

DESARROLLO DE SOFTWARE SOBRE LA GESTIÓN DE CASOS EN EL DERECHO CIVIL

DOCUMENTO DE ANEXOS

Autor(es):

MARUAN ENRIQUE ARIAS CUCUNUBÁ

ANTONIO JOSÉ MATA RODRÍGUEZ

Modalidad de Grado:

DESARROLLO TECNOLÓGICO

**UNIVERSIDAD EL BOSQUE
PROGRAMA DE INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERÍA
Bogotá, 2023**

TABLA DE CONTENIDO

Anexo No. 1: Diagrama del Modelo Biopsicosocial desde la perspectiva del Problema	1
Anexo No. 2: Diagrama del Modelo Biopsicosocial desde la perspectiva de la Solución	2
Anexo No. 3: Modelo BPMN AS - IS del Grupo Montserrat	
Anexo No. 4:Modelo BPMN TO - BE del Grupo Montserrat	
Anexo No. 5:Tablero de Control	
Anexo No. 6:Metodología	
Anexo No. 7:Diagrama C4- Contexto y Componentes	
Anexo No. 8:Diagrama Arquitectura AWS	
Anexo No. 9:Tabla costos - Productos AWS	
Anexo No. 10:Diagramas Base de datos	
Anexo No. 11: Actas reuniones Director	
Anexo No. 12: Actas reuniones Director	
Anexo No. 13: Actas reuniones Grupo Montserrat	
Anexo No. 14: Configuraciones AWS	

A. Anexo No. 1: Diagrama del Modelo Biopsicosocial desde la perspectiva del Problema

Usuario del problema: Trabajador Grupo Montserrat

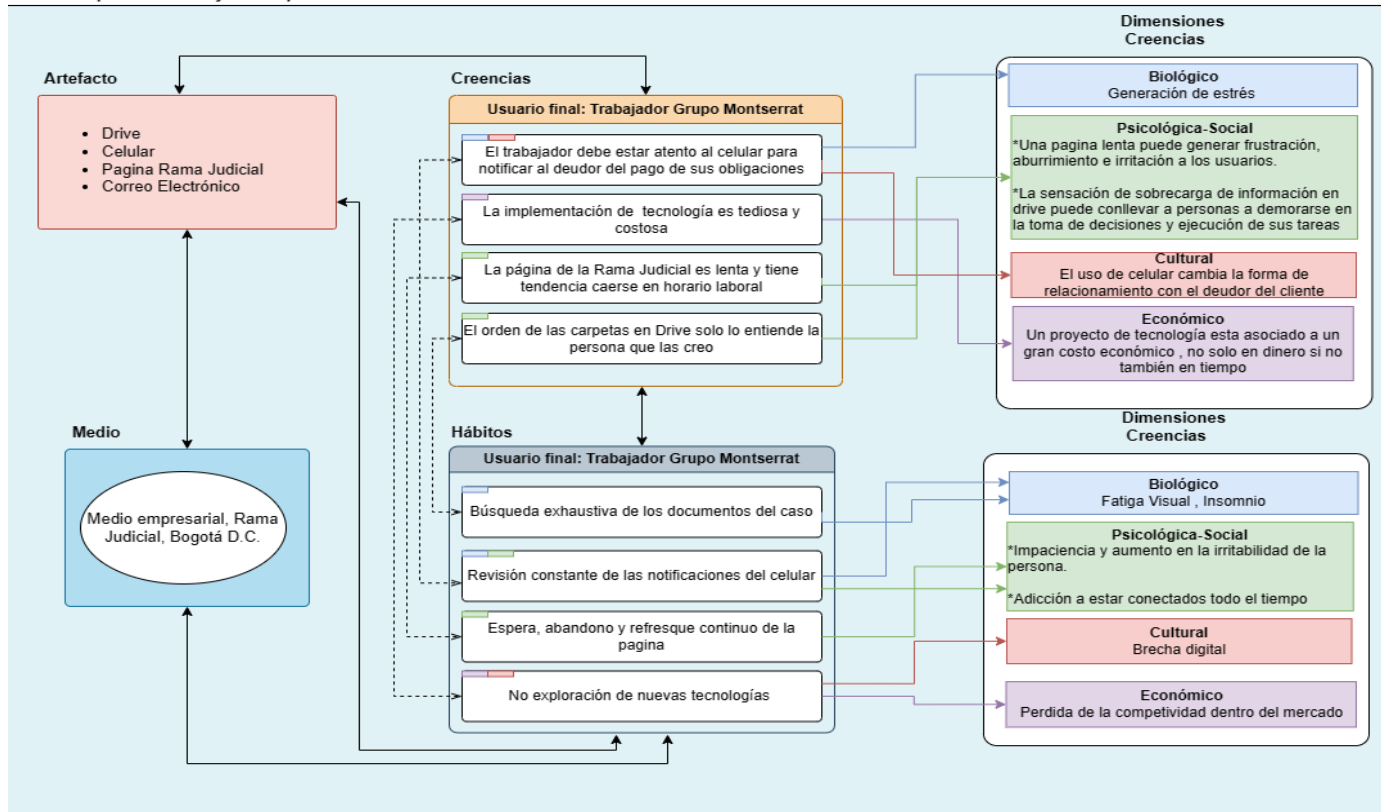


Figura No. 1. Diagrama del Modelo Biopsicosocial vista desde la perspectiva del Trabajador

Fuente: Elaboración propia

Usuario del problema: Deudor

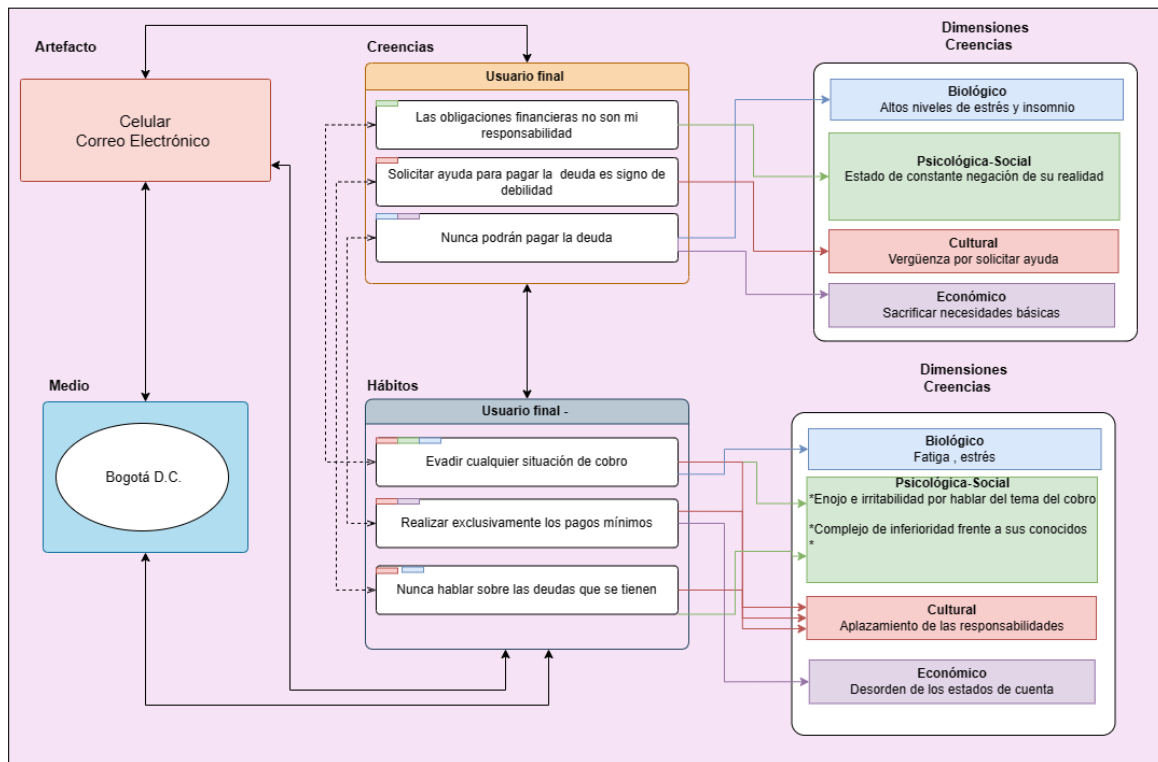


Figura No. 2. Diagrama del Modelo Biopsicosocial desde la perspectiva del Deudor

Fuente: Elaboración propia

B. Anexo No. 2: Diagrama del Modelo Biopsicosocial desde la perspectiva de la Solución

Usuario del problema: Trabajador Grupo Monserrat

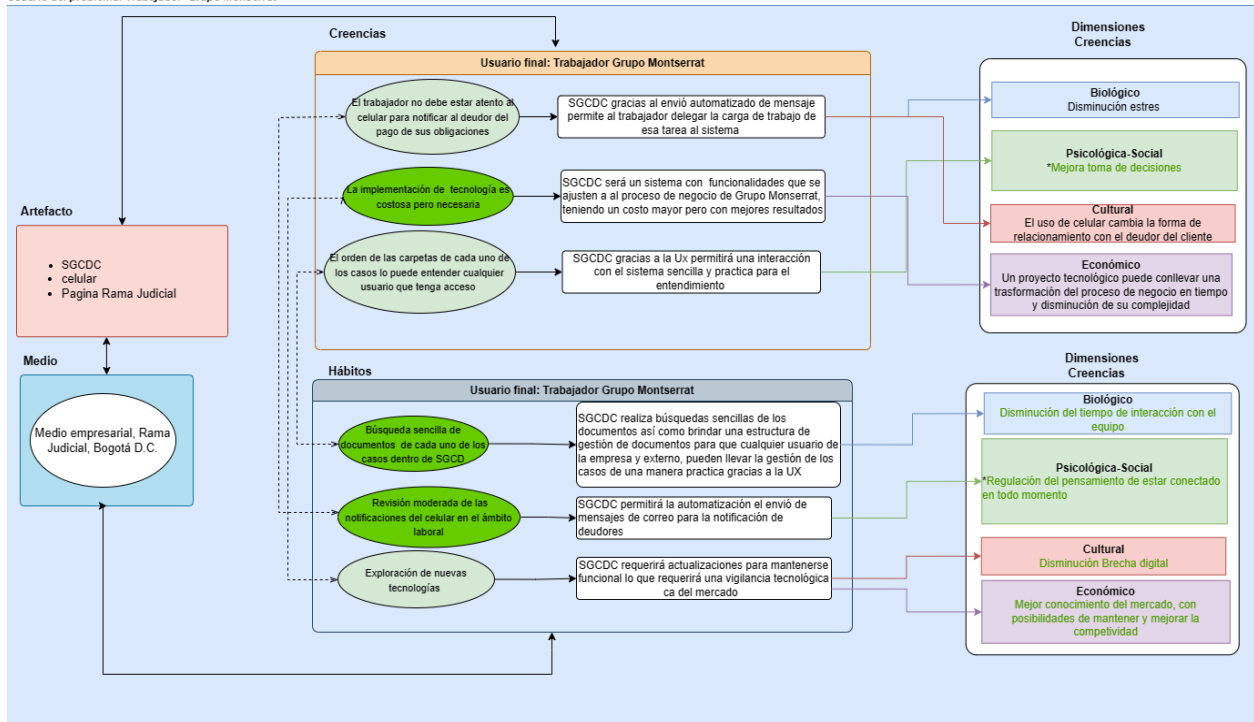


Figura No. 3. Diagrama del Modelo Biopsicosocial solución - empleado Grupo Montserrat

Fuente: Elaboración propia

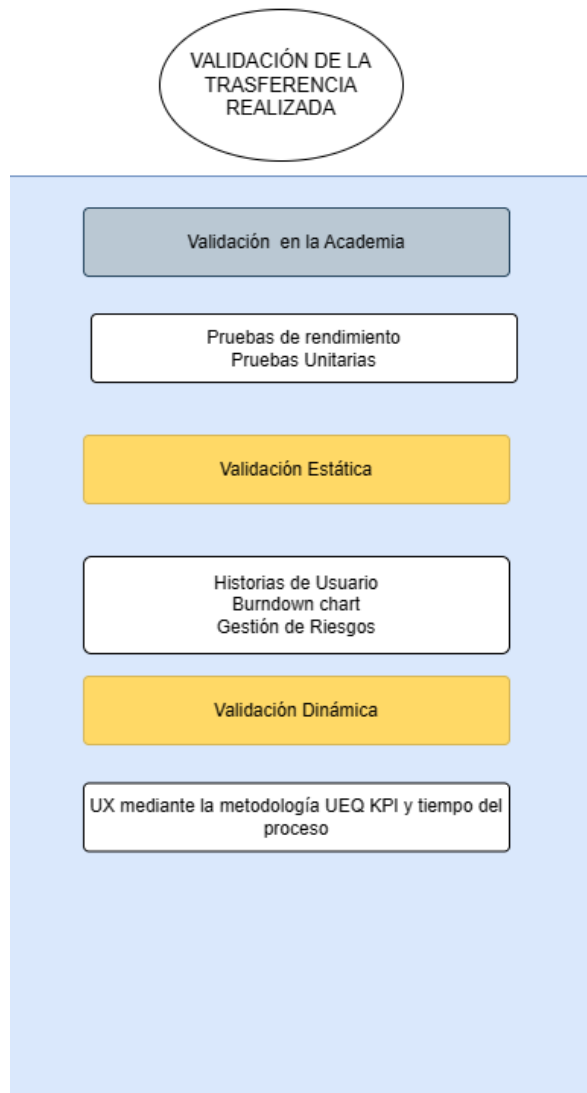


Figura No. 4. Diagrama del Modelo Biopsicosocial en la cual se analiza la validación de la transferencia realizada

