



UNIVERSIDAD NACIONAL DE LOJA

FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES

Computación

INTEGRANTES:

- Ariana Alejandra Córdova Betancourt
- Cristhian Alexander Dávila Sari

CICLO:

3 "A"

DOCENTE:

ING. René Guamán Quinche

ASIGNATURA:

Base de Datos

FECHA:

30 de julio del 2025

A. Datos Informativos		
Asignatura: Base de datos	Semestre: Tercero	Paralelo: A
Docente Supervisor: René Guamán-Quinche	Periodo Académico: Marzo - Agosto de 2025	
Estudiantes: Cristhian Alexander Dávila Sari, Ariana Alejandra Córdova Betancourt		
Periodo de prácticas: abril-agosto 2025		

2. Introducción

Las prácticas preprofesionales tuvieron como objetivo aplicar los conocimientos adquiridos en la asignatura de Base de Datos, permitiendo fortalecer nuestras habilidades técnicas y profesionales como estudiantes de la carrera de Computación. Esta experiencia nos brindó la oportunidad de trabajar en equipo, utilizar herramientas tecnológicas actuales y desarrollar soluciones orientadas a necesidades reales.

Las actividades se realizaron en el área de Computación de la Universidad Nacional, donde participamos en el diseño e implementación de la base de datos para un sistema de gestión de proyectos académicos, el cual involucró a docentes, estudiantes y procesos administrativos.

En este contexto, las bases de datos fueron fundamentales para estructurar y gestionar la información del sistema. Permitieron almacenar los datos de forma organizada, definir relaciones entre entidades como usuarios, proyectos y tareas, y realizar consultas eficientes mediante SQL. Gracias a ello, se garantizó la integridad, coherencia y funcionalidad del sistema desarrollado.

3. Objetivos

3.1 Objetivo General

Diseñar, modelar e implementar una base de datos para un sistema de gestión de proyectos académicos, aplicando conocimientos de la asignatura de Base de Datos, mediante el uso de herramientas como el diagrama entidad-relación, el modelo relacional y la implementación de sentencias SQL, con el fin de garantizar la integridad, organización y eficiencia en el manejo de la información.

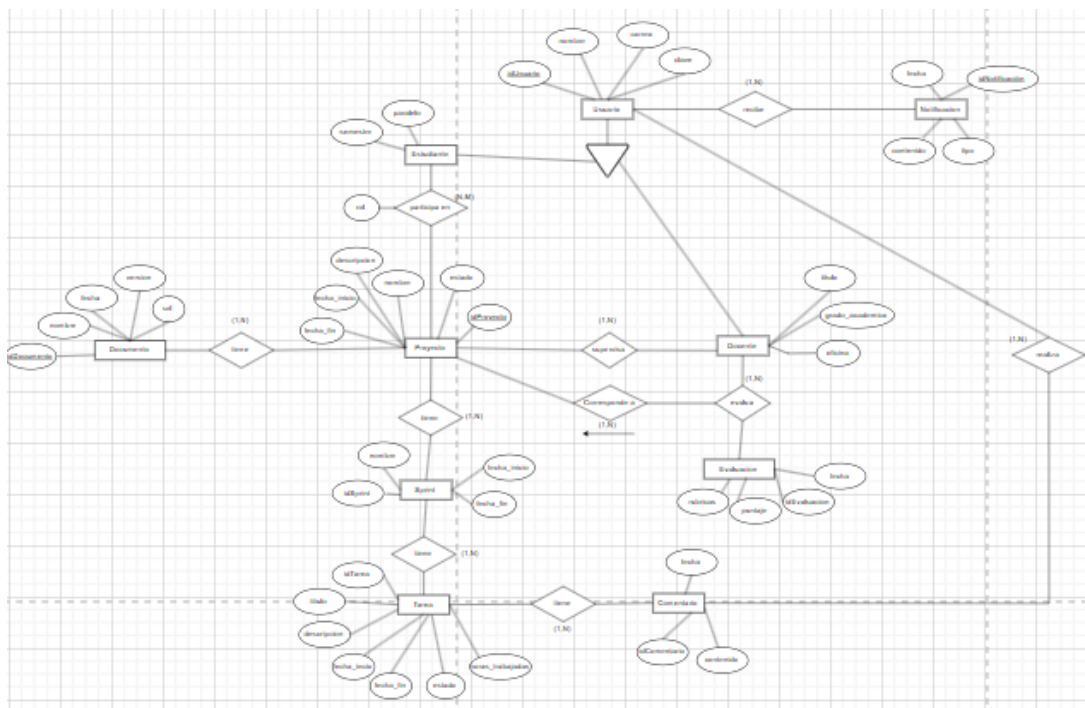
3.2 Objetivos Específicos

- Elaborar el diagrama entidad-relación para representar gráficamente las entidades y relaciones del sistema.
- Transformar el modelo entidad-relación en un modelo relacional optimizado.
- Implementar las estructuras de base de datos utilizando sentencias SQL.
- Crear consultas SQL para la recuperación eficiente de información.
- Diseñar vistas que faciliten el acceso a datos específicos y relevantes.
- Validar la integridad referencial mediante el uso de claves primarias y foráneas.
- Evaluar el funcionamiento de la base de datos a través de casos prácticos y consultas simuladas.

