Universidad de las Ciencias Informáticas
Facultad 3

Universidad de las Ciencias Informáticas

Instalador de Quarxo

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor:
Yordan Remón Reyes

Tutor:
Ing. Hector Peña Torres

La Habana, junio 2013
"Año 54 de la Revolución

DECLARACIÓN DE AUTORÍA

Declaro ser el autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste se firma la presente acta a los ____ días del mes de _____ del año 2013.

Yordan Remón Reyes **Ing. Hector Peña Torres**

_____ _____

Firma del Autor Firma del Tutor

AGRADECIMIENTOS

A mis padres y mis hermanos que son la razón de que hoy este aquí.

A mis tíos, Carlito y Yarina que en estos años se han convertido en mis segundos padres.

A todos mis amigos por estar a mi lado en los momentos buenos y malos, que son mucho pero todos sabes quienes son, gracias por su apoyo y comprensión.

A todos los que han contribuido en mi formación profesional y en la culminación de esta investigación.

A todos, gracias.

DEDICATORIA

A mis padres y hermanos que han apoyado mis estudios toda la vida. En especial a mi *padre, por ser mi guía brillante.*

A mi familia y amigos que son muchos.

RESUMEN

La Universidad de las Ciencias Informáticas (UCI) ha venido desarrollando el Sistema Quarxo para el Banco Nacional de Cuba. En la actualidad la instalación de la aplicación se realiza de forma manual y poco intuitiva. Por ello surge la necesidad de una herramienta de instalación que automatice las principales tareas de este proceso. El presente trabajo tiene como objetivo el desarrollo del instalador de software para Quarxo.

El instalador, debe permitir llevar a cabo el proceso de instalación de forma rápida, sencilla y eficiente. Este proceso se realizará de forma transparente para el usuario y de tal manera que este no necesite tener amplios conocimientos sobre informática para instalar el sistema. En el presente documento, se describen detalladamente los principales instaladores existentes, realizando comparaciones en busca de las mejores propuestas para desarrollar la herramienta con la mayor calidad posible. El modelo general de despliegue estudiado se ha particularizado en un conjunto de procesos para el caso específico de Quarxo. Al final del documento se especifican las pruebas realizadas para validar los requisitos con los cuales la aplicación debe de cumplir. Para dar solución a la problemática existente se ha construido una herramienta de instalación utilizando *BitRock* como base para el desarrollo y *Apache Ant* como tecnología de automatización de código.

Palabras Claves
Despliegue, liberación, instalación, Quarxo.

TABLA DE CONTENIDO

Índice de Tablas

Índice de Figuras

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) son un eslabón importante para las instituciones, atendiendo a que facilitan todo tipo de actividades que se realizan en ellas. Las ventajas del manejo de las mismas a través de sistemas informáticos, son significativamente superiores a las de dicho procedimiento realizado de forma manual. Estos sistemas informatizan los procesos y suministran los datos necesarios para la toma de decisiones, contribuyendo al incremento de su capacidad de organización y eficiencia.

Desde hace algunos años las instituciones bancarias viene haciendo uso de las TIC para mejorar sus procesos y servicios; evidenciándose, en el incremento de las operaciones entre las instituciones financieras donde exigen que cuenten con sistemas contables en función de la informatización de sus procesos, permitiendo incrementar el control de sus operaciones y elevar la calidad del servicio ofrecido a sus clientes, dejando clara la necesidad de perfeccionamiento y evolución de sus soluciones informáticas.

El Sistema Bancario Cubano y en específico el Banco Nacional de Cuba (BNC) en función de lograr el manejo adecuado de la información y la interconexión entre los diferentes bancos y sucursales ha estado empleando el Sistema Automatizado para la Banca Internacional de Comercio (SABIC) en su versión para MS-DOS [1](Microsoft Disk Operating System, Sistema operativo de disco de Microsoft). Hasta la fecha este sistema ha contribuido a agilizar los procesos y gestionar toda la información dentro de la institución. Sin embargo, a pesar de los innegables beneficios de su uso, este sistema se ha quedado por detrás ante los cambios tecnológicos.

Como parte de la modernización del BNC se le solicitó a la UCI el desarrollo de un software que fuera extensible a todas las tareas que se desarrollan dentro del departamento de gerencias del propio banco, permitiendo también la interacción de esos procesos con otros procesos externos que se desarrollan en los demás bancos y entidades bancarias del país. Después de algunos años de desarrollo, la UCI comenzó a desplegar el sistema Quarxo, nombre que recibe el Sistema Automatizado para Gestión Bancaria diseñado para gestionar los procesos en la institución.

[1] MS-DOS, es un sistema operativo para computadoras basados en x86. Fue el miembro más popular de la familia de sistemas operativos DOS de Microsoft.

Quarxo una aplicación web desarrollada bajo los estándares de Java Enterprise Edition (JEE[2]). Hasta la fecha actual la instalación del sistema y su despliegue se concibe de forma desacoplada, existe una separación entre los componentes utilizados en dicha actividad tales como:

- ✓ Máquina Cliente
    - Instalar la JDK 1.6.0_25 de 64 bits o superior
- ✓ El servidor de aplicación Web Apache TomCat.
    - Instalar el Apache Tomcat 7.0.16 o superior
    - Se configuran los parámetros de memoria a utilizar por la aplicación en la sección Java Options de la pestaña Java.
    - Se configura para que el servicio del TomCat se inicie automáticamente al iniciar el sistema operativo.
- ✓ Servidor de Base Datos.
    - Instalar Sql Server 2005.

Para la instalación de todos estos componentes se verifican manualmente las precondiciones fundamentales para desplegar el software tales como: sistema operativo, existencia de la plataforma de desarrollo, un servidor web, un servidor de base de datos, la configuración del navegador entre otras variables de entorno que son necesarias para la puesta en marcha del despliegue. El personal encargado de llevar a cabo este procedimiento necesita tener vastos conocimientos sobre tecnologías JEE, servidores, tanto de aplicaciones como de base de datos, así como características puntuales de la arquitectura del sistema para poder hacer las configuraciones pertinentes.

Pasos que se llevan a cabo manualmente para la instalación y despliegue de Quarxo en la actualidad:

Una vez instalado el Apache Tomcat, se ejecuta el *Monitor Tomcat* instalado y se le configuran los siguientes parámetros de memoria a utilizar por la aplicación en la sección Java Options de la pestaña Java:

[2]JEE, del inglés java Enterprise Edition: Tecnología para desarrollar aplicaciones empresariales basadas en la Web.

-Xms8000m

-Xmx8000m

-XX:PermSize=1500m

-XX:MaxPermSize=1500m

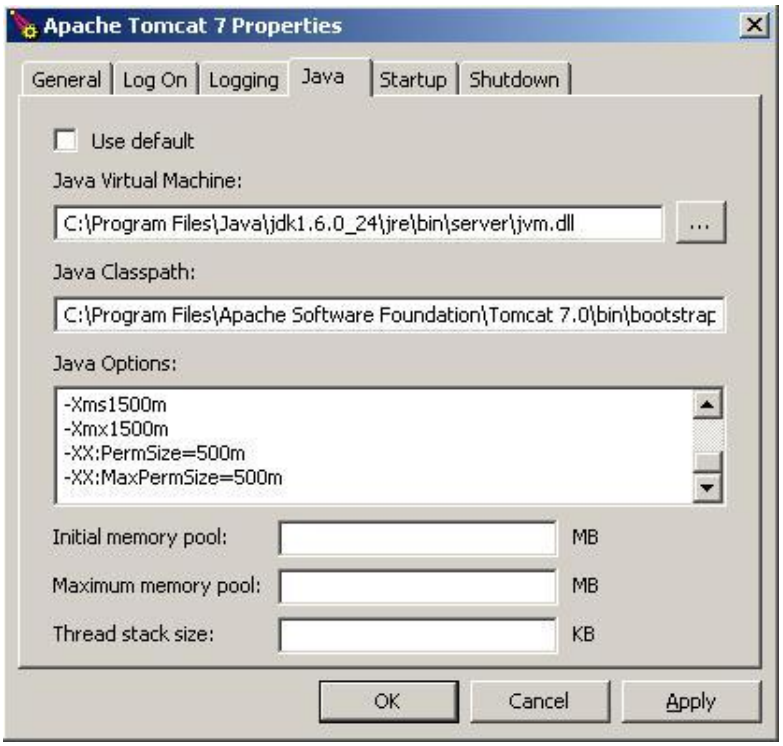
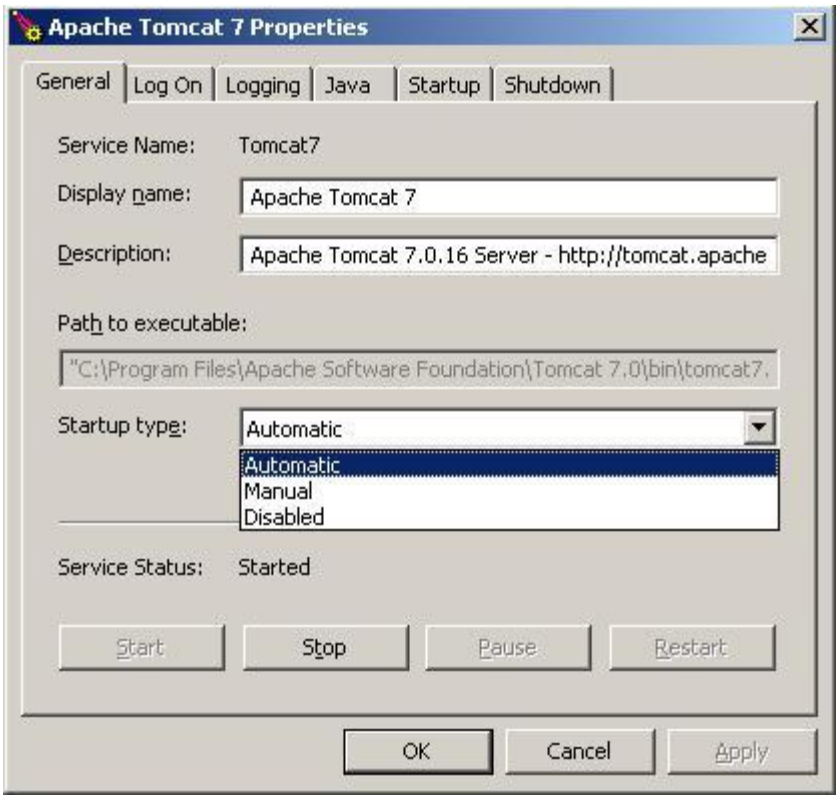
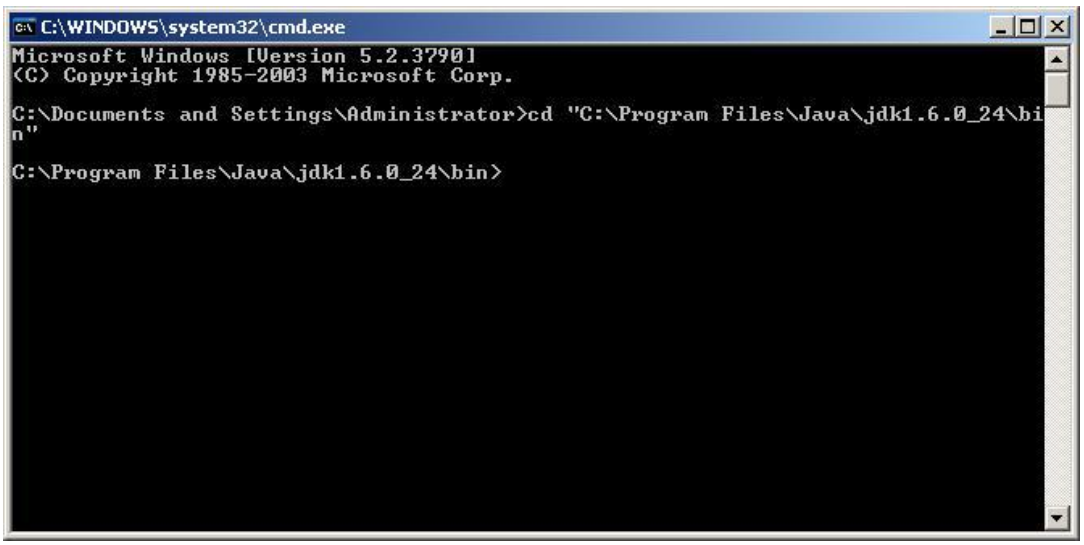


Ilustración 1. Propiedades del TomCat

Luego se configura para que el servicio del TomCat se inicie automáticamente al iniciar el sistema operativo.



Luego se configura el Apache Tomcat para que se pueda utilizar el canal seguro https. Para ello primero se levanta la consola de comandos de Windows y se va a la dirección donde está la JDK.

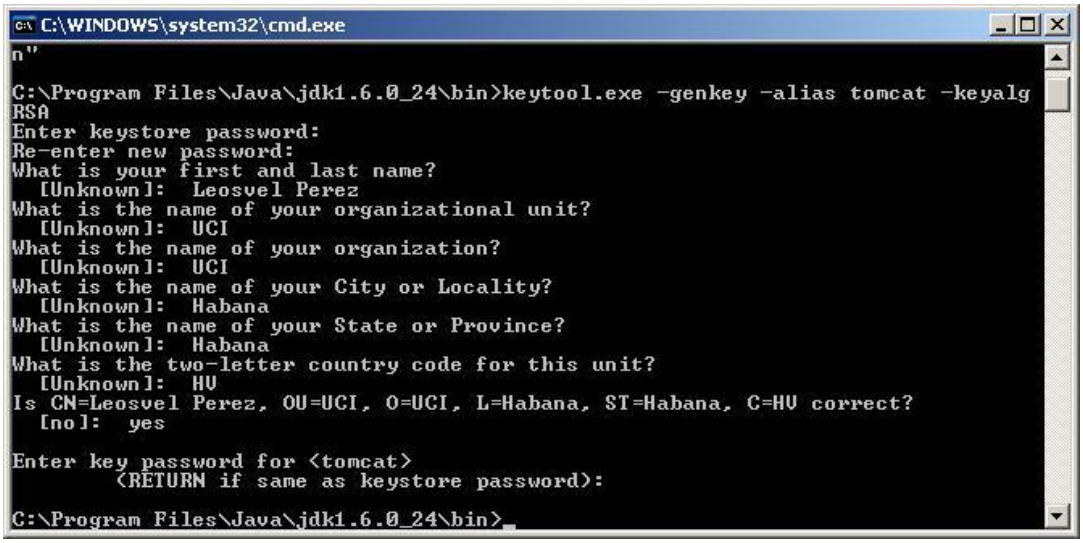


Ubicados en esa dirección se escribe lo siguiente: "*keytool.exe –genkey –alias tomcat –keyalg RSA*" y se da Enter.

Luego sale un diálogo preguntado la contraseña del almacén de llaves, la contraseña es "changeit" y luego de escribirla se presiona Enter.

Luego pide confirmar la contraseña y después una serie de datos que se llenan según el usuario. Luego de introducidos los datos, sale un diálogo preguntando si están correctos dichos datos, se escribe "yes" y se presiona Enter.

Luego sale otro diálogo que pide escribir la contraseña clave para el TomCat, se teclea o si es la misma que la del almacén de claves pues se presiona Enter sin escribir nada.



Hecho esto queda creado un fichero en la dirección del usuario de la pc para el cual se configuró llamado .keystore.

Luego se va a la carpeta donde está instalado el Apache Tomcat y dentro de la carpeta conf se abre el fichero server.xml y se le agrega la siguiente configuración dentro del tag <Service name="Catalina">:

```
    <Connector port="8443" maxThreads="200"
    scheme="https" secure="true" SSLEnabled="true"
    keystoreFile="C:\Documents and
Settings\Administrator\.keystore"
 keystorePass="changeit" clientAuth="false" sslProtocol="TLS" />
```

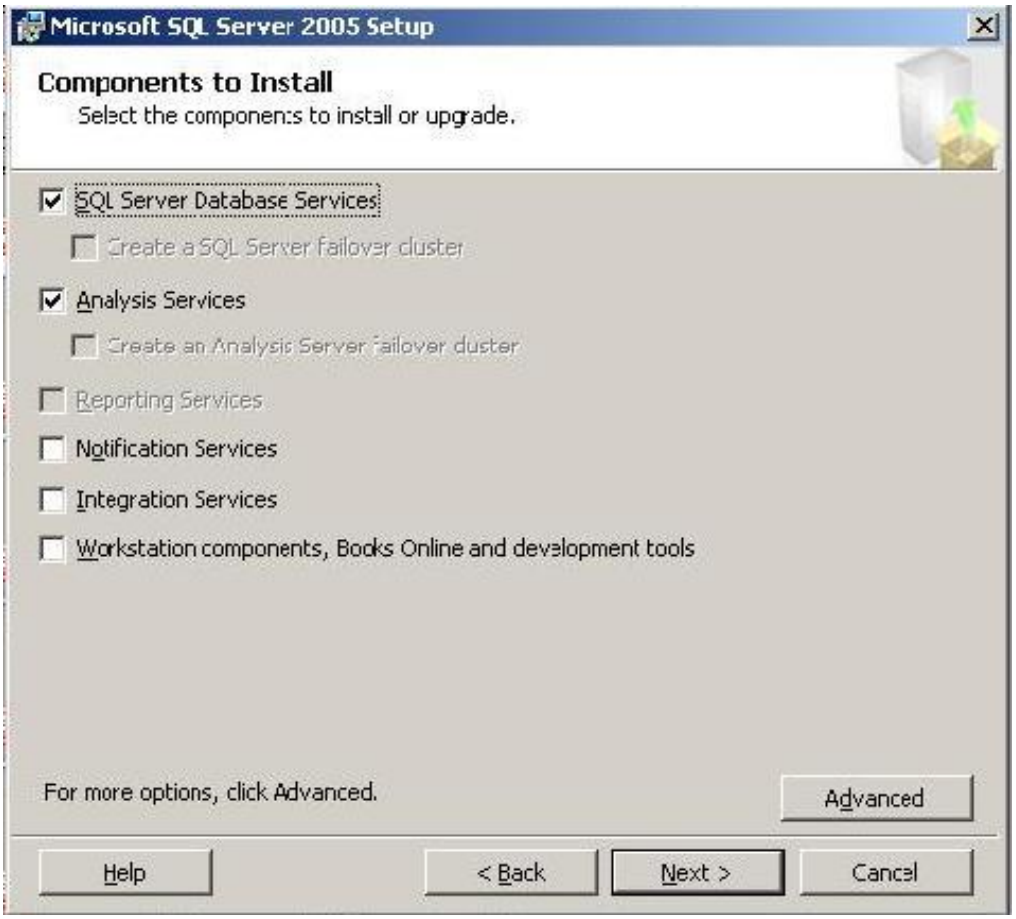
Donde se sustituye el valor de la propiedad *keystoreFile* por la dirección del fichero .keystore creado anteriormente.

