

# **Estimación de indicadores de calidad del suelo mediante modelos de regresión de machine learning a partir de análisis de imágenes multiespectrales de suelos en Cundinamarca**

**Estudiante:**

Maria Alejandra Enriquez Serrano

**Docente:** Wilmer Mesias Lopez Lopez

Bases de Datos

2023-2

# Contextualización



## *Retos en Agricultura*

Productividad  
agrícola

Sostenibilidad  
ambiental

Degradación  
del suelo



# Problemática

## Agricultura tradicional (Empírica)



*Análisis de calidad del suelo*

Laboratorios especializados



Costosos

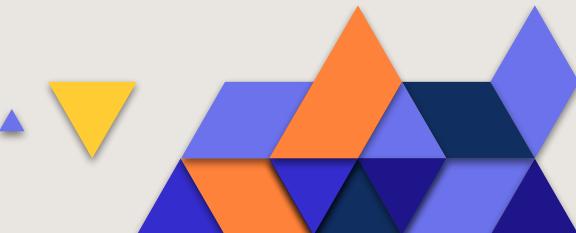
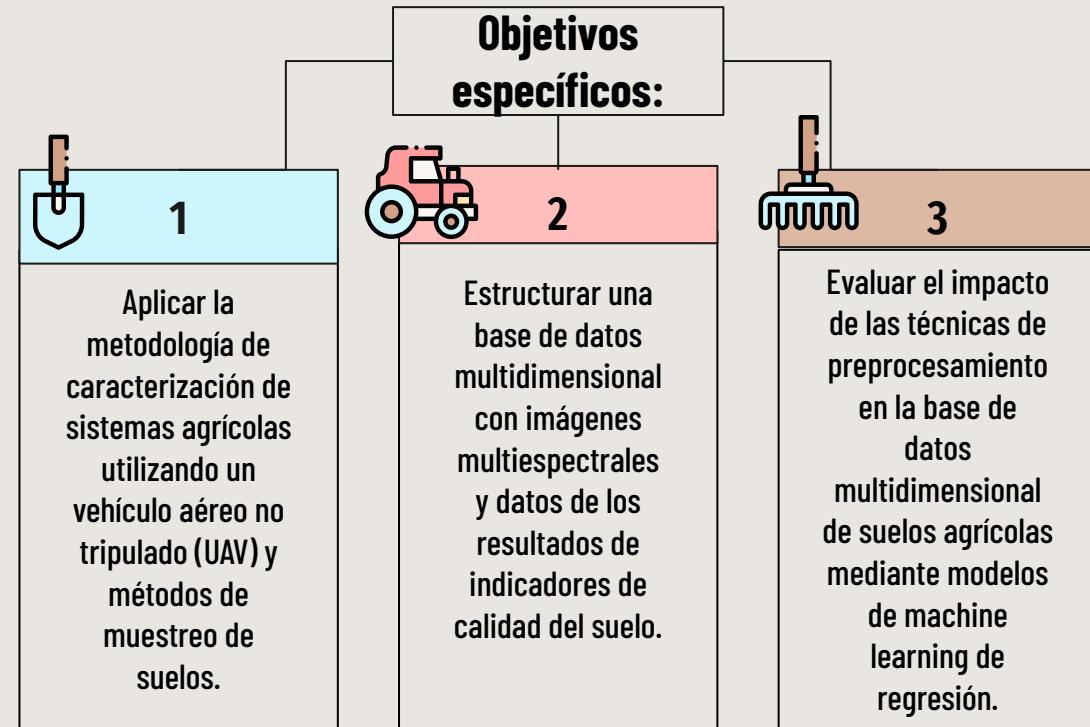


Demorados



# Objetivo general

Determinar indicadores de calidad del suelo mediante modelos de regresión de machine learning a partir de análisis de imágenes multiespectrales de suelos en Cundinamarca.



# Alcance



Diseñar y desarrollar una base de datos robusta y eficiente para almacenar y gestionar datos relacionados con imágenes multiespectrales de suelos y los resultados de laboratorio de los indicadores de calidad del suelo en la región de Cundinamarca

## Pregunta de investigación

¿Cómo determinar indicadores de calidad del suelo por medio de imágenes multiespectrales y modelos de machine learning?



# Origen de la base de datos **EXTRACT**

---





# Desarrollo participativo de una plataforma tecnológica de teledetección para la gestión sostenible de suelos (GSS) en agroecosistemas del Departamento de Cundinamarca

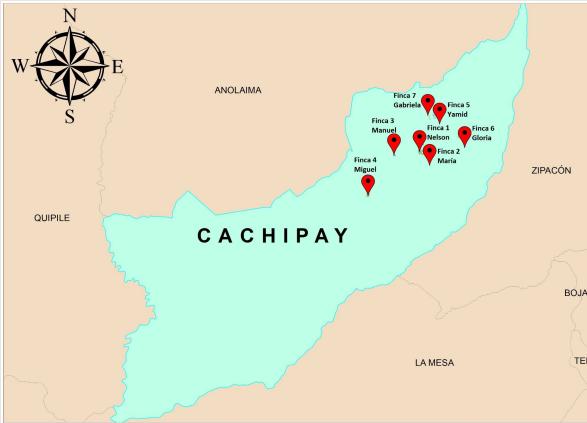
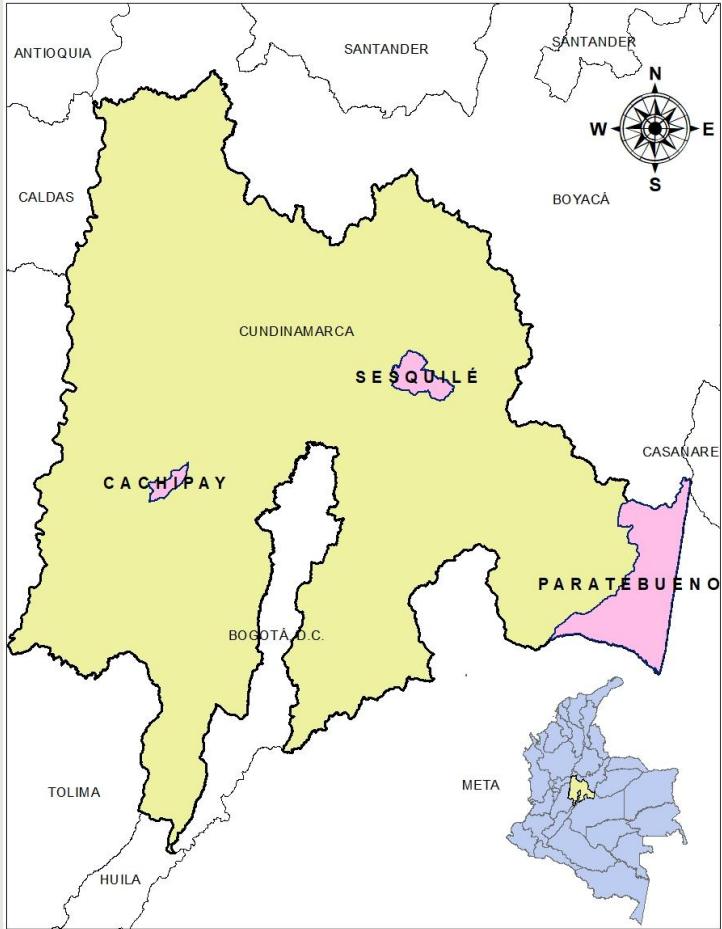
BPIN- 2021000100243

