



Universidad
Nacional
de Loja

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

Carrera Ingeniería en Computación

INFORME DE PRÁCTICAS PREPROFESIONALES EN BASE DE DATOS



Datos de los estudiantes:

- **Nombre:** Miguel Angel Veintimilla Esparza, Richard Vicente Cajas Riofrío
- **Carrera:** Ingeniería en Ciencias Computacionales

Docente supervisor:

- **Nombre:** Ing. Rene Guamán
- **Cargo:** Docente de la Materia Base de Datos

Fecha de entrega:

- **Fecha:** Miércoles 30 de Julio del 2025



A. Datos Informativos		
Asignatura: Base de datos	Semestre: Tercero	Paralelo: A
Docente Supervisor: René Guamán-Quinche	Periodo Académico: Abril - Agosto de 2025	
Estudiante: Miguel Angel Veintimilla Esparza, Richard Vicente Cajas Riofrío		
Periodo de prácticas: Marzo - Agosto de 2025		

2. Introducción

- **Objetivo de las prácticas preprofesionales.**

Estas prácticas preprofesionales constituyen una etapa fundamental en nuestra formación académica, ya que nos permiten aplicar los conocimientos adquiridos durante el ciclo en un entorno real de trabajo. A través de estas prácticas, fortalecemos nuestro análisis, diseño y desarrollo de sistemas informáticos, así como en la resolución de problemas técnicos y la toma de decisiones.

El objetivo principal de estas prácticas fue participar activamente en el desarrollo de un sistema de gestión de pruebas de software, aportando soluciones tecnológicas basadas en bases de datos para optimizar el seguimiento, control y validación de los procesos de prueba dentro de un proyecto de desarrollo.

Las actividades las llevamos a cabo en el entorno de nuestras casas y del aula donde nos imparten las clases, con las asesorías de nuestro docente a cargo que nos iba ayudando o diciendo como podemos realizar las siguiente practica o también que nos iba corrigiendo los errores que presentábamos al momento de presentar los avances

En este contexto, el uso de bases de datos desempeñó un papel clave, ya que permitió modelar y almacenar la información relacionada con los usuarios, proyectos, requisitos, casos de prueba, ejecuciones y roles de manera estructurada. Las bases de datos no solo facilitaron el almacenamiento y recuperación eficiente de la información, sino que también fueron fundamentales para generar reportes, validar resultados y automatizar tareas relacionadas con la trazabilidad de las pruebas. Esto evidencia la importancia de un diseño sólido y funcional de base de datos en entornos donde la calidad del software es prioritaria.

3. Objetivos

3.1 Objetivo General

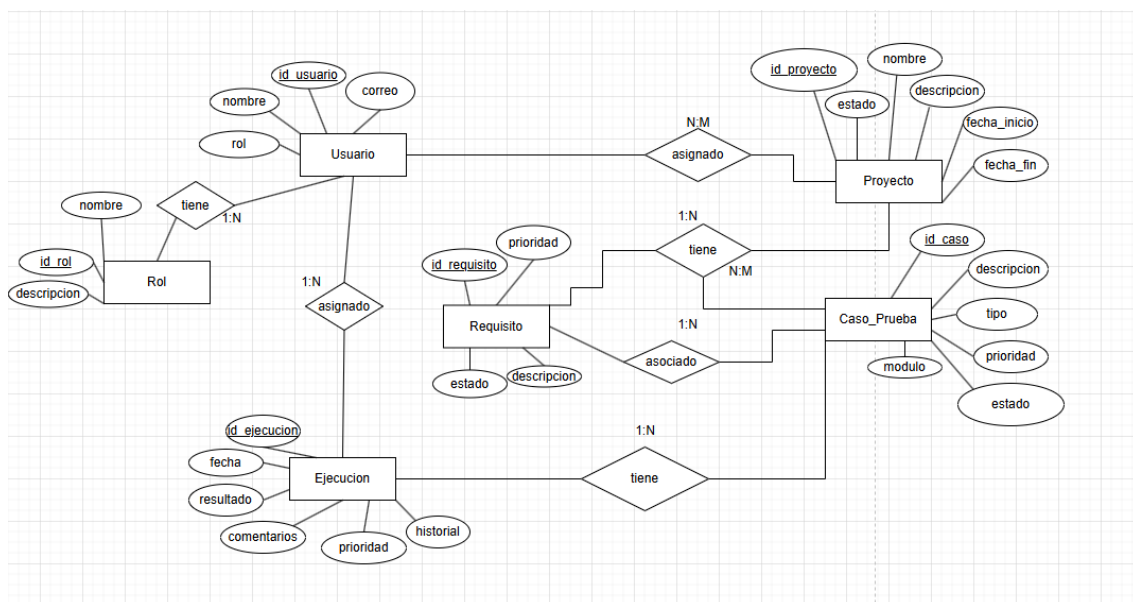
- Participar activamente en el desarrollo de un sistema de gestión de pruebas de software, aplicando conocimientos de análisis, diseño de bases de datos y desarrollo de sistemas informáticos, con el fin de fortalecer las competencias profesionales y contribuir a los procesos de control de calidad en el desarrollo de software.

