

Universidad de las Ciencias Informáticas



Título: Instalador para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0

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

Autora

Neyvis Herrera Ramírez

Tutores

Ing. Aliander Capdezuñer González

Ing. Gilberto Lissabet Hernández

La Habana, Junio del 2013

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a _____ de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

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

Neyvis Herrera Ramírez

Firma de Autor

Ing. Aliander Capdezuñer González

Firma del tutor

Ing. Gilberto Lissabet Hernández

Firma del tutor

Agradecimientos

Agradezco desde el fondo de mi corazón:

Primeramente un beso especial a mi mamá por darme la educación que me dio y enseñarme que las personas debemos superarnos para ser alguien en la vida. Además, de que siempre estuvo apoyándome en cada decisión que tome en mi vida como estudiante. A ti mami un agradecimiento doblemente especial.

A toda mi familia en especial mi papá y mis abuelos por siempre estar a la expectativa de lo que la niña de la casa hacía, a ustedes todo mi cariño por estar ahí para mí.

A mi novio Ernesto por ayudarme en todo momento a superar cada problema y hacerme creer que no había nada difícil, cuando yo no veía la manera de hacer las cosas. A ti, titi muchas gracias por todo.

A una persona que a pesar de no ser un familiar yo lo considero como parte de la familia y lo quiero muchísimo, Nelsito, gracias por tratarme como una hija y siempre estar ahí para ayudar en todo lo que hiciera falta.

A mis tíos Tomas y Dulce por brindarme el calor de una familia cuando yo estaba lejos de la mía todos estos años. Ustedes son para mí como mis segundos abuelos, los quiero mucho.

A mis amistades, en especial Arletys, Ricardo y Yaniel que tantas horas de sueño perdimos estudiando para llegar hasta donde estamos hoy, ustedes saben que son los mejores amigos del mundo.

A todos los profesores que cada año nos brindaron sus conocimientos y nos ayudaron a vencer las diferentes materias.

A todas aquellas personas que me ayudaron y se preocuparon por mí. A todos un beso y un abrazo, cada uno ocupa un importante lugar en mi corazón.

Dedicatoria

Dedico este trabajo a toda mi familia, en especial a mi mamá que tanto me ha apoyado y se ha sacrificado por mí durante todos estos años de vida estudiantil. Quisiera que por medio de este trabajo de diploma supiera que todo el esfuerzo no fue en vano y que su ejemplo fue una gran inspiración para mí. Gracias mami por enseñarme bien.

Resumen

Hoy en día el desarrollo del software ha tomado un gran auge en el mundo debido al avance de las tecnologías, pues mediante el software se hace posible la eficiente realización de los diferentes procesos de las industrias. Cuba ha incursionado en el campo del desarrollo de software llevando a cabo una ardua labor en la informatización de las diferentes empresas. Producto de esto la Universidad de las Ciencias Informáticas se ha dado a la tarea de automatizar diferentes instituciones cubanas entre las que se encuentran los medios de prensa. Para esta organización se desarrolló el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel, el cual no cuenta actualmente con un instalador que facilite el proceso de despliegue del sistema. El desarrollo de un instalador para este sistema permitirá la automatización del proceso de instalación, posibilitando la rapidez y la independencia del usuario con respecto al equipo de desarrollo. En este documento se describen detalladamente las principales tecnologías y herramientas utilizadas para el desarrollo del instalador. Además de realizar comparaciones entre varios sistemas homólogos, en busca de mejores propuestas para desarrollar el sistema, también se muestran los principales artefactos generados durante el desarrollo de la aplicación y se especifican las pruebas realizadas para validar los requisitos con los cuales la aplicación debe cumplir. Como resultado se obtuvo un instalador que se encarga de poner en funcionamiento el sistema posibilitando una mayor agilidad en el proceso de instalación y facilidad de uso para los usuarios finales.

Palabras claves: Software, Ecumene Pyxel, Automatizar, Instalador.

Índice de contenidos

Introducción.....	1
Capítulo 1: Fundamentación teórica que sustenta la investigación para la realización del Instalador de Ecumene Pyxel.....	5
Introducción.....	5
1.1 Instaladores de Software.....	5
1.2. Proceso de instalación.....	5
1.3. Surgimiento de las vías de instalación de software.....	6
1.4 Estudio de sistemas homólogos.....	8
1.4.1 Instalador de Smart Keeper.....	8
1.4.2 Instalador de MOCICE.....	9
1.5 Sistemas generadores de instaladores.....	9
1.5.1 BitRock InstallBuilder 8.....	10
1.5.3 InstallAnywhere 2012.....	11
1.5.4 InstallJammer 1.2.15.....	12
1.6 Metodología de desarrollo.....	14
1.7 Herramientas.....	15
1.7.1 Herramienta Case: Visual Paradigm 8.0 para UML.....	15
1.7.2 ForeUI.....	16
Conclusiones del capítulo.....	16
Capítulo 2: Análisis y Diseño del Instalador de Ecumene Pyxel.....	18
Introducción del capítulo.....	18
2.1 Características del Sistema.....	18
2.2 Mapa de Procesos.....	19
2.3 Modelo Conceptual.....	19
2.4 Descripción de procesos de negocio.....	20
2.5 Requisitos del sistema.....	21
2.5.1. Requisitos Funcionales.....	22
2.5.2 Requisitos No funcionales.....	25
2.3 Diagrama de Casos de Uso del Sistema.....	26
2.4 Especificación de Casos de Uso.....	26
2.5 Arquitectura del sistema.....	27
2.5.1 Adaptación del patrón arquitectónico Modelo Vista Controlador al desarrollo del instalador del Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel.....	27
2.6 Diagrama de Clases del Análisis.....	28
2.7 Diagramas de Interacción.....	28
2.7.1 Diagrama de colaboración: CU_Instalar SGDID Ecumene Pyxel -Tipo de instalación típica.....	29
2.7.2 Diagrama de colaboración: CU_Instalar SGDID Ecumene Pyxel -Tipo de instalación avanzada para el módulo Núcleo.....	30
2.7.3 Diagrama de colaboración: CU_Instalar SGDID Ecumene Pyxel -Tipo de instalación avanzada para el módulo Base de Datos.....	30
2.7.4 Diagrama de colaboración: CU_Instalar SGDID Ecumene Pyxel -Tipo de instalación avanzada para el módulo Interfaz.....	31
2.8 Modelo de Despliegue.....	31
Conclusiones del Capítulo.....	32
Capítulo 3: “Implementación y Prueba del Instalador de Ecumene Pyxel”.....	33

Introducción del capítulo.....	33
3.1 Diagrama de Componentes.....	33
3.2 Utilización del InstallJammer.....	34
3.2.1 Cargar los módulos del SGDID Ecumene Pyxel.....	34
3.2.2 Diseño de interfaces.....	35
3.2.3 Validación de campos.....	36
3.2.4 Conexión con la Base de Datos.....	38
3.2.5 Ejecutar programas externos.....	39
3.2.6 Escribir ficheros.....	39
3.2.7 Comprobar módulo seleccionado para instalar.....	40
3.2.8 Creación de condiciones.....	40
3.2.9 Creación de los grupos de acción.....	41
3.2.10 Verificar dependencias.....	42
3.2.11 Acción para modificar en valor de la variables virtual.....	42
3.2.12 Leer un fichero dentro una variable de texto virtual.....	42
3.2.13 Eliminar base de datos instalada.....	43
3.2.14 Acción para desinstalar archivos sobrantes.....	43
3.2.15 Instalación en modo consola.....	44
3.2.15.1 Acción para preguntar en la consola si o no.....	44
3.2.15.2 Acción para ejecutar acciones.....	44
3.2.15.3 Acción para los mensajes en consola.....	45
3.2.15.4 Acción para obtener la entrada de usuario en la consola.....	45
3.3 Modelo de pruebas	45
3.3.1 Casos de prueba.....	46
3.3.2 Pruebas de portabilidad.....	46
3.3.3 Detección de no conformidades.....	47
Conclusiones del capítulo.....	49
Conclusiones.....	50
Recomendaciones.....	51
Bibliografía.....	52
Anexos.....	55
Glosario de términos.....	60

Índice de figuras

Fig 1: Mapa de procesos	19
Fig 2: Modelo Conceptual.....	20
Fig 3: Diagrama de Casos de Uso.....	26
Fig 4: Arquitectura Modelo Vista Controlador (MVC).....	27
Fig 5: Diagrama de clases del análisis para el CU_Instalar SGDID Ecumene Pyxel.....	28
Fig 6: CU_Instalar SGDID Ecumene Pyxel -Tipo de instalación típica.....	29
Fig 7: DC: CU_Instalar SGDID Ecumene Pyxel: Tipo de instalación avanzada para el módulo Núcleo.....	30
Fig 8: DC: CU_Instalar SGDID Ecumene Pyxel: Tipo de instalación avanzada para el módulo BD.....	30
Fig 9: DC: CU_Instalar SGDID Ecumene Pyxel: Tipo de instalación avanzada para el módulo Interfaz. ...	31
Fig 10: Modelo de Despliegue del instalador para el SGDID Ecumene Pyxel.....	31
Fig 11: Diagrama de Componentes.....	34
Fig 12: Módulos adicionados al sistema.....	35
Fig 13: Acción para agregar un panel en blanco.....	35
Fig 14: Agregar un campo de tipo texto.....	35
Fig 15: Agregar un campo de tipo entrada de datos.....	36
Fig 16: Panel de configuración del instalador.....	36
Fig 17: Verificar el formato del campo perteneciente al IP.....	37
Fig 18: Verificar el formato del campo perteneciente al puerto.....	37
Fig 19: Acción para validar todos los campos.....	38
Fig 20: Función para la conexión con la base de datos.....	39
Fig 21: Funcionalidad: Ejecutar programa externo.....	39
Fig 22: Funcionalidad para escribir un fichero.....	39
Fig 23: Fichero “restoredatabase.sh”.....	40
Fig 24: Función para verificar un módulo seleccionado.....	40
Fig 25: Condición para verificar los paneles que se deben mostrar.....	41
Fig 26: Creación de un grupo de acción.....	41
Fig 27: Condición para verificar las dependencias.....	42
Fig 28: Acción “Set Virtual Text”.....	42
Fig 29: Acción “Read File Into Virtual Text”.....	43
Fig 30: Función para eliminar la base de datos.....	43
Fig 31: Acción “Uninstall Leftover Files”.	44
Fig 32: Acción “Console Ask Yes or No”.....	44
Fig 33: Acción “Execute Action”.....	45
Fig 34: Acción “Console Message”.....	45
Fig 35: Acción “Console Get User Input”.....	45
Fig 36: Diagrama de procesos para desinstalar el sistema.....	55
Fig 37: Diagrama de Clases del análisis para el CU_ Desinstalar SGDID Ecumene Pyxel.....	55
Fig 38: Diagrama de Colaboración para el CU_ Desinstalar SGDID Ecumene Pyxel.....	56

Índice de Tablas

Tabla 1: Requisitos Funcionales del SGDID Ecumene Pyxel.....25
Tabla 2: Requisitos no Funcionales del SGDID Ecumene Pyxel.....26
Tabla 3: No conformidades del sistema.....47
Tabla 4: Especificación de caso de uso: CU_ Desinstalar SGDID Ecumene Pyxel.....58

Introducción

En la historia, la humanidad ha sido testigo de grandes y sorprendentes avances tecnológicos que reflejan con hechos y resultados la importancia del desarrollo de las tecnologías. Un factor relevante en este desarrollo de tecnologías es el software, el cual ha facilitado y agilizado la automatización de varios procesos que eran desarrollados por el hombre de forma manual. Además, de que el uso de estas herramientas tecnológicas proporcionan soporte al proceso de toma de decisiones, permitiendo a las empresas obtener diferentes ventajas competitivas, repercutiendo de manera notable en el rendimiento general de la organización [1].

El software desarrolla funciones específicas dentro de una diversa gama de aplicaciones que necesitan las empresas y a pesar de esto a veces no se percibe como activo dentro de un negocio. Pero eso es una concepción errónea, ya que el software ha logrado que las empresas sean capaces de variar diversas esferas de la sociedad, entre las que se pueden ver, la electrónica de consumo, el comercio, la informática, la publicidad y los medios de comunicación. Sin este el hombre no sería capaz de manejar los procesos industriales, no tendría las facilidades de comunicación actuales, por solo mencionar algunos ejemplos de la importancia que tiene actualmente el software.

Para lograr un funcionamiento eficiente del software es necesario que se realice un correcto proceso de instalación. Este proceso es el encargado de habilitar los diferentes programas para su posterior utilización. En caso de realizarse incorrectamente la instalación, la aplicación corre el riesgo de no funcionar. Diversas son las formas de instalar un software, producto de la amplia variedad de sistemas operativos que se han ido desarrollando, pero sin importar cuán diferentes puedan ser las vías de instalación siempre se ha tenido una meta fija, que este proceso de instalación sea de fácil comprensión para los usuarios.

El software ha evolucionado con el tiempo, ya no son aquellas pequeñas aplicaciones que solo eran consideradas como valor agregado para el hardware, ahora son sistemas que pueden manejar grandes cantidades de información. Producto a esta evolución, fue precisa la migración hacia nuevas técnicas que pudieran ser capaces de poner en funcionamiento los programas con la menor complejidad posible, además de lograr una mayor facilidad para los usuarios.

Actualmente, en Cuba, el desarrollo del software se lleva a cabo en varias instituciones, una de ellas es la Universidad de las Ciencias Informáticas, donde se ponen en marcha diversos proyectos de desarrollo de estos productos, tanto para exportar como para elevar el nivel de informatización del país. En el año 2012

se desarrolló el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0. Este sistema permite la identificación, catalogación, recuperación y publicación de los recursos fotográficos, así como generar automáticamente diferentes perfiles a partir de la imagen original [2]. Ecumene Pyxel está conformado por un núcleo desarrollado en el Administrador de Contenidos Empresariales Alfresco¹ y una interfaz web. Para desplegar el sistema es necesario instalar y configurar cada una de sus partes por separado lo que trae las siguientes consecuencias:

- El proceso de instalación requiere de un elevado número de configuraciones en varios ficheros.
- Necesidad de conocimientos técnicos para poder realizar todas las configuraciones en los archivos. Evitando así la ocurrencia de errores producto del desconocimiento o por haber olvidado algún paso durante la instalación.
- Se necesitan conocimientos del Sistema Operativo GNU/Linux en el que se va a desplegar Ecumene Pyxel para instalar todas las dependencias con las que cuenta.

Todo lo antes planteado puede repercutir en el desinterés y rechazo por parte de los clientes a la hora de utilizar este producto, lo que permite plantear como **problema de la investigación**: ¿Cómo facilitar el proceso de despliegue del Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0, de manera que la instalación del sistema se realice con mayor facilidad y rapidez?

Para enmarcar los límites de esta investigación se define como **objeto de estudio** el proceso de instalación de los sistemas de software, delimitando como **campo de acción** el proceso de instalación de los sistemas de gestión documental sobre plataformas GNU/Linux.

A lo largo de toda la investigación se perseguirá como **objetivo general** desarrollar una aplicación para lograr la facilidad y rapidez del despliegue del Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0 sobre plataformas GNU/Linux.

En la presente investigación se defiende la idea que el desarrollo de un instalador para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel permitirá a los usuarios instalar con mayor facilidad el mismo en distribuciones GNU/Linux sin requerir grandes conocimientos sobre el sistema operativo.

En este trabajo se trazan las siguientes **tareas de investigación**:

- Caracterización de las tendencias de los instaladores en los sistemas de gestión documental.

¹ **ECM Alfresco**: <http://www.alfresco.com>

- Identificación de los requerimientos funcionales y no funcionales del instalador para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0.
- Selección de las tecnologías, herramientas, patrones y metodologías necesarias para el desarrollo del instalador para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0.
- Elaboración de la arquitectura de información del instalador para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0.
- Confección del diseño del instalador para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0.
- Implementación de las funcionalidades del instalador para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0.
- Aplicación de las pruebas de funcionalidad y portabilidad al instalador del Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0.

Dentro de los **métodos de investigación** se hizo uso del **método teórico** utilizando:

- **Método histórico-lógico:** Para el estudio de los antecedentes del funcionamiento del proceso de despliegue del Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel y así poder documentar los elementos y procesos adecuados que garantizan la creación del instalador. Además, de realizar un estudio acerca del surgimiento de las diferentes vías de instalación.
- **Análítico-sintético:** Para el análisis e identificación de la arquitectura, tecnologías y documentación necesaria, permitiendo obtener el conocimiento esencial para generar una propuesta de solución adecuada a las exigencias del proceso de instalación del Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel.

