



# UNIVERSIDAD NACIONAL DE LOJA

FACULTAD DE LA ENERGÍA LAS INDUSTRIAS  
Y LOS RECURSOS NO RENOVABLES

CARRERA DE COMPUTACIÓN

TEMA:

"INFORME DE PRÁCTICAS PREPROFESIONALES EN  
BASE DE DATOS"

NOMBRE:

Gerardo David Naula Saraguro

DOCENTE:

Ing. René Guamán-Quinche

LOJA-ECUADOR

1859

FECHA: 30/07/2025



**UNL**

Universidad  
Nacional  
de Loja

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

Carrera Ingeniería en Computación

### A. Datos Informativos

<b>Asignatura:</b> Base de datos	<b>Semestre:</b> Tercero	<b>Paralelo:</b> A
<b>Docente Supervisor:</b> René Guamán-Quinche	<b>Periodo Académico:</b> Abril - Agosto de 2025	
<b>Estudiante:</b> Gerardo David Naula Saraguro		
<b>Periodo de prácticas:</b> Abril - Agosto de 2025		

## 1. Introducción

Las prácticas preprofesionales constituyen una etapa fundamental en la formación académica de los estudiantes, ya que permiten aplicar los conocimientos teóricos adquiridos en un entorno real de trabajo, favoreciendo el desarrollo de competencias técnicas y profesionales.

El objetivo principal de estas prácticas fue permitir que el estudiante aplique los conocimientos adquiridos en clases en un entorno más práctico y cercano a situaciones reales. En este caso, las actividades se desarrollaron en el contexto académico, dentro de la materia de Base de Datos de la carrera de Computación. A lo largo del proceso, se realizaron tareas orientadas al diseño, desarrollo y manejo de bases de datos, aplicando herramientas y técnicas vistas en clase para reforzar el aprendizaje. Estas actividades ayudaron a fortalecer habilidades técnicas y a comprender mejor cómo se gestionan los datos en sistemas reales. En el contexto de las actividades realizadas, las bases de datos jugaron un papel crucial, ya que constituyen el eje central para el almacenamiento, organización, consulta y análisis de la información. La correcta gestión de las bases de datos permite garantizar la integridad, consistencia y disponibilidad de los datos, elementos fundamentales para la toma de decisiones y la eficiencia operativa.

La experiencia adquirida a través de estas prácticas preprofesionales no solo reforzó los conocimientos técnicos sobre modelado, diseño e implementación de bases de datos, sino que también brindó la oportunidad de trabajar con herramientas reales,



interactuar con profesionales del área y comprender la importancia del trabajo colaborativo y la documentación adecuada de los procesos.

## 2. Objetivos

### 2.1 Objetivo General

Aplicar los conocimientos teóricos y prácticos en el diseño, implementación y mantenimiento de bases de datos, mediante el desarrollo de actividades técnicas en un entorno profesional que favorezca el fortalecimiento de competencias en el manejo de sistemas de gestión de bases de datos

### 2.2 Objetivos Específicos

- Diseñar e implementar modelos de bases de datos utilizando el modelo entidad-relación y su correspondiente traducción al modelo relacional.
- Optimizar consultas SQL mediante el análisis lógico y estructural de los requerimientos de información.
- Realizar tareas de mantenimiento y gestión de bases de datos existentes, incluyendo inserciones, actualizaciones, generación de vistas y respaldo de información.

## 3. Actividades Realizadas

Contexto:

La Carrera de Computación de la Universidad Nacional de Loja (UNL) necesita un Sistema de Gestión de Requisitos (SGR) para apoyar la enseñanza de la asignatura Ingeniería de Requisitos y mejorar la calidad de los proyectos de software desarrollados por los estudiantes. Actualmente, los alumnos y docentes gestionan requisitos mediante documentos compartidos (Google Docs, hojas de cálculo o archivos Word), lo que genera problemas de trazabilidad, inconsistencia y falta de colaboración efectiva. El SGR permitirá a estudiantes, docentes y actores clave registrar, clasificar, priorizar y rastrear requisitos de software de manera estructurada, siguiendo buenas prácticas de la ingeniería de requisitos. Necesidades Generales (Problemas a Resolver):



**UNL**

Universidad  
Nacional  
de Loja

1859

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

Carrera Ingeniería en Computación

**1. Registro y Clasificación de Requisitos:**

- Los estudiantes deben poder registrar requisitos funcionales y no funcionales.
- El sistema debe permitir clasificar requisitos según su tipo (información, funcionalidad, restricciones).
- Debe soportar atributos como: prioridad, fuente (stakeholder), estado (propuesto, aprobado, implementado).

**2. Trazabilidad y Relaciones entre Requisitos:**

- Permitir vincular requisitos (dependencias, conflictos, refinamientos).
- Visualizar el impacto de cambios en requisitos.

