Universidad de las Ciencias Informáticas

Facultad 1

Instalador Multiplataforma para el Gestor de Documentos Administrativos eXcriba

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

Autores: Mabel G. Gutierrez Companioni

Julio C. Pompa Puente

Tutor: Dayelis Blanco Hernández

Co-Tutor: Marcel R. Sánchez Góngora

Ciudad de la Habana, Julio 2011

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la UCI los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año **2011**.

Mabel G. Gutierrez Companioni	Julio C. Pompa Puente
Firma del tutor Dayelis Blanco Hernández	Firma del co-tutor Marcel R. Sánchez Góngora

Resumen

En el departamento de Gestión Documental y Archivo se desarrolla una aplicación que se nombra Gestor de Documentos Administrativos eXcriba; teniendo en cuenta que la mayoría de los programas que se desarrollan y se distribuyen se apoyan de un instalador para quedar bajo correctas condiciones de uso, existe la necesidad de crear uno para esta aplicación en construcción. El objetivo es resolver la dependencia que existe entre el cliente y el equipo de desarrollo por el desconocimiento sobre el proceso con el que se realiza la instalación; para evitar la pérdida de tiempo a la hora de desplegar el sistema.

Teniendo en cuenta que una instalación exitosa es la condición necesaria para el funcionamiento correcto de cualquier software, se tomó como guía de observación el proceso de instalación del Gestor de Contenido Empresarial Alfreso. Este sistema logrará un impacto social por el hecho de ser sencillo, fácil de utilizar y manipular, muy conciso y directo, mientras que apoyará a la economía, fomentando la creación nacional de soluciones de softwares, dando lugar a la proliferación en popularidad y calidad, logrando con esto cerrar nuevos contratos con clientes satisfechos. Dentro del entorno político se demostrará la capacidad que poseen los profesionales que son forjados y cultivados en "*la mayor de las Antillas*" de desarrollar softwares tan potentes como los que se realizan en otros países; con el mismo grado de complejidad, robustez y calidad que estas soluciones comerciales, las cuales son reconocidas a nivel mundial.

Índice General

Introducción.

El surgimiento del hombre y sus necesidades por sobrevivir, dieron lugar a la premura de comunicarse para su desarrollo y progreso, es por eso que la escritura es una de las actividades más antiguas que existe y por lo que se hizo fundamental documentar por escrito para dar fe a los hechos cometidos como fijar actos, transacciones legales y comerciales, desencadenando el surgimiento de la gestión documental. Conseguir esto representó un nuevo salto en la complejidad y exigencias de los sistemas informatizados y en la forma de pensar de los administradores y archiveros.

La **Gestión Documental** es un "*área de la administración general que se encarga de garantizar la economía y la eficiencia en la creación, mantenimiento, uso y disposición de los documentos administrativos durante todo su ciclo de vida*"[1]

En el departamento de Gestión Documental y Archivo perteneciente a la UCI[1], existe un grupo de desarrollo que se encuentra implementando una aplicación de nombre GDA[2] eXcriba que tiene como objetivo *desarrollar productos y brindar servicios que cumplan con los requisitos de la gestión documental y satisfagan las necesidades del cliente*, esta aplicación usa como núcleo el ECM[3] Alfresco, personalizado por el grupo de desarrollo. El mismo permite unificar varios aspectos de la gestión y publicación de contenidos en una única solución informática.

[1] **UCI**: Universidad de las Ciencias Informáticas.
[2] **GDA**: Gestor de Documentos Administrativos.
[3] **ECM**: Gestor de Contenido Empresarial del inglés Enterprise Content Management.

La mayoría de los programas que se desarrollan y se distribuyen cuentan con un instalador que permite la correcta configuración e instalación de todos los componentes necesarios para el uso del sistema. En términos informáticos es:

Instalador: Software que permite realizar la instalación de los programas que se utilizan en las computadoras para trabajar. Este software, por lo general le solicita al usuario el lugar donde se copiarán y se instalarán los archivos correspondientes al programa que se instala, así como qué elementos se deben instalar, de manera que pueda personalizar la instalación.[2]

Instalar: Referido a software, agregar carpetas y archivos de programa al disco duro y los datos relacionados al registro con el fin de que el software se ejecute de forma correcta. Una instalación no es lo mismo que una actualización, en la que las entradas de los registros, la carpeta y los archivos de programas existentes se actualizan con una versión más reciente. Referido a hardware, conecta físicamente el dispositivo al equipo, carga en el equipo los controladores de dispositivo y configurar las propiedades y configuración de los dispositivos. Configurar y preparar el hardware o el software para que funcione. Muchos paquetes de aplicación tienen sus propios programas de instalación, los cuales copian todos los archivos requeridos desde los discos de distribución originales a los directorios apropiados en el disco duro y luego ayudan a configurar el programa para que cumpla con los requisitos específicos de operación. La mayor parte de los programas se instalan mediante un programa denominado *SETUP*.[2]

Una instalación exitosa es una condición necesaria para el funcionamiento correcto de cualquier software. Mientras más complejo sea el software, es decir, entre otras características, mientras más archivos contenga, mientras mayor sea la dispersión de los archivos y mientras mayor sea la interdependencia con otros softwares, mayor es el riesgo de alguna falla durante la instalación. Si la instalación falla aunque sea solo parcialmente, el fin que persigue la instalación posiblemente no podrá ser alcanzado. Por esa razón, sobre todo en casos de software complejo, el desarrollo de un proceso de instalación confiable y seguro es una parte fundamental del desarrollo del software.

Se destaca entonces la necesidad de implementar un proceso que complemente las pautas necesarias de un software para su distribución; siendo este la primera vista del usuario. "*Es de vital importancia ofrecer al usuario una experiencia agradable del software desde el primer momento, esto no solo dará lugar a la proliferación del software por popularidad y calidad, sino que además contribuye a cerrar nuevos contratos con un usuario satisfecho.*"[3] Por lo que han de enfocarse determinados recursos a la consecución de este fin. En la UCI se están dando los primeros pasos en lo que a Sistemas Instaladores o *Sistemas Diseñadores de Instaladores* se refiere, pero durante el proceso de investigación no se han encontrado casos puntuales documentados acerca de este tópico.

Dada la ausencia de un instalador multiplataforma para el GDA eXcriba en la UCI es que surge la idea de generar una solución que garantice un instalador multiplataforma para el GDA eXcriba, dando los primeros pasos de dicha tarea en el ámbito local, obteniendo como **problema científico**: *¿Cómo lograr un instalador multiplataforma adecuado para el GDA eXcriba?*. Siendo el **objeto de estudio** de la investigación los *Sistemas Diseñadores de Instaladores* y el **campo de acción** los *Sistemas Diseñadores de Instaladores Multiplataforma*. EL **objetivo general**: *Desarrollar un instalador multiplataforma para el GDA eXcriba*. Teniendo como **idea a defender**: *Con la existencia de un instalador multiplataforma con alto nivel de usabilidad se favorecerá la distribución del GDA eXcriba*.

Teniendo como tareas de investigación:

- Revisar la bibliografía referente al tema de instaladores multiplataforma.
- Realizar estudios sobre los principales sistemas de diseño de instaladores de aplicaciones multiplataforma.
- Realizar estudios sobre los diferentes escenarios en los cuales se puede encontrar el sistema GDA eXcriba.
- Realizar estudios sobre el sistema de instalación del ECM Alfresco.
- Analizar un sistema que realice la instalación del software, para el flujo de información empresarial.

- Diseñar un sistema que realice la instalación del software, para el flujo de información empresarial.
- Implementar un sistema que realice la instalación del software, para el flujo de información empresarial.
- Realizar estudio sobre los principales componentes y dependencias del sistema GDA eXcriba.
- Seleccionar metodología y tecnologías a usar durante el desarrollo del instalador.
- Generar los artefactos propuestos por la metodología durante el análisis del sistema GDA eXcriba.
- Generar los artefactos propuestos por la metodología durante el diseño del sistema GDA eXcriba.
- Diseñar los casos de prueba del instalador del sistema GDA eXcriba.
- Realizar pruebas al instalador del sistema GDA eXcriba.
- Realizar la implementación del instalador del sistema.

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

Analítico-Sintético: Este método permite analizar y comprender la teoría y documentación analizando instaladores multiplataforma con el objetivo de extraer elementos importantes que se relacionen con el objeto de estudio y así poder alcanzar conocimientos generalizados.

Histórico-lógico: Su utilización se basa en el análisis de la trayectoria y las etapas principales dentro de la investigación.

Modelación: A través de este método se hace tangible la idea del resultado esperando, dando una visión general de lo que se quiere lograr; enriqueciendo las perspectiva.

Observación: La utilización de este método permite estudiar otros trabajos que están estrechamente relacionados con los sistemas diseñadores de instaladores o tienen elementos en común con la investigación. También observa la situación real que se está investigando, permitiendo acercarse al objetivo final.

Experimento: Sobre la base de un conocimiento previamente establecido, se realizan diferentes pruebas como método principal de control sobre las diferentes etapas del producto en desarrollo.

Para su mejor comprensión, análisis y estudio, el presente documento ha sido estructurado en capítulos como se describe a continuación:

- **Capítulo I: Fundamentación teórica.**

  Incluye una investigación de los instaladores multiplataforma, a nivel internacional, nacional y de la universidad, además de las tendencias, técnicas, tecnología, metodologías y de software usados en la actualidad o en las que se apoya para la solución del problema.

- **Capítulo II: Características del sistema.**

  - Descripción del problema.
  - Objeto de automatización.
  - Propuesta del sistema.
  - Modelo de Dominio.
  - Especificación de los requisitos de software.
  - Definición de casos de uso.

- **Capítulo III: Análisis y diseño del sistema.**
  - **Análisis:**
    - Definición del modelo de análisis. Diagrama de clases de análisis.
    - Diagramas de interacción.
  - **Diseño:**
    - Diagrama de clases.
    - Descripción de las clases.
- **Capítulo IV: Implementación y prueba.**
  - **Implementación:**
    - Diagrama de despliegue.
    - Diagrama de componentes.
  - **Modelo de prueba:**
    - Descripción de cada uno de los casos de prueba.

Capítulo 1

Fundamentación Teórica.

Desde los inicios de la informática, esta siempre ha sido esencialmente para el uso de personas capacitadas, para poder realizar procesos informáticos de cualquier índole; las grandes máquinas con las que se trabajaba en lo que puede llamarse la *primera generación de ordenadores*, eran para uso exclusivo de científicos y/o investigadores que requerían de los últimos adelantos científico-técnicos para desarrollar su trabajo. Con el decursar del tiempo y el desarrollo inestable pero siempre ascendente de las tecnologías, estas condiciones indudablemente tenían que variar pues se hacía necesario, cada vez más, la inclusión de estas tecnologías en la sociedad y que dejaran de ser de uso exclusivo de científicos.

Siendo principalmente impulsadas por la mercadotecnia y la competencia las tecnologías han ido minando progresivamente el ámbito social, desde el personal no científico hasta el uso simplista en hogares y/o áreas de recreo. Luego de haber logrado que se expandiera el rango de acción tecnológico se hizo necesario que las personas pudieran manejar los artefactos de cómputo por sí solas y he aquí donde comienza a insertarse toda una gama de empresas de diversas envergaduras para tratar de solventar los problemas que esto conlleva, desde la creación de simples y avanzados sistemas de soporte técnico hasta el detalle que fue más obviado, pero el más importante en toda la gama de desarrollo: *los sistemas de instalación*.

La instalación del software es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados y configurado. Constituye la etapa final en el desarrollo propiamente dicho del software. Luego de ésta el producto entrará en la fase de funcionamiento, para el que fuera diseñado.

La instalación se define como "*software instalador*" a aquel que permite la realización del proceso para desplegar el programa que se desea tener instalado en el equipo sobre el cual se esté trabajando.

Los ordenadores son máquinas incapaces de realizar alguna operación por sí mismas y todas sus funcionalidades se hacen posibles a través de los softwares, quienes contienen las instrucciones para realizar las tareas, esto hace que los ordenadores se conviertan en máquinas maravillosas, lo que conlleva a que los usuarios tengan la necesidad de instalar programas y para aquellos que nunca han tenido esta experiencia suele causar temor. Esto es contrastante pues el proceso de instalación de un programa suele ser simple, dependiendo del sistema operativo y del software en cuestión.

Los usuarios del sistema operativo Microsoft Windows suelen hacer búsquedas de software y una vez que encuentran su objetivo lo descargan y lo instalan dando clic solo en *Next* o *Siguiente* dependiendo del lenguaje, más tarde al terminar el proceso de instalación proceden a ejecutar el programa. Hay que tener en cuenta que básicamente todos estos programas que se instalan son archivos **.exe*, o sea ficheros binarios, que con tan solo hacer doble click encima, se ejecutan.

Antes de comenzar la instalación de programas, es necesario revisar el espacio disponible en donde se desee realizar la misma. No se debe utilizar nunca la totalidad del espacio libre ya que muchos programas, para funcionar, necesitan crear archivos temporales y de no tener espacio para ellos, no funcionarán correctamente. Una vez concluida la instalación de forma correcta el sistema debe funcionar, *como es de esperar*, el ordenador habrá "*adquirido la habilidad*" de hacer nuevas cosas brindando más y mejores servicios.

1.1. Procedimientos para la instalación de programas.

Para instalar un programa, independientemente de su plataforma – *sea un sistema DOS o UNIX* – primero se necesita tener acceso al instalador o archivo binario del programa para luego proceder con su instalación dependiendo del sistema operativo: cada uno de estos tienen procedimientos descritos acerca de cómo ha de instalarse un programa; también lo referente a los programas previos que se requieren para el correcto despliegue/funcionamiento del que se desea introducir al sistema.

Un archivo binario es una archivo informático que contiene información de cualquier tipo, codificada en forma binaria para el propósito de almacenamiento y procesamiento de ordenadores. Por ejemplo los archivos informáticos que almacenan texto formateado o fotografías. Muchos formatos binarios contienen partes que pueden ser interpretados como texto. Un archivo binario que solo contiene información de tipo textual sin información sobre el formato del mismo, se dice que es un archivo de texto plano. Habitualmente se contraponen los términos "archivo binario" y "archivo de texto" de forma que los primeros no contienen solamente texto.[4]

1.1.1. Pasos de la instalación.

Para proceder a realizar una instalación no existen pasos predefinidos para ejecutar la misma, por la diversidad de formas en que puede hacerse y también la existencia de múltiples plataformas donde este proceso puede variar ligeramente o completamente, en cualquiera que sea la situación se han identificado una serie de aspectos a tomar en cuenta en este proceso, obviamente, no es una guía o un procedimiento; son pasos que se destacan en todos los procesos de este tipo:

Verificación de la compatibilidad: Se debe comprobar si se cumplen los requisitos para la instalación en cuanto a hardware y software. A veces es necesario desinstalar versiones antiguas del mismo software.

