

REPORTE DE PRÁCTICA

NOMBRE DE LA PRÁCTICA: DATA WAREHOUSE

ALUMNO: Axel Damian Ortiz Simon
Dr. Eduardo Cornejo-Velázquez



1. Marco teórico

En esta práctica, se abordaron conceptos clave relacionados con la gestión de bases de datos distribuidas, incluyendo los procesos ETL, el concepto de Data Warehouse, y el uso de comandos como 'SELECT ... INTO FILE', 'LOAD DATA INFILE', y consultas que involucran múltiples nodos. Se presentará el esquema conceptual local de cada nodo junto con los scripts necesarios para la creación, extracción, carga y consulta de datos en un entorno distribuido.

1. Data Warehouse

Un Data Warehouse es un sistema utilizado para el reporte y análisis de datos, que actúa como un repositorio centralizado donde se integran datos de múltiples fuentes. Su principal objetivo es facilitar la toma de decisiones mediante la consolidación de datos históricos y actuales, permitiendo a los usuarios realizar análisis complejos y generar informes detallados.

Los Data Warehouses se caracterizan por su capacidad para manejar grandes volúmenes de datos y por su estructura optimizada para consultas analíticas. A menudo, los datos se extraen de diversas bases de datos operativas a través de un proceso conocido como ETL

2. Procesos ETL (Extract, Transform, Load)

Los procesos ETL son utilizados para extraer datos de una fuente, transformarlos según sea necesario y cargarlos en un destino. En esta práctica, se utilizaron procesos ETL para:

- **Extraer:** Obtención de datos desde diversas fuentes (GCS) usando 'SELECT ... INTO FILE' recopilando datos de diferentes fuentes, que pueden incluir bases de datos relacionales, archivos planos, y sistemas de gestión de contenido.
- **Transformar:** Limpieza y adaptación de los datos para cumplir con los requisitos del sistema destino. Ajustar el formato de los datos para su carga en las bases de datos locales. Esto puede incluir la normalización de formatos, la eliminación de duplicados y la aplicación de reglas de negocio.
- **Cargar:** Inserción de los datos en la base de datos final o en los nodos correspondientes locales usando 'LOAD DATA INFILE', insertando los datos transformados en el Data Warehouse, donde se organizan en un formato que facilita el análisis.

3. SELECT + INTO FILE

El comando 'SELECT ... INTO OUTFILE' permite exportar datos desde una tabla a un archivo en el servidor. Es útil para extraer datos y prepararlos para su carga en otros sistemas.

Sintaxis:

Listing 1: SELECT INTO FILE

```
SELECT * INTO OUTFILE 'ruta/al/archivo.csv'
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
FROM nombre_tabla;
```

Ejemplo:

Listing 2: Exportar datos de Vehículo

```
SELECT * INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/vehiculo.csv'
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n'
FROM sistemaGestionFlotillas.Vehiculo;
```

4. LOAD DATA INFILE

El comando 'LOAD DATA INFILE' permite cargar datos desde un archivo en una tabla de la base de datos. Es útil para importar datos extraídos previamente.

Sintaxis:

Listing 3: LOAD DATA INFILE

```
LOAD DATA INFILE 'ruta/al/archivo.csv'
INTO TABLE nombre_tabla
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Ejemplo:

Listing 4: Cargar datos en Vehiculo

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/vehiculo.csv'
INTO TABLE LCS_Principal.Vehiculo
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

5. Herramientas Utilizadas

En este proyecto se emplearon diversas herramientas que facilitaron el desarrollo y la gestión de datos, cada una con un rol específico que contribuyó al éxito de la práctica.

DataGrip

DataGrip es un entorno de desarrollo integrado (IDE) diseñado específicamente para trabajar con bases de datos. Su interfaz visual permite a los desarrolladores gestionar bases de datos de manera eficiente, facilitando tareas como la creación de esquemas, la ejecución de consultas SQL y la visualización de datos.

Python con Tkinter

Python es un lenguaje de programación versátil y poderoso, ideal para el desarrollo de aplicaciones y la manipulación de datos. En este proyecto, se utilizó junto con Tkinter, una biblioteca estándar de Python para crear interfaces gráficas de usuario (GUI).

LaTeX

LaTeX es un sistema de preparación de documentos que se utiliza ampliamente para la creación de documentos técnicos y científicos. En este proyecto, LaTeX se utilizó para elaborar el reporte final, proporcionando.

6. Esquema Conceptual Local de cada nodo

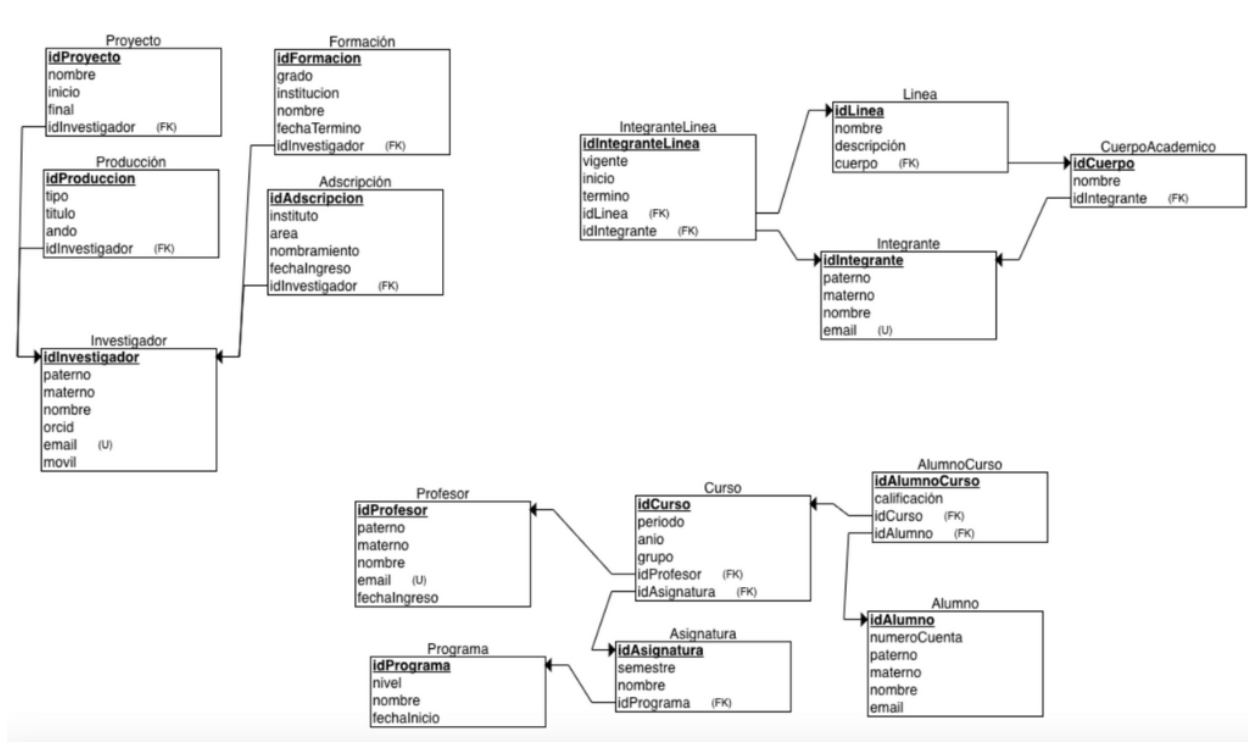


Figure 1: Esquema Entidad Relación.

GCS:

- **Proyecto:** Almacena información sobre los proyectos.
- **Formacion:** Almacena información sobre la formacion.
- **Produccion:** Almacena informacion de la produccion.
- **Adscripcion:** Almacena informacion relacionados con la adscripcion.
- **Investigador:** Almacena registros relacionados con los investigadores.
- **IntegranteLinea:** Almacena información relacionada a IntegranteLinea.
- **Linea:** Almacena informacion de relacionada a Linea.
- **CuerpoAcademico:** Almacena información sobre los el CuerpoAcademico.
- **Integrante:** Almacena registros de los integrantes.
- **Profesor:** Almacena registros de los Profesores.
- **Programa:** Almacena información sobre los distintos programas.

- **Curso:** Almacena rla informacion sobre los cursos.
- **Asignatura:** Almacena información sobre las asignaturas.
- **AlumnoCurso:** Almacena información relacionada a AlumnoCurso.
- **Alumno:** Almacena registros de los Alumnos.

LCS-1:

- **Proyecto:** Almacena información sobre los proyectos.
- **Formacion:** Almacena información sobre la formacion.
- **Produccion:** Almacena informacion de la produccion.
- **Adscripcion:** Almacena informacion relacionados con la adscripcion.
- **Investigador:** Almacena registros relacionados con los investigadores.

LCS-2:

- **IntegranteLinea:** Almacena información relacionada a IntegranteLinea.
- **Linea:** Almacena informacion de relacionada a Linea.
- **CuerpoAcademico:** Almacena información sobre los el CuerpoAcademico.
- **Integrante:** Almacena registros de los integrantes.

LCS-3:

- **Profesor:** Almacena registros de los Profesores.
- **Programa:** Almacena información sobre los distintos programas.
- **Curso:** Almacena rla informacion sobre los cursos.
- **Asignatura:** Almacena información sobre las asignaturas.
- **AlumnoCurso:** Almacena información relacionada a AlumnoCurso.
- **Alumno:** Almacena registros de los Alumnos.

7. Script de creación de nodos

Listing 5: Creacion LCS-1

