Software Requirements Specification

Project: Management and Generation of Production Order

Made by:

Díaz Cervantes Amieva Alejandro

Martínez Pérez Diana

Medina Becerra Alan Oswaldo

Ramirez Navarro Marcos David

10 - 31st - 2024

TABLE OF CONTENTS

DOCUMENTATION FORM

Date	Revision	Author	Verified Quality dep.
09 - 25 - 2024		Díaz Cervantes Amieva Alejandro Martínez Pérez Diana Medina Becerra Alan Oswaldo Ramirez Navarro Marcos David	

Document validated by the parties on date:

By the Client	By the Supplying Company
Dra. Marylu Lara Lagunes	
Sgd. Mr./ Mrs.	**Sgd. Mr./ Mrs.**

1. INTRODUCTION

Over the years, as technology has evolved, it has become the best tool for managing businesses of all kinds, from small shops to large industries. However, there is still a part of the sector that has not made the leap to updating these tools to fully improve productivity and data security.

This project will focus on a very specific area: the generation of production orders for pharmaceutical and dietary supplement laboratories. The creation and management of these orders requires an efficient, accurate and secure process that guarantees the satisfaction of customer requirements. For this reason, a web app, specifically designed for these companies, will be presented, providing solutions to optimize and improve the necessary processes.

1.1. PURPOSE

The main purpose is the development of a specialized software for the generation of production orders for pharmaceutical laboratories and dietary supplement companies. This software will ensure the efficiency, precision and security of the data in the process of creating the orders. As a result, customers will receive high quality and reliable work.

1.2. SCOPE

The 'OrdenesProd' project will be a web application for laboratories involved in the production of pharmaceuticals and food supplements. It will be an essential tool for a good management of the production orders.

1.3. PERSONAL INVOLUCRADO

Name	Martinez Perez Diana
Role	Collection of the necessary information for the creation and correct documentation of the project.
Professional Category	Student
Responsibilities	Integradora
Contact Information	0323105965@ut-tijuana.edu.mx
Approved	

Name	Ramirez Navarro Marcos David
Role	Performing Ubuntu Operating system installation and testing.
Professional Category	Student
Responsibilities	Operating Systems
Contact Information	0323105847@ut-tijuana.edu.mx
Approved	

Name	Diaz Cervantes Amieva Alejandro
Role	Creation of a database to store the information entered in the software and to be able to access it freely.
Professional Category	Student
Responsibilities	Data Base
Contact information	0323105898@ut-tijuana.edu.mx
Approved	

Name	Medina Becerra Alan Oswaldo
Role	Create a web page using the PHP language.
Professional category	Student
Responsibilities	Web Development
Contact information	0323106021@ut-tijuana.edu.mx
Approved	

1.4. DEFINITIONS, ACRONYMS AND ABBREVIATIONS

Some words used in this software media may be unfamiliar to most people, even if they are basic, here are some brief descriptions of the words and abbreviations that may be unfamiliar or unknown what they mean completely may be better in major measure to understand this document.

User: A user is a person who uses or interacts with an application, software, electronic device, or computer system to accomplish a specific goal.

Software: This is a term that refers to a program or a set of computer programs, as well as the data, procedures, and guidelines that allow various tasks to be performed in a computer system

Module: A module in the software is a component that performs a specific function and is composed of different computer programs, as well as it can be a class, package, library or even a complete application that intersects with others to form more robust and complete systems.

Database: A database is a software product designed to store large amounts of information in an organized and structured manner. It can be stored locally on your personal computer or on an external remote server.

Reliability: It's the reliability with which an interface can be used by a user, reliability implies that the software must be understandable, learnable, usable, and attractive, and contribute to the functionality and efficiency of the product.

Portability: It's the possibility to compile the source code of a program so that it can be run on different computer platforms. Is the measure of the ease with which an application can transfer a computing environment.

Optimization: Software optimization seeks to adapt computer systems to perform their tasks as efficiently as possible and use as few resources as possible.

Scalability: This refers to the ability of an application or system to handle an increased workload or demand efficiently, without compromising performance.

Security: It is essential to protect privacy and ensure a secure environment for users and for the program itself, protecting the integrity of data and ensuring the safety of users.

Performance: It is the measure of how efficiently the application uses the system resources to do Performance encompasses different aspects of how the program interacts with the underlying device.

1.5. REFERENCES

Reference	Title	Date	Author
https://weremote.net/terminos-basicos-programacion/	21 términos básicos de programación que debes dominar	16 de febrero del 2023	Nicholas Bonder
https://redwerk.es/blog/vocabulario-de-terminos-de-desarrollo-de-software-para-no-tecnicos-los-60-mas-importantes/	Vocabulario de términos de desarrollo de software para no técnicos: Los 60 más importantes	08 de agosto del 2023	REDWERK

1.6. SUMMARY

With the development of this project, a technological tool will be implemented to provide better information management to those companies that are in the business of manufacturing pharmaceutical products and dietary supplements, focusing specifically on the management and creation of production orders. The program will be designed to work in a Web environment, maintaining an easy access to the information of all products, customers, orders, etc. that belong to the corresponding company, facilitating the contact and visibility of the necessary information.

The structure of the program is divided into different modules that allow users to easily understand how the program works.

2. GENERAL DESCRIPTION

This document describes the development of a comprehensive Production Order Management software system. The system is designed to optimize the planning, coordination, and execution of orders with the goal of improving operational efficiency, billing accuracy, customer satisfaction, and overall control of operations.

The software integrates several critical functions, including order management, human resource management within the system, and order fulfillment. These functionalities allow for more efficient and accurate management of resources and processes involved in production. In addition, the system ensures the security of data storage, giving users the confidence they need to handle sensitive and critical information securely and reliably.

2.1. PRODUCT PERSPECTIVE

The "Production Orders" system is designed as a standalone software that is not limited by integration with other specific programs or applications. It also provides tools for business logic.

2.2. PRODUCT FUNCTIONALITY

The system is divided into two types of users, one will have full access to the system (Administrator) and the other one will have limited access.

Product: The user will be able to register, update, view and delete any product.

Customer: The user will be able to register, update, view and delete any customer.

Employee: The user will be able to register, update, view and delete any employee

User : The user will be able to register, update, view and disable any user, assign user roles and unlock users.

Order: The user will be able to register, update, view and delete any order.

Production Order: The user will be able to register, update, view and delete any Production Order.

Formula: The user will be able to register, update, view and delete new formulas

BASIC USER

Order: The user will just view the order created.

Production Order: The user register, update and view the production orders.

2.3. USER CHARACTERISTICS

User Type	Administrator
Education	Specialized in production orders.
Abilities	Basic program management
Activities	Register, administer and manage all information about customers, employees, orders and products

2.4. RESTRICTIONS

- The system must have a user friendly and intuitive interface.
- The system should have a data structure based on the standard MySQL model.
- The system must be available 365 days a year.
- The system must be able to handle an increase in the number of production orders without compromising its performance or functionality.
- The software should be compatible with the latest version of the client's operating system.

2.5. ASSUMPTIONS AND DEPENDENCE

- The devices on which the software is deployed must have a minimum of resources to function properly.
- Content management and information input should be the responsibility of the company.
- The person responsible for managing the programme should have a basic knowledge of similar software.

2.6. EXPECTED SYSTEM EVOLUTION

The web and technology trends are evolving rapidly. We want to spend time continuing to learn new tools, languages and techniques to help us keep the website up to date and competitive, including adapting our software to cell phones.

3. SPECIFIC REQUIREMENTS

Requirement number	FR 1
Name	DataBase
Type	☐ Requirement ☐ Constraint
Description	The system will store and obtain all the information recorded through this requirement, for example; employee information, customers, formulas, orders, production orders.
Priority	☐ High/Essential ☐ Medium/wished ☐ Low/Optional

Requirement number	FR 2
Name	Orders management
Type	☐ Requirement ☐ Constraint
Description	In this module you can modify or consult the data about the orders that customers have requested.
Priority	☐ High/Essential ☐ Medium/wished ☐ Low/Optional

Requirement number	FR 3
Name	Users management
Type	☐ Requirement ☐ Constraint
Description	In this module you can assign the access restriction, you can choose between a normal user and an administrator user.
Priority	☐ High/Essential ☐ Medium/wished ☐ Low/Optional

Requirement number	FR 4
Name	Products management
Type	☐ Requirement ☐ Constraint
Description	In this module you can modify or consult the products already done
Priority	☐ High/Essential ☐ Medium/wished ☐ Low/Optionall

Requirement number	FR 5
Name	Production orders management
Type	☐ Requirement ☐ Constraint

Description	in this module you can modify or consult the data about the production orders, example: the ingredients, the presentation.
Priority	☐ High/Essential ☐ Medium/wished ☐ Low/Optional

Requirement number	FR 6
Name	Employees management
Type	☐ Requirement ☐ Constraint
Description	In this module you can modify or consult the data about the employees
Priority	☐ High/Essential ☐ Medium/wished ☐ Low/Optional

Requirement number	FR 7
Name	Customers management
Type	☐ Requirement ☐ Constraint
Description	In this module you can modify or consult the data about the customers.
Priority	☐ High/Essential ☐ Medium/wished ☐ Low/Optional

3.1. COMMON INTERFACE REQUIREMENTS

3.1.1. USER INTERFACES

1. General Design

We seek to appeal to as many people as possible through a simple, beautiful and eye-pleasing interface.

2. Main Screen, Menu Options

Users will be able to select an option using the keyboard or by clicking on the corresponding button on the graphical interface. Each option will take the user to a new screen or section where they can perform the specific actions related to that category.

3.1.2. HARDWARE INTERFACES

It will be necessary to have computing devices that are in good condition and fully functional, it will be required for the execution of the program.

3.1.3. SOFTWARE INTERFACES

For our product to be fully functional the only other software product that must be installed is the linux operating system, ubuntu distribution.

3.1.4. COMMUNICATION INTERFACES

The server and clients will communicate with each other, using standard protocols on the Internet, whenever possible. For example, existing protocols (FTP or other convenient protocols) should be used to transfer files.

3.2. FUNCTIONAL REQUIREMENTS

DataBase

Helps to properly manage and store all information entered into the program.

Customers module

In this module new clients are registered, as well as the data of the already registered clients can be modified and consulted.

Employees module

In this module you can register employees to enter the system, as well you can register a user and a role for the employee, such as modify and consult the information already entered.

Production Orders module

In this module the orders are registered, as well as the data of the already registered orders can be modified and consulted.

Orders module

In this module the purchase orders are registered, as well the data of the already registered purchase orders can be modified and consulted.

Products module

Here you can register new products, as well the data of the already registered products can be modified and consulted. Also in this module you can register, modify or consult the data of raw materials, packaging material, product type, presentation type and the ingredients.

Formula module

In this module you can register new formulas, as well the data of the already registered formulas can be modified and consulted.

3.3. NON FUNCTIONAL REQUIREMENTS

Security

Total security is guaranteed in the use of the software, safeguarding all stored information and only managed by authorized people without the administration affecting the security of our system.

Reliability

User interfaces will be easy to access, streamlining their understanding, usefulness and functionality.

Availability

The system will be available 24 hours a day, every day of the week.

