



Universidad de las Ciencias Informáticas

Centro de Informatización Universitaria

Facultad 1

Herramienta para la migración de datos bajo la norma ISAD (G)

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

Autor: Ilda López Coll

Tutores: MSc. Laritza Cabrera Barroso
Lic. Vadim Turiño Sviatchenko

**Ciudad de La Habana
Junio, 2012**



"Muchos me dirán aventurero, y lo soy, sólo que de un tipo diferente y de los que ponen el pellejo para demostrar sus verdades."

Che

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y ofrezco 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_____.

Firma del Autor
Ilda López Coll

Firma del Tutor
MSc. Laritza Cabrera Barroso

Firma del Tutor
Lic. Vadim Turiño Sviatchenko

RESUMEN

El departamento Gestión Documental de la Universidad de las Ciencias Informáticas, tiene como función principal, el desarrollo de sistemas automatizados que facilitan la gestión de los documentos de archivo en distintas instituciones del país. Muchas de estas instituciones tienen digitalizada parte de la información que preservan, empleando Winisis, un software diseñado para crear y administrar bases de datos. Las bases de datos utilizadas por esta herramienta son incompatibles con las empleadas por las aplicaciones desarrolladas en el departamento. La presente investigación tiene como objetivo, desarrollar un sistema que permita la migración de datos ISIS hacia los sistemas gestores de bases de datos MySQL y PostgreSQL, bajo la Norma Internacional General de Descripción Archivística.

Este sistema informático garantizará mayor rapidez en el proceso de implantación y puesta en práctica de los sistemas desarrollados por el departamento, en las instituciones clientes; logrando que estas entidades puedan preservar la información contenida en sus herramientas de almacenamiento.

Palabras clave: Conjunto Integrado de Sistemas de Información, Norma Internacional General de Descripción Archivística, Migración de datos, MySQL, PostgreSQL.

AGRADECIMIENTOS

Ha llegado el momento de enfrentarme al mundo como profesional y son innumerables las personas a las que tengo que agradecerles por ayudarme a llegar aquí.

A mi mamá: "gracias mami; sin tu apoyo y sin tu ejemplo no estaría ahora aquí cumpliendo mis sueños, ¡ERES MI EJEMPLO MÁS GRANDE! El ejemplo que quiero seguir y es a ti a quien quiero premiar con mi esfuerzo, gracias má por sacrificararte y darme todo, pero sobre todo el amor y cariño que siempre me has regalado." Este es tu regalo.

A mi pequeño bebé Jarol, aunque te hayas convertido en un hombrecito, aún recuerdo la primera vez que te tuve entre mis brazos, fue como tocar el cielo en ese preciso instante. Eres esa personita que me dio fuerzas para seguir diciendo te quiero. Eres y serás lo más importante y hermoso que tenga en la vida. No olvides nunca que mamá te adora y ama con todo el corazón. Te amo.

A mi hermana Ismary por sus consejos y apoyarme en cada decisión.

A mi hermana Ismaray por ser mi guía y aunque no pueda verme ahora, sé que estaría muy orgullosa de mí.

A mi otra madre Amarilis, porque me has legado el tesoro más valioso que puede dársele a un hijo: amor. Porque sin escatimar esfuerzo alguno, has sacrificado gran parte de tu vida para formarme y educarme.

A mis primos Leo, Amanda, Amalia, Lideisy, Deisy, Yaisy, Jorge, Alexis, Pedro y a mis tías por siempre darme fuerzas y ánimo para seguir adelante y triunfar en la vida.

A mis abuelos que donde quiera que estén se sientan orgullosos de su nieta.

A mi Tyto, porque eres de esa clase de personas que todo lo comprenden y dan lo mejor de sí mismos sin esperar nada a cambio... porque sabes escuchar y brindar ayuda cuando es necesario... porque cambias todo lo que encuentras a tu paso... porque te has convertido en parte esencial de mi vida. Te amo.

Al tío Ino por su apoyo, comprensión e incluirme como parte de su familia.

A mi tutora Laritza por todo su apoyo incondicional, por dedicarme siempre su tiempo, por transmitirme sus enseñanzas y demostrar que siempre puedo contar con ella para lo que sea, porque además de ser tutora se convirtió en una amiga.

A Yoani por ayudarme durante el desarrollo del trabajo de diploma y siempre estar disponible a cualquier hora.

A mi tutor Vadim, por la ayuda brindada.

A Liuba, Lisbet, Yanet, Inalbis, Leanet, Alibech, Maire, las mellizas, Ayane, Vila, Ricardo, Joaquín, Gustavo, Rey, Yoency, Marry, Méndez, Alfredo, Adonis, Yusniel, José, Rayko, Michel, Rosales, Durán amigos que marcarán para siempre mi vida en esta Universidad.

A las muchachitas del proyecto, por todos los buenos y malos momentos que pasamos juntas. A mi gente del grupo, los antiguos y los nuevos, por ser parte de mis mejores días en esta Universidad.

A mis dos eternos guías Fidel y el Che, porque sin su ejemplo no hubiera llegado a ser lo que soy.

A Dios por permitirme por ayudarme a superar todas las adversidades que han estado a mi paso, a levantarme cada vez que me caigo y por demostrarme que la vida no es fácil, pero que vale la pena vivirla.

A todos los que de una forma u otra han hecho de mí la persona que soy.

DEDICATORIA

A mi bebé, mi mamá y mi familia por su amor, por su preocupación, por su paciencia, su confianza en mí, por cada granito de arena que pusieron en mi vida y en mi futuro, para ustedes es este regalo.

Índice

Introducción	1
Capítulo 1 Fundamentación Teórica	5
1.1 Introducción	5
1.2 Conceptos fundamentales.....	5
1.3 Normas de descripción archivística	8
1.4 Tecnologías	12
1.5 Migración de datos.....	16
1.6 Estado del arte de las herramientas para la migración de datos ISIS hacia los SGBD MySQL y PostgreSQL	17
1.7 Aplicaciones de Escritorio y Aplicaciones Web.....	18
1.8 Lenguaje de programación.....	19
1.8.1 Lenguaje de programación Java	19
1.9 Herramienta de desarrollo de software.....	20
1.10 Metodología de Desarrollo de Software.....	20
1.11 Herramienta CASE para modelar UML.....	23
1.12 Conclusiones parciales	24
Capítulo 2 Características de la herramienta para la migración de datos hacia la norma ISAD (G)	25
2.1 Introducción	25
2.2 Modelo de dominio.....	25
2.3 Especificación de requisitos de software	26
2.3.1 Requisitos funcionales.....	27
2.3.2 Requisitos no funcionales.....	28
2.4 Propuesta de solución.....	30
2.5 Definición de actores y casos de uso	31
2.5.1 Patrones de caso de uso utilizados	32
2.5.2 Descripciones textuales de los casos de uso del sistema.....	33
2.6 Conclusiones parciales	39
Capítulo 3 Análisis y diseño del sistema	40
3.1 Introducción	40

3.2 Análisis del sistema.....	40
3.3 Diseño del sistema.....	43
3.4 Arquitectura del sistema.....	43
3.4.1 Patrones de arquitectura	43
3.4.2 Patrones de diseño	45
3.5 Tratamiento de errores	48
3.6 Conclusiones parciales	48
Capítulo 4 Implementación y prueba.....	49
4.1 Introducción	49
4.2 Implementación.....	49
4.2.1 Diagrama de despliegue.....	49
4.2.2 Diagrama de componentes	50
4.3 Estándar de codificación	51
4.4 Pruebas de software	53
4.4.1 Pruebas de caja negra	53
4.5 Conclusiones parciales	57
Conclusiones generales.....	58
Recomendaciones	59
Glosario de términos.....	60
Referencias bibliográficas	61
Bibliografía consultada.....	65
Anexos.....	67

Introducción

La época actual es llamada Era de la Información. En este sentido se puede expresar que la información constituye un producto o un elemento vital para cualquier organización, así como los recursos humanos, materiales y financieros, sin los cuales no podría sobrevivir ni funcionar la misma. Debido al enorme volumen de información que se genera diariamente en el mundo y a la necesidad de su almacenamiento, surge el archivo, institución o parte estructural de ella, que realiza la recepción, organización y conservación de los documentos para su utilización (CASTRO, 2009), de forma tal que puedan ser accedidos y consultados en cualquier momento.

La archivística permite proporcionar un procedimiento metodológico a los documentos de archivo, conducente a su organización, conservación, accesibilidad y recuperación de la información de forma expedita, para los fines que sean útiles (GARCÍA, 2006). Además, se encarga del estudio teórico y práctico de los principios, procedimientos y problemas concernientes al almacenamiento de documentos, con el objetivo que dicha documentación se mantenga en el tiempo, pueda ser consultada y clasificada (LOPERA, 2009). Naturalmente, como todo recurso, la información debe ser conservada y gestionada eficazmente, pues el ser humano día a día realiza un conjunto de actividades relacionadas con el quehacer archivístico, que comprende desde la elaboración del documento hasta su eliminación o conservación permanente; este proceso recibe el nombre de Gestión Archivística.

En la Gestión Archivística, se debe seguir una serie de pautas, normas y requisitos para cumplir con los estándares internacionales determinados para la descripción de los documentos de archivo. Esta es una actividad que debe ser realizada en forma conjunta a la ordenación, con el fin de ayudar en la consulta tanto a usuarios externos, como a la administración productora, facilitándole al archivero¹ el control y administración de la documentación. La norma Internacional General de Descripción Archivística conocida como ISAD (G) por sus siglas en inglés, es un ejemplo de estas. Esta norma significó un avance en la cooperación archivística pues permite que todas las entidades describan sus documentos bajo un mismo formato, garantizando así una mejor comprensión de la descripción realizada. Además, la norma ISAD (G) contiene un conjunto de reglas generales para la descripción archivística que pueden aplicarse con independencia del tipo documental o del soporte físico de los documentos de archivo.

En el año 2007, en el departamento de Gestión Documental perteneciente al Centro de Informatización Universitaria de la Universidad de las Ciencias Informáticas se desarrolló el Sistema de Gestión de

¹ Profesional que gestiona los documentos a lo largo de todo su ciclo de vida, desde su creación en las dependencias administrativas hasta su llegada al archivo.

Documentos Históricos ArchiVenHIS. Esta herramienta permite, entre otras funcionalidades, realizar la conformación del cuadro de clasificación en que se encuentra estructurada la documentación almacenada en un archivo y la descripción de cada uno de esos niveles de organización según la norma ISAD (G). En varias instituciones se han realizado personalizaciones del sistema y se han implementado otros que permiten realizar estas descripciones, de acuerdo a las especificidades de los clientes.

Muchas instituciones ya habían optado por la automatización de sus actividades y han empleado algunas herramientas para realizar estas descripciones, como el Winisis, un software de código privado y acceso gratuito, diseñado por la UNESCO² con el objetivo de administrar y crear bases de datos, así como almacenar y recuperar información. Cuando se implantan las nuevas soluciones desarrolladas por el departamento, es conveniente aprovechar los datos que ya se tienen en esta herramienta y realizar su migración hacia las bases de datos que emplean los sistemas implementados.

Con el uso de Winisis, el trabajo se ha dificultado, debido a que la información comprendida en sus bases de datos no está estandarizada bajo ninguna norma de descripción, lo que impide el intercambio de los datos entre esta herramienta y las bases de datos que utilizan los sistemas desarrollados, al no corresponderse los campos empleados en ambas. La búsqueda o modificación en ellas es tediosa pues debe ser realizada de forma manual lo que ocasiona una pérdida elevada de tiempo y esfuerzo por parte del equipo de desarrollo.

Teniendo en cuenta la situación problemática anteriormente expresada, la presente investigación contribuye a solucionar el siguiente **problema científico**: ¿cómo facilitar la migración de datos de descripción archivística realizados con la herramienta Winisis hacia los sistemas gestores de bases de datos MySQL y PostgreSQL?

El **objeto de estudio** del presente trabajo comprende el proceso de migración de datos y el **campo de acción** está orientado a la migración de datos ISIS³ hacia los sistemas gestores de bases de datos MySQL y PostgreSQL.

En el contexto investigativo se define como **objetivo general**: desarrollar un sistema que permita la migración de datos ISIS hacia los sistemas gestores de bases de datos MySQL y PostgreSQL bajo la norma ISAD (G).

A partir del objetivo general se identifican los siguientes **objetivos específicos**:

- Fundamentar los principales conceptos y definiciones asociados a la gestión archivística y el

² Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura.

³ Conjunto Integrado de Sistemas de Información.

proceso de migración de datos.

- Analizar el estado del arte de los sistemas migradores de datos ISIS.
- Diseñar la propuesta del sistema migrador de datos ISIS.
- Implementar el sistema para la migración de datos.
- Validar la propuesta de migración de datos.

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

- Revisión de los aspectos teóricos conceptuales sobre la descripción archivística realizada bajo la norma ISAD (G).
- Investigación de la estructura del sistema de almacenamiento de datos Winisis.
- Estudio de los sistemas gestores de bases de datos MySQL y PostgreSQL.
- Análisis del proceso de estandarización de datos bajo el formato de la norma ISAD (G).
- Definición de los requisitos funcionales y no funcionales para el sistema.
- Elaboración del Modelo de dominio.
- Diseño de las interfaces de usuario.
- Implementación de los elementos del diseño.
- Definición de las configuraciones del entorno de prueba.
- Implementación y ejecución de las pruebas del sistema.

Métodos teóricos:

- Analítico - Sintético: permite analizar las teorías, documentos y sistemas, facilitando la extracción de elementos importantes relacionados con el proceso de estandarización en la descripción archivística.
- Histórico - Lógico: permite analizar el recorrido histórico de los sistemas para la migración de datos ISIS y la estandarización de datos en el proceso de descripción de documentos de archivos.

Método empírico:

Fue utilizada la técnica de la entrevista para obtener a través del personal que labora en procesos relacionados con la descripción archivística, en distintas instituciones, criterios en vista al desarrollo del sistema, garantizando así el perfeccionamiento de la solución propuesta.

Se espera obtener análisis, diseño, implementación y prueba del sistema para la migración de datos hacia los sistemas gestores de bases de datos MySQL y PostgreSQL bajo la norma ISAD (G).

El presente trabajo de diploma está estructurado en cuatro capítulos, conclusiones, recomendaciones, así como las referencias bibliográficas, bibliografía, glosario de términos y anexos, donde se da cumplimiento a los objetivos planteados en el trabajo. A continuación se describe los principales aspectos abordados en cada uno de los capítulos:

Capítulo 1: Fundamentación teórica, se exponen los principales conceptos relacionados con la gestión archivística y el proceso de migración de datos, así como un estudio del estado del arte de los sistemas para la migración de datos ISIS. Se analiza la estructura de almacenamiento del sistema Winisis y los sistemas gestores de bases de datos MySQL y PostgreSQL. Se detallan las tecnologías y metodología a usar para el desarrollo de la solución propuesta.

Capítulo 2: Características de la herramienta para la migración de datos hacia la norma ISAD (G), se confecciona la fundamentación de la propuesta que se realiza, se describe el flujo de los procesos involucrados en la solución a modo de comprenderlos totalmente. Se plantea la elaboración del modelo de dominio, los requisitos funcionales y no funcionales del sistema, así como la solución propuesta para el sistema que se desea diseñar.

Capítulo 3: Análisis y diseño del sistema para la migración de datos, de manera general se describe la arquitectura de la solución propuesta. Se exponen a través de un conjunto de artefactos definidos por la metodología seleccionada la solución que se le da al problema en cuestión, que incluyen los diagramas pertinentes; posibilitando el modelado de la solución de manera visual.

Capítulo 4: Implementación y validación de los resultados, contempla la implementación del sistema desarrollado, incluyendo los diagramas de componentes y despliegue como elementos fundamentales de la fase de implementación. Se efectúa además la validación y prueba de la solución implementada a partir de los requisitos previamente definidos.

Capítulo 1 Fundamentación Teórica

1.1 Introducción

En este acápite se realiza un estudio de los aspectos teóricos sobre la gestión archivística que sirven de punto de partida para la investigación. Se presentan algunos conceptos esenciales en el marco de la migración de datos, que facilitan una mejor visión de la propuesta de solución desarrollada. Se hace alusión a algunos software existentes relacionados con la migración de datos ISIS y se seleccionan las herramientas y metodología de desarrollo de software utilizadas, mencionando sus principales características, ventajas y desventajas.

1.2 Conceptos fundamentales

Archivo

Para autores como Cruz Mundet un archivo consiste en uno o más conjuntos de documentos, sea cual sea su fecha, su forma y soporte material acumulados en un proceso natural por una persona o institución pública o privada en el transcurso de su gestión, conservados, respetando el orden en que fueron creados, para servir como testimonio en formación para la persona o institución que los produce, para los ciudadanos o para servir de fuente de historia (MUNDET, 1996).

Por la idea expuesta anteriormente, se puede expresar que un archivo es una institución en la cual se recogen varios documentos de cualquier fecha y soporte, siendo estos el resultado de la actividad de una persona o entidad pública o privada, atendidos por un personal competente para esta labor, que los mantiene organizados y conservados para servir a personas o entidades como testimonio jurídico o administrativo, científico o cultural.

Documento de archivo

Según Mayra Mena Mugica, un documento de archivo es el testimonio material de un hecho o acto realizado en el ejercicio de sus funciones por personas físicas o jurídicas, públicas o privadas, de acuerdo con características de tipo material y formal (MUGICA, 2005a).

Otros autores plantean que un documento de archivo, es aquel en el que se refleja el testimonio material de un hecho o acto realizado por una persona natural o jurídica en el ejercicio de sus funciones, y que por su valor administrativo, fiscal, legal, científico, económico, histórico o cultural, debe ser objeto de conservación (CASTRO, 1960).

Por tanto, se considera documento de archivo, a todo aquel registro de información independiente de su soporte físico, reflejan la expresión del trabajo de la creación humana, en su constante desarrollo con la sociedad.

Gestión documental

Según el diccionario de Terminología Archivística elaborado por el Consejo Internacional de Archivos, la gestión documental es un área de la administración general que se encarga de garantizar la economía y eficiencia en la creación, mantenimiento, uso y disposición de los documentos administrativos durante todo su ciclo de vida.

Para Mayra Mena Mugica, la gestión documental abarca el ciclo de vida completo de los documentos, es decir, el tratamiento secuencial y coherente que se da a los documentos desde que se producen o reciben en las distintas unidades hasta el momento en que son eliminados o conservados, en función de su valor testimonial o histórico (MUGICA, 2005b).

Por tanto, se puede manifestar que la gestión documental abarca las acciones que le son realizadas a los documentos durante su ciclo de vida, incluyendo el tratamiento y conservación que reciben desde que son creados por alguna entidad o persona hasta el momento de su eliminación o conservación permanente.

En la actualidad la gestión documental es un pilar fundamental a ejecutar en todas las organizaciones para una mejor competencia y desarrollo de la misma.

Descripción archivística

La descripción archivística es el análisis realizado por el archivero sobre los fondos y los documentos de archivo agrupados natural o artificialmente, a fin de sintetizar y condensar la información en ellos contenida para ofrecerla a los interesados. Equivale a dar al documento de archivo o sus agrupaciones sus señas de identidad, aquellos rasgos que los definen con precisión (eligiendo los elementos que mejor los identifiquen) y que permiten y facilitan la comunicación (consulta y recuperación) (HERRERA, 1991). Es el proceso mediante el cual se enumeran los elementos formales e informativos de los documentos, que permite conocer sus rasgos determinantes y de manera general, las características estructurales y funcionales de la institución que les dio origen, o bien las actividades o temas desarrollados por el personaje que los generó o recopiló (LEÓN, 2006).

El proceso de descripción es un conjunto de principios y técnicas que se han desarrollado en los últimos 100 años para ser aplicadas en diferentes fondos documentales de origen público, privado y particular.

De manera que la descripción trata no solo las características como el tipo de documento (jurídico o diplomático), el contenido (fecha de creación y lugar), sino también los datos para su localización. La descripción de un documento es la cima del trabajo archivístico, realiza la misma tarea que un documento de archivo: informar.

Para que instituciones e individuos puedan utilizar la información contenida en un archivo es necesario tener los documentos ordenados, labor que desempeña el archivero. Por esto existen niveles de descripción o niveles de organización para describir los documentos, que son la referencia a divisiones documentales a las que se aplica la descripción, tanto si son divisiones naturales como el fondo, la sección, la serie y la unidad archivística; y las divisiones artificiales donde están las colecciones.

➤ Fondo documental

Se entiende por fondo documental la agrupación orgánica de documentos generados por una institución o persona física o jurídica en el ejercicio de sus funciones, que constituyen la expresión del conjunto de las actividades desarrolladas. Un fondo puede ser identificado con un archivo; sin embargo, se pueden encontrar varios fondos procedentes de diversas instituciones en un mismo archivo (MORA, 2010b).

➤ Sub-Fondo documental

El sub-fondo documental es una subdivisión del fondo, consistente en un conjunto de documentos relacionados entre sí que se corresponde con las subdivisiones administrativas de la institución u organismo de lo origina, y cuando esto no es posible, con agrupaciones geográficas, cronológicas, funcionales o similares de la propia documentación. Cuando el productor posee una estructura jerárquica compleja, cada sub-fondo tiene a su vez tantos sub-fondos subordinados como sean necesarios para reflejar los niveles de la estructura jerárquica de la unidad administrativa primaria subordinada (MORA, 2010d).

➤ Serie documental

La serie es la división de una sección que corresponde al conjunto de documentos producidos en el desarrollo de una misma atribución general y que versan sobre una materia o asunto específico. Las series pueden estar formadas por documentos singulares o por unidades archivísticas. Pueden ser pequeñas o voluminosas, pero generalmente están formadas por más de una sola unidad (pieza documental o expediente) (MORA, 2010c).

➤ Sub-Serie documental

Una sub-serie es un conjunto de unidades documentales dentro de una serie, agrupadas en razón de modalidades de las actividades que testimonia la serie, en razón de destinatarios o de beneficiarios de dicha actividad. Se puede aceptar como una subdivisión de la serie, cuya vida es argumentada no solo en la inmediatez de recuperación de la información, sino además a causas de propuestas de selección (MORA, 2010e).

➤ Unidad Documental Compuesta/ Expediente

Puede definirse una unidad documental compuesta como la unidad organizada de documentos reunidos por el productor para su cotidiano uso o durante el proceso de organización archivística, pues dichos documentos presentan relación en el tema, actividad o asunto. Es generalmente la unidad básica de la serie y se conoce además como expediente (GONZÁLEZ, 2003a).

➤ Unidad Documental Simple/ Documento

Una unidad documental simple es la unidad más pequeña intelectualmente indivisible en la descripción archivística, puede ser una fotografía, una memoria, un informe, una carta, entre otros. Es la información reflejada en los objetos de la realidad objetiva y la actividad intelectual del hombre. Es utilizada frecuentemente para justificar o acreditar algo. Desde un enfoque jurídico es un testimonio escrito, elaborado de acuerdo con los requisitos establecidos por la ley y que tiene por fin un acto jurídico (GONZÁLEZ, 2003b).

➤ Colección documental

Es el conjunto de documentos acumulados, relacionados por alguna característica común sin tener importancia su origen. Se encuentran con cierta frecuencia en los archivos producto de las donaciones o compras. Respecto al origen no responde a ninguna institución o individuo en particular, sino a la voluntad de alguien. Son documentos cuyo vínculo viene dado gran parte de las veces por el tema o asunto, o bien el autor o coleccionista (MORA, 2010a).

1.3 Normas de descripción archivística

Las normas para la descripción de documentos de archivos son un grupo de reglas establecidas que sirven como patrón para realizar las descripciones de los documentos. El motivo fundamental para el apoyo y adaptación de normas de descripción bibliográfica, según los propios autores, es el hecho de que pueden de esta forma aprovechar las redes de intercambio de información bibliográfica existente.

Las normas para la descripción del material archivístico aseguran una enorme mejora en cuanto a la creación de descripciones coherentes, apropiadas e inteligibles por sí mismas, facilitan la recuperación y el intercambio de información sobre la documentación de archivo, hacen posible la integración de descripciones de diferentes depósitos en un sistema unificado de información y traen beneficios económicos, organizativos y de mejora del servicio. Además, unas normas descriptivas claras son una premisa obligatoria para cualquier proyecto de automatización serio (PEIS y RUIZ, 2004).

ISAD (G)

La norma ISAD (G) (Norma Internacional General de Descripción Archivística) es una normativa internacional para la descripción de documentos. Está basada en los modelos estadounidense, británico y canadiense. El principal aporte viene dado por la norma MAD⁴, que brindó una estandarización de la estructura de datos, es decir, determina cuales son los datos descriptivos (veintiséis elementos) que puede contener una descripción, y los estructura jerárquicamente en las siguientes áreas:

- Área de Mención de Identidad: contiene información mínima descriptiva.
- Área de Contexto: empleada para recoger los datos provenientes del análisis contextual.
- Área de Contenido y Estructura: información relativa al objeto y la organización de la unidad de descripción.
- Área de Condiciones de Acceso y Uso: información relativa a la accesibilidad de la unidad de descripción.
- Área de Documentación Asociada: información relativa a aquellos documentos que tienen una relación significativa con la unidad de descripción.
- Área de Notas: información especial y aquella no incluida en ninguna de las demás áreas.
- Área de Control de la Descripción: información relativa al cómo, cuándo y quién ha elaborado la descripción archivística.

De los veintiséis elementos necesarios para la descripción (datos descriptivos), se consideran de carácter obligatorio seis elementos para la descripción archivística internacional, estos son:

- Código de Referencia.
- Título.
- Productor(es).
- Fecha(s).
- Extensión de la Unidad de Descripción.
- Nivel de Descripción.

La norma ISAD (G) propone determinadas reglas de descripción multinivel. La descripción multinivel consiste en llevar a cabo la descripción de un fondo y todas sus partes componentes (secciones,

⁴ Manual de Descripciones Archivísticas, utilizado en los archivos británicos, para la producción de instrumentos de descripción archivística normalizados.

series, expedientes, documentos y todas las categorías intermediarias), utilizando siempre los elementos apropiados para cada unidad de descripción y relacionando las descripciones resultantes de forma jerárquica (CARNICER y FERNÁNDEZ, 2000).

A cada nivel de clasificación (agrupación) documental le corresponde un nivel de descripción. Las descripciones resultantes, relacionadas jerárquicamente, suponen la representación de la estructura de clasificación mediante la descripción.

La norma ISAD (G) establece un grupo de 5 niveles de descripción que van de lo general a lo específico: nivel de fondo, nivel de sub-fondo, nivel de serie, nivel de expediente y nivel de unidad documental. No obstante, la norma no obliga el nivel de descripción al que se debe llegar.

Para una correcta aplicación de la descripción multinivel, la norma ISAD (G) propone cuatro reglas generales: regla de descripción de lo general a lo particular, regla de información pertinente para el nivel de descripción, regla de vinculación de las descripciones y regla de no repetición de la información. Tras superar varias fases de revisión, la versión actualmente vigente es la norma ISAD (G) 2000 (PEIS y RUIZ, 2004).

La norma ISAD ha supuesto un paso de gigante hacia la necesaria búsqueda de rentabilidad en la explotación de los recursos archivísticos, ya que permite compartir, a nivel internacional, una misma estructuración descriptiva, lo que posibilita el intercambio.

