

Proyecto 1 Sistemas Transaccionales

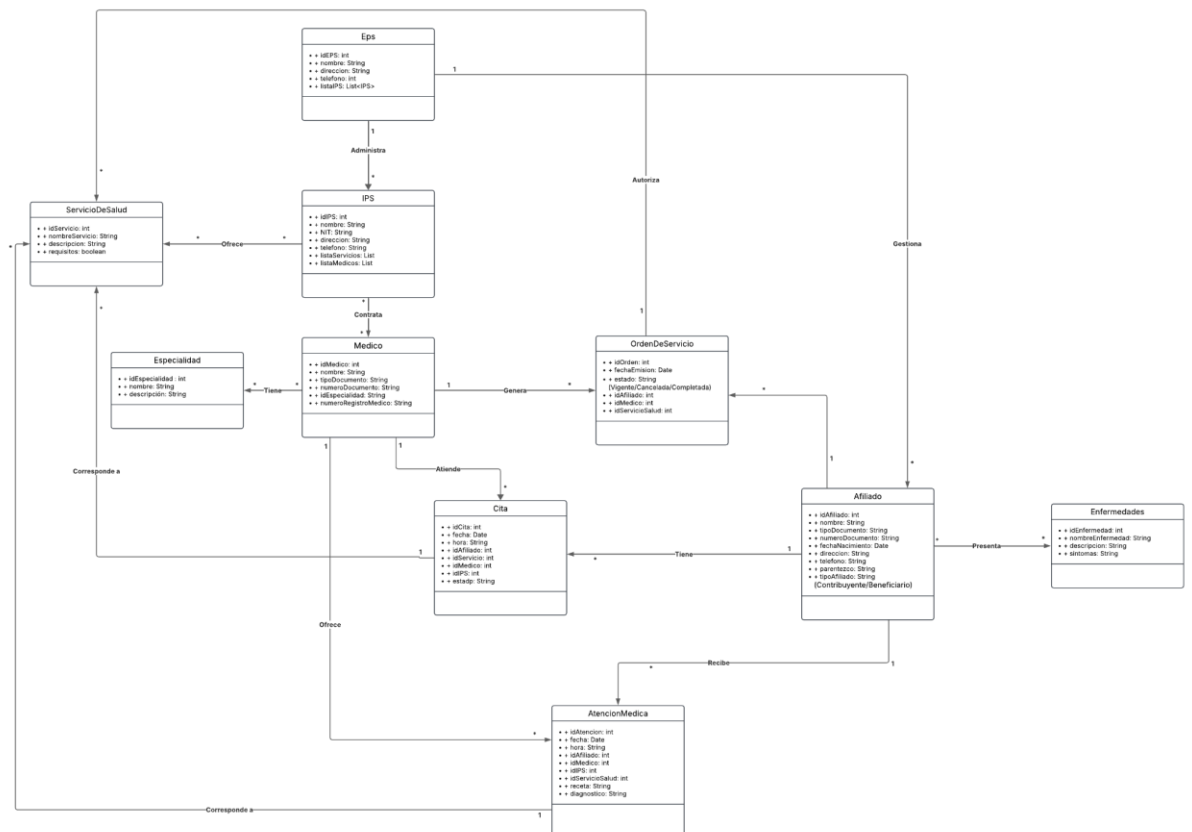
Grupo: Andrea Aroca, Paula Carreño, Santiago Delgado y Samuel Ovalle

a) Diagrama UML (todos):

➤ Actores (clases) principales del caso:

- **EPS:** Administra afiliados y contrata IPS para prestar servicios de salud.
- **Afiliado:** Persona inscrita en la EPS que usa sus servicios médicos
- **Enfermedades:** Problemas de salud que los afiliados pueden tener.
- **Médicos:** Profesionales que atienden pacientes y dan órdenes médicas
- **Especialidades:** Lista de especialidades posibles de cada médico
- **IPS:** Clínicas y hospitales que prestan servicios de salud
- **Orden Servicio:** Autorización de un médico para un tratamiento o consulta
- **Cita:** Fecha y hora programada para un servicio de salud.
- **Servicio Salud:** Consulta, examen o tratamiento ofrecido por una IPS.
- **Atención Médica:** Registro del momento en que un paciente recibe un servicio de salud

➤ Imagen UML:



Este diagrama UML representa el caso de EPSAndes, cada entidad está representada por una clase que contiene los atributos para describir su función dentro del sistema y las relaciones necesarias para conectar los procesos.

La clase **EPS** administra los servicios para sus afiliados y contratan a las IPS para la prestación de servicios médicos. Esta clase incluye atributos como su identificación, nombre y dirección. Por su parte, las **IPS** son los hospitales o clínicas que brindan los servicios de salud a los afiliados, contiene datos como su nombre, dirección y los médicos asociados, servicios. etc.

Los **Afiliados** son los usuarios del sistema que tienen un nombre, fecha de nacimiento y contacto, así como su tipo de afiliación (contributivo o beneficiario), si tiene un parentesco (familiar también registrado en la eps e identificación. Estos afiliados pueden estar relacionados con una o varias **Enfermedades**, las cuales están descritas en otra clase que incluye su identificador, nombre, descripción y síntomas.

El sistema también incluye a los **Médicos**, quienes son los responsables de atender a los afiliados y emitir órdenes de servicio. Esta clase tiene atributos como nombre, número de documento, número de registro médico para validar su licencia y la especialidad que poseen. Esta última está definida en la clase **Especialidad**, que describe las áreas médicas en las que los profesionales están capacitados para atender.

