Práctica empresarial: análisis, diseño, desarrollo e implementación de mejoras en el Repositorio Institucional UIS relacionadas con la migración de trabajos de grado e integraciones para mayor visibilidad de la producción científica y académica.

Samuel Yesid Cadena Pinilla y Sergio Raul Roa Ortiz

Trabajo de Grado para Optar al Título de Ingeniero de Sistemas

Director

Ferney Mauricio Calderón

MBA. Magister en Gerencia de Negocios

Codirector

Lola Xiomara Bautista Rozo

PhD. en Ciencias y Tecnologías de la Comunicación y la Información

Universidad Industrial de Santander

Facultad de Ingenierías Físico-Mecánicas

Escuela de Ingeniería de Sistemas e Informática

Ingeniería de Sistemas

Bucaramanga

2023

Dedicatoria

Dedicatoria Sergio Raul Roa Ortiz

A Dios, por brindarme la fuerza y paciencia necesaria para poder cumplir esta meta, superando las dificultades y obstáculos que surgieron en el camino.

A mis padres, Raul Antonio y María Elena, por sus esfuerzos para darme siempre lo mejor, por el amor y el apoyo incondicional recibido en todo momento. A mi hermana Melissa por ser un ejemplo a seguir, gracias por el apoyo y acompañamiento recibido. A mi novia por estar siempre ahí, por todo su amor y cariño.

"Empecé temprano y me quedé hasta tarde, día tras día, año tras año, me tomó 17 años y 114 días convertirme en un éxito de la noche a la mañana."

Lionel Messi.

Dedicatoria

Dedicatoria Samuel Yesid Cadena Pinilla

En esta dedicatoria quiero honrar a dos partes fundamentales de mi vida, Dios y mis padres, que son bendiciones que han ayudado a encontrar un camino y llenado mi corazón de alegrías y amor.

A Dios, quien me ha guiado en cada paso, por permíteme alcanzar este nuevo logro, porque me ha dado fortaleza en momentos de debilidad y la perseverancia para realizar este sueño.

A mi familia, en especial a mis padres, quienes me han brindado todo su cariño y palabras de aliento en momentos difíciles. Agradezco por los momentos compartidos, las lecciones aprendidas y por el amor incondicional que nos une.

En familia hemos afrontado retos, celebrado logros y construido momentos inolvidables. En ustedes siempre encuentro confianza y motivación sin importar los obstáculos que se presenten. Son un pilar que sostienen mi vida y son la inspiración que me motiva a sacar la mejor versión de mí mismo.

Agradecimientos

Queremos agradecer a la Universidad Industrial de Santander, nuestra alma mater por todas las enseñanzas, experiencias y buenos momentos allí vividos. También agradecer a la Biblioteca de la Universidad por acogernos durante casi un año para realizar nuestras prácticas. Gracias al profesor Pedro García, director de la Biblioteca, por la contratación y por confiar en nosotros para realizar este proyecto. Gracias a nuestro director Ferney Mauricio Calderón, por la ayuda, la enseñanza, el tiempo, las correcciones y el acompañamiento brindado durante estos nueve meses. Gracias a nuestra codirectora Lola Xiomara Bautista Rozo por el acompañamiento dado.

A nuestros compañeros de práctica en Biblioteca, por su apoyo y guía durante la realización de este proyecto.

Tabla de Contenido

Pág.

Lista de Tablas

Pág.

Lista de Figuras

Pág.

Lista de Apéndices

pág.

Glosario

DIR: ruta especifica de cada documento dentro del algoritmo.

Dspace: software de código abierto para la creación de Repositorios y Bibliotecas digitales.

Dublin Core: es un modelo de metadatos elaborado y auspiciado por DCMI.

LIBRUIS: sistema integrado de gestión de bibliotecas antiguo de la Biblioteca UIS.

Noesis: nombre oficial del Repositorio Institucional de la UIS.

Pandas: es una herramienta de código abierto enfocada en la manipulación y análisis de datos.

Pdf2Image: es una librería de Python utilizada para convertir archivos PDF en imágenes.

Pikepdf: es una librería de Python que permite la creación, manipulación y restauración de PDF's.

PIL: añade capacidades de procesamiento de imágenes a Python.

PyMUPDF: es una librería de Python usada para extraer texto de archivos PDF.

PYPDF2: es una librería de código abierto de Python que trabaja con archivos PDF.

Pytesseract: es una herramienta de reconocimiento óptico de caracteres en Python.

SIGB: sistema integrado de gestión de bibliotecas.

Tqdm: es una librería de Python para mostrar el progreso de la ejecución.

UIS: Universidad Industrial de Santander.

Waterfall: metodología para el desarrollo secuencial de tareas.

workdir: ruta principal desde donde se procesan los documentos en el algoritmo.

Resumen

Título: Práctica empresarial: análisis, diseño, desarrollo e implementación de mejoras en el Repositorio Institucional UIS relacionadas con la migración de trabajos de grado e integraciones para mayor visibilidad de la producción científica y académica.[*]

Autores: Samuel Yesid Cadena Pinilla y Sergio Raul Roa Ortiz[**]

Palabras Clave: Sistema de Información, Repositorio Institucional, Migración, Integración, SIGB, LIBRUIS, ORCID, Dspace.

Descripción: La Biblioteca de la Universidad Industrial de Santander ha gestionado sus recursos bibliográficos mediante el sistema de desarrollo propio llamado LIBRUIS desde 1979 hasta la actualidad. Una de las colecciones más importantes son los trabajos de grado, tesis y disertaciones, las cuales eran catalogadas y clasificadas en este sistema y puestos a disposición mediante el Catálogo Bibliográfico. Con el auge de los Repositorios Institucionales para recolectar, resguardar y difundir la producción científica y académica de las universidades, desde la Biblioteca se han implementado tres versiones (2013, 2018 y 2022) basadas en el software de acceso abierto DSpace; pero al intentar migrar la colección de trabajos de grados del catálogo al repositorio, se ha detectado que por motivos de violación de seguridad en los datos personales, de acuerdo con las leyes de privacidad y derechos de autor, estos documentos contienen firmas de estudiantes y profesores, las cuales no pueden estar públicas en la red.

En la presente práctica empresarial se realizan mejoras al Repositorio institucional – Noesis, relacionadas con la migración de los documentos de interés, con la estructura requerida de los metadatos y el procesamiento de los archivos, contemplando el cumplimiento de los derechos de autor, así como la definición de requerimientos necesarios para la integración con redes de colaboración que mejoran la visibilidad de los recursos.

---

[*] Trabajo de Grado

[**] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Ingeniería de Sistemas. Director: Ferney Mauricio Calderón. MBA. Magister en Gerencia de Negocios. Codirector: Lola Xiomara Bautista Rozo. PhD. en Ciencias y Tecnologías de la Comunicación y la Información

Abstract

Title: Business practice: analysis, design, development, and implementation of improvements in the UIS Institutional Repository related to degree work migration and integrations for greater visibility of scientific and academic production.[*]

Author(s): Samuel Yesid Cadena Pinilla and Sergio Raul Roa Ortiz[**]

Key Words: Information System, Institutional Repository, Migration, Integration, SIGB, LIBRUIS, ORCID, Dspace.

Description: The Universidad Industrial de Santander Library has managed its bibliographic resources through the self-developed system called LIBRUIS from 1979 to the present. One of the most important collections are the degree works, thesis, and dissertations, which were cataloged and classified in this system and made available through the Bibliographic Catalog. Due to the rise of Institutional Repositories to collect, store and disseminate the scientific and academic production of universities, the Library has implemented three versions (2013, 2018, and 2022) based on the open-access software DSpace; however, when trying to migrate the degree work collection from the catalog to the repository, it has been detected that for reasons of security violation of personal data, according to privacy and copyright laws, these documents contain signatures of students and professors, which cannot be public on the network.

In this business practice, improvements are made to the Institutional Repository - Noesis, related to the migration of documents of interest, with the required structure of metadata and file processing, considering copyright compliance, as well as the definition of requirements necessary for integration with collaborative networks that improve the visibility of resources.

[*] Degree Work

[**]Faculty of Physical-Mechanical Engineering. School of Systems and Informatics Engineering. Systems Engineering. Director: Ferney Mauricio Calderón. MBA. Master in business management. Co-director: Lola Xiomara Bautista Rozo. PhD. in Communication and Information Sciences and Technologies

Introducción

En la actualidad, gran cantidad de universidades de todo el mundo utilizan repositorios institucionales para recolectar, preservar y difundir la información referente a su producción científica y académica. En la Universidad Industrial de Santander este tema se ha venido trabajando mediante proyectos de gestión de la Biblioteca donde se realizaba análisis y configuración de la herramienta, junto a gestión con otras unidades de la Universidad involucradas para su correcto funcionamiento, hasta concluir todas estas labores con la implementación de Noesis, el Repositorio Institucional donde se almacenan diferentes colecciones de producción intelectual UIS, con el fin de resguardar y darle visibilidad e impacto ante la comunidad científica. Noesis funciona sobre un software de acceso abierto llamado DSpace, el cual es utilizado para la administración de colecciones digitales y en la mayoría de los casos como repositorio bibliográfico institucional.

Antes de la llegada de Noesis, toda la información relacionada con la producción intelectual de los estudiantes, representada en trabajos de grado de pregrado y posgrado, se almacenaba en el Sistema de Gestión de Biblioteca de desarrollo propio de la Universidad, LIBRUIS, y se accedía mediante el Catálogo Bibliográfico disponible en la página web de la Biblioteca, para la consulta de la comunidad universitaria. En el año 2013 se implementó por primera vez el Repositorio Institucional UIS y luego en el año 2018 se realizó un nuevo montaje, se lograron migrar masivamente los datos referenciales que se pudieron recuperar de LIBRUIS, junto a los documentos PDF existentes de los trabajos de grado que se encontraban en formato digital, desde el 2004 hasta ese momento; pero al entrar en producción y empezar a recuperar esta

información desde metabuscadores como Google, se descubre que hay una violación a las leyes de privacidad y derechos de autor, ya que dentro de los documentos de trabajo de grado se incluían las páginas con la nota del proyecto y la autorización de publicación, las cuales contenían información personal tanto de los autores como de los directores y evaluadores. Esta información corresponde a firmas personales que podían ser consultadas por cualquier persona sin necesidad de validación, de acuerdo con las políticas oficiales de los repositorios que establecen que su información se encuentre en acceso abierto. Teniendo en cuenta lo anterior y debido a que varios egresados manifestaron su inconformidad, se debieron retirar estos documentos del Repositorio y se decidió que a partir del año 2022 se reciben estos documentos en archivos separados para poder dejar disponible solo el trabajo de grado y poner restricción de acceso a los otros dos archivos (nota del proyecto, carta de autorización de publicación).

Teniendo en cuenta lo anterior, nació la necesidad de iniciar la labor de extraer estas páginas de cada uno de los documentos, con el fin de poder migrar de forma correcta los trabajos de grado presentados del 2004 al 2021, desde LIBRUIS a Noesis. Esta labor inicialmente se realizó con apoyo de algunos funcionarios de Biblioteca, que debían extraer manualmente la información referencial de los metadatos y separar los archivos correspondientes, para después migrarlos por lotes. Se inicio realizando pruebas con los trabajos de grado de posgrados en doctorados y maestrías, pero se observó que realizándolo de esta forma era un trabajo extenso que llevaría mucho tiempo y esfuerzo, además se requería con urgencia ya que la Biblioteca actualmente se encuentra en proceso de actualización de su sistema de gestión y es posible que LIBRUIS deje de funcionar en un futuro cercano. Por lo anterior, se tenía la necesidad de implementar un software que pueda extraer algunas páginas del documento y crear nuevos archivos PDF para luego

parametrizar la información referencial de los metadatos y poder realizar nuevamente una migración masiva sin tener inconvenientes. Finalmente, para optimizar las opciones de visibilidad del contenido disponible en Noesis, se revisaron los requerimientos necesarios para la integración y compatibilidad con el perfil de investigador ORCID y redes de colaboración para la visibilidad de producción científica y académica, como la Red Colombiana de Información Científica (REDCOL), La Referencia a nivel de América Latina o Google Académico.

1. Generalidades

1.1 Modalidad – Trabajo de grado

El presente trabajo de grado se realizó bajo la modalidad de práctica empresarial en la Biblioteca central de la Universidad Industrial de Santander. Fue avalado por la Escuela de Ingeniería de Sistemas como opción para desarrollar el proyecto de grado y optar por el título de Ingeniero de Sistemas.

1.2 Sobre la Biblioteca de la Universidad Industrial de Santander

La Biblioteca UIS en su actualización de sistema integrado de información, plantea un mejoramiento de los sistemas de información el cual abre la posibilidad de realizar algunas prácticas empresariales donde se tratan temas como el análisis de datos de la Biblioteca Virtual, análisis de datos histórico del Catálogo Bibliográfico y mejoras en el Repositorio Institucional Noesis, relacionadas con la migración de los trabajos de grado contenidos en el servidor de LIBRUIS y la integración con las redes de colaboración.

1.2.1 Historia

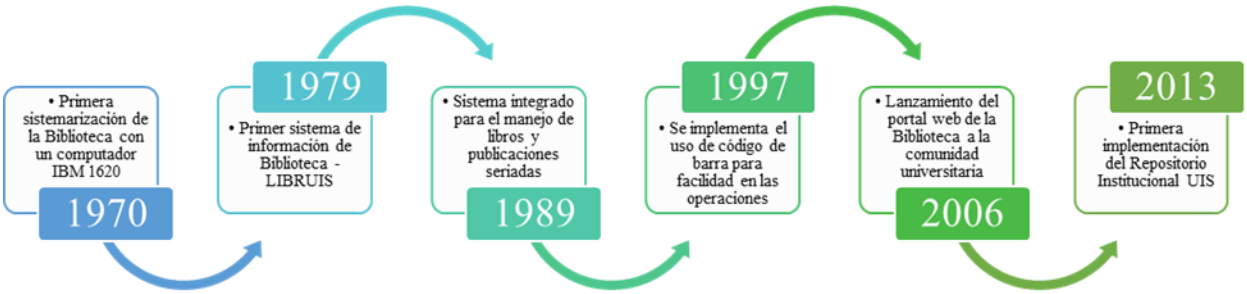
La Biblioteca de la Universidad Industrial de Santander (UIS) es una institución que ha estado presente en el desarrollo académico y cultural de la región desde hace más de 70 años. Fue creada en 1948, al mismo tiempo que la fundación de la Universidad Industrial de Santander. En sus primeros años, la Biblioteca se estableció en las instalaciones del Instituto Técnico Superior Dámaso Zapata, pero en el año 1976 se trasladó a su actual sede en el Edificio de la Biblioteca Central, ubicado en el campus principal de la Universidad Industrial de Santander en Bucaramanga, Colombia.

En la actualidad, la Biblioteca de la UIS es una de las bibliotecas universitarias más importantes del país, cuenta con un acervo bibliográfico de más de 100.000 volúmenes y ofrece servicios y recursos en línea para toda la comunidad académica, incluyendo estudiantes, profesores e investigadores. Esta institución es vital en el panorama académico y cultural de la región, y ha contribuido significativamente al desarrollo y la formación de varias generaciones de profesionales y líderes en diferentes áreas del conocimiento.

Desde la perspectiva tecnológica, la Biblioteca ha venido presenciando una serie de cambios a través del tiempo, que requieren de actualización en sus sistemas de información para poder adaptarse a las exigencias, estar a la vanguardia para brindar una mejor experiencia a los usuarios y poder seguir contribuyendo a la investigación y educación del país. A continuación, se pueden observar algunos eventos importantes en cuanto a su avance tecnológico:

Figura 1

Línea de tiempo factor tecnológico de la Biblioteca UIS

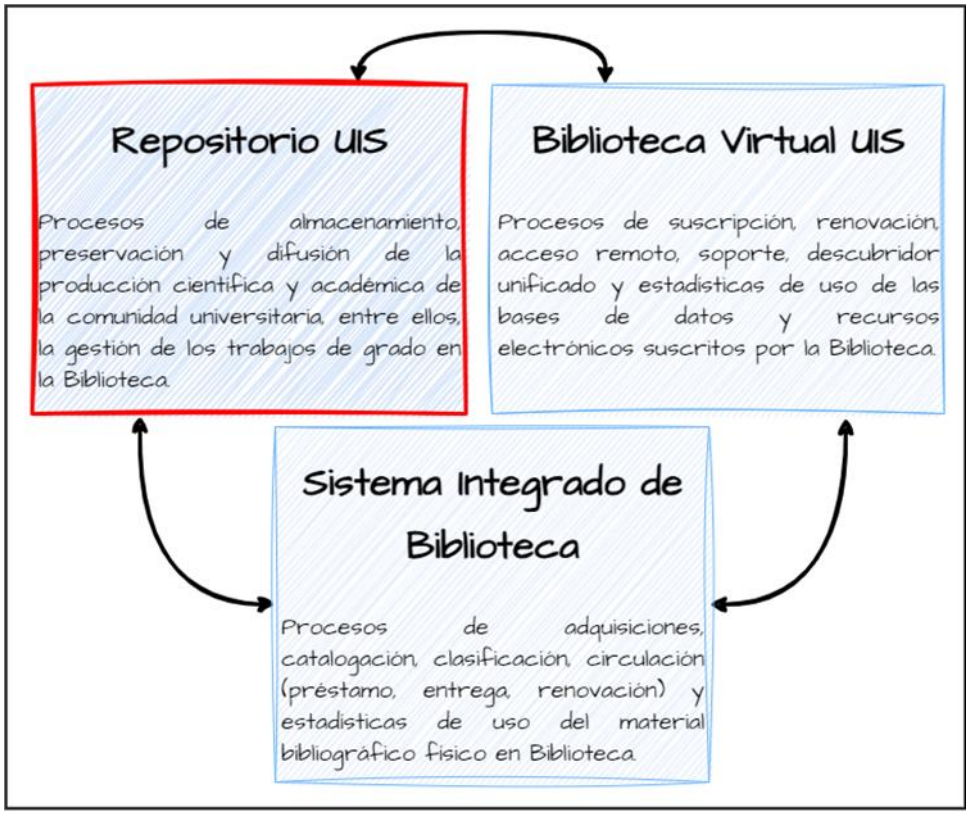


1.2.2 Sistemas de Información

Actualmente la Biblioteca UIS cuenta con sistemas de información que se encargan de gestionar los recursos disponibles para la comunidad universitaria, como las colecciones de material bibliográfico, las bases de datos, trabajos de grado y otros. El presente proyecto está enfocado en el Repositorio Institucional UIS, señalado de color rojo en la siguiente figura:

Figura 2

Sistemas de información de la Biblioteca UIS



1.2.3 Misión

Ser un centro integral de información capaz de satisfacer y anticiparse a las necesidades de documentación de la comunidad universitaria, académica e investigativa a nivel regional, nacional e internacional, mediante la prestación de servicios de adquisición, procesamiento, recuperación y diseminación de información con criterios de calidad. Para ello se apoya en la utilización de tecnología moderna y talento humano idóneo, constituyéndose de esta forma en líder del desarrollo y promoción de actividades intelectuales que estimulen procesos de enseñanza y aprendizaje.

1.2.4 Visión

La Biblioteca de la Universidad Industrial de Santander será un sistema conectado a la red mundial de información, mediante una infraestructura digital que permita nuevas formas de conocimiento que contribuyan a la formación integral de sus usuarios. Así mismo, se espera lograr un posicionamiento local, regional e internacional para ofrecer servicios abiertos, dinámicos y oportunos, como soporte principal a la academia e investigación. El concurso de un equipo humano interdisciplinario, competente y comprometido con la institución, además de la utilización de una metodología innovadora, serán factores vitales para lograr un ambiente adecuado y garantizar la calidad de sus servicios.

1.3 Planteamiento y justificación del problema

La Biblioteca es la unidad encargada de gestionar los servicios de información necesarios para apoyar los procesos de enseñanza, investigación y extensión en la Universidad Industrial de Santander. Dentro de su estructura tecnológica se encuentra la administración del Repositorio Institucional UIS - Noesis, conocido como un espacio de acceso abierto en línea, que tiene el objetivo de almacenar, preservar y difundir la producción científica y académica de la comunidad universitaria, como trabajos de grado, tesis, disertaciones, revistas UIS, producción científica y editorial, material didáctico y multimedia de profesores, eventos académicos y culturales, entre otros. Aunque este sistema es de ámbito institucional e involucra varios roles de la Universidad, la Biblioteca especialmente ha sido la encargada de la producción intelectual de los estudiantes representada en trabajos de grado (pregrado y especializaciones) y tesis (maestría y doctorado) desde la primera promoción de la Universidad hasta la actualidad, disponible en varios formatos:

- 1952 a 2003 se recibieron en formato físico y aunque se encuentran en proceso de digitalización, solo pueden ser consultados dentro de las redes de la Universidad, ya que no se contemplaba solicitar el permiso de publicación en formato digital.
- 2004 al 2019 se recibieron en CD´s, formato digital, con la carta de autorización de publicación firmada por los autores y la nota del proyecto dentro del documento.
- 2020 al 2021 por pandemia se recibieron mediante un formulario virtual, formato digital, con la carta de autorización de publicación firmada por los autores y la nota del proyecto dentro del documento.
- 2022 en adelante se empiezan a recibir mediante autoarchivo en el Repositorio Institucional, formato digital, con la carta de autorización de publicación firmada por los autores y la nota del proyecto en archivos aparte del documento principal.

Previo al lanzamiento oficial de Noesis, en marzo del 2022, los trabajos de grado y tesis eran catalogados y clasificados por el personal de la unidad para ser almacenados en Sistema de Gestión de Bibliotecas LIBRUIS y publicados en el Catálogo Bibliográfico. Esta información, aunque es de acceso abierto y no requiere validación de usuario para su consulta desde cualquier lugar, no es visible mediante metabuscadores generales de internet y requiere el acceso especifico desde el portal de Biblioteca para su búsqueda y recuperación. Contrario a lo que sucede en la actualidad, ya que los autores del proyecto son directamente los encargados de realizar el denominado autoarchivo de los datos y documentos requeridos por la Biblioteca para su publicación en Noesis, el personal de la unidad se encarga de la revisión y una vez avalado son publicados directamente, con la diferencia que el contenido de Noesis si es visible y recuperado desde metabuscadores generales, como Google.