ISAAR (CPF)

La Norma Internacional sobre los Registros de Autoridad de Archivos relativos a Entidades, Personas y Familias (ARCHIVOS, 2003) sirve de guía para elaborar registros de autoridad de archivos que proporcionan descripciones de entidades (instituciones, personas y familias) asociadas a la producción y a la gestión de archivos.

Los registros de autoridad de archivos se pueden utilizar para describir una institución, persona o familia como unidades dentro de un sistema de descripción archivístico y/o controlar la creación y utilización de los puntos de acceso en las descripciones archivísticas, documentar las relaciones entre diferentes productores de documentos entre estas entidades y los documentos creados por ellas y/o otros recursos que les conciernen.

La descripción de los productores de los documentos de archivo es una de las actividades esenciales de los archiveros, tanto si las descripciones se desarrollan en entornos manuales como automatizados. Este proceso requiere una extensa documentación y una actualización continua del contexto de producción y uso de los documentos, especialmente de su procedencia.

Hay una serie de motivos que justifican la utilidad de este tratamiento de la información contextual fuera de la descripción. De esta manera, la información contextual se puede relacionar con las descripciones de documentos de archivo de un mismo productor, que se conservan en más de un archivo, o bien con descripciones relativas a recursos bibliográficos o museográficos relacionados con este productor. Todas estas relaciones pueden facilitar la investigación así como mejorar las prácticas de la gestión documental.

Esta norma está concebida para que los registros de autoridad de archivos se puedan compartir, fomentando la elaboración de descripciones consistentes, adecuadas y auto explicativas de las instituciones, personas y familias que producen los documentos. Está pensada para que se utilice conjuntamente con las normas nacionales existentes o como base para desarrollar normas nacionales. El objetivo primordial de esta norma es sentar las reglas generales para la normalización de las descripciones archivísticas relativas a los productores de documentos de archivo y al contexto de su producción, de forma que permitan:

- El acceso a los archivos y sus documentos basado en una descripción de su contexto de producción que está relacionada con las descripciones de los propios documentos, con frecuencias diversas y físicamente dispersa.
- La comprensión por parte de los usuarios del contexto que subyace en la producción y la utilización de los documentos de archivo que les permita interpretar mejor su significado y trascendencia.
- La identificación precisa de los productores de documentos, incorporando la descripción de las relaciones entre diferentes entidades, especialmente información relativa al cambio administrativo en las instituciones o al cambio de circunstancias personales en individuos y familias.
- El intercambio de estas descripciones entre instituciones, sistemas y redes.

Los elementos de la descripción de un registro de autoridad de archivos están organizados en cuatro áreas de información:

- Área de identificación: en esta área se incluye la información que identifica unívocamente a la entidad que se está describiendo y que define un punto de acceso normalizado.
- Área de descripción: proporciona información pertinente sobre la naturaleza, contexto y actividades de la entidad que se está describiendo.
- Área de relaciones: se consignan y describen las relaciones con otras instituciones, personas y/o familias.

- Área de control: se identifica de forma unívoca el registro de autoridad y se incluye la información necesaria sobre cómo, cuándo y por qué agencia se creó y actualizó el registro de autoridad.

Con la aceptación de la Norma Internacional General de Descripción Archivística ISAD (G) para elaborar descripciones propias e identificar y explicar los documentos en su contexto y contenido con el fin de hacerlos accesibles, y al seguir las recomendaciones que ofrece la Norma Internacional sobre los encabezamientos autorizados archivísticos relativos a entidades, personas y familias ISAAR (CPF), el camino hacia la normalización de las descripciones documentales en archivos comienza a ser más afable.

1.4 Tecnologías

La motivación de la presente investigación parte de la necesidad de reutilizar la información contenida en las herramientas de almacenamiento, utilizadas por las instituciones en las que se pretende instalar los sistemas gestores de documentos de archivo, desarrollados por el departamento. La herramienta mayormente utilizada con este fin es Winisis, software de código privado y acceso gratuito, diseñado por la UNESCO con el objetivo de administrar y crear bases de datos en formato ISIS. Por otro lado, los sistemas desarrollados por el departamento, utilizan para la gestión y almacenamiento de los datos, los sistemas gestores de bases de datos MySQL y PostgreSQL. A continuación se describen las principales características de estos sistemas.

Winisis

Winisis es un sistema de almacenamiento y recuperación de información que permite construir y manejar bases de datos estructuradas no numéricas, es decir, bases de datos constituidas en su mayor parte por texto (BOGLIOLO y BORCHES, 1999).

En términos muy generales se debe pensar de una base de datos en formato ISIS como en un archivo de datos relacionados recogidos por el usuario para satisfacer los requerimientos de información de una determinada comunidad de usuarios. Dicha base puede consistir en un simple fichero de direcciones o un archivo más complejo como puede ser el catálogo de una biblioteca o un directorio de proyectos de investigación. Cada unidad de información almacenada en una base de datos consiste en un conjunto de elementos datos, cada uno de los cuales contienen una característica particular de la entidad a la que describen. Por ejemplo, una base de datos bibliográfica puede contener información

de libros, informes, artículos de revista, entre otros. Cada unidad puede consistir, en este caso, en elementos de datos tales como autor, título, fecha de publicación.

Los elementos datos están almacenados en campos, a cada uno de los cuales se le asigna una etiqueta numérica indicativa de su contenido. Se puede pensar que dicha etiqueta es el nombre del campo con el que es reconocido por CDS/ISIS.

El conjunto de campos que contienen todos los elementos de datos de una determinada unidad de información es llamado registro.

La característica particular de CDS/ISIS es que ha sido diseñado específicamente para manejar campos (y por consiguiente registros) de longitud variable, lo que permite, por una parte una utilización óptima de disco de almacenamiento, y por otra, una completa libertad a la hora de definir la longitud máxima de cada campo.

Un campo puede ser opcional (es decir, puede estar ausente en uno o más registros), puede contener un simple elemento dato, o dos o más elementos datos de longitud variable. En este último caso se dice que el campo contiene sub-campos, cada uno de los cuales se identifica por un delimitador de sub-campo de dos caracteres que precede al correspondiente elemento dato. Además, un campo puede ser repetible, es decir, un determinado registro puede contener más de una instancia u ocurrencia en el campo.

Una definición de base de datos consiste en la definición de los siguientes componentes, cada uno de ellos almacenado en un archivo separado:

- Tabla de Definición de Campos (FDT): define los campos que forman parte de los registros de la base de datos y sus características.
- Hoja(s) de trabajo de entrada de Datos: una o más pantallas definidas por el usuario son utilizadas para crear y/o actualizar los registros principales de la base de datos.
- Formato(s) de visualización: los formatos de visualización definen los requisitos precisos de formateo para cada forma de visualización en pantalla de los registros durante los procesos de búsqueda o para la generación a través de la salida de impresora de productos tales como catálogos o índices.
- Tabla(s) de Selección de Campos (FST): define los campos de la base de datos que entran a formar parte del Archivo Inverso (diccionario) asociado a su base de datos, así como la forma en que su contenido deba ser extraído para facilitar las búsquedas (BOGLIOLO y BORCHES, 1999).

Exportación e importación de registros

La exportación e importación de registros guardados en un archivo en el formato estándar ISO 2709⁵, son útiles para el intercambio de registros, actualización, traspaso a otros formatos, optimización de la base de datos y resguardo de los datos de una base.

En la exportación de los registros son utilizados separadores que permiten delimitar la información almacenada en el archivo con formato ISO 2709.

➤ Separador de Campo

Indica el carácter que se usa como separador de campos en el archivo de salida. Si se deja vacío CDS/ISIS usa el separador de campos estándar definido en la norma ISO 2709, el código ASCII⁶ 30 (1E hexadecimal). El carácter seleccionado debe ser tal que no aparezca nunca como parte de los datos en la base de datos, ya que CDS/ISIS reserva su uso para estos efectos.

➤ Separador de Registro

Indica el carácter que se utiliza como separador de registros en el archivo de salida. Si se deja vacío CDS/ISIS usa el separador de registros estándar definido en la norma ISO 2709 es decir el código ASCII 29 (1D hexadecimal). Las mismas consideraciones mencionadas para el campo anterior son aplicables en este caso.

MySQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multi-hilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB⁷, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, uno de los más usados en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Las principales características de este gestor de bases de datos son las siguientes:

➤ Aprovecha la potencia de sistema multiprocesador gracias a su implementación multi-hilo.

⁵ Estándar ISO (Organización Internacional de Normalización) para la descripción bibliográfica.

⁶ Código Estándar Estadounidense para el Intercambio de Información.

⁷ Compañía de software fundada en 1995.

- Soporta gran cantidad de tipos de datos para las columnas.
- Dispone de API⁸s en gran cantidad de lenguajes (C, C++, Java, PHP, entre otros).
- Gran portabilidad entre sistemas.
- Soporta hasta 32 índices por tabla.
- Gestión de usuarios y contraseñas, manteniendo un muy buen nivel de seguridad en los datos (CONVERSE *et al.*, 2004).

PostgreSQL

PostgreSQL es un potente sistema de bases de datos objeto-relacional de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación por su fiabilidad, integridad de datos y la corrección. Se ejecuta en todos los principales sistemas operativos, incluyendo Linux, y Windows. Posee soporte completo para claves foráneas, uniones, vistas, triggers y procedimientos almacenados (en varios idiomas). También soporta almacenamiento de objetos binarios grandes, incluyendo imágenes, sonidos o vídeo. Cuenta con interfaces nativas de programación C / C + +, Java, NET, Perl, Python, Ruby, Tcl, ODBC⁹, entre otros (POSTGRESQL, 2012).

A continuación se enumeran las principales características de este gestor de bases de datos:

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos, además del soporte para los tipos base, soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, entre otros. También permite la creación de tipos propios.
- Incorpora una estructura de datos array.
- Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, entre otras.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

⁸ Interfaz de programación de aplicaciones.

⁹ Conectividad abierta de bases de datos.

1.5 Migración de datos

Para darle solución a la problemática anteriormente expuesta, donde se expresa la necesidad de utilizar la información contenida en la herramienta de almacenamiento Winisis, utilizada por las instituciones en las que se pretende instalar las aplicaciones desarrollados por el departamento; en la presente investigación se plantea la migración de datos como respuesta a dicha cuestión.

El propósito de la migración de datos es transferir datos existentes al nuevo ambiente. Necesita ser transformado a un formato conveniente para el nuevo sistema, mientras que se preserva la información presente en el viejo.

Muchas circunstancias existen cuando una organización necesita migrar las aplicaciones o las bases de datos. Puede ser que sea tan simple como una mejora a una nueva versión del sistema. O puede ser que implique cambiar a una nueva base de datos o aplicación. Después de una fusión o de una adquisición, a menudo se retiran las aplicaciones redundantes, pero los datos tienen que ser preservados en el sistema de supervivencia (TALEND, 2012).

La migración de datos está compuesta por los procesos expuestos en la siguiente figura.

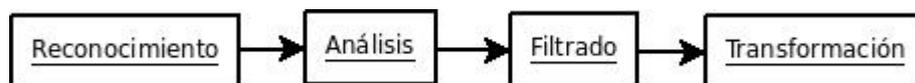


Figura 1: Proceso de migración de datos.

- Reconocimiento: proceso en el cual se identifica la estructura de los datos en la fuente de datos.
- Análisis: proceso donde se define una estructura para almacenar de forma óptima los datos de la fuente, de modo que puedan realizarse los siguientes procesos de migración de forma eficiente, usualmente este proceso se mezcla con el de reconocimiento.
- Filtrado: este es un proceso opcional se realiza solo si se quiere aplicarle un filtro a los datos de alguna manera, ejemplo reducir su contenido tomando en cuenta cierto factor.
- Transformación: en este proceso se realizan las transformaciones requeridas para que los datos tomen el formato deseado.

1.6 Estado del arte de las herramientas para la migración de datos ISIS hacia los SGBD¹⁰ MySQL y PostgreSQL

La presente investigación contribuye a examinar soluciones análogas que permitan migrar datos ISIS hacia los SGBD MySQL y PostgreSQL, con la finalidad de identificar aquellas características que puedan aportar al desarrollo de la investigación.

Para realizar el estudio de estos sistemas, se parte del principio de que la herramienta que se necesita construir debe ser un sistema libre, que permita realizar la migración de datos ISIS hacia los sistemas gestores de bases de datos MySQL y PostgreSQL mediante una aplicación de escritorio, que a su vez sea multiplataforma y que estos datos estén estandarizados bajo normas de descripción internacionales que permitan el intercambio de información entre archivos, como la norma ISAD (G), y que preferentemente sean en idioma español. En la bibliografía consultada se encontraron las siguientes:

Isis2sql

Herramienta desarrollada en Java por Nagy Karoly¹¹, permite exportar datos desde ISIS hacia sentencias SQL. El programa soporta la conversión de MST¹²(con o sin fluorescencia de rayos X) e ISO 2709 y búsquedas básicas así como la edición de registros. Requiere el API JDBC¹³ que es usado para enviar comandos SQL hacia una base de datos relacional y garantiza una integración de SQL hacia Java. Isis2sql fue registrado en SourceForge.net¹⁴ en marzo del 2006 y solo es compatible con las versiones de Sistemas Operativos Windows (FERREYRA, 2006).

Rhombifex

Software diseñado para la exportación de datos desde una base de datos ISIS a MySQL, mediante la creación de script del Lenguaje de Consulta Estructurado o SQL; lo cual no descarta la utilización de estos script almacenados en archivos de textos, en otros gestores de bases de datos como: PostgreSQL y Oracle. La primera versión fue creada en el año 2009 y solamente es disponible para Microsoft Windows, es una aplicación de escritorio y aunque no fue desarrollada con herramientas libres se ofreció como Freeware. En la actualidad existe una versión de prueba, reescrito el código a

¹⁰ Sistema Gestor de Bases de Datos.

¹¹ Químico, matemático y astrónomo.

¹² Archivo de transformación personalizada.

¹³ Java Database Connectivity, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java.

¹⁴ Sitio web de colaboración para proyectos de software comercializado por VA Software.

C++ que utiliza las librerías de Qt¹⁵; pues inicialmente se desarrolló en el lenguaje Object Pascal (TEJEDA, 2011).

Estos sistemas aunque poseen las funcionalidades necesarias para realizar la migración de datos ISIS hacia sistemas gestores de bases de datos en entornos de escritorio, poseen limitantes que no se adecuan a las necesidades por las que se decidió realizar la investigación, estas herramientas están creadas para trabajar sobre determinada plataforma, no cumplen con los principios que se proponen en esta investigación pues solamente trabajan sobre el sistema operativo Windows y los datos migrados en ellas no están estandarizados bajo las normas internacionales de descripción archivística. Realizando una investigación de las características y el alcance logrado por las herramientas que permiten realizar la migración de datos ISIS, se puede concluir que no existe un sistema que satisfaga los requisitos que se espera encontrar en un sistema que implemente la migración de datos ISIS hacia sistemas gestores de bases de datos. Por estas razones se ha decidido crear una herramienta que satisfaga los requisitos planteados en la presente investigación.

1.7 Aplicaciones de Escritorio y Aplicaciones Web

Cada día son más las aplicaciones que se realizan en entornos web. Entre sus ventajas se pueden encontrar que no necesitan de ninguna instalación para que los usuarios accedan a ella, pudiendo utilizar igualmente su navegador favorito que conectan al servidor, ahorrando tiempo en reinstalar, actualizar o cambiar lo que sea en el servidor para acceder a una versión nueva.

Así mismo los problemas de los usuarios son más limitados y el mantenimiento de la aplicación se reduce al servidor. Sin embargo, no se deja de ver las carencias que tienen estas aplicaciones desde el punto de vista de la usabilidad hasta el retardo en cualquier acción del usuario mientras se recarga la página. Pueden presentarse además problemas con JavaScript¹⁶, con las cookies¹⁷ o que se pierda la conexión con el servidor.

Entre las ventajas que tienen las aplicaciones de escritorio se pueden mencionar el hecho de que los usuarios no necesitan acceder a la aplicación desde un lugar diferente a su puesto de trabajo, ya que estas herramientas se instalan directamente en la computadora del usuario, tampoco se necesita disponer de Internet o Intranet para ello, a diferencia de las aplicaciones web.

Es cierto que para las aplicaciones de escritorio se hace necesario instalar en cada equipo el sistema además de hacer trabajoso el problema de las actualizaciones. No obstante, trabajar sobre la web

¹⁵ Biblioteca multiplataforma ampliamente usada para desarrollar aplicaciones con una interfaz gráfica de usuario.

¹⁶ Lenguaje de programación interpretado.

¹⁷ Término informático relacionado con la información que guarda un servidor sobre un usuario en su equipo.

podría hacer el trabajo un poco lento pues se requiere tener en cuenta no solo el entorno de ejecución sino también las premisas de velocidad y eficiencia, el número de usuarios, etcétera.

Cada entorno, no obstante, tiene una serie de ventajas e inconvenientes, sin embargo, se ha decidido realizar esta herramienta como una aplicación de escritorio ya que las mismas ofrecen al usuario el uso de procesos e interfaces sofisticadas. La simplicidad y límites de las aplicaciones web marcan la diferencia entre la interacción con una aplicación web y una de escritorio, además de que desarrolladores web sacrifican la experiencia del usuario por la compatibilidad entre navegadores.

1.8 Lenguaje de programación

Conjunto de reglas semánticas y sintácticas utilizadas para dar instrucciones a un ordenador. Facilita la creación y mantenimiento de programas informáticos. A través del mismo se definen un conjunto de acciones contenidas dentro de algoritmos, las que son ejecutadas en el ordenador con un objetivo específico. Un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo (CORPORATION, MICROSOFT, 2001).

1.8.1 Lenguaje de programación Java

Como parte de la línea de desarrollo, el proyecto de Xcriba al cual pertenece la presente investigación, estableció que para el desarrollo de aplicaciones de escritorio se utilice Java como lenguaje de programación por las características que este lenguaje presenta y que se muestran a continuación:

Java es un lenguaje de programación orientado a objetos que toma muchas de sus sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel como punteros. Se creó con cinco objetivos principales: usar la metodología de la programación orientada a objetos, permitir la ejecución de un mismo programa en múltiples sistemas operativos, incluir por defecto soporte para trabajo en red, diseñarse para ejecutar código en sistemas remotos de forma segura, ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Una de sus características es la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Es un lenguaje creado simplificando algunas cosas de C++ y añadiendo otras, que se utiliza para realizar aplicaciones en Internet.

Este lenguaje es independiente de la plataforma y posee un entorno de ejecución ligero y gratuito. Hoy en día existen multitud de aplicaciones gráficas de usuario basadas en Java. El entorno de ejecución Java se ha convertido en un componente habitual en los ordenadores de usuario de los sistemas

operativos más usados en el mundo. Además, muchas aplicaciones Java lo incluyen dentro del propio paquete de la aplicación, de modo que se pueda ejecutar en cualquier ordenador.

En las primeras versiones de la plataforma Java existían importantes limitaciones en las API de desarrollo gráfico, que ya con la evolución de las mismas, estos problemas fueron decreciendo, logrando el desarrollo de aplicaciones de escritorio complejas y con gran dinamismo, usabilidad y que sea relativamente sencillo (SCHILDT, 2001).

1.9 Herramienta de desarrollo de software

Las herramientas de desarrollo de software (HDS) han desempeñado un importante papel en el desarrollo de aplicaciones. Actualmente se consideran herramientas basadas en computadoras que asisten el ciclo de vida de un software.

El soporte que brindan las HDS al proceso de desarrollo proporciona importantes ventajas para el equipo de trabajo. Permiten automatizar acciones bien definidas, reduciendo la carga cognitiva del ingeniero de software, además establecen métodos efectivos para almacenar y utilizar los datos, y automatizan porciones del análisis y diseño.

NetBeans

NetBeans IDE es un reconocido entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris. El proyecto de NetBeans está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y móviles utilizando la plataforma Java, así como PHP, JavaScript y Ajax, Ruby y Ruby, Groovy y C/C++. El proyecto de NetBeans está apoyado por una comunidad de desarrolladores dinámica y ofrece documentación y recursos de formación exhaustivos, así como una selección diversa de complementos de terceros.

NetBeans 7.0 además de las funcionalidades que proporcionaban sus anteriores versiones añade integración mejorada con el servidor de Oracle WebLogic, permite aprovechar las últimas novedades en la plataforma Java SE. Sigue manteniendo una de sus características que a pesar de ser una extensión de Oracle, es que es Gratuito. Por lo anteriormente mencionado además de ser parte del objeto de estudio del presente trabajo se ha decidido la utilización del IDE NetBeans en su versión 7.0 (CORPORATION, ORACLE, 2012).

1.10 Metodología de Desarrollo de Software

En el mundo de la informática la evolución constante de los procesos de desarrollo ha permitido que se construyan software más potentes y complejos. Esto trajo consigo una mayor exigencia en la calidad y

funcionalidad de los mismos, permitiendo que sean lo menos costoso posible y realizados en el menor período de tiempo. Todo desarrollo de software requiere un conjunto de actividades complejas, las cuales al realizarse sin tener una metodología puede producir insatisfacción en los clientes. Una metodología es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software.

En estos últimos años se han desarrollado dos corrientes en lo referente a los procesos de desarrollo los métodos pesados y los métodos ágiles. Entre las metodologías más reconocidas mundialmente están eXtreme Programming (XP) y Rational Unified Process (RUP) siendo esta última, la metodología planteada por el proyecto Gestión Documental y Archivística para la planificación y desarrollo de software.

Rational Unified Process (RUP)

RUP es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes. Define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. RUP entre sus principales elementos define:

Trabajadores (“quién”) Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos. Actividades (“cómo”) es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos. Artefactos (“qué”) Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables. Flujo de actividades (“Cuándo”) Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable. Está compuesto por cuatro fases y nueve flujos de trabajo.

Las 4 fases en el desarrollo del software son:

1. Inicio: el objetivo en esta etapa es determinar la visión del proyecto.
2. Elaboración: en esta etapa el objetivo es determinar la arquitectura óptima.
3. Construcción: en esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
4. Transición: el objetivo es llegar a obtener una versión del proyecto.

Flujos de trabajo:

- Modelamiento del negocio: describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- Requerimientos: define qué es lo que el sistema debe hacer, para lo cual se identifican las

funcionalidades requeridas y las restricciones que se imponen.

- Análisis y diseño: describe cómo el sistema es realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- Implementación: define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizan y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- Prueba (Testeo): busca los defectos a lo largo del ciclo de vida.
- Instalación: produce versión del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- Administración del proyecto: involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- Administración de configuración y cambios: describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- Ambiente: contiene actividades que describen los procesos y herramientas que soportan el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Entre sus principales características se contemplan:

- Dirigido por casos de uso: los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo, ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
- Centrado en la arquitectura: la arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción.
- Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Es práctico dividir el trabajo en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto (JACOBSON *et al.*, 2000h).

Lenguaje de Modelado UML

UML (en inglés *Unified Modeling Language*) es un lenguaje para el desarrollo de software orientado a objetos, su propósito es visualizar, especificar, construir y documentar proyectos de software.

El éxito de los proyectos de desarrollo de aplicaciones o sistemas se debe a que sirven como enlace entre quien tiene la idea y el desarrollador. El UML es una herramienta que cumple con esta función, ya que le ayuda a capturar la idea de un sistema para comunicarla posteriormente a quien esté involucrado en su proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas.

El UML proporciona una organización en los procesos de diseño de tal forma que los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprendan y convengan con él. UML se apoya para su modelado en las herramientas CASE que permiten automatizar el proceso de diseño y desarrollo de software (LARMAN, 1999b).

1.11 Herramienta CASE para modelar UML

Las herramientas CASE (*Computer Aided Software Engineering*) son utilizadas para automatizar o apoyar una o más fases del proceso de desarrollo de software. Entre las más utilizadas se encuentran Rational Rose y Visual Paradigm, siendo esta última la usada en el desarrollo de software del proyecto de Xcriba.

Visual Paradigm

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

Premiado por Visibilidad en el Entorno de Modelado, Visual Paradigm para UML es un producto galardonado que facilita a las organizaciones la diagramación visual y el diseño de sus proyectos de sistemas, que les brinda la posibilidad de integrar y desplegar sus aplicaciones empresariales de misión crítica y de sus bases de datos subyacentes. Esta herramienta ayuda a los equipos de desarrollo de software a destacar todo el modelo de acumulación de trabajo y desplegar el proceso de desarrollo de software, lo que permite maximizar y acelerar tanto las contribuciones individuales como las de equipo

Además de Visual Paradigm ser multiplataforma, proporciona el código y compatibilidad hasta con 10 lenguajes y posee una alta interoperabilidad ya que introdujo la importación y exportación de modelos de proyecto desde o hasta un formato XML¹⁸ (PARADING, 2012).

1.12 Conclusiones parciales

Luego de realizar un estudio de las aplicaciones encontradas en la bibliografía consultada que realizan la migración de datos ISIS, se descarta su uso por no satisfacer las necesidades que se requieren.

Una vez analizadas las metodologías y tecnologías más referenciadas en la bibliografía consultada, sus ventajas y desventajas, se toma como punto de partida la elección del lenguaje Java para el desarrollo de la posible solución. Además, se utiliza la metodología RUP según lo establece el Programa de mejora que la UCI está llevando a cabo para alcanzar una certificación internacional del nivel 2 de CMMI¹⁹, con el objetivo de lograr productos y servicios de alta calidad y aumentar la productividad. Como herramienta CASE para el modelado de los distintos diagramas definidos por la metodología, se seleccionó Visual Paradigm, por ser multiplataforma, fácil de usar, instalar y actualizar. Esta combinación de metodología y tecnologías permite dar respuesta a la problemática planteada.

¹⁸ Lenguaje de marcado extensible.

¹⁹ Integración de Modelo de Madurez de Capacidades.

Capítulo 2 Características de la herramienta para la migración de datos hacia la norma ISAD (G)

2.1 Introducción

En este capítulo se exponen las características del sistema y los procesos que son objeto de automatización. Se realiza la descripción de las actividades metodológicas orientadas por RUP para la construcción de la solución propuesta, confeccionando el modelo de dominio correspondiente, para facilitar la mejor comprensión del sistema y su contexto. Se traducen las necesidades que debe suplir la solución a requisitos funcionales, a partir de los cuales se formula la lista de casos de uso del sistema. Se define la lista de requisitos no funcionales, importantes también en la calidad final del producto.

2.2 Modelo de dominio

El modelo de dominio es una de las alternativas que brinda RUP para la identificación de requisitos y la comprensión del contexto cuando existe poca estructuración en los procesos de negocio, y con la que se muestra al usuario, de manera visual, los principales conceptos que se manejan en el dominio del negocio, sus partes y sus relaciones. En el caso de la presente investigación se decide utilizar un modelo de dominio, ya que el proceso que pretende automatizar (la migración de datos), carece de una estructura sólida, en la que se puedan observar actividades del negocio concretas. El proceso de migración de datos en la mayoría de las instituciones no se realizaba, a las aplicaciones desplegadas se le insertaban nuevamente los datos sin reutilizar la información anteriormente digitalizada.

