

## PUNTO 3

A)

## a. Identificación de entidades y atributos

Entidad	Atributos principales
Afiliado	id, nombre, tipoDocumento, numeroDocumento, fechaNacimiento, direccion, telefono
Medico	id, nombre, tipoDocumento, numeroDocumento, especialidad, numeroRegistroMedico, adscritoIps, ipsIds
IPS	id, nit, nombre, direccion, telefono
ServicioSalud	id, nombre, fechaDisponibilidad, horaDisponibilidad, ipsIds, medicId
Orden	id, afiliadoId, medicId, fechaEmision, estado, detalleOrden (embebido)
Agendamiento	id, afiliadoId, medicId, fechaAgendamiento, horaAgendamiento, estado, servicioSalud (embebido)

## b. Cuantificación aproximada de registros

Entidad	Estimación aproximada de registros
Afiliado	900.000 a 10.000.000
Medico	1.000 a 9.000
IPS	Hasta 10.000
ServicioSalud	Alrededor de 20
Orden	90.000 por mes → 1.000.000/año
Agendamiento	90.000 por mes → 1.000.000/año

## c. Tabla de operaciones de lectura y escritura (Anexo A)

Entities	Operations	Information Needed	Type
Afiliado	Consultar datos del afiliado	Detalles del afiliado	Read
	Registrar/modificar afiliado	Detalles del afiliado	Write

Medico	Consultar datos de médico	Nombre, especialidad, registro médico, IPS asociadas	Read
	Registrar/modificar médico	Detalles del médico e IPS asignadas	Write
IPS	Consultar IPS disponibles	Nombre, dirección, servicios prestados	Read
	Registrar nueva IPS	NIT, nombre, contacto	Write
ServicioSalud	Consultar agenda del servicio	Nombre del servicio, fechas, médico, IPS	Read
	Registrar/modificar servicio	Nombre, IPS asociadas, disponibilidad	Write
Orden	Registrar nueva orden de servicio	Afiliado, médico, servicios prescritos	Write
	Consultar órdenes existentes	Estado, servicio, médico que la emitió	Read
Agendamiento	Consultar disponibilidad de cita	Fecha, hora, servicio, IPS, médico	Read
	Reservar cita (agendamiento)	Afiliado, orden previa, fecha/hora	Write

d. Tabla de cuantificación de operaciones (Anexo B)

Entities	Operation	Information Needed	Type	Rate
Afiliado	Consultar datos del afiliado	Datos personales y contacto	Read	7/seg (10.000/día)
	Registrar/modificar afiliado	Nombre, documento, dirección, etc.	Write	0.2/seg (9.000/mes)
Medico	Consultar médicos disponibles	Nombre, especialidad, IPS	Read	3/seg (5.000/día)
	Registrar/modificar médico	Datos personales + IPS asociadas	Write	0.003/seg (100/mes)

IPS	Consultar IPS	Nombre, servicios, dirección	Read	2/seg (5.000/día)
	Registrar/modificar IPS	NIT, nombre, contacto	Write	<0.01/seg (10/mes)
ServicioSalud	Consultar servicios disponibles	Nombre, disponibilidad, médico, IPS	Read	7/seg (10.000/día)
	Registrar/modificar servicio	Nombre, médico, IPS	Write	0.001/seg (1/mes)
Orden	Registrar nueva orden	Afiliado, médico, servicio	Write	35/seg (3.000/día)
	Consultar órdenes	Estado, afiliado, servicio	Read	35/seg (3.000/día)
Agendamiento	Consultar disponibilidad	Servicio, médico, IPS, fecha y hora	Read	60/seg (5.000/día)
	Agendar servicio	Afiliado, fecha/hora, orden si aplica	Write	23/seg (2.000/día)

## B)

## a.

## 1. afiliados

- Descripción: Representa a los usuarios del sistema que están afiliados a una EPS. Pueden ser contribuyentes o beneficiarios.
- Campos principales: id, nombre, tipoDocumento, numeroDocumento, fechaNacimiento, direccion, telefono.
- Relaciones:
  - Referenciado por órdenes, agendamientos, terapias.

