Camilo Linares – c.linares1

Cristian Cortes – cc.cortesm1

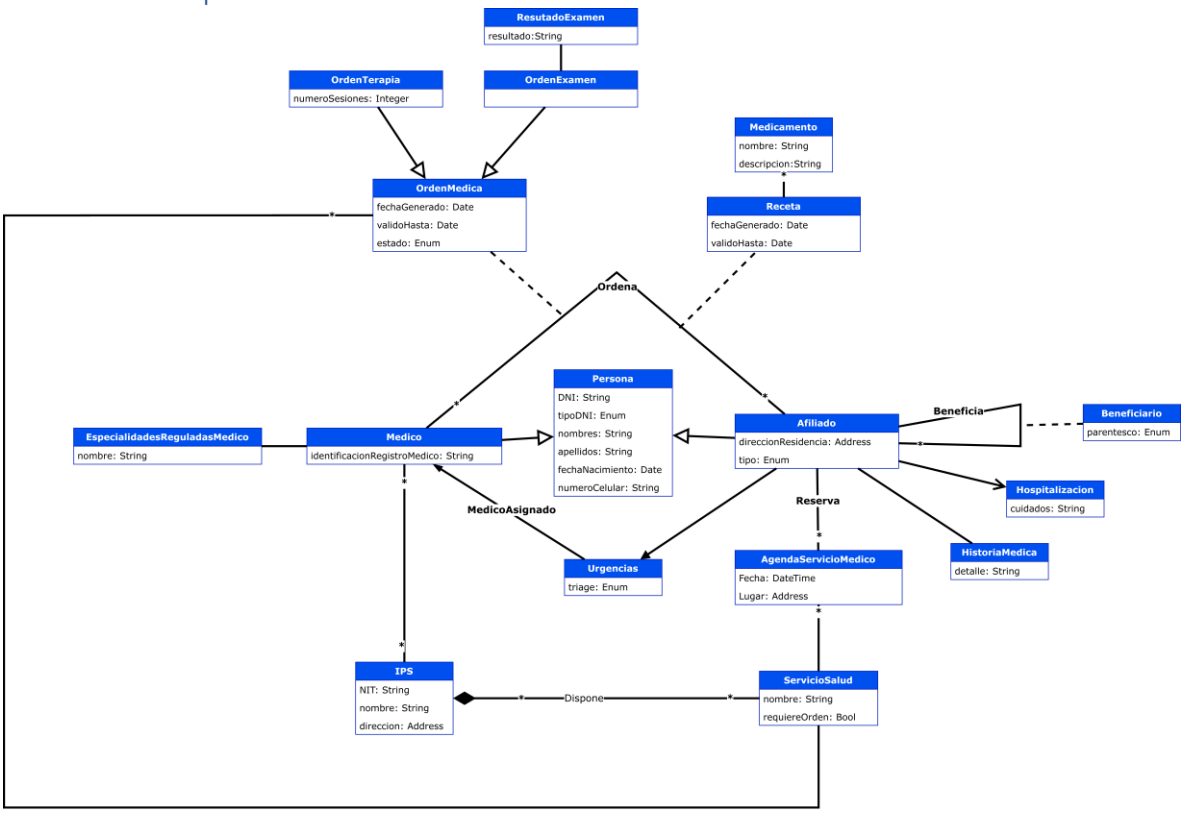
Isaac Bermudez – i.bermudezl

Proyecto EPS Andes Etapa 1

Caso de estudio

El caso del proyecto consta de una EPS la cual presta atención a sus afiliados tercerizando la prestación de servicios a las IPS. Dentro de lo más importante cabe destacar el manejo de cada una de estas instituciones las cuales prestan el servicio junto al de los médicos y pacientes. Donde las tareas más recurrentes van a ser las peticiones que posea un afiliado frente a una cuestión médica, ya sea sacar una cita a medico general, ser remitido a medico con especializada, recibir medicamentos, etc. En torno a esto rondará las reglas de negocio las cuales se planteará su diseño para una posterior implementación en este documento.

