

Tecnologías de Gestión de Datos

Diseño de una aplicación para un Hospital

Víctor Iranzo Jiménez

Adriano Vega Llobell

Índice

Índice	1
1. Presentación del caso.....	2
Introducción.....	2
Funciones del sistema.....	2
Gestión de personal.....	3
Gestión de instalaciones	5
Gestión de hospitalizaciones.....	7
Gestión de patologías y medicamentos	11
2. Desarrollo del trabajo	13
Cronograma	14
3. Diseño conceptual del sistema.....	15
4. Plan de pruebas.....	16
5. Implementaciones del sistema	17
Implementación relacional con Oracle.....	17
Diseño lógico	17
Diseño físico	17
Implementación.....	19
Pruebas	19
Implementación NoSQL con Apache Cassandra.....	21
Diseño lógico	21
Diseño físico	23
Implementación.....	23
Pruebas	24
Implementación NoSQL con Neo4J.....	26
Diseño lógico	26
Diseño físico	26
Implementación.....	28
Pruebas	28
Anexo 1: datos para pruebas	30

1. Presentación del caso

Introducción

El proyecto a desarrollar es un programa de gestión para el área socio-sanitaria de un hospital. Es decir, para el área relacionada con el tratamiento de los pacientes. Se ha enfocado el proyecto para los pacientes que tienen estancia dentro del hospital.

Funciones del sistema

Una aplicación real para la gestión de un hospital comprende un gran número de áreas y ámbitos: gestión de personal, gestión económica de recursos materiales y personales, mantenimiento de instalaciones, gestión de citas para el paciente, control de acceso al histórico de un paciente... Debido a las dimensiones de este proyecto, no se van a abordar todas ellas y el esfuerzo se va a centrar en modelar parte del área socio-sanitaria.

Vamos a presentar las funciones del sistema en 4 grandes bloques:

- Gestión de personal
- Gestión de instalaciones
- Gestión de hospitalizaciones
- Gestión de patologías y medicamentos

Gestión de personal

La gestión de personal se centra en la creación y edición de empleados. Vamos a considerar únicamente dos tipos de empleados: médicos y enfermeras. En general, se almacenarán datos básicos para el contacto con el empleado (Nombre, dirección, teléfono) y la planta en la que llevan a cabo su labor. De cada uno de ellos nos interesa almacenar atributos diferentes: el número de colegiado del primero y un *flag* para la Enfermera que indica si es Supervisora.

CU001	Operaciones CRUD sobre médico
El usuario puede registrar nuevos médicos, listar los médicos que se encuentran en el hospital, filtrando por aquellos que trabajan en una determinada planta, actualizar cualquiera de sus propiedades y eliminar alguno de ellos.	
Campos	<ul style="list-style-type: none">• Identificador: (entero) autogenerado y requerido.• DNI: (string) requerido.• Nombre: (string) requerido.• Primer apellido: (string) requerido.• Segundo apellido: (string).• Dirección: (string) requerido.• Teléfono: (string) requerido.• Activo: (bool) requerido.• Trabaja en el turno de noche: (bool) requerido.• Número de colegiado: (string) requerido.
Claves foráneas	<ul style="list-style-type: none">• Id Planta: (entero) representa la planta en la que trabaja.
Consideraciones	<ul style="list-style-type: none">• La eliminación en cascada está prohibida: no se pueden eliminar médicos que hayan firmado algún tratamiento. En caso de que se desee ocultar un médico del sistema, se usará el booleano Activo para que no se puedan vincular nuevos tratamientos al médico. Los tratamientos existentes firmados por ese médico seguirán vinculados a él.

CU002	Operaciones CRUD sobre enfermera
El usuario puede registrar nuevas enfermeras, listar las enfermeras que se encuentran en el hospital, filtrando por aquellas que trabajan en una determinada planta o tienen el rol de supervisora, actualizar cualquiera de sus propiedades y eliminar alguna de ellas.	
Campos	<ul style="list-style-type: none"> • Identificador: (entero) autogenerado y requerido. • DNI: (string) requerido. • Nombre: (string) requerido. • Primer apellido: (string) requerido. • Segundo apellido: (string). • Dirección: (string) requerido. • Teléfono: (string) requerido. • Activo: (bool) requerido. • Trabaja en el turno de noche: (bool) requerido. • Es supervisora: (bool) requerido.
Claves foráneas	<ul style="list-style-type: none"> • Id Planta: (entero) representa la planta en la que trabaja.
Consideraciones	<ul style="list-style-type: none"> • La eliminación en cascada está prohibida: no se pueden eliminar enfermeras que hayan escrito algún seguimiento. En caso de que se desee ocultar una enfermera del sistema, se usará el booleano Activo para que no se puedan vincular nuevos seguimientos a la enfermera. Los seguimientos existentes escritos por esa enfermera seguirán vinculados a ella.

Gestión de instalaciones

De nuevo, las instalaciones y los recursos materiales de un hospital son muchos: se pueden gestionar tanto a una granularidad de habitaciones o llegar incluso a la granularidad del mobiliario que en ellas se encuentran (camas, sofás, mesas, etc). Otros recursos que se pueden querer gestionar son: pasillos, equipos de ventilación, ordenadores, etc. Por simplicidad, vamos a representar únicamente dos entidades relacionadas con las instalaciones del hospital: las habitaciones y las plantas.

En cuanto a las habitaciones, muchos hospitales emplean una misma habitación para que pernocte más de un paciente, por lo que nos interesa almacenar la capacidad de cada habitación.

CU003	Operaciones CRUD sobre Planta
El usuario puede crear nuevas plantas, listar las plantas que existen en el hospital, actualizar su nombre y eliminar alguna de ellas. Las plantas llevan asociadas un nombre, que indica el área sanitaria en la que se centran: planta de Dermatología, planta de Maternidad, planta de Pediatría...	
Campos	<ul style="list-style-type: none">• Identificador: (entero) autogenerado y requerido.• Nombre: (string) requerido.
Claves foráneas	Ninguna.
Consideraciones	<ul style="list-style-type: none">• La eliminación en cascada está prohibida: no se pueden eliminar plantas en las que trabaja algún empleado o que tengan relacionada una habitación.

CU004	Operaciones CRUD sobre Habitación
<p>El usuario puede crear nuevas habitaciones, listar las habitaciones que existen en el hospital, filtrando por aquellas que están en una determinada planta, actualizar su capacidad y eliminar alguna de ellas. La capacidad representa el número de pacientes que pueden pernoctar a la vez en una habitación. Por último, una habitación siempre se encuentra vinculada a una planta, que representa el lugar donde se encuentra.</p>	
Campos	<ul style="list-style-type: none"> • Identificador: (entero) autogenerado y requerido. • Capacidad: (entero) requerido.
Claves foráneas	<ul style="list-style-type: none"> • Id Planta: (entero) representa la planta en la que se encuentra.
Consideraciones	<ul style="list-style-type: none"> • La eliminación en cascada está prohibida: no se pueden eliminar habitaciones en las que un paciente haya pernoctado.

Gestión de hospitalizaciones

La gestión de hospitalizaciones no incluye únicamente el ingreso de un paciente. En este apartado incluimos también actividades como la asignación de una habitación y la introducción de cuidados al paciente, que clasificamos en la realización de seguimientos y la pauta de tratamientos.

CU005	Operaciones CRUD sobre Hospitalización
<p>Una hospitalización es el ingreso de un paciente por su sintomatología en una determinada planta. La hospitalización se produce en un rango de fechas y puede provenir o no de Urgencias, condición que interesa almacenar para priorizar en acciones como la asignación de habitaciones. En el momento que se crea una Hospitalización se debe asignar una habitación al paciente.</p> <p>El usuario puede registrar la hospitalización de un paciente, listar las hospitalizaciones que se han dado en el hospital, filtrando por aquellas que están activas (no se ha producido el fin de la hospitalización) o las que se dan en una determinada planta. Además, se pueden crear y vincular seguimientos y tratamientos con una hospitalización. Por último, se pueden eliminar hospitalizaciones atendiendo a las restricciones.</p>	
Campos	<ul style="list-style-type: none">• Identificador: (entero) autogenerado y requerido.• Fecha alta: (fecha y hora) requerido.• Fecha baja: (fecha y hora).• Procede de Urgencias: (bool) requerido.
Claves foráneas	<ul style="list-style-type: none">• Id Paciente: (entero) representa el paciente que ha sido hospitalizado.• Id Habitación: (entero) representa la habitación en la que pernocta el paciente.
Consideraciones	<ul style="list-style-type: none">• La eliminación en cascada está prohibida: no se pueden eliminar hospitalizaciones que tengan asignado un Tratamiento o un Seguimiento.

