Universidad Autónoma del Estado de Hidalgo

ICBI

REPORTE DE PRÁCTICA NO. 3.3

Integración con estrategia Bootom-Up

ALUMNO:
Ariana García Melo

1. Introducción

En el ámbito de las bases de datos distribuidas, la integración de múltiples esquemas locales en un esquema global coherente representa un desafío significativo. Este reporte documenta el proceso de integración bottom-up de tres modelos relacionales locales (representando diferentes subsistemas académicos) en un modelo conceptual global unificado. El objetivo es crear una base de datos distribuida que permita gestionar integralmente información sobre estudiantes, profesores, investigadores, líneas de investigación y programas académicos, eliminando redundancias y manteniendo la consistencia de los datos.

El enfoque bottom-up utilizado implica analizar primero los esquemas locales individuales, identificar entidades y relaciones comunes, y luego sintetizar progresivamente un esquema global que incorpore todas las necesidades de los subsistemas originales. Este método es particularmente útil en entornos distribuidos donde los sistemas han evolucionado de manera independiente pero requieren posterior integración.

El objetivo principal es centralizar la información dispersa, optimizar las consultas interrelacionadas y facilitar futuras tareas de análisis de datos.

Para lograrlo, se diseñó un proceso ETL (Extract, Transform, Load) que permite extraer los datos de los esquemas originales, realizar las transformaciones necesarias para su homogeneización, y cargarlos en una base de datos física de integración denominada IntegracionProyecto.

El proceso contempla la generación de datos ficticios para poblar las bases locales, así como la verificación de la consistencia y calidad de los datos integrados.

2. Marco teórico

Bases de Datos Distribuidas

Una base de datos distribuida es un conjunto de bases de datos lógicamente relacionadas que están distribuidas físicamente en diferentes ubicaciones conectadas por una red. Presentan ventajas como mayor disponibilidad, mejor rendimiento para usuarios locales y escalabilidad, pero también introducen desafíos en diseño e implementación.

Integración de Esquemas (Bottom-Up)

El enfoque bottom-up para integración de esquemas comienza con el análisis de esquemas locales existentes para luego derivar el esquema global. Este método es especialmente relevante cuando:

- Existen múltiples bases de datos autónomas que necesitan interoperar.
- Los sistemas han evolucionado de manera independiente.
- Se requiere mantener cierta autonomía local.

Proceso ETL (Extract, Transform, Load): Es una metodología usada en sistemas de integración de datos. Se compone de tres etapas:

1. Extracción: Obtención de los datos desde los sistemas fuente.
2. Transformación: Adaptación y depuración de los datos para corregir inconsistencias, unificar formatos y aplicar reglas de negocio.
3. Carga: Inserción de los datos ya transformados en el sistema de destino.

La normalización de Datos es el proceso que organiza los datos para minimizar la redundancia y dependencia, logrando una estructura de datos más eficiente.

La Unificación de esquemas es la técnica mediante la cual se integran diferentes estructuras de datos en un modelo común, resolviendo incompatibilidades entre nombres de atributos, tipos de datos y estructuras de las tablas.

La correcta implementación de un proceso ETL es fundamental para garantizar la calidad, confiabilidad y disponibilidad de los datos en sistemas de información centralizados.

Conflictos en Integración

Durante la integración pueden surgir diversos tipos de conflictos

- Conflictos de nombres: La misma entidad con nombres diferentes en esquemas locales.
- Conflictos estructurales: Diferentes representaciones para el mismo concepto.
- Conflictos de escalas: Mismos atributos con diferentes unidades o formatos.

3. Herramientas empleadas

Para la implementación de las consultas se utilizaron:

- MySQL como gestor de bases de datos.
- MySQL Workbench para la ejecución de consultas y visualización de resultados.
- ERDPlus para la creacion de diagramas.
- LaTeX para la elaboración del reporte.
- Respaldo de la base de datos de Integración: `https://github.com/idkAriana/Bases-de-Datos-Disribuidas/blob/e2234b53481f2178afa2f0203b9a215ce8028519/Base%20de%20datos%20de%20Integraci%C3%B3n.sql`

4. Desarrollo

En esta práctica se trabajó con la integración de distintos esquemas de bases de datos relacionales pertenecientes a un contexto académico. Inicialmente, se proporcionaron tres esquemas locales diferentes: uno correspondiente a la información de Integrantes y Cuerpos Académicos, otro relacionado con Investigadores y sus Producciones Académicas, y uno más asociado a Profesores, Alumnos y Programas Académicos.

El objetivo principal fue diseñar un esquema global integrado, que unificara los datos de manera coherente mediante una nueva entidad central denominada Persona, respetando las relaciones existentes entre los diferentes elementos de cada esquema. Posteriormente, se construyeron nuevamente los esquemas locales, adaptándolos a la nueva estructura global, y se realizaron consultas de prueba para validar la correcta integración.

Esquemas Conceptuales Originales

4.1.1 Esquema Local Original 1: Integrantes y Cuerpos Académicos

Este esquema almacenaba información sobre integrantes de cuerpos académicos, sus líneas de investigación, y su relación con los cuerpos académicos.

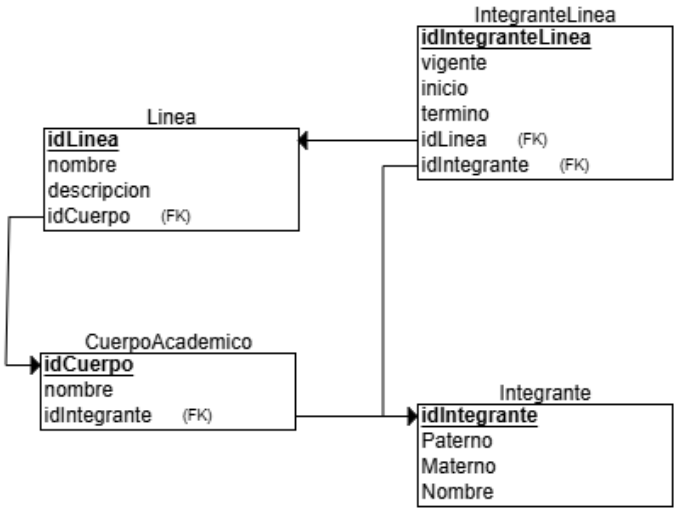

Figure 1: Esquema Original de Integrante

Cada integrante se asociaba directamente a un cuerpo académico, y luego a líneas de investigación vigentes, registrando fechas de inicio y término de participación.