Fuente: Elaboración propia

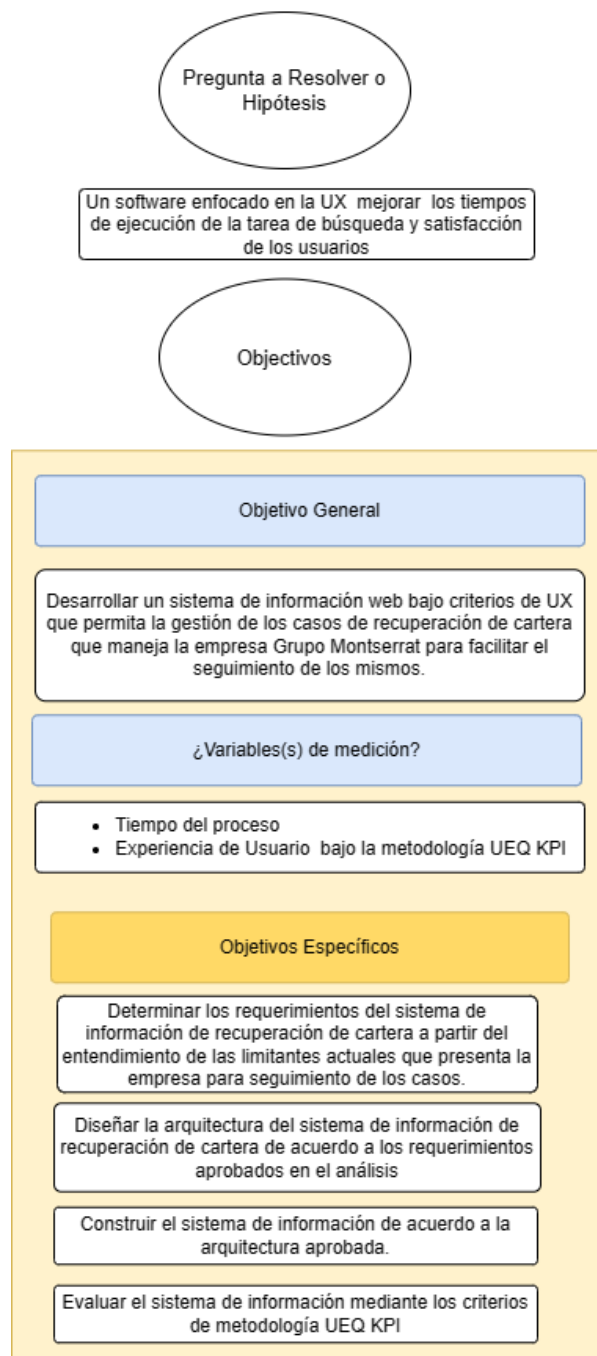


Figura No.5. Diagrama del Modelo Biopsicosocial donde se realiza un planteamiento de la Hipótesis a resolver y se declaran los objetivos

Fuente: Elaboración propia

C. Anexo No. 3: Modelo BPMN AS - IS del Grupo Montserrat

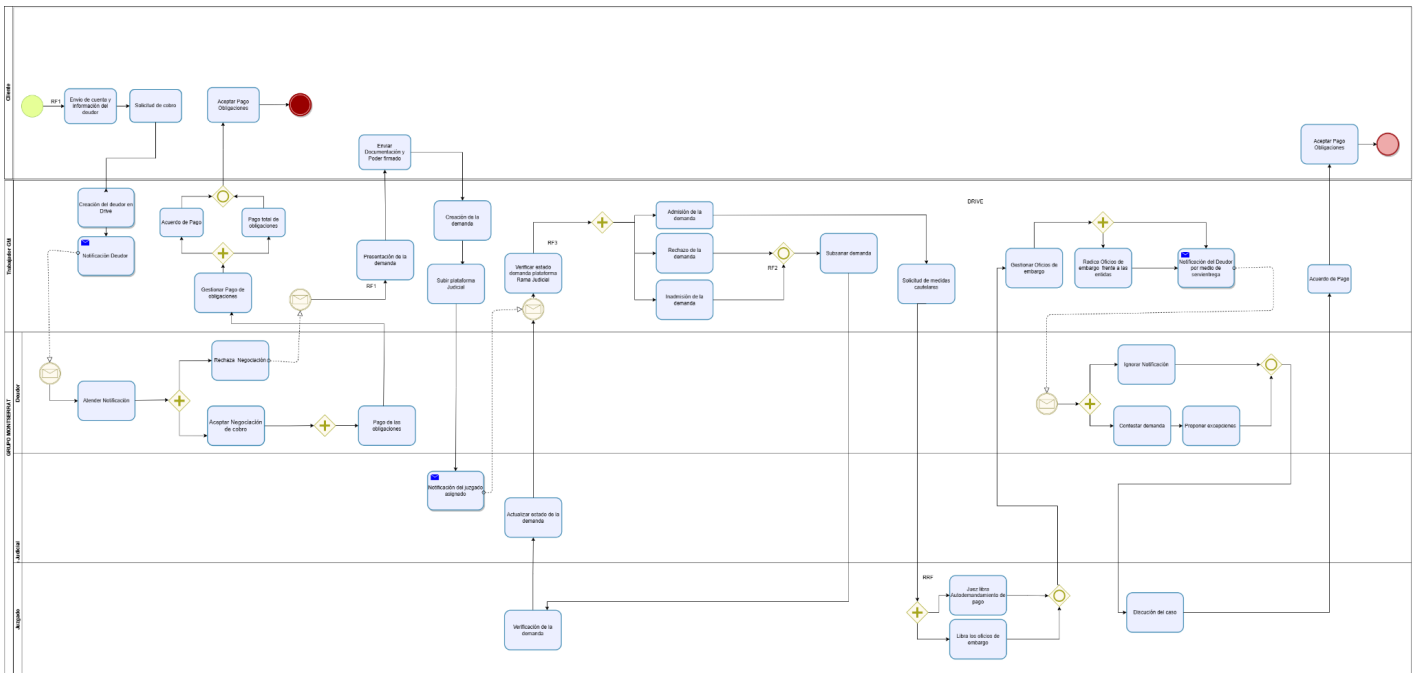


Figura No. 6. Modelo BPMN desde el estado AS - IS del Grupo Montserrat

Fuente: Elaboración propia

D. Anexo No. 4: Modelo BPMN TO - BE del Grupo Montserrat

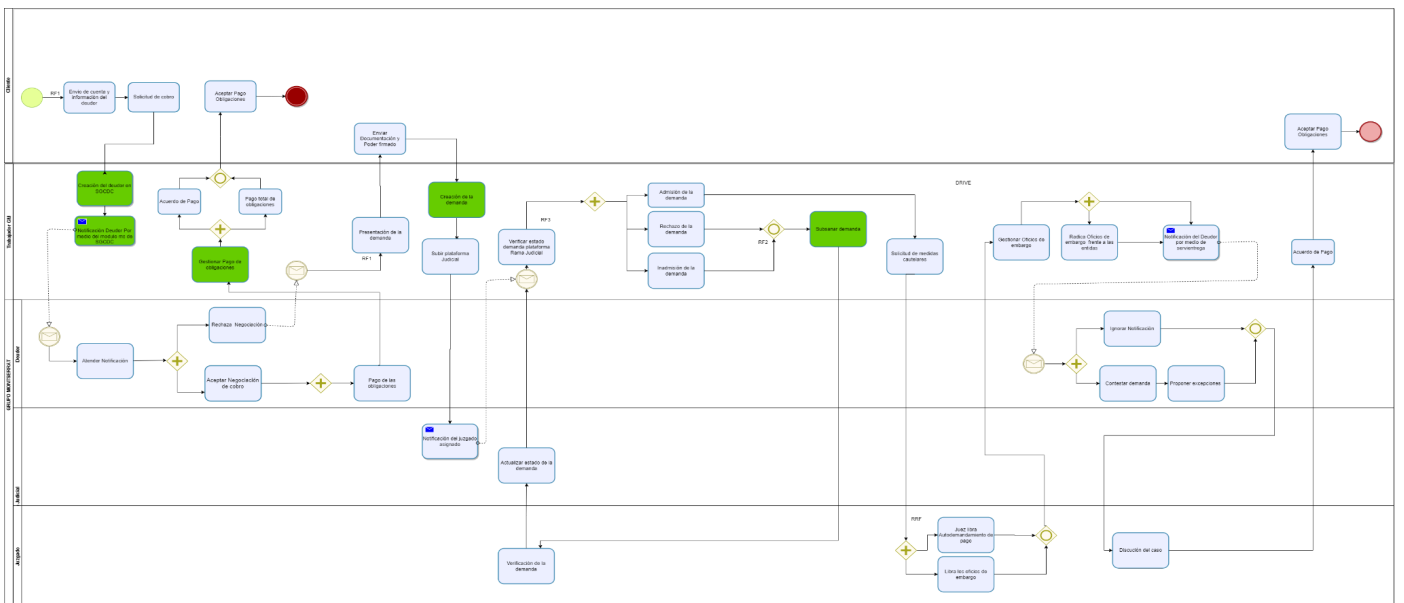


Figura No. 7. Modelo BPMN desde el estado TO -BE del Grupo Montserrat

Fuente: Elaboración propia

E. Anexo No. 5: Tablero de Control

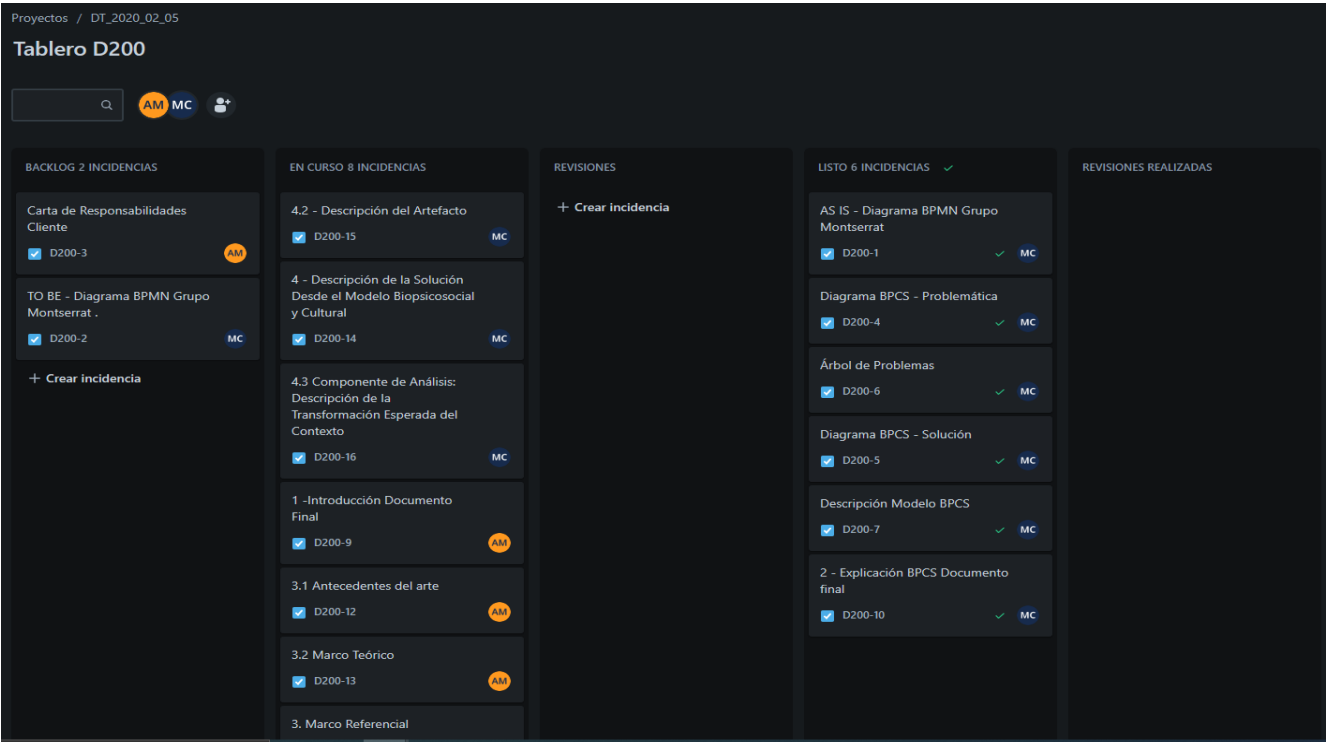


Figura No. 8. Tablero de Control realizado en la plataforma Jira
Fuente: Elaboración propia

