

Proyecto Sistemas Transaccionales

Laura Guiza 201920926

Alberto Pertuz 202025856

Angie Gutierrez 202223607

Uniandes

Entrega 1 – Diseño

2025-10

<b>Proyecto Sistemas Transaccionales .....</b>	<b>4</b>
Análisis y modelo conceptual .....	4
Modelo UML .....	4
Modelo conceptual en E/R.....	7
Diseño de la base de datos .....	10
Modelo de datos relacional .....	10
Nivel de normalización del modelo .....	15
Escenarios de prueba.....	20
REQUERIMIENTOS FUNCIONALES.....	20
RF1 – Registrar IPS.....	20
RF2 – Registrar un servicio .....	21
RF3 – Asignar un servicio de salud a una IPS .....	21
RF4 – Registrar médico .....	21
RF5 – Registrar afiliado.....	22
RF6 – Registrar una orden de servicio de salud para un afiliado por parte de un médico.....	22
RF7 – Agendar un servicio de salud por parte de un afiliado .....	23
RF8 – Registrar la prestación de un servicio de salud a un afiliado por parte de una IPS.....	24
REQUERIMIENTOS FUNCIONALES DE CONSULTA.....	24
RFC1 – Consultar la agenda de disponibilidad de un servicio de salud en las siguientes 4 semanas .....	24
RFC2 – Mostrar los 20 servicios más solicitados en un período de tiempo dado	25

RFC3 – Mostrar el índice de uso de cada uno de los servicios provistos .....	25
RFC4 – Mostrar la utilización de servicios de EPSAndes por un afiliado en un período dado.....	26

## Proyecto Sistemas Transaccionales

En el presente, se realiza el diseño de EPSAndes, una aplicación que apoya a las Entidades Promotoras de Salud en su operación diaria.

### Análisis y modelo conceptual

#### Modelo UML

El siguiente es el modelo UML

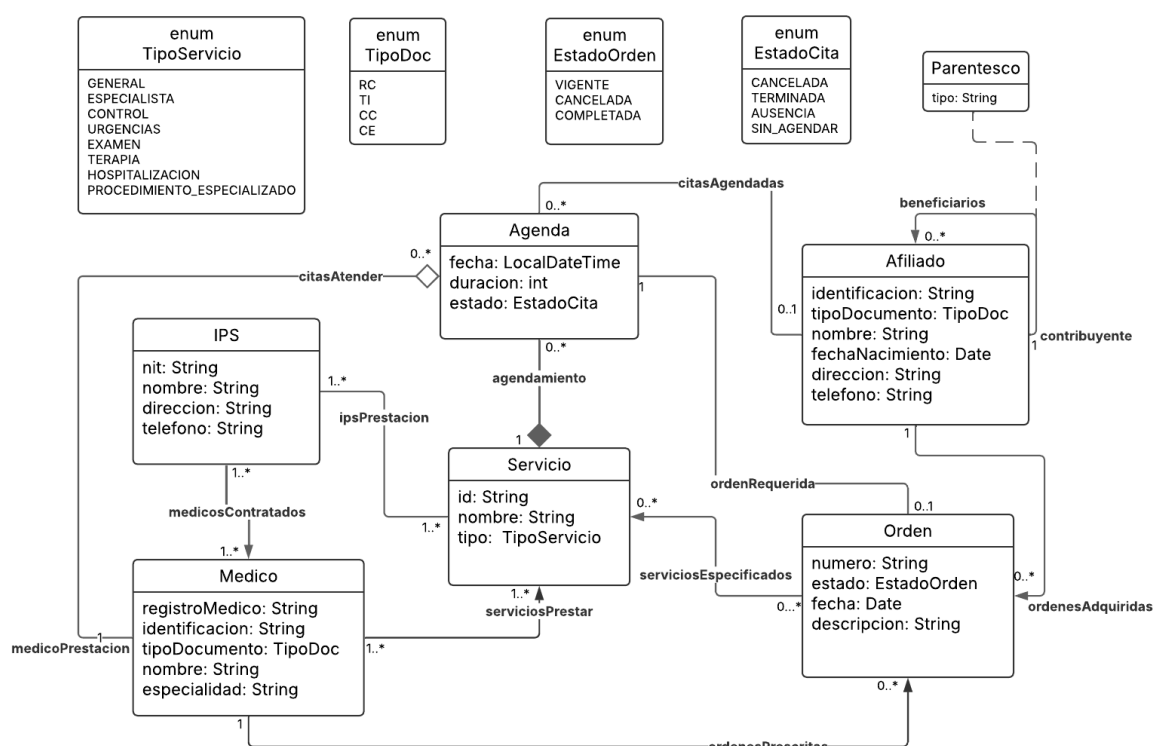


Ilustración 1. Modelo UML EPSANDES

#### 1. Entidades principales y sus relaciones

1.1. Afiliado • Representa a una persona inscrita en la EPS, que puede ser contribuyente (cotizante) o beneficiario. • Atributos: identificacion: Identificador único. tipoDocumento: Tipo de documento de identidad (RC, TI, CC, CE). nombre, fechaNacimiento, direccion, telefono. • Relaciones: Puede tener beneficiarios asociados (a

través de la relación con Parentesco). Recibe órdenes médicas (Orden). Puede tener cero o una cita agendada en la Agenda. Relación Afiliado - Agenda (0..1) Un afiliado puede tener una cita activa en la agenda, pero no es obligatorio. No todos los afiliados tienen citas pendientes, pero cuando una es agendada, debe asociarse a una Agenda específica.

