

Proyecto de Bases de datos para la gestión de peticiones, quejas y reclamos (PQR) en empresas prestadoras de salud (EPS) en Colombia

Sebastian Moncada A.¹, Johan Sanchez R.²

¹⁻²Facultad de Ingeniería y Ciencias Basicas, ³Dpto. de Comunicación,
Universidad Central

Maestria en Analítica de Datos

Curso de Bases de Datos

Bogotá, Colombia

¹jmoncadaa@ucentral.edu.com, ²jsanchezr26@ucentral.edu.co

October 7, 2023

Contents

1	Introducción	3
2	Características del proyecto	3
2.1	Titulo del proyecto de investigación	3
2.2	Objetivo general	3
2.2.1	Objetivos especificos	3
2.3	Alcance	4
2.4	Pregunta de investigación	4
2.5	Hipotesis	5
3	Reflexiones sobre el origen de datos e información (Max 400 Palabras) - (Primera entrega)	6
3.1	¿Cual es el origen de los datos e información ? (Max 100 Palabras) - (Primera entrega)	6
3.2	¿Cuales son las consideraciones legales o eticas del uso de la información? (Max 100 Palabras) - (Primera entrega)	6
3.3	¿Cuales son los retos de la información y los datos que utilizara en la base de datos en terminos de la calidad y la consolidación? (Max 100 Palabras) - (Primera entrega)	7
3.4	¿Que espera de la utilización de un sistema de Bases de Datos para su proyecto? (Max 100 Palabras) - (Primera entrega)	7

4	Diseño del Modelo de Datos del SMBD (Sistema Manejador de Bases de Datos)(Primera entrega)	8
4.1	Características del SMBD (Sistema Manejador de Bases de Datos) para el proyecto (Primera entrega)	8
4.2	Diagrama modelo de datos (Primera entrega)	8
4.3	Imágenes de la Base de Datos (Primera entrega)	8
4.4	Código SQL - lenguaje de definición de datos (DDL) (Primera entrega)	10
4.5	Código SQL - Manipulación de datos (DML) (Primera entrega) . .	12
4.6	Código SQL + Resultados: Vistas (Primera entrega)	14
4.7	Código SQL + Resultados: Triggers (Primera entrega)	15
4.8	Código SQL + Resultados: Funciones (Primera entrega)	15
4.9	Código SQL + Resultados: procedimientos almacenados (Primera entrega)	16
5	Bases de Datos No-SQL (Segunda entrega)	17
5.1	Diagrama Bases de Datos No-SQL (Segunda entrega)	17
5.2	SMBD utilizado para la Base de Datos No-SQL (Segunda entrega)	17
6	Aplicación de ETL (Extract, Transform, Load) y Bodega de Datos (Tercera entrega)	18
6.1	Ejemplo de aplicación de ETL y Bodega de Datos (Tercera entrega)	18
6.2	Automatización de Datos (Tercera entrega)	18
6.3	Integración de Datos (Tercera entrega)	18
7	Proximos pasos (Tercera entrega)	19
8	Lecciones aprendidas (Tercera entrega)	20
9	Bibliografía	21

1 Introducción

Las Empresas Promotoras de Salud (EPS) desempeñan un papel crucial en el ecosistema de la atención médica, actuando como intermediarios entre los pacientes y los proveedores de servicios de salud. En este contexto, la gestión eficaz de Peticiones, Quejas y Reclamos (PQR) se ha convertido en un pilar esencial para garantizar la satisfacción de los usuarios y la mejora constante de los servicios de salud.

Este proyecto se centra en abordar la imperante necesidad de optimizar la gestión de PQR en las EPS mediante la creación de una base de datos integral. Esta base de datos no solo funcionará como un almacén de información, sino como un sistema completo que permitirá un seguimiento minucioso de cada caso, desde su registro inicial hasta su resolución. Este enfoque no solo agilizará los procesos internos de las EPS, sino que también facilitará la toma de decisiones informadas al proporcionar una visión completa de las preocupaciones de los usuarios.