**3. Colaboración y Revisión:**

- Docentes y equipos de estudiantes deben poder comentar, aprobar o rechazar requisitos.
- Notificaciones cuando un requisito sea modificado.

**4. Integración con Metodologías Ágiles (Opcional):**

- Posibilidad de organizar requisitos en historias de usuario (si se usa Scrum).
- Generación de diagramas simples (casos de uso, modelos de datos).

**5. Reportes y Exportación:**

- Generar documentos estructurados (Especificación de Requisitos - ERS).
- Exportar requisitos a formatos como PDF, Excel o herramientas como Jira.

**6. Seguridad:**

- Acceso diferenciado (estudiantes solo editan sus proyectos, docentes supervisan todos).
- Autenticación con credenciales institucionales.

- **Modelo entidad – relación**



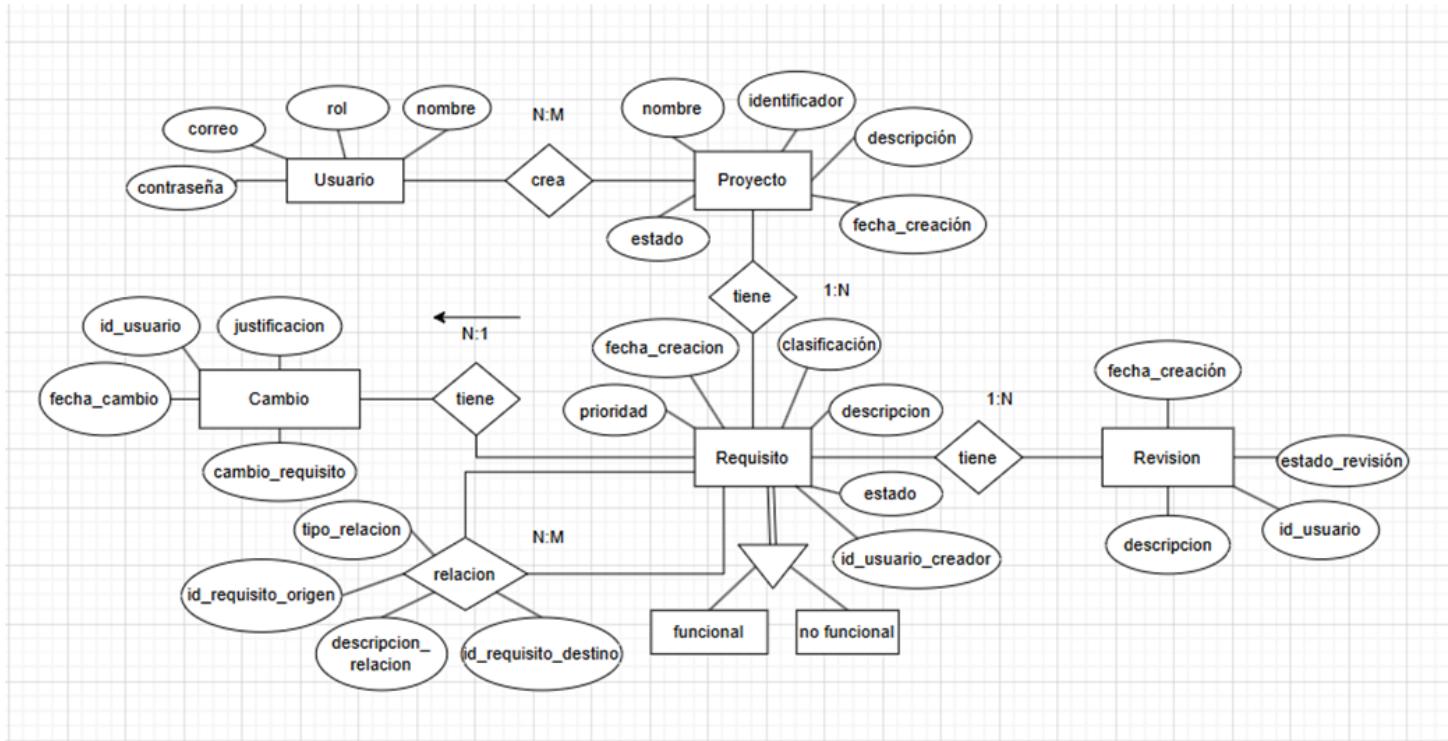
UNL

Universidad  
Nacional  
de Loja

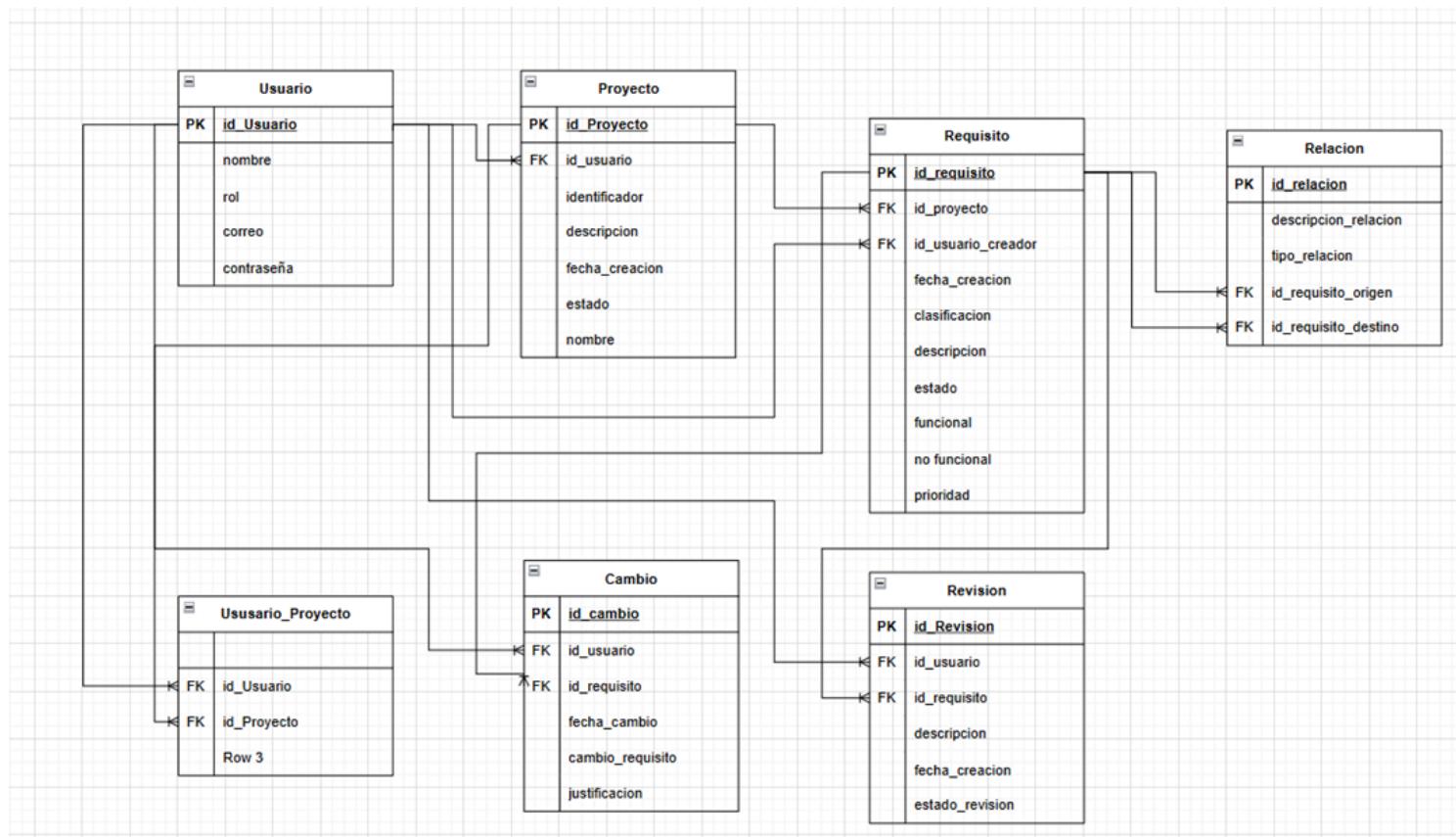
1859

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

Carrera Ingeniería en Computación



#### 4. Modelo Relacional: traducción del modelo entidad - relación





## 5. Detalle de normalización de la base de datos

### Primera Forma Normal (1FN)

Objetivo: Eliminar los grupos repetitivos y asegurar que todos los atributos sean atómicos.

Aplicación:

- Todas las tablas del modelo cumplen con esta forma normal, ya que:
- Cada atributo almacena un único valor por celda (por ejemplo, nombre, correo, estado, etc.).
- No hay campos multivaluados ni listas separadas por comas.
- Las claves primarias (PK) están claramente definidas para cada entidad.

### Segunda Forma Normal (2FN)

Objetivo: Eliminar dependencias parciales (atributos que dependen solo de una parte de una clave primaria compuesta).

Aplicación:

- La mayoría de las entidades tienen una clave primaria simple, por lo tanto, no hay dependencias parciales.
- La tabla Usuario\_Proyecto tiene una clave primaria compuesta (id\_Usuario, id\_Proyecto), y no contiene atributos adicionales, por lo que también cumple con 2FN.

### Tercera Forma Normal (3FN)

Objetivo: Eliminar las dependencias transitivas (cuando un atributo no clave depende de otro atributo no clave).

Aplicación:

- En la tabla Requisito, aunque hay varios atributos (clasificación, estado, funcional, no funcional, prioridad), no hay atributos que dependan transitivamente de otro atributo no clave, ya que todos se refieren directamente al requisito.
- Las tablas Cambio y Revision están correctamente estructuradas: los atributos como justificación, estado\_revision o descripcion dependen directamente de sus respectivas claves primarias.

