

Base de datos del equipamiento de un quirófano



Israel Brea Piñero, Alberto Enrique
García Mármol, Alejandro Inglés
Martínez, Manuel López Navarrete y
Rafael Prieto García.

ÍNDICE

Descripción del problema.....	2
Glosario de Términos.....	3
Catálogo de Requisitos.....	3
Usuarios del sistema.....	3
Requisitos generales.....	3
Requisitos detallados.....	4
Requisitos de Información.....	4
Requisitos de conducta.....	5
Reglas de Negocio.....	7
Modelo Conceptual y Escenarios	8
Modelo Conceptual	8
Diagrama de Objetos.....	9
Modelo Relacional.....	12
Justificación de la tercera forma normal.....	13
Código del proyecto.....	14

1.Descripción del problema

Las exigencias de cumplimiento de unos determinados estándares de calidad y asepsia en un quirófano llevan aparejado una adecuada gestión y control de todos sus elementos, de su conocimiento, mantenimiento y sustitución o reparación. Intervienen diferentes perfiles en esta función y deben poder acceder a la información de forma ordenada. Se plantea, para dar respuesta a esta situación, la creación de una base de datos para gestionar toda la información relativa a los equipos, instalaciones y equipamiento electro-médico que tiene un quirófano. El objetivo es que los tres perfiles de acceso (el ingeniero biomédico, el director de servicios generales del hospital y el ingeniero de mantenimiento) puedan acceder y conocer todos los elementos de nuestro quirófano, así como sus atributos, la empresa a la que han sido comprados, saber cada cuanto hay que hacer una revisión de la maquinaria o conocer los valores entre los que se debe mover nuestro equipo para que desempeñe un correcto funcionamiento (temperatura, voltaje...) entre otros. En el caso que estamos trabajando, se plantea crear una base de datos para un quirófano estándar, es decir, una base de datos general que podrá ser aplicada a cualquier tipo de quirófano, y que si es necesario podrá ser escalable para un tipo de quirófano más complejo. El equipamiento del que disponen los quirófanos se divide en:

Equipamiento electro-médico:

-3 elementos fundamentales:

- Columna de gases
- Luminaria (no referida a la iluminación como tal de la sala, sino a las lámparas utilizadas para operar).
- Panel técnico.

-De estos elementos solo se encargará el perfil del ingeniero biomédico, que va a ser la persona encargada tanto de su mantenimiento como de contactar con la empresa en el caso de que haga falta cambiar algún equipo del quirófano. La mayoría de las veces este ingeniero no va a formar parte de la clínica u hospital, sino que será un profesional de la empresa a la que le hayamos comprado las máquinas nos proporcionará sus servicios para que se encargue del mantenimiento y la gestión de estas.

Instalaciones:

-2 elementos fundamentales:

- Climatización.
- Elementos eléctricos.

-Con respecto a este punto, del mantenimiento de la parte de climatización y electricidad se encargará el ingeniero de mantenimiento, que en este caso sí que formará parte del hospital. Sin embargo, el que se encarga de reponer el equipo cuando este deje de funcionar o cuando haga falta uno más moderno es el director de servicios generales, responsable de las compras y adquisiciones.

2.Glosario de términos

-Columna de gases: Las tomas de gases medicinales del quirófano se ubican en este equipo. Son tomas de oxígeno, vacío, aire medicinal, etc. cercanas a la cama de intervención.

-Lámpara del quirófano: la lámpara quirúrgica es un equipo electro-médico que sirve para dotar a la cirugía de una iluminación adecuada de la zona donde se está operando, teniendo en cuenta las exigencias de temperatura, higiene (para que no entren partículas contaminantes en la zona de operación) e intensidad de luz entre otras.

-Panel técnico: Los paneles técnicos están diseñados para permitir el control centralizado sobre el estado del quirófano, además de suministrar gases medicinales y electricidad. Se trata de un equipo modular y adaptable a las necesidades del quirófano, con tamaños a medida y variedad de accesorios de control, comunicación, alumbrado e informático.

3.CATÁLOGO DE REQUISITOS

Usuarios del sistema:

Los tres usuarios (ingeniero de mantenimiento, ingeniero biomédico y director de servicios generales) tendrán acceso a la base de datos tanto para consultar sus datos como para modificarlos.

Existen multitud de usuarios posibles que podrán hacer uso de la base de datos, como por ejemplo, ingenieros biomédicos, ingenieros de mantenimiento o el director de servicios generales...

Requisitos generales:

- 1.Almacenar los datos personales del ingeniero biomédico encargado del mantenimiento de las máquinas de la empresa a la que pertenece.
- 2.Guardar toda la información del equipamiento electro-médico del que se dispone, así como información sobre las normativas de climatización y electricidad.
- 3.El sistema deberá ser capaz de detectar cualquier fallo en la información tanto del equipamiento electro-médico como en la electricidad y la climatización del quirófano.
- 4.El sistema deberá gestionar las revisiones de las instalaciones del quirófano y controlar su adecuado funcionamiento.
- 5.El sistema permitirá consultar la información de la base de datos y filtrarla a partir de los diferentes atributos; como por ejemplo el precio, el año de adquisición y el peso del equipamiento electro-médico, atributos personales, como nombre y apellidos y número de seguridad social.

Requisitos detallados:

Requisitos de información:

RI-001. Información sobre trabajadores: El sistema deberá registrar los datos personales sobre los ingenieros biomédicos. Nombre y apellidos, titulación, fecha de nacimiento, código postal, empresa a la que pertenecen, teléfono, email, número de la seguridad social y DNI.

RI-002. Información sobre empresas: El sistema registrará los datos sobre las empresas proveedoras del equipamiento electro-médico. Tales datos serán el CIF de la empresa, su nombre, dirección web, país de sede, dirección y teléfono.

RI-003. Información sobre equipamiento electro-médico: El sistema deberá registrar información sobre el equipamiento electro-médico: el precio, el año de adquisición, el peso, la vida útil del equipamiento, valores para su adecuado funcionamiento (valores máximos y mínimos de voltaje, temperatura o presión entre otros), empresa a la que pertenece, periodicidad de revisión, código del producto, fecha de la última y de la próxima revisión y una nota en función del estado de este.

El equipamiento electro-médico está formado por la columna quirúrgica, la lámpara y panel técnico.

RI-004. Información sobre la electricidad: El sistema deberá almacenar datos sobre la normativa vigente que dicte como debe ser la iluminación en un quirófano. Estos datos son los niveles máximo y mínimo de iluminación, así como la fecha de dicha normativa, y la periodicidad con que debe revisarse los sistemas que proporcionan luz a la sala.

RI-005. Información sobre la climatización: El sistema deberá almacenar datos sobre la normativa vigente que dicte como debe ser la climatización en un quirófano. Estos datos son los niveles máximo y mínimo del caudal de aire que entra, de temperatura y de humedad, así como la fecha de dicha normativa, y la periodicidad con que debe revisarse el sistema de climatización.

RI-006. Información sobre las revisiones: La base de datos almacenará datos sobre las revisiones que recibe tanto el equipamiento electro-médico como los sistemas de iluminación y climatización de nuestro quirófano. Estos datos serán las fechas de la última y la próxima revisión, su estado actual y el código del producto correspondiente a dicha revisión.

Además de todo esto, la base de datos cuenta con otra tabla de revisión auxiliar que almacena la información de aquellas revisiones que son defectuosas y el motivo del porqué.

Requisitos de conducta:

RC-001. Informe de empresa: La base de datos nos podrá proporcionar el nombre, los datos de contacto y la empresa a la que pertenecen todos los ingenieros biomédicos encargados de la gestión de nuestro equipamiento.

RC-002. Informe de equipamiento electro-médico: La base de datos nos podrá proporcionar los datos relativos a nuestro equipamiento electro-médico actualmente en uso.

RC-003. Ingeniero con más revisiones: El sistema nos podrá informar de quien es el ingeniero que más revisiones ha realizado para poder monitorizar su rendimiento, por ejemplo.

RC-004. Informe de las medidas realizadas al quirófano: El sistema nos deberá informar sobre si las mediciones obtenidas en una revisión están dentro de los intervalos entre los que deben estar comprendidas para el correcto funcionamiento de nuestro equipamiento electro-médico, las instalaciones de climatización y electricidad.

RC-005. Coste de las instalaciones: El sistema deberá proporcionarnos una suma de los precios de los elementos del equipamiento que conforme nuestro quirófano en ese momento.

RC-006. Tiempo restante de vida útil: El sistema podrá decirnos cuanto tiempo de vida útil les queda a las máquinas de nuestro equipo.

RC-007. Fecha próxima revisión: El sistema calculará la fecha de la próxima revisión en función de la última, y de la periodicidad con la que estas se llevan a cabo.

RC-008. Rebaja de equipamiento: El sistema calculará una rebaja de un 60% sobre el precio original de los equipos que le queden un año o menos para alcanzar el fin de su vida útil, por ejemplo, para su futura venta a instituciones académicas.

RC-009. Retraso de las revisiones: Si por algún motivo algún equipo no puede recibir una revisión en la fecha prevista, el sistema nos permitirá atrasarla 5 días.

RC-010. Acotación de los valores de luminosidad: El sistema puede acotar los valores de luminosidad de la sala aumentando el nivel mínimo y disminuyendo el nivel máximo ambos en un 5%.

RC-011. Cambio de los valores mínimos de la normativa de climatización: El sistema nos permite aumentar en un 15% todos los valores mínimos de la normativa de climatización vigente (caudal de aire, temperatura y porcentaje de humedad).

RC-012. Aumento de la frecuencia de las revisiones: El sistema permite aumentar la frecuencia de las revisiones de las normativas de iluminación y climatización vigentes disminuyendo en un 10% la periodicidad de estas.

RC-013. Peor estado de un equipo: El sistema puede proporcionar la peor nota recibida por un determinado equipamiento electro-médico.

RC-014. Revisiones por debajo del 5 de un ingeniero: El sistema nos proporcionará el número de revisiones que haya llevado a cabo un ingeniero obteniendo una nota inferior al 5, para así determinar si estos desempeñan correctamente su trabajo o si alguna empresa está proporcionando maquinaria que no rinde de manera óptima.

RC-015. Lista de proveedores: El sistema nos podrá proporcionar una lista con todas las empresas que provean nuestro quirófano en ese momento.

RC-016. Revisión de cada máquina: Si queremos consultar la fecha de la próxima revisión de alguna máquina en concreto, el sistema deberá proporcionárnosla introduciendo el código de dicha máquina.

RC-017. Dinero invertido por empresa: El sistema nos podrá informar del número de máquinas compradas y de cuánto dinero hemos gastado en cada una de las empresas proveedoras de maquinaria electro-médica, así como el precio medio por equipo ordenado de mayor a menor.

RC-018. Empresa de cada ingeniero: Introduciendo el DNI de un ingeniero la base de datos nos proporcionará la empresa a la que pertenece.

RC-019. Precios del equipamiento: El sistema nos podrá informar sobre el precio mínimo, máximo y medio de las lámparas, columnas y paneles que conformen nuestro quirófano.

RC-020. Peso total: El sistema podrá calcular el peso actual de nuestro quirófano.

RC-021. Número de revisiones: El sistema nos podrá calcular cuantas revisiones ha recibido cada elemento de nuestro equipo para saber si están recibiendo un análisis apropiado.

Además, podrá informarnos sobre cuál ha sido el equipo que ha recibido más revisiones.

RC-022. Equipamiento más nuevo: El sistema podrá proporcionarnos una lista con todos los elementos de nuestro equipo que hayan sido adquiridos en los últimos 3 años para así saber qué máquinas son las menos susceptibles de necesitar un reemplazo.

RC-023. Malas revisiones recientes: El sistema podrá decirnos el código del producto y la fecha de adquisición de aquel elemento del sistema que en la última revisión haya obtenido una nota inferior a siete para saber si esta nota se corresponde con la antigüedad de la máquina.

RC-024. Ingenieros al cargo: El sistema nos permitirá seleccionar una empresa y saber el código de las máquinas que tenemos compradas a dicha empresa, y conocer los datos de los ingenieros biomédicos encargados del mantenimiento de estas.

También podremos conocer los datos de todos los ingenieros biomédicos que estén trabajando con nosotros, y conocer el número de revisiones que ha realizado cada uno.

RC-025. Precio del quirófano en una determinada fecha: Dada una determinada fecha el sistema nos deberá ofrecer el precio del quirófano en esa determinada fecha.

RC-026.Diferencia de precio: Dado un determinado año, la base de datos nos calculará la diferencia de precio del quirófano en ese entonces con respecto al actual.

RC-027. Normativa más exigente: El sistema devolverá la normativa con los criterios más exigentes de climatización.

Reglas de negocios:

RN-001. Las revisiones no pueden tener lugar 5 días después de la fecha que se preveía.

RN-002. El sistema debe garantizar que los equipos sean repuestos antes de superar su vida útil. Para nos mandará un aviso 3 meses antes de cumplirla.

RN-003. El valor del atributo estado de las revisiones debe ser un numero comprendido entre 0 y 10 (con una precisión de un decimal).

RN-004. El código de producto debe comenzar por la primera letra de su nombre (Lámpara-L, Columna-C y Panel-P).

RN-005. Si en una revisión, la nota varia 3 puntos por debajo de la nota anterior o más, se produce un aviso. Esta revisión será archivada en la tabla revisión auxiliar.

RN-006. Si algún elemento del equipamiento recibe una nota de un tres o inferior, será reemplazada. Esta revisión será archivada en la tabla revisión auxiliar.

RN-007. Si algún elemento del equipamiento recibe en tres revisiones consecutivas una nota inferior a 5.5, será reemplazada por una nueva. Esta revisión será archivada en la tabla revisión auxiliar.

RN-008. El equipamiento que conforme nuestro quirófano no podrá ser proveído en su totalidad por la misma empresa (debe al menos haber dos proveedores por quirófano).

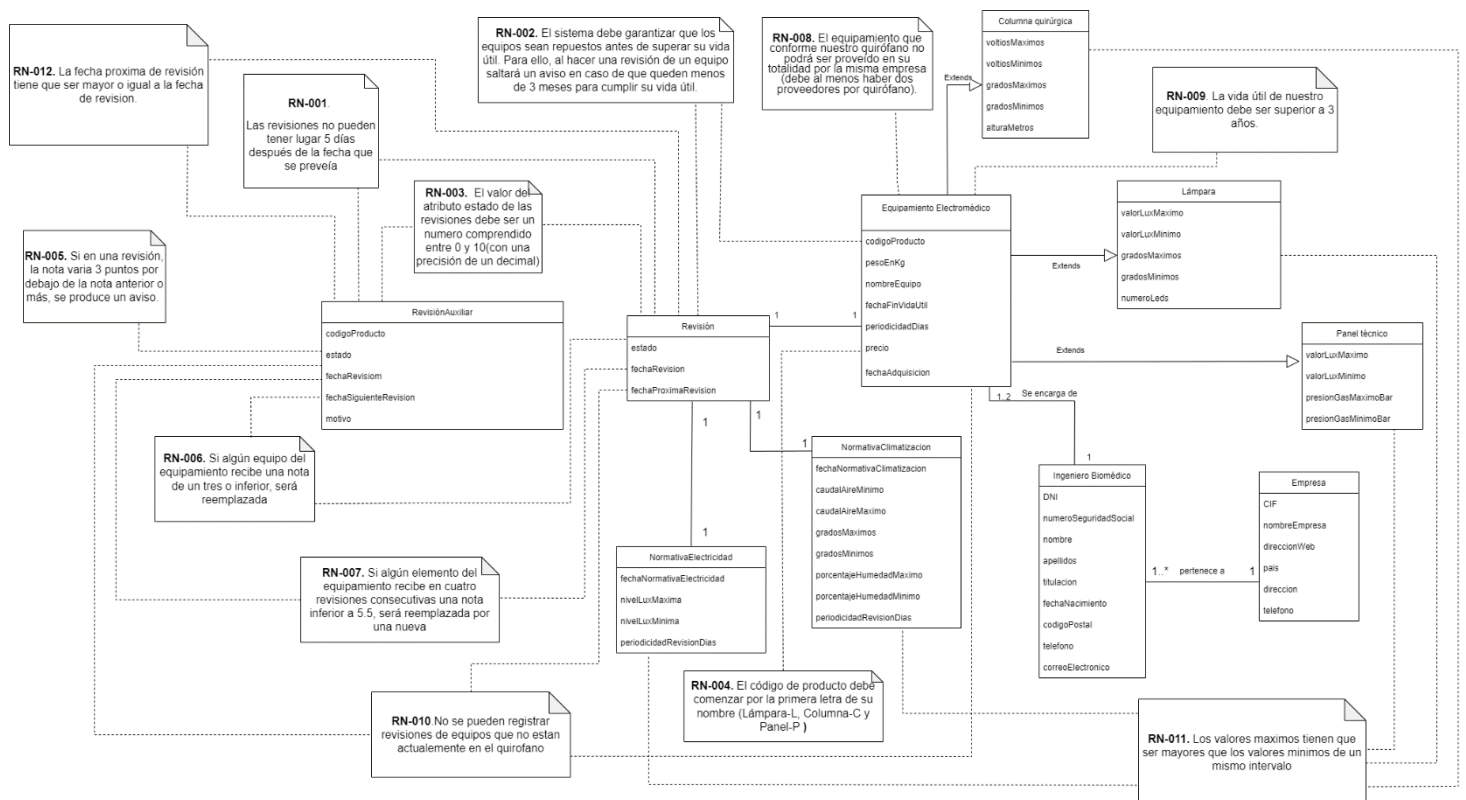
RN-009. La vida útil de nuestro equipamiento debe ser superior a 3 años.

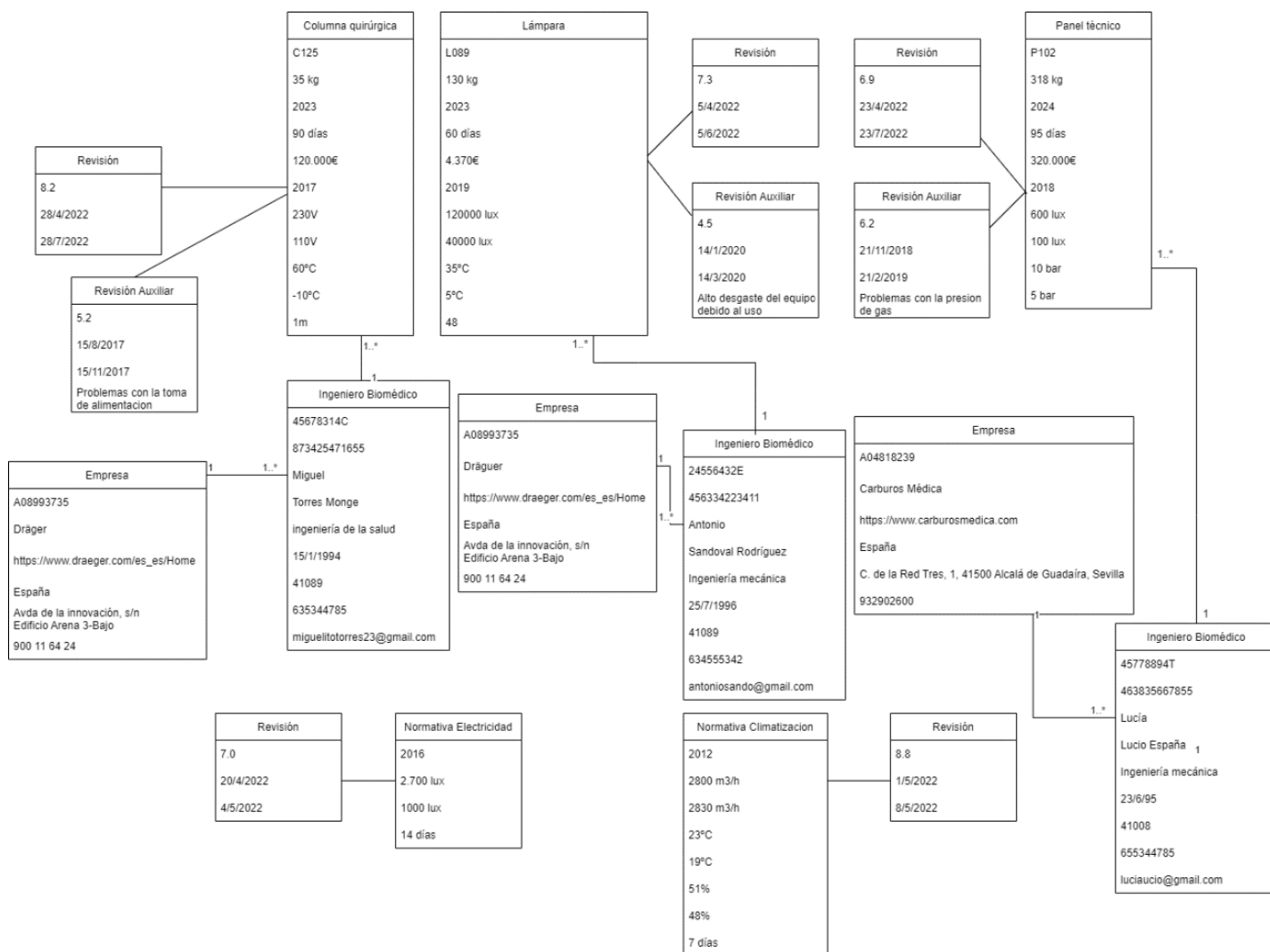
RN-010. No se pueden registrar revisiones de equipos que no están actualmente en el quirófano.

RN-011. Los valores máximos tienen que ser mayores que los valores mínimos de un mismo intervalo.

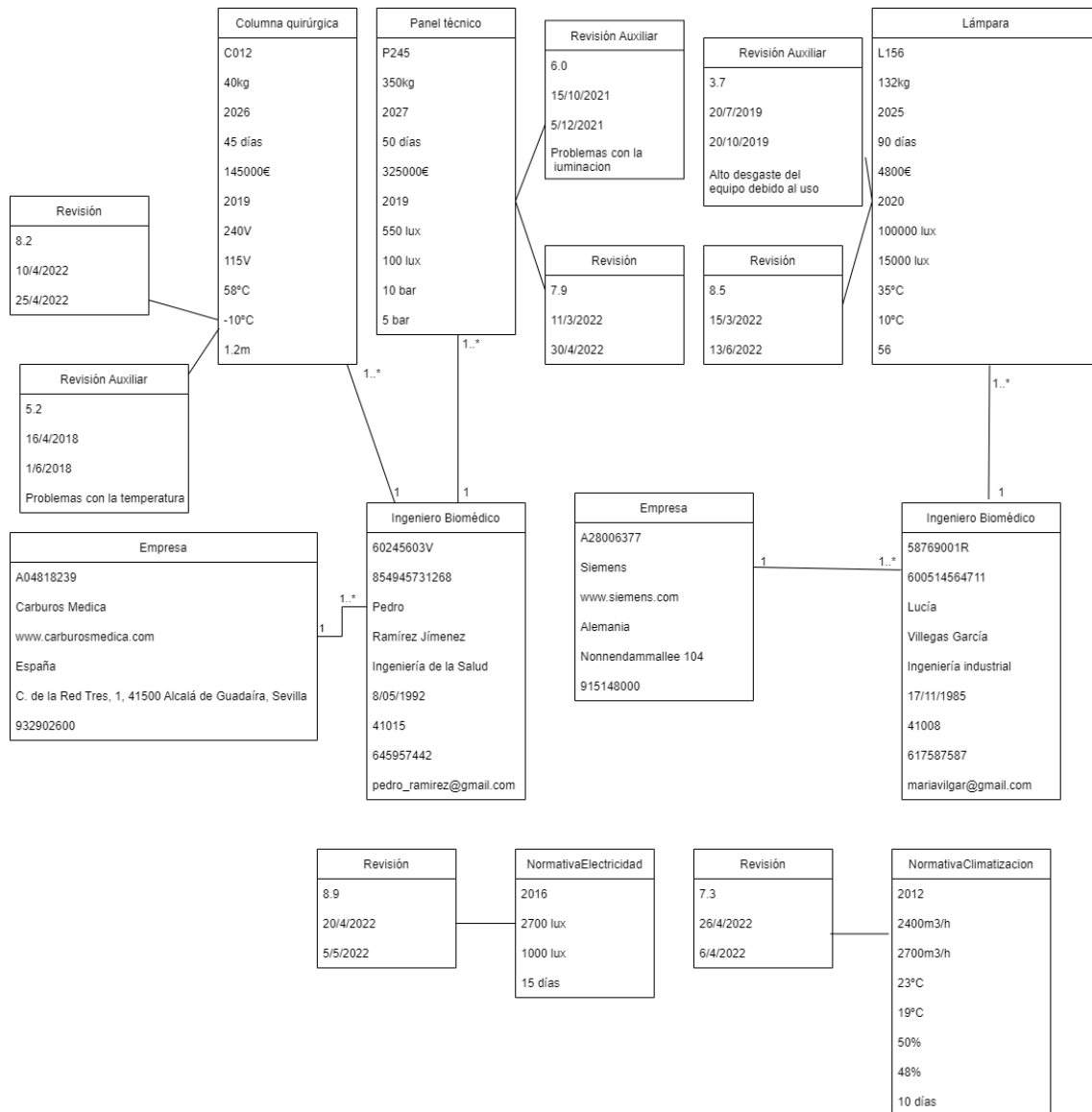
RN-012. La fecha próxima de revisión tiene que ser mayor o igual a la fecha de revisión.

Diagrama conceptual



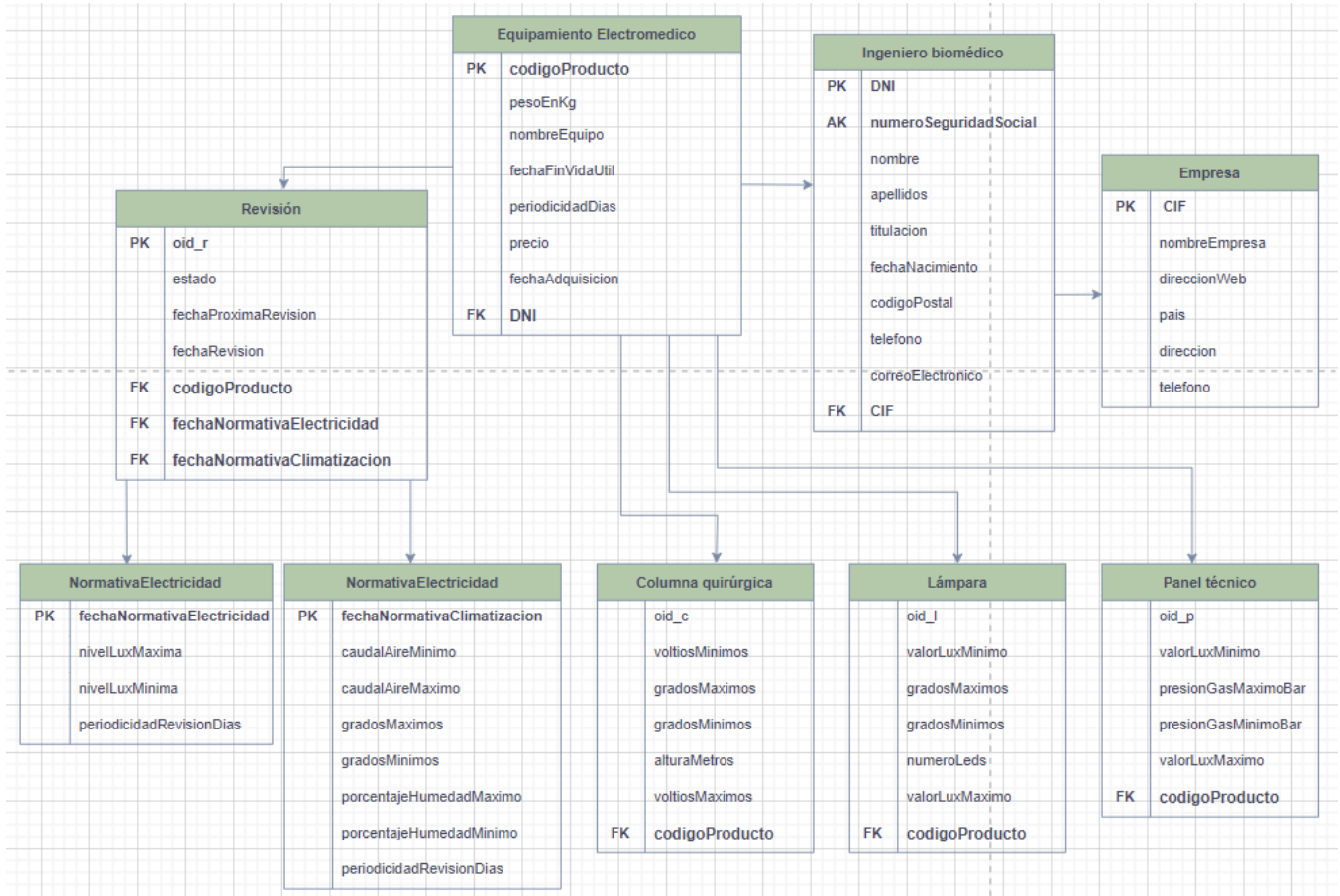


Ensayo de prueba 1. Quirófano del Hospital Universitario Puerta del Mar.



