



Career: IT Multiplatform Software Development

Project: Industrial Inventory Control System

Subject: Application Design

Work: Software Requirements Specification

Members:

Moreno Ramirez Josue Elihu

Perez Salazar Jesús Santiago

Dominguez Galvez Antonio Azael

Bojorquez Barraza Jesus Alejandro

Group: 4B

Teacher: Ray Brunett Parra Galaviz

Table of Contents

1. Introduction	2
1.1 Purpose	2
1.2 Project Scope	2
1.3 Personnel Involved	2
1.4 Definitions, acronyms and abbreviations	3
1.5 References	3
1.6 Summary	4
2. General Description	4
2.1 Product Perspective	4
2.2 Use Case Diagram	5
2.3 User characteristics	5
2.4 Restrictions	6
2.5 Assumptions and dependencies	6
3. Specific Requirements	7
Functional Requirements	7
Non-functional requirements	10
3.1 Common requirements of interfaces	12
3.1.1 User interfaces	12
3.1.2 Hardware interfaces	12
3.1.3 Software interfaces	12
3.1.4 Communication interfaces	12
3.2 Functional requirements	12
3.3 Non-functional requirements	14

1. Introduction

This document contains a Software Requirements Specification (SRS) for an Industrial warehouse system focus on a factory that makes laptop. The document has been structured based on the guidelines given by the IEEE 830 standard.

1.1 Purpose

The purpose of this document is to define the functional and non-functional specifications for the development of an Industrial Inventory Control System. The system is designed to optimize and digitalize the management of processes related to the reception, storage, and dispatch of materials in a warehouse dedicated to the assembly and distribution of laptops. This system will be used by the personnel responsible for inventory management, enabling accurate tracking, categorization, and reporting of materials to ensure operational efficiency.

1.2 Project Scope

This Software Requirements Specification focuses on the development of an inventory control system for a warehouse in a laptop manufacturing factory. The primary goal is to streamline the management of materials through automated categorization, allocation to appropriate storage locations, and real-time tracking using RFID (IoT).

1.3 Personnel Involved

Name	Bojorquez Barraza Jesus Alejandro
Role	Documenter, Data analyst
Professional Category	TSU-DSM
Responsibility	Project documentation, database modeling and processes.
Contact	0323105855@ut-tijuana.edu.mx

Name	Perez Salazar Jesé Santiago
Role	Designer, Data analyst
Professional Category	TSU-DSM
Responsibility	System design and database modeling
Contact	0323105984@ut-tijuana.edu.mx

Name	Dominguez Galvez Antonio Azael
Role	Programmer
Professional Category	TSU-DSM
Responsibility	system coding
Contact	0323105973@ut-tijuana.edu.mx

Name	Moreno Ramirez Josue Elihu
Role	Documenter, Programmer
Professional Category	TSU-DSM
Responsibility	Project documentation and system coding
Contact	0323105975@ut-tijuana.edu.mx

1.4 Definitions, acronyms and abbreviations

Name	Description
User	Person who will use the system to manage equipment maintenance.
SRS	Software Requirements Specification

1.5 References

Title	Reference
Standard IEEE 830 - 1998	IEEE

1.6 Summary

This document consists of multiple sections. The first section will introduce the project and provide a more general overview of the system's resource specifications.

The second section will provide an overview of the product, where you will learn about the system, with the purpose of learning about the main functions it seeks to satisfy.

It will also contain a section where the specific and non-specific requirements of the system are defined in detail.

2. General Description

2.1 Product Perspective

The Industrial Inventory Control System is designed to enhance the management and tracking of materials within a warehouse dedicated to the assembly and storage of laptops. This system focuses on improving inventory accuracy, efficiency, and traceability by digitalizing and automating processes such as material reception, categorization, storage, and dispatch. The system integrates a centralized database containing detailed information about each material, including its description, supplier, quantity storage location etc. It enables the automated categorization of materials based on predefined criteria and assigns them to specific warehouse areas, ensuring optimized storage utilization and organization. By centralizing and automating inventory processes, this system aims to improve operational efficiency, reduce errors, and provide the tools necessary for effective inventory management in a highly dynamic manufacturing environment.

2.2 User characteristics

User type	Programmer
Training	Working
Activities	Programming, database management, and related technologies, such as RFID (IoT).

