Elaboración de un modelo relacional de base de datos, para la Administración y Seguimiento de Proyectos del Departamento Nacional de Planeación

Paula M. Bohorquéz M.[1], Diana C. Gómez B.[2]

Facultad de Ingeniería y Ciencias Básicas
Universidad Central
Maestría en Analítica de Datos
Curso de Bases de Datos
Bogotá, Colombia
[1]pbohorquezm@ucentral.edu.co
[2]dgomezb10@ucentral.edu.co

Octubre 18, 2024

Contents

1 Introducción **3**

2 Características del proyecto de investigación que hace uso de Bases de Datos **3**
2.1 Titulo del proyecto de investigación . . . 3
2.2 Objetivo general . . . 3
2.2.1 Objetivos especificos . . . 3
2.3 Alcance . . . 3
2.4 Pregunta de investigación . . . 3
2.5 Hipotesis . . . 3

3 Reflexiones sobre el origen de datos e información **4**
3.1 ¿Cual es el origen de los datos e información ? (Max 100 Palabras) - (*Primera entrega*) . . . 4
3.2 ¿Cuales son las consideraciones legales o eticas del uso de la información? (Max 100 Palabras) - (*Primera entrega*) . . . 4
3.3 ¿Cuales son los retos de la información y los datos que utilizara en la base de datos en terminos de la calidad y la consolidación? (Max 100 Palabras) - (*Primera entrega*) . . . 4
3.4 ¿Que espera de la utilización de un sistema de Bases de Datos para su proyecto? (Max 100 Palabras) - (*Primera entrega*) . . . 4

4 Diseño del Modelo de Datos del SMBD (Sistema Manejador de Bases de Datos)(*Primera entrega*) **5**
4.1 Características del SMBD (Sistema Manejador de Bases de Datos) para el proyecto (*Primera entrega*) . . . 5
4.2 Diagrama modelo de datos (*Primera entrega*) . . . 5
4.3 Imágenes de la Base de Datos (*Primera entrega*) . . . 5
4.4 Código SQL - lenguaje de definición de datos (DDL) (*Primera entrega*) . . . 7
4.5 Código SQL - Manipulación de datos (DML) (*Primera entrega*) . . . 9
4.6 Código SQL + Resultados: Vistas (*Primera entrega*) . . . 10
4.7 Código SQL + Resultados: Triggers (*Primera entrega*) . . . 10
4.8 Código SQL + Resultados: Funciones (*Primera entrega*) . . . 11
4.9 Código SQL + Resultados: procedimientos almacenados (*Primera entrega*) . . . 11

5 Bases de Datos No-SQL (*Segunda entrega*) 12
5.1 Diagrama Bases de Datos No-SQL (*Segunda entrega*) 12
5.2 SMBD utilizado para la Base de Datos No-SQL (*Segunda entrega*) 12

6 Aplicación de ETL (Extract, Transform, Load) y Bodega de Datos (*Tercera entrega*) 13
6.1 Ejemplo de aplicación de ETL y Bodega de Datos (*Tercera entrega*) 13
6.2 Automatización de Datos (*Tercera entrega*) 15
6.3 Integración de Datos (*Tercera entrega*) 15

7 Proximos pasos (*Tercera entrega*) 16

8 Lecciones aprendidas (*Tercera entrega*) 16

9 Bibliografía 16

1 Introducción

El Departamento Nacional de Planeación (DNP) es la entidad técnica encargada de diseñar, coordinar y apoyar la planificación del presupuesto de los recursos de inversión del país. Actualmente, existen diversas fuentes de información que permiten conocer el manejo de los proyectos, incluyendo la asignación de presupuestos y el avance de cada uno, contratos, entre otros. Sin embargo, no se cuenta con una base de datos consolidada que permita centralizar y optimizar el seguimiento de estos proyectos. La existencia de una herramienta de este tipo es esencial para garantizar la transparencia, mejorar la eficiencia en la gestión y fortalecer la precisión en la toma de decisiones.

En este contexto, el objetivo de este trabajo es diseñar un modelo de base de datos que permita gestionar y analizar eficientemente los datos relacionados con los proyectos aprobados por el DNP. Este modelo, relacional,está orientado en garantizar la trazabilidad de los recursos asignados y utilizados; de esta menera el modelo contribuirá a un monitoreo efectivo a lo largo del tiempo, promoviendo una mayor rendición de cuentas y fortalecimiento de la gestión pública. .

2 Características del proyecto de investigación que hace uso de Bases de Datos

2.1 Titulo del proyecto de investigación

Elaboración de un modelo relacional de base de datos, para la Administración y Seguimiento de Proyectos del Departamento Nacional de Planeación (DNP).

2.2 Objetivo general

Diseñar un modelo de base de datos que centralice, gestione y analice de manera eficiente los datos relacionados con los proyectos aprobados por el DNP, promoviendo la transparencia, garantizando la trazabilidad de los recursos y facilitando un monitoreo efectivo y continuo a lo largo del tiempo.

2.2.1 Objetivos especificos

- Evaluar la calidad de los datos para garantizar su integridad y coherencia, asegurando que la información utilizada en el modelo de base de datos sea precisa y confiable..
- Elaborar un diagrama entidad-relación (ER) para representar de manera estructurada las entidades y relaciones de los proyectos del Departamento Nacional de Planeación, facilitando el diseño y comprensión del modelo de base de datos.
- Desarrollar un conjunto de consultas que validen el correcto funcionamiento del modelo de base de datos implementado, permitiendo verificar la precisión en la gestión y recuperación de la información.

2.3 Alcance

El alcance de este proyecto es diseñar un modelo de base de datos que permita gestionar y analizar de manera eficiente los datos relacionados con los proyectos aprobados por el Departamento Nacional de Planeación (DNP) durante el período 2023-2024 para Colombia.

