Facultad 2

Sistema de gestión de inventario de las tiendas de venta de materiales de la construcción de La Habana

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autores:

- Damisela Thalía Gómez Cudina
- Jorge Saúl Pérez Peña

Tutor:

Ing. Vladimir Milián Núñez

La Habana, septiembre de 2023

Año 65 de la Revolución

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título “Sistema de Gestión de inventarios de las tiendas de venta de materiales para la construcción en La Habana” concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firman la presente a los 20 días del mes de septiembre del año 2023.

_____
Firma del Autor

Ing. Vladimir Milián Núñez

_____
Firma del Tutor

_____
Firma del Tutor

Resumen

Para toda empresa es de vital importancia la gestión de inventario de sus productos. Si este proceso es asistido por un sistema informático tomando en cuenta los nuevos avances de las Tecnologías de la Información y las Comunicaciones se garantiza un impacto positivo provocando que la información generada esté más organizada y disponible. La investigación, cuyos resultados se describen en este informe, tuvo como propósito desarrollar un sistema de gestión de inventario para las tiendas de ventas de materiales de construcción de La Habana, que permitió aumentar la disponibilidad y eficiencia de este proceso a sus usuarios. El desarrollo del mismo fue guiado por la metodología de desarrollo de software XP. Para la implementación de este se utilizó como marco de trabajo web Django con su lenguaje de programación Python. Para validar que la solución cumpliera con los requisitos definidos por el cliente, se aplicaron pruebas de aceptación.

Palabras clave: eficiencia, inventario, materiales

Abstract

For every company, inventory management of its products is of vital importance. If this process is assisted by a computer system taking into account the new advances in Information and Communications Technologies, a positive impact is guaranteed, causing the information generated to be more organized and available. The purpose of the research, whose results are described in this report, was to develop an inventory management system for construction materials sales stores in Havana, which allowed them to increase the availability and efficiency of this process for their users. Its development was guided by the XP software development methodology. To implement this, Django with its Python programming language was used as the web framework. To validate that the solution met the requirements defined by the client, acceptance tests were applied.

Keywords: efficiency, inventory, materials

Agradecimientos

Índice

Introducción *1*

Capítulo 1 Fundamentación Teórica *6*

1.1 Gestión de inventarios. Conceptos fundamentales **6**

1.2 Análisis de sistemas de gestión de inventarios **7**

1.2.1 Sistemas internacionales 8
1.2.2 Sistemas Nacionales 10
1.2.3 Valoración crítica 12

1.3 Informatizacion de la gestion de inventarios.Tendencias y tecnologias utilizadas **13**

1.3.1 Herramientas utilizadas 13
1.3.2 Lenguaje de Desarrollo 14
1.3.3 Metodología de Desarrollo 17

Capítulo 2: Planificación y Diseño de la propuesta de solución *23*

2.1 Descripción de la propuesta de solución **23**

2.2 Fase de planeación **23**

2.2.1 Funcionalidades del sistema 23
2.2.2 Historias de Usuario 25
2.2.3 Estimación de esfuerzo 31
2.2.4 Plan de Iteraciones 32
2.2.5 Plan de Entrega 34

2.3 Fase de Diseño **34**

2.3.1 Tarjetas TRC 35
2.3.2 Patrón Arquitectónico 36
2.3.3 Patrones de Diseño 37
2.3.4 Modelado de Datos 41

Conclusiones Parciales **42**

Capítulo 3: Implementación y prueba de la propuesta de solución *43*

3.1 Implementación **43**

3.2.1 Diagrama de despliegue 43
3.2.2 Estándares de codificación 44
3.2.3 Tareas de ingeniería 46

3.3 Fase de Prueba **49**

3.3.1 Pruebas unitarias 52
3.3.2 Pruebas de Aceptación 54

Conclusiones parciales **63**

Conclusiones Generales *64*

Recomendaciones *65*

Referencias bibliográficas *66*

Anexos *68*

Anexo 1 Historias de usuario **68**

Anexo 2 Tareas de ingeniería.....72
Anexo 3 Pruebas de aceptación.....76

Índice de tablas

Tabla 1 Comparación de sistemas homólogos 12
Tabla 2 HU Filtrar Productos 25
Tabla 3 Insertar Producto ..... **¡Error! Marcador no definido.**
Tabla 4 HU Visualizar Reporte ..... **¡Error! Marcador no definido.**
Tabla 5 Exportar Datos ..... **¡Error! Marcador no definido.**
Tabla 6 Estimación de esfuerzo por HU ..... **¡Error! Marcador no definido.**
Tabla 7 Plan de Iteraciones 32
Tabla 8 Plan de Entregas 34
Tabla 9 Tarjeta CRC #1 ..... **¡Error! Marcador no definido.**
Tabla 10 Tarjeta CRC #2 ..... **¡Error! Marcador no definido.**
Tabla 11 Tarjeta CRC #3 ..... **¡Error! Marcador no definido.**
Tabla 12 Tarea de ingeniería 1 ..... **¡Error! Marcador no definido.**
Tabla 13 Tarea de ingeniería 2 ..... **¡Error! Marcador no definido.**
Tabla 14 Tarea de ingeniería 3 ..... **¡Error! Marcador no definido.**
Tabla 15 Tarea de ingeniería 4 ..... **¡Error! Marcador no definido.**
Tabla 16 Tarea de ingeniería 5 ..... **¡Error! Marcador no definido.**
Tabla 17 Tarea de ingeniería 6 ..... **¡Error! Marcador no definido.**
Tabla 18 Técnica de camino básico a la funcionalidad Eliminar_productos() ..... **¡Error! Marcador no definido.**
Tabla 19 Caso de Prueba para el camino básico 1 ..... **¡Error! Marcador no definido.**
Tabla 20 Caso de Prueba para el camino básico 2 ..... **¡Error! Marcador no definido.**
Tabla 21 Caso de Prueba para el camino básico 3 ..... **¡Error! Marcador no definido.**
Tabla 22 Caso de prueba para la funcionalidad Insertar Personal() . **¡Error! Marcador no definido.**
Tabla 23 Caso de prueba de aceptación #1 ..... **¡Error! Marcador no definido.**
Tabla 24 Caso de prueba de aceptación #2 ..... **¡Error! Marcador no definido.**
Tabla 25 Caso de prueba de aceptación #3 ..... **¡Error! Marcador no definido.**
Tabla 26 Caso de prueba de aceptación #4 ..... **¡Error! Marcador no definido.**
Tabla 27 Caso de prueba de aceptación #5 ..... **¡Error! Marcador no definido.**
Tabla 28 Caso de prueba de aceptación #6 ..... **¡Error! Marcador no definido.**
Tabla 29 No conformidades encontradas ..... **¡Error! Marcador no definido.**
Tabla 30 HU Autenticar Usuario 68
Tabla 31 HU Registrar Usuario ..... **¡Error! Marcador no definido.**
Tabla 32 HU Modificar Usuario ..... **¡Error! Marcador no definido.**
Tabla 33 HU Eliminar Usuario ..... **¡Error! Marcador no definido.**
Tabla 34 HU Buscar Usuario ..... **¡Error! Marcador no definido.**
Tabla 35 HU Listar Usuarios ..... **¡Error! Marcador no definido.**
Tabla 36 HU Modificar Producto ..... **¡Error! Marcador no definido.**
Ta**bla 37** HU Eliminar Producto ..... **¡Error! Marcador no definido.**
Tabla 38 HU Listar Productos ..... **¡Error! Marcador no definido.**
Ta**bla 39** HU Buscar Producto ..... **¡Error! Marcador no definido.**
Tabla 40 HU Insertar Proveedor ..... **¡Error! Marcador no definido.**
Tabla 41 HU Modificar Proveedor ..... **¡Error! Marcador no definido.**
Ta**bla 42** HU Eliminar Proveedor ..... **¡Error! Marcador no definido.**

Tabla 43 HU Listar Proveedores ..... **¡Error! Marcador no definido.**
Tabla 44 Tarea de ingeniería 7 ..... **¡Error! Marcador no definido.**
Tabla 45 Tarea de ingeniería 8 ..... **¡Error! Marcador no definido.**
Tabla 46 Tarea de ingeniería 9 ..... **¡Error! Marcador no definido.**
Tabla 47 Tarea de ingeniería 10 ..... **¡Error! Marcador no definido.**
Tabla 48 Tarea de ingeniería 11 ..... **¡Error! Marcador no definido.**
Tabla 49 Tarea de ingeniería 12 ..... **¡Error! Marcador no definido.**
Tabla 50 Tarea de ingeniería 13 ..... **¡Error! Marcador no definido.**
Tabla 51 Tarea de ingeniería 14 ..... **¡Error! Marcador no definido.**
Tabla 52 Tarea de ingeniería 15 ..... **¡Error! Marcador no definido.**
Tabla 53 Tarea de ingeniería 16 ..... **¡Error! Marcador no definido.**
Tabla 54 Tarea de ingeniería 17 ..... **¡Error! Marcador no definido.**
Tabla 55 Caso de prueba de aceptación #7 ..... **¡Error! Marcador no definido.**
Tabla 56 Caso de prueba de aceptación #8 ..... **¡Error! Marcador no definido.**
Tabla 57 Caso de prueba de aceptación #9 ..... **¡Error! Marcador no definido.**
Tabla 58 Caso de prueba de aceptación #10 ..... **¡Error! Marcador no definido.**
Tabla 59 Caso de prueba de aceptación #11 59
Tabla 60 Caso de prueba de aceptación #12 ..... **¡Error! Marcador no definido.**
Tabla 61 Caso de prueba de aceptación #13 ..... **¡Error! Marcador no definido.**
Tabla 62 Caso de prueba de aceptación #14 ..... **¡Error! Marcador no definido.**
Tabla 63 Caso de prueba de aceptación #15 60
Tabla 64 Caso de prueba de aceptación #16 ..... **¡Error! Marcador no definido.**
Tabla 65 Caso de prueba de aceptación #17 ..... **¡Error! Marcador no definido.**

Índice de Figuras

Figura 1 Modelo-Vista-Template 37
Figura 2 Captura de fragmento de código del archivo urls.py 38
Figura 3 Fragmento de código de models.py 39
Figura 4 Utilización del patrón front-controller 40
Figura 5 Utilización del patrón decorador 41
Figura 6 Modelo de datos 42
Figura 7 Diagrama de Despliegue 43
Figura 8 Fragmento de código del archivo de pruebas unitarias automatizadas 53
Figura 9 Resultado de las pruebas de aceptación ..... **¡Error! Marcador no definido.**

Introducción

Las Tecnologías de la Información y las Comunicaciones (TIC) son el conjunto de recursos, herramientas, equipos, programas informáticos, aplicaciones, redes y medios, que permiten la recopilación, el procesamiento, el almacenamiento, la transmisión y la difusión de la información. Las TIC han generado incontables beneficios para la sociedad, al facilitar el acceso, el intercambio y la creación de conocimiento, así como el impacto que ha tenido en la comunicación, la educación, la salud, el entretenimiento y la participación ciudadana. Las TIC también han transformado el mundo empresarial, al permitir la automatización, la optimización y la innovación de los procesos, productos y servicios (Costa Rica, 2020).

El manejo de inventarios es una actividad esencial para el funcionamiento de cualquier empresa, ya que implica el control de los recursos materiales que se utilizan para la producción o la comercialización de bienes o servicios. El manejo de inventarios requiere de un sistema que permita registrar, clasificar, valorar y analizar los movimientos de los recursos materiales, con el objetivo de optimizar su uso, reducir los costos y mejorar la rentabilidad.

El enfoque tradicional, en lo que respecta a la gestión de inventarios, se basa en los conceptos de punto de pedido y cantidad a pedir como base para tomar las decisiones de: ¿qué pedir?, ¿cuánto pedir?, ¿cuándo pedir? y ¿cómo pedir? (Portugal et al., 2023). La gestión de inventarios constituye una de las actividades fundamentales dentro de la gestión de la cadena de suministro, ya que el nivel de inventarios puede llegar a suponer la mayor inversión de la compañía, abarcando incluso más del 50% del total de activo en el sector de la distribución (Ballou, 1991)

En el VI Congreso del Partido Comunista de Cuba se evaluó que a pesar de estudiarse y adoptarse en el transcurso de los años diversas medidas con el objetivo de mejorar el desempeño de la economía nacional, aún existe un grupo de insuficiencias y problemáticas que son necesarias resolver en el presente. En función de todo lo anterior, se definieron y aprobaron por el Congreso del Partido los Lineamientos de la Política Económica y Social del Partido y la Revolución, dentro de los cuales se encuentra el lineamiento 312, que plantea: ejercer un efectivo control sobre la gestión de compra y la rotación de los inventarios con vista a minimizar la inmovilización de recursos y las pérdidas (Vitlloch et al., 2012). En Cuba, el manejo de inventarios se realiza de acuerdo con las normas y procedimientos establecidos por el Ministerio de Finanzas y Precios, que regulan los aspectos contables, administrativos y tributarios de esta actividad.

Introducción

La construcción es una actividad económica fundamental en cualquier país, ya que contribuye al desarrollo de la infraestructura, la vivienda, el empleo y el bienestar social. En Cuba, la construcción es un sector clave para el desarrollo económico y social, especialmente en La Habana, la capital y la ciudad más poblada del país. Las tiendas de materiales de construcción juegan un papel importante en este sector, ya que suministran los materiales necesarios para la construcción de obras públicas y privadas.

Por esta razón, la gestión de inventarios en estas tiendas deviene en una de las actividades claves para el correcto desempeño empresarial. Sin embargo, se ha evidenciado que esta actividad se ejecuta de manera deficiente, al no tener definidos procedimientos de inventarios adecuados para cada uno de los productos, lo que repercute directamente en altos costos, inventarios ociosos, poca variedad de productos, poca optimización de las capacidades de almacenaje según las normas establecidas y bajos niveles de servicio al cliente. Además, el proceso de control de sus inventarios es realizado manualmente y los informes son elaborados en un documento de Excel.

A partir de la aplicación de entrevistas y reuniones informales con los encargados de almacén, vendedores y comerciales como técnica seleccionada para la recopilación de información, se identificaron varias limitaciones en la realización del proceso de gestión de inventario de las tiendas de materiales de la construcción de La Habana, que dan origen a la presente investigación:

1. Demora en la búsqueda de un determinado producto que existe o falta en el almacén.
2. Falta de inmediatez en la obtención de informes del inventario.
3. Atraso en la obtención de reportes de productos faltantes.
4. En ocasiones se han hecho solicitudes de productos los cuales no se encuentran en existencia o su entrega ha sido tardía.
5. Ocurre pérdida de información asociada a la gestión del inventario de los materiales debido al deterioro de los documentos archivados
6. Problemas con productos que presentan lenta rotación en los inventarios.

Todos estos aspectos que de forma negativa repercuten en la empresa objeto de estudio, exigen la búsqueda de una herramienta que permita gestionar los inventarios con el fin de satisfacer las necesidades de sus clientes y a la vez, les permita ser más eficientes en su trabajo e incurrir en costos mínimos.

Teniendo en cuenta la situación problemática anterior surge el siguiente **problema científico**: ¿Cómo automatizar el proceso de gestión de los inventarios de las tiendas de venta de materiales de la construcción de La Habana?

Objeto de estudio: Gestión de los inventarios de tiendas de venta de materiales de la construcción.

Campo de acción: Informatización de la gestión de los inventarios de las tiendas de venta de materiales de la construcción de La Habana**.**

Objetivo general: Desarrollar una aplicación web para la gestión automática de los inventarios de las tiendas de venta de materiales de la construcción de La Habana, que permita agilizar este proceso.

Tareas de investigación:

1. Sistematización de los principales fundamentos teóricos y metodológicos de la gestión de los inventarios de las tiendas de venta de materiales de la construcción de La Habana.
2. Identificación de la estructura (o los componentes) que poseen las soluciones existentes para la gestión de los inventarios de las tiendas de venta de materiales de construcción.
3. Caracterización de las funcionalidades a tener en cuenta para desarrollar el sistema informático que gestionará los inventarios de las tiendas de venta de materiales de construcción de La Habana.
4. Diseño de las funcionalidades del sistema de gestión de los inventarios de las tiendas de venta de materiales de la construcción de La Habana.
5. Implementación de las funcionalidades del sistema de gestión de los inventarios de las tiendas de venta de materiales de la construcción de La Habana.
6. Validación, a través de una estrategia de pruebas, de las funcionalidades del sistema de gestión de los inventarios de las tiendas de venta de materiales de la construcción de La Habana.

Métodos de investigación:

El método científico de investigación es la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones. Los métodos científicos se pueden clasificar en teóricos y empíricos, los cuales están dialécticamente relacionados.

Métodos teóricos:

- **Analítico-Sintético:** se utilizó en el análisis bibliográfico del tema y para tomar puntos de vista de los aspectos esenciales relacionados con el desarrollo de los procesos de inventario de tiendas de venta de materiales de la construcción. Obteniendo la información necesaria mediante la revisión de documentos, artículos científicos y sitios webs revisados y acordes al tema, además se analizaron las diferentes herramientas, metodologías y tecnologías de desarrollo de software a utilizar en la propuesta de solución.
- **Inductivo-deductivo:** para arribar a conclusiones acerca de la gestión de inventarios en tiendas de venta de materiales de la construcción en La Habana, que permitan desarrollar un sistema informático que asegure a la vez, una mejor gestión del negocio.
- **Modelado:** se utiliza en la modelación de los diagramas correspondientes a la etapa de análisis, diseño e implementación del módulo a desarrollar en la presente investigación.

Métodos empíricos:

- **Observación:** para comprender la forma en que se realiza el proceso de la gestión de los inventarios de tiendas de venta de materiales de la construcción.
- **Entrevista:** se realizó para recoger información sobre la gestión de los inventarios de las tiendas de venta de materiales de la construcción desde la perspectiva de la Empresa Comercial de La Habana. Se entrevistó a la directora general de la empresa, quien explicó los problemas y desafíos que enfrenta, y las expectativas y necesidades que tiene. Los resultados de la entrevista se analizaron para caracterizar las funcionalidades de la aplicación web que se propone como solución.

La presente investigación está estructurada en tres (3) capítulos, los cuales se resumen brevemente a continuación:

Capítulo 1: Fundamentación Teórica

En este capítulo se presentan los elementos teóricos que sirven de base a la investigación del problema planteado. Se describen los conceptos fundamentales asociados al dominio del problema; además, se proporciona una descripción de las herramientas, tecnologías y metodología de desarrollo de software a utilizar para dar solución al problema planteado.

Capítulo 2: Planificación y diseño de la propuesta de solución

En este capítulo se realiza la planificación y diseño de la propuesta de solución según los requisitos funcionales obtenidos. Se realiza una descripción del sistema que se pretende

desarrollar, se elaboran las Historias de Usuarios para caracterizar las funcionalidades con las que debe cumplir el sistema, además de presentar la arquitectura seleccionada para su desarrollo.

Capítulo 3: Implementación y prueba de la propuesta de solución

En este capítulo se presentan los elementos relacionados con la implementación del sistema. Se define la estrategia de pruebas a realizar para verificar el cumplimiento de los requisitos de software y se presenta el resultado del proceso de prueba y de la validación del sistema.

Capítulo 1 Fundamentación Teórica

El presente capítulo recoge los elementos teóricos que fundamentan el desarrollo de un sistema de gestión de los inventarios de las tiendas de venta de materiales de la construcción de La Habana**.** Se realiza un análisis de la situación actual y perspectivas de las soluciones existentes en el mercado nacional e internacional, resaltando las tendencias y tecnologías más comunes en su desarrollo. Finalmente, se exponen los elementos más relevantes de la metodología, lenguajes y herramientas definidas para el desarrollo.

1.1 Gestión de inventarios. Conceptos fundamentales

A continuación, se presenta una definición de los conceptos que pueden ayudar a contextualizar el tema:

Sistema de Planificación de Recursos Empresariales (ERP): son sistemas de gestión de información gerenciales que integran, automatizan y manejan muchas de las acciones asociadas con las operaciones de producción y distribución de una compañía dedicada a la producción de bienes o servicios. Los ERP pueden intervenir en el control de muchas actividades de negocios, tales como ventas, compras, entregas, pagos, producción, contabilidad, administración de inventarios y de recursos humanos. Funcionan en todo tipo de empresas. Para ello, integran todos los departamentos funcionales de la empresa, herramientas de mercadotecnia y administración estratégica en un solo sistema (Wailgum, 2008).

Características de los sistemas ERP

Un ERP se distingue de otro software empresarial porque es un sistema integral, con modularidad y adaptable (Wailgum, 2008):

- Integrales: Todos los departamentos o áreas de una empresa se relacionan entre sí, permitiendo de esta forma controlar los distintos procesos de la entidad, donde la salida de un proceso puede ser la entrada de otro.