Verificación de la integridad: Se verifica que el paquete de software es el original, esto se hace para evitar la instalación de programas maliciosos.

Creación de los directorios requeridos: Para mantener el orden en el directorio cada sistema operativo puede tener un estandar para la instalación de ciertos archivos en ciertos directorios.

Creación de los usuarios requeridos: Para deslindar responsabilidades y tareas se pueden o deben usar diferentes usuarios para diferentes paquetes de software.

Concesión de los derechos requeridos: Para ordenar el sistema y limitar daños adjuntos a la seguridad, se le conceden a los usuarios el mínimo de los derechos o permisos.

Copia, desempaque y descompresión de los archivos desde el paquete de software: Para ahorrar ancho de banda y tiempo en la transmisión por internet o espacio de disco duro, los paquetes vienen empacados y/o comprimidos:

- Archivos principales, sean código fuente o binarios.
- Archivos de datos. – *por ejemplo datos generales, imágenes, modelos... etc.* –
- Documentación.
- Archivos de configuración.
- Bibliotecas.
- Enlaces duros o enlaces simbólico a otros archivos.

Configuración: Por medio de archivos de configuración se le da a conocer al software con que parámetros debe trabajar. Por ejemplo, los nombres de las personas que pueden usar el software, como verificar su clave de ingreso, la ruta donde se encuentran los archivos con datos o la dirección de nuestro proveedor de correo electrónico.

Definir las variables de entorno requeridas: Algunos comportamientos del software solo pueden ser determinados por medio de estas variables. Esto es parte de la configuración, aunque es más dinámica.

Registro ante el dueño de la marca: Para el software comercial a veces el desarrollador de software exige el registro de la instalación si se desea su servicio.

1.1.2. Instalando desde código fuente.

La instalación desde código fuente es muy común en sistemas tipo UNIX, comúnmente llamados POSIX, en estas circunstancias normalmente suele hacerse referencia a sistemas "GNU\Linux", "Distribuciones Linux" o "Distros" por la popularidad que han ganado los mismos en este aspecto.

Cuando se instala un programa desde código fuente, usualmente se han de seguir tres pasos. En el primero, un *script*[1] analizará su sistema, verificando que están instalados todas las dependencias necesarias para el éxito de la compilación y en ocasiones para realizar configuraciones previas al proceso de compilación. En el segundo, la aplicación se compila a partir del código fuente y se obtendrá como resultado final los binarios o código de maquina necesarios para que pueda ser usada dicha aplicación, esto solamente sucede con aplicaciones programadas en lenguajes de alto nivel –*C++, Java, C#...*– pues los lenguajes interpretados –*Bash, Perl, Python...*– no precisan ser compilados. En el tercero, la aplicación compilada, las bibliotecas y otros ficheros auxiliares serán copiados a los directorios apropiados de su sistema para que todos los usuarios puedan acceder a ellos. En el caso de programas muy simples, el primer paso puede no ser necesario; algunas otras aplicaciones están diseñadas para que usted pueda hacerlas funcionar sin necesidad de ejecutar el tercer paso.

Ya en posesión del código fuente del software que se desea instalar hay que extraerlo del archivo comprimido en el que suele venir, ya sea **.zip*, **.rar* o los más habituales: **.tar.gz* y **.tar.bz2*. Antes de compilar cualquier aplicación se precisa leer siempre cualquier fichero de documentación incluido en el archivo. Generalmente existe un fichero llamado "*INSTALL*" que contiene instrucciones específicas de instalación e información de la aplicación.

[1] **script:** Archivo de texto simple que contiene código, es generalmente interpretado, para la realización de tareas simples y comunes.

Pasos para la instalación desde código fuente:

Configuración. Para la mayoría de aplicaciones esta puede hacerse ejecutando el script **./configure** que se encuentra en el directorio del código fuente desempaquetado. La culminación con éxito del proceso de configuración puede ocurrir sin mensajes de error o con un resumen de cómo se compilará el programa, qué opciones serán habilitadas y posiblemente dónde será instalado. Programas más maduros y complejos tendrán usualmente varios parámetros opcionales para el fichero **./configure** que pueden ser usados para habilitar o deshabilitar funcionalidades del programa. Se puede usar la sentencia **./configure −−help** para obtener un listado de éstos parámetros opcionales y valorar su utilidad.

Compilación. Usualmente esta se lleva a cabo ejecutando el comando **make** en el mismo directorio donde se haya ejecutado **./configure**. En los softwares correctamente concebidos, si el *script* configure se completó con éxito, ésta etapa debería completarse sin problemas. El proceso de compilación se ejecuta tomando, por un determinado intervalo de tiempo, los recursos del equipo, normalmente haciendo mucho más uso del procesador – *dependiendo de la complejidad de la aplicación que se compila y de la velocidad de su ordenador, puede tardar desde algunos segundos hasta varias horas* – luego finaliza con un breve mensaje informando de la existencia de algún error o el éxito del proceso.

Las variantes de esta etapa del procedimiento son peculiares, pero ocasionalmente se necesitará ejecutar el comando **make** con algunos parámetros extras para compilar partes opcionales de la aplicación o ejecutar el comando de nuevo en un subdirectorio del árbol del código fuente. Tales variantes deberían estar destacadas en los ficheros "*README*" o "*INSTALL*".

Instalación. Si se ha completado con éxito la etapa de compilación, se puede optar por instalar la aplicación. Esto implica copiar los ficheros compilados a directorios donde estarán disponibles para su uso y – *si es apropiado* – los ficheros compartidos estarán disponibles para otras aplicaciones. Usualmente ejecutando el comando **make install**, con privilegios de usuario *root*[2], es posible realizar dicha instalación.

[2]root: Súper usuario en sistemas POSIX, posee el control – *dígase derechos o permisos* – de todo el sistema operativo.

En muchos casos la aplicación se instalará en el directorio **/usr/local**. Él o los ficheros ejecutables se ubicarán en la carpeta **/usr/local/bin**; las librerías, en **/usr/local/lib**; los ficheros de configuración se copiarán en **/usr/local/etc** y otros ficheros de datos se copiarán en **/usr/local/share**. Esto es resultado de una estandarización de estos procederes en los sistemas GNU\Linux, nunca se instalará algo en **/usr/local** ni tocará nada que esté en ese directorio, así que se puede confiar en que cualquier fichero que se encuentre en ese directorio es el resultado de algo que se haya compilado e instalado desde código fuente o es el resultado de un comando ejecutado manualmente. Siguiendo este "*protocolo*", nada de lo que se instale desde código fuente se enredará de ninguna forma con partes de su sistema que hayan sido instaladas por el gestor de paquetes de su Distribución Linux.

Una alternativa a **make install** es el programa **checkinstall**, previamente instalado el mismo. Acción que automáticamente genera un paquete binario, que podría ser instalado de diversas formas dependiendo de su distro, un binario "*.deb*" para Debian GNU\Linux y derivados y un "*.rpm*" para RedHat GNU\Linux y similares. Desde ese momento, se podrá usar el gestor de paquetes para controlar su instalación o desinstalación.

1.1.3. Instalando desde ficheros binarios.

En los últimos años se han desarrollado normas y técnicas cada vez más potentes para simplificar y estandarizar el proceso de instalación de software. Para la instalación de software se pueden aplicar las siguientes técnicas básicas:

- Los archivos son simplemente copiados en algún lugar del directorio. Este sistema es fácil e intuitivo y el preferido en MacOS X. Un riesgo es que versiones más antiguas hayan quedado abandonadas en algún otro lugar sin que se haga notable.
- Se instala primero un instalador, el que posteriormente instala el software deseado.
- El sistema operativo o algún software permanente se ocupan de instalar un paquete de software con todos los archivos requeridos. Al mismo se le conoce como Sistema de Gestión de Paquetes.

La instalación desde ficheros binarios resulta ser un tanto más sencilla que la instalación desde código fuente, basta solo con ejecutar el fichero para que el mismo, siendo un instalador, ejecute la instalación del software. Los binarios, pueden variar dependiendo de la plataforma, pero su ejecución es un paso "*casi*" omnipresente en este proceso.

Sistemas Microsoft Windows. En estos sistemas se dispone de un fichero binario – *comúnmente de extensión ".exe" o ".msi"* – que basta solo con ejecutar el mismo para poner en funcionamiento el *wizzard*[3] que guía el proceso de instalación, en el mismo se le notifica si se necesita algún software adicional o simplemente ofrece su instalación.

Sistemas *NIX.[4] En estos sistemas existen ficheros binarios – *los más comunes son los ".bin", ".run", ".sh"...etc.* – que pueden ejecutarse para poner en marcha un asistente que ayuda a obtener el resultado esperado de la instalación; las dependencias – *programas necesarios para el funcionamiento del software* – son gestionadas de diversas formas, dependiendo de la Distribución Linux, lo común resulta ser que este tercer programa tiene requerimiento crítico de instalación, de forma que hasta que este no esté en el sistema no es posible instalar el software deseado.

Existen otros binarios nativos que proveen un asistente para la instalación – *esto está en dependencia del sistema operativo, pueden ser ficheros ".deb" para Debian GNU\Linux y derivados, o ".rpm" para RedHat GNU\Linux y derivados* – pero que su manejo/instalación están en dependencia del sistema de gestión de paquetes utilizado.

[3] **Wizard**: Aplicación que ayuda al usuario a ejecutar una tarea en forma eficaz.

[4] ***NIX**: Referencia a los sistemas tipo UNIX, también llamados POSIX – *dígase también de sus derivados* –.

1.2. Sistemas diseñadores de instaladores multiplataforma.

Un *Sistema Diseñador de Instaladores* es un software que permite la creación de instaladores de forma rápida y sencilla, apegados a las necesidades del usuario, en la mayoría de los casos – *dependiendo del software utilizado* –, ofrece facilidad de uso, provee un ahorro de tiempo considerable; además tiene soporte para la modificación de los ficheros de configuración de la aplicación que se está diseñando; posee soporte además para el manejo de los archivos de apoyo – *bibliotecas, documentación...etc.* – generalmente ofrece opciones para instalar sólo ciertos módulos y también brinda al usuario una acción de desinstalar; opcionalmente cubre las necesidades básicas para un sistema multilingüe. En la presente situación se requiere del soporte multiplataforma, por lo que se valoran solo aquellos diseñadores que soportan la mayoría de las plataformas, una instalación multiplataforma que genere instaladores nativos.

Durante todo el proceso de investigación se encontraron los siguientes programas, se destacan por sus marcadas características para ser evaluados en el proceso de selección, para su posterior utilización en aras de lograr el objetivo trazado; los mismos son los que más necesidades solventan para ofrecer una solución al equipo de trabajo del GDA eXcriba.

1.2.1. Install4j.

Install4j es un poderoso constructor de instaladores Java multiplataforma, *herramienta comercial*, que genera instaladores nativos y accesos directos para aplicaciones Java. Install4j sobresale por su facilidad de uso.[5]

Características de Install4j:

- **Excepcional facilidad de uso:** Algunos sistemas instaladores requieren ser utilizados por un experto en la construcción de instaladores. No ocurre esto con install4j, donde todas las etapas de configuración son intuitivas y muy explícitas. El IDE[5] de

[5]**IDE:** Entorno Integrado de Desarrollo del inglés Integrated Development Environment.

install4j guía al usuario a través del proceso de recopilación de información. Construir un instalador es una tarea que consta de pocos minutos.[5]

- **Instalador estético:** El instalador es lo primero que un usuario percibe de una aplicación. Lograr una buena experiencia de la instalación es un paso importante en la creación de aplicaciones exitosas y la fidelización de los usuarios. Con install4j, los instaladores quedan muy estéticos y son transparentes para el usuario independientemente de la plataforma.[5]

- **Vista enriquecida y sistema de acción:** Con install4j es posible configurar el flujo de vistas para el instalador y el desintalador de cualquier forma que se desee. Install4j provee varias vistas y acciones para una amplia variedad de casos de uso y permite crear acciones propias en el IDE. Las acciones pueden ser conectadas a cada vista, haciendo de install4j un sistema de programación visual mucho más poderoso que los constructores tradicionales de instaladores.[5]

- **Auto-actualizadores totalmente flexibles y aplicaciones a medida:** Install4j ofrece varias plantillas para los actualizadores que permiten crear una solución de auto-actualización para la aplicación en desarrollo con solo unos clic. Los auto-actualizadores son totalmente configurables, para poder cubrir las especificidades de la aplicación en un gran número de escenarios. En términos más generales, install4j apoya la creación de aplicaciones de instalación personalizada que se adjuntan a la solicitud.[5]

- **Simple creación de pantallas personalizadas:** La consulta al usuario de información específica para la aplicación no tiene porque ser un proceso difícil. Además de programar a medida sus vistas, install4j incluye un concepto único de vistas de forma que son fáciles de configurar y son de gran aceptación en su aspecto. Con su poderoso sistema variable install4j posibilita el acceso a la información introducida en otras vistas y acciones. La creación de una interfaz de usuario no es posible hacerla más fácil.[5]

- **Generación de lanzadores nativos:** Install4j genera lanzadores para las aplicaciones que son nativos en todas las plataformas soportadas: Windows se compila con la tecnología exe4j, con el estado de la Shell UNIX se crea una secuencia de comandos para

dicha plataforma junto con los paquetes de aplicaciones hacen mezcla perfecta para Mac OS X. Estos lanzadores ofrecen la más flexible detección de la JRE[6] del mercado y se integran con muchas otras características in install4j.[5]

- **Avanzada integración con la JRE:** Uno de los requisitos básicos para un instalador JAVA es la habilidad de integrarse con la JRE. Con características como la descarga a la orden, la compresión Pack200 o instalaciones comunes de paquetes de JRE.[5]

- **Excelente soporte multiplataforma:** Implementar una aplicación multiplataforma es una tarea compleja. Install4j facilita esta carga con órdenes de cierta magnitud. Se define un instalador común para todo el proyecto y se especifica la información detallada para una plataforma en la vista. Inclusive si se desea solo la edición para Microsoft Windows, se pueden seleccionar las opciones para dejarlas disponibles para múltiples plataformas.[5]

- **Amplio soporte de i18n:** Install4j soporta completamente la localización de su instalador en múltiples idiomas. Ofrece gran cantidad de idiomas para la instalación y un mecanismo fácil para adicionar nuevas claves para mensajes propios en el instalador. Puede hacerse instaladores en un solo idioma o en varios que detectan el lenguaje base del sistema en ejecución.[5]

- **Extensibilidad:** Install4j es extensible. Con un modelo de extensión basado en JAVA Beans y una muy bien recomendada API[7] es posible agregar acciones propias, vistas y componentes de formularios a los registros de componentes de install4j. Con esta flexibilidad, es posible integrar rápidamente el código en el programa de instalación o crear extensiones que puedan reutilizarse en múltiples proyectos.[5]

[6] **JRE:** JAVA Runtime Environment

[7] **API:** Interfaz de Programación de Aplicaciones del inglés Application Programming Interface.

