A diagram of a product management system

Description automatically generated

## **Proyecto MPOO**

**Docente: Ing. José Alejandro Duran García**

**Integrantes del grupo:** Stefanny Contreras

Arles González

Margel Castro

Álvaro Fornos

Logotipo, nombre de la empresa

Descripción generada automáticamente

Sistema de inventario sencillo y estructurado

**Link del repositorio:** [**https://github.com/arlesjoel/Proyecto\_Final\_POO.git**](https://github.com/arlesjoel/Proyecto_Final_POO.git)

Análisis de Requisitos

Requerimientos Funcionales

**1. Gestión de Productos:**

- Operaciones: Agregar, modificar, eliminar.

- Atributos: Código único, nombre, descripción, categoría, subcategoría, nivel mínimo y máximo de inventario.

**2. Control de Stock:**

- Registrar entradas y salidas.

- Asociar transacciones a usuarios.

- Registrar fecha y hora.

- Calcular stock actual.

**3. Gestión de Usuarios:**

- Operaciones: Crear, modificar, eliminar.

- Atributos: Nombre, correo electrónico, contraseña (segura), rol.

**4. Búsqueda de Productos:**

- Por código, nombre, categoría, proveedor.

Requerimientos No Funcionales

**1. Desempeño:**

- Respuesta rápida a consultas.

- Capacidad para manejar grandes volúmenes de datos.

**2. Seguridad:**

- Autenticación de usuarios.

- Respaldo regular de datos.

**3. Disponibilidad:**

- Operatividad 24/7.

**4. Usabilidad:**

- Interfaz intuitiva.

**5. Escalabilidad:**

- Adaptabilidad a crecimiento futuro.

**6. Confiabilidad:**

- Resistencia a fallos.

Diseño y Normalización de la Base de Datos

**Esquema de Tablas**

1. Tabla `Productos`:

- `id\_producto` (INT, PRIMARY KEY, AUTO\_INCREMENT)

- `codigo` (VARCHAR, UNIQUE)

- `nombre` (VARCHAR)

- `descripcion` (TEXT)

- `categoria` (VARCHAR)

- `subcategoria` (VARCHAR)

- `nivel\_minimo` (INT)

- `nivel\_maximo` (INT)

2. Tabla `Stock`:

- `id\_transaccion` (INT, PRIMARY KEY, AUTO\_INCREMENT)

- `id\_producto` (INT, FOREIGN KEY)

- `cantidad` (INT)

- `tipo` (ENUM: 'entrada', 'salida')

- `id\_usuario` (INT, FOREIGN KEY)

- `fecha\_hora` (DATETIME)

3. Tabla `Usuarios`:

- `id\_usuario` (INT, PRIMARY KEY, AUTO\_INCREMENT)

- `nombre` (VARCHAR)

- `correo` (VARCHAR, UNIQUE)

- `contrasena` (VARCHAR)

- `rol` (ENUM: 'administrador', 'encargado')

4. Tabla `Categorias`:

- `id\_categoria` (INT, PRIMARY KEY, AUTO\_INCREMENT)

- `nombre` (VARCHAR)

5. Tabla `Subcategorias`:

- `id\_subcategoria` (INT, PRIMARY KEY, AUTO\_INCREMENT)

- `nombre` (VARCHAR)

- `id\_categoria` (INT, FOREIGN KEY)

6. Tabla `Proveedores`:

- `id\_proveedor` (INT, PRIMARY KEY, AUTO\_INCREMENT)

- `nombre` (VARCHAR)

- `contacto` (VARCHAR)

**Proceso de Normalización**

1. Primera Forma Normal (1NF):

- Asegurar que todas las tablas tienen columnas atómicas y no repetitivas.

2. Segunda Forma Normal (2NF):

- Asegurar que cada tabla depende completamente de la clave primaria.

3. Tercera Forma Normal (3NF):

- Asegurar que no hay dependencias transitivas.

**Código SQL para MySQL**

**Creación de Tablas en MySQL**

-- Crear base de datos

CREATE DATABASE inventarioexamen;

USE InventarioDB;

-- Tabla de Categorías

CREATE TABLE Categorias (

id\_categoria INT PRIMARY KEY IDENTITY,

nombre VARCHAR(50) NOT NULL

);

-- Tabla de Subcategorías

CREATE TABLE Subcategorias (

id\_subcategoria INT PRIMARY KEY IDENTITY,

nombre VARCHAR(50) NOT NULL,

id\_categoria INT,

FOREIGN KEY (id\_categoria) REFERENCES Categorias(id\_categoria)

);

-- Tabla de Proveedores

CREATE TABLE Proveedores (

id\_proveedor INT PRIMARY KEY IDENTITY,

nombre VARCHAR(100) NOT NULL,

contacto VARCHAR(100)