Escenario de prueba 2. Quirófano del Hospital San Carlos.

Modelo relacional



Justificación de la tercera forma normal

Para comprobar que todas nuestras relaciones están en 3FN, deberemos empezar por la primera forma normal.

Una relación está en 1FN si en cada tupla se le asigna por cada atributo un solo valor, esto implica la ausencia de grupos repetidos.

codigoProducto	pesoEnKg	nombreEquipo	fechaFinVidaUtil	periodicidadDias	precio	fechaAdquisicion	dni
P101	318	Panel Técnico	2022-06-23	90	128.000,00	2021-09-28	82746372J
P100	290	Panel Técnico	2021-12-03	90	134.000,00	2012-07-15	32536784Z
L800	123	Lámpara	2023-01-23	20	4.000,00	2019-01-12	60245603V
L090	120	Lámpara	2029-02-11	60	5.000,00	2021-10-29	45678314C
L089	130	Lámpara	2024-11-13	60	4.370,00	2011-05-13	60245603V
C129	123	Columna Quirúrgica	2028-12-23	20	1.200,00	2017-01-12	60245603V
C128	1.200	Columna Quirúrgica	2028-01-08	20	17.500,00	2021-10-09	56734268Y
C127	1.300	Columna Quirúrgica	2025-11-12	20	19.000,00	2012-05-12	32536784Z

Por ejemplo, es nuestra tabla de equipamiento electro-médico para cada atributo hay asignado un solo valor. Por lo tanto, se cumple la 1FN.

A continuación, deberemos asegurarnos de que nuestras relaciones estén en 2FN.

Una relación está en segunda forma normal si:

- Está en 1FN y todos los atributos no primos son completamente dependientes de las claves candidatas de la relación (Los atributos no primos son los que no forman parte de ninguna clave candidata).

cif	nombreEmpresa	direccionWeb	pais	direccion	telefono
A04818239	Carburos Medica	www.carburosmedica.com	España	C. de la Red Tres 1	932902600
A08993735	Novaclinic	www.novaclinic.com	España	Avda de la innovación, s/n Edificio Arena 3-Bajo	900116424
A28006377	Siemens	www.siemens.com	Alemania	Nonnendammallee 104	915148000
A28063485	Dräger	www.dräger.com	Alemania	Revalstrabe	902116424
A34568975	novaclinic	novaclinicweb	España	Carpestro, 14	601234556

dni	numeroSeguridadSocial	nombre	apellidos	titulacion	fechaNacimiento	codigoPostal	telefono	correoElectronico	cif
12345678A	123344455	Carlos	Ramirez Manzano	Ingeniería de la Salud	1998-02-21	11100	644890789	carlitosramirez98@gmail.com	A08993735
32536784Z	746283492837	Maria José	Navarrete Inglés	Ingeniería Médica	1990-02-22	41928	672687946	marianavaing@gmail.es	A28063485
45678314C	837462938413	Miguel	Torres Monge	Ingeniería de la Salud	1994-01-15	41089	655344785	miguelitortorres23@gmail.com	A08993735
56734268Y	384754857384	Pepe	Manzano Olivo	Ingeniería Médica	1992-09-26	41920	672687946	pipon@gmail.com	A28006377
60245603V	432873912391	Pedro	Ramirez Jimenez	Ingeniería de la Salud	1992-05-08	41015	645957442	pedro_ramirez@gmail.com	A04818239
82746372J	782468857287	Rosa	Melinda Giménez	Ingeniería Médica	1991-05-02	41929	672836152	rositameli91@gmail.com	A08993735

oid_r	estado	fechaRevision	fechaProximaRevision	codigoProducto	fechaNormativaElectricidad	fechaNormativaClimatizacion
71	8,4	2021-12-30	2022-01-13	(NULL)	2.018	(NULL)
44	9,2	2021-08-14	2021-08-28	(NULL)	2.018	(NULL)
41	9,0	2021-07-30	2021-08-14	(NULL)	2.018	(NULL)
40	8,9	2021-07-23	2021-08-17	(NULL)	(NULL)	2.014
48	8,6	2021-09-11	2021-09-25	(NULL)	2.018	(NULL)
38	8,9	2021-07-16	2021-07-30	(NULL)	2.018	(NULL)
37	9,1	2021-07-02	2021-07-16	(NULL)	2.018	(NULL)
35	8,7	2021-06-28	2021-07-23	(NULL)	(NULL)	2.014
33	9,4	2021-06-18	2021-07-02	(NULL)	2.018	(NULL)
30	9,5	2021-06-04	2021-06-18	(NULL)	2.018	(NULL)
29	8,8	2021-06-03	2021-06-28	(NULL)	(NULL)	2.014
45	8,7	2021-08-17	2021-09-11	(NULL)	(NULL)	2.014
46	8,8	2021-08-28	2021-09-11	(NULL)	2.018	(NULL)
49	8,7	2021-09-11	2021-10-06	(NULL)	(NULL)	2.014
67	8,5	2021-12-16	2021-12-30	(NULL)	2.018	(NULL)
64	9,0	2021-12-02	2021-12-16	(NULL)	2.018	(NULL)
62	8,4	2021-11-18	2021-12-02	(NULL)	2.018	(NULL)
61	9,0	2021-11-25	2021-12-20	(NULL)	(NULL)	2.014
60	9,1	2021-11-04	2021-11-18	(NULL)	2.018	(NULL)
59	9,2	2021-10-31	2021-11-25	(NULL)	(NULL)	2.014
56	9,0	2021-10-23	2021-11-04	(NULL)	2.018	(NULL)
54	8,9	2021-10-09	2021-10-23	(NULL)	2.018	(NULL)
53	8,8	2021-10-06	2021-10-31	(NULL)	(NULL)	2.014
51	9,3	2021-09-25	2021-10-09	(NULL)	2.018	(NULL)
28	9,1	2021-05-21	2021-06-04	(NULL)	2.018	(NULL)
26	8,9	2021-05-09	2021-06-03	(NULL)	(NULL)	2.014
22	8,7	2021-04-14	2021-05-09	(NULL)	(NULL)	2.014
16	9,4	2021-03-20	2021-04-14	(NULL)	(NULL)	2.014
14	9,4	2021-03-12	2021-03-26	(NULL)	2.018	(NULL)
13	9,5	2021-02-25	2021-03-12	(NULL)	2.018	(NULL)

codigoProducto	estado	fechaRevision	fechaSiguienteRevision	motivo
C127	4,0	2021-10-09	2021-10-29	Alto desgaste del equipo debido al uso
L089	3,5	2021-10-10	2021-12-09	Alto desgaste del equipo debido al uso
P100	4,2	2021-09-28	2021-12-27	Alto desgaste del equipo debido al uso
C129	2,0	2018-10-10	2018-10-30	Problemas con la toma de alimentacion
C128	5,2	2022-04-10	2022-04-30	Este equipo lleva teniendo 3 revisiones seguidas una...


codigoProducto	estado	fechaRevision	fechaSiguienteRevision	motivo
C127	4,0	2021-10-09	2021-10-29	Alto desgaste del equipo debido al uso
L089	3,5	2021-10-10	2021-12-09	Alto desgaste del equipo debido al uso
P100	4,2	2021-09-28	2021-12-27	Alto desgaste del equipo debido al uso
C129	2,0	2018-10-10	2018-10-30	Problemas con la toma de alimentacion
C128	5,2	2022-04-10	2022-04-30	Este equipo lleva teniendo 3 revisiones seguidas una...

Como se puede apreciar en las imágenes, todos los valores de los atributos dependen directamente de la clave primaria. Por ejemplo, con respecto a la tabla de equipamiento electro-médico, el precio, el peso, el nombre del equipo, la fecha de adquisición, etc. Dependen directamente de cada equipo, es decir, de su código del producto.

Todas nuestras relaciones cumplen los requisitos de la 2FN.

Por último, nos aseguraremos de que nuestras relaciones están en 3FN.

Una relación se encuentra en tercera forma normal si está en 2FN y ningún atributo no primo depende transitivamente de ninguna clave candidata.

oid_I 	valorLuxMaximo	valorLuxMinimo	gradosMaximos	gradosMinimos	numeroLeds	codigoProducto 
1	1.200	400	55	5	48	L089
2	1.600	400	60	10	14	L090

La 3FN se cumple porque se cumple la 2FN y porque todas nuestras relaciones cumplen que todos los atributos dependen directamente de la clave primaria, es decir, no existe ningún atributo que haga de intermediario entre la relación directa de la clave primaria y el resto de los atributos de nuestra relación.

Código del trabajo:

1.Tablas:

```
-- CREATE TABLE
```

```
CREATE DATABASE if NOT EXISTS quirofano;  
USE quirofano;
```

```
DROP TABLE if EXISTS panelTecnico;  
DROP TABLE if EXISTS lampara;  
DROP TABLE if EXISTS columnaQuirurgica;  
DROP TABLE if EXISTS revision;  
DROP TABLE if EXISTS equipamientoElectromedico;  
DROP TABLE if EXISTS ingenieroBiomedico;  
DROP TABLE if EXISTS normativaClimatizacion;  
DROP TABLE if EXISTS normativaElectricidad;  
DROP TABLE if EXISTS empresa;  
DROP TABLE if EXISTS revisionAuxiliar;
```

```
CREATE TABLE if NOT EXISTS empresa (  
cif VARCHAR(10) PRIMARY KEY UNIQUE,  
nombreEmpresa VARCHAR(20) NOT null,  
direccionWeb VARCHAR(50),  
pais VARCHAR(20),  
direccion VARCHAR(100),  
telefono VARCHAR(14) );
```

```
CREATE TABLE if NOT EXISTS normativaElectricidad (  
fechaNormativaElectricidad INT PRIMARY KEY,  
nivellLuxMaxima INT,  
nivellLuxMinima INT,  
periodicidadRevisionDias INT,  
CONSTRAINT intervalosConSentido CHECK  
(nivellLuxMaxima>nivellLuxMinima));
```

```
CREATE TABLE if NOT EXISTS normativaClimatizacion (  
fechaNormativaClimatizacion INT PRIMARY KEY,  
caudalAireMinimo INT,  
caudalAireMaximo INT,  
gradosMaximos INT,  
gradosMinimos INT,  
porcentajeHumedadMaximo INT,  
porcentajeHumedadMinimo INT,  
periodicidadRevisionDias INT ,  
CONSTRAINT intervalosConSentido2 CHECK  
(caudalAireMaximo>caudalAireMinimo AND gradosMaximos>gradosMinimos AND  
porcentajeHumedadMaximo>porcentajeHumedadMinimo));
```

```

CREATE TABLE if NOT EXISTS ingenieroBiomedico (
dni VARCHAR(9) PRIMARY KEY UNIQUE,
numeroSeguridadSocial VARCHAR(15) UNIQUE,
nombre VARCHAR(50) NOT NULL,
apellidos VARCHAR(50) NOT NULL,
titulacion VARCHAR(30),
fechaNacimiento DATE,
codigoPostal VARCHAR(5) ,
telefono VARCHAR(14),
correoElectronico VARCHAR(60),
cif VARCHAR(9) ,
FOREIGN KEY (cif) REFERENCES empresa (cif) );

```

```

CREATE TABLE if NOT EXISTS equipamientoElectromedico (
codigoProducto VARCHAR(4) PRIMARY KEY UNIQUE,
-- CONSTRAINT restricciónCodigoP CHECK (codigoProducto),
pesoEnKg INT NOT null,
nombreEquipo VARCHAR(20),
fechaFinVidaUtil DATE NOT null,
periodicidadDias INT NOT null,
precio DECIMAL(8,2) NOT null,
fechaAdquisición DATE NOT null,
dni VARCHAR(9) ,
FOREIGN KEY (dni) REFERENCES ingenieroBiomedico (dni) );

```

-- Rgela de negocio: el estado de las revisiones tiene que ir entre 0 y 10.

```

CREATE TABLE if NOT EXISTS revision (
oid_r INT AUTO_INCREMENT PRIMARY KEY,
estado DECIMAL(3,1),
CONSTRAINT calificaciónEstado CHECK (estado<=10 AND estado>=0),
fechaRevision DATE DEFAULT NOW(),
fechaProximaRevision DATE,
CONSTRAINT fechasConSentido1 CHECK
(fechaProximaRevision>=fechaRevision),
codigoProducto VARCHAR(20) ,
fechaNormativaElectricidad INT,
fechaNormativaClimatización INT,
FOREIGN KEY (codigoProducto) REFERENCES equipamientoElectromedico
(codigoProducto),
FOREIGN KEY (fechaNormativaElectricidad) REFERENCES
normativaElectricidad (fechaNormativaElectricidad),
FOREIGN KEY (fechaNormativaClimatización) REFERENCES
normativaClimatización (fechaNormativaClimatización) );

```

```

CREATE TABLE if NOT EXISTS columnaquirurgica (
oid_c INT AUTO_INCREMENT UNIQUE,

```

```

voltiosMaximos INT,
voltiosMinimos INT,
gradosMaximos INT,
gradosMinimos INT,
alturaMetros INT,
codigoProducto VARCHAR(20),
FOREIGN KEY (codigoProducto) REFERENCES equipamientoElectromedico
(codigoProducto),
CONSTRAINT intervalosConSentido3 CHECK
(voltiosMaximos>voltiosMinimos AND gradosMaximos>gradosMinimos) );

```

```

CREATE TABLE if NOT EXISTS lampara (
oid_l INT AUTO_INCREMENT UNIQUE,
valorLuxMaximo INT,
valorLuxMinimo INT,
gradosMaximos INT,
gradosMinimos INT,
numeroLeds INT,
codigoProducto VARCHAR(20),
FOREIGN KEY (codigoProducto) REFERENCES equipamientoElectromedico
(codigoProducto),
CONSTRAINT intervalosConSentido4 CHECK
(valorLuxMaximo>valorLuxMinimo AND gradosMaximos>gradosMinimos));

```

```

CREATE TABLE if NOT EXISTS paneltecnico (
oid_p INT AUTO_INCREMENT UNIQUE,
valorLuxMaximo INT,
valorLuxMinimo INT,
presionGasMaximoBar INT,
presionGasMinimoBar INT,
codigoProducto VARCHAR(20),
FOREIGN KEY (codigoProducto) REFERENCES equipamientoElectromedico
(codigoProducto),
CONSTRAINT intervalosConSentido5 CHECK
(valorLuxMaximo>valorLuxMinimo AND
presionGasMaximoBar>presionGasMinimoBar) );