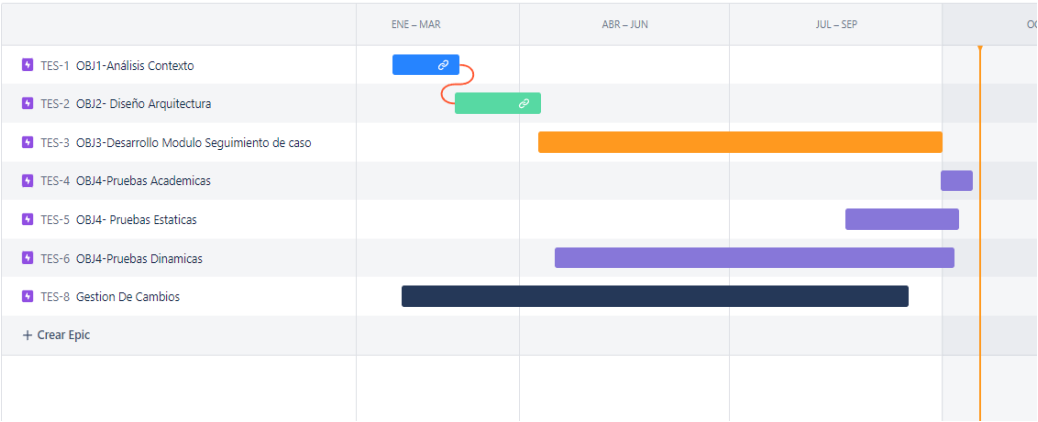


Figura No. 9. Cronograma del proyecto en la plataforma Jira
Fuente: Elaboración propia