También se hizo uso del **método empírico** y de este se utilizó:

- **Observación:** Para estudiar otros trabajos que estén relacionados con los sistemas generadores de instaladores o que tienen elementos en común con la investigación. Además, de aplicar la modalidad Observación participante para entrar en contacto directo con el despliegue de la aplicación.

El presente trabajo de diploma se divide en tres capítulos, a continuación se presentan sus nombres y objetivos de forma general:

- **Capítulo 1:** “Fundamentación teórica que sustenta la investigación para la realización del Instalador de Ecumene Pyxel”. El objetivo del capítulo es analizar los sistemas desarrolladores de instaladores de software existentes en la actualidad y describir las principales herramientas y tecnologías a utilizar para el desarrollo.
- **Capítulo 2:** “Análisis y Diseño del Instalador de Ecumene Pyxel”. En este capítulo se define el modelo de procesos correspondiente al instalador. Se especifican los requisitos que debe tener el sistema, así como los diferentes casos de uso con la descripción de cada uno de ellos. Se realiza la presentación de los diagramas que intervienen en las etapas de análisis y diseño.
- **Capítulo 3:** “Implementación y Prueba del Instalador de Ecumene Pyxel”. En este capítulo se definen los diagramas que intervienen en la etapa de implementación. Se muestran las implementaciones más relevantes dentro del código fuente de la aplicación y se realizará la validación del sistema mediante la aplicación de pruebas funcionales y de portabilidad.

Capítulo 1: Fundamentación teórica que sustenta la investigación para la realización del Instalador de Ecumene Pyxel.

Introducción

Este capítulo se centra en el análisis de los instaladores de software para poder obtener la vía más óptima en el desarrollo del instalador para Ecumene Pyxel. Comenzará el capítulo aportando una definición acerca de los instaladores de software y luego continuará haciendo una breve reseña sobre el surgimiento de las diferentes vías de instalación de software. Seguidamente se pasará a analizar qué elementos se deben tener en cuenta en el proceso de instalación. Se darán a conocer algunos de los sistemas diseñadores de instaladores analizando sus diferentes características y así poder seleccionar el más apropiado para el desarrollo del instalador de Ecumene Pyxel. Para concluir se hace una descripción de las diferentes tecnologías a emplear durante la investigación y se define la metodología de desarrollo a utilizar.

1.1 Instaladores de Software

La instalación de software es el proceso mediante el cual el software queda en condiciones de ser utilizado producto de que las nuevas aplicaciones que se instalarán son transferidas al ordenador con la finalidad de ser configuradas y preparadas para ser ejecutadas. Este proceso se realiza actualmente con la ayuda de programas instaladores de software que vuelven invisibles muchas de las acciones para los usuarios. Estos instaladores son archivos que contienen las instrucciones específicas para que un programa cumpla una determinada función en una Computadora Personal (PC) [3].

1.2. Proceso de instalación

Para que un software pueda ser instalado con la calidad requerida se establecen una serie de requisitos que se deben tener en cuenta. Dentro de estos requerimientos se pueden encontrar:

- Verificar que el sistema destino cumpla con los recursos mínimos de hardware de la aplicación. De esta forma la instalación del programa no terminará en un fracaso por falta de recursos.
- Verificar que la aplicación que se está instalando corresponde a la plataforma de sistema operativo y tener en cuenta que algunos programas informáticos dependen de otros software para su funcionamiento. Para esto se comprueba si se precisan de otras herramientas que forman parte de las dependencias del sistema que se desea instalar [4].
- Indicar las direcciones para el despliegue de los archivos del software que se está instalando. Este

proceso se puede realizar de dos formas, una selección estándar para el usuario de poca experiencia y otra opción de configuración avanzada para los usuarios que estén más capacitados.

- Realizar las diferentes configuraciones con que cuenta la aplicación, dependiendo de sus características particulares [3], para esto se debe brindar una interfaz que solicite los datos necesarios para la configuración de los componentes a instalar. En estas vistas los usuarios introducirán los datos necesarios con el formato indicado y de esta forma el sistema puede trabajar correctamente.

1.3. Surgimiento de las vías de instalación de software

El desarrollo del proceso de instalación del software se ve estrechamente ligado a la evolución de los sistemas informáticos que, con el paso de los años, realizan operaciones más complejas pero a la vez tienen una gran facilidad de uso. En los inicios de la evolución del software este se contemplaba como un valor agregado para el hardware de las computadoras, estos pequeños programas eran diseñados y ejecutados por la misma persona u organización, siendo solamente utilizados por los científicos e investigadores. Una vez que la evolución de las computadoras llegó a su segunda generación el avance de estos sistemas de cómputo permitió que se reconociera el software como producto. Además, el desarrollo de las nuevas tecnologías posibilitó que el software se distribuyera en grandes computadoras y minicomputadoras llegando a miles de usuarios. Desde ese momento comenzó una lucha por desarrollar el mejor paquete de software. Pero todos estos programas necesitaban ser corregidos cuando se detectaban fallos; modificados cuando cambiaban los requisitos del usuario y adaptados a dispositivos de hardware para que funcionaran. Todas estas actividades eran denominadas como mantenimiento de software.

Después de la tercera generación y hasta la actualidad, con el surgimiento de las computadoras personales, crecieron notablemente muchas compañías de desarrollo de software. Estas empresas dedicaron sus esfuerzos para que las personas pudieran manejar los artefactos de cómputo por sí solas, trayendo consigo la creación de simples y avanzados sistemas de soporte técnico, surgiendo así el sistema de instalación.

Este sistema de instalación es el proceso mediante el cual debe pasar el software para que pueda ser utilizado posteriormente. Este proceso se desarrolla actualmente con la ayuda de asistentes o programas instaladores sin que el usuario deba realizar ninguna operación especial, más que decidir el uso del espacio de discos, y otras preferencias particulares de cada programa que va a ser instalado.

Los ordenadores sin el software no serían capaces de realizar ninguna operación, por eso es fundamental para los usuarios la instalación de programas que se ajusten a las necesidades de cada persona. Este proceso de instalación para estos productos fue variando a medida que aparecieron los sistemas operativos.

En las distintas versiones del sistema operativo MS-DOS se utilizaban los archivos ejecutables “.com” que almacenaban en forma directa y lineal la imagen de memoria que sería un programa, solamente se podían almacenar aquellos programas que cabían en un solo segmento. Este formato es uno de los más sencillos puesto que no contenía metadatos, solo código y datos. Pero esta simpleza ocasionó que a partir del surgimiento de los procesadores de 16 bits el formato cayera en desuso. Con el surgimiento de la versión 2.0 de MS-DOS fue introducido en marzo de 1983 el formato de archivos “.exe” que a diferencia del anterior no restringía el tamaño del ejecutable y con el paso de los años se ajustaría a las necesidades tanto del software como del hardware. En la actualidad este formato es utilizado en Windows y DOS y puede ser ejecutado con tan solo dar doble clic en el fichero [5].

En el año 1991 es creado el núcleo para el sistema operativo GNU/Linux. En este nuevo sistema los programas que van a ser instalados son conjuntos de paquetes, por lo que al instalar una aplicación lo que se hace en realidad es instalar paquetes. Esto puede parecer una desventaja en un principio, pero el sistema de paquetería le confiere mucha potencia y mantenimiento a estos sistemas. Existen diferentes métodos para instalar aplicaciones en GNU/Linux entre los que se pueden encontrar:

- **La compilación de los programas desde la consola:** Este fue el primer método para instalar programas en GNU/Linux. Era preciso la descarga del fichero “*tar.gz*”² y los usuarios tenían que compilar cada programa que querían usar en sus sistemas. Pero con el surgimiento de nuevas distribuciones fue necesaria la inclusión de un método para manejar los paquetes instalados en las máquinas y se le dio el nombre “*dpkg*”³ a este sistema.
- **Los programas apt o aptitude, yum o urpmi en dependencia de la distribución que se esté utilizando:** Estos son gestores de paquetes que se ejecutan en modo terminal. Son muy potentes y permiten añadir y eliminar aplicaciones del sistema entre otras funcionalidades. Ambos programas son muy similares salvo en un detalle, y es que “*aptitude*” recuerda las dependencias que se han aplicado en la instalación de un paquete, al contrario de “*apt*” que con la desinstalación borrará solamente el paquete especificado, pero no las dependencias. Por otra parte, estos

² *tar.gz*: Formato de compresión de ficheros.

³ *dpkg*: Es una herramienta para instalación, construcción, eliminación y manejo de paquetes.

programas son empleados por gestores de paquetes gráficos, tales como Synaptic o YaST en dependencia de la distribución en que se esté trabajando. A través de ellos también se puede instalar, desinstalar, actualizar y resolver sus dependencias, desde los repositorios donde se centraliza el software que existe para cada distribución de GNU/Linux.

De esta forma es como han ido evolucionando las vías de instalación a través de los años desde el surgimiento de las grandes computadoras hasta la gran variedad de sistemas operativos, que actualmente existen.

1.4 Estudio de sistemas homólogos.

1.4.1 Instalador de Smart Keeper.

En la Universidad de las Ciencias Informáticas desde el año 2005 se desarrolla el filtro de contenido Smart Keeper, el cual basa su filtrado en reglas o políticas establecidas para la navegación de los usuarios [6]. Este sistema cuenta con un instalador que se realizó utilizando el lenguaje de Programación Bash. Toda la configuración de este instalador se realiza en forma de diálogos de consola, el mismo permite ser instalado tanto en servidores sin interfaz gráfica como en entornos de escritorios. La utilización de diálogos para la realización de la configuración permite que el sistema pueda ser empaquetado en otros instaladores (ejemplo: *rpm*), sin necesidad de realizar nuevo diálogos de configuración. Este instalador detecta parámetros de configuración necesarios y los muestra al usuario permitiéndole modificarlos. En algunos casos hay paquetes del instalador que poseen configuraciones comunes, provocando que al instalar algún paquete nuevo este reconozca toda la configuración del paquete ya instalado y configurado previamente. El instalador valida cada uno de los campos entrados por el usuario, evitando la entrada de datos erróneos. Se encuentra empaquetado en *.deb*, pudiéndose instalar en cualquier distribución basada en Debian. Además, está empaquetado en varios paquetes permitiendo instalar el sistema de forma distribuida, posee un metapquete⁴ que permite instalar todo el sistema en un solo servidor. El instalador final se despliega en un ISO, con un repositorio, las dependencias necesarias para su instalación y un manual de instalación. Una de las desventajas que tiene es que los cuadros de diálogos no poseen una ayuda de cada una de las configuraciones, esto puede provocar que un usuario con pocos conocimientos de Informática pueda cometer algún error. Este instalador se está utilizando en cada uno de los despliegues realizados de Smart Keeper.

⁴ **Metapquete:** Es un paquete que contiene referencias a varios paquetes pero no realiza ninguna funcionalidad. Sirve para instalar fácilmente varios paquetes con una sola orden.

1.4.2 Instalador de MOCICE

El Motor de Clasificación Inteligente de Contenido de Correo Electrónico (MOCICE) permite la clasificación de contenidos de los correos electrónicos a partir de algoritmos de inteligencia artificial [6]. Este sistema posee un instalador para poner en funcionamiento dicha aplicación. El instalador está desarrollado con el lenguaje Python. El funcionamiento de la aplicación está centrado en la instalación de los paquetes que necesita el sistema y de los que forman parte de la aplicación. En el caso de los paquetes que no forman parte de la aplicación, el procedimiento a seguir por parte del instalador es descargar cada uno de esos paquetes del repositorio de la distribución e incorporarlos a un repositorio de paquetes propios del sistema. Si alguno de los paquetes no se encuentra en el repositorio institucional se deberán colocar esos paquetes en un directorio para que puedan ser cargados en el repositorio de MOCICE. Este instalador tiene dos vías de instalación las cuales son:

- **La instalación por separada de todos los paquetes:** Esta forma de instalación es muy útil para instalar los paquetes en varias máquinas, la única condición que se implanta para instalar por esta vía es que los paquetes instalados en otras estaciones de trabajo deberán ser compartidos para que puedan ser utilizados en la aplicación. Además, en este proceso las configuraciones se realizarán manualmente.
- **La instalación mediante un paquete genérico:** Esta vía es muy utilizada en los casos donde se instale en una sola máquina producto a que el paquete de instalación contendrá todo el sistema de paquetería necesario y las configuraciones por defecto.

El instalador de MOCICE a pesar de ser muy eficiente en el manejo de paquetes tiene la desventaja de que no tiene una interfaz gráfica que facilite el trabajo de los usuarios. Por otra parte, se encuentra completamente dependiente de un usuario que tenga conocimientos técnicos sobre las distribuciones GNU/Linux para poder realizar todo el proceso de instalación de paquetes y configuración de manera correcta. Por las razones antes mencionadas se puede determinar que el instalador del sistema MOCICE no se ajusta a las necesidades que exige el instalador que se desea desarrollar.

1.5 Sistemas generadores de instaladores

Para la creación de los instaladores se hace necesario el uso de herramientas generadoras de instaladores que posibiliten su creación de manera sencilla, rápida y que se ajuste a los requerimientos planteados por el usuario. Siempre ajustándose al software que necesita que se le realice el instalador, producto de que en dependencia de la aplicación, se realizará el manejo de archivos de configuración, el

tratamiento en cuanto a las opciones de instalación de ciertos módulos así como la desinstalación del sistema. Estas aplicaciones brindan también la posibilidad de cubrir las necesidades básicas para un sistema multilinguaje y convertirlo en multiplataforma en caso de ser necesario.

1.5.1 BitRock InstallBuilder 8

Es una herramienta de desarrollo para crear instaladores multiplataforma para el software de escritorio y servidor. Con él se pueden crear rápidamente instaladores profesionales [7]. *BitRock InstallBuilder* no es particularmente adecuado para los principiantes debido a la elevada curva de aprendizaje de la herramienta [8]. Algunas de las características que posee esta herramienta son:

- **Soporte multiplataforma:** Los instaladores desarrollados son binarios nativos que se pueden ejecutar en Windows, Mac OS X, FreeBSD, OpenBSD, Solaris, AIX, HP-UX, IRIX, y GNU/Linux. Para la construcción del soporte multiplataforma la herramienta de creación del instalador puede ejecutarse en los sistemas antes mencionados y generar instaladores para todas las plataformas de destino de un archivo de proyecto único. Brinda la facilidad de crear todos los instaladores desde un solo entorno de construcción [9].
- **Multilinguaje:** Los instaladores tienen una variedad de idiomas de instalación, incluyendo Inglés, alemán, japonés, español, italiano, francés, portugués y chino. Se puede especificar un idioma predeterminado o el usuario puede tomar la decisión de elegir uno durante el proceso de instalación para comprender mejor los textos [9].
- **Integración de escritorio:** Los instaladores desarrollados con esta herramienta proporcionan apariencia nativa y la integración de escritorio para Windows, KDE y Gnome [9].
- **Generación para RPM y DEB:** Se pueden generar paquetes RPM y DEB que se instalan utilizando herramientas nativas de administración de paquetes [9].
- **Debugger:** Un debugger hace que sea fácil de identificar y corregir los problemas que se presenten durante el proceso de desarrollo. Permite ver y editar las variables de instalación en tiempo de ejecución. Posibilita la ejecución de pruebas en determinadas partes de la instalación que se encuentren funcionando correctamente. También contribuye a la recuperación de errores inesperados durante el proceso de instalación [9].
- **Múltiples modos de instalación:** Esta característica proporciona varios modos de interfaz gráfica de usuario para la instalación, en una variedad de entornos de escritorio. Presenta un modo de

instalación basado en texto para instalaciones basadas en consola y un modo de instalación silenciosa que puede ser usado para la integración en *script* de *Shell* para el despliegue automatizado [9].

- **Lenguaje y Plataforma Independiente:** Se pueden instalar aplicaciones escritas en cualquier lenguaje, incluyendo: Java, PHP, Perl, Python, Ruby, C / C ++ y .NET / Mono [9].
- **Fácil de usar:** Posibilita un fácil uso de su entorno gráfico de desarrollo. Se pueden diseñar, construir y probar instaladores de manera sencilla [9].
- **Opciones avanzadas de configuración:** Brinda la posibilidad de solicitar al usuario múltiples entradas en una sola pantalla para agilizar el proceso de instalación [9].
- **Funciones de desinstalar:** Un programa de desinstalación se crea como parte de cada instalación, lo que permite a los usuarios desinstalar fácilmente el software. Al igual que con el instalador, se puede ejecutar en una variedad de modos [9].
- **Licencia:** Una desventaja de este software es que es privativo, pero el mismo posee una versión gratuita de prueba que dispone de todas las funciones y opciones necesarias a la creación de paquetes de instalación.