Maintainability

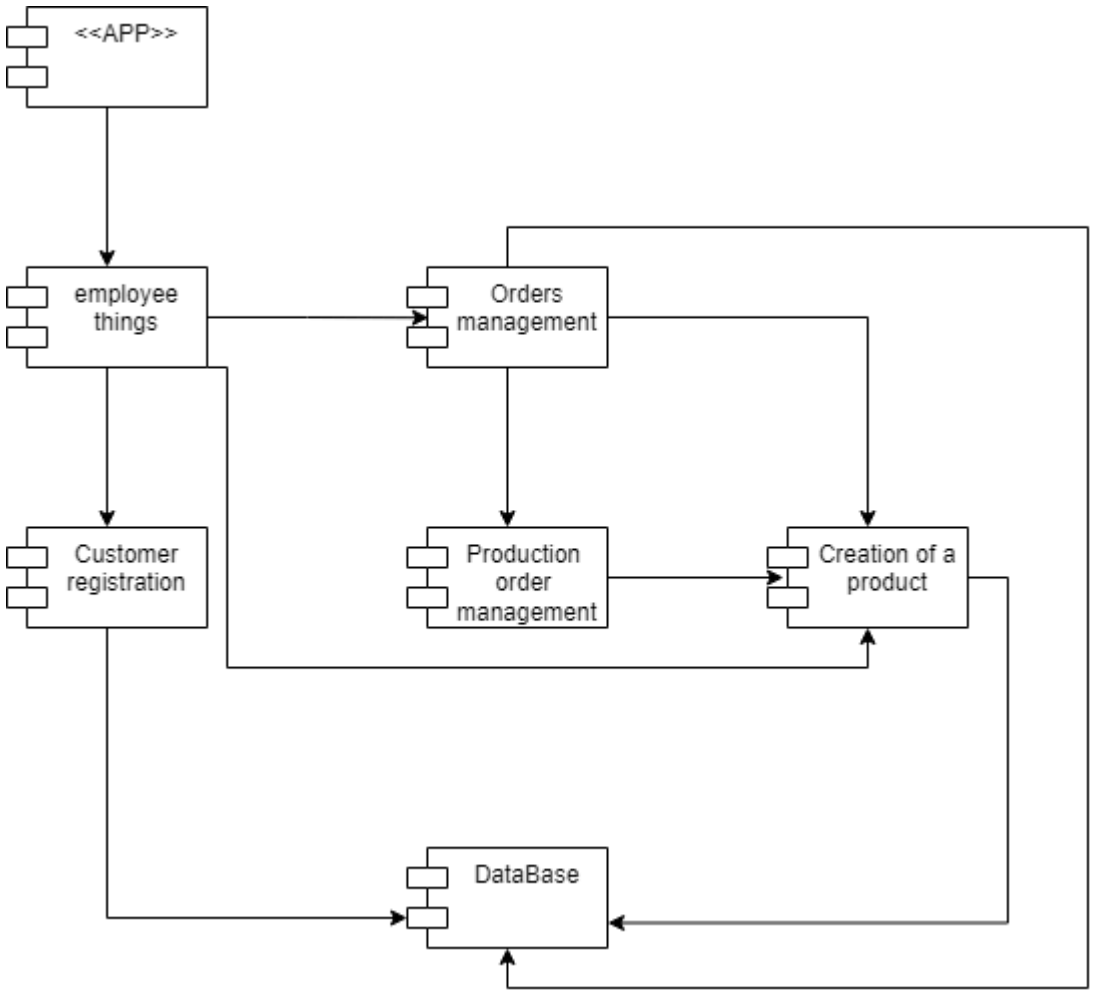
The developed software will have a user manual that offers all the information necessary for its use and will have monthly updates made by our developers.

Portability

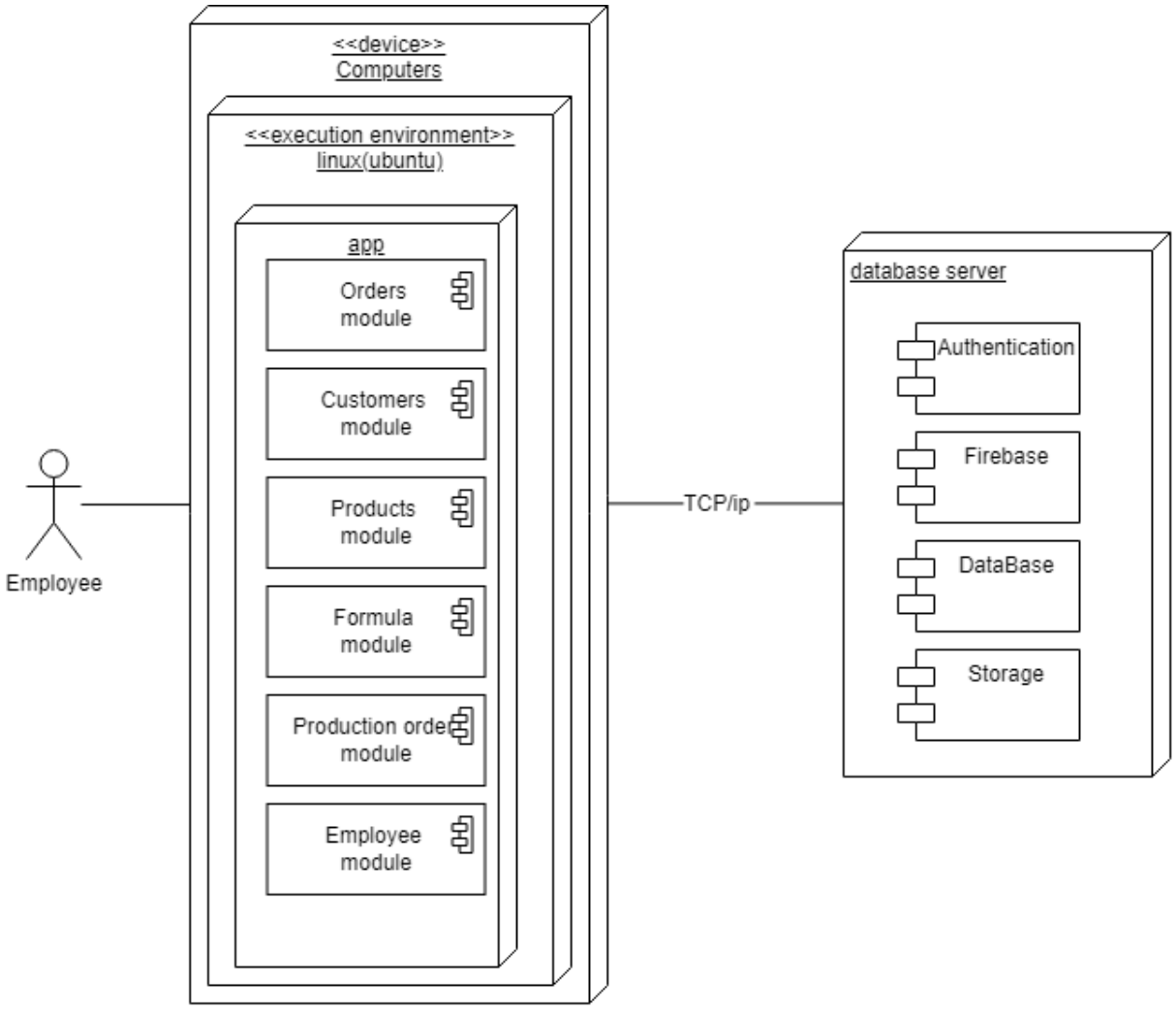
The Production Orders system is compatible with any device with Windows 10 or higher operating system.