1.2.2. InstallShield.

InstallShield es una herramienta de software hecha en Java para crear instaladores o paquetes de software, para plataformas múltiples. Fue hecho originalmente por ZeroG Software hasta que fue adquirido por Macro visión a mediados de junio de 2005. Existen dos versiones del mismo: el estándar – *community* – y el empresarial – *enterprise* –.

InstallShield se utiliza sobre todo para instalar software de escritorio y las plataformas de servidor de Microsoft Windows, pero también se puede usar para administrar aplicaciones y paquetes de software en una amplia gama de dispositivos móviles y portátiles. La versión InstallShield 2010 fue puesta en venta el 18 de junio de 2009. InstallShield 2010 soporta Windows 7, Windows Server 2008 R2, MSI 5 y es el único instalador que soporta Microsoft App-V, que es el formato de virtualización de aplicaciones más extendido. Hay unos 71.000 vendedores independientes de software y clientes empresariales que utilizan InstallShield para realizar instalaciones en las plataformas Windows.[6]

Además, existe una versión del InstallShield Wizard para GNU\Linux, Sun Microsystems lo utilizaba para sus instalaciones de aplicaciones en GNU\Linux, como por ejemplo Java y NetBeans al momento de ser adquirida por Oracle.[6]

Características de InstallShield:

- Existe soporte para más de siete nuevas plataformas, incluyendo las últimas versiones de SuSE 9, RedHat EnterpriseServer 4.0 y Fedora, Solaris 10, Mac OS X Tiger. Se soportan plataformas tanto de 32 bits como de 64, como la AMD-64, Itanium y PowerPC. InstallShield continúa soportando casi cualquier otra plataforma, como AIX, HP-UX, NetWare, IRIX, Tru64, FreeBSD y z/OS.[6]
- Permite la instalación inteligente de los componentes del software y que hace la colaboración entre distintos equipos de desarrollo.[6]

- Posibilita a los desarrolladores acelerar el desarrollo, reducir los costos de instalación y aumentar la satisfacción del cliente al ofrecer una experiencia constante de instalación en cualquier plataforma.[6]

- **Soporte Java J2SE 5.0:** InstallShield puede instalar automáticamente el J2SE 5.0 Java Virtual Machine para aplicaciones de clientes y de servidor[6]

- **Opción de creación de Windows Installer – *MSI* –:** Genera paquetes Windows Installer – *MSI* – con instaladores Linux, Solaris y Mac OS X.[6]

- **Componentes compartidos:** Se pueden compartir los componentes entre aplicaciones distintas, asegurando la consistencia y reduciendo el costo.[6]

- **Dependencia de productos cruzados:** Los instaladores pueden comprobar los requisitos previos y las dependencias de otras instalaciones y configuraciones de productos.[6]

- **Grupos de acción:** Las acciones de instalación pueden agruparse de forma lógica, permitiendo que varias acciones compartan reglas y se simplifique la creación de soluciones complejas de instalación.[6]

1.2.3. InstallBuilder.

Soporte multiplataforma: los instaladores BitRock son binarios que funcionan en: Microsoft Windows 98, ME, 2000, XP, 2003, 2008, Vista, Mac OS X, FreeBSD, OpenBSD, Solaris –*Intel & Sparc*–, AIX, HP-UX, IRIX, and Linux (Intel x86\x64, Itanium, s390 & PPC).[7]

Características de InstallBuilder:

- **Integración RPM:** Los instaladores BitRock pueden registrar el software con una base de datos de paquetes RPM, combinando la facilidad de uso con la potencia de los sistemas de paquetes RPM.[7]

- **Soporte para la compilación multiplataforma:** La herramienta de creación del instalador se puede ejecutar en Microsoft Windows, Mac OS X, Solaris, HP-UX, AIX,

FreeBSD, OpenBSD, IRIX y GNU\Linux (x86\x64 Intel, Itanium, s390, PPC) y generar instaladores para todas las plataformas de destino en un único archivo de proyecto.[7]

- **Facilidad de uso:** InstallBuilder es fácil de aprender, de usar en un entorno gráfico de desarrollo; se puede diseñar, construir y probar instaladores con el clic de un botón.[7]
- **Ahorro de tiempo:** Para usuarios avanzados, un proyecto amistoso en formato XML apoya la integración de los controles y el código, el desarrollo de la colaboración y personalización de proyectos, tanto manual como usando scripts externos. Una interfaz de línea de comandos le permite automatizar e integrar el proceso de construcción.[7]
- **Instaladores optimizados:** Los instaladores BitRock están optimizados en tamaño y velocidad y no requieren de un proceso de extracción. Esto reduce el inicio de descarga, y tiempo de instalación. La compatibilidad integrada con LZMA proporciona *ratios* de gran compresión para reducir aún más el tamaño de los instaladores.[7]
- **No hay dependencias externas:** Los instaladores BitRock multiplataforma son de un solo archivo, en sí mismos, ejecutables nativos sin dependencias externas y una sobrecarga mínima. A diferencia de productos de la competencia, todos los instaladores BitRock son verdaderamente código nativo y no requieren de un paquete de Java Runtime Environment.[7]
- **Múltiples modos de instalación:** Los instaladores BitRock proporcionan varios modos de interfaz gráfica de usuario con vistas nativas para la instalación en una variedad de entornos de escritorio, un modo de instalación basado en texto para las instalaciones basadas en la consola y el mando a distancia, y un modo de instalación desatendida que puede ser utilizado para la integración en scripts de Shell para el despliegue automatizado.[7]
- **Lenguaje y Plataforma Independiente:** Los instaladores BitRock pueden instalar las aplicaciones escritas en cualquier idioma, incluyendo: Java, PHP, Perl, Python, Ruby, C \ C + + y .NET Mono.[7]

- **Integración de escritorio:** Los instaladores BitRock proporcionan un aspecto nativo y la integración de escritorio para Windows, KDE y Gnome.[7]
- **Generación de RPM y DEB:** Además de crear ejecutables nativos que pueden registrarse en el subsistema de RPM, InstallBuilder puede generar paquetes RPM y Debian, que pueden ser instalados utilizando las herramientas nativas de gestión de paquetes.[7]
- **Opciones avanzadas de configuración:** Pide al usuario múltiples entradas en una sola pantalla para agilizar el proceso de instalación.[7]
- **Función de desinstalación:** Un programa de desinstalación se crea como parte de cada instalación, permitiendo a los usuarios desinstalar fácilmente el software. Al igual que con el instalador, se puede ejecutar en una variedad de modos. En Microsoft Windows, la función de desinstalación también se puede acceder desde la opción agregar o quitar tus programas en el Panel de control.[7]
- **Soporte de múltiples idiomas:** Los instaladores BitRock apoyan una variedad de idiomas de instalación, incluyendo Inglés, Alemán, Japonés, Español, Italiano, Francés, Portugués, Chino Tradicional, Holandés, Polaco, Valenciano, Catalán, Estonio, Esloveno, Rumano, Húngaro, Ruso y Gales. Se puede especificar un idioma predeterminado o dejar que el usuario decida.[7]

1.2.4. InstallJammer.

InstallJammer es el instalador con interfaz de usuario multiplataforma que está diseñado para funcionar completamente como multiplataforma. Tiene el constructor de instalación muy poderoso que admite temas múltiples y el nivel alto de configuración para los instaladores.[8]

Los instaladores son construidos como un simple fichero ejecutable para una fácil distribución sobre la web, instalando todo lo que se necesita para la aplicación en cualquier plataforma que se ejecute.[8]

Características de InstallJammer:

Desarrollo rápido: InstallJammer incluye un poderoso constructor de instaladores para el propósito del usuario, el cual permite que este tome el control de todo lo que acontece al programa a crear, se puede obtener un instalador corriendo en pocos minutos.[8]

Personalización: InstallJammer permitirá configurar del instalador hasta el más mínimo detalle: tanto en su interfaz y/o apariencia como en sus funcionalidades.[8]

Soporte multiplataforma: los instaladores de InstallJammer son binarios nativos que pueden ser ejecutados en: Windows 98, ME, 2000, XP, 2003, Vista, 2008, Linux, FreeBSD, Solaris, HP-UX y AIX mientras que añadir más plataformas es posible si es que el equipo de trabajo de InstallJammer aún no la ha desarrollado; también es posible obtener el código fuente del mismo y construir la plataforma.[8]

Soporte multilingüe: InstallJammer soporta múltiples lenguajes en los instaladores con facilidad. Existen muchas traducciones y otras son añadidas constantemente en cada liberación del programa.[8]

Instaladores rápidos: los instaladores de InstallJammer son construidos para ser rápidos y ligeros. Su arranque suele ser muy rápido debido a que estos no requieren espacio adicional en disco para ficheros temporales antes de iniciar.[8]

Instaladores ligeros: los instaladores de InstallJammer son construidos para ser ligeros – *dígase en cuestiones de espacio: pequeños* – sin dependencias externas. No es necesario arrastrar con toda la máquina virtual de Java para instalar el software. Es posible distribuir el software con la confianza de que ha de correr nativamente en cada una de las plataformas para las que ha sido generado sin problemas.[8]

Acciones configurables: Entrelazar en los registros de Windows la aplicación desplegada por el instalador o instalarla desde un fichero "*rpm*" en los sistemas Linux, hacer accesos directos de todas formas, ubicar la Máquina Virtual de Java, todo desde las acciones que incluye.[8]

Simples archivos de proyecto: los ficheros de un proyecto InstallJammer son muy simples, texto plano que puede ser perfectamente salvado en un control de versiones y rápidamente comparados para visualizar diferencias cuando se trabaja en colaboración con otros desarrolladores.[8]

Opciones para múltiples interfaces de ususario: InstallJammer puede ejecutarse de forma estándar, por defecto, modo consola o instalación silenciosa dependiendo de las opciones/necesidades del usuario. Por defecto acepta todas las configuraciones diseñadas con este fin mientras ejecuta una instalación con GUI[8], en modo silencioso no utiliza GUI ni interacción con el usuario. En la consola o terminal permite la instalación por parte de un usuario de un sistema tipo UNIX. Lo más importante de todo es que automáticamente selecciona la mejor opción según sea el caso.[8]

Edición de los diálogos: InstallJammer permite configurar cada panel de la instalación; puede, justamente, crearse un instalador que se comporte y se vea como el usuario desee/necesite.[8]

Control por línea de comandos: los proyectos de InstallJammer pueden ser construidos y probados desde la línea de comandos, ofreciendo una fácil integración con el entorno de desarrollo y sincronizar las configuraciones de la compilación. Cualquier plataforma soportada puede ser construido, ejecutando el InstallJammer en otra plataforma.[8]

Control de temas: InstallJammer permite la selección de su apariencia desde diferentes temas para los futuros instaladores solo con la inclusión de los temas comerciales más populares. Siempre es posible la creación de temas nuevos para propósitos generales.[8]

Apariencia nativa: los instaladores de InstallJammer proveen una apariencia nativa para todas las plataformas soportadas.[8]

Posibilidades de desinstalación: InstallJammer creará automáticamente un desinstalador durante la instalación si se necesita.[8]

[8] **GUI:** Interfaz Gráfica de Usuario del inglés Graphical User Interface.

Libre de usar y distribuir: es esta quizás la mejor característica: todo es libre. Es de código abierto y 100 % libre de usar y distribuir. InstallJammer no requiere de pago monetario o de la recepción de regalías por distribuir el software creado con él o propiamente su uso. Tampoco hace implicación si la distribución es de código abierto, libre, comercial o de cualquier tipo. El usuario es libre de distribuir su software como estime conveniente. InstallJammer permite hacer instaladores propiamente comerciales.[8]

El software InstallJammer se destaca por la integración con las plataformas sobre las que opera, tiene soporte para la compilación multiplataforma, además se sobresale por la facilidad de uso, el ahorro de tiempo y los instaladores optimizados, no posee dependencias externas, cuenta con múltiples modos de instalación, brinda lenguaje y plataforma independiente, integración de escritorio, posibilita opciones avanzadas de configuración, función de desinstalación y soporte de múltiples idiomas. InstallJammer se caracterizó también por ser utilizado para la construcción del instalador del ECM Alfresco.

El languaje de programación a utilizar para la implementación del software "*Instalador Multiplataforma para el Gestor de Documentos Administrativos eXcriba*" está doblegado por el sistema diseñador de instaladores escogido, este ofrece todas las características necesarias, aportando además las herramientas para lograr este propósito: **Tcl/Tk**. TCL es un lenguaje de script creado por John Ousterhout; se combina con el lenguaje Tk para la creación de interfaces gráficas.[9] El InstallJammer será usado también como **entorno de desarrollo** – *adicionalmente puede usarse algún editor de texto avanzado, o un IDE que soporte la sintaxis del mencionado lenguaje* –, sobre éste se trabajarán los *scripts* para dotar al instalador de todas las acciones necesarias para cumplimentar el propósito sobre la plataforma en la que se despliegue.

1.3. Metodologías de Desarrollo de Software.

Las metodologías de desarrollo de software surgieron por la necesidad de la *industria del software* de agilizarse y robustecerse al mismo tiempo. La evolución de estas siempre ha estado indisolublemente ligada al crecimiento del software en sí mismo, aportándole a este último diversas herramientas, las cuales permiten agilizar la documentación, otras el proceso de *producción* en sí mismo y otras el desarrollo de todo el ciclo del desarrollo del software. A lo largo de los años han existido una gran variedad de estas, de las cuales se exponen solo las destacadas para su valoración.

1.3.1. Metodología de Desarrollo de Sistemas Dinámicos.

El DSDM[9] desarrollado en UK[10] desde 1995 es un método que provee un framework para el desarrollo ágil de software, apoyado por su continua implicación del usuario en un desarrollo iterativo y creciente que sea sensible a los requerimientos cambiantes, para desarrollar un sistema que reúna las necesidades de la empresa en tiempo y presupuesto.

1.3.2. SCRUM

Una de las características más importantes es que es muy fácil de explicar y de entender, lo que ayuda mucho a su implantación.

