

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS



*Trabajo de diploma para optar por el título de Ingeniero
en Ciencias Informáticas*

“Herramienta para la migración de estaciones de trabajo”

Autor:

Grabiél Montero Moreno

Tutores:

Ing. Alfredo Pérez Benitez

Ing. María Leisy González Carrera

La Habana, Cuba.

Julio de 2015

Declaración de autoría

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

Para que así conste firmo la presente a los ____ días del mes de ____ del año 2015.

Grabiél Montero Moreno

Ing. Alfredo Pérez Benítez

Ing. María Leisy González Carrera

Agradecimientos

A mis padres y mi hermano.

A toda mi familia, a Mercedes Fuentes, a Juan Carlos y a Delís.

A Karel Rafael, Guillermo (El Guille), Marlon Suarez (El croque), Camili, Ernesto Camué, Leonardo, Carlos Lezcano (El gordo), Osvaldo Pérez Rodríguez (Dragoon), René Ricardo (El Kuro), Roger González (El Bolo), Nilberto.

A los que viven en el 126201.

A los integrantes de 1502.

A Cavernícolas y todos los del Inter. Por ser los dos mejores equipos de la facultad.

A todos aquellos con los que he compartido un juego de fútbol o algún doble.

A mis tutores y todos los profesores que me han impartido alguna asignatura.

A Carlos Rafael Alarcón Moreno, ya hace 10 años que no lloro, QUE EN PAZ DESCANSES mi hermano.

Si hay alguien que no entre en ningún perfil recibirá un agradecimiento personal, solo tienen que pedirlo.

Dedicatoria

A Carlos Rafael Alarcón Moreno mi hermano, que en paz descansas.

A Mirna de la Caridad Moreno Ruffinell mi mamá.

A Alfonso Montero Montoya mi papá.

A Adolfo Fong Moreno mi hermano mayor.

Resumen

La migración de las aplicaciones privativas a estándares libres y de código abierto es uno de los pasos más importantes que da el país en aras de conseguir la independencia y soberanía tecnológica. En la Universidad de las Ciencias Informática existen centros dedicados a la producción de *software*, uno de estos es el Centro de *Software* Libre. Al centro pertenece el departamento de Servicios Integrales en Migración, Asesoría y Soporte, el cual tiene entre sus principales objetivos guiar y apoyar procesos de migración y soporte a *software* libre. En la presente investigación se definen las herramientas, tecnologías, y otros aspectos importantes, que permiten el desarrollo de una herramienta de apoyo para el proceso de migración de estaciones de trabajo. Dicha herramienta garantiza el traslado de los datos de los usuarios y configuraciones de aplicaciones en el proceso de migración de las estaciones de trabajo de *Windows* XP, 7 y 8 a la distribución GNU/Linux Nova. Con la utilización de este sistema se asegura la integridad de los datos y se disminuye el tiempo empleado por un especialista en la realización de los procesos de migración a código abierto.

Palabras clave: código abierto, estándares libres, migración, soberanía tecnológica, *software* libre.

Índice de contenidos

Introducción.....	1
Capítulo 1: Fundamentación teórica.....	7
Introducción.....	7
1.1 Conceptos asociados.....	7
1.2 Etapas de la guía cubana.....	9
1.3 Características de los sistemas operativos.....	11
1.3.1 GNU/Linux.....	11
1.3.2 Windows.....	13
1.4 Migración de datos de usuario y configuración de aplicaciones.....	14
1.4.1 Descripción del proceso de migración de datos de usuario y configuraciones de aplicaciones de Windows XP, 7 y 8 a GNU/Linux Nova.....	14
1.4.2 Definición de los procedimientos para migrar los datos de los usuarios y las configuraciones de aplicaciones en el proceso de migración de Windows XP, 7 y 8 a GNU/Linux Nova.....	15
1.5 Herramientas para migrar los datos de los usuarios y configuraciones de aplicaciones.....	21
1.5.1 Comparativa de las herramientas.....	22
1.6 Lenguajes y herramientas.....	23
1.6.1 Lenguajes.....	23
1.6.2 Herramientas.....	24
1.7 Metodología a utilizar.....	25
Conclusiones parciales.....	26
Capítulo 2: Análisis, diseño e implementación de la solución propuesta.....	27
Introducción.....	27
2.1 Propuesta de solución.....	27
2.1.2 Descripción de la propuesta.....	29
2.2 Lista de Reserva del Producto (LRP).....	29
2.3 Historias de Usuarios.....	32

2.4 Arquitectura del sistema.....	43
2.5 Diagrama de clases.....	44
2.5.1 Principales clases.....	46
2.6 Patrones de diseño.....	46
2.7 Plan de entrega.....	49
2.8 Tareas de Ingeniería.....	50
2.9 Estándares de codificación.....	53
2.9.1 Estándar de código para la implementación de la herramienta.....	53
2.10 Diagrama de Componentes.....	54
Conclusiones parciales.....	56
Capítulo 3: Pruebas a la herramienta propuesta.....	57
Introducción.....	57
3.1 Pruebas de software.....	57
3.1.1 Prueba de aceptación.....	57
3.1.2 Casos de pruebas de aceptación.....	57
3.2 Integridad de los datos.....	63
3.3 Disminución del tiempo en la migración de los datos y configuraciones.....	63
Conclusiones parciales.....	64
Conclusiones generales.....	65
Recomendaciones.....	66
Referencias Bibliográficas.....	67
Anexos.....	71

Introducción

Las Tecnologías de la Información y la Comunicación (TIC) son un conjunto de técnicas, desarrollos y dispositivos avanzados que integran funcionalidades de almacenamiento, procesamiento y transmisión de datos [1]. La tecnología digital unida a la aparición de ordenadores cada vez más potentes ha permitido a la humanidad progresar rápidamente en la ciencia y la técnica desplegando sus armas más poderosas: la información y el conocimiento [2].

La informática es parte de las TIC, ya que esta es la ciencia encargada del tratamiento de la información. Esta procesa un conjunto de datos de entrada para obtener datos finales. El tratamiento que realiza la informática a la información depende de cuatro componentes [3]:

- *Hardware*: son los componentes físicos de la computadora que ayuda a obtener información.
- *Software*: es un componente intangible que sirve para manejar la información.
- *Wetware*: son las personas, usuarios que crean la información y en su momento la solicitan o necesitan.
- *Datos*: es el producto de la relación entre el *hardware* y el *software*.

Un sistema operativo es un grupo de programas de proceso con las rutinas de control necesarias para mantener continuamente operativos dichos programas. Su objetivo primario es optimizar todos los recursos del sistema para soportar los requerimientos. Este se puede clasificar en el conjunto de *software* para computadoras de la siguiente manera: programas de sistema que controlan la operación de la computadora en sí o programas de aplicación que resuelven problemas para los usuarios [4].

Con el paso de los años los usuarios se han visto en la necesidad de migrar de un sistema operativo a otro, debido a que el utilizado quede sin soporte, salga al mercado uno mejor, o simplemente esté

instalado en el nuevo *hardware* que fue comprado. Uno de los principales obstáculos al migrar de un sistema operativo a otro no es la complejidad del nuevo sistema, sino la familiaridad que el usuario ha acumulado a lo largo de los años con el anterior.

La migración a aplicaciones de código abierto en Cuba es necesariamente un factor clave para la seguridad y soberanía tecnológica de la nación. A partir del Acuerdo 084/2004 del Consejo de Ministros se orienta la migración paulatina de los Organismos de la Administración Central del Estado (OACE) a *software* de fuente abierta. El Ministerio de la Informática y las Comunicaciones (MIC)¹ como ente rector creó una estructura nacional para impulsar la tarea, constituida por el Grupo Ejecutivo Nacional y cuatro subgrupos asociados, entre los que se encuentra el Grupo Técnico Nacional (GTN, en lo adelante), encargado de crear el sistema operativo cubano y definir los lineamientos y estrategias que deberán seguirse [5].

El proceso de migración se apoya en gran medida por el Centro de *Software* Libre (CESOL) de la Universidad de Ciencias Informáticas (UCI), en el cual se ha desarrollado la distribución cubana GNU/Linux Nova; además el centro CESOL forma parte del GTN. Al centro CESOL pertenece el departamento de Servicios Integrales en Migración, Asesoría y Soporte (SIMAYS), el cual dirige varios procesos de migración hacia *software* libre. En la ejecución de estos procesos una de las tareas que se realiza es el traspaso de los datos de los usuarios y configuraciones de aplicaciones en el sistema privativo hacia la distribución libre instalada.

La migración hacia tecnologías libres se ha realizado en diversas empresas e instituciones del país tales como: Empresa de Proyectos (EPRO), Empresa Nacional de Investigaciones Aplicadas (ENIA), Diseño Ciudad Habana (DCH), Empresa Constructora de Obras de Arquitectura e Industriales No. 3 (ECOAIND 3), Centro de Cibernética Aplicada a la Medicina (CECAM), entre otras. Esta migración se rige por la Guía Cubana de Migración a código abierto y la misma está constituida por tres etapas. En la etapa

1 Ministerio de la Informática y las Comunicaciones (MIC): En la actualidad es Ministerio de Comunicaciones (MINCOM).

preparación se ejecutan todas las tareas de diagnóstico de los procesos, personas y tecnología de la entidad, además se realizan tareas de análisis de la información recuperada y se emite el Plan de Migración Institucional. La segunda etapa se titula **ejecución** y comprende las actividades necesarias para la migración definitiva de usuarios y tecnologías de la institución, e incluye la migración de los servicios telemáticos y las estaciones de trabajo. Finalmente, la etapa **consolidación** comprende tareas destinadas a garantizar el soporte técnico a los usuarios y la infraestructura [5].

En la actualidad el traslado de los datos de los usuarios y configuraciones de aplicaciones en el proceso de migración del sistema operativo, *Windows* a Linux, se realiza de forma manual, haciendo uso de algún dispositivo de almacenamiento externo, para que luego de instalada la distribución libre sean situados los datos en la ubicación homóloga. Este proceso ocasiona un uso excesivo del tiempo dedicado a esta tarea. Aunque el especialista en migración, quien es la persona encargada de la ejecución de dicho proceso, sea una persona responsable, existe la posibilidad de obviar algún paso en la ejecución del proceso y por consiguiente puede ocasionar que los datos estén incompletos. Este es el punto más frágil de la migración, ya que es donde se interactúa con los datos de los usuarios y un error podría provocar inconformidad de estos.

Partiendo de la problemática anterior se presenta el siguiente **problema de investigación**: ¿Cómo garantizar el traslado de los datos de los usuarios y configuraciones de aplicaciones en el proceso de migración de las estaciones de trabajo de *Windows* XP, 7 y 8 a GNU/Linux Nova, de forma que se asegure la integridad de los datos y se disminuya el tiempo empleado?

Para orientar la investigación se identifica como **objeto de estudio**: la migración de datos de los usuarios y configuraciones de aplicaciones en estaciones de trabajo.

La presente investigación tiene como **objetivo general** desarrollar una herramienta que garantice el traslado de los datos de los usuarios y configuraciones de aplicaciones en el proceso de migración de las estaciones de trabajo de *Windows* XP, 7 y 8 a GNU/Linux Nova para asegurar la integridad de los datos y disminuir el tiempo empleado.

Enmarcado en el **campo de acción**: las herramientas de migración de datos de los usuarios y configuraciones de aplicaciones para estaciones de trabajo de *Windows* XP, 7 y 8 a GNU/Linux.

Para el cumplimiento del objetivo general anteriormente planteado se trazan los siguientes **objetivos específicos**:

1. Analizar la migración de datos de los usuarios y configuraciones de aplicaciones hacia GNU/Linux.
2. Diseñar una aplicación informática para la migración de datos de *Windows* XP, 7 y 8 a GNU/Linux Nova.
3. Implementar la aplicación propuesta.
4. Probar la solución implementada.

Para darle cumplimiento a los objetivos específicos se definen las siguientes **tareas de investigación**:

1. Identificación de las principales herramientas que posibilitan la migración de datos.
2. Comparación de las herramientas identificadas.
3. Definición de los procedimientos para migrar los datos de los usuarios y configuraciones de aplicaciones en el proceso de migración de *Windows* XP, 7 y 8 a GNU/Linux Nova.
4. Definición de los requisitos funcionales y no funcionales de la herramienta a implementar.
5. Diseño de la arquitectura del sistema.
6. Codificación de las funcionalidades diseñadas.
7. Diseño y ejecución de casos de prueba a la herramienta implementada.

Se define como **idea a defender**: El desarrollo de una herramienta que garantice el traslado de los datos de los usuarios y configuraciones de aplicaciones en el proceso de migración de las estaciones de trabajo de *Windows* XP, 7 y 8 a GNU/Linux Nova permitirá asegurar la integridad de los datos y disminuir el tiempo empleado.

Para el desarrollo de la investigación se proponen los siguientes **métodos científicos**:

Método Teórico:

- **Analítico-Sintético:** Se utilizó para el análisis de un grupo de herramientas que posibilitan la migración de datos y configuraciones. Las herramientas se analizaron de manera independiente, se estudiaron cada una de ellas y se sintetizaron sus contenidos para una mejor comprensión.

Métodos Empíricos:

- **La observación:** Se utilizó para analizar cómo trabajan las principales herramientas que posibilitan la migración de datos y configuraciones, y por consiguiente se obtuvieron conocimientos que fueron usados en la implementación.
- **La entrevista:** Se utilizó para adquirir conocimientos importantes sobre el proceso de migración de *Windows* XP, 7 y 8 a GNU/Linux Nova, a partir de la aplicación de la misma a varios especialistas que han participado en procesos de migración. La información obtenida se utilizó para la definición de los requisitos de la herramienta a implementar.

La investigación está estructurada en tres capítulos que a continuación se describen:

Capítulo 1: “Fundamentación Teórica”

En este capítulo se presentan los resultados de la investigación realizada a las herramientas de migración de datos de los usuarios y configuraciones de aplicaciones para estaciones de trabajo, que poseen características similares o desempeñan una labor semejante a la herramienta que se desea desarrollar. También se definen la metodología, herramientas y tecnologías, que se emplean en el desarrollo de la aplicación.

Capítulo 2: “Análisis, diseño e implementación de la solución propuesta”

En este capítulo se describe la solución propuesta. Se presentan las características y funcionalidades de la herramienta a partir de los requisitos funcionales y no funcionales obtenidos. Además se realizan los diagramas correspondientes al diseño e implementación de la misma.

Capítulo 3: “Pruebas a la herramienta propuesta”

En este capítulo se realizan las pruebas necesarias para comprobar que la herramienta cumple con las funcionalidades previstas y el objetivo general de la presente investigación.

Es presente trabajo contiene además: **Conclusiones, Recomendaciones, Referencias Bibliográficas y Anexos.**

Capítulo 1: *Fundamentación teórica*

Introducción

En el presente capítulo se precisan elementos teóricos que sustentan la investigación y el desarrollo del tema propuesto, a través del estudio y análisis de soluciones existentes. Se describen las tecnologías, herramientas, lenguajes y la metodología de desarrollo que se utiliza para el análisis e implementación del sistema.

1.1 Conceptos asociados

En este apartado se plantean algunos conceptos que son necesarios dominar al hablar de migración de datos y configuraciones. La investigación está asociada a una herramienta que apoye a la Guía Cubana de Migración, por lo que es necesario el empleo de las definiciones esclarecidas por la misma.

Migración a Código Abierto: Los componentes de la migración no pueden verse por separado uno de otro, tecnología y personas (ver Figura 1) conforman el escenario de cualquier proceso de migración a código abierto. A partir de tal relación se puede definir que la migración no es solo cambiar un sistema operativo por otro, o eliminar y colocar un conjunto de nuevas aplicaciones, va más allá de eso, los cambios más grandes están en la forma de pensar de las personas. La migración es un proceso ordenado donde las personas están convencidas de las ventajas de implantación de las nuevas tecnologías y apoyan dicho proceso para el desarrollo exitoso del mismo [6].

Migración social: se está en presencia de este proceso cuando las personas son sometidas a un cambio en su forma de pensar, rompiendo el esquema seguido por las herramientas privativas de uso y adaptándose a la “filosofía” asociada al *software* libre y código abierto. Cuando dicha componente de la migración se encuentre en la etapa de satisfacción, entonces los esfuerzos dedicados al proceso pueden minimizarse [6].



Figura 1: Componentes de la migración.

Migración técnica: es la componente relacionada con el cambio de la tecnología en la institución, dicha migración requiere como paso imprescindible para su cumplimiento el cambio de las aplicaciones privativas de la entidad por herramientas libres en su mayor por ciento posible, se debe tener en cuenta que la migración puede no se realice de manera total por el uso de algunas aplicaciones que no posean una alternativa libre a corto plazo, en tal caso se estaría en presencia de una migración técnica de forma parcial. La migración técnica (o tecnológica) requiere de un conjunto de conocimientos centrados en los sistemas, sus configuraciones y uso que requiere de personal especializado en el uso de GNU/Linux mientras se ejecute el proceso [6].

Especialista en migración: es el encargado de ejecutar la mayoría de las actividades del proceso de migración descritas en el plan. Deben poseer sólidos conocimientos sobre la metodología de migración y

saber usar todas las aplicaciones que automatizan el proceso [7].

Función *hash*: es un algoritmo irreversible que permite generar, a partir de un mensaje de entrada, resúmenes que representen de manera casi unívoca a un mensaje (archivo o dato). Por lo general, también se le da el nombre de *hash*, *fingerprint* o *digest* al resultado de esa operación [8].

Colisión: cuando dos objetos diferentes generan igual *hash* [8].

