Proyecto: Sistema de Gestión de Stock (SGS)

Avance 3

Nombre: Mauro Zuppone

Documento: 34.087.437

Mail: [mauro.zuppone@gmail.com](mailto:mauro.zuppone@gmail.com)

Carrera: Tec. Univ. en Tecnología de la Información

Fecha de entrega: 18/11/2024

Materia: Taller de Proyectos II

**Selección del proyecto:**

El proyecto elegido es un sistema de gestión de stock. Abarca todos los aspectos del ciclo de vida del desarrollo de software y tiene la suficiente complejidad como para incluir análisis de requerimientos, implementación y la evaluación de funcionalidad.

Es una herramienta muy útil en cualquier empresa que requiera controlar su inventario, lo que lo hace un proyecto relevante y aplicable.

Dado que el software debe gestionar entradas y salidas de productos, controlar nivel de stock y generar reportes, abarca varias funcionalidades que permiten aplicar lo aprendido en el ciclo de vida del desarrollo de software.

**Análisis y requerimientos:**

**Requerimientos funcionales:**

* Ingreso de productos: el sistema debe permitir la carga de nuevos productos al stock con información como nombre, código, cantidad, proveedor y fecha de ingreso.
* Salida de productos: permitir registrar la salida de productos, reduciendo automáticamente las cantidades del stock.
* Control de stock: el sistema debe mostrar los niveles actuales de stock de cada producto, alertando cuando el nivel esté bajo un límite predeterminado.
* Reportes: generar reportes periódicos sobre el estado del stock, las entradas y salidas de productos.
* Búsqueda y filtrado: el sistema debe permitir buscar productos por nombre, código o categoría, y filtrar por cantidad, proveedor, entre otros.

**Requerimientos no funcionales:**

* Seguridad: solo usuarios autorizados pueden acceder al sistema y realizar modificaciones en el stock.
* Escalabilidad: el sistema debe poder manejar el aumento en la cantidad de productos y usuarios.
* Usabilidad: debe ser fácil de usar, con una interfaz intuitiva para minimizar los errores en la gestión de productos.
* Rendimiento: el tiempo de respuesta debe ser rápido, incluso cuando se manejen grandes cantidades de productos.

**Presentación del proyecto:**

1. **Problema a resolver:**

Una empresa que maneja un inventario de productos necesita gestionar de forma eficiente su stock para evitar faltantes o excesos. Sin un sistema de gestión de stock, se debe realizar de forma manual, lo que conlleva a errores y pérdidas de tiempo. Este sistema automatizará el proceso, mejorando la precisión y reduciendo los tiempos de operación.

Algunos de los problemas provocados por manejos de inventarios de forma manual o con herramientas ineficaces:

* Errores humanos en el registro de entradas y salidas.
* Pérdida de control sobre el stock disponible.
* Falta de alertas preventivas.
* Dificultad para generar informes
* Bajo nivel de integración de información entre sectores retrasando la comunicación.

Estos problemas derivan en ineficiencia operativa causando pérdidas económicas. LA solución propuesta es un sistema de gestión de stock digitalizado que centralice el control y seguimiento de todos los productos, ayudando a optimizar recursos y mejorar la toma de decisiones.

1. **Requisitos:**

El sistema debe permitir:

1. Ingreso y salida de productos del stock.
2. Control de stock con alertas por niveles bajos.
3. Generación de reportes.
4. Seguridad y accesibilidad.
5. **Ciclo de vida:**

Se seguirá un ciclo de vida iterativo, con etapas bien definidas para asegurar el éxito del proyecto:

1. **Análisis de requerimientos:** reunir toda la información sobre las necesidades del cliente y los procesos actuales de la empresa. A partir de ello entender en profundidad lo que el cliente necesita.
2. **Diseño del sistema:** se planificará la arquitectura general del sistema. Se diseñará la base de datos que contendrá la información del inventario y se desarrollarán los diagramas de flujo que representen cómo los productos entran y salen del sistema.
3. **Implementación y codificación:** desarrollo del código del sistema según los diseños establecidos. Se crearán las interfaces para que los usuarios puedan interactuar con el sistema, así como las funcionalidades para gestionar productos y generar reportes.
4. **Pruebas**: realizar pruebas unitarias, de integración y funcionales para asegurar que el sistema cumple con los requerimientos.
5. **Despliegue:** una vez que el sistema esté probado, se implementará en el entorno de producción de la empresa. Se capacitará al personal para que pueda utilizar la herramienta de forma eficaz.
6. **Mantenimiento:** luego del despliegue, se continuará con la fase de mantenimiento, en donde se corregirán los errores encontrados, se realizarán ajustes menores y/o se agregarán nuevas funcionalidades según las necesidades emergentes de la empresa.
7. **Análisis, diagrama y requerimientos:**

Requerimientos funcionales:

* Ingreso de productos: el sistema debe permitir la carga de nuevos productos al stock con información como nombre, código, cantidad, proveedor y fecha de ingreso.
* Salida de productos: permitir registrar la salida de productos, reduciendo automáticamente las cantidades del stock.
* Control de stock: el sistema debe mostrar los niveles actuales de stock de cada producto, alertando cuando el nivel esté bajo un límite predeterminado.
* Reportes: generar reportes periódicos sobre el estado del stock, las entradas y salidas de productos.
* Búsqueda y filtrado: el sistema debe permitir buscar productos por nombre, código o categoría, y filtrar por cantidad, proveedor, entre otros.

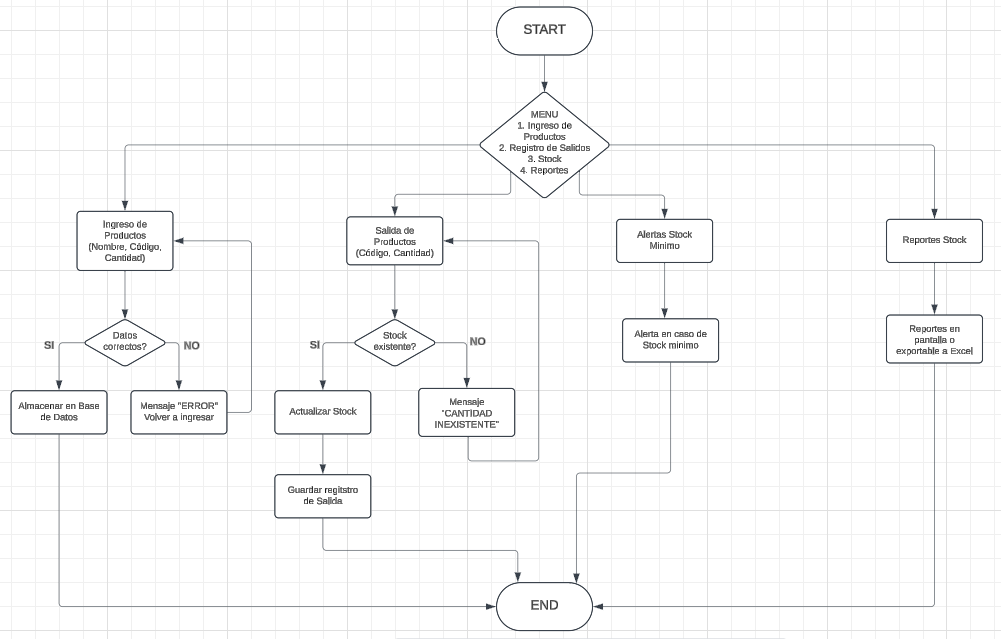
Requerimientos no funcionales:

* Seguridad: solo usuarios autorizados pueden acceder al sistema y realizar modificaciones en el stock.
* Escalabilidad: el sistema debe poder manejar el aumento en la cantidad de productos y usuarios.
* Usabilidad: debe ser fácil de usar, con una interfaz intuitiva para minimizar los errores en la gestión de productos.
* Rendimiento: el tiempo de respuesta debe ser rápido, incluso cuando se manejen grandes cantidades de productos.