Inicialmente cuando se contempló la posibilidad de la migración masiva a Noesis de los trabajos de grado y tesis en formato digital, especialmente los que contaban con autorización de publicación, es decir en el periodo de 2004 a 2021 con aproximadamente 34.000 registros; se observa que estos documentos al contener la carta de autorización y la nota del proyecto con firmas de los autores, directores y calificadores, estaban generando que esta información quedara libre en internet, lo cual genera inconveniente ya que las firmas son datos personales y privados, que no pueden difundirse sin autorización, de acuerdo con la Ley de Protección de Datos Personales en Colombia, la cual "reconoce y protege el derecho que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bases de datos o archivos que sean susceptibles de tratamiento por entidades de naturaleza pública o privada." (Ley 1581 de 2012).

Es importante mencionar que la información que se encuentra registrada en LIBRUIS no sigue ningún formato de catalogación definido oficialmente, como Marc21 o Dublin Core, por lo cual estos metadatos se encuentran organizados mediante una mezcla de modelos, que generan incongruencias y en ocasiones la carencia de campos necesarios al momento de realizar migración de información a otros sistemas, como el Repositorio. De igual forma, se necesita que esos metadatos tengan un formato específico que permita la estandarización y puedan ser relacionados directamente con Noesis, por lo tanto, se deberá realizar una clasificación de acuerdo con la estructura definida en la colección de tesis y trabajos de grado que serán migrados directamente al Repositorio institucional.

Por lo anterior, se requiere estandarizar los datos y tomar los documentos para extraer las hojas que contengan información que no puede ser pública, entes de la respectiva migración. Este procesamiento empezó a realizarse de forma manual, pero por la cantidad de información, se tiene

la necesidad de desarrollar una herramienta que permita reconocer textos e imágenes con datos de interés de los diferentes archivos, con el fin de tomar los resultados obtenidos para extraer las páginas y crear documentos que separen dicha información que debe ser cargada con protección y privacidad, dejando visible en acceso abierto solo el texto completo del trabajo de grado y tesis. Una vez procesados estos documentos con las condiciones de privacidad adecuadas, se deben migrar masivamente a Noesis.

Además, para la Universidad es muy importante que su producción científica y académica pueda ser visualizada desde redes de colaboración tales como la Red Colombiana de Información Científica (REDCOL), La Referencia a nivel de América Latina o Google Académico, por lo cual se requiere una integración o conexión de Noesis con algunas de estas redes para que automáticamente se puedan visualizar su contenido en diferentes sistemas de búsqueda. De igual manera, existe un identificador de perfiles de investigador a nivel mundial que permite unificar su información y diferenciarlos ante otros investigadores con datos similares, como es el caso del ORCID (Open Research and Contributor ID) que corresponde a un “identificador único que proporciona a los investigadores un código de autor persistente e inequívoco que distingue claramente su producción científica y evita las confusiones con nombres personales coincidentes o similares”. Es importante también gestionar la integración automática de estos perfiles con Noesis para que cada investigador pueda tener todo lo que ha publicado directamente en su ORCID.

2. Objetivos

2.1 Objetivo General

Realizar mejoras en el Repositorio Institucional que permitan la migración de trabajos de grado e integraciones para mayor visibilidad de la producción científica y académica de la Universidad Industrial de Santander.

2.2 Objetivos Específicos

Analizar la estructura actual de los trabajos de grado de estudiantes UIS graduados del 2004 al 2021 disponibles en el Sistema de Gestión de Bibliotecas LIBRUIS, con el fin de diseñar la estrategia para la migración de la información que contemple la privacidad de los datos de los autores.

Desarrollar e implementar una herramienta software que permita la extracción y procesamiento de los archivos y metadatos de acuerdo con la configuración del sistema.

Realizar la carga de los trabajos de grado optimizados al Repositorio Institucional UIS.

Implementar los requerimientos necesarios para la integración y compatibilidad del Repositorio Institucional UIS con el perfil de investigador ORCID y redes de colaboración para la visibilidad de producción científica y académica.

3. Marco referencial y metodológico

3.1 Marco teórico

3.1.1 Dspace

DSpace es una aplicación web que le permite a los investigadores y académicos publicar documentos y datos. El software de depósito de DSpace satisface una necesidad como sistema de archivos digitales, centrado en el almacenamiento a largo plazo, el acceso y la preservación del contenido digital (DSpace). Es comúnmente usado como repositorio institucional de organizaciones académicas con repositorios digitales abiertos.

DSpace permite un fácil acceso a todo tipo de contenido digital, incluido texto, imágenes, imágenes en movimiento, archivos MPEG y conjuntos de datos.

La primera versión pública de DSpace se lanzó en noviembre de 2002 como un esfuerzo conjunto entre desarrolladores del MIT y HP LABS, por lo que lleva hoy más de 20 años en funcionamiento.

A nivel mundial, este software de código abierto es utilizado por más de mil instituciones para satisfacer sus necesidades de almacenamiento digital tales como: Cambridge University, University of Edinburgh, The national Library of Finland, CONCYTEC, Texas Digital Library, University of Minnesota, entre otras. (Dspace, s. f.)

3.1.2 ORCID

ORCID, que significa Open Researcher and Contributor ID, es una organización mundial sin fines de lucro que se sustenta con las tarifas de sus organizaciones miembros. ORCID proporciona un identificador único a los investigadores, un código de autor persistente e inequívoco que distingue claramente su producción científica y evita las confusiones con nombres

personales coincidentes o similares". De esta manera, cada investigador tiene su código id el cual podrá compartir y publicar para que cualquier persona pueda acceder a su perfil ORCID y consultar la producción científica que tiene esta persona. (ORCID, s. f.).

3.1.3 Google Académico

Google Scholar proporciona una forma sencilla de buscar ampliamente literatura académica. Desde un solo lugar, puede buscar en muchas disciplinas y fuentes: artículos, tesis, libros, resúmenes y opiniones judiciales, de editoriales académicas, sociedades profesionales, repositorios en línea, universidades y otros sitios web. Google Scholar lo ayuda a encontrar trabajos relevantes en todo el mundo de la investigación académica. (Google, s. f.).

3.1.4 REDCOL

La Red Colombiana de Información Científica, fue conformada por Colciencias hoy Minciencias a través de la Resolución 0166 del 20 de febrero de 2019 como la iniciativa que busca proveer al país de un modelo de gestión que articule los esfuerzos que los actores del Sistema Nacional de Ciencia, Tecnología e innovación en el ámbito de la información, como insumo para fortalecer el desarrollo científico tecnológico, la apropiación social del conocimiento y la articulación con redes internacionales para la gestión de la información científica.

Con el propósito de desarrollar estrategias que promuevan la Ciencia Abierta, el Ministerio de Ciencia tecnología e Innovación - Minciencias como organismo rector de la investigación en el país, asumió la responsabilidad de gestionar una iniciativa que recopilara esfuerzos, creara sinergias y consolidara lazos de cooperación internacional.

La Red Colombiana de Información Científica tiene como objetivo articular los esfuerzos de los actores del Sistema Nacional de Ciencia, Tecnología e Innovación - SNCTI para potenciar el acceso, la visibilidad, circulación y gestión de la información científica nacional a partir de la

formulación de políticas y coordinación de la implementación de componentes de Ciencia Abierta. (REDCOL, s. f.).

3.1.5 Producción científica

La producción científica se le atribuye que es la creación y publicación de artículos, revistas, libros, tesis, entre otros, pero más concretamente es la representación y materialización del conocimiento ya sea académico o científico.

Tras la conceptualización y estudio a través de los tiempos, muchos toman la producción científica como la cantidad de aplicaciones en la práctica y se rigen de dos maneras, la primera beneficia directamente a una entidad o institución donde se tiene en cuenta el número de artículos (investigaciones e innovaciones de las diferentes áreas disciplinares) publicados y la segunda que corresponde al caso de los autores donde se parte del número de publicaciones que están a su nombre (Dorta, 2016).

No obstante, no todos precisan de la misma manera el concepto de producción científica y académica, puesto que esta percibe más que la cantidad. Siendo así, la producción científica comprende todas aquellas creaciones que aportan al conocimiento y amplían las visiones de la realidad en los humanos, añadiendo factores que precisan estos como la calidad, la severidad y el objetivo que agregan valor a la ciencia, independientemente de su publicación y aplicación (Chaui, 1997, p.356).

Consecuentemente, la producción científica al ser uno de los derivados de la información científica, es uno de los principales factores que aporta al desarrollo de nuevos conocimientos ligados a la ciencia por medio de sus resultados. Además, contribuye al desarrollo profesional de los investigadores, puesto que entre más conocimiento científico de un área en específico tenga un

profesional, mayor puede ser el aporte que estos pueden brindar, produciendo nuevas visiones y así ampliando el conocimiento de una investigación.

En principio, la mayoría de las investigaciones son producidas es las universidades e instituciones de investigación profesional, pero en mayor parte, se tiene que la producción científica parte de los conocimientos que son desarrollados por medio de material académico y la enseñanza que aportan a la ciencia e innovación en los conocimientos, que más específicamente son representados y corresponden a trabajos de grado o tesis (pregrado, especialización, maestría y doctorado) y artículos desarrollados por grupos de investigación.

3.1.6 Repositorio Institucional

Se puede definir repositorio institucional como una colección digital o una base de datos que reúne, almacena, preserva y divulga, de manera abierta o libre acceso a las comunidades universitarias, toda aquella producción académica y de investigación de una institución.

Siendo un repositorio institucional, este se centra en la recolección de las investigaciones originales y de propiedad intelectual de una institución la cual es activa en muchos campos de la ciencia, no obstante, los repositorios también son un indicador de calidad académica institucional (Crow, 2002), lo que quiere decir, que además de almacenar, preservar y difundir la producción científica y académica, es uno de los medios que ayudan a medir la calidad de dichas producciones y la utilidad de estas.

3.1.7 Migración de datos

La migración de datos corresponde a la transferencia de datos o material digital desde un sistema gestor o de almacenamiento de origen hacia otro o en busca de una reubicación de datos almacenados.

Cuando se habla de almacenamiento de datos, las bases de datos son uno de los principales componentes que hace parte de este proceso, pero en algún momento algunas tecnologías pueden considerarse como obsoletas debido a la innovación y presentación de nuevas tecnologías con características detalladas que cumplen con las nuevas necesidades, ya sea de organización, preservación, difusión, optimización en procesos, de capacidad de almacenamiento, entre otros.

Partiendo de lo anterior, una de las mejores soluciones para resolver nuevas necesidades e innovar es la migración de datos a nuevos sistemas, teniendo en cuenta los principales pasos que se presentan para llevar la migración de la mejor manera que son: analizar y determinar los formatos de la información y las posibles conversiones que se deben hacer con el fin de adecuarse a las nuevas necesidades, conocer la estructura del nuevo sistema y adecuarlo, realizar pruebas de funcionamiento previas y por ultimo migrar en su totalidad. (Martínez, 2014).

3.2 Estado del arte

3.2.1 Consideraciones para la creación de un repositorio institucional en la Universidad Industrial de Santander

En el año 2011 se presentó un trabajo de grado llamado “Consideraciones para la creación de un repositorio institucional en la Universidad Industrial de Santander” para optar por el título de especialización en telecomunicaciones. Fue desarrollado por Liliana Clemencia Díaz González y allí se presentó un análisis de los aspectos claves a evaluar en caso de implementar un repositorio institucional. Se definieron las directrices de diseño y organización de comunidades, colecciones y contenido; de igual manera se plantearon aquellas políticas relacionadas con autores, beneficios, flujo de trabajo, derechos de autor y metadatos. También se presentó un análisis técnico sobre el software disponible en el mercado y de acceso abierto y finalmente se realizó una comparación entre las diferentes opciones con el propósito de obtener la más indicada para el montaje del

repositorio institucional de la Universidad. Algunas conclusiones de este trabajo de grado fue que es demasiado importante para la Universidad tener un lugar donde se deposite toda la producción científica e intelectual ya que ayuda a que se incremente la visibilidad de esta información y aumente la relevancia en los rankings mundiales.

3.2.2 Diseño, desarrollo e implementación del repositorio institucional en la biblioteca de la Universidad Industrial de Santander.

Para el 2012 y parte del 2013 se realizó un trabajo de grado con modalidad de práctica empresarial titulado: “Diseño, desarrollo e implementación del repositorio institucional en la biblioteca de la Universidad Industrial de Santander.”. Este trabajo de grado fue desarrollado por Ferney Mauricio Calderón para optar por el título de Ingeniero de Sistemas e Informática. Su objetivo fue dar a conocer el desarrollo del Repositorio Institucional de la Universidad que mejoraría el proceso de recolección, almacenamiento, preservación y difusión del contenido académico producido en la institución y maximizar la visibilidad de dicho material de manera abierta y de libre acceso para la comunidad universitaria. Para el desarrollo de este objetivo se opta por el uso del software Dspace, siendo este de libre acceso y con características que cumplen con las necesidades y es uno de los más usados a nivel global por muchas instituciones.

3.2.3 Componente software para la migración de la base de datos de registro académico a colecciones MongoDB

Se realizó un trabajo de grado en el año 2014 presentado por Sergio Andrés Ortiz Machuca para optar por el título de Ingeniero de Sistemas e Informática. En este trabajo de grado se desarrolló un sistema para realizar la migración del sistema de registro académico a la base de datos en MongoDB. Este proyecto está basado en JEE6(Java Enterprise Edition 6). Con la

implementación de este componente, se pretendió mejorar y agilizar todos los procesos asociados al manejo de la información del posgrado de maestría en ingeniería de sistemas.

3.2.4 DSpace en las universidades españolas

De acuerdo con un estudio de febrero de 2015, la lista de repositorios institucionales de REBIUN, Red de Bibliotecas Universitarias, incluye las bibliotecas de las universidades españolas (más el Consejo Superior de Investigaciones Científicas), un total de 74 instituciones. Como resultado del estudio, en el cual se extrajeron datos referentes al software de implementación y otros datos de caracterización de cada repositorio, se obtuvo que 58 instituciones tienen repositorio institucional, 49 tienen instalado el software de DSpace y es por mucho el software más usado en la construcción de repositorios. (Braña, 2015).

3.2.5 La Universidad Autónoma del Caribe implementó DSpace.

La Universidad Autónoma del Caribe implementó en el 2015 el Repositorio Institucional DSpace teniendo en cuenta la gran ayuda que representa al preservar el material digital. Estas herramientas son de gran importancia ya que mantienen el legado de una organización; estas herramientas facilitan la preservación digital, la comunicación y distribución a las comunidades educativas. (UAC, 2015).

Algunas de las comunidades que fueron implementadas en DSpace de esta universidad son:

- Alfabetización Mediática informacional.
- Investigaciones.
- Normatividad, estatutos y actos institucionales.
- Revistas científicas UAC.
- Sistema de gestión de calidad.

3.2.6 El repositorio de la universidad de Palermo

El repositorio digital de la Universidad de Palermo es un servicio que recopila, preserva y distribuye material digital. (Uni. Palermo, 2002).

Algunas de las comunidades que implementaron son:

- Facultad de Arquitectura.
- Facultad de Derecho.
- Facultad de Ingeniería.
- Facultad de Negocios.

3.2.7 Integración ORCID con el repositorio del instituto español de Oceanografía

Dada la importancia que tiene para el IEO la visibilidad de los investigadores y su producción científica, el repositorio E-IEO, basado en el software DSpace, incorporó progresivamente mejoras funcionales en los aspectos de centralidad de los autores, facilidades de depósito de trabajos e incremento de la visibilidad de la producción científica. En el año 2013 se comenzó por normalizar los nombres de autor en sus formas autorizadas, un año después, incorporaron a la base de datos de autores, los identificadores ORCID de aquellos investigadores que en ese momento habían solicitado identificadores personales. Durante el año 2015, implementaron perfiles de autor en el repositorio, mostrando la información en forma de páginas personales de los investigadores pertenecientes al IEO y en el año 2016 decidieron optar por la membresía institucional en ORCID, y de igual manera comenzar el proyecto de integración entre los perfiles de autor presentes en E-IEO y los perfiles ORCID. (Mosquera et al., s.f.).

Se pueden destacar algunos de los proyectos piloto desarrollados en el Reino Unido en algunas instituciones de educación superior tales como Henderson, Johnson & Woodward en el 2015. Algunos de los cambios también tienen que ver con las modificaciones al estándar de

recolección RIOXX para incluir los identificadores ORCID en los metadatos expuestos por los repositorios.

3.2.8 Prensa de la universidad de Cambridge y ORCID

Muchas de las revistas de la universidad de Cambridge requieren que el autor correspondiente de un artículo envíe una identificación ORCID antes de que puedan enviar su trabajo. Para todas las demás revistas, la provisión de una ID ORCID es opcional pero muy recomendable. Ellos creen que una mayor aceptación de ORCID proporcionará beneficios en todo el ecosistema de investigación, tanto para autores, investigadores, instituciones, financiadores y editores. Usados correctamente, los id de ORCID permiten una mayor visibilidad para las publicaciones de investigación, eficiencias para los sistemas de editores y financiadores y comodidad para los investigadores al administrar múltiples cuentas y actividades. (Cambridge University Press, s. f.),

3.3 Marco metodológico

Partiendo de las necesidades de la Biblioteca UIS, relacionadas con el mejoramiento del Repositorio Institucional Noesis, se define la siguiente metodología.

3.3.1 Etapas metodológicas

3.3.1.1 Análisis de los trabajos de grado para la migración. Inicialmente se realiza un análisis detallado de los trabajos de grado, tesis y disertaciones de los estudiantes UIS graduados entre 2004 a 2021 que se encuentran actualmente en el servidor de LIBRUIS, con una cantidad aproximada de 26.000 documentos, tanto de la información referencial disponible en el Catálogo Bibliográfico contrastada con los metadatos necesarios para Noesis, como de la estructura de los archivos teniendo en cuenta su año de publicación, el formato del texto y el estilo de normas que fue usado para su organización.

Seguido a esto se procede a estructurar los metadatos, clasificar los documentos, revisar la seguridad de estos, verificar si la información se encuentra como imagen o solo texto e investigar sobre los mecanismos existentes para la extracción o eliminación de una o varias páginas específicas en un archivo PDF; con el fin de seleccionar el lenguaje de programación adecuado para el desarrollo y la mejor estrategia de migración.

3.3.1.2 Diseño de la estrategia de migración. De acuerdo con el análisis realizado, se comienza por verificar la organización de los metadatos requeridos para la migración, seguido de la clasificación mediante carpetas de los trabajos de grado. Es importante aclarar que se elige una migración por lotes, tomando como característica en común el año de publicación, ya que este garantizaba rasgos similares entre los trabajos de grado, tesis y disertaciones.

Posteriormente se define la estrategia a ejecutar y se diseñan las herramientas software necesarias, con el código para el funcionamiento del algoritmo que permite la extracción de metadatos del sistema LIBRUIS y de páginas PDF en una gran cantidad de archivos al mismo tiempo, junto a la respectiva decodificación para lograr su manipulación. En este momento se completó el primer objetivo específico del proyecto.

3.3.1.3 Desarrollo e implementación de la herramienta. En esta etapa se comienza a desarrollar el algoritmo para reconocer texto dentro de un archivo PDF que permite detectar las páginas a extraer y con ellas crear nuevos archivos PDF. Para el caso en donde las páginas requeridas estaban en formato texto, el algoritmo debe localizar las palabras clave definidas previamente e indicar en qué página está para guardar ese dato en una variable que será utilizada en la siguiente fase.

Ya teniendo identificadas las páginas necesarias para la extracción, se desarrolló la herramienta software que mediante algoritmos crean una copia del archivo original, para

posteriormente extraer de la copia las páginas definidas y crear tres archivos nuevos, uno con la página de la nota, otro con la página de la carta de autorización de publicación y otro con el PDF original sin las páginas que fueron extraídas.

Después de comprobar el correcto funcionamiento del algoritmo para algunos años, se verifica para cada uno de los trabajos de grado requeridos para migración. Se implementa un código para la creación de carpetas nombradas con el número de inventario del trabajo de grado que posteriormente, mediante otro algoritmo, se incluyeron los archivos separados y el texto completo del documento, para cada uno.

Por otra parte, ya teniendo localizadas las palabras claves para la extracción de algunos metadatos en el archivo y previamente identificadas las páginas donde estas se encuentran, se desarrolló una herramienta software que mediante algoritmos lee el texto contenido en las páginas definidas y lo almacena en una variable general. Por medio de las palabras localizadas, se obtiene el contenido de interés, se separa en cinco variables (resumen, palabras clave, title, abstract y key words) y se deposita en un archivo de Excel. Además, se desarrollaron pruebas para validar el método propuesto y los códigos desarrollados, tanto para la separación de archivos como para la extracción de metadatos.

Con la generación de los archivos requeridos para la migración, se completó el segundo objetivo específico correspondiente a desarrollar e implementar una herramienta software que permita la extracción y procesamiento de los archivos y metadatos de acuerdo con la configuración del sistema.

3.3.1.4 Migración de trabajos de grado optimizados. Al tener todos los archivos comprimidos por año y haber extraído todas las páginas de cada trabajo de grado, se comenzaron a cargar carpetas por año junto a los metadatos necesarios directamente a Noesis. Este proceso se

realiza con el apoyo de los administradores del sistema DSpace, por lo cual la información se envía de acuerdo con las características definidas previamente las cuales fueron por año de publicación tal y como se dijo en la fase anterior.

En esta etapa también se realiza una revisión de la información y archivos migrados, con el fin de proyectar las actividades necesarias para la depuración o mejora de los resultados; ya que esto no dependerá únicamente de lo realizado en este proyecto, sino que se debe tener en cuenta las particularidades que puedan existir en el sistema actual LIBRUIS, del cual se extrae la información. De igual manera se realiza una revisión de los archivos Excel con los metadatos y de los documentos finalmente migrados verificando que su estructura estuviera correctamente y que la migración hubiera resultado exitosa. En este punto finalmente se cumplió el tercer objetivo específico el cual corresponde a realizar la carga de los trabajos de grado optimizados a Noesis.