## 2. medicos

- Descripción: Contiene los datos de los profesionales de salud registrados en una o varias IPS.
- Campos principales: id, nombre, tipoDocumento, numeroDocumento, especialidad, numeroRegistroMedico, adscritoIps, ipsIds (lista de IDs de IPS).
- Relaciones:
  - Referenciado por órdenes, serviciosSalud, agendamientos.

### 3. ips

- Descripción: Instituciones Prestadoras de Salud que ofrecen uno o varios servicios médicos. No prestan servicios directamente en el sistema.
- Campos principales: id, nit, nombre, direccion, telefono.
- Relaciones:
  - Referenciada por medicos y serviciosSalud.

### 4. serviciosSalud

- Descripción: Define cada tipo de servicio médico ofrecido por la EPS (consultas, terapias, exámenes, etc.).
- Campos principales: id, nombre, fechaDisponibilidad, horaDisponibilidad, ipsIds, medicold.
- Relaciones:
  - Referenciado por órdenes (embebido en detalleOrden) y agendamientos (embebido directamente).

### 5. ordenes

- Descripción: Documentos generados por un médico para solicitar un servicio de salud para un afiliado.
- Campos principales: id, afiliadold, medicold, fechaEmision, estado, detalleOrden.
- Relaciones:
  - detalleOrden contiene un documento embebido servicioSalud.

### 6. agendamientos

- Descripción: Representa la reserva de una cita para un servicio médico por parte de un afiliado.
- Campos principales: id, afiliadold, medicold, fechaAgendamiento, horaAgendamiento, estado, servicioSalud (embebido).
- Relaciones:
  - Embebe un servicioSalud para reflejar el estado en el momento de la reserva.

### 7. terapias

- Descripción: Almacena información sobre terapias ordenadas a un afiliado, incluyendo tipo y número de sesiones.
- Campos principales: id, tipoTerapia, numeroSesiones, afiliadold.
- Relaciones:
  - Referencia al afiliado que recibe la terapia.

b.

➤ **Afiliado – Orden**

- Un afiliado puede tener muchas órdenes de servicio.
- **Tipo de relación:** uno a muchos (1:N)
- **Representación NoSQL:** afiliadold referenciado en la colección ordenes.

➤ **Afiliado – Agendamiento**

- Un afiliado puede realizar múltiples agendamientos.
- **Tipo de relación:** uno a muchos (1:N)
- **Representación NoSQL:** afiliadold referenciado en la colección agendamientos.

➤ **Afiliado – Terapia**

- Un afiliado puede recibir varias terapias.
- **Tipo de relación:** uno a muchos (1:N)
- **Representación NoSQL:** afiliadold referenciado en la colección terapias.

➤ **Médico – Orden**

- Un médico puede emitir muchas órdenes.
- **Tipo de relación:** uno a muchos (1:N)
- **Representación NoSQL:** medicold referenciado en la colección órdenes.

➤ **Médico – Agendamiento**

- Un médico puede atender múltiples agendamientos.
- **Tipo de relación:** uno a muchos (1:N)
- **Representación NoSQL:** medicold referenciado en la colección agendamientos.

➤ **Médico – IPS**

- Un médico puede estar adscrito a varias IPS y una IPS puede tener varios médicos.
  - **Tipo de relación:** muchos a muchos (N:M)
  - **Representación NoSQL:** Lista ipsIds referenciada en la colección medicos.
- **Médico – ServicioSalud**
- Un médico puede estar asociado a varios servicios de salud.
  - **Tipo de relación:** uno a muchos (1:N)
  - **Representación NoSQL:** medicId referenciado en la colección serviciosSalud.
- **IPS – ServicioSalud**
- Una IPS puede ofrecer varios servicios de salud y un servicio puede ser ofrecido por varias IPS.
  - **Tipo de relación:** muchos a muchos (N:M)
  - **Representación NoSQL:** Lista ipsIds referenciada en la colección serviciosSalud.
- **Orden – ServicioSalud**
- Una orden incluye un servicio de salud específico como parte del detalleOrden.
  - **Tipo de relación:** uno a uno (1:1)
  - **Representación NoSQL:** Servicio embebido dentro de detalleOrden en la colección órdenes.
- **Agendamiento – ServicioSalud**
- Cada agendamiento está vinculado a un único servicio de salud.
  - **Tipo de relación:** uno a uno (1:1)
  - **Representación NoSQL:** Servicio embebido directamente en el documento agendamiento.

