

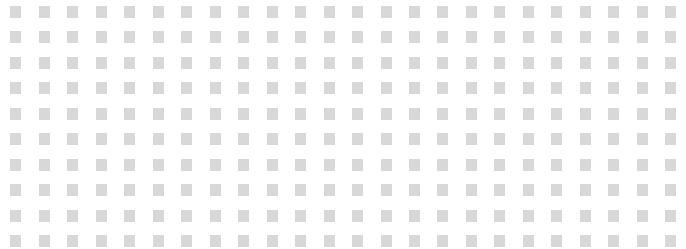


Gestor de inventarios

Ingeniería en Software

Desarrollado por:

Armando Ambrosio
Nocedal
Omar Arias Dominguez



Docente:

Maritza Flores
Dominguez



Contenido

Introducción.....	3
Planteamiento del problema.....	4
Objetivo general del proyecto.....	5
Objetivos específicos del proyecto	6
Hipótesis	7
Requisitos funcionales	8
Requisitos no funcionales	8
Alcances.....	9
Limitaciones	12
Marco Teórico	13
Referencias	18

Introducción

En un entorno empresarial, la gestión eficiente del inventario es esencial para garantizar operaciones rápidas y satisfacer las demandas de los clientes. Para brindar una solución efectiva, este proyecto se enfoca en desarrollar un administrador de inventario en Java. Diseñado para ayudar a las pequeñas empresas a administrar sus productos de manera eficiente, el sistema permite a los usuarios agregar nuevos productos, verificar los niveles de inventario y eliminar productos que ya no necesitan.

Los sistemas de gestión de inventario son muy importantes en los entornos empresariales por diversas razones. En primer lugar, las empresas pueden gestionar efectivamente el inventario y evitar pérdidas financieras causadas por la obsolescencia de los productos, el desabastecimiento o el exceso de existencias. Además, la visibilidad en tiempo real de la disponibilidad del producto facilita la planificación de adquisiciones y la gestión de la cadena de suministro.

Además, el registro de productos no solo ayuda a identificar y buscar productos específicos, sino que también es fundamental para realizar acciones como actualizar stock, eliminar productos obsoletos y tomar decisiones estratégicas relacionadas con la adquisición de nuevos productos.

Los sistemas de gestión de inventario también son cruciales para mejorar la satisfacción del cliente. Obteniendo información precisa sobre la disponibilidad del producto.

Planteamiento del problema

En el entorno empresarial actual, la gestión eficiente del inventario es un aspecto fundamental del éxito empresarial. Sin embargo, muchas organizaciones enfrentan importantes desafíos en este sentido debido a la falta de herramientas modernas y efectivas para gestionar adecuadamente el inventario. El proyecto de desarrollo "Gestor de Inventarios" surge en respuesta a los siguientes problemas identificados:

- Falta de automatización en la gestión de inventarios:

Muchas empresas todavía dependen de procesos manuales y hojas de cálculo para realizar un seguimiento de sus productos e inventario. Esto da como resultado una mayor probabilidad de error humano, retrasos en las actualizaciones del inventario y una falta de visibilidad en tiempo real del inventario disponible.

- Dificultades para identificar productos obsoletos:

La falta de herramientas automatizadas para identificar y gestionar productos obsoletos puede provocar la acumulación de inventario no vendible. Esto requiere una valiosa inversión de capital y espacio de almacenamiento.

- La eficiencia de actualización de la información del producto es baja:

Sin una solución implementada la actualización de los detalles del producto (como cambios en el precio, la descripción o los atributos) suele ser un proceso lento y propenso a errores.

- Riesgo de Pérdida de Clientes:

La gestión ineficiente de inventarios puede resultar en situaciones de escasez de productos, lo que a su vez puede llevar a la pérdida de clientes insatisfechos debido a la incapacidad de satisfacer sus demandas de manera oportuna.

Objetivo general del proyecto

El objetivo general de este proyecto es desarrollar un sistema de gestión de inventarios en Java que permita a los usuarios administrar eficientemente los productos, controlar los niveles de existencias, actualizar la información y gestionar de manera efectiva la eliminación de productos obsoletos. El sistema proporcionará una solución automatizada y fácil de usar para mejorar la precisión, la agilidad y la eficiencia en la administración de inventarios.

