



Universidad Autónoma del Estado de Hidalgo
Instituto de Ciencias Básicas e Ingeniería
Área Académica de Computación y Electrónica
Licenciatura en Ciencias Computacionales



**“Análisis y diseño conceptual de un sistema de gestión de inventario
para la microempresa Calidad Textil”**

T E S I S

Que para obtener el grado de
Licenciado en Ciencias Computacionales

P R E S E N T A

P.L.C.C. Olguín Beltrán Kelvin

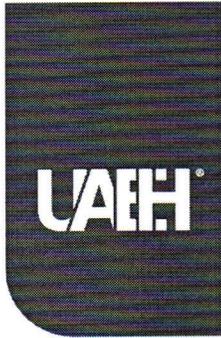
A S E S O R

Mtro. Gonzalo Alberto Torres Samperio

C O A S E S O R

Dr. Edgar Olguín Guzmán

Mineral de la Reforma, Hidalgo. 27 de febrero del 2024



Universidad Autónoma del Estado de Hidalgo
Instituto de Ciencias Básicas e Ingeniería
School of Engineering and Basic Sciences

Mineral de la Reforma, Hgo., a 29 de febrero de 2024

Número de control: ICBI-D/298/2024
Asunto: Autorización de impresión.

MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO
DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH

Con fundamento en lo dispuesto en el Título Tercero, Capítulo I, Artículo 18 Fracción IV; Título Quinto, Capítulo II, Capítulo V, Artículo 51 Fracción IX del Estatuto General de nuestra Institución, por este medio le comunico que el Jurado asignado al Pasante de la Licenciatura en Ciencias Computacionales **Kelvin Olguín Beltrán**, quien presenta el trabajo de titulación **“Análisis y Diseño conceptual de un sistema de gestión de inventario para la microempresa Calidad Textil”**, después de revisar el trabajo en reunión de Sinodales ha decidido autorizar la impresión del mismo, hechas las correcciones que fueron acordadas.

A continuación, firman de conformidad los integrantes del Jurado:

Presidente MID. Alberto Suárez Navarrete

Secretario: MID. Ma. de Jesús Gutiérrez Sánchez

Vocal: M.C.C. Gonzalo Alberto Torres Samperio

Suplente: Dr. Edgar Olguín Guzmán

Sin otro particular por el momento, reciba un cordial saludo.

Atentamente
“Amor, Orden y Progreso”

Dr. Otilio Arturo Acevedo Sandoval
Director del ICBI



OAAS/YCC

Ciudad del Conocimiento, Carretera Pachuca-Tulancingo Km. 4.5 Colonia Carboneras, Mineral de la Reforma, Hidalgo, México. C.P. 42184
Teléfono: 771 71 720 00 Ext. 2231 Fax 2109
direccion_icbi@uaeh.edu.mx



Agradecimientos

Agradecer en primer sitio a Dios por todas las bendiciones recibidas en mi vida. A mis padres por su constante amor y sacrificio para que pudiera realizar mi carrera.

A mis hermanos por apoyar todas mis decisiones, por brindarme su apoyo incondicional y por su comprensión en momentos de dificultad.

A mi futura esposa Lluvia; gracias por motivarme a seguir y concluir esta meta, por brindarme siempre tu amor y apoyo incondicional.

Expresar mi más profundo agradecimiento al Mtro. Gonzalo Alberto Torres Samperio y al Dr. Edgar Olguín Guzmán, por su orientación experta, paciencia y aliento constante a lo largo de este proceso. Su guía ha sido invaluable y ha enriquecido significativamente este trabajo que, sin duda, sin su contribución generosa no habría sido posible.

A todos los que han sido parte de este importante proceso; mil gracias.

Kelvin Olguín Beltrán

ÍNDICE

I. RESUMEN	I
Abstract	II
II. INTRODUCCIÓN	III
III. ANTECEDENTES.....	IV
IV. PLANTEAMIENTO DEL PROBLEMA.....	V
V. SOLUCIÓN PROPUESTA.....	VI
VI. JUSTIFICACIÓN.....	VII
VII. OBJETIVOS	VIII
Objetivo general.....	VIII
Objetivos específicos.....	VIII
VIII. REQUERIMIENTOS	IX
IX. ALCANCES Y LIMITACIONES	X
Alcances	X
Limitaciones.....	X
CAPÍTULO I: FUNDAMENTOS TEÓRICOS.....	11
1.1 Teoría general de sistemas.....	11
1.2 Definición de sistema	12
1.2.1 Tipos de sistemas.....	13
1.2.2 Clasificación de sistemas	13
1.3 Sistema de información.....	14
1.3.1 Definición de sistema de información	14
1.4 Ingeniería de software	15
1.4.1 Definición de la ingeniería de software	15
1.4.2 Características del software	16
1.4.3 Dominios de aplicación.....	16
1.4.4 Etapas del desarrollo de software.....	19

1.4.5 El software en el mundo	20
1.4.6 La industria del software en México.....	21
1.5 Lenguajes de programación.....	22
1.5.1 Clasificación de los lenguajes de programación	22
CAPÍTULO II: METODOLOGÍA	25
2.1 Marco metodológico.....	25
2.1.1 Modelo de desarrollo de software	25
2.1.2 Metodologías tradicionales	25
2.1.2.1 Waterfall (cascada).....	26
2.1.2.2 Prototipado.....	26
2.1.2.3 Espiral	27
2.1.2.4 Modelo de desarrollo concurrente	28
2.1.3 Modelos de proceso especializado	30
2.1.3.1 Modelo de desarrollo basado en componentes	30
2.1.4 Metodologías ágiles	31
2.1.4.1 XP	31
2.1.4.2 RUP.....	32
2.1.4.3 SCRUM	33
CAPÍTULO III: ANÁLISIS	39
3.1 Análisis del Sistema	39
3.1.1 Requisitos funcionales.....	39
3.1.2 Requisitos no funcionales.....	40
3.1.3 Perspectiva del producto	41
3.1.4 Funcionalidad del producto.....	41
3.1.5 Especificación de requerimientos	41
3.1.6 Proceso de ingeniería de requerimientos	44
CAPÍTULO IV: DISEÑO	47

4.1 Diagramas de Base de Datos.....	47
4.2 Diagramas UML	52
4.3 Interfaces gráficas de usuario.....	55
CONCLUSIONES.....	65
TRABAJOS FUTUROS	66
GLOSARIO	67
BIBLIOGRAFÍA	70
ANEXO 1: ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE IEEE 830.....	73

ÍNDICE DE FIGURAS

Figura 1 - Esquema de un complejo de elementos que interactúan y se comportan como un todo. Tomado de (Distancia-UNAM, 2023).....	12
Figura 2 - Clasificación de los sistemas. Elaboración propia.....	14
Figura 3 - Software de sistema. Tomado de (Ranchal, 2014).....	17
Figura 4 - Software de aplicación – Tomado de (Gaviglio, 2023).....	17
Figura 5 - Software de ingeniería. Tomado de (BARTOLINI, s.f.).....	17
Figura 6 - Software empotrado. Tomado de (Solutions, s.f.).....	18
Figura 7 - Procesadores de texto. Tomado de (Villegas, 2014).....	18
Figura 8 - Aplicación web. Tomado de (Londoño, 2023).....	18
Figura 9 - Software de inteligencia artificial. Tomado de (Reuters, 2023).....	19
Figura 10 - Etapas del desarrollo de software. Tomado de (Gómez Fuentes, Cervantes Ojeda, & González Pérez, 2019).....	20
Figura 11 - Modelo de la cascada. Obtenido de (Pressman, 2010).	26
Figura 12 - El paradigma de hacer prototipos. Obtenido de (Pressman, 2010).....	27
Figura 13 - Modelo de espiral común. Tomado de (Pressman, 2010).....	28
Figura 14 - Modelo de proceso concurrente. Tomado de (Pressman, 2010).	29
Figura 15 - Modelo de desarrollo basado en componentes. Tomado de (htt).....	31
Figura 16 - Proceso de XP. Obtenido de (Pressman, 2010).....	32
Figura 17 - Fases en el Proceso Unificado Racional. Tomado de (Sommerville, 2011).....	33
Figura 18 - Sprint Planning – Tomado de (Ferreyra, s.f.).....	36
Figura 19 - Sprint Backlog. Tomada de (Karlsson, s.f.).....	37
Figura 20 - Modelo Relacional. Elaboración propia.....	47
Figura 21 - Diagrama E-R. Elaboración propia.....	48
Figura 22 - Estructura de la base de datos (tablas).....	49
Figura 23 - Estructura de la base de datos (Campos/Panel de la empresa).....	50
Figura 24 - Estructura de la base de datos (Campos/clientes).....	50
Figura 25 - Estructura de la base de datos (Capos/Productos).....	51
Figura 26 - Estructura de la base de datos (Campos/proveedores).....	51
Figura 27 - Caso de uso. Elaboración propia.....	52
Figura 28 - Diagrama de actividades. Elaboración propia.....	53
Figura 29 - Diagrama de estados. Elaboración propia.....	53
Figura 30 - Diagrama de secuencias. Elaboración propia.....	54
Figura 31 - Diagrama de Colaboraciones. Elaboración propia.....	55
Figura 32 - Ventana de login.....	55
Figura 33 - Ventana cambiar contraseña.....	56
Figura 34 - Ventana de clientes.....	56
Figura 35 - Ventana de productos.....	57
Figura 36 - Ventana familia de productos.....	57
Figura 37 - Ventana detalle familia de productos.....	58
Figura 38 - Ventana Lista de precios.....	58
Figura 39 - Ventana configuración de impuestos y descuento.....	59
Figura 40 - Gráfica de ventas de productos.....	59
Figura 41 - Búsqueda de productos.....	60
Figura 42 - Panel de la empresa.....	60
Figura 43 - Módulos.....	61

Figura 44 - Registro de compra..... 61
Figura 45 - Botón añadir partida..... 62
Figura 46 - Selección del producto..... 62
Figura 47 - Control de inventario..... 62
Figura 48 - Detalle inventario..... 63
Figura 49 - Informe de ventas por producto..... 63
Figura 50 - Ticket de venta..... 64
Figura 51 - Importe de venta..... 64
Figura 52 - Reporte mensual de ventas..... 64

ÍNDICE DE TABLAS

Tabla 1 - Requisito Funcional 1.....	41
Tabla 2 - Requisito Funcional 2.....	42
Tabla 3 - Requisito Funcional 3.....	42
Tabla 4 - Requisito Funcional 4.....	42
Tabla 5 - Requisito Funcional 5.....	42
Tabla 6 - Requisito Funcional 6.....	43
Tabla 7 - Requisito Funcional 7.....	43
Tabla 8 - Requisito Funcional 8.....	43
Tabla 9 - Requisito Funcional 9.....	43
Tabla 10 - Requisito Funcional 10.....	44
Tabla 11 - Roles de Seguridad.....	45

I. RESUMEN

En la actualidad el mundo se mueve en una dirección que apunta más allá de una era digital, nadie imaginó que el poder e ingenio de los seres humanos nos traería hasta este punto, una realidad que se mueve en ceros y unos. Como profesionales de las Ciencias Computacionales, estamos conscientes de que en este mundo digital es indispensable estar a la altura de las circunstancias y que la tecnología nos irá rebasando día con día.

El presente trabajo, más allá de ser un elemento para concluir un ciclo como universitario, permite reflexionar muchos aspectos que en lo cotidiano parecen pasar desapercibidos. Actualmente la mayoría de las cosas funcionan a través de la computadora o los dispositivos móviles y estos se han convertido en una extensión más de nuestro cuerpo.

El desarrollo de este proyecto surge a partir de resolver una necesidad de negocio, el proceso inicia desde la concepción de la idea hasta lograr obtener un modelo que servirá de base para su desarrollo posterior. Estudiar cada una de las metodologías de desarrollo permite entender que todas ellas tienen aspectos en común, pero que al final del día, dependerá de la capacidad y experiencia del profesional de las Ciencias Computacionales el uso de estas en cada proyecto.

Actualmente existen infinidad de herramientas y metodologías para desarrollar software, pero su uso dependerá particularmente del tipo de proyecto en el cual se esté trabajando.

Se espera que el resultado de este proceso satisfaga plenamente las necesidades del cliente y que cumpla sus expectativas. Se tiene la seguridad de que el documento de especificación de requisitos del software permitirá al equipo de desarrollo crear software de calidad pues se ha cuidado este aspecto en sus etapas iniciales para lograr plenamente los objetivos.

Las fases de análisis y diseño son cruciales porque de no entender perfectamente lo que quiere el cliente, se corre el riesgo de que el sistema no cumpla sus expectativas y esto implica mayor carga de trabajo y esfuerzo para los desarrolladores. Es por ello que se ha dedicado especial atención en estas dos fases, para darle al cliente la seguridad de que el Ingeniero de Software sabrá interpretar la información plasmada en este presente documento y que desarrollará e implementará un software de calidad en la empresa.

Palabras clave

Desarrollo de Software, Inventarios, Metodologías Ágiles, Análisis, Diseño.

Abstract

Today the world is moving in a direction that points beyond a digital era, no one imagined that the power and ingenuity of human beings would bring us to this point, a reality that moves in zeros and ones. As Computer Science professionals, we are aware that in this digital world it is essential to be at the height of the circumstances and that technology will overtake us day by day.

This work, beyond being an element to conclude a cycle as a university student, allows us to reflect on many aspects that seem to go unnoticed in everyday life. Currently most things work through the computer or mobile devices and these have become an extension of our body.

The development of this project arises from solving a business need, the process starts from the conception of the idea to obtain a model that will serve as a basis for further development. Studying each one of the development methodologies allows understanding that all of them have aspects in common, but at the end of the day, it will depend on the capacity and experience of the Computer Science professional the use of these methodologies in each project.

There are currently an infinite number of tools and methodologies for software development, but their use will depend particularly on the type of project in which you are working.

The result of this process is expected to fully satisfy the customer's needs and meet their expectations. It is certain that the software requirements specification document will allow the development team to create quality software because this aspect has been taken care of in its initial stages in order to fully achieve the objectives.

The analysis and design phases are crucial because if we do not understand perfectly what the customer wants, there is a risk that the system will not meet his expectations and this implies a greater workload and effort for the developers. That is why special attention has been devoted to these two phases, to give the customer the assurance that the Software Engineer will know how to interpret the information contained in this document and that he will develop and implement a quality software in the company.

Keywords

Software Development, Inventory, Agile Methodologies, Analysis, Design.

II. INTRODUCCIÓN

El desarrollo de software ha ido en aumento en estos últimos años, con avances muy significativos, con la adopción de las metodologías ágiles y la expansión y uso de la inteligencia artificial. El software se usa en muchos ámbitos cotidianos, resuelven muchos problemas y facilitan el trabajo humano, en este orden de ideas, el presente trabajo de tesis, presenta una propuesta conceptual para resolver las necesidades de una microempresa que realiza de forma manual sus actividades de negocio. Al concluir el trabajo, el cliente podrá poner en marcha la siguiente fase del proyecto que consiste en el desarrollo e implementación del sistema, ya que el profesional de las Ciencias Computacionales contará con la información suficiente para su desarrollo.

Partiendo de los requerimientos existentes y con el fin de proyectarse a la mejora continua, el presente proyecto contiene la documentación correspondiente al análisis y diseño conceptual de un sistema de gestión de inventario para la microempresa Calidad Textil.

El capítulo I corresponde a los fundamentos teóricos, proporciona la base conceptual y el marco de referencia en el que se basa el análisis y diseño del proyecto, nos permite conocer los términos clave como son el software, la ingeniería de software, los sistemas y su clasificación, etc.

En el capítulo II se explica el funcionamiento de las metodologías de desarrollo de software que actualmente existen.

El capítulo III se compone de la fase de análisis y nos permite comprender cada uno de los requisitos del sistema para poder crear una estructura lógica del mismo, es decir, de aquí obtenemos la documentación necesaria para poder comenzar con la fase de diseño.

El capítulo IV está centrado en el diseño del proyecto, se trabajan diversos elementos para crear una arquitectura sólida y preparar la fase de desarrollo.

III. ANTECEDENTES

Desde hace mucho tiempo ha existido la necesidad de implementar estrategias que faciliten el control de inventarios, ya que un inventario detallado y ordenado es un elemento muy valioso que forma parte del patrimonio de una empresa o de una persona, actualmente conservados en bases de datos digitales.

“Durante la Segunda Guerra Mundial se utilizaron grandes y complejos ordenadores de la época, en los que se ejecutaban programas especializados para controlar la logística de las tropas de guerra. Estas soluciones se denominaron los primeros sistemas de planificación de inventario.” (México, 2023).

Otro antecedente importante data de la época de la revolución industrial en la que los procesos de gestión de inventario fueron factor crucial para lograr los objetivos de las empresas, ya que estas incrementaron sus niveles de producción y por ende de distribución.

Durante este periodo, el uso de tarjetas perforadas fue muy importante, permitiendo a los clientes realizar un seguimiento de sus pedidos. Este era un proceso muy costoso e ineficiente, por lo que las nuevas ideas dieron paso a los lectores de códigos de barras creados entre 1940 y 1950 los cuales podían rastrear artículos en venta. Poco a poco fue mejorando el funcionamiento de este mecanismo y fue ampliamente utilizado amén que las computadoras y el software permitieron que hasta la fecha se siga utilizando.

Como muchos otros procesos, la gestión de inventarios evoluciona día con día, siendo el móvil principal la necesidad de una mayor precisión y eficiencia. Actualmente existe software comercial para la gestión de inventarios, pero podemos decir que es mejor hacer un traje a la medida y acorde a necesidades específicas ya que de acuerdo con una de las características del software, señala que este es “de uso individualizado”.

En un principio los encargados de Calidad Textil pensaban adquirir software comercial bajo un programa de licenciamiento, pero la gran mayoría no cumple con las necesidades específicas de la empresa, por esta razón se plantea un sistema que resuelva sus necesidades de negocio, que se construya desde el seno de la empresa y que sea un traje a la medida.

IV. PLANTEAMIENTO DEL PROBLEMA

En el mundo empresarial, la correcta gestión de inventarios juega un rol importante, ya que con un buen manejo de esta actividad se reducen muchos costos.

Refiere (Navarro, 1999) que “uno de los mayores problemas que enfrentan las empresas en este momento es que una gran parte del capital de trabajo se invierte en inventario, que son recursos ociosos temporalmente, por lo que mantener estos inventarios es costoso.”

No es tarea fácil llevar a cabo una correcta gestión de inventarios, sobre todo cuando no se tienen las herramientas adecuadas para ello, sin estas, es común cometer diversos errores.

A medida que la cartera de clientes crece en Calidad Textil los problemas también, estos problemas que son cometidos por las personas encargadas de recuperar y registrar la información se enlistan a continuación:

1. **No se conoce la demanda de productos:** Con frecuencia este problema genera sobre stock o desabasto de productos.
2. **Falta de control:** Llevar el inventario en hojas de cálculo genera mucha carga manual de trabajo y es común cometer errores.
3. **Desorden en los registros:** Margen de ventas y estimación de inventarios.
4. **Seguridad de la información:** No cuenta con una gestión correcta de la información generada y el respaldo de la misma.

V. SOLUCIÓN PROPUESTA

Para resolver la problemática que tiene que ver con actividades manuales que desarrollan las personas encargadas de recuperar y registrar la información referente a inventarios en Calidad Textil, se propone establecer las bases conceptuales para desarrollar un sistema que permita realizar la correcta gestión de inventario en la microempresa.

Este sistema, de interfaz simple e intuitiva permitirá a los usuarios llevar un control de sus actividades de negocio y para tal efecto, se aborda únicamente la fase de análisis y diseño y de ellas se obtienen los elementos necesarios para en un futuro dar paso a la etapa de desarrollo e implementación.