1.5.3 InstallAnywhere 2012

Esta herramienta es de fácil uso, incluso para los desarrolladores novatos que crean software de instalación profesional para prácticamente cualquier plataforma, tanto en entornos físicos como virtuales. Está disponible en dos ediciones: empresarial y estándar. También soporta los idiomas, inglés y japonés [10].

- **Plataforma de Apoyo ampliada:** Soporta las últimas plataformas y actualizaciones los productores de aplicación. Dentro de las principales plataformas actuales incluye Windows 8, Windows Server 2012, Mac OS X 10.7, SUSE Linux 11.3, Ubuntu 12.04, Solaris 11 Sparc, 11i HP-UX, AIX 7.1, IBM i5/OS, Z / OS, RHEL 6 y Fedora 16 [11]
- **Soporte de pruebas automatizadas:** Ahorra tiempo de pruebas de control de calidad y fiabilidad de la instalación, ejecutando código Java para verificar automáticamente la calidad de sus instaladores [11].
- **Variedad de métodos de instalación:** Proporciona la capacidad de ejecutar las instalaciones en

consola en modo silencio que solamente es empleado en servidores sin interfaz gráfica, o en el modo gráfico [11].

- **Fácil de aprender y usar:** Proporciona una interfaz intuitiva de desarrollo integrado, que le permite construir instaladores de manera rápida y eficiente. Crea una instalación básica con el asistente de proyectos en seis pasos. También puede utilizar una interfaz gráfica avanzada, para crear instalaciones personalizadas que controlan todos los aspectos del proceso de implementación [11].
- **Aprovechamiento de los repositorios de paquetes de Linux:** Esta opción es para añadir componentes de aplicación adicionales a través de la web o de repositorios. Muchas aplicaciones requieren de componentes de terceros que están disponibles a través de los repositorios de paquetes de GNU/Linux. Puede conectarse a cualquiera de los repositorios de paquetes de GNU/Linux, ya sea por medio de la web o local, para agregar requisitos comunes que requiera la aplicación [11].
- **Compatibilidad con 64 bits:** El soporte de 64 bits le permite instalar instalaciones de 32 bits a 32 y 64-bits. Proporciona una mayor flexibilidad para las aplicaciones de 64 bits [11].

1.5.4 InstallJammer 1.2.15

Es un generador de instaladores con una interfaz de usuario multiplataforma y funciona en Windows y la mayoría de todas las versiones de UNIX. Tiene un constructor de instalaciones muy poderoso que admite múltiples temas y un nivel alto de configuraciones. Los instaladores están contruidos como ficheros ejecutables separados, que se pueden distribuir fácilmente por la web y dirigir la instalación de todo lo que necesite de manera sencilla [12]. Algunas de las características de esta aplicación son:

- **Desarrollo rápido:** Incluye un poderoso constructor de instaladores, el cual permite que el usuario tome el control de todo lo que acontece en el programa que se va a crear. Se puede obtener un instalador corriendo en pocos minutos [12].
- **Personalización:** Brinda la posibilidad de configurar el instalador hasta el más mínimo detalle en cuanto a la configuración de la interfaz gráfica y de las distintas funcionalidades [12].
- **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 entre otras [12].
- **Soporte Multi-lenguaje:** Esta aplicación soporta múltiples lenguajes en los instaladores los cuales

son agregados de forma sencilla [12].

- **Instaladores rápidos:** Los instaladores creados con esta herramienta suelen ser rápidos y ligeros. El arranque es muy rápido puesto que no se necesita de espacio adicional del disco para ficheros temporales antes de iniciar [12].
- **Simple archivos de proyecto:** Los ficheros de un proyecto son muy simples, no son más que 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 [12].
- **Opciones para múltiples interfaces de usuario:** Los instaladores creados con esta herramienta pueden ejecutarse en modo consola (instalación silenciosa) o en modo de interfaz de usuario dependiendo de las necesidades del cliente. Por defecto acepta todas las configuraciones diseñadas para una instalación con GUI⁵. Por su parte el 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 [12].
- **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 construida, ejecutando *InstallJammer* en otra plataforma [12].
- **Control de temas:** Permite la selección de su apariencia a partir de diferentes temas para futuros instaladores, solo se precisa de la inclusión de los temas comerciales más populares. Siempre es posible la creación de nuevos temas [12].
- **Lenguaje propio de programación:** Con un lenguaje propio es posible la modificación y creación de un instalador. Este lenguaje es denominado Tcl/Tk. El Tcl es un lenguaje de programación dinámico muy potente, pero fácil de aprender, adecuado para una amplia gama de usos, incluyendo aplicaciones web y de escritorio. De código abierto y accesible para las empresas, Tcl es un lenguaje maduro que se encuentra todavía en evolución y es multiplataforma, fácil de implementar y altamente extensible. Por su parte Tk es un conjunto de herramientas de interfaz gráfica de usuario que tiene el desarrollo de aplicaciones de escritorio a un nivel más alto que los

⁵ **GUI:** Interfaz Gráfica de Usuario del inglés *Graphical User Interface*.

enfoques convencionales. Tk es la interfaz gráfica de usuario estándar no solo para Tcl, también para muchos otros lenguajes dinámicos. Puede producir aplicaciones ricas que se ejecutan sin cambios a través de Windows, Mac OS X y Linux [12].

- **De uso libre, libre de distribuir:** Esta aplicación utiliza la Licencia General Pública (GPL) que pretende garantizar la libertad de compartir y modificar gratuitamente el software [12].
- **Soporte para 32-bits y 64-bits:** Se pueden generar aplicaciones de instalación tanto para una arquitectura de 32-bits como de 64-bits [12].
- **Posibilidad de desinstalación:** Si esta función está activada, se agregará una acción para crear un desinstalador cuando la aplicación se encuentre instalada [12].

Después de un análisis de los sistemas generadores de instaladores, se tomó la decisión de utilizar el software *InstallJammer* el cual se destaca entre las demás aplicaciones por encontrarse bajo la Licencia General Pública (GPL). También se caracteriza por haber sido utilizado para la construcción del instalador del ECM Alfresco. *InstallJammer* está escrito casi en su totalidad en el lenguaje *script* Tcl/Tk con un poco de ayuda de algunas extensiones escritas en C. Este lenguaje *script* es muy útil producto de que brinda la facilidad de programar funcionalidades propias, así como el diseño de la interfaz gráfica. Además, se pueden ejecutar programas externos (ej: *script* y *bash*). Por otra parte, este sistema proporciona una buena documentación que será muy útil para comprender mejor el funcionamiento de esta herramienta y cuenta con un foro donde se pueden encontrar las respuestas a varias de las interrogantes que puedan surgir durante el desarrollo. Dentro de otro de los elementos que repercutieron en la decisión de utilizar el *InstallJammer* se encuentra, la realización de una GUI amigable, entendible y ligera, permitiendo la rápida instalación de todos los componentes y paquetes necesarios para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel.

1.6 Metodología de desarrollo

Se decidió utilizar como metodología de desarrollo OpenUP producto de que el centro de desarrollo al cual corresponde la presente investigación solicitó el uso de dicha metodología. Esta se encuentra organizada en dos dimensiones diferentes pero interrelacionadas: el método y el proceso. En el método es donde se definen sus elementos, que pueden ser (roles, tareas, artefactos y lineamientos). En el proceso, los elementos del método son aplicados de forma ordenada en el tiempo.

OpenUP es una metodología de desarrollo que proporciona un proceso ágil y unificado que tiene un conjunto mínimo de prácticas que ayudan a que el proceso de desarrollo de software se realice de manera

más eficiente. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. También se encuentra organizado dentro de 4 áreas principales de contenido: Comunicación y Colaboración, Intención, Solución y Administración [13].

Se seleccionó esta metodología por una serie de características que son enunciadas a continuación:

- Metodología de desarrollo de software de código abierto diseñado para pequeños equipos organizados, quienes quieren tomar una aproximación ágil del desarrollo.
- Proceso iterativo e incremental que es Mínimo, Completo y Extensible.
- Se valora la colaboración y el aporte de los *stakeholders*⁶ sobre los entregables y las formalidades innecesarias.
- Practicantes de desarrollo de software (desarrolladores, administradores de proyectos, analistas y probadores) trabajan juntos como un equipo de proyecto.

Además de las características es válido destacar que contiene 4 fases de desarrollo las cuales son:

- **Inicio:** Se pretende lograr concurrencia entre todos los *stakeholders* sobre los objetivos del ciclo de vida para el proyecto.
- **Elaboración:** Se establece la línea base de la arquitectura del sistema y proporciona una base estable para el gran esfuerzo de desarrollo de la siguiente fase.
- **Construcción:** Se completa el desarrollo del sistema basado en la arquitectura.
- **Transición:** Asegura que el software está listo para entregarse a los usuarios [13].

1.7 Herramientas

1.7.1 Herramienta Case: Visual Paradigm 8.0 para UML

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. Garantiza una rápida construcción de aplicaciones con una mayor calidad, y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, y generar código desde diagramas.

Es una herramienta multiplataforma distribuida bajo licencia privativa Visual Paradigm Suite. Proporciona un entorno ágil y eficiente para el diseño. Permite el diseño del sistema con el diagrama de clases UML.

⁶ **Stakeholders:** Partes interesadas

También puede soportar la edición simultánea de un mismo proyecto o diagrama [14].

1.7.2 ForeUI

La herramienta ForeUI permite diseñar prototipos de interfaces de aplicaciones mediante la creación de maquetas de imágenes para la visualización de las ideas. La aplicación ofrece un conjunto de componentes *drag and drop*⁷ que incluyen botones, menús, pestañas, barras de desplazamiento, *combobox* y otros elementos comunes, además de herramientas gráficas para insertar formas, líneas, textos e imágenes. Se pueden utilizar estos elementos para diseñar rápidamente una GUI estándar y añadir notas o comentarios al diseño.

ForeUI puede exportar el diseño final en formato de imagen PNG para la visualización de la maqueta. Otras características incluyen soporte para capas, niveles de opacidad y una biblioteca de iconos integrada. Es desarrollado sobre Java, por lo que es multiplataforma. Cuenta con una licencia privativa [15]. Esta herramienta tiene la facilidad de que a pesar de ser privativa se puede descargar libremente la versión de prueba y se puede trabajar con todas sus funciones durante 15 días antes de comprar la licencia. Una vez terminado ese plazo si es necesario comprar la licencia para trabajar con la herramienta. Se recibirá la clave de licencia por correo electrónico después del pago. La licencia le concede una actualización de software al año, y el uso permanente de las versiones publicadas en ese año [15].

Conclusiones del capítulo

La investigación realizada sobre los instaladores demuestra que los mismos están estrechamente ligados a las características particulares de la aplicación que necesita de un instalador. Además, una vez que son desarrollados por una herramienta específica, para realizar alguna modificación es necesario tener el código fuente del instalador, con tener acceso al ejecutable solamente no sería de utilidad para lo que se desea hacer. Por otra parte también se pudo comprender el proceso de instalación con las diferentes características con las que debe contar un instalador. Durante la realización del estudio anterior se lograron definir aspectos como:

- Se determinó que es más factible construir un nuevo instalador que tratar de integrar uno existente que tenga cierta similitud con el que se desea desarrollar.
- Se adquirieron los conocimientos necesarios para desarrollar un instalador con los requerimientos necesarios.

⁷ *drag and drop*: arrastrar y soltar.

- Se definió la metodología a utilizar durante todo el proceso de desarrollo, así como las herramientas adecuadas para emplear tanto en las etapas de análisis y diseño como de implementación.

Capítulo 2: Análisis y Diseño del Instalador de Ecumene Pyxel

Introducción del capítulo

En el presente capítulo se abordarán las características que debe tener la solución propuesta para el instalador del Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel. Se describirán los principales artefactos desarrollados durante las etapas de análisis y diseño del sistema. Estos artefactos serán los que se relacionan con el modelado por procesos y el modelado del sistema. De esta forma quedará especificado todo lo referente con los procesos del sistema, los diferentes requisitos funcionales que darán a conocer las condiciones que el sistema debe cumplir y serán representados mediante los casos de uso con que debe contar la aplicación. Además, se representarán los requisitos no funcionales que mostrarán las características y restricciones que debe tener el sistema para su correcto funcionamiento. Por otra parte, se hará una representación de los diagramas del análisis, donde se verá el flujo de eventos que debe seguir la aplicación. Para culminar con estas etapas se expondrá el modelo de despliegue del sistema.

2.1 Características del Sistema

En la actualidad el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel se instala de forma manual. Para poder instalar esta aplicación es preciso realizar la instalación de manera individual de cada uno de los módulos que integran el SGDID⁸ Ecumene Pyxel. Dentro de los módulos del sistema se tiene el núcleo que se encuentra desarrollado en el ECM Alfresco. Para su puesta en marcha es necesario realizar la instalación de una serie de dependencias así como la configuración de ficheros. Por otra parte, también se encuentra la base de datos, la cual debe ser restaurada para que el núcleo funcione correctamente. Por último pero no menos importante se tiene el módulo de la interfaz, en este también se realizan una serie de modificaciones en ficheros. Debido a toda la serie de pasos que hay que seguir este proceso se vuelve engorroso. Encontrándose el sistema en desventaja en cuanto a su usabilidad, puesto que se requiere de personal calificado para realizar la instalación.

El sistema propuesto no tiene antecedentes, convirtiéndolo en la primera versión del instalador para el SGDID Ecumene Pyxel. Para la realización de esta aplicación se hará uso del sistema generador de instalador *InstallJammer* V1.2.15. El mismo se destaca entre otros sistemas por su facilidad de uso, por ser una herramienta de licencia de código abierto, por la existencia de suficiente documentación y la posibilidad de ejecutar diferentes programas externos que necesite el sistema. Con esta herramienta será

⁸ **SGDID:** Sistema de Gestión Documental de Imágenes Digitales.

posible desarrollar un instalador con una interfaz amigable, que facilite el trabajo para los usuarios. Este instalador posibilitará desplegar el sistema sobre plataformas GNU/Linux de la manera más eficiente posible en cuanto a su rapidez, facilidad de manejo de los componentes del SGDID Ecumene Pyxel y configuraciones pertinentes de la aplicación. La aplicación brindará opciones que se ajusten a las necesidades y conocimientos técnicos del usuario. Esto es posible porque se contará con dos vías de instalación, cada una con sus características particulares haciendo más atractiva la nueva propuesta. También se proporcionará el manejo individual de cada uno de los módulos que integran el sistema.

2.2 Mapa de Procesos

A continuación se hará la presentación de un mapa de procesos que ofrece una visión general del sistema, en la Fig. 1 se representan los procesos que intervienen en la aplicación, así como sus relaciones principales.

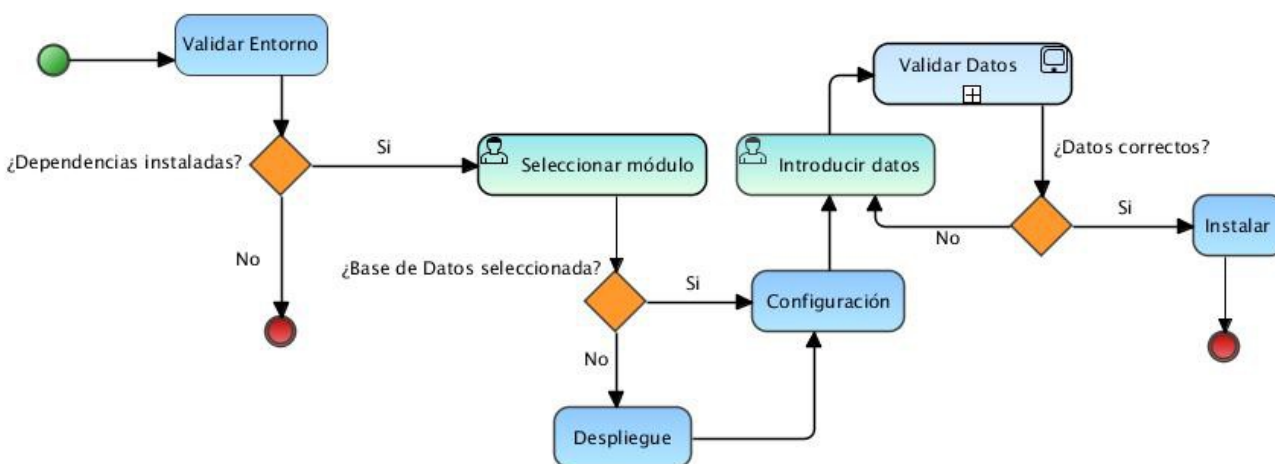


Fig 1: Mapa de procesos

2.3 Modelo Conceptual

Para determinar cuál es el dominio de la aplicación se presentará un Modelo Conceptual. El mismo sirve para ver cuáles son los conceptos más importantes de la aplicación, y las funciones que se deben ejecutar en ella, facilitando de esta forma la comprensión del sistema.