3.3.1.5 Requisitos para la integración y compatibilidad. Finalmente se complementa el proyecto con la revisión de los requisitos necesarios para la integración de Noesis con algunas redes de colaboración tales como la Red Colombiana de Información Científica (REDCOL) y Google Académico. Cada uno de ellos exige un formato en el que se debía tener la información para poder realizar la respectiva integración. Esta información se consulta directamente en las páginas web disponibles. El propósito de esta fase fue dejar claridad sobre los cambios necesarios en la configuración de Noesis para cumplir con las directrices y lograr la integración.

Además, se analizaron los requisitos necesarios para realizar la integración con ORCID. Ya teniendo las directrices y requisitos, se definen las configuraciones necesarias y se implementan algunas de ellas para proceder con dicha integración garantizando la correspondencia de las publicaciones y demás datos importantes directamente en los perfiles de los investigadores UIS.

Para este momento se completaron todos los objetivos de este proyecto incluido el objetivo general.

4. Desarrollo de la práctica

Es importante mencionar que para el desarrollo de la práctica se inicia con la capacitación por parte de algunos funcionarios de Biblioteca, especialmente enfocada en el conocimiento y uso de los sistemas de información, como el Repositorio Institucional Noesis, el sistema de gestión de Biblioteca LIBRUIS y especialmente el Catálogo Bibliográfico; junto a los permisos requeridos para la localización y el acceso a los archivos PDF originales de la colección de los trabajos de grado, tesis y disertaciones, con los que se trabaja en este proyecto. Además, se realizaron reuniones para aclarar dudas técnicas con el profesional interno de Biblioteca y el personal externo implicado en el desarrollo, así como dudas académicas con el director y la codirectora del proyecto. De igual forma se participa en el Comité Primario de la Biblioteca UIS para presentar los avances en el proyecto y recibir algunas recomendaciones, de las cuales se definen los siguientes requerimientos para el desarrollo de la práctica:

- Extracción de páginas de documentos de trabajo de grado. En cuanto a la problemática con las páginas de nota y carta dentro de los trabajos de grado, era necesario extraerlas del documento principal de los trabajos comprendidos entre los años 2004 a 2021, creando archivos aparte para cada uno y eliminando esas páginas de los documentos originales.
- Extraer los metadatos necesarios para realizar la migración. Evaluar la mejor forma de obtener los metadatos requeridos, tanto de la base de datos de LIBRUIS como

de los documentos principales de los trabajos de grado comprendidos entre los años 2004 a 2021.

- Realizar la migración de los trabajos de grado ya procesados a Noesis.
- Integrar Noesis con redes de colaboración como Google Scholar, REDCOL y ORCID. Revisar las directrices necesarias para realizar las configuraciones en el repositorio que permitan dichas integraciones.

4.1 Análisis de los trabajos de grado para la migración

Noesis actualmente se encuentra organizado mediante comunidades y colecciones, las cuales están representadas por el tipo de producción académica y científica de la comunidad universitaria, como se observa en la Figura 3, estas corresponden a Tesis y disertaciones, Trabajos de grado, Revistas científicas UIS, producción científica y editorial, entre otras.

Figura 3

Interfaz principal del Repositorio Institucional Noesis

Nota. La figura muestra la interfaz principal del Repositorio Institucional Noesis, donde se aprecian todas sus comunidades. Tomado de *https://noesis.uis.edu.co/home*.

De acuerdo con los objetivos planteados en este proyecto la migración se enfoca en las comunidades de Trabajos de grado, Tesis y disertaciones, las cuales están compuestas por subcomunidades que corresponden a las facultades de la Universidad, internamente se desglosan por escuelas y en última instancia, colecciones que representan los programas académicos.

4.1.1 Información en LIBRUIS vs metadatos NOESIS

Se inicia con el análisis de los metadatos definidos en la colección de trabajos de grado, tesis y disertaciones de Noesis, seguido de un estudio a la estructura de la base de datos que contiene la información del Catálogo Bibliográfico en LIBRUIS, con el fin de evaluar las similitudes o diferencias al momento de organizarlos para la migración.

Se observa que al Noesis estar implementado en el software de acceso abierto DSpace, su esquema de metadatos oficial es Dublin Core (DC), mientras que la base de datos de LIBRUIS, al ser desarrollo propio de la Universidad, no cuenta con un esquema o formato oficial, tipo Marc 21 o Dublin Core; lo que propuso como reto familiarizarse con el modelo de la base de datos actual de la Biblioteca para detectar las posibles tablas y campos que contenían la información que se requerían para la respectiva organización de acuerdo al esquema necesario. Como resultado se presentan en la Tabla 1 las etiquetas de los metadatos definidos en la colección de trabajos de grado, tesis y disertaciones en Noesis y en la Figura 4 el esquema de las tablas necesarias con sus campos y relaciones basado en el diagrama entidad relación de la base de datos de LIBRUIS.

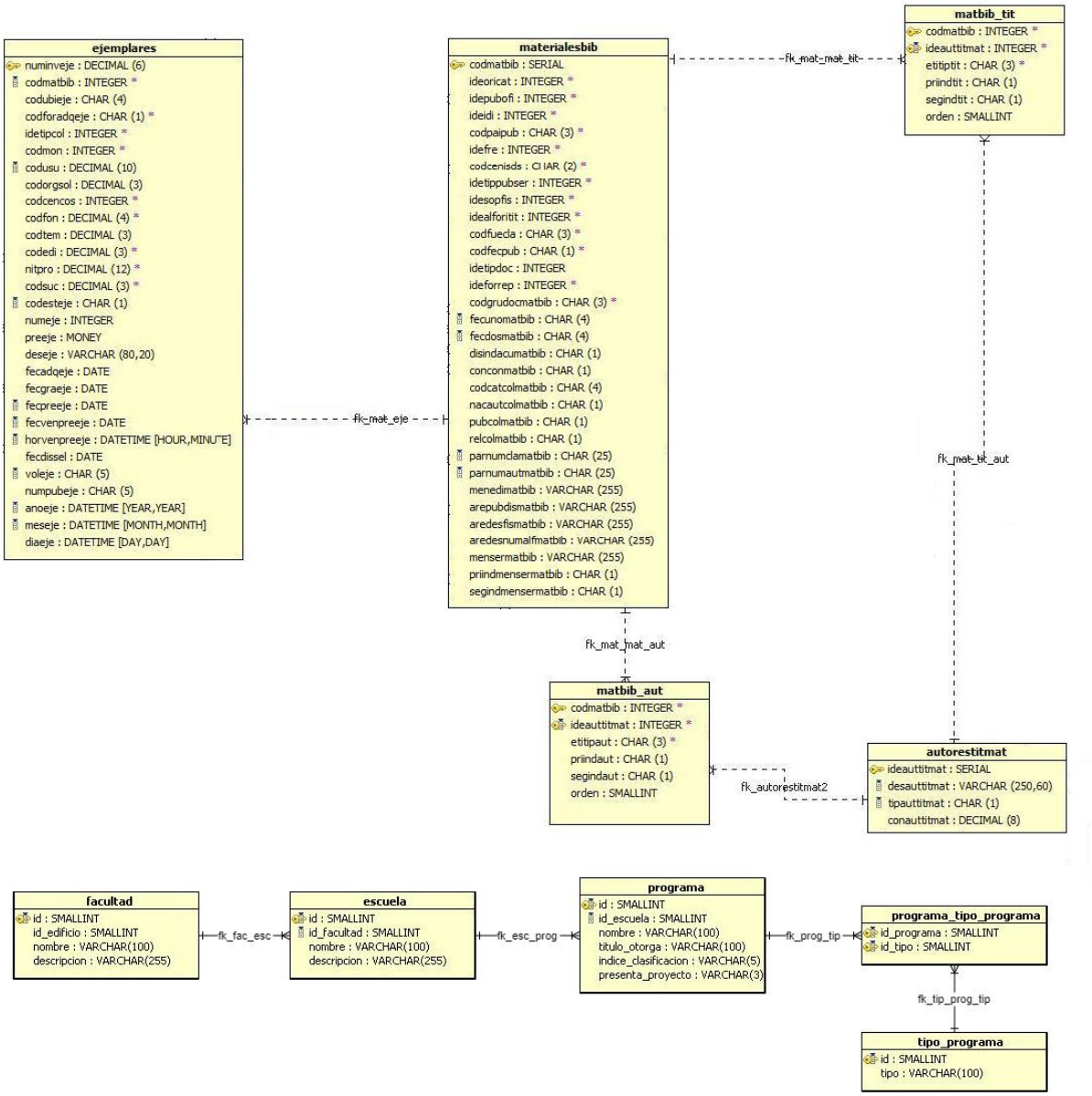
Tabla 1

Descripción de etiquetas de metadatos necesarios para la migración:

Etiqueta Noesis	**descripción**
dc.contributor.author	Persona responsable de la creación del recurso.
dc.contributor.advisor	Persona secundaria que contribuyó en la creación y seguimiento de la creación del recurso.
dc.date.created	Fecha en la que el recurso fue creado.
dc.date.issued	Fecha en la que el recurso fue puesto a disposición de usuarios en su forma actual.
dc.title	Nombre asignado al recurso por el autor o colaborador (español).
dc.title.english	Nombre asignado al recurso por el autor o colaborador (inglés).
dc.description.abstract	Escrito textual (resumen) que describe el contenido de la creación del recurso (español).
dc.description.abstractenglish	Escrito textual (resumen) que describe el contenido de la creación del recurso (inglés).
dc.subject	Palabras clave que identifican y describen brevemente el recurso (español).
dc.subject.keyword	Palabras clave que identifican y describen brevemente el recurso (inglés).
dc.language.iso	Idioma en el que se encuentra el contenido del recurso.
dc.description.degreelevel	Nivel de estudio al que corresponde el recurso.
dc.description.degreename	Título que se obtuvo por la creación y desarrollo del recurso.
dc.publisher.program	Colección a la cual pertenece el recurso.
dc.publisher.school	Comunidad a la cual pertenece el programa.
dc.publisher.faculty	Comunidad a la cual pertenece la escuela.

Figura 4

Diagrama entidad relación de las tablas necesarias y sus campos



Nota. Tablas de LIBRUIS que fueron necesarias al momento de la extracción de los metadatos.

Teniendo en cuenta los metadatos requeridos en Noesis, se realiza una comparación con la base de datos de LIBRUIS, identificando la ruta de donde se pueden recuperar los campos que eran necesarios y sus respectivas posiciones. Esta información se presenta en la siguiente tabla .

Tabla 2

Metadatos necesarios para la migración vs información en LIBRUIS:

etiqueta Noesis (DC)	Metadatos base de datos LIBRUIS
dc.contributor.author	Tabla autorestitmat, campo desauttitmat sin etiqueta ", Dir."
dc.contributor.advisor	Tabla autorestitmat, campo desauttitmat con etiqueta ", Dir."
dc.date.created	Tabla materialesbib, campo fecunomatbib
dc.date.issued	Tabla materialesbib, campo fecunomatbib
dc.title	Tabla autorestitmat, campo desauttitmat
dc.title.english	Ausente
dc.description.abstract	Ausente
dc.description.abstractenglish	Ausente
dc.subject	Ausente
dc.subject.keyword	Ausente
dc.language.iso	Ausente
dc.description.degreelevel	Tabla tipo_programa, campo tipo
dc.description.degreename	Tabla programa, campo titulo_otorga
dc.publisher.program	Tabla programa, campo nombre
dc.publisher.school	Tabla escuela, campo nombre
dc.publisher.faculty	Tabla facultad, campo nombre

Nota. Esta tabla muestra las etiquetas de los metadatos necesarios para realizar la migración de acuerdo con el Repositorio Institucional Noesis vs los metadatos que se recuperan de LIBRUIS; para mayor comprensión de los nombres de los campos y de la tabla en general se debe consultar la Figura 4.

Partiendo de la información recolectada, se analiza la estructura de los datos que están contenidos en LIBRUIS y la manera en que se mostraban en el Catálogo Bibliográfico, tomando como referencia el diagrama entidad relación presente en la Figura 4. Con esto se observa que mediante consultas SQL a las tablas y campos identificados, es posible obtener la mayor parte de

los metadatos necesarios, pero no se cuenta con otros datos referenciales requeridos, como el resumen en español, palabras clave en español, título en inglés, resumen en inglés y palabras clave en inglés, así como el idioma en que se encuentra el documento.

Respecto a los metadatos faltantes, se identifica que sería posible obtenerlos directamente de los archivos de trabajo de grado, tesis o disertación entregados, por lo cual, se realiza esta revisión en la siguiente fase de la metodología, estructura de los documentos (Estilo de norma).

4.1.2 Estructura de los documentos (Estilo de norma)

Al revisar los archivos disponibles en el servidor de LIBRUIS, se encuentra que los documentos estaban clasificados en carpetas por años, y dentro de cada una los archivos correspondientes, nombrados con el número de inventario asignado por la Biblioteca en el proceso de catalogación y clasificación.

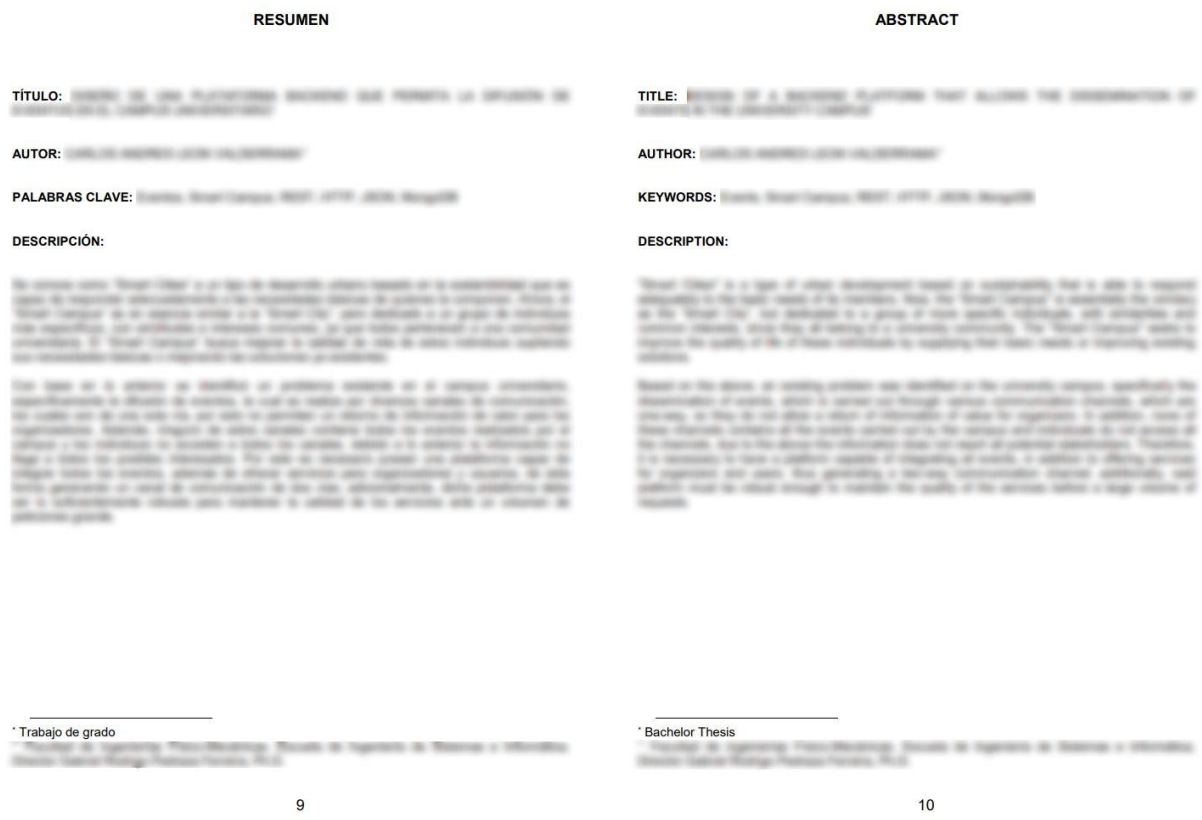
De acuerdo con los lineamientos de presentación y entrega de trabajos de grado, tesis y disertaciones a la Biblioteca UIS, el estudiante podía seleccionar la estructura y organización del documento, basado en uno de los estilos de la norma ICONTEC, APA o Vancouver. Para el caso de interés de este proyecto, realmente la diferencia radicaba en que la norma ICONTEC requiere portada y contraportada, mientras que APA y Vancouver solo cuentan con una portada. Pero en general se encuentra que, sin importar el estilo de la norma seleccionada, los documentos contienen páginas en específico, con el resumen en español y el resumen en inglés, donde se encuentran los metadatos faltantes; pero, aunque estas siguen unas directrices para su organización, se observa que la información de interés podría estar en cualquier posición, lo que conlleva a analizar todo el documento.

En la mayoría de los documentos los metadatos se organizaban con la misma estructura, donde se encontraba primero la página con metadatos en español seguida de la página con los

metadatos en inglés y para ambos casos se encontraba el título, autores, palabras clave y resumen, como se refleja en la Figura 5. Para el caso del idioma del documento se optó por estandarizar el español para todos los documentos de forma manual.

Figura 5

Estructura de páginas que corresponden a resumen y abstract.



Respecto a la posición de estas páginas con los resúmenes se encuentra que, de acuerdo con los lineamientos, hasta el año 2021, se requería incluir la nota del proyecto y la carta de autorización de publicación de los autores del trabajo, después de la portada o contraportada, es decir en las páginas 3 y 4 respectivamente. Pero de igual forma, de observa que no siempre se cumple con esta estructura, ya que algunas no contenían esta información y otros no se encontraban en el mismo orden o dependía del número de autores participantes en el proyecto. Se concluye que

por lo general estas páginas con los metadatos de interés se pueden encontrar entre las páginas 8 y 30 dependiendo del contenido.

Los archivos correspondientes a los años 2004 hasta la mitad del año 2008 no tenían carta de autorización, por lo cual fue necesario buscar estos documentos en físico en la bodega de la Biblioteca. Los archivos de mitad del año 2008 hasta 2020 contaban con la carta de autorización y la nota de proyecto dentro del archivo original. Además, se detecta que los documentos originales en formato PDF disponibles en el Catálogo Bibliográfico estaban encriptados, por lo que no se podían modificar; en una fase posterior (4.2.4 Diseño de la herramienta software) se dará solución a esta problemática. De igual manera se encuentra que algunas de las páginas tanto de nota como de cartas estaban en formato de imagen y otras en formato de texto.

Finalmente, para esta fase no se tuvo en cuenta qué tipo de norma fue utilizada para la elaboración del trabajo, ya que en la mayoría de los casos las hojas de nota y carta estaban en la misma posición y de igual manera los metadatos no se vieron afectados por esto.

4.1.3 Mecanismos para extracción de páginas en archivos y extracción de metadatos

Se realiza investigación con el fin de encontrar los mecanismos que existen para la extracción de páginas y metadatos en archivos PDF a gran escala, inicialmente encontrando que dentro del lenguaje de programación Python, existen varias librerías que permiten este tipo de extracciones, partiendo de la definición de unas páginas en específico y por medio de programación proceder a eliminar y crear nuevos archivos PDF’s. Para ello era necesario tener definidas qué páginas en específico se iban a extraer teniendo en cuenta que podía cambiar la localización de estas en cada archivo. Entonces se debía encontrar la manera de definir esas páginas para que el algoritmo supiera con cuales debía trabajar. De igual manera, se definió la metodología para encontrar las páginas de nota y carta dentro de los documentos. Se planteó utilizar

reconocimiento de texto OCR para las páginas con imágenes que se necesitaran extraer. También, existen librerías dentro de Python que reconocen texto a partir de términos clave y guardan la localización de la página en un vector para su posterior procesamiento.

Para la extracción de metadatos, se encuentra que la mayoría de las páginas que los contenían eran de tipo texto, por lo cual se opta por la búsqueda de librerías que leyeran un archivo y extrajeran el texto de las páginas de interés. Como la estructura del documento no permitía definir páginas en específico para trabajar los metadatos, se definieron términos clave, (para el caso de los metadatos en español son: resumen, autor, palabras clave y descripción; para el caso de los metadatos en ingles son: abstract, author, key words y description), que coincidían en la mayoría de los documentos y en conjunto hacían únicas las páginas de interés, con el fin de identificarlas, extraer el texto de ellas, separar el tipo de metadato por palabras clave, resumen, title, key words y abstract, y almacenar el contenido en vectores de igual manera a como se realizó para la extracción de páginas.

Para el caso de la separación de metadatos se tuvieron en cuenta algunos de los términos clave que ya se poseían, agregando uno extra ubicado en el pie de página y que variaba según el tipo de documento, para página en español (Trabajo de grado, Proyecto de grado, Monografía y Tesis de grado) y para página en inglés (Degree work, Bachelor thesis, Degree Project, Monograph, Research Project y Thesis)

4.1.4 Programación y librerías necesarias para el algoritmo

De acuerdo con la información encontrada en foros y realizando una comparación entre los lenguajes más usados para procesamiento de archivos, se decide utilizar Python ya que es un lenguaje de programación fácil e intuitivo y la mejor opción para trabajar con gran cantidad de

archivos al mismo tiempo. Se utilizaron algunas librerías nativas de Python para el tratamiento de los archivos, creación de vectores y programación en general del algoritmo.