CU006	Operaciones CRUD sobre Tratamiento
<p>Un tratamiento es la pauta que un médico realiza a un paciente hospitalizado para tratar su sintomatología. El tratamiento puede incluir el suministro de un medicamento o puede consistir en una serie de recomendaciones que el médico da y se introducen como texto libre. Un tratamiento se pauta en una fecha determinada, pero su fecha de inicio puede ser distinta. La fecha en la que finaliza el tratamiento puede especificarse al crear el tratamiento o más adelante.</p> <p>Además, durante una hospitalización se puede pautar más de un tratamiento: un médico especialista en un área puede dar un tratamiento y otro médico especialista en otra área puede pautar otro diferente simultáneamente si el paciente sufre de diferentes trastornos. Por último, hay que mencionar que en un escenario real otros profesionales distintos al médico pueden pautar tratamientos, pero por simplicidad vamos a contemplar este escenario únicamente.</p> <p>Por tanto, el usuario puede registrar nuevos tratamientos, listar los tratamientos que se han dado a un paciente durante una hospitalización y se pueden eliminar tratamientos atendiendo a las restricciones.</p>	
Campos	<ul style="list-style-type: none"> • Identificador: (entero) autogenerado y requerido. • Fecha de creación:(fecha y hora) requerido. • Descripción: (string) requerido. • Fecha inicio del tratamiento: (fecha y hora) requerido. • Fecha fin del tratamiento: (fecha y hora).
Claves foráneas	<ul style="list-style-type: none"> • Id Hospitalización: (entero) representa la hospitalización de un paciente durante la cual se ha pautado el tratamiento. • Id Médico: (entero) representa el médico que ha pautado el tratamiento. • Id Medicamento: (entero) no requerida, representa el medicamento que se suministrará con el tratamiento asociado. La pauta del medicamento (frecuencia entre dosis y cantidad a ingerir) se especificará como texto libre en el campo Descripción.
Consideraciones	<ul style="list-style-type: none"> • Ninguna.

CU007	Operaciones CRUD sobre Seguimiento
<p>Un seguimiento es una entrada en el sistema en la que se detalla un cuidado o reconocimiento realizado durante su hospitalización por un profesional. Se han identificado principalmente 3 tipos seguimientos que se pueden realizar: las escalas, que consisten en formularios de respuesta mayoritariamente cerrada para evaluar un aspecto del paciente (su nutrición, movilidad, nivel de ansiedad...), los controles, relacionados con la toma de constantes al paciente (tensión, niveles de azúcar, peso y altura...), y las anotaciones, que son entradas de texto libre para detallar la evolución del paciente. En el proyecto solo vamos a considerar los seguimientos del tercer tipo.</p> <p>Al igual que con los tratamientos, en un entorno real estos pueden ser redactados o introducidos por cualquier profesional, pero para el proyecto vamos a considerar que solo son las enfermeras las que pueden redactarlos, ya que son las más estrechamente vinculadas con los cuidados del paciente.</p> <p>Un usuario puede registrar nuevos seguimientos, listar los seguimientos que se han dado a un paciente durante una hospitalización y se pueden eliminar los seguimientos atendiendo a las restricciones.</p>	
Campos	<ul style="list-style-type: none"> • Identificador: (entero) autogenerado y requerido. • Descripción: (string) requerido. • Fecha de creación:(fecha y hora) requerido.
Claves foráneas	<ul style="list-style-type: none"> • Id Hospitalización: (entero) representa la hospitalización de un paciente durante la cual se ha introducido el seguimiento. • Id Enfermera: (entero) representa al profesional que ha introducido el seguimiento.
Consideraciones	<ul style="list-style-type: none"> • Ninguna.

CU008	Operaciones CRUD sobre Paciente
<p>El paciente es la entidad central en un sistema de gestión hospitalario. En la mayoría de sistemas, el paciente presenta una ficha o histórico en el que se pueden consultar tanto sus hospitalizaciones como la asistencia a consultas de especialistas que haya realizado. Otros datos que se pueden consultar a través de la ficha de un paciente son sus vacunas, patologías o enfermedades, antecedentes personales y familiares... En nuestro proyecto, solo vamos a considerar vinculado al paciente dos entidades: las hospitalizaciones y las patologías (que más adelante detallamos).</p> <p>Las acciones permitidas sobre el paciente son: creación de nuevos pacientes, edición de los datos de este y eliminación de acuerdo con sus restricciones.</p>	
Campos	<ul style="list-style-type: none"> • Identificador: (entero) autogenerado y requerido. • DNI: (string) requerido. • Nombre: (string) requerido. • Primer apellido: (string) requerido. • Segundo apellido: (string). • Dirección: (string) requerido. • Teléfono: (string) requerido. • Número SIP: (string) requerido.
Claves foráneas	<ul style="list-style-type: none"> • Relación N to N con Patologías: representa la serie de patologías que el paciente ha sufrido.
Consideraciones	<ul style="list-style-type: none"> • La eliminación en cascada está prohibida: no se pueden eliminar pacientes que tengan asignada una Hospitalización.

Gestión de patologías y medicamentos

Este apartado se centra en las funciones de registrar nuevos medicamentos disponibles en el hospital, su uso para tratar patologías y aplicación durante algunos tratamientos a pacientes.

CU009	Operaciones CRUD sobre Patología
<p>Una patología es una enfermedad física o mental que sufren uno o más pacientes. Las patologías se clasifican por la Organización Mundial de la Salud (OMS) en el CIE-11 (Clasificación Internacional de Enfermedades), que contiene alrededor de 55.000 códigos únicos para traumatismos, enfermedades y causas de muerte para proporcionar un lenguaje común entre profesionales de todo el mundo.</p> <p>Aparte de almacenar el código asociado a la patología, nos interesa almacenar si la enfermedad es infecciosa, dando igual el agente (virus, bacteria, etc.) a través del que se contagia, y si es hereditaria.</p> <p>Por un lado, se ha añadido una asociación con Paciente para materializar las enfermedades que padece un Paciente. Por otro lado, se ha incluido una relación con Medicamento para mostrar la relación entre una enfermedad y el medicamento que la trata, siendo esta de cardinalidad 0..1 por simplicidad.</p> <p>Las acciones que se pueden realizar sobre una patología son: creación de nuevas enfermedades, listar las enfermedades existentes, eliminar una patología de acuerdo a sus restricciones y vincular una patología con un paciente o un medicamento.</p>	
Campos	<ul style="list-style-type: none">• Identificador: (entero) autogenerado y requerido.• Nombre: (string) requerido.• Es infecciosa: (bool) requerido.• Es hereditaria: (bool) requerido.• Código CIE: (string).
Claves foráneas	<ul style="list-style-type: none">• Id Medicamento: (entero) representa el medicamento que se emplea para tratar la patología.• Relación N to N con Paciente: representa la serie de pacientes que sufren una patología.
Consideraciones	<ul style="list-style-type: none">• Ninguna.

CU010	Operaciones CRUD sobre Medicamento
<p>Un medicamento es una sustancia, fármaco o principio activo que sirve para curar o reducir los efectos de una enfermedad.</p> <p>Las pautas (dosis y tiempo entre tomas) no se especifican en la propia entidad del medicamento, ya que pueden ser distintas de acuerdo al paciente para tratar una misma sintomatología. Además, un medicamento se puede emplear para tratar diferentes patologías.</p> <p>Las acciones permitidas sobre la entidad Medicamento son: registrar nuevos medicamentos, editar medicamentos existentes y eliminarlos de acuerdo a sus restricciones.</p>	
Campos	<ul style="list-style-type: none"> • Identificador: (entero) autogenerado y requerido. • Nombre: (string) requerido. • Proveedor: (string) requerido. • Vía de administración: (string requerido).
Claves foráneas	<ul style="list-style-type: none"> • Ninguna.
Consideraciones	<ul style="list-style-type: none"> • La eliminación en cascada está prohibida: no se pueden eliminar medicamentos que estén vinculados a una Patología o Tratamiento.