4.1.2 Esquema Local Original 2: Investigadores y Producciones Académicas

Este esquema registraba información de investigadores, su producción académica, proyectos, adscripciones y formación académica.

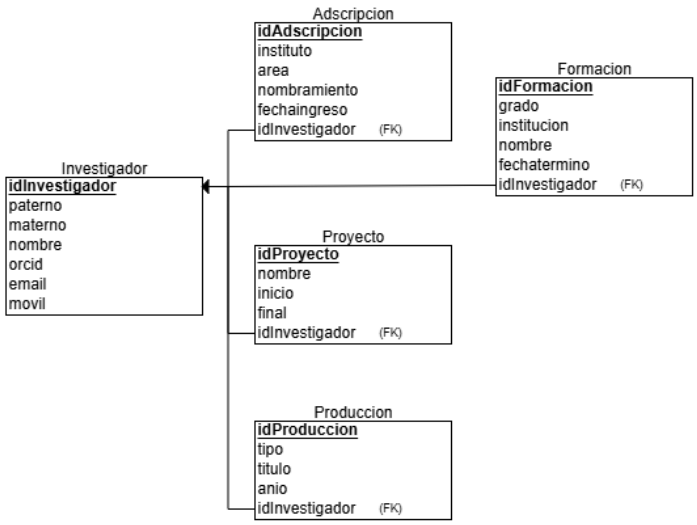

Figure 2: Esquema Original de Investigador

Cada investigador estaba ligado directamente a su producción científica, proyectos en los que participó, historial de adscripción institucional y trayectoria académica.

4.1.3 Esquema Local Original 3: Profesores, Alumnos y Programas Académicos

Este esquema representaba la estructura educativa: profesores, programas, asignaturas, cursos impartidos y la relación de los alumnos con los cursos .

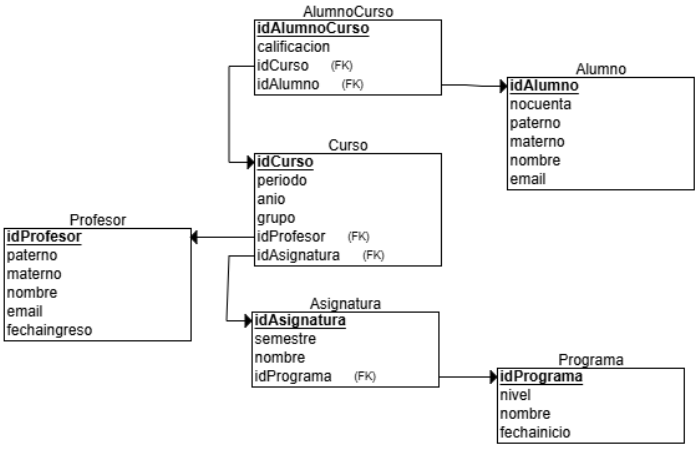

Figure 3: Esquema Original de de Profesor

Permitía asociar a los profesores con los cursos que impartían, dentro de programas académicos específicos, y registrar la participación y calificación de los alumnos.

Integración de los Esquemas

La integración de los esquemas se logró a través de los siguientes pasos:

1. **Creación de la Entidad Global "Persona":** Se detectó que Integrante, Investigador y Profesor compartían atributos comunes: nombre, paterno y materno. Para evitar redundancia, se creó la tabla Persona en el esquema global.

2. **Adaptación de Entidades Dependientes:** Integrante, Investigador y Profesor se modificaron para tener una clave foránea hacia Persona.

3. **Unificación y Normalización:** Se homologaron tipos de datos (por ejemplo, CHAR y VARCHAR en nombres) para mantener consistencia, y se ajustaron las relaciones entre las entidades para conservar la integridad referencial.

4. **Rediseño de Esquemas Locales:** Se actualizaron los esquemas locales para reflejar la integración, cada uno manteniendo solo la parte de la información que le corresponde, pero usando también la nueva entidad Persona.

Esquema Global

Se creó una nueva base de datos llamada GlobalUAEH, donde se integraron todas las entidades de los tres esquemas locales originales de manera armonizada.

