



Universidad  
Nacional  
de Loja

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

*Carrera Ingeniería en Computación*

## **UNIVERSIDAD NACIONAL DE LOJA**



**1859**

**Facultad de la energía, las industrias y los recursos naturales no renovables**

**Ingeniería en ciencias de la computación**

### **INFORME DE PRÁCTICAS PREPROFESIONALES BASE DE DATOS**

**Estudiantes:**

Arellys Ajila  
Viviana Córdova  
Fabricio Ruiz

**Docente:**

Ing. René Guaman

**Fecha de entrega:** 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	
Estudiantes: Arellys Ajila, Viviana Cordova, Fabricio Ruiz		
Periodo de prácticas:  Abril-Agosto del 2025		

## 2. Introducción

La gestión de riesgos en proyectos de software es una parte fundamental dentro del ciclo de vida del desarrollo, ya que permite anticiparse a posibles amenazas que puedan afectar el cumplimiento de los objetivos, el alcance, el presupuesto o el tiempo de entrega de un proyecto. Identificar, analizar y mitigar riesgos de manera sistemática incrementa significativamente la probabilidad de éxito en la construcción de sistemas eficientes y alineados con las expectativas de los usuarios. En este contexto, se propone el diseño de un sistema de gestión de riesgos orientado a proyectos de software, dirigido a docentes y estudiantes de la carrera de Ingeniería en Ciencias de la Computación de la Universidad Nacional de Loja. Este sistema tiene como propósito enseñar a identificar, analizar y mitigar riesgos en proyectos desarrollados en materias clave como Requisitos de Software, Gestión de Proyectos y Calidad de Software, facilitando la toma de decisiones proactivas.

Las prácticas preprofesionales realizadas tienen como objetivo principal aplicar los conocimientos adquiridos en la materia de Base de Datos para desarrollar un modelo de datos adecuado, que permita sustentar la funcionalidad de este sistema. A través de estas prácticas, se fortalecieron habilidades técnicas en diseño lógico de bases de datos, modelado entidad-relación, (SGBD) como PostgreSQL o MySQL. Además, se adquirió experiencia práctica en la implementación de relaciones complejas y diseño de consultas SQL orientadas a la coherencia de datos.

Las actividades se llevaron a cabo dentro del área de desarrollo de software de la carrera de Ingeniería en Ciencias de la Computación. El acompañamiento del docente y la constante revisión de avances permitieron un ambiente propicio para el aprendizaje autónomo y guiado. Finalmente, es importante destacar que la base de datos constituye un componente central en el desarrollo de este sistema, ya que su diseño temprano y bien estructurado define la capacidad del sistema para representar correctamente la información relacionada con los



riesgos, proyectos, actores involucrados y las relaciones entre estos elementos. Una base de datos bien diseñada garantiza el correcto almacenamiento de información. Por lo tanto, el desarrollo del modelo de datos fue una etapa clave que marcó la calidad general del sistema, asegurando que este sea escalable, eficiente y fácil de mantener.

### **3. Objetivos**

#### **3.1 Objetivo General**

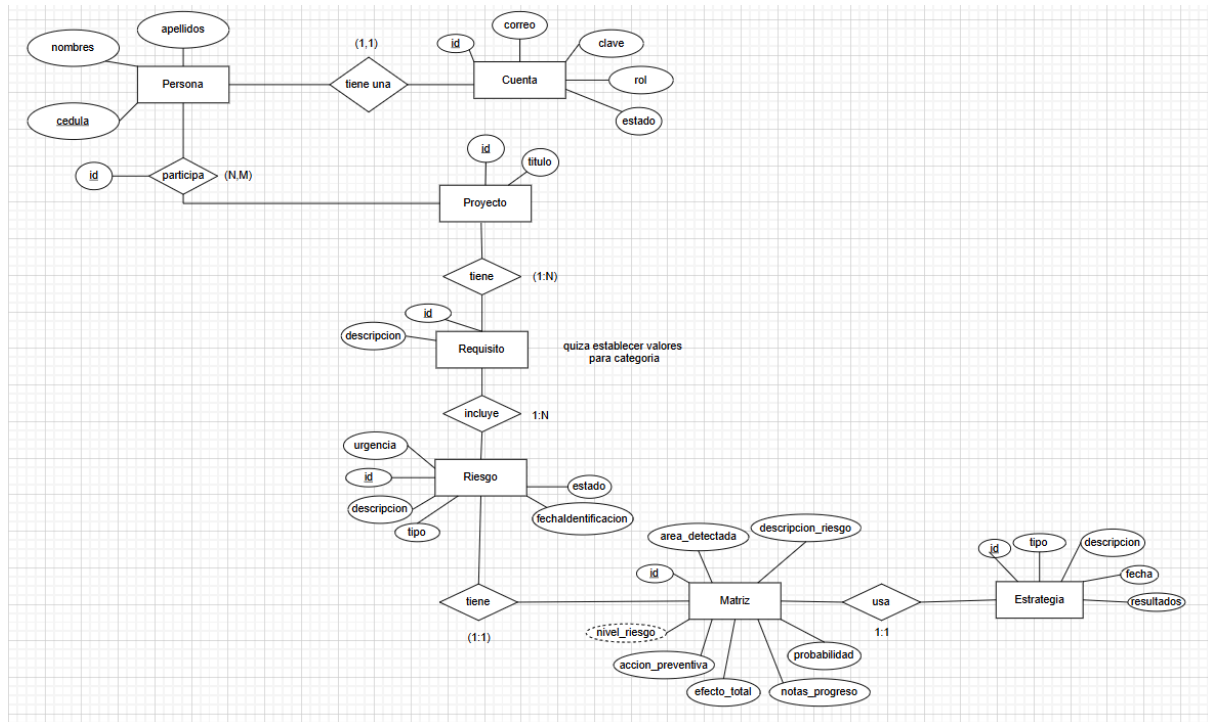
- Desarrollar un modelo de base de datos funcional para un sistema de gestión de riesgos, destinado a estudiantes de la carrera de Ingeniería en Ciencias de la Computación de la Universidad Nacional de Loja, que permita registrar, almacenar y consultar información relacionada con riesgos en proyectos de software.