4. ANNEX

COMPONENT DIAGRAM

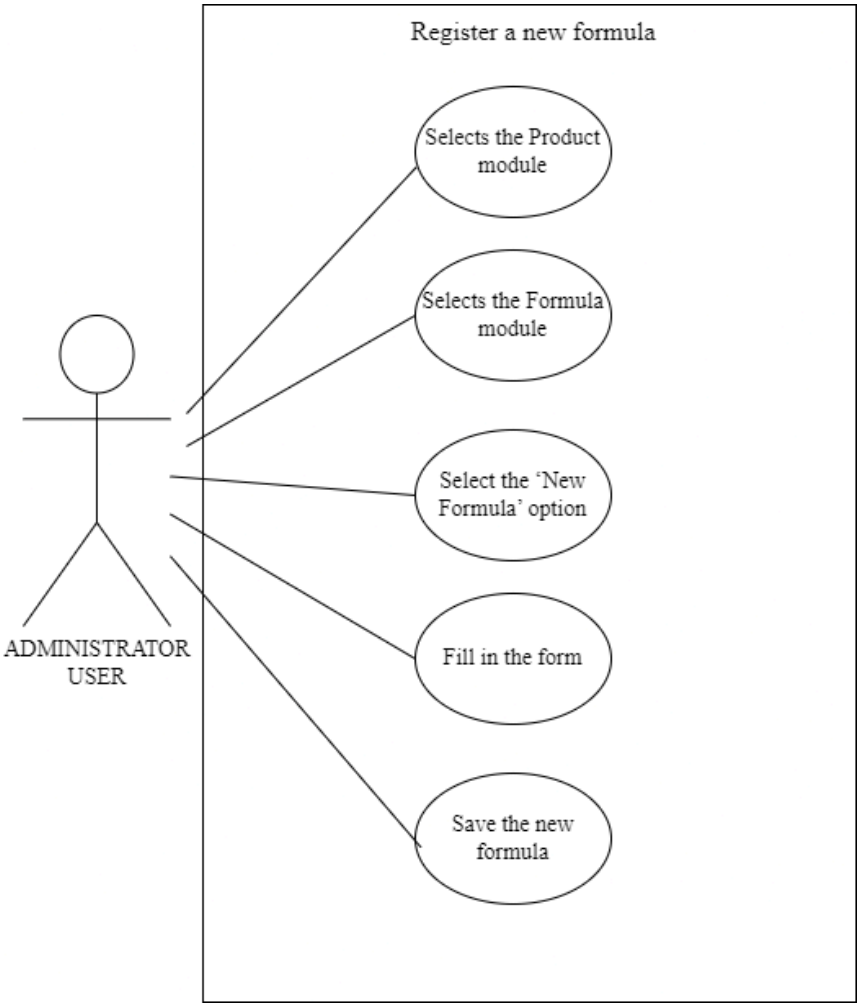


DEPLOYMENT DIAGRAM

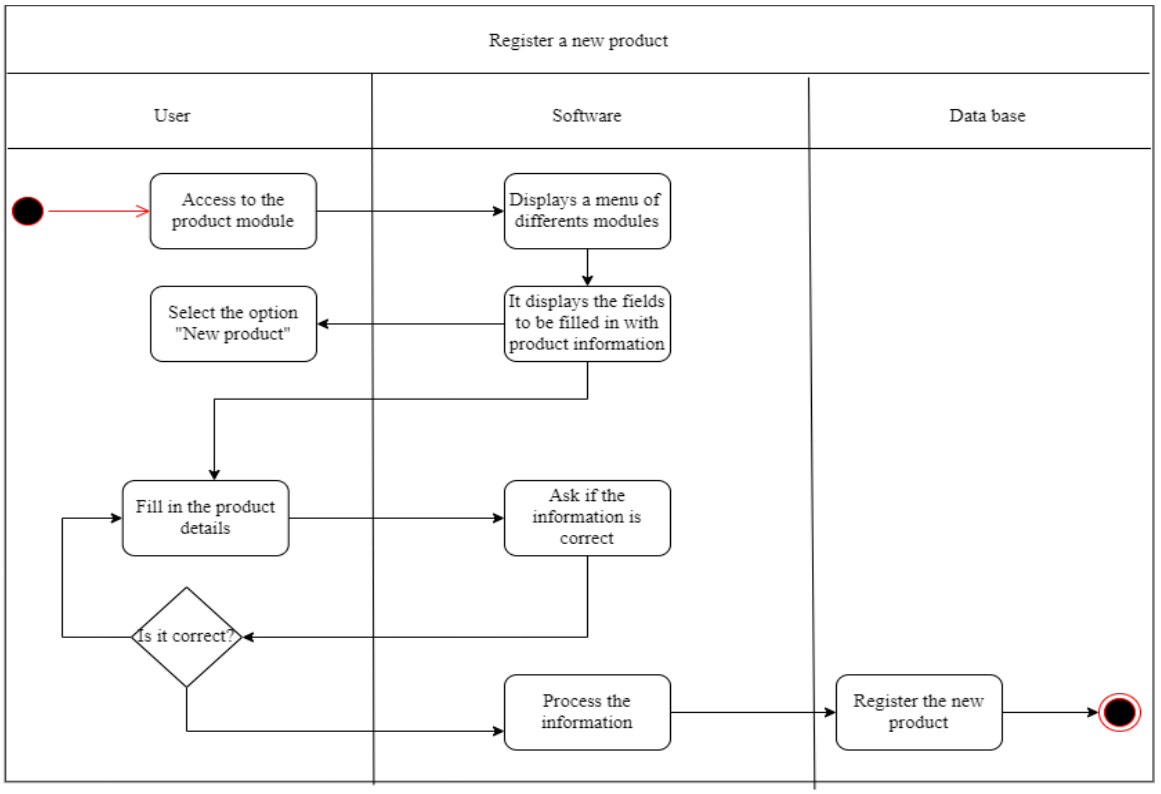


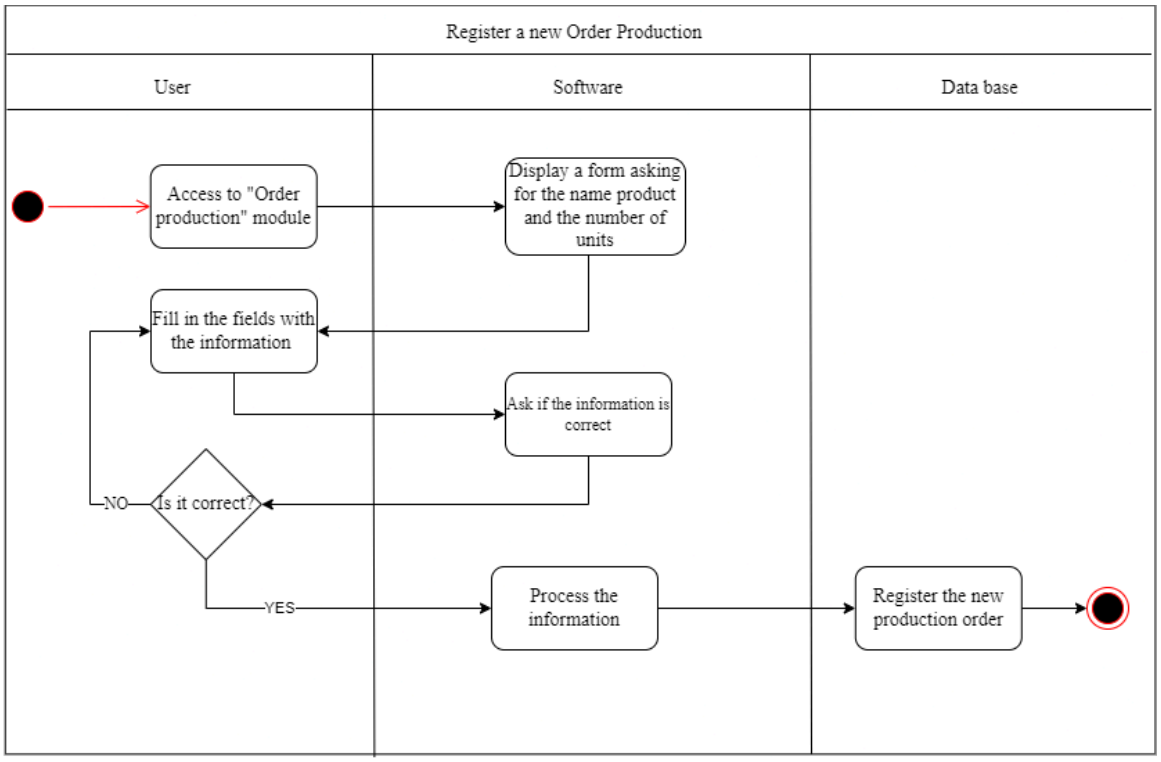
USE CASE

Name	Register a new formula
Author	Martínez Pérez Diana
Date	23/09/24
Description	Allows the user to register a new formula to produce a product.
Actors	User
Preconditions	Must have the formula in order to be registered.
Normal flow	The product module is selected. The system displays a list of the different options available. The Formula option is selected. The system displays a list of the different options available. The ‘New formula’ option is selected. The system displays a form to obtain the formula information. The user fills in the form with the information and presses 'Submit'. The system asks if the information is correct. The user selects Yes.
Alternative Flow	The system asks if the information is correct. The user presses ‘'No’. The system returns the user to the form. The user completes the form with the correct information and presses 'Submit'. The system asks if the information is correct. The user selects Yes.
Postconditions	The system prints the message "Formula correctly registered"

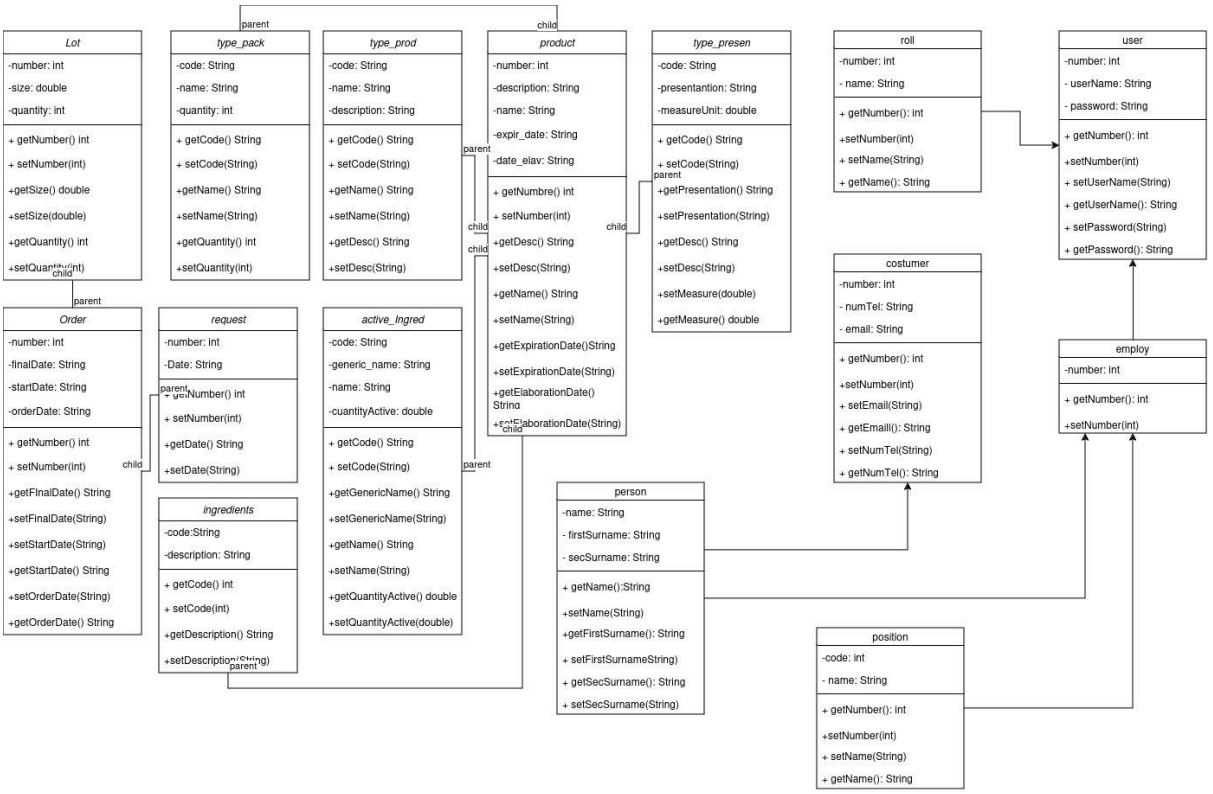

Register a new formula
Selects the Product module
Selects the Formula module
Select the 'New Formula' option
Fill in the form
Save the new formula
ADMINISTRATOR USER

