

Pharmacy Platform

Mariana Margalho Alves Calado, up202003072@fe.up.pt

Susana Catarina Moreira Teixeira, up202103376@fe.up.pt

**Course:** Master’s degree in Biomedical Engineering

**Curricular Unit:** Information Systems

1. **Description**

The pharmacy’s website is an information system that should allow the registration of customers, order products and allows to fill prescriptions fields from doctors that specify a medicine. It should also allow to find out more information about each product such as brand, category, price, etc.

- Each **customer** has a personal account to which is associated his name, phone number, e-mail, address, city, password, VAT number, role and an identification number that is assigned to him after his registration;

- For each **product** an identification number, name, price and description must be stored;

- There are several types of products: medicine, hygiene and beauty products, dermatological products, sports products, food supplements, contraception products, medical equipment, orthopaedic products, first aid and pet products. For each **category** an identification number and name must be stored. A product belongs to a single category but there may be a category with no associated products or have several associated products;

- Information will also be stored about each product **brand**, which will have an identification number and a name. Each brand can have several products. However, each product can only belong to a single brand;

- Information about **prescriptions** (by a doctor) will include the name of the beneficiary and the name of the doctor who wrote the prescription;

- Each **employee** has a personal account to which their name, password, phone number, email, address, role and an identification number is associated. The name of the employee will also be associated to an order. For each order there is only one employee and each of them can be related to several orders;

- For each **supplier**, a unique code, company name, address, e-mail and a phone number should be stored. A supplier sells several products of several categories, so there is a minimum quantity of products; (acrescentar a cena dos employ pediram produtos aos suppliers)

- Each **order** has an identification number, a delivery address, status and the date it was executed. A customer may have several orders, but an order belongs to a single customer. An order can contain several products and one product can belong to several orders;

- A prescription can be associated with a certain order to be able to receive the medicine that need a prescription. An order can contain several products with several prescriptions. Each prescription belongs to one order. Each medicine can have more than one prescription, but each prescription can contain only contain one medicine.

1. Diagram

   Description automatically generated**UML Diagram**
2. **Relational Model**

[Disjoint generalization → Object-oriented approach]

**Employee** (id\_employee, name, phone\_num, e-mail, address, password, role)

{NOT NULL (name, phone\_num, e-mail, password, role)}

{UNIQUE (phone\_num, e-mail)}

**Customer** (id\_customer, name, phone\_num, e-mail, address, city, password, VAT\_num, role)

{NOT NULL (name, phone\_num, e-mail, address, city, password, VAT\_num, role)}

{UNIQUE (VAT\_num, phone\_num, e-mail)}

**Suplier** (id\_suplier, name, phone\_num, e-mail, address)

{NOT NULL (name, phone\_num, e-mail)}

{UNIQUE (phone\_num, e-mail)}

**Product** (id\_product, name, unit\_price, description, category\_id→ Category, brand\_id→ Brand)

{NOT NULL (name, unit\_price, description, category\_id, brand\_id)}

{CHECK (unit\_price > 0)}

**SupProduct** (supplier\_id→ Suplier, product\_id→ Product, quantity)

{NOT NULL (quantity)}

{CHECK (quantity > 0)}

**Orders** (id\_order, date, delv\_address, status, customer\_id→ Customer, employee\_id→ Employee)

{NOT NULL (date, customer\_id, delv\_address, employee\_id, status)}

**OrdersProduct** (order\_id→ Orders, product\_id→ Product, quantity)

{NOT NULL (quantity)}

{CHECK (quantity > 0)}

**Category** (id\_category, name)

{NOT NULL (name)}

**Brand** (id\_brand, name)

{NOT NULL (name)}

**Prescription** (id\_prescription, doct\_name, benf\_name, product\_id→ Product, order\_id → Orders)

{NOT NULL (doct\_name, benf\_name, product\_id, order\_id)}

DML/DQL:

- roles: “emp” e ”cust”

- fazer todos os inserts e selects testar querys – exemplo quais os clientes que não pagam

- Pode se copiar algum código de css