



Control de versiones Introducción

Esteban Saavedra Lopez

CEO AtixLibre

CEO Opentelematics

email: esteban.saavedra@atixlibre.org

estebansaavedra@gmail.com

http://esteban.opentelematics.org



Acerca del facilitador?



- Ingeniero de profesión
- Docente Universitario de Postgrado
- Investigador senior en innovación tecnológica.
- Presidente de la Fundación Atixlibre
- CEO Opentelematics Internacional
- Asesor de Tecnologías de la Información de Gobierno



Objetivo

 Realizar una presentación de los de los sistemas de control de versiones, considerando sus principios, sus características y sus tendencias



Contenido

- Elementos ciclo de proyectos
- Sistema de control de versiones
- Características
- Clasificación
- Funcionamiento
- Conceptos importantes

Libre

Elementos - ciclo del proyecto

- Dentro el ciclo de un proyecto existen varios elementos importantes:
 - > Configuraciones del equipo de desarrollo
 - Configuraciones de servidores
 - > Código fuente
 - > Documentación
 - > Traducciones
 - > HelpDesk
- La tendencia es manejar archivos de texto con cierto manejo de tags o estructura de texto (.php, .py, .rb, .xml, .rst, etc)



Sistema de control de versiones

 Un sistema de control de versiones es el encargado de la gestión de los diversos cambios que se realizan sobre los elementos dentro el desarrollo de un producto software o en las configuraciones

del mismo.

```
\label{local_config} $$ $$ $ ndbroadbent@ndb-desktop $$ $$ $$ on branch: $$ master $| +2 | [*] => $e* $$
Changes not staged for commit
        deleted: [1] _shared.sh
       modified: [2] assets/git_breeze/config.example.sh
       modified: [3] assets/git_breeze/config.sh
       modified: [4] assets/git_breeze/git_breeze.sh
       modified: [5] assets/git_breeze/lib/aliases_and_bindings.sh
        deleted: [6] assets/git_breeze/lib/git_file_shortcuts.sh
       modified: [7] assets/git_breeze/lib/git_repo_management.sh
        deleted: [8] assets/git_breeze/test/git_file_shortcuts_test.sh
       modified: [9] assets/git_breeze/test/git_repo_management_test.sh
       modified: [10] assets/git_breeze/test/support/test_helper
➤ Untracked files
      untracked: [11] assets/git_breeze/install.sh
      untracked: [12] assets/git_breeze/lib/fallback/
      untracked: [13] assets/git_breeze/lib/git_status_shortcuts.rb
      untracked: [14] assets/git_breeze/lib/git_status_shortcuts.sh
      untracked: [15] assets/git_breeze/test/git_status_shortcuts_test.sh
ndbroadbent@ndb-desktop ~/src/ubuntu_config [master±|1.9.2p290]$ |
```



Donde se utiliza

- El control de versiones se realiza principalmente en la industria informática para controlar las distintas versiones del código fuente.
- Los mismos conceptos son aplicables a otros ámbitos como documentos, imágenes, sitios web, etc.



Características (1)

- Gestionar el almacenamiento de cada uno de los elementos del proyecto.
- Llevar un historial de los cambios (revisión) en cada elemento del proyecto y anotar el autor de los cambios.
- Posibilidad de añadir, borrar, mover o editar los elementos.



Características (2)

- Capacidad de gestionar ramas de desarrollo paralelas a la principal.
- Gestión de conflictos, en el caso de que más de un usuario cambie un elemento del proyecto.
- Generación de informes de estado, donde se muestren las diferencias entre distintas revisiones.

Libre

Beneficios de hacer uso de un SCV

- Trabajo Personal
- Posibilidad de revertir una revisión previa
- Facilitar la revisión de nuestro propio código (antes del commit)
- Sincronización de múltiples maquinas
- Desarrollo colaborativo
- Un desarrollador puede trabajar sin interferir con el trabajo de otros desarrolladores
- El proceso de fusión es automatizado en su generalidad.



Sistemas de control de versiones









PERFORCE





Clasificación

Centralizados

Existe un repositorio centralizado de todo el código, del cual es responsable un único usuario (o conjunto de ellos). Se facilitan las tareas administrativas a cambio de reducir la potencia y flexibilidad, pues todas las decisiones fuertes (como crear una nueva rama) necesitan la aprobación del responsable.

Distribuidos

Se aumenta la capacidad de decisión distribuida, característica que dará mayor flexibilidad en el uso, pero en algunos casos puede dificultar la sincronización.