1.2. IPS (Institución Prestadora de Salud) • Representa una institución que ofrece servicios médicos (hospitales, clínicas, centros de diagnóstico). • Atributos: nit: Identificación única. nombre, direccion, telefono. • Relaciones: Contrata múltiples médicos (Medico). Ofrece múltiples servicios (Servicio). Posee agendas de disponibilidad para la prestación de servicios. Relación IPS - Agenda (1..\*) Una IPS posee múltiples Agendas para organizar la disponibilidad de servicios. Sin embargo, una Agenda pertenece a una única IPS.

1.3. Médico • Representa un profesional de la salud adscrito a una o varias IPS. • Atributos: registroMedico: Identificación única del médico. identificacion, tipoDocumento, nombre. especialidad: Área de la medicina en la que está certificado. • Relaciones: Puede trabajar en varias IPS. Prescribe órdenes médicas (Orden). No está directamente ligado a la Agenda. Importante: Los médicos no son parte de la Agenda Si un médico deja de trabajar en una IPS, la agenda de citas sigue existiendo, ya que la disponibilidad pertenece a la IPS y no a un médico específico.

1.4. Orden Representa la prescripción de un servicio de salud. • Atributos: numero: Identificación única de la orden. estado: (VIGENTE, CANCELADA, COMPLETADA). fecha, descripcion. • Relaciones: Es emitida por un médico (Médico). Es asignada a un afiliado (Afiliado). Incluye uno o más servicios (Servicio). Requiere agendamiento en una IPS. Relación Orden - Médico (1..\*) Una Orden siempre es prescrita por un médico, pero un médico puede generar múltiples órdenes a diferentes afiliados.

1.5. Servicio • Representa un tipo de procedimiento médico. • Atributos: id: Identificación única. nombre, tipo: Tipo de servicio (GENERAL, ESPECIALISTA, CONTROL, URGENCIAS, EXAMEN, TERAPIA, HOSPITALIZACIÓN, PROCEDIMIENTO\_ESPECIALIZADO). • Relaciones: Es requerido en una orden médica. Es prestado por IPS específicas. Tiene una agenda de disponibilidad. Relación Servicio - IPS (1..\*) Cada servicio debe ser prestado por al menos una IPS.

1.6. Agenda • Representa la programación de disponibilidad de un servicio de salud en una IPS. • Atributos: fecha: Fecha y hora de la cita. duracion: Duración del servicio en minutos. estado: (CANCELADA, TERMINADA, AUSENCIA, SIN\_AGENDAR). • Relaciones: Vincula un servicio con una IPS. Se usa para programar citas médicas. Composición entre Agenda y Servicio Una Agenda no puede existir sin un Servicio, ya que su única función es indicar la disponibilidad del mismo.

1.7. Parentesco • Representa la relación entre un afiliado contribuyente y sus beneficiarios. • Atributo: tipo: Relación (Hijo, Padre, Cónyuge, etc.). • Relaciones: Define la relación entre afiliados y beneficiarios.

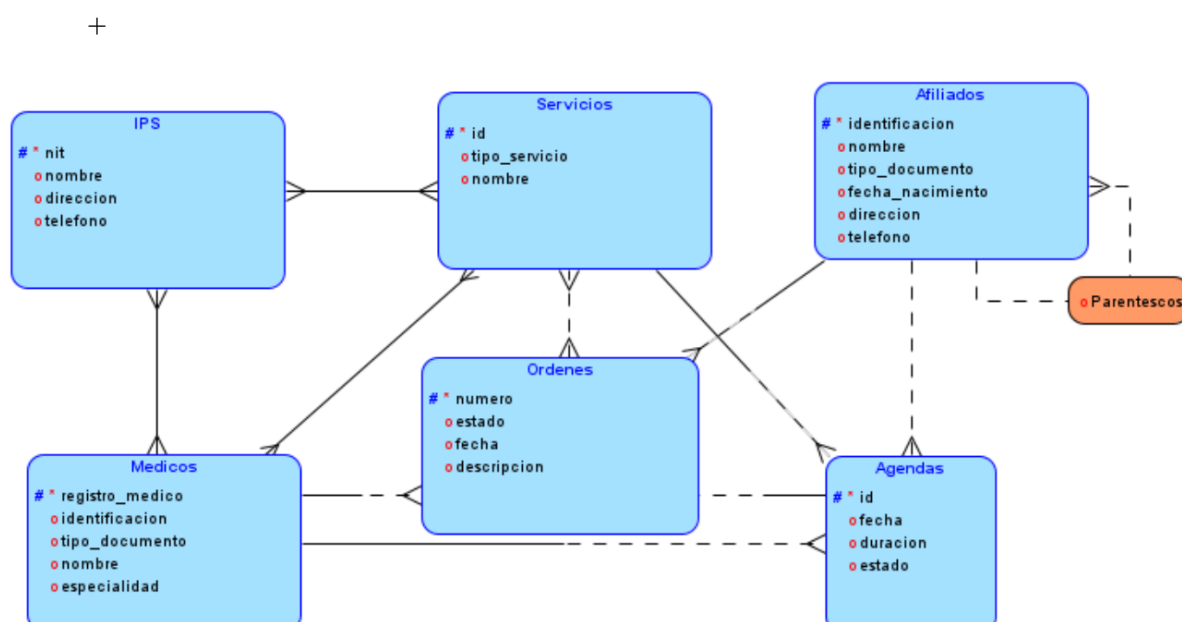
2. Relaciones clave Un Afiliado puede recibir múltiples Órdenes médicas. Un Médico puede trabajar en múltiples IPS y prescribir múltiples Órdenes. Una Orden puede incluir múltiples Servicios. Una IPS ofrece múltiples Servicios y contrata varios Médicos. Un Servicio puede ser prestado por múltiples IPS. Un Afiliado contribuyente puede tener múltiples Beneficiarios asociados a él mediante Parentesco. La Agenda asocia horarios disponibles de una IPS con los Servicios que presta. Si un Médico deja de trabajar, la Agenda sigue existiendo porque depende de la IPS, no del Médico.
3. Justificación de composición y agregación Composición (Agenda - Servicio). Una Agenda no puede existir sin un Servicio.