F. Anexo No. 6: Metodología

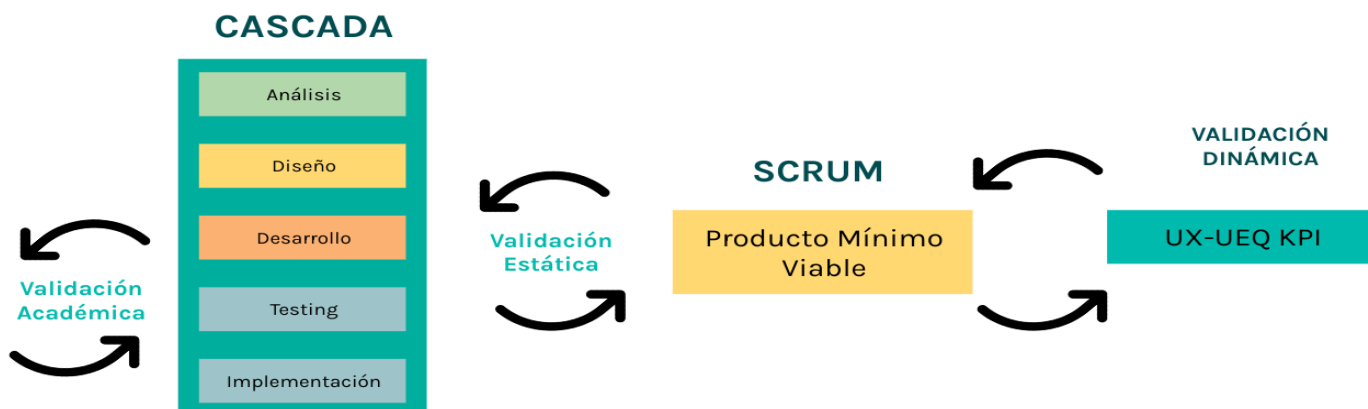


Figura No. 10. Modelo gráfico de la Metodología
Fuente: Elaboración propia

G. Anexo No. 7: Diagrama C4

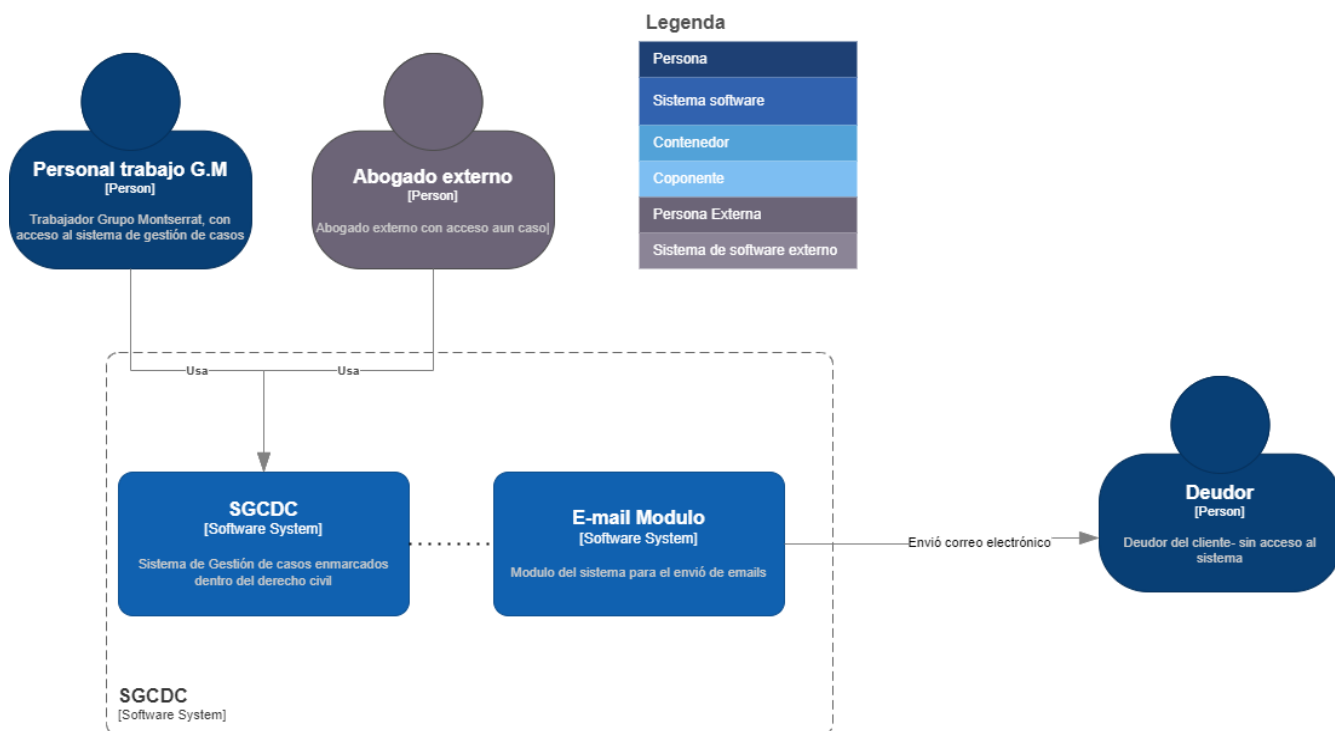


Figura No. 11. Diagrama C4 - contexto del sistema
Fuente: Elaboración propia

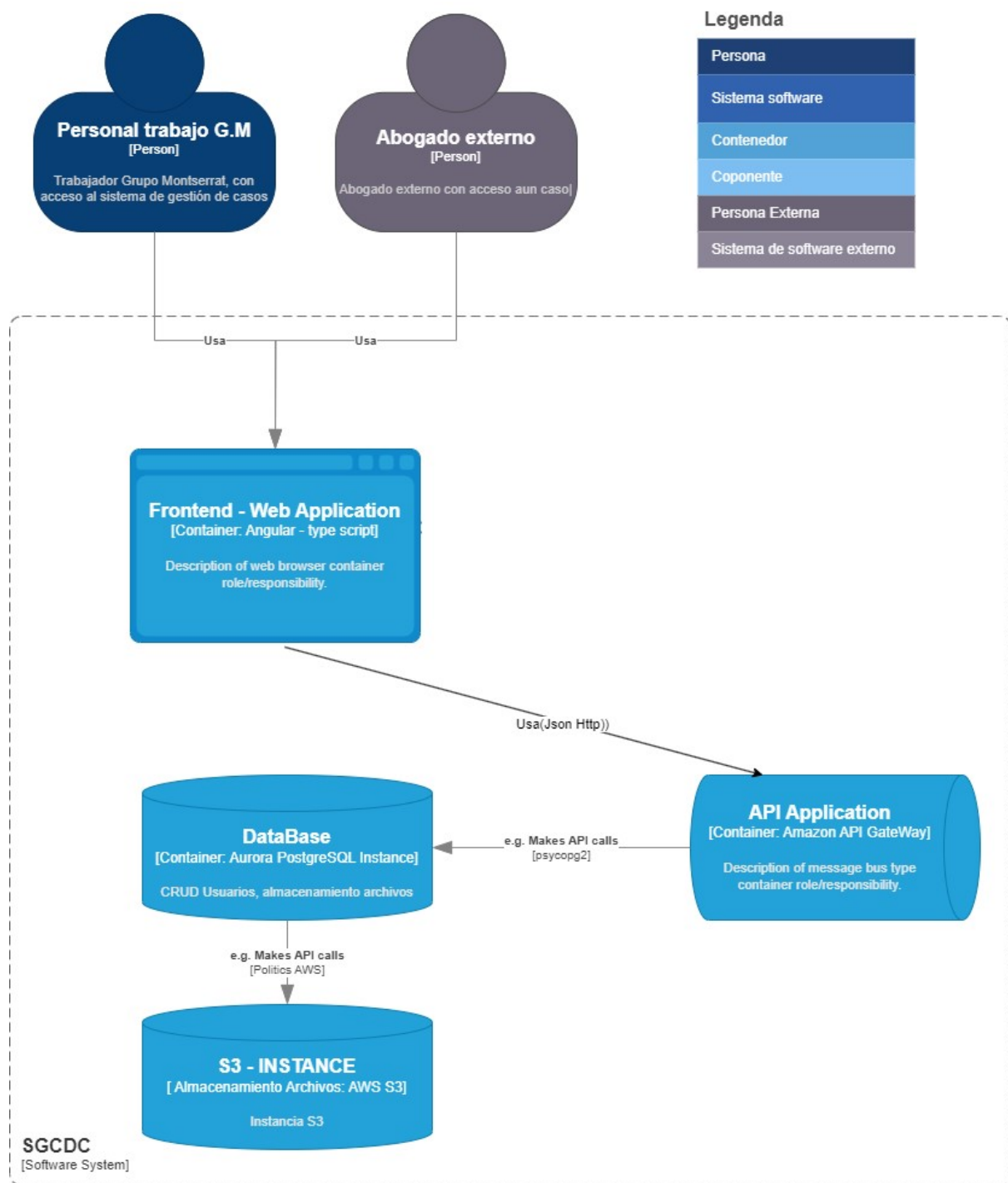


Figura No. 12. Diagrama C4 - contenedores del sistema
Fuente: Elaboración propia

H. Anexo No. 8: Diagrama Arquitectura AWS

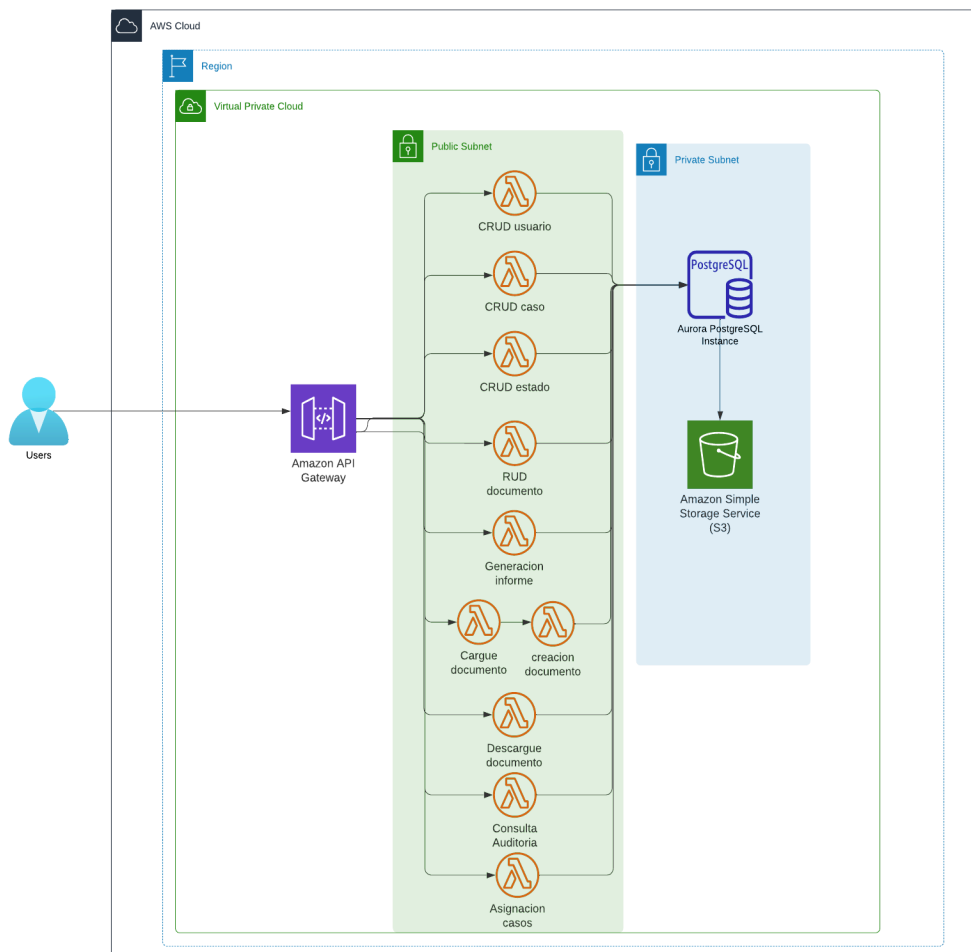


Figura No. 13. Diagrama Arquitectura AWS
Fuente: Elaboración propia

I. Anexo No. 9: Tabla costos - Productos AWS

Producto AWS						
Configuración - API						
Amazon API GateWay	tipo Locacion	Region		US East (Ohio)	Precio US	Tiempo
		Llamadas Api -HTTP			\$ -	12 meses
	Capa Gratuita	1000000			\$ -	
	REST API - request units	millions	Request	1	\$ 3.50	mensual
				2	\$ 7.00	mensual
	HTTP API - request units	millions	Request	1		
		Average size of each request	MB	1	\$ 2.00	mensual
				2	\$ 4.00	mensual
		millions	Request	2		
Average size of each request		MB	1	\$ 4.00	mensual	
			2	\$ 8.00	mensual	
AWS Lambda	tipo Locacion	Region		US East (Ohio)	Precio U	Tiempo
		1 million solicitudes- 1000000			\$ -	
	Capa Gratuita	Arquitectura x86	400 000 GB/segundos Tiempo Computo		\$ -	
	Arquitectura		x86 -arm			
	Number of requests -PER MOUTH	2000000			\$ 0.20	mensual
	Duration of each request (in ms)	100				
	Amount of memory allocated (GB)	2				
	Amount of ephemeral storage allocated(GB)	1				
Aurora PostgreSQL instance	tipo Locacion	Region		US East (Ohio)	Precio U	Tiempo
	Instancia	db.t4g.medium	vCPU:2 Memory: 4 GiB Network Performance: Up to 5 Gigabite			
	Pricing model (modelo precios)	OnDemand				
	Utilización	%Utilized/Moth	100	Hours/Day -- 24	\$ 54.82	mensual
			70	Hours/Day -- 17	\$ 38.83	mensual
			50	Hours/Day -- 12	\$ 28.18	mensual
			40	Hours/Day -- 9.6	\$ 22.00	mensual
	Database Storage	GB	10			
	Baseline IO rate	per second	1			
	Peak IO rate	per second	1			
Duration of peak IO activity	Hours per month	1				
Amazon simple store S3	S3 Standard storage	GB	130		\$ 3.88	mensual
	PUT, COPY, POST, LIST requests to S3 Standard	Number Request	1000000			mensual
	GET, SELECT, and all other requests from S3 Standard	Number Request	1000000			mensual
	Data returned by S3 Select	GB	10			mensual
	Data scanned by S3 Select	GB	65			mensual
	Data Transfer					mensual
	Inbound Data Transfer	Internet Free	TB	1		mensual
	Outbound Data Transfer	Internet 0.05 USD-0.09 per GB		15		mensual

Tabla 1. Costos productos al mes de AWS

Precio Dólar	TOTAL	Dólar	Peso Colombiano
\$ 4,606	Peor Escenario	\$ 71.90	\$ 331,171.40
	Escenario normal	\$ 52.41	\$ 241,400.46
	Escenario Bueno	\$ 41.76	\$ 192,346.56
	Escenario Optimo	\$ 35.58	\$ 163,881.48

Tabla 2. Estimación costos en 4 escenarios AWS al mes

Caso			
Campo	Tipo	Descripción	tamaño (bytes)
id_caso	int	Identificador único del caso	4
nombre_deudor	varchar (50)	Nombre del deudor	50
apellido_deudor	varchar (50)	Apellido del deudor	50
nombreCliente	varchar (50)	nombre del cliente	50
cc_deudor	varchar (15)	Número de identificación del usuario en el territorio Nacional	50
id_estado	int	Identificador único del estado actual del caso	4
fecha_creacion	date	Fecha de creación del caso	8
fecha_cierre	date	Fecha de cierre del caso	8
monto_deuda	decimal	Monto de la deuda o cantidad en disputa	8
ubicacion_geografica	varchar(50)	Ubicación geográfica del caso	50
etapa	varchar(50)	etapa juridico o prejudicial	50
juzgado	varchar(50)	Juzgado asignado	50
numero_radicado	varchar(50)	numero radicado caso	50

Tabla 3. Tabla de caso actualiza

Fuente: Elaboración propia

Trazabilidad			
Campo	Tipo	Descripción	tamaño (bytes)
id_traza	int	Identificador único de la acción de trazabilidad	4
id_usuario	int	Identificador único del usuario que realizó la acción	4
fecha_hora	datetime	Fecha y hora en que se realizó la acción	8
tabla_afectada	varchar	Nombre de la tabla afectada	50
tipo_accion	varchar	Tipo de acción realizada (INSERT, UPDATE, DELETE, etc.)	20
descripcion	varchar	Descripción detallada de la acción realizada	30

Tabla 4 - Tabla de trazabilidad

Fuente: Elaboración propia

J. Anexo No. 10: Diagramas Base de Datos

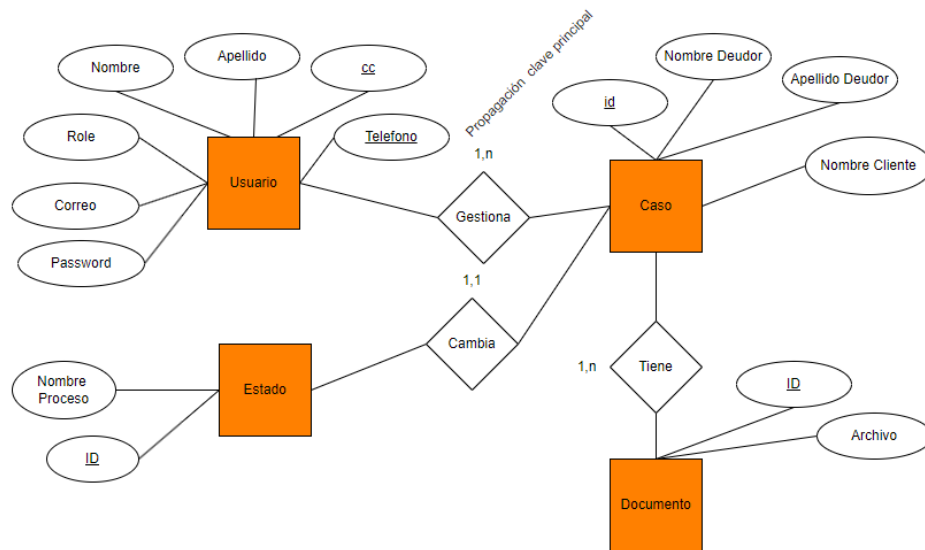


Figura No. 14. Diagrama Entidad relación
Fuente: Elaboración propia

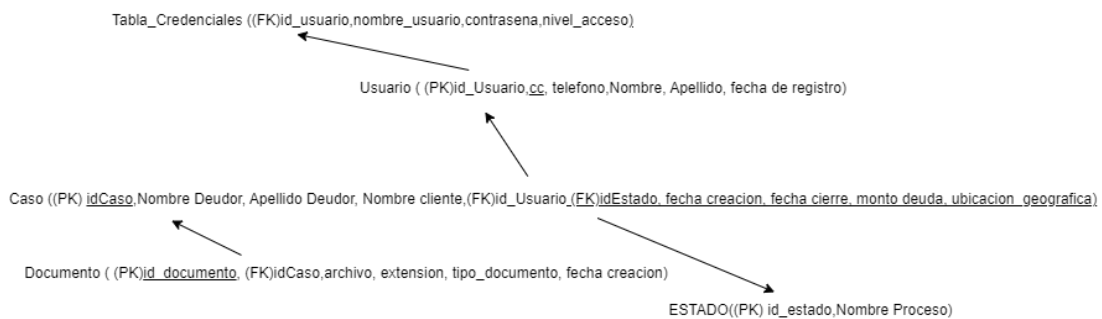


Figura No. 15 Modelo relacional
Fuente: Elaboración propia

```

CREATE TABLE Usuario (
  id_usuario SERIAL PRIMARY KEY,
  cc VARCHAR(20) NOT NULL,
  telefono VARCHAR(20) NOT NULL,
  nombre VARCHAR(50) NOT NULL,
  apellido VARCHAR(50) NOT NULL,
  fecha_registro TIMESTAMP NOT NULL
);

CREATE TABLE Estado (
  id_estado SERIAL PRIMARY KEY,
  nombre_proceso VARCHAR(50) NOT NULL
);

CREATE TABLE Caso (
  id_caso SERIAL PRIMARY KEY,
  nombre_deudor VARCHAR(50) NOT NULL,
  apellido_deudor VARCHAR(50) NOT NULL,
  cc_deudor VARCHAR(20) NOT NULL,
  nombre_empresa VARCHAR(50) NOT NULL,
  id_usuario INTEGER NOT NULL,
  id_estado INTEGER NOT NULL,
  fecha_creacion TIMESTAMP NOT NULL,
  fecha_cierre TIMESTAMP,
  monto_deuda DECIMAL(12,2),
  ubicacion_geografica VARCHAR(50) NOT NULL,
  etapa VARCHAR(50) NOT NULL,
  juzgado VARCHAR(50),
  numero_radicado VARCHAR(50),
  CONSTRAINT fk_caso_usuario FOREIGN KEY (id_usuario) REFERENCES Usuario (id_usuario),
  CONSTRAINT fk_caso_estado FOREIGN KEY (id_estado) REFERENCES Estado (id_estado)
);

CREATE TABLE Asignacion (
  id_asignacion SERIAL PRIMARY KEY,
  id_usuario INTEGER NOT NULL,
  id_caso INTEGER NOT NULL,
  CONSTRAINT fk_asignacion_usuario FOREIGN KEY (id_usuario) REFERENCES Usuario (id_usuario),
  CONSTRAINT fk_asignacion_caso FOREIGN KEY (id_caso) REFERENCES Caso (id_caso)
);

CREATE TABLE Documento (
  id_documento SERIAL PRIMARY KEY,
  id_caso INTEGER NOT NULL,
  ruta_archivo VARCHAR(150) NOT NULL,
  extension VARCHAR(10) NOT NULL,
  tipo_documento VARCHAR(50) NOT NULL,
  fecha_creacion TIMESTAMP NOT NULL,
  CONSTRAINT fk_documento_caso FOREIGN KEY (id_caso) REFERENCES Caso (id_caso)
);

CREATE TABLE Credenciales (
  id_usuario INTEGER PRIMARY KEY,
  nombre_usuario VARCHAR(50) NOT NULL,
  contraseña VARCHAR(255) NOT NULL,
  nivel_acceso INTEGER NOT NULL,
  CONSTRAINT fk_credenciales_usuario FOREIGN KEY (id_usuario) REFERENCES Usuario (id_usuario)
);

```

Figura 16 - Parte del script de la base de datos
Fuente: Elaboración propia

```

CREATE OR REPLACE FUNCTION registrar_trazabilidad()
RETURNS trigger AS
$$
DECLARE
  accion text;
  tabla_afectada text;
  id_afectado integer;
  usuario_afectado text;
  fecha_hora timestamp;
  descripcion text;
  col_name text;
  col_value text;
BEGIN
  tabla_afectada := TG_TABLE_NAME;
  fecha_hora := now();
  usuario_afectado := CURRENT_USER;

  IF (TG_OP = 'INSERT') THEN
    accion := 'insertó';
    id_afectado := NEW.id;
    descripcion := 'El registro con ID ' || id_afectado || ' de la tabla ' || tabla_afectada || ' fue insertado por el usuario ' || usuario_afectado || ' en ' || fecha_hora || '.';
    -- Agrupar valores de columnas a la descripción
    FOR col_name, col_value IN SELECT column_name, NEW.*::text FROM information_schema.columns WHERE table_name = tabla_afectada LOOP
      descripcion := descripcion || ' ' || col_name || ': ' || col_value || ',';
    END LOOP;
    -- Eliminar la última coma de la descripción
    descripcion := substr(descripcion, 1, length(descripcion) - 1);
  ELSEIF (TG_OP = 'UPDATE') THEN
    accion := 'actualizó';
    id_afectado := NEW.id;
    descripcion := 'El registro con ID ' || id_afectado || ' de la tabla ' || tabla_afectada || ' fue actualizado por el usuario ' || usuario_afectado || ' en ' || fecha_hora || ', el valor del campo ' || TG_ARGV[0] || ' fue actualizado de ' || OLD.*::text || ' a ' || NEW.*::text || '.';
  ELSEIF (TG_OP = 'DELETE') THEN
    accion := 'eliminó';
    id_afectado := OLD.id;
    descripcion := 'El registro con ID ' || id_afectado || ' de la tabla ' || tabla_afectada || ' fue eliminado por el usuario ' || usuario_afectado || ' en ' || fecha_hora || '.';
  END IF;

  INSERT INTO Trazabilidad (id_usuario, fecha_hora, tabla_afectada, tipo_accion, descripcion)
  VALUES (usuario_afectado, fecha_hora, tabla_afectada, accion, descripcion);

  RETURN NEW;
END;
$$
LANGUAGE plpgsql;

```

Figura 17 - Trigger Base de datos para la tabla trazabilidad
Fuente: Elaboración propia