#### **3.2 Objetivos Específicos**

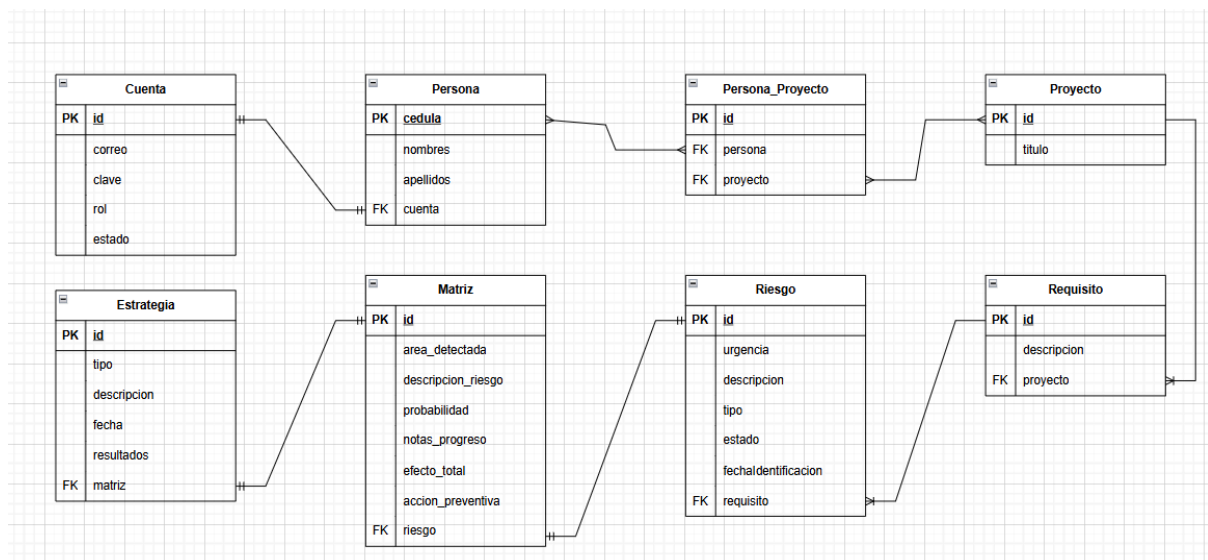
- Diseñar los modelos entidad-relación y relacional que representen verídicamente los datos necesarios para la gestión de riesgos.
- Implementar la base de datos en el sistema de gestión MySQL, incluyendo creación de tablas y claves primarias y foráneas.
- Desarrollar consultas SQL que permitan realizar búsquedas complejas para obtener reportes o estadísticas sobre los riesgos registrados.

### **4. Actividades Realizadas**

- **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 se aplicaron las tres primeras formas normales.

En primer lugar, se cumple la Primera Forma Normal, debido a que cada tabla presenta columnas con valores atómicos y se utilizan correctamente las claves foráneas para evitar la redundancia de datos.

En segundo lugar, se garantiza la Segunda Forma Normal, dado que no existen claves primarias compuestas que puedan generar dependencias parciales entre atributos.

Por último, se cumple la Tercera Forma Normal (3FN), ya que no hay dependencias transitivas entre los atributos no clave.

La correcta aplicación de las reglas de normalización fue fundamental para garantizar un diseño estructurado, funcional y alineado con los principios de integridad y eficiencia en bases de datos.

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

CREATE DATABASE gestion\_riesgos;

```
MariaDB [(none)]> CREATE DATABASE gestion_riesgos;  
Query OK, 1 row affected (0,000 sec)
```

```
CREATE TABLE riesgo (  
    id INT(11) AUTO_INCREMENT PRIMARY KEY,  
    urgencia SET('Alto','Medio','Bajo') NOT NULL,  
    descripcion TEXT,  
    tipo SET('Tecnicos','De Proyecto','Externos','Negocio') NOT NULL,  
    fechaIdentificacion DATE NOT NULL,  
    requisito INT(11),  
    FOREIGN KEY (requisito) REFERENCES requisito(id)  
);
```

```
MariaDB [gestion_riesgos]> CREATE TABLE riesgo(  
-> id INT(11) AUTO_INCREMENT PRIMARY KEY,  
-> urgencia SET('Alto','Medio','Bajo') NOT NULL,  
-> descripcion TEXT,  
-> tipo SET('Tecnicos','De Proyecto','Externos','Negocio') NOT NULL,  
-> fechaIdentificacion DATE NOT NULL,  
-> requisito INT(11),  
-> FOREIGN KEY (requisito) REFERENCES requisito(id)  
-> );  
Query OK, 0 rows affected (0,016 sec)
```

- **Inserciones, modificaciones, actualizaciones, eliminación de BD**
  - **Inserción tabla riesgo**

```
INSERT INTO riesgo (  
id,  
urgencia,  
descripcion,  
tipo,  
estado,  
fechaIdentificacion,  
requisito)  
VALUE (  
23,  
'Alto',  
'Crítico para el funcionamiento',  
'Tecnicos',  
'Controlado',
```



'2025-06-05',  
3 );

```
MariaDB [gestion_riesgos]> INSERT INTO riesgo (
-> id,
-> urgencia,
-> descripcion,
-> tipo,
-> estado,
-> fechaIdentificacion,
-> requisito)
-> VALUE (
-> 23,
-> 'Alto',
-> 'Crítico para el funcionamiento',
-> 'Tecnicos',
-> 'Controlado',
-> '2025-06-05',
-> 3 );
Query OK, 1 row affected (0,004 sec)
```

#### - Modificaciones tabla riesgo

ALTER TABLE riesgo

MODIFY tipo

SET('Tecnicos','De Proyecto','Externos', 'Negocio');

```
MariaDB [gestion_riesgos]> ALTER TABLE riesgo
-> MODIFY tipo
-> SET('Tecnicos','De Proyecto','Externos', 'Negocio');
Query OK, 0 rows affected (0,018 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [gestion_riesgos]> desc riesgo;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
urgencia	set('Alto','Medio','Bajo')	NO		NULL	
descripcion	text	YES		NULL	
tipo	set('Tecnicos','De Proyecto','Externos','Negocio')	YES		NULL	
estado	set('Activo','Controlado','Mitigado','Resuelto')	NO		NULL	
fechaIdentificacion	date	NO		NULL	
requisito	int(11)	YES	MUL	NULL	

7 rows in set (0,001 sec)

#### - Actualizaciones tabla riesgo

UPDATE riesgo

SET estado = 'Activo'

WHERE id = 23

;



```
MariaDB [gestion riesgos]> UPDATE riesgo
-> SET estado = 'Activo'
-> WHERE id = 23
-> ;
Query OK, 1 row affected (0,005 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