2.4 Pregunta de investigación

¿Cómo diseñar un modelo de base de datos que centralice, gestione y analice de manera eficiente los datos relacionados con los proyectos aprobados por el DNP, promoviendo la transparencia, garantizando la trazabilidad de los recursos y facilitando un monitoreo efectivo y continuo a lo largo del tiempo.)?

2.5 Hipotesis

El diseño de un modelo de base de datos relacional eficiente permitirá una gestión y análisis eficiente de los proyectos aprobados por el Departamento Nacional de Planeación (DNP), mejorando la trazabilidad de los proyectos y la visibilidad de su avance, lo que contribuirá a un mayor control y aprovechamiento de los recursos destinados a dichos proyectos.

3 Reflexiones sobre el origen de datos e información

El DNP al ser una entidad pública debe garantizar el acceso a la información según la Ley 1712 de 2014; además, el decreto 1081 de 2015 especifica la responsabilidad de las entidades públicas y la disposición de los datos a través de la plataforma digital de Datos Abiertos de Colombia (https://www.datos.gov.co/) de donde se tomaron 6 fuentes de datos para ser analizados y procesados.

3.1 ¿Cual es el origen de los datos e información ? (Max 100 Palabras) - (*Primera entrega*)

El DNP obtiene la información a través de plataformas como el Sistema Electrónico de Contratación Pública (SECOP), el Sistema de Monitoreo, Seguimiento, Control y Evaluación (SMSCE), y otros repositorios de datos abiertos proporcionados por el Gobierno de Colombia. Estos sistemas permiten a las entidades públicas reportar el estado de avance, los recursos asignados y la ejecución de los proyectos, asegurando que la información esté disponible para el público. Además, el DNP puede complementar esta información con datos de otras entidades y ministerios que también gestionan o participan en proyectos financiados con recursos públicos. Todo esto bajo la premisa de promover la transparencia, la eficiencia en el uso de los recursos y el control ciudadano sobre las inversiones del Estado colombiano.

3.2 ¿Cuales son las consideraciones legales o eticas del uso de la información? (Max 100 Palabras) - (*Primera entrega*)

El uso de la información sobre los proyectos que sigue el DNP en Colombia se rige por consideraciones legales y éticas clave. La Ley de Transparencia (1712 de 2014) garantiza el acceso público a estos datos, mientras que la Ley de Protección de Datos Personales (1581 de 2012) protege la privacidad en caso de información sensible. Éticamente, se promueve la transparencia, la rendición de cuentas y el uso responsable de los datos. Además, se debe respetar la confidencialidad y los derechos de autor, asegurando que la información no sea manipulada ni utilizada de forma malintencionada o para fines indebidos.

3.3 ¿Cuales son los retos de la información y los datos que utilizara en la base de datos en terminos de la calidad y la consolidación? (Max 100 Palabras) - (*Primera entrega*)

Los retos en la gestión de la información en las bases de datos del DNP incluyen garantizar la calidad de los datos, asegurando precisión, completitud. La normalización es clave para consolidar datos de distintas fuentes. Muchas veces no todas las entidades tienen la información actualizada lo que incrementa la dificultad de unificar en una sola base toda la información.

3.4 ¿Que espera de la utilización de un sistema de Bases de Datos para su proyecto? (Max 100 Palabras) - (*Primera entrega*)

Se espera una gestión eficiente y organizada de la información, que facilite la recolección, almacenamiento y consulta de datos de manera rápida y precisa, para poder generar una visualización ordenada y única de todos los datos, reduciendo duplicidad que es lo más común en estos datos al venir de tantas fuentes de información.