Karin Guerrero – 202317306

Samuel Ayala -

Afiliado – Orden:

Guideline Name	Question	Embed	Reference
Simplicity	¿Mantener la información junta simplifica el modelo de datos y el código?	No	Yes
Go Together	¿Las piezas de información tienen una relación "contiene" o "tiene un"?	No	Yes
Query Atomicity	¿La aplicación consulta las piezas de información juntas?	No	Yes
Update Complexity	¿La información se actualiza junta?	No	Yes
Archival	¿La información se archiva al mismo tiempo?	No	Yes
Cardinality	¿Hay alta cardinalidad en el lado hijo de la relación?	Yes	No
Data Duplication	¿La duplicación de datos sería un problema?	Yes	No
Document Size	¿El tamaño combinado genera problemas de memoria o ancho de banda?	Yes	No
Document Growth	¿La parte embebida crecerá sin límite?	Yes	No
Workload	¿Se escriben las piezas en momentos diferentes en una carga de trabajo intensa?	Yes	No
Individuality	¿Las piezas pueden existir sin su padre?	Yes	No

**Resultado: Referenciado**

Afiliado – Agendamiento:

Guideline Name	Question	Embed	Reference
<b>Simplicity</b>	¿Mantener la información junta simplifica el modelo de datos y el código?	No	Yes
<b>Go Together</b>	¿Las piezas de información tienen una relación "contiene" o "tiene un"?	No	Yes
<b>Query Atomicity</b>	¿La aplicación consulta las piezas de información juntas?	No	Yes
<b>Update Complexity</b>	¿La información se actualiza junta?	No	Yes
<b>Archival</b>	¿La información se archiva al mismo tiempo?	No	Yes
<b>Cardinality</b>	¿Hay alta cardinalidad en el lado hijo de la relación?	Yes	No
<b>Data Duplication</b>	¿La duplicación de datos sería un problema?	Yes	No
<b>Document Size</b>	¿El tamaño combinado genera problemas de memoria o ancho de banda?	Yes	No
<b>Document Growth</b>	¿La parte embebida crecerá sin límite?	Yes	No
<b>Workload</b>	¿Se escriben las piezas en momentos diferentes en una carga de trabajo intensa?	No	No
<b>Individuality</b>	¿Las piezas pueden existir sin su padre?	Yes	No

**Resultado: Referenciado**

**Médico – Orden**

Guideline Name	Question	Embed	Reference
----------------	----------	-------	-----------



<b>Simplicity</b>	¿Mantener la información junta simplifica el modelo de datos y el código?	No	Yes
<b>Go Together</b>	¿Las piezas de información tienen una relación "contiene" o "tiene un"?	No	Yes
<b>Query Atomicity</b>	¿La aplicación consulta las piezas de información juntas?	No	Yes
<b>Update Complexity</b>	¿La información se actualiza junta?	No	Yes
<b>Archival</b>	¿La información se archiva al mismo tiempo?	No	Yes
<b>Cardinality</b>	¿Hay alta cardinalidad en el lado hijo de la relación?	Yes	No
<b>Data Duplication</b>	¿La duplicación de datos sería un problema?	Yes	No
<b>Document Size</b>	¿El tamaño combinado genera problemas de memoria o ancho de banda?	Yes	No
<b>Document Growth</b>	¿La parte embebida crecerá sin límite?	Yes	No
<b>Workload</b>	¿Se escriben las piezas en momentos diferentes en una carga de trabajo intensa?	Yes	No
<b>Individuality</b>	¿Las piezas pueden existir sin su padre?	Yes	No

**Resultado: Referenciado**

#### **Médico – Agendamiento**

<b>Guideline Name</b>	<b>Question</b>	<b>Embed</b>	<b>Reference</b>
Simplicity	¿Mantener la información junta	No	Sí

	simplifica el modelo de datos y el código?		
Go Together	¿Las piezas de información tienen una relación "contiene" o "tiene un"?	No	Sí
Query Atomicity	¿La aplicación consulta las piezas de información juntas?	No	Sí
Update Complexity	¿La información se actualiza junta?	No	Sí
Archival	¿La información se archiva al mismo tiempo?	No	Sí
Cardinality	¿Hay alta cardinalidad en el lado hijo de la relación?	Sí	No
Data Duplication	¿La duplicación de datos sería un problema?	Sí	No
Document Size	¿El tamaño combinado genera problemas de memoria o ancho de banda?	Sí	No
Document Growth	¿La parte embebida crecerá sin límite?	Sí	No
Workload	¿Se escriben las piezas en momentos diferentes en una carga de trabajo intensa?	Sí	No
Individuality	¿Las piezas pueden existir sin su padre?	Sí	No