El proyecto abordará diversos aspectos técnicos esenciales, como el diseño meticuloso de la base de datos, la definición de relaciones entre tablas, entre otros. La seguridad de los datos ocupará un lugar prioritario en la planificación, garantizando la confidencialidad y la integridad de la información de los usuarios. A lo largo de este informe, se describirán los pasos necesarios para llevar a cabo este proyecto, las tecnologías a implementar, los beneficios esperados y su importancia en el contexto en constante evolución de la atención médica.

2 Características del proyecto

2.1 Título del proyecto de investigación

Proyecto de Bases de datos para la gestión de peticiones, quejas y reclamos (PQR) en empresas prestadoras de salud (EPS) en Colombia.

2.2 Objetivo general

Este proyecto tiene como objetivo principal la creación y desarrollo de una base de datos integral y eficiente para la gestión de las PQR presentadas por los usuarios de las EPS. La implementación de esta base de datos permitirá una administración más ágil y efectiva de los procesos relacionados con la atención y resolución de PQR, contribuyendo a la mejora continua de los servicios de salud ofrecidos.

2.2.1 Objetivos específicos

- Definir una arquitectura de base de datos que permita un almacenamiento organizado y rápido acceso a la información de PQR de las EPS.

- Crear una base de datos que pueda adaptarse a cambios futuros en la gestión de PQR y a un aumento en la cantidad de datos sin comprometer el rendimiento.
- facilitar la recopilación, búsqueda y análisis de PQR para una gestión más rápida y efectiva de los casos, lo que conducirá a una atención al cliente mejorada.
- Facilitar la identificación de patrones y tendencias en las PQR, lo que permitirá a las EPS tomar medidas proactivas para resolver problemas y mejorar la calidad de la atención médica.

2.3 Alcance

Este proyecto se concentra en el diseño exhaustivo de la estructura de la base de datos para la gestión de Peticiones, Quejas y Reclamos (PQR) en las Empresas Promotoras de Salud (EPS). Su alcance abarca los siguientes aspectos clave:

- Diseño de la Base de Datos: Se desarrollará un diseño completo de la base de datos que comprende la creación de tablas, definición de campos y atributos, y establecimiento de relaciones entre tablas para capturar y organizar de manera eficiente la información relacionada con las PQR.
- Identificación de Campos Relevantes: Se definirán los campos esenciales para registrar datos cruciales, como la información del usuario (nombre, contacto), la fecha de presentación de la PQR, la categoría o tipo de solicitud, el estado de resolución y cualquier otra información relevante.
- Normalización de Datos: Se aplicarán principios de normalización de bases de datos para evitar la redundancia de datos y garantizar la integridad y coherencia de la información almacenada.
- Especificación de Restricciones de Integridad: Se establecerán restricciones de integridad para mantener la coherencia de los datos, como claves primarias, foráneas y restricciones de unicidad.
- Documentación del Diseño: Se proporcionará una documentación detallada que describe la estructura de la base de datos, las tablas, campos, relaciones y restricciones, facilitando la comprensión y el mantenimiento del sistema en el futuro.

2.4 Pregunta de investigación

¿Cuál es el impacto de la implementación de una base de datos eficiente en la gestión de PQR en las EPS?

2.5 Hipotesis

La implementación exitosa de una base de datos específica para la gestión de PQR en las EPS conlleva una reducción significativa en los tiempos de respuesta a las solicitudes de los usuarios.

Esta hipótesis parte de la premisa de que al implementar una base de datos diseñada especialmente para la gestión de PQR en las EPS, se pueden agilizar los procesos de atención a los usuarios. La lógica subyacente es que la base de datos proporcionará a los empleados de las EPS un acceso más rápido a la información relevante, lo que les permitirá responder a las solicitudes de los usuarios de manera más eficiente. Además, al mejorar la organización de los datos y su accesibilidad, se espera que se reduzcan los tiempos de búsqueda y seguimiento de casos, lo que tendrá un impacto positivo en la satisfacción de los usuarios, quienes experimentarán una atención más oportuna y efectiva.

3 Reflexiones sobre el origen de datos e información (Max 400 Palabras) - (*Primera entrega*)

En el ámbito de la atención médica, la gestión eficiente de las PQRS (Peticiones, Quejas, Reclamos y Sugerencias) es esencial para garantizar la satisfacción de los afiliados y mantener la calidad de los servicios. En este contexto, los datos utilizados para evaluar y mejorar la atención provienen de un histórico de PQRS que ha sido recibido por una EPS.