3.2 Objetivos Específicos

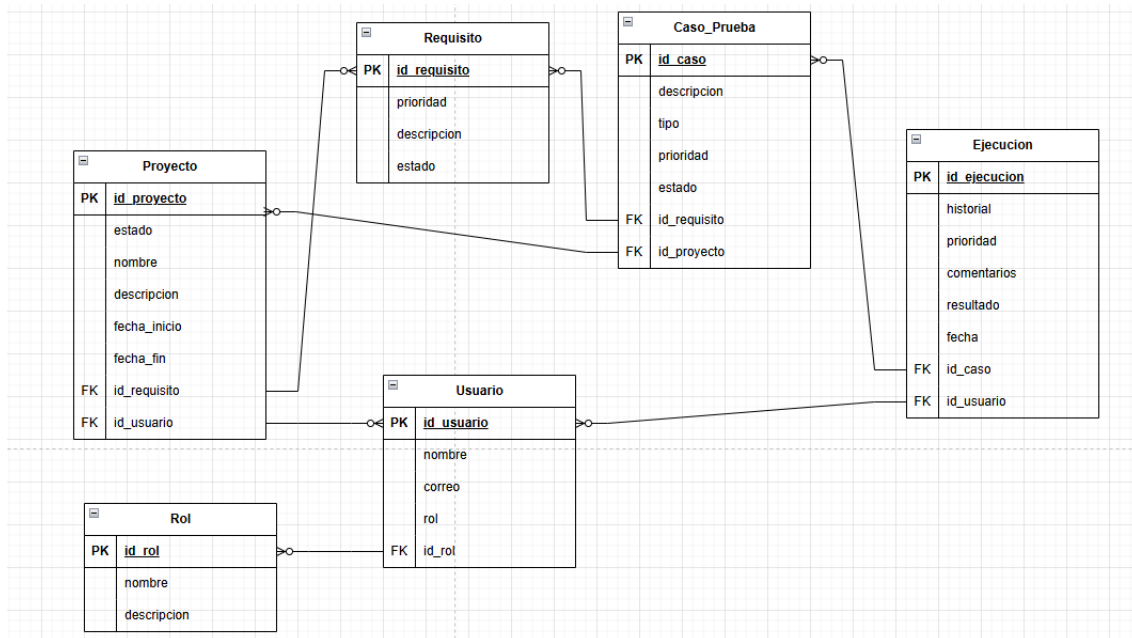
- Aplicar conceptos de modelado entidad-relación para representar los procesos y entidades clave involucradas en la gestión de pruebas.
- Diseñar e implementar una base de datos relacional que permita almacenar y administrar la información de pruebas de manera eficiente.
- Utilizar lenguaje SQL para la creación de tablas, inserción de datos y elaboración de vistas que faciliten la consulta de información relevante.
- Comprender y participar en las actividades del área de calidad de software, especialmente en lo relacionado con el seguimiento de pruebas y validación de requisitos.

4. Actividades Realizadas

- **Modelo entidad – relación**



- **Modelo Relacional: traducción del modelo entidad - relación**



- Detalle de normalización de la base de datos
- Diseño físico: Uso del SGBD (MySQL)
 - Creación de de base de datos, tabla

mysql> create table Rol (

-> id_rol INT auto_increment PRIMARY KEY,
-> nombre VARCHAR(50),
-> descripcion TEXT
->);

Query OK, 0 rows affected (0.32 sec)

```
mysql> create table Rol (
    -> id_rol INT auto_increment PRIMARY KEY,
    -> nombre VARCHAR(50),
    -> descripcion TEXT
    -> );
```

Query OK, 0 rows affected (0.32 sec)

mysql> describe Rol;

Field	Type	Null	Key	Default	Extra
id_rol	int	NO	PRI	NULL	auto_increment
nombre	varchar(50)	YES		NULL	
descripcion	text	YES		NULL	

3 rows in set (0.07 sec)

mysql> CREATE TABLE Usuario (

-> id_usuario INT auto_increment PRIMARY KEY,
-> nombre VARCHAR(50) NOT NULL,
-> correo VARCHAR(50) NOT NULL,
-> rol VARCHAR(50),
-> id_rol INT,
-> FOREIGN KEY (id_rol) REFERENCES Rol(id_rol)

->);

Query OK, 0 rows affected (0.31 sec)

```
mysql> CREATE TABLE Usuario (  
  -> id_usuario INT auto_increment PRIMARY KEY,  
  -> nombre VARCHAR(50) NOT NULL,  
  -> correo VARCHAR(50) NOT NULL,  
  -> rol VARCHAR(50),  
  -> id_rol INT,  
  -> FOREIGN KEY (id_rol) REFERENCES Rol(id_rol)  
  -> );
```

Query OK, 0 rows affected (0.31 sec)

```
mysql> describe Usuario;
```

Field	Type	Null	Key	Default	Extra
id_usuario	int	NO	PRI	NULL	auto_increment
nombre	varchar(50)	NO		NULL	
correo	varchar(50)	NO		NULL	
rol	varchar(50)	YES		NULL	
id_rol	int	YES	MUL	NULL	

5 rows in set (0.00 sec)

```
mysql> CREATE TABLE Requisito (  
  -> id_requisito INT auto_increment PRIMARY KEY,  
  -> prioridad VARCHAR(20),  
  -> descripcion TEXT,  
  -> estado VARCHAR(20)  
  -> );
```

Query OK, 0 rows affected (0.15 sec)

```
mysql> CREATE TABLE Requisito (  
  -> id_requisito INT auto_increment PRIMARY KEY,  
  -> prioridad VARCHAR(20),  
  -> descripcion TEXT,  
  -> estado VARCHAR(20)  
  -> );
```

Query OK, 0 rows affected (0.15 sec)

```
mysql> describe Requisito;
```

Field	Type	Null	Key	Default	Extra
id_requisito	int	NO	PRI	NULL	auto_increment
prioridad	varchar(20)	YES		NULL	
descripcion	text	YES		NULL	
estado	varchar(20)	YES		NULL	

4 rows in set (0.00 sec)

```
mysql> CREATE TABLE Proyecto (  
  -> id_proyecto INT auto_increment PRIMARY KEY,
```



```
-> estado VARCHAR(20),  
-> nombre VARCHAR(50) NOT NULL,  
-> descripcon TEXT,  
-> fecha_inicio DATE,  
-> fecha_fin DATE,  
-> id_requisito INT,  
-> id_usuario INT,  
-> FOREIGN KEY (id_requisito) REFERENCES Requisito(id_requisito),  
-> FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)  
-> );
```

Query OK, 0 rows affected (0.18 sec)

```
mysql> CREATE TABLE Proyecto (  
-> id_proyecto INT auto_increment PRIMARY KEY,  
-> estado VARCHAR(20),  
-> nombre VARCHAR(50) NOT NULL,  
-> descripcon TEXT,  
-> fecha_inicio DATE,  
-> fecha_fin DATE,  
-> id_requisito INT,  
-> id_usuario INT,  
-> FOREIGN KEY (id_requisito) REFERENCES Requisito(id_requisito),  
-> FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)  
-> );
```

Query OK, 0 rows affected (0.18 sec)

```
mysql> describe Proyecto;
```

Field	Type	Null	Key	Default	Extra
id_proyecto	int	NO	PRI	NULL	auto_increment
estado	varchar(20)	YES		NULL	
nombre	varchar(50)	NO		NULL	
descripcon	text	YES		NULL	
fecha_inicio	date	YES		NULL	
fecha_fin	date	YES		NULL	
id_requisito	int	YES	MUL	NULL	
id_usuario	int	YES	MUL	NULL	

