Renovación del Sistema Integrado de Información Universitaria para la Creación y Generación de Conocimiento

Daniel Alejandro Higuita Usuga

Informe de práctica para optar al título de Ingeniero de Sistemas

Directora
Astrid Duque Ramos, Doctora en Ingeniería Informática

Universidad de Antioquia
Facultad de Ingeniería
Departamento Ingeniería de Sistemas
Medellín, Antioquia, Colombia
2022

Dedicatoria

A mis padres Wildeman y Gloria su eterna paciencia, amor, esfuerzo y apoyo me permitieron llegar hasta aquí, gracias por enseñarme a través del ejemplo sobre perseverancia, y de no huir a las dificultades sino enfrentarlas.

Agradecimientos

A la Vicerrectoría de Investigación, por darme la oportunidad de realizar mi práctica académica, a mis asesora Astrid Duque Ramos, y a Omar Zapata, su guía y enseñanzas a lo largo de este proceso me ayudaron a crecer como profesional.

Finalmente al grupo CoLaV, sin el trabajo hecho por ellos, no hubiera sido posible realizar este trabajo.

Tabla de contenido

LISTA DE TABLAS

LISTA DE FIGURAS

SIGLAS, ACRÓNIMOS Y ABREVIATURAS

SIIU	Sistema Integrado de Información Universitaria
ETL	Extract Transform Load (Extracción Transformación y Carga)
UdeA	Universidad de Antioquia
DB	Database (Base de datos)
GB	Gigabyte

RESUMEN

El Sistema Integrado de Información Universitaria (SIIU) de la Universidad de Antioquia es una plataforma creada por la Vicerrectoría de Investigación para gestionar proyectos de investigación a través de cinco módulos: convocatorias, administrativo, reportes, notificaciones y presupuestal. La Vicerrectoría de Investigación requiere mejorar la gestión de la información que se encuentra en cada módulo, para extraer y analizar datos que permitan crear indicadores y reportes en el marco del proyecto ImpactU. Este proyecto tiene como objetivo principal crear un sistema que mejore la gestión de la información de la base de datos del SIIU, con el cuál se hace un cambio en el modelo de la información y se construye una estructura lógica con los datos a través de un Extract Transform Load (ETL). Además se identifican problemas en la base de datos y se proponen soluciones para evitar que se sigan ocurriendo y de esa manera mejorar la calidad de la información para la creación de indicadores, finalmente se crea un software con el cuál se automatiza el proceso de ETL.

***Palabras clave* — Base de datos, SIIU, ETL.**

ABSTRACT

The Integrated System of University Information (SIIU) of the University of Antioquia is a platform made by the Vice Rectory for Research which is used to manage research projects with the help of five modules: application period, administrative, reports, notifications and budget. The Vice Rectory for Research needs to improve the management of the information found in each module, in order to extract and analyze data that allow the creation of indicators and reports within the framework of the ImpactU project. The main objective of this project is to create a system which improves the information management of the SIIU database, for which a change is made in the information model and a logical structure is built with the data through a Extract Transform Load (ETL). In addition, there was a process of identifying problems in the database and proposing solutions in order to prevent such problems appearing again and that way the quality of the information for the creation of indicators and reports is improved, finally a software was created with which the ETL process was automated.

Keywords **— Databases, SIIU, ETL, Research and development management, Research Information Systems.**

I. INTRODUCCIÓN

La Vicerrectoría de Investigación es la dependencia que lidera el Sistema de Investigación de la Universidad de Antioquia y la responsable de la ejecución de las estrategias de fomento de la investigación definidas por el Comité para el Desarrollo de la Investigación, CODI, por medio de la gestión de convocatorias, fondos para la investigación, proyectos especiales y divulgación científica[1]. De allí, que para cumplir con esas responsabilidades construyeron el Sistema Integrado de Información Universitaria (SIIU), es un sistema de información creado para mejorar la eficiencia en la administración de los proyectos de investigación de la Universidad de Antioquia[2]. El SIIU es una plataforma que contiene cinco módulos: convocatorias, administrativo, reportes, notificaciones y presupuestal, para hacer seguimiento y gestión administrativa y presupuestal de los proyectos de investigación; incluye creación y apertura de la convocatoria, inscripción de proyectos, revisión técnica y aval de proyectos, evaluación científica de proyectos por pares académicos, aprobación de proyectos por parte de las dependencias académicas y administrativas; creación automática de actas de inicio, administración de participantes, cumplimiento y homologación de compromisos, gestión de tareas, prórrogas y finalización de proyectos; generación de reportes automáticos y certificados y además envió de diferentes tipos de notificaciones y gestión de presupuesto inicial, cofinanciadores y cambios de rubros de proyectos. El SIIU contiene más de 6560 proyectos registrados y la Vicerrectoría de Investigación requiere mejorar la gestión de estos datos para tener un mayor impacto en la calidad de la información y con esto potenciar la toma de decisiones y el renombre de la Universidad en aspectos como: la innovación, el desarrollo tecnológico y la creación.

Para la gestión de estos datos la Vicerrectoría de Investigación recibe el soporte técnico del grupo CoLaV un grupo colaborativo para el desarrollo de software que implementa procesos de automatización, analítica de información y promueve la implementación de iniciativas de ciencia abierta especialmente en los aspectos de datos abiertos, acceso abierto, políticas para ciencia abierta y estrategias relacionadas[3]; dentro del marco proyecto IMPACTU

Dentro de este proyecto, se enmarca esta práctica académica que consiste en crear un ETL (Extract, Transform and Load) de la base de datos del SIIU para estructurar la información de manera lógica y de acuerdo a las necesidades expresadas por la Vicerrectoría de Investigación.

Para el desarrollo se creó una imagen de Oracle DB en Docker, con la cuál se recreó la base de datos del SIIU y se analizaron los modelos relaciones para crear la estructura lógica de la información, se crearon paquetes de software con los que se automatizó el proceso de transformación de la base de datos, además se adaptaron y usaron paquetes de software creados por el grupo Colav[4] para la extracción de la información. Por último se hizo el proceso completo del ETL, donde la nueva estructura lógica de los datos fue guardada con persistencia en JSON y se generaron propuestas y estrategias para mejorar la calidad de los datos.

II. OBJETIVOS

A. Objetivo general

Crear un sistema para mejorar la gestión de la información de la base de datos del SIIU.

B. Objetivos específicos

- Crear el ETL (Extract, Transform and Load) de la Base de Datos del Sistema de Información de Investigación Universitario - SIIU.
- Generar un documento con una propuesta de estrategias que permitan solucionar los problemas de la base de datos.
- Crear un paquete de software para la automatización del proceso de transformación de los datos.

III. MARCO TEÓRICO

En el proyecto se usaron herramientas, conceptos de desarrollo de software y metodologías, todo esto con el fin de cumplir los objetivos propuestos, a continuación se definen los que fueron empleados.

La metodología secuencial divide el desarrollo de software en varios pasos o fases, donde el fin de cada fase delimita el comienzo de la otra, igualmente propone una gran fase de diseño y otra de construcción donde el alcance de cada una es la totalidad de los requerimientos de un proyecto[5]. En la fase diseño, los proyectos de software requieren un lenguaje universal para tener una comunicación efectiva entre los actores que participan en el desarrollo, esta comunicación se hace a través del Lenguaje unificado de modelado (UML), lenguaje de modelado estandarizado que consiste en un conjunto de diagramas con los cuales se pueden especificar, visualizar, construir y documentar componentes de software[6]. En la fase de construcción es necesario seguir un estándar para desarrollar y documentar código de calidad, una guía que indica las convenciones estilísticas a seguir para escribir código Python es Python Enterprise Proposal (PEP8), un conjunto de recomendaciones cuyo objetivo es ayudar a escribir código más legible y abarca desde cómo nombrar variables hasta el número máximo de caracteres que una línea debe tener[7].

El SIIU almacena sus datos en una base de datos relacional, la cual representa la información en forma de tablas, donde cada registro tiene una ID única, llamada clave[8]. Por el contrario, una base de datos NoSQL puede adaptarse a una amplia variedad de modelos de datos, incluidos los formatos de valor clave, documentos, columnas y gráficos[9].

Las herramientas que se utilizaron para la parte técnica del desarrollo fueron:

- Github: es un sistema de control de versiones distribuido. Esto significa que un clon local del proyecto es un repositorio de control de versiones completo[10].
- Docker: es una plataforma de software de código abierto para crear, implementar y administrar contenedores de aplicaciones virtualizados en un sistema operativo (SO) común, con un ecosistema de herramientas aliadas[11].

- Oracle: es un sistema de administración de bases de datos relacionales (RDBMS) de Oracle Corporation[12].
- Python: es un lenguaje del tipo interpretado, multiparadigma: Soporta orientación a objetos, programación imperativa y funcional, además es de tipado dinámico, multiplataforma y multipropósito[13].
- Pandas: es una librería de Python especializada en el manejo y análisis de estructuras de datos[14].
- Pygraphviz: es una interfaz hecha en Python donde se pueden crear, editar, leer y dibujar grafos usando Python [15].
- Ukupacha: es un software creado por el grupo CoLaV para crear JSON a partir de un esquema definido en un diccionario[16].
- Kaypacha: es un paquete que se usa para extraer datos de una base de datos SQL[17].

IV. METODOLOGÍA

Se definieron tareas secuenciales las cuales se siguieron para crear el ETL y la documentación de propuestas para mejorar la calidad de los datos:

El proceso para crear el ETL de la base de datos del SIIU, consta de 3 pasos

1. *Extracción de la información*

   Este paso consiste en la creación de una imagen de Oracle en Docker que fue adaptada a la base de datos del SIIU. Para ello se definieron las variables de entorno y se elaboraron bash scripts con el fin de crear la imagen de Oracle e importar el dump entregado por el SIIU. Luego se hizo el despliegue de la base datos en un contenedor Docker y se agendaron reuniones semanales con la Vicerrectoría de Investigación donde definió que la tabla proyecto sería usada para crear una entidad, además se discutió los datos y tablas que podrían ser útiles para la creación de indicadores, después de cada reunión se hizo analizaron las tablas sugeridas, se documentaron los problemas encontrados en los registros y se creaba una nueva versión del diccionario.

2. *Transformación*

   Este proceso se fundamenta en la creación de un archivo JSON con algunos registros de la tabla principal Proyecto, utilizando Ukupacha el cuál recorría la tabla principal y extraía la cantidad de registros indicado junto los datos de las tablas relacionadas, durante la extracción ocurrió un error durante la extracción de la información debido a una anomalía en una fecha, se reportó el error durante una reunión y fue corregido por el grupo CoLaV.