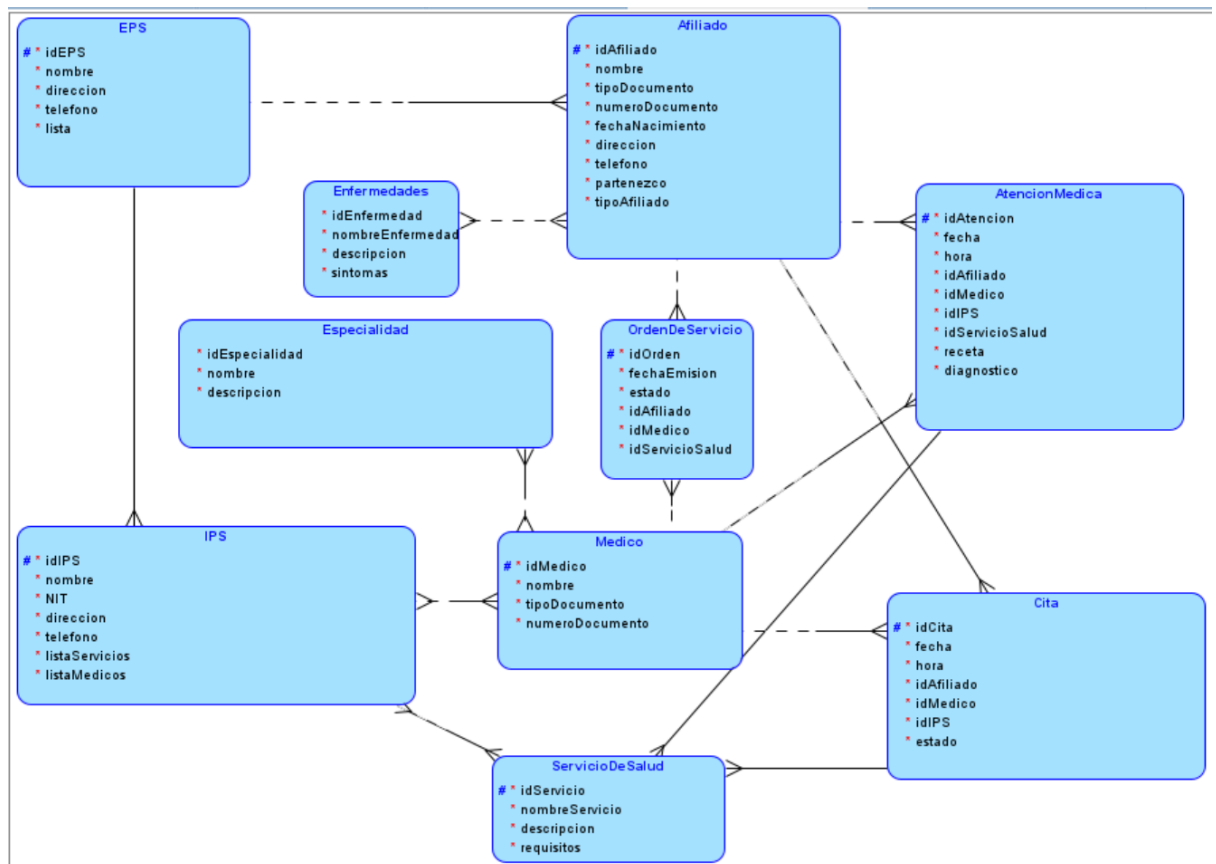
La clase **Servicio de Salud** detalla los diferentes tipos de consultas, exámenes o tratamientos ofrecidos por las IPS. Para acceder a estos servicios, los médicos emiten una **Orden de Servicio**, que sirve como autorización oficial para que un afiliado reciba el tratamiento o consulta necesarios. Estas órdenes contienen información como la fecha de emisión y el servicio al que están asociadas.

La clase **Citas** tiene atributos como la fecha y hora de la cita, el médico asignado y el servicio relacionado. Finalmente, la clase **Atención Médica** donde se registra el servicio prestado guardando información como el diagnóstico realizado y los resultados obtenidos durante la atención.

- Link diagrama UML: https://lucid.app/lucidchart/7d71d573-e84c-47ff-bdb8-db42fb33b3de/edit?viewport_loc=-753%2C-1326%2C4498%2C2008%2C0_0&invitationId=inv_445efc7a-928d-452b-9325-97d84bea8819

➤

b) Modelo E/R:



c) Modelo en Tablas

EPS			
idEPS (PK)	nombre	ubicación	telefono
PK, ND, NN	NN	NN	NN
1	Sanitas	Bogota	35678

EPS-IPS	
idEPS	idIPS
FK, NN	FK, NN
1	3
1	4
1	5

AFILIADO										
idAfiliado	nombre	tipoDocumento	numDocumento	fechaNacimiento	direccion	telefono	cedulaCotizante	tipoAfiliado	parentesco	idEPS
PK, ND, NN	NN	NN	NN, ND	NN	NN	NN	opcional	NN	opcional	FK, NN
1	juan	cedula	123456789	11/02/1980	Calle 116 # 20-30	3205673849	123456789	contribuyente		1
2	maria	cedula	987654321	9/05/2007	Calle 116 # 20-30	3115679840	123456789	Beneficiarios	Papá	1

Enfermedades			
idEnfermedad	nombreEnfermedad	descripcion	sintomas
PK, ND, NN	NN	opcional	opcional
3	Hipertencion	Persona con niveles altos de tensión	Dolor de cabeza, palidez
4	Apendicitis	Inflamación del apéndice, bolsa adherida al intestino grueso	Dolor al lado derecho del estomago

Presenta	
idAfiliado	idEnfermedad
PK, FK, NN	PK, FK, NN
1	3
1	4

IPS				
idIPS	nombre	NIT	direccion	telefono
PK, ND, NN	NN	NN, ND	NN	NN
1	Clinica del country	1234	Calle 100 #15-20	6173840

SERVICIO-SALUD			
idServicio	nombreServicio	descripcion	requisitos
PK, ND, NN	NN	opcional	NN
7	Consulta General	Atencion medica general	Cita previa obligatoria