K. Anexo No. 11: Actas Reuniones Director



FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA DE SISTEMAS
COMITÉ OPCIONES DE GRADO
FORMATO REUNIONES REALIMENTACIÓN PROPUESTAS DE GRADO
OPCIONES DESARROLLO TECNOLÓGICO Y PROYECTO DE GRADO

TÍTULO: DESARROLLO SOFTWARE PARA LA GESTIÓN DE CASOS EN EL DERECHO CIVIL

OPCIÓN DE GRADO: Desarrollo Tecnológico

ESTUDIANTES

Antonio José Mata Rodríguez código 1019151310

Maruan Enrique Arias Cucunuba código 1013688458

DIRECTOR: Carlos Delgado Román

REUNIÓN No. 1

1.	FECHA Y HORA
	10/02/2023 - 5:00 PM a 6:00 PM
2.	REALIMENTACIÓN PROPUESTA PRESENTADA
	Se indicó los pasos a seguir para el inicio del proyecto los cuales comprenden recomendaciones a la hora de realizar el diagrama BPMN del proceso de la empresa Grupo Montserrat, así como también el modelo BPSC que se solicita en "Documento Final Trabajos de Grado", donde se debe tener en cuenta el sistema que se va a desarrollar e implementar para la mejora del proceso.
3.	ACUERDOS Y CONCLUSIONES
	<ul style="list-style-type: none">- Diagrama BPMN del proceso de la empresa grupo montserrat- Adelanto del documento Documento Final Trabajos de Grado DT/PG con la información de la propuesta de grado.- Adelanto modelo BPSC

Fecha próxima reunión: 17 de febrero de 2023

Figura No. 18. Acta reunion director 10-02-2023

Fuente: Elaboración propia

FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA DE SISTEMAS
COMITÉ OPCIONES DE GRADO
FORMATO REUNIONES REALIMENTACIÓN PROPUESTAS DE GRADO
OPCIONES DESARROLLO TECNOLÓGICO Y PROYECTO DE GRADO

TÍTULO: DESARROLLO SOFTWARE PARA LA GESTIÓN DE CASOS EN EL DERECHO CIVIL
OPCIÓN DE GRADO: Desarrollo Tecnológico

ESTUDIANTES

Antonio José Mata Rodríguez código 1019151310
 Maruan Enrique Arias Cucunuba código 1013688458

DIRECTOR: Carlos Delgado Román

REUNIÓN NÚMERO 2

1.	FECHA Y HORA
	20/02/2023 - 3:00 PM a 4:00 PM
2.	REALIMENTACIÓN PROPUESTA PRESENTADA
	Se ilustró un modelo BPMN inicial sobre el proceso "As-Is" de la empresa Grupo Montserrat, a la vez de que se mostró los avances presentados en el documento final, ajustando la información que estaba presente en el documento de la propuesta inicial.
3.	ACUERDOS Y CONCLUSIONES
	<ul style="list-style-type: none"> - Completar diagrama BPMN del proceso de la empresa Grupo Montserrat en el "As-Is" y en el "To-Be". - Adelantar el documento final, revisar marco teórico e información que ya se tiene. - Adelantar modelo BPSC de acuerdo con la información suministrada por medio de entrevistas al Grupo Montserrat.

Figura No. 19. Acta reunion director 20-02-2023

Fuente: Elaboración propia

FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA DE SISTEMAS
COMITÉ OPCIONES DE GRADO
FORMATO REUNIONES REALIMENTACIÓN PROPUESTAS DE GRADO
OPCIONES DESARROLLO TECNOLÓGICO Y PROYECTO DE GRADO

TÍTULO: DESARROLLO SOFTWARE PARA LA GESTIÓN DE CASOS EN EL DERECHO CIVIL

OPCIÓN DE GRADO: Desarrollo Tecnológico

ESTUDIANTES

Antonio José Mata Rodríguez código 1019151310

Maruan Enrique Arias Cucunuba código 1013688458

DIRECTOR: Carlos Delgado Román

REUNIÓN NÚMERO 3

1.	FECHA Y HORA
	22/03/2023 - 2:00 PM a 3:00 PM
2.	REALIMENTACIÓN PROPUESTA PRESENTADA
	Se realizó una evaluación de los objetivos específicos, debido a que, en la retroalimentación realizada por el docente de proyecto de grado se sugirió realizar algunos ajustes a estos. Por tal motivo y con la guía del director de proyecto dentro de los objetivos específicos, se comenzaron a especificar el para qué de estos, así como, establecer los entregables de estos donde se determina para el objetivo #1, los requerimientos funcionales, en el caso del objetivo específico #2, un documento de arquitectura SAD, en el objetivo 3, el sistema de información, por último, en el objetivo #4 un documento de pruebas. Se presentaron los diagramas BPMN (AS y TO BE), el diagrama de arquitectura de alto nivel de AWS y el presupuesto de la aplicación.
3.	ACUERDOS Y CONCLUSIONES
	<ul style="list-style-type: none"> Realizar Diagrama SQL que contenga una tabla de trazabilidad que sirve para responder preguntas de negocio y realizar auditorías. Realizar cambios a los específicos.

Fecha próxima reunión: 1 de abril

Figura No. 20. Acta reunión director 22-03-2023

Fuente: Elaboración propia

FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA DE SISTEMAS
COMITÉ OPCIONES DE GRADO
FORMATO REUNIONES REALIMENTACIÓN PROPUESTAS DE GRADO
OPCIONES DESARROLLO TECNOLÓGICO Y PROYECTO DE GRADO

TÍTULO: DESARROLLO SOFTWARE PARA LA GESTIÓN DE CASOS EN EL DERECHO CIVIL

OPCIÓN DE GRADO: Desarrollo Tecnológico

ESTUDIANTES

Antonio José Mata Rodríguez código 1019151310

Maruan Enrique Arias Cucunuba código 1013688458

DIRECTOR: Carlos Delgado Román

REUNIÓN NÚMERO 4

1.	FECHA Y HORA
	01/04/2023 - 10:30 PM a 12:00 PM
2.	REALIMENTACIÓN PROPUESTA PRESENTADA
	Se presentó el diagrama SQL con la tabla de trazabilidad, el adelanto de los Mockups y el cambio realizado en los objetivos específicos. Para el diagrama SQL se sugiere que en la reunión que se establece con el cliente para el día 4 de abril se le consultará si quisieran agregar más datos a las tablas que le sean de utilidad. En los Mockups igualmente se sugirió el uso de filas o tablas para la visualización de la información de los casos, sin embargo, se recomienda realizar la retroalimentación con el cliente para validar si hay información faltante o de sobra para refinar el mockup. Por último, en los objetivos se establece dejarles los documentos entregables de cada uno.
3.	ACUERDOS Y CONCLUSIONES
	<ul style="list-style-type: none"> Realizar historias de usuario y que sean aprobadas por el cliente, consultarles por posibles atributos o información que requieran en las tablas. Compartir con el cliente el borrador de los mockups, con el fin de evaluar si la información a mostrar en el mismo es relevante, falta o no es necesaria, con base a lo anterior se puedan realizar refinamientos de los mockups.

Fecha próxima reunión: 10 de abril

Figura No. 21. Acta reunión director 01-04-2023

Fuente: Elaboración propia

L. Anexo No. 12: Actas Reuniones Grupo Montserrat



ACTA DE REUNIÓN GRUPO MONTSERRAT

TÍTULO: DESARROLLO SOFTWARE PARA LA GESTIÓN DE CASOS EN EL DERECHO CIVIL

ESTUDIANTES

Antonio José Mata Rodríguez código 1019151310

Maruan Enrique Arias Cucunuba código 1013688458

CLIENTES

Hernando Vélez

Yenifer Gutierrez

REUNIÓN NÚMERO 1

1.	FECHA Y HORA
	25/02/2023 - 11:00 AM a 12:00 PM
2.	REALIMENTACIÓN PROPUESTA PRESENTADA
	Se preparó una entrevista con el fin de conocer el contexto de la empresa Grupo Montserrat. Esto por medio de la identificación de actores del contexto, artefactos que utilizan, medios, creencias y hábitos. Teniendo en cuenta la repercusión de estos en las distintas dimensiones del modelo Biopsicosocial y cultural. Así mismo se profundizó en las causas y efectos de la problemática presentada. También se realizó la revisión de la primera versión del diagrama BPMN para verificar que se realizó un buen análisis del proceso de cobranzas jurídicas y determinar si faltaron agregar más procesos.
3.	ACUERDOS Y CONCLUSIONES
	<ul style="list-style-type: none">• Ajustar diagrama BPMN de acuerdo a las aclaraciones de los procesos jurídicos.

Fecha próxima reunión: 4 de Abril

Figura No. 22. Acta reunión Grupo Montserrat 25-02-2023

Fuente: Elaboración propia

ACTA DE REUNIÓN GRUPO MONTSERRAT

TÍTULO: DESARROLLO SOFTWARE PARA LA GESTIÓN DE CASOS EN EL DERECHO CIVIL

ESTUDIANTES

Antonio José Mata Rodríguez código 1019151310
Maruan Enrique Arias Cucunuba código 1013688458

CLIENTES

Hernando Vélez
Yenifer Gutierrez

REUNIÓN NÚMERO 1

1.	FECHA Y HORA
	04/04/2023 - 11:00 AM a 12:00 PM
2.	REALIMENTACIÓN PROPUESTA PRESENTADA
	Se preparó una presentación para introducir los temas trabajados por los estudiantes, donde se abarcan aspectos como los diagramas SQL, arquitectura AWS, costos del arrendamiento de espacios en AWS y avances en los Mockups. En los diagramas SQL se presentaron las tablas creadas, con la finalidad de preguntarle a los clientes si los atributos mostrados son los indicados y si hicieron falta algunos o estaban de más. Se explicó la arquitectura de la nube por medio del diagrama de arquitectura AWS, a la vez con sus respectivos costos. Por último, se mostró el borrador del mockup para tener una apreciación de los clientes, desde la perspectiva de aspectos visuales y funcionales.
3.	ACUERDOS Y CONCLUSIONES
	<ul style="list-style-type: none"> Se negoció el módulo de mensajería por un generador de informes. Se eliminó el módulo de "Resumen" en los mockups. Realizar módulo de la gestión del caso en el mockup. Se agregaron los siguientes atributos a la tabla <i>Caso</i>: "cc_deudor", "etapa", "juzgado", "numero_radicado" y "monto_obligacion_recuperado".

Fecha próxima reunión: 4 de Abril

Figura No. 23. Acta reunión Grupo Montserrat 04-04-2023

Fuente: Elaboración propia

M. Anexo No. 13: Configuraciones AWS

Summary			
DB identifier database-1gm	CPU -	Status ⊖ Stopped temporarily	Class db.t3.micro
Role Instance	Current activity	Engine PostgreSQL	Region & AZ us-east-2c

Figura No. 24. Configuración Base de datos PostgreSQL en AWS

Fuente: Elaboración propia

```
from dataclasses import dataclass
import psycopg2
from psycopg2.extras import RealDictCursor
import json

print(psycopg2.__version__)

host = "x"
username = "x"
password = "x"
database = "databaseAWSgrupoMontserrat"
try:
    conn = psycopg2.connect(
        host = host,
        database = database,
        user = username,
        password = password
    )
    print("Conexion exitosa")
    conn.set_client_encoding('UTF8')
except psycopg2.Error as e:
    print("Error al conectar a la base de datos:", e)

def lambda_handler(event, context):
    cur = conn.cursor(cursor_factory = RealDictCursor)
    cur.execute("select * from estado")
    results = cur.fetchall()
    json_result = json.dumps(results)
    print(json_result)
    return json_result
```

Figura No.25. Código Lambda Conexion Base datos y consulta tabla estado

Fuente: Elaboración propia



Figura No. 26. Ejecución exitosa en el servicio lambda
Fuente: Elaboración propia

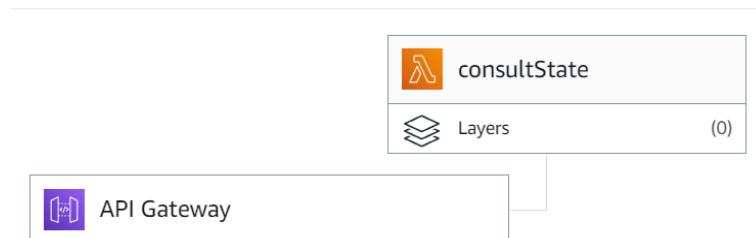


Figura No. 27. Conexión servicio lambda y APIGateway
Fuente: Elaboración propia

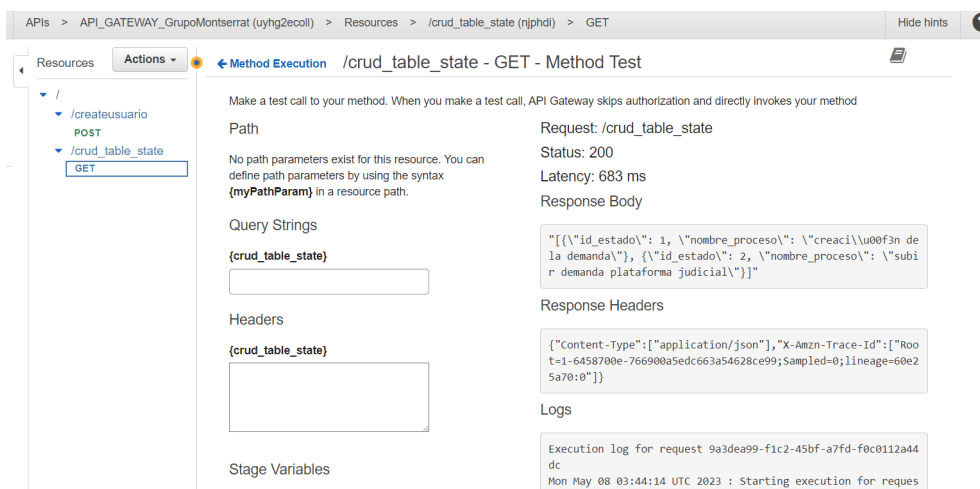


Figura No. 28. Configuración API Gateway con el servicio Lambda
Fuente: Elaboración propia