Esto permite a todos los que de alguna manera están involucrados en el proceso de desarrollo del producto, manejar un vocabulario común que posibilite el entendimiento del contexto en que se sitúa el sistema. Este modelo se realiza a través de un diagrama de clases de UML simplificado, en el cual se representan las clases conceptuales que pueden intervenir en el sistema y sus asociaciones preliminares, así como los objetos más importantes en el mismo. Estos objetos del dominio representan “cosas” que existen o los eventos que acontecen en el medio en el que se desenvuelve la aplicación (JACOBSON *et al.*, 2000f).

Diagrama del modelo de dominio

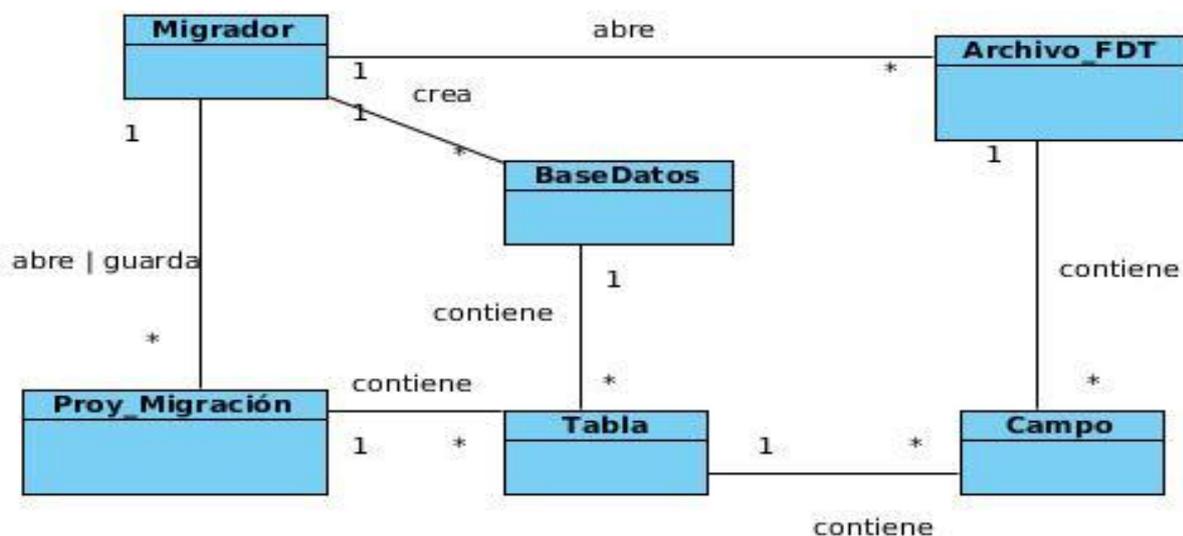


Figura 2: Modelo de dominio de la herramienta para la migración de datos hacia la norma ISAD (G)

A continuación se proporciona un marco conceptual donde se identifican los conceptos fundamentales dentro del dominio, que permiten una mejor comprensión del diagrama del modelo del dominio.

- Migrador: personal encargado de realizar la migración de datos hacia la norma ISAD (G).
- Archivo FDT: archivo donde se encuentran almacenados los campos de la estructura de almacenamiento generada del Winisis.
- Base de datos: archivo que se crea en el servidor con los datos del archivo FDT cargados.
- Tabla: tabla de la base de datos que se crea con los campos de la norma ISAD (G) que tienen asignados los campos del archivo FDT generado del Winisis.
- Campo: campo de la tabla de la base de datos que corresponde a cada uno de los campos generados por el Winisis en el archivo FDT.
- Proyecto de migración: archivo que guarda los datos de un proyecto, tales como los campos cargados a partir del archivo FDT, las asignaciones que han sido realizadas y los campos adicionales creados.

2.3 Especificación de requisitos de software

En el proceso de producción de software es fundamental tener bien claros los requisitos del sistema, con el objetivo de tener definida una línea de lo que se desea desarrollar. La captura de estas especificaciones permite y asegura un sistema con eficiencia y calidad. Estas se pueden obtener utilizando la descripción de las condiciones y/o capacidades que el producto debe cumplir, debiendo

ser lo suficientemente abarcadoras como para que se puedan llegar a acuerdos entre los clientes y los desarrolladores en cuanto a lo que debe o no hacer el software (TURNER, 2005).

2.3.1 Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que un sistema determinado debe cumplir (CÓDOVA, 2007b). Seguidamente se enumeran los que se han capturado para el desarrollo de esta investigación:

[RF1] Abrir el archivo FDT generado del Winisis: permite abrir el archivo FDT y muestra los campos del mismo para ser vinculados con los campos de la norma ISAD (G), ISAAR (CPF) o los campos adicionales.

[RF2] Asignar campos del FDT a los campos de la norma ISAD (G): permite vincular los campos del FDT con los campos de la norma ISAD (G).

[RF3] Asignar campos del FDT a los campos de la norma ISAAR (CPF): permite vincular los campos del FDT con los campos de la norma ISAAR (CPF).

[RF4] Asignar campos del FDT a los campos adicionales: permite vincular los campos del FDT con los campos adicionales.

[RF5] Mostrar los campos asignados a un campo de la ISAD (G) determinado: dado un campo de la ISAD (G) permite mostrar los campos que le han sido asignados.

[RF6] Mostrar los campos asignados a un campo de la ISAAR (CPF) determinado: dado un campo de la ISAAR (CPF) permite mostrar los campos que le han sido asignados.

[RF7] Mostrar los campos asignados a un campo adicional determinado: dado un campo adicional permite mostrar los campos que le han sido asignados.

[RF8] Eliminar campos asignados a un campo de la ISAD (G) determinado: dado un campo de la ISAD (G) permite eliminar los campos que le han sido asignados.

[RF9] Eliminar campos asignados a un campo de la ISAAR (CPF) determinado: dado un campo de la ISAAR (CPF) permite eliminar los campos que le han sido asignados.

[RF10] Eliminar campos asignados a un campo adicional determinado: dado un campo adicional permite eliminar los campos que le han sido asignados.

[RF11] Crear campo adicional: permite al usuario crear un campo que no esté comprendido dentro de las normas internacionales mencionadas y que la institución crea necesario para enriquecer su

descripción.

[RF12] Eliminar campo adicional: permite eliminar los campos adicionales que han sido creados.

[RF13] Crear la base de datos a partir del archivo FDT cargado: permite crear una base de datos en el servidor según los campos asociados a los campos de la norma ISAD (G).

[RF14] Agregar datos a la base de datos: permite agregar datos a la base de datos en el servidor según el archivo con formato ISO 2709, generado del Winisis.

[RF15] Guardar proyecto de migración: permite guardar los datos de un proyecto de migración. En el marco del problema planteado se le llama proyecto de migración al conjunto de funcionalidades explicadas anteriormente y que hayan sido realizadas parcial o totalmente, guardando en un archivo los datos del proyecto, tales como los campos cargados a partir del archivo FDT, las asignaciones que han sido realizadas y los campos adicionales creados.

[RF16] Abrir proyecto de migración: permite abrir un proyecto de migración a partir del archivo donde hayan sido guardados los datos del mismo.

2.3.2 Requisitos no funcionales

Los requisitos no funcionales detallan las propiedades o cualidades que el producto debe tener, añadiéndole funcionalidad al sistema, pues hacen al producto atractivo, fácil de usar, rápido y confiable los cuales se encuentran separados por categorías (CÓDOVA, 2007a):

Usabilidad

- La herramienta debe ser sencilla de usar, resultando más factible si se imparten cursos de capacitación para una rápida y mejor comprensión de los usuarios logrando un mejor aprovechamiento de la aplicación.
- Podrá ser empleada por usuarios con conocimientos mínimos para su uso.
- Se utilizará el idioma español para los mensajes y textos de la interfaz.

Soporte

Una vez concluida la aplicación, se realizan varias pruebas funcionales para comprobar si satisface los requisitos planteados. Con la puesta en práctica de la aplicación deben quedar satisfechas las necesidades de los usuarios, mediante actualizaciones y mejoras a la misma.

Características del Sistema

- Servidores de Base de Datos: PostgreSQL 9.1 y MySQL 5.1.
- Se utilizará Java como lenguaje de programación.
- Sistema operativo Linux.

Restricciones del diseño

- El sistema será desarrollado utilizando como lenguaje de programación Java, para el diseño de la base de datos se utilizó MySQL y PostgreSQL.
- El sistema contará con las librerías mysql-connector-java-5.0.4-bin.jar, postgresql-9.1-901.jdbc3.jar y javahelp-2.0.02.jar de java.
- El entorno de desarrollo integrado será NetBeans 7.0.
- Como metodología de desarrollo de software RUP, usando el lenguaje de modelado UML.

Requisitos para la documentación de usuarios en línea y ayuda del sistema

El sistema contará con una ayuda para el usuario con lo que podrá aprender rápidamente a utilizar la aplicación.

Interfaz

El diseño de la interfaz debe ser sencillo y fácil de usar, con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus acciones. Además, tiene que ser formal, seria y con una navegación sugerente, en correspondencia con el fin que se desarrolla la aplicación, por lo que es necesario mantener una coherencia y estilo común entre todas las vistas del sistema.

Requisitos de Licencia

- Todas las herramientas utilizadas deben responder a licencias de software libre.
- Se patentará bajo la licencia GNU/GPL.
- El uso del producto no debe incurrir en el quebrantamiento de ninguna ley.

Requisitos Legales, de Derecho de Autor y otros

El producto debe responder a las normas político-culturales que existen dentro de la universidad, que estas además responden a las normas del país.

2.4 Propuesta de solución

Se pretende desarrollar una herramienta que permita la migración de datos hacia los sistemas gestores de bases de datos MySQL y PostgreSQL bajo la norma ISAD (G). Se implementa una aplicación de escritorio, lo que permite a los usuarios utilizarla independientemente de que posean o no conexión a una red. Se propone para el desarrollo de dicha aplicación, el lenguaje de programación Java, garantizando su uso en múltiples plataformas.

La herramienta una vez desarrollada debe permitir al usuario interactuar directamente con los datos ISIS contenidos en el archivo con formato FDT, exportado de la herramienta de almacenamiento Winisis y enlazarlos con los elementos de la norma ISAD (G), para luego crear la base de datos a partir de ellos y agregarle los datos provenientes del archivo con formato ISO 2709, generado también del Winisis.

Estructura de la base de datos

Debido a que la herramienta de almacenamiento Winisis tiene definida su propia estructura de datos, la cual presenta inconsistencias con la estructura utilizada por los sistemas gestores de bases de datos hacia los cuales se desea migrar la información, se hizo necesario modelar los datos logrando una relación entre ambas estructuras. Como resultado de dicha relación, la estructura de datos quedó dispuesta de la siguiente manera:

- Tabla principal: esta tabla es nombrada “descripción” y contiene los elementos de la norma ISAD (G) que tienen un único campo asociado.
- Tabla secundaria: esta tabla es nombrada “productor” y contiene los elementos de la norma ISAAR (CPF) que tienen un único campo asociado.
- Tabla secundaria: esta tabla es nombrada “adicional” y contiene los campos adicionales creados que tienen un único campo asociado.
- Tabla de campos repetibles: para cada campo repetible (multievaluado), se crea una tabla con su id correspondiente y se genera otra que corresponde a la relación entre esta y la tabla principal.
- Tabla de campos compuestos: corresponde a los elementos que tienen más de un campo asociado, esta tabla tiene además el id de la tabla principal como llave foránea.
- Tabla de campos compuestos repetibles: corresponde a los elementos que tienen más de un campo asociado y a su vez estos son repetibles, para cada elemento se crea una tabla con su id correspondiente y se genera otra que es la relación entre esta y la tabla principal.

Los elementos que no tienen campos asociados no se toman en cuenta en el proceso de migración.

En dependencia de los campos que contenga el archivo FDT cargado y las asignaciones que han sido realizadas, se crea la base de datos estructurándola según los elementos a los cuales le han sido asignados estos campos y las propiedades de los mismos. A continuación se muestra un ejemplo de cómo quedó estructurada la base de datos a través de las tablas creadas, según las especificaciones anteriormente mencionadas:

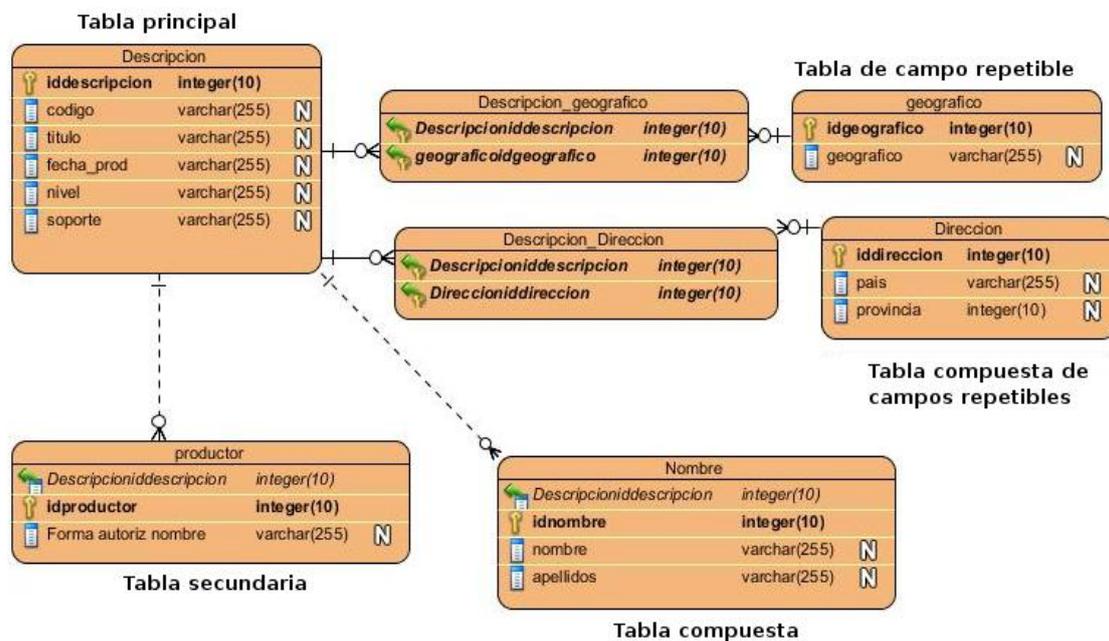


Figura 3: Ejemplo de estructura de datos

2.5 Definición de actores y casos de uso

Definición de actores del sistema

Actor	Objetivo
Migrador	Realizar la migración de datos del Software Winisis hacia la norma ISAD (G) en los sistemas gestores de bases de datos MySQL y PostgreSQL.

Tabla 1: Definición de actores del sistema

Casos de uso del sistema

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema. A continuación se presentan los casos de usos correspondientes a los requisitos descritos en el acápite anterior.

Diagrama de casos de uso del sistema

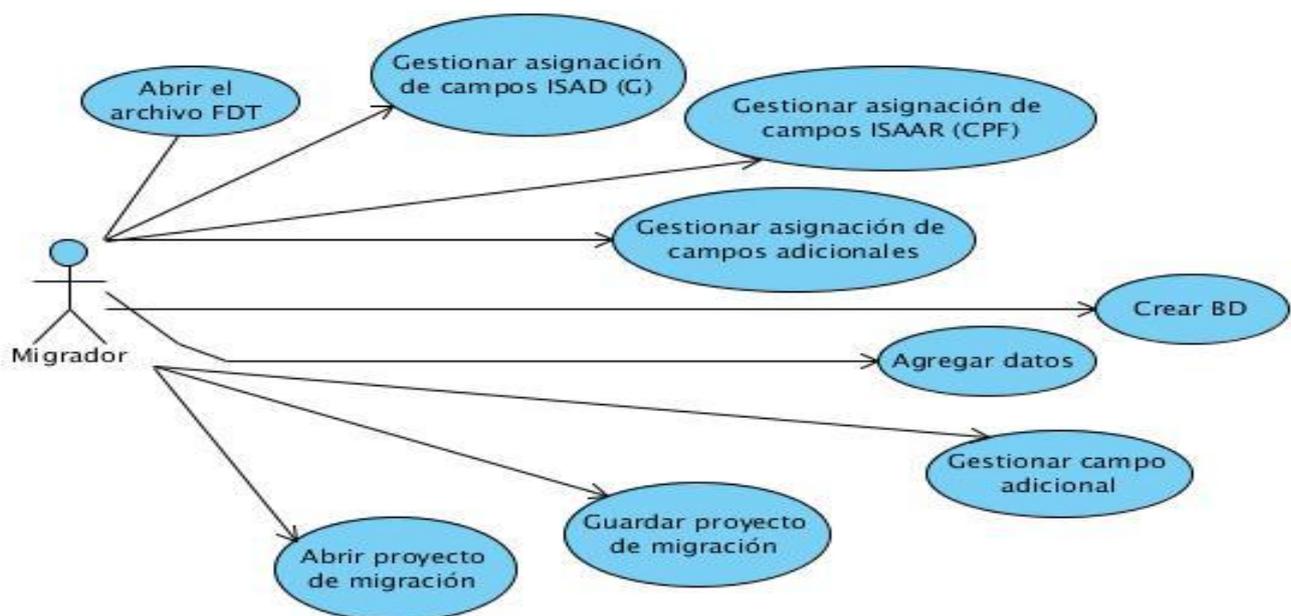


Figura 4: Diagrama de casos de uso de la herramienta para la migración de datos hacia la norma ISAD (G)

2.5.1 Patrones de caso de uso utilizados

Los patrones de casos de uso son utilizados generalmente como plantillas que describen como deberían ser estructurados y organizados los casos de uso. Son patrones que capturan mejores prácticas para modelar casos de uso. En la presente investigación el patrón utilizado fue:

CRUD: este patrón se utiliza en los casos donde se quiere crear, eliminar, hacer cambios y consultas a alguna entidad del sistema. Su nombre es un acrónimo de las palabras en inglés: Create, Read, Update, Delete (Crear, Leer, Actualizar y Eliminar). La variación denominada CRUD Parcial, indica que en caso de que solo algunas de las cuatro operaciones sean simples mientras que otras son complejas, se puede agrupar las operaciones simples en un caso de uso y dejar las otras modeladas

como un caso de uso separado. La utilización de este patrón se puede observar a través de los casos de uso:

- Gestionar asignación de campos ISAD (G), Gestionar asignación de campos ISAAR (CPF), Gestionar asignación de campos adicionales: encierran las acciones crear, mostrar y actualizar.
- Gestionar campos adicionales: encierra las acciones crear y eliminar.

2.5.2 Descripciones textuales de los casos de uso del sistema

El objetivo principal de detallar cada caso de uso es describir su flujo de procesos en detalle, incluyendo cómo comienza, termina e interactúan con los actores. Tras la valoración realizada a los casos de uso según criterios de complejidad y prioridad definidos por la metodología utilizada, se lograron identificar los casos de uso arquitectónicamente significativos, los cuales se muestran a continuación, los restantes pueden ser consultados en el [Anexo 1](#).

Objetivo	Abrir un archivo FDT generado del Winisis.	
Actores	Migrador	
Resumen	El actor selecciona la opción abrir archivo FDT, el sistema muestra una interfaz para seleccionar el archivo a abrir, se abre el fichero y se muestran los campos del mismo.	
Complejidad	Media	
Prioridad	Crítico	
Precondiciones	Debe abrirse un fichero que tenga la extensión FDT y que haya sido generado correctamente del Winisis.	
Postcondiciones	Se muestran los campos del FDT.	
Flujo de eventos		
Flujo básico <Abrir archivo FDT>		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona la opción de abrir FDT en el Menú Archivo.	El sistema le muestra una ventana para seleccionar el archivo a abrir.

2.	El usuario selecciona el archivo a abrir.	El sistema muestra una lista con los campos cargados del archivo finalizando así el caso de uso.
Flujos alternos		
Nº Evento <2a El archivo seleccionado no es válido>		
	Actor	Sistema
1.		El sistema muestra el siguiente mensaje de error: "El fichero seleccionado no es válido".
Prototipo de Interfaz de usuario: Ver Anexo 2		

Tabla 2: Descripción del caso de uso Abrir archivo FDT.

Objetivo	Gestionar la asignación de los campos del archivo FDT generado del Winisis a los campos de la norma ISAD (G).	
Actores	Migrador	
Resumen	El usuario asigna, elimina o muestra la asignación de los campos del archivo FDT a los campos de la norma ISAD (G).	
Complejidad	Media	
Prioridad	Crítico	
Precondiciones	Debe haber abierto el archivo FDT.	
Postcondiciones	Se permite crear la base de datos.	
Flujo de eventos		
Flujo básico <Gestionar asignación de campos ISAD (G)>		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor abre el archivo FDT.	El sistema le muestra una lista con los campos del archivo FDT, una lista de selección con los campos de la norma ISAD (G) y permite realizar varias

	acciones: <ul style="list-style-type: none"> ➤ Asignar un campo del FDT a un campo de la norma ISAD (G). Ver Sección 1: Asignar campos. ➤ Mostrar los campos asignados a un campo de la norma ISAD (G). Ver Sección 2: “Mostrar campos asignados” ➤ Eliminar campos asignados a un campo de la norma ISAD (G). Ver Sección 3: “Eliminar campos asignados.”
--	---

Sección 1: “Asignar campos”

Flujo básico Asignar campos

	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona uno a uno los campos del FDT o marca la opción de seleccionar todos para ser asignados y el campo de la norma ISAD (G) al que serán asignados y presiona el botón >>.	El sistema muestra los elementos en la lista de campos asignados asociados al campo de la norma ISAD (G) que fue seleccionado y los elimina de la lista de campos del FDT, finalizando así el caso de uso.

Flujos alternos

Nº Evento <1a Datos vacíos>

	Actor	Sistema
1.		El sistema muestra el siguiente mensaje de error: "Debe seleccionar al menos un campo a asignar."

Sección 2: “Mostrar campos asignados”

Flujo básico Mostrar campos asignados

	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona el campo de la norma ISAD (G).	El sistema muestra los elementos en la lista de campos asignados asociados al

		campo de la norma ISAD (G) que fue seleccionado, finalizando así el caso de uso.
Sección 3: “Eliminar campos asignados”		
Flujo básico Eliminar campos asignados		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona el campo de la norma ISAD (G).	
2.	Selecciona los elementos de la lista de campos asignados a ser eliminados marcándolos uno a uno o en la opción de seleccionar todos y presiona el botón <<.	El sistema muestra un mensaje de confirmación.
3.	El usuario acepta confirmando que desea eliminar.	El sistema elimina los elementos de la lista de campos asignados y los muestra en la lista de campos del FDT, finalizando así el caso de uso.
Flujos alternos		
Nº Evento <1a Cancelar eliminación>		
	Actor	Sistema
1.		El sistema no realiza ninguna acción.
Nº Evento <2a Lista vacía>		
1.		El sistema muestra el siguiente mensaje de error: "No existen campos a eliminar."
Nº Evento <2b Datos vacíos>		
2.		El sistema muestra el siguiente mensaje de error: "Debe seleccionar al menos un campo a eliminar."
Prototipo de Interfaz de usuario: Ver Anexo 3		

Tabla 3: Descripción del caso de uso Gestionar asignación de campos ISAD (G)

Objetivo	Crear base de datos en el servidor.	
Actores	Migrador	
Resumen	El usuario selecciona la opción crear base de datos, el sistema especifica algunos datos y se crea la base de datos.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	En la computadora donde se creará la base de datos debe estar instalado el servidor de bases de datos MySQL y PostgreSQL. Se deben haber gestionado la asignación de campos.	
Postcondiciones	Crea la base de datos en el servidor.	
Flujo de eventos		
Flujo básico <Crear base de datos>		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona la opción crear base de datos.	El sistema le muestra una ventana para especificar algunos datos para la creación de la base de datos como nombre de la base de datos, servidor, puerto y en que gestor se creará.
2.	El usuario especifica los datos.	El sistema crea la base de datos en el servidor finalizando así el caso de uso.
Flujos alternos		
Nº Evento <2a Datos vacíos>		
	Actor	Sistema
1.		El sistema muestra el siguiente mensaje de error: "Los datos no deben estar en blanco."
Nº Evento <2b Datos incorrectos>		
	Actor	Sistema
1.		El sistema muestra el siguiente mensaje

		de error: "No se pudo conectar al servidor.
Nº Evento <2c Nombre de base de datos existente>		
	Actor	Sistema
1.		El sistema muestra el siguiente mensaje de error: "Ya existe una base de datos con ese nombre, debe renombrarla."
Prototipo de Interfaz de usuario: Ver Anexo 4		

Tabla 4: Descripción del caso de uso Crear BD

Objetivo	Agregar datos a la base de datos.	
Actores	Migrador	
Resumen	El usuario selecciona la opción agregar datos, el sistema muestra una interfaz para especificar algunos datos y se agregan a las tablas de la base de datos.	
Complejidad	Alta	
Prioridad	Crítico	
Precondiciones	Se debe haber creado la base de datos.	
Postcondiciones	Se agregan los datos a la base de datos.	
Flujo de eventos		
Flujo básico <Agregar datos>		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona la opción de agregar datos en el menú Migración	El sistema le muestra una ventana para especificar el separador de registro, el separador de campos y el separador de sub-campos utilizados por el Winisis en la generación del archivo con formato .ISO 2709.
2.	El usuario especifica los datos y selecciona	El sistema agrega los datos a la base de

	la opción Aceptar.	datos en el servidor finalizando así el caso de uso.
Flujos alternos		
Nº Evento <2a El archivo seleccionado no es válido>		
	Actor	Sistema
1.		El sistema muestra el siguiente mensaje de error: "El fichero seleccionado no es válido".
Nº Evento <2b Datos vacíos>		
	Actor	Sistema
1.		El sistema muestra el siguiente mensaje de error: "Los datos no deben estar en blanco.".
Nº Evento <2c Datos incorrectos>		
	Actor	Sistema
1.		El sistema muestra el siguiente mensaje de error: "Los datos especificados son incorrectos".
Prototipo de Interfaz de usuario: Ver Anexo 5		

Tabla 5: Descripción del caso de uso Agregar datos

2.6 Conclusiones parciales

Con la realización de este capítulo se desarrolló un análisis crítico de los procesos involucrados en el campo de acción. Se planteó la estructura de datos de la propuesta de solución, definiendo el formato que tendrá la base de datos migrada. Se define la propuesta del sistema presentando un listado de requisitos funcionales y no funcionales donde se recogen las principales necesidades del sistema a desarrollar. Estas necesidades fueron traducidas a un conjunto de casos de usos que representan las principales funcionalidades del sistema. Se realizaron las descripciones detalladas de cada uno de estos casos de uso donde se definen los prototipos de interfaz de usuario correspondientes a cada caso de uso.

Capítulo 3 Análisis y diseño del sistema

3.1 Introducción

En este capítulo se desarrollan los procesos comprendidos en el análisis y el diseño del sistema, con el objetivo de entender las especificaciones de requisitos y casos de usos. Como parte de los procesos de análisis y diseño se confeccionan y refinan un conjunto de artefactos que son parte de la guía a una mejor implementación, entre estos artefactos se destacan los diagramas de clases del análisis y los diagramas de clases del diseño.