#### - Eliminación de columna estado en tabla riesgo

ALTER TABLE riesgo

DROP COLUMN estado;

```
MariaDB [gestion riesgos]> ALTER TABLE riesgo
-> DROP COLUMN estado;
Query OK, 0 rows affected (0,021 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [gestion riesgos]> desc riesgo;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
urgencia	set('Alto','Medio','Bajo')	NO		NULL	
descripcion	text	YES		NULL	
tipo	set('Técnicos','De Proyecto','Externos','Negocio')	YES		NULL	
fechaIdentificacion	date	NO		NULL	
requisito	int(11)	YES	MUL	NULL	

6 rows in set (0,001 sec)

#### ● Consultas y reportes:

##### ○ Consultas en álgebra relacional

##### - Arellys Ajila

#### 1. Listar los requisitos asociados al proyecto UNL\_GASOLINERA

$\pi$  id, descripcion, proyecto ( $\sigma$  id\_proyecto == proyecto ( $\rho$  id  $\rightarrow$  id\_proyecto ( $\sigma$  titulo == "UNL\_GASOLINERA" (proyecto))  $\bowtie$  requisito))

id	descripcion	proyecto
3	El sistema debe generar reportes de ingresos mensuales.	1
1	El sistema debe registrar las ventas diarias de combustible.	1
4	El sistema debe permitir el registro de turnos de empleados.	1
2	El sistema debe permitir la gestión de inventario de combustibles.	1

#### 2. Mostrar el id\_riesgo, descripcion, tipo, estado y fechIdentificacion de los riesgos con estado "Activo" del proyecto "UNL\_FARMACIA".



$\pi$  id\_riesgo, desc, tipo, estado, fechaIdentificacion ( $\sigma$  id == requisito ( $\sigma$  id\_proyecto == proyecto ( $\rho$  id  $\Rightarrow$  id\_proyecto ( $\sigma$  titulo == "UNL\_FARMACIA" (proyecto))  $\bowtie$  requisito)  $\bowtie$   $\rho$  id  $\Rightarrow$  id\_riesgo, descripcion  $\Rightarrow$  desc ( $\sigma$  estado == "Activo" (riesgo))))

id_riesgo	desc	tipo	estado	fechaIdentificacion
5	Medicamentos vencidos no detectados	Tecnicos	Activo	2025-06-05

### 3. Mostrar los riesgos que tengan una matriz con la acción preventiva “Control por rol”.

$\pi$  id\_riesgo, urgencia, descripcion, tipo, estado, fechaIdentificacion, requisito ( $\sigma$  id\_riesgo == riesgo ( $\rho$  id  $\Rightarrow$  id\_riesgo (riesgo)  $\bowtie$   $\sigma$  accion\_preventiva == "Control por rol" (matriz)))

id_riesgo	urgencia	descripcion	tipo	estado	fechaIdentificacion	requisito
18	Medio	Descuentos incorrectos	Tecnicos	Mitigado	2025-06-18	18

### 4. Mostrar los riesgos con urgencia “Alta” de tipo “Tecnicos” y con estado “Activo”.

$\sigma$  estado=="Activo"( $\sigma$  tipo=="Tecnicos"( $\sigma$  urgencia=="Alto"(riesgo)))

id	urgencia	descripcion	tipo	estado	fechaIdentificacion	requisito
5	Alto	Medicamentos vencidos no detectados	Tecnicos	Activo	2025-06-05	5
13	Alto	Pedidos perdidos	Tecnicos	Activo	2025-06-13	13
17	Alto	Lectura errónea de códigos	Tecnicos	Activo	2025-06-17	17
1	Alto	Error en registro de ventas	Tecnicos	Activo	2025-06-01	1

### 5. Mostrar los riesgos que hayan presentado una probabilidad mayor a 60 y un efecto total mayor a 40 en su matriz.

$\pi$  id\_riesgo, urgencia, descripcion, tipo, estado, fechaIdentificacion, requisito, probabilidad, efecto\_total( $\sigma$  efecto\_total > 40 ( $\sigma$  riesgo == id\_riesgo ( $\sigma$  probabilidad > 60 (( $\rho$  id  $\Rightarrow$  id\_riesgo (riesgo))  $\bowtie$  matriz))))

id_riesgo	urgencia	descripcion	tipo	estado	fechaIdentificacion	requisito	probabilidad	efecto_total
17	Alto	Lectura errónea de códigos	Tecnicos	Activo	2025-06-17	17	75	70
5	Alto	Medicamentos vencidos no detectados	Tecnicos	Activo	2025-06-05	5	85	80
1	Alto	Error en registro de ventas	Tecnicos	Activo	2025-06-01	1	80	75
19	Alto	Stock no actualizado	Tecnicos	Controlado	2025-06-19	19	90	85
7	Alto	Errores en facturación	Tecnicos	Controlado	2025-06-07	7	90	85
11	Alto	Manipulación de evaluaciones	Externos	Activo	2025-06-11	11	70	65
9	Medio	Tareas mal asignadas	De Proyecto	Mitigado	2025-06-09	9	65	60
13	Alto	Pedidos perdidos	Tecnicos	Activo	2025-06-13	13	85	80

- Viviana Cordova

### 1. Listar todos aquellos requisitos(id) que sean clasificados como urgencia ‘Bajo’.

$\pi$  requisito( $\sigma$  urgencia == "Bajo"(riesgo))





requisito	urgencia
15	Bajo
10	Bajo
8	Bajo
3	Bajo
20	Bajo

2. Listar todas los nombres de las personas que integran a el proyecto 1, del proyecto solo se quiere su título.

$\pi$  nombres, titulo ( $\sigma$  proyecto == idP( $(\sigma$  persona == cedula( $(\sigma$  proyecto == 1(persona\_proyecto))) $\bowtie$ persona)) $\bowtie$ proyecto))

nombres	titulo
Ana	UNL_GASOLINERA
Veronica	UNL_GASOLINERA
Kevin	UNL_GASOLINERA
Luis	UNL_GASOLINERA

3. Selecciona todos aquellos requisitos asociados al proyecto 5(nombre).

$\pi$  id, titulo ( $\sigma$  proyecto == Id( $(\sigma$  proyecto == 5(requisito)) $\bowtie$ proyecto))

id	titulo
18	UNL_TIENDA
19	UNL_TIENDA
17	UNL_TIENDA
20	UNL_TIENDA

4. Seleccionar todos aquellas riegos registrados a partir del 16 de junio en adelante.

$\pi$  requisito, fechaIdentificacion( $\sigma$  fechaIdentificacion  $\geq$  '2025-06-16'(riesgo))

requisito	fechaIdentificacion
17	2025-06-17
18	2025-06-18
19	2025-06-19
16	2025-06-16
20	2025-06-20

5. Seleccionar todos los registros de cada matriz en la que fue aplicada la estrategia de “Aceptacion” y en qué “area\_detectada” se aplicó.

$\pi$  tipo, area\_detectada( $(\sigma$  tipo == 'Aceptacion'(estrategia)) $\bowtie$ matriz)

tipo	area_detectada	matriz
Aceptacion	Reportes	3
Aceptacion	Reportes	10
Aceptacion	Ventas	20



- Fabricio Ruiz

**1. Obtener todas las cuentas activas pertenecientes a un Jefe de Carrera.**

$\sigma$  estado == 'Activa' and rol == 'Jefe Carrera' (cuenta)

id	correo	clave	rol	estado
3	carlos.mendoza@unl.edu.ec	pass3	Jefe Carrera	Activa
9	jose.villavicencio@unl.edu.ec	pass9	Jefe Carrera	Activa

**2. Obtener la probabilidad, área detectada, acción preventiva de aquellas matrices cuya probabilidad sea mayor o igual a 70.**

$\pi$  probabilidad, area\_detectada, accion\_preventiva( $\sigma$  probabilidad  $\geq 70$  (matriz))

probabilidad	area_detectada	accion_preventiva
90	Facturación	Control doble
90	Stock	Confirmación postventa
80	Ventas	Pruebas automáticas
85	Medicamentos	Verificación de fechas
85	Pedidos	Revisión de endpoints
75	Escáner	Escaneo doble
70	Evaluaciones	Token de seguridad

**3. Obtener la descripción del riesgo, la fecha en que fue identificado y el estado de todos los riesgos que están en estado "Mitigado", combinando la información de la matriz de riesgos con los datos detallados de los riesgos.**

$\pi$  descripcion\_riesgo, fechaIdentificacion, estado( $\sigma$  estado == 'Mitigado' (matriz  $\bowtie$  riesgo))

descripcion_riesgo	fechaIdentificacion	estado
Descuentos no aplicados	2025-06-18	Mitigado
Alertas no llegan	2025-06-06	Mitigado
Desfase en conteo	2025-06-02	Mitigado
Sin ingredientes	2025-06-14	Mitigado
Desorden en archivo	2025-06-12	Mitigado
Duplicación	2025-06-09	Mitigado

**4. Obtener la descripción y el tipo de riesgos, junto con la descripción detallada del riesgo desde la matriz, para todos los riesgos cuyo tipo sea "Tecnicos".**

$\pi$  descripcion, tipo, descripcion\_riesgo ( $\sigma$  tipo == "Tecnicos" (riesgo  $\bowtie$  matriz))



descripcion	tipo	descripcion_riesgo
Pedidos perdidos	Tecnicos	Pedidos perdidos
Error en registro de ventas	Tecnicos	Error de registro
Medicamentos vencidos no detectados	Tecnicos	Vencimientos no detectados
Faltante de ingredientes	Tecnicos	Sin ingredientes
Descuentos incorrectos	Tecnicos	Descuentos no aplicados
Stock no actualizado	Tecnicos	No actualización
Lectura errónea de códigos	Tecnicos	Falla en lectura
Documentos mal organizados	Tecnicos	Desorden en archivo
Errores en impuestos	Tecnicos	Impuestos mal calculados
Errores en facturación	Tecnicos	Errores en precios
Alertas no funcionales	Tecnicos	Alertas no llegan
Desfase en stock de combustibles	Tecnicos	Desfase en conteo

5. Listar la descripción de los riesgos y su efecto total para aquellos riesgos que están en estado "Controlado", combinando la información de las tablas riesgo y matriz.

$\pi$  descripcion, efecto\_total ( $\sigma$  estado == "Controlado" (riesgo  $\bowtie$  matriz ))

descripcion	efecto_total
Errores en facturación	85
Fallas en generación de reportes	35
Stock no actualizado	85
Reservas duplicadas	50
Datos incompletos en reportes	30

- Consultas en SQL
  - Arelys Ajila

1. Listar los requisitos asociados al proyecto UNL\_GASOLINERA.