3. *Carga*

   En este paso se creó un algoritmo con el cuál se configuró la persistencia por línea de comandos, y se cargaron todos los registros de la tabla Proyecto junto con los datos de las tablas relacionadas que se definieron en el diccionario. Ukupacha

recorre la tabla principal y extrae sus registros junto con los de las tablas relacionadas.

La definición y documentación de estrategias que permitan solucionar los problemas de la base de datos, se hizo con los siguientes pasos:

En las reuniones con la Vicerrectoría de Investigación se identificaron las tablas que podían contener información útil para la creación de indicadores y métricas, luego se verificó el estado de esos datos, se documentaron aquellos que tenían características sospechosas como: valores anómalos y campos nulos, estos hallazgos, se reportaban en las reunión y al mismo tiempo se buscan soluciones para prevenir que se volvieran a presentar.

V. RESULTADOS

La arquitectura necesaria para crear el ETL se encuentra en el diagrama (**Fig. 1**), el sistema está compuesto por una base de datos Oracle, la cuál se desplegó en un contenedor Docker; un paquete de software Kaypacha, que a su vez está compuesto por otros dos paquetes, uno que representa el grafo de la entidad proyecto y otro con el cúal se automatiza la creación del ETL; por último, se tiene un módulo Ukupacha, proveído por el grupo CoLaV, que tiene los métodos para recorrer el grafo e ir creando el ETL.

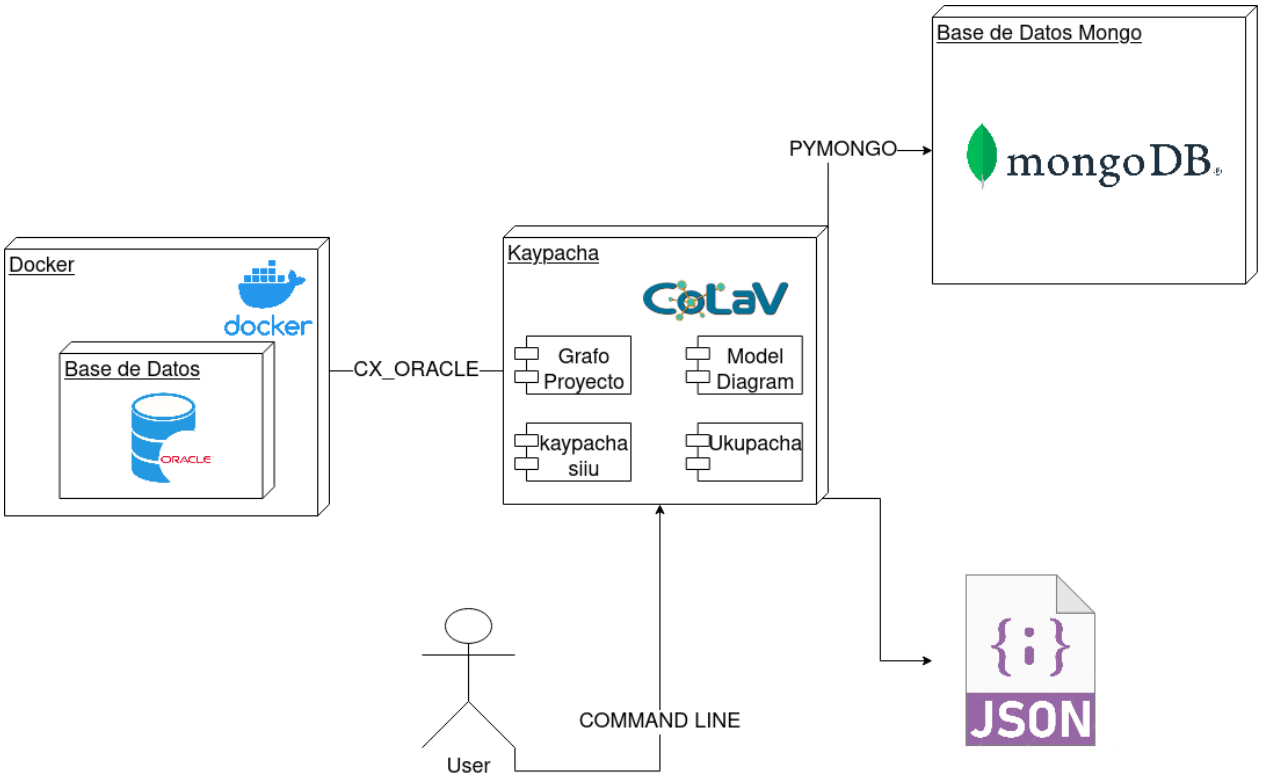

Fig. 1. Arquitectura del sistema para crear el ETL de la Base de Datos del SIIU.