```

```

CREATE TABLE if NOT EXISTS revisionauxiliar (
codigoProducto VARCHAR(20),
estado DECIMAL (3,1),
CONSTRAINT calificacionEstado CHECK (estado<=10 AND estado>=0),
fechaRevision DATE DEFAULT NOW(),
fechaSiguienteRevision DATE,
motivo VARCHAR(200),
CONSTRAINT fechaConSentido CHECK
(fechaSiguienteRevision>=fechaRevision));

```

2.Sentencias insert:

-- INSERT

```
INSERT INTO empresa(cif, nombreEmpresa, direccionWeb, pais, direccion, telefono)
```

```
VALUES ('A28063485', 'Dräger', 'www.dräger.com', 'Alemania', 'Revalstrabe', 902116424), ('A04818239', 'Carbueros Medica', 'www.carbuerosmedica.com', 'España', 'C. de la Red Tres 1', 932902600), ('A08993735', 'Novacclinic', 'www.novacclinic.com', 'España', 'Avda de la innovación, s/n Edificio Arena 3-Bajo', 900116424), ('A28006377', 'Siemens', 'www.siemens.com', 'Alemania', 'Nonnendammallee 104', 915148000);
```

```
INSERT INTO normativaElectricidad(nivelLuxMaxima, nivelLuxMinima, fechaNormativaElectricidad, periodicidadRevisionDias)
```

```
VALUES (2700, 1000, 2016, 14), (2700, 1000, 2017, 7), (2700, 1000, 2018, 14);
```

```
INSERT INTO normativaClimatizacion(caudalAireMinimo, caudalAireMaximo, gradosMaximos, gradosMinimos, porcentajeHumedadMinimo, PorcentajeHumedadMaximo, fechaNormativaClimatizacion, periodicidadRevisionDias)
```

```
VALUES (2700, 3500, 23, 19, 48, 50, 2012, 20), (2830, 4000, 24, 19, 47, 50, 2013, 22), (2830, 3980, 23, 19, 48, 56, 2014, 25);
```

```
INSERT INTO IngenieroBiomedico(dni, numeroSeguridadSocial, nombre, apellidos, titulacion, fechaNacimiento, codigoPostal, telefono, correoElectronico, cif)
```

```
VALUES ('32536784Z', '746283492837', 'Maria José', 'Navarrete Inglés', 'Ingeniería Médica', '1990-02-22', 41928, 672687946, 'marianavaing@gmail.es', 'A28063485'), ('60245603V', '432873912391', 'Pedro', 'Ramírez Jiménez', 'Ingeniería de la Salud', '1992-05-8', 41015, 645957442, 'pedro_ramirez@gmail.com', 'A04818239'), ('45678314C', '837462938413', 'Miguel', 'Torres Monge', 'Ingeniería de la Salud', '1994-1-15', 41089, 655344785, 'miguelitotorres23@gmail.com', 'A08993735'), ('56734268Y', '384754857384', 'Pepe', 'Manzano Olivo', 'Ingeniería Médica', '1992-09-26', 41920, 672687946, 'papon@gmail.com', 'A28006377'), ('82746372J', '782468857287', 'Rosa', 'Melinda Giménez', 'Ingeniería Médica', '1991-05-2', 41929, 672836152, 'rositameli91@gmail.com', 'A08993735');
```

```
INSERT INTO equipamientoElectromedico(codigoProducto, pesoEnKg, nombreEquipo, fechaFinVidaUtil, periodicidadDias, precio, fechaAdquisicion, dni)
```

```
VALUES
```

```

('L089', 130, 'Lámpara', '2024-11-13', 60,
4370, '2011-5-13', '60245603V'),
('L090', 120, 'Lámpara', '2029-2-11', 60,
5000, '2021-10-29', '45678314C'),
('C127', 1300, 'Columna Quirúrgica', '2025-11-12', 20,
19000.0, '2012-5-12', '32536784Z'),
('C128', 1200, 'Columna Quirúrgica', '2028-1-8', 20,
17500.0, '2021-10-9', '56734268Y'),
('P100', 290, 'Panel Técnico', '2021-12-3', 90,
335000, '2012-7-15', '32536784Z'),
('P101', 318, 'Panel Técnico', '2022-6-23', 90,
320000, '2021-9-28', '82746372J');

```

```

INSERT INTO revision(oid_r, estado, fechaRevision,
fechaProximaRevision, codigoProducto,
fechaNormativaElectricidad, fechaNormativaClimatizacion)

```

VALUES

```

(1, 9.6, '2021-1-1', '2021-1-14', NULL, 2018, NULL),
(2, 5.6, '2021-1-2', '2021-3-31', 'P100', NULL, NULL),
(3, 9.3, '2021-1-4', '2021-1-29', NULL, NULL, 2014),
(4, 9.7, '2021-1-14', '2021-1-28', NULL, 2018, NULL),
(5, 6.5, '2021-1-16', '2021-2-5', 'C127', NULL, NULL),
(6, 9.8, '2021-1-28', '2021-2-11', NULL, 2018, NULL),
(7, 9.5, '2021-1-29', '2021-2-23', NULL, NULL, 2014),
(8, 6.3, '2021-2-5', '2021-2-25', 'C127', NULL, NULL),
(9, 9.8, '2021-2-11', '2021-2-25', NULL, 2018, NULL),
(10, 5.4, '2021-2-12', '2021-4-13', 'L089', NULL, NULL),
(11, 9.2, '2021-2-23', '2021-3-20', NULL, NULL, 2014),
(12, 6.3, '2021-2-25', '2021-3-17', 'C127', NULL, NULL),
(13, 9.5, '2021-2-25', '2021-3-12', NULL, 2018, NULL),
(14, 9.4, '2021-3-12', '2021-3-26', NULL, 2018, NULL),
(15, 6.1, '2021-3-17', '2021-4-8', 'C127', NULL, NULL),
(16, 9.4, '2021-3-20', '2021-4-14', NULL, NULL, 2014),
(17, 9.6, '2021-3-26', '2021-4-9', NULL, 2018, NULL),
(18, 5.3, '2021-3-31', '2021-6-30', 'P100', NULL, NULL),
(19, 6, '2021-4-8', '2021-4-28', 'C127', NULL, NULL),
(20, 9.7, '2021-4-9', '2021-4-23', NULL, 2018, NULL),
(21, 6.2, '2021-4-13', DATE_ADD(fechaRevision, INTERVAL 60 DAY), 'L089',
NULL, NULL),
(22, 8.7, '2021-4-14', '2021-5-9', NULL, NULL, 2014),
(23, 9, '2021-4-23', '2021-5-7', NULL, 2018, NULL),
(24, 5.8, '2021-4-28', '2021-5-18', 'C127', NULL, NULL),
(25, 9.2, '2021-5-7', '2021-5-21', NULL, 2018, NULL),
(26, 8.9, '2021-5-9', '2021-6-3', NULL, NULL, 2014),
(27, 5.6, '2021-5-18', '2021-6-7', 'C127', NULL, NULL),
(28, 9.1, '2021-5-21', '2021-6-4', NULL, 2018, NULL),
(29, 8.8, '2021-6-3', '2021-6-28', NULL, NULL, 2014),
(30, 9.5, '2021-6-4', '2021-6-18', NULL, 2018, NULL),
(31, 5.8, '2021-6-7', '2021-6-27', 'C127', NULL, NULL),
(32, 6, '2021-6-12', '2021-8-11', 'L089', NULL, NULL),

```

```

(33, 9.4, '2021-6-18', '2021-7-2', NULL, 2018, NULL),
(34, 5.6, '2021-6-27', '2021-7-17', 'C127', NULL, NULL),
(35, 8.7, '2021-6-28', '2021-7-23', NULL, NULL, 2014),
(36, 5.3, '2021-6-30', '2021-9-28', 'P100', NULL, NULL),
(37, 9.1, '2021-7-2', '2021-7-16', NULL, 2018, NULL),
(38, 8.9, '2021-7-16', '2021-7-30', NULL, 2018, NULL),
(39, 5, '2021-7-17', '2021-8-6', 'C127', NULL, NULL),
(40, 8.9, '2021-7-23', '2021-8-17', NULL, NULL, 2014),
(41, 9, '2021-7-30', '2021-8-14', NULL, 2018, NULL),
(42, 5.2, '2021-8-10', '2021-8-30', 'C127', NULL, NULL),
(43, 5.4, '2021-8-11', '2021-10-10', 'L089', NULL, NULL),
(44, 9.2, '2021-8-14', '2021-8-28', NULL, 2018, NULL),
(45, 8.7, '2021-8-17', '2021-9-11', NULL, NULL, 2014),
(46, 8.8, '2021-8-28', '2021-9-11', NULL, 2018, NULL),
(47, 5.4, '2021-8-30', '2021-9-19', 'C127', NULL, NULL),
(48, 8.6, '2021-9-11', '2021-9-25', NULL, 2018, NULL),
(49, 8.7, '2021-9-11', '2021-10-6', NULL, NULL, 2014),
(50, 5, '2021-9-19', '2021-10-9', 'C127', NULL, NULL),
(51, 9.3, '2021-9-25', '2021-10-9', NULL, 2018, NULL),
(52, 9.5, '2021-9-28', '2021-12-27', 'P101', NULL, NULL),
(53, 8.8, '2021-10-6', '2021-10-31', NULL, NULL, 2014),
(54, 8.9, '2021-10-9', '2021-10-23', NULL, 2018, NULL),
(55, 9.9, '2021-10-10', '2021-12-9', 'L090', NULL, NULL),
(56, 9, '2021-10-23', '2021-11-4', NULL, 2018, NULL),
(57, 9.8, '2021-10-12', '2021-10-30', 'C128', NULL, NULL),
(58, 9.5, '2021-10-30', '2021-11-19', 'C128', NULL, NULL),
(59, 9.2, '2021-10-31', '2021-11-25', NULL, NULL, 2014),
(60, 9.1, '2021-11-4', '2021-11-18', NULL, 2018, NULL),
(61, 9, '2021-11-25', '2021-12-20', NULL, NULL, 2014),
(62, 8.4, '2021-11-18', '2021-12-2', NULL, 2018, NULL),
(63, 9.4, '2021-11-19', '2021-12-9', 'C128', NULL, NULL),
(64, 9, '2021-12-2', '2021-12-16', NULL, 2018, NULL),
(65, 9.4, '2021-12-9', '2021-12-29', 'C128', NULL, NULL),
(66, 9.9, '2021-12-9', '2022-2-7', 'L090', NULL, NULL),
(67, 8.5, '2021-12-16', '2021-12-30', NULL, 2018, NULL),
(68, 9.1, '2021-12-20', '2022-1-14', NULL, NULL, 2014),
(69, 9.4, '2021-12-27', '2022-2-28', 'P101', NULL, NULL),
(70, 9.4, '2021-12-29', '2022-1-18', 'C128', NULL, NULL),
(71, 8.4, '2021-12-30', '2022-1-13', NULL, 2018, NULL);

```

```

INSERT INTO columnaQuirurgica(codigoProducto,
voltiosMinimos, voltiosMaximos,
gradosMinimos, gradosMaximos, alturaMetros)
VALUES ('C127', 20, 60, -5, 50, 2.2),
('C128', 22, 75, -10, 58, 1.2);

```

```

INSERT INTO lampara(valorLuxMaximo, valorLuxMinimo, gradosMaximos,
gradosMinimos, numeroLeds, codigoProducto)
VALUES (1200.00, 400.00, 55, 5, 48, 'L089'),
(1600.00, 400.00, 60, 10, 14, 'L090');

```

```
INSERT INTO panelTecnico(valorLuxMaximo, valorLuxMinimo,
presionGasMaximoBar, presionGasMinimoBar, codigoProducto)
VALUES (7000, 1000, 10, 5, 'P100'),
(5000, 1000, 10, 5, 'P101');
```

```
INSERT INTO revisionauxiliar(codigoProducto, estado, FechaRevision,
fechaSiguieteRevision, motivo)
VALUES ('C127', 4 , '2021-10-9', '2021-10-29', 'Alto desgaste del
equipo debido al uso' ),
('L089', 3.5 , '2021-10-10', '2021-12-9', 'Alto desgaste del equipo
debido al uso' ),
('P100', 4.2 , '2021-9-28', '2021-12-27', 'Alto desgaste del equipo
debido al uso' );
```

3.Consultas a tablas:

```
-- CONSULTAS
```

```
-- Informacion sobre los equipamientos electromedicos, actualmente en
uso.
```

```
-- dos formas de hacerlo, una a traves de la fecha de adquisicion de
la tabla equipamiento y otra a traves del oid de las tablas lampara,
panel...
```

```
CREATE OR REPLACE VIEW equipoactual2 AS
SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'L%' AND
fechaAdquisicion>= (SELECT MAX(ee.fechaAdquisicion) FROM
equipamientoelectromedico ee WHERE ee.codigoProducto LIKE 'L%' )
UNION
SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'P%' AND
fechaAdquisicion>= (SELECT MAX(ee.fechaAdquisicion) FROM
equipamientoelectromedico ee WHERE ee.codigoProducto LIKE 'P%' )
UNION
SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'C%' AND
fechaAdquisicion>= (SELECT MAX(ee.fechaAdquisicion) FROM
equipamientoelectromedico ee WHERE ee.codigoProducto LIKE 'C%' );
```

```
CREATE OR REPLACE VIEW equipoactual AS
SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico e NATURAL JOIN lampara l WHERE
l.oid_l1>=(SELECT MAX(l2.oid_l1) FROM lampara l2)
```

```

UNION
SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico e NATURAL JOIN columnaquirurgica c
WHERE c.oid_c>=(SELECT MAX(c2.oid_c) FROM columnaquirurgica c2)
UNION
SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico e NATURAL JOIN paneltecnico p WHERE
p.oid_p>=(SELECT MAX(p2.oid_p) FROM paneltecnico p2);

-- Vista donde se muestren ordenados adecuadamente las revisiones por
equipo y por fecha. (quitando las revisiones de clima y elec)

CREATE or replace VIEW revisionesOrdenadas AS
SELECT * FROM revision WHERE codigoProducto <> 'NULL' ORDER BY
codigoProducto, fechaRevision DESC;

-- Vista que constiene la proxima revision de los equipos actuales del
quirofano

CREATE OR REPLACE VIEW proxRevEquipActuales as
SELECT r.codigoProducto,MAX(r.fechaProximaRevision) 'proximaRevision'
FROM equipoactual ea, revision r
WHERE ea.codigoProducto = r.codigoProducto GROUP BY ea.codigoProducto;

-- Número de lamparas en la BD, precio máximo y precio mínimo y precio
medio

SELECT COUNT(*) 'N° de equipos', MAX(precio) 'Precio máximo',
MIN(precio) 'Precio minimo', AVG(precio) 'Precio medio'
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'L%';

-- Número de Columnas quirúrgicas en la BD, precio máximo y precio
mínimo y precio medio

SELECT COUNT(*) 'N° de equipos', MAX(precio) 'Precio máximo',
MIN(precio) 'Precio minimo', AVG(precio) 'Precio medio'
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'C%';

-- Número de Paneles técnicos en la BD, precio máximo y precio mínimo
y precio medio

SELECT COUNT(*) 'N° de equipos', MAX(precio) 'Precio máximo',
MIN(precio) 'Precio minimo', AVG(precio) 'Precio medio'
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'P%';

-- El nombre, el correo y numero de telefono y la empresa a la que
pertenecen(datos de contacto) de los ingenieros que han
-- trabajado en el mantenimiento de los equipamientos electromedicos.