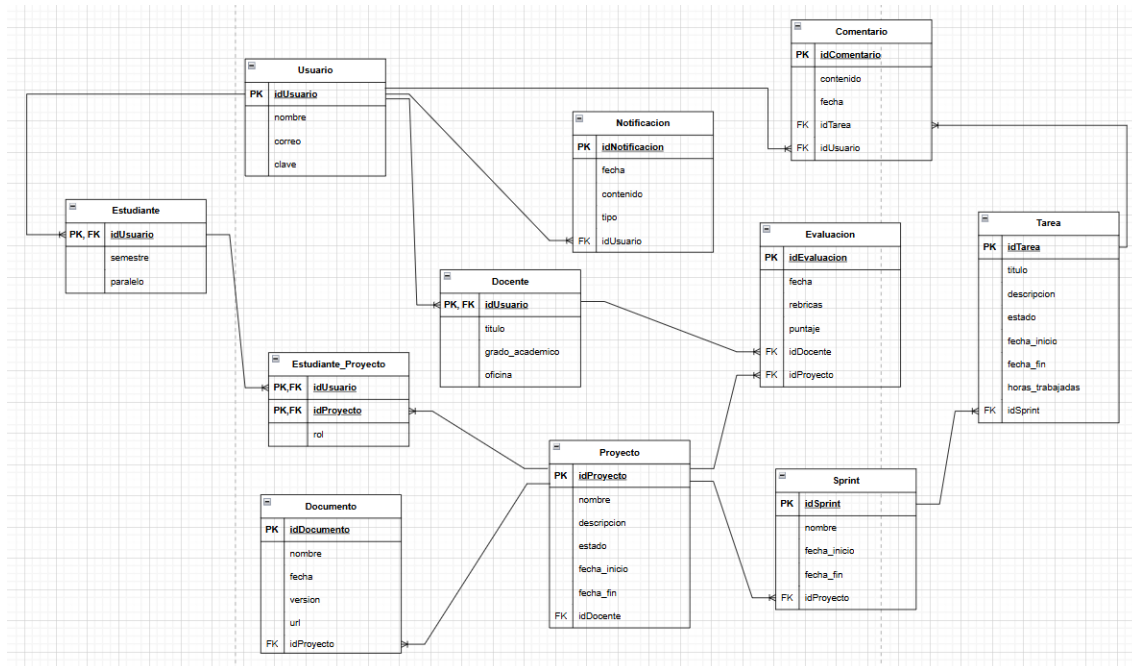
4. Actividades Realizadas

Desglose detallado de las tareas ejecutadas relacionadas con bases de datos:

- Modelo entidad – relación



- Modelo Relacional: traducción del modelo entidad - relación



- Detalle de normalización de la base de datos

Durante el diseño de la base de datos del sistema de gestión de proyectos académicos se aplicaron las tres primeras formas normales con el objetivo de asegurar la integridad, consistencia y eficiencia en el almacenamiento de la información.

Primera Forma Normal (1FN):

Se eliminó la presencia de atributos multivaluados o repetidos. Todas las columnas de cada tabla contienen valores atómicos, es decir, indivisibles. Se garantiza que cada campo contenga un solo valor y que todas las filas sean únicas.

Segunda Forma Normal (2FN):

Se eliminaron dependencias parciales asegurando que todos los atributos no clave dependan completamente de la clave primaria de su tabla. Las tablas con claves compuestas fueron estructuradas para que ningún atributo dependa solo de una parte de la clave.

Tercera Forma Normal (3FN):

Se eliminaron dependencias transitivas entre atributos no clave. Se garantizó que cada atributo no clave dependa únicamente de la clave primaria, evitando que un atributo dependa de otro atributo no clave dentro de la misma tabla.