Para la lectura de archivos, extracción de metadatos y de páginas se utilizan las siguientes librerías: PYPDF2, PyMUPDF, Pikepdf, tqdm, Pandas, Pytesseract, PIL, pdf2image y OS las cuales están especializadas en archivos PDF. La instalación y la documentación de todas las librerías se encuentra en los siguientes links:

- PYPDF2: Se instala con “pip install PyPDF2” y la documentación se encuentra en https://pypi.org/project/PyPDF2/.
- PyMUPDF: Se instala con “pip install PyMuPDF” y la documentación se encuentra en https://pypi.org/project/PyMuPDF/.
- Pikepdf Se instala con “pip install pikepdf” y la documentación se encuentra en https://pypi.org/project/pikepdf/.
- Tqdm: Se instala con “pip install tqdm” y la documentación se encuentra en https://pypi.org/project/tqdm/.
- Pandas: Se instala con “pip install pandas” y la documentación se encuentra en https://pypi.org/project/pandas/.
- Pytesseract: Se instala con “pip install pytesseract” y la documentación se encuentra en https://pypi.org/project/pytesseract/. Adicionalmente, se debe descargar el paquete de instalación más reciente de Tesseract, este se encuentra en el siguiente enlace: https://github.com/UB-Mannheim/tesseract/wiki.
- PIL: Se instala con “pip install Pillow” y la documentación se encuentra en https://pypi.org/project/Pillow/.

- pdf2image: Se instala con “pip install pdf2image” y la documentación se encuentra en https://pypi.org/project/pdf2image/. Adicionalmente, se debe descargar el paquete de instalación más reciente de Poppler, este se encuentra en http://blog.alivate.com.au/poppler-windows/.
- OS: No se requiere instalar nada, ya viene con Python, pero se necesita realizar la importación con “import OS”.

Algunas de estas librerías se usaron para la lectura del texto en cada uno de los archivos; de igual manera para las páginas que no tuvieran texto plano, se tuvo que extraer el texto de la imagen o localizar de alguna manera la posición de las páginas de interés con las que se trabajaría. Se tenía como requerimiento crear archivos con las páginas de nota de proyecto y de carta de autorización, al cual se le asignó el nombre de *#inventario-Nota trabajo de grado* y *#inventario-Carta de autorización* respectivamente, como también la extracción de metadatos correspondientes al resumen, palabras clave, key words, abstract y title.

También por seguridad se crearon copias de los archivos originales y a estos se les ubicaba y extraían las páginas de interés ya que estos eran los que iban a quedar expuestos al público y de esa manera no se modificaban los archivos originales.

Para el caso de las funcionalidades que dependen del Sistema Operativo se usó la librería OS, la cual nos permitía manipular la estructura de los directorios y más específicamente para la lectura y escritura de archivos, también se tenían librerías como pandas que nos permitían el manejo de datos en Python y en este caso para la organización de metadatos y la creación de un archivo Excel que los contenga.

En el análisis que se realiza se detecta que algunos archivos no eran claros al momento de revisar el texto que contenían, por lo que también se decide integrar la librería llamada pdf2image,

que ayuda a convertir las páginas de los PDF’s en imágenes, en conjunto con la librería llamada PIL, que ayuda a la apertura y edición de imágenes, y por último la librería llamada Pytesseract, que a menudo se usa en OCR, se utilizó para extraer el texto que contenían las imágenes creadas.

4.1.5 Carga por lotes vs Carga masiva

Al momento de avanzar en el análisis de la estructura de los trabajos de grado, se obtuvieron resultados que arrojaban una similitud entre los documentos, indiferente al año de creación y tipo de norma en los que estaban organizados, una vez teniendo claro con que partes del documento se trabajaría y verificando que esto no afectaba el flujo de trabajo, se toma la decisión de realizar una carga por lotes organizada por años, ya que era la forma en que se venía llevando el control en la Biblioteca de acuerdo con las fechas de graduación establecidas por la Universidad y que para realizar el proceso se debía contar con el tiempo del proveedor del servicio DSpace, quien interviene en la fase final de la migración. Se concluye que esta es la mejor manera para realizar una migración rápida y ordenada, permitiendo controlar el momento en que se tengan listos los archivos y se extrajeran los metadatos necesarios de un año para enviar a la migración, mientras se iba realizando el mismo proceso para los demás. La carga masiva obligaba a trabajar directamente con los archivos de todos los años a la vez, lo cual no era pertinente para ir monitoreando el avance, realizando pruebas y depuración del proceso.

4.2 Diseño de la estrategia de migración

Como resultado de la fase de análisis se decide que la extracción de las páginas requeridas del documento, como la extracción de los metadatos, se haría por medio de dos algoritmos diferentes, con el fin de optimizar dichos procesos y agilizar su implementación, ya que se estima que al hacerlo en conjunto se podría tener demoras al momento de la ejecución y sería de cierta

manera más complicada la solución de posibles errores que pudieran surgir a lo largo del desarrollo de la práctica.

4.2.1 Organización de metadatos requeridos

El Repositorio institucional utiliza metadatos que cumplen con el modelo de metadatos Dublin Core por lo que ya están definidos los campos necesarios para cada colección, entonces se deben extraer directamente de la base de datos de LIBRUIS o de los archivos del trabajo de grado, tal y como se menciona previamente en la fase de análisis.

Al momento de organizar los metadatos, se empieza realizando una consulta SQL que permitan obtener la mayoría de los metadatos contenidos en la base de datos de LIBRUIS. Para esto, tomando como referencia los resultados presentados en la Tabla 2 en conjunto con la base de datos reflejada en la Figura 4, se inicia con una consulta entre la tabla principal *materialesbib* con un filtro en el campo *idetipdoc = 2* que indicaba el tipo de documento trabajo de grado, la tabla *ejemplares* con un filtro en el campo *codubieje = 16*, el cual corresponde a la ubicación de los trabajos de grado disponibles (base de datos segundo piso) que no son confidenciales y una condición que trae la clave foránea *codmatbib* entre las dos tablas, obteniendo como primer resultado el número de inventario (*numinveje*), la fecha de creación (*fecunomatbib*) y la fecha en la que fue puesta a disposición a los usuarios, que corresponde a la misma fecha de creación.

Se agregaron relaciones con las tablas *matbib_tit* y *matbib_aut* por medio de la clave foránea *codmatbib* desde *materialesbib* obteniendo acceso a nuevos metadatos. Se tiene que la tabla *matbib_tit* en relación con la tabla *autorestitmat* por medio del *ideauttitmat* se obtiene el titulo (*desauttitmat*), al igual que con la tabla *matbib_aut* en relación con la tabla *autorestitmat* por medio del *ideauttitmat* se extrajeron los autores y directores o colaboradores de los trabajos de grado (*desauttitmat*) clasificados con tipo de autor 100 o 700 si son autor principal o autores

secundarios. Es importante mencionar que los directores o colaboradores, aparecen en la base de datos de LIBRUIS como autores secundarios, con la diferencia que se les agrega al final del nombre completo el sufijo “, dir.”. Si en la consulta no había ninguno con ese sufijo el campo quedaba vacío y se tenía que extraer directamente del documento o agregarlo manualmente al resultado de la consulta. Para este momento ya se podían obtener los metadatos correspondientes a la fecha de creación, fecha de publicación, título, autores y directores.

En el caso de los metadatos relacionados con el nivel de estudio, título que se obtiene, programa, escuela y facultad, se tuvo en cuenta la clasificación que tenía la tabla *materialesbib* en el campo *parnumclamatbib* relacionado directamente con el índice de clasificación que corresponda en la tabla *programa*. Por otra parte, se relacionaron las tablas de *escuela* con el *id* de *programa* y la tabla *facultad* que se relaciona con el *id* de *escuela*. Como resultado desde la tabla *programa* se obtuvo el nombre del programa y el título que se obtiene(titulo_otorga) así como también el nombre de la escuela y el nombre de la facultad. Por último, se relacionaron las tablas *programa_tipo_programa* y *tipo_programa* que son las que brindan el nivel de estudio al que corresponde el recurso. Debido a que se tenía planteada una migración por lotes se limitaron las consultas a obtener los datos por años de publicación y así tener una mejor organización de los datos y facilidad al momento de procesarlos.

Cabe mencionar que al momento de hacer el análisis a LIBRUIS y realizar las consultas de prueba con todas las condiciones para la extracción de metadatos, se observa la inconsistencia de datos en ciertas tablas de la base de datos respecto a los documentos que se tenían en el servidor, ya que algunos se habían dado de baja, la ubicación no correspondía al tipo de documento o los números de inventario no pertenecían al material correspondiente. De igual manera, se notó la ausencia de ciertos programas académicos en la tabla programa y se encontraron errores en el

índice de clasificación de esta. Para dar solución a esto, se realiza un proceso de corrección de errores, actualización de los datos que ya estaban y la indexación de los programas faltantes.

Para este momento se había logrado obtener la mayoría de los metadatos básicos y necesarios para la migración, con la ausencia de algunos reflejados en la Tabla 2 que serían extraídos directamente de los documentos, proceso que se detalla más adelante.

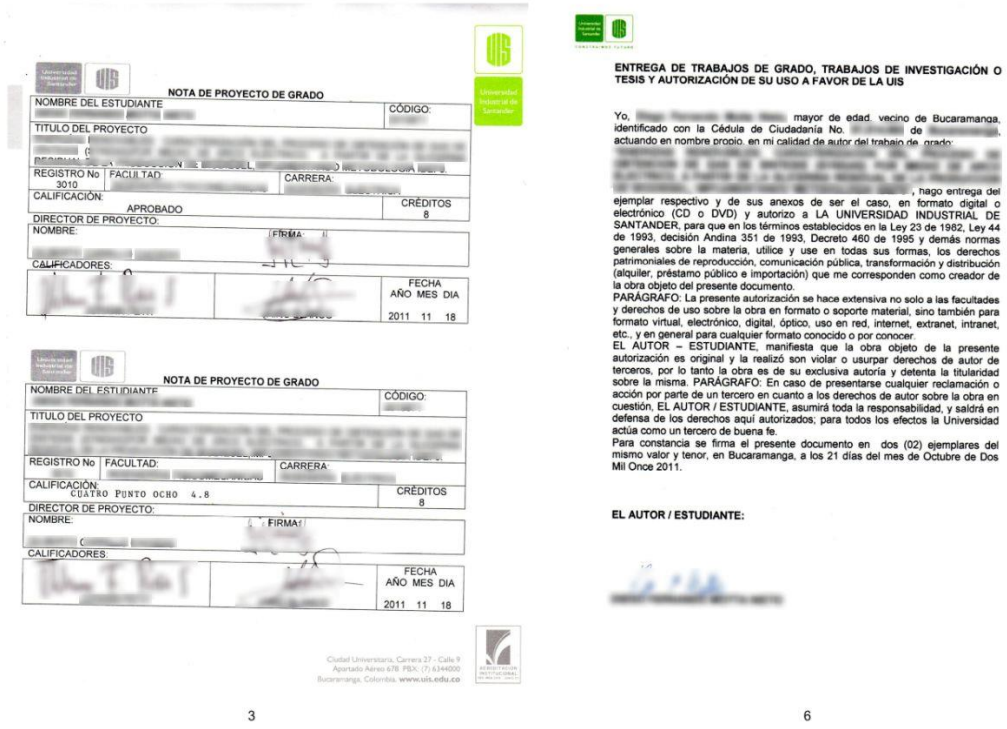
4.2.2 Clasificación de documentos a procesar

Dentro de la clasificación se realiza una verificación de los trabajos de grado, tesis o disertaciones que habían sido entregadas posterior al 2004 en formato físico, las cuales, aunque en su mayoría ya habían sido digitalizadas, no contaban con la autorización de publicación firmada por los autores, por lo cual solo se permite su consulta por el Catálogo Bibliográfico, conectado a una red propia de la Universidad; estos documentos no serían migrados a Noesis. De igual forma se hace una clasificación para el año 2021 donde se verifica cuales faltaban por migrar, ya que previamente se había realizado una prueba de migración de algunos documentos, de forma manual.

Al inicio se planteó revisar algún tipo de clasificación partiendo del tipo de norma con el que estaba elaborado el documento, pero al ver que la estructura respecto a la organización no interfería con la realización de los objetivos planteados, se decide que la clasificación se debía realizar por medio del año de publicación. Las páginas de interés que se debían extraer se pueden apreciar en la Figura 6. Estas muestran los formatos definidos para las hojas de carta de autorización y nota de proyecto.

Figura 6

Páginas de nota y carta de proyecto extraídas de un trabajo de grado.



NOTA DE PROYECTO DE GRADO

NOMBRE DEL ESTUDIANTE: [redacted] CÓDIGO: [redacted]

TITULO DEL PROYECTO: [redacted]

REGISTRO No 3010 | FACULTAD: [redacted] | CARRERA: [redacted]

CALIFICACIÓN: APROBADO | CRÉDITOS 8

DIRECTOR DE PROYECTO:

NOMBRE: [redacted] FIRMA:

CALIFICADORES: [redacted]

FECHA AÑO MES DIA: 2011 11 18

3

NOTA DE PROYECTO DE GRADO

NOMBRE DEL ESTUDIANTE: [redacted] CÓDIGO: [redacted]

TITULO DEL PROYECTO: [redacted]

REGISTRO No [redacted] | FACULTAD: [redacted] | CARRERA: [redacted]

CALIFICACIÓN: CUATRO PUNTO OCHO 4.8 | CRÉDITOS 8

DIRECTOR DE PROYECTO:

NOMBRE: [redacted] FIRMA:

CALIFICADORES: [redacted]

FECHA AÑO MES DIA: 2011 11 18

Ciudad Universitaria, Carrera 27 - Calle 9
Apartado Aéreo 678 PBX: (7) 6344000
Bucaramanga, Colombia. www.uis.edu.co

ENTREGA DE TRABAJOS DE GRADO, TRABAJOS DE INVESTIGACIÓN O TESIS Y AUTORIZACIÓN DE SU USO A FAVOR DE LA UIS

Yo, [redacted] mayor de edad, vecino de Bucaramanga, identificado con la Cédula de Ciudadanía No. [redacted] de [redacted], actuando en nombre propio, en mi calidad de autor del trabajo de grado: [redacted], hago entrega del ejemplar respectivo y de sus anexos de ser el caso, en formato digital o electrónico (CD o DVD) y autorizo a LA UNIVERSIDAD INDUSTRIAL DE SANTANDER, para que en los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia, utilice y use en todas sus formas, los derechos patrimoniales de reproducción, comunicación pública, transformación y distribución (alquiler, préstamo público e importación) que me corresponden como creador de la obra objeto del presente documento.

PARÁGRAFO: La presente autorización se hace extensiva no solo a las facultades y derechos de uso sobre la obra en formato o soporte material, sino también para formato virtual, electrónico, digital, óptico, uso en red, internet, extranet, intranet, etc., y en general para cualquier formato conocido o por conocer.

EL AUTOR – ESTUDIANTE, manifiesta que la obra objeto de la presente autorización es original y la realizó son violar o usurpar derechos de autor de terceros, por lo tanto la obra es de su exclusiva autoría y detenta la titularidad sobre la misma. PARÁGRAFO: En caso de presentarse cualquier reclamación o acción por parte de un tercero en cuanto a los derechos de autor sobre la obra en cuestión, EL AUTOR / ESTUDIANTE, asumirá toda la responsabilidad, y saldrá en defensa de los derechos aquí autorizados; para todos los efectos la Universidad actúa como un tercero de buena fe.

Para constancia se firma el presente documento en dos (02) ejemplares del mismo valor y tenor, en Bucaramanga, a los 21 días del mes de Octubre de Dos Mil Once 2011.

EL AUTOR / ESTUDIANTE:

[redacted]

6

4.2.3 Estrategia para la extracción de páginas y metadatos

En la planeación de estrategias para la extracción de páginas, se plantea en un primer momento la opción de reconocer una o varias palabras clave dentro de cada uno de los documentos y de esta manera saber en qué páginas se encontraban las dos hojas que se debían extraer. Se definieron las palabras clave que iban a ser usadas, partiendo de la comparación en varios documentos de diferentes años; para el caso de la nota de proyecto, se planteó utilizar las siguientes palabras clave “Nota”, “Sistema de trabajos de grado”, “Administración de trabajos de grado” y para el caso de la carta de autorización se planteó utilizar “Autorización de su uso a favor de la

UIS", "Entrega de trabajos de grado" como palabras clave para que el algoritmo las detectara y de esta manera guardara en un vector el número de página que posteriormente sería extraída. Lo anterior para el caso en que las páginas se encontraban en formato texto.

También se tiene en cuenta que en algunos casos los archivos PDF iban a traer páginas tanto de nota como de carta en formato de imagen, por lo que los algoritmos para detectar texto no iban a funcionar, entonces se busca otra alternativa para detectar en que páginas estaban estas imágenes. Se encuentra que casi en el 100% de los documentos estaba primero la página de nota y después la página de carta de autorización, por lo cual se define un rango en el cual tomar las imágenes que existieran en el documento y guardar esas páginas en el vector teniendo en cuenta la cantidad de notas y cartas que podrían existir en cada documento. Tomando como referencia el análisis realizado anteriormente, se tiene en cuenta que en algunos casos muy particulares se tienen hasta seis o siete páginas de nota e igualmente de carta, de acuerdo con la cantidad de autores del trabajo, entonces se debe detallar mejor la localización de estas imágenes. Por lo tanto, se pone un rango para que el algoritmo busque las imágenes dentro del archivo y de esta manera guarde la información en ambos vectores y así poder crear los archivos separados, después se revisa manualmente cada uno de estos para verificar que estuvieran correctamente y en caso de que no, arreglarlos.

En cuanto a la extracción de metadatos, partiendo de resultados del análisis de los trabajos de grado y los mecanismos para extracción de páginas en archivos y extracción de metadatos, se plantea la estrategia de realizarlo mediante el uso de términos clave que ayudarían a precisar la posición de las páginas con las que se trabajaría. Luego de hacer una revisión más a fondo, se decide buscar solo la página que contiene los metadatos en español por medio de los términos clave "resumen", "autor", "palabras clave" y "descripción", ya que, en la mayoría de los

documentos la página que contenía los metadatos en inglés corresponde a la siguiente, es decir, al momento de encontrar la página de metadatos en español por consiguiente la página posterior es la de metadatos en inglés. Estas fueron encontradas mediante la lectura de los archivos y la extracción del texto plano página por página en un vector, buscando en cada una de ellas los términos clave propuestos.

Se encuentra que cerca del 99% de los documentos poseían las páginas que contienen los metadatos en texto plano, lo que generó cierta facilidad al momento de trabajarlas, donde el porcentaje correspondiente al 1% contenían los metadatos en formato de imagen y se decidió realizar la extracción usando OCR para identificar las páginas y extraer el texto.

4.2.4 Diseño de la herramienta software

Para el caso del diseño de las herramientas software se planteó que el primer paso sería desencriptar todos los documentos originales ya que contaban con protección y era necesario para permitir el acceso y la modificación. Luego se define como implementar una lectura de todos los archivos de cada año al mismo tiempo, se precisa utilizar ciclos For y vectores para la recolección de los nombres de todos los archivos dentro de una misma carpeta. Para el desarrollo de estos algoritmos se utilizaron algunas librerías que ya fueron mencionadas previamente.

En esta fase, para el primer algoritmo se buscaban las palabras clave dentro del documento y se llenaban dos vectores con estas páginas, uno para el caso de la página de nota y otro para la carta de autorización. Luego estos vectores eran utilizados por el siguiente algoritmo el cual tomaba el documento original y a partir de cada vector creaba tres archivos, uno con la(s) página(s) de nota de proyecto, otro con la(s) página(s) de carta de autorización y por último un archivo sin esas páginas. Después de tener creados los tres archivos, se guardaban en una ruta especifica con el siguiente nombre: Para el caso de la nota de proyecto era #Inventario-NOTA DE PROYECTO

DE GRADO, para el caso de la carta de autorización era #Inventario-CARTA DE AUTORIZACIÓN y por último el archivo sin estas páginas fue nombrado como #Inventario-SP. Después se procesaron estas carpetas teniendo en cuenta que para la migración debían tener un nombre en específico por lo que se renombraron los documentos a "Documento.pdf" para el archivo sin páginas de nota y carta, "Nota de proyecto" para el archivo con la(s) página(s) de nota y "Carta de autorización" para el archivo con la(s) página(s) de carta de autorización. Después se creó una carpeta nombrada con el número de inventario y llenarla con los tres archivos que pertenecían a ella.

En el caso del segundo algoritmo, se debía tomar el vector que contenía la lista de los nombres de todos los archivos y se abriría uno por uno para extraer el texto de cada página, tomando como referencia los términos clave definidos para encontrar las páginas especificas donde se encontrarían los metadatos. Si las páginas que se buscaban existían, se debía agregar el texto extraído de dicha página y se convertiría a una lista para tratamiento y separación de los metadatos requeridos en diferentes variables de texto. Dichos metadatos debían corresponder al resumen y palabras clave tanto en español como en inglés, así como también el título en inglés.

Se tiene en cuenta que por lo general los archivos estaban compuestos por más de 50 páginas y como se realizaba la lectura una a una para extraer el texto y buscar los términos clave, fue necesario limitar la lectura de archivos a solo las primeras 30 páginas con el fin de optimizar el funcionamiento del algoritmo y evitar demoras innecesarias.

Al momento de encontrar las páginas de interés, se procesa el texto plano y se pasa a la separación de metadatos. Para esto, se agrega el contenido de la página en texto a una lista separada palabra por palabra una en cada posición en el orden de lectura del texto. En el caso de la página en español, se tuvieron en cuenta los términos clave "palabras clave" y "descripción", que se

encontraron entre el contenido de la página, “Trabajo de grado, Proyecto de grado, Monografía, Tesis de grado”, que se encuentran en el pie de página y que son los que delimitan el contenido que se debía obtener, es decir, los metadatos correspondientes a las palabras clave se delimitaron por el termino “palabras clave”, que es el que indicaba la posición inicial en la que se encontraba dentro del vector, finalizando en el término clave “Descripción” que es el que indicaba la posición final de las palabras clave así como también la posición inicial del contenido del resumen.

Una vez separados se anexaron en una hoja de cálculo con su respectivo número de inventario para identificar a qué documento correspondía.

4.3 Desarrollo e implementación de la herramienta

4.3.1 Desarrollo de las herramientas software

Ya habiendo definido anteriormente el ambiente de programación Python y contando con el diseño de la estrategia a ejecutar, se decide usar Anaconda ya que es una distribución libre y abierta que contiene dicho ambiente, por medio de la interfaz Jupyter Notebook.