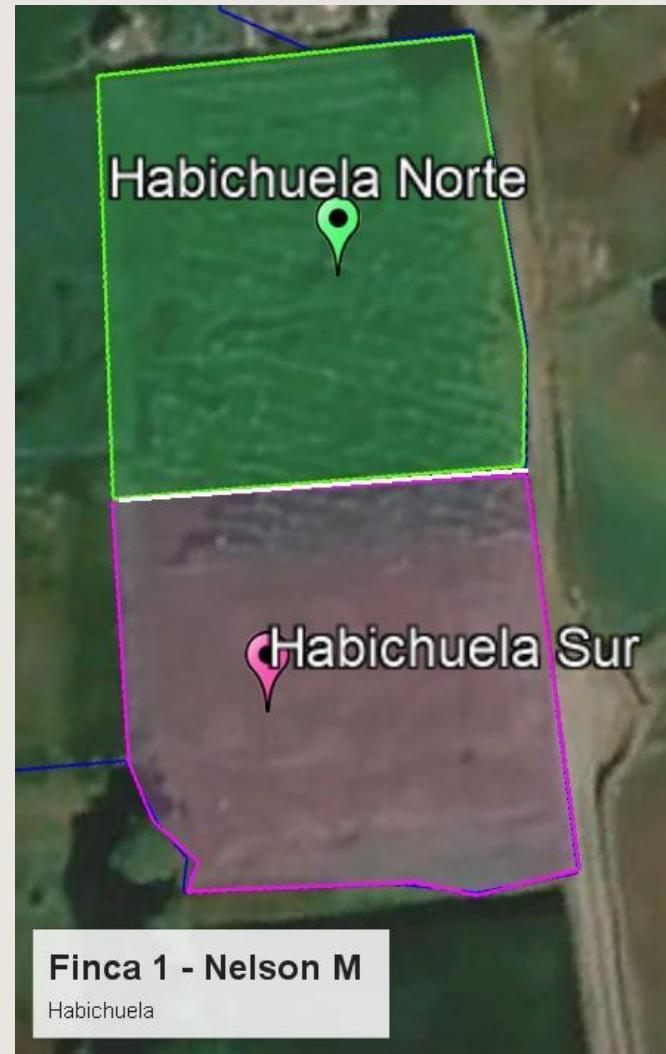


Puertas abiertas a la excelencia



# Zonas de estudio





# Trabajo de campo



Toma de muestras de suelo  
**RESULTADO:** Indicadores de calidad del suelo

# Trabajo de campo



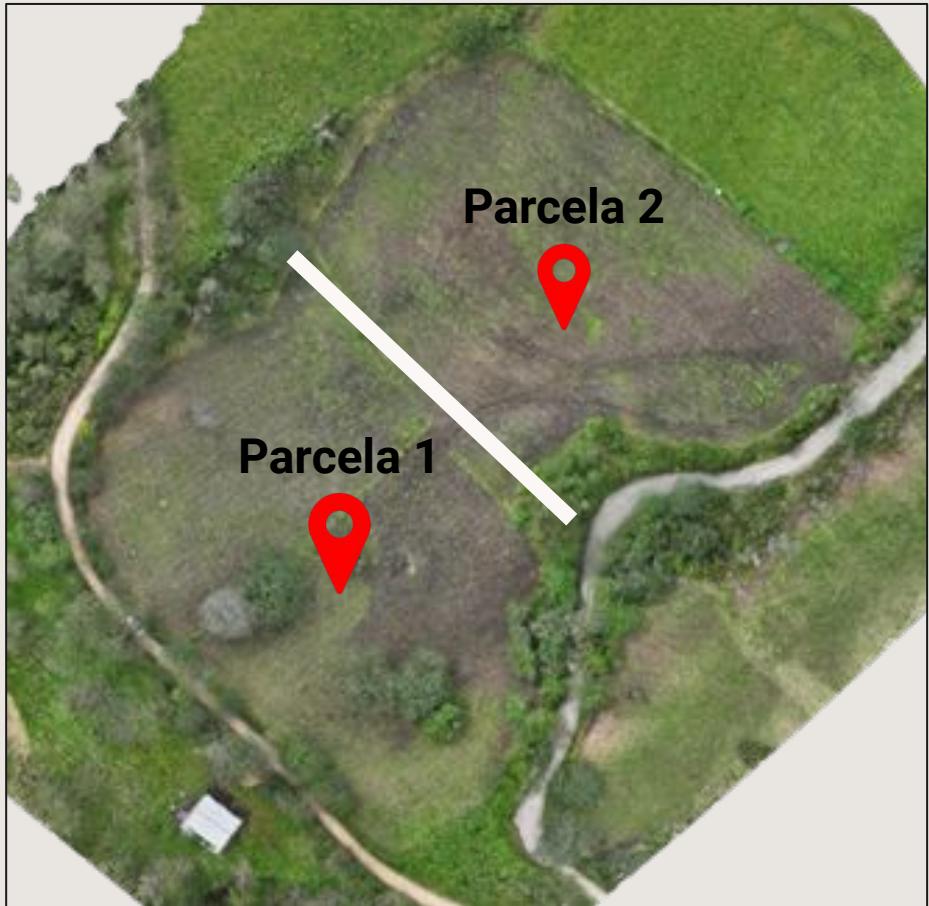
**RESULTADO:** Imágenes multiespectrales

Alistamiento de drone, diseño de rutas de vuelo y sobrevuelo de las zonas de cultivo



# Trabajo de campo

Generación de ortomosaicos y  
extracción de valores de  
reflectancia



# Trabajo de campo



Paratebueno



Cachipay



Sesquilé



Socialización y realización de guia in situ  
**RESULTADO:** Indicadores in situ



# Resultados de muestras de suelo

- Descripción Física de la Muestra
- pH
- Conductividad Eléctrica
- Saturación de Humedad Media
- Capacidad de Intercambio Catiónica Efectiva
- Carbono Orgánico Oxidable
- Materia Orgánica
- Nitrógeno Total
- Densidad Aparente
- Arcilla
- Arena
- Limo
- Textura
- Potasio Intercambiable
- Calcio Intercambiable
- Magnesio Intercambiable
- Sodio Intercambiable

- Acidez Intercambiable
- Hierro
- Cobre
- Zinc
- Boro
- Fósforo
- Azufre
- Nitrógeno Amoniacal
- Nitrógeno Nitrico
- Saturación de Magnesio
- Saturación de Sodio
- Saturación de Aluminio
- Saturación de Potasio
- Saturación de Calcio
- Relación Calcio/Magnesio
- Relación Calcio/Potasio
- Relación Magnesio/Potasio
- Relación (Ca+Mg)/K



# Indicadores In situ

- Estructura y consistencia
- Porosidad
- Color
- Cantidad de lombrices
- Cantidad de organismos
- Resistencia al rompimiento por horizonte
- Estabilidad de agregados
- Materia orgánica
- Capas endurecidas
- Resistencia a la penetración
- Crecimiento de raíces
- Conductividad eléctrica
- Velocidad de infiltración

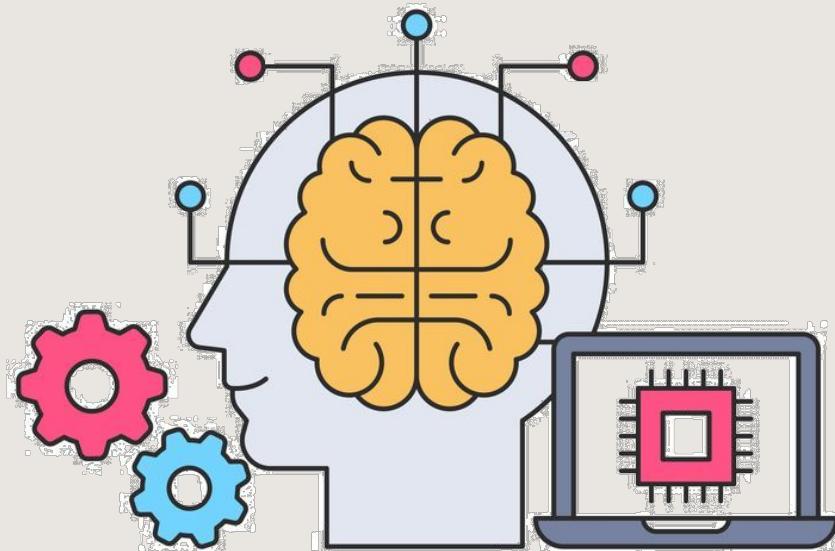
# Reflectancia

- Blue
- Green
- NIR
- Red
- Red EDGE



Transform: eliminación de filas fue necesaria en aquellos casos en que algunas zonas agrícolas carecían de imágenes multiespectrales.