Es importante destacar que las PQRS pueden variar dependiendo de la ciudad o región a la que pertenezcan los afiliados. Esta variabilidad geográfica se debe a una serie de factores, incluyendo las condiciones de salud prevalentes en la zona, la disponibilidad de recursos médicos y las particularidades socioeconómicas de la población local. Por lo tanto, una comprensión precisa de la geolocalización de las PQRS es esencial para abordarlas de manera efectiva.

Entre las PQRS más comunes, destacan las relacionadas con la necesidad de oxígeno o la solicitud de atención médica domiciliaria. Estas solicitudes son críticas, ya que a menudo están vinculadas a la salud y el bienestar de los afiliados. La respuesta oportuna y adecuada a tales solicitudes es fundamental para garantizar la seguridad y el cuidado de los pacientes.

La gestión efectiva de las PQRS no solo es un requisito para brindar una atención de calidad, sino que también tiene implicaciones legales. Si no se atienden de manera adecuada y oportuna, las PQRS pueden escalar a tutelas, un proceso legal que permite a los afiliados buscar la protección de sus derechos a través de la intervención de una autoridad judicial. Esto no solo puede ser costoso y llevar tiempo, sino que también puede dañar la reputación de la EPS y socavar la confianza de los afiliados.

3.1 ¿Cual es el origen de los datos e información ? (Max 100 Palabras) - (*Primera entrega*)

Los datos anonimizados se generaron sintéticamente a partir de PQRS encontradas en línea, que simulan quejas de usuarios sobre servicios de una EPS. Además, se incorporó información demográfica para mejorar la clasificación demográfica de las PQRS y optimizar su análisis.

3.2 ¿Cuales son las consideraciones legales o eticas del uso de la información? (Max 100 Palabras) - (*Primera entrega*)

Para salvaguardar la privacidad y cumplir con las regulaciones legales, ciertos registros han sido anonimizados en este conjunto de datos, que incluye información de personas que solicitan servicios. Además, gran parte de los datos ha sido generada sintéticamente y adaptada para los fines del proyecto. Es esencial respetar la confidencialidad de esta información y evitar su divulgación.

3.3 ¿Cuales son los retos de la información y los datos que utilizara en la base de datos en terminos de la calidad y la consolidación? (Max 100 Palabras) - (Primera entrega)

El desafío incluye la tarea crucial de mapear y validar la información antes de cargarla en la base de datos. Esto es fundamental para prevenir problemas de rendimiento y lectura, especialmente cuando se enfrentan caracteres especiales o codificaciones incompatibles. Además, debemos mantener la confidencialidad de los datos en todo momento.

3.4 ¿Que espera de la utilización de un sistema de Bases de Datos para su proyecto? (Max 100 Palabras) - (Primera entrega)

Se anticipa una mejora significativa en la organización de la información en el sistema de la base de datos, lo que permitirá consultas más eficientes gracias a la implementación de un modelo relacional adecuado. Además, esta optimización allana el camino hacia la posibilidad de una integración web para el consumo y la visualización más accesible de las PQRS.

4 Diseño del Modelo de Datos del SMBD (Sistema Manejador de Bases de Datos) (Primera entrega)

4.1 Características del SMBD (Sistema Manejador de Bases de Datos) para el proyecto (Primera entrega)

Nuestra base de datos está diseñada para gestionar una alta carga transaccional en tiempo real, ya que se espera que reciba una gran variedad de consultas. Además, se debe garantizar que el proceso de carga masiva de información, independientemente de la cantidad, se realice de manera eficiente para que la información esté disponible en el menor tiempo posible.

Por estas razones, hemos optado por utilizar PostgreSQL como nuestro Sistema de Gestión de Bases de Datos (SMBD). PostgreSQL ha demostrado ser una elección sólida que satisface las necesidades mencionadas anteriormente. Además, su reputación en la industria y el constante respaldo de la comunidad lo han posicionado como uno de los SMBD más populares y confiables. Esta elección se basa en la capacidad de PostgreSQL para brindar un rendimiento óptimo tanto en transacciones en tiempo real como en cargas masivas de datos, lo que garantiza que nuestra base de datos cumpla con los estándares más exigentes en términos de disponibilidad y eficiencia.

