

Universidad de las Ciencias Informáticas

Facultad 4



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

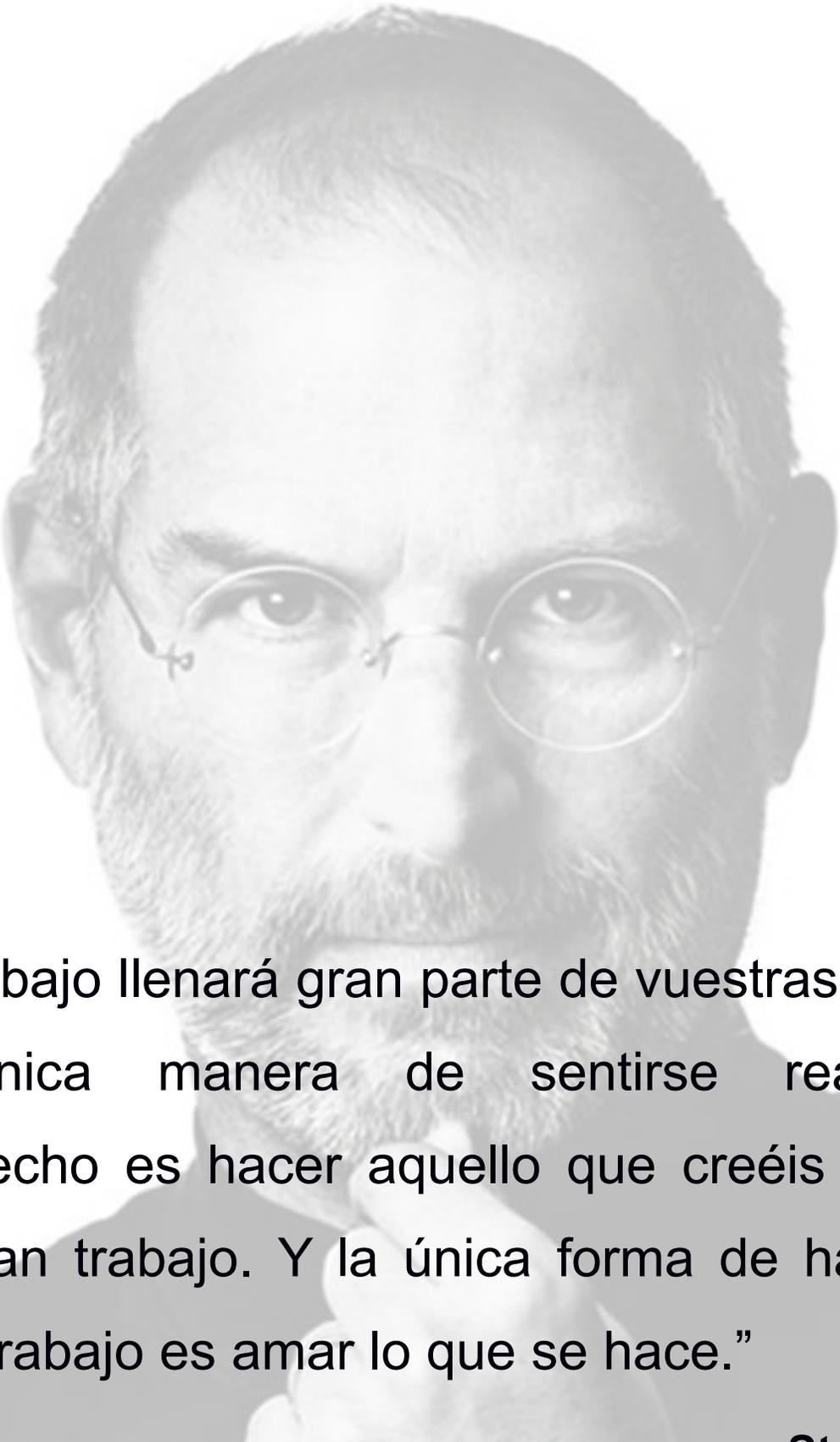
**Título: Módulo Gestor de Archivos para el Agente Transitario
de Cargas TRANSCARGO**

Autor: Arian Danilo Escalona Márquez

Tutores: Ing. Leannys Rodríguez Moreno
Ing. Daryl Yturralde López

La Habana, junio de 2017.

“Año 59 de la Revolución”



“El trabajo llenará gran parte de vuestras vidas y la única manera de sentirse realmente satisfecho es hacer aquello que creéis que es un gran trabajo. Y la única forma de hacer un gran trabajo es amar lo que se hace.”

Steve Jobs

Declaración de Autoría

Declaro ser el único autor de este trabajo y otorgo 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 ____.

Autor: Arian Danilo Escalona Márquez

Tutora: Ing. Leannys Rodriguez Moreno

Tutor: Ing. Daryl Yturalde López

Agradecimientos:

A mi mamá: por hacerme feliz, por ser lo más grande en mi vida, por ser la más fuerte guerrera que exista, por sus consejos siempre certeros, por sus abrazos, por las enseñanzas, por sus besos, por su calor, por su cariño, por todo.

A mi papá: por ser mi amigo, a mi novia por ayudarme en todo, por entenderme, por su amor, dedicación, apoyo incondicional, por amarme, por todo.

A mis abuelos: por su cariño, amor, comprensión, consentimiento y dedicación, por los consejos, por siempre estar a mi lado.

A mis tutores: por guiarme en todo el proceso, por su participación activa, por los conocimientos transmitidos.

A mis amigos: por las experiencias vividas, por la amistad, comprensión, consejos, por todo.

A la Universidad de las Ciencias Informáticas: por la oportunidad de cumplir un sueño.

Dedicatoria

Dedicatoria:

Dedico esta tesis a la persona más importante en mi vida, mi mamá, por su dedicación, amor, comprensión, apoyo, amistad incondicional, por su incansable exigencia, por su ayuda incondicional y por haberme traído al mundo.

A mi primo Roberto por compartir y formar parte de mi vida e inculcarme valores, ayudarme a tomar decisiones y aconsejarme en momentos críticos.

A mi novia quien estuvo a mi lado en los momentos buenos y malos, muchas veces fajados pero siempre juntos ya que somos inseparables y en mi corazón siempre está presente.

A dos personas importantes en mi vida a mis abuelos por los conocimientos y las preocupaciones transmitidas, por el cariño dado y por siempre preocuparse por su nieto. A mi hermanita por convertirse con su pequeño tamaño en algo muy grande para mí.

Dedico esta tesis a mi padre que lo querré por siempre y a todos los que he conocido a través de mi vida.

Resumen

El creciente desarrollo de la industria de *software* cubano y el avance logrado en la informatización del país han brindado grandes beneficios y adelantos en todas las esferas de la sociedad, convirtiéndose el uso de la información en la unidad más importante con que cuenta una empresa. Su crecimiento ha propiciado la creación de sistemas capaces de salvaguardar y mantener su disponibilidad, evitando su duplicidad y dispersión. El Agente Transitario de Cargas TRANSCARGO no cuenta con un sistema capaz de gestionar la información generada en la empresa debido a la gran cantidad de documentos que se debe controlar, conservar, recuperar y acceder, siendo una de sus debilidades la carencia de políticas de acceso y la descentralización de la información. Por tal motivo, se pretende desarrollar un módulo capaz de gestionar la información que se genera dentro de los procesos y servicios de TRANSCARGO. La solución propuesta fue realizada con el *framework* OpenObject utilizado para desarrollar sistemas que se encargan de la planificación de recursos empresariales, guiado por la metodología de desarrollo AUP en su versión UCI. Se realizaron pruebas funcionales aplicando el método de caja negra y pruebas de aceptación con el objetivo de examinar exhaustivamente el módulo. Como aporte de esta investigación se encuentra el desarrollo de un módulo para la gestión de la información que incluye todas las acciones sobre los documentos tales como: crear, clasificar, exportar, gestionar flujos documentales, almacenar documentos en diferentes formatos electrónicos y gestionar los trámites de los documentos que se generan o reciben.

Palabras claves: directorios, gestión de información.

Índice de Contenidos

Índice de Contenidos

Introducción.....	1
Capítulo 1. Fundamentación teórica de la investigación.....	5
1.1 Sistema integral de gestión empresarial.....	5
1.2 Análisis de soluciones similares.....	6
1.3 Metodología, lenguajes, herramientas, tecnologías y marco de trabajo.....	8
1.3.1 <i>Metodologías de desarrollo de software</i>	9
1.3.2 <i>Lenguaje Unificado de Modelado</i>	10
1.3.3 <i>Herramienta para el modelado</i>	11
1.3.4 <i>Lenguaje del lado del cliente. JavaScript</i>	12
1.3.5 <i>Lenguajes del lado del servidor. Python</i>	13
1.3.6 <i>Marco de trabajo de desarrollo. OpenObject</i>	13
1.3.7 <i>Sistema de planificación de recursos empresariales. Odoo</i>	15
1.3.8 <i>Sistema Gestor de Base de Datos. PostgreSQL</i>	16
1.3.9 <i>Entorno integrado de desarrollo. PyCharm</i>	17
1.4 Conclusiones del capítulo.....	17
Capítulo 2. Propuesta de solución.....	18
2.1 Propuesta del sistema.....	18
2.2 Modelo de dominio.....	19
2.2.1 <i>Diagrama del modelo de dominio</i>	19
2.3 Requisitos del sistema.....	20
2.3.1 <i>Requisitos funcionales</i>	20
2.3.2 <i>Requisitos no funcionales</i>	21

Índice de Contenidos

2.3.3	<i>Descripción de requisitos por procesos</i>	22
2.3.4	<i>Arquitectura de Odo</i>	25
2.4	Patrón arquitectónicos.....	26
2.5	Modelo de diseño.....	27
2.6	Patrones de diseño.....	28
2.7	Patrones GRASP.....	28
2.8	Patrones GOF.....	29
2.9	Modelo de datos.....	29
2.10	Conclusiones del capítulo.....	31
Capítulo 3. Implementación y Prueba.....		32
3.1	Estándar de codificación.....	32
3.2	Diagrama de componentes.....	34
3.3	Diagrama de despliegue.....	35
3.4	Pruebas de software.....	36
3.5	Niveles de Prueba.....	36
3.5.1	<i>Resultados de las pruebas</i>	38
3.6	Conclusiones del capítulo.....	40
Conclusiones.....		41
Recomendaciones.....		42
Referencias Bibliográficas.....		43

Índice de Figuras

Índice de Figuras

Figura 1: Modelo conceptual	22
Figura 2: Arquitectura Cliente/Servidor en Odoo.....	29
Figura 3: Modelo Vista Controlador.....	30
Figura 4: Diagrama de clases del diseño del RF Crear Carpeta.....	31
Figura 5: Estándar de codificación.....	35
Figura 6: Estándar de codificación.....	35
Figura 7: Estándar de codificación.....	36
Figura 8: Diagrama de componentes del módulo Gestor de Archivos.....	37
Figura 9: Diagrama de despliegue.....	38

Índice de Tablas

Índice de Tablas

Tabla 1. Requisitos funcionales.....	25
Tabla 2.Especificación de Requisitos.....	27
Tabla 3.Descripción de RF: Crear carpeta.....	28
Tabla 4: Diseño de caso de prueba correspondiente a la funcionalidad Crear Carpeta.....	45
Tabla 5: No Conformidades del Módulo Gestor de archivos 1ra iteración.....	46
Tabla: 6 Descripción de No Conformidades 1ra iteración.....	47

Introducción

La sociedad ha experimentado un desarrollo tecnológico acelerado en las últimas décadas. Con el surgimiento de las computadoras es posible el almacenamiento de gran cantidad de información haciendo uso de poco trabajo manual. A nivel mundial, los procesos en las organizaciones se encuentran automatizados de manera integral para ganar en orden y reutilización de la información, en aras de optimizar, tiempo y recursos computacionales y así controlar dichos procedimientos de manera más eficiente y segura (Visbal, 2013). La informatización de procesos depende en gran medida de los sistemas de gestión de la información. En los últimos años, este cambio ha provocado que la mayoría de las empresas tengan la necesidad de enfrentar este desafío para su beneficio, a tal punto que se les ha hecho imprescindible contar con estos sistemas para su mejor desarrollo (Visbal, 2013).

En Cuba, una de las empresas que se encuentra inmersa en el perfeccionamiento empresarial y la informatización de sus procesos es el Agente Transitario de Cargas TRANSCARGO. Esta empresa se encarga de organizar y coordinar el transporte de mercancías de sus clientes, mediante la contratación de diversos medios de transporte. La principal misión de TRANSCARGO es prestar servicios transitarios y de operador logístico internacional de manera ética y eficaz para organizar la cadena de servicios competitivos desde el origen hasta la entrega de las mercancías, personalizando los intereses de sus clientes (Márquez, 2014). Para prestar los servicios de corresponsalía referentes a la actividad transitoria, agenciamiento aduanal, logísticas asociadas al arrendamiento de contenedores, almacenajes y despacho de cargas comerciales y no comerciales; se utiliza el Sistema de Gestión de Actividades Transitarias, Aduaneras y Logísticas de Carga (SIGAX).

El trabajo con los archivos en TRANSCARGO se hace engorroso debido a la descentralización de la información y la cantidad de documentos generados. Cada especialista independientemente del área a la que pertenezca, deber generar los documentos asociados a la parte del ciclo de vida de una solicitud de embarque que le corresponda según la misma y guarda los documentos en su ordenador. Seguidamente para que la solicitud pueda ser utilizada por el resto de las áreas, el especialista debe enviarla por correo electrónico o trasladarla en algún dispositivo de almacenamiento hacia la siguiente área que la necesite. Esta situación trae consigo dependencia innecesaria entre los usuarios, errores y engorrosas búsqueda de información, retrasando la

Introducción

dinámica de la empresa. Además, carece de un mecanismo adecuado para la manipulación de la información ya que si el ordenador sufre fallas técnicas o errores del usuario, se pierde la información contenida en él.

Existen varios problemas que imposibilitan una adecuada gestión de la documentación tales como:

- Carencia de políticas de acceso correctamente establecidas sobre la información;
- Ausencia de un adecuado control sobre los documentos provocando pérdida y duplicidad de la información pues esta no se encuentra clasificada ni organizada.
- La falta de acceso inmediato a toda la documentación dificulta la localización de ficheros, lo que provoca un aumento del tiempo de consulta y tareas de archivo.

A partir de la problemática antes descrita, se plantea como **problema a resolver** la siguiente interrogante: ¿Cómo contribuir a la gestión de la información que generan los procesos y servicios en TRANSCARGO?

Se determina como **objeto de estudio**: la gestión de la información de los procesos y servicios de TRANSCARGO. Enfocando el **campo de acción** en: la gestión de la información de los procesos y servicios de TRANSCARGO para el SIGAX.

Para dar cumplimiento al problema planteado se propone como **objetivo general**: Desarrollar un módulo para SIGAX, que contribuya a la gestión de la información que se genera dentro de los procesos y servicios de TRANSCARGO.

Del objetivo general planteado se derivan los siguientes **objetivos específicos**:

- Construir los referentes teóricos relacionando los aspectos fundamentales que sustentan la investigación, mediante los cuales se consulta, extrae y recopila información relevante sobre el problema a investigar.
- Realizar el modelado y descripción de los procesos identificados.
- Realizar el análisis y diseño del módulo para la gestión de la información generada de los procesos en TRANSCARGO basado en los requisitos descritos.
- Implementar la propuesta de solución utilizando las herramientas, lenguajes y tecnologías definidas.
- Realizar las pruebas a la solución informática.

Para guiar la investigación se define como **hipótesis**: el desarrollo del módulo para la gestión de la información, contribuirá a la gestión de la información dentro de TRANSCARGO.

Con el fin de dar cumplimiento a los objetivos específicos planteados se concretan las siguientes **tareas de la investigación:**

- Elaboración del marco teórico de la investigación.
- Realización de los diagramas de procesos para obtener un mejor entendimiento de los mismos.
- Realización del levantamiento de los requisitos de software.
- Elaboración del diseño de la propuesta de solución a partir del análisis de los requisitos.
- Implementación del módulo Gestor de Archivos basado en los requisitos descritos.
- Realización de pruebas a las funcionalidades del módulo.

Para la realización de esta investigación se utilizaron los siguientes **métodos de investigación:**

Métodos teóricos

Análítico-Sintético: la aplicación de este método facilitó la búsqueda, investigación y análisis de los documentos necesarios para entender el proceso que se lleva a cabo en la gestión de la información, tanto a nivel mundial como en las distintas instituciones del país, se consultaron disímiles fuentes bibliográficas.

Análisis Histórico-Lógico: utilizado para determinar el estado del arte y construir otros referentes teóricos que sustentan la investigación.

Métodos empíricos

Entrevista individual semiestructurada: se utilizó para recopilar, analizar y conformar la información a través del personal que labora en los procesos relacionados con la documentación, para establecer criterios en vista al desarrollo del módulo, garantizando así, el perfeccionamiento de la solución propuesta relacionada con el funcionamiento de la gestión de la información en la empresa.

El documento consta de tres capítulos en los que se abordan diferentes temáticas. Estos están estructurados de la siguiente forma:

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN. En este capítulo se puntualizan los principales conceptos relacionados con el tema, se realiza el estudio del estado del arte a nivel nacional e internacional sobre los sistemas de gestión de la información. Además, se analizan las herramientas y lenguajes de programación idóneos para el desarrollo del módulo

Introducción

Gestor de Archivos para el sistema de gestión de TRANSCARGO, así como la metodología a emplear en el desarrollo del mismo.

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA. En este capítulo se realiza el análisis y diseño de la propuesta de solución, la relación de actores del sistema, la descripción de requisitos por procesos, el diagrama de despliegue y el diseño de la base de datos.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS. En este capítulo se exponen aspectos a tener en cuenta en la implementación del módulo, la estrategia de pruebas diseñadas para comprobar el cumplimiento de los objetivos trazados y el análisis de los resultados obtenidos luego de su aplicación.

Capítulo 1. Fundamentación teórica de la investigación

La base teórica del presente trabajo está sustentada sobre los sistemas de gestión que sirven de punto de partida para la investigación. Además, el estudio de las herramientas y tecnologías a utilizar durante el desarrollo de la propuesta de solución, sirven de apoyo para que toda la información recopilada oriente y guie conceptualmente el desarrollo de la investigación.

1.1 Sistema integral de gestión empresarial

ERP (*Enterprise Resource Planning*) es un sistema integral de gestión empresarial diseñado para modelar y automatizar la mayoría de los procesos de una empresa. Además, unifica y ordena la información en un solo lugar posibilitando la toma de decisiones de forma más rápida y segura. Los ERP son el núcleo de otras aplicaciones como CRM (Gestión de relaciones con los clientes), *Data Mining* (Conversión de datos en información útil) (EXXIS, 2010).

ERP se define según dos principios básicos:

- Aplicaciones informáticas como módulos independientes, pero perfectamente compatibles con una única base de datos común.
- El uso de un motor de flujos de trabajo debe definir las tareas de un proceso y gestionar su aplicación en todos los módulos del sistema.

Principales ventajas de los sistemas ERP:

- Permiten agilizar los procesos empresariales gracias a la capacidad de obtener información y procesarla de forma inmediata.
- Permiten disminuir los tiempos y los costes de los procesos.
- Eliminan la posible duplicidad de datos y operaciones innecesarias.
- Proporcionan información fiable e íntegra.
- Crean una visión unificada de la empresa, común a todos los departamentos.

Implantar un ERP y automatizar los procesos ahorra costes y aumenta la productividad de la empresa. Para escoger un ERP que se adapte a las necesidades de la empresa y sus condiciones se analizarán algunos sistemas existentes.

1.2 Análisis de soluciones similares

En el mundo, el desarrollo de las tecnologías va aparejado al incremento de sistemas para la informatización de los procesos en las distintas esferas. La necesidad de comprender y analizar la

trascendencia de los sistemas informáticos relacionados con la gestión de la información, en el ámbito internacional y nacional, es un factor clave para la realización de un nuevo sistema.

1.3 Análisis de soluciones similares a nivel nacional

En Cuba se han desarrollado varias aplicaciones que gestionan la información. A continuación se expone la solución más importante encontrada a través de la revisión bibliográfica realizada.

eXcriba

El GDA eXcriba o Gestor de Documentos Administrativos eXcriba, es un sistema basado en el ECM (Enterprise Content Management) Alfresco, que lleva su contenido a lo largo de su ciclo de vida, permitiendo la gestión de los múltiples documentos de trabajo, ya sean documentos de archivo o administrativos. Su funcionamiento se rige por normas tales como: la Norma ISO 15489 y la Norma ISAD (G), así como el Modelo de Requisitos para la Gestión de Documentos Electrónicos (MoReq). Es un producto informático genérico, que se ha propuesto desde su comienzo el cumplimiento de normas, desde su concepción hasta la personalización y despliegue del producto mediante soluciones a la medida en distintas organizaciones (Vidal, 2016).