Agregación (Agenda - Médico). Un Médico puede estar en la Agenda si tiene citas programadas, pero su eliminación no afecta la existencia de la Agenda.

Relación Afiliado - Agenda (0..1). Un afiliado puede tener como máximo una cita activa en la Agenda, pero no siempre la tiene. Relación Orden - Médico (1..). Cada Orden debe ser prescrita por un médico, pero un médico puede emitir muchas órdenes. Relación Servicio - IPS (1..). Cada servicio debe ser prestado por al menos una IPS.

### Modelo conceptual en E/R

El siguiente es nuestro modelo en E/R construido en Data Modeler.



*Ilustración 2. Modelo E/R EPSANDES*

Este modelo de EPSANDES está compuesto por diversas entidades principales que representan los actores y procesos involucrados en la prestación de servicios de salud. Estas entidades incluyen afiliados, médicos, IPS, servicios, órdenes, agendas y parentesco, esta última, está dentro de una relación de afiliados a afiliados, la cual representa cuando un afiliado es beneficiario de otro, el parentesco de este. En la imagen se ven las relaciones que modelan la interacción entre estas entidades.

Las siguientes son sus entidades con sus atributos:

#### IPS (Instituciones Prestadoras de Salud)

- ✓ NIT (identificador único)
- ✓ Nombre
- ✓ Dirección
- ✓ Teléfono

#### Servicios

- ✓ ID (identificador único)
- ✓ Tipo de servicio (enum: General, Especialista, Control, Urgencias, Examen, Terapia, Hospitalizacion Y Procedimiento\_Especializado)
- ✓ Nombre del servicio

#### Afiliados

- ✓ Identificación (identificador único)
- ✓ Nombre
- ✓ Tipo de documento: (enumeración: RC (Registro Civil), TI (Tarjeta de Identidad), CC (Cedula de Ciudadanía) y CE (Cedula de Extranjería))
- ✓ Fecha de nacimiento
- ✓ Dirección
- ✓ Teléfono

Parentescos, la cual está en la relación entre afiliados (beneficiario y contribuyente)

#### Médicos

- ✓ Registro médico (identificador único)
- ✓ Identificación
- ✓ Tipo de documento: (enumeración: RC (Registro Civil), TI (Tarjeta de Identidad), CC (Cedula de Ciudadanía) y CE (Cedula de Extranjería))
- ✓ Nombre



- ✓ Especialidad

#### Órdenes

- ✓ Número (identificador único)
- ✓ Estado (enumeración: vigente, cancelada, completada)
- ✓ Fecha
- ✓ Descripción del servicio solicitado

#### Agendas

- ✓ ID (identificador único)
- ✓ Fecha
- ✓ Duración
- ✓ Estado: (enumeración: Cancelada, Terminada, Ausencia y Sin Agendar)

Y en cuanto las relaciones son las siguientes:

- ✓ IPS - Servicios

Relación de muchos a muchos. Una IPS ofrece varios servicios de salud y los servicios de salud son ofrecidos por muchas IPS.

- ✓ Médicos - IPS

Relación de muchos a muchos. Las IPSS tienen muchos médicos Los médicos tienen una o muchas IPS en las cuales están asociados.

- ✓ Afiliados - Órdenes

Relación de uno a muchos. Un afiliado puede tener varias órdenes de servicio, pero una orden solo puede y tiene que ser de un afiliado.

- ✓ Médicos - Órdenes

Relación de uno a muchos. Un médico puede generar varias órdenes de servicio y una orden puede ser generada solamente por un médico.

- ✓ Afiliados - Parentescos

Relación de uno a muchos. Un afiliado contribuyente puede tener varios beneficiarios.

✓ Afiliados - Agendas

Relación de uno a muchos. Un afiliado puede agendar varias citas, y una cita puede ser solo para un afiliado.

✓ Médicos – Servicios

Relación muchos a muchos. Un médico tiene uno o más servicios asociados y los servicios tienen que ser prestados por uno o más médicos.

✓ Médicos – Agendas

Relación de uno a muchos. Un medico puede prestar muchas citas y una cita tiene que ser prestada por un medico.

✓ Ordenes – Agendas

Relación de uno a uno. Una orden esta asociada a una cita y una cita puede tener una orden o no, dependiendo del servicio.

✓ Servicios – Ordenes

Relación de muchos a muchos. Un servicio puede tener muchas ordenes, y una orden puede ser de muchos servicios o puede ser de medicamentos por ende no de servicios.