```
SELECT r.id, r.descripcion, r.proyecto
FROM requisito r
JOIN (
    SELECT idP AS id_proyecto
    FROM proyecto
    WHERE titulo = 'UNL_GASOLINERA'
) p ON r.proyecto = p.id_proyecto;
```



id	descripcion	proyecto
1	El sistema debe registrar las ventas diarias de combustible.	1
2	El sistema debe permitir la gestión de inventario de combustibles.	1
3	El sistema debe generar reportes de ingresos mensuales.	1
4	El sistema debe permitir el registro de turnos de empleados.	1

**2. Mostrar el id\_riesgo, descripcion, tipo, estado y fechaIdentificacion de los riesgos con estado “Activo” del proyecto “UNL\_FARMACIA”.**

```
SELECT
    r.id AS id_riesgo, r.descripcion AS descripcion_riesgo, r.tipo, r.estado,
    r.fechaIdentificacion
FROM riesgo r
JOIN requisito req ON r.requisito = req.id
JOIN proyecto p ON req.proyecto = p.idP
WHERE r.estado = 'Activo'
AND p.titulo = 'UNL_FARMACIA';
```

id_riesgo	descripcion_riesgo	tipo	estado	fechaIdentificacion
5	Medicamentos vencidos no detectados	Tecnicos	Activo	2025-06-05

**3. Mostrar los riesgos que tengan una matriz con la acción preventiva “Control por rol”.**

```
SELECT
    r.id AS id_riesgo,
    r.urgencia,
    r.descripcion,
    r.tipo,
    r.estado,
    r.fechaIdentificacion,
    r.requisito
FROM
    riesgo r
JOIN
    matriz m ON r.id = m.id_riesgo
WHERE
    m.accion_preventiva = 'Control por rol';
```



id_riesgo	urgencia	descripcion	tipo	estado	fechaIdentificacion	requisito
18	Medio	Descuentos incorrectos	Tecnicos	Mitigado	2025-06-18	18

#### 4. Mostrar los riesgos con urgencia “Alta” de tipo “Tecnicos” y con estado “Activo”.

```
SELECT *
FROM riesgo
WHERE urgencia = 'Alto'
AND tipo = 'Tecnicos'
AND estado = 'Activo';
```

id	urgencia	descripcion	tipo	estado	fechaIdentificacion	requisito
1	Alto	Error en registro de ventas	Tecnicos	Activo	2025-06-01	1
5	Alto	Medicamentos vencidos no detectados	Tecnicos	Activo	2025-06-05	5
13	Alto	Pedidos perdidos	Tecnicos	Activo	2025-06-13	13
17	Alto	Lectura errónea de códigos	Tecnicos	Activo	2025-06-17	17

#### 5. Mostrar los riesgos que hayan presentado una probabilidad mayor a 60 y un efecto total mayor a 40 en su matriz.

```
SELECT
    r.id AS id_riesgo, r.urgencia, r.descripcion, r.tipo, r.estado, r.fechaIdentificacion,
    r.requisito, m.probabilidad, m.efecto_total
FROM riesgo r
JOIN matriz m ON m.riesgo = r.id
WHERE m.probabilidad > 60
AND m.efecto_total > 40;
```

id_riesgo	urgencia	descripcion	tipo	estado	fechaIdentificacion	requisito	probabilidad	efecto_total
1	Alto	Error en registro de ventas	Tecnicos	Activo	2025-06-01	1	80	75
5	Alto	Medicamentos vencidos no detectados	Tecnicos	Activo	2025-06-05	5	85	80
7	Alto	Errores en facturación	Tecnicos	Controlado	2025-06-07	7	90	85
9	Medio	Tareas mal asignadas	De Proyecto	Mitigado	2025-06-09	9	65	60
11	Alto	Manipulación de evaluaciones	Externos	Activo	2025-06-11	11	70	65
13	Alto	Pedidos perdidos	Tecnicos	Activo	2025-06-13	13	85	80
17	Alto	Lectura errónea de códigos	Tecnicos	Activo	2025-06-17	17	75	70
19	Alto	Stock no actualizado	Tecnicos	Controlado	2025-06-19	19	90	85

- Viviana Cordova

#### 1. Listar todos aquellos requisitos(id) que sean clasificados como urgencia ‘Bajo’.

```
SELECT requisito, urgencia
FROM riesgo WHERE urgencia = "Bajo";
```



```
MariaDB [gestion_riesgos]> SELECT requisito, urgencia FROM riesgo WHERE urgencia = 'Bajo';
```

requisito	urgencia
3	Bajo
8	Bajo
10	Bajo
15	Bajo
20	Bajo

```
5 rows in set (0,000 sec)
```

2. Listar todas los nombres de las personas que integran a el proyecto 1, del proyecto solo se quiere su título.

```
SELECT persona.nombres, proyecto.titulo
```

```
FROM persona_proyecto
```

```
JOIN persona ON persona_proyecto.persona = persona.cedula
```

```
JOIN proyecto ON persona_proyecto.proyecto = proyecto.idP
```

```
WHERE persona_proyecto.proyecto = 1;
```

```
MariaDB [gestion_riesgos]> SELECT nombres, titulo FROM persona_proyecto JOIN persona ON persona_proyecto.persona = p  
persona.cedula JOIN proyecto ON persona_proyecto.proyecto = proyecto.idP WHERE persona = cedula AND proyecto= 1;
```

nombres	titulo
Ana	UNL_GASOLINERA
Luis	UNL_GASOLINERA
Kevin	UNL_GASOLINERA
Veronica	UNL_GASOLINERA

```
4 rows in set (0,000 sec)
```

3. Selecciona todos aquellos requisitos asociados al proyecto 5(nombre).