4.3.1.1 Decodificación de documentos. Para desencriptar los documentos con el fin de poder accederlos y modificarlos, se implementa en algoritmo usando la librería pikepdf. Por la cual se procesa cada uno de los archivos presentes en la ruta principal y los guarda de nuevo en la misma ubicación. De esta manera se tienen los archivos originales desencriptados.

Para realizar esta decodificación se utiliza el código fuente disponible en el Apéndice A, Algoritmo creado para desencriptar documentos PDF.

4.3.1.2 Desarrollo herramienta de extracción de páginas. Para la implementación del algoritmo principal de extracción de páginas, lo primero fue instalar e importar las librerías necesarias, lo cual se realiza de la siguiente manera:

pip install PyPDF2==1.26 PyMUPDF pikepdf tqdm

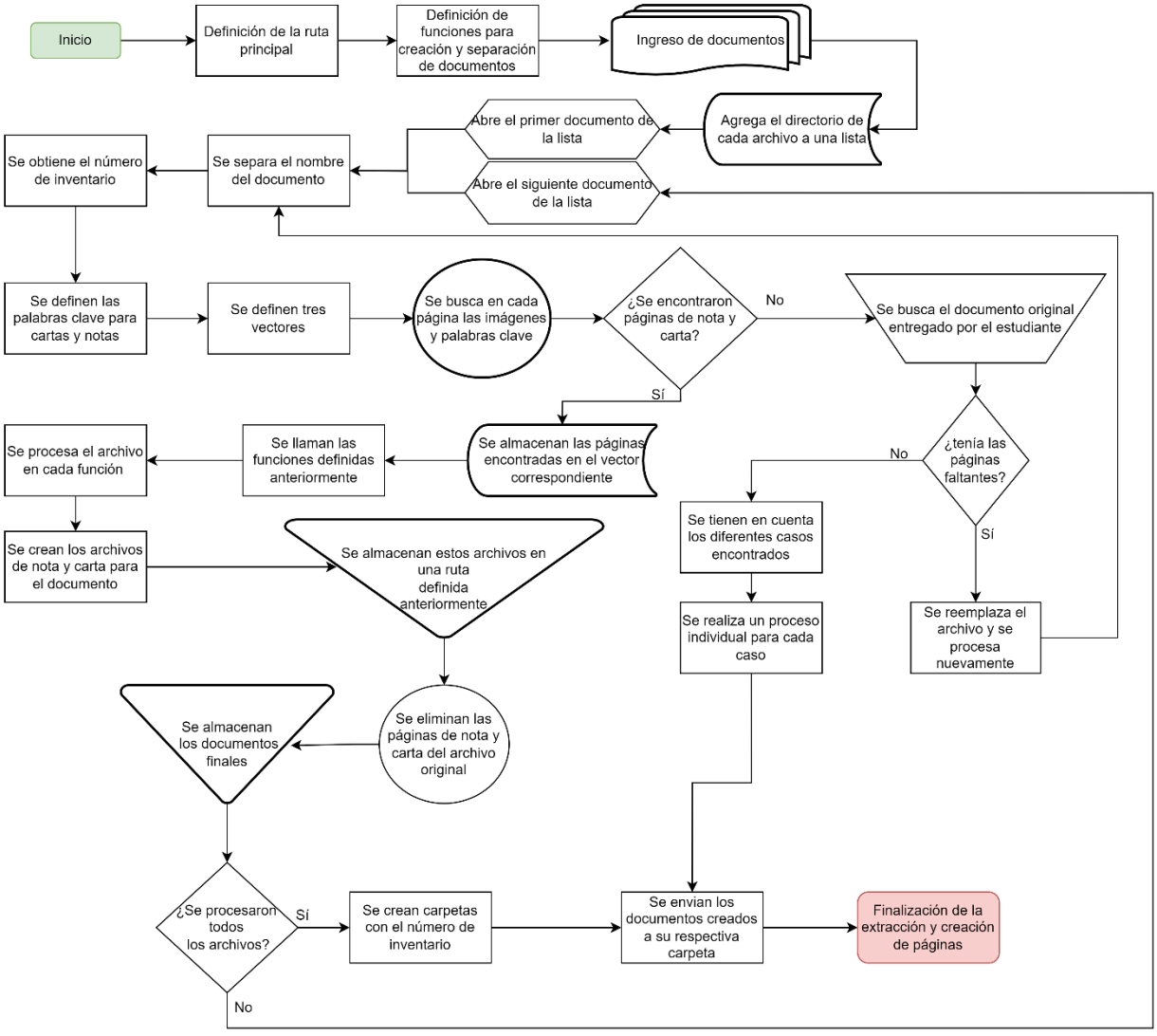
Así mismo se importaron Os, pikepdf, fitz, PdfFileWriter, PdfFileReader y tqdm.

Después se define una ruta principal desde donde se iban a procesar todos los archivos; de igual forma se definieron varias funciones dependiendo del número de páginas con nota de proyecto y carta de autorización; estas funciones se utilizaban para extraer las páginas almacenadas en los vectores y crear los tres archivos por separado. Después se recorrían todos los archivos de la ruta principal de nuevo y se separaba el nombre completo el cual era "#inventario.pdf" en dos partes y de esta manera ya se tenía el número de inventario de cada trabajo de grado. Avanzando con el algoritmo, se precisaron los términos clave de reconocimiento de texto que debía buscar dentro de cada documento. Se definieron tres vectores los cuales serían utilizados posteriormente para comparar las páginas con imágenes encontradas y el algoritmo de reconocimiento de texto. Se leía el texto en cada página para el documento procesado y se buscaban en cada página los términos clave gracias al método Load_Page. De igual manera, se define un límite para la lectura de ocho páginas por documento. Se realiza la búsqueda de todas los términos tanto de nota como de carta y respectivamente fueron agregadas al vector correspondiente; se compararon los vectores para evitar duplicidad de páginas dentro de los mismos y se agregaron los datos limpios de las páginas que se tenían que extraer. Se definieron dos nuevos vectores para llenar con las páginas con imágenes que iban a ser verificadas, al igual que con el algoritmo de reconocimiento de texto se tomó un rango de ocho páginas. Se implementa un algoritmo que buscaba en cada archivo PDF las imágenes que tenía dentro de las ocho primeras páginas y las agregaba a cada uno de los vectores dependiendo del tamaño del array. Luego se comparaban ambos vectores tanto del algoritmo de reconocimiento de texto como el de reconocimiento de imágenes, con el fin de dejar fijo un solo vector con las páginas que se debían extraer. Finalmente se llamaban las funciones definidas anteriormente y se les daba como argumento la ubicación del archivo con su dirección

completa y el vector con las páginas para realizar la extracción. Estos archivos quedaban almacenados en una carpeta aparte con el nombre que se definió anteriormente. El código fuente de este proceso de implementación se puede consultar en el Apéndice B. El diagrama de flujo de la herramienta se puede apreciar en la siguiente figura.

Figura 7

Diagrama de flujo de la herramienta para extracción de páginas y creación de archivos.



4.3.1.3 Desarrollo de herramienta de extracción de metadatos. Para el caso de la extracción de metadatos, se hace uso de las librerías PyPDF2 y Os instaladas previamente en la decodificación de los archivos y en la extracción de páginas.

Adicionalmente se instaló la librería pandas de la siguiente manera:

pip install pandas

Ya teniendo estas librerías se importan para el desarrollo. Se especifica la ruta donde se encuentran los archivos ya decodificados para su debido proceso, se lee el directorio de cada uno y se agregan en una lista, donde cada campo de esta corresponde al nombre de los archivos. Una vez por nombres, mediante un ciclo for se toman uno a uno y se abren.

Al momento de abrir el archivo se tenía una primera condición que restringía la lectura de las páginas a un rango de solo las primeras 30, donde si el documento estaba compuesto por menos páginas, el rango se igualaba al número total del documento, esto se hizo con el fin de no tener demoras a la hora de recorrer los archivos en busca de las páginas. Además, se define e inicializan las variables que iban a contener los metadatos de los archivos.

Ahora bien, ya definido el rango de páginas y las variables declaradas, se pasa a recorrer el archivo página por página hasta culminar el ciclo, donde al ingresar se declaran las variables extra definidas como variables temporales, se lee el contenido de dos páginas, la primera corresponde al número del ciclo y la segunda a la página siguiente, se declararon los arreglos que llevaban el contenido de las páginas, se reemplazaron los saltos de línea “\n” por caracteres que fueran identificables al momento de la búsqueda de metadatos y se empieza la búsqueda de las páginas que contenían los metadatos por medio de los términos clave ya definidos. Al momento de encontrar las páginas que contenían los metadatos, se envía su contenido a unas funciones que se encargan de la separación de estos, según lo requerido.

Dichas funciones se crearon mediante términos clave que delimitan los metadatos para detectar las posiciones en la lista que corresponden, por ejemplo, donde iniciaba el resumen y donde terminaba, tomando como ayuda las variables temporales encargadas de almacenar dicha

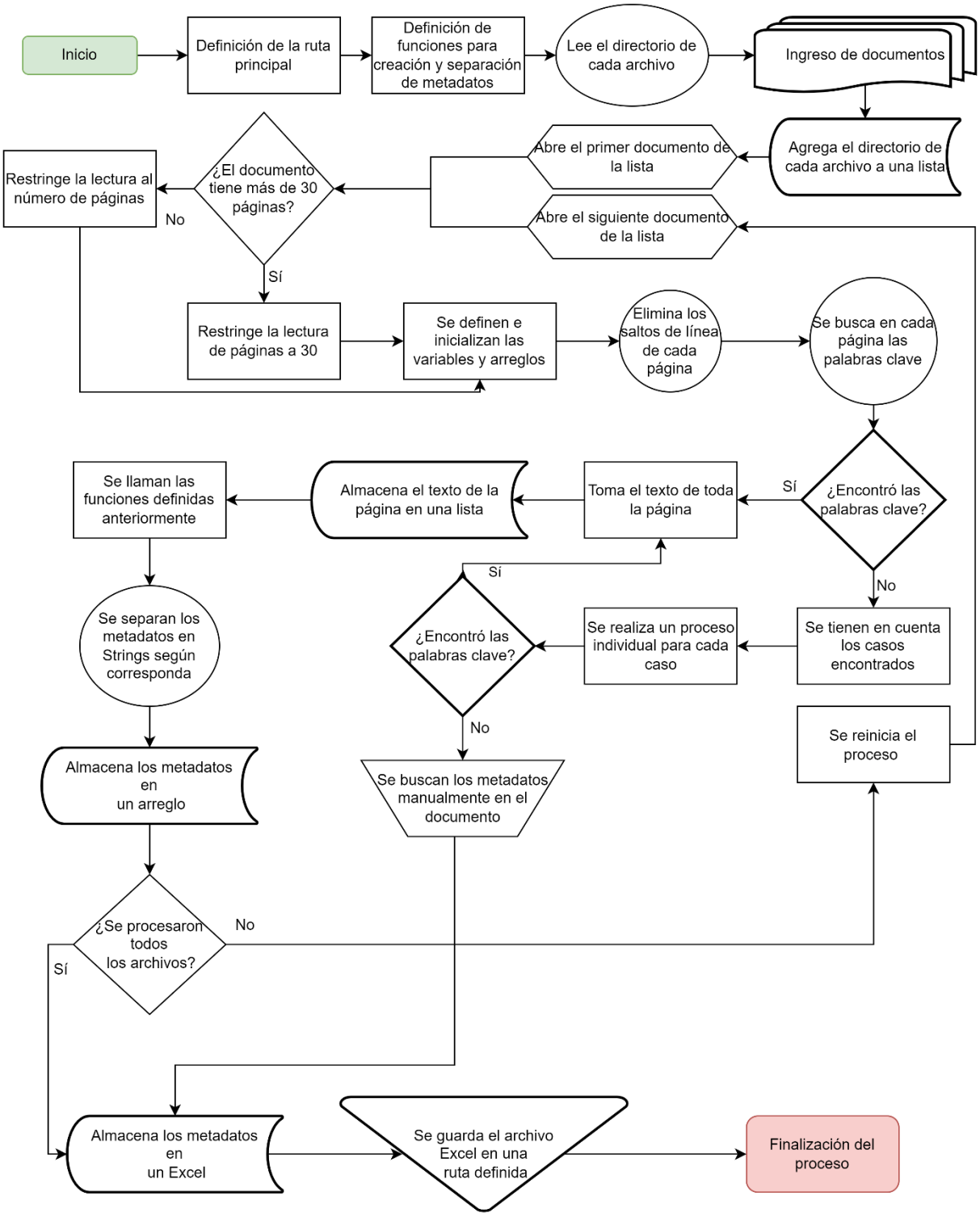
información. Al momento de realizar la separación, se recorrió nuevamente la lista teniendo en cuenta las posiciones, almacenando los metadatos en las variables a las que pertenecían.

Dichas variables que contenían los metadatos eran llevadas a un arreglo que almacenaba todos los datos de los archivos, y por último se reiniciaba la secuencia, inicializando nuevamente las variables para el siguiente archivo.

Al momento de haber procesado todos los archivos y con los metadatos contenidos en el arreglo, se creaba un archivo Excel, agregando lo necesario como último paso del desarrollo. Para comprender el proceso de implementación del código de extracción de metadatos se puede consultar el Apéndice C.

Figura 8

Diagrama de flujo para la herramienta de extracción de metadatos.



4.3.2 Pruebas de funcionamiento del sistema

Para realizar la prueba de funcionamiento de las herramientas de extracción de páginas y metadatos se utilizaron los archivos del año 2019, los cuales fueron copiados a la ruta principal definida previamente y se ejecutó el algoritmo. La demora en procesar estos archivos fue de aproximadamente 30 minutos para el caso de la extracción de páginas y alrededor de una hora para el caso de la ejecución del código para extraer metadatos. Posteriormente se seleccionan muestras aleatorias de documentos para verificar los resultados del algoritmo y se encuentra que la mayoría de los casos estaban correctos, en otros casos particulares, la extracción de páginas no fue la esperada.

Para detallar esta situación, se muestran todos los casos encontrados, incluidos los que obtuvieron el resultado esperado:

- El primer caso fue el escenario perfecto, en el cual la extracción de páginas y creación de archivos PDF salió correctamente. Se crearon los archivos de nota de proyecto y carta de autorización necesarios; de igual forma se eliminaron las páginas del documento copia.
- El segundo caso fue cuando el formato de carta de autorización estaba divido en dos páginas, es decir que, el autor no organizó el documento para que todo el formato quedara en una sola página si no que se distribuye en dos.
- El tercer caso es para los archivos que tenían un número diferente de páginas de cartas y notas, además estas estaban como imágenes.
- El cuarto caso es cuando no existían páginas de notas o de cartas. Había trabajos de grado los cuales no tenían alguna de las dos páginas necesarias.

- El quinto caso es para los archivos que no tenían ninguno de los dos archivos; se encontró que en algunos casos el algoritmo no encontraba términos clave en todo el archivo por lo que se procedió a verificar manualmente cada uno de estos y se encontró que no tenían páginas de carta ni de nota.
- El sexto caso fue para los trabajos de grado con más de tres autores. Se encontraron trabajos de grado con hasta ocho autores por lo que existían hasta ocho páginas de cada una, de cartas y notas.
- El séptimo caso fue para los trabajos de grado que tenían la página de carta antes que la de nota y estas eran imágenes. Anteriormente se definió que el formato para los trabajos de grado tenía las páginas de nota antes que las de carta, pero se encontraron trabajos de grado al revés.

En la Tabla 3 se puede apreciar la cantidad de trabajos de grado, tesis o disertaciones que resultaron correctas y con errores después de ejecutar el algoritmo para cada uno de los años, con el total de documentos procesados y se observa un promedio de 9.2% de porcentaje de error.

Tabla 3

Resultados de la ejecución del algoritmo en primera instancia y cantidad de tesis migradas a Noesis.

Año	Tesis correctas	Tesis con errores	Total	% Éxito	% Error
2004	1.023	95	1.118	91,5%	8,5%
2005	1.076	86	1.162	92,6%	7,4%
2006	1.105	106	1.211	91,2%	8,8%
2007	970	95	1.065	91,1%	8,9%
2008	1.256	157	1.413	88,9%	11,1%
2009	1.098	186	1.284	85,5%	14,5%
2010	1.478	153	1.631	90,6%	9,4%

2011	1.389	146	1.535	90,5%	9,5%
2012	1.480	197	1.677	88,3%	11,7%
2013	1.587	158	1.745	90,9%	9,1%
2014	1.859	171	2.030	91,6%	8,4%
2015	1.792	127	1.919	93,4%	6,6%
2016	1.754	172	1.926	91,1%	8,9%
2017	1.841	127	1.968	93,5%	6,5%
2018	1.799	147	1.946	92,4%	7,6%
2019	1.383	166	1.549	89,3%	10,7%
2020	1.167	143	1.310	89,1%	10,9%
Tesis Procesadas			26.489		

Al momento de realizar pruebas de funcionamiento al algoritmo que extrae metadatos, se observa que se tenían muchos metadatos faltantes, puesto que había diferencias en la estructura de ciertos archivos ya que los términos clave no funcionaban para todos. Respecto a esto se realizó un análisis de los documentos que no arrojaban resultados según lo esperado, obteniendo como resultado cinco posibles casos que se debieron considerar y son resaltados a continuación:

- El primero de los casos fue el ideal planteado en el análisis que se realizó de los trabajos de grado, donde a partir de los términos clave “resumen, autor, palabras clave y descripción” se identificaban las páginas que contenían los metadatos para la mayoría de los documentos.
- El segundo caso es muy similar al anterior, pero se pudo observar la ausencia de uno de los términos clave “descripción” usándose solamente los términos “resumen, autor y palabras clave” para identificar las páginas de los metadatos.
- Para el tercero de los casos en sí representa una minoría de los archivos, donde la página de los metadatos en ingles no se encontraba en la página siguiente a la de

los metadatos en español, sino que se encontraba antes de esta, cumpliéndose la estructura de los términos clave.

- El cuarto de los casos corresponde a los archivos que era imposible la lectura del contenido de las páginas por medio de las librerías de extracción de texto, donde fue necesario la implementación de la herramienta de conversión a imágenes y la herramienta de OCR con el fin de identificar las páginas y procesar el texto.
- El último de los casos concierne a los archivos en los que no se pudo identificar con ninguno de los anteriores casos y por ende no se extrajo información. La solución plateada fue abrir los archivos y realizar la extracción de forma manual.

Después de terminar las correcciones y las mejoras al algoritmo, se obtuvo una versión final que arrojaba mejores resultados. Estos se representan en la Tabla 4 donde se puede apreciar la cantidad de trabajos de grado, tesis o disertaciones que resultaron correctas y con errores después de ejecutar el algoritmo para cada uno de los años, con el total de documentos procesados y se observa un promedio de 14.6% de porcentaje de error.

Tabla 4

Tabla de resultados ejecución algoritmo extracción de metadatos por años

Año	Tesis correctas	Tesis con errores	Total	% Éxito	% Error
2004	987	131	1.118	88,3%	11,7%
2005	1.012	150	1.162	87,1%	12,9%
2006	930	281	1.211	76,8%	23,2%
2007	847	218	1.065	79,5%	20,5%
2008	1.198	215	1.413	84,8%	15,2%
2009	1.106	178	1.284	86,1%	13,9%
2010	1.455	176	1.631	89,2%	10,8%

2011	1.385	150	1.535	90,2%	9,8%
2012	1.501	176	1.677	89,5%	10,5%
2013	1.586	159	1.745	90,9%	9,1%
2014	1.783	247	2.030	87,8%	12,2%
2015	1.649	270	1.919	85,9%	14,1%
2016	1.666	260	1.926	86,5%	13,5%
2017	1.652	316	1.968	83,9%	16,1%
2018	1.640	306	1.946	84,3%	15,7%
2019	1.303	246	1.549	84,1%	15,9%
2020	1.005	305	1.310	76,7%	23,3%
Tesis Procesadas			26.489		

4.3.3 Correcciones y mejoras de la herramienta software

Algunas de las correcciones realizadas al algoritmo para extracción de páginas se basaron en los casos presentados en la fase de pruebas de funcionamiento del sistema. A continuación, se mencionan las mejoras:

- Para el segundo caso, se encuentra que había páginas que no se podían clasificar por lo que se tuvieron que implementar nuevos métodos en la detección de imágenes. El algoritmo actual no tenía la forma de revisar la segunda página ya que, si no encontraba algún término clave o imagen en un rango, no lo tomaba, entonces se corrigió el algoritmo y se realizaron algunas comprobaciones para detectar el tamaño de imágenes que había en el rango y de esta manera se clasificaron en nota o carta para realizar posteriormente la creación de archivos.
- Para el tercer caso se realizó la clasificación de páginas de forma manual ya que no se tenía la manera de clasificar las imágenes. El algoritmo simplemente las extraía y localizaba, pero no definía cual página era de nota y cuál de carta. Esta clasificación se realizó de forma manual pero masiva teniendo en cuenta que se

creaban carpetas con archivos con características similares y posteriormente se extraían esas páginas dependiendo de la posición de estas de forma masiva y no individual.

- Para el cuarto caso se tuvo que buscar el archivo original en el disco del trabajo de grado entregado por el estudiante para verificar que efectivamente no tuviera carta o nota dependiendo del caso y agregarla en caso de que si existiera. Este proceso se realizó manualmente y se comprobó para cada archivo. En caso de que no existiera la página con nota o carta se debían subir los archivos de esa manera. Finalmente se migraban dos documentos para ese número de inventario y no tres.
- Para el quinto caso se realizó una verificación manual y se encontraron varios archivos que no tenían ninguna de las dos páginas de interés. Fue necesario recurrir a los discos entregados por el estudiante con los archivos originales para verificar que no tuvieran esas páginas. En caso de que así fuera se debía migrar el archivo original entonces se desarrolló otro algoritmo que simplemente desencriptaba el documento y creaba una copia nombrada con #inventario-SP.
- Para el sexto caso fue necesario agregar un nuevo método dentro del algoritmo para verificar la cantidad de páginas de nota y carta que tenía el archivo; de esta manera crear los archivos PDF y realizar la eliminación de páginas del archivo copia. Posteriormente se verificaron uno a uno los archivos generados en este caso para comprobar que el resultado fuera el esperado.
- Para el séptimo caso se realizó una clasificación manual para los archivos con imágenes dentro. El ajuste de páginas se ejecutó por medio de otro pequeño

algoritmo el cual eliminaba las páginas dadas y creaba archivos PDF con estas tal y como se le indicara.