**Resultado: Referenciado****Médico – IPS**

Guideline Name	Question	Embed	Reference
Simplicity	¿Mantener la información junta simplifica el modelo	No	Sí

	de datos y el código?		
Go Together	¿Las piezas de información tienen una relación "contiene" o "tiene un"?	No	Sí
Query Atomicity	¿La aplicación consulta las piezas de información juntas?	No	Sí
Update Complexity	¿La información se actualiza junta?	No	Sí
Archival	¿La información se archiva al mismo tiempo?	No	Sí
Cardinality	¿Hay alta cardinalidad en el lado hijo de la relación?	Sí	No
Data Duplication	¿La duplicación de datos sería un problema?	Sí	No
Document Size	¿El tamaño combinado genera problemas de memoria o ancho de banda?	Sí	No
Document Growth	¿La parte embebida crecerá sin límite?	Sí	No
Workload	¿Se escriben las piezas en momentos diferentes en una carga de trabajo intensa?	Sí	No
Individuality	¿Las piezas pueden existir sin su padre?	Sí	No

**Resultado: Referenciado**

#### Médico – ServicioSalud

Guideline Name	Question	Embed	Reference
Simplicity	¿Mantener la información junta simplifica el modelo de datos y el código?	No	Sí

Go Together	¿Las piezas de información tienen una relación "contiene" o "tiene un"?	No	Sí
Query Atomicity	¿La aplicación consulta las piezas de información juntas?	No	Sí
Update Complexity	¿La información se actualiza junta?	No	Sí
Archival	¿La información se archiva al mismo tiempo?	No	Sí
Cardinality	¿Hay alta cardinalidad en el lado hijo de la relación?	Sí	No
Data Duplication	¿La duplicación de datos sería un problema?	Sí	No
Document Size	¿El tamaño combinado genera problemas de memoria o ancho de banda?	Sí	No
Document Growth	¿La parte embebida crecerá sin límite?	Sí	No
Workload	¿Se escriben las piezas en momentos diferentes en una carga de trabajo intensa?	Sí	No
Individuality	¿Las piezas pueden existir sin su padre?	Sí	No

**Resultado: Referenciado**

#### IPS – ServicioSalud

Guideline Name	Question	Embed	Reference
Simplicity	¿Mantener la información junta simplifica el modelo de datos y el código?	No	Sí
Go Together	¿Las piezas de información tienen	No	Sí

	una relación "contiene" o "tiene un"?		
Query Atomicity	¿La aplicación consulta las piezas de información juntas?	No	Sí
Update Complexity	¿La información se actualiza junta?	No	Sí
Archival	¿La información se archiva al mismo tiempo?	No	Sí
Cardinality	¿Hay alta cardinalidad en el lado hijo de la relación?	Sí	No
Data Duplication	¿La duplicación de datos sería un problema?	Sí	No
Document Size	¿El tamaño combinado genera problemas de memoria o ancho de banda?	Sí	No
Document Growth	¿La parte embebida crecerá sin límite?	Sí	No
Workload	¿Se escriben las piezas en momentos diferentes en una carga de trabajo intensa?	Sí	No
Individuality	¿Las piezas pueden existir sin su padre?	Sí	No

### Resultado: Referenciado

### Orden – ServicioSalud

Guideline Name	Question	Embed	Reference
Simplicity	¿Mantener la información junta simplifica el modelo de datos y el código?	Yes	No
Go Together	¿Las piezas de información tienen una relación	Yes	No