```
SELECT id, titulo
```

```
FROM requisito JOIN proyecto ON requisito.proyecto = proyecto.idP
```

```
WHERE proyecto = 5 AND proyecto = idP;
```

```
MariaDB [gestion_riesgos]> SELECT id, titulo FROM requisito JOIN proyecto ON requisito.proyecto  
= proyecto.idP WHERE proyecto =  
5 AND proyecto = idP;
```

id	titulo
17	UNL_TIENDA
18	UNL_TIENDA
19	UNL_TIENDA
20	UNL_TIENDA

```
4 rows in set (0,000 sec)
```

4. Seleccionar todos aquellas riegos registrados a partir del 16 de junio en adelante  
SELECT requisito, fechaIdentificacion.

```
FROM riesgo
```

```
WHERE fechaIdentificacion >= "2025-06-16";
```



```
MariaDB [gestion_riesgos]> SELECT requisito, fechaIdentificacion FROM riesgo WHERE fechaIdentificacion >= '2025-06-16';
```

requisito	fechaIdentificacion
16	2025-06-16
17	2025-06-17
18	2025-06-18
19	2025-06-19
20	2025-06-20

```
5 rows in set (0,000 sec)
```

**5. Seleccionar todos los registros de cada matriz en la que fue aplicada la estrategia de “Aceptacion” y en qué “area\_detectada” se aplicó.**

```
SELECT tipo, area_detectada, matriz  
FROM estrategia JOIN matriz ON estrategia.matriz = matriz.id  
WHERE tipo = 'Aceptacion' ;
```

```
MariaDB [gestion_riesgos]> SELECT tipo, area_detectada, matriz FROM estrategia JOIN matriz ON  
estrategia.matriz = matriz.id WHERE tipo = 'Aceptacion';
```

tipo	area_detectada	matriz
Aceptacion	Reportes	3
Aceptacion	Reportes	10
Aceptacion	Ventas	20

```
3 rows in set (0,000 sec)
```

- Fabricio Ruiz

**1. Obtener todas las cuentas activas pertenecientes a un Jefe de Carrera.**

```
SELECT * FROM cuenta WHERE estado = 'Activa' AND rol = 'Jefe Carrera';
```

```
mysql> SELECT FROM cuenta WHERE estado = 'Activa' AND rol = 'Jefe Carrera';
```

id	correo	clave	rol	estado
3	carlos.mendoza@unl.edu.ec	pass3	Jefe Carrera	Activa
9	jose.villavicencio@unl.edu.ec	pass9	Jefe Carrera	Activa

```
2 rows in set (0,00 sec)
```

**2. Obtener la probabilidad, área detectada, acción preventiva de aquellas matrices cuya probabilidad sea mayor o igual a 70.**

```
SELECT probabilidad, area_detectada, accion_preventiva FROM matriz WHERE  
probabilidad >= 70;
```



```
mysql> SELECT probabilidad, area_detectada, accion_preventiva FROM matriz WHERE probabilidad >= 70;
```

probabilidad	area_detectada	accion_preventiva
80	Ventas	Pruebas automáticas
85	Medicamentos	Verificación de fechas
90	Facturación	Control doble
70	Evaluaciones	Token de seguridad
85	Pedidos	Revisión de endpoints
75	Escáner	Escaneo doble
90	Stock	Confirmación postventa

```
7 rows in set (0,00 sec)
```

3. Obtener la descripción del riesgo, la fecha en que fue identificado y el estado de todos los riesgos que están en estado "Mitigado", combinando la información de la matriz de riesgos con los datos detallados de los riesgos.

SELECT m.descripcion\_riesgo, r.fechaIdentificacion, r.estado FROM matriz m JOIN riesgo r ON m.riesgo = r.id WHERE r.estado = 'Mitigado';

```
mysql> SELECT m.descripcion_riesgo, r.fechaIdentificacion, r.estado FROM matriz m JOIN riesgo r ON m.riesgo = r.id WHERE r.estado = 'Mitigado';
```

descripcion_riesgo	fechaIdentificacion	estado
Desfase en conteo	2025-06-02	Mitigado
Alertas no llegan	2025-06-06	Mitigado
Duplicación	2025-06-09	Mitigado
Desorden en archivo	2025-06-12	Mitigado
Sin ingredientes	2025-06-14	Mitigado
Descuentos no aplicados	2025-06-18	Mitigado

```
6 rows in set (0,00 sec)
```

4. Obtener la descripción y el tipo de riesgos, junto con la descripción detallada del riesgo desde la matriz, para todos los riesgos cuyo tipo sea "Técnicos".

SELECT r.descripcion, r.tipo, m.descripcion\_riesgo FROM riesgo r JOIN matriz m ON r.id = m.riesgo WHERE r.tipo = 'Técnicos';

```
mysql> SELECT r.descripcion, r.tipo, m.descripcion_riesgo FROM riesgo r JOIN matriz m ON r.id = m.riesgo WHERE r.tipo = 'Técnicos';
```

descripcion	tipo	descripcion_riesgo
Error en registro de ventas	Técnicos	Error de registro
Desfase en stock de combustibles	Técnicos	Desfase en conteo
Medicamentos vencidos no detectados	Técnicos	Vencimientos no detectados
Alertas no funcionales	Técnicos	Alertas no llegan
Errores en facturación	Técnicos	Errores en precios
Documentos mal organizados	Técnicos	Desorden en archivo
Pedidos perdidos	Técnicos	Pedidos perdidos
Faltante de ingredientes	Técnicos	Sin ingredientes
Errores en impuestos	Técnicos	Impuestos mal calculados
Lectura errónea de códigos	Técnicos	Falla en lectura
Descuentos incorrectos	Técnicos	Descuentos no aplicados
Stock no actualizado	Técnicos	No actualización

```
12 rows in set (0,00 sec)
```

5. Listar la descripción de los riesgos y su efecto total para aquellos riesgos que están en estado "Controlado", combinando la información de las tablas riesgo y matriz.

SELECT r.descripcion, m.efecto\_total FROM riesgo r JOIN matriz m ON r.id = m.riesgo WHERE r.estado = 'Controlado';





```
mysql> SELECT r.descripcion, m.efecto_total FROM riesgo r JOIN matriz m ON r.id = m.riesgo WHERE r.estado = 'Controlado';
```

descripcion	efecto_total
Fallas en generación de reportes	35
Errores en facturación	85
Datos incompletos en reportes	30
Reservas duplicadas	50
Stock no actualizado	85

```
5 rows in set (0,00 sec)
```

## Views

- Arelys Ajila

### 1. Listar los requisitos asociados al proyecto UNL\_GASOLINERA.