- Diseño físico: Uso del SGBD (MySQL)

- Creación de de base de datos, tabla

```
CREATE DATABASE SistemaGestionProyectos;
CREATE TABLE Usuario (
  idUsuario INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR (50) NOT NULL,
  correo VARCHAR (50) NOT NULL,
  clave VARCHAR (30) NOT NULL
);
```

```
CREATE TABLE Estudiante (  
  idUsuario INT PRIMARY KEY,  
  paralelo VARCHAR (10),  
  semestre INT,  
  FOREIGN KEY (idUsuario) REFERENCES Usuario (idUsuario)  
);
```

```
CREATE TABLE Docente (  
  idUsuario INT PRIMARY KEY,  
  titulo VARCHAR (50),  
  grado_academico VARCHAR (50),  
  oficina VARCHAR (50),  
  FOREIGN KEY (idUsuario) REFERENCES Usuario (idUsuario)  
);
```

```
CREATE TABLE Notificacion (  
  idNotificacion INT AUTO_INCREMENT PRIMARY KEY,  
  fecha DATE NOT NULL,  
  contenido TEXT NOT NULL,  
  tipo VARCHAR (50) NOT NULL,  
  idUsuario INT NOT NULL,  
  FOREIGN KEY (idUsuario) REFERENCES Usuario (idUsuario)  
);
```

```
CREATE TABLE Proyecto (  
  idProyecto INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR (50) NOT NULL,  
  descripción TEXT,  
  estado VARCHAR (40),  
  fecha_inicio DATE NOT NULL,  
  fecha_fin DATE NOT NULL,  
  idDocente INT NOT NULL,  
  FOREIGN KEY (idDocente) REFERENCES Docente (idUsuario)  
);
```

```
CREATE TABLE Estudiante_Proyecto (  
  idUsuario INT,  
  idProyecto INT,  
  rol VARCHAR (50),  
  PRIMARY KEY (idUsuario, idProyecto),  
  FOREIGN KEY (idUsuario) REFERENCES Estudiante (idUsuario),  
  FOREIGN KEY (idProyecto) REFERENCES Proyecto (idProyecto)  
);
```

```
CREATE TABLE Documento (  
  idDocumento INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR (50) NOT NULL,  
  fecha DATE NOT NULL,  
  version VARCHAR (50) NOT NULL,  
  url VARCHAR (150) NOT NULL,  
  idProyecto INT NOT NULL,  
  FOREIGN KEY (idProyecto) REFERENCES Proyecto (idProyecto)  
);
```

```
CREATE TABLE Sprint (  
  idSprint INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR (50) NOT NULL,  
  fecha_inicio DATE NOT NULL,  
  fecha_fin DATE NOT NULL,  
  idProyecto INT NOT NULL,  
  FOREIGN KEY (idProyecto) REFERENCES Proyecto (idProyecto)  
);
```

```
CREATE TABLE Tarea (  
  idTarea INT AUTO_INCREMENT PRIMARY KEY,  
  descripcion VARCHAR (50),  
  estado VARCHAR (50) NOT NULL,  
  fecha_inicio DATE NOT NULL,  
  fecha_fin DATE NOT NULL,  
  horas_trabajadas DECIMAL,  
  idSprint INT NOT NULL,  
  FOREIGN KEY (idSprint) REFERENCES Sprint (idSprint)  
);
```

```
CREATE TABLE Comentario (  
  idComentario INT AUTO_INCREMENT PRIMARY KEY,  
  contenido TEXT NOT NULL,  
  fecha DATE NOT NULL,  
  idTarea INT NOT NULL,  
  idUsuario INT NOT NULL,  
  FOREIGN KEY (idTarea) REFERENCES Tarea (idTarea),  
  FOREIGN KEY (idUsuario) REFERENCES Usuario (idUsuario)  
);
```

```
CREATE TABLE Evaluacion (  
  idEvaluacion INT AUTO_INCREMENT PRIMARY KEY,  
  fecha DATE NOT NULL,  
  rubricas TEXT,  
  puntaje DECIMAL,  
  idDocente INT NOT NULL,  
  idProyecto INT NOT NULL,  
  FOREIGN KEY (idDocente) REFERENCES Docente (idUsuario),  
  FOREIGN KEY (idProyecto) REFERENCES Proyecto (idProyecto)  
);
```

- Inserciones, modificaciones y actualizaciones

Inserciones

Usuario

```
INSERT INTO Usuario VALUES  
(1, 'Ana Torres', 'ana@example.com', 'clave123'),  
(2, 'Luis Méndez', 'luis@example.com', 'clave123'),  
(3, 'Camila Ríos', 'camila@example.com', 'clave123'),  
(4, 'Pedro Suárez', 'pedro@example.com', 'clave123'),  
(5, 'Laura Pérez', 'laura@example.com', 'clave123'),  
(6, 'Carlos León', 'carlos@example.com', 'clave123'),  
(7, 'Dra. Roberta Díaz', 'roberta@example.com', 'clave123'),  
(8, 'Dr. Julio Aguilar', 'julio@example.com', 'clave123'),  
(9, 'Mg. Teresa Quispe', 'teresa@example.com', 'clave123'),  
(10, 'Ing. Marco Lema', 'marco@example.com', 'clave123');
```

Estudiante

```
INSERT INTO Estudiante VALUES  
(1, 'A1', 3),  
(2, 'A1', 3),  
(3, 'B1', 4),  
(4, 'B1', 4),  
(5, 'C1', 5),  
(6, 'C1', 5);
```

