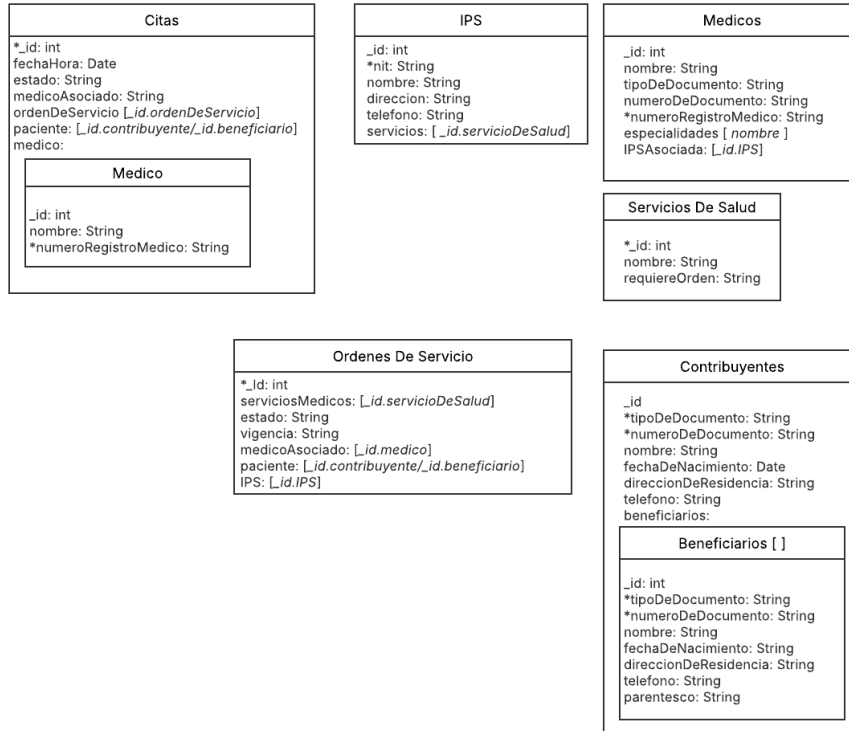


Documentación Entrega 3 Grupo 2

Diseño y construcción de la aplicación:

Modelo conceptual UML:



Cuantificación de entidades

<u>Entidad</u>	<u>Registros estimados</u>
Servicios de Salud	8 tipos de servicios definidos en el enunciado
IPS	Hasta 10000 instituciones prestadoras
Medicos	Entre 1000 y 9000 profesionales registrados
Afiliados	Entre 900000 y 10000000 contribuyente/beneficiarios
Ordenes de Servicio	aproximadamente 90000 ordenes generadas por mes
Citas	aproximadamente 90000 citas reservadas por mes

Descripción de entidades, relaciones entre entidades y su cardinalidad:

IPS

Atributos:

- nit: String (PK)
- nombre: String
- direccion: String

- telefono: String

Descripción:

- La IPS es una institución prestadora de salud la cual ofrece servicios de salud a afiliados de la EPS y contrata médicos.

Relaciones y cardinalidad:

- Una IPS puede contratar uno o varios médicos, o ninguno, y un médico puede estar vinculado a una o varias IPS, o a ninguna.
- Una IPS puede ofrecer uno o más servicios de salud, y cada servicio de salud puede ser brindado por una o varias IPS, o no estar asociado a ninguna.
- Una IPS puede gestionar múltiples citas, pero cada cita debe realizarse en una única IPS.

ServicioDeSalud

Atributos:

- idServicio: int (PK)
- nombre: String
- requiereOrden: Boolean

Descripción:

- Define un tipo de servicio médico digamos consulta general.

Relaciones y cardinalidad:

- Un servicio de salud puede ser ofrecido por múltiples IPS, y una IPS puede ofrecer múltiples servicios de salud.
- Un servicio de salud puede estar asociado a múltiples órdenes de servicio, pero cada orden de servicio está relacionada con un único servicio de salud.

Médico

Atributos:

- numeroRegistroMedico: String (PK)
- nombre: String
- tipoDocumento: String
- especialidad: String

Descripción:

- Representa al profesional que prescribe órdenes médicas y atiende a los pacientes.

Relaciones y cardinalidad:

- La ya mencionada con IPS (Un médico puede trabajar en múltiples IPS, y una IPS puede contratar múltiples médicos.)
- Un médico puede prescribir múltiples órdenes de servicio, pero cada orden de servicio es firmada por un único médico.
- Un médico puede atender múltiples citas, y una cita puede involucrar a varios médicos.