);

-- Tabla de Productos

CREATE TABLE Productos (

id\_producto INT PRIMARY KEY IDENTITY,

codigo VARCHAR(50) UNIQUE NOT NULL,

nombre VARCHAR(100) NOT NULL,

descripcion TEXT,

id\_categoria INT,

id\_subcategoria INT,

nivel\_minimo INT,

nivel\_maximo INT,

FOREIGN KEY (id\_categoria) REFERENCES Categorias(id\_categoria),

FOREIGN KEY (id\_subcategoria) REFERENCES Subcategorias(id\_subcategoria)

);

-- Tabla de Usuarios

CREATE TABLE Usuarios (

id\_usuario INT PRIMARY KEY IDENTITY,

nombre VARCHAR(100) NOT NULL,

correo VARCHAR(100) UNIQUE NOT NULL,

contrasena VARCHAR(255) NOT NULL,

rol varcharv (25) not null

);

-- Tabla de Stock

CREATE TABLE Stock (

id\_transaccion INT PRIMARY KEY IDENTITY,

id\_producto INT,

cantidad INT NOT NULL,

tipo varchar(25) not null,

id\_usuario INT,

fecha\_hora DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (id\_producto) REFERENCES Productos(id\_producto),

FOREIGN KEY (id\_usuario) REFERENCES Usuarios(id\_usuario)

);

**Código SQL para SQL Server**

**Creación de Tablas en SQL Server**

-- Crear base de datos

CREATE DATABASE InventarioDB;

GO

USE InventarioDB;

GO

-- Tabla de Categorías

CREATE TABLE Categorias (

id\_categoria INT PRIMARY KEY IDENTITY(1,1),

nombre VARCHAR(50) NOT NULL

);

-- Tabla de Subcategorías

CREATE TABLE Subcategorias (

id\_subcategoria INT PRIMARY KEY IDENTITY(1,1),

nombre VARCHAR(50) NOT NULL,

id\_categoria INT,

FOREIGN KEY (id\_categoria) REFERENCES Categorias(id\_categoria)

);

-- Tabla de Proveedores

CREATE TABLE Proveedores (

id\_proveedor INT PRIMARY KEY IDENTITY(1,1),

nombre VARCHAR(100) NOT NULL,

contacto VARCHAR(100)

);

-- Tabla de Productos

CREATE TABLE Productos (

id\_producto INT PRIMARY KEY IDENTITY(1,1),

codigo VARCHAR(50) UNIQUE NOT NULL,

nombre VARCHAR(100) NOT NULL,

descripcion TEXT,

id\_categoria INT,

id\_subcategoria INT,

nivel\_minimo INT,

nivel\_maximo INT,

FOREIGN KEY (id\_categoria) REFERENCES Categorias(id\_categoria),

FOREIGN KEY (id\_subcategoria) REFERENCES Subcategorias(id\_subcategoria)

);

-- Tabla de Usuarios

CREATE TABLE Usuarios (

id\_usuario INT PRIMARY KEY IDENTITY(1,1),

nombre VARCHAR(100) NOT NULL,

correo VARCHAR(100) UNIQUE NOT NULL,

contrasena VARCHAR(255) NOT NULL,

rol VARCHAR(50) CHECK (rol IN ('administrador', 'encargado'))

);

-- Tabla de Stock

CREATE TABLE Stock (

id\_transaccion INT PRIMARY KEY IDENTITY(1,1),

id\_producto INT,

cantidad INT NOT NULL,

tipo VARCHAR(10) CHECK (tipo IN ('entrada', 'salida')),

id\_usuario INT,

fecha\_hora DATETIME DEFAULT GETDATE(),

FOREIGN KEY (id\_producto) REFERENCES Productos(id\_producto),

FOREIGN KEY (id\_usuario) REFERENCES Usuarios(id\_usuario)

);

Descripción del Proceso

- Creación de Base de Datos: Creamos una base de datos llamada `InventarioDB` que contiene todas las tablas necesarias.

- Tablas y Relaciones: Definimos tablas para productos, categorías, subcategorías, usuarios, stock, y proveedores. Establecemos relaciones entre ellas usando claves foráneas.

- Columnas y Restricciones: Añadimos restricciones de unicidad, claves primarias, y comprobaciones para garantizar la integridad de los datos.

- Normalización: La estructura está diseñada para cumplir con la tercera forma normal (3NF), eliminando redundancias y asegurando la integridad referencial.

Este diseño proporciona una base sólida para implementar un sistema de inventario eficiente y escalable. Si tienes preguntas o necesitas más detalles, no dudes en preguntar.

**BD Creadas:**

**En MySQL:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**En SQL:**

A computer screen shot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated