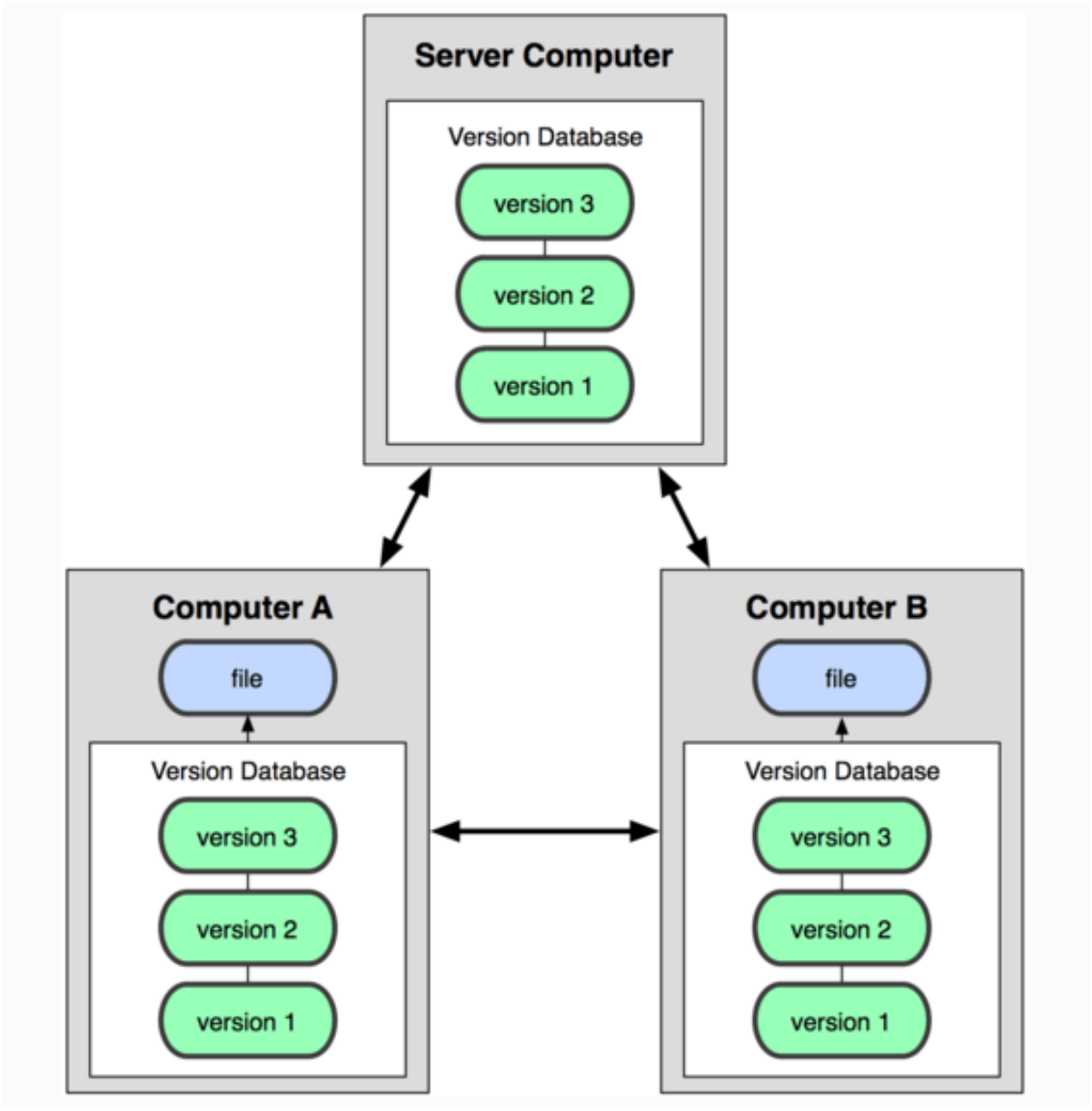


Figura 3: Sistemas de Control de Versiones Distribuidos

4. Actividad No 04 – CUARTO LATEX AGREGADO

Nombre : Fiorella Salamanca Contreras

- **¿Que es GitHub?**

  GitHub es un servicio de alojamiento web para proyectos que utilizan el sistema de control de revisiones Pequeño icono de GitGit . Está escrito en Ruby on Rails por los desarrolladores de Logical Awesome Chris Wanstrath, PJ Hyett y Tom Preston-Werner. GitHub ofrece planes comerciales y cuentas gratuitas para proyectos de código abierto.

  El sitio proporciona funciones de redes sociales como una alimentacion de contenido, seguidores y el gráfico de red para mostrar cómo los desarrolladores trabajan en sus versiones de un repositorio.

  Otros autores lo señalan como una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git,debido a que el código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

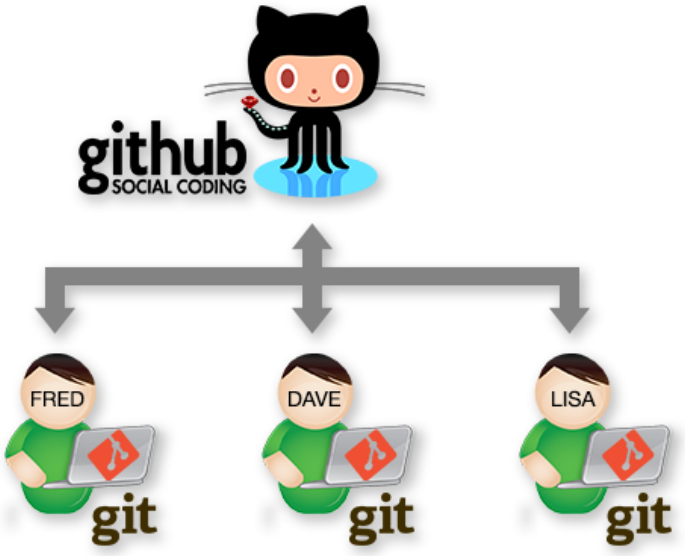
GitHub

- **¿Porque es importante GitHub?**

  GitHub permite alojar en tu repositorio el código de desarrollo de proyecto y te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto.

  Además de eso, puedes contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls.

  Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en tu cuenta), para eliminar algún bug o modificar cosas de él. Una vez realizadas tus modificaciones puedes enviar un pull al dueño del proyecto. Éste podrá analizar los cambios que has realizado fácilmente, y si considera interesante tu contribución, adjuntarlo con el repositorio original.



5. Actividad No 05 – 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

   Mercuria] es un sistemas de control de versiones distribuido que ofrece, entre otras cosas, ünа completa indexación cruzada de ficheros y conjutos de cambios; unosprocotolos 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 Windowsz 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 arbol 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 distribuído, 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 procotolo 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.