8 rows in set (0.01 sec)

```
mysql> CREATE TABLE Caso_Prueba (  
-> id_caso INT auto_increment PRIMARY KEY,  
-> descripcion TEXT,  
-> tipo VARCHAR(30),  
-> prioridad VARCHAR(20),  
-> estado VARCHAR(20),  
-> id_requisito INT,  
-> id_proyecto INT,  
-> FOREIGN KEY (id_requisito) REFERENCES Requisito(id_requisito),  
-> FOREIGN KEY (id_proyecto) REFERENCES Proyecto(id_proyecto)  
-> );
```

Query OK, 0 rows affected (0.11 sec)



```
mysql> CREATE TABLE Caso_Prueba (  
  -> id_caso INT auto_increment PRIMARY KEY,  
  -> descripcion TEXT,  
  -> tipo VARCHAR(30),  
  -> prioridad VARCHAR(20),  
  -> estado VARCHAR(20),  
  -> id_requisito INT,  
  -> id_proyecto INT,  
  -> FOREIGN KEY (id_requisito) REFERENCES Requisito(id_requisito),  
  -> FOREIGN KEY (id_proyecto) REFERENCES Proyecto(id_proyecto)  
  -> );  
Query OK, 0 rows affected (0.11 sec)
```

```
mysql> describe Caso_Prueba;
```

Field	Type	Null	Key	Default	Extra
id_caso	int	NO	PRI	NULL	auto_increment
descripcion	text	YES		NULL	
tipo	varchar(30)	YES		NULL	
prioridad	varchar(20)	YES		NULL	
estado	varchar(20)	YES		NULL	
id_requisito	int	YES	MUL	NULL	
id_proyecto	int	YES	MUL	NULL	

```
7 rows in set (0.01 sec)
```

```
mysql> CREATE TABLE Ejecucion (  
  -> id_ejecucion INT auto_increment PRIMARY KEY,  
  -> historial TEXT,  
  -> prioridad VARCHAR(20),  
  -> comentarios TEXT,  
  -> resultado VARCHAR(20),  
  -> fecha DATE,  
  -> id_caso INT,  
  -> id_usuario INT,  
  -> FOREIGN KEY (id_caso) REFERENCES Caso_Prueba(id_caso),  
  -> FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)  
  -> );
```

```
Query OK, 0 rows affected (0.21 sec)
```

```
mysql> CREATE TABLE Ejecucion (  
  -> id_ejecucion INT auto_increment PRIMARY KEY,  
  -> historial TEXT,  
  -> prioridad VARCHAR(20),  
  -> comentarios TEXT,  
  -> resultado VARCHAR(20),  
  -> fecha DATE,  
  -> id_caso INT,  
  -> id_usuario INT,  
  -> FOREIGN KEY (id_caso) REFERENCES Caso_Prueba(id_caso),  
  -> FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario)  
  -> );
```

```
Query OK, 0 rows affected (0.21 sec)
```




```
mysql> describe Ejecucion;
```

Field	Type	Null	Key	Default	Extra
id_ejecucion	int	NO	PRI	NULL	auto_increment
historial	text	YES		NULL	
prioridad	varchar(20)	YES		NULL	
comentarios	text	YES		NULL	
resultado	varchar(20)	YES		NULL	
fecha	date	YES		NULL	
id_caso	int	YES	MUL	NULL	
id_usuario	int	YES	MUL	NULL	

```
3 rows in set (0.00 sec)
```

○ **Inserciones, modificaciones y actualizaciones**

```
mysql> INSERT INTO Rol (nombre, descripcion) VALUES
```

```
-> ('Administrador', 'Gestión total del sistema'),  
-> ('Tester', 'Realiza pruebas funcionales y de regresión'),  
-> ('Desarrollador', 'Implementa funcionalidades y corrige errores')  
-> ;
```

Query OK, 3 rows affected (0.04 sec)

Records: 3 Duplicates: 0 Warnings: 0

```
mysql> INSERT INTO Rol (nombre, descripcion) VALUES  
-> ('Administrador', 'Gestión total del sistema'),  
-> ('Tester', 'Realiza pruebas funcionales y de regresión'),  
-> ('Desarrollador', 'Implementa funcionalidades y corrige errores')  
;  
Query OK, 3 rows affected (0.04 sec)  
Records: 3 Duplicates: 0 Warnings: 0
```

```
mysql> select * from Rol;
```

id_rol	nombre	descripcion
1	Administrador	Gestión total del sistema
2	Tester	Realiza pruebas funcionales y de regresión
3	Desarrollador	Implementa funcionalidades y corrige errores

```
3 rows in set (0.04 sec)
```

```
mysql> INSERT INTO Usuario (nombre, correo, rol, id_rol) VALUES
```

```
-> ('Ana', 'ana@gmail.com', 'Administrador', 1),  
-> ('Carlos', 'carlos@gmail.com', 'Tester', 2),  
-> ('Luis', 'luis@gmail.com', 'Desarrollador', 3),  
-> ('Marta', 'marta@gmail.com', 'Tester', 2);
```

Query OK, 4 rows affected (0.02 sec)

Records: 4 Duplicates: 0 Warnings: 0



```
mysql> INSERT INTO Usuario (nombre, correo, rol, id_rol) VALUES
-> ('Ana', 'ana@gmail.com', 'Administrador', 1),
-> ('Carlos', 'carlos@gmail.com', 'Tester', 2),
-> ('Luis', 'luis@gmail.com', 'Desarrollador', 3),
-> ('Marta', 'marta@gmail.com', 'Tester', 2);
Query OK, 4 rows affected (0.02 sec)
Records: 4 Duplicates: 0 Warnings: 0
```

```
mysql> select * from Usuario;
+-----+-----+-----+-----+-----+
| id_usuario | nombre | correo | rol | id_rol |
+-----+-----+-----+-----+-----+
| 1 | Ana | ana@gmail.com | Administrador | 1 |
| 2 | Carlos | carlos@gmail.com | Tester | 2 |
| 3 | Luis | luis@gmail.com | Desarrollador | 3 |
| 4 | Marta | marta@gmail.com | Tester | 2 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