4 Diseño del Modelo de Datos del SMBD (Sistema Manejador de Bases de Datos)(*Primera entrega*)

4.1 Características del SMBD (Sistema Manejador de Bases de Datos) para el proyecto (*Primera entrega*)

4.2 Diagrama modelo de datos (*Primera entrega*)

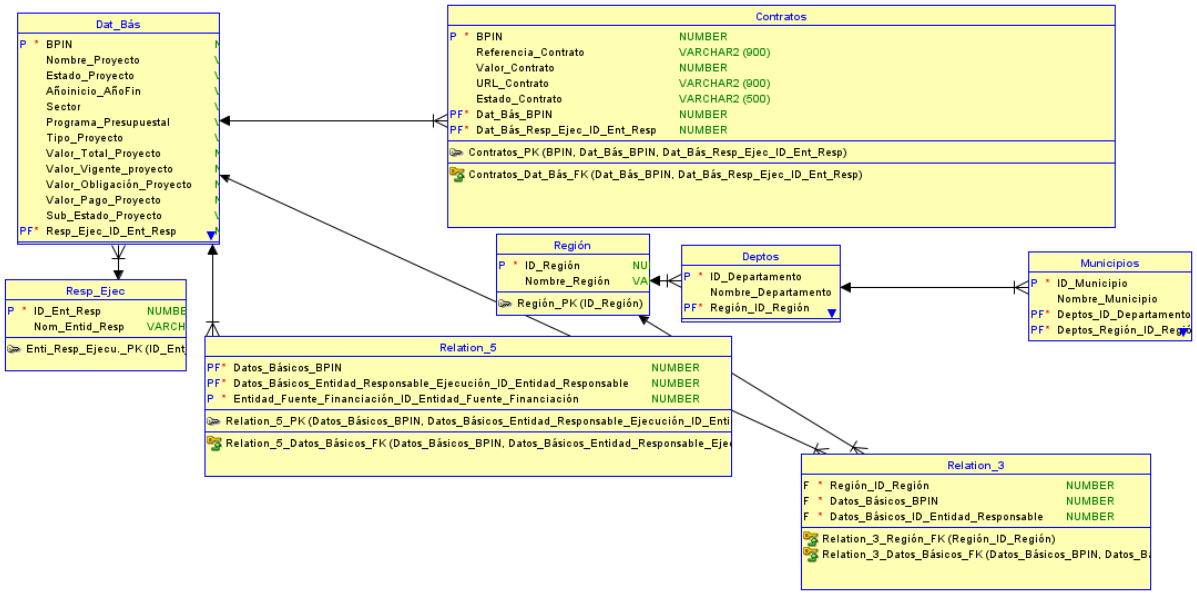

Figure 1: Diagrama Entidad-Relación Proyecto DNP

4.3 Imágenes de la Base de Datos (*Primera entrega*)

taller dml.sql | PROYECTO V1.sql | Página de bienvenida | DCGB | DAT_BÁS

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | S

Acciones...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BPIN	NUMBER	No	(null)	1	(null)
2	NOMBRE_PROYECTO	VARCHAR2(1000 ...	Yes	(null)	2	(null)
3	ESTADO_PROYECTO	VARCHAR2(500 B...	Yes	(null)	3	(null)
4	AÑOINICIO_AÑOFIN	VARCHAR2(200 B...	Yes	(null)	4	(null)
5	SECTOR	VARCHAR2(500 B...	Yes	(null)	5	(null)
6	PROGRAMA_PRESUP...	VARCHAR2(900 B...	Yes	(null)	6	(null)
7	TIPO_PROYECTO	VARCHAR2(900 B...	Yes	(null)	7	(null)
8	VALOR_TOTAL_PRO...	NUMBER	Yes	(null)	8	(null)
9	VALOR_VIGENTE_P...	NUMBER	Yes	(null)	9	(null)
10	VALOR_OBLIGACIÓ...	NUMBER	Yes	(null)	10	(null)
11	VALOR_PAGO_PROY...	NUMBER	Yes	(null)	11	(null)
12	SUB_ESTADO_PROY...	VARCHAR2(1000 ...	Yes	(null)	12	(null)
13	RESP_EJEC_ID_EN...	NUMBER	No	(null)	13	(null)

Figure 2: Base de datos Datos Básicos

taller dml.sql | PROYECTO V1.sql | Página de bienvenida | DCGB | CONTRATOS

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Acciones...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BPIN	NUMBER	No	(null)	1	(null)
2	REFERENCIA_CONT...	VARCHAR2(900 B...	Yes	(null)	2	(null)
3	VALOR_CONTRATO	NUMBER	Yes	(null)	3	(null)
4	URL_CONTRATO	VARCHAR2(900 B...	Yes	(null)	4	(null)
5	ESTADO_CONTRATO	VARCHAR2(500 B...	Yes	(null)	5	(null)
6	DAT_BÁS_BPIN	NUMBER	No	(null)	6	(null)
7	DAT_BÁS_RESP_EJ...	NUMBER	No	(null)	7	(null)

Figure 3: Base de datos Contratos

PROYECTO V1.sql | DCGB | REGIÓN

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Acciones...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_REGIÓN	NUMBER	No	(null)	1	(null)
2	NOMBRE_REGIÓN	VARCHAR2(60 BYTE)	Yes	(null)	2	(null)

Figure 4: Base de datos Región

PROYECTO V1.sql | DCGB | PROVEEDORES

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Acciones...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_PROVEEDOR	NUMBER	No	(null)	1	(null)
2	NOMBRE_PROVEEDOR	VARCHAR2(500 B...	No	(null)	2	(null)
3	TIPO_DOCUMENTO	VARCHAR2(50 BYTE)	Yes	(null)	3	(null)

Figure 5: Base de datos Proveedores

taller dml.sql | PROYECTO V1.sql | Página de bienvenida | DCGB | DEPTOS

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Acciones...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_DEPARTAMENTO	NUMBER	No	(null)	1	(null)
2	NOMBRE_DEPARTAM...	VARCHAR2(500 B...	Yes	(null)	2	(null)
3	REGIÓN_ID_REGIÓN	NUMBER	No	(null)	3	(null)

Figure 6: Base de datos Departamentos

PROYECTO V1.sql | DCGB | MUNICIPIOS

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Acciones...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_MUNICIPIO	NUMBER	No	(null)	1	(null)
2	NOMBRE_MUNICIPIO	VARCHAR2(500 B...	Yes	(null)	2	(null)
3	DEPTOS_ID_DEPAR...	NUMBER	No	(null)	3	(null)
4	DEPTOS_REGIÓN_I...	NUMBER	No	(null)	4	(null)

Figure 7: Base de datos Municipios

PROYECTO V1.sql | DCGB | RESP_EJEC

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Acciones...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID_ENT_RESP	NUMBER	No	(null)	1	(null)
2	NOM_ENTID_RESP	VARCHAR2(500 B...	Yes	(null)	2	(null)