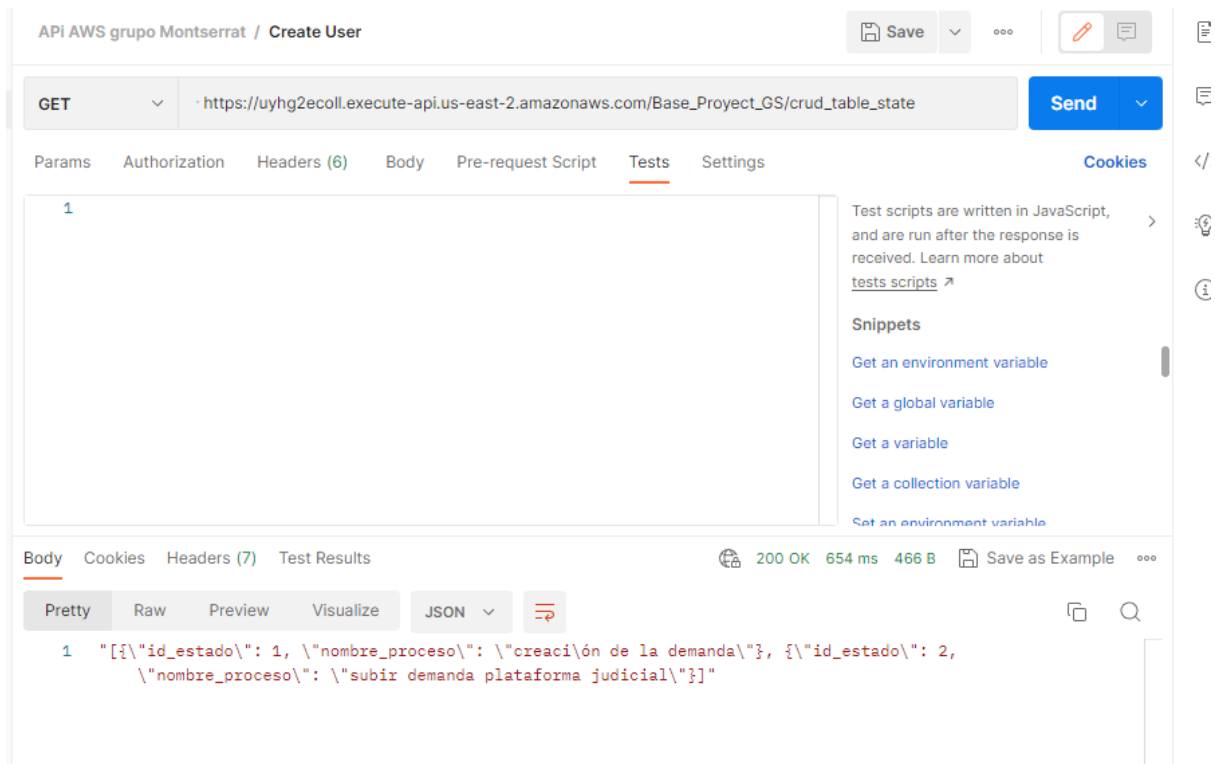


Figura No. 29. EndPoint Servicio Lambda en el API Gateway

Fuente: Elaboraci3n propia

<input type="checkbox"/>	Function name	Description	Package type	Runtime
<input type="checkbox"/>	updateState	-	Zip	Python 3.9
<input type="checkbox"/>	deleteState	-	Zip	Python 3.9
<input type="checkbox"/>	consultState	-	Zip	Python 3.9
<input type="checkbox"/>	insertState	-	Zip	Python 3.9

Figura No. 30. Configuraci3n funciones Lambda CRUD tabla estado

Fuente: Elaboraci3n propia

/crud_table_state Methods

DELETE arn:aws:lambda:us-east-2:870876887632:functi... Authorization None API Key Not required	GET arn:aws:lambda:us-east-2:870876887632:functi... Authorization None API Key Not required
POST arn:aws:lambda:us-east-2:870876887632:functi... Authorization None API Key Not required	PUT arn:aws:lambda:us-east-2:870876887632:functi... Authorization None API Key Not required

Figura No. 31 Configuración funciones Lambda CRUD tabla estado en APiGateway
Fuente: Elaboración propia

N. Anexo No. 14: Configuraciones CRUD tablas casó, usuario y credenciales AWS

/crud_table_case Methods	
<div> <div>DELETE</div> <div>arn:aws:lambda:us-east-2:870876887632:functi...</div> <div> <div>Authorization</div> <div>None</div> </div> <div> <div>API Key</div> <div>Not required</div> </div> </div>	<div> <div>GET</div> <div>arn:aws:lambda:us-east-2:870876887632:functi...</div> <div> <div>Authorization</div> <div>None</div> </div> <div> <div>API Key</div> <div>Not required</div> </div> </div>
<div> <div>OPTIONS</div> <div>Mock Endpoint</div> <div> <div>Authorization</div> <div>None</div> </div> <div> <div>API Key</div> <div>Not required</div> </div> </div>	<div> <div>POST</div> <div>arn:aws:lambda:us-east-2:870876887632:functi...</div> <div> <div>Authorization</div> <div>None</div> </div> <div> <div>API Key</div> <div>Not required</div> </div> </div>
<div> <div>PUT</div> <div>arn:aws:lambda:us-east-2:870876887632:functi...</div> <div> <div>Authorization</div> <div>None</div> </div> <div> <div>API Key</div> <div>Not required</div> </div> </div>	

Figura No. 32. Configuración funciones Lambda CRUD tabla caso en APiGateway
Fuente: Elaboración propia

/crud_table_users Methods	
<div> <div>DELETE</div> <div>arn:aws:lambda:us-east-2:870876887632:functi...</div> <div> <div>Authorization</div> <div>None</div> </div> <div> <div>API Key</div> <div>Not required</div> </div> </div>	<div> <div>GET</div> <div>arn:aws:lambda:us-east-2:870876887632:functi...</div> <div> <div>Authorization</div> <div>None</div> </div> <div> <div>API Key</div> <div>Not required</div> </div> </div>
<div> <div>OPTIONS</div> <div>Mock Endpoint</div> <div> <div>Authorization</div> <div>None</div> </div> <div> <div>API Key</div> <div>Not required</div> </div> </div>	<div> <div>POST</div> <div>arn:aws:lambda:us-east-2:870876887632:functi...</div> <div> <div>Authorization</div> <div>None</div> </div> <div> <div>API Key</div> <div>Not required</div> </div> </div>
<div> <div>PUT</div> <div>arn:aws:lambda:us-east-2:870876887632:functi...</div> <div> <div>Authorization</div> <div>None</div> </div> <div> <div>API Key</div> <div>Not required</div> </div> </div>	

Figura No. 33. Configuración funciones Lambda CRUD tabla usuario en APiGateway
Fuente: Elaboración propia

/crud_table_credentials Methods

POST	PUT
arn:aws:lambda:us-east-2:870876887632:functi...	arn:aws:lambda:us-east-2:870876887632:functi...
Authorization None	Authorization None
API Key Not required	API Key Not required

Figura No. 34. Configuración funciones Lambda CRUD tabla credenciales en APiGateway

Fuente: Elaboración propia

/crud_table_asignation Methods

DELETE	GET
arn:aws:lambda:us-east-2:870876887632:functi...	arn:aws:lambda:us-east-2:870876887632:functi...
Authorization None	Authorization None
API Key Not required	API Key Not required

OPTIONS	POST
Mock Endpoint	arn:aws:lambda:us-east-2:870876887632:functi...
Authorization None	Authorization None
API Key Not required	API Key Not required

PUT
arn:aws:lambda:us-east-2:870876887632:functi...
Authorization None
API Key Not required

Figura No. 35. Configuración funciones Lambda CRUD tabla asignación en APiGateway

Fuente: Elaboración propia

O. Anexo No. 15: Maquetación de los mockups en Angular y CRUD tablas tarjeta, documento y comentario

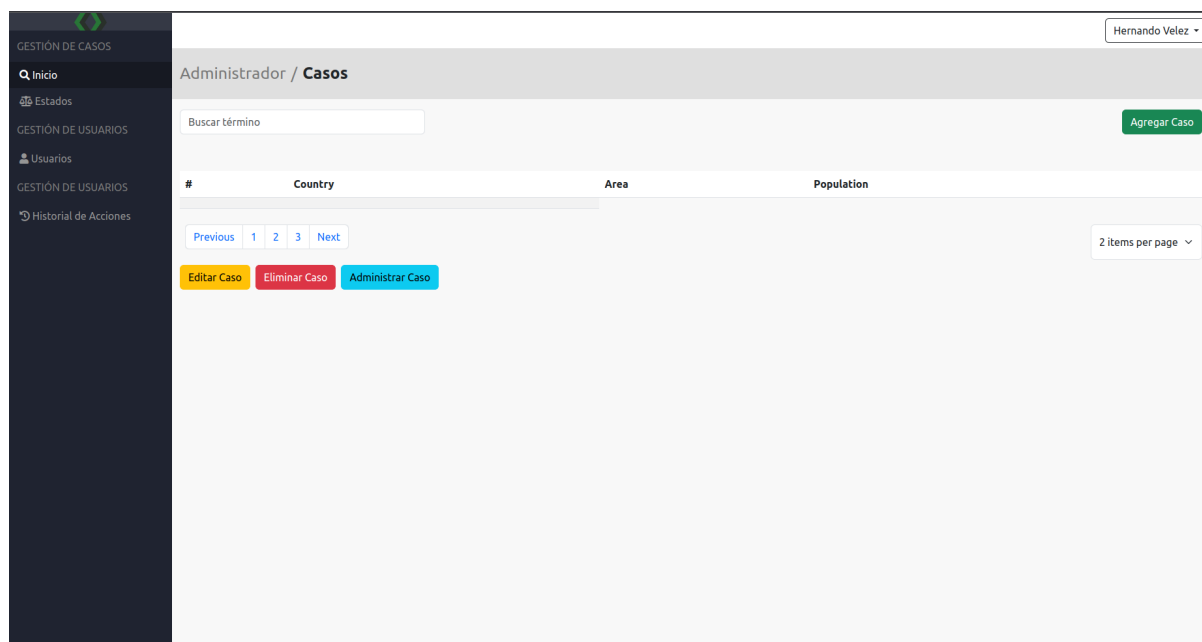


Figura No. 36. Maquetación vista "casos" en el usuario administrador

Fuente: Elaboración propia

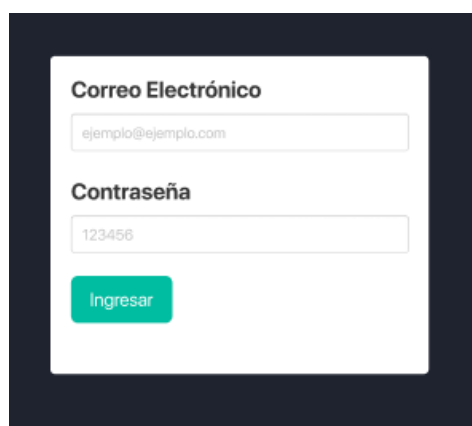


Figura No.37. Mockup Login

Fuente: Elaboración propia

GESTIÓN DE CASOS

[Inicio](#)

[Estados](#)

GESTIÓN DE USUARIOS

- [Usuarios](#)
- [Gestión de Usuarios](#)
- [Historial de Acciones](#)

Hernando Velez ▾

Administrador / Casos

Agregar Caso

ID	Nombre Deudor	Apellido Deudor	Cedula Deudor	Nombre Empresa	Razón Social	NIT	Monto Deuda	Ubicación	Estado	Etapas	# Radicado Demanda	Fecha Creación	Fecha Cierre	Acción
51	PEDRO	PEREZ	19454886	AVIANCA			\$10,000,000.00	Medellin	Juridico	creación de la demanda		Oct 16, 2023		Editar Caso Eliminar Caso Administrar Caso
34	HERNANDO editado2	VELEZ	19429890	MAPFRE	MAPFRE	07874747474	\$50,000,000.00	Bogotá D.C	Prejudicio	creación de la demanda	1221212	Oct 14, 2023	Dec 23, 2023	Editar Caso Eliminar Caso Administrar Caso
38	Mata333	Ramirez	10134567	empresa_ff	IBM_COLOMBIA-SAS	0	\$1,600,000.00	Bogotá	Juridico	creación de la demanda	33333555	Oct 15, 2023		Editar Caso Eliminar Caso Administrar Caso
40	Nombre Nuevo Deudor	Apellido Nuevo Deudor	11111122	Empresa-Nuevo			\$0.00	Ubicación Nuevo	Juridico	creación de la demanda	666666	Oct 15, 2023	Oct 8, 2023	Editar Caso Eliminar Caso Administrar Caso
11	Alberto	Cruz	1272728211	Propaganda S & S.	IBM_COLOMBIA-SAS	123456789	\$1,700,000.00	Bogotá	Judicial	creación de la demanda	13456121213333	Sep 13, 2023	Jul 2, 2023	Editar Caso