```
mysql> INSERT INTO Requisito (prioridad, descripcion, estado) VALUES
-> ('Alta', 'Login de usuario', 'Completado'),
-> ('Media', 'Registro de nuevo cliente', 'En progreso'),
-> ('Baja', 'Visualización de reportes', 'Pendiente');
Query OK, 3 rows affected (0.03 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```
mysql> INSERT INTO Requisito (prioridad, descripcion, estado) VALUES
-> ('Alta', 'Login de usuario', 'Completado'),
-> ('Media', 'Registro de nuevo cliente', 'En progreso'),
-> ('Baja', 'Visualización de reportes', 'Pendiente');
Query OK, 3 rows affected (0.03 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```
mysql> select * from Requisito;
+-----+-----+-----+-----+
| id_requisito | prioridad | descripcion | estado |
+-----+-----+-----+-----+
| 1 | Alta | Login de usuario | Completado |
| 2 | Media | Registro de nuevo cliente | En progreso |
| 3 | Baja | Visualización de reportes | Pendiente |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> INSERT INTO Proyecto (estado, nombre, descripcion, fecha_inicio, fecha_fin,
id_requisito, id_usuario ) VALUES
-> ('Activo', 'Sistema de Gestión Escolar', 'Proyecto para administrar estudiantes y
profesores', '2023-03-01', '2023-10-30', 1, 1),
-> ('Finalizado', 'App de Ventas', 'Aplicación móvil para ventas en
línea', '2023-01-15', '2023-06-20', 2, 3);
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0
```



```
mysql> INSERT INTO Proyecto (estado, nombre, descripcon, fecha_inicio, fecha_fin, id_requisito, id_usuario) VALUES
-> ('Activo', 'Sistema de Gestión Escolar', 'Proyecto para administrar estudiantes y profesores', '2023-03-01', '2023-10-30', 1, 1),
-> ('Finalizado', 'App de Ventas', 'Aplicación móvil para ventas en línea', '2023-01-15', '2023-06-20', 2, 3);
Query OK, 2 rows affected (0.02 sec)
Records: 2 Duplicates: 0 Warnings: 0
```

```
mysql> select * from Proyecto;
```

id_proyecto	estado	nombre	descripcon	fecha_inicio	fecha_fin	id_requisito	id_usuario
1	Activo	Sistema de Gestión Escolar	Proyecto para administrar estudiantes y profesores	2023-03-01	2023-10-30	1	1
2	Finalizado	App de Ventas	Aplicación móvil para ventas en línea	2023-01-15	2023-06-20	2	3

```
2 rows in set (0.02 sec)
```

```
mysql> INSERT INTO Caso_Prueba (descripcion, tipo, prioridad, estado, id_requisito, id_proyecto) VALUES
```

```
-> ('Validar login con credenciales válidas', 'Funcional', 'Alta', 'Finalizado', 1, 1),
-> ('Validar login con contraseña incorrecta', 'Funcional', 'Alta', 'Finalizado', 1, 1),
-> ('Registrar cliente nuevo', 'Funcional', 'Media', 'En ejecución', 2, 2),
-> ('Registrar cliente con datos incompletos', 'Funcional', 'Media', 'Pendiente', 2, 2),
-> ('Visualizar reporte diario', 'UI', 'Baja', 'Pendiente', 3, 2),
-> ('Acceder sin login', 'Seguridad', 'Alta', 'Finalizado', 1, 1);
```

Query OK, 6 rows affected (0.19 sec)

Records: 6 Duplicates: 0 Warnings: 0

```
mysql> INSERT INTO Caso_Prueba (descripcion, tipo, prioridad, estado, id_requisito, id_proyecto) VALUES
-> ('Validar login con credenciales válidas', 'Funcional', 'Alta', 'Finalizado', 1, 1),
-> ('Validar login con contraseña incorrecta', 'Funcional', 'Alta', 'Finalizado', 1, 1),
-> ('Registrar cliente nuevo', 'Funcional', 'Media', 'En ejecución', 2, 2),
-> ('Registrar cliente con datos incompletos', 'Funcional', 'Media', 'Pendiente', 2, 2),
-> ('Visualizar reporte diario', 'UI', 'Baja', 'Pendiente', 3, 2),
-> ('Acceder sin login', 'Seguridad', 'Alta', 'Finalizado', 1, 1);
Query OK, 6 rows affected (0.19 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

```
mysql> select * from Caso_Prueba;
```

id_caso	descripcion	tipo	prioridad	estado	id_requisito	id_proyecto
1	Validar login con credenciales válidas	Funcional	Alta	Finalizado	1	1
2	Validar login con contraseña incorrecta	Funcional	Alta	Finalizado	1	1
3	Registrar cliente nuevo	Funcional	Media	En ejecución	2	2
4	Registrar cliente con datos incompletos	Funcional	Media	Pendiente	2	2
5	Visualizar reporte diario	UI	Baja	Pendiente	3	2
6	Acceder sin login	Seguridad	Alta	Finalizado	1	1

```
6 rows in set (0.01 sec)
```

```
mysql> INSERT INTO Ejecucion (historial, prioridad, comentarios, resultado, fecha, id_caso, id_usuario) VALUES
```



-> ('Ejecución exitosa del login', 'Alta', 'Todo funcionó correctamente', 'Exitoso', '2023-05-10', 1, 2),
-> ('Falla por contraseña incorrecta', 'Alta', 'Se mostró mensaje de error', 'Fallido', '2023-05-12', 2, 2),
-> ('Cliente registrado correctamente', 'Media', 'Datos almacenados en la base', 'Exitoso', '2023-06-01', 3, 4),
-> ('Error por falta de datos', 'Media', 'Validación no activada', 'Fallido', '2023-06-02', 4, 4),
-> ('Diseño mal renderizado', 'Baja', 'Botones desalineados', 'Fallido', '2023-06-03', 5, 4),
-> ('Acceso denegado correctamente', 'Alta', 'El sistema bloqueó correctamente', 'Exitoso', '2023-06-04', 6, 2);

Query OK, 6 rows affected (0.08 sec)

Records: 6 Duplicates: 0 Warnings: 0

```
mysql> INSERT INTO Ejecucion (historial, prioridad, comentarios, resultado, fecha, id_caso, id_usuario) VALUES
-> ('Ejecución exitosa del login', 'Alta', 'Todo funcionó correctamente', 'Exitoso', '2023-05-10', 1, 2),
-> ('Falla por contraseña incorrecta', 'Alta', 'Se mostró mensaje de error', 'Fallido', '2023-05-12', 2, 2),
-> ('Cliente registrado correctamente', 'Media', 'Datos almacenados en la base', 'Exitoso', '2023-06-01', 3, 4),
-> ('Error por falta de datos', 'Media', 'Validación no activada', 'Fallido', '2023-06-02', 4, 4),
-> ('Diseño mal renderizado', 'Baja', 'Botones desalineados', 'Fallido', '2023-06-03', 5, 4),
-> ('Acceso denegado correctamente', 'Alta', 'El sistema bloqueó correctamente', 'Exitoso', '2023-06-04', 6, 2);
Query OK, 6 rows affected (0.08 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