A continuación se detallan cada uno de los componentes de la arquitectura, primero se creó la imagen de Oracle, la cual consta de 4 pasos:

1. Definición de variables de ambiente: con los usuarios, contraseña y tablespaces de la base de datos del SIIU y el lugar donde está almacenado el dump (**Figura 2**).

```
## Setting up the db users.
SIIU_USERS="BUPP,MARES,MARES1,SISINVEST,SOPORTEBUPP,CONSULTABUPP,CONSULTASIIU,WSGENERAL"
SIIU_USERS="$SIIU_USERS,WSSIIU,CONSULTASIPE,SAP_COMUN,CONSULTAOBSERVATORIO,REPORTESSIIUAPP"
SIIU_USERS="$SIIU_USERS,SUBAPLICACIONESSIIUAPP,SIIUADMAPP,SIIUCONFIGAPP,CONSULTAMARES"
SIIU_USERS="$SIIU_USERS,CONSULTAMOISES,BPP_VICE_INVES,BPP_ADMON,BPP_CONSULTA,BPP_DEPEN"
SIIU_USERS="$SIIU_USERS,BPP_ORDENADOR,BPP_PLANEACION,BPP_RESPON_PROY,ANALISPLAN3,COMPRAS"
SIIU_USERS="$SIIU_USERS,OPS\$COMPRAS,OPS\$TESO,BUPP1,PPTO,BUPP_ADMON,BUPP_INTEGRACIONSAP"
SIIU_USERS="$SIIU_USERS,SIVI,SIIU_CONTROL_INTERNO,ANAVICEDOC,COMPUTO,BUPP_DEPEN"
SIIU_USERS="$SIIU_USERS,GESTIONINFORELINTER,SIPE_ADMIN,SIPE_CATED,SIPE"
SIIU_USERS="$SIIU_USERS,SECCONCATE,MUA,AUXPPTO1,ANALISPPTO2"
export SIIU_USERS
export SIIU_TABLESPACES="BUPP,IDX_BUPP"
# Change this for the desire password
export ORACLE_PWD="colavudea"

# This is the path where you have the .dmp file, change it for the path where is located your dump file.
# by default if looking in your home in the folder dump
export DUMP_PATH="$HOME/dump"

# Dump/Log  files ex:
export DUMP_FILES="expdp_bupp_12_mayo2022.dmp"
```

Fig. 2. Variables de Entorno con Metadatos acerca de la Base de Datos del SIIU.

2. Bash Script Load Dump: se crearon algoritmos, implementados en bash script, para automatizar la creación de los usuarios, tablespaces e importar el dump (**Figura 3**), una de las tablas (SIIU_ARCHIVO_ADJUNTO) tuvo que ser excluida ya que contenía imágenes, pdfs, etc. cuyo tamaño hacía que toda la base de datos excediera los 12 GB permitidos en la versión gratis de Oracle.

```
IFS=', ' read -r -a array <<< "$SIIU_TABLESPACES"
for tablespace in "${array[@]}"
do
    echo "Creating TableSpace: $tablespace"
    createTablespace $tablespace
    echo "The tablespace $tablespace was created with $? errors"
done

IFS=', ' read -r -a array <<< "$SIIU_USERS"
for user in "${array[@]}"
do
    echo "Creating User: $user"
    createUser $user $ORACLE_PWD "BUPP"
    echo "The user $user was created with $? errors"
done

IFS=', ' read -r -a array <<< "$DUMP_FILES"
for dump_file in "${array[@]}"
do
    echo "Loading dump file $dump_file"
    impdp system/$ORACLE_PWD@localhost:1521 exclude=table:\"IN \'SIIU_ARCHIVO_ADJUNTO\'\"  \
    directory=siiu_pump_dir dumpfile=$dump_file data_options=skip_constraint_errors version=11.2.0.4.0
    echo "The dump file $dump_file was created"
done
```

Fig. 3. Bash Script para Automatizar la Creación de Usuarios.

3. Bash Script Crear Imagen:se modificó el bash script para crear la imagen de la base de datos , donde se incluyó el repositorio Docker Hub del grupo CoLaV (**Figura 4**).

```
# Which Dockerfile should be used?
DOCKERFILE="${DOCKERFILE}.${EDITION}"

# Oracle Database image Name
# If provided using -t build option then use it; Otherwise, create with version and edition
if [ -z "${IMAGE_NAME}" ]; then
  #IMAGE_NAME="colav/oracle-docker:${VERSION}-${EDITION}"
  IMAGE_NAME="colav/siiu-oracle-docker:latest"
fi;

# Go into version folder
cd "${VERSION}" || {
  echo "Could not find version directory '${VERSION}'";
  exit 1;
}

if [ ! "${SKIPMD5}" -eq 1 ]; then
  checksumPackages
else
  echo "Ignored MD5 checksum."
fi
echo "========================="
echo "Container runtime info:"
"${CONTAINER_RUNTIME}" info
echo "========================="
```

Fig. 4. Bash Script para Crear la Imagen de Oracle.

4. Archivo YML: se adaptó el archivo yml con el que se despliega el contendor de docker (**Figura 5**).

```
version: '2'
services:
  oracle-docker:
    image: colav/siiu-oracle-docker:latest
    volumes:
      - ${DUMP_PATH}:/home/oracle/dump # dump path (where the file .dmp resides).
    ports:
      - 1521:1521
      #- 8080:8080  # required by oracle apex (commented because we dont need apex)
    env_file:
      - config.env   # WARNING!!!!  CREATE THIS FILE BEFORE LAUNCH!
    environment:
      - ORACLE_PWD=${ORACLE_PWD}
    mem_reservation: 8000M
    cpuset: "0-1"
```

Fig. 5. Archivo yml para Crear el Contenedor de la Base de Datos del SIIU.