```
CREATE VIEW view_requisitosAsociadosUNL
AS SELECT r.id, r.descripcion, r.proyecto
FROM requisito r
JOIN ( SELECT idP AS id_proyecto
FROM proyecto
WHERE titulo = 'UNL_GASOLINERA' ) p
ON r.proyecto = p.id_proyecto;
```

```
MariaDB [gestion_riesgos]> desc view_requisitosAsociadosUNL ;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO		0	
descripcion	text	YES		NULL	
proyecto	int(11)	YES		NULL	

```
3 rows in set (0,001 sec)
```

```
MariaDB [gestion_riesgos]> select * from view_requisitosAsociadosUNL;
```

id	descripcion	proyecto
1	El sistema debe registrar las ventas diarias de combustible.	1
2	El sistema debe permitir la gestión de inventario de combustibles.	1
3	El sistema debe generar reportes de ingresos mensuales.	1
4	El sistema debe permitir el registro de turnos de empleados.	1

```
4 rows in set (0,001 sec)
```

### 2. Mostrar el id\_riesgo, descripcion, tipo, estado y fechaIdentificacion de los riesgos con estado "Activo" del proyecto "UNL\_FARMACIA".

```
CREATE VIEW view_riesgoActivoUNL AS SELECT
r.id AS id_riesgo, r.descripcion AS descripcion_riesgo, r.tipo, r.estado,
r.fechaIdentificacion
FROM riesgo r
JOIN requisito req ON r.requisito = req.id
JOIN proyecto p ON req.proyecto = p.idP
WHERE r.estado = 'Activo'
AND p.titulo = 'UNL_FARMACIA';
```



```
MariaDB [gestion_riesgos]> select * from view_riesgoActivoUNL ;
```

id_riesgo	descripcion_riesgo	tipo	estado	fechaIdentificacion
5	Medicamentos vencidos no detectados	Tecnicos	Activo	2025-06-05

1 row in set (0,001 sec)

```
MariaDB [gestion_riesgos]> desc view_riesgoActivoUNL ;
```

Field	Type	Null	Key	Default	Extra
id_riesgo	int(11)	NO		0	
descripcion_riesgo	text	YES		NULL	
tipo	set('Tecnicos','De Proyecto','Externos')	NO		NULL	
estado	set('Activo','Controlado','Mitigado','Resuelto')	NO		NULL	
fechaIdentificacion	date	NO		NULL	

5 rows in set (0,001 sec)

### 3. Mostrar los riesgos que tengan una matriz con la acción preventiva “Control por rol”.

$\pi$  id\_riesgo, urgencia, descripcion, tipo, estado, fechaIdentificacion, requisito ( $\sigma$  id\_riesgo == riesgo ( $\rho$  id  $\rightarrow$  id\_riesgo (riesgo)  $\bowtie$   $\sigma$  accion\_preventiva == "Control por rol" (matriz)))

```
CREATE VIEW view_riesgoMatriz AS
```

```
SELECT
  r.id AS id_riesgo,
  r.urgencia,
  r.descripcion,
  r.tipo,
  r.estado,
  r.fechaIdentificacion,
  r.requisito
FROM
  riesgo r
JOIN
  matriz m ON r.id = m.riesgo
WHERE
  m.accion_preventiva = 'Control por rol';
```

```
MariaDB [gestion_riesgos]> desc view_riesgoMatriz ;
```

Field	Type	Null	Key	Default	Extra
id_riesgo	int(11)	NO		0	
urgencia	set('Alto','Medio','Bajo')	NO		NULL	
descripcion	text	YES		NULL	
tipo	set('Tecnicos','De Proyecto','Externos')	NO		NULL	
estado	set('Activo','Controlado','Mitigado','Resuelto')	NO		NULL	
fechaIdentificacion	date	NO		NULL	
requisito	int(11)	YES		NULL	

7 rows in set (0,001 sec)



```
MariaDB [gestion_riesgos]> desc view_riesgoMatriz ;
```

Field	Type	Null	Key	Default	Extra
id_riesgo	int(11)	NO		0	
urgencia	set('Alto','Medio','Bajo')	NO		NULL	
descripcion	text	YES		NULL	
tipo	set('Tecnicos','De Proyecto','Externos')	NO		NULL	
estado	set('Activo','Controlado','Mitigado','Resuelto')	NO		NULL	
fechaIdentificacion	date	NO		NULL	
requisito	int(11)	YES		NULL	

7 rows in set (0,001 sec)

#### 4. Mostrar los riesgos con urgencia “Alta” de tipo “Tecnicos” y con estado “Activo”.

```
CREATE VIEW view_riesgoAlto AS SELECT *  
FROM riesgo  
WHERE urgencia = 'Alto'  
AND tipo = 'Tecnicos'  
AND estado = 'Activo';
```

```
MariaDB [gestion_riesgos]> desc view_riesgoAlto ;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO		0	
urgencia	set('Alto','Medio','Bajo')	NO		NULL	
descripcion	text	YES		NULL	
tipo	set('Tecnicos','De Proyecto','Externos')	NO		NULL	
estado	set('Activo','Controlado','Mitigado','Resuelto')	NO		NULL	
fechaIdentificacion	date	NO		NULL	
requisito	int(11)	YES		NULL	

7 rows in set (0,001 sec)

```
MariaDB [gestion_riesgos]> select * from view_riesgoAlto ;
```

id	urgencia	descripcion	tipo	estado	fechaIdentificacion	requisito
1	Alto	Error en registro de ventas	Tecnicos	Activo	2025-06-01	1
5	Alto	Medicamentos vencidos no detectados	Tecnicos	Activo	2025-06-05	5
13	Alto	Pedidos perdidos	Tecnicos	Activo	2025-06-13	13
17	Alto	Lectura errónea de códigos	Tecnicos	Activo	2025-06-17	17

4 rows in set (0,000 sec)

#### 5. Mostrar los riesgos que hayan presentado una probabilidad mayor a 60 y un efecto total mayor a 40 en su matriz.

```
CREATE VIEW view_riesgoProbabilidad AS SELECT  
r.id AS id_riesgo, r.urgencia, r.descripcion, r.tipo, r.estado, r.fechaIdentificacion,  
r.requisito, m.probabilidad, m.efecto_total  
FROM riesgo r  
JOIN matriz m ON m.riesgo = r.id  
WHERE m.probabilidad > 60
```



AND m.efecto\_total > 40;

```
MariaDB [gestion_riesgos]> desc view_riesgoProbabilidad ;
```

Field	Type	Null	Key	Default	Extra
id_riesgo	int(11)	NO		0	
urgencia	set('Alto','Medio','Bajo')	NO		NULL	
descripcion	text	YES		NULL	
tipo	set('Técnicos','De Proyecto','Externos')	NO		NULL	
estado	set('Activo','Controlado','Mitigado','Resuelto')	NO		NULL	
fechaIdentificacion	date	NO		NULL	
requisito	int(11)	YES		NULL	
probabilidad	int(11)	NO		NULL	
efecto_total	int(11)	NO		NULL	

9 rows in set (0,001 sec)

```
MariaDB [gestion_riesgos]> select * from view_riesgoProbabilidad ;
```

id_riesgo	urgencia	descripcion	tipo	estado	fechaIdentificacion	requisito	probabilidad	efecto_total
1	Alto	Error en registro de ventas	Técnicos	Activo	2025-06-01	1	80	75
5	Alto	Medicamentos vencidos no detectados	Técnicos	Activo	2025-06-05	5	85	80
7	Alto	Errores en facturación	Técnicos	Controlado	2025-06-07	7	90	85
9	Medio	Tareas mal asignadas	De Proyecto	Mitigado	2025-06-09	9	65	60
11	Alto	Manipulación de evaluaciones	Externos	Activo	2025-06-11	11	70	65
13	Alto	Pedidos perdidos	Técnicos	Activo	2025-06-13	13	85	80
17	Alto	Lectura errónea de códigos	Técnicos	Activo	2025-06-17	17	75	70
19	Alto	Stock no actualizado	Técnicos	Controlado	2025-06-19	19	90	85

8 rows in set (0,001 sec)

- Viviana Cordova

1. Listar todos aquellos requisitos(id) que sean clasificados como urgencia 'Bajo'.