Docente

```
INSERT INTO Docente VALUES  
(7, 'Dra.', 'PhD en Ciencias Computacionales', 'Oficina A-101'),  
(8, 'Dr.', 'PhD en Ingeniería de Software', 'Oficina B-202'),  
(9, 'Mg.', 'Magíster en Informática', 'Oficina C-303'),  
(10, 'Ing.', 'Ingeniero en Sistemas', 'Oficina D-404');
```

Notificación

INSERT INTO Notificacion VALUES

(1, '2025-06-01', 'Entrega de avance del proyecto', 'Recordatorio', 1),
(2, '2025-06-01', 'Reunión con el docente', 'Aviso', 2),
(3, '2025-06-02', 'Cambio de fecha de entrega', 'Alerta', 3),
(4, '2025-06-02', 'Nueva rúbrica de evaluación', 'Informativo', 4),
(5, '2025-06-03', 'Revisión de tareas', 'Aviso', 5),
(6, '2025-06-03', 'Entrega final próxima', 'Recordatorio', 6),
(7, '2025-06-04', 'Revisión de proyecto', 'Sistema', 7),
(8, '2025-06-04', 'Evaluaciones disponibles', 'Sistema', 8),
(9, '2025-06-05', 'Actualización de plataforma', 'Aviso', 9),
(10, '2025-06-05', 'Recordatorio de tutoría', 'Recordatorio', 10);

Proyecto

INSERT INTO Proyecto VALUES

(1, 'App Gestión Tareas', 'Sistema para administrar tareas estudiantiles', 'En desarrollo', '2025-05-01', '2025-08-01', 7),
(2, 'Sistema Académico', 'Módulo de notas y materias', 'Completado', '2025-04-15', '2025-07-15', 8),
(3, 'Plataforma de Cursos', 'Sistema de e-learning', 'Pendiente', '2025-05-10', '2025-09-01', 9),
(4, 'Control de Inventario', 'App para gestión de inventario', 'En desarrollo', '2025-06-01', '2025-09-30', 10),
(5, 'Gestión de Biblioteca', 'Sistema de préstamos y catálogos', 'Planeado', '2025-06-10', '2025-09-10', 7);

Estudiante-Proyecto

INSERT INTO Estudiante_Proyecto VALUES

(1, 1, 'Frontend'),
(2, 1, 'Backend'),
(3, 2, 'Base de datos'),
(4, 2, 'Documentación'),
(5, 3, 'Testing'),
(6, 4, 'Desarrollador');

Documento

INSERT INTO Documento VALUES

- (1, 'Documento Inicial', '2025-05-10', 'v1.0', 'https://example.com/doc1.pdf', 1),
- (2, 'Requisitos', '2025-05-15', 'v1.1', 'https://example.com/doc2.pdf', 1),
- (3, 'Diseño', '2025-05-20', 'v2.0', 'https://example.com/doc3.pdf', 2),
- (4, 'Manual de Usuario', '2025-06-01', 'v1.0', 'https://example.com/doc4.pdf', 2),
- (5, 'Planificación', '2025-06-05', 'v0.9', 'https://example.com/doc5.pdf', 3),
- (6, 'Informe Parcial', '2025-06-15', 'v1.2', 'https://example.com/doc6.pdf', 4),
- (7, 'Plan de Pruebas', '2025-06-20', 'v1.0', 'https://example.com/doc7.pdf', 4),
- (8, 'Resumen Ejecutivo', '2025-06-25', 'v1.0', 'https://example.com/doc8.pdf', 5);

Sprint

INSERT INTO Sprint VALUES

- (1, 'Sprint 1', '2025-05-01', '2025-05-15', 1),
- (2, 'Sprint 2', '2025-05-16', '2025-05-31', 1),
- (3, 'Sprint Inicial', '2025-04-15', '2025-04-30', 2),
- (4, 'Sprint Final', '2025-07-01', '2025-07-15', 2),
- (5, 'Sprint Planeación', '2025-05-10', '2025-05-25', 3),
- (6, 'Sprint Diseño', '2025-06-01', '2025-06-15', 4);

Tarea

INSERT INTO Tarea VALUES

- (1, 'Definir requerimientos', 'Completado', '2025-05-01', '2025-05-05', 8, 1),
- (2, 'Diseño UI', 'En progreso', '2025-05-06', '2025-05-10', 5, 1),
- (3, 'Programación backend', 'Pendiente', '2025-05-16', '2025-05-25', 0, 2),
- (4, 'Pruebas unitarias', 'En progreso', '2025-05-26', '2025-05-31', 2, 2),
- (5, 'Modelado BD', 'Completado', '2025-04-15', '2025-04-18', 6, 3),
- (6, 'Redacción documento final', 'Completado', '2025-07-01', '2025-07-10', 4, 4);