Contribuyente

Atributos:

- tipoDocumento : String. (PK)
- numeroDocumento : String. (PK)
- Nombre : String.
- fechaDeNacimiento : Date.
- direccionDeResidencia : String.

- teléfono : String.

Descripción:

- Afiliado que paga aportes al sistema de salud.

Relaciones y cardinalidad:

- Un contribuyente puede tener múltiples beneficiarios, pero cada beneficiario sólo puede tener un contribuyente.
- Un contribuyente puede agendar múltiples citas, pero cada cita está asociada a un único contribuyente.
- Un contribuyente puede tener múltiples órdenes de servicio, pero cada orden de servicio debe pertenecer a un único contribuyente.
-

Beneficiario

Atributos:

- tipoDocumento : String. (PK)
- numeroDocumento : String. (PK)
- Nombre : String.
- fechaDeNacimiento : Date.
- direccionDeResidencia : String.
- teléfono : String
- parentesco: String

Descripción:

- Afiliado dependiente de un contribuyente.

Relaciones y cardinalidad:

- La relación ya mencionada de beneficiario con contribuyente (Un beneficiario tiene un único contribuyente, mientras que un contribuyente puede tener varios beneficiarios).
- Un beneficiario puede tener múltiples citas, pero cada cita pertenece a un único beneficiario.
- Un beneficiario puede tener múltiples órdenes de servicio, pero cada orden de servicio está asociada a un único beneficiario.

OrdenSeServicio

Atributos:

- IdOrden: String (PK)
- servicioMedico: String
- estado: CK[Vigente, cancelada, completada]
- vigencia: Boolean

Descripción:

- Autorización emitida por un médico para que un afiliado reciba un servicio de salud.

Relaciones y cardinalidad:

- La relación de médico y ordenDeServicio ya explicada (Cada orden de servicio es prescrita por un único médico.)
- La relación de servicioDeSalud y ordenDeServicio ya explicada (Cada orden de servicio está asociada a un único servicio de salud.)
- Cada orden de servicio pertenece a un único afiliado (puede ser un contribuyente o un beneficiario).
- Cada orden de servicio está asociada exactamente a una única cita.

Cita

Atributos:

- idCita: int (PK)
- fechaHora: DateTime
- estado: CK[agendada, cancelada, realizada]

Descripción:

- Representa la reserva o agendamiento de un servicio en una fecha y hora concretas. También permite registrar la prestación.

Relaciones y cardinalidad:

- Cada cita se realiza en una única IPS
- Una cita puede involucrar a múltiples médicos, y un médico puede atender múltiples citas.
- Cada cita está asociada a una única orden de servicio.
- Cada cita pertenece a un único afiliado (puede ser un contribuyente o un beneficiario).

Selección del esquema de asociación:

Relación Contribuyente - Beneficiario: embebida

Justificación:

Aunque la relación es de uno a muchos, cada beneficiario tiene un único contribuyente, y este no suele cambiar ni necesita consultarse por separado. Se acceden y archivan juntos, las actualizaciones son poco frecuentes, y el tamaño del contribuyente es pequeño, lo que no afecta el rendimiento. La duplicación de datos es aceptable y se compensa con un modelo más simple y consultas más atómicas. Según los criterios de la tabla del anexo C del enunciado, se justifica que el contribuyente sea embebido dentro del beneficiario.

Relación IPS - Médico: referenciada

Justificación:

Aunque una IPS puede mostrar la lista de médicos que atienden allí, cada médico también trabaja en otras IPS y se consulta o actualiza por separado. Si los incrustamos dentro de cada IPS, duplicaríamos su información, los documentos crecerían sin control y cualquier cambio obligaría a editar muchas copias. Según la tabla del anexo C, esta relación debe manejarse con referencias: la IPS guarda solo los *id* de los médicos y cada médico vive en su propio documento.

Relación Cita - Médico: embebida

Justificación:

Se embeben los datos básicos del médico (como nombre y registro) para conservar el histórico de la atención. No se requiere mantener una relación viva con el médico; la información es estática y no se actualiza luego de realizada la cita. El volumen de datos embebidos es bajo y permite trazabilidad completa del evento clínico.

Relación IPS - Servicio de Salud: referenciada

Justificación:

Un servicio de salud puede estar disponible en muchas IPS. Esta alta cardinalidad, junto con el hecho de que los servicios no dependen directamente de la IPS (y se pueden consultar o modificar por

separado), hace que la referencia sea más adecuada. Referenciar evita duplicaciones y facilita la reutilización de servicios comunes como “consulta general”.

Relación Orden de servicio - Contribuyente: referenciada

Justificación:

Aunque puede usarse información del contribuyente, este no cambia con cada orden. Su relación es contextual pero no exclusiva, y se mantiene independiente. Dado que múltiples órdenes pueden asociarse a un mismo contribuyente (alta cardinalidad), embebido no sería adecuado. La referencia permite escalabilidad y control de datos.