Figure 8: Base de datos Responsable de Ejecución

4.4 Código SQL - lenguaje de definición de datos (DDL) (*Primera entrega*)

```
CREATE TABLE contratos (
    bpin NUMBER NOT NULL,
    referencia_contrato VARCHAR2(900),
    valor_contrato NUMBER,
    url_contrato VARCHAR2(900),
    estado_contrato VARCHAR2(500)
);

ALTER TABLE contratos
    ADD CONSTRAINT contratos_pk PRIMARY KEY ( bpin);

CREATE TABLE dat_bas (
bpin                        NUMBER NOT NULL,
nombre_proyecto             VARCHAR2(1000),
estado_proyecto             VARCHAR2(500),
añoinicio_añofin            VARCHAR2(200),
sector                      VARCHAR2(500),
programa_presupuestal       VARCHAR2(900),
tipo_proyecto               VARCHAR2(900),
valor_total_proyecto        NUMBER,
valor_vigente_proyecto      NUMBER,
valor_obligación_proyecto NUMBER,
valor_pago_proyecto         NUMBER,
sub_estado_proyecto         VARCHAR2(1000),
resp_ejec_id_ent_resp       NUMBER NOT NULL
);

ALTER TABLE dat_bas ADD CONSTRAINT dat_bas_pk PRIMARY KEY
( bpin,resp_ejec_id_ent_resp );


CREATE TABLE deptos (
id_departamento     NUMBER NOT NULL,
```

```
nombre_departamento VARCHAR2(500)
);


ALTER TABLE deptos ADD CONSTRAINT deptos_pk PRIMARY KEY
(id_departamento);


CREATE TABLE fuent_finan (
id_fuente_finan       NUMBER NOT NULL,
"Enti_Fuente_Financ." VARCHAR2(500),
valor_vigente_fuente  NUMBER,
valor_pagado_fuente   NUMBER);

ALTER TABLE fuent_finan
ADD CONSTRAINT fuent_finan_pk PRIMARY KEY ( id_fuente_finan);

CREATE TABLE municipios (
id_municipio            NUMBER NOT NULL,
nombre_municipio        VARCHAR2(500)
);

ALTER TABLE municipios
ADD CONSTRAINT municipios_pk PRIMARY KEY ( id_municipio);

CREATE TABLE proveedores (
id_proveedor     NUMBER NOT NULL,
nombre_proveedor VARCHAR2(500) NOT NULL,
tipo_documento   VARCHAR2(50));

ALTER TABLE proveedores
ADD CONSTRAINT proveedores_pk PRIMARY KEY ( id_proveedor);

CREATE TABLE region1 (
id_region     NUMBER NOT NULL,
nombre_region VARCHAR2(60));

ALTER TABLE region1 ADD CONSTRAINT region1_pk PRIMARY KEY ( id_region );



CREATE TABLE resp_ejec (
id_ent_resp    NUMBER NOT NULL,
nom_entid_resp VARCHAR2(500));

CREATE TABLE relation_70 (
región_id_región     NUMBER NOT NULL,
datos_básicos_bpin   NUMBER NOT NULL,
dat_bas_resp_ejec_id_ent_resp_      NUMBER NOT NULL
);

CREATE TABLE relation_50 (
    dat_bas_bpin           NUMBER NOT NULL,
    dat_bas_resp_ejec_id_ent_resp_ NUMBER NOT NULL
);
```

```
ALTER TABLE relation_71
    ADD CONSTRAINT relation_71_pk PRIMARY KEY ( dat_bas_bpin,
    dat_bas_resp_ejec_id_ent_resp_);

ALTER TABLE contratos
ADD CONSTRAINT contratos_dat_bas_fk FOREIGN KEY ( dat_bás_bpin,
 dat_bás_resp_ejec_id_ent_resp)
        REFERENCES dat_bas ( bpin, resp_ejec_id_ent_resp );


ALTER TABLE deptos
 ADD CONSTRAINT deptos_región_fk FOREIGN KEY ( región_id_región )
    REFERENCES región ( id_región );

ALTER TABLE municipios
ADD CONSTRAINT municipios_deptos_fk FOREIGN KEY
(deptos_id_departamento,deptos_región_id_región )
REFERENCES deptos ( id_departamento, región_id_región );


ALTER TABLE relation_70
ADD CONSTRAINT relation_70_datos_básicos_fk FOREIGN KEY
(DATOS_BÁSICOS_BPIN,dat_bas_resp_ejec_id_ent_resp_ )
REFERENCES dat_bas ( bpin,resp_ejec_id_ent_resp );

ALTER TABLE relation_70
    ADD CONSTRAINT relation_70_región_fk FOREIGN KEY ( región_id_región )
        REFERENCES región ( id_región );


Los códigos anteriores crean el esquema de la base de datos para recibir información
sobre los proyectos del Departamento Nacional de Proyectos, descritos de la siguiente manera:

* Dat_Bas: Tabla que almacena información de todos los proyectos,
incluyendo el valor del proyecto y el estado de su ejecución.
* Resp_Ejec: Tabla donde se registra la entidad responsable del
proyecto junto con su nombre correspondiente.
* Región: Ubicación del proyecto a nivel nacional.
* Contratos: Tabla que contiene información consolidada sobre el
contrato del proyecto, su estado de ejecución y la URL de
seguimiento.
* Proveedores: Tabla que reúne los datos de los proveedores
asociados a los contratos.
* Deptos: Tabla que lista los departamentos a nivel nacional junto con su código de homologación.
* Municipios: Tabla que presenta los municipios a nivel nacional,
también con su código de homologación.
* Relaciones:  Contiene las relaciones de uno a muchos entre cada una de las tablas.
```

4.5 Código SQL - Manipulación de datos (DML) (*Primera entrega*)

```
INSERT INTO CONTRATOS (BPIN, REFERENCIA_CONTRATO, VALOR_CONTRATO, URL_CONTRATO, ESTADO_CONTRATO)
VALUES (2023157200002,'MSN OP 06 DE
2023',831849119.7,'','En ejecución');

INSERT INTO DEPTOS (ID_DEPARTAMENTO, NOMBRE_DEPARTAMENTO, "REGIÓN_ID_REGIÓN")
VALUES (5,'Antioquia',2);
```

```
INSERT INTO REGION1 (ID_REGION,NOMBRE_REGIO) VALUES (3,'Caribe');

INSERT INTO
PROVEEDORES(ID_PROVEEDOR,NOMBRE_PROVEEDOR,TIPO_DOCUMENTO) VALUES
(1040322659,'Eduardo Arango Tamayo','Cedula de Ciudadania');

INSERT INTO DAT_BAS
(BPIN,NOMBRE_PROYECTO,ESTADO_PROYECTO,AÑOINICIO_AÑOFIN,SECTOR,
PROGRAMA_PRESUPUESTAL,TIPO_PROYECTO,VALOR_TOTAL_PROYECTO,
VALOR_VIGENTE_PROYECTO,VALOR_OBLIGACIÓN_PROYECTO,
VALOR_PAGO_PROYECTO,SUB_ESTADO_PROYECTO,RESP_EJEC_ID_ENT_RESP) VALUES
(2023053060033,'Apoyo A La Rendición Y Presentación De Gestión De
La Administración Municipal En El Municipio De   Giraldo','En Ejecución',2023-2024,
'Gobierno Territorial',
'4599 Fortalecimiento a la gestión y dirección de la administración pÃºblica
territorial','T',54850519,54850519,43300000,43300000,'En ejecución (PGN, Territorio)',5306);


DELETE FROM municipios
WHERE ROWID NOT IN (
    SELECT MIN(ROWID)
    FROM municipios
    GROUP BY id_municipio);
```