Integridad: el autor de la presente investigación se refiere a integridad de los datos como la característica que muestra que los datos transferidos son exactos, evidenciada por la ausencia de los datos alterados entre los transferidos.

Luego de la definición de los conceptos principales para una mejor comprensión en la investigación es necesario profundizar en otros aspectos esenciales de la Guía Cubana de Migración.

1.2 Etapas de la Guía Cubana

La guía cubana de migración está constituida por tres etapas (ver Figura 2) a lo largo de la ejecución del proceso [5]. La investigación se centra en la migración del escritorio de los usuarios establecida en la etapa de **Ejecución**, las actividades a ejecutar en este subproceso se pueden apreciar en la Figura 3.

La actividad de instalar aplicaciones libres sobre el entorno privativo es fundamental, debido a que permite a los usuarios conocer las tecnologías libres en el sistema operativo que ya dominan. En dicha actividad no se desinstalan las herramientas privativas, así los usuarios se forman y se capacitan interactuando con ambas tecnologías, ya sean privativas o libres. Además en la guía se definen un conjunto de alternativas libres que debe emplear el especialista de migración.

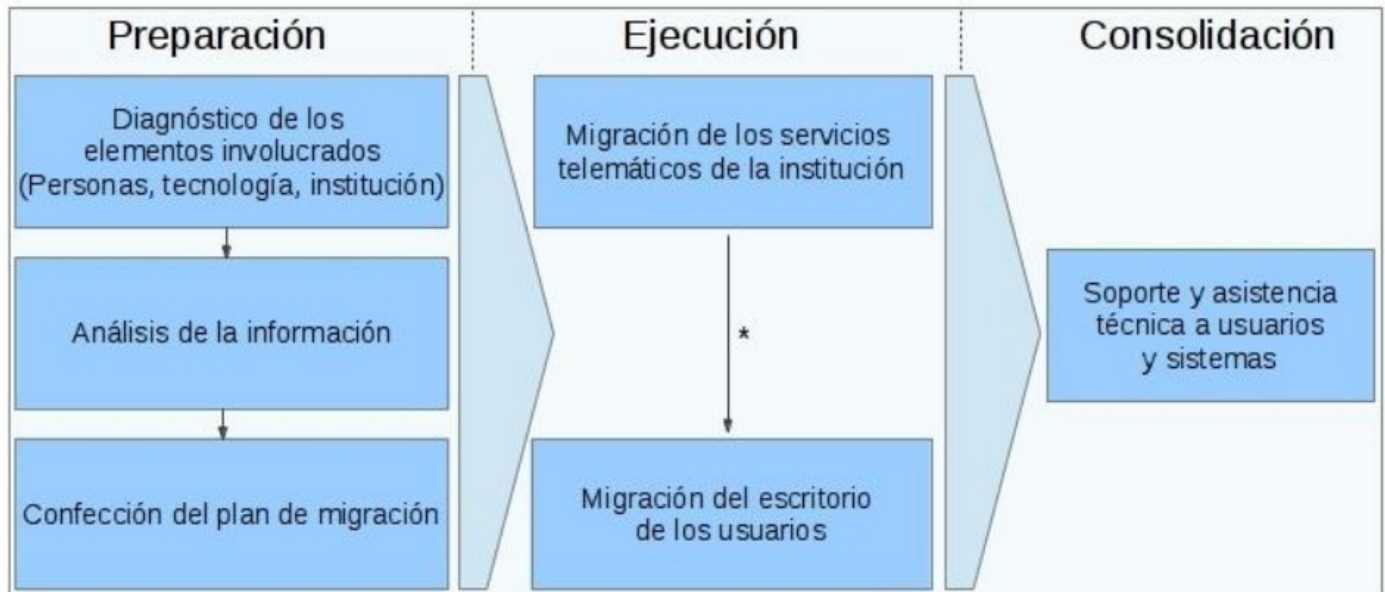


Figura 2: Etapas de la migración.

La actividad **migrar el sistema operativo** se enmarca en la sustitución de Microsoft *Windows* y permite instalar un sistema operativo GNU/Linux u otro libre. El proceso de cambiar el sistema requiere [5]:

1. Realizar un diagnóstico para detectar los archivos importantes existentes en cada ordenador de los usuarios que deban ser resguardados.
2. Realizar una salva de respaldo de dichos archivos en otra computadora o en un servidor centralizado.
3. Formatear e instalar el nuevo sistema.
4. Configurar el nuevo sistema, manteniendo la configuración de la red (número de IP, recursos compartidos). Esta información es extraída del diagnóstico realizado en la etapa de preparación de la migración.

5. Restaurar los archivos guardados en el paso 2.
6. Realizar las configuraciones y ajustes necesarios para el correcto funcionamiento del sistema, por ejemplo: montar particiones de forma permanente, configurar el cliente de navegación web, cliente de correo y mensajería instantánea, entre otras.

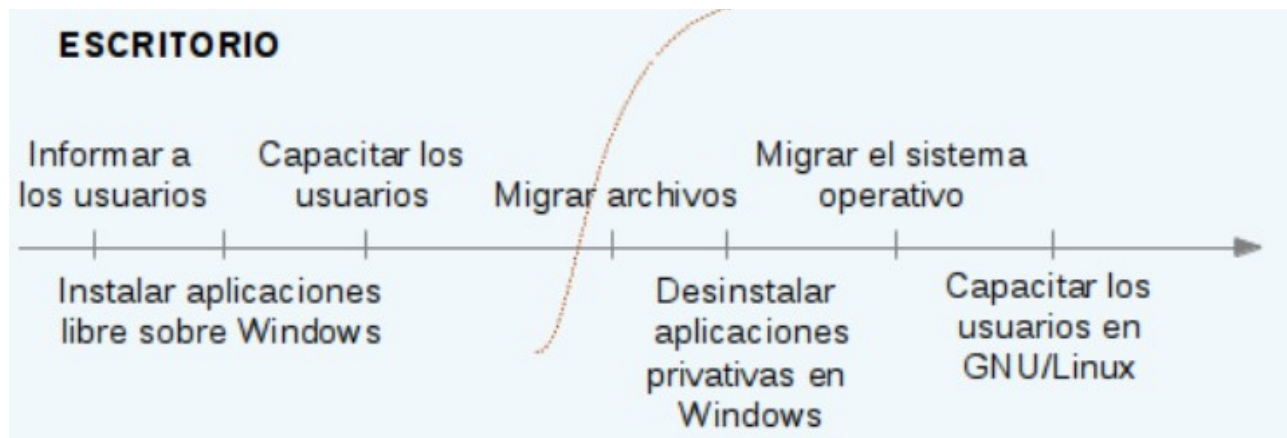


Figura 3: Actividades de la migración de escritorio.

1.3 Características de los sistemas operativos

Los sistemas operativos *Windows* y *GNU/Linux* poseen varias diferencias algunas de ellas son: el tipo de sistema de archivo y la manera de organizar sus carpetas y archivos. A continuación se explican algunas características de estos sistemas, enfocados a la necesidad de la investigación.

1.3.1 GNU/Linux

El Proyecto GNU es un esfuerzo de varios programadores y desarrolladores para crear un sistema operativo libre, Unix². GNU es un acrónimo recursivo que significa *GNU No es Unix*, ya que es similar a Unix pero no contiene ningún código de Unix y es gratis. La fundación GNU ve “libre” como algo más que el significado económico de libre: el *software* debería ser libre de utilizar para cualquier fin, libre para

² **UNIX:** es un sistema operativo de propósito general, multiusuario, e interactivo para las computadora

estudiar y modificar el código fuente y el comportamiento, la libertad de copiar y libre de distribuir los cambios realizados [9].

Esta idea del *software* libre es un pensamiento noble que está activo en la mente de muchos programadores: por lo tanto, muchos títulos de *software* están disponibles gratuitamente. Un *software* generalmente es acompañado de una licencia que explica lo que se puede y no se puede hacer con él (también conocida como el "*End User License Agreement*", en español Acuerdo de licencia de usuario final); el *Software Libre* tiene una licencia de este tipo que realmente permiten la mayoría de las cosas en lugar de negarlo, a diferencia de los EULA³. Un ejemplo de dicha licencia es GPL - Licencia Pública General de GNU [9].

El sistema de archivos de Linux es un árbol estructurado jerárquicamente, donde cada lugar tiene su significado. La estructura del sistema de archivos ha sido estandarizada a través del estándar de jerarquía del sistema de archivos que se puede apreciar en la Figura 16. Últimamente cada vez más distribuciones están haciendo un pequeño cambio hacia su diseño de sistema de archivos por lo que la norma está en la necesidad de una reforma. Por supuesto, un sistema de archivos siempre se almacena en los medios de comunicación, ya sea un disco duro, un CD (disco compacto) o un fragmento de la memoria [10].

Un sistema de archivos es la estructura subyacente que un equipo usa para organizar los datos de un disco duro. Si está instalando un disco duro nuevo, se tienen que crear las particiones y formatear estas empleando un sistema de archivos para poder comenzar a almacenar datos o programas [11].

El *kernel* de Linux tiene una buena arquitectura de los sistemas de ficheros, el trabajo interno se basa en una abstracción de un sistema virtual (*VFS, Virtual File System*), que puede ser adaptada fácilmente a cualquier sistema real [12].

Algunos de los sistemas de archivos más usados por Linux son Fat16, Fat32, NTFS, ext2, ext3, ext4,

³ **EULA:** Son las siglas de *End-User License Agreement* o traducido al español Acuerdo de Licencia con el Usuario Final, que son las condiciones o limitaciones que se deben aceptar para poder utilizar ese programa, aplicación, juego o producto.

reiserFS, JFS, XFS. De estos en el único que no se puede instalar Linux es el NTFS. No se recomienda instalar Linux en cualquier tipo de sistema de archivos FAT [13].

GNU/Linux cuenta con una estructura jerárquica para organizar sus directorios como se muestra en la Figura 16, en la cual todos los directorios cuentan con una misma raíz. El carácter “/” representa la raíz y además el separador de directorios. Los directorios y archivos son sensibles a mayúsculas y minúsculas, lo que quiere decir que *hola* y *Hola* tienen nombres distintos, y pueden colgar del mismo directorio. La carpeta */home* es la encargada de tener la carpeta personal de cada usuario creado en la computadora. La carpeta personal de un usuario está organizada en subdirectorios que permiten al usuario guardar sus archivos, además muchos programas guardan en dicha ubicación las configuraciones personales del usuario. El directorio */media* es el directorio donde se encuentran todas las unidades físicas que hay montadas⁴ en el sistema. El directorio */etc* es el que contiene todos los archivos de configuración locales, es decir, del ordenador donde se está ejecutando el sistema.

1.3.2 Windows

Los sistemas operativos *Windows* XP, 7 y 8 reconocen los sistemas de archivos FAT, FAT32, NTFS, el CDFS utilizado en CD-ROM, el UDF utilizado en DVD-ROM y el LFS para discos sin registro de arranque maestro [14].

Cuenta con una raíz para cada unidad lógica, ya sea dispositivo externo, unidades de red o particiones, a la cual va a estar asignada una letra en el rango A-Z y cada unidad está separada una de la otra. En la partición del sistema se encuentra un directorio en el que hay un perfil para cada usuario del sistema, es el directorio *Users* en *Windows* 7 y 8, y *Documents and Settings* en *Windows* XP. Dentro del perfil del usuario, carpeta que tiene el nombre del usuario y a la que solo puede acceder él y los administradores, existe una carpeta que se denomina *AppData* en la que se guardan configuraciones de aplicaciones entre otras cosas; también dentro de la carpeta personal del usuario se encuentra una serie de carpetas que el

4 **Unidades físicas montadas:** se le llama así a la acción de asignarle una carpeta a dispositivo de almacenamiento externo, una partición, u otros.

usuario puede utilizar para guardar sus datos, entre estas están Mis Documentos, Mi Música, Mis Imágenes. En la Figura 15 se muestra la estructura de directorios de *Windows*.

1.4 Migración de datos de usuario y configuración de aplicaciones

La migración de datos y configuraciones es el proceso de mover datos y configuraciones de un sistema operativo a otro, o entre versiones de un mismo sistema. Si no se mueven los datos y las configuraciones de manera intencionada al cambiar de sistema, los usuarios invertirán mucho tiempo en restaurar sus preferencias, fondos de escritorio, configuraciones del navegador *web*, y en recuperar sus archivos. A continuación se describe el proceso y se definen procedimientos para migrar los datos de *Windows* a GNU/Linux Nova.

1.4.1 Descripción del proceso de migración de datos de usuario y configuraciones de aplicaciones de *Windows* XP, 7 y 8 a GNU/Linux Nova

El proceso de migración de datos de usuario y configuraciones de aplicaciones de *Windows* XP, 7 y 8 a GNU/Linux Nova se rige por la Guía Cubana de Migración elaborada por el autor Yoandy Pérez Villazón. Antes de hablar del proceso es necesario explicar que la etapa de Ejecución, etapa en la que se realiza el proceso en cuestión, propone tres fases: primero la Migración de los servicios telemáticos, luego la Instalación de *software* libre en *Windows*, estas fases tienen un nivel de solapamiento entre ellas, y para terminar, Instalación del sistema operativo GNU/Linux Nova. Durante la migración de las estaciones de trabajo se realizan las siguientes actividades: se informa al usuario del cambio a realizar, se instalan aplicaciones libres sobre *Windows*, se capacitan los usuarios, se migran los archivos, se desinstalan aplicaciones privativas en *Windows*, se cambia el sistema operativo por GNU/Linux Nova y se capacitan los usuarios en el nuevo sistema [5]. El proceso de migración de datos de usuarios y configuraciones de aplicaciones se encuentra en la actividad de migrar el sistema operativo, en la cual se salvan los datos que los usuarios necesiten, así como las configuraciones de las aplicaciones que deseen y puedan ser

utilizadas en GNU/Linux Nova.

1.4.2 Definición de los procedimientos para migrar los datos de los usuarios y las configuraciones de aplicaciones en el proceso de migración de *Windows* XP, 7 y 8 a GNU/Linux Nova

Paso genéricos para realizar la migración:

1. En el sistema operativo *Windows* se localiza y se copian los datos y configuraciones que se desea migrar.
2. Para realizar la salva de esta información se introduce el dispositivo de almacenamiento externo que se utiliza para migrar y en él se pega la información copiada en el paso anterior.
3. Luego, con el sistema operativo libre ya instalado se introduce el dispositivo de almacenamiento externo y se procede a ubicar la información en la carpeta homóloga.

A continuación se exponen las configuraciones específicas de varias aplicaciones que pueden ser migradas:

Mozilla Firefox: Es un navegador *web* en el que todos los cambios que se realizan en la página de inicio, las barras de herramientas, las contraseñas, complementos y los marcadores que se guardan, están almacenados en una carpeta especial denominada perfil. La carpeta perfil se encuentra ubicada en un lugar independiente de la ubicación en la que está instalado el *Firefox*, de forma que, si es necesario desinstalar el *Firefox* no se perderían las configuraciones realizadas y no hay que volver a instalarlo para borrar las configuraciones. La ubicación de la carpeta varía en función del sistema operativo (ver Tabla 1).

Tabla 1: Ubicación Mozilla Firefox.

Sistema Operativo	Ubicación
<i>Windows</i> XP	C:\Documents and Settings\ <nombreusuario>\Application Data\Mozilla\Firefox\Profiles</nombreusuario>

Windows 7 y 8	C:\Users\ <nombreusuario>\AppData\Roaming\Mozilla\Firefox\Profiles\</nombreusuario>
GNU/Linux	/home/<nombreusuario>/.mozilla

Mozilla Thunderbird: Este cliente de correo guarda información personal como mensajes, contraseñas y las preferencias del usuario en un grupo de archivos denominado perfil, se almacena en una ubicación independiente de la ubicación donde está instalado el *Thunderbird*. El perfil almacena dos tipos de elementos principalmente. En primer lugar, el correo electrónico y copias de los mensajes que se almacenan en el servidor del correo electrónico, dependiendo de la configuración de la cuenta. En segundo lugar, almacena cualquier cambio que se haga mientras se utiliza *Thunderbird*. Cuando se instala se crea un perfil denominado "predeterminado". Esa carpeta recibe un nombre en función de los siguientes parámetros:

<random_string>.<profile_name> donde <random_string> son ocho dígitos que *Thunderbird* crea aleatoriamente, y <profile_name> es el nombre que el usuario le asigna al perfil. La ubicación de la carpeta varía en función del sistema operativo (ver Tabla 2).

Tabla 2: Ubicación Mozilla Thunderbird.

Sistema Operativo	Ubicación
Windows XP	C:\Documents and Settings\ <nombreusuario>\Application Data\Thunderbird\Profiles</nombreusuario>
Windows 7 y 8	C:\Users\ <nombreusuario>\AppData\Roaming\Thunderbird\Profiles\</nombreusuario>
GNU/Linux	/home/<nombreusuario>/.thunderbird/

Internet Explorer y Microsoft Outlook: La presente investigación se centra en la actividad de migración del sistema operativo que se realiza luego de la actividad de instalación de aplicaciones libres sobre

Windows (ver Figura 3), siendo esta última en la que se instalan las aplicaciones *Mozilla Firefox* y *Mozilla Tunderbird* y es realizada la migración de las configuraciones de las homólogas privativas hacia estas; por lo que no es necesario incluir en esta investigación la migración del *Internet Explorer* y de Microsoft Outlook.

Pidgin: Es un cliente de mensajería instantánea y es la alternativa libre usada en el proceso de migración de las estaciones de trabajo. Este guarda la configuración de las cuentas de usuario en la carpeta *.purple* y la ubicación de la carpeta varía en función del sistema operativo (ver Tabla 3).

