

Integrantes:

Josef Alf rez Garc a – 202317078 , Martin Riveira – 202321727, Xiongyu chen wu – 202116588

ServicioSalud

| codigo | nombre | descripcion |
|--------|--------|-------------|
| PK     | NN     | NN          |
| int    | string | string      |

2.1. Modelo Relacional:

Afiliado

| numDoc | tipoDoc | nombre | fechaNacimiento | direccion | telefono | tipoAfiliado | parentesco | contribuyente      |
|--------|---------|--------|-----------------|-----------|----------|--------------|------------|--------------------|
| PK     | NN,CK   | NN     | NN              | NN        | NN       | NN           | NN         | FK(numDocAfiliado) |
| int    | string  | string | datetime()      | string    | int      | string       | string     | int                |

Medico

| registroMedico | tipoDoc | nombre | numDocumento | idEspecialidad |
|----------------|---------|--------|--------------|----------------|
| PK             | NN      | NN     | NN           | NN             |
| int            | string  | string | int          | int            |

Disponibilidad

| idDisponibilidad | fechaHora  | RegistroMedico | nitlps  | codigoServicio | idCita   | estadoDisponibilidad |
|------------------|------------|----------------|---------|----------------|----------|----------------------|
| PK               | NN         | FK(m dico)     | FK(lps) | FK(int)        | FK(cita) | NN                   |
| int              | datetime() | string         | int     | int            | int      | bool                 |

lps

| nitlps | nombrelps | direcci n | telefono | idEps   |
|--------|-----------|-----------|----------|---------|
| PK     | NN        | NN        | NN       | FK(Eps) |
| int    | string    | string    | string   | int     |

Cita

| idCita | fechaHora  | numDoc       | idDisponibilidad   | nitlps  |
|--------|------------|--------------|--------------------|---------|
| PK     | NN         | FK(Afiliado) | FK(Disponibilidad) | FK(lps) |
| int    | datetime() | int          | int                | int     |

Eps

| idEps | nombreEps |
|-------|-----------|
| PK    | NN        |
| int   | string    |

OrdenServicio

| idOrdenServicio | fechaHora  | idEstadoOrden      | numDoc       | RegistroMedico | idCita   | codigo            |
|-----------------|------------|--------------------|--------------|----------------|----------|-------------------|
| PK              | NN         | NN,FK(estadoOrden) | FK(Afiliado) | FK(medico)     | FK(cita) | FK(ServicioSalud) |
| int             | datetime() | int                | int          | int            | int      | int               |

Medico-lps

| RegistroMedico | nitlps     |
|----------------|------------|
| PK,FK(medico)  | PK,FK(lps) |
| int            | int        |

lps-ServicioSalud

| nitlps     | codigo               |
|------------|----------------------|
| PK,FK(lps) | PK,FK(ServicioSalud) |
| int        | int                  |

## Integrantes:

Josef Alf rez Garc a – 202317078 , Martin Riveira – 202321727, Xiongyu chen wu – 202116588

## 2.2 Nivel de Normalizaci n:

- Luego de analizar el diagrama llegamos a la conclusi n de que, seg n nuestra idea del c mo se van a usar las PK a la hora de implementarlo, el diagrama relacional est  correctamente normalizado en nivel 3FN BC.  nicamente identificamos un posible riesgo en la relaci n recursiva que se presenta en la tabla de Afiliado con el atributo de contribuyente para la relaci n Contribuyente-Beneficiario(1...0\*), all  puede presentarse una transitividad si no se relaciona cuidadosamente.
- Relaciones detalladas para Normalizaci n:

## 1. ServiciosSalud

### Formas normales

1. 1NF:
  - Los atributos son at micos (cadenas de texto y un entero para la PK).
  - No hay repetici n de grupos de atributos.
  - Cumple 1NF.
2. 2NF:
  - La clave primaria es simple (codigo), por lo que no hay dependencia parcial posible.
  - Cumple 2NF.
3. 3NF:
  - Revisa si hay dependencia entre atributos no clave, por ejemplo, nombre → descripci n o viceversa.
  - Normalmente, “nombre” y “descripci n” son independientes.
  - Cumple 3NF.
4. BCNF:
  - La  nica dependencia funcional real es codigo → (nombre, descripci n).
  - codigo es la clave primaria (y clave candidata).
  - Cumple BCNF.

## Integrantes:

Josef Alf rez Garc a – 202317078 , Martin Riveira – 202321727, Xiongyu chen wu – 202116588

---

## 2. Afiliado

### Formas normales

1. 1NF:
  - Todos los campos son at micos (strings, fechas, etc.).
  - Cumple 1NF.
2. 2NF:
  - La PK es simple (numDoc), as  que no hay dependencia parcial.
  - Cumple 2NF.
3. 3NF:
  - Se debe verificar que ning n atributo no clave dependa de otro atributo no clave.
  - Para caso de la FK recursiva, al hacer referencia a un solo Id PK no tiene inconveniente, m s a n si tomamos en cuenta que el ser contribuyente es principalmente un dato que a la hora de restricciones o l gica, no tiene mucho peso.
  - Cumple 3NF.
4. BCNF:
  - La  nica determinante “fuerte” es la PK numDoc.