Modelo conceptual



Modelo Entidad Relación

Modelo Relacional

Justificación

Herencia entre clases (Generalización/Especialización)

En UML, Persona tiene subclases como Afiliado y Medico. En el modelo relacional, las bases de datos no manejan herencia de forma nativa, por lo que hay que optar por una estrategia como

usar la cual llamamos Personas, y subtablas Afiliados y Medicos en las cuales guardamos los atributos extras que tiene cada subclase.

Relaciones N:M (Muchos a Muchos)

En el Relacional, estas relaciones de muchos a muchos se modelan con tablas intermedias entre las relaciones, como por ejemplo la relación entre IPS y Medicos tiene la relacion MedicosIPS la cual representaria una tabla en el modelo de base de datos.

Enumeraciones (Enum)

UML permite definir atributos como tipoDNI, estado en OrdenMedica, y triage en Urgencias como Enum. En el modelo Relaiconal simplemente usamos un campo de string con valores limitados.

Agregaciones y Composiciones

En UML, relaciones fuertes como Afiliado -> Beneficiario se expresan con composición. En el modelo relacional, esto se modela con una llave foránea con restricción de eliminación en cascada.

Fechas y Periodos de Validez

validoHasta en Receta y OrdenMedica indica que ciertos registros tienen validez temporal.

En SQL, puede ser necesario agregar restricciones de validación o triggers para gestionar estos períodos, además de implementar restricciones para evitar que se venzan antes de que fueran gestionados.

1. IPS
La entidad IPS en el UML se definió con sus atributos identificadores y descriptivos. En el modelo relacional se refleja como la tabla IPS, donde el atributo NIT actúa como clave primaria, y los atributos Nombre y Dirección se convierten en columnas obligatorias. La conversión se realizó de forma directa, respetando la unicidad y la obligatoriedad definidas en el diagrama.

2. Medicos
En el UML, la clase Medicos se relaciona con la clase Persona mediante una asociación de herencia o identidad, y posee atributos propios (como Identificación de Registro Médico y Especialidad, donde esta última es una referencia a la entidad EspecialidadesReguladasMedicos). En la tabla Medicos, la clave compuesta (DNI y TipoDNI) – que es también clave foránea a Persona – se conserva, y se incluyen las columnas IdentificacionRegistroMedico y Especialidad (con FK a EspecialidadesReguladasMedicos). Esto se obtuvo aplicando el algoritmo modificado de Chen, que mapea la relación IS-A y las asociaciones a través de claves foráneas.

3. EspecialidadesReguladasMedicos
La entidad EspecialidadesReguladasMedicos se muestra en el UML como una clase con su identificador y nombre. En la tabla se convierte directamente en una relación donde Id es la clave primaria y Nombre es el atributo descriptivo, de acuerdo con la conversión de clases independientes.

4. Medicamentos
La clase Medicamentos en el UML se transforma en la tabla Medicamentos, conservando el atributo Id como clave primaria y Nombre como columna obligatoria. La conversión se realizó sin alteraciones, ya que la entidad es básica y no posee asociaciones complejas.

5. Personas
La entidad Persona del UML se convierte en la tabla Personas, con la clave compuesta (DNI, TipoDNI) que identifica de forma única cada registro. Todos los atributos (Nombre, Apellido, FechaDeNacimiento y NumeroCelular) se trasladan a columnas, manteniendo la correspondencia exacta entre el diseño conceptual y el modelo relacional.

6. Afiliados
La clase Afiliados se deriva de Persona (o está asociada a ella) y se distingue mediante atributos propios (como Tipo, y la referencia autorreferencial con DNI_Afiliado y TipoDNI_Afiliado). La tabla Afiliados utiliza la clave compuesta (DNI, TipoDNI) – que es FK a Personas – y agrega los demás atributos, de manera que se respeta la identidad de la clase y las relaciones de dependencia definidas en el UML.

7. ServiciosSalud
La entidad ServiciosSalud se muestra en el diagrama UML con atributos que indican la naturaleza del servicio (Nombre, Tipo, RequiereOrden). En la tabla ServiciosSalud, el atributo Id es la clave primaria y los demás atributos se convierten en columnas obligatorias, siguiendo la conversión directa de una clase independiente.

8. AgendasServiciosMedicos
Esta entidad se define en el UML para representar la programación de los servicios médicos. La tabla AgendasServiciosMedicos conserva el atributo Id como clave primaria y, mediante la aplicación del algoritmo, se incluye ServicioSalud_Id como clave foránea que relaciona la agenda con el servicio, y los atributos Fecha, Lugar, ReservaAfiliado_DNI y ReservaAfiliado_TipoDNI para reflejar la relación de reserva con Afiliados.

