

**APROXIMACIÓN HACIA UN ESQUEMA DE CALIDAD A TRAVÉS  
DE UN SISTEMA DE CONTROL DE VERSIONES PARA LA  
ADECUADA GESTIÓN DE ARCHIVOS A PARTIR DEL USO DE  
SOFTWARE LIBRE**

Luis Eduardo Sepúlveda Rodríguez

**Universidad Autónoma de Bucaramanga (UNAB)**

**Universidad Oberta de Catalunya (UOC)**

**Bucaramanga, Santander**

**Julio 2010**

**APROXIMACIÓN HACIA UN ESQUEMA DE CALIDAD A TRAVÉS  
DE UN SISTEMA DE CONTROL DE VERSIONES PARA LA  
ADECUADA GESTIÓN DE ARCHIVOS A PARTIR DEL USO DE  
SOFTWARE LIBRE**

Luis Eduardo Sepúlveda Rodríguez

Tesis de grado para optar el título de:

**Magíster en Software Libre**

Director:

**PhD. Eduardo Carrillo Zambrano**

**Universidad Autónoma de Bucaramanga (UNAB)**

**Universidad Oberta de Catalunya (UOC)**

**Bucaramanga, Santander**

**Julio de 2010**

Nota de Aceptación

---

---

---

---

---

Presidente del Jurado

---

Jurado

---

Jurado

---

Bucaramanga, julio de 2010

## DEDICATORIA

*Dedico esta tesis a Dios quien con su infinita  
bondad me ha dado al fuerza para lograr alcanzar  
mis metas; a mi familia que con su apoyo incondicional  
han propiciado mi formación personal y a mi esposa,  
quien ha llenado mi vida de amor y alegría.*

## **AGRADECIMIENTOS**

Deseo expresar mis agradecimientos a:

- Dios quien me ha dado la fuerza y la dedicación para realizar mis sueños.
- A mi familia por estar presentes en cada instante de mi vida brindándome su apoyo y calor de hogar.
- A mi director, el Doctor Eduardo Carrillo Zambrano, quien me orientó y asesoró incondicionalmente.
- A los profesor Carlos Eduardo Gómez, Chritian Candela Uribe y Luis Fernando Londoño, quienes con sus oportunos consejos lograron contribuir en los aspectos técnicos de la tesis.
- Gracias a cada uno de los maestros que participaron en mi desarrollo profesional durante la maestría, sin su ayuda y conocimientos no estaría en donde me encuentro ahora.

# TABLA DE CONTENIDO

<b>1. INTRODUCCIÓN .....</b>	<b>11</b>
<b>2. DESCRIPCIÓN DEL PROBLEMA .....</b>	<b>14</b>
<b>3. OBJETIVOS.....</b>	<b>17</b>
3.1. OBJETIVO GENERAL .....	17
3.2. OBJETIVOS ESPECÍFICOS .....	17
<b>4. MARCO TEÓRICO .....</b>	<b>18</b>
4.1. CALIDAD.....	18
4.2. POLÍTICA ORGANIZACIONAL DE CALIDAD .....	22
4.3. TERMINOLOGÍA SEGÚN ISO 8402.....	23
4.4. NOTACIÓN PARA EL MODELADO DE PROCESO DE NEGOCIO (BPMN): .....	28
4.5. CALIDAD EN EL SOFTWARE.....	29
4.6. ACTIVIDADES PARA EL ASEGURAMIENTO DE CALIDAD DE SOFTWARE .....	30
4.7. MÉTRICAS DE SOFTWARE .....	30
4.8. VERIFICACIÓN Y VALIDACIÓN DEL SOFTWARE .....	31
4.9. MODELOS DE CALIDAD PARA SOFTWARE.....	32
4.10. GESTIÓN DE SERVICIOS DE TECNOLOGÍA INFORMÁTICA .....	34
4.11. ITIL.....	35
4.12. ALGUNOS COMPONENTES DE ITIL .....	36
4.13. SOPORTE AL SERVICIO:.....	36
4.14. PROVISIÓN DEL SERVICIO:.....	38
4.15. GESTIÓN DE LA SEGURIDAD:.....	39
4.16. GESTIÓN DE CONFIGURACIONES:.....	39
4.17. GESTIÓN DE CAMBIOS .....	41
4.18. GESTIÓN DE VERSIONES.....	43
4.19. GESTIÓN DE CONFIGURACIONES DE SOFTWARE .....	57
4.20. TRABAJO RELACIONADO.....	59
<b>5. ESQUEMA DE CALIDAD PARA LA GESTIÓN DE ARCHIVOS DIGITALES.....</b>	<b>62</b>
5.1. DISEÑO DE UN ESQUEMA ORIENTADO A CALIDAD PARA LA GESTIÓN DE ARCHIVOS DIGITALES .....	62
5.2. OBJETIVOS .....	63
5.3. CARACTERIZACIÓN DEL PROCESO .....	63
5.4. IDENTIFICACIÓN DE ACTORES.....	64

5.5. INTERACCIÓN DEL PROCESO .....	65
5.6. ACTIVIDADES DEL PROCESO GESTIÓN DE ARCHIVOS DIGITALES.....	67
5.6.1. <i>Planear la gestión de archivos digitales</i> .....	67
5.6.2. <i>Preparar servidor de repositorios</i> .....	68
5.6.3. <i>Crear repositorio</i> .....	68
5.6.4. <i>Poblar repositorio</i> .....	68
5.6.5. <i>Preparar Estación de Trabajo Cliente</i> .....	69
5.6.6. <i>Interactuar con archivos digitales</i> .....	70
5.6.7. <i>Sincronizar archivos digitales</i> .....	70
5.6.8. <i>Generar Versión</i> .....	71
5.6.9. <i>Resolver Conflictos</i> .....	71
5.6.10. <i>Respaldar información</i> .....	72
5.6.11. <i>Restaurar Información</i> .....	72
5.6.12. <i>Monitorear el servidor de repositorios</i> .....	73
5.6.13. <i>Realizar auditorías</i> .....	73
5.6.14. <i>Evaluar desempeño del proceso</i> .....	74
5.6.15. <i>Establecer mejora del proceso</i> .....	74
5.6.16. <i>Criterios de calidad del proceso</i> .....	75
5.7. RECURSOS DEL PROCESO .....	76
5.8. INDICADOR DEL PROCESO .....	77
5.9. PROCESOS RELACIONADOS.....	77
5.10. MODELO DEL PROCESO GESTIÓN DE ARCHIVOS DIGITALES CON BPMN .....	78
5.11. VISIÓN GENERAL DEL PROCESO .....	79
5.11.1. <i>Planear</i> .....	80
5.11.2. <i>Hacer</i> .....	81
5.11.3. <i>Verificar</i> .....	82
5.11.4. <i>Actuar</i> .....	82
<b>6. HERRAMIENTAS PARA IMPLEMENTAR UN SISTEMA DE CONTROL DE VERSIONES .....</b>	<b>83</b>
6.1. CVS.....	83
6.2. <i>SUBVERSION</i> .....	84
6.3. GIT.....	85
6.4. MERCURIAL .....	86
6.5. BAZAAR.....	87
6.6. SELECCIÓN DE LA HERRAMIENTA PARA EL CONTROL DE VERSIONES .....	88
6.6.1. <i>Por qué elegir Subversion</i> .....	89
6.6.2. <i>Arquitectura de Subversion</i> .....	92

<b>7. DISEÑO DE UN ENTORNO DE OPERACIÓN BASADO EN MÁQUINAS VIRTUALES.....</b>	<b>94</b>
7.1. DISEÑO DEL PROTOTIPO FUNCIONAL.....	94
7.1.1. Descripción del “Ambiente Real”.....	95
7.1.2. Descripción del “Ambiente Virtual”.....	96
7.2. REQUERIMIENTO PARA LA INSTALACIÓN DE <i>SUBVERSION</i> .....	100
7.3. INSTALACIÓN DE LA HERRAMIENTA <i>SUBVERSION</i> EN EL SERVIDOR.....	100
7.3.1. Instalación asistida de <i>Subversión</i> .....	101
7.3.2. Instalación manual de <i>Subversion</i> .....	109
7.3.3. Paquetes requeridos para la instalación manual de <i>Subversion</i> .....	110
7.3.4. Librerías requeridas para la instalación manual de <i>Subversion</i> .....	111
7.3.5. Instalar el motor de bases de datos <i>Berkeley</i> .....	111
7.3.6. Instalar <i>Apache Runtime Project (APR)</i> .....	113
7.3.7. Instalar las <i>APR-UTIL</i> .....	114
7.3.8. Instalación manual de <i>Apache Web Server</i> .....	116
7.3.9. Construcción de binarios para la instalación manual de <i>Subversión</i> .....	118
7.4. PLANEACIÓN DEL REPOSITORIO.....	119
7.4.1. Estructura del repositorio.....	120
7.4.2. Definición de roles.....	123
7.4.3. Especificación reglas de control de acceso.....	125
7.5. CREACIÓN DEL REPOSITORIO, USUARIOS Y PERMISOS.....	125
7.5.1. Creación del repositorio.....	126
7.5.2. Creación de usuarios.....	126
7.5.3. Asignación de permisos.....	127
7.6. POBLADO INICIAL DEL REPOSITORIO.....	128
7.7. INSTALACIÓN DE HERRAMIENTAS EN LAS ESTACIONES DE TRABAJO CLIENTE.....	131
7.7.1. Configuración del cliente con sistema operativo <i>Ubuntu GNU/Linux 10.4</i> .....	132
7.7.2. Configuración del cliente con sistema operativo <i>MS-Windows XP</i> .....	135
7.8. SINCRONIZAR ARCHIVOS DIGITALES CON EL REPOSITORIO.....	137
7.8.1. Obtener una copia de trabajo.....	137
7.8.2. Confirmar cambios hacia el repositorio.....	140
7.9. INTERACCIÓN DE LOS CLIENTES CON EL REPOSITORIO CENTRAL.....	141
7.9.1. Interacción a través de la herramienta cliente <i>RabbitVCS</i> .....	141
7.9.2. Interacción a través de la herramienta cliente <i>TorotiseSVN</i> .....	143
7.9.3. Historial de cambios sobre un archivo.....	145
7.9.4. Comparación de diferencias entre revisiones de un documento.....	145
7.10. GENERAR VERSIONES SEGÚN HITOS DEL NEGOCIO.....	146

7.11. ASEGURAMIENTO DEL REPOSITORIO DE INFORMACIÓN.....	148
7.11.1. Respaldo del repositorio.....	148
7.11.2. Restauración del repositorio.....	150
<b>8. CUMPLIMIENTO DE OBJETIVOS.....</b>	<b>151</b>
<b>9. CONCLUSIONES.....</b>	<b>153</b>
<b>10. APOORTE Y TRABAJO FUTURO.....</b>	<b>155</b>
<b>11. REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>157</b>
<b>12. ANEXOS.....</b>	<b>160</b>
12.1. ANEXO 1: ELEMENTOS BÁSICOS PARA LA DIAGRAMACIÓN DE BPD.....	160
12.1.1. Tarea o Actividad.....	160
12.1.2. Subproceso.....	161
12.1.3. Compuertas.....	161
12.1.4. Eventos.....	162
12.1.5. Eventos de Inicio.....	162
12.1.6. Eventos intermedios.....	163
12.1.7. Eventos de Fin.....	166
12.1.8. Canales.....	167
12.1.9. Objetos de conexión.....	167
12.1.10. Artefactos.....	168
12.2. ANEXO 2: VIDEOS DE APOYO PARA LA OPERACIÓN DE LAS HERRAMIENTAS UTILIZADAS.....	169

## TABLA DE ILUSTRACIONES

<i>Ilustración 1: Porcentaje de empresas que utilizan TICs - Total Nacional 2006</i> .....	14
<i>Ilustración 2: Círculo de Calidad de Deming</i> .....	33
<i>Ilustración 3: Mejora continua de la calidad</i> .....	34
<i>Ilustración 4: ITIL V3 Soporte al Servicio</i> .....	37
<i>Ilustración 5: ITIL V3 Provisión del Servicio</i> .....	38
<i>Ilustración 6: ITIL V3 Gestión de la Seguridad</i> .....	39
<i>Ilustración 7: ITIL V3 Gestión de Configuraciones</i> .....	41
<i>Ilustración 8: ITIL V3 Gestión de Cambios</i> .....	43
<i>Ilustración 9: ITIL V3 Gestión de Versiones</i> .....	44
<i>Ilustración 10: ITIL V3 Evolución de las Versiones</i> .....	49
<i>Ilustración 11: ITIL V3 Proceso de Gestión de Versiones</i> .....	51
<i>Ilustración 12: Modelo de mejora de proceso</i> .....	62
<i>Ilustración 13: Modelo del Proceso Gestión de Archivos Digitales</i> .....	79
<i>Ilustración 14: Actividades del proceso Gestión de Archivos Digitales según PHVA</i> .....	80
<i>Ilustración 15: Comparación de Subversion con otras herramientas para el control de versiones</i> .....	90
<i>Ilustración 16: Detalles de la comparación de Subversion con otras herramientas para el control de versiones</i> .....	91
<i>Ilustración 17: Arquitectura Subversion</i> .....	92
<i>Ilustración 18: Prototipo funcional - Escenario Virtual</i> .....	94
<i>Ilustración 19: Oracle Virtual Box versión utilizada en el proyecto</i> .....	96
<i>Ilustración 20: Detalles técnicos de la máquina virtual para el Servidor de repositorios</i> .....	97
<i>Ilustración 21: Detalles técnicos de la máquina virtual para el Cliente 1</i> .....	98
<i>Ilustración 22: Detalles técnicos de la máquina virtual para el Cliente 2</i> .....	99
<i>Ilustración 23: Instalación asistida de Subversion</i> .....	101
<i>Ilustración 24: Descarga de binarios de Subversión desde el Sitio Web oficial del proyecto</i> .....	102
<i>Ilustración 25: Binarios de Subversión para Debian GNU/Linux desde el Sitio Web oficial del proyecto</i> ..	103
<i>Ilustración 26: Script de Shell para la automatización de la actualización e instalación de Subversion...</i>	104
<i>Ilustración 27: Ejecución del script de Shell <code>svninstall_debianlenny_wandisco.sh</code></i> .....	105
<i>Ilustración 28: Script de Shell <code>svninstall_debianlenny_wandisco.sh</code> y la actualización de librerías dependientes</i> .....	106
<i>Ilustración 29: Instalación automatizada de herramientas complementarias a Subversion</i> .....	107
<i>Ilustración 30: Instalación de módulos que integran Subversion y Apache Web Server</i> .....	108
<i>Ilustración 31: Comprobando la nueva versión instalada de Subversion</i> .....	109
<i>Ilustración 32: Descargando el código fuente de Subversion</i> .....	110

<i>Ilustración 33: Ubicación del directorio que contendrá los repositorios de información .....</i>	<i>120</i>
<i>Ilustración 34: Estructura de repositorio TTB.....</i>	<i>121</i>
<i>Ilustración 35: Estructura de repositorio propuesta.....</i>	<i>122</i>
<i>Ilustración 36: Definición de grupos sobre la estructura del repositorio .....</i>	<i>124</i>
<i>Ilustración 37: Creación de un repositorio con Subversion.....</i>	<i>126</i>
<i>Ilustración 38: Creación de usuarios para la interacción con el Sistema de Control de Versiones.....</i>	<i>127</i>
<i>Ilustración 39: Asignación de reglas de control de acceso al repositorio “prototipo”.....</i>	<i>127</i>
<i>Ilustración 40: Verificación de la publicación del repositorio “prototipo”.....</i>	<i>128</i>
<i>Ilustración 41: Creación de la estructura del repositorio “prototipo”.....</i>	<i>129</i>
<i>Ilustración 42: Poblado inicial del repositorio “prototipo”.....</i>	<i>129</i>
<i>Ilustración 43: Verificación del poblado inicial.....</i>	<i>131</i>
<i>Ilustración 44: Opciones clásicas de Nautilus ante un clic secundario en un directorio cualquiera .....</i>	<i>132</i>
<i>Ilustración 45: Incluyendo rutas para descargar RabbitVCS en el gestor de paquetes.....</i>	<i>133</i>
<i>Ilustración 46: Instalación de RabbitVCS a través de apt-get en Ubuntu 10.4 .....</i>	<i>133</i>
<i>Ilustración 47: Opciones de RabbitVCS integradas con Nautilus.....</i>	<i>134</i>
<i>Ilustración 48: Inicio de la instalación de TortoiseSVN sobre MS Windows XP.....</i>	<i>135</i>
<i>Ilustración 49: Herramienta TortoiseSVN integrada con el sistema de archivos de Widows.....</i>	<i>136</i>
<i>Ilustración 50: Paquete de lenguaje español de TortoiseSVN.....</i>	<i>136</i>
<i>Ilustración 51: Directorio de trabajo local para contener la copia de trabajo del repositorio .....</i>	<i>138</i>
<i>Ilustración 52: Conexión al repositorio para obtener la copia de trabajo .....</i>	<i>138</i>
<i>Ilustración 53: Autenticación para obtener una copia de trabajo .....</i>	<i>139</i>
<i>Ilustración 54: Copia de trabajo obtenida exitosamente .....</i>	<i>139</i>
<i>Ilustración 55: Información obtenida del repositorio “prototipo” .....</i>	<i>140</i>
<i>Ilustración 56: Confirmando cambios de los archivos en la copia de trabajo. ....</i>	<i>140</i>
<i>Ilustración 57: Opciones de RubbitVCS para la gestión de Archivos Digitales.....</i>	<i>141</i>
<i>Ilustración 58: Adicionar archivos a la copia de trabajo .....</i>	<i>142</i>
<i>Ilustración 59: GUI para confirmar cambios hacia el repositorio .....</i>	<i>143</i>
<i>Ilustración 60: Resultado de una confirmación de cambios hacia el repositorio .....</i>	<i>143</i>
<i>Ilustración 61: Obteniendo una copia de trabajo a través de TortoiseSVN .....</i>	<i>144</i>
<i>Ilustración 62: GUI TortoiseSVN para establecer conexión con el Repositorio.....</i>	<i>144</i>
<i>Ilustración 63: Registro con el historial de cambios de una archivo.....</i>	<i>145</i>
<i>Ilustración 64: Llamado a la comparación de dos revisiones .....</i>	<i>146</i>
<i>Ilustración 65: Comparación de dos revisiones de un archivo.....</i>	<i>146</i>
<i>Ilustración 66: Opción de TortoiseSVN para la creación de una Versión.....</i>	<i>147</i>
<i>Ilustración 67: GUI TortoiseSVN para generación de una versión.....</i>	<i>147</i>
<i>Ilustración 68: Mensaje de confirmación de la creación de una Versión .....</i>	<i>148</i>
<i>Ilustración 69: Realizando copia de seguridad del repositorio.....</i>	<i>149</i>

*Ilustración 70: Restaurando un repositorio desde su copia de seguridad..... 150*

## 1. INTRODUCCIÓN

Cualquier organización sin importar su actividad económica, debe enfrentarse a altos niveles de competencia laboral a nivel nacional e internacional, todo esto fruto de las políticas cada vez más generalizadas de globalización y tratados de libre comercio en todo el mundo. Ante esta situación, las organizaciones deben presentar estrategias de negocio con altos niveles de calidad en la prestación de sus servicios o en la elaboración de sus productos y para ello necesitan contar con *Sistemas de Gestión de Calidad* que permitan formalizar los procesos y procedimientos que garanticen el cumplimiento de sus objetivos.

Un *Sistema de Gestión de Calidad* contiene la clara especificación de los procesos y procedimientos que la organización debe realizar; generalmente estos procesos están divididos en tres grupos. En primer lugar están los *procesos misionales* son todos aquellos relacionados con la actividad de negocio a la cual se dedica la organización; estos procesos son de carácter estacionario a lo largo del ciclo de vida de la organización. En segundo lugar están los *procesos estratégicos* los cuales están relacionados con la visión de la organización; esta generalmente cambia con el tiempo y en consecuencia afecta la definición de los *procesos estratégicos*. Por último están los *procesos de apoyo*; estos procesos brindan el soporte necesario a la organización de tal forma que faciliten la realización de *procesos estratégicos* y los *procesos misionales*.

En un escenario de confrontación empresarial a nivel mundial, existe una clara ventaja en las organizaciones de aquellos países con alta presencia de las tecnologías de la información y la comunicación (TIC), donde estas han pasado a convertirse en un factor determinante en la automatización de los procesos empresariales y en la reducción de costos de operación que garantizan la permanencia de los negocios.

Según el DANE (2006), en Colombia se tiene una clara aceptación hacia las TICs en el sector empresarial, lo cual brinda la posibilidad de utilizarla como una eficaz herramienta que permita alcanzar los niveles de calidad que se requieren para enfrentar a los mercados actuales y emergentes.

Las herramientas tecnológicas por sí mismas no constituyen un avance significativo para una organización, si alrededor de su uso, no está definida la forma correcta para utilizarlas según las necesidades de cada organización. Lo que para una organización puede ser adecuado, para otra puede no serlo. En ocasiones, el uso indebido de la tecnología puede incluso causar efectos negativos en las organizaciones, tal es el caso de una organización bancaria que ofrece sus servicios a través de Internet sin haber realizado los procedimientos que brinden seguridad en las transacciones ofrecidas.

Con la inclusión de la tecnología informática, las organizaciones han pasado del uso de los archivos en papel al uso de los archivos digitales. En su mayoría, las organizaciones carecen de un proceso formal que permita ejercer un control efectivo sobre los activos de información de la organización, representados por el conjunto de archivos digitales. Situaciones similares han enfrentado los desarrolladores de software, los cuales utilizan sistemas de *Control de Versiones* como buena práctica utilizada para cumplir las condiciones de *Calidad en el software*, permitiendo una gestión efectiva sobre los activos de información (código fuente) que deben ser gestionados por un grupo de personas, en este caso desarrolladores.

La *Calidad en el Software* busca la satisfacción de los usuarios (clientes) al permitir la entrega de un producto que cumple las expectativas y requerimientos previamente establecidos; por lo tanto, la *Calidad en el Software* es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad (Fernandez, Garcia, & Beltrán, 1995); y es precisamente este conjunto de características favorables las que se buscan recrear en otros escenarios como lo es la gestión de Archivos Digitales en las organizaciones.

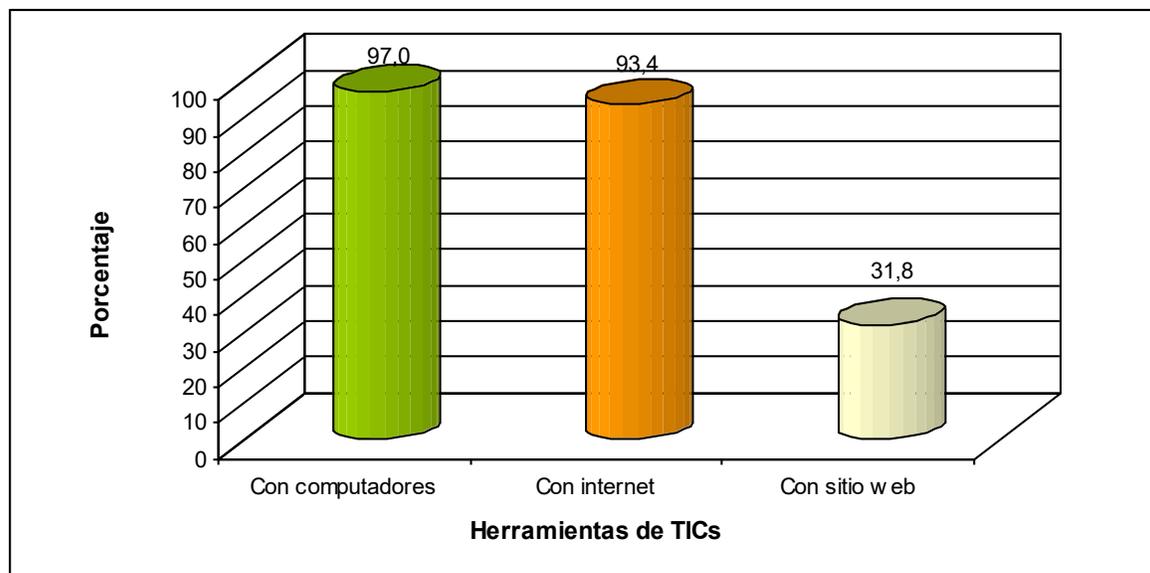
---

Esta tesis de maestría estudia la necesidad de gestionar adecuadamente los archivos digitales en cualquier tipo de organización y define un *proceso de apoyo* que especifica la forma como debe realizarse la *Gestión de Archivos Digitales* utilizando los *Sistemas de Control de Versiones*.

Este documento se encuentra organizada de la siguiente manera: La descripción del problema se puede encontrar en el capítulo 2. Los Objetivos están en el capítulo 3. El marco teórico dentro del cual se encuentra el presente trabajo está en el capítulo 4. Trabajos relacionados con la presente tesis se encuentran en el capítulo 5. El capítulo 6 contiene el trabajo desarrollado en torno a la *Gestión de Archivos Digitales*. En el capítulo 7 se realiza una revisión de diversas herramientas de Software Libre con las cuales se pueden implementar *Sistemas de Control de Versiones*. El capítulo 8, presenta un entorno de operación basado en *Máquinas Virtuales* el cual permite recrear un prototipo funcional del proceso *Gestión de Archivos Digitales*. Finalmente en el capítulo 9 presenta las conclusiones, los aportes y el trabajo futuro.

## 2. DESCRIPCIÓN DEL PROBLEMA

Según estudios realizados en Colombia (DANE, 2006), en las organizaciones se realizan las funciones administrativas y de negocios utilizando para el apoyo de las tecnologías de la información y la comunicación (TICs). Esta situación además de determinar una clara aceptación de la industria colombiana hacia la tecnología informática, ha suscitado la transición desde la manipulación de documentos físicos tales como: reglamentos, manuales, artículos, libros, informes, presentaciones de productos, etc., por la utilización de estos mismos tipos de documentos pero ahora en forma digital.



**Ilustración 1:** Porcentaje de empresas que utilizan TICs - Total Nacional 2006<sup>1</sup>

Los archivos en formato digitales, pese a su condición de elementos intangibles, siguen siendo considerados como activos de información de gran valor para las

<sup>1</sup> Tomado de DANE – Encuesta Anual de Comercio EAC

---

organizaciones, los cuales generalmente son creados y gestionados por personas que trabajan en forma colaborativa. En este escenario es típico encontrarse con diversos problemas asociados a la producción de los archivos digitales. Por ejemplo: duplicación de esfuerzos; desconocimiento de la versión más reciente de un documento; desconocimiento de la fecha y del autor de determinados cambios realizados a un documento, etc.

Por consiguiente, es de gran impacto en las organizaciones colombianas, la realización de una adecuada gestión de este tipo de información. Por lo anterior, las organizaciones necesitan contar con estrategias, procesos y herramientas que les permitan facilitar la gestión de archivos digitales.

El control de versiones es un proceso que se realiza en la industria informática concretamente en el desarrollo de software para controlar las distintas versiones resultados de la evolución del código fuente de un software específico. Sin embargo, los mismos conceptos pueden ser aplicables a otros ámbitos como la gestión de documentos de texto con formato, imágenes, sitios Web, etc. (Collins-Sussan, Fitzpatrick, & Pilato, 2008).

Tener un sistema de control de versiones es conveniente por diferentes motivos, por ejemplo, poder tener al alcance en todo momento la versión más actualizada de cada documento; saber quién y cuándo realizó determinado aporte o modificación al documento; tener la posibilidad de comparar versiones anteriores y de este modo saber qué se adicionó o se eliminó; tener la posibilidad de ver el registro de la evolución del documento y volver atrás si es necesario. (James & Collins, 2009)

Este proyecto formalizar el proceso de control de versiones de tal forma que se pueda lograr extender su utilización más allá de los límites del mundo del desarrollo de software, para que de esta forma, pueda ser utilizado por cualquier organización que desee modernizar el tratamiento de sus archivos informáticos obteniendo de esta

manera beneficios en el tratamiento de la información como son la integridad, disponibilidad y confiabilidad.

## 3. OBJETIVOS

### 3.1. Objetivo General

Diseñar una aproximación a la solución de los problemas de calidad originados en la gestión de archivos digitales de las organizaciones a través de un sistema de control de versiones basado en software libre.

### 3.2. Objetivos Específicos

- Diseñar un esquema orientado a calidad que determine una adecuada gestión de archivos digitales en entornos organizacionales.
- Seleccionar la herramienta de software libre adecuada para implementar una solución basada en el diseño de calidad previamente realizado.
- Diseñar un entorno basado en máquinas virtuales para una organización genérica que permita validar la efectividad de la solución.

## 4. MARCO TEÓRICO

### 4.1. Calidad

Tal como lo expresan (Alvarez, Alvarez, & Bullón, 2005), la calidad representa un proceso de mejora continua, en el cual todas las áreas de la empresa buscan satisfacer las necesidades del cliente o anticiparse a ellas, participando activamente en el desarrollo de productos o en la prestación de servicios. Esto tiene especial importancia en la actualidad para cualquier organización debido a la tendencia de los mercados a estar cada vez más globalizados. Por consiguiente, la única alternativa que tiene las organizaciones es blindar sus productos y servicios mediante la calidad de los mismos para poder ser competentes en un escenario geográficamente abierto.

Además, según lo expresado por Bon (2008) los servicios permiten establecer una relación con el cliente donde es importante considerar que el servicio no puede ser evaluado por anticipado, sino una vez ha sido prestado. Por lo anterior, las organizaciones necesitan establecer sistemas especiales que les ayuden a alcanzar el nivel de calidad necesario para sobrevivir comercialmente.

Benavides (2003) afirma que conocimiento de las organizaciones es un recurso intangible y cada vez más crítico para las organizaciones y paradójicamente, no aparece valorado en los documentos contables. Al tratarse de recursos heterogéneos, difíciles de ser medido y de importancia específica, las organizaciones no suelen generalmente plantearse su identificación ni su medición. A menudo ni siquiera están correctamente identificados por los ejecutivos y su gestión puede ser poco atendida o considerada superficialmente en los procesos relativos a la calidad.

Si los elementos informáticos intangibles como los archivos son cada vez más sustanciales para la competitividad de las empresas hay que empezar a preguntarse

---

hasta dónde y de qué manera las personas participan de su creación, y por lo tanto, habrá que considerar establecer métodos fiables para su conservación o eventual incremento, integrándolo mediante los correspondientes modelos en la gestión de la calidad. (Benavides & Quintana, 2003).

Las Tecnologías de la Información y la Comunicación (TICs) han contribuido al sorprendente crecimiento empresarial visto en los últimos años porque entre otras cosas han hecho posible los cambios producidos en las organizaciones y una mayor creación y utilización del conocimiento. Las TICs son causa y efecto de la revolución del conocimiento.

Las TICs no sólo han permitido una nueva reflexión sino que han logrado que "cosas" que no eran alcanzables por su precio o por su dificultad gestión ahora lo sean. En este sentido, conocimiento, TICs y gestión de la calidad total no son sino elementos interdependientes del sistema. La referencia a las TICs no es caprichosa, como tampoco sería las referencias a los sistemas de información y a la necesidad de replantear los diseños organizativos, etc.

En esta dirección, durante la década de los noventa se desarrolló una teoría sobre la gestión de la calidad hasta darle forma de disciplina estructurada. A finales de los noventa se inició el mismo proceso en lo se ha dado en llamar Gestión y Economía del Conocimiento acuñando los nombres de capital intelectual, intangibles, etc. (Benavides & Quintana, 2003)

La Gestión de la Información y la Gestión del Conocimiento no son procesos simultáneos. La Gestión de la Información es previa a la Gestión el Conocimiento. De hecho, la información se convierte en conocimiento cuando "alguien" la ha contextualizado, deliberadamente o no, de forma que gracias a este proceso de contextualización mejore su capacidad de actuar de forma inteligente. Se trata del llamado proceso cognitivo. Poder actuar es lo que separa a la información del conocimiento. Así, lo que para unas personas puede ser información (disposición de

datos), para otra es conocimiento (capacidad para la acción). (Benavides & Quintana, 2003)

La Gestión del Conocimiento es la manera en que la información se procesa con el fin de que se convierta en conocimiento. Así, las organizaciones deben averiguar cómo, cuándo y por qué se produce ese cambio, para posteriormente establecer patrones de comportamiento o pautas que relacionan esa transformación de información en conocimiento con el crecimiento de los recursos intangibles de la organización y aquí es donde el papel de la Gestión de la Calidad Total debe establecer el marco a la naturaleza de la empresa y de su situación presente y futura en los mercados. (Alvarez, Alvarez, & Bullón, 2005)

De acuerdo con los conceptos de Benavides (2005), es adecuado plantearse en la presente investigación los siguientes interrogantes:

- ¿Por qué debe gestionarse la información?
- ¿Por qué debe gestionarse el conocimiento?
- ¿Cuál es el impacto de una adecuada gestión de la información y el conocimiento?

A continuación es conveniente conocer los conceptos básicos relacionados con la gestión de la calidad en los procesos que deben seguir las organizaciones de hoy.

El término calidad tiene muchos significados diferentes en el lenguaje corriente pero para la norma ISO 8402 calidad es "el conjunto de características de una entidad que le confieren la aptitud para satisfacer las necesidades establecidas e implícitas". También podría decirse que es la "conformidad con los requisitos" y el "grado de excelencia". En un servicio la calidad se define como la diferencia entre la percepción

---

del servicio y la expectativa que el cliente tenía del mismo. Calidad es la satisfacción del cliente. Actualmente hay un solo enfoque de las empresas que son los clientes.

La calidad la definen los clientes (Benavides & Quintana, 2003). Por lo que la oferta deberá estar de acuerdo con lo que desean los clientes (sus exigencias) y entonces deberá producir exactamente lo que dichos clientes desean, en el plazo convenido y al mínimo costo e incluso anticiparse a sus necesidades y prestarle un servicio que no sólo satisfaga las expectativas del cliente sino deleite, lo que le dará una ventaja competitiva frente a la competencia.

La calidad no sólo hace referencia a que un producto o servicio se ajuste a las exigencias. La percepción que los clientes tienen sobre la empresa está basada en el producto o servicio que les suministra, pero también en el contacto diario que mantienen con los directivos. El concepto de calidad abarca no sólo como se atienden las exigencias de sus clientes sino también la forma en que se hace, cómo se atiende el teléfono, la rapidez con que se satisfacen consultas, tener nuevos servicios cuando se los requiere, asegurarse que la factura que sale de la empresa es correcta. Cada contacto, llamados momentos de la verdad con el cliente, muestra un reflejo de la compañía a los ojos de ese cliente. (Alvarez, Alvarez, & Bullón, 2005)

El recepcionista que atiende las llamadas, los agentes de seguridad que controlan los accesos, los ascensoristas que transportan a la gente y la orientan, los administrativos que envían las facturas, los dictámenes de los abogados, todos deben involucrarse en el concepto de calidad, en satisfacer las exigencias del cliente, en crear una compañía con clase reconocida. La calidad es un concepto objetivo que cada empleado puede comprender y evaluar, y sobre el cual puede asumir responsabilidades. (Alvarez, Alvarez, & Bullón, 2005)

La única forma de alcanzar el objetivo de calidad es (Benavides & Quintana, 2003):

- Comprender las exigencias de los clientes externos.

- Conocer a fondo los procesos internos que le permitan satisfacer esas exigencias.
- Desarrollar un sistema y cultura empresarial que le aseguren que los errores son evitados o eliminados.

La calidad de un producto está relacionada a que sea adecuado para el uso.

#### **4.2. Política organizacional de calidad**

Según la ISO 8402, la política organizacional de calidad se define como: "Directrices y objetivos generales de una empresa relativos a la calidad, expresados formalmente por la Dirección General". Es una de las vías para hacer operativa la estrategia. Al desplegar verticalmente la política a través de los niveles jerárquicos de la organización:

- Se proporciona orientación precisa para que ejecutivos y mandos elaboren planes concretos de acción.
- Se cohesionan la organización para el cumplimiento de los objetivos de calidad.
- Se refuerza el compromiso del personal.

La política de calidad debe ser:

- Adecuada a la empresa.
- Coherente con las necesidades y expectativas de sus clientes.

- Muy simple y fácilmente comprensible para que sea comunicable y entendida sin dificultad.

En cuanto a su contenido, debería hacer referencia a:

- Los grandes objetivos de la empresa.
- Los colectivos que en ella conviven: personal, accionistas, clientes<sup>1</sup> proveedores, etc.
- La disponibilidad de los recursos necesarios; por ejemplo, formación.
- La vía a utilizar (ISO, Mejora Continua, etc.).

#### **4.3. Terminología según ISO 8402**

- Calidad: “Conjunto de propiedades y características de un producto o servicio que le confieren su aptitud para satisfacer unas necesidades explícitas o implícitas”.
- Control de calidad: “Conjunto de técnicas y actividades de carácter operativo, utilizadas para verificar los requerimientos relativos a la calidad del producto o servicio”.
- Garantía de calidad: “Conjunto de acciones planificadas y sistemáticas necesarias para proporcionar la confianza adecuada de que un producto o servicio cumplirá los requerimientos dados sobre calidad”.

- **Gestión de calidad:** “Aspecto de la función de gestión que determina y aplica la política de la calidad, los objetivos y las responsabilidades y que lo realiza con medios tales como la planificación de la calidad, el control de la calidad, la garantía de calidad y la mejora de la calidad”. La gestión de la calidad es responsabilidad de todos los niveles ejecutivos, pero debe estar guiada por la alta dirección. Su realización involucra a todos los miembros de la organización. En la gestión de la calidad, se tienen en cuenta también criterios de rentabilidad.
- **Sistema de gestión de la calidad:** “Conjunto de la estructura de la organización, de responsabilidades, procedimientos, procesos y recursos que se establecen para llevar a término la gestión de calidad”.
- **Aseguramiento de la calidad:** "Es un conjunto de actividades preestablecidas y sistematizadas, aplicadas al sistema de calidad, que ha sido demostrado que son necesarias para dar confianza adecuada de que un producto o servicio satisfará los requisitos para la calidad".
- **Acción Correctiva:** Acción tomada para eliminar las causas de una no conformidad, defecto o cualquier situación indeseable existente, para evitar su repetición.
- **Acción Preventiva:** Acción tomada para eliminar las causas de una no conformidad, defecto o cualquier situación indeseable potencial, con el fin de evitar que se produzca.
- **Auditor de la Calidad:** Persona calificada para efectuar auditorias de la calidad.
- **Auditoria de la Calidad:** Examen sistemático e independiente con el fin de determinar si las actividades y los resultados relativos a la Calidad satisfacen

---

las disposiciones preestablecidas, y si éstas disposiciones son aplicadas en forma efectiva y son apropiadas para alcanzar los objetivos.

- **Calidad:** La totalidad de las características de un producto o servicio que le confieren aptitud para satisfacer necesidades establecidas e implícitas.
- **Cliente:** Destinatario de un producto provisto por el proveedor.
- **Conformidad:** Cumplimiento de requisitos especificados.
- **Control de la Calidad:** Técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos de Calidad de un producto o servicio.
- **Comprador:** Cliente en una situación contractual.
- **Contratista:** Proveedor en una situación contractual.
- **Costo de la No Calidad:** Costos asociados con la provisión de productos o servicios de baja calidad.
- **Defecto:** No cumplimiento de un requisito o de una expectativa razonable, ligada a un uso previsto, incluyendo los relativos a la seguridad.
- **Especificación:** Documento que establece los requisitos que un producto o servicio debe cumplir.
- **Evidencia Objetiva:** Información cuya veracidad puede demostrarse, basada en hechos y obtenida por observación, medición, ensayo u otros medios.

- **Gestión de la Calidad:** Actividades de la función empresaria que determinan la política de la calidad, los objetivos y las responsabilidades, y que se implementan a través de la planificación de la calidad, el control de la calidad, el aseguramiento de la calidad y el mejoramiento de la calidad, en el marco del sistema de la calidad.
- **Gestión de la Calidad Total:** Forma de gestión de un organismo centrada en la calidad, basada en la participación de todos sus miembros, y que apunta al éxito a largo plazo a través de la satisfacción del cliente y a proporcionar beneficios para todos los miembros del organismo y para la sociedad.
- **Inspección:** Actividades como medir, examinar, ensayar o comparar una o más características de un producto o servicio, y comparar los resultados con los requisitos especificados, con el fin de determinar la conformidad con respecto a cada una de esas características.
- **ISO:** International Organization for Standardization.
- **Manual de la Calidad:** Documento que enuncia la política de la calidad y que describe el sistema de la calidad de un organismo.
- **Mejoramiento de la Calidad:** Acciones emprendidas en todo el organismo con el fin de incrementar la efectividad y la eficiencia de las actividades y de los procesos para brindar beneficios adicionales al organismo y a sus clientes.
- **No Conformidad:** No satisfacción de un requisito especificado.
- **Organismo:** Compañía, sociedad, firma, empresa o institución, o parte de éstas, pública o privada, que posee su propia estructura funcional y administrativa.

- 
- Organización: Responsabilidades, autoridades y relaciones, ordenadas según una estructura jerárquica, a través de la cual un organismo cumple sus funciones.
  - Plan de la Calidad: Documento que enuncia las prácticas, los medios y la secuencia de las actividades ligadas a la calidad, ya sean específicas de un producto, proyecto o contrato particular.
  - Planificación de la Calidad: Actividades que establecen los objetivos y los requisitos para la calidad, así como los requisitos para la aplicación de los elementos del sistema de la calidad.
  - Política de la Calidad: Orientaciones y objetivos generales de un organismo concernientes a la calidad, expresados formalmente por el nivel más alto de dirección.
  - Prestación del Servicio: Aquellas actividades del proveedor que son necesarias para proveer el servicio.
  - Procedimiento: Manera especificada de realizar una actividad.
  - Proceso: Conjunto de recursos y actividades relacionadas entre sí que transforman elementos entrantes (input) en elementos salientes (output).
  - Producto: Resultado de actividades o de procesos.
  - Proveedor: Organismo que provee un producto a un cliente.
  - Registro: Documento que provee evidencias objetivas de las actividades efectuadas o de los resultados obtenidos.

