
	Implantación de un sistema de control de versiones	 Mapro test&assemblysystems	carburesGroup
---	---	--	---------------

Implantación de un sistema de control de versiones



Título: Implantación de un sistema de control de versiones

Empresa: Mapro Sistemas de Ensayo

Web: <http://www.maprotest.com>

Autor: Alexis Abrutsky

Fecha: 10/07/2016

1.1 Appendix A. Licencia de Documentación Libre de GNU

Versión 1.2, Noviembre 2002

This is an unofficial translation of the GNU Free Documentation License into Spanish. It was not published by the Free Software Foundation, and does not legally state the distribution terms for documentation that uses the GNU FDL -- only the original English text of the GNU FDL does that. However, we hope that this translation will help Spanish speakers understand the GNU FDL better.

Ésta es una traducción no oficial de la GNU Free Document License a Español (Castellano). No ha sido publicada por la Free Software Foundation y no establece legalmente los términos de distribución para trabajos que usen la GFDL (sólo el texto de la versión original en Inglés de la GFDL lo hace). Sin embargo, esperamos que esta traducción ayude los hispanohablantes a entender mejor la GFDL. La versión original de la GFDL esta disponible en la [Free Software Foundation](#).

Esta traducción está basada en una de la versión 1.1 de Igor Támara y Pablo Reyes. Sin embargo la responsabilidad de su interpretación es de Joaquín Seoane.

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. Se permite la copia y distribución de copias literales de este documento de licencia, pero no se permiten cambios[1].

1.2 A.1. PREÁMBULO

El propósito de esta Licencia es permitir que un manual, libro de texto, u otro documento escrito sea *libre* en el sentido de libertad: asegurar a todo el mundo la libertad efectiva de copiarlo y redistribuirlo, con o sin modificaciones, de manera comercial o no. En segundo término, esta Licencia proporciona al autor y al editor[2] una manera de obtener reconocimiento por su trabajo, sin que se le considere responsable de las modificaciones realizadas por otros.

Esta Licencia es de tipo *copyleft*, lo que significa que los trabajos derivados del documento deben a su vez ser libres en el mismo sentido. Complementa la Licencia Pública General de GNU, que es una licencia tipo copyleft diseñada para el software libre.

Hemos diseñado esta Licencia para usarla en manuales de software libre, ya que el software libre necesita documentación libre: un programa libre debe venir con manuales que ofrezcan la mismas libertades que el software. Pero esta licencia no se limita a manuales de software; puede usarse para cualquier texto, sin tener en cuenta su temática o si se publica como libro impreso o no. Recomendamos esta licencia principalmente para trabajos cuyo fin sea instructivo o de referencia.

1.3 A.2. APLICABILIDAD Y DEFINICIONES

Esta Licencia se aplica a cualquier manual u otro trabajo, en cualquier soporte, que contenga una nota del propietario de los derechos de autor que indique que puede ser distribuido bajo los términos de esta Licencia. Tal nota garantiza en cualquier lugar del mundo, sin pago de derechos y sin límite de tiempo, el uso de dicho trabajo según las condiciones aquí estipuladas. En adelante la palabra *Documento* se referirá a cualquiera de dichos manuales o trabajos. Cualquier persona es un licenciataria y será referido como *Usted*. Usted acepta la licencia si copia, modifica o distribuye el trabajo de cualquier modo que requiera permiso según la ley de propiedad intelectual.

Una *Versión Modificada* del Documento significa cualquier trabajo que contenga el Documento o una porción del mismo, ya sea una copia literal o con modificaciones y/o traducciones a otro idioma.

Una *Sección Secundaria* es un apéndice con título o una sección preliminar del Documento que trata exclusivamente de la relación entre los autores o editores y el tema general del Documento (o temas relacionados) pero que no contiene nada que entre directamente en dicho tema general (por ejemplo, si el Documento es en parte un texto de matemáticas, una Sección Secundaria puede no explicar nada de matemáticas). La relación puede ser una conexión histórica con el tema o temas relacionados, o una opinión legal, comercial, filosófica, ética o política acerca de ellos.

Las *Secciones Invariantes* son ciertas Secciones Secundarias cuyos títulos son designados como Secciones Invariantes en la nota que indica que el documento es liberado bajo esta Licencia. Si una sección no entra en la definición de Secundaria, no puede designarse como Invariante. El documento puede no tener Secciones Invariantes. Si el Documento no identifica las Secciones Invariantes, es que no las tiene.

Los *Textos de Cubierta* son ciertos pasajes cortos de texto que se listan como Textos de Cubierta Delantera o Textos de Cubierta Trasera en la nota que indica que el documento es liberado bajo esta Licencia. Un Texto de Cubierta Delantera puede tener como mucho 5 palabras, y uno de Cubierta Trasera puede tener hasta 25 palabras.

Una copia *Transparente* del Documento, significa una copia para lectura en máquina, representada en un formato cuya especificación está disponible al público en general, apto para que los contenidos puedan ser vistos y editados directamente con editores de texto genéricos o (para imágenes compuestas por puntos) con programas genéricos de manipulación de imágenes o (para dibujos) con algún editor de dibujos ampliamente disponible, y que sea adecuado como entrada para formateadores de texto o para su traducción automática a formatos adecuados para formateadores de texto. Una copia hecha en un formato definido como Transparente, pero cuyo marcaje o ausencia de él haya sido diseñado para impedir o dificultar modificaciones posteriores por parte de los lectores no es Transparente. Un formato de imagen no es Transparente si se usa para una cantidad de texto sustancial. Una copia que no es *Transparente* se denomina *Opaca*.

Como ejemplos de formatos adecuados para copias Transparentes están ASCII puro sin marcaje, formato de entrada de Texinfo, formato de entrada de LaTeX, SGML o XML

usando una DTD disponible públicamente, y HTML, PostScript o PDF simples, que sigan los estándares y diseñados para que los modifiquen personas. Ejemplos de formatos de imagen transparentes son PNG, XCF y JPG. Los formatos Opacos incluyen formatos propietarios que pueden ser leídos y editados únicamente en procesadores de palabras propietarios, SGML o XML para los cuáles las DTD y/o herramientas de procesamiento no estén ampliamente disponibles, y HTML, PostScript o PDF generados por algunos procesadores de palabras sólo como salida.

La *Portada* significa, en un libro impreso, la página de título, más las páginas siguientes que sean necesarias para mantener legiblemente el material que esta Licencia requiere en la portada. Para trabajos en formatos que no tienen página de portada como tal, *Portada* significa el texto cercano a la aparición más prominente del título del trabajo, precediendo el comienzo del cuerpo del texto.

Una sección *Titulada XYZ* significa una parte del Documento cuyo título es precisamente XYZ o contiene XYZ entre paréntesis, a continuación de texto que traduce XYZ a otro idioma (aquí XYZ se refiere a nombres de sección específicos mencionados más abajo, como *Agradecimientos*, *Dedicatorias*, *Aprobaciones* o *Historia*. *Conservar el Título* de tal sección cuando se modifica el Documento significa que permanece una sección *Titulada XYZ* según esta definición[3].

El Documento puede incluir Limitaciones de Garantía cercanas a la nota donde se declara que al Documento se le aplica esta Licencia. Se considera que estas Limitaciones de Garantía están incluidas, por referencia, en la Licencia, pero sólo en cuanto a limitaciones de garantía: cualquier otra implicación que estas Limitaciones de Garantía puedan tener es nula y no tiene efecto en el significado de esta Licencia.

1.4 A.3. COPIA LITERAL

Usted puede copiar y distribuir el Documento en cualquier soporte, sea en forma comercial o no, siempre y cuando esta Licencia, las notas de copyright y la nota que indica que esta Licencia se aplica al Documento se reproduzcan en todas las copias y que usted no añada ninguna otra condición a las expuestas en esta Licencia. Usted no puede usar medidas técnicas para obstruir o controlar la lectura o copia posterior de las copias que usted haga o distribuya. Sin embargo, usted puede aceptar compensación a cambio de las copias. Si distribuye un número suficientemente grande de copias también deberá seguir las condiciones de la sección 3.

Usted también puede prestar copias, bajo las mismas condiciones establecidas anteriormente, y puede exhibir copias públicamente.

1.5 A.4. COPIADO EN CANTIDAD

Si publica copias impresas del Documento (o copias en soportes que tengan normalmente cubiertas impresas) que sobrepasen las 100, y la nota de licencia del Documento

exige Textos de Cubierta, debe incluir las copias con cubiertas que lleven en forma clara y legible todos esos Textos de Cubierta: Textos de Cubierta Delantera en la cubierta delantera y Textos de Cubierta Trasera en la cubierta trasera. Ambas cubiertas deben identificarlo a Usted clara y legiblemente como editor de tales copias. La cubierta debe mostrar el título completo con todas las palabras igualmente prominentes y visibles. Además puede añadir otro material en las cubiertas. Las copias con cambios limitados a las cubiertas, siempre que conserven el título del Documento y satisfagan estas condiciones, pueden considerarse como copias literales.

Si los textos requeridos para la cubierta son muy voluminosos para que ajusten legiblemente, debe colocar los primeros (tantos como sea razonable colocar) en la verdadera cubierta y situar el resto en páginas adyacentes.

Si Usted publica o distribuye copias Opacas del Documento cuya cantidad exceda las 100, debe incluir una copia Transparente, que pueda ser leída por una máquina, con cada copia Opaca, o bien mostrar, en cada copia Opaca, una dirección de red donde cualquier usuario de la misma tenga acceso por medio de protocolos públicos y estandarizados a una copia Transparente del Documento completa, sin material adicional. Si usted hace uso de la última opción, deberá tomar las medidas necesarias, cuando comience la distribución de las copias Opacas en cantidad, para asegurar que esta copia Transparente permanecerá accesible en el sitio establecido por lo menos un año después de la última vez que distribuya una copia Opaca de esa edición al público (directamente o a través de sus agentes o distribuidores).

Se solicita, aunque no es requisito, que se ponga en contacto con los autores del Documento antes de redistribuir gran número de copias, para darles la oportunidad de que le proporcionen una versión actualizada del Documento.

1.6 A.5. MODIFICACIONES

Puede copiar y distribuir una Versión Modificada del Documento bajo las condiciones de las secciones 2 y 3 anteriores, siempre que usted libere la Versión Modificada bajo esta misma Licencia, con la Versión Modificada haciendo el rol del Documento, por lo tanto dando licencia de distribución y modificación de la Versión Modificada a quienquiera posea una copia de la misma. Además, debe hacer lo siguiente en la Versión Modificada:

- A. Usar en la Portada (y en las cubiertas, si hay alguna) un título distinto al del Documento y de sus versiones anteriores (que deberían, si hay alguna, estar listadas en la sección de Historia del Documento). Puede usar el mismo título de versiones anteriores al original siempre y cuando quien las publicó originalmente otorgue permiso.
- B. Listar en la Portada, como autores, una o más personas o entidades responsables de la autoría de las modificaciones de la Versión Modificada, junto con por lo menos cinco de los autores principales del Documento (todos sus autores principales, si hay menos de cinco), a menos que le eximan de tal requisito.

- C. Mostrar en la Portada como editor el nombre del editor de la Versión Modificada.
- D. Conservar todas las notas de copyright del Documento.
- E. Añadir una nota de copyright apropiada a sus modificaciones, adyacente a las otras notas de copyright.
- F. Incluir, inmediatamente después de las notas de copyright, una nota de licencia dando el permiso para usar la Versión Modificada bajo los términos de esta Licencia, como se muestra en la Adenda al final de este documento.
- G. Conservar en esa nota de licencia el listado completo de las Secciones Invariantes y de los Textos de Cubierta que sean requeridos en la nota de Licencia del Documento original.
- H. Incluir una copia sin modificación de esta Licencia.
- I. Conservar la sección Titulada *Historia*, conservar su Título y añadirle un elemento que declare al menos el título, el año, los nuevos autores y el editor de la Versión Modificada, tal como figuran en la Portada. Si no hay una sección Titulada *Historia* en el Documento, crear una estableciendo el título, el año, los autores y el editor del Documento, tal como figuran en su Portada, añadiendo además un elemento describiendo la Versión Modificada, como se estableció en la oración anterior.
- J. Conservar la dirección en red, si la hay, dada en el Documento para el acceso público a una copia Transparente del mismo, así como las otras direcciones de red dadas en el Documento para versiones anteriores en las que estuviese basado. Pueden ubicarse en la sección *Historia*. Se puede omitir la ubicación en red de un trabajo que haya sido publicado por lo menos cuatro años antes que el Documento mismo, o si el editor original de dicha versión da permiso.
- K. En cualquier sección Titulada *Agradecimientos* o *Dedicatorias*, Conservar el Título de la sección y conservar en ella toda la sustancia y el tono de los agradecimientos y/o dedicatorias incluidas por cada contribuyente.
- L. Conservar todas las Secciones Invariantes del Documento, sin alterar su texto ni sus títulos. Números de sección o el equivalente no son considerados parte de los títulos de la sección.
- M. Borrar cualquier sección titulada *Aprobaciones*. Tales secciones no pueden estar incluidas en las Versiones Modificadas.
- N. No cambiar el título de ninguna sección existente a *Aprobaciones* ni a uno que entre en conflicto con el de alguna Sección Invariante.
- O. Conservar todas las Limitaciones de Garantía.

Si la Versión Modificada incluye secciones o apéndices nuevos que califiquen como Secciones Secundarias y contienen material no copiado del Documento, puede opcionalmente designar algunas o todas esas secciones como invariantes. Para hacerlo, añada sus títulos a la lista de Secciones Invariantes en la nota de licencia de la Versión Modificada. Tales títulos deben ser distintos de cualquier otro título de sección.

Puede añadir una sección titulada *Aprobaciones*, siempre que contenga únicamente aprobaciones de su Versión Modificada por otras fuentes --por ejemplo, observaciones de peritos o que el texto ha sido aprobado por una organización como la definición oficial de un estándar.

Puede añadir un pasaje de hasta cinco palabras como Texto de Cubierta Delantera y un pasaje de hasta 25 palabras como Texto de Cubierta Trasera en la Versión Modificada. Una entidad solo puede añadir (o hacer que se añada) un pasaje al Texto de Cubierta Delantera y uno al de Cubierta Trasera. Si el Documento ya incluye un textos de cubiertas añadidos previamente por usted o por la misma entidad que usted representa, usted no puede añadir otro; pero puede reemplazar el anterior, con permiso explícito del editor que agregó el texto anterior.

Con esta Licencia ni los autores ni los editores del Documento dan permiso para usar sus nombres para publicidad ni para asegurar o implicar aprobación de cualquier Versión Modificada.

1.7 A.6. COMBINACIÓN DE DOCUMENTOS




Usted puede combinar el Documento con otros documentos liberados bajo esta Licencia, bajo los términos definidos en la sección 4 anterior para versiones modificadas, siempre que incluya en la combinación todas las Secciones Invariantes de todos los documentos originales, sin modificar, listadas todas como Secciones Invariantes del trabajo combinado en su nota de licencia. Así mismo debe incluir la Limitación de Garantía.

El trabajo combinado necesita contener solamente una copia de esta Licencia, y puede reemplazar varias Secciones Invariantes idénticas por una sola copia. Si hay varias Secciones Invariantes con el mismo nombre pero con contenidos diferentes, haga el título de cada una de estas secciones único añadiéndole al final del mismo, entre paréntesis, el nombre del autor o editor original de esa sección, si es conocido, o si no, un número único. Haga el mismo ajuste a los títulos de sección en la lista de Secciones Invariantes de la nota de licencia del trabajo combinado.

En la combinación, debe combinar cualquier sección Titulada *Historia* de los documentos originales, formando una sección Titulada *Historia*; de la misma forma combine cualquier sección Titulada *Agradecimientos*, y cualquier sección Titulada *Dedicatorias*. Debe borrar todas las secciones tituladas *Aprobaciones*.

1.8 A.7. COLECCIONES DE DOCUMENTOS

Puede hacer una colección que conste del Documento y de otros documentos liberados bajo esta Licencia, y reemplazar las copias individuales de esta Licencia en todos los documentos por una sola copia que esté incluida en la colección, siempre que siga las reglas de esta Licencia para cada copia literal de cada uno de los documentos en cualquiera de los demás aspectos.

	Implantación de un sistema de control de versiones	  <small>test&assemblysystems</small>
---	---	--

Puede extraer un solo documento de una de tales colecciones y distribuirlo individualmente bajo esta Licencia, siempre que inserte una copia de esta Licencia en el documento extraído, y siga esta Licencia en todos los demás aspectos relativos a la copia literal de dicho documento.

1.9 A.8. AGREGACIÓN CON TRABAJOS INDEPENDIENTES

Una recopilación que conste del Documento o sus derivados y de otros documentos o trabajos separados e independientes, en cualquier soporte de almacenamiento o distribución, se denomina un *agregado* si el copyright resultante de la compilación no se usa para limitar los derechos de los usuarios de la misma más allá de lo que los de los trabajos individuales permiten. Cuando el Documento se incluye en un agregado, esta Licencia no se aplica a otros trabajos del agregado que no sean en sí mismos derivados del Documento.

Si el requisito de la sección 3 sobre el Texto de Cubierta es aplicable a estas copias del Documento y el Documento es menor que la mitad del agregado entero, los Textos de Cubierta del Documento pueden colocarse en cubiertas que enmarquen solamente el Documento dentro del agregado, o el equivalente electrónico de las cubiertas si el documento está en forma electrónica. En caso contrario deben aparecer en cubiertas impresas enmarcando todo el agregado.

1.10 A.9. TRADUCCIÓN

La Traducción es considerada como un tipo de modificación, por lo que usted puede distribuir traducciones del Documento bajo los términos de la sección 4. El reemplazo las Secciones Invariantes con traducciones requiere permiso especial de los dueños de derecho de autor, pero usted puede añadir traducciones de algunas o todas las Secciones Invariantes a las versiones originales de las mismas. Puede incluir una traducción de esta Licencia, de todas las notas de licencia del documento, así como de las Limitaciones de Garantía, siempre que incluya también la versión en Inglés de esta Licencia y las versiones originales de las notas de licencia y Limitaciones de Garantía. En caso de desacuerdo entre la traducción y la versión original en Inglés de esta Licencia, la nota de licencia o la limitación de garantía, la versión original en Inglés prevalecerá.

Si una sección del Documento está Titulada *Agradecimientos*, *Dedicatorias* o *Historia* el requisito (sección 4) de Conservar su Título (Sección 1) requerirá, típicamente, cambiar su título.

1.11 A.10. TERMINACIÓN

Usted no puede copiar, modificar, sublicenciar o distribuir el Documento salvo por lo permitido expresamente por esta Licencia. Cualquier otro intento de copia, modificación, sublicenciamiento o distribución del Documento es nulo, y dará por terminados

automáticamente sus derechos bajo esa Licencia. Sin embargo, los terceros que hayan recibido copias, o derechos, de usted bajo esta Licencia no verán terminadas sus licencias, siempre que permanezcan en total conformidad con ella.

1.12 A.11. REVISIONES FUTURAS DE ESTA LICENCIA

De vez en cuando la Free Software Foundation puede publicar versiones nuevas y revisadas de la Licencia de Documentación Libre GNU. Tales versiones nuevas serán similares en espíritu a la presente versión, pero pueden diferir en detalles para solucionar nuevos problemas o intereses. Vea <http://www.gnu.org/copyleft/>.

Cada versión de la Licencia tiene un número de versión que la distingue. Si el Documento especifica que se aplica una versión numerada en particular de esta licencia o *cualquier versión posterior*, usted tiene la opción de seguir los términos y condiciones de la versión especificada o cualquiera posterior que haya sido publicada (no como borrador) por la Free Software Foundation. Si el Documento no especifica un número de versión de esta Licencia, puede escoger cualquier versión que haya sido publicada (no como borrador) por la Free Software Foundation.

1.13 A.12. ADENDA: Cómo usar esta Licencia en sus documentos

Para usar esta licencia en un documento que usted haya escrito, incluya una copia de la Licencia en el documento y ponga el siguiente copyright y nota de licencia justo después de la página de título:

Copyright (c) AÑO SU NOMBRE. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, Versión 1.2 o cualquier otra versión posterior publicada por la Free Software Foundation; sin Secciones Invariantes ni Textos de Cubierta Delantera ni Textos de Cubierta Trasera. Una copia de la licencia está incluida en la sección titulada *GNU Free Documentation License*.

Si tiene Secciones Invariantes, Textos de Cubierta Delantera y Textos de Cubierta Trasera, reemplace la frase *sin ... Trasera* por esto:

siendo las Secciones Invariantes LISTE SUS TÍTULOS, siendo los Textos de Cubierta Delantera LISTAR, y siendo sus Textos de Cubierta Trasera LISTAR.

Si tiene Secciones Invariantes sin Textos de Cubierta o cualquier otra combinación de los tres, mezcle ambas alternativas para adaptarse a la situación.

Si su documento contiene ejemplos de código de programa no triviales, recomendamos liberar estos ejemplos en paralelo bajo la licencia de software libre que usted elija, como la Licencia Pública General de GNU (*GNU General Public License*), para permitir su uso en software libre.

1.14 Notes

- [1] Ésta es la traducción del Copyright de la Licencia, no es el Copyright de esta traducción no autorizada.
- [2] La licencia original dice *publisher*, que es, estrictamente, quien publica, diferente de *editor*, que es más bien quien prepara un texto para publicar. En castellano *editor* se usa para ambas cosas.
- [3] En sentido estricto esta licencia parece exigir que los títulos sean exactamente *Acknowledgements*, *Dedications*, *Endorsements* e *History*, en inglés.

Resumen

El principal problema a la hora de desarrollar proyectos de desarrollo de software es el control de versiones del mismo, no solo cuando este es desarrollado por diferentes personas, como por ejemplo en una comunidad de programadores o como en el caso que nos ocupa personal de un grupo de trabajo de una empresa, sino también cuando son proyectos individuales o unipersonales debido a que cada vez las estructuraciones y la organización de dichos proyectos son más complejas. Para poder solventar el problema se recurre a las llamadas herramientas de control de versiones, las cuales facilitan principalmente el almacenamiento de los elementos a gestionar, la recuperación de cada uno de ellos y el registro histórico e identificación de cada una de las modificaciones realizadas en las sucesivas versiones del código del proyecto. Para empezar, vamos a hacer una breve presentación y comparativa de algunos de los sistemas de control de versiones, aunque este no sea el objeto de la tesis nos ayudara para poder hacer una buena elección del sistema que mejor se adapta a las necesidades actuales y futuras de la empresa. Este proyecto en concreto trata de la implementación de un SCV en el que la mayor dificultad será la gestión de los permisos los cuales estarán en el dominio de la empresa ya creado y que deberán ser gestionados por el gestor de proyectos de la empresa para poder dar de alta dichos usuarios a determinados proyectos, hacer que la versión sea la definitiva y bloquear la edición o modificación del mismo, además de las problemáticas técnicas de la propia instalación del SCV con sus servidor, elección del hardware, etc.

Tabla de contenidos

1.	Introducción	13
2.	Objetivos.....	14
2.1	Diagrama de GANTT	15
3.	Requisitos.....	17
4.	¿Por qué necesitamos un SCV?.....	19
5.	Objeto del análisis y breve contexto histórico	20
5.1	Estructuración del análisis	20
5.2	Información general	20
5.3	Información técnica.....	21
6.	Infraestructura	22
6.1	Disposición de la red de la empresa	22
6.2	Disposición de aplicaciones y sistemas	24
6.3	Seguridad	25
6.4	Integridad de los datos	26
6.5	Especificaciones del servidor.....	27
7.	Instalación del servidor y clientes	30
7.1	Elección del OS para el servidor.....	30
7.2	Creación de la máquina virtual	30
7.3	Instalación del OS del servidor (Debian 8.5).....	39
7.4	Elección del sistema de control de versiones elegido.	55
7.5	Instalación de subversión en el servidor SVN.....	56
7.6	Instalación y configuración del dominio y los usuarios para poder acceder desde internet.....	68
7.7	Instalación de los clientes en los PC de trabajo	68
7.8	Estructuración de proyectos en el servidor	70
7.9	Estructuración de las carpetas de proyecto en los PC clientes	71
8.	Primeras pruebas	72
8.1	Antes de comenzar a versionar código.....	72
8.2	CVI/Labwindows.....	74
8.3	LabView.....	89
9.	Usuarios y control de acceso	101
9.1	Breve descripción	101
9.2	Necesidades a cubrir para la gestión de usuarios.....	101
10.	Conclusiones.....	102

1. Introducción

En el inicio este proyecto vino motivado por dos razones, la primera es puramente de gestión del contenido y de proyectos software ya que actualmente todo el software, y cuando hablamos de software me refiero a software en general, ya sea de PC, PLC, cámaras de visión artificial, células robotizadas, servomotores, etc., está contenido en un servidor del tipo “NAS” es una estructuración de carpetas donde cada departamento tiene su espacio (departamento de software, control, mecánica, hidráulica, etc.), en el caso que nos ocupa, dejamos de lado los departamentos que no generas software, en el cual cada programador o Team Leader se encarga de versionar de forma manual cada una de las diferentes versiones de código, obviamente no hay un estándar definido para realizarlo y en ocasiones es un poco caótico el hecho de tener que recuperar y comparar dos versiones diferentes de código, por ello se decide implementar un SCV desde la propia empresa.

El segundo motivo es personal, puesto que trabajo directamente con el desarrollo de software y al haber realizado diversas practicas durante el master utilizando diferentes sistemas de control de versiones, en el momento que se propuso la posible implantación de un SCV de la empresa pedí participar activamente del proyecto interno de la empresa por tal de intentar aplicar algunos de los conocimientos adquiridos durante el master, como puede ser la gestión y administración de sistemas LINUX, aplicación de una solución de software libre en la empresa en la que en un principio solo se previó solo la implantación de un servidor de SCV simples y en la cual se está pensando actualmente en implementar algunos código en HTML+CSC para poder hacer las operaciones cuando los programadores están desplazados al exterior.

2. Objetivos

Debido a la problemática agregada a la empresa en los últimos años que se ve obligada manejar un gran volumen de software y diferentes versiones de los mismos, el siguiente proyecto tiene como objeto la implementación de un sistema de control de versiones de software libre, puesto que debido al crecimiento de la empresa y a que se dedica a realizar maquinaria industrial automatizada a medida requiere un sistema de control de versiones de software para poder gestionar los cambios realizados de una manera más ágil a la actual. Asimismo, el sistema de poder gestionar software no solo de PC en diferentes lenguajes de programación sino también de PLC, Servomotores, Sistemas de Visión Artificial, etc. Puesto que la empresa tiene subcontratas y a menudo deben de reportar los cambios realizados en proyectos nuevos como en modificaciones o mejoras de los ya existentes no solo se requiere que el sistema sea accesible solo a personal de la empresa sino también a mediante conexiones seguras y autenticadas permitir a dichas subcontratas el acceso al sistema.

En un primer momento se le dará control total solo al Project Leader quien podrá dar permisos de lectura y/o edición a las personas correspondientes que participen en los diversos proyectos, asimismo se les dará acceso solo de lectura al personal restante de la empresa.

Cabe destacar que al ser un proyecto aplicado a una empresa real y con una actividad económica y debido a que yo no formo parte del departamento de IT, sino que simplemente colabora con ellos debido a que soy desarrollador de software dentro de la empresa cabe la posibilidad de que debido a las fluctuantes cargas de trabajo y otros factores el plan inicial se vea alterado y modificado debido a la disponibilidad del personal.

A continuación, se presentará en un gráfico de GANTT la planificación inicial con las diferentes fases del proyecto y las dependencias del mismo ya que se trata de una implantación hay cosas que puede hacerse en paralelo o en el tiempo que sea posible, pero hay otras que deben cumplirse si o si en las fechas previstas ya que de lo contrario retrasaran las fases posteriores que de ellas dependan y el proyecto se verá afectado por un retraso que muy probablemente no pueda absorberse.

3.1 Diagrama de GANTT

El siguiente diagrama muestra las diferentes fases del proyecto definidas, así como las fechas en las que deben de estar finalizadas cada una de las fases.

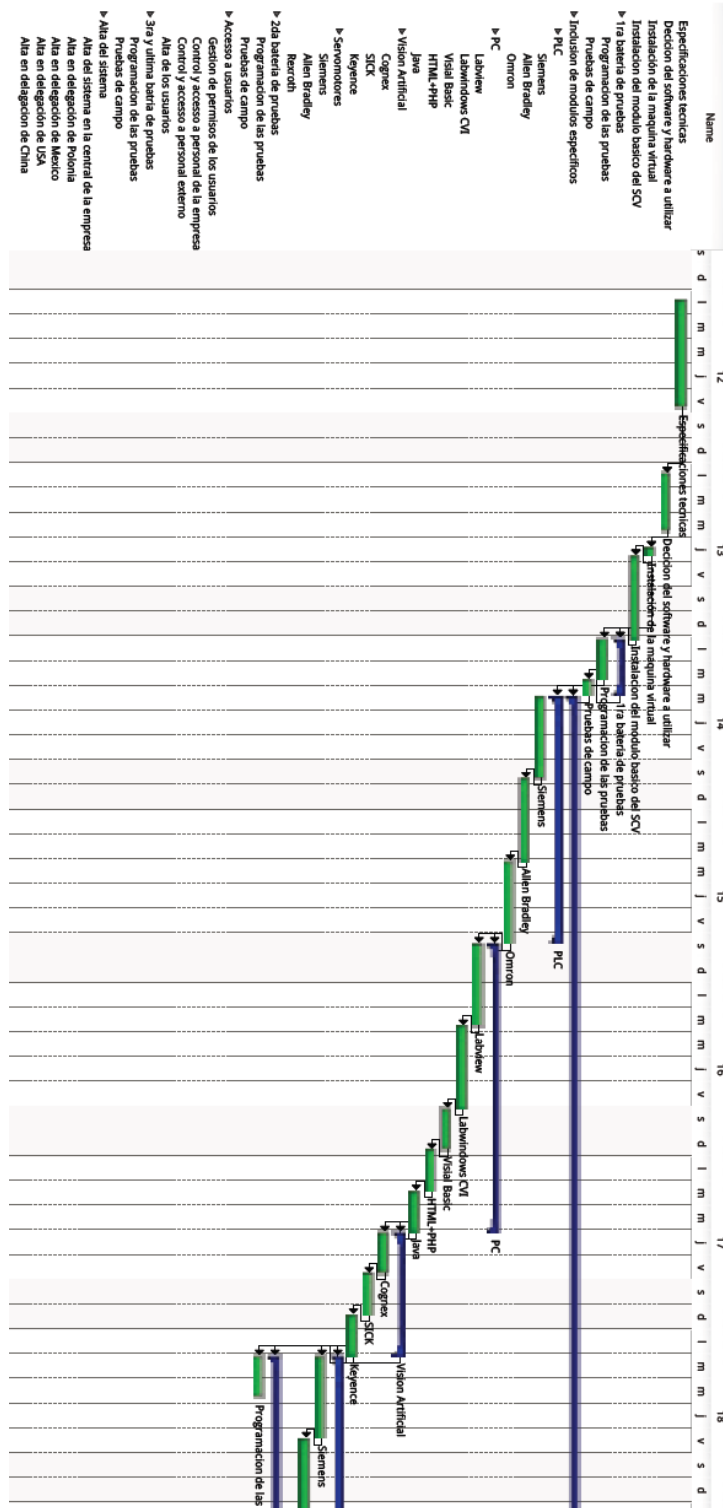


Ilustración 1 Diagrama de GANTT V1-1

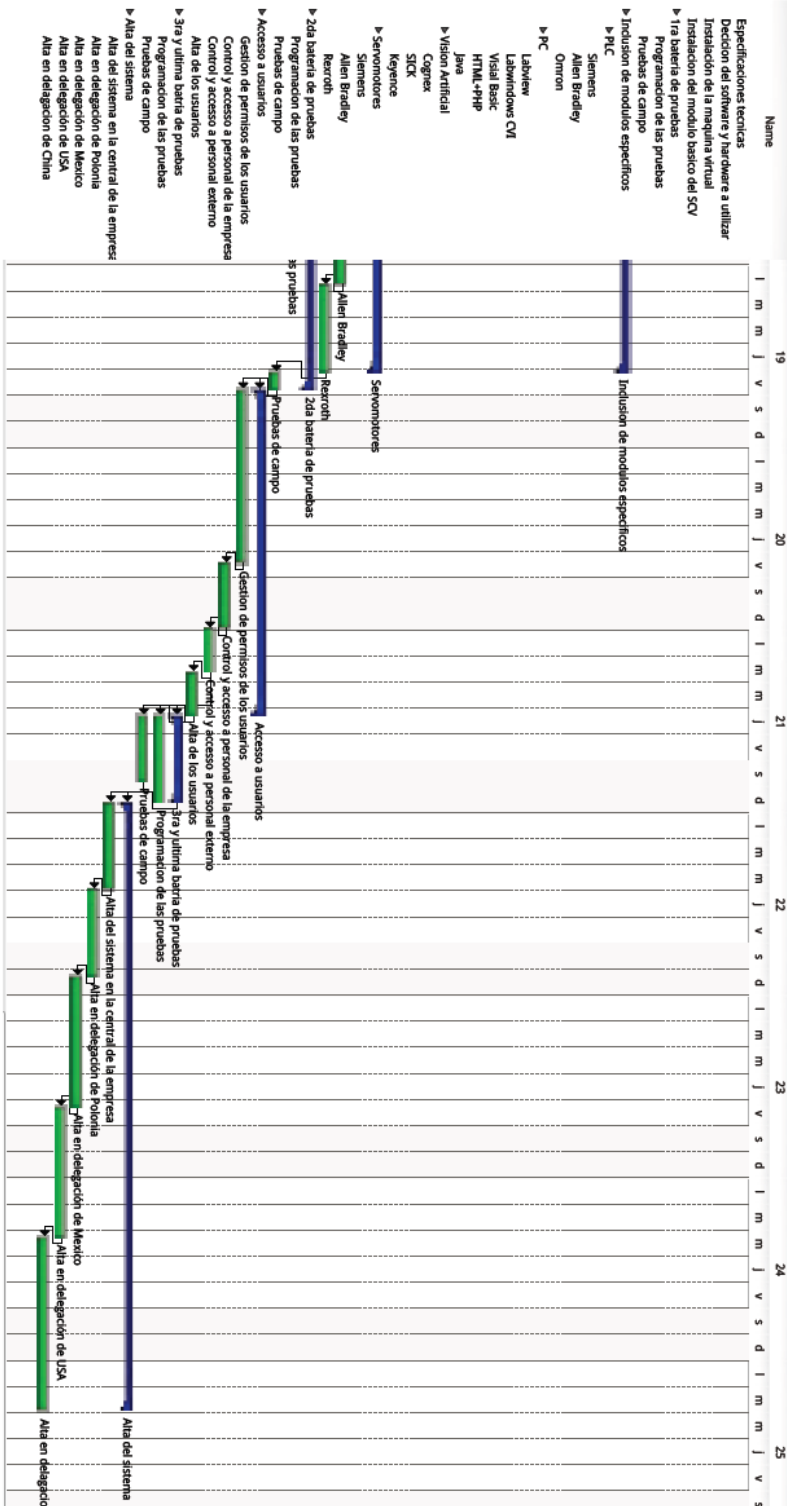


Ilustración 2 diagrama de GANTT V1-2

3. Requisitos

Debido al entorno meramente industrial que tiene la empresa y a dedicar su mayor actividad a la fabricación de maquinaria industrial a medida, aun teniendo un departamento de IT dedicado a ello, esto hace que el sistema de control de versiones debe ser apto para poder versionar diferentes tipos de código escritos en diferentes lenguajes y de diferentes plataformas y marcas, por ello debemos hacer que el sistema sea lo más amplio posible y este debe poder controlar versiones de:

- Software para PC.
 - ↳ CVI (ANSI C).
 - ↳ LabVIEW.
 - ↳ Visual Basic.
 - ↳ HTML + PHP.
 - ↳ Java.

- Software de PLC (Programable Logic Controllers)
 - ↳ Siemens.
 - ↳ Allen Bradley.
 - ↳ Omron.

- Software de Servomotores.
 - ↳ Siemens (Starter).
 - ↳ Allen Bradley.
 - ↳ Rexroth (Indradrive).

- Apto para controlar versiones de dispositivos de visión artificial.
 - ↳ SICK.
 - ↳ Cognex.
 - ↳ Keyence.

Además, el hecho de tener una carga fluctuante de trabajo que no es predecible a simple vista y que obliga a tener personal externo en determinados periodos de tiempo y a tener gente desplazada en puestas en marchas y asistencias a cliente se definen tres zonas o ámbitos en los que el SCV deberá ser funcional, cada uno con su peculiaridad. Estos son:

- Para trabajadores de la empresa conectados a la red local.
 - ↳ En la central de la empresa ubicada en Sant Fruitos de Bagues
 - ↳ En las delegaciones que la empresa tiene en Polonia, México,
 - ↳ USA, China

- Para trabajadores de la empresa conectados en otras redes (desplazados al exterior)

- Para personal externo a la empresa, en este caso deberá crearse un usuario y un password temporal para cada usuario externo. Debido a que el proyecto depende de varias áreas dentro de la empresa las tareas a realizar deben involucrar a todas ellas por ello, junto con los

responsables de cada una de las áreas de desarrollo de software se deberá realizar una especificación de los requisitos que el sistema de control de versiones debe tener, además de realizar el seguimiento y la integración del sistema, así como asegurar la robustez y seguridad del mismo mediante pruebas de funcionalidad, control de error, gestión de incidencias automatizada, y cualquier otra funcionalidad que se prevea conveniente en el transcurso de la implementación.

4. ¿Por qué necesitamos un SCV?

Tradicionalmente el desarrollo de software se relegaba a un pequeño grupo de programadores que tenían una filosofía común, colaboraban en la realización del trabajo y gestionaban las versiones de forma manual, con el tiempo de vida de un proyecto esta gestión se volvía ineficiente y se transformaba en una tarea ardua donde se invertía mucho tiempo en encontrar las últimas versiones de un archivo o intentando descubrir si la versión de uno de ellos era la última o si había sido modificada.

Actualmente el desarrollo de software se ha convertido en algo mucho más amplio, los productos generados son altamente susceptibles a cambios, tampoco se reducen a proyectos individuales ni a pequeñas comunidades o grupos de desarrolladores, por ello surge la necesidad de que muchos programadores tengan que acceder al mismo código de forma paralela realizando sus propias modificaciones. Intentar gestionar esto manualmente se traduciría a una pérdida de tiempo muy valiosa ya que deberían gestionarse además las notificaciones y hacerlo por conversaciones entre los miembros del equipo es algo tedioso y muchas veces ineficaz. Los sistemas de control de versiones se encargan de almacenar el historial de las modificaciones que van sufriendo los diferentes archivos de un proyecto, es decir, de gestionar los sucesivos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo manteniendo de manera estructurada los avances, retrocesos y modificaciones del software mientras está siendo desarrollado.

Las ventajas de utilizar un sistema de control de versiones son innegables, no solo son eficaces en la gestión del trabajo concurrente de varios desarrolladores permitiendo manejar las diferentes versiones del proyecto, sino que permiten seguir la evolución del mismo. Cada ínfimo cambio estará registrado en el sistema con todos los datos que se puedan llegar a necesitar (qué, quién, por qué, cuando, etc.). Favorecen la colaboración entre los integrantes del equipo facilitando la integración de sus versiones, encontrando cambios potencialmente incompatibles y ayudando a resolver tales conflictos. Sin dejar de lado la posibilidad de solucionar los errores introducidos con mayor facilidad volviendo a una versión anterior con solo elegir la deseada. No obstante, lo anterior y contando con todas las ventajas que ofrece un sistema de control de versiones no se debe olvidar que por el momento ninguna herramienta es capaz de sustituir la comunicación entre los desarrolladores, y en último caso, de existir cambios incompatibles sobre la misma información, será necesaria la intervención humana para tomar la decisión definitiva.