4.6 Código SQL + Resultados: Vistas (*Primera entrega*)

```
CREATE OR REPLACE VIEW vista_contratos_activos AS
SELECT bpin, referencia_contrato, valor_contrato, estado_contrato
FROM contratos
WHERE estado_contrato = 'En ejecución';

Esta vista muestra todos los contratos que están en ejecución de la tabla contratos.

CREATE OR REPLACE VIEW vista_proyectos_contratos AS
SELECT d.bpin, d.nombre_proyecto, c.referencia_contrato, c.valor_contrato, c.estado_contrato
FROM dat_bas d
JOIN contratos c ON d.bpin = c.bpin;

Esta vista relaciona los contratos con los proyectos de la tabla dat_bas
```

4.7 Código SQL + Resultados: Triggers (*Primera entrega*)

```
CREATE OR REPLACE TRIGGER check_valor_contrato
BEFORE INSERT ON contratos
FOR EACH ROW
BEGIN
IF :NEW.valor_contrato < 0 THEN
RAISE_APPLICATION_ERROR(-20001,
'El valor del contrato no puede ser negativo.');

END IF;
END;

Se crea trigger para asegurar que no se puedan insertar contratos con un valor negativo
```

en la tabla contratos. Si alguien intenta hacerlo, se genera un error y la inserción se cancela. Esto se hace con el fin de mantener la integridad de los datos en la base de datos.

```
CREATE OR REPLACE TRIGGER update_estado_contrato
BEFORE UPDATE ON contratos
FOR EACH ROW
BEGIN
    -- Verificar si el valor del contrato ha cambiado
    IF :NEW.valor_contrato <> :OLD.valor_contrato THEN
        -- Cambiar el estado a 'Actualizado'
        :NEW.estado_contrato := 'Actualizado';
    END IF;
END;
```

Se crea trigger para asegurar que una vez que se actualice el valor de un contrato en la tabla contratos, su estado se cambie automáticamente a "Actualizado". Puedes modificar la lógica de actualización del estado según tus necesidades.

4.8 Código SQL + Resultados: Funciones (*Primera entrega*)

Esta función devuelve los detalles de los proyectos asociados a un contrato específico.

```
CREATE OR REPLACE FUNCTION obtener_proyecto_por_contrato(
    p_bpin NUMBER
) RETURN SYS_REFCURSOR IS
    v_cursor SYS_REFCURSOR;
BEGIN
    OPEN v_cursor FOR
    SELECT d.bpin, d.nombre_proyecto, c.referencia_contrato, c.valor_contrato, c.estado_contrato
    FROM dat_bas d
    JOIN contratos c ON d.bpin = c.bpin
    WHERE c.bpin = p_bpin;

    RETURN v_cursor;
END obtener_proyecto_por_contrato;
```

La siguiente función devuelve el número de contratos que están en ejecución.

```
CREATE OR REPLACE FUNCTION contar_contratos_activos RETURN NUMBER IS
v_count NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_count
    FROM contratos
    WHERE estado_contrato = 'En ejecución';

    RETURN v_count;
END contar_contratos_activos;
```

4.9 Código SQL + Resultados: procedimientos almacenados (*Primera entrega*)

Este procedimiento cuenta el número de contratos que están celebrados y devuelve el resultado.

```
CREATE OR REPLACE PROCEDURE contar_contratos_celebrados(
    o_count OUT NUMBER
) IS
BEGIN
    SELECT COUNT(*)
    INTO o_count
    FROM contratos
    WHERE estado_contrato = 'CELEBRADO';
END contar_contratos_celebrados;
```

5 Bases de Datos No-SQL (*Segunda entrega*)

5.1 Diagrama Bases de Datos No-SQL (*Segunda entrega*)

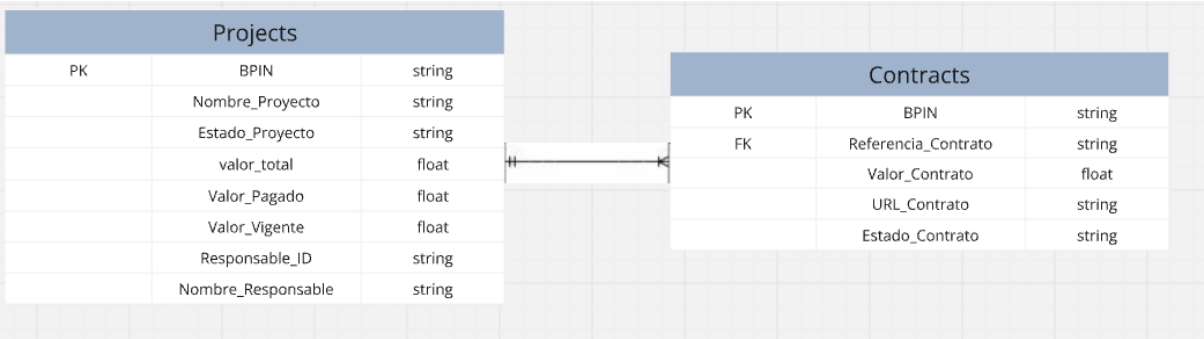

Figure 9: Meta-Modelo Lógico Proyecto DNP

En los modelos meta-modelo lógico se describen las propuedades susceptibles a ser utilizados, sus tipos (cadenas, numéricos, etc.). Las relaciones entre concetos, dependen de las que se permitan dentro de un documento No-SQL.