VI. JUSTIFICACIÓN

La tecnología se ha vuelto un pilar importante en el funcionamiento de los negocios a nivel global, muchos de estos dependen prácticamente de ella para poder llevar a cabo todos sus procesos, en este mundo digital es imprescindible el uso de sistemas que faciliten el trabajo humano y que se minimicen errores que representen pérdidas económicas.

En el mundo empresarial a nivel macro y micro, el uso de herramientas de software ha facilitado mucho el trabajo de sus empleados, permitiendo que se dedique mayor tiempo a otras actividades como la expansión de su mercado de ventas.

Para Calidad Textil es importante estar a la vanguardia y hacer uso de herramientas de software que le permitan tener un mejor rendimiento empresarial, sabedores de los beneficios que estas otorgan, se abre paso para una nueva etapa en la empresa en la que se espera obtener más y mejores beneficios.

Con el análisis y diseño conceptual de un sistema de gestión de inventario, se espera obtener una propuesta atractiva para que el cliente pueda simplificar el trabajo, así como mejorar la rentabilidad de su negocio.

Para los clientes es importante que la empresa cuente con un sistema eficiente y que los productos o servicios estén al alcance de un clic. Esta tendencia hacia lo digital camina a pasos agigantados, aunque se tiene la creencia de que los sistemas sustituyen al hombre, es erróneo afirmarlo ya que estos no son totalmente autónomos.

Estamos en una era digital en donde el nivel de competencia es extremadamente alto y hacer uso de herramientas tecnológicas no solo nos permite estar a la vanguardia, además, nos permite entrar en otros mercados y escenarios donde no imaginamos estar.

Para que un sistema funcione correctamente y de respuesta a las necesidades y expectativas del usuario es necesario que pase por una fase de análisis y diseño conceptual muy detallada porque esta garantiza que el sistema funcione de acuerdo a las necesidades y expectativas del usuario.

VII. OBJETIVOS

Objetivo general

- Desarrollar el análisis y diseño conceptual de un sistema con la capacidad de gestión de inventario para la microempresa Calidad Textil, que facilite las tareas a los administradores y trabajadores, que minimice los errores operativos y que además les permita tener un ahorro significativo en las finanzas al reducir costos de operación en el manejo y control de inventarios.

Objetivos específicos

- Analizar los requerimientos funcionales y no funcionales que debe cumplir el software.
- Diseñar conceptualmente un software mediante técnicas de ingeniería para crear una arquitectura sólida y sentar las bases para la fase de desarrollo.
- Obtener el Documento de Especificación de Requisitos del Software con base en el estándar IEEE 830-1998.

VIII. REQUERIMIENTOS

Para realizar en su fase de análisis y diseño conceptual un sistema de gestión de inventario para la microempresa Calidad Textil; un equipo de cómputo adecuado requiere mínimo las siguientes características:

- 6 GB de memoria RAM.
- Procesador 3.5 GHz.
- 2 GB de almacenamiento libre.
- Conexión a internet.

Software requerido:

- Microsoft Office Professional Plus 2019
- Photoshop CS6 Portable

Requerimientos mínimos recomendados para la fase de desarrollo:

- 8 GB de memoria RAM.
- Procesador 3.5 GHz.
- 4 GB de almacenamiento libre.
- Conexión a internet.

Software sugerido para el desarrollo:

- FileMaker

IX. ALCANCES Y LIMITACIONES

Como todo proyecto, está sujeto a múltiples variables que muchas veces están fuera del alcance de quien está inmerso en él, pero con una buena gestión es posible alcanzar metas y superar límites.

Alcances

- En el presente proyecto solo se aborda en su fase conceptual el análisis y diseño de un software de funcionalidad simple e intuitiva que permita a quienes forman parte de Calidad Textil, llevar a cabo sus procesos de gestión de inventario.

Limitaciones

- Respecto al objetivo de construir un software administrativo para una empresa mediana y bajo un giro de consumo en red local, cumpliría perfectamente los requisitos en control de inventario y ventas, sin embargo, si se requiere una visión de construcción de ERPs (Enterprise Resource Planning) bajo lógica transaccional, logs y Kardex no cumpliría los requisitos para un desarrollo de dicha complejidad, optando a los desarrolladores el uso de lenguajes de programación potentes con motores de bases de datos eficaces.

CAPÍTULO I: FUNDAMENTOS TEÓRICOS

1.1 Teoría general de sistemas

Para entender esta teoría es importante conocer su contexto histórico ya que la teoría de sistemas o teoría general de sistemas (TGS) tiene su origen en la biología. “Fue el biólogo Ludwig Von Bertalanffy (1901-1972) quien dio vida a esta teoría a mediados del siglo XX, al buscar unificar la ciencia en una teoría que permitiera explicar los principios de todo sistema” (Distancia-UNAM, 2023).

De acuerdo con (Osorio, 2009), esta teoría está basada en tres premisas básicas son las siguientes:

1. **Los sistemas existen dentro de sistemas:** Cada sistema realiza tareas con la finalidad de cumplir con los objetivos planteados en representación de una dependencia superior, a la cual pertenece.
2. **Los sistemas son abiertos:** Presenta un margen amplio acerca de los beneficiados por las funciones de un determinado sistema.
3. **Las funciones de un sistema dependen de su estructura:** Hace referencia a la influencia que ejerce la estructura sobre las funciones que lleva a cabo un sistema, definiendo a la estructura como la relación entre las partes que conforman las organizaciones, partes como idea de sectores o áreas (gerencias, departamentos, áreas, divisiones) que están interrelacionadas y que obedecen a ciertas normas que las ordena y las nutre.

La TGS se encarga del estudio de los sistemas y sus propiedades, no importa la naturaleza de estos o su ámbito de aplicación.

“Basada en la idea de que todo sistema consta de una serie de elementos interrelacionados que interactúan para lograr un objetivo común. Esta idea se aplica a cualquier tipo de sistema, desde células biológicas hasta empresas e incluso ecosistemas.” (Huerta, 2023).

Este autor, afirma que “la teoría se basa en un conjunto de principios que se aplican a todos los sistemas, independientemente de su naturaleza o ámbito de aplicación.” (Huerta, 2023). Además, rescata algunos de los principios más importantes de la teoría de sistemas:

- **Principio de totalidad:** Un sistema debe estudiarse como un todo integrado y no como la suma de sus partes individuales. Están compuestos por partes interconectadas que funcionan como un todo.
- **Principio de interdependencia:** Los componentes de un sistema están interrelacionados y se afectan mutuamente.

- **Principio de equifinalidad:** Un sistema puede lograr un mismo objetivo a través de diversas rutas y medios.
- **Principio de retroalimentación:** Un sistema tienen la capacidad de recibir información y retroalimentarse, lo que le permite adaptarse y evolucionar constantemente.

La TGS es entonces de suma importancia en la ingeniería de software ya que proporciona una visión integral en el diseño, implementación y gestión de sistemas.

1.2 Definición de sistema

El término “sistemas” es muy común hoy por hoy, es un concepto en boga y sin duda todo mundo lo utiliza en la actualidad. Este concepto es usado en muchos campos de la ciencia, es muy común y mucha gente conoce su significado.

“Recientemente han aparecido profesiones y ocupaciones, desconocidas hasta hace nada, cuyos nombres incluyen proyecto de sistemas, análisis de sistemas, ingeniería de sistemas” (Bertalanffy, 1976). En la actualidad este concepto se ha ido redefiniendo dado que posee diversas interpretaciones según el contexto en el que se utilice, pero, para fines de la ingeniería y particularmente la ingeniería de software, el concepto se adecúa a las siguientes definiciones:

- Para Ludwig Von Bertalanffy; autor de la TGS “un sistema es un complejo de elementos que interactúan”.
- “La esencia de un sistema, según la definición de Bertalanffy, es ser un complejo de elementos, su característica es que estos elementos interactúan y su objetivo es que dicha interacción hace que este complejo de elementos se comporte como un todo” (Distancia-UNAM, 2023).
- De acuerdo con el autor del artículo titulado: Teoría general de sistemas de Ludwig Von Bertalanffy. “El término de sistema hace referencia a un todo organizado y complejo; un conjunto o combinación de cosas o partes que forman un todo complejo o unitario. Es un conjunto de objetos unidos por alguna forma de interacción o interdependencia” (Osorio, 2009).

Lo anterior se puede observar en la figura 1 que representa lo dicho por Bertalanffy.



Figura 1 - Esquema de un complejo de elementos que interactúan y se comportan como un todo. Tomado de (Distancia-UNAM, 2023)

Se entiende entonces que un sistema es un conjunto de elementos que interactúan entre sí para lograr un fin.

1.2.1 Tipos de sistemas

Para Ludwig Von Bertalanffy, autor de la Teoría General de Sistemas, existen dos tipos de sistemas cuyo criterio básico que los define es el grado de interacción con el suprasistema y otros sistemas; de tal manera que estos son abiertos y cerrados.

Sistema abierto

Es aquel que interactúa con su entorno y puede intercambiar información, materia y energía con él. Lo que lo hace adaptable y dinámico es su capacidad de enviar y recibir información.

Son relativos a su entorno; a medida que los sistemas se vuelven más complejos, el comportamiento de estos sistemas tiende a tener en cuenta su medio, es decir, su totalidad. “Generalmente este tipo de sistemas son los que tienen un periodo de vida más largo ya que se encuentran en una constante retroalimentación de los resultados que están obteniendo por lo que pueden mejorarse y actualizarse o incluso sufrir una reingeniería, si así se requiere” (Domínguez Ríos, 2017).

Sistema cerrado

Aquel que se encuentra aislado completamente de su entorno exterior, no intercambia materia con su entorno, aunque sí puede intercambiar energía. Esto quiere decir que no permite el flujo de materiales a través de sus fronteras.

“No tienen mecanismos de recolección de información del exterior, por lo que tienden a desaparecer al no contar con una retroalimentación que les de información sobre el resultado de sus acciones pasadas” (Méndez).

1.2.2 Clasificación de sistemas

Los sistemas pueden clasificarse de diversas maneras, una de las razones es para facilitar su estudio, por ello, existen varias clasificaciones, mismas que “permiten identificarlos y delimitarlos con una serie de características que a su vez reducen su complejidad, todo a razón que se conoce de manera más precisa el tipo de problema que se pretende atender” (Domínguez Ríos, 2017).

Algunas de las clasificaciones más comunes se describen en la figura 2:

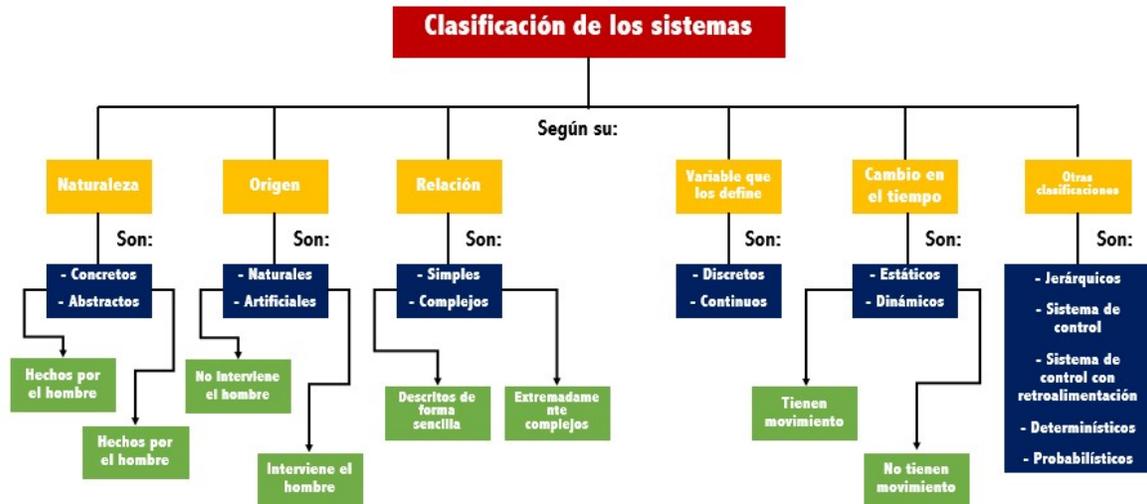


Figura 2 - Clasificación de los sistemas. Elaboración propia.

La clasificación anterior ayuda a comprender el funcionamiento y su interacción en diferentes contextos.

1.3 Sistema de información

Los sistemas de información juegan un papel preponderante y cada vez más relevante en todas las organizaciones comerciales, sabemos que el fin de las empresas en el mundo es poder satisfacer las necesidades de las personas y para esto se requiere de esfuerzos cada vez mayores que no sería posible atender sin el apoyo de la tecnología y de los llamados sistemas de información. El éxito de estas depende en gran medida de las herramientas que se tengan a disposición y la forma de resolverlas.

1.3.1 Definición de sistema de información

Los autores del boletín: Sistemas de información en las organizaciones, refieren que “un sistema de información es un sistema organizacional formalizado que se define como un conjunto de componentes interrelacionados que incluyen elementos sociales y técnicos organizados para recolectar, procesar, ordenar, almacenar y convertir los datos en información; la cual se distribuye en la organización para apoyar los procesos de toma de decisiones relacionados con la coordinación, control de recursos, análisis de problemas y temas complejos así como en la creación de nuevos productos” (C. Laudon & P. Laudon, 2012).

Podemos entenderlo entonces como un conjunto organizado de componentes interactuando para recopilar, procesar, almacenar y distribuir información para apoyar en la toma de decisiones, el control y la coordinación en una organización.

Estos pueden contener, software, hardware, datos, procesos y personas que en conjunto pueden aportar información importante y útil para los usuarios dentro de una organización. Estos sistemas cobran preeminencia ya que en el mundo empresarial permiten la toma de decisiones informadas, la gestión de los datos, la mejora en la comunicación, la eficiencia operativa y la competitividad.

1.4 Ingeniería de software

Resulta imposible operar el mundo moderno sin software ya que “las infraestructuras nacionales y los servicios públicos se controlan mediante sistemas basados en computadoras, y la mayoría de los productos eléctricos incluyen una computadora y un software de control” (Sommerville, 2011).

Con el apogeo de la tecnología y la necesidad de sistematizar diversas actividades cotidianas, la ingeniería de software se ha convertido en un elemento importante en la sociedad actual dado que es una disciplina que hace uso de herramientas y técnicas para construir programas informáticos.

“La ingeniería de software es un campo de estudio que cubre las estructuras, herramientas y métodos utilizados en el desarrollo de programas informáticos. Por lo tanto, la ingeniería de software es una rama de la ingeniería que utiliza técnicas y experimentos de la informática, la gestión de proyectos y otras disciplinas para desarrollar y gestionar sistemas de software.” (UNIR - Universidad Internacional de La Rioja, 2021).

Es entonces tan importante y fundamental en la actualidad, es crucial para el desarrollo exitoso de aplicaciones y sistemas que impulsan muchas áreas de nuestra vida y negocios.

1.4.1 Definición de la ingeniería de software

Muchos autores han definido el término, uno de los principales fue (Sommerville, 2011) quien la define como “una disciplina que se interesa por todos los aspectos de la producción de software, desde las primeras etapas de la especificación del sistema hasta el mantenimiento del sistema después de que se pone en operación”.

La IEEE (The Institute of Electrical and Electronics Engineers: El Instituto de Ingenieros Electricistas y en Electrónica), define a la ingeniería de software como "la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software".

Para (Gómez Fuentes, Cervantes Ojeda, & González Pérez, 2019), una de las definiciones más interesantes es la de Bohem (1976) la cual sostiene lo siguiente: "Ingeniería de software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida

para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software".

Para Fritz Bauer es el establecimiento y uso de principios fundamentales de la ingeniería para desarrollar de forma económica software.

Definimos entonces la ingeniería de software como: Una disciplina en la que se aplican técnicas y principios de forma sistemática en el desarrollo de sistemas de software para garantizar su calidad.

1.4.2 Características del software

"El software es elemento de un sistema lógico y no de uno físico. Por tanto, tiene características que difieren considerablemente de las del hardware" (Pressman, 2010).

Generalmente son tres las características del software y se describen a continuación:

El software se desarrolla o modifica con intelecto; no se manufactura en el sentido clásico

Esto quiere decir que, aunque hay algunas similitudes entre el desarrollo de software y la fabricación de hardware, ambas actividades son diferentes.

El software no se "desgasta"

Es decir, el software no es susceptible a los problemas ambientales que hacen que el hardware se desgaste, pero sí se deteriora.

Aunque la industria se mueve hacia la construcción basada en componentes, la mayor parte del software se construye para un uso individualizado

Es decir, se hace un traje a la medida.

1.4.3 Dominios de aplicación

De acuerdo con (Pressman, 2010) autor del libro "Ingeniería del Software: Un enfoque práctico"; actualmente, hay siete grandes categorías de software de computadora que plantean retos continuos a los ingenieros de software:

Software de sistemas

Son escritos para dar servicio a otros programas, por ejemplo: Compiladores, editores y herramientas para administrar archivos.



Figura 3 - Software de sistema. Tomado de (Ranchal, 2014).

Software de aplicación

Resuelven una necesidad específica. Las aplicaciones en esta área procesan datos comerciales o técnicos en una forma que facilita las operaciones de negocios o la toma de decisiones administrativas o técnicas.



Figura 4 - Software de aplicación – Tomado de (Gaviglio, 2023).

Software de ciencias e ingeniería

Destacan aplicaciones de vulcanología y astronomía, apps encargadas de análisis de tensiones en automóviles o aquellas que analizan la orbital de un transbordador espacial, otras que estudian la biología molecular y también la manufactura automatizada.

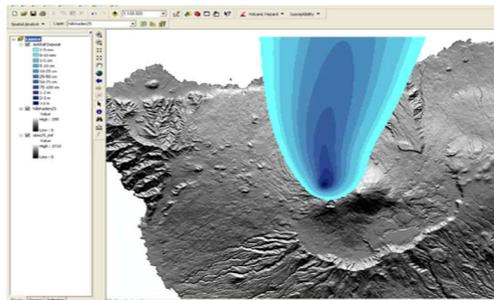


Figura 5 - Software de ingeniería. Tomado de (BARTOLINI, s.f.).

Software embebido o incrustado

Este reside dentro de un sistema o producto y se usa para implementar y controlar características y funciones para el usuario final y para el sistema en sí.

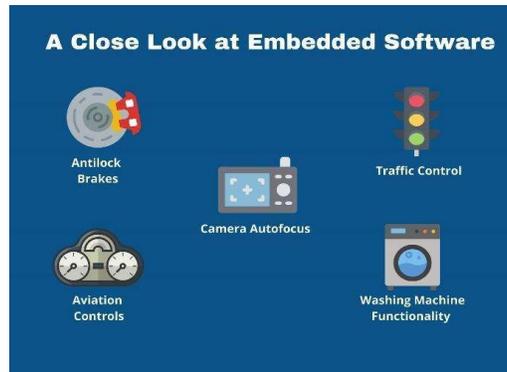


Figura 6 - Software empotrado. Tomado de (Solutions, s.f.).

Software de línea de productos

Desarrollados para ofrecer una capacidad específica para uso de distintos consumidores.



Figura 7 - Procesadores de texto. Tomado de (Villegas, 2014).

Aplicaciones web

Conocidas como “WebApps”, esta categoría congrega un amplio contenido de aplicaciones.



Figura 8 - Aplicación web. Tomado de (Londoño, 2023).

Software de IA (Inteligencia Artificial)

Con el uso de algoritmos no numéricos puede resolver problemas que computacionalmente o con el análisis directo no es posible tratar.

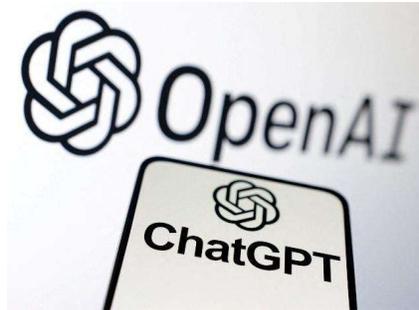


Figura 9 - Software de inteligencia artificial. Tomado de (Reuters, 2023).