ACTIVITY DIAGRAM

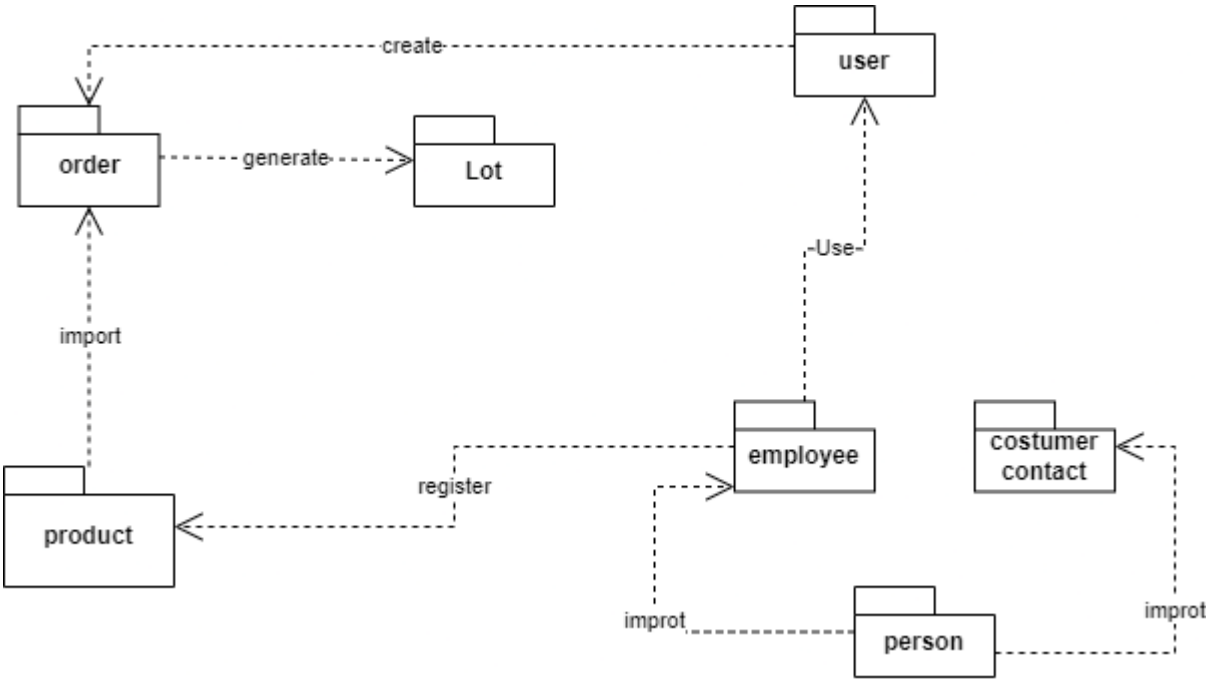




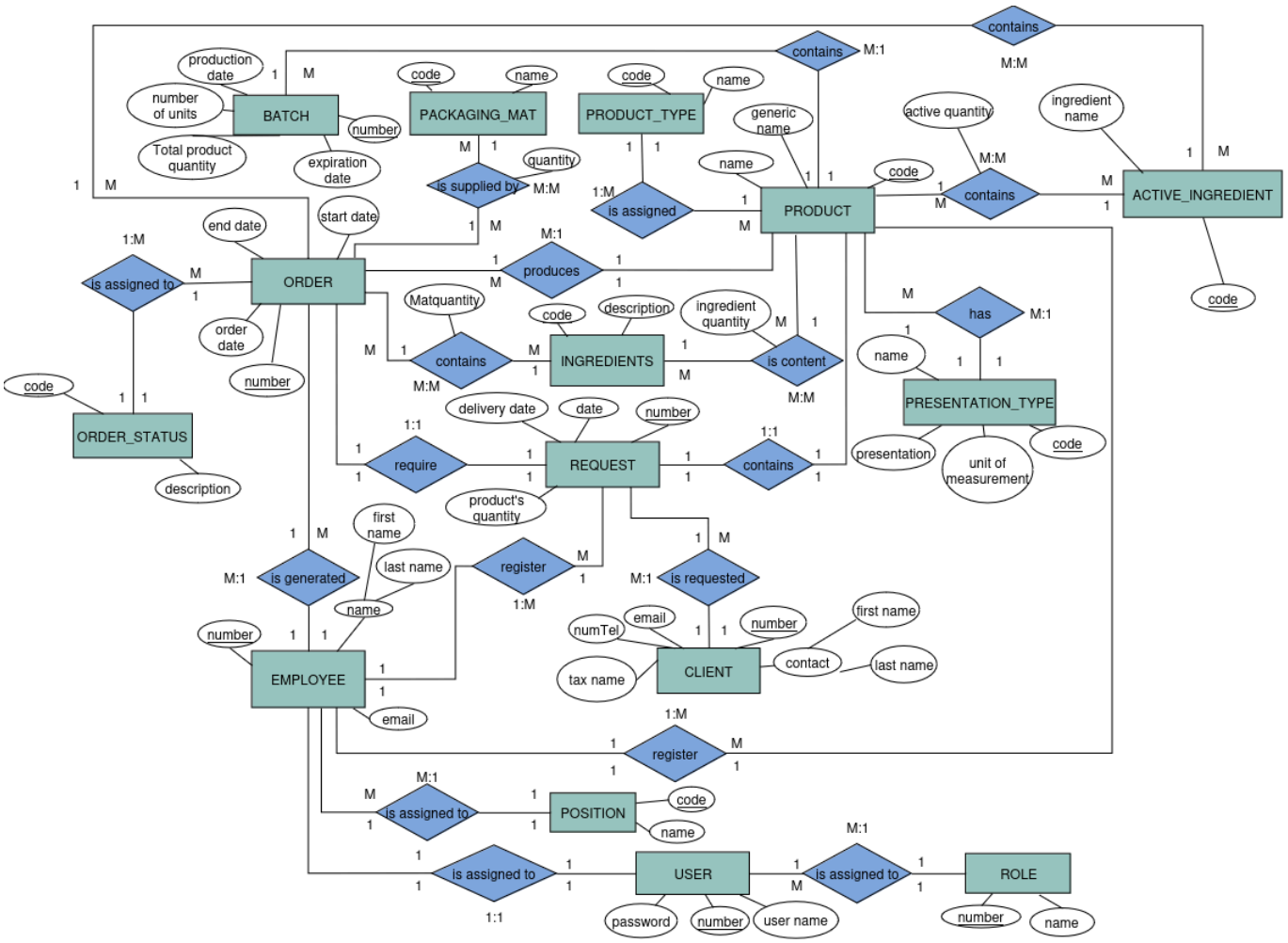
CLASS DIAGRAM



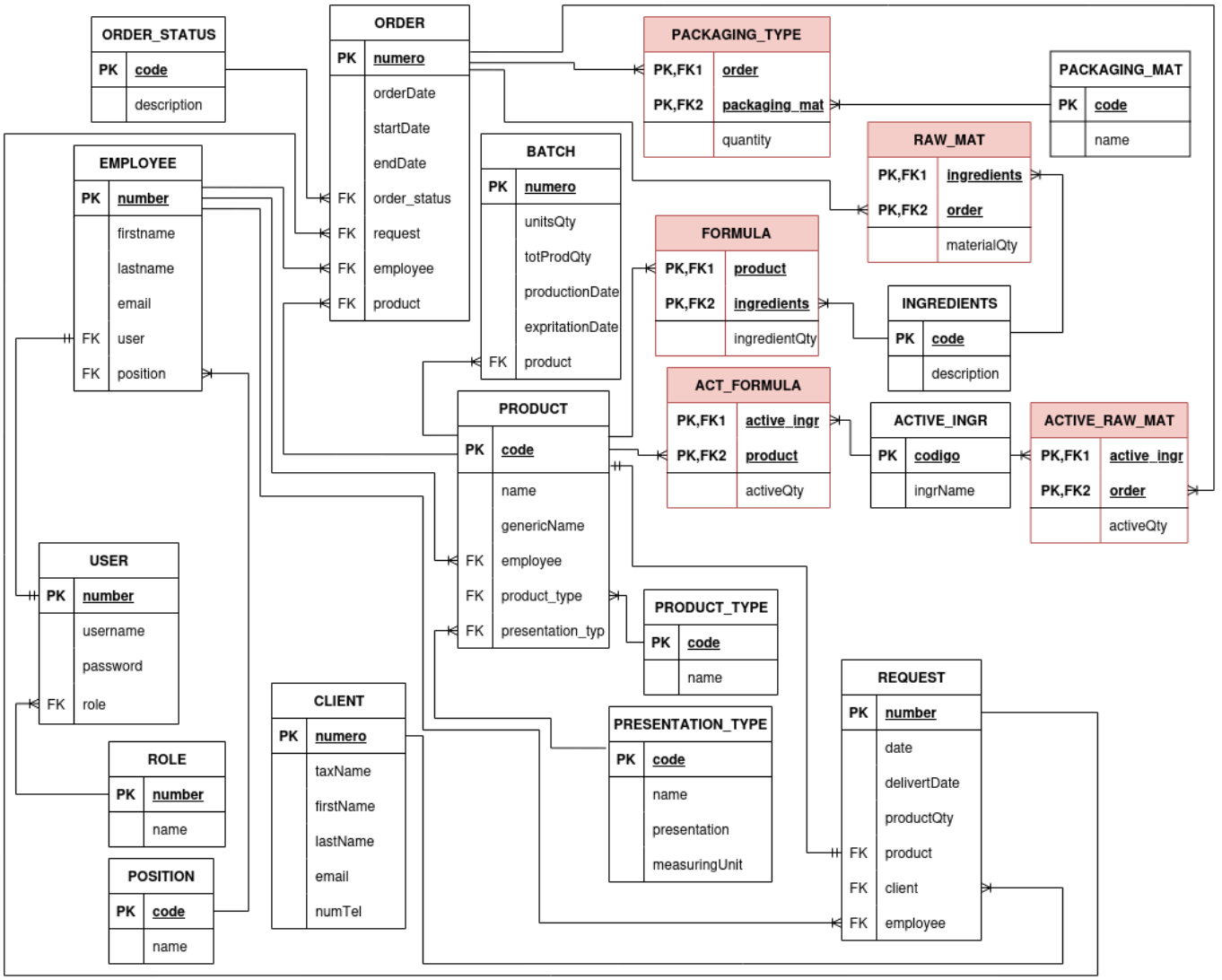
PACKAGE DIAGRAM



ENTITY RELATIONSHIP DIAGRAM (ERD)

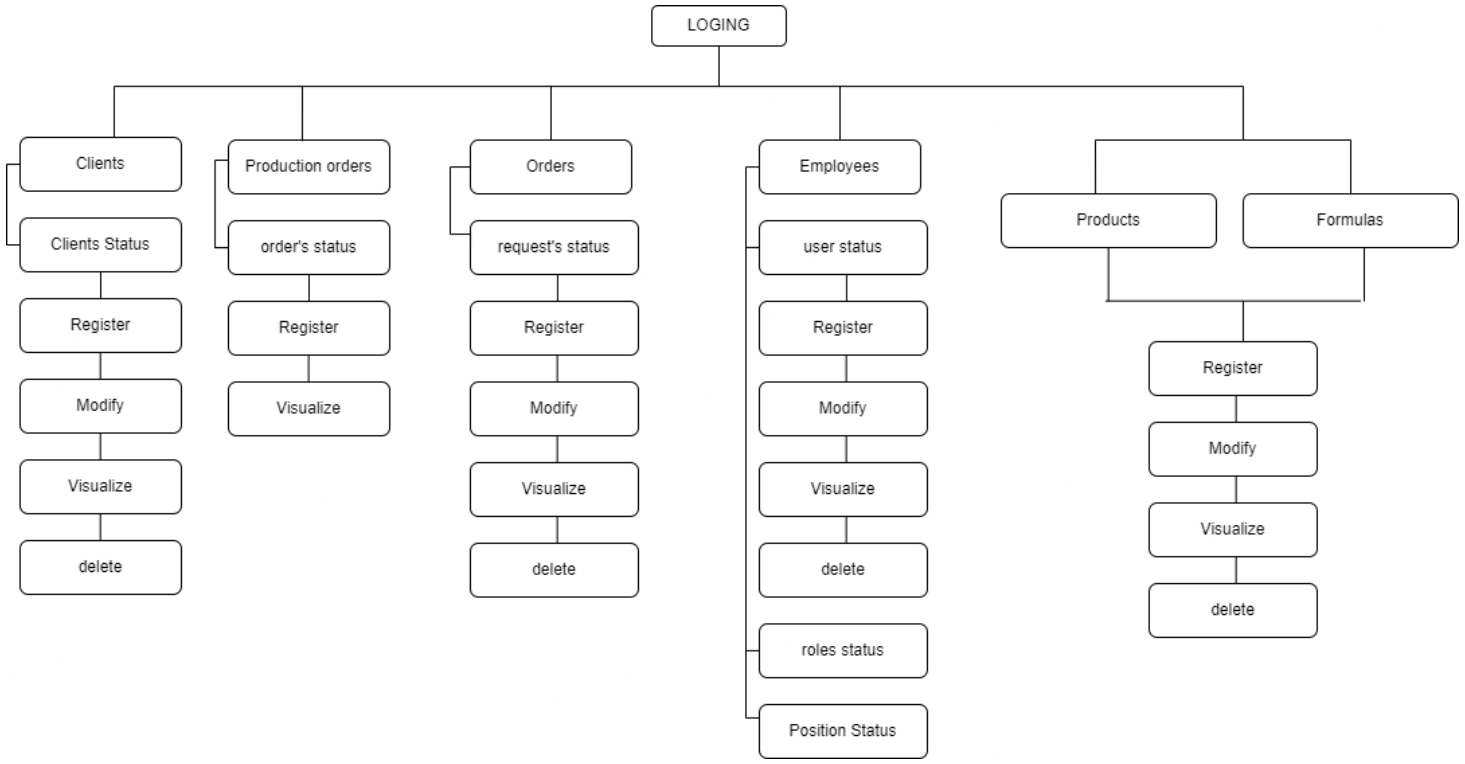


RELATIONAL MODEL (RM)