```
CREATE TABLE Investigador (  
  idInvestigador INT AUTOINCREMENT PRIMARY KEY,  
  paterno VARCHAR(80) NOT NULL,  
  materno VARCHAR(80) NOT NULL,  
  nombre VARCHAR(120) NOT NULL,  
  orcid VARCHAR(30) NOT NULL,  
  email VARCHAR(150) NOT NULL UNIQUE ,  
  movil VARCHAR(15) NOT NULL  
);  
  
CREATE TABLE Proyecto (  
  idProyecto INT AUTOINCREMENT PRIMARY KEY,  
  nombre VARCHAR(250) NOT NULL,  
  inicio DATE NOT NULL,  
  final DATE,  
  idInvestigador INT NOT NULL,  
  FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)  
);  
  
CREATE TABLE Formacion (  
  idFormacion INT AUTOINCREMENT PRIMARY KEY,  
  grado VARCHAR(18) NOT NULL,  
  institucion VARCHAR(70) NOT NULL,  
  nombre VARCHAR(120) NOT NULL,  
  fechaTermino DATE NOT NULL,  
  idInvestigador INT NOT NULL,  
  FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)  
);  
  
CREATE TABLE Produccion (  
  idProduccion INT AUTOINCREMENT PRIMARY KEY,  
  tipo VARCHAR(60) NOT NULL,  
  titulo VARCHAR(230) NOT NULL,  
  anio INT NOT NULL,  
  idInvestigador INT NOT NULL,  
  FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)  
);  
  
CREATE TABLE Adscripcion (  
  idAdscripcion INT AUTOINCREMENT PRIMARY KEY,  
  instituto VARCHAR(30) NOT NULL,  
  area VARCHAR(130) NOT NULL,  
  nombramiento VARCHAR(10) NOT NULL,  
  fechaIngreso DATE NOT NULL,  
  idInvestigador INT NOT NULL,  
  FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)  
);
```

Listing 6: Creacion de LCS-2

```

CREATE TABLE IF NOT EXISTS 'Integrante' (
    'idIntegrante' INT AUTOINCREMENT PRIMARY KEY,
    'paterno' VARCHAR(90) NOT NULL,
    'materno' VARCHAR(90) NOT NULL,
    'nombre' VARCHAR(110) NOT NULL,
    'email' VARCHAR(150) NOT NULL UNIQUE
);

CREATE TABLE IF NOT EXISTS 'CuerpoAcademico' (
    'idCuerpo' INT AUTOINCREMENT PRIMARY KEY,
    'nombre' VARCHAR(230) NOT NULL,
    'idIntegrante' INT NOT NULL,
    FOREIGN KEY ('idIntegrante') REFERENCES 'Integrante' ('idIntegrante')
);

CREATE TABLE IF NOT EXISTS 'Linea' (
    'idLinea' INT AUTOINCREMENT PRIMARY KEY,
    'nombre' VARCHAR(120) NOT NULL,
    'descripcion' VARCHAR(300) NOT NULL,
    'cuerpo' INT NOT NULL,
    FOREIGN KEY ('cuerpo') REFERENCES 'CuerpoAcademico' ('idCuerpo')
);

CREATE TABLE IF NOT EXISTS 'IntegranteLinea' (
    'idIntegranteLinea' INT AUTOINCREMENT PRIMARY KEY,
    'vigente' BOOLEAN NOT NULL,
    'inicio' DATE NOT NULL,
    'termino' DATE,
    'idLinea' INT NOT NULL,
    'idIntegrante' INT NOT NULL,
    FOREIGN KEY ('idLinea') REFERENCES 'Linea' ('idLinea'),
    FOREIGN KEY ('idIntegrante') REFERENCES 'Integrante' ('idIntegrante'),
    CONSTRAINT 'uq_integrante_linea' UNIQUE ('idLinea', 'idIntegrante')
);

```

Listing 7: Creacion de LCS-3

```

CREATE TABLE IF NOT EXISTS 'Profesor' (
    'idProfesor' INT AUTOINCREMENT PRIMARY KEY,
    'paterno' VARCHAR(80) NOT NULL,
    'materno' VARCHAR(80) NOT NULL,
    'nombre' VARCHAR(80) NOT NULL,
    'email' VARCHAR(250) NOT NULL UNIQUE,
    'fechalngreso' DATE NOT NULL
);

CREATE TABLE IF NOT EXISTS 'Programa' (
    'idPrograma' INT AUTOINCREMENT PRIMARY KEY,
    'nivel' VARCHAR(25) NOT NULL,
    'nombre' VARCHAR(130) NOT NULL,
    'fechalnicio' DATE NOT NULL
);

CREATE TABLE IF NOT EXISTS 'Asignatura' (

```

```

        'idAsignatura' INT AUTO_INCREMENT PRIMARY KEY,
        'semestre' INT NOT NULL,
        'nombre' VARCHAR(120) NOT NULL,
        'idPrograma' INT NOT NULL,
        FOREIGN KEY ('idPrograma') REFERENCES 'Programa'('idPrograma')
    ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

CREATE TABLE IF NOT EXISTS 'Alumno' (
    'idAlumno' INT AUTO_INCREMENT PRIMARY KEY,
    'numeroCuenta' VARCHAR(80) NOT NULL UNIQUE,
    'paterno' VARCHAR(80) NOT NULL,
    'materno' VARCHAR(80) NOT NULL,
    'nombre' VARCHAR(80) NOT NULL,
    'email' VARCHAR(120) NOT NULL UNIQUE
);

CREATE TABLE IF NOT EXISTS 'Curso' (
    'idCurso' INT AUTO_INCREMENT PRIMARY KEY,
    'periodo' VARCHAR(15) NOT NULL,
    'anio' INT NOT NULL,
    'grupo' INT NOT NULL,
    'idProfesor' INT NOT NULL,
    'idAsignatura' INT NOT NULL,
    FOREIGN KEY ('idProfesor') REFERENCES 'Profesor'('idProfesor'),
    FOREIGN KEY ('idAsignatura') REFERENCES 'Asignatura'('idAsignatura')
);

CREATE TABLE IF NOT EXISTS 'AlumnoCurso' (
    'idAlumnoCurso' INT AUTO_INCREMENT PRIMARY KEY,
    'calificacion' DECIMAL(3,2) NOT NULL, DECIMAL(3,2)
    'idCurso' INT NOT NULL,
    'idAlumno' INT NOT NULL,
    FOREIGN KEY ('idCurso') REFERENCES 'Curso'('idCurso'),
    FOREIGN KEY ('idAlumno') REFERENCES 'Alumno'('idAlumno'),
    CONSTRAINT 'uq_alumno_curso' UNIQUE ('idCurso', 'idAlumno')
);

```

Listing 8: Creacion de GCS

```

CREATE TABLE IF NOT EXISTS 'Investigador' (
    'idInvestigador' INT AUTO_INCREMENT PRIMARY KEY,
    'paterno' VARCHAR(90) NOT NULL,
    'materno' VARCHAR(90) NOT NULL,
    'nombre' VARCHAR(120) NOT NULL,
    'orcid' VARCHAR(30) NOT NULL,
    'email' VARCHAR(150) NOT NULL UNIQUE,
    'movil' VARCHAR(15) NOT NULL
);

CREATE TABLE IF NOT EXISTS 'Proyecto' (
    'idProyecto' INT AUTO_INCREMENT PRIMARY KEY,
    'nombre' VARCHAR(250) NOT NULL,
    'inicio' DATE NOT NULL,
    'final' DATE,
    'idInvestigador' INT NOT NULL,

```



```

        FOREIGN KEY ('idInvestigador') REFERENCES 'Investigador' ('idInvestigador')
    );

CREATE TABLE IF NOT EXISTS 'Formacion' (
    'idFormacion' INT AUTO_INCREMENT PRIMARY KEY,
    'grado' VARCHAR(18) NOT NULL,
    'institucion' VARCHAR(70) NOT NULL,
    'nombre' VARCHAR(120) NOT NULL,
    'fechaTermino' DATE NOT NULL,
    'idInvestigador' INT NOT NULL,
    FOREIGN KEY ('idInvestigador') REFERENCES 'Investigador' ('idInvestigador')
);

CREATE TABLE IF NOT EXISTS 'Produccion' (
    'idProduccion' INT AUTO_INCREMENT PRIMARY KEY,
    'tipo' VARCHAR(60) NOT NULL,
    'titulo' VARCHAR(230) NOT NULL,
    'anio' INT NOT NULL,
    'idInvestigador' INT NOT NULL,
    FOREIGN KEY ('idInvestigador') REFERENCES 'Investigador' ('idInvestigador')
);

CREATE TABLE IF NOT EXISTS 'Adscripcion' (
    'idAdscripcion' INT AUTO_INCREMENT PRIMARY KEY,
    'instituto' VARCHAR(30) NOT NULL,
    'area' VARCHAR(130) NOT NULL,
    'nombramiento' VARCHAR(10) NOT NULL,
    'fechaIngreso' DATE NOT NULL,
    'idInvestigador' INT NOT NULL,
    FOREIGN KEY ('idInvestigador') REFERENCES 'Investigador' ('idInvestigador')
);