### **Diseño de la base de datos**

#### **Modelo de datos relacional**

1) Primer paso del algoritmo de Peter Chen: Conversión de clases básicas

- ◆ La tabla de *afiliados* contiene los siguientes atributos con sus respectivas restricciones: identificacion, que es de tipo User Assigned (US) y actúa como Primary Key (PK), asegurando que cada afiliado tenga un número único; tipo\_documento, que tiene restricciones Not Null (NN) y Check (CK), garantizando que solo se acepten los valores definidos en el enumeration TipoDoc; nombre, que no puede ser nulo (NN); fecha\_nacimiento, que también es Not Null (NN); direccion, que no admite valores nulos (NN); y telefono, que igualmente es Not Null (NN). Adicionalmente, se consideró definir el atributo telefono como No Duplicado (ND), pero se decidió no

aplicarlo, ya que un número de teléfono puede repetirse si está asociado a un mismo núcleo familiar

- ♦ La tabla de *órdenes* contiene los siguientes atributos con sus respectivas restricciones: número, que es de tipo System Assigned (SA) y actúa como Primary Key (PK), asegurando que cada orden de servicio tenga un identificador único; estado, que tiene restricciones Not Null (NN) y Check (CK), garantizando que solo se acepten los valores definidos en el enumeration estadoOrden; fecha, que es de tipo Not Null (NN) y almacena la fecha en la que se genera la orden; y descripción, que debe ser no nula porque incluye detalles como medicamentos y especificaciones requeridas para la orden.
- ♦ La tabla de *agendas* contiene los siguientes atributos con sus respectivas restricciones: id, que es de tipo System Assigned (SA) y actúa como Primary Key (PK), asegurando que sea una llave subrogada de identificación única; fecha, que es de tipo Not Null (NN) y almacena la fecha en la que se ejecutará la cita; duración, que tiene restricciones Not Null (NN) y Check (CK), garantizando que el valor sea mayor a 0 minutos; y estado, que debe ser no nulo (NN) y cuenta con una restricción Check (CK) para asegurar que sus valores correspondan a los definidos en el enumeration estadoCita.
- ♦ La tabla de *médicos* contiene los siguientes atributos con sus respectivas restricciones: registro médico, que es de tipo User Assigned (UA) y actúa como Primary Key (PK), asegurando que sea una llave de identificación única; identificación, que debe ser no nula (NN) y no duplicada (ND); tipo\_documento, que no debe ser nulo (NN) y debe cumplir con una restricción Check (CK) para garantizar que tenga valores que están dentro de la enumeration Tipo\_Doc; nombre, que no puede ser nulo (NN) porque es el nombre del médico; y especialidad, que proviene de una lista definida por el ministerio de salud y no puede ser nula (NN).

- ◆ La tabla de *IPS (Instituciones Prestadoras de Salud)* contiene los siguientes atributos: nit, que actúa como la llave primaria (PK) y es de tipo User Assigned (UA), asegurando que sea un identificador único; y atributos que no pueden ser nulos (NN), como nombre, dirección y teléfono.
- ◆ La tabla de *Servicios* contiene los siguientes atributos: id, que es una llave subrogada de tipo System Assigned (SA) y actúa como la llave primaria (PK), asegurando un identificador único; tipo de servicio, que debe tomar valores definidos en la enumeración tipoServicio; y nombre, que no puede ser nulo (NN).

## 2) Segundo paso del algoritmo de Peter Chen: Conversión de asociaciones

### 2.1) Asociaciones uno a uno, uno a muchos, compartidas y composiciones.

- ✓ A la tabla de *afiliados* se le agregaron dos atributos (columnas): el primero, llamado contribuyente, se añadió debido a la autoasociación que existe en el caso de que un afiliado sea beneficiario de otro afiliado. Este atributo almacena la identificación del afiliado contribuyente y se decidió incluirlo porque un contribuyente puede tener múltiples beneficiarios, mientras que un beneficiario solo puede tener un único contribuyente. Dado que no es obligatorio que un afiliado tenga un contribuyente, no se aplicó la restricción de Not Null (NN). El segundo atributo, parentesco, se añadió porque este era el tipo de relación entre los afiliados que se describía mediante una clase de asociación en la auto-relación, permitiendo identificar el tipo de relación familiar entre el contribuyente y el beneficiario.

Afiliados

identificacion	tipo_documento	nombre	fecha_nacimiento	direccion	telefono	contribuyente	parentesco
PK, UA	NN, CK[RC, TI, CC, CE]	NN	NN	NN	NN	FK <sub>afiliados</sub> identificacion	

Tabla 1. Afiliados

- ✓ A la tabla de *órdenes* se le agregaron dos atributos (columnas): el primero, llamado prescritaPor, se añadió debido a que, si existe una orden, esta debe ser prescrita por un

médico. Por ello, se incluyó como Foreign Key (FK) la PK de Médicos, llamada registro. Este atributo no puede ser nulo (NN), ya que es una asociación uno a muchos donde el extremo univalue es obligatorio, en este caso, el médico. También se añadió la columna prescritaPara, debido a que, si existe una orden, esta debe estar dirigida hacia un afiliado. Por ello, se incluyó como Foreign Key (FK) la PK de Afiliados, llamada identificación. Este atributo tampoco puede ser nulo (NN), ya que la asociación es uno a muchos donde el extremo univalue es obligatorio, en este caso, el afiliado.