Tabla 3: Ubicación de Pidgin.

Sistema Operativo	Ubicación
Windows XP	C:\Documents and Settings\ <nombreusuario>\Application Data\purple</nombreusuario>
Windows 7 y 8	C:\Users\ <nombreusuario>\AppData\Roaming\purple</nombreusuario>
GNU/Linux	/home/<nombreusuario>/purple

Configuración del fondo del escritorio: El fondo de escritorio (también llamado fondo de pantalla o *wallpaper*) es la imagen sobre la cual se sitúan los iconos del escritorio. Es la que se ve continuamente cuando se trabaja con el ordenador, de ahí que los usuarios quieran mantenerlo cuando van a cambiar de sistema operativo. En la Tabla 4, se muestra la ubicación del archivo *TranscodedWallpaper.jpg* (para Windows). El proceso para restaurar la configuración guardada sería ejecutar el siguiente comando en la terminal **gsettings set org.gnome.desktop.background picture-uri** "Dirección de la ubicación del archivo", el usuario podrá disfrutar nuevamente de su fondo de pantalla.

Tabla 4: Ubicación de la configuración del fondo del escritorio.

Sistema Operativo	Ubicación
Windows XP	C:\Documents and Settings\ <nombreusuario>\Local Settings\Application Data\Microsoft\Wallpaper1.bmp</nombreusuario>

Windows 7 y 8	C:\Users\<<nombreusuario>\AppData\Roaming\Microsoft\Windows\Themes\Transc odedWallpaper o C:\Users\<<nombreusuario>\AppData\Roaming\ Microsoft\Windows\Themes\TranscodedWallpaper.jpg
---------------	---

Configuración de red: Se puede tener la red configurada de dos formas, por DHCP (protocolo de configuración dinámica de *host*), lo que significa que el *router*⁵ asignará una IP⁶ automáticamente junto al resto de la configuración y la otra opción es configurar una IP estática de forma que sea siempre la misma. La configuración por DHCP también es conocida como configuración dinámica. Para conocer la configuración de la red en *Windows* se ejecuta en la consola el comando **ipconfig/all**. Para restablecerla en GNU/Linux Nova, se establece de la siguiente manera:

En la terminal se edita el fichero **/etc/network/interfaces**:

En caso de ser estática, introducir los siguientes datos:

```

auto lo
iface lo inet loopback
auto ethX
iface ethX inet static
address XXX.XXX.XXX.XXX
netmask XXX.XXX.XXX.XXX
gateway XXX.XXX.XXX.XXX
  
```

Figura 4: Configuración estática.

A continuación se describen los parámetros:

- *localhost*, la X de *ethX* hay que sustituirla por el número de la tarjeta de red.

5 Router o enrutador: Es un dispositivo que proporciona conectividad a nivel de red. Su función principal consiste en enviar o encaminar paquetes de datos de una red a otra.

6 IP (*Internet Protocol*): Se trata de un estándar que se emplea para el envío y recepción de información mediante una red.

- *auto*, seguido de la tarjeta de red es para que esta arranque automáticamente al encender la computadora.
- *iface* sirve para especificar si la configuración IP será dinámica o estática.
- *address* se le asigna la dirección IP que tendrá el ordenador.
- *netmask* la máscara de la red.
- *gateway* la dirección de la puerta de enlace.

En caso de ser dinámica introducir los siguientes datos:

```
auto lo
iface lo inet loopback
auto ethX
iface ethX inet dhcp
```

*Figura 5: Configuración
dinámica.*

En ambos casos se debe ejecutar ***/etc/init.d/networking restart*** para reiniciar las interfaces de red del ordenador y aplicar los cambios.

Para restablecer la configuración de los servidores DNS (Sistemas de Nombre de Dominio), en la terminal se debe editar el archivo ***/etc/resolv.conf*** con los siguientes datos ***name server XXX.XXX.XXX.XXX***, donde XXX.XXX.XXX.XXX es el IP del servidor, se puede poner más de un servidor DNS poniendo en la línea de abajo *name server* asignándole la dirección IP alternativa.

Perfil de Usuario: Del perfil del usuario en *Windows* se debe copiar para el dispositivo que se va a utilizar para migrar el contenido las carpetas Descargas, Escritorio, Mis Documentos, Mis vídeos, Mis imágenes y Mi música, para luego de instalado el sistema operativo GNU/Linux Nova copiarlo en la ubicación análoga.

Nombre de la Computadora: Para obtener el nombre de la computadora en la consola del sistema *Windows* se ejecuta el comando **hostname** y para poner el nombre luego de instalado el sistema GNU/Linux Nova se debe editar en la terminal los archivos **/etc/hostname**, **/etc/hosts**, para terminar se debe reiniciar el servicio con el siguiente comando **letclinit.d/hostname restart**.

Datos de los Usuarios: Para copiar los datos del usuario que no se encuentren dentro del perfil de este, es necesario buscar su ubicación en el disco duro y copiarlos para el dispositivo de almacenamiento que se va a utilizar para migrar para luego de instalado el sistema GNU/Linux Nova puedan ser copiados a donde prefiera el usuario.

1.5 Herramientas para migrar los datos de los usuarios y configuraciones de aplicaciones

En esta sección se presentan algunas herramientas que posibilitan la migración de datos y configuraciones de aplicaciones. En casos específicos permiten realizar salvadas de seguridad.

Windows Easy Transfer: Es una herramienta destinada para transferir archivos y configuraciones que el usuario elige de un equipo a otro. Puede transferir correo electrónico, documentos, imágenes, música, vídeos, cuentas de usuario, favoritos de Internet, archivos en el escritorio. Se distribuye instalada en *Windows 8* y *Windows 7*, y para su uso es necesario que los equipos posean uno de estos dos sistemas y un dispositivo de almacenamiento USB (*Universal Serial Bus*) o que los equipos estén conectados a una misma red local. Esta herramienta realiza una serie de preguntas al usuario para saber qué transferir y en dónde realizar la salva [15].

Pcmover: Con esta herramienta se pueden transferir los datos a través una red local, un cable USB de Laplink⁷, un cable USB compatible con *Windows Easy Transfer* o un dispositivo de almacenamiento externo. Entre sus principales características se encuentran la de transferir programas instalados,

⁷ **Cable LaplinkK:** Es un cable utilizado para crear un conexión directa entre dos computadoras, sin necesidad de utilizar una tarjeta de red o un módem.

archivos, y cuenta con un asistente para la transferencia. Compatible con los sistemas *Windows*, tiene la opción de migrar uno o todos los usuarios a la vez. Además si el usuario utiliza Mac OS⁸ basados en Intel que ejecuten *Apple Boot Camp* o *Parallels Desktop* para Mac pueden utilizar *PCmover* para transferir archivos, configuraciones y aplicaciones desde sus antiguos ordenadores a sus nuevos ordenadores ejecutando Mac [16].

Asistente de Migración de *Windows*: Es utilizado para migrar archivos desde una computadora con Sistema operativo *Windows* a otra con Mac OS X Mountain Lion u Mac OS X Lion. Esta herramienta transfiere la información deseada hacia la ubicación adecuada, además transfiere únicamente los datos del usuario con que se ha iniciado en *Windows*, para transferir los datos de otro usuario hay que repetir el proceso iniciando sesión con su respectiva cuenta [17].

Asistente de Migración de *Guadalinex (AMIGU)*: Permite transferir los archivos y configuraciones de los sistemas operativos *Windows* 2000, XP, Vista y *Windows* 7 que estén instalados en la computadora. Este programa está distribuido bajo una licencia GPL⁹. Puede copiar documentos personales, favoritos y marcadores de navegadores web (Internet Explorer, Mozilla Firefox). Además permite transferir los datos y configuraciones de programas de correo (Outlook, Thunderbird) y configurar el fondo de escritorio. AMIGU funciona en cualquier distribución basada en Debian [18].

1.5.1 Comparativa de las herramientas

Luego del estudio realizado a las anteriores herramientas se concluye que, ***Windows Easy Transfer*, *Pc mover* y el Asistente de Migración de *Windows*** son privativas y no son compatible con GNU/Linux Nova, por lo que estas no cumplen con las necesidades requeridas, aunque pueden servir de referencia para la interfaz de usuario ya que realizan un proceso muy parecido al que se necesita realizar. Por otro lado **AMIGU** no es privativa y es compatible con GNU/Linux, pero no es multiplataforma y realiza el

8 **Mac Os o Sistema operativo Macintosh:** Es el sistema operativo creado por Apple para su línea de computadoras Macintosh.

9 GPL: Es una licencia de *software* libre.

proceso de una forma muy distinta al necesitado, para más detalles ver Tabla 5. Por lo antes planteado es necesario desarrollar una herramienta nueva.

Tabla 5: Tabla comparativa.

Características	Windows Easy Transfer	Pcmover	Asistente de Migración de Windows	de AMIGU
¿Migra los datos y configuraciones?	Sí	Sí	Sí	Sí
¿Es multiplataforma?	No	Sí	Sí	No
¿Garantiza la integridad de los datos?	Sí	Sí	Sí	No
¿Sus tecnologías son de código a abierto?	No	No	No	Sí

1.6 Lenguajes y herramientas

En la presente investigación se propone utilizar los lenguajes y herramientas necesarias para el desarrollo de la solución propuesta.

1.6.1 Lenguajes

Los lenguajes empleados en la propuesta de solución son:

Python: Es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “*Monty Python*”. Es un lenguaje con una sintaxis muy limpia y que favorece un código legible. Se trata de un lenguaje interpretado¹⁰ o de *script*, con tipado

¹⁰ **Lenguaje interpretado:** Un lenguaje interpretado o de *script* es aquel que se ejecuta utilizando un programa intermedio llamado intérprete.

dinámico¹¹, fuertemente tipado¹², multiplataforma y orientado a objetos. Al ser un lenguaje interpretado, implica el ahorro considerable de tiempo durante el desarrollo de un programa ya que no necesita de compilar ni enlazar. Permite escribir programas muy compactos y legibles. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa [19]. Este lenguaje de programación se emplea en su versión 2.7 para la implementación de la herramienta a proponer.

Lenguaje Unificado de Modelado (UML por sus siglas en inglés): Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de *software*. Se usa además para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas [20]. Este lenguaje de modelado se propone a utilizar en su versión 2.0 para realizar el diseño de las clases, diagrama de componentes y diagramas de paquetes.

1.6.2 Herramientas

Una vez seleccionados los lenguajes destinados al desarrollo del sistema, es preciso definir las herramientas necesarias que servirán de base para su construcción.

Scite: Es un editor de textos multiplataforma, sencillo, que soporta una gran variedad de formatos y lenguajes de programación con resaltado de código de sintaxis, permite ejecución de código dentro de su misma interfaz [21]. Esta herramienta se emplea en la investigación en su versión 3.0.2 como entorno de desarrollo.

Qt: Es una plataforma de desarrollo que incluye clases, librerías y herramientas para la producción de aplicaciones de interfaz gráfica. Es gratuito para aplicaciones de código abierto y las herramientas, sus librerías y clases, están disponibles para la plataforma Unix y la familia *Windows*. Estas librerías y clases están bien documentadas, y tienen una gran herencia de programación orientada [22]. Esta herramienta

11 **Tipado dinámico:** No es necesario declarar el tipo de dato que van a contener las variables.

12 **Fuertemente tipado:** No se permite tratar una variable como si fuera de un tipo distinto al que tiene.

se propone en el presente trabajo en su versión 4.0 para el desarrollo de la interfaz gráfica.

Qt-Designer: Es la herramienta de Qt para el diseño y la creación gráfica de interfaces de usuario (GUI), es multiplataforma y entre sus características principales cuenta con extensa documentación de sus componentes. Todas las propiedades establecidas en Qt-Designer se pueden cambiar de forma dinámica dentro del código [23]. Esta se propone en la presente investigación en su versión 4.0 para el diseño de la interfaz gráfica.

Visual Paradigm: Es una herramienta que utiliza UML. Está diseñada para una amplia gama de usuarios, incluyendo ingenieros de *software*, analistas de sistemas y negocios, y arquitectos de sistemas, o para cualquier persona que esté interesada en la construcción de forma fiable los sistemas de *software* a gran escala con un enfoque orientado a objetos [24]. Esta se emplea en su versión 8.0 para la realización de diagramas a través del UML.

1.7 Metodología a utilizar

Todo proceso de desarrollo de *software* es riesgoso y difícil de controlar si no se hace uso de una metodología de desarrollo, entonces se obtienen clientes y desarrolladores insatisfechos con el resultado.

Por ser la metodología definida por el departamento SIMAYS, para el desarrollo de la presente investigación se escoge la metodología SXP. SXP es un híbrido de metodologías ágiles que toma las mejores prácticas de las metodologías SCRUM y XP. Consta de 4 fases principales: **Planificación-Definición** donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto; **Desarrollo** es donde se realiza la implementación del sistema hasta que esté listo para ser entregado; **Entrega** puesta en marcha y por último **Mantenimiento**, donde se realiza el soporte para el cliente [25].

En cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto (LRP), definición de las Historias de Usuario, diseño,

implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. La metodología se caracteriza por ser iterativa e incremental con pequeñas mejoras unas tras otras, basada en Historias de Usuario (HU), está atenta al cambio y permite que el equipo de programación se mantenga en frecuente interacción con el cliente o usuario. Está especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, y se orienta a una entrega rápida de resultados [25].

Conclusiones parciales

En este capítulo se profundiza en algunos conceptos relacionados con la problemática en cuestión, los cuales serán necesarios para la definición de la propuesta de solución. Se realiza un análisis de algunas de las herramientas que hacen posible la migración de datos y configuraciones llegando a la conclusión de que no cumplen con las necesidades requeridas, ya que tres no son compatible con GNU/Linux y la otra realiza el proceso de una forma muy distinta. Se definen las herramientas y lenguajes que guiados por la metodología SXP permitirán una programación más rápida partiendo de iteraciones incrementales con ciclos cortos en el desarrollo de la aplicación.

Capítulo 2: Análisis, diseño e implementación de la solución propuesta

Introducción

En este capítulo se desarrolla el análisis, diseño y la implementación de la herramienta que se propone. Se identifican los estándares de codificación a seguir, entre otros puntos de vital importancia.

2.1 Propuesta de solución

Durante la presente investigación se realiza una entrevista a varios especialistas del departamento SIMAYS que han participado en el proceso de migración de estaciones de trabajo, con el fin de adquirir conocimientos sobre el proceso. El resultado de la entrevista se puede apreciar en la Tabla 6, esta se aplica a 8 especialistas en migración.

Tabla 6: Resultado de la entrevista.

Preguntas	Sí	No
¿Ha migrado la configuración de la red?	5	3
¿Ha migrado la configuración del fondo del escritorio?	5	3
¿Ha migrado la configuración del navegador web <i>Mozilla Firefox</i> ?	8	0
¿Ha migrado la configuración del navegador web <i>Internet Explorer</i> ?	7	1
¿Ha migrado la configuración del cliente de correo <i>Mozilla Thunderbird</i> ?	6	2
¿Ha migrado la configuración del cliente de correo <i>Microsoft Outlook</i> ?	5	3

¿Ha migrado la configuración del cliente de mensajería <i>Pidgin</i> ?	4	4
¿Ha migrado la configuración del nombre de la computadora?	4	4
¿Ha migrado el contenido de la carpeta Escritorio?	6	2
¿Ha migrado el contenido de la carpeta Mis Documentos?	8	0
¿Ha migrado el contenido de la carpeta Mi Música?	4	4
¿Ha migrado el contenido de la carpeta Mis Vídeos?	4	4
¿Ha migrado el contenido de la carpeta Descargas?	6	2
¿Ha migrado el contenido de la carpeta Mis Imágenes?	5	3
¿Ha migrado el contenido de una partición?	5	3

Además se puede añadir que uno de los especialistas sugiere que además del *pidgin* se puede migrar también el cliente de mensajería PSI. Aunque el que se ha estado utilizando en la tarea de instalación de aplicaciones libres sobre el sistema *Windows*, y el que debería estar instalado en la computadora en el momento de realizar la migración del equipo, es el *pidgin*.

Se propone desarrollar una herramienta multiplataforma que permita migrar los datos de los usuarios y las configuraciones de aplicaciones de *Windows* a GNU/Linux Nova. Entre las principales funcionalidades de la presente propuesta, y a partir del resultado de la entrevista, está la de migrar las configuraciones de aplicaciones tales como: *Mozilla Firefox*, *Mozilla Tunderbird*, *Pidgin* y Configuración de la red. Además debe ser capaz de migrar el perfil de usuario, así como las configuraciones de red y los datos de los usuarios. Con esta propuesta se proveerá de una herramienta que permita al especialista realizar de una

manera más rápida su trabajo, garantizando la integridad de los datos.

2.1.2 Descripción de la propuesta

Para migrar los datos y las configuraciones de aplicaciones de *Windows* XP, 7 y 8 a GNU/Linux Nova en el proceso de migración es necesario crear una salva de estos y luego restaurarlos (ver Figura 17). Para comenzar, el especialista ejecuta la aplicación, si se encuentra en el sistema operativo *Windows* XP, 7 u 8 se procede a seleccionar y salvar en un dispositivo de almacenamiento externo las configuraciones y datos que se deseen. En el caso que se encuentre en GNU/Linux Nova se comienza a restablecer las configuraciones y datos previamente salvados. Si los datos a salvar o a restablecer tienen una capacidad mayor que la del destino no debe dejar comenzar la copia, advirtiendo al usuario de este problema. Además, se propone después de realizada la salva o la reubicación de los datos, analizar los datos para comprobar la integridad, este proceso será abordado más adelante.