SERVICIOS-IPS	
idIPS	idServicio
FK, NN	FK, NN
1	7

MEDICOS			
idMedico	nombre	tipoDocumento	numDocumento
PK, ND, NN	NN	NN	NN, ND
10	Dr. Martinez	CC	51479803

MEDICOS-IPS	
idMedico	idIPS
FK, NN	FK, NN
10	1

ESPECIALIDAD		
idEspecialidad	nombre	descripcion
PK, ND, NN	NN	opcional
1	Cardiologia	Especialidad dedicada al sistema cardiovascular

ESPECIALIDAD-MEDICO	
idMedico	idEspecialidad
FK, NN	FK, NN
10	1

CITA						
idCita	fecha	hora	idAfiliado	idMedico	idIPS	estado
PK, ND, NN	NN	NN	FK, NN	FK, NN	FK, NN	NN
101	17/02/2025	10:00	1	10	1	programada

ATENCIONMEDICA								
idAtencionMedica	idCita	fecha	hora	idAfiliado	idIPS	idServicio	receta	diagnostico
PK, ND, NN	FK, NN	NN	NN	FK, NN	FK, NN	FK, NN	opcional	NN
201	101	17/02/2025	10:00	1	1	10	Paracetamol 500mg	Gripa aguda

ORDENSERVICIO						
idOrden	fechaEmision	estado	idAfiliado	idMedico	idServicio	
PK, ND, NN	NN	NN	FK, NN	FK, NN	FK, NN	
301	17/02/2025	emitido	1	10	7	

d) Verificación del nivel de normalización de las tablas

1. Primera Forma Normal (1FN)

Para cumplir con la 1FN, cada atributo debe ser atómico, es decir, no debe contener listas ni conjuntos de valores. Todas las tablas en el modelo EPSANDES cumplen con esta condición, ya que cada atributo contiene un único valor y no existen atributos multivaluados.

2. Segunda Forma Normal (2FN)

Para cumplir con la 2FN, es necesario que cada atributo dependa completamente de la clave primaria y no existan dependencias parciales. En este modelo:

- Todas las relaciones con claves primarias simples cumplen con la 2FN.
- Las relaciones con claves compuestas han sido diseñadas correctamente para evitar dependencias parciales.
- Se garantiza que los atributos en cada tabla dependen únicamente de su clave primaria.

3. Tercera Forma Normal (3FN)

Para cumplir con la 3FN, no deben existir dependencias transitivas entre atributos. La mayoría de las tablas en EPSANDES cumplen con esta condición excepto OrdenDeServicio y Cita donde el estado de la orden podría depender transitivamente del afiliado y el servicio asignado.

4. Forma Normal de Boyce-Codd (BCNF)

Para que una tabla esté en BCNF, todas las dependencias funcionales deben ser determinadas por una superllave. Dado que la mayoría de las tablas en EPSANDES cumplen con la 3FN sin problemas adicionales, también cumplen con BCNF. La única excepción es OrdenDeServicio, donde se detectó una posible dependencia transitoria que puede requerir una descomposición adicional.

Cambios:

Cita: El campo estado depende transitivamente de idAfiliado y idServicio, no solo de la clave primaria idCita.

Problema: estado podría depender de idAfiliado y idServicio, generando una dependencia transitiva.
Solución: Separar el estado en una tabla aparte:

- CitaBase (idCita PK, fecha, hora, idAfiliado FK, idMedico FK, idIPS FK)
- EstadoCita (idCita PK, estado)

Nueva estructura:

CitaBase

idCita (PK)	fecha	hora	idAfiliado (FK)	idMedico (FK)	idIPS (FK)
101	17/02/2025	10:00	1	10	1

EstadoCita

idCita (PK, FK)	Estado
101	programada

OrdenServicio: El estado de la orden podría depender de idAfiliado y idServicio en lugar de la clave primaria completa.

Problema: El estado podría depender de idAfiliado y idServicio, en lugar de la clave primaria completa.

Solución: Descomponer la relación:

- OrdenBase (idOrden PK, fechaEmision, idAfiliado FK, idMedico FK, idServicio FK)
- EstadoOrden (idOrden PK, estado)

Nueva estructura:

OrdenBase

idOrden (PK)	fechaEmision	idAfiliado (FK)	idMedico (FK)	idServicio(FK)
301	17/02/2025	1	10	7

EstadoOrden

idOrden (PK, FK)	estado
301	emitido

e) Escenarios de prueba (Santiago → Sábado/Domingo):

1. Pruebas de Funcionalidad

RF1 - Registrar IPS

Escenario:

- **Caso exitoso:** Se registra una IPS con un nombre único, NIT válido y dirección.
- **Caso fallido:** Intentar registrar una IPS con un NIT ya existente (debe fallar).

RF2 - Registrar un Servicio de Salud

Escenario:

- **Caso exitoso:** Se registra un servicio de salud con una descripción válida.
- **Caso fallido:** Intentar registrar un servicio sin descripción (debe fallar).

RF3 - Asignar un Servicio de Salud a una IPS

Escenario:

- **Caso exitoso:** Se asigna un servicio de salud a una IPS existente.
- **Caso fallido:** Intentar asignar un servicio a una IPS que no existe (debe fallar).

RF4 - Registrar Médico

Escenario:

- **Caso exitoso:** Se registra un médico con una especialidad válida.
- **Caso fallido:** Intentar registrar un médico sin número de registro (debe fallar).

RF5 - Registrar Afiliado

Escenario:

- **Caso exitoso:** Se registra un afiliado con un tipo de documento válido.
- **Caso fallido:** Intentar registrar un afiliado con un número de documento repetido (debe fallar).

RF6 - Registrar una Orden de Servicio de Salud

Escenario:

- **Caso exitoso:** Se registra una orden de servicio para un afiliado con un médico existente.
- **Caso fallido:** Intentar registrar una orden con un servicio que no existe (debe fallar).

RF7 - Agendar un Servicio de Salud

Escenario:

- **Caso exitoso:** Se agenda un servicio en una fecha y hora disponibles.
- **Caso fallido:** Intentar agendar un servicio en una fecha sin disponibilidad (debe fallar).

RF8 - Registrar la Prestación de un Servicio de Salud

Escenario:

- **Caso exitoso:** Se registra una atención médica realizada a un afiliado.
- **Caso fallido:** Intentar registrar una prestación sin médico asociado (debe fallar).

2. Pruebas de Integridad en la Base de Datos

Pruebas de unicidad de tuplas

- Insertar una tupla con una clave primaria válida.
- Intentar insertar otra tupla con la misma clave primaria (debe fallar).

Pruebas de integridad referencial

- Insertar una orden de servicio con un afiliado y médico válidos (debe ser exitoso).
- Intentar insertar una orden de servicio con un afiliado que no existe (debe fallar).

Pruebas de restricciones de chequeo

- Intentar insertar una cita médica en una fecha fuera del horario permitido (debe fallar).
 - Intentar insertar un afiliado menor de edad como contribuyente (debe fallar).
-

3. Pruebas de Consultas y Estadísticas

RFC1 - Consultar la Agenda de Disponibilidad

Escenario:

- **Caso exitoso:** Se consulta la disponibilidad de un servicio en las próximas 4 **semanas**.
- **Caso fallido:** Intentar consultar disponibilidad de un servicio inexistente (debe fallar).

RFC2 - Mostrar los 20 Servicios más Solicitados

Escenario:

- **Caso exitoso:** Se muestra el ranking de los 20 servicios más solicitados en un periodo.
- **Caso fallido:** Intentar consultar sin definir un rango de fechas válido (debe fallar).

RFC3 - Mostrar el Índice de Uso de los Servicios

Escenario:

- **Caso exitoso:** Se calcula el índice de uso correctamente.
- **Caso fallido:** Intentar calcular el índice con un servicio sin registros (debe mostrar 0).

RFC4 - Mostrar Utilización de Servicios por un Afiliado

Escenario:

- **Caso exitoso:** Se consulta correctamente el historial de servicios de un afiliado.
- **Caso fallido:** Intentar consultar sin especificar un afiliado (debe fallar).

Entrega 2

1. Análisis y revisión del modelo de datos:

2. Creación de la base de datos utilizando SQL Developer:

- - **ID_IPS_MEDICO:** Identificador local (aunque no lo incluye en la PK).
 - **ID_IPS, ID_EPS, ID_MEDICO, ID_SERVICIO:** Llave primaria compuesta.
 - **Falta** la definición de llaves foráneas que relacionen con `IPS (ID_IPS)`, `EPS (ID_EPS)`, `MEDICO (ID_MEDICO)`, `SERVICIO_SALUD (ID_SERVICIO)`.
-

En conjunto:

- Estas tablas abarcan:
 - **EPS** e **IPS** (una EPS puede tener varias IPS).
 - **Afiliados** y sus **enfermedades** (afiliado-enfermedad es una relación N-N).
 - **Especialidades** y **Médicos** (1-N).
 - **Servicios de salud** (tabla `SERVICIO_SALUD`) asociables a IPS y/o médicos.
 - **Órdenes de servicio** y su estado.
 - **Citas** y su estado.
 - **Atención médica** resultante de la cita.
 - Tablas auxiliares como **AGENDA**, **ESPECIALIDAD_SERVICIO**, **IPS_MEDICO**.

Este esquema describe un **sistema de salud** con la posibilidad de gestionar:

- Quién es el afiliado y a qué EPS pertenece.
- Cuáles IPS hay (cada IPS puede asociarse a una EPS).
- Qué médicos trabajan en cada IPS (posiblemente en `IPS_MEDICO`).
- Qué servicios se ofrecen y qué médicos los atienden.

- Órdenes generadas por un médico para un afiliado con un servicio.
- Citas agendadas con fecha/hora y estado.
- Atención final a la cita (diagnóstico, resultados).

3. Aplicación usando Java Spring:

4. Sentencias SQL para los requerimientos de consulta (RFC1 a RFC4):

a. Documentación las tablas usadas:

1. RFC1 - CONSULTAR LA AGENDA DE DISPONIBILIDAD QUE TIENE UN SERVICIO DE SALUD INGRESADO POR EL USUARIO EN LAS SIGUIENTES 4 SEMANAS.

Tablas usadas:

IPS_MEDICO: Relaciona médicos con IPS y servicios.

AGENDA: Contiene los horarios disponibles y asignados para los médicos.

SERVICIO_SALUD: Define los servicios de salud disponibles.

IPS: Contiene información de las Instituciones Prestadoras de Salud.

MEDICO: Almacena los datos de los médicos.

Joins:

IPS_MEDICO.ID_IPS_MEDICO = AGENDA.ID_IPS_MEDICO: Para relacionar los horarios de agenda con los médicos y servicios.

IPS_MEDICO.ID_SERVICIO = SERVICIO_SALUD.ID_SERVICIO: Para filtrar por un servicio específico.

IPS_MEDICO.ID_IPS = IPS.ID_IPS: Para obtener el nombre de la IPS.

IPS_MEDICO.ID_MEDICO = MEDICO.ID_MEDICO: Para obtener el nombre del médico.

Explicación del join:

Se usó un INNER JOIN implícito (en el WHERE) porque solo nos interesan los registros que tienen coincidencias en todas las tablas. Esto asegura que solo se muestren las citas disponibles para el servicio específico (ID 13) en el rango de fechas dado.