Ordenes

numero	estado	fecha	descripcion	prescritaPor	prescritaPara
PK, SA	NN, CK[Vigente, Completada, Cancelada]	NN	NN	FKmedicos registro, NN	FKafiliados identificacion, NN

Tabla 2. Ordenes

- ✓ A la tabla de *agendas* se le agregaron cuatro atributos (columnas): el primero, llamado paciente\_cita, se añadió debido a que un paciente puede o no adquirir una cita disponible, por lo que se permite que este atributo sea nulo. Se incluye como Foreign Key (FK) la Primary Key (PK) de Afiliados, que es la identificación. Es importante tener en cuenta que un afiliado puede tener muchas citas asociadas o ninguna, por esta razón se agregó la columna en órdenes y no en afiliados. El segundo atributo, medico\_cita, se añadió porque un médico tiene una relación de agregación con las agendas, lo que implica que toda cita debe estar vinculada a un médico, por lo que este atributo no puede ser nulo (NN). Es de recordar que las asociaciones de tipo agregación compartida se tratan de la misma manera que las asociaciones para su conversión al modelo relacional. El tercer atributo, servicio\_cita, se incluyó debido a la relación de composición con la tabla de servicios, donde la Foreign Key (FK) será la PK de Servicios, es decir, el id. Es importante tener en cuenta que una asociación compuesta es obligatoria con la clase 'todo', por lo que este atributo no puede ser nulo

(NN) y debería colocarse en la tabla de la clase parte, en este caso son las Agendas.

Finalmente, el cuarto atributo, orden\_cita, se añadió porque una cita puede o no necesitar una orden asociada, por lo que no se restringe la nulidad; y se usa como Foreign Key (FK) la Primary Key (PK) de la tabla de Ordenes, llamada número.

Agendas							
id	fecha	duracion	estado	paciente_cita	medico_cita	servicio_cita	orden_cita
PK, SA	NN	NN, CK[>0]	NN, CK[Cancelada, Terminada, Ausencia, Sin_Agendar]	FK.alfados.identificacion	FK.medicos.registro, NN	FK.servicio.id, NN	FK.ordenes.numero

Tabla 3. Agendas

## 2.2) Asociaciones muchos a muchos

Se construyeron nuevas tablas de relaciones muchos a muchos debido a que, para modelar la asociación entre las dos clases, con FK a las tablas originales y como PK ambos atributos.

- ✓ En la primera tabla, llamada Contratados\_En, se establece la relación entre médicos e IPS (Instituciones Prestadoras de Salud). Esta tabla contiene dos Foreign Keys (FK): la FK de **Médicos, que es el registro, y la FK de IPS, que es el nit. Ambas claves se catalogan como Primary Keys (PK) compuestas en esta tabla, ya que un médico puede estar contratado en varias IPS y, a su vez, una IPS puede contratar a varios médicos.**

Contratados_En	
registro_med	nit_ips
PK, Fkmedicos.registro	PK, Fkips.nit

Tabla 4. Contratados\_En

- ✓ En la segunda tabla, llamada Ips\_Prestacion, se establece la relación entre IPS (Instituciones Prestadoras de Salud) y Servicios. Esta tabla contiene dos Foreign Keys (FK): la FK de IPS, que es el nit, y la FK de Servicios, que es el id. Ambas claves se catalogan como Primary Keys (PK) compuestas en esta tabla, ya que una ips puede prestar varios servicios y, a su vez, un Servicio puede ser prestado por varias IPS.

Ips_Prestacion	
nit_ips	id_servicio
PK, Fkips.nit	PK, Fkservicios.id

*Tabla 5. Ips Prestación*

- ✓ En la tercera tabla, llamada Servicio\_A\_Prestar, se establece la relación entre Servicios y Médicos. Esta tabla contiene dos Foreign Keys (FK): la FK de Servicios, que es el id, y la FK de Médicos, que es el registro. Ambas claves se catalogan como Primary Keys (PK) compuestas en esta tabla, ya que un Servicio puede ser prestado por varios médicos y, a su vez, un Médico puede prestar varios Servicios.

Servicio_A_Prestar	
id_servicio	registro_med
PK, Fkservicios.id	PK, Fkmedicos.registro

*Tabla 6. Servicio a prestar*

## Nivel de normalización del modelo

### *1. Primera Forma Normal (1FN)*

Para que una tabla esté en **1FN**, no deben existir atributos multivalor en la relación, es decir, todos los atributos tienen dominios atómicos.

### *2. Segunda Forma Normal (2FN)*

La **2FN** requiere que la tabla esté en 1FN y que no existan atributos no-primos que dependan parcialmente de llaves candidatas.

### 3. Tercera Forma Normal (3FN)

Para que una tabla este en **3FN**, La relación debe estar en 2FN y además cada atributo no-primo debe depender únicamente de llaves candidata y no de otros atributos no-primos, es decir, no hay dependencias funcionales transitivas entre atributos no-primos.

### 4. Forma Normal de Boyce-Codd (BCNF)

Para que una tabla esté en BCNF, debe cumplir con **3FN** y, además, todas las dependencias funcionales deben estar definidas por una superllave.

### Evaluación de Normalización por Tabla

#### Afiliados

**Cumple con 1FN:** Sí. Todos los atributos tienen valores atómicos.

**Cumple con 2FN:** No. Presenta dependencias parciales.

#### Ejemplo de dependencia parcial:

identificacion → nombre, direccion, telefono, contribuyente, parentesco

contribuyente → parentesco

En la tabla Afiliados, la clave primaria es identificacion, lo que significa que todos los atributos deberían depender únicamente de identificacion. Sin embargo, contribuyente no depende de todos los afiliados de la tabla, sino solo de los beneficiarios (es decir, aquellos que tienen un contribuyente). Esto implica que: Hay afiliados que son contribuyentes (por ejemplo, un trabajador que paga su propia afiliación y la de sus hijos). Hay afiliados que son beneficiarios (por ejemplo, los hijos de ese trabajador, que dependen de él). Dado que contribuyente solo aplica a los beneficiarios, no depende de toda la clave primaria de la tabla. Es decir, su valor depende solo de ciertos afiliados, lo que indica una dependencia parcial.