	"contiene" o "tiene un"?		
Query Atomicity	¿La aplicación consulta las piezas de información juntas?	Yes	No
Update Complexity	¿La información se actualiza junta?	Yes	No
Archival	¿La información se archiva al mismo tiempo?	Yes	No
Cardinality	¿Hay alta cardinalidad en el lado hijo de la relación?	No	Yes
Data Duplication	¿La duplicación de datos sería un problema?	No	Yes
Document Size	¿El tamaño combinado genera problemas de memoria o ancho de banda?	No	Yes
Document Growth	¿La parte embebida crecerá sin límite?	No	Yes
Workload	¿Se escriben las piezas en momentos diferentes en una carga de trabajo intensa?	No	Yes
Individuality	¿Las piezas pueden existir sin su padre?	No	Yes

**Resultado: Embebido**

#### Agendamiento- ServicioSalud

Guideline Name	Question	Embed	Reference
Simplicity	¿Mantener la información junta simplifica el modelo de datos y el código?	Yes	No
Go Together	¿Las piezas de información tienen una relación "contiene" o "tiene un"?	Yes	No

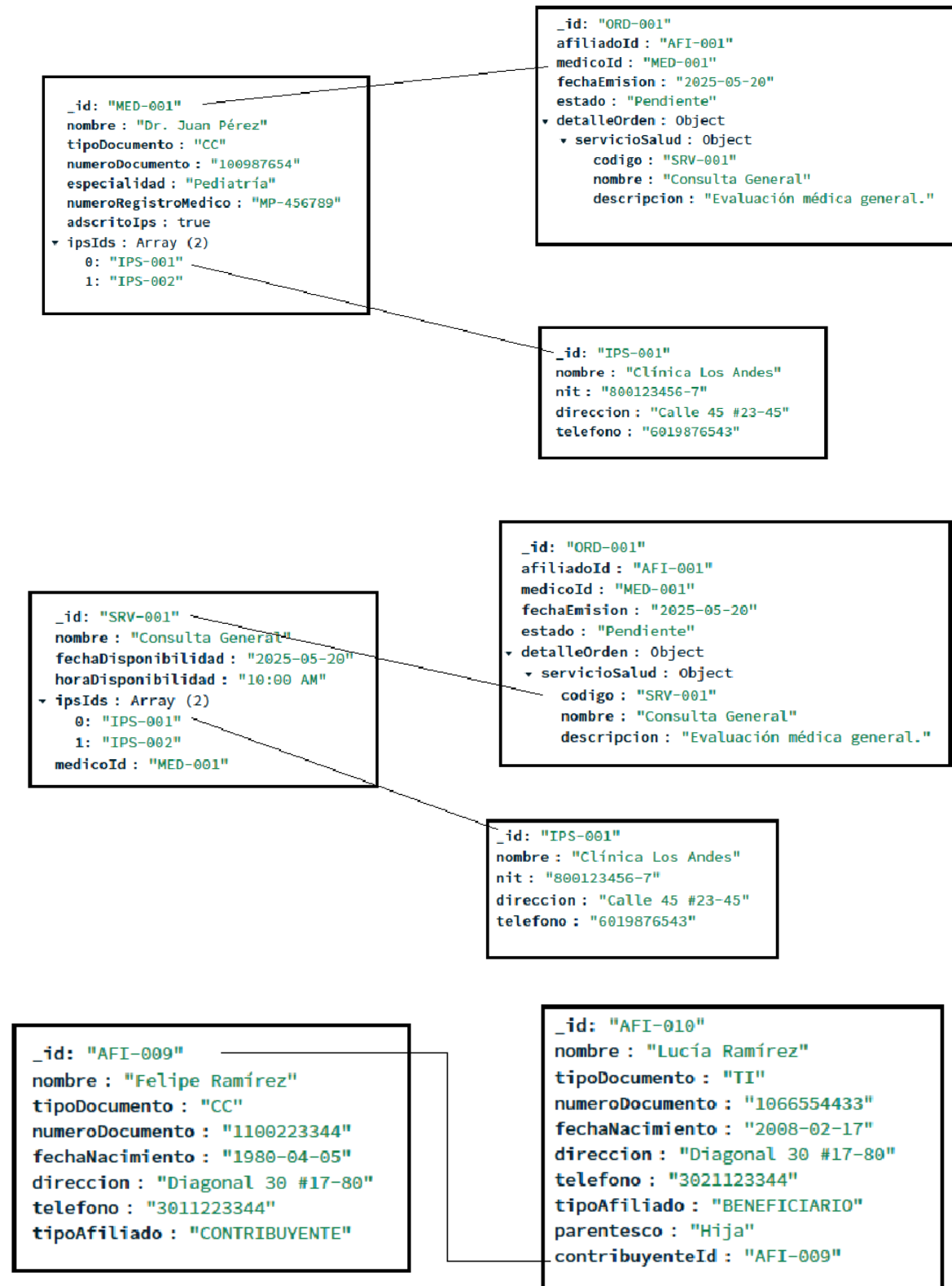
Query Atomicity	¿La aplicación consulta las piezas de información juntas?	Yes	No
Update Complexity	¿La información se actualiza junta?	Yes	No
Archival	¿La información se archiva al mismo tiempo?	Yes	No
Cardinality	¿Hay alta cardinalidad en el lado hijo de la relación?	No	Yes
Data Duplication	¿La duplicación de datos sería un problema?	No	Yes
Document Size	¿El tamaño combinado genera problemas de memoria o ancho de banda?	No	Yes
Document Growth	¿La parte embebida crecerá sin límite?	No	Yes
Workload	¿Se escriben las piezas en momentos diferentes en una carga de trabajo intensa?	No	Yes
Individuality	¿Las piezas pueden existir sin su padre?	No	Yes