2.2 Lista de Reserva del Producto (LRP)

La LRP es una lista priorizada que define el orden por importancia del trabajo que se va a realizar en el desarrollo del proyecto. En la fase de inicio es muy difícil tener claro todos los requerimientos sobre el producto. Sin embargo, los más importantes suelen surgir en los comienzos del ciclo de desarrollo y por lo general son suficientes para una primera iteración [26].

La lista de reserva del producto se encuentra en constante cambio a medida que se vaya profundizando en el sistema hasta su completa implementación o por nuevas peticiones del cliente agregando nuevas funcionalidades, dejando aperturas para nuevas versiones con mejoras en su implementación. Se deja escrito correctamente los requisitos no funcionales que aunque no sean de gran impacto para la aplicación, se deben tener en cuenta para el soporte del sistema. Una buena documentación de estos requisitos no funcionales garantiza que en el futuro, en caso de mantenimiento del sistema, el tiempo que se requiera sea el mínimo o de menor costo.

Tabla 7: Lista de reserva del producto.

Asignado a	Ítem	Descripción	Estimación	Estimado por
Windows				
Prioridad		Alta		
Grabiél Montero	1.	Salvar datos que los usuarios tienen en la carpeta Escritorio.	0.25	Grabiél Montero
Grabiél Montero	2.	Salvar datos que los usuarios tienen en la carpeta Descargas.	0.25	Grabiél Montero
Grabiél Montero	3.	Salvar datos que los usuarios tienen en la carpeta Mi música.	0.25	Grabiél Montero
Grabiél Montero	4.	Salvar datos que los usuarios tienen en la carpeta Mis documentos.	0.25	Grabiél Montero
Grabiél Montero	5.	Salvar datos que los usuarios tienen en la carpeta Mis vídeos.	0.25	Grabiél Montero
Grabiél Montero	6.	Salvar datos que los usuarios tienen en la carpeta Mis imágenes.	0.25	Grabiél Montero
Grabiél Montero	7.	Salvar configuración del Mozilla Thunderbird.	0.25	Grabiél Montero
Grabiél Montero	8.	Salvar configuración del Mozilla Firefox.	0.25	Grabiél Montero
Grabiél Montero	9.	Salvar datos fuera del perfil.	0.25	Grabiél Montero
Prioridad		Media		
Grabiél Montero	10.	Salvar configuración del Pidgin.	0.25	Grabiél Montero
Grabiél Montero	11.	Salvar el fondo del escritorio.	0.25	Grabiél Montero

Grabiél Montero	12.	Salvar configuración de Red.	0.25	Grabiél Montero
Grabiél Montero	13.	Salvar nombre de la computadora.	0.25	Grabiél Montero
Linux				
Prioridad		Alta		
Grabiél Montero	14.	Restaurar en la ubicación homóloga el contenido de la carpeta Escritorio.	0.25	Grabiél Montero
Grabiél Montero	15.	Restaurar en la ubicación homóloga el contenido de la carpeta Descargas.	0.25	Grabiél Montero
Grabiél Montero	16.	Restaurar en la ubicación homóloga el contenido de la carpeta Mi música.	0.25	Grabiél Montero
Grabiél Montero	17.	Restaurar en la ubicación homóloga el contenido de la carpeta Mis documentos.	0.25	Grabiél Montero
Grabiél Montero	18.	Restaurar en la ubicación homóloga el contenido de la carpeta Mis vídeos.	0.25	Grabiél Montero
Grabiél Montero	19.	Restaurar en la ubicación homóloga el contenido de la carpeta Mis imágenes.	0.25	Grabiél Montero
Grabiél Montero	20.	Restaurar configuración del Mozilla Thunderbird.	0.25	Grabiél Montero
Grabiél Montero	21.	Restaurar configuración del Mozilla Firefox.	0.25	Grabiél Montero
Grabiél Montero	22.	Restaurar datos de fuera del perfil.	0.25	Grabiél Montero
Prioridad		Media		
Grabiél Montero	23.	Restaurar configuración del Pidgin	0.25	Grabiél Montero

Grabiél Montero	24.	Restaurar el fondo del escritorio.	0.25	Grabiél Montero
Grabiél Montero	25.	Restaurar configuración de Red.	0.25	Grabiél Montero
Grabiél Montero	26.	Restaurar nombre de la computadora.	0.25	Grabiél Montero
Prioridad		Baja		
Grabiél Montero	27.	Salir de la aplicación.	0.1	Grabiél Montero
Grabiél Montero	28.	Mostrar listado de los datos salvados.	0.05	Grabiél Montero
Requisitos no funcionales				
<i>Software</i>	29.	Ejecutar la aplicación en los sistemas operativos <i>Windows XP/7/8</i> y <i>GNU/Linux Nova</i> .		Grabiél Montero
	30.	Utilizar el lenguaje de programación <i>Python</i> .		Grabiél Montero

2.3 Historias de Usuarios

Las Historias de Usuarios son la técnica utilizada en SXP para especificar los requisitos de la aplicación. Son las tareas y funciones que el *software* debe hacer, escritas en lenguaje natural, sin un formato predeterminado, no excediendo su tamaño de unas pocas líneas de texto [25]. A continuación se muestran las principales:

Tabla 8: Historia de usuario: Salvar configuración.

Historia de usuario	
Número: HU-1	Nombre Historia de Usuario: Salvar configuración
Modificación de Historia de Usuario Número: 1	

Usuario : Grabiél Montero Moreno	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 4
Riesgo en Desarrollo: Alto	Puntos Reales: 4
<p>Descripción: Permite salvar la configuración de las siguientes aplicaciones: cliente de correo Mozilla Thunderbird, navegador web Mozilla Firefox, cliente de mensajería instantánea Pidgin, la configuración de red y el fondo del escritorio.</p> <p>El usuario una vez que ejecute la aplicación debe seleccionar las configuraciones que desea salvar, en la interfaz se muestran las siguientes configuraciones:</p> <p>Configuración de la red: es un campo de tipo <i>checkbox</i> y es un parámetro opcional. La configuración se salva de acuerdo al método de conexión, este puede ser dinámico o estático. En caso de ser estático se salva dirección IP, máscara, puerta de enlace y los DNS, de ser dinámico se salvan los DNS y el nombre de la computadora.</p> <p>Además se muestra un <i>treeview</i> que contiene la lista de los usuarios existentes en la computadora con sus respectivas configuraciones. Luego se debe seleccionar el usuario para desglosar sus configuraciones:</p> <p>Mozilla Firefox: es un campo de tipo <i>checkbox</i> y es un parámetro opcional.</p> <p>Mozilla Thunderbird: es un campo de tipo <i>checkbox</i> y es un parámetro opcional.</p> <p>Pidgin: es un campo de tipo <i>checkbox</i> y es un parámetro opcional.</p> <p>Fondo del escritorio: es un campo de tipo <i>checkbox</i> y es un parámetro opcional.</p> <p>Al finalizar se debe seleccionar la opción Siguiente de la interfaz.</p> <p>Observación: El usuario debe ejecutar la aplicación en el dispositivo de almacenamiento externo que se emplea en los procesos de migración. La aplicación debe ser ejecutada en el sistema operativo <i>Windows</i> XP, 7 o 8. Si las configuraciones de las aplicaciones no se encuentran en la ubicación definida en el capítulo 1, estas no se mostrarán.</p>	

Prototipo de Interfaz de usuario:

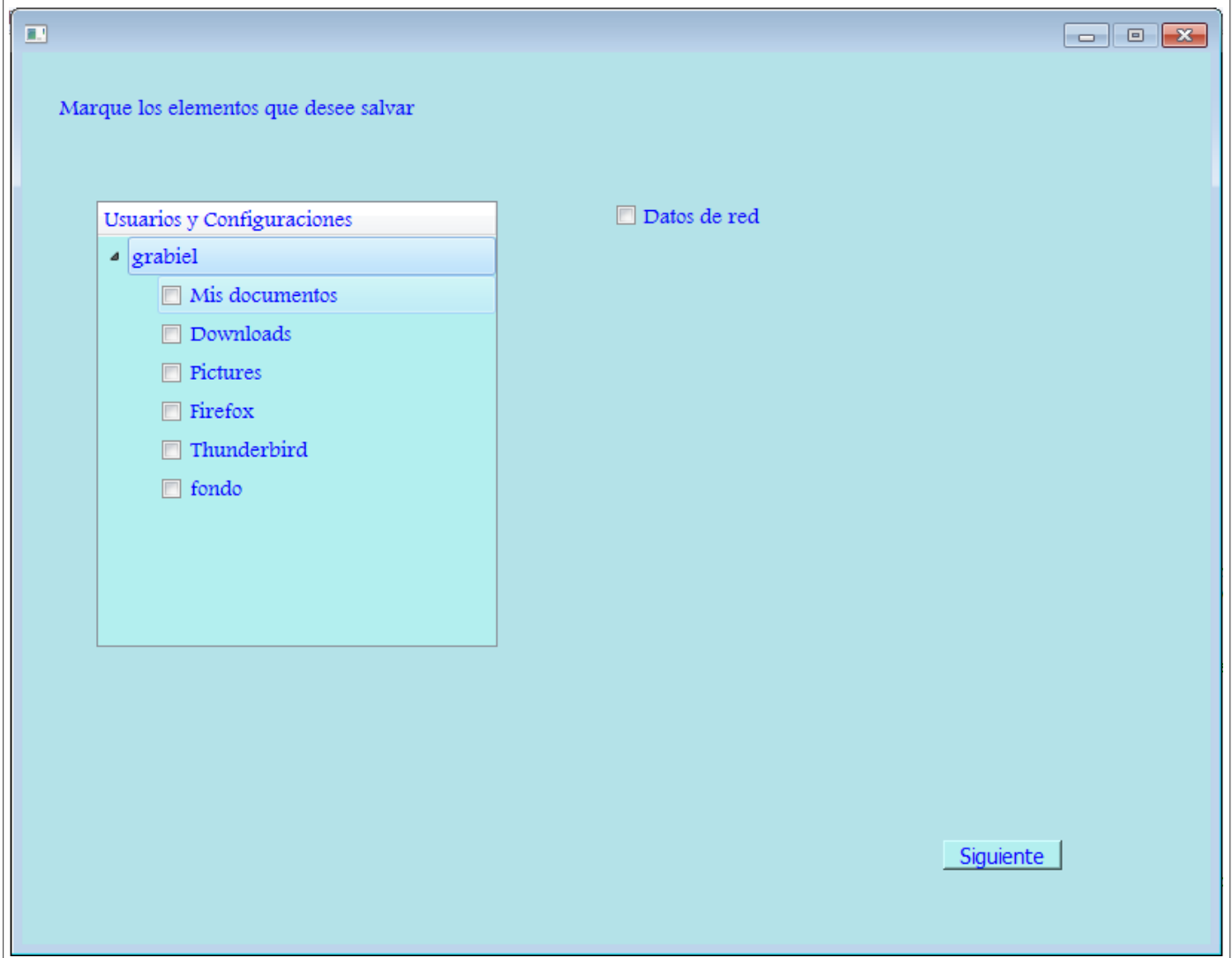


Tabla 9: Historia de usuario: Salvar datos del perfil del usuario.

Historia de usuario	
Número: HU-2	Nombre Historia de Usuario: Salvar datos del perfil del usuario
Modificación de Historia de Usuario Número: 1	
Usuario : Grabiél Montero Moreno	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 4
Riesgo en Desarrollo: Alto	Puntos Reales: 4
<p>Descripción: Permite salvar los datos de las carpetas Escritorio, Mis documentos, Mis imágenes, Mis vídeos, Mi música y Descargas contenidas en la carpeta del perfil del usuario.</p> <p>El usuario una vez que ejecute la aplicación debe seleccionar las carpetas que desea salvar. En la interfaz se muestra un <i>treeview</i> que contiene la lista de los usuarios existentes en la computadora. Luego se debe seleccionar el usuario para desglosar sus carpetas:</p> <p>Escritorio: es un campo de tipo <i>checkbox</i> y es un parámetro opcional.</p> <p>Mis documentos: es un campo de tipo <i>checkbox</i> y es un parámetro opcional.</p> <p>Mis imágenes: es un campo de tipo <i>checkbox</i> y es un parámetro opcional.</p> <p>Mis vídeos: es un campo de tipo <i>checkbox</i> y es un parámetro opcional.</p> <p>Mi música: es un campo de tipo <i>checkbox</i> y es un parámetro opcional.</p> <p>Descargas: es un campo de tipo <i>checkbox</i> y es un parámetro opcional.</p> <p>Al finalizar se debe seleccionar la opción Siguiente de la interfaz.</p>	
<p>Observación: El usuario debe ejecutar la aplicación en el dispositivo de almacenamiento externo que se emplea en los procesos de migración. La aplicación debe ser ejecutada en el sistema operativo <i>Windows XP, 7 o 8</i>. Si las carpetas del perfil del usuario no se encuentran en la ubicación definida en el capítulo 1, estas no se mostrarán.</p>	

Prototipo de Interfaz de usuario:

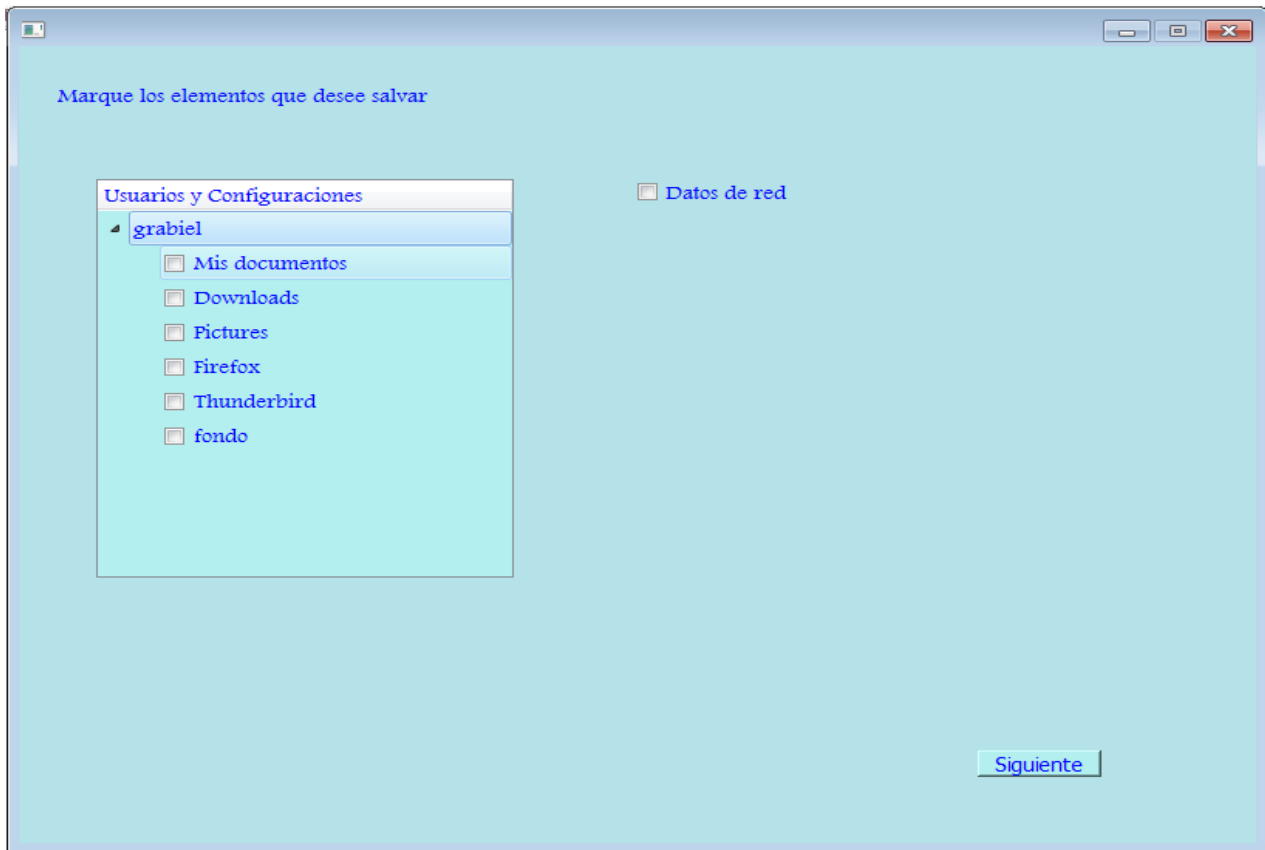


Tabla 10: Historia de usuario: Salvar datos de los usuario.