2.Desarrollo del trabajo

El desarrollo del trabajo se ha hecho de la siguiente forma:

- Captación de requisitos.
- Implementación:
 - Uno de los miembros del equipo desarrollaba el esquema de base de datos para una tecnología concreta.
 - El otro miembro revisaba la implementación del esquema y poblaba las bases de datos, para detectar los posibles problemas que pudiera haber.
 - De esta forma, ambos logramos entender cómo funcionaba cada uno de los esquemas.
- Redacción de la memoria:
 - Muchos de los artefactos (en especial, diagramas) han sido desarrollados durante las fases de captación de requisitos (modelo conceptual) e implementación (al definir el esquema).

Modelo de BD	Esquema	Revisión	Data seeding
Oracle SQL	Adriano	Víctor	Víctor
Cassandra	Adriano	Víctor	Víctor
Neo4J	Víctor	Adriano	Víctor y Adriano

Cronograma

El desarrollo del trabajo se puede resumir en el siguiente cronograma:



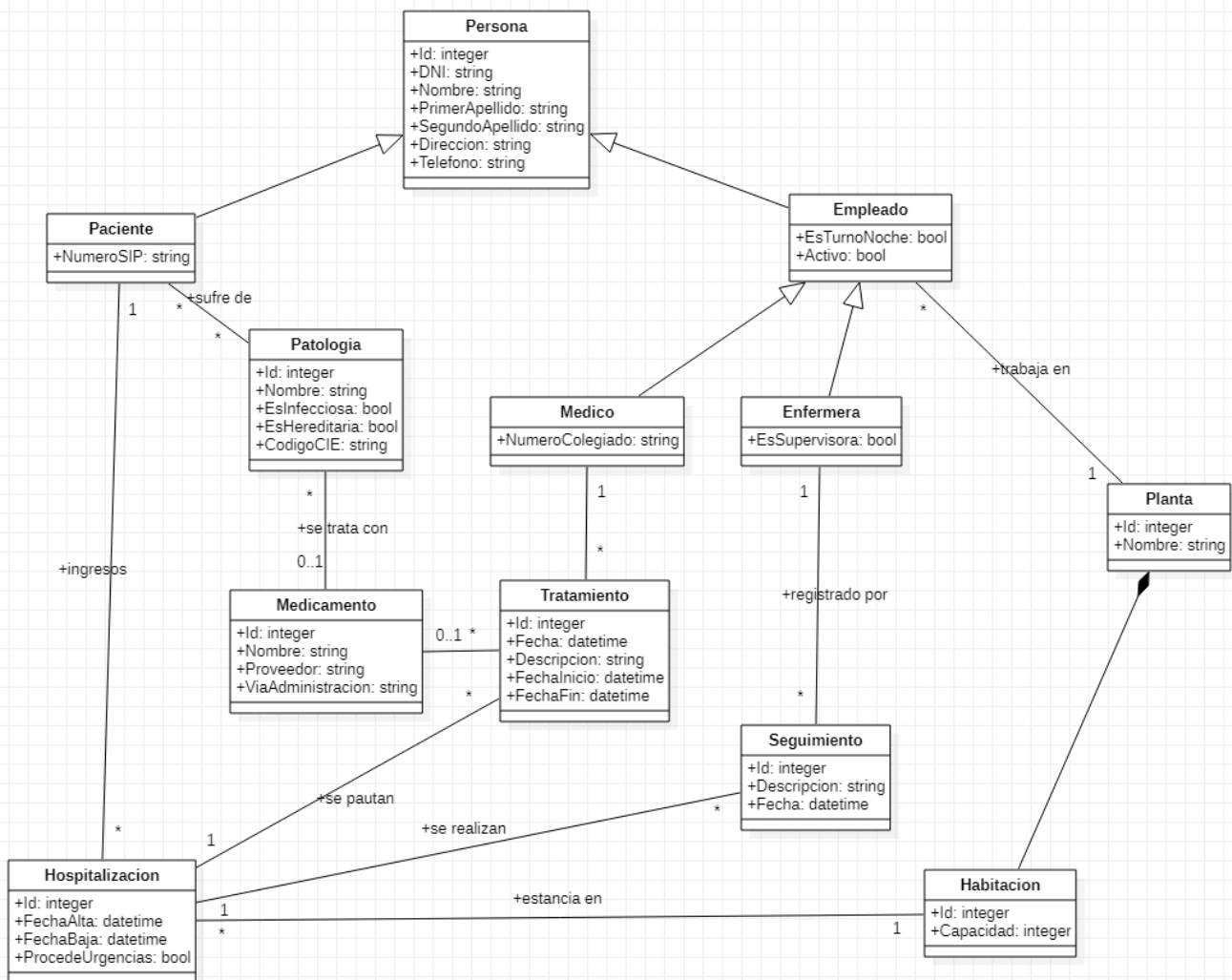
Las fechas de cada tarea están en la siguiente tabla:

Actividad	Fecha Inicio	Duración (días)	Fecha Fin
Diseño conceptual del sistema	20/12/2019	1	21/12/2019
Especificación de requisitos	22/12/2019	5	27/12/2019
Reunión con el profesor	09/01/2020	1	10/01/2020
Implementación en SQL	11/01/2020	2	13/01/2020
Implementación en Neo4J	12/01/2020	3	15/01/2020
Implementación en Cassandra	13/01/2020	3	16/01/2020
Revisión de impl. SQL	17/01/2020	1	18/01/2020
Revisión de impl. Cassandra	17/01/2020	1	18/01/2020
Revisión de impl. Neo4J	18/01/2020	1	19/01/2020
Pruebas en impl. SQL	25/01/2020	2	27/01/2020
Pruebas en impl. Cassandra	31/01/2020	3	03/02/2020
Pruebas en impl. Neo4J	31/01/2020	2	02/02/2020
Completar redacción de la memoria	03/02/2020	5	08/02/2020

Durante todo el proceso se han intercambiado correos con el profesor siempre que ha sido necesario. Fruto de eso, ocurrió un parón entre la especificación de requisitos y el comienzo de las implementaciones de las bases de datos, ya que el equipo quería esperar a resolver ciertas dudas y quisimos esperarnos a reunirnos en persona.

También ha habido otros parones durante el desarrollo del proyecto debido a la carga de trabajo de otras asignaturas y la realización de exámenes.

3. Diseño conceptual del sistema



Antes de comenzar la definición de los requisitos del sistema, se realizó el diseño conceptual del sistema: un diagrama de clases UML. Esta versión inicial fue mejorada iterativamente según se especificaban los requisitos del sistema.

Este diseño conceptual sirvió de base para la implementación en los distintos modelos de base de datos. Como veremos a continuación, en la mayoría de los casos hubo de ser adaptado a las peculiaridades de cada sistema. Por ejemplo, se puede apreciar que contiene relaciones de herencia (*Empleado* hereda de *Persona*), relaciones de muchos a muchos (*Paciente* con *Patología*), etc. Estas relaciones no se representan igual en los distintos modelos de datos.

4. Plan de pruebas

Algunas de las pruebas para comprobar el correcto comportamiento del sistema se realizaron mediante la inserción de datos. La definición de los datos a insertar hizo partiendo del modelo conceptual. Los mismos datos se usaron en las 3 implementaciones para que el resultado obtenido con las consultas sea similar.

La definición de los datos se realizó buscando variabilidad: Los datos son reales conforme a los que serían introducidos en el sistema real en ejecución. También se busca explotar las cardinalidades en las diferentes asociaciones. Estos son algunos de ejemplos a los que hacemos referencia:

- Los códigos CIE de las patologías introducidas se han obtenido consultando la página del Ministerio de Sanidad para este propósito.
eCIEMaps: https://eciemaps.mscbs.gob.es/ecieMaps/browser/index_10_mc.html
- La descripción de los seguimientos y tratamientos de una hospitalización son acordes a la planta donde el paciente ingresa y el médico y enfermera que los realiza. También lo son los medicamentos que se pautan. Se busca reflejar así una historia de un paciente real. Por ejemplo, el paciente 'Miguel' ingresa en Traumatología.

Se le realiza un seguimiento y dos tratamientos hablando sobre el traumatismo que ha sufrido por parte de un enfermero y un médico en dicha planta. Se le pauta un medicamento para aliviar el dolor y un tratamiento de una semana de duración que consiste en una venda compresora.