5. Objeto del análisis y breve contexto histórico

Para poder proceder a un análisis correcto y coherente debemos conocer los objetivos del proyecto, así como un poco de contexto histórico de la evolución del software y la necesidad que surge para necesitar utilizar control de versiones.

El uso de herramientas de control de versionado es esencial para una gestión del cambio eficaz. En la actualidad, la gestión de versiones se apoya siempre en herramientas CASE que administran el almacenamiento de cada versión del sistema y controlan el acceso a los diferentes componentes del mismo. A pesar de que las herramientas difieren obviamente en funcionalidades de interfaz, la base de todas estas herramientas de soporte es el control de la gestión de versiones, por ello a partir de los años 70 se empezaron a desarrollar herramientas con este fin, por este motivo el primer paso antes de ponernos a implementar un sistema de control de versiones es hacer un breve análisis de los que se encuentran actualmente en el mercado para poder elegir el más adecuado para los requisitos de la empresa.

6.1 Estructuración del análisis

Se analizarán diversas características de los SCV para poder utilizar la información de la mejor manera posible, entre ellas se analizarán, la funcionalidad específica de cada una de las herramientas se realizará una comparación de las características técnicas de cada una de ellas y para concluir dentro de la información técnica se realizará una comparativa de las interfaces de que disponen.

6.2 Información general

En este caso se realiza el estudio de 4 Herramientas de software libre para saber cuál es la que mejor se ajusta a nuestras necesidades.

A continuación, se muestra una tabla comparativa de las 4 herramientas previamente seleccionada para poder hacernos una idea del estado de cada una

Tabla 1 información general de SCV + Clientes

Herramienta	Licencia	Versión Primera	Versión Actual	Estado
CVS	GNU	1990	11.11.23 May. 2008	Estable
Subversion	Apache Licence	2000	1.9.3 Dic. 2015	Estable/Activa
Tortoise	GNU	2002	1.9.3 Dic. 2015	Estable/Activa
CVSNT	GPL	1998	2.5.05 Mar. 2010	Estable

6.3 Información técnica

La siguiente tabla presenta una comparativa de los parámetros representativos técnicamente hablando,

1. Tipo de repositorio: centralizado si existe un único repositorio central para todo el proyecto en un servidor dedicado o distribuido si cada usuario tiene su propio repositorio del cual es administrador.
2. Modelo de concurrencia: lock-modify-unlock (LMU) si el sistema sólo permite a una persona modificar un archivo al mismo tiempo y se utiliza la técnica del bloqueo para conseguirlo o copy-modify-merge (CMM) si el sistema permite a múltiples usuarios trabajar de manera simultánea sobre las copias locales que tienen del repositorio.
3. Modelo de almacenamiento: historia del cambio (changeset) si almacena únicamente el conjunto de los nuevos cambios realizados o snapshot si almacena una instantánea del árbol de directorios antes y después del cambio
4. Alcance del cambio: árbol o ficheros individuales
5. Identificación de las revisiones: criptográficas SHA-1 hashes números, pseudoaleatorio o funciones hash criptográficas SHA-1 hashes.

Tabla 2 Comparativa de gestión de contenido

Herramienta	Tipo de repositorio	Modelo de concurrencia	Historial de cambio	Alcance del cambio	Identificación de revisiones
CVS	Centralizado	LMU / CMM	Changegest	Fichero	Números
Rapid Subversion	Centralizado	LMU / CMM	Changegest/ Snapshot	Árbol	Números
TortoiseSVN	Centralizado	LMU / CMM	Changegest/ Snapshot	Árbol	Números
CVSNT	Centralizado	LMU / CMM	Changegest/ Snapshot	Árbol	Números

Después de realizar diversas pruebas con diferentes clientes de gestión de versiones se decide implementar TortoiseSVN debido a su integra independencia al lenguaje utilizado y a su facilidad de uso, ya que queda integrado como un Shell directamente y se pueden realizar muchas de las acciones directamente haciendo clic derecho sobre las carpetas y/o ficheros deseados, algo muy útil si tenemos en cuenta la variedad de códigos, dispositivos y lenguajes utilizados y dado que cada uno de estos lenguajes y dispositivos utilizan un software determinado de parametrización/programación y no siempre es posible integrar en el propio IDE un sistema de control de versiones utilizándolo como extensión Shell de Windows es la manera más óptima de trabajar en este caso.

6. Infraestructura

7.1 Disposición de la red de la empresa

La empresa consta de 5 naves Industriales en España (maproheadquarters) y dispone de una en Méjico y otra en Polonia donde si dispone de red interna intranet conectada.

En esta se puede ver la alta disponibilidad del switch central lo que nos permite asegurar el funcionamiento de la red en caso de fallo de algún hardware. Además, al tener la red segmentada en subredes, tanto en las instalaciones locales como en las delegaciones nos permite reducir el impacto de algún tipo de fallo debido al hardware, hay que decir que todos los switches de la red están replicados para poder tener concurrencia en caso de daño/fallo de alguno de ellos.

En el esquema también puede verse que se dispone de firewalls los cuales nos protegen los servidores que deben ser accesibles desde el exterior, aunque la instalación de dichos no está terminada y se prevé terminarla durante el transcurso del año.

Cabe destacar de que se dispone de aceleradores WLAN de la marca Riverbed, que permiten aumentar la velocidad de transferencia de datos y comunicaciones entre las delegaciones, ya que realizan un trabajo de compresión y descompresión que mejoran las comunicaciones cuando estas sales a Internet mediante los túneles IPsec.

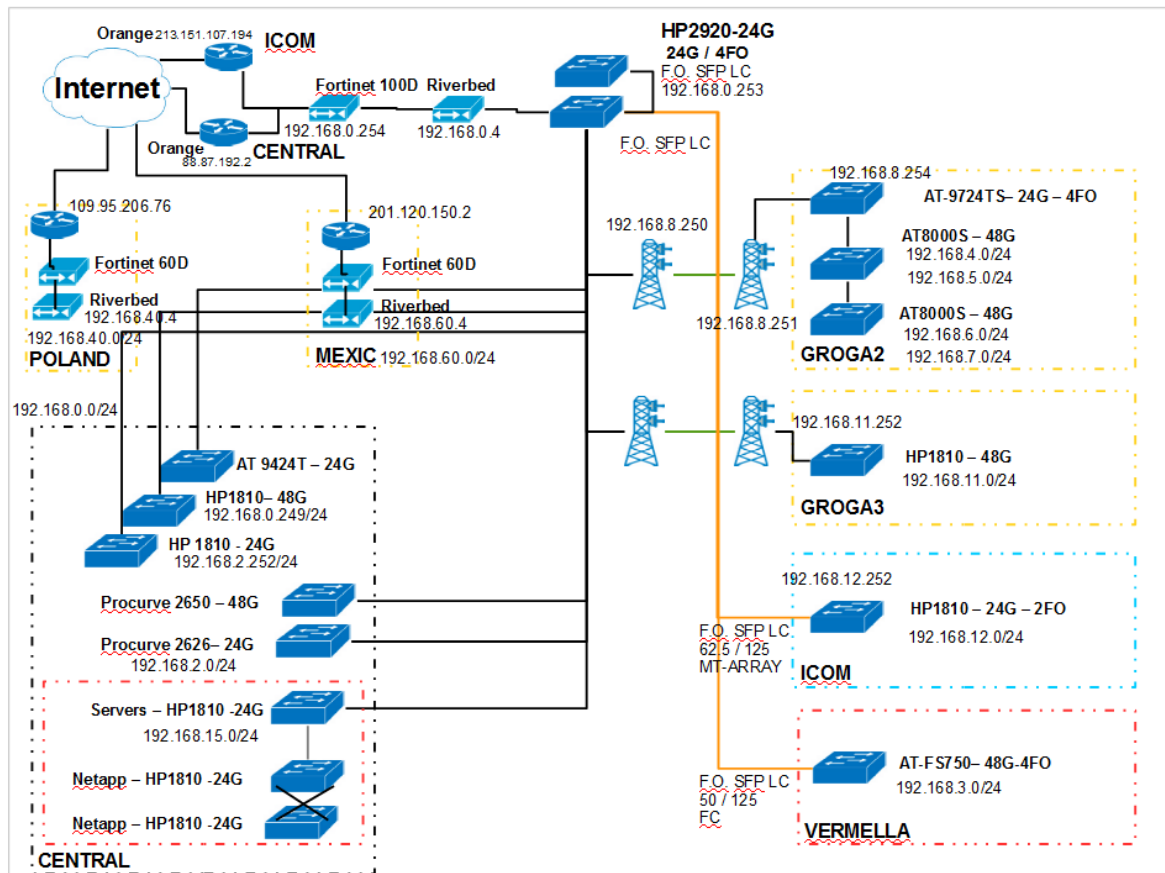


Ilustración 3 Esquema de red corporativo de la empresa

Para entrar en más detalle presentamos un pequeño esquema del Switch Core de Mapro. Es un equipo de Layer 3, lo que nos permite es dividir la red en diferentes WLAN y tenerla sí segmentada para garantizar la calidad de la misma. Como ya se comenta, en caso de parada, el segundo Switch toma el control gracias a la implantación de un segundo equipo con las mismas características y a la configuración de la alta disponibilidad de estos.

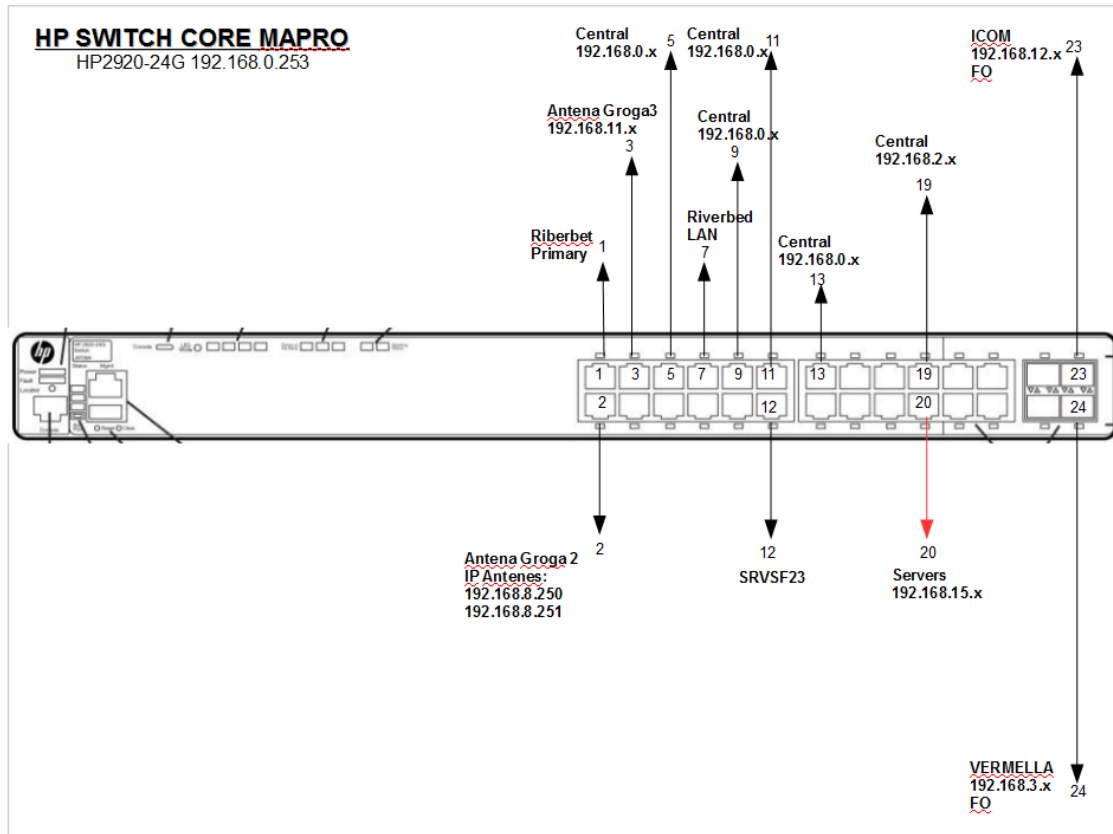


Ilustración 4 Grafica Switch central localizado en la nave de la calle la Coma 29ª.

7.2 Disposición de aplicaciones y sistemas

Todos nuestros sistemas, tanto de ficheros como de servidores, están basados en dos cabinas NetApp FAS2220 de 12TB, que las tenemos divididas en 9TB para todos los ficheros de la compañía, y 3TB para el sistema de virtualización de los servidores.

Disponemos de otra cabina exactamente del mismo modelo y capacidad configurada en alta disponibilidad. Esto nos garantiza, aunque cada cabina dispone de sus controladoras también doblada, que, en caso de falla de Hardware, esta segunda cabían asuma el control y permita a la compañía continuar con su actividad.

La estructura de virtualización esta basa en VMWare vSphere 5.5 Essential Plus, que nos proporciona la gestión de los servidores virtualidades y la tolerancia de fallos ante alguna caída por fallo de hardware de los servidores que están dentro de esta infraestructura.

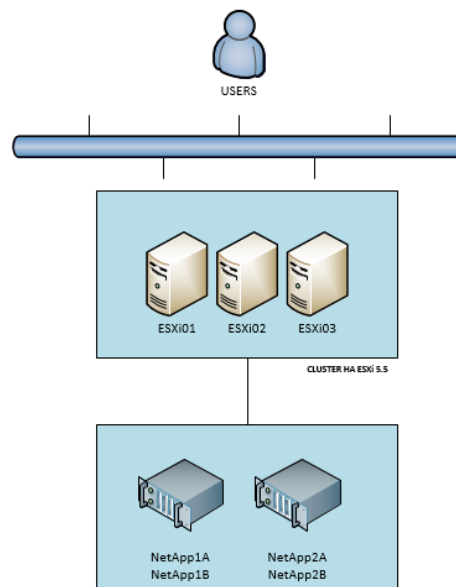


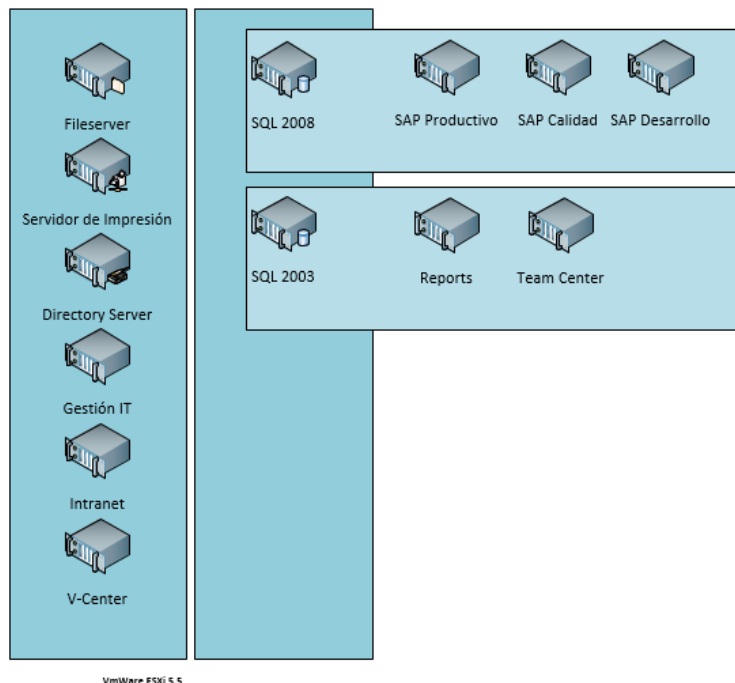
Ilustración 5 disposición de los servidores

La estructura consta de tres servidores y dos cabinas tal como se ve en el esquema. El sistema está pensado y dimensionado para que, si un servidor falla, los otros dos asumen todo su trabajo de forma automática.

Si dividimos el software de la compañía en dos grupos, uno sería el sistema Windows que es la base de todo el sistema, y el otro, el software que nos ayuda en la actividad de la compañía, como es SAP R3 6.0, Team center, Office365 (ofimática y correo) y web Reports desarrollada en php por Mapro.

EL sistema Windows consta de un dominio basado en Windows Server 2012 R2 Standard Edition y el software de la compañía está basado en Microsoft SQL Server 2008 (SAP R3 6.0) y SQL 2003 (Reports Web y TeamCenter).

En el siguiente esquema explicamos que esta estructura los servidores relacionados con las bases de datos que utilizan, con el fin de por ver la relación que pueden tener entre ellos.



Windows Server 2012:

- Fileserver
- Printers
- Active Directory
- Gestion IT
- Intranet
- V-Center

SQL 2008:

- SAP (DEV, QUA, PRD)

SQL 2003:

- Reports
- TeamCenter

Ilustración 6 Esquema de servidores relacionadas con sus DB

7.3 Seguridad

Se dispone de dos tipos de seguridad bien definidas y diferentes, una de ellas es la lógica y típica, se trata de la seguridad digital y de mantener la integridad y fiabilidad de los datos, la otra es la física puesto que la sala de servidores dispone de control de acceso físico.

I. Digital, Firewall

Los firewalls con los que la compañía trabaja son de la marca Fortinet, concretamente en la delegación central usamos el modelo 100D (firewall central) y en las delegaciones tenemos los modelos 60D. Estos dispositivos son los que nos facilitan la conexión entre las delegaciones con túneles IPsec, y de esta manera las delegaciones tienen conexión total con todos los servicios IT que la compañía tiene.

Estamos en proceso de configuración de la alta disponibilidad con un segundo firewall en cada delegación, para que, en caso de parada, el segundo pueda asumir todo el trabajo.

El firewall nos permite tener control de los puertos que tenemos públicos en internet y el consiguiente filtro de estos, por lo tanto, estos no hacen de sistema IDS ya que el log que nos facilita nos permite realizar un análisis para poder detectar posibles ataques.

De sistema Antispam, al tener contratado el servicio de correo con el pack Office365 que nos incluye un sistema de Antispam y Antivirus. Para más seguridad en este sentido, el antivirus Symantec que el Grupo Carbures ha contratado también tiene funciones de Antispam, pero a nivel de usuario en cada máquina por separado, a parte

de las funciones de Antivirus que también realiza.

En este momento no hemos aplicado filtro de navegación web, pero los firewalls se pueden ocupar de esta tarea.

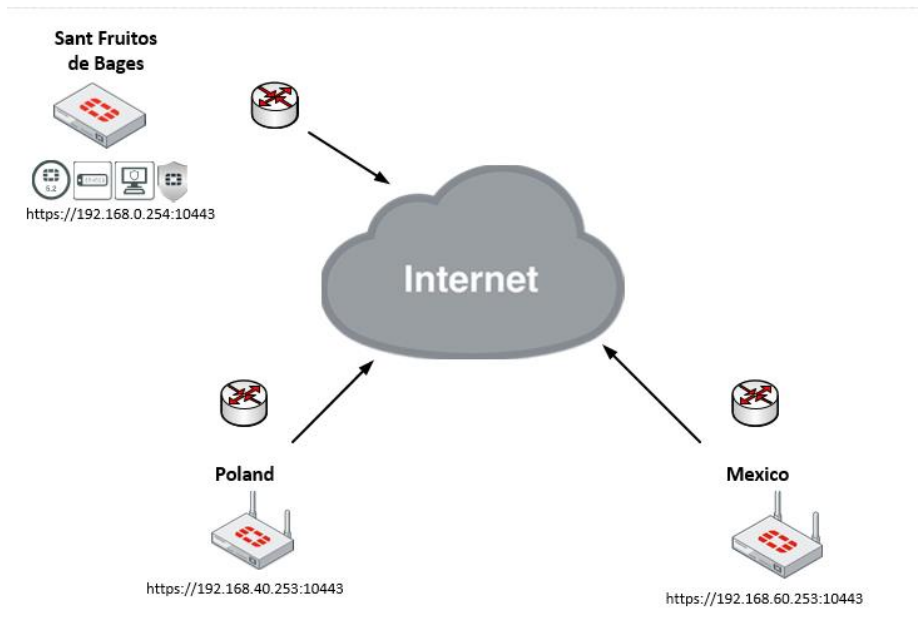


Ilustración 7 Esquema Firewalls entre delaciones y la central.

II. Física

La seguridad física del CPD está controlada a nuestro sistema de control de acceso biométrico con huella dactilar de la marca Supreme. Este sistema permite autenticación con huella dactilar, tarjeta magnética y/o con código numérico. En caso de fallo del sistema biométrico la puesta de entrada dispone de una cerradura con llave, de la que tiene copia todos los integrantes del departamento de IT y gerencia.

7.4 Integridad de los datos

Está implantada una política de copias de seguridad de los datos de carácter personal junto con el resto de datos (ingeniería, servidores, etc....) de la empresa, en base al programa BackupExec 12 de Symantec.

El sistema se basa en un robot de copias LTO5 de 1.5TB (3TB comprimidos) HP con capacidad de 8 cintas. El robot está conectado directamente a la librería de discos NetApp.

La política es incremental, haciéndose copias diarias, semanales y mensuales.

- Se establecen 3 meses de retención para las copias mensuales (copia completa).
- 3 semanas para las copias semanales (copia completa).
- 2 semanas para las diarias (copia incremental).

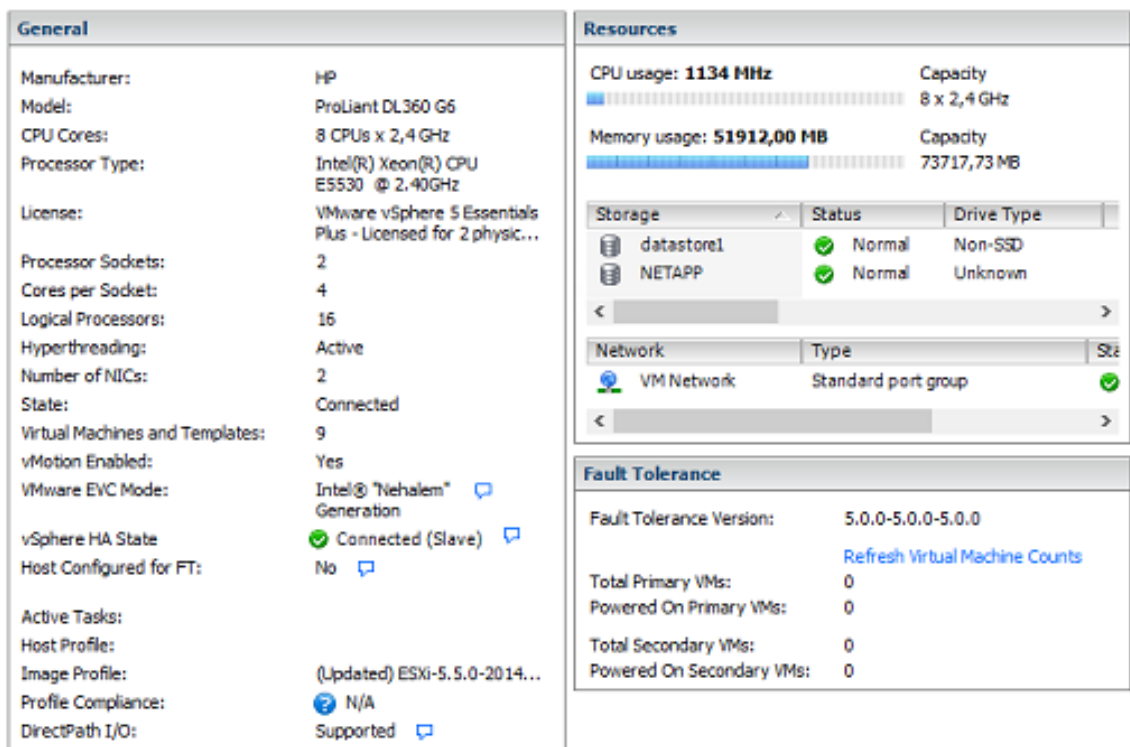
Esto implica que en una operación de copia incremental sólo son volcados aquellos archivos que hayan sufrido cambios desde la última copia de seguridad del mismo nivel o de un nivel superior.

Existen cintas para las copias diarias y semanales que se reutilizan mensualmente, guardándose la copia mensual de los últimos, como mínimo, 3 meses.

7.5 Especificaciones del servidor

I. Hardware

- Marca: HP
- Modelo: ProLiant DL360G6
- CPU: 8CPU a 2.4GHz Intel® Xeon® E5530



The screenshot displays the configuration details for a VMware host. It is divided into three main sections: General, Resources, and Fault Tolerance.

General:

- Manufacturer: HP
- Model: ProLiant DL360 G6
- CPU Cores: 8 CPUs x 2,4 GHz
- Processor Type: Intel(R) Xeon(R) CPU E5530 @ 2.40GHz
- License: VMware vSphere 5 Essentials Plus - Licensed for 2 physic...
- Processor Sockets: 2
- Cores per Socket: 4
- Logical Processors: 16
- Hyperthreading: Active
- Number of NICs: 2
- State: Connected
- Virtual Machines and Templates: 9
- vMotion Enabled: Yes
- VMware EVC Mode: Intel® "Nehalem" Generation
- vSphere HA State: Connected (Slave)
- Host Configured for FT: No
- Active Tasks:
- Host Profile:
- Image Profile: (Updated) ESXi-5.5.0-2014...
- Profile Compliance: N/A
- DirectPath I/O: Supported

Resources:

- CPU usage: 1134 MHz (Capacity: 8 x 2,4 GHz)
- Memory usage: 51912,00 MB (Capacity: 73717,73 MB)
- Storage:

Storage	Status	Drive Type
datastore1	Normal	Non-SSD
NETAPP	Normal	Unknown
- Network:

Network	Type	Status
VM Network	Standard port group	Connected

Fault Tolerance:

- Fault Tolerance Version: 5.0.0-5.0.0-5.0.0
- Total Primary VMs: 0
- Powered On Primary VMs: 0
- Total Secondary VMs: 0
- Powered On Secondary VMs: 0

Ilustración 8 recursos del servidor donde va a ir hospedado la VM de subversion

II. Virtualización.

Para poder entender un poco mejor por qué se virtualizan los servidores (debido a que nosotros trabajaremos con un servidor virtual) se hace una breve descripción a lo que es la virtualización.

En un sentido amplio y general cuando hablamos de virtualización nos referimos a la virtualización de servidores, lo significa particional un servidor físico en varios servidores virtuales. Cada una de estas particiones en forma de máquina virtual puede interactuar de forma independiente con otros dispositivos, aplicaciones, datos, usuarios, etc. como si se tratara de un recurso físico independiente.

Estas diferentes máquinas virtuales pueden ejecutar diferentes sistemas operativos y múltiples aplicaciones cada una utilizando un solo equipo físico. Esto es debido y posible ya que cada máquina virtual está aislada y trabaja de forma independiente a el resto.

Para poder utilizar diferentes máquinas virtuales en un mismo equipo este debe de prepararse previamente con un software dedicado a ello, este software se denomina hyperservidor o administrador de virtualización este software trabaja entre el hardware y el sistema operativo, separando el sistema operativo y las aplicaciones de hardware. El hyperservidor se encarga de asignar la cantidad de acceso que los sistemas operativos y aplicaciones tienen al procesador físico, memoria física, disco duro y demás recursos.

El anterior es uno de los múltiples usos de la virtualización y es el que nos interesa explicar ya que es el que utilizaremos, aunque vale la pena mencionar que además la virtualización puede utilizarse para:

- Virtualización de redes
- Virtualización de aplicaciones
- Virtualización de escritorios

para ser un poco más objetivos para decantarnos por si virtualizamos el servidor o bien lo hacemos con un servidor físico, debemos sopesar los pros y contras de virtualizar, para ello se realiza una tabla en la que se definen las ventajas y desventajas de virtualizar un servidor, de manera general.

En la siguiente tabla se pueden observar algunas de las ventajas e inconvenientes más relevantes de la virtualización

Tabla 3 ventajas e inconvenientes de la virtualización

Ventajas de virtualizar un servidor	Desventajas de virtualizar un servidor
Disminuye el número de servidores físicos esto repercute en la disminución de costes y mantenimiento del hardware	A mayor maquinas a virtualizar mas recursos necesarios en el hardware físico
Aumenta la eficiencia de la utilización de expansión en los centros de datos	Ocasionalmente hay incompatibilidad con el hardware virtualizado
Al tener sistemas diferenciados, independientes, en cada servidor virtual, evita que una aplicación impacte a otras en el momento de realizar mejoras o cambios o si simplemente falla	Dificultad elevada de configuración de algunos de los recursos en la máquina virtual izada, ya sea por entendimiento o por falta de opción en el hardware utilizado para virtualizar
Si se desarrolla una norma de construcción de servidores virtuales esto acelera y facilita la creación de nuevos servidores virtuales	
Pueden desplegarse diversas tecnologías con una sola plataforma hardware, como pueden ser Windows server 2008, Windows server 2012, Windows server 10, Servidores Linux debian o Ubuntu, Cent OS, CloudLinux, etc.	

7. Instalación del servidor y clientes

En este apartado se describirá como crear una máquina virtual, instalar el servidor y como se instalará subversión en el servidor y en los clientes con las diversas opciones de configuración

8.1 Elección del OS para el servidor

Lo primero que se debe tener en cuenta antes de crear un servidor virtual es que sistema operativo correrá bajo el SCV, puesto que es importante para poder definir los recursos de hardware físico asignados a la creación del servidor. Puesto que este como ya se ha descrito se instalará utilizando un hyperservidor/administrador de virtualización llamado *VMWare vSphere 5.5 Essential Plus*, en el que se creará una máquina virtual con los requisitos necesarios para poder hospedar un SCV con subversión en el que se instalará Debian server 8.5 sin interface gráfica para reducir consumo de recursos innecesarios. Se elige esta opción debido a que es una LTS y a la estabilidad que Debian proporcionara a nuestro sistema, además de que el coste en recursos no es muy elevado.

8.2 Creación de la máquina virtual

- I. Elección de la localización de la máquina virtual: en este caso se instalará el OS en el servidor que más recursos libres disponga.

The screenshot displays the vCenter vSphere Client interface. The left sidebar shows a tree view of the vCenter environment, including a cluster named 'MAPRO' and various servers. The main pane shows the configuration for a specific ESXi host (192.168.15.1). The 'General' tab provides details about the hardware and software, such as the manufacturer (HP), model (ProLiant DL360 G6), and processor information. The 'Resources' tab shows the current usage of CPU and memory, as well as the configuration of storage and network resources. The 'Fault Tolerance' section indicates that the host is configured for fault tolerance.

Ilustración 9 recursos del servidor a utilizar

- II. Upload del OS a la cabina datastore: para poder instalar el OS el sistema requiere que carguemos la imagen en el datastore para poder acceder y hacer una instalación via network.

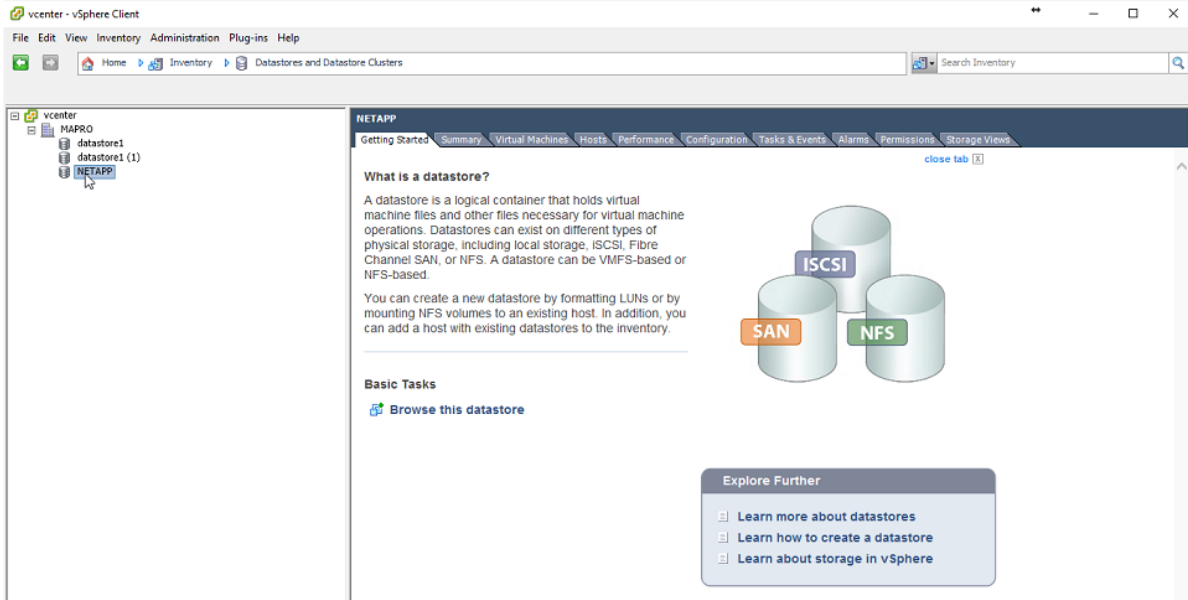


Ilustración 10 subiendo la iso al datastore

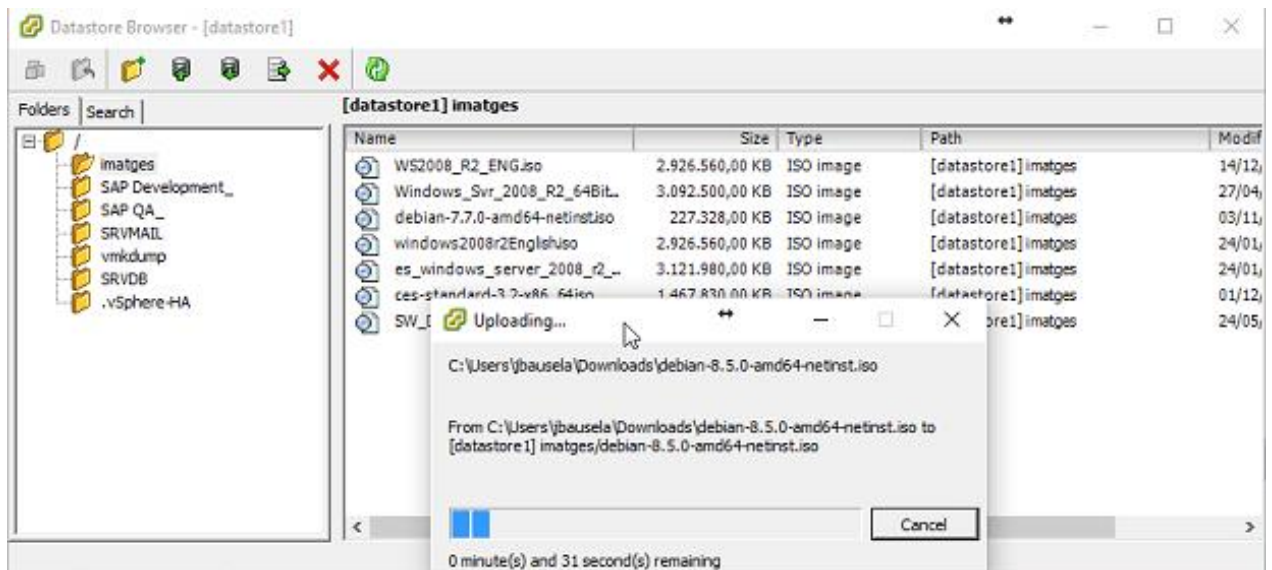


Ilustración 11 progreso de la carga de la iso

- III. Creación de la VM: a partir de que tenemos el OS en el datastore podemos proceder a la creación de la VM y a su posterior instalación.
Para empezar, seleccionamos el servidor donde ira instalado.



Ilustración 12 creando una VM nueva

Después desde el submenú que aparece al pulsar el botón derecho del mouse elegimos la opción de agregar una nueva máquina virtual.

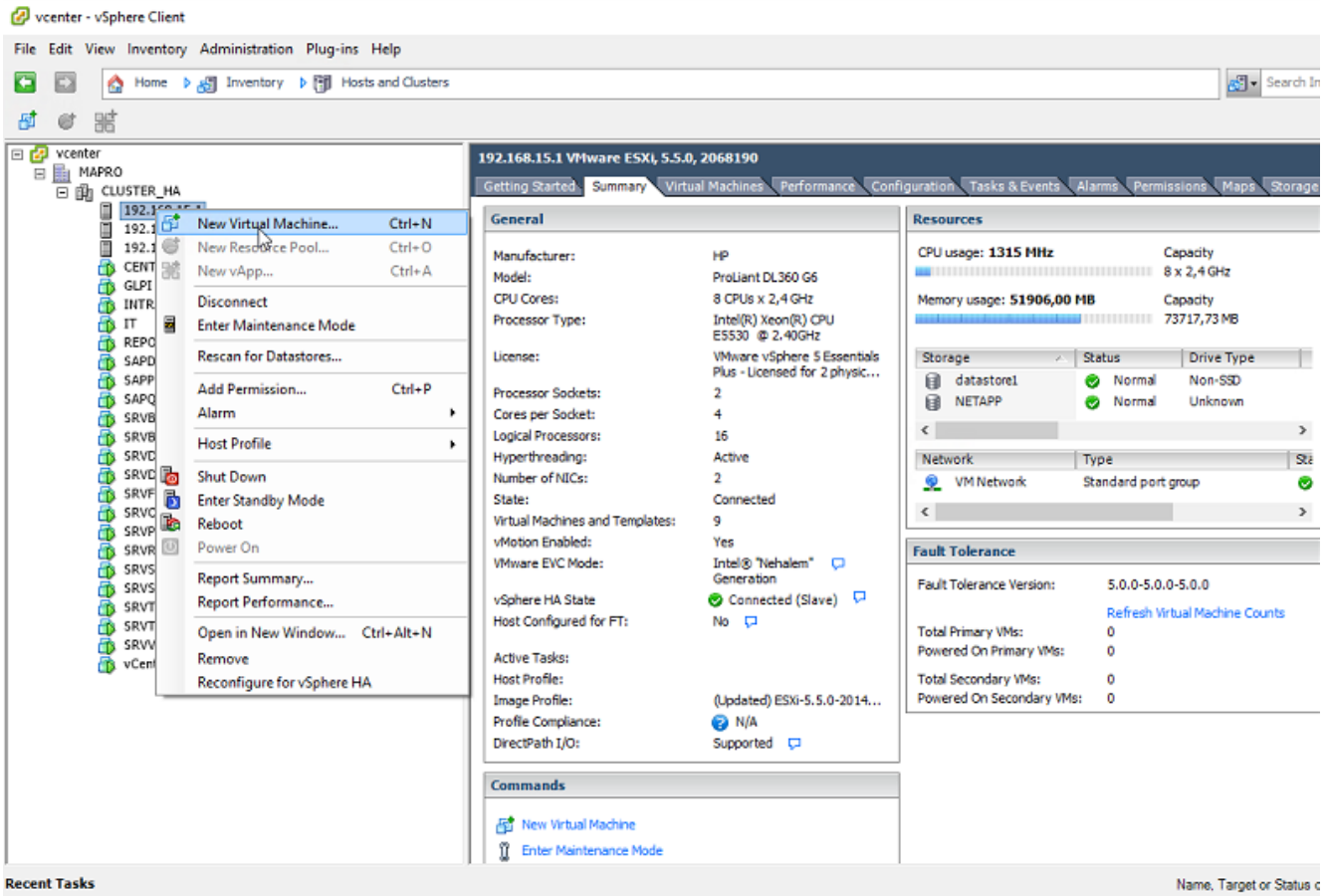


Ilustración 13 New VM

Para continuar elegimos las opciones de creación de la VM.
Para empezar, seleccionamos la creación de una VM típica

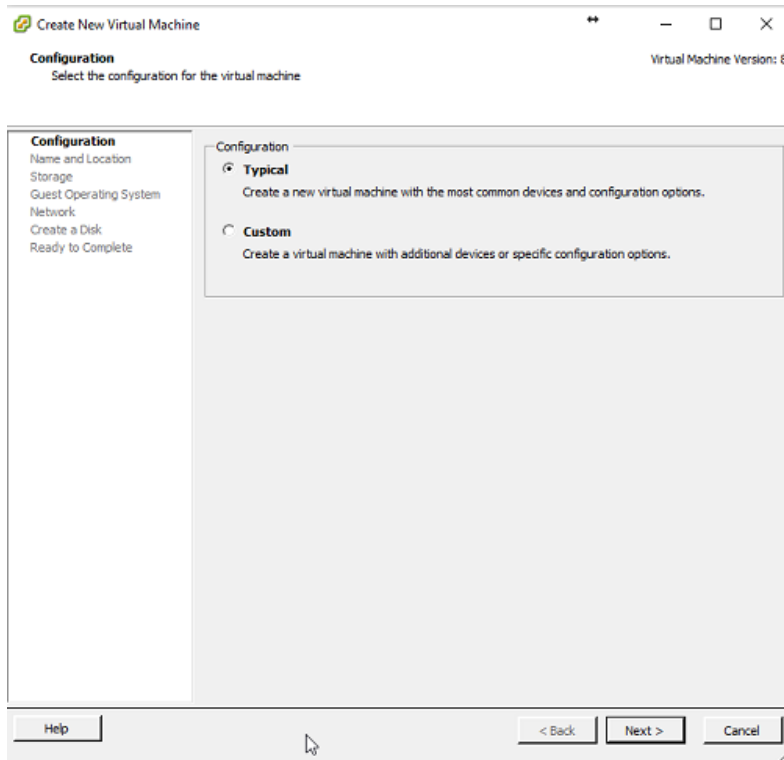


Ilustración 14 primeros pasos para la creación de una VM

Le damos un nombre descriptivo para poder identificarlo, en este caso SRVSVN.

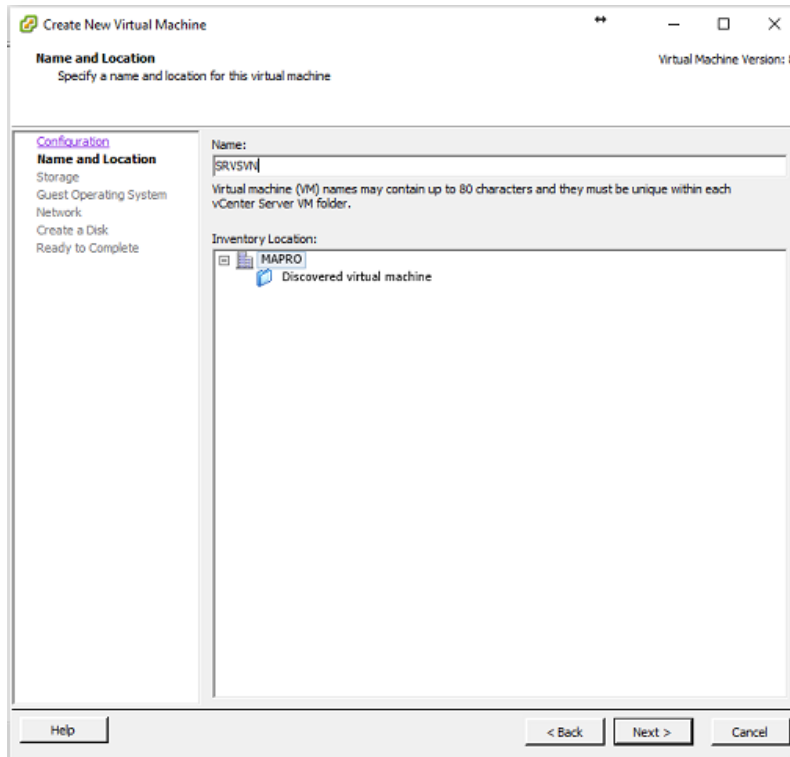


Ilustración 15 configurando la VM

Seleccionamos donde se va a alojar la VM, este sería el espacio físico para crear el HDD virtual

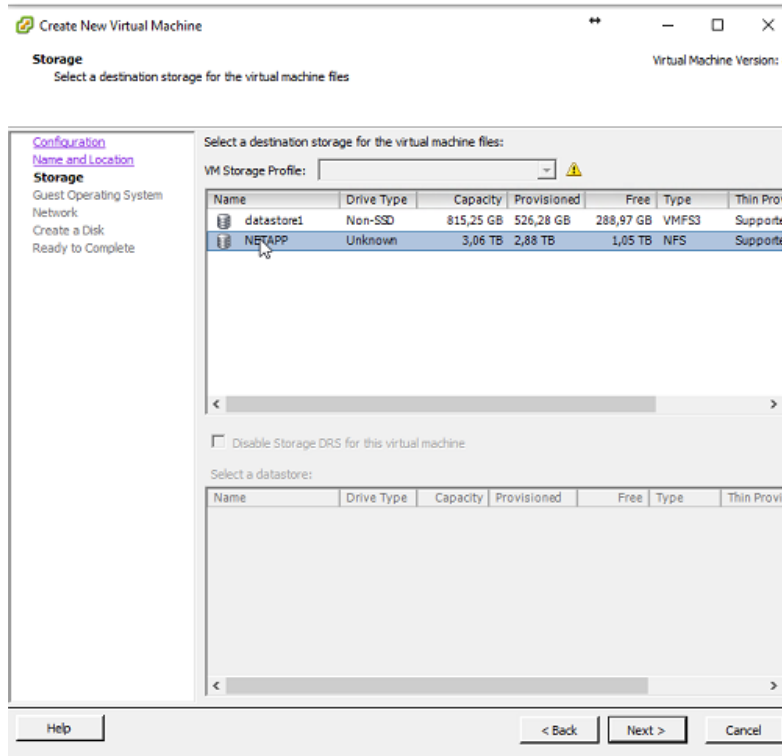


Ilustración 16 asignación de HDD

Seleccionamos el tipo de sistema operativo que vamos a instalar

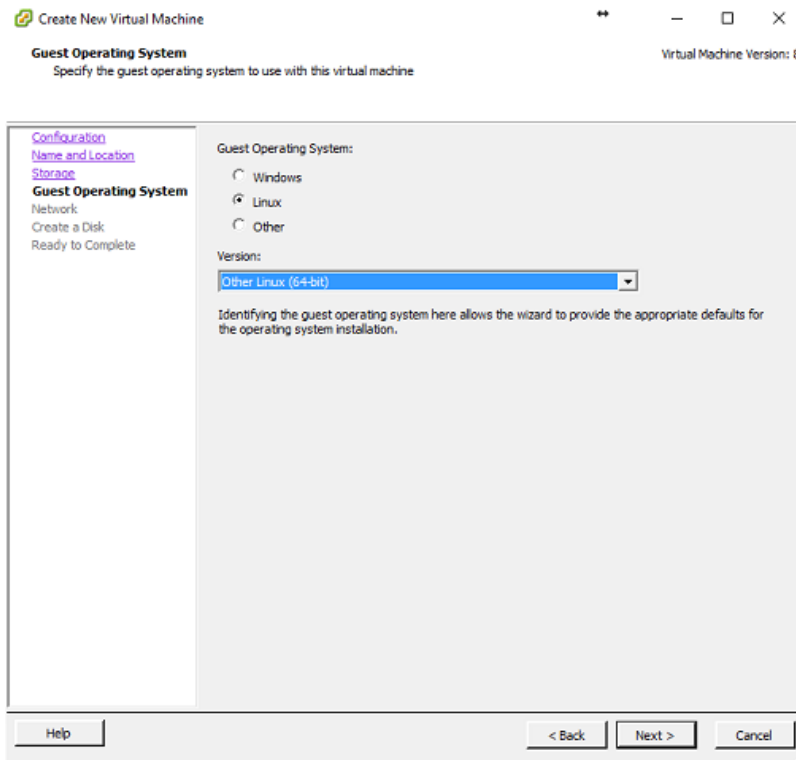


Ilustración 17 Asignamos tipo de OS

Un paso muy importante es la selección del tipo de tarjeta de red que el sistema emulara, puesto que si no se configura adecuadamente puede tenerse problemas de compatibilidad de hardware en el momento de realizar las conexiones de red.

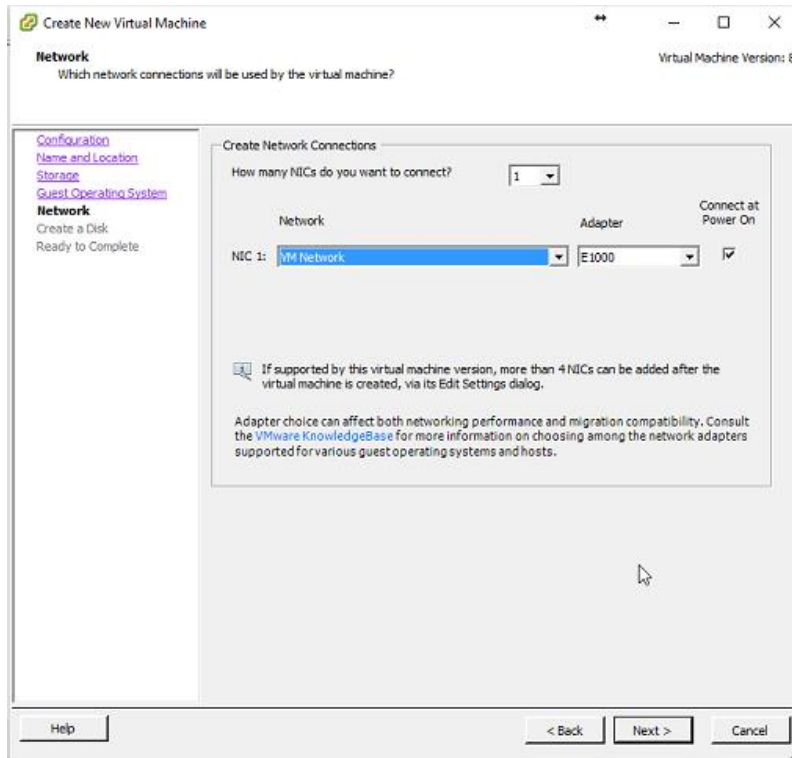


Ilustración 18 seleccionando la tarjeta de red

Damos la medida del HDD

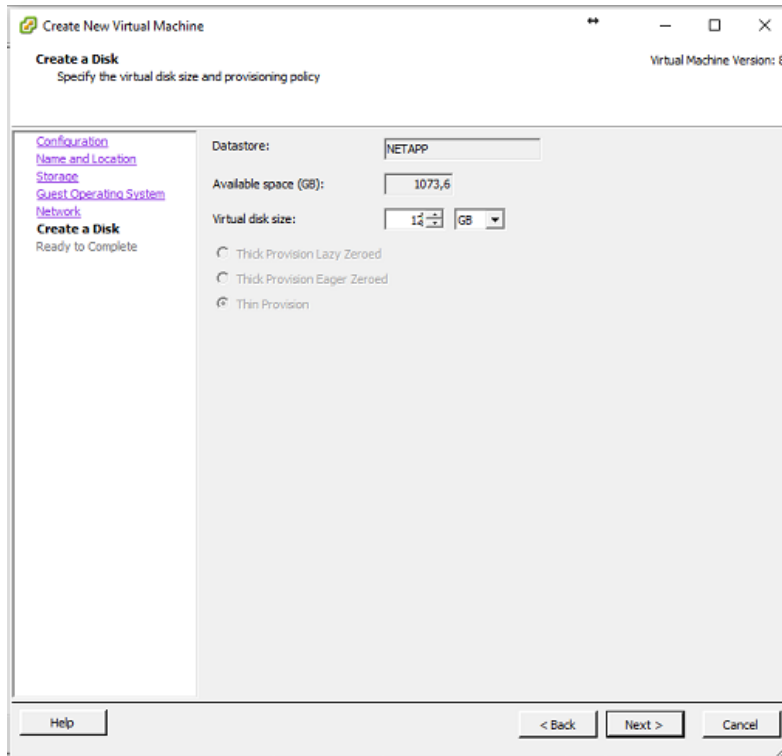


Ilustración 19 dimensionando el HDD

Resumen de las tareas a efectuarse por el hyperservidor

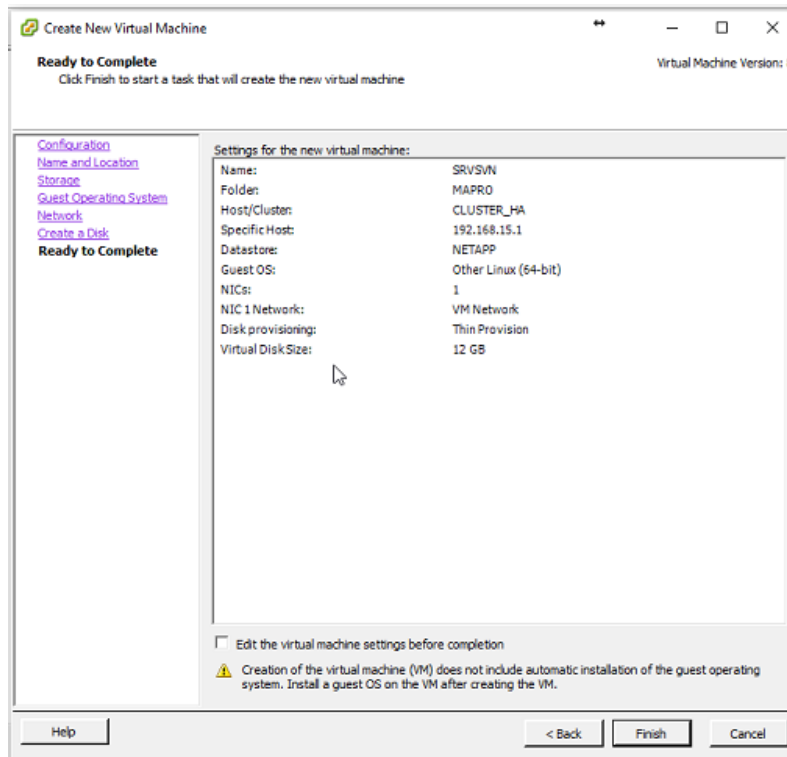


Ilustración 20 resumen de la VM

Una vez creado tenemos el servidor introducido en nuestro clúster virtual de servidores

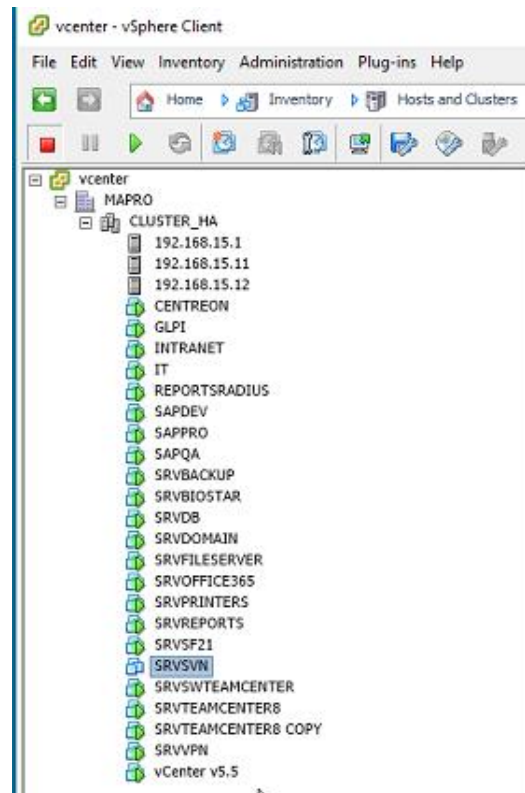


Ilustración 21 vista del servidor en la nube

8.3 Instalación del OS del servidor (Debian 8.5)

- I. Para poder instalar el OS en la máquina virtual el primer paso es cargar la imagen previamente cargada en nuestro datastore para empezar la instalación al arrancar la máquina.

Para ello vamos a opciones de la VM y en el apartado del CD/DVD Drive seleccionamos que haga la carga desde la localización de nuestra imagen .iso.

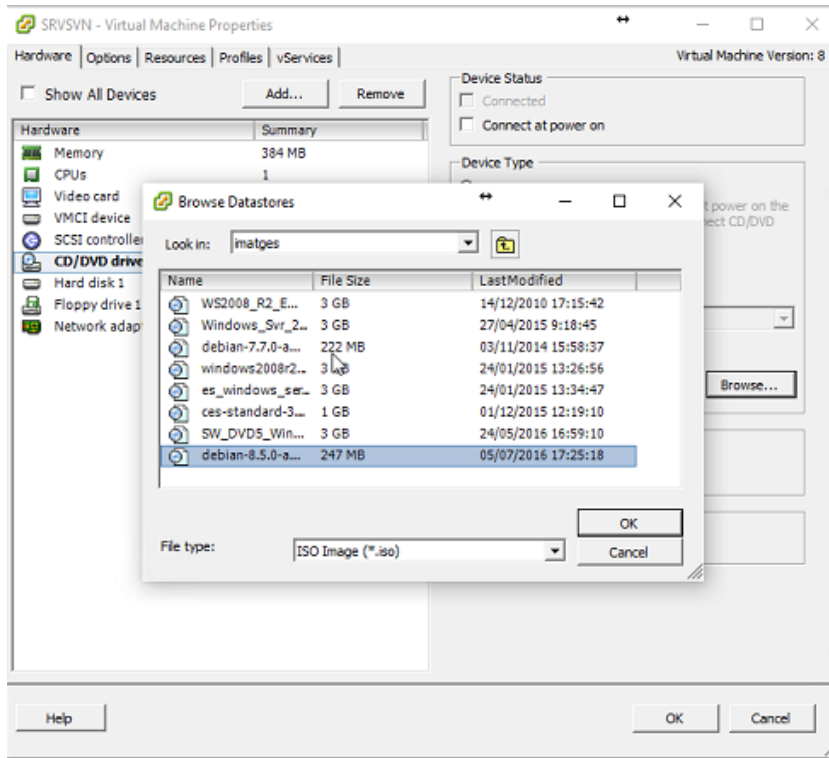


Ilustración 22 cargamos la iso del OS desde el datastore

Una vez seleccionada y aceptada la imagen estará montada para cuando arranquemos la VM

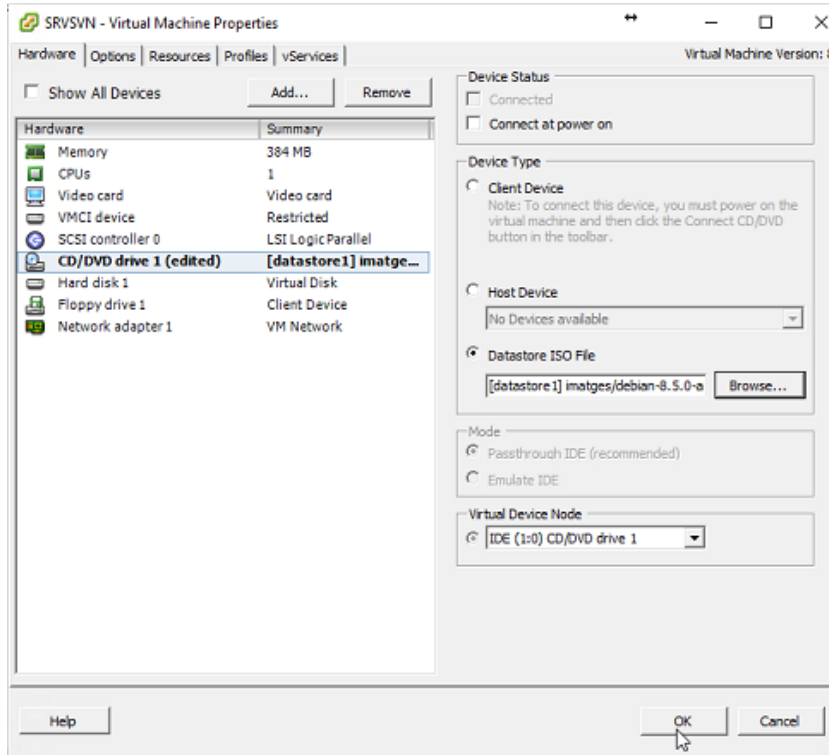


Ilustración 23 iso cargada en el datastore

II. Arrancamos la máquina virtual con la opción **Power on**

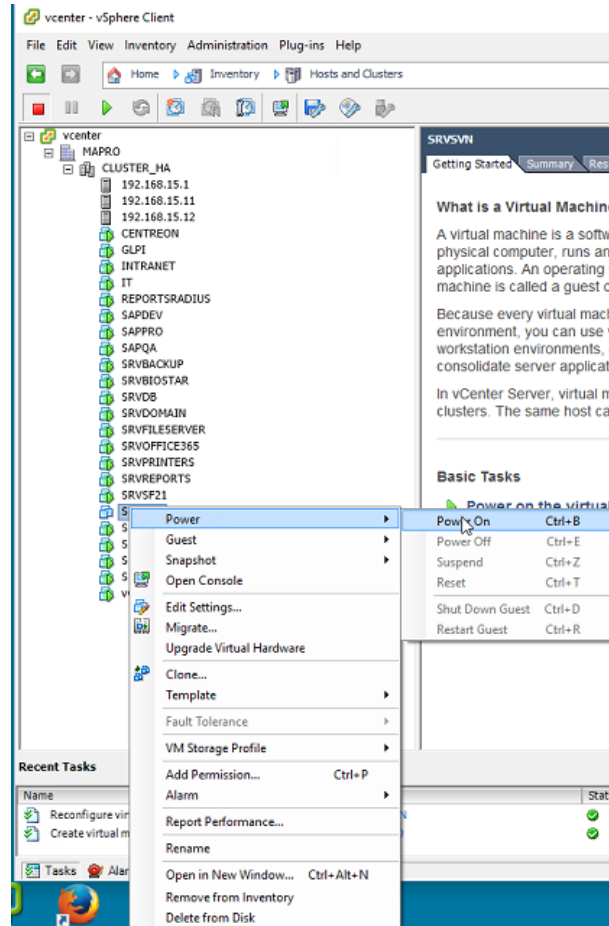


Ilustración 24 arrancando la VM

III. La máquina virtual arranca y nos da las opciones de que la imagen .iso tiene al arrancar.

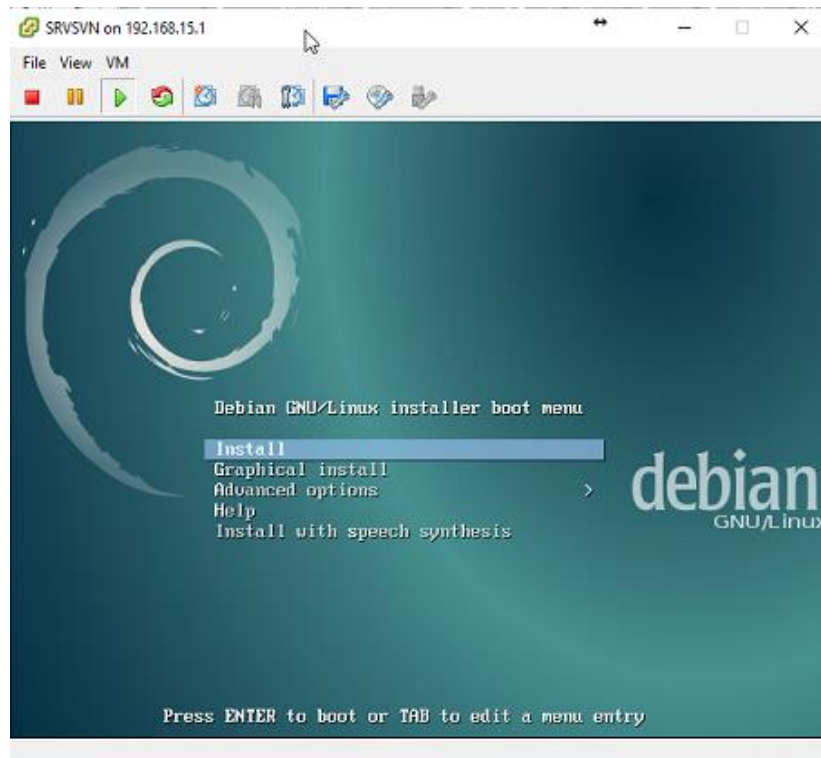


Ilustración 25 instalando debian

Le damos a la opción de instalar y configuramos el sistema

- Selección del idioma

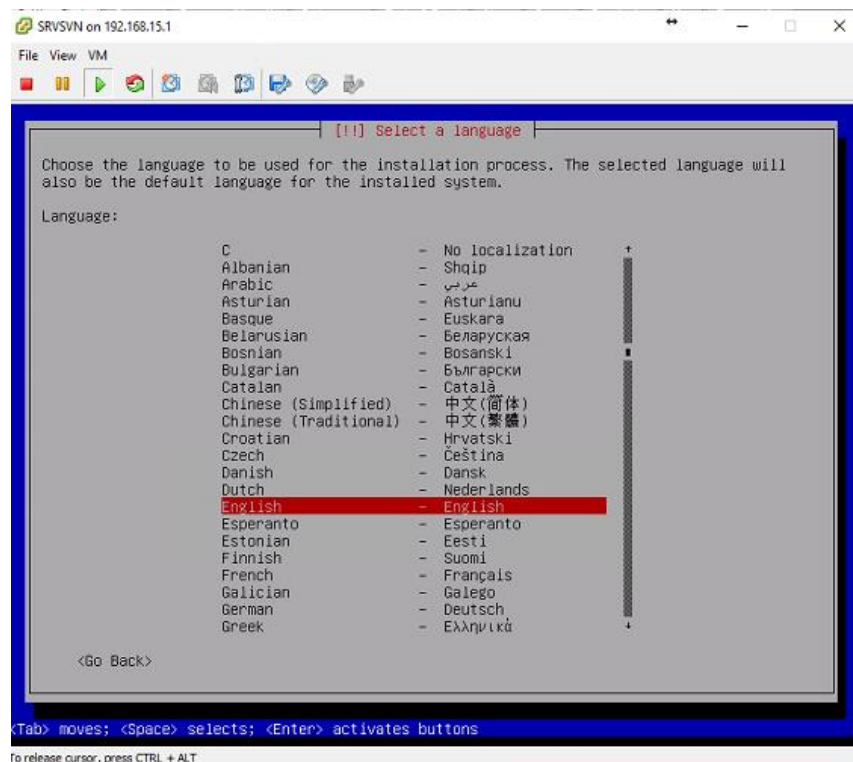


Ilustración 26 selección de idioma

- Localización

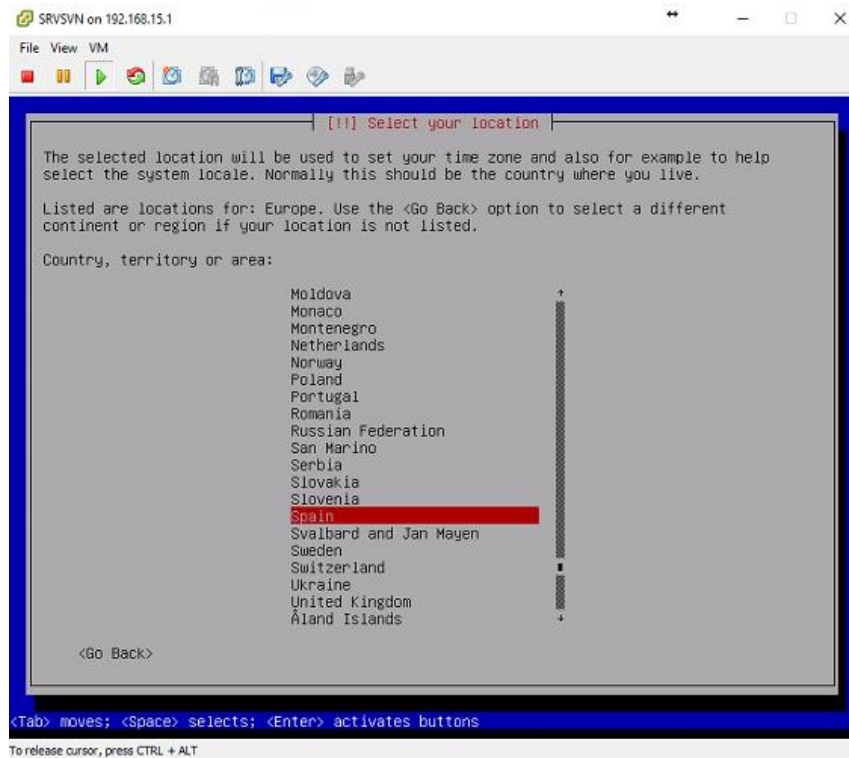


Ilustración 27 localización

Seguimos con todos los pasos guiados por el instalador del OS para configurar y dejar el sistema listo para empezar el set up del mismo.

- Progreso de la instalación

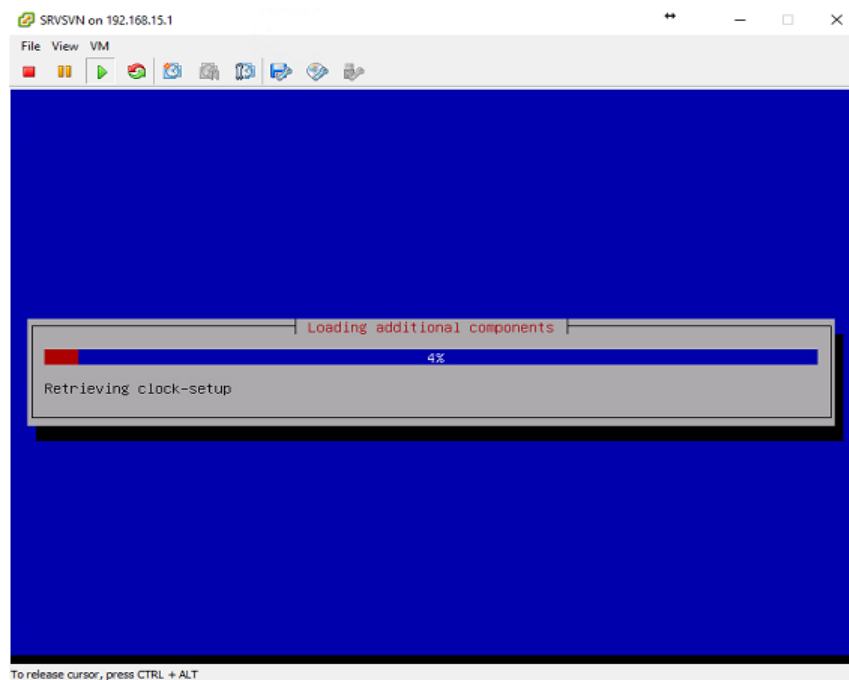


Ilustración 28 progreso de la instalación

- Configuración de la red, en este caso se realiza la configuración de manera manual para poder controlar la IP, Gateway, proxy... del sistema

Para poder tener accesible el servidor en la red y conocer siempre su dirección IP debemos darle una que sea fija y dentro del rango correspondiente.

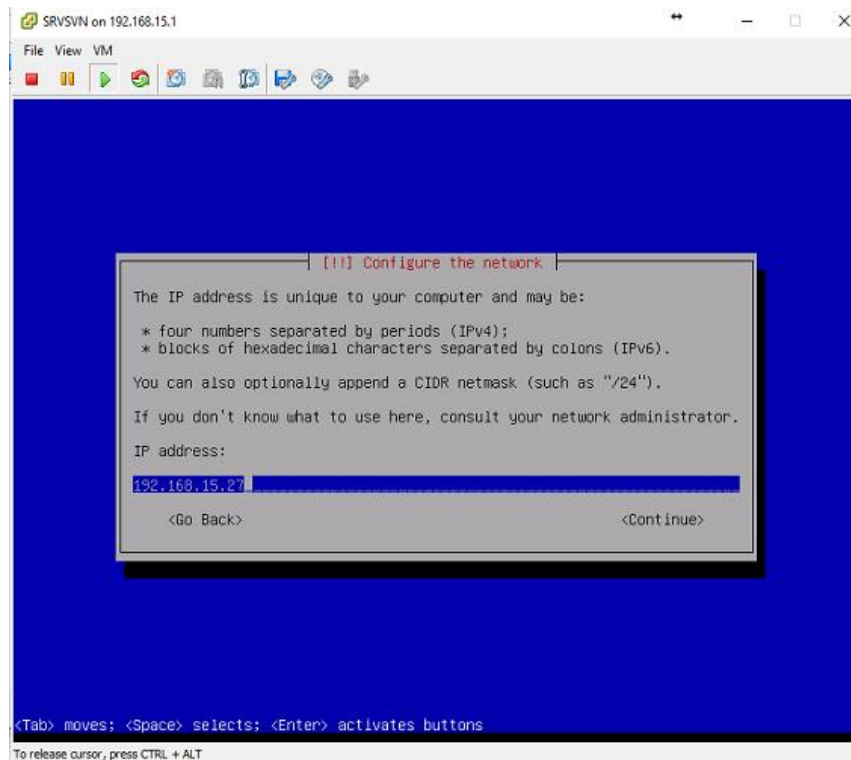


Ilustración 29 configuramos la red

Configuramos el Gateway para poder salir al exterior y poder entrar desde el exterior

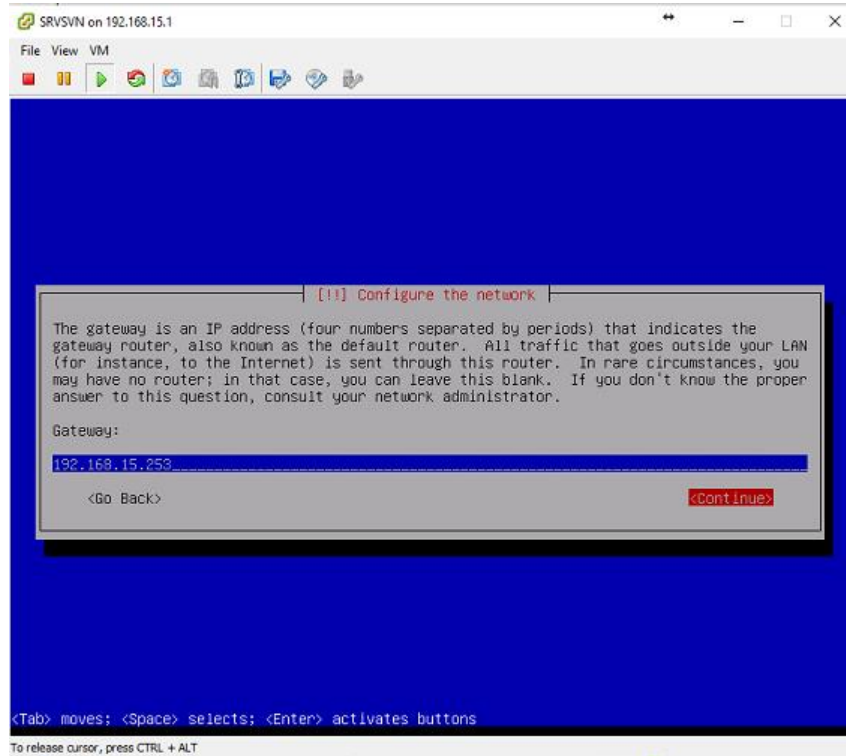


Ilustración 30 configuración del Gateway

Dirección IP de host

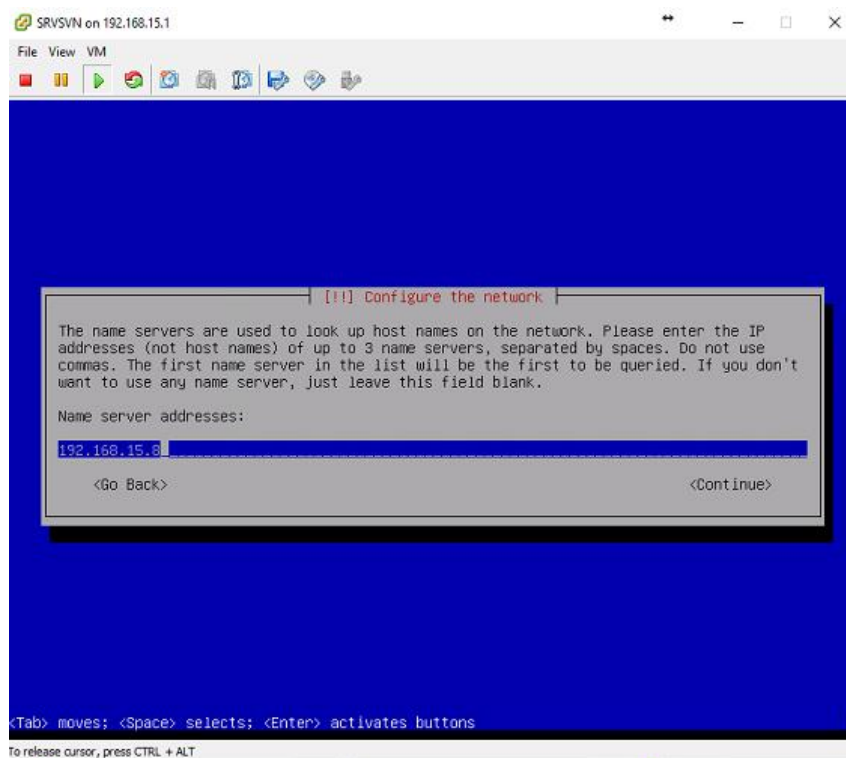


Ilustración 31 damos una IP al host

Le damos un nombre al host para poder identificarlo de una manera más descriptiva y además tenerlo identificado dentro del clúster de servidores.

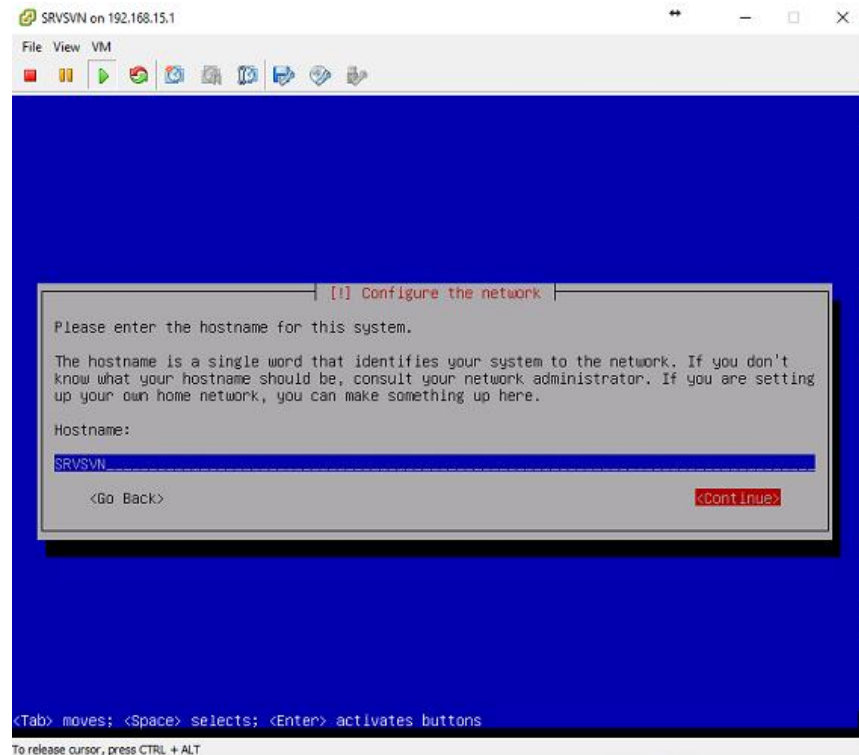


Ilustración 32 nombre del host

Configuramos el dominio en el que nos encontramos en este caso es un dominio propio de mapro y es “mapro.cat”

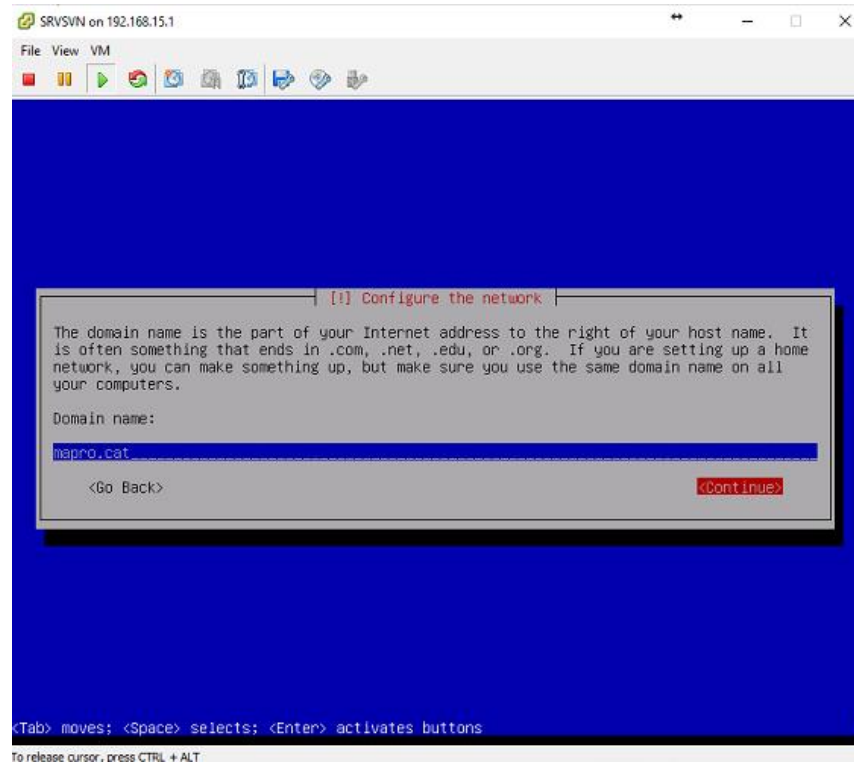


Ilustración 33 nombre del dominio, en este caso es “mapro.cat”

Creamos el usuario root y le damos un password seguro, además se crea un usuario específico también root para poder acceder en caso de bloqueo del primero

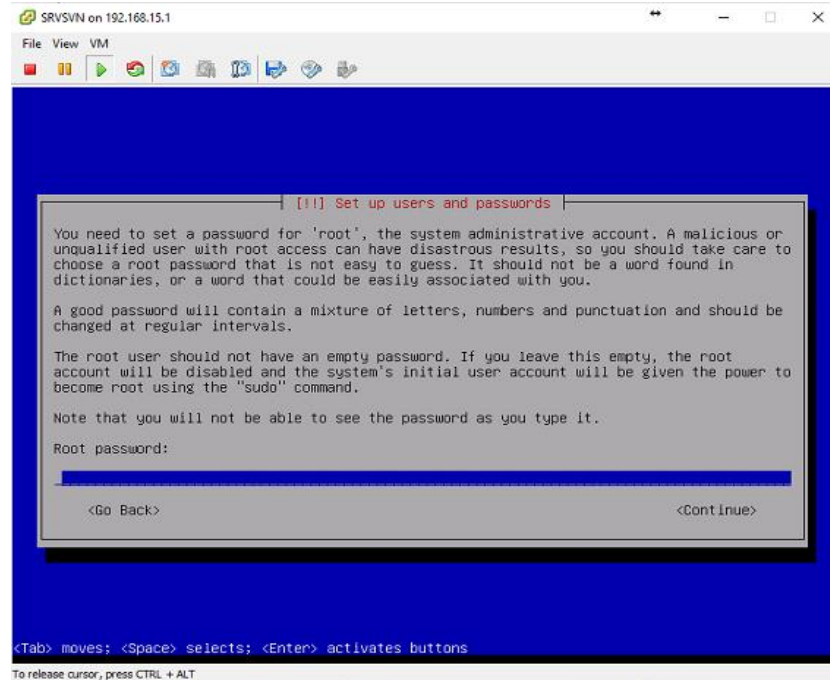


Ilustración 34 usuario root, creación de password

HDD a utilizar en la instalación

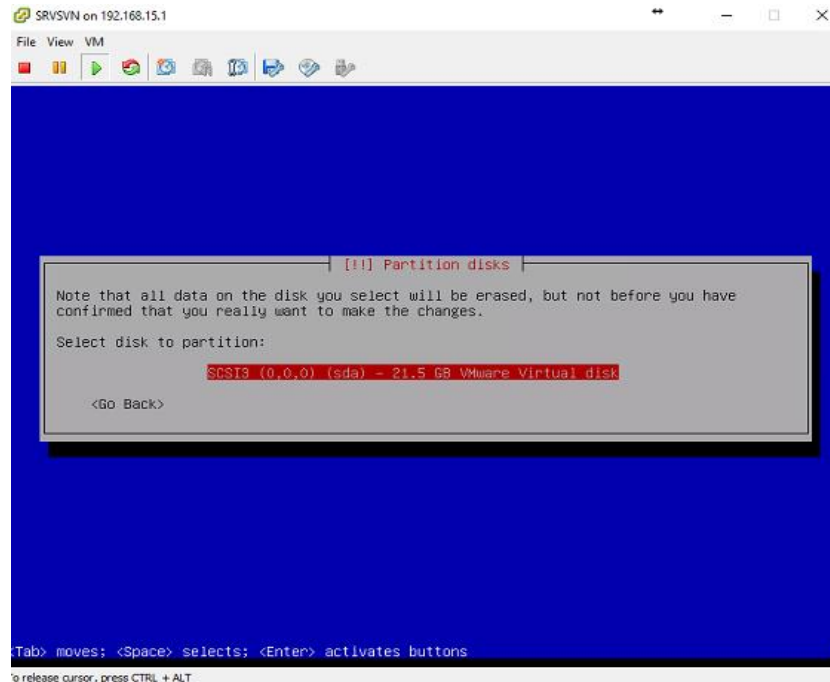


Ilustración 35 seleccionamos el HDD

Resumen del particionado del HDD

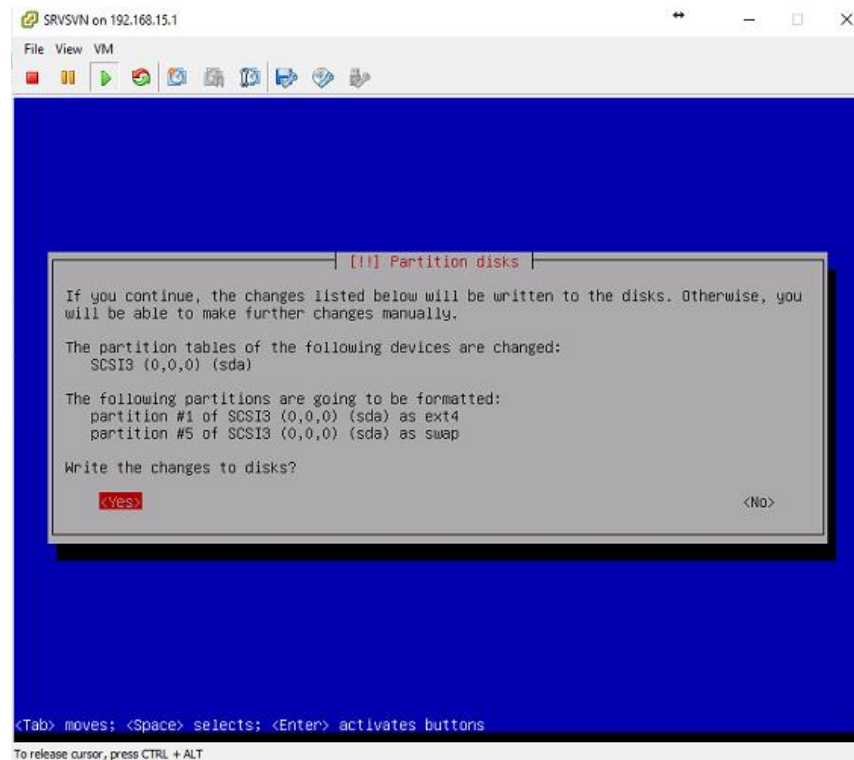


Ilustración 36 resumen del particionado del HDD

Selección del servidor de actualizaciones de debían

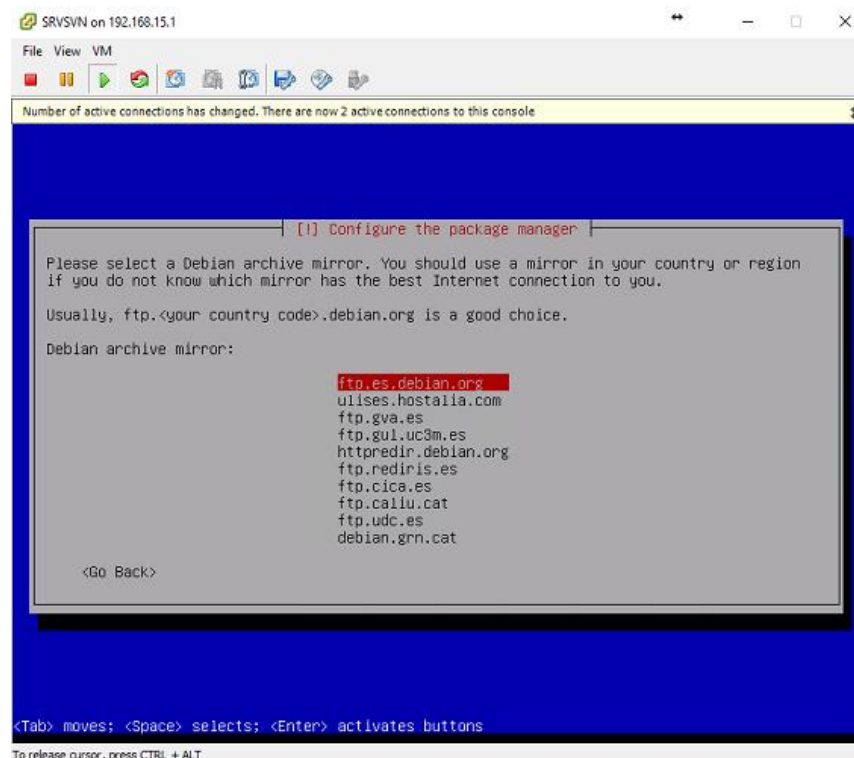


Ilustración 37 seleccionamos el servidor para actualizar debían

Después de la instalación del OS debemos instalar el cargador GRUB

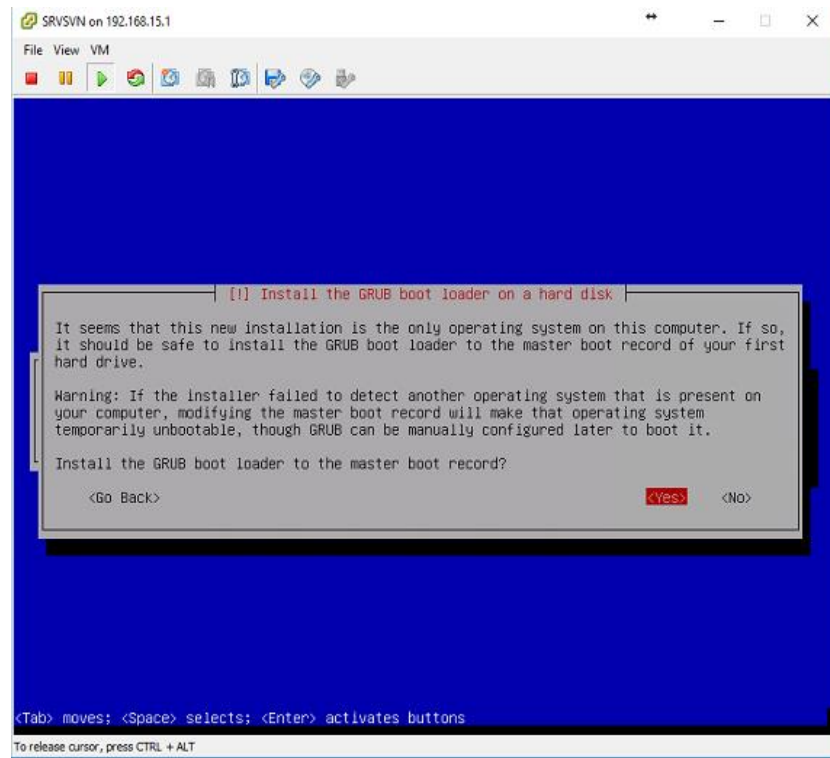


Ilustración 38 instalamos el cargador GRUB

Elegimos la localización, aunque al solo tener un solo OS instalado en este HDD virtual no tiene mucha importancia.

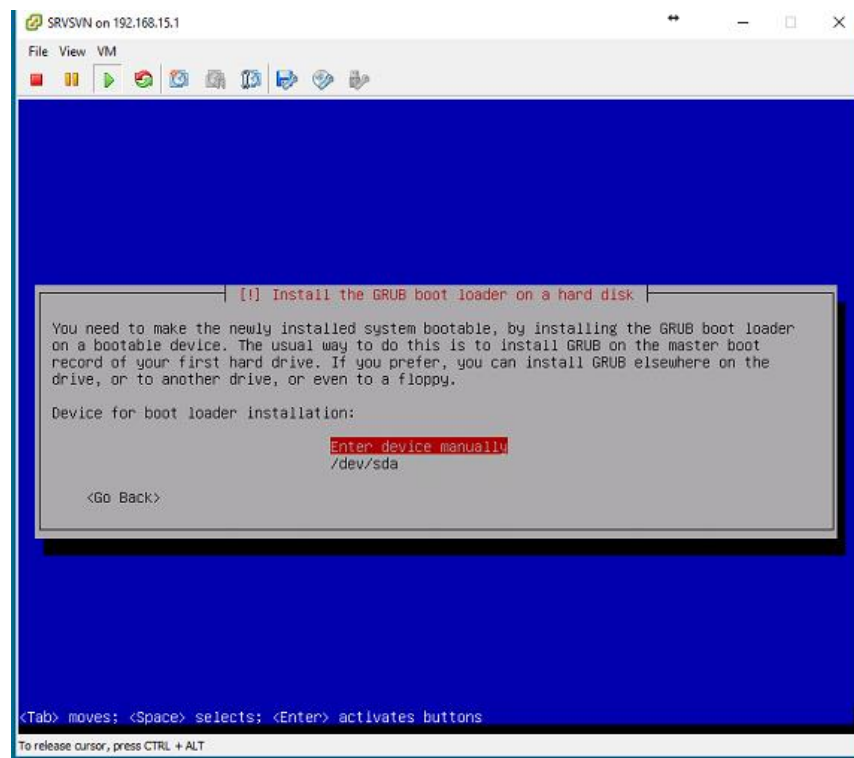


Ilustración 39 selección de donde se instalara el cargador GRUB

Resumen de una instalación completa

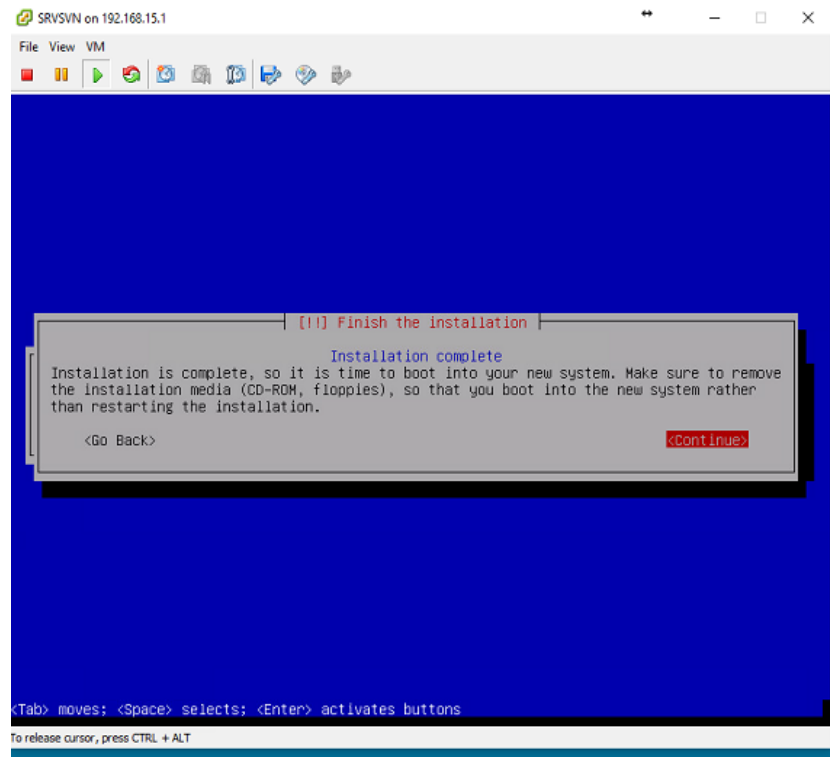
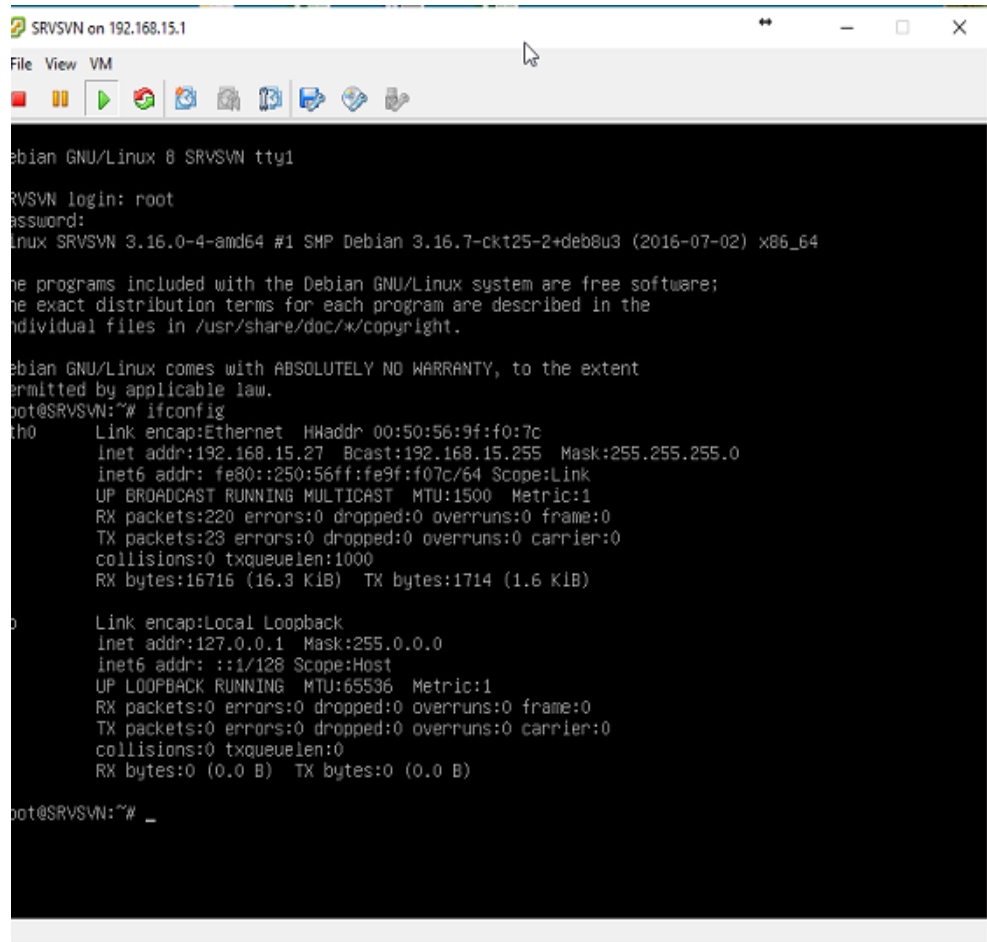


Ilustración 40 resumen de la instalación completa

Una vez instalado el OS podemos proceder a arrancarlo y a la configuración e instalación de subversion apache y todas las opciones que necesitemos para nuestro sistema de gestión y control de versiones



```

SRVSVN on 192.168.15.1
File View VM
Debian GNU/Linux 8 SRVSVN tty1
SRVSVN login: root
Password:
Linux SRVSVN 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt25-2+deb8u3 (2016-07-02) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@SRVSVN:~# ifconfig
eth0:
    Link encap:Ethernet  HWaddr 00:50:56:9f:f0:7c
    inet addr:192.168.15.27  Bcast:192.168.15.255  Mask:255.255.255.0
    inet6 addr: fe80::250:56ff:fe9f:f07c/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
    RX packets:220 errors:0 dropped:0 overruns:0 frame:0
    TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:16716 (16.3 KiB)  TX bytes:1714 (1.6 KiB)

lo:
    Link encap:Local Loopback
    inet addr:127.0.0.1  Mask:255.0.0.0
    inet6 addr: ::1/128 Scope:Host
    UP LOOPBACK RUNNING  MTU:65536  Metric:1
    RX packets:0 errors:0 dropped:0 overruns:0 frame:0
    TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:0
    RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@SRVSVN:~# _
    
```

Ilustración 41 sistema arrancado

Para poder acceder se configura un paso seguro SSH y con los programas “PUTTY” y WinScp nos conectamos tanto por medio del terminal como para hacer un FTP para cargar y descargar archivos desde el exterior. A continuación, mostramos dos imágenes que corresponden a la entrada de IP de PuTTY y de WinScp

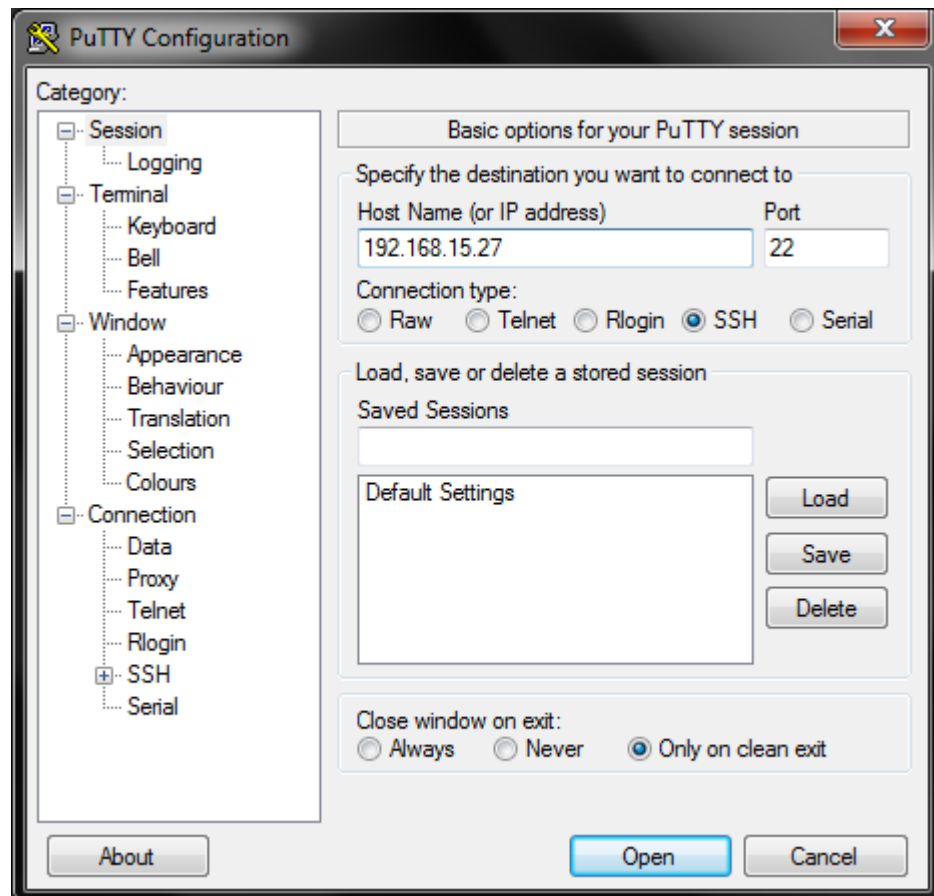


Ilustración 42 configuración de PuTTY.

Log in completo con PUTTY desde el cual podemos acceder a todos los servicios y configuraciones del servidor. PuTTY es un sistema de conexión vía SSH que nos permite abrir un terminal remoto conectado hacia el servidor, básicamente funciona como un cliente que se conecta en remoto, esto nos permite no necesariamente estar físicamente en la localización del servidor.

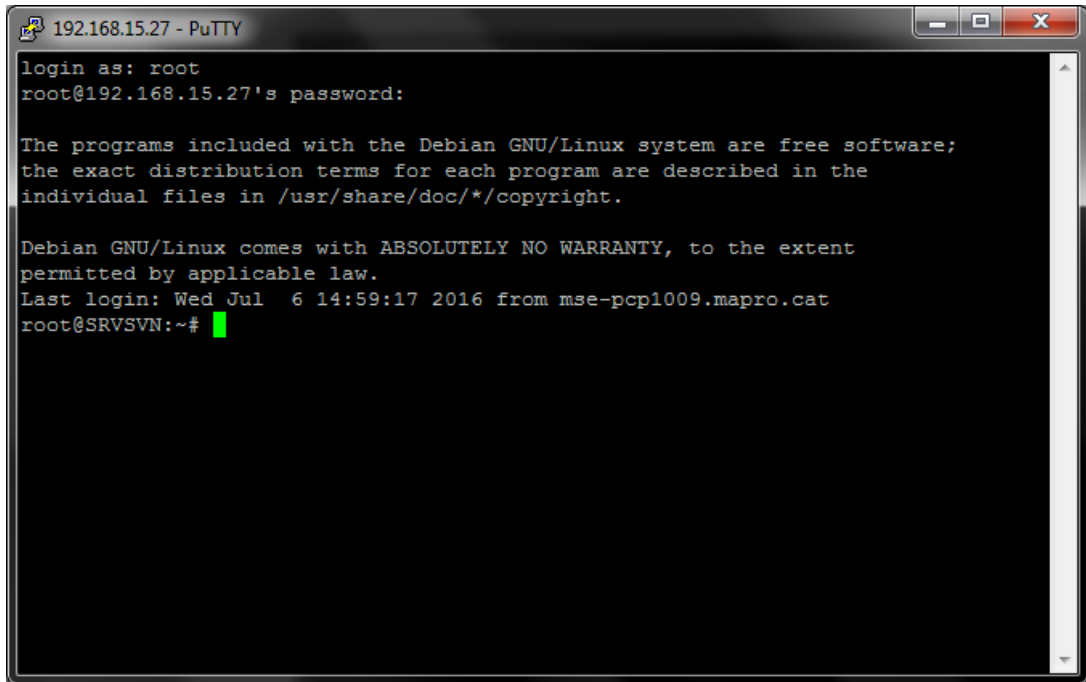


Ilustración 43 login completo con PuTTY

También utilizamos otro software que se llama WinSCP que nos permite abrir fia FTP/SSH archivos del servidor en remoto, esto nos ayuda a poder editar si es necesario o a poder realizar alguna copia directa de la estructura de ficheros que disponemos.

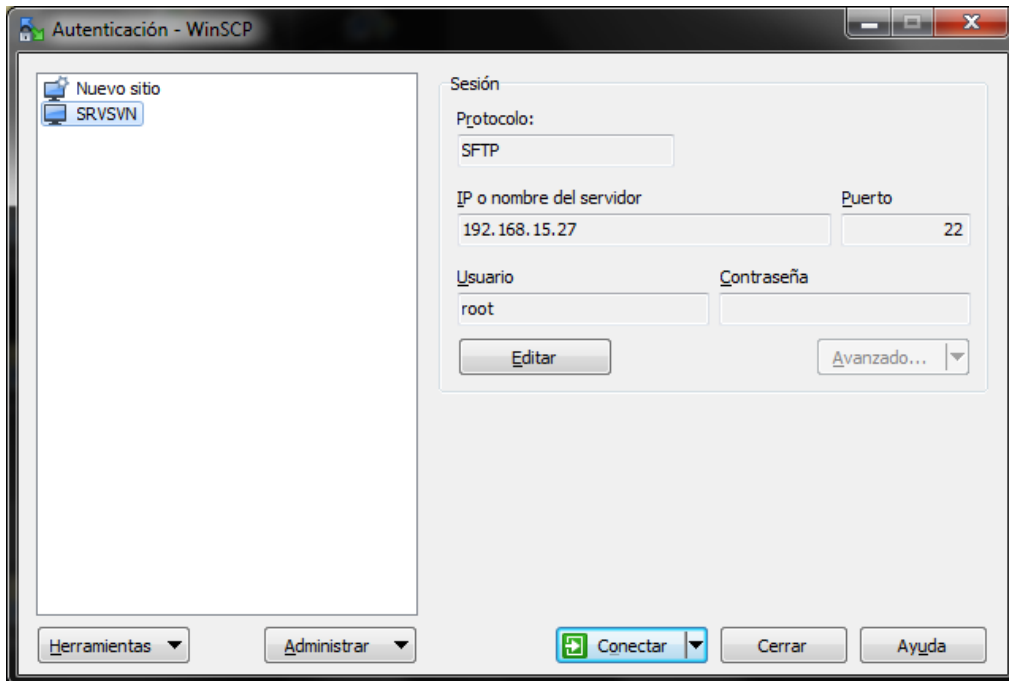


Ilustración 44 conectando por WinSCP

Una vez dentro con WinSCP podemos ver y acceder a toda la estructura del árbol de directorios que hay creada en el servidor

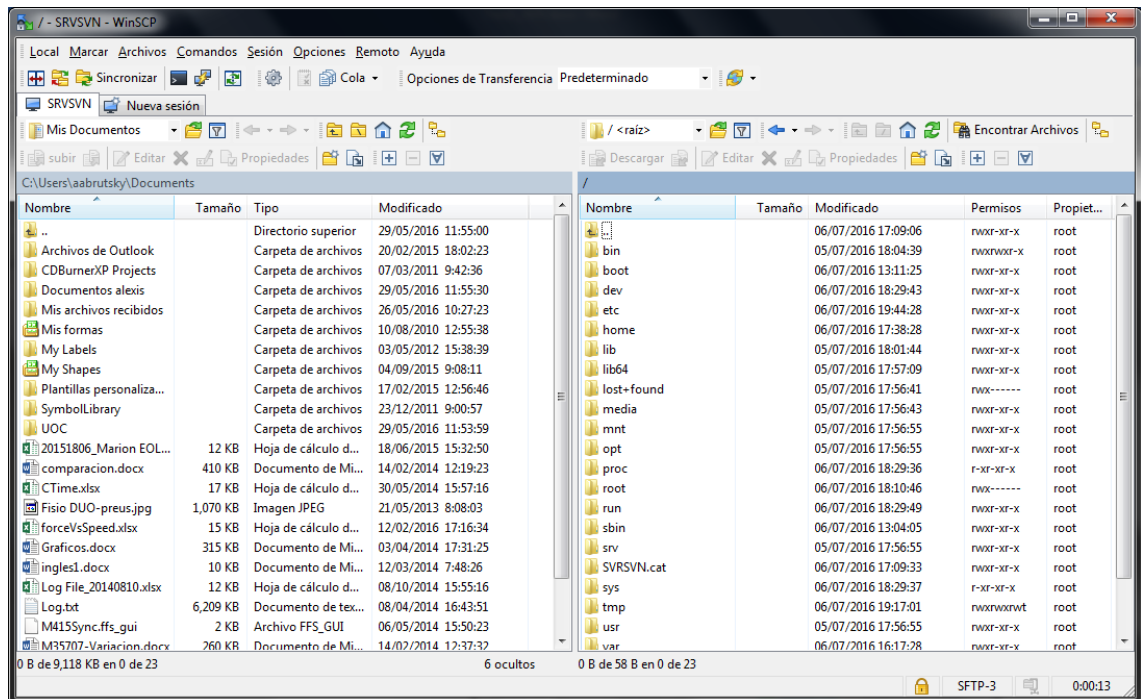
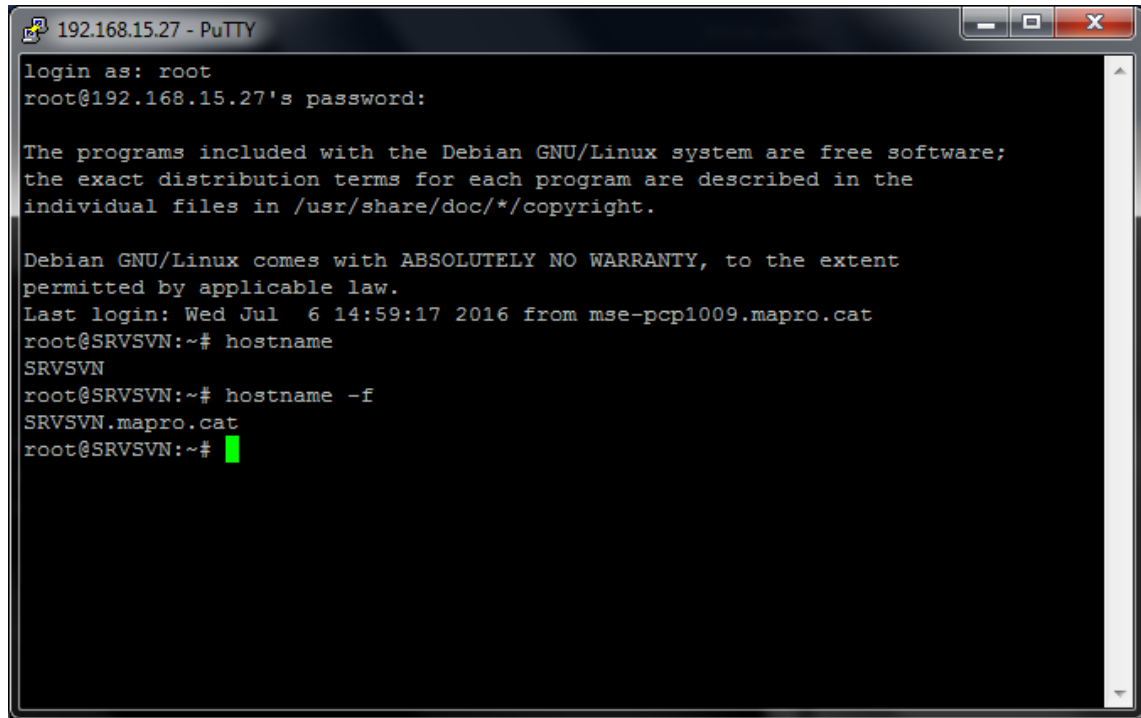


Ilustración 45 Vista de WinSCP una vez conectado al servidor

8.4 Elección del sistema de control de versiones elegido.

Después de los análisis previos se decide por instalar un SCV bajo subversion utilizando el cliente TortoiseSVN debido a la cantidad de herramientas y a la facilidad de integración con los sistemas que se utilizan para el desarrollo de aplicaciones software. Además, verificamos el nombre del servidor para saber que realmente estamos en el servidor que queremos



```

192.168.15.27 - PuTTY
login as: root
root@192.168.15.27's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jul 6 14:59:17 2016 from mse-pcp1009.mapro.cat
root@SRVSVN:~# hostname
SRVSVN
root@SRVSVN:~# hostname -f
SRVSVN.mapro.cat
root@SRVSVN:~#
  
```

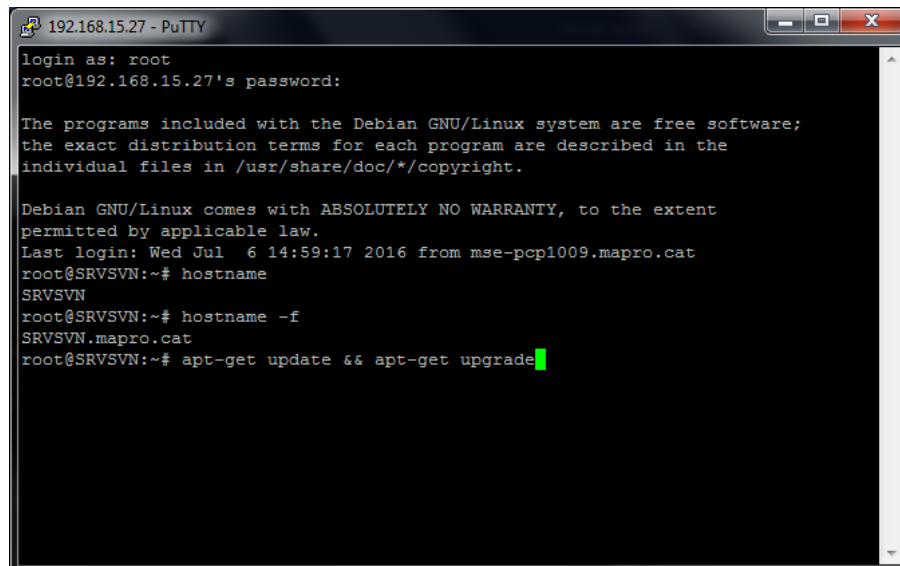
Ilustración 46 verificación del nombre y dominio

8.5 Instalación de subversión en el servidor SVN

- Apache2
Para instalar apache 2 hay que seguir unos cuantos pasos ya que debe instalarse y configurarse adecuadamente para poder conseguir nuestro propósito.

a) Actualizar repositorio

```
apt-get update && apt-get upgrade
```



```

192.168.15.27 - PuTTY
login as: root
root@192.168.15.27's password:

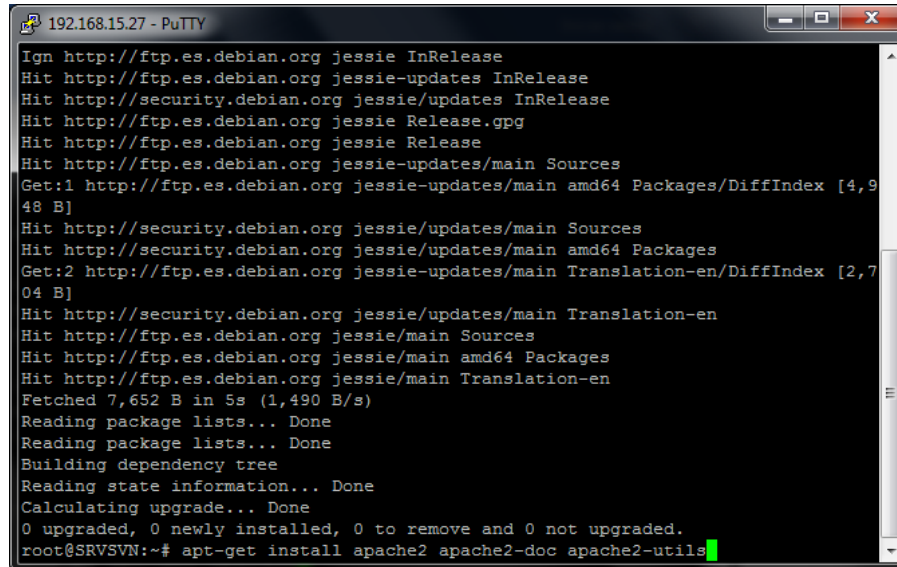
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jul 6 14:59:17 2016 from mse-pcp1009.mapro.cat
root@SRVSVN:~# hostname
SRVSVN
root@SRVSVN:~# hostname -f
SRVSVN.mapro.cat
root@SRVSVN:~# apt-get update && apt-get upgrade
  