Instalación y configuración del SQL Server 2005

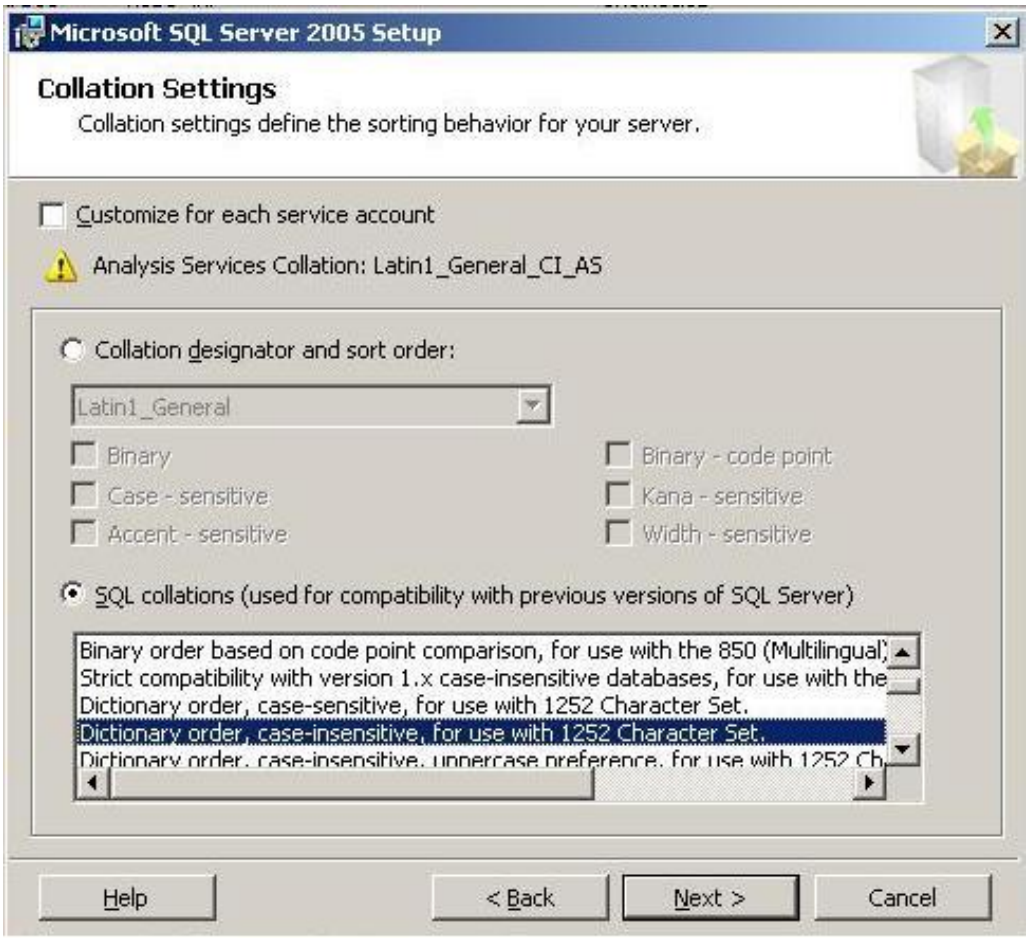
Pasos para la instalación de SQL Server 2005.

1. Ejecutar el archivo de instalación \sql server 2005\SQL Server x86\Servers62192\ejecutar splash.hta.

2. Seguir con la instalación sin hacer cambios hasta llegar a la ventana de los componentes a instalar y ajustar cambios como muestra la siguiente figura.



3. Continuar con la instalación según se muestra en las siguientes figuras.



Activar las conexiones TCP/IP al servidor.

1. Ejecutar SQL Server Configuration Manager/SQL Server 2005 Network Configuration/Protocols from MSSQLSERVER.
2. Activar Named Pipes y TCP/IP (Clic derecho, Enable).
3. Reiniciar el Servicio del SQL Server.

Activar el modo de autenticación SQL Server And Windows Authentication Mode.

1. Propiedades del servidor/Security.
2. En Server Authentication, seleccionar SQL Server And Windows Authentication Mode.
3. Reiniciar el Servicio de SQL Server.

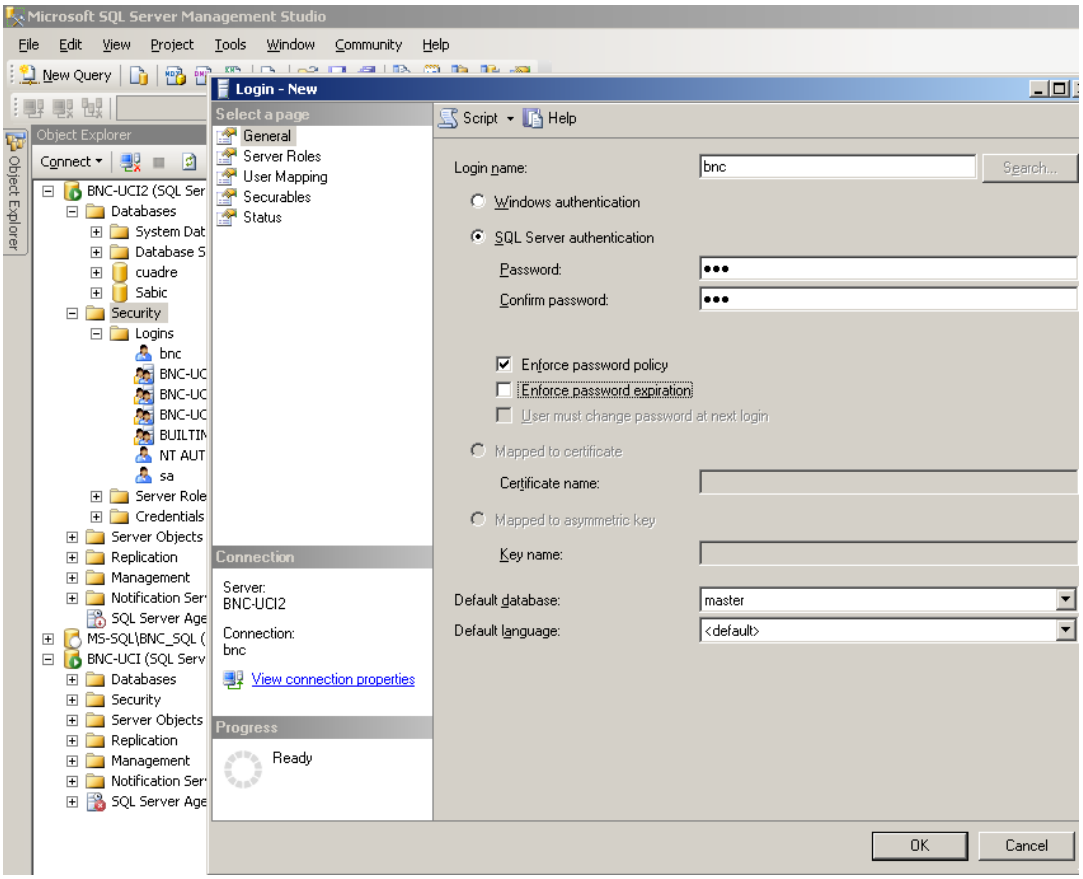
Ejecutar la siguiente consulta para poder activar las DLL de seguridad.

```
sp_configure 'clr enabled', 1
GO
RECONFIGURE
GO
sp_configure 'clrenabled'
GO
```

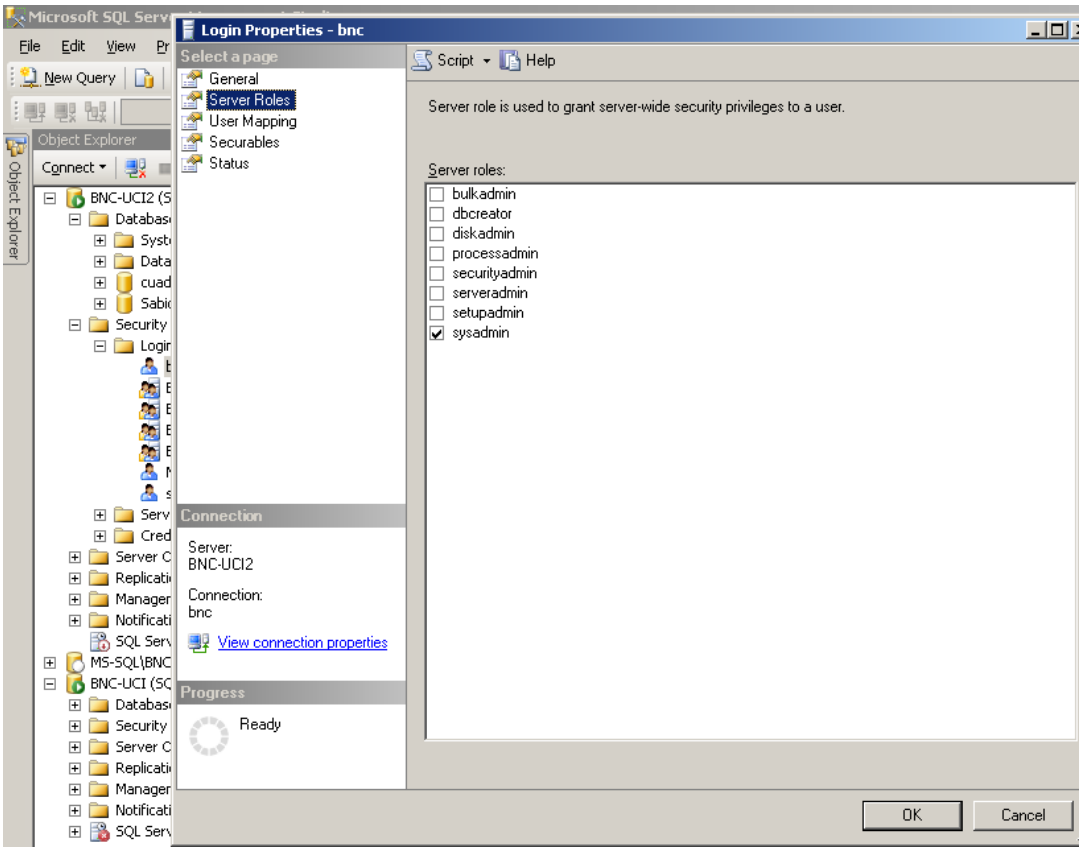
Configuraciones para el correcto funcionamiento de la BD:

- ✓ Como premisa se debe tener instalado el Sql Server 2005.

Antes de montar la BD Sabic: Se debe crear un nuevo login con nombre *bnc*.



En la opción “Server Roles” se escoge la opción “sysadmin”.



Luego se carga la BD Sabic.

- ✓ Para que funcione correctamente el subsistema de usuarios en el Quarxo, se deben seguir los siguientes pasos:
  - Copiar las dll: *SabicDES.dll* y *SwSecure.dll* (dependiendo del sistema operativo que tenga instalado, en este caso las dll para PC de 64 bit) para la dirección C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\.
  - Ejecutar los scripts siguientes:
    - exec sp_changedbowner 'bnc'
    - alter database sabic set trustworthy on
    - select dbo.sfn_calc_crc ('bnc') //esta consulta no debe devolver ningún error.

La instalación y despliegue de Quarxo se convierte en una tarea complicada y riesgosa; es necesario realizar todo el conjunto de configuraciones descritas anteriormente, que al hacerlas manualmente, se puede incurrir en errores con el costo de tener que iniciar de nuevo el proceso de instalación y configuración. Estos problemas e inversión de tiempo se

podrían erradicar y minimizar considerablemente si se automatizara la instalación y configuración del sistema.

Tomando en cuenta la situación anterior se plantea el siguiente **problema a resolver**: ¿Cómo contribuir a agilizar las tareas referentes a la instalación del sistema Quarxo para disminuir el tiempo de instalación y la complejidad de las actividades en la instalación del sistema?

Definiéndose como **objeto de estudio:** Proceso de instalación de software.

Según lo planteado anteriormente se establece como **campo de acción:** Herramienta de instalación de software.

Para dar solución al problema anteriormente planteado se trabajó sobre la siguiente **Idea a defende**r: “Si se desarrolla una herramienta de instalación para el sistema Quarxo, se agilizarán las actividades referentes a los procesos de instalación del software permitiendo disminuir el tiempo de instalación y la complejidad de las actividades en la instalación del sistema.”

El **objetivo general** es**:** Desarrollar un instalador que permita disminuir el tiempo de instalación y la complejidad de las actividades en la instalación del sistema Quarxo.

De acuerdo con lo anterior se declaran los **objetivos específicos** siguientes:

1. Fundamentar la investigación, mediante la elaboración del Marco Teórico.
2. Analizar, diseñar e implementar un instalador para las distintas tecnologías del sistema Quarxo.
3. Validar el instalador de Quarxo.

Para darle cumplimiento a los objetivos se trazan las siguientes **tareas**:

1. Caracterización de los procesos relacionados con la creación de instaladores.
2. Valoración de las herramientas existentes para la creación de instaladores.
3. Caracterización y análisis de las herramientas necesarias para el desarrollo del análisis, diseño e implementación del instalador.
4. Identificación, especificación y validación de los requisitos funcionales del software.

5. Realización de los diagramas correspondientes al análisis y el diseño del desarrollo del instalador.
6. Implementación del instalador de Quarxo.
7. Validación del instalador de Quarxo.

Resultados esperados

Un instalador conforme a las especificaciones del sistema Quarxo que permita disminuir el tiempo de instalación y la complejidad de las actividades en la instalación del sistema Quarxo.

Los métodos científicos que se van a emplear en la investigación del presente trabajo se describen a continuación:

Métodos Teóricos

- ✓ **Histórico – Lógico:** Este método científico se aplicó en el estudio realizado acerca del estado del arte sobre los instaladores que se utilizan en la actualidad prestando especial atención en los instaladores web. Permitió analizar la evolución de estos sistemas y cuál es su tendencia actual.

- ✓ **Analítico – Sintético:** Con el uso de este método se analizó los resultados arrojados en el estudio realizado acerca de los diferentes instaladores, donde se establecieron una series de parámetros, atendiendo principalmente a las características relacionadas con sus objetivos fundamentales, para realizar una comparación entre ellas y tomar los resultados arrojados por dicha comparación, como datos de gran interés para la actual investigación.

Métodos Empíricos

- ✓ **Observación:** Mediante el método científico de la observación, se detectó la situación actual existente en el proyecto Quarxo, en cuanto a las dificultades existentes a la hora de llevar a cabo el proceso de instalación del sistema, así como la necesidad de creación de una herramienta para cumplir con este requerimiento, fundamental a la hora de llevar a cabo el despliegue de Quarxo.

- ✓ **Entrevista:** Mediante la entrevista que se le realizó a los profesionales que conforman el equipo de despliegue se pudo obtener toda la información del

proceso en cuestión.

Este trabajo está conformado por tres capítulos que recogen todo lo abordado en la investigación:

- ✓ Capítulo 1: Fundamentación Teórica del instalador de Quarxo, se brinda un estado del arte sobre los elementos que forman parte del problema que se presenta, así como aquellos que son importantes a tener en cuenta para dar la solución que se requiere.
- ✓ Capítulo 2: Análisis y Diseño de la solución propuesta: Características del Sistema, teniendo en cuenta los requisitos funcionales y no funcionales y las Historias de Usuarios relacionadas con los mismos. Además, se realiza una descripción de las tareas de ingeniería a través de los diagramas necesarios según plantea la metodología empleada.
- ✓ Capítulo 3: Implementación y Prueba de la solución propuesta: se describirá el estándar de programación utilizado para la implementación del sistema y se realizan las pruebas al sistema según la metodología definida, ya sean pruebas unitarias o de aceptación y, en el caso de ser pruebas de aceptación, se elaboran los casos de pruebas correspondientes.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DEL INSTALADOR DE QUARXO

1.1 Introducción

En este capítulo se presentan las definiciones y conceptos fundamentales relacionados a los procesos que enmarcan el problema a resolver de la investigación en curso. Se muestra un estado del arte de cada una de las herramientas que brindan un soporte para la creación de instaladores de software. Además se establece una comparación entre los exponentes fundamentales de estas herramientas. Se describen las principales tecnologías para la generación o automatización de código así como el estado actual de la instalación de las aplicaciones Web.

1.2 El Proceso de Despliegue del Software

Las aplicaciones de software actualmente no son sistemas autónomos. Son cada vez más el resultado de la integración de colecciones de componentes heterogéneos, muchas veces distribuidos sobre una red informática. Estos componentes pueden ser proporcionados por diferentes proveedores y pueden ser parte de diversos sistemas al mismo tiempo. Los componentes cambian y se transforman muy rápidamente, lo que hace difícil gestionar todo el sistema de manera coherente.