Este modelo representa la gestión de información sobre los Proyectos y sus contratos asociados. Está estructurado en un formato semidesnormalizado, usual en bases de datos No-SQL, donde los datos relacionados pueden tratarse como colecciones separadas, manteniendo claves para su referencia.

La colección Projects tiene la información básica de cada proyecto como, Identificador único del proyecto, Nombre de proyecto, Valor ya pagado del presupuesto.

La colección Contracts detalla los contratos asociados a cada proyecto, tiene campos como Referencias de contratos (ID contratos), Valor del contrato, URL del contrato.

5.2 SMBD utilizado para la Base de Datos No-SQL (*Segunda entrega*)

Se cargó una muestra de la información total en las dos colecciones (Proyectos y Contratos). La conexión con python para cargar los datos simplificó mucho el proceso, permitiendo una integración fluida entre esta y Mongo Atlas.
A partir de esto se pueden realizar consultas rápidas y eficientes para los estados de los proyectos y sus contratos; por ejemplo se puede identificar los proyectos que tienen los mayores presupuestos o los que han recibido más dinero a la fecha y su estado de ejecución.

Mediante el BPIN se facilita la trazabilidad, lo cual se alinea al objetivo de monitorear el avance y los recursos de cada proyecto.

Hay datos faltantes que pueden ser gestionados para evitar inconsistencias futuras en los análisis que se requieran.

MongoDB permite que estas colecciones sean escalables, soportando el crecimiento tanto de volumen de documentos como relaciones entre ellos.