# Diseño del Modelo de Datos del SMBD

---

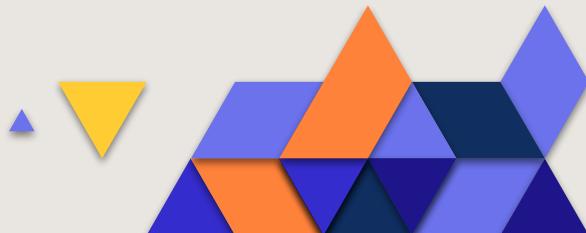


# Características del SMBD

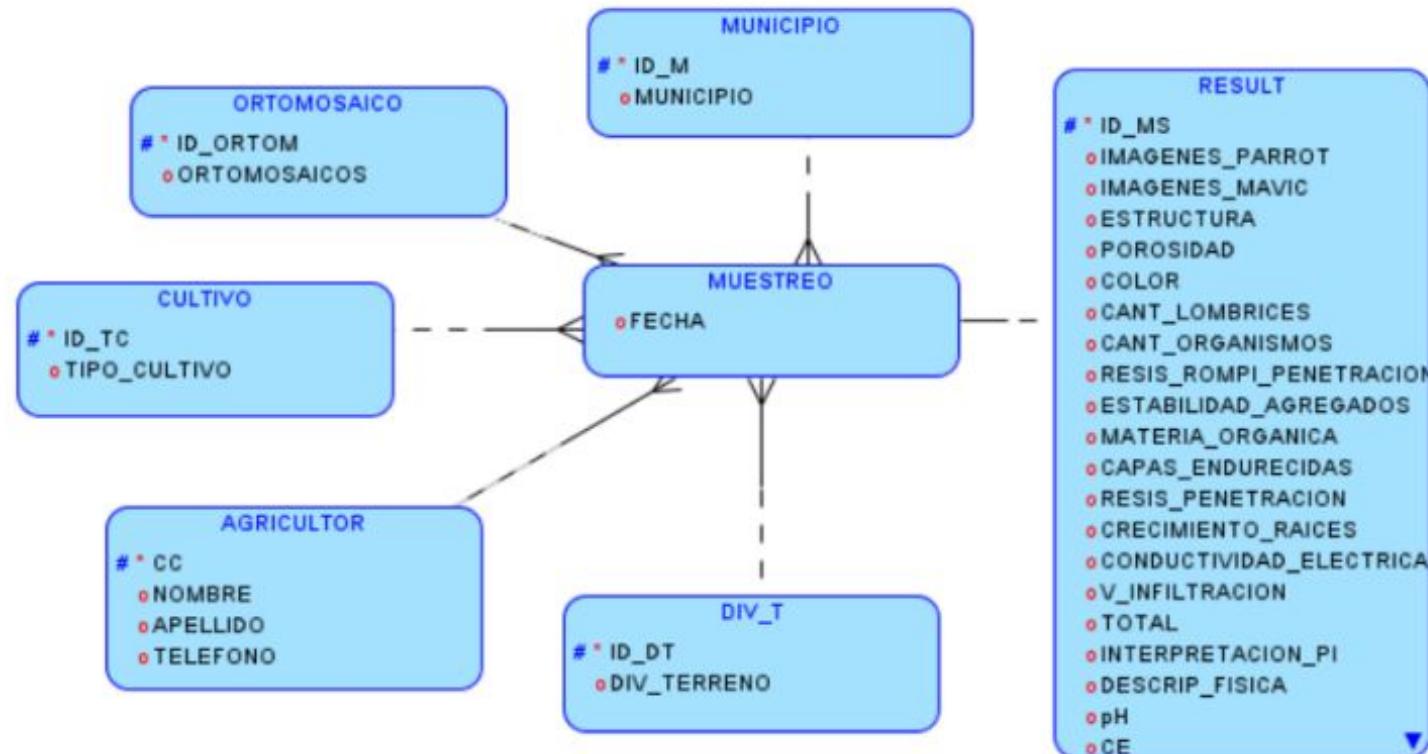


**Modelado de Datos  
Lenguaje de Consulta  
Procedimientos Almacenados  
Seguridad**

**Load: La fase de carga**



# Diagrama modelo de datos



# Lenguaje de definición de datos - DDL

```
CREATE TABLE agricultor (
    cc      INTEGER NOT NULL,
    nombre  VARCHAR2(50),
    apellido VARCHAR2(50),
    telefono INTEGER
);

ALTER TABLE agricultor ADD CONSTRAINT agricultor_pk PRIMARY KEY ( cc );

CREATE TABLE cultivo (
    id_tc      INTEGER NOT NULL,
    tipo_cultivo VARCHAR2(50)
);

ALTER TABLE cultivo ADD CONSTRAINT cultivo_pk PRIMARY KEY ( id_tc );

CREATE TABLE div_t (
    id_dt      INTEGER NOT NULL,
    div_terreno VARCHAR2(50)
);

ALTER TABLE div_t ADD CONSTRAINT div_t_pk PRIMARY KEY ( id_dt );
```

**Creación de las tablas: Agricultor, Cultivo, Div t, Muestreo, Municipio, Ortomosaico y Resultados**

```
CREATE TABLE result (
    id_ms          INTEGER NOT NULL,
    imagenes_parrot VARCHAR2(350),
    imagenes_mavic VARCHAR2(350),
    estructura      FLOAT,
    porosidad       FLOAT,
    color           FLOAT,
    cant_lombrices FLOAT,
    cant_organismos FLOAT,
    resis_rompi_penetracion FLOAT,
    estabilidad_agregados FLOAT,
    materia_organica FLOAT,
    capas_endurecidas FLOAT,
    resis_penetracion FLOAT,
);

CREATE TABLE muestreo (
    fecha          DATE,
    municipio_id_m INTEGER NOT NULL,
    ortomosaico_id_ortom INTEGER NOT NULL,
    cultivo_id_tc  INTEGER NOT NULL,
    agricultor_cc  INTEGER NOT NULL,
    div_t_id_dt    INTEGER NOT NULL,
    result_id_ms   INTEGER NOT NULL
);

CREATE TABLE municipio (
    id_m      INTEGER NOT NULL,
    municipio VARCHAR2(50)
);

ALTER TABLE municipio ADD CONSTRAINT municipio_pk PRIMARY KEY ( id_m );

CREATE TABLE ortomosaico (
    id_ortom     INTEGER NOT NULL,
    ortomosaicos VARCHAR2(350)
);

ALTER TABLE ortomosaico ADD CONSTRAINT ortomosaico_pk PRIMARY KEY ( id_ortom );
```

# Lenguaje de definición de datos - DDL

```
ALTER TABLE result ADD CONSTRAINT result_pk PRIMARY KEY ( id_ms );

ALTER TABLE muestreo
    ADD CONSTRAINT muestreo_agricultor_fk FOREIGN KEY ( agricultor_cc )
        REFERENCES agricultor ( cc );

ALTER TABLE muestreo
    ADD CONSTRAINT muestreo_cultivo_fk FOREIGN KEY ( cultivo_id_tc )
        REFERENCES cultivo ( id_tc );

ALTER TABLE muestreo
    ADD CONSTRAINT muestreo_div_t_fk FOREIGN KEY ( div_t_id_dt )
        REFERENCES div_t ( id_dt );

ALTER TABLE muestreo
    ADD CONSTRAINT muestreo_municipio_fk FOREIGN KEY ( municipio_id_m )
        REFERENCES municipio ( id_m );

ALTER TABLE muestreo
    ADD CONSTRAINT muestreo_ortomosaico_fk FOREIGN KEY ( ortomosaico_id_ortom )
        REFERENCES ortomosaico ( id_ortom );

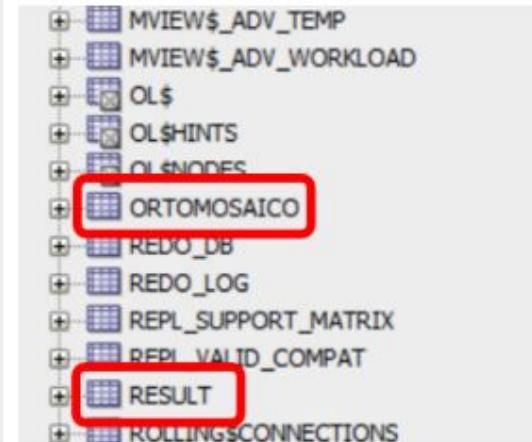
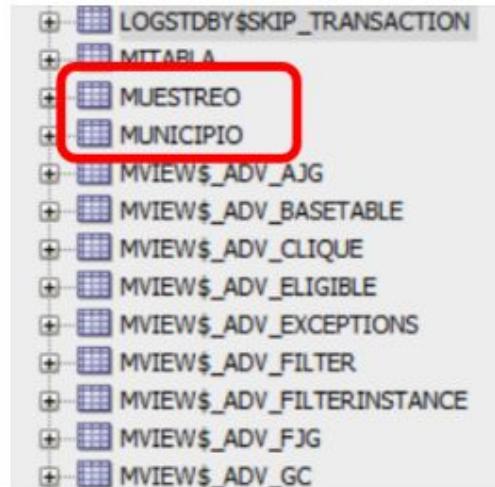
ALTER TABLE muestreo
    ADD CONSTRAINT muestreo_result_fk FOREIGN KEY ( result_id_ms )
        REFERENCES result ( id_ms );
```