Propósito:

Esta consulta muestra las citas disponibles para un servicio específico (en este caso, Dermatología, ID 13) en un rango de fechas, incluyendo la fecha, hora, nombre de la IPS y nombre del médico.

Resultados obtenidos:

Dermatología | 01/03/24 07:00 | Clínica Santa Fe | Dra. Laura Torres

Dermatología | 01/03/24 08:00 | Clínica Santa Fe | Dra. Laura Torres

Dermatología | 01/03/24 08:30 | Clínica Santa Fe | Dra. Laura Torres

Dermatología | 01/03/24 15:30 | Clínica Santa Fe | Dra. Laura Torres

Dermatología | 05/03/24 13:30 | Clínica Santa Fe | Dra. Laura Torres

Dermatología | 15/03/24 09:30 | Clínica Santa Fe | Dra. Laura Torres

Dermatología | 20/03/24 14:30 | Clínica Santa Fe | Dra. Laura Torres

Dermatología | 29/03/24 16:30 | Clínica Santa Fe | Dra. Laura Torres

Total de filas seleccionadas: 8.

2. RFC2 - MOSTRAR LOS 20 SERVICIOS MÁS SOLICITADOS. Los que fueron más solicitados en un período de tiempo dado:

Tablas usadas:

ORDEN_BASE: Contiene las órdenes médicas emitidas.

IPS_MEDICO: Relaciona médicos con IPS y servicios.

IPS: Contiene información de las Instituciones Prestadoras de Salud.

EPS: Almacena los datos de las EPS.

MEDICO: Contiene los datos de los médicos.

SERVICIO_SALUD: Define los servicios de salud.

Joins:

IPS_MEDICO.ID_IPS = IPS.ID_IPS: Para obtener el nombre de la IPS.

IPS_MEDICO.ID_EPS = EPS.ID_EPS: Para obtener el nombre de la EPS.

IPS_MEDICO.ID_MEDICO = MEDICO.ID_MEDICO: Para obtener el nombre del médico.

IPS_MEDICO.ID_SERVICIO = SERVICIO_SALUD.ID_SERVICIO: Para obtener la descripción del servicio.

Explicación del join:

Se usó un INNER JOIN implícito (en el WHERE) porque solo nos interesan los registros que tienen coincidencias en todas las tablas. Esto asegura que solo se muestren los detalles de la IPS específica (ID 104) y sus servicios asociados.

Propósito:

La primera parte de la consulta cuenta el número total de órdenes médicas por servicio en un rango de fechas.

La segunda parte muestra los detalles de la IPS con ID 104, incluyendo el nombre de la IPS, la EPS a la que pertenece, el nombre del médico y el servicio que ofrece.

Resultados obtenidos:

Primera parte (órdenes médicas por servicio):

ID_SERVICIO | TOTAL_ORDENES

6 | 1

7 | 1

2 | 1

8 | 1

9 | 1

4 | 1

5 | 1

10 | 1

3 | 1

11 | 1

Total de filas seleccionadas: 10.

Segunda parte (detalles de la IPS 104):

ID_IPS | NOMBRE_IPS | NOMBRE_EPS | NOMBRE_MEDICO | DESCRIPCION

104 | Clínica Santa Fe | EPS SANITAS | Dra. Laura Torres | Dermatología

104 | Clínica Santa Fe | EPS SANITAS | Dra. Camila Suárez | Urgencias

3. RFC3 - MOSTRAR EL ÍNDICE DE USO DE CADA UNO DE LOS SERVICIOS PROVISTOS

Tablas usadas:

SERVICIO_SALUD: Define los servicios de salud disponibles.

CITA_BASE: Contiene las citas médicas programadas.

Joins:

SERVICIO_SALUD.ID_SERVICIO = CITA_BASE.ID_SERVICIO: Para relacionar las citas con los servicios.

Explicación del join:

Se usó un LEFT JOIN porque queremos incluir todos los servicios disponibles, incluso si no han sido usados en el período de tiempo especificado. Esto asegura que se muestren todos los servicios, no solo aquellos con citas programadas.

Propósito:

Esta consulta calcula el índice de uso de cada servicio en un período de tiempo específico. El índice se obtiene dividiendo el número de servicios usados (citas) entre el total de servicios disponibles.

ID_SERVICIO | SERVICIO | SERVICIOS_USADOS | SERVICIOS_DISPONIBLES | INDICE_USO

1	Consulta general	0	15	0
2	Urgencias	0	15	0
3	Hospitalización	0	15	0
4	Cirugía ambulatoria	0	15	0
5	Laboratorio clínico	0	15	0
6	Radiología e imágenes	0	15	0
7	Fisioterapia	0	15	0
8	Psicología	1	15	0.07
9	Odontología general	0	15	0
10	Medicina interna	0	15	0
11	Pediatría	1	15	0.07
12	Ginecología y obstetricia	1	15	0.07
13	Dermatología	0	15	0
14	Cardiología	0	15	0
15	Oftalmología	0	15	0

Total de filas seleccionadas: 15.

4. RFC4 - MOSTRAR LA UTILIZACIÓN DE SERVICIOS DE EPSANDES POR UN AFILIADO DADO, EN UN PERIODO DADO (RANGO DE FECHAS INDICADO).

Tablas usadas:

CITA_BASE: Contiene las citas médicas programadas.

AFILIADO: Almacena los datos de los afiliados.

MEDICO: Contiene los datos de los médicos.

IPS: Contiene información de las Instituciones Prestadoras de Salud.

Joins:

CITA_BASE.ID_AFILIADO = AFILIADO.ID_AFILIADO: Para obtener el nombre del afiliado.

CITA_BASE.ID_MEDICO = MEDICO.ID_MEDICO: Para obtener el nombre del médico.

CITA_BASE.ID_IPS = IPS.ID_IPS: Para obtener el nombre de la IPS.