- Modulares: Se dividen en módulos, que son las áreas que se integran como un todo para el manejo óptimo de la información. Estos módulos se instalan según las necesidades del cliente.

- Adaptables: Son diseñados para ser configurables según lo que cada empresa necesite.

- Base de datos centralizada.

- Sus componentes interactúan entre sí consolidando todas las operaciones.
- En un sistema ERP los datos se ingresan sólo una vez y deben ser consistentes, completos y comunes.
- Las empresas que lo implanten suelen tener que modificar alguno de sus procesos para alinearlos con los del sistema ERP. Este proceso se conoce como Reingeniería de Procesos, aunque no siempre es necesario

Inventario: Es una lista elaborada que corrobora la existencia real de productos y materias primas almacenadas, verifica su estado y define su estatus. Para ser capaz de llevar un correcto control de todas tus existencias, se recomienda el uso de una herramienta que pueda estipular el total de movimientos de tu almacén de forma actualizada (MEANA COALLA, PEDRO, 2017).

Gestión de inventarios y control de inventario: Para este trabajo de diploma es necesario ver los conceptos de gestión y control del inventario como uno solo, ya que el inventario a controlar no es una mercancía ni está almacenado en su totalidad por lo que parte del está actualmente en uso y uno solo de los dos conceptos no abarca la definición de dicho inventario. Para llegar a esta conclusión fue necesario estudiar estos conceptos y compararlos en paralelo con la problemática a investigar:

La **gestión de inventarios** se incluye dentro de la rama de la contabilidad de costes y se define como la administración adecuada del registro, compra y salida de inventario dentro de la empresa (Sepúlveda, 2022). Su objetivo es mantener la productividad en el manejo de las existencias. Es decir, llevar a cabo todas las tareas en el menor plazo y con el menor esfuerzo posible. Esto se logra, por ejemplo, usando códigos de barras para identificar rápidamente de forma electrónica las mercancías almacenadas. Igualmente, parte de estas tareas es la distribución estratégica de la mercancía, de manera que aquella de mayor rotación se encuentre más cerca a la salida del almacén y pueda ser despachada en menos tiempo (Westreicher, 2020).

El **control de inventario:** permite llevar el registro de las existencias de la compañía. Así, en base a dichos datos pueden tomarse decisiones, por ejemplo, para que las firmas mantengan en lo posible el menor stock inmovilizado, lo cual implica un coste de almacenamiento (Westreicher, 2020).

1.2 Análisis de sistemas de gestión de inventarios

En el contexto de la presente investigación, se aborda un aspecto fundamental en la gestión empresarial, específicamente en el ámbito de las tiendas de venta de materiales de construcción:

la gestión de inventarios. La gestión de inventarios es un pilar esencial para la eficiencia operativa y la rentabilidad de cualquier empresa, y se torna particularmente crítica en el caso de establecimientos que ofrecen materiales de construcción, dada la variabilidad en la demanda y la diversidad de productos que deben mantener en stock.

Es relevante destacar que, a pesar de la singularidad de las necesidades y productos que caracterizan a estas tiendas, no existe un conjunto de sistemas de gestión de inventarios específicamente diseñados para ellas. En su lugar, lo que prevalece en el mercado son sistemas de gestión de inventarios en general, que ofrecen una amplia gama de funcionalidades y adaptabilidad para ser implementados en diversas industrias, incluyendo la venta de materiales de construcción.

A través de esta investigación, se pretende demostrar que, aunque no existan sistemas de gestión de inventario exclusivamente orientados a tiendas de materiales de construcción, existen soluciones versátiles y adaptables que han demostrado ser efectivas en este contexto; ejemplo de ellas son:

1.2.1 Sistemas internacionales

Herramienta SOROLLA2. (España)

Dentro del sistema SOROLLA2, el módulo de gestión de inventario (GDI), se propone como una herramienta perfectamente integrada dentro de SOROLLA2 para los diversos órganos gestores a los que pueda interesar, desde distintas perspectivas, la composición del inventario de un ente y su evolución en el tiempo. Para ello, en GDI, se realiza la gestión de forma individualizada de todos los bienes inventariables, es decir, aquellos que constituyen el inmovilizado, tanto material como inmaterial, del ente, sea cual sea el título que dé lugar a su inclusión en el inventario, así como de los derechos que puedan recaer sobre este tipo de bienes. Queda excluido el inmovilizado financiero. El sistema se basa en un modelo organizativo que en la versión actual está compuesto por órganos gestores, unidades tramitadoras de tipo expediente, unidades tramitadoras de tipo caja, centros de información de gestión presupuestaria. La versión evolucionada incluirá también el órgano de contratación, las unidades proponentes y al gestor de inventario (Pérez S., 2020).

Este sistema tiene como requisitos de Software:

- Ordenador con conexión a la red SARA[1] ((Sistemas de aplicaciones y redes para las administraciones) o a Internet con navegador de Internet:
    - Internet Explorer 7.0 o superior
    - Firefox 3.5 o superior.
- Certificado electrónico para el acceso al sistema.
- Visor de PDF para la visualización de los informes generados.
- Requisitos para la firma electrónica mediante certificado

Este sistema es una aplicación web, y los requisitos de software que necesitan constituyen una limitante para su uso en nuestro país. Primeramente, necesita de la conexión a la red SARA. Esta red implementa importantes medidas de seguridad entre las que destaca el establecimiento de redes virtuales privadas. Es una red extremadamente segura en la que todo el tráfico circula cifrado por la Troncal. De esta manera queda asegurada la confidencialidad de la información que viaja a través de la red SARA.

Se requiere además de un certificado electrónico para poder acceder al sistema, lo cual implica un alto costo por concepto de licencia, que la empresa no puede pagar.

CS – Almacenes (España)

CS-Almacenes es una plataforma que permite controlar el inventario de almacenes. El programa se basa en ir registrando las entradas y salidas de manera que siempre conocen qué cantidad hay de un producto en tiempo real. Controla productos en ilimitados almacenes, llevando a cabo una realización de ajustes manuales si se producen inventarios físicos y conteos, así como realizar traspasos entre almacenes.

Es una aplicación fácil de utilizar debido a que está dirigida a profesionales que no necesitan el control de trazabilidad compleja, si no que permite conocer en todo momento su existencia. La desventaja que presenta esta plataforma es que no permite la generación de estadísticas para determinados productos ni el conocimiento de la localización de estos. Es una plataforma privativa (PAE, 2021).

[1] SARA: conjunto de infraestructuras de comunicaciones y servicios básicos que conecta las redes de las Administraciones Públicas Españolas e Instituciones Europeas facilitando el intercambio de información y el acceso a los servicios.

1.2.2 Sistemas Nacionales

ConDor

El sistema de control de inventarios ConDor está concebido para el control de las entradas o salidas de uno o varios almacenes y además para la contabilización de los movimientos que se realizan con los productos almacenados. Está generalizado en más de mil entidades del territorio nacional siendo los organismos más representativos el MITRANS y el Instituto nacional de recursos hidráulicos (INRH). Cuenta con una suite integrada por siete módulos: contabilidad general, activos fijos, inventario, nómina/prenómina, disponibilidad financiera, condexce y efectos. Actualmente este software se encuentra implementado para el sistema operativo windows en cualquiera de sus versiones. La versión del módulo existente presenta la característica que está implementado en un entorno monousuario, lo que ha sido señalado por parte de los clientes como una significativa desventaja. También este sistema presenta la característica que cada módulo tiene una base de datos diferente, lo que trae redundancia en la información (Fuentes, 2008).

Entre sus módulos se encuentra el de Gestión de Inventarios, el cual permite entre otras funcionalidades:

- Control de los esquemas de depósitos y almacenes.
- Realizar inventarios rotativos.
- Serializar los productos.
- Generar transacciones de costeo.

ASSETS

El sistema de gestión integral (ASSETS) es un sistema multiusuario que se monta en una plataforma de servidores SQL, dividido en módulos económicos que trabajan en conjunto para el control de las actividades económica, financiera y contable sobre los medios materiales y financieros. El módulo inventario del ASSETS divide los medios en dos tipos de cuentas: activos fijos y útiles herramientas. Estas últimas son las que se utilizan para realizar las actividades de mantenimiento, talleres, almacenes, así como los equipos de protección física. Comprende entre otros, herramientas manuales, artículos de protección personal, utensilios de laboratorios, mini calculadoras, utensilios menores de cocina. Los medios que pertenecen a esta cuenta tienen poco tiempo de duración (González Brito & Lezano Lozadas, 2005).

Versat-Sarasola

Es un producto cubano. Es el primer Sistema Integral de gestión de contabilidad certificado, desarrollado para la gestión económica eficaz y fiable. Actualmente es utilizado en Cuba en alrededor de 200 entidades de varias provincias y en lo adelante será introducido en más de dos mil 500 unidades presupuestadas.

Este sistema integrado cuenta con un conjunto de 12 módulos entre los que se encuentran:

- Configuración y seguridad
- Contabilidad general y de gastos
- Costos y procesos
- Análisis económico empresarial
- Control de activos fijos
- Finanza y caja
- Planificación y presupuestos
- Control de inventarios
- Pago de salario
- Paquete de gestión
- Contratación
- Facturación

En el módulo de Control de Inventarios se definen formatos del clasificador de productos para lograr una uniformidad en el registro y la agregación de información en los reportes de salida, se conceptualizan los movimientos para lograr una información amplia sobre los orígenes y destinos de los recursos. Permite el control de las existencias y movimientos en diferentes monedas. Muestra el cuadre diario de cada uno de los almacenes por las diferentes cuentas. Ofrece la posibilidad de duplicar documentos para agilizar los pases de los mismos y lograr que un mismo documento se convierta en otro con solo adicionar un mínimo de información, realiza un control

de las existencias y movimientos por custodios y se emiten diferentes reportes e información de utilidad para la correcta administración de los recursos materiales (Del Toro Ríos & González Brito, 2009).

Características:

- Es una aplicación de escritorio.
- Implementado en Delphi.
- Trabaja sobre el sistema operativo Windows.
- Soporte para base de datos SQL Server 2000.

1.2.3 Valoración crítica

Tabla 1 *Comparación de sistemas homólogos*

Sistema	Localización de Productos	Estadísticas	Conexión a Internet	Registros de Transacciones	Registros de Ventas
Herramienta SOROLLA2	No	No	Sí	Sí	Sí
CS-Almacenes	No	No	No	Sí	Sí
ConDor	No	Sí	No	Sí	Sí
ASSETS	No	No	No	No	No
Versat-Sarasola	No	Sí	No	Sí	Sí

En conclusión, tras analizar diversos sistemas internacionales como Herramienta SOROLLA2 y CS-Almacenes, se observa que, si bien ofrecen funcionalidades atractivas, sus requisitos de software y limitaciones financieras los descartan como opciones viables para una empresa en Cuba. Por otro lado, tanto ConDor como ASSETS presentan limitaciones significativas, como la ausencia de generación de estadísticas o un enfoque centrado en activos fijos, lo que los hace menos adecuados para una tienda de materiales de construcción. Se destaca que el sistema más idóneo para satisfacer las necesidades específicas es Versat-Sarasola.

A pesar de la ausencia de funcionalidades de localización de productos en todos los sistemas evaluados, Versat-Sarasola sobresale al ofrecer estadísticas detalladas, aspecto esencial para la gestión efectiva de inventarios. Además, su capacidad para proporcionar registros de transacciones y ventas facilita un control del flujo de productos y la gestión de ventas en tiempo real. En virtud de estas consideraciones, se concluye que la implementación de Versat-Sarasola se presenta como la elección más apropiada para optimizar la gestión de inventarios en una tienda de materiales de construcción.

1.3 Informatización de la gestión de inventarios. Tendencias y tecnologías utilizadas

A continuación, se realiza un análisis de las metodologías con el propósito de determinar la más conveniente para la solución propuesta. Además, se describen las tecnologías y herramientas seleccionadas para el desarrollo de la solución.

1.3.1 Herramientas utilizadas

Microsoft Visual Studio Code v1.82

Es un editor de código ligero y adaptable desarrollado por Microsoft, puede correr sobre Windows, macOS y Linux. Tiene soporte para JavaScript, TypeScript and Node.js, pero usando su rico ecosistema de extensiones se puede personalizar a niveles importantes, permitiendo construir un entorno de desarrollo ideal para cualquier programador, en casi cualquier lenguaje de programación, agregándole capacidades que rivalizan con cualquier Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) especializado (Microsoft, 2022).

Visual Paradigm v17.1

Visual Paradigm es una herramienta CASE: Ingeniería de Software Asistida por Computación. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

Visual Paradigm ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta privada disponible en varias ediciones, cada una destinada a satisfacer diferentes necesidades: Enterprise, Professional, Community, Standard, Modeler y Personal. Existe una alternativa libre y gratuita de este software, la versión Visual Paradigm UML 6.4 Community Edition (Community Edition, ya que existe la Enterprise, Professional, etc.). Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos (Visual Paradigm, 2021).

1.3.2 Lenguaje de Desarrollo

Lenguaje de modelado Unificado (UML)

UML, abreviatura de Unified Modeling Language, es un lenguaje de modelado estandarizado que consiste en un conjunto integrado de diagramas, desarrollado para ayudar a los desarrolladores de sistemas y software a especificar, visualizar, construir y documentar los artefactos de los sistemas de software, así como para el modelado de negocios y otros sistemas que no son de software. La UML representa una colección de las mejores prácticas de ingeniería que han demostrado tener éxito en el modelado de sistemas grandes y complejos. La UML es una parte muy importante del desarrollo de software orientado a objetos y del proceso de desarrollo de software. La UML utiliza principalmente notaciones gráficas para expresar el diseño de proyectos de software. El uso de la UML ayuda a los equipos de proyecto a comunicarse, explorar posibles diseños y validar el diseño arquitectónico del software. En este artículo, le daremos ideas detalladas sobre lo que es UML, la historia de UML y una descripción de cada tipo de diagrama UML, junto con ejemplos de UML (Visual Paradigm, 2022).

Python v3.11.5

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el aprendizaje automático. Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

Los beneficios de Python incluyen los siguientes:

- Los desarrolladores pueden leer y comprender fácilmente los programas de Python debido a su sintaxis básica similar a la del inglés.

- Python permite que los desarrolladores sean más productivos, ya que pueden escribir un programa de Python con menos líneas de código en comparación con muchos otros lenguajes.
- Python cuenta con una gran biblioteca estándar que contiene códigos reutilizables para casi cualquier tarea. De esta manera, los desarrolladores no tienen que escribir el código desde cero.
- Los desarrolladores pueden utilizar Python fácilmente con otros lenguajes de programación conocidos, como Java, C y C++.
- La comunidad activa de Python incluye millones de desarrolladores alrededor del mundo que prestan su apoyo. Si se presenta un problema, puede obtener soporte rápido de la comunidad.
- Hay muchos recursos útiles disponibles en Internet si desea aprender Python. Por ejemplo, puede encontrar con facilidad videos, tutoriales, documentación y guías para desarrolladores.
- Python se puede trasladar a través de diferentes sistemas operativos de computadora, como Windows, macOS, Linux y Unix (Amazon, 2019).

HTML 5

HTML es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto definiéndolo de forma sencilla, "HTML es lo que se utiliza para crear todas las páginas web de Internet". Más concretamente, HTML es el lenguaje con el que se "escriben" la mayoría de páginas web. Los diseñadores utilizan el lenguaje HTML para crear sus páginas web, los programas que utilizan los diseñadores generan páginas escritas en HTML y los navegadores que utilizamos los usuarios muestran las páginas web después de leer su contenido HTML. Aunque HTML es un lenguaje que utilizan los ordenadores y los programas de diseño, es muy fácil de aprender y escribir por parte de las personas. En realidad, HTML son las siglas de HyperText Markup Language y más adelante se verá el significado de cada una de estas palabras (JIMÉNEZ ALVARADO, 2018).

CSS 3

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc (Eguiluz Pérez, 2008).

JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java (Eguiluz, 2018).

Bootstrap 5

El framework bootstrap 5 es una de las versiones recientemente lanzadas del Framework Bootstrap para el desarrollo web. Esta es una de las librerías más conocidas, pues se pueden construir aplicaciones web adaptables para móvil con el CDN de open source jsDelivr y una página con una plantilla de inicio.

En términos de la versión 5, además de algunas mejoras sobre muchas herramientas, se tomaron decisiones relacionadas al uso y a la aplicación del framework. Por ejemplo, esta versión no es compatible con Internet Explorer ni con JQuery. Asimismo, se integraron con el generador de sitios web estáticos Hugo, dejando de lado a Jekyll.

Por otro lado, se agregaron variables CSS, perfeccionaron el Grid CSS y se permitió añadir una nueva API de interfaz con Sass, donde puedes incluir tus propias herramientas (KeepCoding, 2022).

Django 4.2.5

El framework Django es un software que puede utilizar para desarrollar aplicaciones web de forma rápida y eficiente. La mayoría de las aplicaciones web tienen varias funciones comunes, como la autenticación, la recuperación de información de una base de datos y la administración de cookies. Los desarrolladores tienen que codificar una funcionalidad similar en cada aplicación web que escriban. Django facilita su trabajo al agrupar las diferentes funciones en una gran colección de módulos reutilizables, llamada marco de aplicación web. Los desarrolladores utilizan el marco web de Django para organizar y escribir su código de manera más eficiente y reducir significativamente el tiempo de desarrollo web (Holovaty, Kaplan-Moss, 2020).

1.3.3 Metodología de Desarrollo

Programación Extrema (Extreme Programming, XP)

Para dar cumplimiento a uno de los objetivos de esta investigación es imprescindible el uso de una metodología de desarrollo de software que controle cada uno de los procesos que lleva a cabo el equipo de trabajo. Teniendo en cuenta que el equipo es pequeño, integrado en su totalidad por dos personas (el desarrollador y el cliente, quien participa constantemente con el equipo de desarrollo), los requerimientos están sujetos a cambios, el proyecto es pequeño y de corto plazo, se selecciona como metodología XP para guiar el proceso de desarrollo de la solución.

La Programación Extrema, también conocida como Extreme Programming (XP), constituye una metodología de desarrollo en ingeniería de software creada por Kent Beck, quien es reconocido como el autor pionero en el tema a través de su obra "Extreme Programming Explained: Embrace Change" publicada en 1999. Destacándose como la modalidad más prominente dentro de los procesos ágiles de desarrollo de software, la programación extrema se aparta de las metodologías tradicionales al poner un énfasis considerable en la adaptabilidad en lugar de la previsibilidad, alineándose así con los principios fundamentales de los enfoques ágiles.

Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Se aplica de manera dinámica durante el ciclo de vida del software. Es capaz de adaptarse a los cambios de requisitos. Los individuos e interacciones son más importantes que los procesos y herramientas. Es más crucial formar un equipo sólido que construir el entorno en el que operará. A menudo, se incurre en el error de establecer el entorno inicialmente y confiar en que el equipo se adapte automáticamente. Sin embargo, resulta más efectivo crear primero el equipo y permitir que este configure su propio entorno de desarrollo conforme a sus necesidades específicas.

La metodología XP determina que un Software que funcione es más importante que documentación exhaustiva por tanto se enfocan en desarrollar un software que funciona más que conseguir una buena documentación. La regla a seguir es no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante. Estos documentos deben ser cortos y centrarse en lo fundamental (Bustamante, 2014).

En el contexto de la metodología XP, se introducen diversas fases que son fundamentales para el desarrollo efectivo de software. Estas etapas, diseñadas para priorizar la entrega de un software funcional, enfatizan la colaboración y la adaptabilidad.

En la elección de las fases de la metodología en la presente investigación, se consideraron dos elementos de importancia. En primer lugar, se persiguió el objetivo de minimizar el tiempo necesario para la elaboración de los artefactos correspondientes a cada fase. En segundo lugar, se determinó que dos individuos llevaran a cabo todas las tareas asociadas con la implementación del sistema.

Cabe destacar que la metodología XP se compone de diversos artefactos diseñados para orientar el desarrollo del proyecto. Entre estos artefactos se incluyen las Historias de Usuario, las Tareas de Ingeniería, las tarjetas Clase-Responsabilidad-Colaborador, las Pruebas Unitarias y de Aceptación, así como la Integración Continua.

Las **fases de la metodología de programación extrema o XP** se basan en 4 pasos, estos son:

1. **Planificación**: va de acuerdo con las historias de usuario, estas se priorizan y descomponen en mini versiones, luego la planificación se va revisando cada dos semanas aproximadamente, después de las iteraciones, para obtener un software útil, funcional, listo para las pruebas y su lanzamiento. Los artefactos usados en esta metodología para la fase de planificación son el plan de iteraciones, plan de entregas e Historias de Usuarios
2. **Diseño**: se enfoca en la creación de código de manera simple, implementando lo esencial para lograr funcionalidad y obtener un prototipo inicial. En el caso de que el diseño del software siga una orientación hacia objetos, se procede a generar tarjetas CRC (Clase-Responsabilidad-Colaboración) como parte del proceso de diseño. Esta fase busca garantizar eficiencia y claridad en la implementación, priorizando la funcionalidad básica y sentando las bases para el desarrollo subsiguiente.