1.4.4 Etapas del desarrollo de software

Un proceso de desarrollo de software es el conjunto organizado de actividades requeridas para construir un sistema.

“El proceso de desarrollo de software se utiliza para mejorar la comprensión del problema a resolver, la comunicación entre los participantes del proyecto y el mantenimiento de un sistema complejo” (Gómez Fuentes, Cervantes Ojeda, & González Pérez, 2019).

Existen tres etapas primordiales en el desarrollo de software que se describen a continuación:

1.- Definición: Centrada en el ¿Qué?

Información que será procesada, función y rendimiento, interfaces que serán establecidas, restricciones de diseño, criterios de validación y requisitos clave del sistema.

2.- Desarrollo: Centrada en el ¿Cómo?

Cómo han de diseñarse las estructuras de datos y cómo ha de efectuarse la función dentro de una arquitectura de software.

3.- Mantenimiento: Centrada en el cambio y asociada a la corrección de errores.

Para asegurar que el software funcionará como lo ha hecho, aquí se le da una conservación a su buen funcionamiento, además de mejorar y reparar sus errores, para ello, el mantenimiento se clasifica de la siguiente manera:

- **Preventivo:** Hace cambios en el programa a fin de que se puedan corregir, adaptar y mejorar, conocido también como reingeniería del software.

- **Correctivo:** Aun cuando se lleven a cabo todas las actividades de garantía de calidad, es probable que el cliente halle defectos en el software, por ello, en este tipo de mantenimiento se cambia el software para corregir sus defectos.
- **Adaptativo:** Con el paso del tiempo, es posible que el entorno original para el cual se diseñó el software cambie. Por este motivo este tipo de mantenimiento produce modificaciones en el software para ajustarse a los cambios de su entorno.
- **Perfectivo o mejora:** Conforme los usuarios y el cliente utilicen el software, pueden descubrir funciones adicionales que traerán beneficios y es por ello que aquí se lleva al software más allá de sus funciones originales.

En la figura 10 se ilustran las actividades mencionadas anteriormente.



Figura 10 - Etapas del desarrollo de software. Tomado de (Gómez Fuentes, Cervantes Ojeda, & González Pérez, 2019).

1.4.5 El software en el mundo

Sector cuya tarea principal es la producción, desarrollo y comercialización de software. En el mundo de las TIC, no existe un modelo de organización global definido para la industria de software que guíe su desarrollo o la manera de implementarse. Cada país lo

hace conforme a sus posibilidades, esta industria está sujeta a muchos factores como el potencial y desarrollo económico de cada región.

De acuerdo con (Cutrell), el autor del artículo; el contexto y el diseño de TIC para el desarrollo mundial, “las TIC están profundamente arraigadas en el tejido social y forman parte de la forma en que conducimos nuestros negocios”.

Actualmente existen más de cinco mil millones de teléfonos móviles en el mundo y por ende el alcance de la tecnología es cada vez más global. “El mundo digital no tiene límites y las TIC son el elemento más disruptivo de la economía moderna” (Salamanca, s.f.).

1.4.6 La industria del software en México

Esta industria en nuestro país ha experimentado un crecimiento significativo en los últimos años. Un gran número de empresas nacionales e internacionales se han establecido y han iniciado operaciones en nuestro territorio, sirviéndose de la mano de obra calificada y los costos competitivos.

De acuerdo con (Vázquez, 2022), “la crisis sanitaria por covid-19 y la innovación han propiciado que el software se coloque en un lugar importante y que a su vez genere nuevos modelos de negocio”.

En nuestro país, la venta de software como servicio (SAAS) se perfila como una tendencia que llegó para quedarse ya que ciudades como Monterrey y Guadalajara han surgido como centros tecnológicos clave, albergando numerosas empresas de desarrollo de software.

“Durante la pandemia por covid-19, la solicitud de servicios de software creció de manera importante, ubicándola como una de las industrias más importantes a nivel económico” (Vázquez, 2022).

Datos relevantes sobre la industria del SaaS

De acuerdo con datos e información del Statista (plataforma global de datos e inteligencia empresarial):

- En 2021 creció 90% el uso de aplicaciones compradas en México.
- 5 de cada 10 empresas en México duplicaron su crecimiento en internet, y 2 de cada 10 registraron crecimientos superiores al 300% en sus ventas en línea.
- El mercado de software en México logrará un tamaño aproximado de \$3,900 millones de dólares y presentará un crecimiento promedio de 6.28% hasta 2027.
- En el 2022, el mercado de Software como Servicio (SaaS) en México tiene un tamaño de 800 millones de dólares con un crecimiento cercano al 25%.

1.5 Lenguajes de programación

Un lenguaje de programación es la vía para comunicarnos con una computadora y decirle qué queremos hacer, pero desde un punto de vista técnico, es una herramienta que nos permite desarrollar software, aplicaciones y sistemas.

Los lenguajes de programación existen hoy en día en gran número, muchos de ellos siguen vigentes después de mucho tiempo y otros tantos se han dejado de usar.

La función de los lenguajes de programación es escribir programas que faciliten la comunicación entre los usuarios y las computadoras y para ello, estos se conforman de reglas y símbolos que definen la estructura del lenguaje.

1.5.1 Clasificación de los lenguajes de programación

Una clasificación habitual de los lenguajes de programación se da de la manera siguiente:

Lenguaje máquina

Basa su funcionamiento con el cero (0) y el uno (1), estos signos simbolizan dos únicos niveles de tensión, lo cual significa que los circuitos microprogramables son sistemas digitales.

Es un sistema de códigos interpretable directamente por un circuito microprogramable, tal es el caso de un microprocesador de computadora. Está compuesto por instrucciones que determina las acciones que realizará la máquina.

Los programas de computadoras, son básicamente una cadena de instrucciones de lenguaje máquina, donde habitualmente son ejecutadas secuencialmente, con cambios de flujo causados por el propio programa. Estos lenguajes son particulares en cada máquina y dependen de la arquitectura de la misma.

Características:

- Son una serie de instrucciones reconocidas por los circuitos del procesador.
- Se codifica en binario.
- Los datos se referencian por su posición en memoria.

Sostiene (Barber & Ferrís) que “los lenguajes de programación se pueden clasificar en lenguajes de bajo y alto nivel dependiendo de lo cercanos o lejanos que estén de la arquitectura de la máquina en la que van a funcionar”.

Lenguaje de bajo nivel

Referente a la abstracción reducida entre el lenguaje y el hardware, lo cual significa que es trasladado fácilmente a lenguaje de máquina.

Aquel que proporciona poca o nula abstracción del microprocesador de una computadora y, en consecuencia, su traslado al lenguaje máquina es fácil.

Dicho con palabras de (UNAM, s.f.), “el término ensamblador (del inglés assembler) se refiere a un tipo de programa informático encargado de traducir un archivo fuente, escrito en un lenguaje ensamblador, a un archivo objeto que contiene código máquina ejecutable directamente por la máquina para la que se ha generado”.

Características:

- Basados en la máquina de Von Neumann, por tal motivo, están a un nivel muy cercano a la máquina.
- Las instrucciones del lenguaje son las instrucciones del microprocesador del ordenador, que normalmente son demasiado simples.
- Es exclusivo de cada computadora.
- Difícil y costoso de programar.

Lenguajes de alto nivel

Su principal característica es que expresan los programas de forma sencilla y su estructura semántica es muy similar a la forma como escriben los humanos, razón que le permite codificar los algoritmos de una forma más natural, en vez de codificarlos en binario, o a nivel de ensamblador.

Características:

- Basados en máquinas abstractas, que facilitan la comprensión de las personas.
- Instrucciones flexibles y potentes.
- Requiere de un traductor para transformar el programa en lenguaje máquina, que es el que entiende el ordenador.
- No depende del procesador, por esta razón el programa sirve para diferentes ordenadores.
- Es más lento e ineficiente que el lenguaje de bajo nivel.

Ejemplos de lenguajes de alto nivel: C++, Fortran, Java, Perl, PHP y Python.

Resulta difícil establecer una clasificación de los lenguajes de alto nivel ya que actualmente son muchos, pero atendiendo a la forma de trabajar de los programas y la filosofía en la que fueron concebidos, son los siguientes:

- **Imperativos:** Emplean instrucciones como unidad de trabajo de los programas, es decir, la unidad de trabajo es la sentencia (acción). Ejemplos: Cobol, Pascal, C, Ada.
- **Declarativos:** Los programas se construyen mediante descripciones de funciones o expresiones lógicas, es decir, describen el problema a solucionar. Ejemplos: Lisp, Prolog.

Orientado a objetos: El diseño de programas se basa en los datos y su estructura. Tiene como unidad de proceso al objeto y en él se incluyen los datos (variables) y operaciones que actúan sobre ellos. Ejemplo: Smalltalk, C++.

CAPÍTULO II: METODOLOGÍA

2.1 Marco metodológico

El proceso de desarrollo de software también es conocido como ciclo de vida. “Un ciclo de vida es el proceso que se sigue para construir, entregar y hacer funcionar el software, desde la concepción de la idea del sistema hasta su entrega y el desuso” (Gómez Fuentes, Cervantes Ojeda, & González Pérez, 2019).

Es importante saber que en la actualidad existen diferentes metodologías o ciclos de vida del desarrollo de software y dependerá del grado de experiencia del ingeniero de software el uso de ellas en cada proyecto.

2.1.1 Modelo de desarrollo de software

Se define como una representación abstracta del proceso de desarrollo de software y determina el orden en el que se llevan a cabo las actividades del proceso de desarrollo de software. También se define como el procedimiento que se sigue durante el proceso de desarrollo de un sistema y a este también se le denomina paradigma del proceso.

En la actualidad existen varios modelos de desarrollo de software, cada uno con enfoques y metodologías distintas. “El modelo indica el orden de las etapas del desarrollo del software y proporciona un criterio para comenzar, para continuar a la siguiente etapa y para finalizar” (Gómez Fuentes, Cervantes Ojeda, & González Pérez, 2019).

El modelado de sistemas es un elemento fundamental en el desarrollo de software y establece la base del ciclo de vida, “este empieza con un enfoque general y luego del análisis inicial se descompone detalladamente estableciendo un proceso o mediante un flujo con entradas y salidas, componentes, supuestos, restricciones, entre otros” (Durand, 2017).

La elección de un modelo de desarrollo depende de factores tales como la naturaleza del proyecto, los requisitos del cliente y la flexibilidad requerida.

2.1.2 Metodologías tradicionales

Estas metodologías hasta hace algún tiempo eran las más utilizadas por las empresas de desarrollo de software con la finalidad de organizar a sus equipos de trabajo y facilitar las tareas, con el tiempo han ido evolucionando y mejorando. Suelen seguir un enfoque secuencial y planificado.

“Las metodologías ágiles se han impuesto sobre las metodologías tradicionales, así lo indica el último estudio de Project Manager Institute (PMI), que señala que el 71 % de las empresas de ingeniería de software utiliza estas metodologías” (Universidades, 2020).

2.1.2.1 Waterfall (cascada)

Definición

Un enfoque lineal y secuencial que sigue una estructura de fases fijas como requisitos, diseño, implementación, pruebas y mantenimiento.

Características

1. El resultado de cada fase consiste en uno o más documentos que se autorizaron.
2. La siguiente fase no debe comenzar sino hasta que termine la fase previa.
3. El proceso de software no es un simple proceso lineal, sino que implica retroalimentación de una fase a otra.
4. Debido a los costos de producción y aprobación de documentos las iteraciones suelen ser onerosas e implicar un rediseño significativo.
5. Al final de cada fase el personal técnico y los usuarios tienen la oportunidad de revisar el progreso del proyecto.
6. El resultado de todo el proceso solo se comprueba al final, por lo que es difícil solucionar errores a causa de su aparición cuando ya ha terminado el proyecto.
7. Las fases del proyecto no se mezclan ya que está muy organizado.

Diagrama

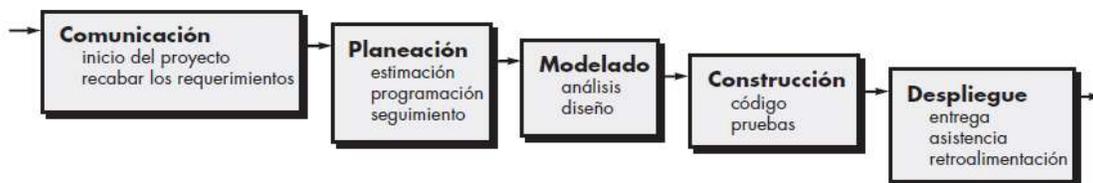


Figura 11 - Modelo de la cascada. Obtenido de (Pressman, 2010).

2.1.2.2 Prototipado

Definición

Implica la creación de versiones prototipo del software para comprender y refinar los requisitos del usuario antes de la implementación completa.

Características

1. Reduce el riesgo de generar productos que no satisfagan los requisitos.
2. Reduce costos y aumenta las posibilidades de éxito.
3. Una vez identificados todos los requisitos mediante el prototipo, se construye el producto de ingeniería.
4. Deben desarrollarse versiones funcionales rápidamente.
5. Es de gran importancia la interfaz de usuario.
6. El cliente, al ver el prototipo en operación puede creer que el sistema está completo.
7. El desarrollador puede construir el sistema solamente ampliando el prototipo, lo que disminuye la calidad del producto final.
8. Centrado en la representación de aquellos aspectos del software que serán visibles para los usuarios finales.
9. Sirve para la venta ya que el prototipo es considerado una maqueta.
10. El diseño rápido siempre se desecha.

Diagrama

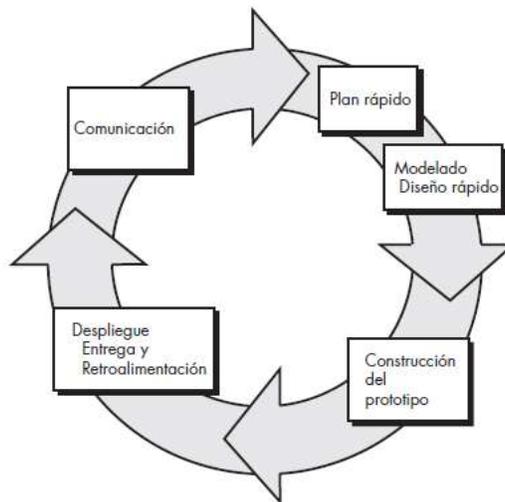


Figura 12 - El paradigma de hacer prototipos. Obtenido de (Pressman, 2010).

2.1.2.3 Espiral

Definición

Es un modelo evolutivo que bien se acopla con la naturaleza de hacer prototipos, incorpora elementos del modelo en cascada con iteraciones y especial énfasis en la identificación y mitigación de riesgos a lo largo del proyecto. Se adapta fácilmente para emplearse a lo largo de todo el ciclo de vida de una aplicación, desde la etapa de desarrollo del concepto hasta la fase de mantenimiento.

Características

1. El enfoque que tiene es cíclico para el crecimiento incremental del grado de definición del sistema y su implementación.
2. Es un conjunto de puntos de referencia de anclaje puntual para asegurar el compromiso del participante con soluciones factibles y mutuamente satisfactorias.
3. El software se desarrolla en una serie de entregas evolutivas y durante las primeras iteraciones, lo que se entrega puede ser un modelo o prototipo.
4. A diferencia de otros modelos del proceso que finalizan cuando se entrega el software, el modelo espiral puede adaptarse para aplicarse a lo largo de toda la vida del software.
5. Es un enfoque realista para el desarrollo de sistemas a gran escala.
6. Usa los prototipos como mecanismo de reducción de riesgos, pero, más importante, permite aplicar el enfoque de hacer prototipos en cualquier etapa de la evolución del producto.
7. Máximo control sobre los costos, recursos y la calidad del proyecto.
8. Relación de por vida con el cliente.
9. Trabaja con regiones de tareas.

Diagrama

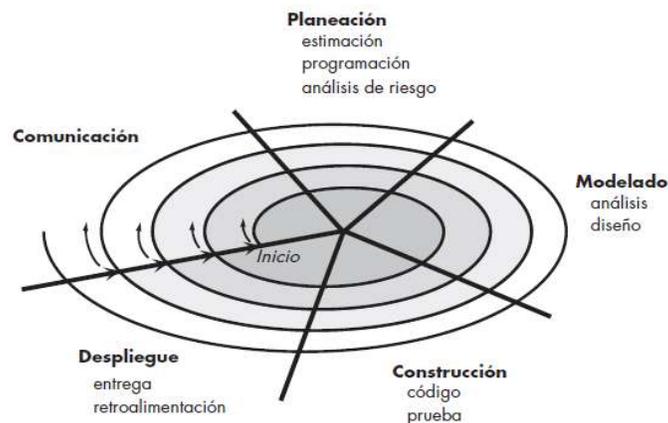


Figura 13 - Modelo de espiral común. Tomado de (Pressman, 2010).

2.1.2.4 Modelo de desarrollo concurrente

Definición

Este modelo también es conocido como desarrollo concurrente de software (CSD), se centra en la ejecución simultánea de actividades en lugar de seguir un enfoque secuencial. A veces es llamado ingeniería concurrente.

Permite que un equipo de software represente elementos iterativos y concurrentes de cualquiera de los modelos de proceso, define una serie de eventos que desencadenan transiciones de un estado a otro para cada una de las actividades, acciones o tareas de la ingeniería de software.

Características

1. Apropiado para proyectos en los que se involucran varios equipos de trabajo.
2. Aplicable a todos los tipos de desarrollo de software y proporciona un panorama apropiado del estado actual del proyecto.
3. En lugar de confinar las actividades, acciones y tareas de la ingeniería de software a una secuencia de eventos, define una red del proceso.
4. Todas las actividades de ingeniería de software existen de manera concurrente, pero se hallan en diferentes estados.
5. Los eventos generados en cierto punto de la red del proceso desencadenan transiciones entre los estados.
6. Un estado es algún modo de comportamiento observable externamente.

Diagrama

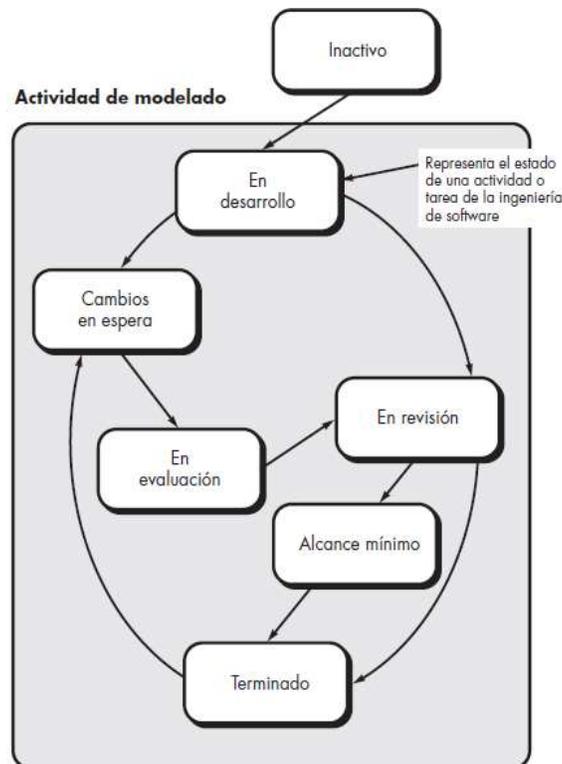


Figura 14 - Modelo de proceso concurrente. Tomado de (Pressman, 2010).

2.1.3 Modelos de proceso especializado

Considerando a (Cevallos, 2015) “Un modelo de proceso es un cúmulo de actividades, tareas y acciones que se realizan con el propósito de alcanzar el desarrollo completo de un proyecto de software”.

En la actualidad existen distintos modelos de proceso tales como los prescriptivos que son útiles cuando los requerimientos se encuentran completamente definidos, los especializados los cuales tienen características de uno o más modelos tradicionales y son utilizados cuando el enfoque del proyecto se encuentra bien definido.