Relación Orden de servicio - Beneficiario: referenciada

Justificación:

Un beneficiario puede tener muchas órdenes de servicio. Esta alta cardinalidad, junto con la necesidad de acceder al beneficiario por separado, justifica su tratamiento como entidad independiente. No siempre se consultan ni se actualizan juntas, por lo que referenciar reduce duplicación y mejora mantenibilidad.

Relación Orden de servicio - Médico: referenciada

Justificación:

Un médico puede generar muchas órdenes de servicio. Dado que tiene vida y administración propia, es más limpio y escalable usar referencias. No se actualizan juntas, y la carga de trabajo es independiente. Referenciar evita replicar datos médicos en cada orden.

Relación Orden de servicio - IPS: referenciada

Justificación:

Una IPS puede emitir muchas órdenes, lo que implica alta cardinalidad. Además, la IPS tiene su propia estructura de datos y funcionalidades. No se actualizan juntas y se consultan por separado en la mayoría de los casos. Referenciar permite eficiencia, integridad y modularidad.

Relación Orden de servicio - Servicio de Salud: referenciada

Justificación:

Los servicios son reutilizables en múltiples órdenes. Actualizar o consultar servicios es una operación independiente de la orden. La alta cardinalidad y la necesidad de evitar duplicación justifican el uso de referencia.

Relación Orden de servicio - Cita: referenciada

Justificación:

Una orden puede generar muchas citas (alta cardinalidad), y las citas son entidades que se actualizan y archivan de forma independiente (estado: agendada, realizada, cancelada). Además, se consultan frecuentemente sin requerir toda la orden. Referenciar permite una mejor administración de estos ciclos separados.

ANEXO A

Entidad	Operación	Información Necesaria	Tipo
---------	-----------	-----------------------	------

MEDICOS	Obtener lista de medicos	_id, nombre, especialidades, IPSAsociada	Read
MEDICOS	Obtener detalle de un médico	Todos los campos de la colección MEDICOS	Read
MEDICOS	Agregar/actualizar médico	nombre, tipoDeDocumento, numeroDeDocumento, numeroRegistroMedico, especialidades, IPSAsociada	Write
CONTRIBUYENTES	Obtener lista de contribuyentes	_id, nombre, tipoDeDocumento, numeroDeDocumento	Read
CONTRIBUYENTES	Obtener detalle de un contribuyente	Todos los campos de la colección CONTRIBUYENTES	Read
CONTRIBUYENTES	Agregar/actualizar contribuyente	nombre, tipoDeDocumento, numeroDeDocumento, fechaDeNacimiento, direccionDeResidencia, telefono, beneficiarios	Write
ORDENES_DE_SERVICIO	Obtener lista de órdenes	_id, serviciosMedicos, vigencia, estado	Read
ORDENES_DE_SERVICIO	Obtener detalle de una orden	Todos los campos de la colección ORDENES_DE_SERVICIO	Read
ORDENES_DE_SERVICIO	Crear orden de servicio	serviciosMedicos, vigencia, medicoAsociado, paciente, IPS	Write
ORDENES_DE_SERVICIO	Actualizar estado de orden	_id, estado	Write
CITAS	Obtener lista de citas	_id, fechaHora, estado, medicoAsociado, paciente	Read
CITAS	Obtener detalle de una cita	Todos los campos de la colección CITAS	Read
CITAS	Agendar cita	fechaHora, medicoAsociado, ordenDeServicio,	Write