El segundo componente de la arquitectura es el paquete de software Kaypacha que contiene

1. El grafo proyecto: la creación de éste sigue un formato de diccionario de Python específico para ser procesado por Ukupacha (**Fig. 6**). Los campos definidos en el diccionario son:

- MAIN_TABLE: en este campo se especifica la tabla con la que Ukupacha comienza a recorrer el grafo.
- CHECKPOINT: es un campo que se usa para llevar un historial de registros que ya fueron procesados.
- SCHEMA_VERSION: aquí se determina la versión actual del diccionario.
- GRAPH: aquí se construye el grafo, los atributos de la tabla principal se relacionan con los de la tabla de interés, ya que es una estructura recursiva, se pueden crear niveles de profundidad que consisten en seguir relacionando tablas.

```
graph_project = {
    "MAIN_TABLE": "SIIU_PROYECTO",
    "CHECKPOINT": {"DB": "BUPP", "KEYS": ["CODIGO"]},
    "SCHEMA_VERSION": 0.1,
    "GRAPH":[
        {
            "SIIU_PROYECTO":[

                {
                    "KEYS": ["CODIGO/PROYECTO"],
                    "DB":"BUPP",
                    "TABLES":[
                        {
                            "SIIU_COMPROMISO_POR_PROYECTO": [
                                {
                                    "KEYS": ["COMPROMISO/IDENTIFICADOR"],
                                    "DB":"BUPP",
                                    "TABLES": [
                                        {"SIIU_COMPROMISO": None}
                                    ]
                                }
                            ]
                        },
                        {
                            "SIIU_ESTADO_PROYECTO": [
                                {
                                    "KEYS": ["DOCUMENTO_SOPORTE/IDENTIFICADOR"],
                                    "DB":"BUPP",
                                    "TABLES": [
                                        {"SIIU_DOCUMENTO_SOPORTE": None}
                                    ]
                                }
                            ]
                        },
                        {
```

Fig. 6. Grafo de la entidad Proyecto del SIIU.

Para la creación del diccionario requirió hacer un análisis de la base de datos, donde se buscaba información de las tablas: cantidad de registros, tipos de datos, existencia de una tabla en especifico, campos que no fueron llenados, datos con valores anómalos e información redundante; para automatizar el análisis se crearon un conjunto de comandos, como se muestra en la Figura 7.

```
commands.add_argument('--get_tables', action='store_true',
                      help='Get the table names from the DB')
commands.add_argument('--describe', type=str, nargs='*', dest='table_name_desc_list',
                      action='store',
                      help='Get the name, type and size of the columns of the given tables')
commands.add_argument('--filter_tables_by_prefix', type=str, dest='prefix', action='store',
                      help='Filter table names by a given prefix')
commands.add_argument('--match_table_name', type=str, dest='table_name_match',
                      help='Get the table names that match a given name')
commands.add_argument('--get_df', type=str, nargs='*', dest='table_df_args',
                      action='store',
                      help='Create a dataframe using the table from the DB')
commands.add_argument('--get_df_exclude', type=str, nargs='*',
                      dest='table_df_exclude_args', action='store',
                      help='Create a dataframe excluding the indicated columns')
commands.add_argument('--r_sample', type=float, dest='fraction',
                      help='Get a random sample from the DB')
commands.add_argument('--heatmap', type=str, dest='histogram_df_name', action='store',
                      help='Display a heatmap which shows missing values')
commands.add_argument('--get_ocurrences_percentage',  type=str, nargs='*',
                      dest='get_ocurrences_per_df', action='store',
                      help='Display data ocurrences and their percentage over the total amount')
commands.add_argument('--count_null', type=str, nargs=2, dest='count_null_args',
                      help='Get the amount of null values found on a table')
commands.add_argument('--exit', action='store_true',
                      help='Stops the program')
```

Fig. 7. Comandos Utilizados para Automatizar el Análisis de Tablas.

2. El software Kaypacha SIIU: para el cual se hizo el análisis de la base de datos y se consultó con la Vicerrectoría de Investigación sobre la información que necesitaban de las tablas, con la información proveida se creó el modelo y se hicieron adaptaciones a Kaypacha para automatizar la transformación de la base datos de un modelo relacional a otro modelo no relacional. En la (**Fig. 8**) se encuentra el pull request para hacer la conversión de Oracle a Mongo.

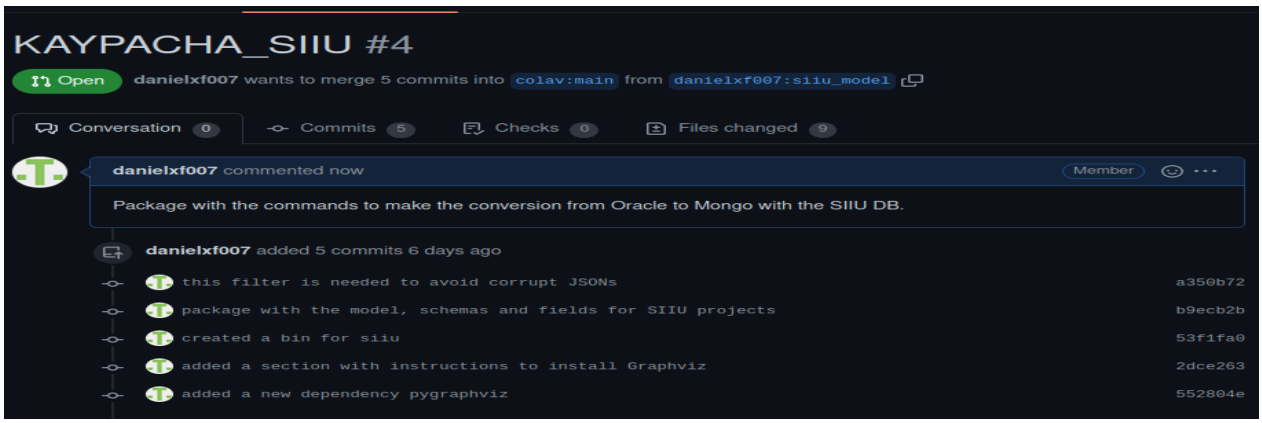

Fig. 8. Pull Request Hacer la Conversión de los Datos del SIIU.