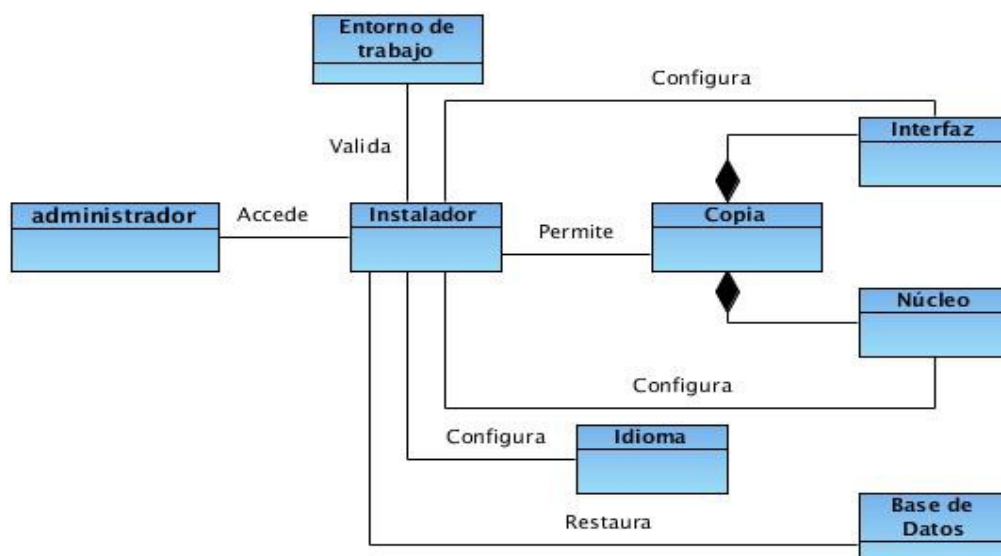


Fig 2: Modelo Conceptual

2.4 Descripción de procesos de negocio.

Los diferentes procesos que intervienen en el instalador para el SGDID Ecumene Pyxel forman parte de una secuencia de acciones que se deben realizar. A continuación se explicarán cada uno de estos procesos:

- **Validar entorno:** Este proceso es realizado por el sistema una vez que se comienza la instalación. El mismo se lleva a cabo para verificar que el sistema tenga todas las dependencias necesarias. Se encuentra ligado a la condición de dependencias instaladas, donde en el caso que estén todas se continúa con el proceso de instalación. En el caso de la ausencia de algunas dependencias se terminará la instalación.
- **Seleccionar módulos:** Este proceso se realizará solo si se está llevando a cabo una instalación de tipo avanzada. En este tipo de instalación se brinda la posibilidad de seleccionar que módulo del SGDID Ecumene Pyxel se desea instalar. Solo se debe marcar el módulo deseado que puede ser el Núcleo, la Interfaz o la Base de Datos. Este proceso se encuentra vinculado a una matriz de decisión encargada de diferenciar los procesos a seguir entre la Base de Datos y los demás módulos. Siendo esto necesario producto de que no tienen la misma secuencia.
- **Despliegue:** Este proceso solo se llevará a cabo cuando se realice la instalación de tipo avanzada y en caso de que sean seleccionados los componentes Interfaz y Núcleo. El presente proceso se

encargará de copiar los ficheros correspondientes al componente a instalar en la ruta a seleccionar.

- **Configuración:** En este proceso no existe distinción de tipos de instalación o de componentes seleccionados. El proceso de configuración es necesario puesto que se exigen datos que deberán ser introducidos por el usuario en dependencia de la máquina en que se encuentre. Algunos de estos elementos son el IP, nombre del usuario, contraseña, puerto de comunicación y nombre de la base de datos. Pero también será utilizado este proceso para configurar el idioma de instalación y activar algunas funcionalidades que se desee tenga incorporadas.
- **Introducir datos:** Este proceso se realiza para que el usuario pueda introducir los datos necesarios y así realizar las configuraciones necesarias que tienen que ver con cada uno de los módulos que se estén instalando.
- **Validar datos:** Este subproceso es realizado internamente por el sistema, para validar que los datos introducidos por el usuario estén correctos. La validación de los datos está vinculada a una condición que determinará si se continúa para el próximo proceso o se regresa a la entrada de datos.
- **Instalar:** En este proceso se mostrará el avance de la instalación mediante una barra de estado. Además, se mostrarán mediante un texto los diferentes ficheros que se están manipulando y las acciones que se están realizando en cada instante. Con este proceso se culmina la instalación y con él se materializan todos los procesos que se realizaron durante la instalación.

2.5 Requisitos del sistema

Para lograr que el desarrollo de un software tenga éxito, es esencial comprender perfectamente los requisitos del software. La captura de los requisitos va a influenciar en todo el proceso de desarrollo del software, repercutiendo en el resto de las fases de desarrollo. Una definición eficiente de los requisitos funcionales y no funcionales, permitirá al equipo de desarrollo reducir los riesgos que se puedan presentar durante la implementación del sistema y se puede determinar lo que el sistema debe hacer. Además, de establecer aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes del mismo [16].

2.5.1. Requisitos Funcionales

	Requisito Funcional	Descripción	Complejidad
1	Instalar el SGDID Ecumene Pyxel en el modo de Interfaz Gráfica	La instalación en modo gráfico propondrá una serie de vistas que facilitarán el trabajo de los usuarios.	Media
1.1	Seleccionar el idioma de instalación	Con la selección del idioma el usuario podrá realizar toda la instalación en el lenguaje que le sea más cómodo entender.	Media
1.2	Seleccionar el tipo de instalación	Se debe de seleccionar un tipo de instalación teniendo en cuenta el nivel de conocimientos técnicos del usuario. Las vías de instalación son: <ul style="list-style-type: none"> • Básica: Se establecerá una instalación predefinida. • Avanzada: Se establecerá una instalación en la cual el usuario decidirá la ruta de instalación de los componentes a instalar. 	Media
1.3	Validar el entorno de trabajo	Se deberá validar el entorno de trabajo verificando que las dependencias estén instaladas. En caso de no estarlo no se podrá continuar con la instalación y aparecerá un mensaje indicando que dependencia falta.	Alta
1.4	Configurar el núcleo	Se configurarán los elementos relacionados con la comunicación con la base de datos. Para ello el sistema pedirá los datos que	Media

		deben ser introducidos para configurar el núcleo.	
1.5	Configurar la interfaz	Se configurarán los elementos relacionados con la interfaz para su correcto funcionamiento. El sistema se encarga de solicitar los diferentes datos que necesita la interfaz para su configuración.	Media
1.6	Configurar la Base de Datos	Se configurarán los elementos referentes a la restauración de la base de datos. El sistema se encarga de obtener mediante una interfaz los datos que se necesitan.	Alta
1.7	Desplegar el sistema	El sistema debe ser desplegado en la máquina en que se está realizando la instalación para esto es necesario solicitarle al usuario que introduzca una dirección válida para cada componente.	Alta
2	Instalar el SGDID Ecumene Pyxel en modo Consola	La instalación con esta modalidad se realizará en todo momento mediante la interacción del usuario con la consola. En este modo no existen interfaces gráficas, solo se indican las instrucciones de lo que está sucediendo y el usuario solamente introduce los datos que se exigen.	Media
2.1	Validar el entorno de trabajo	Se deberá validar el entorno de trabajo verificando que las dependencias estén instaladas. En caso de no estarlo no se podrá continuar con la instalación hasta que no sean instaladas.	Alta

2.2	Configurar el núcleo	Se configurará un fichero del núcleo, modificando los elementos relacionados con la comunicación con la Base de Datos. Los datos serán solicitados mediante mensajes que se muestran en la consola.	Media
2.3	Configurar la interfaz	Se configurarán los elementos relacionados con la interfaz para su correcto funcionamiento. El sistema se encarga de solicitar los datos necesarios mediante mensajes en la consola.	Media
2.4	Configurar la Base de Datos	Se configurará todo lo referente a la creación y restauración de las bases de datos del sistema.	Alta
2.5	Desplegar el sistema	El sistema debe ser desplegado en la máquina en que se está realizando la instalación para poner en funcionamiento los diferentes módulos del sistema.	Alta
3	Desinstalar el SGDID Ecumene Pyxel en modo de Interfaz Gráfica.	Se debe proporcionar un desinstalador en el caso de que no se desee utilizar más la aplicación. Esta desinstalación puede realizarse mediante la interfaz gráfica.	Media
4	Desinstalar el SGDID Ecumene Pyxel en modo Consola	Se brinda un desinstalador que sea en modo Consola para el caso de aquellos servidores que no tengan interfaz gráfica.	Media

Tabla 1: Requisitos Funcionales del SGDID Ecumene Pyxel.

2.5.2 Requisitos No funcionales

Requerimientos de Software
<p>RnFSW 1: El instalador se utilizará en cualquier sistema operativo perteneciente a distribuciones GNU/Linux.</p> <p>RnFSW 2: Se deberán utilizar una serie de aplicaciones que forman parte de las dependencias de Ecumene Pyxel, las cuales son:</p> <ul style="list-style-type: none"> • Imagemagick • Libimage-exiftool-perl <p>RnFSW 3: Se utilizará como Gestor de Base de Datos PostgreSQL V8.4 o superior.</p> <p>RnFSW 4: La máquina virtual de java a emplear es:</p> <ul style="list-style-type: none"> • openjdk6
Requerimientos de Hardware
<p>RnFH 1. Recomendable servidores con un microprocesador Core i3. Memoria RAM de 2 GB y un disco duro de 160 GB.</p>
Restricciones en el diseño y la implementación
<p>RnFRDI 1. Las dependencias del sistema deben ser instalas antes de ejecutar el instalador y así posibilitar que la aplicación funcione correctamente.</p>
Requerimientos de Seguridad
<p>RnFS 1. El instalador solo estará disponible para el administrador del sistema, debido a la necesidad de privilegios de administración para poder instalar la aplicación. Cumpliendo así con los principios de confidencialidad y disponibilidad.</p>
Requerimientos de Usabilidad
<p>RnFU 1. El instalador podrá ser usado tanto por usuarios con pocos conocimientos técnicos como por usuario con un nivel más avanzado.</p>

RnFU 2. Proporcionar facilidad de uso para aquellos usuarios que deseen instalar en otro idioma el sistema ya sea en español o inglés.

RnFU 3. Posibilitar que el usuario elija el tipo de instalación, dependiendo del nivel de conocimientos técnicos que el usuario tenga.

Requerimientos de Ayudas y Documentación en línea.

RnFRA 1. Proporcionar un manual de usuario que contenga cada uno de los pasos a seguir durante el proceso de instalación y desinstalación. Para brindar una guía a los usuarios que no se encuentren relacionados con ciertos procedimientos.

Tabla 2: Requisitos no Funcionales del SGDID Ecumene Pyxel.

2.3 Diagrama de Casos de Uso del Sistema.

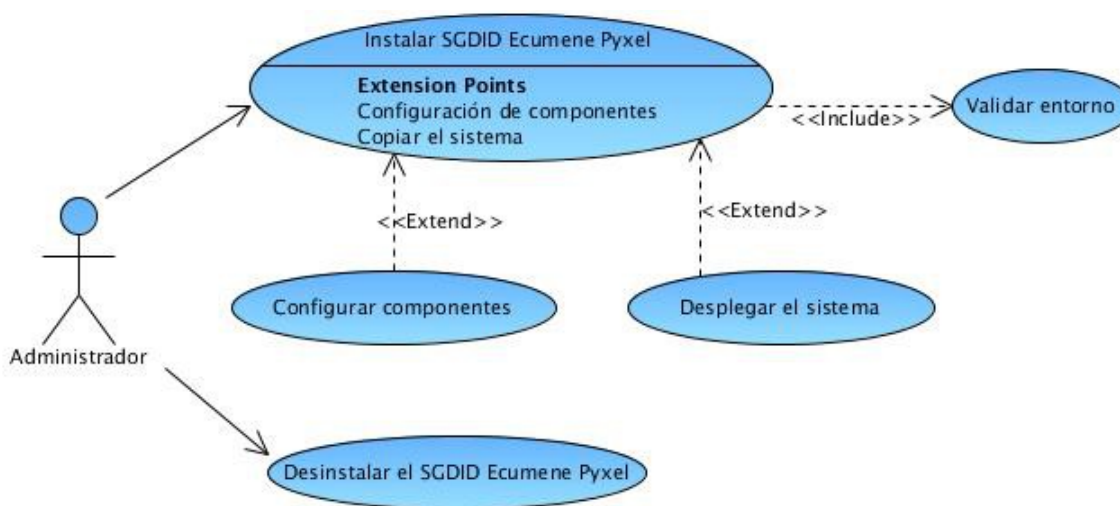


Fig 3: Diagrama de Casos de Uso.

2.4 Especificación de Casos de Uso

La especificación del Caso de Uso es el proceso mediante el cual se detalla de forma descriptiva la historia de los diferentes escenarios por los que debe atravesar el Caso de Uso. La especificación del Caso de Uso Instalar SGDID Ecumene Pyxel se encuentra visible en el documento CIDI_ISGDIDEP_0114_Especificación de casos de uso [17]. En caso de que se desee obtener una mayor información sobre el funcionamiento del proceso de instalación se puede consultar dicho documento.

2.5 Arquitectura del sistema

La Arquitectura de Software consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software en un sistema de información. Es de gran importancia definir la arquitectura de un sistema puesto que ella permite la comunicación entre las partes interesadas en el desarrollo de un sistema [18].

Producto de esto se hace necesario definir una arquitectura que sea capaz de guiar el proceso de desarrollo del instalador. El patrón arquitectónico que se utilizará en la solución propuesta es el Modelo Vista Controlador (MVC).

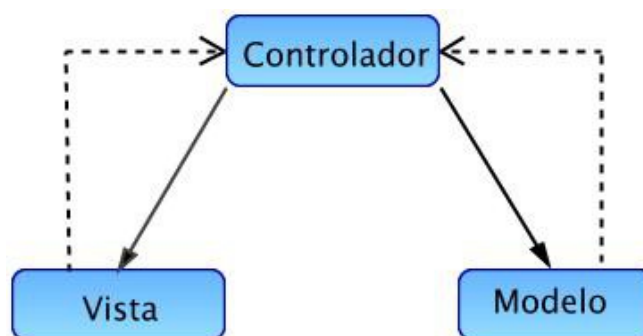


Fig 4: Arquitectura Modelo Vista Controlador (MVC)

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 a trabajar. El modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas [19].
- **Vista:** Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.[20].
- **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 [19].

2.5.1 Adaptación del patrón arquitectónico Modelo Vista Controlador al desarrollo del instalador del Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel.

Todo proceso de desarrollo tiene características particulares, por lo que se debe adaptar la arquitectura candidata para su posterior utilización. Este procedimiento también es necesario realizarlo en la presente

investigación. Las adaptaciones realizadas al patrón arquitectónico Modelo Vista Controlador son:

- **Modelo:** En este nivel se encuentra toda la información con que cuenta el instalador, referido a los módulos Núcleo, la Interfaz y la Base de Datos.
- **Vista:** En este nivel se reflejan cada una de las interfaces con las que cuenta el instalador. Dichas interfaces serán desarrolladas mediante el lenguaje Tk.
- **Controlador:** Se encargará de todos los eventos con los que cuenta el sistema. Estos eventos serán elaborados mediante el lenguaje Tcl y algunas funciones desarrolladas en Bash.

2.6 Diagrama de Clases del Análisis

El diagrama de Clases del Análisis está formado por una clase interfaz que será visualizada por el usuario. Esta clase está relacionada con las clases controladoras que se encargarán de proporcionar las diferentes funcionalidades con las que cuenta el instalador. Estas clases estarán vinculadas a los módulos del instalador. Los módulos son el Núcleo, la Interfaz y la Base de Datos, cada uno de estos componentes serán cargados en el instalador para su previa instalación.