		paciente	
CITAS	Actualizar estado de cita	_id, estado	Write

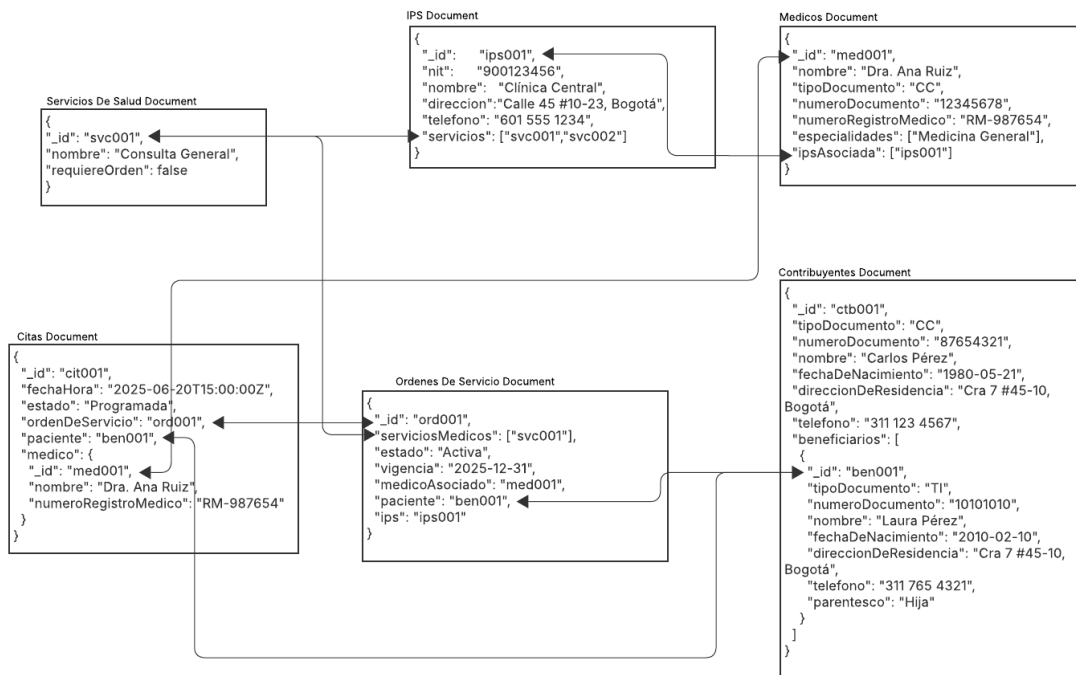
ANEXO B

Es la misma tabla de arriba pero ahora tiene un campo adicional que es la tasa estimada por segundo

Entidad	Operación	Información Necesaria	Tipo	Tasa estimada
MEDICOS	Obtener lista de medicos	_id, nombre, especialidades, IPSAsociada	Read	20 req/s
MEDICOS	Obtener detalle de un médico	Todos los campos de la colección MEDICOS	Read	5 req/s
MEDICOS	Agregar/actualizar médico	nombre, tipoDeDocumento, numeroDeDocumento, numeroRegistroMedico, especialidades, IPSAsociada	Write	1 req/h
CONTRIBUYENTES	Obtener lista de contribuyentes	_id, nombre, tipoDeDocumento, numeroDeDocumento	Read	50 req/s
CONTRIBUYENTES	Obtener detalle de un contribuyente	Todos los campos de la colección CONTRIBUYENTES	Read	20 req/s
CONTRIBUYENTES	Agregar/actualizar contribuyente	nombre, tipoDeDocumento, numeroDeDocumento, fechaDeNacimiento, direccionDeResidencia, telefono, beneficiarios	Write	5 req/h
ORDENES_DE_SERVICIO	Obtener lista de órdenes	_id, serviciosMedicos, vigencia, estado	Read	100 req/s
ORDENES_DE_SERVICIO	Obtener detalle de	Todos los campos	Read	50 req/s

RVICIO	una orden	de la colección ORDENES_DE_SE RVICIO		
ORDENES_DE_SE RVICIO	Crear orden de servicio	serviciosMedicos, vigencia, medicoAsociado, paciente, IPS	Write	10 req/s
ORDENES_DE_SE RVICIO	Actualizar estado de orden	_id, estado	Write	20 req/s
CITAS	Obtener lista de citas	_id, fechaHora, estado, medicoAsociado, paciente	Read	200 req/s
CITAS	Obtener detalle de una cita	Todos los campos de la colección CITAS	Read	100 req/s
CITAS	Agendar cita	fechaHora, medicoAsociado, ordenDeServicio, paciente	Write	5 req/s
CITAS	Actualizar estado de cita	_id, estado	Write	10 req/s

Ejemplo gráfico del esquema de asociación:



Creacion de colecciones con el esquema de validación:

```

db.createCollection("CITAS", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["_id", "fechaHora", "estado", "medicoAsociado", "ordenDeServicio", "paciente", "medico"],
      properties: {
        _id: { bsonType: "int" },
        fechaHora: { bsonType: "date" },
        estado: { bsonType: "string" },
        medicoAsociado: { bsonType: "string" },
        ordenDeServicio: { bsonType: "int" },
        paciente: {
          bsonType: "object",
          required: ["_id", "tipo"],
          properties: {
            _id: { bsonType: "int" },
            tipo: { bsonType: "string" }
          }
        },
        medico: {
          bsonType: "object",
          required: ["_id", "nombre", "numeroRegistroMedico"],
          properties: {
            _id: { bsonType: "int" },

```

```

        nombre: { bsonType: "string" },
        numeroRegistroMedico: { bsonType: "string" }
    }
}
}
}
}
});