GESTIÓN DE CASOS

Inicio

Estados

GESTIÓN DE USUARIOS

Usuarios

GESTIÓN DE USUARIOS

Historial de Acciones

Administración

Nombre Deudor

34

HERNANDO editado2

52

Pedro Enrique

51

PEDRO

40

Nombre Nuevo Deudor

Agregar Caso

X

Área Cliente

Nombre de la Empresa (OBLIGATORIO)

Empresa S.A.S

Área Deudor

Nombre Deudor (OBLIGATORIO)

John Alexander

Cedula Deudor (OBLIGATORIO)

123456789

NIT (OPCIONAL)

123456789

Apellido Deudor (OBLIGATORIO)

Doe Smith

Razón Social (OPCIONAL)

Razón Social

Estado Actual (OBLIGATORIO)

Seleccione un Estado

Monto de la Deuda (OPCIONAL)

0

Ubicación Geográfica (OBLIGATORIO)

Bogotá D.C - Colombia

Etapas (OBLIGATORIO)

Prejuridico

Juzgado (OPCIONAL)

Juzgado 99

Numero Radicado Demanda (OPCIONAL)

12345678

Agregar Caso

Cerrar

P. Anexo No. 16: CRUD tabla tarjeta, comentarios y documentos

/crud_tarjeta Methods

DELETE arn:aws:lambda:us-east-2:870876887632:functi... Authorization None API Key Not required	GET arn:aws:lambda:us-east-2:870876887632:functi... Authorization None API Key Not required
OPTIONS Mock Endpoint Authorization None API Key Not required	POST arn:aws:lambda:us-east-2:870876887632:functi... Authorization None API Key Not required

Figura No. 42. Configuración funciones Lambda CRUD tabla tarjeta en APiGateway

Fuente: Elaboración propia

	id [PK] integer	id_caso integer	id_usuario integer	fecha_hora timestamp without time zone	id_estado integer
1	4	8	2	2023-07-10 01:42:44.177032	1
2	5	8	2	2023-09-11 01:36:33.075608	1
3	6	8	2	2023-09-11 01:45:21.313052	1

Figura No. 43. Cambios tabla Tarjeta

Fuente: Elaboración propia

/crud_table_comment Methods

DELETE arn:aws:lambda:us-east-2:870876887632:functi... Authorization None API Key Not required	GET arn:aws:lambda:us-east-2:870876887632:functi... Authorization None API Key Not required
OPTIONS Mock Endpoint Authorization None API Key Not required	POST arn:aws:lambda:us-east-2:870876887632:functi... Authorization None API Key Not required
PUT arn:aws:lambda:us-east-2:870876887632:functi... Authorization None API Key Not required	

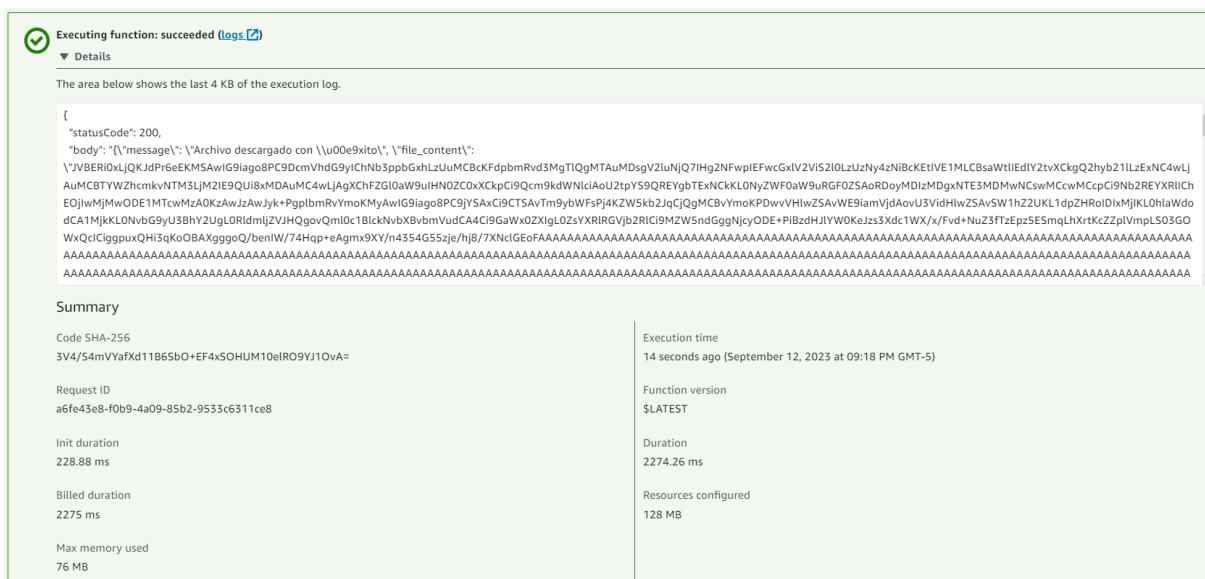
Figura No. 44. Configuración funciones Lambda CRUD tabla comentarios en APiGateway
Fuente: Elaboración propia

/downloadfiles Methods

OPTIONS Mock Endpoint Authorization None API Key Not required	POST arn:aws:lambda:us-east-2:870876887632:functi... Authorization None API Key Not required
--	---

Figura No. 45. Configuración funciones Lambda Download tabla Documento en APiGateway
Fuente: Elaboración propia

OPTIONS	POST
<p data-bbox="344 295 780 304">Mock Endpoint</p> <hr data-bbox="344 304 780 304"/> <p data-bbox="344 304 780 313">Authorization None</p> <p data-bbox="344 313 780 322">API Key Not required</p>	<p data-bbox="780 295 1230 304">arn:aws:lambda:us-east-2:870876887632:func...</p> <hr data-bbox="780 304 1230 304"/> <p data-bbox="780 304 1230 313">Authorization None</p> <p data-bbox="780 313 1230 322">API Key Not required</p>



Code source Info

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

Environment

downloadFiles / download.py

```

5
6
7 def lambda_handler(event, context):
8     # Inicializa el cliente de S3
9     s3 = boto3.client('s3')
10    bucket_name = 'testifrontangular'
11
12    # Recuperar el nombre del archivo de la petición HTTP
13    file_name = event.get('nombre_documento')
14
15    # Crear una variable para almacenar el contenido del archivo
16    file_content = None
17
18    try:
19        # Descarga el archivo desde S3
20        s3_response = s3.get_object(Bucket=bucket_name, Key=file_name)
21        file_content = s3_response['Body'].read()
22
23        encoded_file_content = base64.b64encode(file_content).decode("utf-8")
24    except Exception as e:
25        return {
26            'statusCode': 400,
27            'body': json.dumps(f'Error al descargar el archivo desde S3: {str(e)}')
28        }
29
30    # Continúa con el resto de tu código aquí
31    # ...
32
33    # Retorno del contenido del archivo
34    return {
35        'statusCode': 200,
36        'body': json.dumps({
37            'message': 'Archivo descargado con éxito',
38            'file_content': encoded_file_content
39        })
40    }
41

```

Figura No. 48. Código lambda de descargue del archivo
Fuente: Elaboración propia

✓ Executing function: succeeded ([logs](#))

▼ Details

The area below shows the last 4 KB of the execution log.

```

{
  "statusCode": 200,
  "body": "\"Archivo subido y ruta guardada en la base de datos.\""}

```

Summary

Code SHA-256	Execution time
V5tixwuGSm7sXKHldwbGsPbZmF3VsTHiv85LKUovfAA=	4 seconds ago (September 12, 2023 at 09:31 PM GMT-5)
Request ID	Function version
e5d5b0ac-63d3-4aec-b7db-2dcce47f22dc	\$LATEST
Init duration	Duration
379.24 ms	2351.18 ms
Billed duration	Resources configured
2352 ms	128 MB
Max memory used	
89 MB	

Figura No. 49. Test lambda de cargue del archivo
Fuente: Elaboración propia

```
def lambda_handler(event, context):
    # Obtener el nombre del archivo de la petición HTTP
    file_name = event.get('nombre_documento')

    # Obtener el archivo del cuerpo de la petición
    encoded_file = event['body'].get('file', '')
    decoded_file = base64.b64decode(encoded_file)

    # Subir el archivo a S3
    s3 = boto3.client('s3')
    bucket_name = 'test1frontangular'
    s3.put_object(Body=decoded_file, Bucket=bucket_name, Key=file_name)

    # Conexión a la base de datos

    cursor = conn.cursor()

    # Guardar la ruta en la base de datos
    ruta_s3 = f"s3://{bucket_name}/{file_name}"
    now = datetime.now()
    id_tarjeta= event.get('id_tarjeta')
    extension= event.get('extension')
    tipo_documento=event.get('tipo_documento')

    cursor.execute("""
INSERT INTO Documento (id_tarjeta, ruta_archivo, extension, tipo_documento, fecha_creacion)
VALUES (%s, %s, %s, %s, %s)
""", (id_tarjeta, ruta_s3, extension, tipo_documento, now))

    conn.commit()
    cursor.close()
    conn.close()

    return {
        'statusCode': 200,
        'body': json.dumps('Archivo subido y ruta guardada en la base de datos.')
    }
```

Figura No. 50. Código lambda de cargue del archivo

Fuente: Elaboración propia

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you must create an IAM policy.

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified
<input type="checkbox"/>	prueba1.pdf	pdf	August 27, 2023, 23:31:20 (UTC-05:00)

Figura No. 51. Archivos guardados en s3 bucket

Fuente: Elaboración propia

```
def lambda_handler(event, context):
    try:
        cur = conn.cursor()
        # Obtener los datos del evento (puedes ajustar esto según la estructura de tu evento)
        id_estado = event.get('id_estado')
        id_caso = event.get('id_caso')
        limit = event.get('limit')
        offset = event.get('offset')

        # Insertar los datos en la tabla "caso"
        cur.execute("""
            SELECT
                tarjeta.id AS tarjeta_id,
                to_char(tarjeta.fecha_hora, 'YYYY-MM-DD HH24:MI:SS') as tarjeta_fecha_hora,
                Usuario.nombre AS usuario_nombre,
                Usuario.apellido AS usuario_apellido,
                array_agg(DISTINCT comentarios.texto) AS comentarios_texto,
                array_agg(DISTINCT (regexp_split_to_array(Documento.ruta_archivo, '/'))[array_length(regexp_split_to_array(Documento.ruta_archivo, '/'), 1)]) AS
                array_agg(DISTINCT Documento.extension) AS documentos_extension
            FROM tarjeta
            LEFT JOIN comentarios ON tarjeta.id = comentarios.id_tarjeta
            LEFT JOIN Documento ON tarjeta.id = Documento.id_tarjeta
            INNER JOIN Usuario ON tarjeta.id_usuario = Usuario.id_usuario
            WHERE tarjeta.id_estado = %s AND tarjeta.id_caso = %s
            GROUP BY tarjeta.id, Usuario.nombre, Usuario.apellido
            LIMIT %s OFFSET %s
        """, (id_estado, id_caso, limit, offset))
        results = cur.fetchall()
        transformed_results = transform_results_to_dict(results)
        return transformed_results

    except psycopg2.Error as e:
        print("Error al insertar el registro:", e)
        conn.rollback() # Deshacer la transacción en caso de error
        return "Error al insertar el registro"
    except Exception as e:
        print("Otro error:", e)
        return "Error al insertar el registro"
```

Figura No. 52. Código lambda consulta Paginada
Fuente: Elaboración propia

Figura No. 53. Vista del módulo del seguimiento de casos
Fuente: Elaboración propia