- Existe una patología que sufren más de un paciente (Fibrosis quística) y pacientes que sufren más de una patología (Javier). Esto se hace así para probar la relación de muchos a muchos.
- Existen hospitalizaciones con más de un tratamiento, pacientes con más de una hospitalización, tratamientos sin medicamentos pautados, patologías sin medicamentos para tratarlos... Esto se hace así para probar que las asociaciones tiene una cardinalidad N (es decir, en la relación se acepta una colección de tratamientos y no una instancia únicamente) y para probar que algunos campos no tienen la restricción de requerido (por ejemplo, no es obligatorio pautar siempre un medicamento en un tratamiento).

Los datos que se introducen se pueden ver en el Anexo 1. También ofrecemos una vista por paciente, ya que la consulta que se hará para probar las diferentes implementaciones consiste en obtener todas las hospitalizaciones de un paciente, incluyendo sus tratamientos y seguimientos.

5. Implementaciones del sistema

El sistema debía implementarse en tres modelos de bases de datos distintos, con el objetivo de entender las diferencias de implementación entre ellos. Los modelos fueron el modelo relacional, modelo basado en grafos y el modelo basado en clave valor.

Los sistemas usados para implementar cada uno de ellos fueron, respectivamente, Oracle Database, Neo4J y Apache Cassandra. A continuación, se describe la implementación de cada uno de los sistemas, resaltando las particularidades en cada uno.

Implementación relacional con Oracle

Para la implementación siguiendo el modelo relacional, se usó el sistema de gestión de base de datos Oracle Database.



Diseño lógico

Para implementar el sistema en el modelo relacional, se aplicaron técnicas de **normalización** al esquema de la base de datos. Así, por un lado, se logra la correcta representación de las relaciones entre las entidades en el esquema; y por otro se logra evitar la redundancia de datos.

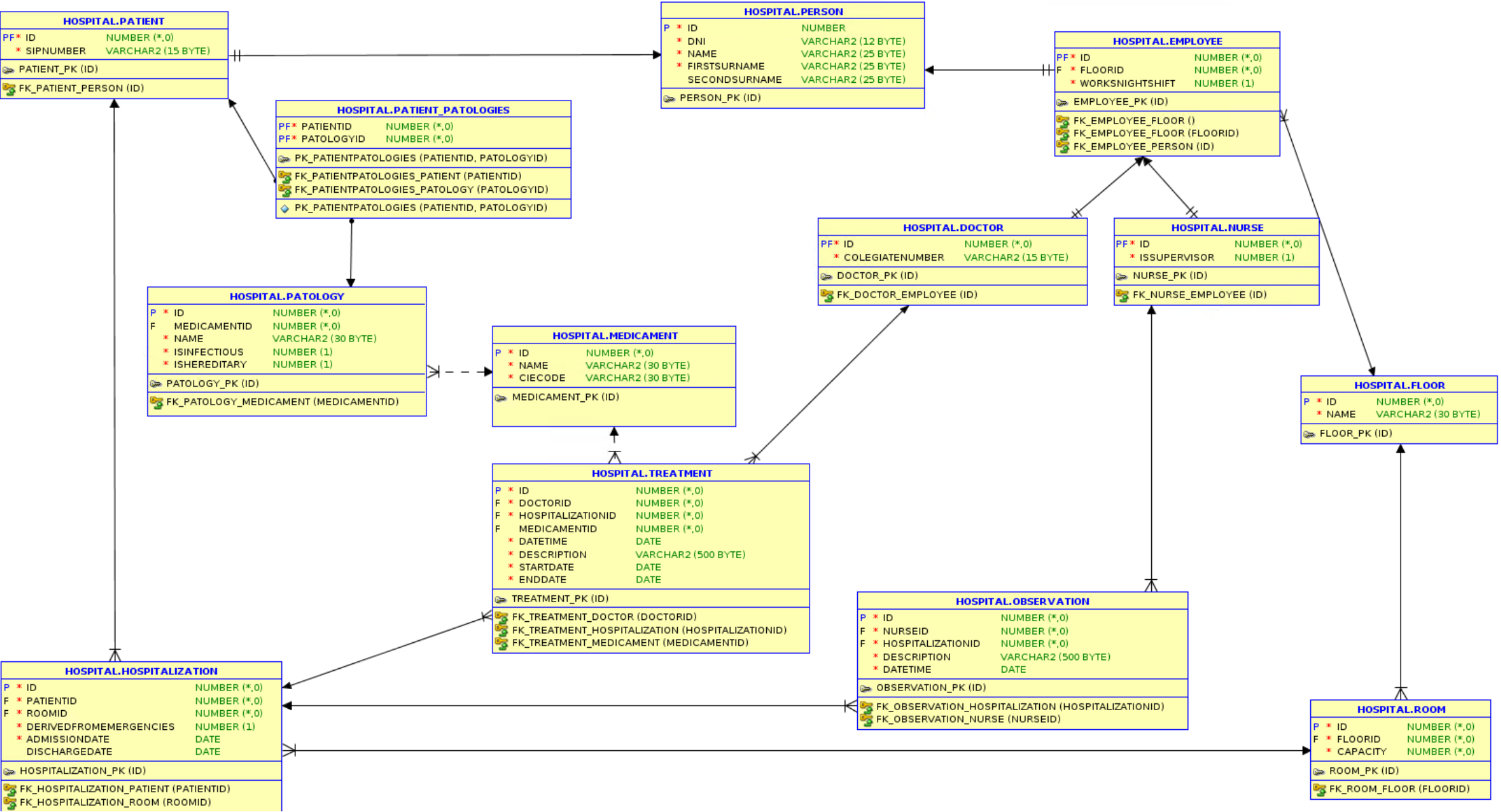
Como se aprecia en la imagen anterior, el esquema pudo traducirse al esquema relacional sin muchos cambios. Principalmente destacan dos cambios:

- La relación muchos a muchos de *Patient* y *Patology*, que tuvo que ser implementada mediante una tabla intermedia.
- La herencia en *Person* fue implementada siguiendo el patrón *Table per Type* (tabla por tipo), para evitar la redundancia de información en los esquemas. Cada heredero referencia a un registro en la tabla *Person*, que contiene sus datos.

Diseño físico

Se ha implementado la base de datos con un solo nodo, en lugar de convertirla en una base de datos distribuida. La base de datos está implementada sobre un fichero físico llamado 'Hospital_TGD'. Este fichero se ha mantenido con la configuración por defecto, por lo que se trata de un fichero desordenado.

Dicho fichero contiene el tablespace 'Hospital_TGD' al completo, es decir, tiene todas las tablas de la base de datos.



Implementación

La base de datos se ha implementado en la máquina virtual TGD-028-01, que tiene las credenciales por defecto (oracle, TGD.2018).

Para la creación de las tablas y los ficheros de almacenamiento, se desarrollaron una serie de *scripts* SQL. Utilizando esta aproximación el despliegue de los cambios en la base de datos se vuelve repetible, y nos permite también reducir el tiempo para hacer pruebas sobre la misma. El script es 'hospital.sql', por si se desea consultar los detalles de la implementación.

Respecto a los mecanismos de lógica en la BD, no cuenta con *triggers* ni otros mecanismos de lógica, ya que se acordó con el profesor que esto escapaba el ámbito del proyecto.

Pruebas

La inserción de los datos definidos en el plan de pruebas se ha realizado utilizando el script seeding.sql que se proporciona junto con la memoria.