```

Ilustración 47 actualizamos repositorios de debian

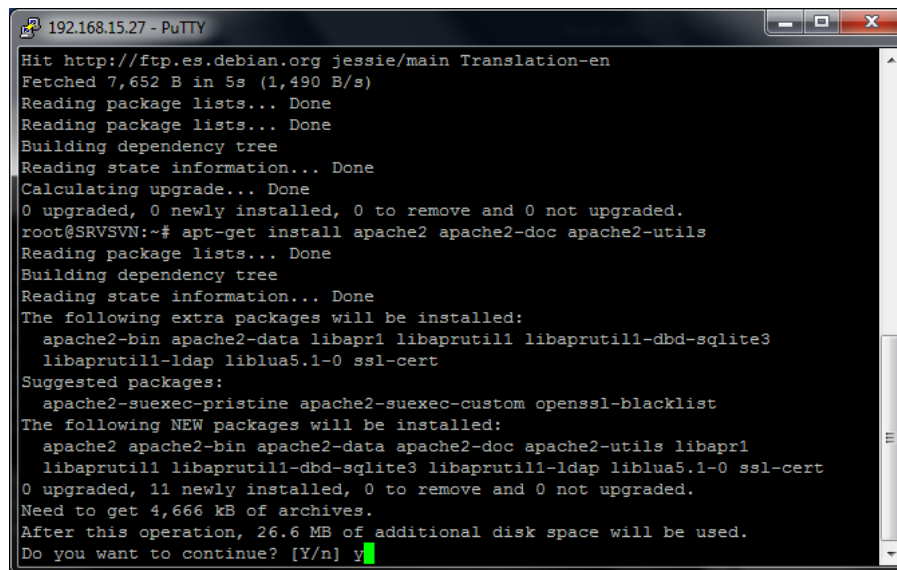
b) Instalar apache2

```
apt-get install apache2 apache2-doc apache2-utils
```



```
192.168.15.27 - PuTTY
Ign http://ftp.es.debian.org jessie InRelease
Hit http://ftp.es.debian.org jessie-updates InRelease
Hit http://security.debian.org jessie/updates InRelease
Hit http://ftp.es.debian.org jessie Release.gpg
Hit http://ftp.es.debian.org jessie Release
Hit http://ftp.es.debian.org jessie-updates/main Sources
Get:1 http://ftp.es.debian.org jessie-updates/main amd64 Packages/DiffIndex [4,948 B]
Hit http://security.debian.org jessie/updates/main Sources
Hit http://security.debian.org jessie/updates/main amd64 Packages
Get:2 http://ftp.es.debian.org jessie-updates/main Translation-en/DiffIndex [2,704 B]
Hit http://security.debian.org jessie/updates/main Translation-en
Hit http://ftp.es.debian.org jessie/main Sources
Hit http://ftp.es.debian.org jessie/main amd64 Packages
Hit http://ftp.es.debian.org jessie/main Translation-en
Fetched 7,652 B in 5s (1,490 B/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@SRVSVN:~# apt-get install apache2 apache2-doc apache2-utils
```

Ilustración 48 instalando apache 2



```
192.168.15.27 - PuTTY
Hit http://ftp.es.debian.org jessie/main Translation-en
Fetched 7,652 B in 5s (1,490 B/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@SRVSVN:~# apt-get install apache2 apache2-doc apache2-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-bin apache2-data libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.1-0 ssl-cert
Suggested packages:
  apache2-suexec-pristine apache2-suexec-custom openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-doc apache2-utils libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 4,666 kB of archives.
After this operation, 26.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Ilustración 49 aceptamos la instalacion

```

192.168.15.27 - PuTTY
root@SRVSVN:~# apache2 -version
Server version: Apache/2.4.10 (Debian)
Server built:   Nov 28 2015 14:05:48
root@SRVSVN:~#
    
```

Ilustración 50 Verificamos la versión de apache instalada

Verificamos conexión al servidor apache

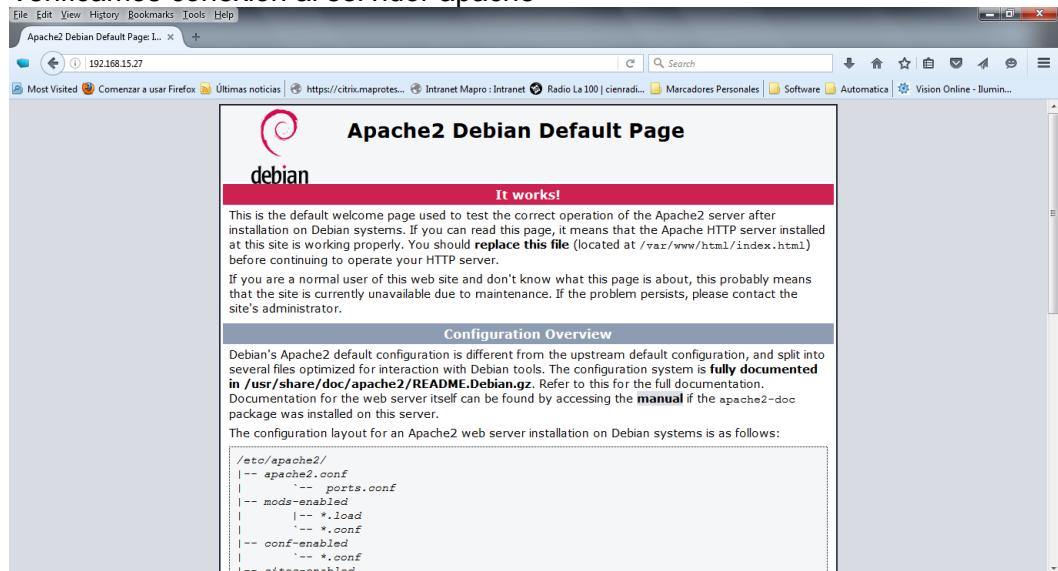


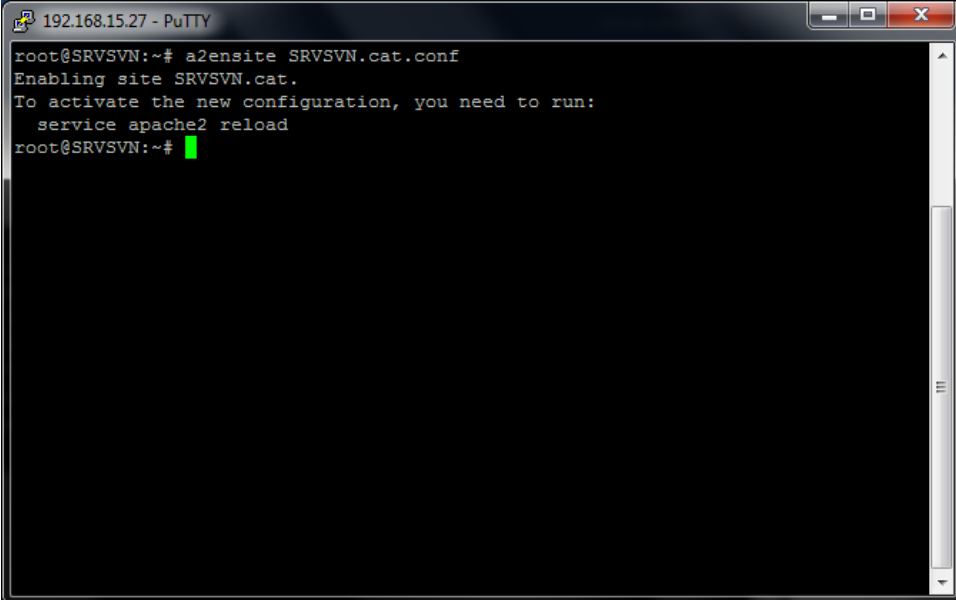
Ilustración 51 verificación de que apache 2 funciona

- c) Apaches como virtual hosting
Deshabilitamos es sitio por defecto

`a2dissite default`

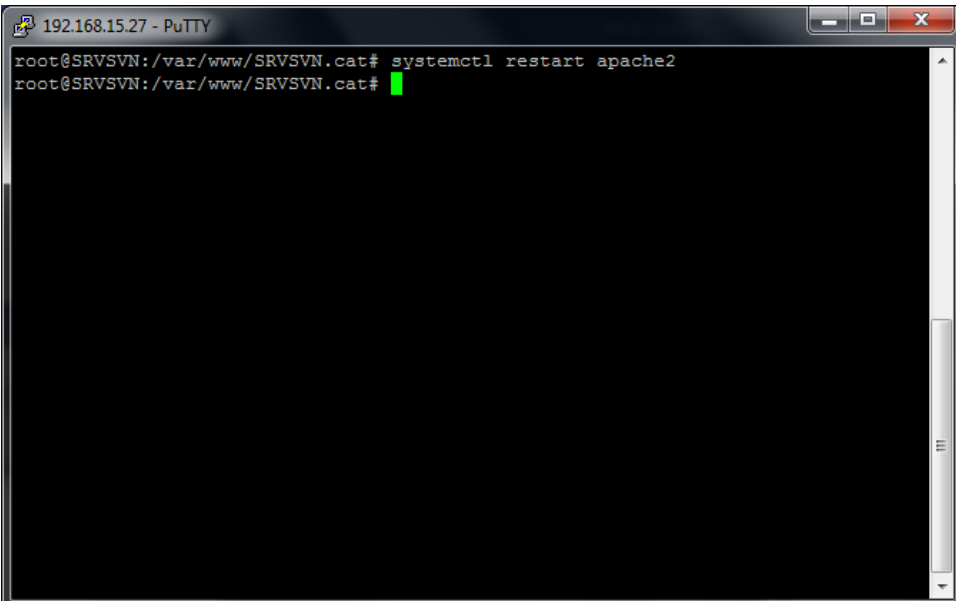
Después editamos el fichero `“/etc/apache2/sites-available/SRVSVN.cat.conf”` de manera que quede como:

- d) Una vez creado el fichero de configuración debemos recargarlo y reiniciar el servidor



```
192.168.15.27 - PuTTY
root@SRVSVN:~# a2ensite SRVSVN.cat.conf
Enabling site SRVSVN.cat.
To activate the new configuration, you need to run:
  service apache2 reload
root@SRVSVN:~# █
```

Ilustración 54 recargamos el servidor



```
192.168.15.27 - PuTTY
root@SRVSVN:/var/www/SRVSVN.cat# systemctl restart apache2
root@SRVSVN:/var/www/SRVSVN.cat# █
```

Ilustración 55 reiniciamos el servidor

De esta manera y con esos pasos sencillos tenemos el servidor APACHE2 V2.4 ya instalado y operativo.

- Subversion.
Para instalar subversión + la librería de apache2 en debian es necesario realizar estas dos operaciones:

```

$ apt-get install subversion
$ apt-get install libapache2-svn
    
```

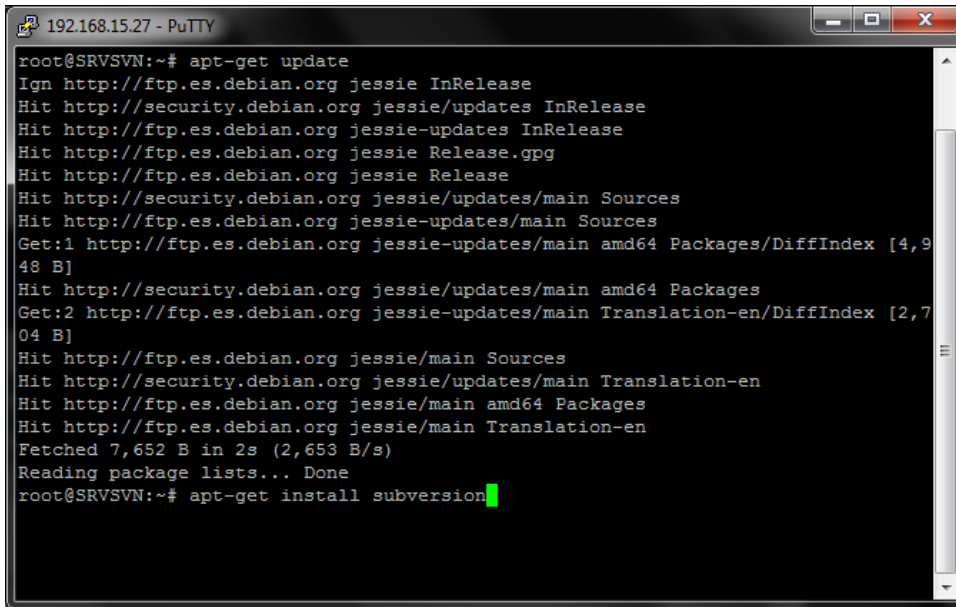


Ilustración 56 instalar SVN

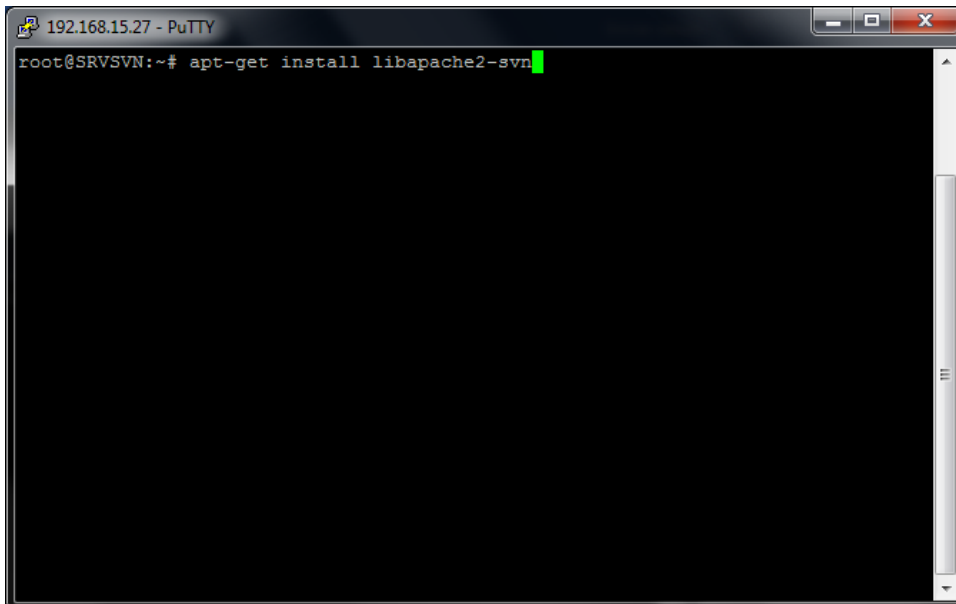


Ilustración 57 instalamos la librería de apache 2

- Una vez instalado subversión creamos los repositorios

```

Debian 5 - VMware Player  File  VM  Help
Get:3 http://debian.grn.cat/debian/ jessie/main subversion i386 1.8.10-6+deb8u4 [931
Fetched 2,147 kB in 7s (281 kB/s)
Selecting previously unselected package libserf-1-1:i386.
(Reading database ... 33746 files and directories currently installed.)
Preparing to unpack .../libserf-1-1_1.3.8-1_i386.deb ...
Unpacking libserf-1-1:i386 (1.3.8-1) ...
Selecting previously unselected package libsvn1:i386.
Preparing to unpack .../libsvn1_1.8.10-6+deb8u4_i386.deb ...
Unpacking libsvn1:i386 (1.8.10-6+deb8u4) ...
Selecting previously unselected package subversion.
Preparing to unpack .../subversion_1.8.10-6+deb8u4_i386.deb ...
Unpacking subversion (1.8.10-6+deb8u4) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up libserf-1-1:i386 (1.3.8-1) ...
Setting up libsvn1:i386 (1.8.10-6+deb8u4) ...
Setting up subversion (1.8.10-6+deb8u4) ...
Processing triggers for libc-bin (2.19-18+deb8u4) ...
root@SRVSVN:~# apt-get install -y tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  tree
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 47.9 kB of archives.
After this operation, 137 kB of additional disk space will be used.
Get:1 http://debian.grn.cat/debian/ jessie/main tree i386 1.7.0-3 [47.9 kB]
Fetched 47.9 kB in 0s (140 kB/s)
Selecting previously unselected package tree.
(Reading database ... 33851 files and directories currently installed.)
Preparing to unpack .../archives/tree_1.7.0-3_i386.deb ...
Unpacking tree (1.7.0-3) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up tree (1.7.0-3) ...
root@SRVSVN:~# mkdir -p /svnrepos/project1
root@SRVSVN:~# svnadmin create /svnrepos/project1
root@SRVSVN:~#
    
```

Ilustración 58

- Una vez creado los repositorios damos permisos a los usuarios para poder acceder a ellos

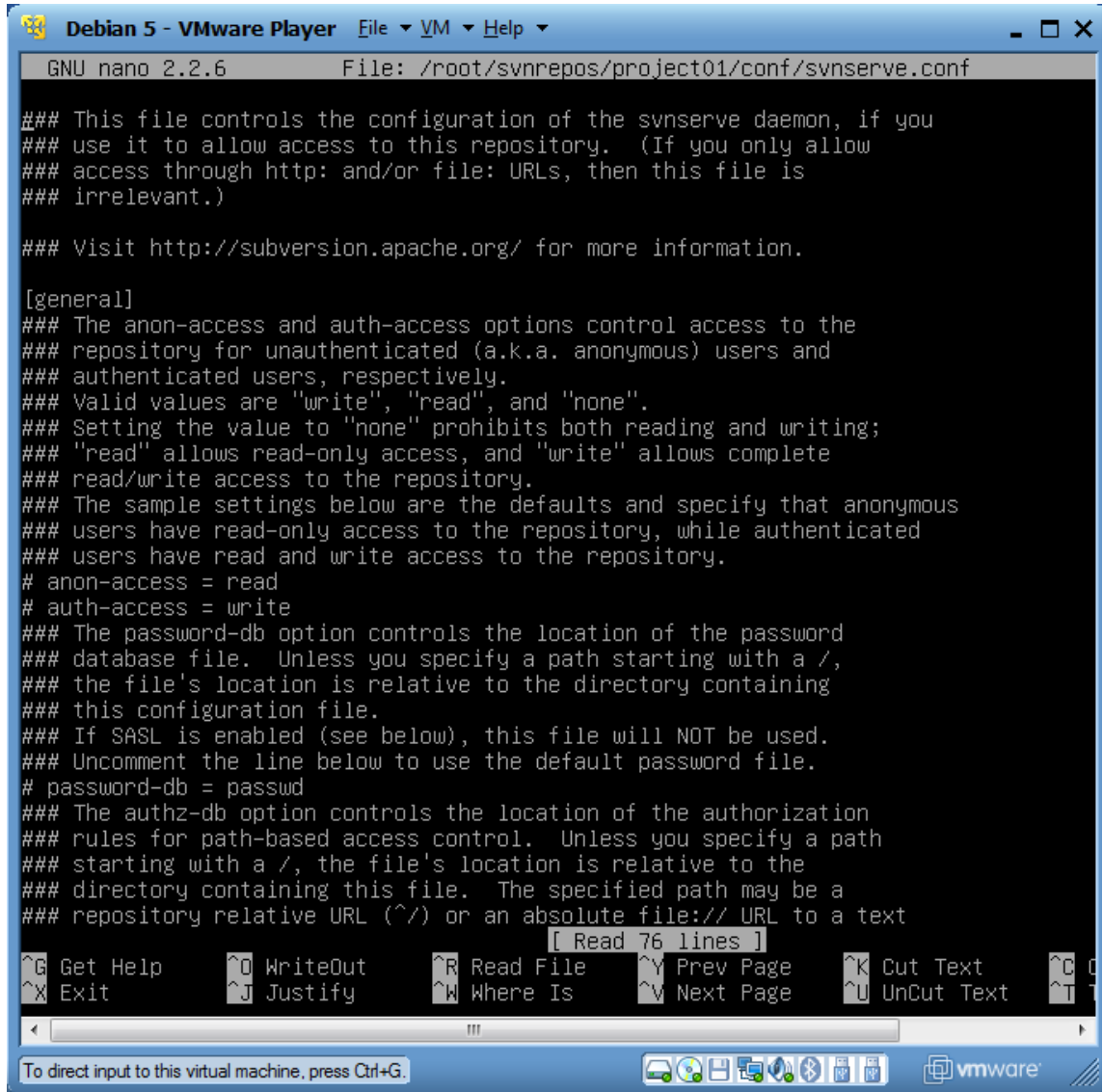
```

Debian 5 - VMware Player  File  VM  Help
root@SRVSVN:~# nano /svnrepos/project/conf/svnserve.conf_
    
```

Ilustración 59

- Configuramos svnservicio descomentando un par de líneas para dar permisos de escritura y lectura y autenticación por password

-



```

GNU nano 2.2.6      File: /root/svnrepos/project01/conf/svnservicio.conf

### This file controls the configuration of the svnservicio daemon, if you
### use it to allow access to this repository. (If you only allow
### access through http: and/or file: URLs, then this file is
### irrelevant.)

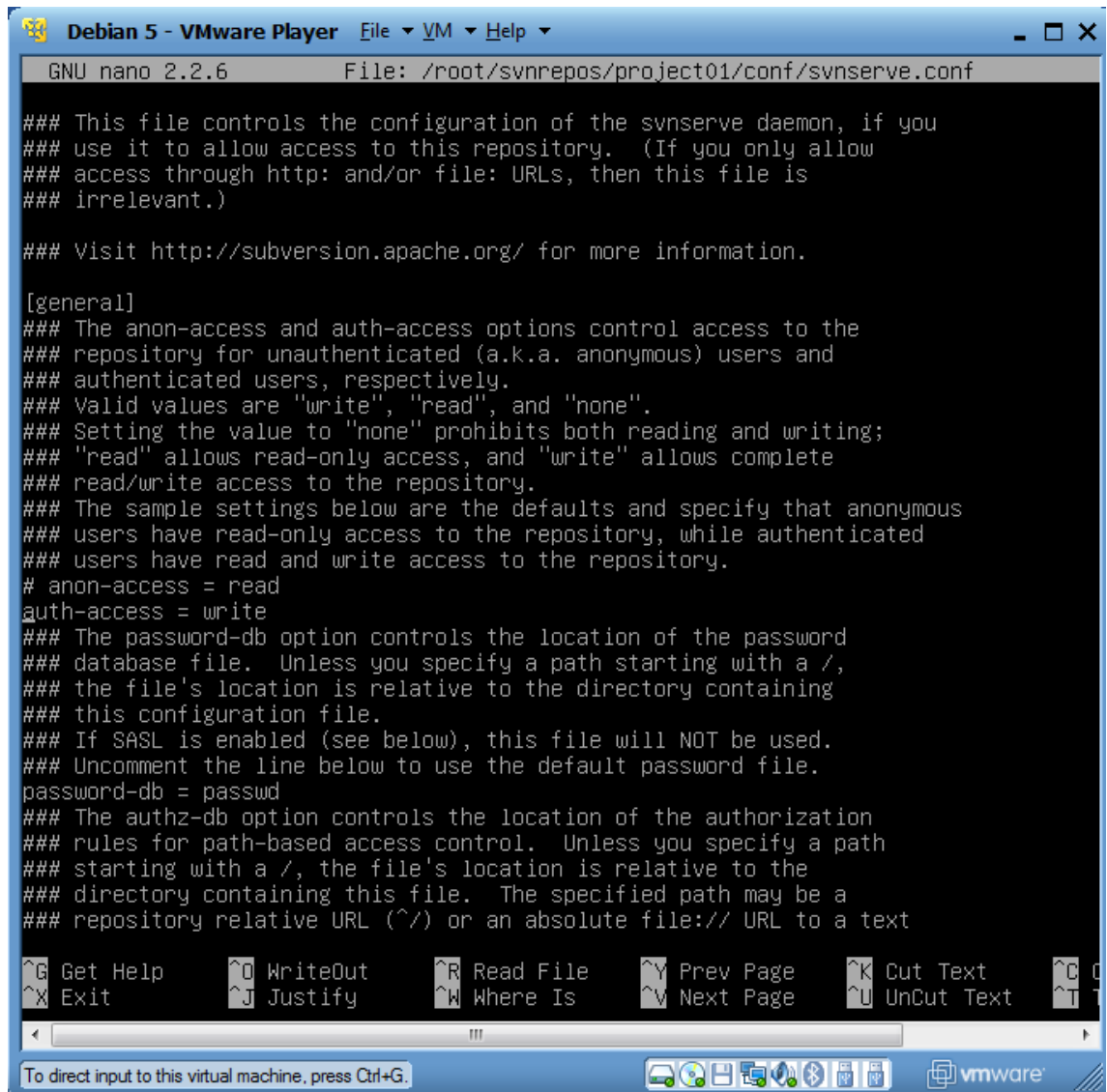
### Visit http://subversion.apache.org/ for more information.

[general]
### The anon-access and auth-access options control access to the
### repository for unauthenticated (a.k.a. anonymous) users and
### authenticated users, respectively.
### Valid values are "write", "read", and "none".
### Setting the value to "none" prohibits both reading and writing;
### "read" allows read-only access, and "write" allows complete
### read/write access to the repository.
### The sample settings below are the defaults and specify that anonymous
### users have read-only access to the repository, while authenticated
### users have read and write access to the repository.
# anon-access = read
# auth-access = write
### The password-db option controls the location of the password
### database file. Unless you specify a path starting with a /,
### the file's location is relative to the directory containing
### this configuration file.
### If SASL is enabled (see below), this file will NOT be used.
### Uncomment the line below to use the default password file.
# password-db = passwd
### The authz-db option controls the location of the authorization
### rules for path-based access control. Unless you specify a path
### starting with a /, the file's location is relative to the
### directory containing this file. The specified path may be a
### repository relative URL (^/) or an absolute file:// URL to a text
### file.

[ Read 76 lines ]
^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C
^X Exit          ^J Justify      ^W Where Is    ^V Next Page    ^U UnCut Text   ^T

```

Ilustración 60



```

GNU nano 2.2.6      File: /root/svnrepos/project01/conf/svnserve.conf

### This file controls the configuration of the svnserve daemon, if you
### use it to allow access to this repository.  (If you only allow
### access through http: and/or file: URLs, then this file is
### irrelevant.)

### Visit http://subversion.apache.org/ for more information.

[general]
### The anon-access and auth-access options control access to the
### repository for unauthenticated (a.k.a. anonymous) users and
### authenticated users, respectively.
### Valid values are "write", "read", and "none".
### Setting the value to "none" prohibits both reading and writing;
### "read" allows read-only access, and "write" allows complete
### read/write access to the repository.
### The sample settings below are the defaults and specify that anonymous
### users have read-only access to the repository, while authenticated
### users have read and write access to the repository.
# anon-access = read
auth-access = write
### The password-db option controls the location of the password
### database file.  Unless you specify a path starting with a /,
### the file's location is relative to the directory containing
### this configuration file.
### If SASL is enabled (see below), this file will NOT be used.
### Uncomment the line below to use the default password file.
password-db = passwd
### The authz-db option controls the location of the authorization
### rules for path-based access control.  Unless you specify a path
### starting with a /, the file's location is relative to the
### directory containing this file.  The specified path may be a
### repository relative URL (^/) or an absolute file:// URL to a text

```

[^]G Get Help [^]O WriteOut [^]R Read File [^]V Prev Page [^]K Cut Text
[^]X Exit [^]J Justify [^]W Where Is [^]N Next Page [^]U UnCut Text

To direct input to this virtual machine, press Ctrl+G.

Ilustración 61

- Agregamos usuarios a svn.

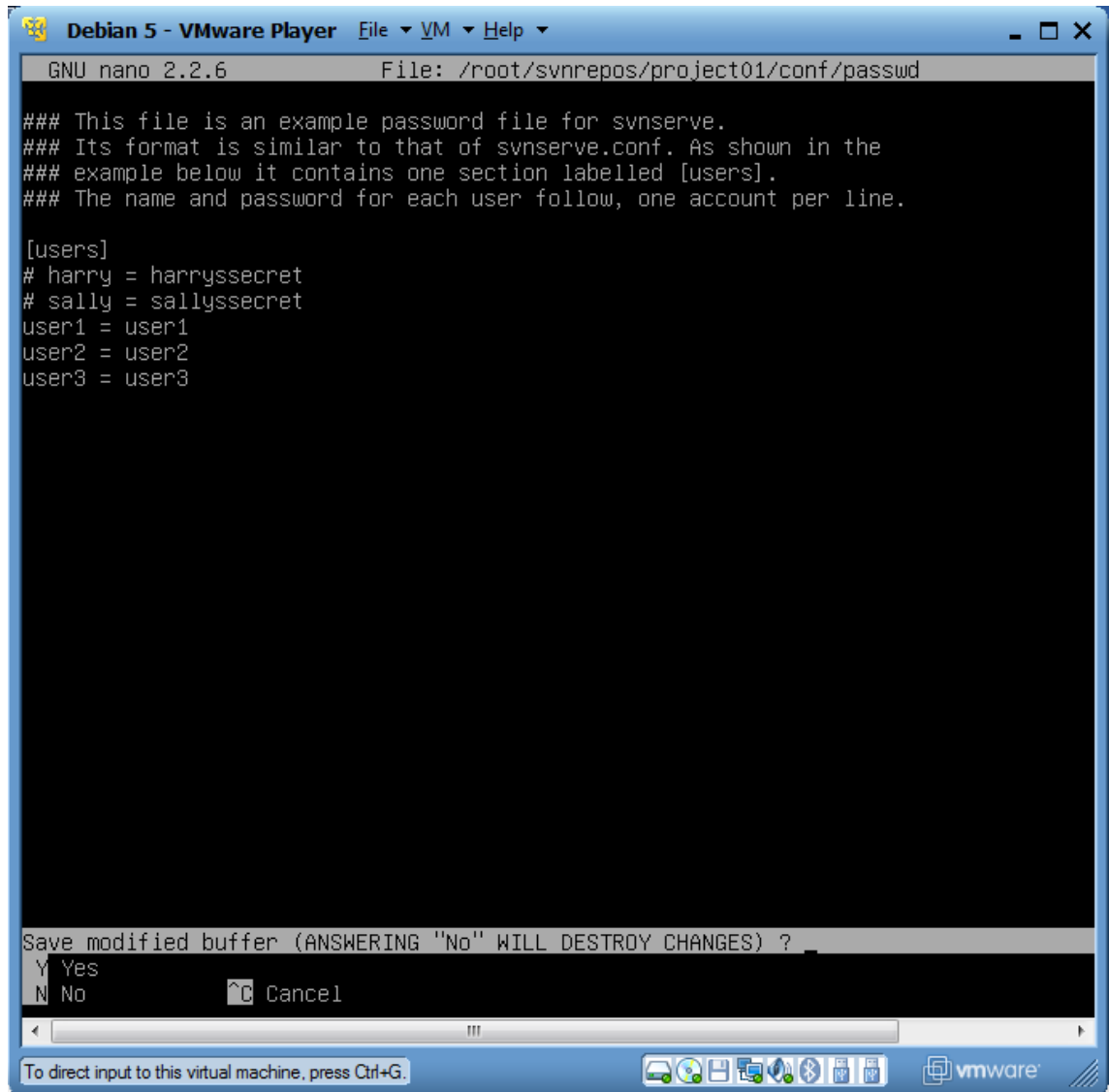
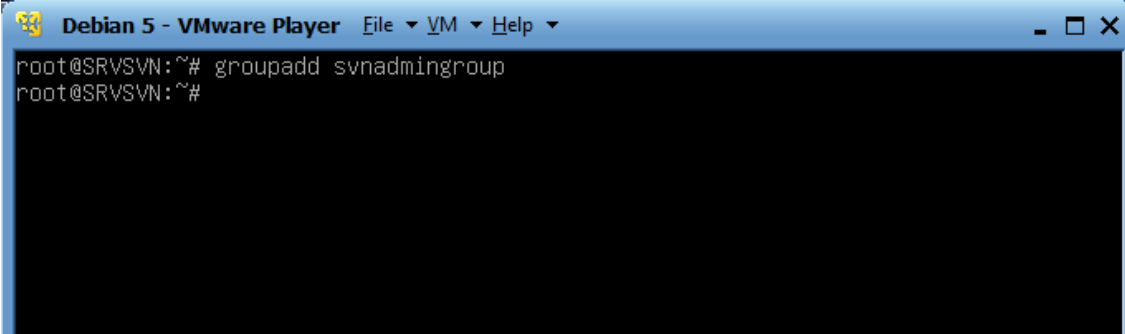


Ilustración 62

- Se crea un grupo administrador para svn y se insertan los usuarios
-

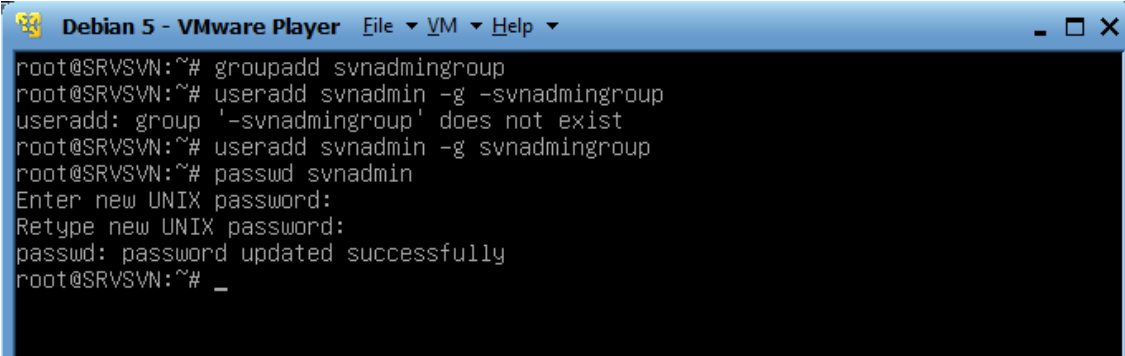


```

Debian 5 - VMware Player  File  VM  Help
root@SRVSVN:~# groupadd svnadmingroup
root@SRVSVN:~#

```

Ilustración 63



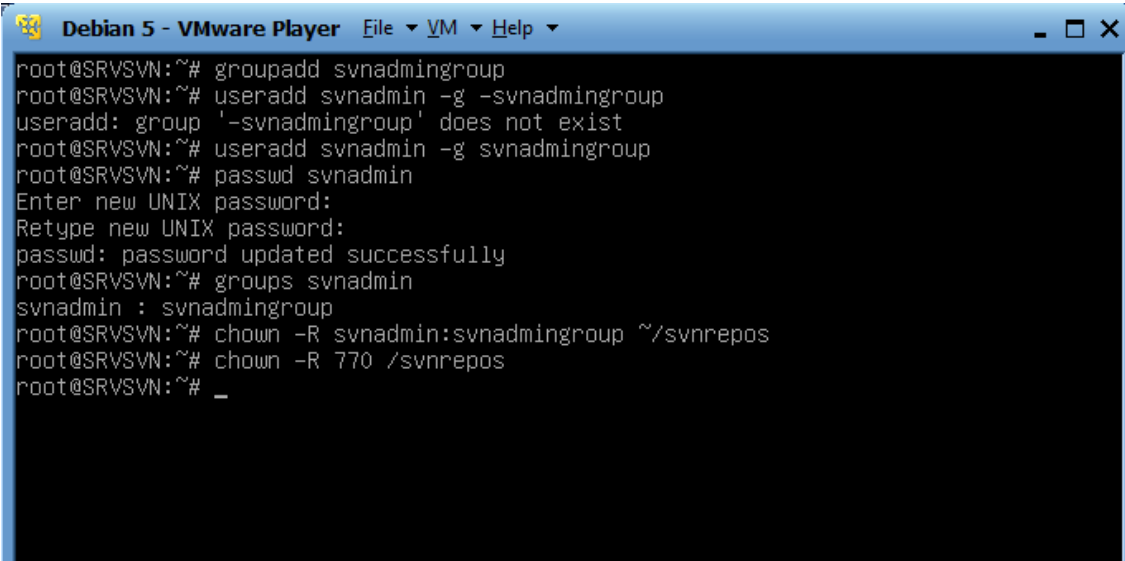
```

Debian 5 - VMware Player  File  VM  Help
root@SRVSVN:~# groupadd svnadmingroup
root@SRVSVN:~# useradd svnadmin -g -svnadmingroup
useradd: group '-svnadmingroup' does not exist
root@SRVSVN:~# useradd svnadmin -g svnadmingroup
root@SRVSVN:~# passwd svnadmin
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@SRVSVN:~# _

```

Ilustración 64

- Cambiamos el propietario del repositorio



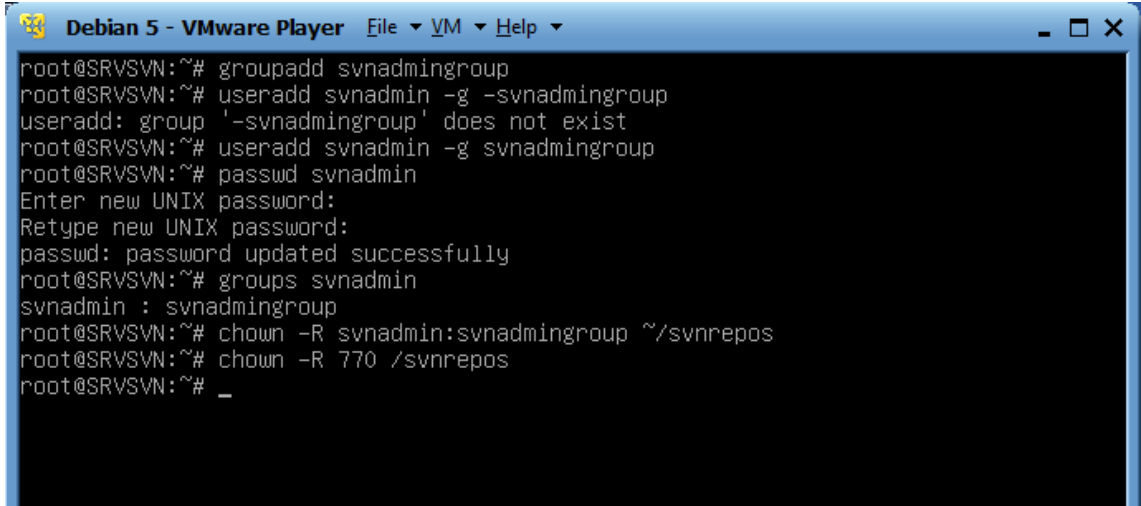
```

Debian 5 - VMware Player  File  VM  Help
root@SRVSVN:~# groupadd svnadmingroup
root@SRVSVN:~# useradd svnadmin -g -svnadmingroup
useradd: group '-svnadmingroup' does not exist
root@SRVSVN:~# useradd svnadmin -g svnadmingroup
root@SRVSVN:~# passwd svnadmin
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@SRVSVN:~# groups svnadmin
svnadmin : svnadmingroup
root@SRVSVN:~# chown -R svnadmin:svnadmingroup ~/svnrepos
root@SRVSVN:~# chown -R 770 /svnrepos
root@SRVSVN:~# _

```

Ilustración 65

- Para hacer que la seguridad vaya por SSH creamos las llaves y damos de altas los usuarios