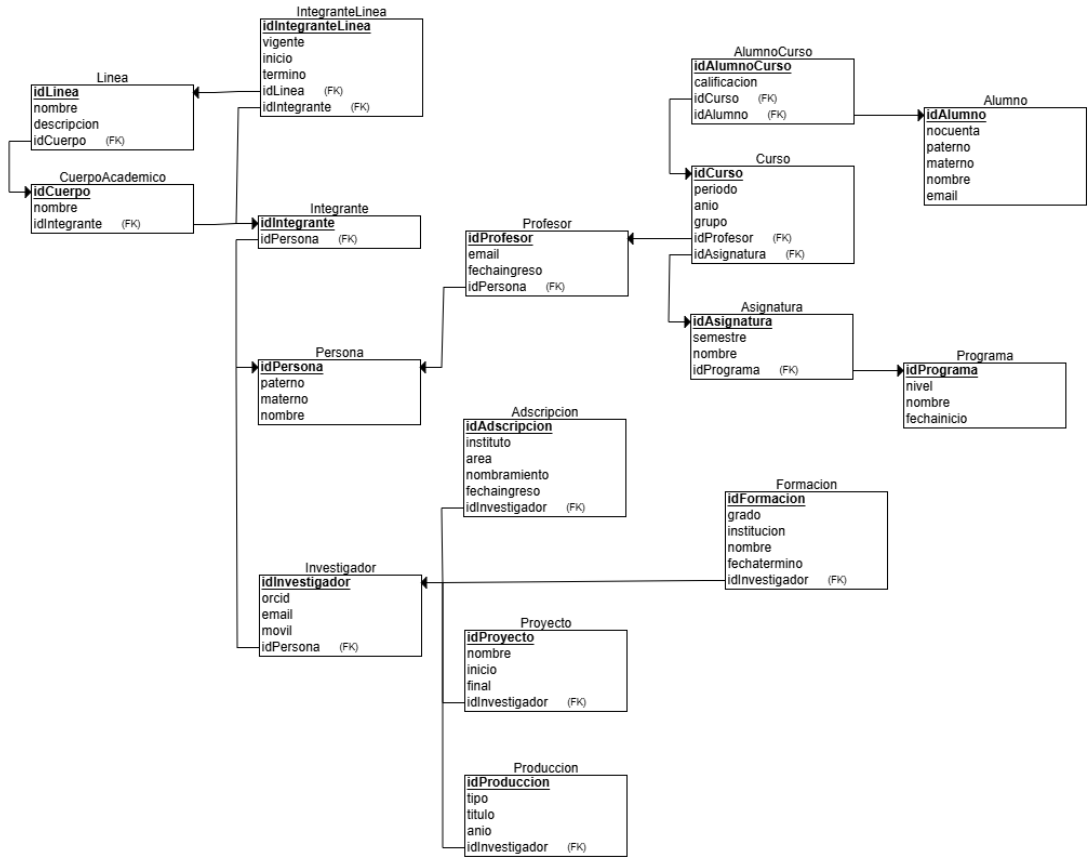

Figure 4: Esquema Global

El diseño incluye:

- La entidad central Persona.
- Relaciones adecuadas para Integrante, Investigador, y Profesor hacia Persona.
- Conservación de la estructura original de Proyectos, Producciones, Adscripciones, Formaciones, Programas, Asignaturas, Cursos, Alumnos y Relaciones Alumno-Curso.
- Una correcta normalización para evitar duplicidad de datos.

Base de Datos de Integración

El proceso ETL (Extract-Transform-Load) para integrar los esquemas de las bases de datos $Local_Integrante$, $Local_Investigado$

1. Extracción (Extract): En esta fase se extrajeron los datos relevantes de cada base de datos local. Se utilizaron consultas SQL simples para recuperar la información desde las tablas origen.

```sql
-- Base Local_Integrante
SELECT idPersona, paterno, materno, nombre FROM Local_Integrante.
    Persona;

-- Base Local_Investigador
SELECT idPersona, paterno, materno, nombre FROM Local_Investigador
    .Persona;

-- Base Local_Profesor
SELECT idPersona, paterno, materno, nombre FROM Local_Profesor.
    Persona;
```

Listing 1: Script MySQL

2. Transformación (Transform): En esta fase se aplicaron transformaciones necesarias para homogeneizar la estructura de los datos, corregir diferencias en tipos de datos, estandarizar nombres de atributos y eliminar duplicados.
3. Carga (Load): Finalmente, se cargaron los datos transformados en la base de datos de integración $Integracion_Proyecto, respetandolasnuevasllavesprimariasyrelacionesentrelasentidades$.

```sql
CREATE DATABASE Global_uaeh;
CREATE TABLE Programa
(
  idPrograma INT NOT NULL,
  nivel VARCHAR(25) NOT NULL,
  nombre VARCHAR(150) NOT NULL,
  fechainicio DATE NOT NULL,
  PRIMARY KEY (idPrograma)
);

CREATE TABLE Alumno
(
  idAlumno INT NOT NULL,
  nocuenta VARCHAR(25) NOT NULL,
  paterno VARCHAR(80) NOT NULL,
  materno VARCHAR(80) NOT NULL,
  nombre VARCHAR(80) NOT NULL,
  email VARCHAR(120) NOT NULL,
  PRIMARY KEY (idAlumno)
);

```

```
CREATE TABLE Persona
(
  idPersona INT NOT NULL,
  paterno CHAR(80) NOT NULL,
  materno CHAR(80) NOT NULL,
  nombre CHAR(120) NOT NULL,
  PRIMARY KEY (idPersona)
);

CREATE TABLE Integrante
(
  idIntegrante INT NOT NULL,
  idPersona INT NOT NULL,
  PRIMARY KEY (idIntegrante),
  FOREIGN KEY (idPersona) REFERENCES Persona(idPersona)
);

CREATE TABLE CuerpoAcademico
(
  idCuerpo INT NOT NULL,
  nombre VARCHAR(250) NOT NULL,
  idIntegrante INT NOT NULL,
  PRIMARY KEY (idCuerpo),
  FOREIGN KEY (idIntegrante) REFERENCES Integrante(idIntegrante)
);

CREATE TABLE Linea
(
  idLinea INT NOT NULL,
  nombre VARCHAR(120) NOT NULL,
  descripcion VARCHAR(500) NOT NULL,
  idCuerpo INT NOT NULL,
  PRIMARY KEY (idLinea),
  FOREIGN KEY (idCuerpo) REFERENCES CuerpoAcademico(idCuerpo)
);

CREATE TABLE Profesor
(
  idProfesor INT NOT NULL,
  email VARCHAR(120) NOT NULL,
  fechaingreso DATE NOT NULL,
  idPersona INT NOT NULL,
  PRIMARY KEY (idProfesor),
  FOREIGN KEY (idPersona) REFERENCES Persona(idPersona)
);

CREATE TABLE Investigador
(
  idInvestigador INT NOT NULL,
  orcid VARCHAR(30) NOT NULL,
  email VARCHAR(150) NOT NULL,
  movil VARCHAR(15) NOT NULL,
  idPersona INT NOT NULL,
  PRIMARY KEY (idInvestigador),
  FOREIGN KEY (idPersona) REFERENCES Persona(idPersona)
);

CREATE TABLE Proyecto
(
```

```
81                idProyecto INT NOT NULL,
82                nombre VARCHAR(250) NOT NULL,
83                inicio DATE NOT NULL,
84                final DATE NOT NULL,
85                idInvestigador INT NOT NULL,
86                PRIMARY KEY (idProyecto),
87                FOREIGN KEY (idInvestigador) REFERENCES Investigador(
                      idInvestigador)
88              );
89
90              CREATE TABLE Produccion
91              (
92                idProduccion INT NOT NULL,
93                tipo VARCHAR(60) NOT NULL,
94                titulo VARCHAR(250) NOT NULL,
95                anio INT NOT NULL,
96                idInvestigador INT NOT NULL,
97                PRIMARY KEY (idProduccion),
98                FOREIGN KEY (idInvestigador) REFERENCES Investigador(
                      idInvestigador)
99              );
100
101             CREATE TABLE Adscripcion
102             (
103               idAdscripcion INT NOT NULL,
104               instituto VARCHAR(80) NOT NULL,
105               area VARCHAR(150) NOT NULL,
106               nombramiento VARCHAR(10) NOT NULL,
107               fechaingreso DATETIME NOT NULL,
108               idInvestigador INT NOT NULL,
109               PRIMARY KEY (idAdscripcion),
110               FOREIGN KEY (idInvestigador) REFERENCES Investigador(
                      idInvestigador)
111             );
112
113             CREATE TABLE Formacion
114             (
115               idFormacion INT NOT NULL,
116               grado VARCHAR(18) NOT NULL,
117               institucion VARCHAR(70) NOT NULL,
118               nombre VARCHAR(120) NOT NULL,
119               fechatermino DATE NOT NULL,
120               idInvestigador INT NOT NULL,
121               PRIMARY KEY (idFormacion),
122               FOREIGN KEY (idInvestigador) REFERENCES Investigador(
                      idInvestigador)
123             );
124
125             CREATE TABLE Asignatura
126             (
127               idAsignatura INT NOT NULL,
128               semestre INT NOT NULL,
129               nombre VARCHAR(120) NOT NULL,
130               idPrograma INT NOT NULL,
131               PRIMARY KEY (idAsignatura),
132               FOREIGN KEY (idPrograma) REFERENCES Programa(idPrograma)
133             );
134
135             CREATE TABLE IntegranteLinea
```

```
            (
              idIntegranteLinea INT NOT NULL,
              vigente BOOLEAN NOT NULL,
              inicio DATE NOT NULL,
              termino DATE NOT NULL,
              idLinea INT NOT NULL,
              idIntegrante INT NOT NULL,
              PRIMARY KEY (idIntegranteLinea),
              FOREIGN KEY (idLinea) REFERENCES Linea(idLinea),
              FOREIGN KEY (idIntegrante) REFERENCES Integrante(idIntegrante)
            );

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

            CREATE TABLE AlumnoCurso
            (
              idAlumnoCurso INT NOT NULL,
              calificacion FLOAT NOT NULL,
              idCurso INT NOT NULL,
              idAlumno INT NOT NULL,
              PRIMARY KEY (idAlumnoCurso),
              FOREIGN KEY (idCurso) REFERENCES Curso(idCurso),
              FOREIGN KEY (idAlumno) REFERENCES Alumno(idAlumno)
            );
```