```



```

SELECT e.codigoProducto, e.fechaAdquisicion, i.nombre,
i.correoElectronico, i.telefono, em.nombreEmpresa
FROM equipamientoelectromedico e, ingenierobiomedico i, empresa em
WHERE i.dni=e.dni AND i.cif=em.cif;

```

-- Las empresas que proveen actualmente la bd

```

SELECT unique(e.nombreEmpresa) FROM equipoactual ea,
ingenierobiomedico i, empresa e
WHERE ea.dni= i.dni AND i.cif= e.cif;

```

-- El precio total actual del quirofano.

```

SELECT SUM(precio) FROM equipamientoelectromedico where precio IN (
SELECT precio
FROM equipamientoelectromedico e NATURAL JOIN lampara l WHERE
l.oid_l>=(SELECT MAX(l2.oid_l) FROM lampara l2)
UNION
SELECT precio
FROM equipamientoelectromedico e NATURAL JOIN columnaquirurgica c
WHERE c.oid_c>=(SELECT MAX(c2.oid_c) FROM columnaquirurgica c2)
UNION
SELECT precio
FROM equipamientoelectromedico e NATURAL JOIN paneltecnico p WHERE
p.oid_p>=(SELECT MAX(p2.oid_p) FROM paneltecnico p2)
);

```

-- La media de precio de los diferentes equipamientos electromedicos, así como su máximo precio y su min.

```

SELECT nombreEquipo, AVG(precio)'Precio medio', Min(precio)'Precio
mínimo', MAX(precio)'Precio máximo'
FROM equipamientoelectromedico
GROUP BY nombreEquipo;

```

-- La cantidad de dinero que se ha gastado en cada empresa y el numero de equipos que se le han comprado,
-- así como el precio medio por equipo (ordenandolo de mas dinero a menos)

```

SELECT em.nombreEmpresa, SUM(precio)'Dinero total gastado',
COUNT(*)'Nº de equipos', SUM(precio)/COUNT(*) 'precioMedioEquipo'
FROM equipamientoelectromedico e, ingenierobiomedico i, empresa em
WHERE e.dni= i.dni AND i.cif = em.cif
GROUP BY em.nombreEmpresa
ORDER BY precio desc;

```

-- Peso actual del quirofano

```

SELECT SUM(e.pesoEnKg) FROM equipoactual ea NATURAL JOIN
equipamientoelectromedico e;

```

```

-- Número de revisiones por equipo

SELECT codigoProducto, COUNT(*) 'numeroRevisiones'
from revision where codigoProducto != 'NULL' GROUP BY codigoProducto
;

-- Equipamiento electromedico que se haya comprado en los últimos 3
años

SELECT codigoProducto, nombreEquipo, fechaAdquisicion,
fechaFinVidaUtil, precio FROM equipamientoelectromedico E
WHERE YEAR(E.fechaAdquisicion) BETWEEN YEAR(NOW())-3 AND YEAR(NOW());

-- Nombre, codigo del producto, fecha adquisicion y estado del último
que haya obtenido menos de un 7 en la ultima revisión

SELECT nombreEquipo, codigoProducto, fechaAdquisicion, estado
FROM equipamientoelectromedico NATURAL JOIN revision
WHERE fechaRevision>=(SELECT MAX(fechaRevision) FROM revision WHERE
estado<=7);

-- Datos del equipamiento electromedico perteneciente a una empresa y
el ingeniero al cargo(En este caso Carbueros Medica)

SELECT em.nombreEmpresa, ing.nombre, ing.apellidos,
eq.codigoProducto, ing.dni, ing.telefono, ing.correoElectronico
FROM equipamientoelectromedico eq, ingenierobiomedico ing ,
empresa em
WHERE eq.dni= ing.dni AND ing.cif= em.cif AND
em.nombreEmpresa='Carbueros Medica';

-- (VISTA para la siguiente consulta) Información sobre los ingenieros
biomédicos, y el número de revisiones que han hecho
-- aprovechamos el fallo de mariaDB con el nombre. (aquí si tiene
sentido que se pueda)

CREATE OR REPLACE VIEW revisionesHechasIngenieros as
SELECT i.dni, i.nombre,i.apellidos, COUNT(*) 'Revisiones Hechas' FROM
revision r, equipamientoelectromedico e, ingenierobiomedico i
WHERE r.codigoProducto= e.codigoProducto AND e.dni= i.dni GROUP BY
i.dni;

-- El ingeniero que haya hecho mas revisiones

SELECT * FROM revisioneshechasingenieros rhi
WHERE rhi.`Revisiones Hechas`>= (SELECT MAX(rhii.`Revisiones Hechas`)
FROM revisioneshechasingenieros rhii) ;

```

```

-- Equipamiento al que se le haya hecho mas revisiones

create or replace VIEW countRevisiones as SELECT r.codigoProducto,
COUNT(*) 'nRevisiones' FROM revision r
WHERE codigoProducto <> 'NULL'
GROUP BY r.codigoProducto;

SELECT codigoProducto, max(nRevisiones) FROM countRevisiones;

-- Dias de utilidad del equipamiento actual del quirofano

SELECT codigoProducto, DATEDIFF(fechaFinVidaUtil, fechaAdquisicion)
FROM equipoactual NATURAL JOIN equipamientoelectromedico;

-- Días que llevan activos los equipos desde su compra

SELECT codigoProducto , DATEDIFF(NOW(), fechaAdquisicion) FROM
equipamientoelectromedico;

```

4.Funciones y procedimientos:

```

-- PROCEDIMIENTOS

-- PROCEDIMIENTO PARA INSERTAR REVISIONES: y que se calculen
automaticamente la fecha de la proxima revision,
-- en función del equipo al que se le lleve a cabo la revisión, y su
periodicidad de revision.

DELIMITER //
CREATE OR REPLACE PROCEDURE

insertaRevision (
estado DECIMAL(3,1),
fechaRevision DATE,
codigoProducto VARCHAR(20) ,
fechaNormativaElectricidad INT,
fechaNormativaClimatizacion INT)

BEGIN

if (codigoProducto IS NOT NULL) then
    SET fechaNormativaElectricidad= NULL;
    SET fechaNormativaClimatizacion= NULL;
END if;

INSERT INTO revision (estado,
fechaRevision,
fechaProximaRevision,
codigoProducto ,
fechaNormativaElectricidad,
fechaNormativaClimatizacion)

```

```

VALUES (estado, fechaRevision,
DATE_ADD(fechaRevision, INTERVAL (SELECT e.periodicidadDias from
equipamientoelectromedico e
WHERE codigoProducto= e.codigoProducto) DAY),
codigoProducto , fechaNormativaElectricidad,
fechaNormativaClimatizacion); #aqui se ponen los parametros

END //
DELIMITER ;

-- PROCEDIMIENTO PARA INSERTAR EQUIPAMIENTO ELECTROMEDICO

DELIMITER //
CREATE OR REPLACE PROCEDURE
insertaEquipamientoElectromedico (codigoProducto VARCHAR(4),
pesoEnKg INT,
nombreEquipo VARCHAR(20),
fechaFinVidaUtil DATE,
periodicidadDias INT,
precio DECIMAL(8,2),
fechaAdquisicion DATE,
dni VARCHAR(9))
BEGIN
INSERT INTO equipamientoelectromedico (codigoProducto,
pesoEnKg,
nombreEquipo,
fechaFinVidaUtil,
periodicidadDias,
precio,
fechaAdquisicion,
dni)
VALUES (codigoProducto, pesoEnKg, nombreEquipo, fechaFinVidaUtil,
periodicidadDias, precio, fechaAdquisicion,dni); #aqui se ponen los
parametros
END //
DELIMITER ;

-- PROCEDIMIENTO PARA INSERTAR INGENIERO BIOMEDICO

DELIMITER //
CREATE OR REPLACE PROCEDURE
insertaIngenieroBiomedico (dni VARCHAR(9),
numeroSeguridadSocial VARCHAR(15),
nombre VARCHAR(50),
apellidos VARCHAR(50),
titulacion VARCHAR(30),
fechaNacimiento DATE,
codigoPostal VARCHAR(5),
telefono VARCHAR(14),
correoElectronico VARCHAR(60),
cif VARCHAR(9))
BEGIN

```

```

INSERT INTO ingenierobiomedico (dni,
numeroSeguridadSocial,
nombre,
apellidos,
titulacion,
fechaNacimiento,
codigoPostal,
telefono,
correoElectronico,
cif)
VALUES (dni, numeroSeguridadSocial, nombre, apellidos, titulacion,
fechaNacimiento, codigoPostal, telefono, correoElectronico,
cif);#aqui se ponen los parametros
END //
DELIMITER ;

```

-- PROCEDIMIENTO PARA INSERTAR EMPRESA

```

DELIMITER //
CREATE OR REPLACE PROCEDURE
insertaEmpresa (cif VARCHAR(10),
nombreEmpresa VARCHAR(20),
direccionWeb VARCHAR(50),
pais VARCHAR(20),
direccion VARCHAR(100),
telefono VARCHAR(14))
BEGIN
INSERT INTO empresa (cif,
nombreEmpresa,
direccionWeb,
pais,
direccion,
telefono)
VALUES (cif, nombreEmpresa, direccionWeb, pais, direccion,
telefono);#aqui se ponen los parametros
END //
DELIMITER ;

```

-- PROCEDIMIENTO PARA INSERTAR REVISION AUXILIAR

```

DELIMITER //
CREATE OR REPLACE PROCEDURE
insertaRevisionAuxiliar (
codigoProducto VARCHAR(20),
estado VARCHAR(50),
fechaRevision DATE,
motivo VARCHAR(200)
)

BEGIN
INSERT INTO revisionauxiliar (
codigoProducto,
estado,

```

```

fechaRevision,
fechaSiguienteRevision,
motivo)

```

```

VALUES (codigoProducto, estado, fechaRevision,
DATE_ADD(fechaRevision, INTERVAL (SELECT e.periodicidadDias from
equipamientoelectromedico e
WHERE codigoProducto= e.codigoProducto) DAY),
motivo);
END //
DELIMITER ;

```

```

-- PROCEDIMIENTO QUE REBAJE UN 60% A LOS EQUIPAMIENTOS ELECTROMEDICOS
QUE LE QUEDE
-- un año o menos de un año para acabar su vida util
-- PARA SU POSTERIOR VENTA a instituciones académicas

```

```

DELIMITER //
CREATE OR REPLACE PROCEDURE
rebajaPrecioFinVida()
BEGIN
UPDATE equipamientoelectromedico
SET precio = precio*0.4
WHERE DATE_ADD(NOW(), INTERVAL 365 DAY) >= fechaFinVidaUtil ;

END //
DELIMITER ;

```

```

-- PROCEDIMIENTO QUE atrasa la fecha de la proxima revision en 5 dias
(solo del equipamiento actual)

```

```

DELIMITER //
CREATE OR REPLACE PROCEDURE
aumentarFechaProximaRevision()
BEGIN
UPDATE revision r, proxrevequipactuales a1
SET r.fechaProximaRevision= DATE_ADD(r.fechaProximaRevision,
INTERVAL 5 DAY)
WHERE a1.proximaRevision= r.fechaProximaRevision AND
a1.codigoProducto = r.codigoProducto;

END //
DELIMITER;

```

```

-- PROCEDIMEINTO QUE CAMBIE los parámetros minimos de la última
normativaClimatizacion en un 15%

```

```

DELIMITER //
CREATE OR REPLACE PROCEDURE
    cambioPorcentajeNormativaClima()
BEGIN
    UPDATE normativaclimatizacion n
    SET n.caudalAireMinimo= n.caudalAireMinimo*1.15
    WHERE n.fechaNormativaClimatizacion>= (SELECT
MAX(fechaNormativaClimatizacion) FROM normativaclimatizacion);
    UPDATE normativaclimatizacion n
    SET n.gradosMinimos= n.gradosMinimos*1.15
    WHERE n.fechaNormativaClimatizacion>= (SELECT
MAX(fechaNormativaClimatizacion) FROM normativaclimatizacion);
    UPDATE normativaclimatizacion n
    SET n.porcentajeHumedadMinimo= n.porcentajeHumedadMinimo*1.15
    WHERE n.fechaNormativaClimatizacion>= (SELECT
MAX(fechaNormativaClimatizacion) FROM normativaclimatizacion);
    END //
DELIMITER;

-- PROCEDIMEINTO QUE acota el intervalo de valor de luz de laas
normativas de electricidad

DELIMITER //
CREATE OR REPLACE PROCEDURE
    cambioIntervaloNormativaElectricidad()
BEGIN
    UPDATE normativaelectricidad n
    SET n.nivelLuxMinima= n.nivelLuxMinima*1.05
    WHERE n.fechaNormativaElectricidad>= (SELECT
MAX(ne.fechaNormativaElectricidad) FROM normativaelectricidad ne);

    UPDATE normativaelectricidad n
    SET n.nivelLuxMaxima= n.nivelLuxMaxima*0.95
    WHERE n.fechaNormativaElectricidad>= (SELECT
MAX(ne.fechaNormativaElectricidad) FROM normativaelectricidad ne);

    END //
DELIMITER;

-- Procedimiento que aumente la periodicidad (disminuye los dias entre
revisiones) de revisiones de las normativas de clima y electricidad en
un 10%

DELIMITER //
CREATE OR REPLACE PROCEDURE
    cambioPeriodicidadNormativas()
BEGIN
    UPDATE normativaclimatizacion n
    SET n.periodicidadRevisionDias= n.periodicidadRevisionDias*0.9
    WHERE n.fechaNormativaClimatizacion>= (SELECT
MAX(fechaNormativaClimatizacion) FROM normativaclimatizacion);