**Cumple con 3FN:** No. Presenta dependencias transitivas.

#### Ejemplo de dependencia transitiva:

identificacion → contribuyente → parentesco

parentesco depende de contribuyente, que a su vez depende de identificacion, lo que rompe la Tercera Forma Normal.

**Cumple con BCNF:** No. No todas las dependencias funcionales están en una superllave.



## 2. Ordenes

**Cumple con 1FN:** Sí. Todos los atributos tienen valores atómicos.

**Cumple con 2FN:** No. Presenta dependencias parciales.

**Ejemplo de dependencia parcial:**

numero  $\rightarrow$  estado, fecha, descripcion, prescritaPor, prescritaPara

prescritaPor  $\rightarrow$  especialidad

prescritaPor no depende de numero, sino de la relación con Médicos, lo que genera una dependencia parcial. Las órdenes de servicio son prescritas por médicos, y cada médico tiene una especialidad definida.

**Cumple con 3FN:** No. Presenta dependencias transitivas.

**Ejemplo de dependencia transitiva:**

numero  $\rightarrow$  prescritaPor  $\rightarrow$  especialidad

especialidad depende de prescritaPor, que a su vez depende de numero, lo que rompe la Tercera Forma Normal.

**Cumple con BCNF:** No. No todas las dependencias funcionales están en una superllave.

## 3. AgendasDisponibles

**Cumple con 1FN:** Sí. Todos los atributos tienen valores atómicos.

**Cumple con 2FN:** No. Presenta dependencias parciales.

**Ejemplo de dependencia parcial:**

id  $\rightarrow$  fecha, duracion, estado, paciente\_cita, medico\_cita, servicio\_cita, orden\_cita

medico\_cita  $\rightarrow$  especialidad

medico\_cita está relacionado con Médicos, lo que indica una dependencia parcial dentro de la tabla AgendasDisponibles.

**Cumple con 3FN:** No. Presenta dependencias transitivas.

**Ejemplo de dependencia transitiva:**

id  $\rightarrow$  medico\_cita  $\rightarrow$  especialidad

especialidad depende de medico\_cita, que a su vez depende de id, lo que rompe la Tercera Forma Normal.

**Cumple con BCNF:** No. No todas las dependencias funcionales están en una superllave.

#### 4. Médicos

**Cumple con 1FN:** Sí. Todos los atributos tienen valores atómicos.

**Cumple con 2FN:** No. Presenta dependencias parciales.

##### **Ejemplo de dependencia parcial:**

registro  $\rightarrow$  identificacion, tipo\_documento, nombre, especialidad

identificacion  $\rightarrow$  tipo\_documento, nombre

identificacion genera una dependencia parcial dentro de la tabla Medicos, ya que tipo\_documento y nombre no dependen directamente de registro.

**Cumple con 3FN:** No. Presenta dependencias transitivas.

##### **Ejemplo de dependencia transitiva:**

registro  $\rightarrow$  identificacion  $\rightarrow$  tipo\_documento, nombre

tipo\_documento y nombre dependen de identificacion, que a su vez depende de registro, lo que rompe la Tercera Forma Normal.

**Cumple con BCNF:** No. No todas las dependencias funcionales están en una superllave.

#### 5. IPS

**Cumple con 1FN:** Sí. Todos los atributos tienen valores atómicos.

**Cumple con 2FN:** No. Presenta dependencias parciales.

##### **Ejemplo de dependencia parcial:**

nit  $\rightarrow$  nombre, direccion, telefono

nombre  $\rightarrow$  telefono

telefono depende de nombre, y no exclusivamente de nit, lo que indica una dependencia parcial.

**Cumple con 3FN:** No. Presenta dependencias transitivas.

##### **Ejemplo de dependencia transitiva:**

nit  $\rightarrow$  nombre  $\rightarrow$  telefono

telefono depende de nombre, que a su vez depende de nit, lo que rompe la Tercera Forma Normal.

**Cumple con BCNF:** No. No todas las dependencias funcionales están en una superllave.

## 6. Servicios

**Cumple con 1FN:** Sí. Todos los atributos tienen valores atómicos.

**Cumple con 2FN:** No. Presenta dependencias parciales.

**Ejemplo de dependencia parcial:**

$id \rightarrow tipo\_servicio, nombre$

$tipo\_servicio \rightarrow nombre$

nombre depende de tipo\_servicio, no de id, lo que genera una dependencia parcial en la tabla Servicios.

**Cumple con 3FN:** No. Presenta dependencias transitivas.

**Ejemplo de dependencia transitiva:**

$id \rightarrow tipo\_servicio \rightarrow nombre$

nombre depende de tipo\_servicio, que a su vez depende de id, lo que rompe la Tercera Forma Normal.

**Cumple con BCNF:** No. No todas las dependencias funcionales están en una superllave.

## 7. Contratados\_En (*Tabla de relación muchos a muchos*)

**Cumple con 1FN:** Sí.

**Cumple con 2FN:** Sí.

**Cumple con 3FN:** Sí.

**Cumple con BCNF:** Sí.

Esta tabla representa la relación entre médicos e IPS, indicando en qué institución cada médico está contratado. Dado que solo contiene claves foráneas y no presenta dependencias parciales ni transitivas, cumple con BCNF.