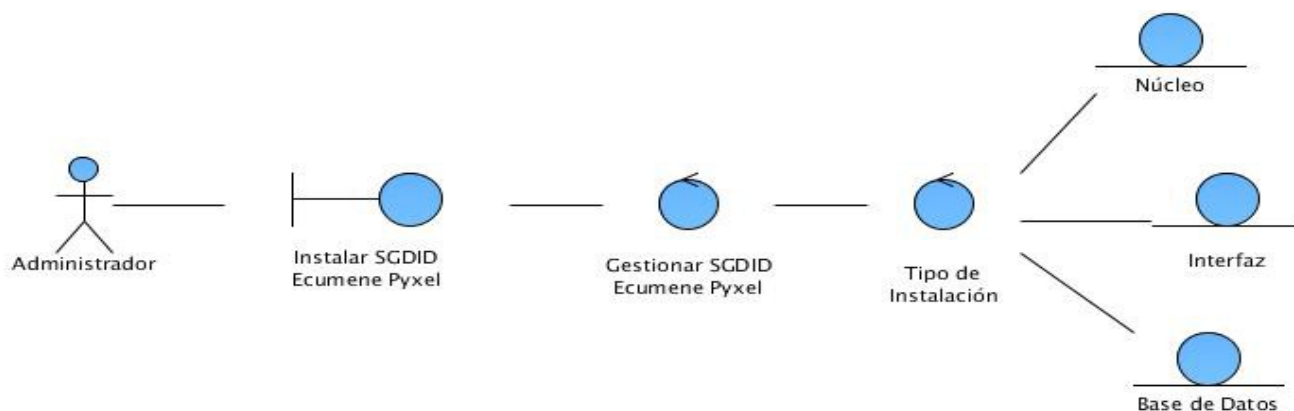


Fig 5: Diagrama de clases del análisis para el CU_Instalar SGDID Ecumene Pyxel

2.7 Diagramas de Interacción

Los Diagramas de Interacción modelan el comportamiento dinámico del sistema y el flujo de control en una operación. También describen la interacción entre objetos, los cuales interactúan a través de mensajes para cumplir ciertas tareas. Existen dos tipos de diagramas de interacción: los Diagramas de Secuencia y los Diagramas de Colaboración [21].

Para una mejor representación y comprensión del sistema en desarrollo se utilizarán los Diagramas de Colaboración para enfatizar en la organización estructural de los objetos que participan en una interacción. En este tipo de diagramas se describe el intercambio de mensajes entre objetos y las relaciones entre ellos. Por otra parte, hay que tener en cuenta que los mensajes se enumeran, indicando el sentido de la información y el lugar donde se originan. También es válido tener en cuenta que mediante estos diagramas se aprecia la forma en que funciona el instalador y de donde parte su flujo de información.

2.7.1 Diagrama de colaboración: CU_Instalar SGDID Ecumene Pyxel -Tipo de instalación típica

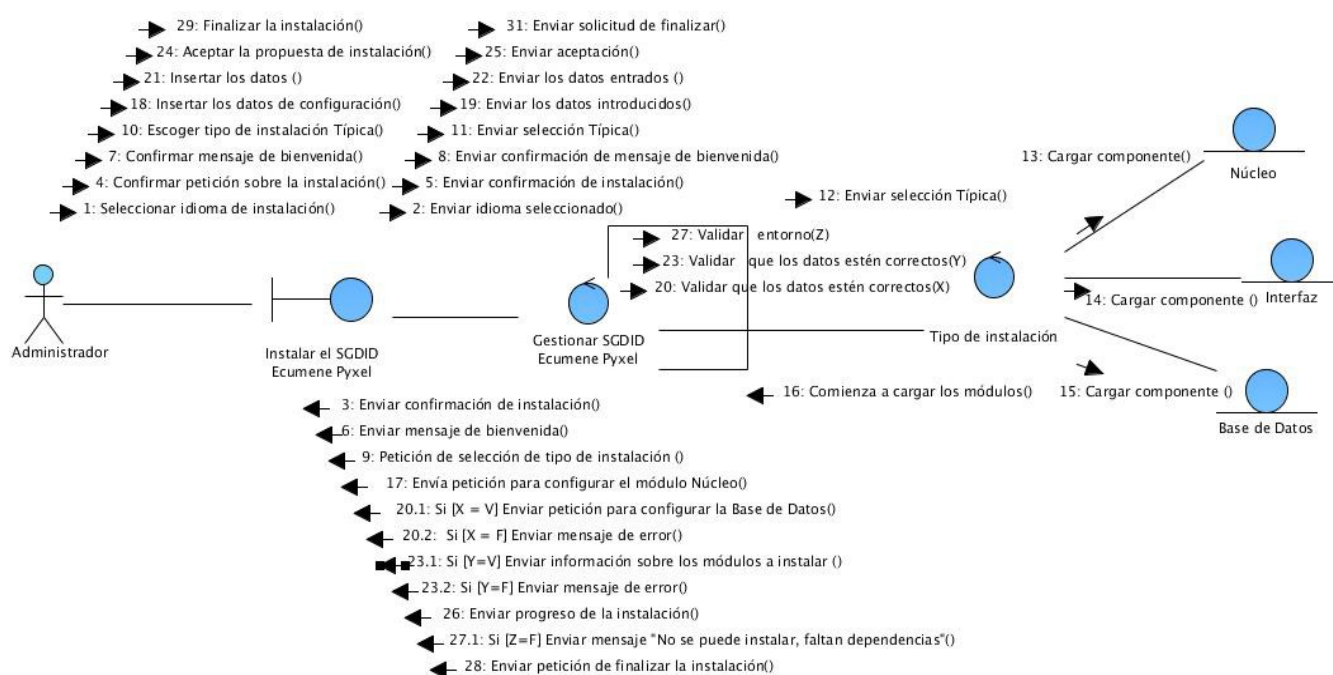


Fig 6: CU_Instalar SGDID Ecumene Pyxel -Tipo de instalación típica

2.7.2 Diagrama de colaboración: CU_Instalar SGDID Ecumene Pyxel -Tipo de instalación avanzada para el módulo Núcleo

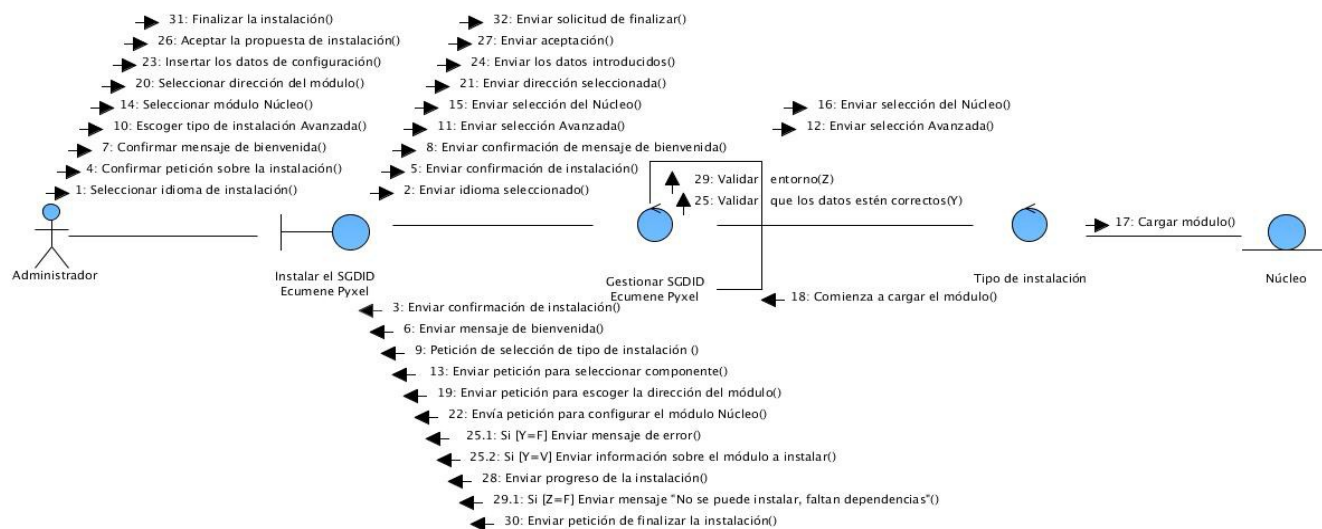


Fig 7: DC: CU_Instalar SGDID Ecumene Pyxel: Tipo de instalación avanzada para el módulo Núcleo

2.7.3 Diagrama de colaboración: CU_Instalar SGDID Ecumene Pyxel -Tipo de instalación avanzada para el módulo Base de Datos

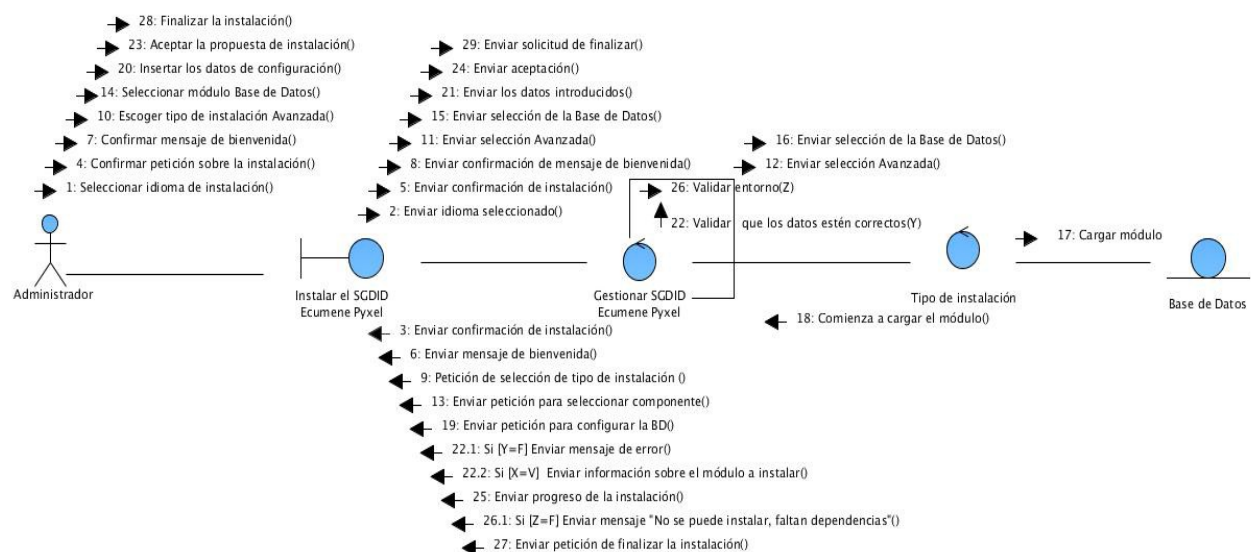


Fig 8: DC: CU_Instalar SGDID Ecumene Pyxel: Tipo de instalación avanzada para el módulo BD

2.7.4 Diagrama de colaboración: CU_Instalar SGDID Ecumene Pyxel -Tipo de instalación avanzada para el módulo Interfaz

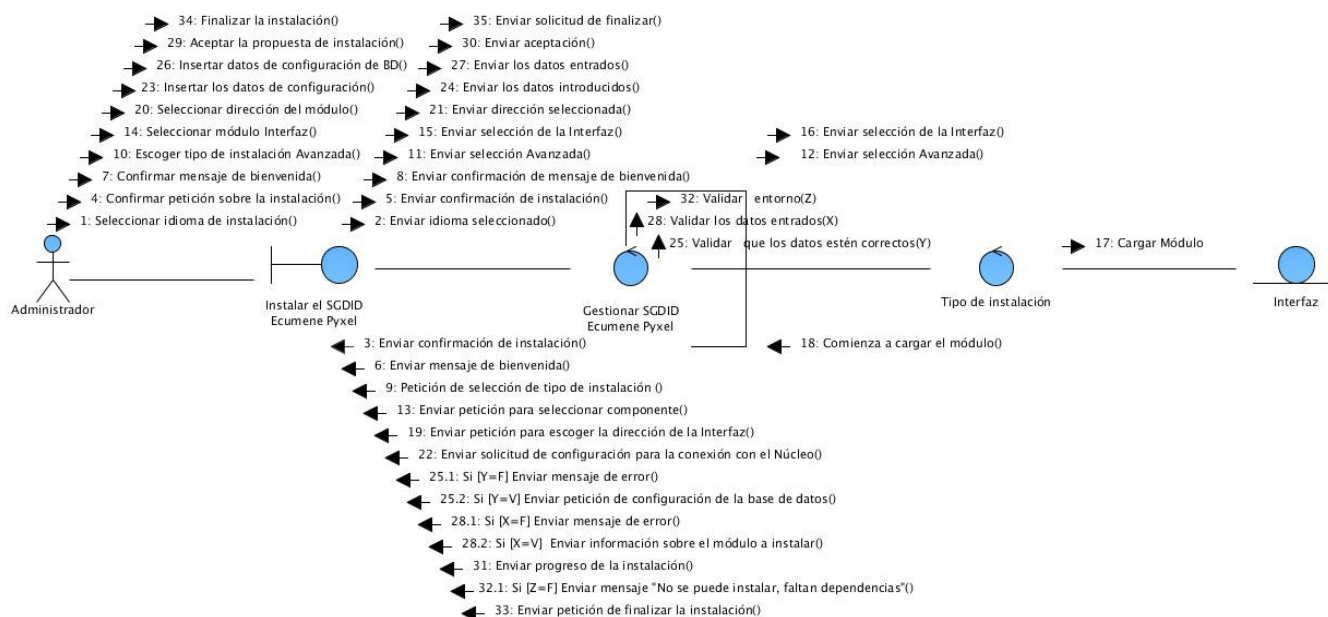


Fig 9: DC: CU_Instalar SGDID Ecumene Pyxel: Tipo de instalación avanzada para el módulo Interfaz.

2.8 Modelo de Despliegue

Mediante un Modelo de Despliegue se pueden mostrar las relaciones físicas entre los componentes de hardware y software para el despliegue del sistema. En la Fig. 10 se refleja este tipo de diagrama, el cual muestra, como la PC cliente contiene el artefacto de instalación del Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel. Pero para el funcionamiento del instalador se debe de tener en cuenta el entorno de ejecución, el cual debe ser una distribución del Sistema Operativo GNU/Linux.

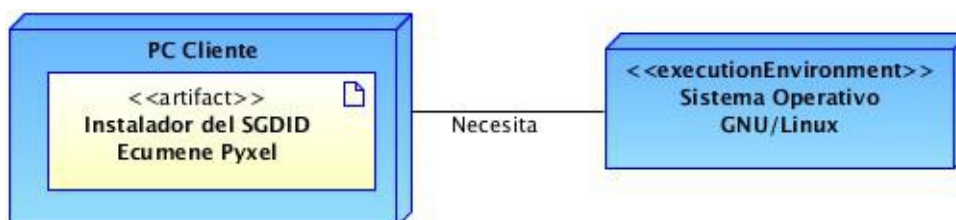


Fig 10: Modelo de Despliegue del instalador para el SGDID Ecumene Pyxel

Conclusiones del Capítulo

Después de desarrolladas las fases de análisis y diseño del instalador para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel se puede decir que para el desarrollo de esta aplicación se han adquirido los conocimientos necesarios para comenzar la etapa de implementación. Durante estas fases se han podido definir diferentes elementos que han formado parte de la concepción del sistema. Algunas de las conclusiones a las que se han podido arribar una vez culminadas estas fases son:

- Se realizó la descripción de la propuesta de solución para obtener un producto que cumpla con las expectativas esperadas facilitando el trabajo de instalación de los usuarios.
- Se realizaron una serie de diagramas que sirven de apoyo visual para poder visualizar de una manera más clara el flujo de eventos que contiene el instalador. También se realizó una representación de lo que necesita el instalador para ser desplegado.
- Se estableció durante el análisis del sistema cuales son los procesos que rigen la instalación de la aplicación. Además, se definieron cada uno de los requisitos con que cuenta el sistema para de esta forma llegar a una mejor comprensión de lo que el instalador debe proporcionar y lo que debe tener.
- Se definió la arquitectura del sistema a utilizar la cual servirá de guía y de esta forma trabajar con una misma línea de desarrollo.

Capítulo 3: “Implementación y Prueba del Instalador de Ecumene Pyxel”.

Introducción del capítulo

En el presente capítulo se expone como se ha implementado la herramienta de instalación propuesta. Se presenta una descripción sobre los elementos que se han construido, describiendo la forma en que se utilizó la herramienta *InstallJammer* durante el proceso de construcción. Finalmente se muestran los resultados obtenidos en la realización de las pruebas que se aplicaron al sistema.

3.1 Diagrama de Componentes

Mediante el Diagrama de Componentes se muestra la relación entre los componentes del software, sus dependencias, comunicaciones y localización. Estos diagramas son usados para estructurar los componentes en los sistemas del software. Ellos examinan y controlan las dependencias entre componentes o interfaces. Un componente representa una parte modular, desplegable y reutilizable de un sistema [18].