El objetivo principal del producto es automatizar los procesos documentales y archivísticos que se ejecutan dentro de cualquier entidad, desde la elaboración de un documento en su fase de inicio hasta su conservación en el archivo de gestión. El sistema incluye todas las acciones sobre los documentos tales como: crear, clasificar, describir mediante la Norma ISAD(G), control de versiones, definir tipologías documentales, gestionar flujos documentales, almacenar documentos en diferentes formatos electrónicos, gestionar los trámites de los documentos que se generan o reciben, salvaguardar el patrimonio documental y automatizar el cuadro de clasificación. Este *software* es la propuesta que se ha optado para implantar el componente tecnológico de un sistema de gestión integral de documentos y archivos. El mismo está compuesto por un módulo para la interfaz de usuario (Aplicación Web), el ECM Alfresco como núcleo del sistema informático y herramientas de apoyo para la administración de la solución como valor añadido. Todos estos componentes enunciados permiten poder llevar a los clientes las bondades que el Alfresco provee como repositorio documental y a la vez ofrecerles una interfaz amigable para interactuar entre el usuario y el *software*. Veamos a continuación algunas de las funcionalidades (eXcriba, 2016):

- Automatización de los flujos documentales
- Gestión de documentos

- Gestión de carpetas
- Control de versiones
- Control de acceso y permisos
- Notificaciones

1.4 Análisis de soluciones similares a nivel internacional

En diferentes países se han desarrollado varias aplicaciones que gestionan la información. A continuación se exponen algunas de las soluciones más importantes encontradas a través de la revisión bibliográfica realizada.

BKMplus:

Con BKMplus cada departamento puede gestionar sus procesos aportando eficiencia y eficacia en la búsqueda y en la recuperación de documentos y expedientes. El sistema permite buscar por todo el contenido del documento (PDF, Word, Excel) y también por sus metadatos (fecha, departamento, autor, cliente) elimina el tiempo dedicado a la clasificación de la documentación, encuentra de manera rápida los documentos y controla los procesos de trabajo, asegurando en todo momento su ejecución y el correcto funcionamiento de la organización.

Este sistema es la evolución tecnológica de un gestor documental hacia la gestión del documento/expediente electrónico, siendo uno de sus principales objetivos facilitar el paso de la gestión de la documentación en papel a la electrónica. Se fundamenta con las siguientes características (Pérez, 2014):

- Repositorio centralizado de documentación electrónica.
- Organización y clasificación automática o manual.
- Gestión de procesos.
- Búsqueda y recuperación eficaz de documentos.
- Firma electrónica y sellado de tiempo.
- Gestión de usuarios y control de seguridad.
- Gestión de las publicaciones de los documentos.
- Control de versiones.
- Amplias posibilidades de integración con otras herramientas.

Alfresco:

Alfresco ofrece una gestión de documentos que utiliza interfaces habituales para que el usuario las adopte rápidamente, basadas en un repositorio que proporciona servicios transparentes e invisibles para una completa gestión de contenido empresarial. Permite sustituir las unidades

compartidas y ofrece la misma interfaz. Permite configurar reglas de conexión para automatizar el procesamiento manual y ofrece una compatibilidad invisible, busca directamente desde Firefox o IE7. Realiza extracción y categorización automática de metadatos, espacios para la colaboración con procedimientos recomendados y asistencia de ciclo de vida transparente (Alfresco, 2017).

Athento:

Athento permite importar documentos de distintas maneras: Desde Dropbox, correo electrónico, arrastrar y soltar desde la interfaz de usuario, carpetas monitorizadas, FTPs, subida de metadatos desde archivos CSV y sincronización con un repositorio CMIS. Soporta diferentes tipos de documentos e imágenes (PDF, TIFF, PNG, JPG), permite exportar documentos y metadatos en distintos formatos y permite importar desde un fichero CSV (Athento, 2014).

Conclusiones de las soluciones similares

Después de haber realizado un análisis de los sistemas similares a la solución propuesta, se llega a la conclusión de que ninguno puede ser utilizado como solución a la problemática planteada. Solo cumplen parcialmente algunas de las exigencias por lo que se decide utilizarlas como punto de referencia a la hora de elaborar la propuesta de solución.

Se tomaron elementos utilizados por estos sistemas, ejemplo de ello son operaciones como copiar, mover y borrar. El eXcriba facilitó el esquema en forma de árbol que permite la búsqueda, visualización e identificación de ficheros, cada nodo representa una carpeta de archivos, desde las cuales, se van ramificando otras vertientes y nodos que irán representando sub-carpetas y archivos. Athento brindó la posibilidad de exportar documentos y metadatos en distintos formatos así como importarlos desde un fichero CSV. El análisis de las soluciones similares permitió mayor grado de comprensión relacionado con el funcionamiento de la gestión de la información en una institución.

1.5 Metodología, lenguajes, herramientas, tecnologías y marco de trabajo

El módulo que se propone fue diseñado para integrarse al SIGAX, esto condicionó la reutilización de la metodología de desarrollo, los lenguajes, tecnologías y marcos de trabajo empleados en el desarrollo de dicho módulo. Considerando el estudio realizado por los especialistas del proyecto TRANSCARGO, se asume y adopta la metodología de desarrollo, los lenguajes, herramientas y tecnologías utilizadas por el equipo de desarrollo del SIGAX.

1.5.1 Metodologías de desarrollo de software

Una metodología de desarrollo de *software* se refiere al entorno que se usa para estructurar, planificar y controlar el proceso de progreso de un sistema de información. Consiste en una filosofía con una base de procesos de desarrollo de *software* como múltiples herramientas, modelos y métodos, para asistir en este proceso. Cada metodología tiene su propio enfoque de cómo se debe de consistir un proyecto de desarrollo de *software*. Pero todas ellas se basan en una serie de enfoques generalistas (Hermenegildo, 2012).

AUP variación UCI

Proceso ágil unificado, por sus siglas en inglés *Agil Unified Process*, en su versión UCI. Esta metodología propone cuatro fases (inicio, elaboración, construcción, transición) para el ciclo de vida de los proyectos de la UCI mantiene la fase de inicio, pero modificando el objetivo de la misma, se unifican las restantes tres fases de AUP en una sola, a la que se llamó ejecución y se agrega la fase de cierre (SANCHEZ RODRIGUEZ, 2015).

AUP propone siete disciplinas (modelado, implementación, prueba, despliegue, gestión de configuración, gestión de proyecto y entorno), se decide para el ciclo de vida de los proyectos de la UCI tener ocho disciplinas, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: modelado de negocio, requisitos y análisis y diseño en AUP están unidos en la disciplina modelo, en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina Implementación, en el caso de prueba se desagrega en tres disciplinas: pruebas internas, de liberación y aceptación y la disciplina despliegue se considera opcional. Las restantes tres disciplinas de AUP asociadas a la parte de gestión para la variación UCI se cubren con las áreas de procesos que define CMMI-DEV (Capacidad de Integración de Madurez del Modelo, por su significado en español) v1.3 para el nivel 2, serían CM (Gestión de la configuración), PP (Planeación de proyecto) y PMC (Monitoreo y control de proyecto) (SANCHEZ RODRIGUEZ, 2015).

AUP propone nueve roles (administrador de proyecto, ingeniero de procesos, desarrollador, administrador de BD (Base Datos), modelador ágil, administrador de la configuración, *Stakeholder* (experto en dominio), administrador de pruebas, probador). Se decide para el ciclo de vida de los proyectos de la UCI tener once roles, manteniendo algunos de los propuestos por AUP y unificando o agregando otros.

Capítulo 1

En la variación de la metodología AUP-UCI, la disciplina Modelado de negocio propone tres variantes a utilizar en los proyectos (Casos de Uso del Negocio (CUN), Descripción de Proceso de Negocio (DPN) y Modelo Conceptual (MC). Además, existen tres formas de encapsular los requisitos Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por proceso (DRP), agrupados en cuatro escenarios condicionados por el Modelado de negocio. A partir de las variantes anteriores se condicionan cuatro escenarios para modelar el sistema en la disciplina Requisitos, manteniendo en dos de ellos el MC, quedando de la siguiente forma (SANCHEZ RODRIGUEZ, 2015):

Escenario No 1:

Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.

$CUN + MC = CUS$

Escenario No 2:

Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.

$MC = CUS$

Escenario No 3:

Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.

$DPN + MC = DRP$

Escenario No 4:

Proyectos que no modelen negocio solo pueden modelar el sistema con HU.

HU

El desarrollo de la propuesta de solución se enmarca en el escenario 3 aunque no exista proceso de negocio en la presente investigación, pero como es el seleccionado por el proyecto la investigación tiene que seguir las líneas de desarrollo para que exista homogeneidad en el trabajo en equipo.

1.5.2 Lenguaje Unificado de Modelado

Es un lenguaje que permite modelar cualquier sistema a través de diagramas, permite realizar presentaciones gráficas que contiene la información relevante del sistema. Se utiliza para

especificar, visualizar, construir y documentar artefactos de un sistema de *software*. Se usa para entender, diseñar, configurar y controlar la información sobre los sistemas (Castro, 2008).

Propiedades que hacen de UML un lenguaje de modelado estándar:

- Concurrencia, es un lenguaje distribuido y adecuado a las necesidades de conectividad actuales y futuras.
- Ampliamente utilizado por la industria desde su adopción por el Grupo de Gestión de Objetos (OMG, por sus siglas en inglés).
- Reemplaza a decenas de notaciones empleadas con otros lenguajes.
- Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clases, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.
- Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboración, que sirven para evaluar el estado de las máquinas.

Para la realización de este trabajo se requiere de un lenguaje gráfico, con el objetivo de especificar el módulo de *software* de forma estándar. Este lenguaje permite que todo *software* de diseño, se visualice, especifique y documente con un lenguaje de forma común. Además, se ajusta a la metodología seleccionada.

1.5.3 Herramienta para el modelado

Ingeniería del Software Asistida por Computadora (por sus siglas en inglés CASE) comprende un amplio abanico de diferentes tipos de programas que se utilizan para ayudar a las actividades del proceso de *software*, como el análisis de requerimientos, el modelado de sistemas, la depuración y las pruebas (CASTRO, 2008).

Visual Paradigm

Es una herramienta CASE que usa UML (*Unified Modeling Language*) como lenguaje de modelado y es multiplataforma. La herramienta está diseñada para una amplia gama de usuarios, incluidos los ingenieros de *software*, analistas de sistemas, analistas de negocios y arquitectos de sistemas, o para cualquier persona que esté interesada en la construcción de sistemas de *software* fiables a gran escala con un enfoque orientado a objetos (CASTRO, 2008).

1.5.4 Lenguaje del lado del cliente. JavaScript

JavaScript es un lenguaje script ligero, multiparadigma, basado en prototipos, dinámico, imperativo e interpretado. Además, es orientado a objetos y generalmente implementado del lado del cliente como parte de un navegador web (Mozilla Developer Network, 2015).

XML (Extensible Markup Language)

Es una especificación/lenguaje de programación desarrollada por el W3C. Es una versión de SGML (Standard Generalized Markup Language) diseñado especialmente para los documentos de la web. Permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones.

Algunas de las ventajas de XML (UDIMA, 2009):

- Separa radicalmente el contenido y el formato de presentación.
- Diseñado para cualquier lenguaje y alfabeto.

Características de este lenguaje:

- Es un subconjunto de SGML que incorpora las tres características más importantes de este:
 - Extensibilidad, Estructura y Validación.
 - Basado en texto.
 - Orientado a los contenidos no a presentación.
 - Las etiquetas se definen para crear los documentos, no tienen un significado preestablecido.
- No existe un visor genérico de XML.

CSS

Hojas de Estilo en Cascada (Cascading Style Sheets, CSS), es un lenguaje utilizado para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo y funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos (Sitio Web de la Oficina Española del W3C, 2016).

1.5.5 Lenguajes del lado del servidor. Python

Es un lenguaje de *scripting* independiente de plataforma y orientado a objeto, preparado para realizar cualquier tipo de programa. Python es interpretado, lo que trae consigo que no necesita compilar los programas para ejecutarlos, por lo que ofrece como ventajas la rapidez de desarrollo e inconvenientes como una menor velocidad (ALEGSA, 2014).

En los últimos años este lenguaje se ha hecho muy popular, algunas de las principales razones son (ALEGSA, 2014):

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje.
- La sencillez y velocidad con que se crean los programas.
- La cantidad de plataformas en las que es posible desarrollar, como Unix, Windows y Mac.
- Además, es gratuito hasta para propósitos empresariales.

1.5.6 Marco de trabajo de desarrollo. OpenObject

Es un *framework* (marco de trabajo) de código abierto, rápido e inteligente para el desarrollo de aplicaciones en Python. Es el marco de referencia utilizado para la construcción y desarrollo de módulos de OpenERP como son: CRM, servicio de asistencia, producción, proyectos, eventos, acciones y recursos humanos (ADINS, 2011).

Algunas características de este marco de referencia:

- Capacidad de un módulo para llevar la modificación a los objetos y las vistas existentes. El concepto de módulo OpenObject hace posible utilizar la misma base de código para la instalación múltiple, por ejemplo, diferentes filiales, sin embargo, permite la personalización y funcionalidades de plataforma por base de datos.
- Basado en la web y cliente-servidor por el mismo precio.
- Las vistas se definen en lenguaje basado en XML y OpenObject sobre la marcha convierte estas en una aplicación cliente-servidor o el cliente basado en Web.
- Permite al desarrollador construir rápidamente aplicaciones con diferente tipo de vista: la forma, la búsqueda, árbol lista-, calendario y Kanban, Gantt.
- Está construido alrededor de Python:
- Moderno, orientado a objetos.
- Portátil (Linux, Windows, Mac).

OpenObject ofrece, en un solo paquete el componente básico para la construcción de una aplicación de negocios:

- Mapeo objeto-relacional (ORM): simplifica el acceso a la base de datos.
- Sistema de permiso: completo sistema de permisos, menús, campos, acceso a datos, acciones comerciales.
- Multi-idioma / internacionalización: unicode UTF-8, varios idiomas por base de datos, los datos temporales almacenados en UTC.
- Campos traducibles: campos de texto pueden ser marcados como traducibles.
- Motor de informes: incorpora un motor de informes con la generación de PDF.
- Los servicios Web: ofrece sus servicios a través de XML-RPC y JSON (Javascript).
- Proporciona un marco integrado de aplicación para las pruebas.
- Trabaja con el sistema de gestión de base de datos de código abierto PostgreSQL.
- Tamaño máximo de base de datos ilimitado.
- Máximo Tabla Size 32 TB.
- Máximo Fila Size 1.6 TB.
- Máximo Campo Tamaño 1 GB.

Este *framework* está escrito sobre Python, permite el almacenamiento de información en la base de datos mediante PostgreSQL. Además, incluye una serie de funcionalidades primarias como generación de PDF, motor de correo.

1.5.7 Sistema de planificación de recursos empresariales. Odoo

Odoo (conocido anteriormente como OpenERP) es un sistema ERP de código abierto y sin coste de licencias que cubre las necesidades de las áreas de: contabilidad y finanzas, ventas, recursos humanos (RRHH), compras, proyectos, almacenes y fabricación (Odoo – OpenERP – ERP, 2012).

Cuenta con varios módulos base como son:

- Gestión de relaciones con el cliente (CRM).
- Gestión de proyectos.
- Gestión de almacenes.
- Gestión contable y financiera.
- Gestión de compras.
- Gestión de ventas.
- Recursos humanos.
- Mercadotecnia.
- Fabricación.
- Gestión del conocimiento.

Además, de los módulos oficiales existen más de 500 que completan el programa y le permitirá adaptar Odoo a las necesidades del negocio. También dispone de soluciones verticales para sectores como sanidad, hostelería, ingeniería civil, industria alimentaria (Odoo S.A).

Posee fortalezas que lo hacen estar por encima de sus homólogos:

- Libertad: sin dependencia del proveedor. Como producto no pertenece a ninguno de sus distribuidores, es decir, tiene la opción de elegir el proveedor que más se ajuste a sus necesidades.
- Filosofía Open: puede contratar únicamente lo que necesite. Desarrollo de algún módulo en específico o formación/soporte técnico de algún módulo oficial.
- Código abierto: puede disponer del código para realizar cualquier mejora a algún módulo existente o crear uno nuevo adaptado a las necesidades.
- Conectividad: visualización de informes en formato estándar PDF, importación/exportación con MS Office, Excel o CVS y la posibilidad de conexión con casi cualquier software utilizando servicios web.
- Gratuito: es un producto que no tiene coste de licencias.
- Multiplataforma: la interfaz Web de Odoo le permite acceder desde cualquier ordenador independiente del sistema operativo (GNU/Linux, Mac OS X o Windows) incluso tablets y smartphone con Android o IOS.
- OpenObject: el framework de OpenERP permite el desarrollo rápido de funcionalidades o conectividad con otras plataformas.
- Variedad: cuenta actualmente con una gran cantidad de módulos liberados que se pueden combinar y/o servir como base para construir casi cualquier solución vertical.
- Integración: módulos de gestión y aplicaciones web propias para construir portales, tiendas en línea, foros o eventos; todo dentro de una plataforma única y control unificado.
- Fácil migración: la herramienta oficial importa y exporta datos maestros en formato osv que le resulta más sencillo seguir trabajando con los datos desde su aplicación actual.

1.5.8 Sistema gestor de base de datos. PostgreSQL

Es un sistema de gestor de base de datos (SGBD) objeto-relacional, con su código de fuente disponible libremente. Utiliza un modelo cliente/servidor y usa multiprocesos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (MARTINEZ, 2010).

Dispone de una gran cantidad de características, algunas de ellas son:

- Integridad referencial.

- Completa documentación.
- Acceso encriptado vía SSL.
- Disponible para Linux y Unix en todas sus variantes (AIX, BSD, SGI, Mac OS X) y Windows 32/64 bit.

Este SGBD posee características que lo hacen destacarse por encima de otros gestores de base de datos como es la manipulación de multiprocesos y es el gestor que utiliza Odoos para su manejo con los datos.

1.5.9 Entorno integrado de desarrollo. PyCharm

Desarrollado por la compañía JetBrains, está basado en IntelliJ IDEA, el IDE de la misma compañía, pero enfocado hacia Java y la base de Android Studio. PyCharm tiene cientos de funciones que lo puede ver como una herramienta pesada, pero ayuda con el desarrollo del día a día (JetBrains, 2016).

Algunas de las características de PyCharm:

- Autocompletado, resaltador de sintaxis, herramienta de análisis y refactorización.
- Integración con framework web como: Django, Flask, Pyramid, Web2Py.
- Soporta entornos virtuales e intérpretes de Python 2.x, 3.x, PyPy, Iron Python y Jython.
- Sistemas de control de versiones: Git, CVS, Mercurial.
- Frameworks javascripts: jQuery, AngularJS.
- Debugger avanzado de Python y Javascript.

1.6 Conclusiones del capítulo

- El estudio de sistemas similares arribó a que estos cumplen algunas de las exigencias, pero no son integrables al sistema, por lo que se demuestra la necesidad de dicho módulo.
- Se describió la metodología, herramientas y tecnologías que permitirá guiar el proceso de diseño e implementación del módulo Gestor de Archivos.

Capítulo 2. Propuesta de solución

En el siguiente capítulo se lleva a cabo la descripción de la solución propuesta para el presente trabajo. Además, se presentan los requerimientos que debe tener el módulo divididos en requisitos funcionales, que muestran las funcionalidades específicas que serán implementadas, representadas a través de los casos de uso y los requisitos no funcionales, que definirán las características y restricciones que deberá cumplir la aplicación para poder funcionar correctamente. También se incluyen los patrones de arquitectura y diseño que se tuvieron en cuenta, más una descripción del diseño a través de los diagramas de clases y el modelo lógico de datos.

2.1 Propuesta de solución

Se propone como solución el desarrollo del módulo Gestor de Archivos, el cual pretende satisfacer la necesidad de gestionar documentos de forma electrónica, permitiendo al usuario que interactúe con el mismo trabajar de forma sencilla sobre las funcionalidades que brindará.

El objetivo principal del producto es centralizar la información y automatizar los procesos documentales y archivísticos que se ejecutan dentro de la entidad, desde la elaboración de un documento en su fase de inicio hasta su conservación. El módulo incluye todas las acciones sobre los documentos tales como: crear, clasificar, exportar, gestionar flujos documentales, almacenar documentos en diferentes formatos electrónicos, gestionar los trámites de los documentos que se generan o reciben y salvaguardar el patrimonio documental.

El módulo brindará la posibilidad de añadir archivos y directorios, mostrando un listado de estos, así como editar información de los mismos, ya sea por error o para añadirle algún dato o información. También permitirá realizar operaciones con los directorios y documentos así como la colaboración entre los empleados, socios, clientes y en general entre los usuarios que comparten información, de modo que esta puede ser reutilizada, por los demás usuarios. Garantizará el cumplimiento legal y regulatorio de los documentos, reduciendo los costes mediante la racionalización de procesos y la normalización.

Con el desarrollo del módulo propuesto se espera obtener los siguientes beneficios:

- Conservación de la información.

- Salvaguardar el patrimonio documental presente y futuro, dado que el tratamiento de los documentos en su primera fase de vida incrementa la calidad en su conservación y posterior difusión.
- Garantizar la protección de los datos ante cualquier falla o error ocasionado mediante salvadas de seguridad.
- Optimizar los procesos de gestión documental: acceso, búsqueda, manipulación de documentos, resultados de eficacia y rentabilidad administrativa de todas las operaciones.
- Compartir y aprovechar la información como un recurso colectivo.