## 6. Diseño físico: Uso del SGBD (MySQL)

### 6.1 Creación de base de datos, tabla

```
mysql> CREATE TABLE Usuario (
-> id_usuario INT AUTO_INCREMENT PRIMARY KEY,
```



**UNL**

Universidad  
Nacional  
de Loja

1859

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

Carrera Ingeniería en Computación

```
-> nombre VARCHAR(100),  
-> correo VARCHAR(100) UNIQUE NOT NULL,  
-> contraseña VARCHAR(100) NOT NULL,  
-> rol VARCHAR(50) NOT NULL  
-> );
```

Query OK, 0 rows affected (0.07 sec)

Field	Type	Null	Key	Default	Extra
id_usuario	int	NO	PRI	NULL	auto_increment
nombre	varchar(100)	YES		NULL	
correo	varchar(100)	NO	UNI	NULL	
contraseña	varchar(100)	NO		NULL	
rol	varchar(50)	NO		NULL	
5 rows in set (0.00 sec)					

```
mysql> CREATE TABLE Proyecto (  
-> id_proyecto INT AUTO_INCREMENT PRIMARY KEY,  
-> nombre VARCHAR(100) NOT NULL,  
-> identificador VARCHAR(50) UNIQUE NOT NULL,  
-> descripcion TEXT,  
-> estado VARCHAR(50),  
-> fecha_creacion DATE NOT NULL  
-> );
```

Query OK, 0 rows affected (0.05 sec)

Field	Type	Null	Key	Default	Extra
id_proyecto	int	NO	PRI	NULL	auto_increment
nombre	varchar(100)	NO		NULL	
identificador	varchar(50)	NO	UNI	NULL	
descripcion	text	YES		NULL	
estado	varchar(50)	YES		NULL	
fecha_creacion	date	NO		NULL	
6 rows in set (0.01 sec)					

```
mysql> CREATE TABLE Usuario_Proyecto (  
-> id_usuario INT NOT NULL,  
-> id_proyecto INT NOT NULL,  
-> PRIMARY KEY (id_usuario, id_proyecto),  
-> FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),  
-> FOREIGN KEY (id_proyecto) REFERENCES Proyecto(id_proyecto)  
-> );
```

Query OK, 0 rows affected (0.06 sec)



**UNL**

Universidad  
Nacional  
de Loja

1859

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

Carrera Ingeniería en Computación

```
mysql> CREATE TABLE Requisito (
-> id_requisito INT AUTO_INCREMENT PRIMARY KEY,
-> descripcion TEXT NOT NULL,
-> estado VARCHAR(50),
-> prioridad VARCHAR(50),
-> clasificacion VARCHAR(50),
-> fecha_creacion DATE NOT NULL,
-> tipo VARCHAR(50),
-> id_usuario_creador INT,
-> id_proyecto INT,
-> FOREIGN KEY (id_usuario_creador) REFERENCES Usuario(id_usuario),
-> FOREIGN KEY (id_proyecto) REFERENCES Proyecto(id_proyecto)
-> );
```

Query OK, 0 rows affected (0.07 sec)

Field	Type	Null	Key	Default	Extra
id_requisito	int	NO	PRI	NULL	auto_increment
descripcion	text	NO		NULL	
estado	varchar(50)	YES		NULL	
prioridad	varchar(50)	YES		NULL	
clasificacion	varchar(50)	YES		NULL	
fecha_creacion	date	NO		NULL	
tipo	varchar(50)	YES		NULL	
id_usuario_creador	int	YES	MUL	NULL	
id_proyecto	int	YES	MUL	NULL	

```
mysql> CREATE TABLE Cambio (
-> id_cambio INT AUTO_INCREMENT PRIMARY KEY,
-> id_usuario INT,
-> fecha_cambio DATE NOT NULL,
-> justificacion TEXT NOT NULL,
-> cambio_requisito TEXT NOT NULL,
-> id_requisito INT,
-> FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),
-> FOREIGN KEY (id_requisito) REFERENCES Requisito(id_requisito)
-> );
```

Query OK, 0 rows affected (0.06 sec)

Field	Type	Null	Key	Default	Extra
id_cambio	int	NO	PRI	NULL	auto_increment
id_usuario	int	YES	MUL	NULL	
fecha_cambio	date	NO		NULL	
justificacion	text	NO		NULL	
cambio_requisito	text	NO		NULL	
id_requisito	int	YES	MUL	NULL	



**UNL**

Universidad  
Nacional  
de Loja

1859

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

Carrera Ingeniería en Computación

```
mysql> CREATE TABLE Revision (
-> id_revision INT AUTO_INCREMENT PRIMARY KEY,
-> fecha_creacion DATE NOT NULL,
-> descripcion TEXT,
-> estado_revision VARCHAR(50),
-> id_usuario INT,
-> id_requisito INT,
-> FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),
-> FOREIGN KEY (id_requisito) REFERENCES Requisito(id_requisito)
-> );
```