```
mysql> select * from Ejecucion;
```

id_ejecucion	historial	prioridad	comentarios	resultado	fecha	id_caso	id_usuario
1	Ejecución exitosa del login	Alta	Todo funcionó correctamente	Exitoso	2023-05-10	1	2
2	Falla por contraseña incorrecta	Alta	Se mostró mensaje de error	Fallido	2023-05-12	2	2
3	Cliente registrado correctamente	Media	Datos almacenados en la base	Exitoso	2023-06-01	3	4
4	Error por falta de datos	Media	Validación no activada	Fallido	2023-06-02	4	4
5	Diseño mal renderizado	Baja	Botones desalineados	Fallido	2023-06-03	5	4
6	Acceso denegado correctamente	Alta	El sistema bloqueó correctamente	Exitoso	2023-06-04	6	2

6 rows in set (0.00 sec)

- Consultas y reportes:

- Consultas en álgebra relacional

1. Obtener todas las ejecuciones exitosas

$\sigma_{\text{resultado} = \text{'Exitoso'}}(\text{ejecucion})$

The screenshot shows a Relational database query editor. The query is $\sigma_{\text{resultado} = \text{'Exitoso'}}(\text{ejecucion})$. The results are displayed in a table with 8 columns: id_ejecucion, historial, prioridad, comentarios, resultado, fecha, id_caso, and id_usuario. The results are the same as the data in the Ejecucion table, but only the rows where the resultado is 'Exitoso' are shown.

id_ejecucion	historial	prioridad	comentarios	resultado	fecha	id_caso	id_usuario
1	Ejecución exitosa del login	Alta	Todo funcionó correctamente	Exitoso	2023-05-10	1	2
6	Acceso denegado correctamente	Alta	El sistema bloqueó correctamente	Exitoso	2023-06-04	6	2

2. Casos de prueba tipo 'Funcional' y prioridad 'Alta'

σ (tipo == 'Funcional') (σ (prioridad == 'Alta') (caso_prueba))

Relational							
File	Relations	Settings	Help				
Operators				id_caso	descripcion	tipo	prioridad
*	-			1	Validar login con credenciales válidas	Funcional	Alta
u	n			2	Validar login con contraseña incorrecta	Funcional	Alta

3. Casos Obtener los nombres de los casos de prueba con estado pendientes junto con el nombre del proyecto al que pertenecen

π nombre_caso, nombre_proyecto (ρ descripcion \rightarrow nombre_caso (σ estado == 'Pendiente' (caso_prueba)) \bowtie ρ nombre \rightarrow nombre_proyecto (π id_proyecto, nombre (proyecto)))

Relational							
File	Relations	Settings	Help				
Operators				nombre_caso	nombre_proyecto		
*	-			Registrar cliente con datos incompletos	App de Ventas		
u	n			Visualizar reporte diario	App de Ventas		

4. Casos de prueba y su ejecución con resultado fallido

π id_caso, descripcion, resultado(σ resultado == 'Fallido' (ejecucion \bowtie caso_prueba))

File	Relations	Settings	Help				
Operators				id_caso	descripcion	resultado	
*	-			4	Registrar cliente con datos incompletos	Fallido	
u	n			5	Visualizar reporte diario	Fallido	
				2	Validar login con contraseña incorrecta	Fallido	

5. Usuarios que han ejecutado al menos un caso de prueba

π nombre, correo ((σ rol == 'Tester' (usuario)) \bowtie (σ resultado == 'Exitoso', prioridad == 'Alta' (ejecucion)) \bowtie (σ tipo == 'Funcional' (caso_prueba)))

Relational							
File	Relations	Settings	Help				
Operators				nombre	correo		
*	-			Marta Ruiz	marta.ruiz@mail.com		
u	n			Carlos Pérez	carlos.perez@mail.com		

6. Usuarios con rol 'Tester' que hayan ejecutado al menos un caso con resultado 'Exitoso'

π nombre (σ (rol == 'Tester') (usuario) \bowtie σ (resultado == 'Exitoso') (ejecucion)))

File	Relations	Settings	Help
Operators		nombre	
*	-	Marta Ruiz	
u	n	Carlos Pérez	
⋈	⋈		

7. Descripción de caso de prueba con requisito con estado en progreso

$\pi_{desc} (\rho \text{ descripcion} \rightarrow desc(\text{caso_prueba}) \bowtie \pi_{id_requisito} (\sigma \text{ estado} == 'En \text{ progreso}'(\text{requisito})))$

Operators	desc
*	-
u	n
⋈	⋈
	Registrar cliente nuevo
	Registrar cliente con datos incompletos

8. Nombre de los usuarios y su cantidad de ejecuciones realizadas

$\pi_{nombre}(\text{usuario} \bowtie \text{ejecucion})$

Operators	nombre
*	-
u	n
⋈	⋈
	Carlos
	Marta

9. Obtener nombre del proyecto y nombre del usuario responsable

$\pi_{proyecto_nombre, usuario_nombre} (\rho \text{ nombre} \rightarrow proyecto_nombre (\text{proyecto}) \bowtie \rho \text{ nombre} \rightarrow usuario_nombre (\text{usuario}))$

Operators	proyecto_nombre	usuario_nombre
*	-	
u	n	
⋈	⋈	
	Sistema de Gestión Escolar	Ana
	App de Ventas	Luis

10. Casos de prueba y el nombre del proyecto al que pertenecen

$\pi_{desc, nombre} (\rho \text{ descripcion} \rightarrow desc(\text{caso_prueba}) \bowtie \pi_{id_proyecto, nombre} (\text{proyecto}))$

Operators	desc	nombre
*	Validar login con contraseña incorrecta	Sistema de Gestión Escolar
-	Registrar cliente nuevo	App de Ventas
U	Visualizar reporte diario	App de Ventas
∩	Acceder sin login	Sistema de Gestión Escolar
⋈	Registrar cliente con datos incompletos	App de Ventas
⋈	Validar login con credenciales válidas	Sistema de Gestión Escolar

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

1. select * from Ejecucion WHERE resultado = 'Exitoso';