- Re-trabajo: Acción tomada sobre un producto no conforme de modo que satisfaga los requisitos especificados.
- Servicio: Resultado generado por actividades en la interfaz entre el proveedor y el cliente, y por actividades internas del proveedor, con el fin de responder a las necesidades del cliente.
- Sistema de la Calidad: Organización, procedimientos, procesos y recursos necesarios para implementar la gestión de la calidad.
- Subcontratista: Organismo que provee un producto al proveedor.
- Trazabilidad: Aptitud de reconstruir la historia, la utilización o la localización de un producto por medio de identificaciones registradas.
- Validación: Confirmación por examen y aporte de evidencias objetivas de que los requisitos particulares para un uso específico previsto han sido satisfechos.
- Verificación: Confirmación por examen y aporte de evidencias objetivas que los requisitos especificados han sido satisfechos

#### **4.4. Notación para el Modelado de Proceso de Negocio (BPMN):**

Para realizar la definición de procesos, se utilizará la Notación para el Modelado de Procesos de Negocio o BPMN por sus siglas en inglés "*Business Process Modeling Notation*", en la versión 1.1 (OMG, 2008).

---

BPMN define un Diagrama de Proceso de Negocios o BPN por sus siglas en inglés “*Bussiness Process Diagram*”, un BPD representa gráficamente la secuencia de todas las actividades que ocurren durante la realización de un proceso de negocio.

Un BPD es un diagrama diseñado para ser usado por los analistas de procesos, quienes diseñan, controlan y gestionan los procesos. Un BPD puede contener varios procesos

BPMN propone tres tipos básicos de Modelos de procesos:

- **Proceso de Negocio Interno (Privado):** Se muestra un único proceso de negocio, donde normalmente se muestra toda la secuencia del proceso.
- **Proceso de Negocio abstracto (Público):** Representa un proceso de negocios externo, es decir, un proceso donde desconocemos el flujo detallado de las actividades.
- **Proceso de Negocio colaborativo (Global):** Representa la interacción entre dos o más unidades de negocio, estas interacciones están definidas por una secuencia de actividades que representan los mensajes intercambiados entre las entidades involucradas.

Nota: Para realizar correctamente un diagrama BPD, es necesario conocer con claridad los símbolos que se deben utilizar; para ellos se sugiere revisar el anexo 1 (Elementos básicos para la diagramación de BPD).

#### 4.5. Calidad en el software

El concepto de calidad aplicado al software no es nada nuevo, según afirman diversos autores (Fernandez, Garcia, & Beltrán, 1995), desde la década del 70, este tema ha

sido motivo de preocupación para especialistas, ingenieros, investigadores y comercializadores de software. Todo esto en búsqueda de la satisfacción de sus clientes al entregarles un producto que cumpla las expectativas y de esta manera garantizar la sostenibilidad organizacional.

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad. (Fernandez, Garcia, & Beltrán, 1995)

La calidad del software es medible y varía de un sistema a otro o de un programa a otro. Un software elaborado para el control de naves espaciales debe ser confiable al nivel de "cero fallas"; un software hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad; mientras que un producto de software para ser explotado durante un largo período (10 años o más), necesita ser confiable, de fácil mantenimiento y flexible para disminuir los costos de perfeccionamiento durante el tiempo de explotación. (Fernandez, Garcia, & Beltrán, 1995)

La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software.

#### **4.6. Actividades para el aseguramiento de calidad de software**

#### **4.7. Métricas de software**

Para controlar la calidad del software es necesario, ante todo, definir los parámetros, indicadores o criterios de medición, ya que, como bien plantea Tom

---

De Marco, "usted no puede controlar lo que no se puede medir". (Fernandez, Garcia, & Beltrán, 1995)

Las cualidades para medir la calidad del software son definidas por innumerables autores, los cuales las denominan y agrupan de formas diferentes. Por ejemplo, John Wiley define métricas de calidad y criterios, donde cada métrica se obtiene a partir de combinaciones de los diferentes criterios. La Metodología para la evaluación de la calidad de los medios de programas de la CIC, de Rusia, define indicadores de calidad estructurados en cuatro niveles jerárquicos: factor, criterio, métrica, elemento de evaluación, donde cada nivel inferior contiene los indicadores que conforman el nivel precedente. Otros autores identifican la calidad con el nivel de complejidad del software y definen dos categorías de métricas: de complejidad de programa o código, y de complejidad de sistema o estructura. (Fernandez, Garcia, & Beltrán, 1995)

#### **4.8. Verificación y validación del software**

La verificación y validación del software es un aspecto que merece toda la atención por parte de quienes desarrollan software; hoy en día, por el alto nivel de competitividad organizacional existentes, se requiere establecer con estrategia fundamental, un nuevo enfoque basado en anticipar la detección de errores, ya que éstos se producen con mayor frecuencia en las primeras fases del ciclo de vida del desarrollo. Una cantidad significativa de los errores cometidos en la producción de software provienen de las primeras fases del ciclo de vida (Análisis de requisitos y Diseño funcional y técnico).

Por otra parte, el costo de corregir los errores crece exponencialmente según avanza el proyecto, siendo éste significativamente menor cuanto antes se detecte los fallos.

Dada la importancia que tiene ahora para las organizaciones el alcanzar elevados niveles de eficiencia en el desarrollo de software, es de considerar un planteamiento más extenso y profundo de lo que habitualmente se conoce como fase de pruebas, aunque no garantice por sí mismo la calidad del software implantado, si aporta interesantes beneficios para alcanzar los objetivos de coste, plazos y calidad final del producto terminado.

#### **4.9. Modelos de calidad para software**

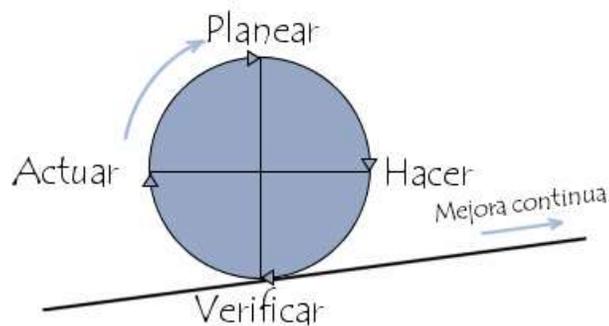
Un modelo de calidad para software es básicamente un conjunto de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos. Existen diversos modelos de calidad para software y todos ellos se encargan de determinar lo que se debe hacer, aunque no se enfocan en decir cómo debe hacerse ninguna actividad específica. Esta característica tan flexible de los modelos, facilita la adaptabilidad de las buenas prácticas a las necesidades puntuales de las organizaciones.

Los modelos de calidad más conocidos son:

- CMMI: Orientado a mejora de procesos en diferentes niveles de madurez, más hacia proyectos específicos.
- Norma ISO/IEC 12007: Orientado al proceso del ciclo de vida del software.
- Métrica 3: Utilizado por el ministerio de administración pública de España y es considerado y utilizado comúnmente en todo el mundo.

Uno de los aspectos en los que más se debe enfatizar es el Aseguramiento de la Calidad, por lo tanto se extenderá el concepto basado en lo planteado por (Bon, 2008). Suministrar productos o servicios requiere de actividades. La calidad de un

producto o servicio depende mucho de la manera en la que se organizan estas actividades. El Círculo de Calidad de Deming (ver Ilustración 2) muestra un modelo simple y eficaz para controlar la calidad. El modelo asume que para dar una calidad apropiada, se debe seguir los siguientes pasos:



**Ilustración 2:** Círculo de Calidad de Deming<sup>1</sup>

- Acción (Act - en inglés): Ajustar los planes basándose en la información recogida al comprobar. Si es el inicio del proceso, deben existir necesidades que para satisfacerse en la organización.
- Plan (Plan - en inglés): ¿Qué se debe hacer, cuándo quién debe hacerlo, cómo y utilizando qué?
- Ejecución (Do - en inglés): Se proceden a llevar a cabo las actividades programadas.
- Verificación (Check - en inglés): Determinar si las actividades dan los resultados esperados.

<sup>1</sup> Bon, J. V. (2008). *Fundamentos de Gestión de Servicios TI basados en ITIL*. itSMF Internacional.

A continuación se muestra en la Ilustración 3, la integración del Círculo de Calidad de Deming dentro de le esquema general de necesidades y resultados de una organización genérica.

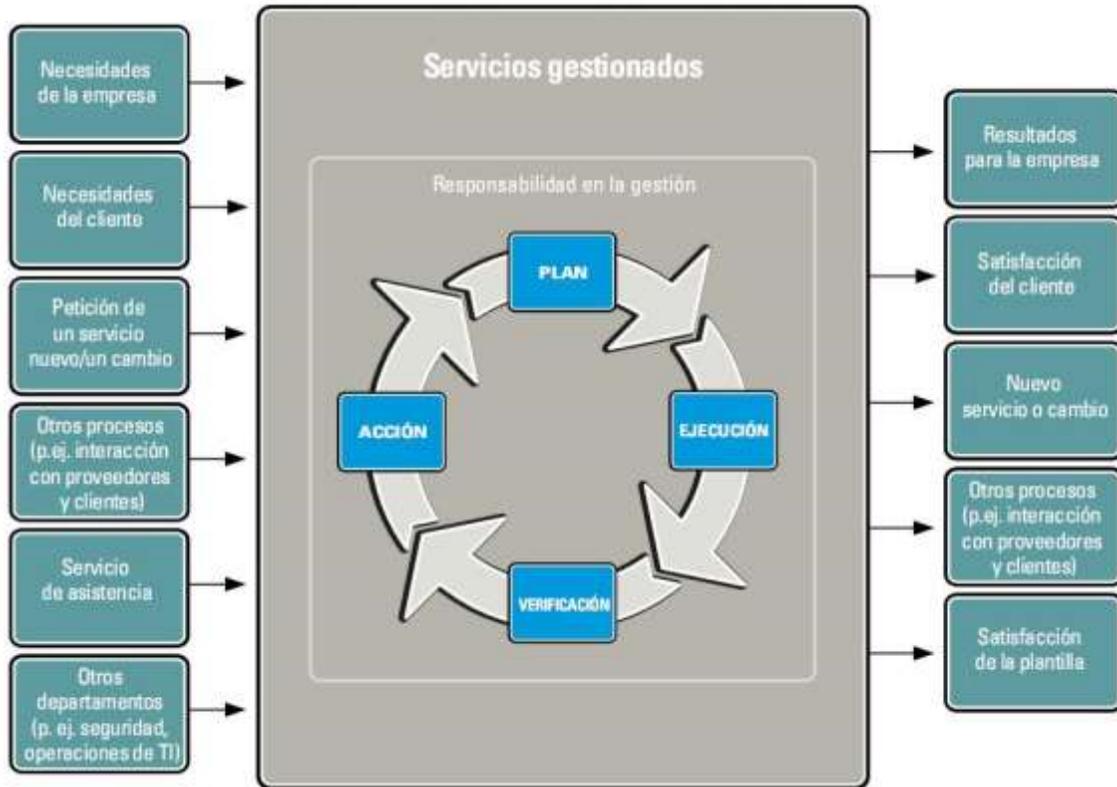


Ilustración 3: Mejora continua de la calidad<sup>2</sup>

#### 4.10. Gestión de Servicios de Tecnología Informática

De acuerdo a lo expuesto por Bon (2008), quien plantea que es indudable que con el desarrollo de las Tecnologías de la Información y la comunicación (TICs), se ha logrado potenciar a las organizaciones un alto grado de competitividad, provocando un gran impacto en los procesos de negocio y los resultados obtenidos hacia sus clientes. Las organizaciones en buscando la mayor adaptabilidad a un medio muy

<sup>2</sup> Tomado del Institute of IT Service Management.

---

cambiante, ha establecido estrategias donde se implantan procesos horizontales que se extienden a lo largo de la organización, y se otorgan a personas de menor nivel jerárquico, lo que puede suponer un riesgo si las personas encargadas no poseen el adecuado dominio del tema de sus funciones; para mitigar este hecho, se han desarrollado procesos operativos relacionados con la Gestión de los Servicios de Tecnología Informática (GSTI).

El estudio relacionado con GSTI no es nuevo, ya desde los años 80, el gobierno británico presentó gran interés en el tema y delegó a la Agencia Central de Telecomunicaciones y Computación (CCTA – por sus siglas en inglés) para que desarrollara una propuesta metodológica que permitiera al gobierno utilizar de manera eficaz y eficiente los recursos de TI. (OGC Norma ISO 8402 - 9004, 2009)

Los procesos eficaces y eficientes de la GSTI se convierten en esenciales para el éxito de los departamentos de TI. Esto se aplica a cualquier tipo de organización, grande o pequeña, pública o privada, con servicios TI centralizados o descentralizados, con servicios TI internos o suministrados por terceros. En todos los casos, el servicio debe ser fiable, consistente, de alta calidad, y de costo aceptable. (OSIATIS, 2009)

La intención principal, era independizar la dependencia de TI de los proveedores. (OGC Norma ISO 8402 - 9004, 2009). Esto dio como resultado todo un marco de referencia de buenas prácticas de TI conocido como Information Technology Infrastructure Library (ITIL).

#### **4.11. ITIL**

La biblioteca de infraestructura de tecnologías de la información (ITIL, por sus siglas en inglés) un estándar de facto, el cual proporciona una descripción detallada de una serie de buenas prácticas de TI, donde se expresan los roles, tareas, procedimientos y responsabilidades que pueden adaptarse a cualquier organización (Bon, 2008).

La OGC (2009) considera que la extensa cantidad de temas cubiertos por las publicaciones incluidas en ITIL la convierten en un elemento de referencia útil para fijar nuevos objetivos de mejora en torno a las organizaciones involucradas con las Tecnológicas de la Información y la Comunicación (TICs).

Según la (OGC Norma ISO 8402 - 9004, 2009), el principal objetivo de ITIL es alinear las organizaciones y la TI, permitiendo a las organizaciones implementar aquellos aspectos que sean relevantes para su negocio. ITIL es sólo un elemento de sentido común, documentado a partir de años de aprendizaje.

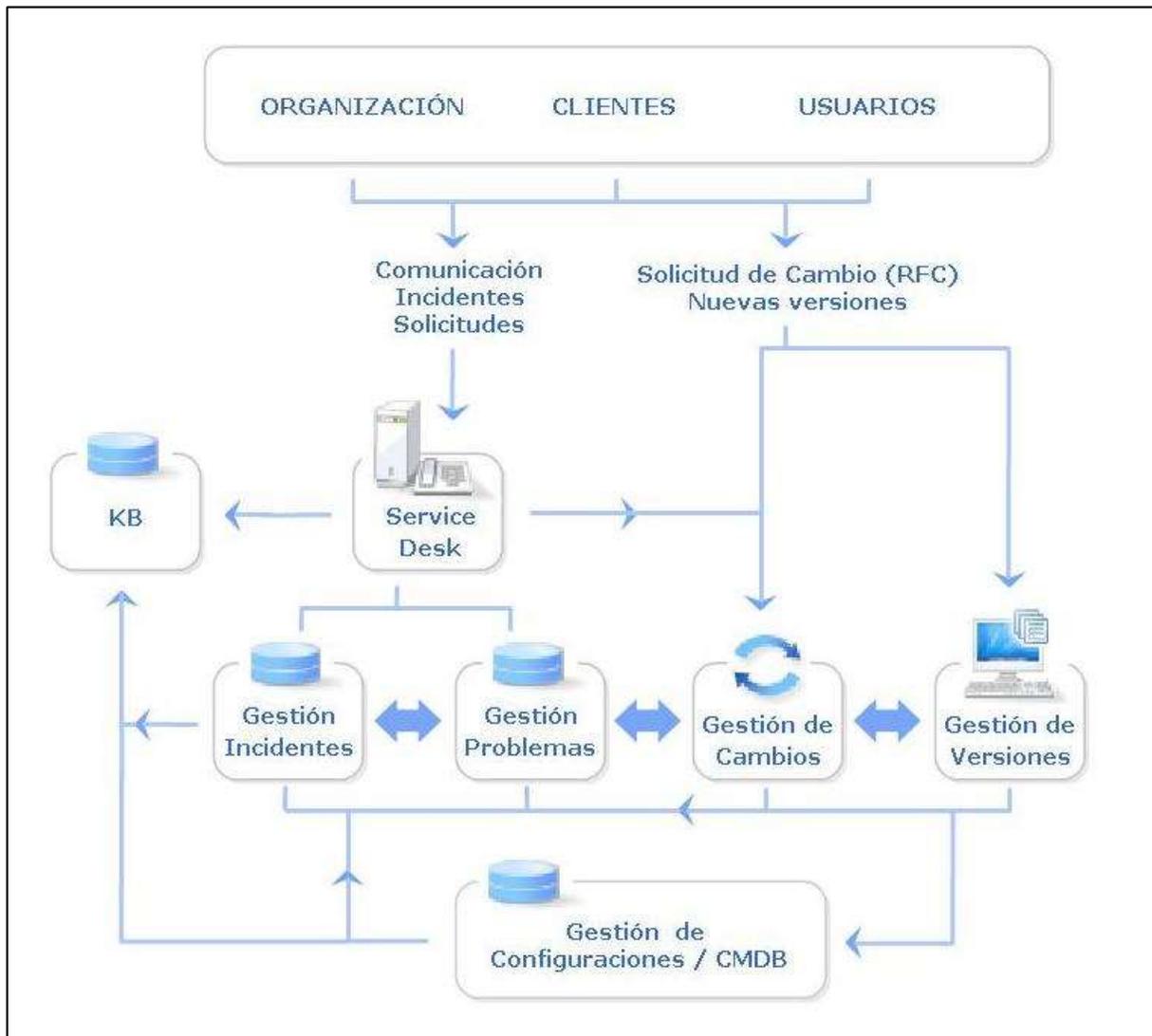
En los años 80, ITIL constaba de 10 libros centrales cubriendo las dos principales áreas de Soporte del Servicio y Prestación del Servicio. Estos libros centrales fueron más tarde soportados por 30 libros complementarios que cubrían una numerosa variedad de temas, desde el cableado hasta la gestión de la continuidad del negocio. A partir del año 2000, se acometió una revisión de la biblioteca. En esta revisión, ITIL ha sido reestructurado para hacer más simple el acceder a la información necesaria para administrar sus servicios. Los libros centrales se han agrupado en dos, cubriendo las áreas de Soporte del Servicio y Prestación del Servicio, en aras de eliminar la duplicidad y mejorar la navegación. El material ha sido también actualizado y revisado para un enfoque conciso y claro. (OSIATIS, 2009).

#### **4.12. Algunos componentes de ITIL**

##### **4.13. Soporte al servicio:**

Se centra en garantizar que el cliente tiene acceso a los servicios apropiados para apoyar las funciones empresariales. Incluye temas de cómo el servicio de escritorio y la gestión de incidentes, problemas, configuraciones, cambios y el control de

versiones. Es este el punto uno de los más relevantes a tener en cuenta sobre el parte que puede realizar ITIL al presente proyecto. (OSIATIS, 2009)

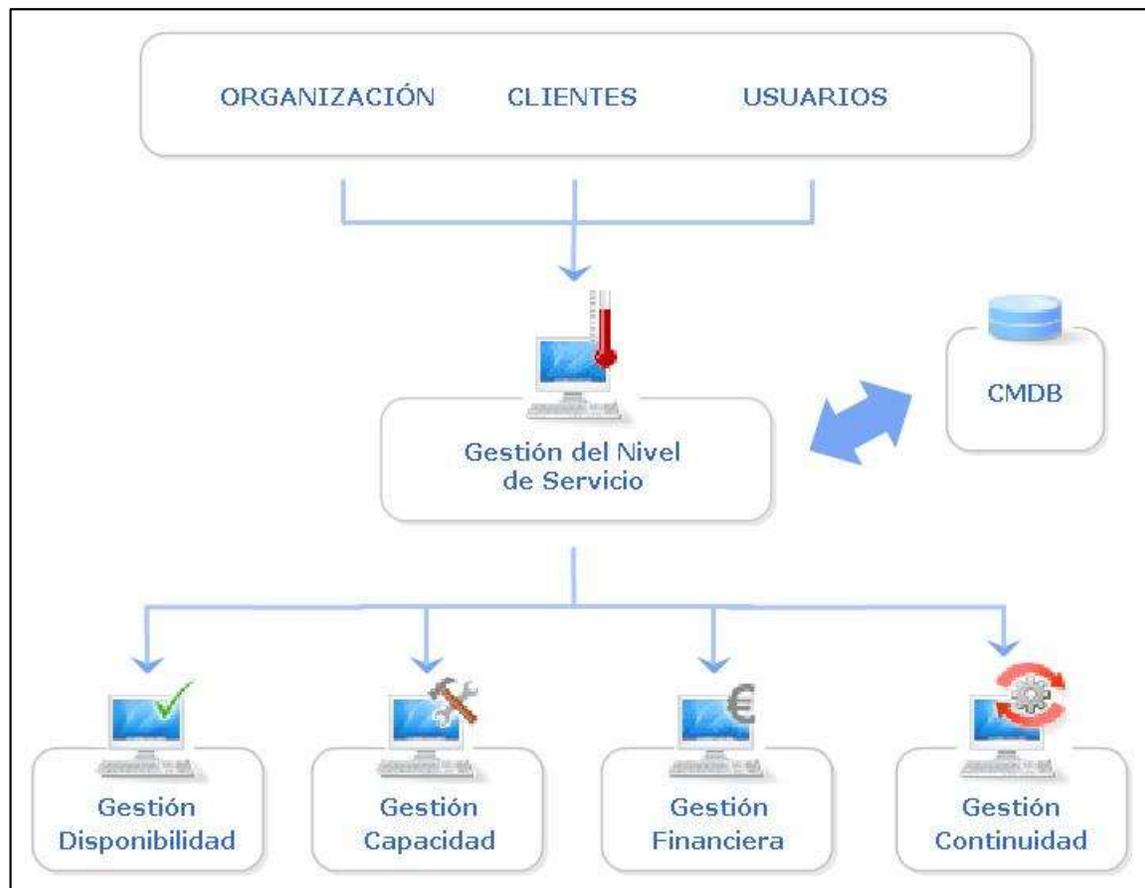


**Ilustración 4:** ITIL V3 Soporte al Servicio<sup>3</sup>

<sup>3</sup> Tomado de OSIATIS. (2009). *Osiatis*. From <http://itil.osiatis.es>

#### 4.14. Provisión del servicio:

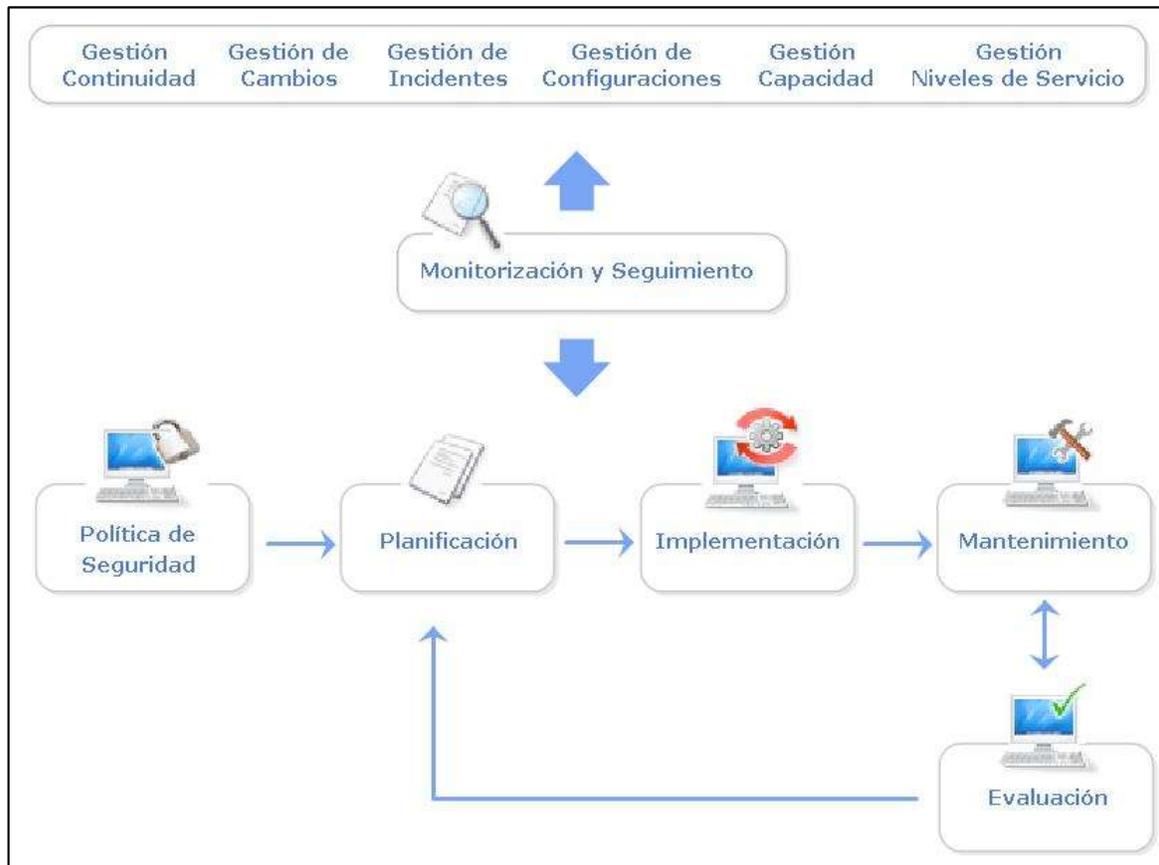
Los temas cubiertos incluyen son la Gestión de Niveles de Servicio, Gestión Financiera de los servicios de TI, Gestión de la continuidad del Servicio, Gestión de la Disponibilidad, Planes de Contingencia y Gestión de la Capacidad. El propósito de la Prestación del Servicio es mostrar los vínculos y las relaciones principales entre todos los servicios administrativos y otros procesos de Gestión de Infraestructura. (OSIATIS, 2009).



**Ilustración 5:** ITIL V3 Provisión del Servicio

#### 4.15. Gestión de la Seguridad:

La Gestión de la seguridad es una de las principales responsabilidades que se deben cubrir en una organización generando una adecuada política de seguridad. (OSIATIS, 2009).



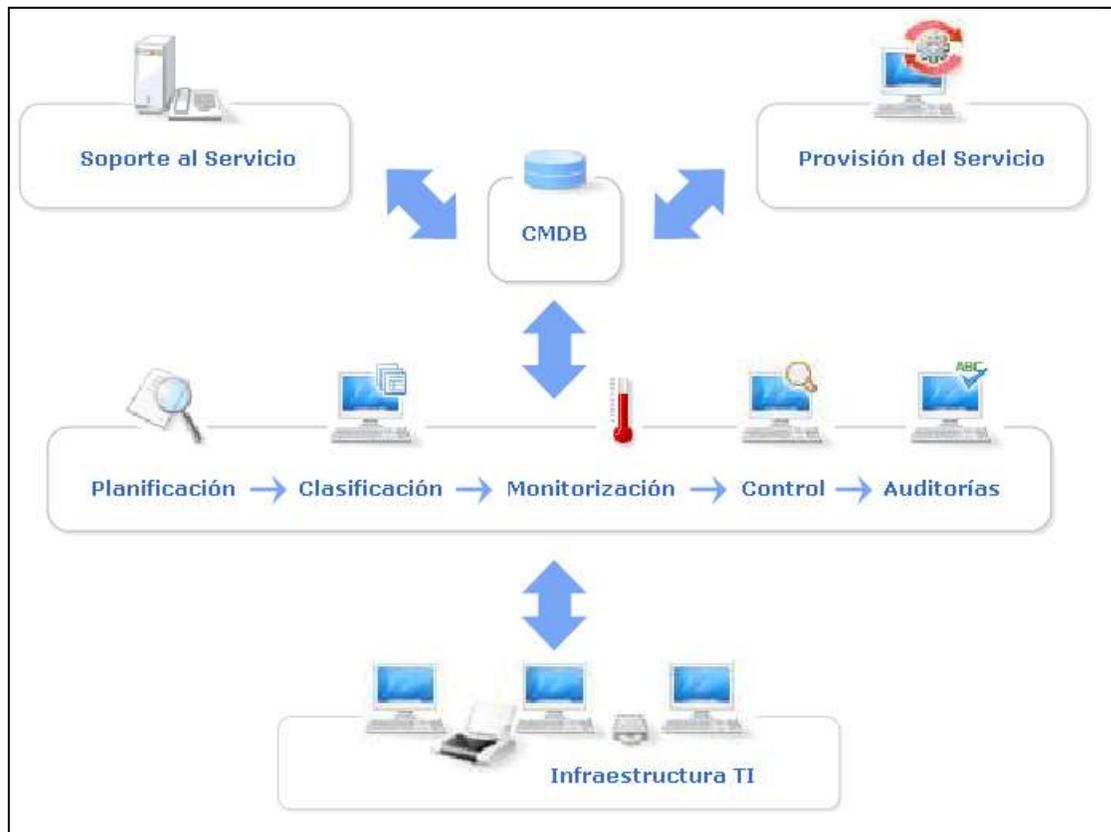
**Ilustración 6:** ITIL V3 Gestión de la Seguridad

#### 4.16. Gestión de Configuraciones:

La Gestión de Configuraciones tiene puede resumirse en cuatro grandes funciones como son:

- a. Llevar el control de todos los elementos de configuración de la infraestructura TI con el adecuado nivel de detalle y gestionar dicha información a través de la Base de Datos de Configuración (CMDB).
- b. Proporcionar información precisa sobre la configuración TI a todos los diferentes procesos de gestión.
- c. Interactuar con las Gestiones de Incidentes, Problemas, Cambios y Versiones de manera que estas puedan resolver más eficientemente las incidencias, encontrar rápidamente la causa de los problemas, realizar los cambios necesarios para su resolución y mantener actualizada en todo momento la CMDB.
- d. Monitorizar periódicamente la configuración de los sistemas en el entorno de producción y contrastarla con la almacenada en la CMDB para subsanar discrepancias.

Las interacciones y funcionalidades de la Gestión de Configuraciones se resumen en la Ilustración 7: (OSIATIS, 2009)



**Ilustración 7:** ITIL V3 Gestión de Configuraciones<sup>4</sup>

#### 4.17. Gestión de Cambios

El objetivo primordial de la Gestión de Cambios es que se realicen e implementen adecuadamente todos los cambios necesarios en la infraestructura y servicios TI garantizando el seguimiento de procedimientos estándar.

La Gestión de Cambios debe trabajar para asegurar que los cambios:

- Están justificados.

<sup>4</sup> Tomado de OSIATIS. (2009). *Osiatis*. From <http://itil.osiatis.es>

- Se llevan a cabo sin perjuicio de la calidad del servicio TI.
- Están convenientemente registrados, clasificados y documentados.
- Han sido cuidadosamente testeados en un entorno de prueba.
- Se ven reflejados en la CMDB.
- Pueden deshacerse mediante planes de "retirada del cambio" (back-outs) en caso de un incorrecto funcionamiento tras su implementación.

Las actividades principales de la Gestión de Cambios se resumen sucintamente en el siguiente diagrama (ver Ilustración 8: ITIL V3 Gestión de Cambios)

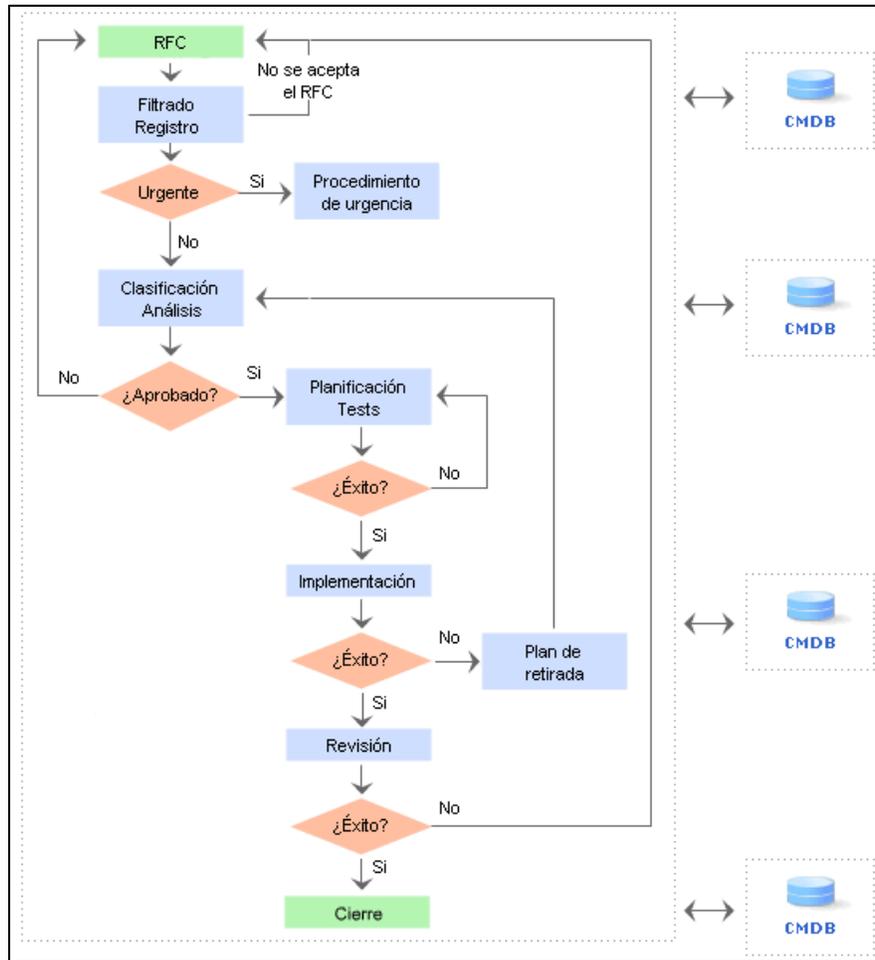


Ilustración 8: ITIL V3 Gestión de Cambios<sup>5</sup>

#### 4.18. Gestión de Versiones

La Gestión de Versiones es la encargada de la implementación y control de calidad de todo el software y hardware instalado en el entorno de producción.

La Gestión de Versiones debe colaborar estrechamente con la Gestión de Cambios y de Configuraciones para asegurar que toda la información relativa a las nuevas versiones

<sup>5</sup> Tomado de OSIATIS. (2009). *Osiatis*. From <http://itil.osiatis.es>

se integra adecuadamente en la CMDB de forma que ésta se halle correctamente actualizada y ofrezca una imagen real de la configuración de la infraestructura TI.

La Gestión de Versiones también debe mantener actualizada la Biblioteca de Software Definitivo (DSL), donde se guardan copias de todo el software en producción.

Las interacciones y funcionalidades de la Gestión de Versiones se resumen en la Ilustración 9 (OSIATIS, 2009):

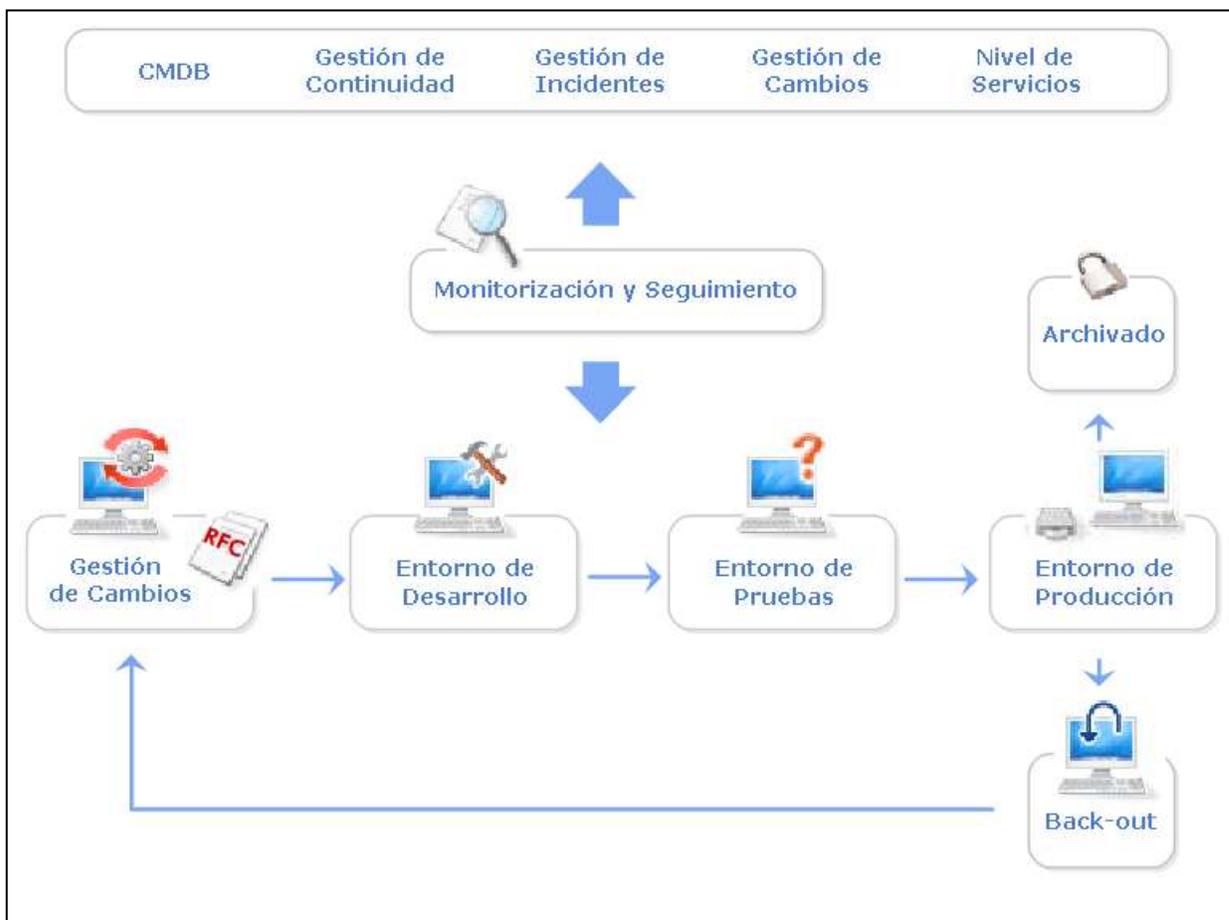


Ilustración 9: ITIL V3 Gestión de Versiones<sup>6</sup>

<sup>6</sup> Tomado de OSIATIS. (2009). *Osiatis*. From <http://itil.osiatis.es>

---

Las complejas interrelaciones entre todos los elementos que componen una infraestructura TI convierten en tarea delicada la implementación de cualquier cambio.

La Gestión de Cambios es la encargada de aprobar y supervisar todo el proceso pero es tarea específica de la Gestión de Versiones el diseñar, poner a prueba e instalar en el entorno de producción los cambios preestablecidos. Todo ello requiere de una cuidadosa planificación y coordinación con el resto de procesos asociados a la Gestión de Servicios TI.

Entre los principales objetivos de la Gestión de Versiones se incluyen:

- Establecer una política de implementación de nuevas versiones de hardware y software.
- Implementar las nuevas versiones de software y hardware en el entorno de producción tras su verificación en un entorno realista de pruebas.
- Garantizar que el proceso de cambio cumpla las especificaciones de la RFC correspondiente.
- Asegurar, en colaboración con la Gestión de Cambios y Configuraciones, que todos los cambios se ven correctamente reflejados en la CMDB.
- Archivar copias idénticas del software en producción, así como de toda su documentación asociada, en la Biblioteca de Software Definitivo (DSL).
- Mantener actualizado el Depósito de Hardware Definitivo (DHS).

Los beneficios de una correcta Gestión de Versiones se resumen en:

- El proceso de cambio se realiza sin deterioro de la calidad de servicio.
- Las nuevas versiones cumplen los objetivos propuestos.
- Se reduce el número de incidentes por incompatibilidades con otro software o hardware instalado.
- El proceso de pruebas asociado no sólo permite asegurar la calidad del software y hardware a instalar sino que también permite conocer la opinión de los usuarios sobre la funcionalidad y usabilidad de las nuevas versiones.
- El correcto mantenimiento de la DSL impide que se pierdan (valiosas) copias de los archivos fuente.
- Se reduce el número de copias de software ilegales.
- Control centralizado del software y hardware desplegado.
- Protección contra virus y problemas asociados a versiones de software incontroladas.

Las principales dificultades con las que topa la Gestión de Versiones son:

- No existe una clara asignación de responsabilidades y/o la organización TI no acepta la figura dominante de la Gestión de Versiones en todo el proceso de implementación del cambio.

- 
- No se dispone de un entorno de pruebas adecuado en donde se puedan testear de forma realista las nuevas versiones de software y hardware.
  - Hay resistencia en los diferentes departamentos a la centralización del proceso de cambio. Es habitual que existan reticencias a adoptar sistemas estandarizados en toda la organización, sobre todo cuando ésta no ha sido la política tradicional de la misma.
  - Se realizan cambios sin tener en cuenta a la Gestión de Versiones argumentado que estos sólo son responsabilidad de un determinado grupo de trabajo o que su "urgencia" requería de ello.
  - Hay resistencias a aceptar posibles planes de "back-out". Ciertos entornos de producción pueden elegir "ignorar" los problemas que una nueva versión puede provocar en otras áreas y resistirse a volver a la última versión estable.
  - La implementación sincronizada de versiones en entornos altamente distribuidos.

La solución a estos problemas pasa por:

- Un firme compromiso de la organización con la Gestión de Versiones y sus responsables.
- Un adecuado plan de comunicación que informe a todos los responsables y usuarios de la organización TI de las ventajas de una correcta gestión de todo el proceso de cambio.

Una versión es un grupo de elementos de configuración de nueva creación o modificados que han sido validados para su instalación en el entorno de producción. Las especificaciones funcionales y técnicas de una versión están determinadas en el

documento de petición de cambio o RFC por sus siglas en inglés (Request For Change) (OSIATIS, 2009).

Las versiones pueden clasificarse, según su impacto en la infraestructura TI, en:

- **Versiones mayores:** que representan importantes despliegues de software y hardware y que introducen modificaciones importantes en la funcionalidad, características técnicas, etc.
- **Versiones menores:** que suelen implicar la corrección de varios errores conocidos puntuales y que a menudo son modificaciones que vienen a implementar de una manera correctamente documentada soluciones de emergencia.
- **Versiones de emergencia:** modificaciones que reparan de forma rápida un error conocido.