Con una consulta como la siguiente podemos obtener las hospitalizaciones, seguimientos y tratamientos de un paciente por su DNI:

```
SELECT PE.NAME, PE.DNI, PE.FIRSTSURNAME, PE.SECONDSURNAME, PE.TELEPHONE, PE.ADDRESS, PA.SIPNUMBER
FROM person PE
INNER JOIN patient PA ON PE.Id = PA.Id
WHERE PE.DNI = '22543281Z';

SELECT * FROM hospitalization H WHERE H.PATIENTID = (SELECT P.Id FROM person P WHERE P.DNI = '22543281Z' AND rownum = 1);

SELECT * FROM observation O WHERE O.HOSPITALIZATIONID IN (
    SELECT H.Id FROM hospitalization H WHERE H.PATIENTID = (SELECT P.Id FROM person P WHERE P.DNI = '22543281Z' AND rownum = 1));

SELECT * FROM treatment T WHERE T.HOSPITALIZATIONID IN (
    SELECT H.Id FROM hospitalization H WHERE H.PATIENTID = (SELECT P.Id FROM person P WHERE P.DNI = '22543281Z' AND rownum = 1));
```

El resultado de la consulta, con el DNI del paciente introducido es el siguiente, que se corresponde con el dispuesto en el Anexo 1:

NAME	DNI	FIRSTSURNAME	SECONDSURNAME	TELEPHONE	ADDRESS	SIPNUMBER
Paco	22543281Z	Pérez	Peláez	619145283	Av. Pérez Galdós	1234515673
ID	PATIENTID	ROOMID	DERIVEDFROMEMERGENCIES	ADMISSIONDATE	DISCHARGEDATE	
1	1	4	0	2019-08-01 18:40:30	2019-08-03 10:15:30	
ID	NURSEID	HOSPITALIZATIONID	DESCRIPTION			
2	9	1	Paciente tiene una gran pérdida de sangre. Se notifica al médico de guardia y se aconseja suministrarle un gotero y realizarle una transfusión.			
ID	DOCTORID	HOSPITALIZATIONID	MEDICAMENTID	DATETIME	DESCRIPTION	
1	6	1	4	2019-08-01 12:00:00	Paciente con síntomas de Neumonía. Se le recomienda tomar antibióticos durante una semana, haciendo 2 tomas diarias. Permanecerá hospitalizado al menos 2 d.	
2	6	1	5	2019-08-02 12:00:00	El paciente no ha evolucionado favorablemente. Se le añade también al tratamiento con el que contaba, administración de Paracetamol durante el día de hoy.	

Por último, vamos a probar la restricción de eliminación en cascada. Si se trata de eliminar un paciente, vemos que el sistema no nos lo permite porque tiene hospitalizaciones asignadas:

```
DELETE FROM patient WHERE id = 1;
```

Salida de Script x

Tarea terminada en 0,099 segundos

Error que empieza en la línea: 1 del comando :
DELETE FROM patient WHERE id = 1
Informe de error -
ORA-02292: restricción de integridad (HOSPITAL.FK_HOSPITALIZATION_PATIENT) violada - registro secundario encontrado

Lo mismo ocurre si intentamos eliminar la tupla en la tabla Person:

```
DELETE FROM person WHERE id = 1;
```

Salida de Script x

Tarea terminada en 0,05 segundos

Error que empieza en la línea: 1 del comando :
DELETE FROM person WHERE id = 1
Informe de error -
ORA-02292: restricción de integridad (HOSPITAL.FK_PATIENT_PERSON) violada - registro secundario encontrado

En cambio, sí se puede eliminar un medicamento que no está vinculado a ningún tratamiento ni a ninguna patología:

```
DELETE FROM medicament WHERE id = 2;
```

Salida de Script x

Tarea terminada en 0,079 segundos

1 fila eliminado

Implementación NoSQL con Apache Cassandra

El siguiente modelo en el que se hubo de implementar la aplicación fue en Apache Cassandra. Cassandra es un sistema de gestión de base de datos NoSQL distribuido y basado en el almacenamiento clave-valor orientado a la escalabilidad lineal y la alta disponibilidad.¹



Diseño lógico

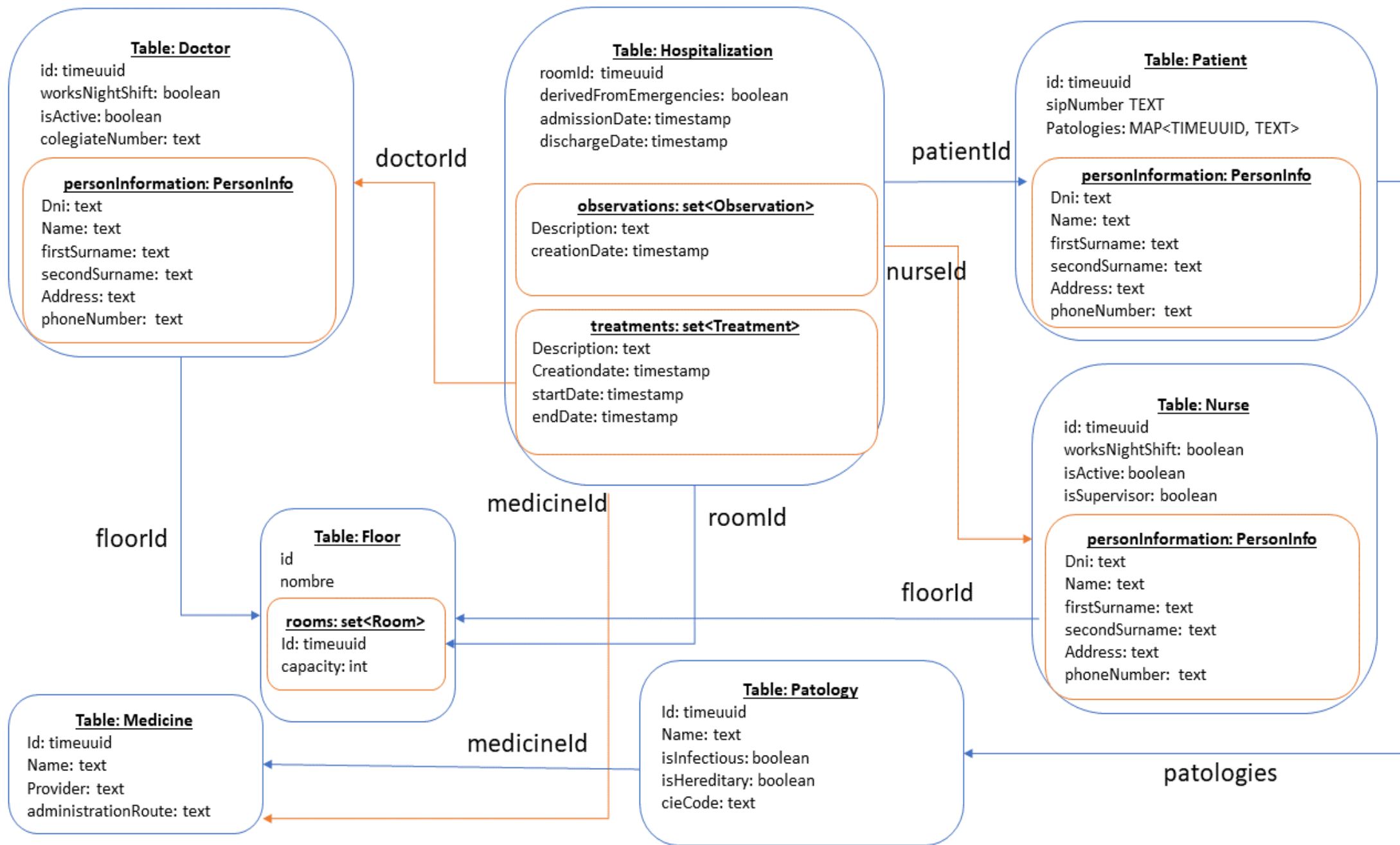
A diferencia de en el modelo relacional, en Cassandra se fomenta la desnormalización del esquema lógico. Dado que las consultas son muy costosas y no hay *joins*, se prefiere tener los datos en la misma tabla. Así, se acepta la duplicación de ciertos datos.

En el esquema del diseño lógico que se puede ver abajo, se distinguen estos casos. Por ejemplo, *PersonInformation* se ha definido como una entidad *type*, que se reutiliza en cada entidad del tipo *Person*. Por ejemplo, en *Doctor* o *Nurse*.

También se ha optado por realizar agregaciones de datos. Este es el caso de los tratamientos o los seguimientos, que se encuentran contenidos como una colección dentro de la hospitalización.

Cabe destacar que en Cassandra no se ha implementado el borrado en cascada, ya que esa lógica se deja a nivel de aplicación. Las referencias externas realmente no existen aquí.

¹ https://es.wikipedia.org/wiki/Apache_Cassandra



Diseño físico

La base de datos se ha implementado en un *cluster* compuestos por 3 nodos. Estos nodos estaban instalados en máquinas virtuales distintas. Se usó la estrategia de replicación simple, con un factor de 2 nodos. Así, cada valor dato introducido en la base de datos, estará replicada en dos nodos, mejorando la disponibilidad de la base de datos.