Historia de usuario	
Número: HU-3	Nombre Historia de Usuario: Salvar datos de los usuarios
Modificación de Historia de Usuario Número: 1	

Usuario : Grabiél Montero Moreno	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 4
Riesgo en Desarrollo: Alto	Puntos Reales: 4
<p>Descripción: Permite salvar los datos que tienen los usuarios en la computadora. Una vez que se seleccionen las configuraciones y los datos de la carpeta del perfil, se muestra una interfaz que contiene un <i>treeview</i> constituido por el árbol de directorio del sistema operativo. Luego se debe seleccionar los datos que se desean salvar mediante la selección de <i>checkbox</i>. Al finalizar se debe seleccionar la opción Siguiente de la interfaz.</p>	
<p>Observación: El usuario debe ejecutar la aplicación en el dispositivo de almacenamiento externo que se emplea en los procesos de migración. La aplicación debe ser ejecutada en el sistema operativo <i>Windows</i> XP, 7 o 8.</p>	
<p>Prototipo de Interfaz de usuario:</p>	

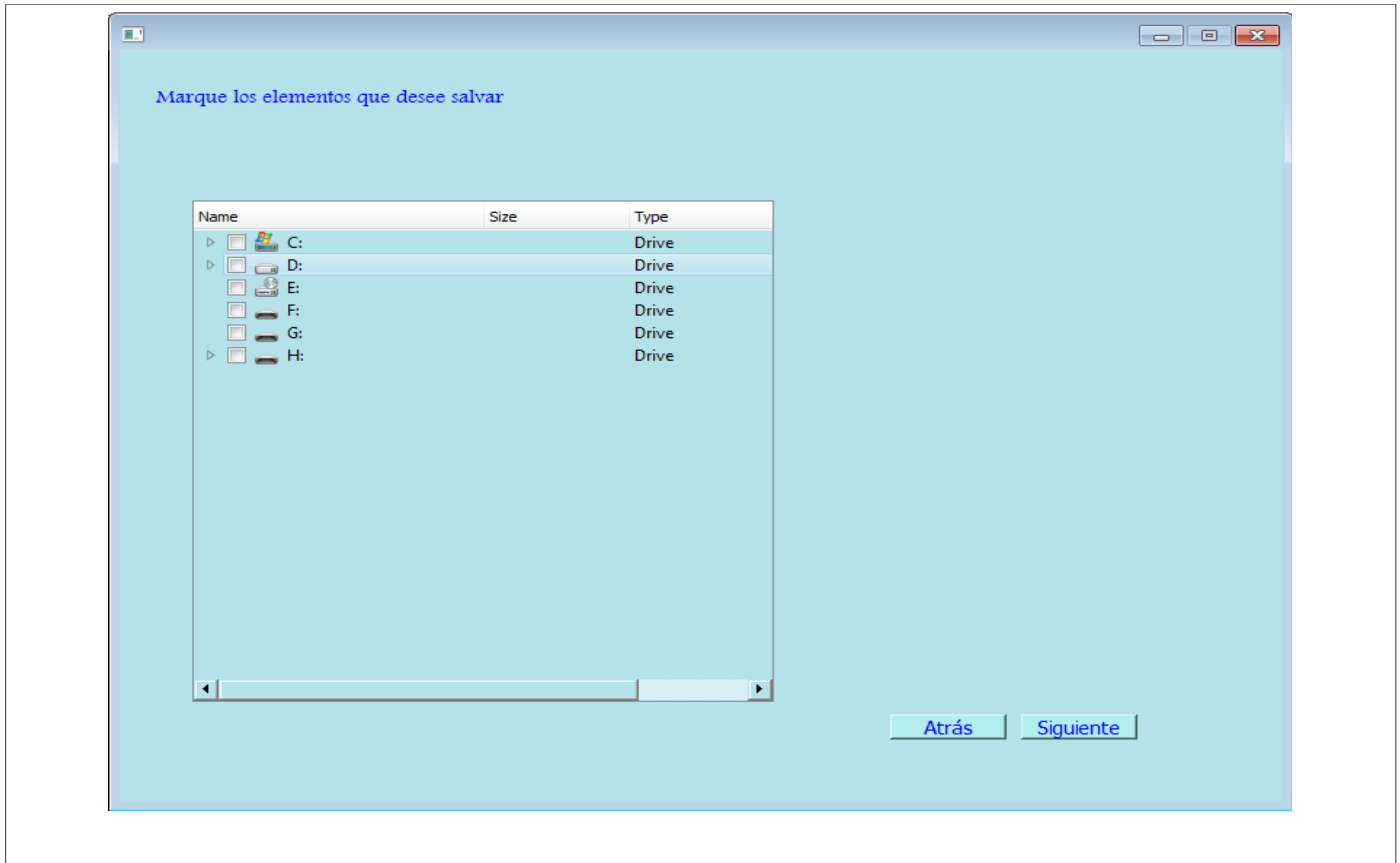


Tabla 11: Historia de usuario: Restaurar datos salvados.

Historia de usuario	
Número: HU-4	Nombre Historia de Usuario: Restaurar datos salvados
Modificación de Historia de Usuario Número: 1	
Usuario : Grabiel Montero Moreno	Iteración asignada: 2
Prioridad en Negocio: Media	Puntos Estimados: 2

Riesgo en Desarrollo: Alta	Puntos Reales: 2
Descripción: Permite al usuario escoger la ubicación para restaurar los datos salvados. El usuario una vez que ejecute la aplicación debe seleccionar del <i>combobox</i> la partición de la computadora donde desea restablecer los datos. Al finalizar se debe seleccionar la opción Siguiente de la interfaz.	
Observación: El usuario debe ejecutar la aplicación en el dispositivo de almacenamiento externo que se emplea en los procesos de migración. La aplicación debe ser ejecutada en el sistema operativo <i>Windows</i> XP, 7 o 8.	
Prototipo de Interfaz de usuario:	

Seleccione dónde desea reestablecer los datos adicionales

Name	Size	Type	Date Modified
▶ dev		Folder	18/06/15 21:
▶ etc		Folder	18/06/15 21:
▶ home		Folder	10/06/15 12:
initrd.img	18,7 MB	img File	11/06/15 07:
▶ lib		Folder	11/06/15 07:
lost+found		Folder	10/06/15 12:
▶ media		Folder	17/06/15 00:
mnt		Folder	05/02/13 10:
▶ opt		Folder	12/06/15 01:
▶ proc		Folder	18/06/15 21:
root		Folder	11/06/15 23:
▶ run		Folder	18/06/15 21:
▶ sbin		Folder	11/06/15 07:
selinux		Folder	05/03/12 08:

[Siguiente](#)

Tabla 12: Historia de usuario: Mostrar listado de los datos salvados.

Historia de usuario	
Número: HU-5	Nombre Historia de Usuario: Mostrar listado de los datos salvados
Modificación de Historia de Usuario Número: 1	
Usuario : Grabiél Montero Moreno	Iteración asignada: 2
Prioridad en Negocio: Baja	Puntos Estimados: 2
Riesgo en Desarrollo: Alta	Puntos Reales: 2
Descripción: Permite al usuario ver el listado de los datos salvados. El listado está constituido por la dirección y el peso del archivo.	
Observación: El usuario debe ejecutar la aplicación en el dispositivo de almacenamiento externo que se emplea en los procesos de migración. La aplicación debe ser ejecutada en el sistema operativo GNU/Linux Nova.	
Prototipo de Interfaz de usuario:	

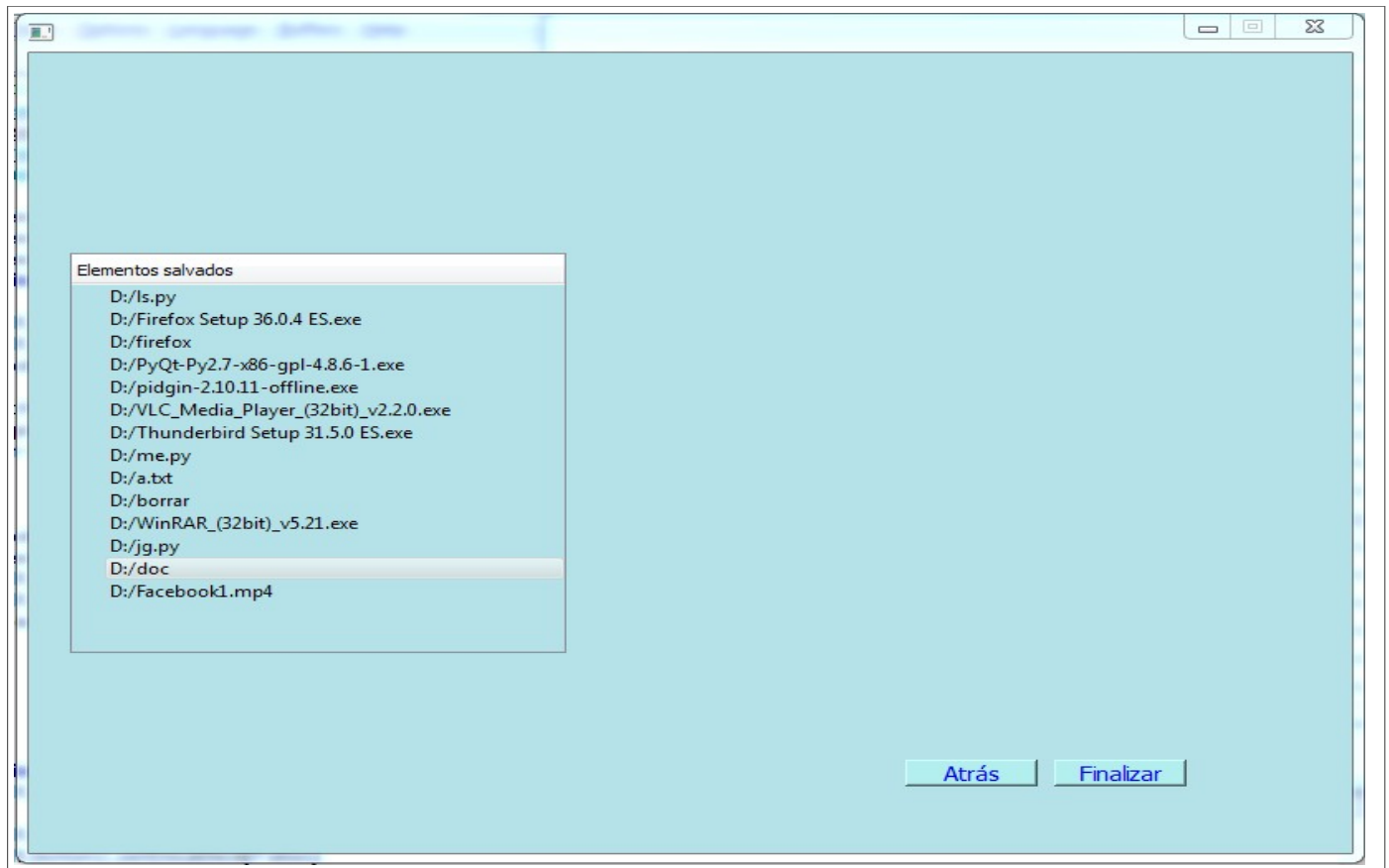


Tabla 13: Historia de usuario: Salir de la aplicación.

Historia de usuario	
Número: HU-6	Nombre Historia de Usuario: Salir de la aplicación
Modificación de Historia de Usuario Número: 1	
Usuario : Grabiél Montero Moreno	Iteración asignada: 2
Prioridad en Negocio: Baja	Puntos Estimados: 0.5
Riesgo en Desarrollo: Baja	Puntos Reales: 0.5
Descripción: Permite al usuario salir de la aplicación a través de dos vías. Una primera es, salir en cualquier instante seleccionando el botón Cerrar (X) de la esquina superior derecha y una segunda es, cuando se termine de salvar o restaurar los datos de los usuarios seleccionando la opción <i>Finalizar</i> .	
Observación: El usuario debe ejecutar la aplicación en el dispositivo de almacenamiento externo que se emplea en los procesos de migración. La aplicación debe ser ejecutada en el sistema operativo GNU/Linux Nova o <i>Windows XP, 7 o 8</i> .	
Prototipo de Interfaz de usuario:	

2.4 Arquitectura del sistema

La arquitectura de *software* de un programa es la estructura del sistema que incluyen los componentes de *software*, las propiedades visibles externamente de esos componentes y las relaciones entre ellos. Permite a un ingeniero de *software* analizar la efectividad del diseño para cumplir con los requisitos establecidos, considerar opciones arquitectónicas en una etapa en la que aún resulta relativamente fácil hacer cambios al diseño y reducir los riesgos asociados a la construcción del *software* [27].

La propuesta de solución se estructura con una arquitectura en capas. El estilo 2 capas se divide en las capas de presentación y lógica de negocio (ver Figura 6). La capa de presentación es la encargada de interactuar con el usuario, muestra el sistema, captura y comunica la información al usuario. Por otro lado, la capa de negocio (lógica del negocio) recibe las peticiones del usuario, es donde se establecen las

reglas que deben cumplirse, se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados [28].

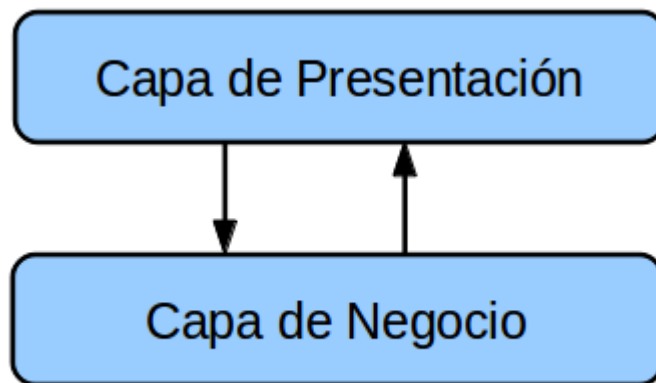


Figura 6: Arquitectura del sistema.

2.5 Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación [29]. Para ver el diagrama de clases de la propuesta de solución ver la Figura 7.

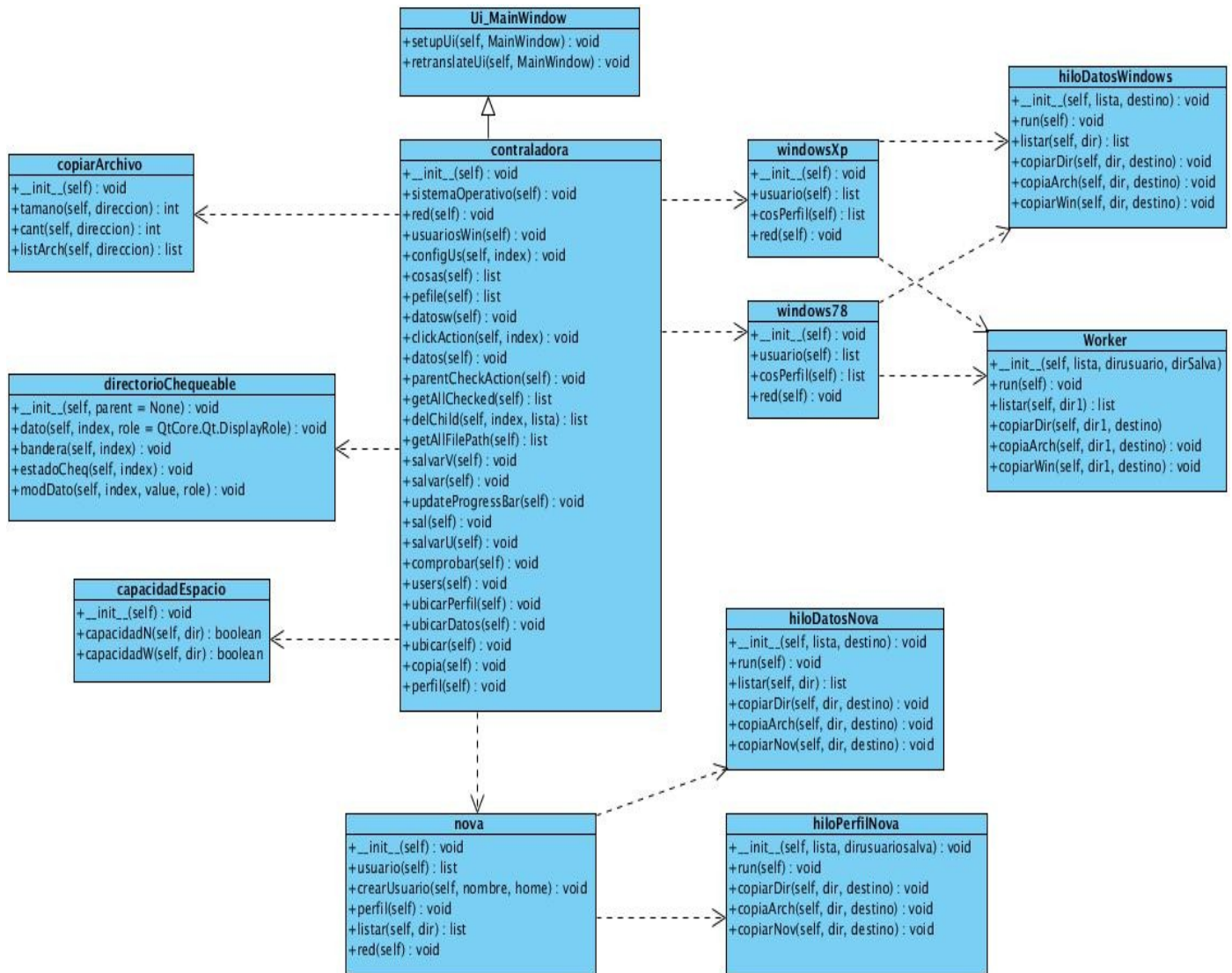


Figura 7: Diagrama de clases del sistema.

2.5.1 Principales clases

Se presenta la descripción de algunas de las principales clases:

1. **Ui_MainWindow:** Es la clase que contiene los elementos de la interfaz gráfica. Esta clase fue generada con el comando **pyuic4** y el archivo generado por el **Qt4-Designer**.
2. **windows78, windowsXp:** estas clases contienen los métodos necesarios para obtener los usuarios, sus datos y configuraciones dependiendo del sistema operativo.
3. **nova:** contiene los métodos necesarios para crear los usuarios y sus perfiles.
4. **controladora:** esta clase contiene las funcionalidades necesarias para obtener las peticiones de los usuarios y cumplir con las reglas del negocio.