3.2 Análisis del sistema

El análisis es desarrollado en la fase de elaboración. Consiste en obtener una visión del sistema que se preocupa por ver qué hace el mismo, de modo que solo se interesa por los requisitos funcionales. Representa las cosas del mundo real y no de la implementación automatizada de las mismas (JACOBSON *et al.*, 2000b).

Diagrama de clases del análisis

Los diagramas de clases de análisis representan un modelo conceptual temprano que describe las características y comportamiento comunes de un conjunto de elementos que existen en el sistema. Se expresa que es conceptual pues pospone todos los elementos de diseño ya que no considera posibles tecnologías a emplear en el desarrollo del software; constituyen un prototipo de las futuras clases que dan vida al mismo. Las clases del análisis están siempre identificadas con uno de los tres estereotipos existentes, los cuales son:

Interfaz: se encargan de la modelación de toda la interacción que puede existir entre los actores y el sistema (CI_<Nombre de la clase>).

Control: representan la coordinación, secuenciación, transacciones y a veces la lógica del negocio; se emplean a menudo para encapsular el control referido a un caso de uso (CC_<Nombre de la clase>).

Entidad: representa la información de larga duración y a menudo persistente que se maneja en el sistema (CE_<Nombre de la clase>) (JACOBSON *et al.*, 2000b).

Diagrama de interacción

Los diagramas de interacción representan una vista dinámica del sistema y se pueden clasificar en dos tipos, diagramas de colaboración o diagramas de secuencia. Un diagrama de interacción representa la secuencia de acciones que ocurren desde que el actor comienza el caso de uso, así como los

mensajes que se envían entre cada una de las clases. En el análisis se utilizan los diagramas de colaboración, pues el objetivo principal es identificar las funcionalidades de cada objeto y las responsabilidades sobre ellos (JACOBSON *et al.*, 2000c).

A continuación se presenta los diagramas de colaboración a partir de los casos de uso descritos en el acápite anterior obtenidos en esta investigación. Los restantes se pueden ver en el [Anexo 6](#).

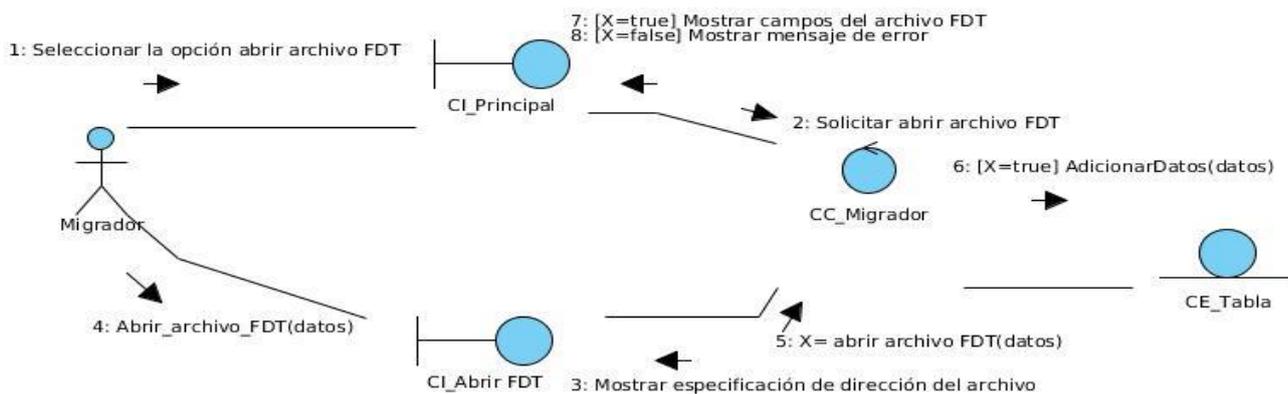


Figura 5: Diagrama de interacción CU: Abrir el archivo FDT

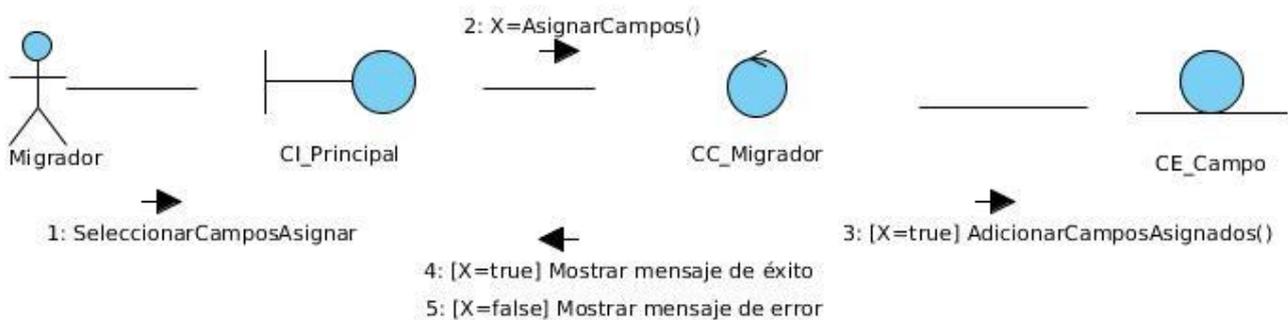


Figura 6: Diagrama de interacción CU: Gestionar asignación de campos ISAD (G) –SC1 Asignar campos ISAD (G)

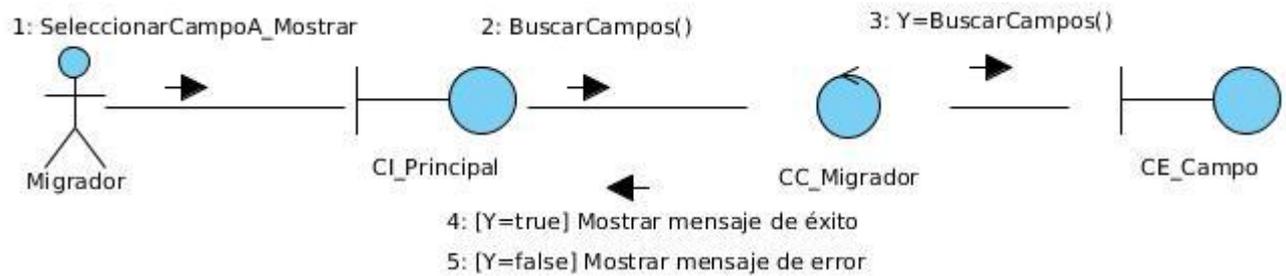


Figura 7: Diagrama de interacción CU: Gestionar asignación de campos ISAD (G) –SC2 Mostrar asignación de campos ISAD (G)

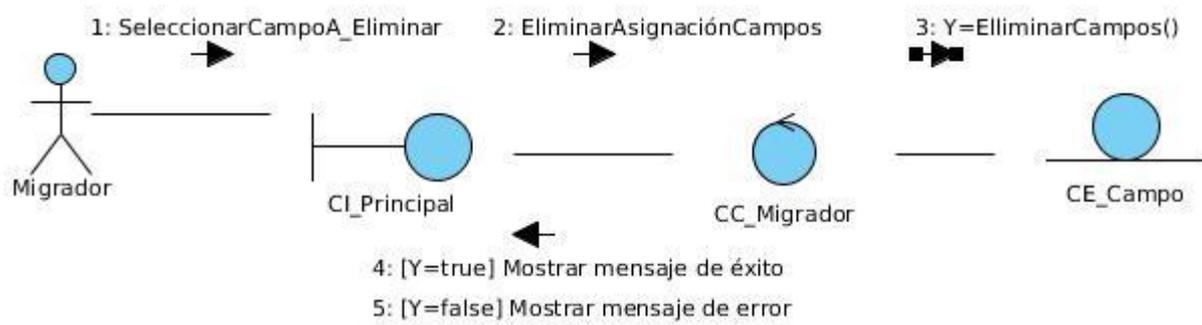


Figura 8: Diagrama de interacción CU: Gestionar asignación de campos ISAD (G) –SC3 Eliminar asignación de campos ISAD (G)

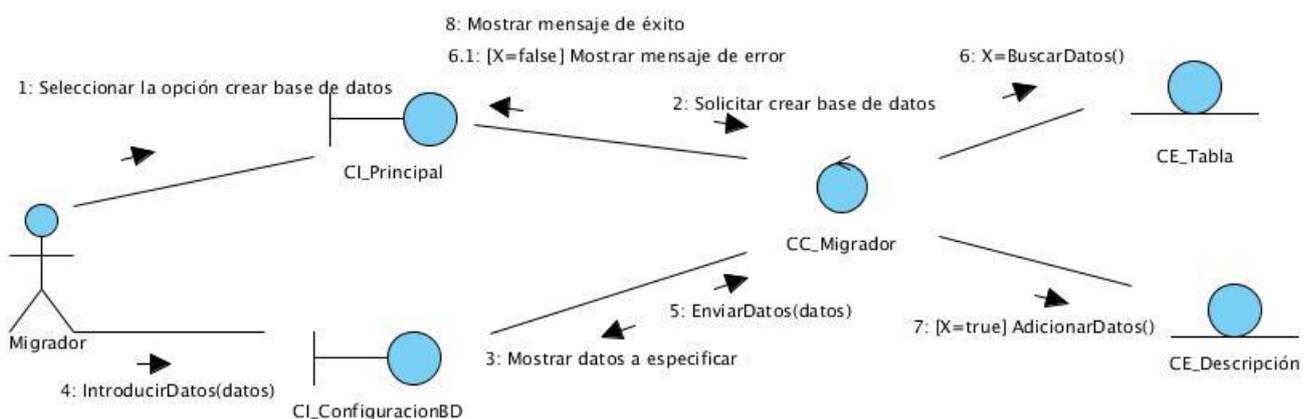


Figura 9: Diagrama de interacción CU: Crear BD

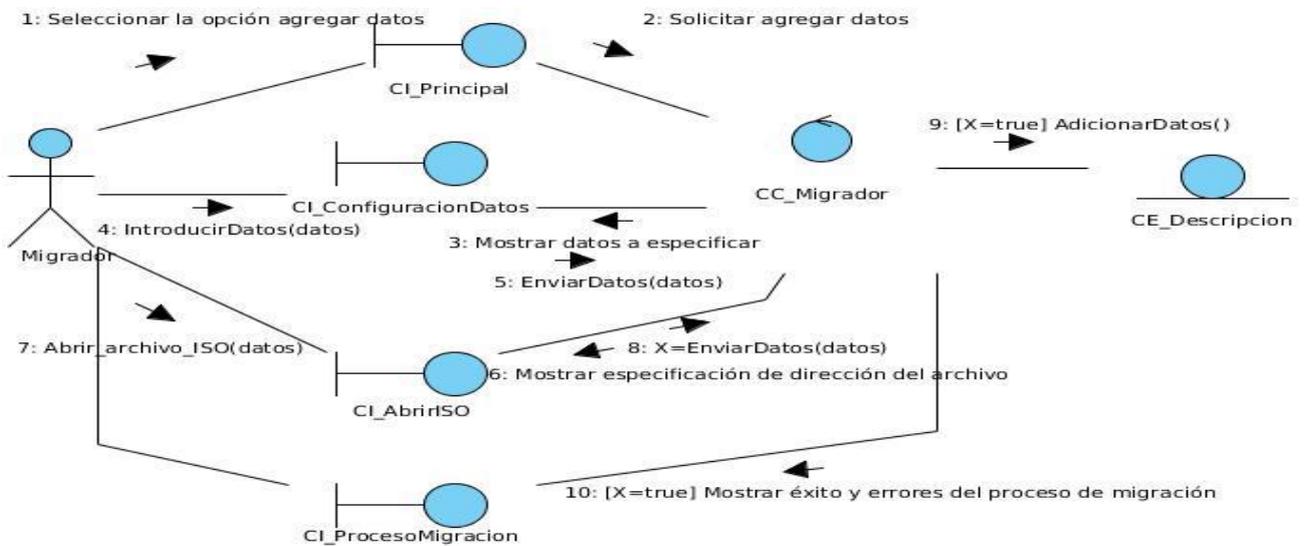


Figura 10: Diagrama de interacción CU: Agregar datos

3.3 Diseño del sistema

El diseño se realiza al final de la fase de elaboración después del análisis. Este es un refinamiento del análisis, es una representación más concreta del diagrama de clases del análisis, ya que durante el diseño se tienen en cuenta los requisitos no funcionales, responde además al cumplimiento de los objetivos planteados para el sistema. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

3.4 Arquitectura del sistema

La arquitectura de software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes desde la visión del resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del mismo. La arquitectura de un sistema consiste en la vista conceptual de toda su estructura (JACOBSON *et al.*, 2000a).

3.4.1 Patrones de arquitectura

Los patrones de arquitectura son los que definen la estructura de un sistema software, los cuales a su vez se componen de subsistemas con sus responsabilidades, también tienen una serie de directivas

para organizar los componentes del mismo sistema, con el objetivo de facilitar la tarea del diseño de este (JACOBSON *et al.*, 2000g). A continuación se muestra el patrón arquitectónico utilizado.

Arquitectura en capas

La arquitectura en capas es un estilo arquitectónico de llamada y retorno, en el mismo cada capa proporciona servicios a la capa superior y se sirve de las prestaciones que le brinda la inferior, al dividir un sistema en capas, cada capa puede tratarse de forma independiente, sin tener que conocer los detalles de las demás. La división de un sistema en capas facilita el diseño modular, en la que cada capa contiene un aspecto concreto del sistema y permite además la construcción de sistemas débilmente acoplados, lo que significa que si se minimiza las dependencias entre capas resulta más fácil sustituir la implementación de una capa sin afectar el resto del sistema (MARQUINA, 2008).

El patrón de arquitectura en 3 capas es básicamente una descripción de los subsistemas y componentes, y las relaciones entre ellos. El mismo determina la organización estructural del sistema software en tres capas fundamentales:

- Capa de presentación: es la que interactúa directamente con el usuario, captura la información entrada por este y hace las peticiones a la capa inferior mostrando al usuario la respuesta proveniente de esta. En la solución propuesta representa las vistas del sistema. Únicamente se comunica con la capa de lógica del negocio.
- Capa de lógica del negocio: en esta capa se reciben las peticiones del usuario, y tras ejecutar una acción se le envían las respuestas del proceso. Desde el punto de vista del diseño, esta capa es contenedora de las clases entidades y controladoras. Se comunica con la capa de acceso a datos y brinda información, como se ha dicho, a la capa de presentación, la cual le envía las peticiones y esta le responde con los resultados.
- Capa de acceso a datos: es donde se accede a los datos. Se hace referencia a uno o más gestores de bases de datos que realizan el almacenamiento, modificación y consulta de los datos. Recibe peticiones desde la capa de negocios y realiza todas las operaciones de forma transparente para esta otra.

Beneficios e inconvenientes de uso:

- Facilita la descomposición de la aplicación en varios niveles de abstracción.
- Facilita la evolución del sistema, ya que los cambios sólo deben de afectar a la capa donde se encuentre la modificación.
- Si la interfaz accede a la misma función, no se repite código. Lo que posibilita una mayor

facilidad de mantenimiento de la aplicación.

- Si se añade un nuevo formulario no ocasionara verdaderos quebraderos de cabeza, ya que los formularios no dependen unos de otros, con lo que el cambiar el flujo de trabajo es algo trivial.
- Los formularios no acceden de forma independiente a los datos, al modificar los datos. Esto implica que no se mostrará diferencia alguna.

Y como inconvenientes se encuentran:

- No todo sistema puede ser estructurado en capas.
- Y aun pudiendo ser estructurado en capas, la separación entre una y otra no es trivial. No sólo porque para un desarrollador no lo es, sino también porque muchos lenguajes y/o *frameworks* no están preparados para ello.

Para el sistema que se desarrolla, las desventajas que presenta esta arquitectura son irrelevantes ya que se desea un sistema de fácil estructuración y que es desarrollado con tecnologías que facilitan el trabajo con dicha arquitectura.

3.4.2 Patrones de diseño

Los patrones de diseño son una herramienta muy potente para el desarrollo de software, su objetivo fundamental es que sean reutilizados en el contexto donde los mismos puedan presentarse, puesto que cada patrón tiene un objetivo específico a resolver. La reutilización permite: reducción de tiempos, disminución del esfuerzo de mantenimiento, eficiencia, consistencia, fiabilidad y protección de la inversión en desarrollos. En este epígrafe se analizan los patrones de diseño que puedan servir de ayuda durante el diseño de la arquitectura y posteriormente en la implementación de la solución propuesta, además se hace referencia a sus principales definiciones (GAMMA y HELM, 1994).

Controlador: el patrón controlador sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos y la que los envía a las distintas clases según el método llamado. Un ejemplo de cómo este patrón se utiliza en la solución propuesta, se muestra en la clase Migrador que recibe los datos enviados desde la interfaz Principal y los envía a la clase Tabla, la cual es la encargada del manejo de la información relacionada con la tabla y los campos asociados a estas.

Bajo acoplamiento: el acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Acoplamiento bajo significa que una clase

no depende de muchas clases. Durante el trabajo realizado con las clases del diseño se utiliza este patrón pues existen pocas relaciones entre las clases lo que demuestra que existen muy pocas dependencias y cada clase realiza sus funciones sin necesidad de otras.

Polimorfismo: es un principio fundamental en que se fundan las estrategias globales, o planes de ataque, al diseñar como organizar un sistema para que se encargue del trabajo. Un diseño basado en la asignación de responsabilidades mediante el polimorfismo puede ser extendido fácilmente para que realice nuevas variantes. Un ejemplo de cómo este patrón se pone de manifiesto se muestra cuando al agregar una nueva clase Campo, su función polimórfica Devolver_Sentencia afectará poco el diseño actual, dada la forma como se maneja dicha operación.

Diagramas de clases del diseño

Los diagramas de clases de diseño describen gráficamente las especificaciones de las clases del software y contienen las clases, atributos, métodos, navegabilidad y dependencias existentes entre ellas (LARMAN, 1999a). A continuación se muestran las clases modeladas a partir de las clases del análisis mostradas en el capítulo anterior. Las restantes pueden observarse en el [Anexo 7](#).

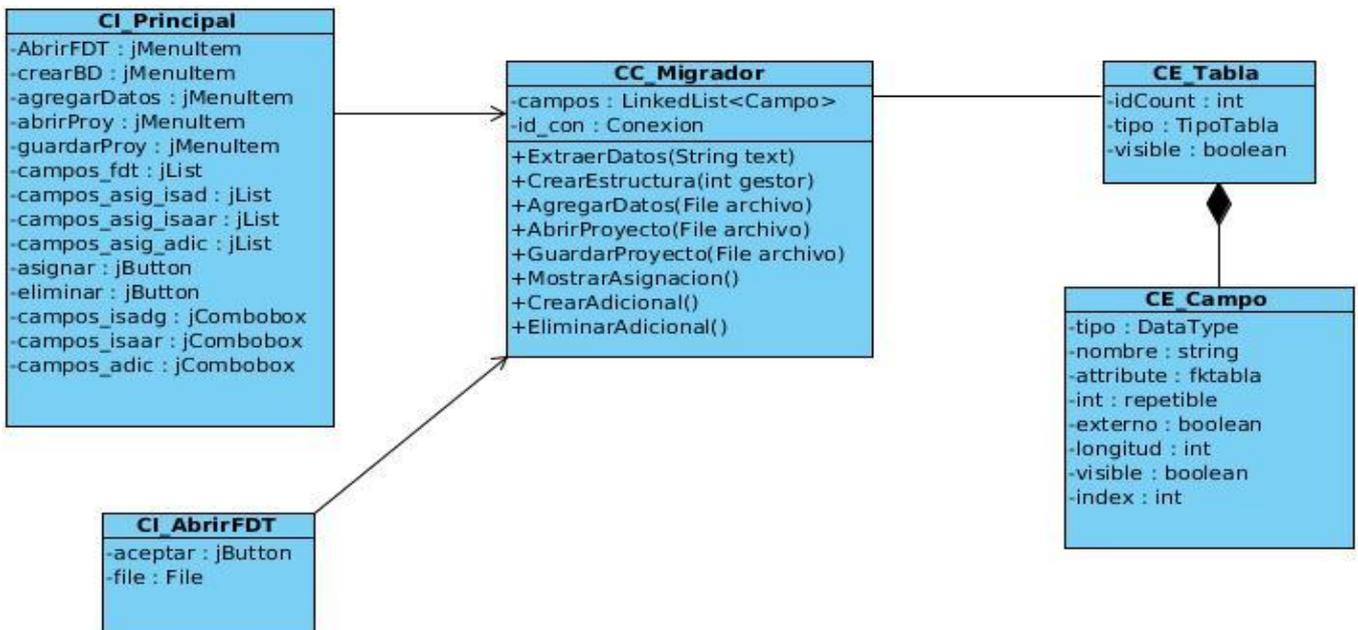


Figura 11: Diagrama de clases del diseño del caso de uso Abrir el archivo FDT

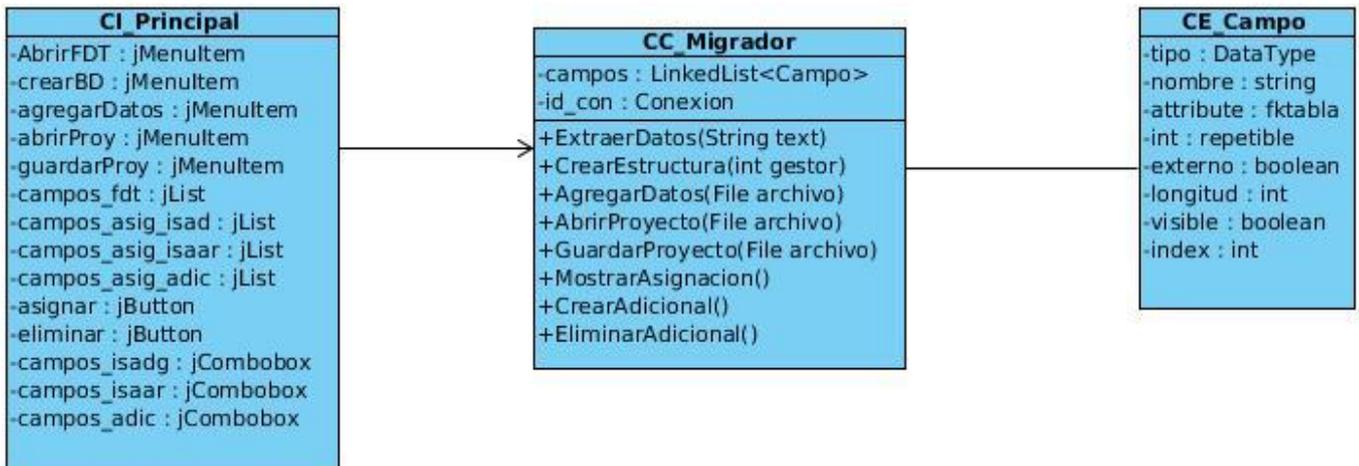


Figura 12: Diagrama de clases del diseño del caso de uso Gestionar asignación de campos ISAD (G)

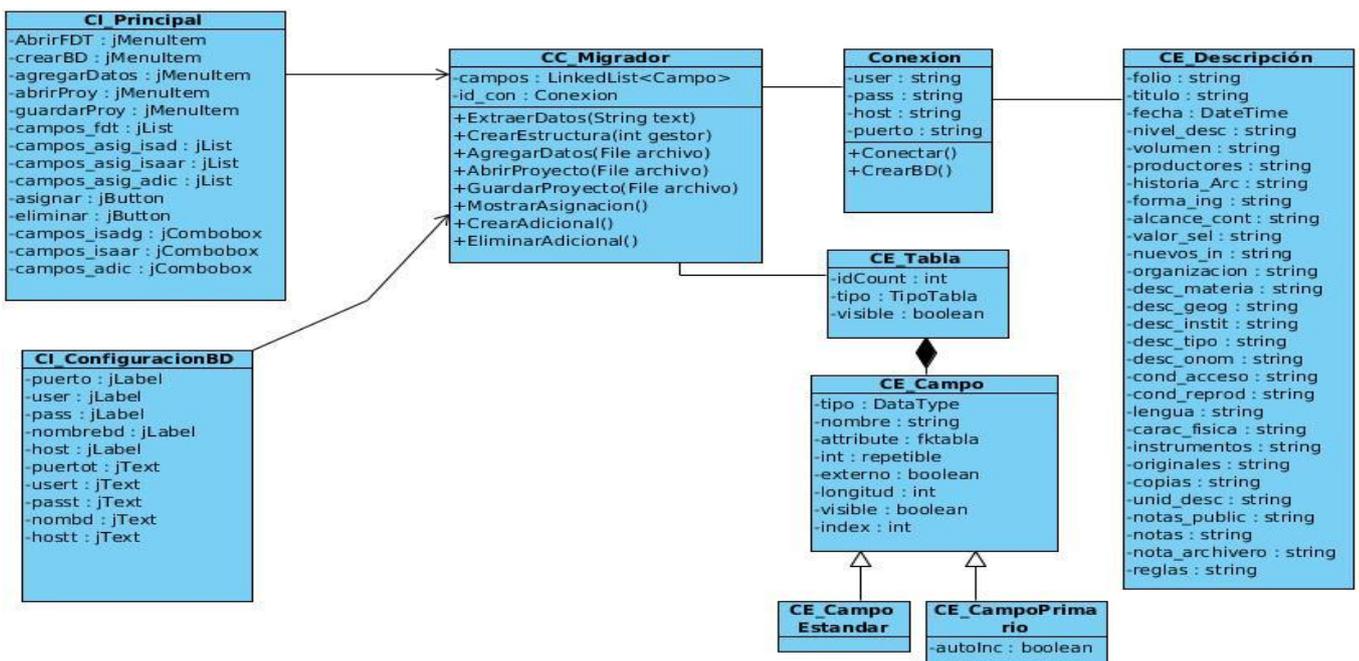


Figura 13: Diagrama de clases del diseño del caso de uso Crear BD

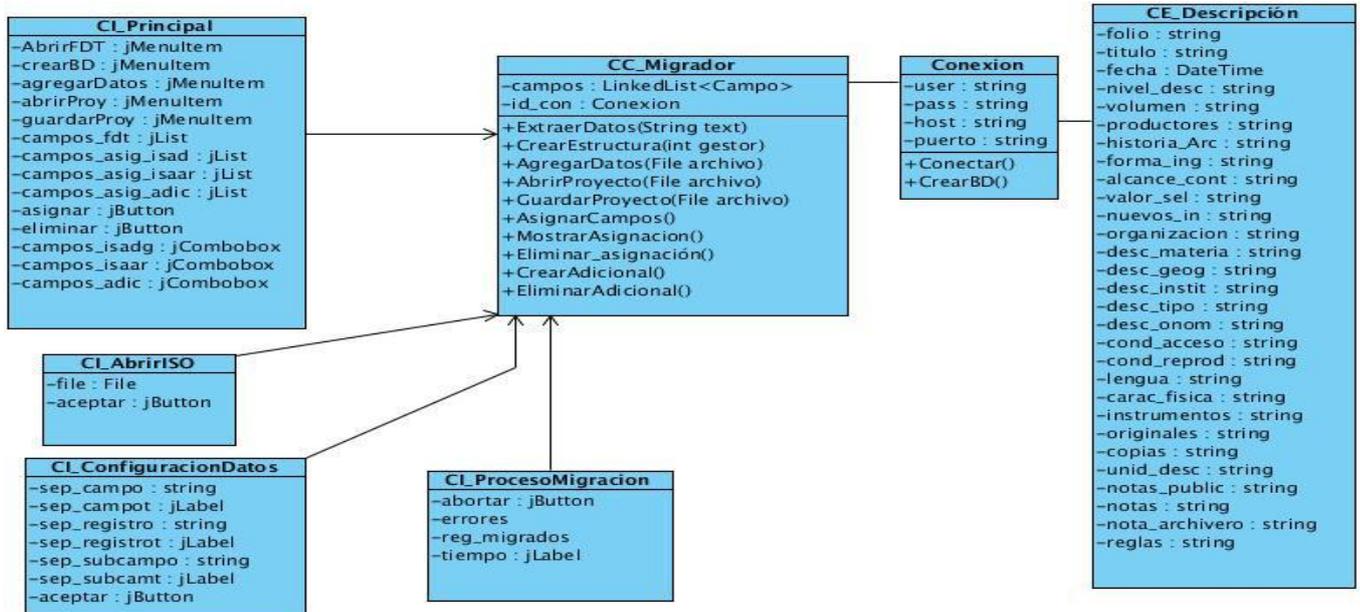


Figura 14: Diagrama de clases del diseño del caso de uso Agregar datos

3.5 Tratamiento de errores

Para una mayor seguridad y confiabilidad de los usuarios con el sistema, se lleva a cabo el tratamiento de los errores que se puedan generar durante el trabajo con la aplicación. Cada vez que se lanza una excepción u ocurre un error en el sistema, el usuario inmediatamente es notificado de dicho error y de las posibles causas que generaron el mismo.

3.6 Conclusiones parciales

En este capítulo se llevó a cabo el desarrollo del flujo de trabajo de análisis y diseño propuesto por el proceso de desarrollo RUP, teniendo en cuenta que el mismo es uno de los flujos más importantes que se llevan a cabo en el ciclo de vida de cualquier producto software.

Se realizó el modelado del análisis y diseño con sus artefactos requeridos, a partir de los casos de uso del sistema propuesto, como son los diagramas de clases del análisis y del diseño, diagramas de interacción. También se plantearon los principios del diseño, incluyendo en él aquellos patrones utilizados para modelar el diseño, teniendo en cuenta la arquitectura propuesta; siendo estos artefactos un resultado observable para la implementación del sistema.

Capítulo 4 Implementación y prueba

4.1 Introducción

En el presente capítulo se lleva a cabo el desarrollo de los flujos de trabajo de implementación y prueba. Se muestra la distribución física de los distintos componentes lógicos desarrollados, a través del modelo de despliegue. Además se muestra la organización del sistema mediante el modelo de componentes, el cual representa la vista estática del sistema. Se define el modelo de prueba, describiendo cómo hacer las pruebas al sistema verificando que todos los requisitos hayan sido implementados, determinando los defectos del mismo.

4.2 Implementación

El flujo de trabajo Implementación enmarca el comienzo de la fase de Construcción. El propósito general de la implementación es desarrollar la arquitectura y el sistema como un todo. La aplicación que se desarrolle debe tener la calidad requerida para su uso y cumplir con los requisitos de software determinados en el segundo capítulo.

4.2.1 Diagrama de despliegue

En el diagrama de despliegue se muestra cómo y dónde se despliega el sistema. Las máquinas físicas y los procesadores se representan como nodos, y la construcción interna puede ser representada por nodos o artefactos embebidos. Los estereotipos permiten precisar la naturaleza del equipo: dispositivos, procesadores y memoria (JACOBSON *et al.*, 2000e).

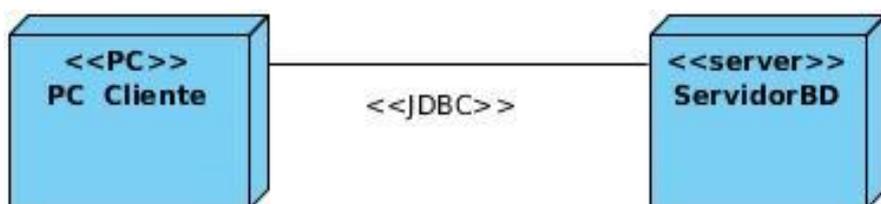


Figura 15: Diagrama de despliegue de la herramienta para la migración de datos hacia la norma ISAD (G)

Descripción de los nodos:

- PC Cliente: ejecuta la aplicación y se comunica con el servidor de base de datos mediante el driver JDBC.

- ServidorBD: permite la creación de las bases de datos correspondientes y organiza los datos en una o más tablas relacionadas. Se utiliza debido a que la aplicación necesita almacenamiento de datos estructurados.

4.2.2 Diagrama de componentes

Un diagrama de componentes representa cómo 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. Muestra la organización y las dependencias entre un conjunto de componentes. No es del todo necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes, cada diagrama describe un apartado del sistema. Uno de sus usos principales es que puede servir para ver qué componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema (JACOBSON *et al.*, 2000d).

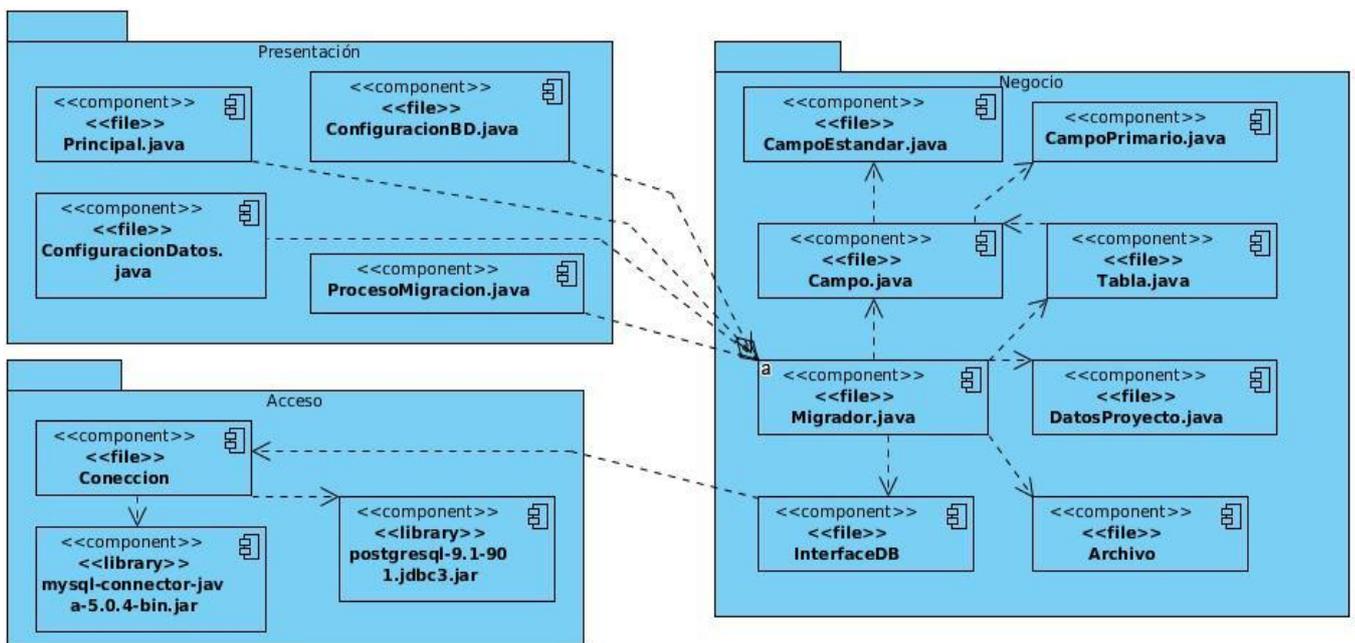


Figura 16: Diagrama de componentes de la herramienta para la migración de datos hacia la norma ISAD (G)

4.3 Estándar de codificación

Cuando se habla de un estándar de codificación completo este comprende todos los aspectos de la generación de código y debe tender siempre a lo práctico. El código fuente debe reflejar un estilo armonioso, como si hubiese sido escrito por un solo programador de una sola vez.

Al comenzar un proyecto de software debe de establecerse un estándar de codificación para que todos los implicados trabajen de forma coordinada. En caso de que el proyecto incorpore código fuente previo, el estándar debe establecer cómo operar con la base de código existente. La legibilidad del código es de vital importancia, pues repercute directamente en los programadores con su capacidad de comprender el sistema, en caso de darle mantenimiento reduce los costos en tiempo para añadirle nuevas características, modificar las ya existentes, depurar errores y mejorar el rendimiento. Aunque la legibilidad y el mantenimiento son el resultado de muchos factores, se le presta especial atención al uso del estándar de codificación como uno de los factores principales que influye en estas dos características.

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vista a generar un código de alta calidad es de gran importancia para el producto final del software, y como se ha mencionado anteriormente, para obtener un buen rendimiento. Además si se aplica de forma continua un estándar de codificación bien definido, se utilizan técnicas de programación apropiadas y posteriormente, se efectúan revisiones de código de rutina, caben muchas posibilidades de que un proyecto de software se convierta en un sistema de software fácil de comprender y mantener.

Tipo de identificador	Reglas de nomenclatura	Ejemplos
Constantes	Los nombres de las variables declaradas como constantes deben ir totalmente en mayúsculas separando las palabras con un guion bajo (“_”). (Las constantes ANSI ²⁰ se deben evitar, para facilitar la depuración).	static final int ANCHURA_MINIMA = 4; static final int ANCHURA_MAXIMA = 999;
Paquetes	El prefijo del nombre de un paquete se escribe siempre con letras ASCII en minúsculas, y debe ser uno de los nombres de dominio de alto nivel	com.sun.eng com.apple.quicktime.v2 edu.cmu.cs.bovik.cheese

²⁰ Instituto Nacional de Normalización Estadounidense.

	(actualmente com, edu, gov, mil, net, org) o uno de los códigos ingleses de dos letras que identifican cada país como se especifica en el estándar ISO 3166, 1981. Los siguientes componentes del nombre del paquete varían de acuerdo a las convenciones de nomenclatura internas de cada organización. Dichas convenciones pueden especificar que algunos nombres de los directorios correspondan a divisiones, departamentos, proyectos o máquinas.	
Clases	Los nombres de las clases deben ser sustantivos, cuando son compuestos tienen la primera letra de cada palabra que lo forma en mayúsculas. Se deben mantener simples y descriptivos los nombres de las clases. Usar palabras completas, evitar acrónimos y abreviaturas (a no ser que la abreviatura sea mucho más conocida que el nombre completo, como URL ²¹ o HTML ²²).	class Cliente; class ImagenAnimada;
Interfaces	Los nombres de las interfaces siguen la misma regla que las clases.	Interface ObjetoPersistente; interface Almacen;
Métodos	Excepto las constantes, todas las instancias y variables de clase o método empezarán con minúscula. Las palabras internas que lo forman (si son compuestas) empiezan con su primera letra en mayúsculas. Los nombres de variables no deben empezar con los caracteres guion bajo “_” y coma “,” aunque ambos están permitidos por el lenguaje. Los nombres de las variables deben ser	int i; char c; float miAnchura

²¹ Localizador de Recurso Uniforme, la dirección global de documentos y de otros recursos en la red.

²² Lenguaje de marcas de hipertexto.

	<p>cortos pero significativos. La elección del nombre de una variable debe ser un mnemónico, designado para indicar a un observador ocasional su función. Los nombres de variables de un solo carácter se deben evitar, excepto para variables índices temporales. Nombres comunes para variables temporales son i, j, k, m, y n para enteros; c, d, y e para caracteres.</p>	
--	---	--

Tabla 6: Descripción del estándar de codificación

4.4 Pruebas de software

Las pruebas de software son un elemento crítico para la garantía de la calidad del mismo y representan una revisión final de las especificaciones, el diseño y la codificación. Son una actividad en la cual un sistema o componente de este es ejecutado bajo condiciones o requerimientos especificados, los resultados se observan y registran, y se hace una evaluación de algún aspecto del sistema o componente. El objetivo de las pruebas de software es descubrir errores (PRESSMAN, 2002b).

4.4.1 Pruebas de caja negra

Las pruebas de caja negra se llevan a cabo sobre la interfaz del sistema y se centran en los requisitos funcionales del mismo. Es decir, permite al ingeniero de sistemas obtener un conjunto de condiciones de entrada, que ejerciten completamente todos los requisitos funcionales de un programa.

La prueba de caja negra pretende demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, y se mantiene la integridad de la información externa. Una prueba de caja negra examina algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software.

Según Pressman los tipos de errores que intenta descubrir esta prueba son: funciones incorrectas o ausentes, errores de interfaz, errores en las estructuras de datos o en accesos a bases de datos externas, errores de rendimiento, errores de inicialización y de terminación (PRESSMAN, 2002a). En las siguientes tablas se muestran los casos de prueba realizados a los casos de uso que fueron identificados arquitectónicamente significativos.

Entrada	Resultado	Condiciones
El usuario abre un archivo FDT correctamente.	El sistema muestra la interfaz principal y los campos del archivo abierto.	El archivo que se desea abrir, debe estar en formato FDT.
El usuario abre un archivo con formato incorrecto.	El sistema muestra el mensaje de error: "El fichero seleccionado no es válido".	

Tabla 7: Caso de prueba Abrir archivo FDT

Entrada	Resultado	Condiciones
El usuario asigna correctamente campos del FDT al campo de la norma ISAD (G) seleccionado.	El sistema elimina los campos del FDT seleccionados y los agrega a la lista de campos asignados al campo de la norma ISAD (G).	Que se haya abierto el archivo FDT. Seleccionar al menos un campo a asignar.
El usuario no selecciona campos del FDT a asignar.	El sistema muestra el mensaje de error: "El fichero seleccionado no es válido".	

Tabla 8: Caso de prueba Gestionar asignación de campos ISAD (G) –SC1 Asignar campos ISAD (G)

Entrada	Resultado	Condiciones
El usuario selecciona el campo de la ISAD (G).	El sistema muestra los campos asignados al campo que fue seleccionado.	Que se haya abierto el archivo FDT. Seleccionar el campo de la ISAD (G).

Tabla 9: Caso de prueba Gestionar asignación de campos ISAD (G) –SC2 Mostrar asignación de campos ISAD (G)

Entrada	Resultado	Condiciones
El usuario elimina correctamente campos del FDT asignados al campo de la norma ISAD (G) seleccionado.	El sistema muestra un mensaje de confirmación, de ser confirmado elimina los campos asignados y los agrega a la lista de campos del FDT.	Seleccionar el campo de la norma ISAD (G) y al menos un campo asignado a este que se desea eliminar.
El usuario elimina campos del FDT de una lista vacía.	El sistema muestra el mensaje: "No existen campos a eliminar."	
El usuario no selecciona campos asignados a eliminar.	El sistema muestra el mensaje: "Debe seleccionar al menos un campo a eliminar."	

Tabla 10: Caso de prueba Gestionar asignación de campos ISAD (G) –SC3 Eliminar asignación de campos ISAD (G)

Entrada	Resultado	Condiciones
El usuario crea la base de datos correctamente.	El sistema crea la base de datos.	Seleccionar la opción CrearBD y especificar los datos correspondientes a la creación de la base de datos.
El usuario crea la base de datos especificando los datos incorrectamente.	El sistema muestra un mensaje informando del error detectado.	
El usuario crea la base de datos dejando campos vacíos.	El sistema muestra el mensaje de error: "No debe dejar campos obligatorios en blanco".	
El usuario crea la base de datos con un nombre existente en el servidor.	El sistema muestra el mensaje de error: "Ya existe una base de datos con ese nombre. Debe renombrarla o eliminarla".	

Tabla 11: Caso de prueba Crear base de datos

Entrada	Resultado	Condiciones
El usuario agrega los datos correctamente.	El sistema agrega los datos correspondientes a la base de datos creada.	Seleccionar la opción Agregar datos y especificar los datos correspondientes al archivo ISO a cargar, abrir el archivo con formato ISO 2709.
El usuario agrega los datos especificándolos incorrectamente.	El sistema muestra el siguiente mensaje de error: "Los datos especificados son incorrectos".	
El usuario agrega los datos dejando campos vacíos.	El sistema muestra el siguiente mensaje de error: "No debe dejar campos obligatorios en blanco".	
El usuario agrega datos cargando un archivo incorrecto.	El sistema muestra el mensaje de error: "El fichero seleccionado no es válido".	

Tabla 12: Caso de prueba Agregar datos

Las pruebas funcionales realizadas al software utilizando el método de caja negra arrojaron en una primera iteración 12 no conformidades dentro de ellas 9 significativas, 1 no significativa y 2 recomendaciones, en una segunda iteración 5 no conformidades entre ellas 2 significativas, 2 no significativas y 1 recomendación; y en una tercera iteración se encontró 1 recomendación, lo que logró constatar que las funcionalidades cumplen con los objetivos propuestos en ellas.

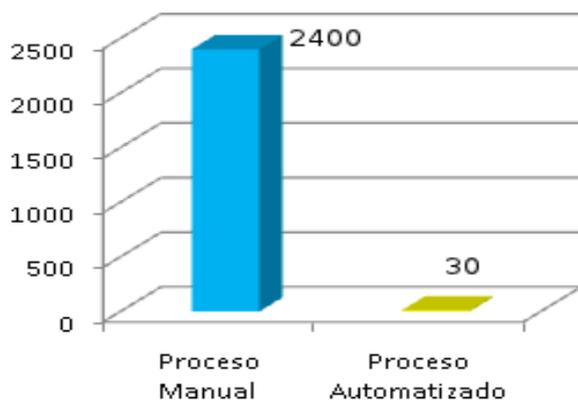
Además, con el objetivo de medir las ventajas reales que brinda la herramienta para la migración de datos se desarrollan pruebas de rendimiento del software determinando que tan rápido el sistema realiza dicho proceso, bajo ciertas condiciones.

El proceso de inserción de datos realizado manualmente a los sistemas gestores de bases de datos que utilizan los sistemas desarrollados por el proyecto, a partir de una estructura de datos exportada del Winisis, que contiene 54 campos y un total de 3019 registros fue desarrollado por un conjunto de 20 personas, en aproximadamente 10 días trabajando 4 horas diarias. Mientras que la migración de datos realizada por la herramienta para la migración de datos hacia la norma ISAD (G) se desarrolló en aproximadamente 30 minutos por una persona, detectando 22 registros con errores en el formato que

no pudieron ser migrados, lo que permitió cumplir con las expectativas esperadas, minimizando el tiempo y esfuerzo por parte del equipo de trabajo y garantizando que estos datos puedan ser consultados en la red a través de los servidores de bases de datos.

En las siguientes gráficas se demuestra lo expresado anteriormente:

Tiempo (min)



Personal

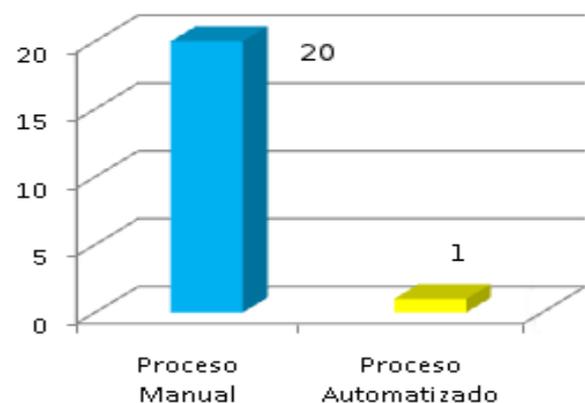


Figura 17: Gráficas asociadas a las pruebas de rendimiento relativas a tiempo y personal empleado.

4.5 Conclusiones parciales

En el presente capítulo se expuso la estructura de la aplicación a través de los modelos de despliegue y componente, además se concibieron un conjunto de pruebas para ser aplicadas al sistema con el objetivo de medir la calidad del producto realizado y validar la solución propuesta.

Se aplicaron un total de 9 diseños de casos de pruebas, de los cuales se identificaron 18 no conformidades en tres iteraciones realizadas al software, las cuales fueron corregidas en su totalidad.

Conclusiones generales

Para la realización de la propuesta de solución se hizo necesaria la elaboración del marco teórico-conceptual de la investigación y posterior a ello la realización del análisis de los sistemas que permiten realizar la migración de datos ISIS hacia los sistemas gestores de bases de datos MySQL y PostgreSQL.

Se logró diseñar e implementar una herramienta para migrar datos desde Winisis hacia los sistemas gestores de bases de datos MySQL y PostgreSQL, logrando la estandarización de estos datos bajo normas internacionales para la descripción de archivos.

La herramienta desarrollada permitió reducir más del 90%, el tiempo y esfuerzo dedicado a la reutilización de la información digitalizada, en las instituciones donde se despliegan los productos informáticos desarrollados por el departamento.

Los resultados obtenidos en el presente trabajo de diploma permitieron dar cumplimiento a los objetivos específicos planteados.

Recomendaciones

Luego de haber concluido la presente investigación se recomienda:

Continuar el desarrollo de este sistema, generalizando la migración desde otras herramientas de almacenamiento y brindar además la posibilidad de hacerlo extensivo hacia otros gestores de bases de datos.

Permitir la utilización de otras normas de descripción de archivos como la norma ISDIAH para la descripción de entidades que custodian los archivos, en el proceso de estandarización de datos, aprovechando las funcionalidades realizadas.

Glosario de términos

CASE: Computer Aided Software Engineering (Ingeniería de Software Asistida por Ordenador), aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero.

Campo: unidad menor de información sobre un objeto (almacenada en la BD) y representa una propiedad de un objeto.

Hardware: dispositivos físicos como la placa base, la CPU o el monitor.

Herramienta: programa o conjunto de programas que permiten la realización de una tarea.

IDE: programa informático compuesto por un conjunto de herramientas de programación.

Internet: conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP.

JDBC: Java Database Connectivity, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java.

Llaves foráneas: una llave foránea es un atributo (puede ser compuesto) de una relación cuyos valores deben de concordar con los de una llave primaria de alguna relación.

ODBC: es un estándar de acceso a bases de datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué SGBD almacene los datos.

QT: framework multiplataforma para la creación de interfaces gráficas de aplicaciones o desarrollo de aplicaciones sin entorno gráfico.

Software: conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

SQL: lenguaje de acceso a bases de datos relacionales que permite la especificación de distintos tipos de operaciones entre estas.

XML: lenguaje de marcado ampliable o extensible. Es un estándar para el intercambio de datos entre diversas aplicaciones.

Referencias bibliográficas

- ARCHIVOS, C. I. D. *ISAAR (CPF)*. 2 ed. Consejo Internacional de Archivos, 2003. 8-35 p. ISAAR (CPF) International Standard Archival Authority Record For Corporate Bodies, Persons and Families.
- BOGLIOLO, E. y BORCHES, P. *Winisis*. Buenos Aires: 1999. Manual para instructores de Winisis. ISBN 987-97059-5-5.
- CARNICER, M. D. y FERNÁNDEZ, B. D. *Norma ISAD(G)*. España: Consejo Nacional de Archivos, 2000. 19-25 p. ISAD(G): Norma Internacional de Descripción Archivística.
- CASTRO, R. *Archivo* Última actualización: 10 de Abril de 2009. [Consultado el: 10 de Enero de 2012]. (Decreto-Ley No. 265 del Sistema Nacional de Archivos de la República de Cuba). Disponible en: <http://correo.servisa.tur.cu/Capacitacion/Consultor/09%20Legislacion/Decreto%20Ley/DL-2008-265.htm>.
- CASTRO, F. Documento de archivo. Ley No.714 Consejo de estado de la República de Cuba, 1960, 2 p.
- CÓDOVA, C. Requisitos funcionales. Cumaná, Centro Local Sucre: 2007a. 66 p. Implementación de requisitos no funcionales a través de la programación orientada a aspectos.
- . *Requisitos no funcionales*. Cumaná, Centro Local Sucre: 2007b. 67 p. Implementación de requisitos no funcionales a través de la programación orientada a aspectos.
- CONVERSE, T.; PARK, J., *et al.* *MySQL*. Indianapolis: Wiley Publishing, Inc, 2004. 3 p. PHP5 and MySQL® Bible. ISBN 0-7645-5746-7.
- CORPORATION, M. *Diccionario de Informática e Internet de Microsoft*. España: McGraw-Hill, 2001. ISBN 84-481-2893-1.
- CORPORATION, O. *NetBeans IDE 7.0.1 Release Information* [Consultado el: 25 de Enero de 2012]. Disponible en: <http://netbeans.org/community/releases/70/>.
- FERREYRA, D. *Isis2SQL R020* Bibliotecología y Ciencias de la Información,, [Consultado el: 3 de Marzo de 2012]. Disponible en: <http://www.r020.com.ar/kiosko.php>.
- GAMMA, E. y HELM, R. *Patrones de diseño*. Addison Wesley Professional, 1994. 11-12 p. Desing Patterns. ISBN 978-0201633610.

- GARCÍA, L. F. J. *Arhivística*. Costa Rica: Comisión Electoral de la Universidad de Costa Rica, 2006. 8 p. El Sistema Nacional de Información Archivística: Como Modelo de Unificación. ISBN 9968-936-26-x.
- GONZÁLEZ, G. J. *Unidad Documental Compuesta*. Colombia: Gráficos Colombia Ltda, 2003a. 16 p. Ordeación Documental. División de clasificación y descripción. ISBN 958-9298-87-7.
- . *Unidad Documental Simple*. Colombia: Gráficos Colombia Ltda, 2003b. 19 p. Ordeación Documental. División de clasificación y descripción. ISBN 958-9298-87-7.
- HERRERA, A. H. *Descripción archivística*. Gráficas del Sur. Becas, 1991. 299-300 p. Archivística general. Teoría y Práctica. ISBN 84-7798-056-x.
- JACOBSON, I.; BOOSH, G., *et al.* *Arquitectura*. Imprenta Fareso S.A: Addison Wesley Logan Inc, 2000a. 56-58 p. El Proceso Unificado de Desarrollo de Software. ISBN 0-201-57169-2.
- . *Diagrama de clase del análisis*. Imprenta Fareso S.A: Addison Wesley Logan Inc, 2000b. El Proceso Unificado de Desarrollo de Software. ISBN 0-201-57169-2.
- . *Diagrama de colaboración*. Imprenta Fareso S.A: Addison Wesley Logan Inc, 2000c. 44-45 p. El Proceso Unificado de Desarrollo de Software. ISBN 0-201-57169-2.
- . *Diagrama de componente*. Imprenta Fareso S.A: Addison Wesley Logan Inc, 2000d. 257-259 p. El Proceso Unificado de Desarrollo de Software. ISBN 0-201-57169-2.
- . *Diagrama de despliegue*. Imprenta Fareso S.A: Addison Wesley Logan Inc, 2000e. 217 p. El Proceso Unificado de Desarrollo de Software. ISBN 0-201-57169-2.
- . *Modelo de dominio*. Imprenta Fareso S.A: Addison Wesley Logan Inc, 2000f. 112-113 p. El Proceso Unificado de Desarrollo de Software. ISBN 0-201-57169-2.
- . *Patrón de arquitectura*. Imprenta Fareso S.A: Addison Wesley Logan Inc, 2000g. 67-69 p. El Proceso Unificado de Desarrollo de Software. ISBN 0-201-57169-2.
- . *RUP*. Imprenta Fareso S.A: Addison Wesley Logan Inc, 2000h. 4 p. El Proceso Unificado de Desarrollo de Software. ISBN 0-201-57169-2.
- LARMAN, C. *Diagrama de clases del diseño*. Dawn Speth White: Prentice Hall, Inc., 1999a. 265-266 p. UML y Patrones. ISBN 970-17-0261-1.
- . *UML*. Dawn Speth White: Prentice Hall, Inc., 1999b. 15 p. UML y Patrones. ISBN 970-17-0261-1.