Implementación

Como se explicó en el apartado anterior, la base de datos se implementó como un cluster de 3 nodos. Cada uno de los nodos está ubicado en una máquina virtual distinta: TGD-028-c1, TGD-028-c2 y TGD-028-c3, respectivamente.

Dado que se ha configurado el factor de autorreplicación a 2, todas las escrituras están consensuadas por dos de los tres nodos. No se optó por 3 nodos para reducir los tiempos de escritura; además de que se economiza espacio de almacenamiento al no estar totalmente replicados. Los nodos por tanto pueden tener información distinta entre ellos.

Para la creación de las tablas, se optó una vez más por utilizar scripts. En este caso, se ha creado un script CQL, 'hospital.cql', que viene adjunto con el documento. Estos scripts crean tanto el keyspace como las tablas.

Pruebas

La inserción de los datos de prueba se puede encontrar en el fichero seeding.cql. Debido a que se decidió usar para los identificadores de las entidades el tipo `TIMEUUID`, se crearon identificadores empleando una API (<https://www.uuidgenerator.net/version1>) y se utilizaron para establecer las relaciones entre las diferentes instancias.

Durante este proceso de pruebas se detectaron errores que se habían cometido debido principalmente a atributos que estaban en el modelo conceptual y no en el modelo lógico.

Una vez insertados los datos, se hizo una consulta para obtener el historial de un paciente (recordar que el historial de un paciente lo definimos como la suma de sus patologías y sus hospitalizaciones, mostrando sus seguimientos y tratamientos).

Como la información está agregada, la consulta es más simple, pero se debe realizar en dos pasos: primero, se obtiene el identificador del paciente con el DNI provisto y una vez obtenido esto, se pueden obtener las hospitalizaciones del paciente. Si el DNI del paciente se hubiera replicado en la hospitalización y se hubiera creado un índice para el DNI, la consulta se hubiera podido realizar en un único paso, pero no lo hemos considerado así oportuno. En la imagen inferior, se están obteniendo las hospitalizaciones del paciente 'Paco':

```
cqlsh> SELECT id, personinformation.dni, personinformation.name FROM hospitaldb.Patient;
```

id	personinformation.dni	personinformation.name
49b8e666-4aca-11ea-b77f-2e728ce88125	12533487A	María
a20d55d8-4acd-11ea-b77f-2e728ce88125	26755234P	Héctor
0d95346a-4ace-11ea-b77f-2e728ce88125	21735438Q	Javier
6439de24-4ace-11ea-b77f-2e728ce88125	22543281Z	Paco

```
(4 rows)
cqlsh> SELECT * FROM hospitaldb.Hospitalization WHERE patientid=6439de24-4ace-11ea-b77f-2e728ce88125;
```

patientid	roomid	admissiondate	derivedfromemergencies	dischargedate	observations	treatments
6439de24-4ace-11ea-b77f-2e728ce88125	e69373a1-4a9f-11ea-8d59-7bd78f6f4dca	2019-08-01 16:40:30+0000	False	2019-08-03 08:15:30+0000	{nurseid: 37770d40-4ac8-11ea-b77f-2e728ce88125, description: 'Paciente tiene una gran pérdida de sangre. Se notifica al médico de guardia y se aconseja suministrarle un gotero y realizarle una transfusión.', creationdate: '2019-08-02 07:10:00+0000'}} {{doctorid: ebda2abc-4ac6-11ea-b77f-2e728ce88125, medicamentid: bc434c70-4acb-11ea-b77f-2e728ce88125, description: 'Paciente con síntomas de Neumonía. Se le recomienda tomar antibióticos durante una semana, haciendo 2 tomas diarias. Permanecerá hospitalizado al menos 2 días. Presenta tos seca y febrícula.', creationdate: '2019-08-01 10:00:00+0000', startdate: '2019-08-01 10:00:00+0000', enddate: '2019-08-08 10:00:00+0000'}, {doctorid: ebda2abc-4ac6-11ea-b77f-2e728ce88125, medicamentid: eb6bf9e8-4acb-11ea-b77f-2e728ce88125, description: 'El paciente no ha evolucionado favorablemente. Se le añade también al tratamiento con el que contaba, administración de Paracetamol durante el día de hoy.', creationdate: '2019-08-02 10:00:00+0000', startdate: '2019-08-02 12:00:00+0000', enddate: '2019-08-02 20:00:00+0000'}}	

```
(1 rows)
cqlsh>
```

Obtener las patologías también es sencillo ya que estas se encuentran en la tabla Patient:

```
cqlsh> SELECT * FROM hospitaldb.Patient WHERE id=0d95346a-4ace-11ea-b77f-2e728ce88125;
```

id	patologies	personinformation	sipnumber
0d95346a-4ace-11ea-b77f-2e728ce88125	{39f16d72-4acd-11ea-b77f-2e728ce88125: 'Cáncer de pulmón', 7b6a9144-4acc-11ea-b77f-2e728ce88125: 'Fibrosis quística'} {dni: '21735438Q', name: 'Javier', firstsurname: 'Iranzo', secondsurname: 'Sevilla', address: 'C. San Antonio', phonenumber: '611728393'} 1923146513		

```
(1 rows)
cqlsh>
```

Por último, se muestra una imagen de cómo está configurado el clúster:

```
[root@TGD-028-c1 ~]# nodetool status hospitaldb
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address            Load          Tokens         Owns (effective)  Host ID
UN  192.168.233.142    3,65 MB      256            65.8%             217604a4-9afb-44d
c-a8f9-100b7da81020 rack1
UN  192.168.233.143    3,66 MB      256            67.2%             2108724d-77fd-403
f-ae0a-b6093c5d2d4e rack1
UN  192.168.233.144    3,56 MB      256            67.0%             417ab682-912a-483
d-bd06-c27be8429978 rack1
```

Implementación NoSQL con Neo4J

Neo4j es un software libre de Base de datos orientada a grafos. Hemos podido comprobar en el proyecto lo sencillo que es de emplear, tanto durante la inserción de datos como para la consulta de estos. A continuación, detallamos los pasos que hemos seguido en la implementación.



Diseño lógico

Durante la definición del diseño lógico es necesario establecer los diferentes tipos de nodos que existen y las relaciones entre estos. Los tipos de nodos que se han definido son muy similares a los definidos en el modelo conceptual, con la excepción de que no los atributos procedentes de entidades bases se han representado en las entidades derivadas. En las relaciones se ha incluido una dirección para que el tipo de la relación concuerde con el sentido en el que se recorre esta.