User type	Client
Training	Owner
Activities	System control and management. Use the system to oversee material reception, storage, and dispatch.

2.3 Restrictions

- Languages and technologies in use: HTML, CSS, PHP, SQL, React Native for mobile application development
- Will feature both web and mobile interfaces to allow flexibility for users
- The server must efficiently store and manage all inventory-related data, such as material information, warehouse assignments, and RFID tracking.

2.5 Assumptions and dependencies

- The equipment on which the system is run must be suitable for using 100% of its functionalities.

3. Specific Requirements

Functional Requirements

Requirement Identification:	FR 01
Requirement Name	Material Registration
Characteristics	The system will allow the registration of materials upon receipt.
Requirement Description	The administrator can register new materials, including details like quantity, description, supplier, date of receipt, batch number, and serial number.
Priority	High

Requirement Identification:	FR 02
Requirement Name	Material Categorization
Characteristics	The system will categorize materials based on their type.
Requirement Description	Automatically assign a category to the materials upon registration and allocate them to a warehouse with the corresponding category.
Priority	High

Requirement Identification:	FR 03
Requirement Name	Material Entry Report
Characteristics	The system will generate detailed reports of material entries.
Requirement Description	Generates an automatic report for each material entry, including specific data such as quantity, description and assigned category.
Priority	High

Requirement Identification:	FR 04
Requirement Name	Material Movement Management
Characteristics	The system will allow the management of material transfers between warehouses or processes.
Requirement Description	Enables users to register material movements, specifying origin, destination, and quantities, and generates movement reports.
Priority	High

Requirement Identification:	FR 05
Requirement Name	Material Dispatch Report
Characteristics	The system will allow reporting for material dispatches.
Requirement Description	Automatically generate reports for dispatched materials, detailing quantities, destination, and responsible personnel.
Priority	High

Requirement Identification:	FR 06
Requirement Name	Real-Time Inventory Tracking
Characteristics	The system will allow users to monitor inventory levels in real-time.
Requirement Description	Displays current stock levels for each material and notifies when stock is low or exceeds capacity.
Priority	High

Requirement Identification:	FR 07
Requirement Name	User Role
Characteristics	The system will two roles the administrator and the warehouse operator
Requirement Description	Administrators can register warehouse operator and see all data. The warehouse operator will be able to use the system and do action such as material registration, material categorization and material entry and dispatch
Priority	High

Non-functional requirements

Requirement identification:	NFR 01
Requirement Name:	Security
Characteristics:	The system will guarantee the security of the information.
Requirement Description:	Guarantee the security of the information that circulates in the program, preventing possible intruders from accessing it.
Priority of the Requirement	High

Requirement identification:	NFR 02
Requirement Name:	Scalability
Characteristics:	The system will be scalable.
Requirement Description:	Have the possibility of increasing the size of the program, made to adapt to greater volumes of data and users.
Priority of the	Average

Requirement	
--------------------	--

Requirement identification:	NFR 03
Requirement Name:	Portability
Characteristics:	The system will be portable.
Requirement Description:	That the program can run on different platforms.
Priority of the Requirement	Low

Requirement identification:	NFR 04
Requirement Name:	Efficiency
Characteristics:	The system will be efficient in its use.
Requirement Description:	The program should be efficient, avoiding slowness or unexpected closures that affect the user experience.
Priority of the Requirement	High

Requirement identification:	NFR 05
Requirement Name:	Reliability
Characteristics:	The system will be reliable.
Requirement Description:	The program should not be prone to errors, slowdowns, or sudden crashes, ensuring continuous operation.
Priority of the Requirement	High

Requirement identification:	NFR 06
Requirement Name:	Usability
Characteristics:	The system will be easy to use.
Requirement Description:	The program should be easy to use and understand, maintaining a simple interface so that both the administrator and the client can use it without difficulties.
Priority of the Requirement	High

3.1 Common requirements of interfaces

3.1.1 User interfaces

The user interface will consist of a window. The window will be displayed on the computer the user is using and will feature highly descriptive texts about what is happening in each section of the program, along with usage instructions so that the user can use the program without difficulties.