Como pueden llegar a existir múltiples versiones es conveniente definir una referencia o código que los identifique unívocamente. El sistema universalmente aceptado es:

- Versiones mayores: 1.0, 2.0, etc.
- Versiones menores: 1.1, 1.2, 1.3, etc.
- Versiones de emergencia: 1.1.1, 1.1.2, etc

En su ciclo de vida una versión puede encontrarse en diversos estados: desarrollo, pruebas, producción y archivado.

El siguiente diagrama que se muestra en la Ilustración 10, se evidencia gráficamente la evolución temporal de una versión:

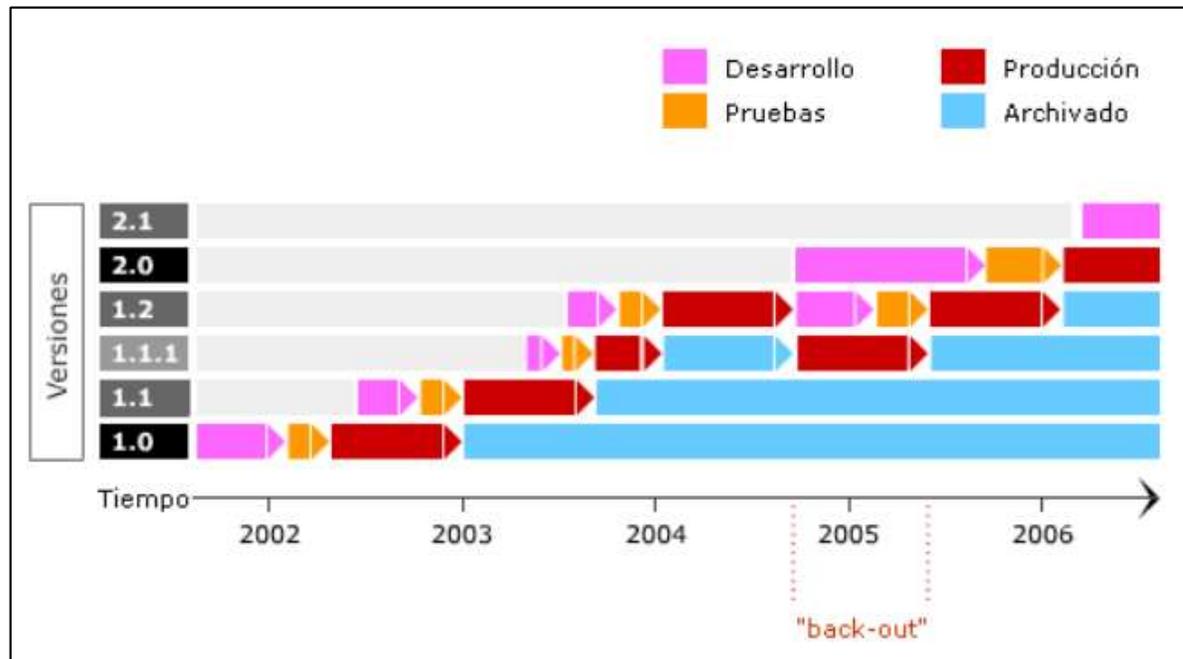


Ilustración 10: ITIL V3 Evolución de las Versiones<sup>7</sup>

El despliegue de nuevas versiones puede realizarse de diferentes maneras y es responsabilidad de la Gestión de Cambios el determinar la forma más conveniente de hacerlo. Entre las opciones más habituales cabe contar: (OSIATIS, 2009)

- **Versión delta:** sólo se testean e instalan los elementos modificados. Esta opción tiene como ventaja su mayor simplicidad pero conlleva el peligro de que puedan aparecer problemas e incompatibilidades en el entorno de producción.
- **Versión completa:** Se distribuyen todos los elementos afectados ya hayan sido modificados o no. Aunque esta opción es obviamente más trabajosa es

<sup>7</sup> Tomado de OSIATIS. (2009). *Osiatis*. From <http://itil.osiatis.es>.

más improbable que se generen incidentes tras la instalación si se han realizado las pruebas pertinentes.

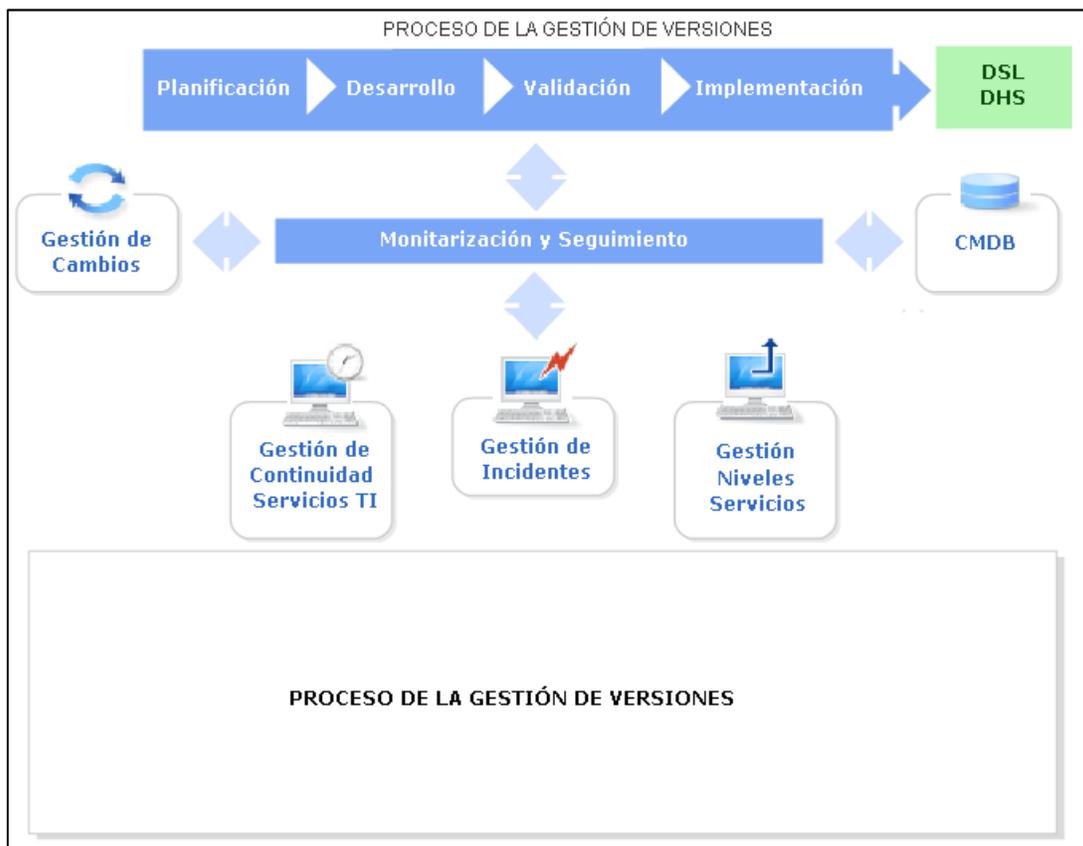
- **Paquete de Versiones:** La Gestión de Cambios puede optar por distribuir de forma sincronizada diferentes paquetes de versiones, de esta forma se ofrece una mayor estabilidad al entorno TI. En algunos casos esta opción es obligada por incompatibilidades entre una nueva versión con software o hardware previamente instalado. Pensemos, por ejemplo, en la migración a un nuevo sistema operativo que requiere hardware más avanzado y/o nuevas versiones de los programas ofimáticos.

Las principales actividades de la Gestión de Versiones se resumen en (OSIATIS, 2009):

- Establecer una política de planificación para la implementación de nuevas versiones.
- Desarrollar o adquirir de terceros las nuevas versiones.
- Poner a prueba las nuevas versiones en un entorno que simule lo mejor posible el entorno de producción.
- Validar las nuevas versiones.
- Implementar las nuevas versiones en el entorno de producción.
- Llevar a cabo los planes de back-out o retirada de la nueva versión si esto fuera necesario.
- Actualizar la DSL, el DHS y la CMDB.

- Comunicar y formar a los clientes y usuarios sobre las funcionalidades de la nueva versión.

El siguiente diagrama muestra los procesos implicados en la correcta Gestión de Versiones:



**Ilustración 11:** ITIL V3 Proceso de Gestión de Versiones<sup>8</sup>

## Planificación

Es crucial establecer un marco general para el lanzamiento de nuevas versiones que fije una metodología de trabajo. Esto es especialmente importante para los casos de

<sup>8</sup> Tomado de OSIATIS. (2009). *Osiatis*. From <http://itil.osiatis.es>

versiones menores y de emergencia pues en el caso de lanzamientos de gran envergadura se deben desarrollar planes específicos que tomen en cuenta las peculiaridades de cada caso (OSIATIS, 2009).

A la hora de planificar correctamente el lanzamiento de una nueva versión se deben de tomar en cuenta los siguientes factores:

- ¿Cómo puede afectar la nueva versión a otras áreas del entramado TI?
- ¿Qué elementos de configuración se verán directa o indirectamente implicados durante y tras el lanzamiento de la nueva versión?
- ¿Cómo ha de construirse el entorno de pruebas para que éste sea fiel reflejo del entorno de producción?
- ¿Qué planes de back-out son necesarios?
- ¿Cómo y cuándo se deben implementar los planes de back-out para minimizar el posible impacto negativo sobre el servicio y la integridad del sistema TI?
- ¿Cuáles son los recursos humanos y técnicos necesarios para llevar a cabo la implementación de la nueva versión con garantías de éxito?
- ¿Quiénes serán los responsables directos en las diferentes etapas del proceso?
- ¿Qué planes de comunicación y/o formación deben desarrollarse para que los usuarios estén puntualmente informados y puedan percibir la nueva versión como una mejora?
- ¿Cuál es la vida media útil esperada de la nueva versión?

- ¿Qué impacto puede tener el proceso de lanzamiento de la nueva versión en la calidad del servicio?
- ¿Si es posible establecer métricas precisas que determinen el grado de éxito del lanzamiento de la nueva versión?

## **Desarrollo**

La Gestión de Versiones es la encargada del diseño y construcción de las nuevas versiones siguiendo las pautas marcadas en las RFCs correspondientes (OSIATIS, 2009).

A veces el desarrollo se realizara "en la casa" y muchas otras requerirá la participación de proveedores externos. En este segundo caso, la tarea de la Gestión de Versiones será la de asegurar que el paquete o paquetes de software o hardware ofrecidos cumple las especificaciones detalladas en la RFC. Asimismo, la Gestión de Versiones será la responsable de todo el proceso de configuración necesario.

El desarrollo debe incluir, si esto fuera necesario o simplemente recomendable, todos los scripts de instalación necesarios para el despliegue de la versión. Estos scripts deberán tener en cuenta aspectos tales como:

- Back-up automático de datos.
- Actualizaciones necesarias de las Bases de Datos asociadas.
- Instalación de las nuevas versiones en diferentes sistemas o emplazamientos geográficos.

- Creación de archivos de registro o logs asociados al proceso de instalación.

Parte integrante del desarrollo lo componen los planes de back-out asociados. Estos tendrán que tomar en cuenta la disponibilidad acordada con los clientes en los acuerdos de niveles de servicio correspondientes

## **Validación**

Un bien planificado protocolo de pruebas es absolutamente indispensable para lanzar al entorno de producción una nueva versión con razonables garantías de éxito.

Las pruebas no deben limitarse a una validación de carácter técnico (ausencia de errores) sino que también deben realizarse pruebas funcionales con usuarios reales para asegurarse de que la versión cumple los requisitos establecidos y es razonablemente usable (siempre existe una inevitable resistencia al cambio en los usuarios que debe ser tenida en consideración).

Es importante que las pruebas incluyan los planes de back-out para asegurarnos que se podrá volver a la última versión estable de una forma rápida, ordenada y sin pérdidas de valiosa información.

Las principales actividades realizadas en el proceso de prueba deben incluir:

- Pruebas del correcto funcionamiento de la versión en un entorno realista.
- Pruebas de los procedimientos automáticos o manuales de instalación.
- Listas de "bugs" o errores detectados, si se diera el caso.

- 
- Pruebas de los planes de back-out.
  - Documentación para usuarios y personal de servicio.

La Gestión de Cambios será la encargada de dar la validación final a la versión para que se proceda a su instalación. Si la versión no fuera aceptada se devolverá a la Gestión de Cambios para su reevaluación.

### **Implementación**

Finalmente se tiene lista la distribución de la nueva versión, también conocida como rollout.

El rollout puede ser de varios tipos:

- **Completo y sincronizado:** se realiza de manera integral y simultánea en todos los emplazamientos.
- **Fragmentado:** ya sea bien espacial o temporalmente. Por ejemplo, introduciendo la nueva versión por grupos de trabajo o incrementando progresivamente la funcionalidad ofrecida.

El procedimiento de *roll-out* debe ser cuidadosamente documentado para que todas las partes conozcan sus tareas y responsabilidades específicas. En particular los usuarios finales deben estar puntualmente informados del calendario de lanzamiento y de cómo este puede afectar a sus actividades diarias.

Es imprescindible determinar claramente:

- Los elementos de configuración que deben borrarse e instalarse y en qué orden debe realizarse este proceso.
- Cuándo debe realizarse este proceso para diferentes grupos de trabajo y/o localizaciones geográficas.
- Que métricas determinan la puesta en marcha de los planes de back-out y si estos deben ser completos o parciales.

Tras la distribución la Gestión de Versiones debe asegurarse de que:

- Se incluya una copia de la versión en la biblioteca de software definitivo.
- La CMDB esté correctamente actualizada.
- Los usuarios están debidamente informados de las nuevas funcionalidades y han recibido la formación necesaria para poder sacar el adecuado provecho de las mismas.

## **Comunicación y Formación**

Es frecuente, y a su vez un grave error, que cuando se aborden cuestiones de carácter técnico se obvie el factor humano.

Salvo contadas excepciones, es necesaria la interacción usuario-aplicación y ésta suele representar el eslabón más débil de la cadena.

Es inútil disponer de un sistema adecuado para entregar una completa formación que permite aprovechar las ventajas tecnológicas.

---

La formación de los usuarios debe estructurarse en distintos niveles:

- Los usuarios deben conocer el próximo lanzamiento de una nueva versión y conocer con anterioridad la nueva funcionalidad planificada o los errores que se pretenden resolver para participar, a su discreción, en el proceso.
- Siempre que sea posible las pruebas de carácter funcional deben ser realizadas por un selecto grupo de usuarios finales. Durante este proceso de prueba se documentarán y analizarán:
  - La experiencia subjetiva de usuario.
  - Los comentarios y sugerencias sobre usabilidad y funcionalidad. o Las dudas que hayan surgido durante el uso de la nueva versión.
  - La claridad de la documentación que se pondrá a disposición del usuario final.
- Cuando se considere oportuno se impartirán cursos presenciales o remotos mediante módulos de e-learning sobre el funcionamiento de la nueva versión.
  - Se desarrollará una página de FAQs donde los usuarios puedan aclarar las dudas más habituales y puedan solicitar ayuda o soporte técnico en el uso de la nueva versión.

#### **4.19. Gestión de Configuraciones de Software**

Según Berczuk y Appleton (2003), la Gestión de Configuraciones de Software (SCM, por sus siglas en inglés) promueve las estrategias que orientan a los equipos de

desarrollo para obtener el mejor software posible. Además, SCM entrega las pautas para calmar las preocupaciones comunes de los ingenieros de software relacionadas con el proceso de desarrollo de software. Mediante el establecimiento de patrones, los desarrolladores evidencian como el flujo de trabajo se gestiona adecuadamente y evitando así los retrasos, valoraciones subjetivas y sobre costos. El enfoque basado en patrones muestra cómo SCM puede ser fácil y de aplicabilidad con éxito en las pequeñas y medianas organizaciones. Al aprender cómo estos patrones se relacionan entre sí, los desarrolladores pueden evitar errores comunes que a menudo dan como resultado frustraciones y reducción en productividad del desarrollo de software.

Algunas de los aspectos que cubre SCM son (Berczuk & Appleton, 2003):

- Desarrollar la nueva versión de un producto, mientras que se puede estar concentrado en la resolución de problemas relacionados con la versión actual.
- Desarrollar código en paralelo con otros desarrolladores y unirse con el estado actual del código en línea.
- Identificar cuáles son las versiones de código dentro de un componente en particular.
- Analizar los cambios ocurridos en el historial del desarrollo de un componente.
- Ayuda en la utilización de las herramientas de desarrollo actuales con más eficacia, y decidir cuándo utilizar un proceso manual.
- Introduce gradualmente buenas prácticas en relación con el desarrollo de software en espacios de trabajo individuales y de toda la organización.

- Construir y promover un entorno de desarrollo centrado en la producción de trabajo en equipo y productos de óptima calidad.

#### 4.20. Trabajo relacionado

Antes de iniciar con el proceso de toma de decisiones que condujeron el desarrollo del presente trabajo, se revisó la forma como otros investigadores en el mundo han realizado trabajos en temas relacionados.

Tal como se asegura en algunas investigaciones (Berczuk & Appleton, 2003), realizar gestión sobre los archivos que utilizan los equipos de desarrolladores especialmente el código fuente, es una necesidad que se han presentado desde hace mucho tiempo.

Durante todas las etapas del desarrollo de software se generan activos de información en forma de código fuente y documentación, los cuales son de gran valor para todo el equipo de desarrollo. Una particularidad del código fuente es que son archivos que evolucionan a medida que el proyecto avanza en el tiempo y esta evolución se realiza generalmente gracias a la intervención de diferentes desarrolladores. Para controlar esta situación, la industria de software ha propuesto diversas estrategias, las cuales convergen en la inclusión del concepto de *Calidad en el Software*. Por lo anterior, existen conjuntos de buenas prácticas que conducen al mejoramiento de los procesos de desarrollo de software, donde se destaca la “*Administración de Configuraciones de Software*”.

La “*Administración de Configuraciones de Software*” (ACS) o SCM por sus siglas en inglés “*Software Configuration Management*” es todo una disciplina que controla la evolución de los sistemas de software (Berczuk & Appleton, 2003). Esta disciplina es la que permitió la especificación y construcción de los *Sistemas de Control de*

*Versiones*; los cuales según Nagel (2005) se han convertido en una herramienta crítica para los desarrolladores.

Aunque el control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión, entre las más utilizadas se tiene las siguientes: CVS, Subversion, SourceSafe, ClearCase, Darcs, Bazaar, Plastic SCM, Git, Mercurial, etc.

Compañías con gran liderazgo tecnológico e impacto a nivel mundial como es el Google Inc, dedican gran parte de su presupuesto a realizar investigación y desarrollo aplicado en las tecnologías de información y la comunicación, con el propósito de introducir nuevos productos y servicios. Google a través de su producto "*Google Apps*" ha logrado introducir la utilización de herramientas de ofimática disponibles a través de Internet; lo que significa que *Google Apps*, es una solución alojada basada en Web, lo que permite que sea posible acceder a *Google Apps* en cualquier momento, siempre que se tenga una conexión a Internet. A este tipo de tecnología se le conoce como "software alojado como servicio" o "*cloud computing*". "software alojado como servicio" es un modelo informático en que las aplicaciones de TI se ofrecen "como servicio", lo que permite que los usuarios tengan acceso a ellas sin necesidad de mantener la infraestructura que las aloja. Un ejemplo para comprender el modelo es compararlo con el funcionamiento de las empresas de electricidad: son éstas las que producen y gestionan la electricidad en lugar de generarla los usuarios en sus propias casas u oficinas (Google Inc, 2009).

El la solución *Google Apps* se ha introducido el concepto de control de versiones como un gran característica para facilitar el trabajo colaborativo. El concepto es revolucionario y además han logrado que se realice de forma automática siendo esto una condición favorable por el alto grado de usabilidad presentado.

---

Aunque la iniciativa es muy buena, no ha logrado establecerse con fuerza en países con limitaciones en la conexión a Internet de alta velocidad, siendo esta una condición necesaria para aprovechar los beneficios de “*cloud computing*”.

Existe otro producto ampliamente conocido en el mundo de las herramientas ofimáticas “*OpenOffice*”. Este producto es considerado software libre bajo la licencia GPL, el cual ha venido desarrollándose desde octubre de 2000.

La herramienta OpenOffice, ha presentado una evolución constante y ha logrado posicionarse como una alternativa seria frente a las propuestas comerciales de herramientas ofimáticas. Ahora bien, una de sus características recientes es la incorporación del control de versiones sobre los documentos que sean escritos con la herramienta *Write* utilizada en la generación de documentos de texto; esta característica permite conocer los cambios realizados sobre un documento y la posibilidad de regresar a ellos si fuera necesario (OpenOffice, 2002). Sin embargo, esta característica no goza de gran aceptación debido al desconocimiento del concepto de control de versiones por parte de los usuarios.

## 5. ESQUEMA DE CALIDAD PARA LA GESTIÓN DE ARCHIVOS DIGITALES

### 5.1. Diseño de un esquema orientado a calidad para la gestión de archivos digitales

Esta etapa del proyecto corresponde a la realización de la definición del proceso relacionado con la gestión de archivos digitales en una organización.

En primera instancia hay que tener presente que un proceso es una serie de actividades relacionadas lógicamente que conducen a un objetivo definido (Bon, 2008), es este caso es la adecuada gestión de archivos digitales. Para proceder a definir el proceso se debe tener clara la respuesta a las interrogantes propuestas en el modelo de mejora de proceso de Bon (2008) que se muestra a continuación.

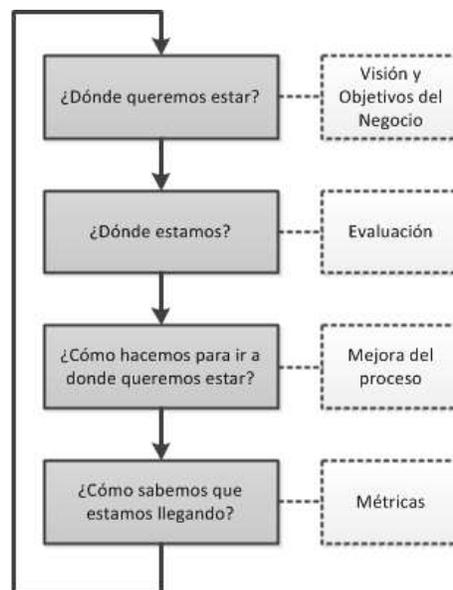


Ilustración 12: Modelo de mejora de proceso<sup>9</sup>

<sup>9</sup> Tomado de Bon, J. V. (2008). *Fundamentos de Gestión de Servicios TI basados en ITIL*. itSMF Internacional.

A continuación se establece una respuesta a cada interrogante planteada en el Modelo de mejora de proceso de la Ilustración 12.

¿Dónde queremos estar?	<i>Realizando una gestión adecuada de los archivos digitales que hacen parte de los valores intangibles utilizados para el tratamiento de la información empresarial.</i>
¿Dónde estamos?	<i>Generalmente se tiene dependencia de los archivos físicos, con una marcada tendencia hacia la utilización de las TIC's para tratar los archivos pero no se formalizan procesos para garantizar disponibilidad y respaldo de la información, se deja en cada usuario la responsabilidad sobre la gestión de los archivos; es decir que no se orienta al usuario sobre cómo crear, almacenar, organizar ni respaldar la información.</i>
¿Cómo hacemos para ir a donde queremos estar?	<i>Se deben definir procesos que enmarquen procedimientos claros para ayudar a la gestión de los archivos</i>
¿Cómo sabemos que hemos llegado?	<i>Estableciendo métricas respecto a la cantidad de archivos, la intervención de los usuarios y la evolución de los mismos.</i>

De las respuestas dadas anteriormente, se deben determinar los objetivos del proceso que se está definiendo tal como se muestra a continuación.

## 5.2. Objetivos

Realizar una adecuada gestión de archivos digitales donde se establezcan las actividades para determinar la forma de crear, almacenar, modificar, respaldar y recuperar la información que se trata a través de archivos digitales.

## 5.3. Caracterización del proceso

Definición	
Nombre del proceso	<i>Gestión de Archivos Digitales</i>
Subsistema	<i>Gestión de Configuraciones</i>
Responsable	<i>Se debe definir el rol de <b>Administrador de la Configuración</b> el cual debe asumir la Gestión de los Archivos Digitales.</i>
Objetivo del proceso	<i>Realizar una gestión de archivos digitales con eficiencia, donde se tenga disponibilidad, confiabilidad y respaldo sobre la información que utilizan los miembros de la organización.</i>
Alcance	<i>Determinar los procedimientos que permitan guiar la gestión eficiente de archivos digitales en la organización, definiendo esquemas de almacenamiento, sincronización respaldo y recuperación de la información.</i>
Clasificación	<i>Este proceso está considerado como un proceso de apoyo para fortalecer la entrega de valor de los demás procesos misionales y estratégicos de la organización.</i>

#### 5.4. Identificación de actores

Actores	
Usuario Final	<i>Son aquellas personas que pertenecen a la organización y que tienen la responsabilidad de gestionar archivos digitales sin importar la unidad de negocio a la cual pertenecen.</i>
Administrador de Repositorio	<i>Son las personas que asumen la responsabilidad de controlar quién debe acceder a ciertas partes del repositorio de información y con diferentes permisos como son la lectura y la escritura; debe existir como mínimo un administrador de repositorio por cada unidad de negocio para facilitar el</i>

		<i>control.</i>
Administrador de la Configuración		<p><i>Es la persona con alto grado de responsabilidad quien establece en compañía con los administradores de repositorios de información, las políticas sobre la estructura de los repositorios de información y los permisos que deben existir sobre la misma. También debe especificar la periodicidad de las copias de seguridad y detallar el plan de restauración que sea llevado a cabo cumpliendo los requerimientos propios del negocio.</i></p> <p><i>El administrador de la configuración debe entregar el Plan de Gestión de Archivos Digitales, el cual debe incluir los detalles de operación técnica y metodológica sobre cómo operar los archivos digitales en la organización.</i></p>
Departamento de Sistemas	de	<p><i>El departamento de sistemas hace referencia a la unidad de apoyo tecnológico que debe existir en cada organización que ha incluido el uso de las TIC's como estrategia de fomento de la calidad y productividad en una organización. Este departamento tiene la responsabilidad de apoyar los demás actores del sistema.</i></p>
Aseguramiento de Calidad	de	<p><i>Es el departamento de una organización que se encarga de la verificación periódica del estado de los procesos de una organización y que busca mantener el nivel de calidad que la organización se ha fijado como meta.</i></p>

## 5.5. Interacción del proceso

<b>Interacción</b>	
Proveedor	<i>Departamento de Sistemas</i>
Entradas	<ul style="list-style-type: none"> <li>• <i>Todos los archivos digitales que utilicen las unidades de negocio de una organización</i></li> <li>• <i>Acuerdos de nivel de servicio</i></li> <li>• <i>Política de seguridad</i></li> </ul>
Actividades	<ul style="list-style-type: none"> <li>• <i>Planear la gestión de archivos digitales</i></li> <li>• <i>Preparar servidor de repositorios</i></li> <li>• <i>Crear repositorio</i></li> <li>• <i>Poblar repositorio</i></li> <li>• <i>Preparar estación de trabajo cliente</i></li> <li>• <i>Generar versión</i></li> <li>• <i>Resolver conflictos</i></li> <li>• <i>Respaldar información</i></li> <li>• <i>Restaurar información</i></li> <li>• <i>Monitorear servidor de repositorios</i></li> <li>• <i>Realizar auditorías</i></li> </ul>

	<ul style="list-style-type: none"> <li>• <i>Evaluar desempeño del proceso</i></li> <li>• <i>Establecer mejoras al proceso</i></li> </ul>
Salidas	<i>Plan de Gestión con Archivos Digitales</i> <i>Archivos Digitales gestionados eficientemente</i>
Cliente	<i>Todo el personal de la organización</i>

## 5.6. Actividades del proceso Gestión de Archivos Digitales

### 5.6.1. Planear la gestión de archivos digitales

<b>Planear la Gestión de Archivos Digitales</b>	
Responsable	<i>Administrador de la Configuración</i>
Descripción	<i>Esta actividad consiste en la identificación de necesidades organizacionales para producir un documento que sirve de guía para la elaboración de la estructura de directorios de los repositorios de información; al igual que la especificación de roles y la asignación de los mismos al personal de la organización</i>
Entradas	<i>Documento con la descripción de las necesidades de las unidades de negocio</i>
Salidas	<ul style="list-style-type: none"> <li>• <i>Plan de Gestión de Archivos Digitales: Este formato contiene la estructura del repositorio y la especificación de roles para el control de acceso respectivo.</i></li> <li>• <i>Acuerdo de Nivel de Servicio (ANS): Es un documento que especifica las condiciones del servicio que se ofrece a los usuarios finales de la organización.</i></li> </ul>
PHVA	<i>Planear</i>

### 5.6.2. Preparar servidor de repositorios

<b>Preparar Servidor de Repositorios</b>	
Responsable	<i>Departamento de Sistemas</i>
Descripción	<i>Esta actividad consiste en preparar los elementos hardware y software que son necesarios para poder desplegar el servidor de repositorios</i>
Entradas	<i>Formato Requerimiento Técnico</i>
Salidas	<i>Servicio funcional</i>
PHVA	<i>Hacer</i>

### 5.6.3. Crear repositorio

<b>Crear Repositorio</b>	
Responsable	<i>Departamento de Sistemas</i>
Descripción	<i>Esta actividad consiste en lleva a cabo una labor técnica donde se crea el repositorio de información en el servidor destinado para esta función, además, se crean los usuarios y se asignan sus respectivos permisos</i>
Entradas	<ul style="list-style-type: none"> <li>• <i>Formato Creación Repositorio</i></li> <li>• <i>Formato Usuarios y Permisos de Repositorio</i></li> </ul>
Salidas	<i>Repositorio funcional y disponible a los usuarios especificados en los formatos de entrada.</i>
PHVA	<i>Hacer</i>

### 5.6.4. Poblar repositorio

<b>Poblar Repositorio</b>	
Responsable	<i>Administrador de Repositorio</i>
Descripción	<i>Esta actividad consiste en la ubicación de la información inicial o información base dentro del repositorio de tal forma que esté distribuida según se estableció en el Plan de Gestión de Archivos Digitales</i>
Entradas	<i>Información base para realizar el poblado inicial del repositorio</i>
Salidas	<i>Repositorio poblado y primera versión creada como una etiqueta de línea base</i>
PHVA	<i>Hacer</i>

#### 5.6.5. Preparar Estación de Trabajo Cliente

<b>Preparar Estación de Trabajo Cliente</b>	
Responsable	<i>Departamento de Sistemas</i>
Descripción	<i>Esta actividad consiste en un procedimiento técnico centrado en la preparación de las estaciones de trabajo en las cuales se integra con la herramienta adecuada para poder realizar la sincronización de la información con el servidor central, además se debe realizar con los usuarios finales una capacitación sobre la forma de utilizar dicha herramienta</i>
Entradas	<i>Formato Requerimiento técnico</i>
Salidas	<ul style="list-style-type: none"> <li>• <i>Estación de trabajo debidamente configurada con la herramienta para realizar la interacción con el repositorio de información.</i></li> <li>• <i>Usuario final capacitado para interactuar con la herramienta de la forma adecuada.</i></li> </ul>

PHVA	<i>Hacer</i>
------	--------------

### 5.6.6. Interactuar con archivos digitales

<b>Interactuar con Archivos Digitales</b>	
Responsable	<i>Usuario Final</i>
Descripción	<i>Esta actividad consiste la interacción que realizan las personas de las organización al crear, modificar y eliminar información de cualquier índole a través de archivos digitales que deben ser almacenados inicialmente en lo que se denomina copia de trabajo</i>
Entradas	<i>Estación de trabajo funcional</i>
Salidas	<i>Interacción con archivos digitales</i>
PHVA	<i>Hacer</i>

### 5.6.7. Sincronizar archivos digitales

<b>Sincronizar Archivos Digitales</b>	
Responsable	<i>Usuario Final</i>
Descripción	<i>Esta actividad consiste la sincronización de los archivos entre la copia de trabajo del usuario que se encuentra su respectiva estación de trabajo y el servidor central de repositorios. Para realizar esta función es necesario que al usuario se le asigne un respectivo rol en el repositorio de información el cual determinará el alcance sobre el mismo.</i>
Entradas	<i>Copia de trabajo</i>
Salidas	<i>Sincronización de archivos digitales</i>
PHVA	<i>Hacer</i>

**5.6.8. Generar Versión**

<b>Generar Versión</b>	
Responsable	<i>Administrador de Repositorio</i>
Descripción	<i>Esta actividad consiste en realizar una copia de los archivos que se consideran importantes en un determinado tiempo para una unidad de negocio en especial. Uno de los usos más comunes debe ser el cumplimiento de hitos dentro de los cronogramas que tenga establecidos los proyectos de la organización, de esta forma se recuperarán rápidamente los archivos digitales que conforman los entregables de un proyecto específico en una organización, dando cumplimiento de esta forma a exigencias de calidad en cuanto a la conservación de información relevante para los negocios</i>
Entradas	<i>Formato Generar Versión</i>
Salidas	<i>Versión de la documentación creada y debidamente etiquetada</i>
PHVA	<i>Hacer</i>

**5.6.9. Resolver Conflictos**

<b>Resolver Conflictos</b>	
Responsable	<i>Administrador de Repositorio</i>
Descripción	<i>Esta actividad consiste en la conciliación de posibles conflictos que se presentan cuando se realizan cambios sobre archivos y no se tiene claro cuál cambio debe permanecer como el adecuado para el archivo o los archivos específicos. Es esta</i>

	<i>caso debe ser el Administrador de Repositorio quien realice este ajuste debido al amplio conocimiento de la unidad de negocio que este posee</i>
Entradas	<i>Situación de conflicto con el contenido de archivos en una unidad de negocio</i>
Salidas	<i>Conciliación respecto al contenido de la información que debe llevar un archivo u otro</i>
PHVA	<i>Hacer</i>

#### 5.6.10. Respalda información

<b>Respalda Información</b>	
Responsable	<i>Departamento de Sistemas</i>
Descripción	<i>Consiste en un procedimiento técnico de copia de los repositorios de información. Esta actividad debe estar automatizada para garantizar que se realicen las copias en el momento específico</i>
Entradas	<i>Formato Copia de seguridad de información</i>
Salidas	<i>Copias de seguridad realizadas con la frecuencia adecuada según lo especificado en el formato de Copia de seguridad de información</i>
PHVA	<i>Hacer</i>

#### 5.6.11. Restaurar Información

<b>Restaurar Información</b>	
Responsable	<i>Departamento de Sistemas</i>
Descripción	<i>Esta actividad consiste en la localización de una copia de seguridad de una fecha específica y realizar su correspondiente restauración</i>
Entradas	<i>Formato Restauración de información</i>
Salidas	<i>Restauración de la información especificada</i>
PHVA	<i>Hacer</i>

#### 5.6.12. Monitorear el servidor de repositorios

<b>Monitorear el Servidor de Repositorios</b>	
Responsable	<i>Departamento de Sistemas</i>
Descripción	<i>Esta actividad proactiva consiste en la revisión constante de la disponibilidad del servidor de repositorios de información donde se publican los archivos digitales de la organización. Se deben utilizar herramientas software que faciliten el monitoreo constante y automatizado</i>
Entradas	<i>Formato Monitoreo de Servidor y Servicio</i>
Salidas	<i>Verificación constante del estado del servidor de repositorios de la organización</i>
PHVA	<i>Verificar</i>

#### 5.6.13. Realizar auditorías

<b>Realizar Auditorías</b>	
Responsable	<i>Aseguramiento de calidad</i>
Descripción	<i>Esta actividad consiste en la verificación de todo el proceso de Gestión de Archivos Digitales, donde se determina si existe un cumplimiento adecuado del mismo o no. En el caso de existir inconsistencias, se genera una <b>no conformidad</b> que suscita una Acción Preventiva o Acción Correctiva.</i>
Entradas	<i>Formato Auditoría de Proceso</i>
Salidas	<i>Diagnóstico del proceso con conformidad o no conformidad según sea el caso</i>
PHVA	<i>Verificar</i>

#### 5.6.14. Evaluar desempeño del proceso

<b>Evaluar Desempeño del Proceso</b>	
Responsable	<i>Aseguramiento de calidad</i>
Descripción	<i>Esta actividad consiste en la verificar que el proceso de Gestión de Archivos Digitales, se está realizando dentro de los Acuerdos de Niveles de Servicio que fueron definidos en el Plan de Gestión de Archivos Digitales</i>
Entradas	<i>Formato Evaluación Desempeño de proceso</i>
Salidas	<i>Diagnóstico del proceso con conformidad o no conformidad según sea el caso.</i>
PHVA	<i>Verificar</i>

#### 5.6.15. Establecer mejora del proceso

<b>Establecer Mejora del Proceso</b>	
Responsable	<i>Administrador de la Configuración</i>
Descripción	<i>Esta actividad consiste en la definición de acciones de mejora para proceso según la evolución y el resultado de las evaluaciones de desempeño y las auditorías</i>
Entradas	<i>Resultado de auditorías Resultado de evaluación de desempeño</i>
Salidas	<i>Reforma al Plan de Gestión de Archivos Digitales</i>
PHVA	<i>Actuar</i>

#### 5.6.16. Criterios de calidad del proceso

<b>Criterios de Calidad</b>	
Disponibilidad	<i>La información debe estar dispuesta para sea utilizada en el momento que el usuario legítimo así lo disponga mitigando la dependencia de aspectos técnicos tales como: hardware específico, electricidad, red de datos, servidores, etc.</i>
Control de acceso	<i>El acceso a los archivos debe estar limitado sólo a los usuarios legítimos según perfiles de usuarios claramente definidos en la organización. Ejemplo: Gerente, administrador, secretaria, auxiliar, etc.</i>
Trazabilidad	<i>Los archivos deben almacenarse de tal forma que permita conocerse datos históricos relacionados tales como quién y cuándo ha realizado cambios sobre los archivos.</i>
Respaldo	<i>Los archivos deben someterse a procedimientos de respaldo que aseguren la recuperación de los archivos en caso de pérdida de información. Esta actividad debe ser una responsabilidad que no debe recaer sobre los usuarios finales. Para este procedimiento debe ser realizado por personal</i>

	<i>técnico calificado además de la utilización de herramientas con alto grado de automatización.</i>
--	--

### 5.7. Recursos del proceso

<b>Recursos</b>	
Humanos	<i>Es necesario contar con personas que tengan sentido de responsabilidad para que acepten los términos planteados por el proceso y operen la herramienta cliente de la forma adecuada garantizando la sincronización de la información entre las estaciones cliente y el servidor de repositorios de información.</i>
Hardware	<ul style="list-style-type: none"> <li>• <b>Cliente:</b> <i>Las personas que interactúan creando, modificando y eliminando archivos digitales necesariamente deben contar con una estación de trabajo de escritorio o portátil desde la cual puedan acceder al repositorio.</i></li> <li>• <b>Servidor:</b> <i>Es necesario contar con el hardware para servidor que brinde condiciones ideales de alta disponibilidad, rendimiento y capacidad necesaria para satisfacer las necesidades puntuales de la organización.</i></li> </ul>
Software	<ul style="list-style-type: none"> <li>• <b>Cliente:</b> <i>Las estaciones de trabajo deben contar con una herramienta software cliente instalada que permita la comunicación con el servidor y que brinde con eficiencia y facilidad las operaciones necesarias para sincronizar los archivos digitales con el servidor de repositorios.</i></li> <li>• <b>Servidor:</b> <i>El servidor o conjunto de servidores deben</i></li> </ul>

	<i>implementar una solución de gestión de versiones que permita la interacción de los usuarios a través de una red de datos.</i>
--	--

### 5.8. Indicador del proceso

Indicador	
Nombre Indicador	<i>Disponibilidad</i>
Formula	$100 - \frac{\text{Horas Sistema Caído}}{8760} \times 100$ <p>Donde el <b>Horas Sistema Caído</b> corresponde al número de horas que permanece el sistema de repositorios caído durante el año. La medida puede ser ajustada para el mes cambiando el valor 8760 horas por 720 horas.</p>
Meta	<i>100% Anual</i>
Unidad de medida	<i>Hora</i>
Objetivo del indicador	<i>Establecer un valor de referencia para medir la calidad del servicio prestado respecto a la cantidad de tiempo que permanece disponible la información hacia los usuarios.</i>

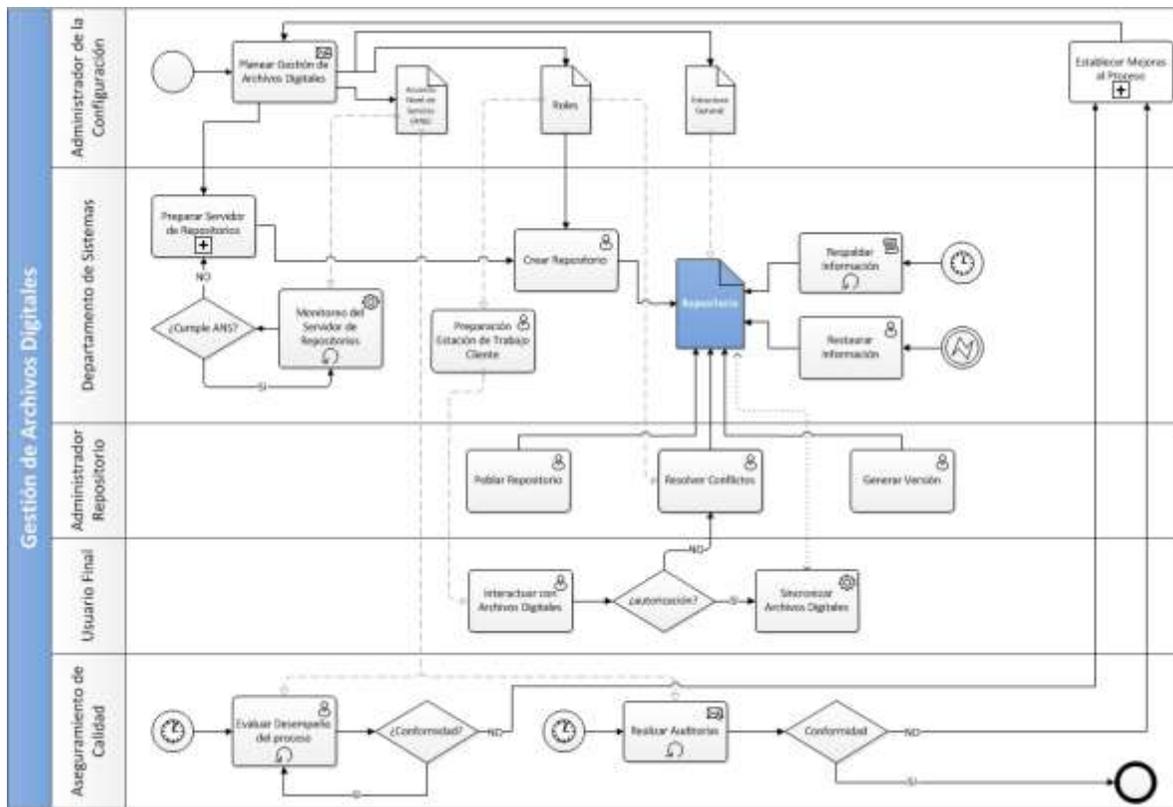
### 5.9. Procesos Relacionados

Procesos Relacionados	
Gestión de la Configuración	<i>La Gestión de la Configuración es un gran proceso que debe contener al proceso de Gestión de Archivos Digitales, debido a que los archivos digitales de una organización son considerados elementos de la</i>

	<i>configuración y los lineamientos de sobre la Administración de la Configuración en una organización deben impactar el Plan de Gestión de Archivos Digitales.</i>
Gestión de la Continuidad	<i>La Gestión de la Continuidad es un proceso que se relaciona fuertemente con el indicador del proceso asociado a la Disponibilidad. Para poder cumplir con el indicador, se deben cumplir los lineamientos que se proponen a nivel macro como política de Continuidad y Disponibilidad de la organización; esto afectará incluso a los recursos hardware y las configuraciones software que son necesarias para satisfacer las necesidades de continuidad y disponibilidad según los requerimientos mismos del negocio.</i>
Gestión de la Capacidad	<i>La Gestión de la Capacidad es un proceso que se relaciona con el establecimiento de políticas de escalabilidad sobre el servicio ofrecido y su impacto en la satisfacción a los usuarios finales.</i>
Gestión de la Seguridad	<i>Este proceso indica las pautas generales sobre la seguridad de la información</i>

### 5.10. Modelo del Proceso Gestión de Archivos Digitales con BPMN

Esquematizar un proceso con la notación para el modelado de procesos de negocio (BPMN por sus siglas en inglés) facilita la comprensión del mismo debido a que se define la forma como deben interactuar las actividades del proceso *Gestión de Archivos Digitales*. La Ilustración 13 evidencia la forma como interactúan los diferentes actores y sus respectivas actividades.