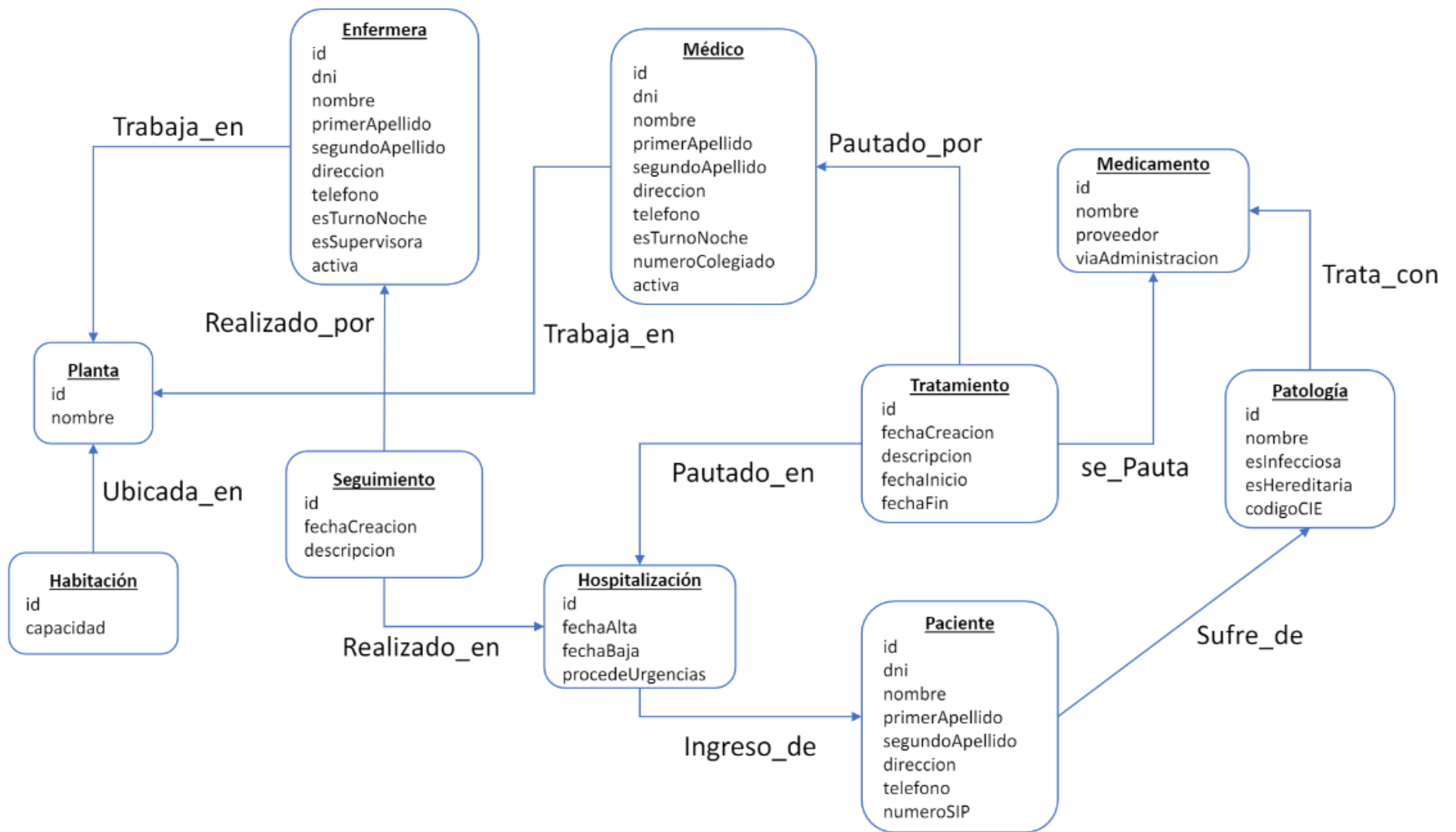
Por ejemplo, la asociación entre Seguimiento y Hospitalización se ha decidido representar como una relación 'Realizado_en' que va del primero al segundo porque semánticamente es como mejor encaja (ya que una Hospitalización tiene una colección de seguimientos, representar la relación en sentido contrario es más difícil de nombrar).

En el siguiente diagrama se muestran los tipos de nodos y relaciones que se han definido.

Diseño físico

Los requisitos de Neo4J a nivel de software que requiere tener instalado estaban cubiertos por la imagen de la máquina que se ha empleado. La versión que se ha empleado de Neo4J es la Community Edition.

Más allá de estos requisitos, no ha sido necesario especificar ningún parámetro adicional para configurar el nivel físico y emplear la base de datos.



La implementación del sistema se ha realizado en la máquina TGD-014-c1. Debido a la naturaleza de Neo4J, la inserción de datos de prueba es lo que se ha hecho durante la fase de implementación. El fichero para la inserción de datos se llama `seeding.cypher`, que contiene los datos que se llevan usando a lo largo de toda la memoria. Parte del resultado de aplicar este script lo forman los nodos de la siguiente imagen:



```

1 MATCH (n) -[:Realizado_en|Pautado_en]-> (h:Hospitalizacion) -[:Ingreso_de]-> (p:Paciente) -[:Sufre_de]-> (pa:Patologia)
2 WHERE p.dni = "26755234P"
3 RETURN p,pa,h,n

```

\$ MATCH (n) -[:Realizado_en|Pautado_en]-> (h:Hospitalizacion) -[:Ingreso_de]-> (p:Paciente) -[:Sufre_de]-> (pa:Patologia) WHERE p.dni = "26755234P"

Graph

*(6) Hospitalizacion(1) Paciente(1) Patologia(1) Seguimiento(1) Tratamiento(2)

*(5) Ingreso_de(1) Pautado_en(2) Realizado_en(1) Sufre_de(1)

Rows

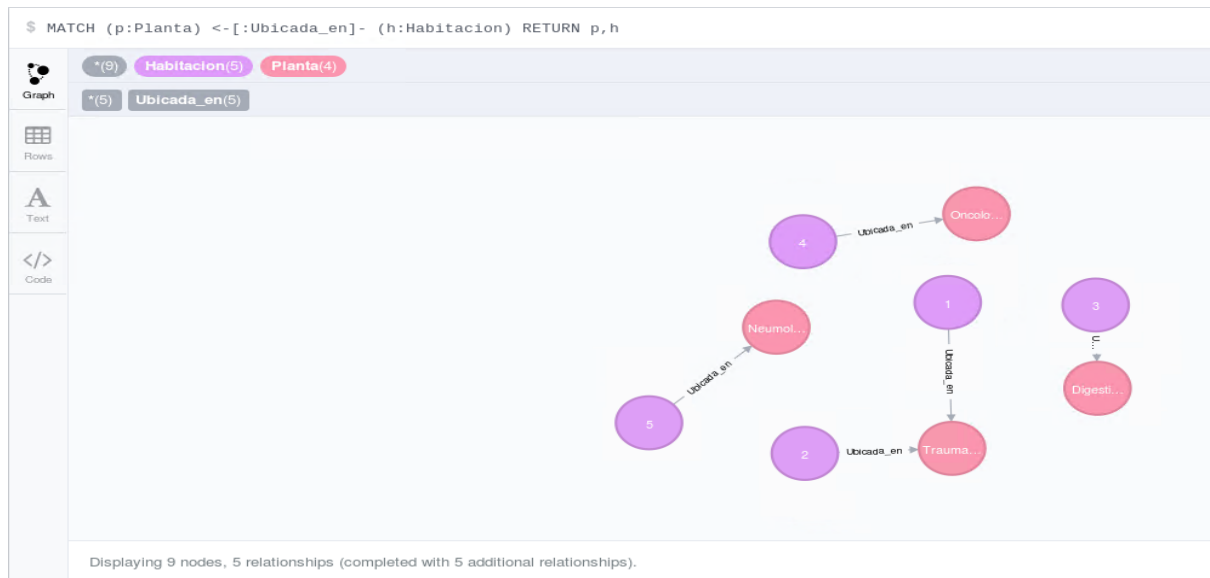
Text

Code

Displaying 6 nodes, 5 relationships (completed with 5 additional relationships).

Para el paciente 'Héctor Miguel' obtenemos la hospitalización que tiene asociada con 2 tratamientos y un seguimiento, así como la patología que sufre. La otra hospitalización que tiene no se muestra ya que no tiene ningún seguimiento o tratamiento, por lo que no hace match con el camino especificado.

Se han realizado muchas otras pruebas que han permitido hallar algunas inconsistencias (principalmente, datos faltantes). Entre ellas, podemos mencionar que tras la implementación inicial faltaron por introducir algunas plantas de las definidas en el Plan de pruebas. Esto puedo comprobarse gracias a una consulta como la que sigue:



Anexo 1: datos para pruebas

Pacientes							
Id	DNI	Nombre	Primer apellido	Segundo apellido	Dirección	Teléfono	Número SIP
1	22543281Z	Paco	Pérez	Peláez	Av. Pérez Galdós	619145283	1234515673
2	12533487A	María	Sebastián	Segarra	Av. Juan Llorens	621124353	4421115673
3	26755234P	Héctor	Miguel	Maldonado	C. Pedro Alonso	631527383	4923116712
4	21735438Q	Javier	Iranzo	Sevilla	C. San Antonio	611728393	1923146513

Hospitalizaciones					
Id	Fecha alta	Fecha baja	Procede de urgencias	Paciente	Habitación
1	2019-08-01T18:40:30	2019-08-03T10:15:30	FALSE	1 (Paco)	4
2	2019-11-10T09:10:00	2019-11-10T21:15:40	TRUE	2 (María)	4
3	2018-01-23T11:20:00	2018-01-24T16:45:00	FALSE	3 (Héctor)	1
4	2018-08-21T21:20:00	2018-08-22T09:45:40	FALSE	4 (Javier)	2
5	2016-05-27T19:10:00	2016-05-27T21:10:00	FALSE	3 (Héctor)	3