4.2 Diagrama modelo de datos (Primera entrega)

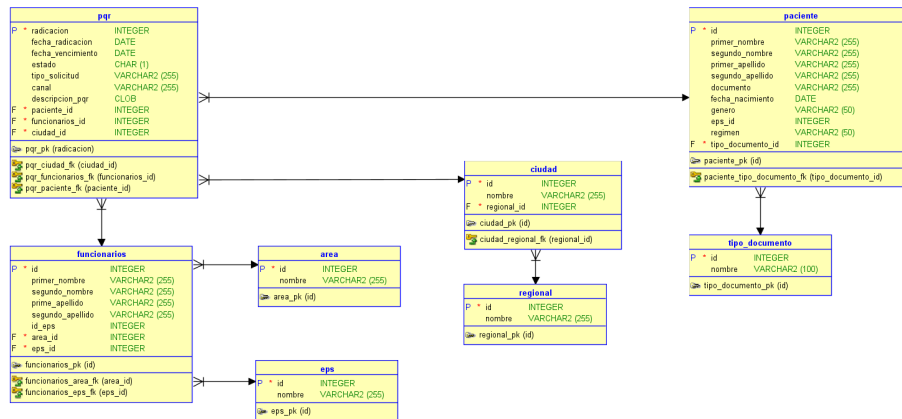


Figure 1: Modelo entidad relación.

4.3 Imágenes de la Base de Datos (Primera entrega)

Después de llevar a cabo el proceso de construcción de la base de datos y de realizar las operaciones de inserción de datos, hemos obtenido resultados

sumamente prometedores que arrojan luz sobre la eficacia y el potencial del proyecto. Este esfuerzo significativo ha dado lugar a una serie de hallazgos clave.

En cuanto a los resultados de las operaciones de inserción, podemos afirmar con satisfacción que se han completado de manera exitosa y sin errores significativos. Esto demuestra la eficiencia de nuestras metodologías y la capacidad de nuestro equipo para llevar a cabo tareas complejas de gestión de datos.

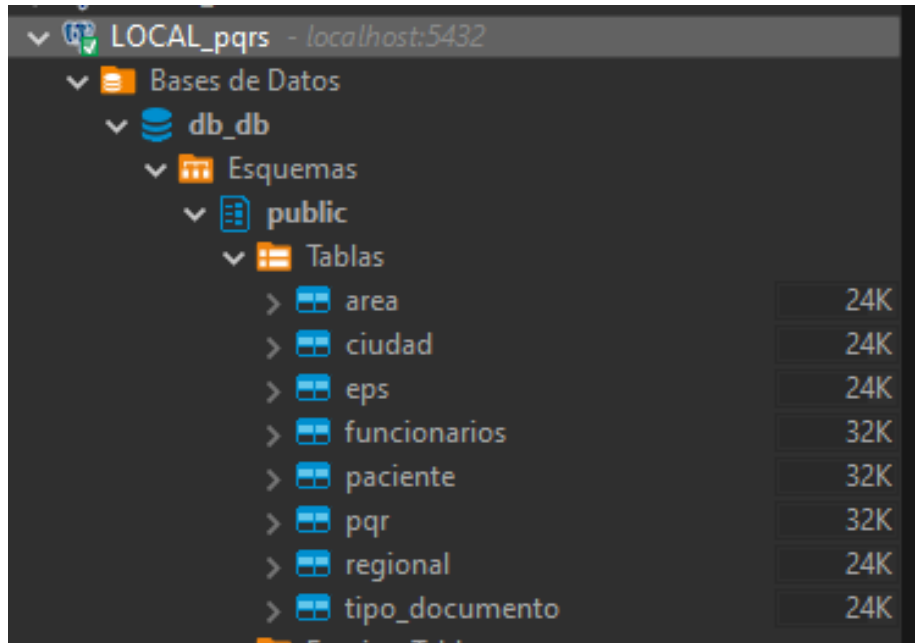


Tabla	Tamaño
area	24K
ciudad	24K
eps	24K
funcionarios	32K
paciente	32K
pqr	32K
regional	24K
tipo_documento	24K

Figure 2: Lista de tablas.