2.6 Patrones de diseño

Los patrones de diseños brindan una solución ya probada y documentada a problemas de desarrollo de *software* que están sujetos a contextos similares [30].

Patrones GRASP:

Los patrones **GRASP**¹³ describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones [27].

Experto: este sugiere que se debe asignar las responsabilidades a aquellos objetos, o clases de objetos, que disponen de la información para hacerlo [27]. A continuación se muestra un ejemplo de su uso en la aplicación.

13 **GRASP:** es un acrónimo que significa *General Responsibility Assignment Software Patterns* (patrones generales de *software* para asignar responsabilidades)

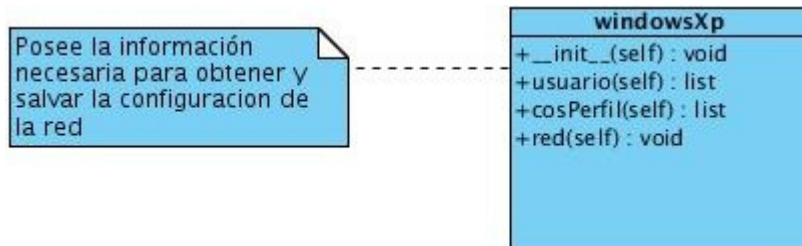


Figura 8: Patrón experto.

Creador: este guía la asignación de responsabilidades con la creación de objetos, asigna la responsabilidad de que una clase B cree un objeto de la clase A solamente cuando: B contiene, almacena o usa la clase A, cuando B es una agregación de A o cuando B tiene los datos de inicialización de A [27].

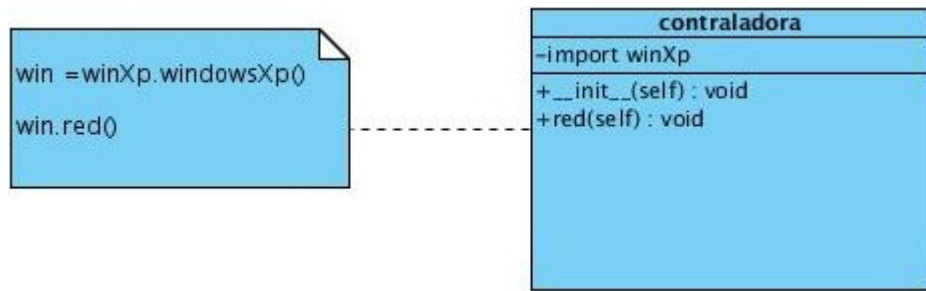


Figura 9: Patrón creador.

Controlador: Asigna la responsabilidad del manejo de los eventos de un sistema a una clase. La principal idea es crear una clase que implemente métodos dedicados a atender los eventos del sistema [27].

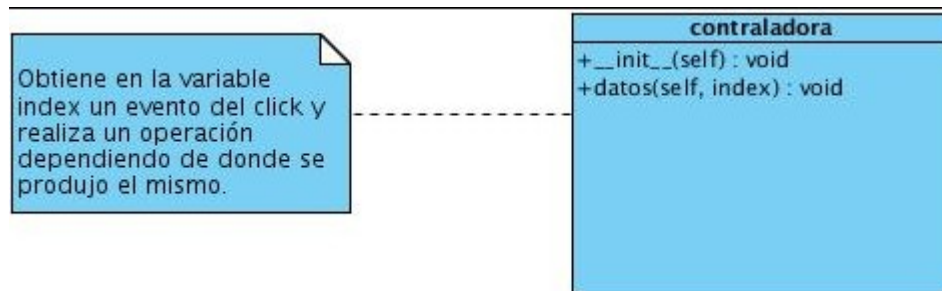


Figura 10: Patrón controlador.

Patrones GoF:

Los patrones “Banda de los Cuatro” (GoF, por sus iniciales en inglés *Gang of Four*), describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Tratan la relación entre clases, la combinación de clases y la formación de estructuras de mayor complejidad. Existen tres tipos de patrones: los de creación que abstraen el proceso de creación de instancias, estructurales que se ocupan de cómo las clases y objetos son utilizados para componer estructuras de mayor tamaño y de comportamiento que se refieren a los algoritmos y a la asignación de responsabilidades entre objetos [31].

Decorador:

Es un patrón de tipo estructural encargado de asociar responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades. Brinda una mayor flexibilidad que la herencia estática, permitiendo añadir una funcionalidad dos o más veces. Propicia concentrar en lo alto de la jerarquía de clases guiadas por las responsabilidades. De esta forma las nuevas funcionalidades se componen de piezas simples que se crean y se combinan con facilidad, independientemente de los objetos cuyo comportamiento extienden [32].

El uso de este patrón en la herramienta se evidencia en el uso del decorador **cursorAction**, el cual es el encargado de poner el cursor en modo de Cargando y retornarlo al modo normal cuando se ejecutan métodos que requieren algún tiempo de procesamiento visible al usuario.

```
def cursorAction():  
    def decorator(fun):  
        def inside(self, *arg):  
            self.main.setCursor(QtCore.Qt.WaitCursor)  
            fun(self, *arg)  
            self.main.setCursor(QtCore.Qt.ArrowCursor)  
  
        return inside  
  
    return decorator
```

Figura 11: Patrón decorador

2.7 Plan de entrega

De acuerdo con las funcionalidades descritas en las HU y en concordancia con la prioridad asignada para su realización, se planificaron 3 iteraciones que abarcan el desarrollo de la aplicación.

Tabla 14: Plan de entrega

Iteraciones	Descripción de la Iteración	Orden a Implementar	Duración Total
1	En esta iteración se realizarán las historias de usuarios que tienen que ver con los requerimientos de prioridad alta	HU-1	5 semanas
2	En esta iteración se realizarán las historias de usuarios que tienen que ver con los requerimientos de prioridad media	HU-2, HU-3	4 semanas
3	En esta iteración se realizarán las historias de usuarios que tienen que ver con los requerimientos de prioridad baja	HU-4	2 semanas

2.8 Tareas de Ingeniería

A continuación se relacionan algunas de las tareas de ingenierías correspondientes a las historias de usuario. Debido a que el proceso es cambiante estas deben ir adecuándose a las nuevas funcionalidades propuestas, por lo que esta es solo una planificación inicial.

Tabla 15: Tarea de ingeniería 1

Tarea de Ingeniería	
Número de la tarea: 1	Nombre de la historia de usuario: HU-1 y HU-2
Nombre Tarea: Identificar las configuraciones de aplicaciones y carpetas del perfil de usuario a migrar.	

Tipo de Tarea: Investigación	Puntos Estimados: 4
Fecha Inicio: 02/02/2015	Fecha Fin: 12/02/2015
Programador Responsable: Grabiél Montero Moreno	
Descripción: Permite obtener información acerca de las configuraciones de aplicaciones y carpetas del perfil de usuario mediante entrevistas a especialistas que han realizado el proceso de migración de sistema operativo.	

Tabla 16: Tarea de ingeniería 1.2

Tarea de Ingeniería	
Número de la tarea: 1.2	Nombre de la historia de usuario: HU-1 y HU-2
Nombre Tarea: Implementar la funcionalidad para migrar las configuraciones y carpetas del perfil de usuario.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 12/02/2015	Fecha Fin: 10/03/2015
Programador Responsable: Grabiél Montero Moreno	
Descripción: Permite programar la interfaz en la cual el usuario debe escoger las configuraciones y carpetas del perfil de usuario que desea salvar.	

Tabla 17: Tarea de ingeniería 2

Tarea de Ingeniería	
Número de la tarea: 2	Nombre de la historia de usuario: HU-3
Nombre Tarea: Implementar la funcionalidad para salvar los datos de los usuarios.	
Tipo de Tarea: Investigación	Puntos Estimados: 4
Fecha Inicio: 11/03/2015	Fecha Fin: 10/04/2015
Programador Responsable: Grabiél Montero Moreno	

Descripción: Permite programar la interfaz en la cual el usuario debe escoger los datos que desea salvar.

Tabla 18: Tarea de ingeniería 3

Tarea de Ingeniería	
Número de la tarea: 3	Nombre de la historia de usuario: HU-4
Nombre Tarea: Identificar la ubicación de las configuraciones y carpetas del perfil de los usuarios.	
Tipo de Tarea: Investigación	Puntos Estimados: 2
Fecha Inicio: 11/04/2015	Fecha Fin: 24/04/2015
Programador Responsable: Grabiél Montero Moreno	
Descripción: Permite identificar la ubicación de las configuraciones y carpetas salvadas.	

Tabla 19: Tarea de ingeniería 4

Tarea de Ingeniería	
Número de la tarea: 4	Nombre de la historia de usuario: HU-5
Nombre Tarea: Implementar la funcionalidad para mostrar el listado de los archivos copiados.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 25/04/2015	Fecha Fin: 8/05/2015
Programador Responsable: Grabiél Montero Moreno	
Descripción: Permite programar la interfaz en la cual se mostrará una lista de todos los archivos salvados de los usuarios.	

Tabla 20: Tarea de ingeniería 4.1

Tarea de Ingeniería	
Número de la tarea: 4.1	Nombre de la historia de usuario: HU-5

Nombre Tarea: Estudiar cómo se muestra la capacidad de los datos salvados.	
Tipo de Tarea: Investigación	Puntos Estimados: 1
Fecha Inicio: 9/05/2015	Fecha Fin: 12/05/2015
Programador Responsable: Grabiél Montero Moreno	
Descripción: Permite adquirir conocimientos para mostrar la capacidad de cada archivo salvado.	

2.9 Estándares de codificación

Las convenciones de código son un conjunto de reglas a seguir para escribir código fuente uniforme y legible. Estas reglas comprenden cómo se deben utilizar el espaciado, los saltos de línea, cómo se deben nombrar las variables, clases y ficheros y cómo se deben escribir instrucciones específicas del lenguaje de programación. Las convenciones ayudan a entender el código, por lo que resultan útiles a los desarrolladores en las actividades de mantenimiento a programas escritos por otros o por ellos mismos. La convención de código utilizada para el lenguaje *Python* en la implementación, es la establecida en el estándar PEP 08 con algunas variantes [33].

2.9.1 Estándar de código para la implementación de la herramienta

Para el cumplimiento de las Tareas de Ingenierías trazadas por el equipo de desarrollo, es de vital importancia que exista un entendimiento entre los programadores y el código de la aplicación. Se muestran algunos ejemplos del estándar de codificación escogido.

Nomenclatura de clases y métodos:

Los nombres de las clases y sus métodos deben estar escritos en minúscula, si tienen múltiples palabras la primera se escribe completa en minúscula y a partir de la segunda con la primera letra en mayúscula.

Ejemplo:

```
class clase:
```

```
class claseEjemplo:
```

```
def metodo():
```

```
def metodoEjemplo():
```

Argumentos de métodos:

Siempre usa *self* para el primer argumento de los métodos de instancia. Ejemplo:

```
def metodo(self):
```

Nomenclatura de variables:

Las variables deben contener solo minúsculas y si contienen múltiples palabras deben estar separadas por guion bajo. Ejemplo:

```
variable = 0
```

```
variable_ejemplo = 0
```

Indentación:

Se usarán 4 espacios para cada nivel de indentación. Ejemplo:

```
def clase:
```

```
    def metodo():
```

```
        if ():
```

```
            print " "
```

2.10 Diagrama de Componentes

Un diagrama de componentes representa cómo un sistema de *software* es dividido en componentes y muestra las dependencias entre estos. Los Diagramas de Componentes prevalecen en el campo de la arquitectura de *software* pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema. Estos son utilizados para modelar la vista estática y dinámica de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes [27].

A continuación se muestra el diagrama de componentes de la herramienta, donde queda reflejada la forma en que el proyecto se encuentra estructurado:

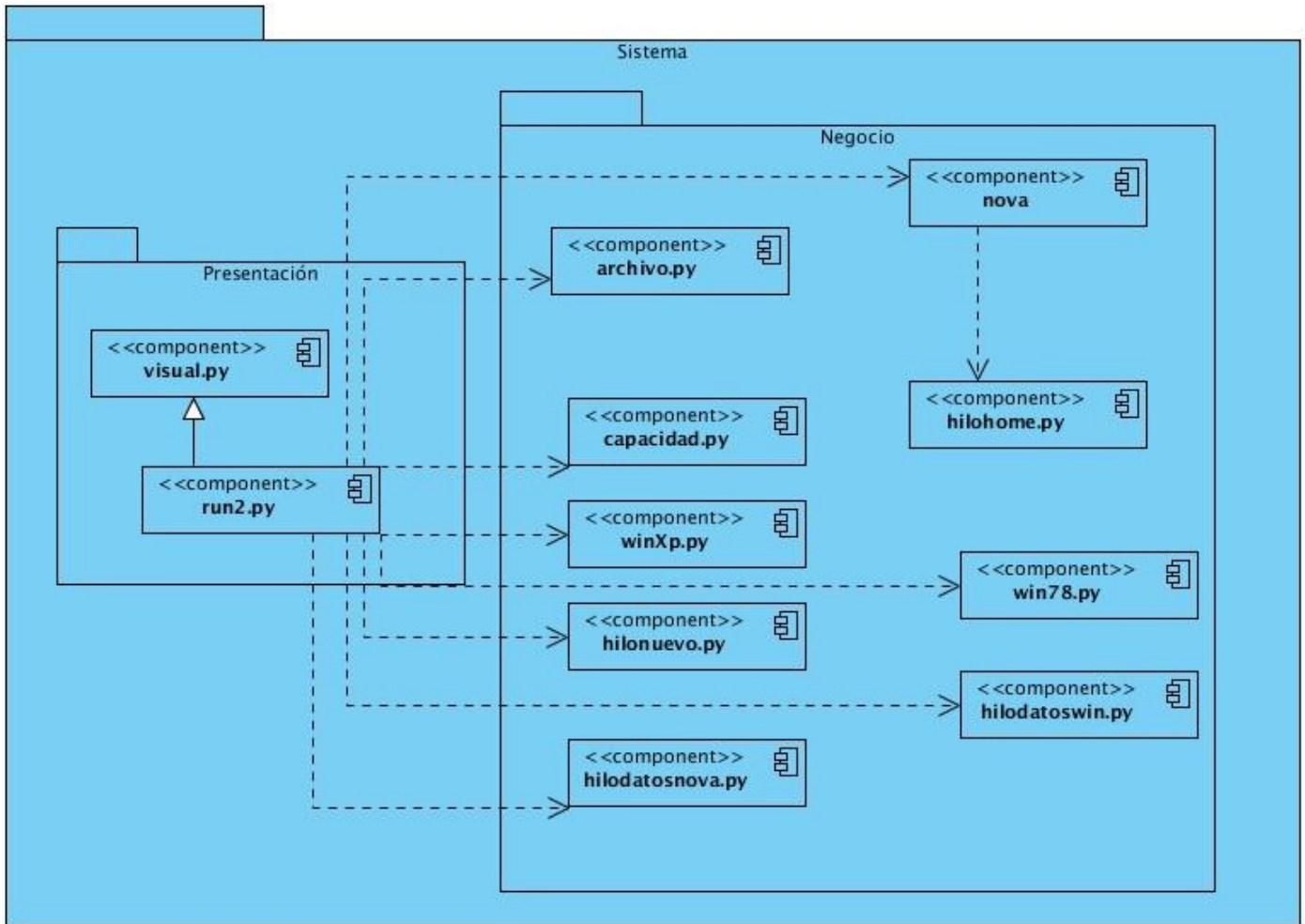


Figura 12: Diagrama de componentes

El paquete **Sistema** contiene toda la estructura de la herramienta, los componente **visual.py** y **run2.py** que están dentro del paquete **Presentación** poseen todos los elementos necesarios para la interfaz gráfica. El componente del paquete **Negocio**, contiene la clase encargada de procesar las solicitudes del usuario y de hacer cumplir las reglas del negocio.

Conclusiones parciales

La entrevista aplicada a los especialistas en migración permitió realizar el levantamiento de los requisitos funcionales y no funcionales de la herramienta. Con la definición de la arquitectura propuesta se logra organizar el código de la solución. Los patrones de diseño permitieron asignar correctamente las relaciones y responsabilidades de cada clase.

Capítulo 3: Pruebas a la herramienta propuesta

Introducción

En el presente capítulo se realizan las pruebas correspondientes a la herramienta. Se explica como se comprueba la integridad de los datos y que se disminuya el tiempo empleado en la migración de los datos y configuraciones.

3.1 Pruebas de software

Las pruebas de *software* son un conjunto de herramientas, técnicas y métodos que evalúan la excelencia y el desempeño de un *software*. Involucra las operaciones del sistema bajo condiciones controladas y evalúa los resultados. Las técnicas para encontrar problemas en un programa son extensamente variadas y van desde el uso del ingenio por parte del personal de prueba hasta herramientas automatizadas que ayudan a aliviar el peso y el costo de tiempo de esta actividad. Una vez completada la implementación, es necesario probar el software con el objetivo de descubrir y corregir la mayor cantidad de errores posibles antes de entregárselo al cliente [27].

3.1.1 Prueba de aceptación

El objetivo de las pruebas de aceptación es probar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento. Estas pruebas están destinadas a probar que el producto está listo para el uso operativo.

3.1.2 Casos de pruebas de aceptación

En la metodología SXP las pruebas se documentan en un artefacto denominado caso de prueba de aceptación, en la que el desarrollador, el cliente y el probador comprueban y validan las funcionalidades del sistema a partir de las historias de usuario implementadas, para finalmente decidir la liberación del

producto. En este epígrafe se muestran los principales casos de prueba de aceptación más significativos.