Listing 2: Script MySQL para Integrante

Esquemas locales actualizados

Tras la integración global, cada esquema local fue ajustado para alinearse al modelo unificado. Se conservaron las funciones principales de cada uno, pero se reestructuraron para incluir la referencia a la nueva entidad Persona y asegurar la integridad de los datos.

4.5.1 Esquema Local Actualizado 1: Integrante

Los cambios principales son la eliminación de los atributos personales del Integrante y el agregado de la clave foránea idPersona para identificar a cada integrante.

```
CREATE DATABASE Local_Integrante;
CREATE TABLE Persona
(
  idPersona INT NOT NULL,
  paterno CHAR(80) NOT NULL,
  materno CHAR(80) NOT NULL,
  nombre CHAR(120) NOT NULL,
  PRIMARY KEY (idPersona)
);
```

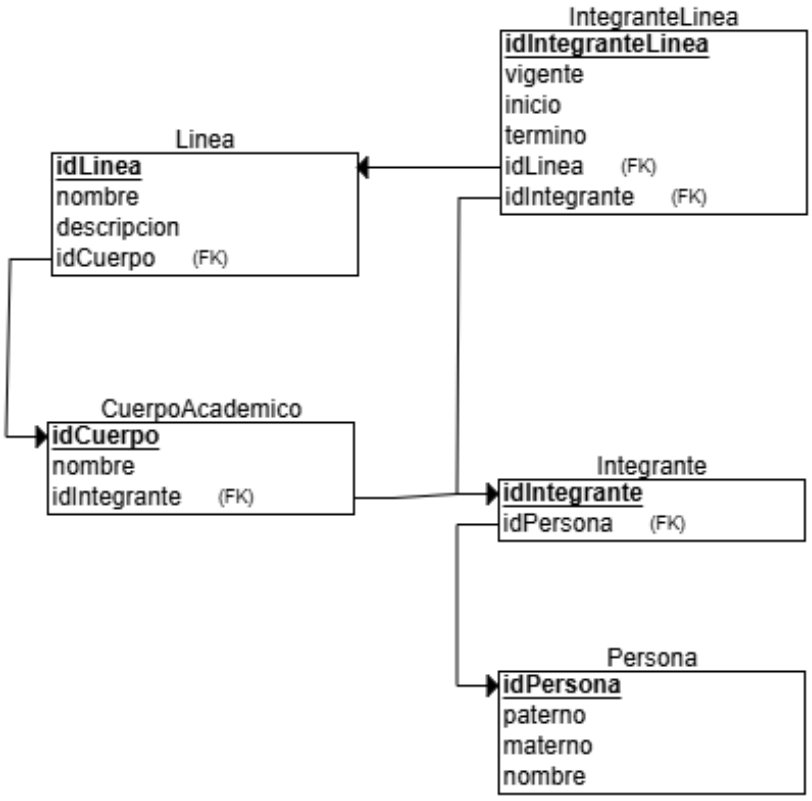

Figure 5: Esquema Actualizado de Integrante

```sql

CREATE TABLE Integrante
(
  idIntegrante INT NOT NULL,
  idPersona INT NOT NULL,
  PRIMARY KEY (idIntegrante),
  FOREIGN KEY (idPersona) REFERENCES Persona(idPersona)
);

CREATE TABLE CuerpoAcademico
(
  idCuerpo INT NOT NULL,
  nombre VARCHAR(250) NOT NULL,
  idIntegrante INT NOT NULL,
  PRIMARY KEY (idCuerpo),
  FOREIGN KEY (idIntegrante) REFERENCES Integrante(idIntegrante)
);

CREATE TABLE Linea
(
  idLinea INT NOT NULL,
  nombre VARCHAR(120) NOT NULL,
  descripcion VARCHAR(500) NOT NULL,
  idCuerpo INT NOT NULL,
```

```
    PRIMARY KEY (idLinea),
    FOREIGN KEY (idCuerpo) REFERENCES CuerpoAcademico(idCuerpo)
);

CREATE TABLE IntegranteLinea
(
    idIntegranteLinea INT NOT NULL,
    vigente BOOLEAN NOT NULL,
    inicio DATE NOT NULL,
    termino DATE NOT NULL,
    idLinea INT NOT NULL,
    idIntegrante INT NOT NULL,
    PRIMARY KEY (idIntegranteLinea),
    FOREIGN KEY (idLinea) REFERENCES Linea(idLinea),
    FOREIGN KEY (idIntegrante) REFERENCES Integrante(idIntegrante)
);
```

Listing 3: Script MySQL para Integrante

4.5.2 Esquema Local Actualizado 2: Investigador

Los cambios principales son la eliminación de la información redundante de nombres y el agregado de la clave foránea idPersona para enlazar cada investigador con su registro personal.