Objetivos específicos del proyecto

1. **Diseñar la Estructura del Sistema:** Definir la arquitectura, las clases y las relaciones necesarias para la representación de productos, inventarios y funcionalidades asociadas.
2. **Implementar una Interfaz de Usuario:** Desarrollar una interfaz de usuario intuitiva y amigable que permita a los usuarios acceder a las funciones del sistema de manera efectiva y que no tengan problemas con su uso.
3. **Implementar Funcionalidad de Agregar Productos:** Crear la función de agregar nuevos productos al inventario, permitiendo a los usuarios ingresar la información relevante del producto.
4. **Desarrollar la Función de Verificación de Existencias:** Implementar la función que permitirá a los usuarios verificar los niveles actuales de existencias de los productos en el inventario.
5. **Realizar Actualizaciones de Información:** Diseñar y programar métodos que permitan a los usuarios actualizar la cantidad, así como el retiro de productos en stock.
6. **Crear Función de Eliminación de Productos:** Implementar la capacidad de eliminar productos que ya no estén disponibles o que vayan a ser retirados.
7. **Implementar Búsqueda:** Desarrollar funciones de búsqueda que permitan a los usuarios encontrar productos por nombre.

Hipótesis

Se postula que, mediante la implementación de un Gestor de Inventarios en Java, será posible mejorar significativamente la eficiencia en la administración de inventarios. Al automatizar procesos que antes eran manuales como el seguimiento de existencias y la actualización de información de manera escrita y en continuas hojas de cálculo.

Requisitos funcionales

1. Registro de nuevos productos: Los usuarios podrán ingresar los detalles de un nuevo producto, incluyendo nombre, descripción, precio y cantidad inicial.
2. Consulta de niveles de existencias: Los usuarios podrán ver la cantidad actual de un producto específico en el inventario.
3. Actualización de información de productos: Se permitirá a los usuarios modificar la información de un producto, como cambiar su precio, descripción, etc.
4. Eliminación de productos: Los usuarios podrán eliminar productos que ya no están disponibles o que se vayan a retirar del inventario.

Requisitos no funcionales

1. Rendimiento: El sistema debe ser capaz de manejar un gran número de productos y usuarios de manera eficiente.
2. Usabilidad: La interfaz de usuario debe ser intuitiva y fácil de entender incluso para usuarios que no tienen conocimientos técnicos.
3. Escalabilidad: El sistema debe ser diseñado de manera que pueda escalarse fácilmente en caso de un aumento en la cantidad de productos o usuarios.
4. Mantenibilidad: El código debe estar bien estructurado y documentado para facilitar futuras actualizaciones y mejoras.

Alcances

1. Registro de Productos:

- Los usuarios podrán registrar nuevos productos en el sistema, ingresando información como nombre, descripción, precio y cantidad inicial en stock.

2. Visualización de Inventarios:

- Los usuarios podrán acceder y visualizar los inventarios existentes en el sistema.

3. Consulta de Niveles de Existencias:

- Los usuarios podrán consultar los niveles de existencias de productos específicos en el inventario.

4. Actualización de Información de Productos:

- Los usuarios tendrán la capacidad de actualizar la información de productos existentes, lo que incluye modificar su precio, descripción y cantidad en stock.

5. Eliminación de Productos:

- Se proporcionará a los usuarios la opción de eliminar productos que ya no estén disponibles o que se hayan vuelto obsoletos.

6. Búsqueda y Filtrado de Productos:

- Los usuarios podrán buscar productos por nombre.

7. Interfaz de Usuario Intuitiva:

- El sistema contará con una interfaz de usuario intuitiva y fácil de usar, diseñada para usuarios de diversos niveles de experiencia técnica.

8. Seguridad de Acceso:

- Se implementará un sistema de autenticación para garantizar que solo usuarios autorizados puedan acceder al sistema.

9. Rendimiento Eficiente:

- El sistema se diseñará y optimizará para manejar eficientemente un gran número de productos, evitando ralentizaciones significativas en las operaciones clave.

10. Escalabilidad:

- La arquitectura del sistema se diseñará teniendo en cuenta la posibilidad de escalabilidad, de modo que pueda adaptarse a un aumento en la cantidad de productos o usuarios sin requerir cambios drásticos en la infraestructura.

11. Mantenibilidad:

- El código fuente se documentará adecuadamente y se seguirán buenas prácticas de programación para facilitar futuras actualizaciones y mejoras en el sistema.

12. Integración Futura:

- Se dejará espacio para la posible integración con otros sistemas o aplicaciones empresariales existentes en el futuro.

13. Soporte Técnico:

- Se proporcionará documentación de usuario y manuales para ayudar a los usuarios a comprender y utilizar el sistema de manera efectiva.
- Se establecerá un mecanismo de soporte técnico para atender consultas y problemas relacionados con el sistema.

14. Pruebas y Validación:

- Se llevarán a cabo pruebas exhaustivas, incluyendo pruebas de unidad, pruebas de integración y pruebas de usuario, para garantizar la calidad y el funcionamiento adecuado del sistema.