Query OK, 0 rows affected (0.07 sec)

Field	Type	Null	Key	Default	Extra
id_revision	int	NO	PRI	NULL	auto_increment
fecha_creacion	date	NO		NULL	
descripcion	text	YES		NULL	
estado_revision	varchar(50)	YES		NULL	
id_usuario	int	YES	MUL	NULL	
id_requisito	int	YES	MUL	NULL	
6 rows in set (0.01 sec)					

```
mysql> CREATE TABLE Relacion (
-> id_relacion INT AUTO_INCREMENT PRIMARY KEY,
-> id_requisito_origen INT,
-> id_requisito_destino INT,
-> tipo_relacion VARCHAR(50),
-> descripcion_relacion TEXT,
-> FOREIGN KEY (id_requisito_origen) REFERENCES Requisito(id_requisito),
-> FOREIGN KEY (id_requisito_destino) REFERENCES Requisito(id_requisito)
-> );
```

Query OK, 0 rows affected (0.06 sec)

Field	Type	Null	Key	Default	Extra
id_relacion	int	NO	PRI	NULL	auto_increment
id_requisito_origen	int	YES	MUL	NULL	
id_requisito_destino	int	YES	MUL	NULL	
tipo_relacion	varchar(50)	YES		NULL	
descripcion_relacion	text	YES		NULL	
5 rows in set (0.00 sec)					



**UNL**

Universidad  
Nacional  
de Loja

1859

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

Carrera Ingeniería en Computación

Tables_in_gestion_requisitos
Cambio
CambiosFuncionales
CambiosLuis
Proyecto
ProyectosActivos
Relacion
Requisito
RequisitosAnaBaja
RequisitosEstudiantes
Revision
Usuario
Usuario_Proyecto

## 6.2 Inserciones, modificaciones y actualizaciones

### Inserciones:

```
mysql> INSERT INTO Proyecto (
-> idProyecto, nombre, identificador, descripcion, estado, fecha_creacion
-> ) VALUES
-> (1,'SGR','PR001','Sistema de Gestión de Requisitos','activo','2025-07-01'),
-> (2,'Sistema Notas','PR002','Sistema de Notas UNL','inactivo','2025-06-15'),
-> (3,'Evaluador UNL','PR003','Sistema de evaluación docente','activo','2025-07-02');
```

mysql> select * from Proyecto;					
id_proyecto	nombre	identificador	descripcion	estado	fecha_creacion
1	SGR	PR001	Sistema de Gestión de Requisitos	activo	2025-07-01
2	Sistema Notas	PR002	Sistema de Notas UNL	inactivo	2025-06-15
3	Evaluador UNL	PR003	Sistema de evaluación docente	activo	2025-07-02

```
mysql> INSERT INTO Usuario (
-> idUsuario, nombre, correo, contraseña, rol
-> ) VALUES
-> (1,'Ana','ana@unl.edu.ec','1234','estudiante'),
-> (2,'Luis','luis@unl.edu.ec','abcd','estudiante'),
-> (3,'Carlos','carlos@unl.edu.ec','pass','docente'),
-> (4,'María','maria@unl.edu.ec','4321','estudiante'),
```

id_usuario	nombre	correo	contraseña	rol
1	Ana	ana@unl.edu.ec	1234	estudiante
2	Luis	luis@unl.edu.ec	abcd	estudiante
3	Carlos	carlos@unl.edu.ec	pass	docente
4	María	maria@unl.edu.ec	4321	estudiante
5	Pedro	pedro@unl.edu.ec	xyz987	docente



**UNL**

Universidad  
Nacional  
de Loja

1859

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

Carrera Ingeniería en Computación

```
mysql> INSERT INTO Usuario_Proyecto (
```

```
-> idUsuario, idProyecto
-> ) VALUES
-> (1,1),
-> (2,1),
-> (3,1),
-> (3,2),
-> (2,3),
-> (4,3),
-> (5,3);
```

```
mysql> INSERT INTO Requisito (
```

```
-> idRequisito, nombre, estado, prioridad, clasificacion, fecha_creacion, tipo,
id_usuario_creador, id_proyecto
-> ) VALUES
-> (1,'Registrar requisitos','propuesto','alta','funcionalidad','2025-07-
01','funcional',1,1),
-> (2,'Seguridad de acceso','aprobado','media','restriccion','2025-07-01','no
funcional',2,1),
-> (3,'Exportar a PDF','implementado','baja','informacion','2025-07-02','funcional',1,1),
-> (4,'Registrar usuarios','propuesto','alta','funcionalidad','2025-07-02','funcional',4,3),
-> (5,'Autenticación por token','aprobado','alta','restriccion','2025-07-02','no
funcional',5,3),
-> (6,'Generar reportes','propuesto','media','informacion','2025-07-02','funcional',2,3);
```