```
mysql> select * from Ejecucion WHERE resultado = 'Exitoso';
```

id_ejecucion	historial	prioridad	comentarios	resultado	fecha	id_caso	id_usuario
1	Ejecución exitosa del login	Alta	Todo funcionó correctamente	Exitoso	2023-05-10	1	2
3	Cliente registrado correctamente	Media	Datos almacenados en la base	Exitoso	2023-06-01	3	4
6	Acceso denegado correctamente	Alta	El sistema bloqueó correctamente	Exitoso	2023-06-04	6	2

3 rows in set (0.01 sec)

2. select * from Caso_Prueba WHERE prioridad = 'Alta' AND tipo = 'Funcional';

```
mysql> select * from Caso_Prueba WHERE prioridad = 'Alta' AND tipo = 'Funcional';
```

id_caso	descripcion	tipo	prioridad	estado	id_requisito	id_proyecto
1	Validar login con credenciales válidas	Funcional	Alta	Finalizado	1	1
2	Validar login con contraseña incorrecta	Funcional	Alta	Finalizado	1	1

2 rows in set (0.06 sec)

3. SELECT Caso_Prueba.descripcion AS nombre_caso, Proyecto.nombre AS nombre_proyecto FROM Caso_Prueba JOIN Proyecto ON Caso_Prueba.id_proyecto = Proyecto.id_proyecto WHERE Caso_Prueba.estado = 'Pendiente';

```
mysql> select * from Caso_Prueba WHERE estado = 'Pendiente';
```

id_caso	descripcion	tipo	prioridad	estado	id_requisito	id_proyecto
4	Registrar cliente con datos incompletos	Funcional	Media	Pendiente	2	2
5	Visualizar reporte diario	UI	Baja	Pendiente	3	2

2 rows in set (0.04 sec)

4. SELECT Caso_Prueba.id_caso, Caso_Prueba.descripcion, Ejecucion.resultado FROM Ejecucion JOIN Caso_Prueba ON Ejecucion.id_caso = Caso_Prueba.id_caso WHERE Ejecucion.resultado = 'Fallido';

```
mysql> SELECT Caso_Prueba.id_caso, Caso_Prueba.descripcion, Ejecucion.resultado FROM Ejecucion JOIN Caso_Prueba ON Ejecucion.id_caso = Caso_Prueba.id_caso WHERE Ejecucion.resultado = 'Fallido';
```

id_caso	descripcion	resultado
2	Validar login con contraseña incorrecta	Fallido
4	Registrar cliente con datos incompletos	Fallido
5	Visualizar reporte diario	Fallido

3 rows in set (0.00 sec)

5. SELECT Usuario.nombre, Usuario.correo FROM Usuario JOIN Ejecucion ON Usuario.id_usuario = Ejecucion.id_usuario;

```
mysql> SELECT Usuario.nombre, Usuario.correo FROM Usuario JOIN Ejecucion
ON Usuario.id_usuario = Ejecucion.id_usuario;
+-----+-----+
| nombre | correo |
+-----+-----+
| Carlos | carlos@gmail.com |
| Carlos | carlos@gmail.com |
| Carlos | carlos@gmail.com |
| Marta  | marta@gmail.com  |
| Marta  | marta@gmail.com  |
| Marta  | marta@gmail.com  |
+-----+-----+
6 rows in set (0.00 sec)
```

6. SELECT DISTINCT u.nombre FROM Usuario u JOIN Ejecucion e ON u.id_usuario = e.id_usuario WHERE u.rol = 'Tester' AND e.resultado = 'Exitoso';

```
mysql> SELECT DISTINCT u.nombre FROM Usuario u JOIN Ejecucion e ON u.id_
usuario = e.id_usuario WHERE u.rol = 'Tester' AND e.resultado = 'Exitoso
';
+-----+
| nombre |
+-----+
| Carlos |
| Marta  |
+-----+
2 rows in set (0.08 sec)
```

7. SELECT cp.descripcion AS `desc` FROM Caso_Prueba cp JOIN (SELECT DISTINCT id_requisito FROM Requisito WHERE estado = 'En progreso') AS requisitos_en_progreso ON cp.id_requisito = requisitos_en_progreso.id_requisito;

```
mysql> SELECT cp.descripcion AS `desc` FROM Caso_Prueba cp JOIN ( SELECT DISTINCT id_requisito FROM Requisito WHERE es
tado = 'En progreso' ) AS requisitos_en_progreso ON cp.id_requisito = requisitos_en_progreso.id_requisito;
+-----+
| desc |
+-----+
| Registrar cliente nuevo |
| Registrar cliente con datos incompletos |
+-----+
2 rows in set, 1 warning (0.18 sec)
```

8. SELECT DISTINCT Usuario.nombre FROM Usuario JOIN Ejecucion ON Usuario.id_usuario = Ejecucion.id_usuario;

```
MariaDB [SistemaGP]> SELECT DISTINCT Usuario.nombre FROM Usuario JOIN Ejecucion O
N Usuario.id_usuario = Ejecucion.id_usuario;
+-----+
| nombre |
+-----+
| Carlos |
| Marta  |
+-----+
2 rows in set (0.006 sec)
```

9. SELECT Proyecto.nombre AS proyecto_nombre, Usuario.nombre AS usuario_nombre FROM Proyecto JOIN Usuario ON Proyecto.id_usuario = Usuario.id_usuario;


```
MariaDB [SistemaGP]> SELECT DISTINCT Proyecto.nombre AS proyecto_nombre, Usuario.nombre AS usuario_nombre FROM Proyecto JOIN Usuario ON Proyecto.id_usuario = Usuario.id_usuario;
```

proyecto_nombre	usuario_nombre
Sistema de Gestión Escolar	Ana
App de Ventas	Luis

2 rows in set (0.005 sec)

10. SELECT Caso_Prueba.descripcion AS descripcion_caso, Proyecto.nombre FROM Caso_Prueba JOIN Proyecto ON Caso_Prueba.id_proyecto = Proyecto.id_proyecto;