9. HistoriasMedicas
La entidad HistoriasMedicas se modela en el UML como la historia clínica asociada a un Afiliado. La tabla HistoriasMedicas utiliza la clave compuesta formada por Afiliado_DNI y Afiliado_TipoDNI – que actúa como FK a Afiliados – y el atributo Historia se convierte en la columna que almacena la información clínica, reflejando la dependencia total con la entidad Afiliados.

10. RecetasMedicas
En el UML, RecetasMedicas se representa con su identificador, fechas (FechaGenerado y ValidoHasta) y las relaciones con Medicos y Afiliados. La tabla RecetasMedicas respeta esta estructura, utilizando Id como clave primaria y mapeando las asociaciones mediante los atributos MedicoPrescribe_DNI, MedicoPescribe_TipoDNI (FK a Medicos) y RecetaAfiliado_DNI, RecetaAfiliado_TipoDNI (FK a Afiliados).

11. MedicamentosRecetas
La asociación muchos a muchos entre RecetasMedicas y Medicamentos, descrita en el UML, se transforma en la tabla intermedia MedicamentosRecetas. Esta tabla utiliza la clave compuesta conformada por Id_Receta y Id_Medicamento, que son claves foráneas a RecetasMedicas y Medicamentos respectivamente, lo que garantiza la correspondencia exacta de la relación M:N.

12. OrdenesMedicas
La entidad OrdenesMedicas se define en el UML para gestionar las órdenes emitidas en el sistema. La tabla OrdenesMedicas conserva Id como clave primaria y refleja las asociaciones con Afiliados (OrdenAfiliado_DNI y OrdenAfiliado_TipoDNI), ServiciosSalud (ServicioSalud_Id) y Medicos (MedicoPrescribe_DNI y MedicoPescribe_TipoDNI), de forma que la estructura y las cardinalidades definidas en el diagrama UML se mantienen.

13. OrdenesTerapia
Esta tabla es resultado de una especialización o detalle adicional de las OrdenesMedicas en el UML. Se mapea como OrdenesTerapia, donde Orden_Id es clave primaria y FK a OrdenesMedicas, y se incluye el atributo NumeroSesiones, reflejando la relación de detalle en la entidad OrdenesMedicas.

14. HospitalizacionesAfiliados
El UML indica que los Afiliados pueden tener hospitalizaciones con ciertos cuidados. La tabla HospitalizacionesAfiliados utiliza la clave compuesta (Afiliado_DNI y Afiliado_TipoDNI), actuando como FK a Afiliados, y contiene el atributo Cuidados, de forma que se refleja la relación directa definida en el modelo conceptual.

15. ExamenesResultados
La entidad ExamenesResultados se representa en el UML para registrar resultados de exámenes asociados a una orden médica. La tabla ExamenesResultados utiliza Orden_Id (FK a OrdenesMedicas) como clave primaria y contiene el atributo Resultados, garantizando la correspondencia con la relación UML.

16. UrgenciasAfiliados
Esta entidad, representada en el UML, indica que se pueden generar registros de urgencias asociados a Afiliados, con la participación de un médico asignado. La tabla UrgenciasAfiliados define Id como clave primaria y mapea las relaciones mediante MedicoAsignado_DNI, MedicoAsignado_TipoDNI (FK a Medicos) y Paciente_DNI, Paciente_TipoDNI (FK a Afiliados).

17. ServiciosSaludIPS
En el UML se modela la relación entre ServiciosSalud e IPS a través de una asociación que se traduce en la tabla ServiciosSaludIPS. Esta tabla utiliza una clave compuesta formada por Id (FK a ServiciosSalud) y NIT (FK a IPS), lo que refleja la relación directa y la correspondencia de la cardinalidad definida.

18. MedicosIPS
Finalmente, la entidad MedicosIPS, que asocia Medicos con IPS, se convierte en la tabla MedicosIPS. Se emplea una clave compuesta de NIT (FK a IPS), Medico_DNI y Medico_TipoDNI (FK a Medicos), manteniendo la integridad referencial y la relación tal como se especifica en el diagrama UML.