Limitaciones

- **Escalabilidad:** Dependiendo de la arquitectura y el diseño del sistema, podría haber limitaciones en cuanto a la capacidad de manejar grandes volúmenes de productos y usuarios simultáneos de manera eficiente.
- **Rendimiento:** A medida que la cantidad de productos y usuarios vaya creciendo, el rendimiento del sistema puede verse afectado. Las funciones de búsqueda, actualización y eliminación podrían volverse más lentas si no se optimizan adecuadamente.
- **Seguridad:** La seguridad de los datos es una preocupación importante dentro del sistema. Si no se implementan adecuadamente medidas de autenticación y autorización, podría haber riesgo de acceso no autorizado a la información del inventario y perder información del inventario.
- **Capacidades de Personalización:** Dependiendo de la complejidad del sistema y del tiempo disponible, podría haber limitaciones en la capacidad de personalizar la interfaz de usuario y las funciones según las necesidades de cada negocio.
- **Usabilidad:** Si la interfaz de usuario no se diseña de manera intuitiva, los usuarios podrían tener dificultades para aprender y utilizar el sistema de manera eficiente por lo que el sistema no podría ser aprovechado.
- **Actualización y Mantenimiento:** Si no se planifica un proceso de evolución y mantenimiento adecuado el sistema puede volverse obsoleto con el tiempo y enfrentar problemas de compatibilidad con nuevas tecnologías.

Marco Teórico

La gestión de inventario es una función importante en cualquier entorno empresarial porque afecta directamente la eficiencia operativa, la satisfacción del cliente y la rentabilidad de una empresa. En este contexto, se plantea el proyecto de desarrollo de “Gestor de Inventarios en Java” con el objetivo de resolver desafíos comunes relacionados con la gestión de productos e inventarios en entornos empresariales.

La gestión de inventario implica monitorear, controlar y optimizar los productos y recursos disponibles en una organización. En otras palabras, se trata de garantizar que la empresa tenga siempre el producto adecuado en el lugar adecuado y en el momento adecuado. Esto es fundamental para evitar la escasez de productos que afecte las ventas y la satisfacción del cliente y para minimizar la inversión de capital en el exceso de inventario.

El proyecto Gestor de Inventarios en Java tiene como objetivo resolver desafíos comunes en la gestión de inventario, como la falta de automatización y las ineficiencias en el seguimiento de las fluctuaciones de productos e inventario. La mayoría de los sistemas tradicionales carecen de la funcionalidad necesaria para manejar eficazmente la creciente diversidad de productos y la complejidad de la gestión de inventario.

El sistema se centrará en proporcionar soluciones modernas y eficaces que permitan a las empresas gestionar sus productos de forma eficaz. Los usuarios podrán agregar nuevos productos, consultar niveles de stock, actualizar información relevante y eliminar productos obsoletos, todo a través de una interfaz intuitiva y fácil de usar.

Otros sistemas existentes

- **SAP Inventory Management:** SAP ofrece una suite de soluciones empresariales que incluyen gestión de inventarios. Su módulo de gestión de inventarios proporciona herramientas para el seguimiento de existencias, planificación de inventario y optimización de la cadena de suministro.

- **Oracle Inventory Management:** Oracle proporciona un sistema integral de gestión de inventarios que se integra con su suite de aplicaciones empresariales. Ofrece capacidades de seguimiento de existencias en tiempo real, control de costos y gestión de pedidos de reposición.
- **WMS (Warehouse Management System):** Los sistemas de gestión de almacenes son aplicaciones especializadas que se centran en la gestión de inventarios en almacenes y centros de distribución. Ejemplos populares incluyen Manhattan Associates, HighJump, y Blue Yonder (anteriormente JDA Software).
- **QuickBooks Inventory:** QuickBooks es un software de contabilidad ampliamente utilizado que incluye módulos de gestión de inventarios para pequeñas y medianas empresas. Ofrece características básicas de seguimiento de existencias y generación de informes.
- **Zoho Inventory:** Zoho Inventory es una solución basada en la nube que permite a las empresas gestionar su inventario, ventas y compras en un solo sistema. Ofrece seguimiento de productos, gestión de pedidos y análisis de inventarios.
- **Odoo Inventory:** Odoo es una plataforma de código abierto que incluye una amplia gama de módulos empresariales. Su módulo de inventario proporciona funcionalidades como el seguimiento de existencias, la gestión de ubicaciones y la optimización de pedidos.
- **TradeGecko:** TradeGecko es una solución de gestión de inventarios en la nube diseñada para empresas de comercio electrónico y mayoristas. Ofrece seguimiento de existencias, administración de pedidos y sincronización con múltiples canales de venta.
- **Fishbowl Inventory:** Fishbowl es un sistema de gestión de inventarios diseñado para integrarse con QuickBooks. Ofrece funciones avanzadas de seguimiento de existencias, gestión de órdenes de compra y ventas, y automatización de procesos.
- **DEAR Inventory:** DEAR Inventory es un sistema de gestión de inventarios en la nube que se integra con múltiples plataformas de comercio electrónico. Ofrece seguimiento de productos, gestión de pedidos y análisis de inventarios.
- **Cin7:** Cin7 es una solución de gestión de inventarios que se adapta a empresas minoristas, mayoristas y de fabricación. Ofrece funciones de seguimiento de existencias, gestión de pedidos y automatización de procesos.