En el siguiente diagrama se detalla la interacción de varios componentes, para que de esta forma puedan funcionar como un todo. El sistema parte del ejecutable del SGDID Ecumene Pyxel el cual está encargado de la instalación del sistema. Este componente se encuentra vinculado a los módulos del sistema que se dividen en: Base de Datos, Núcleo, Interfaz. Cada uno de estos componentes dependen de otros paquetes para poder ser instalados. El componente base de datos tiene como dependencia el Sistema Gestor de Base de Datos *Postgres*, el Núcleo depende de los paquete *Imagemagik*, *Libimage-exiftool-perl* y de la máquina virtual de *Java openjdk6*, por su parte la Interfaz depende del Servidor Web *Apache* y del lenguaje de programación del lado del servidor *PHP*. Los módulos del sistema se encuentran conectados a la API⁹ *InstallJammer*, en esta se utilizan los lenguajes *script Tcl*, para la realización de las acciones del sistema y *Tk* para crear las interfaces. A esta API se le incorporan componentes adicionales para complementar los requerimientos, entre los que se encuentran el Lenguaje, ofreciendo un conjunto de idiomas que se utilizarán durante el proceso de instalación, para el instalador propuesto solo se usará el idioma español y el inglés.

⁹ **API:** Del inglés *Application Programming Interface*: es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

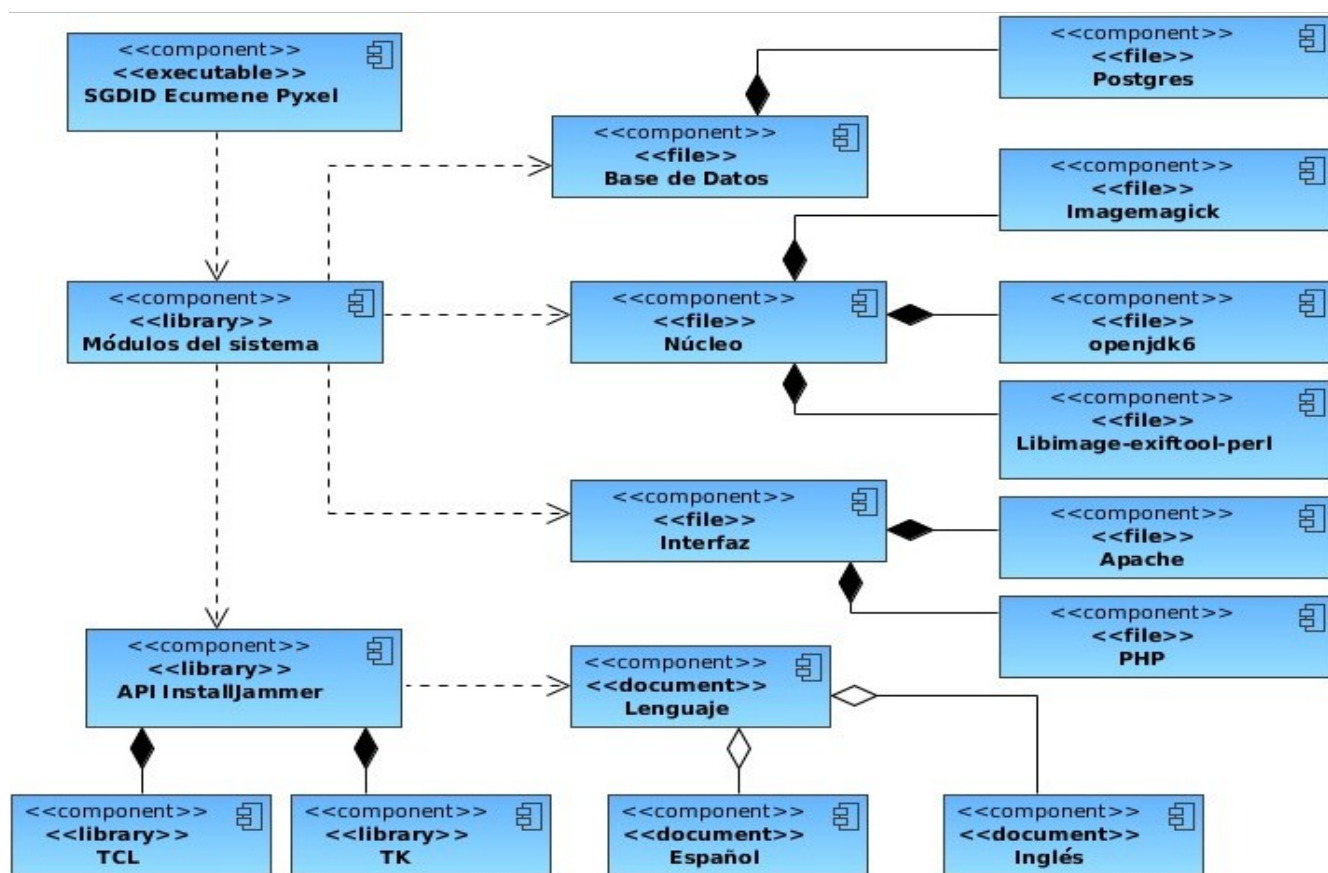


Fig 11: Diagrama de Componentes

3.2 Utilización del InstallJammer

3.2.1 Cargar los módulos del SGDID Ecumene Pyxel.

El SGDID Ecumene Pyxel está formado por una serie de componentes que son cargados en el instalador y de esta forma quedan incorporados al empaquetado que crea la herramienta *InstallJammer*. En la Fig. 12 se refleja como queda el sistema una vez que se cargan los componentes del instalador.

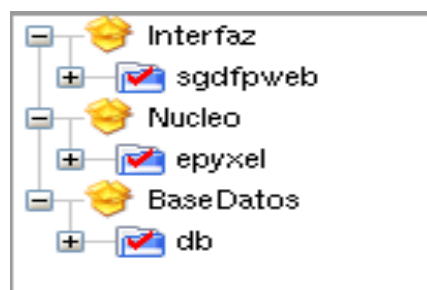


Fig 12: Módulos adicionados al sistema.

3.2.2 Diseño de interfaces

Para automatizar el proceso de instalación del SGDID Ecumene Pyxel se han creado una serie de funcionalidades que contribuyen al cumplimiento de los requisitos necesarios para instalar. La herramienta *InstallJammer* se encarga de crear un empaquetado por defecto al cual se le incorporan una serie de vistas necesarias para realizar varias configuraciones y acciones que son necesarias para el instalador. El primer paso para crear un nuevo panel es agregar uno en blanco con la funcionalidad que se muestra en la Fig. 13.



Fig 13: Acción para agregar un panel en blanco.

Después de haber adicionado este panel comienza el proceso de agregarle los diferentes campos que debe de llevar el panel que se va a crear, en este caso se diseñará una interfaz de configuración que contendrá los diferentes campos. En la Fig. 14 se refleja como se adicionan los campos de tipo textos con las propiedades que debe tener.

	Ej: NameUser	Type	label
	Ej: Passw	Width	100
	Ej: IP	X	320
	Ej: Port	Y	160

Fig 14: Agregar un campo de tipo texto.

Para poder introducir los datos de configuración se agregarán otro tipo de campos los cuales se visualizan

en la siguiente figura.

E_Port2	Type	entry
E_NameBD	Width	160
E_User	X	160
E_Password	Y	120

Fig 15: Agregar un campo de tipo entrada de datos.

Una vez terminado todo el proceso de diseño de los paneles el panel de configuración quedará como se refleja en la Fig. 16.

Fig 16: Panel de configuración del instalador

3.2.3 Validación de campos

Para realizar la validación de los datos introducidos por el usuario fue necesario crear dos funciones en los paneles de configuración. En la primera función se valida que los caracteres que se están escribiendo en el campo del IP sean los correctos, esta función se muestra en la Fig. 17. Primeramente se capturan los datos introducidos por el usuario, una vez capturada la cadena entrada se le aplica una expresión regular comprobando que solo sean entrados números y puntos. Después esa cadena es cortada por los puntos con la función “split” y almacenada en una lista. A la lista se le determina el tamaño para comprobar que

sea siempre menor que 4. Seguidamente se comprueba que el rango de los números que se están escribiendo en cada posición de la lista estén entre 0 y 255.

```

proc isIP {str} {
set ok 1

::InstallAPI::ModifyWidget -widget NextButton -state disabled
if {[regexp {[^0-9.]} $str]} {
    return 0
}

set list [split $str .]
if {[length $list] > 4} {
set ok 0
}

if {"[lindex $list end]" == ""} {
    set list [lrange $list 0 end-1]
}

foreach n $list {
    if {$n < 0 || $n > 255} { set ok 0 }
}

if {[length $list] == 4} {
::InstallAPI::ModifyWidget -widget NextButton -state normal
}

::InstallAPI::ExecuteAction -action ValidateIPkernel
return $ok
}

```

Fig 17: Verificar el formato del campo perteneciente al IP

La segunda función se encarga de verificar que el puerto de comunicación se encuentre correctamente escrito. La validación realizada para este campo solo admite que se introduzcan números, esto se puede lograr producto de que se utiliza la función “*string is int*”.

```

proc isPort { ptr } {
set resultk [string is int $ptr]
::InstallAPI::ExecuteAction -action ValidateIPkernel

return $resultk
}

```

Fig 18: Verificar el formato del campo perteneciente al puerto

En cada uno de las funciones antes mencionadas se hace una llamada a una acción de validación llamada “*ValidateIPkernel*”. Esta acción fue creada para verificar todos los campos que se encuentran en el panel de configuración. Primeramente se captura el tamaño de los valores que se encuentran en las variables globales y se verifica que no existan campos vacíos. Después se realiza una segunda verificación del campo del IP validando que el IP que está escrito en ese campo este completo. Si todas

las validaciones se encuentran correctas entonces se activará el botón “Siguiente” que inicialmente estaba deshabilitado.

```

set dbname [string length $::info(edb dbname)]
set user [string length $::info(edb user)]
set pass [string length $::info(edb password)]
set ips [string length $::info(edb ip db)]
set port [string length $::info(edb port)]

if { $dbname == 0 || $user == 0 || $pass == 0 || $ips == 0 || $port == 0 } {
  ::InstallAPI::ModifyWidget -widget NextButton -state disabled
  return 0;
} else {
  ::InstallAPI::ModifyWidget -widget NextButton -state normal
  return 1
}

set ip $::info(edb ip db)

set list [split $ip .]
if {[llength $list] != 4} {
  ::InstallAPI::ModifyWidget -widget NextButton -state disabled
  return 0
}
foreach n $list {
  if {$n < 0 || $n > 255} {
    ::InstallAPI::ModifyWidget -widget NextButton -state disabled
    return 0
  }
}

return 1

```

Fig 19: Acción para validar todos los campos

3.2.4 Conexión con la Base de Datos

El núcleo del sistema cuenta con una Base de Datos que es necesario restaurar. Para lograr esto fue preciso realizar un “script” que se va a encargar de verificar la conexión con el gestor de base de datos, crear la base de datos y crear el usuario propietario. La variable “conninfo” se encarga de almacenar los datos referentes a la conexión con la base de datos. Una vez que se haya logrado establecer la conexión se realizará una consulta que se encarga de crear la base de datos. Luego se ejecuta la función “pg_exec” a la cual se le pasan los datos de conexión y la variable encargada de realizar la consulta. Una vez realizado este proceso se volverá a establecer una conexión pero esta vez para crear el usuario propietario de la base de datos creada.

```

package require pgintcl

set conninfo "host=<db_ip> dbname=postgres user=<db_user> password=<db_password>"

if {[catch {pg_connect -conninfo $conninfo} pg]} {
    ::InstallAPI::ErrorMessage -message "Failed to connect to database"
    return 0
}

set query "CREATE DATABASE <db_name>"
pg_exec $pg $query

set query "CREATE ROLE alfresco LOGIN SUPERUSER CREATEDB REPLICATION VALID UNTIL 'infinity';"
pg_exec $pg $query

pg_disconnect $pg

```

Fig 20: Función para la conexión con la base de datos.

3.2.5 Ejecutar programas externos.

La ejecución de un programa externo es una de las funcionalidades que brinda el *InstallJammer*. Esta funcionalidad fue utilizada para restaurar la base de datos del Núcleo mediante la ejecución en consola del fichero "restoredatabase.sh".

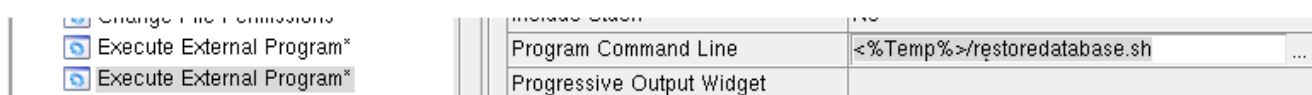


Fig 21: Funcionalidad: Ejecutar programa externo

3.2.6 Escribir ficheros.

Esta acción sirve para escribir texto dentro de un fichero que puede ser sobrescrito o creado automáticamente en caso de que no exista. El fichero que se ejecuta en el programa externo anteriormente explicado se crea mediante esta funcionalidad y a la vez es escrito con los comandos que deben ser ejecutados. En la Fig. 22 se muestra como se emplea esta funcionalidad creando este fichero en una carpeta temporal para que solo se encuentre temporalmente en la PC.

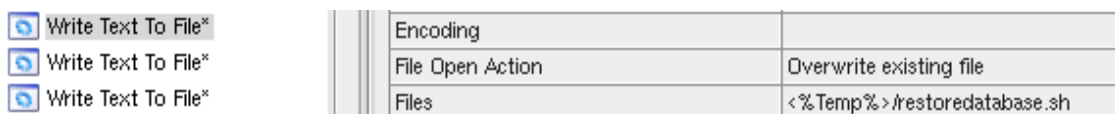


Fig 22: Funcionalidad para escribir un fichero.

Dentro de este fichero, que se ejecuta en la consola se encuentran una serie de comandos en lenguaje

“*bash*” los cuales se presentan a continuación.

```
pg_restore --host <%edb_ip_db%> --port <%edb_port%> --username <%edb_user%> --dbname <%edb_dbname%>
--verbose <%Temp%>/db/epyxel.backup
update-rc.d alfresco defaults
```

Fig 23: Fichero “*restoredatabase.sh*”

3.2.7 Comprobar módulo seleccionado para instalar.

El instalador cuenta con una serie de módulos que pueden ser instalados de manera independiente. Para tener control sobre las diferentes vistas y funcionalidades que deben aparecer en dependencia de la selección realizada, es preciso conocer que módulo fue seleccionado. Se realizan una serie de “*script*” que se encargan de verificar el componente seleccionado. En la Fig. 24 se puede ver que en la variable “*SelectedComponents*” se almacenan los módulos que son seleccionados para su instalación. Esta variable se corta con la función “*split*” por cada uno de los componentes seleccionados y se asignan a una lista que es recorrida mediante un ciclo. Después se compara cada posición de la lista con la variable que tiene el valor del módulo que se desee comprobar que se encuentre seleccionado. En el caso del “*script*” a continuación se realiza la verificación para el módulo base de datos, si está seleccionado, la función devolverá el número 2 que pertenece al tipo de instalación avanzada. Si no se cumplen las condiciones antes mencionadas se retornará 0 y no se realiza ninguna operación con el módulo base de datos. Este “*script*” se realiza para cada uno de los módulos del sistema, siempre que el usuario seleccione el tipo de instalación avanzada.

```
set selectedcomponent <#SelectedComponents#>
set intA "BaseDatos"
set complist [split $selectedcomponent]
foreach compname $complist {
  if ( [string match -nocase $intA $compname] )
  {
    return 2
  }
}
return 0
```

Fig 24: Función para verificar un módulo seleccionado.

3.2.8 Creación de condiciones

La creación de condiciones se puede realizar de manera visual en dependencia de la función que se

quiera comprobar en un determinado proceso. La que se muestra a continuación se hace para verificar los procesos pertenecientes a la base de datos. Se realizan dos condiciones, una para el tipo de instalación típica y otra para el tipo de instalación avanzada, verificando que se cumpla al menos una de las dos. Para el caso independiente de cada restricción se establece el “*String Equal Condition*” para comprobar la igualdad de las cadenas. Estas cadenas serán la variable “<%check_db%>” que almacena lo que devuelve la función que aparece en la Fig. 24 y esto se compara con la segunda cadena que en el ejemplo mostrado tiene el valor 1 siendo esta condición para comprobar el tipo de instalación típica.

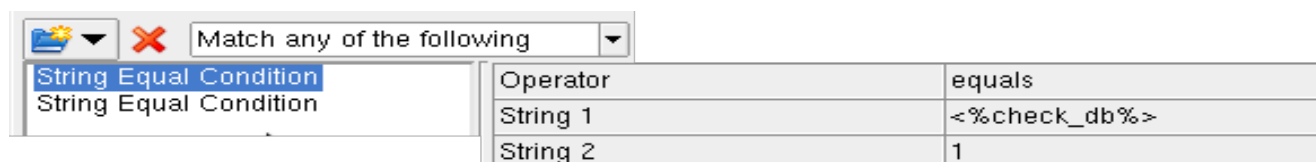


Fig 25: Condición para verificar los paneles que se deben mostrar.