The mobile version of the system will provide a user-friendly experience on smartphones and tablets, featuring a simplified layout for easy interaction on smaller screens. Touch controls, such as optimized buttons and dropdowns. The interface will be adaptive, adjusting dynamically to different screen sizes and quick navigation will be facilitated by navigation bar.

3.1.2 Hardware interfaces

It will be necessary to have computer equipment in condition with the following characteristics:

- Network adapters
- 1.66GHz or higher processor

- Minimum memory of 256Mb
- Mouse
- Keyboard

And a smartphone in condition with the following characteristics:

- Network adapters (Wi-Fi)
- 1.2GHz or higher processor
- Minimum 256 GB of RAM

.1.3 Software interfaces

- Web application
- Mobile application

3.1.4 Communication interfaces

The servers, users and computers will be able to communicate with each other through standard Internet protocols, in order to communicate between the database and the computing equipment.

3.2 Functional requirements

Inventory Management

- Allows administrators and warehouse operators to register, update, and delete inventory items.
- Ensures tracking of inventory levels, including available stock and stock shortages.

Material Reception and Registration

- Enables warehouse operators to log incoming materials, including details such as supplier, quantity or serial number.
- Automatically categorizes materials based on predefined criteria and assigns them to appropriate storage locations.

Storage and Organization

- Implements an automated categorization system to allocate materials efficiently within the warehouse.

Material Movement Tracking

- Allows users to register material transfers between storage areas or different warehouses.
- Generates reports for material movements, detailing origin, destination, and quantities transferred.

Dispatch and Distribution Management

- Supports the tracking of materials dispatched for production or shipment.
- Generates detailed reports on dispatched materials, including destination, responsible personnel, and inventory updates.

User Role Management

- Defines two main user roles: Administrator and Warehouse Operator.
- Administrators can add and manage users, as well as oversee all warehouse operations.
- Warehouse Operators can perform inventory tasks such as material registration, categorization, and dispatch.

3.3 Non-functional requirements

- Security

Guarantee the security of the information that circulates in the program, preventing possible intruders from accessing it.

- Scalability

Have the possibility of increasing the size of the program carried out.

- Portability

That the program can run on different platforms.

- Efficiency

That the program is efficient in its use, this is described in that the program does not become slow or prone to closures for the user.

- Reliability

That the program is not prone to errors, slowness and sudden closures.

- Usability

That the program is easy to use and understand, maintaining an interface with a simple design so that the administrator and the client can use the program without much difficulty.

Implementation of packages for easy choice of services oriented to the type of event.

4. Attachments

4.1 Use cases

Name	Perform Warehouse Transfer Transaction
Author	Alejandro Bojorquez
Date	20/09/24
Description	The warehouse manager performs a transaction to move materials between warehouses.
Actors	Warehouse Manager
Preconditions	The manager must be logged in and have access to the transactions module.
Normal Flow	1. The manager logs in.
	2. Navigates to the warehouse transfer module.
	3. Selects material and destination warehouse.
	4. Confirms the transaction.
	5. The system records the transfer.
Alternative Flow	If there is insufficient material, the system displays an error and does not allow the transaction.
	The manager adjusts quantities and repeats the process.
Postconditions	The material is transferred between warehouses and the transaction is recorded.

Name	View Material Orders
Author	Elihu Moreno
Date	20/09/24
Description	The warehouse manager or supervisor reviews pending or completed material orders.
Actors	Purchasing Manager, Supervisor
Preconditions	The manager or supervisor must be logged in and have access to the orders module.
Normal Flow	1. The manager or supervisor logs in.
	2. Navigates to the orders module.
	3. The system displays a list of orders.
	4. They can view the details of each order.
Alternative Flow	If there are no orders, the system displays a message stating that no orders are available.
Postconditions	The manager or supervisor reviews the material orders.

Name	Change Warehouse Information
Author	Jese Perez
Date	20/09/24
Description	The administrator or warehouse manager changes the category of a warehouse to reflect a change in its use.
Actors	Administrator, Warehouse Manager
Preconditions	The administrator or manager must be logged in and have access to the warehouse configuration module.
Normal Flow	1. The administrator logs in.
	2. Navigates to the configuration module.
	3. Selects the warehouse and changes its information.
	4. Confirms the change.
Alternative Flow	If the category is not defined, the system prompts to select a category.
	The manager selects the new category and confirms.
Postconditions	The warehouse category is updated.