```
MariaDB [SistemaGP]> SELECT Caso_Prueba.descripcion AS descripcion_caso, Proyecto.nombre FROM Caso_Prueba JOIN Proyecto ON Caso_Prueba.id_proyecto = Proyecto.id_proyecto;
```

descripcion_caso	nombre
Validar login con credenciales válidas	Sistema de Gestión Escolar
Validar login con contraseña incorrecta	Sistema de Gestión Escolar
Registrar cliente nuevo	App de Ventas
Registrar cliente con datos incompletos	App de Ventas
Visualizar reporte diario	App de Ventas
Acceder sin login	Sistema de Gestión Escolar

6 rows in set (0.020 sec)

- Views:

- Creación de vistas por cada consulta.

1. CREATE VIEW vista_ejecu_exitosas AS SELECT * FROM Ejecucion WHERE resultado = 'Exitoso';

```
mysql> CREATE VIEW vista_ejecu_exitosas AS SELECT * FROM Ejecucion WHERE resultado = 'Exitoso';
Query OK, 0 rows affected (0.05 sec)

mysql> SELECT * FROM vista_ejecu_exitosas;
```

id_ejecucion	historial	prioridad	comentarios	resultado	fecha	id_caso	id_usuario
1	Ejecución exitosa del login	Alta	Todo funcionó correctamente	Exitoso	2023-05-10	1	2
3	Cliente registrado correctamente	Media	Datos almacenados en la base	Exitoso	2023-06-01	3	4
6	Acceso denegado correctamente	Alta	El sistema bloqueó correctamente	Exitoso	2023-06-04	6	2

3 rows in set (0.01 sec)

2. CREATE VIEW vista_casos_funcional_alto AS SELECT * FROM Caso_Prueba WHERE prioridad = 'Alta' AND tipo = 'Funcional';

```
mysql> CREATE VIEW vista_casos_funcional_alto AS SELECT * FROM Caso_Prueba WHERE prioridad = 'Alta' AND tipo = 'Funcional';
Query OK, 0 rows affected (0.04 sec)

mysql> SELECT * FROM vista_casos_funcional_alto;
```

id_caso	descripcion	tipo	prioridad	estado	id_requisito	id_proyecto
1	Validar login con credenciales válidas	Funcional	Alta	Finalizado	1	1
2	Validar login con contraseña incorrecta	Funcional	Alta	Finalizado	1	1

2 rows in set (0.00 sec)

3. CREATE VIEW vista_caso_pend_pro AS SELECT Caso_Prueba.descripcion AS nombre_caso, Proyecto.nombre AS nombre_proyecto FROM Caso_Prueba JOIN Proyecto ON Caso_Prueba.id_proyecto = Proyecto.id_proyecto WHERE Caso_Prueba.estado = 'Pendiente';



```
mysql> CREATE VIEW vista_caso_pend_pro AS SELECT Caso_Prueba.descripcion  
AS nombre_caso, Proyecto.nombre AS nombre_proyecto FROM Caso_Prueba JOI  
N Proyecto ON Caso_Prueba.id_proyecto = Proyecto.id_proyecto WHERE Caso_  
Prueba.estado = 'Pendiente';  
Query OK, 0 rows affected (0.13 sec)
```

```
mysql> SELECT * FROM vista_caso_pend_pro;  
+-----+-----+  
| nombre_caso | nombre_proyecto |  
+-----+-----+  
| Registrar cliente con datos incompletos | App de Ventas |  
| Visualizar reporte diario | App de Ventas |  
+-----+-----+  
2 rows in set (0.01 sec)
```

4. CREATE VIEW vista_caso_ejecu_falli AS SELECT Caso_Prueba.id_caso, Caso_Prueba.descripcion, Ejecucion.resultado FROM Ejecucion JOIN Caso_Prueba ON Ejecucion.id_caso = Caso_Prueba.id_caso WHERE Ejecucion.resultado = 'Fallido';

```
mysql> CREATE VIEW vista_caso_ejecu_falli AS SELECT Caso_Prueba.id_caso,  
Caso_Prueba.descripcion, Ejecucion.resultado FROM Ejecucion JOIN Caso_P  
rueba ON Ejecucion.id_caso = Caso_Prueba.id_caso WHERE Ejecucion.resulta  
do = 'Falli  
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> SELECT * FROM vista_caso_ejecu_falli;  
+-----+-----+-----+  
| id_caso | descripcion | resultado |  
+-----+-----+-----+  
| 2 | Validar login con contraseña incorrecta | Fallido |  
| 4 | Registrar cliente con datos incompletos | Fallido |  
| 5 | Visualizar reporte diario | Fallido |  
+-----+-----+-----+  
3 rows in set (0.00 sec)
```

5. CREATE VIEW vista_usuario_ejecu AS SELECT Usuario.nombre, Usuario.correo, id_usuario = Ejecucion.id_usuario;

```
mysql> CREATE VIEW vista_usuario_ejecu AS SELECT Usuario.nombre, Usuario  
.corre  
d_usuario = Ejecucion.id_usuario;  
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> SELECT * FROM vista_usuario_ejecu;  
+-----+-----+-----+  
| nombre | correo | id_ejecucion |  
+-----+-----+-----+  
| Carlos | carlos@gmail.com | 1 |  
| Carlos | carlos@gmail.com | 2 |  
| Carlos | carlos@gmail.com | 6 |  
| Marta | marta@gmail.com | 3 |  
| Marta | marta@gmail.com | 4 |  
| Marta | marta@gmail.com | 5 |  
+-----+-----+-----+  
6 rows in set (0.00 sec)
```

```
CREATE VIEW vista_ejecu_usua AS SELECT Usuario.nombre, Usuario.correo Usuario  
JOIN Ejecucion ON Usuario.id_usuario = Ejecucion.id_usuario;
```

```
mysql> CREATE VIEW vista_ejecu_usua AS SELECT Usuario.nombre, Usuario.co  
rre  
Usuario JOIN Ejecucion ON Usuario.id_usuario = Ejecucion.id_usuario;  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> SELECT * FROM vista_ejecu_usua;  
+-----+-----+-----+-----+-----+  
| nombre | correo          | id_ejecucion | fecha      | resultado |  
+-----+-----+-----+-----+-----+  
| Carlos | carlos@gmail.com | 1            | 2023-05-10 | Exitoso   |  
| Carlos | carlos@gmail.com | 2            | 2023-05-12 | Fallido   |  
| Marta  | marta@gmail.com  | 3            | 2023-06-01 | Exitoso   |  
| Marta  | marta@gmail.com  | 4            | 2023-06-02 | Fallido   |  
| Marta  | marta@gmail.com  | 5            | 2023-06-03 | Fallido   |  
| Carlos | carlos@gmail.com | 6            | 2023-06-04 | Exitoso   |  
+-----+-----+-----+-----+-----+  
6 rows in set (0.00 sec)
```

6. CREATE VIEW vista_test_exit AS SELECT DISTINCT u.nombre FROM Usuario u JOIN Ejecucion e ON u.id_usuario = e.id_usuario WHERE u.rol = 'Tester' AND e.resultado = 'Exitoso';