3.2.9 Creación de los grupos de acción.

Los grupos de acción son una forma de agrupar las acciones individuales en un solo conjunto de condiciones. Estos grupos se definen independientemente de cualquier panel en particular y se puede llamar desde cualquier lugar de la instalación.

En el instalador para el SGDID Ecumene Pyxel fue necesaria la creación de varios grupos, entre los que se encuentran el grupo que se muestra en la Fig. 26. Este grupo contiene las acciones encargadas de verificar las dependencias pertenecientes al Núcleo. Las acciones que se crearon son denominadas “*Set Virtual Text*” y se encargan de comprobar las dependencias por medio de condiciones. Por otra parte en este grupo se muestra un mensaje de error en caso de que no se cumpla alguna condición. Además, se coloca la acción “*Exit*” para una vez leído el mensaje se cierre toda la aplicación.

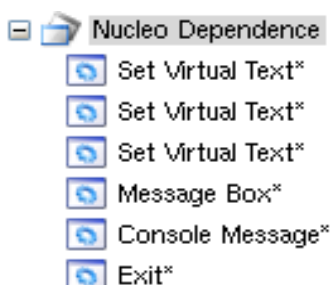


Fig 26: Creación de un grupo de acción.

3.2.10 Verificar dependencias

Para validar el entorno de trabajo es necesario verificar que las dependencias del sistema se encuentren instaladas en la PC. Para comprobar esto se establecen una serie de condiciones, la que se presenta en la Fig. 27 es para comprobar que el paquete “*Imagemagick*” se encuentre instalado. Para ello se utiliza la condición “*File Exists Condition*” que es para verificar la existencia de un fichero. Dentro de los datos que exige esta condición está la dirección del fichero que se desea comprobar que se encuentre en el entorno de trabajo.



The image shows a configuration window with a title bar that says "Match all of the following". Below the title bar, there is a dropdown menu with "File Exists Condition" selected. To the right of the dropdown is a table with two rows:

Filename	/usr/lib/mime/packages/imagemagick
Operator	does not exist

Fig 27: Condición para verificar las dependencias

3.2.11 Acción para modificar en valor de la variables virtual.

La acción “*Set Virtual Text*” se encarga de llamar a las variables virtuales y en la aplicación se visualiza el valor de esta variable. En la Fig. 28 se muestra la utilización de esta acción para mostrar el texto del mensaje de error que se muestra durante la validación de las dependencias.



The image shows a configuration window with a title bar that says "Set Virtual Text". Below the title bar is a table with two rows:

Value	<%dependence_kernel%> imagemagick
Virtual Text	dependence_kernel

Fig 28: Acción “*Set Virtual Text*”.

3.2.12 Leer un fichero dentro una variable de texto virtual.

Esta acción va a leer un archivo en el sistema destino que se almacena en la variable de texto virtual “<%InstallDir%>”. Esto es útil para obtener las propiedades o información de un archivo, en este caso se utiliza para obtener la información que se encuentra escrita en el archivo “*infodb.sav*”. Este archivo se crea en caso de que se haya instalado la base de datos del sistema y almacena la información que pertenece a los datos de conexión con el gestor de base de datos. Esta acción es necesaria para desinstalar la base de datos.

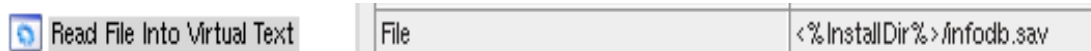


Fig 29: Acción "Read File Into Virtual Text"

3.2.13 Eliminar base de datos instalada.

Para desinstalar la base de datos del SGDID Ecumene Pyxel fue necesaria la realización de un "script" que se encarga de leer el fichero "infodb" y por medio de la función "split" va a cortar el contenido de este fichero por los saltos de línea y adicionarlos a una lista. Una vez conformada la lista se realizará una verificación de su tamaño para comprobar que tenga cinco elementos. En caso de que la longitud de la lista no coincida se lanzará un mensaje de error indicando que la conexión con la base de datos ha fallado. Si la condición antes mencionada se cumple, se realiza la conexión con la base de datos mediante la utilización de los datos almacenados en la lista y se elimina la base de datos del sistema por medio de una consulta que se encuentra en la variable "\$query" y después se le pasa a la función "pg_exec" junto con la variable "\$pg". que tiene los datos de conexión con el gestor de base de datos.

```
package require pgintcl
set list [split <%infoDB%> \n]
if {[llength $list] != 5} {
    ::InstallAPI::ErrorMessage -message "Failed to connect to database
2"
    return 0
}
set message [lindex $list 1]
set conninfo "host=[lindex $list 3] dbname=postgres user=[lindex $list 1]
password=[lindex $list 2] port=[lindex $list 4]"
if {[catch {pg_connect -conninfo $conninfo} pg]} {
    ::InstallAPI::ErrorMessage -message "Failed to connect to database"
    return 0
}
set query "DROP DATABASE [lindex $list 0];"
pg_exec $pg $query
```

Fig 30: Función para eliminar la base de datos.

3.2.14 Acción para desinstalar archivos sobrantes

Esta acción se utiliza para la eliminación de cualquier rastro de la desinstalación que no puedo ser

eliminado en la primera pasada. Esto significa eliminar directorios a pesar de que no están vacíos o por la fuerza tratando de eliminar un archivo que no se pudo eliminar. En el desinstalador para el SGDID Ecumene Pyxel fue necesaria la utilización de esta acción para eliminar archivos que no eran borrados con la acción por defecto que tiene el desinstalador esta acción se denomina "Uninstall Selected Files¹⁰" o eliminar archivos seleccionados.



Fig 31: Acción "Uninstall Leftover Files".

3.2.15 Instalación en modo consola

En la instalación en modo consola fue necesaria la creación de un conjunto de acciones que se realizarán de manera secuencial, de esta forma cada uno de los mensajes que guiarán la instalación del sistema aparecerán en el orden correcto. A continuación se presentarán una serie de funcionalidades que fueron utilizadas en la construcción de la instalación en el modo consola.

3.2.15.1 Acción para preguntar en la consola si o no

Se hizo uso de la acción "Console Ask Yes or No" o pregunta en la consola de si o no. Esta acción se utiliza en la instalación por modo consola para solicitar que el usuario responda una pregunta que requiere una respuesta de sí o no. El usuario tiene la obligación de dar una respuesta válida antes de continuar la instalación. Esta funcionalidad es utilizada para determinar si se desea continuar con la instalación o se cancela.



Fig 32: Acción "Console Ask Yes or No"

3.2.15.2 Acción para ejecutar acciones.

Esta acción se utiliza como intermediaria para ejecutar otra acción o grupo de acciones. Mediante esta funcionalidad se pueden comprobar las condiciones anteriormente mencionadas en la instalación por modo interfaz gráfica. En el instalador para el SGDID Ecumene Pyxel se hizo uso de esta acción para ejecutar las condiciones de los grupos de acción que se encargan de validar tanto el entorno de desarrollo

¹⁰ *Uninstall Selected Files*: Desinstalar archivos seleccionados.

como los datos introducidos por el usuario.



Fig 33: Acción "Execute Action"

3.2.15.3 Acción para los mensajes en consola

Con esta acción se pueden mostrar cada uno de los mensajes que intervienen en la instalación en modo consola. De esta forma se mantiene al usuario informado sobre lo que está pasando en la instalación y lo que debe hacer.



Fig 34: Acción "Console Message"

3.2.15.4 Acción para obtener la entrada de usuario en la consola

Esta acción se utiliza en una instalación de la consola para obtener una cadena de entrada del usuario. La cadena no se comprueba de ninguna manera, por lo que depende del desarrollador, realizar una función que compruebe los datos entrados por el usuario. Esta acción se utiliza para pedirle al usuario que introduzca los datos de configuración de los componentes del sistema y así poder capturarlos.



Fig 35: Acción "Console Get User Input"

3.3 Modelo de pruebas

En el presente trabajo se realiza el proceso de pruebas de caja negra. Estas pruebas se llevan a cabo sobre la interfaz del software y se centran principalmente en los requisitos funcionales del sistema. Permite obtener un conjunto de condiciones de entrada que ejercitan completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos [22]. Se considera que estas pruebas permiten detectar:

- Funciones incorrectas o ausentes.

- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien [22].

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 [22].
2. **Técnica del Análisis de Valores Límites:** Esta Técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables [22].
3. **Técnica de Grafos de Causa-Efecto:** Es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones [22].

Determinando las funciones que realiza el instalador se le pueden aplicar estas pruebas a la aplicación, para de esta forma comprobar que las funcionalidades del programa están correctamente desarrolladas. 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 descubren errores, reduciendo así el número de casos de prueba a desarrollar [22].

3.3.1 Casos de prueba

Los casos de prueba que se realizaron se encuentran plasmados en el documento *CIDI_ISGDIDEP_0122_Diseño de Casos de Pruebas basado en Requisitos* [23], que pertenece al expediente de proyecto.

3.3.2 Pruebas de portabilidad

Las pruebas de portabilidad se realizan para determinar la habilidad del software para ejecutarse en

diferentes entornos informáticos. La portabilidad presenta como característica la capacidad de transferir el software de una organización o entorno de hardware o software a otro [24]. Para darle cumplimiento a esta prueba se ejecutó el instalador para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel en diferentes distribuciones del sistema operativo GNU/Linux, entre las que se encuentran Debian, Ubuntu y Linux Mint. En cada una de estas distribuciones el instalador funciona correctamente logrando una instalación exitosa.

3.3.3 Detección de no conformidades

Durante la realización de las pruebas quedó demostrada la importancia de esta fase en el desarrollo de un software. Se lograron detectar una serie de no conformidades las cuales fueron corregidas. Dentro de estas no conformidades se pueden mencionar:

Iteración	No conformidad	Descripción	Resuelta
1	Incorrecta validación del entorno de trabajo.	La validación del entorno de trabajo se realizó en diferentes distribuciones del sistema operativo GNU/Linux y específicamente en la distribución "Ubuntu" los paquetes pertenecientes a las dependencias del SGDID Ecumene Pyxel no son instalados en las mismas direcciones. Esto provocaba que el instalador no encontrara las dependencias y lanzara un error aunque tuviera los paquetes instalados. Para erradicar este problema se hizo un análisis de donde eran instalados los paquetes en cada distribución logrando establecer una relación entre las distribuciones y de esta forma la verificación de la existencia de las dependencias se pudo realizar de	Si

		manera más estándar.	
	Insuficiente validación de los datos de configuración.	El proceso de validación de los datos de configuración del sistema no verificaba si los datos introducidos por el usuario eran los correctos para establecer una conexión con el gestor de base de datos.	Si
2	Incorrecta verificación de los módulos a instalar.	En la selección de los módulos a instalar, cuando no se seleccionaba ningún módulo, el proceso de instalación continuaba aunque no estuviera instalando ninguno de los componentes que forman el SGDID Ecumene Pyxel.	Si
3	Incorrecto proceso de desinstalación de las base de datos.	Durante el proceso de desinstalación se detectó que el desinstalador no borraba las bases de datos creadas y fue necesario agregarle una nueva funcionalidad que fuera capaz de obtener los datos de conexión con el gestor de base de datos y de esta forma las borrar.	Si
	Presencia de errores en algunos mensajes de la aplicación.	Fueron detectados una serie de mensajes que presentaban errores ortográficos en el idioma español. Además, de que algunos mensajes no quedaban claros en cuanto a la información que debían transmitirle al usuario.	Si

Tabla 3: No conformidades del sistema.

Una vez que fueron corregidas las deficiencias expuestas en la tabla anterior se aplicaron nuevamente estas pruebas logrando satisfacer cada uno de los requisitos del sistema que fueron sometidos a estas pruebas.

Conclusiones del capítulo

Luego de haber implementado el instalador para el SGDID Ecumene Pyxel mediante el uso del sistema generador de instaladores *InstallJammer* se puede determinar que:

- El instalador cumple con los requisitos definidos tanto para facilitar como agilizar el proceso de instalación.
- Se logró la realización exitosa de cada uno de los procesos de configuración, restauración y despliegue del sistema.
- Las funcionalidades brindadas por la herramienta de desarrollo proporcionaron diversas acciones que fueron explotadas en la implementación del instalador y del desinstalador.
- El procedimiento de prueba permitió que se detectaran algunos errores en la aplicación que fueron posteriormente corregidos para que el instalador funcionara correctamente.

Conclusiones

Se puede concluir que para el desarrollo del trabajo de diploma se hizo un análisis de la problemática existente para identificar los principales objetivos y tareas a ejecutar en el desarrollo de la investigación. Logrando de esta forma obtener un ejecutable capaz de ofrecer diferentes beneficios entre los que se pueden identificar:

- La instalación del SGDID Ecumene Pyxel de manera automática, ya sea por medio de la interfaz gráfica como mediante la consola.
- El tiempo de las actividades del proceso de despliegue disminuyó comparado con el método que se utiliza actualmente para instalar el sistema, producto de que el proceso de instalación se realiza con mayor facilidad y rapidez.
- Se logra disminuir la dependencia hacia los recursos humanos del equipo de desarrollo para poder instalar el sistema en las diferentes instituciones que lo utilizarán. Esto ocurre producto de que las características del instalador no requieren de un elevado nivel de conocimientos técnicos para su comprensión.

Por lo antes planteado se puede determinar que el objetivo perseguido durante el desarrollo del instalador para el SGDID Ecumene Pyxel fue alcanzado satisfactoriamente.

Recomendaciones

Luego de la investigación realizada en el presente trabajo de diploma se recomienda tanto al equipo de desarrollo del sistema como a los futuros usuarios del instalador para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel V1.0 que:

- Se distribuya la aplicación junto con el manual de usuario y así el usuario pueda obtener una capacitación antes de comenzar el proceso de instalación.
- Se realice la instalación previa de las dependencias del SGDID Ecumene Pyxel, asegurando de esta forma que el proceso de instalación se efectúe satisfactoriamente.
- En futuras versiones del instalador sería recomendable incorporarle la instalación de las dependencias del sistema, para que de esta forma el usuario no las tenga que tener en cuenta. Además, de realizar la validación de los campos pertenecientes al IP para IPv6¹¹, debido a que el instalador actual solo está validado para IPv4.

¹¹ **IPv6:** Internet Protocol version 6 del español Protocolo de Internet versión 6.