3. **Codificación**: se adopta un enfoque colaborativo, donde dos programadores trabajan juntos frente al mismo ordenador. En algunos casos, se realiza un intercambio de parejas para garantizar que el código resultante sea más universal, facilitando que cualquier otro miembro del equipo pueda trabajar en él y comprenderlo. El objetivo es lograr una coherencia tal en la programación que dé la impresión de haber sido desarrollada por un único individuo. Este enfoque busca asegurar una organización y planificación efectivas en el código generado.
4. **Pruebas**: se prioriza la automatización continua, una característica esencial en proyectos de corto plazo. Incluso el cliente tiene la oportunidad de participar activamente, llevando a cabo pruebas, sugiriendo nuevas pruebas y validando las mini versiones. Los elementos esenciales utilizados en esta metodología durante la fase de pruebas incluyen las pruebas unitarias y las pruebas de aceptación. Esta fase busca garantizar la calidad del software mediante un enfoque colaborativo y una validación constante.

En la presente investigación, se opta por la metodología XP (Extreme Programming) como enfoque metodológico ideal para la consecución de los objetivos propuestos. La elección de XP se fundamenta en su capacidad demostrada para fomentar la agilidad y eficiencia en el desarrollo de proyectos, factores cruciales en la obtención de resultados precisos y oportunos. XP se distingue por su enfoque iterativo e incremental, lo cual se alinea de manera óptima con las necesidades de la investigación, permitiendo una adaptación ágil a los cambios y una constante retroalimentación durante el proceso. La metodología XP se caracteriza, además, por su énfasis en la colaboración estrecha entre los miembros del equipo, promoviendo una comunicación efectiva y una comprensión compartida de los objetivos del proyecto. En este contexto, la adopción de XP se presenta como un recurso estratégico para asegurar la calidad, la eficiencia y el éxito en la ejecución de la presente investigación.

1. Entregas pequeñas: En el transcurso del proyecto, se generan versiones del sistema, aunque estas versiones pueden carecer de todas las funcionalidades planificadas. No obstante, las versiones entregadas representan resultados de valor sustancial para la organización, siguiendo una política de entregas que no excede el plazo de tres meses. Este enfoque permite proporcionar a la entidad beneficiaria incrementos periódicos de funcionalidad, garantizando así una respuesta continua a sus necesidades y prioridades, y facilitando la adaptación del sistema a medida que se desarrolla. La estrategia de entrega en ciclos cortos, característica de este proceso, optimiza la capacidad de respuesta del proyecto a los cambios en los requisitos y contribuye a la pronta obtención de resultados tangibles y beneficios para la organización.

2. Juego de planificación: Los resultados esperados de la presente investigación se definen como los objetivos a alcanzar dentro de los plazos establecidos, así como los pasos específicos que se seguirán para lograrlos. En el desarrollo de la investigación, se ha identificado una lista de funcionalidades del sistema, agrupadas en historias de usuario detalladas que proporcionan una comprensión profunda del problema que la solución abordará. Asimismo, se han fijado los plazos de entrega y realizado una estimación de los esfuerzos que el equipo de desarrollo deberá llevar a cabo, dividiéndolos por iteraciones de acuerdo con la prioridad previamente establecida en colaboración con el cliente. Estos resultados, alineados con una planificación cuidadosa y una colaboración continua, constituyen la base para el éxito del proyecto y la consecución de los objetivos propuestos.
3. Diseño simple: La metodología XP ha sido concebida con la premisa de ser simple y eficiente, centrada en la entrega de lo estrictamente necesario. En el contexto de este estudio, se ha considerado de manera particular la incorporación de patrones de diseño y arquitectónicos como parte integral de la implementación de la metodología. Estos patrones, cuidadosamente seleccionados, aportan una estructura coherente al desarrollo del sistema, permitiendo abordar desafíos recurrentes con soluciones probadas y promoviendo la reutilización de buenas prácticas. La inclusión de tales elementos en el marco de la metodología XP busca optimizar la calidad del software resultante, así como facilitar la adaptación ágil a los requisitos cambiantes del proyecto.
4. Programación en pareja: En el contexto de la Programación Extrema (XP), la ejecución de código se lleva a cabo de manera simultánea mediante la participación activa de dos programadores que se encuentran físicamente presentes. Este enfoque, conocido como programación a dos manos, implica una colaboración estrecha entre los desarrolladores, quienes trabajan conjuntamente en la elaboración y revisión del código. La programación simultánea facilita la detección temprana de posibles errores, fomenta la generación de soluciones más robustas y promueve un intercambio continuo de conocimientos y perspectivas entre los miembros del equipo. Este método, inherente a la filosofía XP, busca mejorar la eficiencia del desarrollo al aprovechar las fortalezas y experiencias individuales de los programadores, contribuyendo así al logro de los objetivos del proyecto de manera colaborativa y ágil.
5. Desarrollo guiado a prueba: Se llevan a cabo pruebas automatizadas tanto en unidades aisladas como en el sistema integral final como parte integral del proceso de desarrollo. Esta práctica tiene como objetivo verificar la funcionalidad y cohesión de las diferentes

partes del software, garantizando que estas cumplan con los requerimientos iniciales establecidos. La automatización de las pruebas facilita una evaluación exhaustiva y repetible del software, identificando posibles errores de manera eficiente y permitiendo su corrección temprana en el ciclo de desarrollo. Este enfoque contribuye significativamente a la calidad del producto final, asegurando que el software mantenga su integridad y continúe cumpliendo con los estándares y especificaciones definidos desde las etapas iniciales del proyecto.

6. Integración continua: Se llevará a cabo la integración regular de las secciones finalizadas del proyecto como medida preventiva para evitar problemas y abordar cualquier conflicto que pueda surgir durante el proceso. Esta práctica sistemática de integración busca garantizar la coherencia y la compatibilidad entre las distintas partes del proyecto en desarrollo. Al incorporar con frecuencia las secciones completadas, se facilita la detección temprana de posibles incompatibilidades y se propicia una resolución ágil de conflictos, contribuyendo así a mantener la integridad del sistema en evolución. Este enfoque proactivo no solo minimiza los riesgos asociados con la integración final, sino que también promueve una mayor eficiencia en el desarrollo al abordar de manera oportuna cualquier desafío que pueda surgir durante el avance del proyecto.
7. Estándares de codificación: Los equipos que aplican la metodología XP se adhieren a un estándar riguroso en el desarrollo de software. En este contexto, se establece la práctica de codificar de manera uniforme, con el objetivo de lograr una coherencia visual en el código, de modo que parezca haber sido escrito por un único desarrollador. Esta uniformidad en el estilo de codificación no solo optimiza la legibilidad y comprensión del código entre los miembros del equipo, sino que también facilita la colaboración y el intercambio de responsabilidades. La adopción de un estándar común contribuye a minimizar la complejidad y a mantener la consistencia a lo largo del desarrollo del proyecto, aspectos clave en la búsqueda de la eficiencia y calidad del software producido por el equipo de XP.

Conclusiones Parciales

En este capítulo, se ha establecido una sólida base teórica que sustenta la investigación en curso. Se han definido conceptos clave relacionados con el tema, se han explorado tecnologías y tendencias actuales, y se ha seleccionado una metodología de desarrollo adecuada, por lo tanto, se arriba a las siguientes conclusiones:

- Se realizó un análisis de la situación actual y las perspectivas de las soluciones de gestión de inventarios disponibles en el mercado nacional. Se destacaron las tendencias y tecnologías más comunes en el desarrollo de sistemas de este tipo. Este análisis reveló la falta de sistemas específicamente diseñados para las tiendas de materiales de construcción en el mercado actual.
- Se definieron conceptos fundamentales relacionados con la gestión documental, el sistema de gestión documental, el inventario físico y la gestión de inventarios y control de inventario. Se estableció que, para los fines de este trabajo de diploma, la gestión y el control del inventario se consideran un solo concepto debido a la naturaleza particular del inventario en estas tiendas.
- Se identificaron soluciones existentes en el mercado, como ASSETS, Condor, CS-Almacenes, Versat-Sarasola y Sorolla2, que ofrecen funcionalidades útiles para la gestión de inventarios en tiendas de materiales de construcción.
- Se presentaron las herramientas y tecnologías seleccionadas para el desarrollo de la solución, que incluyen Microsoft Visual Studio Code, Visual Paradigm, Python, HTML 5, CSS 3, JavaScript, Bootstrap 5 y Django 4.2.5.
- La metodología de desarrollo seleccionada para este proyecto es la Programación Extrema (XP), debido a su capacidad de adaptación a cambios en los requisitos, la participación activa del cliente y la adecuación para proyectos pequeños y de corto plazo. Las Historias de Usuario desempeñarán un papel fundamental en la definición de los requisitos y la planificación de las iteraciones.

Capítulo 2: Planificación y diseño de la propuesta de solución

En el presente capítulo, se examinan los elementos fundamentales asociados con las especificidades de la solución propuesta. Se describen de manera tanto los requisitos funcionales como los no funcionales que deben ser cumplidos por la solución, mientras se abordan el estilo arquitectónico y los patrones de diseño con el objetivo de asegurar la adhesión a buenas prácticas en la concepción y ejecución del sistema. Asimismo, se emplean los artefactos clave de ingeniería de software propuestos por la metodología utilizada.

2.1 Descripción de la propuesta de solución

La propuesta de solución tiene como objetivo principal optimizar el proceso de gestión de inventario en las tiendas de materiales de construcción ubicadas en La Habana. Los responsables de la administración, desempeñando la función de administradores, asumirán la tarea de mantener y garantizar el correcto funcionamiento del sistema, incluyendo la gestión de usuarios y la administración de permisos. Por otro lado, los usuarios designados como almacenistas desempeñarán un papel clave al registrar productos en el sistema, gestionar proveedores y generar informes de inventario. La interfaz del sistema se caracterizará por ser intuitivita, diseñada para ser fácilmente memorable para los usuarios y presentar una carga de información moderada. Asimismo, se implementarán funcionalidades de asistencia destinadas a mejorar la experiencia de aquellos que se beneficiarán de esta solución.

2.2 Fase de planeación

En la fase de planificación dentro de la metodología XP, se realiza un meticuloso proceso dirigido a establecer las bases sólidas para el desarrollo del software. A través de la definición clara de los objetivos y requisitos, se busca garantizar una comprensión compartida entre los miembros del equipo. La inclusión activa del cliente en este proceso, así como la utilización de artefactos específicos como el plan de iteraciones, el plan de entregas y las Historias de Usuario, son fundamentales para dar forma a una estrategia de desarrollo eficaz y centrada en la entrega de valor. En esta fase crucial, se sientan las bases para un proyecto ágil, orientado a resultados y adaptable a los cambios, aspectos intrínsecos a la filosofía de la Programación Extrema.

2.2.1 Funcionalidades del sistema

No	Nombre	Prioridad	Complejidad	Descripción

1	Autenticar Usuario	Media	Baja	El usuario deberá ingresar las credenciales correctamente para autenticarse en el sistema.
2	Registrar Usuario	Media	Baja	El sistema deberá permitir al administrador registrar una nueva cuenta de usuario.
3	Modificar Usuario	Media	Baja	El sistema permitirá al administrador modificar los datos de la cuenta de usuario.
4	Visualizar Usuario	Media	Baja	El sistema permitirá al administrador visualizar las cuentas de los usuarios.
5	Eliminar Usuario	Media	Baja	El sistema permitirá al administrador eliminar una cuenta de usuario.
6	Insertar producto	Alta	Media	El sistema permitirá al usuario insertar los datos del producto en el sistema.
7	Modificar producto	Media	Media	El sistema permitirá al usuario modificar los datos de los productos en caso de algún error o de algún rol.
8	Buscar producto	Alta	Media	El sistema permitirá al usuario buscar algún producto.
9	Eliminar producto	Alta	Media	El sistema permitirá al usuario eliminar cualquier producto.
10	Visualizar productos	Alta	Media	El sistema permitirá al usuario visualizar los productos.
11	Filtrar producto	Alta	Media	El sistema permitirá el usuario filtrar los productos.
12	Insertar proveedor	Media	Baja	El sistema permitirá al usuario insertar los datos del proveedor en el sistema.

13	Modificar proveedor	Media	Baja	El sistema permitirá al usuario modificar los datos de los proveedores en caso de algún error o de algún rol.
14	Eliminar proveedor	Media	Baja	El sistema permitirá al usuario eliminar cualquier proveedor.
15	Buscar proveedor	Media	Baja	El sistema permitirá al usuario buscar algún proveedor.
16	Visualizar proveedores	Media	Baja	El sistema permitirá al usuario visualizar los proveedores.
17	Exportar datos	Media	A	Se podrá exportar datos en formato Excel
18	Visualizar reporte del almacén	Alta	Media	El sistema permitirá al usuario visualizar el reporte del almacén.

Tabla 2 Tabla de funcionalidades

2.2.2 Historias de Usuario

En este proceso, se define el orden y la prioridad de cada historia de usuario, consolidándose esta información en un detallado plan de iteraciones. Paralelamente, se elabora un plan de duración de iteración por semana, con el objetivo de proporcionar estimaciones temporales que faciliten la evaluación del tiempo de desarrollo necesario para la implementación de las historias de usuario. Este enfoque estructurado y colaborativo en la planificación se orienta a garantizar

una ejecución eficiente y coordinada del proyecto, destacando la importancia de las historias de usuario como elementos fundamentales para la consecución de los objetivos definidos.

Las Historias de Usuario se encuentran representadas en las tablas de la 4 a la 6 y en el anexo 1 que incorpora diversos campos para su detallada descripción y gestión:

- Número: Este campo refleja el número incremental cronológico asignado a cada historia de usuario.
- Nombre de la Historia de Usuario: Identifica de manera única las historias de usuario que son objeto de descripción y acuerdo entre el desarrollador y el cliente.
- Usuario: Indica el rol del usuario que ejecuta la funcionalidad descrita.
- Prioridad en el negocio: Asigna una prioridad (Alta, Medio, Bajo) a las Historias de Usuario, determinada por las necesidades de desarrollo.
- Riesgo de desarrollo: Establece la prioridad (Alta, Medio, Bajo) asociada a las posibles ocurrencias de errores durante el proceso de desarrollo de las Historias de Usuario.
- Iteración asignada: Número correspondiente a la iteración en la que se planea desarrollar la historia de usuario.
- Puntos Estimados: Representa el tiempo en semanas ideales (equivalente a 40 horas) estimado para el desarrollo de cada historia de usuario.
- Descripción: Proporciona una breve descripción que resume los aspectos esenciales de cada historia de usuario.
- Observaciones: Ofrece señalamientos o advertencias relevantes para el sistema respecto a la historia de usuario en cuestión.

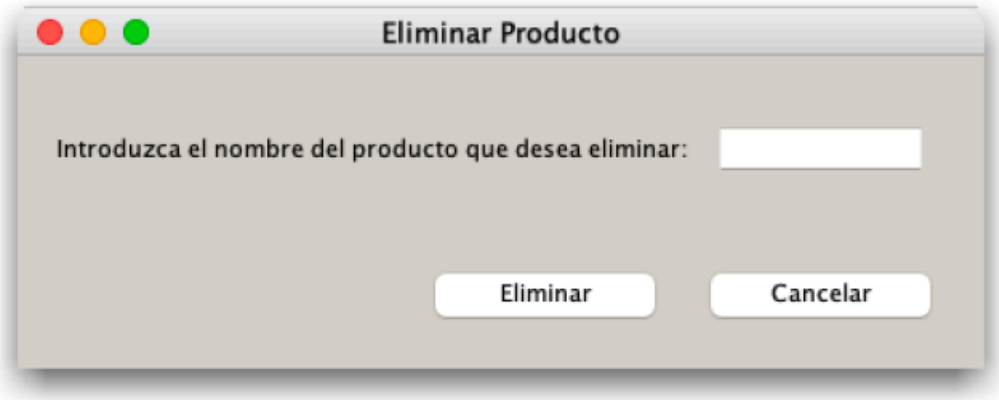
Historia de Usuario	
Número: 2	**Nombre:** Filtrar Productos
Usuario: Almacenero	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Media
Estimación: 0.4	**Iteración Asignada:** 1
Descripción: muestra al usuario el listado de productos filtrado. **Objetivo:** permite al usuario filtrar el listado de productos. **Acciones para lograr el objetivo (precondiciones y datos):** - Estar autenticado en el sistema **Flujo de acción a realizar:**	

Cuando el usuario selecciona la opción listar productos, estos se muestran en una tabla y en ella se activa el botón Filtrar, el usuario selecciona la opción por la cual desea filtrar la información (nombre, categoría, proveedor), una vez seleccionada, el sistema muestra los productos filtrados.
Prototipo de Interfaz: N/A

Tabla 3HU Filtrar Productos

Historia de Usuario	
Número: 1	**Nombre:** Gestionar Productos
Usuario: Almacenero	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Media
Estimación: 0.8	**Iteración Asignada:** 1
Descripción: Esta acción permite insertar, modificar, eliminar y visualizar la información de los productos. **Objetivo:** El almacenero podrá gestionar los productos registrados en el sistema. **Acciones para lograr el objetivo (precondiciones y datos):** - Estar autenticado en el sistema como almacenero. **Flujo de acción a realizar:** Cuando el usuario selecciona la opción Productos en el sistema, este le muestra una tabla donde visualiza la información de los productos, junto a los botones “Agregar Nuevo Producto”, “Eliminar” y “Editar”.	

Prototipo de Interfaz:



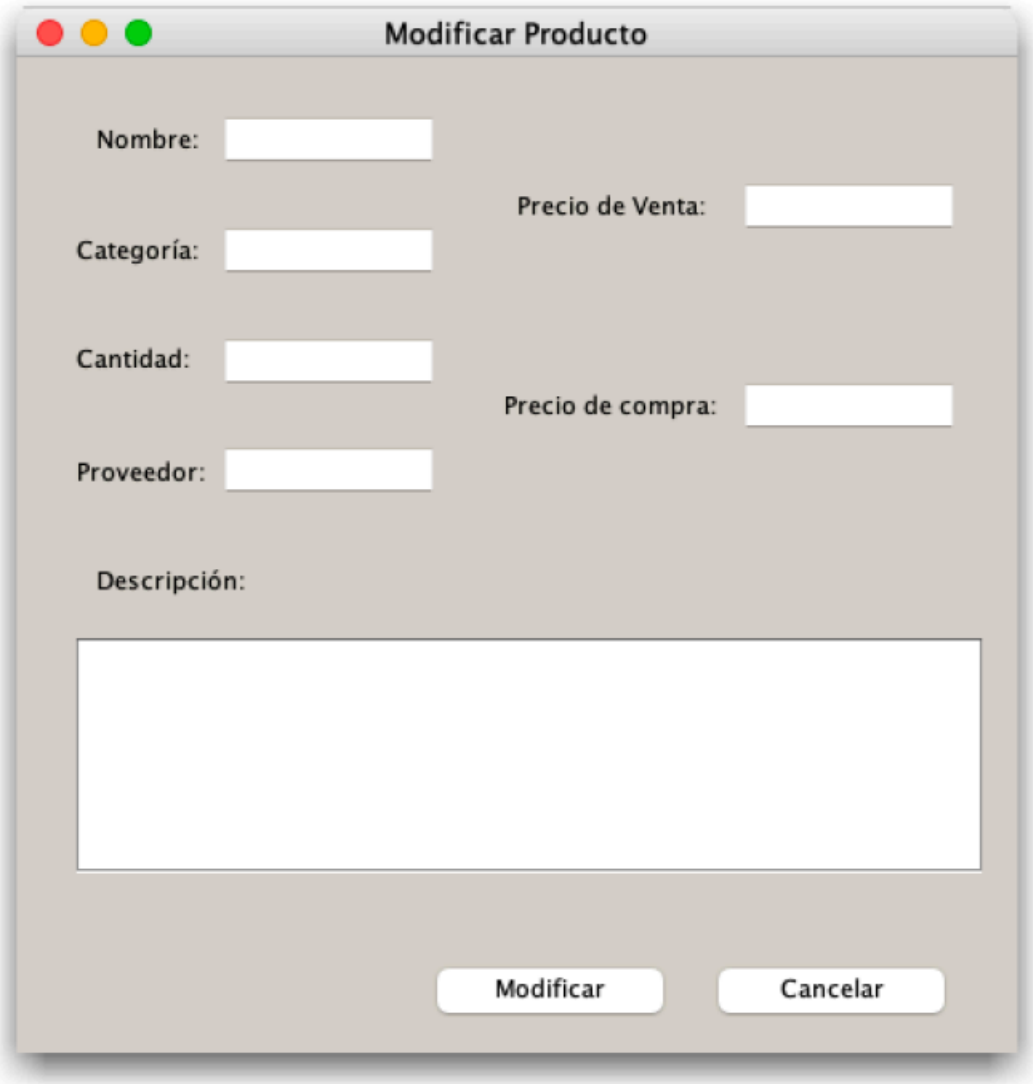


Tabla 4 HU Gestionar Productos

Historia de Usuario	
Número: 4	**Nombre:** Visualizar Reporte de Inventario
Usuario: Almacenero	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Media
Estimación: 1	**Iteración Asignada:** 1

Descripción: muestra al usuario un reporte de inventario.