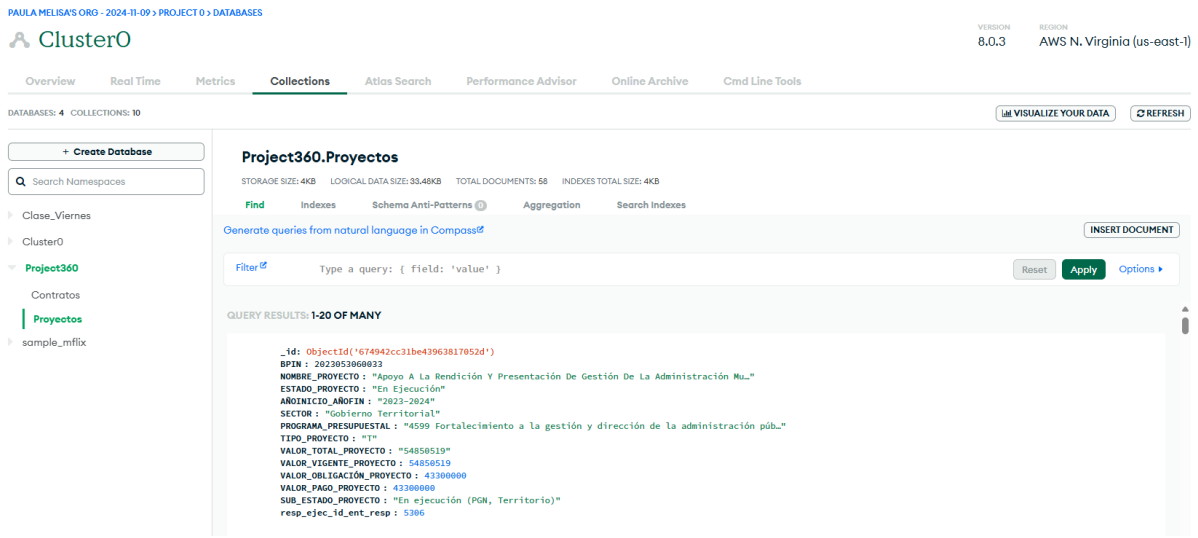

Figure 10: Colección Proyectos de la base de datos Project360 Mongo Atlas

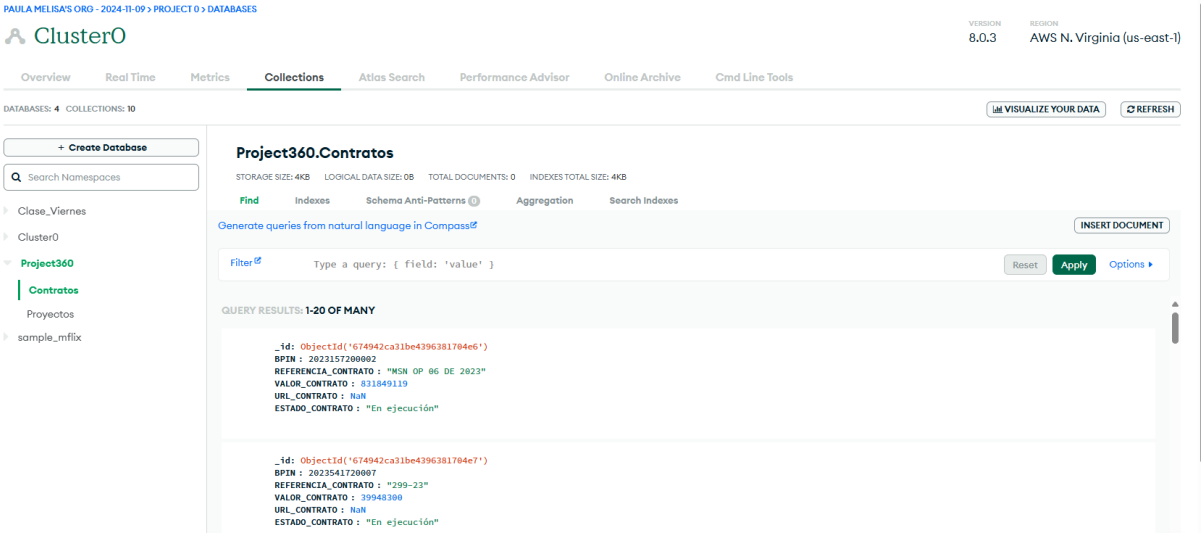

Figure 11: Colección Contratos de la base de datos Project360 Mongo Atlas

6 Aplicación de ETL (Extract, Transform, Load) y Bodega de Datos (*Tercera entrega*)

6.1 Ejemplo de aplicación de ETL y Bodega de Datos (*Tercera entrega*)

En esta fase del proyecto, se llevó a cabo el proceso ETL (Extracción, Transformación, Carga), una metodología clave para el manejo eficiente de datos. Utilizando Python, se ejecutó una serie de transformaciones en los datos con el objetivo de garantizar su calidad y fiabilidad antes de ser almacenados en un sistema de base de datos. Primero, se realizó la extracción de los datos desde una fuente externa, en este caso, el Departamento Nacional de Planeación (DNP). Luego, durante la fase de transformación, se realizaron varias tareas críticas para limpiar y estandarizar los datos:

* Eliminación de columnas innecesarias: Se eliminaron las columnas que no eran relevantes para el análisis, lo que permitió concentrarse únicamente en la información esencial.

* Renombrado de columnas: Algunas columnas fueron renombradas para asegurar que los nombres fueran más claros y fáciles de comprender, facilitando la posterior manipulación de los datos.

* Conversión de texto a minúsculas: Se transformaron todas las cadenas de texto de mayúsculas a minúsculas, con el fin de mantener una consistencia en el formato y evitar posibles discrepancias en el análisis.

* Eliminación de valores nulos: Se reemplazaron los valores faltantes (NaN) por un valor predeterminado, como "Desconocido", para evitar que los registros incompletos interfirieran en el proceso de análisis.

* Corrección de caracteres especiales: Se solucionaron problemas de codificación que pudieran haber afectado la visualización de caracteres especiales, asegurando que todos los textos fueran correctamente representados.

* Eliminación de duplicados: Se eliminaron las filas duplicadas para garantizar que cada registro fuera único, evitando redundancias que pudieran distorsionar los resultados.

Una vez que los datos fueron transformados y limpiados, el siguiente paso fue cargar la información. Para ello, se utilizó MongoDB como sistema de almacenamiento. A través de la autenticación y conexión a MongoDB, se realizaron solicitudes para obtener información detallada sobre los datos básicos de los proyectos del DNP. Esta información fue almacenada temporalmente en estructuras de datos de Python, como listas y diccionarios, lo que permitió una manipulación flexible y eficiente.

```python
import pandas as pd
import numpy as np
from datetime import datetime

# 1. **Extract (Extracción)**
# Leer archivo Excel
read_file = pd.read_excel("DATOS BASICOS.xlsx")
# 2. **Transform (Transformación)**
# 2.1. Eliminar columnas innecesarias
columns_to_drop = ['PlanDesarrolloNacional']
read_file.drop(columns=columns_to_drop, axis=1, inplace=True)
# 2.2. Renombrar columnas para un mejor formato
read_file.rename(columns={'Horizonte': 'Año_del_proyecto'}, inplace=True)
# 2.3. Convertir a minúsculas todas las cadenas de texto en ciertas columnas
read_file['NombreProyecto'] = read_file['NombreProyecto'].str.lower()
# 2.4. Limpiar valores nulos (NaN)
read_file['EstadoProyecto'] = read_file['EstadoProyecto'].fillna('Desconocido')
# 2.5. Corregir codificación de caracteres (si es necesario)
read_file['EstadoProyecto'] = read_file['EstadoProyecto'].apply(
    lambda x: x.encode('utf-8', errors='ignore').decode('utf-8', errors='ignore') if isinstance(x, str) else x
)
read_file['NombreProyecto'] = read_file['NombreProyecto'].apply(
    lambda x: x.encode('utf-8', errors='ignore').decode('utf-8', errors='ignore') if isinstance(x, str) else x
)

# 2.6. Eliminar filas duplicadas (si es necesario)
read_file.drop_duplicates(inplace=True)
# 3. **Load (Carga)**
# 3.1. Guardar el dataframe transformado en un nuevo archivo Excel
read_file.to_excel('DATOS_BASICOS_PROCESADOS.xlsx', index=False)
# 3.2. Cargar los datos a una base de datos (por ejemplo, MongoDB, MySQL, etc.)
from sqlalchemy import create_engine
# Conectar a la base de datos (en este caso, una base de datos SQL)
from pymongo.mongo_client import MongoClient
from pymongo.server_api import ServerApi
# Conectar a MongoDB Atlas
uri = "mongodb+srv://dgomezb10:Carolina90*@cluster0.wfv03.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0"
```

```python
client = MongoClient(uri, server_api=ServerApi('1'))
# Enviar un ping para confirmar la conexión
try:
    client.admin.command('ping')
    print("Conexión exitosa a MongoDB Atlas!")
except Exception as e:
    print(f"Error de conexión: {e}")
# Seleccionar la base de datos y la colección
db = client["proyecto_final"]
collection = db["proyecto"]
# Convertir el DataFrame a una lista de diccionarios
data_to_insert = read_file.to_dict(orient='records')
# Insertar los datos en MongoDB
collection.insert_many(data_to_insert)
print("Datos insertados correctamente en MongoDB.")
```