**Creación de las relaciones entre cada tabla.**



# Lenguaje de definición de datos - DDL

## Resultado obtenido al momento de crear las tablas



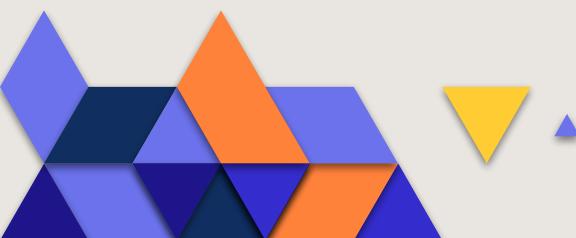
# Manipulación de datos (DML)

```
--Fila 1
INSERT INTO MUESTREO (FECHA, MUNICIPIO_ID_M, ORTOMOSAICO_ID_ORTOM, CULTIVO_ID_TC, AGRICULTOR_CC, DIV_T_ID_DT, RESULT_ID_MS) VALUES (to_date('08/01/2023', 'MM/DD/YYYY'),1,1,1,69622656,1,1);
--Fila 2
INSERT INTO MUESTREO (FECHA, MUNICIPIO_ID_M, ORTOMOSAICO_ID_ORTOM, CULTIVO_ID_TC, AGRICULTOR_CC, DIV_T_ID_DT, RESULT_ID_MS) VALUES (to_date('08/01/2023', 'MM/DD/YYYY'),1,1,1,69622656,2,2);
--Fila 3
INSERT INTO MUESTREO (FECHA, MUNICIPIO_ID_M, ORTOMOSAICO_ID_ORTOM, CULTIVO_ID_TC, AGRICULTOR_CC, DIV_T_ID_DT, RESULT_ID_MS) VALUES (to_date('04/24/2023', 'MM/DD/YYYY'),1,2,2,69622656,3,3);
--Fila 4
INSERT INTO MUESTREO (FECHA, MUNICIPIO_ID_M, ORTOMOSAICO_ID_ORTOM, CULTIVO_ID_TC, AGRICULTOR_CC, DIV_T_ID_DT, RESULT_ID_MS) VALUES (to_date('04/24/2023', 'MM/DD/YYYY'),1,2,2,69622656,4,4);
--Fila 5
INSERT INTO MUESTREO (FECHA, MUNICIPIO_ID_M, ORTOMOSAICO_ID_ORTOM, CULTIVO_ID_TC, AGRICULTOR_CC, DIV_T_ID_DT, RESULT_ID_MS) VALUES (to_date('06/06/2023', 'MM/DD/YYYY'),1,3,3,69622656,1,5);
--Fila 6
INSERT INTO MUESTREO (FECHA, MUNICIPIO_ID_M, ORTOMOSAICO_ID_ORTOM, CULTIVO_ID_TC, AGRICULTOR_CC, DIV_T_ID_DT, RESULT_ID_MS) VALUES (to_date('06/06/2023', 'MM/DD/YYYY'),1,3,3,69622656,2,6);
--Fila 7
INSERT INTO MUESTREO (FECHA, MUNICIPIO_ID_M, ORTOMOSAICO_ID_ORTOM, CULTIVO_ID_TC, AGRICULTOR_CC, DIV_T_ID_DT, RESULT_ID_MS) VALUES (to_date('03/23/2023', 'MM/DD/YYYY'),1,4,4,43073097,3,7);
--Fila 8
INSERT INTO MUESTREO (FECHA, MUNICIPIO_ID_M, ORTOMOSAICO_ID_ORTOM, CULTIVO_ID_TC, AGRICULTOR_CC, DIV_T_ID_DT, RESULT_ID_MS) VALUES (to_date('03/23/2023', 'MM/DD/YYYY'),1,4,4,43073097,4,8);
--Fila 9
```

Salida de Script | Resultado de la Consulta | Se han recuperado 50 filas en 0,003 segundos

SQL

FECHA	MUNICIPIO_ID_M	ORTOMOSAICO_ID_ORTOM	CULTIVO_ID_TC	AGRICULTOR_CC	DIV_T_ID_DT	RESULT_ID_MS
1 01/08/23	1	1	1	69622656	1	1
2 01/08/23		1	1	69622656	2	2
3 24/04/23		1	2	69622656	3	3
4 24/04/23		1	2	69622656	4	4
5 06/06/23		1	3	69622656	1	5



Agregar nuevos registros con "INSERT" a la tabla MUESTREO

# Manipulación de datos (DML)

```
SELECT * FROM AGRICULTOR;
```

Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 21 en 0,003 seg

	CC	NOMBRE	APELLIDO	TELEFONO
1	69622656	Guillermo	Mahecha	3052811572
2	43073097	Armando	Gaona	3125154497
3	71324643	Manuel	Lopez	3119828825
4	66017300	Orlando	Beltran	3221458880
5	62654864	Maria	Mahecha	3223090312
6	77203826	Dario	Urrego	3125217112

**Recuperar todos los registros y columnas de la tabla AGRICULTOR**

# Vistas

**Vista 1 : Visualiza qué agricultores están cultivando piña**

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Vistas' schema tree is displayed, with the 'VISTAAGRICULTORESPINNA' view highlighted by a red rectangle. To the right, the SQL editor contains a query to create a view named 'VistaAgricultoresConPiña' and a select statement to retrieve all data from it. Below the editor, the 'Salida de Script' and 'Resultado de la Consulta' windows are open, showing the results of the query. The results table has columns 'CC', 'NOMBRE', and 'APELIDO', and contains 8 rows of data.

```
-- VISTAS
CREATE VIEW VistaAgricultoresConPiña AS
SELECT a.cc, a.nombre, a.apellido
FROM agricultor a
INNER JOIN muestreo m ON a.cc = m.agricultor_cc
INNER JOIN cultivo c ON m.cultivo_id_tc = c.id_tc
WHERE c.tipo_cultivo = 'Piña';

SELECT * FROM VistaAgricultoresConPiña;
```

	CC	NOMBRE	APELLIDO
1	77203826	Dario	Urrego
2	77203826	Dario	Urrego
3	77203826	Dario	Urrego
4	77203826	Dario	Urrego
5	47001020	Pedro	Gutierrez
6	47001020	Pedro	Gutierrez
7	47001020	Pedro	Gutierrez
8	47001020	Pedro	Gutierrez

# Vistas

The screenshot shows a database management interface. On the left, a tree view of database objects is displayed under the 'Vistas' node. One object, 'VISTAAGRICULTORESCONCOCULO', is highlighted with a red border. The central pane shows the SQL code for this view:

```
CREATE VIEW VistaAgricultoresConCoculo AS
SELECT a.cc, a.nombre, a.apellido
FROM agricultor a
INNER JOIN muestreo m ON a.cc = m.agricultor_cc
INNER JOIN cultivo c ON m.cultivo_id_tc = c.id_tc
WHERE c.tipo_cultivo = 'Coculo';

SELECT * FROM VistaAgricultoresConCoculo;
```