Objetivo: permite al usuario visualizar un reporte sobre los productos en el sistema.

Acciones para lograr el objetivo (precondiciones y datos):

- Estar autenticado en el sistema como almacenero.

Flujo de acción a realizar:

Cuando el usuario selecciona la opción Visualizar Reporte de inventario, este le muestra la cantidad de productos disponibles, la cantidad de productos en stock y la disponibilidad de cada producto (en stock o agotado), se activa el botón Cerrar. El usuario selecciona el botón Cerrar para salir de la interfaz luego de visualizar la información.

Prototipo de Interfaz:

Tabla 5HU Visualizar Reporte de inventario

Historia de Usuario	
Número: 3	**Nombre:** Exportar Datos

Usuario: Almacenero	
Prioridad en negocio: Media	**Riesgo en desarrollo:** Alta
Estimación: 0.4	**Iteración Asignada:** 3
Descripción: muestra al usuario un documento en Excel **Objetivo:** permite al usuario exportar la información de las tablas en formato Excel. **Acciones para lograr el objetivo (precondiciones y datos):** - Estar autenticado en el sistema - Haber listado los productos, los proveedores o visualizado el reporte de inventario. **Flujo de acción a realizar:** Cuando el usuario selecciona la opción Visualizar Reporte de inventario, listar productos o listar registros, este le muestra la información correspondiente y se activa el botón Exportar. El usuario selecciona el botón y exporta la información en formato Excel.	
Prototipo de Interfaz: N/A	

Tabla 5 HU Exportar Datos

2.2.3 Estimación de esfuerzo

La programación utilizando la metodología XP basa sus procesos de planificación en estimaciones temporales de las historias de usuario, las cuales deben ser realizadas por los desarrolladores durante las diversas reuniones de planificación.

Una historia de usuario debe ser lo suficientemente pequeña como para que el equipo la desarrolle durante una entrega de una a tres semanas; más de tres semanas implica que debe ser señalar al cliente que debe dividir una historia de usuario y menos de una semana implica que la historia de es demasiado sencilla y por lo que se deben unir dos o más de ellas para su mejor interpretación.

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto de estimación. Un punto de estimación, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos de estimación. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos de iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. En la siguiente tabla de estimación de esfuerzo de las historias de usuario:

Historia de usuario	Estimación de esfuerzo

Autenticar usuario	0.2
Gestionar Usuarios	0.8
Gestionar Productos	0.8
Gestionar Proveedor	1
Filtrar Productos	0.4
Exportar datos	0.4
Visualizar reporte de inventario	1

Tabla 6Estimación de esfuerzo por HU

Se procedió a agrupar un total de siete Historias de Usuario (HU) distribuidas en tres iteraciones. La determinación del número de iteraciones se fundamentó en dos indicadores fundamentales: la experiencia del equipo de desarrollo y la adaptación permitida por la planificación, guiada por los límites establecidos por el alcance del sistema. La experiencia acumulada por el equipo brindó perspectivas valiosas para estructurar un cronograma coherente y eficiente. Simultáneamente, la planificación se ajustó de manera flexible, considerando las necesidades específicas del proyecto y garantizando que la distribución de las HU en las iteraciones se alineara de manera óptima con los objetivos y restricciones previamente establecidos. Este enfoque estratégico, respaldado por la experiencia y la planificación cuidadosa, contribuyó a la optimización del proceso de desarrollo y a la consecución exitosa de las metas propuestas.

2.2.4 Plan de Iteraciones

En la metodología XP, la creación del sistema se divide en iteraciones. La duración ideal de una iteración está entre una y tres semanas. Para cada una de las iteraciones el cliente establece un conjunto de HU que serán implementadas en cada iteración del sistema. Al final de cada iteración se realizan las pruebas de aceptación y la aplicación tendrá implementadas funcionalidades para dar cumplimiento a los objetivos propuestos (Molina, 2013).

La estructura modular de la metodología XP, basada en estas iteraciones cortas y centradas en objetivos concretos, contribuye a una adaptación ágil a los cambios en los requisitos del proyecto y a la consecución progresiva de los objetivos propuestos.

Después de la identificación y descripción de las Historias de Usuario (HU), así como la estimación detallada de los esfuerzos necesarios para su implementación, se procede a la elaboración del plan de iteraciones. Este plan se configura como un documento estratégico que detalla la secuencia y distribución de las HU que serán implementadas en cada iteración subsiguiente. Su finalidad es optimizar el desempeño del equipo de desarrollo al proporcionar una guía clara y estructurada para la ejecución de las tareas planificadas. Al vincular las HU a

las iteraciones específicas, el plan facilita una visión global del progreso del proyecto y asegura una asignación eficiente de recursos, permitiendo al equipo abordar de manera focalizada las metas establecidas en cada etapa del desarrollo. Este enfoque estratégico contribuye a la mejora continua del rendimiento del equipo y a la consecución exitosa de los objetivos planteados.

Iteración 1: En la primera iteración, se llevará a cabo la implementación de las Historias de Usuario (HU) que han sido clasificadas con un nivel de prioridad Alta en el proceso de extracción de información para la base de datos. Esta fase del desarrollo se enfoca en aspectos críticos y prioritarios, garantizando la correcta integración de funcionalidades esenciales. Además, se proporcionará al cliente la primera versión del sistema, brindándole la oportunidad de verificar directamente algunas de las funcionalidades descritas. Esta entrega inicial, realizada por el equipo de desarrollo, permitirá al cliente obtener una visión tangible de los avances y validar de manera práctica la implementación de características clave. Este enfoque iterativo, centrado en la prioridad y la validación temprana, refleja la orientación eficiente de la metodología hacia la entrega incremental de funcionalidades significativas.

Iteración 2: Durante la actual iteración, se procederá a la implementación de las Historias de Usuario (HU) catalogadas con un nivel de prioridad media. Estas historias estarán directamente vinculadas a la etapa intermedia del desarrollo del sistema, abordando funcionalidades de menor complejidad. En esta fase, se dedicará especial atención a la revisión y corrección de errores identificados en la iteración anterior. La priorización basada en la complejidad y la corrección de errores previos responde a la filosofía iterativa de la metodología, garantizando una mejora constante y un avance sistemático hacia los objetivos establecidos.

Iteración 3: En la presente iteración, se llevará a cabo la implementación de las Historias de Usuario (HU) identificadas con prioridad baja, marcando así la culminación del proceso de desarrollo del sistema. Esta etapa se traducirá en la obtención de la primera versión del sistema informático diseñado para la gestión de inventarios en las tiendas de venta de materiales de construcción. En este punto, se brindará al cliente la oportunidad de realizar una revisión a fondo de todas las funcionalidades y procesos programados. Además, se le invitará a poner a prueba el sistema, lo que permitirá validar directamente su desempeño y proporcionar valiosos comentarios para posibles ajustes y mejoras. La entrega de esta primera versión marca un hito significativo en el desarrollo del proyecto, alineándose con la metodología iterativa al proporcionar resultados tangibles y la posibilidad de iterar sobre el sistema para su perfeccionamiento continuo.

Iteración	Historias de Usuario	Duración total de las iteraciones
1	Gestionar Productos	2 semanas
	Filtrar Productos	
	Visualizar reporte de inventario	
2	Gestionar Usuarios	2 semanas
	Gestionar Proveedor	
3	Autenticar usuario	1 semana
	Exportar datos	

Tabla 7Plan de Iteraciones

2.2.5 Plan de Entrega

En el marco del plan de entregas en la metodología XP, se elabora un cronograma que detalla las fechas y prioridades de entrega establecidas por el cliente. En este proceso, el cliente determina la prioridad de cada Historia de Usuario (HU), así como la agrupación y secuencia en que se llevarán a cabo. De manera complementaria, los programadores proceden a realizar una estimación del esfuerzo requerido para la implementación de cada historia planificada. Este enfoque estructurado y colaborativo busca establecer una clara hoja de ruta para la entrega incremental del software, permitiendo al cliente influir activamente en la priorización y disposición de las funcionalidades durante el desarrollo del proyecto.

Entregable	Versión	Fecha
1ra Iteración	1.0	13/10/23
2da Iteración	1.0	27/10/23
3ra Iteración	1.0	03/11/23

Tabla 8Plan de Entregas

2.3 Fase de Diseño

En XP se diseñan aquellas historias de usuario que le cliente ha seleccionado para la integración actual por dos motivos: por un lado, se considera que no es posible tener un diseño completo del sistema y sin errores desde el principio. El segundo motivo es que, dada la naturaleza cambiante del proyecto, el hacer un diseño muy extenso en fases iniciales del proyecto para luego modificarlo, se considera una pérdida de tiempo. Es importante resaltar que esta tarea es permanente durante la vida del proyecto partiendo del diseño inicial que va corrigiendo y mejorando en el transcurso del proyecto (ASANA, 2022).

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar (Bustamante, 2014).

2.3.1 Tarjetas TRC

Las Tarjetas CRC (Clase-Responsabilidades-Colaboradores), permiten conocer que clases componen el sistema y cuales interactúan entre sí. Se dividen en tres secciones: Nombre de la Clase, Responsabilidades y Colaboradores.

Tarjeta CRC	
Nombre de la clase: Producto	
Responsabilidades: - Almacenar información de los productos	**Colaboradores:** - Usuario - Proveedor

Tabla 9Tarjeta CRC #1

Tarjeta CRC	
Nombre de la clase: Usuario	
Responsabilidades: - Almacenar información de los usuarios	**Colaboradores:** - Producto

Tabla 10Tarjeta CRC #2

Tarjeta CRC	
Nombre de la clase: Proveedor	
Responsabilidades: - Almacenar información de los proveedores	**Colaboradores:** - Producto

Tabla 11Tarjeta CRC #3

2.3.2 Patrón arquitectónico

Los patrones arquitectónicos, o patrones de arquitectura, también llamados arquetipos, ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. En comparación con los patrones de diseño, los patrones arquitectónicos tienen un nivel de abstracción mayor (Pressman, 2014).

Django se basa en la arquitectura MVT(Modelo-Vista-Plantilla). MVT es un patrón de diseño de software para desarrollar una aplicación web.

La estructura **MVT** tiene las siguientes tres partes:

Modelo: El modelo se considera como la fuente única y definitiva de los datos que posee la aplicación desarrollada, y que permiten ejecutar las operaciones. En la arquitectura MVT se establece que el modelo se encuentra en forma de clases, por lo que Python puede indicar y controlar el comportamiento de los datos almacenados (Hidalgo et al., 2020).

Vista: La vista se considera como el elemento de la aplicación que contiene la lógica de negocio; necesaria para devolver una respuesta hacia el cliente que la solicita, también procesan las peticiones o solicitudes que accederán al modelo para poder entregar u obtener los datos. La sección de vista del modelo MVT en Django se presenta en forma de clases y funciones en Python, en el que se determinan los datos que serán visualizados entre otras cosas, además, permite escribir código Python en lugar de instrucciones SQL (Idioma de consulta de imagen de estudio) para realizar las consultas que se requieran en la vista; determinando que la vista no se responsabiliza del estilo para presentar los datos sino se encarga de los datos en sí (Hidalgo et al., 2020).

Plantilla: En la plantilla se debe decidir la forma en la que se presentarán los datos devueltos por la vista en el navegador web; utilizando estilos CSS o brindando dinamismo a través de JavaScript. Es importante recalcar que las etiquetas utilizadas por Django, permiten que sea flexible para los diseñadores del front-end, estas estructuras son limitadas para evitar un desorden poniendo cualquier tipo de código escrito en Python, con el objetivo de que la lógica del sistema siga permaneciendo en la vista (Hidalgo et al., 2020).

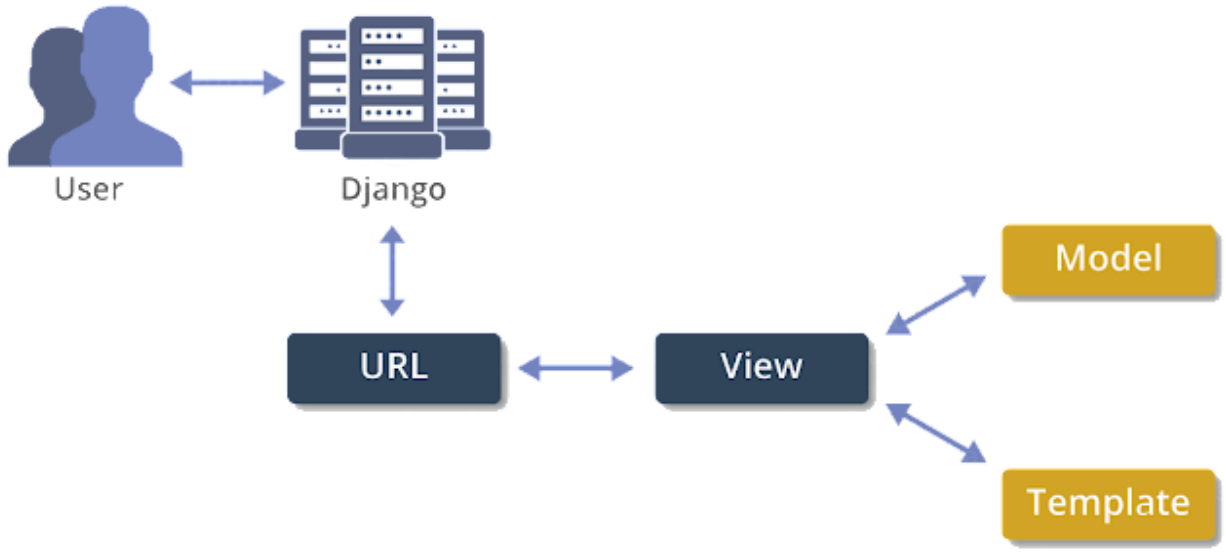


Figura 1 *FUNCIONAMIENTO DE MVT EN DJANGO(HIDALGO ET AL., 2020)*

2.3.3 Patrones de diseño

Un patrón de diseño puede ser caracterizado como una regla de tres partes, la cual expresa una relación entre cierto contexto, un problema y una solución. Para el diseño de software, el contexto permite a quien usa el patrón entender el entorno en el cual reside el problema y qué solución pudiera ser apropiada dentro de ese entorno. Un conjunto de requisitos, incluyendo limitaciones y restricciones, actúan como un sistema de fuerzas que influencia en cómo el problema puede ser interpretado dentro de su contexto y cómo la solución puede ser aplicada efectivamente (Pressman, 2014).

Patrones GRASP

Un patrón de diseño se caracteriza como “una regla de tres partes que expresa una relación entre cierto contexto, un problema y una solución”. Para el diseño de software, el con texto permite al lector entender el ambiente en el que reside el problema y qué solución sería apropiada en dicho ambiente. Un conjunto de requerimientos, incluidas limitaciones y restricciones, actúan como sistema de fuerzas que influyen en la manera en la que puede interpretarse el problema en este contexto y en cómo podría aplicarse con eficacia la solución (Pressman, 2014).

Bajo acoplamiento: debe haber pocas dependencias entre las clases. El propósito de este patrón es aumentar la reutilización y eliminar las dependencias entre las clases para propiciar un

fácil mantenimiento y entendimiento (Larman, 2003). En una aplicación de Django, la definición de la URL y la función de vista que se llamará están débilmente acopladas.

```
Tienda > Tienda > urls.py
17  from django.contrib import admin
18  from django.urls import path
19  from Tareas import views
20
21  urlpatterns = [
22      path('admin/', admin.site.urls),
23      path('home/',views.inicio, name="Home"),
24      path('home/productos/',views.productos, name="productos"),
25      path('home/productos/insertar/',views.InsertarProducto, name='Inser
26      path('home/productos/editar/', views.EditarProducto, name="EditarPr
27      path('home/productos/eliminar/<codigo>', views.EliminarProducto),
28      path('home/productos/eliminar_productos/', views.eliminar_productos
29      path('home/registros/',views.registros, name="registros"),
30      path('home/ventas',views.ventas, name="ventas"),
31      path('home/ventas/insertarventa/',views.InsertarVenta, name="insert
32      path('home/insertarventa/',views.InsertarVenta, name="V"),
33
```

Figura 2 *Captura de fragmento de código del archivo urls.py*

Experto: Este patrón define la clase experta en información, dado que esta es la que contiene toda la información para cumplir con la responsabilidad. De este modo se obtendrá un diseño con mayor cohesión y así la información se mantiene encapsulada. (Larman, 2003) Con el uso del *Object Relational Mapping* (ORM) de Django, se crean las clases que representan las entidades del modelo de datos. Asociado a cada una de estas clases, son generadas un conjunto de funcionalidades que las relacionan de forma directa con la entidad que representan. Las clases contienen toda la información necesaria de la tabla que representan en la base de datos.

urls.py U models.py U ×

Tienda > Tareas > models.py

```python
from django.db import models

class Producto (models.Model):
    codigo = models.CharField(max_length=12, primary_key=True)
    nombre = models.CharField(max_length=40)
    descripcion = models.CharField(max_length=255)
    categoria = models.CharField(max_length=25)
    precioCompra = models.FloatField()
    precioVenta = models.FloatField()
    proveedor = models.CharField(max_length=25)
    nivelStock = models.PositiveIntegerField()

```

Figura 3 *Fragmento de código de models.py*

Patrones GOF

Publicados por Gamma, Helm, Johnson y Vlossodes en 1995: patrones de la banda de los cuatro (del inglés, Gang of Four). Esta serie de patrones permiten ampliar el lenguaje, aprender nuevos estilos de diseño y además introducir más notación UML. Los patrones de diseño del grupo GoF se clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento (Larman, 2003).

Front Controller es el primer punto de entrada para la solicitud. Maneja el envío general al controlador y se ocupa de las necesidades comunes tales como, autenticación, gestión de sesiones, seguridad, redirección. Contiene la lógica que todas las solicitudes deben considerar, reduciendo la duplicación de código (Larman, 2003). También agrega datos al contexto de ejecución. Se evidencia, por ejemplo, en la siguiente función:

```
def autenticar(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username, password=password)

        if user is not None:
            if user.is_superuser:
                # Si el usuario es un superusuario, redirigir al panel de administración
                login(request, user)
                return redirect('admin:index')
            else:
                # Si el usuario no es un superusuario, redirigir a otra página
                login(request, user)
                return redirect('inicio')
        else:
            messages.error(request, 'Credenciales inválidas. Inténtalo de nuevo.')

    return render(request, 'login.html')
```

Figura 4 *Utilización del patrón front-controller*

Se puede observar autenticación, gestión de sesiones, seguridad y redirección.

- Se verifica si el método de la solicitud HTTP es 'POST'.
- Si la solicitud es de tipo 'POST', se extraen los datos del formulario. request.POST es un diccionario que contiene los datos enviados por el usuario a través del formulario, y en este caso, se extraen el nombre de usuario (username) y la contraseña (password).
- Luego, se utiliza la función authenticate de Django para intentar autenticar al usuario. Esta función toma el objeto request, el nombre de usuario y la contraseña y devuelve un objeto de usuario si las credenciales son válidas, o None si no lo son.
- Se verifica si el objeto de usuario retornado por authenticate no es None, lo que significa que las credenciales son válidas y se ha autenticado al usuario.
- Si el usuario es válido y además es un superusuario (administrador), se autentica al usuario utilizando la función login de Django y se redirige al panel de administración ('admin:index'). La función login registra la sesión del usuario como autenticada.
- Si el usuario es válido, pero no es un superusuario, también se autentica y se redirige a otra página, en este caso, 'inicio'.
- Si las credenciales no son válidas y el objeto de usuario es None, se agrega un mensaje de error al objeto request utilizando messages.error. Esto le indica al usuario que las credenciales son inválidas y debe intentarlo nuevamente.
- Si la solicitud no es de tipo 'POST', es decir, una solicitud de visualización de la página de inicio de sesión, se renderiza la plantilla de la página de inicio de sesión, que en este caso se llama 'login.html'

Patrón Decorator: Añade responsabilidades adicionales a un objeto dinámicamente, proporcionando una alternativa flexible a la especialización mediante herencia, cuando se trata de añadir funcionalidades (Larman, 2003).