2.1.3.1 Modelo de desarrollo basado en componentes

Definición

Este modelo se centra en la construcción de sistemas utilizando componentes reutilizables. Los COTS (componentes comerciales de software), desarrollados por vendedores que los ofrecen como productos, brindan una funcionalidad con interfaces definidas que permiten que dichos componentes se puedan integrar en el software que se va a construir.

Posee muchas características del modelo de desarrollo en espiral, su naturaleza es evolutiva y demanda un enfoque iterativo para poder crear software. Sin embargo, este modelo de desarrollo construye aplicaciones a partir de los fragmentos de software prefabricados.

Estos modelos buscan mejorar la eficiencia y la calidad del software al aprovechar la reutilización de componentes, además facilitan la escalabilidad de los sistemas.

Características

1. Se investigan y evalúan productos disponibles basados en componentes.
2. Se consideran los aspectos de integración de los componentes.
3. Se diseña una arquitectura del software para que reciba los componentes.
4. Se integran los componentes en la arquitectura.
5. Se efectúan pruebas exhaustivas para asegurar la funcionalidad apropiada.
6. Se basa en la información desarrollada como parte del modelo de requerimientos.
7. Trabaja con bibliotecas de clase.
8. Principio de segregación de la interfaz, es mejor tener muchas interfaces específicas del cliente que una sola de propósito general.
9. Ensamblaje de componentes.

Diagrama

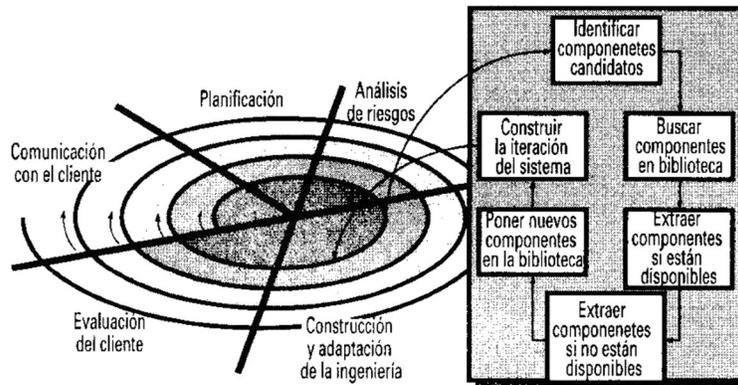


Figura 15 - Modelo de desarrollo basado en componentes. Tomado de (htt).

2.1.4 Metodologías ágiles

Es un enfoque de desarrollo que se centran en la flexibilidad, la colaboración y la entrega iterativa de productos, combina una filosofía con un conjunto de lineamientos de desarrollo de software.

Citando a (Del Carmen Ramírez & Flórez Fuentes, 2014), esta filosofía se centra en:

- La satisfacción del cliente y en la rápida entrega de software incremental;
- Los equipos pequeños y muy motivados para efectuar el proyecto;
- Los métodos informales;
- Los productos de trabajo con mínima ingeniería de software y
- La sencillez general en el desarrollo.

Los lineamientos de desarrollo enfatizan la entrega sobre el análisis y el diseño y la comunicación activa y continua entre desarrolladores y clientes.

Estas metodologías comparten valores clave, tales como la adaptabilidad al cambio, la entrega rápida y la colaboración constante con los stakeholders (partes interesadas). La elección de la metodología depende del contexto y los requisitos específicos del proyecto.

2.1.4.1 XP

Definición

XP o Programación Extrema, es el enfoque más utilizado del desarrollo de software ágil. Es una metodología que se enfoca en mejorar la calidad del software y la capacidad de respuesta a los cambios en los requisitos del cliente. Se basa en el enfoque orientado a objetos como paradigma preferido de desarrollo.

Esta metodología engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades principales que son la planeación, el diseño, codificación y pruebas.

XP busca la agilidad y la adaptabilidad mediante la incorporación de prácticas que permiten a los equipos responder de manera eficiente a los cambios y entregar software de alta calidad de manera consistente.

Características

1. Pone énfasis en la colaboración estrecha, pero informal (verbal) entre los clientes y desarrolladores.
2. Establece metáforas para comunicar conceptos importantes en la retroalimentación continua y evita la documentación voluminosa.
3. Restringe a los desarrolladores para que diseñen solo para las necesidades inmediatas, en lugar de considerar las del futuro.
4. La retroalimentación se obtiene de tres fuentes: el software implementado, el cliente y otros miembros del equipo de software.
5. Usa la prueba unitaria como táctica principal de pruebas.
6. Ideal para programación orientada a objetos.
7. Usa las tarjetas CRC (Clase – Responsabilidad – Colaboración) en las que se lleva registro de los requerimientos del sistema para el diseño.

Diagrama

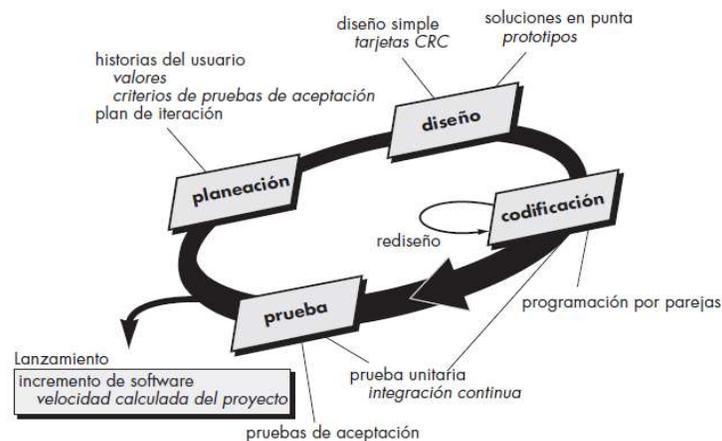


Figura 16 - Proceso de XP. Obtenido de (Pressman, 2010).

2.1.4.2 RUP

Definición

Marco de trabajo de desarrollo de software que proporciona directrices y mejores prácticas para la planificación, ejecución y control de proyectos de software. RUP, por las

siglas de Rational Unified Process o Proceso Unificado Racional, es un modelo de proceso moderno que derivó del trabajo sobre UML y el proceso asociado de desarrollo de software unificado.

Es un modelo de cuatro fases discretas en el proceso de software: Inicio, elaboración, construcción y transición. Cada una de las fases tiene objetivos específicos y actividades asociadas. Sin embargo, a diferencia del modelo de cascada, donde las fases se igualan con actividades del proceso, las fases en el RUP están estrechamente vinculadas con la empresa que con las preocupaciones técnicas.

Características

1. Desarrollo iterativo.
2. Administración de requisitos.
3. Uso de arquitectura basada en componentes.
4. Control de cambios.
5. Modelado visual del software.
6. Verificación de la calidad del software.
7. Pretende impulsar mejores prácticas en ingeniería de software, de forma que se adapte a cualquier proyecto.
8. Control de calidad.
9. Es una implementación del desarrollo espiral.

Diagrama



Figura 17 - Fases en el Proceso Unificado Racional. Tomado de (Sommerville, 2011).

2.1.4.3 SCRUM

Empleando las palabras de (Schwaber & Sutherland, 2020), “scrum es un marco de trabajo que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos”. Scrum es simple y se basa en la inteligencia colectiva de las personas que lo utilizan.

A juicio de (Schwaber & Sutherland, 2020), “en lugar de proporcionar a las personas instrucciones detalladas, las reglas de scrum guían sus relaciones e interacciones. Scrum hace visible la eficacia relativa de las técnicas actuales de gestión, entorno y trabajo, de modo que se puedan realizar mejoras”.

Scrum se centra en la entrega iterativa e incremental de productos. Los aspectos clave de esta metodología se describen a continuación.

Roles

Scrum Team

Se conforma por un pequeño grupo de personas, este equipo consta básicamente de un scrum master, un product owner y desarrolladores. Es un equipo en el que no existen jerarquías ni subequipos, dentro de él existe cohesión y todos sus esfuerzos están centrados en un objetivo a la vez; el product objective.

Estos equipos son multifuncionales, es decir, cada miembro del equipo cuenta con las habilidades y capacidad técnica para crear valor en el producto de cada sprint. Estos equipos se autogestionan, lo que quiere decir que deciden quién hace qué, cuándo y cómo.

Afirma (Schwaber & Sutherland, 2020), “el scrum team es responsable de todas las actividades relacionadas con el producto, desde la colaboración de los interesados, la verificación, el mantenimiento, la operación, la experimentación, la investigación y el desarrollo, y cualquier otra cosa que pueda ser necesaria”.

El scrum team es empoderado por la organización y gestiona su propio trabajo y es el principal responsable de crear un incremento de valor en cada sprint.

Refiere (Schwaber & Sutherland, 2020) que “define tres responsabilidades específicas dentro del scrum team: Los developers, el product owner y el scrum master”.

Scrum Master

Facilita el proceso y elimina obstáculos para el equipo.

Afirma (Schwaber & Sutherland, 2020), que el scrum master “es responsable de lograr la efectividad del scrum team”.

Apoya al scrum team de la siguiente manera:

- Guiar a los miembros del equipo;
- Ayudar a enfocarse en crear incrementos de alto valor;
- Eliminar impedimentos para el progreso; y,
- Garantizar que todos los eventos de scrum se realicen y obtengan resultados positivos, sean productivos y que se mantengan dentro de los parámetros de tiempo.

Apoya al product owner de la siguiente manera:

- Ayuda a encontrar técnicas para una definición efectiva del product objective y la gestión del product backlog;
- Ayuda al scrum team a comprender la necesidad de tener elementos del product backlog claros y concisos;
- Ayuda a establecer una planificación empírica de productos para un entorno complejo; y,
- Facilitar de la colaboración entre los interesados.

Apoya a la organización de la siguiente manera:

- Liderar a la organización en su adopción de scrum;
- Asesorar a la organización para implementar scrum dentro de la misma;
- Ayudar a todos los interesados a aplicar un enfoque empírico para el trabajo; y,
- Eliminar las barreras entre los interesados y el scrum team.

Product Owner

Representa los intereses del cliente y gestiona el product backlog.

Refiere (Sola, 2022), que “es responsable de maximizar el valor del producto resultante del trabajo del scrum team” y la manera en que se hace, varía significativamente entre organizaciones, individuos y equipos.

De acuerdo con Guillem Hernández Sola; el product owner es responsable de la gestión del product backlog y sus tareas son, además:

- Desarrollar y comunicar claramente el objective product;
- Crear y comunicar con toda claridad cada elemento del product backlog;
- Ordenar estos elementos; y,
- Asegurarse de que el product backlog sea comprensible y transparente.

Debemos entender que el product owner es una persona y no un comité, ya que como afirma (Sola, 2022), “el product owner puede representar las necesidades de muchos interesados en el product backlog”.

Equipo de desarrollo

Profesionales de las TIC's como programadores, analistas de sistemas, diseñadores, ingenieros de software, etc. Son los encargados de presentar un incremento en cada sprint.

Sostiene (Schwaber & Sutherland, 2020) que “las habilidades específicas que necesitan los developers suelen ser amplias y variarán según el ámbito de trabajo”. Sin embargo, en todo momento tiene la responsabilidad de:

- Planificar el sprint, el sprint backlog;
- Inculcar calidad;
- Adaptar su plan cada día hacia el objetivo del sprint; y,
- Responsabilizarse mutuamente como profesionales.

Eventos

Sprint Planning

Se trata de una reunión al comienzo de cada sprint para planificar el trabajo.

Sostiene (West, s.f.) en su artículo que “la planificación de sprint es un evento de scrum que define qué se puede entregar en el sprint próximo y cómo se va a conseguir ese trabajo”. La planificación de sprints se lleva a cabo con la participación del equipo scrum y tiene como objetivo definir lo que se puede entregar en cada sprint y cómo se hará para conseguirlo tal como se muestra en la figura 18.



Figura 18 - Sprint Planning – Tomado de (Ferreyra, s.f.)

Daily Scrum

Consiste en una reunión diaria de 15 minutos en la que se trabaja con el scrum team para sincronizar las tareas y lograr la meta del sprint y hacer ajustes en el backlog cuando sea necesario.

Sprint Review

Es una revisión del trabajo completado al final del sprint (máximo 4 horas), permite mostrar el progreso al cliente y hacer una retroalimentación de aquellos que tienen interés en el proyecto. Marca la conclusión del sprint, se revisa el incremento y se muestra el producto funcionando.

Sprint Retrospective

Consiste en hacer una reflexión sobre el sprint y realizar mejoras al proceso scrum, su objetivo principal es mejorar las habilidades y la productividad del equipo, además de la calidad en el producto.

Artefactos

Product Backlog

Es una lista de todas las funcionalidades pendientes de realizar, es decir, requerimientos, además es la fuente de información sobre el producto en scrum.

Enfatiza (Raeburn, 2022) que “llamamos product backlog a una lista de funciones y elementos, ordenados según las prioridades, que son necesarios para cumplir con los objetivos y las expectativas del proyecto”.

Sprint Backlog

Son los elementos seleccionados del product backlog para el sprint actual (figura 19), consiste en una lista de tareas que han sido identificadas por el scrum team y las cuales tienen que ser completada durante cada Sprint.

Como afirma (Integrat, s.f.); “el sprint backlog es representado a través de un tablero de tareas”.

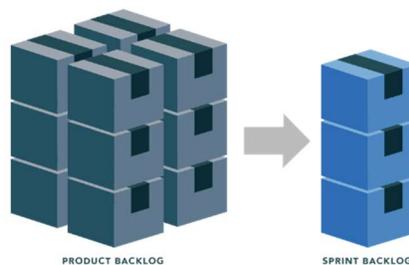


Figura 19 - Sprint Backlog. Tomada de (Karlsson, s.f.).

Incremento

Es un producto funcional y potencialmente entregable al final de cada sprint, es decir la suma de cada uno de los elementos del product backlog completados durante el actual sprint y los incrementos de todos los sprints hechos con anterioridad.

Sprints

Es el núcleo central de la metodología, son periodos fijos de tiempo (normalmente de dos a cuatro semanas), en las cuales se desarrolla y entrega un incremento del producto. Puede considerarse un proyecto con una duración no mayor a un mes, cuando su duración es mayor, puede cambiar la definición de lo que se está construyendo.

Priorización

El product owner prioriza el product backlog según el valor que aportará al cliente.

Adaptabilidad (flexibilidad)

Scrum se adapta a los cambios de requisitos del cliente durante el desarrollo, permitiendo así una mayor flexibilidad.

Transparencia

La información sobre el trabajo y el progreso es transparente para todos los miembros del equipo, es decir, todos sabemos lo que está pasando.

Enfoque en la colaboración

Fomenta la colaboración constante entre los miembros del equipo y promueve la comunicación abierta.

CAPÍTULO III: ANÁLISIS

3.1 Análisis del Sistema

En Calidad Textil es necesario un sistema de control de inventario ya que, aunque cuenta con el manejo de sus entradas y salidas de forma manual, necesita sistematizar sus procesos para que sean eficientes.

Actualmente el manejo de inventario se lleva a cabo manualmente en una tabla de Excel en donde se toma registro de los movimientos de los productos, por lo que este proceso es ineficiente para el tipo de información que actualmente se maneja y la creciente expansión del negocio.

El problema se origina debido a que cuando un empleado realiza una entrada o salida de algún producto, debe registrar en la bitácora el movimiento realizado; a pesar de esto, la mayoría de las veces no existe coincidencia entre el inventario lógico y el físico, por lo tanto cuando el administrador desea consultar el inventario, se pierde mucho tiempo en consultar uno a uno los registros de entrada y salida, para encontrar el error, que generalmente se debe a la duplicidad de registros o al extravío de éstos.

3.1.1 Requisitos funcionales

RF1 - Autenticación de usuarios: Deberán identificarse para poder acceder al sistema, este podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad.

RF2 - Registrar usuarios: El sistema permitirá al usuario (administrador) registrarse. El usuario debe proporcionar datos como: Usuario y contraseña.

RF3 - Datos maestros: El usuario deberá dar de alta el catálogo de proveedores junto a sus condiciones comerciales, catálogo de productos y su clasificación. El usuario contará con permisos a los catálogos de crear, modificar y eliminar.

RF4 - Operación: Todas las operaciones del día a día serán registradas en tablas del sistema denominadas transaccionales mediante el uso de transacciones las cuales contarán con validaciones y ejecuciones por cada transición que tenga acceso el usuario.

RF5 - Consultar información: El sistema ofrecerá al usuario información general acerca de las ventas.

- Consultar informes estadísticos: Muestra información general sobre las entradas y salidas de mercancías.

- Consultar operaciones: Muestra información a detalle por cada movimiento operativo de inventarios registrado en el sistema.
- Consultar proveedores: Muestra datos de cada proveedor junto a sus condiciones comerciales como precios y descuentos.
- Consultar existencias: Muestra en tiempo real el inventario disponible para ser ocupado en cada movimiento.

RF6 - Gestionar Reportes: Permite al administrador imprimir reportes de las operaciones de inventarios, mediante parámetros de búsqueda, realizadas con detalles y totales.

3.1.2 Requisitos no funcionales

Rendimiento

Asegurar que, al realizar un proceso o consulta no se vea afectado el desempeño de la base de datos ni el tráfico de la red.

Seguridad

- Asegurar a los distintos usuarios un buen desempeño, la seguridad y confiabilidad del sistema. Es decir, la información y registros almacenados, podrán ser actualizados y consultados de manera simultánea sin que esto afecte el tiempo de respuesta.
- Dar garantía de que toda la información y datos que se manejan son seguros y están bien resguardados de agentes externos.
- Facilitar y controlar el acceso a la información al personal autorizado.

Fiabilidad

Asegurar que la interfaz de usuario sea simple e intuitiva.

Disponibilidad

Asegurar la disponibilidad del sistema 24/7 para todos los usuarios.

Portabilidad

El sistema deberá ser implantado bajo la plataforma de Windows y Mac, considerando las especificaciones técnicas de hardware y software.

3.1.3 Perspectiva del producto

El sistema de gestión de inventarios funcionará con una arquitectura en monolito, este término se utiliza para describir sistemas donde la aplicación y la base de datos están combinadas o integradas en una única entidad, por tanto, es un enfoque arquitectónico en el desarrollo de software en el que todos los componentes y funcionalidades de una aplicación se encuentran dentro de un solo código base.

En este tipo de sistemas, la aplicación y la base de datos suelen estar acopladas y desplegadas juntas en un mismo entorno.

3.1.4 Funcionalidad del producto

- El sistema tendrá funciones tales como administración de catálogos maestros para administración de información base (proveedores, productos, clientes etc.)
- Contará con módulo de compras, el cual permitirá la captura de la adquisición de mercancías solicitada al proveedor y surtimiento al almacén.
- Administrará las existencias de los productos en sus respectivos almacenes, permitiendo el registro de transacciones operativas para flujos de inventarios.
- Relacionarán todos los datos para obtención de informes de movimientos, estadístico y evaluación del almacén.
- El sistema se registrará mediante la seguridad de información con roles por usuarios.

3.1.5 Especificación de requerimientos

Tabla 1 - Requisito Funcional 1.

Número de requisito	RF01
Nombre de requisito	Autenticación de Usuarios
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	El ingreso al sistema será mediante una pantalla de login donde se requiere el nombre del usuario y una contraseña. La pantalla de conexión deberá contar con la cadena de conexión a la instancia indicando el nombre de la base de datos a conectar; los datos de la cadena de conexión serán implícitos y no dependientes de captura por el usuario.

Tabla 2 - Requisito Funcional 2.

Número de requisito	RF02
Nombre de requisito	Registrar Usuarios
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	El sistema permitirá al usuario (Administrador) registrarse y este deberá proporcionar datos como: Usuario y contraseña.

Tabla 3 - Requisito Funcional 3.