```

```

        UPDATE normativaelectricidad ne
        SET ne.periodicidadRevisionDias= ne.periodicidadRevisionDias*0.9
        WHERE ne.fechaNormativaElectricidad>= (SELECT
MAX(ne.fechaNormativaElectricidad) FROM normativaelectricidad ne);

    END //
DELIMITER;

#####

-- FUNCIONESS

#Devuelveme la nota de la peor revision de un determinado equipamiento
electromedico

DELIMITER //
CREATE OR REPLACE FUNCTION
    estadoPeorEquipo(codigoP VARCHAR(20)) RETURNS DECIMAL(3,1)
BEGIN

DECLARE res DECIMAL (3,1) DEFAULT (SELECT MIN(r.estado) FROM revision
r where r.codigoProducto= codigoP);
DECLARE res2 DECIMAL (3,1) DEFAULT (select MIN(ra.estado) from
revisionauxiliar ra where ra.codigoProducto=codigoP);

if (res > res2)
then
RETURN res2 ;
END if;

RETURN res;

END //
DELIMITER ;

#Cantidad de revisiones de un ingeniero por debajo del 5(el estado)

DELIMITER //
CREATE OR REPLACE FUNCTION
    numRevisionesMenor5(DNI VARCHAR(9)) RETURNS INT
BEGIN
DECLARE res INT;
DECLARE res2 int;
SET res= (SELECT COUNT(*) FROM revision r NATURAL JOIN
equipamientoelectromedico e WHERE e.dni= DNI AND r.estado<6 GROUP BY
e.dni);

```



```

SET res2= (SELECT COUNT(*) FROM revisionauxiliar ra,
equipamientoelectromedico e WHERE ra.codigoProducto= e.codigoProducto
and e.dni= DNI AND ra.estado<6 GROUP BY e.dni);
RETURN res + res2;
END //
DELIMITER ;

#QUE ME DIGA SI LAS MEDICIONES QUE ESTOY LLEVANDO A CABO, ESTÁN
CORRECTAMENTE (Lamparas).

DELIMITER //

CREATE OR REPLACE FUNCTION
    evaluacionLampara(codigoP VARCHAR(20), valorLuz INT,
valorTemperatura INT) RETURNS VARCHAR (1000)
BEGIN
DECLARE res VARCHAR(1000) DEFAULT 'El resultado de la medición es: ';
DECLARE p1 VARCHAR(50) DEFAULT 'Falla la medición de la luz';
DECLARE p2 VARCHAR(50) DEFAULT 'Falla la medición de la temperatura';
DECLARE p3 VARCHAR(50) DEFAULT 'La medición es correcta';
DECLARE c1 BOOLEAN DEFAULT true ;
DECLARE c2 BOOLEAN DEFAULT true;

if (valorLuz < (SELECT valorLuxMinimo from equipamientoelectromedico
NATURAL JOIN lampara WHERE codigoP= codigoProducto) OR
valorLuz > (SELECT valorLuxMaximo from equipamientoelectromedico
NATURAL JOIN lampara WHERE codigoP= codigoProducto ))
then
SET res= CONCAT(res, p1, ';') ;
SET c1= false;
END if;

IF ( valorTemperatura < (SELECT gradosMinimos from
equipamientoelectromedico NATURAL JOIN lampara WHERE codigoP=
codigoProducto) OR
valorTemperatura > (SELECT gradosMaximos from
equipamientoelectromedico NATURAL JOIN lampara WHERE codigoP=
codigoProducto ) )
then
SET res= CONCAT(res, p2, ';') ;
SET c2= false;
END if;

if(c1 and c2)
then
set res= CONCAT(res, p3, ';') ;
END if;

RETURN res;
END //
DELIMITER ;

```

#QUE ME DIGA SI LAS MEDICIONES QUE ESTOY LLEVANDO A CABO, ESTÁN CORRECTAMENTE (Columnas Quirúrgicas).

DELIMITER //

CREATE OR REPLACE FUNCTION

```
    evaluacionColumna(codigoP VARCHAR(20), valorVoltaje INT,
valorTemperatura INT) RETURNS VARCHAR (1000)
BEGIN
DECLARE res VARCHAR(1000) DEFAULT 'El resultado de la medición es: ';
DECLARE p1 VARCHAR(50) DEFAULT 'Falla la medición del voltaje';
DECLARE p2 VARCHAR(50) DEFAULT 'Falla la medición de la temperatura';
DECLARE p3 VARCHAR(50) DEFAULT 'La medición es correcta';
DECLARE c1 BOOLEAN DEFAULT true ;
DECLARE c2 BOOLEAN DEFAULT true;
```

```
if (valorVoltaje < (SELECT voltiosMinimos from
equipamientoelectromedico NATURAL JOIN columnaquirurgica WHERE
codigoP= codigoProducto) OR
valorVoltaje > (SELECT voltiosMaximos from equipamientoelectromedico
NATURAL JOIN columnaquirurgica WHERE codigoP= codigoProducto ))
then
SET res= CONCAT(res, p1, ';') ;
SET c1= false;
END if;
```

```
IF ( valorTemperatura < (SELECT gradosMinimos from
equipamientoelectromedico NATURAL JOIN columnaquirurgica WHERE
codigoP= codigoProducto) OR
valorTemperatura > (SELECT gradosMaximos from
equipamientoelectromedico NATURAL JOIN columnaquirurgica WHERE
codigoP= codigoProducto ) )
then
SET res= CONCAT(res, p2, ';') ;
SET c2= false;
END if;
```

```
if(c1 and c2)
then
set res= CONCAT(res, p3, ';') ;
END if;
```

```
RETURN res;
END //
DELIMITER ;
```

#QUE ME DIGA SI LAS MEDICIONES QUE ESTOY LLEVANDO A CABO, ESTÁN CORRECTAMENTE (Paneles).

DELIMITER //

CREATE OR REPLACE FUNCTION

```

        evaluacionPanel(codigoP VARCHAR(20), valorLuz INT, valorPresion
INT) RETURNS VARCHAR (1000)
BEGIN
DECLARE res VARCHAR(1000) DEFAULT 'El resultado de la medición es: ';
DECLARE p1 VARCHAR(50) DEFAULT 'Falla la medición de la luz';
DECLARE p2 VARCHAR(50) DEFAULT 'Falla la medición de la presión del
gas';
DECLARE p3 VARCHAR(50) DEFAULT 'La medición es correcta';
DECLARE c1 BOOLEAN DEFAULT true ;
DECLARE c2 BOOLEAN DEFAULT true;

if (valorLuz < (SELECT valorLuxMinimo from equipamientoelectromedico
NATURAL JOIN paneltecnico WHERE codigoP= codigoProducto) OR
valorluz > (SELECT valorLuxMaximo from equipamientoelectromedico
NATURAL JOIN paneltecnico WHERE codigoP= codigoProducto ))
then
SET res= CONCAT(res, p1, ';') ;
SET c1= false;
END if;

IF ( valorPresion < (SELECT presionGasMinimoBar from
equipamientoelectromedico NATURAL JOIN paneltecnico WHERE codigoP=
codigoProducto) OR
valorPresion > (SELECT presionGasMaximoBar from
equipamientoelectromedico NATURAL JOIN paneltecnico WHERE codigoP=
codigoProducto ) )
then
SET res= CONCAT(res, p2, ';') ;
SET c2= false;
END if;

if(c1 and c2)
then
set res= CONCAT(res, p3, ';') ;
END if;

RETURN res;
END //
DELIMITER ;

```

#QUE ME DIGA SI LAS MEDICIONES QUE ESTOY LLEVANDO A CABO, ESTÁN CORRECTAMENTE (Segun la normativa actual de la Electricidad).

DELIMITER //

CREATE OR REPLACE FUNCTION

```

        evaluacionElectricidad(valorLuz INT) RETURNS VARCHAR (1000)
BEGIN
DECLARE res VARCHAR(1000) DEFAULT 'El resultado de la medición es: ';
DECLARE p1 VARCHAR(50) DEFAULT 'Falla la medición de la luz';
DECLARE p2 VARCHAR(50) DEFAULT 'La medición es correcta';
DECLARE c1 BOOLEAN DEFAULT true ;

```

```

if (valorLuz < (SELECT nivelLuxMinima
from normativaelectricidad n
WHERE n.fechaNormativaElectricidad=(SELECT
MAX(ne.fechaNormativaElectricidad) FROM normativaelectricidad ne)) OR
valorLuz > (SELECT nivelLuxMaxima
from normativaelectricidad
WHERE fechaNormativaElectricidad=(SELECT
MAX(ne.fechaNormativaElectricidad) FROM normativaelectricidad ne)))
then
SET res= CONCAT(res, p1, ';') ;
SET c1= false;
END if;

if(c1)
then
set res= CONCAT(res, p2, ';') ;
END if;

RETURN res;
END //
DELIMITER ;

```

#QUE ME DIGA SI LAS MEDICIONES QUE ESTOY LLEVANDO A CABO, ESTÁN CORRECTAMENTE (Segun la normativa actual de la Climatizacion).

```
DELIMITER //
```

```
CREATE OR REPLACE FUNCTION
```

```
    evaluacionClimatizacion(valorCaudalAire INT, valorGrados INT,
valorHumedad INT) RETURNS VARCHAR (1000)
```

```
BEGIN
```

```
DECLARE res VARCHAR(1000) DEFAULT 'El resultado de la medición es: ';
```

```
DECLARE p1 VARCHAR(50) DEFAULT 'Falla la medición del caudal del
aire';
```

```
DECLARE p2 VARCHAR(50) DEFAULT 'Falla la medición de la temperatura';
```

```
DECLARE p3 VARCHAR(50) DEFAULT 'Falla la medición del porcentaje
humedad';
```

```
DECLARE p4 VARCHAR(50) DEFAULT 'La medición es correcta';
```

```
DECLARE c1 BOOLEAN DEFAULT true;
```

```
DECLARE c2 BOOLEAN DEFAULT TRUE;
```

```
DECLARE c3 BOOLEAN DEFAULT true;
```

```
if (valorCaudalAire < (SELECT caudalAireMinimo from
normativaclimatizacion
```

```
WHERE fechaNormativaClimatizacion=(SELECT
```

```
MAX(nc.fechaNormativaClimatizacion) FROM normativaclimatizacion nc))
```

```
OR
```

```
valorCaudalAire > (SELECT caudalAireMaximo from normativaclimatizacion
```

```
WHERE fechaNormativaClimatizacion=(SELECT
```

```
MAX(nc.fechaNormativaClimatizacion) FROM normativaclimatizacion nc)))
```

```
then
```

```

SET res= CONCAT(res, p1, ';');
SET c1= false;
END if;

IF ( valorGrados < (SELECT gradosMinimos from normativaclimatizacion
WHERE fechaNormativaClimatizacion=(SELECT
MAX(nc.fechaNormativaClimatizacion) FROM normativaclimatizacion nc))
OR
valorGrados > (SELECT gradosMaximos from normativaclimatizacion
WHERE fechaNormativaClimatizacion=(SELECT
MAX(nc.fechaNormativaClimatizacion) FROM normativaclimatizacion nc)))
then
SET res= CONCAT(res, p2, ';') ;
SET c2= false;
END if;

if (valorHumedad < (SELECT porcentajeHumedadMinimo from
normativaclimatizacion
WHERE fechaNormativaClimatizacion=(SELECT
MAX(nc.fechaNormativaClimatizacion) FROM normativaclimatizacion nc))
OR
valorHumedad > (SELECT porcentajeHumedadMaximo from
normativaclimatizacion
WHERE fechaNormativaClimatizacion=(SELECT
MAX(nc.fechaNormativaClimatizacion) FROM normativaclimatizacion nc)))
then
SET res= CONCAT(res, p3, ';') ;
SET c3= false;
END if;

if(c1 and c2 AND c3)
then
set res= CONCAT(res, p4, ';') ;
END if;

RETURN res;
END //
DELIMITER ;

```

-- Función que dado el código de un producto te devuelva la fecha de la próxima revisión

```

DELIMITER //
CREATE OR REPLACE FUNCTION revProx (cProducto VARCHAR(4))
RETURNS DATE
BEGIN

RETURN (SELECT max(fechaProximaRevision)
FROM revision r
WHERE r.codigoProducto= cProducto);
END //
DELIMITER ;

```

```

-- Función que dado el nombre de una empresa devuelva el dinero
invertido

DELIMITER //
CREATE OR REPLACE FUNCTION DineroInvEmpr (nEmp VARCHAR(20))
RETURNS DECIMAL(10,2)
BEGIN
RETURN (SELECT SUM(precio)
        FROM equipamientoelectromedico eq, ingenierobiomedico ing
, empresa em
        WHERE eq.dni= ing.dni AND ing.cif= em.cif AND nEmp=
em.nombreEmpresa);
END //
DELIMITER ;

-- Funcion que dado el dni de un ingeniero devuelva el nombre de la
empresa a la que pertenece

DELIMITER //
CREATE OR REPLACE FUNCTION EmpresaALaQuePertenece (DNIIngeniero
VARCHAR(9))
RETURNS VARCHAR(20)
BEGIN
RETURN (SELECT nombreEmpresa
        FROM ingenierobiomedico i, empresa e
        WHERE i.cif= e.cif AND DNIIngeniero=i.dni);
END //
DELIMITER ;

-- Nos devuelve dada una fecha el coste del quirófano en esa fecha

DELIMITER //
CREATE OR REPLACE FUNCTION PrecioEnFecha (fecha date)
RETURNS DECIMAL(12,2)
BEGIN
RETURN (SELECT SUM(precio) FROM equipamientoelectromedico WHERE
codigoProducto in
(SELECT codigoProducto FROM equipamientoelectromedico WHERE
codigoProducto LIKE 'L%' AND
fechaAdquisicion = (SELECT MAX(ee.fechaAdquisicion) FROM
equipamientoelectromedico ee
WHERE ee.codigoProducto LIKE 'L%' AND ee.fechaAdquisicion<=fecha)
UNION
SELECT codigoProducto FROM equipamientoelectromedico WHERE
codigoProducto LIKE 'P%' AND
fechaAdquisicion= (SELECT MAX(ee.fechaAdquisicion) FROM
equipamientoelectromedico ee
WHERE ee.codigoProducto LIKE 'P%' AND ee.fechaAdquisicion<=fecha)
UNION
SELECT codigoProducto FROM equipamientoelectromedico WHERE
codigoProducto LIKE 'C%' AND