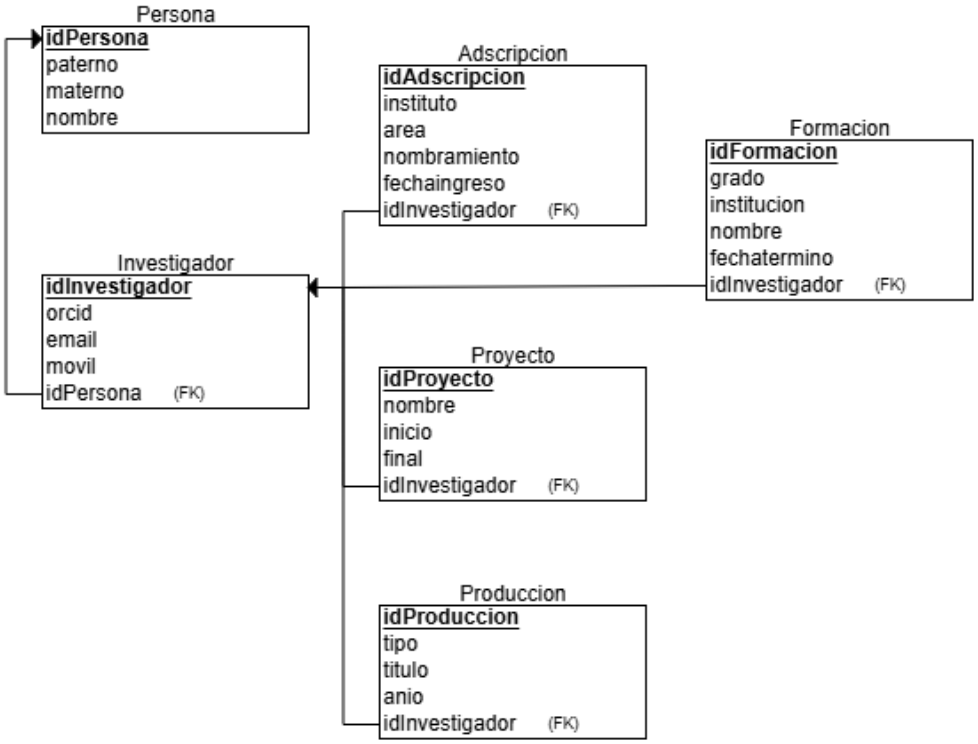

Figure 6: Esquema Actualizado de Investigador

```sql
CREATE DATABASE Local_Investigador;
CREATE TABLE Persona
(
  idPersona INT NOT NULL,
  paterno CHAR(80) NOT NULL,
  materno CHAR(80) NOT NULL,
  nombre CHAR(120) NOT NULL,
  PRIMARY KEY (idPersona)
);

CREATE TABLE Investigador
(
  idInvestigador INT NOT NULL,
  orcid VARCHAR(30) NOT NULL,
  email VARCHAR(150) NOT NULL,
  movil VARCHAR(15) NOT NULL,
  idPersona INT NOT NULL,
  PRIMARY KEY (idInvestigador),
  FOREIGN KEY (idPersona) REFERENCES Persona(idPersona)
);

CREATE TABLE Proyecto
(
  idProyecto INT NOT NULL,
  nombre VARCHAR(250) NOT NULL,
  inicio DATE NOT NULL,
  final DATE NOT NULL,
```

```
    idInvestigador INT NOT NULL,
    PRIMARY KEY (idProyecto),
    FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)
);

CREATE TABLE Produccion
(
    idProduccion INT NOT NULL,
    tipo VARCHAR(60) NOT NULL,
    titulo VARCHAR(250) NOT NULL,
    anio INT NOT NULL,
    idInvestigador INT NOT NULL,
    PRIMARY KEY (idProduccion),
    FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)
);

CREATE TABLE Adscripcion
(
    idAdscripcion INT NOT NULL,
    instituto VARCHAR(80) NOT NULL,
    area VARCHAR(150) NOT NULL,
    nombramiento VARCHAR(10) NOT NULL,
    fechaingreso DATETIME NOT NULL,
    idInvestigador INT NOT NULL,
    PRIMARY KEY (idAdscripcion),
    FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)
);

CREATE TABLE Formacion
(
    idFormacion INT NOT NULL,
    grado VARCHAR(18) NOT NULL,
    institucion VARCHAR(70) NOT NULL,
    nombre VARCHAR(120) NOT NULL,
    fechatermino DATE NOT NULL,
    idInvestigador INT NOT NULL,
    PRIMARY KEY (idFormacion),
    FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)
);
```

Listing 4: Script MySQL para Investigador

4.5.3 Esquema Local Actualizado 3: Profesor

Los cambios principales son la eliminación de atributos personales directos del profesor y el agregado de la clave foránea idPersona para relacionar cada profesor con sus datos personales.

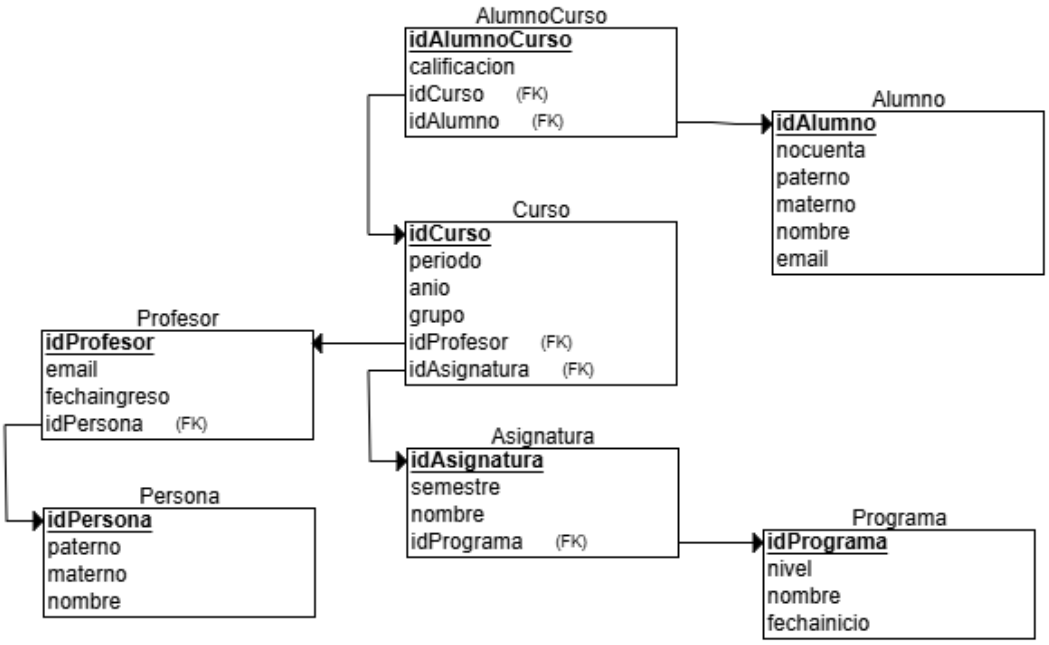

Figure 7: Esquema Actualizado de Profesor

```sql
CREATE DATABASE Local_Profesor;
CREATE TABLE Programa
(
  idPrograma INT NOT NULL,
  nivel VARCHAR(25) NOT NULL,
  nombre VARCHAR(150) NOT NULL,
  fechainicio DATE NOT NULL,
  PRIMARY KEY (idPrograma)
);

CREATE TABLE Alumno
(
  idAlumno INT NOT NULL,
  nocuenta VARCHAR(25) NOT NULL,
  paterno VARCHAR(80) NOT NULL,
  materno VARCHAR(80) NOT NULL,
  nombre VARCHAR(80) NOT NULL,
  email VARCHAR(120) NOT NULL,
  PRIMARY KEY (idAlumno)
);

CREATE TABLE Persona
(
  idPersona INT NOT NULL,
  paterno CHAR(80) NOT NULL,
  materno CHAR(80) NOT NULL,
  nombre CHAR(120) NOT NULL,
  PRIMARY KEY (idPersona)
);

```

```
CREATE TABLE Profesor
(
  idProfesor INT NOT NULL,
  email VARCHAR(120) NOT NULL,
  fechaingreso DATE NOT NULL,
  idPersona INT NOT NULL,
  PRIMARY KEY (idProfesor),
  FOREIGN KEY (idPersona) REFERENCES Persona(idPersona)
);

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

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

CREATE TABLE AlumnoCurso
(
  idAlumnoCurso INT NOT NULL,
  calificacion FLOAT NOT NULL,
  idCurso INT NOT NULL,
  idAlumno INT NOT NULL,
  PRIMARY KEY (idAlumnoCurso),
  FOREIGN KEY (idCurso) REFERENCES Curso(idCurso),
  FOREIGN KEY (idAlumno) REFERENCES Alumno(idAlumno)
);
```

Listing 5: Script MySQL para Profesor

Consultas de Validación

Para asegurar la correcta integración de los esquemas, se realizaron diferentes consultas de prueba sobre las bases de datos locales.

4.6.1 Consulta 1

Ver datos básicos de Personas que sean Integrantes, Investigadores y Profesores. Esto une las tres bases a través de Persona. Muestra si una persona es Integrante, Investigador o Profesor.

```
SELECT
    I.idIntegrante,
    Inv.idInvestigador,
    P.idProfesor,
    PI.nombre AS NombrePersona,
    PI.paterno AS ApellidoPaterno,
    PI.materno AS ApellidoMaterno
FROM Local_Integrante.Persona PI
LEFT JOIN Local_Integrante.Integrante I ON PI.idPersona = I.idPersona
LEFT JOIN Local_Investigador.Persona PInv ON PI.idPersona = PInv.
    idPersona
LEFT JOIN Local_Investigador.Investigador Inv ON PInv.idPersona = Inv.
    idPersona
LEFT JOIN Local_Profesor.Persona PProf ON PI.idPersona = PProf.
    idPersona
LEFT JOIN Local_Profesor.Profesor P ON PProf.idPersona = P.idPersona;
```

Listing 6: Script MySQL

idIntegrante	idInvestigador	idProfesor	NombrePersona	ApellidoPaterno	ApellidoMaterno
1	1	1	Juan Carlos	García	López
2	2	2	María Elena	Martínez	Sánchez
3	3	3	Luis Fernando	Rodríguez	Gómez
4	4	4	Ana María	Hernández	Pérez
5	5	5	José Antonio	González	Ramírez
6	6	6	Laura Isabel	Fernández	Torres
7	7	7	Carlos Alberto	Díaz	Flores
8	8	8	Patricia Guadalupe	Vázquez	Cruz
9	9	9	Miguel Ángel	Moreno	Ruiz
10	10	10	Sofía Alejandra	Álvarez	Mendoza
11	11	11	David Alejandro	Jiménez	Ortega
12	12	12	Gabriela Fernanda	Castillo	Herrera
13	13	13	Jorge Luis	Rivera	Vargas
14	14	14	Claudia Beatriz	Romero	Medina
15	15	15	Francisco Javier	Soto	Reyes
16	16	16	Adriana Elizabeth	Morales	Castro
17	17	17	Ricardo Manuel	Ortega	Guerrero
18	18	18	Verónica Patricia	Delgado	Juárez
19	19	19	Roberto Carlos	Silva	Núñez
20	20	20	Daniela Carolina	Rojas	Salazar
21	21	21	Arturo Benjamín	Mendoza	Ortiz
22	22	22	Lucía Margarita	Aguilar	Chávez