Número de requisito	RF03
Nombre de requisito	Datos Maestros
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	La estructura de la DB contará con tablas en la DB de tipo catálogo, las cuales tendrán la capacidad de albergar datos maestros, mismos que serán ocupados como medios de captura por cada transacción registrada.

Tabla 4 - Requisito Funcional 4.

Número de requisito	RF04
Nombre de requisito	Operación
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	Todas las operaciones del día a día serán registradas en tablas del sistema denominadas transaccionales mediante el uso de transacciones las cuales contarán con validaciones y ejecuciones por cada transición que tenga acceso el usuario

Tabla 5 - Requisito Funcional 5.

Número de requisito	RF05
Nombre de requisito	Consultar Información
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	El sistema ofrecerá al usuario información general acerca de las ventas.

Tabla 6 - Requisito Funcional 6.

Número de requisito	RF06
Nombre de requisito	Gestionar Reportes
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	Permite al administrador imprimir reportes de las operaciones de inventarios, mediante parámetros de búsqueda, realizadas con detalles y totales.

Tabla 7 - Requisito Funcional 7.

Número de requisito	RF07
Nombre de requisito	Roles de Seguridad
Tipo	Restricción
Prioridad del requisito	Alta/Esencial
Descripción	El sistema contará con un repositorio de usuarios, a cada uno de ellos le será asignado un rol dentro del sistema, el cual será los objetos a los cuales tendrá acceso y las transacciones que podrá ejecutar.

Tabla 8 - Requisito Funcional 8.

Número de requisito	RF08
Nombre de requisito	Transacción de Ordenes de Compras
Tipo	Requisito
Prioridad del requisito	Media/Deseado
Descripción	En el funcionamiento transaccional, el usuario tendrá la opción de capturar órdenes de compra a proveedor, las condiciones comerciales deberán tomarse previamente del catálogo de proveedores correspondientes al módulo de Datos Maestros.

Tabla 9 - Requisito Funcional 9.

Número de requisito	RF09
Nombre de requisito	Transacción de Compras
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	Posterior al flujo de la orden de compra, el responsable del área de almacén podrá realizar la recepción total o parcial del contenido de la orden, afectando con ello el inventario en piso y obteniendo como resultado el costo promedio.

Tabla 10 - Requisito Funcional 10.

Número de requisito	RF10
Nombre de requisito	Transacción de Inventarios
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	Una vez ingresado el producto, el sistema contará con transacciones para manejo de existencias bajo los cuales el usuario podrá capturar si es una entrada o salida, clasificando el motivo que la origina.

3.1.6 Proceso de ingeniería de requerimientos

Adquisición y análisis de requerimientos

En esta fase se determinaron las necesidades específicas por parte del usuario final y que deberán ser cubiertas por el software que se desarrollará considerando en todo momento el alcance del proyecto; así mismo, se determinaron las restricciones de hardware y software.

Cada requerimiento fue trabajado de manera independiente con cada usuario según sea su desempeño en la empresa Calidad Textil. Con ello se obtiene una comprensión de la función y se recolecta información suficiente para el desarrollo de sus transacciones.

Alcance del proyecto

La comprensión del problema a resolver es la traducción en la práctica del alcance del proyecto, donde siempre es sugerido la construcción de software de fácil crecimiento.

Roles de Seguridad

Los usuarios a intervenir en el uso del sistema se enlistan según el nivel de seguridad que tendrán en el flujo del sistema, para cumplimiento de ello se desarrollará la herramienta denominada roles de seguridad (tabla 11) donde un rol será el conjunto de permisos y restricciones que se asocian como permisos a un usuario registrado en el software, acorde a ello será el acceso a la plataforma y lo que podrán desempeñar en ella.

Tabla 11 - Roles de Seguridad.

Rol	Funciones	Transacciones de Acceso
Administrador del Sistema	<ul style="list-style-type: none"> • Será la persona encargada de administrar el control de usuarios, asignación de roles y permisos en la plataforma. • Así como regular las actualizaciones al software y su mantenimiento. • Determinará el método de valuación del almacén. 	<ul style="list-style-type: none"> • Acceso total a transacciones sin permisos de desarrollador.
Administrador de Datos Maestros	<ul style="list-style-type: none"> • Responsable de la actualización de catálogos, datos maestros en el aplicativo garantizando que la información sea confiable. 	Administración de los catálogos: <ul style="list-style-type: none"> • Clientes. • Productos. • Proveedores.
Supervisor de Almacén	<ul style="list-style-type: none"> • Responsable del inventario físico en el almacén, recibe la mercancía del proveedor y gestiona las salidas de las mismas bajo los conceptos que la operación indique. 	<ul style="list-style-type: none"> • Recepción de compra de proveedor. • Ajuste de inventario entradas y salidas. • Inventario físico. • Cierre de fecha de operación. • Reporte de existencias y Kardex de inventario. • Tendrá facultad de autorizar las transacciones de inventarios afectando el almacén.
Capturista	Responsable de gestionar las entradas y salidas de inventario realizando una captura en el sistema sobre los movimientos realizados.	Tendrá acceso a las mismas transacciones de inventario que el supervisor de almacén, sin embargo, no tendrá facultad de afectar el inventario en piso. Su labor será únicamente de captura de información en el sistema. No tendrá acceso a reporte de Kardex de inventario.

Administración de Datos Maestros

El éxito en los resultados arrojados por un sistema informático reside en la integridad de la información de los datos maestros, ello constituye información confiable y fidedigna.

El software deberá contar con un módulo exclusivo para la administración de los catálogos denominados datos maestros, bajo el cual se registrarán las transacciones operativas de inventarios realizadas por los usuarios con rol de capturista y supervisores.

Contará con la administración de catálogos enlistados:

- **Proveedores:** Deberá existir un catálogo de proveedores almacenando sus datos de contacto y dirección, RFC y cuentas bancarias considerando a futuro manejo de cuentas por pagar, el dato relevante dentro de la administración de proveedores que contará el sistema será la administración de precios de proveedor en conjunto con las condiciones comerciales.
- **Productos:** Para un correcto análisis de las operaciones realizadas en un sistema, es necesario contar con un catálogo de clasificación de los productos, donde el software a desarrollar considerará factores como línea, sublínea, marca, fabricante y presentación entre otros factores relevantes. Cada clasificación se administra en tablas del sistema tipo catálogo con sus respectivos id de clasificación y será el listado de mercancías operables y con su respectiva clasificación, bajo el cual se utilizará en cada transacción realizada en el software.
- **Clientes:** Deberá existir un catálogo de clientes con información básica de contacto de cada uno de ellos.

CAPÍTULO IV: DISEÑO

4.1 Diagramas de Base de Datos

Modelo Relacional

El modelo relacional (figura 20), nos ayudará a definir la implementación lógica de la información de Calidad Textil mediante una serie de tablas, campos, restricciones, relaciones entre las mismas que nos ayudará a identificar la semántica de Calidad Textil.

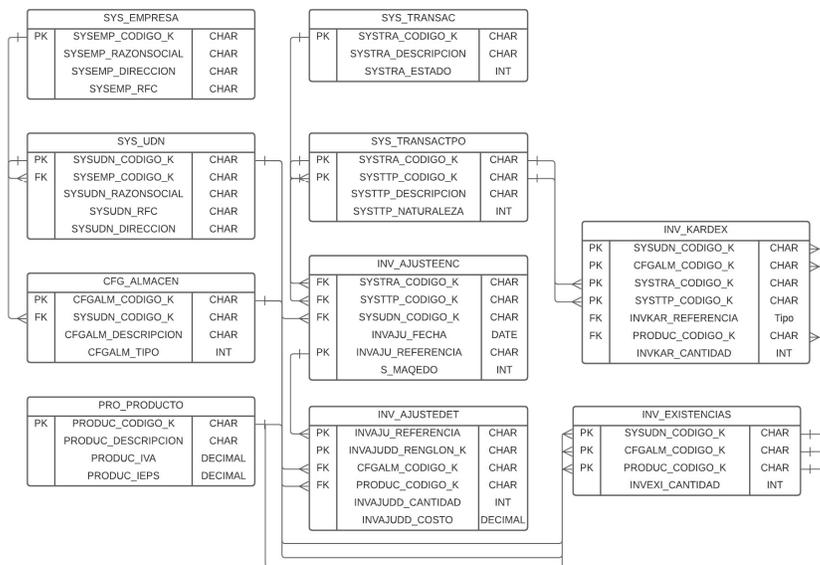


Figura 20 - Modelo Relacional. Elaboración propia.

Diagrama Entidad-Relación

Un diagrama o modelo entidad relación (ERD), es un tipo de diagrama de flujo que ilustra cómo las "entidades", se relacionan entre sí dentro de un sistema (figura 21).

El diagrama ER es usado para diseñar o depurar bases de datos relacionales en los campos de ingeniería de software, y en este caso sistemas de información empresarial.

Las tablas se unen entre sí a través de Id denominadas llaves foráneas, una de las ventajas contra SGBDs tradicionales es que permite en el diseño establecer las reglas de relación para integridad de la información.

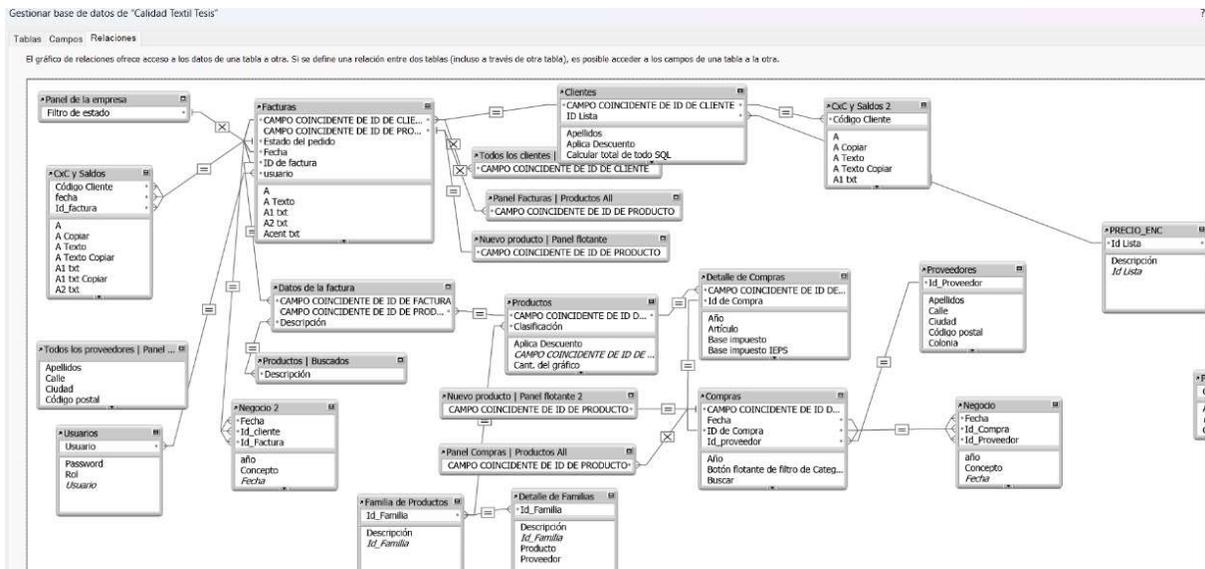


Figura 21 - Diagrama E-R. Elaboración propia.

Tablas y campos

La lógica empleada en el diseño de la estructura de la base de datos es con las siguientes tablas:

- **Panel de la empresa:** La cual funcionará como dashboard principal para funcionamiento de inteligencia de negocio.
- **Clientes:** Tabla tipo catálogo la cual almacenará la información y características de negocio de cada cliente.
- **Productos:** Tabla tipo catálogo la cual almacenará la información y características de venta de los productos, configuración de impuestos y formas de venta. Esta tabla está diseñada particularmente bajo el concepto tabla control para administración del inventario.
- **Compras / Detalle de compras:** Tablas transaccionales maestra y esclava la cual tiene el propósito de registrar las compras realizadas a proveedores para ingreso del producto al inventario.
- **CxC y Saldos:** Tablas de tipo control para administración de las cuentas por cobrar en acciones de venta con clientes a crédito.
- **Negocio:** Tabla tipo DataMart funcional para inteligencia de negocio y toma de decisiones.
- **Usuarios:** Tabla de catálogo donde se enlistan los distintos usuarios y sus respectivos roles en el uso del aplicativo.
- **Familia de productos y detalle de familias:** Tablas de tipo catálogo maestra y catálogo detalle respectivamente, tienen el propósito de agrupar productos por su naturaleza utilizado acorde a cada giro de negocio.

- **Listas de precios y detalle de listas de precios:** Tablas de tipo catálogo maestra y catálogo detalle respectivamente, su funcionamiento será administrar las diferentes listas de precios que se otorgan a cada cliente.

La figura 22 muestra los detalles de las tablas.

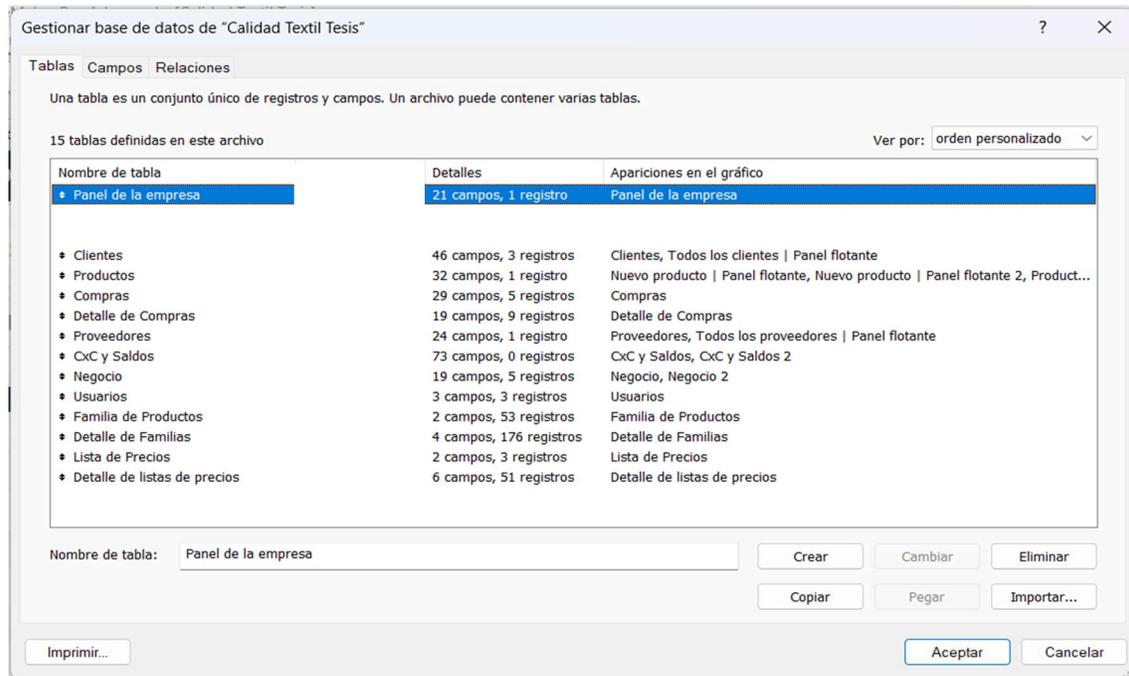


Figura 22 - Estructura de la base de datos (tablas).

La figura 23 muestra los campos de la tabla panel de la empresa.

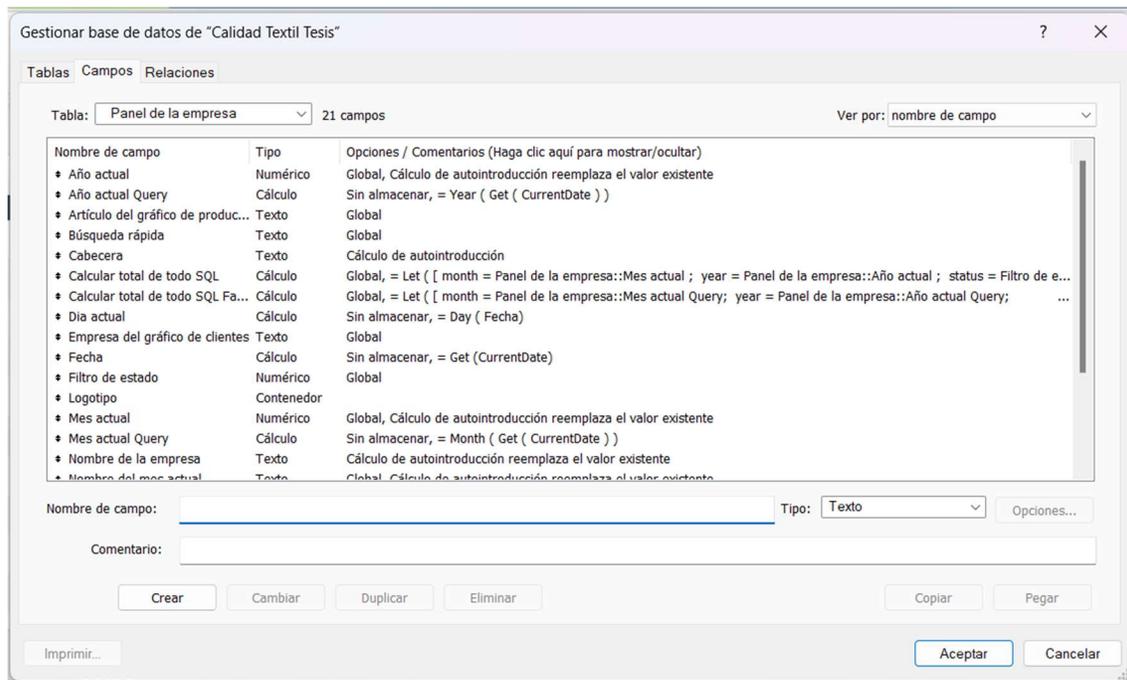


Figura 23 - Estructura de la base de datos (Campos/Panel de la empresa).

La figura 24 muestra los campos de la tabla clientes.

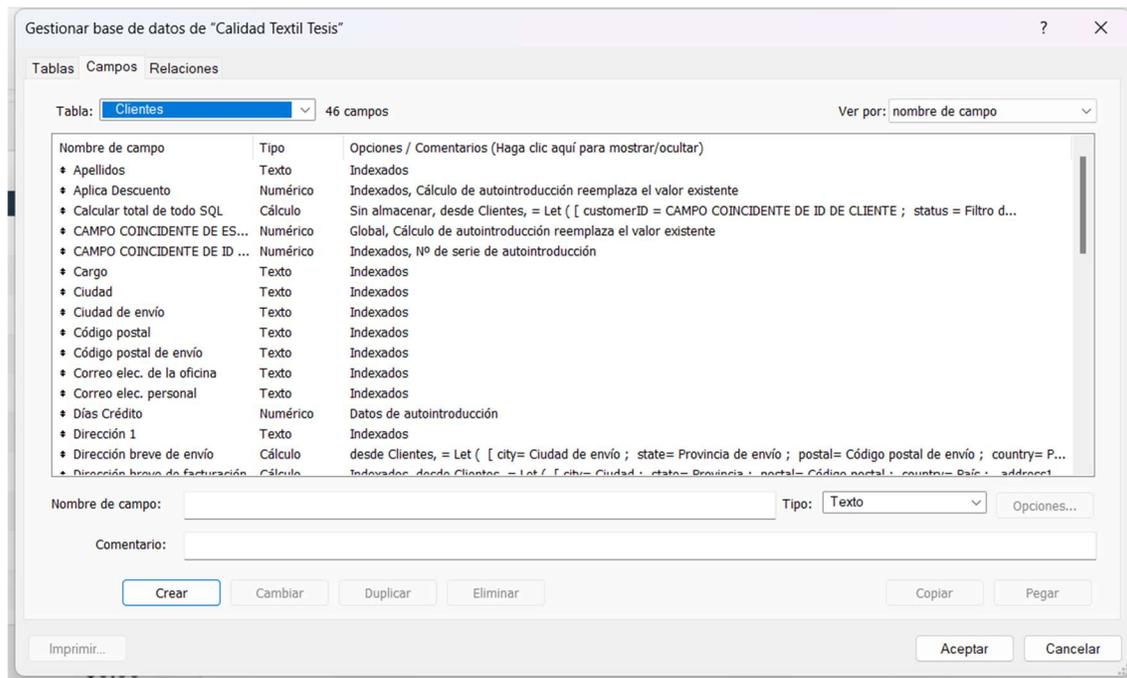


Figura 24 - Estructura de la base de datos (Campos/clientes).