LEÓN, Y. R. *La descripción como parte del tratamiento de los archivos personales en el siglo XXI: en busca de nuevas alternativas* [Consultado el: 20 de Enero de 2012]. Disponible en: http://bvs.sld.cu/revistas/aci/vol14_5_06/aci18506.htm.

LOPERA, M. L. G. *Arhivística. Revista Interamericana de Bibliotecología*, 2009, vol. 32, nº p. 43. ISSN 0120-0976.

MARQUINA, E. *Arquitectura en capas*. Estados Unidos: Microsoft Corporation, 2008. 6-7 p. Guía de Patrones, Prácticas y Arquitectura .NET.

MORA, M. E. R. *Colección documental* Concla.net, [Consultado el: 21 de Enero de 2012]. (Glosario Ilustrado de Terminología Archivística Costarricense). Disponible en: http://www.concla.net/Glosario/Glosario_terminologia_Archivistica_Costarricense_C.html.

---. *Fondo documental* Concla.net, [Consultado el: 21 de Enero de 2012]. (Glosario Ilustrado de Terminología Archivística Costarricense). Disponible en: http://www.concla.net/Glosario/Glosario_terminologia_Archivistica_Costarricense_F.html.

---. *Serie documental* Concla.net, [Consultado el: 21 de Enero de 2012]. (Glosario Ilustrado de Terminología Archivística Costarricense). Disponible en: http://www.concla.net/Glosario/Glosario_terminologia_Archivistica_Costarricense_S.html.

---. *Sub-Fondo documental* Concla.net, [Consultado el: 21 de Enero de 2012]. (Glosario Ilustrado de Terminología Archivística Costarricense). Disponible en: http://www.concla.net/Glosario/Glosario_terminologia_Archivistica_Costarricense_S.html.

---. *Sub-Serie documental* Concla.net, [Consultado el: 21 de Enero de 2012]. (Glosario Ilustrado de Terminología Archivística Costarricense). Disponible en: http://www.concla.net/Glosario/Glosario_terminologia_Archivistica_Costarricense_S.html.

MUGICA, M. M. *Documento de archivo*. La Habana: Editorial Félix Varela, 2005a. 17 p. Gestión documental y organización de archivos ISBN 959-258-950-x.

---. *Gestión documental*. La Habana: Editorial Félix Varela, 2005b. 4 p. Gestión documental y organización de archivos ISBN 959-258-950-x.

MUNDET, J. R. C. *Archivo*. 2 ed. Madrid, España: Idavel, Los Llanos, 1996. 91-92 p. Manual de archivística. ISBN 84-368-0860-6.

- PARADING, V. *Visual Paradigm for UML* [Consultado el: 22 de Enero de 2012]. Disponible en: <http://www.visual-paradigm.com>.
- PEIS, E. y RUIZ, A. A. *Normas de descripción archivística* Universitat Pompeu Fabra, Última actualización: 6 de Mayo de 2010. [Consultado el: 21 de Enero de 2012]. (EAD (Encoded Archival Description): Desarrollo, estructura, uso y aplicaciones). Disponible en: <http://www.hipertext.net/web/pag223.htm>.
- POSTGRESQL. *About PostgreSQL* Global Development Group, [Consultado el: 22 de Enero de 2012]. Disponible en: <http://www.postgresql.org/about/>.
- PRESSMAN, R. S. *Pruebas de caja negra*. 5 ed. Indianapolis: McGraw-Hill, 2002a. 294-296 p. Ingeniería del Software, Un Enfoque Practico. ISBN 84-481-3214-9.
- . *Pruebas del software*. 5 ed. Indianapolis: McGraw-Hill, 2002b. 282-283 p. Ingeniería del Software, Un Enfoque Practico. ISBN 84-481-3214-9.
- SCHILDT, H. *Java*. Fareso S.A: Mcgraw-Hill Companies, 2001. 3-17 p. Java 2. Manual de referencia. ISBN 84-481-3173-8.
- TALEND. *Migración de datos* Talend, [Consultado el: 25 de Enero de 2012]. (Data Migration). Disponible en: <http://www.talend.com/solutions-data-integration/data-migration.php>.
- TEJEDA, Y. S. *Rhombifex* Comunidad de Software Libre en Granma, [Consultado el: 6 de Febrero de 2012]. (Proyectos en Granma). Disponible en: <http://www.grm.jovenclub.cu/comunidadswl/index.php/proyectos>.
- TURNER, R. Especificación de requisitos de software. *Journal of Logic and Computation*, 2005, vol. 15, nº 5, p. 623-650.

Bibliografía consultada

- ARCHIVOS, C. I. D. *ISAD (G): Norma Internacional de Descripción Archivística*. Fernando Luis ed. España: Consejo Internacional de Archivos, 1999.
- CDS/ISIS, D. N. *Manual para instructores de Winisis*. Argentina: Comisión Nacional de Energía Atómica, 1999. ISBN 987-97059-5-5.
- CORNEJO, J. E. G. *Arquitectura en capas* [Consultado el: 2 de Febrero de 2012]. (Arquitectura en Capas ~ DNA. Un camino hacia los procesos distribuidos). Disponible en: http://www.docirs.cl/arquitectura_tres_capas.htm.
- CORPORATION, M. *Diccionario de Informática e Internet de Microsoft*. España: McGraw-Hill, 2001. ISBN 84-481-2893-1.
- CORPORATION, O. *MySQL-AB* [Consultado el: 4 de Febrero de 2012]. (Panorámica de MySQL AB). Disponible en: <http://dev.mysql.com/doc/refman/5.0/es/what-is-mysql-ab.html>.
- . *Qué es MySQL* [Consultado el: 4 de Febrero de 2012]. (Panorámica del sistema de gestión de base de datos MySQL). Disponible en: <http://dev.mysql.com/doc/refman/5.0/es/what-is.html>.
- DONAY, M. C. y NOGUERA, M. C. *Estrategias para promover los archivos en el seno de las administraciones*. Lligall, 2001, vol. 4, nº
- EDITORIAL, O. G. *Diccionario Ilustrado Océano de la Lengua Española*. España: MMI-Océano Grupo Editorial S.A, 2001. ISBN 84-494-1535-7.
- ESTRADA, A. F. y MARTINEZ, R. D. *Proceso de mejora* Calisoft, Centro de Calidad para Soluciones Informáticas, [Consultado el: 2 de Mayo de 2012]. Disponible en: <http://calisoft.uci.cu/index.php/proceso-de-mejora>.
- FERNÁNDEZ, F. y MUÑOZ, Y. *JDBC* Universidad de Chile: Departamento de Ciencias de la Computación., [Consultado el: 5 de Febrero de 2012]. Disponible en: <http://www.dcc.uchile.cl/~lmateu/CC60H/Trabajos/ifernand/>.
- FORTÚN, L. J. y HUESO, A. D. *Cuadro de Clasificación para los Archivos Municipales y Concejiles de Navarra*. España: Huarte Gráfica, S.A.L., 2001. vol. 2, ISBN 84-235-1976-7.
- FUGUERAS, R. A. I. *Los sistemas de gestión integral de documentos como aliados de las políticas de eficiencia y rentabilidad en los ayuntamientos*. *Gestión del Patrimonio Local: El Patrimonio Documental*, 2001, vol. 1.

- GILFILLAN, I. *La Biblia de MySQL*. España: Anaya Multimedia, 2003. ISBN 8441515581.
- HERRERA, A. H. *Archivística General*. Diputación Provincial de Sevilla ed. España: Gráficas del Sur. Becas, 1991. vol. 5, ISBN 84-7798-056-X.
- LABORIS. Archivero Anuntis Segundamano España S.L, [Consultado el: 4 de Febrero de 2012]. (Archivero). Disponible en: http://www.laboris.net/static/ca_profesion_archivero.aspx.
- MARQUINA, E. y PARRA, J. D. *Guía de Patrones, Prácticas y Arquitectura .NET*. Microsoft Corporation ed. Estados Unidos: Microsoft Consulting Services, 2006.
- MEJIA, M. *Servicios de Archivo*. Diego Tenorio Conde ed. Colombia: Archivo General de la Nación, 1994. ISBN 958-9298-27-3.
- ROJAS, J. R. *Lenguajes De Programación* [Página web]. SlideShare Inc, Última actualización: 12 de Noviembre de 2007. [Consultado el: 1 de Mayo de 2012]. Disponible en: <http://www.slideshare.net/jrojas/tema1-lenguajes-de-programacion>.
- THIELEDISEÑO. *Aplicaciones Web vs Aplicaciones de Escritorio* ThieleDiseño, Última actualización: 24 de octubre de 2011. [Consultado el: 2 de Mayo de 2012]. Disponible en: <http://loretothiele.blogspot.com/2011/10/aplicaciones-web-vs-aplicaciones-de.html>.
- VALIENTE, J. *Archivística. Manejo y organización de archivos (Manual del archivero)* Mailxmail.com, [Consultado el: 2 de Mayo de 2012]. Disponible en: <http://www.emagister.com/curso-archivistica-manejo-organizacion-archivos-manual-archivero/servicios-archivisticos-administracion-archivos-medidas-ubicacion-estanterias>.
- VILLAVERDE, M. L. C. *Manual de tratamiento de archivos administrativos*. Imprenta Ministerio de Cultura: Dirección de archivos estatales, 1992. ISBN 84-74S3-S70-7.
- ZORRILLA, U. y CALVARRO, J. *Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0*. Krasis Consulting S.L ed. España: Krasis Press, 2010. ISBN 978-84-936696-3-8.

Anexos

Anexo 1: Descripción de casos de uso.

Objetivo	Gestionar la asignación de los campos del archivo FDT generado del Winisis a los campos de la norma ISAAR (CPF).	
Actores	Migrador	
Resumen	El usuario asigna, elimina o muestra la asignación de los campos del archivo FDT a los campos de la norma ISAAR (CPF).	
Complejidad	Media	
Prioridad	Secundario	
Precondiciones	Debe haber abierto el archivo FDT.	
Postcondiciones	Se permite crear la base de datos.	
Flujo de eventos		
Flujo básico <Gestionar asignación de campos ISAAR (CPF)>		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor abre el archivo FDT.	<p>El sistema le muestra una lista con los campos del archivo FDT, una lista de selección con los campos de la norma ISAAR (CPF) y permite realizar varias acciones:</p> <ul style="list-style-type: none"> ➤ Asignar un campo del FDT a un campo de la norma ISAAR (CPF). Ver Sección 1: Asignar campos. ➤ Mostrar los campos asignados a un campo de la norma ISAAR (CPF). Ver Sección 2: “Mostrar campos asignados” ➤ Eliminar campos asignados a un campo de la norma ISAAR (CPF). Ver Sección 3: “Eliminar campos asignados.”
Sección 1: “Asignar campos”		
Flujo básico Asignar campos		

	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona uno a uno los campos del FDT o marca la opción de seleccionar todos para ser asignados y el campo de la norma ISAAR (CPF) al que serán asignados y presiona el botón >>.	El sistema muestra los elementos en la lista de campos asignados asociados al campo de la norma ISAAR (CPF) que fue seleccionado.
2.		Los elimina de la lista de campos del FDT, terminando el caso de uso.

Flujos alternos

Nº Evento <1a Datos vacíos>

	Actor	Sistema
1.		El sistema muestra el siguiente mensaje de error: "Debe seleccionar al menos un campo a asignar."

Sección 2: "Mostrar campos asignados"

Flujo básico Mostrar campos asignados

	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona el campo de la norma ISAAR (CPF).	El sistema muestra los elementos en la lista de campos asignados asociados al campo de la norma ISAAR (CPF) que fue seleccionado, terminando el caso de uso.

Sección 3: "Eliminar campos asignados"

Flujo básico Eliminar campos asignados

	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona el campo de la norma ISAAR (CPF).	
2.	Selecciona los elementos de la lista de campos asignados a ser eliminados marcándolos uno a uno o	El sistema muestra un mensaje de confirmación.

	en la opción de seleccionar todos y presiona el botón <<.	
3.	El usuario acepta confirmando que desea eliminar.	El sistema elimina los elementos de la lista de campos asignados.
4.		Muestra los elementos eliminados en la lista de campos del FDT, terminando el caso de uso.
Flujos alternos		
Nº Evento <1a Cancelar eliminación>		
	Actor	Sistema
1		El sistema no realiza ninguna acción.
Nº Evento <2a Lista vacía>		
1		El sistema muestra el siguiente mensaje de error: "No existen campos a eliminar."
Nº Evento <2b Datos vacíos>		
		El sistema muestra el siguiente mensaje de error: "Debe seleccionar al menos un campo a eliminar."
Prototipo de Interfaz de usuario		

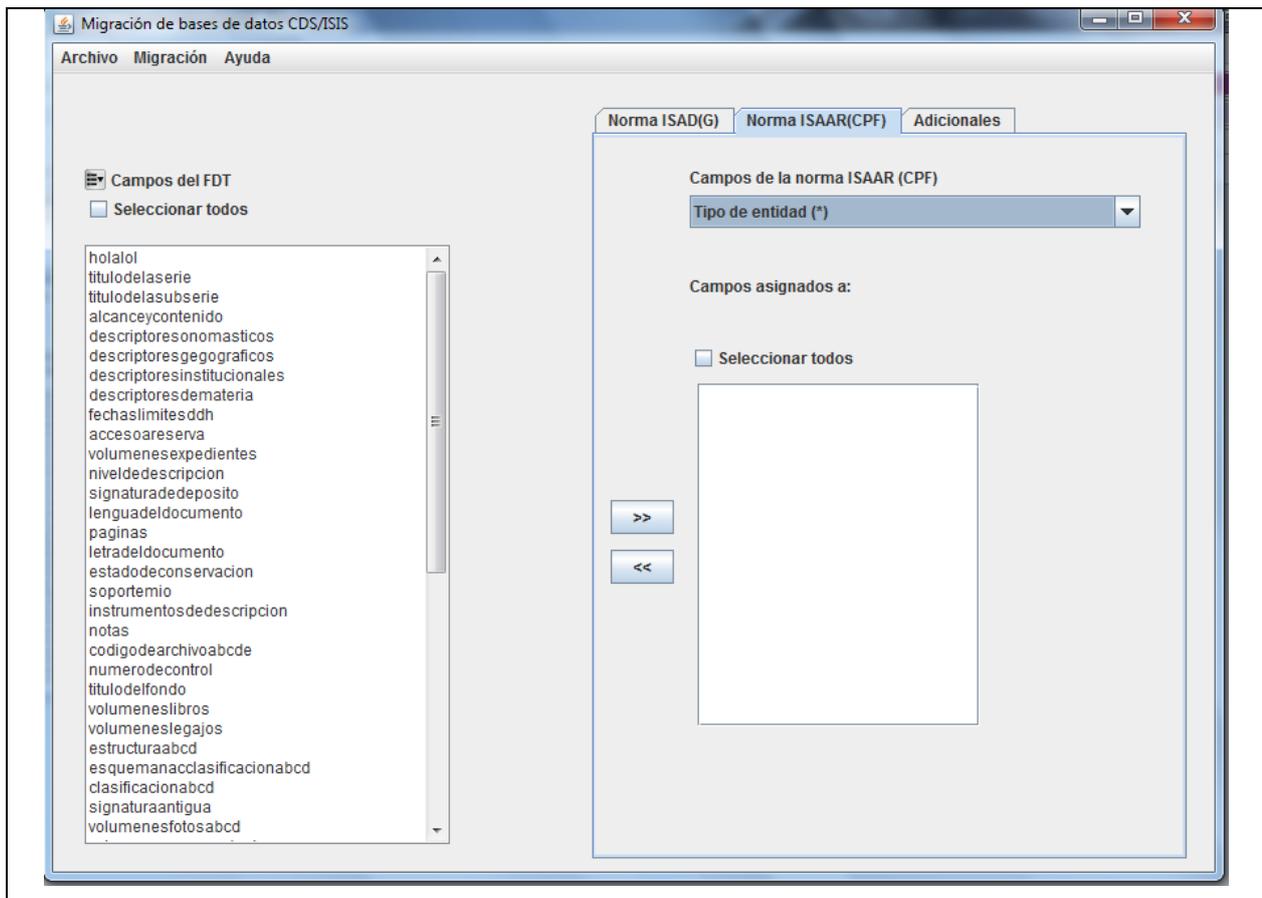


Tabla 13: Descripción del caso de uso Gestionar asignación de campos ISAAR (CPF)

Objetivo	Gestionar la asignación de los campos del archivo FDT generado del Winisis a los campos adicionales.
Actores	Migrador
Resumen	El usuario asigna, elimina o muestra la asignación de los campos del archivo FDT a los campos adicionales.
Complejidad	Media
Prioridad	Secundario
Precondiciones	Debe haber abierto el archivo FDT.
Postcondiciones	Se permite crear la base de datos.

Flujo de eventos**Flujo básico <Gestionar asignación de campos adicionales>**

	Actor	Sistema
1.	El caso de uso se inicia cuando el actor abre el archivo FDT.	<p>El sistema le muestra una lista con los campos del archivo FDT, una lista de selección con los campos adicionales y permite realizar varias acciones:</p> <ul style="list-style-type: none"> ➤ Asignar un campo del FDT a un campo adicional. Ver Sección 1: Asignar campos. ➤ Mostrar los campos asignados a un campo adicional. Ver Sección 2: "Mostrar campos asignados" ➤ Eliminar campos asignados a un campo adicional. Ver Sección 3:"Eliminar campos asignados.

Sección 1: "Asignar campos"**Flujo básico Asignar campos**

	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona uno a uno los campos del FDT o marca la opción de seleccionar todos para ser asignados y el campo adicional al que serán asignados y presiona el botón >>.	El sistema muestra los elementos en la lista de campos asignados asociados al campo adicional que fue seleccionado.
2.		Los elimina de la lista de campos del FDT, terminando el caso de uso.

Flujos alternos**Nº Evento <1a Datos vacíos>**

	Actor	Sistema
1.		El sistema muestra el siguiente mensaje de error: "Debe seleccionar al menos un campo a

		asignar."
Sección 2: "Mostrar campos asignados"		
Flujo básico Mostrar campos asignados		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona el campo adicional.	El sistema muestra los elementos en la lista de campos asignados asociados al campo adicional que fue seleccionado, terminando el caso de uso.
Sección 3: "Eliminar campos asignados"		
Flujo básico Eliminar campos asignados		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona el campo adicional.	
2.	Selecciona los elementos de la lista de campos asignados a ser eliminados marcándolos uno a uno o en la opción de seleccionar todos y presiona el botón <<.	El sistema muestra un mensaje de confirmación.
3.	El usuario acepta confirmando que desea eliminar.	El sistema elimina los elementos de la lista de campos asignados.
4.		Muestra los elementos eliminados en la lista de campos del FDT, terminando el caso de uso.
Flujos alternos		
Nº Evento <1a Cancelar eliminación>		
	Actor	Sistema
1		El sistema no realiza ninguna acción.
Nº Evento <2a Lista vacía>		
1		El sistema muestra el siguiente mensaje de error: "No existen campos a eliminar."
Nº Evento <2b Datos vacíos>		
		El sistema muestra el siguiente mensaje de error: "Debe seleccionar al menos un campo a

eliminar."

Prototipo de Interfaz de usuario

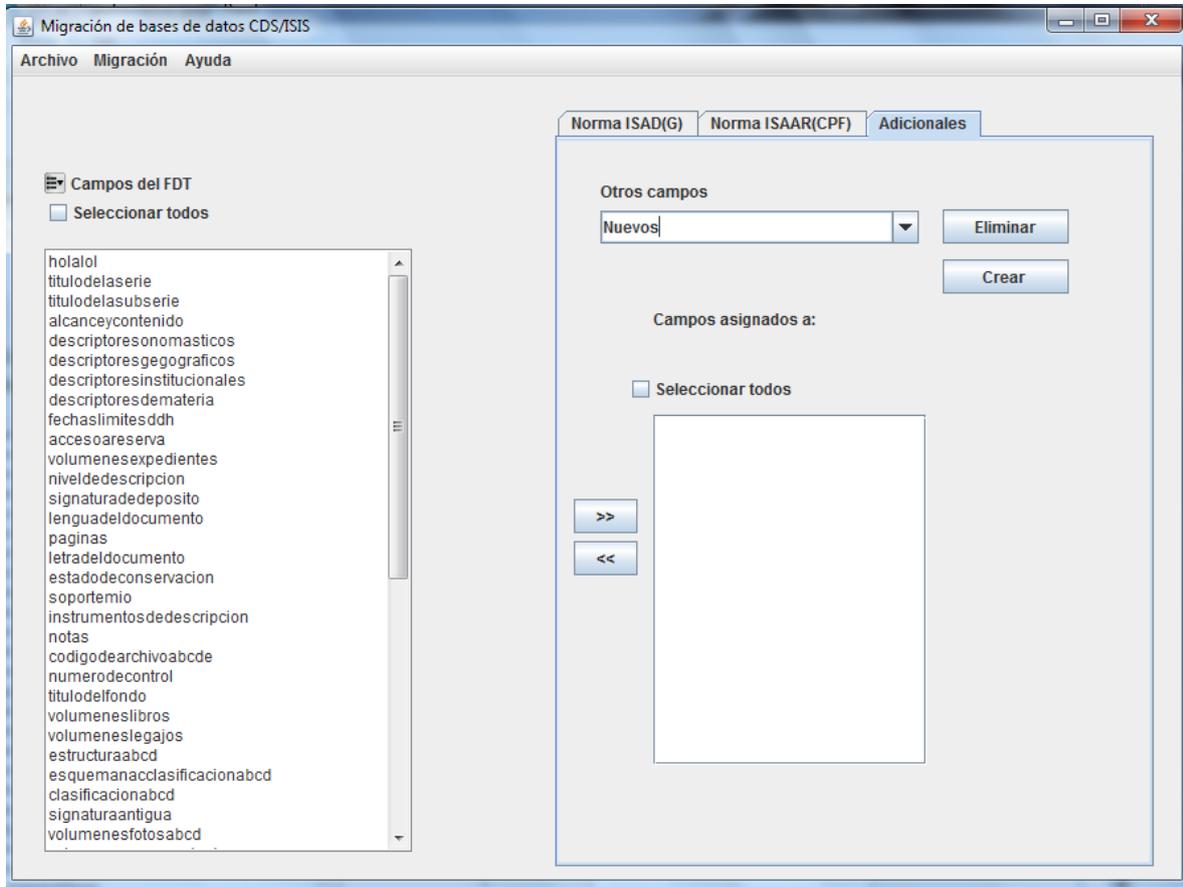


Tabla 14: Descripción del caso de uso Gestionar asignación de campos adicionales

Objetivo	Gestionar campo adicional.
Actores	Migrador
Resumen	El usuario selecciona la opción crear nuevo, el sistema muestra una interfaz para escribir el nuevo campo y se adiciona a la lista de campos adicionales.
Complejidad	Media
Prioridad	Secundario
Precondiciones	Debe haber abierto el archivo FDT.

	Acceder a la pestaña Adicionales.	
Postcondiciones	Se gestiona el campo adicional creado.	
Flujo de eventos		
Flujo básico < Crear campo adicional >		
	Actor	Sistema
	El caso de uso se inicia cuando el actor abre el archivo FDT.	El sistema le muestra una lista con los campos adicionales y permite realizar varias acciones: <ul style="list-style-type: none"> ➤ Crear un campo adicional. Ver Sección 1: Crear campo adicional. ➤ Eliminar un campo adicional. Ver Sección 2: "Eliminar campo adicional.
Sección 1: "Crear campo adicional"		
Flujo básico Crear campo adicional		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona la opción de crear nuevo.	El sistema le muestra una ventana para especificar el nuevo campo.
2.	El usuario especifica el campo nuevo y selecciona la opción crear.	El sistema añade el campo creado a la lista de campos adicionales.
Flujos alternos		
Nº Evento <2a Cancelar >		
	Actor	Sistema
1.		El sistema vuelve a la interfaz principal
2.		
Sección 2: "Eliminar campo adicional"		
Flujo básico Eliminar campo adicional		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona el campo a eliminar.	El sistema verifica que sea un campo válido a ser eliminado.
2.		El sistema elimina dicho campo y si tiene campos asignados al mismo los devuelve a la

		lista de campos del FDT.
Flujos alternos		
Nº Evento <2a El campo seleccionado no es válido >		
	Actor	Sistema
1.		El sistema muestra el siguiente mensaje de error: " No es un campo válido a eliminar".