Por ejemplo, en la siguiente función:

```
33  @login_required
34  def salir(request):
35      logout(request)
36      return redirect('login')
37
```

Figura 5 *Utilización del patrón decorador*

2.3.4 Modelo de datos

Un modelo de datos es un conjunto de herramientas conceptuales para describir la representación de la información en términos de datos. Los modelos de datos comprenden aspectos relacionados con: estructuras y tipos de datos, operaciones y restricciones. El modelo Entidad/Relación ha tenido una gran difusión en la comunidad informática dedicada a las bases de datos, prueba de ello es que ha sido el modelo más extendido en las herramientas CASE de ayuda al diseño de base de datos. El modelo E/R puede ser usado como una base para una vista unificada de los datos, adoptando el enfoque más natural del mundo real que consiste en entidades y relaciones (Zorrilla Pantaleon, Martha E., 2011).

Después de la concepción de las tarjetas CRC y la definición de las clases de entidades, el siguiente paso consiste en elaborar el modelo de datos que el sistema empleará para la gestión de la información. El propósito principal de este modelo radica en describir las clases fundamentales que constituyen la capa modelo del sistema, así como especificar el tipo de dato de los atributos que serán almacenados en la base de datos y la naturaleza de las relaciones entre las entidades. Este modelo se erige como un marco conceptual que permite representar de manera sistemática los elementos de la realidad involucrados en el problema específico que aborda el sistema, detallando la forma en que estas entidades se interrelacionan entre sí. Este enfoque metodológico resulta esencial para establecer una base estructurada y coherente que

guiará la implementación y operación efectiva del sistema en cuestión.

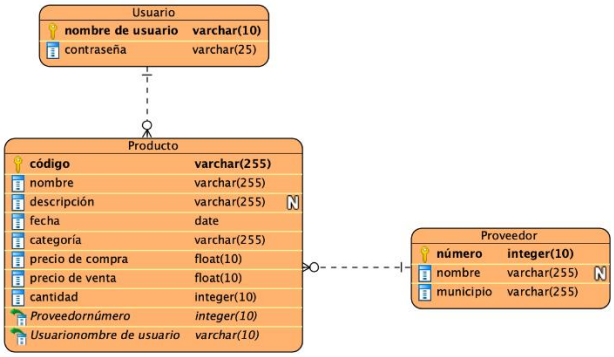


Figura 6 *Modelo de datos*

Conclusiones parciales

- Se identificaron las características clave, así como los requisitos tanto funcionales como no funcionales que son esenciales para la creación exitosa de la plataforma. Estos elementos se han organizado de manera efectiva en forma de historias de usuario, lo que proporciona una visión clara de las necesidades del proyecto.
- La adopción de una arquitectura bien definida y la aplicación de patrones de diseño específicos desempeñan un papel crucial en la simplificación y eficiencia del proceso de desarrollo de la solución propuesta. Estos enfoques arquitectónicos y de diseño establecidos permiten una implementación más coherente y robusta, asegurando que el portal cumpla con los objetivos planteados.
- Los artefactos generados en este capítulo, como las historias de usuario y las directrices arquitectónicas, servirán como una guía esencial durante la fase de implementación. Proporcionarán un marco sólido y orientación práctica para el equipo de desarrollo, lo que aumentará la eficiencia y la calidad del producto final.

Capítulo 3: Implementación y prueba de la propuesta de solución

El capítulo actual detalla la fase de implementación y prueba del sistema, que constituye una parte integral del proceso de construcción destinado a garantizar su entrega y operación efectiva. Siguiendo la metodología XP, este proceso se sitúa después de la fase de diseño, con el propósito de realizar análisis y evaluaciones continuas durante el desarrollo. Se llevan a cabo pruebas unitarias exhaustivas para verificar el funcionamiento preciso de la aplicación, y se realizan pruebas de aceptación para asegurar que el producto final coincida con las expectativas del cliente. Este enfoque iterativo, característico de la metodología XP, contribuye a la identificación temprana de posibles problemas y a la consecución de un sistema robusto y alineado con los requisitos y expectativas del usuario final.

3.1 Implementación

3.2.1 Diagrama de despliegue

Los diagramas de despliegue se utilizan para visualizar los procesadores/ nodos/dispositivos de hardware de un sistema, los enlaces de comunicación entre ellos y la colocación de los archivos de software en ese hardware. Un diagrama de despliegue es un tipo de diagrama UML que muestra la arquitectura de ejecución de un sistema, incluyendo nodos como entornos de ejecución de hardware o software, y el middleware que los conecta.

Los diagramas de despliegue se utilizan normalmente para visualizar el hardware y el software físico de un sistema. Usándolo puedes entender cómo el sistema se desplegará físicamente en el hardware. Los diagramas de despliegue ayudan a modelar la topología de hardware de un sistema (Siriwardhana, 2020).

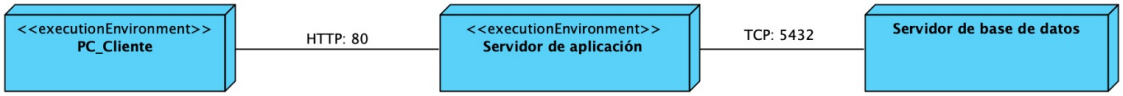


Figura 7 *Diagrama de Despliegue*

Descripción de los Nodos

• Servidor de Base de Datos: Es el nodo donde se almacenan los datos del sistema mediante el sistema de Base de Datos PostgreSQL 14.

• Servidor de Aplicaciones: Es el nodo encargado de tener instalado el sistema al que tendrán acceso los usuarios desde las estaciones de trabajo, con el framework Django y el sistema de Base de Datos PostgreSQL 14.

• PC Cliente: Es el nodo que representa la estación de trabajo que permite al usuario mediante el protocolo HTTP: 80 y el TCP: 5432, utilizando los navegadores web para acceder al sistema y la impresora para otras funciones.

3.2.2 Estándares de codificación

Los estándares de codificación o programación son términos que describen convenciones para escribir código fuente en los lenguajes de programación. Garantizan un mantenimiento óptimo del código por parte de los programadores, independientemente de si son los creadores del mismo. Relejan un estilo armonioso como si un único programador hubiera escrito todo el código de una sola vez. La confección de estos estándares debe ser definida al comienzo de la implementación para garantizar que todos los programadores trabajen de manera coordinada En este caso se utilizó el estándar de codificación de Django.

Grosor de línea: Cada línea de código no debe exceder los 80 caracteres en la medida de lo posible (en circunstancias especiales, puede exceder ligeramente los 80, pero la más larga no puede exceder los 120)

```
class Producto (models.Model):
codigo = models.CharField(max_length=12, primary_key=True)
nombre = models.CharField(max_length=40)
descripcion = models.CharField(max_length=255)
categoria = models.CharField(max_length=25)
precioCompra = models.FloatField()
precioVenta = models.FloatField()
proveedor = models.CharField(max_length=25)
nivelStock = models.PositiveIntegerField()
```

Comillas: En pocas palabras, el lenguaje natural usa comillas dobles y las etiquetas de máquina usan comillas simples, por lo que la mayor parte del código debe usar comillas simples.

```
            response_data = {'success': True, 'message': 'Productos eliminados exitosamente.'}
        except Exception as e:
            response_data = {'success': False, 'error_message': str(e)}

        return JsonResponse(response_data)
    else:
        return JsonResponse({'success': False, 'error_message': 'Método no permitido'})
```

Declaración de importación: La declaración de importación debe escribirse en líneas separadas.

```
from django.shortcuts import render, redirect, get_object_or_404
from django.core.paginator import Paginator
from django.http import JsonResponse
from django.contrib import messages
from datetime import datetime
from .models import *
import json
```

Espacio: Un espacio a cada lado del operador binario [=, -, +=, ==,>, in,is not, and]. En la lista de parámetros de la función, no agregue espacios a ambos lados del signo igual predeterminado.

```
if request.method == 'POST':
        try:
            codigos = request.POST.getlist('codigos[]')
            for codigo in codigos:
                producto = Producto.objects.get(codigo=codigo)
                producto.delete()
```

- En la lista de parámetros de la función, debe haber un espacio después.

```
def EliminarProducto(request, codigo):
```

3.2.3 Tareas de ingeniería

Una Historias de Usuario se descompone en varias tareas de ingeniería, las cuales describen las actividades que se realizarán en cada historia de usuario, así mismo las tareas de ingeniería se vinculan más al desarrollador, ya que permite tener un acercamiento con el código (Ferreira Escutia, 2014). El resto de las tareas se encuentran en el anexo 2.

Se muestran sus respectivos campos:

1. Número de la tarea: Los números deben ser consecutivos.
2. Número de HU: Número. de la historia de usuario a la que pertenece la tarea.
3. Nombre Tarea: Nombre que identifica a la tarea.
4. Tipo de Tarea: Las tareas pueden ser de: Desarrollo, Mejora, Otra (Especificar).
5. Puntos Estimados: Tiempo estimado en días que se le asignará a su desarrollo.
6. Fecha Inicio: Fecha en que se inicia el desarrollo de la tarea.
7. Fecha Fin: Fecha en que finaliza el desarrollo de la tarea.
8. Programador Responsable: Nombre y apellidos del programador.
9. Descripción: Breve descripción de la tarea. Con los campos anteriores se muestran las siguientes tareas de ingeniería.

Tareas de la ingeniería para la iteración 1

Tarea de ingeniería	
Número de la Tarea: 1	**Número de la Historia de usuario:** 1
Nombre de la Tarea: Insertar producto	
Tipo de Tarea: Desarrollo	**Estimación:** 0.2
Fecha de inicio: 02/10/23	**Fecha de fin:** 02/10/23
Programador responsable: Jorge Saúl Pérez Peña Damisela Thalía Gómez Cudina	
Descripción: Implementar una vista que permita insertar un producto. Debe muestran en una tabla y en ella se activa el botón Filtrar, el usuario selecciona la opción por la cual desea filtrar la información (nombre, categoría, proveedor), una vez seleccionada, el sistema muestra los productos filtrados	

Tabla 11Tarea de ingeniería 1

Tarea de ingeniería	
Número de la Tarea: 2	**Número de la Historia de usuario:** 1
Nombre de la Tarea: Modificar producto	

Tipo de Tarea: Desarrollo	**Estimación:** 0.2
Fecha de inicio: 03/10/23	**Fecha de fin:** 03/10/23
Programador responsable: Jorge Saúl Pérez Peña Damisela Thalía Gómez Cudina	
Descripción: Implementar una vista que permita modificar un producto.Cuando el usuario selecciona el botón para modificar un producto, el sistema muestra un formulario con los campos a que puede modificar (código, nombre, municipio) y los botones de “Modificar” y “Cancelar”. El usuario modifica los campos y presiona “Modificar”, el sistema guarda los cambios realizados. Si selecciona la acción “cancelar” regresa a la vista previa.	

Tabla 12Tarea de ingeniería 2

Tarea de ingeniería	
Número de la Tarea: 3	**Número de la Historia de usuario:** 1
Nombre de la Tarea: Eliminar producto	
Tipo de Tarea: Desarrollo	**Estimación:** 0.05
Fecha de inicio: 04/10/23	**Fecha de fin:** 04/10/23
Programador responsable: Jorge Saúl Pérez Peña Damisela Thalía Gómez Cudina	
Descripción: Implementar una vista que permita eliminar un producto. cuando el usuario selecciona el botón para eliminar un producto, el sistema muestra un mensaje pidiendo la confirmación de la eliminación y los botones de “Eliminar” y “Cancelar”. El usuario presiona “Eliminar” y el sistema elimina el producto. Si selecciona la acción “cancelar” regresa a la vista previa.	

Tabla 13Tarea de ingeniería 3

Tarea de ingeniería	
Número de la Tarea: 4	**Número de la Historia de usuario:** 1
Nombre de la Tarea: Buscar producto	
Tipo de Tarea: Desarrollo	**Estimación:** 0.1
Fecha de inicio: 05/10/23	**Fecha de fin:** 05/10/23
Programador responsable: Jorge Saúl Pérez Peña Damisela Thalía Gómez Cudina	
Descripción: Implementar una vista que permita buscar un producto. El usuario de rol de administrador introduce el nombre del producto que desea buscar, y luego se muestra los datos del mismo.	

Tabla 14Tarea de ingeniería 4

Tarea de ingeniería	
Número de la Tarea: 5	**Número de la Historia de usuario:** 1
Nombre de la Tarea: Visualizar productos	
Tipo de Tarea: Desarrollo	**Estimación:** 0.2
Fecha de inicio: 06/10/23	**Fecha de fin:** 06/10/23
Programador responsable: Jorge Saúl Pérez Peña Damisela Thalía Gómez Cudina	
Descripción: Implementar una vista que permita listar los productos registrados en el sistema.	

Tabla 15Tarea de ingeniería 5

Tarea de ingeniería	
Número de la Tarea: 6	**Número de la Historia de usuario:** 2
Nombre de la Tarea: Filtrar productos	
Tipo de Tarea: Desarrollo	**Estimación:** 0.4
Fecha de inicio: 09/10/23	**Fecha de fin:** 10/10/23
Programador responsable: Jorge Saúl Pérez Peña Damisela Thalía Gómez Cudina	
Descripción: Implementar un botón que permita filtrar el listado de productos. Cuando el usuario selecciona la opción listar productos, estos se muestran en una tabla y en ella se activa el botón Filtrar, el usuario selecciona la opción por la cual desea filtrar la información (nombre, categoría, proveedor), una vez seleccionada, el sistema muestra los productos filtrados	

Tabla 16Tarea de ingeniería 6

Tarea de ingeniería	
Número de la Tarea: 7	**Número de la Historia de usuario:** 3
Nombre de la Tarea: Exportar Datos	
Tipo de Tarea: Desarrollo	**Estimación:** 0.6
Fecha de inicio: 11/10/23	**Fecha de fin:** 13/10/23
Programador responsable: Jorge Saúl Pérez Peña Damisela Thalía Gómez Cudina	

Descripción: Implementar un botón que permita exportar las tablas en formato Excel. Cuando el usuario selecciona la opción Visualizar Reporte de inventario, listar productos o listar registros, este le muestra la información correspondiente y se activa el botón Exportar. El usuario selecciona el botón y exporta la información en formato Excel.

Tabla 17Tarea de ingeniería 7

3.3 Pruebas de Software

Los sistemas de software desempeñan un papel fundamental e integral en las actividades cotidianas. Es imperativo tener presente que dichos sistemas o aplicaciones son concebidos e implementados por individuos, lo que implica que en cualquier fase de su desarrollo podría surgir un error o defecto. La falta de identificación de tales defectos antes de la ejecución de las aplicaciones conlleva un riesgo sustancial, ya que existe la posibilidad de que la aplicación no cumpla con su propósito previsto, generando así un fallo o desperfecto.

Las pruebas de software son procesos que permiten verificar y revelar la calidad de un producto. Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa. (Paz, 2016).

Pasos que se deben tener en cuenta para una prueba de software segura:

- ✓ Estrategia de prueba.
- ✓ Niveles de prueba.
- ✓ Tipo de prueba.
- ✓ Método de prueba.
- ✓ Caso de prueba.

3.3.1 Estrategia de pruebas

Una estrategia de pruebas de software proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos requieren. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de la prueba y la recolección y evaluación de los resultados (Pressman, 2014).

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación (Pressman, 2014). La Metodología XP propone que las pruebas de software sean realizadas al término de cada iteración, garantizando el funcionamiento deseado y la aceptabilidad por el cliente para realizar una entrega funcional y acorde a las exigencias de un producto con calidad.

3.3.2 Niveles de Prueba

El esquema de los niveles de pruebas de software permite que se analicen con mayor detalle los riesgos que un componente, aplicación o sistema puede tener, logrando un proceso de pruebas más eficaz. Esto tiene una relación directa con los tipos de pruebas de software. (Tamushi, 2023).

En la metodología XP (Extreme Programming), se enfatiza la realización de pruebas a lo largo de todo el proceso de desarrollo de software. Aunque no existen niveles de prueba específicos definidos de manera estricta como en algunas metodologías tradicionales, XP promueve varias prácticas de prueba en diferentes niveles. Estos pueden incluir:

- Pruebas Unitarias (Unit Testing): Las pruebas unitarias son pequeñas pruebas automatizadas que se centran en verificar el correcto funcionamiento de unidades individuales de código, como funciones o métodos.

- Pruebas de Aceptación del Cliente (Customer Acceptance Testing): Implica la participación del cliente para validar que las historias de usuario cumplen con sus requisitos y expectativas. Esta es una forma de prueba de aceptación continua.

- Pruebas de Integración (Integration Testing): Aseguran que los diferentes módulos o componentes del sistema funcionen correctamente cuando se integran. Pueden realizarse pruebas de integración continuas a medida que se desarrolla el software.

- Pruebas de Sistema (System Testing): Verifican que el sistema en su conjunto cumple con los requisitos y funciona según lo esperado. Pueden incluir pruebas de rendimiento, seguridad y otras pruebas no funcionales.

- Pruebas de Regresión (Regression Testing): Se realizan para garantizar que los cambios recientes no hayan introducido nuevos errores y que las funcionalidades existentes sigan siendo válidas después de las modificaciones.

- Pruebas de Humo (Smoke Testing): Verifican que las funcionalidades básicas del sistema funcionan correctamente antes de realizar pruebas más exhaustivas. Es una prueba rápida para detectar problemas evidentes.

- Pruebas Exploratorias (Exploratory Testing): Se basan en la exploración y descubrimiento del software sin seguir un plan de prueba predefinido. Los miembros del equipo exploran el sistema para identificar posibles problemas.

- Pruebas de Aceptación Automatizadas (Automated Acceptance Testing): Se automatizan las pruebas de aceptación del cliente para garantizar una ejecución rápida y consistente de las mismas.

En XP, la idea es que las pruebas sean parte integral del proceso de desarrollo, realizadas de manera continua y automatizada siempre que sea posible. Esto contribuye a la entrega de software de alta calidad de manera frecuente y rápida.

En el desarrollo de este sistema se desea trabajar de la siguiente manera:

- En el nivel de unidad, con el método de caja blanca para realizar la prueba unitaria en pequeñas partes del código para verificar su correcto funcionamiento.
- A nivel de sistema, para probar las funcionalidades del sistema a desarrollar, se desea utilizar el método de caja negra mediante las pruebas de aceptación con los casos de prueba propuestos.

3.3.3 Tipos de Pruebas

Los tipos de pruebas son un conjunto de actividades de prueba que pueden tener por objeto comprobar el sistema en base a un motivo u objetivo específico mejorando la calidad de lo que hacemos mientras estamos desarrollando software. (Morales, 2019).

Diversos tipos de pruebas de calidad de software desempeñan un papel crucial en la mejora continua del desarrollo, pudiendo clasificarse como manuales o automáticas. Algunas de estas modalidades de prueba incluyen:

- Pruebas de Funcionalidad: Este enfoque implica la evaluación de las funciones ejecutadas por el software para verificar la eficacia de su funcionalidad y rendimiento.

- Pruebas de Características No Funcionales: Dirigidas hacia aspectos que influyen en la experiencia del usuario, el diseño, la seguridad y la calidad global del sistema, estas pruebas se centran en características no funcionales.

- Pruebas de Caja Blanca: Estas pruebas, también denominadas pruebas estructurales, se concentran en evaluar la estructura interna del software, permitiendo una revisión detallada de su diseño y lógica interna.

3.3.4Pruebas unitarias

Pruebas de Caja Blanca: También suelen ser llamadas estructurales o de cobertura lógica. En ellas se pretende investigar sobre la estructura interna del código, exceptuando detalles referidos a datos de entrada o salida, para probar la lógica del programa desde el punto de vista algorítmico. Realizan un seguimiento del código fuente según se va ejecutando los casos de prueba, determinándose de manera concreta las instrucciones, bloques, etc. que han sido ejecutados por los casos de prueba (Ferreira Escutia, 2014).

Las pruebas de caja blanca se basan en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez (Viera Ipanaque, 2019).