SCRUM puede ser aplicado a distintos modelos de calidad – *como podría ser CMMI* – pues estos exponen lo que debe hacerse, pero no dicen cómo. Ahí es donde entra SCRUM como modelo de gestión del proyecto.[10]

[9] **DSDM:** Método de Desarrollo de Sistemas Dinámicos del inglés Dynamic Systems Development Method.
[10] **UK:** Reino Unido del inglés United Kingdom.

Principales características:

- Se obtiene software lo más rápido posible y este cumple con los requerimientos más importantes.[11]
- Se trabaja en iteraciones cortas, de alto enfoque y total transparencia.[11]
- Se acepta que el cambio es una constante universal y se adapta el desarrollo para integrar los cambios que son importantes.[11]
- Se incentiva la creatividad de los desarrolladores haciendo que el equipo sea auto administrado.[11]
- Se mantiene la efectividad del equipo habilitando y protegiendo un entorno libre de interrupciones e interferencias.[11]
- Permite producir software de una forma consistente, sostenida y competitiva.[11]
- Las reuniones se dedican a inconvenientes recientes, evitando el estancamiento.[11]

SCRUM es una forma de gestionar proyectos de software. No es una metodología de análisis, ni de diseño.[10] Requiere delegar responsabilidades al equipo, incluso permite fallar si es necesario.[11]

1.3.3. RUP

El RUP[11] se está implementando desde 1999, constituye la metodología tradicional estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas. Se caracteriza por ser iterativa e incremental, centrada en la arquitectura y guiada por los casos de uso. Divide el proceso de desarrollo en ciclos de vida, obteniendo un producto al final de cada ciclo, los cuales se dividen en fases que deben de terminar con un hito, dentro de estas fases se encuentran: Inicio, Elaboración, Construcción y Transición.[12]

[11] **RUP:** Rational Unified Process.

Es una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. Unifica los mejores elementos de metodologías anteriores, está preparada para desarrollar grandes y complejos proyectos y utiliza el UML[12] como lenguaje de representación visual para preparar todos los esquemas de un sistema software. De forma general proporciona una guía para ordenar las actividades del equipo de trabajo, dirige las áreas de cada desarrollador por separado y del equipo como un todo, especifica los artefactos que deben desarrollarse y ofrece criterios para el control, la medición de los productos y actividades de proyectos.[12]

Estudiadas y expuestas las metodologías de desarrollo de software, se ha inclinado la balanza por RUP, es la metodología tradicional más ampliamente documentada, utilizada y apoyada por distintos equipos de trabajo. La forma en que los ciclos se dividen en fases y estos terminan con un hito es aplicable al actual modelo de desarrollo empleado por el equipo de trabajo del GDA eXcriba; fue electo además por su integración intrínseca con el lenguaje UML.

1.4. Ingeniería de Software Asistida por Computadora.

La Ingeniería de Software Asistida por Computadora, conocidas sus herramientas como CASE, permite diseñar completamente el software de forma que provea ahorro tiempo y recursos humanos en lugar de proceder a implementar el software directamente; utilizando esta ingeniería se evita, *inclusive*, realizar regresiones propias por concepto de diseño, de estas herramientas se presentan las de mayor impacto en el medio informático por su calidad y utilidad.

[12] **UML:** Unified Modeling Language

1.4.1. Power Designer

Brinda potentes técnicas de análisis, diseño y gestión de metadatos a la empresa. Combina varias técnicas estándares de modelamiento con herramientas líder de desarrollo, como .NET, Sybase WorkSpace, Sybase Powerbuilder, Java y Eclipse, para darle a las empresas soluciones de análisis de negocio y de diseño formal de base de datos. Además trabaja con más de 60 bases de datos relacionales.

1.4.2. Rational Systems Developer

Software de IBM, es una herramienta de diseño y desarrollo que saca provecho de todas las posibilidades de Eclipse e incluye plugins que permiten el desarrollo dirigido por modelos y una arquitectura de software para la creación de aplicaciones con arquitecturas sólidas basadas en Corba, C\C++ y Java J2SE, mediante UML.

1.4.3. Visual Paradigm

Visual Paradigm para UML, herramienta CASE profesional, soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad a menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y también la documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

La herramienta CASE a utilizar será el Visual Paradigm, el cual posee características que lo hacen destacar sobre sus competidores, el soporte para la creación de proyectos con trabajos realmente grandes, además de su potente integración con el lenguaje de modelado UML, el cual será utilizado en el proceso de modelación, análisis y diseño del software – *instalador* –.

Presentada la fundamentación teórica, quien guiará el proceso de desarrollo del instalador multiplataforma del GDA eXcriba, se ofreció información relevante de los procedimientos de instalación de programas y de los sistemas diseñadores de instaladores multiplataforma. Se seleccionaron las herramientas y tecnologías a utilizar, la metodología de desarrollo de software y el lenguaje para la implementación, dando paso a evaluar las características del sistema.

Capítulo 2

Características del Sistema.

En el presente capítulo se describe la solución propuesta estableciendo el funcionamiento del sistema; para ello, se explicará la dinámica del proyecto a través de sus diferentes etapas y procesos, haciendo uso de la metodología RUP, además se ofrece una valoración del diseño propuesto para los cambios necesarios en la transición del diseño a la implementación, teniendo en cuenta los requerimientos funcionales que especifican las condiciones que el sistema debe cumplir, representadas a través de casos de uso del sistema y los no funcionales que describen las características y restricciones que deberá tener el sistema para su correcto funcionamiento.

2.1. Descripción del problema.

En la actualidad la instalación del GDA eXcriba se realiza de forma manual, al igual que sus dependencias y configuraciones, por lo que este proceso es engorroso y se encuentra en desventaja para competir en el mercado; careciendo de usabilidad y dependiendo de personal capacitado para efectuar la instalación, con conocimientos previos del sistema operativo en que se realizará el despliegue, contando con el uso de la línea de comandos. Esto puede acarrear errores comunes como una configuración errónea, dejando inactivo o inservible un módulo del GDA eXcriba, causando pérdida de recursos y de tiempo, viéndose afectado con la ausencia de la automatización del sistema, ya que para realizar esta actividad este será relativo en dependencia de las destreza de quien lo ejecute. El cliente que desea desplegar el sistema depende directamente de la asistencia del equipo de desarrollo, lo que puede causar rechazo.

2.2. Objeto de automatización.

Para el proceso de instalación del GDA eXcriba será implementada una solución automatizada, estará basada en la construcción de un instalador multiplataforma expandiendo la cobertura de despliegue. Se cubrirán elementos que eran muy engorrosos de la forma tradicional o inexistentes, como son los de actualización de los ficheros de configuración del mismo, en el cual hay que descubrir manualmente los programas anexos, identificarlos y poner las direcciones físicas en los mencionados ficheros; en el proceso de resolución de dependencias se precisa conocer, obtener e instalar todas estas, de forma autónoma. Como agregado de esta automatización se obtendrá un instalador con interfaz gráfica. El instalador, en su concepción ha de ser, como el del ECM Alfresco, aportando la granularidad de este al del GDA eXcriba; además será simple, sencillo, robusto y resolverá la instalación de los programas de terceros que requiere el GDA eXcriba para su funcionamiento.

2.3. Propuesta del sistema.

El sistema propuesto no tiene antecedentes dentro del proyecto GDA eXcriba. Se hará uso del sistema diseñador de instaladores InstallJamme para crear una aplicación de escritorio. Con este sistema será posible el despliegue del GDA eXcriba en varias plataformas entre las que se incluyen las más utilizadas actualmente y aquellas de mayor prioridad para el proyecto como son: Microsoft Windows XP, Microsoft Windows 7, Canonical Limited Ubuntu y Debian GNU\Linux. El sistema proporcionará una interfaz sencilla facilitando el proceso de instalación para el usuario. Básicamente este instalador ha de permitir desplegar el GDA eXcriba en las plataformas mencionadas con una interfaz de usuario agradable, sencilla y sobre todo lo más intuitiva posible. El sistema diseñador de instaladores InstallJammer destaca por encima de los restantes anteriormente analizados en la facilidad de uso, integración con las plataformas soportadas, la utilización eficiente de los recursos del computador, existencia de abundante documentación; se estudiaron las características sobre la portabilidad, fiabilidad, compresión y la inclusión de una licencia en la herramienta de código abierto y la posibilidad de la comercialización del producto restante que es interés del equipo de desarrollo de GDA eXcriba.

Dentro del ciclo de vida de RUP se incluye la especificación de requisitos, los casos de uso del sistema, además de el modelo de la arquitectura. Todos estos *artefactos* constituyen una representación del producto que es necesaria para que los desarrolladores puedan implementar. Forman parte de la representación del software, la arquitectura y el modelo de dominio que describen el contexto del dominio en el que se encuentra el sistema. Todos estos modelos están relacionados representando de conjunto al sistema como un todo.

Dentro de las actividades más importantes definidas en la metodología RUP está la definición del modelo del negocio, en el que se hace una descripción detallada del negocio; pero si no está bien definido entre los clientes y los ejecutores del proyecto, entonces es factible el llamado *Modelo de Dominio*. Para este sistema no están bien especificados los pasos a seguir, los actores del negocio, ni definidas las actividades necesarias que intervienen en él, por lo que la modelación del sistema propuesto se orientará en la definición del **Modelo de Dominio**.

Modelo de dominio. Es una abstracción de un sistema cerrado semánticamente.[12] Construido con las reglas de UML durante la fase de Inicio; en la construcción del modelo de dominio se presenta como uno o más diagramas de clases que contiene, no conceptos propios de un sistema de software sino de la propia realidad física.

Para el desarrollo de un instalador multiplataforma para el GDA eXcriba se propone una interfaz de usuario que le proporcione al actor acciones necesarias para desplegar el sistema, además de tener mensajes de información para definir los pasos que se deben seguir. El sistema presentará opciones, como escoger el tipo de idioma para de esta forma universalizar el sistema, el tipo de instalación para dar la posibilidad de personalizarla, además la ruta donde quedará plasmada la instalación y logrando resolver este paso, le dará la oportunidad de configurar el gestor de base de datos y en el paso posterior configurar el módulo de eXcribaWeb, permitiendo nutrir al sistema de herramientas para la administaración/utilización. Una vez finalizada esta secuencia de pasos, el instalador procederá a desplegar y configurar los módulos escogidos, obteniendo como resultado final el GDA eXcriba instalado.

2.4. Modelo de dominio.

Modelo de dominio.

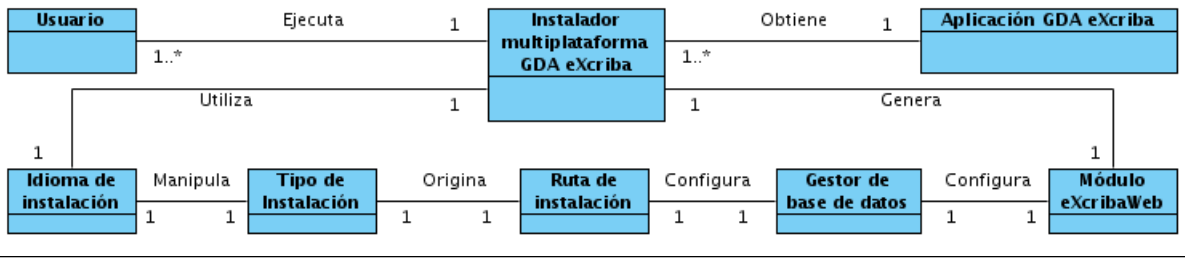


Tabla 2.1: Modelo de dominio.

2.5. Especificación de requisitos.

Para el Análisis y Diseño de un sistema se toma como punto de partida la captura de requisitos, la definición de estos es un paso muy importante para el desarrollo de las siguientes etapas, pues un error en esta fase inicial puede traer consigo la implementación de un sistema que no cumpla las expectativas y en el peor de los casos no aporte valor agregado al negocio para el que debe ser concebido.

2.5.1. Requisitos Funcionales.

1. **RF - Instalar GDA eXcriba:** Especifica la implementación de un instalador multiplataforma para GDA eXcriba.

   1.1. **RF - Seleccionar el idioma de la instalación:** El sistema mostrará un componente de tipo *combo –una lista desplegable–* donde desplegará la opción de idioma que el usuario deberá escoger *–inglés o español–* para realizar la instalación.

   1.2. **RF - Seleccionar el Tipo de Instalación:** El sistema brindará dos opciones de instalación:

   - **Típica:** donde se ejecutará una instalación preestablecida.
   - **Personalizada:** el usuario podrá seleccionar qué módulos o componentes instalará.

1.2.1 RF - Seleccionar Módulo: El sistema mostrará una lista con las opciones previamente seleccionadas que determinan como se establecerá la instalación del sistema, tales como:

- **eXcribaCore:** es un servidor ECM Alfresco v3.0 personalizado por el equipo de desarrollo del GDA eXcriba. Ofrece una serie de servicios que son consumidos esencialmente por el módulo eXcribaWeb.
- **eXcribaWeb:** la función principal de este módulo es consumir los servicios que provee el eXcribaCore, nutre al GDA eXcriba de herramientas para su administración/utilización en la web.

1.3. RF- Seleccionar la Ruta de instalación: El sistema dará la posibilidad de especificar la ruta donde quedará plasmada la instalación, con la ayuda de un formulario.

1.4 RF - Configurar las opciones de la base de datos: El sistema le permitirá al usuario configurar la base de datos del módulo eXcribaCore, llenando un formulario con los siguientes datos:

- **Usuario:** usuario para la autenticación.
- **Contraseña:** clave del usuario.
- **Repetir contraseña:** clave del usuario.
- **Servidor:** dirección del servidor de base de datos.
- **Puerto:** puerto de acceso al servidor de base de datos.
- **Nombre:** base de datos a utilizar.

1.5 RF - Configurar las opciones del módulo eXcribaWeb: El sistema le permitirá al usuario configurar el módulo eXcribaWeb, indicando qué servidor eXcribaCore usará, llenando un formulario con los siguientes datos:

- **Servidor:** servidor eXcriba.
- **Puerto:** puerto de acceso al servicio brindado por el núcleo del sistema.

2.5.2. Requisitos No Funcionales.

1. **RNF - Usabilidad:** Debe poseer una funcionalidad adecuada, o sea, satisfacer los requisitos funcionales declarados, de modo que el esfuerzo para usarlo sea mínimo, además debe ser atractivo, útil e intuitivo para los usuarios. La mantenibilidad debe ser alta, de modo que pueda adaptarse a condiciones cambiantes del entorno en que se ejecute y las modificaciones puedan hacerse fácilmente.