CREATE TABLE IF NOT EXISTS 'Integrante' (
    'idIntegrante' INT AUTO_INCREMENT PRIMARY KEY,
    'paterno' VARCHAR(90) NOT NULL,
    'materno' VARCHAR(90) NOT NULL,
    'nombre' VARCHAR(110) NOT NULL,
    'email' VARCHAR(150) NOT NULL UNIQUE
);

CREATE TABLE IF NOT EXISTS 'CuerpoAcademico' (
    'idCuerpo' INT AUTO_INCREMENT PRIMARY KEY,
    'nombre' VARCHAR(230) NOT NULL,
    'idIntegrante' INT NOT NULL,
    FOREIGN KEY ('idIntegrante') REFERENCES 'Integrante' ('idIntegrante')
);

CREATE TABLE IF NOT EXISTS 'Linea' (
    'idLinea' INT AUTO_INCREMENT PRIMARY KEY,
    'nombre' VARCHAR(120) NOT NULL,
    'descripcion' VARCHAR(300) NOT NULL,
    'cuerpo' INT NOT NULL,
    FOREIGN KEY ('cuerpo') REFERENCES 'CuerpoAcademico' ('idCuerpo')
);

```

```

CREATE TABLE IF NOT EXISTS 'IntegranteLinea' (
    'idIntegranteLinea' INT AUTO_INCREMENT PRIMARY KEY,
    'vigente' BOOLEAN NOT NULL,
    'inicio' DATE NOT NULL,
    'termino' DATE NOT NULL,
    'idLinea' INT NOT NULL,
    'idIntegrante' INT NOT NULL,
    FOREIGN KEY ('idLinea') REFERENCES 'Linea'('idLinea'),
    FOREIGN KEY ('idIntegrante') REFERENCES 'Integrante'('idIntegrante'),
    CONSTRAINT 'uq_integrante_linea' UNIQUE ('idLinea', 'idIntegrante')
);

```

```

CREATE TABLE IF NOT EXISTS 'Profesor' (
    'idProfesor' INT AUTO_INCREMENT PRIMARY KEY,
    'paterno' VARCHAR(80) NOT NULL,
    'materno' VARCHAR(80) NOT NULL,
    'nombre' VARCHAR(80) NOT NULL,
    'email' VARCHAR(250) NOT NULL UNIQUE,
    'fechaIngreso' DATE NOT NULL
);

```

```

CREATE TABLE IF NOT EXISTS 'Programa' (
    'idPrograma' INT AUTO_INCREMENT PRIMARY KEY,
    'nivel' VARCHAR(25) NOT NULL,
    'nombre' VARCHAR(130) NOT NULL,
    'fechaInicio' DATE NOT NULL
);

```

```

CREATE TABLE IF NOT EXISTS 'Asignatura' (
    'idAsignatura' INT AUTO_INCREMENT PRIMARY KEY,
    'semestre' INT NOT NULL,
    'nombre' VARCHAR(120) NOT NULL,
    'idPrograma' INT NOT NULL,
    FOREIGN KEY ('idPrograma') REFERENCES 'Programa'('idPrograma')
);

```

```

CREATE TABLE IF NOT EXISTS 'Alumno' (
    'idAlumno' INT AUTO_INCREMENT PRIMARY KEY,
    'numeroCuenta' VARCHAR(80) NOT NULL UNIQUE,
    'paterno' VARCHAR(80) NOT NULL,
    'materno' VARCHAR(80) NOT NULL,
    'nombre' VARCHAR(80) NOT NULL,
    'email' VARCHAR(120) NOT NULL UNIQUE
);

```

```

CREATE TABLE IF NOT EXISTS 'Curso' (
    'idCurso' INT AUTO_INCREMENT PRIMARY KEY,
    'periodo' VARCHAR(15) NOT NULL,
    'anio' INT NOT NULL,
    'grupo' INT NOT NULL,
    'idProfesor' INT NOT NULL,
    'idAsignatura' INT NOT NULL,
    FOREIGN KEY ('idProfesor') REFERENCES 'Profesor'('idProfesor'),

```

```

FOREIGN KEY ('idAsignatura') REFERENCES 'Asignatura'('idAsignatura')
);

CREATE TABLE IF NOT EXISTS 'AlumnoCurso' (
    'idAlumnoCurso' INT AUTOINCREMENT PRIMARY KEY,
    'calificacion' DECIMAL(3,2) NOT NULL,
    'idCurso' INT NOT NULL,
    'idAlumno' INT NOT NULL,
    FOREIGN KEY ('idCurso') REFERENCES 'Curso'('idCurso'),
    FOREIGN KEY ('idAlumno') REFERENCES 'Alumno'('idAlumno'),
    CONSTRAINT 'uq_alumno_curso' UNIQUE ('idCurso', 'idAlumno')
);

```

Scripts insert

Listing 9: Tabla Investigador

```

INSERT INTO Investigador (paterno, materno, nombre, orcid, email, movil)
VALUES
('García', 'López', 'María Elena', '0000-0001-2345-6789',
'maria.garcia@email.com', '5512345678'),
('Martínez', 'Sánchez', 'Juan Carlos', '0000-0002-3456-7890',
'juan.martinez@email.com', '5523456789'),
('Rodríguez', 'Pérez', 'Ana Isabel', '0000-0003-4567-8901',
'ana.rodriguez@email.com', '5534567890');

```

Listing 10: Tabla Proyecto

```

INSERT INTO Proyecto (nombre, inicio, final, idInvestigador) VALUES
('Desarrollo de algoritmos cuánticos', '2020-01-15', '2023-06-30', 1),
('Nanomateriales para energía solar', '2019-05-20', '2022-11-30', 2),
('Machine Learning para finanzas', '2022-02-01', '2025-01-31', 3);

```

Listing 11: Tabla Formacion

```

INSERT INTO Formacion (grado, institucion, nombre, fechaTermino, idInvestigador)
VALUES
('Doctorado', 'UNAM', 'Ciencias Computacionales', '2015-06-20', 1),
('Doctorado', 'UAM', 'Física de Materiales', '2016-11-25', 2),
('Doctorado', 'ITESM', 'Bioinformática', '2017-03-18', 3);

```

Listing 12: Tabla Produccion

```

INSERT INTO Produccion (tipo, titulo, año, idInvestigador) VALUES
('Libro', 'Fundamentos de IA en medicina', 2022, 1),
('Capítulo', 'Aplicaciones de nanomateriales', 2020, 2),
('Artículo', 'ML aplicado a mercados financieros', 2023, 3);

```

Listing 13: Tabla Adscripcion

```

INSERT INTO Adscripcion (instituto, area, nombramiento, fechaIngreso,
idInvestigador) VALUES
('Instituto de Ciencias Físicas', 'Computación Cuántica', 'Titular',
'2016-02-15', 1),
('Instituto de Biotecnología', 'Bioinformática',
'Asociado', '2018-07-01', 2),
('Instituto de Ingeniería', 'Robótica', 'Titular',
'2015-09-10', 3),

```

Listing 14: Tabla Integrante

```
INSERT INTO Integrante (paterno, materno, nombre, email) VALUES
('Garc a ', 'L pez ', 'Mar a Elena', 'maria.garcia@email.com'),
('Mart nez ', 'S nchez ', 'Juan Carlos', 'juan.martinez@email.com'),
('Rodr guez ', 'P rez ', 'Ana Isabel', 'ana.rodriguez@email.com');
```

Listing 15: Tabla CuerpoAcademico

```
INSERT INTO CuerpoAcademico (nombre, idIntegrante) VALUES
('Ciencias Computacionales Avanzadas', 1),
('Nanomateriales y Energ a ', 2),
('Biotecnolog a Aplicada', 3);
```

Listing 16: Tabla Linea

```
INSERT INTO Linea (nombre, descripcion, cuerpo) VALUES
('Algoritmos Cu nticos ', 'Desarrollo de algoritmos para
computaci n cu ntica y sus aplicaciones ', 1),
('Materiales Fotovoltaicos ', 'Dise o y caracterizaci n
de nuevos materiales para celdas solares ', 2),
('CRISPR en Agricultura ', 'Aplicaci n de t cnicas
de edici n gen tica en cultivos ', 3);
```

Listing 17: Tabla IntegranteLinea

```
INSERT INTO IntegranteLinea (vigente, inicio, termino, idLinea, idIntegrante) VALUES
(1, '2020-01-10', '2025-12-31', 1, 1),
(1, '2019-05-15', '2024-11-30', 2, 2),
(1, '2021-03-20', '2026-06-30', 3, 3);
```

Listing 18: Tabla Profesor

```
INSERT INTO Profesor (paterno, materno, nombre, email, fechaIngreso) VALUES
('Garc a ', 'L pez ', 'Mar a Elena', 'maria.garcia@email.com', '2015-08-20'),
('Mart nez ', 'S nchez ', 'Juan Carlos', 'juan.martinez@email.com', '2016-03-15'),
('Rodr guez ', 'P rez ', 'Ana Isabel', 'ana.rodriguez@email.com', '2017-01-10');
```

Listing 19: Tabla Programa