**Ilustración 13:** Modelo del Proceso Gestión de Archivos Digitales

### 5.11. Visión general del proceso

Para tener una visión generalizada del proceso es adecuado seguir las etapas que se establecen el *Círculo de Calidad de Deming* (Bon, 2008) como son Planear, Hacer, Verificar y Actuar (PHVA). En la Ilustración 14 se especifica la ubicación de las actividades que plantea el proceso de Gestión de Archivos Digitales.

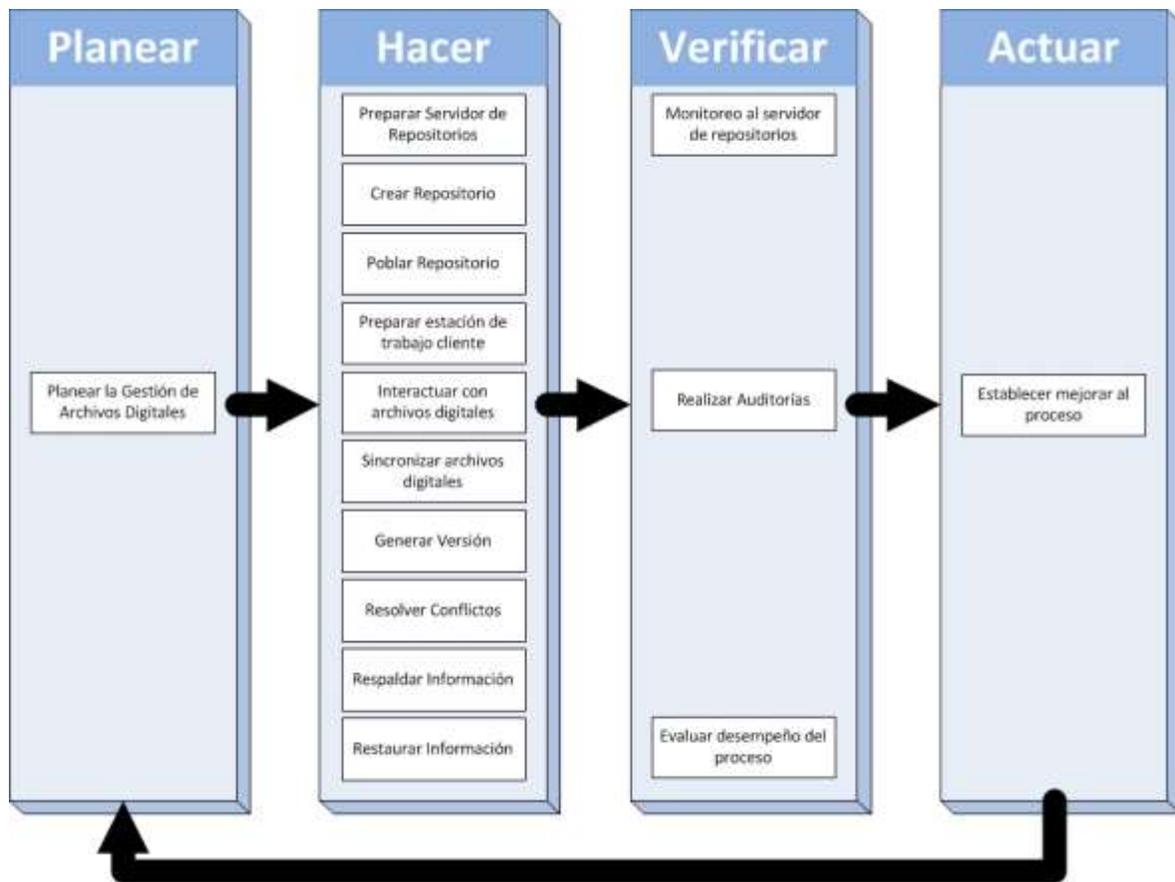


Ilustración 14: Actividades del proceso Gestión de Archivos Digitales según PHVA

A continuación se describen cada una de las etapas PHVA del *Proceso Gestión de Archivos Digitales*.

### 5.11.1. Planear

El proceso *Gestión de Archivos Digitales* inicia con la planeación misma del proceso el cual debe producir el denominado *Plan de Gestión de Archivos Digitales*. Este plan debe contener la definición de tres aspectos fundamentales como:

- **Estructura General del Repositorio:** Consiste en la especificación de un documento que debe determinar la estructura del repositorio de información en términos generales tales como:
  - Cantidad de directorios de primer nivel.
  - Cantidad de directorios de segundo nivel.
  - Esquema de nombres para archivos y directorios.
  
- **Definición de Roles de Usuario:** Consiste en especificar cuáles son los perfiles de los usuarios y el alcance de los mismos respecto al uso de la información de la organización.
  
- **Acuerdo de Nivel de Servicio (ANS):** Consiste en la especificación de los compromisos asociados al servicio que se deben cumplir al cliente en este caso los usuarios finales, con el propósito de garantizar la calidad del servicio.

#### 5.11.2. Hacer

Consiste en la realización de las actividades correspondientes a la ejecución misma del proceso una vez ha sido planeado adecuadamente. Inicia con la preparación de la infraestructura hardware y software según los lineamientos del Plan de Gestión de Archivos Digitales. En esta etapa se procede a crear los repositorios, poblarlos con información y publicarlos para que puedan estar listos a ser utilizados por los usuarios finales.

Una vez los usuarios finales interactúan con los archivos digitales, se tendrán copias de la información (copias de trabajo) en las estaciones de los usuarios finales, quienes deberán sincronizarlas con el servidor central si se cuenta con el rol adecuado para hacerlo; también se deben realizar otras actividades como la generación de versiones cuando los administradores de repositorios así lo consideren, resolver conflictos si

existieran y finalmente proceder a realizar copias de seguridad para estar preparados en caso de ser necesario la restauración de la información.

### **5.11.3. Verificar**

Esta etapa consiste en la realización de un conjunto de actividades que permiten determinar si se está cumpliendo o no el proceso de Gestión de Archivos Digitales tal como fue establecido en el plan de gestión. También se verifica si se están cumpliendo o no los acuerdos de niveles de servicio (ANS) y en el caso de existir inconsistencia se dejan las evidencias correspondientes a través de **no conformidades** en el sistema de gestión de calidad, las cuales deberán ser levantadas en el tiempo acordado entre el actor responsable y los delegados de aseguramiento de calidad.

### **5.11.4. Actuar**

Esta etapa consisten en la realización de Acciones de Mejora o Acciones Correctivas según sea el caso para responder al levantamiento de posibles **no conformidades** encontradas en la etapa de verificación del proceso y es esto lo que permite la realización de un ciclo de mejora continua del proceso debido a que las acciones tomadas en esta etapa impactan directamente al Plan de Gestión de Archivos Digitales y por consiguiente a todo el proceso en la siguiente iteración del proceso de calidad.

## 6. Herramientas para implementar un Sistema de control de versiones

Para realizar el proceso de control de versiones existen diversas herramientas; algunas son software comercial y otras tantas son consideradas software libre. Los beneficios del software libre como son el acceso al código fuente, la posibilidad de estudiarlo, modificarlo y redistribuirlo son desde todo punto de vista grandes argumentos para que el presente proyecto se centre este conjunto de software.

Como resultado de estudios comparativos realizados a las herramientas de software libre que implementan sistemas de control de versiones (Stansberry, 2008) se puede extraer la siguiente información:

### 6.1. CVS

CVS (Concurrent Version System), es el sistema de control de versiones más antiguo que existe. Es software libre; su primera versión fue liberada en 1986; debido a esto, es ampliamente conocido y usado por una gran comunidad hasta el punto de considerarse el estándar de facto en los sistemas de control de versiones.

Algunas de las principales ventajas de CVS son las siguientes (Vesperman, 2007):

- Cualquier revisión de un archivo puede ser recuperada, vista y cambiada.
- Pueden ser mostradas las diferencias entre dos revisiones cualesquiera.
- Múltiples usuarios pueden trabajar simultáneamente sobre el mismo proyecto o archivo sin sobrescribir los cambios que realiza el uno al otro.

- Los usuarios pueden estar ubicados en pequeñas o grandes redes de datos.

CVS utiliza un repositorio de información centralizada que se basa en archivos RCS (*Revision Control System*) (Vesperman, 2007). Cada archivo conectado a CVS es un archivo ordinario que contiene alguna información adicional. Al ser un sistema basado en archivos presenta deficiencias en cuanto al rendimiento general del en comparación con otros sistemas que implementan su arquitectura a través de bases de datos relacionales. Otra limitación ampliamente conocida es el hecho de no poder renombrar archivos.

CVS manipula adecuadamente archivos de texto y para el tratamiento de otros tipos de archivos necesita configuraciones especiales (Vesperman, 2007), quedando aún más limitado para su control eficiente.

Esta herramienta utiliza una arquitectura cliente-servidor, con los archivos almacenados en un el servidor central.

La principal complicación con CVS es que maneja en forma deficiente los archivos binarios. Esto representa limitaciones en los proyectos que deben utilizar este tipo de herramienta, quedando en su gran mayoría enfocado en proyectos de desarrollo de software donde los archivos que se desean versionar obedecen a archivos código fuente que generalmente son archivos de texto plano.

## **6.2. Subversion**

*Subversion* también conocido como SVN es un sistema de control de versiones que fue diseñado en 2001 para ser el sucesor de CVS y corregir algunos de los problemas más notables de CVS siendo ampliamente compatible con este último, dando como

---

resultado un potente, fácil de usar y muy flexible sistema de control de versiones. De igual forma es software libre. (Collins-Sussan, Fitzpatrick, & Pilato, 2008).

*Subversion* utiliza una base de datos relacional para la gestión de los archivos la base de datos BerkeleyDB, lo que permite mejorar el desempeño en las operaciones generales del sistema (Mason, 2006).

Diferentes tipos de archivos pueden ser gestionados a través de *Subversion* sin tener que realizar configuraciones adicionales sobre la herramienta, lo que constituye un gran avance frente a CVS (Nagel, 2005).

*Subversion* puede realizar revisiones sobre directorios conservando su historial de cambios sin problema alguno, debido a que trata los directorios como si fueran archivos convencionales (Mason, 2006).

Por todo lo anterior, *Subversion* es probablemente el sistema de control de versiones actual con la más amplia adopción, incluso muchos se han migrado desde CVS por su amplia compatibilidad conceptual y debido a que la mayoría de los proyectos de código abierto utilizan *Subversion* como repositorio, porque otros proyectos de mayor envergadura, como SourceForge, Apache, Python, Ruby y muchos otros también lo utilizan. Google Code utiliza *Subversion* exclusivamente para distribuir el código entre los desarrolladores (Stansberry, 2008).

### **6.3. Git**

Git es un sistema de control de versiones libre y de código abierto que fue publicado en el año 2005 y que está iniciando rápidamente su acenso (Stansberry, 2008). Git fue diseñado por Linus Torvalds el creador del Kernel Linux. Los principios de Git buscan eficiencia y confiabilidad en el tratamiento de las versiones cuando los proyectos tiene

un gran número de archivos y los usuarios se encuentran distribuidos geográficamente distantes tal como sucede con el Kernel Linux (Swicegood, 2008).

Git permite que los repositorios creados con *Subversion* se pueden usar directamente con una herramienta llamada *git-svn*.

Existen proyectos importantes que ya están utilizando a Git como sistema de control de versiones entre ellos Git, Linux Kernel, Perl, Gnome, QT, Fedora, Debian, X.org, entre otros.

#### **6.4. Mercurial**

Mercurial es un sistema de control de versiones que al igual que Git ofrecen la posibilidad de utilizarlo en un esquema distribuido para realizar la gestión de las revisiones de un proyecto (O'Sullivan, 2007).

Mercurial fue publicado en el año 2005 y pertenece al tipo de herramientas de control de revisiones que operan tal como lo hacen los sistemas *peer-to-peer*, es decir, que la información no se encuentra almacenada en un solo servidor central y han permitido que los usuarios distribuyan sus datos (O'Sullivan, 2007).

Mercurial se destaca por ser una herramienta rápida y poderosa además de estar implementado principalmente haciendo uso del lenguaje de programación Python, lo cual permite extender sus funciones con facilidad utilizando scripts de Python. (O'Sullivan, 2007).

## 6.5. Bazaar

Bazaar es otro nuevo sistema de control de versiones distribuido, que facilita que varias personas puedan trabajar de forma conjunta en proyectos de software; fue publicado en el año 2007, lo que lo convierte en uno de los sistemas de control de versiones más recientes. (Canonical Ltd, 2009).

Las principales características de Bazaar son (Canonical Ltd, 2009):

- **Adaptable:** Soporta administración de código centralizado y distribuido con facilidad. Cambiarse desde una herramienta de control de versiones centralizada no requiere que el equipo de personas cambie su dinámica de trabajo.
- **Productivo:** La herramienta de línea de comandos hace énfasis en la facilidad de uso: por ejemplo, cada comando tiene ayuda que explica la sintaxis y opciones. Las interfaces gráficas de usuario (GUI) para Bazaar están disponible para Windows, Linux y OS X.
- **Rápido y Eficiente:** Bazaar tiene un desempeño eficiente en estructuras de directorio grandes, a través de redes de datos lentas y proyectos con un historial de revisiones muy grande.
- **Único:** Bazar ofrece una combinación única de características incluyendo el renombrado de archivos y directorios conservando el seguimiento de cambios.

### Cuadro resumen de las herramientas para el control de versiones

Característica	Herramientas Software Libre para el control de versiones				
	CVS	Subersión (SVN)	Git	Mercurial	Bazaar
Arquitectura	Centralizada	Centralizada	Distribuida	Distribuida	Distribuida
Fortaleza	Ampliamente conocido en el mundo por ser el precursor de la tecnología del control de versiones.  Amplia documentación existente.	Corrige muchos de los errores de CVS y facilita la migración hacia él.  Es uno de los más fuertes en el mundo, debido a la herencia recibido de CVS.	Eficiente y confiable en el tratamiento de las versiones cuando los proyectos con gran número de archivos  Compatible con SNV.	Ofrece la posibilidad de utilizarlo en un esquema distribuido.  Extensible a través de scripts de Python.	Adaptable entre centralizado y distribuido.  Eficiente en el almacenamiento redes lentas.
Formato del repositorio	Basado en archivos	Base de datos	Base de datos	Base de datos	Base de datos
Renombrar archivos y directorios	No	Si	Si	Si	Si
Soporte para archivos binarios	No	Si	Si	Si	Si
Multiplataforma	Si	Si	Si	Si	Si
Año de liberación	1986	2001	2005	2005	2007
Aceptación	Muy popular debido a que fue el primer sistema de control de versiones disponibles.	Una gran comunidad de usuarios alrededor del mundo. Muchos usuarios de CVS siguen migrado hacia SVN sin inconvenientes.	Muy aceptado en proyectos grandes de desarrollo.  Está ganando adeptos cada vez con más frecuencia.	Comunidad reducida de usuarios en comparación con Git y SVN.	Comunidad reducida de usuarios en comparación con Git y SVN.

#### 6.6. Selección de la herramienta para el control de versiones

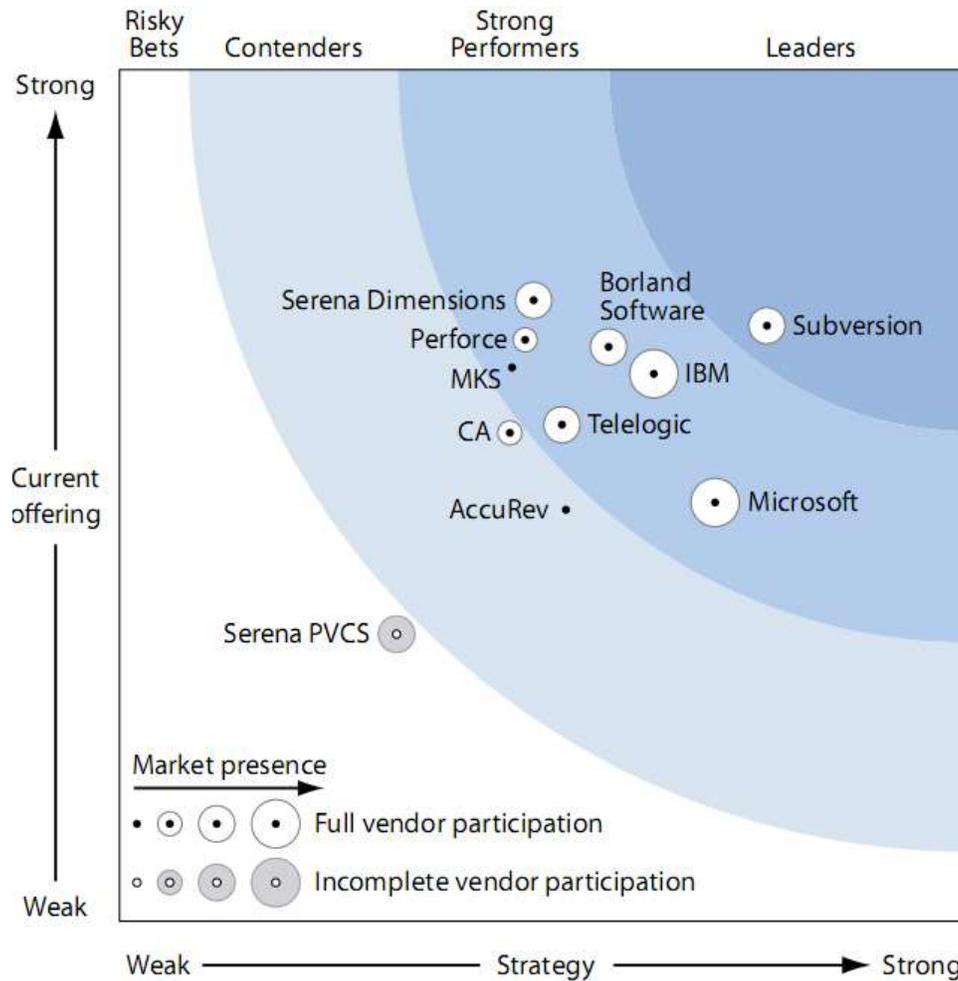
El proceso de *Gestión de Archivos Digitales* que describe el presente trabajo es completamente independiente de la herramienta para el control de versiones seleccionada; es decir, puede ser implementado con cada uno de las herramientas anteriormente descritas a pesar de que cada herramienta tiene un grupo objetivo de usuarios a los que les satisface cada su funciones específicas a plenitud. Sin embargo, se ha tomado la decisión de elegir *Subversion* para implementar el prototipo funcional a través de máquinas virtuales. Las razones que ha llevado a tomar esta decisión se exponen a continuación.

### 6.6.1. Por qué elegir *Subversion*

El proyecto *Subversion* fue iniciado por un grupo de desarrolladores con amplia experiencia en CVS, los cuales sabían de las deficiencias de CVS y propusieron construir un sistema con mejores prestaciones el cual ha ido evolucionando paulatinamente desde su publicación en agosto de 2001 (Mason, 2006).

Como inicialmente se tenía una gran comunidad de usuarios que conocían y utilizaban CVS, fue una estrategia adecuada por parte de los desarrolladores de *Subversion* facilitar la compatibilidad funcional de proyectos basados en CVS, este hecho suscitó que un gran conjunto de proyectos pudieran migrarse a *Subversion* y aprovechar características adicionales como son la posibilidad de versionar directorios y metadatos (Mason, 2006).

Aunque el presente proyecto se centra en aquellas herramientas basadas específicamente en software libre, es prudente mostrar parte del informe de Forrester Research (Schwaber, Gilpin, & Stone, 2007), en el cual se compara a *Subversion* con otras herramientas propietarias y se reconoce a *Subversion* (SVN) como el líder en la categoría de sistema de control de versiones.



**Ilustración 15:** Comparación de *Subversion* con otras herramientas para el control de versiones<sup>10</sup>

Según el informe Forrester Research (Schwaber, Gilpin, & Stone, 2007), *Subversion* es el líder de los SCM independientes el cual ha calado hondo en las organizaciones de todos los tamaños alrededor del mundo. A continuación se muestran los criterios que se utilizaron para la valoración:

<sup>10</sup> Tomado de The Forrester Wave. Software Change And Configuration Management.

	Forrester's Weighting	AccuRev	Borland Software	CA	IBM	Microsoft	MKS	Perforce	Serena Dimensions	Subversion	Telelogic
<b>CURRENT OFFERING</b>	50%	2.41	3.38	2.87	3.22	2.47	3.26	3.41	3.65	3.50	2.93
Platform support	10%	2.76	1.53	3.81	3.66	0.00	2.75	3.38	3.77	3.78	2.58
Scalability	10%	0.70	3.80	3.80	4.30	2.30	2.30	3.80	3.60	4.10	4.40
Performance	0%	1.98	3.36	1.88	3.02	3.14	3.30	3.98	3.36	2.52	0.00
Administration	15%	3.00	3.50	1.50	1.50	2.50	3.00	5.00	2.50	5.00	0.00
Geographic distribution	10%	3.20	3.40	1.00	1.80	2.90	3.00	3.00	3.80	3.80	2.40
Security	10%	2.93	4.10	4.00	2.94	3.47	4.75	2.93	4.43	4.45	4.39
Configuration management	25%	2.96	3.89	3.50	4.25	2.42	3.63	3.52	4.06	3.00	3.35
Change management	0%	2.53	3.78	3.56	3.59	3.44	4.04	2.00	3.54	0.37	3.82
Reporting and analytics	5%	2.60	4.10	2.10	3.60	5.00	4.90	3.10	3.70	1.70	5.00
Life-cycle integration	15%	0.90	2.60	2.70	3.20	2.50	2.50	2.10	3.45	2.00	3.08
<b>STRATEGY</b>	50%	2.62	2.86	2.30	3.13	3.50	2.30	2.38	2.43	3.79	2.59
Product strategy	35%	2.60	3.00	2.60	4.40	4.20	1.60	1.00	2.80	2.40	2.60
Corporate strategy	20%	4.00	4.00	3.50	1.75	2.50	4.00	3.50	4.00	4.50	2.50
Price	35%	2.30	2.30	1.40	2.10	3.50	2.10	3.50	0.70	5.00	2.80
R&D	10%	1.00	2.00	2.00	5.00	3.00	2.00	1.00	4.00	3.00	2.00
<b>MARKET PRESENCE</b>	0%	1.52	3.98	2.82	4.22	4.52	1.64	2.52	3.76	3.22	3.02
Installed base	60%	1.20	4.10	2.50	4.30	4.80	1.20	2.60	4.10	4.40	2.50
Financial strength	10%	3.40	3.60	2.60	2.60	2.60	2.60	3.40	1.80	0.00	4.20
Employees	10%	1.00	3.00	4.00	5.00	5.00	1.00	1.00	2.00	1.00	3.00
Support services	10%	2.60	4.60	4.60	3.80	3.80	2.60	3.20	4.20	2.80	3.00
Channel partnerships	10%	1.00	4.00	2.00	5.00	5.00	3.00	2.00	5.00	2.00	5.00

All scores are based on a scale of 0 (weak) to 5 (strong).

**Ilustración 16:** Detalles de la comparación de *Subversion* con otras herramientas para el control de versiones<sup>11</sup>

<sup>11</sup> Tomado de The Forrester Wave. Software Change And Configuration Management.

### 6.6.2. Arquitectura de *Subversion*

La Ilustración 17 muestra una visión panorámica del diseño de *Subversion*.

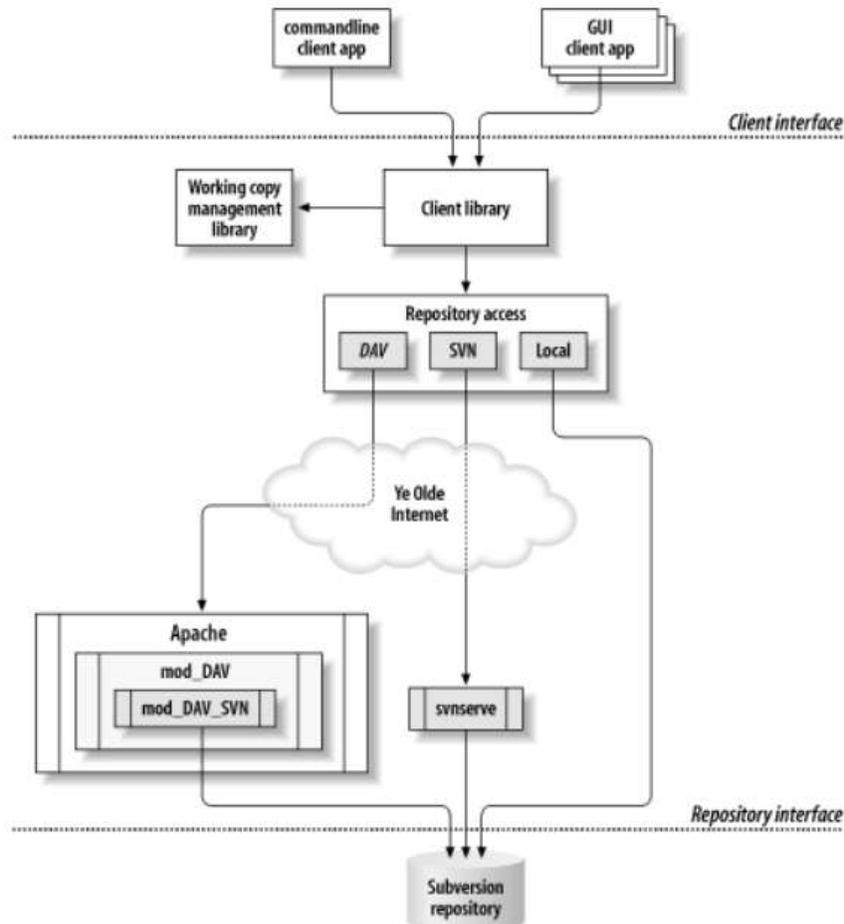


Ilustración 17: Arquitectura *Subversion*<sup>12</sup>

En el extremo inferior se encuentra un repositorio de *Subversion* que conserva todos los datos versionados. Al otro lado, hay un programa cliente *Subversion* que administra réplicas parciales de esos datos versionados (llamadas “copias de trabajo”). Entre estos extremos hay múltiples rutas a través de varias capas de acceso al repositorio. Algunas de estas rutas incluyen redes de computadores y servidores de

<sup>12</sup> Version Control with Subversion (2008).

---

red que después acceden al repositorio. Otras pasan por alto la red y acceden al repositorio directamente (Collins-Sussan, Fitzpatrick, & Pilato, 2008).

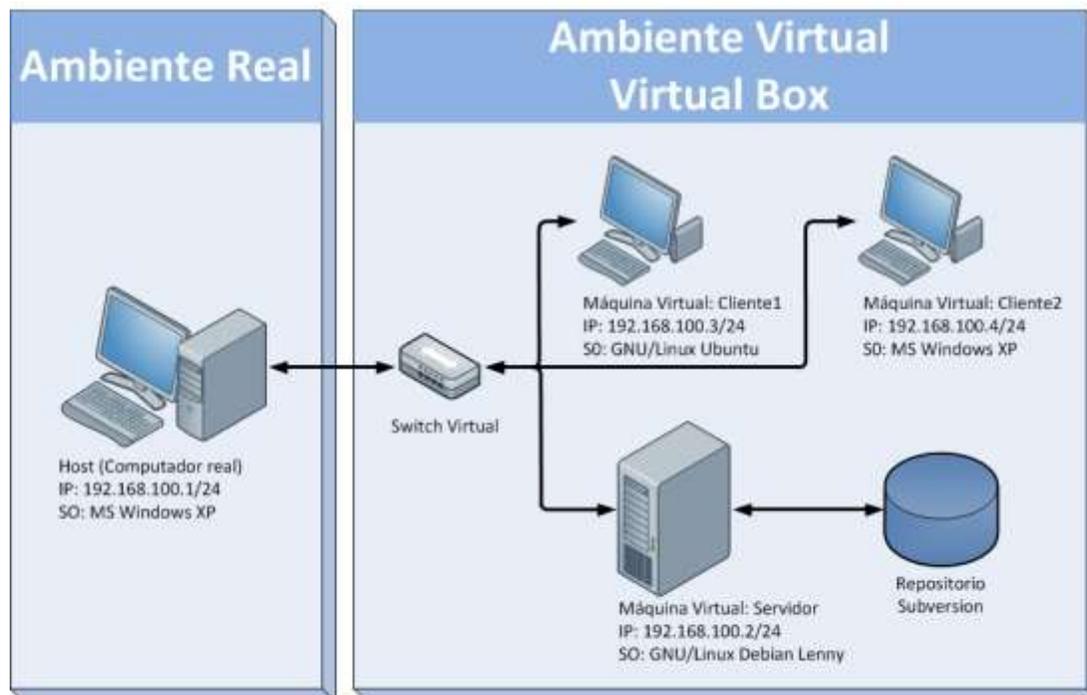
## 7. Diseño de un entorno de operación basado en máquinas virtuales

Con el propósito de validar la efectividad de la solución planteada por el presente trabajo, se diseña a continuación un entorno de operaciones basado en máquinas virtuales para instalar tanto la herramienta servidor como cliente del sistema de control de versiones. Este ambiente virtual permitirá concretar mediante un ejemplo de aplicación la forma como debe materializarse el proceso de Gestión de Archivos Digitales diseñado anteriormente.

### 7.1. Diseño del prototipo funcional

#### Relación con el proceso *Gestión de Archivos Digitales*

**Actividad relacionada:** 5.6.1 *Planear la gestión de archivos digitales*



**Ilustración 18:** Prototipo funcional - Escenario Virtual

Para poder instalar adecuadamente la herramienta *Subversion*, se debe tener presente que según su arquitectura de despliegue, deben existir como mínimo dos instancias, una para el cliente y otra para el servidor; sin embargo, para el presente proyecto fue necesario desplegar una máquina virtual para el servidor de *Subversion* y dos máquinas virtuales para las estaciones cliente cada una con la herramienta cliente adecuada según el sistema operativo, una utilizando Ubuntu GNU/Linux y la otra MS-Windows XP respectivamente; todo esto con el fin de realizar una simulación de dos usuarios que trabajan sobre el mismo repositorio. La razón de utilizar estaciones con sistema operativo MS-Windows es debido a que se quiere evidenciar la transparencia y portabilidad que ofrece esta solución pudiéndose llevar al escenario real de las empresas colombianas, donde la presencia del sistema operativo MS-Windows sigue teniendo una amplia base instalada.

La Ilustración 18 muestra el esquema general de despliegue para la solución que se desea utilizar como prototipo funcional. Básicamente se tienen dos segmentos, uno de ellos denominado “Ambiente Real” y el otro denominado “Ambiente Virtual” los cuales se explican a continuación.

#### 7.1.1. Descripción del “Ambiente Real”

El ambiente real representa a la estación de trabajo donde se realiza el hospedaje de la implementación virtual del proyecto. Está constituida por una estación de trabajo tipo PC con arquitectura X86 con las siguientes características hardware.

Descripción Ambiente Real	
Tipo de Hardware	<i>Computadora Personal (PC)</i>
Arquitectura	<i>X86</i>
CPU	<i>Intel(R) Core(TM)2 Duo CPU E7400 @ 2.80GHz</i>
RAM	<i>3 GB DDR2-667</i>

Sistema Operativo	<i>MS-Microsoft Windows XP</i>
Herramienta Utilizada	<i>Virtual Box</i>
Dirección IP	<i>192.168.100.1</i>

### 7.1.2. Descripción del “Ambiente Virtual”

El ambiente virtual representa al conjunto de estaciones que deben ser desplegados por quien desee implementar esta solución. Como se había dicho anteriormente, consta de una máquina virtual para el servidor donde se instalará la herramienta para el control de versión *Subversion* y dos máquinas virtuales para los clientes 1 y 2 respectivamente.

Para realizar la construcción de las máquinas virtuales es necesario contar con un software especializado en esta función. Existen muchas soluciones en el mercado, unas propietarias y otras tantas de código abierto o software libre. Una herramienta para virtualización muy popular, estable y además de código abierto es *Virtual Box* que ahora es desarrollada por la empresa Oracle. Virtual Box ofrece las características funcionales adecuadas para facilitar el cumplimiento de los objetivos del presente proyecto. La Ilustración 19 muestra la versión de la Virtual Box (3.2.4) utilizada en el desarrollo del proyecto.



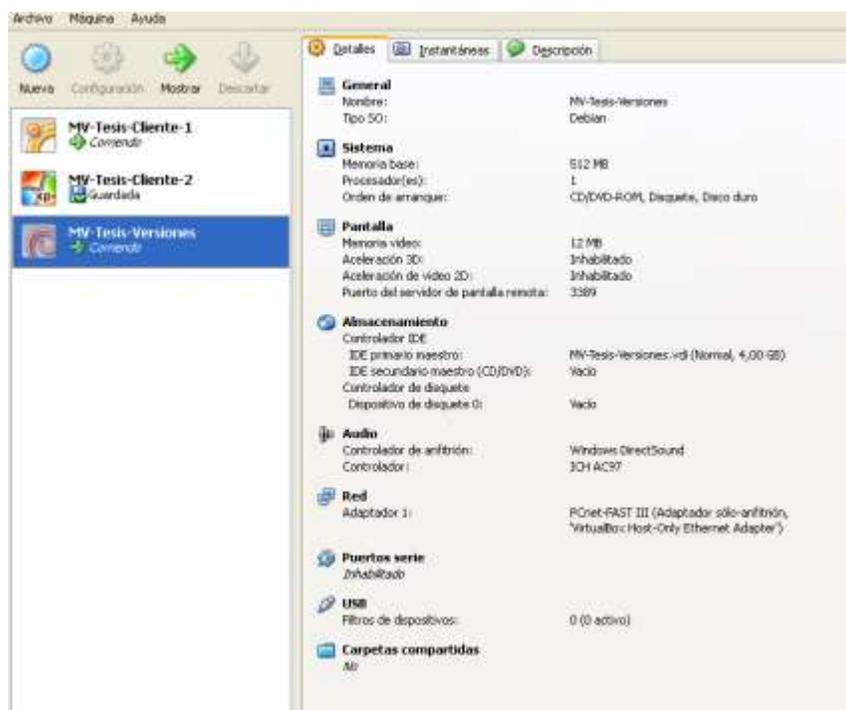
**Ilustración 19:** Oracle Virtual Box versión utilizada en el proyecto

A continuación se muestra la descripción de los detalles en las máquinas virtuales construidas para el proyecto.

### 7.1.2.1. Descripción máquina virtual Servidor

<b>Máquina Virtual: Servidor</b>	
Nombre	<i>MV-Tesis-Versiones</i>
Propósito	<i>Servidor donde se instala el sistema de control de versión Subversion y donde estarán almacenados los repositorios.</i>
Sistema Operativo	<i>Debian GNU/Linux (Versión 5.0 - Lenny)</i>
Herramienta utilizada	<i>Subversión</i>
Dirección IP	<i>192.168.100.2</i>

La Ilustración 20 muestra los detalles técnicos asociados a la configuración de la máquina virtual Servidor.



**Ilustración 20:** Detalles técnicos de la máquina virtual para el Servidor de repositorios

### 7.1.2.2. Descripción máquina virtual Cliente 1

Máquina Virtual: Cliente 1	
Nombre	<i>MV-Tesis-Cliente1</i>
Propósito	<i>Estación de trabajo que representa a los clientes que interactúan con el repositorio utilizando el sistema operativo GNU/Linux.</i>
Sistema Operativo	<i>Ubuntu GNU/Linux 10.4</i>
Herramienta utilizada	<i>RabbitVCS</i>
Dirección IP	<i>192.168.100.3</i>

La Ilustración 21 muestra los detalles técnicos asociados a la configuración de la máquina virtual que se utiliza como cliente 1.



**Ilustración 21:** Detalles técnicos de la máquina virtual para el Cliente 1

### 7.1.2.3. Descripción máquina virtual Cliente 2

Máquina Virtual: Cliente 2	
Nombre	<i>MV-Tesis-Cliente2</i>
Propósito	<i>Estación de trabajo que representa a los clientes que interactúan con el repositorio utilizando el sistema operativo MS-Windows.</i>
Sistema Operativo	<i>MS-Windows XP</i>
Herramienta utilizada	<i>TortoiseSVN</i>
Dirección IP	<i>192.168.100.4</i>

La Ilustración 22 muestra los detalles técnicos asociados a la configuración de la máquina virtual que se utiliza como cliente 2.



**Ilustración 22:** Detalles técnicos de la máquina virtual para el Cliente 2

## 7.2. Requerimiento para la instalación de *Subversion*

### Relación con el proceso *Gestión de Archivos Digitales*

**Actividad relacionada:** 5.6.2 *Preparar servidor de repositorios*

Los requerimientos relacionados con el hardware no tienen mayores exigencias técnicas, pues a nivel de CPU *Subversion* es funcional a partir de los 133 Mhz. Lo que si debe tenerse presente a la hora de realizar una estimación de capacidad para un servicio en general son las necesidades de cada negocio, es decir, la cantidad de usuarios que estarán trabajando concurrentemente sobre el servidor, el tipo y tamaño de los archivos que se utilizarán; además, si los usuarios utilizan redes de datos, se debe considerar la capacidad de transferencia que estas brindan al igual que las tendencias de demanda de recursos por parte de los usuarios.

## 7.3. Instalación de la herramienta *Subversion* en el servidor

### Relación con el Proceso *Gestión de Archivos Digitales*

**Actividad relacionada:** 5.6.2 *Preparar servidor de repositorios*

El proceso de instalación y configuración de la herramienta *Subversion*, es diferente según el sistema operativo del que se esté utilizando. A continuación se detalla dos maneras de realizar la instalación de *Subversion* sobre el sistema operativo Debian GNU/Linux.

Los procedimientos que a continuación se describen, pueden ser complementados con videos que se encuentran disponibles en un servidor disponible en Internet tal como se muestra en el anexo 2.

### 7.3.1. Instalación asistida de *Subversión*

Esta forma de realizar la instalación consiste en la utilización del gestor de paquetes disponibles en el sistema operativo, en este caso se trata de *Synaptic*. Una vez desplegado *Synaptic*, sólo basta con realizar la búsqueda del paquete tal como se muestra en la Ilustración 23. Una vez ubicado el paquete adecuado, se realiza el proceso de instalación de una forma casi transparente para el usuario.

Este tipo de instalación es muy fácil de realizar, pero limita a los usuarios a utilizar la versión de la *Subversión* que esté disponible con el sistema operativo, siendo esta versión tal vez inferior a la última liberada por el grupo de desarrolladores del proyecto *Subversión*.