Funcionamiento

- Propio o personal
- Equipo
- Colaborativo entre equipos

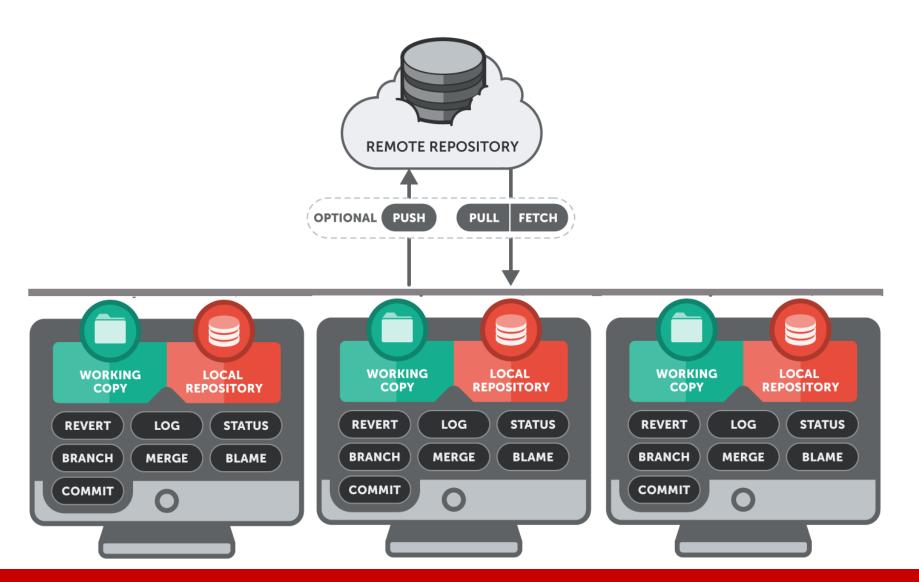


Funcionamiento Personal



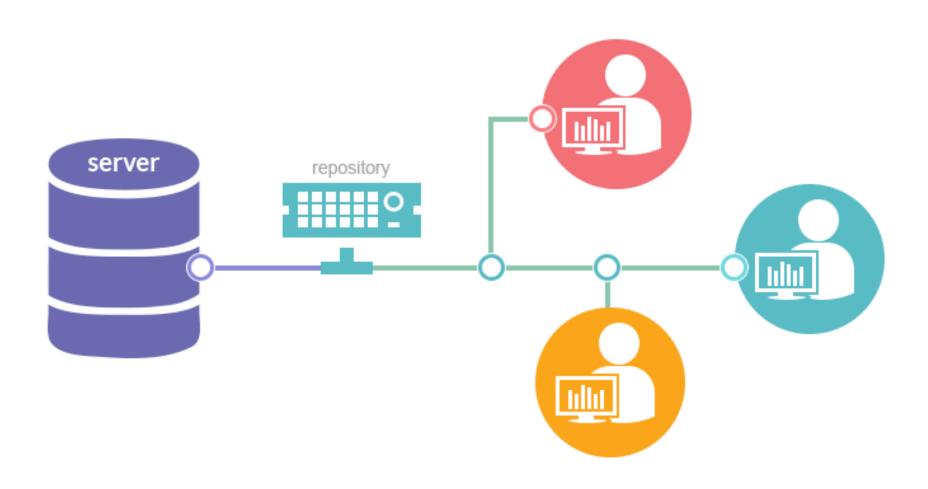


Funcionamiento en Equipo





Funcionamiento colaborativo



Fat III

Conceptos Importantes (1)

Repositorio

Es el lugar en el que se almacenan los datos actualizados e históricos, a menudo un servidor (proceso centralizado), el PC de los desarrolladores (proceso distribuido)

Módulo

Conjunto de directorios y/o archivos dentro del repositorio que pertenecen a un proyecto común.

Rótulo - Tag

Identificador asignado a un módulo o fichero, asignado en un momento determinado, orientado a una búsqueda posterior.

Fundación

Conceptos Importantes (2)

Revisión

> Una revisión es una versión determinada de un archivo.

Línea base

Una revisión aprobada de un fichero, a partir del cual se pueden realizar cambios subsiguientes.

Injertar rama o branch

Un módulo puede ser branched o bifurcado en un momento dado, de forma que, en adelante, dos copias de esos ficheros puedan ser desarrolladas a diferentes velocidades o de diferentes formas, de modo independiente.

Fundación

Conceptos Importantes (3)

Check-out

> Permite crear una copia de trabajo local desde el repositorio. Se puede especificar una revisión específica, por defecto se suele obtener la última.

Commit o check-in

Los cambios realizados localmente son escritos o integrados sobre el repositorio.

Conflicto

Un conflicto ocurre cuando el sistema es incapaz de fusionar los cambios, algunas veces precisa intervención manual.



Conceptos Importantes (4)

Cambio

Un cambio representa una modificación específica a un objeto (fichero) bajo control de versiones.

Lista de cambios

Son listas que identifican una serie de cambios a realizarse en un solo commit. Útil al momento de revisar un cambio a partir de un identificador.

Funda s Libre

Conceptos Importantes (5)

Exportación

Similar a un check-out, crea un árbol de directorios limpio sin los metadatos de control de versiones presentes en la copia de trabajo. Se utiliza a menudo de forma previa a la publicación de los contenidos.

Importación

Una importación es la acción de copia un árbol de directorios local (que no es en ese momento una copia de trabajo) en el repositorio por primera vez.



Conceptos Importantes (6)

Integración o fusión

Una integración o fusión une dos conjuntos de cambios sobre un fichero o un conjunto de ficheros en una revisión unificada de dicho fichero o ficheros.

Actualización

Integra los cambios que han sido hechos en el repositorio (por ejemplo por otras personas) en la copia de trabajo local.

Fat III

Conceptos Importantes (6)

Copia de trabajo

La copia obtenida desde el repositorio en el área de trabajo local, se la realiza en un momento del tiempo o revisión específica

Congelar

Representa permitir los últimos cambios (commits) para solucionar las fallas a resolver en una entrega (release) y suspender cualquier otro cambio antes de una liberación, con el fin de obtener una versión consistente.



Preguntas - Consultas







Control de versiones Introducción

Esteban Saavedra Lopez

CEO AtixLibre

CEO Opentelematics

email: esteban.saavedra@atixlibre.org

estebansaavedra@gmail.com

http://esteban.opentelematics.org