Comentario

INSERT INTO Comentario VALUES

- (1, 'Buen trabajo en los requerimientos.', '2025-05-06', 1, 7),
- (2, 'Faltan íconos en la UI.', '2025-05-10', 2, 7),
- (3, 'El backend no compila.', '2025-05-20', 3, 8),
- (4, 'Agregar más casos de prueba.', '2025-05-28', 4, 8),
- (5, 'Modelo relacional correcto.', '2025-04-20', 5, 9),
- (6, 'Documento completo, bien hecho.', '2025-07-10', 6, 10);

Evaluación

INSERT INTO Evaluacion VALUES

- (1, '2025-06-01', 'Requisitos, Diseño, Funcionalidad', 9.5, 7, 1),
- (2, '2025-06-10', 'Calidad del código, documentación', 8.0, 8, 2),
- (3, '2025-06-15', 'Presentación y pruebas', 9.0, 9, 3),
- (4, '2025-06-20', 'Entrega final, organización', 9.2, 10, 4);

Modificaciones y Actualizaciones:

Ariana Cordova

1. Cambiar nombre y correo de un usuario

```
UPDATE Usuario SET nombre = 'Ana M. Torres', correo = 'ana.mtorres@example.com'
WHERE idUsuario = 1;
```

2. Actualizar paralelo y semestre de un estudiante

```
UPDATE Estudiante SET paralelo = 'B2', semestre = 4 WHERE idUsuario = 2;
```

3. Cambiar oficina de un docente

```
UPDATE Docente SET oficina = 'Oficina E-505' WHERE idUsuario = 10;
```

4. Modificar contenido y tipo de notificación

```
UPDATE Notificacion SET contenido = 'Reunión reprogramada para el lunes', tipo = 'Aviso
importante' WHERE idNotificacion = 2;
```

5. Actualizar estado y fecha de fin de un proyecto

```
UPDATE Proyecto SET estado = 'Finalizado', fecha_fin = '2025-08-15' WHERE idProyecto =
1;
```

Cristhian Dávila

6. Corregir versión y URL de un documento

```
UPDATE Documento SET version = 'v1.1', url = 'https://example.com/actualizado.pdf'
WHERE idDocumento = 2;
```

7. Cambiar rol del estudiante en un proyecto

```
UPDATE Estudiante_Proyecto SET rol = 'Líder de proyecto' WHERE idUsuario = 3 AND
idProyecto = 2;
```

8. Reprogramar fin de sprint

```
UPDATE Sprint SET fecha_fin = '2025-06-20' WHERE idSprint = 1;
```

9. Marcar tarea como completada y registrar horas trabajadas

```
UPDATE Tarea SET estado = 'Completado', horas_trabajadas = 6 WHERE idTarea = 2;
```

10. Ajustar contenido de un comentario

```
UPDATE Comentario SET contenido = 'Buen trabajo, solo ajustar formato.' WHERE
idComentario = 1;
```

- **Consultas y reportes:**

- Consultas en álgebra relacional
- Consultas en SQL (traducción de consultas de álgebra relacional)

Ariana Córdova

a) Obtener los nombres de estudiantes que estén participando en el proyecto "App Gestión Tareas"

- **Relational:** π nombreU(usuario \bowtie estudiante \bowtie estudiante_proyecto \bowtie (σ nombreP == 'App Gestión Tareas' (proyecto)))

nombreU
Ana Torres
Luis Méndez

nombre
Ana Torres
Luis Méndez

- **SQL:** SELECT u.nombre FROM Usuario u JOIN Estudiante_Proyecto ep ON u.idUsuario = ep.idUsuario JOIN Proyecto p ON ep.idProyecto = p.idProyecto WHERE p.nombre = 'App Gestión Tareas';

b) Obtener las tareas que se encuentran "Completadas" y el nombre del proyecto al que pertenecen.

- Relational:** π estadoT, nombreP ((σ estadoT == 'Completado' (tarea)) \bowtie sprint \bowtie proyecto)

estadoT	nombreP
Completado	App Gestión Tareas
Completado	Sistema Académico

estado	nombre_proyecto
Completado	App Gestión Tareas
Completado	Sistema Académico

SQL: SELECT T.estado, P.nombre AS nombre_proyecto FROM Tarea T JOIN Sprint S ON T.idSprint = S.idSprint JOIN Proyecto P ON S.idProyecto = P.idProyecto WHERE T.estado = 'Completado';

c) Obtener el nombre del proyecto en los que participa el estudiante "Camila Ríos", el paralelo y semestre en el que se encuentra cursando.