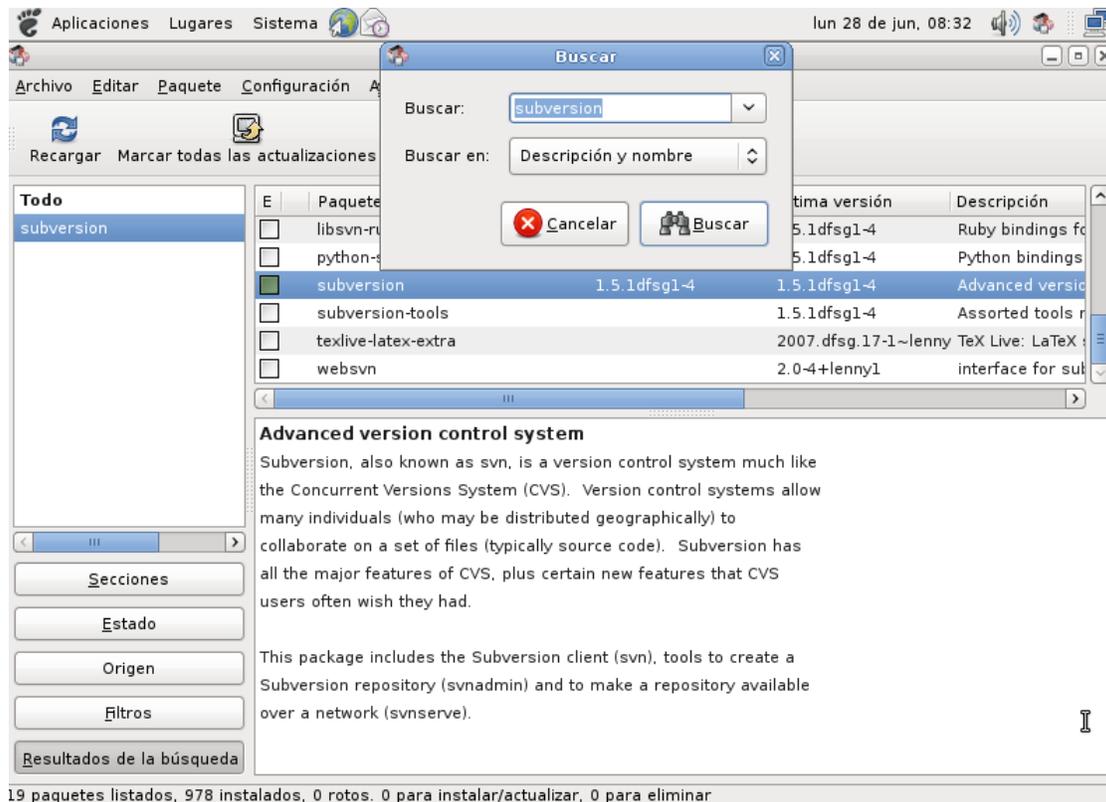
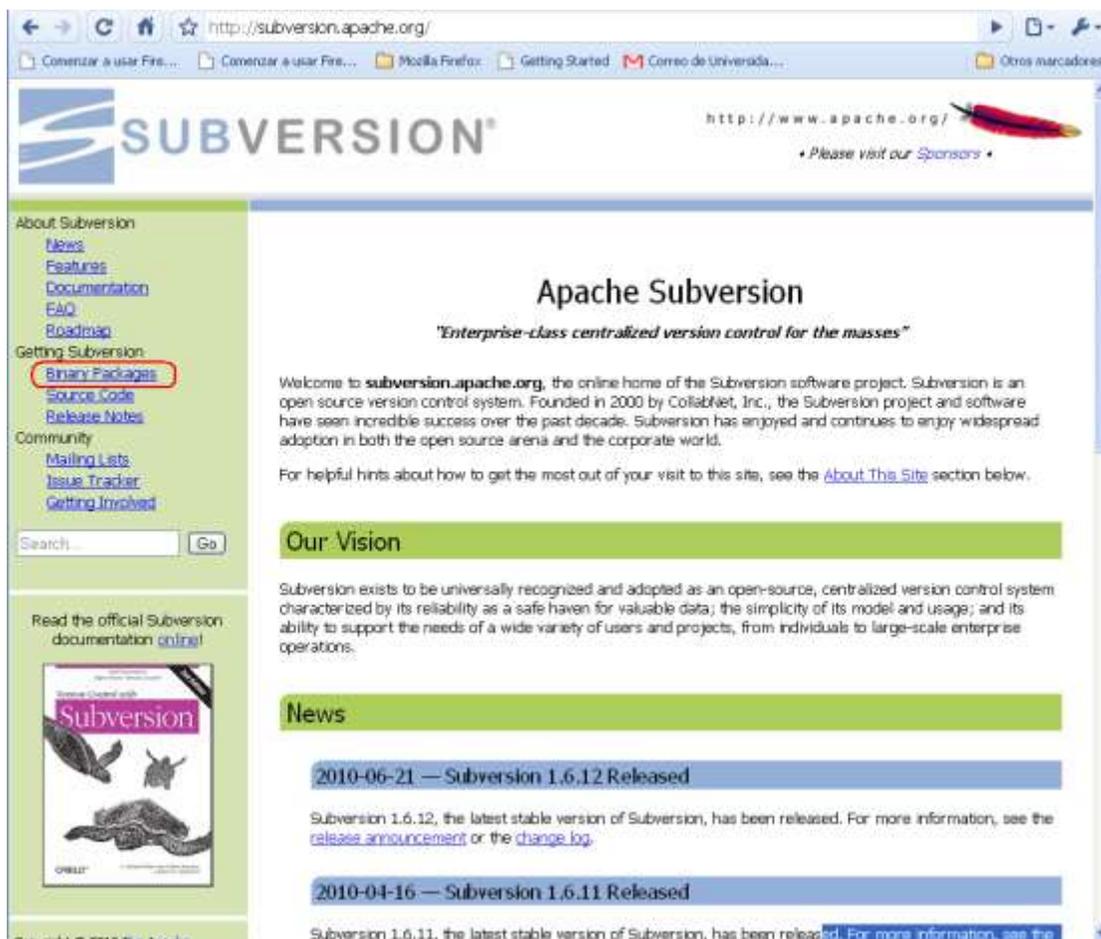


Ilustración 23: Instalación asistida de *Subversión*

Para integrar la última versión de *Subversion* liberada por el grupo de desarrollo de la herramienta al sistema de paquetes del sistema operativo, se deben realizar los pasos que se describen a continuación:

- **Paso 1:**

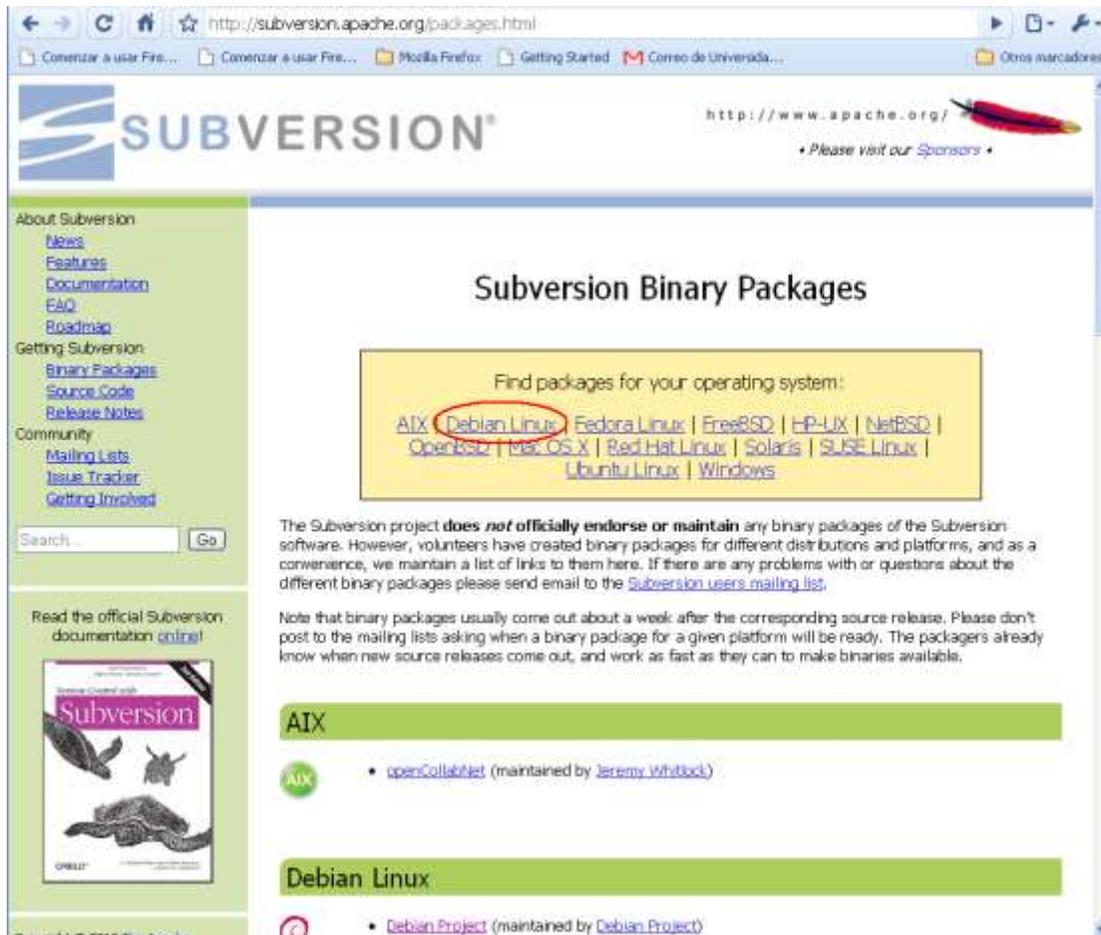
Dirigirse al sitio oficial del proyecto *Subversion*: <http://subversion.apache.org/> específicamente a la sección de “*Binary Packages*” tal como se muestra en la Ilustración 24.



**Ilustración 24:** Descarga de binarios de Subversión desde el Sitio Web oficial del proyecto

- **Paso 2:**

Los binarios que se se que se desean descargar deben están sujetos al sistema operativo que tenemos para el servidor de la solución; en este caso se trata de Debian GNU/Linux ver Ilustración 25.



**Ilustración 25:** Binarios de Subversión para Debian GNU/Linux desde el Sitio Web oficial del proyecto

- **Paso 3:**

Los binarios para Debian GNU/Linux pueden ser descargados en diversos formatos y realizar posteriormente el procedimiento de instalación manual, pero este tipo de actividad puede traer dificultades al momento de realizar el

chequeo de dependencia de las librerías y demás software que es requerido por *Subversion*. Para evitar esta situación, en el sitio web oficial de Subversión, se encuentra un Script del Shell que puede ser utilizado para automatizar el proceso de actualización del repositorio correspondientes para realizar una actualización de software que incluya las dependencias necesarias para la ejecución correcta de la herramienta *Subversion*. En la Ilustración 26 se muestra el enlace correspondiente al Script.

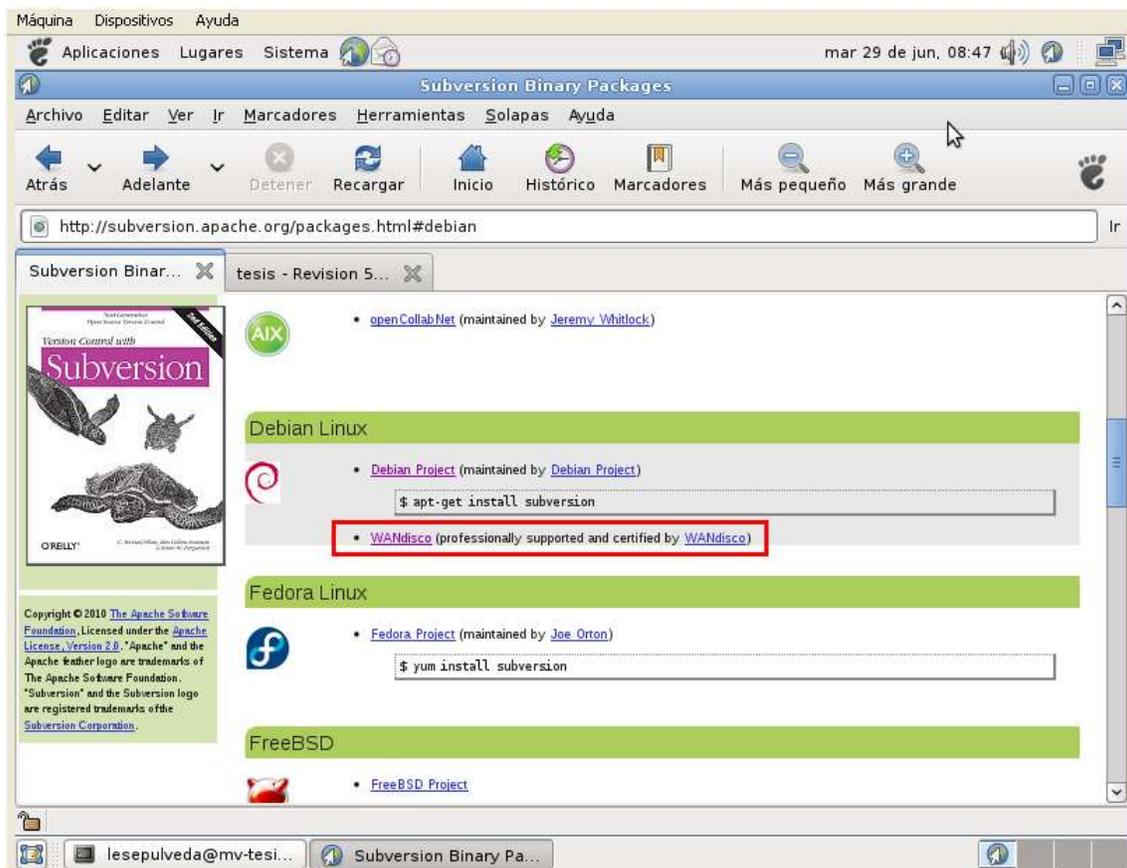


Ilustración 26: Script de Shell para la automatización de la actualización e instalación de *Subversion*.

- **Paso 4:**

Una vez de descargo el script de Shell para la automatización de la actualización de *Subversion*, se tendrá un archivo con el nombre

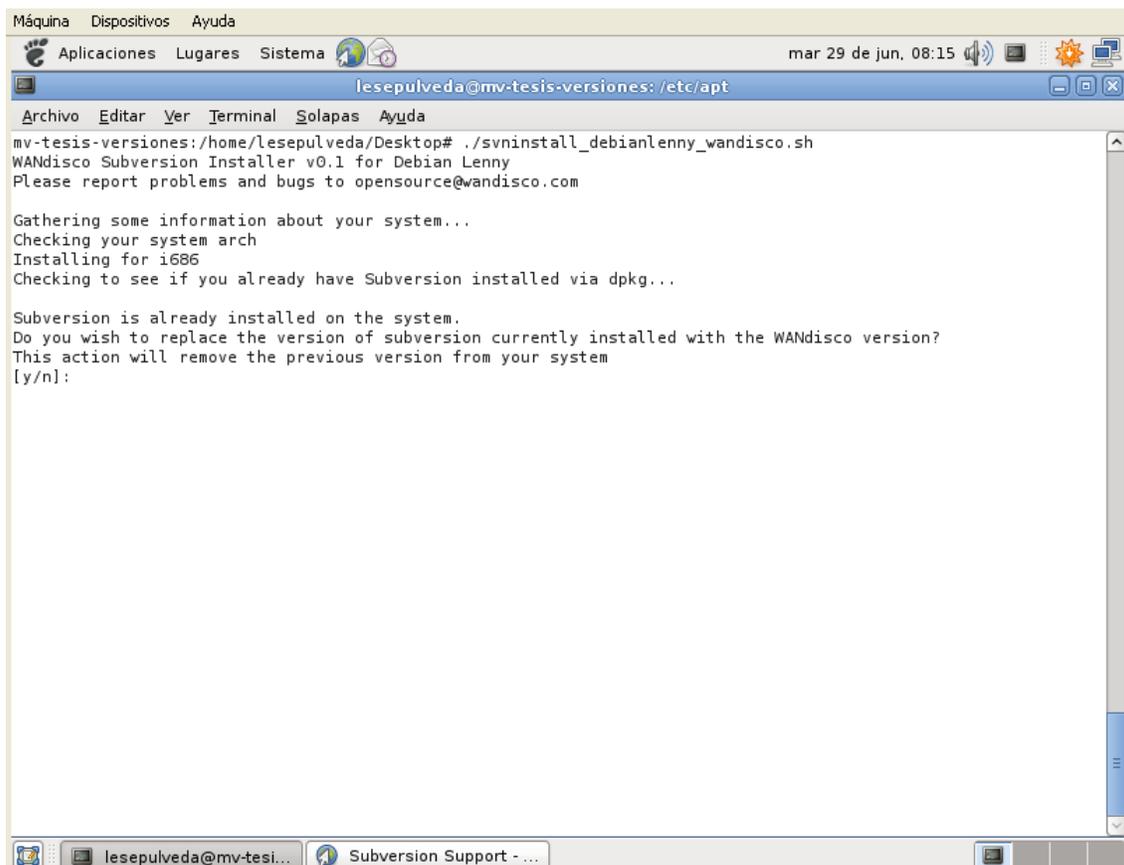
`svninstall_debianlenny_wandisco.sh` al cual se debe adicionar el premissa de ejecución a través de la instrucción siguiente:

```
# chmod u+x svninstall_debianlenny_wandisco.sh
```

Esta acción deber ser realizada desde luego por el root o un usuario del sistema operativo con altos privilegios administrativos.

- **Paso 4:**

Realizar la ejecución sobre el script de automatización para la actualización del *Subversion* de tal forma que se inicie el proceso al cual se debe estar atento para aceptar las preguntas que allí se generan tal como lo muestra la Ilustración 27.



```
Máquina Dispositivos Ayuda
Aplicaciones Lugares Sistema mar 29 de jun, 08:15
lesepulveda@mv-tesis-versiones: /etc/apt
Archivo Editar Ver Terminal Solapas Ayuda
mv-tesis-versiones:/home/lesepulveda/Desktop# ./svninstall_debianlenny_wandisco.sh
WANDisco Subversion Installer v0.1 for Debian Lenny
Please report problems and bugs to opensource@wandisco.com

Gathering some information about your system...
Checking your system arch
Installing for i686
Checking to see if you already have Subversion installed via dpkg...

Subversion is already installed on the system.
Do you wish to replace the version of subversion currently installed with the WANDisco version?
This action will remove the previous version from your system
[y/n]:
```

**Ilustración 27:** Ejecución del script de Shell `svninstall_debianlenny_wandisco.sh`

Una vez inicia la ejecución del script Shell se procede a verificación de las versiones existentes de las librerías necesarias para el funcionamiento de *Subversion*, procede eliminando las viejas versiones e instalando las nueva para finalmente instalar la última versión disponible de *Subversion*, en este caso *Subversion 1.6.11-1* tal como lo muestra la Ilustración 28.

```

Máquina Dispositivos Ayuda
Aplicaciones Lugares Sistema
lesepulveda@mv-tesis-versiones: /etc/apt
Archivo Editar Ver Terminal Solapas Ayuda
Ign http://opensource.wandisco.com lenny/svn Packages
Obj http://security.debian.org lenny/updates/contrib Sources
Ign http://volatile.debian.org lenny/volatile/main Packages/DiffIndex
Des:3 http://opensource.wandisco.com lenny/svn Packages [2647B]
Ign http://volatile.debian.org lenny/volatile/contrib Packages/DiffIndex
Ign http://volatile.debian.org lenny/volatile/main Sources/DiffIndex
Ign http://volatile.debian.org lenny/volatile/contrib Sources/DiffIndex
Des:4 http://volatile.debian.org lenny/volatile/main Packages [7653B]
Obj http://volatile.debian.org lenny/volatile/contrib Packages
Des:5 http://volatile.debian.org lenny/volatile/main Sources [3181B]
Obj http://volatile.debian.org lenny/volatile/contrib Sources
Descargados 54,7kB en 5s (9891B/s)
Leyendo lista de paquetes... Hecho

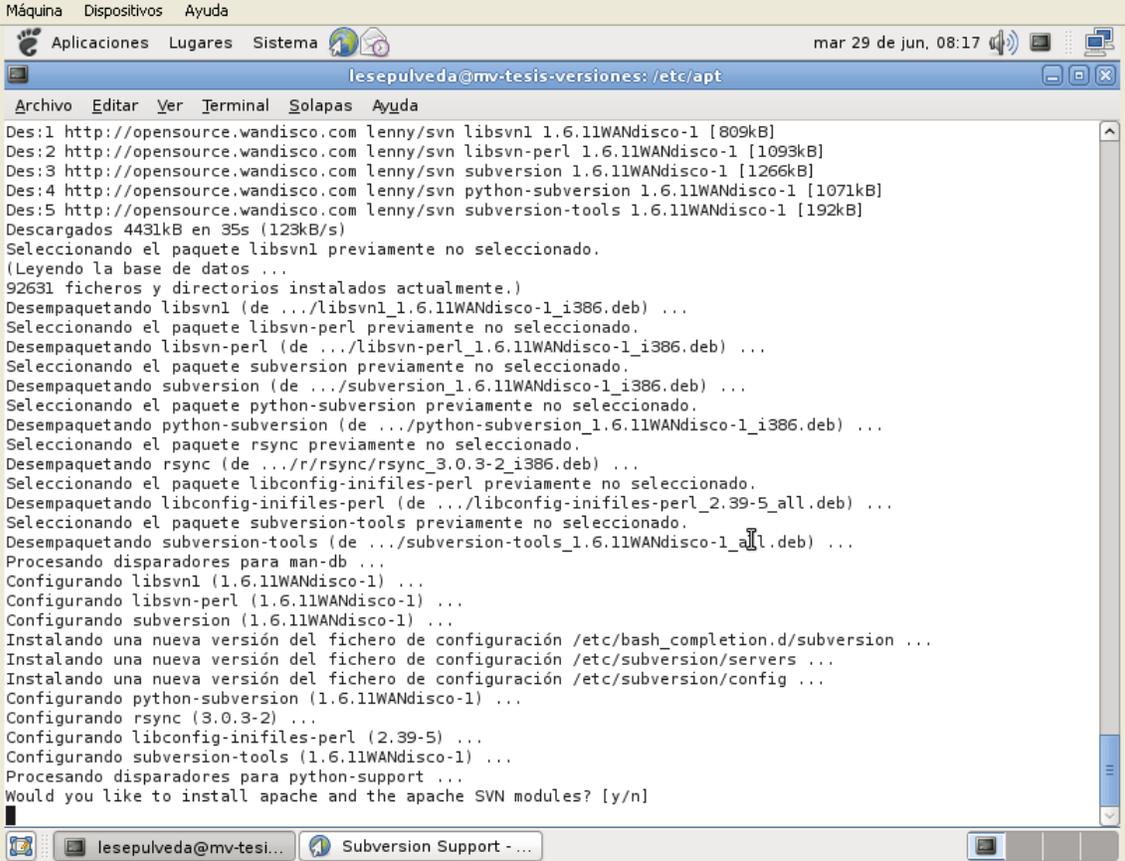
Installing Subversion 1.6.11-1

Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalaron de forma automática los siguientes paquetes y ya no son necesarios.
  libsilk-1.1-2 libserf-0-0
Utilice «apt-get autoremove» para eliminarlos.
Se instalarán los siguientes paquetes extras:
  libconfig-inifiles-perl libsvn1 rsync
Paquetes sugeridos:
  openssh-server db4.8-util libsvn-ruby1.8
Se instalarán los siguientes paquetes NUEVOS:
  libconfig-inifiles-perl libsvn-perl libsvn1 python-subversion rsync subversion subversion-tools
0 actualizados, 7 se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 4431kB/4811kB de archivos.
Se utilizarán 16,3MB de espacio de disco adicional después de esta operación.
AVISO: ¡No se han podido autenticar los siguientes paquetes!
  libsvn1 libsvn-perl subversion python-subversion subversion-tools
Des:1 http://opensource.wandisco.com lenny/svn libsvn1 1.6.11WANDisco-1 [809kB]
Des:2 http://opensource.wandisco.com lenny/svn libsvn-perl 1.6.11WANDisco-1 [1093kB]
28% [2 libsvn-perl 439262/1093kB 40%]
133kB/s 23s
lesepulveda@mv-tesi... Subversion Support - ...

```

**Ilustración 28:** Script de Shell `svninstall_debianlenny_wandisco.sh` y la actualización de librerías dependientes

El script Shell, también instala componentes complementarios para facilitar la gestión de Subversión con son las *Subversion-tools*. Esto se evidencia en la Ilustración 29.

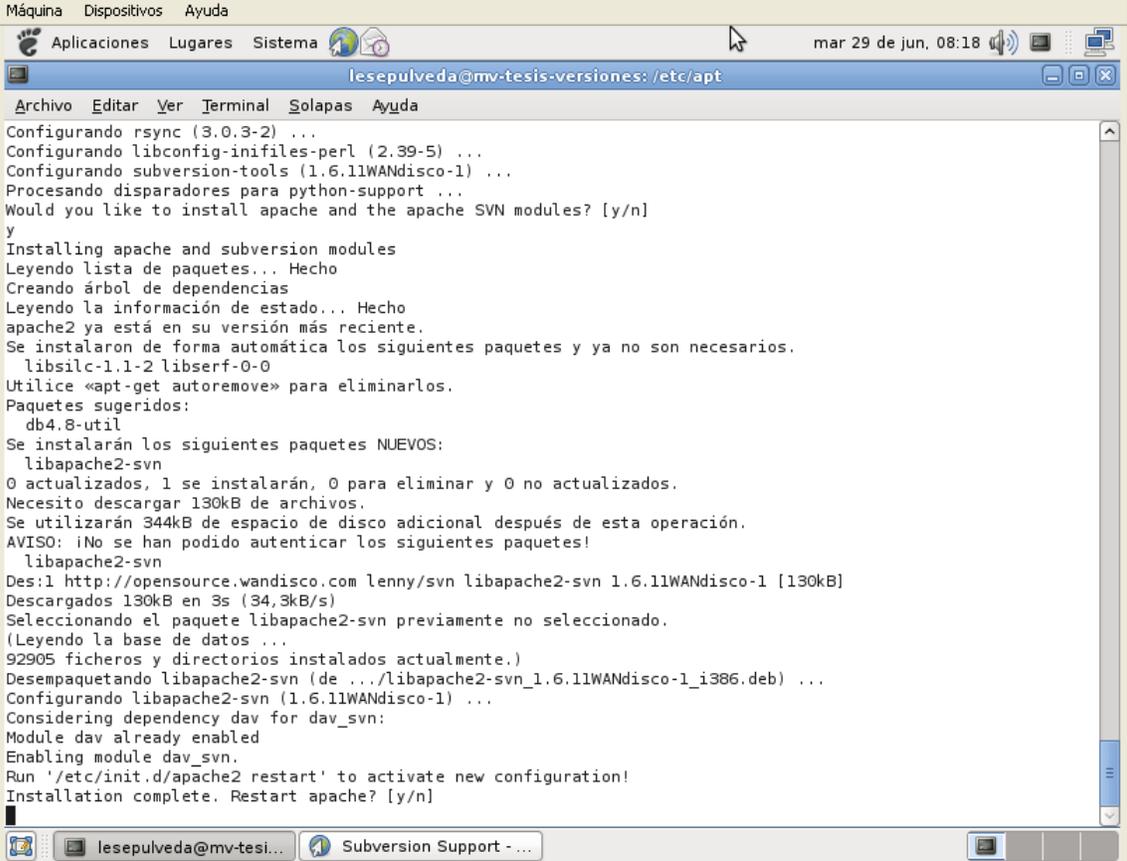


```
Máquina Dispositivos Ayuda
Aplicaciones Lugares Sistema
lesepulveda@mv-tesis-versiones: /etc/apt
Archivo Editar Ver Terminal Solapas Ayuda
Des:1 http://opensource.wandisco.com lenny/svn libsvn1 1.6.11WANDisco-1 [809kB]
Des:2 http://opensource.wandisco.com lenny/svn libsvn-perl 1.6.11WANDisco-1 [1093kB]
Des:3 http://opensource.wandisco.com lenny/svn subversion 1.6.11WANDisco-1 [1266kB]
Des:4 http://opensource.wandisco.com lenny/svn python-subversion 1.6.11WANDisco-1 [1071kB]
Des:5 http://opensource.wandisco.com lenny/svn subversion-tools 1.6.11WANDisco-1 [192kB]
Descargados 4431kB en 35s (123kB/s)
Seleccionando el paquete libsvn1 previamente no seleccionado.
(Leyendo la base de datos ...
92631 ficheros y directorios instalados actualmente.)
Desempaquetando libsvn1 (de ../libsvn1_1.6.11WANDisco-1_i386.deb) ...
Seleccionando el paquete libsvn-perl previamente no seleccionado.
Desempaquetando libsvn-perl (de ../libsvn-perl_1.6.11WANDisco-1_i386.deb) ...
Seleccionando el paquete subversion previamente no seleccionado.
Desempaquetando subversion (de ../subversion_1.6.11WANDisco-1_i386.deb) ...
Seleccionando el paquete python-subversion previamente no seleccionado.
Desempaquetando python-subversion (de ../python-subversion_1.6.11WANDisco-1_i386.deb) ...
Seleccionando el paquete rsync previamente no seleccionado.
Desempaquetando rsync (de ../r/rsync/rsync_3.0.3-2_i386.deb) ...
Seleccionando el paquete libconfig-inifiles-perl previamente no seleccionado.
Desempaquetando libconfig-inifiles-perl (de ../libconfig-inifiles-perl_2.39-5_all.deb) ...
Seleccionando el paquete subversion-tools previamente no seleccionado.
Desempaquetando subversion-tools (de ../subversion-tools_1.6.11WANDisco-1_i386.deb) ...
Procesando disparadores para man-db ...
Configurando libsvn1 (1.6.11WANDisco-1) ...
Configurando libsvn-perl (1.6.11WANDisco-1) ...
Configurando subversion (1.6.11WANDisco-1) ...
Instalando una nueva versión del fichero de configuración /etc/bash_completion.d/subversion ...
Instalando una nueva versión del fichero de configuración /etc/subversion/servers ...
Instalando una nueva versión del fichero de configuración /etc/subversion/config ...
Configurando python-subversion (1.6.11WANDisco-1) ...
Configurando rsync (3.0.3-2) ...
Configurando libconfig-inifiles-perl (2.39-5) ...
Configurando subversion-tools (1.6.11WANDisco-1) ...
Procesando disparadores para python-support ...
Would you like to install apache and the apache SVN modules? [y/n]
```

**Ilustración 29:** Instalación automatizada de herramientas complementarias a *Subversion*

Una vez realizado el reemplazo de los módulos de Apache para la interoperabilidad con Subversion, se debe proceder a realizar un reinicio del servicio Apache para dejar el sistema completamente funcional, por lo tanto debemos responder de forma afirmativa a la pregunta que se muestra en la parte inferior de la Ilustración 30.

Con el reinicio del servicio Apache se da por terminada la ejecución del script Shell `svninstall_debianlenny_wandisco.sh`.

The image shows a terminal window from a Linux desktop environment. The window title is 'lesepulveda@mv-tesis-versiones: /etc/apt'. The terminal output shows the process of installing 'libapache2-svn' using 'apt-get'. It lists several packages being configured: 'rsync (3.0.3-2)', 'libconfig-inifiles-perl (2.39-5)', and 'subversion-tools (1.6.11WANdisco-1)'. It asks if the user wants to install 'apache' and 'apache SVN modules', to which 'y' is entered. The terminal shows that 'apache2' is already up-to-date and that 'libapache2-svn' is being installed from a remote source. It also shows the installation of 'libapache2-svn' and the enabling of the 'dav' module for Apache. The process concludes with 'Installation complete. Restart apache?' and 'y/n' is entered.

```
Máquina Dispositivos Ayuda
Aplicaciones Lugares Sistema
lesepulveda@mv-tesis-versiones: /etc/apt
Archivo Editar Ver Terminal Solapas Ayuda
Configurando rsync (3.0.3-2) ...
Configurando libconfig-inifiles-perl (2.39-5) ...
Configurando subversion-tools (1.6.11WANdisco-1) ...
Procesando disparadores para python-support ...
Would you like to install apache and the apache SVN modules? [y/n]
y
Installing apache and subversion modules
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
apache2 ya está en su versión más reciente.
Se instalaron de forma automática los siguientes paquetes y ya no son necesarios.
  libsilc-1.1-2 libserf-0-0
Utilice «apt-get autoremove» para eliminarlos.
Paquetes sugeridos:
  db4.8-util
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-svn
0 actualizados, 1 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 130kB de archivos.
Se utilizarán 344kB de espacio de disco adicional después de esta operación.
AVISO: ¡No se han podido autenticar los siguientes paquetes!
  libapache2-svn
Des:1 http://opensourc.wandisco.com lenny/svn libapache2-svn 1.6.11WANdisco-1 [130kB]
Descargados 130kB en 3s (34,3kB/s)
Seleccionando el paquete libapache2-svn previamente no seleccionado.
(Leyendo la base de datos ...
92905 ficheros y directorios instalados actualmente.)
Desempaquetando libapache2-svn (de ../libapache2-svn_1.6.11WANdisco-1_i386.deb) ...
Configurando libapache2-svn (1.6.11WANdisco-1) ...
Considering dependency dav for dav_svn:
Module dav already enabled
Enabling module dav_svn.
Run '/etc/init.d/apache2 restart' to activate new configuration!
Installation complete. Restart apache? [y/n]
y
```

**Ilustración 30:** Instalación de módulos que integran Subversion y Apache Web Server

Para comprobar que efectivamente se ha realizada una actualización de la herramienta Subversion que tiene el sistema operativo Debian GNU/Linux, se procede a iniciar nuevamente el gestor de paquetes Synaptic sobre el cual se realiza nuevamente la búsqueda con la palabra clave Subversion y se debe encontrar que ahora se encuentra instalada la nueva versión del producto, la 1.6.11 tal como lo muestra la Ilustración 31.

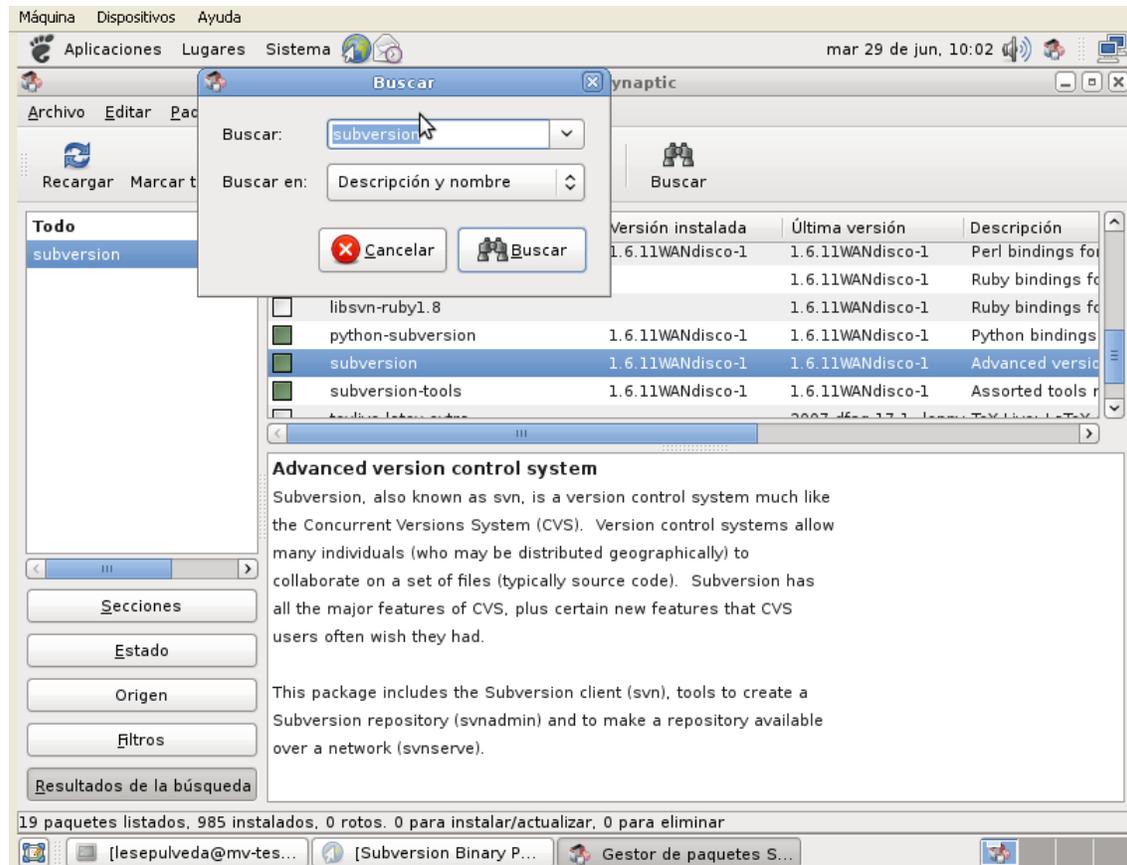
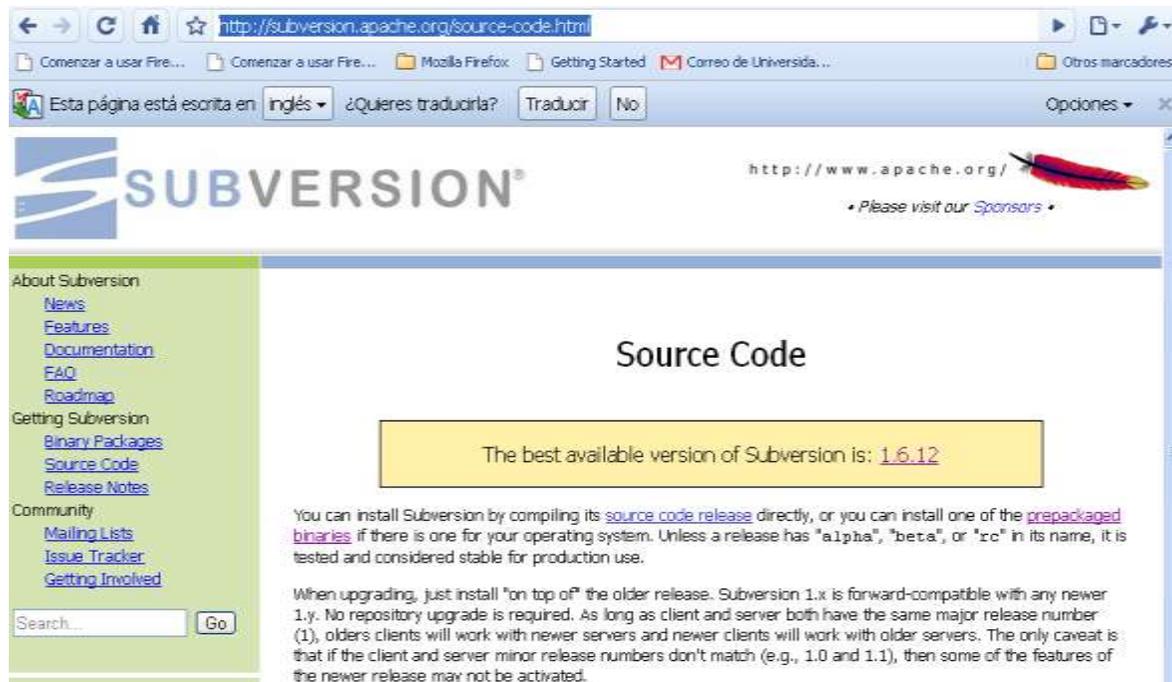


Ilustración 31: Comprobando la nueva versión instalada de *Subversion*

### 7.3.2. Instalación manual de *Subversion*

Otra forma de realizar la instalación de *Subversion* es a través de su código fuente y la posterior compilación de éste, para obtener los binarios de la herramienta con un alto grado de especialización según las necesidades puntuales de una organización y aprovechando incluso la posibilidad de utilizar versiones más recientes que las disponibles a través de los binarios tal como lo muestra la Ilustración 32.

A continuación se detalla el procedimiento de instalación y configuración manual de la herramienta *Subversion*.



**Ilustración 32:** Descargando el código fuente de *Subversion*

Antes de iniciar con la instalación del software para la gestión de versiones *Subversion*, es necesario comprender que este software está compuesto por un conjunto de herramientas que deben organizarse de una forma adecuada para que puedan operar como un conjunto funcional y significativo para los usuarios finales.

Los paquetes que a continuación se describen, deben ser instalados sobre el sistema operativo Debian GNU/Linux.

### 7.3.3. Paquetes requeridos para la instalación manual de *Subversion*

Archivo	URL
db-4.6.21.NC.tar.gz	<a href="http://www.oracle.com/technology/software/products/berkeley-db/htdocs/popup/db/4.6.21/db-">http://www.oracle.com/technology/software/products/berkeley-db/htdocs/popup/db/4.6.21/db-</a>

---

	<a href="#">nctargz.html</a>
apr-1.2.12.tar.gz	<a href="http://download.filehat.com/apache/apr/">http://download.filehat.com/apache/apr/</a>
apr-util-1.2.12.tar.gz	<a href="http://download.filehat.com/apache/apr/">http://download.filehat.com/apache/apr/</a>
httpd-2.2.8.tar.gz	<a href="http://download.filehat.com/apache/httpd/">http://download.filehat.com/apache/httpd/</a>
subversion-1.6.12.tar.gz	<a href="http://subversion.apache.org/source-code.html">http://subversion.apache.org/source-code.html</a>

#### 7.3.4. Librerías requeridas para la instalación manual de Subversion

Para llevar a cabo la instalación de los distintos módulos, se requiere tener instalados los siguientes paquetes y librerías:

- ✓ zlib1g
- ✓ zlib1g-dev
- ✓ zlib-bin
- ✓ libcurl
- ✓ libexpat1-dev
- ✓ libssl-dev
- ✓ libldap-2.3-0 y libldap-dev

#### 7.3.5. Instalar el motor de bases de datos *Berkeley*

Los datos almacenados dentro de repositorios *Subversion*, realmente se encuentran en una base de datos, más concretamente, un archivo de base de datos *Berkeley* (Collins-Sussan, Fitzpatrick, & Pilato, 2008). Los desarrolladores de *Subversion* decidieron usar una base de datos *Berkeley* por una serie de razones, como su licencia open-source, soporte de transacciones, ser de confianza, funcionamiento, simplicidad de su API, soporte de hilos, cursores, y más.

Para proceder con su instalación manualmente una vez se ha obtenido sus archivos fuente se hace lo siguiente:

Descomprimir el archivo db-4-6.21.NC.tar tal como se muestra a continuación.

```
# tar zxvf db-4.6.21.NC.tar
```

Ingresar al directorio generado en el paso anterior.

```
# cd db-4.6.21.NC
```

La configuración se debe realizar desde un directorio diferente al directorio dist; en este caso será el siguiente:

```
# db-4.6.21.NC/docs
```

Una vez ubicado sobre dicho directorio se procede a ejecutar el script configure para determinar el cumplimiento de las dependencias en cuanto a librerías y demás herramientas:

```
# ../dist/configure
```

Si se presenta algún error de dependencias, se debe asegurar de instalarlas antes de continuar con la instalación.