Tabla 21: Caso de prueba de aceptación 1.

Caso de Prueba de Aceptación	
Código Caso de Prueba: CP-1	Nombre Historia de Usuario: HU-1
Nombre de la persona que realiza la prueba: Grabiél Montero Moreno	
Descripción de la Prueba: Permite comprobar que se creen las carpetas necesarias para migrar las configuraciones.	
Condiciones de Ejecución: Sistema operativo <i>Windows</i> XP, 7 o 8. Tener instalado PyQT4 y Python 2.7.	
Entrada / Pasos de ejecución: Ejecutar la aplicación en el dispositivo de almacenamiento externo. Presionar el botón Siguiente . Seleccionar de los usuarios las configuraciones. Presionar el botón Siguiente .	
Resultado Esperado: En el directorio <dispositivo de almacenamiento externo>/salva/users/ se crea un carpeta para cada usuario de los cuales seleccionó las aplicaciones, con el nombre de cada usuario, y a su vez esta tiene dentro las carpetas con el nombre de las configuraciones.	

Tabla 22: Caso de prueba de aceptación 2.

Caso de Prueba de Aceptación	
Código Caso de Prueba: CP-2	Nombre Historia de Usuario: HU-2
Nombre de la persona que realiza la prueba: Grabiél Montero Moreno	
Descripción de la Prueba: Permite comprobar que se creen las carpetas necesarias para migrar las carpetas del perfil de usuario.	
Condiciones de Ejecución: Sistema operativo <i>Windows</i> XP, 7 u 8. Tener instalado PyQT4 y Python 2.7.	
Entrada / Pasos de ejecución: Ejecutar la aplicación en el dispositivo de almacenamiento externo. Presionar el botón Siguiente . Seleccionar de los usuarios las carpetas del perfil de usuario. Presionar el botón Siguiente .	
Resultado Esperado: En el directorio <dispositivo de almacenamiento externo>/salva/users/ se crea un carpeta para cada usuario de los cuales seleccionó las aplicaciones, con el nombre de cada usuario, y a su vez esta tiene dentro las carpetas con el nombre de las carpetas del perfil de usuario.	

Tabla 23: Caso de prueba de aceptación 3.

Caso de Prueba de Aceptación	
Código Caso de Prueba: CP_3	Nombre Historia de Usuario: HU_3
Nombre de la persona que realiza la prueba: Grabiél Montero Moreno	
Descripción de la Prueba: Permite al usuario seleccionar los datos que desea salvar.	
Condiciones de Ejecución: Sistema operativo <i>Windows</i> XP, 7 u 8. Tener instalado PyQT4 y Python 2.7.	

Entrada / Pasos de ejecución:

Ejecutar la aplicación en el dispositivo de almacenamiento externo.

Presionar el botón **Siguiente**.

Seleccionar de los usuarios las carpetas del perfil de usuario y configuraciones que desea salvar.

Presionar el botón **Siguiente**.

Seleccione datos que desee salvar.

Presionar el botón siguiente.

Resultado Esperado:

El sistema debe mostrar una barra de proceso, indicando que se están copiando los datos.

Tabla 24: Caso de prueba de aceptación 4.

Caso de Prueba de Aceptación	
Código Caso de Prueba: CP_4	Nombre Historia de Usuario: HU_4
Nombre de la persona que realiza la prueba: Grabiél Montero Moreno	
Descripción de la Prueba: Permite al usuario cerrar la aplicación en cualquier momento de la ejecución	
Condiciones de Ejecución: Sistema operativo <i>Windows</i> XP, 7 u 8. Tener instalado PyQT4 y Python 2.7.	
Entrada / Pasos de ejecución: Ejecutar el programa en el disco duro. Presionar el botón siguiente. Marcar de alguna opción. Presionar el botón siguiente. Marcar datos que desee salvar. Presionar el botón siguiente.	

Presionar salvar.

Presione cancelar.

Resultado Esperado: Se cierra la aplicación.

Se realizaron un total de 12 pruebas, divididas en 3 iteraciones según los resultados obtenidos en cada caso de prueba de aceptación. En la primera iteración fueron arrojadas 4 no conformidades siendo corregidas posteriormente. En la segunda iteración, se obtuvo como resultado 1 no conformidad que fue corregida de manera inmediata. En la tercera los resultados arrojados fueron satisfactorios, comprobando que la herramienta se encuentra lista para ser utilizada. A continuación se ilustra un gráfico que muestra el proceso detalladamente.

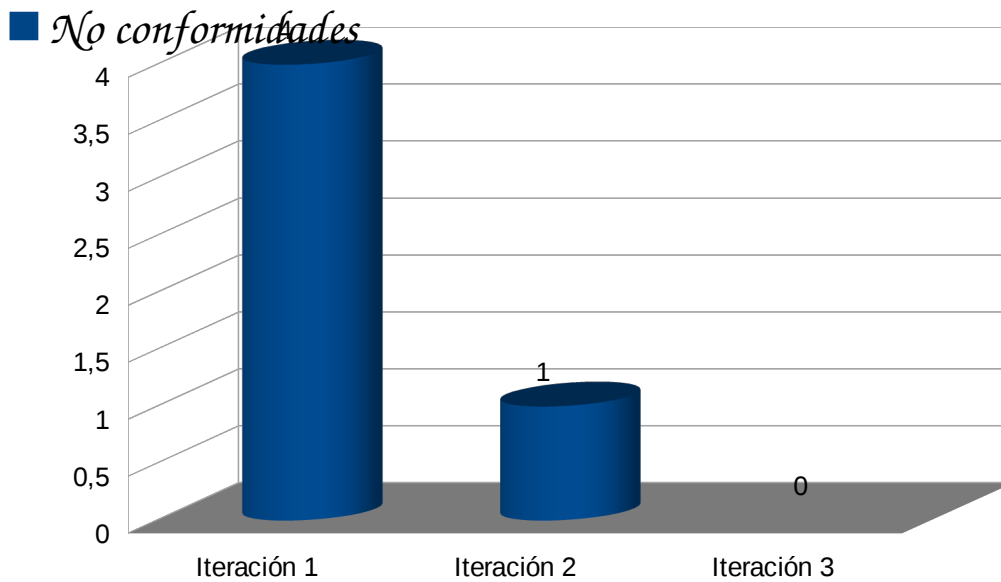


Figura 13: Resultado de las pruebas de aceptación

No conformidades detectadas en la iteración 1:

- No se salva la configuración del fondo del escritorio.
- La herramienta no muestra los datos copiados.
- Luego de restablecer las configuraciones del Firefox no carga las configuraciones.
- Salva carpetas .link que no son necesarias en GNU/Linux Nova.

No conformidades detectadas en la iteración 2:

- No restablece el usuario salvado.

3.2 Integridad de los datos

Entre las funciones *hash* utilizadas con mayor frecuencia se encuentran MD5 y SHA-1 [8]. MD5 toma un mensaje (archivo,cadena) de cualquier longitud y produce un valor *hash* de 126 bits. El mensaje es procesado en bloques de 512 bits. SHA-1 toma un mensaje de hasta 256 bits y produce un valor *hash* de 160 bits; es computacionalmente imposible encontrar dos *hash* iguales [34].

Para garantizar la integridad de los datos en la solución propuesta se hace uso del módulo **hashlib**. Dicho módulo implementa algunos algoritmos para generar la función resumen de un archivo. En el trabajo se utiliza el algoritmo SHA-1 para generar la función resumen del archivo original, luego de copiado el archivo este *hash* se almacena, como el valor, junto con la dirección del archivo copiado en un diccionario¹⁴, para que luego de terminado el proceso de salva se verifique si el archivo copiado no ha sufrido ningún cambio; y así asegurar la integridad en el proceso.

3.3 Disminución del tiempo en la migración de los datos y configuraciones

Para demostrar que se cumple con el objetivo de disminuir el tiempo en la migración de los datos y configuraciones se define el siguiente entorno de prueba:

Tabla 25: Entorno de prueba

Dispositivos	Características
Computadora	Un disco duro SATA de 1TB de capacidad, CPU Intel Core i3 de segunda generación, con 4 núcleos y una velocidad de 3,30 GHz, y memoria RAM DDR3 de 4 GB.
Dispositivo de almacenamiento externo	USB 2,0 y una capacidad de 16 GB.
Datos	

14 **Diccionario:** son colecciones que relacionan una clave y un valor, también son llamados matrices asociativas.

Mis documentos	17 elementos con capacidad de 14,2 MB en total.
Descargas	115 elementos con capacidad de 256,8 MB en total.
Escritorio	73 elementos con capacidad de 192,1 MB en total.
Mozilla <i>Firefox</i>	24860 elementos con capacidad de 643,1 MB en total.
Mozilla <i>Thunderbird</i>	42 elementos con capacidad de 15,0 MB en total.
Pidgin	161 elementos con capacidad de 1,7 MB en total.
Datos adicionales	1354 elementos con capacidad de 13,9 GB en total.
Total	26622 elementos con capacidad de 15356,5 MB o 14,997 GB.

Este entorno de pruebas fue realizado en 3 iteraciones (ver Figura 18 y Figura 19) y luego se calculó la media de los tiempos obtenidos, arrojando como resultados que: en *Windows XP*, *Windows 7*, *Windows 8*, GNU/Linux Nova con la aplicación el proceso tarda 56.1, 49.83, 43.3, 37.43 respectivamente, sin el uso de esta 58.23, 50.26, 48.53, 38.86 respectivamente. Con los resultados obtenidos se llega a la conclusión de que utilizando la herramienta se disminuye el tiempo del proceso.

Conclusiones parciales

Las pruebas de aceptación ejecutadas, el uso de SHA-1 para asegurar la integridad de los datos y el resultado del entorno de prueba definido, permitieron obtener una herramienta libre de errores y ajustada a las necesidades del cliente.

Conclusiones generales

Como resultado del trabajo realizado se concluye que:

1. Se realizó una investigación acerca de las herramientas que posibilitan la migración de datos y configuraciones de aplicaciones, arribando a la conclusión de que era necesaria la implementación de una nueva aplicación para cumplir con las necesidades requeridas.
2. El establecimiento de SXP como metodología rectora del desarrollo de la herramienta, y el uso de las tecnologías y herramientas definidas, permitieron analizar, describir e implementar los procesos que debían ejecutarse, materializando así una solución acorde a los requerimientos del cliente.
3. Con la implementación de la herramienta para la migración de estaciones de trabajo de *Windows* XP, 7 y 8 a GNU/Linux Nova en la ejecución de la migración, es posible asegurar la integridad de los datos en el proceso.
4. Las pruebas realizadas evidenciaron el correcto funcionamiento de la herramienta implementada y la disminución del tiempo empleado por un especialista en la ejecución de la migración de los datos y configuraciones de los usuarios.

Recomendaciones

En la presente investigación se recomienda para futuras versiones de la herramienta las siguientes consideraciones:

- Incorporar funcionalidad para salvar y restaurar las configuraciones del navegador *web Google Chrome*.
- Incorporar funcionalidades para realizar el proceso de salvar y restaurar a través de un servidor ftp.
- Incorporar funcionalidad para asegurar la seguridad de los datos.

Referencias Bibliográficas

- [1] Martínez, Fernando Zambrano, “Las TICS en nuestro ámbito social”, Revista Digital Universitaria, 1 de noviembre de 2009, Vol.10, No.11,[En línea]. [Accedido el 27 de noviembre de 2014]. Disponible en: <http://www.revista.unam.mx/vol.10/num11/art79/int79.htm>.
- [2] Diario Digital de Noticias, “Definición de TIC”,[En línea]. [Accedido el 27 de noviembre de 2014]. Disponible en: <http://www.serviciostic.com/las-tic/definicion-de-tic.html>.
- [3] Relación entre la informática y las TICS, 24 de Marzo del 2010,[En línea]. [Accedido el 30 de noviembre de 2014]. Disponible en: <http://ticssalle21.blogspot.com/2010/03/relacion-entre-la-informatica-y-las.html>.
- [4] La Red, David Luis Martínez, “Sistemas Operativos”,[En línea]. [Accedido el 30 de noviembre de 2014]. Disponible en: <https://books.google.es/books?hl=es&lr=&id=N5yxiwILBhoC&oi=fnd&pg=PR1&dq=definici%C3%B3n+de+sistema+operativo&ots=01QCDG4AT&sig=IZRjQF50p4SiV7cojMiHgOnyqXE#v=onepage&q=definici%C3%B3n%20de%20sistema%20operativo&f=false>.
- [5] Pérez, Yoandy Villazón, “El proceso de migración a aplicaciones de código abierto en Cuba desde un enfoque metodológico”. Revista Cubana de Ciencias Informáticas, 2013, vol.7, n.4, pp. 31-41. ISSN 2227-1899,[En línea]. [Accedido el 27 de noviembre de 2014]. Disponible en: http://scieloprueba.sld.cu/scielo.phpscript=sci_arttext&pid=S222718992013000400003&lng=es&nrm=iso.
- [6] PÉREZ, Y., GARCÍA, A., GARCÍA, J., HERNÁNDEZ, Y., CUESTA, E., VIERA, A., PÉREZ, A. y SÁNCHEZ, S. 2014. Buenas prácticas para la migración a código abierto. *V Taller Nacional Software Libre Presente y Futuro*. Villa Clara: s.n., ISBN 978-959-250-999-3.
- [7] Pérez, Anadalis Rodríguez, Junio 2011, “Propuesta de expediente de proyecto para la Metodología Cubana de Migración a *Software* Libre y Código Abierto”.

- [8] Jara, Héctor, Pacheco, G. Federico. Ethical hackin 2.0. Buenos Aires. Fox Adina, Dalaga. 2012. 352 p.
- [9] Vermeulen, Sven. What is Linux?. [En línea]. [Accedido el 30 de noviembre de 2014]. Disponible en: http://swift.siphos.be/linux_sea/whatislinux.html.
- [10] Vermeulen, Sven. The Linux File System. [En línea]. [Accedido el 30 de noviembre de 2014]. Disponible en: <http://swift.siphos.be/linuxsea/linuxfs.html>.
- [11] Comparación de los sistemas de archivos NTFS y FAT, [En línea]. [Accedido el 27 de noviembre de 2014]. Disponible en: <http://windows.microsoft.com/es-419/windows-vista/comparing-ntfs-and-fat-file-systems>.
- [12] Jorba, Josep Esteve, Suppi, Remo Boldrito. "Administración Avanzada en GNU_Linux".[En línea]. [Accedido el 3 de enero de 2015]. Disponible en: http://www.uoc.edu/masters/softwarelibre/esp/materials/Admin_GNUlinux.pdf.
- [13] Introduction to File systems. [En línea]. [Accedido el 3 de enero de 2015]. [Disponible en] <https://help.ubuntu.com/community/LinuxFilesystemsExplained>.
- [14] Sistemas de archivos más populares,[En línea]. [Accedido el 3 de enero de 2015]. Disponible en: <http://www.informaticamoderna.com/sistemaarch.htm>.
- [15] ¿Cómo hace *Windows* Easy Transfer que configurar un equipo sea más fácil?.[En línea]. [Accedido el 7 de enero de 2015]. Disponible en: <http://windows.Micros oft.com/es-419/windows7/how-does-windows-easy-transfer-make-setting-up-a-computer-easier>.
- [16] La manera más fácil para moverse a un ordenador con *Windows* XP o Vista PC,[En línea]. [Accedido el 7 de enero de 2015]. Disponible en <http://ww2.laplink.com/esp/pcmover>.
- [17] Asistente de migración, la forma más sencilla de transferir la información de un Mac a otro. 20 de enero de 2015. [En línea]. [Accedido el 27 de enero de 2015]. Disponible en: <http://www.applesfera.com/applesfera-responde/asistente-de-migracion-la-forma-mas-sencilla-de->

[transferir-la-informacion-de-un-mac-a-otro.](#)

[18] AMIGU, Asistente de Migración de Guadalinex. [En línea]. [Accedido el 3 de enero de 2015]. [Disponible en] <http://forja.guadalinex.org/webs/amigu/index.php?pag=0>.

[19] González, Raúl Duque, “Python para todos”, [En línea]. [Accedido el 15 de enero de 2015]. Disponible en <https://launchpadlibrarian.net/18980633/Python%20para%20todos.pdf>.

[20] Rumbaugh, James. Jacobson, Ivar. Booch, Grady. El Lenguaje Unificado de Modelado. Manual de Referencia. Addison Wesley, 2007.

[21] [En línea]. [Accedido el 15 de enero de 2015]. [Disponible en] <http://www.scintilla.org/SciTE.html>.

[22] [En línea]. [Accedido el 17 de enero de 2015]. [Disponible en] http://es.wikibooks.org/wiki/Programaci%C3%B3n_con_Qt4.

[23] [En línea]. [Accedido el 17 de enero de 2015]. [Disponible en] <http://doc.qt.io/qt-4.8/designer-manual.html>.

[24] [En línea]. [Accedido el 3 de enero de 2015]. [Disponible en] <http://www.visual-paradigm.com/aboutus/newsreleases/vp121.jsp>.

[25] Peñalver Romero, Gladys Marsí. Metodología ágil para proyectos de *software* libre. Trabajo de Diploma. Universidad de las Ciencias Informáticas Ciudad de La Habana. Junio de 2008.

[26] Fernández Céspedes, Raycel. Pino García, Susel. Propuesta de un expediente, para los proyectos productivos del Polo de *Software* Libre. Trabajo de Diploma. Universidad de las Ciencias Informáticas. 2008.

[27] Larman, Craig. UML y Patrones. 2a Edición. Prentice Hall. 2003 .507.