Diagrama:

A continuación se presenta en el siguiente link el diseño relacionado con el flujo. También se adjunta captura del mismo:

<https://lucid.app/lucidchart/ee5f8505-7b22-4e84-a335-5509197076aa/edit?invitationId=inv_9fb51751-0a2f-44bf-b302-b3a3372f5b44>



1. **Procedimiento para la entrada de datos:**
2. **Pantalla de Inicio de Sesión**

El usuario ingresa sus credenciales (nombre de usuario y contraseña).

Una vez autenticado, el usuario accede al menú principal del sistema.

1. **Pantalla de Ingreso de Producto al Stock**

**Formulario de Ingreso de Producto:**

Campos a completar:

* Nombre del producto: identificación del producto.
* Código del producto: código único de identificación del producto.
* Cantidad inicial: número que indica la cantidad de unidades que ingresarán al stock.
* Proveedor: nombre o código del proveedor del producto.
* Categoría: tipo de categoría a la que pertenece el producto

**Botón de Guardar:**

Cuando el usuario hace clic en "Guardar", se ejecuta una validación de los datos ingresados.

Validaciones:

* Campos obligatorios (nombre, código, y cantidad) deben estar completos.
* El campo "Cantidad inicial" debe ser un valor numérico positivo.
* El campo "Código de producto" no puede duplicarse en el sistema.

**Confirmación de datos:**

* Si los datos son correctos, el sistema almacena el producto en la base de datos y muestra un mensaje de confirmación: "Producto ingresado exitosamente".
* Si alguno de los datos no cumple con los criterios de validación, el sistema muestra un mensaje de error “Cantidad inválida”) y permite editar/corregir el dato.

1. **Pantalla de Registro de Salida de Producto del Stock:**

**Formulario de registro de salida:**

Campos a completar:

* Código del producto: producto que se retira del stock.
* Cantidad de salida: Número de unidades que se retiran.

**Botón de Actualizar Stock:**

Valida que el stock sea suficiente para cubrir la cantidad ingresada.

* Validaciones:
* La cantidad debe ser un valor positivo y menor o igual al stock actual.
* Si el stock no es suficiente, se muestra un mensaje de alerta.

**Confirmación de salida:**

Si la transacción es válida, se guarda el registro y el sistema actualiza el stock.

1. **Diseño de la Interfaz:**
2. **Menú Principal**

Ubicación: Barra de navegación lateral.

Opciones:

Inicio: muestra resumen general del sistema, incluyendo notificaciones de stock bajo.

Ingreso de Producto: redirige al formulario para ingresar productos.

Salida de Producto: redirige al formulario para registrar salidas del inventario.

Verificación de Stock: muestra stock actual de cada producto.

Reportes: generación de reportes.

1. **Pantalla de Ingreso de Producto**

Diseño del Formulario:

* Campos de entrada alineados verticalmente para facilitar la lectura y el ingreso.
* Validación de campos activa, con mensajes de error en rojo y descripciones breves junto a cada campo para evitar errores.

Botón de Guardado:

* Al pie del formulario boton de “Guardar” (claramente visible).

1. **Pantalla de Registro de Salida**

Formulario:

* Campo de búsqueda: para seleccionar el producto, utilizando autocompletado para facilitar la búsqueda por nombre o código.
* Campo para ingresar cantidad de salida: con una pequeña ventana emergente que muestre el stock actual del producto seleccionado.

Botón de Confirmación:

* Activo sólo si los datos cumplen con la validación.
* Mensaje de confirmación visible después de la actualización exitosa.

1. **Pantalla de Verificación de Stock**

Interfaz de Tabla: muestra todos los productos, su cantidad actual, el mínimo recomendado, y un icono de alerta si el stock es insuficiente.

Filtro: opciones de filtro para mostrar solo productos con bajo stock o productos de una categoría específica.

1. **Reportes**

Selector de Reporte:

* Menú desplegable para elegir el tipo de reporte (movimientos de productos, productos con bajo stock, resumen general).
* Botón de Generación de Reporte: muestra el reporte en pantalla y permite exportarlo a PDF o Excel.