Name	Create a New Warehouse
Author	Antonio Dominguez
Date	20/09/24
Description	The administrator or manager creates a new warehouse in the system, specifying its location and details.
Actors	Administrator, Warehouse Manager
Preconditions	The administrator or manager must be logged in and have access to the warehouse creation module.
Normal Flow	1. The administrator logs in.
	2. Navigates to the warehouse creation module.
	3. Completes the form with warehouse details.
	4. Confirms the creation.
Alternative Flow	If any mandatory field is empty, the system displays an error and prompts to complete the information.
Postconditions	The new warehouse is registered in the system.

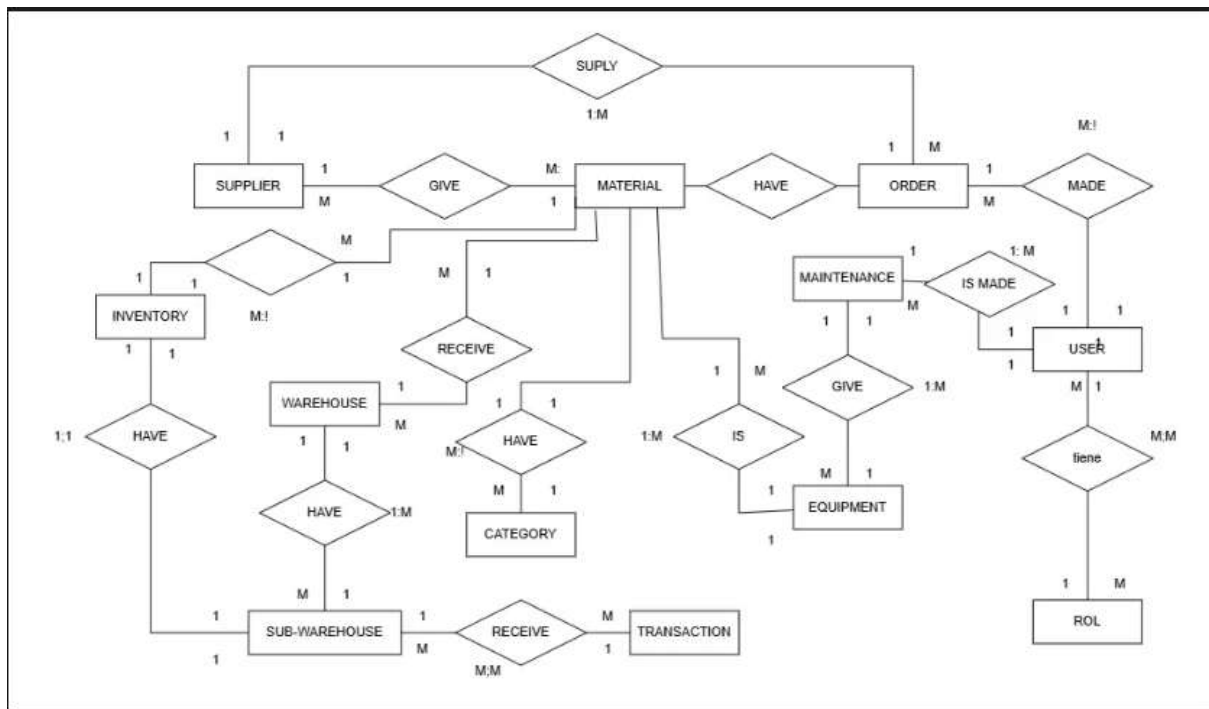
Name	View Warehouse Transaction History
Author	Elihu Moreno
Date	20/09/24
Description	The warehouse manager or supervisor consults the transaction history of a warehouse.
Actors	Warehouse Manager, Supervisor
Preconditions	The manager or supervisor must be logged in and have access to the transaction history module.
Normal Flow	1. The manager or supervisor logs in.
	2. Navigates to the transaction history module.
	3. Selects the warehouse and date range.
	4. The system displays the transaction history.
Alternative Flow	If there are no transactions, the system displays a message stating that no transactions are recorded.
Postconditions	The manager or supervisor reviews the transaction history.