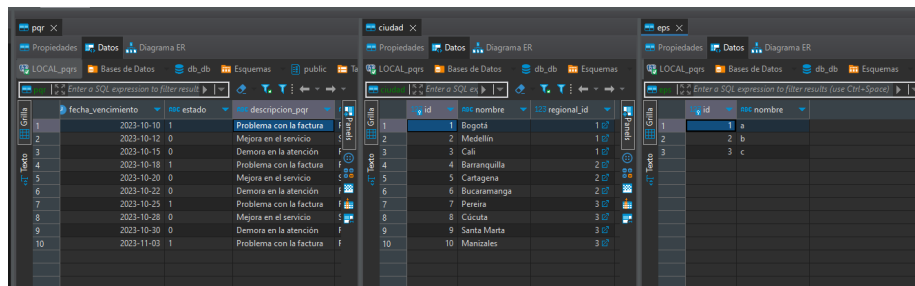


Tabla	ID	Nombre	regional_id
pqr	1	Problema con la factura	1
	2	Mejora en el servicio	1
	3	Demora en la atención	1
	4	Problema con la factura	2
	5	Mejora en el servicio	2
	6	Demora en la atención	2
	7	Problema con la factura	3
	8	Mejora en el servicio	3
	9	Demora en la atención	3
	10	Problema con la factura	3
ciudad	1	Bogotá	1
	2	Medellín	1
	3	Cali	1
	4	Barranquilla	2
	5	Cartagena	2
	6	Bucaramanga	2
	7	Pereira	3
	8	Cúcuta	3
	9	Santa Marta	3
	10	Manizales	3
eps	1	a	
	2	b	
	3	c	

Figure 3: Algunas tablas con datos.

4.4 Código SQL - lenguaje de definición de datos (DDL) (Primera entrega)

```
CREATE TABLE area (  
    id      INTEGER NOT NULL,  
    nombre  VARCHAR2(255)  
)  
ALTER TABLE area ADD CONSTRAINT area_pk PRIMARY KEY ( id );
```

```
CREATE TABLE ciudad (  
    id          INTEGER NOT NULL,  
    nombre      VARCHAR2(255),  
    regional_id INTEGER NOT NULL  
)  
ALTER TABLE ciudad ADD CONSTRAINT ciudad_pk PRIMARY KEY ( id );
```

```
CREATE TABLE eps (  
    id      INTEGER NOT NULL,  
    nombre  VARCHAR2(255)  
)  
ALTER TABLE eps ADD CONSTRAINT eps_pk PRIMARY KEY ( id );
```

```
CREATE TABLE funcionarios (  
    id              INTEGER NOT NULL,  
    primer_nombre   VARCHAR2(255),  
    segundo_nombre  VARCHAR2(255),  
    prime_apellido  VARCHAR2(255),  
    segundo_apellido VARCHAR2(255),  
    id_eps          INTEGER,  
    area_id         INTEGER NOT NULL,  
    eps_id          INTEGER NOT NULL  
)  
ALTER TABLE funcionarios ADD CONSTRAINT funcionarios_pk PRIMARY KEY ( id );
```

```
CREATE TABLE paciente (  
    id              INTEGER NOT NULL,  
    primer_nombre   VARCHAR2(255),  
    segundo_nombre  VARCHAR2(255),  
    primer_apellido VARCHAR2(255),  
    segundo_apellido VARCHAR2(255),  
    documento       VARCHAR2(255),
```

```

        fecha_nacimiento    DATE,
        genero               VARCHAR2(50),
        eps_id               INTEGER,
        regimen              VARCHAR2(50),
        tipo_documento_id    INTEGER NOT NULL
    )
ALTER TABLE paciente ADD CONSTRAINT paciente_pk PRIMARY KEY ( id );

CREATE TABLE pqr (
    radicacion               INTEGER NOT NULL,
    fecha_radicacion         DATE,
    fecha_vencimiento        DATE,
    estado                   CHAR(1),
    tipo_solicitud            VARCHAR2(255),
    canal                     VARCHAR2(255),
    descripcion_pqr          CLOB,
    paciente_id              INTEGER NOT NULL,
    funcionarios_id          INTEGER NOT NULL,
    ciudad_id                INTEGER NOT NULL
)
ALTER TABLE pqr ADD CONSTRAINT pqr_pk PRIMARY KEY ( radicacion );

CREATE TABLE regional (
    id                       INTEGER NOT NULL,
    nombre VARCHAR2(255)
)
ALTER TABLE regional ADD CONSTRAINT regional_pk PRIMARY KEY ( id );

CREATE TABLE tipo_documento (
    id                       INTEGER NOT NULL,
    nombre VARCHAR2(100)
)
ALTER TABLE tipo_documento ADD CONSTRAINT tipo_documento_pk PRIMARY KEY ( id );

ALTER TABLE ciudad
    ADD CONSTRAINT ciudad_regional_fk FOREIGN KEY ( regional_id )
        REFERENCES regional ( id )
        NOT DEFERRABLE;

ALTER TABLE funcionarios
    ADD CONSTRAINT funcionarios_area_fk FOREIGN KEY ( area_id )