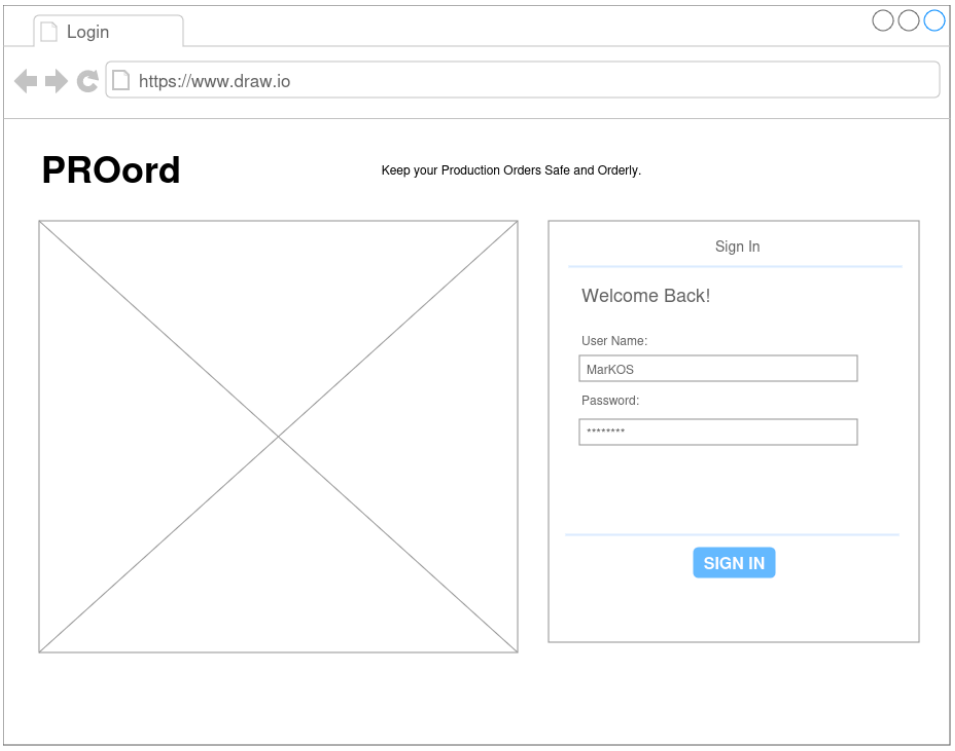


WIREFRAMES

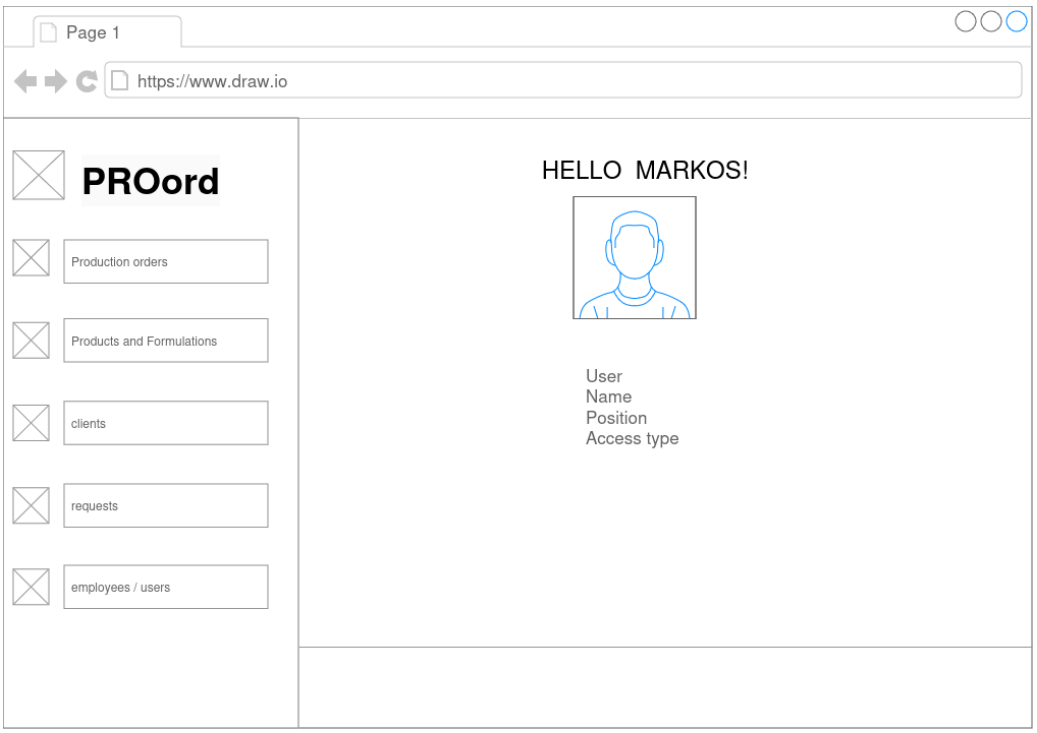
Architecture



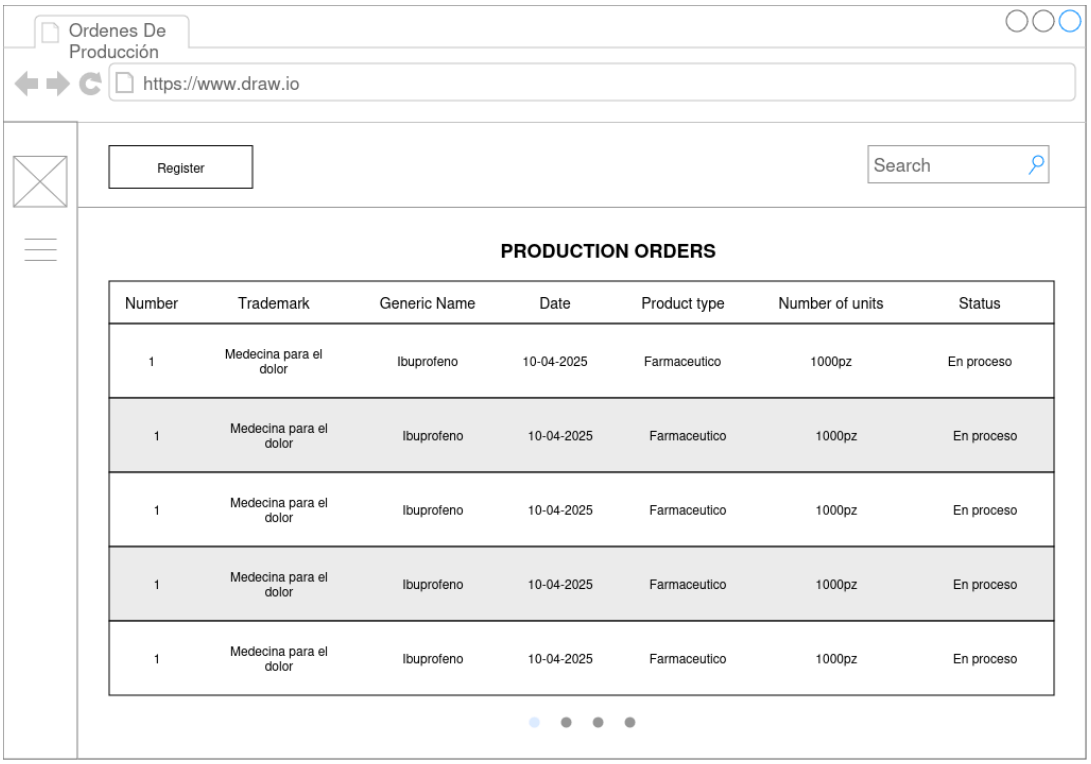
Login



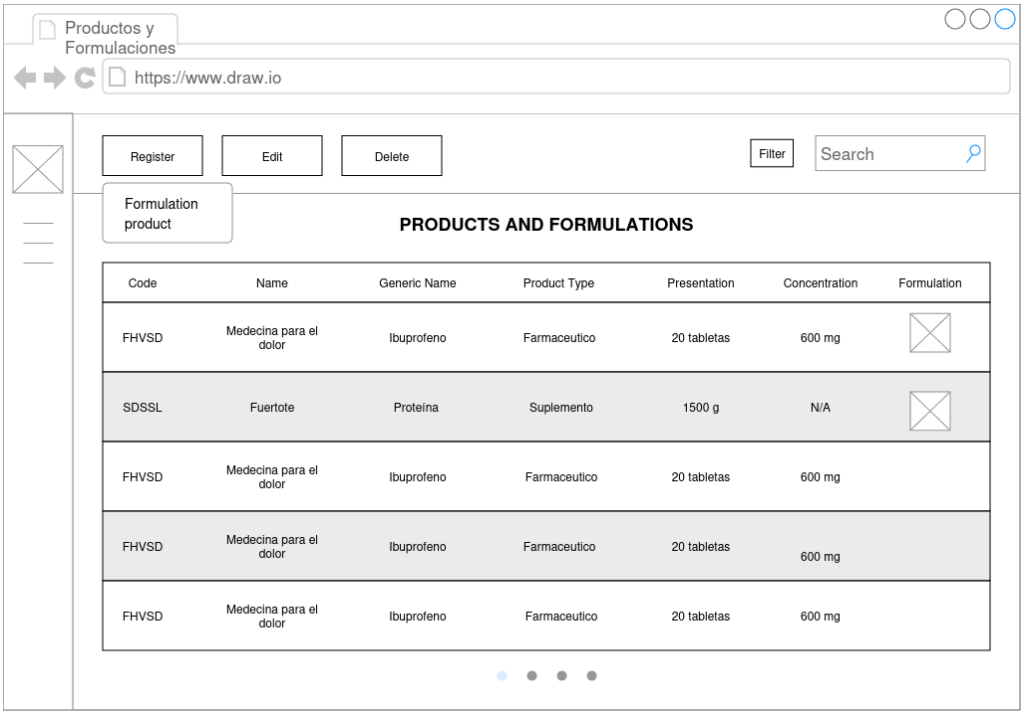
Admin options



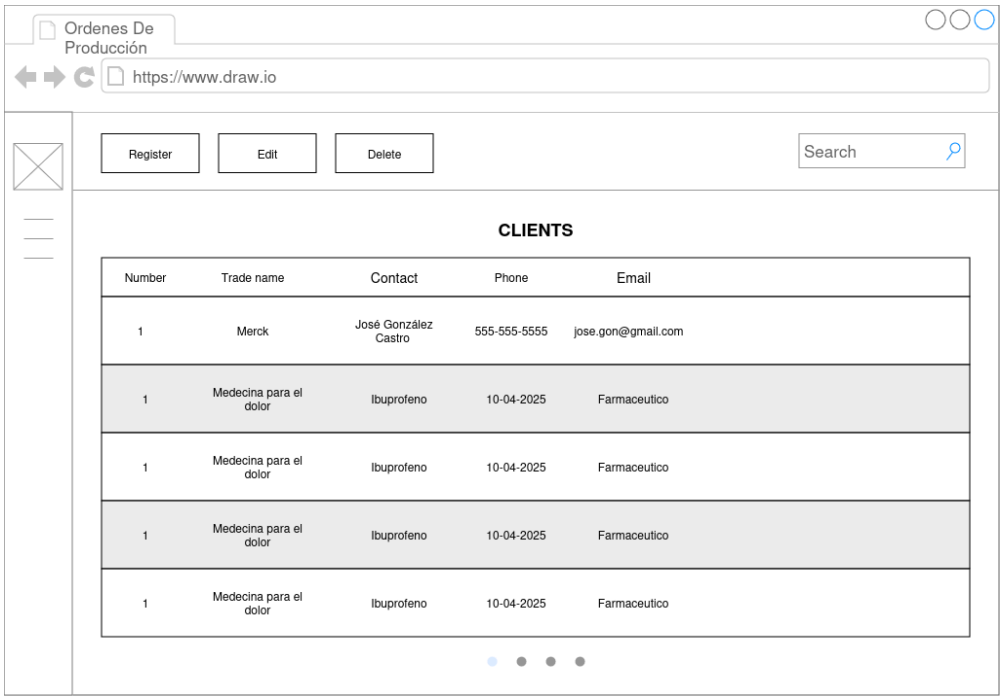
Production orders



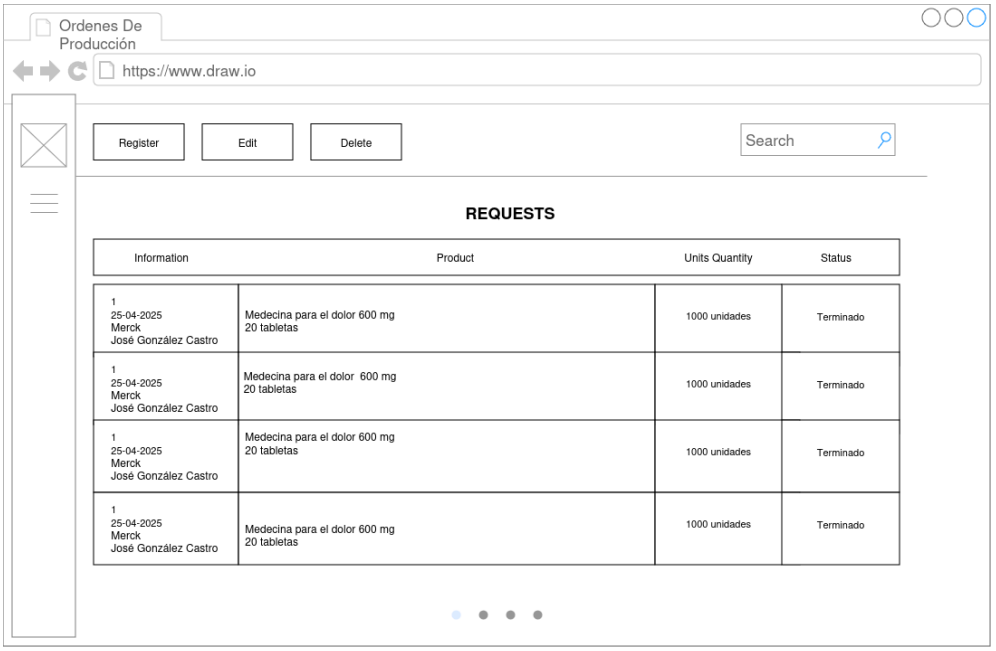
Products and formulations