```

```

ddb.createCollection("CONTRIBUYENTES", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: [ "_id", "tipoDeDocumento", "numeroDeDocumento", "nombre", "fechaDeNacimiento",
"direccionDeResidencia", "telefono", "beneficiarios"],
      properties: {
        _id: { bsonType: "string" },
        tipoDeDocumento: { bsonType: "string" },
        numeroDeDocumento: { bsonType: "string" },
        nombre: { bsonType: "string" },
        fechaDeNacimiento: { bsonType: "date" },
        direccionDeResidencia: { bsonType: "string" },
        telefono: { bsonType: "string" },
        beneficiarios: {
          bsonType: "array",
          items: {
            bsonType: "object",
            required: [ "_id", "tipoDeDocumento", "numeroDeDocumento", "nombre", "fechaDeNacimiento",
"direccionDeResidencia", "telefono", "parentesco"],
            properties: {
              _id: { bsonType: "string" },
              tipoDeDocumento: { bsonType: "string" },
              numeroDeDocumento: { bsonType: "string" },
              nombre: { bsonType: "string" },
              fechaDeNacimiento: { bsonType: "date" },
              direccionDeResidencia: { bsonType: "string" },
              telefono: { bsonType: "string" },
              parentesco: { bsonType: "string" }
            }
          }
        }
      }
    }
  }
});

```

```

db.createCollection("IPS", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["_id", "nit", "nombre", "direccion", "telefono", "servicios"],
      properties: {
        _id: { bsonType: "int" },
        nit: { bsonType: "string" },
        nombre: { bsonType: "string" },
        direccion: { bsonType: "string" },
        telefono: { bsonType: "string" },
        servicios: {
          bsonType: "array",
          items: { bsonType: "int" }
        }
      }
    }
  }
});

```

```

db.createCollection("MEDICOS", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["_id", "nombre", "tipoDeDocumento", "numeroDeDocumento", "numeroRegistroMedico",
"especialidades", "IPSAsociada"],
      properties: {
        _id: { bsonType: "int" },
        nombre: { bsonType: "string" },
        tipoDeDocumento: { bsonType: "string" },
        numeroDeDocumento: { bsonType: "string" },
        numeroRegistroMedico: { bsonType: "string" },
        especialidades: {
          bsonType: "array",
          items: { bsonType: "string" }
        },
        IPSAsociada: {
          bsonType: "array",
          items: { bsonType: "int" }
        }
      }
    }
  }
});

```

```

db.createCollection("ORDENES_DE_SERVICIO", {
  validator: {

```

```

$jsonSchema: {
  bsonType: "object",
  required: ["_id", "serviciosMedicos", "estado", "vigencia", "medicoAsociado", "paciente", "IPS"],
  properties: {
    _id: { bsonType: "int" },
    serviciosMedicos: {
      bsonType: "array",
      items: { bsonType: "int" }
    },
    estado: { bsonType: "string" },
    vigencia: { bsonType: "string" },
    medicoAsociado: { bsonType: "int" },
    paciente: {
      bsonType: "object",
      required: ["_id", "tipo"],
      properties: {
        _id: { bsonType: "int" },
        tipo: { bsonType: "string" }
      }
    },
    IPS: {
      bsonType: "array",
      items: { bsonType: "int" }
    }
  }
}
});

```

```

db.createCollection("SERVICIOS_DE_SALUD", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required: ["_id", "nombre", "requiereOrden"],
      properties: {
        _id: { bsonType: "int" },
        nombre: { bsonType: "string" },
        requiereOrden: { bsonType: "string" }
      }
    }
  }
});

```

Inserción de datos:

```

db.CITAS.insertMany([
  {

```

```

    _id: 1,
    fechaHora: new Date("2025-06-01T10:00:00Z"),
    estado: "Agendada",
    medicoAsociado: "12345",
    ordenDeServicio: 1,
    paciente: { _id: 1, tipo: "contribuyente" },
    medico: { _id: 1, nombre: "Dr. Carlos Pérez", numeroRegistroMedico: "REG123" }
  },
  {
    _id: 2,
    fechaHora: new Date("2025-06-02T09:30:00Z"),
    estado: "Realizada",
    medicoAsociado: "23456",
    ordenDeServicio: 2,
    paciente: { _id: 2, tipo: "beneficiario" },
    medico: { _id: 2, nombre: "Dra. Luisa Martínez", numeroRegistroMedico: "REG124" }
  },
  {
    _id: 3,
    fechaHora: new Date("2025-06-03T11:15:00Z"),
    estado: "Cancelada",
    medicoAsociado: "34567",
    ordenDeServicio: 3,
    paciente: { _id: 3, tipo: "contribuyente" },
    medico: { _id: 3, nombre: "Dr. Felipe Torres", numeroRegistroMedico: "REG125" }
  },
  {
    _id: 4,
    fechaHora: new Date("2025-06-04T14:00:00Z"),
    estado: "Agendada",
    medicoAsociado: "45678",
    ordenDeServicio: 4,
    paciente: { _id: 4, tipo: "beneficiario" },
    medico: { _id: 4, nombre: "Dra. Paula Gómez", numeroRegistroMedico: "REG126" }
  },
  {
    _id: 5,
    fechaHora: new Date("2025-06-05T08:30:00Z"),
    estado: "Realizada",
    medicoAsociado: "56789",
    ordenDeServicio: 5,
    paciente: { _id: 5, tipo: "contribuyente" },
    medico: { _id: 5, nombre: "Dr. Jorge Ruiz", numeroRegistroMedico: "REG127" }
  }
]);