2.2 Modelo de dominio

El modelo de dominio es utilizado por el analista como un medio para comprender el módulo, debido a que es una representación de los conceptos y elementos de la vida real que serán usados en el mismo. Se crea con el fin de representar los conceptos claves del dominio del sistema, las propiedades más importantes y las relaciones entre los conceptos facilitando una mejor comunicación entre desarrolladores y el cliente (LARMAN, y otros, 2003). A continuación, se muestra el diagrama conceptual del modelo de dominio del módulo Gestor de Archivos.

2.2.1 Diagrama del modelo de dominio

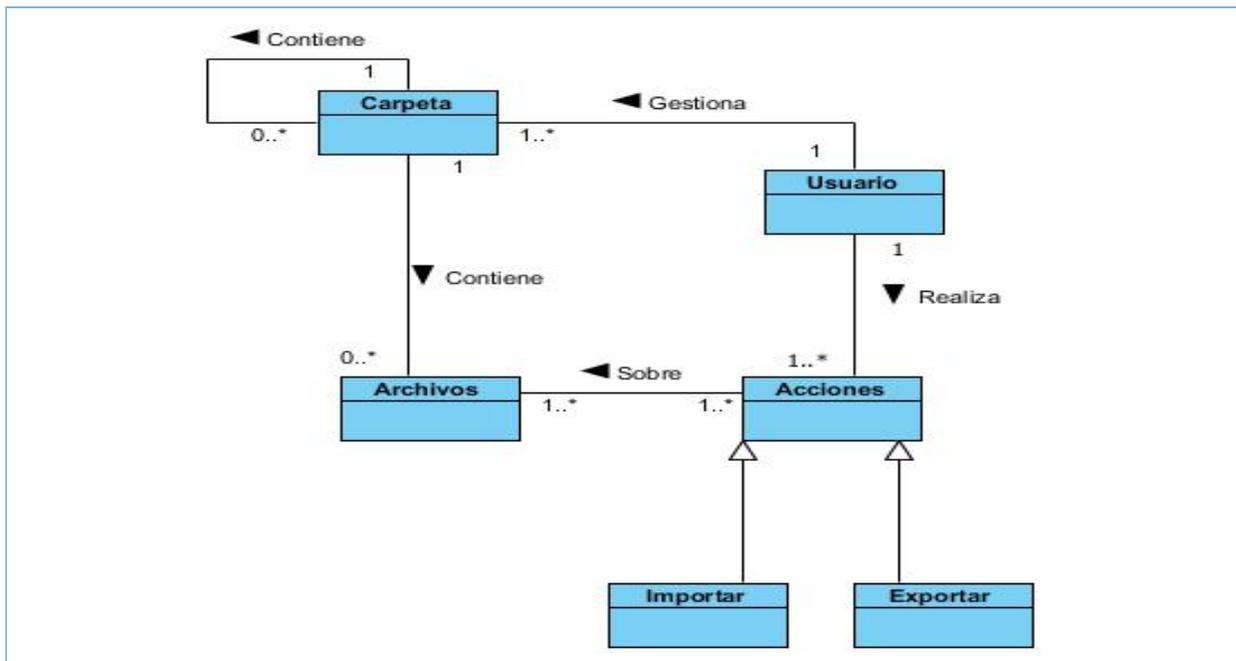


Figura 1: Modelo conceptual

A continuación, se describen cada una de los conceptos del modelo de dominio:

Usuario: identifica a las personas que tienen acceso al módulo con el objetivo de gestionar la información.

Carpeta: contiene archivos y subdirectorios.

Archivos: unidad de datos o información almacenada que tiene un nombre propio y una extensión que lo identifica.

Acciones: operaciones de importar y exportar que se realizan a los archivos.

2.3 Requisitos del sistema

La ingeniería de requisitos, como todas las demás actividades de la ingeniería del *software*, debe adaptarse a las necesidades del proceso, el producto y las personas que realicen el trabajo. Además, proporciona el mecanismo apropiado para entender lo que el cliente desea, analizar las necesidades, evaluar la factibilidad, validar la especificación y administrar los requisitos (PRESSMAN, 2005).

Los requerimientos para un módulo son la descripción de los servicios proporcionados por el módulo y sus restricciones operativas. Los requerimientos del módulo son para designar la descripción detallada de lo que el módulo debe hacer. Establecen las funciones, servicios y restricciones operativas del módulo y su descripción debe ser precisa, debe definir exactamente qué es lo que se va a implementar. Estos requerimientos se clasifican en funcionales y no funcionales o como requerimientos del dominio (Sommerville, 2005).

2.3.1 Requisitos funcionales

Los requisitos funcionales definen las funcionalidades del módulo. Estos son los que dictan lo que debe realizar el módulo.

Tabla 1: Requisitos funcionales

RF1- Crear carpetas	RF9- Compartir archivos y carpetas
RF2- Editar carpetas	RF10- Insertar color a las carpetas
RF3- Mover carpetas	RF11- Eliminar archivos
RF4- Listar carpetas	RF12- Crear subcarpetas
RF5- Eliminar carpetas	RF13- Adjuntar subcarpetas y archivos
RF6- Importar archivos	RF14- Buscar carpetas o archivos
RF7- Exportar archivos	RF15- Importar carpetas
RF8- Listar archivos	RF16- Exportar carpetas

2.3.2 Requisitos no funcionales

Son aquellos que imponen restricciones en el diseño, la implementación y estándares de calidad. Para especificar los requerimientos fueron definidas categorías encargadas de recoger las cualidades que debe tener el módulo para su correcto funcionamiento y la prestación adecuada de sus servicios.

Usabilidad: el módulo debe cumplir con las pautas de diseño establecidas en la Estrategia Marcaria de la Universidad y las establecidas por el diseñador del proyecto. Cuando se crea, actualiza, cancela y se elimina un elemento el módulo muestra un mensaje con el resultado de la acción. Cuando el puntero del *mouse* se ubica sobre un botón o hipervínculo cambia el color del elemento.

Soporte: la aplicación debe ser implementada sobre tecnología web, ya que será accedida a través de intranet. Ejecutarse sobre navegadores web, siendo compatible con:

- Mozilla Firefox 7.0 o superior
- Opera 10.0.0 o superior
- Chrome 7.0 o superior.

Portabilidad: ser instalado en cualquier entorno que cuente con un servidor de base de datos relacional con cualquier gestor de BD, se recomienda PostgreSQL v 9.2 o superior. Los servidores web deben tener como mínimo 2Gb de RAM y 100GB de HDD. Las estaciones clientes desde las

que se accederá al módulo pueden tener cualquier sistema operativo y solo deberán tener un navegador web moderno instalado y como mínimo 512 MB de RAM.

2.3.3 Descripción de requisitos por procesos

La descripción de requisitos muestra el flujo de eventos que debe tener cada requisito cuando se ejecuta, la secuencia de pasos y condiciones que se cumplen tanto por parte del usuario como del sistema.

A continuación, se presenta la descripción del requisito funcional correspondiente al RF: Crear carpeta. Para tener información sobre las demás descripciones de requisitos consultar los Anexos 3.

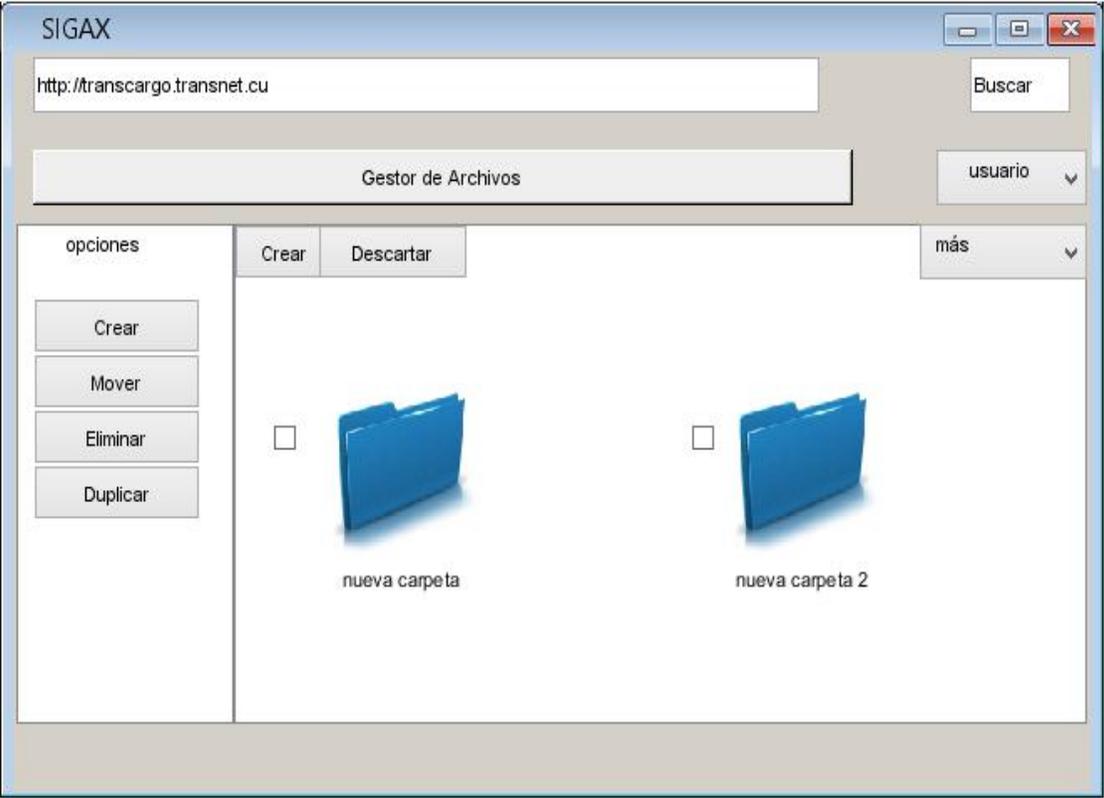
Tabla 2: Descripción de RF: Crear carpeta

Precondiciones	El usuario debe estar autenticado en el sistema. El usuario debe poseer los permisos para crear carpetas.
Flujo de eventos	
Flujo básico Crear carpeta	
1.	El usuario selecciona la opción crear carpeta.
2.	El módulo debe permitir introducir y/o seleccionar los siguientes datos para crear una carpeta: (*) Nombre: representa el nombre de la carpeta. Campo de texto, admite el siguiente formato, letras, números o símbolos, con excepción de " # % *: < > ? / \ . Color: representa el color de la carpeta. Campo de selección que incluye varios colores El módulo brinda las siguientes opciones: Guardar Descartar
3.	Se introducen y/o seleccionan los datos para crear una carpeta.
4.	El usuario selecciona la opción Guardar.
5.	El módulo <i>valida los datos</i> .
6.	El módulo guarda los datos de la nueva carpeta.
7.	Concluye así el requisito.

Capítulo 2

Pos-condiciones	
	Se incluyó una nueva carpeta satisfactoriamente.
Flujos alternativos	
Flujo alternativo 6.a Información incompleta	
	El módulo señala el o los campos obligatorios que no hayan sido introducidos y/o seleccionados y muestra el mensaje de información: Los siguientes campos son obligatorios; indicando los campos en cuestión.
	El usuario introduce y/o selecciona los datos.
	Volver al paso 2 del flujo básico.
Pos-condiciones	
1.	NA
Flujos alternativo 6.b Información incorrecta	
	El módulo señala el o los campos incorrectos y muestra el mensaje de información: Los siguientes campos son incorrectos; indicando los campos en cuestión.
	El usuario introduce y/o selecciona correctamente los datos.
	Volver al paso 2 del flujo básico.
Pos-condiciones	
1.	N/A
Flujo alternativo 4.a Descartar cuando se hayan introducido datos	
1.	El usuario selecciona la opción Descartar.
2.	El módulo muestra un mensaje de información: Sus cambios se perderán y solicita confirmación ¿Está seguro que quiere abandonar la página? El módulo brinda las siguientes opciones: Aceptar Cancelar
3.	El usuario selecciona la opción Aceptar.
4.	El módulo elimina los datos introducidos y regresa al listado de carpetas.
5.	Concluye así el requisito.

Capítulo 2

Pos-condiciones	
1.	No se crea la carpeta.
Validaciones	
	NA
Conceptos	NA
Requisitos especiales	
Asuntos pendientes	NA
Interfaz de usuario	



2.4 Arquitectura de Odo

Cliente/Servidor es una arquitectura distribuida que permite a los usuarios obtener acceso a la información aún en entornos multiplataforma. El esquema de funcionamiento de un sistema Cliente-Servidor es (Oposiciones TIC, 2011):

- El cliente solicita información al servidor y este recibe la petición.
- El servidor procesa dicha solicitud y envía el resultado al cliente.
- El cliente recibe el resultado y lo procesa.

Los elementos principales de esta arquitectura son (Oposiciones TIC, 2011):

- Cliente: permite al usuario formular los requerimientos y pasarlos al servidor. Encargado de las funciones de relacionadas con la manipulación y despliegue de datos, además de acceder a los servicios desde cualquier parte de la red.
- Servidor: permite atender a múltiples clientes que hacen peticiones de algún recurso. Encargado de las funciones relacionados con la mayoría de las reglas del negocio y los recursos de datos.

- **Middleware:** es un módulo intermedio que actúa como mediador entre sistemas. Es el intermediario entre el cliente y el servidor y se ejecuta en ambas partes.

La arquitectura de trabajo de Odoo está basada en Cliente/Servidor, esta tiene los componentes de la misma separados. El servidor se ejecuta separado del cliente, maneja la lógica del negocio y realiza el enlace entre la aplicación y la base de datos. El cliente es el que presenta la información a los usuarios y permite interactuar con el servidor (Bargaw, 2014).

- **Servidor y módulos:** la parte del servidor está escrita en Python y el cliente se comunica con él a través de interfaces XML-RPC. La parte del negocio se agrupa en módulos que son carpetas con estructura predefinida, con código Python y archivos XML. Estos definen la estructura de los datos, formularios, menús, procedimientos, etc. Los módulos se conforman con una sintaxis independiente del cliente, lo que permite que al crear nuevos objetos lo realizas para todos los clientes.
- **Clientes:** estos son rápidos ya que no cuentan con la parte del negocio y muestra los datos en una aplicación web implementado como un servidor HTTP, permitiendo a los usuarios conectarse mediante un navegador web.

En la Figura 2 se puede apreciar la representación de la arquitectura Cliente/Servidor en Odoo. Fig. 2.

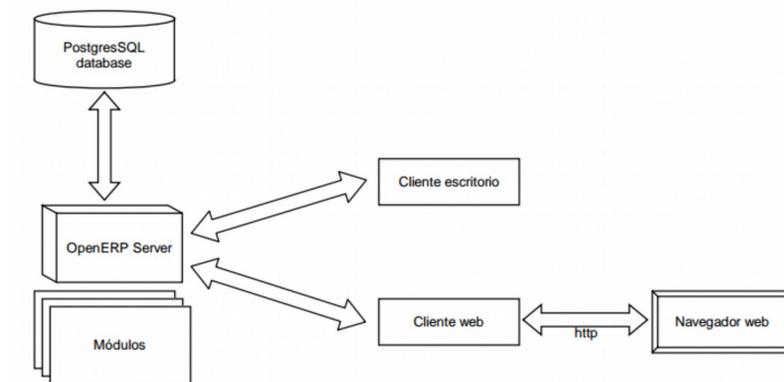


Figura 2: Arquitectura Cliente/Servidor en Odoo

2.5 Patrón arquitectónicos

El MVC o Modelo-Vista-Controladores es un patrón de arquitectura de *software* que separa la lógica de la aplicación de la lógica de la vista de una aplicación, garantizando así la actualización y

mantenimiento de forma sencilla de la aplicación y en un corto espacio de tiempo (Patrón Modelo-Vista-Controlador, 2012).

Los elementos del patrón son los siguientes (Patrón Modelo-Vista-Controlador, 2012):

- Modelo: es el encargado de los datos, es decir, generalmente consultando la base de datos.
- Vista: es la representación visual de cómo se van a ver los datos, es decir, la información relacionada con las interfaces se incluye en este elemento.
- Controlador: es el encargado de controlar, es decir, recibe orden del usuario y es el encargado de pedir los datos al modelo y comunicárselos a la vista.

A continuación, se muestra una representación del funcionamiento del patrón MVC en la Figura 3

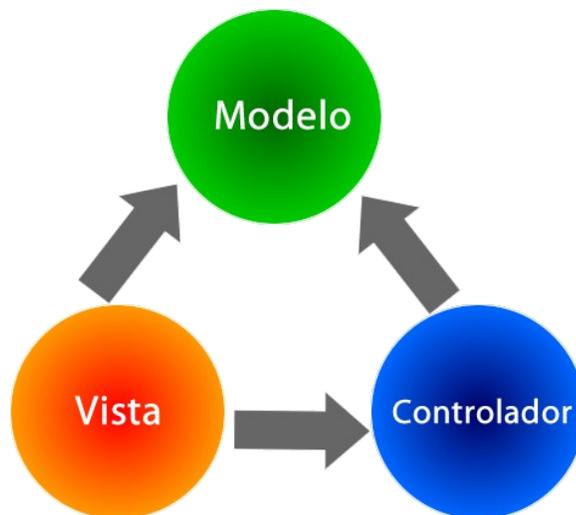


Figura 3: Modelo Vista Controlador

Odoo utiliza este patrón aprovechando al máximo todas las características del mismo donde:

- Modelo: los objetos de Odoo con sus columnas que normalmente se guardan en las tablas de PostgreSQL con sus campos. Permite la creación y actualización automática de las tablas y acceder a las mismas sin utilizar SQL.
- Vista: listas, formularios, calendarios, gráficos, definidas en archivos XML. En estos archivos también se definen, menús, acciones, informes y asistentes.

- Controlador: métodos Python definidos dentro de los objetos de Odoo que proporcionan la lógica: validación de datos y cálculos.

2.6 Diagrama de clases y secuencia del diseño

Los diagramas de clases del diseño son empleados para representar las relaciones que se establecen entre las clases, mientras que los de secuencia muestran la interacción de un conjunto de objetos en una aplicación a través del tiempo. A continuación, se presenta el diagrama de clase y secuencia del diseño correspondiente al requisito funcional del módulo Crear carpeta, el resto se encuentran disponibles en el Anexo 4 y Anexo 5 del documento (LARMAN, y otros, 2003).

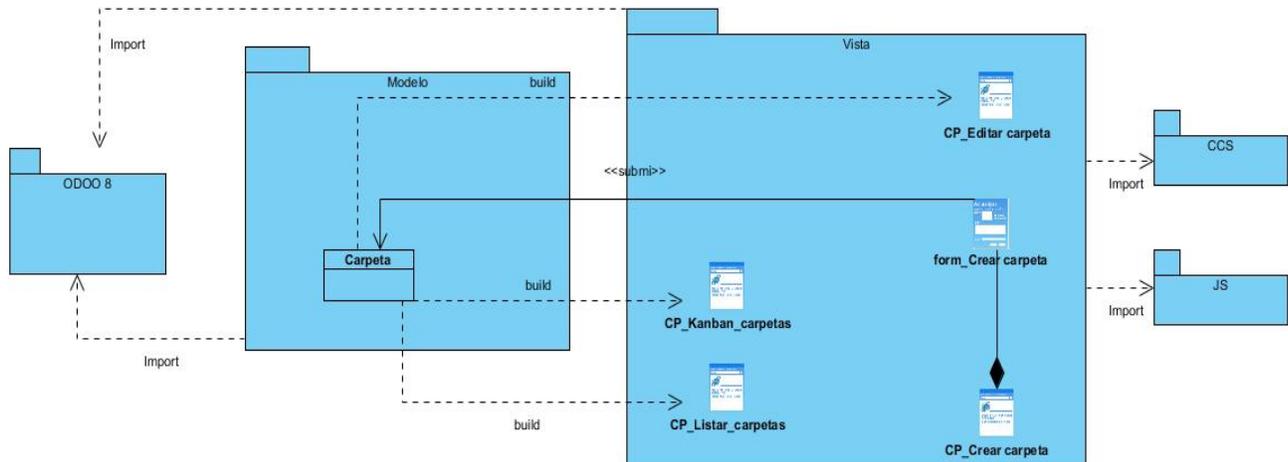


Figura 4: Diagrama de clases del diseño del RF Crear carpeta

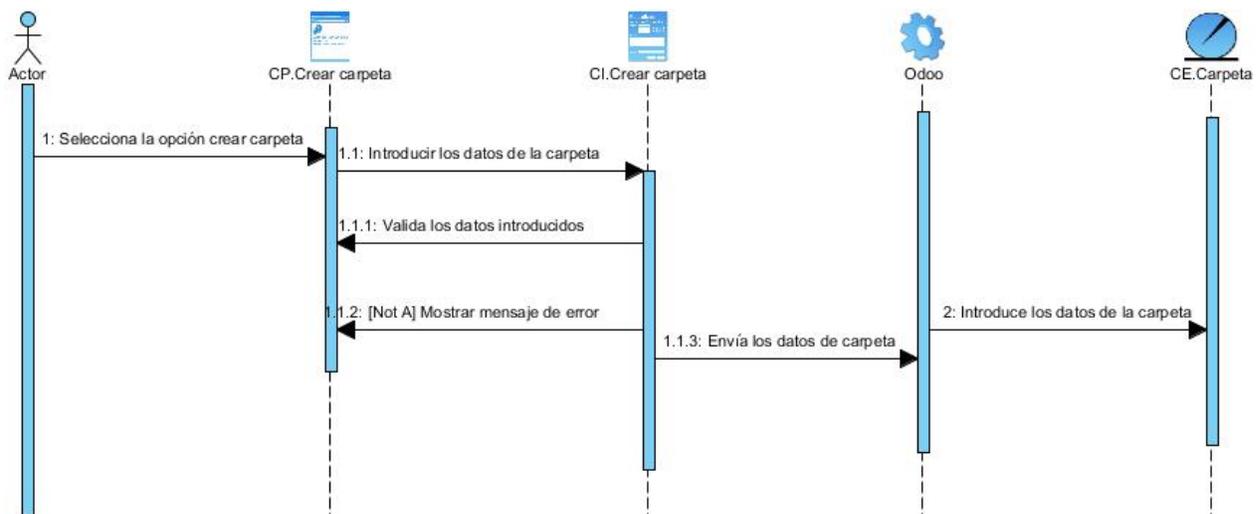


Figura 5: Diagrama de secuencia del diseño del RF Crear carpeta

2.7 Patrones de diseño

Un patrón de diseño es una solución estándar para un problema común de programación. En un *software* contribuye a la reutilización de su diseño, ellos nombran e identifican los aspectos claves de su estructura (LARMAN, 1999). Estas características garantizan una reducción de tiempo en el desarrollo y mantenimiento del módulo. Además, mejora la seguridad, eficiencia y consistencia de los datos.