2. **RNF - Portabilidad:** El sistema será independiente y multiplataforma, lo que significa que podrá ser usado sobre cualquier plataforma de interés del proyecto GDA eXcriba: Microsoft Windows XP, Microsoft Windows 7, Canonical Limited Ubuntu y Debian GNU\Linux.

3. **RNF - Soporte:** Se debe de generar un documento detallado o un manual de usuario que explique el funcionamiento del sistema a través de imágenes para aquellos usuarios que no tienen dominio sobre el tema, mostrándole paso a paso las decisiones que se pueden tomar y las respuestas del sistema. Para satisfacer este requisito el sistema deberá contar con un **manual de usuario**.

4. **RNF - Software:** Para el funcionamiento del sistema deberá existir un gestor de base de datos, bibliotecas de ejecución de la plataforma Java y la suite ofimática OpenOffice para el módulo eXcribaCore, para el eXcribaWeb es imprescindible un servidor web Apache con los módulos correspondientes para interpretar lenguaje php5.

5. **RNF - Hardware:** El sistema necesitará 1 GB de memoria RAM y 80 GB de disco duro.

6. **RNF - Restricciones de diseño e implementación:**

   - **Lenguaje de programación:** El lenguaje de programación que deberá ser usado para la implementación de la aplicación será Tcl/Tk; ha de utilizarse para la realización de *scripts* específicos por plataforma.

2.6. Definiciones de Actores y Casos de Uso.

2.6.1. Definición de Actores del Sistema.

Actores:	Justificación:
Usuario	Es la persona que ejecutará el instalador multiplataforma del GDA eXcriba.

Tabla 2.2: Definición de Actores del Sistema.

2.6.2. Definición de Casos de Uso.

CU - 1	Instalar GDA eXcriba
Actor:	Usuario.
Descripción:	Permite al usuario realizar la instalación de la aplicación GDA eXcriba.
Referencia:	RF - 1, RF - 1.1, RF - 1.2, RF - 1.2.1, RF - 1.3, RF - 1.4, RF - 1.5

Tabla 2.3: Definición de Casos de Uso.

2.6.3. Diagrama de Casos de Uso del Sistema.

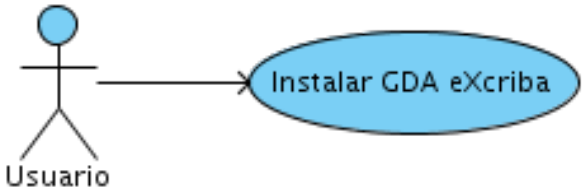
Diagrama de Casos de Uso del Sistema.


Tabla 2.4: Diagrama de Casos de Uso del Sistema.

2.7. Descripción del Caso de Uso del Sistema.

Caso de uso:	**Instalar GDA eXcriba.**
Actor:	Usuario
Resumen:	El caso de uso comienza cuando el actor decide instalar la aplicación GDA eXcriba con el instalador multiplataforma. Terminando el caso de uso una vez instalado el mismo.
Precondiciones:	Debe tener 500 mb espacio mínimo disponible para guardar los archivos de instalación.
Referencias:	RF - 1, RF - 1.1, RF - 1.2, RF - 1.2.1, RF - 1.3, RF - 1.4, RF - 1.5.
Prioridad:	Crítica.
Flujo Normal de Eventos	
Sección 1 "Instalar GDA eXcriba"	
Acción del Actor:	**Respuesta del Sistema:**
1. El usuario ejecuta el instalador de GDA eXcriba.	2. El sistema muestra una interfaz con la opción de seleccionar el idioma con el que desea realizar la instalación, con el botón "Ok" para confirmar la selección y el "Cancelar" para terminar la instalación.
3. El usuario selecciona el idioma y presiona el botón "Ok".	4. El sistema muestra una interfaz para confirmar que el GDA eXcriba se va a instalar en su equipo; para ello brinda las opciones "Si" o "No".
5. El usuario selecciona el botón "Si".	6. El sistema muestra una interfaz con un mensaje de bienvenida y las opciones de "Siguiente" o "Cancelar" la instalación.
Continúa en la próxima página...	

7. El usuario presiona el botón “Siguiente”.	8. El sistema muestra una interfaz dando la opción de seleccionar el tipo de instalación que puede ser “Típica” o “Personalizada” y brinda la opción de ir “Atrás” , “Cancelar” o “Siguiente”.
9. El usuario selecciona la opción “Típica” y presiona el botón “Siguiente”.	10. El sistema muestra una interfaz para seleccionar la ruta de instalación del “eXcribaCore” y brinda la opción de ir “Atrás”, “Cancelar” o “Siguiente”.
11. El usuario especifica la ruta y presiona el botón “Siguiente”.	12. El sistema muestra una interfaz para configurar la base de datos mediante un formulario con los siguientes campos: ▪ Nombre de la base de datos ▪ Servidor de base de datos ▪ Puerto de acceso al servidor de base de datos ▪ Usuario para conectarse al servidor de base de datos ▪ Contraseña del ususario ▪ Confirmación de la contraseña del ususario Además muestra los botones “Atrás”, “Siguiente” y “Cancelar”
13. El usuario inserta los datos para la configuración de la base datos y presiona el botón “Siguiente”.	14. El sistema valida que los campos obligatorios estén correctos.
Continúa en la próxima página...	

	15. En caso de que los campos obligatorios estén introducidos correctamente, el sistema mostrará una interfaz con un reporte de la instalación.
16. El usuario presiona el botón “Siguiente”.	17. El sistema muestra una interfaz para visualizar el progreso de la instalación y brinda la opción de “Cancelar”.
	18. El sistema muestra una interfaz especificándole al usuario que se ha realizado la instalación de GDA eXcriba dándole como única opción la de “Finalizar” para salir del programa.
19. El usuario presiona el botón “Finalizar” culminado la instalación.	
Flujos Alternos:	
Acción del Actor:	**Respuesta del Sistema:**
3.1. El usuario selecciona el botón “Cancelar”.	3.2 El sistema culmina la instalación.
5.1 El usuario selecciona el botón “No”.	5.2 El sistema cancela la instalación.
7.1 El usuario selecciona el botón “Cancelar” **ver Sección 4 “Cancelar Instalación”**.	
9.a.1 El usuario selecciona el botón “Cancelar” **ver Sección 4 “Cancelar Instalación”**.	
9.b.1. El usuario selecciona el botón “Atrás”.	9.b.2 El sistema muestra el flujo de eventos correspondiente al paso 6.
Continúa en la próxima página...	

9.c.1 El usuario selecciona la opción “Personalizada” y presiona el botón “Siguiente” **ver Sección 2 “Tipo de Instalación Personalizada”**.	
11.a.1 El usuario selecciona el botón “Cancelar” **ver Sección 4 “Cancelar Instalación”**.	
11.b.1. El usuario selecciona el botón “Atrás”.	11.b.2 El sistema muestra el flujo de eventos correspondiente al paso 8.
13.a.1 El usuario selecciona el botón “Cancelar” **ver Sección 4 “Cancelar Instalación”**.	
13.b.1 El usuario selecciona el botón “Atrás”.	13.b.2 El sistema muestra el flujo de eventos correspondiente al paso 10.
	14.1 Es sistema muestra un mensaje de error en dependencia del campo inválido.
16.a.1 El usuario selecciona el botón “Cancelar” **ver Sección 4 “Cancelar Instalación”**.	
16.b.1 El usuario selecciona el botón “Atrás”.	16.b.2 El sistema muestra el flujo de eventos correspondiente al paso 12.
17.1 El usuario selecciona el botón “Cancelar” **ver Sección 4 “Cancelar Instalación”**.	
Sección 2 “Tipo de Instalación Personalizada”	
Acción del Actor:	**Respuesta del Sistema:**
Continúa en la próxima página...	

1. El usuario selecciona “Personalizada” y presiona el botón “Siguiente”	2. El sistema muestra una interfaz con los dos módulos “eXcribaCore” y “eXcribaWeb” que se pueden instalar y con sus dependencias previamente seleccionadas, además de los botones “Atrás”, “Siguiente” y “Cancelar”.
3. El usuario desmarca el módulo “eXcribaWeb” y presiona el botón “Siguiente”.	4. El sistema verifica que las dependencias para este módulo estén correctas.
	5. En caso que las dependencias estén seleccionada para su instalación, el sistema muestra la interfaz correspondiente al paso 10 del flujo normal de eventos.
Flujos Alternos:	
Acción del Actor:	**Respuesta del Sistema:**
1.a.1 El usuario selecciona el botón “Cancelar” **ver Sección 4 “Cancelar Instalación”**.	
1.b.1 El usuario selecciona el botón “Atrás”.	1.b.2 El sistema muestra el flujo de eventos correspondiente al paso 6.
3.a.1. El usuario desmarca el módulo “eXcribaCore” y presiona el botón “Siguiente” **ver Sección 3 “Módulo eXcribaWeb”**.	
3.b.1 El usuario selecciona el botón “Cancelar” **ver Sección 4 “Cancelar Instalación”**.	
3.c.1 El usuario selecciona el botón “Atrás”.	3.c.2 El sistema muestra el flujo de eventos correspondiente al paso 8.
Continúa en la próxima página...	

	4.1 En caso de que las dependencias no estén seleccionada para su instalación remitirce al paso 12 del flujo normal de eventos.
	4.2 Luego de ejecutarce al paso 12 del flujo normal de eventos, el sistema muestra una interfaz intercediendo por la dependencia que no fue seleccionada, dando la posisbilidad de propiciar la busqueda de la dependencia a través de una ruta.
Sección 3 "Módulo eXcribaWeb"	
Acción del Actor:	**Respuesta del Sistema:**
1. El usuario desmarca el módulo "eXcribaCore" y presiona el botón "Siguiente"	2. El sistema verifica que las dependencias para este módulo estén correctas.
	3. El sistema muestra uns interfaz para configurar el módulo "eXcribaWeb" mediente un formulario con los siguientes campos:. ▪ Servidor ▪ Puerto Además muestra los botones "Atrás", "Siguiente" y "Cancelar"
4. El ususrio inserta los datos para la configuración del módulo "eXcribaWeb" y presiona el botón "Siguiente"	5. El sistema valida que los campos obligatorios estén correctos.
Continúa en la próxima página...	

	6. En caso de que los campos obligatorios estén correctos, remitirce al paso 15 del flujo normal de eventos.
Flujos Alternos:	
Acción del Actor:	**Respuesta del Sistema:**
1.a.1. El usuario selecciona el botón "Cancelar" **ver Sección 4 "Cancelar Instalación"**.	
1.b.1 El usuario selecciona el botón "Atrás".	1.b.2 El sistema muestra el flujo de eventos correspondiente al paso 8.
	2.1 En caso de que las dependencias no estén seleccionada para su instalación remitirce al paso 12 del flujo normal de eventos.
	2.2 Luego de ejecutarce al paso 12 del flujo normal de eventos, el sistema muestra una interfaz intercediendo por la dependencia que no fue seleccionada, dando la posisbilidad de propiciar la busqueda de la dependencia a través de una ruta.
4.a.1 El usuario selecciona el botón "Cancelar" **ver Sección 4 "Cancelar Instalación"**.	
	5.1 El sistema muestra un mensaje de error en dependencia del campo inválido.
Sección 4 "Cancelar Instalación"	
Acción del Actor:	**Respuesta del Sistema:**
Continúa en la próxima página...	

	2. El sistema muestra una interfaz para confirmar la acción dándole las opciones de “Si” o ”No”.
3. El usuario presiona el botón “Si”.	4. El sistema cancela el proceso de instalación.
Flujos Alternos:	
Acción del Actor:	**Respuesta del Sistema:**
3.1 El usuario presiona el botón “No”.	3.1.1 El sistema no cancela y continúa con el proceso de instalación.
Prototipo de Interfaz:	
Poscondiciones:	Queda instalado el GDA eXcriba.

Tabla 2.5: Descripción textual del Caso de Uso: Instalar GDA eXcriba.

En este capítulo se expone la estructura del sistema a través del lenguaje UML que es un estándar *diseñado para visualizar, especificar, construir y documentar software*[13]. Por medio de este lenguaje se mostraron las principales clases del dominio para una mayor comprensión de este, además de analizar las características y funciones del sistema para la creación de un instalador multiplataforma para el GDA eXcriba, las que se representaron mediante un diagrama de casos de uso del sistema, realizando además la descripción detallada del caso de uso.

Se encuentran sentadas las bases para el próximo Flujo de Trabajo –*Análisis y Diseño según RUP*– con un desarrollo intensivo en el modelado del negocio y en los requisitos del sistema. Es posible comenzar a realizar el Flujo de Trabajo de Análisis y Diseño de la aplicación teniendo en cuenta los requerimientos especificados.

Capítulo 3

Análisis y Diseño del Sistema.

Este capítulo está enfocado al análisis y diseño como otra fase dentro de los flujos de trabajo y modelos del RUP. En el flujo de trabajo anterior se capturaron los requisitos en forma de casos de uso en el modelo de casos de uso del sistema, para dar continuidad se analizará y diseñará el sistema cumpliendo con estos requisitos, creando en primer lugar los diagramas de clases del análisis, luego los de interacción y por último los diagramas de clases del diseño para que en el próximo capítulo se formule la implementación del sistema.

Un sistema, tanto del mundo real como en el mundo del software, es bastante complejo, por ello es necesario dividir el sistema en partes o fragmentos si se quiere entender y administrar su complejidad. Estas partes se pueden representar como modelos que describan sus aspectos esenciales. Por tanto un paso útil en la construcción de un sistema de software es el de crear modelos que organicen y comuniquen los detalles más importantes de la vida real con que se relacionan. Los modelos se componen de artefactos, diagramas y documentos que describen cosas.

3.1. Diagramas de Clases del Análisis.

El diagrama de clases del análisis está conformado por la interfaz que se le mostrará al usuario y que a la vez estará relacionada con las clases controladoras, las que le brindarán las funcionalidades para que el sistema funcione correctamente. Estas están relacionadas con la entidad módulo para ser utilizados por el instalador una vez que se necesite.

Diagrama de clase del análisis - CU Instalar GDA eXcriba.

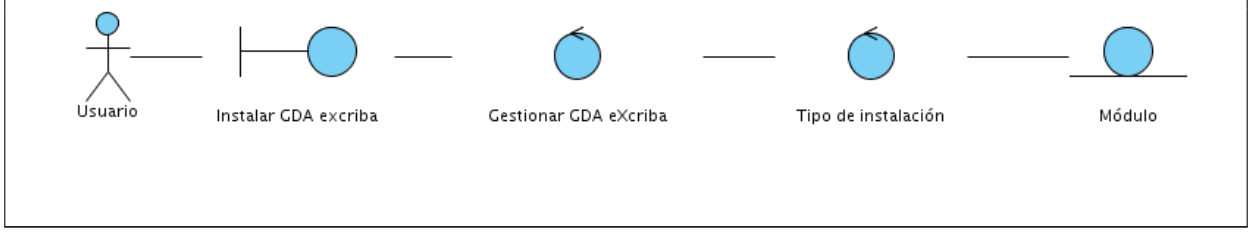


Tabla 3.1: Diagrama de clase del análisis - CU Instalar GDA eXcriba.