En este escenario, un paso crucial del ciclo de vida del software es el proceso de despliegue. Informalmente, el término de despliegue se refiere a todas las actividades que hacen que un sistema esté disponible para sus usuarios. Si bien esto es razonable e intuitivo, la creación de un marco de evaluación para el despliegue de tecnologías, requiere una precisa y amplia comprensión de la naturaleza y las características del despliegue de software como proceso.

El despliegue de un sistema de software involucra la transferencia o copia de sus componentes desde el lado del proveedor hacia el lado del cliente, este constituye su principal objetivo. Una vez desplegado, dicho sistema estará disponible para su uso en el lado del cliente. El proceso de despliegue de software puede ser definido como la entrega, el ensamblaje y la gestión, en un determinado sitio, de los recursos necesarios para utilizar una versión de un sistema.

1.2.1 El Proceso de Despliegue según IEEE

Para el estándar de desarrollo de procesos de software IEEE[3], el despliegue, al cual se refieren con el término de grupo de actividades posteriores al desarrollo [1], tiene definido como sus principales actividades la instalación, la operación y soporte, el mantenimiento y la retirada de un determinado producto de software.

Las actividades de instalación consisten en la transportación e instalación de un sistema de software desde el entorno de desarrollo hasta el cliente. Incluyen la correcta distribución del producto, las modificaciones necesarias, el chequeo y la aceptación en un ambiente operacional. Si surge un problema, deberá ser identificado y reportado[1].

Por otra parte, la operación y el soporte implican la utilización del sistema por parte del usuario y el apoyo permanente a este. El soporte incluye la prestación de asistencia técnica, consultoría con el usuario, y el registro de las solicitudes de los usuarios [1]. El grupo de actividades de mantenimiento se refiere a la identificación de las mejoras y la resolución de los errores de software, fallos y fracasos. Las principales tareas son: la identificación de las necesidades de mejora del producto y la aceptación de cualquier anomalía en un recurso, reportándola como un posible problema[1].

Por último, la retirada de una aplicación informática supone la eliminación de un sistema existente, ya sea por el cese de su funcionamiento o de apoyo, o por su sustitución por un nuevo sistema o una versión actualizada del sistema existente[1]. De una forma más sintetizada se puede decir que el despliegue de *software* son todas las actividades posteriores al desarrollo que se llevan a cabo para producir versiones exitosas de un determinado sistema, teniendo en cuenta también el mantenimiento de este y la migración de tecnología si fuera necesaria.

1.2.2 Actividades que se realizan en el Proceso de Despliegue

El proceso de despliegue es generalmente descrito como un ciclo de vida que involucra varios subprocesos o tareas interrelacionadas posteriores al desarrollo, tales como: liberación, instalación, activación, desinstalación, desactivación y actualización [2] (figura

[3]IEEE, del inglés Institute of Electrical and ElectronicsEngineers: asociación técnico-profesional mundial dedicada a la estandarización en las ciencias de la computación, telecomunicaciones, tecnologías biomédicas y otras ramas de las ciencias.

I). La creciente complejidad de los sistemas de software requiere que a estas actividades del despliegue se les dé más atención.

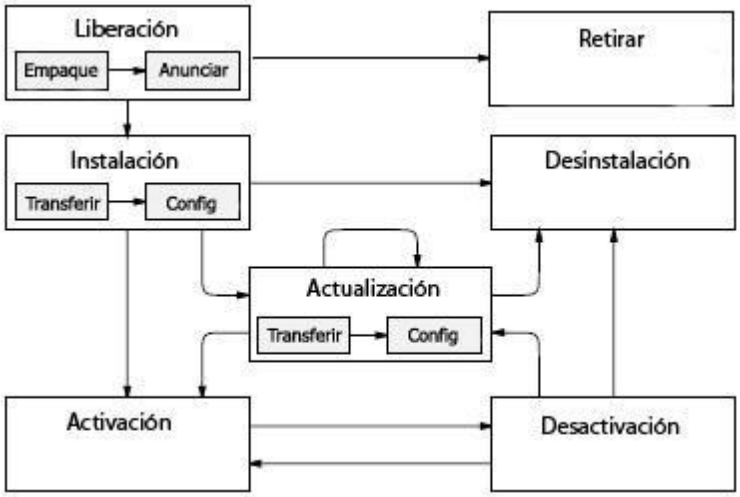


Figura I. Actividades del proceso de despliegue se software

Aunque se puede identificar un conjunto de las principales actividades que constituyen un proceso de despliegue genérico, no se puede precisar las prácticas particulares y los procedimientos específicos de cada una. Se dependerá en gran parte de la naturaleza del *software* a ser liberado, y de las características y requisitos de los proveedores y clientes. Por lo tanto, el proceso descrito en la figura # I se debe interpretar de una manera razonable, debe ser personalizado y enriquecido de acuerdo a requisitos específicos de despliegue de la actividad que se observó. A continuación se presenta una descripción de cada una de las actividades del proceso de despliegue:

Liberación: Constituye la interfaz entre el proceso de desarrollo y el proceso de despliegue. Abarca todas las operaciones necesarias para preparar un producto y que pueda ser correctamente implantado en el sitio del consumo. Así, la liberación debe determinar todos los recursos necesarios por un sistema de *software* para operar correctamente en el sitio de consumo.

También debe recoger toda la información necesaria para la realización de actividades posteriores del proceso de despliegue. Esta información puede ser derivada de una variedad de fuentes, entre ellas el proceso de desarrollo y el conocimiento humano sobre la estructura y funcionamiento del sistema. La liberación incluye el empaquetado del producto para que pueda ser transportado. Este paquete debe contener los componentes y una descripción del sistema que incluye los requisitos y las dependencias de otros

componentes externos, los procedimientos de instalación, y toda la información que sea relevante para la gestión de la aplicación. Otro paso en la liberación es la capacitación, es decir, el conjunto de operaciones que se necesitan para difundir la información pertinente a las partes interesadas en las características y la utilización del sistema que se va a desplegar [3].

Instalación: Comprende la inserción inicial de un sistema en el lado del cliente. Por lo general, es la más compleja de las actividades de despliegue, ya que se ocupa de la correcta concentración de todos los recursos necesarios para utilizar el producto de *software*. Con el término Instalación se hace referencia a dos tareas principales. La primera de ellas es la transferencia y entrega del producto desde el proveedor hasta el cliente en el sitio a ser utilizado. La segunda consiste en todas las operaciones de configuración que son necesarias para que el sistema esté listo para la activación[3].

Activación: Se refiere a la ejecución de aquellos componentes de un sistema que son necesarios para el funcionamiento del producto que se va a desplegar. Para una herramienta simple, la activación involucra el establecimiento de alguna forma de mando (hacer *clic* en el ícono gráfico) para la ejecución de componentes binarios de la herramienta. Para un sistema complejo, podría ser necesario iniciar servidores demonios[4] antes de que el sistema de *software* sea activado. Tener en cuenta que el proceso de instalación en sí mismo, puede requerir la invocación de otros instrumentos y posiblemente, la instalación de estos[3].

Desactivación: Constituye la actividad inversa de la activación, y se refiere al cierre de cualquier componente en ejecución de un sistema instalado. En general, la desactivación es necesaria antes de que otras actividades de despliegue se puedan llevar a cabo tales como la actualización y la desinstalación [3].

Desinstalación: En algún momento, un sistema en su conjunto ya no es necesario en un determinado sitio de consumo y puede ser eliminado. Se supone que la desinstalación está precedida por la desactivación. La actividad posiblemente implique una modificación de las configuraciones de otros sistemas, así como la retirada de los archivos pertenecientes a la aplicación que ha de ser desinstalada. La desinstalación no es necesariamente un proceso trivial. Una de las razones es que este proceso podrá asumir

[4]demonio (en inglés daemon, Disk And Execution Monitor): tipo especial de proceso informático que se ejecuta en segundo plano en vez de ser controlado directamente por el usuario (es un proceso no interactivo).

ser el único usuario de algún recurso compartido, como por ejemplo diversos archivos de datos, por lo que pueden liberar esos recursos y causar que otros sistemas puedan fallar [3].

Actualización: Esta actividad constituye un caso especial de la instalación. Es por lo general menos compleja debido a que muchos de los recursos necesarios ya se han obtenido durante el proceso de instalación. Normalmente, el ciclo de vida del despliegue incluye una secuencia repetida en el que un sistema es desactivado, se instala una nueva versión y este vuelve a ser activado. Para algunas aplicaciones, la desactivación puede no ser necesaria y la actualización se puede realizar al mismo tiempo donde una versión anterior esta aún activa. De manera similar a la instalación, la actualización incluye la transferencia de todos los componentes necesarios para completar la operación[3].

Todas estas actividades forman parte de un mismo proceso lógico. En el despliegue de un producto informático, la integración de cada una de las actividades anteriormente descritas es fundamental. En este punto se necesita de un mecanismo que acople todo este proceso lógico de una forma automatizada. La herramienta que responde a esta necesidad es conocida como instalador de *software*.

1.3 Herramientas para la creación de instaladores de *software*

En la actualidad existe una gran variedad de instaladores, cada uno de estos se ha construido sobre la base de tecnologías y herramientas que están en constante desarrollo. Se pueden encontrar bajo licencias comerciales, software propietario, o expuestas a las distintas comunidades de desarrollo como código abierto, libre o gratuito.

1.3.1 Herramientas privativas

Un popular formato de *Microsoft Windows* es el paquete de instalación **MSI**[5], el cual es instalado por la herramienta de empaquetamiento de *software* ***Windows Installer***. Diferentes empresas comerciales han desarrollado herramientas con el propósito de crear instaladores de aplicaciones para el sistema operativo *Windows* tales como ***InstallShield***, ***InstallAnywhere***, ***Wise, SetupBuilder***y ***Desktop Authority MSI Studio*** (su significado

[5]MSI, del inglés Microsoft Installer: los paquetes MSI sirven para poder instalar aplicaciones directamente en el equipo sin pedir interacción alguna por parte del usuario.

será explicado más adelante). La mayoría de estas herramientas pueden crear paquetes MSI, así como sus propios ejecutables.

Microsoft Windows Installer Service (WIS): Es un poderoso servicio que ha sido creado por *Microsoft* para ayudar a la gestión del ciclo de vida de una aplicación en el sistema operativo *Windows*. Un aspecto importante de *Windows Installer* es que brinda una serie de funcionalidades que han estado deshabilitadas en sistemas de instalación convencionales.[4]

InstallShield: Es el estándar de la industria para instalaciones con ***Windows Installers***e ***InstallScript***. Permite a los productores de *software* realizar instalaciones de alta calidad para las plataformas *Windows*, incluyendo Vista™, y ampliarlas para configurar servidores de bases de datos, servicios Web y dispositivos móviles. *InstallShield* está disponible actualmente en tres ediciones comerciales: de primera, profesional y *Express* inglés y japonés.[5]

SetupBuilder™: Ha sido diseñado para ofrecer una fusión de la instalación, el despliegue Web, la gestión de la configuración y las tecnologías de *script*. Su versión denominada *Developer Edition* proporciona características avanzadas para la instalación y actualización Web además de un depurador visual para la resolución de errores mediante el control de sus instalaciones, soporte a múltiples lenguajes. Su versión profesional es una herramienta indispensable para los desarrolladores que necesiten una suite de instalación más sofisticada con capacidad para la escritura de *scripts*.[6]

Incluye una funcionalidad para la creación de instaladores de parches con el objetivo de actualizar las versiones antiguas de las aplicaciones y muchas otras funciones avanzadas, incluyendo soporte a DLL[6]y el llamado a las API[7] de *Windows*, además de un editor de ficheros *script*. *SetupBuilder*™ es un rápido desarrollador de instaladores y una herramienta de gestión de la configuración para sistemas operativos *Windows*, tales como: *Windows Server* 2008 ("*Longhorn*"), *Windows Vista*, *Windows* 2003, *Windows XP*, *Windows* 2000, *Windows NT4* y *Windows Me*/98/95. Ofrece el conjunto completo de las características que se necesitan para construir instaladores y actualizadores robustos

[6]DLL, del inglés Dynamic Linking Library: término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda del programa por parte del sistema operativo.
[7]API, del inglés Application Programming Interface: es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

para este sistema operativo. Se trata de una potente, flexible, rápida e intuitiva herramienta basada en *scripts* utilizada en la gestión de configuración.[6]

Desktop Authority MSI Studio™: Es una aplicación de empaquetado e instalación de *software* y una solución optimizada en la gestión de instaladores para el uso de los administradores de sistemas. Cuenta con un conjunto completo de características para capturar, crear y editar archivos MSI. Presenta un mecanismo que controla el comportamiento de los paquetes de instalación en *Windows*. Además, la edición profesional permite probar y validar los paquetes sin que ello afecte a los sistemas de producción.[7]

InstallAnywhere: Posibilita a los desarrolladores de *software* de una manera rápida y fácil, instalaciones para los siguientes sistemas operativos*UNIX*, *Linux*, *Solaris*, HP-UX, Mac OSX, *Windows*. Ahorra tiempo y reduce los costos mediante la creación de un único archivo de proyecto que genere configuraciones fiables para cada una de las plataformas mencionadas. *InstallAnywhere* se encuentra disponible en dos versiones (*Standard* y *Enterprise*) y en cuatro idiomas (inglés, alemán, francés y japonés).[8]

1.3.2 Herramientas código abierto

Alternativas libres incluyen **NSIS, *Clickteam Install Creator, InnoSetup*, BitNami**, así como una herramienta de *Microsoft* llamado WiX. ***BitRock*** tiene una herramienta multiplataforma, llamado ***InstallBuilder***, que crea instaladores para *Windows*, Mac OSX, *Linux* y es gratuita para los proyectos de código abierto. La comunidad de desarrolladores de Java también se interesan en este aspecto y ha creado ***IzPack***, herramienta que también permite generar instaladores multiplataforma.

Dentro de las alternativas libres mencionadas **NSIS** (*Nullsoft Install scripts System*) es un sistema profesional de código abierto para crear instaladores sobre *Windows*. Está diseñado para construir instaladores pequeños y flexibles tanto como sea posible y, por lo tanto, es muy apropiado para la distribución por Internet. NSIS soporta diferentes idiomas, está basado en *scripts* y permite generar la lógica para manejar incluso las más complejas tareas de instalación. Muchos *plugins*[8] y *scripts* ya están disponibles, con los que se

[8]Plugin: aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica.

pueden crear instaladores Web, comunicarse con *Windows* y otros componentes de *software*, instalar o actualizar elementos compartidos.[9]

Clickteam Install Creator: Permite crear sistemas de instalación llevando todo el proceso paso a paso a través de un asistente. Estos pasos comprenden la selección de ficheros que incluye el programa, la elección del directorio donde se instalarán, los posibles acuerdos de licencia, hasta llegar al último paso, que compila la instalación y la deja lista para usar. *Clickteam Install Creator* permite una instalación personalizada, desde los textos y logos que aparecen hasta la inclusión de opciones adicionales como registro de componentes (DLL, OCX, REG y TLB) o creación de íconos de acceso directo a la aplicación. [10]

Inno Setup: *E*s otro instalador gratuito para programas de *Windows*. Presentado por primera vez en 1997, hoy *Inno Setup* compite e incluso supera muchos instaladores comerciales en cuanto a características y estabilidad. La instalación de los archivos: Incluye soporte integrado para "desinflar", bzip2, y 7-Zip LZMA/LZMA2 compresión de archivos. [11]

***Windows Installer XML* (WiX):** Es un conjunto de herramientas que construye los paquetes de instalación sobre *Windows* basado en ficheros XML[9]. Este conjunto de herramientas soporta un entorno de comandos que los desarrolladores pueden integrar en sus procesos de construcción para crear paquetes de instalación MSI y MSM. Fue liberado por *Microsoft* bajo una licencia de código abierto denominada *Common Public License[12]*.

BitRock: *E*s otra de las herramientas que permite crear instaladores multiplataforma que se pueden compilar para *Windows*, *Linux*, Mac, *FreeBSD*, *Solaris* (x86/Sparc), IRIX, AIX y HP-UX. Los instaladores generados tienen un nativo "*look-and-feel*" y no requiere de dependencias, puede funcionar en modo gráfico (GUI), en modo texto y en modo desatendido. Esta herramienta de instalación puede también generar paquetes RPM independientes[13].

IzPack: Es un creador de instaladores basado en tecnología Java. Es un proyecto *SourceForge* cuyo diseño modular permite definir cómo debe verse el instalador, además

[9]XML, del inglés Extensible Markup Language: metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

es muy configurable mediante un API. Dentro de sus características fundamentales se encuentra el soporte multiplataforma que le permite ejecutarse en cualquier sistema operativo que tenga instalada la máquina virtual de Java. Esta herramienta está bajo la licencia GPL (*GNU General Public License*), sus cláusulas dictaminan que puede modificarse, usarse y redistribuirse gratuitamente, o sea que se puede modificar el software y publicarse de forma gratuita pero con la versión modificada es obligatorio compartir también el código cambiado[14].

BitNami: Es un instalador multiplataforma, y con licencia GPL, de aplicaciones web de software libre. Es decir, proporciona instaladores para Linux, Windows y Mac OS y para este último, incluso proporciona en algunos casos versiones para PowerPC y para Intel. Su objetivo es facilitar la instalación y configuración de gran cantidad de aplicaciones web. Además instala todos los elementos que requiere el funcionamiento de la aplicación, como puede ser un servidor HTTP Apache, o una base de datos como MySQL. Funcionan de forma nativa o en virtual es decir permite la instalación de la pila directamente en el sistema o se puede ejecutar como una máquina virtual[15].