```

Debian 5 - VMware Player  File  VM  Help
root@SRVSVN:~# groupadd svnadmingroup
root@SRVSVN:~# useradd svnadmin -g -svnadmingroup
useradd: group '-svnadmingroup' does not exist
root@SRVSVN:~# useradd svnadmin -g svnadmingroup
root@SRVSVN:~# passwd svnadmin
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@SRVSVN:~# groups svnadmin
svnadmin : svnadmingroup
root@SRVSVN:~# chown -R svnadmin:svnadmingroup ~/svnrepos
root@SRVSVN:~# chown -R 770 /svnrepos
root@SRVSVN:~# _
  
```

Ilustración 66

8.6 Instalación y configuración del dominio y los usuarios para poder acceder desde internet

Debido a problemas de disponibilidad de recursos esta parte del proyecto fue aplazada, aun así, se pudo montar un servidor provisional para poder realizar las pruebas necesarias.

8.7 Instalación de los clientes en los PC de trabajo

- I. Descargamos la versión 1.8.9 para Windows de TortoiseSVN



TortoiseSVN-1.8.9.26117-win32-svn-1.8.11.msi

Ilustración 67

- II. Doble clic en el icono para poder instalarlo

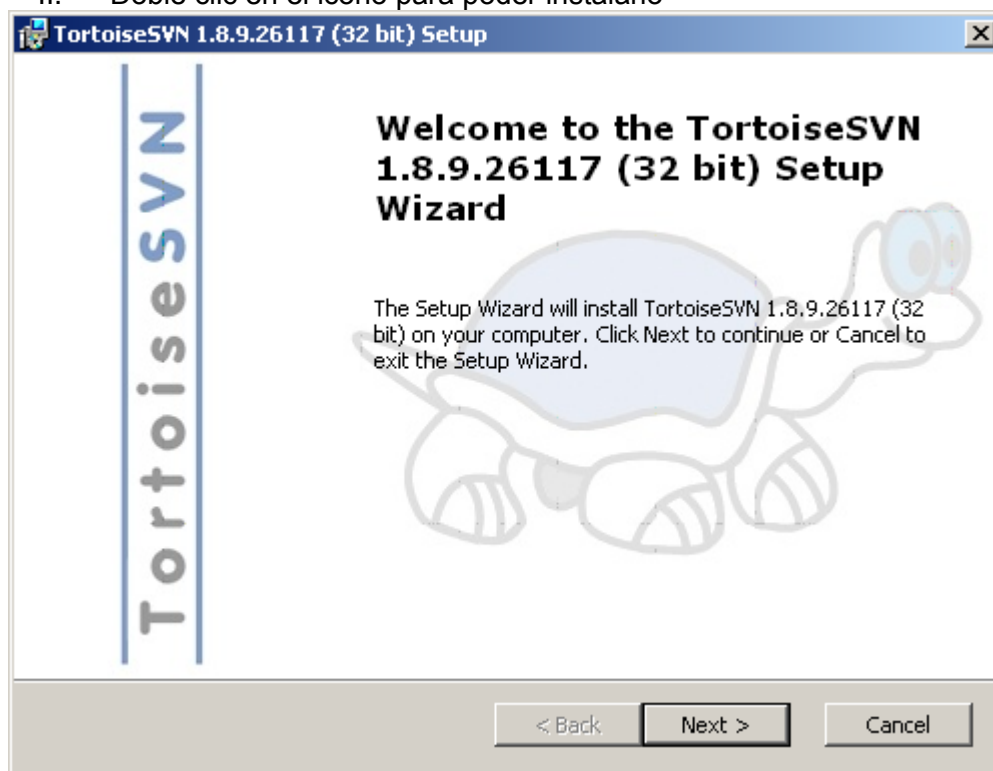


Ilustración 68 Instalando TortoiseSVN

III. Leemos y aceptamos los términos.

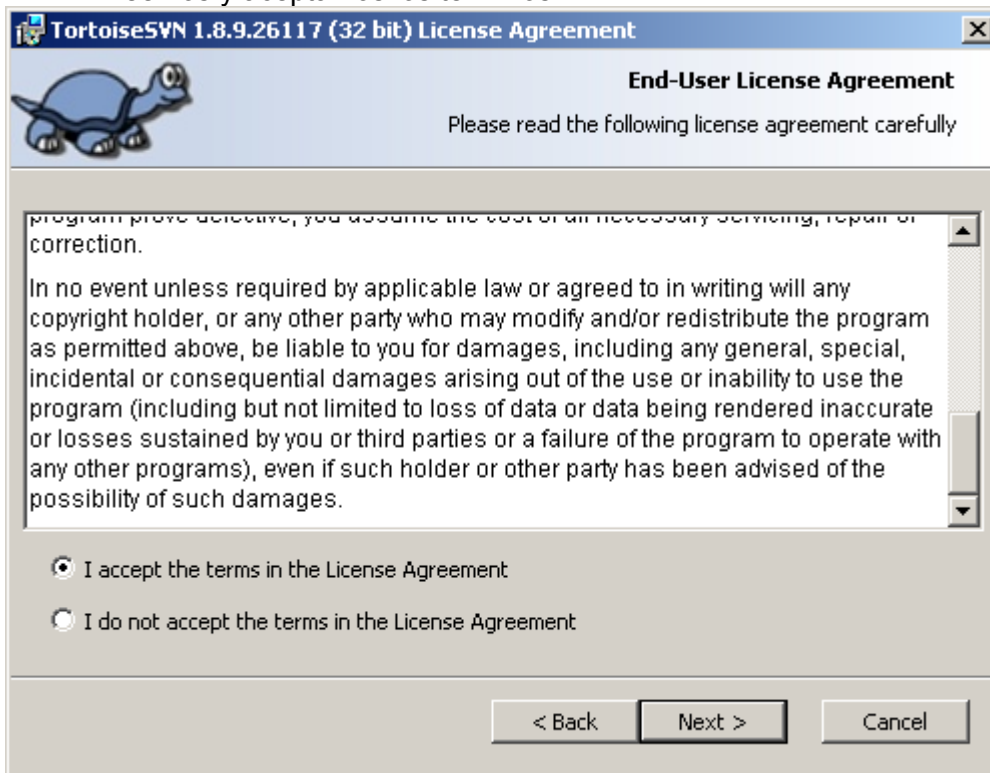


Ilustración 69

IV. Instalando

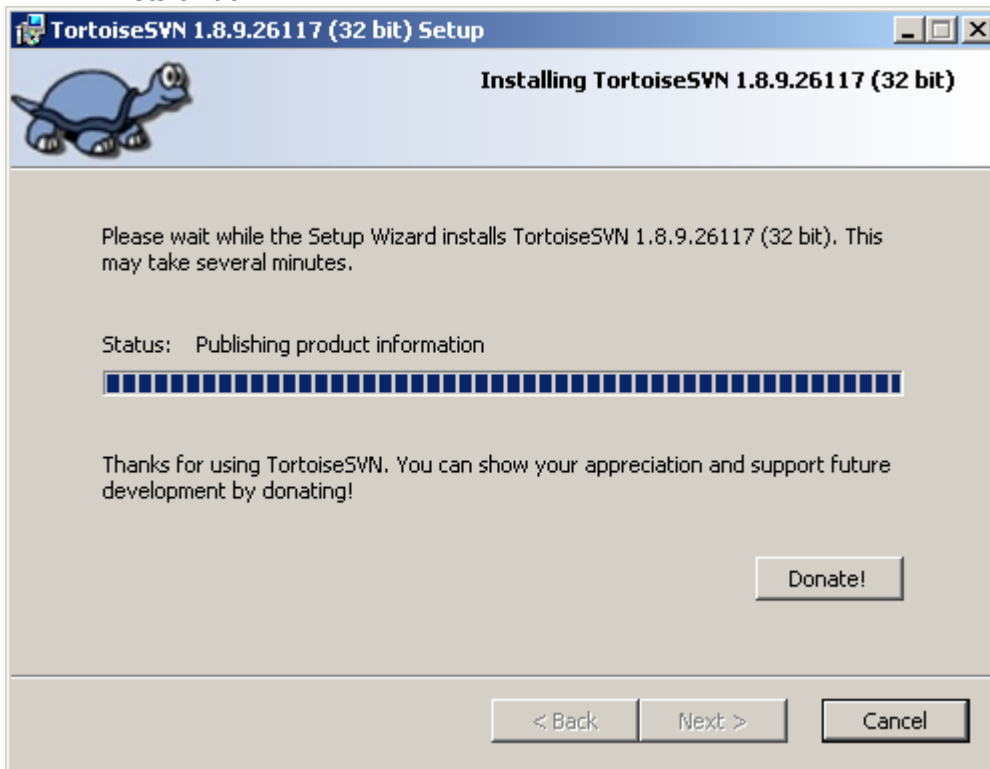


Ilustración 70

V. Una vez instalado ya lo tenemos en el sistema

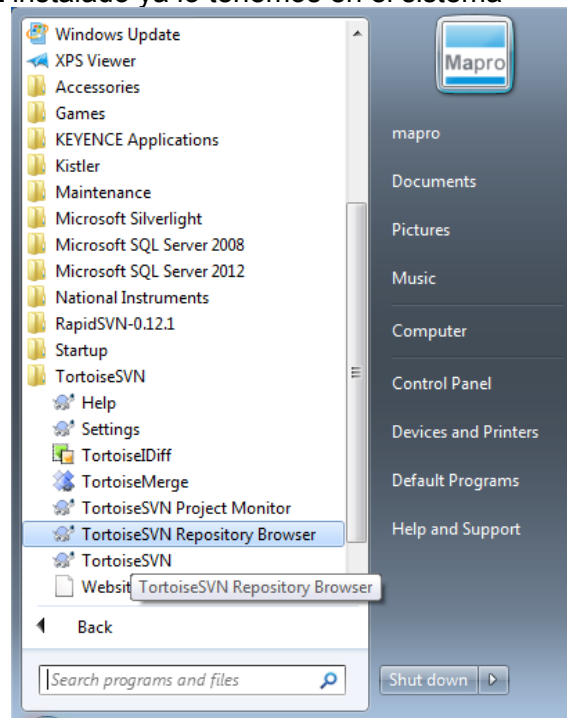


Ilustración 71 Menú Tortoise en Windows

8.8 Estructuración de proyectos en el servidor

Para poder tener bien organizados los proyectos en el servidor y no hacer copias de seguridad en lugares que no corresponden es necesario disponer de una estructuración sólida y organizada, por ello se propone una estructuración estándar para cada proyecto.

- Proyecto (MXXXX-000X)
 - Software PC
 - Software de producción
 - Recuperadores de resultados
 - Editores de parámetros
 - Software PLC
 - Programa de PLC
 - Pantallas HMI
 - Servomotores

- Devices
 - Camaras Visión
 - Prensas
 - Lectores de código de barras
 - Amplificadores Células (par o fuerza)
 - Impresoras
 - Marcadoras Lacers

8.9 Estructuración de las carpetas de proyecto en los PC clientes

Los PC clientes básicamente se van a dividir en dos, los que se utilizan para los proyectos de análisis de datos (se programa en PC) y los que se utilizan para la programación de PLC, servomotores y demás dispositivos.

Software de PC

- Software PC
 - Software de producción
 - Recuperadores de resultados
 - Editores de parámetro
- Devices
 - Camaras Visión
 - Lectores de código de barras
 - Amplificadores Células (par o fuerza)
 - Impresoras

Software de PLC

- Software PLC
 - Programa de PLC
 - Pantallas HMI
 - Servomotores
- Devices
 - Camaras Visión
 - Prensas
 - Lectores de código de barras
 - Amplificadores Células (par o fuerza)
 - Impresoras
 - Marcadoras Laser

8. Primeras pruebas

Una vez se ha instalado el servidor de empresa para poder albergar el sistema de control de versiones y todas las versiones de cada uno de los proyectos procedemos a hacer un primer Batch de pruebas para comprobar la funcionalidad del mismo con cada una de las plataformas con las que se trabaja en la empresa tanto a nivel de PC como de PLC.

9.1 Antes de comenzar a versionar código.

Para empezar el primer uso, debemos configurar el servidor en el cual vamos a hacer el versionado del código, para ello una vez instalado el paquete de “TORTOISESVN” en el PC de trabajo, vamos a ir al menú inicio (En estos momentos el 100% de plantilla que trabaja con Windows debido a la compatibilidad con el software de programación de los diversos dispositivos) y seleccionamos “TortoiseSVN Repository Browser” del menú “TortoiseSVN”, ver ilustración 64.

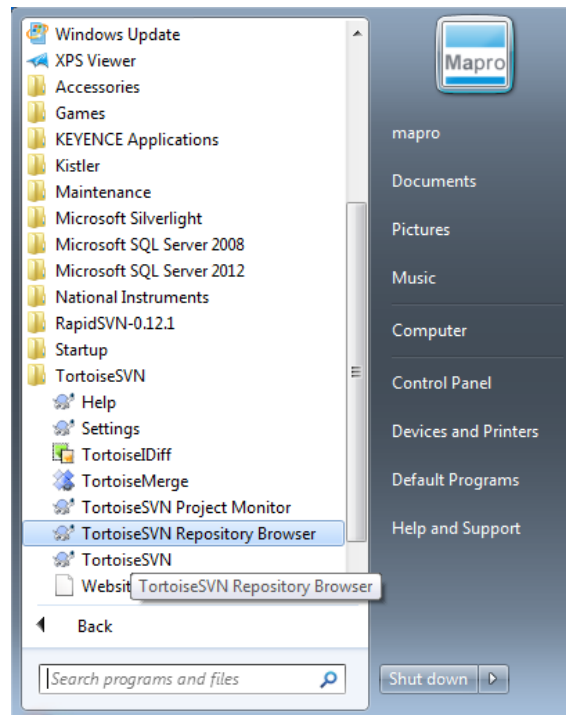


Ilustración 72. Menú Tortoise en Windows

Una vez clicado, se abre la siguiente ventana, ver ilustración 65. en la cual debemos dar la ruta del servidor donde vamos a colocar nuestro versionado de código.

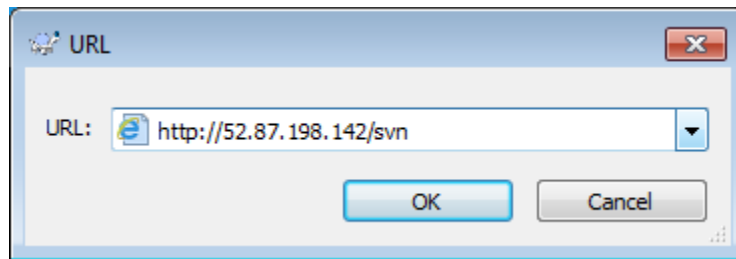


Ilustración 73 Insertar URL del servidor

Una vez insertado la ruta/ULR del servidor se abre la siguiente venta en la cual tenemos al servidor con las carpetas que hay creadas y que tenemos acceso ver ilustración 66.

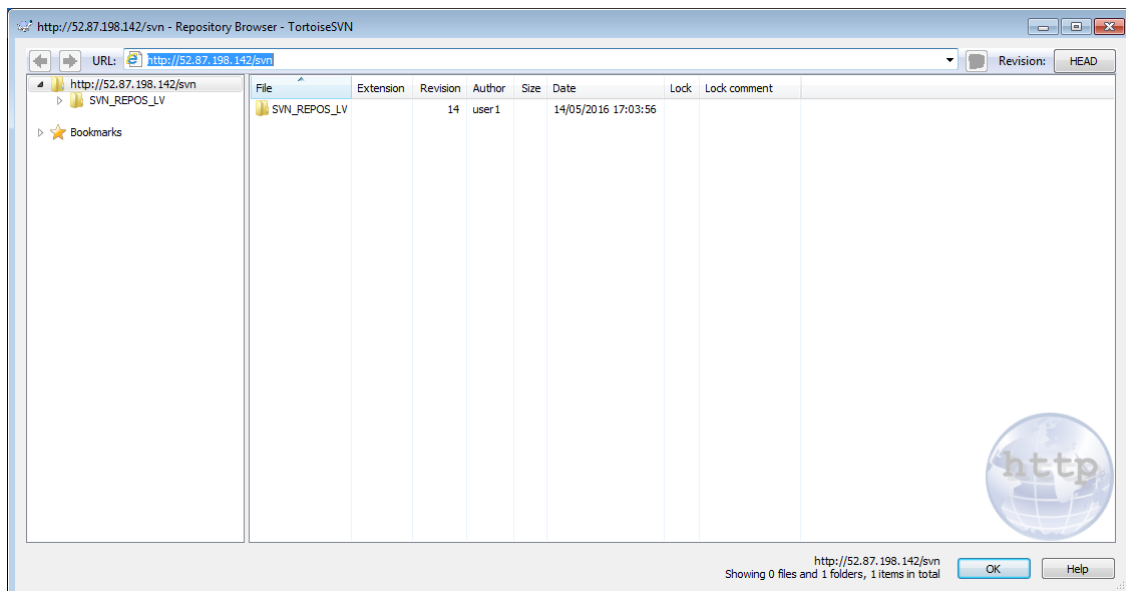


Ilustración 74 vista del servidor SVN.

Una vez realizados estos sencillos pasos ya podemos empezar a trabajar con nuestro servidor y sus diferentes directorios y opciones.

9.2 CVI/Labwindows.

La programación con labwindows está basada en “ANSI C” y orientada a la adquisición y tratamiento de datos ya sean analógicos o digital además de tener las librerías estándares de ANSI C tiene librerías propias, el entorno de programación el IDE ya cuenta con un comparador de código, aunque es bastante manual ya que hay que saber exactamente que módulos se han modificado para poder hacer la comparación es muy útil por las funciones que tiene, el objetivo de tener este tipo de software en un SCV no solo es para tener control de las versiones y copias seguras sino también el de automatizar los procesos de comparación de código de esta manera no será necesario conocer los cambios que han sufrido los códigos ya que el SCV ya se encargará de hacernos saber que módulos han cambiado.

En el caso en que no se tenga la carpeta deseada del proyecto en el que estamos trabajando, deberemos crear una carpeta para poder entrar el versionado de nuestro software, en este caso se seguirán las recomendaciones del capítulo “4.4. Estructuración de proyectos en el servidor” para ello y puesto que no tenemos la carpeta creada vamos a crear una carpeta contenedora del proyecto.

1. Creación de la carpeta, en el “Repository Browser” de “TortoiseSVN” hacemos clic derecho en el servidor y seleccionamos “Create Folder” en el submenú que aparecerá en pantalla ver ilustración 67.

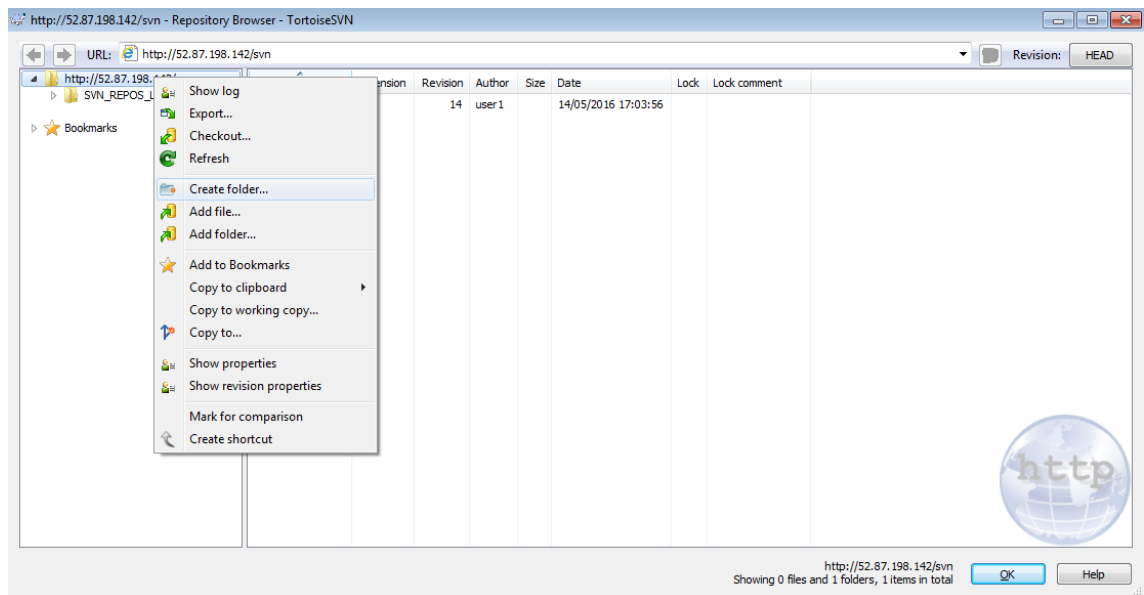


Ilustración 75 Creando una carpeta nueva

Introducimos el nombre de nuestra carpeta, (en este caso es el nombre interno del proyecto), ver ilustración 68

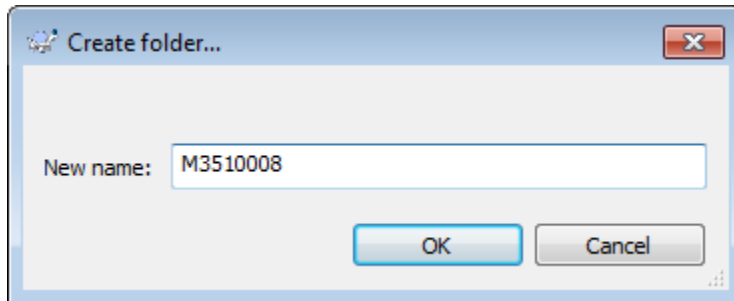


Ilustración 76 Nombre de la carpeta.

Una vez tenemos el nombre de la carpeta se abre la siguiente ventana en la que nos permite introducir una descripción, en este caso al ser la carpeta del proyecto, introducimos la información relevante al proyecto ver ilustración 69.

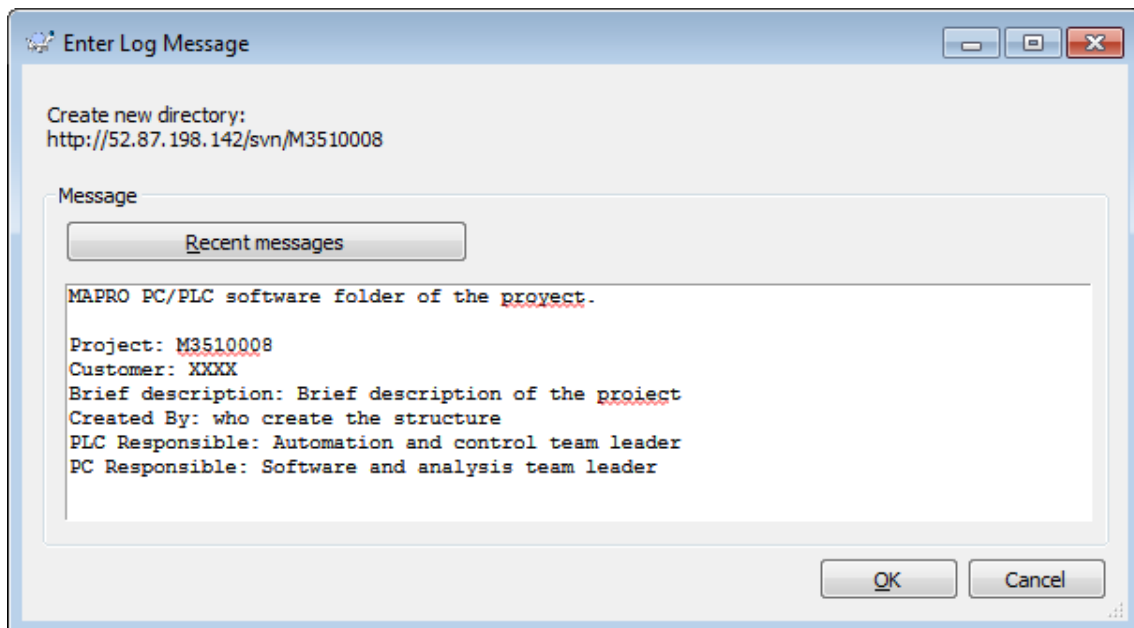


Ilustración 77 Descripción proyecto.

De esta manera y siguiendo el capítulo correspondiente creamos la estructura de carpetas.

2. Subiendo la primera versión.
Una vez creada la estructura vamos a subir nuestra primera versión de código. Para ello navegamos hasta la carpeta deseada en este caso será la de software de PC.

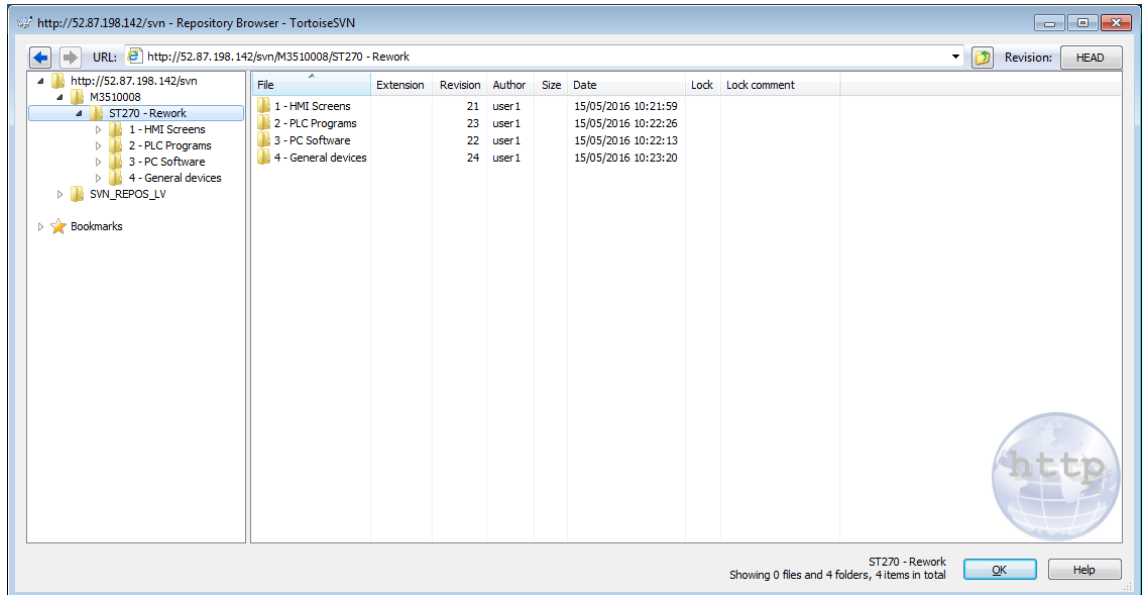


Ilustración 78 Vista de las carpetas en el servidor.

- Para subir la primera versión de nuestro código debemos seguir los siguientes pasos:
- I. Navegamos a la carpeta local donde tenemos nuestro software, una vez que lo tenemos a punto para poder hacer la primera versión de código.

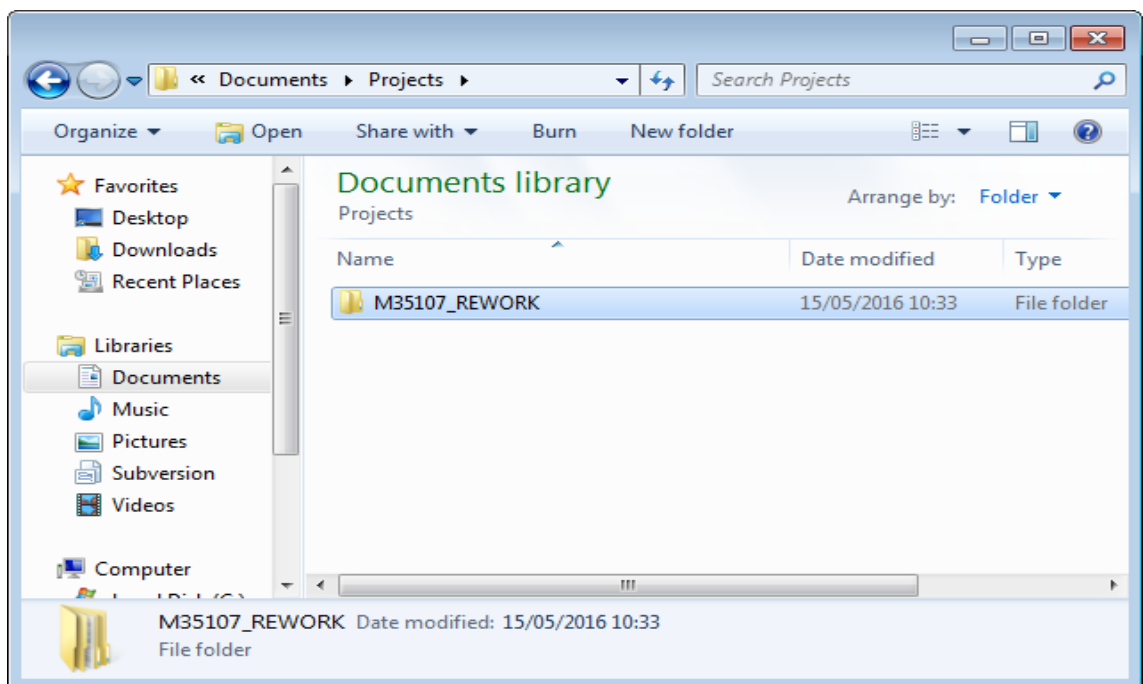


Ilustración 79 carpeta local del proyecto, carpeta de trabajo.

- II. Hacemos clic derecho sobre la carpeta. Y seleccionamos la opción “import” del submenú de tortoissvn

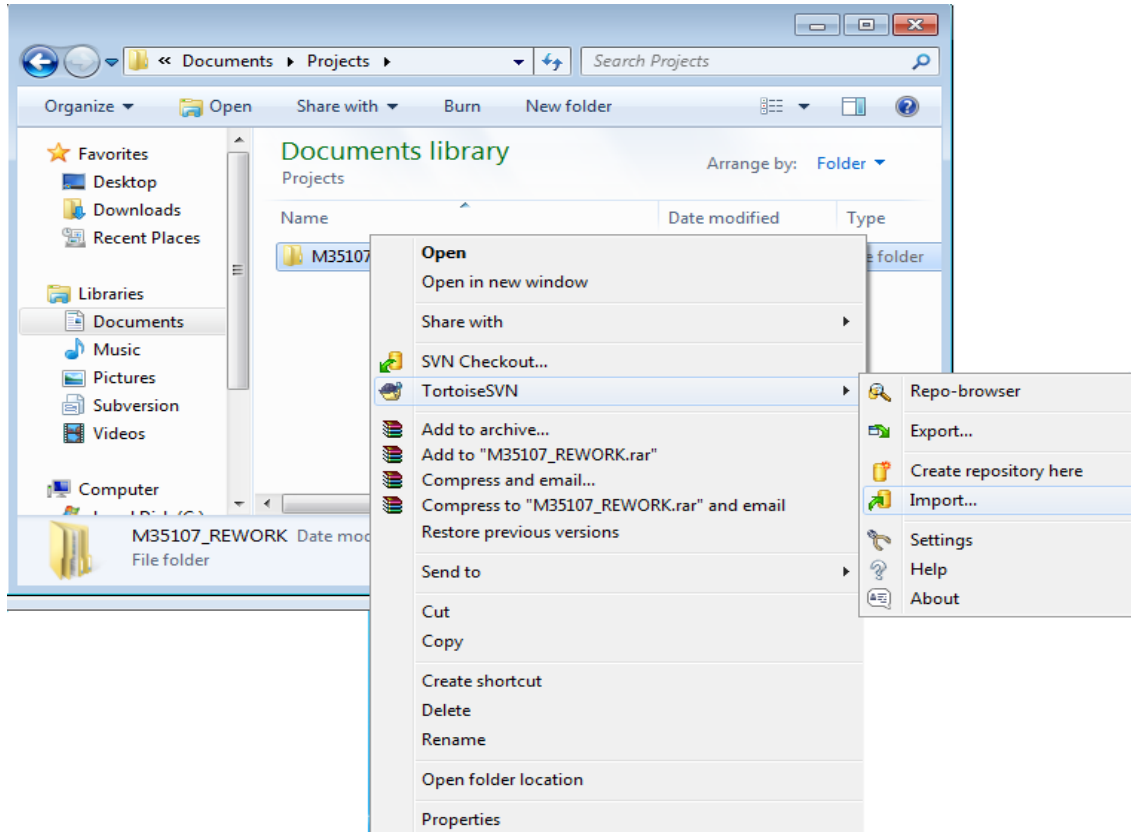


Ilustración 80 Import del submenú tortoissvn

- III. Seleccionamos la carpeta donde queremos que se versione nuestro código

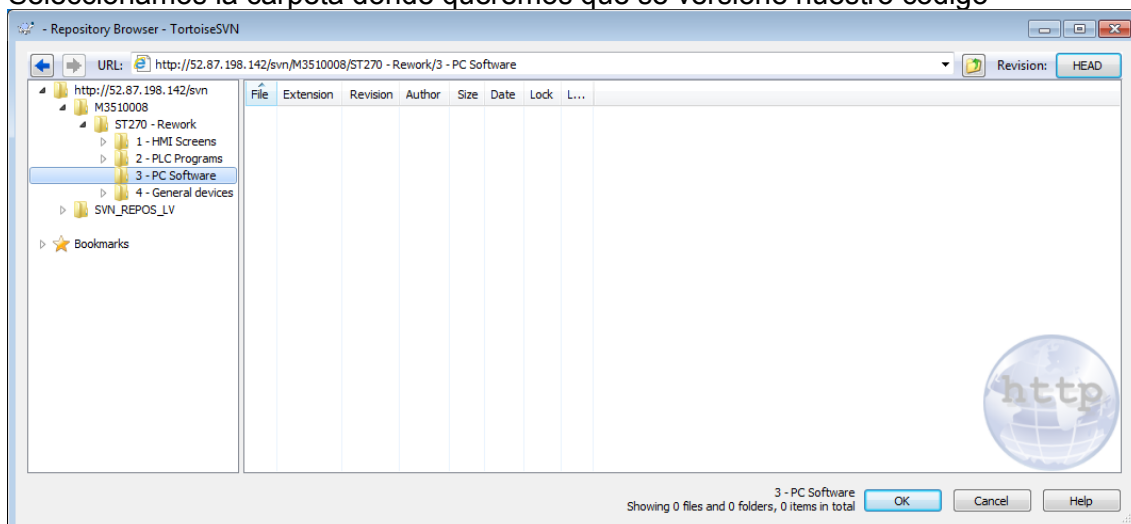


Ilustración 81 Selección de la carpeta donde vamos a versionar nuestro código.