Prototipo de Interfaz de usuario

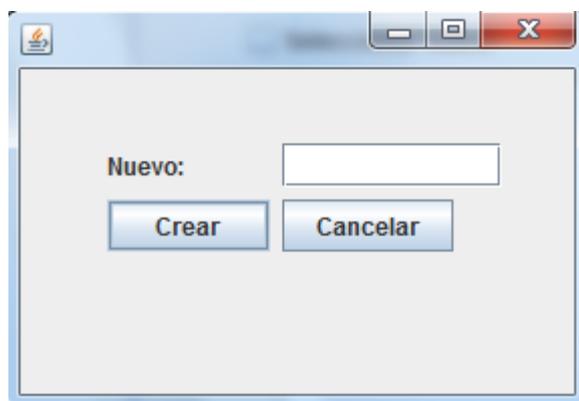
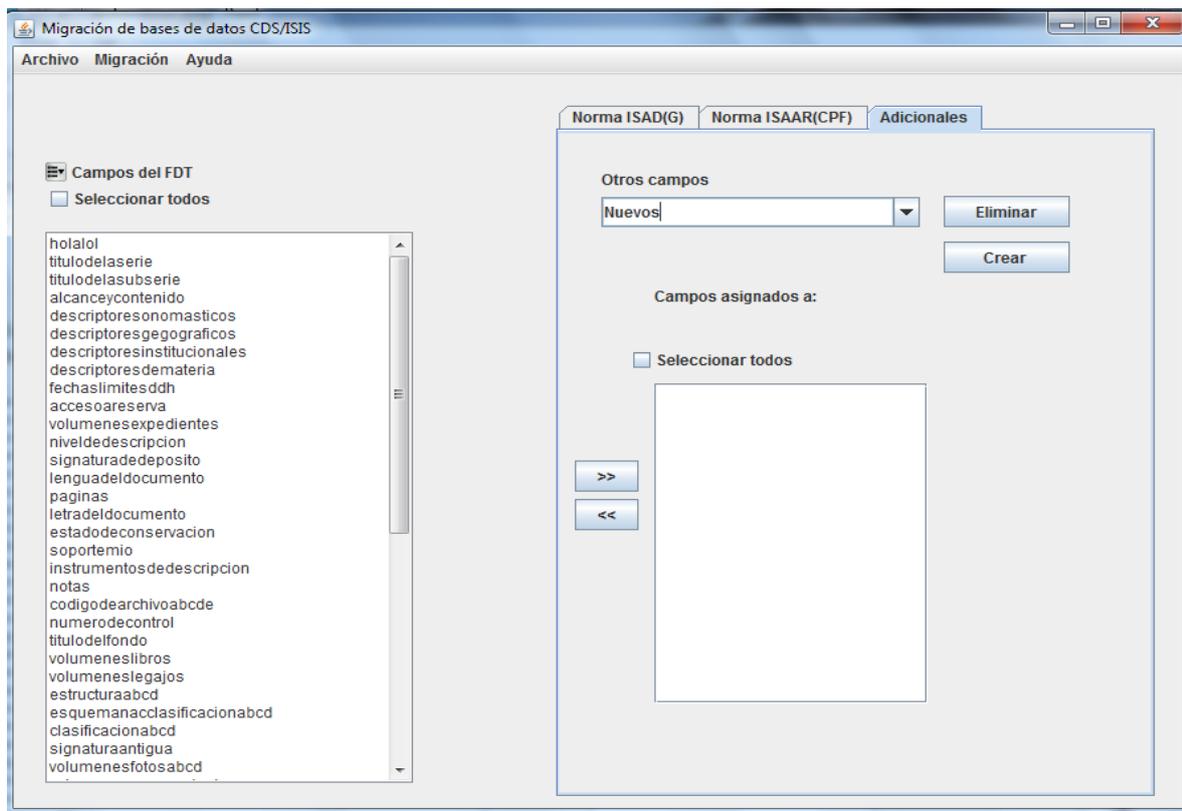


Tabla 15: Descripción del caso de uso Gestionar campo adicional

Objetivo	Abrir los datos de un proyecto de migración que haya sido guardado con anterioridad.	
Actores	Migrador	
Resumen	El usuario selecciona la opción abrir proyecto, el sistema muestra una interfaz para seleccionar el fichero a abrir, se abre el fichero y se muestran los datos del mismo.	
Complejidad	Baja	
Prioridad	Opcional	
Precondiciones	Debe abrirse un fichero que tenga la extensión wpr.	
Postcondiciones	Se deben mostrar los datos del proyecto abierto.	
Flujo de eventos		
Flujo básico <Abrir proyecto de migración>		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona la opción de abrir en el Menú Archivo.	El sistema le muestra una ventana para seleccionar el archivo a abrir.
2.	El usuario selecciona el archivo a abrir.	El sistema muestra la interfaz con los datos del proyecto que fueron guardados.
Flujos alternos		
Nº Evento <2a El archivo seleccionado no es válido>		
	Actor	Sistema
1.		El sistema muestra el siguiente mensaje de error: "El fichero seleccionado no es válido".
Prototipo de Interfaz de usuario		

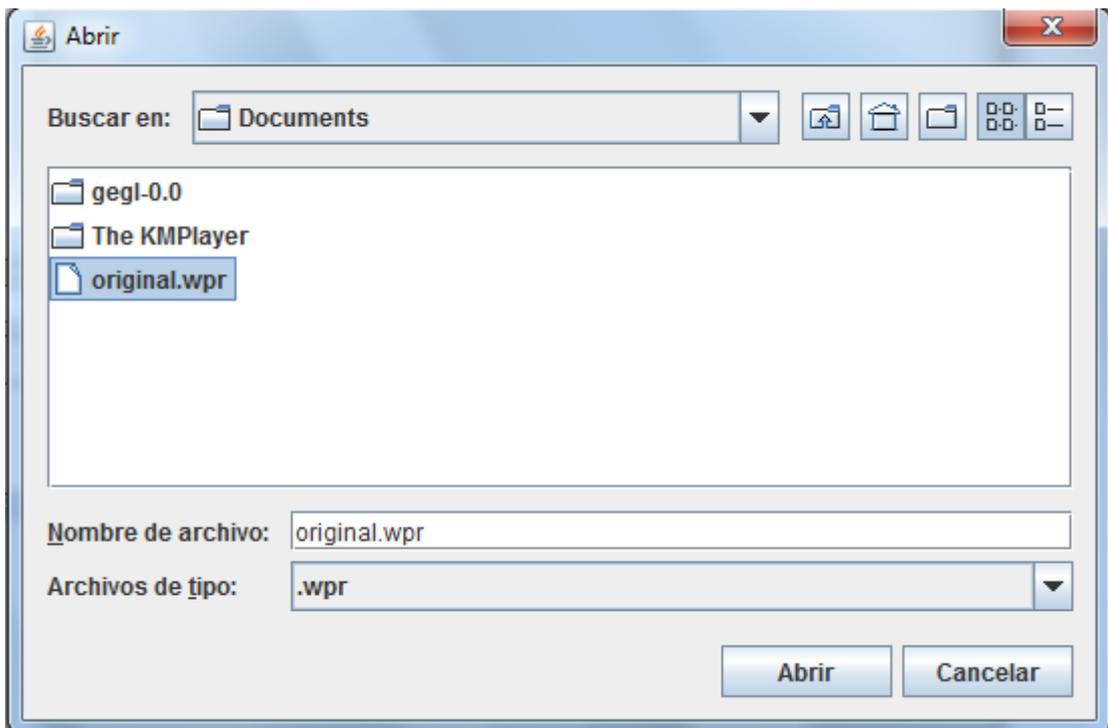
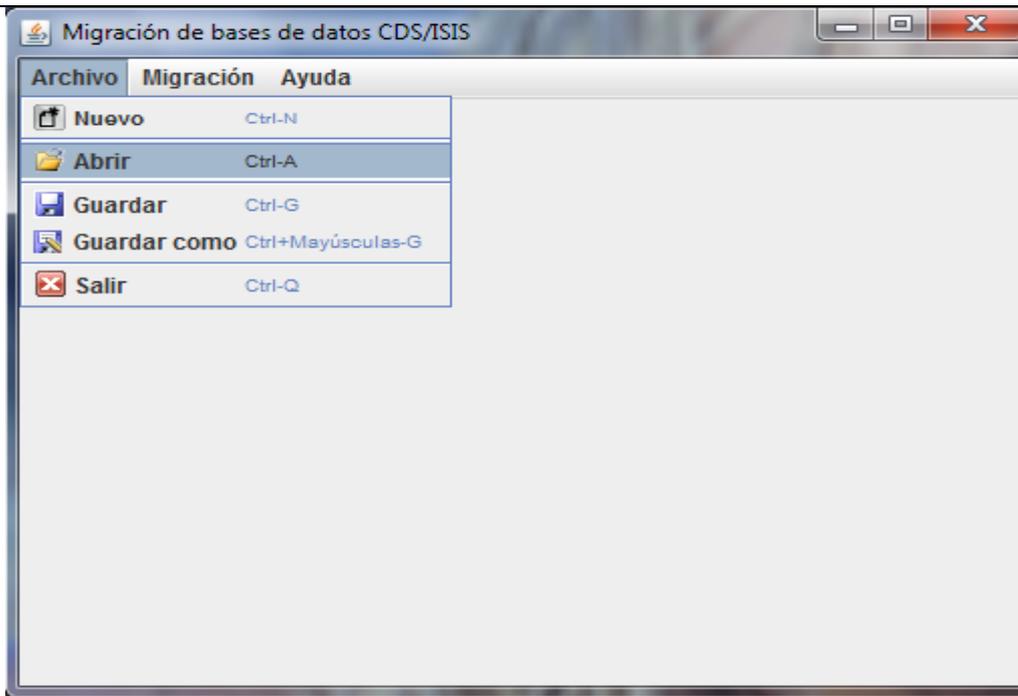


Tabla 16: Descripción del caso de uso Abrir proyecto de migración

Objetivo	Guardar los datos del proyecto de migración en el que se está
-----------------	---

	trabajando.	
Actores	Migrador	
Resumen	El usuario selecciona la opción guardar proyecto, el sistema muestra una interfaz para escribir el nombre del archivo a guardar y se guardan los datos del mismo.	
Complejidad	Baja	
Prioridad	Opcional	
Precondiciones	Se debe abrir el archivo FDT y realizar la asignación de campos.	
Postcondiciones	Se guardan los datos del proyecto abierto.	
Flujo de eventos		
Flujo básico <Guardar proyecto de migración>		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor selecciona la opción de guardar en el Menú Archivo.	El sistema le muestra una ventana para escribir el nombre del archivo a guardar.
2.	El usuario especifica el nombre del archivo a guardar y la dirección donde será guardado.	El sistema guarda los datos en un archivo con la extensión wpr en la dirección especificada con el nombre escrito.
Prototipo de Interfaz de usuario		

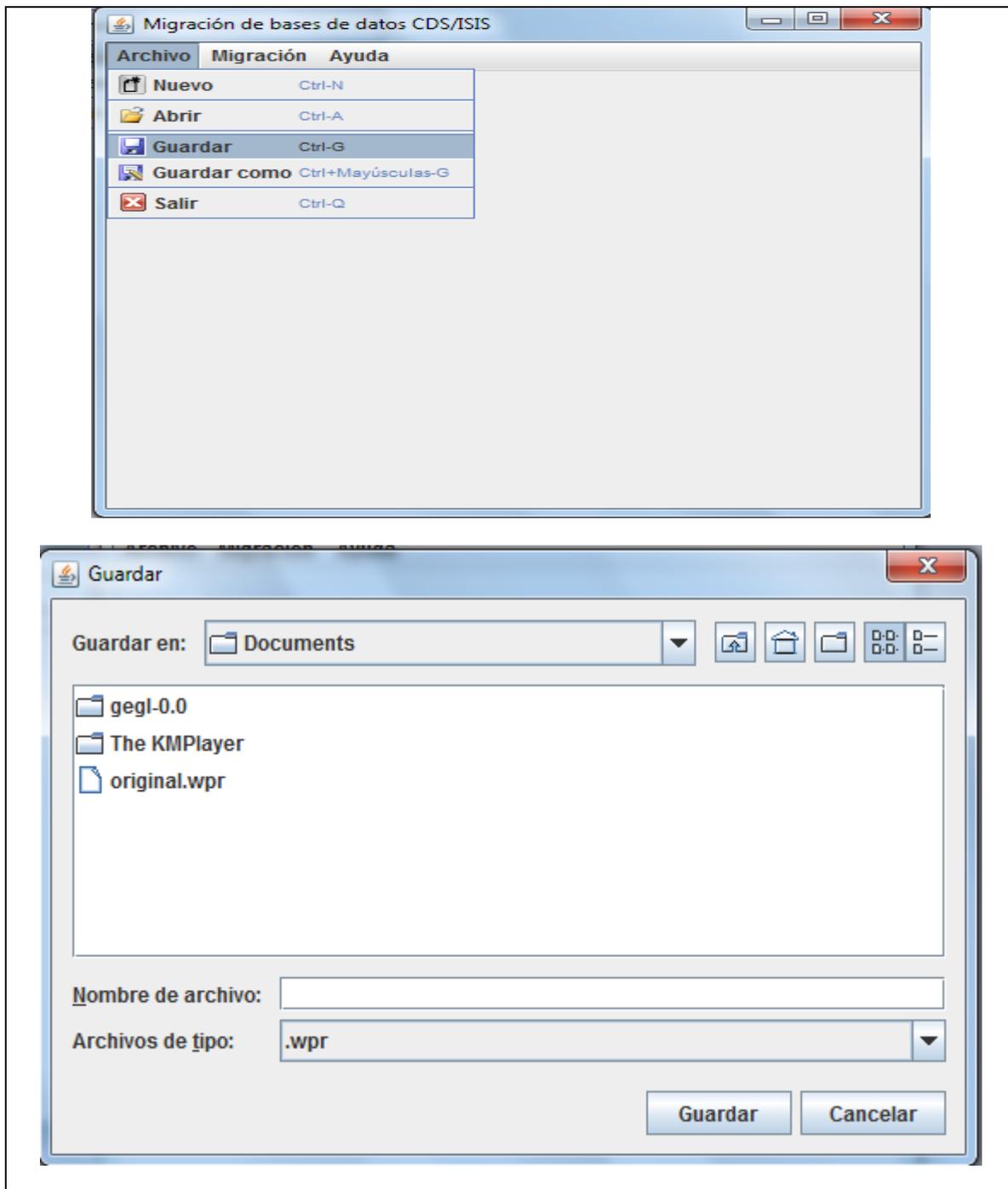


Tabla 17: Descripción del caso de uso Guardar proyecto de migración

Anexo 2: Prototipo de interfaz de usuario del caso de uso Abrir archivo FDT.

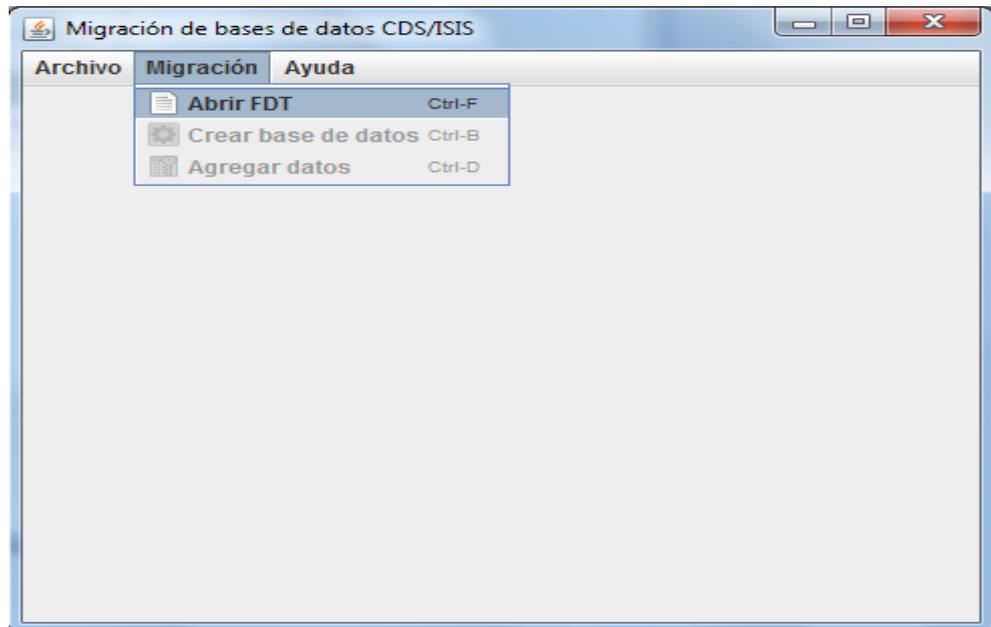


Figura 18: Prototipo de interfaz de usuario del caso de uso Abrir archivo FDT

Anexo 3: Prototipo de interfaz de usuario del caso de uso Gestionar asignación de campos ISAD (G).

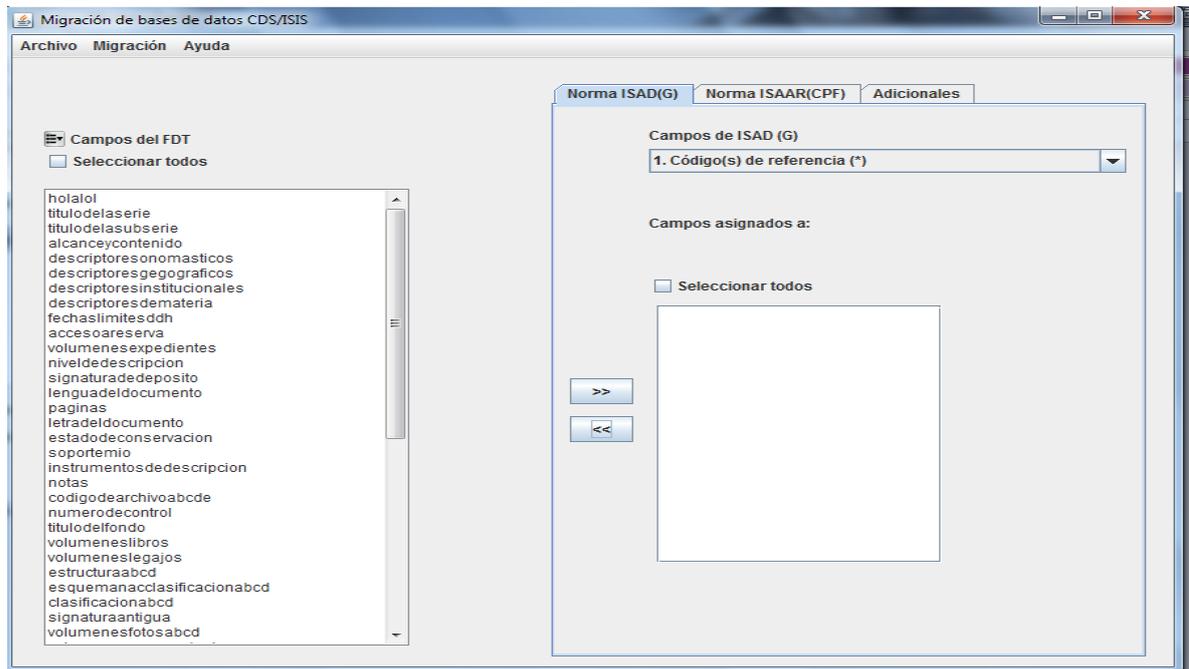
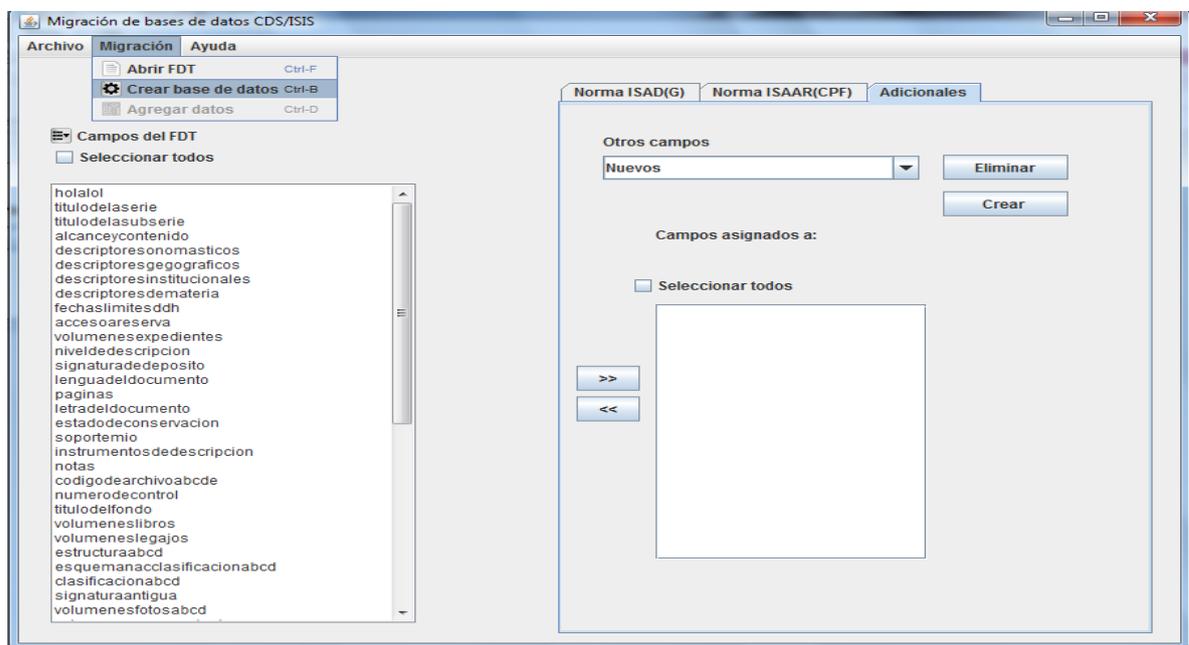


Figura 19: Prototipo de interfaz de usuario del caso de uso Gestionar asignación de campos ISAD (G)

Anexo 4: Prototipo de interfaz de usuario del caso de uso Crear BD.



Servidor (*) localhost

Base de Datos (*) fuentefondoserie

Usuario (*) root

Contraseña

Puerto (*) 3306

MySQL PostgreSQL

Aceptar

Figura 20: Prototipo de interfaz de usuario del caso de uso Crear BD

Anexo 5: Prototipo de interfaz de usuario del caso de uso Agregar datos.

Migración de bases de datos CDS/ISIS

Archivo Migración Ayuda

- Abrir FDT Ctrl-F
- Crear base de datos Ctrl-B
- Agregar datos Ctrl-D

Campos del FDT

Seleccionar todos

hola!ol

titulodelaserie

titulodelasubserie

alcanceycontenido

descriptoresonomasticos

descriptoresgegograficos

descriptoresinstitucionales

descriptoresdemateria

fechaslimitesddh

accesoareserva

volumenes expedientes

nivelde descripcion

signaturadedeposito

lenguadeldocumento

paginas

letradeldocumento

estadodeconservacion

soportemio

instrumentosde descripcion

notas

codigodearchivoabcde

numero de control

titulodelfondo

volumeneslibros

volumeneslegajos

estructuraabcd

esquemanacclasificacionabcd

clasificacionabcd

signaturaantigua

volumenesfotosabcd

Norma ISAD(G) Norma ISAAR(CPF) Adicionales

Otros campos

Nuevos

Eliminar

Crear

Campos asignados a:

Seleccionar todos

>>

<<

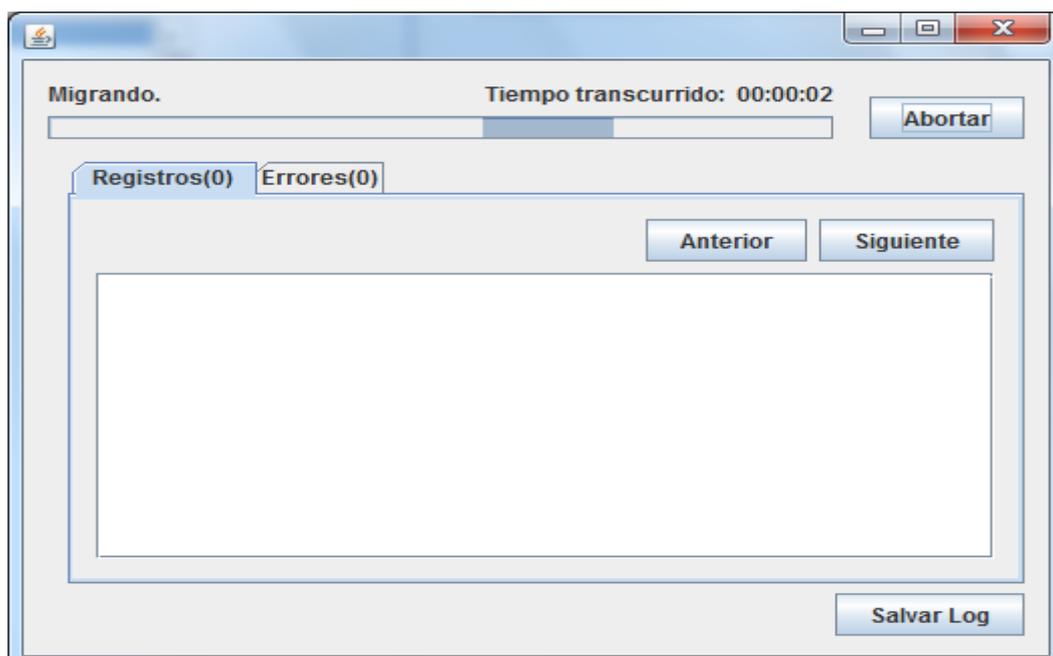
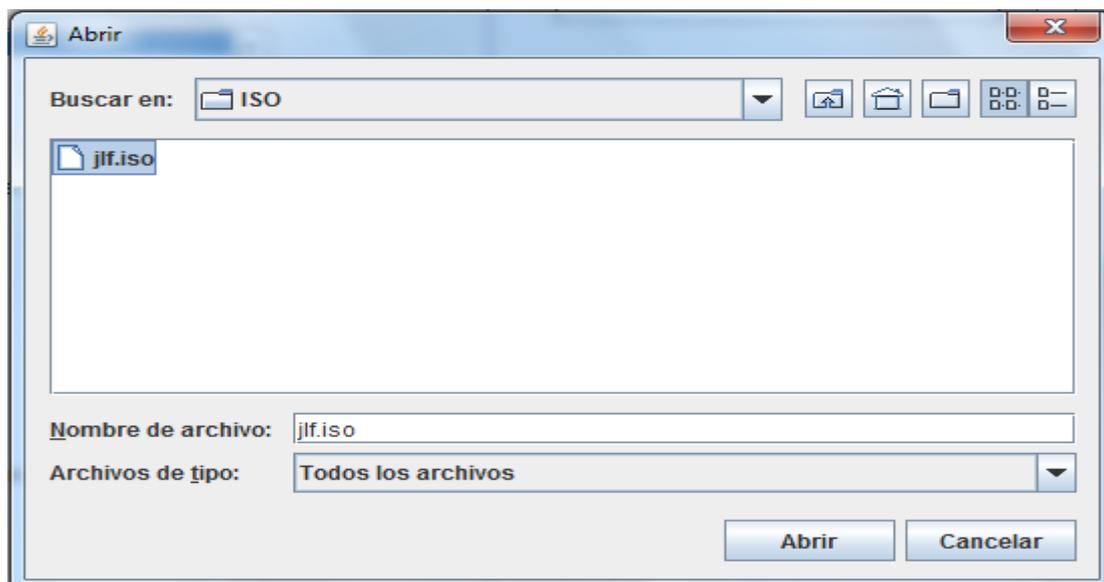
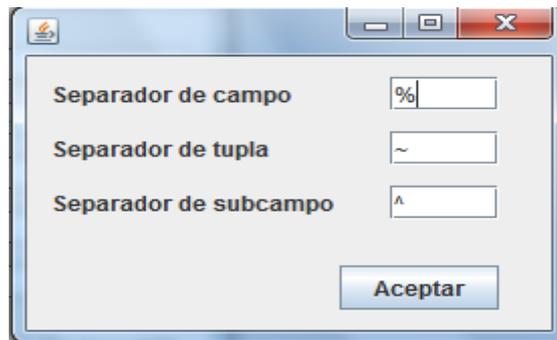


Figura 21: Prototipo de interfaz de usuario del caso de uso Agregar datos

Anexo 6: Diagramas de interacción.