id_requisito   descripcion	estado	prioridad	clasificacion	fecha_creacion	tipo	id_usuario_creador	id_proyecto
1   Registrar requisitos	propuesto	alta	funcionalidad	2025-07-01	funcional	1	1
2   Seguridad de acceso	aprobado	media	restriccion	2025-07-01	no funcional	2	1
3   Exportar a PDF	implementado	baja	informacion	2025-07-02	funcional	1	1
4   Registrar usuarios	propuesto	alta	funcionalidad	2025-07-02	funcional	4	3
5   Autenticación por token	aprobado	alta	restriccion	2025-07-02	no funcional	5	3
6   Generar reportes	propuesto	media	informacion	2025-07-02	funcional	2	3

```
mysql> INSERT INTO Relacion (
```

```
-> id_Relacion, id_requisito_origen, id_requisito_destino, tipo_relacion,
descripcion_relacion
-> ) VALUES
-> (1,1,3,'dependencia','Exportar depende de registrar requisitos'),
-> (2,2,1,'restriccion','Seguridad restringe el registro'),
-> (3,4,6,'dependencia','Reportes necesitan datos de usuarios registrados'),
-> (4,5,4,'restriccion','Autenticación limita creación de usuarios');
```

id_relacion   id_requisito_origen   id_requisito_destino   tipo_relacion   descripcion_relacion
1   1   3   dependencia   Exportar depende de registrar requisitos
2   2   1   restriccion   Seguridad restringe el registro
3   4   6   dependencia   Reportes necesitan datos de usuarios registrados
4   5   4   restriccion   Autenticación limita creación de usuarios



**UNL**

Universidad  
Nacional  
de Loja

1859

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

Carrera Ingeniería en Computación

```
mysql> INSERT INTO Revision (
-> idRevision, fecha_creacion, descripcion, estado_revision, id_Usuario, id_Requisito
-> ) VALUES
-> (1,'2025-07-02','Revisión inicial','aprobado',3,1),
-> (2,'2025-07-02','Observaciones funcionales','pendiente',3,3),
-> (3,'2025-07-02','Validación de requerimientos','aprobado',5,5),
-> (4,'2025-07-02','Corrección de inconsistencias','pendiente',2,6);
```

id_revision	fecha_creacion	descripcion	estado_revision	id_usuario	id_requisito
1	2025-07-02	Revisión inicial	aprobado	3	1
2	2025-07-02	Observaciones funcionales	pendiente	3	3
3	2025-07-02	Validación de requerimientos	aprobado	5	5
4	2025-07-02	Corrección de inconsistencias	pendiente	2	6

4 rows in set (0.00 sec)

```
mysql> INSERT INTO Cambio (
-> id_cambio, id_usuario, fecha_cambio, justificacion, cambio_requisito, id_requisito
-> ) VALUES
-> (1,2,'2025-07-02','Corrección descripción','Se actualizó el alcance',1),
-> (2,1,'2025-07-02','Cambio de prioridad','Alta a Media',3),
-> (3,3,'2025-07-02','Nuevo campo agregado','Se añadió campo de estado',2),
-> (4,4,'2025-07-02','Corrección ortográfica','Revisión en nombres del formulario',4);
```

id_cambio	id_usuario	fecha_cambio	justificacion	cambio_requisito	id_requisito
1	2	2025-07-02	Corrección descripción	Se actualizó el alcance	1
2	1	2025-07-02	Cambio de prioridad	Alta a Media	3
3	3	2025-07-02	Nuevo campo agregado	Se añadió campo de estado	2
4	4	2025-07-02	Corrección ortográfica	Revisión en nombres del formulario	4

4 rows in set (0.00 sec)

## MODIFICACIONES Y ACTUALIZACIONES:

### Actualizar varios campos de un requisito

UPDATE Requisito

```
SET estado = 'implementado',
prioridad = 'alta'
```

```
WHERE idRequisito = 6;
```

### Cambiar el tipo de relación entre dos requisitos

UPDATE Relacion

```
SET tipo_relacion = 'complemento'
WHERE id_Relacion = 2;
```

### Corregir un error ortográfico en la descripción de un cambio



**UNL**

Universidad  
Nacional  
de Loja

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

Carrera Ingeniería en Computación

UPDATE Cambio

SET cambio\_requisito = 'Se actualizó correctamente el alcance'

WHERE id\_cambio = 1;

### Modificar la prioridad de un requisito

UPDATE Requisito

SET prioridad = 'media'

WHERE idRequisito = 1;

## 7. Consultas y reportes:

### 7.1 Consultas en álgebra relacional

Mostrar todos los cambios realizados por el usuario llamado “Luis”:

$\pi \text{id\_cambio, fecha\_cambio, justificacion, cambio\_requisito} (\sigma \text{nombre} == \text{"Luis"}(\text{usuario}) \bowtie \text{cambio})$

File Relations Settings Help					
Operators		id_cambio	fecha_cambio	justificacion	cambio_requisito
*	-	1	2025-07-02	Corrección	Se actualizó el alcance
U	n				
▷	◁				
◁	▷				
÷					
σ	π				
ρ	→				

$\pi \text{id\_cambio, fecha\_cambio, justificacion, cambio\_requisito} (\sigma \text{nombre} == \text{"Luis"}(\text{usuario}) \bowtie \text{cambio})$

Listar los requisitos creados por estudiantes:

$\pi \text{id\_requisito, descripcion} (\sigma \text{rol} == \text{"estudiante"}(\text{usuario}) \bowtie \text{requisito})$

File Relations Settings Help			
Operators		id_requisito	descripcion
*	-	1	Registrar requisitos
U	n	6	Generar reportes
▷	◁	5	Autenticación por token
◁	▷	2	Seguridad de acceso
÷		4	Registrar usuarios
σ	π	3	Exportar a PDF
ρ	→		

$\pi \text{id\_cambio, fecha\_cambio, justificacion, cambio\_requisito} (\sigma \text{nombre} == \text{"Luis"}(\text{usuario}) \bowtie \text{cambio})$

$\pi \text{id\_requisito, descripcion} (\sigma \text{rol} == \text{"estudiante"}(\text{usuario}) \bowtie \text{requisito})$

Cambios realizados sobre requisitos funcionales:



$\pi \text{id\_cambio,justificacion} (\sigma \text{ tipo} == \text{"funcional"}(\text{requisito}) \bowtie \text{cambio})$

File	Relations	Settings
Operators		
*	-	id_cambio   justificacion
U	∩	1 Corrección descripción 2 Cambio de prioridad 4 Corrección ortográfica
◁	▷	
◀	▶	
÷		
σ	π	π id_cambio,fecha_cambio,justificacion,cambio_requisito (σ nombre=="Luis"(usuario) ▷ cambio) π id_requisito,descripcion (σ rol=="estudiante"(usuario) ▷ requisito)
ρ	→	π id_cambio,justificacion (σ tipo=="funcional"(requisito) ▷ cambio)

#### **Proyección de nombre y estado de proyectos activos:**

$\pi$  nombre,estado ( $\sigma$  (estado=="activo") (proyecto))

<b>Operators</b>	nombre	estado
*	-	SGR activo
		Evaluador UNL activo
U	∩	
⊗	⊗	
⊗	⊗	
÷		
σ	π	π id_cambio,fecha_cambio,justificacion,cambio_requisito (σ nombre=="Luis"(usuario) ⊗ cambio)
ρ	→	π id_requisito,descripcion (σ rol=="estudiante"(usuario) ⊗ requisito) π id_cambio,justificacion (σ tipo=="funcional"(requisito) ⊗ cambio) π nombre,estado (σ (estado=="activo") (proyecto))

Listar los requisitos creados por el usuario “Ana” y que tienen prioridad “baja”:

$\pi$  descripcion ( $\sigma$  nombre=="Ana"(usuario)  $\bowtie$   $\sigma$  prioridad=="baja"(requisito))

File		Relations	Settings	Help
Operators	*	descripcion		
	-	Exportar a PDF		
U	∩			
☒	☒			
☒	☒			
÷				
σ	π	π id_cambio,fecha_cambio,justificacion,cambio_requisito (σ nombre=="Luis"(usuario) ⋈ cambio)		
ρ	→	π id_requisito,descripcion (σ rol=="estudiante"(usuario) ⋈ requisito)		
		π id_cambio,justificacion (σ tipo=="funcional"(requisito) ⋈ cambio)		
		π nombre,estado (σ (estado=="activo") (proyecto))		
		π descripcion (σ nombre=="Ana"(usuario) ⋈ σ prioridad=="baja"(requisito))		

## 7.2 Consultas en SQL (traducción de consultas de álgebra relacional)

**Mostrar todos los cambios realizados por el usuario llamado “Luis”:**



**UNL**

Universidad  
Nacional  
de Loja

1859

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

Carrera Ingeniería en Computación

```
SELECT c.id_cambio, c.fecha_cambio, c.justificacion, c.cambio_requisito n FROM Usuario u
JOIN Cambio c ON u.id_usuario = c.id_usuario WHERE u.nombre = 'Luis';
```

id_cambio	fecha_cambio	justificacion	n
1	2025-07-02	Corrección descripción	Se actualizó el alcance

1 row in set (0.00 sec)

#### Listar los requisitos creados por estudiantes

```
SELECT r.id_requisito, r.descripcion FROM Usuario u JOIN Requisito r ON u.id_usuario =
r.id_usuario_creador WHERE u.rol = 'estudiante';
```

id_requisito	descripcion
1	Registrar requisitos
2	Seguridad de acceso
3	Exportar a PDF
4	Registrar usuarios
6	Generar reportes