```

```

        REFERENCES area ( id )
    NOT DEFERRABLE;

ALTER TABLE funcionarios
    ADD CONSTRAINT funcionarios_eps_fk FOREIGN KEY ( eps_id )
        REFERENCES eps ( id )
    NOT DEFERRABLE;

ALTER TABLE paciente
    ADD CONSTRAINT paciente_tipo_documento_fk FOREIGN KEY ( tipo_documento_id )
        REFERENCES tipo_documento ( id )
    NOT DEFERRABLE;

ALTER TABLE pqr
    ADD CONSTRAINT pqr_ciudad_fk FOREIGN KEY ( ciudad_id )
        REFERENCES ciudad ( id )
    NOT DEFERRABLE;

ALTER TABLE pqr
    ADD CONSTRAINT pqr_funcionarios_fk FOREIGN KEY ( funcionarios_id )
        REFERENCES funcionarios ( id )
    NOT DEFERRABLE;

ALTER TABLE pqr
    ADD CONSTRAINT pqr_paciente_fk FOREIGN KEY ( paciente_id )
        REFERENCES paciente ( id )
    NOT DEFERRABLE;

```

4.5 Código SQL - Manipulación de datos (DML) *(Primera entrega)*

```

--Insert Eps
INSERT INTO public.eps (id,nombre) VALUES (1,'Compensar')
INSERT INTO public.eps (id,nombre) VALUES (2,'Sanitas')
INSERT INTO public.eps (id,nombre) VALUES (3,'Nueva EPS')

--Insert area
INSERT INTO public.area (id,nombre) VALUES (1,'Atencion al cliente')
INSERT INTO public.area (id,nombre) VALUES (2,'Medicina')
INSERT INTO public.area (id,nombre) VALUES (3,'Administrativa')

--Insert document-type
INSERT INTO public.tipo_documento (id,nombre) VALUES (1,'CC')
INSERT INTO public.tipo_documento (id,nombre) VALUES (2,'TI')
INSERT INTO public.tipo_documento (id,nombre) VALUES (3,'PA')

```

```

--Regional
INSERT INTO public.regional (id,nombre) VALUES (1,'Bogota')
INSERT INTO public.regional (id,nombre) VALUES (2,'Barranquilla')
INSERT INTO public.regional (id,nombre) VALUES (3,'Sur Occidente')

--Ciudades
INSERT INTO public.ciudad (id,nombre,regional_id) VALUES (1,'Bogotá',1)
INSERT INTO public.ciudad (id,nombre,regional_id) VALUES (2,'Medellín',1)
INSERT INTO public.ciudad (id,nombre,regional_id) VALUES (3,'Cali',1)
INSERT INTO public.ciudad (id,nombre,regional_id) VALUES (4,'Barranquilla',2)
INSERT INTO public.ciudad (id,nombre,regional_id) VALUES (5,'Cartagena',2)
INSERT INTO public.ciudad (id,nombre,regional_id) VALUES (6,'Bucaramanga',2)
INSERT INTO public.ciudad (id,nombre,regional_id) VALUES (7,'Pereira',3)
INSERT INTO public.ciudad (id,nombre,regional_id) VALUES (8,'Cúcuta',3)
INSERT INTO public.ciudad (id,nombre,regional_id) VALUES (10,'Manizales',3)

--Funcionarios
INSERT INTO funcionarios (id, primer_nombre, segundo_nombre,
prime_apellido, segundo_apellido, area_id, eps_id)
VALUES (1, 'Juan', 'Carlos', 'Gómez', 'López', 1, 1);
INSERT INTO funcionarios (id, primer_nombre, segundo_nombre,
prime_apellido, segundo_apellido, area_id, eps_id)
VALUES (2, 'Ana', 'María', 'Rodríguez', 'Pérez', 2, 1);
INSERT INTO funcionarios (id, primer_nombre, segundo_nombre,
prime_apellido, segundo_apellido, area_id, eps_id)
VALUES (3, 'Luis', 'Felipe', 'Martínez', 'García', 1, 2);
INSERT INTO funcionarios (id, primer_nombre, segundo_nombre,
prime_apellido, segundo_apellido, area_id, eps_id)
VALUES (4, 'María', 'Isabel', 'Fernández', 'Ramírez', 2, 2);

--Pacientes
INSERT INTO paciente (id, primer_nombre, segundo_nombre, primer_apellido,
segundo_apellido, documento, fecha_nacimiento, genero, eps_id, regimen,
tipo_documento_id)
VALUES (1, 'Juan', 'Carlos', 'Gómez', 'López', '123456789', '1990-05-15',
'Masculino', 1, 'Contributivo', 1);
INSERT INTO paciente (id, primer_nombre, segundo_nombre, primer_apellido,
segundo_apellido, documento, fecha_nacimiento, genero, eps_id, regimen,
tipo_documento_id)
VALUES (2, 'Ana', 'María', 'Rodríguez', 'Pérez', '987654321', '1985-08-20',
'Femenino', 2, 'Subsidiado', 2);
INSERT INTO paciente (id, primer_nombre, segundo_nombre, primer_apellido,
segundo_apellido, documento, fecha_nacimiento, genero, eps_id, regimen,
tipo_documento_id)

```

```
VALUES (3, 'Luis', 'Felipe', 'Martínez', 'García', '456789123', '1992-03-10',
'Masculino', 3, 'Contributivo', 3);
```

```
--PQRS
```

```
INSERT INTO pqr (radicacion, fecha_radicacion, fecha_vencimiento, estado,
tipo_solicitud, canal, descripcion_pqr, paciente_id, funcionarios_id, ciudad_id)
VALUES (1, '2023-09-10', '2023-10-10', 1, 'Reclamo', 'Correo',
'Problema con la factura', 2, 1, 3);
INSERT INTO pqr (radicacion, fecha_radicacion, fecha_vencimiento, estado,
tipo_solicitud, canal, descripcion_pqr, paciente_id, funcionarios_id, ciudad_id)
VALUES (2, '2023-09-12', '2023-10-12', 0, 'Sugerencia', 'Teléfono',
'Mejora en el servicio', 1, 3, 2);
INSERT INTO pqr (radicacion, fecha_radicacion, fecha_vencimiento, estado,
tipo_solicitud, canal, descripcion_pqr, paciente_id, funcionarios_id, ciudad_id)
VALUES (3, '2023-09-15', '2023-10-15', 0, 'Reclamo',
'Chat en Línea', 'Demora en la atención', 3, 2, 1);
INSERT INTO pqr (radicacion, fecha_radicacion, fecha_vencimiento, estado,
tipo_solicitud, canal, descripcion_pqr, paciente_id, funcionarios_id, ciudad_id)
VALUES (4, '2023-09-18', '2023-10-18', 1, 'Reclamo', 'Correo',
'Problema con la factura', 1, 1, 3);
```

4.6 Código SQL + Resultados: Vistas (*Primera entrega*)

Enter a SQL expression to filter results (use Ctrl+Space)						
	pac primer_nombre	pac primer_apellido	pac documento	123 radicacion	pac tipo_solicitud	pac descripcion_pqr
1	Ana	Rodríguez	987654321		1 Reclamo	Problema con la factura
2	Ana	Rodríguez	987654321		5 Sugerencia	Mejora en el servicio
3	Ana	Rodríguez	987654321		7 Reclamo	Problema con la factura
4	Luis	Martínez	456789123		6 Reclamo	Demora en la atención
5	Luis	Martínez	456789123		3 Reclamo	Demora en la atención
6	Luis	Martínez	456789123		9 Reclamo	Demora en la atención

Figure 4: Resultado View.

```
create or replace
view view_patients as
select
p.primer_nombre,
p.primer_apellido,
p.documento,
p2.radicacion,
p2.tipo_solicitud,
```

```

p2.descripcion_pqr
from
paciente p
inner join pqr p2 on
p2.paciente_id = p.id
order by
p.primer_nombre

SELECT * FROM view_patients;

```

4.7 Código SQL + Resultados: Triggers (*Primera entrega*)

```

CREATE TRIGGER insert_eps
  AFTER INSERT ON eps
  FOR EACH ROW
  EXECUTE FUNCTION create_eps_funcionario(1);

```

4.8 Código SQL + Resultados: Funciones (*Primera entrega*)

```

CREATE OR REPLACE FUNCTION resolve_pqr(id_pqr integer)
RETURNS integer AS $$
BEGIN
  UPDATE public.pqr set estado = 1 where id = id_pqr;
END;
$$ LANGUAGE plpgsql;

```

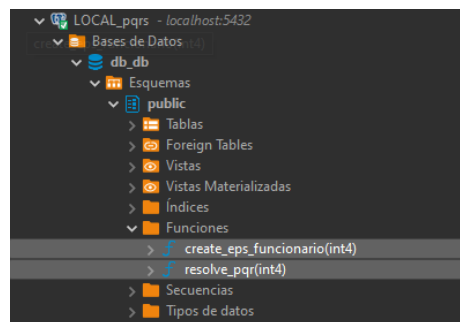


Figure 5: Resultado functions.

```

CREATE OR REPLACE FUNCTION create_eps_funcionario(id_eps integer)

```

```

RETURNS integer AS $$
BEGIN
    INSERT INTO funcionarios (id, primer_nombre, segundo_nombre, prime_apellido,
        segundo_apellido, area_id, eps_id)
VALUES (1, 'Lider', '', 'EPS', '', 1, 1);
END;
$$ LANGUAGE plpgsql;

```

4.9 Código SQL + Resultados: procedimientos almacenados (Primera entrega)

```

CREATE OR REPLACE FUNCTION pqrs_by_employee(employee_id INT)
RETURNS TABLE (
    id INT,
    nombre VARCHAR(255),
    pqr VARCHAR(255)
) AS $$
BEGIN
    RETURN QUERY
    SELECT f.ID, f.Nombre, pqr.PQR
    FROM funcionario f
    INNER JOIN pqr ON pqr.funcionario_id = f.id
    WHERE f.ID = employee_id;
END;
$$ LANGUAGE plpgsql;

```

Name	Value
Updated Rows	0
Query	<pre> CREATE OR REPLACE FUNCTION pqrs_by_employee(employee_id INT) RETURNS TABLE (id INT, nombre VARCHAR(255), pqr VARCHAR(255)) AS \$\$ BEGIN RETURN QUERY SELECT f.ID, f.Nombre, pqr.PQR FROM funcionario f INNER JOIN pqr ON pqr.funcionario_id = f.id WHERE f.ID = employee_id; END; \$\$ LANGUAGE plpgsql </pre>
Start time	Sat Oct 07 08:56:40 COT 2023

Figure 6: Resultado procedimiento almacenado.

5 Bases de Datos No-SQL (*Segunda entrega*)

5.1 Diagrama Bases de Datos No-SQL (*Segunda entrega*)

5.2 SMBD utilizado para la Base de Datos No-SQL (*Segunda entrega*)

- 6 Aplicación de ETL (Extract, Transform, Load) y Bodega de Datos** (*Tercera entrega*)
 - 6.1 Ejemplo de aplicación de ETL y Bodega de Datos** (*Tercera entrega*)
 - 6.2 Automatización de Datos** (*Tercera entrega*)
 - 6.3 Integración de Datos** (*Tercera entrega*)

7 Proximos pasos (*Tercera entrega*)

8 Lecciones aprendidas (*Tercera entrega*)

9 Bibliografía