Análisis nivel de normalización

Normalización de las tablas

El análisis de normalización aplicado a las tablas verifica el cumplimiento de 1FN, 2FN, 3FN y FNBC, asegurando: atributos atómicos, eliminación de dependencias parciales en claves compuestas y ausencia de dependencias transitivas, garantizando un modelo libre de redundancias y anomalías. A continuación, el análisis por tablas realizado:

1. Tabla IPS

Estructura:

- **NIT:** Clave primaria (SA, PK)
- **Nombre:** Atributo requerido (NN, ND)
- **Dirección:** Atributo requerido (NN)

Análisis:

- **1FN:** Todos los campos son atómicos, es decir, cada celda contiene un único valor.
- **2FN:** Al tratarse de una clave simple (NIT), los atributos *Nombre* y *Dirección* dependen enteramente de ella.
- **3FN y FNBC:** No se identifican dependencias transitivas, ya que la única relación funcional es NIT → {Nombre, Dirección} y, al ser NIT la clave, se cumple la condición de BCNF.

Conclusión: La tabla IPS se encuentra normalizada

2. Tabla Médicos

Estructura:

- **DNI y TipoDNI:** Clave compuesta (además, son FK hacia la tabla Persona)
- **IdentificaciónRegistroMédico:** Atributo requerido (NN, ND)
- **Especialidad:** Clave foránea hacia EspecialidadesReguladasMedico (NN)

Análisis:

- **1FN:** Los datos son atómicos.
- **2FN:** La clave compuesta (DNI, TipoDNI) determina de forma completa los demás campos, por lo que no existen dependencias parciales.
- **3FN y FNBC:** No se detectan dependencias transitivas, ya que la relación funcional (DNI, TipoDNI) → {IdentificaciónRegistroMédico, Especialidad} cumple con las condiciones de BCNF.

Conclusión: La tabla Médicos está normalizada.

3. Tabla EspecialidadesReguladasMedico

Estructura:

- **Id:** Clave primaria (SA, PK)
- **Nombre:** Atributo requerido (NN, ND)

Análisis:

- Se verifica la atomicidad de todos los atributos (1FN).
- Al contar con una clave simple y no existir atributos adicionales, no se presentan dependencias parciales ni transitivas.

Conclusión: La tabla se encuentra normalizada.

4. Tabla Medicamentos

Estructura:

- **Id:** Clave primaria (SA, PK)

- **Nombre:** Atributo requerido (NN, ND)

Análisis:

- Se cumplen los requisitos de 1FN, 2FN y 3FN, al contar con una clave simple y no disponer de otros atributos que puedan generar dependencias.

Conclusión: La tabla Medicamentos está normalizada.

5. Tabla Persona

Estructura:

- **DNI y TipoDNI:** Clave compuesta (ambos SA, PK)
- **Nombre, Apellido, FechaDeNacimiento, NúmeroCelular:** Atributos requeridos (NN)

Análisis:

- Cada uno de los atributos es atómico, cumpliendo la 1FN.
- La clave compuesta (DNI, TipoDNI) determina por completo el resto de los atributos, sin que se detecten dependencias parciales ni transitivas.

Conclusión: La tabla Persona se encuentra normalizada.

6. Tabla Afiliado

Estructura:

- **DNI y TipoDNI:** Clave primaria (además, son FK hacia Persona)
- **Tipo:** Atributo requerido (NN)
- **DNI_Afiliado y TipoDNI_Afiliado:** Claves foráneas que referencian a la misma tabla Afiliado (relación de autorreferencia o “sponsor”)

Análisis:

- Se cumplen los requisitos de 1FN, ya que cada atributo almacena un solo valor.
- La clave compuesta (DNI, TipoDNI) determina de manera completa los atributos *Tipo* y la relación de autorreferencia, garantizando la 2FN.
- No se observan dependencias transitivas adicionales, cumpliéndose la 3FN.

Conclusión: La tabla Afiliado está normalizada.

7. Tabla ServicioSalud

Estructura:

- **Id:** Clave primaria (SA, PK)
- **Nombre:** Atributo requerido (NN, ND)
- **Tipo:** Atributo requerido (NN)
- **RequiereOrden:** Atributo requerido (NN)

Análisis:

- Cada atributo depende directamente de la clave simple *Id*, cumpliendo así con la 1FN, 2FN y 3FN.