1.3.3 Selección de la herramienta para la creación del instalador de Quarxo.

Dadas las herramientas anteriormente descritas, a continuación se muestra una comparación de acuerdo a las principales características que requieren de atención. La plataforma de instalación, plataforma de desarrollo, el soporte a diversos lenguajes, licencia, tipo de archivos que se manejan, la presencia de componentes gráficos (GUI) son los principales criterios de comparación a medir entre los distintos tipos de herramientas.

Tabla 1. Comparación entre las herramientas para la creación de instaladores

Producto	Licencia	Plataforma de Instalación		Plataforma de Desarrollo		Multilenguaje	Modo de Instalación		Tipo de Archivo
		Linux	Windows	Linux	Windows		Texto	Gráfico	
SetupBuilder™	Comercial		X		X	X		X	Script
Desktop Authority MSI Studio	Comercial		X		X	X		X	Script
InstallAnywhere	Comercial	X	X	X	X	X	X	X	Script
InstallShield	Comercial	X	X	X	X	X	X	X	Script
MSI Studio	CPL		X	X	X	X	X	X	Script
Clickteam Install Creator	Libre		X		X			X	Script
Inno Setup	Open Source		X		X	X	X	X	Script
Windows Installer XML (WiX)	CPL		X		X			X	XML
BitRock	Open Source	X	X	X	X	X	X	X	XML
BitNami	GPL	X	X			X		X	Script
IzPack	GPL	X	X	X	X	X		X	XML

De acuerdo al contenido de la Tabla #1 se concluye que la totalidad de las herramientas mostradas cumplen con la mayoría de los parámetros propuestos. La diferencia radica en la licencia bajo la cual estos productos están publicados, el soporte para plataformas Linux, los modos de instalación y el tipo de ficheros que manejan dichas herramientas.

Es necesario resaltar que tanto el cliente de este producto de software, la Dirección General de Servicios Bancarios; como sus desarrolladores en la UCI, están interesados

en una aplicación completamente *Open Source*. Por lo anteriormente descrito se podrían descartar las herramientas cuyas licencias son comerciales y no son de interés para esta investigación, lo que deja solo dos propuestas dentro de las herramientas analizadas: *IzPack* y *BitRock*. Ambas herramientas tienen características similares como el soporte multiplataforma, soporte a múltiples lenguajes, el tipo de archivos que manejan (XML), la presencia de componentes GUI y la generación de una herramienta de desinstalación.

Tanto *IzPack* como *BitRock* soportan aplicaciones escritas en el lenguaje Java y la integración con herramientas de generación o automatización de código. BitRock crea Instaladores multiplataforma independientes del lenguaje y de la plataforma es decir pueden instalar aplicaciones escritas en cualquier lenguaje, incluyendo: Java, PHP, Perl, Python, Ruby, C/C ++ y .NET/Mono. Esto, unido a que Quarxo está desarrollado sobre el lenguaje Java hace necesario que la herramienta para crear su instalación sea BitRock.

1.3.4 Herramientas para la generación o automatización de código

Muchas de las herramientas anteriormente mencionadas están diseñadas para la integración con tecnologías de generación o automatización de código, compactando así el proceso de desarrollo de cada uno de los componentes que conforman en su totalidad los instaladores de aplicaciones. Los principales exponentes de estas tecnologías son ***Make*, *Apache Ant y Maven***.

Make: *E*s una herramienta de generación o automatización de código, muy usada en los sistemas operativos tipo *Unix*/*Linux*. Es una utilidad que permite definir reglas de dependencia entre ficheros. Aunque puede utilizarse para diferentes fines, está especialmente orientada a la compilación de código. El propósito de *Make* es determinar automáticamente las piezas de un programa que necesitan ser recompiladas y lanzar las órdenes necesarias para lograrlo. Por defecto lee las instrucciones para generar el programa u otra acción de un determinado fichero (*makefile*). Las instrucciones escritas en este fichero se denominan dependencias. La herramienta *Make* se usa para las labores de creación de un fichero ejecutable o programa, su instalación, la limpieza de los archivos temporales en la creación del fichero, entre otras funcionalidades, como la creación de documentos del formato *docbook*[10]y el mantenimiento del sistema. Todo esto

[10]docbook: dialecto de SGML (Standard Generalized Markup Language) que permite la escritura de documentación técnica.

lo realiza usando o creando los llamados *makefiles* que ejecuten cada una de estas tareas.[16]

Por otra parte, ***Maven:*** *E*s una herramienta para la gestión y compresión de proyectos Java. Estaba integrado dentro de *Jakarta* pero ahora es una tecnología de nivel superior de la *Apache Software Foundation*. Esta herramienta puede compilar un proyecto Java, ejecutar pruebas unitarias, generar paquetes (JAR[11], WAR[12], EARS o distribuciones en ZIP) y una serie de reportes. Una de sus características más importantes es su actualización en línea mediante servidores repositorios. *Maven* es capaz de descargar nuevas actualizaciones de las bibliotecas de las que depende el proyecto y de igual manera subir una nueva distribución a un repositorio de versiones, dejándola a disposición de todos los usuarios.[17]

Otro ejemplo de este tipo de herramientas es ***Apache Ant (Another Neat Tool)*** la cual es una tecnología basada en Java. Se utilizan en el desarrollo de software para transformar el código fuente y otros archivos de entrada en un formato ejecutable. *Ant* es similar a *Make* en cómo define las dependencias entre tareas de construcción, sin embargo, en lugar de implementarlas por medio de una plataforma específica, utiliza la plataforma Java. Con *Apache Ant*, se puede escribir un único fichero de construcción que opera constantemente sobre cualquier plataforma Java, lo que constituye su mayor fortaleza. Otros puntos fuertes a tener en cuenta son su simplicidad y su poderosa capacidad de ser extendido.[18]

Dada las características de estas tres herramientas para la generación o automatización de código, y teniendo en cuenta la comparación entre las distintas herramientas para la creación de instaladores de *software* anteriormente descrita se llega a la conclusión que la más aceptada para esta investigación es *Apache An por lo siguiente:*

- ✓ Es una tecnología basada en Java.
- ✓ Se puede escribir un único fichero de construcción
- ✓ Transforma el código fuente en un formato ejecutable.
- ✓ Su poderosa capacidad de ser extendido.

[11]JAR, del inglés Java Archive: tipo de archivo que permite ejecutar aplicaciones escritas en lenguaje Java
[12]WAR, del inglés Web Aplication Archives: archivo utilizado en el empaquetamiento de una aplicación Web.

1.4 Lenguaje de Modelado

El lenguaje de modelado es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar parte de un diseño de software orientado a objetos. Se utiliza en combinación con una metodología de desarrollo para avanzar de una especificación inicial a un plan de implementación y para comunicar dicho plan a todo el equipo de desarrolladores.

1.4.1 UML

El Lenguaje Unificado de Modelado (UML) es un lenguaje visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Está pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios.

El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticos, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar, pero está pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos[19].

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. La estructura estática define los tipos de objetos importantes para un sistema y para su implementación, así como las relaciones entre los objetos. El comportamiento dinámico define la historia de los objetos en el tiempo y la comunicación entre objetos para cumplir sus objetivos[19].

El modelar un sistema desde varios puntos de vista, separados pero relacionados, permite entenderlo para diferentes propósitos. UML también contiene construcciones organizativas para agrupar los modelos en paquetes, lo que permite a los equipos de

software dividir grandes sistemas en piezas de trabajo, para entender y controlar las dependencias entre paquetes, y para gestionar las versiones de las unidades del modelo, en un entorno de desarrollo complejo. Contiene construcciones para representar decisiones de implementación y para elementos de tiempo de ejecución en componentes[19].
UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguajes de programación, así como construir modelos por ingeniería inversa a partir de programas existentes[19].

Se utilizó UML a la hora de modelar los artefactos del sistema principalmente porque sus diseños se pueden implementar en cualquier lenguaje de programación que soporte las posibilidades de UML (principalmente lenguajes orientados a objetos) esto es debido a que permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos) y tener actualizado la estructura del proyecto con una visión en el diseño de más alto nivel.

1.4.2 BPMN

BPMN (Business Process Modeling Notación, Notación de Modelado de Procesos de Negocio) es el estándar más reciente para modelado de procesos del negocio y servicios web. BPMN es una notación necesaria para expresar los procesos de negocio en un único diagrama de proceso de negocio (Business Process Diagram – BPD) BPMN permite hacer un mejor uso de la gestión de procesos del negocio (BPM), ya que normaliza el método de notación que sirve como ayuda en la automatización de los procesos[20].

BPMN está dirigido a gerentes, directores, dueños de empresas, ingenieros de procesos, analistas de negocios, analistas de sistemas, administradores de proyectos, responsables de calidad y todo aquel que necesita definir, documentar y hacer más eficientes sus procesos de negocio con el estándar más avanzado y aceptado en el mundo.

UML toma un perfil orientado a objetos en el modelado de aplicaciones, mientras que BPMN toma un perfil orientado a procesos en el modelado de sistemas. Tiene un enfoque en procesos de negocio, UML se enfoca al diseño de software y por lo tanto ambas notaciones son totalmente compatibles entre sí.

Se utilizó BPMN para el modelado porque proveer una notación estándar que es fácilmente legible y entendible por parte de todos los involucrados e interesados del

negocio mediante diagramas muy simples con un conjunto muy pequeño de elementos gráficos, con esto se busca que para los usuarios del negocio y los desarrolladores técnicos sea fácil entender el flujo y el proceso del mismo.

1.5 Metodología de desarrollo

Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. La metodología indica cómo hay que obtener los distintos productos parciales y finales[21].

Una buena elección de la metodología a utilizar garantiza la calidad del producto y el buen desarrollo del proceso del software para que se arrojen los mejores resultados.
En el documento Propuesta de informatización se establecen las metodologías ágiles Scrum y Programación Extrema (XP), por las propiedades y características del sistema a desarrollar se propone a utilizar la metodología SXP.

1.5.1 Scrum - Programación Extrema

Se selecciona la metodología Scrum - Programación Extrema (SXP), que es la unión de las ya referidas Scrum y XP por las características del sistema a desarrollar. Para llevar a cabo el proceso de desarrollo del proyecto se tomarán en cuenta las mejores prácticas de ambas. [22]
Se usará Scrum para la planificación del proyecto. De la misma serán tomadas algunas prácticas para su implantación al igual que de la metodología XP, procurando que el proceso sea efectivo y eficiente.

Consta de 4 fases principales:[23]

- ✓ **Planificación-Definición**: Se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- ✓ **Desarrollo**: Se realiza la implementación del sistema hasta que esté listo para ser entregado.
- ✓ **Entrega**: Puesta en marcha.
- ✓ **Mantenimiento**: Se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición

de las historias de usuario (HU), Diseño, Implementación, Pruebas, de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, y existe una refactorización continua, que permite mejorar el diseño cada vez que se añade una nueva funcionalidad.

SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo. Esta metodología es aplicada a la presente investigación por la razón de que debe ser desarrollada en un corto tiempo y es un proyecto pequeño.[23]

Scrum se enfoca en las prácticas de organización y gestión, mientras que XP se centra más en las prácticas de programación. Esa es la razón de que funcionen tan bien juntas: tratan de áreas diferentes y se complementan entre ellas[24].

1.6 Herramienta de Modelado utilizada

A la hora de realizar un modelo informático destinado a aumentar la eficiencia y productividad de algún software determinado se tiene que utilizar herramienta de tipo CASE (Computer Aided Software Engineering), por sus siglas en inglés o (Ingeniería de Software Asistida por Computadora) en español.

Se puede definir a las herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software. Como es sabido, los estados en el Ciclo de Vida de desarrollo de un Software son: Investigación Preliminar, Análisis, Diseño, Implementación e Instalación[25].

Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida del software en tareas como realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores.

1.6.2 Visual Paradigm 8.0

En todas las empresas de desarrollo es importantes utilizar diagramas para documentar sus sistemas y aplicaciones para tener un aspecto más profesional. Visual Paradigm es una herramienta case que permite construir diagramas UML, ya que soporta el ciclo completo de vida del software, análisis y diseño orientado a objeto y dibujar diagramas de clases: código inverso, generar código desde diagramas y generar documentación. Tiene disponibilidad en múltiples plataformas y en múltiples versiones. Visual Paradigm es un producto galardonado en la realización de UML, facilita al diagrama visual y al diseño de sus proyectos y les brinda la posibilidad de integrar y desplegar sus aplicaciones empresariales y sus bases de datos subyacentes[26].

Para el desarrollo del modelado se utilizó la herramienta CASE Visual Paradigm ya que además de ser muy fácil su manejo se tiene un gran conocimiento de la misma y favorece la política de migración de la Universidad.

1.7 Instaladores de Software

En la actualidad existe un gran número de instaladores de software, es muy inusual que una aplicación se encuentre sin su respectivo instalador. Es necesario desplegar una aplicación sin incurrir en configuraciones demasiado engorrosas que no son de la incumbencia de los usuarios finales. Estos usuarios no están obligados a tener conocimientos avanzados para instalar un determinado producto.

1.7.1 Tipología

De manera informal un instalador es un programa de computación que instala archivos, tales como aplicaciones, controladores de dispositivos, u otro tipo de *software*, al ordenador [27]. De acuerdo a la gran variedad de aplicaciones en el mundo de la informática y a sus características fundamentales, en esta investigación los instaladores se clasifican en dos grandes grupos:

- ✓ Instaladores estándares.
- ✓ Sistemas de gestión de paquetes.

Los instaladores estándares están específicamente realizados para desplegar los archivos que contienen. Algunos de estos son de uso general y su función consiste en leer el contenido del paquete de *software* que va a ser instalado.

Un *paquete de software* es un conjunto de uno o más archivos que son necesarios para la ejecución de un programa de computación o añadir a las características de un programa ya instalado en uno o más ordenadores [28].

Linux y otros sistemas Unix suelen administrar miles de paquetes. Los sistemas de gestión de paquetes, también conocidos como gestor de paquetes o sistema gestor de instalación, son una colección de herramientas para automatizar el proceso de instalación, actualización, configuración y eliminación de paquetes de un ordenador. Los paquetes son las distribuciones de *software* y los metadatos, como el nombre completo, la descripción de su objeto, el número de versión, los proveedores, y una lista de las dependencias necesarias para que el sistema pueda funcionar correctamente. Tras la instalación, los metadatos se almacenan en una base de datos de paquetes locales. Un sistema de gestión de paquetes proporciona un método coherente para la instalación de una aplicación informática.

Los instaladores que predominan son para aplicaciones de escritorio. Estos son mucho más fáciles de desarrollar debido a que existe toda una gama de herramientas para su confección. En el caso particular de las aplicaciones Web, esta cantidad se ve mucho más restringida. Normalmente no existe una herramienta que automatice el proceso de instalación por completo. En algunos casos, si existe dicha herramienta, esta requiere de configuraciones manuales que hacen el proceso muy complicado. La mayoría de las veces los procesos de instalación para las aplicaciones Web se llevan a cabo por un personal especializado en las características propias de la aplicación.

1.8 Estado actual de la instalación de aplicaciones Web.

Originalmente las aplicaciones Web construidas por desarrolladores eran instaladas por los mismos desarrolladores o administradores de sistemas. Hoy, sin embargo, hay un número creciente de usuarios para la instalación de aplicaciones Web con limitadas habilidades y conocimientos. Hay miles de códigos abiertos de aplicaciones Web disponibles, pero no están normalizadas de una forma sencilla para los que van a instalarlas. Muchas de las aplicaciones Web más populares vienen con su propio programa de instalación, pero los usuarios aún están obligados a desempaquetar los ficheros de un archivo y visitar una dirección Web para ejecutar el programa de instalación. Esto puede ser simple para desarrolladores, pero para el usuario medio no lo es.

Además, el programa de instalación por lo general carece de los privilegios necesarios para realizar todas las operaciones y lograr así una instalación completa de la aplicación. Muchas veces, el programa de instalación pide al usuario cambiar los permisos sobre determinados archivos y carpetas. No sólo se trata de una petición irritante, sino que además para muchos usuarios está más allá de sus conocimientos. El usuario no debe realizar tareas adicionales después de ejecutar el instalador, es decir, la propia herramienta debe realizar todas las operaciones necesarias para instalar la aplicación Web.

Una aplicación Web se define como una aplicación de *software* entregada a usuarios desde un servidor Web a través de una red como la Red informática mundial (WWW) o una Intranet. Conceptualmente se puede encontrar también como una aplicación que requiere de un servidor Web para funcionar e interactuar con los usuarios vía navegador [29].

Servidor Web: Programa de *software* que se ejecuta como un servicio en un equipo que es responsable de brindar el contenido de la Web[30].

La forma común de instalar una aplicación Web en la actualidad consiste en seguir instrucciones que se brindan en la documentación de la misma. Muchas de estas aplicaciones incluyen un programa de instalación que facilita algunas de las tareas de instalación. Sin embargo, todavía hay una gran cantidad de trabajo manual que se debe hacer antes y después de usar una de estas herramientas.
En primer lugar, el usuario debe descargar el paquete de la aplicación, desempaquetarlo y seguir todas las orientaciones para la ejecución del mismo. Frecuentemente, un archivo de configuración debe ser modificado o creado antes de que el instalador pueda comenzar su ejecución. Una vez concluida su labor, a menudo hay tareas adicionales que deben ser llevadas a cabo para completar el proceso de configuración. La mayoría de las veces, los permisos de directorios deben ser modificados para que el servidor Web pueda escribir archivos a estos directorios entre otras configuraciones.

1.9 Conclusiones del capítulo.

En el presente capítulo se determinó con qué no se cuenta en la actualidad, con una herramienta de instalación o componente reutilizable adecuado para realizar la actividad de instalación de software de Quarxo. En los proyectos que se desarrollan en la

Universidad de las Ciencias Informáticas no existe solución para esta problemática. Se propone la creación de un instalador de software cuyas herramientas y tecnologías de construcción sean: *BitRock* y *Apache Ant.*

CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN PROPUESTA

2.1 Introducción

El presente capítulo aborda las principales características del sistema. Con este fin se realiza el modelado de los principales procesos del negocio, se especifican las personas relacionadas con el sistema, además son identificadas las funcionalidades que debe brindar a partir de la Lista de Reserva del Producto (LRP) y las descripciones de las historias de usuarios.