Figure 12: Parte 1 ETL

Finalmente, los datos transformados se exportaron a un archivo Excel denominado "DATOS BASICOS PROCESADOS.xlsx", lo que facilitó su acceso y posterior análisis. En resumen, este proceso ETL no solo mejoró la calidad y estructura de los datos, sino que también permitió su integración fluida en un entorno de base de datos para su uso futuro.

```
0  Colombia Potencia Mundial de la Vida (2022-2026)        5.485052e+07
1  Colombia Potencia Mundial de la Vida (2022-2026)        1.610235e+08
2  Colombia Potencia Mundial de la Vida (2022-2026)        9.273763e+08
3  Colombia Potencia Mundial de la Vida (2022-2026)        1.500000e+07
4  Colombia Potencia Mundial de la Vida (2022-2026)        2.342078e+07

   ValorVigenteProyecto  ValorObligacionProyecto  ValorPagoProyecto  \
0         5.485052e+07               43300000.0         43300000.0
1         1.610235e+08                5100000.0          5100000.0
2         9.273763e+08                      0.0                0.0
3         1.500000e+07               15000000.0         15000000.0
4         2.342078e+07               16746800.7         16746800.7

                                 SubEstadoProyecto  \
0                   En ejecuciÃ³n (PGN, Territorio)
1                   En ejecuciÃ³n (PGN, Territorio)
2                   En ejecuciÃ³n (PGN, Territorio)
3  Sin recursos para la vigencia actual (PGN, Ter...
4  Sin recursos para la vigencia actual (PGN, Ter...

   CodigoEntidadResponsable  TotalBeneficiario
0                      5306               6022
1                     15293               2497
2                     15753               1393
3                      5854              17403
4                      5306                300
Conexión exitosa a MongoDB Atlas!
Datos insertados correctamente en MongoDB.
```

Figure 13: Cargue Mongo

6.2 Automatización de Datos (*Tercera entrega*)

En el proceso de automatización de datos, se definió una función que permite gestionar la información relacionada con los proyectos del Departamento Nacional de Planeación (DNP). Esta función está diseñada para procesar los datos en formato JSON, lo cual facilita la integración y manipulación de la información. A medida que los datos sobre los proyectos se van alimentando, la función ejecuta el proceso de Extracción, Transformación y Carga (ETL), garantizando que los datos sean correctamente obtenidos, procesados y almacenados. La extracción se realiza directamente desde una API, donde se recupera la información en tiempo real. Luego, durante la fase de transformación, se aplican diversas operaciones de limpieza y reestructuración de los datos, tales como renombrar columnas, manejar valores nulos y asegurar la coherencia de la información. Finalmente, los datos transformados son cargados en una base de datos MongoDB para su posterior análisis o uso.

Este enfoque automatizado es especialmente valioso, ya que permite manejar grandes volúmenes de datos de manera eficiente, ahorrando tiempo y reduciendo errores humanos. La capacidad de gestionar el flujo de datos en tiempo real y asegurar que la información del DNP esté siempre actualizada facilita la toma de decisiones y el seguimiento adecuado de los proyectos, contribuyendo a la mejora continua en la gestión pública.

6.3 Integración de Datos (*Tercera entrega*)

La integración de datos en esta parte del proyecto implica el proceso de recolectar datos desde una fuente externa (API), transformarlos para asegurar su calidad y consistencia, y luego almacenarlos en una base de datos MongoDB. Este flujo de trabajo automatizado permite gestionar grandes volúmenes de datos de manera eficiente, manteniendo la información del Departamento Nacional de Planeación (DNP) actualizada y lista para su análisis o uso en decisiones institucionales.

7 Proximos pasos (*Tercera entrega*)

En una segunda etapa, se propone gestionar el gran volumen de datos utilizando técnicas avanzadas y modelos de tipo Big Data. La automatización desempeñará un papel fundamental, ya que al integrar diversas fuentes de datos (a nivel distrital, municipal e institucional), existe el riesgo de duplicación de información, lo cual podría afectar el rendimiento y la calidad de los análisis.

Contar con un conocimiento más profundo sobre los proyectos del DNP y colaborar con expertos en la materia resulta esencial para comprender la estructura y la importancia de los datos. Por esta razón, los participantes actuales de este proyecto no abordarán esta fase en detalle; sin embargo, participar en esta siguiente etapa representa una valiosa oportunidad para ampliar el alcance y la eficacia del modelo desarrollado.

8 Lecciones aprendidas (*Tercera entrega*)

Se aplicaron las habilidades desarrolladas en clase; sin embargo, se vieron oportunidades de mejora al implementar la base de datos SQL en Oracle al tener más de 20 fuentes de datos. La normalización y estructura del modelo entidad-relación tomaron mas tiempo del estimado por lo que al final tuvo que simplificarse el ejercicio para cumplir con los entregables.

Las bases de datos No-SQL son mas sencillas de usar y escalables; sin embargo es importante seguir profundizando en su uso, ya que la mayor parte el semestre se centró en SQL.

La automatización fue un desafio que permitió aprender a integrar diferentes fuentes de datos y manipular grandes volumenes de información de manera eficiente con el objetivo de mantener la información actualizada en menor tiempo.

9 Bibliografía

- Capacho, J. R., Nieto Bernal, W. (2017). Diseño de bases de datos. Universidad del Norte.
- Charito. (2018, agosto 3). Modelos de datos NoSQL. EAMinds. https://eaminds.com/2018/08/03/modelando-nosql
- Meier, A., Kaufmann, M. (2019). SQL NoSQL Databases: Models, languages, consistency options and architectures for Big Data management. Springer.
- ¿Qué es una base de datos NoSQL? (2024, septiembre 3). Ibm.com. https://www.ibm.com/es-es/topics/nosql-databa