A continuación se debe ejecutar lo siguiente en su respectivo orden:

```
# make  
# make install
```

**Nota:** La ruta de instalación por defecto es la siguiente:

```
/usr/local/BerkeleyDB.4.6
```

Se agrega las siguientes líneas al archivo `/etc/ld.so.conf`

```
/usr/local/BerkeleyDB.4.6/lib  
include /etc/ld.so.conf.d/*.conf
```

Por último se ejecuta el comando `ldconfig` para levantar adecuadamente la configuración de la base de datos.

```
# ldconfig
```

### 7.3.6. Instalar Apache Runtime Project (APR)

El objetivo principal de APR, es proporcionar una API para que desarrolladores de software de código y puede estar seguro de que si no es predecible el comportamiento idéntico independientemente de la plataforma en la que está construido su software. APR es necesario para garantizar la interoperabilidad que se desea tener entre *Subversion* y el *Apache Web Server*.

Una vez tenemos el código fuente de las APR se procede de la siguiente forma:

Descomprimir el archivo `apr-1.2.12.tar.gz`

```
# tar xvfz apr-1.2.12.tar.gz
```

Ubicado sobre el directorio que se acaba de crear al descomprimir el archivo se realiza el llamado al chequeo de dependencias con la siguiente instrucción:

```
# ./configure
```

Una vez chequeadas las dependencias respectivas se procede a realizar la compilación de para obtener los binarios de las APR. Esto se consigue con la siguiente instrucción:

```
# make
```

Al realizar la ejecución de la instrucción *make*, esta realiza la utilización de las herramientas de compilación con *gcc* produce los binarios solicitados y ubicándolos en una ruta temporal dentro del mismo directorio donde se está ejecutando. Para realizar la ubicación adecuada en el sistema operativo de los binarios se debe realizar una acción complementaria que se podría considerar como el proceso de instalación tal como se muestra a continuación:

```
# make install
```

### 7.3.7. Instalar las APR-UTIL

Es otro conjunto de herramientas permiten la interoperabilidad entre la base de datos de *Berkeley* y *Subversion*. A continuación se detalla la forma para realizar su

En primera instancia, y una vez realizada la correspondiente adquisición del código fuente, se debe descomprimir el archivo `arp-util-1.2.12.tar.gz` tal como se muestra a continuación.

```
# tar xvfz apr-util-1.2.12.tar.gz
```

Ubicados en el directorio que se acaba de crear por la descompresión del archivo anterior, se debe realizar una configuración de los archivos antes de iniciar el proceso de compilación; para este fin se deben adicionar parámetros que van a determinar aspectos tales como la ubicación de las *apr-util* en el sistemas de archivos; al igual que la ruta donde se encuentra ubicada la base de datos de *Berkeley* instalada en procedimientos anteriores tal como se muestra a continuación:

```
# ./configure
  --with-apr=/usr/local/apr
  --with-berkeley-db=/usr/local/BerkeleyDB.4.6/
```

Al igual que en los casos anteriores, se deber realizar la compilación utilizando para ello la instrucción del sistema operativo *make*:

```
# make
```

Una vez terminado este proceso, también debemos realizar la instalación de los binarios recién construidos y para eso se utiliza la siguiente instrucción:

```
# make install
```

Una vez terminados los procedimientos anteriores con éxito, se debe realizar una configuración de ajuste manual adicionando cierta información sobre el archivo

/etc/ld.so.conf para garantizar que se creen los enlaces necesarios sobre las librerías que ahora serán compartidas garantizando el acceso a la base de datos de Berkeley por otras aplicaciones como es el caso de Subversión y Apache Web Server. El procedimiento se muestra a continuación:

Agregar la siguiente línea al archivo /etc/ld.so.conf

```
/usr/local/BerkeleyDB.4.6/lib
```

Luego se ejecuta el comando ldconfig para subir los enlaces de las librerías al caché

```
# ldconfig
```

### 7.3.8. Instalación manual de *Apache Web Server*

Como parte de las herramientas que integran la solución del *Sistema de Control de Versión*, la herramienta *Apache Web Server* es utilizada para exponer los repositorios a través de un módulo WebDAV. Este módulo comprende un conjunto de extensiones al protocolo HTTP que permite a los usuarios editar y administrar archivos de forma colaborativa en servidores web remotos. A continuación se detalla el procedimiento para la configuración manual de *Apache Web Server* a partir de su código fuente.

Como en los casos anteriores, una vez descargado el archivo el archivo que contiene el código fuente se procede a descomprimirlo tal como se muestra a continuación:

```
# tar xvfz httpd-2.2.8.tar.gz
```

Una vez realiza la descompresión del archivo se debe ubicar al interior del directorio que ha dado como resultado del proceso anterior. Una vez allí se debe ejecutar un script Shell llamado *buildconf*, el cual ha diseñado para realizar la tarea de chequear las dependencias funcionales asociadas a librerías tal como se muestra a continuación:

```
# ./buildconf
```

En caso de requerir librerías o programas, estas deben ser instaladas con antes de realizar nuevamente el script *buildconf* y así garantizar un despliegue exitoso.

Una vez realizadas las tareas correspondientes a los prerequisites funcionales se debe proceder a preparar la compilación a través del establecimiento de parámetros que indican las funcionalidades que deberá incluir el servicio *Apache*. Para realizar esta acción se deben proceder como sigue:

```
#!/configure
--enable-dav=shared
--with-gdbm=no
--enable-deflate=share
--enable-so
--with-berkeley-db=/usr/local/BerkeleyDB.4.6
--with-dbm=db4
--enable-http
```

En el caso de requerir algunas funcionalidades no previstas en el presente proyecto se debe hacer uso de la ayuda respectiva sobre el script Shell *configure* tal como se muestra a continuación

```
# ./configure --help
```

Al igual que los procesos de compilación que se ha realizado anteriormente, luego de tener luz verde en el chequeo de dependencias, se procede a construir los binarios de Apache Web Server mediante la herramienta *make* tal como sigue:

```
# make
```

De igual forma se procede a realiza la instalación de la herramienta compilada.

```
# make install
```

Esta es la última actividad para instalar Apache Web Server.

### 7.3.9. Construcción de binarios para la instalación manual de *Subversión*

Hasta este punto se han instalado a partir del código fuente correspondiente las diversas herramientas necesarias para poderse integrarse junto con *Subversion* en un verdadero *Sistema de Control de Versiones*. Ahora sólo falta realizar la compilación e instalación de *subversión*. Los detalles de la implementación se muestran a continuación.

Al igual que las demás herramientas, se debe partir de la descarga del código fuente desde el sitio Web oficial; luego de esto, se debe descomprimir este archivo correspondiente tal como se muestra a continuación.

```
# tar zxvf subversion-1.6.12.tar
```

Se realiza el chequeo correspondiente para las dependencias antes de realizar la construcción de los binarios de *Subversion* adicionando un conjunto de parámetros que son necesarios para integrar las herramientas que comprenden este ambiente.

```
# ./configure
--with-berkeley-db=/usr/local/BerkeleyDB.4.4
--with-apxs=/usr/local/apache2/bin/apxs
--with-apr=/usr/local/apr
--with-apr-util=/usr/local/apr
```

Cuando termine el proceso de chequeo de dependencias se debe proceder a la respectiva compilación así:

```
# make
```

De igual forma se procede a realiza la instalación de la herramienta compilada.

```
# make install
```

#### 7.4. Planeación del repositorio

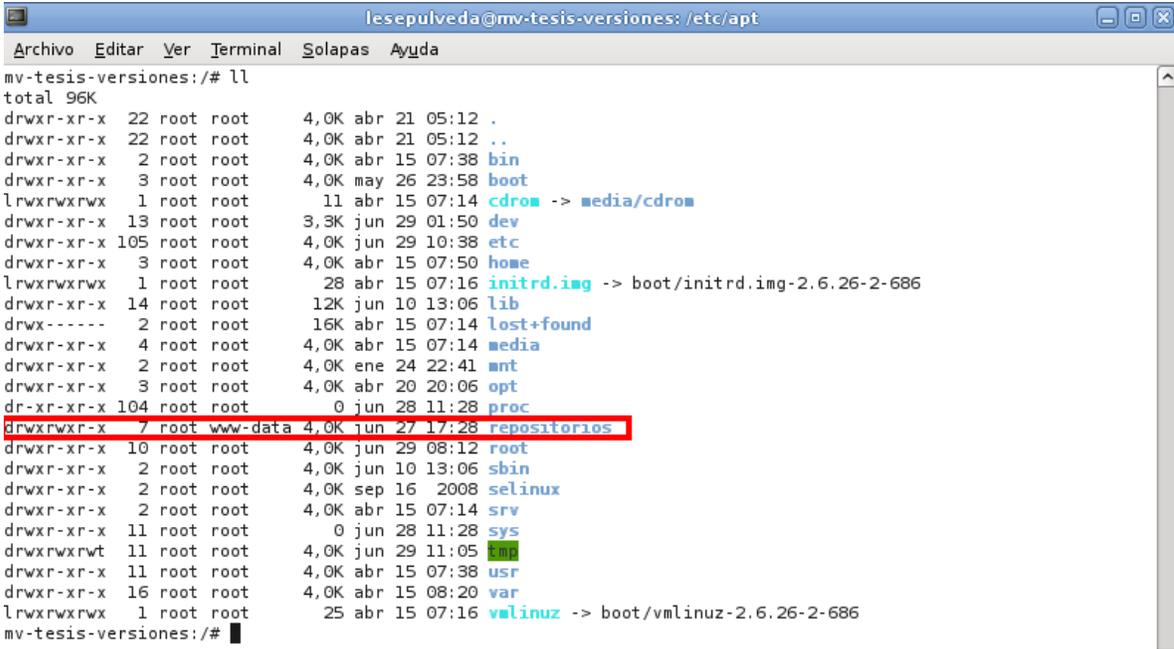
##### Relación con el Proceso Gestión de Archivos Digitales

**Actividad relacionada:** 5.6.1 Planear la gestión de archivos digitales

La planeación del un repositorio es una actividad que debe ser realizada por un grupo de personas calificadas en los aspectos metodológicos y técnicos y debe entregar reglas clara para que las personas puedan saber cómo han de interactuar con esta tecnología. A continuación se definirán tres aspectos fundamentales relacionados con la estructura que debe tener el repositorio; la definición de roles, y por último la especificación del las reglas de control de acceso.

### 7.4.1. Estructura del repositorio

Se debe definir el lugar del sistema de archivos donde estarán almacenados los repositorios de Subversion. En esta caso de ha establecido que deberá ser un directorio a partir de la raíz del sistema de archivos con el nombre “**repositorios**” Tal como se muestra en la Ilustración 33

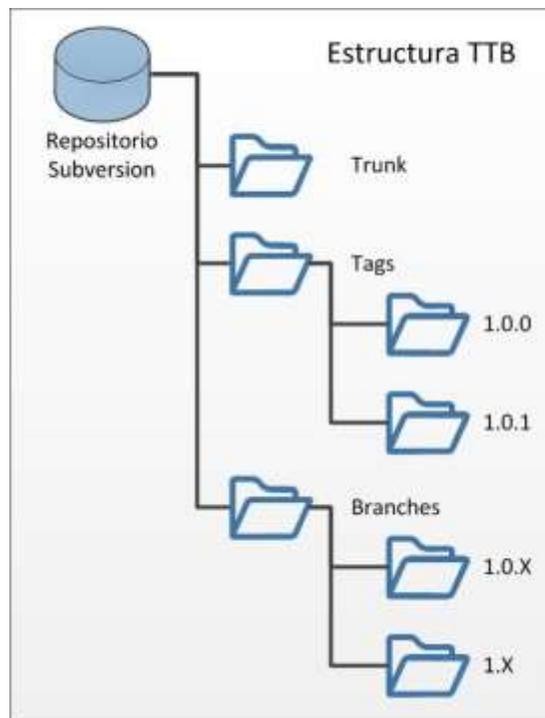


```
lesepulveda@mv-tesis-versiones: /etc/apt
Archivo Editar Ver Terminal Solapas Ayuda
mv-tesis-versiones:/# ll
total 96K
drwxr-xr-x 22 root root 4,0K abr 21 05:12 .
drwxr-xr-x 22 root root 4,0K abr 21 05:12 ..
drwxr-xr-x 2 root root 4,0K abr 15 07:38 bin
drwxr-xr-x 3 root root 4,0K may 26 23:58 boot
lrwxrwxrwx 1 root root 11 abr 15 07:14 cdrom -> media/cdrom
drwxr-xr-x 13 root root 3,3K jun 29 01:50 dev
drwxr-xr-x 105 root root 4,0K jun 29 10:38 etc
drwxr-xr-x 3 root root 4,0K abr 15 07:50 home
lrwxrwxrwx 1 root root 28 abr 15 07:16 initrd.img -> boot/initrd.img-2.6.26-2-686
drwxr-xr-x 14 root root 12K jun 10 13:06 lib
drwx----- 2 root root 16K abr 15 07:14 lost+found
drwxr-xr-x 4 root root 4,0K abr 15 07:14 media
drwxr-xr-x 2 root root 4,0K ene 24 22:41 mnt
drwxr-xr-x 3 root root 4,0K abr 20 20:06 opt
dr-xr-xr-x 104 root root 0 jun 28 11:28 proc
drwxrwxr-x 7 root www-data 4,0K jun 27 17:28 repositorios
drwxr-xr-x 10 root root 4,0K jun 29 08:12 root
drwxr-xr-x 2 root root 4,0K jun 10 13:06 sbin
drwxr-xr-x 2 root root 4,0K sep 16 2008 selinux
drwxr-xr-x 2 root root 4,0K abr 15 07:14 srv
drwxr-xr-x 11 root root 0 jun 28 11:28 sys
drwxrwxrwt 11 root root 4,0K jun 29 11:05 tmp
drwxr-xr-x 11 root root 4,0K abr 15 07:38 usr
drwxr-xr-x 16 root root 4,0K abr 15 08:20 var
lrwxrwxrwx 1 root root 25 abr 15 07:16 vmlinuz -> boot/vmlinuz-2.6.26-2-686
mv-tesis-versiones:/# █
```

Ilustración 33: Ubicación del directorio que contendrá los repositorios de información

Además del directorio donde estarán almacenados los repositorios es necesario definir la estructura que tendrá cada repositorio que se cree en la organización. Esta decisión debe estar asociada a las necesidades puntuales de cada organización.

Vale la pena echar una mirada a la estructura TTB (Trunk, Tags y Branches), la cual es una buena práctica que se realiza en el desarrollo de software. TTB se ha convertido en el estándar de facto en los repositorios de los desarrolladores. Esta estructura es la que se muestra en la Ilustración 34.



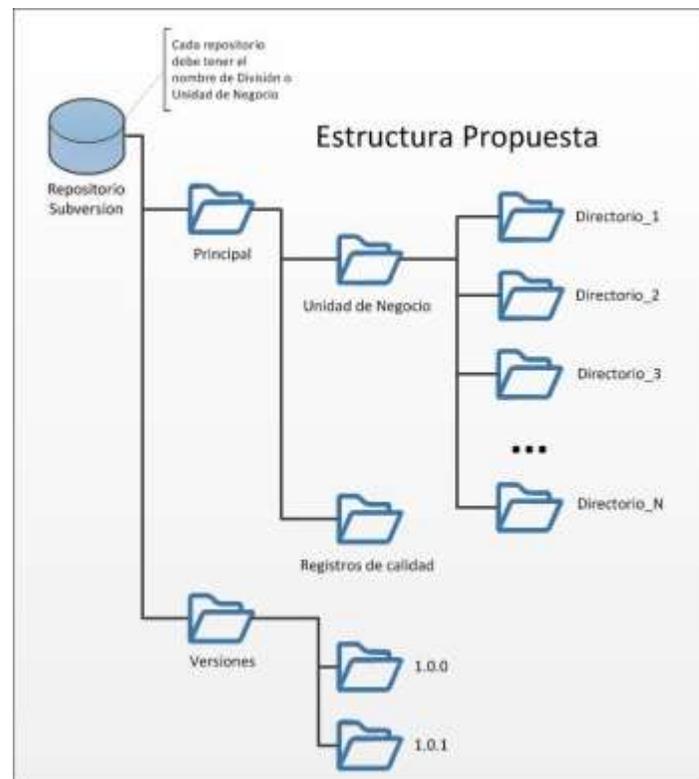
**Ilustración 34:** Estructura de repositorio TTB

En la estructura TTB el *Trunk* hace referencia al directorio principal en el cual se ubicará la información general que será utilizada constantemente por el equipo de desarrollo. El directorio *Tags* es utilizado por ciertas personas para establecer allí copias de los archivos según los hitos que se van alcanzando con el desarrollo; esta es la forma como se van etiquetando las versiones de un producto software. Por último

se tiene el directorio *Branches*, el cual es utilizado para almacenar información que debe realizar una evolución paralela a la del directorio principal.

De la estructura TTB podremos utilizar dos conceptos importantes que nutren el diseño que se está construyendo, el primer concepto es la existencia del directorio *trunk* que para éste modelo se seguirá llamando "**Principal**" y el segundo concepto es el directorio *Tags* que se llamará "**Versiones**".

A partir del directorio **Principal** se ubicaran dos directorios, el primero llamado "**Unidad de Negocio**", el cual contendrá la subdivisión respectiva según sus necesidades particulares. El segundo deberá ser nombrado "**Registro de calidad**", el cual contendrá los documentos que sean necesarios según las políticas de calidad existentes. Para facilitar la comprensión de la estructura propuesta véase la Ilustración 35.



**Ilustración 35:** Estructura de repositorio propuesta

Cada repositorio creado debe tener el nombre de la división o unidad de negocio a la que representa, adicionalmente el *Administrador de la Configuración* debe asociar un número que facilite su identificación al que llamaremos ***Id del Repositorio***.

#### 7.4.2. Definición de roles

La definición de roles consiste en especificar grupos a los cuales podrán pertenecer o no los usuarios. Estos grupos estarán relacionados con la estructura de repositorio propuesta anteriormente.

Para facilitar la identificación de los grupos se utilizará un esquema de prefijos como el que se muestra a continuación:

Los grupos que inician con el prefijo **GG** representan Grupos Globales que tiene acceso a todos los repositorios de la organización. Generalmente estos grupos deben ser utilizados por personal altamente calificad y gran confianza. Su función principal es facilitar las actividades copias de seguridad y auditorías. Ejemplo:

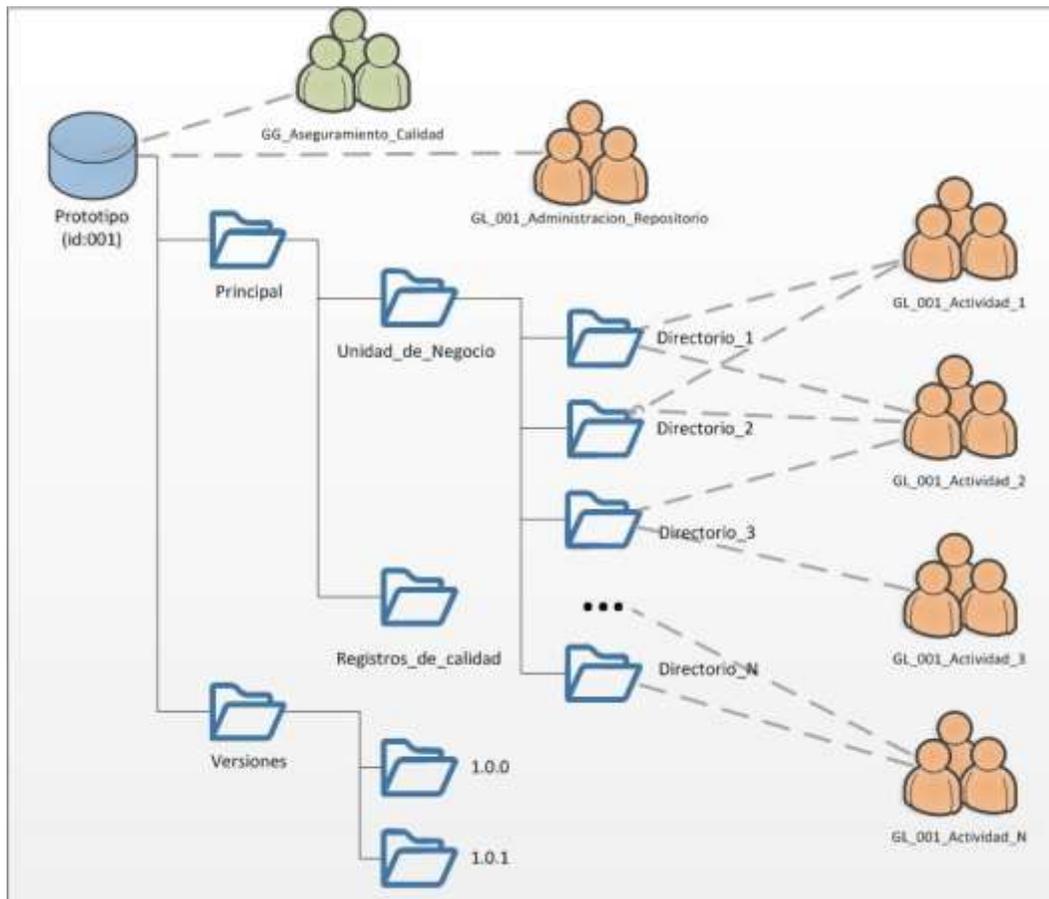
- GG\_Auditoria
- GG\_Aseguramiento\_Calidad

Los grupos de inician con el prefijo **GL** representan a Grupos Locales que tiene acceso únicamente al un repositorio específico y en ciertos directorios según sean las necesidades de seguridad de cada división o unidad de negocio. Ejemplo:

- GL\_001\_Administracion\_Repositorio
- GL\_001\_Actividad\_1
- GL\_001\_Actividad\_2
- ...

- GL\_001\_Actividad\_N
- GL\_002\_Administracion\_Repositorio
- GL\_002\_Actividad\_1
- GL\_002\_Actividad\_2
- GL\_002\_Actividad\_N

Para tener una idea más clara respecto del alcance de los grupos, en la Ilustración 36 se muestra como los grupos globales y locales pueden tener acceso a diversos niveles de la estructura del repositorio.



**Ilustración 36:** Definición de grupos sobre la estructura del repositorio

### 7.4.3. Especificación reglas de control de acceso

La especificación de reglas de control de acceso consiste en determinar si los grupos globales o locales tienen o no acceso a un repositorio específico y/o a niveles determinados según un formato como el que se muestra en el siguiente ejemplo.

ORGANIZACIÓN				
Formato: Reglas de acceso a repositorio				Fecha: AAA-MM-DD
Repositorio: <i>Nombre del repositorio</i>				
Nombre del solicitante: <i>Usuario solicitante</i>				
Nombre del responsable: <i>Usuario Responsable</i>				
No	Grupo	Repositorio/Directorio	Permisos	
			R	RW
1	GL_001_Actividad_1	Prototipo/Principal/Unidad_de_Negocio/Directorio_1		X
2	GL_001_Actividad_1	Prototipo/Principal/Unidad_de_Negocio/Directorio_2		X
3	GL_001_Actividad_2	Prototipo/Principal/Unidad_de_Negocio/Directorio_1		X
4	GL_001_Actividad_2	Prototipo/Principal/Unidad_de_Negocio/Directorio_2	X	
5	GL_001_Actividad_2	Prototipo/Principal/Unidad_de_Negocio/Directorio_3	X	
6	GL_001_Actividad_3	Prototipo/Principal/Unidad_de_Negocio/Directorio_3		X
7	GL_001_Administracion_Repositorio	Prototipo/		X
8	GG_Aseguramiento_Calidad	Prototipo/	X	

Adicionalmente debe existir otro documento donde se relacione a los usuarios y a qué grupos pertenece.

### 7.5. Creación del repositorio, usuarios y permisos

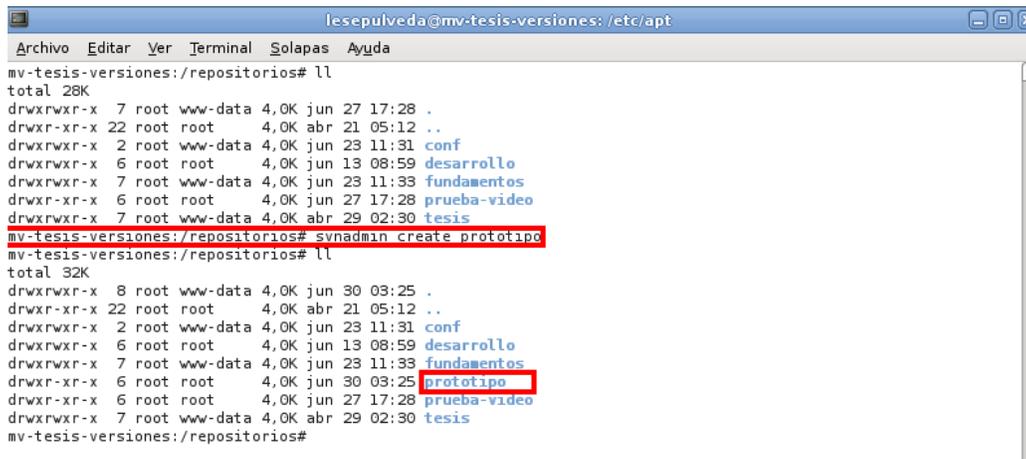
#### Relación con el Proceso Gestión de Archivos Digitales

**Actividad relacionada:** 5.6.3 Crear repositorio

Esta actividad técnica es relativamente simple cuando se tiene la información que se definió en la sección anterior. A continuación se muestra el procedimiento realizado para la creación del repositorio y la puesta en marcha de las reglas de control de acceso.

### 7.5.1. Creación del repositorio

Esta actividad consiste en la utilización del comando “*svnadmin create*” seguido del nombre del repositorio en este caso “*prototipo*”; tal como se muestra en Ilustración 37.

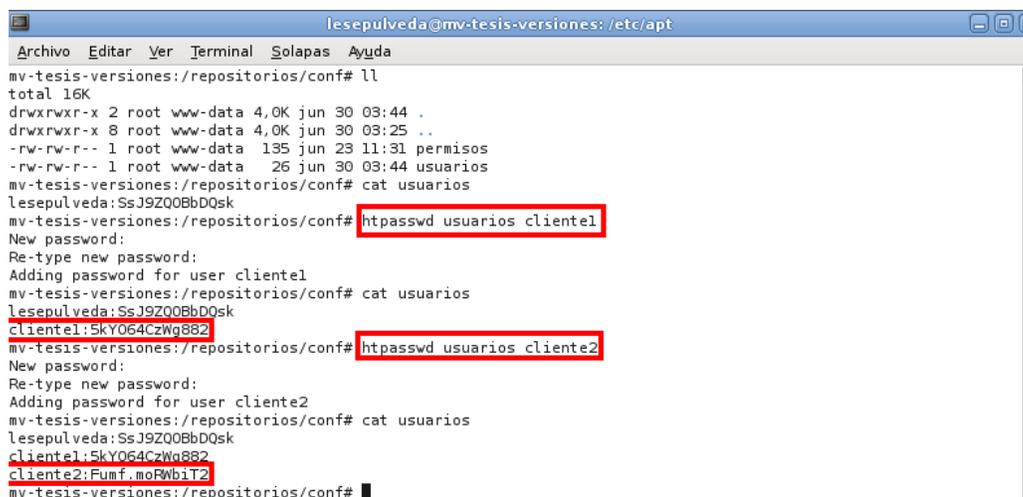


```
lesepulveda@mv-tesis-versiones: /etc/apt
Archivo Editar Ver Terminal Solapas Ayuda
mv-tesis-versiones:/repositorios# ll
total 28K
drwxrwxr-x 7 root www-data 4,0K jun 27 17:28 .
drwxr-xr-x 22 root root 4,0K abr 21 05:12 ..
drwxrwxr-x 2 root www-data 4,0K jun 23 11:31 conf
drwxrwxr-x 6 root root 4,0K jun 13 08:59 desarrollo
drwxrwxr-x 7 root www-data 4,0K jun 23 11:33 fundamentos
drwxr-xr-x 6 root root 4,0K jun 27 17:28 prueba-video
drwxrwxr-x 7 root www-data 4,0K abr 29 02:30 tesis
mv-tesis-versiones:/repositorios# svnadmin create prototipo
mv-tesis-versiones:/repositorios# ll
total 32K
drwxrwxr-x 8 root www-data 4,0K jun 30 03:25 .
drwxr-xr-x 22 root root 4,0K abr 21 05:12 ..
drwxrwxr-x 2 root www-data 4,0K jun 23 11:31 conf
drwxrwxr-x 6 root root 4,0K jun 13 08:59 desarrollo
drwxrwxr-x 7 root www-data 4,0K jun 23 11:33 fundamentos
drwxr-xr-x 6 root root 4,0K jun 30 03:25 prototipo
drwxr-xr-x 6 root root 4,0K jun 27 17:28 prueba-video
drwxrwxr-x 7 root www-data 4,0K abr 29 02:30 tesis
mv-tesis-versiones:/repositorios#
```

Ilustración 37: Creación de un repositorio con *Subversion*

### 7.5.2. Creación de usuarios

Para realizar el proceso de creación de usuario se debe simplemente adicionar una línea con la información en un archivo de texto (/repositorios/conf/usuarios). La información debe contener el nombre de usuario (login), seguido de un separador (:) y a continuación la clave de usuario encriptada. Para realizar esta acción se pueden utilizar diversas estrategias, siendo la más utilizada el llamado al comando “*htpasswd*” el cual es una herramienta de Apache Web Server que entrega la información tal y como se requiere. Ver Ilustración 38.

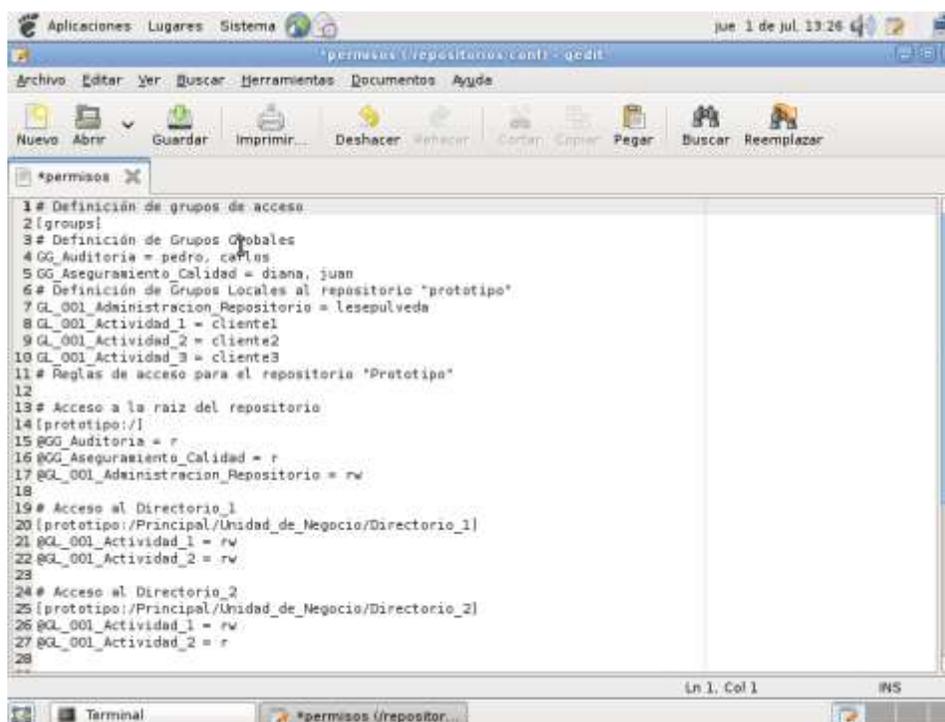


```
lesepulveda@mv-tesis-versiones: /etc/apt
Archivo Editar Ver Terminal Solapas Ayuda
mv-tesis-versiones:/repositorios/conf# ll
total 16K
drwxrwxr-x 2 root www-data 4,0K jun 30 03:44 .
drwxrwxr-x 8 root www-data 4,0K jun 30 03:25 ..
-rw-rw-r-- 1 root www-data 135 jun 23 11:31 permisos
-rw-rw-r-- 1 root www-data 26 jun 30 03:44 usuarios
mv-tesis-versiones:/repositorios/conf# cat usuarios
lesepulveda:SsJ9Z00BbD0sk
mv-tesis-versiones:/repositorios/conf# htpasswd usuarios cliente1
New password:
Re-type new password:
Adding password for user cliente1
mv-tesis-versiones:/repositorios/conf# cat usuarios
lesepulveda:SsJ9Z00BbD0sk
cliente1:5kY064CzWq882
mv-tesis-versiones:/repositorios/conf# htpasswd usuarios cliente2
New password:
Re-type new password:
Adding password for user cliente2
mv-tesis-versiones:/repositorios/conf# cat usuarios
lesepulveda:SsJ9Z00BbD0sk
cliente1:5kY064CzWq882
cliente2:Fumf_m0RwbiT2
mv-tesis-versiones:/repositorios/conf# █
```

**Ilustración 38:** Creación de usuarios para la interacción con el Sistema de Control de Versiones

### 7.5.3. Asignación de permisos

Para realizar la asignación de permisos se debe editar el archivos de texto */repositorios/conf/permisos*. Allí debe configurarse las reglas de control de acceso que se definieron con anterioridad. Ver Ilustración 39.



```
1# Definición de grupos de acceso.
2 [groups]
3# Definición de Grupos Globales
4 GG_Auditoria = pedro, carlas
5 GG_Aseguramiento_Calidad = diana, juan
6# Definición de Grupos Locales al repositorio "prototipo"
7 GL_001_Administración_Repositorio = lesepulveda
8 GL_001_Actividad_1 = cliente1
9 GL_001_Actividad_2 = cliente2
10 GL_001_Actividad_3 = cliente3
11# Reglas de acceso para el repositorio "Prototipo"
12
13# Acceso a la raíz del repositorio
14 [prototipo:/]
15 @GG_Auditoria = r
16 @GG_Aseguramiento_Calidad = r
17 @GL_001_Administración_Repositorio = rw
18
19# Acceso al Directorio_1
20 [prototipo:/Principal/Unidad_de_Negocio/Directorio_1]
21 @GL_001_Actividad_1 = rw
22 @GL_001_Actividad_2 = rw
23
24# Acceso al Directorio_2
25 [prototipo:/Principal/Unidad_de_Negocio/Directorio_2]
26 @GL_001_Actividad_1 = rw
27 @GL_001_Actividad_2 = r
28
```

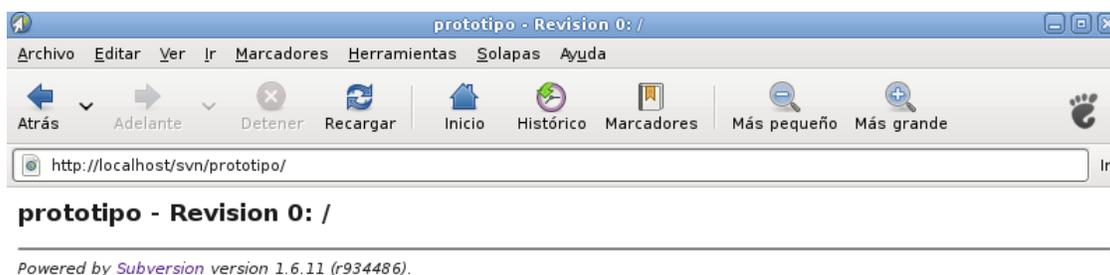
**Ilustración 39:** Asignación de reglas de control de acceso al repositorio "prototipo"

## 7.6. Poblado inicial del repositorio

### Relación con el Proceso Gestión de Archivos Digitales

**Actividad relacionada:** 5.6.4 Poblar repositorio

En primera instancia, es prudente verificar que efectivamente el repositorio “*prototipo*” se encuentra en la revisión cero y totalmente vacío. Para realizar esta verificación se utiliza un navegador de páginas Web y se localiza una dirección a la cual queda asociada la publicación del repositorio tal como lo muestra la Ilustración 40.

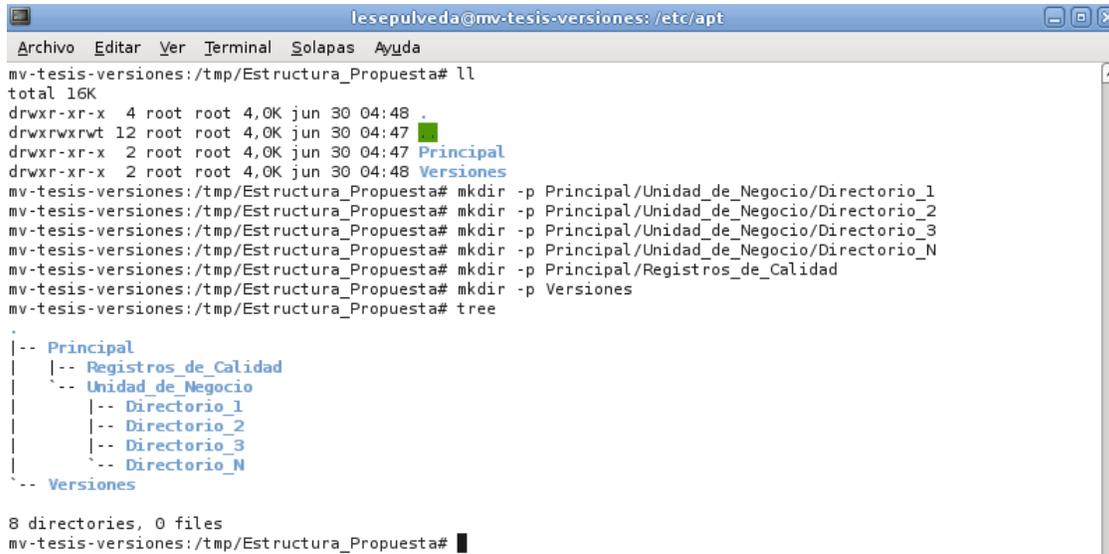


**Ilustración 40:** Verificación de la publicación del repositorio “prototipo”

En la mayoría de las veces, la actividad de poblar inicialmente un repositorio, consiste en la ubicación de la estructura de repositorio definida en secciones anteriores al interior del repositorio mismo. Para realizar el poblado inicial deben seguirse algunos pasos como se muestra a continuación:

**Paso 1:**

Se debe realizar la construcción de la estructura propuesta para el repositorio de información en un directorio temporal, tal como se muestra en la Ilustración 41.



```
lesepulveda@mv-tesis-versiones: /etc/apt
Archivo Editar Ver Terminal Solapas Ayuda
mv-tesis-versiones:/tmp/Estructura_Propuesta# ll
total 16K
drwxr-xr-x  4 root root 4,0K jun 30 04:48 .
drwxrwxrwt 12 root root 4,0K jun 30 04:47 ..
drwxr-xr-x  2 root root 4,0K jun 30 04:47 Principal
drwxr-xr-x  2 root root 4,0K jun 30 04:48 Versiones
mv-tesis-versiones:/tmp/Estructura_Propuesta# mkdir -p Principal/Unidad_de_Negocio/Directorio_1
mv-tesis-versiones:/tmp/Estructura_Propuesta# mkdir -p Principal/Unidad_de_Negocio/Directorio_2
mv-tesis-versiones:/tmp/Estructura_Propuesta# mkdir -p Principal/Unidad_de_Negocio/Directorio_3
mv-tesis-versiones:/tmp/Estructura_Propuesta# mkdir -p Principal/Unidad_de_Negocio/Directorio_N
mv-tesis-versiones:/tmp/Estructura_Propuesta# mkdir -p Principal/Registros_de_Calidad
mv-tesis-versiones:/tmp/Estructura_Propuesta# mkdir -p Versiones
mv-tesis-versiones:/tmp/Estructura_Propuesta# tree
.
|-- Principal
|   |-- Registros_de_Calidad
|   |-- Unidad_de_Negocio
|       |-- Directorio_1
|       |-- Directorio_2
|       |-- Directorio_3
|       |-- Directorio_N
|-- Versiones
8 directories, 0 files
mv-tesis-versiones:/tmp/Estructura_Propuesta#
```

**Ilustración 41:** Creación de la estructura del repositorio “prototipo”

**Paso 2:**

Se procede a copiar el árbol de directorios sin versionar al interior del repositorio, esta acción generará la revisión 1. Ver Ilustración 42.



```
lesepulveda@mv-tesis-versiones: /etc/apt
Archivo Editar Ver Terminal Solapas Ayuda
mv-tesis-versiones:/tmp/Estructura_Propuesta# svn import . file:///repositorios/prototipo -m "Poblado inicial d
el repositorio"
Añadiendo      Principal
Añadiendo      Principal/Unidad_de_Negocio
Añadiendo      Principal/Unidad_de_Negocio/Directorio_N
Añadiendo      Principal/Unidad_de_Negocio/Directorio_1
Añadiendo      Principal/Unidad_de_Negocio/Directorio_2
Añadiendo      Principal/Unidad_de_Negocio/Directorio_3
Añadiendo      Principal/Registros_de_Calidad
Añadiendo      Versiones

Commit de la revisión 1.
mv-tesis-versiones:/tmp/Estructura_Propuesta#
```

**Ilustración 42:** Poblado inicial del repositorio “prototipo”

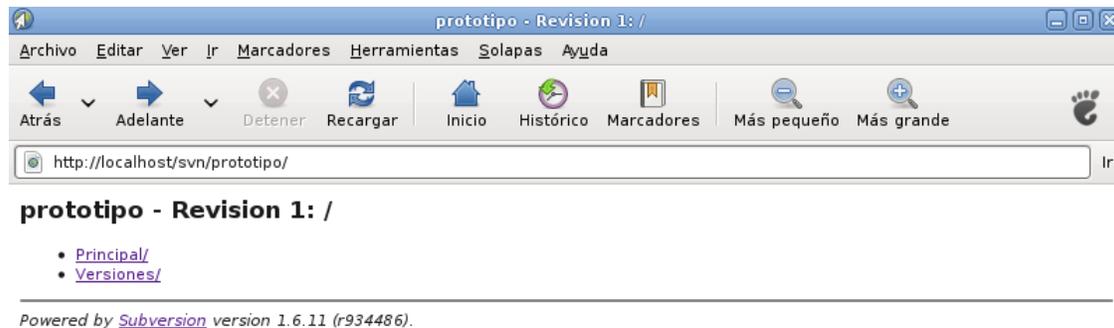
El comando utilizado es *“svn import”* a esta instrucción se deben adicionar pasarse tres parámetros, los cuales representan lo siguiente:

- Ruta del directorio donde construyo la estructura de directorios y si es el caso con su respectiva información de base que va a ser copiada al repositorio. En la Ilustración 42, este parámetro es representado por un punto (.); este punto representa una ruta relativa y quiere decir que el directorio que contiene la información que se desea importar al repositorio está justo en el directorio actual de trabajo.
- El parámetro siguiente corresponde a la ruta donde se encuentra ubicado el repositorio al que se le desea adicionar la información. En la Ilustración 42 este parámetro es representado por una ruta con el siguiente formato *“file:///repositorios/prototipo”*; *file:///* hace referencia a una URL con ubicación solamente local, otra forma que sería igualmente válido es *“file://localhost/”*; el resto de la información (repositorios/prototipo), representa la ruta en el servidor donde se encuentra efectivamente el repositorio destino.
- El tercer parámetro inicia con *-m* seguido por una información entre comillas dobles “mensaje”. En la Ilustración 42 este mensaje corresponde a “Poblado inicial del repositorio”. Este parámetro es opcional, pero como una buena práctica debe utilizarse siempre para permitir comprender el por qué de los cambios realizados a la información cuando se esté haciendo análisis de trazabilidad de la información.