3. Software Ukupacha: contiene los métodos usados para recorrer el grafo de la entidad Proyecto y extrae la información de la base datos, comenzando con la tabla principal.

4. Visualización del diccionario de datos: con el fin de comunicar los resultados obtenidos de una manera gráfica, se propuso un algoritmo que comenzó con un diagrama de clases donde se definieron los atributos y métodos necesarios para dibujar la estructura del diccionario (**Fig. 9**).

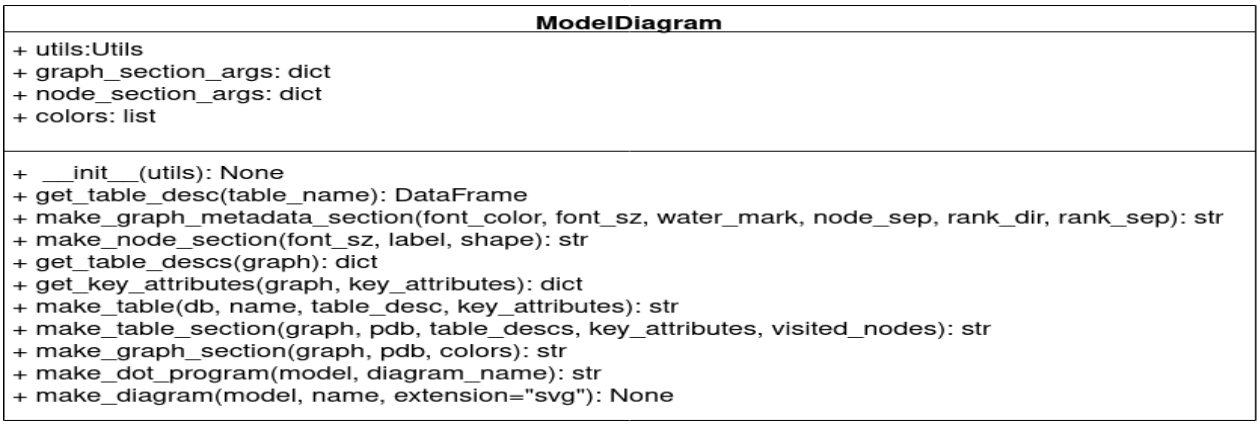

Fig. 9. Diagrama de la Clase Model Diagram.

El método principal para elaborar el diagrama consistió en crear un programa en formato DOT, con el cúal se indicó como se debe dibujar el diagrama del grafo, luego se configuró Pygraphviz para interpretar el programa y finalmente se procesó (**Fig. 10**).

```
def make_diagram(self, model, name, extension="svg"):
    """
    Create a diagram from a dot program using pygraphviz.
    Parameters:
    ----------
    model:dict
        Dictionary which contains metadata and other nodes.
    diagram_name:str
        Name of the diagram.
    extension:str
        File extension for the diagram.
    """
    dot_program = self.make_dot_program(model, name)
    G = pgv.AGraph(dot_program)
    G.layout(prog="dot")
    G.draw(f"{name}.{extension}")
```

Fig. 10. Función Principal para Crear el Diagrama.

Este proceso de visualización de la estructura concluyó con el gráfico del diccionario para el SIIU que contiene las tablas, sus nombres, tablespaces, atributos y conexión con las otras tablas definidas en el diccionario (**Fig. 11**).