La figura 25 muestra los campos de la tabla productos.

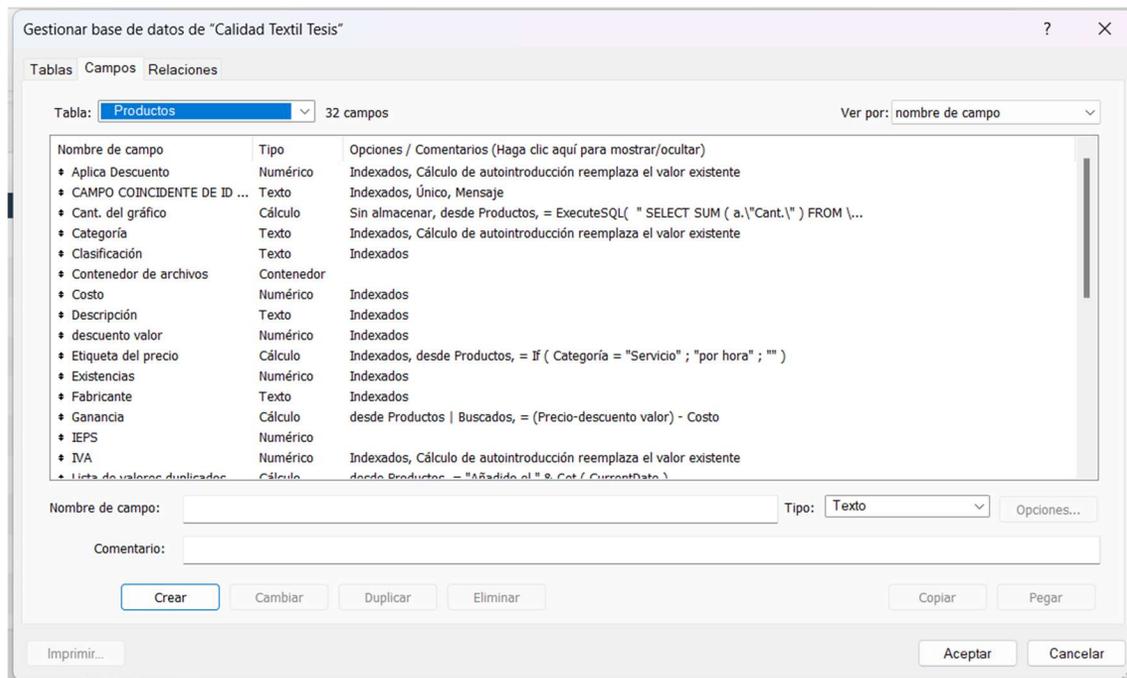


Figura 25 - Estructura de la base de datos (Campos/Productos).

La figura 26 muestra los campos de la tabla proveedores.

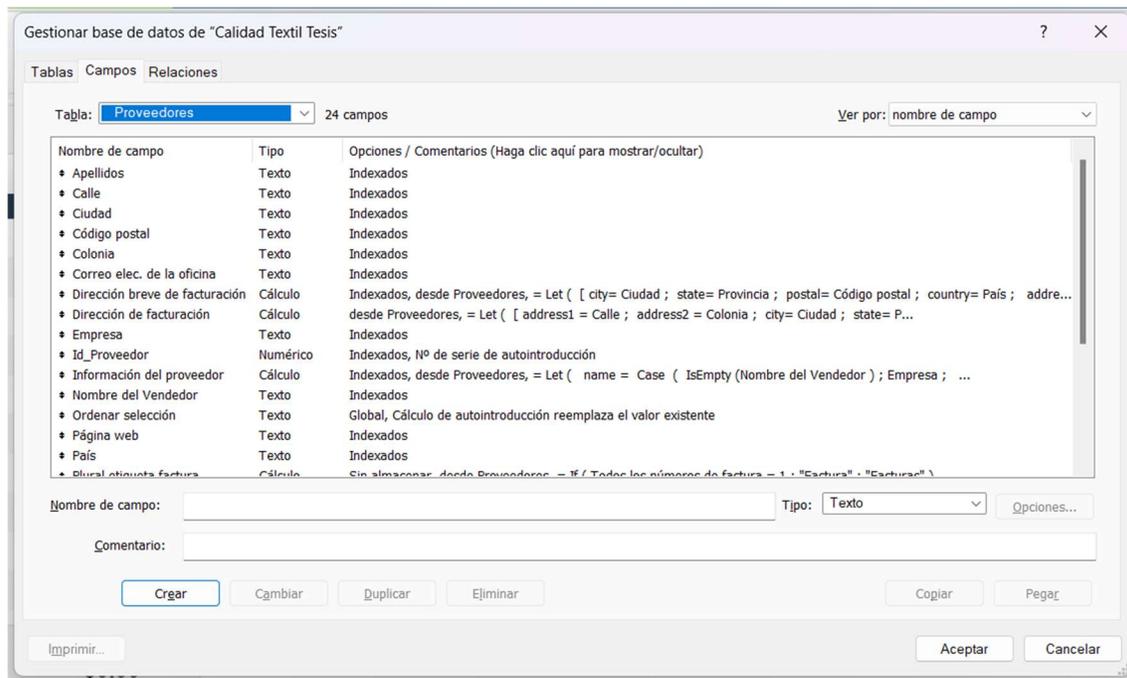


Figura 26 - Estructura de la base de datos (Campos/proveedores).

4.2 Diagramas UML

Casos de uso

Ayudará a definir una secuencia de acciones que da lugar a un resultado de valor observable y proporcionará una estructura para expresar requisitos funcionales en el contexto de procesos empresariales del sistema de información de inventario tal y como se muestra en la figura 27.

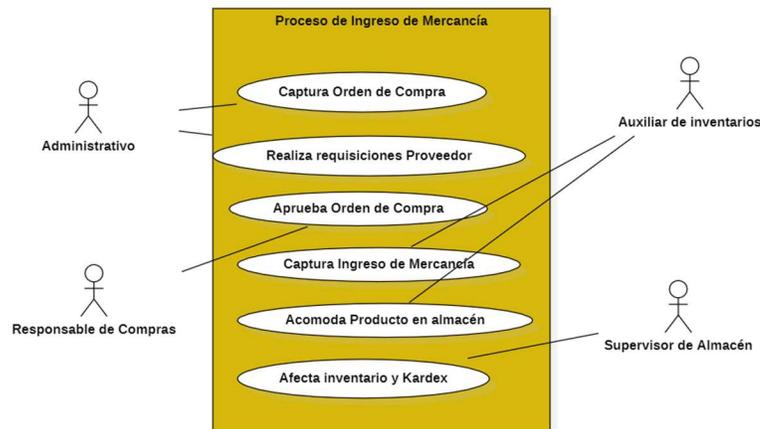


Figura 27 - Caso de uso. Elaboración propia.

Diagrama de Actividades

El flujo de control se desencadena al completar actividades dentro del sistema.

Usaremos un diagrama de actividad para describir cómo se coordinan las actividades de inventario en Calidad Textil para proporcionar un mejor control de los productos, entradas y salidas que existen. Estos diagramas nos muestran cómo los eventos de un caso de uso se relacionan entre sí o cómo unas colecciones de casos de uso se coordinan para representar un flujo de trabajo como se muestra en la figura 28.

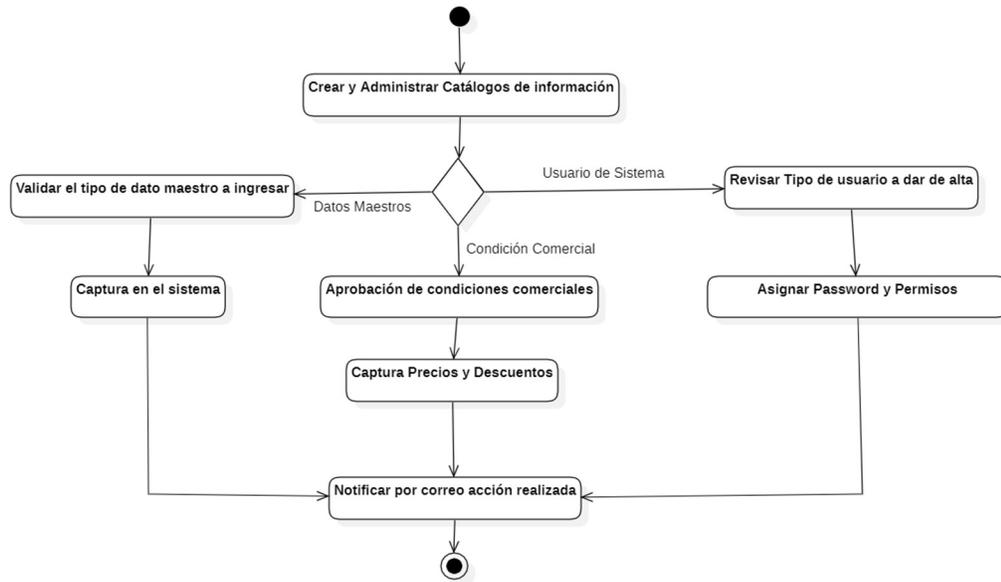


Figura 28 - Diagrama de actividades. Elaboración propia.

Diagrama de Estados

El diagrama de estados (figura 29), tiene diferentes usos:

- Ilustrar escenarios de casos de uso en un contexto de negocios para inventario.
- Mostrar el comportamiento general del sistema.

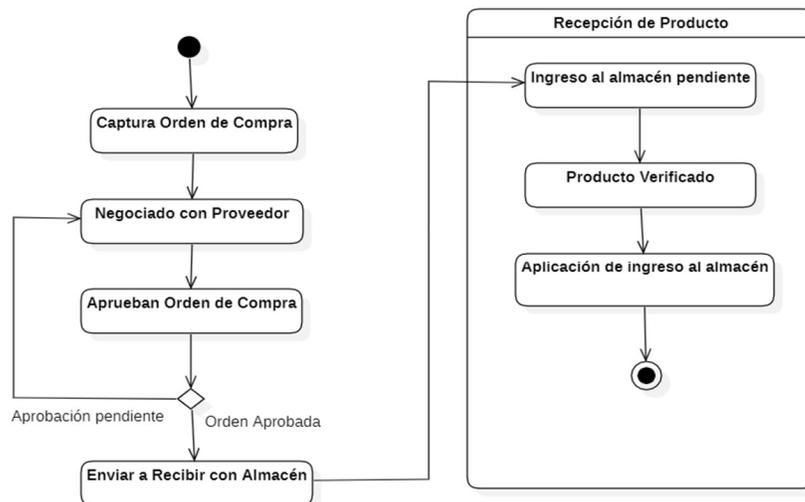


Figura 29 - Diagrama de estados. Elaboración propia.

Diagrama de Secuencias

En este diagrama (figura 30), nos centraremos específicamente en líneas de vida o en los procesos y objetos que existen simultáneamente, y los mensajes intercambiados entre ellos para ejecutar una función antes de que la línea de vida termine.

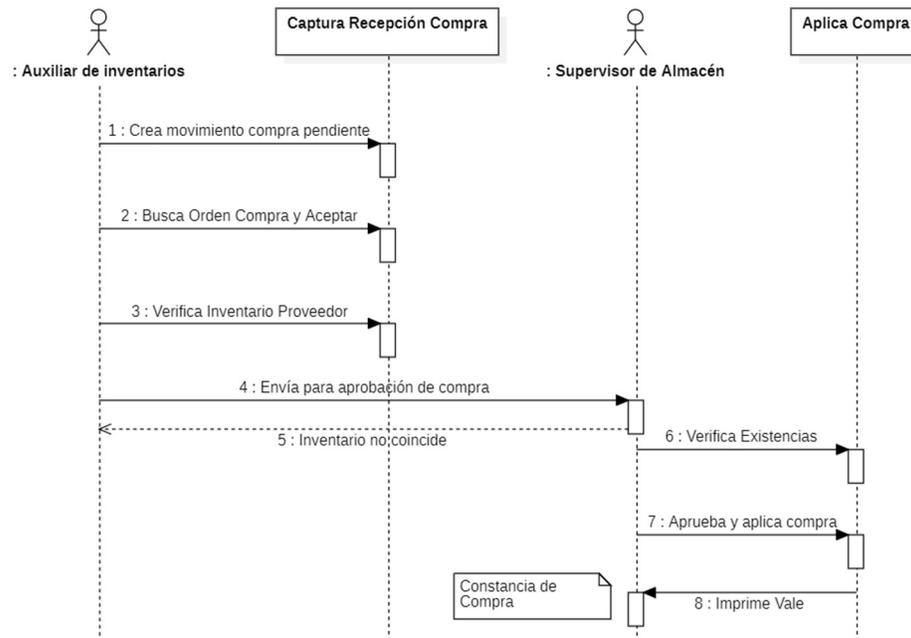


Figura 30 - Diagrama de secuencias. Elaboración propia.

Diagrama de Colaboraciones

En el diagrama de colaboración (figura 31), representaremos las relaciones entre los objetos. El diagrama de colaboración ilustra cómo se intercambian mensajes entre clases y objetos. Se genera para cada proceso del sistema y se relaciona con la fase de desarrollo existente.

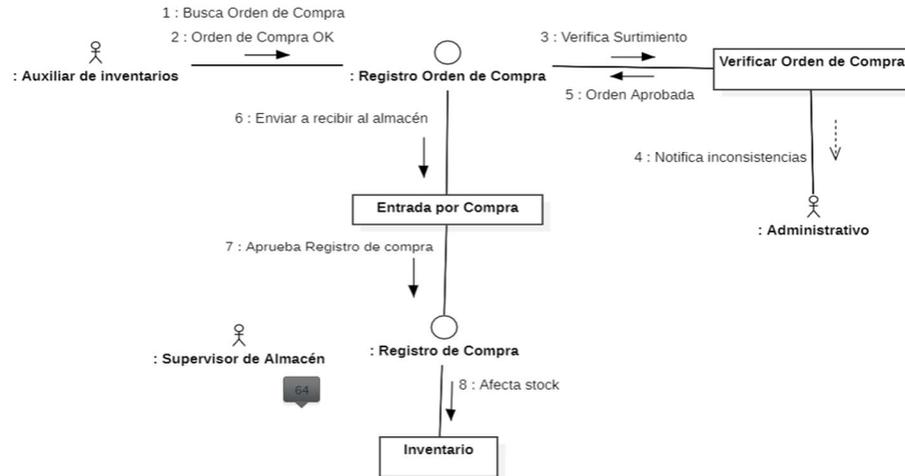


Figura 31 - Diagrama de Colaboraciones. Elaboración propia.

4.3 Interfaces gráficas de usuario

4.3.1 Login

Es la pantalla donde el usuario ingresará su nombre de usuario y contraseña para acceder al sistema de gestión de inventarios.

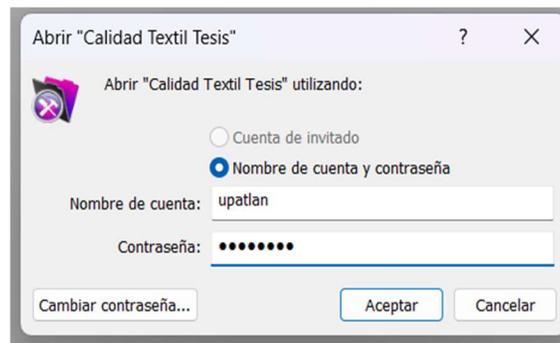


Figura 32 - Ventana de login.

Figura 33 - Ventana cambiar contraseña.

4.3.2 Pantalla de clientes

Es la pantalla de navegación que representará el front para el usuario donde un usuario con los suficientes privilegios, en la mayoría de los negocios, un rol de MasterData para captura de la información básica de cada cliente, una página web, dirección de sus domicilios de envío y lo más importante su configuración de las reglas de negocio bajo el cual se comportará cada cliente.

Figura 34 - Ventana de clientes.

4.3.3 Pantalla de productos

Similar al funcionamiento que se tendrá con la pantalla de navegación de los clientes, en dicha pantalla contará con la administración del portafolio de productos, almacenando sus datos principales, la familia a la cual pertenece y su configuración de venta tal como descuentos e impuestos.

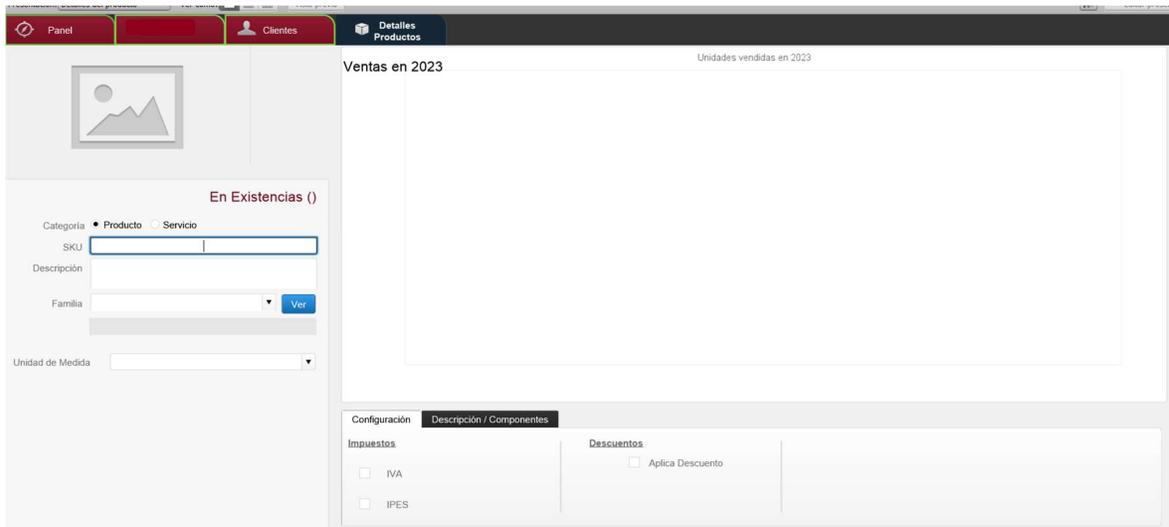


Figura 35 - Ventana de productos.

4.3.4 Familia de productos

Concepto de clasificación y agrupación de productos acorde a un tipo de telas, una familia permite la organización y manipulación de las telas donde las variantes son mínimas.

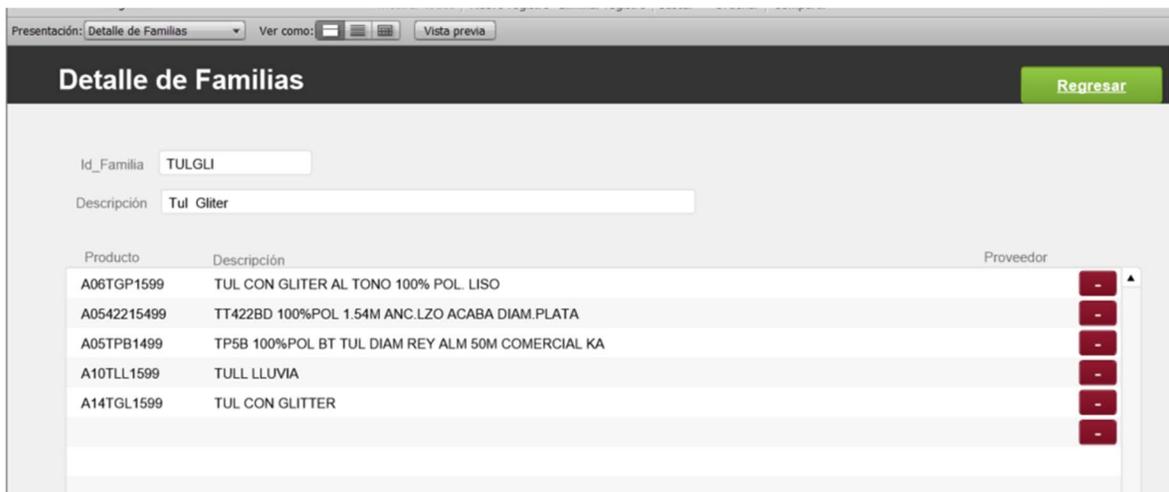


Figura 36 - Ventana familia de productos.