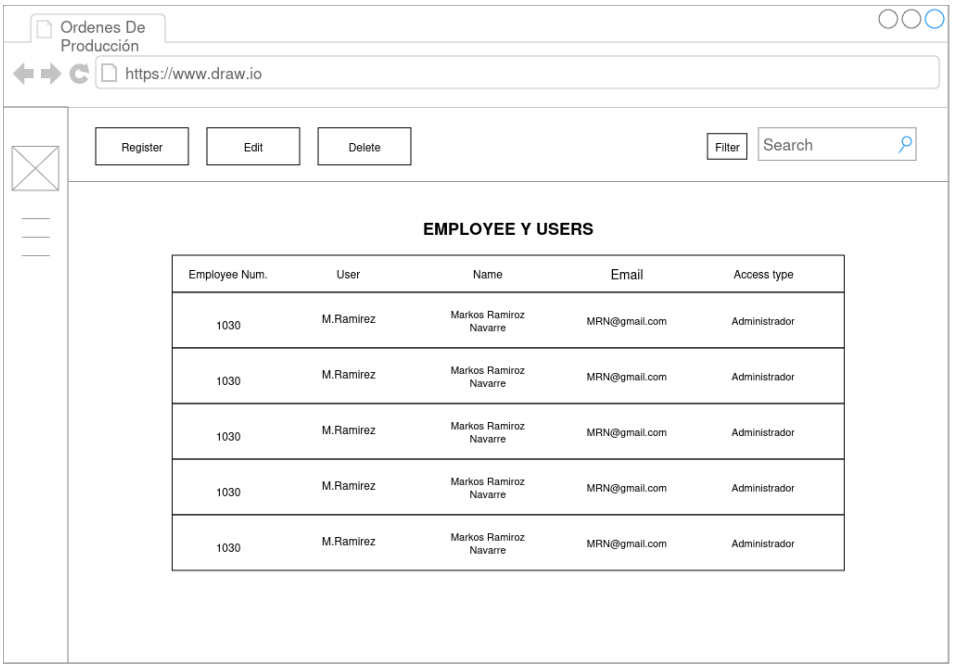
Clients



Requests



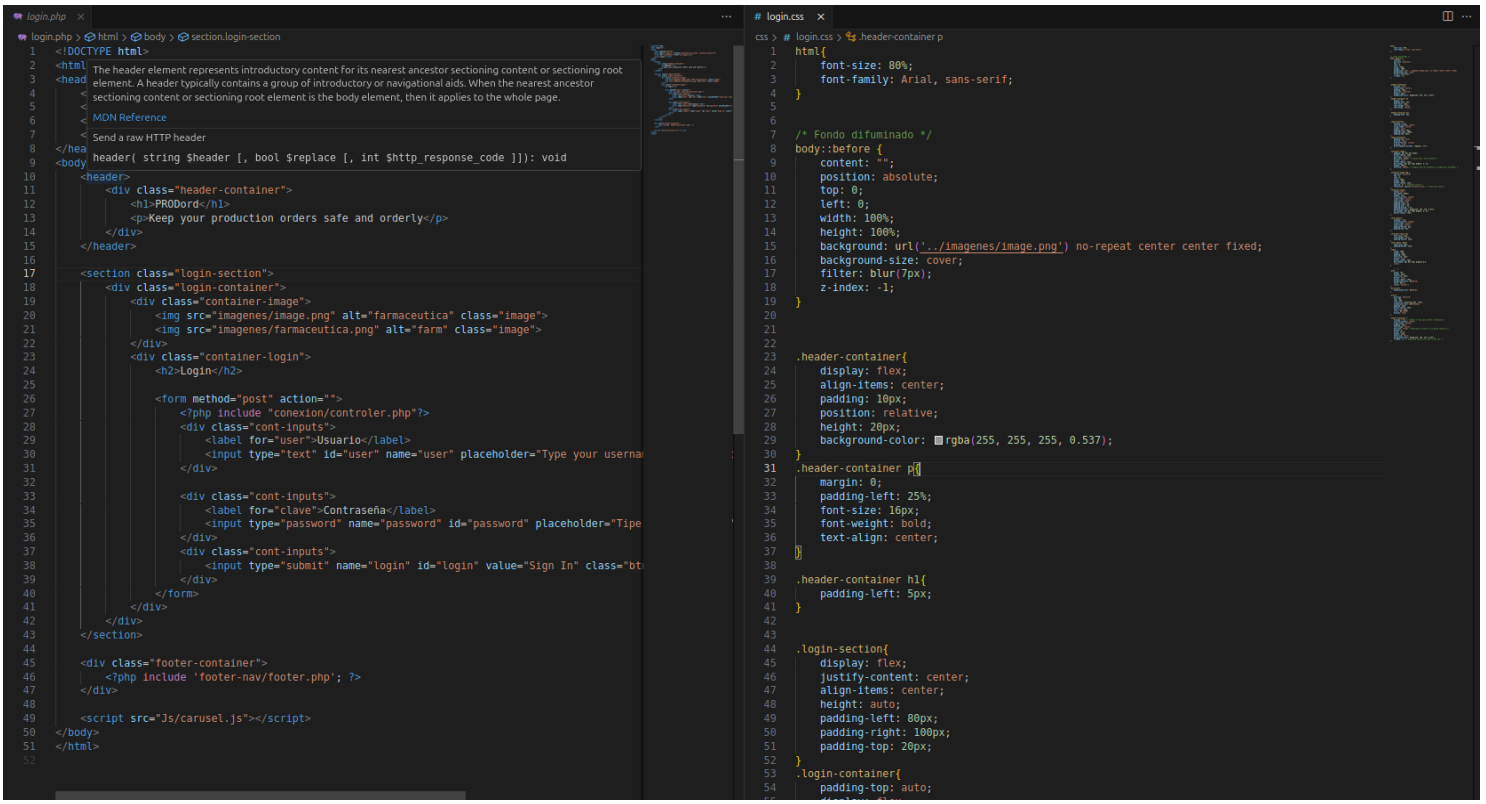
Employee & Users



CODIFICATION IN PHP, JAVASCRIPT, HTML & CSS

Login

For the site's login, a carousel image was created to show a larger design; in a single section. Divs were added to border the screen and center the username and password request.