Below the code, there are tabs for 'Salida de Script' and 'Resultado de la Consulta'. The 'Resultado de la Consulta' tab is active, showing the following data:

	CC	NOMBRE	APELLIDO
1	73515682	Nelson	Mancera
2	73515682	Nelson	Mancera
3	62383365	Yamid	Silva
4	62383365	Yamid	Silva
5	59797382	Maria	Munar
6	59797382	Maria	Munar
7	35744648	Gabriela	Correa

**Vista 2: Visualiza  
qué agricultores  
están cultivando  
Coculo**

# Vistas

- + MVVIEW\_RECOMMENDATIONS
- + MVVIEW\_WORKLOAD
- + PRODUCT\_PRIVS
- + SCHEDULER\_JOB\_ARGS
- + SCHEDULER\_PROGRAM\_ARGS
- + VISTAAGRICULTORESCONBOSQUE **+** **VISTAAGRICULTORESCONBOSQUE**
- + VISTAAGRICULTORESCONCOCULO
- + VISTAAGRICULTORESCONPINIA
- Índices
- Paquetes
- Procedimientos
- Funciones
- Operadores
- Colas
- Tablas de Colas
- Disparadores
- + REGISTRAR\_FECHA\_MUESTREO
- Tipos
- Secuencias
- Vistas Materializadas
- Logs de Vistas Materializadas
- Sinónimos
- Sinónimos Públicos
- Enlaces de Base de Datos
- Enlaces de Base de Datos Pública

```
CREATE VIEW VistaAgricultoresConBosque AS
    SELECT a.cc AS cc_agricultor, a.nombre AS nombre_agricultor, a.apellido AS apellido_agricultor,
           m.fecha AS fecha_cultivo, c.tipo_cultivo AS tipo_cultivo
    FROM agricultor a
    INNER JOIN muestreo m ON a.cc = m.agricultor_cc
    INNER JOIN cultivo c ON m.cultivo_id_tc = c.id_tc
    WHERE c.tipo_cultivo = 'Bosque';

SELECT * FROM VistaAgricultoresConBosque;
```

Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 14 en 0,032 segundos

	CC_AGRICULTOR	NOMBRE_AGRICULTOR	APELLIDO_AGRICULTOR	FECHA_CULTIVO	TIPO_CULTIVO
1	69622656	Guillermo	Mahecha	06/06/23	Bosque
2	69622656	Guillermo	Mahecha	06/06/23	Bosque
3	47001020	Pedro	Gutierrez	18/02/23	Bosque
4	47001020	Pedro	Gutierrez	18/02/23	Bosque
5	73515682	Nelson	Mancera	20/02/23	Bosque
6	73515682	Nelson	Mancera	20/02/23	Bosque
7	62383365	Yamid	Silva	16/07/23	Bosque
8	62383365	Yamid	Silva	16/07/23	Bosque
9	59511808	Manuel	Rodriguez	10/08/23	Bosque

**Vista 3: Visualiza qué agricultores que tienen Bosque**

# Triggers

```
CREATE OR REPLACE TRIGGER registrar_fecha_muestreo
BEFORE INSERT ON muestreo
FOR EACH ROW
BEGIN
    :NEW.fecha := SYSTIMESTAMP;
END;
```

Este trigger garantiza que cada vez que se realice una inserción en la tabla MUESTREO, el campo fecha en la nueva fila se establecerá automáticamente en la fecha y hora actual del sistema



# Triggers

```
SELECT * FROM MUESTREO;

INSERT INTO muestreo (fecha, municipio_id_m, ortomosaico_id_ortom, cultivo_id_tc, agricultor_cc, div_t_id_dt, result_id_ms)
VALUES (TO_DATE('2052-04-12', 'YYYY-MM-DD'), 1, 3, 3, 12345, 4, 5);

SELECT * FROM MUESTREO;
```

Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 127 en 0,011 segundos

FECHA	MUNICIPIO_ID_M	ORTOMOSAICO_ID_ORTOM	CULTIVO_ID_TC	AGRICULTOR_CC	DIV_T_ID_DT	RESULT_ID_MS
124 30/07/23	3	62	16	50838356	4	124
125 22/03/23	3	63	16	50838356	1	125
126 22/03/23	3	63	16	50838356	2	126
127 03/10/23	1	3	3	12345	4	5

Compilador - Log

Mensajes Pagina de Registro Sintaxis Opciones Comprobador

| Línea 587 Columna 1 | Insertar | Modificado | Windows: C



Búsqueda



ESP  
LAA



10:40 p.m.  
3/10/2023



# Funciones

```
CREATE OR REPLACE FUNCTION calcular_PromedioMo(ac_c IN INTEGER, ctc IN INTEGER)
RETURN VARCHAR2
IS
    promedio_mo NUMBER;
    mensaje VARCHAR2(200);
    agricultor_nombre VARCHAR2(50);
    cultivo_tipo VARCHAR2(50);
BEGIN
    SELECT a.nombre INTO agricultor_nombre
    FROM agricultor a
    WHERE a.cc = ac_c;
    SELECT c.tipo_cultivo INTO cultivo_tipo
    FROM cultivo c
    WHERE c.id_tc = ctc;
    SELECT AVG(mo) INTO promedio_mo
    FROM result r
    INNER JOIN muestreo m ON r.id_ms = m.result_id_ms
    WHERE m.agricultor_cc = ac_c
    AND m.cultivo_id_tc = ctc;
    IF promedio_mo IS NULL THEN
        mensaje := 'El cultivo "' || cultivo_tipo || '" no pertenece al agricultor "' || agricultor_nombre || '"';
    ELSE
        mensaje := 'El promedio de MO para el agricultor ' || agricultor_nombre || ' del cultivo ' || cultivo_tipo || ' es ' || TO_CHAR(promedio_mo);
    END IF;
    RETURN mensaje;
END calcular_PromedioMo;
```

**Función 1: calcular el promedio de Materia orgánica (mo) de la tabla result para un agricultor específico (ac\_c) y un tipo de cultivo específico (ctc).**



# Funciones

```
SELECT calcular_PromedioMo(69622656, 2) AS promedio_Materia_onica FROM dual;
```

lida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 1 en 0,161 segundos

PROMEDIO\_MATERIA\_ONICA

1 El promedio de MO para el agricultor Guillermo del cultivo Pastizal es 13,6

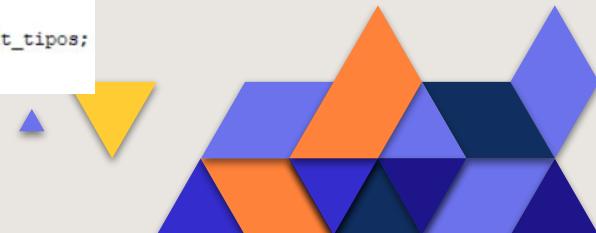
**Resultados de la función para calcular el promedio  
de Materia orgánica**



# Funciones

```
CREATE OR REPLACE FUNCTION contarTiposCultivo(cedula_agri IN INTEGER)
RETURN VARCHAR2
IS
    cult_tipos VARCHAR2(4000);
    cantidad NUMBER;
    agri_nombre VARCHAR2(50);
BEGIN
    SELECT a.nombre INTO agri_nombre
    FROM agricultor a
    WHERE a.cc = cedula_agri;
    SELECT COUNT(DISTINCT c.tipo_cultivo) INTO cantidad
    FROM cultivo c
    WHERE c.id_tc IN (
        SELECT DISTINCT m.cultivo_id_tc
        FROM muestreo m
        WHERE m.agricultor_cc = cedula_agri );
    IF cantidad = 0 THEN
        RETURN 'El agricultor no tiene tipos de cultivo registrados.';
    ELSE
        SELECT LISTAGG(c.tipo_cultivo, ', ') WITHIN GROUP (ORDER BY c.tipo_cultivo)
        INTO cult_tipos
        FROM cultivo c
        WHERE c.id_tc IN (
            SELECT DISTINCT m.cultivo_id_tc
            FROM muestreo m
            WHERE m.agricultor_cc = cedula_agri);
        RETURN 'El agricultor ' || agri_nombre || ' tiene ' || TO_CHAR(cantidad) || ' tipo(s) de cultivo: ' || cult_tipos;
    END IF;
END contarTiposCultivo;
```

**Función 2:  
determinar y contar  
los distintos tipos  
de cultivo asociados  
a un agricultor  
específico**



# Funciones

```
SELECT contarTiposCultivo(59797382) AS Agricultor_cultivos FROM dual;
```

Vista de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 1 en 0,035 segundos

AGRICULTOR\_CULTIVOS

1 El agricultor Maria tiene 3 tipo(s) de cultivo: Aguacate, Cocolo, Rusco

**Función 2:  
Resultados de la  
función para  
determinar y contar  
los distintos tipos  
de cultivo asociados  
a un agricultor**



# Procedimientos almacenados

```
CREATE OR REPLACE PROCEDURE insertar_agricultor(
    p_cc INTEGER,
    p_nombre VARCHAR2,
    p_apellido VARCHAR2,
    p_telefono INTEGER
)
IS
BEGIN
    INSERT INTO agricultor(cc, nombre, apellido, telefono)
    VALUES (p_cc, p_nombre, p_apellido, p_telefono);

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Registro de agricultor insertado correctamente.');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error al insertar el registro de agricultor: ' || SQLERRM);
        ROLLBACK;
END insertar_agricultor;
```

**Procedimiento 1:  
Procedimiento para  
agregar nuevos  
registros a la tabla  
AGRICULTOR**



```
BEGIN
    insertar_agricultor(12345, 'Juan', 'Pérez', 5551234567);
END;

SELECT * FROM AGRICULTOR;
```

The screenshot shows a MySQL Workbench interface. At the top, there is a code editor window containing the provided SQL code. Below it is a results window titled "Resultado de la Consulta" which displays the output of the SELECT query. The results show 22 rows from the AGRICULTOR table, with the last row (ID 22) highlighted by a red border. The columns are labeled CC, NOMBRE, APELLIDO, and TELEFONO.

CC	NOMBRE	APELLIDO	TELEFONO	
6	77203826	Dario	Urrego	3125217112
7	47001020	Pedro	Gutierrez	3017090312
8	73515682	Nelson	Mancera	3163532065
9	55953920	Miguel	Castro	3213322520
10	62383365	Yamid	Silva	3122222520
11	59797382	Maria	Munar	3211260726
12	59511808	Manuel	Rodriguez	3118800137
13	39449546	Gloria	Esperanza	3118549345
14	35744648	Gabriela	Correa	3151617451
15	57020603	Julio	Rozo	3120824107
16	43183578	Miguel	Rozo	3121988472
17	67057502	Guilberto	Gutierrez	3120179606
18	75756144	Hector	Sarmiento	3118755489
19	45959472	Marlen	Rodriguez	3120972619
20	68970773	Rosa	Maldonado	3119401166
21	50838356	Fabio	Rodriguez	3122595310
22	12345	Juan	Pérez	5551234567

# Procedimientos almacenados

**Procedimiento 1:  
Resultados del  
procedimiento para  
agregar nuevos registros  
a la tabla AGRICULTOR**

# Procedimientos almacenados

```
CREATE OR REPLACE PROCEDURE calcular_resultado_muestreo (
    p_result_id_ms INTEGER
) AS
    v_total FLOAT;
BEGIN
    -- Calcular el total (suma de varios campos) según tus necesidades
    SELECT (estructura + porosidad + color + cant_lombrices) INTO v_total
    FROM result
    WHERE id_ms = p_result_id_ms;
    -- Actualizar el campo 'total' en la tabla 'result' con el valor calculado
    UPDATE result
    SET Prueba_Procedimiento = v_total
    WHERE id_ms = p_result_id_ms;
    -- Otros cálculos o actualizaciones que deseas realizar aqui
    COMMIT;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('No se encontró un registro con el ID especificado.');
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error al calcular y actualizar el resultado: ' || SQLERRM);
        ROLLBACK;
END calcular_resultado_muestreo;
```

## Procedimiento 2: Procedimiento para calcular suma de algunas variables

# Procedimientos almacenados

## Visualizar tabla RESULT

```
SELECT * FROM RESULT;
```

Salida de Script x Resultado de la Consulta x

SQL | Se han recuperado 50 filas en 0,006 segundos

N_AMONIACAL	N_NITRICO	SATURACION_MG	SATURACION_SODIO	SATURACION_ALUMINIO	SATURACION_K	SATURACION_CA	RELACION_CA_MG	RELACION_CA_K	RELACION_MG_K	RELACION_CAMG_K	
1	308	24,1	332	18,6	0,612	52,1	7,63	73,1	3,93	9,57	2,43
2	428	23,8	406	18,7	0,666	39,1	7,07	73,6	3,94	10,4	2,64
3	14,6	22,1	63,5	19,1	0,461	43,5	3,36	77	4,03	22,9	5,69

## Crear nueva columna en la tabla RESULT

```
SELECT * FROM RESULT;
```

Salida de Script x Resultado de la Consulta x

SQL | Se han recuperado 50 filas en 0,02 segundos

SATURACION_MG	SATURACION_SODIO	SATURACION_ALUMINIO	SATURACION_K	SATURACION_CA	RELACION_CA_MG	RELACION_CA_K	RELACION_MG_K	RELACION_CAMG_K	PRUEBA_PROCEDIMIENTO	
1	332	18,6	0,612	52,1	7,63	73,1	3,93	9,57	2,43	(null)
2	406	18,7	0,666	39,1	7,07	73,6	3,94	10,4	2,64	(null)

# Procedimientos almacenados

```
BEGIN  
    calcular_resultado_muestreo(1);  
END;  
  
SELECT * FROM RESULT;
```

Salida de Script x Resultado de la Consulta x

SQL | Se han recuperado 50 filas en 0,005 segundos

	SATURACION_MG	SATURACION_SODIO	SATURACION_ALUMINIO	SATURACION_K	SATURACION_CA	RELACION_CA_MG	RELACION_CA_K	RELACION_MG_K	RELACION_CAGM_K	PRUEBA_PROCEDIMIENTO
1	11	332	18,6	0,612	52,1	7,63	73,1	3,93	9,57	2,43
2	3	406	18,7	0,666	39,1	7,07	73,6	3,94	10,4	2,64
3	1	63,5	19,1	0,461	43,5	3,36	77	4,03	22,9	5,69
4	4	78	18,9	0,38	20,2	2,96	77,7	4,12	26,2	6,36
										16
										(null)
										(null)
										(null)