Al configurar un producto se especificará la familia a la cual pertenece y a través de un botón será posible ver el detalle de la familia seleccionada.

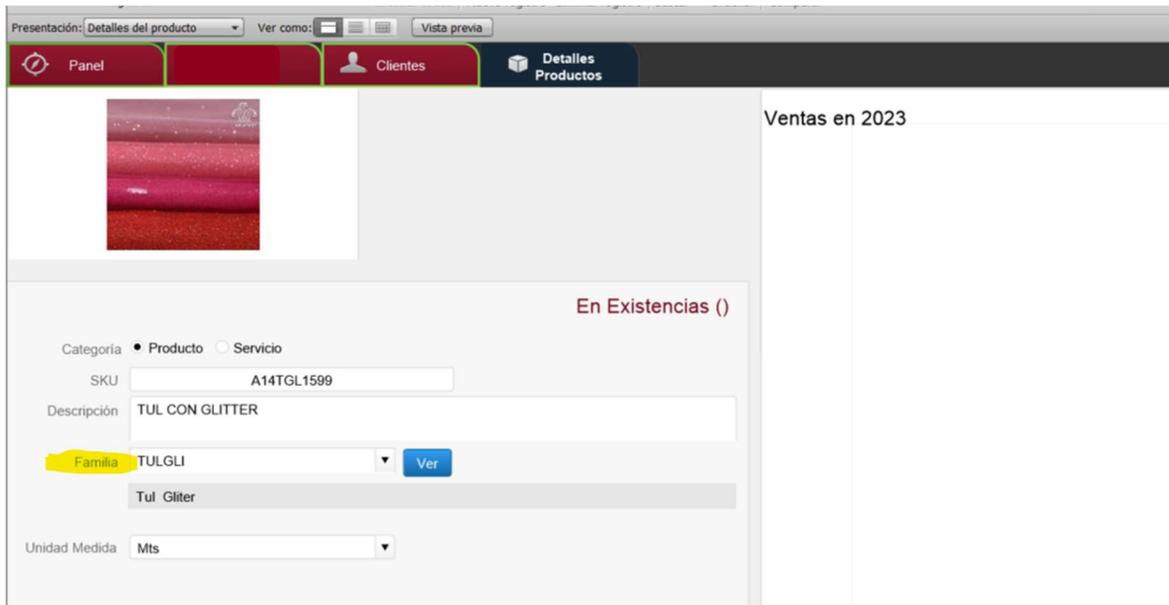


Figura 37 - Ventana detalle familia de productos.

4.3.5 Lista de precios

Se conformará de dos tipos de tablas; catálogo encabezado y detalle, en la cual el sistema permitirá configurar N listas de precios para abarcar distinto sector de mercado y así ofrecer precios competitivos según el canal de venta.

En la misma lista de precios se podrá configurar un descuento a cada producto, el sistema permitirá manejar dos tipos de descuentos, el denominado comercialmente como “descuento oculto” el cual se configura en la misma lista de precios y permite llevar el control sobre el precio original y el descuento aplicado; dicho descuento no se reflejará de manera desglosada en la venta ya que usualmente se ocupa por temporadas.

Producto	Descripción	Precio Original	Descuento	Precio Venta
1	A06TGP1599 TUL CON GLITER AL TONO 100% POL. LISO	30	\$0	30
1	A0542215499 TT422BD 100%POL 1.54M ANC.LZO ACABA DIAM.PLATA	32	\$0	32
1	A05TPB1499 TP5B 100%POL BT TUL DIAM REY ALM 50M COMERCIAL KA	29.5	\$0	29.5
1	A10TLL1599 TULL LLUVIA	48.5	\$10	38.5
1	A14TGL1599 TUL CON GLITTER	29.5	\$0	29.5

Figura 38 - Ventana Lista de precios.

A diferencia del descuento convenido comercialmente, este sí es público desglosándose en la venta.

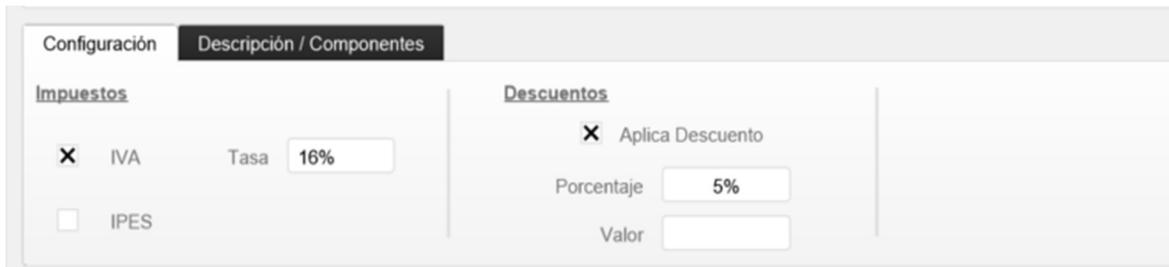


Figura 39 - Ventana configuración de impuestos y descuento.

4.3.6 Gráfica de venta de productos

En el récord de productos se mostrará una gráfica donde se visualizarán las ventas por meses de cada producto, se podrá realizar la búsqueda sólo en meses del año actual y el gráfico será dinámico ya que permitirá conocer la cifra posicionando el puntero del mouse.

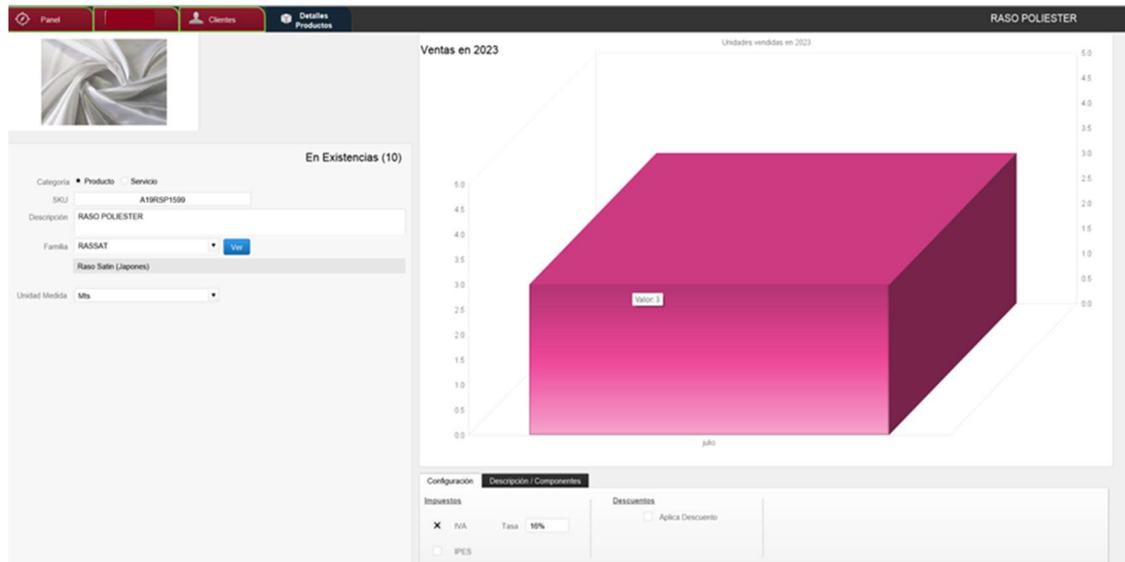


Figura 40 - Gráfica de ventas de productos.

4.3.7 Búsqueda de productos

Pantalla de búsqueda de productos, ordenamiento y existencias en donde el usuario podrá tener distintas opciones, ordenar por nombre, familia o categoría de los productos.

Estos resultados los podrá exportar a Excel y de la misma forma podrá importar productos de manera masiva desde un layout predefinido en Excel.

Panel	Cientes	Productos	Clasificación
A13RPL1599	RAZO 100% POLIESTER	Razo Satin (Japones)	
A19RSP1599	RAZO POLIESTER	Razo Satin (Japones)	10
TULBOR			
A08VRS1499	TUL VARSOVIA 100% POLIESTER	Tul Bordado	
A06PRG1399	TUL PRAGO 100% POLIESTER	Tul Bordado	
A00TLB1399	TUL BRUSELAS 100% POLIESTER 062	Tul Bordado	
A00TCG1399	TUL COPENHAGEN 100% POLIESTER	Tul Bordado	
A06GLM1399	GLAMOUR 100% POLIESTER 726	Tul Bordado	
A06CKT1399	COCKTAIL 100% POLIESTER 022	Tul Bordado	
TULGLL			
A06GP1599	TUL CON GLITER AL TONO 100% POL. LISO	Tul Gliter	
A054Z215499	TT4228D 100%POL. 1.54M ANC. LZO ACABA DIAM PLATA	Tul Gliter	
A05TPB1499	TP5B 100%POL. BT TUL DIAM REY ALM 50M COMERCIAL KA	Tul Gliter	
A10TLL1599	TULL LLLUVA	Tul Gliter	50
A14TGL1599	TUL CON GLITTER	Tul Gliter	

Figura 41 - Búsqueda de productos.

4.3.8 Panel de la empresa

Representará una de las pantallas más importantes del sistema, ya que reflejará un bosquejo general del año de gráficas interactivas con funcionamiento de “Business Intelligence” mostrando las ventas del año en curso, los 5 productos principales y los mejores 5 clientes.

El desarrollo de BI será importante porque deberá mezclar en el lenguaje de programación una potente explotación de SQL a través del panel interactivo.

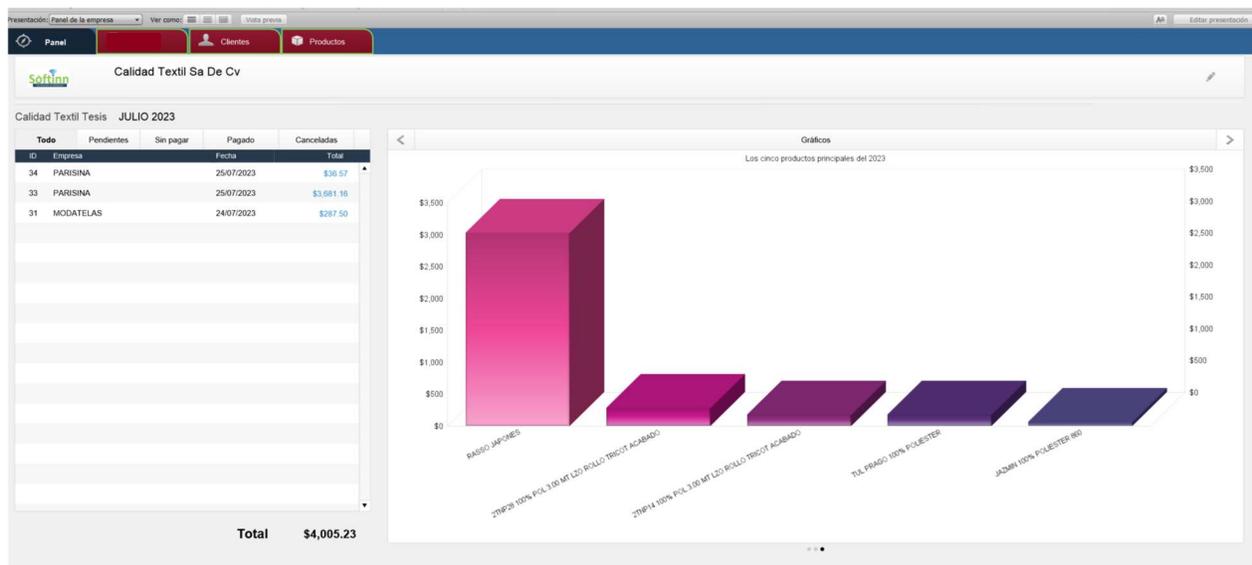


Figura 42 - Panel de la empresa.

4.3.9 Proceso de compras

Transacción que se encargará de registrar las distintas compras realizadas a cada uno de los proveedores, así como el registro del costo de compra, mismo que lo toma del dato maestro del producto.

Ingresando desde el menú principal de la empresa se dirigirá el sistema hacia el módulo de compras, donde se ingresará el código del proveedor, la fecha de compra y alguna observación que se considere pertinente.



Figura 43 - Módulos.

Existirá la opción de captura directa en el grid de detalle para usuarios que conozcan los skus, pero también contará con una pantalla de búsqueda y selección desde un botón de añadir partida.

El sistema irá registrando el costo del dato maestro o permitirá al usuario editarlo en dado caso que el costo en factura del proveedor sea distinto, obteniendo con ello un resumen total de la compra.

Existirá un botón para registrar la compra el cual afectará el inventario de cada uno de los productos:

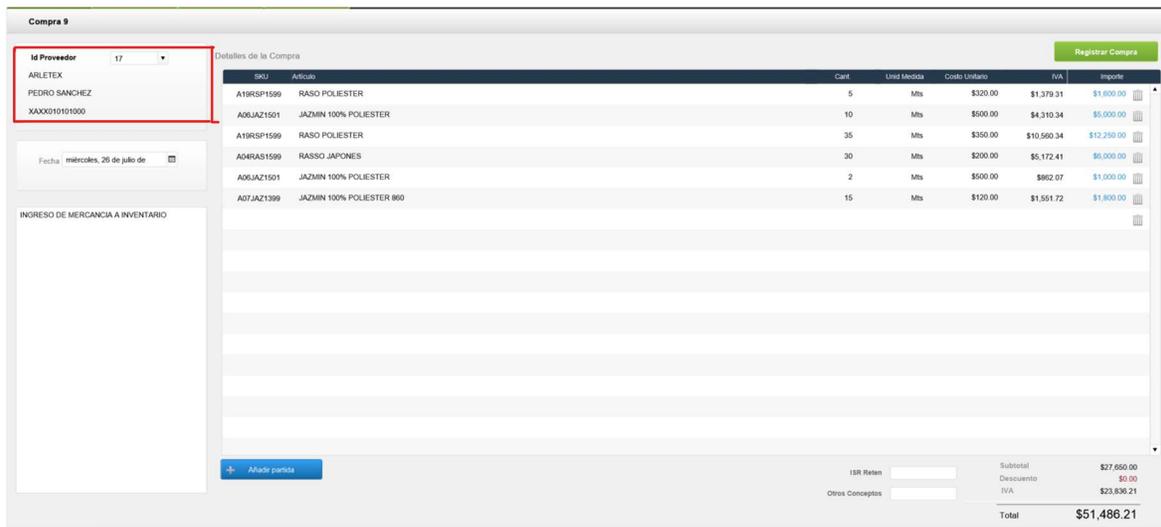


Figura 44 - Registro de compra.

Pantalla de búsqueda y selección de producto:



Figura 45 - Botón añadir partida.

Nuevo producto

• Producto Servicio Buscar Producto...

SKU	Descripción
A19RSP1599	RASO POLIESTER
A14TGL1599	TUL CON GLITTER
A10TSL1599	TELA SATIN
A10TLL1599	TULL LLUVIA
A09RST1599	RAZO SATIN ANCHO 1.50 POLIESTER 100%
A07JAZ1399	JAZMIN 100% POLIESTER 860
A06VRS1499	TUL VARSOVIA 100% POLIESTER
A06TLB1399	TUL BRUSELAS 100% POLIESTER 082
A06GTP1599	TUL CON GLITER AL TONO 100% POL. LISO
A06TCG1399	TUL COPENHAGEN 100% POLIESTER
A06PRG1399	TUL PRAGO 100% POLIESTER
A06JAZ1501	JAZMIN 100% POLIESTER
A06GLM1399	GLAMOUR 100% POLIESTER 726
A06CKT1399	COCKTAIL 100% POLIESTER 022
A05TPB1499	TP5B 100%POL BT TUL DIAM REY ALM 50M COMERCIAL KA
A0542215499	TT422BD 100%POL 1.54M ANC.LZO ACABA DIAM PLATA
A04RAS1599	RASSO JAPONES

Figura 46 - Selección del producto.

4.3.10 Control del inventario

Cada transacción afectará al inventario acorde a la naturaleza de origen de cada una de ellas y podremos verlo reflejado desde el reporte de listado de productos o en la tarjeta de cada uno de ellos.

Mostrando 17 Resultados

SKU	Producto	Familia	Existencias
A06JAZ1501	JAZMIN 100% POLIESTER	Jazmin	11
A07JAZ1399	JAZMIN 100% POLIESTER 860	Jazmin	30
A04RAS1599	RASSO JAPONES	Raso Satin (Japones)	27

Figura 47 - Control de inventario.

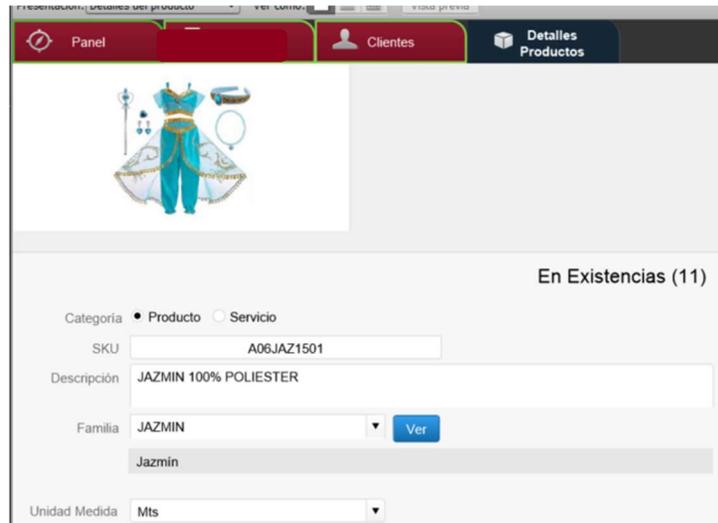


Figura 48 - Detalle inventario.

4.3.11 Reportes

- **Informe de ventas por producto:** Contará con un formato auditable y funciones de exportación en PDF, al anterior reporte desarrollado con clase lista exportable a Excel, esta clase mantiene un ordenamiento predefinido.

SKU	Descripción	Cantidad	Importe
PARISINA			
A04RAS1599	RASSO JAPONES	115	\$3,218.22
A06PRG1399	TUL PRAGO 100% POLIESTER	6	\$177.00
A06TCG1399	TUL COPENHAGEN 100% POLIESTER	2	\$59.00
A06TLB1399	TUL BRUSELAS 100% POLIESTER 082	1	\$43.00
A07JAZ1399	JAZMIN 100% POLIESTER 860	6	\$66.00
A09RST1599	RAZO SATÍN ANCHO 1.50 POLIESTER 100%	1	\$20.00
A10TLL1599	TULL LLUVIA	1	\$36.57
A10TLS1599	TELA SATIN	1	\$36.50
A19RSP1599	RASO POLIESTER	7	\$94.50
SubTotal		140	\$3,750.80

Figura 49 - Informe de ventas por producto.

- **Impresión de Ticket:** Se podrá adaptar a formatos de impresión en impresoras térmicas y cotará con función de envío por correo electrónico al cliente.