Una vez seleccionada nos vuelve a aparecer la una ventana para poder insertar un descriptivo de la acción que realizamos

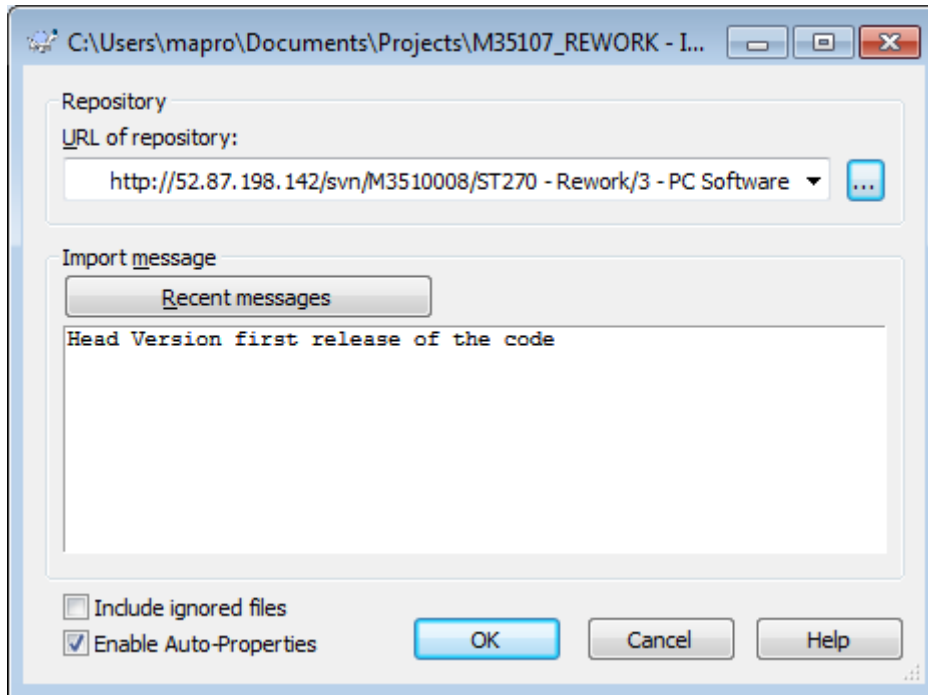


Ilustración 82

Una vez terminamos de subir el código aparece la siguiente pantalla:

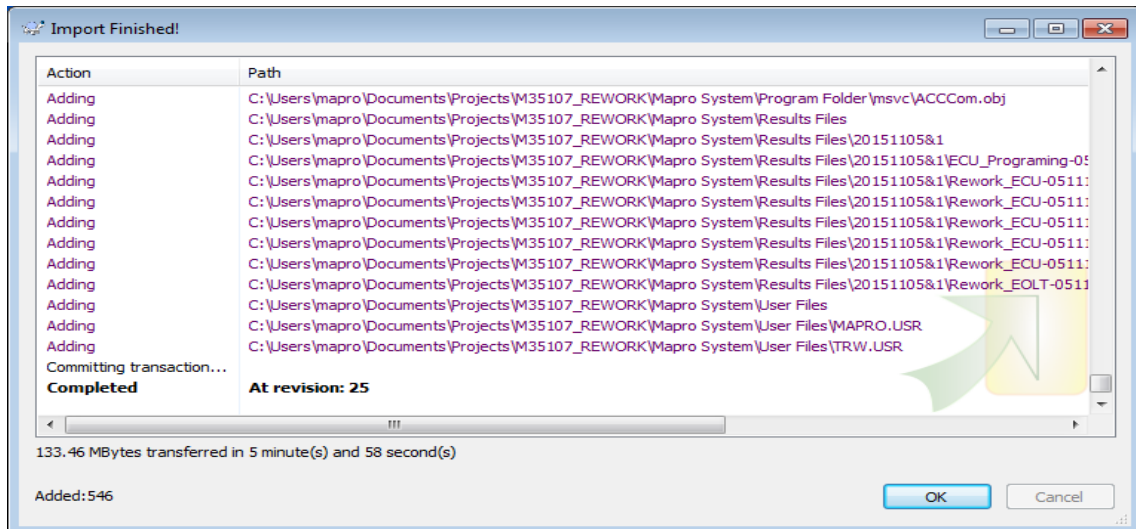


Ilustración 83

1. Haciendo un checkout del código.
Una vez tenemos el código subido debemos hacer un checkout del directorio para linkar las carpetas de trabajo con la del servidor, para ello navegamos hasta la carpeta del proyecto en local y con un clic derecho seleccionamos la opción de “checkout” del submenú.

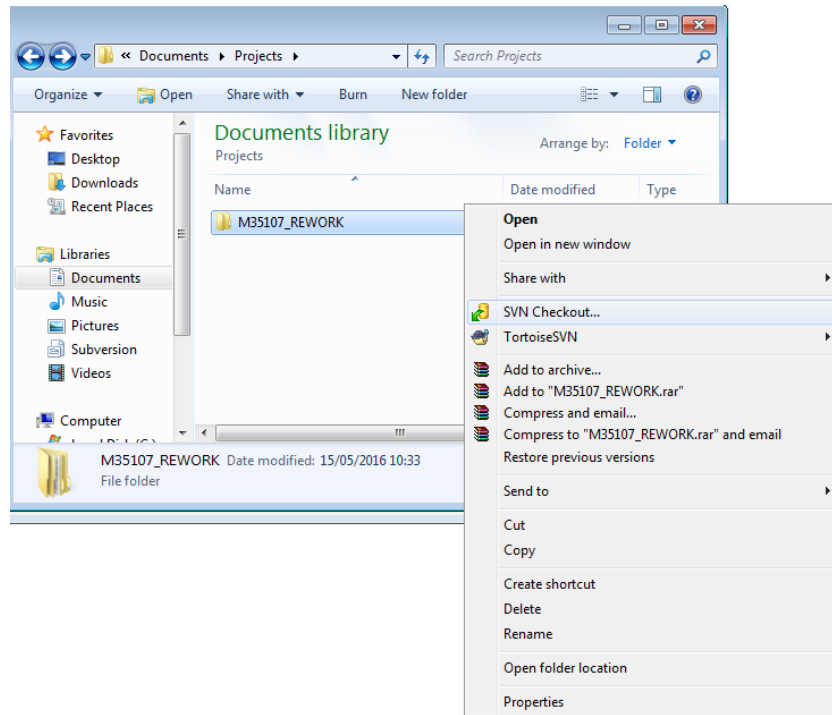


Ilustración 84 Submenú para selección del checkout.

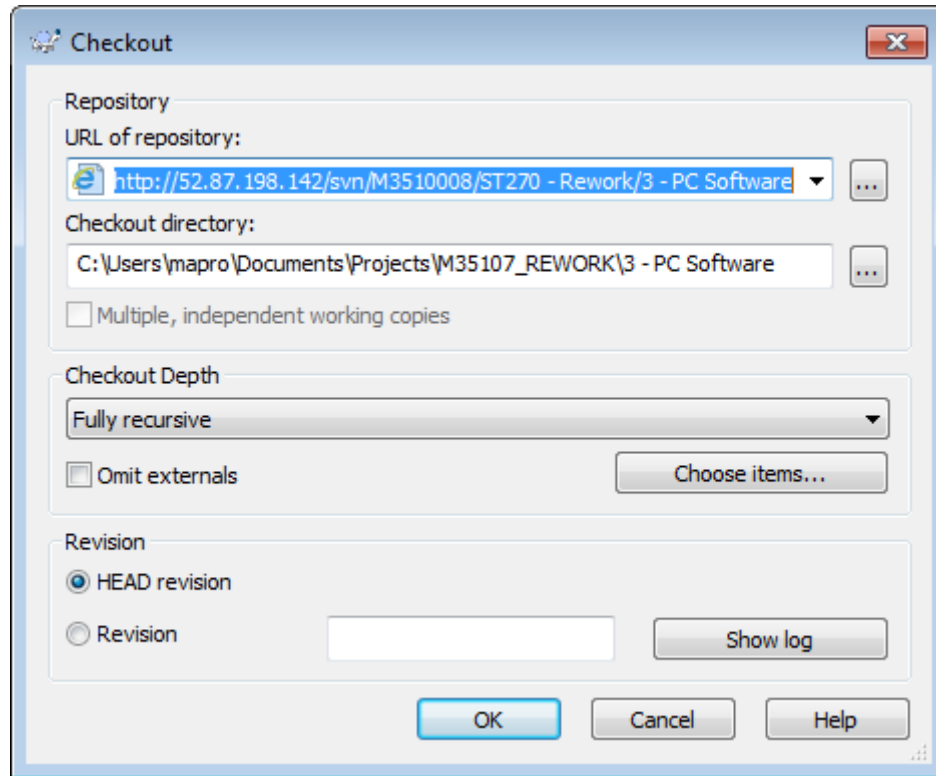


Ilustración 85 opciones del checkout.

Una vez lo tenemos linkado aparecerá un icono de directorio indicando que esta linkado y que los códigos son iguales.

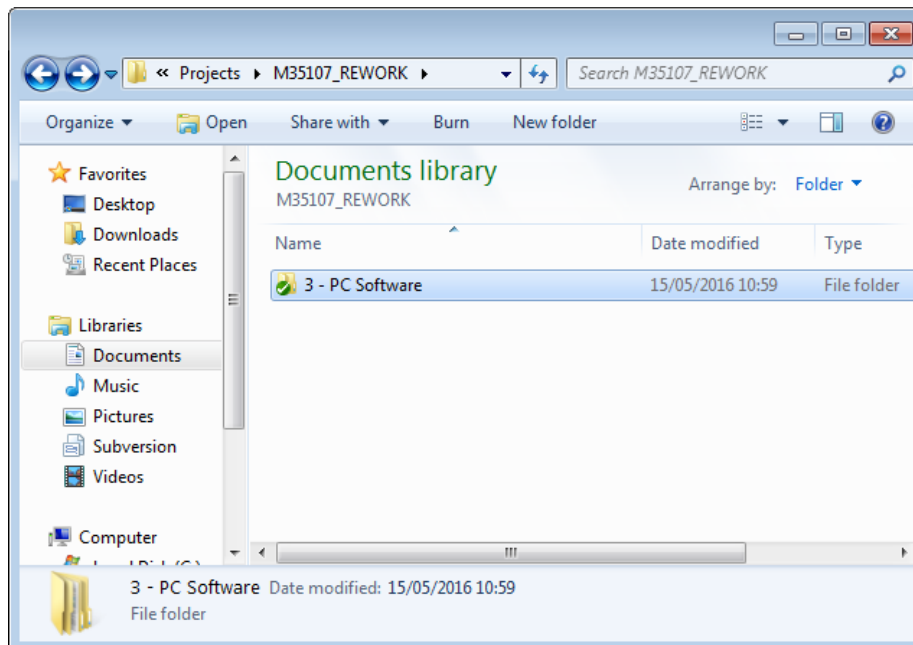


Ilustración 86 Carpeta linkada.

2. Modificaciones en el código.
 Modificamos el código en la carpeta de trabajo local y aparecerá un icono de color rojo en la carpeta del proyecto. Esto indica que hemos modificado el código.

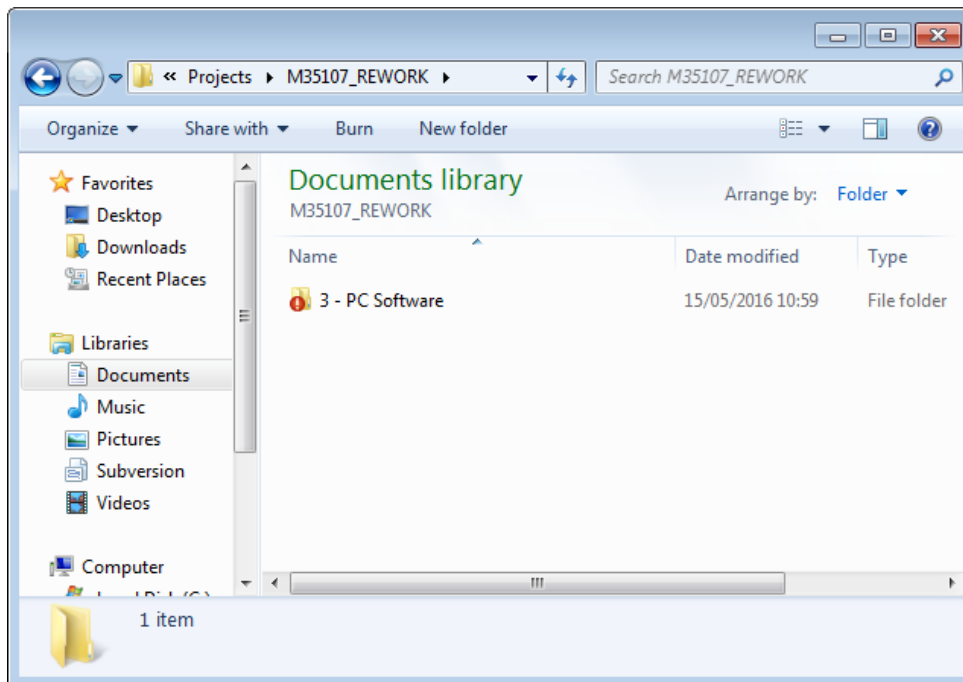


Ilustración 87 Código modificado.

Si tenemos ficheros de texto no solo nos indica el cambio en el código sino también en todos los ficheros modificados, indicándolo con un icono de color rojo.

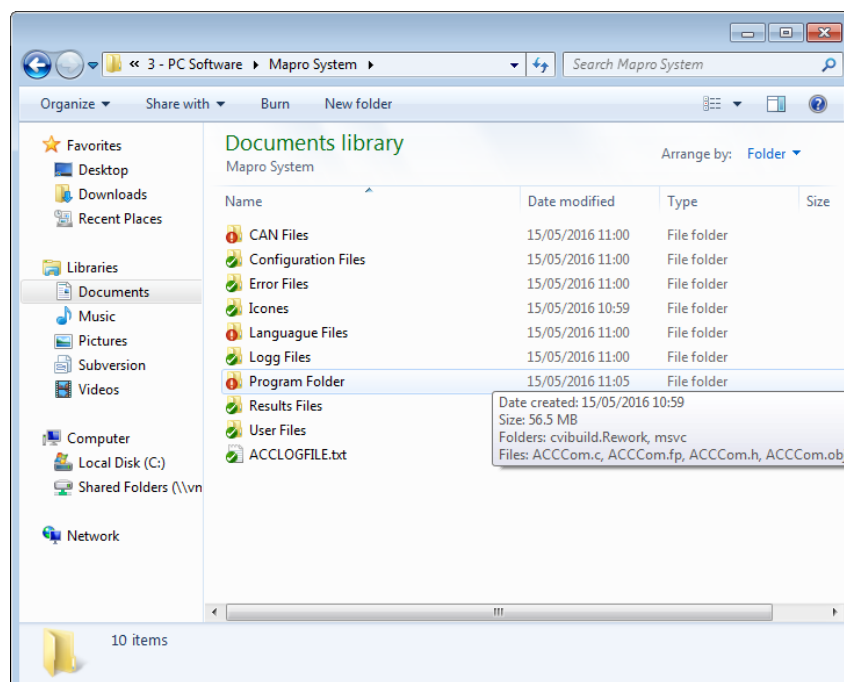


Ilustración 88 Carpetas que contienen código modificado

Llegados a este punto en el que sabemos que hay código/ficheros modificados podemos hacer una verificación de lo que es diferente en la versión reléase en el servidor y la del directorio de trabajo. Para ello navegamos hasta una carpeta o fichero que tenga diferencias y con un clic con el botón derecho navegamos hasta la opción “Check for Modification “de “TortoiseSVN”.

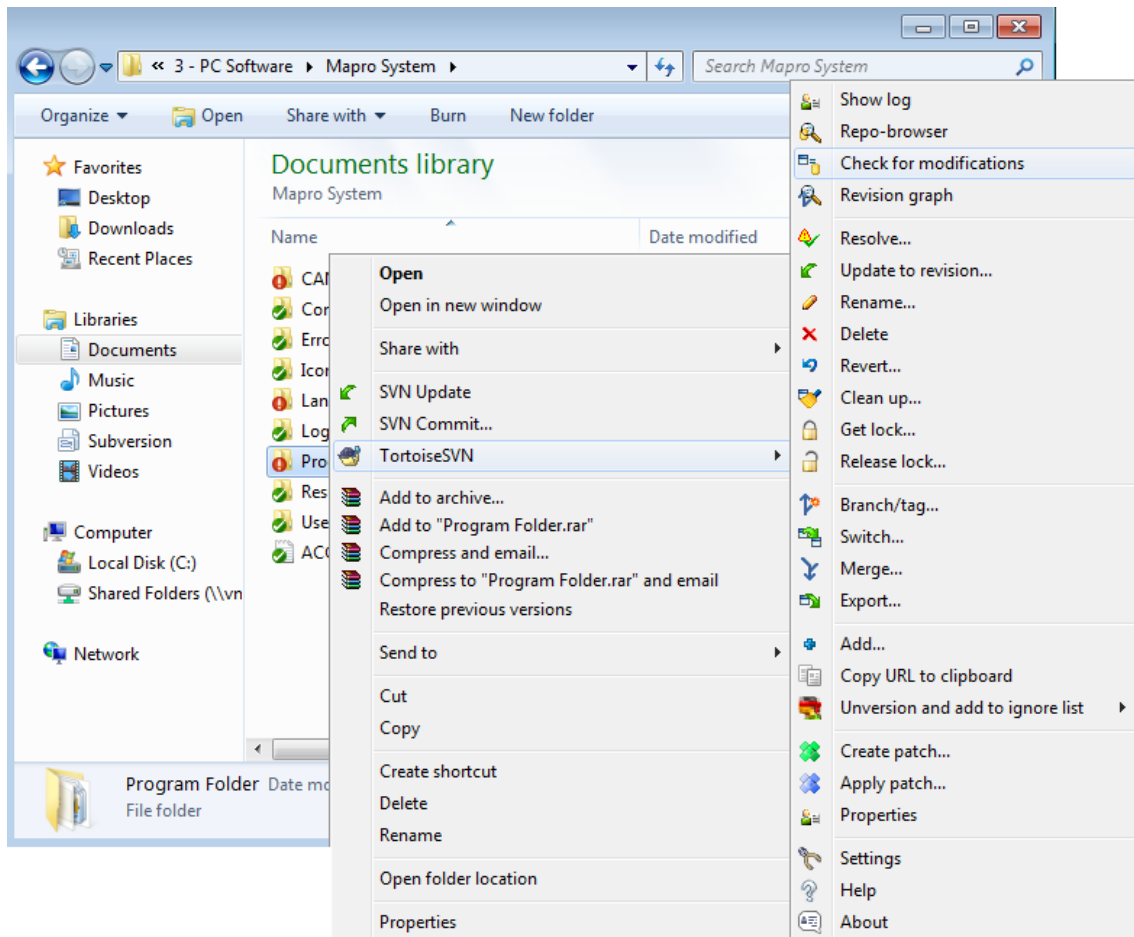


Ilustración 89 Check for modificación

Una vez verificadas las diferencias se abre una ventana que nos indica que ficheros fueron modificados.

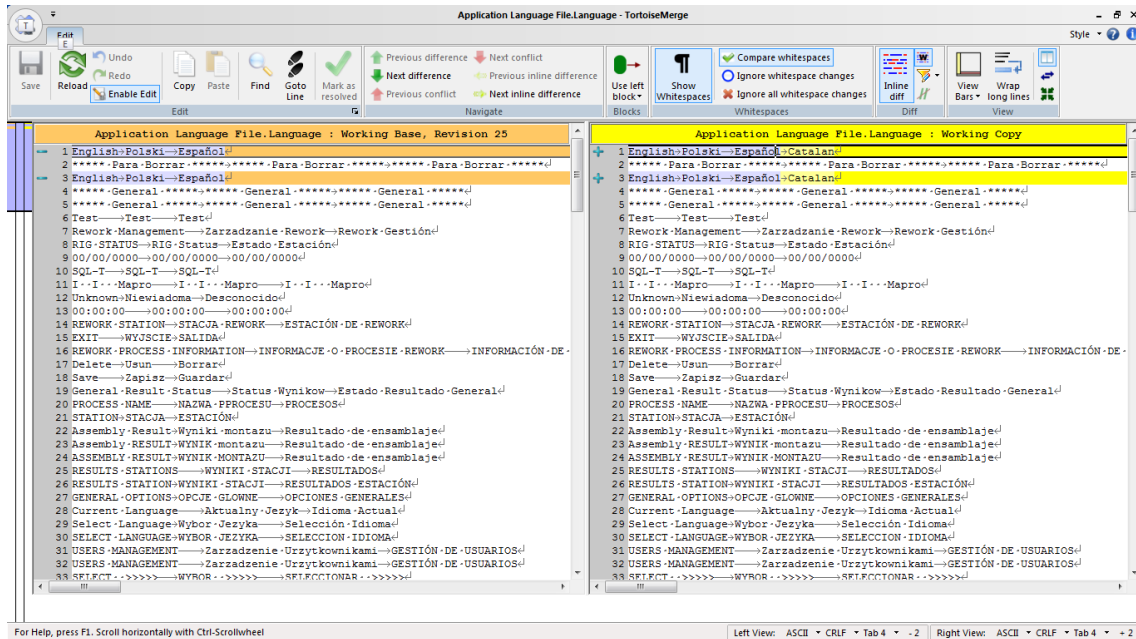


Ilustración 92 Vista de las diferencias en ficheros de configuración con la herramienta de tortoise.

Para poder utilizar la aplicación de comparación de nuestro IDE debemos hacer una pequeña configuración. Para ello abrimos las opciones de configuración de tortoiseSVN, vamos al menú inicio “TortoiseSVN”→”Settings”.

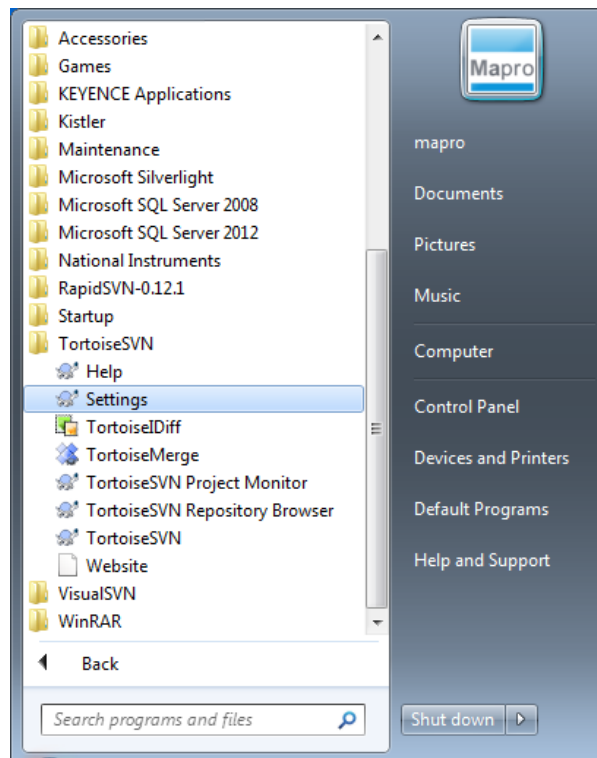


Ilustración 93 Navegando a Settings

Dentro de settings damos la ruta del software que queremos que realice las comparaciones:

- Vamos a Diff Viewer
- En la primera opción seleccionamos el software que queremos que realice las comparaciones de código.

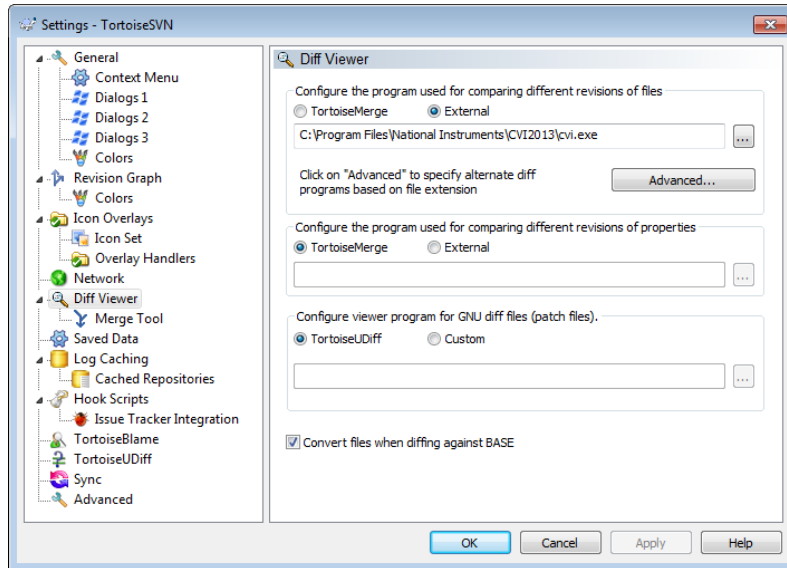


Ilustración 94 Navegando a Settings

En la carpeta deseada vamos al submenú TortoiseSVN y seleccionamos “Check For Modifications”, esta vez se abrirá el software específico que hemos configurado previamente con los dos ficheros abiertos. Uno será el versionado y el otro el modificado en local.

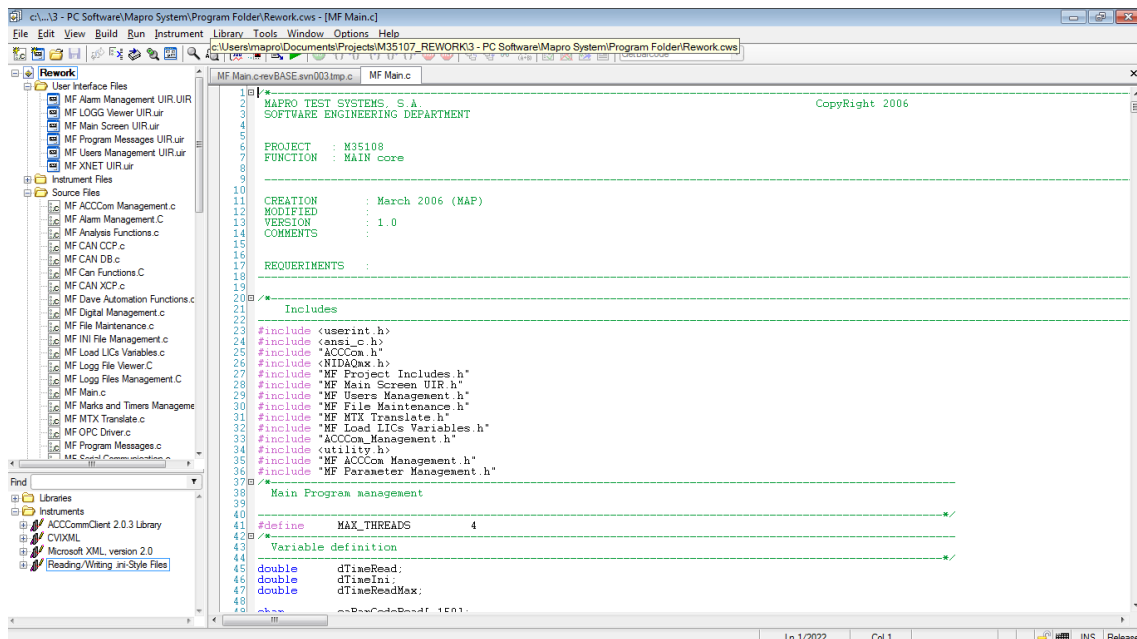


Ilustración 95 Ficheros abiertos en nuestra IDE de programación.

Para comparar utilizamos la propia herramienta del software utilizado, en este caso navegando al menú correspondiente podemos comparar el código.

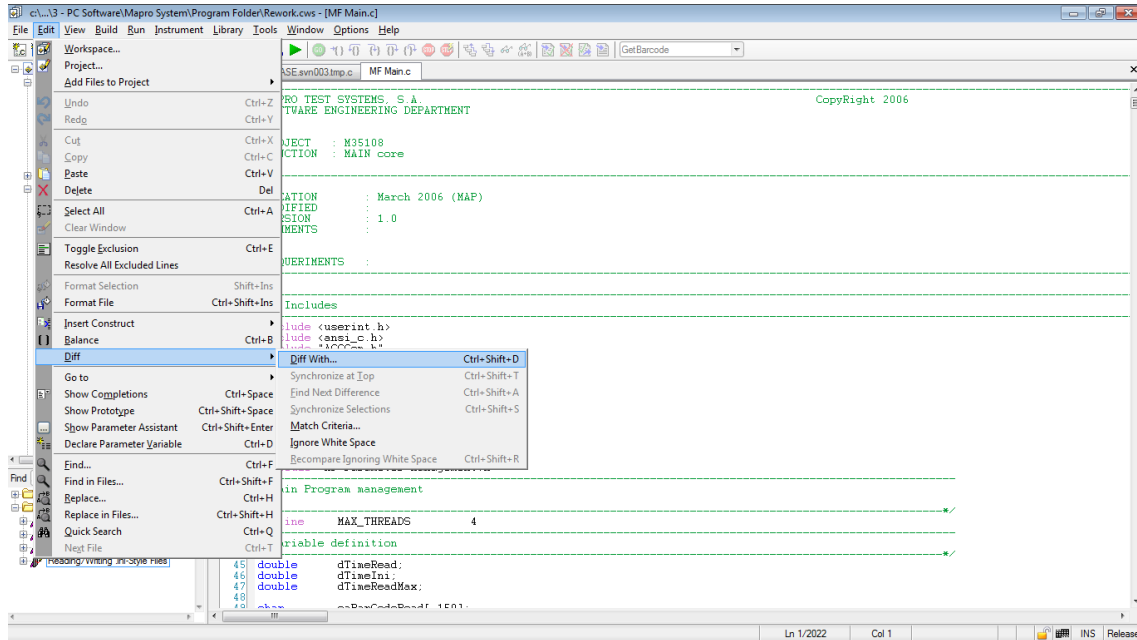


Ilustración 96 Utilizando la herramienta de comparación del IDE

En este caso y después de comparar apretando la combinación de teclas "ctrl+shift+A" vamos recorriendo el código y nos va indicando las diferencias.

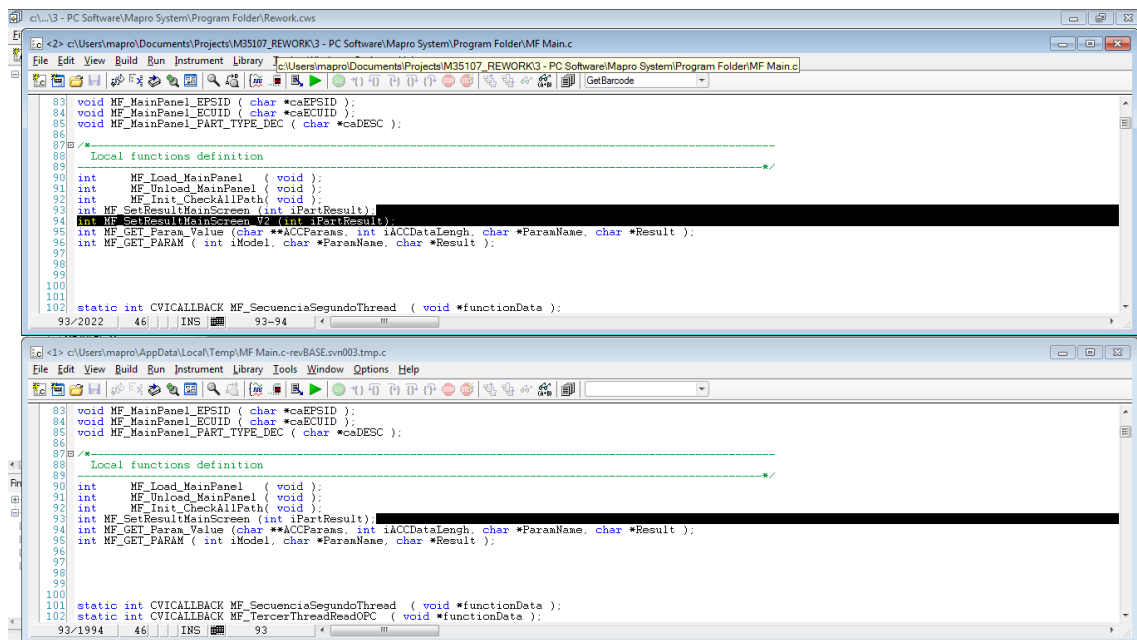


Ilustración 97 Comparación con el IDE

1. Re-versionado.

Una vez hechas las modificaciones deseadas en el código y estamos preparados para hacer un nuevo reléase para poder cargar una versión de nuestro código, para ello solo debemos realizar un sencillo paso que es navegar al submenú “SVN Commit” de tortoise

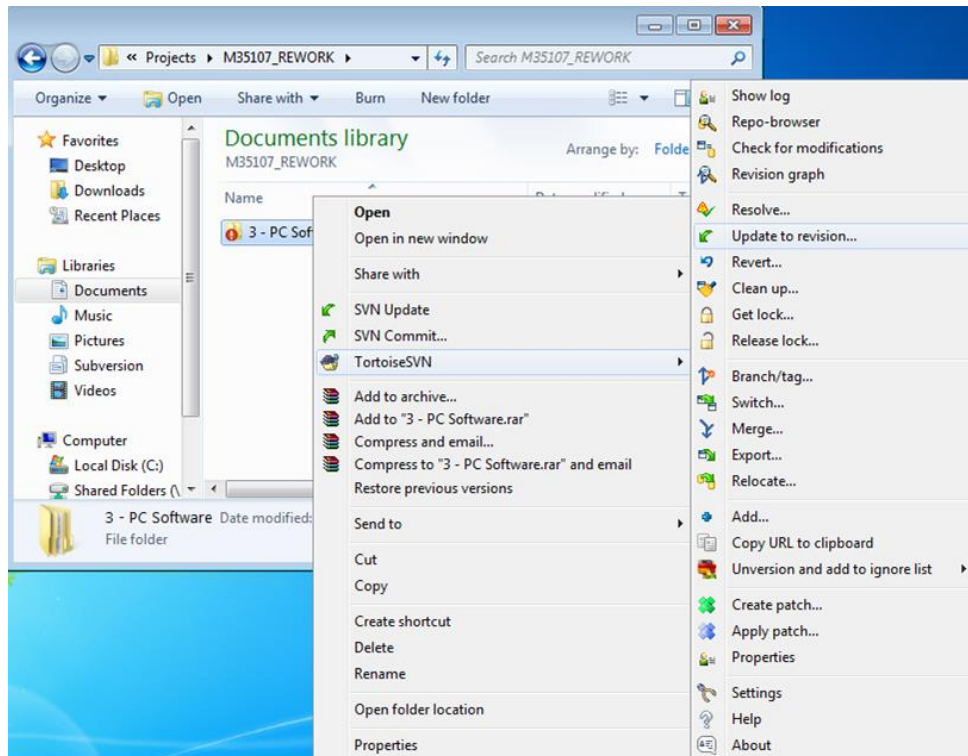


Ilustración 98 Update to revisión...

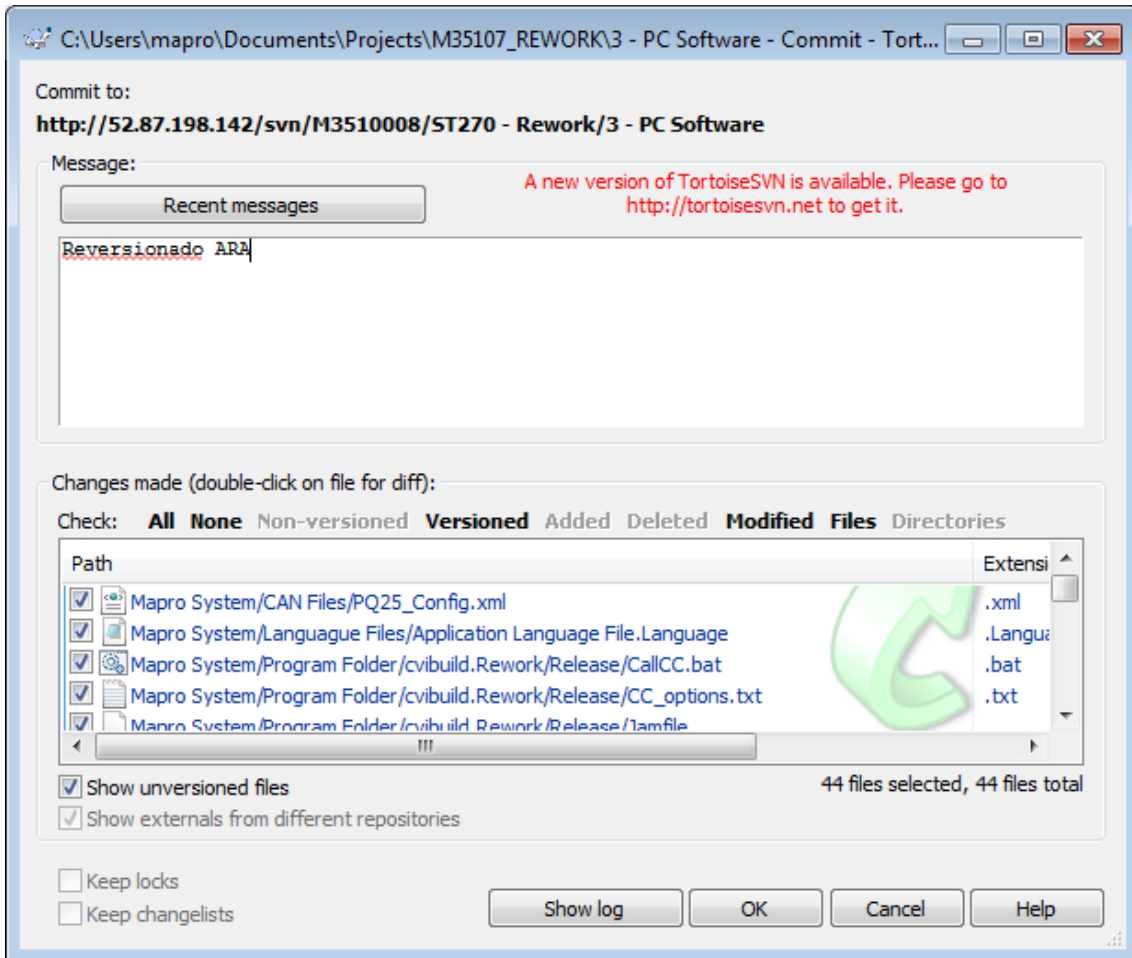
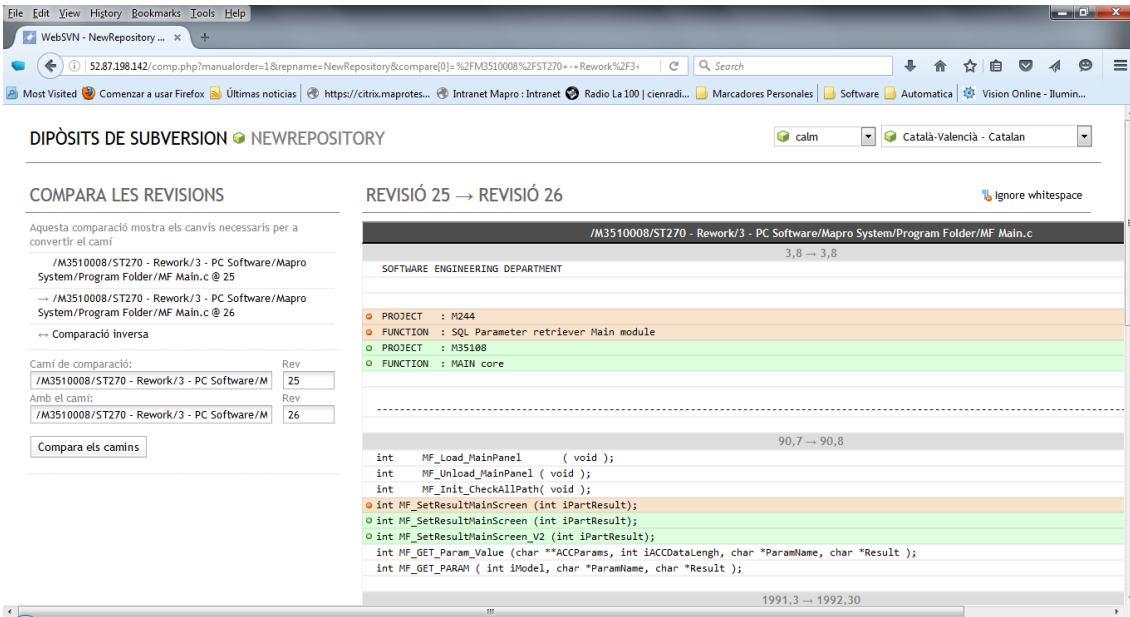


Ilustración 99 Opciones de re versionado

2. Vista web.

Una de las opciones de las que disponemos es la visualización web de las diferentes versiones en este caso puede realizarse ya que es texto plano y por ello podemos hacer la comparativa web, en el caso de que sea programación esquemática o en diagrama de contactos (PLC) esta opción queda relegada y lo óptimo es realizarlo con el propio software de comparación.



Il·lustració 100 Comparació web

9.3 LabView

La programació amb LabView està basada en un diagrama de fluxos i orientada a la adquisició i tractament de dades ja siguin analògics o digital, l'objectiu de tenir aquest tipus de software en un SCV no només és per tenir control de les versions i còpies segures sinó també el de automatitzar els processos de comparació de codi. D'aquesta manera no serà necessari conèixer els canvis que han patit els codis ja que el SCV se'n encarregarà de fer-nos saber quins mòduls han canviat.

En el cas en què no es tingui la carpeta desitjada del projecte en el que estem treballant, hauríem de crear una carpeta per poder entrar al versionat del nostre software, en aquest cas es seguiran les recomanacions del capítol "4.4. Estructuració de projectes en el servidor" per a això i ja que no tenim la carpeta creada anem a crear una carpeta contenedora del projecte.

1. Creación de la carpeta, en el “Repository Browser” de “TortoiseSVN” hacemos clic derecho en el servidor y seleccionamos “Create Folder” en el submenú que aparecerá en pantalla ver imagen 5.3.1.

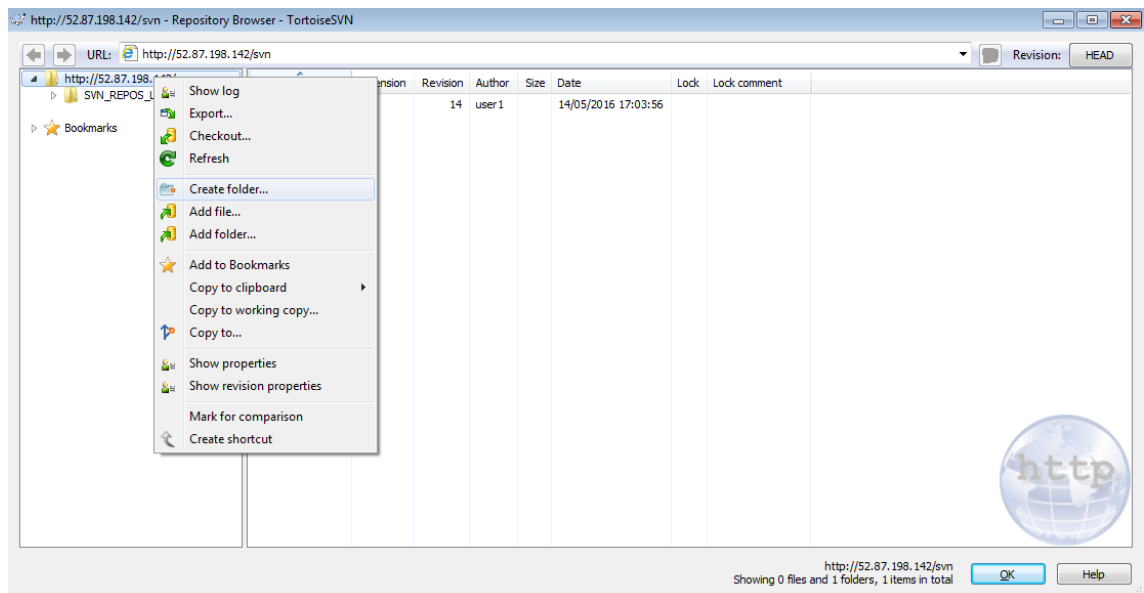


Ilustración 101 Creando una carpeta nueva

Introducimos el nombre de nuestra carpeta, (en este caso es el nombre interno del proyecto), ver imagen 5.3.2.

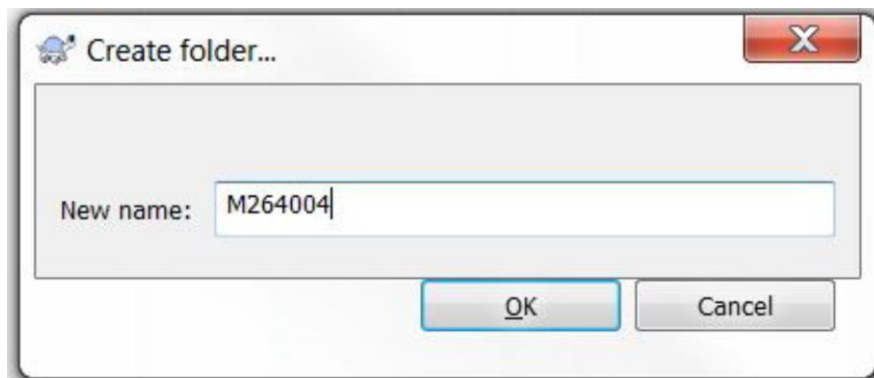


Ilustración 102 Nombre de la carpeta.

Una vez tenemos el nombre de la carpeta se abre la siguiente ventana en la que nos permite introducir una descripción, en este caso al ser la carpeta del proyecto, introducimos la información relevante al proyecto ver imagen 5.3.2.

De esta manera y siguiendo el capítulo correspondiente creamos la estructura de carpetas.

1. Subiendo la primera versión.
Una vez creada la estructura vamos a subir nuestra primera versión de código. Para ello navegamos hasta la carpeta deseada en este caso será la de software de PC.

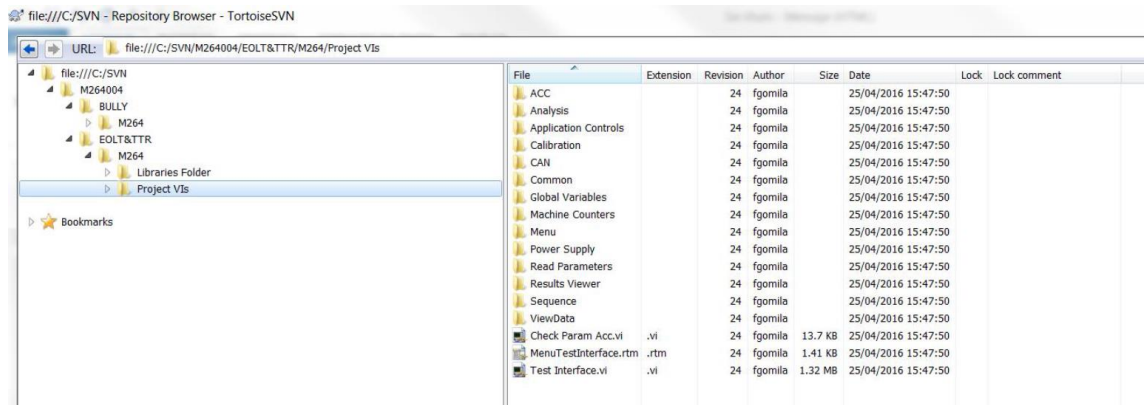


Ilustración 103 Vista de las carpetas en el servidor.

Para subir la primera versión de nuestro código debemos seguir los siguientes pasos:

- I. Navegamos a la carpeta local donde tenemos nuestro software, una vez que lo tenemos a punto para poder hacer la primera versión de código.

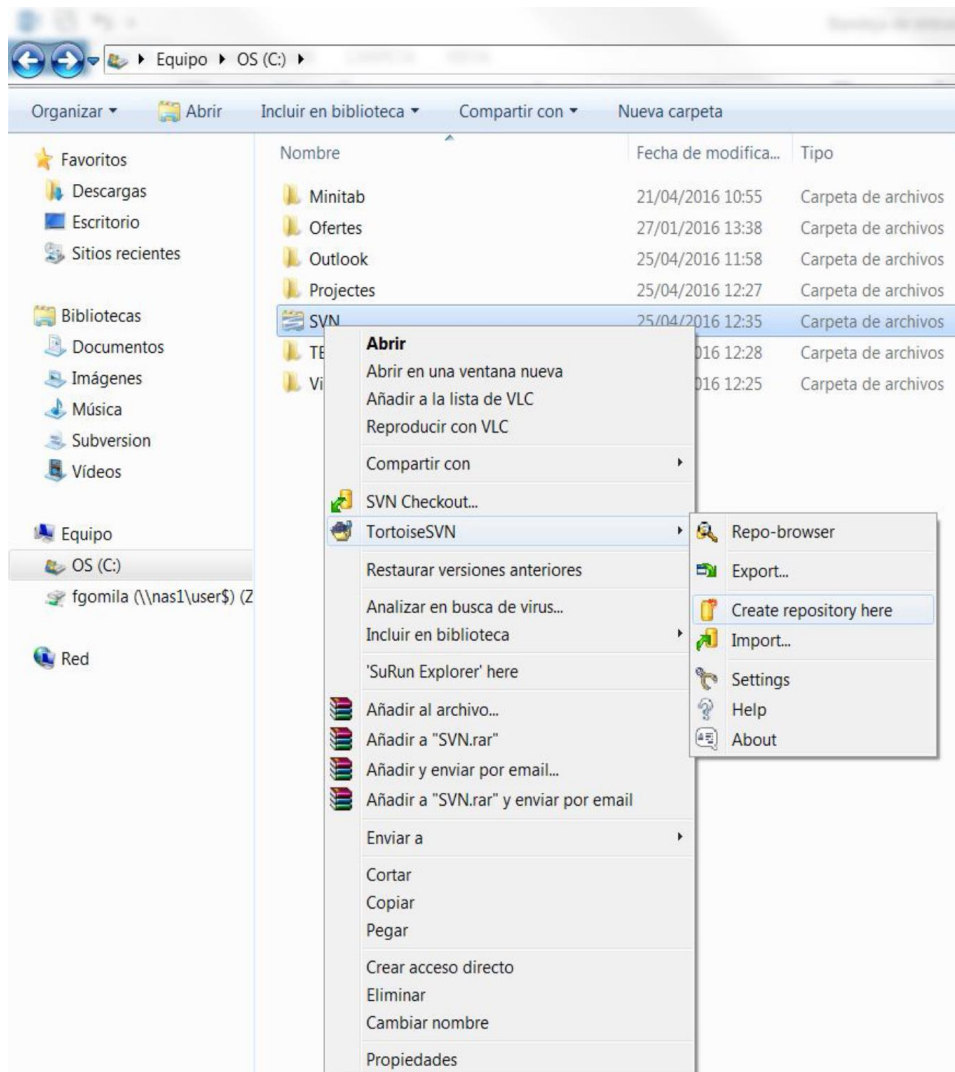


Ilustración 104 carpeta local del proyecto, carpeta de trabajo.

II. Hacemos clic derecho sobre la carpeta. Y seleccionamos la opción “import” del submenú de tortoissvn

III. Seleccionamos la carpeta donde queremos que se versione nuestro código

Una vez seleccionada nos vuelve a aparecer la una ventana para poder insertar un descriptivo de la acción que realizamos

- Haciendo un checkout del código.
Una vez tenemos el código subido debemos hacer un checkout del directorio para linkar las carpetas de trabajo con la del servidor, para ello navegamos hasta la carpeta del proyecto en local y con un clic derecho seleccionamos la opción de “checkout” del submenú.