Con las pruebas unitarias del software se comprueban los caminos lógicos del software proponiendo casos de prueba que se ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado.
Las pruebas unitarias automatizadas en Django se realizan utilizando el framework de pruebas integrado de Django, que se basa en el módulo unittest de Python. Este framework proporciona

un conjunto de clases y métodos que facilitan la creación y ejecución de pruebas unitarias para la propuesta.

```
Tienda > Tareas > tests.py
6   from django.contrib.messages import get_messages
7   from django.http import HttpResponse
8   import json
9
10  class ViewsTestCase(TestCase):
11      def setUp(self):
12          self.factory = RequestFactory()
13          self.user = User.objects.create_user(username='testuser', password='testpassword')
14
15      def test_autenticar_view(self):
16          # Prueba para la vista autenticar
17          response = self.client.post(reverse('autenticar'), {'username': 'testuser', 'password': 'testpassword'})
18          self.assertEqual(response.status_code, 302)  # Debería redirigir después de autenticarse
19
20      def test_salir_view(self):
21          # Prueba para la vista salir
22          self.client.login(username='testuser', password='testpassword')
23          response = self.client.get(reverse('salir'))
24          self.assertEqual(response.status_code, 302)  # Debería redirigir después de salir
25
26      def test_proveedores_view(self):
27          # Prueba para la vista proveedores
28          response = self.client.get(reverse('proveedores'))
29          self.assertEqual(response.status_code, 200)  # Debería devolver un código 200
30
31      def test_insertar_proveedor_view(self):
32          # Prueba para la vista InsertarProveedor
33          response = self.client.post(reverse('insertar_proveedor'), {'txtnombre': 'ProveedorTest', 'txtmunicipio':
34          self.assertEqual(response.status_code, 302)  # Debería redirigir después de insertar un proveedor
35
36      def test_editar_proveedor_view(self):

PROBLEMAS   SALIDA   CONSOLA DE DEPURACIÓN   PUERTOS   TERMINAL        bash - Tienda

----------------------------------------------------------------------
Ran 10 tests in 6.037s

OK
Destroying test database for alias 'default'...
```

Figura 8 *Fragmento de código del archivo de pruebas unitarias automatizadas*

Las pruebas unitarias realizadas han proporcionado resultados altamente satisfactorios, destacando la estabilidad y consistencia del código subyacente. La configuración inicial, gestionada a través de la función `setUp`, ha asegurado un entorno de prueba controlado y predecible para cada escenario evaluado. La eficiencia en el manejo de errores fue evidente, ya que todas las excepciones se gestionaron sin afectar la integridad del sistema. Las pruebas de vistas y URL confirmaron la corrección de las rutas definidas y su capacidad para responder adecuadamente a las solicitudes. Las aserciones utilizadas para verificar los resultados fueron efectivas, asegurando la coherencia entre los resultados obtenidos y los esperados. Además, las pruebas abordaron aspectos de seguridad y acceso, garantizando la protección adecuada de las vistas y el acceso apropiado para usuarios autenticados. En caso de errores, las pruebas proporcionaron información detallada, facilitando la identificación y corrección de problemas. En resumen, los resultados exitosos reflejan una aplicación robusta y bien desarrollada que cumple con las especificaciones establecidas. Aunque estos resultados son alentadores, se enfatiza la importancia de mantener la práctica de pruebas unitarias a medida que la aplicación evoluciona para preservar la calidad del código y prevenir posibles problemas en futuras actualizaciones.

3.3.2 Pruebas de aceptación

Pruebas de Caja Negra: También suelen ser llamadas funcionales y basadas en especificaciones. En ellas se pretende examinar el programa en busca de que cuente con las funcionalidades que debe tener y como lleva a cabo las mismas, analizando siempre los resultados que devuelve y probando todas las entradas en sus valores válidos e inválidos. (Martínez, 2014).

Por lo tanto, son los clientes los responsables de verificar que los datos de estas pruebas sean correctos. Para la correcta realización de estas pruebas se utilizó **la técnica de partición equivalente.**

La técnica de Partición Equivalente es una técnica de prueba que se utiliza para reducir el número de casos de prueba necesarios para probar un sistema. Esta técnica se basa en dividir el conjunto de datos de entrada en grupos o particiones que se espera que tengan un comportamiento similar. Luego, se selecciona un caso de prueba de cada partición para probar el sistema (Gomez, 2023).

Para aplicar la técnica de Participación Equivalencia, se debe seguir los siguientes pasos:

1. Se procederá a la identificación de distintos conjuntos de datos de entrada que pueden ser empleados en el sistema. Por ejemplo, en el caso de que el sistema admita números enteros, se llevará a cabo la identificación de particiones correspondientes a números positivos, negativos y cero.
2. Se establecerán las condiciones límite para cada partición, definiendo los valores extremos que pueden ser utilizados en cada una de ellas. Por ejemplo, en el caso de la partición de números positivos, se determinarán las condiciones límite como el número más pequeño y el número más grande que el sistema puede aceptar.
3. Se procederá a la selección de un caso de prueba representativo para cada partición. Este caso de prueba deberá reflejar el comportamiento típico de la partición correspondiente.
4. Se llevará a cabo la ejecución de los casos de prueba seleccionados y se verificará si el sistema se comporta conforme a las expectativas establecidas para cada partición. Este proceso garantizará la robustez y adecuado desempeño del sistema en distintos escenarios de entrada.

Método de partición de equivalencia

Identificador	Entrada (clases)	Clases válidas	Clases inválidas
Nombre	Nombre	Nombre = a-z	Nombre = vacío
Apellidos	Apellidos	Apellidos = a-z	Apellidos = vacío
Identificador	ID	Id = a-z; 0-9	ID = vacío
Municipio	Municipio	Municipio = seleccionar	Municipio = sin seleccionar
Entidad	Entidad	Entidad = seleccionar	Entidad= sin seleccionar

Tabla 18Case de Partición de equivalencia para Gestionar Proveedor

Escenario: Operación de Insertar Proveedor **Identificadores**	**Descripción:** Selecciona la opción Insertar Proveedor	**Respuesta del sistema**	**Flujo central**
Nombre	-	Brinda la posibilidad de introducir o seleccionar de manera obligatoria los siguientes datos: • Nombre • ID • Entidad • Municipio	/templates/tasks/create_task.htm
Descripción	-		
ID	-		
Nombre de la entidad	-		
Municipio	-		
Escenario: Opción de Aceptar.	**Descripción:** Introduce y/o selecciona los	**Respuesta del sistema**	**Flujo central**

Identificadores	datos de una solicitud y selecciona la opción Aceptar.		
Título	Nuevo Proveedor	Valida los datos, crea un nuevo proveedor, muestra una vista previa de la solicitud creada y muestra un mensaje de información (**Mensaje de estado**: El proveedor ha sido creada correctamente.	/templates/tasks/create_task.htm
Descripción			
ID	524P		
Nombre	Claudia María Fonseca		
Nombre de la entidad	ETECSA		
Municipio	Boyeros		
Escenario: Opción de Aceptar con datos incompletos.	**Descripción:** Introduce y/o selecciona alguno de los datos de una solicitud y selecciona la opción Aceptar.	**Respuesta del sistema**	**Flujo central**
Identificadores			
Título		Muestra un indicador sobre los campos	/templates/tasks/create_task.htm
ID	E-		

Nombre	Claudia María Fonseca	vacíos y muestra un mensaje de información (**Mensaje de error:** El campo es obligatorio).	
Nombre de la entidad	-		
Municipio	-1		
Escenario: Opción de Regresar al sitio.	**Descripción:** Selecciona la opción de Regresar al sitio.	**Respuesta del sistema**	**Flujo central**
Identificadores			
Título		Elimina los datos creados y regresa a la página anterior.	/templates/tasks/create_task.htm
ID			
Nombre			
Nombre de la entidad			
Municipio			

Tabla 19Caso de prueba Insertar un nuevo proveedor

Es fundamental considerar que la técnica de Partición Equivalente no asegura la detección exhaustiva de todos los errores en el sistema. Sin embargo, esta metodología desempeña un papel significativo al contribuir a la minimización del número de casos de prueba requeridos para abordar la totalidad de las condiciones posibles. El resto de las pruebas están en el Anexo 3

3.3.2 Resultados de las pruebas

Durante la ejecución de las pruebas, se efectuó una primera iteración con la finalidad de evaluar las historias de usuario y determinar en qué medida la aplicación cumplía con las funcionalidades implementadas. En esta fase inicial, se identificaron un total de 15 No Conformidades (NC), distribuidas en 3 de ortografía, 7 de validación y 5 de funcionalidades. Tras corregir estas irregularidades, se procedió a una segunda iteración, donde se detectaron 8 nuevas NC, compuestas por 3 de ortografía, 2 de validación y 3 de funcionalidades

Posteriormente, se llevó a cabo una tercera iteración que reveló la presencia de 3 nuevas NC, abarcando 1 de validación y 1 de funcionalidad y 1 de ortografía. Tras subsanar estas deficiencias, se ejecutó una cuarta iteración de pruebas que arrojó resultados satisfactorios. Con base en estos hallazgos, se tomó la decisión de no realizar más iteraciones de pruebas, considerándose concluido el proceso de evaluación.

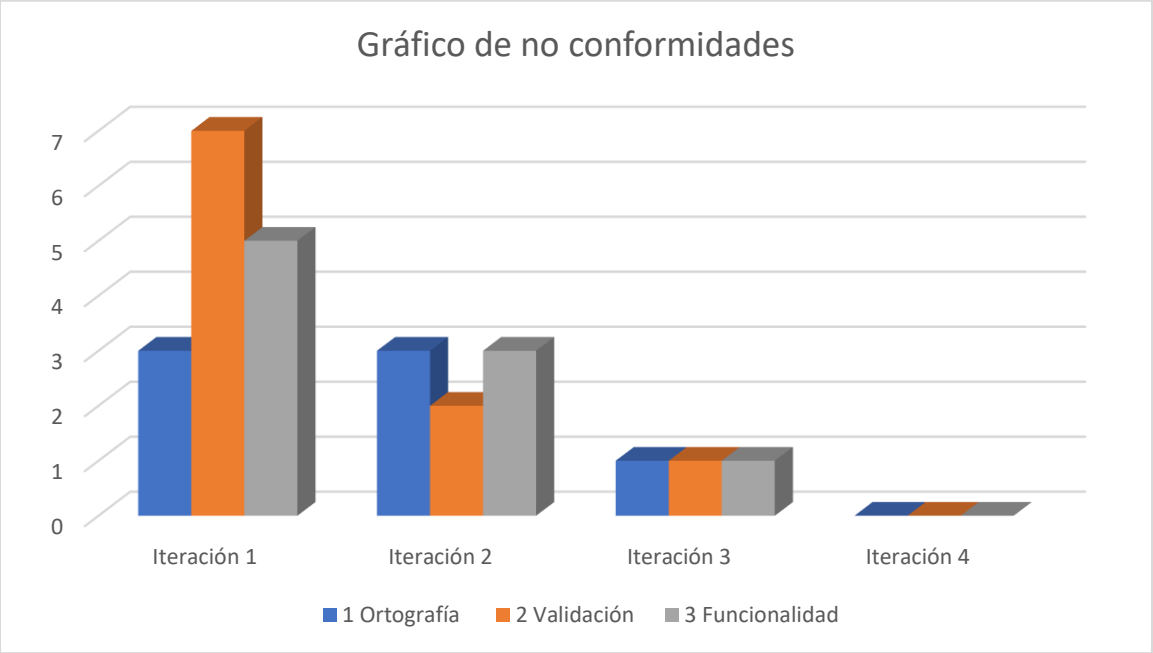


Tabla 19 No Conformidades

Las pruebas de aceptación son importantes dado que significan la satisfacción del cliente con el producto desarrollado, el final de una iteración y el comienzo de la siguiente. Se elaboran a lo largo de la iteración, en paralelo con el desarrollo del sistema, y adaptándose a los cambios que el sistema sufra (Pressman, 2014). A continuación, se muestran las pruebas de aceptación realizadas por cada HU con prioridad alta, las restantes se encuentran en el Anexo 4.

En la realización de estas pruebas se definieron los siguientes elementos:

CU: Número de la HU a la cual pertenece.

Nombre: Junto al código conforma el identificador del caso de prueba.

Descripción: Acción que debe realizar el sistema.

Condiciones de ejecución: Describe las características y elementos que debe contener el sistema para realizar el caso de prueba.

Entrada/Pasos de Ejecución: Incluye las entradas necesarias para realizar el sistema, además de los pasos para realizar el caso de prueba.

Resultados Esperados: Descripción de la respuesta del sistema ante el caso de prueba.

Evaluación de la prueba: Clasificación de la prueba en satisfactoria o insatisfactoria.

Pruebas de aceptación para la iteración 1

Caso de Prueba de Aceptación	
Código: HU1-P1	**Historia de Usuario:** HU-1
Nombre: Insertar producto	
Descripción: el sistema debe permitir insertar un nuevo producto en caso de haber llenado correctamente todos los campos, en caso contrario deberá mostrar una notificación de campos requeridos.	
Condiciones de ejecución: El usuario de rol almacenero debe estar autenticado en el sistema.	
Entradas/Pasos de ejecución: El usuario selecciona la opción Insertar producto y se le muestra un formulario con los campos a llenar, una vez llena los campos selecciona el botón Insertar y recibe un mensaje de notificación indicando el registro exitoso.	
Resultados esperados: Se inserta un producto en el sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 20Caso de prueba de aceptación #1

Tabla 12 *Caso de prueba de aceptación #2*

Caso de Prueba de Aceptación	
Código: HU1-P2	**Historia de Usuario:** HU-1
Nombre: Modificar producto	
Descripción: el sistema debe permitir modificar un producto previamente insertado.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema como almacenero.	
Entradas/Pasos de ejecución: El usuario selecciona la opción Modificar y se le muestra un formulario con los campos que puede cambiar, una vez modifica los campos selecciona el botón Modificar y recibe un mensaje de notificación indicando la modificación exitosa.	

Resultados esperados: Se modifica los datos de un producto.	
Evaluación de la prueba: Insatisfactoria	

Tabla 21Caso de prueba de aceptación #2

Caso de Prueba de Aceptación	
Código: HU1-P3	**Historia de Usuario:** HU-1
Nombre: Eliminar producto	
Descripción: el sistema debe permitir eliminar un producto previamente insertado.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema como almacenero.	
Entradas/Pasos de ejecución: El usuario selecciona la opción Eliminar y se le muestra una ventana de notificación para confirmar la eliminación, selecciona el botón Eliminar y recibe un mensaje de notificación indicando la eliminación exitosa.	
Resultados esperados: Se elimina un producto del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 22Caso de prueba de aceptación #3

Caso de Prueba de Aceptación	
Código: HU1-P4	**Historia de Usuario:** HU-1
Nombre: Buscar producto	
Descripción: el sistema debe permitir buscar un producto	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema como Almacenero.	
Entradas/Pasos de ejecución: El usuario escribe el nombre del producto que desea buscar, selecciona el botón buscar y se le muestra una ventana toda la información del producto correspondiente.	
Resultados esperados: Se muestra toda la información del producto.	
Evaluación de la prueba: Insatisfactoria	

Tabla 23Caso de prueba de aceptación #4

Caso de Prueba de Aceptación	
Código: HU1-P5	**Historia de Usuario:** HU-1
Nombre: Visualizar productos	
Descripción: el sistema debe permitir listar productos	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema como almacenero.	
Entradas/Pasos de ejecución: El usuario selecciona el botón Productos y se le muestra una tabla con todos los productos existentes.	

Resultados esperados: Se listan los productos.	
Evaluación de la prueba: Insatisfactoria	

Tabla 24 Caso de prueba de aceptación #5

Caso de Prueba de Aceptación	
Código: HU2-P6	**Historia de Usuario:** HU-2
Nombre: Filtrar productos	
Descripción: Prueba para la funcionalidad Filtrar productos	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema como almacenero.	
Entradas/Pasos de ejecución: el usuario selecciona el botón Filtros, y se muestra un menú de opciones, en dependencia de la selección se activa el filtrado.	
Resultados esperados: El usuario filtra la tabla de productos.	
Evaluación de la prueba: Insatisfactoria	

Tabla 25 Caso de prueba de aceptación #6

Caso de Prueba de Aceptación	
Código: HU3-P7	**Historia de Usuario:** HU-3
Nombre: Exportar datos	
Descripción: el sistema debe permitir exportar los datos de las tablas en formato Excel.	
Condiciones de ejecución: El usuario de rol almacenero debe estar autenticado en el sistema.	
Entradas/Pasos de ejecución: El usuario selecciona la opción Exportar datos y se muestra la opción de seleccionar una ubicación para guardar el archivo, luego toca aceptar y recibe un mensaje de notificación indicando la exportación exitosa.	
Resultados esperados: Se exporta correctamente los datos.	
Evaluación de la prueba: Insatisfactoria	

Tabla 26 Caso de prueba de aceptación #7

Resultado de las pruebas

En la siguiente tabla se muestra el comportamiento de las no conformidades encontradas durante el proceso de prueba:

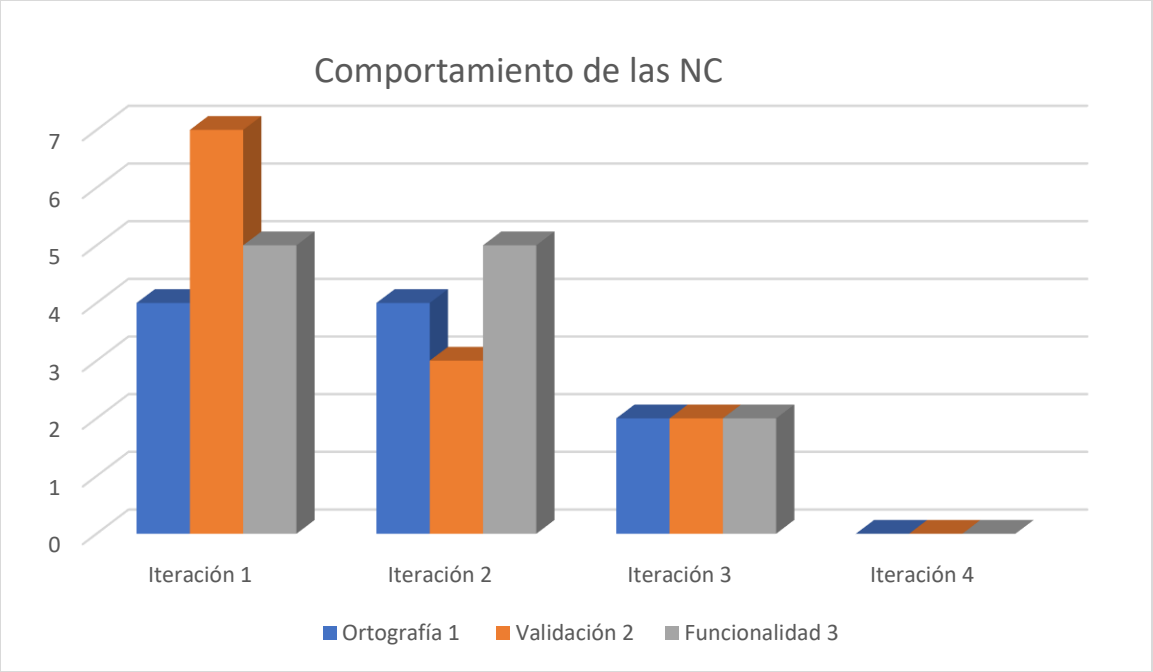


Tabla 27 Comportamiento de las no conformidades

Durante la ejecución de las pruebas, se llevó a cabo una primera iteración con el propósito de evaluar las historias de usuario y determinar en qué medida la aplicación cumplía con las funcionalidades implementadas. En esta fase inicial, se identificaron un total de 12 problemas detectados (PD), distribuidos en 4 de ortografía, 7 de validación y 5 de funcionalidades. Tras corregir estas irregularidades, se procedió a una segunda iteración, donde se detectaron 12 nuevos PD, compuestos por 4 de ortografía, 3 de validación y 5 de funcionalidades.

Posteriormente, se llevó a cabo una tercera iteración que reveló la presencia de 6 nuevos PD, abarcando 2 de validación, 2 de funcionalidad y 2 de ortografía. Tras subsanar estas deficiencias, se ejecutó una cuarta iteración de pruebas que arrojó resultados satisfactorios. Con base en estos hallazgos, se tomó la decisión de no realizar más iteraciones de pruebas, considerándose concluido el proceso de evaluación.

Conclusiones parciales

Al concluir este capítulo se termina la realización de la aplicación, obteniendo los siguientes resultados:

- La elaboración de los diseños de casos de pruebas permitió identificar los errores del sistema.
- Las pruebas de software aplicadas permitieron validar el correcto funcionamiento de la propuesta de solución.

Conclusiones Generales

Considerando los resultados arribados luego de realizar este informe se llegó a la conclusión de:

1. El análisis sistematizado de los referentes teóricos y plataformas homólogas de integralidad determinó que estas últimas no pueden ser propuestas como solución emergente a la problemática. Sin embargo, fueron seleccionados como solución parcial para el desarrollo del sistema de gestión de inventario.
2. Los resultados de la investigación que describe este informe demostraron la viabilidad de la solución propuesta, cuyo desarrollo contribuye a la automatización del proceso de gestión de inventario de las tiendas de venta de materiales de construcción de La Habana.

Recomendaciones

1. Dar continuidad a la presente investigación con vistas al mantenimiento del sistema propuesto y/o el diseño de nuevas funcionalidades que los cambios del entorno impongan.

2. Habilitar la conexión directa de las tiendas de ventas de materiales con la dirección de la empresa.

Referencias bibliográficas

Amazon. (2019). *¿Qué es Python? - Explicación del lenguaje Python - AWS*. Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/python/

Ballou, R. H. (1991). *Logística empresarial: Control y planificación*.