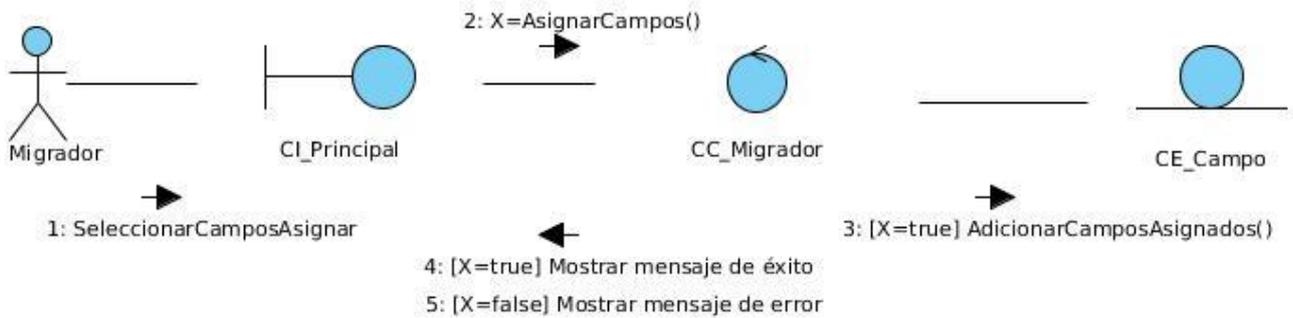


Figura 22: Diagrama de interacción CU: Gestionar asignación de campos ISAAR (CPF) –SC1 Asignar campos ISAAR (CPF)

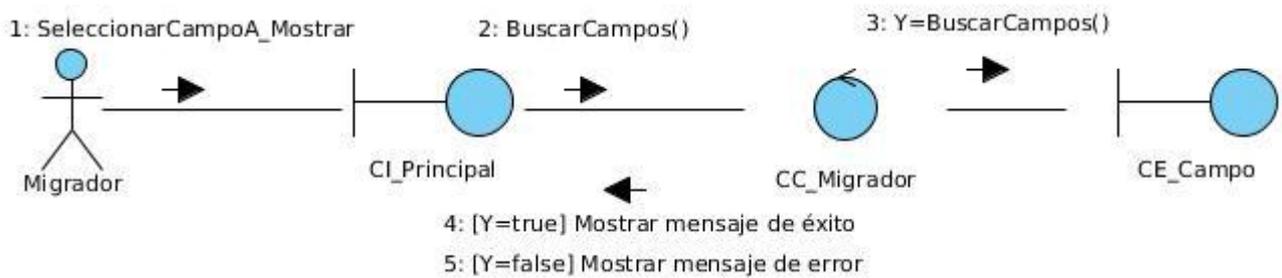


Figura 23: Diagrama de interacción CU: Gestionar asignación de campos ISAAR (CPF) –SC2 Mostrar asignación de campos ISAAR (CPF)

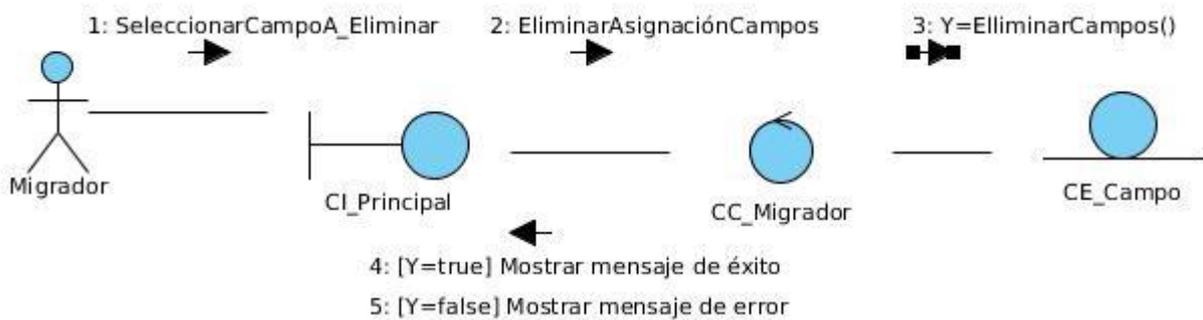


Figura 24: Diagrama de interacción CU: Gestionar asignación de campos ISAAR (CPF) –SC3 Eliminar asignación de campos ISAAR (CPF)

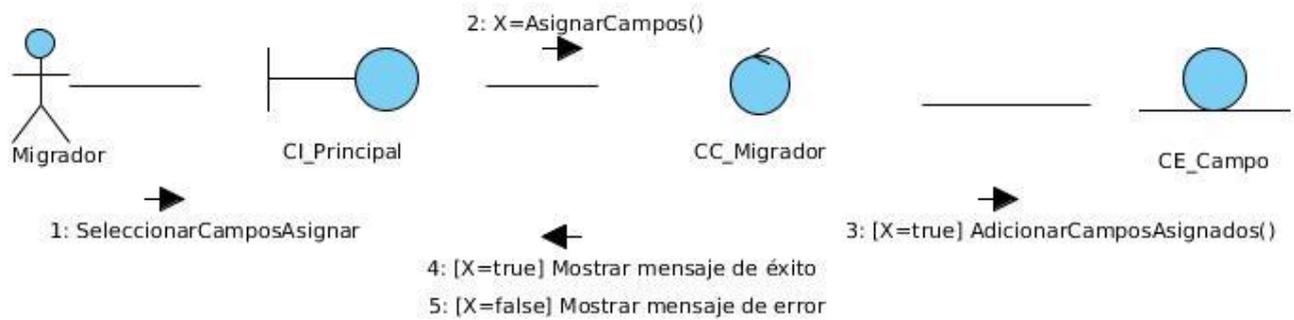


Figura 25: Diagrama de interacción CU: Gestionar asignación de campos adicionales –SC1 Asignar campos adicionales

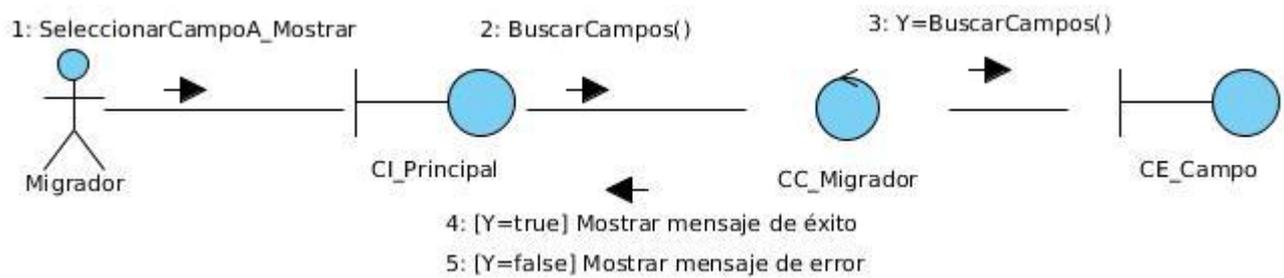


Figura 26: Diagrama de interacción CU: Gestionar asignación de campos adicionales –SC2 Mostrar asignación de campos adicionales

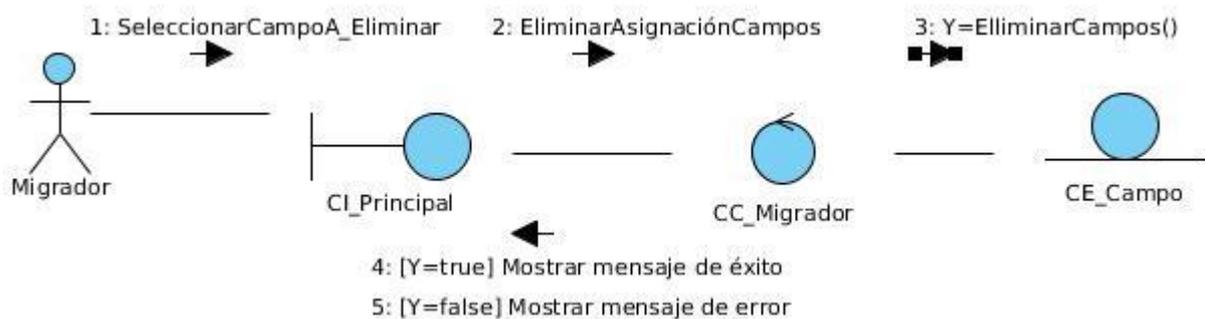


Figura 27: Diagrama de interacción CU: Gestionar asignación de campos adicionales –SC3 Eliminar asignación de campos adicionales

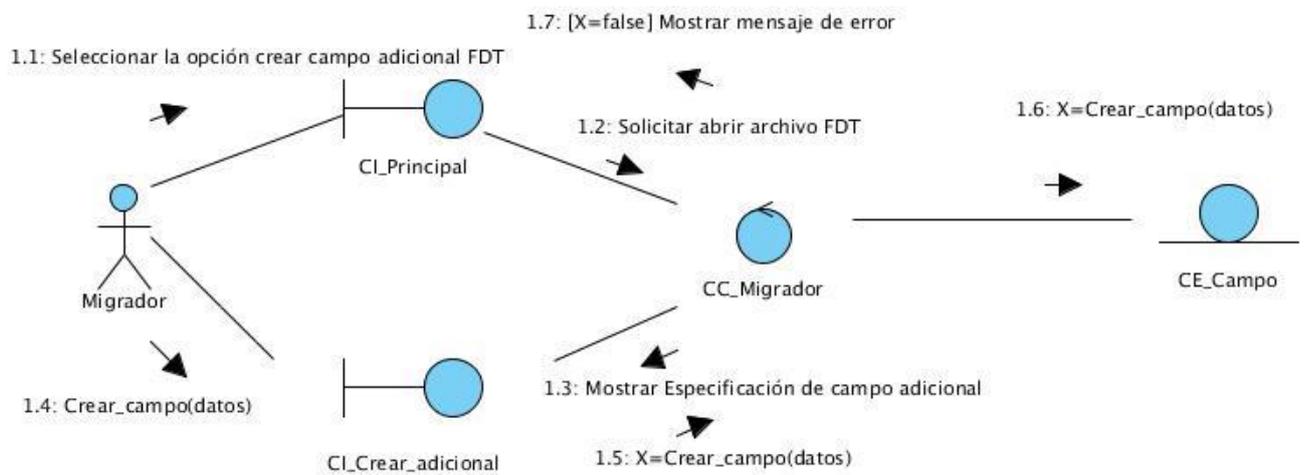


Figura 28: Diagrama de interacción del CU: Gestionar campo adicional adicionales –SC1 Adicionar campo

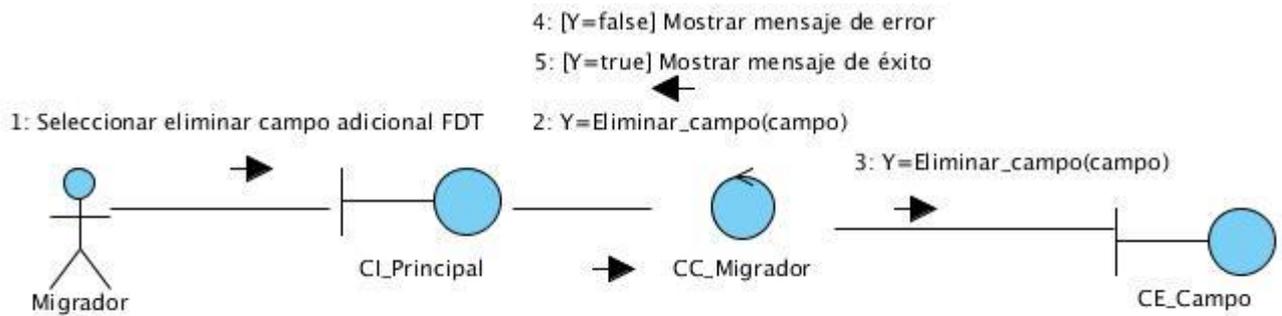


Figura 29: Diagrama de interacción del CU: Gestionar campo adicional adicionales –SC2 Eliminar campo

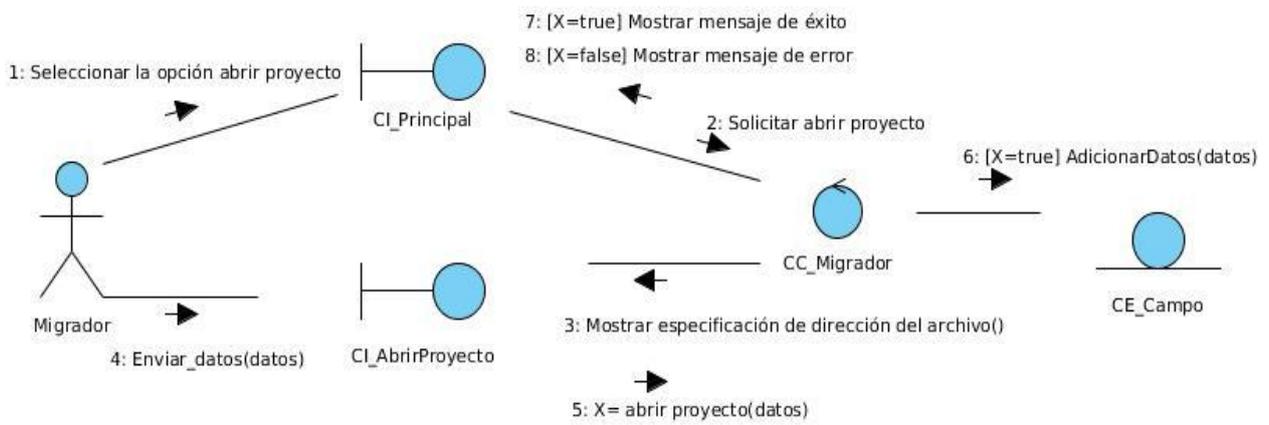


Figura 30: Diagrama de interacción del CU: Abrir proyecto de migración

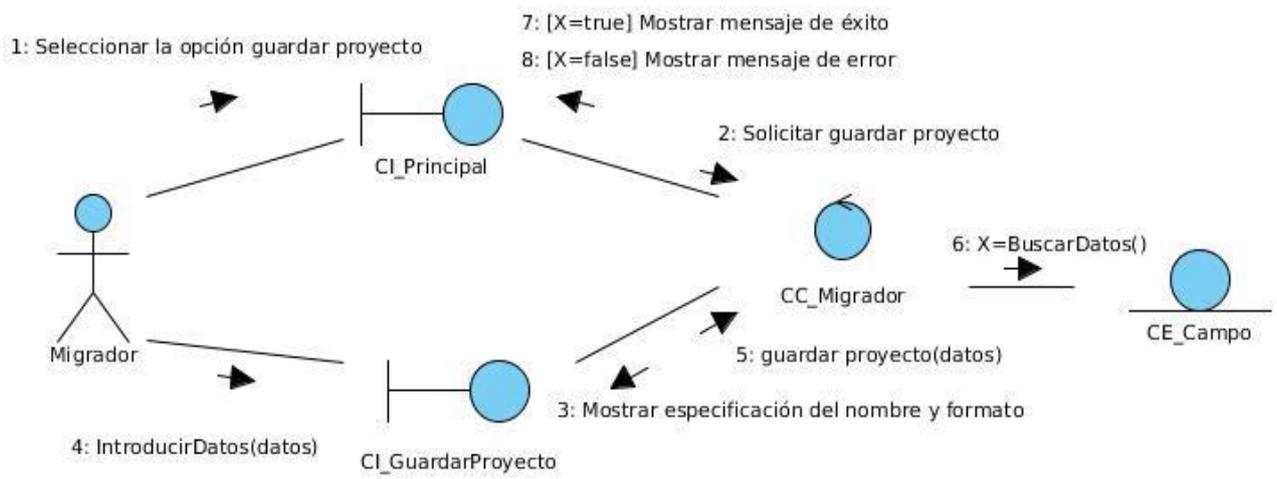


Figura 31: Diagrama de interacción del CU: Guardar proyecto de migración

Anexo 7: Diagramas de clases del diseño.

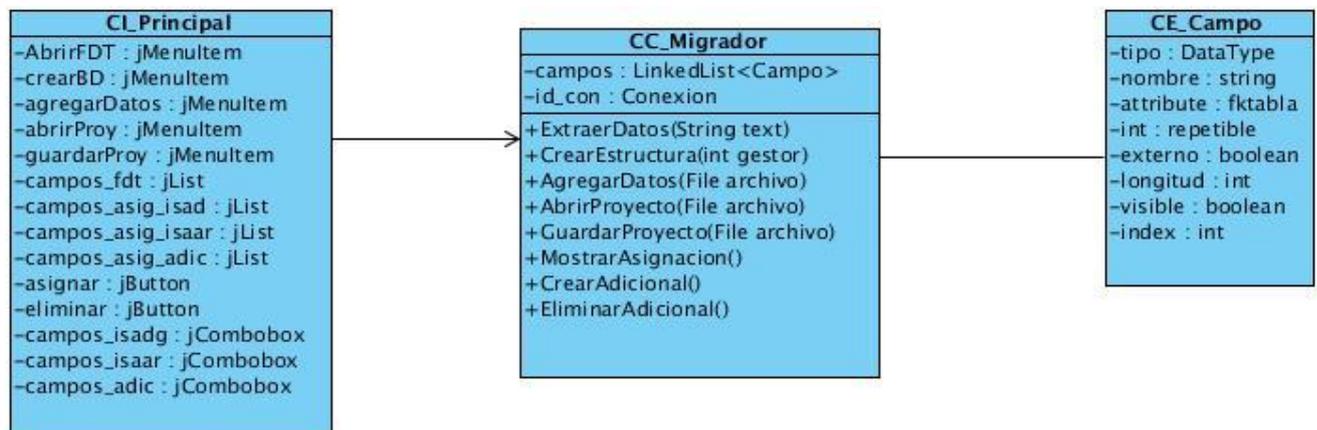


Figura 32: Diagrama de clases del diseño del CU: Gestionar asignación de campos ISAAR (CPF)

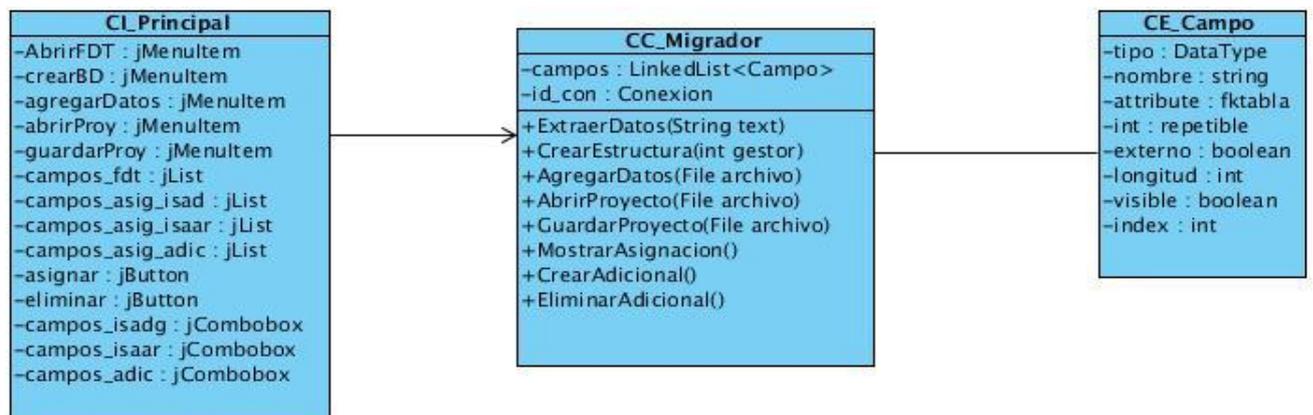


Figura 33: Diagrama de clases del diseño del CU: Gestionar asignación de campos adicionales

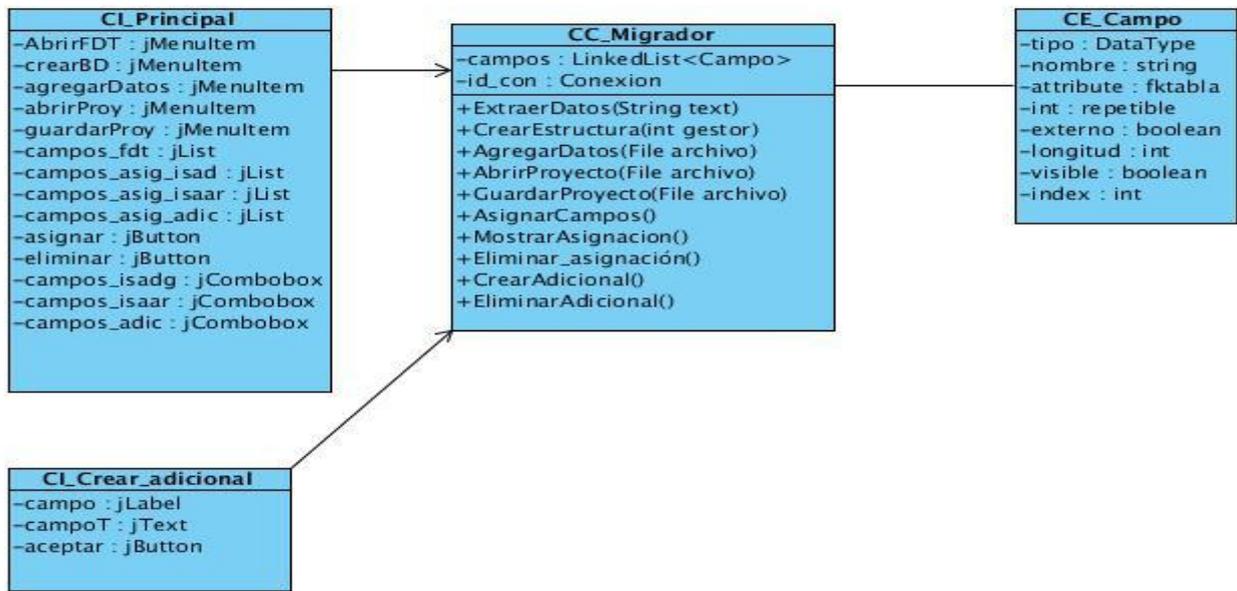


Figura 34: Diagrama de clases del diseño del CU: Gestionar campo adicional

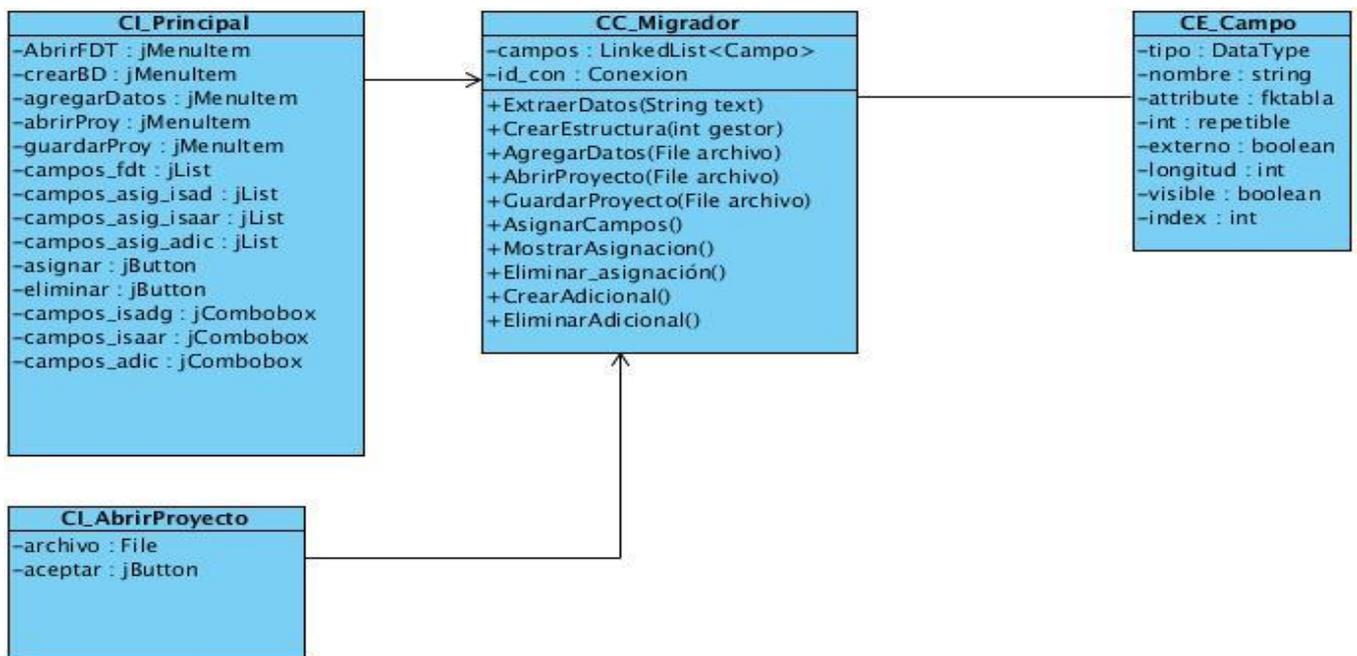


Figura 35: Diagrama de clases del diseño del CU: Abrir proyecto de migración

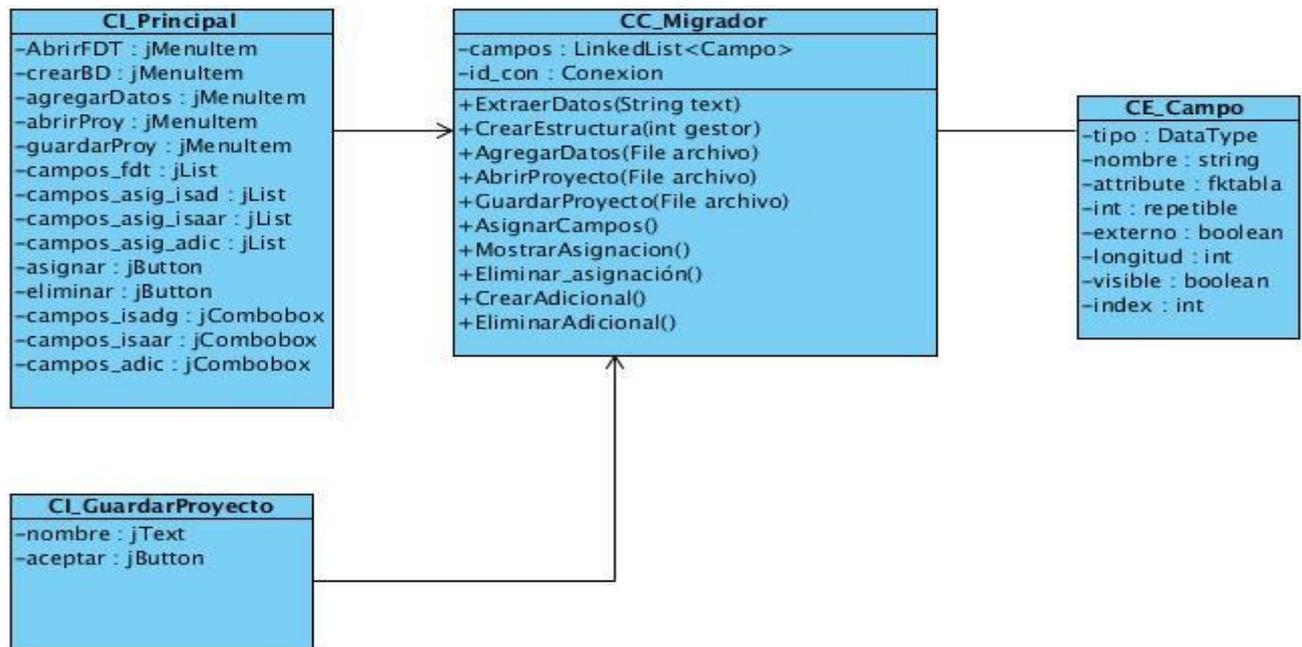


Figura 36: Diagrama de clases del diseño del CU: Guardar proyecto de migración