## Integrantes:

Josef Alf rez Garc a – 202317078 , Martin Riveira – 202321727, Xiongyu chen wu – 202116588

---

## 3. Medico

### Formas normales

1. 1NF:
    - o Todos los atributos son at micos.
    - o Cumple 1NF.
  2. 2NF:
    - o PK simple (regMedico), no hay dependencia parcial.
    - o Cumple 2NF.
  3. 3NF:
    - o Verificar si “regMedico” → “numDocumento” es la  nica dependencia y si “numDocumento” no determina nada m s.
    - o Normalmente, “numDocumento” del m dico es solo un dato adicional, decidimos tomar como PK al registro m dico por lo que el numDocumento no tiene mucho peso en l gica etc.
    - o Cumple 3NF.
  4. BCNF:
    - o De nuevo, la PK regMedico determina los dem s campos.
    - o No hay dependencia donde el determinante sea un atributo no clave.
    - o Cumple BCNF.
-

## Integrantes:

Josef Alf rez Garc a – 202317078 , Martin Riveira – 202321727, Xiongyu chen wu – 202116588

## 4. Disponibilidad

### Formas normales

1. 1NF:
    - Atributos at micos.
    - Cumple 1NF.
  2. 2NF:
    - PK simple (idDisponibilidad), no hay dependencia parcial.
    - Cumple 2NF.
  3. 3NF:
    - Se asume que “fecha” dependen directamente de idDisponibilidad.
    - No hay un atributo no clave que dependa de otro atributo no clave.
    - Cumple 3NF.
  4. BCNF:
    - La  nica dependencia es  $\text{idDisponibilidad} \rightarrow (\text{fecha}, \text{codigoServicio}, \text{estadoDisponibilidad}, \text{regMedico})$ .
    - Cumple BCNF.
- 

## 5. ips

### Formas normales

1. 1NF:
  - Atributos at micos (strings, enteros).
  - Cumple 1NF.
2. 2NF:
  - PK simple (idIps), no hay dependencia parcial.
  - Cumple 2NF.
3. 3NF:
  - No hay se al de que “nombreIps”, “direccionIps” dependan de “fkEps” (u otro atributo no clave).
  - Cumple 3NF.
4. BCNF:
  - La dependencia principal es  $\text{idIps} \rightarrow (\text{nombreIps}, \text{direccionIps}, \text{telefonoIps}, \text{fkEps})$ .
  - Cumple BCNF.

## Integrantes:

Josef Alf rez Garc a – 202317078 , Martin Riveira – 202321727, Xiongyu chen wu – 202116588

---

## 6. Eps :Formas normales

1. 1NF:
    -   Atributos at micos.
    -   Cumple 1NF.
  2. 2NF:
    -   PK simple, sin dependencia parcial.
    -   Cumple 2NF.
  3. 3NF:
    -   No hay otro atributo no clave que genere dependencia transitiva.
    -   Cumple 3NF.
  4. BCNF:
    -   numEps determina nombreEps.
    -   Cumple BCNF.
- 

## 7. Cita

### Formas normales

1. 1NF:
  -   Campos at micos.
  -   Cumple 1NF.
2. 2NF:
  -   PK simple (idCita), no hay dependencia parcial.
  -   Cumple 2NF.
3. 3NF:
  -   Se asume que no hay dependencia “numDocAfilado  $\rightarrow$  regMedico”, porque cada afiliado puede tener muchos medicos
  -   Cumple 3NF.
4. BCNF:
  -   De nuevo, la dependencia principal es idCita  $\rightarrow$  (fechaCita, horaCita, numDocAfilado, regMedico).
  -   Cumple BCNF.

**Integrantes:**

Josef Alf  rez Garc  a – 202317078 , Martin Riveira – 202321727, Xiongyu chen wu – 202116588

---

## 8. OrdenServicio

### Formas normales

1. 1NF:
  -   Atributos at  micos.
  -   Cumple 1NF.
2. 2NF:
  -   PK simple (idOrden), no hay dependencia parcial.
  -   Cumple 2NF.
3. 3NF:
  -   No se observa dependencia transitiva entre los atributos no clave ( numDocAfiliado no determina regMedico).
  -   Cumple 3NF.
4. BCNF:
  -   idOrden determina los dem  s atributos.
  -   Cumple BCNF.

## Integrantes:

Josef Alf  rez Garc  a – 202317078 , Martin Riveira – 202321727, Xiongyu chen wu – 202116588

---

## 9. Medico-lps (tabla puente)

### Estructura

- PK (compuesta): regMedico, idlps

### Formas normales

1. 1NF:
  - Atributos at  micos (ambos FKs).
  - Cumple 1NF.
2. 2NF:
  - La PK es compuesta: (regMedico, idlps).
  - Sin hay otros atributos, no hay riesgo de dependencia parcial.
  - Cumple 2NF.
3. 3NF:
  - No hay otros atributos no clave que dependan de parte de la PK.
  - Cumple 3NF.
4. BCNF:
  - La   nica dependencia funcional es (regMedico, idlps) → (nada m  s).
  - Cumple BCNF.