Explicación del join:

Se usó un INNER JOIN implícito (en el WHERE) porque solo nos interesan los registros que tienen coincidencias en todas las tablas. Esto asegura que solo se muestren las citas del afiliado específico (ID 2019) en el rango de fechas dado.

Propósito:

Esta consulta muestra todas las citas de un afiliado específico (ID 2019) en un rango de fechas, incluyendo la fecha de la cita, el nombre del médico, el nombre de la IPS y el número total de citas.

Resultados obtenidos:

ID_AFILIADO	NOMBRE_AFILIADO	FECHA_CITA	NOMBRE_MEDICO	NOMBRE_IPS	TOTAL_CITAS
-------------	-----------------	------------	---------------	------------	-------------

2019	Santiago López	08/03/24	Dr. Daniel Pineda	Clínica Universitaria	1
------	----------------	----------	-------------------	-----------------------	---

5. Población de Tablas:

Se realizó la población de las tablas de la base de datos con información de prueba para simular el funcionamiento del sistema en un entorno real. Lo que se hizo en cada tabla:

- En la tabla AFILIADO, se insertaron datos de personas afiliadas a una EPS, incluyendo sus nombres, fechas de nacimiento, tipo de afiliación (contributivo o beneficiario) y la EPS a la que pertenecen.
- En la tabla AFILIADO_ENFERMEDAD, se relacionaron algunos afiliados con enfermedades específicas, indicando la fecha en que fueron diagnosticados.

- En la tabla ATENCION_MEDICA, se agregaron detalles de atenciones médicas realizadas, como diagnósticos y resultados.
- En la tabla CITA_BASE, se insertaron datos de citas médicas programadas, incluyendo la fecha, hora, el afiliado, el médico, la IPS y el servicio a prestar.
- En la tabla ENFERMEDAD, se incluyó una lista de enfermedades comunes con sus síntomas y descripciones.
- En la tabla EPS, se agregaron datos de EPS, como su nombre y dirección de ejemplo.
- En la tabla ESPECIALIDAD, se insertaron las especialidades médicas disponibles, como cardiología, neurología, entre otras.
- En la tabla ESTADO_CITA, se definió el estado de algunas citas (pendientes o completadas).
- En la tabla ESTADO_ORDEN, se indicó el estado de las órdenes médicas (pendientes, canceladas o atendidas).
- En la tabla IPS, se incluyó información de instituciones prestadoras de salud (IPS), con su nombre, dirección y la EPS a la que están asociadas.
- En la tabla IPS_SERVICIO, se relacionaron las IPS con los servicios de salud que ofrecen.
- En la tabla MEDICO, se insertaron datos de médicos, incluyendo su nombre, registro médico y especialidad.
- En la tabla ORDEN_BASE, se agregaron detalles de órdenes médicas, con la fecha de emisión, el afiliado, el médico y el servicio solicitado.
- En la tabla SERVICIO_SALUD, se incluyó una lista de servicios de salud, como consultas generales, urgencias, hospitalización, entre otros.
- En la tabla ESPECIALIDAD_SERVICIO, se relacionaron las especialidades médicas con los servicios que pueden ofrecer.
- En la tabla IPS_MEDICO, se vincularon los médicos con las IPS en las que trabajan y los servicios que brindan.
- Finalmente, en la tabla AGENDA, se insertaron horarios disponibles para 2 médicos en específico y en una ips para un mes.

En resumen, se simuló un sistema de salud completo con datos de prueba, permitiendo visualizar cómo interactúan las diferentes entidades.

6. Escenarios de prueba:

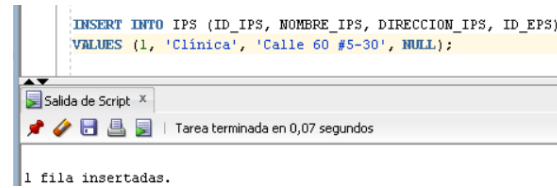
1. Pruebas de Unicidad de Tuplas

a. Caso Exitoso: Insertar una IPS con un NIT único

- Sentencia SQL:

```
INSERT INTO IPS (ID_IPS, NOMBRE_IPS, DIRECCION_IPS, ID_EPS)
```

```
VALUES (1, 'Clínica', 'Calle 60 #5-30', NULL);
```



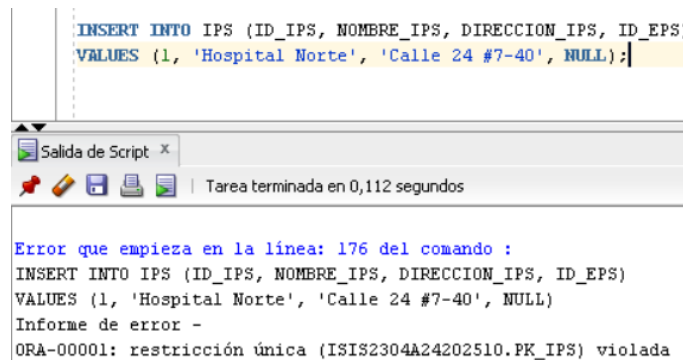
b. Caso Fallido: Insertar una IPS con un NIT ya existente

- Sentencia SQL:

```
INSERT INTO IPS (ID_IPS, NOMBRE_IPS, DIRECCION_IPS, ID_EPS)
```

```
VALUES (1, 'Hospital Norte', 'Calle 24 #7-40', NULL);
```