1. **Estructura de la Base de Datos:**

Tabla 1: Productos

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo de dato** | **Descripción** |
| Id\_producto (PK) | INT, AUTO\_INCREMENT | Clave Primaria |
| Nombre\_producto | VARCHAR(20), NOT NULL |  |
| Codigo\_producto | VARCHAR(20), UNIQUE, NOT NULL |  |
| Categoría\_producto | VARCHAR(20), |  |
| Proveedor | VARCHAR(20), NOT NULL |  |
| Stock\_actual | INT, NOT NULL, DEFAULT 0 |  |
| Stock\_minimo | INT, NOT NULL, DEFAULT 0 |  |
| Fecha\_ingreso | DATE |  |
| Fecha\_actualización | DATE |  |

Tabla 2: Movimientos

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo de dato** | **Descripción** |
| Id\_movimiento (PK) | INT, AUTO\_INCREMENT | Clave Primaria |
| Id\_producto (FK) | INT NOT NULL | Clave Foránea |
| Tipo\_movimiento | ENUM ('entrada', 'salida', NOT NULL | Entrada o salida |
| cantidad | INT, NOT NULL | Cantidad unid |
| Fecha\_movimieto | DATE |  |
| usuario | VARCHAR(20) |  |

Tabla 3: Usuarios

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo de dato** | **Descripción** |
| Id\_usuario (PK) | INT, AUTO\_INCREMENT | Clave primaria |
| Nombre\_usuario | VARCHAR(20), UNIQUE, NOT NULL |  |
| contraseña | VARCHAR(20), NOT NULL |  |
| rol | ENUM ('admin', 'operat', NOT NULL |  |
| Fecha\_alta\_usuario | DATE |  |

Tabla 4: Proveedores

|  |  |  |
| --- | --- | --- |
| **Campo** | **Tipo de dato** | **Descripción** |
| Id\_proveedor (PK) | INT, AUTO\_INCREMENT | Clave Primaria |
| Nombre\_proveedor | VARCHAR (20), NOT NULL |  |
| Contacto | VARCHAR(30), NULL |  |
| Direcccion | VARCHAR(20) |  |

1. **Código SQL para la creación de la Base de Datos:**

CREATE DATABASE gestion\_stock;

USE gestion\_stock;

**Creación de tabla Productos**

CREATE TABLE productos (

id\_producto INT PRIMARY KEY AUTO\_INCREMENT,

nombre:producto VARCHAR(20) NOT NULL,

codigo\_producto VARCHAR(50) UNIQUE NOT NULL,

categoria\_producto VARCHAR(20),

proveedor VARCHAR(20) NOT NULL

stock\_actual INT NOT NULL DEFAULT 0,

stock\_minimo INT NOT NULL DEFAULT 0,

fecha\_ingreso DATE

fecha\_actualizacion DATE

);

**Creación de tabla Movimientos**

CREATE TABLE movimientos (

id\_movimiento INT PRIMARY KEY AUTO\_INCREMENT,

id\_producto INT NOT NULL,

tipo\_movimiento ENUM('entrada', 'salida') NOT NULL,

cantidad INT NOT NULL,

fecha\_movimiento DATE

usuario VARCHAR(20)

FOREIGN KEY (id\_producto)

);

**Creación de tabla Usuarios**

CREATE TABLE usuarios (

id\_usuario INT PRIMARY KEY AUTO\_INCREMENT,

nombre\_usuario VARCHAR(20) UNIQUE NOT NULL,

contraseña VARCHAR(20) NOT NULL,

rol ENUM('admin', 'operat') NOT NULL,

fecha\_alta\_usuario DATE

);

**Creación de tabla Proveedores**

CREATE TABLE proveedores (

id\_proveedor INT PRIMARY KEY AUTO\_INCREMENT,

nombre\_proveedor VARCHAR(20) NOT NULL,

contacto VARCHAR(30) NOT NULL

direccion VARCHAR(20)

);