Conclusión: La tabla ServicioSalud se encuentra normalizada.

8. Tabla AgendaServicioMedico

Estructura:

- **Id:** Clave primaria (SA, PK)
- **Fecha:** Atributo requerido (NN)
- **Lugar:** Atributo requerido (NN, ND)
- **ServicioSalud_Id:** Clave foránea hacia ServicioSalud.Id (NN)
- **ReservaAfiliado_DNI y ReservaAfiliado_TipoDNI:** Claves foráneas que referencian a Afiliado

Análisis:

- Se verifica la atomicidad de los datos (1FN).
- Al tratarse de una clave simple (Id), se cumple la 2FN.
- No se identifican dependencias transitivas, ya que la información de la reserva depende únicamente del identificador de la agenda, cumpliendo la 3FN.

Conclusión: La tabla AgendaServicioMedico está normalizada.

9. Tabla HistoriaMedica

Estructura:

- **Afiliado_DNI y Afiliado_TipoDNI:** Clave compuesta (SA y FK hacia Afiliado)
- **Historia:** Atributo requerido (NN)

Análisis:

- Todos los atributos son atómicos (1FN).
- La clave compuesta determina el único atributo no clave, cumpliendo la 2FN.
- No existen dependencias transitivas (3FN).

Conclusión: La tabla HistoriaMedica se encuentra normalizada.

10. Tabla RecetaMedica

Estructura:

- **Id:** Clave primaria (SA, PK)
- **FechaGenerado y ValidoHasta:** Atributos requeridos (NN)

- **MedicoPrescribe_DNI y MedicoPescribe_TipoDNI:** Claves foráneas hacia Médicos (NN)
- **RecetaAfiliado_DNI y RecetaAfiliado_TipoDNI:** Claves foráneas hacia Afiliado (NN)

Análisis:

- Se cumplen los requisitos de la 1FN, ya que los valores son atómicos.
- Al disponer de una clave simple, todos los atributos dependen de *Id* (2FN).
- No se aprecian dependencias transitivas, por lo que se satisface la 3FN.

Conclusión: La tabla RecetaMedica está normalizada.

11. Tabla MedicamentosRecetas

Estructura:

- **Id_Receta e Id_Medicamento:** Conforman una clave compuesta, actuando además como claves foráneas hacia RecetaMedica y Medicamentos respectivamente.

Análisis:

- Los atributos son atómicos (1FN).
- Al no existir atributos adicionales a la clave compuesta, se cumplen de forma directa la 2FN y la 3FN.

Conclusión: La tabla MedicamentosRecetas se encuentra normalizada.

12. Tabla OrdenMedica

Estructura:

- **Id:** Clave primaria (SA, PK)
- **FechaGenerado y ValidoHasta:** Atributos requeridos (NN)
- **OrdenAfiliado_DNI y OrdenAfiliado_TipoDNI:** Claves foráneas hacia Afiliado (NN)
- **Estado:** Atributo requerido (NN)
- **ServicioSalud_Id:** Clave foránea hacia ServicioSalud (NN)
- **MedicoPrescribe_DNI y MedicoPescribe_TipoDNI:** Claves foráneas hacia Médicos (NN)

Análisis:

- Cada atributo es atómico (1FN).
- Con la clave simple *Id*, se cumple la 2FN, ya que todos los campos dependen de ella.
- No se detectan dependencias transitivas, lo que asegura la 3FN.

Conclusión: La tabla OrdenMedica está normalizada.

13. Tabla OrdenTerapia

Estructura:

- **Orden_Id:** Clave primaria y clave foránea hacia OrdenMedica.Id
- **NumeroSesiones:** Atributo requerido (NN)

Análisis:

- Se cumple la 1FN, al tener atributos atómicos.
- La única dependencia funcional es Orden_Id → NumeroSesiones, lo que satisface los requisitos de la 2FN y 3FN.

Conclusión: La tabla OrdenTerapia se encuentra normalizada.