Name	Register New Material
Author	Alejandro Bojorquez
Date	20/09/24
Description	The warehouse manager registers a new material in the system, specifying its properties.
Actors	Warehouse Manager
Preconditions	The manager must be logged in and have access to the material registration module.
Normal Flow	1. The manager logs in.
	2. Navigates to the material registration module.
	3. Completes the form with material details.
	4. Confirms and submits the form.
Alternative Flow	If any required field is left empty, the system requests that the manager provide the missing information.
Postconditions	The new material is registered in the system and available for management.

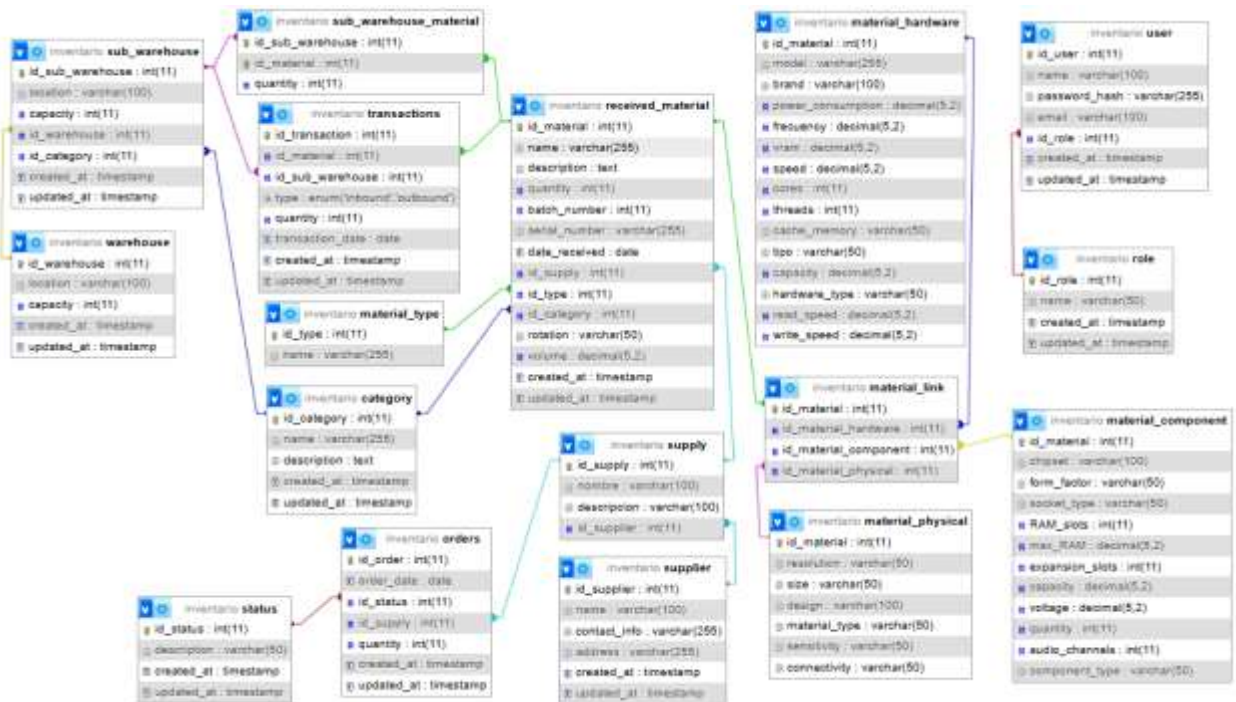
Name	View Warehouse Stock
Author	Antonio Dominguez
Date	20/09/24
Description	The warehouse manager or supervisor reviews the current stock levels in each warehouse.
Actors	Warehouse Manager, Supervisor
Preconditions	The manager or supervisor must be logged in and have access to the stock management module.
Normal Flow	1. The manager logs in.
	2. Navigates to the stock management module.
	3. The system displays the stock levels of all warehouses.
	4. The manager can filter by warehouse or material.
Alternative Flow	If the stock levels are unavailable, the system notifies the manager to update the stock information.
Postconditions	The manager or supervisor reviews the stock levels.