## 8. Ips\_Prestacion (*Tabla de relación muchos a muchos*)

**Cumple con 1FN:** Sí.

**Cumple con 2FN:** Sí.

**Cumple con 3FN:** Sí.

**Cumple con BCNF:** Sí.

Esta tabla modela la relación entre IPS y los servicios de salud que prestan. Cada IPS puede ofrecer múltiples servicios y cada servicio puede ser ofrecido por varias IPS. Al contener solo claves primarias y cumplir con los requisitos de BCNF, se encuentra correctamente normalizada.

### **9. A\_Prestar (Tabla de relación muchos a muchos)**

**Cumple con 1FN:** Sí.

**Cumple con 2FN:** Sí.

**Cumple con 3FN:** Sí.

**Cumple con BCNF:** Sí.

Esta tabla establece la relación entre los servicios de salud y los médicos que los prestan. Un servicio puede ser realizado por múltiples médicos y un médico puede estar habilitado para realizar múltiples servicios. Al ser una tabla de relación con claves primarias y sin dependencias indebidas, cumple con BCNF.

## **Escenarios de prueba**

### **REQUERIMIENTOS FUNCIONALES**

#### **RF1 – Registrar IPS**

##### ***Escenario RF1.1. IPS completa***

Caso exitoso: La IPS debe tener la información de todos sus atributos llena. Como lo son su NIT, nombre, dirección y teléfono.

Caso de falla: Si hace falta algún atributo, debe arrojar un mensaje de error que diga que a esa IPS le hace falta la información de un atributo.

##### ***Escenario RF1.2: NIT duplicado***

Caso exitoso: No puede existir una IPS con un NIT que ya existe en la base de datos.

Caso de falla: Se muestra un mensaje de error indicando que el NIT ya está registrado.

**RF2 – Registrar un servicio*****Escenario RF2.1: Servicio completo***

Caso exitoso: El servicio de salud tiene todos los atributos obligatorios (por ejemplo, id, nombre y tipo).

Caso de falla: Si hace falta algún atributo, debe arrojar un mensaje de error que diga que a esa IPS le hace falta la información de un atributo.

***Escenario RF2.2: Servicio de salud duplicado***

Caso exitoso: No se permite registrar un servicio de salud que ya se encuentre registrado (basado en su identificador único).

Caso de falla: Se muestra un mensaje de error indicando que el servicio ya existe.

**RF3 – Asignar un servicio de salud a una IPS*****Escenario RF3.1: Asignación exitosa***

Caso exitoso: La asignación se realiza correctamente cuando tanto la IPS como el servicio de salud han sido creados previamente.

Caso de falla: Si la IPS o el servicio de salud no existen en el sistema, se muestra un mensaje de error indicando la inexistencia de alguno de ellos.

**RF4 – Registrar médico*****Escenario RF4.1: Médico con información completa***

Caso exitoso: Se registra correctamente un médico cuando se proporcionan todos los atributos obligatorios: nombre, tipo y número de identificación, especialidad, número de registro médico y la IPS (o IPSs) a la(s) que está adscrito.

Caso de falla: Si falta algún atributo, se arroja un mensaje de error especificando el dato que falta.

***Escenario RF4.2: Número de registro duplicado***

Caso exitoso: No se permite el registro de un médico con un número de registro que ya exista en la base de datos.

Caso de falla: Se muestra un mensaje de error indicando que el número de registro ya está en uso.

## **RF5 – Registrar afiliado**

### ***Escenario RF5.1: Afiliado con información completa***

Caso exitoso: Se registra un afiliado cuando se ingresan todos sus datos obligatorios: tipo y número de identificación, nombre, fecha de nacimiento, dirección y teléfono.

Caso de falla: Si falta alguno de estos atributos, se muestra un mensaje de error.

### ***Escenario RF5.2: Afiliado beneficiario con vinculación correcta***

Caso exitoso: Se registra un afiliado de tipo beneficiario únicamente si se incluye la relación de parentesco y la vinculación es satisfactoria.

Caso de falla: Si se omite la relación de parentesco o la vinculación con el contribuyente, se arroja un mensaje de error.

### ***Escenario RF5.3: Identificación duplicada***

Caso exitoso: No se permite registrar un afiliado con una identificación que ya se encuentre registrada en el sistema.

Caso de falla: Se muestra un mensaje de error indicando que la identificación ya está registrada.

## **RF6 – Registrar una orden de servicio de salud para un afiliado por parte de un médico**

### ***Escenario RF6.1: Orden de servicio completa***

Caso exitoso: Se registra correctamente la orden cuando se ingresan todos los atributos requeridos: fecha, paciente, servicio de salud y el médico que prescribe.

Caso de falla: Si falta alguno de estos atributos, se muestra un mensaje de error indicando el dato faltante.

### ***Escenario RF6.2: Orden de servicio emitida por un medico***

Caso exitoso: Se registra correctamente la orden cuando la prescribe un médico autorizado.

Caso de falla: Si alguien no autorizado intenta crear la orden, se muestra un mensaje de error y no se logra crear la orden.

***Escenario RF6.3: Datos inexistentes en la orden***

Caso exitoso: La orden se registra correctamente si el afiliado y el médico existen en el sistema.

Caso de falla: Si el afiliado o el médico no se encuentran registrados, se arroja un mensaje de error señalando el dato inexistente.

**RF7 – Agendar un servicio de salud por parte de un afiliado**

***Escenario RF7.1: Consulta de agenda y agendamiento exitoso***

Caso exitoso: El afiliado consulta la agenda de disponibilidad para el servicio de salud en las próximas 4 semanas y agenda la cita en una fecha y hora disponibles, utilizando un número de orden válido previamente emitido.