14. Tabla HospitalizacionAfiliado

Estructura original:

- **Afiliado_DNI y Afiliado_TipoDNI:** Marcados como SA (PK)
- **Cuidados:** Atributo requerido (NN)

Observación:
Los datos son atómicos por lo tanto cumple 1 FN, y la única dependencia funcional es aquella del afiliado (DNI, TipoDNI -> Cuidados) por ende está en FNBC

15. Tabla ExamenResultado

Estructura:

- **Orden_Id:** Clave primaria (SA y FK hacia OrdenMedica)
- **Resultados:** Atributo requerido (NN)

Análisis:

- Se cumplen los requisitos de 1FN, 2FN y 3FN, ya que la única dependencia es Orden_Id → Resultados.

Conclusión: La tabla ExamenResultado está normalizada.

16. Tabla UrgenciasAfiliado

Estructura:

- **Id:** Clave primaria (SA, PK)
- **Triage:** Atributo requerido (NN)
- **MedicoAsignado_DNI y MedicoAsignado_TipoDNI:** Claves foráneas hacia Médicos (NN)
- **Paciente_DNI y Paciente_TipoDNI:** Claves foráneas hacia Afiliado (NN)

Análisis:

- Los atributos son atómicos (1FN).
- Al contar con una clave simple (*Id*), se cumple la 2FN.
- No se observan dependencias transitivas, cumpliéndose la 3FN.

Conclusión: La tabla UrgenciasAfiliado se encuentra normalizada.

17. Tabla ServicioSaludIPS

Estructura:

- **Id:** Clave primaria (SA, FK hacia ServicioSalud.Id)
- **NIT:** Clave primaria (FK hacia IPS.NIT)

Análisis:

- La combinación (Id, NIT) conforma la clave compuesta y no existen otros atributos en la tabla.

Conclusión: La tabla ServicioSaludIPS está normalizada.

18. Tabla MedicosIPS

Estructura:

- **NIT:** Clave primaria (FK hacia IPS.NIT)
- **Medico_DNI y Medico_TipoDNI:** Conforman una clave compuesta adicional (FK hacia Médicos)

Análisis:

- La clave compuesta (NIT, Medico_DNI, Medico_TipoDNI) abarca la totalidad de la información sin atributos extra, cumpliéndose la 1FN, 2FN y 3FN.

Conclusión: La tabla MedicosIPS se encuentra normalizada.

Resumen Final del Modelo Normalizado

A continuación, se presenta un listado final de las tablas, con sus respectivos campos, tal como quedaron tras el proceso de normalización:

1. **IPS:** (NIT, Nombre, Dirección)
2. **Médicos:** (DNI, TipoDNI, IdentificaciónRegistroMédico, Especialidad)
3. **EspecialidadesReguladasMedico:** (Id, Nombre)
4. **Medicamentos:** (Id, Nombre)
5. **Persona:** (DNI, TipoDNI, Nombre, Apellido, FechaDeNacimiento, NúmeroCelular)
6. **Afiliado:** (DNI, TipoDNI, Tipo, DNI_Afiliado, TipoDNI_Afiliado)
7. **ServicioSalud:** (Id, Nombre, Tipo, RequiereOrden)
8. **AgendaServicioMedico:** (Id, Fecha, Lugar, ServicioSalud_Id, ReservaAfiliado_DNI, ReservaAfiliado_TipoDNI)
9. **HistoriaMedica:** (Afiliado_DNI, Afiliado_TipoDNI, Historia)

10. **RecetaMedica:** (Id, FechaGenerado, ValidoHasta, MedicoPrescribe_DNI, MedicoPescribe_TipoDNI, RecetaAfiliado_DNI, RecetaAfiliado_TipoDNI)
11. **MedicamentosRecetas:** (Id_Receta, Id_Medicamento)
12. **OrdenMedica:** (Id, FechaGenerado, ValidoHasta, OrdenAfiliado_DNI, OrdenAfiliado_TipoDNI, Estado, ServicioSalud_Id, MedicoPrescribe_DNI, MedicoPescribe_TipoDNI)
13. **OrdenTerapia:** (Orden_Id, NumeroSesiones)
14. **HospitalizacionAfiliado (modificada):** (Afiliado_DNI, Afiliado_TipoDNI, Cuidados)
15. **ExamenResultado:** (Orden_Id, Resultados)
16. **UrgenciasAfiliado:** (Id, Triage, MedicoAsignado_DNI, MedicoAsignado_TipoDNI, Paciente_DNI, Paciente_TipoDNI)
17. **ServicioSaludIPS:** (Id, NIT)
18. **MedicosIPS:** (NIT, Medico_DNI, Medico_TipoDNI)

Todas las tablas cumplen con 1FN, 2FN, 3FN y FNBC (determinantes como superclaves), salvo *HospitalizacionAfiliado*, ajustada para garantizar unicidad.