Name	Create a New Order
Author	Jese Perez
Date	20/09/24
Description	The purchasing manager creates a new material order to restock or acquire new items.
Actors	Purchasing Manager
Preconditions	The purchasing manager must be logged in and have access to the order creation module.
Normal Flow	1. The purchasing manager logs in.
	2. Navigates to the order creation module.
	3. Completes the form with material and supplier details.
	4. Confirms and submits the order.
Alternative Flow	If any fields are left empty, the system prompts the manager to complete the information.
Postconditions	The order is created and sent to the supplier.

4.2 ENTITY RELATION DIAGRAM



4.3 RELATIONAL MODEL



5.0 Coding

Login Module

The login module is responsible for ensuring secure access to the warehouse management system by authenticating users before granting entry. It provides an interface where users input their credentials, such as a username and password. The authentication logic verifies these credentials against the database. The login uses php files that pull the information from the database and verifies the credentials



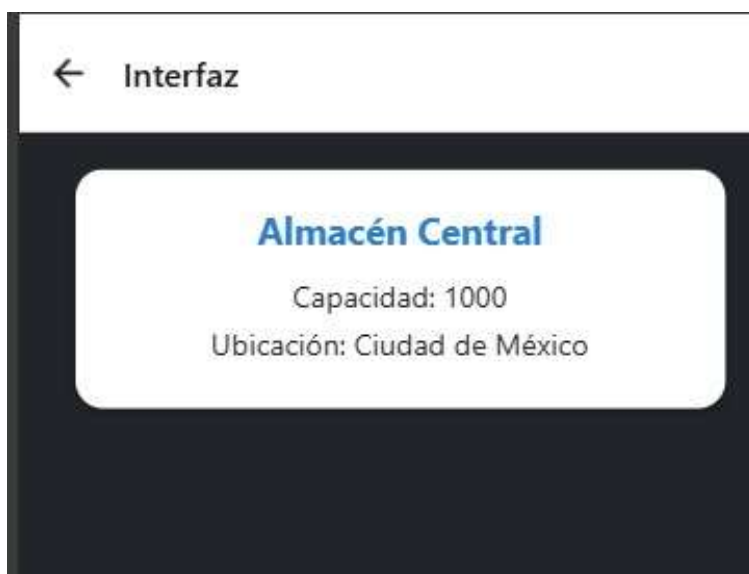
The image shows a web interface for a login module. It features a dark header bar at the top. Below the header, the word "Login" is displayed in a dark font. The main content area has a light gray background. In the center, the text "Inicio de Sesión" is prominently displayed in a bold, black font. Below this, there are two input fields: the first is labeled "Usuario" and the second is labeled "Contraseña". Both fields are white with a light gray border. Below the input fields, there are two blue buttons with white text. The top button is labeled "INICIAR SESIÓN" and the bottom button is labeled "IR A DASHBOARD". Below the buttons, there is a red error message that reads "Credenciales incorrectas."

```
// Constructor
public function __construct($id_user = null, $name = null, $email = null, $id_role = null)
{
    $this->id_user = $id_user;
    $this->name = $name;
    $this->email = $email;
    $this->id_role = $id_role;
}

private static $loginQuery = "
SELECT
u.id_user,
u.name,
u.password_hash,
u.email,
r.name as role_name
FROM user as u
INNER JOIN `role` as r on r.id_role = u.id_role
WHERE u.name = ?
";
```

Warehouse Interface

The warehouse interface acts as the central hub for managing materials and sub-warehouses, offering a structured overview of all warehouses within the system. Users can view detailed information about each warehouse, including its name, location, and storage capacity. The interface enables seamless navigation to specific warehouse details or associated sub-warehouses. Php files are used as API's to pull data from the database and then show it to the user



```

<?php
require_once __DIR__ . '/../config/conection.php';
require_once '../modals/Warehouse.php';

header("Access-Control-Allow-Origin: *");
header("Access-Control-Allow-Methods: GET, POST, PUT, DELETE");
header("Content-Type: application/json");

$method = $_SERVER['REQUEST_METHOD'];

switch ($method) {
    case 'GET': // Obtener warehouse
        $warehouses = Warehouse::getWarehouse();
        if (is_array($warehouses)) {
            echo json_encode($warehouses);
        } else {
            echo json_encode(["error" => $warehouses]); // Si hay un error
        }
        break;
    default:
        echo json_encode(["error" => "Invalid request method"]);
}
?>