```

```

db.CONTRIBUYENTES.insertMany([
  {

```

```
_id: "1",
tipoDeDocumento: "CC",
numeroDeDocumento: "10000001",
nombre: "Ana Gómez",
fechaDeNacimiento: new Date("1985-04-15"),
direccionDeResidencia: "Calle 10 #20-30",
telefono: "3201234567",
beneficiarios: [
  {
    _id: "2",
    tipoDeDocumento: "TI",
    numeroDeDocumento: "20000002",
    nombre: "Juan Gómez",
    fechaDeNacimiento: new Date("2010-08-21"),
    direccionDeResidencia: "Calle 10 #20-30",
    telefono: "3201234567",
    parentesco: "Hijo"
  }
],
{
  _id: "3",
  tipoDeDocumento: "CC",
  numeroDeDocumento: "10000003",
  nombre: "Pedro Rodríguez",
  fechaDeNacimiento: new Date("1978-01-11"),
  direccionDeResidencia: "Carrera 5 #22-44",
  telefono: "3106543210",
  beneficiarios: [
    {
      _id: "4",
      tipoDeDocumento: "TI",
      numeroDeDocumento: "20000004",
      nombre: "Camila Rodríguez",
      fechaDeNacimiento: new Date("2012-03-09"),
      direccionDeResidencia: "Carrera 5 #22-44",
      telefono: "3106543210",
      parentesco: "Hija"
    }
  ]
},
{
  _id: "5",
  tipoDeDocumento: "CC",
  numeroDeDocumento: "10000005",
  nombre: "Lucía Pérez",
  fechaDeNacimiento: new Date("1990-09-23"),
  direccionDeResidencia: "Avenida 68 #30-15",
```

```

telefono: "3118889988",
beneficiarios: []
},
{
  _id: "6",
  tipoDeDocumento: "CC",
  numeroDeDocumento: "10000006",
  nombre: "Marcos Jiménez",
  fechaDeNacimiento: new Date("1982-06-30"),
  direccionDeResidencia: "Diagonal 60 #11-99",
  telefono: "3129991112",
  beneficiarios: [
    {
      _id: "7",
      tipoDeDocumento: "TI",
      numeroDeDocumento: "20000007",
      nombre: "Diego Jiménez",
      fechaDeNacimiento: new Date("2013-02-01"),
      direccionDeResidencia: "Diagonal 60 #11-99",
      telefono: "3129991112",
      parentesco: "Hijo"
    },
    {
      _id: "8",
      tipoDeDocumento: "TI",
      numeroDeDocumento: "20000008",
      nombre: "Sofía Jiménez",
      fechaDeNacimiento: new Date("2015-07-12"),
      direccionDeResidencia: "Diagonal 60 #11-99",
      telefono: "3129991112",
      parentesco: "Hija"
    }
  ]
},
{
  _id: "9",
  tipoDeDocumento: "CC",
  numeroDeDocumento: "10000009",
  nombre: "Elena Ruiz",
  fechaDeNacimiento: new Date("1995-12-05"),
  direccionDeResidencia: "Transversal 45 #100-10",
  telefono: "3152345678",
  beneficiarios: []
}
]);

```

```
db.IPS.insertMany([
```

```

{
  _id: 1,
  nit: "900123456",
  nombre: "IPS Salud Total",
  direccion: "Carrera 50 #15-30",
  telefono: "6012345678",
  servicios: [1, 2]
},
{
  _id: 2,
  nit: "900654321",
  nombre: "IPS Vida Sana",
  direccion: "Calle 77 #10-25",
  telefono: "6019876543",
  servicios: [2, 3]
},
{
  _id: 3,
  nit: "900112233",
  nombre: "IPS Integral",
  direccion: "Avenida 1 #2-3",
  telefono: "6013332222",
  servicios: [1, 3, 4]
},
{
  _id: 4,
  nit: "900445566",
  nombre: "IPS Bienestar",
  direccion: "Carrera 99 #77-66",
  telefono: "6014445555",
  servicios: [4, 5]
},
{
  _id: 5,
  nit: "900778899",
  nombre: "IPS Sanar",
  direccion: "Calle 15 #55-55",
  telefono: "6016667777",
  servicios: [2, 5]
}
]);