Escenarios de prueba

RF1 – Registrar una IPS

Prueba de unicidad

Insertar una IPS con el NIT 1 y otra con el mismo NIT con diferentes valores. Al momento de hacer esto nos debería soltar un error de violación de unicidad debido a que estamos insertando dos veces a una tabla un dato con la misma llave primaria.

Prueba de integridad

No se pueden insertar dos IPS con el mismo nombre debido a que la regla de ND (Not Duplicated) lo restringe. Junto a que no tendría sentido no tener nombre o dirección por lo que ambas cuentan con la restricción NN.

RF2 – Registrar un servicio de salud

Para registrar un servicio de salud primero es necesario que existan registros de IPS.

Prueba de unicidad

No hay problema con unicidad frente a la llave primaria debido a que es SA. Luego las reglas que valdrán son aquellas que nos confirmen la integridad de la información

Prueba de integridad

Tanto para el nombre del servicio, su tipo y su atributo indicando si requiere orden son no nulos debido a que tener estos datos incompletos no tendría sentido para las normas del negocio. Y su nombre no puede estar duplicado.

Prueba de integridad llaves foráneas

Para las llaves foráneas tenemos IPS las cuales no debe ser nulo. Lo que implica que poner nulo violaría NN. Y una llave foránea inexistente violaría la integridad referencial.

RF3 – Registrar un servicio de salud a una IPS

Esto este ligado al RF anterior, ya que, para registrar un servicio de salud a una IPS, debemos poner a que IPS corresponde en cuanto a las pruebas de integridad de llaves foráneas.

RF4 – Registrar un medico

Para registrar a un médico primero debemos añadir a una persona con sus datos y luego su registro en la tabla Médicos, con su correspondiente especialidad.

Prueba de unicidad

Si tenemos

Medicos			
DNI	TipoDNI	IdentificacionRegistroMedico	Especialidad
PK, FK(Persona.DNI)	PK, FK(Persona.TipoDNI)	ND, NN	FK(EspecialidadesReguladasMedico.Id), NN
1	1	1	1
1	1	x	x

Violaremos la unicidad de la tabla, ya que solo podemos tener registrado a un medico con su correspondiente documento de identidad. Sin embargo, la tabla MedicosIPS nos permite registrarlo a una o varias IPS

Prueba de integridad

Necesitamos una identificación de registro medico la cual esta marcada como no nulo, en caso de no ser así se nos debe marcar como un error.

Prueba de integridad llaves foráneas

En la tabla EspecialidadesReguladasMedico debemos tener las especialidades que nuestra EPS maneja para así registrarlo a un médico.

RF5 – Registrar un afiliado

Prueba de unicidad

Para registrar un afiliado contribuyente primero debemos añadir una persona con sus datos y luego su registro en la tabla Afiliados, con su tipo de Afiliación como contribuyente, para luego dejar los campos DNI_Afiliado y TipoDNI_Afiliado en NULL. Si vamos a agregar a un Beneficiario tocaria repetir este proceso utilizando el tipo beneficiario en vez de contribuyente y llamar a el DNI y el TipoDNI del contribuyente dentro de la misma tabla. Si tratamos de insertar a la misma persona dos veces se generará un error.

Prueba de integridad

Necesitamos un tipo de Afiliado la cual se marca como no nulo y check, en caso de no ser así habrá un error.

Prueba de integridad llaves foráneas

En la tabla Afiliados debemos tener en cuenta que esta tabla se apunta a sí misma, así que debemos tener los registros de los afiliados contribuyentes ya registrados antes de los

beneficiarios, o si no marcara error. Asi mismo cada afiliado debe primero ser una tupla en la tabla personas o se generara un error.