Bustamante, D. C. (2014). *Metodología XP*. Universidad Nacional Experimental De Los Llanos Occidentales Ezequiel Zamora.

Costa Rica, U. L. (2020, julio 9). ¿Que son las TIC y para qué sirven? *Universidad Latina de Costa Rica*. http://ulatina.ac.cr/articulos/que-son-las-tic-y-para-que-sirven

Del Toro Ríos, J. carlos, & González Brito, H. R. (2009). *Documento Visión. Proyecto ERP-Cuba.* Universidad de Ciencias Informáticas.

Eguiluz, J. (2018). *Capítulo 1. Introducción (Introducción a JavaScript)*. https://uniwebsidad.com/libros/javascript/capitulo-1

Eguiluz Pérez, J. (2008). *Introduccion a CSS*.

Ferreira Escutia, R. (2014). *"XP Extreme Programming"*. https://slideplayer.es/slide/84721/

Fuentes, I. L. L. (2008). *Modelado de una aplicación Web para el módulo inventario del sistema Cóndor*. Universidad de Ciencias Informáticas.

González Brito, H. R., & Lezano Lozadas, Y. (2005). *Sistema de Inventario Participativo en la UCI*. Universidad de Ciencias Informáticas.

Holovaty, Kaplan-Moss, A., Jacob. (2020). *El libro de Django*. https://uniwebsidad.com/libros/django-1-0

Jiménez Alvarado, Y. P. (2018). *HTML - Ensayos universitarios—7335 Palabras*. Buenas Tareas. https://www.buenastareas.com/ensayos/Html/81314431.html

KeepCoding, R. (2022, septiembre 14). *¿Qué es Bootstrap 5? | KeepCoding Bootcamps*. https://keepcoding.io/blog/que-es-bootstrap-5/

Larman, C. (2003). *UML y Patrones*.

Meana Coalla, Pedro, P. (2017). *Gestión de inventarios*. Ediciones Paraninfo, S.A.

Microsoft. (2022). *Get Started with Visual Studio Code*. https://code.visualstudio.com/learn/overview

Molina, Silvia Gabriela R. (2013). Metodologías ágiles enfocadas al modelado de requerimientos. *Informes Científicos Técnicos-UNPA*, *5*(1), 1-29.

PAE. (2021). *Portad de Administración Electrónica* [Red Sara]. http://administracionelectronica.gob.es/ctt/verPestanaGeneral.htm?idIniciativa=redsara.

Pérez S., A. (2020). *Ministerio de Hacienda y administración pública*. Gobierno de España. http://www.oficinavirtual.pap.minhap.gob.es/sitios/oficinavirtual/esES/CatalogoSistemasInformacion/sorolla2/Paginas/ModuloInventario.aspx.

Portugal, P. A. C., Álvarez, F. C., Tejedor, M. B. G., & Rodríguez, B. O. (2023). La administración de la cadena de suministro y su importancia en las empresas, como parte de la estrategia en los nuevos modelos de negocios. *Ciencia Latina Revista Científica Multidisciplinar*, *7*(3), 7203-7219.

Pressman. (2014). *Ingeniería del software: Un enfoque práctico, 7ma Edición – Roger S. Pressman*. https://www.freelibros.net/ingenieria/ingenieria-del-software-un-enfoque-practico-7ma-edicion-roger-s-pressman

Sepúlveda, S. A. (2022). *Gestión de inventarios—¿Qué es la gestión de inventarios?* | *SumUp Facturas*. https://www.sumup.com/es-es/facturas/glosario/gestion-de-inventarios/

Siriwardhana, S. (2020, octubre 26). Tutorial de Diagrama de Despliegue | ¿Qué es un Diagrama de Despliegue? *Blog de Creately*. https://creately.com/blog/es/diagramas/tutorial-de-diagrama-de-despliegue/

Viera Ipanaque, T. A. (2019). *Implementación de un generador de casos de prueba para procedimientos en lenguaje Java*. https://revistas.udem.edu.co/index.php/ingenierias/article/view/1646/2547

Visual Paradigm. (2021). *About Visual Paradigm*. https://www.visual-paradigm.com/aboutus/

Visual Paradigm. (2022). *What is Unified Modeling Language (UML)?* https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/

Vitlloch, A. C., Torres, I. C., & Davila, A. L. (2012). Modelo de gestión económica de la empresa estatal socialista de alta tecnología a la luz de los Lineamientos del VI Congreso del Partido Comunista de Cuba (PCC). *Economía y Desarrollo*, *148*(2), 94-123.

Wailgum. (2008). *An introduction to ERP*.

Westreicher, G. (2020). *Control de inventario*. Economipedia. https://economipedia.com/definiciones/control-de-inventario.html

Zorrilla Pantaleon, Martha E. (2011). *Modelo de datos*.

Anexos

Anexo 1 Historias de usuario

Tabla 13 *HU Autenticar Usuario*

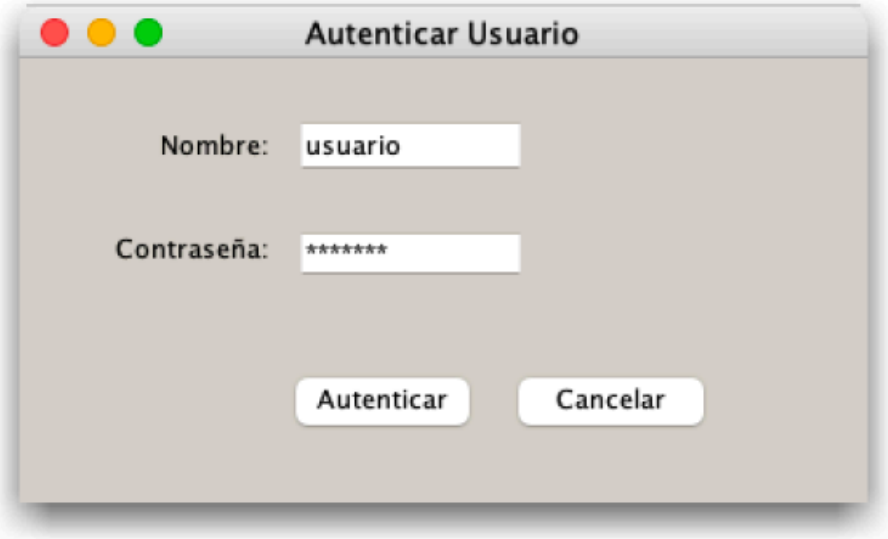
Historia de Usuario	
Número: 6	**Nombre:** Autenticar Usuario
Usuario: Usuario del sistema	
Prioridad en negocio: Media	**Riesgo en desarrollo:** Bajo
Estimación: 0.2	**Iteración asignada:** 3
Descripción: **Objetivo:** Permitir a los usuarios autenticarse al sistema. **Condiciones para lograr el objetivo:** El usuario debe de estar registrado **Comportamientos válidos y no válidos (flujo central y alternos):** - Si el usuario introduce un nombre de usuario que no está registrado en el sistema se muestra un mensaje de error. - Si el usuario introduce la contraseña incorrecta se muestra un mensaje de error. **Flujo de la acción a realizar:** El usuario selecciona la opción autenticar e introduce sus credenciales y de ser correctas el sistema carga el sitio con la información perteneciente al usuario.	
Prototipo de Interfaz:  	

Tabla 28 HU Autenticar Usuario

Historia de Usuario	
Número: 7	**Nombre:** Registrar Usuario
Usuario: Administrador	
Prioridad en negocio: Media	**Riesgo en desarrollo:** Bajo
Estimación: 0.8	**Iteración asignada:** 2

Descripción: Esta acción permite insertar, modificar, eliminar y visualizar la información de los usuarios.

Objetivo: El administrador podrá gestionar los usuarios registrados en el sistema.

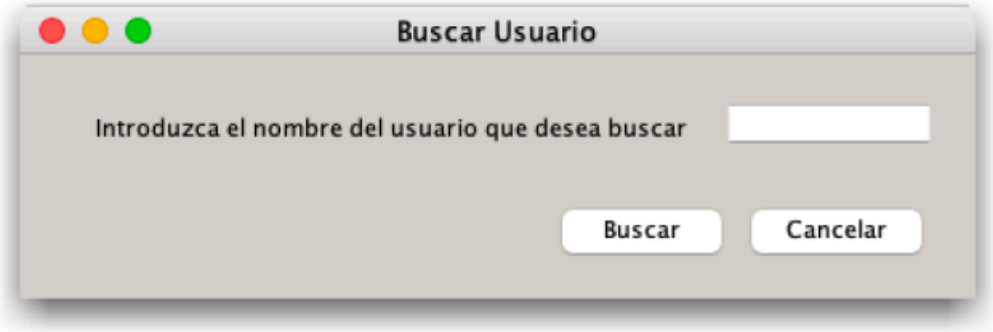
Acciones para lograr el objetivo (precondiciones y datos):

- Estar autenticado en el sistema como administrador.

Flujo de acción a realizar:

Cuando el usuario se autentica como Administrador el sistema le muestra una tabla donde visualiza la información de los usuarios, junto a los botones “Agregar Nuevo Usuario”, “Eliminar” y “Editar”.

Prototipo de Interfaz:



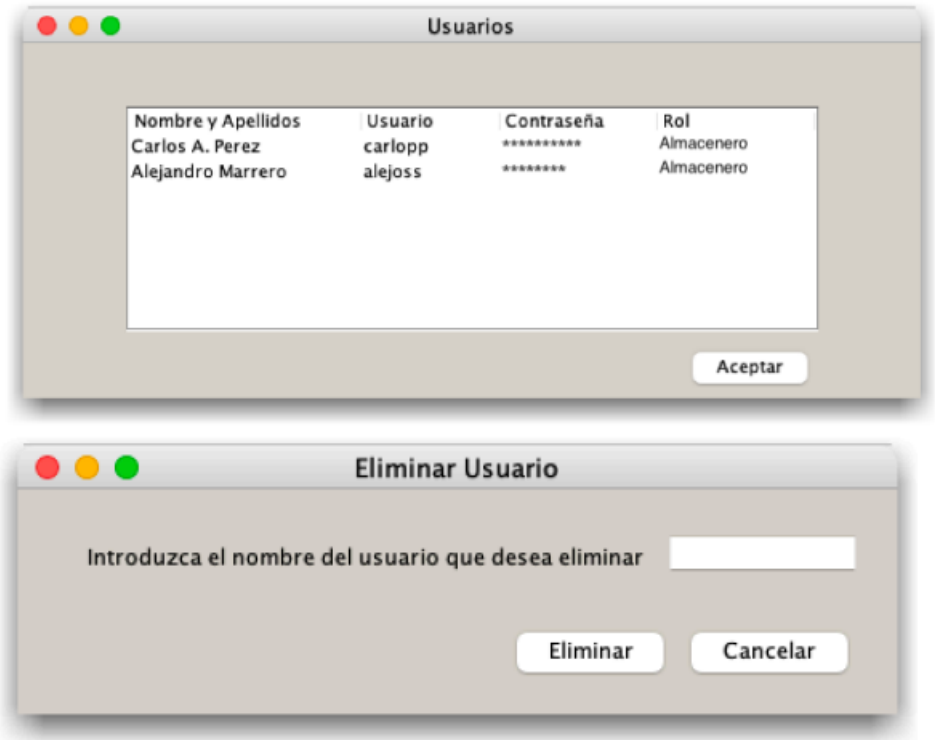


Tabla 29 HU Gestionar Usuarios

Historia de Usuario	
Número: 5	**Nombre:** Gestionar Proveedor
Usuario: Almacenero	
Prioridad en negocio: Media	**Riesgo en desarrollo:** Baja
Estimación: 1	**Iteración asignada:** 2
Descripción: Esta acción permite insertar, modificar, eliminar y visualizar la información de los proveedores. **Objetivo:** El almacenero podrá gestionar los proveedores registrados en el sistema. **Acciones para lograr el objetivo (precondiciones y datos):** - Estar autenticado en el sistema como almacenero. **Flujo de acción a realizar:** Cuando el usuario se autentica como Almacenero el sistema le muestra una tabla donde visualiza la información de los proveedores, junto a los botones “Agregar Nuevo Proveedor”, “Eliminar” y “Editar”.	
Prototipo de Interfaz:	

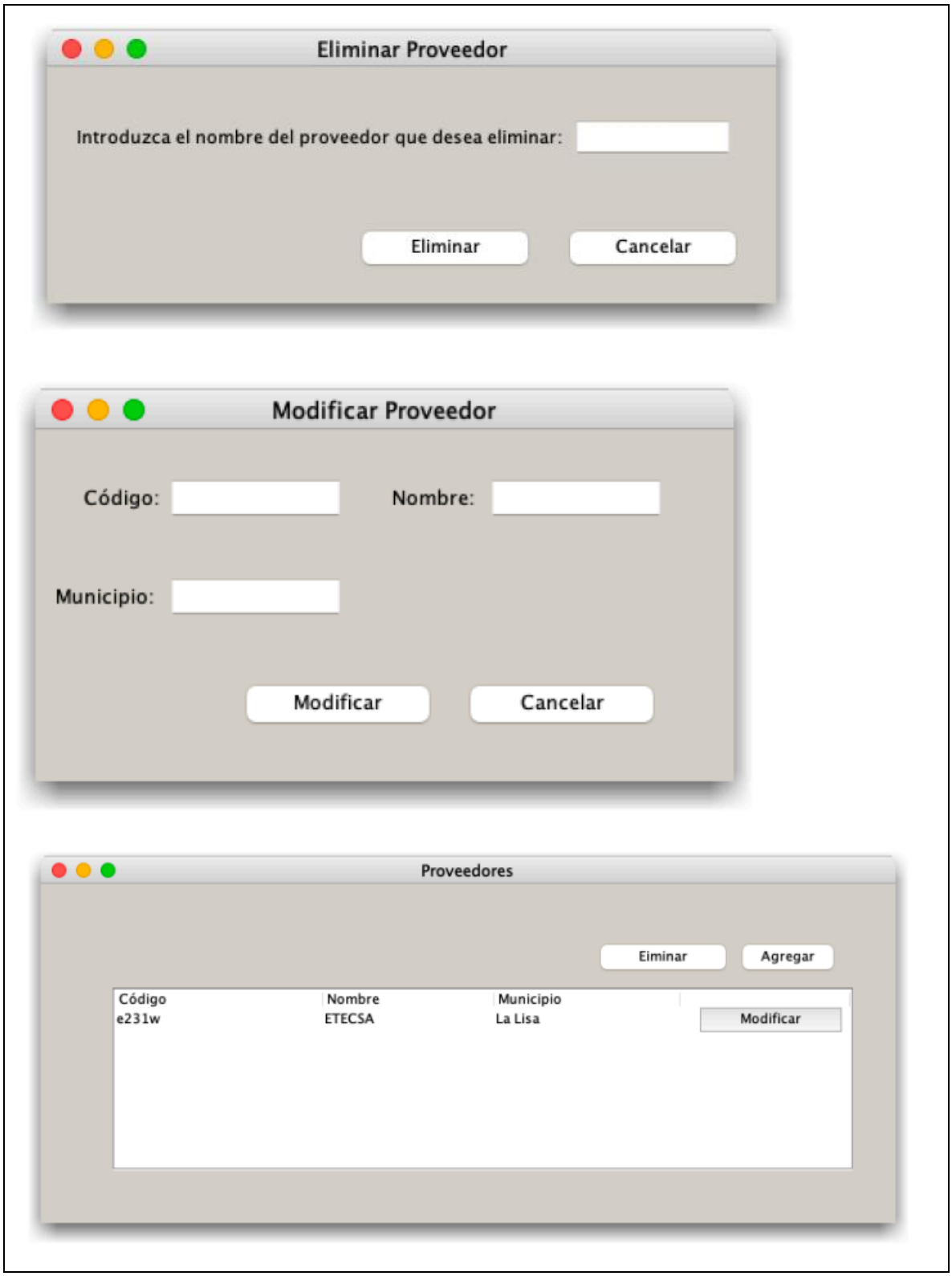


Tabla 30 HU Gestionar Proveedor

Anexo 2 Tareas de ingeniería

Tareas de la ingeniería para la iteración 2

Tarea de ingeniería	
Número de la Tarea: 8	**Número de la Historia de usuario:** 4
Nombre de la Tarea: Visualizar reporte de inventario	
Tipo de Tarea: Desarrollo	**Estimación:** 1
Fecha de inicio: 16/10/23	**Fecha de fin:** 20/10/23
Programador responsable: Damisela Thalía Gómez Cudina Jorge Saúl Pérez Peña	
Descripción: Implementar una vista que permita visualizar reporte de inventario.	

Tabla 31 Tarea de ingeniería 8

Tarea de ingeniería	
Número de la Tarea: 9	**Número de la Historia de usuario:** 5
Nombre de la Tarea: Insertar Proveedor	
Tipo de Tarea: Desarrollo	**Estimación:** 0.4
Fecha de inicio: 23/10/23	**Fecha de fin:** 24/10/23
Programador responsable: Damisela Thalía Gómez Cudina Jorge Saúl Pérez Peña	
Descripción: Implementar una vista que permita insertar un proveedor que cuando el usuario selecciona el botón para insertar un proveedor, el sistema muestra un formulario con los campos a llenar (código, nombre, municipio) y los botones de “Insertar” y “Cancelar”. El usuario llena los campos y presiona “Agregar”, el sistema guarda de manera consistente el proveedor. Si selecciona la acción “cancelar” regresa a la vista previa	

Tabla 31 Tarea de ingeniería 9

Tarea de ingeniería	
Número de la Tarea: 10	**Número de la Historia de usuario:** 5
Nombre de la Tarea: Modificar proveedor	
Tipo de Tarea: Desarrollo	**Estimación:** 0.2

Fecha de inicio: 25/10/23	**Fecha de fin:** 25/10/23
Programador responsable: Damisela Thalía Gómez Cudina Jorge Saúl Pérez Peña	
Descripción: Implementar una vista que permita modificar un proveedor cuando el usuario selecciona el botón para modificar un proveedor, el sistema muestra un formulario con los campos a que puede modificar (código, nombre, municipio) y los botones de "Modificar" y "Cancelar". El usuario modifica los campos y presiona "Modificar", el sistema guarda los cambios realizados. Si selecciona la acción "cancelar" regresa a la vista previa.	

Tabla 32 Tarea de ingeniería 10

Tarea de ingeniería	
Número de la Tarea: 11	**Número de la Historia de usuario:** 5
Nombre de la Tarea: Eliminar proveedor	
Tipo de Tarea: Desarrollo	**Estimación:** 0.2
Fecha de inicio: 26/10/23	**Fecha de fin:** 26/10/23
Programador responsable: Damisela Thalía Gómez Cudina Jorge Saúl Pérez Peña	
Descripción: Implementar una vista que permita eliminar un proveedor. Cuando el usuario selecciona el botón para eliminar un proveedor, el sistema muestra un mensaje pidiendo la confirmación de la eliminación y los botones de "Eliminar" y "Cancelar". El usuario presiona "Eliminar" y el sistema elimina el proveedor. Si selecciona la acción "cancelar" regresa a la vista previa.	

Tabla 33 Tarea de ingeniería 11

Tarea de ingeniería	
Número de la Tarea: 12	**Número de la Historia de usuario:** 5
Nombre de la Tarea: Visualizar proveedores	
Tipo de Tarea: Desarrollo	**Estimación:** 0.2
Fecha de inicio: 27/10/23	**Fecha de fin:** 27/10/23
Programador responsable: Damisela Thalía Gómez Cudina Jorge Saúl Pérez Peña	

Descripción: Implementar una vista que permita mostrar un listado de proveedores.

Tabla 34 Tarea de ingeniería 12

Tareas de la ingeniería para la iteración 3

Tarea de ingeniería	
Número de la Tarea: 13	**Número de la Historia de usuario:** 6
Nombre de la Tarea: Autenticar usuario	
Tipo de Tarea: Desarrollo	**Estimación:** 0.2
Fecha de inicio: 30/10/23	**Fecha de fin:** 30/10/23
Programador responsable: Jorge Saúl Pérez Peña Damisela Thalía Gómez Cudina	
Descripción: Implementar una vista que permita autenticar el usuario al sistema. Si el usuario introduce un nombre de usuario que no está registrado en el sistema se muestra un mensaje de error. Si el usuario introduce la contraseña incorrecta se muestra un mensaje de error. El usuario selecciona la opción autenticar e introduce sus credenciales y de ser correctas el sistema carga el sitio con la información perteneciente al usuario	

Tabla 35 Tarea de ingeniería 13

Tarea de ingeniería	
Número de la Tarea: 14	**Número de la Historia de usuario:** 7
Nombre de la Tarea: Registrar usuario	
Tipo de Tarea: Desarrollo	**Estimación:** 0.2
Fecha de inicio: 31/10/23	**Fecha de fin:** 31/10/23
Programador responsable: Damisela Thalía Gómez Cudina Jorge Saúl Pérez Peña	
Descripción: Implementar una vista que permita registrar un usuario en el sistema. El usuario con rol Administrador llena los campos del formulario (nombre y apellidos, usuario, contraseña, rol) y presiona el botón Registrar	

Tabla 36 Tarea de ingeniería 14

Tarea de ingeniería	
Número de la Tarea: 15	**Número de la Historia de usuario:** 7
Nombre de la Tarea: Modificar usuario	
Tipo de Tarea: Desarrollo	**Estimación:** 0.2
Fecha de inicio: 01/11/23	**Fecha de fin:** 01/11/23
Programador responsable: Jorge Saúl Pérez Peña Damisela Thalía Gómez Cudina	
Descripción: Implementar una vista que permita modificar un usuario. El usuario de rol administrador introduce el nombre de usuario que desea modificar, y luego modifica los campos que desee y el sistema guarda los cambios.	