```
CREATE VIEW view_riesgoUrgente AS
```

```
SELECT requisito, urgencia
```

```
FROM riesgo WHERE urgencia = "Bajo";
```

```
MariaDB [gestion_riesgos]> desc view_riesgoUrgente ;
```

Field	Type	Null	Key	Default	Extra
requisito	int(11)	YES		NULL	
urgencia	set('Alto','Medio','Bajo')	NO		NULL	

2 rows in set (0,001 sec)

```
MariaDB [gestion_riesgos]> select * from view_riesgoUrgente;
```

requisito	urgencia
3	Bajo
8	Bajo
10	Bajo
15	Bajo
20	Bajo

5 rows in set (0,000 sec)



- 2. Listar todas los nombres de las personas que integran a el proyecto 1, del proyecto solo se quiere su título.**

```
CREATE VIEW view_integrantesProyecto
AS SELECT persona.nombres, proyecto.titulo
FROM persona_proyecto
JOIN persona ON persona_proyecto.persona = persona.cedula
JOIN proyecto ON persona_proyecto.proyecto = proyecto.idP
WHERE persona_proyecto.proyecto = 1 ;
```

```
MariaDB [gestion_riesgos]> desc view_integrantesProyecto ;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nombres | varchar(25)   | NO   |     | NULL    |       |
| titulo  | varchar(100)  | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0,001 sec)
```

```
MariaDB [gestion_riesgos]> select * from view_integrantesProyecto ;
+-----+-----+
| nombres | titulo          |
+-----+-----+
| Ana     | UNL_GASOLINERA |
| Luis    | UNL_GASOLINERA |
| Kevin   | UNL_GASOLINERA |
| Veronica| UNL_GASOLINERA |
+-----+-----+
4 rows in set (0,001 sec)
```

- 3. Selecciona todos aquellos requisitos asociados al proyecto 5(nombre).**

```
SELECT id, titulo
FROM requisito JOIN proyecto ON requisito.proyecto = proyecto.idP
WHERE proyecto = 5 AND proyecto = idP;
```

```
MariaDB [gestion_riesgos]> desc view_requisitosProyecto5 ;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)       | NO   |     | 0        |       |
| titulo | varchar(100)  | NO   |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0,001 sec)
```



```
MariaDB [gestion_riesgos]> select * from view_requisitosProyecto5 ;
```

id	titulo
17	UNL_TIENDA
18	UNL_TIENDA
19	UNL_TIENDA
20	UNL_TIENDA

```
4 rows in set (0,000 sec)
```

**4. Seleccionar todos aquellas riegos registrados a partir del 16 de junio en adelante.**

CREATE VIEW view\_requisitoProyecto

AS SELECT id, titulo

FROM requisito JOIN proyecto ON requisito.proyecto = proyecto.idP

WHERE proyecto = 5 AND proyecto = idP;

```
MariaDB [gestion_riesgos]> desc view_requisitoProyecto ;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO		0	
titulo	varchar(100)	NO		NULL	

```
2 rows in set (0,001 sec)
```

```
MariaDB [gestion_riesgos]> select *from view_requisitoProyecto;
```

id	titulo
17	UNL_TIENDA
18	UNL_TIENDA
19	UNL_TIENDA
20	UNL_TIENDA

```
4 rows in set (0,000 sec)
```

**5. Seleccionar todos los registros de cada matriz en la que fue aplicada la estrategia de “Aceptacion” y en qué “area\_detectada” se aplicó.**

CREATE VIEW view\_matrizRegistro

AS SELECT tipo, area\_detectada, matriz

FROM estrategia JOIN matriz ON estrategia.matriz = matriz.id

WHERE tipo = 'Aceptacion' ;

```
MariaDB [gestion_riesgos]> desc view_matrizRegistro ;
```

Field	Type	Null	Key	Default	Extra
tipo	set('Mitigacion','Transferencia','Aceptacion','Evitacion')	NO		NULL	
area_detectada	text	NO		NULL	
matriz	int(11)	YES		NULL	

```
3 rows in set (0,001 sec)
```



```
MariaDB [gestion_riesgos]> select * from view_matrizRegistro ;
```

tipo	area_detectada	matriz
Aceptacion	Reportes	3
Aceptacion	Reportes	10
Aceptacion	Ventas	20

```
3 rows in set (0,000 sec)
```

## - Fabricio Ruiz

### 1. Obtener todas las cuentas activas pertenecientes a un Jefe de Carrera.

```
CREATE VIEW view_cuentas_jefescar AS SELECT * FROM cuenta WHERE estado =  
'Activa' AND rol = 'Jefe Carrera';
```

```
mysql> CREATE VIEW view_cuentas_jefescar AS SELECT * FROM cuenta WHERE estado =  
-> 'Activa' AND rol = 'Jefe Carrera';  
Query OK, 0 rows affected (0,01 sec)
```

```
SELECT * FROM view_cuentas_jefescar;
```

```
mysql> SELECT * FROM view_cuentas_jefescar;
```

id	correo	clave	rol	estado
3	carlos.mendoza@unl.edu.ec	pass3	Jefe Carrera	Activa
9	jose.villavicencio@unl.edu.ec	pass9	Jefe Carrera	Activa

```
2 rows in set (0,00 sec)
```

### 2. Obtener la probabilidad, área detectada, acción preventiva de aquellas matrices cuya probabilidad sea mayor o igual a 70.

```
CREATE VIEW view_matrices_alta_probabilidad AS SELECT probabilidad,  
area_detectada, accion_preventiva FROM matriz WHERE probabilidad >= 70;
```

```
mysql> CREATE VIEW view_matrices_alta_probabilidad AS SELECT probabilidad, area_  
detectada, accion_preventiva FROM matriz WHERE probabilidad >= 70;  
Query OK, 0 rows affected (0,00 sec)
```

```
SELECT * FROM view_matrices_alta_probabilidad;
```



```
mysql> SELECT * FROM view_matrices_alta_probabilidad;
```

probabilidad	area_detectada	accion_preventiva
80	Ventas	Pruebas automáticas
85	Medicamentos	Verificación de fechas
90	Facturación	Control doble
70	Evaluaciones	Token de seguridad
85	Pedidos	Revisión de endpoints
75	Escáner	Escaneo doble
90	Stock	Confirmación postventa

```
7 rows in set (0,00 sec)
```