Comparación con SAP Inventory Management

SAP Inventory Management es parte de la suite de productos SAP ERP (Enterprise Resource Planning) una de las soluciones de software empresarial más utilizadas en todo el mundo. Esto ha contribuido a su amplia adopción en empresas grandes y medianas en diversos sectores.

Este sistema se va a tomar en cuenta por su amplia adopción en diversas empresas de manera que se va a hacer una comparación de sus características con las funcionalidades del sistema a desarrollar.

1. Registro de Productos:

- Sistema: Permite a los usuarios registrar nuevos productos en el sistema, incluyendo información como nombre, descripción, precio y cantidad inicial en stock.
- SAP Inventory Management: Permite la creación de registros de productos con detalles como nombre, descripción, precio y unidades iniciales en stock. Además, SAP permite la importación masiva de datos de productos desde fuentes externas.

2. Visualización de Inventarios:

- Sistema: Permite a los usuarios acceder y visualizar el inventario existente en el sistema.
- SAP Inventory Management: Ofrece una vista completa de los inventarios, mostrando el estado actual de las existencias, ubicaciones y movimientos de productos. Además, permite generar informes detallados sobre el inventario.

3. Consulta de Niveles de Existencias:

- Sistema: Permite a los usuarios consultar los niveles de existencias de productos específicos en el inventario.
- SAP Inventory Management: Proporciona información detallada sobre los niveles de existencias de productos en tiempo real, lo que permite a los usuarios monitorear y tomar decisiones basadas en datos actualizados.

4. Eliminación de Productos:

- Sistema: Proporciona a los usuarios la opción de eliminar productos obsoleto.
- SAP Inventory Management: Permite la eliminación de productos, pero generalmente con procedimientos de control de acceso y registros de auditoría para rastrear quién y cuándo se eliminaron los productos.

5. Búsqueda y Filtrado de Productos:

- Sistema: Permite a los usuarios buscar productos por nombre.
- SAP Inventory Management: Ofrece capacidades de búsqueda avanzada y filtrado que permiten a los usuarios buscar productos utilizando múltiples criterios, como nombre, código, ubicación, etc.

6. Interfaz de Usuario Intuitiva:

- Sistema: Cuenta con una interfaz de usuario intuitiva diseñada para usuarios sin un conocimiento técnico.
- SAP Inventory Management: Proporciona una interfaz de usuario profesional, pero suele requerir una curva de aprendizaje mayor debido a su complejidad y funcionalidades avanzadas.

Tecnologías empleadas

- Java

Java es el lenguaje de programación central en este proyecto. Debido a su portabilidad, flexibilidad y robustez Java permite la creación de aplicaciones escalables y confiables, lo que lo convierte en una elección ideal para un sistema de gestión de inventarios que pueda escalar con futuras mejoras.

- Biblioteca Swing

Swing es una biblioteca gráfica de Java que se utiliza para desarrollar interfaces de usuario. Proporciona componentes gráficos como botones, etiquetas, tablas y paneles que son esenciales para crear una interfaz de usuario intuitiva y atractiva.

- IDE

Se va a hacer uso del IDE NetBeans para desarrollar el proyecto. Esta herramienta proporciona un entorno de desarrollo eficiente y facilita la escritura de código Java y la creación de interfaces de usuario con Swing.

Referencias

- **Bibliografía**

Schildt, H. (2009). Java. (7ma ed.). Mc Graw Hill

Sommerville, I. (2011). Ingeniería de Software. (9na ed.). Pearson

Pressman, R. (2010). Ingeniería del Software. (7ma ed.). Mc Graw Hill.

- **Recursos digitales**

Overview Inventory Management. (s. f.). SAP.

https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/91b21005dded4984bccccf4a69ae1300c/9d688de41e684f2c9c6b467ef8e0ba6f.html

Vaibhavi, K. (1 de octubre de 2022). SAP Inventory Management – Everything you must know. SAP.

[SAP Inventory Management – Everything you must know | SAP Blogs](#)