Al momento de realizar las pruebas de funcionamiento y partiendo de los casos que se obtuvieron como resultado, se implementaron nuevos métodos a la herramienta de extracción de metadatos, los cuales ya integraban el uso de OCR.

Para esto se tuvieron que instalar las librerías pdf2image, PIL y pytesseract de la siguiente manera:

pip install pdf2image PIL pytesseract

- Para dar solución a los casos, al momento que se terminara de procesar un archivo, si hacía falta uno de los metadatos entonces pasaba a utilizar el OCR. Este caso se trabajaba en conjunto con el lector de texto donde se leía el contenido del archivo, se identificaban las páginas y separaban del archivo original en un documento aparte, esto con el fin de no tener que trabajar nuevamente con todo el archivo original. Se tomaba el nuevo archivo creado (que contenía dos páginas, metadatos en español y metadatos en inglés) y se pasaba a convertir en imágenes con ayuda de la librería ***pdf2image*** mediante el parámetro ***convert_from_path.*** Ya teniendo las imágenes se abrían con la librería PIL y se pasaba a extraer el texto con la función OCR de Pytesseract. De esta manera se solucionó una de las primeras dificultades.
- Para otro de los casos, se tenía que él extractor de texto plano no identificaba las páginas que contenían los metadatos ya sea porque estaban en formato de imagen o simplemente el formato del archivo era incompatible con la librería. Para dar solución a esto, se implementó una nueva versión que, al momento de terminar de

revisar un archivo si no se tenían páginas identificadas, se recorrían las páginas de dicho archivo una a una en un rango que iniciaba desde la página 4 hasta la 30, donde se convertía cada una de ellas en imagen, se extraía el texto y se analizaban si correspondían a las páginas de interés.

4.3.4 Extracción masiva de páginas identificadas y metadatos

Para realizar la extracción masiva de páginas se comenzó por el año 2021 hasta llegar al año 2004. Se copiaron los archivos originales a la ruta principal para ejecutar el algoritmo y así sucesivamente con los siguientes años. A medida que iba terminando un año se eliminaban los archivos y se copiaban a la ruta principal los del siguiente año. Cabe aclarar que antes de realizar este proceso se desencriptaron todos los archivos de todos los años utilizando el algoritmo para desencriptar definido en la fase **4.3.1.1 Decodificación de documentos.**

Para la extracción masiva de metadatos se inició desde el año 2020, ya que debido a la pandemia se generó un cambio en la entrega de trabajo de grado a Biblioteca, de formato CD a diligenciamiento mediante formulario, en el cual se empezaron a recibir los metadatos correspondientes al resumen y el abstract en documentos aparte. Partiendo de la decodificación de los archivos, se extrajeron los metadatos que se solicitaban, complementando con los metadatos que se extrajeron de la base de datos y se unificaron en un solo archivo Excel.

Ya teniendo todos los metadatos necesarios, se hicieron algunas correcciones a ciertos datos tales como capitalizar los nombres de los autores, pasar los títulos a minúscula, eliminar los espacios al inicio de cada uno de los campos en el Excel y se verificó que no hubieran campos vacíos en los metadatos principales, considerando que debían cumplir lo siguiente: al menos debía tener un autor, un director y el título en español; los campos de idioma, programa, escuela, facultad, título obtenido, año y nivel de estudio son obligatorios. Para el resto de los metadatos, resumen,

palabras clave, title, key words y abstract, podían omitirse de cierta forma ya que algunos documentos no lo poseían y debía tener la etiqueta N/A para ser identificados.

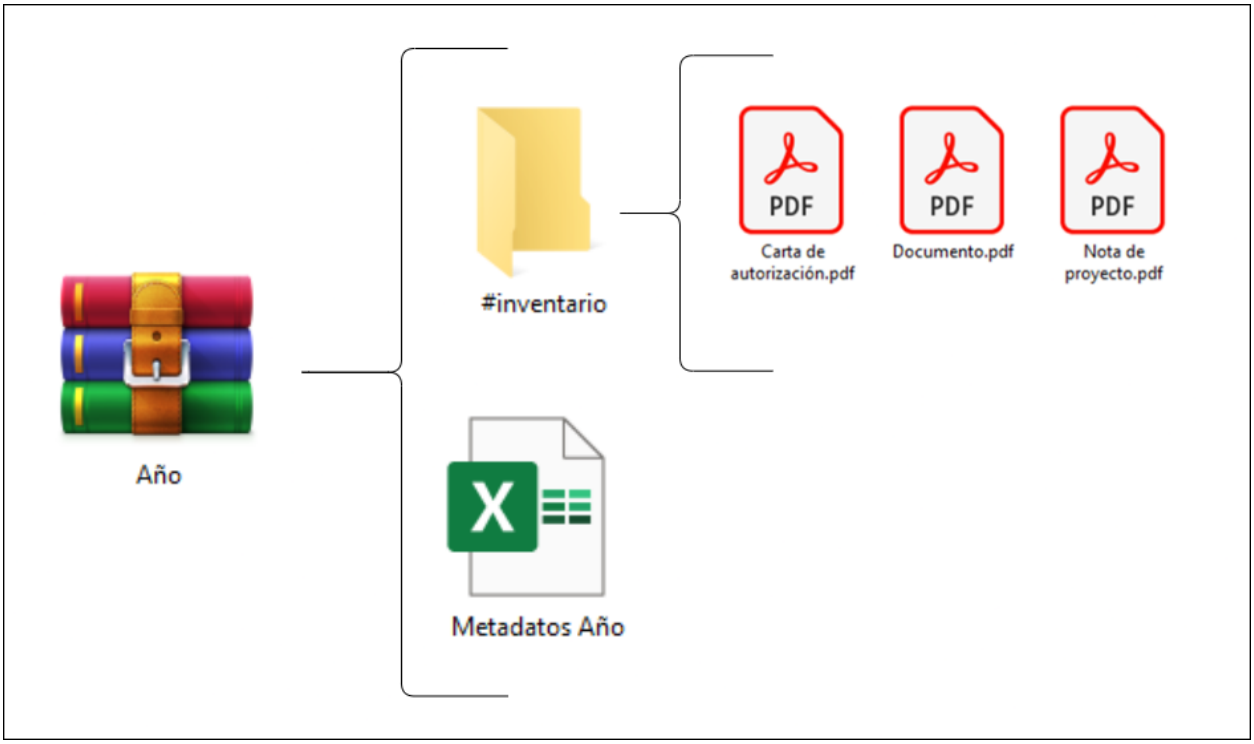
4.4 Migración de trabajos de grado optimizados

4.4.1 Organización final de trabajos de grado

Para realizar la migración se cuenta con el apoyo del proveedor del servicio y administradores de DSpace, los cuales solicitaban los archivos de una forma específica, por lo cual fue necesario que los archivos ya separados estuvieran nombrados como se iban a recuperar en el sistema y guardados en carpetas con el número de inventario correspondiente, todo esto dentro de un .RAR nombrado con el año al que pertenecían. Para cumplir este requerimiento fue necesario desarrollar otra herramienta la cual creaba una carpeta por cada archivo PDF y añadía a cada una de ellas los archivos separados a los que pertenecían con el nombre necesario para realizar el cargue a Noesis; dentro de cada carpeta por años se incluía el archivo de metadatos necesario. De esta manera ya se tenían listos los archivos para realizar la respectiva migración, que sería realizada gradualmente por años. Los archivos finalmente eran organizados tal y como se puede observar en la siguiente figura.

Figura 9

Organización final de archivos para migración.



4.4.2 Migración de metadatos y documentos

De acuerdo con lo planteado, se inicia la migración por años, después de ejecutar el algoritmo y tener listos los archivos y metadatos de cada uno, se comienza por el año 2020 partiendo de que algunos documentos ya no tenían páginas de nota y carta dentro del archivo original, seguido del año 2019 que fue tomado como prueba para el cargue inicial, donde se requería realizar desde ceros todo el proceso mencionado en este proyecto.

Se realiza la migración compartiendo a los administradores de Dspace el archivo .RAR con los documentos y el archivo Excel que contenía los metadatos. En el momento que los administradores de DSpace recibían los archivos con su respectiva información referencial, debían hacer una pequeña modificación en el esquema de los metadatos, lo cual realizaban mediante un

código en Python que les permie complementar y adaptar algunas etiquetas de acuerdo con las especificaciones de Dublin Core, esto les ayudaba en la preparación final para su debido cargue. También se agrupaban todos los autores en una sola columna con un separador y de igual forma todos los directores. Se agregaron nuevos metadatos tales como qué tipo de trabajo es, la versión de COAR, los metadatos del tipo de COAR, los rights, tipo de atribución, entre otros.

Con este script de Python se clasificaron por el tipo de colección a la que pertenece cada uno de los archivos, se les asigna un ID que los identifica y se estandarizan según los requerimientos de Dublin Core. Junto a esto también se generaron los directorios que se encargaban de hacer un seguimiento de lo que sucedía al momento de procesar los archivos con sus datos, esto se realizó dentro del servidor.

Después de todo el proceso de modificación y clasificación de la información, junto al alistamiento de los archivos de acuerdo con lo requerido por Dspace, se ejecuta un comando que testea que los documentos a subir no tuvieran errores, si uno de los registros arrojaba error, se debía revisar de forma manual y si el resultado era satisfactorio finalmente se ejecutaba el siguiente comando de cargue al repositorio para realizar la migración.

cd /home/dspace/dspace: Primero situarse en la carpeta del dspace

bin/dspace import -a -e [usuario] -c [identificador coleccion] -s [ubicación carpeta programa] -m [ubicación carperta mapfile] -t

Donde *bin/dspace import* es el comando de importación

-a --add permite añadir registros al repositorio

-e --eperson corresponde al usuario que realiza la operación

-c --collection es el Handle o identificador de programa o colección

-s --source es el directorio del programa o colección

-m --mapfile es el directorio del mapfile del programa o colección

-t --test permite comprobar que los registros están correctos para ser subidos

Para mayor detalle de este proceso, se puede revisar el Apéndice D sobre migración de documentos al Repositorio Noesis.

Figura 10

Evidencia de trabajo de grado migrado al Repositorio Institucional.

Nota. La figura muestra uno de los documentos migrados al Repositorio Institucional Noesis. Tomado de *https://noesis.uis.edu.co/handle/20.500.14071/13356*.

4.4.3 Revisión y depuración de los resultados

Al finalizar la migración por años se realizó una revisión exhaustiva de los trabajos de grado migrados, tomando como referencia el año 2019 que fue utilizada para las pruebas, se encuentra que de los archivos migrado había siete documentos con error, ya que las carpetas no tenían los archivos necesarios para la migración, entonces no se había realizado el cargue. Para solucionar este error fue necesario desencriptar los archivos originales y migrar manualmente estos documentos.

4.5 Requisitos para las integraciones y compatibilidad

4.5.1 Integración con redes de colaboración

En cuanto a la integración con redes de colaboración, se realiza una revisión en las plataformas web oficiales de cada recurso, encontrando, por ejemplo, la manera en que Google Scholar realizaba la indexación automática de repositorios. Se observa que para realizar dicha indexación se tenían que cumplir las siguientes directrices en la configuración del Repositorio (Google Scholar, s.f.):

1. **Formato de los archivos:** Los archivos dentro del repositorio deben estar en formato HTML o PDF.
2. **Interfaz de navegación:** Una interfaz de navegación es necesaria para que los robots encuentren los enlaces de los artículos.
3. **Disponibilidad del sitio web:** Se necesita tener disponibilidad del sitio web ya que los robots visitan la página del repositorio periódicamente para realizar actualizaciones del contenido y de igual manera verifican que los enlaces estén disponibles.

4. **Protocolo de exclusión de robots:** Si el sitio web del repositorio usa un archivo robots.txt se le deben dar algunos permisos a los robots de Google Scholar para que tengan acceso a la página del repositorio. Estos tendrán acceso a los enlaces de cada documento, a los metadatos a primera vista de cada documento tales como resumen, autores, fecha de cargue, palabras clave y el documento como tal del trabajo de grado. Los permisos necesarios son los siguientes:

   User-agent: Googlebot

   Allow: /

Al realizar esta revisión con el administrador del sistema, se encuentra que dentro de la configuración general de Noesis se tenían bien todas las opciones, excepto la última. Revisando el archivo robots.txt del repositorio, se encuentra que no estaban definidos los permisos a los bots de Google, por lo cual fue necesario solicitar este requerimiento y así cumplir con todos los requisitos para la integración con Google Scholar.

Por otro lado, en cuanto a la integración con REDCOL, la Red Colombiana de Información Científica, el proceso para ser parte de la Red se compone en primera instancia de firmar la carta de compromiso presente en la página web de REDCOL y registrar la institución, el repositorio y el personal responsable utilizando el formulario de vinculación. (REDCOL, s.f.)

Este proceso ya se había realizado en el año 2020, pero debido a que esta versión del repositorio sufrió varios cambios por errores dentro de la migración pasada y la falta de soporte durante la pandemia, esta integración había quedado suspendida durante todo ese tiempo hasta que se vuelve a retomar en este proyecto, mencionando que aunque se había realizado un diagnóstico inicial previamente, era necesario hacerlo nuevamente, ya que la configuración de los metadatos y la migración de los documentos dentro de Noesis eran totalmente diferentes. Teniendo en cuenta

los correos e información suministrada por REDCOL sobre la integración del año 2020, se encuentran los resultados del diagnóstico previo, en el cual no indexaba ningún documento por problemas con los metadatos. El formato definido de acuerdo con las directrices establecidas no era el correcto para realizar la integración, algunos metadatos no contenían la información correspondiente y otros simplemente no estaban habilitados, por esto, a la hora de realizar la indexación no los encontraba y fallaba totalmente el proceso.

Al retomar esta información para verificar el estado actual de los metadatos con el fin de solicitar un nuevo diagnóstico, se encuentra que hace falta un metadato necesario para definir el nivel de acceso de los documentos, correspondiente al *dc.rights.accessrights.* Este metadato debe tener la siguiente configuración en formato Dublin Core:

dc.rights.accessrights : info:eu-repo/semantics/openAccess

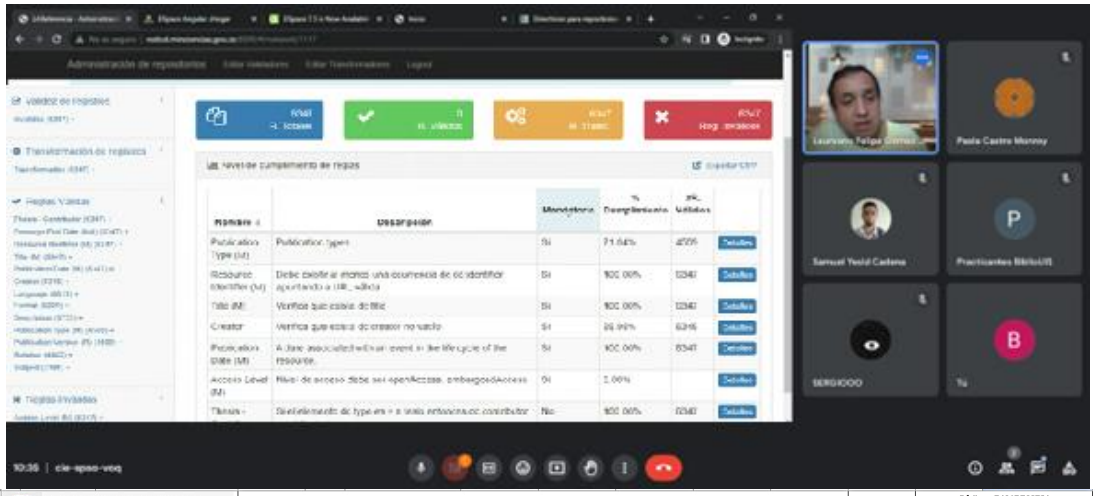
Se busca la manera de agregar este metadato a todos los artículos migrados y modificar la información de los que ya estaban almacenados en el repositorio. En colaboración con los administradores de Dspace, se gestiona la inclusión de este metadato, mediante reunión virtual con el equipo a cargo se trataron varios aspectos, entre ellos la inclusión del metadato por medio de la base de datos para realizar un cargue global a todas las colecciones de Noesis.

Posteriormente se programa reunión virtual con el equipo de soporte de REDCOL para definir cuál era el siguiente paso en la integración. Se tiene en cuenta que REDCOL solo cosecha tesis de maestría, tesis de doctorado, artículos de revista e informes finales de proyectos de investigación, por lo cual, en Noesis solo se recuperaría información de las comunidades de “Tesis y Disertaciones” y “Revistas Científicas UIS”. De acuerdo con esta información, se realiza un nuevo intento de cosecha de las tesis y disertaciones por parte de los encargados de REDCOL y se

presentaron los resultados en una reunión realizada por Google Meets. La asistencia a la reunión se puede apreciar en la siguiente figura.

Figura 11

Evidencia de reunión virtual Biblioteca UIS - REDCOL.



MINISTERIO DE CIENCIA, TECNOLOGÍA E INNOVACIÓN	ASISTENCIA A REUNIONES	Código: D101PR03F01
		Versión: 01
		Fecha: 2020-10-29

Dependencia:	Dirección de Capacidades y Divulgación de la CTeI
Tema:	Asistencia UIS
Objetivo:	Asistencia UIS

Fecha:	2/6/2023	Hora inicio:	9:00	Hora finalización:	11:00	Lugar:	

				ENFOQUE DIFERENCIAL E INCLUYENTE							
No	NOMBRE	CARGO	DEPENDENCIA Y/O ENTIDAD	Población Víctima del Conflicto	Población en Situación de Discapacidad	Sexo	Grupo Étnico	Edad	TELÉFONO/Ext.	E-MAIL
1	Linda Paola Castro Monroy	contratista	Dirección de Capacidades y Divulgación de la CTeI						5824	pcastro@miniciencias.gov.co
2	Sergio Roa	Practicante	Biblioteca UIS		Ninguna	Hombre	Ningún grupo étnico	24	3223151622	sergio2162112@correo.uis.edu.co
3	Ferney Mauricio Calderón	Profesional Biblioteca	Biblioteca		Ninguna	Hombre	Ningún grupo étnico	-1	3162334445	ferneymc@uis.edu.co
4	Samuel Yesid Cadena Pinilla	Practicante	Biblioteca UIS		Ninguna	Hombre	Ningún grupo étnico	25	3188905291	samuel2171399@correo.uis.edu.co

De acuerdo a lo establecido en el artículo 12 la Ley 1581 de 2012, la información registrada en este documento tiene como propósito verificar su asistencia y asuntos relacionados con el objeto de la reunión. Adicionalmente de acuerdo a lo establecido en el artículo 2 de la Ley 1712 de 2014, esta información se encuentra en el marco del principio de máxima publicidad para titular universal

En esta nueva cosecha se encuentra que se intentan recuperar aproximadamente 6.347 registros de tesis y artículos de revistas en Noesis, pero al final ninguna es validada correctamente para ser indexada. Se presentan los resultados del nuevo diagnóstico y se encuentra que faltaban algunas configuraciones dentro de Noesis para que se integrara correctamente. Una de las principales recomendaciones es que se detecta que Noesis está en una versión de DSpace 7.1 y se debería actualizar a la última versión vigente de DSpace 7.5. En cuanto a las configuraciones faltantes en Noesis, el diagnóstico muestra que faltaban implementaciones del protocolo OAI-

PMH. El enlace al protocolo estaba configurado así: **https://noesis.uis.edu.co/server/oai**. La configuración correcta debía ser: **https://noesis.uis.edu.co/oai**. También se encuentra que la configuración general del protocolo mostraba errores con el enlace con el que se estaba conectando, ya que este no se mostraba correctamente la configuración, como se puede apreciar en la siguiente figura.

Figura 12

Error al ingresar en el enlace del protocolo OAI-PMH.

De igual manera se encuentra que el protocolo y los vocabularios de OpenAire 3 y 4 no se encuentran configurados correctamente, ya que las tesis no tenían la tipología normalizada según COAR OpenAire v3 y v4 para el tipo de colección, lo cual se debe configurar de acuerdo con las directrices establecidas para detectar el tipo de documento (https://redcol.readthedocs.io/es/latest/field_publicationtype.html) y el nivel de acceso abierto (https://redcol.readthedocs.io/es/latest/field_accessrights.html)

Por otra parte, la localización del identificador persistente HANDLE en el repositorio no llamaba directamente a handle.net por lo que se debía configurar correctamente la URI.

Teniendo en cuenta el diagnóstico presentado en esta reunión y que la implementación de estos cambios no depende directamente del personal de la Biblioteca sino del proveedor del servicio y administrador de Dspace, se informan las modificaciones sugeridas en la configuración para realizar las respectivas correcciones y dar cumplimiento a los requisitos de integración con REDCOL.

Finalmente, para el caso de la integración con La Referencia a nivel de América Latina, se menciona que está toma directamente los datos de un nodo en Colombia, el cual es precisamente REDCOL, por lo que al realizar la integración con este e indexar y cosechar todos los datos de Noesis allí, automáticamente quedaría integrado con La Referencia.

4.5.2 Integración con perfil de investigación

La integración con el perfil de investigación era un poco más complicada, ya que no dependía solo de configuración de los metadatos, sino que para realizar la integración de Noesis con ORCID para lograr el intercambio automatizado de información y la interoperabilidad entre sistemas, era necesario actualizar la versión de DSPACE de 7.1, en la que se encontraba actualmente, a una versión superior a 7.3 o 7.5.