```

```

fechaAdquisicion= (SELECT MAX(ee.fechaAdquisicion) FROM
equipamientoelectromedico ee
WHERE ee.codigoProducto LIKE 'C%' AND ee.fechaAdquisicion<=fecha));

END //
DELIMITER ;

-- Nos devuelve la diferencia de coste entre el quirófano actualmente
y el coste del quirófano en esa fecha

DELIMITER //
CREATE OR REPLACE FUNCTION DiferenciaCoste (fecha date)
RETURNS DECIMAL(12,2)
BEGIN
RETURN (SELECT SUM(precio) FROM equipoactual2) -
PrecioEnFecha(fecha);

END //
DELIMITER ;

-- Calcula el año de la normativa más exigente

DELIMITER //
CREATE OR REPLACE FUNCTION NormativaMasExigente ()
RETURNS int
BEGIN
RETURN
(SELECT nn.fechaNormativaClimatizacion FROM normativaclimatizacion nn
WHERE ((nn.caudalAireMaximo- nn.caudalAireMinimo)
+( nn.gradosMaximos - nn.gradosMinimos ) +( nn.porcentajeHumedadMaximo
- nn.porcentajeHumedadMinimo)) =
(SELECT MIN((n.caudalAireMaximo- n.caudalAireMinimo)
+( n.gradosMaximos - n.gradosMinimos ) +( n.porcentajeHumedadMaximo -
n.porcentajeHumedadMinimo))
FROM normativaclimatizacion n));

END //
DELIMITER ;

/*
-- Función que dado una fecha me diga el equipo que había en el
quiroyfano en esa fecha.

DELIMITER //
CREATE OR REPLACE FUNCTION EquipoDelMomento (fecha DATE)
RETURNS TABLE AS
RETURN (
(SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'L%' AND
fechaAdquisicion = (SELECT MAX(ee.fechaAdquisicion) FROM
equipamientoelectromedico ee
WHERE ee.codigoProducto LIKE 'L%' AND ee.fechaAdquisicion<=fecha)

```

```

UNION
SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'P%' AND
fechaAdquisicion= (SELECT MAX(ee.fechaAdquisicion) FROM
equipamientoelectromedico ee
WHERE ee.codigoProducto LIKE 'P%' AND ee.fechaAdquisicion<=fecha )
UNION
SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'C%' AND
fechaAdquisicion= (SELECT MAX(ee.fechaAdquisicion) FROM
equipamientoelectromedico ee
WHERE ee.codigoProducto LIKE 'C%' AND ee.fechaAdquisicion<=fecha))

);

DELIMITER ;

*/

```

5.Triggers asociados a reglas de negocios:

```

-- TRIGGERS

#RN-001: La fecha de revision de un equipo no puede cambiar en mas de
5 dias
#con respecto a la fecha en la que se preveia revisarlo segun su
periodicidad

DELIMITER //
CREATE OR REPLACE TRIGGER margenesFechaRevision
before INSERT ON revision
FOR EACH ROW
BEGIN
SET @error_message =
'Se ha excedido en más de 5 días la fecha en la que se preveía revisar
';

if (datediff(NEW.fechaRevision, (SELECT MAX(r.fechaProximaRevision)
FROM revision r
WHERE NEW.codigoProducto= r.codigoProducto)) > 5 OR datediff((SELECT
MAX(r.fechaProximaRevision) FROM revision r
WHERE NEW.codigoProducto= r.codigoProducto), NEW.fechaRevision ) > 5)

then

INSERT INTO revisionauxiliar (codigoProducto, estado, FechaRevision,
fechaSiguienteRevision, motivo) VALUES
(new.codigoProducto,new.estado,new.fechaRevision,
new.fechaProximaRevision, @error_message);

```



```

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @error_message;
END if;

END//

DELIMITER ;

#RN-002. El sistema debe garantizar que los equipos sean repuestos
antes de superar su vida útil.
-- para ello, avisa al hacer una revision del equipo en cuestion 3
meses antes de insertar una revision de estas.

DELIMITER //
CREATE OR REPLACE TRIGGER triggerFinVidaÚtil
before INSERT ON revision
FOR EACH ROW
BEGIN
SET @error_message =
'Queda menos de tres meses para la fecha asociada a la vida útil del
equipo';

if (DATE_ADD(NOW(), INTERVAL (90) DAY) >=
(SELECT UNIQUE(e.fechaFinVidaUtil) from revision r NATURAL JOIN
equipamientoelectromedico e
WHERE NEW.codigoProducto = e.codigoProducto)) then

INSERT INTO revisionauxiliar (codigoProducto, estado, FechaRevision,
fechaSiguieteRevision, motivo) VALUES
(new.codigoProducto,new.estado,new.fechaRevision,
new.fechaProximaRevision, @error_message);

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @error_message;

END if;

END//

DELIMITER ;

#RN-004 El codigo de producto, la primera sigla tienen que coincidir
con la primera letra del atributo nombre de ese equipamiento
electromedico
DELIMITER //
CREATE OR REPLACE TRIGGER letraCodigoProducto
before INSERT ON equipamientoelectromedico
FOR EACH ROW
BEGIN
SET @error_message =
'La primera letra del código del producto tiene que coincidir con la
sigla de su nombre';

```

```

IF ( SUBSTRING(NEW.codigoProducto, 1, 1) <> SUBSTRING(NEW.nombreEquipo
,1,1) )
THEN
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @error_message;
END IF;
END//
DELIMITER ;

```

*#RN-005 Si en una revision la nota cambia 3 puntos frente a la revision anterior del mismo equipo, salta un aviso.
#Esa revision se guarda en la nueva tabla Revision Auxiliar*

```

DELIMITER //
CREATE OR REPLACE TRIGGER revisisionMuyDefectuosa3
before insert ON revision
FOR EACH ROW
BEGIN
SET @error_message =
'La revision baja tres puntos frente a la revision anterior';

if ( (SELECT estado FROM revision r WHERE NEW.codigoProducto=
r.codigoProducto AND r.fechaRevision >=
(SELECT MAX(r2.fechaRevision) FROM revision r2 WHERE
NEW.codigoProducto= r2.codigoProducto)) - NEW.estado >=3)
then

INSERT INTO revisionauxiliar (codigoProducto, estado, FechaRevision,
fechaSiguieteRevision, motivo) VALUES
(new.codigoProducto,new.estado,new.fechaRevision,
new.fechaProximaRevision, @error_message);

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @error_message ;
END if;
END//
DELIMITER ;

```

#RN-006 Cambiar el equipamiento despues de que baje en una revision la calificacion del 3

```

DELIMITER //
CREATE OR REPLACE TRIGGER revisisionMuyDefectuosa1
before insert ON revision
FOR EACH ROW
BEGIN
SET @error_message =
'La revision es muy defectuosa, baja del 3';

if (new.estado<=3 )
then

```

```

INSERT INTO revisionauxiliar (codigoProducto, estado, FechaRevision,
fechaSiguienteRevision, motivo) VALUES
(new.codigoProducto,new.estado,new.fechaRevision,
new.fechaProximaRevision, @error_message);
END if;

if (new.estado<=3 )
then
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @error_message;

END if;

END//
DELIMITER ;

```

*#RN-007 que lleve 3 revisiones seguidas anteriores por debajo del 5.5.
(la que meto seria la cuarta)*

```

DELIMITER //
CREATE OR REPLACE FUNCTION cursorAuxiliarTriger(NEWcodigoProducto
VARCHAR (4))
RETURNS Boolean
BEGIN
DECLARE contBoolean BOOLEAN DEFAULT true;
DECLARE contInt INT default 0;
DECLARE done BOOLEAN DEFAULT false;
DECLARE revisiones ROW TYPE OF revision;

DECLARE cursorTriger CURSOR FOR
SELECT * FROM revision r
WHERE NEWcodigoProducto= r.codigoProducto ORDER BY fechaRevision DESC;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done := TRUE;
OPEN cursorTriger;
readLoop:LOOP
FETCH cursorTriger INTO revisiones;
    IF done OR contInt=3 THEN # para el bucle cuando o bien se
acabe los datos de farmacos o bien la cuenta llegue a 3
        LEAVE readLoop;
    END IF;
    if (revisiones.estado > 5.5) then
SET contBoolean = FALSE;
END if;
SET contInt = contInt + 1;
END LOOP;
CLOSE cursorTriger;

if((SELECT COUNT(*) FROM revision r WHERE NEWcodigoProducto =
r.codigoProducto) < 3)
then

```

```

set contBoolean= FALSE;
END if;

RETURN contBoolean;
END //
DELIMITER ;

DELIMITER //
CREATE OR REPLACE TRIGGER revisisionMuyDefectuosa2
before insert ON revision
FOR EACH ROW
BEGIN
SET @error_message =
'Este equipo lleva teniendo 3 revisiones seguidas una nota nota
inferior a 5.5 ';

if (NEW.estado<5.5 and cursorAuxiliarTriger(NEW.codigoProducto))
then

INSERT INTO revisionauxiliar (codigoProducto, estado, FechaRevision,
fechaSiguienteRevision, motivo) VALUES
(new.codigoProducto,new.estado,new.fechaRevision,
new.fechaProximaRevision, @error_message);

#SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @error_message;
END if;
END//
DELIMITER ;

#RN-008. Todo el equipamiento electro médico que este en el quirofono
(al mismo tiempo) no podrá ser proveído por la misma empresa.

-- Creemos que debería de funcionar.
/*
DELIMITER //
CREATE OR REPLACE TRIGGER monopolioEmpresa
before INSERT ON equipamientoelectromedico
FOR EACH ROW
BEGIN
SET @error_message =
'NO puede haber 3 equipos de la misma empresa a la vez';

if ((SELECT e.nombreEmpresa FROM ingenierobiomedico i , empresa e
WHERE e.cif= i.cif and NEW.dni= i.dni)
= (SELECT UNIQUE(e.nombreEmpresa) FROM equipoactual2 ea,
ingenierobiomedico i, empresa e
WHERE ea.dni = i.dni AND e.cif= i.cif
and SUBSTRING(ea.codigoProducto, 1, 1) <>
SUBSTRING(NEW.codigoProducto, 1, 1)) )

```

```

then

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @error_message;
END if;

END//

DELIMITER;
*/

#RN-008.Todo el equipamiento electro médico que este en el quirofono
(al mismo tiempo) no podrá ser proveído por la misma empresa.
#LA CLAVE ESTA EN EL AFTER

DELIMITER //
CREATE OR REPLACE TRIGGER monopolioEmpresa2
after INSERT ON equipamientoelectromedico
FOR EACH ROW
BEGIN
SET @error_message =
'NO puede haber 3 equipos de la misma empresa a la vez';

if (auxMonopolio()) then
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @error_message;
END if;

END//

DELIMITER ;

-- funcion auxiliar para monopolio2

DELIMITER //
CREATE OR REPLACE FUNCTION auxMonopolio() RETURNS boolean
BEGIN
DECLARE res BOOLEAN DEFAULT FALSE;

DECLARE v1 VARCHAR(50) DEFAULT (SELECT e.nombreEmpresa FROM
equipoactual2 ea, ingenierobiomedico i, empresa e
WHERE ea.dni= i.dni AND i.cif= e.cif AND ea.codigoProducto LIKE 'L%');

DECLARE v2 VARCHAR(50) DEFAULT (SELECT e.nombreEmpresa FROM
equipoactual2 ea, ingenierobiomedico i, empresa e
WHERE ea.dni= i.dni AND i.cif= e.cif AND ea.codigoProducto LIKE 'C%');

DECLARE v3 VARCHAR(50) DEFAULT (SELECT e.nombreEmpresa FROM
equipoactual2 ea, ingenierobiomedico i, empresa e
WHERE ea.dni= i.dni AND i.cif= e.cif AND ea.codigoProducto LIKE 'P%');

if (v1=v2 AND v1=v3 AND v2=v3) then

```

```
set res= TRUE;
END if;
```

```
RETURN res;
END //
DELIMITER ;
```

#RN-009: El equipo debe tener al menos tres años de utilidad, desde que ssu compra

```
DELIMITER //
CREATE OR REPLACE TRIGGER añosUtilidadMinima
before INSERT ON equipamientoelectromedico
FOR EACH ROW
BEGIN
SET @error_message =
'Tiene que tener al menos 3 años de utilidad desde su compra';

if (DATEDIFF(NEW.fechaFinVidaUtil, NEW.fechaAdquisicion) < 365*3)
then

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @error_message;
END if;

END//
```

```
DELIMITER ;
```

RN-010: NO se puede registrar revisiones de equipos que no están actualmete en el quirófano

```
DELIMITER //
CREATE OR REPLACE TRIGGER revisionesConSentido2
before INSERT ON revision
FOR EACH ROW
BEGIN
SET @error_message =
'NO se puede registrar revisiones de equipos que no están actualmete
en el quirófano';

if NOT( NEW.codigoProducto IN(SELECT ea.codigoProducto FROM
equipoactual2 ea ))
then

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @error_message;
END if;

END//
```

```
DELIMITER ;
```

```
/*
DELIMITER //
CREATE OR REPLACE TRIGGER revisionesConSentido
before INSERT ON revision
FOR EACH ROW
BEGIN
SET @error_message =
'NO se puede registrar revisiones de equipos que no están actualmete
en el quirófano 1';

if ( NEW.codigoProducto <> (SELECT ea.codigoProducto FROM
equipoactual2 ea
                                WHERE
SUBSTRING(NEW.codigoProducto,1, 1) = SUBSTRING(ea.codigoProducto ,1,
1) ))
then

SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = @error_message;
END if;

END//

DELIMITER ;
*/
```

6.Ensayos y pruebas:

```
# INDICACIONES: está preparado para que todo esté perfecto, si se
ejecuta todas las lineas de codigo de arriba a abajo.
#                 si no es posible que se des controle. :)
```

```
#PRUEBAS PROCEDIMIENTOS
```

```
-- Procedimientos para insertar
```

```
CALL `insertaRevision`('7', '2022-02-7', 'L090', '67', '80');
```

```
CALL `insertaEquipamientoElectromedico`('C129', '123', 'Columna
Quirúrgica',
'2028-12-23', '20', '1200', '2017-01-12', '60245603V');
```

```
CALL `insertaRevisionAuxiliar`('C129', '2', '2018-10-10', 'Problemas
con la toma de alimentacion');
```

```
CALL `insertaIngenieroBiomedico`('12345678A', '1223344455', 'Carlos',
'Ramirez Manzano', 'Ingenieria de la Salud',
'1998-02-21', '11100', '644890789', 'carlitosramirez98@gmail.com',
'A08993735');
```

```
CALL `insertaEmpresa`('A34568975', 'novaclinic', 'novaclinicweb',
'España', 'Carpestro,14', '601234556');
```

Procedimiento para rebajar un 60% el precio de los equipos que le quedan un año para caducar.

```
CALL `insertaEquipamientoElectromedico`('L800', '123', 'Lámpara',
'2023-1-23', '20', '10000', '2019-01-12', '60245603V');
```

```
SELECT e.codigoProducto, e.fechaFinVidaUtil, e.precio FROM
equipamientoelectromedico e
WHERE DATE_ADD(NOW(), INTERVAL 365 DAY) >= e.fechaFinVidaUtil;
```

```
CALL rebajaPrecioFinVida();
```