2.2 Características que debe tener la instalación del software Quarxo.

La instalación de Quarxo debe ser un proceso automatizado y parametrizable que facilite al usuario las configuraciones pertinentes. Para esto se debe contar con una herramienta que apoye y guíe todo el proceso.

Parámetros configurables

La herramienta necesita una serie de parámetros de los cuales depende la configuración e instalación de la aplicación:

- ✓ Parámetros de configuración de la base de datos:
- ✓ Usuario, contraseña, dirección URL[13]del servidor de base de datos. Esta última seccionada en: puerto, nombre del servidor y nombre.
- ✓ Configuraciones para el correcto funcionamiento del componente de impresión:
    - El componente para la impresión de archivos .pdf, está concebido sobre la base de facilitar la correcta impresión de los reportes en formato .pdf generados por Quarxo durante sus operaciones diarias. La idea básica de su funcionamiento es la de sustituir el envío del .pdf generado como un parámetro de la petición que finalmente invoca el código de los *applets*. El componente solventa dicha situación almacenando físicamente el fichero

[13]URL, del inglés Uniform Resource Locator: secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.

generado en recurso compartido de acceso restringido en la red local, al que posteriormente se accede para su impresión. El componente delega las tareas de impresión del archivo en aplicaciones externas especializadas en dichos fines, con lo cual se logra una calidad de impresión óptima.

- Los parámetros necesarios para el funcionamiento desde las estaciones clientes son los siguientes:
  - Tener instalados los visores de .pdf *Foxit Reader* y *Adobe Acrobat Reader*.
  - Tener en la variable de entorno *Path* del sistema operativo el camino a las carpetas donde se encuentran los binarios ejecutables de las aplicaciones anteriores.

En el lado del servidor, no se necesitan configuraciones adicionales.

✓ El servidor de aplicación Web Apache TomCat.

- Se configuran los parámetros de memoria a utilizar por la aplicación en la sección Java Options de la pestaña Java.
- Se configura para que el servicio del TomCat se inicie automáticamente al iniciar el sistema operativo.
- Se configura el Apache TomCat para que se pueda utilizar el canal seguro https.

2.3 Planificación por roles para la realización del instalador.

Rol	Responsabilidad del Rol	Nombre y Apellidos
Gerente	Es el encargado de tomar las decisiones finales, acerca de estándares y convenciones a seguir durante el desarrollo del software.	Yordan Remón Reyes
Cliente	El cliente participa en las	Hector Peña Torres

	tareas que involucran la lista de reserva del producto.	
Programador	Es el encargado de producir el código y escribir las pruebas unitarias.	Yordan Remón Reyes
Analista	Es el encargado de escribir las historias de usuario y las pruebas funcionales para validar su implementación.	Yordan Remón Reyes
Diseñador	Encargado del diseño del sistema. Máximo responsable de la realización del diseño y supervisa el proceso de construcción.	Yordan Remón Reyes
Probador	Es el encargado de ayudar al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo.	Yordan Remón Reyes
Arquitecto	Se vincula directamente con el analista y el diseñador debido a que su trabajo tiene que ver con la estructura y el diseño en grande del sistema.	Yordan Remón Reyes

2.5 Actores del instalador

Actores	Descripción
Usuario	Usuarios que acceden al software, que tienen la posibilidad de Instalar Maquinas Clientes, Servidor de Aplicaciones y Servidor de Base de Datos.

2.6 Definición de los requisitos

La especificación de los requisitos de software se realizó en conjunto con el cliente a través de las técnicas de captura de requisitos: tormenta de ideas y entrevistas realizadas al cliente. Se tuvo en cuenta todas las ideas planteadas, sacando así las más óptimas para realizar el software.

2.6.1 Requisitos Funcionales

Los requisitos funcionales son los que definen las funciones que el sistema será capaz de ejecutar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas[31].

2.6.1.1 Desarrollo de la instalación de las Máquinas Clientes

1. Mozilla Firefox 3.6.x
2. Máquina virtual de Java 6.0u25 o superior de la versión 6
3. Foxit Reader
4. Adobe Acrobat Reader

2.6.1.2 Desarrollo de la instalación del Servidor de Aplicaciones

1. Apache Tomcat 7.0.16
2. Máquina virtual de Java 6.0u25 o superior de la versión 6
3. DLL copiadas en el directorio de Windows para calcular CRC (cc3260.dll, mscoree.dll, Proyecto2.dll).

2.6.1.3 Desarrollo de la instalación del Servidor de Base de Datos

1. Microsoft SQL Server 2005 o superior.

2. Idera-SQLsafe de 64 bits. DLL copiadas en el directorio de Windows para la gestión de los usuarios de la aplicación (SabicDES.dll y SwSecure.dll).

Este programa permite:

- ✓ Ahorro de tiempo: backup más rápido que SQL nativa con compresión dinámica.
- ✓ Reducir los fallos: sin interrupción durante interrupciones de la red.
- ✓ Automatizar: completa, diferencial y copias de seguridad del registro de transacciones.
- ✓ Gestión de Empresas: escalable y centralizada de la consola y repositorio. Ahorra espacio: compresión inteligente entre versiones de SQL.

2.6.2 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Representan las características que hacen al producto atractivo, usable, rápido o confiable. Especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad[31].

- ✓ **Hardware**

  - **Para explotación del cliente:** Procesador Pentium IV o superior 2.0 GHZ o superior. 512 Mb de memoria RAM (recomendado 1Gb).40 GB de disco duro.

  - **Para explotación del Servidor de Base Dato:** Procesador Intel Xeon e5620 a 2.4 Ghz, 16 GB de memoria RAM, 1 TB de disco duro.

  - **Para explotación del Servidor de Aplicaciones:** Procesador Intel Xeon e5620 a 2.4 Ghz, 16 GB de memoria RAM,1 TB de disco duro.

✓ **Usabilidad**

- El software puede ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora.

2.7 Lista de Reserva del Producto (LRP)

Unas de las actividades más importantes definidas en la metodología SXP es la Lista de Reserva del Producto (LRP), en la cual se recoge en una lista priorizada todo el trabajo a desarrollar en el proyecto. Cuando un proyecto comienza es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo, suelen surgir los más importantes que casi siempre son más que suficientes para una iteración. Esta lista puede crecer y modificarse a medida que se obtienen más conocimientos acerca del producto y del cliente. Con la restricción de que sólo puede cambiarse entre iteraciones. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto.

Prioridad	Ítem	Descripción	Estimación	Estimado por
Muy Alta	1	Caracterización de las herramientas y metodología de desarrollo a utilizar para la realización del software.	4 semanas	Yordan
Muy Alta	2	Estudio del estado de arte de sistemas de notificación de eventos existentes a nivel nacional e internacional.	2 semanas	Yordan
Muy Alta	3	Implementación de la instalación del Mozilla Firefox 3.6.	3 semanas	Yordan
Muy Alta	4	Implementación de la instalación de la Máquina virtual de Java 6.0u25 o superior de la versión 6.	3 semanas	Yordan
Muy Alta	5	Implementación de la instalación de las variables de entorno Foxit	3 semanas	Yordan

		Reader y Adobe Acrobat Reader		
Muy Alta	6	Implementación de la instalación de Apache TomCat 7.0.16	3 semanas	Yordan
Muy Alta	7	Implementación de la instalación del Microsoft SQL Server 2005	3 semanas	Yordan
Alta	8	DLL copiadas en el directorio de Windows para calcular CRC (cc3260.dll, mscoree.dll, Proyecto2.dll).	2 semanas	Yordan
Alta	9	Idera-SQLsafe de 64 bits. DLL copiadas en el directorio de Windows para la gestión de los usuarios de la aplicación (SabicDES.dll y SwSecure.dll).	2 semanas	Yordan

2.8 Historias de Usuarios y Tareas de Ingeniería

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software, lo que equivaldría a los casos de uso en el proceso unificado. Las mismas son escritas por los clientes como las tareas que el sistema debe hacer y su construcción depende principalmente de la habilidad que tenga el cliente para definirlas. Son utilizadas como el único documento de requisitos que se genera. Son escritas en lenguaje natural, sin un formato predeterminado, no excediendo su tamaño de unas pocas líneas de texto.[32]

Tabla 2. HU_01 Instalar Mozilla Firefox.

Historia de Usuario	
Número: *HU_01*	**Nombre Historia de Usuario:** *Implementación de la instalación del Mozilla Firefox 3.6*
Modificación de Historia de Usuario Número: *Ninguna*	

Usuario: *Yordan Remón Reyes*	**Iteración Asignada:** *1*
Prioridad en Negocio: *Alta*	**Puntos Estimados:***3*
Riesgo en Desarrollo: *Alta*	**Puntos Reales:** *3*
Descripción: *Es el navegador Web utilizado para el cliente interactuar con la aplicación.*	
Observaciones:	

Tabla 3. HU_02 Instalación de la Máquina virtual de java.

Historia de Usuario	
Número:*HU_02*	**Nombre Historia de Usuario:** *Implementación de la instalación de la Máquina virtual de java 6.x.x.*
Modificación de Historia de Usuario Número: *Ninguna*	
Usuario: *Yordan Remón Reyes*	**Iteración Asignada:** *1*
Prioridad en Negocio: *Alta*	**Puntos Estimados:***3*
Riesgo en Desarrollo: *Alta*	**Puntos Reales:** *3*
Descripción: *Es de vital importancia para el funcionamiento de la aplicación*	
Prototipo de interface: *Anexo 1*	

Tabla 4. HU_03 Instalación de Apache TomCat.

Historia de Usuario	
Número:*HU_03*	**Nombre Historia de Usuario:** *Implementación de la instalación de Apache TomCat 7.0.16*
Modificación de Historia de Usuario Número: *Ninguna*	
Usuario: *Yordan Remón Reyes*	**Iteración Asignada:** *1*
Prioridad en Negocio: *Alta*	**Puntos Estimados:** *3*

Riesgo en Desarrollo: *Alta*	**Puntos Reales:** *3*
Descripción: *Es de vital importancia para el funcionamiento del Servidor Web*	
Prototipo de interface: *Anexo1*	

Tabla 5. HU_04 Instalación de Foxit Reader y Adobe Reader.

Historia de Usuario	
Número: *HU_04*	**Nombre Historia de Usuario:** *Implementación de las variables de entorno Foxit Reader y Adobe Acrobat Reader.*
Modificación de Historia de Usuario Número: *Ninguna*	
Usuario: *Yordan Remón Reyes*	**Iteración Asignada:** *1*
Prioridad en Negocio: *Alta*	**Puntos Estimados:** *3*
Riesgo en Desarrollo: *Alta*	**Puntos Reales:** *3*
Descripción: *Es de vital importancia para la impresión de los documentos en la institución.*	
Observaciones:	

Tabla 6. HU_05 Instalación de SQL Server 2005.

Historia de Usuario	
Número: *HU_05*	**Nombre Historia de Usuario:** *Implementación de la instalación de Microsoft SQL Server 2005*
Modificación de Historia de Usuario Número: *Ninguna*	
Usuario: *Yordan Remón Reyes*	**Iteración Asignada:** *1*
Prioridad en Negocio: *Alta*	**Puntos Estimados:** *3*
Riesgo en Desarrollo: *Alta*	**Puntos Reales:** *3*

Descripción: *Es de vital importancia para interactuar con el servido de Base de Dato*
Prototipo de interface: *Anexo 1*

Tabla 7. HU_06 Copiar las DLL del Servidor de Aplicaciones.

Historia de Usuario		
Número: *HU_06*	**Nombre Historia de Usuario:** *Copiar las DLL en el directorio de Windows para calcular CRC (cc3260.dll, mscoree.dll, Proyecto2.dll).*	
Modificación de Historia de Usuario Número: *Ninguna*		
Usuario: *Yordan Remón Reyes*		**Iteración Asignada:** *1*
Prioridad en Negocio: *Media*		**Puntos Estimados:***2*
Riesgo en Desarrollo: *Media*		**Puntos Reales:** *2*
Descripción: *Es de vital importancia para la seguridad del servidor de aplicaciones*		
Observaciones:		

Tabla 8. HU_06 Copiar las DLL del Servidor de Base Dato

Historia de Usuario		
Número: *HU_07*	**Nombre Historia de Usuario:** *Idera-SQLsafe de 64 bits. Copiar las DLL en el directorio de Windows para la gestión de los usuarios de la aplicación (SabicDES.dll y SwSecure.dll).*	
Modificación de Historia de Usuario Número: *Ninguna*		
Usuario: *Yordan Remón Reyes*		**Iteración Asignada:** *1*
Prioridad en Negocio: *Media*		**Puntos Estimados:** *2*

Riesgo en Desarrollo: *Media*	**Puntos Reales:** *2*
Descripción: *Es de vital importancia para la seguridad del servidor de Base Datos*	
Observaciones:	

2.9 Tareas de ingeniería

Las tareas de ingeniería son actividades que los programadores conocen que el sistema debe hacer. Deben ser estimables, y poder ser implementadas entre uno y tres días ideales. La mayoría de estas tareas se derivan directamente de las Historias de Usuario. Existen dos tipos de tareas de ingeniería las que provienen de las historias de usuario y las técnicas. [23]

Las tareas de ingeniería son aquellas que no son resultado del análisis de ninguna historia de usuario pero deben ser realizadas para que el sistema funcione. Cada tarea de ingeniería será comprobada a través de los casos de prueba y no tienen por qué ser comprendidas por el cliente.[23]

Tabla 9. Tarea de Ingeniería 01.

Tarea de Ingeniería	
Número Tarea: *01*	**Número Historia de Usuario:** *HU_01*
Nombre Tarea: *Implementar código para instalación del Mozilla Firefox 3.6*	
Tipo de Tarea : *Desarrollo*	**Puntos Estimados:** *3*
Programador Responsable: *Yordan Remón Reyes*	
Descripción: *Permite instalar el navegador web además de abrir ventanas emergentes y permitir que los script modifiquen la página.*	

Tabla 10. Tarea de Ingeniería 02.

Tarea de Ingeniería	
Número Tarea: *02*	**Número Historia de Usuario:** *HU_02*
Nombre Tarea: *Implementar código para la instalación de la Máquina Virtual*	

de Java.	
Tipo de Tarea : *Desarrollo*	**Puntos Estimados:** *3*
Programador Responsable: *Yordan Remón Reyes*	
Descripción: *Permitir instalar la máquina virtual de java y realizar las configuraciones pertinentes.*	

Tabla 11. Tarea de Ingeniería 03.

Tarea de Ingeniería	
Número Tarea: *03*	**Número Historia de Usuario:** *HU_03*
Nombre Tarea: *Implementar código para instalación de Apache TomCat.*	
Tipo de Tarea : *Desarrollo*	**Puntos Estimados: 3**
Programador Responsable: *Yordan Remón Reyes*	
Descripción: *Una vez instalado el Apache TomCat, se ejecuta el Monitor TomCat instalado y se le configuran los siguientes parámetros de memoria a utilizar:* • *Xms8000m* • *Xmx8000m* • *XX:PermSize=1500m* • *XX:MaxPermSize=1500m*	

Tabla 12. Tarea de Ingeniería 04.

Tarea de Ingeniería	
Número Tarea: *04*	**Número Historia de Usuario:** *HU_04*
Nombre Tarea: *Implementar código para la Implementación de Foxit Reader y Adobe Acrobat Reader.*	

Tipo de Tarea : *Desarrollo*	**Puntos Estimados:** *3*
Programador Responsable: *Yordan Remón Reyes*	
Descripción: *Permitir instalar Foxit Reader y Adobe Acrobat Reader. Agregarla como variables de entorno.*	

Tabla 13. Tarea de Ingeniería 05.

Tarea de Ingeniería	
Número Tarea: *05*	**Número Historia de Usuario:** *HU_05*
Nombre Tarea: *Implementar código para la instalación de Microsoft SQL Server 2005*	
Tipo de Tarea : *Desarrollo*	**Puntos Estimados:** *3*
Programador Responsable: *Yordan Remón Reyes*	
Descripción: *Permitir instalar Microsoft SQL Server 2005 y realizar las configuraciones pertinentes.*	

Tabla 14. Tarea de Ingeniería 06.

Tarea de Ingeniería	
Número Tarea: *06*	**Número Historia de Usuario:** *HU_06*
Nombre Tarea: *Implementar código para Copiar las DLL en el directorio de Windows para calcular CRC (cc3260.dll, mscoree.dll, Proyecto2.dll).*	
Tipo de Tarea : *Desarrollo*	**Puntos Estimados:** *2*
Programador Responsable: *Yordan Remón Reyes*	
Descripción: *Permitir instalar y correr las DLL de Seguridad en el servidor de aplicaciones.*	

Tabla 15. Tarea de Ingeniería 07.

Tarea de Ingeniería	
Número Tarea: *07*	**Número Historia de Usuario:** *HU_07*
Nombre Tarea: *Implementar código para Idera-SQLsafe de 64 bits y copiar las DLL en el directorio de Windows para la gestión de los usuarios de la aplicación (SabicDES.dll y SwSecure.dll).*	
Tipo de Tarea : *Desarrollo*	**Puntos Estimados:** *2*
Programador Responsable: *Yordan Remón Reyes*	
Descripción: *Permitir instalar y correr las DLL de Seguridad en el servidor de base de datos.*	

2.10 Plan de Releases

En este paso se define el plan de raleases e iteraciones para realizar las entregas intermedias y la entrega final. Tiene como entrada la relación de Historias de Usuario definidas previamente. Para colocar una historia en cada iteración se tiene en cuenta la prioridad que definió el cliente para dicha historia. Como resultado de la priorización de historias se llegó a la siguiente planificación:

Tabla 16. Plan de raleases.