- Relational:** π nombreP, nombreU, paralelo, semestre(σ nombreU == 'Camila Ríos'(usuario \bowtie estudiante \bowtie proyecto \bowtie estudiante_proyecto))

nombreP	nombreU	paralelo	semestre
Sistema Académico	Camila Ríos	B1	4

nombre	nombre_proyecto	paralelo	semestre
Sistema Académico	Camila Ríos	B1	4

SQL: SELECT P.nombre, U.nombre AS nombre_proyecto, E.paralelo, E.semestre FROM Usuario U JOIN Estudiante E ON U.idUsuario = E.idUsuario JOIN Estudiante_Proyecto EP ON E.idUsuario = EP.idUsuario JOIN Proyecto P ON EP.idProyecto = P.idProyecto WHERE U.nombre = 'Camila Ríos';

- d) Obtener el nombre de los proyectos con una puntuación menor a 9

Relational: π nombreP, puntajeE $((\sigma$ puntajeE < 9 (evaluacion)) \bowtie proyecto)

nombreP	puntajeE
Sistema Académico	8.0

nombre	puntaje
Sistema Académico	8

SQL: SELECT DISTINCT P.nombre, E.puntaje FROM Evaluacion E JOIN Proyecto P ON E.idProyecto = P.idProyecto WHERE E.puntaje < 9;

- e) Obtener los nombres y comentarios realizados por docentes cuyo grado académico sea 'PhD en Ingeniería de Software'.

Relational: π nombreU, contenidoC $((\sigma$ grado_academicoD == 'PhD en Ingeniería de Software'(docente \bowtie usuario)) \bowtie comentario)

nombreU	contenidoC
Dr. Julio Aguilar	Agregar más casos de prueba.
Dr. Julio Aguilar	El backend no compila.

nombre	contenido
Dr. Julio Aguilar	El backend no compila.
Dr. Julio Aguilar	Agregar más casos de prueba.

SQL: SELECT U.nombre, C.contenido FROM Comentario C JOIN Usuario U ON C.idUsuario = U.idUsuario JOIN Docente D ON U.idUsuario = D.idUsuario WHERE D.grado_academico = 'PhD en Ingeniería de Software';

Cristhian Dávila

- f) Obtener los comentarios hechos por los docentes

- **Relational:** π nombreU, contenidoC (comentario \bowtie (usuario \bowtie docente))

nombre	contenido
Dra. Roberta Díaz	Buen trabajo en los requerimientos.
Dra. Roberta Díaz	Faltan íconos en la UI.
Dr. Julio Aguilar	El backend no compila.
Dr. Julio Aguilar	Agregar más casos de prueba.
Mg. Teresa Quispe	Modelo relacional correcto.
Ing. Marco Lema	Documento completo, bien hecho.

nombreU	contenidoC
Ing. Marco Lema	Documento completo, bien hecho.
Dr. Julio Aguilar	El backend no compila.
Dr. Julio Aguilar	Agregar más casos de prueba.
Dra. Roberta Díaz	Buen trabajo en los requerimientos.
Mg. Teresa Quispe	Modelo relacional correcto.
Dra. Roberta Díaz	Faltan íconos en la UI.

- **SQL:** SELECT u.nombre, c.contenido FROM Comentario c JOIN Usuario u ON c.idUsuario = u.idUsuario JOIN Docente d ON u.idUsuario = d.idUsuario;

- g) Obtención del nombre de los documentos y proyecto al que pertenecen

Relational: π nombreD, nombreP (documento \bowtie proyecto)

nombre	nombre
Documento Inicial	App Gestión Tareas
Requisitos	App Gestión Tareas
Diseño	Sistema Académico
Manual de Usuario	Sistema Académico
Planificación	Plataforma de Cursos
Informe Parcial	Control de Inventario
Plan de Pruebas	Control de Inventario
Resumen Ejecutivo	Gestión de Biblioteca

nombreD	nombreP
Requisitos	App Gestión Tareas
Diseño	Sistema Académico
Manual de Usuario	Sistema Académico
Informe Parcial	Control de Inventario
Resumen Ejecutivo	Gestión de Biblioteca
Documento Inicial	App Gestión Tareas
Plan de Pruebas	Control de Inventario
Planificación	Plataforma de Cursos

SQL: SELECT d.nombre, p.nombre FROM Documento d JOIN Proyecto p ON d.idProyecto = p.idProyecto;

- h) Mostrar los nombres de los estudiantes con su nombre del proyecto y el rol respectivo

Relational: π nombreP,nombreU,rolEP ((estudiante_proyecto \bowtie estudiante) \bowtie usuario \bowtie proyecto)

nombre	nombre_proyecto	rol
Ana Torres	App Gestión Tareas	Frontend
Luis Méndez	App Gestión Tareas	Backend
Camila Ríos	Sistema Académico	Base de datos
Pedro Suárez	Sistema Académico	Documentación
Laura Pérez	Plataforma de Cursos	Testing
Carlos León	Control de Inventario	Desarrollador

nombreP	nombreU	rolEP
App Gestión Tareas	Luis Méndez	Backend
Control de Inventario	Carlos León	Desarrollador
Sistema Académico	Camila Ríos	Base de datos
Plataforma de Cursos	Laura Pérez	Testing
App Gestión Tareas	Ana Torres	Frontend
Sistema Académico	Pedro Suárez	Documentación