```

```

db.MEDICOS.insertMany([
{
  _id: 1,
  nombre: "Dr. Carlos Pérez",
  tipoDeDocumento: "CC",
  numeroDeDocumento: "10203040",

```



```

    numeroRegistroMedico: "REG123",
    especialidades: ["Medicina General"],
    IPSAsociada: [1]
  },
  {
    _id: 2,
    nombre: "Dra. Luisa Martínez",
    tipoDeDocumento: "CC",
    numeroDeDocumento: "20406080",
    numeroRegistroMedico: "REG124",
    especialidades: ["Pediatria"],
    IPSAsociada: [2]
  },
  {
    _id: 3,
    nombre: "Dr. Felipe Torres",
    tipoDeDocumento: "CC",
    numeroDeDocumento: "12312312",
    numeroRegistroMedico: "REG125",
    especialidades: ["Radiología"],
    IPSAsociada: [3]
  },
  {
    _id: 4,
    nombre: "Dra. Paula Gómez",
    tipoDeDocumento: "CC",
    numeroDeDocumento: "98765432",
    numeroRegistroMedico: "REG126",
    especialidades: ["Odontología"],
    IPSAsociada: [4]
  },
  {
    _id: 5,
    nombre: "Dr. Jorge Ruiz",
    tipoDeDocumento: "CC",
    numeroDeDocumento: "13579135",
    numeroRegistroMedico: "REG127",
    especialidades: ["Cardiología"],
    IPSAsociada: [5]
  }
]);

db.ORDENES_DE_SERVICIO.insertMany([
  {
    _id: 1,
    serviciosMedicos: [1, 2],
    estado: "Activa",
    vigencia: "2025-06-30",
  }
]);

```

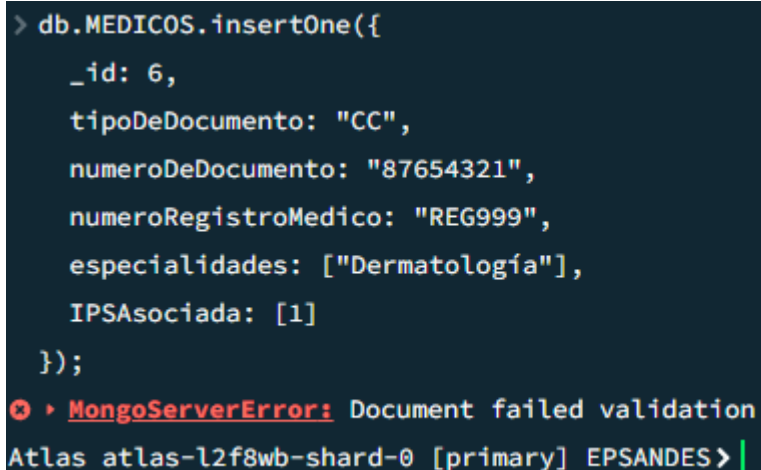
```
    medicoAsociado: 1,
    paciente: { _id: 1, tipo: "contribuyente" },
    IPS: [1]
  },
  {
    _id: 2,
    serviciosMedicos: [2, 3],
    estado: "Finalizada",
    vigencia: "2025-05-15",
    medicoAsociado: 2,
    paciente: { _id: 2, tipo: "beneficiario" },
    IPS: [2]
  },
  {
    _id: 3,
    serviciosMedicos: [3, 4],
    estado: "Cancelada",
    vigencia: "2025-07-10",
    medicoAsociado: 3,
    paciente: { _id: 3, tipo: "contribuyente" },
    IPS: [3]
  },
  {
    _id: 4,
    serviciosMedicos: [4, 5],
    estado: "Activa",
    vigencia: "2025-06-15",
    medicoAsociado: 4,
    paciente: { _id: 4, tipo: "beneficiario" },
    IPS: [4]
  },
  {
    _id: 5,
    serviciosMedicos: [5, 1],
    estado: "Finalizada",
    vigencia: "2025-06-20",
    medicoAsociado: 5,
    paciente: { _id: 5, tipo: "contribuyente" },
    IPS: [5]
  }
];
```

```
db.SERVICIOS_DE_SALUD.insertMany([
  { _id: 1, nombre: "Consulta General", requiereOrden: "No" },
  { _id: 2, nombre: "Rayos X", requiereOrden: "Sí" },
  { _id: 3, nombre: "Exámenes de sangre", requiereOrden: "Sí" },
  { _id: 4, nombre: "Odontología", requiereOrden: "No" },
  { _id: 5, nombre: "Cardiología", requiereOrden: "Sí" }
]);
```