Figure 8: Resultado de la consulta

4.6.2 Consulta 2

Consultar todos los Proyectos de Investigadores, junto con su información de Integrante. Consulta Proyectos relacionados a un Integrante (comprobando que son la misma persona).

```
SELECT
    Inv.idInvestigador,
    PI.nombre AS NombreInvestigador,
    Proy.nombre AS NombreProyecto,
    Proy.inicio,
    Proy.final
FROM Local_Investigador.Persona PI
INNER JOIN Local_Investigador.Investigador Inv ON PI.idPersona = Inv.
    idPersona
INNER JOIN Local_Investigador.Proyecto Proy ON Inv.idInvestigador =
    Proy.idInvestigador
INNER JOIN Local_Integrante.Persona PIn ON PI.idPersona = PIn.
    idPersona
INNER JOIN Local_Integrante.Integrante Intg ON PIn.idPersona = Intg.
    idPersona;
```

Listing 7: Script MySQL

idInvestigador	NombreInvestigador	NombreProyecto	inicio	final
1	Juan Carlos	Desarrollo de algoritmos de IA para diagnóstico ...	2020-01-15	2023-12-15
2	María Elena	Sistemas autónomos para agricultura de precisión	2019-06-20	2022-06-20
3	Luis Fernando	Blockchain para transacciones seguras en gobie...	2021-03-10	2024-03-10
4	Ana María	Realidad virtual para terapia de fobias	2020-09-05	2023-09-05
5	José Antonio	Modelado predictivo de pandemias	2019-11-15	2022-11-15
6	Laura Isabel	Robótica colaborativa para manufactura	2021-01-20	2024-01-20
7	Carlos Alberto	Computación cuántica aplicada a criptografía	2020-05-12	2023-05-12
8	Patricia Guadalupe	Plataforma de aprendizaje adaptativo	2019-08-30	2022-08-30
9	Miguel Ángel	Análisis de sentimientos en redes sociales	2021-02-18	2024-02-18
10	Sofía Alejandra	Sistemas de recomendación personalizados	2020-07-22	2023-07-22
11	David Alejandro	Detección temprana de cáncer con IA	2019-10-10	2022-10-10
12	Gabriela Fernanda	Ciudades inteligentes y movilidad urbana	2021-04-05	2024-04-05
13	Jorge Luis	Energías renovables y optimización	2020-02-28	2023-02-28
14	Claudia Beatriz	Telemedicina para zonas rurales	2019-07-15	2022-07-15
15	Francisco Javier	Nanomateriales para almacenamiento energético	2021-05-20	2024-05-20
16	Adriana Elizabeth	Biotecnología para tratamiento de aguas	2020-03-10	2023-03-10
17	Ricardo Manuel	Modelos climáticos regionales	2019-09-25	2022-09-25
18	Verónica Patricia	Sistemas expertos para diagnóstico industrial	2021-06-15	2024-06-15
19	Roberto Carlos	Realidad aumentada para mantenimiento	2020-04-18	2023-04-18
20	Daniela Carolina	Algoritmos genéticos para logística	2019-12-01	2022-12-01
21	Arturo Benjamín	Interfaces cerebro-computadora	2021-07-10	2024-07-10
22	Lucía Margarita	Seguridad en IoT industrial	2020-01-30	2023-01-30
23	Oscar Eduardo	Deep learning para imágenes médicas	2019-05-22	2022-05-22

Figure 9: Resultado de la consulta

4.6.3 Consulta 3

Consultar Profesores que además tienen Producción Académica como Investigadores. Encuentra profesores que también son investigadores con producción académica registrada.

```
SELECT
    Prof.email AS EmailProfesor,
    Inv.orcid AS ORCID,
    Prod.titulo AS TituloProduccion,
    Prod.anio AS Anio
FROM Local_Profesor.Persona PProf
```

```
INNER JOIN Local_Profesor.Profesor Prof ON PProf.idPersona = Prof.
    idPersona
INNER JOIN Local_Investigador.Persona PInv ON PProf.idPersona = PInv.
    idPersona
INNER JOIN Local_Investigador.Investigador Inv ON PInv.idPersona = Inv
    .idPersona
INNER JOIN Local_Investigador.Produccion Prod ON Inv.idInvestigador =
    Prod.idInvestigador;
```

Listing 8: Script MySQL

	EmailProfesor	ORCID	TituloProduccion	Año
▶	juan.garcia@universidad.edu	0000-0001-5000-0001	Algoritmos de deep learning para diagnóstico te...	2020
	maria.martinez@universidad.edu	0000-0001-5000-0002	Inteligencia Artificial: Fundamentos y Aplicaciones	2019
	luis.rodriguez@universidad.edu	0000-0001-5000-0003	Blockchain en el sector financiero	2021
	ana.hernandez@universidad.edu	0000-0001-5000-0004	Terapias de realidad virtual para fobias específicas	2020
	jose.gonzalez@universidad.edu	0000-0001-5000-0005	Modelos predictivos para brotes epidémicos	2019
	laura.fernandez@universidad.edu	0000-0001-5000-0006	Sistema robótico colaborativo para ensamblaje	2021
	carlos.diaz@universidad.edu	0000-0001-5000-0007	Criptografía post-cuántica	2020
	patricia.vazquez@universidad.edu	0000-0001-5000-0008	Plataformas de e-learning adaptativo	2019
	miguel.moreno@universidad.edu	0000-0001-5000-0009	Análisis de sentimientos en Twitter durante crisis	2021
	sofia.alvarez@universidad.edu	0000-0001-5000-0010	Sistemas de recomendación basados en grafos	2020
	david.jimenez@universidad.edu	0000-0001-5000-0011	Redes neuronales convolucionales para detecci...	2019
	gabriela.castillo@universidad.edu	0000-0001-5000-0012	Ciudades Inteligentes: Tecnologías y Desafíos	2021
	jorge.rivera@universidad.edu	0000-0001-5000-0013	Optimización de redes de energía renovable	2020
	claudia.romero@universidad.edu	0000-0001-5000-0014	Sistema de telemedicina para zonas remotas	2019
	francisco.soto@universidad.edu	0000-0001-5000-0015	Nanomateriales para almacenamiento de hidróg...	2021
	adriana.morales@universidad.edu	0000-0001-5000-0016	Biorremediación de aguas contaminadas	2020
	ricardo.ortega@universidad.edu	0000-0001-5000-0017	Modelado Climático Regional	2019
	veronica.delgado@universidad.edu	0000-0001-5000-0018	Sistemas expertos para mantenimiento predictivo	2021
	roberto.silva@universidad.edu	0000-0001-5000-0019	Realidad aumentada en procesos industriales	2020
	daniela.rojas@universidad.edu	0000-0001-5000-0020	Algoritmos genéticos para routing logístico	2019
	arturo.mendoza@universidad.edu	0000-0001-5000-0021	Interfaz cerebro-computadora no invasiva	2021
	lucia.aguilar@universidad.edu	0000-0001-5000-0022	Seguridad en IoT industrial	2020
	oscar.vega@universidad.edu	0000-0001-5000-0023	Deep Learning en Imagenología Médica	2019