3.2. Diagrama de Interacción.

El Diagrama de Interacción es una clasificación del Diagrama de Colaboración o del Diagrama de Secuencia, estos diagramas se representan de manera diferente aunque cumplen con una característica y es que son isomorfos, lo que significa que un Diagrama de Secuencia puede transformarse mecánicamente en un Diagrama de Colaboración y viceversa.

Para una mejor representación y entendimiento del sistema en desarrollo se modelará a través de un Diagrama de Colaboración, para resaltar la organización estructural de los objetos que intercambian mensajes. La distribución de los objetos en el diagrama permite observar adecuadamente la interacción de un objeto con respecto a los otros y la vista dinámica de la interacción viene indicada por el envío de mensajes a través de los enlaces existentes entre los objetos. Hay que tener en cuenta que los mensajes se enumeran para ilustrar el orden en que se emiten.

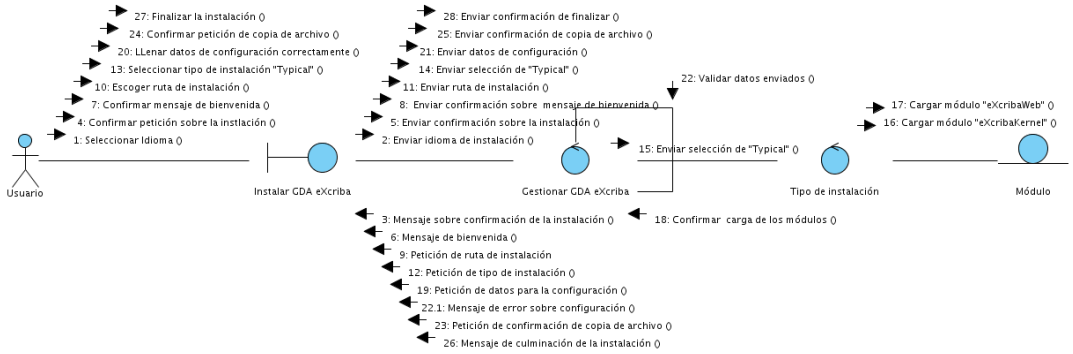
Diagrama de Colaboración - CU Instalar GDA eXcriba –*Tipo de Instalación Típica*–.


Tabla 3.2: Diagrama de Colaboración - CU Instalar GDA eXcriba –*Tipo de Instalación Típica*–.

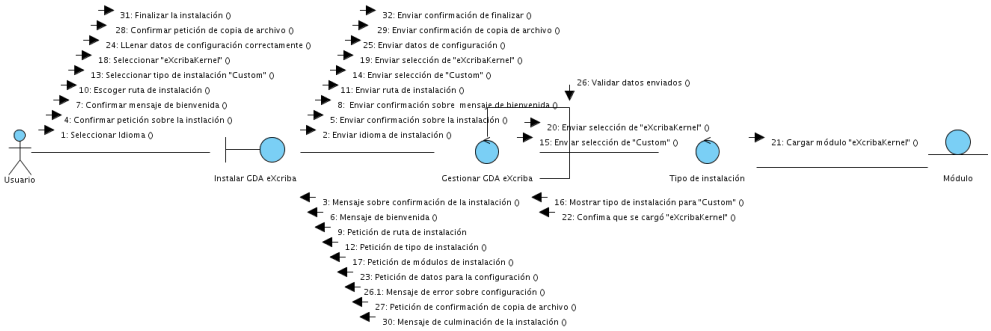
Diagrama de Colaboración - CU Instalar GDA eXcriba –*Tipo de Instalación Personalizada para el módulo eXcribaCore*–.


Tabla 3.3: Diagrama de Colaboración - CU Instalar GDA eXcriba –*Tipo de Instalación Personalizada para el módulo eXcribaCore*–.

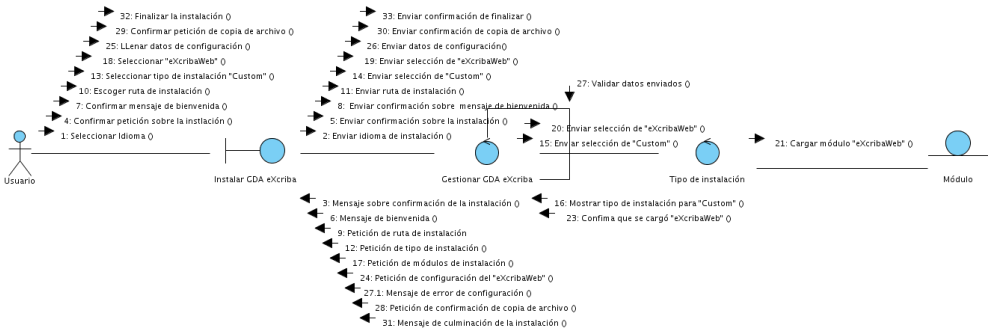
Diagrama de Colaboración - CU Instalar GDA eXcriba –*Tipo de Instalación Personalizada para el módulo eXcribaWeb*–.


Tabla 3.4: Diagrama de Colaboración - CU Instalar GDA eXcriba –*Tipo de Instalación Personalizada para el módulo eXcribaWeb*–.

3.3. Diagramas de Clases del Diseño.

El diagrama de clases del diseño proporciona una perspectiva estática del sistema, mostrando un conjunto de clases y sus relaciones entre ellas. Cada clase se caracteriza por tener atributos y métodos los que ayudan al desarrollo de la implementación. En este diagrama existen dos casos en que hay clases que son especialización de una generalización, lo que significa que la clase genérica es la padre y proporcionará herencia sobre las hijas.

Diagrama de Clases - CU Instalar GDA eXcriba.

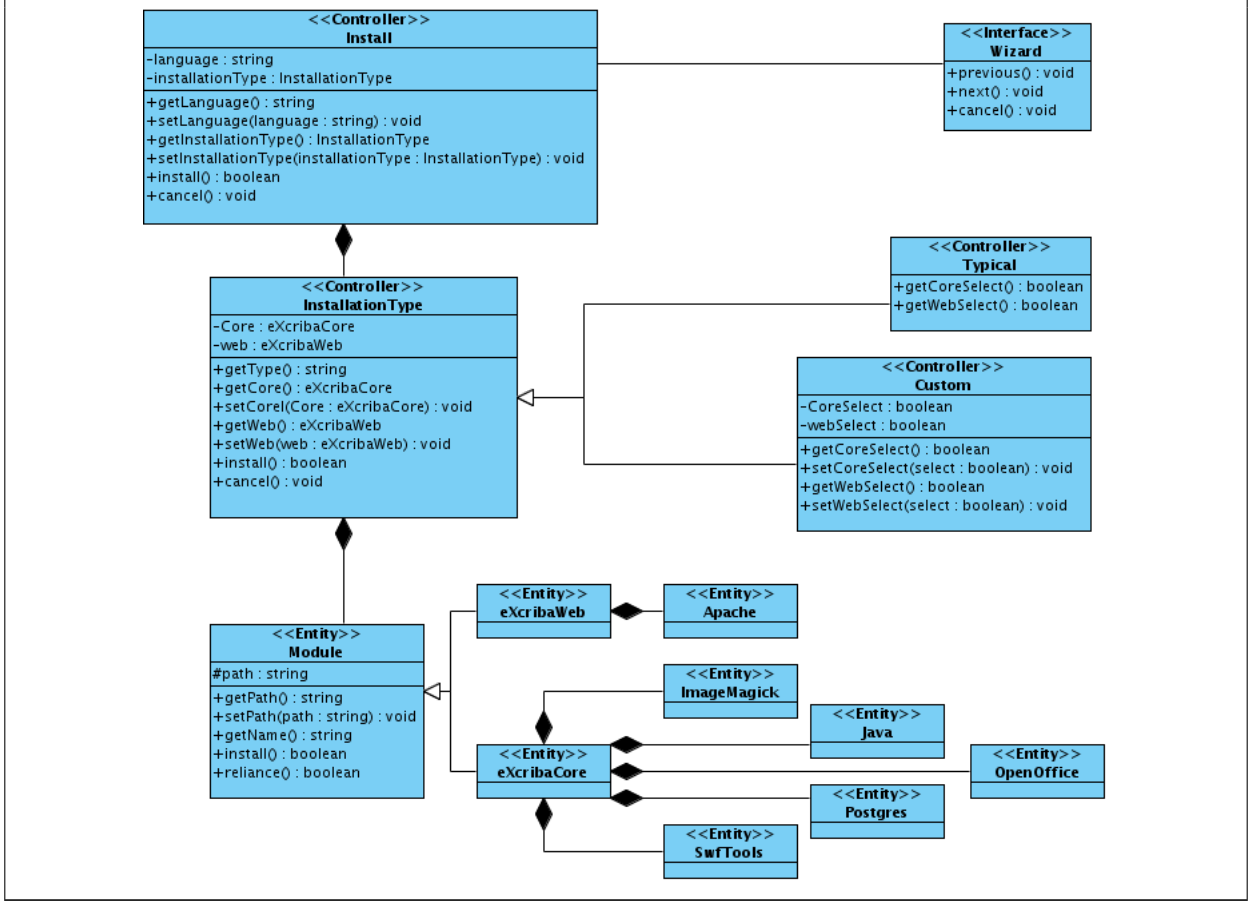


Tabla 3.5: Diagrama de Clases - CU Instalar GDA eXcriba.

3.4. Descripción de las clases.

A continuación se especifican en tablas – *para cada clase del Diagrama de Clases* – el nombre y el tipo de clase – *Clase Controladora, Clase Interfaz o Clase Entidad* –. Estarán definidos los atributos y el tipo de dato para cada uno de ellos. Además del nombre de la operación con los parámetros requeridos, incluyendo también una breve explicación sobre las operaciones.

Nombre:	Install
Tipo de clase:	Controladora.
Atributo:	**Tipo:**
language	string
installationType	InstallationType
Para cada responsabilidad:	
Nombre:	**Descripción:**
getLanguage() : string	Obtener el idioma de instalación.
setLanguage(language : string) : void	Establecer el idioma de instalación.
getInstallationType() : InstallationType	Obtener el tipo de instalación.
getInstallationType(installationType : InstallationType) : void	Establecer el tipo de instalación.
install() : boolean	Instalar e informar del proceso.
cancel() : void	Cancelar la instalación.

Tabla 3.6: Descripción de la clase: **Install**.

Nombre:	InstallationType
Tipo de clase:	Controladora.
Atributo:	**Tipo:**
Core	eXcribaCore
module	Module
Para cada responsabilidad:	
Nombre:	**Descripción:**
getType() : string	Obtener el tipo de instalación.
getCore() : eXcribaCore	Obener el módulo "*eXcribaCore*".
setCore(Core : eXcribaCore) : void	Establecer el módulo "*eXcribaCore*".
Continúa en la próxima página...	

getWeb() : eXcribaWeb	Obener el módulo "*eXcribaWeb*".
setWeb(web : eXcribaWeb) : void	Establecer el módulo "*eXcribaWeb*".
install() : boolean	Instalar e informar del proceso.
cancel() : void	Cancelar la instalación.

Tabla 3.7: Descripción de la clase: **InstallationType**.

Nombre:	Typical
Tipo de clase:	Controladora.
Atributo:	**Tipo:**
Para cada responsabilidad:	
Nombre:	**Descripción:**
getCoreSelect() : boolean	Obtener si el módulo "*eXcribaCore*" está o no seleccionado.
getWebSelect() : boolean	Obtener si el módulo "*eXcribaWeb*" está o no seleccionado.

Tabla 3.8: Descripción de la clase: **Custom**.

Nombre:	Custom
Tipo de clase:	Controladora.
Atributo:	**Tipo:**
CoreSelect	boolean
webSelect	boolean
Para cada responsabilidad:	
Nombre:	**Descripción:**
Continúa en la próxima página...	

getCoreSelect() : boolean	Obtener si el módulo "*eXcribaCore*" está o no seleccionado.
setCoreSelect(CoreSelect : boolean) : void	Establecer la selección o no del módulo "*eXcribaCore*".
getWebSlect() : boolean	Obtener si el módulo "*eXcribaWeb*" está o no seleccionado.
setWebSelect(webSelect : boolean) : void	Establecer la selección o no del módulo "*eXcribaWeb*".

Tabla 3.9: Descripción de la clase: **Custom**.

Nombre:	Module
Tipo de clase:	Entidad.
Atributo:	**Tipo:**
path	string
Para cada responsabilidad:	
Nombre:	**Descripción:**
getPath() : string	Obtener el directorio de instalación.
setPath(path : string) : void	Establecer el directorio de instalación.
getName() : string	Obtener el nombre del Módulo.
install() : boolean	Instalar e informar del proceso.
reliance() : boolean	Investigar sobre las dependencias para instalar este módulo.

Tabla 3.10: Descripción de la clase: **Module**.

Nombre:	Wizard
Continúa en la próxima página...	

Tipo de clase:	Interfaz.
Atributo:	**Tipo:**
titlePanel	Panel
infoPanel	Panel
actionPanel	Panel
Para cada responsabilidad:	
Nombre:	**Descripción:**
getTitlePanel() : Panel	Obtener el "*Panel*" de título.
setTitlePanel(titlePanel : Panel) : void	Establecer el "*Panel*" de título.
getInfoPanel() : Panel	Obtener el "*Panel*" de información.
setInfoPanel(infoPanel : Panel) : void	Establecer el "*Panel*" de información.
getActionPanel() : Panel	Obtener el "*Panel*" de acción.
setActionPanel(actionPanel : Panel) : void	Establecer el "*Panel*" de acción.

Tabla 3.11: Descripción de la clase: **Wizard**.