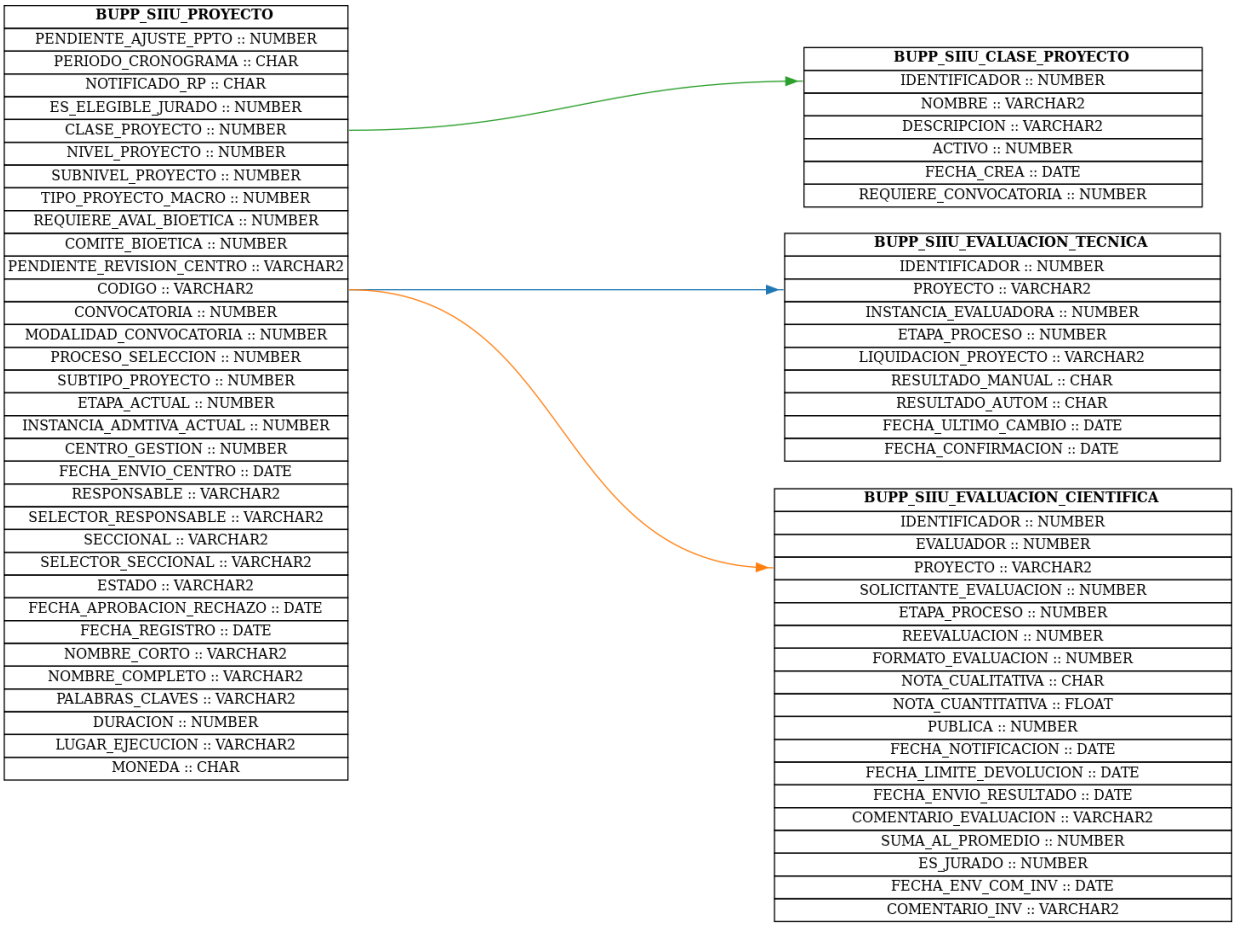

Fig. 11. Gráfico de la entidad Proyecto creada por la clase Model Diagram.

Fig. 12. Gráfico del Modelo Relacional de los Datos del SIIU.

Nota. Fuente Diagrama entregado por el SIIU

En las figuras (**Fig. 11**) y (**Fig. 12**) se muestran respectivamente el modelo relacional y la entidad Proyecto de la base de datos del SIIU.

El tercer componente es la Base de datos Mongo: almacena en un documento todos los registros de la tabla principal Proyecto donde cada uno tiene los datos de las tablas con las que se relaciona.

El último componente es el JSON: contiene los datos de Proyectos donde cada registro tiene la información de las tablas con las que se relaciona.

Análisis de datos y documentación de estrategias

En el proceso de análisis de lo datos se encontraron tablas con valores que afectan negativamente la calidad de la información, los problemas encontrados fueron:

1. Falta de estándares para definir valores booleanos, esto genera errores de interpretación acerca del significado del dato, las consultas deben tener en cuenta cada caso y esto puede causar problemas para generalizar el query, por ejemplo, con el fin de obtener los proyectos que requieren aval de bioética, el query debe ser:

   SELECT * FROM SIIU_PROYECTO

   WHERE REQUIERE_AVAL_BIOETICA=1 OR REQUIERE_AVAL_BIOETICA='S';

   En el peor de los casos, para un registro se debe evaluar ambas expresiones del OR y aún así no se incluiría e incluso si se identifica cuál valor REQUIERE_AVAL_BIOETICA usa para referirse a un valor booleano, se tendría que hacer el mismo proceso de identificación para las otras columnas.

TABLA I

PROBLEMA CON VALORES BOOLEANOS

Problema	
Valores booleanos que se representan de dos formas diferentes: 1. 0 y S, N.	
Tablas Afectadas	
1	En **SIIU_PROYECTO** en las columnas: **PENDIENTE_AJUSTE_PPTO, REQUIERE_AVAL_BIOETICA**, y **PENDIENTE_REVISION_CENTRO**
Soluciones Propuestas	
Estandarizar el valor booleano con números, siendo 0, falso y 1, verdadero. De esta manera se evitan las interpretaciones sobre el significado del valor.	

2. Valores nulos en campos requeridos, estos valores afectan directamente la creación de métricas e informes ya que en algunos casos como en LIQUIDACION_PROYECTO, no hay datos para trabajar (**TABLA II)**

TABLA II

PROBLEMA CON CAMPOS REQUERIDOS

Problema	
Valores nulos en campos requeridos.	
Tablas Afectadas	
1	En **SIIU_PARTICIPANTE_PROYECTO** se presenta el problema en las columnas: **CONFIRMADO, CONTRATO** y **DEPENDENCIA**
2	En **SIIU_EVALUACION_TECNICA** la columna **LIQUIDACION_PROYECTO** tiene todos sus valores nulos, este campo es útil para saber si el proyecto se cerró cuando se terminó.
3	En **SIIU_PERSONA_NATURAL** la columna **SEXO** tiene más de la mitad de sus valores nulos, esto hace difícil aplicar metodos estadisticos a este campo, por ejemplo para comparar la proporción de hombres y mujeres que participan en un proyecto.
Soluciones Propuestas	
Hacer obligatorio el llenado de campos que puedan ser útiles para hacer indicadores como **DEPENDENCIA** y **SEXO**.	
Revisar el ciclo de vida del proyecto para identificar porque los proyectos que terminaron no aparecen como liquidados.	