2.7.1 Patrones GRASP

Los patrones de asignación de responsabilidades, conocidos como GRASP acrónimo de *General Responsibility Assignment Software Patterns*, los cuales tienen como objetivo fundamental orientar al diseñador en cómo asignar las responsabilidades a cada clase en diferentes circunstancias (LARMAN, 1999).

A continuación, se muestran los patrones GRASP utilizados en la solución propuesta, con una breve descripción y ejemplo de donde fueron utilizados.

- Experto: expresa que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. Se evidencia su uso en la figura 6, donde la clase <<models.py>> es la responsable en conocer toda la información necesaria de un directorio.
- Bajo acoplamiento: plantea que las clases deben estar lo menos ligadas entre sí que sea permisible. Posibilitando que se pueda modificar alguna clase teniendo la menor repercusión posible en el resto del módulo. Se evidencia la utilización de este patrón en el módulo, dado que la clase <<action.py>> hereda únicamente de <<models.py >> mostrando un bajo acoplamiento de clases.



Figura 7: Ejemplo de uso del patrón Experto

2.7.2 Patrones GOF

Los patrones GOF (Pandilla de los Cuatro) acrónimo de *Gang of Four*, los cuales constituyen un catálogo de veintitrés patrones de diseño publicados por Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides en el libro *Design Patterns: Elements of Reusable Object Oriented Software* (BEDOYA, 2011).

- Singleton: pertenece a la familia de patrones creacional, la cual se refiere a la creación de objetos. Su propósito consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso a ella. Analizando los módulos de OpenObject como objetos, pueden ser pensados como una clase con una instancia única, permitiendo identificar este patrón. Estos objetos no encapsulan datos, lo que separa un módulo de otro es el conjunto de funciones que contiene, permitiendo así la existencia de una única instancia para un módulo. Se evidencia en la implementación del módulo, cuando se crea un objeto de la clase models, se obtiene siempre una instancia única de este objeto.

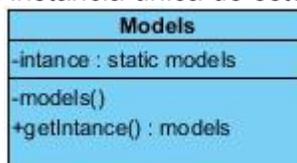


Figura 7: Ejemplo de uso del patrón Singleton

2.8 Modelo de datos

Un modelo de datos es un conjunto de conceptos utilizados para organizar los datos de interés y describir su estructura en forma comprensible para un sistema informático. En general un modelo de datos es la estructura o representación física de las tablas de la base de datos (Sommerville, 2005).

Para implementar la solución propuesta se añadieron dos tablas nuevas al modelo de datos del SIGAX. En la siguiente ilustración se presentan las tablas utilizadas en la base de datos del módulo Gestor de Archivos.

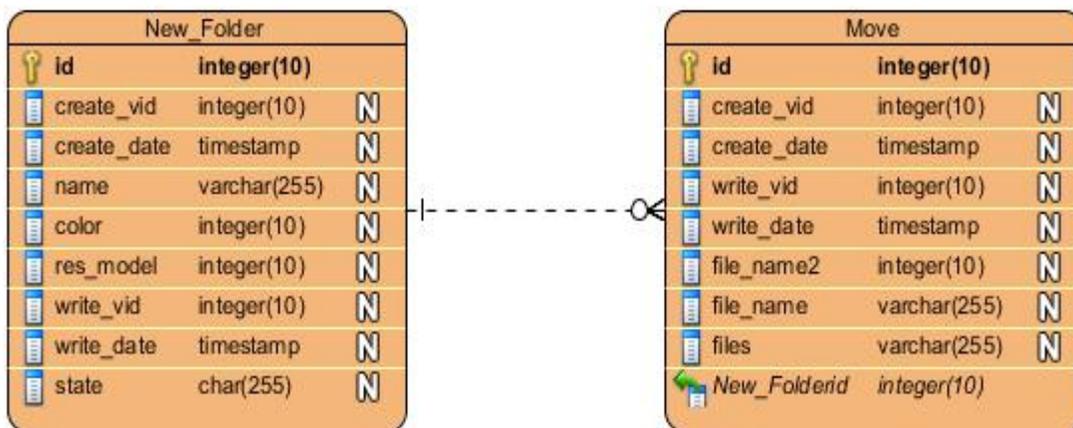


Figura 8: Modelo de datos

En la tabla 4 y tabla 5 se muestra la descripción de las tablas creadas en el modelo de datos.

Tabla 3. Descripción de las tablas.

Nombre	Descripción
New_Folder	Contiene todos los datos de las funcionalidades del módulo.
Move	Contiene todos los datos de la funcionalidad mover carpetas.

Tabla 4. Descripción de la tabla New_Folder.

Capítulo 2

Atributos	Tipo	Descripción
id	integer	Id necesario en cada entidad para las referencias en las relaciones entre tablas.
create_vid	integer	Nombre de usuario por el que fue creado.
create_date	timestamp	Fecha de cuando fue creado.
name	character	Nombre de la carpeta.
color	integer	Color que contiene la carpeta.
res_model	integer	Archivos relacionados.
write_vid	integer	Nombre del usuario que realizó la última actualización.
write_date	timestamp	Fecha de la última actualización.
state	character	Si está activo o no.

Tabla 5. Descripción de la tabla Move.

Atributos	Tipo	Descripción
id	integer	Id necesario en cada entidad para las referencias en las relaciones entre tablas.
create_vid	integer	Nombre de usuario por el que fue creado.
create_date	timestamp	Fecha de cuando fue creado.
file_name	character	Nombre de la carpeta destino.
file_name2	integer	Nombre de la carpeta a transferir.
res_model	integer	Archivos relacionados.
write_vid	integer	Nombre del usuario que realizó la última actualización.
write_date	timestamp	Fecha de la última actualización.
files	character	Archivos a transferir

2.10 Conclusiones del capítulo

- La identificación de los requisitos funcionales y no funcionales del sistema sirvió de guía para la implementación de las distintas funcionalidades del módulo Gestor de Archivos.
- La selección del patrón arquitectónico Modelo Vista Controlador permitió generar una estructura de sistema robusta y establecer una correcta relación entre sus componentes.
- El uso de los patrones de diseño permitió dar solución a problemas en la construcción del software, estandarizar y reutilizar código, para un mejor entendimiento y desarrollo del módulo.
- Los artefactos generados durante el diseño de la solución contribuyeron a un mejor entendimiento del sistema para dar paso a la implementación de la solución propuesta.

Capítulo 3. Implementación y Prueba

En el presente capítulo se describen los estándares de código fundamentales usados en la implementación de la solución propuesta. Además, se modela el diagrama de componentes para mostrar la organización y las dependencias lógicas que existen entre los ficheros que contienen código fuente. Una vez culminado el proceso de implementación se procede a aplicar un conjunto de pruebas a la aplicación informática.

3.1 Estándar de codificación

Un estándar de codificación son reglas que se siguen para la escritura del código fuente. Estos estándares permiten que otros programadores puedan identificar las variables, las funciones o métodos al leer los códigos de otras personas. Se definen estándares de codificación porque un estilo de programación homogéneo permite que todos los participantes lo puedan entender en menos tiempo y que cualquier persona que se desempeñe como codificador de dicho lenguaje pueda interpretar de manera eficiente (ÁLVAREZ, 2016).

A continuación, se exponen los estándares de codificación seguidos en el desarrollo del módulo:

Directorios: este estándar permite la organización de los directorios que están integrados al módulo, los cuales son:

- data/: demo y datos XML
- models/: modelos de definición
- controllers/: contiene los controladores de rutas (HTTP).
- views/: contiene las vistas y plantillas
- static/: contiene los elementos web, separados en css / js /, img /, lib /, ...

Nomenclatura de archivos: permite enunciar las clases del modelo y vistas, separando la lógica de negocio por conjuntos de modelos. Facilitando la comprensión de los elementos correspondientes en cada modelo permitiendo crear para cada conjunto denominado <main_model> los siguientes archivos:

- models/<main_model>.py
- views/<main_model>_templates.xml
- views/<main_model>_views.xml

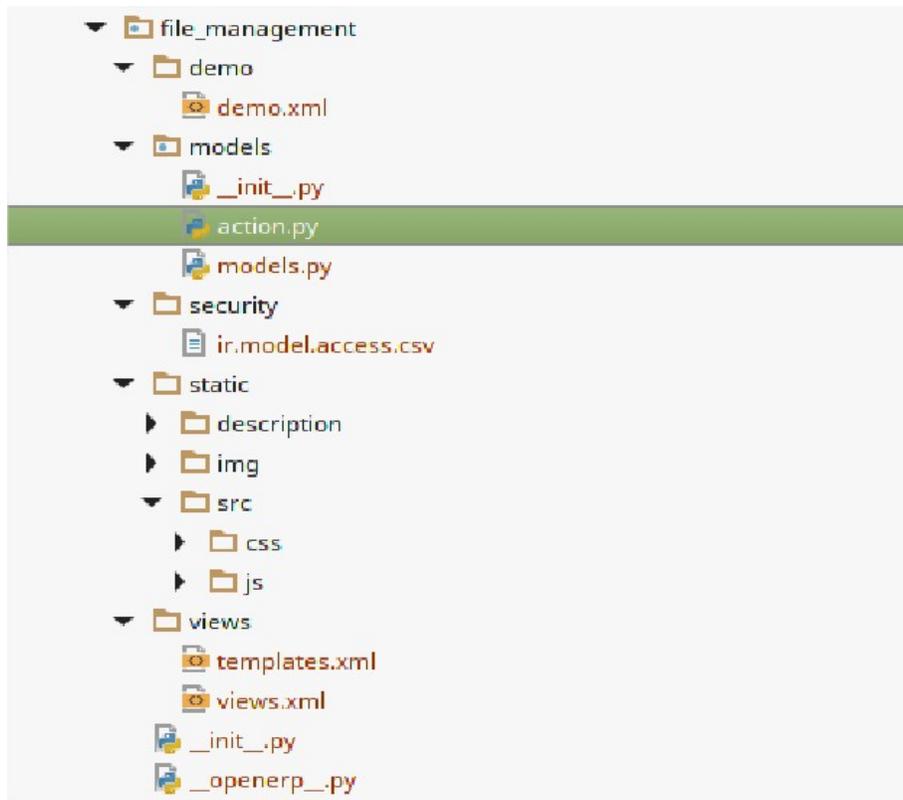


Figura 9: Estándar de codificación

Archivos XML

Formato: se utilizó para la declaración de registros en XML en los cuales el identificador único (id) para un elemento se escribe antes del modelo y en la declaración de campo el nombre de atributo es el primero.

```
<record model="ir.ui.view" id="move_form">
  <field name="name">move form</field>
  <field name="model">move</field>
  <field name="arch" type="xml">
    <sheet>
```

Figura 10: Estándar de codificación

Nombre xml_id: define los nombres de los elementos de seguridad, las vistas y las funcionalidades los cuales se utilizaron de la siguiente manera:

- En la vista la estructura es: <model_name>_view_<view_type>, donde view_type es Kanban, form, tree, search.
- En la funcionalidad: se representa así <model_name>_action.

Opciones PEP8: resalta las buenas prácticas de programación, donde cada línea cumple que no sea demasiado larga, se respeta la sangría, hay cierre de corchete y los bloques de comentario empiezan con '#'.

Import: se utilizó para importar las librerías tanto internas como externas las cuales se organizan alfabéticamente de arriba hacia abajo y de izquierda a derecha.

```
from openerp import models, fields, api, exceptions
from datetime import datetime
from openerp.exceptions import except_orm, Warning, RedirectWarning
```

Figura 11: Estándar de codificación

3.2 Diagrama de componentes

Un diagrama de componentes al igual que un sistema de [software](#) es dividido en [componentes](#) y muestra las [dependencias](#) entre ellos. Los componentes físicos incluyen [archivos](#), cabeceras, [bibliotecas compartidas](#), [módulos](#), [ejecutables](#), o [paquetes](#). 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 (Castro, 2008).

En el diagrama de componente de la presente investigación se cuenta con tres paquetes los cuales son el paquete de modelo, de vistas y de interacción con la plataforma de desarrollo Odoo. El paquete de modelo está compuesto por todos los modelos de la propuesta de solución y el paquete de vista por las vistas de cada modelo. Este se encarga de guardar los datos en el sistema gestor de base de datos y las vistas son realizadas en archivos xml las cuales también son almacenadas en la base de datos. Odoo es el responsable de adquirir la información guardada por el en la base de datos y a su vez de renderizar los archivos xml para convertirlos en vistas.

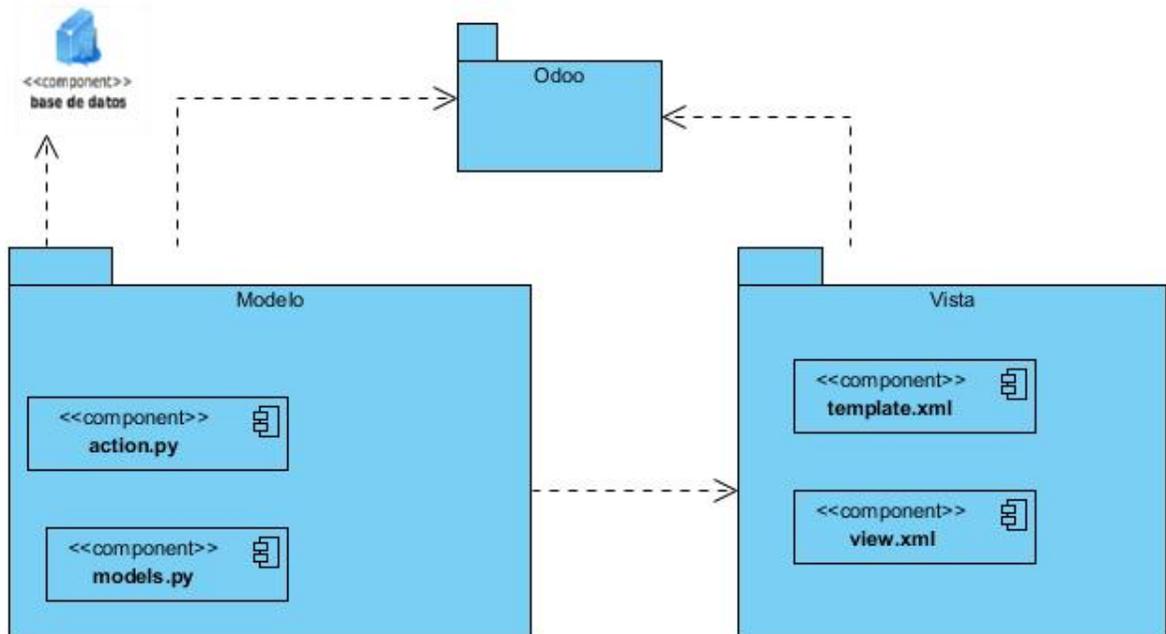


Figura 12: Diagrama de componentes del módulo Gestor de Archivos

3.3 Diagrama de despliegue

Modela la arquitectura en tiempo de ejecución y muestra la disposición física de los nodos que componen el módulo. Presenta la configuración de los elementos de *hardware* (nodos) y muestra como los elementos del *software* se encuentran conectados por enlaces de comunicación. A continuación, se muestra el referente al módulo Gestor de Archivos y se describen sus componentes.



Figura 13: Diagrama de despliegue

PC_Cliente: dispositivo que utiliza el cliente para interactuar con el módulo.

Servidor media: destinado para el almacenamiento y manejo de los recursos del sistema.

Servidor de BD: reservado para el trabajo con los datos de la BD.

Servidor de aplicaciones web: para desplegar la aplicación.

3.4 Pruebas de software

Las pruebas son actividades en las cuales un sistema o componente es ejecutado bajo condiciones o requerimientos especificados. Son realizadas con el objetivo de encontrar errores y no conformidades, verificando de esta manera la calidad del producto (SANCHEZ RODRIGUEZ, 2015). En su realización se comprueba la cobertura de los requisitos, dado que su incumplimiento puede comprometer la aceptación del módulo por el cliente responsable de realizar las pruebas de implantación del módulo.

3.5 Niveles de Prueba

Las pruebas son aplicadas para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo (PRESSMAN, 2005). Entre los niveles de prueba que propone la metodología de desarrollo AUP-UCI se encuentran pruebas internas, pruebas de liberación y pruebas de aceptación. Las pruebas realizadas a la solución propuesta fueron pruebas funcionales y pruebas de aceptación.

Pruebas de aceptación: son responsabilidad del cliente, se llevan a cabo antes de que el programa se ponga en funcionamiento real y tienen que satisfacer las expectativas del cliente. (Sánchez Peño, 2015) A menudo se realizan en dos etapas:

- Alfa: se lleva a cabo por el cliente, en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario y registrando los errores y problemas de uso. Las pruebas alfa se llevan a cabo en un entorno controlado. (Valdez Huaraca, y otros, 2013)
- Beta: se realizan con posterioridad a las pruebas alfa, y se desarrollan en el entorno del cliente. En este caso, el cliente se queda a solas con el producto y trata de encontrarle fallos de los que informa al desarrollador. (Valdez Huaraca, y otros, 2013)

Pruebas del Sistema: se prueba el módulo para comprobar que se cumplen los requisitos funcionales. Cualquier pieza de *software* completo, desarrollado o adquirido, puede verse como un sistema que debe probarse, ya sea para decidir acerca de su aceptación, para analizar defectos

globales o para estudiar aspectos específicos de su comportamiento, tales como seguridad o rendimiento. Este tipo de pruebas estudia el producto completo.

3.6 Métodos de pruebas

Los métodos de prueba de *software* incluyen la identificación, medición y evaluación de una o más cualidades, características o propiedades del *software* en cuestión. Definen un procedimiento que permite descubrir diferentes tipos de errores con menor tiempo y esfuerzo (Rodríguez Trello, 2012). Se reconoce la existencia de dos métodos básicos para diseñar casos de prueba: de caja negra (o funcional) y de caja blanca (o estructural), a continuación se describen ambos métodos.

Pruebas de caja negra: realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, solo la funcionalidad que debe realizar (PRESSMAN, 2005).

Pruebas de caja blanca: al contrario de las pruebas de caja negra, consiste en verificar la estructura interna del *software*. Verifican la correcta implementación de las unidades internas, las estructuras y sus relaciones.

3.7 Casos de prueba

El diseño de casos de prueba se realiza con el objetivo crear un conjunto de casos de prueba que sean efectivos para descubrir defectos en los programas y muestren que el módulo satisface sus requerimientos. Como parte de este proceso se definen las entradas y las salidas esperadas para probar el módulo (Sommerville, 2005). A continuación se muestra el caso de prueba correspondiente al RF Crear carpeta, puede consultar otros en el Anexo 6.

Descripción general:

Permite al Usuario crear una nueva carpeta.

Condiciones de ejecución:

- El usuario debe estar autenticado en el sistema.
- El usuario debe tener permisos para crear carpetas.

SC 1 Crear carpeta

ID del escenario	Escenario	Nombre	Respuesta del módulo	Flujo central
EC1.1 Crear Carpeta	Se selecciona la opción crear para crear una nueva carpeta		Brinda la posibilidad de introducir los siguientes datos: -(*) Nombre Además permite: -Aceptar -Cancelar	Módulo Gestor de Archivos /Crear Carpeta
EC 1.2 Guardar los datos	Se Introducen los datos de la carpeta y selecciona la opción guardar.	V	El módulo valida y guarda los datos de la nueva carpeta.	Módulo Gestor de Archivos /Crear Carpeta/ Guardar
EC 1.3 Información incompleta	Existen campos obligatorios que no se completaron		Señala el campo incompleto (en este caso el campo Nombre) y muestra el mensaje: Rellene este campo.	Módulo Gestor de Archivos /Crear Carpeta/ Guardar
EC 1.4 Información incorrecta	Existen campos incorrectos.		El módulo señala el o los campos que no hayan sido introducidos correctamente.	Módulo Gestor de Archivos /Crear Carpeta/ Guardar
EC 1.5 Descartar cuando se hayan introducido datos opción	Se selecciona la opción Cancelar	V	El módulo mantiene los cambios realizados y regresa al listado de carpetas.	Módulo Gestor de Archivos /Crear Carpeta/ Descartar/Cancelar

EC 1.6 Descartar cuando se hayan introducido datos opción Aceptar	Se selecciona la opción Aceptar	V	El módulo elimina los datos introducidos y regresa al listado de carpetas.	Módulo Gestor de Archivos /Crear Carpeta/ Descartar/Aceptar
--	---------------------------------	---	--	---

3.8 Resultados de las pruebas

Se realizaron pruebas de caja negra en tres iteraciones, obteniéndose en la primera iteración diez No Conformidades (NC), en la segunda iteración se detectaron cuatro NC y en la tercera iteración el módulo estaba libre de NC, cumpliendo con los requisitos funcionales especificados. Estas fueron agrupadas en la siguiente tabla siendo evaluadas en un rango comprendido entre: Alta, Media, Baja y No procede. A continuación, se presentan los resultados arrojado por estas pruebas en la 1ra iteración. Para ver más información sobre las demás iteraciones ver Anexos 8.

Tabla 7: No Conformidades del Módulo Gestor de archivos 1ra iteración

No. CP	Caso de prueba	No conformidades				
		Alta	Media	Baja	No procede	Total
1	CP Crear carpetas	-	-	2	-	2
2	CP Editar carpetas	-	-	1	-	1
3	CP Mover carpetas	-	-	1	-	1
4	CP Listar carpetas	-	-	-	-	-
5	CP Eliminar carpetas	-	-	1	-	1
6	CP Importar archivos	-	-	-	-	-
7	CP Exportar archivos	-	-	-	-	-
8	CP Listar archivos	-	-	1	-	1
9	CP Duplicar carpetas	-	-	-	-	-
10	CP Insertar color a las carpetas	-	-	1	-	1
11	CP Eliminar archivos	-	-	-	-	-
12	CP Crear subcarpetas	-	-	2	-	2
13	CP Adicionar subcarpeta	-	-	1	-	1
14	CP Buscar carpetas y archivos	-	-	-	-	-
15	CP Importar Carpetas	-	-	-	-	-

Capítulo 3

16	CP Exportar Carpetas	-	-	-	-	-
	Total	-	-	10	-	10

Para verificar las NC detectadas se presenta una tabla con los siguientes datos: el número de caso de prueba (No. CP), clasificación (clasificada en validación, funcionalidad, correspondencia con otro artefacto, errores de idioma, ortografía, redacción, recomendación, error de caso de prueba) y estado con respecto a la solución. Para ver más información sobre las demás iteraciones ver Anexos 8.

Tabla 8: Descripción de No Conformidades 1ra iteración

No. NC	No. CP	Clasificación	Complejidad	Estado
1	1	Correspondencia con otro artefacto	Baja	RA
2	1	Error de caso de prueba	Baja	RA
3	2	Redacción	Baja	RA
4	3	Correspondencia con otro artefacto	Baja	RA
5	5	Error de caso de prueba	Baja	RA
6	8	Correspondencia con otro artefacto	Baja	RA
7	10	Correspondencia con otro artefacto	Baja	RA
8	12	Error de caso de prueba	Baja	RA
9	12	Redacción	Baja	RA
10	13	Correspondencia con otro artefacto	Baja	RA

Resultados de las Pruebas de aceptación de tipo alfa

La prueba de aceptación de tipo alfa fue realizada al módulo desarrollado como propuesta de solución. Para ello se efectuaron dos iteraciones. En la primera de ellas se detectaron 2 no conformidades; de estas 1 se clasificó de complejidad alta y 1 de complejidad baja. En la 2da iteración el módulo se encontraba libre de no conformidades.

3.9 Conclusiones del capítulo

- La creación de un diagrama de componentes permitió realizar un desglose de los diferentes elementos de software que componen el módulo para un mejor entendimiento de cómo está conformado.
- Las pruebas realizadas permitieron detectar a tiempo un grupo de no conformidades que fueron resueltas, obteniéndose finalmente un producto de software con las funcionalidades requeridas.

Conclusiones

Conclusiones

A continuación, se exponen las conclusiones generales a las que fueron arribadas luego del desarrollo de la propuesta de solución:

- El estudio del marco teórico evidenció la necesidad de desarrollar un módulo Gestor de Archivos para SIGAX utilizando la metodología, tecnologías y herramientas seleccionadas.
- Se desarrolló un módulo para la gestión de archivos que permitió contribuir a la centralización y control de gestión de la información.
- Se realizaron pruebas de software al módulo Gestor de Archivos que permitieron identificar y corregir no conformidades mejorando así la calidad del producto desarrollado.

Recomendaciones

Recomendaciones

Con el desarrollo de futuras investigaciones se recomienda perfeccionar e incorporar nuevas funcionalidades para lograr mejoras superiores en la gestión de la información dentro del objeto de estudio de la presente investigación, entre las cuales se pueden considerar las siguientes:

- Agregar al módulo una funcionalidad para el control de versiones.
- Agregar al módulo una funcionalidad que permitan el trabajo con archivos comprimidos.

Referencias Bibliográficas

Referencias Bibliográficas

Academia, Real. DLE: transitario, ria - Diccionario de la lengua española - Edición del Tricentenario. [En línea] [Citado el: 20 de Noviembre de 2015.] <http://dle.rae.es/?id=aKdgVUI>.

ADINS. 2011. OpenObject. [En línea] 2011. [Citado el: 23 de Febrero de 2016.] <http://adins.be/en/openobject>.

—. Technical foundation. [En línea] [Citado el: 24 de Febrero de 2016.] <http://adins.be/en/openobject/36-fondation-technique>.

Aguavil, Rocío Maribel Villota. 2015. *Aplicación informática ERP, para mejorar la gestión de ventas de la empresa Master Soluciones*. 2015.

ALEGSA, Leandro. 2014. Definición de Python (lenguaje de programación). [En línea] 2014. [Citado el: 24 de Noviembre de 2015.] <http://www.alegsa.com.ar/Dic/python.php>.

Bargaw. 2014. Odoo - Bargaw. [En línea] 2014. [Citado el: 4 de Marzo de 2016.] <http://www.bargaw.com/erpcrmodoo/>.

BECK, Kent. 1999. *Embracing Change with Extreme Programming*. 1999.

BEDOYA, Andrés. 2011. Patrones GOF. [En línea] 2011. [Citado el: 15 de Marzo de 2016.] <http://geektheplanet.net/5462/patrones-gof.xhtml>.

CALVO LOPEZ, Eva. 2010. [En línea] 2010. [Citado el: 31 de Enero de 2016.] <http://upcommons.upc.edu/bitstream/handle/2099.1/8947/Memoria.pdf?sequence=1&isAllowed=y>.

Cargax. 2012. Transitarios y Transporte de Mercancías en Grupaje y Contenedor. [En línea] 2012. [Citado el: 10 de Noviembre de 2015.] <http://www.cargax.com/el-origen-de-los-transitarios>.

CASTRO, Eliseo. 2008. UML - Lenguaje de Modelamiento Unificado. [En línea] 2008. [Citado el: 26 de Noviembre de 2015.] <http://es.slideshare.net/ecastrojimenez/uml-lenguaje-de-modelamiento-unificado-presentation>.

Castro, Eliseo. 2008. UML - Lenguaje de Modelamiento Unificado . [En línea] 2008. <http://es.slideshare.net/ecastrojimenez/uml-lenguaje-de-modelamiento-unificado-presentation>.

Chiavenato, Idalberto. 2006. *Introducción a la Teoría General de la Administración*. 2006.

ClubEnsayos. 2011. Concepto E Importancia De La Gestion Empresarial - Composiciones de Colegio. [En línea] 2011. [Citado el: 10 de Diciembre de 2015.] <https://www.clubensayos.com/Negocios/Concepto-E-Importancia-De-La-Gestion-Empresarial/69236.html>.

Referencias Bibliográficas

CORTES, Carlos, MOLINA, Vanessa, PATERNINA, Liseth and VARGAS, Oscar. 2013.

MetodologiasagilesProgramacionXtremaslideshare. [En línea] 2013. [Citado el: 28 de Noviembre de 2015.] <http://www.slideshare.net/LisPater1/metodologias-agiles-xp>.

Datazucar. Datazucar » Datazucar » Versat-Sarasola. [En línea] [Citado el: 5 de Marzo de 2016.] <http://www.datazucar.cu/?p=65>.

Definicion. 2010. Definición de contenedor - Qué es, Significado y Concepto. [En línea] 2010. [Citado el: 11 de Noviembre de 2015.] <http://definicion.de/contenedor/>.

DefinicionABC. 2010. Definición de Arrendamiento » Concepto en Definición ABC. [En línea] 2010. [Citado el: 13 de Noviembre de 2015.] <http://www.definicionabc.com/derecho/arrendamiento.php>.

Departamento de Lenguajes y Sistemas Informaticos. Universidad de Sevilla. 2006. Características y Fases de la Prueba. [En línea] 2006. [Citado el: 20 de Abril de 2016.] <http://www.lsi.us.es/docencia/get.php?id=361>.

DESOFT. Los ERP y sistemas contables en Cuba | MINCOM. [En línea] [Citado el: 15 de Marzo de 2016.] <http://www.mincom.gob.cu/?q=node/550>.

Domatrix. Odoos – OpenERP – ERP, CRM, MRP, SGA 100% Libre – | Sin Licencias ¿Por qué elegir OpenErp? [En línea] [Citado el: 25 de Enero de 2016.] <http://openerpspain.com/openerp/porque-elegir-openerp/>.

Evaluando ERP. eTES primer ERP cubano registrado en EvaluandoERP.com. [En línea] [Citado el: 29 de Febrero de 2016.]

eXcriba, Gestor de Documentos Administrativos. https://www.ecured.cu/Gestor_de_Documentos_Administrativos_eXcriba. [En línea]

EXXIS. 2010. ¿Qué es un ERP? [En línea] 2010. [Citado el: 29 de Enero de 2016.] http://www.exxis.cl/index.php?option=com_content&view=article&id=3:ique-es-un-erp&catid=1:preguntas-frecuentes&Itemid=3.

GÓNZALEZ BLANCO, Rubén y PÉREZ TOBALINA, Sergio. 2010. LESE-2 Introducción a Rational Rose. [En línea] 2010. [Citado el: 28 de Noviembre de 2015.] <http://es.scribd.com/doc/38823255/LESE-2-Introduccion-a-Rational-Rose>.

Hermenegildo, Romero. 2012. <http://es.slideshare.net/MeneRomero/metodologias-de-desarrollo>. <http://es.slideshare.net>. [En línea] 2012.

IBACH, MICHEL. 1999. *Management information system*. 1999.

IBERTRANSIT. 2013. Transitario | IBERTRANSIT. [En línea] 2013. [Citado el: 10 de Noviembre de 2015.] <http://www.ibertransit.com/inicio-transitario>.

Referencias Bibliográficas

Implementacion SIG. 2015. Que es un Sistema de Gestión. [En línea] 2015. [Citado el: 12 de Diciembre de 2015.] <http://www.implementacionsig.com/index.php/23-noticiac/28-que-es-un-sistema-de-gestion>.

Innova. 2013. ADUANAS - Contenedores. Definición. [En línea] 2013. [Citado el: 13 de Noviembre de 2015.] <http://www.aduanas.gub.uy/innovaportal/v/2542/8/innova.front/contenedores-definicion.html>.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software*. 4.

Jaramillo. 2009. *Unidad 1 Elementos Conceptuales*. [En línea]. 2009.

JetBrains. PyCharm. [En línea] [Citado el: 6 de Febrero de 2016.] <https://www.jetbrains.com/pycharm/>.

LARMAN, Craig. 1999. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. s.l. : Felix Varela, 1999.

LARMAN, Craig y HALL, Prentice. 2003. El Modelo de Diseño. [En línea] 2003. [Citado el: 28 de Abril de 2016.] is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDiseno.pdf.

—. 2003. Modelo de Dominio. [En línea] 2003. [Citado el: 1 de Mayo de 2016.] <http://is.ls.fi.upm.es/docencia/is2/documentacion/ModeloDominio.pdf>.

LÓPEZ, JoséMaría. 2012. Dia. [En línea] 2012. [Citado el: 23 de Noviembre de 2015.] <http://dia.softonic.com/>.

López-Hermoso, J. 2004. *La gestión del conocimiento en las organizaciones*. 2004.

márquez, javier. 2014. <http://www.transcargo.transnet.cu/>. [En línea] 2014. 2.

MARTINEZ, Rafael. 2010. Sobre PostgreSQL | www.postgresql.org.es. [En línea] 2010. [Citado el: 8 de Diciembre de 2015.] http://www.postgresql.org.es/sobre_postgresql.

Microsoft. Microsoft Solutions Framework (MSF). [En línea] [Citado el: 26 de Noviembre de 2015.] <https://msdn.microsoft.com/es-es/library/jj161047%28v=vs.120%29.aspx>.

NUBISTALIA. 2011. Odoo on demadn - SaaS - OpenERP - ¿Para qué sirve un ERP? [En línea] 2011. [Citado el: 1 de Febrero de 2016.] <http://www.openerpondemand.com/qu%C3%A9-es-odoo/para-qu%C3%A9-sirve-un-erp/>.

—. 2011. OpenERP OnDemand. ¡El ERP lider de Europa desde 99€/mes! - Odoo - Módulos. [En línea] 2011. [Citado el: 6 de Febrero de 2016.] <http://www.nubistalia.es/soluciones-y-servicios/odoo-openerp/odoo-m%C3%B3dulos/>.

Referencias Bibliográficas

O'Reilly Media, Inc. 2010. XML.com. [En línea] 2010. [Citado el: 12 de Febrero de 2016.] <http://www.xml.com/pub/a/98/10/guide0.html?page=2>.

Odoo – OpenERP – ERP, CRM, MRP, SGA 100% Libre – | Sin Licencias ¿Que es OpenERP? ERP 100% Libre. 2012. [En línea] 2012. [Citado el: 20 de Enero de 2016.] <http://openerpspain.com/openerp/que-es-openerp/>.

Odoo. Odoo Guidelines — odoo 8.0 documentation. [En línea] [Citado el: 20 de Abril de 2016.] <https://www.odoo.com/documentation/8.0/reference/guidelines.html>.

ODOO. Open Source ERP and CRM | Odoo. [En línea] [Citado el: 8 de Febrero de 2016.] <https://www.odoo.com/>.

Odoo S.A. Product Container | Odoo Apps. [En línea] [Citado el: 2 de Marzo de 2016.] https://www.odoo.com/apps/modules/7.0/product_container/.

—. Product Container Tracking | Odoo Apps. [En línea] [Citado el: 3 de Marzo de 2016.] https://www.odoo.com/apps/modules/7.0/product_container_tracking/.

Oposiciones TIC. 2011. Arquitectura cliente servidor - Oposiciones TIC. [En línea] 2011. [Citado el: 8 de Marzo de 2016.] <http://oposicionestic.blogspot.com/2011/06/arquitectura-cliente-servidor.html>.

ORÉ, Alexander. 2009. Pruebas Funcionales - Software Testing and QA. [En línea] 2009. [Citado el: 20 de Abril de 2016.] http://www.calidadyssoftware.com/testing/pruebas_funcionales.php.

Patricio Letelier, M^a Carmen Penadés. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [En línea] 15 de 01 de 2006. [En línea] <http://www.cyta.com.ar/ta0502/v5>.

Patrón Modelo-Vista-Controlador. **Fernández Romero, Yenisleidy y Díaz González, Ynette.** 2012. 1, 2012, Vol. XI. ISSN 1729-3804.

Pérez, Jesús. 2014. <http://www.comunidadbaratz.com/blog/bkmplus-una-potente-solucion-para-la-gestion-del-conocimiento-organizacional/>. [En línea] 2014.

PRESSMAN, Roger S. 2005. *Ingeniería de software, un enfoque practico.* s.l. : McGraw-Hill Companies, 2005. ISBN 8448132149.

Prodware. 2016. Definición de ERP. [En línea] 2016. [Citado el: 30 de Enero de 2016.] <https://www.elegirerp.com/definicion-erp>.

Quiroga, Lic. Lourdes Aja. 2011. *Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones.* 2011.

Referencias Bibliográficas

RedAccenir, S.L. 2009. IMB Container - Descargar Gratis. [En línea] 2009. [Citado el: 30 de Enero de 2016.] <http://www.portalprogramas.com/imb-container/>.

RUMBAUG, James, JACOBSON, Ivar y BOOCH, Grady. 2007. El Lenguaje Unificado de Modelado. Manual de referencia. [En línea] 2007. [Citado el: 22 de Abril de 2016.] <http://librosgratisparauniversitario.blogspot.com/2014/03/el-lenguaje-unificado-de-modelado.html>. ISBN 978-84-7829-087-1.

SANCHEZ RODRIGUEZ, Tamara. 2015. *Metodología de desarrollo para la UCI*. La Habana : s.n., 2015.

SÁNCHEZ, TAMARA RODRÍGUEZ. 2014. *Metodología de desarrollo para la Actividad productiva de la UCI*. 2014.

Significados. 2013. <http://www.significados.com/gestion/>. [En línea] 2013.

—. Significado de Metodología - Qué es, Concepto y Definición. [En línea] [Citado el: 27 de Noviembre de 2015.] <http://www.significados.com/metodologia/>.

SoftDoit. 2009. SAP ERP características y funcionalidades. Entra y compara gratis » SoftDOit. [En línea] 2009. [Citado el: 10 de Febrero de 2016.] <https://www.softwaredoit.es/sap-erp-caracteristicas-y-funcionalidades/sap-erp-caracteristicas-y-funcionalidades.html>.

UDIMA. 2009. XML ¿Qué es? | Manual de XML. [En línea] 2009. [Citado el: 15 de Febrero de 2016.] <http://www.mundolinux.info/que-es-xml.htm>.

Valdés, María de las Mercedes Fernández. 2008. *Análisis conceptual de las principales interacciones entre la gestión de información, la gestión documental y la gestión del conocimiento*. 2008.

Verde, Luis Tortosa. 2015. *Ingeniería en Informática de Gestión*. 2015.

Vidal, Zenia Camps. <http://www.uci.cu/el-excriba-una-herramienta-necesaria>. [En línea]

Villagrana, A. L. 2014. RUP (Rational Unified Process) Proceso Unificado Racional. [En línea] 2014. [Citado el: 6 de Diciembre de 2015.] http://proceso-unificado-racional.blogspot.com/2014/02/proceso-unificado-racional_24.html.

Visbal, Sara Artilles. 2013. *El Modelo de Gestión de Información y Conocimiento: resultados de su aplicación en una empresa*. 2013.

Visual Paradigm International. 2014. Visual Paradigm Searchfor Visual Paradigm. [En línea] 2014. [Citado el: 5 de Diciembre de 2015.] <http://www.visual-paradigm.com/search/search.jsp?query=Visual+Paradigm&x=22&y=9>.

Referencias Bibliográficas

Anexo 1: Preguntas de la entrevista realizada al Ing. Osvaldo Ernesto Stable Vilches

Osvaldo Ernesto Stable Vilches es Ingeniero en Ciencias Informáticas, graduado en la Universidad de Ciencias Informáticas (UCI). Se desempeñó como especialista en el centro FORTES perteneciente a la facultad 4 en la UCI. A continuación se relacionan las preguntas realizadas a este especialista.

1. ¿Cuándo comienza a implementarse en TRANSCARGO el SIGAX? ¿Cuál es su objetivo?
2. ¿Cómo se realiza actualmente el trabajo con la documentación en TRANSCARGO?
3. ¿Qué ventajas brinda el SIGAX para la empresa?
4. ¿Cuáles son las principales características y necesidades del SIGAX?
5. ¿Existe una metodología y herramientas definidas por el proyecto para la gestión de los proyectos?

Anexo 2: Especificación de requisitos funcionales

Tabla 1: Especificación del RF Crear carpeta

No.	Nombre	Descripción	PRC	C	RC
RF 1	Crear carpetas	<p>El módulo debe permitir crear una carpeta, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> • (*) Nom. de la carpeta: identificador único de una carpeta. Campo de texto, admite el siguiente formato, letras, números o símbolos, con excepción de " # % * : < > ? / \ . 	Alta	Alta	FORTES_ Transcarg o_Especifi cación_de _requisitos

		<p>Ejemplo: barca92.</p> <ul style="list-style-type: none"> Color: representa el color de la carpeta. Campo de selección que incluye varios colores 			_de_softw are
--	--	--	--	--	------------------

Tabla 2: Especificación del RF Editar carpeta

No.	Nombre	Descripción	PRC	C	RC
RF 2	Editar carpetas	<p>El módulo debe permitir editar una carpeta, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> (*) Nom. de la carpeta: identificador único de una carpeta. Campo de texto, admite el siguiente formato, letras, números o símbolos, con excepción de " # % * : < > ? / \ . <p>Ejemplo: barca92.</p>	Alta	Alta	FORTES _Transca rgo_Esp ecificació n_de_re quisitos_ de_softw are

Tabla 3: Especificación del RF Mover carpetas

No.	Nombre	Descripción	PRC	C	RC
RF 3	Mover carpetas	<p>El módulo debe permitir mover una carpeta, solicitando los siguientes datos:</p> <ul style="list-style-type: none"> (*) Nom. de la carpeta a mover: identificador único de una carpeta. (*) Nom. de la carpeta de destino: identificador único de una carpeta. 	Baja	Baja	FORTES_Transcarg o_Especifi cación_de _requisitos _de_softw are

Tabla 4: Especificación del RF Listar carpetas

No.	Nombre	Descripción	PRC	C	RC
RF 4	Listar carpetas	<p>El módulo debe permitir listar las carpetas existentes en forma ascendente o descendente según el nombre. En el listado se deben mostrar los siguientes datos:</p> <ul style="list-style-type: none"> Nombre de la carpetas Cantidad de subcarpetas relacionadas a cada carpeta 	Media	Media	FORTES _Transca rgo_Espe cificación _de_requ sitos_de_

					software
--	--	--	--	--	----------

Tabla 5: Especificación del RF Eliminar carpetas

No.	Nombre	Descripción	PRC	C	RC
RF 5	Eliminar carpetas	El módulo debe permitir eliminar una o varias (según sea el caso) carpetas.	Media	Media	FORTE S_Trans cargo_E specifica ción_de _requisit os_de_s oftware

Tabla 6: Especificación del RF Importar archivos

No.	Nombre	Descripción	PRC	C	RC
RF 6	Importar archivos	El módulo debe permitir importar uno o varios (según sea el caso) archivos.	Media	Media	FORTES _Transca rgo_Esp ecificació

					n_de_re quisitos_ de_softw are
--	--	--	--	--	---

Tabla 7: Especificación del RF Exportar archivos

No.	Nombre	Descripción	PRC	C	RC
RF 7	Exportar archivos	El módulo debe permitir exportar uno o varios (según sea el caso) archivos.	Media	Media	FORTES _Transca rgo_Esp esificació n_de_re quisitos_ de_softw are

Tabla 8: Especificación del RF Listar archivos y Duplicar carpetas

No.	Nombre	Descripción	PRC	C	RC
RF	Listar	El módulo debe permitir listar los archivos existentes de forma ascendente o descendente. En el listado se deben	Alta	Alta	FORTES

8	archivos	<p>mostrar los siguientes datos:</p> <ul style="list-style-type: none"> - Nombre del archivo. - Extensión del archivo. 			_Transcargos_Especificación_de_requisitos_de_software
No.	Nombre	Descripción	PRC	C	RC
RF 9	Duplicar carpetas	<p>El módulo debe permitir duplicar una carpeta , solicitando los siguientes datos:</p> <ul style="list-style-type: none"> - Nombre de la carpeta. 	Media	Media	FORTES _Transcargos_Especificación_de_requisitos_de_software

Tabla 10: Especificación del RF Insertar color a carpetas

No.	Nombre	Descripción	PRC	C	RC
-----	--------	-------------	-----	---	----

RF 10	Insertar color a la carpetas	El módulo debe permitir insertar color a las carpetas, solicitando los siguientes datos: -Nombre de la carpeta. - nombre de color a insertar.	Media	Media	FORTES _Transc rgo_Esp ecificació n_de_re quisitos_ de_softw are
--------------	------------------------------	---	-------	-------	---

Tabla 11: Especificación del RF Eliminar archivos

No.	Nombre	Descripción	PRC	C	RC
RF 11	Eliminar archivos	El módulo debe permitir eliminar uno o varios (según sea el caso) archivos.	Media	Media	FORTE S_Tran scargo_ Especifi cación_ de_req usitos_ de_soft ware

Tabla 12: Especificación del RF Crear subcarpetas

No.	Nombre	Descripción	PRC	C	RC
RF 12	Crear subcarpetas	El módulo debe permitir crear carpetas dentro de carpetas (subcarpetas) una o varias (según sea el caso) carpetas.	Media	Media	FORTES _Transca rgo_Esp esificació n_de_req usitos_d e_softwa re

Tabla 13: Especificación del RF Mover carpetas

No.	Nombre	Descripción	PRC	C	RC
RF 13	Adicionar subcarpetas	El módulo debe permitir adicionar subcarpetas ya existentes dentro de carpetas.	Media	Media	FORTE S_Tran scargo_ Especifi cación_ de_requ

					sitos_d e_softw are
--	--	--	--	--	---------------------------

Tabla 14: Especificación del RF Buscar carpetas y archivos

No.	Nombre	Descripción	PRC	C	RC
RF 14	Buscar carpetas o archivos	El módulo debe permitir buscar carpetas o archivos , solicitando los siguientes datos : -Nombre de la carpeta o archivo	Media	Media	FORTES _Transca rgo_Esp ecificació n_de_req usitos_d e_softwa re

Tabla 15: Especificación del RF Importar carpetas

No.	Nombre	Descripción	PRC	C	RC
RF 15	Importar carpetas	El módulo debe permitir importar una o varias (según sea el caso) carpetas que se encuentren en formato csv.	Media	Media	FORTES _Transca

					rgo_Esp pecificació n_de_req usitos_d e_softwa re
--	--	--	--	--	--

Tabla 16: Especificación del RF Exportar carpetas

No.	Nombre	Descripción	PRC	C	RC
RF 16	Exportar carpetas	El módulo debe permitir exportar una o varias (según sea el caso) carpetas en formato csv.	Media	Media	FORTES _Transca rgo_Esp pecificació n_de_re quisitos_ de_softw are

Anexo 3: Descripción de requisitos por procesos

Tabla 17: Descripción de RF: Crear carpeta

Precondiciones	<p>El usuario debe estar autenticado en el sistema.</p> <p>El usuario debe poseer los permisos para crear carpetas.</p>
Flujo de eventos	
Flujo básico Crear carpeta	
1.	El usuario selecciona la opción crear carpeta.
2.	<p>El módulo debe permitir introducir y/o seleccionar los siguientes datos para crear una carpeta:</p> <p>(*) Nombre: representa el nombre de la carpeta. Campo de texto, admite el siguiente formato, letras, números o símbolos, con excepción de " # % * : < > ? / \ .</p> <p>Color: representa el color de la carpeta. Campo de selección que incluye varios colores</p> <p>El módulo brinda las siguientes opciones:</p> <p style="padding-left: 40px;"><i>Guardar</i></p> <p style="padding-left: 40px;"><i>Descartar</i></p>
3.	Se introducen y/o seleccionan los datos para crear una carpeta.
4.	El usuario selecciona la opción Guardar.
5.	<i>El módulo valida los datos.</i>
6.	<i>El módulo guarda los datos de la nueva carpeta.</i>
7.	Concluye así el requisito.
Pos-condiciones	
	Se incluyó una nueva carpeta satisfactoriamente.
Flujos alternativos	

Flujo alternativo 6.a Información incompleta	
	El módulo señala el o los campos obligatorios que no hayan sido introducidos y/o seleccionados y muestra el mensaje de información: <i>Los siguientes campos son obligatorios;</i> indicando los campos en cuestión.
	El usuario introduce y/o selecciona los datos.
	Volver al paso 2 del flujo básico.
Pos-condiciones	
1.	NA
Flujos alternativo 6.b Información incorrecta	
	El módulo señala el o los campos incorrectos y muestra el mensaje de información: <i>Los siguientes campos son incorrectos;</i> indicando los campos en cuestión.
	El usuario introduce y/o selecciona correctamente los datos.
	Volver al paso 2 del flujo básico.
Pos-condiciones	
1.	N/A
Flujo alternativo 4.a Descartar cuando se hayan introducido datos	
1.	El usuario selecciona la opción <i>Descartar</i> .
2.	El módulo muestra un mensaje de información: <i>Sus cambios se perderán y solicita confirmación ¿Está seguro que quiere abandonar la página?</i>
3.	El módulo brinda las siguientes opciones: <i>Aceptar</i> <i>Cancelar</i>

4.	El usuario selecciona la opción <i>Aceptar</i> .	
5.	El módulo elimina los datos introducidos y regresa al listado de carpetas.	
6.	Concluye así el requisito.	
Pos-condiciones		
1.	No se crea la carpeta.	
1	N/A	
Flujo alternativo * <i>Descartar</i>		
1.	El módulo regresa al listado de carpetas.	
2.	Concluye así el requisito.	
Pos-condiciones		
1.	N/A	
Validaciones		
	N/A	
Conceptos	NA	NA
Requisitos especiales		
Asuntos pendientes	NA	

Tabla 18: Descripción de RF: Editar carpeta

Precondiciones	El usuario debe estar autenticado en el sistema.
-----------------------	--

	<p>Debe haberse incluido al menos una carpeta.</p> <p>El usuario debe poseer los permisos para editar carpetas.</p>
Flujo de eventos	
Flujo básico editar carpeta	
1.	El usuario selecciona la opción editar.
2.	<p>El módulo debe permitir modificar los datos de una carpeta:</p> <p>(*) Nombre: representa el nombre de la carpeta. Campo de texto, admite el siguiente formato, letras, números o símbolos, con excepción de " # % * : < > ? / \ .</p> <p>Color: representa el color de la carpeta. Campo de selección que incluye varios colores</p> <p>El módulo brinda las siguientes opciones:</p> <ul style="list-style-type: none"> • <i>Guardar</i> • <i>Descartar</i>
3.	Se modifican los datos de la carpeta.
4.	El usuario selecciona la opción Guardar.
5.	<i>El módulo valida los datos.</i>
6.	<i>El módulo actualiza los datos de la carpeta.</i>
7.	Concluye así el requisito.

Pos-condiciones	
	Se modificó la carpeta satisfactoriamente.
Flujos alternativos	
Flujo alternativo 6.a Información incompleta	
	El módulo señala el o los campos obligatorios que no hayan sido introducidos y/o seleccionados y muestra el mensaje de información: <i>Los siguientes campos son obligatorios;</i> indicando los campos en cuestión.
	El usuario introduce y/o selecciona los datos.
	Volver al paso 5 del flujo básico
Pos-condiciones	
1.	NA
Flujos alternativo 6.b Información incorrecta	
	El módulo señala el o los campos incorrectos y muestra el mensaje de información: <i>Los siguientes campos son incorrectos;</i> indicando los campos en cuestión.
	El usuario introduce y/o selecciona correctamente los datos.
	Volver al paso 5 del flujo básico.
Pos-condiciones	
1.	N/A
Flujo alternativo 4.a Descartar cuando se hayan introducido datos	
1.	El usuario selecciona la opción <i>Descartar</i>
2.	El módulo muestra un mensaje de información: <i>Sus cambios se perderán y solicita confirmación ¿Está seguro que quiere abandonar la página?</i>
3.	El módulo brinda las siguientes opciones:

	<i>Aceptar</i>
	<i>Cancelar</i>
4.	El usuario selecciona la opción <i>Aceptar</i>
5.	El módulo elimina los datos introducidos y regresa al listado de carpetas.
6.	Concluye así el requisito.
Pos-condiciones	
1.	No se modifica la carpeta.
Flujo alternativo * <i>Descartar</i>	
1.	El módulo regresa al listado de carpetas.
2.	Concluye así el requisito.
Pos-condiciones	
1.	N/A
Validaciones	
	NA
Conceptos	NA NA
Requisitos especiales	
Asuntos pendientes	NA

Tabla 19: Descripción del RF Mover carpeta

Precondiciones	<p>El usuario debe estar autenticado en el sistema.</p> <p>Debe haberse incluido al menos dos carpetas.</p> <p>El usuario debe poseer los permisos para mover carpetas.</p>
Flujo de eventos	
Flujo básico mover carpeta	
1.	El usuario selecciona la opción mover.
2.	<p>El módulo debe permitir mover una carpeta hacia una carpeta señalada:</p> <p>(*) Nombre de carpeta a mover: representa el nombre de la carpeta que se le realizara la operación. Campo de selección.</p> <p>(*) Nombre de carpeta destino: representa el nombre de la carpeta destino de la carpeta a mover. Campo de selección.</p>
3.	Se mueve la carpeta hacia el destino seleccionado.
4.	<p>El módulo brinda las siguientes opciones:</p> <ul style="list-style-type: none"> • <i>Guardar</i> • <i>Descartar</i>
5.	El usuario selecciona la opción Guardar.
6.	<i>El módulo valida los datos.</i>
7.	<i>El módulo actualiza los datos de la carpeta.</i>

8.	Concluye así el requisito.
Pos-condiciones	
	Se movió la carpeta satisfactoriamente.
Flujos alternativos	
Flujo alternativo 6.a Información incompleta	
	El módulo señala el o los campos obligatorios que no hayan sido introducidos y/o seleccionados y muestra el mensaje de información: <i>Los siguientes campos son obligatorios;</i> indicando los campos en cuestión.
	El usuario introduce y/o selecciona los datos.
	Volver al paso 5 del flujo básico
Pos-condiciones	
1.	NA
Flujos alternativo 6.b Información incorrecta	
	El módulo señala el o los campos incorrectos y muestra el mensaje de información: <i>Los siguientes campos son incorrectos;</i> indicando los campos en cuestión.
	El usuario introduce y/o selecciona correctamente los datos.
	Volver al paso 5 del flujo básico.
Pos-condiciones	
1.	N/A
Flujo alternativo 4.a Descartar cuando se hayan introducido datos	
1.	El usuario selecciona la opción <i>Descartar</i>
2.	El módulo muestra un mensaje de información: <i>Sus cambios se perderán</i> y solicita confirmación <i>¿Está seguro que quiere abandonar la página?</i>

3.	El módulo brinda las siguientes opciones: <i>Aceptar</i> <i>Cancelar</i>	
4.	El usuario selecciona la opción <i>Aceptar</i>	
5.	El módulo elimina los datos introducidos y regresa al listado de carpetas.	
6.	Concluye así el requisito.	
Pos-condiciones		
1.	No se mueve la carpeta.	
Flujo alternativo * <i>Descartar</i>		
2.	El módulo regresa al listado de carpetas.	
3.	Concluye así el requisito.	
Pos-condiciones		
1.	N/A	
Validaciones		
	NA	
Conceptos	NA	NA
Requisitos especiales		
Asuntos pendientes	NA	

Tabla 20: Descripción del RF Listar carpetas

Precondiciones	<p>El usuario debe estar autenticado en el sistema.</p> <p>Debe haberse incluido al menos una carpeta.</p>
Flujo de eventos	
Flujo básico Listar carpetas	
1.	<p>El módulo debe permitir listar las carpetas existentes de forma ascendente o descendente según el campo nombre. En el listado se deben mostrar los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre de la carpeta • Cant. de subcarpetas asociadas • Color • Fecha de creación • Fecha de última modificación
2.	Concluye así el requisito.
Pos-condiciones	
1.	Se mostró el listado de las carpetas existentes.
Flujos alternativos	
Flujo alternativo 1.a Cambiar cantidad de elementos	

1.	El usuario cambia la cantidad de elementos a mostrar seleccionando la opción <i>Mostrar más</i> .	
2.	El listado es actualizado con la nueva cantidad de elementos a mostrar.	
3.	Concluye así el requisito.	
Pos-condiciones		
1.	Se muestra el listado de carpetas.	
Validaciones		
1.	N/A	
Conceptos	N/A	N/A
Requisitos especiales	N/A	
Asuntos pendientes	N/A	

Tabla 21: Descripción del RF Eliminar carpeta

Precondiciones	El usuario debe estar autenticado en el sistema.
	Debe haberse incluido al menos una carpeta.
	El usuario debe poseer los permisos necesarios para eliminar carpetas.
Flujo de eventos	
Flujo básico Eliminar carpetas	

1.	El usuario da clic encima de la carpeta a eliminar o selecciona las carpetas a eliminar del listado.
2.	El usuario selecciona la opción <i>Más</i> y se despliega un conjunto de opciones.
3.	Se selecciona la opción <i>Eliminar</i> .
4.	El módulo muestra un mensaje de confirmación <i>¿Está seguro que desea eliminar la carpeta?</i> y para el caso que se seleccionen varias carpetas se muestra el mensaje <i>¿Está seguro que desea eliminar las carpetas seleccionadas?</i>
5.	El módulo brinda las siguientes opciones: <ul style="list-style-type: none"> • <i>Aceptar</i> • <i>Cancelar</i>
6.	El usuario selecciona la opción <i>Aceptar</i> .
7.	Se muestra el mensaje de información: <i>La(s) carpeta(s) han sido eliminadas satisfactoriamente.</i>
8.	Concluye así el requisito.
Pos-condiciones	
1	Se eliminó la o las carpetas satisfactoriamente.
Flujos alternativo 5.a No se puede eliminar la carpeta	
1.	El módulo muestra el siguiente mensaje de información: <i>La carpeta no se puede eliminar, está vinculada a otra(s) carpeta(s).</i>

2.	Concluye así el requisito.	
Flujo alternativo * 5.b Cancelar		
1.	El usuario selecciona el botón <i>Cancelar</i> .	
2.	El módulo muestra los datos de la carpeta.	
3.	Concluye así el requisito.	
Pos-condiciones		
1	No se elimina la o las carpetas.	
Validaciones		
1.	N/A	
Conceptos	N/A	N/A
Requisitos especiales	N/A	
Asuntos pendientes	N/A	

Tabla 22: Descripción del RF Listar archivos

Precondiciones	El usuario debe estar autenticado en el sistema. Debe haberse incluido al menos una carpeta.
Flujo de eventos	
Flujo básico Listar archivos	
1.	El módulo debe permitir listar los archivos existentes de forma ascendente o descendente según el campo nombre. En el listado se deben mostrar los siguientes datos:

	<ul style="list-style-type: none"> • Nombre del archivo • Fecha de creación • Fecha de última modificación
2.	Concluye así el requisito.
Pos-condiciones	
1.	Se mostró el listado de los archivos existentes.
Flujos alternativos	
Flujo alternativo 1.a Cambiar cantidad de elementos	
2.	El usuario cambia la cantidad de elementos a mostrar seleccionando la opción <i>Mostrar más</i> .
3.	El listado es actualizado con la nueva cantidad de elementos a mostrar.
4.	Concluye así el requisito.
Pos-condiciones	
2.	Se muestra el listado de archivos.
Validaciones	
2.	N/A
Conceptos	N/A
Requisitos especiales	N/A
Asuntos pendientes	N/A

Tabla 23: Descripción del RF Duplicar carpeta

Precondiciones	<p>El usuario debe estar autenticado en el sistema.</p> <p>Debe haberse incluido al menos una carpeta.</p> <p>El usuario debe poseer los permisos necesarios para duplicar carpetas.</p>
Flujo de eventos	
Flujo básico duplicar carpetas	
2.	El usuario da clic encima de la carpeta a duplicar del listado.
3.	El usuario selecciona la opción <i>Más</i> y se despliega un conjunto de opciones.
4.	Se selecciona la opción <i>Duplicar</i> .
5.	El módulo muestra un mensaje de confirmación <i>¿Está seguro que desea duplicar la carpeta?</i>
6.	<p>El módulo brinda las siguientes opciones:</p> <ul style="list-style-type: none"> • <i>Aceptar</i> • <i>Cancelar</i>
7.	El usuario selecciona la opción <i>Aceptar</i> .
8.	Se muestra el mensaje de información: <i>La carpeta ha sido duplicada satisfactoriamente.</i>
9.	Concluye así el requisito.
Pos-condiciones	

1	Se duplicó la carpeta satisfactoriamente.	
Flujos alternativo 5.a No se puede duplicar la carpeta		
2.	El módulo muestra el siguiente mensaje de información: <i>La carpeta no se puede duplicar.</i>	
3.	Concluye así el requisito.	
Flujo alternativo * 5.b Cancelar		
2.	El usuario selecciona el botón <i>Cancelar</i> .	
3.	El módulo muestra los datos de la carpeta.	
4.	Concluye así el requisito.	
Pos-condiciones		
1	No se duplico la carpeta.	
Validaciones		
2.	N/A	
Conceptos	N/A	N/A
Requisitos especiales	N/A	
Asuntos pendientes	N/A	

Tabla 24: Descripción del RF Insertar color a carpetas

Precondiciones	El usuario debe estar autenticado en el sistema.
	Debe haberse incluido al menos una carpeta.

	El usuario debe poseer los permisos necesarios para insertar color a carpetas.
Flujo de eventos	
Flujo básico duplicar carpetas	
1.	El usuario da clic encima de la carpeta a insertar color del listado.
2.	Se despliega un conjunto de colores como opciones.
3.	Se selecciona el color <i>deseado</i> .
4.	El módulo muestra un mensaje de confirmación <i>¿Está seguro que desea insertar el color a la carpeta?</i>
5.	El módulo brinda las siguientes opciones: <ul style="list-style-type: none"> • <i>Aceptar</i> • <i>Cancelar</i>
6.	El usuario selecciona la opción Aceptar.
7.	Se muestra el mensaje de información: <i>Color insertado satisfactoriamente.</i>
8.	Concluye así el requisito.
Pos-condiciones	
1	Se insertó el color a la carpeta satisfactoriamente.
Flujos alternativo 5.a No se puede insertar color a la carpeta	

1.	El módulo muestra el siguiente mensaje de información: <i>A la carpeta seleccionada no se le puede insertar color.</i>
2.	Concluye así el requisito.
Flujo alternativo * 5.b Cancelar	
1.	El usuario selecciona el botón <i>Cancelar</i> .
2.	El módulo muestra los datos de la carpeta.
3.	Concluye así el requisito.
Pos-condiciones	
1	No se insertó el color a la carpeta.
Validaciones	
1.	N/A
Conceptos	N/A
Requisitos especiales	N/A
Asuntos pendientes	N/A

Tabla 25: Descripción del RF Eliminar archivos

Precondiciones	El usuario debe estar autenticado en el sistema.
	Debe haberse incluido al menos una carpeta.
	El usuario debe poseer los permisos necesarios para eliminar archivos.

Flujo de eventos	
Flujo básico Eliminar archivos	
1.	El usuario da clic encima del archivo a eliminar o selecciona los archivos a eliminar del listado.
2.	El usuario selecciona la opción <i>Más</i> y se despliega un conjunto de opciones.
3.	Se selecciona la opción <i>Eliminar</i> .
4.	El módulo muestra un mensaje de confirmación <i>¿Está seguro que desea eliminar el archivo?</i> y para el caso que se seleccionen varias carpetas se muestra el mensaje <i>¿Está seguro que desea eliminar los archivos seleccionados?</i>
5.	El módulo brinda las siguientes opciones: <ul style="list-style-type: none"> • <i>Aceptar</i> • <i>Cancelar</i>
6.	El usuario selecciona la opción <i>Aceptar</i> .
7.	Se muestra el mensaje de información: <i>Los archivos han sido eliminados satisfactoriamente.</i>
8.	Concluye así el requisito.
Pos-condiciones	
1	Se eliminó el o los archivos satisfactoriamente.
Flujos alternativo 5.a No se puede eliminar el archivo	

1.	El módulo muestra el siguiente mensaje de información: <i>El archivo no se puede eliminar.</i>	
2.	Concluye así el requisito.	
Flujo alternativo * 5.b Cancelar		
1.	El usuario selecciona el botón <i>Cancelar</i> .	
2.	El módulo muestra los datos de la carpeta.	
3.	Concluye así el requisito.	
Pos-condiciones		
1	No se elimina el o los archivos.	
Validaciones		
3.	N/A	
Conceptos	N/A	N/A
Requisitos especiales	N/A	
Asuntos pendientes	N/A	

Tabla 26: Descripción del RF Crear subcarpetas

Precondiciones	<p>El usuario debe estar autenticado en el sistema.</p> <p>Debe haberse incluido al menos una carpeta.</p> <p>El usuario debe poseer los permisos para crear subcarpetas.</p>
Flujo de eventos	

Flujo básico Crear subcarpeta	
1.	El usuario se adentra en una carpeta selecciona.
2.	El usuario selecciona la opción adjuntar carpeta.
3.	<p>El módulo debe permitir introducir y/o seleccionar los siguientes datos para crear una subcarpeta:</p> <p>(*) Nombre: representa el nombre de la carpeta. Campo de texto, admite el siguiente formato, letras, números o símbolos, con excepción de " # % * : < > ? / \ .</p> <p>Color: representa el color de la carpeta. Campo de selección que incluye varios colores</p>
4.	Se introducen y/o seleccionan los datos para crear una subcarpeta.
5.	<p>El módulo brinda las siguientes opciones:</p> <p style="text-align: center;"><i>Guardar</i></p> <p style="text-align: center;"><i>Descartar</i></p>
5.	El usuario selecciona la opción Guardar.
6.	<i>El módulo valida los datos.</i>
7.	<i>El módulo guarda los datos de la nueva subcarpeta.</i>
8.	Concluye así el requisito.
Pos-condiciones	
	Se incluyó una subcarpeta satisfactoriamente.
Flujos alternativos	

Flujo alternativo 6.a Información incompleta	
	El módulo señala el o los campos obligatorios que no hayan sido introducidos y/o seleccionados y muestra el mensaje de información: <i>Los siguientes campos son obligatorios;</i> indicando los campos en cuestión.
	El usuario introduce y/o selecciona los datos.
	Volver al paso 2 del flujo básico
Pos-condiciones	
1.	NA
Flujos alternativo 6.b Información incorrecta	
	El módulo señala el o los campos incorrectos y muestra el mensaje de información: <i>Los siguientes campos son incorrectos;</i> indicando los campos en cuestión.
	El usuario introduce y/o selecciona correctamente los datos.
	Volver al paso 2 del flujo básico.
Pos-condiciones	
1.	N/A
Flujo alternativo 4.a Descartar cuando se hayan introducido datos	
1.	El usuario selecciona la opción <i>Descartar</i>
2.	El módulo muestra un mensaje de información: <i>Sus cambios se perderán y solicita confirmación ¿Está seguro que quiere abandonar la página?</i>
3.	El módulo brinda las siguientes opciones: <i>Aceptar</i> <i>Cancelar</i>

4.	El usuario selecciona la opción <i>Aceptar</i>	
5.	El módulo elimina los datos introducidos y regresa al listado de carpetas.	
6.	Concluye así el requisito.	
Pos-condiciones		
1.	No se crea la subcarpeta.	
Flujo alternativo 4.b Descartar cuando se hayan introducido datos opción Cancelar		
1.	El usuario selecciona la opción <i>Cancelar</i> .	
2.	El módulo mantiene los cambios realizados.	
3.	Volver al paso 4 del flujo básico.	
Pos-condiciones		
1	N/A	
Flujo alternativo * <i>Descartar</i>		
2.	El módulo regresa al listado de carpetas.	
3.	Concluye así el requisito.	
Pos-condiciones		
1.	N/A	
Validaciones		
	NA	
Conceptos	NA	NA
Requisitos especiales		

Asuntos pendientes	NA
---------------------------	----

Tabla 27: Descripción del RF Adjuntar subcarpeta

Precondiciones	<p>El usuario debe estar autenticado en el sistema.</p> <p>Debe haberse incluido al menos una carpeta.</p> <p>El usuario debe poseer los permisos para crear subcarpetas.</p>
Flujo de eventos	
Flujo básico Adicionar subcarpetas	
1.	El usuario se adentra en una carpeta selecciona.
2.	El usuario selecciona la opción adicionar carpeta.
3.	<p>El módulo debe permitir seleccionar los siguientes datos de una lista para adicionar subcarpetas:</p> <p>(*) Nombre: representa el nombre de la subcarpeta. Campo de selección.</p>
4.	Se seleccionan los datos para crear una subcarpeta.
5.	<p>El módulo brinda las siguientes opciones:</p> <p><i>Guardar</i></p> <p><i>Descartar</i></p>

6.	El usuario selecciona la opción Guardar.
7.	<i>El módulo valida los datos.</i>
7.	<i>El módulo guarda los datos de la subcarpeta.</i>
8.	Concluye así el requisito.
Pos-condiciones	
	Se adicionó una subcarpeta satisfactoriamente.
Flujos alternativos	
Flujo alternativo 6.a Información incompleta	
	El módulo señala el o los campos obligatorios que no hayan sido introducidos y/o seleccionados y muestra el mensaje de información: <i>Los siguientes campos son obligatorios;</i> indicando los campos en cuestión.
	El usuario introduce y/o selecciona los datos.
	Volver al paso 5 del flujo básico
Pos-condiciones	
1.	NA
Flujos alternativo 6.b Información incorrecta	
	El módulo señala el o los campos incorrectos y muestra el mensaje de información: <i>Los siguientes campos son incorrectos;</i> indicando los campos en cuestión.
	El usuario introduce y/o selecciona correctamente los datos.
	Volver al paso 5 del flujo básico.
Pos-condiciones	
1.	N/A

Flujo alternativo 4.a Descartar cuando se hayan introducido datos	
1.	El usuario selecciona la opción <i>Descartar</i>
2.	El módulo muestra un mensaje de información: <i>Sus cambios se perderán</i> y solicita confirmación <i>¿Está seguro que quiere abandonar la página?</i>
3.	El módulo brinda las siguientes opciones: <i>Aceptar</i> <i>Cancelar</i>
4.	El usuario selecciona la opción <i>Aceptar</i>
5.	El módulo elimina los datos introducidos y regresa al listado de carpetas.
6.	Concluye así el requisito.
Pos-condiciones	
1.	No se adiciona la subcarpeta.
Flujo alternativo 4.b Descartar cuando se hayan introducido datos opción Cancelar	
2.	El usuario selecciona la opción <i>Cancelar</i> .
3.	El módulo mantiene los cambios realizados.
4.	Volver al paso 4 del flujo básico.
Pos-condiciones	
1	N/A
Flujo alternativo * <i>Descartar</i>	
3.	El módulo regresa al listado de carpetas.
4.	Concluye así el requisito.

Pos-condiciones		
1.	N/A	
Validaciones		
	NA	
Conceptos	NA	NA
Requisitos especiales		
Asuntos pendientes	NA	

Tabla 28: Descripción del RF Importar carpetas

Precondiciones	El usuario debe estar autenticado en el sistema.
	El usuario debe poseer los permisos para importar carpetas.
Flujo de eventos	
Flujo básico importar carpeta	
1.	El usuario selecciona la opción importar.
2.	El módulo debe permitir importar una carpeta con formato csv:

3.	Se importa la carpeta.
4.	El módulo brinda las siguientes opciones: <ul style="list-style-type: none"> • <i>Guardar</i> • <i>Descartar</i>
5.	El usuario selecciona la opción <i>Guardar</i> .
6.	<i>El módulo valida los datos.</i>
7.	<i>El módulo importa la carpeta.</i>
8.	Concluye así el requisito.
Pos-condiciones	
	Se importó la carpeta satisfactoriamente.
Flujo alternativo 4.a Descartar cuando se hayan introducido datos	
1.	El usuario selecciona la opción <i>Descartar</i>
2.	El módulo muestra un mensaje de información: <i>Sus cambios se perderán</i> y solicita confirmación <i>¿Está seguro que quiere abandonar la página?</i>
3.	El módulo brinda las siguientes opciones: <ul style="list-style-type: none"> <i>Aceptar</i> <i>Cancelar</i>
4.	El usuario selecciona la opción <i>Aceptar</i>
5.	El módulo elimina los datos introducidos y regresa al listado de carpetas.
6.	Concluye así el requisito.

Pos-condiciones		
1.	No se importa la carpeta.	
Flujo alternativo 4.b Descartar cuando se hayan introducido datos opción Cancelar		
1.	El usuario selecciona la opción <i>Cancelar</i> .	
2.	El módulo mantiene los cambios realizados.	
3.	Volver al paso 4 del flujo básico.	
Pos-condiciones		
1	N/A	
Flujo alternativo * <i>Descartar</i>		
3.	El módulo regresa al listado de carpetas.	
4.	Concluye así el requisito.	
Pos-condiciones		
1.	N/A	
Validaciones		
	NA	
Conceptos	NA	NA
Requisitos especiales		
Asuntos pendientes	NA	

Tabla 29: Descripción del RF Importar archivos

Precondiciones	El usuario debe estar autenticado en el sistema. El usuario debe poseer los permisos para importar archivos.
Flujo de eventos	
Flujo básico importar carpeta	
1.	El usuario selecciona la opción importar.
2.	El módulo debe permitir importar uno o varios archivos
3.	Se importa el o los archivos.
4.	El módulo brinda las siguientes opciones: <ul style="list-style-type: none"> • <i>Guardar</i> • <i>Descartar</i>
5.	El usuario selecciona la opción Guardar.
6.	<i>El módulo valida los datos.</i>
7.	<i>El módulo importa el o los archivos.</i>
8.	Concluye así el requisito.
Pos-condiciones	
	Se importó el archivo satisfactoriamente.
Flujo alternativo 4.a Descartar cuando se hayan introducido datos	

1.	El usuario selecciona la opción <i>Descartar</i>
2.	El módulo muestra un mensaje de información: <i>Sus cambios se perderán</i> y solicita confirmación <i>¿Está seguro que quiere abandonar la página?</i>
3.	El módulo brinda las siguientes opciones: <i>Aceptar</i> <i>Cancelar</i>
4.	El usuario selecciona la opción <i>Aceptar</i>
5.	El módulo elimina los datos introducidos y regresa al listado de archivos.
6.	Concluye así el requisito.
Pos-condiciones	
1.	No se importa el o los archivos.
Flujo alternativo 4.b Descartar cuando se hayan introducido datos opción Cancelar	
1.	El usuario selecciona la opción <i>Cancelar</i> .
2.	El módulo mantiene los cambios realizados.
3.	Volver al paso 4 del flujo básico.
Pos-condiciones	
1	N/A
Flujo alternativo * <i>Descartar</i>	
4.	El módulo regresa al listado de archivos.
5.	Concluye así el requisito.

Pos-condiciones		
1.	N/A	
Validaciones		
	NA	
Conceptos	NA	NA
Requisitos especiales		
Asuntos pendientes	NA	

Tabla 30: Descripción del RF Exportar carpetas

Precondiciones	El usuario debe estar autenticado en el sistema.
	Debe haberse incluido al menos una carpeta.
	El usuario debe poseer los permisos necesarios para exportar carpetas.
Flujo de eventos	
Flujo básico Exportar carpetas	
1.	El usuario da clic encima de la carpeta a exportar o selecciona las carpetas a exportar del listado.
2.	El usuario selecciona la opción <i>Más</i> y se despliega un conjunto de opciones.
3.	Se selecciona la opción <i>Exportar</i> .

4.	El módulo muestra un mensaje de confirmación <i>¿Está seguro que desea exportar la carpeta?</i> y para el caso que se seleccionen varias carpetas se muestra el mensaje <i>¿Está seguro que desea exportar las carpetas seleccionadas?</i>
5.	El módulo brinda las siguientes opciones: <ul style="list-style-type: none"> • <i>Aceptar</i> • <i>Cancelar</i>
6.	El usuario selecciona la opción <i>Aceptar</i> .
7.	Se muestra el mensaje de información: <i>La(s) carpeta(s) han sido exportadas satisfactoriamente.</i>
8.	Concluye así el requisito.
Pos-condiciones	
1	Se exportó la o las carpetas satisfactoriamente.
Flujo alternativo * 5.a Cancelar	
1.	El usuario selecciona el botón <i>Cancelar</i> .
2.	El módulo muestra los datos de la carpeta.
3.	Concluye así el requisito.
Pos-condiciones	
1	No se exporta la o las carpetas.
Validaciones	
1.	N/A

Conceptos	N/A	N/A
Requisitos especiales	N/A	
Asuntos pendientes	N/A	

Tabla 31: Descripción del RF Exportar archivos

Precondiciones	<p>El usuario debe estar autenticado en el sistema.</p> <p>Debe haberse incluido al menos un archivo.</p> <p>El usuario debe poseer los permisos necesarios para exportar archivos.</p>
Flujo de eventos	
Flujo básico Exportar archivos	
1.	El usuario da clic encima del archivo a exportar o selecciona los archivos a exportar del listado.
2.	El usuario selecciona la opción <i>Más</i> y se despliega un conjunto de opciones.
3.	Se selecciona la opción <i>Exportar</i> .
4.	El módulo muestra un mensaje de confirmación <i>¿Está seguro que desea exportar el archivo?</i> y para el caso que se seleccionen varias carpetas se muestra el mensaje <i>¿Está seguro que desea exportar los archivos seleccionados?</i>

5.	El módulo brinda las siguientes opciones: <ul style="list-style-type: none"> • <i>Aceptar</i> • <i>Cancelar</i>
6.	El usuario selecciona la opción <i>Aceptar</i> .
7.	Se muestra el mensaje de información: <i>Los archivos han sido exportados satisfactoriamente.</i>
8.	Concluye así el requisito.
Pos-condiciones	
1	Se exportó la o las carpetas satisfactoriamente.
Flujo alternativo * 5.a Cancelar	
1.	El usuario selecciona el botón <i>Cancelar</i> .
2.	El módulo muestra los datos de la carpeta.
3.	Concluye así el requisito.
Pos-condiciones	
1	No se exporta el o los archivos.
Validaciones	
1.	N/A
Conceptos	N/A
Requisitos especiales	N/A
Asuntos pendientes	N/A