Caso de falla: Si el servicio no existe o el número de orden es inválido, se muestra un mensaje de error.

***Escenario RF7.2: Servicio sin disponibilidad***

Caso exitoso: El sistema no permite agendar un servicio cuando no hay disponibilidad en la fecha y hora solicitadas.

Caso de falla: Se muestra un mensaje de error indicando que no hay disponibilidad en el horario seleccionado.

***Escenario RF7.3: Agendamiento cumple pre-requisitos***

Caso exitoso: El sistema no permite agendar un servicio, a no ser que sea cita por médico general o por urgencia, sin un número de orden.

Caso de falla: Se muestra un mensaje de error indicando que no se está ingresando el número de orden y por ende no se puede generar.

**RF8 – Registrar la prestación de un servicio de salud a un afiliado por parte de una IPS*****Escenario RF8.1: Prestación del servicio con información completa***

Caso exitoso: Se registra la prestación del servicio correctamente cuando se ingresan todos los datos obligatorios: afiliado, IPS, médico, fecha y hora de la prestación. Y le ingresa el estado correcto de la cita.

Caso de falla: Si falta alguno de estos datos, se muestra un mensaje de error indicando cuál es el campo faltante.

***Escenario RF8.2: Datos inexistentes en la prestación***

Caso exitoso: La prestación se registra correctamente siempre que la IPS, el médico y el afiliado existan en el sistema.

Caso de falla: Si alguno de estos elementos no existe, se arroja un mensaje de error indicando el elemento no encontrado.

**REQUERIMIENTOS FUNCIONALES DE CONSULTA****RFC1 – Consultar la agenda de disponibilidad de un servicio de salud en las siguientes 4 semanas*****Escenario RFC1.1: Consulta con código de servicio válido y con disponibilidad***

Caso exitoso: Al ingresar el código de un servicio existente, el sistema muestra la agenda para las próximas 4 semanas, listando en cada registro: el nombre del servicio, la fecha y hora de disponibilidad, la IPS que lo ofrece y el nombre del médico asignado.

Caso de falla: Si el código de servicio no corresponde a un servicio existente o no se encuentra disponibilidad en el período consultado, se muestra un mensaje de error indicando que el servicio no existe o que no hay horarios disponibles.

***Escenario RFC1.2: Consulta con código de servicio válido, pero sin disponibilidad***

Caso exitoso: Se ingresa un código de servicio existente y, en caso de no haber horarios disponibles en las próximas 4 semanas, el sistema informa que no hay disponibilidad en el período consultado.



Caso de falla: Si se muestra información incompleta o errónea, se arroja un mensaje de error pertinente.

## **RFC2 – Mostrar los 20 servicios más solicitados en un período de tiempo dado**

### ***Escenario RFC2.1: Consulta con período de fechas válido y datos de solicitudes***

Caso exitoso: Al ingresar un rango de fechas válido, el sistema muestra una lista ordenada de los 20 servicios más solicitados durante ese período, basándose en la cantidad de solicitudes realizadas.

Caso de falla: Si en el período consultado no se encuentran registros de solicitudes, se muestra un mensaje informando que no hay datos suficientes para generar el listado.

### ***Escenario RFC2.2: Consulta con período de fechas inválido***

Caso exitoso: Se ingresa una fecha que se encuentre en el futuro y que sea válida.

Caso de falla: Si se ingresa un rango de fechas inválido (por ejemplo, la fecha de inicio es posterior a la fecha final), se muestra un mensaje de error indicando que el rango de fechas es incorrecto.

## **RFC3 – Mostrar el índice de uso de cada uno de los servicios provistos**

### ***Escenario RFC3.1: Consulta con período de fechas válido y datos disponibles***

Caso exitoso: Con fechas de inicio y fin válidas, el sistema calcula y muestra el índice de uso para cada servicio (dividiendo el número de servicios disponibles por el número de servicios usados) en el período indicado.

Caso de falla: Si para algún servicio el número de servicios usados es cero, el sistema debe manejar adecuadamente la división (por ejemplo, mostrando un valor predeterminado o un mensaje específico) o, en caso de error, mostrar un mensaje indicando que la operación no es válida.

### ***Escenario RFC3.2: Consulta con período de fechas inválido***

Caso exitoso: Se ingresan las fechas validas.

Caso de falla: Al ingresar un rango de fechas inválido, se muestra un mensaje de error que indique que el período es incorrecto.

#### **RFC4 – Mostrar la utilización de servicios de EPSAndes por un afiliado en un período dado**

##### ***Escenario RFC4.1: Consulta con afiliado existente y rango de fechas válido***

Caso exitoso: Al ingresar el identificador de un afiliado registrado junto con fechas de inicio y fin válidas, el sistema muestra una lista en la que cada registro contiene: el nombre del servicio, la fecha en que fue tomado o el estado de la cita si no fue tomada, el médico que lo atendió y la IPS que ofreció el servicio.

Caso de falla: Si el afiliado no se encuentra registrado o alguno de los datos ingresados es incorrecto, se muestra un mensaje de error indicando el problema.

##### ***Escenario RFC4.2: Consulta con rango de fechas inválido o sin registros***

Caso exitoso: Se ingresa un rango de fechas valido.

Caso de falla: Si se ingresa un rango de fechas inválido o no se encuentran registros de utilización para el afiliado en el período indicado, se muestra un mensaje informando que no se encontraron registros o que el rango de fechas es incorrecto.