```
INSERT INTO Programa (nivel, nombre, fechaInicio) VALUES
('Licenciatura ', 'Ingenier a en Computaci n ', '2010-08-01'),
('Maestr a ', 'Ciencias de la Computaci n ', '2012-01-15'),
('Doctorado ', 'Tecnolog as de la Informaci n ', '2015-08-20');
```

Listing 20: Tabla Asignatura

```
INSERT INTO Asignatura (semestre, nombre, idPrograma) VALUES
(1, 'Programaci n B sica ', 1),
(3, 'Estructuras de Datos ', 1),
(5, 'Bases de Datos ', 1);
```

Listing 21: Tabla Alumno

```
INSERT INTO Alumno (numeroCuenta, paterno, materno, nombre, email) VALUES
('A01234567', 'L pez ', 'Garc a ', 'Juan', 'juan.lopez@alumno.edu.mx'),
('A02345678', 'Mart nez ', 'S nchez ', 'Ana', 'ana.martinez@alumno.edu.mx'),
('A03456789', 'Gonz lez ', 'P rez ', 'Luis', 'luis.gonzalez@alumno.edu.mx');
```

Listing 22: Tabla Curso

```
INSERT INTO Curso (periodo , anio , grupo , idProfesor , idAsignatura) VALUES
('Primavera' , 2023 , 101 , 1 , 1),
('Oto o' , 2023 , 201 , 2 , 5),
('Verano' , 2023 , 301 , 3 , 8);
```

Listing 23: Tabla AlumnoCurso

```
INSERT INTO AlumnoCurso (calificaci n , idCurso , idAlumno) VALUES
(8.5 , 1 , 1),
(9.0 , 2 , 2),
(7.5 , 3 , 3);
```

8. Scripts de extracción de datos

Scripts usados para extraer datos de LCS-1, LCS-2 y LCS-3:

Listing 24: Investigador

```
SELECT * FROM 'lcs-1'.Investigador
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Investigador.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Listing 25: Proyecto

```
SELECT * FROM 'lcs-1'.Proyecto
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Proyecto.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Listing 26: Formacion

```
SELECT * FROM 'lcs-1'.Formacion
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Formacion.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Listing 27: Produccion

```
SELECT * FROM 'lcs-1'.Produccion
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Produccion.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Listing 28: Adscripcion

```
SELECT * FROM 'lcs-1'.Adscripcion
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Adscripcion.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Listing 29: Integrante

```
SELECT * FROM 'lcs-2'.Integrante
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Integrante.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Listing 30: CuerpoAcademico

```
SELECT * FROM 'lcs-2'.CuerpoAcademico
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/CuerpoAcademico.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Listing 31: Linea

```
SELECT * FROM 'lcs-2'.Linea
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Linea.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Listing 32: IntegranteLinea

```
SELECT * FROM 'lcs-2'.IntegranteLinea
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/IntegranteLinea.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Listing 33: Profesor

```
SELECT * FROM 'lcs-3'.Profesor
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Profesor.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Listing 34: Programa

```
SELECT * FROM 'lcs-3'.Programa
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Programa.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Listing 35: Asignatura

```
SELECT * FROM 'lcs-3'.Asignatura
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Asignatura.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Listing 36: Alumno

```
SELECT * FROM 'lcs-3'.Alumno
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Alumno.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Listing 37: Curso

```
SELECT * FROM 'lcs-3'.Curso
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Curso.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

Listing 38: AlumnoCurso

```
SELECT * FROM 'lcs-3'.AlumnoCurso
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/AlumnoCurso.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

9. Script de carga de datos

Scripts para cargar datos de las bases de datos locales en GCS:

Listing 39: Investigador

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Investigador.csv'
INTO TABLE Investigador
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

Listing 40: Integrante

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Integrante.csv'
INTO TABLE Integrante
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

Listing 41: Profesor

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Profesor.csv'
INTO TABLE Profesor
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

Listing 42: Programa

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Programa.csv'
INTO TABLE Programa
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

Listing 43: Alumno

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Alumno.csv'
INTO TABLE Alumno
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

Listing 44: CuerpoAcademico

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/CuerpoAcademico.csv'
INTO TABLE CuerpoAcademico
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

Listing 45: Linea

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Linea.csv'
INTO TABLE Linea
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

Listing 46: Proyecto

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Proyecto.csv'
INTO TABLE Proyecto
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

Listing 47: Formacion

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Formacion.csv'
INTO TABLE Formacion
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

Listing 48: Produccion

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Produccion.csv'
INTO TABLE Produccion
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

Listing 49: Adscripcion

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Adscripcion.csv'
INTO TABLE Adscripcion
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

Listing 50: IntegranteLinea

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/IntegranteLinea.csv'
INTO TABLE IntegranteLinea
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

Listing 51: Asignatura

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Asignatura.csv'
INTO TABLE Asignatura
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

Listing 52: Curso

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Curso.csv'
INTO TABLE Curso
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```

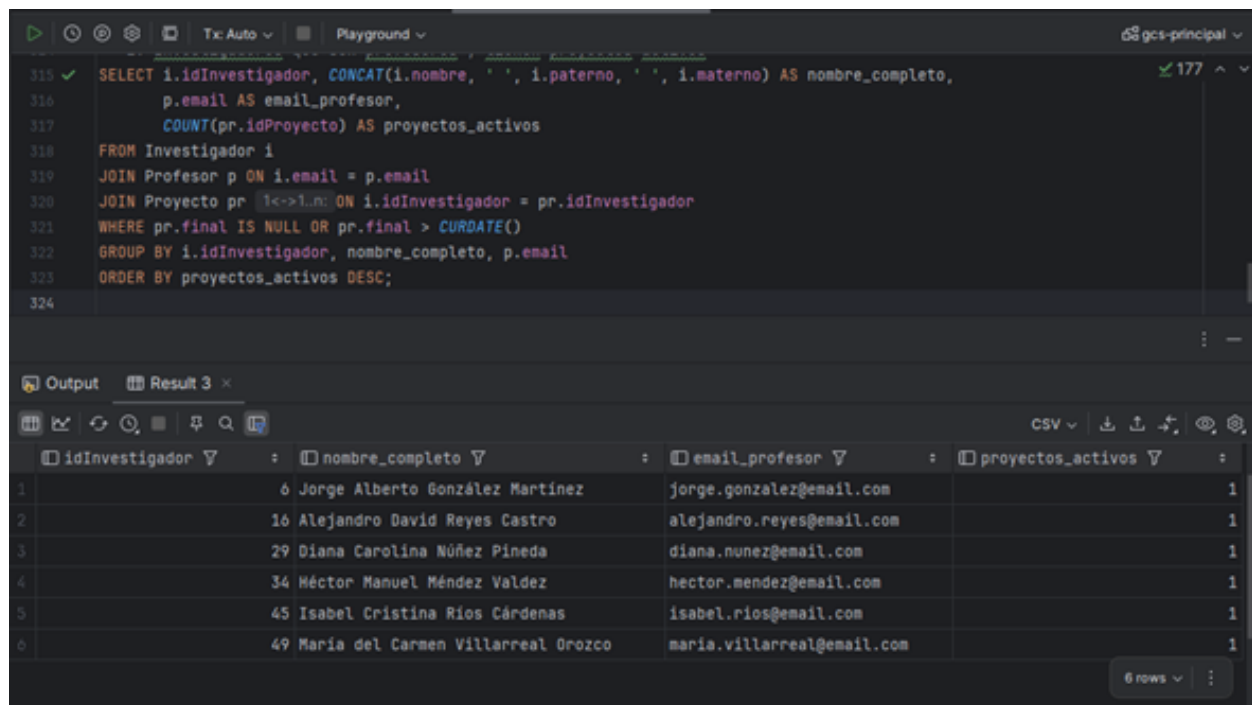
Listing 53: AlumnoCurso

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/AlumnoCurso.csv'
INTO TABLE AlumnoCurso
FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n' IGNORE 1 ROWS;
```


10. Consultas de datos

Listing 54: Investigadores que son profesores y tienen proyectos activos

```
SELECT i.idInvestigador , CONCAT(i.nombre, ' ', i.paterno, ' ', i.materno)
AS nombre_completo,
p.email AS email_profesor ,
COUNT(pr.idProyecto) AS proyectos_activos
FROM Investigador i
JOIN Profesor p ON i.email = p.email
JOIN Proyecto pr ON i.idInvestigador = pr.idInvestigador
WHERE pr.final IS NULL OR pr.final > CURDATE()
GROUP BY i.idInvestigador , nombre_completo , p.email
ORDER BY proyectos_activos DESC;
```



The screenshot shows a SQL playground interface with a query editor and a results pane. The query is the same as in Listing 54. The results pane shows 6 rows of data, sorted by the number of active projects in descending order.

idInvestigador	nombre_completo	email_profesor	proyectos_activos
6	Jorge Alberto González Martínez	jorge.gonzalez@email.com	1
16	Alejandro David Reyes Castro	alejandro.reyes@email.com	1
29	Diana Carolina Núñez Pineda	diana.nunez@email.com	1
34	Héctor Manuel Méndez Valdez	hector.mendez@email.com	1
45	Isabel Cristina Ríos Cárdenas	isabel.rios@email.com	1
49	María del Carmen Villarreal Orozco	maria.villarreal@email.com	1

Figure 2: Consulta 1.

Listing 55: Cursos impartidos por Investigadores con sus Líneas de Investigación