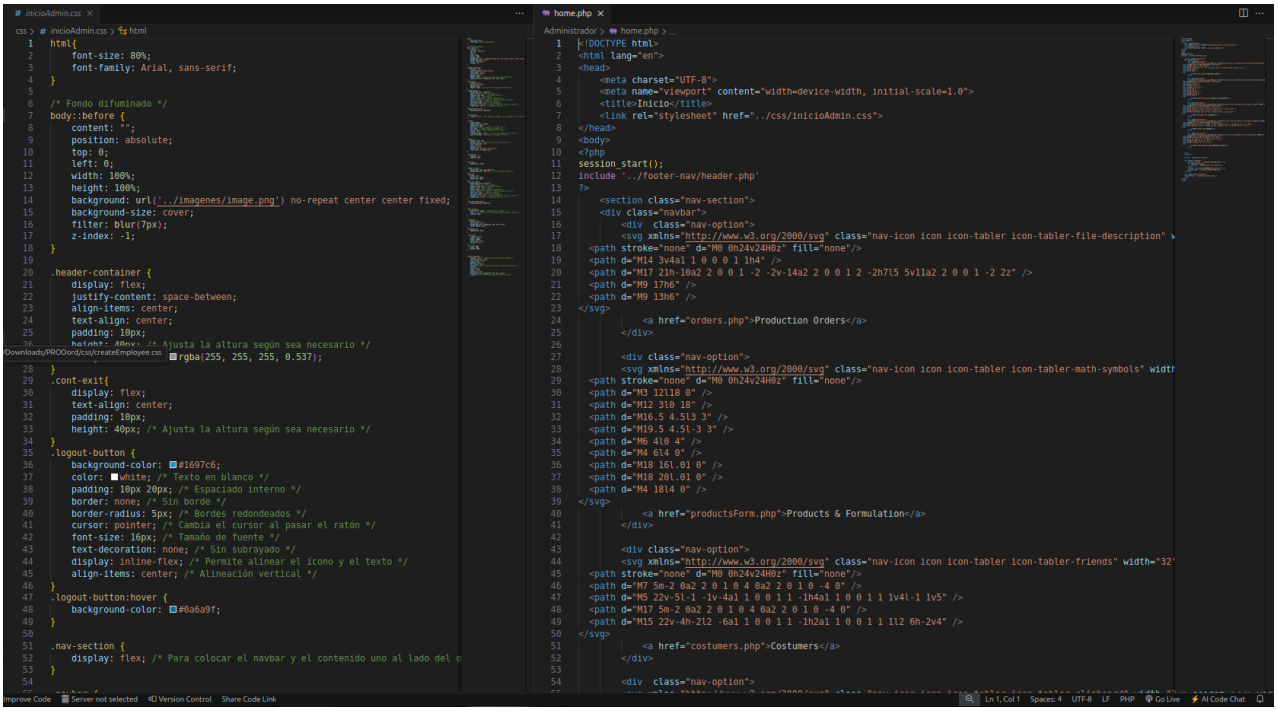


```php
<!DOCTYPE html>
<html
<head
</hea
<body
    <header>
        <div class="header-container">
            <h1>PRODord</h1>
            <p>Keep your production orders safe and orderly</p>
        </div>
    </header>

    <section class="login-section">
        <div class="login-container">
            <div class="container-image">
                <img src="imagenes/image.png" alt="farmaceutica" class="image">
                <img src="imagenes/farmaceutica.png" alt="farm" class="image">
            </div>
            <div class="container-login">
                <h2>Login</h2>

                <form method="post" action="">
                    <?php include "conexion/controler.php"?>
                    <div class="cont-inputs">
                        <label for="user">Usuario</label>
                        <input type="text" id="user" name="user" placeholder="Type your usern
                    </div>

                    <div class="cont-inputs">
                        <label for="clave">Contraseña</label>
                        <input type="password" name="password" id="password" placeholder="Tipe
                    </div>
                    <div class="cont-inputs">
                        <input type="submit" name="login" id="login" value="Sign In" class="bt
                    </div>
                </form>
            </div>
        </div>
    </section>

    <div class="footer-container">
        <?php include 'footer-nav/footer.php'; ?>
    </div>

    <script src="Js/carusel.js"></script>
</body>
</html>
```

```css
html{
    font-size: 80%;
    font-family: Arial, sans-serif;
}


/* Fondo difuminado */
body::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: url('../imagenes/image.png') no-repeat center center fixed;
    background-size: cover;
    filter: blur(7px);
    z-index: -1;
}



.header-container{
    display: flex;
    align-items: center;
    padding: 10px;
    position: relative;
    height: 20px;
    background-color: rgba(255, 255, 255, 0.537);
}
.header-container p{
    margin: 0;
    padding-left: 25%;
    font-size: 16px;
    font-weight: bold;
    text-align: center;
}

.header-container h1{
    padding-left: 5px;
}


.login-section{
    display: flex;
    justify-content: center;
    align-items: center;
    height: auto;
    padding-left: 80px;
    padding-right: 100px;
    padding-top: 20px;
}
.login-container{
    padding-top: auto;
```

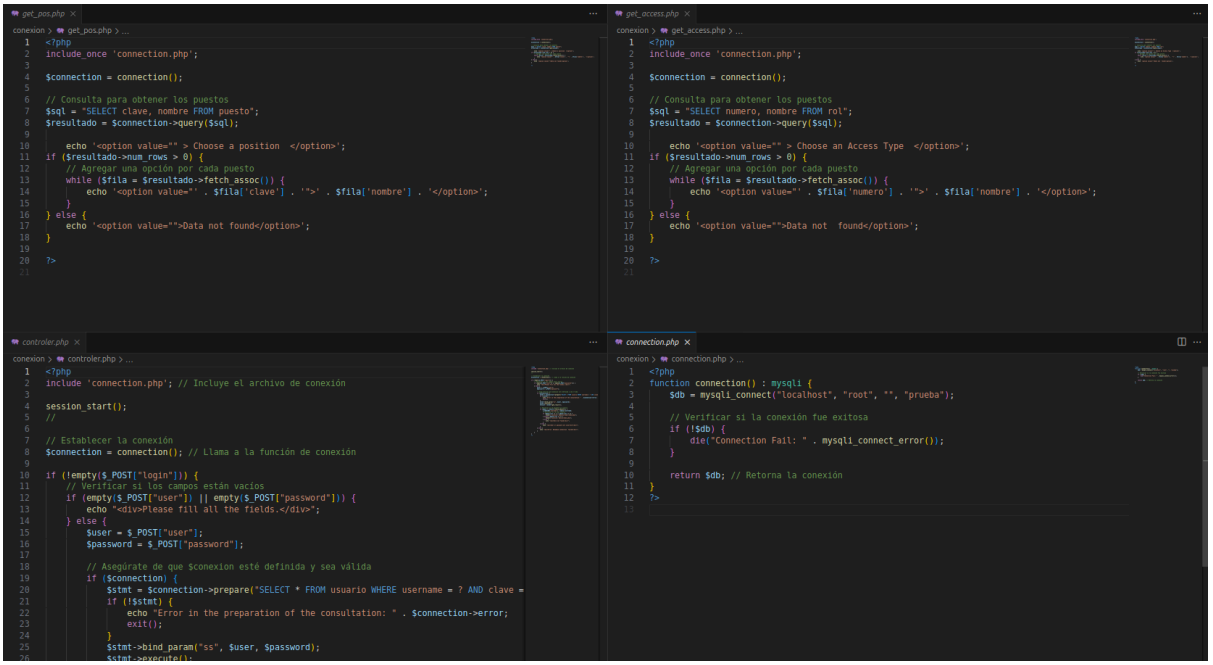
Home after login

On the home page after login, a navigation bar was designed on one side of the screen to select different modules such as Production Orders, Products and Recipes, Customers, Orders, Employees and Users. On the home page it displays the information of the logged in user, such as their username, employee name, their position and the type of access they have.

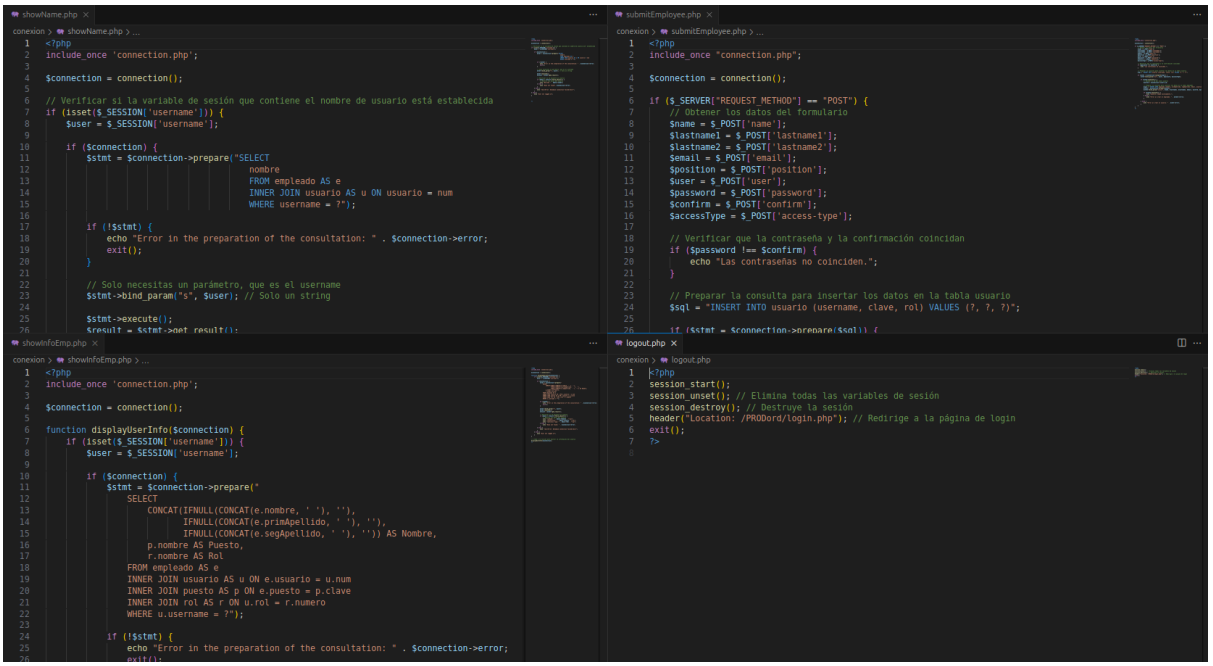


Connections

- “connection.php” is responsible for establishing the connection with the database.
- “controler.php” is used as an authentication method that verifies a user's login and redirection based on the user's role.
- “get_access.php” is used to generate a drop-down menu (<select>) with the available roles that are stored in the database and allows the user to select a specific role from the data that was obtained from the database.
- “get_pos.php” is used to connect to a database, retrieve the positions availables from the positions table, and generate an HTML drop-down list of those positions, if there is not one available, it provides a message indicating that no data was found.

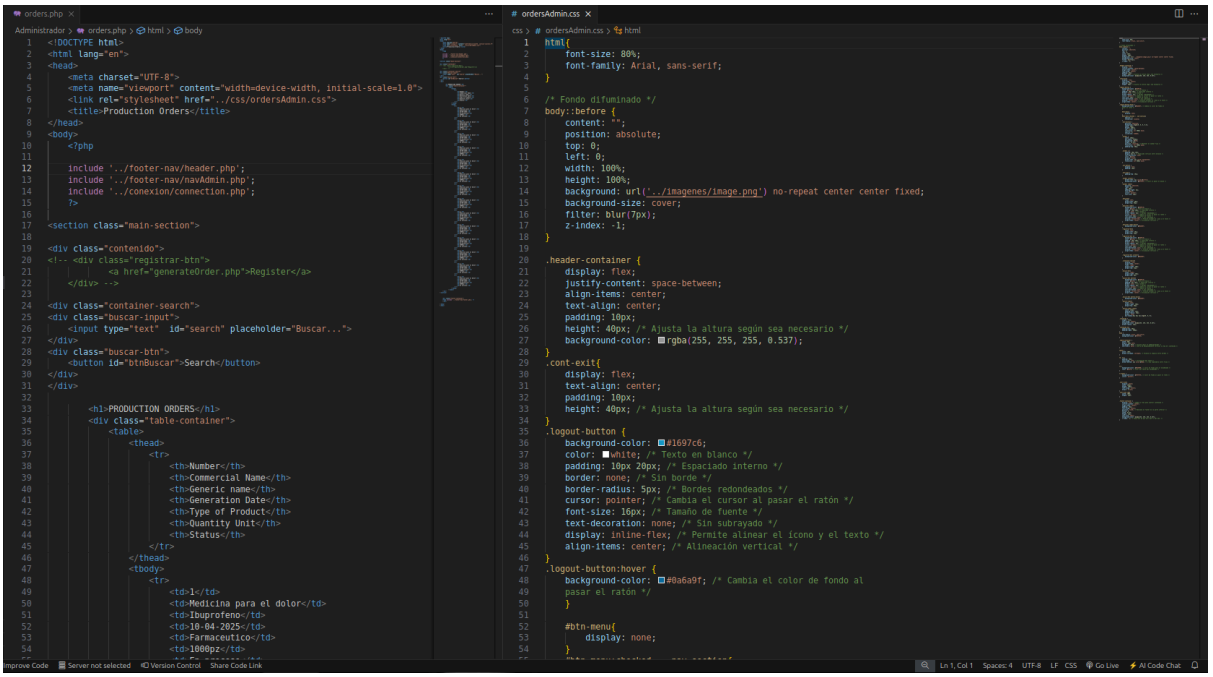


- “showInfoEmp.php” is used for retrieving user information that finds a logged in user, queries their information in a database and displays it on the web page.
- “showName.php” is used to check if a user is logged in, retrieves their name from the database using their username, and displays a welcome message.
- “submitEmployee.php” is used to handle user registration by connecting to a database and inserting user and employee information into two tables: user (for user data) and employee (for employee details).
- “logout.php” is used for logout which clears the user session data, destroys the session and then redirects the user to the login page.