SQL: SELECT u.nombre, p.nombre AS nombre_proyecto, ep.rol
FROM Estudiante_Proyecto ep
JOIN Estudiante e ON ep.idUsuario = e.idUsuario
JOIN Usuario u ON e.idUsuario = u.idUsuario
JOIN Proyecto p ON ep.idProyecto = p.idProyecto;

- i) Obtener los nombres de los usuarios con su respectiva notificación

Relational: π nombreU, fechaN, tipoN, contenidoN (notificacion \bowtie usuario)

nombre	fecha	tipo	contenido
Ana Torres	2025-06-01	Recordatorio	Entrega de avance del proyecto
Luis Méndez	2025-06-01	Aviso	Reunión con el docente
Camila Ríos	2025-06-02	Alerta	Cambio de fecha de entrega
Pedro Suárez	2025-06-02	Informativo	Nueva rúbrica de evaluación
Laura Pérez	2025-06-03	Aviso	Revisión de tareas
Carlos León	2025-06-03	Recordatorio	Entrega final próxima
Dra. Roberta Díaz	2025-06-04	Sistema	Revisión de proyecto
Dr. Julio Aguilar	2025-06-04	Sistema	Evaluaciones disponibles
Mg. Teresa Quispe	2025-06-05	Aviso	Actualización de plataforma
Ing. Marco Lema	2025-06-05	Recordatorio	Recordatorio de tutoría

nombreU	fechaN	tipoN	contenidoN
Camila Ríos	2025-06-02	Alerta	Cambio de fecha de entrega
Ing. Marco Lema	2025-06-05	Recordatorio	Recordatorio de tutoría
Luis Méndez	2025-06-01	Aviso	Reunión con el docente
Ana Torres	2025-06-01	Recordatorio	Entrega de avance del proyecto
Pedro Suárez	2025-06-02	Informativo	Nueva rúbrica de evaluación
Laura Pérez	2025-06-03	Aviso	Revisión de tareas
Carlos León	2025-06-03	Recordatorio	Entrega final próxima
Dra. Roberta Díaz	2025-06-04	Sistema	Revisión de proyecto
Mg. Teresa Quispe	2025-06-05	Aviso	Actualización de plataforma
Dr. Julio Aguilar	2025-06-04	Sistema	Evaluaciones disponibles

SQL: SELECT u.nombre, n.fecha, n.tipo, n.contenido FROM Notificacion n
JOIN Usuario u ON n.idUsuario = u.idUsuario;

- j) Obtener las Tareas con su respectivo Sprint

Relational: π idTarea, descripcionT, estadoT, fecha_inicioT, fecha_finT, nombreS (tarea \bowtie sprint)

idTarea	descripcion	estado	fecha_inicio	fecha_fin	nombre_sprint
1	Definir requerimientos	Completado	2025-05-01	2025-05-05	Sprint 1
2	Diseño UI	En progreso	2025-05-06	2025-05-10	Sprint 1
3	Programación backend	Pendiente	2025-05-16	2025-05-25	Sprint 2
4	Pruebas unitarias	En progreso	2025-05-26	2025-05-31	Sprint 2
5	Modelado BD	Completado	2025-04-15	2025-04-18	Sprint Inicial
6	Redacción documento final	Completado	2025-07-01	2025-07-10	Sprint Final

idTarea	descripcionT	estadoT	fecha_inicioT	fecha_finT	nombreS
1	Definir requerimientos	Completado	2025-05-01	2025-05-05	Sprint 1
5	Modelado BD	Completado	2025-04-15	2025-04-18	Sprint Inicial
2	Diseño UI	En progreso	2025-05-06	2025-05-10	Sprint 1
4	Pruebas unitarias	En progreso	2025-05-26	2025-05-31	Sprint 2
3	Programación backend	Pendiente	2025-05-16	2025-05-25	Sprint 2
6	Redacción documento final	Completado	2025-07-01	2025-07-10	Sprint Final

SQL: SELECT t.idTarea, t.descripcion, t.estado, t.fecha_inicio, t.fecha_fin, s.nombre AS
nombre_sprint FROM Tarea t JOIN Sprint s ON t.idSprint = s.idSprint;

- **Views:**
 - Creación de vistas por cada consulta.

