Celia Martín Cubillo
Pablo Ligero Isla
Alejandro Pastrana San Martín
Rosario Ferrer Pérez-Blanco

MtM Data Base of Surgical Instruments

1) Why have we chosen this topic and what is it for?
While considering what to make our Data Base about, we all agreed in some things, like the fact that we wanted it to be related with our studies in biomedical engineering as well as relevant in some field of this discipline so that it could be actually applied in the hypothetical case in which it would be a real and practical Data Base.

This is the reason why we got to the thought of doing our project about the surgical instruments that are provided to hospitals. This is a very important issue, as long as not only the building of these instruments must be properly measured, but also the delivery and use in every hospital.

We specially centred in the construction of the instruments from the moment the hospital makes an order, following with the company that sells the materials, the machinery used and the employees working with them, and how these instruments are built.

Therefore, this Data Base would mean a very accurate and easy way of controlling each factor involved in the manufacturing of the instruments and the orders made by hospitals, so that when something fails or must be revised, there can be a quick access to any of the variants involved.

Instrument:
This is the **main entity** of our diagram, in this entity we can find all the **properties** that are relevant regarding the manufacturing, storing and selling of surgical instruments.

The **relation** it has with the **Warehouse** entity is also **Store**, because these instruments together with the materials will be stored in the same warehouse once they have been built. This relation is then **n** (Instruments) to **1** (Warehouse).

The **properties** of the entity Instrument are:
-ID Instrument → Is the identifier for each instrument so that they are registered properly. (Primary Key)
-Name→Is the name of the instrument.
- Amount→ Number of this type of instruments available at the moment.
- Model → Which is type of instrument.
- Purpose→ Aim for which the instrument is made.
-Number of uses → Number of times the instrument can be used.
-Body Location→ Refers to the part of the body on where the instrument will be used.
-Price→ Sell price of the instrument.

Company:
This **entity** refers to the different companies we have relationships with to supply us the materials we need for the factory process.
The **relation** of the company with materials is to **supply them.**Each factory can provide us different materials. This relation is **n** (materials) **to 1**(company).
The **properties** for this entity are:
- Company ID → Refers to the ID of each company (Primary Key).
-Company name→Refers to the name of the company.
- Location→ Refers to the location of the company. Closer companies are better for us.

Warehouse:

The **entity** Warehouse refers to the storage of the materials and instruments that have been manufactured. This is the same Warehouse for everything, so it is important to control its capacity and how empty or full it is.

This is why it has the following **properties**:
-Warehouse ID→Is the ID of the warehouse(Primary Key).
- Warehouse location →Where the warehouse is placed inside a city.
- Capacity → How much space it has to store all the material and instruments.
- Filled space → How much of this space is actually filled with the stored materials and instruments, so that we can know if we can continue storing or not.

Material:
The **entity** Material refers to the different kind o material that will be used in order to manufacture each instrument.

The **relation** of the materials with **Warehouse** is **Store**, as long as they are all stored in the same warehouse. This relation is then **n** (Materials) to **1** (Warehouse).

The **properties** for this entity are:
- Material ID → Is the identifier for each material so that they are registered properly. (Primary Key)
- Weight → The weight of each material differs from one to another. (e.g.: plastic and metal have different weights)
- Volume → The volume of the materials is a property that will affect to the capacity of the warehouse where they will be stored.
-Type→It refers to type of material the machinery is going to use.

Hospital:
This **entity** refers to the Hospital that is going to make the perform the main order depending of the necessities that the hospital have of instruments.

The **relation**of the order with **Hospital** is to **Perform.** Between the entities Hospital and Order the relationis n – n, because many hospital can perform the same order and one hospital can perform many orders.

The **Properties** for this entity are:
- Hospital ID → The ID of the hospital make reference to a specific hospital (Primary Key).
- Name →The name of the Hospital that performs the order.
- Location → The main location of the hospital in the city.
- Medical Specialization → The specialization in terms of medicine branch that the hospital have.

Order:

The **entity** Order make reference to the order that the hospital performs to our company in order to supply them with different type of surgical instruments.

The **relation**of order with **Instrument** is **Contain.** Between the entities Order and Instrument the relation is an n – n, because one Order can contain many instruments, and also many instruments can be contained by one order. As this las relation is an n – n relation, we have a different table to relate this two entities in order to relate all the possible combinations, being the union between the two foreign keys of the new table (primary keys of their table) the primary key of this new table, called Instrument-Order.

The **properties** for this entity are:
- Order ID→The order of the ID used to make reference to each one of the orders (Primary Key).
- Delivery Date → Date in which the order should be delivered to the hospital.
- Total Amount of Instruments →The total amount of instruments of a specific instrument.
- Order Date →This property make reference to the date when the order was performed.

Employee:

The **entity**Employee refers to the person is going to work in our company.

The **relation** of the employee with **Machinery** is **Use**. We decide to use an n – 1 relation which means that many employees can use one type of machine

The**properties** for this entity are:
- Employee ID → The Id of the employee (Primary Key).
- Name →Name of the employee
- Type of Contract → Specifies if the employee has a permanent or a temporal contract.
- Specialization Type →Specifies the type of machine that the employee is prepared to use.

Machinery:

This **entity** refers to the different machine we have in the factory to build the surgical instruments.

The **relation** of the machinery with **Materials** is to **use them**, as each machinery may need different materials for the process. This relation is **n** (materials) **to 1**(machinery).

The **relation** of the machinery with **Instrument** is to **manufacture them.** Each machinery can build different instruments, as each instrument can be built in different machineries. This relation is **n** (materials) **to n**(machinery).

The **properties** for this entity are:
-Machinery ID→It refers to the ID of the machinery (Primary Key).
- Machinery Type → Refers to the different type so we can registered properly.
- Size→ Size of the machinery is important for the disposition in the factory.
- Date of Installation→Refersto the moment of installation of the machinery, so we can maintain them properly and substitute them if needed.
- State → Used to know if the machine is working at the moment or not.

Use of the classes:

Package Interface:

- **DBInterface:** we use this class as Interface for our JPA and JDBC Manager classes. To make a suitable use of the methods and for making sure we implement them correctly.
- **UserInterface:** this is the class we use to interact with the user by the console. So, it may have all the facilities for the user, and access to all of our methods to work properly with the data base.
- **Validator:** we use this class as an auxiliar of the User Interface. Will help us in the cases the user introduce a value into the console with which we cannot work with. This will provide us security for possibles errors in the use of the database.

Package JDBC:

- **JDBCManager:** this is the main class to work with JDBC. Instead of working with connexions in each SQL method, we use a single connexion from this class, while we control the use of the other classes from the package from here. Connexions methods of opening are closing will be performed.
- **JDBCCreate:** class used to create all the SQL tables from the database.
- **JDBCDelete:** class used to delete a specific value from a specific table.
- **JDBCInsert:** class used to insert a specific value from a specific table.
- **JDBCSearch:** class used to search into the database for a specific value from a specific table.
- **JDBCSelect:** class used to select a specific value from a specific table.
- **JDBCUpdate:** class used to update a specific attribute from a specific value inside a specific table.

Package JPA:

- **JPAManager:** in JPA case, we will use a single class, inside which we perform the different SQL methods, such as insert, create, delete or update of several pojos. Also this class will have methods to open and close connexion to the database.

Package XMLs:

- **Java2Xml:** used to make the marshalling.
- **Xml2Java:** used to make the unmarshalling.
- **SQLDateAdapter:** class used to transform both from String to Date format or Date to String.
- **XMLManager:** class used to control the xml performing from the UserInterface.