```

Sub-Warehouse Materials

This module focuses on the inventory stored within sub-warehouses, offering a detailed view of materials and their attributes. Users can track materials ensuring accurate stock monitoring.



```

header("Access-Control-Allow-Origin: *"); // Permite peticiones desde cualquier origen
header("Access-Control-Allow-Methods: GET, POST, PUT, DELETE"); // Permite diferentes métodos
header("Content-Type: application/json");

$method = $_SERVER['REQUEST_METHOD']; // Se lee el método para llamar diferentes datos dependiendo del método

switch ($method) {
    case 'GET': // Obtener subalmacenes
        // Verifica si se pasa un parámetro 'id'
        $id_warehouse = isset($_GET['id']) ? intval($_GET['id']) : null;

        if ($id_warehouse) {
            // Llama al método para obtener subalmacenes filtrados por 'id_warehouse'
            $SubWarehouse = SubWarehouse::getSubWarehousesByWarehouseId($id_warehouse);
        } else {
            // Si no se pasa 'id', devuelve todos los subalmacenes
            $SubWarehouse = SubWarehouse::getSubWarehouses();
        }

        if (is_array($SubWarehouse)) {
            echo json_encode($SubWarehouse, true);
        } else {
            echo json_encode(["error" => "No se encontraron subalmacenes"]);
        }
        break;

    default:
        echo json_encode(["error" => "Invalid request method"]);
}

```

Sub-Warehouse Details

The sub-warehouse details section provides a comprehensive summary of an individual sub-warehouse. It displays essential information such as its name, storage capacity, and assigned materials.



```

public static function getSubWarehouses(){
    $connection = Conexion::get_connection();
    if ($connection->connect_error) {
        return "Error en la conexión: " . $connection->connect_error;
    }

    $query = "SELECT id_sub_warehouse, location, capacity, id_warehouse, id_category, c
    $command = $connection->prepare($query);
    $command->execute();
    $command->bind_result(
        $id_sub_warehouse,
        $location,
        $capacity,
        $id_warehouse,
        $id_category,
        $created_at,
        $updated_at
    );

    $subWarehouses = [];
    while ($command->fetch()) {
        $subWarehouses[] = [
            "id_sub_warehouse" => $id_sub_warehouse,
            "location" => $location,
            "capacity" => $capacity,
            "id_warehouse" => $id_warehouse,
            "id_category" => $id_category,
            "created_at" => $created_at,
            "updated_at" => $updated_at
        ];
    }

    return $subWarehouses;
}

```


Sub-Warehouse Interface

The sub-warehouse interface facilitates the management of smaller storage locations within a primary warehouse. It provides an organized display of all sub-warehouses under a specific warehouse, allowing users to easily access relevant details



```

const getWarehouses = async () => {
  try {
    const response = await fetch(API_warehouse, {
      method: 'GET'
    });
    const data = await response.json();
    console.log("Almacenes ",data);
    setWarehouses(data);

    // Toma de 2 en 2
    const groupedData = [];
    for (let i = 0; i < data.length; i += 2) {
      groupedData.push(data.slice(i, i + 2));
    }

    setGroupedWarehouses(groupedData);
  } catch (error) {
    console.error('Error obteniendo almacenes:', error);
  }
};

```

Warehouse Details

The warehouse details module presents in-depth information about a specific warehouse. It includes general warehouse data



```

public static function getWarehouse(){
    $connection = Conexion::get_connection();
    if ($connection->connect_error) {
        return "Error en la conexión: " . $connection->connect_error;
    }

    $query = "SELECT id_warehouse, name, location, capacity, created_at, updated_at";
    $command = $connection->prepare($query);
    $command->execute();
    $command->bind_result(
        $id_warehouse,
        $name,
        $location,
        $capacity,
        $created_at,
        $updated_at
    );

    // Para poder llamar los datos con PHP desde la base de datos principal
    // mande de esta forma. Se tiene que usar formato key => value por ejemplo
    $warehouses = [];
    while ($command->fetch()) {
        $warehouses[] = [
            "id_warehouse" => $id_warehouse,
            "name" => $name,
            "location" => $location,
            "capacity" => $capacity,
            "created_at" => $created_at,
            "updated_at" => $updated_at
        ];
    }

    return $warehouses;
}

```

You, 21 hours ago • TODOS LOS GETS