```
SELECT c.idCurso , a.nombre AS asignatura ,
CONCAT(i.nombre, ' ', i.paterno) AS profesor ,
l.nombre AS linea_investigacion , ca.nombre AS cuerpo_academico
FROM Curso c
JOIN Asignatura a ON c.idAsignatura = a.idAsignatura
JOIN Profesor p ON c.idProfesor = p.idProfesor
JOIN Investigador i ON p.email = i.email
JOIN Integrante ing ON i.email = ing.email
JOIN IntegranteLinea il ON ing.idIntegrante = il.idIntegrante
JOIN Linea l ON il.idLinea = l.idLinea
JOIN CuerpoAcademico ca ON l.cuerpo = ca.idCuerpo
WHERE il.vigente = 1;
```

gcs-principal [Practicas] x GCS [Practicas] lcsprincipal [Practicas] lcsmantenimil

Playground

```

326 SELECT c.idCurso, a.nombre AS asignatura,
327        CONCAT(i.nombre, ' ', i.paterno) AS profesor,
328        l.nombre AS linea_investigacion, ca.nombre AS cuerpo_academico
329 FROM Curso c
330 JOIN Asignatura a ON c.idAsignatura = a.idAsignatura
331 JOIN Profesor p ON c.idProfesor = p.idProfesor
332 JOIN Investigador i ON p.email = i.email
333 JOIN Integrante ing ON i.email = ing.email
334 JOIN IntegranteLinea il ON ing.idIntegrante = il.idIntegrante
335 JOIN Linea l ON il.idLinea = l.idLinea
336 JOIN CuerpoAcademico ca ON l.cuerpo = ca.idCuerpo
337 WHERE il.vigente = 1;
338

```

Output Result 4 x

CSV

idCurso	asignatura	profesor	linea_investigacion	cuerpo_academico
1	1 Programación Básica	María Elena García	Algoritmos Cuánticos	Ciencias Computacionales Avanz
2	2 Proyecto Integrador	Juan Carlos Martínez	Materiales Fotovoltaicos	Nanomateriales y Energía
3	3 Procesamiento de Lenguaje Natural	Ana Isabel Rodríguez	CRISPR en Agricultura	Biotechnología Aplicada
4	4 Teoría de la Computación	Luis Fernando Hernández	Robots Colaborativos	Robótica e Inteligencia Artifi
5	5 Seminario de Investigación	Carmen Rosa López	Gamificación Educativa	Educación y Te

331:41 CRLF UTF-8 4 spaces

Figure 3: Consulta 2.

11. Trigger

Un disparador en base de datos es un código que se ejecuta automáticamente cuando se produce un evento en una tabla. También se le conoce como trigger en inglés

Trigger 1

Listing 56: Trigger para registrar actividad cuando un profesor califica

```
DELIMITER //
CREATE TRIGGER after_calificacion_update
AFTER UPDATE ON AlumnoCurso
FOR EACH ROW
BEGIN
    DECLARE profesor_id INT;
    DECLARE investigador_id INT;

    — Obtener el ID del profesor del curso
    SELECT c.idProfesor INTO profesor_id FROM Curso c WHERE c.idCurso = NEW.idCurso;

    — Obtener el ID del investigador asociado al profesor
    SELECT i.idInvestigador INTO investigador_id
    FROM Investigador i
    JOIN Profesor p ON i.email = p.email
    WHERE p.idProfesor = profesor_id;

    — Registrar producci n acad mica si la calificaci n es alta
    IF NEW.calificacion >= 9.0 THEN
        INSERT INTO Produccion (tipo, titulo, anio, idInvestigador)
        VALUES ('Reporte', CONCAT('Alumno destacado en curso ID ', NEW.idCurso),
        YEAR(CURDATE()), investigador_id);
    END IF;
END//
DELIMITER ;
```

12. Procedimiento Almacenado

Un procedimiento almacenado es un conjunto de instrucciones que se guardan en una base de datos y se pueden ejecutar desde otras consultas o procedimientos almacenados. Son una herramienta fundamental para la gestión de bases de datos relacionales.

Procedimiento 1

Listing 57: Al ingresar un nuevo investigador

```
DELIMITER //
CREATE PROCEDURE registrar_investigador_completo(
    IN p_paterno VARCHAR(90),
    IN p_materno VARCHAR(90),
    IN p_nombre VARCHAR(120),
    IN p_orcid VARCHAR(30),
    IN p_email VARCHAR(150),
    IN p_movil VARCHAR(15),
    IN p_instituto VARCHAR(30),
    IN p_area VARCHAR(130),
    IN p_nombramiento VARCHAR(10)
)
BEGIN
    DECLARE new_id INT;
```

```

— Insertar en Investigador
INSERT INTO Investigador (paterno, materno, nombre, orcid, email, movil)
VALUES (p_paterno, p_materno, p_nombre, p_orcid, p_email, p_movil);

SET new_id = LAST_INSERT_ID();

— Insertar en Integrante
INSERT INTO Integrante (paterno, materno, nombre, email)
VALUES (p_paterno, p_materno, p_nombre, p_email);

— Insertar en Profesor
INSERT INTO Profesor (paterno, materno, nombre, email, fechaIngreso)
VALUES (p_paterno, p_materno, p_nombre, p_email, CURDATE());

— Insertar adscripci n
INSERT INTO Adscripcion (instituto, area, nombramiento, fechaIngreso, idInvestigador)
VALUES (p_instituto, p_area, p_nombramiento, CURDATE(), new_id);