- Error Esperado: Violación de restricción de unicidad en NIT



2. Pruebas de Integridad

a. Caso Exitoso: Insertar un médico con una especialidad válida

- Sentencia SQL:

```
INSERT INTO ESPECIALIDAD (ID_ESPECIALIDAD, NOMBRE_ESPECIALIDAD)
```

```
VALUES (1, 'Cardiología');
```

```
INSERT INTO MEDICO (ID_MEDICO, NOMBRE_MEDICO, REGISTRO_MEDICO, ID_ESPECIALIDAD)
```

```
VALUES (1, 'Juan', 'MED-1234', 1);
```



```
INSERT INTO ESPECIALIDAD (ID_ESPECIALIDAD, NOMBRE_ESPECIALIDAD)
VALUES (1, 'Cardiología');

INSERT INTO MEDICO (ID_MEDICO, NOMBRE_MEDICO, REGISTRO_MEDICO, ID_ESPECIALIDAD)
VALUES (1, 'Juan', 'MED-1234', 1);
```

Salida de Script x

Tarea terminada en 0,113 segundos

Error que empieza en la línea: 219 del comando :

```
INSERT INTO ESPECIALIDAD (ID_ESPECIALIDAD, NOMBRE_ESPECIALIDAD)
VALUES (1, 'Cardiología')
```

Informe de error -

ORA-00001: restricción única (ISIS2304A24202510.PK_ESPECIALIDAD) violada

<https://docs.oracle.com/error-help/db/ora-00001/>

More Details :

<https://docs.oracle.com/error-help/db/ora-00001/>

Error que empieza en la línea: 222 del comando :

```
INSERT INTO MEDICO (ID_MEDICO, NOMBRE_MEDICO, REGISTRO_MEDICO, ID_ESPECIALIDAD)
VALUES (1, 'Juan', 'MED-1234', 1)
```

Informe de error -

ORA-00001: restricción única (ISIS2304A24202510.PK_MEDICO) violada

b. Caso Fallido:

- Sentencia SQL: Insertar un médico con especialidad no existente

```
INSERT INTO MEDICO (ID_MEDICO, NOMBRE_MEDICO,
REGISTRO_MEDICO, ID_ESPECIALIDAD)
```

```
VALUES (2, 'Diana', 'MED-5678', 5);
```

- Error esperado: Violación de clave foránea en idEspecialidad

```
INSERT INTO MEDICO (ID_MEDICO, NOMBRE_MEDICO, REGISTRO_MEDICO, ID_ESPECIALIDAD)
VALUES (2, 'Diana', 'MED-5678', 5);
```

Salida de Script x

Tarea terminada en 0,114 segundos

Error que empieza en la línea: 225 del comando :

```
INSERT INTO MEDICO (ID_MEDICO, NOMBRE_MEDICO, REGISTRO_MEDICO, ID_ESPECIALIDAD)
VALUES (2, 'Diana', 'MED-5678', 5)
```

Informe de error -

ORA-00001: restricción única (ISIS2304A24202510.PK_MEDICO) violada

3. Pruebas de restricción de chequeo

- ### a. Caso Exitoso: Insertar un afiliado con fecha de nacimiento válida

- Sentencia SQL:

```
INSERT INTO AFILIADO (ID_AFILIADO,
NOMBRE_AFILIADO, FECHA_NACIMIENTO,
TIPO_AFILIACION)
```

```
VALUES (1, 'Andres Perez', DATE '2003-05-10', 'Contributivo');
```

```
INSERT INTO AFILIADO (ID_AFILIADO, NOMBRE_AFILIADO, FECHA_NACIMIENTO, TIPO_AFILIACION)
VALUES (1, 'Andres Perez', DATE '2003-05-10', 'Contributivo');
```

1 fila insertadas.

b. Caso Fallido: Insertar un afiliado con fecha de nacimiento inválida

■ Sentencia SQL:

```
INSERT INTO AFILIADO (ID_AFILIADO,
NOMBRE_AFILIADO, FECHA_NACIMIENTO,
TIPO_AFILIACION)
```

```
VALUES (2, 'Pedrito', DATE '2028-09-20', 'Contributivo');
```

■ Error esperado: La fecha de nacimiento debe ser anterior a la actual.

```
INSERT INTO AFILIADO (ID_AFILIADO, NOMBRE_AFILIADO, FECHA_NACIMIENTO, TIPO_AFILIACION)
VALUES (2, 'Pedrito', DATE '2028-09-20', 'Contributivo');
```

1 fila insertadas.

4. Pruebas de funcionalidad:

a. Requerimiento Funcional 1 - Registrar IPS

■ Sentencia SQL:

```
INSERT INTO IPS (ID_IPS, NOMBRE_IPS, DIRECCION_IPS, ID_EPS)
```

```
VALUES (3, 'Centro Médico 1', 'Calle 60a #10-20', NULL);
```

```
INSERT INTO IPS (ID_IPS, NOMBRE_IPS, DIRECCION_IPS, ID_EPS)
VALUES (3, 'Centro Médico 1', 'Calle 60a #10-20', NULL);
```

1 fila insertadas.

b. Requerimiento Funcional 2 - Registrar Servicio de Salud

■ Sentencia SQL:

```
INSERT INTO SERVICIO_SALUD (ID_SERVICIO,
DESCRIPCION)
```

```
VALUES (1, 'Consulta médica básica');
```

```
INSERT INTO SERVICIO_SALUD (ID_SERVICIO, DESCRIPCION)
VALUES (1, 'Consulta médica básica');
```

Salida de Script x

Tarea terminada en 0,194 segundos

1 fila insertadas.

c. Requerimiento Funcional 3 - Asignar un Servicio de Salud a IPS

- Sentencia SQL:

```
INSERT INTO IPS_SERVICIO (ID_IPS, ID_SERVICIO) VALUES
(3, 1);
```

```
INSERT INTO IPS_SERVICIO (ID_IPS, ID_SERVICIO) VALUES (3, 1);
```

1 fila insertadas.

