

En el Mundo Real(TM)

En cuanto la complejidad del software aumenta es necesario recurrir a técnicas de Ingeniería de Software para su diseño además es muy probable que para cumplir los tiempos de desarrollo se necesite un equipo de programadores, dbas y Arquitectos de Software.

Los proyectos decentes tienen al menos 4 desarrolladores incluyendo a un dba y un arquitecto de software.

Modelo de Desarrollo Open Source

Todo gira en torno a la disponibilidad del código para mejorarlo o repararlo.

Encontrar una manera de poner a disponibilidad todo el código y además tener control sobre los cambios es prioridad en el mundo del desarrollo de Software.

Manejadores de Versiones

- Se encargan de mantener disponible el código.
- Mantienen la integridad de los datos.
- Nunca se borra nada.

Existen distintos manejadores de versiones respondiendo a algun paradigma en particular:

- CVS (Orientado a Archivos)
- SVN (Orientado a Proyectos)
- Git (Orientado a Versiones y Distribuido)

Existen muchos mas.

Motivos para usar un manejador de versiones

- Persistencia.
- Integración.
- Contabilidad.
- Branching.
- Trabajo Distribuido.

CVS

- Orientado a Archivos.
- Nació con y para el proyecto GNU.
- Cada archivo se “versiona” por separado.
- No permite renombrar archivos.

Comandos Básicos:

- Import, checkout, export, add, commit, log, status, diff, update.

CVS

Login

```
$cvs login
```

Creando un repositorio.

```
$cvs import Proyecto <Algo> <Algo>
```

Consiguiendo el codigo fuente (crea SandBox)

```
$cvs checkout URL:proyecto
```

Consiguiendo el codigo

```
$cvs export URL:Proyecto
```

CVS

- Agregar Directorio

`$cvs add directorio`

- Agregar Archivo

`$cvs add archivo.c && cvs commit archivo.c`

- Modificar Archivo

`$cvs commit archivo.?`

- Estado del proyecto

`$cvs status`

SVN

- Orientado a proyectos.
- Acepta renombrado de archivos y lo agrega al historial.
- Solo envia/recibe las diferencias.
- Se puede integrar a apache2.
- Las modificaciones son atómicas permitiendo funciones transacciones.
- Fue creado como el reemplazo de CVS.

SVN

- Crear Repositorio

```
$svnadmin create --fs-type fsfs /var/svn
```

- Checkout

```
$svn co <Alla> <Aca>
```

- Agregar archivo o directorio.

```
$svn add dir/archivo
```

- Hacer cambios en el proyecto

```
$svn ci -m "No te Enojes Manuel"
```

SVN

- Estado del proyecto.

`$svn st`

- Update.

`$svn update`

- Después de programar Borracho.

`$svn revert`

Git

- Facilidad de Ramificaciones
- Repositorios locales y principales.
- Distribuido.
- Diseñado para proyectos grandes y alto grado de actividad.
- Hace mas eficiente la escritura a costo de hacer mas lenta la lectura.
- Otro monstruo creado por Linus Tolvards

Git

- Iniciando Git
git
- Creando repositorio y agregando archivos
git add Directorio/archivo
- Traer código
git clone URL
- Status
git status

Git

- Commit

`git commit -a -m "Quick backup"`

- Mas información en el manual de git.

Administradores de Proyectos

Son herramientas de software, generalmente web, que permiten coordinar el trabajo, cambios, documentación de los proyectos de desarrollo de software.

Trac

- Escrito en Python.
- Complementa a svn con la administración de proyectos.
- Se basa en Tickets.
- Permite documentar el proyecto por medio paginas Wiki.
- Los tickets se dividen por tipo y urgencia.
- Declara por defectos tres versiones del proyecto: stable, testing, unstable. (codename)

RedMine

- Escrito en Ruby sobre RoR.
- Gestiona varios proyectos simultáneamente
- Es compatible con SVN, Git, CVS, Mercurial, Bazaar y Darcs.
- Soporta LDAP.
- Implementa foros y wikis.
- Soporta múltiples bases de datos.
- Se basa en “issues”.