### Paso 3:

Es paso consiste en verificar vía Web que efectivamente se ha logrado realizar el poblado del repositorio con éxito. Para realizar la verificación basta con

consultar nuevamente el sitio Web correspondiente al repositorio tal como se muestra en la Ilustración 43.



**Ilustración 43:** Verificación del poblado inicial

## 7.7. Instalación de herramientas en las estaciones de trabajo cliente

### Relación con el Proceso Gestión de Archivos Digitales

**Actividad relacionada:** 5.6.5 Preparar Estación de Trabajo Cliente

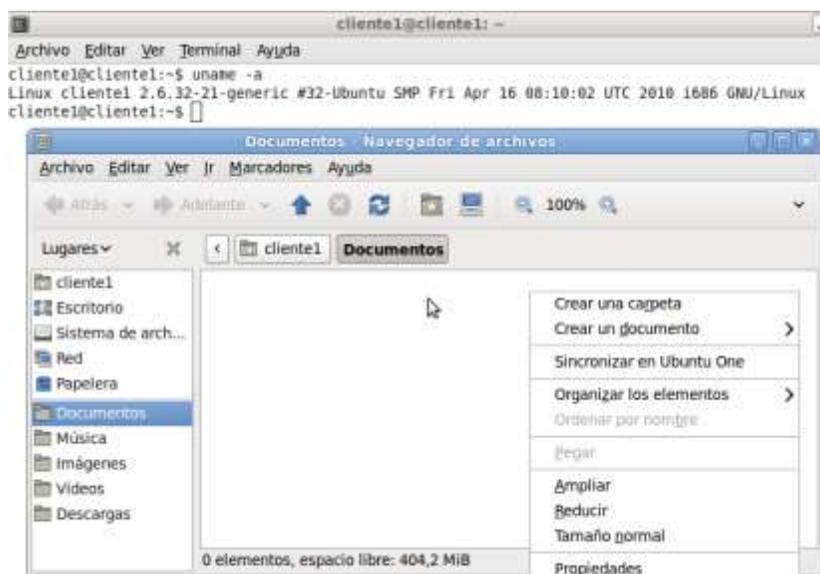
Las actividades para la instalación de la herramienta cliente son relativamente sencillas pero tiene sus obvias diferencias de un sistema operativo a otro. Para ilustrar esto, a continuación se realizará el proceso de instalación de herramientas cliente sobre los sistemas operativos Ubuntu GNU/Linux 10.4 y MS-Windows XP respectivamente.

### 7.7.1. Configuración del cliente con sistema operativo Ubuntu GNU/Linux 10.4

Existen diversas herramientas que sirven para realizar las tareas de un cliente Subversión, pero se deben considerar aquellas que presenten un alto nivel de usabilidad, siendo muy eficientes, aquellas herramientas que se integran con las funciones de los gestores de archivos. Cumpliendo esta condición encontramos a la herramienta llamada **RabbitVCS**.

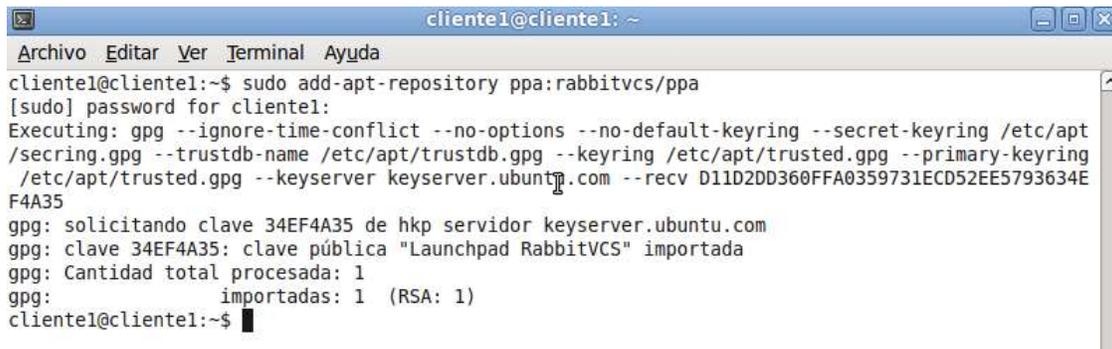
*RabbitVCS* es código abierto, además está compuesto por un conjunto de herramientas gráficas para proporcionar acceso sencillo y directo a los Sistemas de Control de Versiones que utiliza. En la actualidad, está integrado en el gestor de archivos *Nautilus* y sólo es compatible con *Subversion*, pero el objetivo de *RabbitVCS* es incorporarse con otros sistemas de control de versiones, así como otros administradores de archivos (Rabbitvcs, 2009).

Una primera actividad es verificar la respuesta de Nautilus al realizar un clic secundario en un directorio cualquiera; tal como lo muestra la Ilustración 44, sólo se tiene las opciones convencionales de Nautilus.



**Ilustración 44:** Opciones clásicas de Nautilus ante un clic secundario en un directorio cualquiera

A continuación procedemos a realizar la adición de la ruta en el gestor de paquetes del sistema operativo para posteriormente poder realizar la instalación de forma automática. Ver Ilustración 45.



```
cliente1@cliente1: ~
Archivo Editar Ver Terminal Ayuda
cliente1@cliente1:~$ sudo add-apt-repository ppa:rabbitvcs/ppa
[sudo] password for cliente1:
Executing: gpg --ignore-time-conflict --no-options --no-default-keyring --secret-keyring /etc/apt
/secring.gpg --trustdb-name /etc/apt/trustdb.gpg --keyring /etc/apt/trusted.gpg --primary-keyring
/etc/apt/trusted.gpg --keyserver keyserver.ubuntu.com --recv D11D2DD360FFA0359731ECD52EE5793634E
F4A35
gpg: solicitando clave 34EF4A35 de hkp servidor keyserver.ubuntu.com
gpg: clave 34EF4A35: clave pública "Launchpad RabbitVCS" importada
gpg: Cantidad total procesada: 1
gpg:          importadas: 1 (RSA: 1)
cliente1@cliente1:~$
```

**Ilustración 45:** Incluyendo rutas para descargar RabbitVCS en el gestor de paquetes

Para poder disponer de la información del que se acaba de adicionar, es necesario actualiza la lista del software que tiene el gestor de paquetes. Para esta acción se utiliza el comando siguiente:

```
$ sudo apt-get update
```

Una vez realizado el proceso de actualización del software disponible, se procede a realizar la instalación automática de *RabbitCSV*. Ver Ilustración 46.

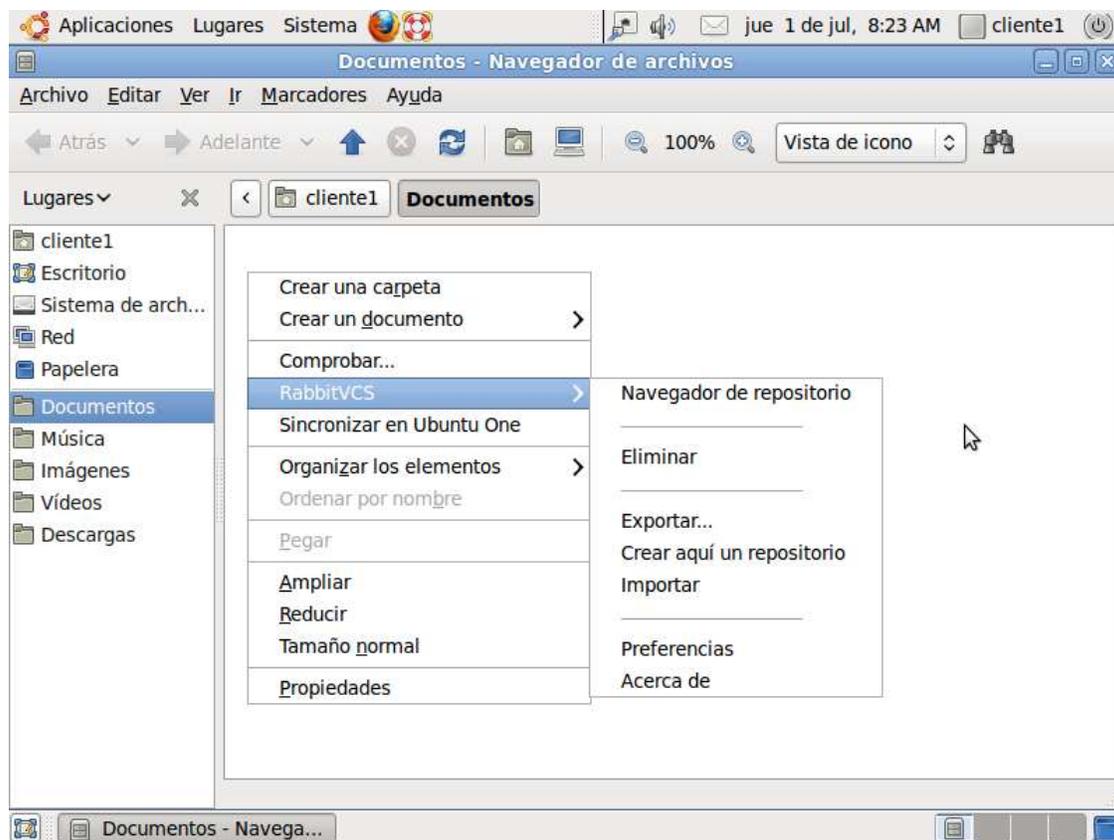


```
cliente1@cliente1:~$ sudo apt-get install rabbitvcs*
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Nota, seleccionando rabbitvcs-core para la expresión regular 'rabbitvcs*'
Nota, seleccionando rabbitvcs-nautilus para la expresión regular 'rabbitvcs*'
Nota, seleccionando rabbitvcs-cli para la expresión regular 'rabbitvcs*'
Nota, seleccionando rabbitvcs-thunar para la expresión regular 'rabbitvcs*'
Nota, seleccionando rabbitvcs-gedit para la expresión regular 'rabbitvcs*'
Nota, seleccionando rabbitvcs para la expresión regular 'rabbitvcs*'
```

**Ilustración 46:** Instalación de RabbitVCS a través de apt-get en Ubuntu 10.4

Una vez realizada la instalación del RabbitVCS, es necesario realizar un cierre de sesión para garantizar que la integración con el sistema de gestión de archivos Nautilus se realiza adecuadamente.

Una vez cargada la sesión, se realiza nuevamente la comprobación de las opciones desplegadas al realizar clic secundario encontrando ahora que existe toda una sección dedicada a RabbitVCS. Ver Ilustración 47.



**Ilustración 47:** Opciones de RabbitVCS integradas con Nautilus

### 7.7.2. Configuración del cliente con sistema operativo MS-Windows XP

Para el sistema operativo MS Windows XP también existe una herramienta que se integra adecuadamente con el gestor de archivos del sistema operativo. Se trata de TortoiseSVN, una herramienta software libre que se integra adecuadamente con el explorador de Windows para entregar alta usabilidad a los usuarios de repositorios tipo Subversion.

La herramienta se encuentra disponible en el sitio Web siguiente:

- <http://tortoisesvn.net/downloads>

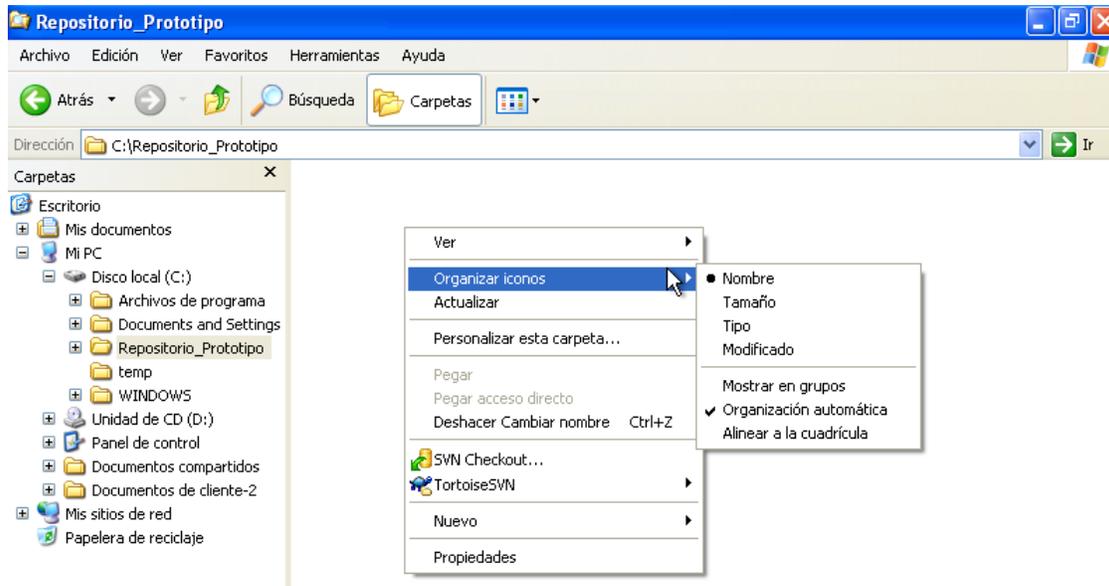
Posteriormente a la descarga, se realiza el proceso de instalación el cual consiste únicamente el ejecutar el archivo descargado y seguir los pasos que propone el asistente. Ver Ilustración 48.



**Ilustración 48:** Inicio de la instalación de TortoiseSVN sobre MS Windows XP

Una vez se ha terminado el proceso de instalación, se debe reiniciar el sistema operativo para garantizar la integración de la herramienta *TortoiseSVN* y el gestor de

archivos del sistema operativo. Al término del reinicio se comprueba que la integración sea exitosa; para esto, simplemente se debe realizar clic secundario en cualquier directorio del sistema utilizando el explorador de Windows y verificando que existan opciones asociadas a *TortoiseSVN* tal como se muestra en la Ilustración 49.



**Ilustración 49:** Herramienta TortoiseSVN integrada con el sistema de archivos de Windows

También se encuentran disponibles complementos para diversos lenguajes. Ver



**Ilustración 50:** Paquete de lenguaje español de TortoiseSVN

## 7.8. Sincronizar archivos digitales con el repositorio

### Relación con el Proceso Gestión de Archivos Digitales

**Actividad relacionada:** 5.6.7 Sincronizar archivos digitales

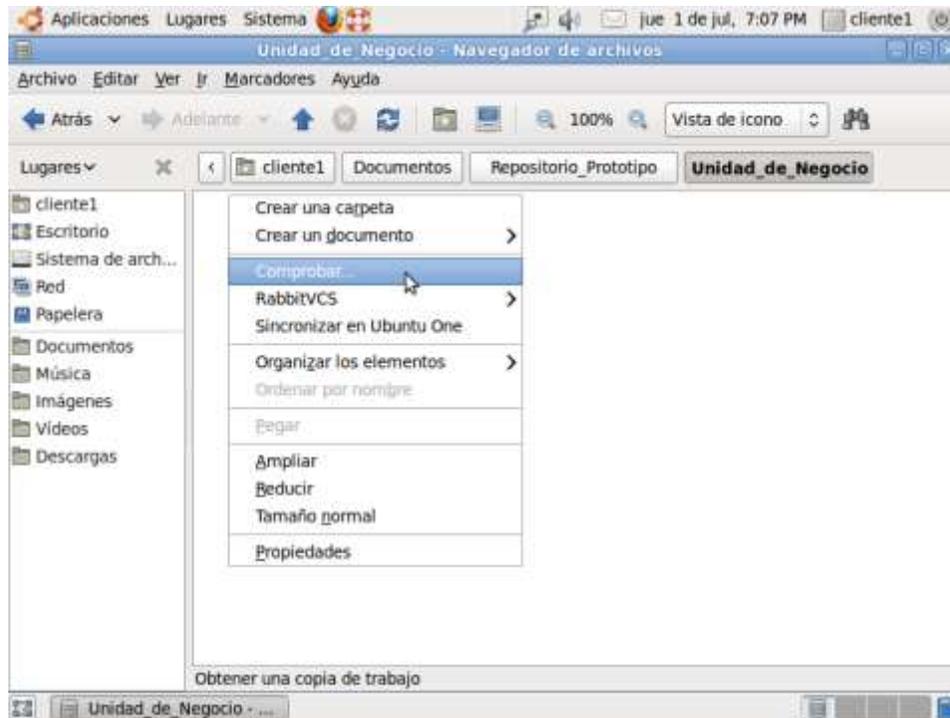
La sincronización de archivos digitales con el repositorio central necesita en primera instancia realizar la obtención de una copia de la sección del repositorio a la se tiene acceso. Una vez se tiene la copia en la estación de trabajo local, se podrá manipular la información y si se tiene permiso de escritura, se podrá realizar la confirmación de los cambios hacia el repositorio. Estas actividades se detallan a continuación.

#### 7.8.1. Obtener una copia de trabajo

Para obtener una copia de trabajo de un repositorio necesitamos tener los siguientes cuatro elementos:

Herramienta cliente para interactuar con el repositorio, podrá ser RabbitVCS o TortoiseSVN según sea el caso. La URL del repositorio. En el caso del ejemplo se trata de la siguiente: <http://192.168.100.1/svn/prototipo>. Además se debe tener un usuario válido para el sistema de control de versiones con su respectiva clave.

Para realizar el proceso de obtener una copia de trabajo desde el repositorio, es necesario primero crear un directorio en el sistema local. Para la presente simulación el directorio se llamará "Repositorio\_Prototipo/Unidad\_de\_Negocio". Ver Ilustración 51.



**Ilustración 51:** Directorio de trabajo local para contener la copia de trabajo del repositorio

La opción comprobar de RabbitVCS permite establecer comunicación con el servidor y expone una interfaz donde se debe colocar la URL del repositorio con la ruta completa a la que se tiene acceso. Ver Ilustración 52.



**Ilustración 52:** Conexión al repositorio para obtener la copia de trabajo

Una vez se ha especificado al URL y la ruta local donde se tendrá la copia de trabajo, se debe realizar la correspondiente autenticación. Ver Ilustración 53.

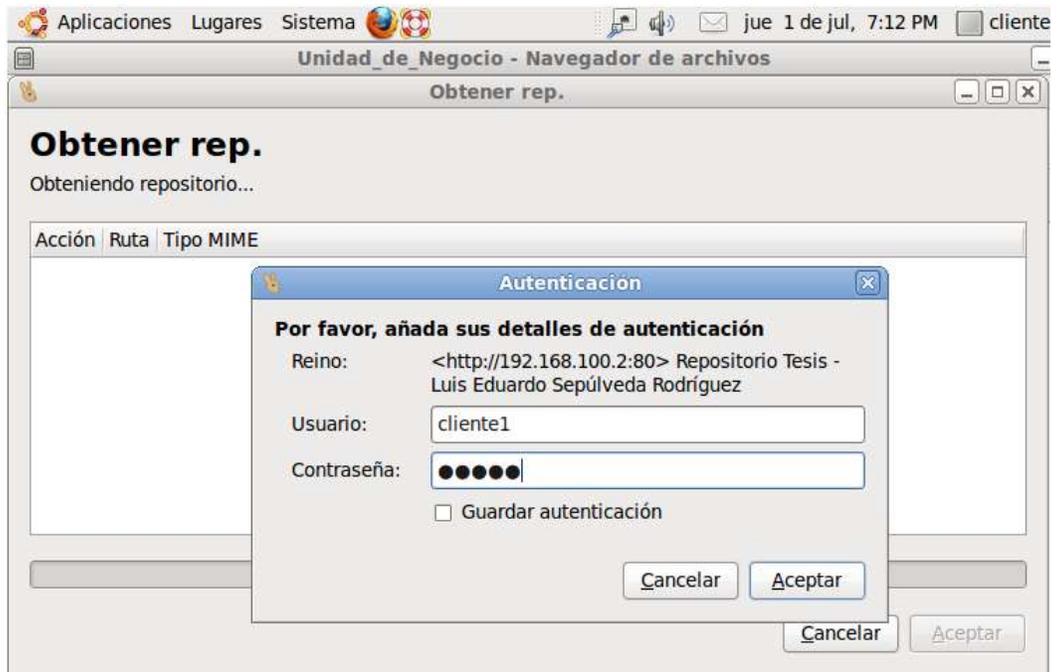


Ilustración 53: Autenticación para obtener una copia de trabajo

Al terminar el proceso se debe verificar la respuesta por parte del servidor aceptando o negando la entrega de archivos según sea el caso. Ver Ilustración 55.

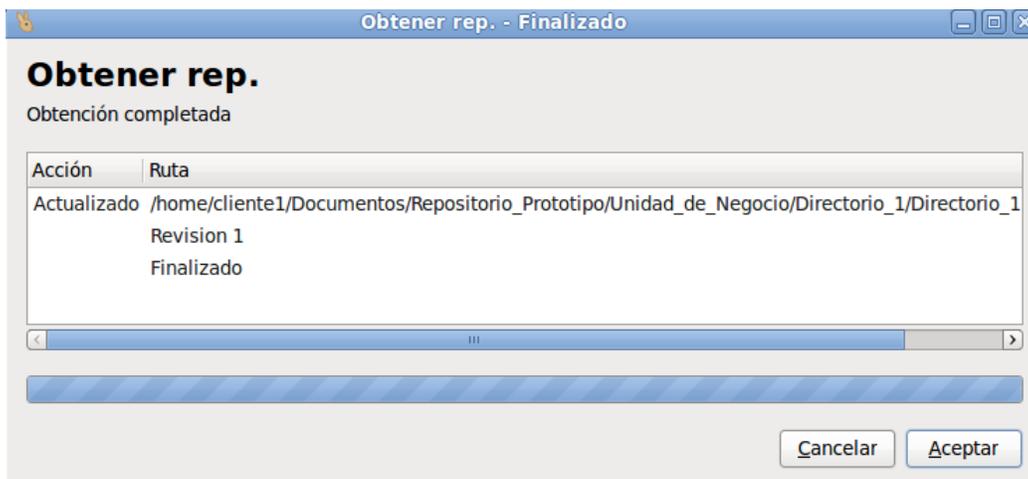


Ilustración 54: Copia de trabajo obtenida exitosamente

Finalmente se puede observar en el gestor de archivos que el directorio correspondiente a una copia de trabajo tiene un icono de adjunto el cual indica según su color, el estado de la copia de trabajo en comparación con el repositorio; verde si está sincronizado y rojo si necesita sincronización. Ver Ilustración 55.

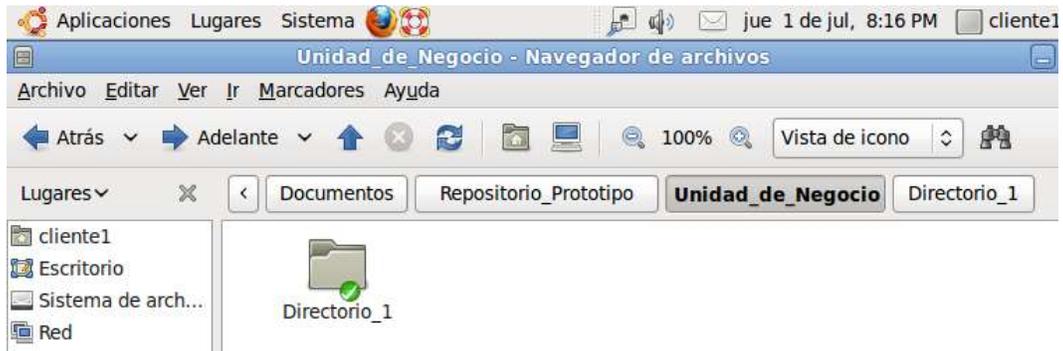


Ilustración 55: Información obtenida del repositorio “prototipo”

### 7.8.2. Confirmar cambios hacia el repositorio

Para realizar la aceptación de los cambios debemos utilizar la opción “*Confirmar*” que se expone al momento de dar clic secundario sobre el archivo o directorio que se desea conformar tal como se muestra en la Ilustración 56.



Ilustración 56: Confirmando cambios de los archivos en la copia de trabajo.

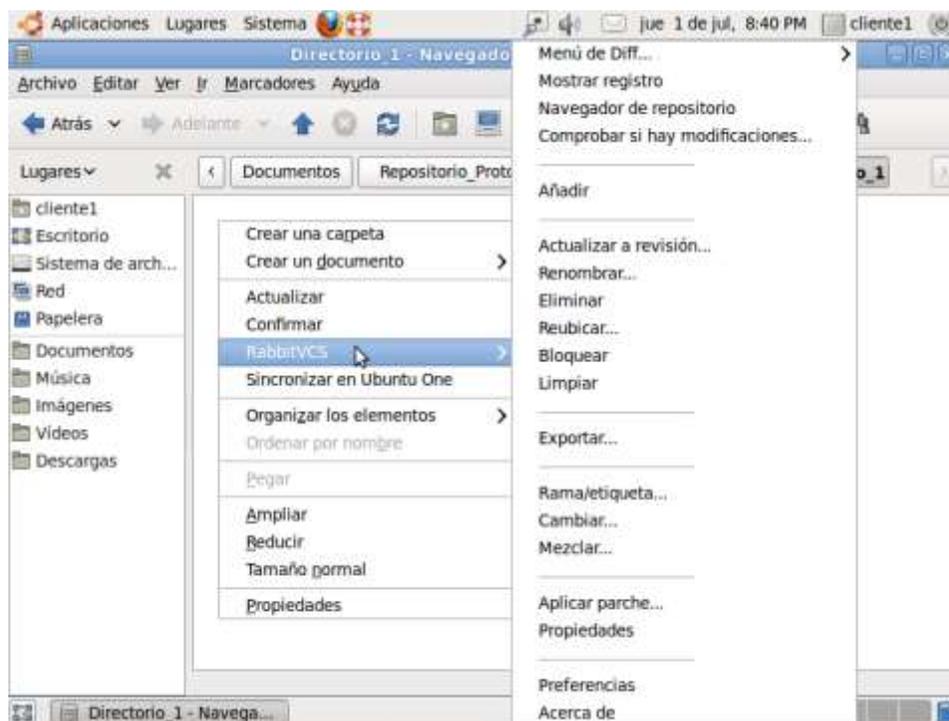
## 7.9. Interacción de los clientes con el repositorio central

### Relación con el Proceso Gestión de Archivos Digitales

**Actividad relacionada:** 5.6.6 Interactuar con archivos digitales

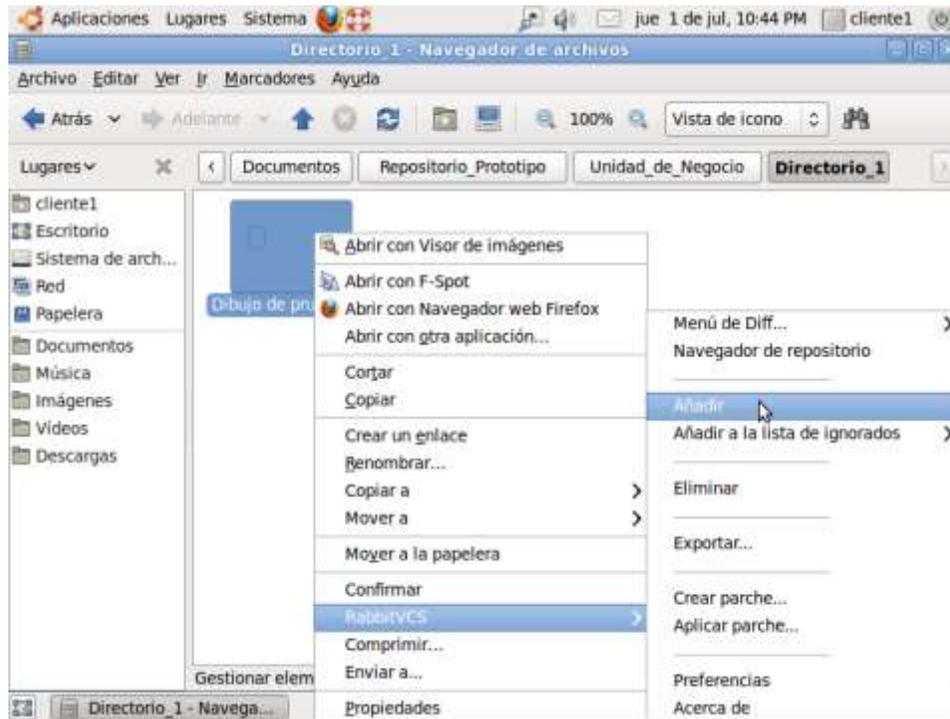
Para realizar las tareas relacionadas con la creación, modificación, eliminación y renombrado de archivos digitales que pertenecen a una *copia de trabajo*, deben realizarse de la forma convencional que provee cada sistema operativo y el único aspecto a considerar es utilizar estos procedimientos a través de las opciones que disponen las herramientas cliente ya se RubbitVCS o TortoiseSVN.

### 7.9.1. Interacción a través de la herramienta cliente RubbitVCS



**Ilustración 57:** Opciones de RubbitVCS para la gestión de Archivos Digitales.

Luego de que los usuario han realizados interacciones sobre archivos digitales como por ejemplo la adición de un nuevo archivo a la copia de trabajo, deberán adicionar estor a través de las opciones que provee la herramienta tal como se muestra en la Ilustración 58.



**Ilustración 58:** Adicionar archivos a la copia de trabajo

Posteriormente se debe *Confirmar* el cambio realizado una sincronización con el repositorio; esta acción generará una nueva revisión como se muestra en la Ilustración 60 y la Ilustración 60.

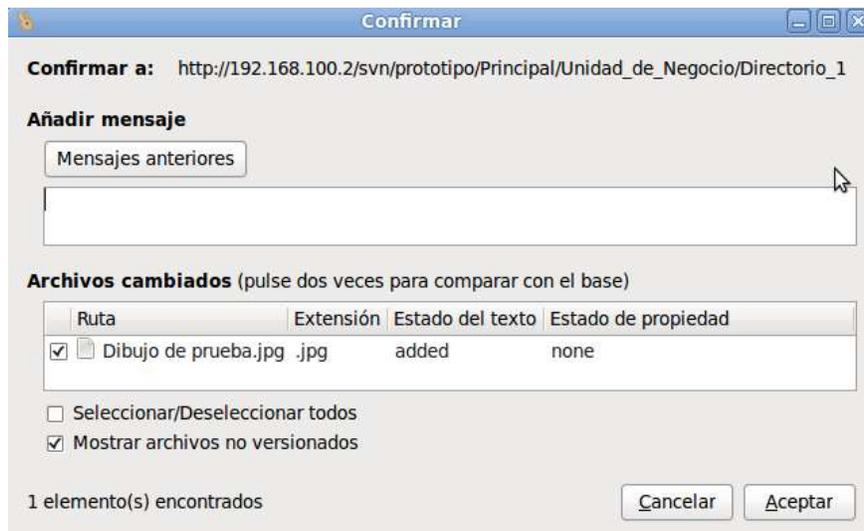


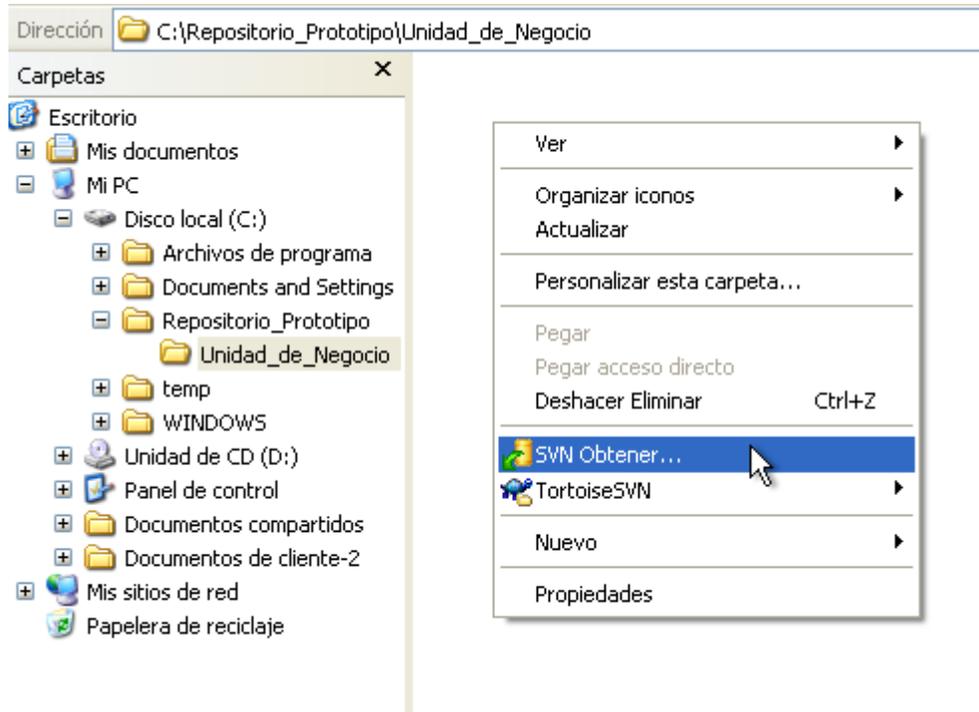
Ilustración 59: GUI para confirmar cambios hacia el repositorio



Ilustración 60: Resultado de una confirmación de cambios hacia el repositorio

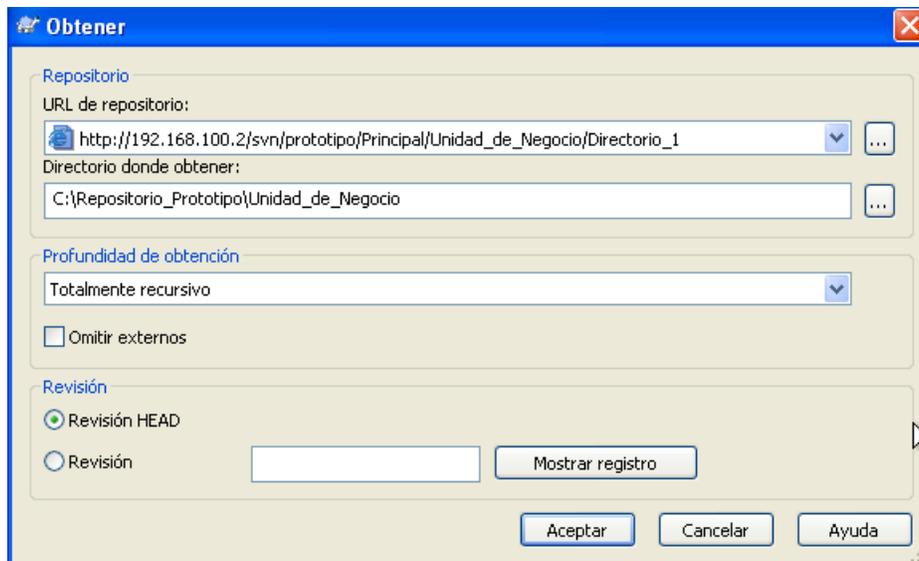
### 7.9.2. Interacción a través de la herramienta cliente *TortoiseSVN*

A continuación se presentan interacciones de archivos digitales a través de la herramienta TortoiseSVN. Ver Ilustración 61



**Ilustración 61:** Obteniendo una copia de trabajo a través de *TortoiseSVN*

Debemos especificar la URL y el directorio que se utilizará como copia de trabajo. Ver Ilustración 62.



**Ilustración 62:** GUI TortoiseSVN para establecer conexión con el Repositorio

### 7.9.3. Historial de cambios sobre un archivo

Luego de haber realizado modificaciones a un archivo específico por parte de los usuarios cliente1 y cliente 2 podremos ver el registro del historial de cambios realizado. Ver

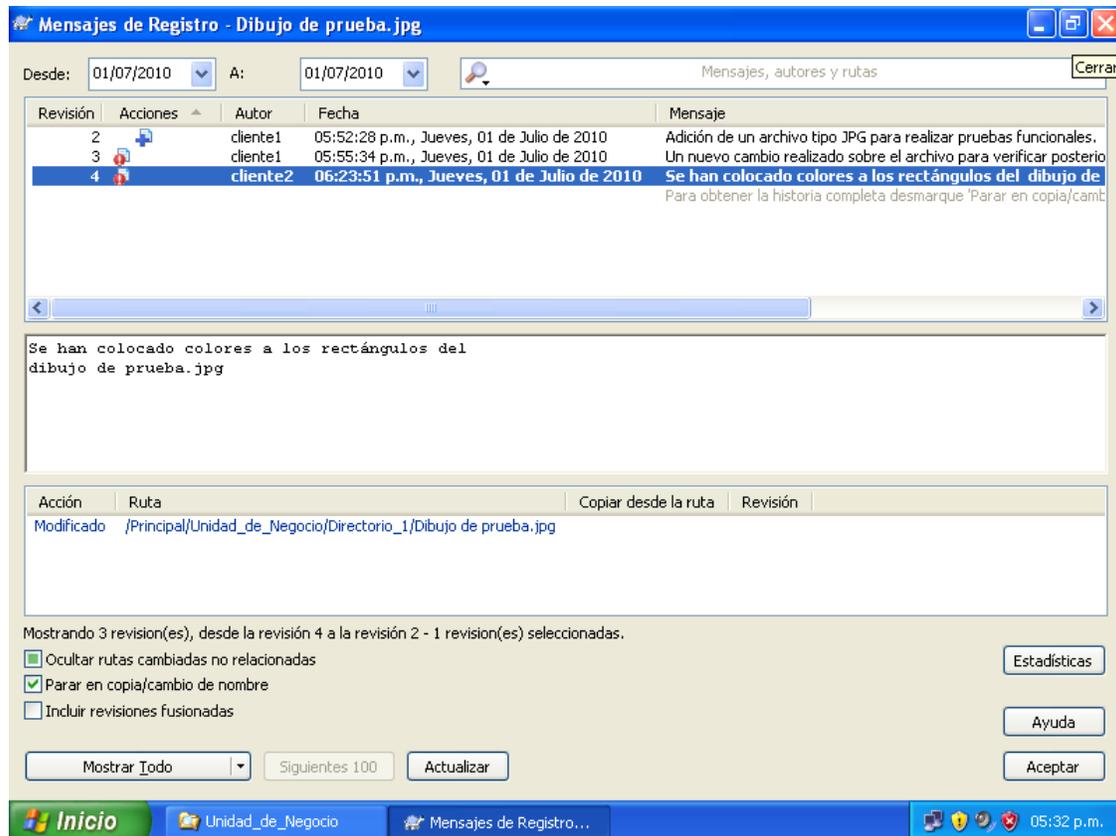


Ilustración 63: Registro con el historial de cambios de una archivo

### 7.9.4. Comparación de diferencias entre revisiones de un documento

También es posible realizar comparación entre dos revisiones del mismo documento para facilitar la resolución de conflictos y la realización de trabajos derivados. Ver Ilustración 64 y la Ilustración 65.