3.5. Arquitectura del sistema.

Los casos de uso solamente no son suficientes para conseguir un sistema de trabajo, se necesitan más cosas, estas "*cosas*" son la arquitectura. Puede pensarse que la arquitectura de un sistema es la visión común en la que todos los empleados – *desarrolladores y otros usuarios* – deben de estar de acuerdo, o como poco, aceptar. La arquitectura brinda una clara perspectiva del sistema completo, necesaria para controlar el desarrollo.[12]

La arquitectura de software abarca decisiones importantes sobre:

- La organización del sistema software.[14]
- Los elementos estructurales que compondrán el sistema y sus interfaces, junto con sus comportamientos, tal y como se especifican en las colaboraciones entre estos elementos.[14]
- La composición de los elementos estructurales y del comportamiento en sistemas progresivamente más grandes.[14]
- El estilo de la arquitectura que guía esta organización: los elementos y sus interfaces, sus colaboraciones y su composición.[14]

Estudiando las condiciones en que se encuentra enmarcado el software en desarrollo es necesario – *doblegado por la selección del sistema diseñador de instaladores escogido* – seleccionar una arquitectura que guíe el proceso de desarrollo del software. En la programación por capas podemos enunciar las siguientes:

Capa de presentación: que en este caso esta formada por los componentes de interfaz de usuario. Los componentes de interfaz de usuario pueden ser vistos como la parte con la cual interactúa el usuario. Las ventanas o páginas web, por decirlo de alguna manera. Los componentes de proceso de interfaz de usuario podríamos asociarlos a clases de tipo controladora en UML. Es decir estos encapsulan lógica de navegación y control de eventos de la interfaz.[15]

Capa de negocios: encapsula lógica de negocios. Los servicios de esta capa son encapsulados en tres tipos de componentes. Las entidades empresariales, que representan objetos que van a ser manejados o consumidos por toda la aplicación, estos podrían ser un modelo de objetos, xml, estructuras de datos...que permitan representar objetos que han sido identificados durante el modelamiento.[15]

Capa de acceso a datos: que contiene clases que interactúan con la base de datos. Estas clases surgen como una necesidad de mantener la cohesión o clases altamente especializadas que ayuden a reducir la dependencia entre las clases y capas.[15]

Estudio previo realizado, conducido además por lo expuesto hasta el momento, arroja que se hace necesaria la utilización de un estilo arquitectónico en capas, por lo que se propone la utilización del patrón arquitectónico: **Modelo Vista Controlador**.

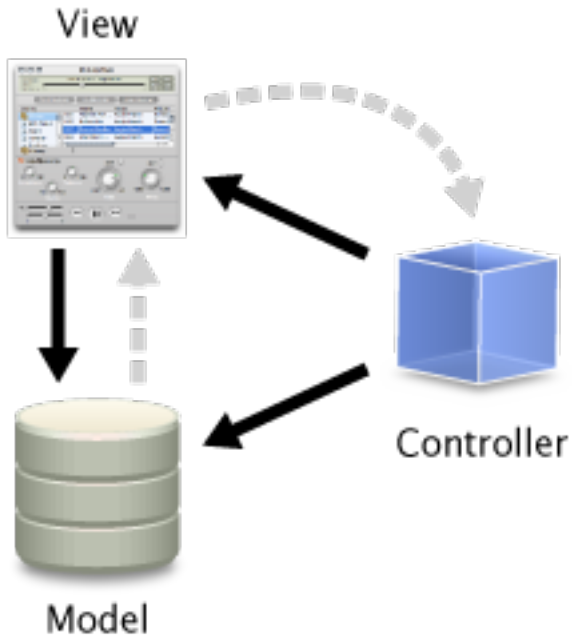


Figura 3.1: Patrón arquitectónico Modelo Vista Controlador.

Los Fundamentos básicos del MVC son los siguientes:

Modelo: Esta capa sirve como representación específica de toda la información con la cual el sistema va trabajar. El modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.[16]

Vista: Presenta el modelo con el que va a interactuar el usuario, más conocida como interfaz.[17]

Controlador: El controlador responde más bien a eventos, normalmente son acciones que el usuario invoca, implica cambios en el modelo y también en la vista.[16]

3.5.1. Adaptación del patrón arquitectónico Modelo Vista Controlador al desarrollo del Instalador Multiplataforma para el Gestor de Documentos Administrativos eXcriba.

Como todo proceso de desarrollo de software tiene características peculiares es necesario adaptar la arquitectura candidata para su utilización. Por consecuencia es prioritario hacerlo también dentro de la presente construcción e implementación. Se exponen las acotaciones pertinentes:

Modelo: En esta capa ha de garantizarse el funcionamiento de los módulos específicos que contiene el GDA eXcriba – *eXcribaWeb y eXcribaCore* – dentro de su instalador; relegando la llamada capa de datos a los ficheros físicos que contienen los citados módulos.

Vista: Dentro de este nivel de implementación han de crearse todas las interfaces con las que el usuario va a interactuar. Las ventanas y demás componentes se realizan sobre el lenguaje Tk.

Controlador: Los eventos a los que se les darán respuestas en esta capa serán materializados con las bondades del lenguaje TcL, creando con este todas las acciones que son imprescindibles para el funcionamiento del software.

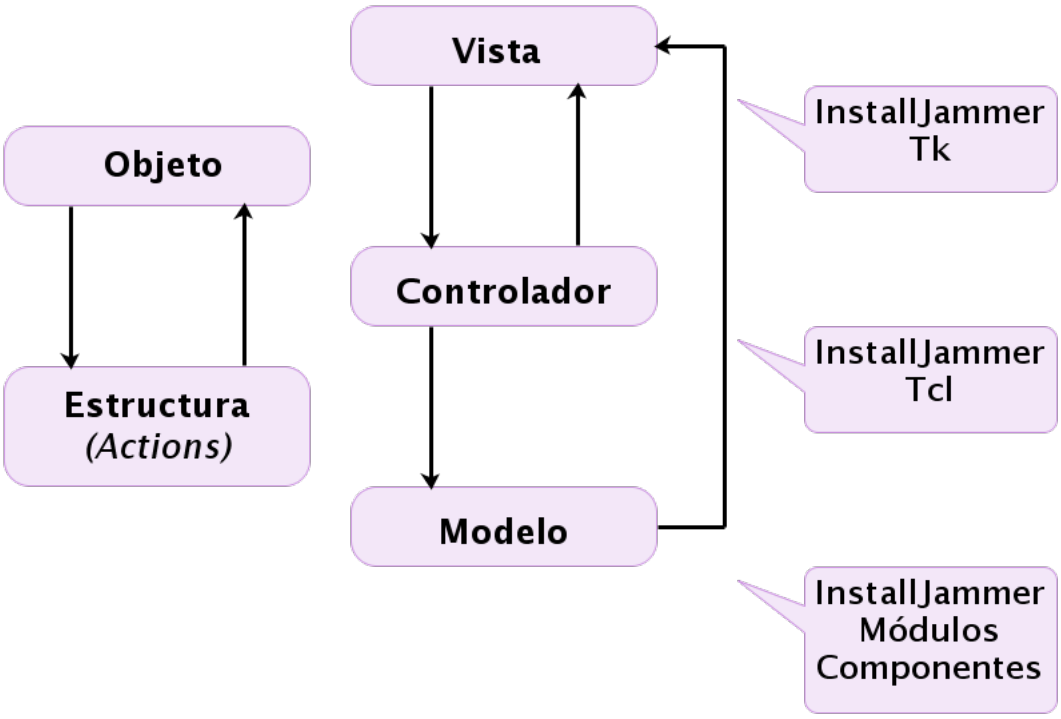


Figura 3.2: Patrón arquitectónico Modelo Vista Controlador en el Instalador Multiplataforma para el Gestor de Documentos Administrativos eXcriba.

Nota: Deviene en imprescindible acotar la estrecha relación existente entre los vocablos *Objeto* y *Estructura*. Donde el primero resulta ser una *instancia de una clase – propiamente en términos de programación y/o diseño* – y el segundo es relativo a las acciones que se realizan dentro de la API del InstallJammer; para exponerlo mejor se hace referencia a una macro como una *action*, tratándose la misma como un objeto en el análisis y diseño del sistema.

En este capítulo se ha especificado el comportamiento del sistema a través de los Diagramas de Interacción y Diagramas de Clases y además con el Modelo de Clase de Análisis para lograr una mejor comprensión de lo que se está construyendo, obteniendo una visión general con la ayuda del UML.

El análisis previo contribuyó en gran medida al diseño del Instalador Multiplataforma del GDA eXcriba contando con todas las clases necesarias para su implementación y las relaciones entre ellas.

Con la utilización del estilo arquitectónico Modelo Vista Controlador se puede tratar cada capa del desarrollo del instalador del GDA eXcriba como un sistema independiente. Donde es imprescindible conocer y crear bien las API de cada uno de estos niveles de trabajo para lograr una mejor interacción entre ellos. Con el desarrollo de la investigación están creadas las condiciones para llevar a cabo la implementación del Instalador Multiplataforma del GDA eXcriba.

Capítulo 4

Implementación y Prueba.

En el presente capítulo es necesario tener definido un procedimiento que dirija el desarrollo de las pruebas para poder explorar todos los caminos implementados usando un orden lógico. De esta forma se logra ampliar las probabilidades de detectar errores antes que el sistema se despliegue, logrando verificar que la aplicación o el sistema se comporta según los requerimientos establecidos; en este trabajo se propone un procedimiento basado en casos de uso, principal artefacto de la metodología RUP, para llevar a cabo el proceso de pruebas.

4.1. Despliegue del GDA eXcriba.

Es necesario modelar para concebir el sistema, se hace uso del Diagrama de Despliegue "*el cual se conforma de nodos de procesamiento y componentes para una correcta configuración del sistema en tiempo de ejecución.*"[18]

El usuario para disfrutar de los servicios necesarios del GDA eXcriba depende de un ordenador, en calidad de cliente, para acceder a un servidor de aplicaciones web donde se encuentra alojado el módulo Web de eXcriba, el mismo realiza una conexión al módulo eXcribaCore en un Apache Tomcat, este consume las bondades de un gestor de bases de datos y de esta forma queda organizada la lógica de la implementación del sistema.

Diagrama de Despliegue.

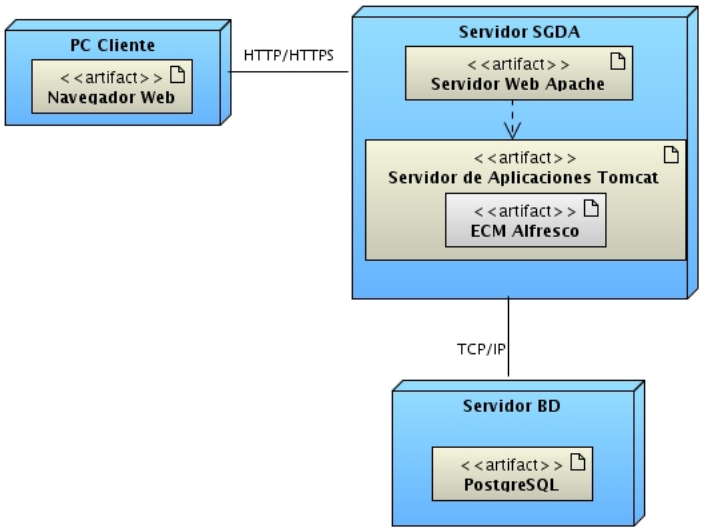


Tabla 4.1: Diagrama de Despliegue.

4.2. Componentes.

Para lograr una organización lógica de la implementación de un sistema se hace necesario del Diagrama de Componentes "*el cual se conforma de componentes y dependencias entre ellos.*"[18]. Representa todos los tipos de elementos software que entran en la fabricación de la aplicación informática, desde simples archivos y paquetes hasta bibliotecas cargadas dinámicamente. Este diagrama describe los elementos físicos del sistema y sus relaciones.

El siguiente diagrama detalla la interacción de varios componentes para funcionar como un todo; girando alrededor del *InstallKit*, el cual está encargado de compilar la solución final del producto apoyándose en otros componentes previamente concebidos: *eXcribaWeb* y *eXcribaCore – los pilares principales que contienen los módulos del mismo nombre –*, Postgres, OpenOffice, Java y Apache; estos son las dependencias del GDA eXcriba. *InstallKit* se relaciona directamente con la *API* que provee el *InstallJammer* para la creación de acciones sobre *TcL* e interfaces sobre *Tk*. A esta *API* se le brindan componentes adicionales para complementar los requerimientos, entre los que se encuentran *Language*, ofreciendo un conjunto de lenguajes y *Resource* que aporta recursos adicionales como imágenes y ficheros externos.

Diagrama de Componentes.

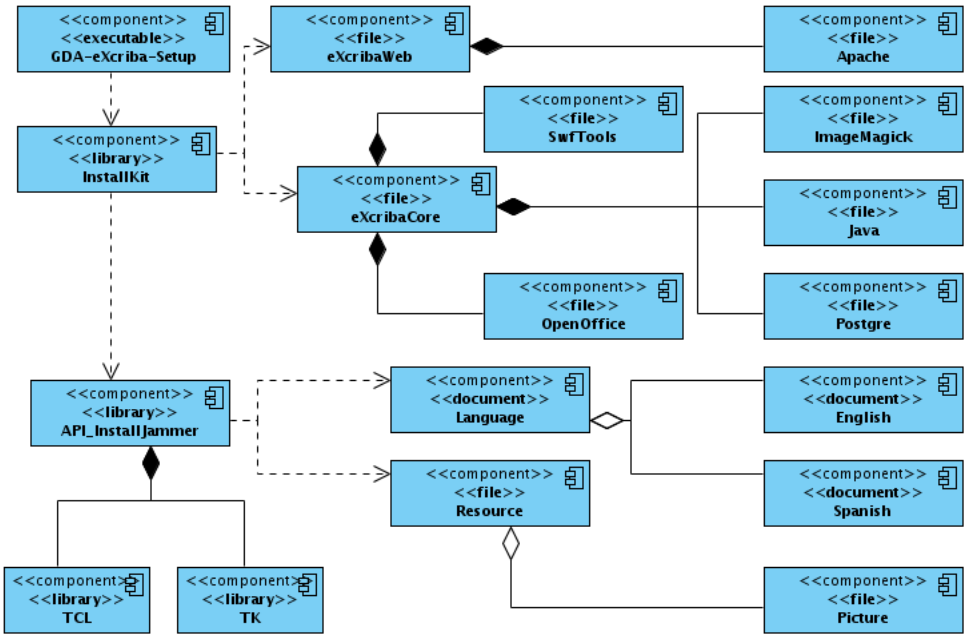


Tabla 4.2: Diagrama de Componentes.

4.3. Modelo de prueba.

En el presente trabajo se realizó el proceso de *prueba de caja negra* para obtener un resultado se hizo necesario tener en cuenta que si se usó la metodología RUP se dependerá de la descripción del caso de uso, este es el artefacto por el que se sigue toda la traza del sistema, es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta el funcionamiento interno del software.