RF6 – Registrar una orden de servicio de salud para un afiliado por parte de un medico

Pruebas de Unicidad:

Para registrar una orden de salud, se puede registrar la misma orden muchas veces dado a que hay un índice generado por el sistema.

Pruebas de integridad:

Si se inserta una tupla con un estado que no está dentro de los valores Vigente, Cancelado o Completo o se agrega una tupla con fecha nula habrá un error.

Pruebas de Integridad FK:

Si se inserta una tupla con un médico que no existe no dejara insertar la tupla dado a que es un foreign key, lo mismo para un afiliado que no existe.

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

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

Pruebas de Unicidad:

Se permite registrar múltiples citas para el mismo afiliado y servicio de salud, ya que el campo Id es generado de manera única por el sistema.

Pruebas de Integridad:

No se puede registrar una cita sin Fecha o Lugar, ya que estos campos son obligatorios. Si se intenta insertar una fecha inválida el sistema debe generar un error.

Pruebas de Integridad FK:

Si ServicioSalud_Id hace referencia a un servicio de salud que no existe en la base de datos, la inserción debe ser rechazada.S i ReservaAfiliado_DNI y ReservaAfiliado_TipoDNI no corresponden a un afiliado registrado, el sistema debe impedir la creación de la cita.

RF8 – Registrar la prestación de un servicio de salud a un afiliado por parte de una IPS

Utilizamos la misma tabla de órdenes para tener una orden completada así que ahí estan las pruebas. Simplemente se registra la orden de servicio de salud como completada.

RFC1 – Consultar la agenda de disponibilidad que tiene un servicio de salud ingresado por el usuario en las siguientes 4 semanas

Utilizamos la tabla *AgendasServiciosMedicos*, filtrando por el código del servicio (ServicioSalud_Id) y el rango de fechas; se une con *ServiciosSalud* para obtener el nombre y demás detalles del servicio, con *ServiciosSaludIPS* y *IPS* para identificar la IPS que lo ofrece, y mediante la relación con

Medicos (y la tabla *Personas*, que contiene el nombre del médico) se obtiene el nombre del profesional asignado. Se realizarán pruebas de integridad referencial en la clave foránea ServicioSalud_Id, pruebas de unicidad en las claves primarias de las tablas involucradas y restricciones de chequeo en los formatos de fecha para asegurar que los horarios se encuentren dentro del rango de 4 semanas.

RFC2 – Mostrar los 20 servicios más solicitados

Agrupamos y cuentamos los registros de la tabla *OrdenesMedicas* por ServicioSalud_Id, filtrando por el rango de fechas indicado; estos datos se unen con la tabla *ServiciosSalud* para obtener el nombre de cada servicio, se ordenan de forma descendente según la cantidad de solicitudes y se limita el resultado a 20 registros. Se implementan pruebas de integridad referencial para asegurar que cada ServicioSalud_Id en *OrdenesMedicas* exista en *ServiciosSalud*, además de pruebas de unicidad en las claves primarias y chequeos de formato en los campos de fecha.

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

Contamos la cantidad de registros de disponibilidad en *AgendasServiciosMedicos* y la cantidad de solicitudes en *OrdenesMedicas* para el período indicado; el índice se obtiene dividiendo el total de servicios disponibles por el total de servicios usados para cada servicio, y estos resultados se unen con *ServiciosSalud* para identificar el nombre del servicio. Se aplican pruebas de integridad referencial y de unicidad, y se definen restricciones de chequeo que aseguren la correcta validación de las fechas y eviten divisiones por cero.

RFC4 – Mostrar la utilización de servicios de EPS Andes por un afiliado dado y en un periodo dado

Filtramos los registros de *OrdenesMedicas* según el identificador del afiliado (OrdenAfiliado_DNI y OrdenAfiliado_TipoDNI) y las fechas de la orden; luego se unen con *ServiciosSalud* para obtener el nombre del servicio, con *Medicos* (y *Personas*) para obtener el nombre del médico que atendió, y con *ServiciosSaludIPS* y *IPS* para conocer la IPS que ofreció el servicio. Se realizan pruebas de integridad referencial para asegurar que el identificador del afiliado y las claves de las demás relaciones sean correctas, se verifican restricciones de chequeo en los campos de fecha y se confirman las pruebas de unicidad en las claves primarias de cada tabla.