Para realizar esta actualización era necesario detener el cargue de nuevos trabajos de grado durante una semana, pero en ese momento se estaba llevando a cabo el proceso de presentación y entrega de trabajos de grado, tesis o disertaciones a la Biblioteca, de acuerdo con el calendario de grados de la Universidad, por lo que no era posible detener el servicio por este tiempo. Teniendo en cuenta esta problemática fue necesaria la creación de un servidor de pruebas, implementado por los administradores de Dspace, con una copia total de Noesis, que se podía acceder internamente mediante el enlace: **noesisdev.uis.edu.co**.

En este repositorio copia se instala la versión necesaria para la integración con ORCID, de acuerdo con la documentación de DSpace, la cual se puede encontrar en su sitio oficial: https://wiki.lyrasis.org/display/DSDOC7x/ORCID+Integration.

Para llevar a cabo este proceso fue necesario la comunicación con el proveedor de la suscripción de ORCID, el Consorcio Colombia, para solicitar el código del Client ID y Client Secret; quienes mencionan que se debía diligenciar el formulario de solicitud de información técnica de Dspace: https://info.orcid.org/register-a-client-application-sandbox-member-api/.

Posteriormente se recibe respuesta a este formulario por medio de correo electrónico, en el cual se mostraban las credenciales necesarias para realizar la configuración inicial del Sandbox, con el Client ID y un Client Secret. Esta información es compartida con el administrador del servicio, quien realiza las respectivas pruebas y corrobora que el VPN al que está conectado el servidor no presentara problemas, ejecutando correctamente la integración.

5. Conclusiones

La práctica empresarial en la Biblioteca de la Universidad Industrial de Santander fue una excelente oportunidad para aplicar los conocimientos recopilados durante la carrera de Ingeniería de Sistemas en todo el proceso de mejoras en el Repositorio Institucional, relacionado con la migración de documentos e integración con redes de colaboración.

Se analiza la estructura de los trabajos de grado, tesis o disertaciones comprendidos entre los años 2004 y 2021 disponibles en el Sistema de Gestión de Bibliotecas LIBRUIS con el fin de diseñar la estrategia de migración más adecuada, que permite respetar la privacidad y derechos de autores, directores y calificadores allí presentes.

Se desarrollan herramientas software en lenguaje de programación Python, para lograr la extracción de metadatos a través del reconocimiento de texto y la extracción de páginas especificas en los documentos para la creación de nuevos archivos por separado.

Mediante las herramientas desarrolladas, se procesan un total de 26.489 documentos que fueron clasificados por año de entrega, con un promedio de éxito de 85.4% procesados correctamente en la extracción de metadatos y un promedio de 90.8% en la extracción de páginas automáticamente del documento principal. Para el resto de los casos reportados con error, se realizan las mejoras pertinentes para finalmente dejar todos listos y organizados de acuerdo con los requerimientos para la migración.

Con apoyo del proveedor del servicio, quien administra el sistema DSpace, se realizan los primeros cargues de prueba a Noesis, con los trabajos de grado, tesis y disertaciones optimizados para tal fin; teniendo en cuenta esta migración, se implementan las mejoras detectadas para los próximos cargues de información.

Se establecen los requerimientos necesarios para realizar la integración con Google Scholar, REDCOL, La Referencia y el perfil investigador ORCID, con el fin de generar mayor visibilidad de la producción científica y académica de la Universidad.

6. Recomendaciones

A continuación, se presentan algunas recomendaciones a futuro relacionadas con la migración realizada y en general, mejoras en Noesis.

Con apoyo del personal de procesos técnicos de la Biblioteca, realizar revisiones periódicas aleatoriamente a los registros disponibles en Noesis, con el fin de detectar y mejorar los que posiblemente requieren cambios manuales en catalogación de metadatos, redacción u organización de la información referencial disponible.

Continuar con la migración a Noesis de toda la producción intelectual de estudiantes, representada en los trabajos de grado, tesis y disertaciones desde el 2004 que cuentan con el permiso de publicación y gestionar los permisos respectivos para las que se encontraban en físico, que fueron digitalizadas, con el fin de migrar también al Repositorio Institucional y poder disponer de toda la información en un solo lugar, desde la primera promoción de la Universidad.

Revisar y llevar un control detallado de los documentos que se encuentran en estado de confidencialidad, con el fin de tener en cuenta su vencimiento y en ese momento poder realizar este proceso de optimización y migración a Noesis.

Evaluar la posibilidad de integración de Noesis con otras redes de colaboración internacionales para generar mayor visibilidad del contenido intelectual de la Universidad.

Revisar y complementar la información de las otras colecciones de Noesis, como el contenido didáctico y multimedia de la Universidad y todo lo relacionado a eventos académicos y culturales, como videos de conciertos, Festivales de Piano, entre otros.

Referencias Bibliográficas

About Google Scholar. (s. f.). Google Scholar. https://scholar.google.es/intl/es/scholar/about.html

Braña, E. (2015, 25 marzo). *Dspace en las Universidades españolas | Hablando de DSpace*. Arvo Consultores. https://www.arvo.es/dspace/dspace-universidades/

Calderón, F., & Parra, L. (2013). *DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL REPOSITORIO INSTITUCIONAL EN LA BIBLIOTECA DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER* (Tesis de pregrado). Universidad Industrial de Santander, Bucaramanga, Colombia.

Cambridge University Press. (s. f.). *Using ORCID*. Cambridge Core. https://www.cambridge.org/core/services/authors/journals/using-orcid

Chauí, M. (1997). O ideal científico e a razão instrumental. En *Convite à Filosofia* (pp. 354-365). São Paulo: Ed. Ática.

¿Cómo ser parte de la Red? (s. f.). Red Colombiana de Información Científica (REDCOL). https://redcol.minciencias.gov.co/vufind/Content/como_ser_parte_de_la_red

Crow, R. (2002). The Case for Institutional Repositories: A SPARC Position Paper. *ARL Bimonthly*, Report 223.

Díaz, L., & Parra, L. (2011). *CONSIDERACIONES PARA LA CREACIÓN DE UN REPOSITORIO INSTITUCIONAL EN LA UNIVERSIDAD INDUSTRIAL DE SANTANDER* (Tesis de pregrado). Universidad Industrial de Santander, Bucaramanga, Colombia.

Dorta-Contreras, A. (2016). Visibilidad de la producción científica publicada por autores del Hospital Universitario «General Calixto García» en Scopus. 1972-2014. *Revista Habanera de Ciencias Médicas*, *15*(1), 123-135.

DSpace. (s. f.). *About DSpace*. DSPACE. https://dspace.lyrasis.org/about/

Inclusion Guidelines for Webmasters. (s. f.). Google Scholar Help. https://scholar.google.com/intl/es/scholar/inclusion.html#crawl

Instalación de Poppler y Tesseract en Windows. (2021, 13 julio). ConTexto. https://ucd-dnp.github.io/ConTexto/versiones/master/instalacion/instalacion_popple_teseract_windows.html

Ley 1581 de 2012: Por la cual se dictan disposiciones generales para la protección de datos personales. Octubre 17 de 2012 (DO. N° 48587). (s. f.).

Lynch, A. (2003). Institutional Repositories: Essential Infrastructure For Scholarship In The Digital Age. *ARL Bimonthly*, *3*(2).

Manjarrez Antaño, A. C., Martínez Castro, J. M., & Cuevas Valencia, R. E. (2014). Migración de Bases de Datos SQL a NoSQL. *Tlamati*, *3*, 144-148.

Mosquera, C., Fernández, S., & Lorenzo, E. (s. f.). INTEGRANDO EL REPOSITORIO E-IEO Y ORCID. En *Ecosistema del Conocimiento Abierto* (pp. 273-284). Ediciones Universidad de Salamanca.

Open Researcher and Contributor ID (ORCID). (2021, 11 diciembre). About ORCID. https://info.orcid.org/what-is-orcid/

Ortiz, S. A., & Gamboa, S. C. (2014). *COMPONENTE SOFTWARE PARA LA MIGRACIÓN DE LA BASE DE DATOS DE REGISTRO ACADÉMICO A COLECCIONES MONGODB* (Tesis de pregrado). Universidad Industrial de Santander, Bucaramanga, Colombia.

Python Package Index. (2022, 16 agosto). *pytesseract*. Python Package Index (PyPI). https://pypi.org/project/pytesseract/

Red Colombia de Información Científica. (REDCOL). (s. f.-a). Objetivos. https://redcol.minciencias.gov.co/vufind/Content/objetivos

Red Colombia de Información Científica. (REDCOL). (s. f.-b). Quiénes somos. https://redcol.minciencias.gov.co/vufind/Content/Quienes_somos

Repositorio de la Universidad Industrial de Santander (Noesis). (s. f.). DSpace Angular:: Home. https://noesis.uis.edu.co/home

Salomón, Y. P., & Rodríguez, A. M. (2007). Producción científica. *Ciencias de la Información*, *38*(3), 33-38.

Ub-Mannheim. (s. f.). *Home*. GitHub. https://github.com/UB-Mannheim/tesseract/wiki

Universidad Autónoma del Caribe (UAC). (2015). Repositorio Digital Institucional. http://repositorio.uac.edu.co/

Universidad de Palermo (UP). (2001). Repositorio de la universidad de Palermo. https://dspace.palermo.edu/dspace/

Apéndices

Apéndice A. Algoritmo creado para desencriptar archivos.

```
import os
import pikepdf#librería necesaria para desencriptar los documentos
workdir = ".//trabajosdegrado"#ruta principal
with os.scandir(workdir) as ficheros:
  ficheros = [fichero.name for fichero in ficheros if fichero.is_file()]
for p in range(len(ficheros)):
  DIR=".//trabajosdegrado/"+ficheros[p]#ruta específica para cada archivo
  pdf=pikepdf.open(DIR,allow_overwriting_input=True)#abre cada archivo
  pdf.save(DIR)#guarda en la misma ruta el archivo desencriptado
```

Apéndice B. Algoritmo creado para la extracción de páginas y creación de archivos.

- Algoritmo para extracción de páginas y creación de archivos:

```
#se importan las librerías necesarias
import os
from PyPDF2 import PdfFileWriter, PdfFileReader
import pikepdf
import fitz
from tqdm import tqdm #pip install tqdm
#definir ruta del archivo al que se le va a aplicar todo el código
workdir = ".//trabajosdegrado"
#leer todos los archivos .pdf en workdir
with os.scandir(workdir) as ficheros:
  ficheros = [fichero.name for fichero in ficheros if fichero.is_file()]
#se definen varias funciones para extraer y dividir las páginas que se
requieren extraer del documento original
def split_pdf1(doc_name, page_num): #los trabajos de grado que tienen 2
hojas de notas y 2 de carta
  pdf_reader = PdfFileReader(open(doc_name, "rb"))
  pdf_writer1 = PdfFileWriter()
  pdf_writer2 = PdfFileWriter()
  pdf_writer3 = PdfFileWriter()
  rutanota = (".//cambios/"+name+"-"+"NOTA DE PROYECTO DE GRADO.pdf")
  rutacarta = (".//cambios/"+name+"-"+"CARTA DE AUTORIZACIÓN.pdf")
  for page in range(page_num,page_num+2):#paginas que se van a extraer y
crea archivo de nota
```

```
    pdf_writer2.addPage(pdf_reader.getPage(page))
  for page in range(page_num+2,page_num+4):#paginas que se van a extraer y
crea archivo de carta
    pdf_writer3.addPage(pdf_reader.getPage(page))
  with open(rutanota, 'wb') as file2:
    pdf_writer2.write(file2)
  with open(rutacarta, 'wb') as file3:
    pdf_writer3.write(file3)
def split_pdf2(doc_name, page_num): #los trabajos de grado que tienen 1 de
nota y 1 de carta
  pdf_reader = PdfFileReader(open(doc_name, "rb"))
  pdf_writer1 = PdfFileWriter()
  pdf_writer2 = PdfFileWriter()
  pdf_writer3 = PdfFileWriter()
  rutanota = (".//cambios/"+name+"-"+"NOTA DE PROYECTO DE GRADO.pdf")
  rutacarta = (".//cambios/"+name+"-"+"CARTA DE AUTORIZACIÓN.pdf")
  for page in range(page_num,page_num+1):#páginas que se van a extraer y
crea archivo de nota
    pdf_writer2.addPage(pdf_reader.getPage(page))
  for page in range(page_num+1,page_num+2):#páginas que se van a extraer y
crea archivo de carta
    pdf_writer3.addPage(pdf_reader.getPage(page))
  with open(rutanota, 'wb') as file2:
    pdf_writer2.write(file2)
  with open(rutacarta, 'wb') as file3:
    pdf_writer3.write(file3)
```

- Algoritmo para detección de texto en archivos PDF:

```
archivosmal=[]
for p in range(len(ficheros)):
  DIR=".//trabajosdegrado/"+ficheros[p]
  x=os.path.basename(DIR)
  c=os.path.splitext(x)
  name=c[0]
  print(name)
  palabra_nota = "NOTA DE PROYECTO"
  palabra_carta = "AUTORIZACIÓN DE SU USO A FAVOR DE LA UIS"
  pdf_document = fitz.open(DIR)
  dato1=[]#se definen los vectores para guardar las páginas encontradas
  dato2=[]
  dato=[]
  dato.clear()
```

```
  for current_page in range(len(pdf_document)):#busca las palabras clave
de nota en todo el documento
    page = pdf_document.load_page(current_page)
    if page.search_for(palabra_nota):
      if current_page < 8:
        dato1.append(current_page)
  for current_page in range(len(pdf_document)):#busca las palabras clave
de carta en todo el documento
    page = pdf_document.load_page(current_page)
    if page.search_for(palabra_carta):
      if current_page < 8:
        dato2.append(current_page)
  for item in dato1:#verificar que no haya datos duplicados en cada vector
    if item not in dato:
      dato.append(item)
  for item in dato2:
    if item not in dato:
      dato.append(item)
dato.sort()
dato.reverse()
```

- Algoritmo para localizar imágenes en un archivo PDF:

```
pags=[]#vectores para comparar datos
pags1=[]
workdir1 = ".//trabajosdegrado"#ruta desde la cual se procesan los
archivos
doc = fitz.Document((os.path.join(DIR))) #encuentra imágenes en todos los
archivos PDF en una carpeta
for i in tqdm(range(8), desc="pages"):
  for img in tqdm(doc.get_page_images(i), desc="page_images"):
    xref = img[0]
    image = doc.extract_image(xref)
    pix = fitz.Pixmap(doc, xref)
    pags.append(i)
    for item in pags:#quitar duplicados de una lista
      if item not in pags1:
        pags1.append(i)
        pags1.reverse()
        paginasfinales=[]
  for item in dato:#vectores para comparar las páginas donde se
encontraron las imágenes
    if item not in paginasfinales:
      paginasfinales.append(item)
  for item in pags1:
```

```
    if item not in paginasfinales:
      paginasfinales.append(item)
  paginasfinales.sort()
```

- Algoritmo para ejecutar las funciones definidas anteriormente para extracción de páginas y creación de archivos:

```
if len(paginasfinales) == 4:#funciona si el vector tiene 4 páginas dentro
  split_pdf1(DIR, paginasfinales[0]) #se define la primera página que va a
extraer
  output_file = (".//cambios/"+name+"-SP.pdf")
  paginasfinales.reverse()
  pages_list=[]
  pages_list.extend(paginasfinales)#se llena el vector pages_list con el
vector paginasfinales
  file_handle = fitz.open(DIR)#se abre el archivo
  file_handle.delete_pages(pages_list)#se borran las páginas del archivo
copia
  file_handle.save(output_file)#se guarda el nuevo documento sin esas
páginas
  dato.clear()#se limpian los vectores
  paginasfinales.clear()
  pags1.clear()
  pages_list.clear()
  pags.clear()
  dato1.clear()
  dato2.clear()
  print(DIR)
elif len(paginasfinales) == 2:#funciona si el vector tiene 2 páginas
dentro
  split_pdf2(DIR, paginasfinales[0]) #se define la primera página que va a
extraer
  output_file = (".//cambios/"+name+"-SP.pdf")
  # Define the pages to keep - 1, 2 and 4
  paginasfinales.reverse()
  pages_list=[]
  pages_list.extend(paginasfinales)#se llena el vector pages_list con el
vector paginasfinales
  file_handle = fitz.open(DIR)#se abre el archivo
  file_handle.delete_pages(pages_list)#se borran las páginas del archivo
copia
  file_handle.save(output_file)#se guarda el nuevo documento sin esas
páginas
```

```
  dato.clear()#se limpian los vectores
  paginasfinales.clear()
  pags1.clear()
  pages_list.clear()
  pags.clear()
  dato1.clear()
  dato2.clear()
  print(DIR)
```

- Algoritmo para creación de carpetas nombradas con el número de inventario:

```
#crear carpetas nombradas con el número de inventario
import os
import pikepdf
workdir = ".//trabajosdegradocompletos/2019/"#ruta en la cual estan los
archivos originales
with os.scandir(workdir) as ficheros:#lee todos los archivos en un
directorio
  ficheros = [fichero.name for fichero in ficheros if fichero.is_file()]
for p in range(len(ficheros)):
  x=os.path.basename(ficheros[p])#extrae el nombre base de un archivo
  c=os.path.splitext(x)#separa el nombre de un archivo
  name=c[0]
  print(name)
  os.mkdir('.//trabajosdegrado/'+name)#crea las carpetas nombradas con
número de inventario
```

- Algoritmo para enviar cada archivo procesado a su respectiva carpeta:

```
#enviar archivos a la carpeta correspondiente de cada número de inventario
import os
import shutil
import pikepdf
año=2019 #se define el año con el que se va a trabajar
workdir = f".//trabajosdegradocompletos/{año}/"#ruta en la cual estan los
archivos originales
with os.scandir(workdir) as ficheros:#lee todos los archivos en un
directorio
  ficheros = [fichero.name for fichero in ficheros if fichero.is_file()]
for p in range(len(ficheros)):
  x=os.path.basename(ficheros[p])#extrae el nombre base de un archivo
  c=os.path.splitext(x)
  name=c[0]
  print(name)
```

```
  if os.path.exists(f".//trabajosdegradocompletos/{año} listo/{name}-CARTA
DE AUTORIZACIÓN.pdf"):#verifica si existe el archivo procesado
    shutil.move(f".//trabajosdegradocompletos/{año} listo/{name}-CARTA DE
AUTORIZACIÓN.pdf",f".//trabajosdegrado/{name}/Carta de autorización.pdf")
  else:
    print("no existe")
  if os.path.exists(f".//trabajosdegradocompletos/{año} listo/{name}-NOTA
DE PROYECTO DE GRADO.pdf"):#verifica si existe el archivo procesado
    shutil.move(f".//trabajosdegradocompletos/{año} listo/{name}-NOTA DE
PROYECTO DE GRADO.pdf",f".//trabajosdegrado/{name}/Nota de proyecto.pdf")
  else:
    print("no existe")
  if os.path.exists(f".//trabajosdegradocompletos/{año} listo/{name}-
SP.pdf"):#verifica si existe el archivo procesado
    shutil.move(f".//trabajosdegradocompletos/{año} listo/{name}-
SP.pdf",f".//trabajosdegrado/{name}/Documento.pdf")
  else:
    print("no existe")
```

Apéndice C. Algoritmo creado para la extracción de metadatos.