Ariana Córdova

- 1) CREATE VIEW Consulta1 AS SELECT u.nombre FROM Usuario u JOIN Estudiante_Proyecto ep ON u.idUsuario = ep.idUsuario JOIN Proyecto p ON ep.idProyecto = p.idProyecto WHERE p.nombre = 'App Gestión Tareas';
- 2) CREATE VIEW Consulta2 AS SELECT T.estado, P.nombre AS nombre_proyecto FROM Tarea T JOIN Sprint S ON T.idSprint = S.idSprint JOIN Proyecto P ON S.idProyecto = P.idProyecto WHERE T.estado = 'Completado';
- 3) CREATE VIEW Consulta3 AS SELECT P.nombre, U.nombre AS nombre_proyecto, E.paralelo, E.semestre FROM Usuario U JOIN Estudiante E ON U.idUsuario = E.idUsuario JOIN Estudiante_Proyecto EP ON E.idUsuario = EP.idUsuario JOIN Proyecto P ON EP.idProyecto = P.idProyecto WHERE U.nombre = 'Camila Ríos';
- 4) CREATE VIEW Consulta4 AS SELECT DISTINCT P.nombre, E.puntaje FROM Evaluacion E JOIN Proyecto P ON E.idProyecto = P.idProyecto WHERE E.puntaje < 9;
- 5) CREATE VIEW Consulta5 AS SELECT U.nombre, C.contenido FROM Comentario C JOIN Usuario U ON C.idUsuario = U.idUsuario JOIN Docente D ON U.idUsuario = D.idUsuario WHERE D.grado_academico = 'PhD en Ingeniería de Software';

Tables_in_SistemaGestionProyectos	Table_type
Consulta1	VIEW
Consulta2	VIEW
Consulta3	VIEW
Consulta4	VIEW
Consulta5	VIEW

Cristhian Dávila

- 1) CREATE VIEW consulta1 AS SELECT u.nombre, c.contenido FROM Comentario c JOIN Usuario u ON c.idUsuario = u.idUsuario JOIN Docente d ON u.idUsuario = d.idUsuario;
- 2) CREATE VIEW consulta7 AS SELECT d.nombre AS nombre_documento, p.nombre AS nombre_proyecto FROM Documento d JOIN Proyecto p ON d.idProyecto = p.idProyecto;
- 3) CREATE VIEW consulta3 AS SELECT u.nombre, p.nombre AS nombre_proyecto, ep.rol FROM Estudiante_Proyecto ep JOIN Estudiante e ON ep.idUsuario = e.idUsuario JOIN Usuario u ON e.idUsuario = u.idUsuario JOIN Proyecto p ON ep.idProyecto = p.idProyecto;
- 4) CREATE VIEW consulta4 AS SELECT u.nombre, n.fecha, n.tipo, n.contenido FROM Notificacion n JOIN Usuario u ON n.idUsuario = u.idUsuario;
- 5) CREATE VIEW consulta5 AS SELECT t.idTarea, t.descripcion, t.estado, t.fecha_inicio, t.fecha_fin, s.nombre AS nombre_sprint FROM Tarea t JOIN Sprint s ON t.idSprint = s.idSprint;

Tables_in_SistemaGestionProyectos	Table_type
consulta1	VIEW
consulta2	VIEW
consulta3	VIEW
consulta4	VIEW
consulta5	VIEW

5. Resultados Obtenidos

- Agregar los productos de la práctica en github o gitlab
 - Diagrama entidad relación
 - Diagrama relacional
 - Backup de la BD en .sql
 - Archivos .csv para las consultas en álgebra relacional
 - Informe de la práctica preprofesional

Link Github: <https://github.com/Cristhian1938/PracticasPreProfesionalesBD>

6. Conclusiones y Recomendaciones

Conclusiones

- En el desarrollo de las presentes practicas preprofesionales, se aplicó de manera efectiva los conocimientos adquiridos hasta la presente fecha, se realizo el diseño, modelado e implementación de una base de datos relaciones orientada a la gestión de proyectos académicos. Esto nos permite poner en práctica nuestras capacidades y habilidades técnicas, en donde se dio uso ha herramientas importantes como MySQL, diagramas de entidad-relación, normalización y lenguaje SQL.
- La participación activa de la creación de estructuras de base de datos, inserción y actualización de datos, al igual que la elaboración de consultas en algebra relacional y SQL, nos facilitó una mejor comprensión dentro del desarrollo del proyecto.
- Con todo esto se puede decir que la práctica nos aportó grata experiencia y aplicación mediante la entrega de una solución funcional que estructura y optimiza el manejo de información en proyectos.

Recomendaciones

- Ante la finalización de las presentes prácticas, damos recomendación en los siguientes puntos:
- Continuar promoviendo prácticas profesionales orientadas a caos reales, ya que permite al estudiante consolidade sus habilidades técnicas.
- Dar varias revisiones al modelo entidad-relación antes de su implementación en los siguientes puntos, ya que se requiere tener una base solidad para un producto final exitoso.
- Ir documentando todo el proyecto paso a paso de todo lo que se vaya realizando.
- Continuar brindando un acompañamiento continuo por parte de profesionales del área para permitir una retroalimentación oportuna durante el receso.

7. Firma y Validación

Estudiante

Docente Supervisor



Cristhian Alexander

René Guamán Quinche

1105834111

Ariana Córdova

René Guamán Quinche