Tratamientos							
Id	Fecha creación	Descripción	Fecha inicio	Fecha fin	Médico	Hospitaliz.	Medicamento
1	2019-08-01T12:00:00	Paciente con síntomas de Neumonía. Se le recomienda tomar antibióticos durante una semana, haciendo 2 tomas diarias. Permanecerá hospitalizado al menos 2 días. Presenta tos seca y febrícula.	2019-08-01T12:00:00	2019-08-08T12:00:00	2 (Marta)	1	4 (Amitron)
2	2019-08-02T12:00:00	El paciente no ha evolucionado favorablemente. Se le añade también al tratamiento con el que contaba, administración de Paracetamol durante el día de hoy.	2019-08-02T14:00:00	2019-08-02T22:00:00	2 (Marta)	1	5 (Paracetamol)
3	2019-11-10T17:00:00	Paciente con resfriado común. Se le recomienda Frenadol. Permanecerá en Observación unas horas.	2019-11-10T17:00:00	2019-11-15T22:00:00	2 (Marta)	2	1 (Frenadol)
4	2018-01-23T21:00:00	Paciente con traumatismo en brazo derecho. Se ha realizado una intervención superficial para recolocar el radio. Se receta Paracetamol para el dolor.	2018-01-23T22:00:00	2018-01-24T09:00:00	1 (Eugenio)	3	5 (Paracetamol)
5	2018-08-21T22:20:00	Paciente con traumatismo en la muñeca. Se le recomienda llevar una venda compresora durante una semana.	2018-08-22T02:50:00	2018-08-29T12:30:00	1 (Eugenio)	3	

Seguimientos				
Id	Fecha creación	Descripción	Enfermera	Hospitalización
1	2018-08-21T21:20:00	Paciente llega con traumatismo en la muñeca. Se hacen curas sobre heridas a lo largo del antebrazo. Se le administra medicación para el dolor.	1 (Antonio)	3
2	2019-08-02T09:10:00	Paciente tiene una gran pérdida de sangre. Se notifica al médico de guardia y se aconseja suministrarle un gotero y realizarle una transfusión.	3 (Amalia)	1
3	2018-08-21T21:50:00	Síntomas de resfriado común. A la espera de las observaciones del médico.	4 (Pilar)	2

Patologías					
Id	Nombre	Es infecciosa	Es hereditaria	Código CIE	Medicamento para tratarlo
1	Neumonía	TRUE	FALSE	P23.0	3 (Bisolgrip)
2	Fibrosis quística	FALSE	TRUE	E84.9	
3	Nasofaringitis aguda [resfriado común]	TRUE	FALSE	J00	1 (Frenadol)
4	Traumatismo del hombro y brazo	FALSE	FALSE	S49.9	5 (Paracetamol)
5	Cáncer de pulmón	FALSE	FALSE	C33	

Pacientes y Patologías	
Paciente	Patología
3 (Héctor)	2 (Fibrosis quística)
4 (Javier)	2 (Fibrosis quística)
4 (Javier)	5 (Cáncer de pulmón)

Medicamentos			
Id	Nombre	Proveedor	Via administración
1	Frenadol	Johnson & Johnson	Pastillas
2	Pharmagrip	Cinfa	Pastillas
3	Bisolgrip	Boehringer Ingelheim	Granulado
4	Amitron	Johnson & Johnson	Granulado
5	Paracetamol	Tylenol	Pastillas

Plantas	
Id	Nombre
1	Traumatología
2	Digestivo
3	Oncología
4	Neumología

Habitaciones		
Id	Capacidad	Planta
1	2	1 (Traumatología)
2	2	1 (Traumatología)
3	1	2 (Digestivo)
4	1	3 (Oncología)
5	2	4 (Neumología)

Médicos										
Id	DNI	Nombre	Primer apellido	Segundo apellido	Dirección	Teléfono	Trabaja noche	Número de colegiado	Planta	Activo
1	11234481P	Eugenio	Pérez	Ramiro	Av. del Puerto	629355672	FALSE	22565678	1 (Traumatología)	TRUE
2	31531491R	Marta	Espinosa	Esteve	C. de Alfonso Santillana	625315262	FALSE	21525598	4 (Neumología)	TRUE

Enfermeras										
Id	DNI	Nombre	Primer apellido	Segundo apellido	Dirección	Teléfono	Trabaja noche	Es supervisora	Planta	Activa
1	66231181D	Antonio	Roca	Salazedo	C. de los Reyes Católicos	623325862	TRUE	FALSE	1 (Traumatología)	TRUE
2	23241597B	Laura	Sánchez	Martí	C. Pompeya	614385902	FALSE	TRUE	2 (Digestivo)	FALSE
3	27221891H	Amalia	Amor	Sevilla	C. de la Paz	613357002	FALSE	FALSE	4 (Neumología)	TRUE
4	26243871G	Pilar	Perpiñá	Alfreda	C. Catalunya	658957309	FALSE	FALSE	4 (Neumología)	FALSE

Paciente							
Id	DNI	Nombre	Primer apellido	Segundo apellido	Dirección	Teléfono	Número SIP
1	22543281Z	Paco	Pérez	Peláez	Av. Pérez Galdós	619145283	1234515673
Patologías							
Id	Nombre	Es infecciosa	Es hereditaria	Código CIE	Medicamento para tratarlo		
Hospitalizaciones							
Id	Fecha alta	Fecha baja	Procede de urgencias	Habitación			Planta
1	2019-08-01T18:40:30	2019-08-03T10:15:30	FALSE	4			3 (Oncología)
Seguimientos							
Id	Fecha creación	Descripción	Enfermera				
2	2019-08-02T09:10:00	Paciente tiene una gran pérdida de sangre. Se notifica al médico de guardia y se aconseja suministrarle un gotero y realizarle una transfusión.	3 (Amalia)				
Tratamientos							
Id	Fecha creación	Descripción	Fecha inicio	Fecha fin	Médico	Medicamento	
1	2019-08-01T12:00:00	Paciente con síntomas de Neumonía. Se le recomienda tomar antibióticos durante una semana, haciendo 2 tomas diarias. Permanecerá hospitalizado al menos 2 días. Presenta tos seca y febrícula.	2019-08-01T12:00:00	2019-08-08T12:00:00	2 (Marta)	4 (Amitron)	
2	2019-08-02T12:00:00	El paciente no ha evolucionado favorablemente. Se le añade también al tratamiento con el que contaba, administración de Paracetamol durante el día de hoy.	2019-08-02T14:00:00	2019-08-02T22:00:00	2 (Marta)	5 (Paracetamol)	

Paciente							
Id	DNI	Nombre	Primer apellido	Segundo apellido	Dirección	Teléfono	Número SIP
2	12533487A	María	Sebastián	Segarra	Av. Juan Llorens	621124353	4421115673
Patologías							
Id	Nombre	Es infecciosa	Es hereditaria	Código CIE	Medicamento para tratarlo		
Hospitalizaciones							
Id	Fecha alta	Fecha baja	Procede de urgencias	Habitación			Planta
2	2019-11-10T09:10:00	2019-11-10T21:15:40	TRUE	4			3 (Oncología)
Seguimientos							
Id	Fecha creación	Descripción	Enfermera				
3	2018-08-21T21:50:00	Síntomas de resfriado común. A la espera de las observaciones del médico.	4 (Pilar)				
Tratamientos							
Id	Fecha creación	Descripción	Fecha inicio	Fecha fin	Médico	Medicamento	
3	2019-11-10T17:00:00	Paciente con resfriado común. Se le recomienda Frenadol. Permanecerá en Observación unas horas.	2019-11-10T17:00:00	2019-11-15T22:00:00	2 (Marta)	2	

Paciente							
Id	DNI	Nombre	Primer apellido	Segundo apellido	Dirección	Teléfono	Número SIP
4	21735438Q	Javier	Iranzo	Sevilla	C. San Antonio	611728393	1923146513
Patologías							
	Id	Nombre	Es infecciosa	Es hereditaria	Código CIE	Medicamento para tratarlo	
	2	Fibrosis quística	FALSE	TRUE	E84.9		
	5	Cáncer de pulmón	FALSE	FALSE	C33		
Hospitalizaciones							
	Id	Fecha alta	Fecha baja	Procede de urgencias	Habitación		Planta
	4	2018-08-21T21:20:00	2018-08-22T09:45:40	FALSE	2		1 (Traumatología)
Seguimientos							
	Id	Fecha creación	Descripción	Enfermera			
Tratamientos							
	Id	Fecha creación	Descripción	Fecha inicio	Fecha fin	Médico	Medicamento