**Resultado del procedimiento para calcular suma de algunas variables**

**(Estructura + Porosidad + Color + Cant lombrices)**

# Procedimientos almacenados

	ESTRUCTURA	POROSIDAD	COLOR	CANT_LOMBRICES
1	3	3	4	6
2	3	3	3	3
3	6	3	2	6
4	0	3	2	3
5	3	6	4	4, 5

**Resultado del procedimiento para calcular suma de algunas variables**

**(Estructura + Porosidad + Color + Cant lombrices)**

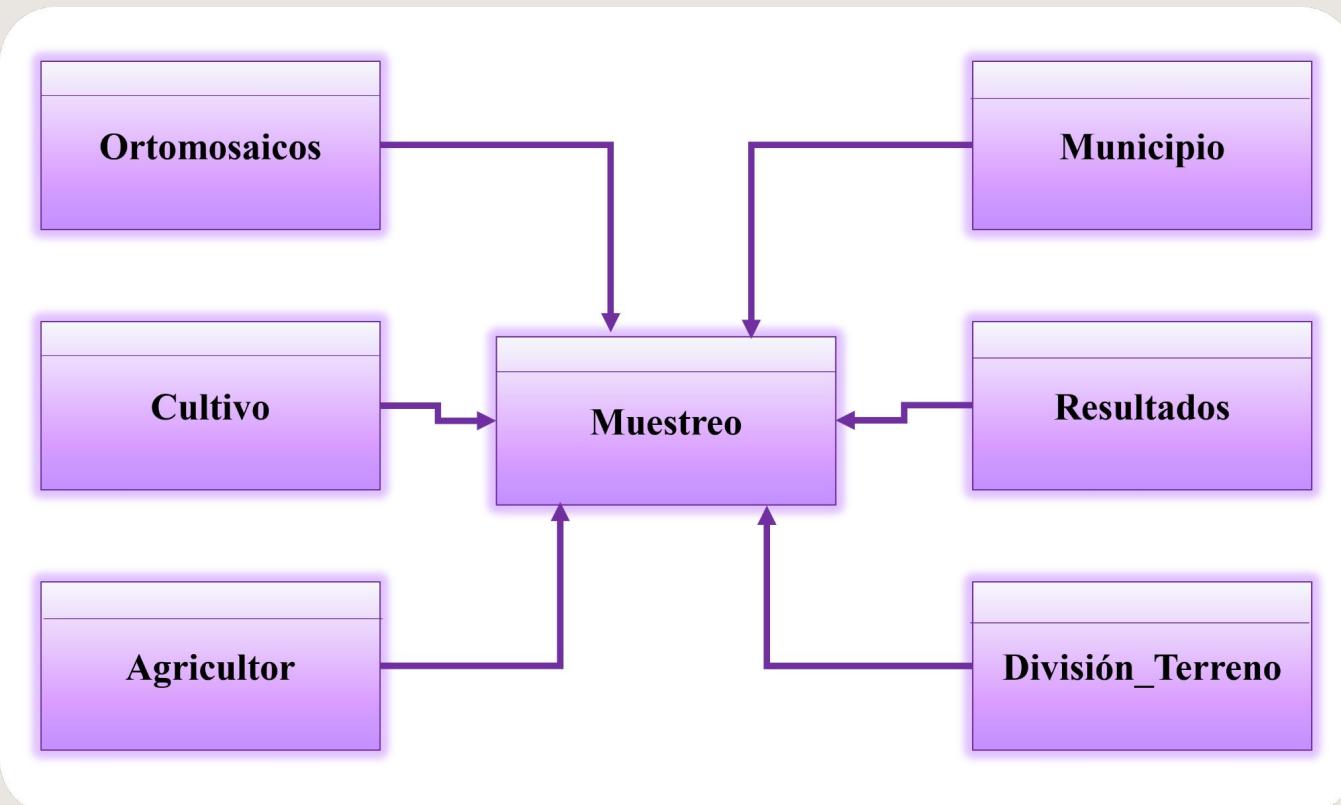


# Bases de Datos No-SQL

---

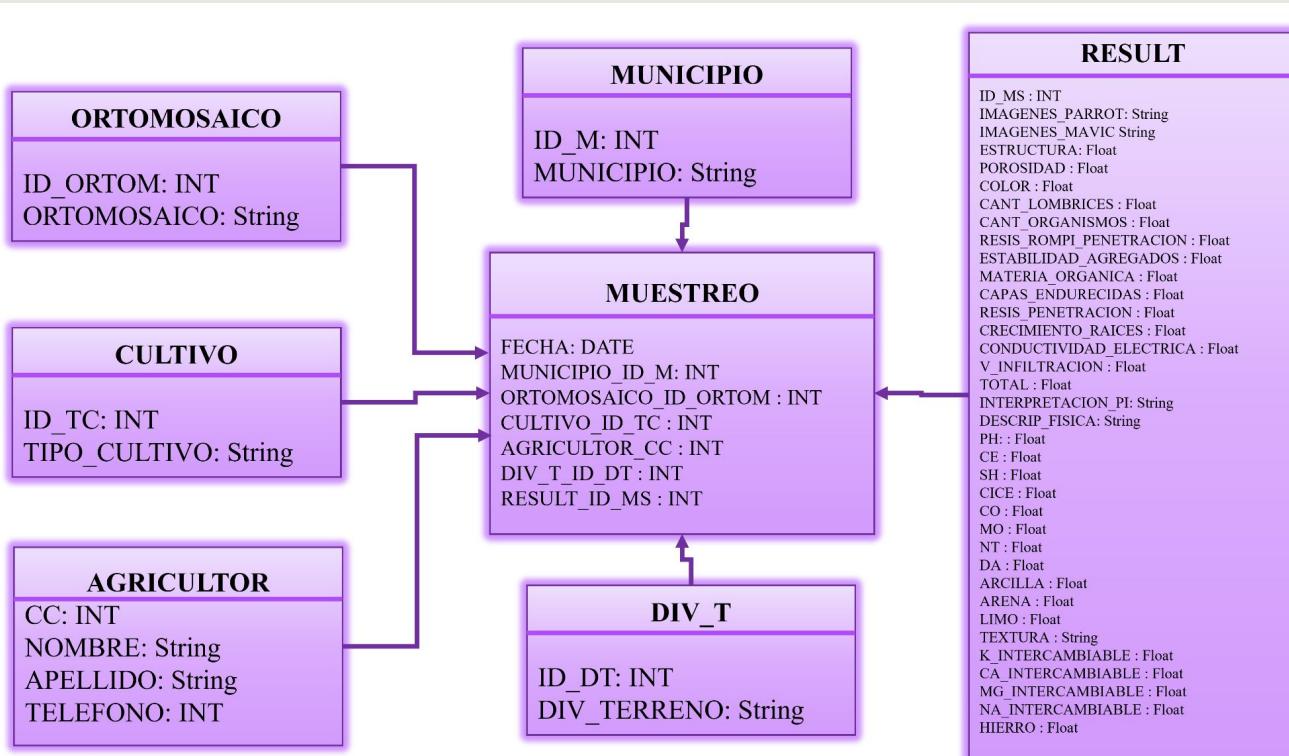


# Diagrama Bases de Datos No-SQL



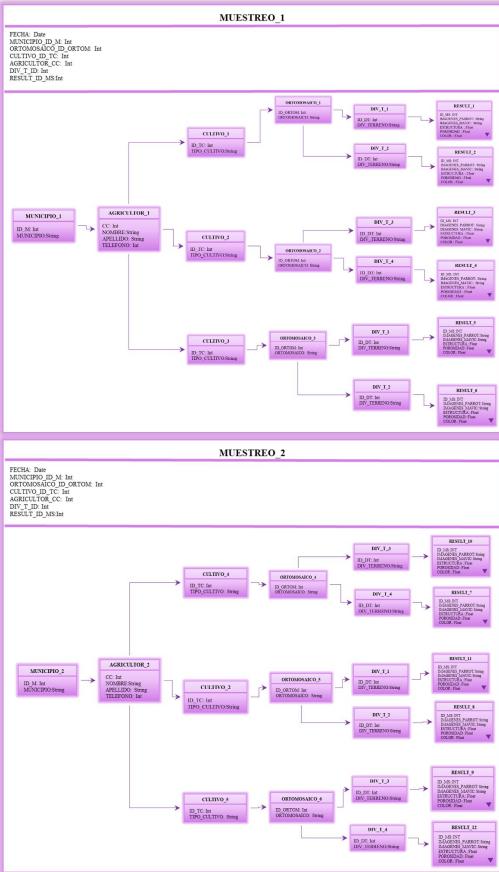
**Meta-modelo  
conceptual**

# Diagrama Bases de Datos No-SQL

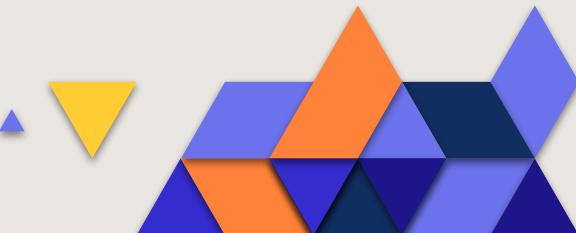


**Meta - Modelo  
Lógico**

# Diagrama Bases de Datos No-SQL



## Meta - Modelo Físico



# SMBD utilizado para la Base de Datos No-SQL

**Modelo de Datos Flexible**

**Basado en documentos JSON llamado BSON.**

**Esta característica posibilita almacenar datos de manera versátil**

**Permitiendo que los documentos puedan contener campos de diferentes tipos sin estar limitados a una estructura fija**



# SMBD utilizado para la Base de Datos No-SQL

## DB\_Multidimensional

AGRICULTOR

CULTIVO

DIV\_T

MUESTREO

MUNICIPIO

ORTOMOSAICO

RESULT

### DB\_Multidimensional.AGRICULTOR

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 2.15KB TOTAL DOCUMENTS: 21 INDEXES: 1