d. Requerimiento Funcional 4 - Registrar Médico

- Sentencia SQL:

```
INSERT INTO MEDICO (ID_MEDICO, NOMBRE_MEDICO,
REGISTRO_MEDICO, ID_ESPECIALIDAD)
VALUES (3, 'Camilo Sanchez', 'MED-1234', 1);
```

```
INSERT INTO MEDICO (ID_MEDICO, NOMBRE_MEDICO, REGISTRO_MEDICO, ID_ESPECIALIDAD)
VALUES (3, 'Camilo Sanchez', 'MED-1234', 1);
```

Salida de Script x

Tarea terminada en 0,184 segundos

Error que empieza en la línea: 242 del comando :

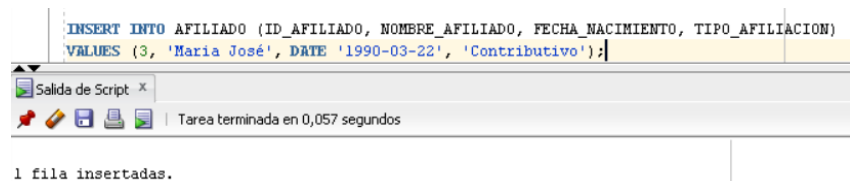
```
INSERT INTO MEDICO (ID_MEDICO, NOMBRE_MEDICO, REGISTRO_MEDICO, ID_ESPECIALIDAD)
VALUES (3, 'Camilo Sanchez', 'MED-1234', 1)
Informe de error -
ORA-00001: restricción única (ISIS2304A24202510.PK_MEDICO) violada
```

e. Requerimiento Funcional 5 - Registrar un Afiliado

- Sentencia SQL:

```
INSERT INTO AFILIADO (ID_AFILIADO,
NOMBRE_AFILIADO, FECHA_NACIMIENTO,
TIPO_AFILIACION)
```

VALUES (3, 'Maria José', DATE '1990-03-22', 'Contributivo');



```
INSERT INTO AFILIADO (ID_AFILIADO, NOMBRE_AFILIADO, FECHA_NACIMIENTO, TIPO_AFILIACION)
VALUES (3, 'Maria José', DATE '1990-03-22', 'Contributivo');
```

Salida de Script x

Tarea terminada en 0,057 segundos

1 fila insertadas.

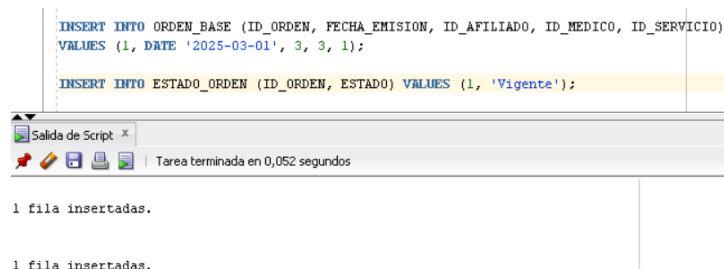
f. Requerimiento Funcional 6 - Registrar una orden de servicio de salud para un afiliado por parte de un médico

■ Sentencia SQL:

```
INSERT INTO ORDEN_BASE (ID_ORDEN, FECHA_EMISION,
ID_AFILIADO, ID_MEDICO, ID_SERVICIO)
```

VALUES (1, DATE '2025-03-01', 3, 3, 1);

```
INSERT INTO ESTADO_ORDEN (ID_ORDEN, ESTADO)
VALUES (1, 'Vigente');
```



```
INSERT INTO ORDEN_BASE (ID_ORDEN, FECHA_EMISION, ID_AFILIADO, ID_MEDICO, ID_SERVICIO)
VALUES (1, DATE '2025-03-01', 3, 3, 1);

INSERT INTO ESTADO_ORDEN (ID_ORDEN, ESTADO) VALUES (1, 'Vigente');
```

Salida de Script x

Tarea terminada en 0,052 segundos

1 fila insertadas.

1 fila insertadas.

g. Requerimiento Funcional 7 - Agendar un servicio de salud por parte de un afiliado

■ Sentencia SQL:

```
INSERT INTO CITA_BASE (ID_CITA, FECHA_CITA,
HORA_CITA, ID_AFILIADO, ID_MEDICO, ID_IPS,
ID_SERVICIO)
```

VALUES (1, DATE '2025-03-10', '10:00', 3, 3, 3, 1);

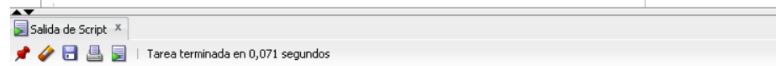
```
INSERT INTO ESTADO_CITA (ID_CITA, ESTADO) VALUES (1,
'Agendada');
```

```

INSERT INTO CITA_BASE (ID_CITA, FECHA_CITA, HORA_CITA, ID_AFILIADO, ID_MEDICO, ID_IPS, ID_SERVICIO)
VALUES (1, DATE '2025-03-10', '10:00', 3, 3, 3, 1);

INSERT INTO ESTADO_CITA (ID_CITA, ESTADO) VALUES (1, 'Agendada');

```



1 fila insertadas.

1 fila insertadas.

h. Requerimiento Funcional 8 - Registrar la prestación de un servicio de salud a un afiliado por parte de una IPS

- Sentencia SQL:

```

INSERT INTO ATENCION_MEDICA (ID_ATENCION, ID_CITA, DIAGNOSTICO, RESULTADOS)

```

```

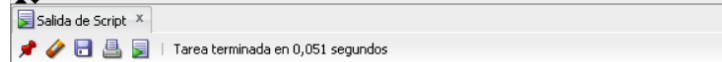
VALUES (1, 1, 'Resfriado', 'Recetado Paracetamol');

```

```

INSERT INTO ATENCION_MEDICA (ID_ATENCION, ID_CITA, DIAGNOSTICO, RESULTADOS)
VALUES (1, 1, 'Resfriado', 'Recetado Paracetamol');

```



1 fila insertadas.