Procedimiento para atrasar la fecha proxima revision en 5 dias del equipo actual

```
SELECT * FROM proxrevequipactuales;
```

```
CALL `aumentarFechaProximaRevision`();
```

PROCEDIMEINTO QUE CAMBIE los parámetros minimos de la última normativaClimatizacion en un 15%

```
SELECT * from normativaclimatizacion n
where n.fechaNormativaClimatizacion>= (SELECT
MAX(fechaNormativaClimatizacion) FROM normativaclimatizacion);
```

```
CALL `cambioPorcentajeNormativaClima`();
```

-- PROCEDIMEINTO QUE acota el intervalo de valor de luz de laas normativas de electricidad

```
SELECT * from normativaelectricidad n
WHERE n.fechaNormativaElectricidad>= (SELECT
MAX(fechaNormativaElectricidad) FROM normativaelectricidad);
```

```
CALL `cambioIntervaloNormativaElectricidad`();
```

Procedimiento que cambia de periodicidad de clima y electricidad

```
SELECT n.fechaNormativaElectricidad 'fechaNormativa',
n.periodicidadRevisionDias from normativaelectricidad n
```



```

WHERE n.fechaNormativaElectricidad>= (SELECT
MAX(fechaNormativaElectricidad) FROM normativaelectricidad)
UNION
SELECT nc.fechaNormativaClimatizacion 'fechaNormativa',
nc.periodicidadRevisionDias from normativaclimatizacion nc
where nc.fechaNormativaClimatizacion>= (SELECT
MAX(fechaNormativaClimatizacion) FROM normativaclimatizacion);

```

```
CALL `cambioPeriodicidadNormativas`();
```

#PRUEBAS FUNCIONES

#Devuelveme la nota de la peor revision de un determinado equipamiento electromedico

```

SELECT MIN(r.estado) FROM revision r where r.codigoProducto= 'C127';
select MIN(ra.estado) from revisionauxiliar ra where
ra.codigoProducto='C127';

```

```
SELECT `estadoPeorEquipo`('C127');
```

#Cantidad de revisiones de un ingeniero por debajo del 5

```

SELECT e.dni,COUNT(*) FROM revision r NATURAL JOIN
equipamientoelectromedico e WHERE e.dni='32536784Z' AND r.estado<6
GROUP BY e.dni;
SELECT e.dni, COUNT(*) FROM revisionauxiliar ra,
equipamientoelectromedico e WHERE ra.codigoProducto= e.codigoProducto
and e.dni= '32536784Z' AND ra.estado<6 GROUP BY e.dni;

```

```
SELECT `numRevisionesMenor5`('32536784Z');
```

#QUE ME DIGA SI LAS MEDICIONES QUE ESTOY LLEVANDO A CABO, ESTÁN CORRECTAMENTE (Lamparas).

```
SELECT * FROM lampara WHERE codigoProducto='L089';
```

```

SELECT `evaluacionLampara`('L089', '800', '50');
SELECT `evaluacionLampara`('L089', '3000', '50');
SELECT `evaluacionLampara`('L089', '800', '1');

```

#QUE ME DIGA SI LAS MEDICIONES QUE ESTOY LLEVANDO A CABO, ESTÁN CORRECTAMENTE (Columnas Quirúrgicas).

```
SELECT * FROM columnaquirurgica WHERE codigoProducto='C127';
```

```

SELECT `evaluacionColumna`('C127', '20', '10');
SELECT `evaluacionColumna`('C127', '1', '10');
SELECT `evaluacionColumna`('C127', '1', '-20');

```

#QUE ME DIGA SI LAS MEDICIONES QUE ESTOY LLEVANDO A CABO, ESTÁN CORRECTAMENTE (Paneles).

```
SELECT * FROM paneltecnico WHERE codigoProducto='P101';
```

```
SELECT `evaluacionPanel`('P101', '3000', '6');
```

```
SELECT `evaluacionPanel`('P101', '120', '6');
```

```
SELECT `evaluacionPanel`('P101', '2000', '1');
```

#QUE ME DIGA SI LAS MEDICIONES QUE ESTOY LLEVANDO A CABO, ESTÁN CORRECTAMENTE (Segun la normativa actual de la Electricidad).

```
SELECT * FROM normativaelectricidad ne WHERE  
ne.fechaNormativaElectricidad=(SELECT  
MAX(ne.fechaNormativaElectricidad) FROM normativaelectricidad ne);
```

```
SELECT `evaluacionElectricidad`('1');
```

```
SELECT `evaluacionElectricidad`('2000');
```

#QUE ME DIGA SI LAS MEDICIONES QUE ESTOY LLEVANDO A CABO, ESTÁN CORRECTAMENTE (Segun la normativa actual de la Climatizacion).

```
SELECT * FROM normativaclimatizacion nc WHERE  
nc.fechaNormativaClimatizacion=(SELECT  
MAX(nc.fechaNormativaClimatizacion) FROM normativaclimatizacion nc);
```

```
SELECT `evaluacionClimatizacion`('1', '1', '1');
```

```
SELECT `evaluacionClimatizacion`('3000', '20', '50');
```

Función que dado el código de un producto te devuelva la fecha de la próxima revisión

```
SELECT max(fechaProximaRevision)  
FROM revision r  
WHERE r.codigoProducto= 'C128';
```

```
SELECT `revProx`('C128');
```

```
SELECT `revProx`('C127');
```

Función que dado el nombre de una empresa devuelva el dinero invertido

```
SELECT SUM(precio)  
FROM equipamientoelectromedico eq, ingenierobiomedico ing  
, empresa em  
WHERE eq.dni= ing.dni AND ing.cif= em.cif AND 'Carbueros  
Medica'= em.nombreEmpresa;
```

```
SELECT `DineroInvEmpr`('Carbueros Medica');
```

```
SELECT `DineroInvEmpr`('NovaClinic');
```

Funcion que dado el dni de un ingeniero devuelva el nombre de la empresa a la que pertenece

```
SELECT nombreEmpresa  
FROM ingenierobiomedico i, empresa e  
WHERE i.cif= e.cif AND '32536784Z'=i.dni;
```

```
SELECT `EmpresaALaQuePertenece`('32536784Z');
```

#Nos devuelve dada una fecha el coste del quirófano en esa fecha

```
SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'L%' AND
fechaAdquisicion = (SELECT MAX(ee.fechaAdquisicion) FROM
equipamientoelectromedico ee
WHERE ee.codigoProducto LIKE 'L%' AND ee.fechaAdquisicion<='2021-1-
1')
UNION
SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'P%' AND
fechaAdquisicion= (SELECT MAX(ee.fechaAdquisicion) FROM
equipamientoelectromedico ee
WHERE ee.codigoProducto LIKE 'P%' AND ee.fechaAdquisicion<='2021-1-1'
)
UNION
SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'C%' AND
fechaAdquisicion= (SELECT MAX(ee.fechaAdquisicion) FROM
equipamientoelectromedico ee
WHERE ee.codigoProducto LIKE 'C%' AND ee.fechaAdquisicion<='2021-1-
1');
```

```
SELECT `PrecioEnFecha`('2021-1-1');
```

#Nos devuelve la diferencia de coste entre el quirófano actualmente y el coste del quirófano en esa fecha

```
SELECT `PrecioEnFecha`(NOW());
SELECT `PrecioEnFecha`('2021-1-1');
```

```
SELECT `DiferenciaCoste`('2021-1-1');
```

#Calcula el año de la normativa más exigente

```
SELECT `NormativaMasExigente`();
```

#PROBAR LOS TRIGGERS

*#RN-001: La fecha de revision de un equipo no puede cambiar en mas de 5 dias
#con respecto a la fecha en la que se preveia revisarlo segun su periodicidad*

```
SELECT `revProx`('C128');
```

```
CALL `insertaRevision`('9.1', '2022-1-29', 'C128', '8', '9');
```

#RN-002. El sistema debe garantizar que los equipos sean repuestos antes de superar su vida útil.
#para ello, avisa al hacer una revision del equipo en cuestion 3 meses antes de insertar una revision de estas.

```
CALL `insertaRevision`('6.7', '2022-2-28', 'P101', '8', '9');
```

#RN-003. Check sobre el estado revision

```
CALL `insertaRevision`('14', '2022-1-25', 'C128', '8', '9');
```

#RN-004 El codigo de producto, la primera sigla tienen que coincidir con la primera letra del atributo nombre de ese equipamiento electromédico

```
CALL insertaEquipamientoElectromedico('C456', '100', 'Lampara', '2021-4-2', '78', '34000.67', '2022-1-1', '89898989H');
```

#RN-005 Si en una revision la nota cambia 3 puntos frente a la revision anterior del mismo equipo, salta un aviso.
#Esa revision se guarda en la nueva tabla Revision Auxiliar

```
CALL `insertaRevision`('6', '2022-1-18', 'C128', '8', '9');
```

#RN-006 Cambiar el equipamiento después de que baje en una revision la calificación del 3 (el estado)

```
CALL `insertaRevision`('7', '2022-1-18', 'C128', '8', '9');  
CALL `insertaRevision`('4.3', '2022-2-8', 'C128', '8', '9');  
CALL `insertaRevision`('2.3', '2022-2-28', 'C128', '8', '9'); #debe saltar este :)
```

#RN-007 que lleve 3 revisiones seguidas anteriores por debajo del 5.5. (la que meto sería la cuarta)
NO DEBE SALTAR MENSAJE, SINO QUE DEBE DE INSERTARSE EN LA TABLA REVISIONAUXILIAR

```
CALL `insertaRevision`('5.4', '2022-2-28', 'C128', '8', '9');  
CALL `insertaRevision`('5.3', '2022-3-20', 'C128', '8', '9');  
CALL `insertaRevision`('5.2', '2022-4-10', 'C128', '8', '9'); # debe de saltar este :)
```

#RN-008. Todo el equipamiento electro médico que este en el quirófano (al mismo tiempo) no podrá ser proveído por la misma empresa.

```
SELECT e.nombreEmpresa, ea.codigoProducto, i.dni, e.cif FROM  
equipoactual2 ea, ingenierobiomedico i, empresa e  
WHERE ea.dni = i.dni AND e.cif= i.cif;
```

```
SELECT e.nombreEmpresa FROM ingenierobiomedico i , empresa e WHERE  
e.cif= i.cif and '45678314C'= i.dni;
```

```
SELECT UNIQUE(e.nombreEmpresa) FROM equipoactual2 ea,  
ingenierobiomedico i, empresa e  
WHERE ea.dni = i.dni AND e.cif= i.cif  
and SUBSTRING(ea.codigoProducto, 1, 1) <> SUBSTRING('C202', 1, 1);
```

```
CALL `insertaEquipamientoElectromedico`('C202', '123', 'Columna', #  
Esto es lo que salta :)  
'2028-12-23', '20', '1200', '2023-02-20', '45678314C');
```

*#RN-009: El equipo debe tener al menos tres años de utilidad, desde
que ssu compra*

```
CALL `insertaEquipamientoElectromedico`('C289', '1000', 'Columna',  
'2023-1-1', '56', '34000.89', '2022-1-1',  
'99999999F');
```

*# RN-010: NO se puede registrar revisiones de equipos que no están
actualmete en el quirófano*

```
CALL `insertaRevision`('8.2', '2021-10-9', 'C127', '8', '8');
```

*# RN-011: Los valores maximos tienen que ser mayores que los valores
minimos de un mismo intervalo.
(en las tablas de las normativas y en las tablas de los equipos;
lampara...)*

```
INSERT INTO normativaElectricidad(nivelLuxMaxima, nivelLuxMinima,  
fechaNormativaElectricidad,  
periodicidadRevisionDias)  
VALUES (270, 1000, 2016, 14);
```

```
INSERT INTO normativaClimatizacion(caudalAireMinimo, caudalAireMaximo,  
gradosMaximos, gradosMinimos, porcentajeHumedadMinimo,  
PorcentajeHumedadMaximo, fechaNormativaClimatizacion,  
periodicidadRevisionDias)  
VALUES (2700, 350, 2, 19, 48, 5, 2012, 20);
```

```
INSERT INTO columnaQuirurgica(codigoProducto,voltiosMinimos,  
voltiosMaximos,gradosMinimos, gradosMaximos,alturaMetros)  
VALUES ('C127',100, 60, 100, 50, 2.2);
```

```
INSERT INTO lampara(valorLuxMaximo, valorLuxMinimo, gradosMaximos,
gradosMinimos, numeroLeds, codigoProducto)
VALUES (100.00, 400.00, 55, 95, 48, 'L089');
```

```
INSERT INTO panelTecnico(valorLuxMaximo, valorLuxMinimo,
presionGasMaximoBar, presionGasMinimoBar, codigoProducto)
VALUES (700, 1000, 1, 5, 'P100');
```

#RN-012: La fecha proxima de revisión tiene que ser mayor o igual a la fecha de revision.(tablas de revision y revisionAux)

```
INSERT INTO revisionauxiliar(codigoProducto, estado, FechaRevision,
fechaSiguienteRevision, motivo)
VALUES ('C127', 4 , '2023-10-9', '2021-10-29', 'Hola ');
```

```
INSERT INTO revision(oid_r, estado, fechaRevision,
fechaProximaRevision, codigoProducto,
fechaNormativaElectricidad, fechaNormativaClimatizacion)
VALUES (1, 9.6, '2023-1-1', '2021-1-14', NULL, 2018, NULL);
```

```
#SELECT row_number() OVER (ORDER BY r.fechaRevision),
r.codigoProducto FROM revision r ;
```

```
/*
SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'L%' AND
fechaAdquisicion = (SELECT MAX(ee.fechaAdquisicion) FROM
equipamientoelectromedico ee
WHERE ee.codigoProducto LIKE 'L%' AND ee.fechaAdquisicion<='2021-1-
1')
UNION
SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'P%' AND
fechaAdquisicion= (SELECT MAX(ee.fechaAdquisicion) FROM
equipamientoelectromedico ee
WHERE ee.codigoProducto LIKE 'P%' AND ee.fechaAdquisicion<='2021-1-1'
)
UNION
SELECT codigoProducto, nombreEquipo, precio, fechaAdquisicion, dni
FROM equipamientoelectromedico WHERE codigoProducto LIKE 'C%' AND
```

```
fechaAdquisicion= (SELECT MAX(ee.fechaAdquisicion) FROM  
equipamientoelectromedico ee  
WHERE ee.codigoProducto LIKE 'C%' AND ee.fechaAdquisicion<='2021-1-  
1');  
*/
```

Gracias por haber llegado hasta aquí 😊