SELECT CONCAT('Investigador registrado con ID: ', new_id) AS mensaje;
END//
DELIMITER ;

```

13. Capa de Software

Sistema de Gestión Académica

Investigadores Proyectos Reportes

ID	Nombre Completo	Email	ORCID	Móvil
1	Maria Elena García López	maria.garcia@email.com	0000-0001-2345-6789	5512345678
2	Juan Carlos Martínez Sánchez	juan.martinez@email.com	0000-0002-3456-7890	5523456789
3	Ana Isabel Rodríguez Pérez	ana.rodriguez@email.com	0000-0003-4567-8901	5534567890
4	Luis Fernando Hernández Gómez	luis.hernandez@email.com	0000-0004-5678-9012	5545678901
5	Carmen Rosa López Díaz	carmen.lopez@email.com	0000-0005-6789-0123	5556789012
6	Jorge Alberto González Martínez	jorge.gonzalez@email.com	0000-0006-7890-1234	5567890123
7	Patricia Beatriz Pérez Rodríguez	patricia.perez@email.com	0000-0007-8901-2345	5578901234
8	Ricardo Javier Sánchez Hernández	ricardo.sanchez@email.com	0000-0008-9012-3456	5589012345
9	Laura Gabriela Ramírez González	laura.ramirez@email.com	0000-0009-0123-4567	5590123456
10	Miguel Ángel Flores Vázquez	miguel.flores@email.com	0000-0010-1234-5678	5501234567
11	Sofía Alejandra Torres Jiménez	sofia.torres@email.com	0000-0011-2345-6789	5512345679
12	Diego Armando Castillo Ruiz	diego.castillo@email.com	0000-0012-3456-7890	5523456790
13	Adriana Elizabeth Morales Ortiz	adriana.morales@email.com	0000-0013-4567-8901	5534567901
14	Fernando Javier Ortega Silva	fernando.ortega@email.com	0000-0014-5678-9012	5545679012
15	Claudia Patricia Vega Mendoza	claudia.vega@email.com	0000-0015-6789-0123	5556790123
16	Alejandro David Reyes Castro	alejandro.reyes@email.com	0000-0016-7890-1234	5567901234
17	Gabriela Alejandra Mendoza Guerrero	gabriela.mendoza@email.com	0000-0017-8901-2345	5578901235
18	Roberto Carlos Guerrero Ríos	roberto.guerrero@email.com	0000-0018-9012-3456	5589012346
19	Verónica Isabel Cruz Navarro	veronica.cruz@email.com	0000-0019-0123-4567	5590123457
20	Daniel Eduardo Ortiz Vargas	daniel.ortiz@email.com	0000-0020-1234-5678	5501345678
21	Lucía Fernanda Juárez Romero	lucia.juarez@email.com	0000-0021-2345-6789	5512456789
22	Oscar Manuel Medina Cervantes	oscar.medina@email.com	0000-0022-3456-7890	5523567890
23	Mariana Guadalupe Cortés Salazar	mariana.cortes@email.com	0000-0023-4567-8901	5534678901
24	José Antonio Rojas Miranda	jose.rojas@email.com	0000-0024-5678-9012	5545789012
25	Silvia Patricia Acosta Campos	silvia.acosta@email.com	0000-0025-6789-0123	5556890123
26	Francisco Javier Moreno Vega	francisco.moreno@email.com	0000-0026-7890-1234	5567901234
27	Teresa Margarita Delgado Rosas	teresa.delgado@email.com	0000-0027-8901-2345	5578901235

Agregar Editar Eliminar Actualizar Ordenar

Buscar: Buscar

Figure 4: Registros Investigador.

Sistema de Gestión Académica

Investigadores Proyectos Reportes

ID	Nombre	Inicio	Final	Investigador
1	Desarrollo de algoritmos cuánticos	2020-01-15	2023-06-30	Maria Elena García
2	Nanomateriales para energía solar	2019-05-20	2022-11-30	Juan Carlos Martínez
3	Machine Learning para finanzas	2022-02-01	2025-01-31	Ana Isabel Rodríguez
4	Biotecnología agrícola sostenible	2020-07-15	2023-09-30	Luis Fernando Hernández
5	Robótica colaborativa industrial	2021-09-01	2024-08-31	Carmen Rosa López
6	Realidad virtual en educación	2022-01-10	2025-06-30	Jorge Alberto González
7	Blockchain para transparencia gubernamental	2020-11-15	2023-12-31	Patricia Beatriz Pérez
8	Materiales compuestos avanzados	2019-08-20	2024-05-15	Ricardo Javier Sánchez
9	Genómica de poblaciones	2021-04-05	2025-03-31	Laura Gabriela Ramírez
10	Sistemas autónomos para agricultura	2020-10-01	2023-12-15	Miguel Ángel Flores
11	Diagnóstico temprano de cáncer con IA	2021-07-20	2024-11-30	Sofía Alejandra Torres
12	Energías renovables en zonas urbanas	2022-03-15	2025-02-28	Diego Armando Castillo
13	Fármacos inteligentes para diabetes	2019-12-10	2023-10-31	Adriana Elizabeth Morales
14	Ciudades inteligentes y movilidad	2020-06-05	2024-05-30	Fernando Javier Ortega
15	Nanotecnología aplicada a medicina	2021-01-25	2024-09-15	Claudia Patricia Vega
16	Big Data para análisis climático	2022-05-10	2025-04-30	Alejandro David Reyes
17	Robótica de rehabilitación	2020-09-15	2023-08-31	Gabriela Alejandra Mendoza
18	Realidad aumentada en museos	2021-11-20	2024-10-15	Roberto Carlos Guerrero
19	Criptografía post-cuántica	2019-04-01	2023-07-31	Verónica Isabel Cruz
20	Biodiversidad en ecosistemas urbanos	2020-08-12	2024-06-30	Daniel Eduardo Ortiz
21	Inteligencia artificial ética	2021-02-28	2025-01-15	Lucía Fernanda Juárez
22	Materiales para almacenamiento de energía	2022-04-05	2025-03-31	Oscar Manuel Medina
23	Telemedicina en zonas rurales	2020-12-10	2023-11-30	Mariana Guadalupe Cortés
24	Agricultura de precisión con drones	2019-07-15	2023-05-31	José Antonio Rojas
25	Modelado de pandemias	2021-05-20	2024-12-31	Silvia Patricia Acosta
26	Tecnologías para el envejecimiento activo	2022-02-15	2025-01-31	Francisco Javier Moreno
27	Sistemas de recomendación personalizados	2020-10-30	2024-09-30	Teresa Margarita Delgado
28	Computación cuántica aplicada	2019-03-25	2023-12-31	Raúl Alejandro Castañeda

Agregar Editar Eliminar Actualizar Ordenar Filtrar Activos

Figure 5: Registros Proyectos.

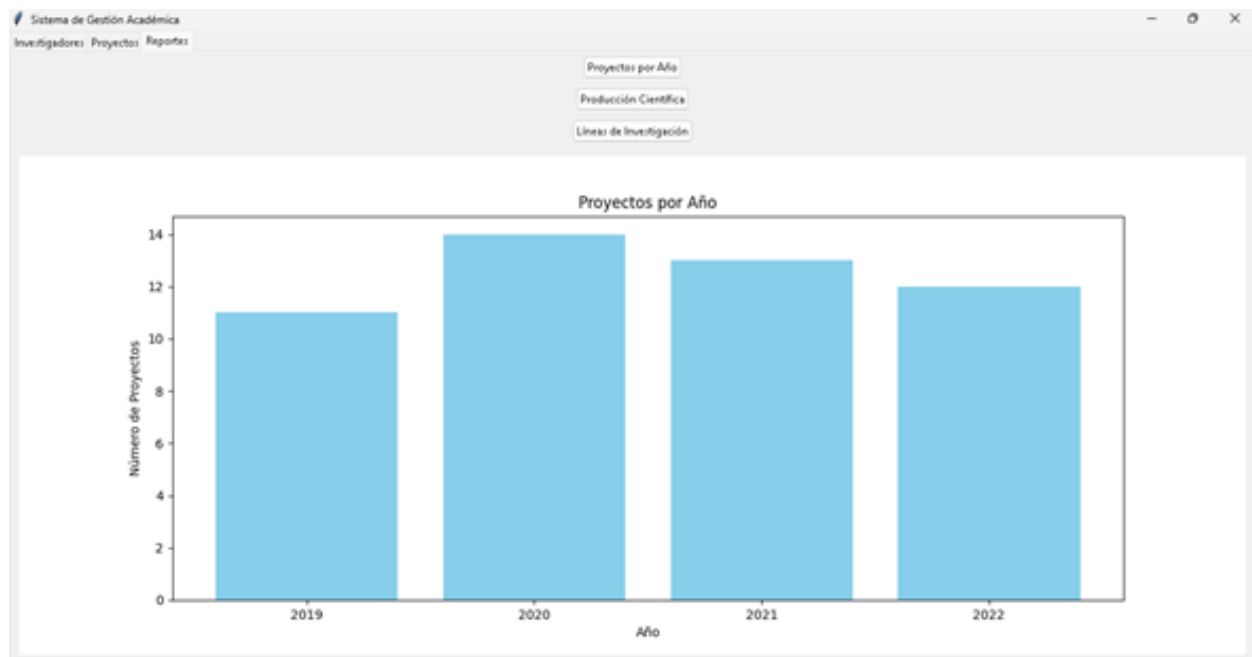



Figure 6: Reportes.

 Agregar Investigador — □ ×

Apellido Paterno:

Apellido Materno:

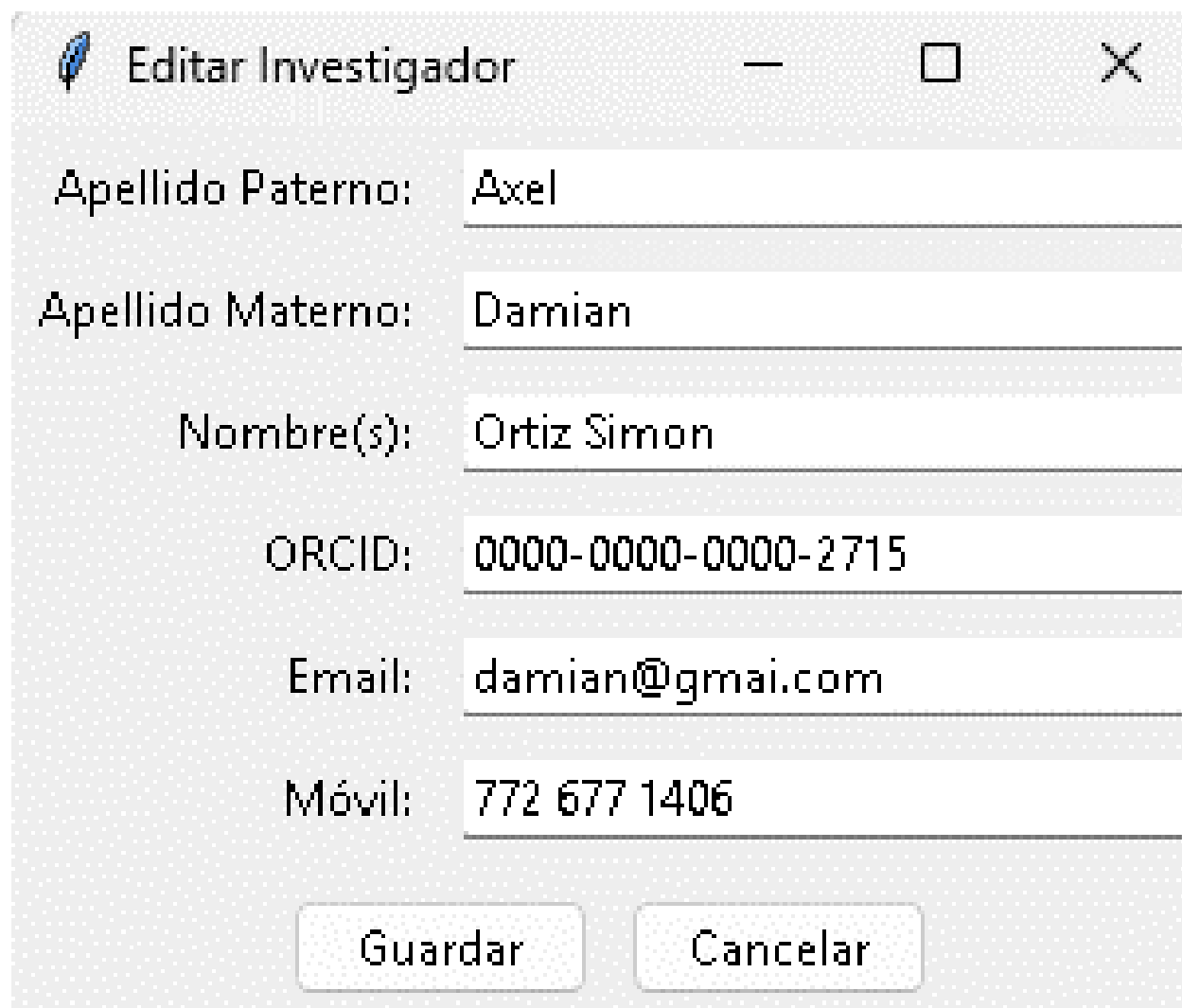
Nombre(s):


ORCID:

Email:

Móvil:

Figure 7: Operaciones CRUD-Crear.



 Editar Investigador

Apellido Paterno: Axel

Apellido Materno: Damian

Nombre(s): Ortiz Simon

ORCID: 0000-0000-0000-2715

Email: damian@gmai.com

Móvil: 772 677 1406

Guardar Cancelar

Figure 8: Operaciones CRUD-Actualizar.

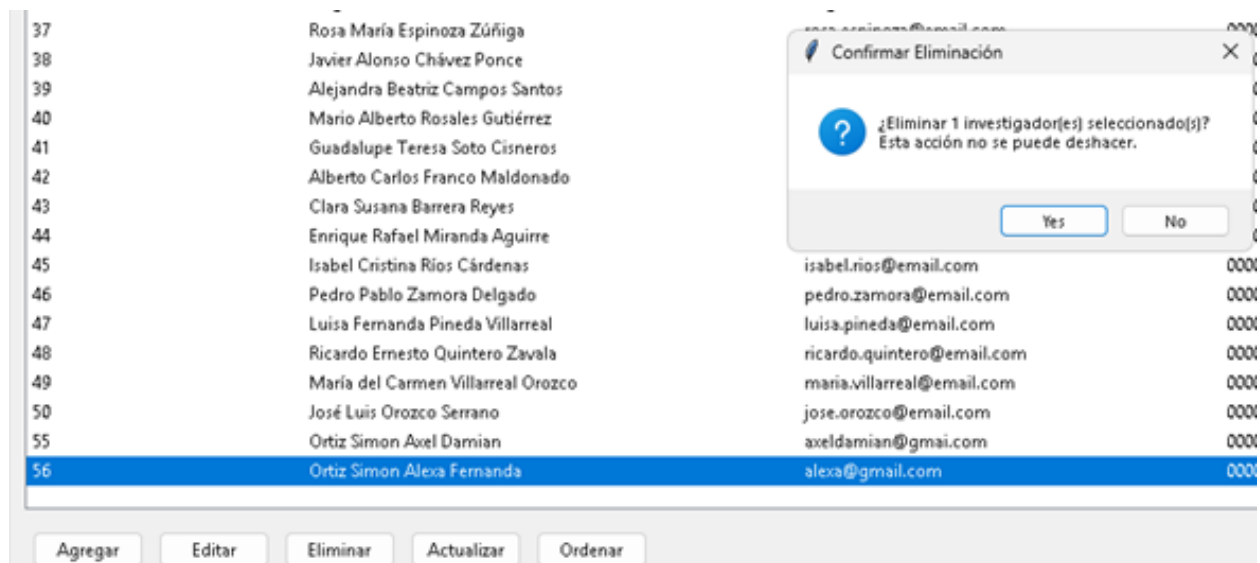


Figure 9: Operaciones CRUD-Eliminar.

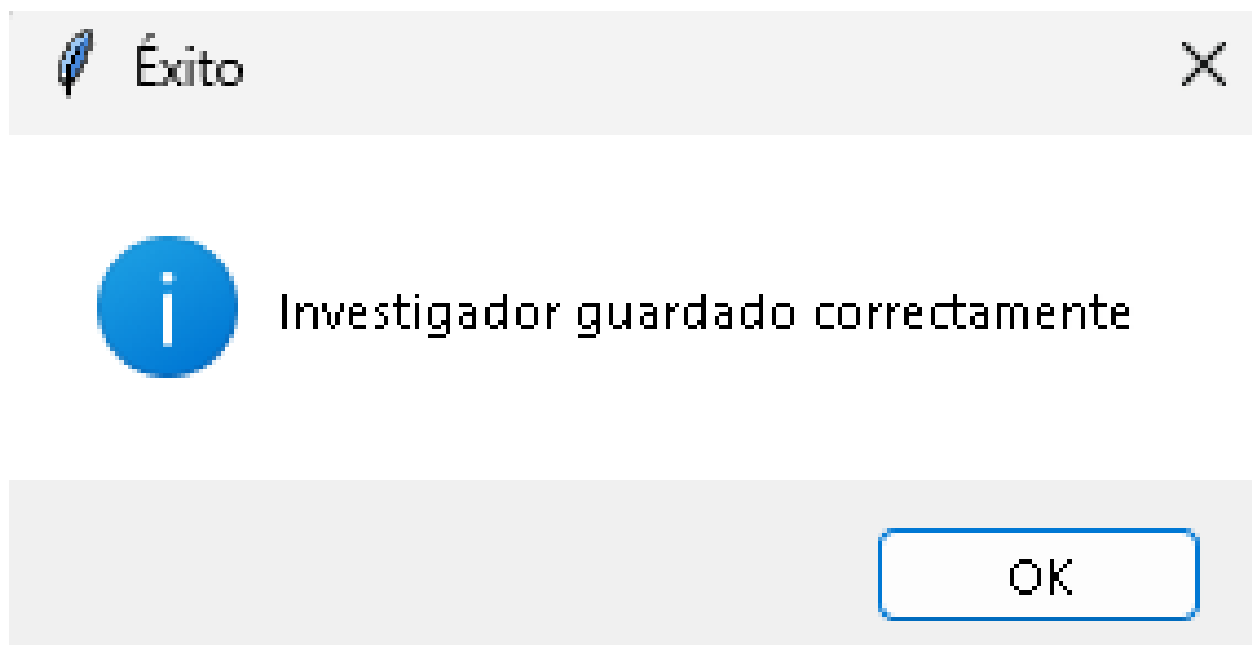


Figure 10: Mensaje de éxito al agregar un nuevo registro.

Sistema de Gestión Académica

Investigadores: Proyectos: Reportes:

ID	Nombre Completo	Email	ORCID	Móvil
26	Francisco Javier Moreno Vega	francisco.moreno@email.com	0000-0016-7890-1234	5567901234
27	Teresa Margarita Delgado Rosas	teresa.delgado@email.com	0000-0017-8901-2345	5579012345
28	Raúl Alejandro Castañeda Ochoa	raul.castaneda@email.com	0000-0018-9012-3456	5589123456
29	Diana Carolina Núñez Pineda	diana.nunez@email.com	0000-0019-0123-4567	5590234567
30	Arturo Benjamín Salazar Mejía	arturo.salazar@email.com	0000-0020-1234-5678	5501345678
31	Beatriz Adriana Vázquez Lara	beatriz.vazquez@email.com	0000-0021-2345-6789	5512456789
32	Gerardo Ignacio Aguilar Sosa	gerardo.aguilar@email.com	0000-0022-3456-7890	5523567890
33	Leticia Margarita Rivera Fuentes	leticia.rivera@email.com	0000-0023-4567-8901	5534678901
34	Héctor Manuel Méndez Valdez	hector.mendez@email.com	0000-0024-5678-9012	5545789012
35	Norma Angélica Cervantes Rangel	norma.cervantes@email.com	0000-0025-6789-0123	5556890123
36	Sergio Eduardo Valencia Barrera	sergio.valencia@email.com	0000-0026-7890-1234	5567901234
37	Rosa María Espinoza Zúñiga	rosa.espinoza@email.com	0000-0027-8901-2345	5578912345
38	Javier Alonso Chávez Ponce	javier.chavez@email.com	0000-0028-9012-3456	5589023456
39	Alejandra Beatriz Campos Santos	alejandra.campos@email.com	0000-0029-0123-4567	5590134567
40	Mario Alberto Rosales Gubínez	mario.rosales@email.com	0000-0040-1234-5678	5501445678
41	Guadalupe Teresa Soto Cisneros	guadalupe.soto@email.com	0000-0041-2345-6789	5512556789
42	Alberto Carlos Franco Maldonado	alberto.franco@email.com	0000-0042-3456-7890	5523667890
43	Clara Susana Barrera Reyes	clara.barrera@email.com	0000-0043-4567-8901	5534778901
44	Enrique Rafael Miranda Aguirre	enrique.miranda@email.com	0000-0044-5678-9012	5545889012
45	Isabel Cristina Ríos Cárdenas	isabel.rios@email.com	0000-0045-6789-0123	5556990123
46	Pedro Pablo Zamora Delgado	pedro.zamora@email.com	0000-0046-7890-1234	5567901234
47	Luisa Fernanda Pineda Villarreal	luisa.pineda@email.com	0000-0047-8901-2345	5578912345
48	Ricardo Ernesto Quintero Zavala	ricardo.quintero@email.com	0000-0048-9012-3456	5589023456
49	Maria del Carmen Villarreal Orozco	maria.villarreal@email.com	0000-0049-0123-4567	5590134567
50	José Luis Orozco Serrano	jose.orozco@email.com	0000-0050-1234-5678	5501445678
51	Axel Damián Ortiz Simón	axel@gmail.com	0000-0001-2715-7799	7716771406
53	Iván Alejandro Iben Cruz	alegoat@gmail.com	0000-0000-0000-1234	771 777 1234

Agregar Editar Eliminar Actualizar Ordenar

Borrar

Figure 11: Tabla investigador con el nuevo registro.

Practicas Version control

Database Explorer

gcs-principal 12 of 28

tables 15

investigador

WHERE

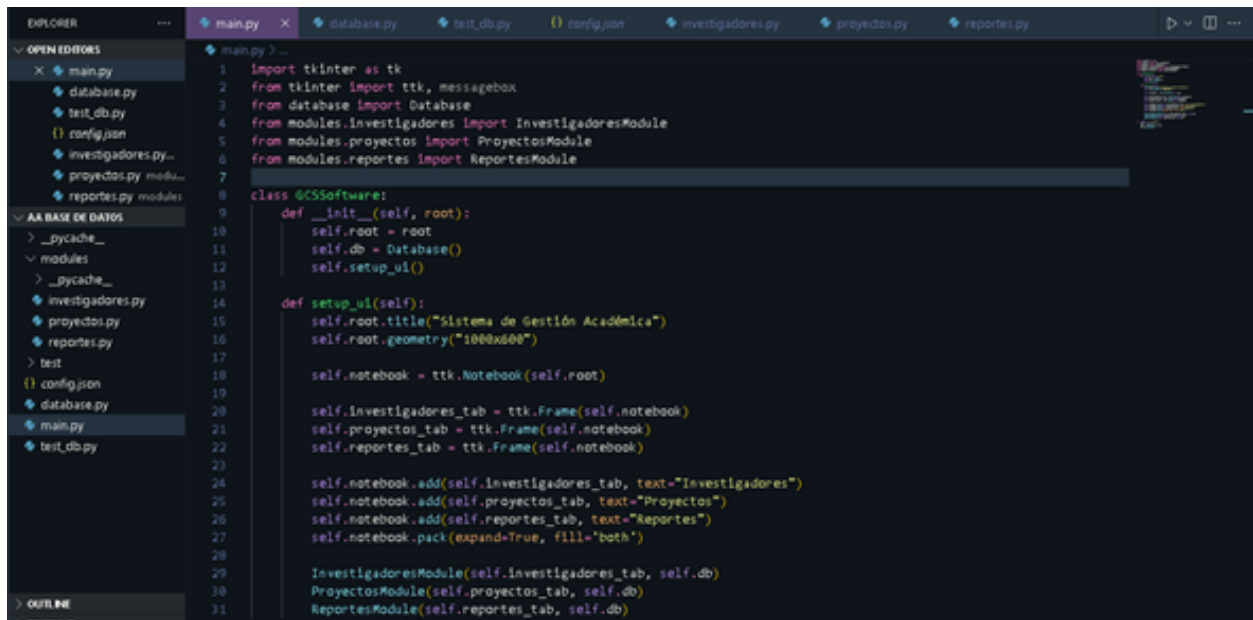
ORDER BY

ID Investigador	paterno	materno	nombre	orcid	email
39	Campos	Santos	Alejandra Beatriz	0000-0039-0123-4567	alejandra.campos@email.com
40	Rosales	Gutiérrez	Mario Alberto	0000-0040-1234-5678	mario.rosales@email.com
41	Soto	Cisneros	Guadalupe Teresa	0000-0041-2345-6789	guadalupe.soto@email.com
42	Franco	Maldonado	Alberto Carlos	0000-0042-3456-7890	alberto.franco@email.com
43	Barrera	Reyes	Clara Susana	0000-0043-4567-8901	clara.barrera@email.com
44	Miranda	Aguirre	Enrique Rafael	0000-0044-5678-9012	enrique.miranda@email.com
45	Ríos	Cárdenas	Isabel Cristina	0000-0045-6789-0123	isabel.rios@email.com
46	Zamora	Delgado	Pedro Pablo	0000-0046-7890-1234	pedro.zamora@email.com
47	Pineda	Villarreal	Luisa Fernanda	0000-0047-8901-2345	luisa.pineda@email.com
48	Quintero	Zavala	Ricardo Ernesto	0000-0048-9012-3456	ricardo.quintero@email.com
49	Villarreal	Orozco	Maria del Carmen	0000-0049-0123-4567	maria.villarreal@email.com
50	Orozco	Serrano	José Luis	0000-0050-1234-5678	jose.orozco@email.com
51	Ortiz	Simón	Axel Damián	0000-0001-2715-7799	axel@gmail.com
53	Jaen	Cruz	Iván Alejandro	0000-0000-0000-1234	alegoat@gmail.com

52 rows

Database > Practicas > gcs-principal > tables > investigador

Figure 12: Actualizacion exitosa en base de datos.



The image shows a Python IDE interface. On the left, there is a sidebar with two main sections: 'EXPLORER' and 'OUTLINE'. The 'EXPLORER' section shows a project structure with folders like 'AA BASE DE DATOS' and 'modules', and files like 'main.py', 'database.py', 'test_db.py', 'config.json', 'investigadores.py', 'proyectos.py', and 'reportes.py'. The 'OUTLINE' section is currently empty. The main area is a code editor showing the content of 'main.py'. The code is written in Python and uses Tkinter for GUI creation. It imports 'tkinter' as 'tk' and 'messagebox' from 'tkinter'. It also imports 'Database' from 'database' and 'InvestigadoresModule' from 'modules.investigadores'. The code defines a class 'GCSSoftware' with two methods: '__init__' and 'setup_ui'. The '__init__' method initializes 'self.root' as the root window, 'self.db' as a 'Database' object, and calls 'self.setup_ui()'. The 'setup_ui' method sets the title of the root window to 'Sistema de Gestión Académica', sets the geometry to '1000x600', creates a 'Notebook' widget, and adds three tabs: 'Investigadores', 'Proyectos', and 'Reportes'. Each tab contains a 'Frame' widget. The 'Investigadores' tab contains an 'InvestigadoresModule' object, the 'Proyectos' tab contains a 'ProyectosModule' object, and the 'Reportes' tab contains a 'ReportesModule' object. The code is numbered from 1 to 31.

```
1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 from database import Database
4 from modules.investigadores import InvestigadoresModule
5 from modules.proyectos import ProyectosModule
6 from modules.reportes import ReportesModule
7
8 class GCSSoftware:
9     def __init__(self, root):
10         self.root = root
11         self.db = Database()
12         self.setup_ui()
13
14     def setup_ui(self):
15         self.root.title("Sistema de Gestión Académica")
16         self.root.geometry("1000x600")
17
18         self.notebook = ttk.Notebook(self.root)
19
20         self.investigadores_tab = ttk.Frame(self.notebook)
21         self.proyectos_tab = ttk.Frame(self.notebook)
22         self.reportes_tab = ttk.Frame(self.notebook)
23
24         self.notebook.add(self.investigadores_tab, text="Investigadores")
25         self.notebook.add(self.proyectos_tab, text="Proyectos")
26         self.notebook.add(self.reportes_tab, text="Reportes")
27         self.notebook.pack(expand=True, fill="both")
28
29         InvestigadoresModule(self.investigadores_tab, self.db)
30         ProyectosModule(self.proyectos_tab, self.db)
31         ReportesModule(self.reportes_tab, self.db)
```

Figure 13: Python.

Referencias Bibliográficas

References

- [1] C. Coronel, S. Morris, y P. Rob, *Database Systems: Design, Implementation, and Management*, 13^a ed. Boston, MA: Cengage Learning, 2020.
- [2] R. Elmasri y S. Navathe, *Fundamentals of Database Systems*, 7^a ed. Boston, MA: Pearson, 2016.
- [3] P. DuBois, *MySQL: The Complete Reference*, 2^a ed. New York, NY: McGraw-Hill Education, 2008.
- [4] G. Ghiani, G. Laporte, y R. Musmanno, *Introduction to Logistics Systems Management*, 2^a ed. Chichester, UK: Wiley, 2013.
- [5] M. J. Hernandez, *Database Design for Mere Mortals*, 3^a ed. Boston, MA: Addison-Wesley, 2013.
- [6] M. T. Özsu y P. Valduriez, *Principles of Distributed Database Systems*, 3^a ed. New York, NY: Springer, 2011.
- [7] R. Kimball y M. Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 2^a ed. Indianapolis, IN: Wiley, 2008.
- [8] Oracle Corporation, *MySQL 8.0 Reference Manual*, 2020. [En línea]. Disponible en: <https://dev.mysql.com/doc/refman/8.0/en/>
- [9] R. Ramakrishnan y J. Gehrke, *Database Management Systems*, 3^a ed. New York, NY: McGraw-Hill, 2003.
- [10] C. J. Date, *Database Design and Relational Theory: Normal Forms and All That Jazz*, 2^a ed. Sebastopol, CA: O'Reilly Media, 2012.