Tabla 37 Tarea de ingeniería 15

Tarea de ingeniería	
Número de la Tarea:16	**Número de la Historia de usuario:** 7
Nombre de la Tarea: Eliminar usuario	
Tipo de Tarea: Desarrollo	**Estimación:** 0.2
Fecha de inicio: 02/11/23	**Fecha de fin:** 02/11/23
Programador responsable: Damisela Thalía Gómez Cudina Jorge Saúl Pérez Peña	
Descripción: Implementar una vista que permita eliminar un usuario. Debe existir el usuario. El usuario de rol administrador introduce el nombre de usuario que desea eliminar, y luego se muestra la eliminación.	

Tabla 38 Tarea de ingeniería 16

Tarea de ingeniería	
Número de la Tarea: 17	**Número de la Historia de usuario:** 7
Nombre de la Tarea: Visualizar usuarios	
Tipo de Tarea: Desarrollo	**Estimación:** 0.2
Fecha de inicio: 03/11/23	**Fecha de fin:** 03/11/23
Programador responsable: Jorge Saúl Pérez Peña	

Damisela Thalía Gómez Cudina
Descripción: Implementar una vista que permita mostrar un listado de usuarios. El usuario presiona Usuarios y el sistema muestra una tabla con todos los usuarios registrados.

Tabla 38 Tarea de ingeniería 17

Anexo 4 Pruebas de Partición de equivalencia

Nombre	Nombre	Título = a-z	Título = vacío
Tipo	Tipo	Tipo = seleccionar	Tipo = sin seleccionar
Cantidad	Cantidad	Cantidad = seleccionar	Cantidad = seleccionar la Cantidad
Precio de entrada	Precio de entrada	Precio de entrada = 0-9	Precio de entrada = caracteres especiales
Precio de venta	Precio de venta	Precio de venta = 0-9	Precio de venta = caracteres especiales
Proveedor	Proveedor	Proveedor = a-z	Proveedor = vacío
Municipio	Municipio	Municipio = seleccionar	Municipio = seleccionar
ID	ID	ID=a-z y 0-9	ID=a-z y 0-9

Tabla 39 Prueba de Partición de equivalencia Modificar producto

Escenario: Operación de Editar.	Descripción: Selecciona la opción Editar.	Respuesta del sistema	Flujo central
Identificadores			
Título	-Modificar Producto	Brinda la posibilidad de	/tasks/2/edit
Nombre	Cemento		

Precio de compra	400	editar los siguientes datos: • Nombre • Precio de compra • Precio de venta • Material • Entidad • Municipio	
Precio de venta	1000		
Entidad	MINCIN		
Municipio	cerro		
Escenario: Opción de Aceptar.	Descripción: Modifica y/o selecciona los datos de una solicitud y selecciona la opción Aceptar.	Respuesta del sistema	Flujo central
Identificadores			
Título	Modificar Producto	Valida los datos, modifica la solicitud, muestra una vista previa de la solicitud modificada y muestra un mensaje de información (Mensaje de estado: Producto Modificado correctamente.	/tasks/2/edit
Nombre	Cemento		
Precio de compra	200		
Precio de venta	400		
Material	Natural		
Entidad	MINCIP		
Municipio	Arroyo Naranjo		
Escenario: Opción de Guardar con datos incompletos.	Descripción: Modifica y/o selecciona los datos de una solicitud, dejando datos vacíos y selecciona la opción Guardar.	Respuesta del sistema	Flujo central
Identificadores			
Título	Modificar Productos	Muestra un indicador sobre los campos vacíos y muestra un mensaje de información (Mensaje de error:	/tasks/2/edit
Nombre	goma		
Precio de compra			
Precio de venta			
Materia	sintético		

Proveedor		El campo es obligatorio).	
Institución	XETID		
Municipio	Lisa		
Escenario: Opción de Regresar al sitio.	Descripción: Selecciona la opción de Regresar al sitio.	Respuesta del sistema	Flujo central
Identificadores			
Título	Modificar Productos	Elimina los datos creados y regresa a la página anterior.	/tasks/2/edit
Dirigido a	Dirección de la Empresa Provincial de Comercio		
Nombre	cinta		
Descripción	Se solicita el producto…		
Material	Sintético		
Precio de compra	508		
Precio de venta	670		
Institución	ANAP		
Municipio	Plaza de la Revolución		

Tabla 40 Caso de prueba Modificar Producto

Identificador	**Entrada (clase)**	**Clases válidas**	**Clases inválidas**
Usuario	Trabajador	Trabajador = a-z	Trabajador = caracteres especiales
Entidad que pertenece	Entidad que pertenece	Entidad que pertenece = seleccionar	Entidad que pertenece = no seleccionar
Descripción	Descripción	Descripción = a-z	Descripción = vacío
Material	Material	Material = seleccionar	Material = vacío

Municipio	Municipio	Municipio = seleccionar	Municipio = caracteres especiales
ID	ID	ID	ID

Tabla 41 Tabla para la prueba Insertar un nuevo usuario

Escenario: Operación de Añadir Identificadores	Descripción: Selecciona la opción Añadir.	Respuesta del sistema	Flujo central
Usuario	Marladom1	Brinda la posibilidad de añadir los siguientes datos: • Usuario • ID • Descripción • Material • Municipio	/solicitud_list
ID	ID		
Entidad a que pertenece	ANAP		
Descripción	Solicitud de un usuario…		
Material	Natural		
Municipio	Playa		
Escenario: Opción de Aceptar. Identificadores	Descripción: Modifica y/o selecciona los datos de una solicitud y selecciona la opción Aceptar.	Respuesta del sistema	Flujo central
Usuario	Marladom1	Valida los datos, modifica la solicitud, muestra una vista previa de la solicitud modificada y muestra un mensaje de información (Mensaje de estado: El usuario ha sido modificado correctamente.	/solicitud_list
ID	ID		
Entidad a que pertenece	ANAP		
Descripción	Solicitud de un usuario…		
Material	Natural		
Municipio	Playa		
Escenario: Opción de Guardar con	Descripción: Modifica y/o selecciona los datos de una solicitud,	Respuesta del sistema	Flujo central

datos incompletos. Identificadores	dejando datos vacíos y selecciona la opción Guardar.		
Usuario		Muestra un indicador sobre los campos vacíos y muestra un mensaje de información (Mensaje de error: El campo es obligatorio).	/solicitud_list
ID	ID		
Entidad a que pertenece	ANAP		
Material	Natural		
Municipio	Playa		
Escenario: Opción de Regresar al sitio. Identificadores	Descripción: Selecciona la opción de Regresar al sitio.	Respuesta del sistema	Flujo central
Usuario	Marladom1	Elimina los datos creados y regresa a la página anterior.	/solicitud_list
ID	ID		
Entidad a que pertenece	ANAP		
Material	Natural		
Municipio	Playa		

Tabla 42 Caso de prueba Insertar Usuario

Anexo 4 Pruebas de aceptación

Pruebas de aceptación para la iteración 2

Caso de Prueba de Aceptación	
Código: HU4-P8	**Historia de Usuario:** HU-4
Nombre: Visualizar reporte de inventario	
Descripción: el sistema debe permitir visualizar un reporte	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema como almacenero.	
Entradas/Pasos de ejecución: El usuario selecciona el botón Reportes y se le muestra un dashboard con las estadísticas.	
Resultados esperados: Se listan los productos.	

Evaluación de la prueba: Insatisfactoria

Tabla 43 Caso de prueba de aceptación #8

Caso de Prueba de Aceptación	
Código: HU5-P9	**Historia de Usuario:** HU-5
Nombre: Insertar proveedor	
Descripción: el sistema debe permitir insertar un nuevo proveedor en caso de haber llenado correctamente todos los campos, en caso contrario deberá mostrar una notificación de campos requeridos.	
Condiciones de ejecución: El usuario de rol almacenero debe estar autenticado en el sistema.	
Entradas/Pasos de ejecución: El usuario selecciona la opción Insertar proveedor y se le muestra un formulario con los campos a llenar, una vez llena los campos selecciona el botón Insertar y recibe un mensaje de notificación indicando el registro exitoso.	
Resultados esperados: Se inserta un proveedor en el sistema.	
Evaluación de la prueba: Insatisfactoria	

Tabla 44 Caso de prueba de aceptación #9

Caso de Prueba de Aceptación	
Código: H5-P10	**Historia de Usuario:** HU-5
Nombre: Modificar proveedor	
Descripción: el sistema debe permitir modificar un proveedor previamente insertado.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema como almacenero.	
Entradas/Pasos de ejecución: El usuario selecciona la opción Modificar y se le muestra un formulario con los campos que puede cambiar, una vez modifica los campos selecciona el botón Modificar y recibe un mensaje de notificación indicando la modificación exitosa.	
Resultados esperados: Se modifica los datos de un proveedor.	
Evaluación de la prueba: Insatisfactoria	

Tabla 45 Caso de prueba de aceptación #10

Caso de Prueba de Aceptación	
Código: HU5-P11	**Historia de Usuario:** HU-5
Nombre: Eliminar proveedor	
Descripción: el sistema debe permitir eliminar un proveedor previamente insertado.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema como almacenero.	
Entradas/Pasos de ejecución: El usuario selecciona la opción Eliminar y se le muestra una ventana de notificación para confirmar la eliminación, selecciona el botón Eliminar y recibe un mensaje de notificación indicando la eliminación exitosa.	
Resultados esperados: Se elimina un proveedor del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 46 Caso de prueba de aceptación #11

Caso de Prueba de Aceptación	
Código: HU5-P12	**Historia de Usuario:** HU-5
Nombre: Visualizar proveedores	
Descripción: el sistema debe permitir listar proveedores	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema como almacenero.	
Entradas/Pasos de ejecución: El usuario selecciona el botón Proveedores y se le muestra una tabla con todos los proveedores existentes.	
Resultados esperados: Se listan los proveedores.	
Evaluación de la prueba: Satisfactoria	

Tabla 47 Caso de prueba de aceptación #12

Pruebas de aceptación para la iteración 2

Caso de Prueba de Aceptación	
Código: HU6-P13	**Historia de Usuario:** HU-6
Nombre: Autenticar Usuario	
Descripción: Prueba para la funcionalidad Autenticar Usuario	
Condiciones de ejecución: El usuario debe estar registrado en el sistema.	

Entradas/Pasos de ejecución: Se introducen el nombre de usuario y la contraseña correspondiente al usuario que desea iniciar sesión, en caso de haber introducido algún dato incorrecto se muestra un mensaje de error.
Resultados esperados: El usuario accede al sistema.
Evaluación de la prueba: Satisfactoria

Tabla 48 Caso de prueba de aceptación #13

Caso de Prueba de Aceptación	
Código: HU7-P14	**Historia de Usuario:** HU-7
Nombre: Registrar usuario	
Descripción: el sistema debe permitir registrar un nuevo usuario en caso de haber llenado correctamente todos los campos, en caso contrario deberá mostrar una notificación de campos requeridos.	
Condiciones de ejecución: El usuario de rol administrador debe estar autenticado en el sistema.	
Entradas/Pasos de ejecución: El usuario selecciona la opción Registrar usuario y se le muestra un formulario con los campos a llenar, una vez llena los campos selecciona el botón Registrar y recibe un mensaje de notificación indicando el registro exitoso.	
Resultados esperados: Se registra un nuevo usuario en el sistema.	
Evaluación de la prueba: Insatisfactoria	

Tabla 49 Caso de prueba de aceptación #14

Caso de Prueba de Aceptación	
Código: HU7-P15	**Historia de Usuario:** HU-7
Nombre: Modificar usuario	
Descripción: el sistema debe permitir modificar un usuario previamente insertado.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema como Administrador.	
Entradas/Pasos de ejecución: El usuario selecciona la opción Modificar y se le muestra un formulario con los campos que puede cambiar, una vez modifica los campos selecciona el botón Modificar y recibe un mensaje de notificación indicando la modificación exitosa.	
Resultados esperados: Se modifica los datos de un usuario.	

Evaluación de la prueba: Satisfactoria

Tabla 50 Caso de prueba de aceptación #15

Caso de Prueba de Aceptación	
Código: HU7-P16	**Historia de Usuario:** HU-7
Nombre: Eliminar usuario	
Descripción: el sistema debe permitir eliminar un usuario previamente insertado.	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema como Administrador.	
Entradas/Pasos de ejecución: El usuario selecciona la opción Eliminar y se le muestra una ventana de notificación para confirmar la eliminación, selecciona el botón Eliminar y recibe un mensaje de notificación indicando la eliminación exitosa.	
Resultados esperados: Se elimina un usuario del sistema.	
Evaluación de la prueba: Satisfactoria	

Tabla 51 Caso de prueba de aceptación #16

Caso de Prueba de Aceptación	
Código: HU7-P17	**Historia de Usuario:** HU-7
Nombre: Listar usuarios	
Descripción: el sistema debe permitir listar usuarios	
Condiciones de ejecución: El usuario debe estar autenticado en el sistema como almacenero.	
Entradas/Pasos de ejecución: El usuario selecciona el botón Usuarios y se le muestra una tabla con todos los usuarios existentes.	
Resultados esperados: Se listan los usuarios.	
Evaluación de la prueba: Satisfactoria	

Tabla 52 Caso de prueba de aceptación #17

Anexo 5 Entrevista

Este instrumento se implementa con la finalidad de indagar, por parte del cuerpo de la dirección de comercio, sobre la actual ejecución del proceso de gestión de inventario en las tiendas de materiales de construcción. La presente investigación se materializa a través de entrevistas diseñadas para respaldar el análisis del

procedimiento de gestión de inventario por parte de los trabajadores, surge la necesidad de comprender las acciones emprendidas por el personal administrativo de la empresa. Se garantiza la confidencialidad en el manejo de los datos obtenidos, y se expresa anticipadamente agradecimiento por la colaboración proporcionada.

Nombre(s) y apellidos del entrevistado: Madelaine Cudina Hechavarría.

Cargo que ocupa Directora General de la Empresa Provincial de Comercio de la Habana.

Años de experiencia como directivo: 22.

Años de experiencia laboral: 30.

Nivel cultural: Universitario.

1. ¿Cuál es el nivel de utilización de las TIC en el proceso de gestión de inventario en las tiendas de materiales de construcción de la Habana?
2. ¿Cómo se lleva a cabo la gestión de ventas y cómo se integra con la gestión de inventarios?
3. ¿Pueden describir el flujo de trabajo actual desde la recepción del material hasta la venta?
4. ¿Cuáles son los problemas más frecuentes o difíciles que encuentran en el proceso actual de gestión de inventarios y ventas?
5. ¿Qué funcionalidades específicas les gustaría ver en un software de gestión de inventarios?
6. ¿Cuáles son los principales desafíos que enfrentan actualmente en la gestión de inventarios de ventas de materiales de construcción?

Respuestas:

1. Ah, en las tiendas de construcción de La Habana, las TIC (tecnologías de la información y comunicación) son como esos amigos que ves de vez en cuando. Mira, algunos usan computadoras para llevar el inventario, pero otros siguen en la era del papel y lápiz, ¡como si estuvieran en un museo! Y ni hablemos de internet, a veces parece que están conectados a una línea de tortugas. No es que no quieran,

es que parece que la tecnología avanza más lento que una tortuga coja. Eso sí, algunos han oído hablar de softwares y cosas así, pero la mayoría ni sueña con eso. ¡Son más de confiar en su instinto de cazadores de gangas que en un programa de computadora! En resumen, en términos de TIC, algunas tiendas están en el futuro, otras en el pasado, y unas cuantas parecen que viven en un limbo tecnológico. ¡Pero así es la vida en el mundo de la construcción, donde la tecnología va a su propio ritmo!

2. Bueno, mira, en las tiendas de construcción de La Habana, el tema de ventas y el inventario es medio como un lío. Usan esos sistemas antiguos de papel y lápiz, y la verdad es que a veces ni se enteran de lo que tienen en las tiendas. La cosa es que cuando venden algo, a veces lo apuntan y a veces no. Y si llega nueva mercancía, pues, ahí sí que se arma un relajo para ponerlo en el inventario.

La integración entre ventas e inventarios es como un misterio. No sé si existe realmente, porque a veces parece que uno no sabe lo que hace el otro. Es un poco caótico, pero bueno, así es como funciona, ¿no? ¡Ah, y a veces se les olvida apuntar cosas, así que imagina el desastre! Pero bueno, son detalles.

3. Ah, sí, el flujo de trabajo en las tiendas de construcción de La Habana es más o menos así: primero, llega la mercancía, ¿no? La descargan, la dejan por ahí y.… bueno, ahí empieza el caos. A veces ni saben bien lo que llegó, pero bueno, al menos está en algún lugar. Luego, cuando alguien viene a comprar algo, cruzan los dedos para tenerlo en la tienda. Van a la parte de atrás, buscan por todos lados y rezan para encontrar lo que el cliente quiere. A veces, hasta hacen magia para que coincida la mercancía con lo que está en el papelito de inventario, aunque eso es como una lotería. Y ya, cuando encuentran el dichoso material, lo llevan al frente, pasan por la caja. Pero en serio, el proceso es medio desordenado, como si estuvieran buscando tesoros perdidos. ¡Pero he y , así es la vida en el mundo de la construcción!

4. Ay, hermano, en las tiendas de construcción de La Habana, los problemas con la gestión de inventarios y ventas son como una telenovela. Primero, el inventario es un enigma, a veces ni ellos saben lo que tienen. ¡Es como jugar al escondite con los productos! Y las ventas, ni te cuento. A veces se les olvida registrar lo que venden, o peor aún, no encuentran la mercancía cuando alguien viene a comprar. ¡Parece que los productos se esfuman!

Ah, y el papelito ese que usan para llevar la cuenta, a veces ni lo entienden ellos mismos. Y cuando llega nueva mercancía, se arma el caos tratando de meter todo en esos estantes llenos hasta el tope. ¡Es un desastre total, pero qué le vamos a hacer, así es la vida en el mundo de la construcción, compadre!

5. Bueno, imagínate, en las tiendas de construcción de La Habana, si tuviéramos un software mágico, le pediríamos unas cosas bien chéveres. Primero, que nos diga de una vez por todas qué demonios tenemos en el inventario, porque eso de perder cosas ya es un clásico. Después, que tenga una función para decirnos cuándo hay que reponer los productos. A veces se nos acaban las cosas y ni enterados. ¡Sería genial tener un aviso tipo "oye, te estás quedando sin clavos"! Ah, y una cosa bien útil sería poder ver las ventas en tiempo real. Que podamos saber qué se está moviendo y qué no, para no perder el ritmo. Y claro, algo fácil de usar, porque si es complicado, ni nos metemos. En resumen, un software que nos saque de este enredo de inventarios, nos mantenga al día con las ventas y que no nos haga sentir como si estuviéramos lidiando con un monstruo de tres cabezas. ¡Eso sería la gloria!

6. Ay, compadre, en las tiendas de construcción de La Habana, los desafíos con la gestión de inventarios son como el cuento de nunca acabar. Primer problema: encontrar las cosas. En ese almacén parece que hay un agujero negro que se traga los productos. ¡Después de buscar como locos, a veces ni encuentran lo que tienen! Otro desafío es cuando llega nueva mercancía. ¡Se arma la fiesta tratando de encajar todo en esos estantes ya hasta el tope! Y lo peor es que a veces ni apuntan bien lo que entra. Es como si la mercancía llegara con invisibilidad temporal. Y claro, el clásico lío de controlar las ventas. A veces se les olvida registrar lo que venden, o ni saben qué está saliendo rápido y qué se queda empolvándose. ¡Es como tratar de domar un tigre con las manos desnudas! En resumen, el inventario es un caos, la mercancía tiene vida propia y las ventas son como tratar de hacer malabares con cucharas. ¡Pero bueno, así es la vida en el mundo de la construcción, lleno de emociones fuertes y desafíos a cada esquina!