**Integrantes:**

Josef Alf  rez Garc  a – 202317078 , Martin Riveira – 202321727, Xiongyu chen wu – 202116588

---

## 10. Ips-ServiciosSalud (tabla puente)

### Formas normales

1. 1NF:
  -   Atributos at  micos (FKs).
  -   Cumple 1NF.
2. 2NF:
  -   PK compuesta, pero no hay otros atributos que puedan depender parcialmente de un componente de la PK.
  -   Cumple 2NF.
3. 3NF:
  -   No hay atributos no clave adicionales.
  -   Cumple 3NF.
4. BCNF:
  -   La   nica dependencia funcional real es (idIps, pkServicioSalud) → (nada m  s).
  -   Cumple BCNF.

**Integrantes:**

Josef Alf rez Garc a – 202317078 , Martin Riveira – 202321727, Xiongyu chen wu – 202116588

### 3. Escenarios de prueba:

**Pruebas de unicidad de tuplas**

Insertar una tupla en una tabla con una clave primaria (PK) conocida.

Intentar insertar otra tupla con la misma PK y verificar que el sistema rechaza la operaci n.

Intentar actualizar una PK existente a un valor ya utilizado y confirmar que la operaci n es rechazada.

**Pruebas de integridad referencial con claves for neas (FK)**

Insertar una tupla con una FK que existe en la tabla referenciada y verificar que se almacena correctamente.

Intentar insertar una tupla con una FK que no existe en la tabla referenciada y confirmar que la operaci n es rechazada.

Eliminar una tupla referenciada por una FK y verificar que la eliminaci n es rechazada o en cascada, seg n el modelo.

**Pruebas de restricciones de chequeo**

Intentar insertar tuplas que violen restricciones de chequeo, como valores fuera de un rango permitido.

Verificar que el sistema rechaza correctamente estas inserciones.

Intentar actualizar un valor a otro que viole una restricci n de chequeo y confirmar que es rechazado.

**Pruebas de los requerimientos funcionales (RF)**

Asegurar que los RFs principales del sistema funcionan correctamente:

**Integrantes:**

Josef Alf rez Garc a – 202317078 , Martin Riveira – 202321727, Xiongyu chen wu – 202116588

**RF1: Registrar IPS**

Insertar una nueva IPS con todos sus atributos y verificar que los datos se almacenan correctamente.

Intentar registrar una IPS con un NIT duplicado y verificar que es rechazado.

**RF2: Registrar un servicio de salud**

Crear un nuevo servicio de salud y verificar que aparece en la base de datos.

Intentar registrar un servicio sin un nombre y verificar que es rechazado.

**RF3: Asignar un servicio de salud a una IPS**

Asociar un servicio de salud a una IPS existente y verificar que la relaci n se ha creado correctamente.

Intentar asignar un servicio a una IPS inexistente y verificar que la operaci n es rechazada.

**RF4: Registrar m dico**

Insertar un m dico con sus datos personales y su especialidad.

Verificar que se puede asociar a una IPS.

Intentar registrar un m dico sin especialidad y confirmar que es rechazado.

**RF5: Registrar afiliado**

Crear un afiliado y confirmar que se almacenan sus datos personales y tipo de afiliaci n (contribuyente o beneficiario).

Intentar registrar un afiliado sin n mero de documento y verificar que la operaci n es rechazada.

**Integrantes:**

Josef Alférez García – 202317078 , Martin Riveira – 202321727, Xiongyu chen wu – 202116588

**RF6: Registrar orden de servicio**

Crear una orden de servicio con un médico y un afiliado válidos.

Intentar registrar una orden sin un médico válido y verificar que es rechazada.

Intentar registrar una orden con una fecha en el pasado y verificar que es rechazada.

**RF7: Agendar un servicio de salud**

Consultar la agenda de un servicio disponible.

Intentar agendar una cita sin una orden de servicio cuando es requerida.

Intentar agendar una cita en un horario en el que el médico no tiene disponibilidad y verificar que es rechazada.

**RF8: Registrar la prestación de un servicio**

Registrar la atención de un afiliado y verificar que se almacena correctamente.

Intentar registrar la prestación de un servicio sin una cita previa y verificar que es rechazada.

**Pruebas de consultas estadísticas (RFC)**

RFC1: Consultar la disponibilidad de servicios en las siguientes 4 semanas y verificar que los datos son correctos.

RFC2: Mostrar los 20 servicios más solicitados en un periodo determinado.

RFC3: Calcular el índice de uso de cada servicio.

RFC4: Consultar la utilización de servicios por un afiliado en un periodo determinado.

RFC5: Consultar el número de citas canceladas en el último mes y verificar que los datos son correctos.

RFC6: Obtener un reporte de médicos con mayor número de consultas atendidas y verificar la exactitud de los datos.