```
mysql> CREATE VIEW vista_test_exit AS SELECT DISTINCT u.nombre FROM Usua  
rio u JOIN Ejecucion e ON u.id_usuario = e.id_usuario WHERE u.rol = 'Tes  
ter' AND e.resultado = 'Exitoso';  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> SELECT * FROM vista_test_exit;  
+-----+  
| nombre |  
+-----+  
| Carlos |  
| Marta  |  
+-----+  
2 rows in set (0.00 sec)
```

7. CREATE VIEW vista_caso_estado AS SELECT cp.descripcion AS `desc` FROM Caso_Prueba cp JOIN (SELECT DISTINCT id_requisito FROM Requisito WHERE estado = 'En progreso')requisitos_en_progreso ON cp.id_requisito = requisitos_en_progreso.id_requisito;

```
mysql> CREATE VIEW vista_caso_estado AS SELECT cp.descripcion AS `desc`  
FROM Caso_Prueba cp JOIN ( SELECT DISTINCT id_requisito FROM Requisito  
WHERE estado = 'En progreso' )requisitos_en_progreso ON cp.id_requisit  
o = requisitos_en_progreso.id_requisito;  
Query OK, 0 rows affected, 1 warning (0.01 sec)  
  
mysql> SELECT * FROM vista_caso_estado;  
+-----+  
| desc |  
+-----+  
| Registrar cliente nuevo |  
| Registrar cliente con datos incompletos |  
+-----+  
2 rows in set (0.00 sec)
```



8. CREATE VIEW nombre_ejecucion AS SELECT DISTINCT Usuario.nombre FROM Usuario JOIN Ejecucion ON Usuario.id_usuario = Ejecucion.id_usuario;

```
MariaDB [SistemaGP]> CREATE VIEW nombre_ejecucion AS SELECT DISTINCT Usuario.nombre FROM Usuario JOIN Ejecucion ON Usuario.id_usuario = Ejecucion.id_usuario;
Query OK, 0 rows affected (0.092 sec)

MariaDB [SistemaGP]> SELECT * FROM nombre_ejecucion;
+-----+
| nombre |
+-----+
| Carlos |
| Marta  |
+-----+
2 rows in set (0.010 sec)
```

9. CREATE VIEW proyecto_responsable AS SELECT Proyecto.nombre AS proyecto_nombre, Usuario.nombre AS usuario_nombre FROM Proyecto JOIN Usuario ON Proyecto.id_usuario = Usuario.id_usuario;

```
MariaDB [SistemaGP]> CREATE VIEW proyecto_responsable AS SELECT Proyecto.nombre AS proyecto_nombre, Usuario.nombre AS usuario_nombre FROM Proyecto JOIN Usuario ON Proyecto.id_usuario = Usuario.id_usuario;
Query OK, 0 rows affected (0.023 sec)

MariaDB [SistemaGP]> SELECT * FROM proyecto_responsable;
+-----+-----+
| proyecto_nombre | usuario_nombre |
+-----+-----+
| Sistema de Gestión Escolar | Ana           |
| App de Ventas      | Luis          |
+-----+-----+
2 rows in set (0.004 sec)
```

10. CREATE VIEW cp_proyecto AS SELECT Caso_Prueba.descripcion AS descripcion_caso, Proyecto.nombre FROM Caso_Prueba JOIN Proyecto ON Caso_Prueba.id_proyecto = Proyecto.id_proyecto;

```
MariaDB [SistemaGP]> CREATE VIEW cp_proyecto AS SELECT Caso_Prueba.descripcion AS descripcion_caso, Proyecto.nombre FROM Caso_Prueba JOIN Proyecto ON Caso_Prueba.id_proyecto = Proyecto.id_proyecto;
Query OK, 0 rows affected (0.028 sec)

MariaDB [SistemaGP]> SELECT * FROM cp_proyecto;
+-----+-----+
| descripcion_caso | nombre |
+-----+-----+
| Validar login con credenciales válidas | Sistema de Gestión Escolar |
| Validar login con contraseña incorrecta | Sistema de Gestión Escolar |
| Registrar cliente nuevo | App de Ventas |
| Registrar cliente con datos incompletos | App de Ventas |
| Visualizar reporte diario | App de Ventas |
| Acceder sin login | Sistema de Gestión Escolar |
+-----+-----+
6 rows in set (0.009 sec)
```



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

Enlace de github: <https://github.com/migdev11/SistemaGP.git>

6. Conclusiones y Recomendaciones

- **Conclusiones:** Aprendizajes adquiridos y aportes a la empresa/institución.
 - Estas prácticas preprofesionales nos permitieron aplicar de manera efectiva los conocimientos adquiridos en clases, tanto en el diseño como en la implementación de bases de datos relacionales.
 - Desarrollar un sistema de gestión de pruebas fortaleció nuestras habilidades técnicas en SQL, modelado de datos y normalización.
 - Realizar consultas de álgebra relacional en la aplicación “Relational” para posteriormente traducirlas a SQL demostró nuestra capacidad de interpretación de datos.
- **Recomendaciones:** Sugerencias para futuras prácticas o mejoras en el área de bases de datos.
 - Implementar escenarios prácticos reales en el entorno laboral para mejorar nuestra experiencia profesional.
 - Solicitar la ayuda del docente a cargo para retroalimentación y verificación de errores poco comunes que contribuyan a nuestro desarrollo.
 - Trabajar con más datos para pulir nuestras habilidades de consultas que permitan al usuario un mejor acceso a la información que necesita.



Universidad
Nacional
de Loja

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

Carrera Ingeniería en Computación

7. Firma y Validación

Estudiante	Docente Supervisor
Richard Vicente Cajas Riofrío 0705739498	Ing. Rene Guamán
Miguel Angel Veintimilla Esparza 1106060542	