### Resultado: Embebido

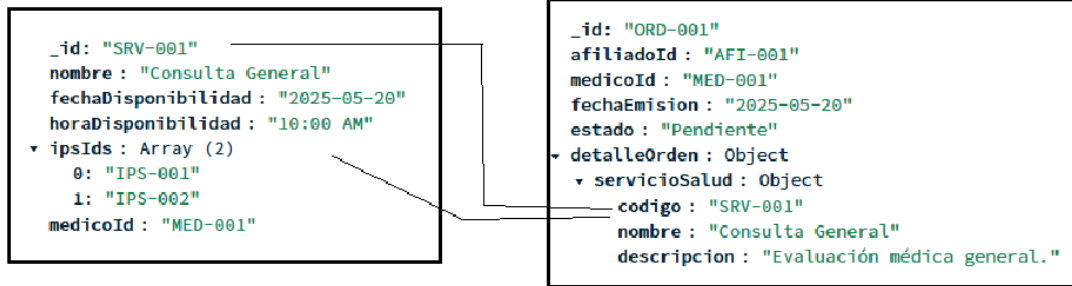
d.

```
_id: "TER-004"  
tipoTerapia : "Psicología"  
numeroSesiones : 12  
afiliadoId : "AFI-007"
```

```
_id: "AFI-007"  
nombre : "Sofia Gómez"  
tipoDocumento : "CC"  
numeroDocumento : "1002456789"  
fechaNacimiento : "1992-10-20"  
direccion : "Calle 50 #12-34"  
telefono : "3001234567"
```







## PUNTO 4

### Poblado a través de postman

The screenshot displays two consecutive POST requests in Postman, both successful (201 Created).

**Request 1: Crear un nuevo afiliado**  
URL: `http://localhost:8080/afiliados/new/save`  
Method: POST  
Body (JSON):

```
{
  "id": "123",
  "nombre": "Ana Pérez",
  "tipoDocumento": "CC",
  "numeroDocumento": "1020304050",
  "fechaNacimiento": "1999-05-10"
}
```

  
Response (Raw):

```
1 Afiliado creado exitosamente
```

**Request 2: Crear un nuevo médico**  
URL: `http://localhost:8080/medicos/new/save`  
Method: POST  
Body (JSON):

```
{
  "id": "123456789",
  "nombre": "Laura Gómez",
  "numeroDocumento": "987654321",
  "especialidad": "Pediatria",
  "adscritoIps": true,
  "ips": {
    "id": "ips001",
  }
}
```

  
Response (Raw):

```
1 Médico creado exitosamente
```

# Karin Guerrero – 202317306

## Samuel Ayala -

POST

http://localhost:8080/ips/new/save

Send

ParamsAuthorizationHeaders (9)BodyScriptsSettings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {
2   "id": "1",
3   "nombre": "Clínica Central",
4   "direccion": "Cra 10 #20-30",
5   "telefono": "123456789",
6   "nit": "900123456"
7 }
```

BodyCookiesHeaders (5)Test Results

201 Created • 121 ms • 192 B • Save Response

Raw

Preview

Visualize

1 IPS creada exitosamente