Escenarios de Prueba

RF4 – MEDICOS: falta campo 'nombre'

```
db.MEDICOS.insertOne({
  _id: 6,
  tipoDeDocumento: "CC",
  numeroDeDocumento: "87654321",
  numeroRegistroMedico: "REG999",
  especialidades: ["Dermatología"],
  IPSAsociada: [1]
});
```



```
> db.MEDICOS.insertOne({
  _id: 6,
  tipoDeDocumento: "CC",
  numeroDeDocumento: "87654321",
  numeroRegistroMedico: "REG999",
  especialidades: ["Dermatología"],
  IPSAsociada: [1]
});
❌ ▶ MongoServerError: Document failed validation
Atlas atlas-l2f8wb-shard-0 [primary] EPSANDES> |
```

RF5 – CONTRIBUYENTES: falta campo 'tipoDeDocumento'

```
db.CONTRIBUYENTES.insertOne({
  _id: "10",
  numeroDeDocumento: "30000010",
  nombre: "Pedro Test",
  fechaDeNacimiento: new Date("1990-01-01"),
  direccionDeResidencia: "Calle Test",
  telefono: "3001234567",
```

```
beneficiarios: []
});
```

```
> db.CONTRIBUYENTES.insertOne({
  _id: "10",
  numeroDeDocumento: "30000010",
  nombre: "Pedro Test",
  fechaDeNacimiento: new Date("1990-01-01"),
  direccionDeResidencia: "Calle Test",
  telefono: "3001234567",
  beneficiarios: []
});
MongoServerError: Document failed validation
Atlas atlas-l2f8wb-shard-0 [primary] EPSANDES>|
```

RF6 – ORDENES_DE_SERVICIO: falta campo 'estado'

```
db.ORDENES_DE_SERVICIO.insertOne({
  _id: 6,
  serviciosMedicos: [1, 2],
  vigencia: "2025-12-31",
  medicoAsociado: 1,
  paciente: { _id: 1, tipo: "contribuyente" },
  IPS: [1]
});
```

```
> db.ORDENES_DE_SERVICIO.insertOne({
  _id: 6,
  serviciosMedicos: [1, 2],
  vigencia: "2025-12-31",
  medicoAsociado: 1,
  paciente: { _id: 1, tipo: "contribuyente" },
  IPS: [1]
});
MongoServerError: Document failed validation
```

RF7 – CITAS: tipo incorrecto en 'fechaHora' (string en vez de date)

```
db.CITAS.insertOne({
  _id: 6,
  fechaHora: "2025-07-01T10:00:00Z",
  estado: "Agendada",
  medicoAsociado: "12345",
  ordenDeServicio: 1,
  paciente: { _id: 1, tipo: "contribuyente" },
```

```
medico: { _id: 1, nombre: "Dr. Test", numeroRegistroMedico: "REG001" }
});
```

```
> db.CITAS.insertOne({
  _id: 6,
  fechaHora: "2025-07-01T10:00:00Z",
  estado: "Agendada",
  medicoAsociado: "12345",
  ordenDeServicio: 1,
  paciente: { _id: 1, tipo: "contribuyente" },
  medico: { _id: 1, nombre: "Dr. Test", numeroRegistroMedico: "REG001" }
});
MongoServerError: Document failed validation
Atlas atlas-l2f8wb-shard-0 [primary] EPSANDES> |
```

RFC1

Disponibilidad del Servicio

Nombre del Servicio	Fecha y Hora de Disponibilidad	Nombre de la IPS	Nombre del Médico
Rayos X	01/06/2025 05:00	IPS Salud Total	Dr. Carlos Pérez
Rayos X	02/06/2025 04:30	IPS Vida Sana	Dra. Luisa Martínez

[Volver a Servicios](#)

RFC2

Consultar Top 20 Servicios Más Solicitados

Fecha Inicio (YYYY-MM-DD):

1999-12-12

Fecha Fin (YYYY-MM-DD):

2030-12-12

[Volver](#)

[Consultar](#)

Top 20 Servicios Más Solicitados

Nombre del Servicio	Cantidad de Solicitudes
Odontología	2
reumatología	2
Rayos X	2
Cardiología	2
Exámenes de sangre	2

[Nueva Consulta](#)[Volver](#)