Anexo 4: Diagramas de secuencia del diseño

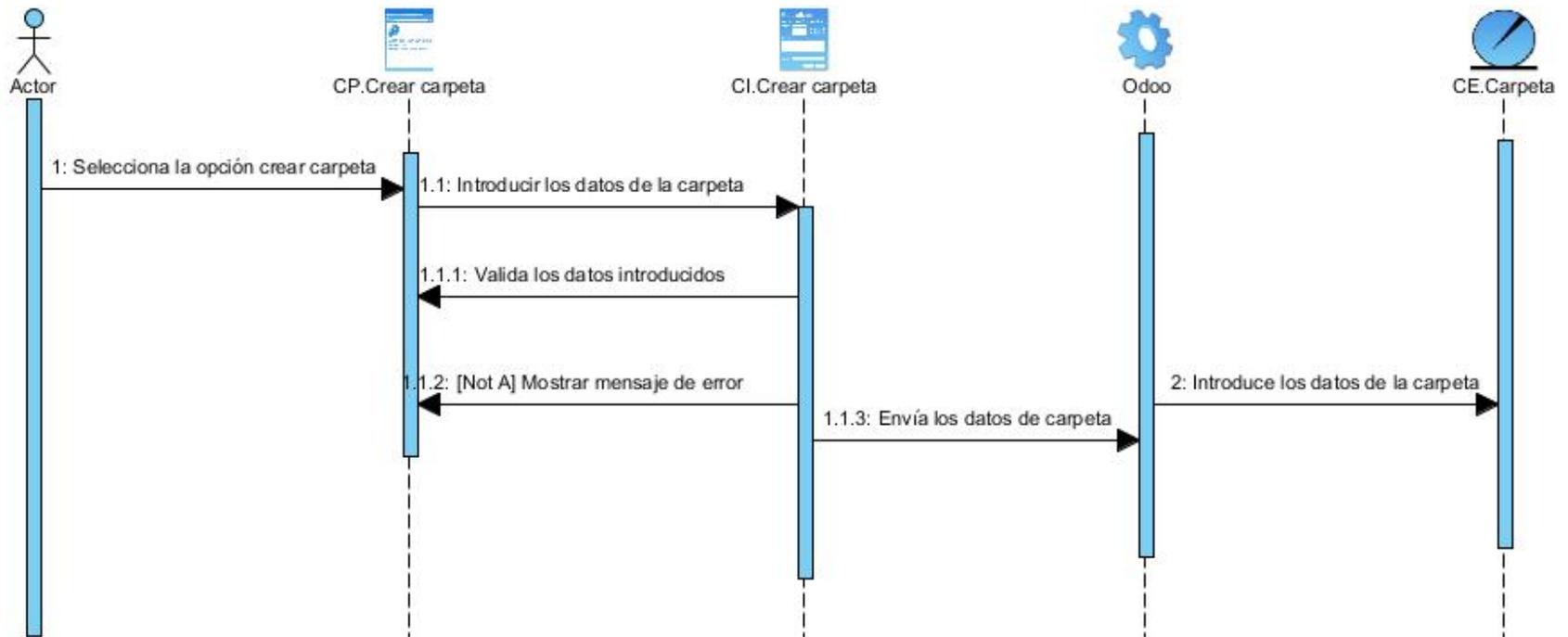


Figura 1: Diagrama de secuencia del diseño del RF Crear carpeta

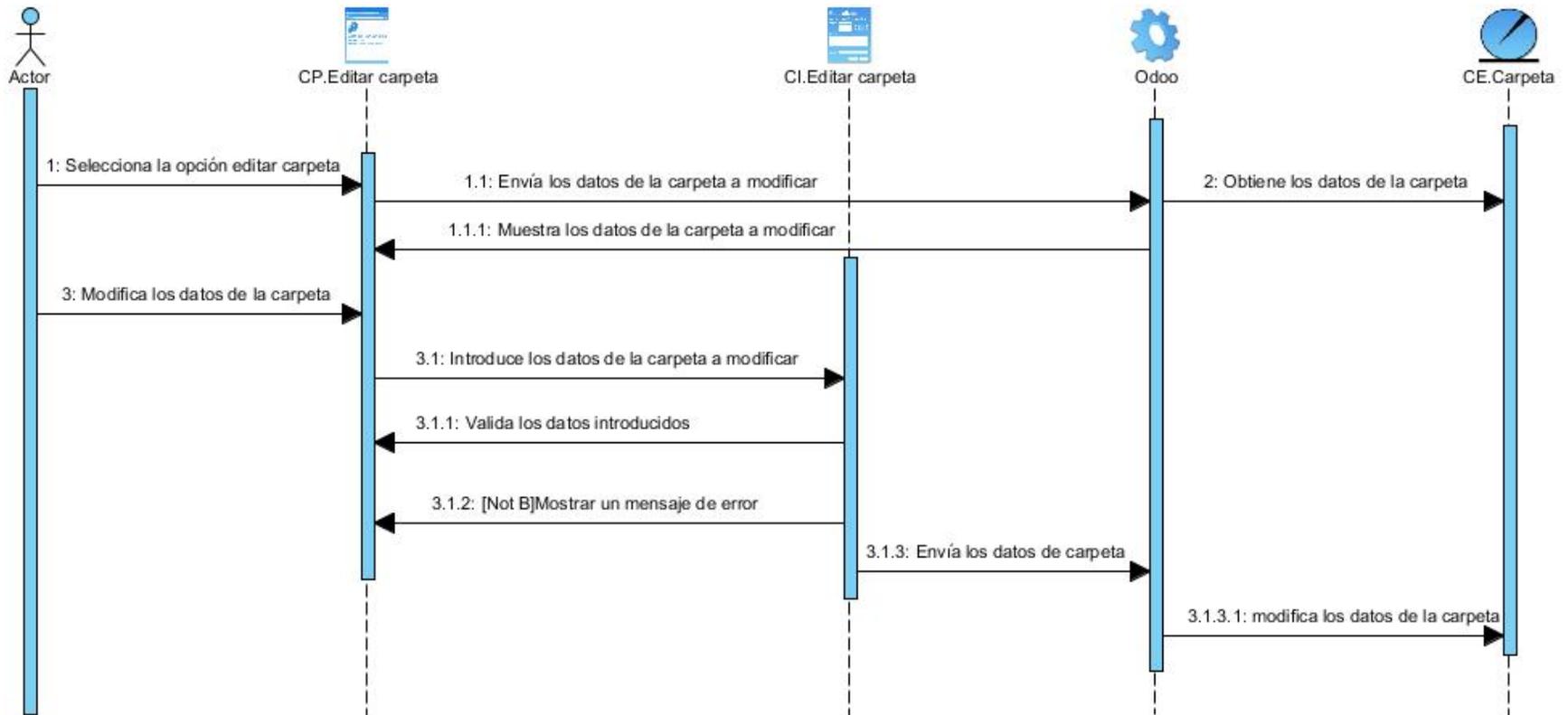


Figura 2: Diagrama de secuencia del diseño del RF Editar carpeta

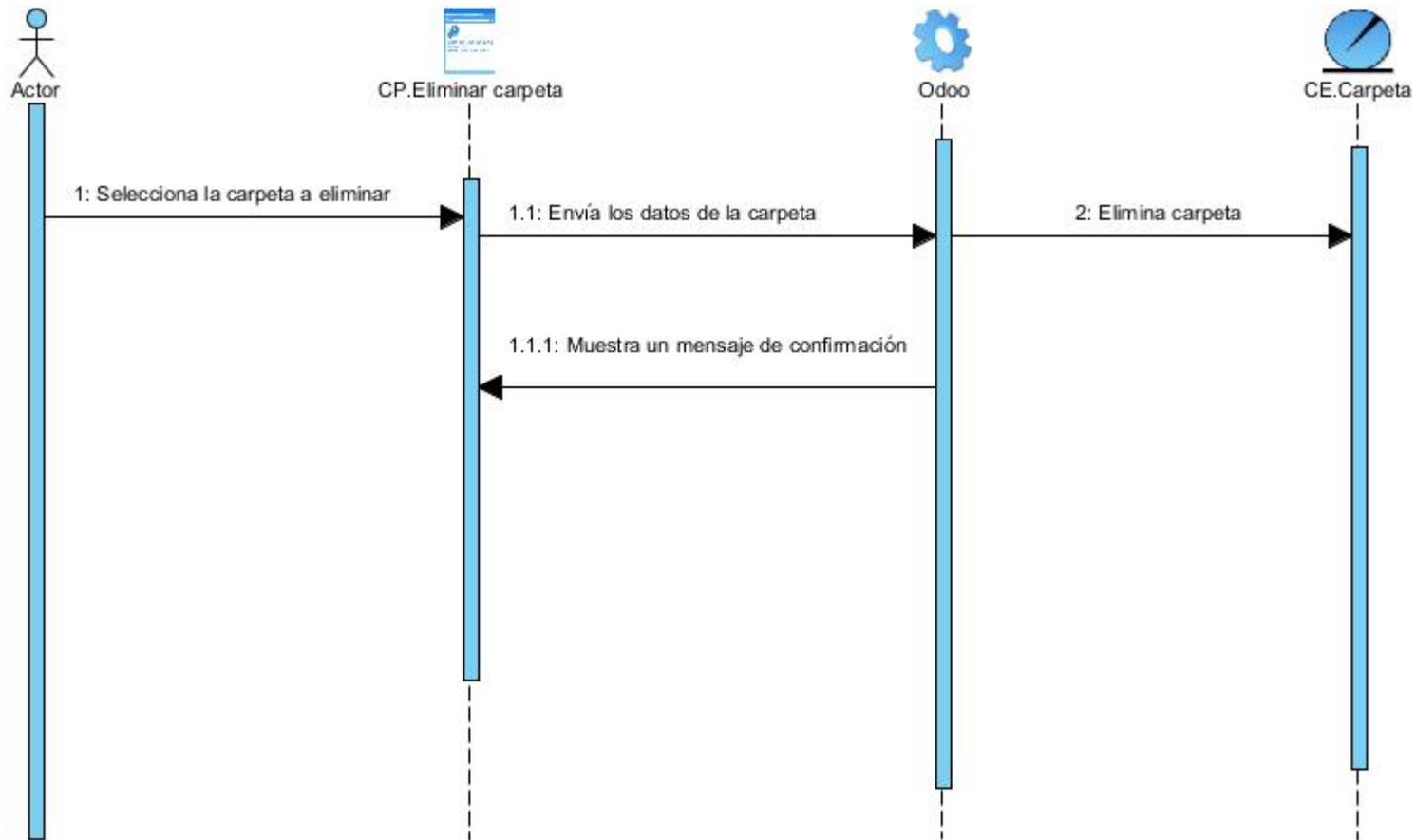


Figura 3: Diagrama de secuencia del diseño del RF Eliminar carpeta

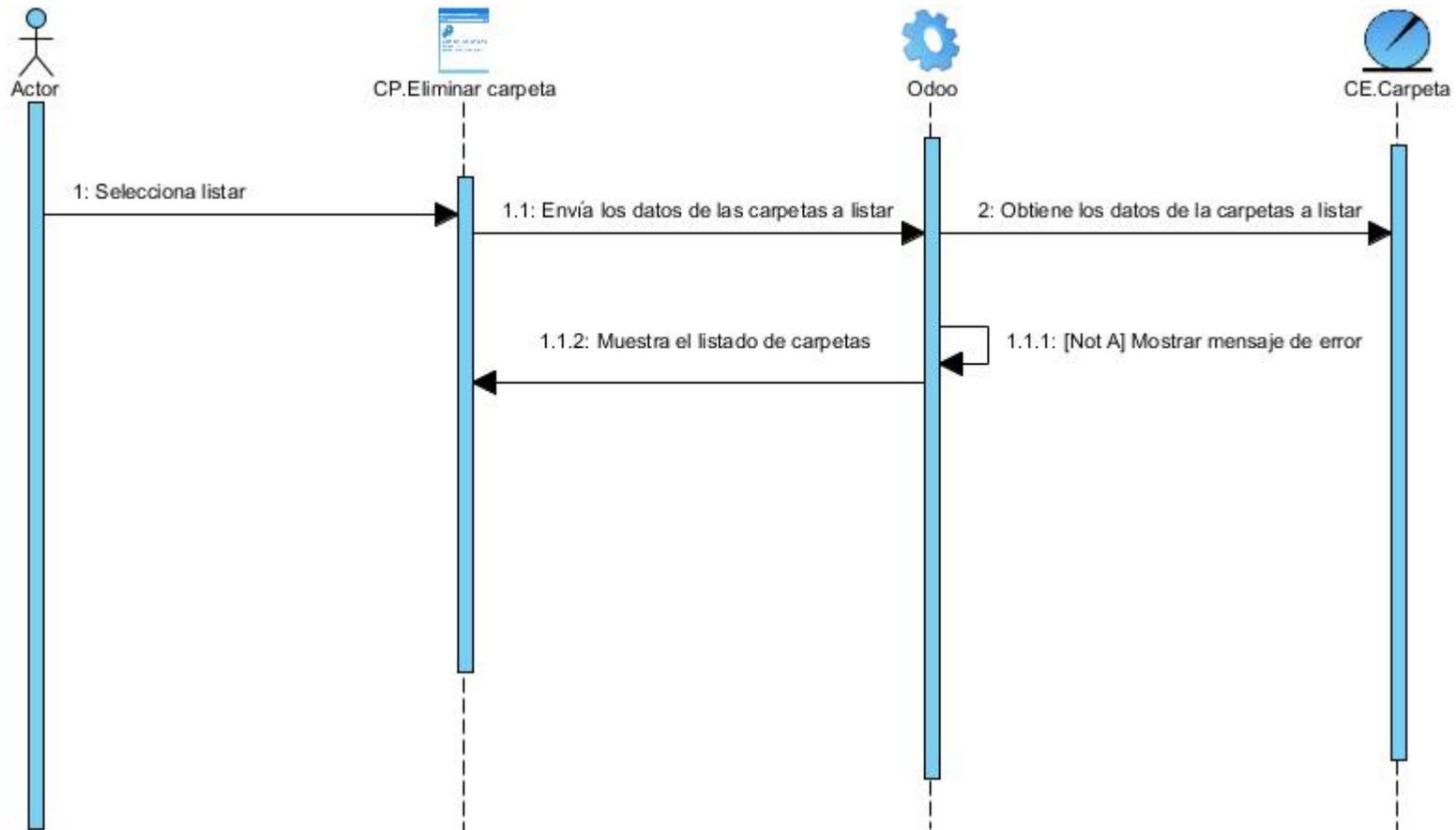


Figura 3: Diagrama de secuencia del diseño del RF Listar Carpetas

Anexo 5: Diagramas de clases del diseño

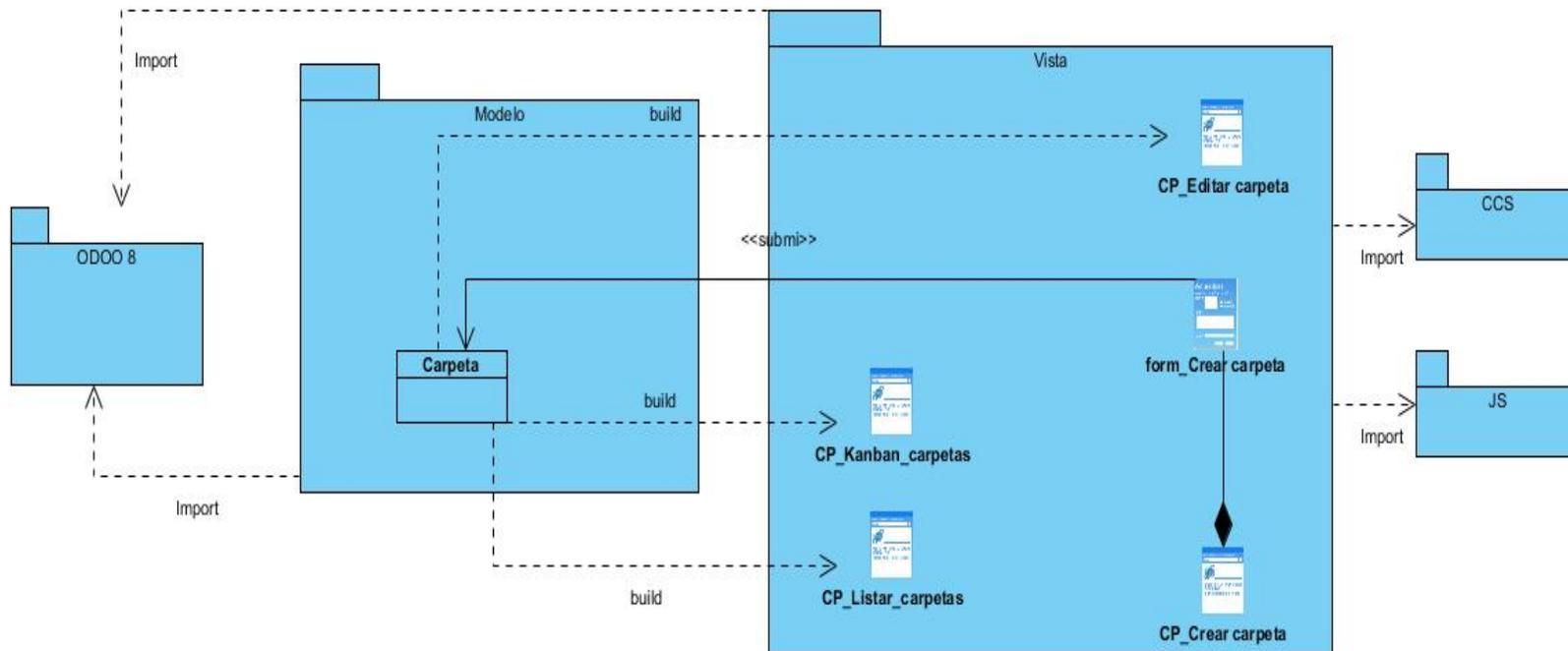


Figura 4: Diagrama de clases del diseño del RF Crear carpeta

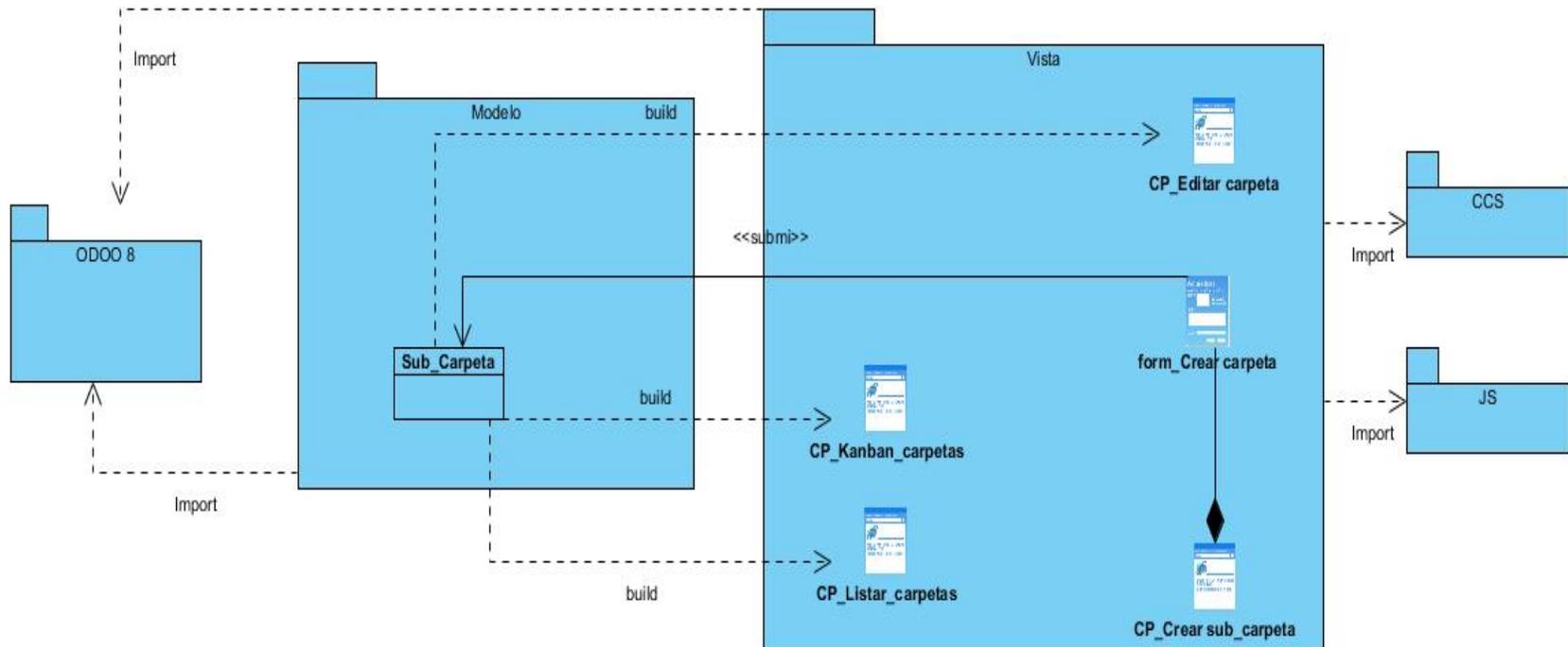


Figura 5: Diagrama de clases del diseño del RF Crear subcarpeta

Anexo 6: Diseños de caso de pruebas

Diseño de caso de prueba: RF Editar carpeta

Descripción general:

Permite al Usuario editar carpetas.

Condiciones de ejecución:

- El usuario debe estar autenticado en el sistema.
- El usuario debe tener permisos para editar carpetas.

SC 1 Editar carpeta

ID del escenario	Escenario	Nombre	Respuesta del módulo	Flujo Central
EC 1.1 Opción Editar carpeta	Selecciona la opción de Editar		Brinda la posibilidad de editar los siguientes datos: -Nombre Además brinda las opciones: -Aceptar -Cancelar	Módulo Gestor de Archivos /Editar
EC 1.3 Opción Aceptar	Selecciona la opción de aceptar	V	Valida los datos. Modifica la carpeta. Muestra el mensaje de información: "Modificado satisfactoriamente. La carpeta ha sido modificada satisfactoriamente." Brinda la opción: Continuar.	Módulo Gestor de Archivos /Editar/Aceptar

EC 1.4 Opción Cancelar	Selecciona la opción de cancelar		Elimina los datos introducidos y regresa a la interfaz que muestra el listado de carpetas.	Módulo Gestor de Archivos /Editar/ Cancelar
EC 1.5 Información incompleta	Información incompleta	v	Señala el campo incompleto (en este caso el campo Nombre) y muestra el mensaje de error: Rellene este campo.	Módulo Gestor de Archivos /Editar/Aceptar
EC 1.6 La carpeta ya existe	Selecciona la opción Aceptar pero la carpeta ya existe	v	Devuelve el mensaje: "Error, La carpeta ya existe. Brinda la opción Continuar."	Módulo Gestor de Archivos /Editar/Aceptar

Anexo 7: No Conformidades del Módulo Gestor de archivos 1ra iteración

Tabla 32: No Conformidades del Módulo Gestor de archivos 1ra iteración

No. CP	Caso de prueba	No conformidades				
		Alta	Media	Baja	No procede	Total
1	CP Crear carpetas	-	-	2	-	2
2	CP Editar carpetas	-	-	1	-	1
3	CP Mover carpetas	-	-	1	-	1

4	CP Listar carpetas	-	-	-	-	-
5	CP Eliminar carpetas	-	-	1	-	1
6	CP Importar archivos	-	-	-	-	-
7	CP Exportar archivos	-	-	-	-	-
8	CP Listar archivos	-	-	1	-	1
9	CP Duplicar carpetas	-	-	-	-	-
10	CP Insertar color a las carpetas	-	-	1	-	1
11	CP Eliminar archivos	-	-	-	-	-
12	CP Crear subcarpetas	-	-	2	-	2
13	CP Adicionar subcarpeta	-	-	1	-	1
14	CP Buscar carpetas y archivos	-	-	-	-	-
15	CP Importar Carpetas	-	-	-	-	-
16	CP Exportar Carpetas	-	-	-	-	-
	Total	-	-	10	-	10

Tabla 33: No Conformidades del Módulo Gestor de archivos 2da iteración

No. CP	Caso de prueba	No conformidades				
		Alta	Media	Baja	No procede	Total
1	CP Crear carpetas	-	-	-	-	-
2	CP Editar carpetas	-	-	-	-	-
3	CP Mover carpetas	-	-	1	-	1
4	CP Listar carpetas	-	-	-	-	-
5	CP Eliminar carpetas	-	-	2	-	2
6	CP Importar archivos	-	-	-	-	-
7	CP Exportar archivos	-	-	-	-	-
8	CP Listar archivos	-	-	-	-	-
9	CP Duplicar carpetas	-	-	-	-	-
10	CP Insertar color a las carpetas	-	-	-	-	-
11	CP Eliminar archivos	-	-	-	-	-
12	CP Crear subcarpetas	-	-	1	-	1
13	CP Adicionar subcarpeta	-	-	-	-	-
14	CP Buscar carpetas y archivos	-	-	-	-	-

15	CP Importar Carpetas	-	-	-	-	-
16	CP Exportar Carpetas	-	-	-	-	-
	Total	-	-	4	-	4

Anexo 8: Descripción de No Conformidades 1ra iteración

Tabla 34: Descripción de No Conformidades 1ra iteración

No. NC	No. CP	Clasificación	Complejidad	Estado
1	1	Correspondencia con otro artefacto	Baja	RA
2	1	Error de caso de prueba	Baja	RA
3	2	Redacción	Baja	RA
4	3	Correspondencia con otro artefacto	Baja	RA
5	5	Error de caso de prueba	Baja	RA
6	8	Correspondencia con otro artefacto	Baja	RA
7	10	Correspondencia con otro artefacto	Baja	RA
8	12	Error de caso de prueba	Baja	RA

9	12	Redacción	Baja	RA
10	13	Correspondencia con otro artefacto	Baja	RA

Tabla 35: Descripción de No Conformidades 2da iteración

No. NC	No. CP	Clasificación	Complejidad	Estado
1	3	Correspondencia con otro artefacto	Baja	RA
2	5	Correspondencia con otro artefacto	Baja	RA
3	5	Redacción	media	RA
4	12	Correspondencia con otro artefacto	Baja	RA