Release	**Descripción de la Iteración**	**Orden de las HU a implementar**	Duración Total
Iteración 1	En esta iteración se desarrollarán las historias de usuario que tienen prioridad alta.	HU_01,HU_02,HU_03, HU_04,HU_05	15 semanas
Iteración 2	En esta iteración se desarrollarán las historias de usuario que tienen prioridad media.	HU_06,HU_07	4 semanas

2.12 Arquitectura de despliegue de Quarxo.

La distribución de Quarxo se concibe de la siguiente forma: un sistema instalado en cada una de las PCs cliente, en el servidor Web y en el de Base de datos (ver figura II).

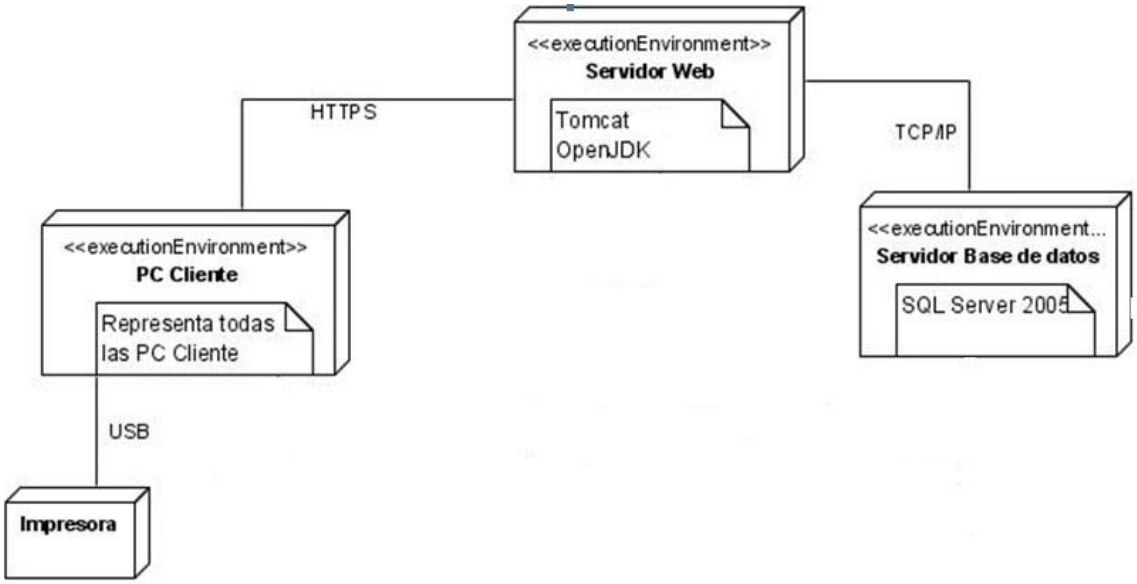


Figura II. Arquitectura de despliegue.

2.13 Arquitectura de software.

La aplicación está dividida en tres capas lógicas fundamentales:

- **Capa de presentación:** En esta capa se encuentran las vistas y la lógica de presentación. En la lógica de presentación se maneja todo el flujo Web utilizando la implementación del patrón Modelo Vista Controlador (MVC) y las vistas son los recursos que le permiten al cliente visualizar la información, estos pueden ser páginas HTML[14], documentos en formato PDF y hojas de cálculo.

- **Capa de servicios de negocio:** La presente capa encapsula toda la lógica de la aplicación en fachadas de negocio que son utilizadas por los controladores en la capa de presentación y se exponen algunos procesos de negocio a través de

[14]HTML, del inglés HyperText Markup Language: es el lenguaje de marcado predominante para la construcción de páginas Web.

interfaces de servicios. A estas fachadas de negocio se le aplican la seguridad a nivel de métodos y de objetos de negocio.

- **Capa de acceso a datos**: Maneja los objetos de acceso a datos abstrayéndolos del mecanismo de persistencia usado, a través de interfaces que exponen las operaciones de persistencia definidas para cada uno de los DAOs y que son implementadas utilizando *Hibernate*, que es un *framework*de Mapeo Objeto-Relacional (ORM).

La arquitectura definida presenta como otra de sus características la utilización del patrón MVC, patrón que propone dividir la aplicación en tres capas distintas, el Modelo, la Vista y el Controlador, potenciando la flexibilidad y la adaptabilidad a futuros cambios. Específicamente el Modelo es la representación de la información que maneja la aplicación, la Vista constituye la representación del modelo en forma gráfica disponible para la interacción con el usuario y el Controlador se encarga de responder a las solicitudes del usuario desde la interfaz, manejando los diferentes eventos a través de las funcionalidades necesarias y la información perteneciente al Modelo.

2.14 Proceso de la instalación de Quarxo en el BNC.

De acuerdo a las actividades del proceso de despliegue de software descritas en el Capítulo # 1 se han definido un conjunto de procesos. Dentro de cada proceso intervienen entidades que interactúan entre sí convirtiéndose en las entradas y salidas de las tareas que conforman a estos procesos. La figura # III representa cómo interactúan de manera general estos procesos. Ver Anexo # 1

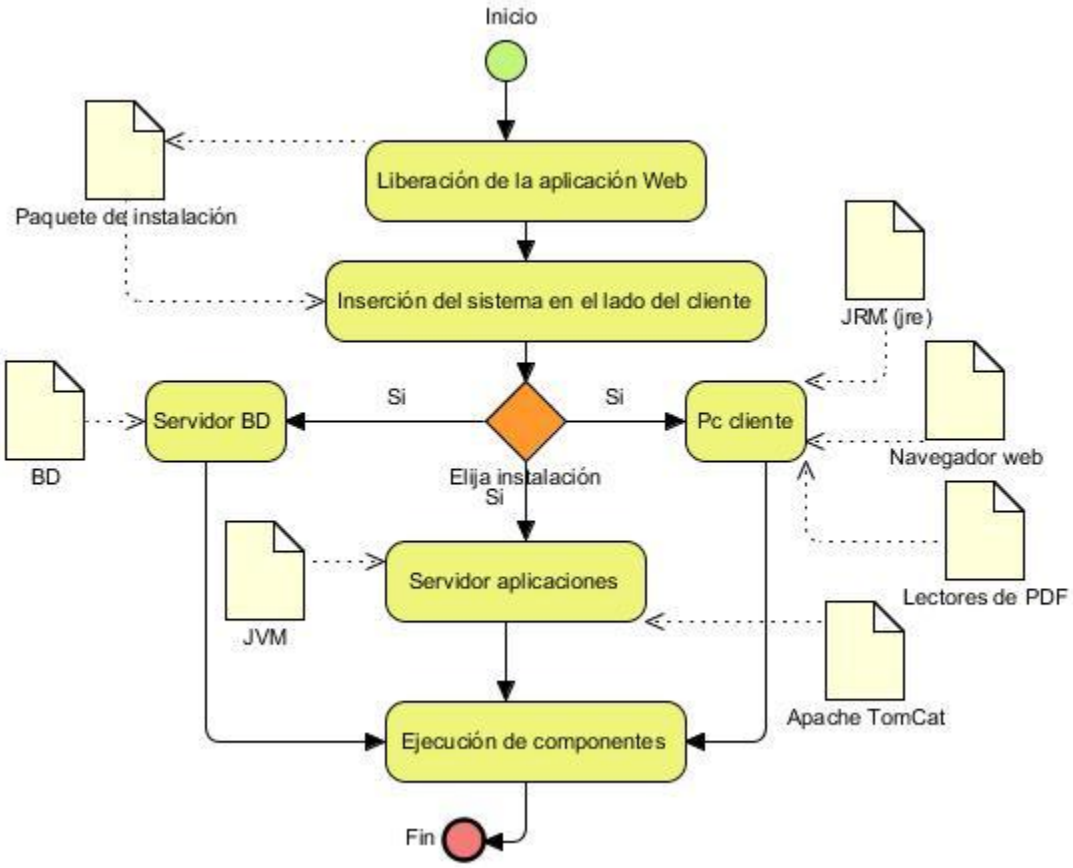


Figura III. Proceso de la instalación de Quarxo.

Existen un conjunto de entidades que transitan a través de los procesos mostrados en la figura# III. Todas se agrupan en una entidad general denominada *paquete de instalación,* la cual se describe a continuación:

Paquete de instalación: Medio de soporte en el cual van a estar agrupados los componentes de instalación de Quarxo en un sistema de archivos y carpetas. Su estructura está dada por los siguientes elementos:

- ✓ Máquina Virtual de Java: Plataforma sobre la cual se ejecutarán los componentes en el servidor.
- ✓ Servidor de aplicaciones: En este caso una versión superior a Apache TomCat 7.0.16.

- ✓ Manual de instalación de Quarxo: En este documento se explica cómo llevar a cabo la instalación de la Máquina Virtual de Java (JVM[15]), Microsoft Sql Server 2005y el TomCat.

2.15 Inserción del sistema en el lado del cliente.

La inserción del sistema en el lado del cliente está enmarcada en la actividad de instalación del modelo general de despliegue. Consta de dos etapas fundamentales: la transferencia o entrega del producto desde el proveedor hasta el cliente y las operaciones de configuración en el servidor de aplicaciones. El paquete de instalación constituye la entidad principal que nutre dicho proceso. Ver Anexos # 2, 3,4, 5, 6.

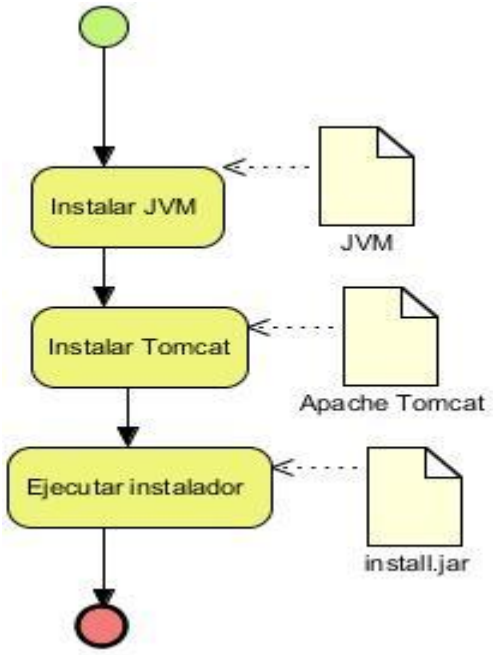


Figura IV. La instalación en el servidor de aplicaciones.

El flujo de tareas comienza con la transferencia de la herramienta hacia el lado del cliente. Esta tarea parte de la entrega del paquete de instalación a la entidad o persona responsable de la transferencia. En este punto el administrador del sistema debe realizar las siguientes verificaciones y configuraciones:

[15] JVM, del inglés *Java Virtual Machine*: Intérprete para el lenguaje de máquina de Java.

- ✓ Se procede a la instalación de la Máquina Virtual de Java 6.0.x, solo así se garantiza que la aplicación funcione correctamente.
- ✓ Instalar el servidor Web *Apache TomCat*7.0.16.

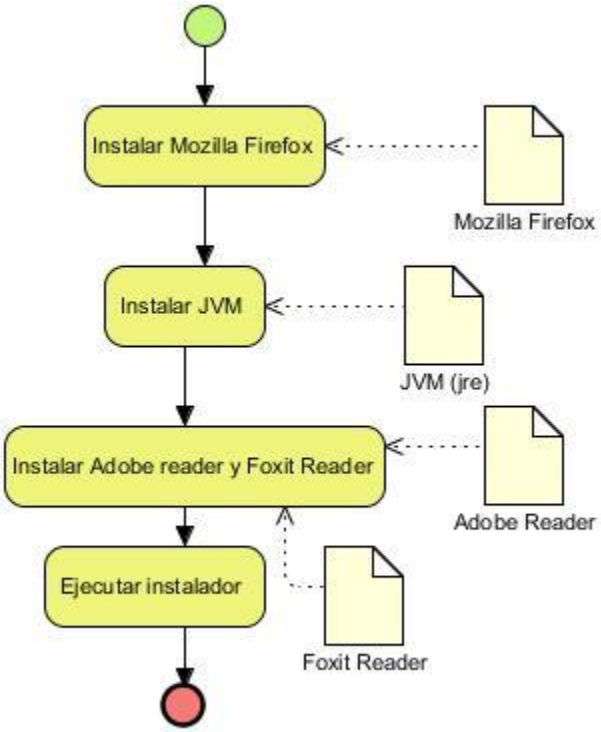


Figura V. La instalación en las máquinas clientes.

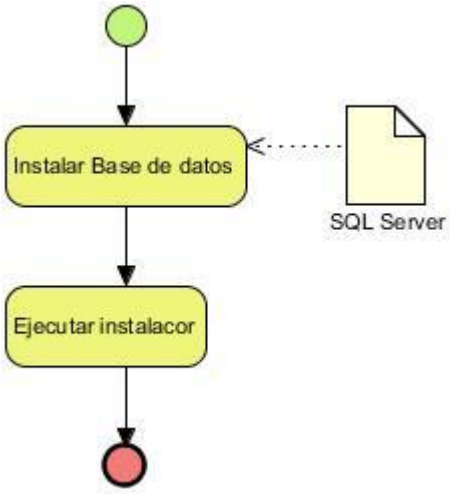


Figura VI. La instalación del servidor de base de datos.

Es necesario destacar que para cada instalación de los componentes mencionados anteriormente el administrador debe cumplir a cabalidad con los pasos incluidos en el

manual de instalación. Solo de esta manera, se garantizará la calidad requerida en la integración de estos componentes.

Luego de tener una correcta configuración, lo cual quiere decir que los elementos previamente configurados están en un estado funcional, el administrador del sistema puede proseguir con la ejecución de las tareas siguientes:

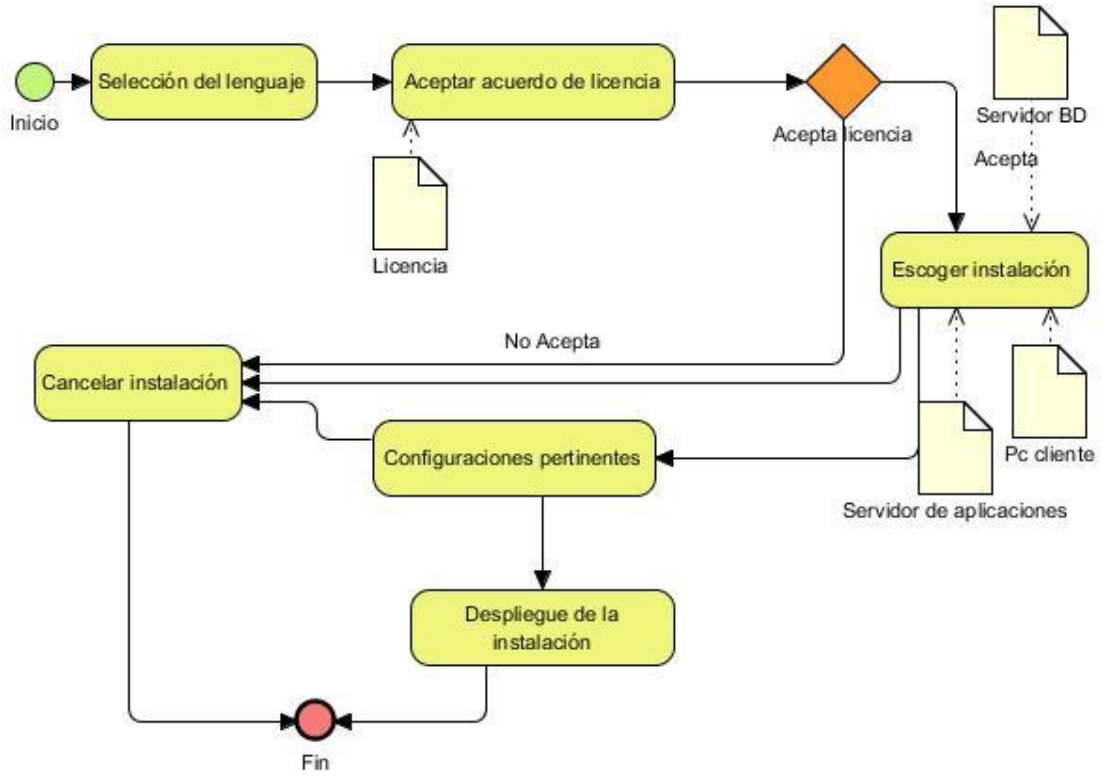


Figura VII. Instalación de la aplicación

La herramienta de instalación constituye un *Wizard*[16] que guía al administrador del sistema a través de un conjunto de ventanas de información y entrada de datos. La primera tarea está dada por la selección del lenguaje. Todas las informaciones en lo adelante se mostrarán en el lenguaje seleccionado. Seguidamente se presentan ventanas de información que incluyen la bienvenida y un conjunto de especificaciones técnicas que instruirán al usuario sobre las características generales del proceso que se lleva a cabo, los requerimientos del sistema y los problemas que se pueden presentar.

[16]Wizard: aplicación que ayuda al usuario a ejecutar una tarea en forma eficaz.

Con el objetivo de representar el contrato establecido entre los proveedores del *software* y el cliente, se muestra el acuerdo de licencia para el producto. La ventana que contiene a este, detendrá el flujo de actividades hasta que no se marque la opción de aceptación de licencia.

La configuración siguiente consiste en la selección de la ruta de instalación. Por defecto la ruta que debe mostrarse es la dirección donde está el servidor de aplicación *Apache TomCat* que dependerá de las variables de entorno y del sistema de archivos del sistema operativo, en este caso Windows Server 2003 de 64 bits.