Figure 10: Resultado de la consulta

4.6.4 Consulta 4

Muestra la línea de investigación y el correo de profesor, si lo tiene.

```
SELECT
    Per.nombre AS Nombre,
    Per.paterno AS Paterno,
    Per.materno AS Materno,
    L.nombre AS LineaInvestigacion,
    Prof.email AS EmailProfesor
FROM Local_Integrante.Persona Per
INNER JOIN Local_Integrante.Integrante I ON Per.idPersona = I.
    idPersona
INNER JOIN Local_Integrante.IntegranteLinea IL ON I.idIntegrante = IL.
    idIntegrante
INNER JOIN Local_Integrante.Linea L ON IL.idLinea = L.idLinea
LEFT JOIN Local_Profesor.Persona PProf ON Per.idPersona = PProf.
    idPersona
LEFT JOIN Local_Profesor.Profesor Prof ON PProf.idPersona = Prof.
    idPersona;
```

Listing 9: Script MySQL

Nombre	Paterno	Materno	LineaInvestigacion	EmailProfesor
Juan Carlos	García	López	Aprendizaje Automático	juan.garcia@universidad.edu
María Elena	Martínez	Sánchez	Redes Neuronales Profundas	maria.martinez@universidad.edu
Luis Fernando	Rodríguez	Gómez	Procesamiento de Lenguaje Natural	luis.rodriguez@universidad.edu
Ana María	Hernández	Pérez	Visión por Computadora	ana.hernandez@universidad.edu
José Antonio	González	Ramírez	Robótica Autónoma	jose.gonzalez@universidad.edu
Laura Isabel	Fernández	Torres	Internet de las Cosas	laura.fernandez@universidad.edu
Carlos Alberto	Díaz	Flores	Computación en la Nube	carlos.diaz@universidad.edu
Patricia Guadalupe	Vázquez	Cruz	Seguridad Informática	patricia.vazquez@universidad.edu
Miguel Ángel	Moreno	Ruiz	Blockchain	miguel.moreno@universidad.edu
Sofía Alejandra	Álvarez	Mendoza	Realidad Virtual	sofia.alvarez@universidad.edu
David Alejandro	Jiménez	Ortega	Bioinformática	david.jimenez@universidad.edu
Gabriela Fernanda	Castillo	Herrera	Computación Cuántica	gabriela.castillo@universidad.edu
Jorge Luis	Rivera	Vargas	Sistemas Distribuidos	jorge.rivera@universidad.edu
Claudia Beatriz	Romero	Medina	Ingeniería de Software	claudia.romero@universidad.edu
Francisco Javier	Soto	Reyes	Interacción Humano-Computadora	francisco.soto@universidad.edu
Adriana Elizabeth	Morales	Castro	Big Data	adriana.morales@universidad.edu
Ricardo Manuel	Ortega	Guerrero	Computación Gráfica	ricardo.ortega@universidad.edu
Verónica Patricia	Delgado	Juárez	Sistemas Embebidos	veronica.delgado@universidad.edu
Roberto Carlos	Silva	Núñez	Computación Móvil	roberto.silva@universidad.edu
Daniela Carolina	Rojas	Salazar	Arquitectura de Software	daniela.rojas@universidad.edu
Arturo Benjamín	Mendoza	Ortiz	Pruebas de Software	arturo.mendoza@universidad.edu
Lucía Margarita	Aguilar	Chávez	DevOps	lucia.aguilar@universidad.edu
Oscar Eduardo	Vega	Contreras	Microservicios	oscar.vega@universidad.edu

Figure 11: Resultado de la consulta

5. Conclusiones

El manejo eficiente de datos en bases de datos distribuidas es crucial para garantizar la integridad, disponibilidad y rendimiento del sistema. La fragmentación vertical permite distribuir los datos de manera optimizada, mejorando la eficiencia de acceso. Los procesos ETL facilitan la integración y transformación de datos desde distintas fuentes, permitiendo una gestión más estructurada de la información.

La integración bottom-up de los tres esquemas locales ha resultado en un modelo global coherente que preserva la información y relaciones de los sistemas originales mientras elimina redundancias y mejora la consistencia. El proceso demostró la importancia de identificar entidades equivalentes en diferentes esquemas la necesidad de establecer convenciones de nombres y estructuras consistentes y el valor de un modelo unificado de personas para distintos roles académicos.

El esquema global resultante permite una visión integral de la institución, facilitando tanto la gestión administrativa como la investigación académica. Futuras mejoras podrían incluir:

- Mayor normalización de atributos de fechas.
- Estándares más estrictos para identificadores.
- Mecanismos para preservar cierta autonomía local.

Referencias Bibliográficas

References

[1] Silberschatz, A., Korth, H. F., Sudarshan, S. (**2020**). Conceptos de sistemas de bases de datos. (7th ed.). *McGraw-Hill*.

[2] Elmasri, R., Navathe, S. (**2015**). Sistemas de bases de datos. (7th ed.). *Pearson*.

[3] Documentación oficial de MySQL: `https://dev.mysql.com/doc/`

[4] Coronel, C., Morris, S. (**2017**). Diseño de bases de datos: Un enfoque práctico. *Cengage Learning*.

[5] Batini, C., Lenzerini, M., Navathe, S. B. (**1986**). A comparative analysis of methodologies for database schema integration. ACM computing surveys. *CSUR*.

[6] Özsu, M. T., Valduriez, P (**2020**). Principles of distributed database systems. *Springer*.