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Filter

Type a query: { field: 'value' }

`_id: ObjectId('6556b29dd48ed0b21af87e10')`  
CC: "69622656"  
NOMBRE: "Guillermo"  
APELIDO: "Mahecha"  
TELEFONO: "3052811572"

`_id: ObjectId('6556b29dd48ed0b21af87e11')`  
CC: "43073097"  
NOMBRE: "Armando"  
APELIDO: "Gaona"  
TELEFONO: "3125154497"

`_id: ObjectId('6556b29dd48ed0b21af87e12')`  
CC: "71324643"  
NOMBRE: "Manuel"  
APELIDO: "Lopez"  
TELEFONO: "3119828825"

### DB\_Multidimensional.CULTIVO

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 971B TOTAL DOCUMENTS: 16 INDEXES: 1

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Filter

Type a query: { field: 'value' }

QUERY RESULTS: 1-16 OF 16

`_id: ObjectId('6556b2dad48ed0b21af87e25')`  
TIPO\_CULTIVO: "Caucho"

`_id: ObjectId('6556b2dad48ed0b21af87e26')`  
TIPO\_CULTIVO: "Pastizal"

`_id: ObjectId('6556b2dad48ed0b21af87e27')`  
TIPO\_CULTIVO: "Bosque"

`_id: ObjectId('6556b2dad48ed0b21af87e28')`  
TIPO\_CULTIVO: "Yuca"

`_id: ObjectId('6556b2dad48ed0b21af87e29')`  
TIPO\_CULTIVO: "Citricos"

# SMBD utilizado para la Base de Datos No-SQL

## DB\_Multidimensional.RESULT

STORAGE SIZE: 64KB LOGICAL DATA SIZE: 186.14KB TOTAL DOCUMENTS: 126 INDEXES TOTAL SIZE: 20KB

[Find](#) [Indexes](#) [Schema Anti-Patterns](#) [Aggregation](#) [Search Indexes](#)

Filter

Type a query: { field: 'value' }

```
_id: ObjectId('6556b588d48ed0b21af87ef9')
IMAGENES_PARROT: "C:\Users\menri\Desktop\Proyecto Bases de datos\Municipios\Paratebueno\..."
IMAGENES_MAVIC: "C:\Users\menri\Desktop\Proyecto Bases de datos\Municipios\Paratebueno\..."
ESTRUCTURA: "3.0"
POROSIDAD: "3.0"
COLOR: "4"
CANT_LOMBRICES: "6.0"
CANT_ORGANISMOS: "6.0"
RESIS_ROMPI_PENETRACION: "4"
ESTABILIDAD_AGREGADOS: "3.0"
MATERIA_ORGANICA: "2"
CAPAS_ENDURECIDAS: "2.0"
RESIS_PENETRACION: "4"
CRECIMIENTO_RAICES: "3.0"
CONDUCTIVIDAD_ELECTRICA: "4"
V_INFILTRACION: "0"
TOTAL: "44.0"
INTERPRETACION_PI: "Moderada Calidad"
DESCRIP_FISICA: "SÓLIDO PARDO OSCURO CON PRESENCIA DE MATERIAL VEGETAL"
PH: "6.02"
CE: "3.08"
SH: "33.8"
```

## DB\_Multidimensional.MUNICIPIO

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 178B TOTAL DOCUMENTS: 3

[Find](#) [Indexes](#) [Schema Anti-Patterns](#) [Agg](#)

Filter

Type a query: { field: 'value' }

QUERY RESULTS: 1-3 OF 3

```
_id: ObjectId('6556b4cad48ed0b21af87ef6')
MUNICIPIO: "Paratebueno"
```

```
_id: ObjectId('6556b4cad48ed0b21af87ef7')
MUNICIPIO: "Cachipay"
```

```
_id: ObjectId('6556b4cad48ed0b21af87ef8')
MUNICIPIO: "Sesquilé"
```

# SMBD utilizado para la Base de Datos No-SQL

## DB\_Multidimensional.ORTOMOSAICO

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 11.25KB TOTAL DOCUMENTS: 63 INDEXES TOTAL SIZE: 20KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes

Filter

Type a query: { field: 'value' }

\_id: ObjectId('6556b463d48ed0b21af87eb7')

ORTOMOSAICOS: "C:\Users\menri\Desktop\Proyecto Bases de datos\Municipios\Paratebueno\..."

\_id: ObjectId('6556b463d48ed0b21af87eb8')

ORTOMOSAICOS: "C:\Users\menri\Desktop\Proyecto Bases de datos\Municipios\Paratebueno\..."

\_id: ObjectId('6556b463d48ed0b21af87eb9')

ORTOMOSAICOS: "C:\Users\menri\Desktop\Proyecto Bases de datos\Municipios\Paratebueno\..."

\_id: ObjectId('6556b463d48ed0b21af87eba')

ORTOMOSAICOS: "C:\Users\menri\Desktop\Proyecto Bases de datos\Municipios\Paratebueno\..."

## DB\_Multidimensional.DIV\_T

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 249B TOTAL DOCUMENTS: 4

Find

Indexes

Schema Anti-Patterns 0

Filter

Type a query: { field: 'value' }

QUERY RESULTS: 1-4 OF 4

\_id: ObjectId('6556b30ed48ed0b21af87e35')

DIV\_TERRENO: "Zona Sur"

\_id: ObjectId('6556b30ed48ed0b21af87e36')

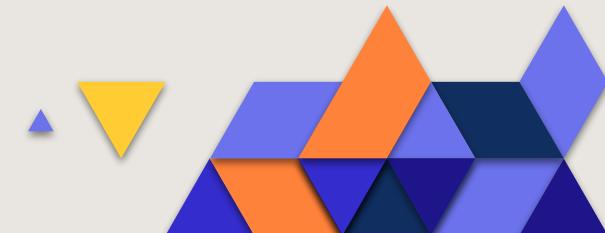
DIV\_TERRENO: "Zona Norte"

\_id: ObjectId('6556b30ed48ed0b21af87e37')

DIV\_TERRENO: "Zona Este"

\_id: ObjectId('6556b30ed48ed0b21af87e38')

DIV\_TERRENO: "Zona Oeste"

- 
- ◀ Barreira, A., Duarte, J., & Guedes, M. (2019). A comprehensive review on UAVs for remote sensing applications in agriculture. *Sensors*, 19(19), 4237. <https://doi.org/10.3390/s19194237>
  - ◀ Cubero-Castan, M., Schneider-Zapp, K., Bellomo, M., Shi, D., Rehak, M., & Strecha, C. (2018). Assessment of the radiometric accuracy in a target less workflow using pix4d software. In 2018 9th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS) (pp. 1-4).
  - ◀ Demir, V., & Kiremitci, A. (2020). Agricultural drone-based remote sensing: A review. *Precision Agriculture*, 21(7), 801-829. <https://doi.org/10.1007/s11119-020-09684-x>
  - ◀ FAO, W., IFAD, UNICEF, & WHO. (2019). The state of food security and nutrition in the world: Safeguarding against economic slowdowns and downturns. Rome Italy: The Food and Agriculture Organization. <http://www.fao.org/3/ca5162en/ca5162en.pdf>
  - ◀ Geisseler, D., & Scow, K. M. (2014). Long-term effects of mineral fertilizers on soil microorganisms - A review. *Soil Biology and Biochemistry*, 75, 54-63. <http://dx.doi.org/10.1016/j.soilbio.2014.03.023>
  - ◀ Gharbi, K., & Bouaziz, N. (2020). Mapping Soil Physical Properties and Soil Organic Carbon Using UAV-Based Remote Sensing Techniques. *Remote Sensing*, 12(4), 759. <https://doi.org/10.3390/rs12040759>
- 

# Referencias

---

# GRACIAS!