Las configuraciones de la base de datos se deben introducir los parámetros que definen la conexión de la aplicación Web con la base de datos, tales como: usuario y contraseña para la autenticación, servidor, puerto y nombre para la construcción de la URL.

Para una verificación de cada parámetro se muestra una ventana con un sumario de todas las configuraciones introducidas. Esto permite el cambio de cada entrada en caso de haber ocurrido algún error.

Se procede entonces al despliegue de los archivos de instalación, mostrando en una barra de progreso el estado de la instalación hasta completar la misma. Cabe destacar que durante este proceso se genera la herramienta de desinstalación para la aplicación instalada.

De manera inusual, pero con una posibilidad real, la ejecución anteriormente descrita puede ser cancelada si el administrador de sistema lo estima conveniente.

2.15.1 Ejecución de componentes.

Este proceso es el encargado de garantizar que cada uno de los componentes que son requeridos para el funcionamiento de la aplicación estén ejecutándose. Hace referencia a la actividad de activación del modelo general de despliegue del Capítulo # 1. Una vez que la aplicación se encuentra en el servidor de aplicaciones, solo restaría reiniciar a este último. La tarea constituiría un paso inherente a la terminación del *wizard* de instalación y se ejecutaría de forma automática preferiblemente a través de un script.

2.17 Desinstalación de la aplicación.

Las tareas referentes a la retirada de la aplicación comienzan con la ejecución de la herramienta de desinstalación previamente generada en el proceso de inserción del sistema. Estas tareas se agrupan en dos secciones fundamentales: Detener componentes en ejecución: Si es necesario se procede a la detención de cada uno de los componentes que el sistema utiliza durante su ejecución. Este proceso hace referencia a la actividad nombrada desactivación descrita en el modelo general de despliegue del Capítulo # 1.

Eliminación de los archivos: Propio de la actividad de desinstalación (ver epígrafe 1.2.2), este proceso se encarga de la retirada de cada uno de los archivos inherentes a la aplicación de software que se desea eliminar. Ver Anexo # 7.

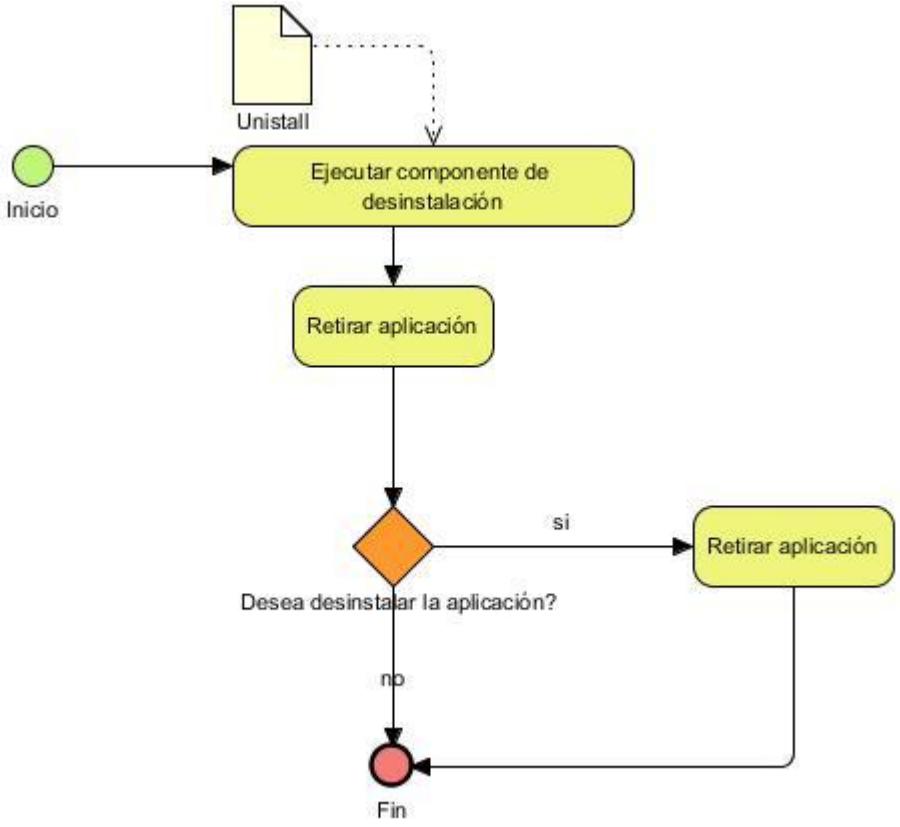


Figura VIII. Desinstalación de la aplicación

Existe una tarea que puede o no realizarse para este proceso en dependencia de las necesidades del usuario. Dicha tarea constituye la desinstalación de la herramienta de actualización.

2.19 Conclusiones del capítulo.

En este capítulo se ha adaptado el modelo de despliegue de software estudiado en el Capítulo # 1 para dar solución a la problemática existente en Quarxo. Además se especificaron las características que debe tener el mecanismo de instalación. Se desglosaron las Historias de Usuario que el desarrollador debe cumplir a través de una serie de actividades del ciclo de vida del producto y se realizó una descripción de las tareas de ingeniería a través de los diagramas necesarios según plantea la metodología.

.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA DE LA SOLUCIÓN PROPUESTA

3.1 Introducción

En el presente capítulo se hace una valoración de los principales artefactos que intervienen en la Fase de Prueba. Se describen las Pruebas Unitarias y de Aceptación que son los requeridos por la metodología SXP. Además, son realizadas las plantillas de las Pruebas de Aceptación para cada una de las Historias de Usuarios implementadas, ya que son las seleccionadas para la validación de las propuestas de solución además del estándar de programación utilizado para el desarrollo del sistema.

3.2 Estándar de Programación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez, con vista de realizar una codificación de alta calidad ya que la calidad del software es de suma importancia [33].

Enfatizar la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación. Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios [33].

- ✓ Son fundamentales cuando los programadores cambian de pareja o hacen *refactoring* del código de otros.
- ✓ Se consigue un código con el mismo estilo, homogéneo, legible.

3.3 Pruebas

En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos definidos en el análisis y diseño. La prueba no puede asegurar la ausencia de defectos, sólo puede demostrar que existen errores. [34]

En la metodología SXP las pruebas de funcionalidad se realizan a través de los casos de prueba. La metodología divide las pruebas en dos grupos: pruebas unitarias y las pruebas de aceptación.

3.3.1 Pruebas de Aceptación

El cliente con ayuda del probador define las pruebas de aceptación para cada Historia de Usuario a principio de cada iteración. Las pruebas de aceptación se utilizan para validar que cada requerimiento implementado funciona como se había especificado. Tienen como objetivo que las funcionalidades del sistema cumplan con lo que el cliente espera de ellas, permiten al cliente y al grupo de desarrollo conocer los problemas que presenta el sistema dándole la facilidad a los desarrolladores de corregirla.

3.3.1.1 Plantilla de Caso de Prueba Aceptación

Tabla 17. Caso de Prueba P1.

Caso de Prueba de Aceptación	
Código Caso de Prueba: *HU1,HU2,HU3,HU5_P1*	**Nombre Historia de Usuario:** *Probar funcionalidad para la instalación de los componentes*
Nombre de la persona que realiza la prueba: *Hector Peña Torres*	
Descripción de la Prueba: *Probar que se instalen los programas con las configuraciones pertinentes.*	
Condiciones de Ejecución: *Tener permisos de administrador*	
Entrada / Pasos de ejecución: *Se escoge los componentes a instalar y seguir los pasos según indican las ventanas.* *En el caso de la Base de Datos agregar usuario, contraseña y puerto. Además de permitir cargar el backup.*	
Resultado Esperado: *Se instala correctamente*	
Evaluación de la Prueba: *Satisfactoria*	

Tabla 18. Caso de Prueba P2.

Caso de Prueba de Aceptación	
Código Caso de Prueba: *HU4_P2*	**Nombre Historia de Usuario:** *Probar funcionalidad para la instalación de Foxit*

	Reader y Adobe Reader.
Nombre de la persona que realiza la prueba: *Hector Peña Torres*	
Descripción de la Prueba: *Probar que se instale Foxit Reader y Adobe Reader con las configuraciones pertinentes.*	
Condiciones de ejecución: *Tener permisos de administrador*	
Entrada / Pasos de ejecución: *Se escoge los componentes a instalar y seguir los pasos según indican las ventanas.*	
Resultado Esperado: *Se instala correctamente*	
Evaluación de la Prueba: *Satisfactoria*	

Tabla 19. Caso de Prueba P3.

Caso de Prueba de Aceptación	
Código Caso de Prueba: *HU6,HU_7_P3*	**Nombre Historia de Usuario:** *Probar funcionalidad para la instalación de las DLL.*
Nombre de la persona que realiza la prueba: *Hector Peña Torres*	
Descripción de la Prueba: *Probar que se instale las DLL con las configuraciones pertinentes.*	
Condiciones de ejecución: *Tener permisos de administrador*	
Entrada / Pasos de ejecución: *Se escoge los componentes a instalar y seguir los pasos según indican las ventanas.*	
Resultado Esperado: *Se instala correctamente*	
Evaluación de la Prueba: *Satisfactoria*	

En conclusión, las pruebas de aceptación realizadas arrojaron no conformidades que fueron resueltas en conjunto con los arquitectos del proyecto y los especialistas del BNC en dos iteraciones.

3.5 Validación de las variables

La investigación desarrollada plantea como idea a defender que: “Si se desarrolla una herramienta de instalación para el sistema Quarxo, se agilizarán las actividades referentes a los procesos de instalación de software permitiendo disminuir el tiempo de instalación y la complejidad de las actividades en la instalación del sistema”. A continuación se evalúan las variables disminuir el tiempo de instalación y la complejidad de las actividades en la instalación del sistema.

3.5.1 Disminución del tiempo de instalación

La disminución del tiempo hace referencia a la demora en la instalación de uno o varios componentes. La disminución de esta variable implicaría necesariamente mayor eficiencia y productividad, a partir de que se dedicaría menos tiempo en solucionar estas operaciones, y se agilizarían los procesos que esperan por su realización.

Se consideraron las actividades que deben realizar en la instalación del sistema. Esta comparación se resume en la tabla posterior.

Tabla 20. Actividades para la Instalación del sistema.

Actividades para la Instalación del sistema			
No	Actividades	Manual	Instalador
1	Mozilla Firefox 3.6.x	✓ Ejecutar el instalador y seguir los pasos según indican las ventanas. ✓ Activar el siguiente código (`dom.allow_scripts_to_close_windows true`) para que los Script modifiquen la página.	✓ Seleccionar el botón y dar aceptar.
2	Máquina virtual de Java 6.0u25	✓ Ejecutar el instalador y darle siguiente hasta el final.	✓ Seleccionar el botón y dar aceptar.

3	Foxit Reader& Adobe Reader	✓ Ejecutar el instalador y seguir los pasos según indican las ventanas.
		✓ Agregar como variables de entorno. (Agregar el path completo hasta la carpeta dónde está el **.exe*)
		✓ Escribir el siguiente código en C:\Program Files\Java\jre7\lib\security la seguridad.
		`grant codeBase "https://el ip de la máquina/" {`
		`permission java.security.AllPermission;`
		`}`
			✓ Seleccionar el botón y dar aceptar.

4	Apache TomCat 7.0.16	✓ Ejecutar el instalador y seguir los pasos según indican las ventanas. ✓ Se le configuran los siguientes parámetros de memoria a utilizar: ✓ Xms8000m ✓ Xmx8000m ✓ XX:PermSize=1500m ✓ XX:MaxPermSize=1500m	✓ Seleccionar el botón y dar aceptar.
5	Microsoft SQL Server 2005	✓ Ejecutar el instalador y seguir los pasos según indican las ventanas. ✓ Ejecutar SQL Server Configuration Manager/SQL Server 2005 Network Configuration/Protocols from MSSQLSERVER. ✓ Activar Named Pipes y TCP/IP	✓ Seleccionar el botón y dar aceptar. ✓ Usuario, contraseña, dirección URLdel servidor BD

6	DLL	✓ DLL copiadas en el directorio de Windows para calcular CRC (cc3260.dll, mscoree.dll, Proyecto2.dll).	✓ Seleccionar el botón y dar aceptar.

Escenario 1: Se realiza una prueba mediante la instalación de las máquinas clientes.

1. Especificaciones de hardware

- Procesador Pentium IV o superior 2.0 GHZ o superior. 512 Mb de memoria RAM (recomendado 1Gb).40 GB de disco duro.

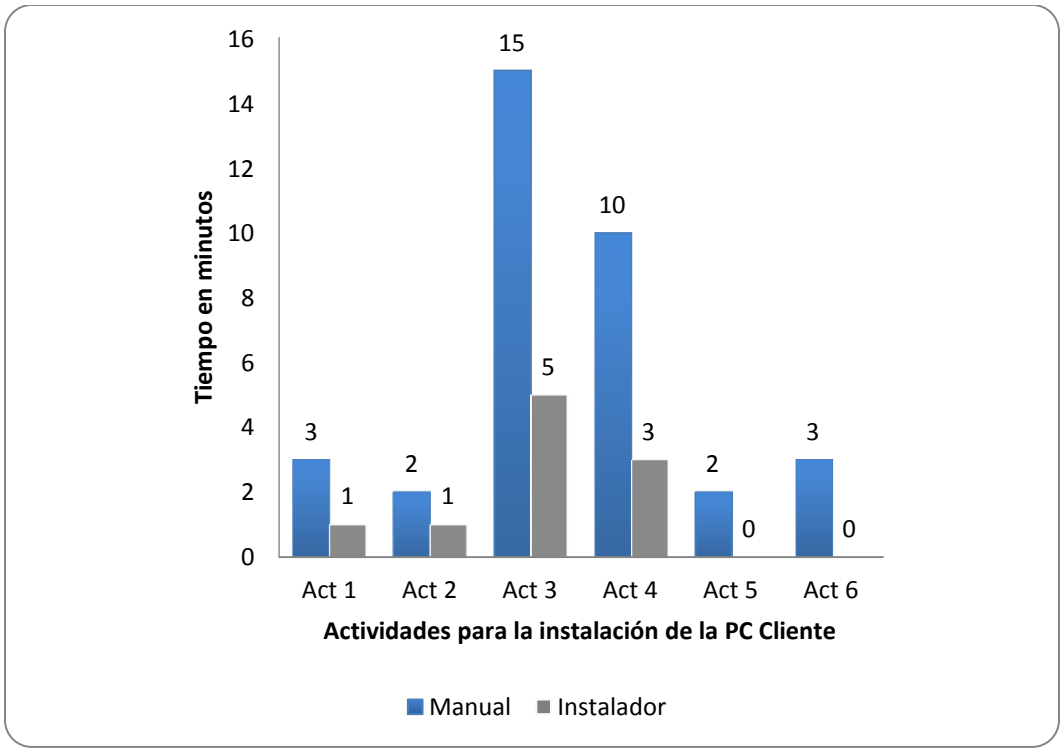


Figura IX. Comparación del tiempo de la instalación de una PC Cliente.

A través del estudio de las actividades realizadas para la instalación de las PC Cliente y su aplicación mediante el *Escenario 1*, se demuestra que con el instalador se acorta el tiempo de realización de estas operaciones, la disminución de la complejidad de las mismas y se elimina la posibilidad de incurrir en errores que conlleven a iniciar nuevamente la instalación.

Escenario 2: Se realiza una prueba mediante la instalación del Servidor de Aplicaciones.

1. Especificaciones de hardware

- Procesador Intel Xeon e5620 a 2.4 Ghz, 16 GB de memoria RAM, 1 TB de disco duro.

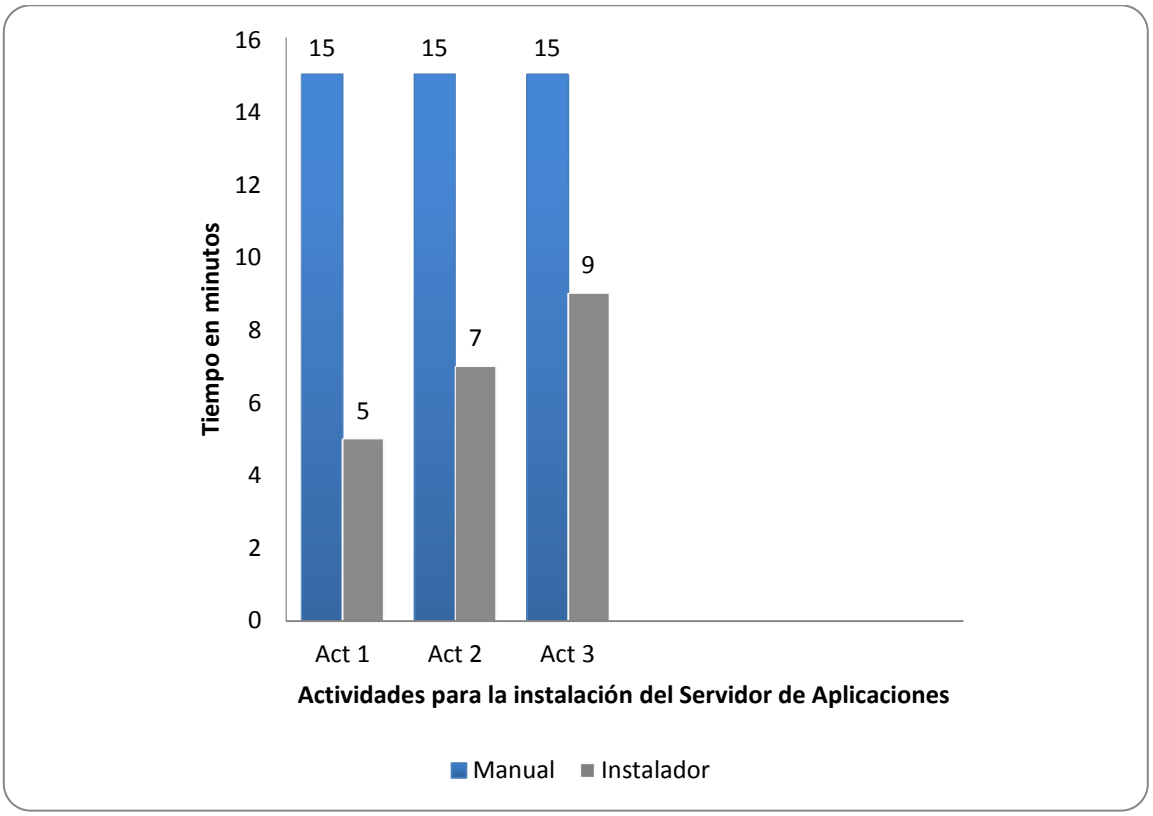


Figura X. Comparación del tiempo de la instalación del Servidor de Aplicaciones.