Production orders

In this section, a table of pharmaceuticals and dietary supplements production orders is designed. Each row of the table contains details of a product in process with the following columns: Number, Trade Name, Generic Name, Generation Date, Product Type, Quantity, and Status. The section also has navigation options such as a menu, a logout button, and a search field to filter results.

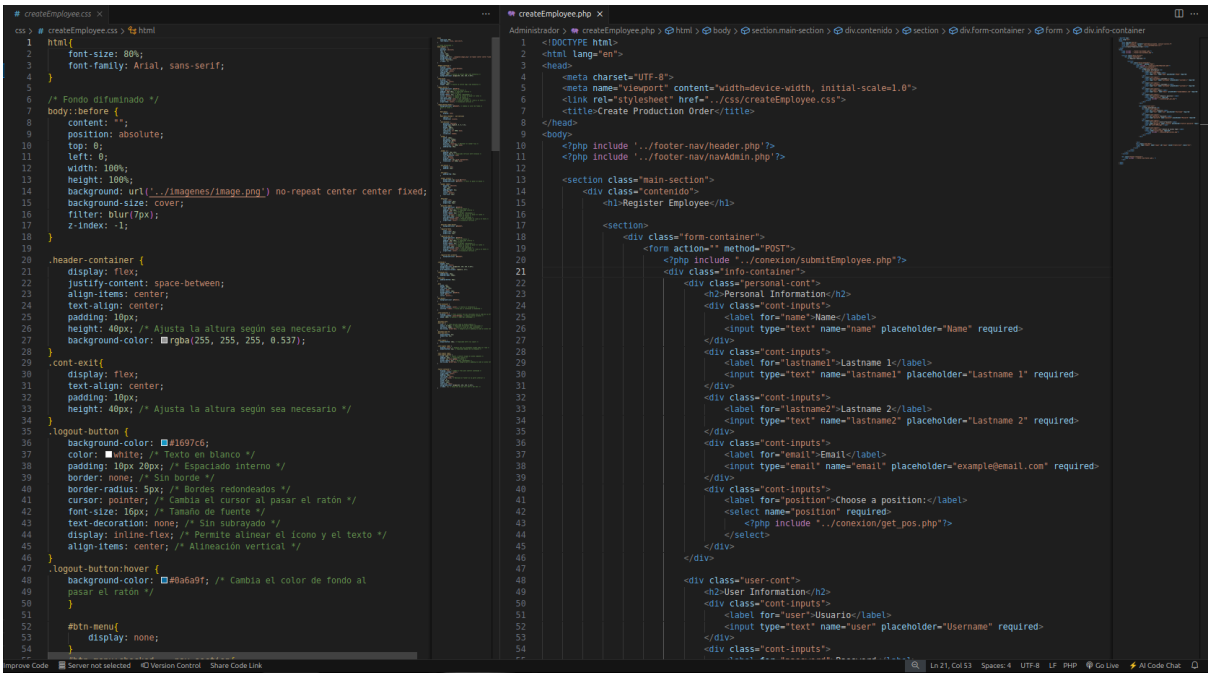


```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../css/ordersAdmin.css">
    <title>Production Orders</title>
</head>
<body>
    <?php

    include '../footer-nav/header.php';
    include '../footer-nav/navAdmin.php';
    include '../conexion/connection.php';
    ?>

<section class="main-section">

<div class="contenido">
<!-- <div class="registrar-btn">
        <a href="generateOrder.php">Register</a>
    </div> -->

<div class="container-search">
<div class="buscar-input">
    <input type="text"  id="search" placeholder="Buscar...">
</div>
<div class="buscar-btn">
    <button id="btnBuscar">Search</button>
</div>
</div>

        <h1>PRODUCTION ORDERS</h1>
        <div class="table-container">
            <table>
                <thead>
                    <tr>
                        <th>Number</th>
                        <th>Commercial Name</th>
                        <th>Generic name</th>
                        <th>Generation Date</th>
                        <th>Type of Product</th>
                        <th>Quantity Unit</th>
                        <th>Status</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td>1</td>
                        <td>Medicina para el dolor</td>
                        <td>Ibuprofeno</td>
                        <td>10-04-2025</td>
                        <td>Farmaceutico</td>
                        <td>1000pz</td>
```

```css
html{
    font-size: 80%;
    font-family: Arial, sans-serif;
}

/* Fondo difuminado */
body::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: url('../imagenes/image.png') no-repeat center center fixed;
    background-size: cover;
    filter: blur(7px);
    z-index: -1;
}

.header-container {
    display: flex;
    justify-content: space-between;
    align-items: center;
    text-align: center;
    padding: 10px;
    height: 40px; /* Ajusta la altura según sea necesario */
    background-color: rgba(255, 255, 255, 0.537);
}
.cont-exit{
    display: flex;
    text-align: center;
    padding: 10px;
    height: 40px; /* Ajusta la altura según sea necesario */
}
.logout-button {
    background-color: #1697c6;
    color: white; /* Texto en blanco */
    padding: 10px 20px; /* Espaciado interno */
    border: none; /* Sin borde */
    border-radius: 5px; /* Bordes redondeados */
    cursor: pointer; /* Cambia el cursor al pasar el ratón */
    font-size: 16px; /* Tamaño de fuente */
    text-decoration: none; /* Sin subrayado */
    display: inline-flex; /* Permite alinear el ícono y el texto */
    align-items: center; /* Alineación vertical */
}
.logout-button:hover {
    background-color: #0a6a9f; /* Cambia el color de fondo al
    pasar el ratón */
    }

    #btn-menu{
        display: none;
    }
```

Create employee

Management was added to the records of new users and employees for the database which includes the connection to the database, a form that receives a POST request for the name, surname, email, username, password and password verification , likewise selecting the position and type of access. This allows a new record to be inserted into the users table of the database if it successfully uses the user's ID to insert the data into the employees table.



```
# createEmployee.css
html{
    font-size: 80%;
    font-family: Arial, sans-serif;
}

/* Fondo difuminado */
body::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: url('../imagenes/image.png') no-repeat center center fixed;
    background-size: cover;
    filter: blur(7px);
    z-index: -1;
}

.header-container {
    display: flex;
    justify-content: space-between;
    align-items: center;
    text-align: center;
    padding: 10px;
    height: 40px; /* Ajusta la altura según sea necesario */
    background-color: rgba(255, 255, 255, 0.537);
}
.cont-exit{
    display: flex;
    text-align: center;
    padding: 10px;
    height: 40px; /* Ajusta la altura según sea necesario */
}
.logout-button {
    background-color: #1697c6;
    color: white; /* Texto en blanco */
    padding: 10px 20px; /* Espaciado interno */
    border: none; /* Sin borde */
    border-radius: 5px; /* Bordes redondeados */
    cursor: pointer; /* Cambia el cursor al pasar el ratón */
    font-size: 16px; /* Tamaño de fuente */
    text-decoration: none; /* Sin subrayado */
    display: inline-flex; /* Permite alinear el icono y el texto */
    align-items: center; /* Alineación vertical */
}
.logout-button:hover {
    background-color: #0a6a9f; /* Cambia el color de fondo al
    pasar el ratón */
    }

    #btn-menu{
        display: none;
    }
```

```
createEmployee.php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../css/createEmployee.css">
    <title>Create Production Order</title>
</head>
<body>
    <?php include '../footer-nav/header.php'?>
    <?php include '../footer-nav/navAdmin.php'?>

    <section class="main-section">
        <div class="contenido">
            <h1>Register Employee</h1>

            <section>
                <div class="form-container">
                    <form action="" method="POST">
                        <?php include "../conexion/submitEmployee.php"?>
                        <div class="info-container">
                            <div class="personal-cont">
                                <h2>Personal Information</h2>
                                <div class="cont-inputs">
                                    <label for="name">Name</label>
                                    <input type="text" name="name" placeholder="Name" required>
                                </div>
                                <div class="cont-inputs">
                                    <label for="lastname1">Lastname 1</label>
                                    <input type="text" name="lastname1" placeholder="Lastname 1" required>
                                </div>
                                <div class="cont-inputs">
                                    <label for="lastname2">Lastname 2</label>
                                    <input type="text" name="lastname2" placeholder="Lastname 2" required>
                                </div>
                                <div class="cont-inputs">
                                    <label for="email">Email</label>
                                    <input type="email" name="email" placeholder="example@email.com" required>
                                </div>
                                <div class="cont-inputs">
                                    <label for="position">Choose a position:</label>
                                    <select name="position" required>
                                        <?php include "../conexion/get_pos.php"?>
                                    </select>
                                </div>
                            </div>

                            <div class="user-cont">
                                <h2>User Information</h2>
                                <div class="cont-inputs">
                                    <label for="user">Usuario</label>
                                    <input type="text" name="user" placeholder="Username" required>
                                </div>
                                <div class="cont-inputs">
```

TESTING DATABASE

home > genkai > Downloads > create database prueba.sql

```sql
create database prueba

create table puesto(
    clave int PRIMARY KEY AUTO_INCREMENT,
    nombre varchar (20) not null
)

insert into puesto (clave, nombre) values (1, 'Supervisor'),
(2, 'Operador')

create table rol(
    numero int PRIMARY KEY AUTO_INCREMENT not null,
    nombre varchar (30) not null
)

insert into rol (numero, nombre) values (1, 'Basico'),
(2, 'Administrador'),
(3, 'Administrador General')

create table usuario(
    num int primary key AUTO_INCREMENT not null,
    username varchar(15) not null,
    clave varchar(20) not null,
    rol int not null,
    foreign key (rol) references rol(numero)
)

insert into usuario(num, username, clave, rol) values (1,'admin', '13245', 3),
(2,'amedina', '123456', 2),
(3,'mramirez', '123456', 2)

create table empleado(
    numero int PRIMARY KEY AUTO_INCREMENT,
    nombre varchar (30) not null,
    primApellido varchar (30) not null,
    segApellido varchar (30) null,
    email varchar (50) not null,
    usuario int,
    puesto int,
    foreign key (usuario) references usuario(num),
    foreign key (puesto) references puesto(clave)
)

insert into empleado (numero, nombre, primApellido, segApellido, email, usuario, puesto) values
(1, 'Alan', 'Medina', 'Becerra', 'alanoswaldo53@gmail.com', 2, 1),
(2, 'Marcos', 'Ramirez', 'Navarro', 'marcos@gmail.com', 3, 1)

```