Pruebas de caja negra Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software.[19]

Algunas ventajas de este tipo de pruebas podrían decirse que son:

- La prueba termina siendo imparcial porque el que diseña el software y el que lo prueba son independientes.[20]
- El que lo prueba no necesita conocimientos de programación.[20]
- Las pruebas se realizan desde un punto de vista de usuario y no como programador, tomando en cuenta todas las funciones.[20]

La prueba de Caja Negra no es una alternativa a las técnicas de prueba de la Caja Blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la Caja Blanca.[19]

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

1. Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.[19]
2. Técnica del Análisis de Valores Límites: esta Técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.[19]
3. Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.[19]

Conociendo una función específica para la que fue diseñado el producto, se pueden diseñar pruebas que demuestren que dicha función está bien realizada. Las pruebas son llevadas a cabo sobre la interfaz del software, es decir evaluando sus funciones, actuando sobre ella como una caja negra, proporcionando unas entradas y estudiando las salidas valorando si son o no las esperadas.“Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así en número de clases de prueba que hay que desarrollar.”[19]

Modelo de Prueba

Instalar GDA eXcriba		
Seleccionar Idioma		
Entrada	**Resultados**	**Condiciones**
Seleccionar idioma correctamente.	El sistema muestra una interfaz para confirmar que el SGID eXcriba se va a instalar en su equipo.	1. El usuario ejecuta el instalador de SGID eXcriba. 2. El usuario selecciona el idioma y presiona el botón "Ok".
Cancelar la selección del idioma	El sistema cancela el proceso de instalación.	1. El usuario ejecuta el instalador de SGID eXcriba. 2. El usuario cancela la selección del idioma y confirma la acción presionando el botón "Si".
Seleccionar Ruta para la instalación		
Entrada	**Resultados**	**Condiciones**
Seleccionar ruta de instalación correctamente.	El sistema continúa con el proceso de instalación.	1. El usuario especifica la ruta y presiona el botón "Siguiente".
		Continúa en la próxima página...

Cancelar la instalación una vez seleccionada la ruta	El sistema cancela el proceso de instalación.	1. El usuario especifica la ruta y presiona el botón “Cancelar” y confirma la acción presionando el botón “Si”.
Seleccionar Tipo de instalación		
Entrada	**Resultados**	**Condiciones**
Seleccionar tipo de instalación correctamente.	El sistema continúa la instalación y muestra la interfaz para seleccionar una base de datos.	1. El usuario selecciona la opción “Típica” y presiona el botón “Siguiente”.
Cancelar la selección del tipo de instalación.	El sistema cancela el proceso de instalación.	1. El usuario selecciona la opción “Típica” y presiona el botón “Cancelar” y confirma la acción presionando el botón “Si”.
Configurar Gestor de Base de Datos		
Entrada	**Resultados**	**Condiciones**
Seleccionar base de datos y configurar correctamente.	El sistema continúa la instalación.	
Continúa en la próxima página...		

Configurar base de datos incorrectamente.	El sistema muestra un mensaje de error: "*en dependencia del error encontrado*".	1. El usuario inserta los datos para la configuración de la base datos y presiona el botón "Siguiente".
Seleccionar tipo de instalación personalizada		
Entrada	**Resultados**	**Condiciones**
Seleccionar Tipo de instalación Personalizada.	El sistema continúa el proceso de instalación.	
Seleccionar Tipo de Instalación Personalizada incorrectamente.	El sistema muestra un mensajer: "*exigiendo las dependencias no marcadas para la instalación*".	1. El usuario selecciona la opción "Personalizar" y presiona el botón "Siguiente". 2. El usuario selecciona los módulos a instalar y presiona el botón "Siguiente".

Tabla 4.3: Modelo de Prueba.

Comité de expertos.

La realización del *Comité de expertos* se efectúa para obtener valoraciones críticas sobre diferentes variables que se deseen medir en el trabajo realizado. Para el Instalador Multiplataforma para el Gestor de Documentos Administrativos eXcriba se seleccionaron tres expertos y sus valoraciones arrojaron resultados que demuestran la viabilidad de la solución automatizada. Para un mejor entendimiento se expone dicho estudio en la siguiente tabla, sus resultados pueden apreciarse en la gráfica.

Instalación del GDA eXcriba								
	Manual				Automatizada			
Expertos:	U[1]	C[2]	D[3]	T[4]	U	C	D	T
Dayelis Blanco Hernández	2	3	5	3	5	4	0	5
Eder Despaigne Herrera	3	3	4	4	5	4	1	5
Marcos L. Ortiz Valmaseda	2	2	4	4	4	4	1	4
Total:	7	8	13	11	14	12	2	14
Media:	2.3	2.7	4.3	3.7	4.7	4.0	0.7	4.7

Tabla 4.4: Comité de expertos para la Instalación del GDA eXcriba.

Nota: para más información sobre los expertos consultados ver **Anexo A.1. Relación del comité de expertos**.

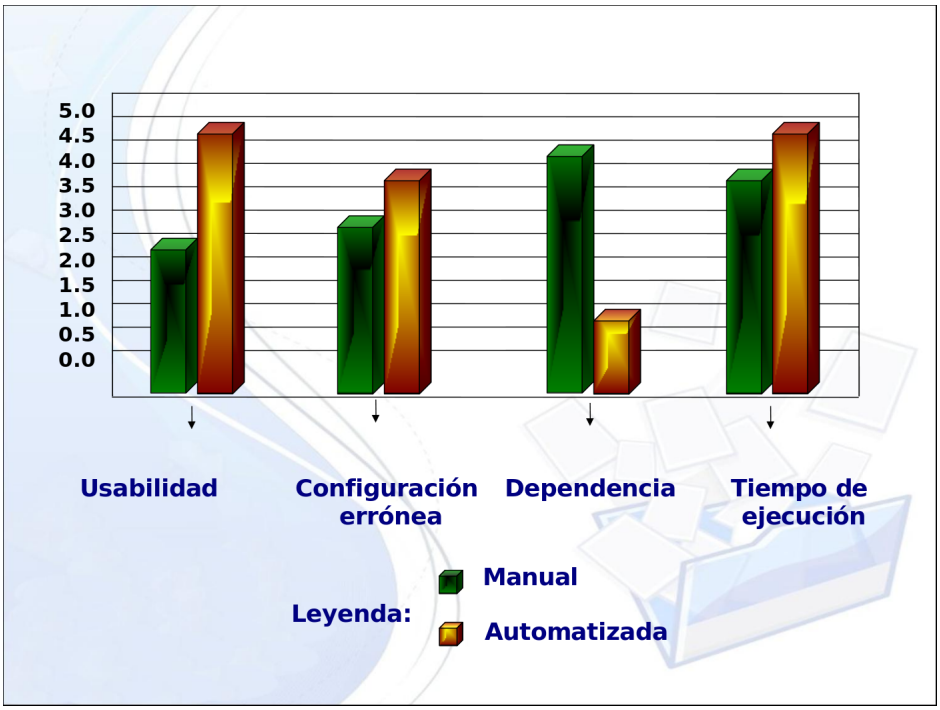


Figura 4.1: Resultados según el comité de expertos.

[1]Valoración de usabilidad.

[2]Valoración de posibles configuraciones erróneas.

[3]Valoración de dependencia hacia el equipo de desarrollo.

[4]Valoración de tiempo para realizar la tarea.

Disponiendo de un correcto Análisis y Diseño del Sistema se le abrió paso automáticamente a la fase de Implementación y Prueba, obteniendo con esta un resultado tangible que permitió la aparición de un entregable. Contrastando con la construcción fehaciente y final del sistema donde se implementan las clases ya diseñadas y se obtiene un resultado coherente con las descripción del Caso de Uso; interpretándose esto como un logro eficaz de este proceso. Puede procederse aquí a la realización de pruebas al sistema.

Existió la necesidad de un buen diseño de caso de prueba de caja negra para asegurarse de que el sistema es correctamente disponible, cumpliendo con la capacidad de resolver el problema para el cual fue creado, para ello se tuvo en cuenta la información que brinda el caso de uso, ya que una vez que se posee la descripción de un caso de uso se puede comenzar a diseñar los casos de pruebas que lo validan. "La realización de las pruebas basadas en casos de usos es un proceso que consta de tres pasos fundamentales: diseño, implementación del método y análisis de los resultados y puede tener varias iteraciones."[21]

Conclusiones

- El estudio de los principales sistemas diseñadores de instaladores para aplicaciones multiplataforma, las funcionalidades de instalación del ECM Alfresco, los diferentes escenarios en los cuales se puede encontrar el software GDA eXcriba y sus principales componentes y dependencias fueron aspectos a tener en cuenta en el desarrollo del un instalador multiplataforma para el GDA eXcriba con el cual se logra solventar la necesidad de automatizar la ditrubución del sistema.

- Mediante el uso de la Metodoloía RUP se desarrollaron todas la faces del ciclo de vida del instalador multiplataforma para el GDA eXcriba.

- La aplicación de prueba de caja negra demostró que el instalador multiplataforma para el GDA eXcriba es óptimo para establecer una correcta instalación del GDA eXcriba.

Recomendaciones

- Automatización total del proceso de compilación, ofreciendo agilidad en la consecución de las dependencias.
- Realizar la compilación de las dependencias para las plataformas que sean diferentes de GNU\Linux, asegurando la optimización y estabilidad de las dependencias del GDA eXcriba en el resto de las plataformas.
- Hacer una descripción detallada del proceso de compilación aportando elementos tangibles que tributen y soluciones de optimización, logrando un mejor desempeño del GDA eXcriba y sus dependencias.
- Sincronizar el idioma del instalador con el idioma en que se utilizará el GDA eXcriba, logrando una mejor adaptación del GDA eXcriba en su internacionalización.

Glosario de términos

A

Actor: Un conjunto coherente de roles que los usuarios de caso de uso desempeñan cuando interactúan con estos casos de uso. [12].

C

Capa: Una capa es un conjunto de subsistemas que comparten el mismo grado de generalidad y de volatilidad en las interfaces: las capas inferiores son de aplicación general a varias aplicaciones y deben poseer interfaces más estables, mientras que las capas más altas son más dependientes de la aplicación y pueden tener interfaces menos estables. [12].

Caso de Uso: Una descripción de un conjunto de secuencias de acciones, incluyendo variaciones, que un sistema lleva a cabo y que conduce a un resultado observable de interés para un actor determinado. [12].

Clase: Una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. [12].

Compilación: Proceso por el cual se *traduce* un programa escrito en un lenguaje de programación a lo que realmente entiende el ordenador.

Código Abierto: Es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios

prácticos de compartir el código que a las cuestiones morales y/o filosóficas, las cuales destacan en el llamado *software libre*.

D

Diagrama de Clases: Un diagrama que muestra un conjunto de clases, interfaces y colaboraciones y las relaciones entre éstos; los diagramas de clases muestran el diseño de un sistema desde el punto de vista estático; un diagrama que muestra una colección de elementos –*estáticos*–declarativos. [12].

Diagrama de Interacción: Un diagrama que muestra una interacción, consistente con un conjunto de objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos; los diagramas interacción tratan la vista dinámica de un sistema; un término genérico que se aplica a varios tipos de diagramas que enfatizan las interacciones de objetos, incluyendo diagramas de colaboración, diagramas de secuencia y diagramas de actividad. [12].

G

Gestor de contenido: Es un programa que permite crear una estructura de soporte para la creación y administración de contenidos, principalmente en páginas web, por parte de los participantes.

I

Instalador multiplataforma: Instaladores de programas que funcionan nativamente en varias plataformas – *en GNU\Linux y Microsoft Windows por ejemplo.* –.

Interfaz gráfica: La interfaz gráfica de usuario, conocida también como GUI es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

L

Lenguaje de programación: Idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras.

R

Requisitos: Flujo de Trabajo fundamental cuyo propósito esencial es orientar el desarrollo hacia el sistema correcto. Esto se lleva a cabo mediante la descripción de los requerimientos del sistema de forma tal que se pueda llegar a un acuerdo entre el cliente –*incluyendo los usuarios*– y los desarrolladores del sistema, acerca de lo que el sistema debe hacer y lo que no. [12].

S

Sistema Operativo: Un Sistema operativo es un software que actúa de interfaz entre los dispositivos de hardware y los programas de usuario o el usuario mismo para utilizar un computador.

Software: Equipamiento lógico o soporte lógico de una computadora digital; comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de tareas específicas. Término informático que define a los programas que se utilizan en una computadora o hacen funcionar completamente a esta.

Referencias Bibliográficas

[1] M. M. Mugia, *Gestión documental y organización de archivos.* Félix Valera., 2005.

[2] N. L. N. Camallea and R. C. Abalo, *Diccionario de Informatica.* Editorial Científico-Técnica., 2005.

[3] G. Asinsten and J. C. Asinsten, *Instalación de Programas.* Tiza y Mouse, 2001.

[4] "Definicion archivos texto binarios."
http://www.mitecnologico.com.

[5] "Multi-Platform java installer builder - install4j."
http://www.ej-technologies.com.

[6] "InstallShield - MSI windows installer and InstallScript software installation - flexera software."
http://www.flexerasoftware.com.

[7] "BitRock :: Easy to use multiplatform installers and customized open source stacks."
http://bitrock.com.

[8] "InstallJammer multiplatform - a free, open source, cross platform installer - home."
http://www.installjammer.com.

[9] "Tcl developer site."
http://www.tcl.tk.

[10] SOFTHOUSE, "Scrum in five minutes.," 2006.

[11] H. Kniberg, *Scrum and XP from the Trenches, How we do Scrum.* InfoQ, 2007.

[12] J. Rumbaugh, I. Jacobson, and G. Booch, *El Proceso Unificado de Desarrollo de Software.* PEARSON EDUCACIÓN, S.A., 2000.

[13] "Bienvenidos al entorno virtual de aprendizaje."
http://eva.uci.cu.

[14] M. G. Ginestà, Álvaro Peña González, M. H. Matías, and D. A. Pérez, *Ingeniería del Software en entornos de Software Libre.* Fundació per a la Universitat Oberta de Catalunya, 2005.

[15] "Arquitectura de tres capas."
http://www.di-mare.com.

[16] "Modelo vista controlador."
http://www.neleste.com.

[17] "Patrón modelo-vista-controlador."
http://www.proactiva-calidad.com.

[18] J. Rumbaugh, I. Jacobson, and G. Booch, *UML, El Lenguaje Unificado de Modelado.* Fernando Berzal., 2004.

[19] "Bienvenidos al entorno virtual de aprendizaje."
http://eva.uci.cu.

[20] "Prueba de caja negra."
http://www.scribd.com.

[21] R. S. Pressman, *Ingeniería de Software un enfoque práctico, 5ta edición.* Mac Graw Hill., 1995.