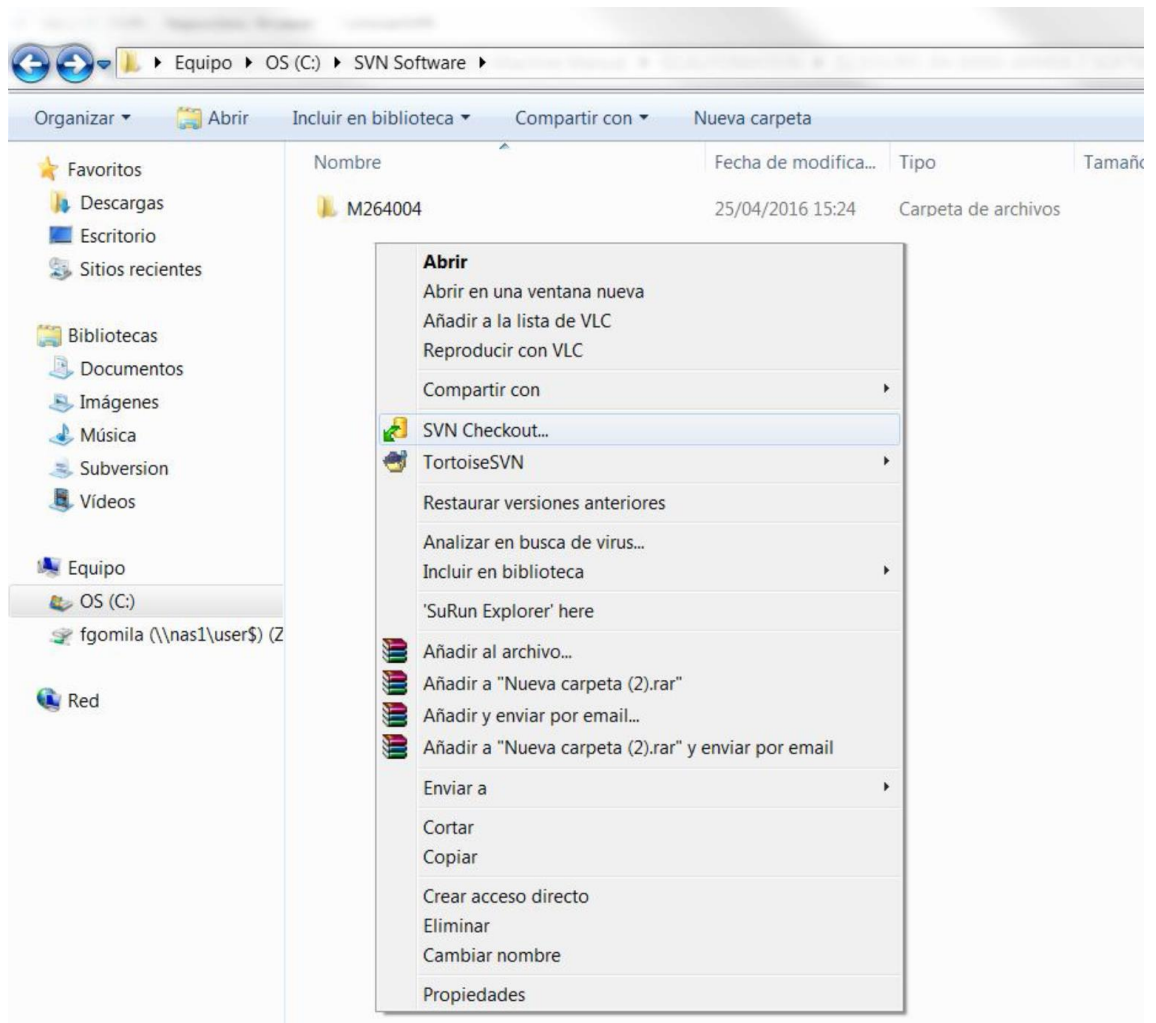


Ilustración 105 Submenú para selección del checkout.

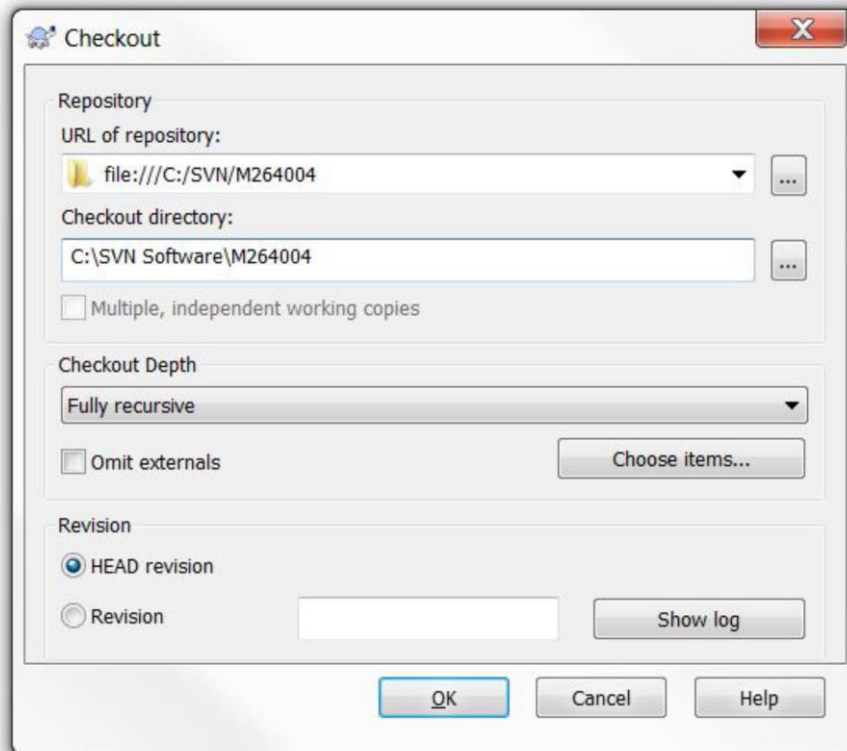


Ilustración 106 opciones del checkout.

Una vez lo tenemos linkado aparecerá un icono de directorio indicando que esta linkado y que los códigos son iguales.

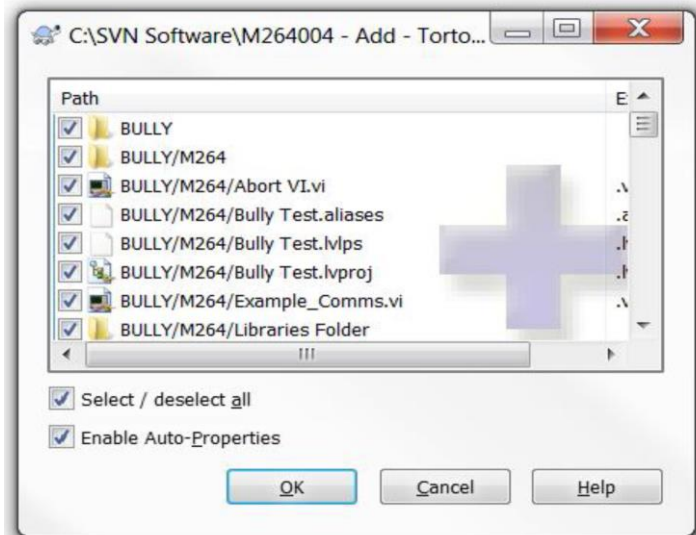


Ilustración 107 Carpeta linkada.

2. Modificaciones en el código.
Modificamos el código en la carpeta de trabajo local y aparecerá un icono de color rojo en la carpeta del proyecto. Esto indica que hemos modificado el código,

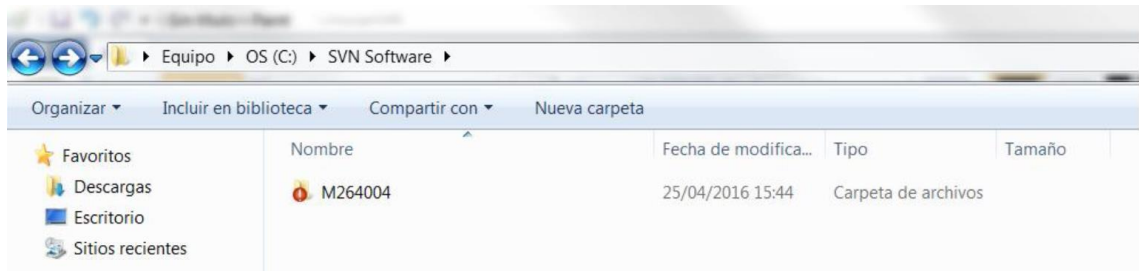


Ilustración 108 Código modificado.

Si tenemos ficheros de texto no solo nos indica el cambio en el código sino también en todos los ficheros modificados, indicándolo con un icono de color rojo.

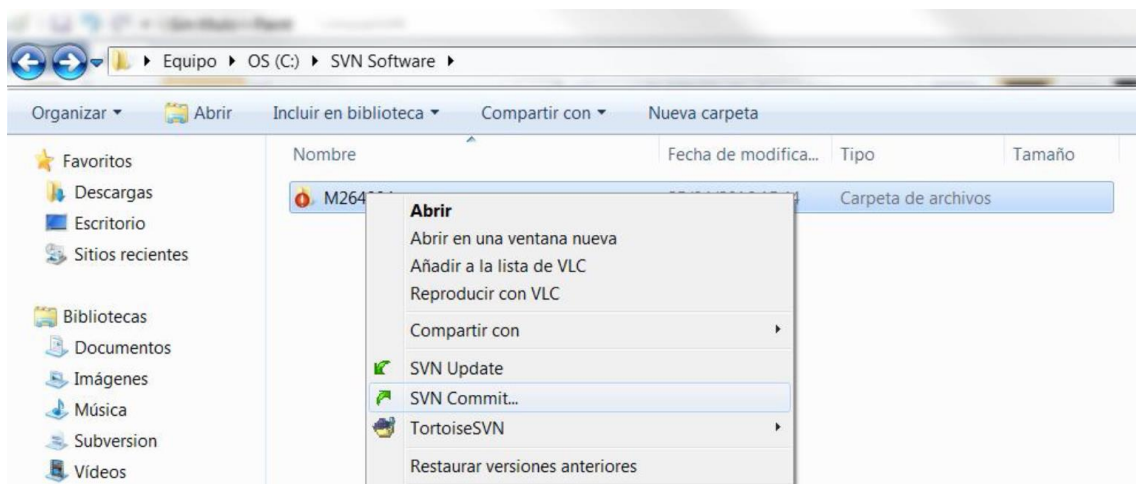


Ilustración 109 Carpetas que contienen código modificado

Llegados a este punto en el que sabemos que hay código/ficheros modificados podemos hacer una verificación de lo que es diferente en la versión reléase en el servidor y la del directorio de trabajo. Para ello navegamos hasta una carpeta o fichero que tenga diferencias y con un clic con el botón derecho navegamos hasta la opción “Check for Modification “de “TortoiseSVN”.

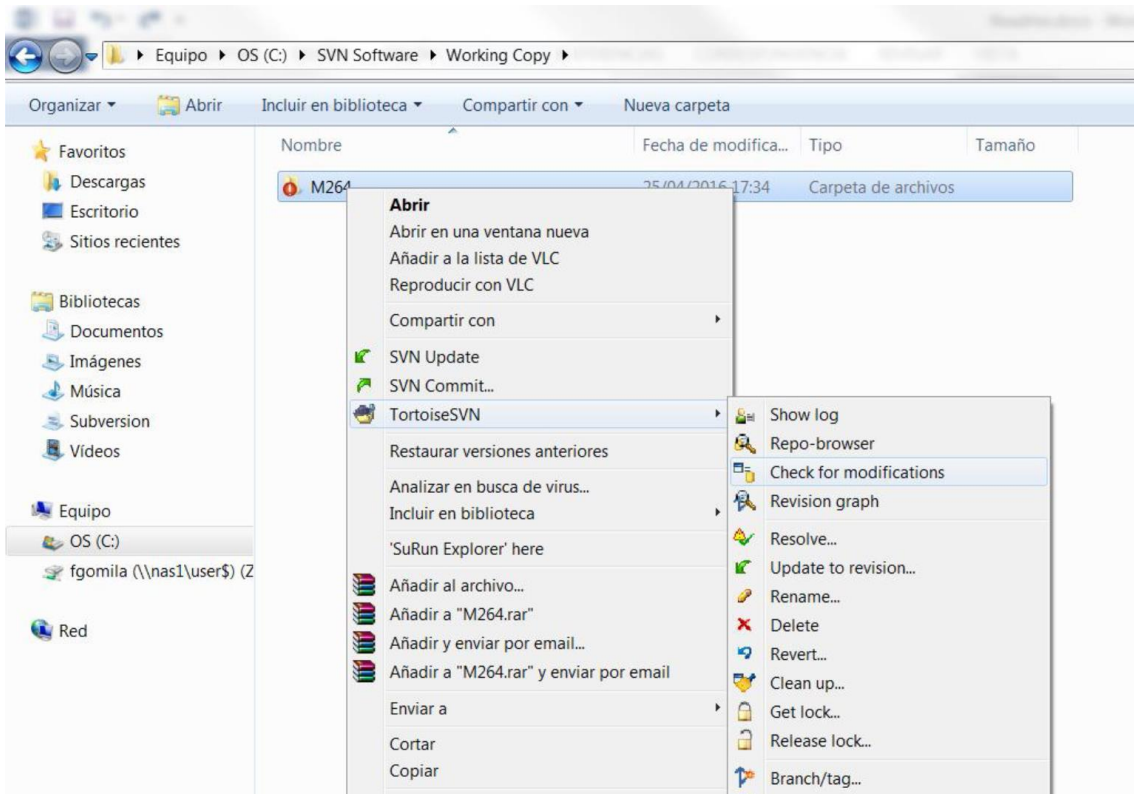


Ilustración 110 Check for modificación

Una vez verificadas las diferencias se abre una ventana que nos indica que ficheros fueron modificados.

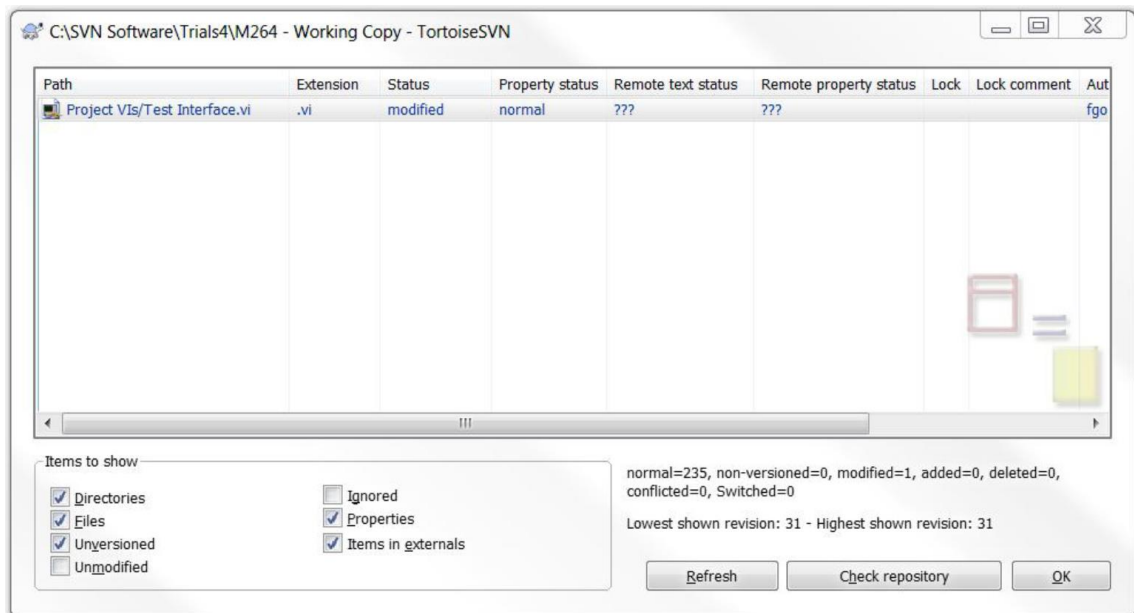


Ilustración 111 Ficheros modificados.

Seleccionando uno de los ficheros modificados obtenemos otra ventana que nos indica los cambios realizados en este caso se utiliza la herramienta de tortoiseSVN la cual nos permite hacer verificaciones de los ficheros de texto plano que no son de código, pero también podemos utilizar la herramienta de nuestro propio IDE para hacer las comparaciones como este sistema no dispone de programación en texto plano sino que es en diagrama de bloques y flujo de datos debemos utilizar directamente el IDE. Para poder utilizar la aplicación de comparación de nuestro IDE debemos hacer una pequeña configuración. Para ello abrimos las opciones de configuración de tortoiseSVN, vamos al menú inicio "TortoiseSVN" → "Settings".

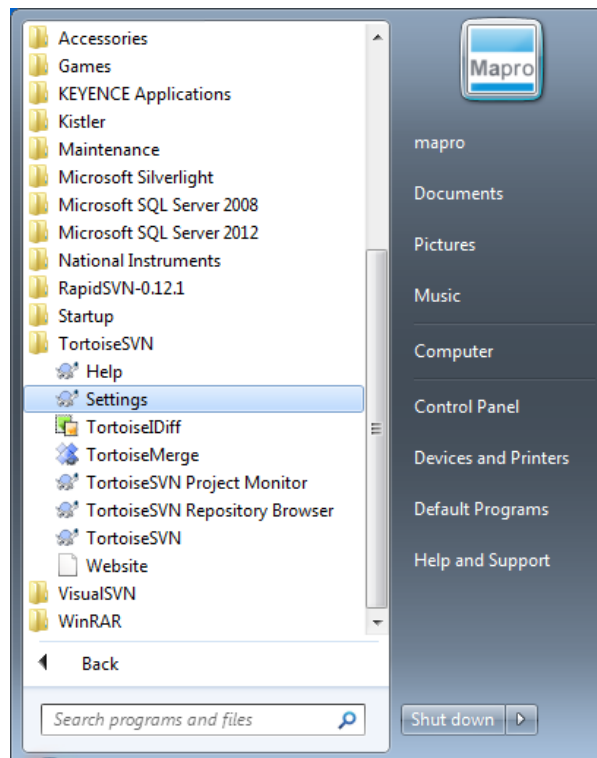


Ilustración 112 Navegando a Settings

Dentro de settings damos la ruta del software que queremos que realice las comparaciones:

- Vamos a Diff Viewer
- En la primera opción seleccionamos el software que queremos que realice las comparaciones de código

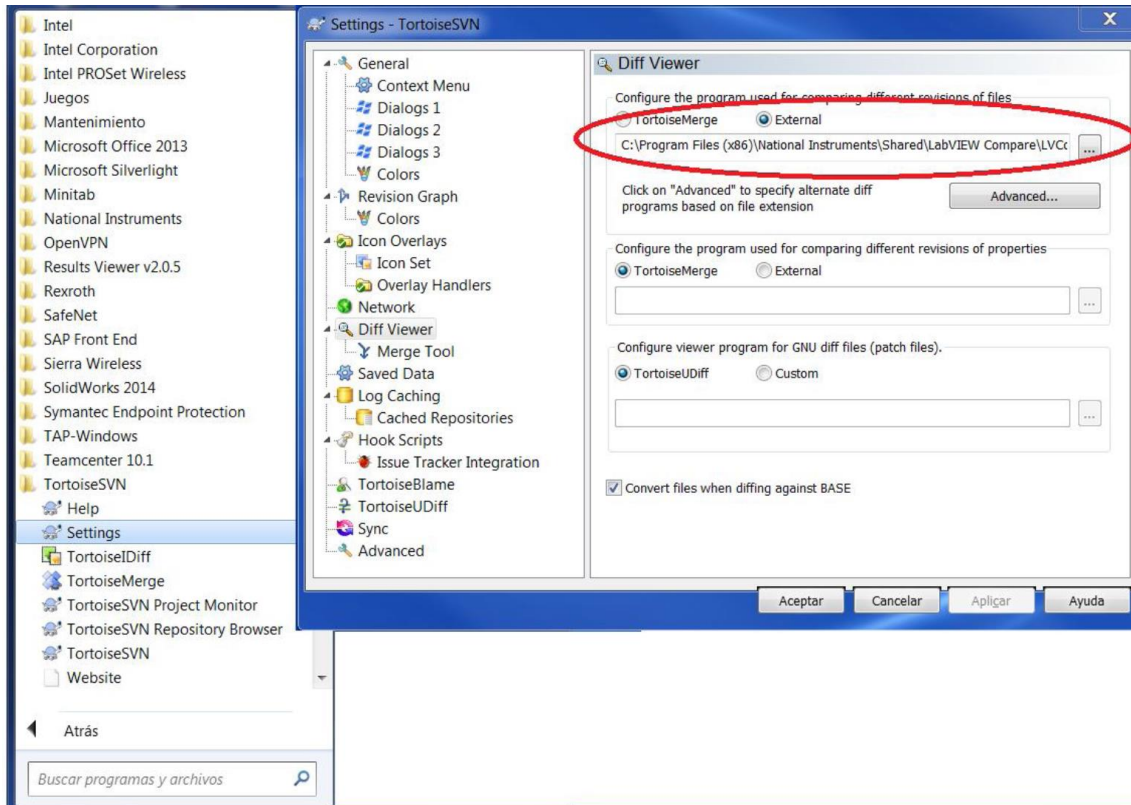


Ilustración 113 Navegando a Settings

En la carpeta deseada vamos al submenú TortoiseSVN y seleccionamos “Check For Modifications”, esta vez se abrirá el software específico que hemos configurado previamente con los dos ficheros abiertos. Uno será el versionado y el otro el modificado en local.

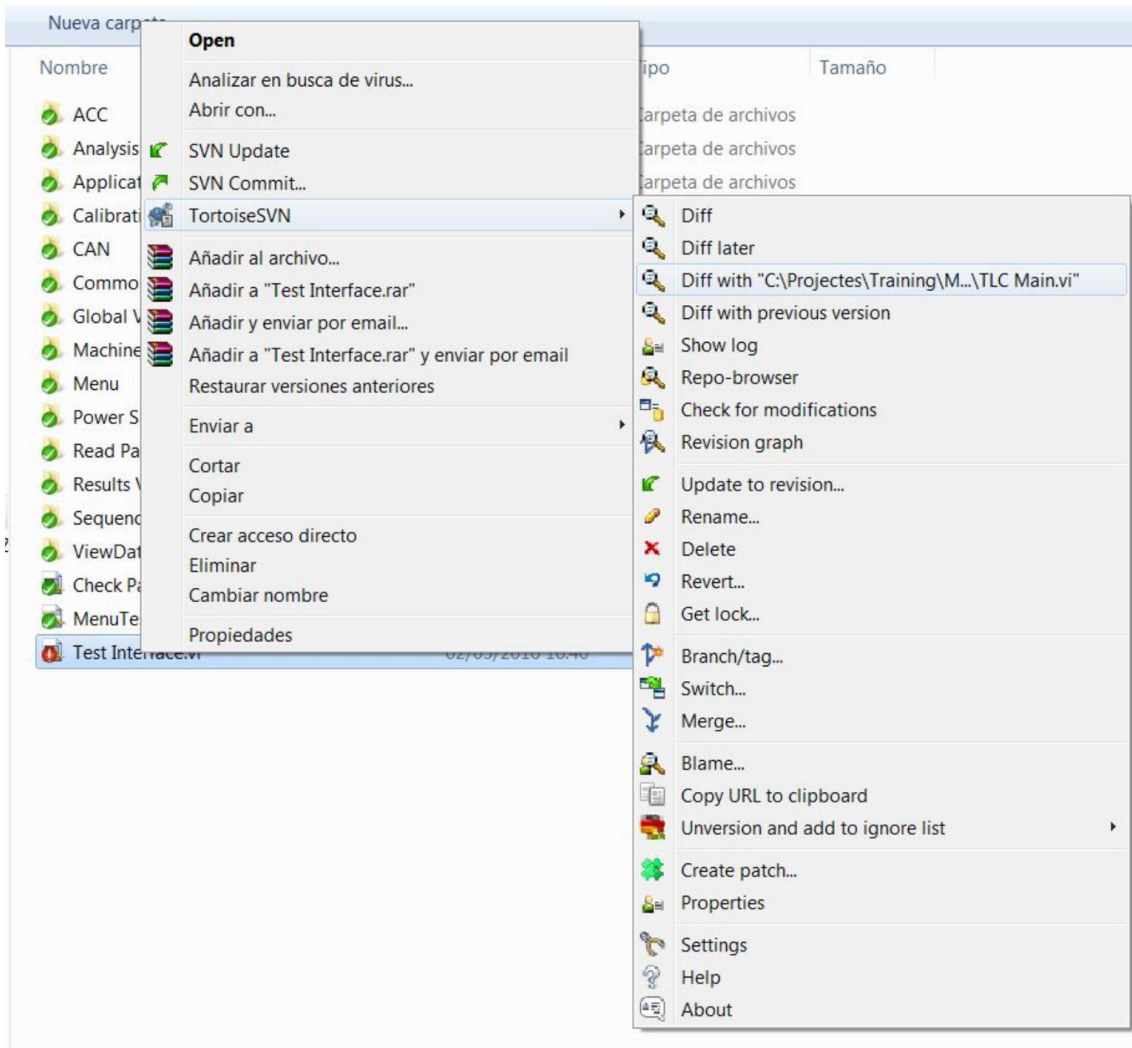


Ilustración 114 Seleccionamos ver las diferencias

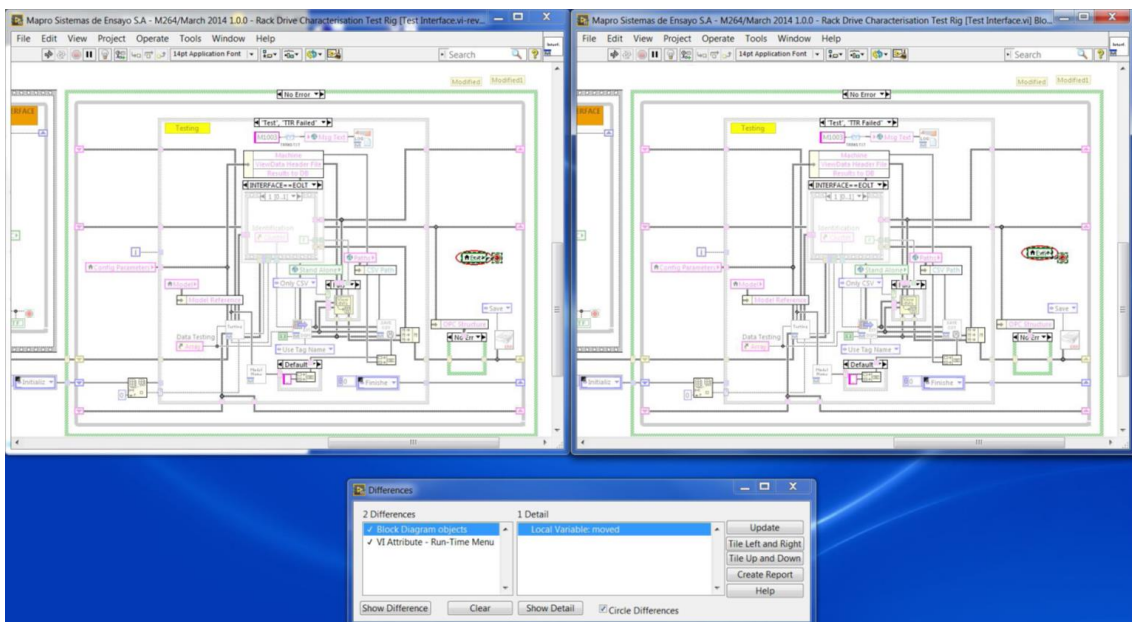


Ilustración 115 Ficheros abiertos en nuestra IDE de programación.

1. Re-versionado.

Una vez hechas las modificaciones deseadas en el código y estamos preparados para hacer un nuevo reléase para poder cargar una versión de nuestro código, para ello solo debemos realizar un sencillo paso que es navegar al submenú “SVN Commit” de tortoise.

9. Usuarios y control de acceso

10.1 Breve descripción

Uno de los requisitos es el de hacer una buena gestión de usuarios para poder garantizar el control de accesos y la integridad de cada una de las copias de seguridad. Es este caso lo que se pretende es poder dar acceso a determinados usuarios sean personal interno de la empresa o subcontratados durante determinadas fases de los proyectos, luego deberán reestructurarse para poder dar o quitar el acceso a determinados usuarios, en este caso se considera que al iniciar el proyecto se designara a una persona encargada de asignar los accesos y el nivel que cada uno deberá tener.

Independientemente de los usuarios que debemos dar de alta o baja en nuestro sistema teniendo en cuenta que se tendrá acceso web a nuestro servidor lo que so que hay que buscar es una herramienta adecuada a las necesidades que debemos cubrir.

Para poder asegurar las comunicaciones lo que se va a realizar es un encriptado en SSH además de hacer que los usuarios que estén dados de altas en el SCV dependan del propio dominio de mapro en el cual se cuentan con toda la información de cada empleado además de usuario y contraseña, para así poder gestionar de una forma más automatizada quien tiene permisos para hacer que y cuando.

10.2 Necesidades a cubrir para la gestión de usuarios

A continuación, se listan las necesidades a cubrir:

- Altas a trabajadores de la empresa.
 - o En la central de la empresa.
 - Trabajando de forma localizada
 - Trabajando de forma deslocalizada (desplazados al exterior VPN ya existente o servidor SSH + HTTPS)
 - o En las delegaciones (VPN ya existente o servidor SSH + HTTPS).
- Alta a trabajadores externos.
 - o Necesidad de acceso temporal con fechas de caducidad para evitar accesos no requeridos.
 - o Necesidad de hacer una conexión segura (VPN para personal externo o SSH + HTTPS).

Uno de los requisitos a seguir en cuanto a los usuarios internos, trabajadores contratados por la empresa es que al estar dado de alta en el sistema las mismas credenciales sean útiles para poder trabajar con el SCV por ello se necesitará hacer una migración para darlos de alta, además se deberá buscar una interface útil de configuración de usuarios para poder dar de alta a los que son trabajadores externos a la empresa de una manera fácil y rápida, con las opciones necesarias para poder configurar, por ejemplo fecha en la que dejaría de tener acceso.

10. Conclusiones.

Una vez realizada la instalación y el setup del SCV teniendo en cuenta que es un proyecto interno de una empresa real y que por determinadas causas la disponibilidad de recursos que lo hizo retrasar, aun así para mí ha sido un reto técnico ya que no pertenezco dentro de la empresa al departamento de IT esto me permitió más allá de la implementación del SCV realizar un proyecto de IT desde la parte de gestión del propio proyecto, definiendo especificaciones técnicas, así como haciendo y definiendo las baterías de pruebas realizadas, control de costes, personal, recursos, etc. Por tanto, sin haber realizado algunas de las asignaturas del master, más allá de las puramente técnicas, como son la de gestión de proyectos, no hubiese podido emprender este camino, lo que queda en el tintero es que el sistema actualmente no es completamente funcional, el trabajo que me queda ahora es el de redefinir las prioridades del proyecto, reestructurar el timing inicial y seguir el resto de la implantación como gestor del mismo.

En estos momentos el sistema es funcional para ciertos usuarios dados de alta directamente en el sistema (debían) y no es accesible desde el exterior.

A partir de aquí los pasos que quedan, explicados a grandes rasgos son.

- La integración de las autenticaciones de usuarios mediante el propio dominio de mapro
- La posibilidad de hacer que cada proyecto de la mano de su gestor tenga en determinadas fases del mismo la posibilidad de dar de alta nuevos usuarios y dar acceso o no a los existentes
- La integración a la salida al exterior.
- Por último queda migrar todos los proyectos de software al nuevo sistema.