MANAGEMENT SYSTEM FOR DRUGSTORES MEDICAL PRODUCT DISTRIBUTORS

|  |  |
| --- | --- |
| Nombre el Proyecto | DROUGS MANAGEMENT |
| Fecha | 2023 SEPTIEMBRE |
| Versión | 2023SEP01.1 |
| Creado Por: | MAURICIO RUEDA CARO |

1. INTRODUCTION
   1. The Scope of the Product

Fusing sales, inventory management, and operations into one centralized ecosystem brings clarity and streamlines business processes.

* 1. Cost

Estimated value at $72,000,000 COP estimated reduction in management costs,

* 1. Public Objective

Administrators of stores and personal purchasing, logistics and inventories

* 1. Intended Use

Centralized management of inventories, costs and product rotations in the stores

* 1. General Drescription

The purpose of the item and ticket management system in the pharmacy is to automate and simplify sales management and inventory control in the establishment. This system allows employees to record items purchased by customers on a purchase ticket, calculate subtotal, Value Added Tax (VAT) and total purchase, and keep inventory of items in inventory up to date. In addition, it facilitates data recovery and reporting for sales analysis and inventory control.

1. REQUISITOS FUNCIONALES
   1. Gestión de Artículos:
      1. Registrar nuevos artículos en el inventario.
      2. Actualizar la información de los artículos existentes.
      3. Eliminar artículos del inventario.
      4. Consultar detalles de artículos, incluyendo nombre, precio, existencias, etc.
   2. Gestión de Tickets
      1. Crear nuevos tickets de compra.
      2. Agregar artículos a un ticket.
      3. Eliminar artículos de un ticket.
      4. Calcular automáticamente el subtotal de un ticket.
      5. Aplicar el IVA al subtotal y calcular el total de la compra.
   3. Control de Inventario.
      1. Actualizar las existencias de artículos automáticamente al vender productos.
      2. Generar alertas o notificaciones cuando las existencias de un artículo estén por debajo de un umbral mínimo.
   4. Acceso de Usuarios.
      1. Autenticación de usuarios (empleados).
      2. Roles y permisos para gestionar quién puede realizar ciertas acciones.
2. REQUISITOS NO FUNCIONALES.
   1. Rendimiento:

El sistema debe ser eficiente y garantizar una respuesta rápida a las operaciones de registro y consulta.

* 1. Seguridad:

Debe implementar medidas de seguridad, como la autenticación y autorización de usuarios, para proteger la información confidencial.

* 1. Escalabilidad:

Debe ser escalable para manejar un crecimiento en el número de artículos y tickets.

* 1. Usabilidad:

La interfaz de usuario debe ser intuitiva y fácil de usar para los empleados.

1. POSIBLES MEJORAS FUTURAS.

Gestión de Clientes: Agregar la capacidad de registrar información de clientes, lo que permitiría ofrecer programas de lealtad y análisis de comportamiento de compra.

Pagos Electrónicos: Integrar métodos de pago electrónicos, como tarjetas de crédito/débito o billeteras digitales.

Generación de Reportes: Implementar la generación automática de informes de ventas, inventario y rendimiento.

Optimización de Inventario: Implementar algoritmos de optimización de inventario para evitar la falta de existencias o el exceso de stock.

Interfaz Móvil: Desarrollar una aplicación móvil para permitir a los empleados realizar operaciones desde dispositivos móviles.

Integración con Proveedores: Integrar el sistema con proveedores para realizar pedidos de reposición de inventario de manera automática.

Pruebas Automatizadas: Implementar pruebas automatizadas para garantizar la calidad del software y la detección temprana de errores.

1. DATABASE.
   1. Tables (SQL)

-- Crear la tabla de Medicamentos

CREATE TABLE medicamentos (

id serial PRIMARY KEY,

nombre VARCHAR(255) NOT NULL,

stock INT NOT NULL,

precio DECIMAL(10, 2) NOT NULL,

impuesto DECIMAL(5, 2) NOT NULL

);

-- Crear la tabla de Clientes (si necesaria)

CREATE TABLE clientes (

id serial PRIMARY KEY,

nombre VARCHAR(255) NOT NULL,

direccion TEXT,

telefono VARCHAR(20)

);

-- Crear la tabla de Ventas

CREATE TABLE ventas (

id serial PRIMARY KEY,

cliente\_id INT, -- Esta columna se relaciona con la tabla "clientes" si es aplicable

fecha TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (cliente\_id) REFERENCES clientes(id),

CONSTRAINT ventas\_fecha\_check CHECK (fecha <= CURRENT\_TIMESTAMP)

);

-- Crear la tabla de Detalles de Ventas

CREATE TABLE venta\_detalles (

id serial PRIMARY KEY,

venta\_id INT,

medicamento\_id INT,

cantidad INT NOT NULL,

FOREIGN KEY (venta\_id) REFERENCES ventas(id),

FOREIGN KEY (medicamento\_id) REFERENCES medicamentos(id),

CONSTRAINT cantidad\_mayor\_cero CHECK (cantidad **> 0)**

**);**