5 rows in set (0.01 sec)

#### Cambios realizados sobre requisitos funcionales

```
SELECT c.id_cambio, c.justificacion FROM Requisito r JOIN Cambio c ON r.id_requisito = c.id_requisito
WHERE r.tipo = 'funcional';
```

id_cambio	justificacion
1	Corrección descripción
2	Cambio de prioridad
4	Corrección ortográfica

3 rows in set (0.00 sec)

#### Proyección de nombre y estado de proyectos activos



**UNL**

Universidad  
Nacional  
de Loja

SELECT nombre, estado FROM Proyecto WHERE estado = 'activo';

nombre	estado
SGR	activo
Evaluador UNL	activo

2 rows in set (0.00 sec)

**Listar los requisitos creados por el usuario “Ana” y que tienen prioridad “baja”**

SELECT r.descripcion FROM Usuario u JOIN Requisito r ON u.id\_usuario = r.id\_usuario\_creador  
WHERE u.nombre = 'Ana' AND r.prioridad = 'baja';

descripcion
Exportar a PDF

1 row in set (0.00 sec)

## 8. Views:

### 8.1 Creación de vistas por cada consulta.

**Mostrar todos los cambios realizados por el usuario llamado “Luis”:**

**Vista:**

CREATE VIEW CambiosLuis AS SELECT c.id\_cambio, c.fecha\_cambio, c.justificacion, c.cambio\_requisito  
FROM Usuario u JOIN Cambio c ON u.id\_usuario = c.id\_usuario WHERE u.nombre = 'Luis';

**Listar los requisitos creados por estudiantes**

**vista:**

CREATE VIEW RequisitosEstudiantes AS SELECT r.id\_requisito, r.descripcion FROM Usuario u JOIN  
Requisito r ON u.id\_usuario = r.id\_usuario\_creador WHERE u.rol = 'estudiante';

**Cambios realizados sobre requisitos funcionales**

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

Carrera Ingeniería en Computación



**UNL**

Universidad  
Nacional  
de Loja

1859

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

Carrera Ingeniería en Computación

**vista:**

```
CREATE VIEW CambiosFuncionales AS SELECT c.id_cambio, c.justificacion FROM Requisito r JOIN
Cambio c ON r.id_requisito = c.id_requisito WHERE r.tipo = 'funcional';
```

**Proyección de nombre y estado de proyectos activos**

**vista:**

```
CREATE VIEW ProyectosActivos AS SELECT nombre, estado FROM Proyecto WHERE estado = 'activo';
```

**Listar los requisitos creados por el usuario “Ana” y que tienen prioridad “baja”**

**vista:**

```
CREATE VIEW RequisitosAnaBaja AS SELECT r.descripcion
FROM Usuario u JOIN Requisito r ON u.id_usuario = r.id_usuario_creador WHERE u.nombre = 'Ana'
AND r.prioridad = 'baja';
```

**VIEWS:**

```
mysql> SHOW FULL TABLES WHERE Table_type = 'VIEW';
+-----+-----+
| Tables_in_gestion_requisitos | Table_type |
+-----+-----+
| CambiosFuncionales           | VIEW      |
| CambiosLuis                  | VIEW      |
| ProyectosActivos             | VIEW      |
| RequisitosAnaBaja            | VIEW      |
| RequisitosEstudiantes        | VIEW      |
+-----+-----+
5 rows in set (0.03 sec)
```

**5. Resultados Obtenidos**

- Agregar los productos de la práctica en github o gitlab
- Link repositorio: <https://github.com/Gerardo006/Restaurante.git>

**6. Conclusiones y Recomendaciones**

- Conclusiones:



**UNL**

Universidad  
Nacional  
de Loja

1859

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

Carrera Ingeniería en Computación

- La práctica preprofesional permitió reforzar los conocimientos teóricos adquiridos en la asignatura de Base de Datos, aplicándolos en actividades concretas como el modelado, diseño, normalización y consultas en SQL.
- Se logró una mejor comprensión del funcionamiento de los Sistemas de Gestión de Bases de Datos (SGBD), especialmente MySQL, mediante el uso de comandos reales para la creación de estructuras, inserción de datos y generación de vistas.
- El trabajo realizado facilitó el desarrollo de competencias técnicas, tales como el análisis de requerimientos, la construcción de modelos entidad-relación y la traducción de estos a modelos físicos implementables.

- Recomendaciones:

- Continuar fortaleciendo las prácticas con escenarios más cercanos a la realidad, incluyendo casos de uso de mayor complejidad o proyectos integradores entre varias asignaturas.

## 7. Firma y Validación

**Estudiante**

**Docente Supervisor**

---

[Nombre y

[Nombre y Firma]

Firma]

---