3. Formas diferentes para referirse al mismo valor, esto genera que en las consultas se tenga que tener en cuenta los valores diferentes y esto puede causar problemas para generalizar el query, por ejemplo: para obtener los participantes en un proyecto que tiene contrato por Prestación de Servicios, se debe crear el query (**TABLA III)**:

   SELECT * FROM SIIU_PARTICIPANTE_PROYECTO

   WHERE CONTRATO=’Prestación de Servicios’ OR CONTRATO=’Prestación de servicios’;

TABLA III
PROBLEMA CON REDUNDANCIA DE VALORES

Problema	
Formas diferentes para referirse al mismo valor.	
Tablas Afectadas	
1	En **SIIU_PARTICIPANTE_PROYECTO** la columna **CONTRATO** tiene valores que se escribieron diferente, pero el significado es el mismo, por ejemplo: Prestación de Servicios y Prestación de servicios.
Soluciones Propuestas	
Para los valores cuyo valor se conoce, como el tipo de contrato, se puede crear una lista de opciones donde el usuario pueda escoger un valor predeterminado, en lugar de escribirlo manualmente.	

4. Valores anómalos, en el caso de FECHA_EMISION se generaba un error en la extracción de los datos, ya que era una fecha que no podía ser procesada (**TABLA IV**)

TABLA IV
VALORES ANÓMALOS

Problema	
Datos con valores anómalos.	
Tablas Afectadas	
1	En **SIIU_TEXTO_DESCRIPTIVO** la columna **FECHA_EMISION** tiene una fecha 0001/01/28.
Soluciones Propuestas	
Revisar desde la plataforma del SIIU, si se están aplicando correctamente los filtros para que no pasen este tipo de valores.	

VI. CONCLUSIONES

Este proyecto ayudó a detectar anomalías en la información, tales como, información redundante, datos nulos de campos que se necesitan y proyectos que nunca se cierran, a través del análisis de los datos con herramientas como pandas y a partir de esto se pueden generar nuevas políticas que ayuden mejorar la captura de la información para generar datos de calidad.

Con este desarrollo se transformaron los datos relacionados de los proyectos del SIIU en una estructura lógica, la cual tiene las tablas más relevantes para elaboración de indicadores y métricas.

La estructura de la entidad es más fácil de entender que la del modelo relacional porque la entidad tiene una estructura lógica desde su creación en el diccionario mientras que el modelo relacional debe ser interpretado y no tiene una estructura lógica explícita. Finalmente la entidad ya tiene sus valores relacionados, por lo que no es necesario hacer consultas complicadas para relacionar los registros de tablas diferentes.

Este proceso permitió adquirir bases sobre Docker, además de que se afianzaron conocimientos previos especialmente en bases de datos y sistemas operativos y se dio la oportunidad de aportar, por primera vez, a un proyecto de código abierto.

REFERENCIAS

[1] «Unidades administrativas». https://bit.ly/3rXCSFK (accedido 14 de octubre de 2022).
[2] «Gestión de la investigación». https://bit.ly/3rXyDdp (accedido 14 de octubre de 2022).
[3] «CoLaV». http://colav.udea.edu.co/ (accedido 14 de octubre de 2022).
[4] «CoLaV», GitHub. https://github.com/colav (accedido 14 de octubre de 2022).
[5] «Introducción a las metodologías de desarrollo de software», uqbar-wiki. https://bit.ly/3eExCDT
(accedido 14 de octubre de 2022).
[6] «What is Unified Modeling Language (UML)?» https://bit.ly/3TqQHYT (accedido 14 de octubre de 2022).
[7] «Python PEP8», El Libro De Python. https://bit.ly/3eAdl2q (accedido 14 de octubre de 2022).
[8] «¿Qué es una base de datos relacional?» https://bit.ly/3evq6Ly (accedido 14 de octubre de 2022).
[9] «¿Qué es NoSQL o base de datos No Solo SQL? - Definición en WhatIs.com», ComputerWeekly.es. https://bit.ly/3g73ZLW (accedido 14 de octubre de 2022).
[10] «¿Qué es Git? - Azure DevOps». https://bit.ly/3rVqINB (accedido 14 de octubre de 2022).
[11] «¿Qué es Docker? - Definición en WhatIs.com», ComputerWeekly.es. https://bit.ly/3MznybN
(accedido 14 de octubre de 2022)
[12] «¿Qué es una base de datos de Oracle? | Pure Storage». https://bit.ly/3g2JO1z (accedido 14 de octubre de 2022).
[13] «¿Qué es Python? Definición, características y sus ventajas». https://bit.ly/3MzmSmL (accedido 14 de octubre de 2022).
[14] A. S. Alberca, «La librería Pandas», Aprende con Alf. https://bit.ly/3CYDMbf (accedido 14 de octubre de 2022)
[15] «PyGraphviz — PyGraphviz documentation». https://pygraphviz.github.io/ (accedido 14 de octubre de 2022).
[16] «Ukupacha», GitHub. https://github.com/colav/UkuPacha (accedido 14 de octubre de 2022).
[17] «Ukupacha», GitHub. https://github.com/colav/KayPacha (accedido 14 de octubre de 2022).