3. Obtener la descripción del riesgo, la fecha en que fue identificado y el estado de todos los riesgos que están en estado "Mitigado", combinando la información de la matriz de riesgos con los datos detallados de los riesgos.

```
CREATE VIEW view_riesgos_mitigados AS SELECT m.descripcion_riesgo, r.fechaIdentificacion, r.estado FROM matriz m JOIN riesgo r ON m.riesgo = r.id WHERE r.estado = 'Mitigado';
```

```
mysql> CREATE VIEW view_riesgos_mitigados AS SELECT m.descripcion_riesgo, r.fechaIdentificacion, r.estado FROM matriz m JOIN riesgo r ON m.riesgo = r.id WHERE r.estado = 'Mitigado';
Query OK, 0 rows affected (0,01 sec)
```

```
SELECT * FROM view_riesgos_mitigados;
```

```
mysql> SELECT * FROM view_riesgos_mitigados;
```

descripcion_riesgo	fechaIdentificacion	estado
Desfase en conteo	2025-06-02	Mitigado
Alertas no llegan	2025-06-06	Mitigado
Duplicación	2025-06-09	Mitigado
Desorden en archivo	2025-06-12	Mitigado
Sin ingredientes	2025-06-14	Mitigado
Descuentos no aplicados	2025-06-18	Mitigado

```
6 rows in set (0,00 sec)
```

4. Obtener la descripción y el tipo de riesgos, junto con la descripción detallada del riesgo desde la matriz, para todos los riesgos cuyo tipo sea "Técnicos".

```
CREATE VIEW view_riesgos_tecnicos AS SELECT r.descripcion, r.tipo, m.descripcion_riesgo FROM riesgo r JOIN matriz m ON r.id = m.riesgo WHERE r.tipo = 'Técnicos';
```





```
mysql> CREATE VIEW view_riesgos_tecnicos AS SELECT r.descripcion, r.tipo, m.descripcion_riesgo FROM riesgo r JOIN matriz m ON r.id = m.riesgo WHERE r.tipo = 'Tecnicos';
Query OK, 0 rows affected (0,01 sec)
```

SELECT \* FROM view\_riesgos\_tecnicos;

```
mysql> SELECT * FROM view_riesgos_tecnicos;
+-----+-----+-----+
| descripcion | tipo | descripcion_riesgo |
+-----+-----+-----+
| Error en registro de ventas | Tecnicos | Error de registro |
| Desfase en stock de combustibles | Tecnicos | Desfase en conteo |
| Medicamentos vencidos no detectados | Tecnicos | Vencimientos no detectados |
| Alertas no funcionales | Tecnicos | Alertas no llegan |
| Errores en facturación | Tecnicos | Errores en precios |
| Documentos mal organizados | Tecnicos | Desorden en archivo |
| Pedidos perdidos | Tecnicos | Pedidos perdidos |
| Faltante de ingredientes | Tecnicos | Sin ingredientes |
| Errores en impuestos | Tecnicos | Impuestos mal calculados |
| Lectura errónea de códigos | Tecnicos | Falla en lectura |
| Descuentos incorrectos | Tecnicos | Descuentos no aplicados |
| Stock no actualizado | Tecnicos | No actualización |
+-----+-----+-----+
12 rows in set (0,00 sec)
```

5. Listar la descripción de los riesgos y su efecto total para aquellos riesgos que están en estado "Controlado", combinando la información de las tablas riesgo y matriz.

CREATE VIEW view\_riesgos\_controlados AS SELECT r.descripcion, m.efecto\_total FROM riesgo r JOIN matriz m ON r.id = m.riesgo WHERE r.estado = 'Controlado';

```
mysql> CREATE VIEW view_riesgos_controlados AS SELECT r.descripcion, m.efecto_total FROM riesgo r JOIN matriz m ON r.id = m.riesgo WHERE r.estado = 'Controlado';
Query OK, 0 rows affected (0,00 sec)
```

SELECT \* FROM view\_riesgos\_controlados;

```
mysql> SELECT * FROM view_riesgos_controlados;
+-----+-----+
| descripcion | efecto_total |
+-----+-----+
| Fallas en generación de reportes | 35 |
| Errores en facturación | 85 |
| Datos incompletos en reportes | 30 |
| Reservas duplicadas | 50 |
| Stock no actualizado | 85 |
+-----+-----+
5 rows in set (0,00 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**

**Link GitHub:** <https://github.com/vivinaCordova/PP-Base-de-Datos.git>

## 6. Conclusiones

Durante el desarrollo de las prácticas preprofesionales, se logró la aplicación de los conocimientos adquiridos en la asignatura de Bases de Datos en conjunto con la materia de requisitos de software. El proyecto tuvo como objetivo el diseño e implementación de un sistema de base de datos orientado a la gestión de riesgos en proyectos de software, permitiendo el desarrollo funcional, estructurado y adecuada en un contexto académico, el cual se destino a estudiantes del tercer ciclo de la carrera de Ingeniería en Ciencias de la Computación de la Universidad Nacional de Loja.

A lo largo del proceso de desarrollo, se logró el fortalecimiento de habilidades de diseño, abstracción y lógica de bases de datos, partiendo por la elaboración de modelos Entidad-Relación y su correspondiente transformación al modelo relacional, logrando así una representación clara del flujo y comportamiento del sistema. Asimismo, se adquirió experiencia técnica en la implementación de bases de datos relacionales mediante el uso del sistema gestor MySQL, lo cual incluyó la creación de esquemas, definición de tablas, dominios, claves primarias y foráneas, la inserción de datos, modificaciones, actualizaciones y eliminaciones, como la aplicación de buenas prácticas de normalización.

Además, se desarrollaron destrezas en la lógica, abstracción y optimización de consultas complejas, empleando en primera instancia el álgebra relacional como una base teórica y posteriormente SQL, en la aplicación práctica, siendo parte esenciales para la generación de reportes, estadísticas y búsquedas específicas dentro del sistema. Contribuyendo significativamente a la formación profesional del estudiante.

Uno de los aportes académicos a considerar fue la entrega de un modelo de base de datos que, con futuras mejoras, actualizaciones y nuevas implementaciones puede utilizarse como recurso educativo, para la institución, sirviendo de ejemplo o referencia a la asignatura. No obstante, se considera que el mayor valor de esta radica en el fortalecimiento de la formación



Universidad  
Nacional  
de Loja

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

*Carrera Ingeniería en Computación*

profesional y práctica del estudiante, por medio de la integración de conocimiento teórico y su aplicación real, en un entorno académico.

## **7. Recomendaciones:**

Se sugiere que en la asignatura se continúe fortaleciendo el trabajo práctico mediante el desarrollo de proyectos que permitan al estudiante aplicar lo aprendido en clase. Este tipo de actividades facilita la comprensión de los temas y permite relacionar de mejor manera la teoría con situaciones reales. Como complemento, se podría considerar la incorporación de estudios de caso reales o simulados, relacionando y evaluando problemáticas comunes en el manejo de información, para que puedan ser analizadas y estudiadas, dando paso a brindar soluciones y reflexiones sobre la importancia del diseño correcto de una base de datos. permitiendo una conexión más directa a las posibles situaciones que podrían enfrentar en su entorno profesional o académico.

## **8. Firma y Validación**

**Estudiante**

**Docente Supervisor**

---

Arelys Ajila

Ing. René Guamán

---

Viviana Córdova

Ing. René Guamán

---

Fabricio Ruiz

Ing. René Guamán