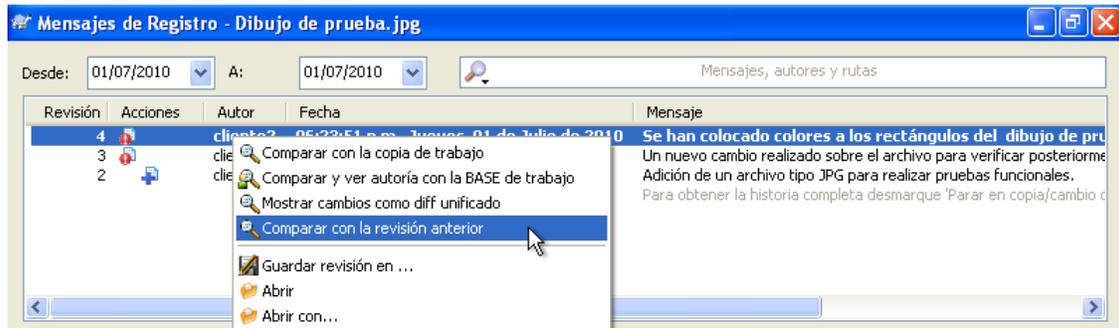


Ilustración 64: Llamado a la comparación de dos revisiones

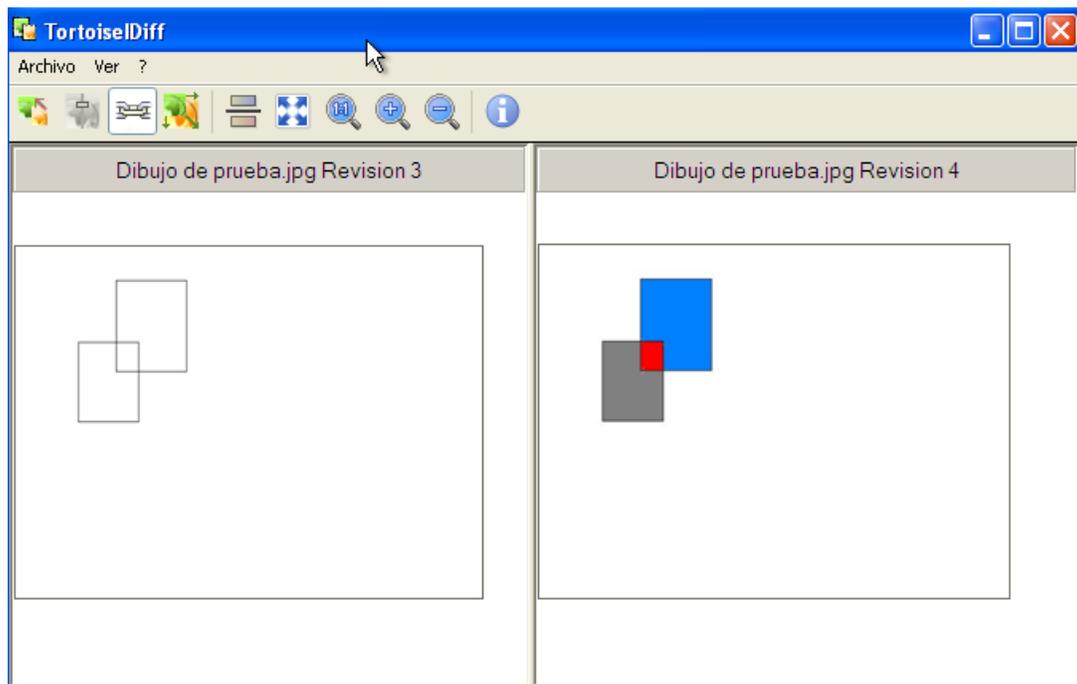


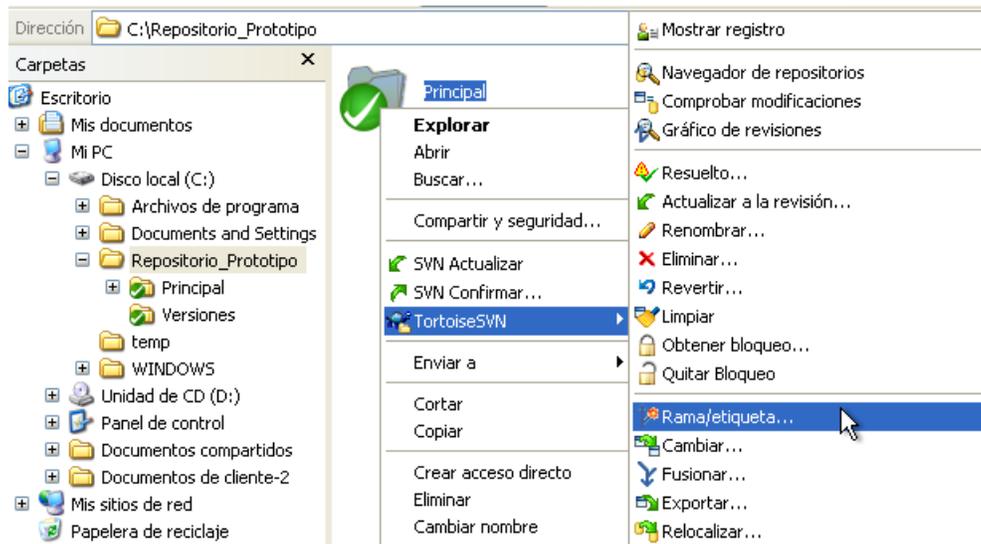
Ilustración 65: Comparación de dos revisiones de un archivo

## 7.10. Generar versiones según hitos del negocio

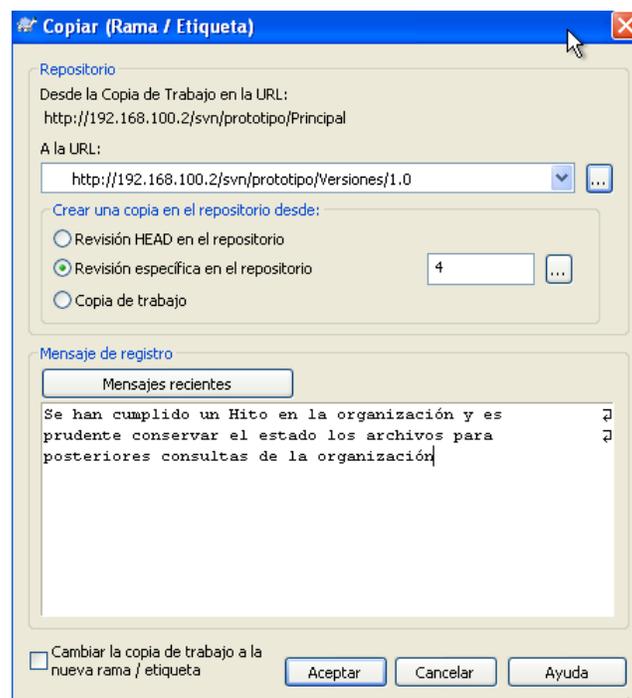
### Relación con el Proceso Gestión de Archivos Digitales

Actividad relacionada: 5.6.8 Generar Versión

Generar una versión es una actividad que deben realizar los administradores de repositorios cuando se cumplan hitos de las unidades de negocio. A continuación las ilustraciones 66 a 68 describen el procedimiento técnico que se debe seguir.



**Ilustración 66:** Opción de TortoiseSVN para la creación de una Versión



**Ilustración 67:** GUI TortoiseSVN para generación de una versión.

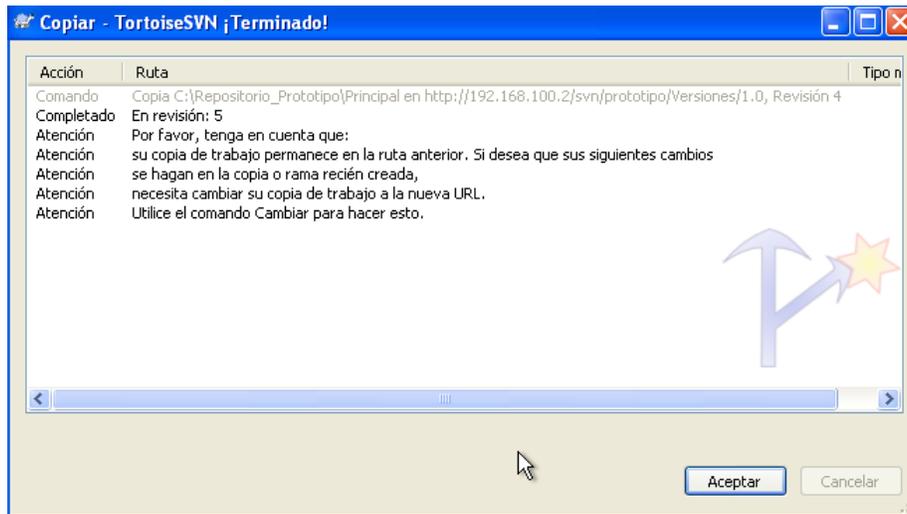


Ilustración 68: Mensaje de confirmación de la creación de una Versión

## 7.11. Aseguramiento del repositorio de información

### Relación con el Proceso Gestión de Archivos Digitales

Actividades relacionadas: 5.6.10 Respaldar información

5.6.11 Restaurar Información

### 7.11.1. Respaldo del repositorio

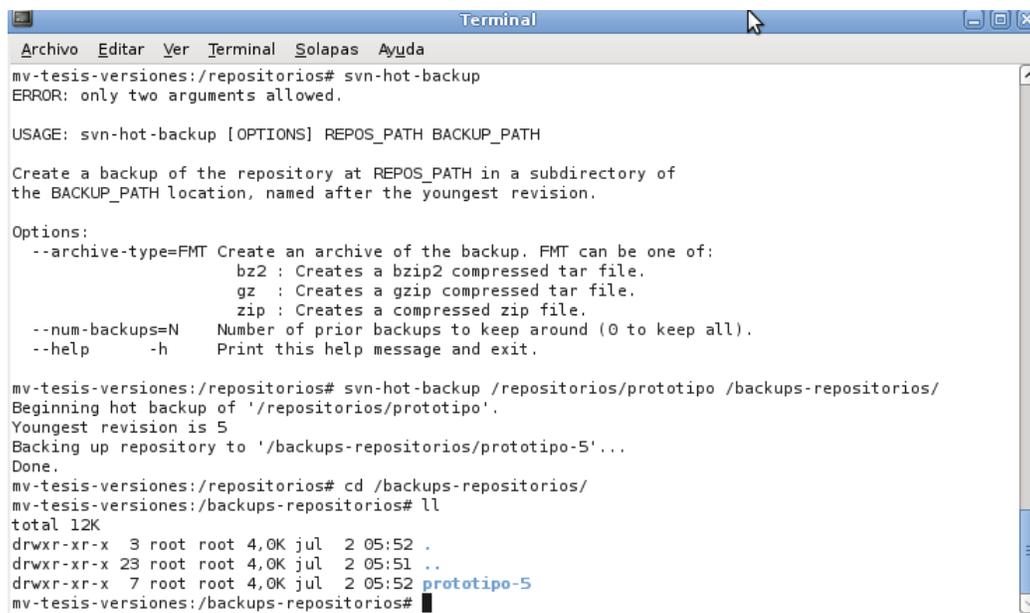
En la herramienta *Subversion* existen dos tipos de métodos de copia de seguridad disponibles para los administradores del servidor del repositorio, las copias incrementales y las copias completas (Collins-Sussan, Fitzpatrick, & Pilato, 2008).

Las copias incrementales consisten en la realización de copias de seguridad de los cambios realizados desde la última vez que se hizo la copia de seguridad anterior.

Una copia de seguridad completa del repositorio es casi una copia literal del directorio que contiene todo el repositorio.

Al momento de realizar una copia de seguridad se tiene la posibilidad de que existan transacciones en curso; esta situación conlleva al riesgo de generar una copia de seguridad errónea, dado que alguien podría estar modificando la base de datos. En consecuencia, la realización de la copia de seguridad de *Subversion* no es una actividad trivial si se desea realizar manualmente; por lo tanto, los desarrolladores de *Subversion*, han dispuesto herramientas adecuadas para realizar esta tarea; se trata del Script *svn-hot-backup*, el cual utiliza el comando *svnadmin hotcopy* para realizar adecuadamente la copia de seguridad.

La forma de utilizar el script *svn-hot-backup* para realizar la copia de seguridad completa del repositorio se muestra en la



```
Terminal
Archivo Editar Ver Terminal Solapas Ayuda
mv-tesis-versiones:/repositorios# svn-hot-backup
ERROR: only two arguments allowed.

USAGE: svn-hot-backup [OPTIONS] REPOS_PATH BACKUP_PATH

Create a backup of the repository at REPOS_PATH in a subdirectory of
the BACKUP_PATH location, named after the youngest revision.

Options:
  --archive-type=FMT Create an archive of the backup. FMT can be one of:
                        bz2 : Creates a bzip2 compressed tar file.
                        gz  : Creates a gzip compressed tar file.
                        zip  : Creates a compressed zip file.
  --num-backups=N     Number of prior backups to keep around (0 to keep all).
  --help              -h   Print this help message and exit.

mv-tesis-versiones:/repositorios# svn-hot-backup /repositorios/prototipo /backups-repositorios/
Beginning hot backup of '/repositorios/prototipo'.
Youngest revision is 5
Backing up repository to '/backups-repositorios/prototipo-5'...
Done.
mv-tesis-versiones:/repositorios# cd /backups-repositorios/
mv-tesis-versiones:/backups-repositorios# ll
total 12K
drwxr-xr-x  3 root root 4,0K jul  2 05:52 .
drwxr-xr-x 23 root root 4,0K jul  2 05:51 ..
drwxr-xr-x  7 root root 4,0K jul  2 05:52 prototipo-5
mv-tesis-versiones:/backups-repositorios#
```

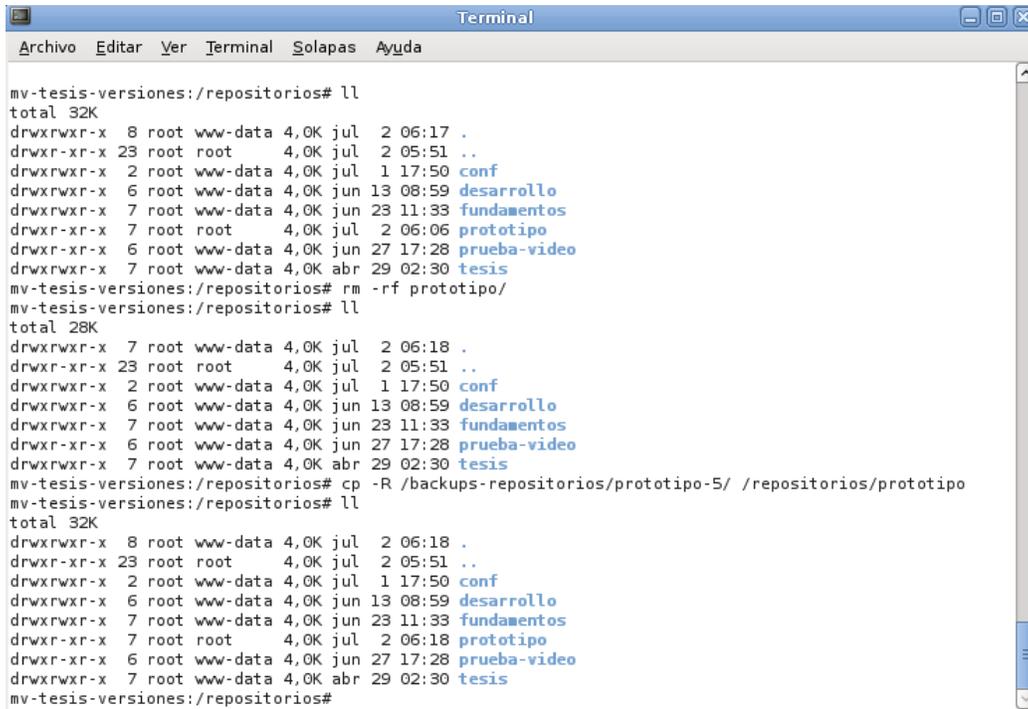
Ilustración 69: Realizando copia de seguridad del repositorio

Luego de obtener una copia del repositorio deberá comprimirse y almacenarse en diversos lugares según las políticas de gestión de copias de seguridad que existan en

la organización. Sin importar cual sea, es importante recordar la buena práctica de replicar esta información en servidores diferentes al servidor de producción y si es posible ubicar las copias de seguridad en lugares externos a cargo de compañía que se especializadas en la custodia de valores.

### 7.11.2. Restauración del repositorio

Si se ha realizado la copia con el script *svn-hot-backup*, la restauración es muy sencilla. Consiste en la reubicación del directorio que se obtuvo como resultado del script *svn-hot-backup* en el directorio respectivo para los repositorios de producción. Ver



```

Terminal
Archivo Editar Ver Terminal Solapas Ayuda

mv-tesis-versiones:/repositorios# ll
total 32K
drwxrwxr-x  8 root www-data 4,0K jul  2 06:17 .
drwxr-xr-x 23 root root    4,0K jul  2 05:51 ..
drwxrwxr-x  2 root www-data 4,0K jul  1 17:50 conf
drwxrwxr-x  6 root www-data 4,0K jun 13 08:59 desarrollo
drwxrwxr-x  7 root www-data 4,0K jun 23 11:33 fundamentos
drwxr-xr-x  7 root root    4,0K jul  2 06:06 prototipo
drwxr-xr-x  6 root www-data 4,0K jun 27 17:28 prueba-video
drwxrwxr-x  7 root www-data 4,0K abr 29 02:30 tesis
mv-tesis-versiones:/repositorios# rm -rf prototipo/
mv-tesis-versiones:/repositorios# ll
total 28K
drwxrwxr-x  7 root www-data 4,0K jul  2 06:18 .
drwxr-xr-x 23 root root    4,0K jul  2 05:51 ..
drwxrwxr-x  2 root www-data 4,0K jul  1 17:50 conf
drwxrwxr-x  6 root www-data 4,0K jun 13 08:59 desarrollo
drwxrwxr-x  7 root www-data 4,0K jun 23 11:33 fundamentos
drwxr-xr-x  6 root www-data 4,0K jun 27 17:28 prueba-video
drwxrwxr-x  7 root www-data 4,0K abr 29 02:30 tesis
mv-tesis-versiones:/repositorios# cp -R /backups-repositorios/prototipo-5/ /repositorios/prototipo
mv-tesis-versiones:/repositorios# ll
total 32K
drwxrwxr-x  8 root www-data 4,0K jul  2 06:18 .
drwxr-xr-x 23 root root    4,0K jul  2 05:51 ..
drwxrwxr-x  2 root www-data 4,0K jul  1 17:50 conf
drwxrwxr-x  6 root www-data 4,0K jun 13 08:59 desarrollo
drwxrwxr-x  7 root www-data 4,0K jun 23 11:33 fundamentos
drwxr-xr-x  7 root root    4,0K jul  2 06:18 prototipo
drwxr-xr-x  6 root www-data 4,0K jun 27 17:28 prueba-video
drwxrwxr-x  7 root www-data 4,0K abr 29 02:30 tesis
mv-tesis-versiones:/repositorios#

```

**Ilustración 70:** Restaurando un repositorio desde su copia de seguridad

La realización de actividades de interacción sobre el repositorio está sujeta a la naturaleza del comportamiento de las personas miembros de las organizaciones. Con lo expuesto en la presente sección se orienta de forma básica los aspectos que deberán seguir las personas que deben tratar con archivos digitales.

## 8. CUMPLIMIENTO DE OBJETIVOS

Objetivo específico	Producto	Observaciones
Diseñar un esquema orientado a calidad que determine una adecuada gestión de archivos digitales en entornos organizacionales	Identificación de los problemas más significativos que tienen las organizaciones en la gestión de sus archivos digitales expuesto en el capítulo 2: <i>Descripción del problema.</i>	Se utilizó como referencia un estudio realizado por el DANE llamado <i>"Uso y penetración de TIC en empresas comerciales, industriales, de servicios y microestablecimientos año 2006"</i> Expuesto en la "XI Reunión de Networking TIC - Septiembre de 2009".
	Definición del proceso <i>Gestión de Archivos Digitales</i> expuesto en el capítulo 5: <i>Esquema de calidad para la gestión de archivos digitales.</i>	Se logro especificar como extender los sistemas de control de versiones para realizar una adecuada gestión de archivos digitales en una organización.  El proceso diseñado puede ser integrado adecuadamente en un Sistema de Gestión de Calidad de cualquier organización.  Para diseñar el proceso fue necesario referirse a diversos marcos de referencia como ISO, ITIL y CMMI.
Seleccionar la herramienta de software libre adecuada para implementar una solución basada en el diseño de calidad previamente realizado	Caracterización de herramientas de software libre que permiten establecer sistemas de control de versiones expuesto en el capítulo 6: <i>Herramientas para implementar un sistema de control de versiones</i>	Se realizó un análisis de las diversas herramientas disponibles para llevar a cabo el despliegue de un sistema para el control de versiones, encontrándose una gran variedad de ellas divididas en dos categorías a saber: los sistemas de control de versiones centralizados y los sistemas de control de versiones distribuidos, siendo estos últimos implementaciones relativamente nuevas.
	Selección de Subversión como herramienta para el sistema de	Además de estudiar las características de las

	control de versiones expuesto en el capítulo 6: <i>Herramientas para implementar un sistema de control de versiones</i>	herramientas para los sistemas de control de versiones, se selecciono a la herramienta <i>Subversion</i> considerando la amplia aceptación que tiene en las comunidades de desarrollo de software, además de presentar gran estabilidad en su base funcional debido a su refinamiento a lo largo de más de nueve años de evolución.  Es adecuado considerar que el proceso <i>Gestión de archivos digitales</i> aquí propuesto puede ser implementado utilizando cualquier sistema que permita realizar control de versiones ya sea distribuido o centralizado.
Diseñar un entorno basado en máquinas virtuales para una organización genérica que permita validar la efectividad de la solución	Diseño de un entorno virtual utilizando máquinas virtuales expuesto en el capítulo 7: <i>Diseño de un entorno de operación basado en máquinas virtuales</i>	Se utilizó la herramienta para virtualización “VirtualBox” para realizar el despliegue de las máquinas virtuales correspondientes al entorno diseñado.
	Una guía de adaptación para la transferencia de conocimiento que permita poner en funcionamiento un sistema de control de versiones para la adecuada gestión de archivos, expuesto en el capítulo 7: <i>Diseño de un entorno de operación basado en máquinas virtuales</i>	La guía de adopción permite realizar una observación general al procedimiento correspondiente para la instalación del servidor de la herramienta <i>Subversion</i> , al igual que la instalación y configuración de las estaciones clientes; además, se puede evidenciar la forma que se debe realizar al interacción con el repositorio de información.

## 9. CONCLUSIONES

### ***Diseño del proceso:***

- Se diseñó el proceso *Gestión de Archivos Digitales*, el cual puede ser integrado adecuadamente en un *Sistema de Gestión de Calidad* de cualquier organización, con el propósito de entregar una especificación concreta y coherente acerca de la forma como se debe realizar la gestión de los activos de información representados en los archivos digitales.
- Para diseñar el proceso de *Gestión de Archivos Digitales* fue necesario incorporar buenas prácticas y especificaciones combinadas de los marcos de referencia metodológicos que están disponibles para las tecnologías de información tales como ISO, ITIL y CMMI.
- Las herramientas hardware y software de la tecnología informática por sí mismas no constituyen un avance significativo para una organización, si alrededor de su uso, no está definida la forma correcta para utilizarlas según las necesidades determinadas de cada empresa.

### ***Herramienta de software libre:***

- Se realizó la caracterización de diversas herramientas consideradas software libre que permiten implementar un *Sistema de Control de Versiones*.
- Las herramientas más destacadas son *Git* y *Subversion*, siendo esta última la seleccionada para el presente proyecto por su estabilidad funcional y aceptación a nivel mundial.

- El desarrollo de los *Sistemas de Control de Versiones* tiende en la actualidad a los sistemas *Distribuidos* aunque hoy en día predomine una amplia base instalada de los *Sistemas de Control de Versiones Centralizados*.

***Diseño del entorno basado en máquinas virtuales:***

- Se construyó un entorno de aplicación basado en máquinas virtuales para el proceso *Gestión de Archivos Digitales*.
- Mediante el entorno de aplicación virtual se pudo comprobar la viabilidad técnica y operativa del proceso diseñado y de la herramienta *Subversion*, dando la posibilidad de ser aplicado posteriormente en un entorno real en cualquier organización que pueda contar con los requerimientos mínimos correspondientes al diseño del ambiente virtual construido.

## 10. APORTE Y TRABAJO FUTURO

### *Aporte:*

- La definición del proceso *Gestión de Archivos Digitales* sirve adicionalmente como un ejemplo de especificación de proceso que puede ser utilizado y adaptado para derivar en la especificación de otros procesos que utilicen tecnología informática.
- El trabajo a través de ambientes simulados con máquinas virtuales facilita la realización de pruebas funcionales de procesos y herramientas, disminuyendo los costos efectivos de implementación que incluso puede ser aprovechado por las empresas para llevarlo a entornos de producción empresarial brindando ventajas competitivas a las organizaciones.

### *Trabajo futuro:*

- Realizar una implementación de la solución sobre un ambiente real de producción donde se puedan obtener métricas para el refinamiento del proceso.
- Realizar un análisis funcional y técnico de los *Sistemas de Control de Versiones Distribuidos* para posteriormente realizar una adaptación al proceso de Gestión de Archivos Digitales para las organizaciones que operen en ambientes distribuidos.



---

## 11. REFERENCIAS BIBLIOGRÁFICAS

Alvarez, J., Alvarez, I., & Bullón, J. (2005). *Introducción a La Calidad: Aproximación a Los Sistemas de Gestión y Herramientas de Calidad*.

Barrios, J. (s.f.). *Linux para todos*. Obtenido de <http://www.linuxparatodos.net/portal/staticpages/index.php?page=19-0-como-squid-general>

Benavides, C. A., & Quintana, G. C. (2003). *Gestión del conocimiento y calidad total*.

Berczuk, S., & Appleton, B. (2003). *Software Configuration Management Patterns: Effective Teamwork, Practical Integration*. Addison-Wesley.

Bon, J. V. (2008). *Fundamentos de Gestión de Servicios TI basados en ITIL*. itSMF Internacional.

Canonical Ltd. (05 de 10 de 2009). *Bazaar*. Recuperado el 2 de 6 de 2010, de <http://bazaar.canonical.com/en/>

Collins-Sussan, B., Fitzpatrick, B. W., & Pilato, M. (2008). *Version Control with Subversion*.

Cuatrecasa, L. (2000). *Seis Sigma. Una iniciativa de calidad total*. Barcelona.

DANE. (2006). *Encuesta Anual de Comercio - EAC*. Departamento Administrativo Nacional de Estadística.

Fernandez, O. M., Garcia, D., & Beltrán, A. (1995). *Un enfoque actual sobre la calidad de software*. From [http://bvs.sld.cu/revistas/aci/vol3\\_3\\_95/aci05395.htm](http://bvs.sld.cu/revistas/aci/vol3_3_95/aci05395.htm)

Garcia, A. M. (2005). *Evaluación de la calidad de los sistemas de información*.

Google Inc. (10 de 05 de 2009). *Google Apps*. Recuperado el 15 de 03 de 2010, de Google Apps: <http://www.google.com/apps/intl/es/business/index.html>

James, W., & Collins, J. (2009). *The Memory jogger 9001:2008: Implementing a Process Approach compliant to ISO 9001:2008 Quantly Management Systems Standards*. Project Editor.

Keyes, Y. (2004). *Software Configuration Management*.

Loeliger, J. (2009). *Version Control with GIT*. O'Reilly Media.

Mason, M. (2006). *Pragmatic Version Control using Subversion*. Raleigh, North Carolina Dallas, Texas, United States of America: Pragmatic Bookshelf.

Moreno Luzón, M. D., Peris, F. J., & Gonzáles, T. (2001). *Gestión de la Calidad y Diseño de Organizaciones. Teoría y estudio de casos*. Madrid, España: Pearson Educación.

Moriera, M. E. (2004). *Software Configuration Management Implementation Roadmap*.

Nagel, W. A. (2005). *Subversion Version Control: Using The Subversion Version Control System in Development Projects*.

O'Sullivan, B. (2007). *Control Distribuido de Revisiones con Mercurial*.

OGC Norma ISO 8402 - 9004. (2009). *OGC Ofical Site*. From <http://www.itil-officialsite.com>

OMG. (2008). *OMG Available Specification*. Recuperado el Mayo de 2010, de <http://www.omg.org/>

Oracle. (01 de 2007). *Virtual Box Org*. Recuperado el 10 de 05 de 2010, de <http://www.virtualbox.org/>

OSIATIS. (2009). *Osiatis*. From <http://itil.osiatis.es>

---

*Rabbitvcs.* (2009). Recuperado el 25 de 06 de 2010, de <http://rabbitvcs.org/>

Schwaber, C., Gilpin, M., & Stone, J. (2007). The Forrester Wave. *Software Change And Configuration Management*, (págs. 1-21).

Stansberry, G. (18 de 09 de 2008). *Smashing Magazine*. Recuperado el 23 de 05 de 2010, de 7 Version Control Systems Reviewed: <http://www.smashingmagazine.com/2008/09/18/the-top-7-open-source-version-control-systems/>

Swicegood, T. (2008). *Pragmatic Version Control Using GIT*. Pragmatic Bookshelf.

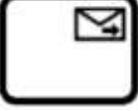
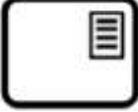
Vesperman, J. (2006). *Essential CVS* (Secon Edition ed.). O'Reilly Media.

## 12. ANEXOS

### 12.1. Anexo 1: Elementos básicos para la diagramación de BPD

#### 12.1.1. Tarea o Actividad

Representan el trabajo realizado dentro de una organización. Consumen recursos. Pueden ser simples o compuestas. Existen diferentes tipos:

Símbolo	Nombre y descripción
	Forma clásica para definir una tarea genérica
	Tarea desarrollada por un Usuario
	Tarea desarrollada de forma Manual
	Tarea desarrollada en forma de Servicio
	Tarea relacionada con el Envío de información
	Tarea relacionada con la Recepción de información
	Tarea construida en forma de script

	Referencia una Tarea
--	----------------------

### 12.1.2. Subproceso

Es una actividad compuesta que incluye un conjunto interno lógico de actividades (proceso) y que puede ser analizado en más detalle.

Símbolo	Nombre y descripción
	Subproceso embebido: Depende del proceso padre. No puede contener subdivisiones
	Subproceso reusable: Es un proceso definido como un diagrama de procesos independiente y que no depende del proceso padre.

### 12.1.3. Compuertas

Las compuertas son los elementos utilizados para controlar la divergencia y convergencia del flujo de un proceso.

	<p>Compuerta exclusiva basada en datos</p> <p><i>Divergencia:</i> Ocurre cuando se escoge un solo camino de varios disponibles.</p> <p><i>Convergencia:</i> Es utilizada para confluir caminos excluyentes.</p>
--	---

	<p>Compuerta exclusiva basada en eventos: representa un punto del proceso donde se escoge un camino de varios disponibles, pero la decisión no se basa en datos del proceso sino en eventos.</p>
	<p>Compuerta paralela</p> <p><i>Divergencia:</i> Se utiliza cuando varias actividades pueden realizarse concurrentemente o en paralelo.</p> <p><i>Convergencia:</i> Permite sincronizar varios caminos paralelos en uno solo. El flujo continúa cuando todos los flujos de secuencia de entrada hayan llegado a figura.</p>
	<p>Compuerta inclusiva</p> <p><i>Divergencia:</i> Se utiliza cuando en un punto se activan uno o más caminos de varios disponibles, basado en los datos del proceso.</p> <p><i>Convergencia:</i> Se utiliza para sincronizar caminos activados previamente por una compuerta inclusiva usada como punto de divergencia.</p>
	<p>Compuerta compleja</p> <p><i>Divergencia:</i> Es utilizada para controlar puntos de decisión complejos.</p> <p><i>Convergencia:</i> Permite continuar al siguiente punto del proceso cuando una condición de negocio se cumple.</p>

#### 12.1.4. Eventos

Un evento representa algo que ocurre o puede ocurrir durante el curso de un proceso. Existen 3 tipos de eventos basados en cómo afectan el flujo de los procesos.

#### 12.1.5. Eventos de Inicio

Indica cuando un proceso inicia, no tiene flujos de secuencia entrantes.

	<p>Eventos de inicio sin especificar:</p> <p>No se especifica ningún comportamiento en particular para iniciar el proceso. Comúnmente usado en subprocessos.</p>
	<p>Evento de inicio de mensaje</p> <p>Un proceso inicia cuando un mensaje es recibido. El mensaje es enviado por otro proceso.</p>
	<p>Evento de inicio de temporización</p> <p>Indica que un proceso inicia cada ciclo de tiempo o en una fecha específica.</p>
	<p><b>Evento de inicio de condición</b></p> <p>Un proceso iniciar cuando una condición de negocio se cumple.</p>
	<p><b>Evento de inicio de señal</b></p> <p>El proceso inicial cuando se captura una señal lanzada desde otro proceso.</p>
	<p><b>Evento de inicio múltiple</b></p> <p>Indica que existen muchas formas de iniciar el proceso y que al cumplirse una de ellas se iniciará el proceso.</p>

### 12.1.6. Eventos intermedios

Indican que ocurre o puede ocurrir durante el transcurso de un proceso, entre el inicio y el fin. Los eventos intermedios pueden utilizarse dentro del flujo de secuencia, o adjuntos a los límites de una actividad; También pueden utilizarse para recibir o lanzar disparadores. Cuando el evento es usado para recibir el icono al interior del círculo se encuentra sin rellenar, cuando el evento es usado para lanzar el icono se encuentra relleno.

	<p><b>Evento intermedio sin especificar</b></p> <p>Indica algo que ocurre o puede ocurrir dentro del proceso, sólo se puede utilizar dentro del flujo de secuencia</p>
	<p><b>Evento intermedio de mensaje</b></p> <p>Indica que un mensaje puede ser enviado o recibido. Si el evento de mensaje es de recepción, indica que el proceso no continúa hasta que el mensaje sea recibido. Puede utilizarse dentro del flujo de secuencia para recibir o enviar el mensaje o adjunto a los límites de una actividad indicando un flujo de excepción cuando se reciba el mensaje.</p>
	<p><b>Evento intermedio de temporización</b></p> <p>Indica una espera dentro del proceso. Este tipo de evento puede utilizarse dentro del flujo de secuencia indicando una espera entre las actividades o adjunto a los límites de una actividad indicando un flujo de excepción.</p>
	<p><b>Evento intermedio de Condición</b></p> <p>Se utiliza para esperar que una condición de negocio se cumpla. Se puede utilizar dentro del flujo de secuencia indicando que se espera a que la condición de negocio se cumpla o adjunto a los límites de una actividad indicando un flujo de excepción que se activará cuando la condición se cumpla.</p>
	<p><b>Evento intermedio de señal</b></p> <p>Se utiliza para enviar o recibir señales. Se puede utilizar dentro del flujo de secuencia para enviar o recibir señales o adjunto a los límites de una actividad indicando un flujo de excepción que se activará cuando la señal sea capturada.</p>

	<p><b>Evento intermedio múltiple</b></p> <p>Esto significa que existen múltiples disparadores asignados al evento. Si el evento es diagramado dentro del flujo de secuencia puede recibir o lanzar los disparadores. Si se encuentra adjunto a los límites de una actividad sólo puede ser utilizado para recibir el disparador. Cuando es usado para recibir, sólo uno de los disparadores asociados al evento es requerido. Cuando es usado para lanzar, todos los disparadores asociados serán lanzados.</p>
	<p><b>Evento intermedio de Cancelación</b></p> <p>Este tipo de evento intermedio es usado en subprocesos Transaccionales. Se diagrama a los límites del Subproceso transaccional indicando un flujo alternativo que se realizaría cuando el subproceso transaccional es cancelado. Se diagrama a los límites del subproceso</p>
	<p><b>Evento intermedio de Error</b></p> <p>Utilizado para capturar errores. Se diagrama a los límites de una actividad.</p>
	<p><b>Evento intermedio de Compensación</b></p> <p>Permite manejar compensaciones, cuando se utiliza dentro del flujo de secuencia de un proceso indica que una compensación es necesaria, es decir que lanza una compensación. Cuando se utiliza adjunto a los límites de una actividad indica que esa actividad será compensada cuando se active el evento.</p>
	<p><b>Evento intermedio de Enlace</b></p> <p>Permite manejar compensaciones, cuando se utiliza dentro del flujo de secuencia de un proceso indica que una compensación es necesaria, es decir se lanza una compensación. Cuando se utiliza adjunto a los límites de una</p>

	actividad indica que esa actividad será compensada cuando se active el evento.
--	--

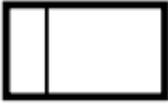
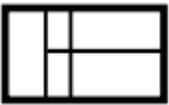
### 12.1.7. Eventos de Fin

Indican cuando un camino del proceso finaliza. No tiene flujos de secuencia saliendo.

	<b>Evento de Fin sin especificar</b> Indica que un camino del flujo llego al fin
	<b>Evento de Fin de Mensaje</b> Permite enviar un mensaje al finalizar el flujo
	<b>Evento de Fin de Señal</b> Permite enviar una señal al finalizar el flujo.
	<b>Evento de Fin Múltiple</b> Indica que varios resultados pueden darse al finalizar un flujo.
	<b>Evento de Fin de Cancelación</b> Permite enviar una excepción de cancelación al finalizar el flujo. Sólo se utiliza en subprocesos transaccionales.
	<b>Evento de Fin de Error</b> Permite enviar una excepción de error al finalizar el flujo
	<b>Evento de Fin de Compensación</b> Este tipo de fin indica que es necesaria una compensación al finalizar el flujo.

	<p><b>Evento de Fin de Terminal</b></p> <p>Indica que el proceso es terminado, es decir cuando algún camino del flujo llega a este fin el proceso termina completamente, sin importar que existan más caminos del flujo pendientes.</p>
---	---

### 12.1.8. Canales

	<p><b>Pool</b></p> <p>Actúa como contenedor de un proceso. El nombre del pools puede ser el del proceso o el del participante (actor).</p>
	<p><b>Lane</b></p> <p>Subdivisiones del Pool. Representan los diferentes participantes al interior de una organización.</p>

### 12.1.9. Objetos de conexión

	<p><b>Secuencia</b></p> <p>Representan el control de flujo y la secuencia de las actividades. Se utilizan para representar la secuencia de los objetos de flujo, donde se encuentran las actividades, las compuertas y los eventos.</p>
	<p><b>Condicional</b></p> <p>Representa el control de flujo asociado a una condición específica.</p>
	<p><b>Por defecto</b></p> <p>Representa el control de flujo que se debe utilizar de forma predeterminada en el caso que existan diversas alternativas y no</p>

	se especifica una determinada.
	<b>Mensaje</b> Representa la interacción entre varios procesos o pools. Representa Señales o Mensajes NO flujos de control.
	<b>Asociaciones</b> Se usan para asociar información adicional sobre el proceso.

### 12.1.10. Artefactos

	<b>Anotaciones</b> Son utilizados para proporcionar información adicional sobre el proceso
	<b>Grupos</b> Se utiliza para agrupar un conjunto de actividades ya sea para efectos de documentación o análisis, no afecta la secuencia del flujo.
	<b>Objetos de datos</b> Permite mostrar la información que una actividad necesita, como las entradas y las salidas. Es decir, representan los documentos, la información y otros objetos que son usados o actualizados durante el proceso. Los objetos de datos no afectan directamente los flujos de secuencia o los flujos de mensajes del proceso.

## 12.2. Anexo 2: Videos de apoyo para la operación de las herramientas utilizadas

<b>Video 01 - Instalación de Subversion</b>	
Descripción	Procedimiento para la instalación de la herramienta <i>Subversion</i>
Enlace	<a href="http://luixip.mysites.com/get_file/video-01-instalacion-de-subvers.mp4">http://luixip.mysites.com/get_file/video-01-instalacion-de-subvers.mp4</a>

<b>Video 02 - Creación de un repositorio</b>	
Descripción	Procedimiento para la creación de un repositorio
Enlace	<a href="http://luixip.mysites.com/get_file/video-02-creaci-n-de-un-reposit.mp4">http://luixip.mysites.com/get_file/video-02-creaci-n-de-un-reposit.mp4</a>

<b>Video 03 - Creación de usuarios para el repositorio</b>	
Descripción	Procedimiento para la creación de usuarios para el repositorio
Enlace	<a href="http://luixip.mysites.com/get_file/video-03-creaci-n-de-usuarios-p.mp4">http://luixip.mysites.com/get_file/video-03-creaci-n-de-usuarios-p.mp4</a>

<b>Video 04 - Poblado inicial del repositorio</b>	
Descripción	Poblar los datos iniciales del repositorio. Lo que se utiliza generalmente es cargar la estructura de directorios según las necesidades de cada organización.
Enlace	<a href="http://luixip.mysites.com/get_file/video-04-poblado-inicial-del-re.mp4">http://luixip.mysites.com/get_file/video-04-poblado-inicial-del-re.mp4</a>

<b>Video 05 - Asignación de permisos</b>	
Descripción	Asignación de permisos para el control de acceso al repositorio de información de <i>Subversion</i>
Enlace	<a href="http://luixip.mysites.com/get_file/video-05-asignaci-n-de-permisos.mp4">http://luixip.mysites.com/get_file/video-05-asignaci-n-de-permisos.mp4</a>

<b>Video 06 - Cliente Subversion - RabbitVCS -GNU_Linux Ubuntu</b>	
Descripción	Procedimiento de Instalación de la herramienta <i>RabbitVCS</i> sobre el sistema operativo GNU/Linux Ubuntu para realizar la gestión de repositorios de <i>Subversion</i> .

Enlace	<a href="http://luixip.mysites.com/get_file/video-06-cliente-subversion-rab.mp4">http://luixip.mysites.com/get_file/video-06-cliente-subversion-rab.mp4</a>
--------	---

#### Video 07 - Cliente Subversion - TortoiseSVN - MS\_Windows XP

Descripción	Procedimiento de Instalación de la herramienta <i>TortoiseSVN</i> sobre el sistema operativo MS-Microsoft Windows XP para realizar la gestión de repositorios de <i>Subversion</i> .
Enlace	<a href="http://luixip.mysites.com/get_file/video-07-cliente-subversion-tor.mp4">http://luixip.mysites.com/get_file/video-07-cliente-subversion-tor.mp4</a>

#### Video 08 - Obtener una copia de trabajo - RabbitVCS - GNU\_Linux Ubuntu

Descripción	Procedimiento para obtener una copia de trabajo del repositorio de Subversion utilizando la herramienta <i>RabbitVCS</i> sobre el sistema operativo GNU/Linux Ubuntu.
Enlace	<a href="http://luixip.mysites.com/get_file/video-08-obtener-una-copia-de-t.mp4">http://luixip.mysites.com/get_file/video-08-obtener-una-copia-de-t.mp4</a>

#### Video 09 - Obtener una copia de trabajo - TortoiseSVN - MS Windows XP

Descripción	Obtener una copia de trabajo del repositorio de <i>Subversion</i> utilizando la herramienta <i>TortoiseSVN</i> sobre el sistema operativo MS Windows XP.
Enlace	<a href="http://luixip.mysites.com/get_file/video-09-obtener-una-copia-de-t.mp4">http://luixip.mysites.com/get_file/video-09-obtener-una-copia-de-t.mp4</a>

#### Video 10 - Adicionando archivo al repositorio - RabbitVCS - GNU/Linux Ubuntu

Descripción	Adicionando un archivo al repositorio de <i>Subversion</i> utilizando la herramienta <i>RabbitVCS</i>
Enlace	<a href="http://luixip.mysites.com/get_file/video-10-adicionando-archivo-al.mp4">http://luixip.mysites.com/get_file/video-10-adicionando-archivo-al.mp4</a>

#### Video 11 - Comparaciones entre archivos del repositorio - TortoiseSVN - MS\_Windows\_XP

Descripción	Comparación entre archivos del repositorio
Enlace	<a href="http://luixip.mysites.com/get_file/video-11-comparaciones-entre-ar.mp4">http://luixip.mysites.com/get_file/video-11-comparaciones-entre-ar.mp4</a>