- Algoritmo que realiza la búsqueda de las páginas

```
import PyPDF2 #librería que procesa los archivos
import pandas as pd #Módulo para acceder a funcionalidades del sistema
import os

workdir = ".//tesis/2019/desencript" #ruta de los archivos decodificados

with os.scandir(workdir) as ficheros: #escanea el directorio
    np_arreglo = [fichero.name for fichero in ficheros if
fichero.is_file()]
    #agrega cada uno de los nombres de los archivos a una lista

for m in range(len(np_arreglo)):
  name=np_arreglo[m] #se agrega el nombre a la variable
  # abrir el archivo PDF
  with open(f".//tesis/2019/desencript/{name}", "rb") as pdf_file:
      #lee el contenido del archivo PDF
      pdf_reader = PyPDF2.PdfFileReader(pdf_file)
        #rango de páginas donde se buscan los metadatos
      if pdf_reader.numPages < 30:
          rango = pdf_reader.numPages
```

```
    else:
        rango = 30
    # crear una lista para almacenar el texto de cada página
    p_clave = "N/A"
    resumen = "N/A"
    p_clave_en = "N/A"
    abstract = "N/A"
    title = "N/A"
    descrp = -1 #indica si se encontraron las páginas requeridas
    # buscar la palabra en cada página
    for page in range(rango):
        temp_p = -1; temp_p1= -1; temp_r = -1; temp_r1= -1; temp_w = -1;
        temp_w1= -1; temp_a = -1; temp_a1= -1; temp_t = -1; temp_t1= -1;
        #selecciona la página "page" segun el for
        pdf_page_es = pdf_reader.getPage(page)
        if (page + 1) < pdf_reader.numPages:
            #si la siguiente página existe, se selecciona
            pdf_page_en = pdf_reader.getPage(page+1)
        else:
            pdf_page_en = pdf_reader.getPage(page-1)
            #si la siguiente página no existe, se selecciona la anterior
        # extraer el texto de las páginas seleccionadas
        text = pdf_page_es.extractText()
        text_en = pdf_page_en.extractText()
        ar_text_es = []
        ar_text_en = []
        #se reemplazan los saltos de línea
        # los caracteres /// se usan para identificarlos saltos de línea
        #para la separación de metadatos
        text = text.replace(" \n "," /// ").strip()
        text = text.replace("\n \n"," /// ").strip()
        text = text.replace("\n","").strip()
        text_en = text_en.replace(" \n "," /// ").strip()
        text_en = text_en.replace("\n \n"," /// ").strip()
        text_en = text_en.replace("\n","").strip()

        if (descrp == -1 and "PALABRA" in text.upper() and
          ("RESUMEN" in text.upper() or
            "RESÚMEN" in text.upper()) and "AUTOR" in text.upper() and
          ("DESCRIPCI" in text.upper() or "CONTENIDO" in text.upper() or
            "INTRODUCC" in text.upper())):
            print(f"El RESUMEN se encuentra en la página {page + 1}")
            #se vuelve cero al momento de identificar las páginas
            descrp = 0
```

```
                #se crea un arreglo con los datos
                ar_text_es = text.split()
                ar_text_en = text_en.split()
                # Función que busca las palabras que delimitan
                #palabras clave y el resumen en español
                p_clave, resumen = clave_resumen (ar_text_es, temp_p,
temp_p1, p_clave, temp_r, temp_r1, resumen)
                # Función que busca las palabras que delimitan key words
                #y el abstract
                p_clave_en, abstract = key_abstract(ar_text_en, temp_w,
temp_w1, p_clave_en, temp_a, temp_a1, abstract)
                # Función que busca las palabras que delimitan title
                title = titulo_en (ar_text_en, temp_t, temp_t1, title)

  #guardar metadatos en un arreglo
  arreglo1[m].append(name)
  arreglo1[m].append(p_clave)
  arreglo1[m].append(resumen)
  arreglo1[m].append(p_clave_en)
  arreglo1[m].append(abstract)
  arreglo1[m].append(title)

# crear un dataframe de pandas con el arreglo de metadatos
df = pd.DataFrame(arreglo1)
#define el título que llevará cada columna de metadatos
df.columns =
["Num_inventario","Palabras_clave","Resumen","Key_Words","Abstract","Title
"]
# escribir el dataframe en un archivo de Excel
df.to_excel(".//tesis/2019/output_2019.xlsx", index=False)
```

- Función que separa los metadatos (resumen y palabras clave en español)

```
def clave_resumen (ar_text_es, temp_p, temp_p1, p_clave, temp_r, temp_r1,
resumen):
  # Busca las palabras que delimitan palabras clave
  for arr in range(len(ar_text_es)):
      # si encuentra el termino clave "palabras clave"
      if len(ar_text_es) > 5 and temp_p == -1 and
          "PALABRA" in ar_text_es[arr].upper()
          and ("C" in ar_text_es[arr + 1].upper()) :
          #devuelve la posición siguiente al termino clave
          temp_p = arr + 2
```

```
      # si encuentra el termino clave separado los caracteres ///
      #"palabras /// clave"
      elif len(ar_text_es) > 5 and temp_p == -1 and
          "PALABRA" in ar_text_es[arr].upper()
          and ("C" in ar_text_es[arr + 2].upper()):
          #devuelve la posición siguiente al termino clave
          temp_p = arr + 3
      #encuentra el termino clave "descripción" o "contenido" o
      #"introducción" que indica finalización de palabras clave
      elif ((arr+2)<len(ar_text_es) and temp_p != -1 and temp_p1 == -1 and
            ("DESCRIPCI" in ar_text_es[arr].upper() or
            "CONTENIDO" in ar_text_es[arr].upper() or
            "INTRODUC" in ar_text_es[arr].upper())):

          temp_p1 = arr #devuelve la posición exacta al termino clave

  if temp_p !=-1 and temp_p1 !=-1:
      p_clave = ""
      # extraer y guardar las palabras clave en una variable
      for i in range(temp_p1  - (temp_p +1)):
          p_clave = p_clave + " " + ar_text_es[temp_p+i]


  # Busca las palabras que delimitan el resumen
  for arr in range(len(ar_text_es)):

      if temp_p1 !=-1 and temp_r == -1:
          temp_r = temp_p1 + 1
      # si encuentra el termino clave "Trabajo de grado, tesis o
monografía"
      if (temp_r1 == -1 and len(ar_text_es) > 5 and
      ("TRABAJO DE GRADO" in ar_text_es[- arr].upper() or
      ("TRABAJO" in ar_text_es[- arr].upper() and
      ("DE" in ar_text_es[(- arr) + 1].upper() and
      (("GRA" in ar_text_es[(- arr) + 2].upper() or
          "MA" in ar_text_es[(- arr) + 2].upper() or
          "IN" in ar_text_es[(- arr) + 2].upper() or
          "AP" in ar_text_es[(- arr) + 2].upper()) or
          "GRA" in ar_text_es[(- arr) + 3].upper())))) or
      ("PROYECTO" in ar_text_es[- arr].upper() and
        ("DE" in ar_text_es[(- arr) + 1].upper() and
        (("GRA" in ar_text_es[(- arr) + 2].upper() or
          "MA" in ar_text_es[(- arr) + 2].upper() or
          "IN" in ar_text_es[(- arr) + 2].upper() or
```

```
            "AP" in ar_text_es[(- arr) + 2].upper()) or
            "GRA" in ar_text_es[(- arr) + 3].upper()))) or
        ("TESIS" in ar_text_es[- arr].upper() and
          ("DE" in ar_text_es[(- arr) + 1].upper())) or
        ("MONOGRA" in ar_text_es[- arr].upper()))):
            #devuelve la posición exacta al termino clave
            temp_r1 = arr
        if "///" in ar_text_es[arr]: # elimina los caracteres ///
            ar_text_es[arr] = ""

    if temp_r !=-1 and temp_r1 !=-1:
        resumen = ""
        # extraer y guardar el resumen en una variable
        for i in range(temp_r1 - (temp_r+1)):
            resumen = resumen + " " + ar_text_es[temp_r+i]

    return p_clave, resumen
```

- Función que separa los metadatos (resumen y palabras clave en inglés)

```
def key_abstract(ar_text_en, temp_w, temp_w1, p_clave_en, temp_a, temp_a1,
abstract):
    # Busca las palabras que delimitan KEY WORDS
    for arr in range(len(ar_text_en)):
        # si encuentra el termino clave "keywords" en un solo termino
        if len(ar_text_en) > 5 and temp_w == -1 and
          ("KEYWOR" in ar_text_en[arr].upper() or
            "KEY-WOR" in ar_text_en[arr].upper() or
            "WORD" in ar_text_en[arr].upper()):
            #devuelve la posición siguiente al termino clave
            temp_w = arr + 1
        # si encuentra el termino clave "key words"
        elif (len(ar_text_en) > 5 and temp_w == -1 and
            ("KEY" in ar_text_en[arr].upper() and
              ("WOR" in ar_text_en[arr + 1].upper()) or
              "PALABRA" in ar_text_en[arr].upper() and
                ("CLAVE" in ar_text_en[arr + 1].upper()))):
            #devuelve la posición siguiente al termino clave
            temp_w = arr + 2
        #encuentra el termino clave separado los caracteres ///
        #"Key /// words"
        elif (len(ar_text_en) > 5 and temp_w == -1 and
```

```
          ("KEY" in ar_text_en[arr].upper() and
            ("WOR" in ar_text_en[arr + 2].upper()) or
            "PALABRA" in ar_text_en[arr].upper() and
            ("CLAVE" in ar_text_en[arr + 2].upper()))):
          #devuelve la posición siguiente al termino clave
          temp_w = arr + 3
      #encuentra el termino clave "description", "content",
      # "introduction" que indica finalización de key words
      elif ((arr+2)<len(ar_text_en) and temp_w1 == -1 and temp_w !=-1 and
            ("DESCRIP" in ar_text_en[arr].upper() or
              "CONTENT" in ar_text_en[arr].upper() or
              "CONTENI" in ar_text_en[arr].upper() or
              "SUMM" in ar_text_en[arr].upper() or
              "ABST" in ar_text_en[arr].upper()or
              "INTRODUC" in ar_text_en[arr].upper())):
          temp_w1 = arr #devuelve la posición exacta al termino clave

      if "///" in ar_text_en[arr]:
          ar_text_en[arr] = ""

  if temp_w !=-1 and temp_w1 !=-1:
      p_clave_en = ""
      # extraer y guardar el resumen en una variable
      for i in range(temp_w1  - (temp_w +1)):
          p_clave_en = p_clave_en + " " + ar_text_en[temp_w+i]
  elif p_clave_en == "N/A":
      p_clave_en = "N/A"
  if len(p_clave_en.split())<3 or len(p_clave_en.split()) > 40:
      p_clave_en = "N/A"

  # Busca las palabras que delimitan el ABSTRACT
  for arr in range(len(ar_text_en)):
      if temp_w1 !=-1 and temp_a == -1:
          temp_a = temp_w1 + 1

      # si encuentra el termino clave "Degree work, thesis, entre otros"
      if (temp_a1 == -1 and len(ar_text_en) > 5  and
      ("DEGREE WORK" in ar_text_en[- arr].upper() or
      ("DEGRE" in ar_text_en[- arr].upper() and
        ("WORK" in ar_text_en[(- arr) + 1].upper() or
        "PAPER" in ar_text_en[(- arr) + 1].upper() or
          "PROJ" in ar_text_en[(- arr) + 1].upper() or
          "WORK" in ar_text_en[(- arr) + 2].upper())) or
      ("BACHELOR" in ar_text_en[- arr].upper() and
```

```
        ("THES" in ar_text_en[(- arr) + 1].upper() or
        "DEGREE" in ar_text_en[(- arr)+1].upper() or
        "TES" in ar_text_en[(- arr)+1].upper())) or
      ("GRADUAT" in ar_text_en[- arr].upper() and
        ("PROJ" in ar_text_en[(- arr) + 1].upper() or
        "WORK" in ar_text_en[(- arr) + 1].upper() or
        "PAPER" in ar_text_en[(- arr) + 1].upper())) or
      ("GRADE" in ar_text_en[- arr].upper() and
        ("WORK" in ar_text_en[(- arr) + 1].upper() or
        "PROJ" in ar_text_en[(- arr) + 1].upper())) or
      ("PROJECT" in ar_text_en[- arr].upper() and
        ("OF" in ar_text_en[(- arr) + 1].upper() or
        "GRADE" in ar_text_en[(- arr) + 1].upper())) or
      ("WORK" in ar_text_en[- arr].upper() and
        ("DEGREE" in ar_text_en[(- arr) + 1].upper() or
        "OF" in ar_text_en[(- arr) + 1].upper())) or
      ("UNDER" in ar_text_en[- arr].upper() and
        ("THES" in ar_text_en[(- arr) + 1].upper() or
        "WORK" in ar_text_en[(- arr) + 1].upper() or
        "PROJ" in ar_text_en[(- arr) + 1].upper())) or
      ("TRABAJO" in ar_text_en[- arr].upper() and
        ("DE" in ar_text_en[(- arr) + 1].upper() and
        ("GRADO" in ar_text_en[(- arr) + 2].upper() or
        "MA" in ar_text_en[(- arr) + 2].upper() or
        "IN" in ar_text_en[(- arr) + 2].upper()))) or
      ("PROYECTO" in ar_text_en[- arr].upper() and
        ("DE" in ar_text_en[(- arr) + 1].upper() and
        ("GRADO" in ar_text_en[(- arr) + 2].upper() or
        "MA" in ar_text_en[(- arr) + 2].upper() or
        "IN" in ar_text_en[(- arr) + 2].upper()))) or
      ("THESIS" == ar_text_en[- arr].upper()) or
      ("THESIS" in ar_text_en[- arr].upper() and
        ("WOR" in ar_text_en[(- arr) + 1].upper())) or
      ("MONOGRA" in ar_text_en[- arr].upper()) or
      ("RESEACH" in ar_text_en[- arr].upper() and
        ("PROJ" in ar_text_en[(- arr) + 1].upper() or
        "WORK" in ar_text_en[(- arr) + 1].upper()))
      )):

            temp_a1 = arr # devuelve la posición exacta al termino clave

    if "///" in ar_text_en[arr]: # elimina los caracteres ///
        ar_text_en[arr] = ""
```

```
  if temp_a !=-1 and temp_a1 !=-1:
      abstract = ""
      # extraer y guardar el abstract en una variable
      for i in range(temp_a1 - (temp_a+1)):
          abstract = abstract + " " + ar_text_en[temp_a+i]

  return p_clave_en, abstract
```

- Función que separa el metadato título en inglés

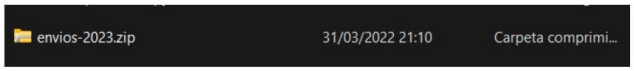
```
def titulo_en (ar_text_en, temp_t, temp_t1, title):
    # Busca las palabras que delimitan el TITLE
    for arr in range(len(ar_text_en)):
        #Encuenta el termino clave inicial
        if(temp_t == -1  and len(ar_text_en) > 5 and
          ("TITLE" in ar_text_en[arr].upper() or
           "TÍTLE" in ar_text_en[arr].upper() or
           "TÍTTLE" in ar_text_en[arr].upper() or
           "TITTLE" in ar_text_en[arr].upper() or
           "TÍTUL" in ar_text_en[arr].upper() or
           "TITUL" in ar_text_en[arr].upper())):
            temp_t = arr # devuelve la posición exacta del termino
        #Encuenta el termino clave final
        if(temp_t1 == -1  and len(ar_text_en) > 5 and
          ("AUTHO" in ar_text_en[arr].upper() or
          "AUTOR" in ar_text_en[arr].upper())):
            temp_t1 = arr - 1

        if "///" in ar_text_en[arr]: # elimina los caracteres ///
            ar_text_en[arr] = ""

    if temp_t !=-1 and temp_t1 !=-1:
        title = ""
        # extraer y guardar el TITLE en una variable
        for i in range(temp_t1 - (temp_t)):
            title = title + " " + ar_text_en[temp_t+i]

    return title
```

Apéndice D. Migración de documentos al Repositorio Noesis

1. Recibir paquete de registros

a. Descomprimir archivo

Se recibe el documento en formato comprimido y se procede a descomprimirlos



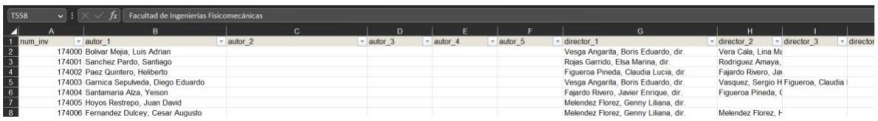
b. Verificar registros

Se verifica que la carpeta descomprimida contenga las carpetas de los registros a ser migrados y un archivo Excel que con la información de cada registro:

Cada carpeta contiene la Carta de autorización, el Documento del registro, y la Nota de proyecto

c. Verificar documento Excel

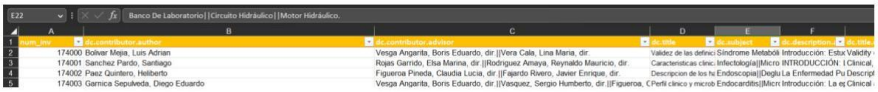
El archivo Excel indica el número al que se relaciona la información con la carpeta de documentos y cada columna contiene la información que debe ir en los metadatos al ser ingresada al repositorio Dspace:



2. Modificar Archivo Excel

a. Adaptar cabeceras a Dublin Core

Se adaptan las cabeceras de las columnas a sus respectivos campos dentro del Dublin Core.



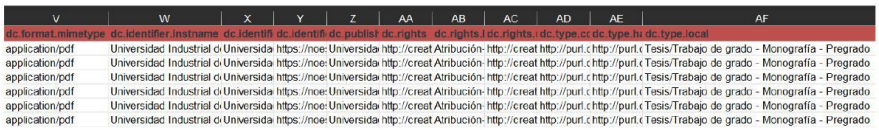
b. Concatenar información de columnas

Las columnas con la misma información (Como autores, asesores, palabras clave), se agrupan en una sola columna con el carácter "||" que separa cada dato.

C
dc.contributor.advisor
Vesga Angarita, Boris Eduardo, dir.||Vera Cala, Lina Maria, dir.
Rojas Garrido, Elsa Marina, dir.||Rodriguez Amaya, Reynaldo Mauricio, dir.
Figueroa Pineda, Claudia Lucia, dir.||Fajardo Rivero, Javier Enrique, dir.
Vesga Angarita, Boris Eduardo, dir.||Vasquez, Sergio Humberto, dir.||Figueroa, C
Fajardo Rivero, Javier Enrique, dir.||Figueroa Pineda, Claudia Lucia, dir.

c. Añadir columnas con metadatos faltantes

Se agregan nuevas columnas con los datos restantes, como los datos de derechos de autor y descriptores del registro:



V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
dc.format.mimetype	dc.identifier.instname	dc.identifi	dc.identifi	dc.publish	dc.rights	dc.rights.l	dc.rights.i	dc.type.co	dc.type.ha	dc.type.local
application/pdf	Universidad Industrial d	Universida	https://noe	Universida	http://creat	Atribución-	http://creat	http://purl.(	http://purl.(	Tesis/Trabajo de grado - Monografía - Pregrado
application/pdf	Universidad Industrial d	Universida	https://noe	Universida	http://creat	Atribución-	http://creat	http://purl.(	http://purl.(	Tesis/Trabajo de grado - Monografía - Pregrado
application/pdf	Universidad Industrial d	Universida	https://noe	Universida	http://creat	Atribución-	http://creat	http://purl.(	http://purl.(	Tesis/Trabajo de grado - Monografía - Pregrado
application/pdf	Universidad Industrial d	Universida	https://noe	Universida	http://creat	Atribución-	http://creat	http://purl.(	http://purl.(	Tesis/Trabajo de grado - Monografía - Pregrado
application/pdf	Universidad Industrial d	Universida	https://noe	Universida	http://creat	Atribución-	http://creat	http://purl.(	http://purl.(	Tesis/Trabajo de grado - Monografía - Pregrado
application/pdf	Universidad Industrial d	Universida	https://noe	Universida	http://creat	Atribución-	http://creat	http://purl.(	http://purl.(	Tesis/Trabajo de grado - Monografía - Pregrado

3. Preparar contenido para la subida

La importación por lotes de Dspace requiere utilizar el formato simple de archivos, más información en el siguiente enlace:

https://wiki.lyrasis.org/display/DSDOC7x/Importing+and+Exporting+Items+via+Simple+Archive+Format

Se debe preparar una carpeta con la siguiente estructura por cada programa que deba importar el repositorio:

```
Coleccion/
└── Registro_001/
       ├── dublin_core.xml: metadados adaptados a Dublin Core
       ├── contents: listado de archivos.pdf
       ├── Carta de autorizacion.pdf
       ├── Documento.pdf
       └── Nota de proyecto.pdf
```

a. Generar archivo “dublin_core.xml” para cada registro

Cada fila del documento Excel representa un registro y cada cabecera el metadato al que corresponde el valor de la columna. Por lo que se genera un archivo “dublin_core.xml” por cada fila y este relaciona con la carpeta que contiene el número de la primera columna. Este archivo debe acompañar a los documentos del registro. El contenido del archivo “dublin_core.xml” sigue la siguiente estructura:

```
<dublin_core>
    <dcvalue element="title" qualifier="none">A Tale of Two Cities</dcvalue>
    <dcvalue element="date" qualifier="issued">1990</dcvalue>
    …
</dublin_core>
```

b. Crear directorio “Collections”

Para mantener el orden se crea una carpeta “Collections” que contendrá las carpetas por cada programa que será importado por el servidor.

c. Organizar los registros por programa y guardarlo en “Collections”

Basándose en la información de programa dentro del archivo Excel, se procede a ordenar los registros por programa y cada programa será representado por una carpeta dentro de “Collections”.

d. Generar documento “content” por registro

El archivo “content” no tiene ninguna extensión y su contenido es el siguiente:

Carta de autorización.pdf

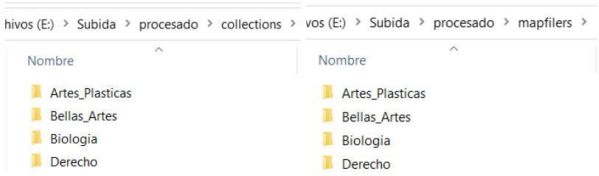
Documento.pdf

Nota de proyecto.pdf

Cabe resaltar que el contenido tiene la misma forma porque todos los documentos tienen la misma estructura y nombre, pero si difieren cada archivo “content” debe tener los nombres de los archivos del registro.

e. Preparar carpetas “Mapfiles”

Las carpetas “Mapfiles” sirven para que el repositorio cree archivos de seguimiento de la importación de registros. Al ordenar los programas a los que pertenecen los registros, se tiene el conocimiento de todos los programas que serán importados, por lo que procede a crear una carpeta vacía por cada carpeta de programa.



4. Importar registros en Repositorio

Después de realizar el proceso anterior, las estructuras de la carpeta “Collections” y “Mapfilles” serían la siguientes:

```
Collections/
├── Programa_001/
│       ├── Registro_001/
│       │       ├── dublin_core.xml: metadados adaptados a Dublin Core
│       │       ├── contents: listado de archivos.pdf
│       │       ├── Carta de autorizacion.pdf
│       │       ├── Documento.pdf
│       │       └── Nota de proyecto.pdf
│       ├── Registro_002/
│       ├── Registro_003/
│       …
├── Programa_002/
├── Programa_003/
…

Mapfiles/
├── Programa_001/
├── Programa_002/
├── Programa_003/
…
```

a. Subir carpetas “Collections” y “Mapfiles” al servidor

Se suben las carpetas al servidor del repositorio, usualmente se escoge la ruta “/home/downloads”, pero se puede escoger otra.

b. Utilizar el comando de importación del repositorio

Los programas se importan siguiendo el siguiente comando:

```
cd /home/dspace/dspace: Primero situarse en la carpeta del dspace

bin/dspace import -a -e [usuario] -c [identificador coleccion] -s [ubicación carpeta programa] -m [ubicación carperta mapfile] -t
```

bin/dspace import		Comando de importación
-a	--add	Añadir registros al repositorio
-e	--eperson	Usuario
-c	--collection	Handle o identificador de programa o colección
-s	--source	Directorio del programa o colección
-m	--mapfile	Directorio del mapfile del programa o colección
-t	--test	Comprueba que los registros están correctos para ser subidos

Ejemplo:

- Probar registros de programa:

```
bin/dspace import -a -e webmaster@noesis.uis.edu.co -c e1506c17-13fb-4ec4-83c4-b4f4b2dd0326 -s /home/downloads/collections/Especializacion_en_Oftalmologia -m /home/downloads/mapfiles_collections/Especializacion_en_Oftalmologia/mapfile -t
```

- Importar registros de programa:

```
bin/dspace import -a -e webmaster@noesis.uis.edu.co -c e1506c17-13fb-4ec4-83c4-b4f4b2dd0326 -s /home/downloads/collections/Especializacion_en_Oftalmologia -m /home/downloads/mapfiles_collections/Especializacion_en_Oftalmologia/mapfile
```