[28] Preesman, Roger. Ingeniería de *software*. Un enfoque práctico. sexta edición.

[29] “Diagrama de Clases”. [En línea]. [Accedido el 27 de enero de 2014]. Disponible en:

<http://sistemasdeinformacion2.wikispaces.com/Diagrama+de+Clases>].

[30] Tedeschi, Nicolás. ¿Qué es un Patrón de Diseño? [En línea]. [Accedido el 27 de enero de 2014]. Disponible en: <http://msdn.microsoft.com/es-es/library/bb972240.aspx>.

[31] Gamma, Erich, Helm, Richard, Johnson, Ralph y Vlissides, Jhon 1995, DesignPatterns: Elements of Reusable Object-Oriented *Software*. S.I.Addison Wesley.

[32] Fowler, Martin, Patterns of Enterprise Application Architecture, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©2002 ISBN:0321127420.

[33] Van Rossum, Guido. Warsaw, Barry. Coghlan, Nick. PEP 0008 -- Style Guide for Python Code. [En línea]. [Accedido el 27 de enero de 2014]. Disponible en: <https://www.python.org/dev/peps/pep-0008/>.

[34] Rhee, Man Young. Internet security: cryptographic principles, algorithms, and protocols. Inglaterra. 2002. 426 p.

Entrevista

Durante el proceso de migración de Windows a GNU/Linux se llavan a cabo varias tareas para lograr un alto grado de aceptación por parte de los usuarios y evitar un rechazo a el proceso.

1 ¿Ha participado en un proceso de migración de las estaciones de trabajo de Windows a GNU/Linux?

Si No

2 Seleccione: ¿cuáles de las siguientes configuraciones migró? Explique brevemente como las migró.

Configuración de la red

Configuración del fondo del escritorio

Configuración del navegador web Mozilla Firefox

Configuración del navegador web Internet Explorer

Configuración del cliente de correo Mozilla Thunderbird

Configuración del cliente de correo Microsoft Outlook

Nombre de la computadora

Otra(s). ¿Cuál(es)? _____

3 Mantener la integridad de los datos es fundamental en un proceso de migración

a) ¿Ha tenido que migrar datos?

Si No

b) ¿Qué datos migró?

Particiones enteras Vídeos

El contenido de la carpeta Mis vídeos

Documentos El contenido de la carpeta Escritorio

El contenido de la carpeta Descarga

Imágenes El contenido de la carpeta Mi Música

El contenido de la carpeta Mis Imagenes

Música El contenido de la carpeta Mis documentos Otro(s). ¿Cuál(es)? _____

Figura 14: Entrevista aplicada a los especialistas.

The image shows a Windows File Explorer window displaying the directory structure of three drives: C:, D:, and E:. The window is organized into columns for Name, Size, and Type. The C: drive contains folders like EasyEclipse-for-LA..., PerfLogs, Program Files, Python27, Users, and Windows, along with files autoexec.bat and config.sys. The D: drive contains folders like ACTIVADOR, AUTORUN.INF, BOOT, BOOTMGR, EFI, SOURCES, SUPPORT, and UPGRADE, along with files SETUP.EXE and AUTORUN.INF. The E: drive is currently empty.

Name	Size	Type
C:		Drive
EasyEclipse-for-LA...		File Folder
PerfLogs		File Folder
Program Files		File Folder
Python27		File Folder
Users		File Folder
Windows		File Folder
autoexec.bat	24 byte(s)	bat File
config.sys	10 byte(s)	sys File
D:		Drive
ACTIVADOR		File Folder
AUTORUN.INF	43 byte(s)	INF File
BOOT		File Folder
BOOTMGR	374 KB	File
EFI		File Folder
SETUP.EXE	109 KB	EXE File
SOURCES		File Folder
SUPPORT		File Folder
UPGRADE		File Folder
E:		Drive

Figura 15: Estructura de directorio en Windows.

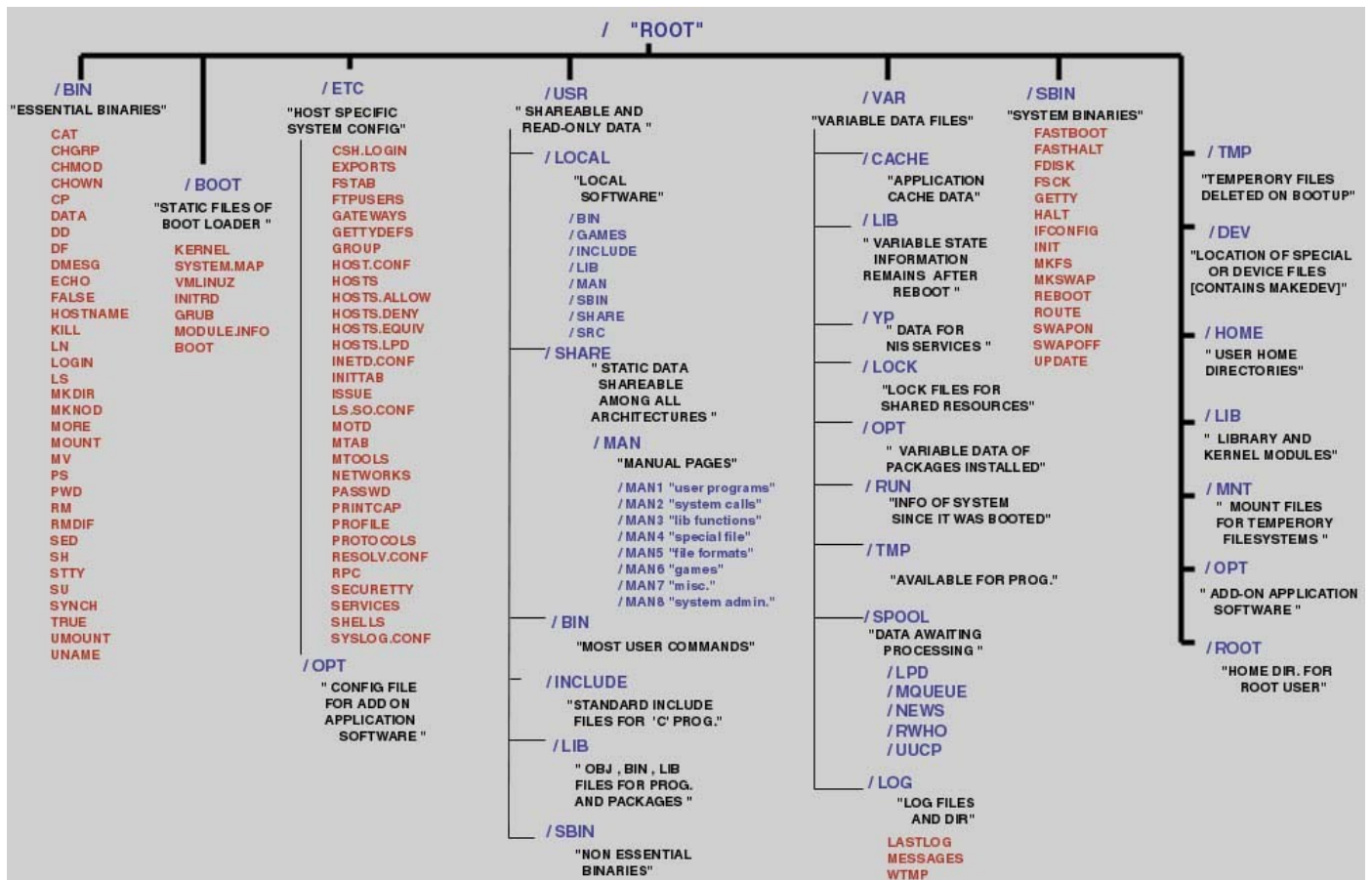


Figura 16: Estructura de directorio en GNU/Linux.

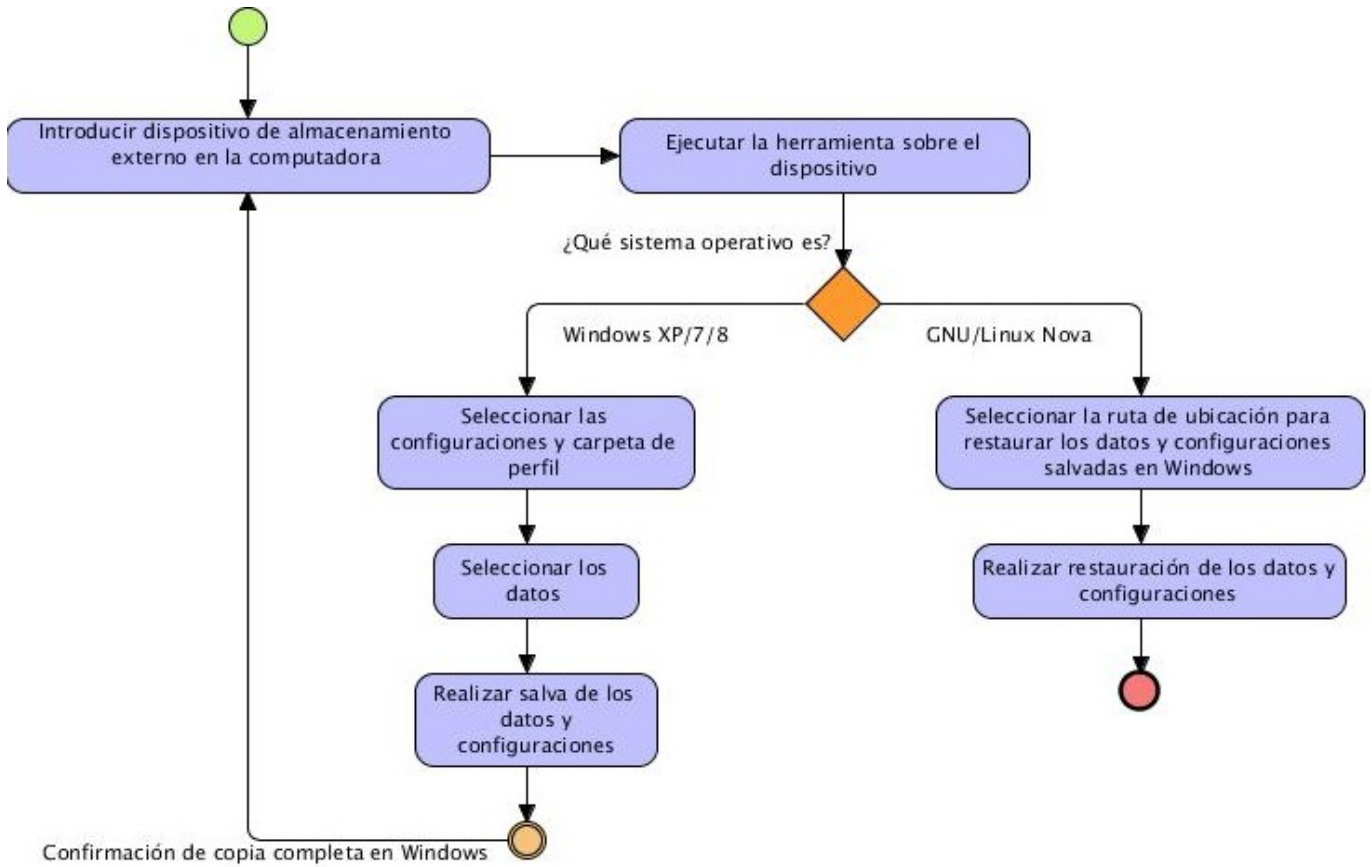


Figura 17: Descripción de la propuesta.

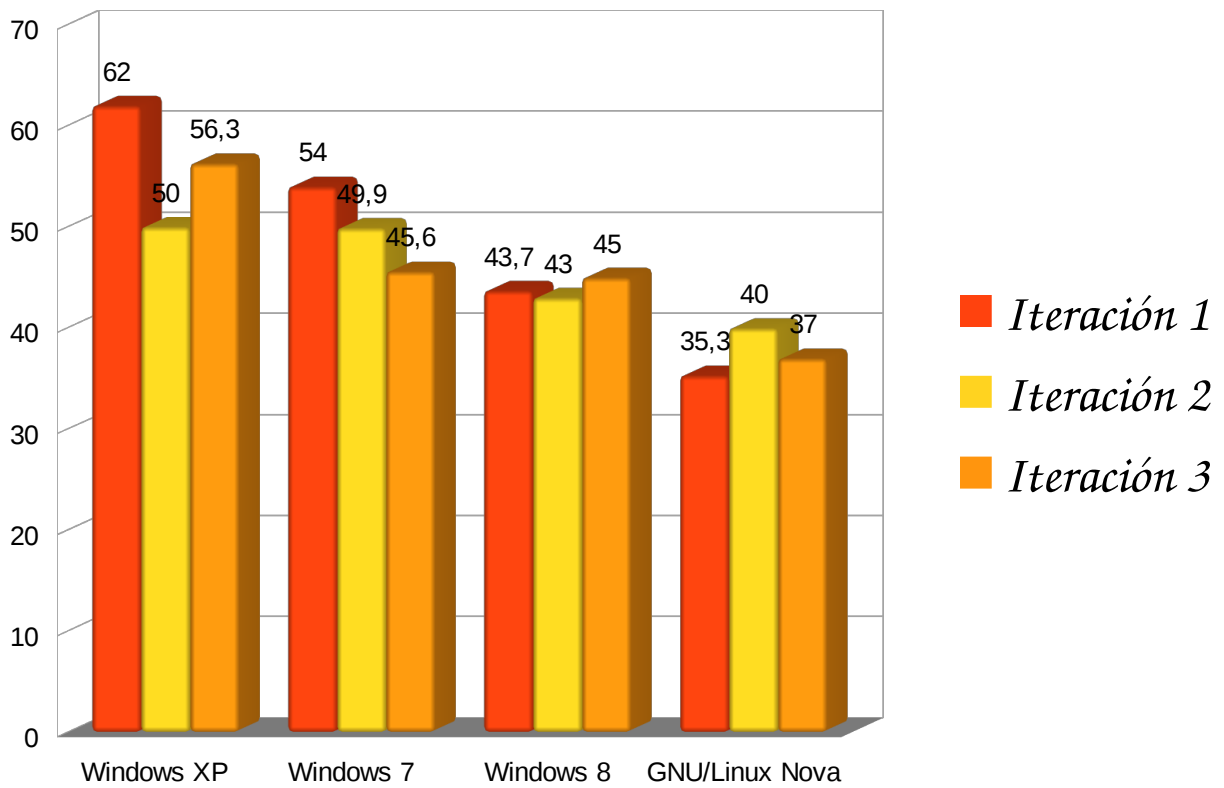


Figura 18: Resultados del entorno de prueba utilizando la herramienta.

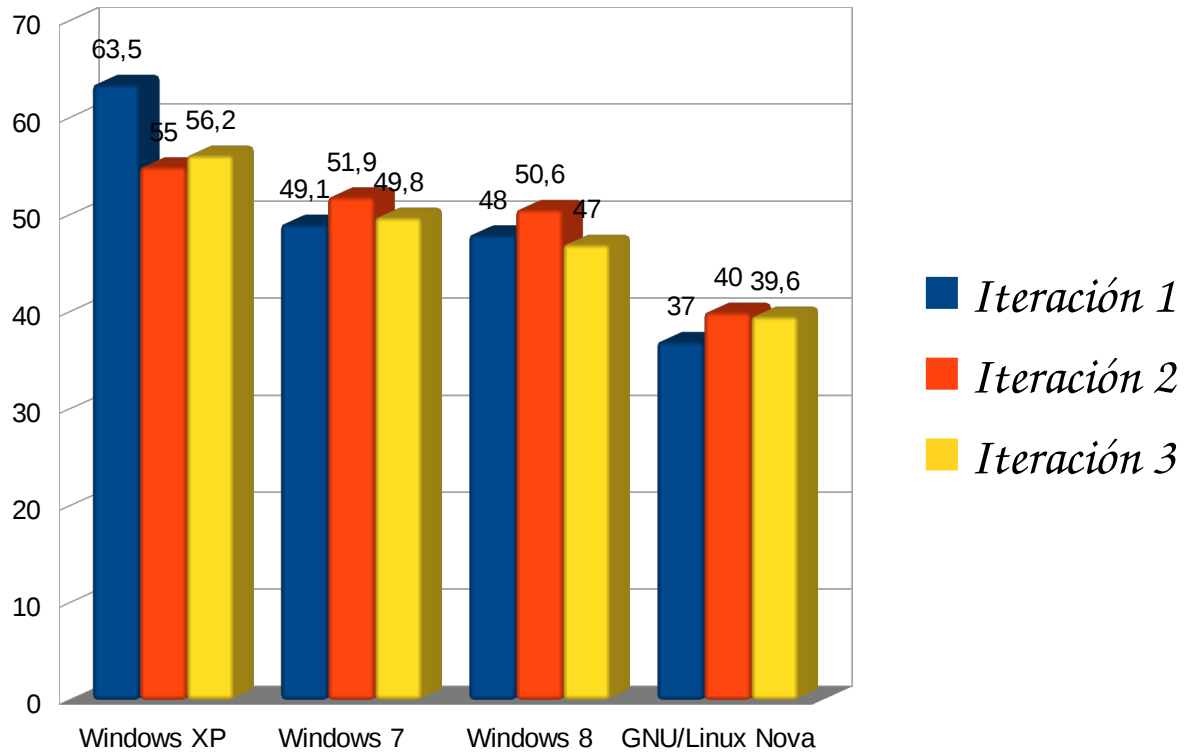


Figura 19: Resultados del entorno de prueba sin utilizar la herramienta.