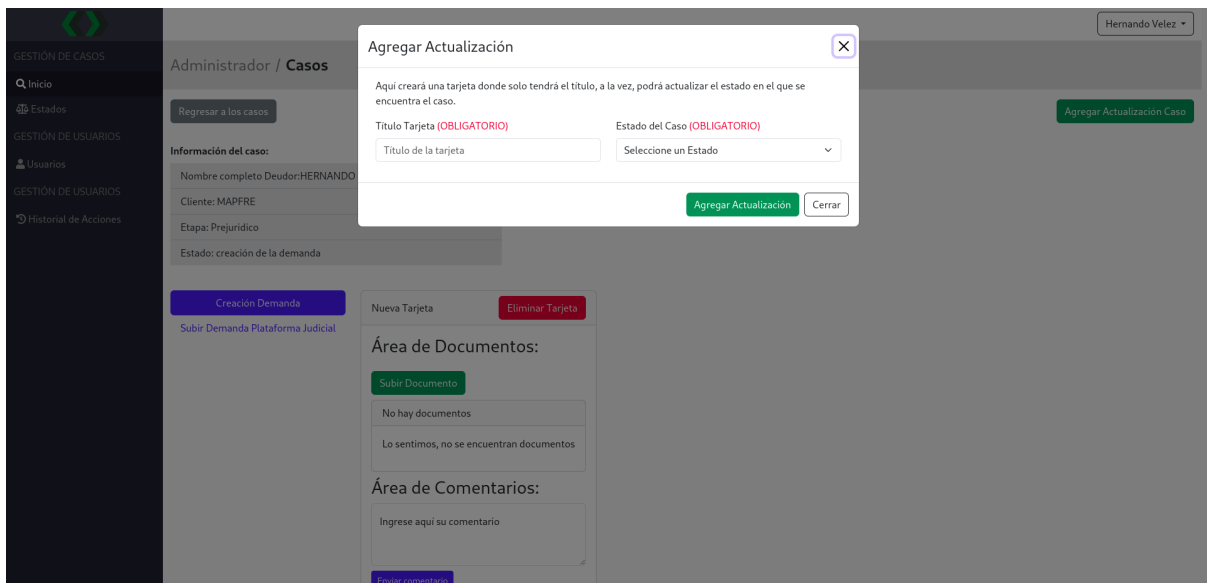


Figura No. 54. Cuadro agregar una nueva “tarjeta”
Fuente: Elaboración propia

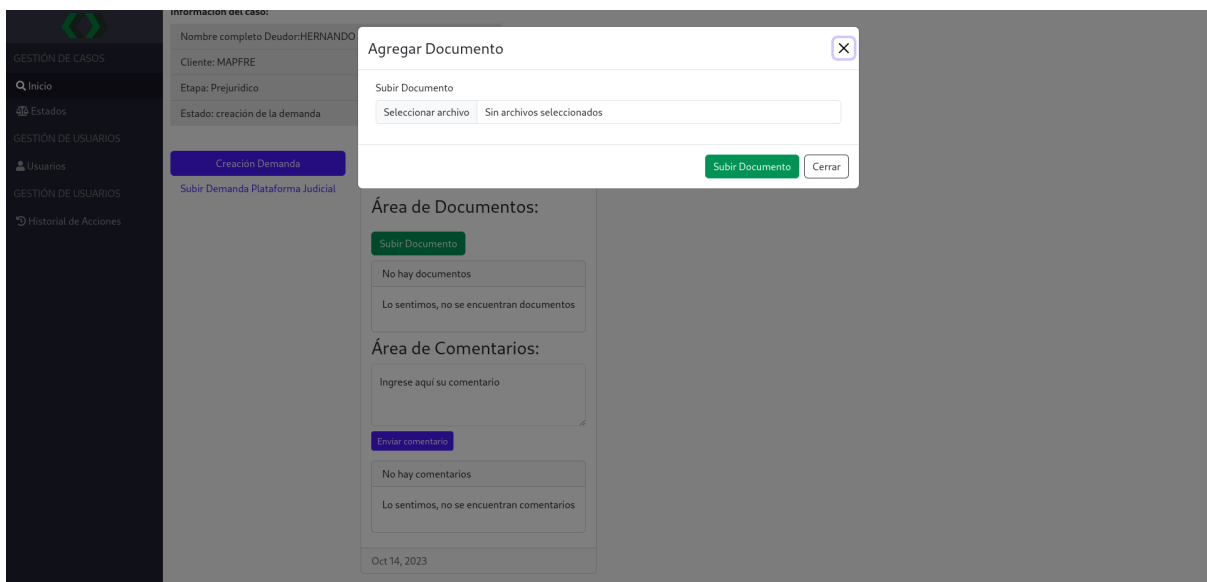


Figura No. 55. Cuadro agregar un nuevo documento asociado a una tarjeta
Fuente: Elaboración propia

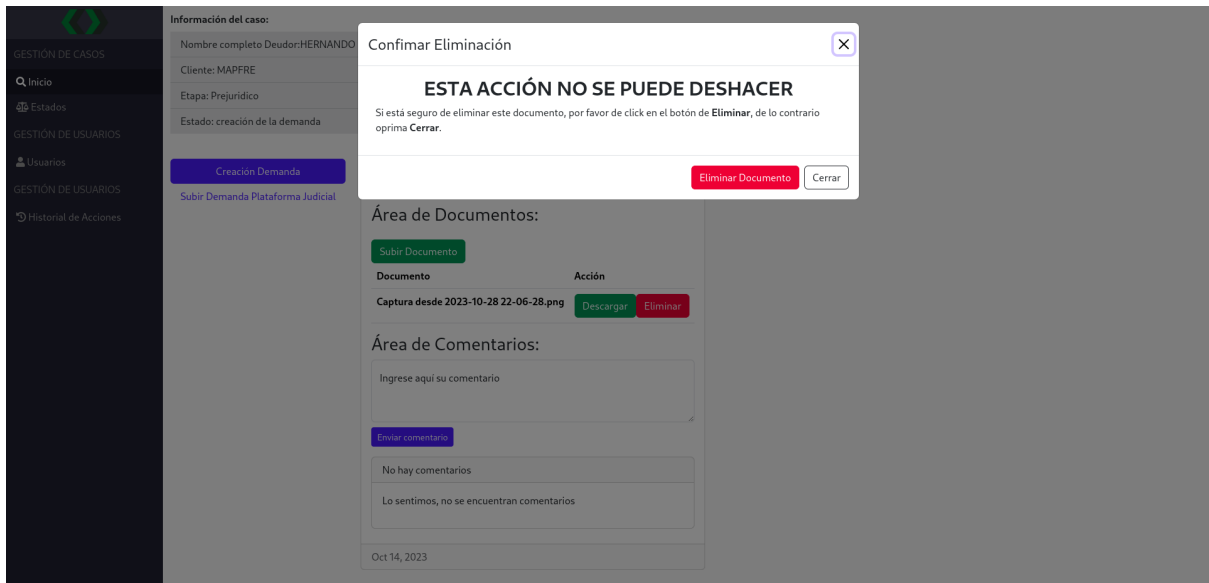


Figura No. 56. Cuadro eliminar un documento asociado a una tarjeta

Fuente: Elaboración propia

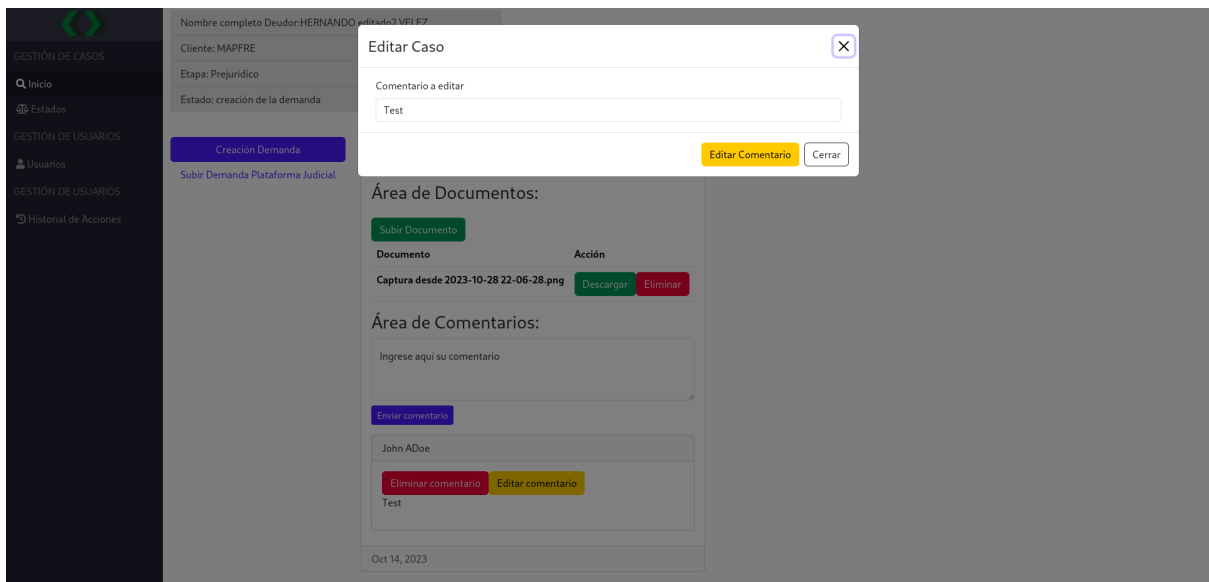


Figura No. 57. Cuadro editar un comentario asociado a una tarjeta

Fuente: Elaboración propia

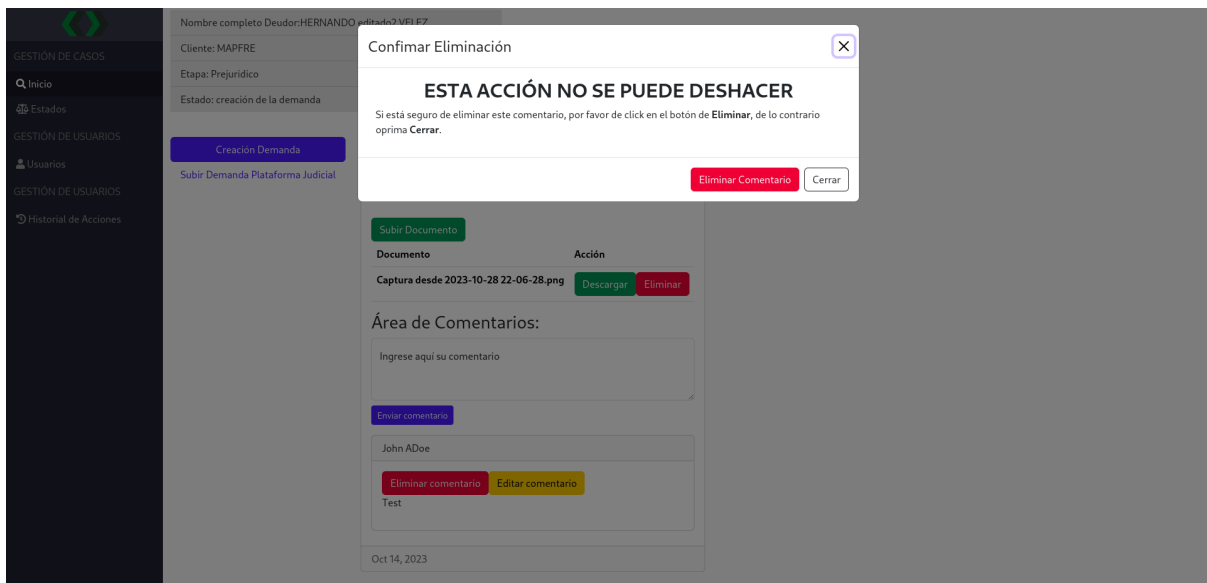


Figura No. 58. Cuadro eliminar un comentario asociado a una tarjeta
Fuente: Elaboración propia

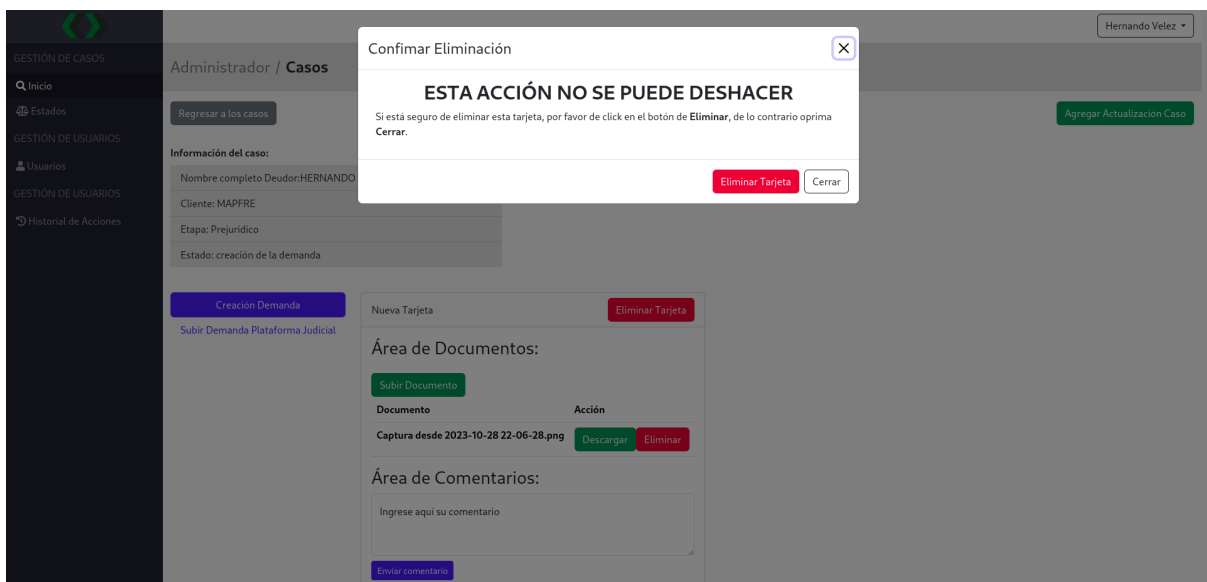


Figura No. 59. Cuadro eliminar una tarjeta ligada a un estado de un caso
Fuente: Elaboración propia