A través del estudio de las actividades realizadas para la instalación del Servidor de Aplicaciones y su aplicación mediante el *Escenario 2*, se demuestra que con el instalador se acorta el tiempo de realización de estas operaciones, la disminución de la complejidad de las mismas y se elimina la posibilidad de incurrir en errores que conlleven a iniciar nuevamente la instalación.

Escenario 3: Se realiza una prueba mediante la instalación del Servidor de Base Dato.

1. Especificaciones de hardware

- Procesador Intel Xeon e5620 a 2.4 Ghz, 16 GB de memoria RAM, 1 TB de disco duro.

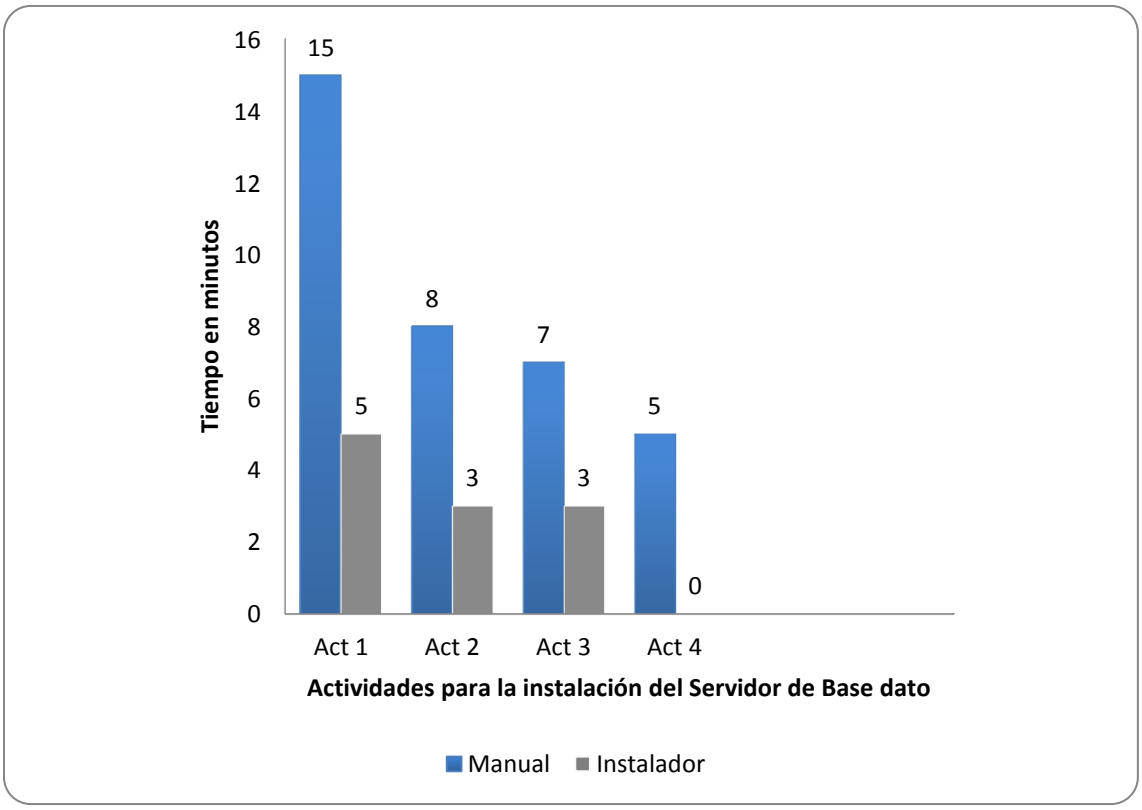


Figura XI. Comparación del tiempo de la instalación del Servidor de Base Dato.

A través del estudio de las actividades realizadas para la instalación del Servidor de Base Dato y su aplicación mediante el *Escenario 3*, se demuestra que con el instalador se acorta el tiempo de realización de estas operaciones, la disminución de la complejidad de las mismas y se elimina la posibilidad de incurrir en errores que conlleven a iniciar nuevamente la instalación.

3.6 Conclusiones del capítulo

En el presente capítulo se ha realizado la implementación del sistema y la descripción del flujo de pruebas propuesto por SXP de las funcionalidades descritas en las HU. Los casos de prueba son el mecanismo usado para asegurar que el sistema cumple con los requerimientos funcionales, asegurando así la calidad del software. De manera general las pruebas fueron satisfactorias, contribuyendo a obtener un producto de mayor calidad.

CONCLUSIONES

Dado el estudio presentado en esta investigación se obtuvo como resultado un mecanismo de automatización para la liberación Web y la herramienta de instalación de Quarxo, los cuales se integran perfectamente. El sistema fue representado por la metodología SXP. La modelación del negocio, la captura de requerimientos funcionales y no funcionales.

Los principales beneficios brindados por la solución propuesta fueron los siguientes:

- ✓ Obtención de la liberación Web de Quarxo completamente automática.
- ✓ El tiempo de las actividades del proceso de despliegue disminuyó considerablemente comparado con el método que se utiliza actualmente en el Banco Nacional de Cuba.

Por lo anteriormente expuesto se puede arribar a la conclusión de que el objetivo de este trabajo se ha cumplido con la realización de dicha herramienta.

RECOMENDACIONES

A partir del estudio realizado en la presente investigación, teniendo en cuenta las experiencias obtenidas a lo largo de su desarrollo, se propone las siguientes recomendaciones:

- ✓ Estudiar la viabilidad de aplicar el mecanismo propuesto en otros proyectos de la Universidad.
- ✓ Utilizar la solución como herramienta de instalación dentro del proceso de despliegue de Quarxo.

BIBLIOGRAFÍA

Richard S. Hall, Dennis Heimbigner, Alexander L. Wolf, A. Cooperative Approach to Support Software Deployment Using the Software Dock, 1998.

A. Carzaniga, A. Fuggetta, R.S. Hall, A. van der Hoek, D. Heimbigner, A.L. Wolf. A Characterization Framework for Software Deployment Technologies. Technical Report CUCS-857-98, Dept. of Computer Science, University of Colorado, April 1998.

Antonio Carzaniga, A. Characterization of the Software Deployment Process and a Survey of Related Technologies, September 1997.

Software Engineering Standards Committee of the IEEE Computer Society, IEEE Standard for Developing Software Life Cycle Processes, December 1997.

Arturo C. Arias Orizondo, Proyecto Técnico de Asesoría Especializada, Colaboración Médica Odontológica, Comunicación Institucional y Solución Tecnológica para apoyar la modernización del Sistema Penitenciario de la República Bolivariana de Venezuela, Caracas, Agosto 2006.

Julien Ponge, Elmar Grom, Fabrice Mirabile, Tino Schwarze, Klaus Bartz, IzPack Documentation.
programacion. [En línea] [Citado el: 8 de 12 de 2012.] www.programacion.com

Apache Software Fundation. [En línea] [Citado el: 10 de 4 de 2013.] http://www.apache.org/.

VP, Desarrollo. VParading. *freedownloadmanager.* [En línea] [Citado el: 20 de 1 de 2013.]
http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/.

Pressman, Roger S. *Ingeniería del software, un enfoque práctico.*

Aplicaciones Libres. [En línea] [Citado el: 5 de febrero de 2013.] http://www.aplicacioneslibres.com.ar/.

REFERENCIAS BIBLIOGRÁFICAS

1. *IEEE Standard for Developing Software Life Cycle Processes.*
2. A.Carzaniga, A.F., R.S. Hall, A. van der Hoek, D. Heimbigner, A.L. Wolf, *A Characterization Framework for Software Deployment Technologies.* 1998.
3. Carzaniga, A., *A Characterization of the Software Deployment Process and a Survey of Related Technologies.* 1997. .
4. Oney, W., *Programming the Microsoft® Windows® Driver Model*. 2010: Microsoft Press.
5. Curtis, B.A., *Cross-platform program, system, and method having a global registry object for mapping registry equivalent functions in an OS/2 operating system environment*. 2003, Google Patents.
6. Lupini, F., L. Pichetti, and A. Secomandi, *Wizard-based installation package with run-time debugging support*. 2004, Google Patents.
7. Honeycutt, J., *Microsoft® Windows® Desktop Deployment Resource Kit*. 2009: Microsoft Press.
8. Murta, L., et al., *Run-time Variability through Component Dynamic Loading.*XVIII Simpósio Brasileiro de Engenharia de Software, Sessão de Ferramentas, Brasília, DF, Brazil, 2004: p. 67-72.
9. Smith, P., *Software Build Systems: Principles and Experience*. 2011: Addison-Wesley Professional.
10. Saka, B., G. Ertek, and C.H. Türkseven, *Development of an interactive simulation of steel cord manufacturing for industrial engeneering education.* 2006.
11. Fox, J., *Getting Started With the R Commander: A Basic-Statistics Graphical User Interface to R.* Journal of Statistical Software, 2005. 14(9): p. 1-42.
12. Wilson, P., *The definitive guide to Windows installer*. 2004: Apress.
13. BitRock. *BitRock InstallBuilder Professional*. 2012 [cited 2012 octubre]; Available from: http://www.softpedia.es/programa-BitRock-InstallBuilder-Professional-122351.html.
14. Izpak. Available from: http://izpack.org/documentation/.
15. BitNami. *BitNami: instalador de aplicaciones web*. 2009; Available from: http://recursostic.educacion.es/observatorio/web/es/software/software-general/767-bitnami-instalador-de-aplicaciones-web.
16. Tropea, S.E., et al., *FPGAlibre: Herramientas de software libre para diseño con FPGAs.* FPGA Based Systems, 2006: p. 173-180.
17. Alonso Capuz, H., *Genereitor 2: Generador de código J2EE.* 2008.
18. Rodriguez, J.M.A., *DesarrolloAgil en J2EE con herramientas OpenSource.*
19. Dugarte, A. *Breve resumen de UML*. Available from: http://www.oocities.org/es/annadugarte/ads1/UML.htm.
20. *LENGUAJES DE PROGRAMACIÓN*. Available from: http://www.frt.utn.edu.ar/sistemas/paradigmas/lenguajes.htm.
21. Recopilación, d.a.a. *INGENIERÍA DEL SOFTWARE I.*; Available from: http://alarcos.inf-cr.uclm.es/doc/ISOFTWAREI/.
22. Beck, K., *Una explicación de la programación extrema. Aceptar el cambio.* Pearson Education, 1999.

23. Kniberg, H. *Scrum y XP desde las trincheras*

    2007; Available from: http://infoq.com/minibooks/scrum-xp-from-the-trenches
24. *in Metodologías Ágiles en el Desarrollo de Software.*
25. Scribd. *Capitulo I HERRAMIENTAS CASE*. Available from: http://es.scribd.com/doc/3062020/Capitulo-I-HERRAMIENTAS-CASE.
26. Sierra Cruz, D.S. 2007; Available from: http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml.
27. Lasso, I. [cited 2013 9 de enero]; Available from: http://www.proyectoautodidacta.com/comics/%C2%BFque-es-un-instalador.
28. *linux, Articulos y noticias de. microteknologias*. Available from: http://microteknologias.wordpress.com/2009/03/13/%C2%BFque-es-exactamente-un-paquete-de-software.
29. Mora, L. *Que es una AW*. Available from: http://ocw.ua.es/ingenieria-arquitectura/programacion-en-internet/03c-AplicacionesWeb.pdf.
30. Cavsi. Que es servidor web; Available from: http://www.cavsi.com/preguntasrespuestas/que-es-un-servidor-web.
31. Terminology, I.S.G.o.S.E., *Institute of Electrical and Electronics Engineers.*New York: IEEE Standards Board, 1900.
32. Romero, G.M.P., *MA-GMPR-UR2: Metodología ágil para proyectos de software libre.*2008.
33. *The Web Standards Project is a grassroots coalition fighting for standards which ensure simple, affordable access to web technologies for all.*; Available from: http://www.webstandards.org/.
34. *El único instrumento adecuado para determinar el status de la calidad.*Asociación de Técnicos, 2008.

GLOSARIO DE TÉRMINOS

1. API, del inglés Application Programming Interface: es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

2. DLL, del inglés Dynamic Linking Library: término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda del programa por parte del sistema operativo.

3. docbook: dialecto de SGML (Standard Generalized Markup Language) que permite la escritura de documentación técnica.

4. HTML, del inglés HyperText Markup Language: es el lenguaje de marcado predominante para la construcción de páginas Web.

5. IDE, del inglés Integrated Development Environment: programa compuesto por un conjunto de herramientas para un programador.

6. J2EE, del inglés Java 2 Platform, Enterprise Edition: Tecnología para desarrollar aplicaciones empresariales basadas en la Web.

7. JAR, del inglés Java Archive: tipo de archivo que permite ejecutar aplicaciones escritas en lenguaje Java

8. JVM, del inglés Java Virtual Machine: Intérprete para el lenguaje de máquina de Java.

9. MSI, del inglés Microsoft Installer: los paquetes MSI sirven para poder instalar aplicaciones directamente en el equipo sin pedir interacción alguna por parte del usuario.

ANEXOS

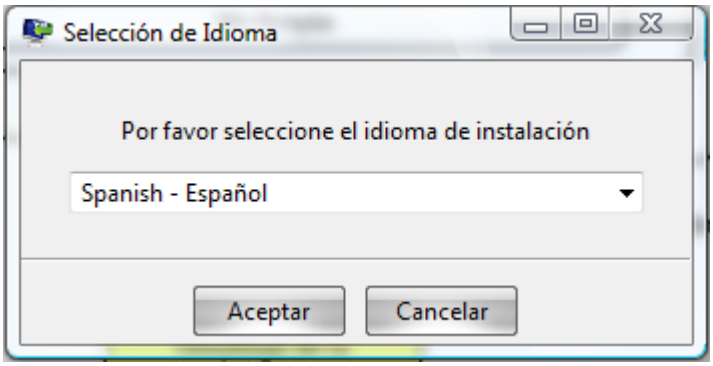
Anexo#1: Prototipo de interface.

Anexo# 2: Instalar Servidor de aplicaciones.

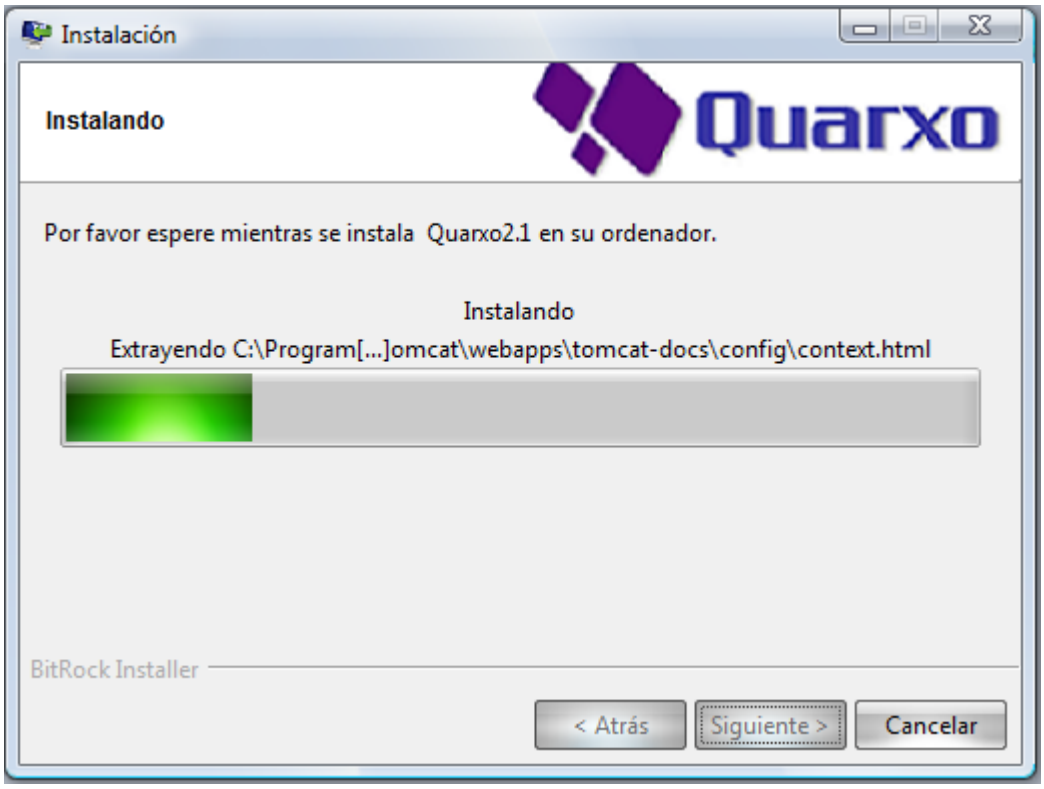
Anexo# 3: Instalar Máquinas clientes.

Anexo# 4: Instalar Servidor de base de datos.

Anexo# 5: Selección del lenguaje.



Anexo# 6: Despliegue de la instalación.



Anexo# 6: Desinstalación de la aplicación.