Bibliografía

- [1]. ChullitoHost. Software Contabilidad - Restaurant - Colegios - Distribuidoras - Ferreterías-Hoteles: Importancia del Software en las Empresas, 2010. [Disponible en: <http://softwarearequipa.blogspot.com/2010/05/importancia-del-software-en-las.html>]
- [2]. Isabel María García Hilarión , Sergio Antonio Segura Fernández . Interfaz Web para el Sistema de Gestión Documental de Imágenes Digitales Ecumene Pyxel, 2012. [Disponible en: <https://10.58.16.16/svn/gdprensa/Tesis-2012/pyxel-web/Documentacion/>]
- [3]. Rolando Díaz Cotilla. Diseño e Implementación de la versión 2.0 del instalador de Cedrux. Universidad de la Ciencias Informáticas. La Habana, 2011. 62
- [4]. Uniblue. How To Install Software on the Computer, 2007. [Disponible en: <http://www.liutilities.com/how-to/install-software-on-the-computer/>]
- [5]. Claudio Ernesto Florián Arenas. Historia del Software. 2003
- [6]. Indira Escalona Martínez. Propuesta de filtrado de correo electrónico para el sistema Smart Keeper. Universidad de las Ciencias Informáticas. La Habana, 2012. 71
- [7]. BitRock. InstallBuilder 8, 2012. [Disponible en: <http://installbuilder.bitrock.com/>]
- [8]. BitRock. Descargar BitRock InstallBuilder en español, 2012. [Disponible en: <http://www.descargas5.com/descargar/12798/BitRock-InstallBuilder.html>]
- [9]. BitRock. InstallBuilder Features, 2012. [Disponible en: <http://installbuilder.bitrock.com/installbuilder-features.html>]
- [10]. Flexera Software. InstallAnywhere Installation Software. *Cross-Platform Installers*, 2012. [Disponible en: <http://www.flexerasoftware.com/products/installanywhere.htm>]
- [11]. Flexera Software. InstallAnywhere: Features, 2012. [Disponible en: <http://www.flexerasoftware.com/products/installanywhere/features.htm>]
- [12]. InstallJammer. InstallJammer Multiplatform Installer, 2010. [Disponible en: <http://www.installjammer.com>]
- [13]. Yulaine Hernández Alonso. Configuración de la metodología, 2012
- [14]. Centro de Ideoinformática. Base Tecnológica CIDI, 2012

- [15]. ForeUI. Buy ForeUI License, 2012. [Disponible en: <http://www.foreui.com/buy.htm>]
- [16]. Angel Olivera. Requerimientos funcionales y no funcionales, 2010. [Disponible en: <http://es.scribd.com/doc/37187866/Requerimientos-funcionales-y-no-funcionales>]
- [17]. Neyvis Herrera Ramírez. CIDI_ISGDIDEP_0114_Especificación de casos de uso, 2012
- [18]. Liliana Pacheco. Arquitectura de Software Fundamentos de Ingeniería de Software, 2012. [Disponible en: <http://www.slideshare.net/lilyPacheco7/arquitectura-de-software-13925226>]
- [19]. Neleste. Modelo Vista Controlador, 2008. [Disponible en: <http://www.neleste.com/modelo-vista-controlador/>]
- [20]. Ángel Febe. Utilización del Patrón Modelo Vista Controlador (MVC) en el diseño de software educativos , 2013. [Disponible en: <http://www.monografias.com/trabajos43/patron-modelo-vista/patron-modelo-vista.shtml>]
- [21]. Slideshare. Diagramas De Interacción Análisis y Diseño Orientado a Objetos, 2009. [Disponible en: <http://www.slideshare.net/jent46/diagramas-de-interaccion-presentation-892165>]
- [22]. Entorno Virtual de Aprendizaje. Material_de_caja_b_y_caja_n, 2013. [Disponible en: <http://eva.uci.cu/mod/resource/view.php?id=8649&subdir=/Comun>]
- [23]. Neyvis Herrera Ramírez. CIDI_ISGDIDEP_0122_Diseño de Casos de Pruebas basado en Requisitos, 2012
- [24]. Natalia Juristo, Ana M. Moreno, Sira Vegas. Técnicas de evaluación de software versión 12.0 . 2013. [Disponible en: <http://eva.uci.cu>]
- [25]. Martín Fernández, Francisco J, Yusef Hassan Montero. Qué es la Arquitectura de la Información, 2003. [Disponible en: <http://www.nosolousabilidad.com/articulos/ai.htm>]
- [26]. CarlosPes. Definición de Compilación (de un programa) - Concepto y Significado, 2013. [Disponible en: Available from: <http://www.carlospes.com/minidiccionario/compilacion.php>.]
- [27]. James Rumbaugh, Ivar Jacobson, Grady Booch. El Proceso Unificado de Desarrollo de Software. España, Addison Wesley, 2000. 435
- [28]. Infocopy. Definición y funciones de la gestión documental, 2010. [Disponible en: <http://infocopysolucionesdeimpresion.wordpress.com/2010/11/15/definicion-y-funciones-de-la->

- gestion-documental/]
- [29]. Definición.de. Definición de hardware, 2013. [Disponible en: <http://definicion.de/hardware/>]
- [30]. Definición.org. Definición de interfaz grafica de usuario, 2013. [Disponible en: <http://www.definicion.org/interfaz-grafica-de-usuario>]
- [31]. EcuRed. Inteligencia artificial, 2013. [Disponible en: http://www.ecured.cu/index.php/Inteligencia_artificial]
- [32]. Eva Méndez, José A. Senso. Metadatos: concepto y motivacion, 2004. [Disponible en: <http://www.sedic.es/autoformacion/metadatos/tema1.htm.>]
- [33]. InformaticandoTf. Multiplataforma (definición), 2011. [Disponible en: <http://informaticandotf.blogspot.com/2010/08/multiplataforma-definicion.html.>]
- [34]. Slideshare. metodologia, 2012. [Disponible en: <http://www.slideshare.net/germaniabetty/metodologia-12836108.>]
- [35]. EcuRed. Máquinas virtuales, año. [Disponible en: http://www.ecured.cu/index.php/Máquinas_virtuales#Definici.C3.B3n.]
- [36]. Definición.de. Definición de software, 2013. [Disponible en: <http://definicion.de/software/>]
- [37]. Admin UCI. Stakeholders, 2011. [Disponible en: <http://www.slideshare.net/adminuci/stakeholders-10596185>]
- [38]. Definición.de. Definición de usabilidad, 2013. [Disponible en: <http://definicion.de/usabilidad/>]

Anexos

Anexo 1: Diagrama de procesos para desinstalar el sistema

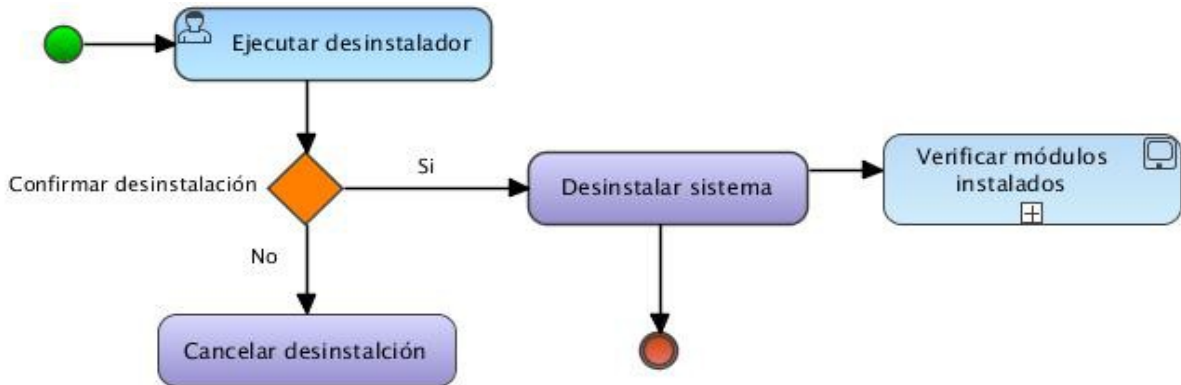


Fig 36: Diagrama de procesos para desinstalar el sistema

Anexo 2: Diagrama de Clases del análisis para el CU_ Desinstalar SGDID Ecumene Pyxel.

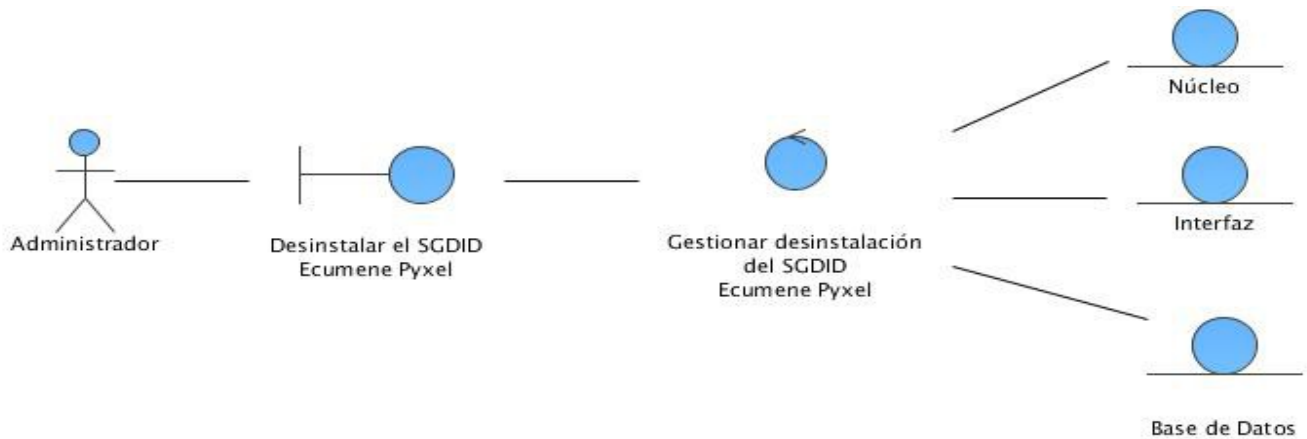


Fig 37: Diagrama de Clases del análisis para el CU_ Desinstalar SGDID Ecumene Pyxel.

Anexo 3: Diagrama de Colaboración para el CU_ Desinstalar SGDID Ecumene Pyxel.

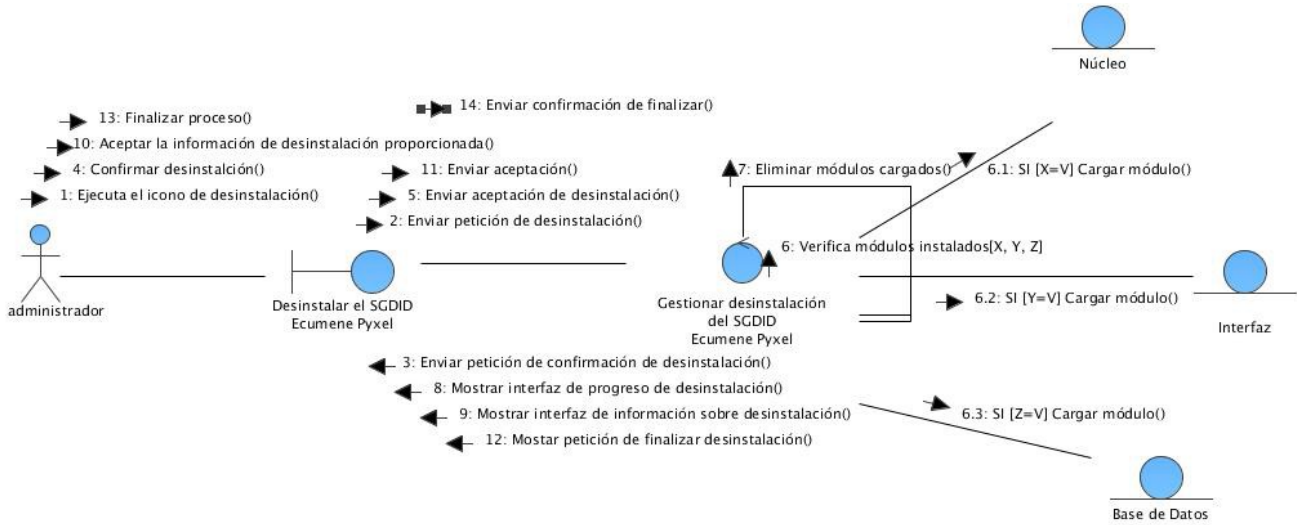


Fig 38: Diagrama de Colaboración para el CU_Desinstalar SGDID Ecumene Pyxel

Anexo 4: Especificación de caso de uso: CU_ Desinstalar SGDID Ecumene Pyxel.

Objetivo	Desinstalar el SGDID Ecumene Pyxel.	
Actores	Administrador del sistema: Inicia el proceso de desinstalación	
Resumen	Se desinstala el SGDID Ecumene Pyxel	
Complejidad	Media	
Prioridad	Media	
Precondiciones	No se debe tener la aplicación funcionando en el momento de la desinstalación.	
Postcondiciones		
Flujo de eventos		
Flujo básico <Desinstalar el SGDID Ecumene Pyxel>		
	Actor	Sistema
1	Presiona el ícono de desinstalación del SGDID Ecumene Pyxel.	
2		Muestra una interfaz para confirmar la desinstalación del sistema con las opciones de

		“Aceptar” y “Cancelar”
3	Presiona el botón “Aceptar”.	
4		Verifica que módulos del SGDID Ecumene Pyxel fueron instalados.
5		Si la base de datos fue instalada muestra una interfaz con la opción de seleccionar la base de datos para ser desinstalada. Brinda las opciones de “Siguiete” y “Cancelar”.
6	Marca la base de datos para desinstalar y presiona el botón “Siguiete”.	
7		Muestra la interfaz que visualiza el proceso de desinstalación.
8		Muestra la interfaz de finalizar el proceso de desinstalación.
9	Presiona el botón “Finalizar”.	

Flujos alternos

1 Cancelar la desinstalación.

	Actor	Sistema
3.1	Presiona el botón “Cancelar”.	
3.2		Se cierra la interfaz quedando cancelada la desinstalación de la aplicación.
6.1	Presiona el botón “Cancelar”.	
6.2		Se cierra la interfaz quedando cancelada la

		desinstalación de la aplicación.
Sección 1: “Desinstalar el SGDID en modo consola”		
Flujo básico <“Desinstalar el SGDID en modo consola”>		
	Actor	Sistema
1	Ejecutar el desinstalar en la consola central del sistema.	
2		Muestra un mensaje indicando que se desinstalará el SGDID Ecumene Pyxel. Brinda las opciones de “Y” para continuar y “n” para cancelar.
3	Presiona la tecla “Y” y seguidamente “Enter”.	
4		Muestra un mensaje indicando si desea eliminar la base de datos creada. Con las opciones de “n” y “Y”.
5	Presiona la tecla “Y” para confirmar la eliminación de la base de datos y presiona “Enter”.	
6		Inicia la desinstalación del sistema.
Flujos alternos		
1. Cancelar instalación		
	Actor	Sistema
3.1	Presiona la tecla “n”.	

3.2		Se cancela el proceso de desinstalación.
5.1	Presiona la tecla "n".	
5.2		Se cancela el proceso de eliminar la base de datos creada.
Relaciones	CU Incluidos	
	CU Extendidos	
Requisitos no funcionales		
Asuntos pendientes		

Tabla 4: Especificación de caso de uso: CU_ Desinstalar SGDID Ecumene Pyxel.

Glosario de términos

A

Arquitectura de información: El estudio de la organización de la información con el objetivo de permitir al usuario encontrar su vía de navegación hacia el conocimiento y la comprensión de la información [25].

C

Compilación: En programación, cuando se desarrolla un programa, la compilación consiste en que el compilador traducirá el código fuente a código máquina, también llamado código objeto [26].

D

Diagrama de clases: Un diagrama que muestra un conjunto de clases, interfaces, colaboraciones y las relaciones entre estos; los diagramas de clases muestran el diseño de un sistema desde el punto de vista estático [27].

Diagrama de interacción: Un diagrama de interacción se entiende como la representación de secuencias que componen un sistema, para facilitar el entendimiento de información y la relación que mantiene entre las diferentes partes [21].

G

Gestión Documental: La gestión documental consiste en un conjunto de técnicas y prácticas usadas para administrar toda la documentación de una organización. No existen dos organizaciones idénticas, por lo que la gestión de esos documentos deberá adaptarse a cada empresa [28].

H

Hardware: Conjunto de componentes que conforman la parte material o física de una computadora [29].

I

Interfaz gráfica: Componente de una aplicación informática que el usuario visualiza y a través de la cual opera con ella. Está formada por ventanas, botones, menús e íconos, entre otros elementos [30].

Inteligencia artificial: Aquella inteligencia exhibida por artefactos creados por humanos. A menudo se aplica hipotéticamente a los computadores. El término también se usa para referirse al campo de la investigación científica que intenta acercarse a la creación de tales sistemas que tratan de conseguir que los ordenadores simulen en cierta manera la inteligencia humana [31].

L

Licencia privativa: Establece que el software bajo esta licencia no se pueden usar, copiar, modificar ni redistribuir sin el permiso del propietario del software.

M

Metadato: Es toda aquella información descriptiva sobre el contexto, calidad, condición o características de un recurso, dato u objeto que tiene la finalidad de facilitar su recuperación, autenticación, evaluación, preservación y/o interoperabilidad [32].

Multiplataforma: En informática se dice que un elemento (programas, lenguajes de programación, elementos de hardware) es multiplataforma cuando tiene la capacidad de funcionar en más de un sistema operativo con similares características y sin que su funcionalidad varíe en exceso [33].

Modo consola: Aplicación informática en modo texto, en este modo el usuario puede escribir comandos para que sean interpretados y ejecutados por el sistema operativo.

Metodología: Un conjunto de procedimientos, reglas, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software [34].

Máquinas virtuales: Contenedor de software perfectamente aislado que puede ejecutar sus propios sistemas operativos y aplicaciones como si fuera un ordenador físico [35].

R

Repositorio: Deposito o archivo, es un sitio centralizado donde se almacena y mantiene la información digital habitualmente base de datos o archivos informáticos

S

Software: Conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora. Es el equipamiento lógico e intangible de un ordenador [36].

Stakeholders: Son personas u organizaciones cuyos intereses pueden ser impactados de forma positiva o negativa por el proyecto. Esto puede incluir a individuos y grupos que no se hayan considerado con anterioridad [37].

U

Usabilidad: Referencia la facilidad con que un usuario puede utilizar una herramienta fabricada por otras personas con el fin de alcanzar un cierto objetivo. Está vinculada, a la simpleza, la facilidad, la comodidad y la practicidad [38].