Presentación: Imprimir/enviar facturas Ver como: Abandonar Vista previa



CALIDAD TEXTIL SA DE CV
 Venustiano Carranza 138-A
 Col. Centro, Deleg. Cuahutémoc
 CDMX. Sucursal 2
 Tel. 55 22 62 24, 55 22 55 47

REMISIÓN: 39
 19/09/2023

CLIENTE: MODATELAS
RFC: XAXX010101000

VENDEDOR: Upatlan

Pagado

PRODUCTO	DESCRIPCIÓN	CANT.	P. UNIT	DESCUENTO	IMPORTE
A04RAS1599	RASSO JAPONES	3	\$5.00	\$1.29	\$13.50
A06JAZ1501	JAZMIN 100% POLIESTER	1	\$13.80		\$13.80
A06TLB1399	TUL BRUSELAS 100% POLIESTER 082	2	\$49.45		\$98.90
A10TLL1599	TULL LLUVIA	13	\$55.78	\$31.26	\$688.88

Figura 50 - Ticket de venta.

IMPORTE CON LETRA

SUBTOTAL	\$735.21
DESCUENTO	\$32.55
IVA	\$112.43

Figura 51 - Importe de venta.

- **Reportes de ventas mensual:** Se podrá exportar en formato PDF para análisis de negocio acorde al tipo de producto y será una alternativa más para toma de decisiones de la empresa.


Calidad Textil Sa De Cv
 Reporte de Ventas
 2023
 05/10/2023

SKU	Descripción	Cant.	IVA	Total
JULIO				
Venta de Productos Con Iva				
A04RAS1	RASSO JAPONES	125	\$479.58	\$3,476.98
A06PRG1	TUL PRAGO 100% POLIESTER	6	\$24.41	\$177.00
A06TCG1	TUL COPENHAGEN 100% POLIESTER	2	\$8.14	\$59.00
A06TLB13	TUL BRUSELAS 100% POLIESTER 082	1	\$5.93	\$43.00
A07JAZ13	JAZMIN 100% POLIESTER 860	6	\$9.10	\$66.00
A09RST1	RAZO SATÍN ANCHO 1.50 POLIESTER 100%	1	\$2.76	\$20.00
A10TLL15	TULL LLUVIA	1	\$5.04	\$36.57
A10TLL15	TELA SATIN	1	\$5.03	\$36.50
A19RSP1	RASO POLIESTER	3	\$5.59	\$40.50
SubTotal		146	\$545.59	\$3,955.55

Figura 52 - Reporte mensual de ventas.

CONCLUSIONES

El desarrollo de sistemas ha ido perfeccionándose con el paso del tiempo, se han mejorado las técnicas y métodos para su construcción, en la actualidad existe una gran variedad de ellas, pero a criterio de los profesionales se hará uso de la que él considere mejor o la que más se adapte a las necesidades del proyecto.

El software, como muchas otras cosas, surge por la necesidad de resolver problemas o tareas humanas, facilita el trabajo y dota de mejores herramientas a los humanos para hacer más fácil las tareas cotidianas. El sistema que se propone a Calidad Textil ofrece resolver una necesidad que facilitará el trabajo de la empresa, le permitirá mejorar significativamente el ahorro de recursos y le permitirá tomar mejores decisiones a partir del correcto manejo de los datos.

Para poder entrar a la fase de desarrollo de todo sistema es importante definir bien que es lo que queremos que haga el sistema, esta fase de definición es crucial. Se prevé que, al implementar este sistema propuesto a Calidad Textil, les permitirá a sus usuarios dedicar tiempo que antes dedicaban a realizar tareas manuales; en hacer otras tareas como la expansión de su negocio.

Algunos beneficios que se prevé obtendrán Calidad Textil al desarrollar e implementar el sistema propuesto en este documento son los siguientes:

- Permitirá ahorrar tiempo y reducir errores en comparación con los métodos manuales. Permitirá una **gestión eficiente** de los niveles de inventario y reducirá la necesidad de realizar tareas repetitivas y tediosas.
- Al tener una visión clara de los niveles de inventario, se podrá **optimizar el uso de los recursos** y la falta de existencias. Esto ayudará a reducir los costos y aumentar la rentabilidad.
- Los informes proporcionarán información valiosa sobre las tendencias de ventas y el nivel de existencias. Esto permitirá a los propietarios **tomar decisiones informadas** sobre el abastecimiento.
- El sistema garantizará que los productos estén disponibles cuando los clientes los necesiten. Esto **mejorará la satisfacción** y ayudará a mantener una buena reputación empresarial.

Hoy nadie puede ser ajeno al uso de herramientas de software que faciliten sus actividades cotidianas y no se puede entender al mundo sin antes entender que al igual que él; los humanos son parte de un sistema muy complejo.

Durante la universidad, a todos se les dota de herramientas y conceptos que hoy están plasmados en este trabajo y se concluye entonces que el software es un aliado del humano, no una amenaza como a menudo se piensa.

TRABAJOS FUTUROS

- Desarrollar el sistema de gestión de inventarios en la microempresa como se ha planteado en este trabajo.
- Enriquecer el módulo de compras al implementar la funcionalidad de ordenes de compras y cuentas por pagar.
- Implementar un módulo para facturación.
- Desarrollar y ejecutar un plan de mantenimiento del sistema de gestión de inventario una vez que se haya implementado en la empresa.

GLOSARIO

Software: La RAE lo define como un conjunto de programas, instrucciones y reglas para ejecutar ciertas tareas en una computadora u ordenador.

Hardware: La RAE lo define como un conjunto de los componentes que integran la parte material de una computadora u ordenador.

Scrum: De acuerdo con el sitio web Atlassian; Scrum es un marco de gestión de proyectos de metodología ágil que ayuda a los equipos a estructurar y gestionar el trabajo mediante un conjunto de valores, principios y prácticas. Al igual que un equipo de rugby (de donde proviene su nombre) cuando entrena para un gran partido, scrum anima a los equipos a aprender a través de las experiencias, a autoorganizarse mientras aborda un problema y a reflexionar sobre sus victorias y derrotas para mejorar continuamente.

CXC: Abreviatura que se emplea para referirse a las cuentas por cobrar aprovechando las letras iniciales del término y creando un acrónimo con ellas, "Cuentas X Cobrar".

Sistema Operativo: La Real Academia Española lo define como un programa o conjunto de programas que realizan funciones básicas y permiten el desarrollo de otros programas.

Por su parte, la Enciclopedia Universal 2012, lo define como un conjunto de programas o software destinado a permitir la comunicación del usuario con un ordenador y gestionar sus recursos de manera cómoda y eficiente.

Web Services: Un servicio web (o web service) es una vía de intercomunicación e interoperabilidad entre máquinas conectadas en Red y es un software con un formato basado en texto que funciona con Internet. Este sistema se encarga de permitir la transmisión de solicitudes y respuestas entre diferentes servidores o aplicaciones, sin importar las diferencias que existan entre los lenguajes de programación en el que fueron desarrolladas o la plataforma en la que se ejecutan.

Web Services Description Languages: WSDL es una gramática XML que se utiliza para describir la interfaz pública de los servicios Web.

ERP: El término se refiere a Enterprise Resource Planning, que significa "sistema de planificación de recursos empresariales". El software ERP sirve para hacerse cargo de distintas operaciones internas de una empresa, desde producción a distribución o incluso recursos humanos.

WebApps: Es una versión de la página web optimizada y adaptable a cualquier dispositivo móvil. Dicho de otra manera, es una página que se puede abrir desde el navegador de cualquier terminal independientemente del sistema operativo que utilice.

IA: La inteligencia artificial o IA, es la simulación de procesos de inteligencia humana por parte de máquinas, especialmente sistemas informáticos. Estos procesos incluyen el aprendizaje (la adquisición de información y reglas para el uso de la información), el razonamiento (usando las reglas para llegar a conclusiones aproximadas o definitivas) y la autocorrección.

SAAS: El software como servicio (SaaS) es un modelo de software basado en la nube que ofrece aplicaciones a los usuarios finales a través de un navegador de Internet. Los proveedores de SaaS alojan servicios y aplicaciones para que los clientes puedan acceder a ellos bajo demanda.

UML: Significa Lenguaje de Modelado Unificado. Es un lenguaje genérico de modelado de desarrollo utilizado para el análisis, diseño e implementación de sistemas de software. El propósito de UML es proporcionar un método simple y común para visualizar las propiedades arquitectónicas inherentes de un sistema de software.

TICS: Las Tecnologías de la Información y la Comunicación son los recursos y herramientas que se utilizan para el proceso, administración y distribución de la información a través de elementos tecnológicos, como: ordenadores, teléfonos, televisores, etc.

Base de datos: Es una recopilación organizada de información o datos estructurados, que se almacena de forma electrónica en un sistema informático. Una base de datos está controlada por un sistema de gestión de bases de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado normalmente a simplemente base de datos.

Kardex: Es un documento administrativo de control, en donde se lleva el registro de la existencia de la mercancía, materia prima o productos de una empresa. Pero, no solo de su situación como stock sino también de los diferentes cambios que se realizan con ella, por ejemplo, ventas, cambios, cancelaciones e incluso pérdidas por desperfectos.

PEPS: Las siglas PEPS significa “Primero en Entrar, Primero en Salir”, lo cual quiere expresar que los artículos que se agregaron primero al inventario, los más antiguos, son los primeros artículos que se retirarán del inventario para su venta.

Esto significa que aquellos artículos que entraron primero al inventario serán los primeros en salir; de este modo, el inventario se renueva constantemente, así como se mantiene un control de los precios actualizados.

UEPS: Las siglas UEPS significa “Último en Entrar, Primero en Salir”, lo cual quiere indicar que los artículos agregados más recientemente al inventario se consideran como los primeros artículos a retirar del inventario para la venta.

Si los costos están aumentando, los últimos artículos que ingresaron al inventario, que son los más costosos, son los que se venden primero, incrementando el costo de la

mercancía vendida, y reportando así menos ganancias. Por tanto, se paga una menor cantidad de impuesto sobre la renta en el corto plazo.

Backlog: Esta palabra inglesa significa “acumulación de algo, especialmente trabajo incompleto o cosas de las que debemos ocuparnos”.

Dashboard: Es una herramienta de gestión de la información que monitoriza, analiza y muestra de manera visual los indicadores clave de desempeño (KPI), métricas y datos fundamentales para hacer un seguimiento del estado de una empresa, un departamento, una campaña o un proceso específico.

Podemos pensar en el dashboard como una especie de "resumen" que recopila datos de diferentes fuentes en un solo sitio y los presenta de manera digerible para que lo más importante salte a la vista.

SKU: Un SKU es un conjunto de números y letras, empleado para identificar, localizar y hacer seguimiento interno de un producto en una empresa o tienda. De ahí el origen del término inglés, Stock Keeping Unit (Unidad de Mantenimiento de Stock), que en español usamos como referencia de almacén.

BIBLIOGRAFÍA

- (s.f.). Obtenido de <https://matriarm.wordpress.com/desarrollo-basado-en-componentes/>
- Barber, F., & Ferrís, R. (s.f.). Obtenido de http://informatica.uv.es/iiguia/AED/oldwww/2004_05/AED.Tema.02.pdf
- BARTOLINI, S. (s.f.). Obtenido de <http://www.gvb-csic.es/software-y-database/index.html>
- Bertalanffy, L. V. (1976). Teoría General de los Sistemas. En L. V. Bertalanffy, *Teoría General de los Sistemas* (pág. 336). México: Fondo de Cultura Económica.
- C. Laudon, K., & P. Laudon, J. (2012). *Sistemas de Información Gerencial*. México: Ed. Pearson.
- Cevallos, K. (29 de Abril de 2015). Obtenido de <https://ingsoftwarekarlacevallos.wordpress.com/2015/04/29/modelos-de-procesos-especializado/>
- Clavel Maqueda, M., & Cornejo Velázquez, E. (s.f.). Obtenido de <https://www.uaeh.edu.mx/scige/boletin/icea/n5/e9.html#:~:text=Un%20Sistema%20de%20Informaci%C3%B3n%20es,distribuye%20en%20la%20organizaci%C3%B3n%20para>
- Cutrell, E. (s.f.). *Naciones Unidas*. Obtenido de <https://www.un.org/es/chronicle/article/el-contexto-y-el-diseno-de-tic-para-el-desarrollo-mundial>
- Del Carmen Ramírez, L., & Flórez Fuentes, A. S. (Diciembre de 2014). BUENAS PRÁCTICAS, UNA SOLUCIÓN PARA UN MEJOR DESARROLLO DE SOFTWARE. pág. 9.
- Distancia-UNAM, C. d. (14 de 07 de 2023). Obtenido de https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1100/mod_resource/content/17/uapa_teor%C3%ADa_conceptos_u4_/index.html
- Domínguez Ríos, V. A. (23 de 01 de 2017). *Teoría General de Sistemas, un enfoque práctico: General Systems Theory, a practical approach*. Obtenido de <https://doi.org/10.54167/tch.v10i3.174>
- Durand, S. W. (2017). *Análisis y requerimientos de software: manual autoformativo*. Huancayo: Universidad Continental.
- Ferreya, F. (s.f.). *Proyectosagiles.org*. Obtenido de <https://proyectosagiles.org/planificacion-iteracion-sprint-planning/>
- FileMaker, C. (28 de 11 de 2023). *Claris*. Obtenido de Claris: <https://www.claris.com/es/resources/>
- Gaviglio, S. (2023). *Infonegocios*. Obtenido de <https://infonegocios.info/infotecnologia/red-link-mejora-la-experiencia-de-usuario-en-sus-cajeros-automaticos>
- Gómez Fuentes, M., Cervantes Ojeda, J., & González Pérez, P. P. (2019). *Fundamentos de Ingeniería de Software*. México: UNIVERSIDAD AUTONOMA METROPOLITANA.
- Huerta, A. E. (26 de 03 de 2023). *SlideShare*. Obtenido de <https://es.slideshare.net/epifania/importancia-de-la-teora-general-de-sistemaspptx>

- Integratit. (s.f.). *Integratit*. Obtenido de <https://integratit.com.mx/blog/sprint-y-sprint-backlog/>
- Karlsson, J. (s.f.). *PERFORCE*. Obtenido de <https://www.perforce.com/resources/hns/agile-product-backlog-basics>
- Londoño, P. (20 de 07 de 2023). *HubSpot*. Obtenido de <https://blog.hubspot.es/website/que-es-aplicacion-web>
- López, D. Y. (07 de 2017). Obtenido de https://www.uaeh.edu.mx/docencia/P_Conceptual/sahagun/industrial/2017/Clasificacion_de_lo_s_sistemas.pdf
- Méndez, P. (s.f.). *Studocu*. Obtenido de <https://www.studocu.com/latam/document/universidad-tecnologica-centroamericana/sociologia/teoria-general-de-sistemas-modelo-de-kant-y-kanz/14919010>
- México, V. (13 de 07 de 2023). *Visual México*. Obtenido de <https://visualmexico.com.mx/software-de-inventarios/#:~:text=Durante%20la%20Segunda%20Guerra%20Mundial,sistemas%20de%20planificaci%C3%B3n%20de%20inventario.>
- Navarro, M. J. (1999). *INVESTIGACIÓN DE OPERACIONES Control de Inventarios y Teoría de Colas*. San José, Costa Rica: Universidad Estatal a Distancia.
- Osorio, D. E. (21 de 07 de 2009). *Gestiópolis*. Obtenido de <https://www.gestiopolis.com/teoria-general-de-sistemas-ludwig-von-bertalanffy/>
- Pressman, R. S. (2010). *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO*. México: McGRAW-HILL.
- Raeburn, A. (25 de Agosto de 2022). *asana*. Obtenido de https://asana.com/es/resources/product-backlog?psafe_param=1&gad_source=1&gclid=Cj0KCQiAr8eqBhD3ARIsAie-buPuAlk0-228rNsYYaiNcsDETDIBoMRmESacqRgMTaaqXu7fATaZh0QaAt25EALw_wcB&gclsrc=aw.ds
- Ranchal, J. (2014 de 07 de 2014). *MuyComputer*. Obtenido de <https://www.muycomputer.com/2014/07/30/comandos-cmd/>
- Reuters. (22 de 03 de 2023). *Reuters*. Obtenido de <https://www.reuters.com/technology/chatgpt-owner-openai-fixes-significant-issue-exposing-user-chat-titles-2023-03-22/>
- Salamanca, U. d. (s.f.). *Grupo de investigación BISITE*. Obtenido de <https://bisite.usal.es/es/blog/formacion/21/05/06/importancia-de-las-tic-en-la-actualidad-BISITE>
- Schwaber, K., & Sutherland, J. (2020). *La Guía de Scrum*.
- Services, A. (04 de 10 de 2018). *ACL Analytics Help*. Obtenido de https://help.highbond.com/helpdocs/analytics/13/user-guide/en-us/Content/acl_analytics_help_new.htm
- Sola, G. H. (19 de Noviembre de 2022). *Scrum.org*. Obtenido de <https://www.scrum.org/resources/blog/un-product-owner-no-es-un-project-manager>

- Solutions, P. (s.f.). *Pyramid Solutions*. Obtenido de <https://pyramidsolutions.com/network-connectivity/blog-nc/what-is-embedded-software/>
- Sommerville, I. (2011). *Ingeniería de Software*. México: PEARSON EDUCACIÓN.
- UNAM. (s.f.). *Unidad de Apoyo para el Aprendizaje*. Obtenido de https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/index.html#introduccion
- UNIR - Universidad Internacional de La Rioja. (06 de 04 de 2021). *UNIR - Universidad Internacional de La Rioja*. Obtenido de <https://mexico.unir.net/ingenieria/noticias/ingenieria-de-software-que-es-objetivos/>
- Universidades, S. (21 de 12 de 2020). *Santander*. Obtenido de <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>
- Vázquez, K. (10 de 10 de 2022). *eSemanal*. Obtenido de <https://esemanal.mx/2022/10/evolucion-y-futuro-del-software-en-mexico-y-las-oportunidades-del-canal/>
- Villegas, M. (06 de 10 de 2014). *Informática TUAJ*. Obtenido de <https://informaticatuai.wordpress.com/2014/10/06/procesadores-de-textos-definicion-funcion-y-tipos-2/>
- West, D. (s.f.). *ATLASSIAN*. Obtenido de <https://www.atlassian.com/es/agile/scrum/sprint-planning>

**ANEXO 1: ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE IEEE
830.**

Especificación de requisitos de software

Proyecto: Sistema de gestión de inventario para la microempresa Calidad Textil.
Revisión 1.1

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. calidad.
13/02/2024	1.1	PLCC. Kelvin Olgún Beltrán	

Documento validado por las partes en fecha: 15/02/2024

Por el cliente	Por la empresa suministradora
Calidad Textil José Ulises Patlán Cadena	

Contenido

FICHA DEL DOCUMENTO	2
CONTENIDO	3
1 INTRODUCCIÓN	5
1.1 Propósito	5
1.2 Alcance	5
1.3 Personal involucrado	5
1.4 Definiciones, acrónimos y abreviaturas	6
1.5 Referencias	6
1.6 Resumen	6
2 DESCRIPCIÓN GENERAL	7
2.1 Perspectiva del producto	7
2.2 Funcionalidad del producto	7
2.3 Características de los usuarios	7
2.4 Restricciones	8
2.5 Suposiciones y dependencias	8
2.6 Evolución previsible del sistema	8
3 REQUISITOS ESPECÍFICOS	8
3.1 Requisitos comunes de los interfaces	11
3.1.1 Interfaces de usuario	11
3.1.2 Interfaces de hardware	11
3.1.3 Interfaces de software	11
3.1.4 Interfaces de comunicación	11
3.2 Requisitos funcionales	11
3.2.1 Requisito funcional 1	11
3.2.2 Requisito funcional 2	11
3.2.3 Requisito funcional 3	12
3.2.4 Requisito funcional n	12
3.3 Requisitos no funcionales	12
3.3.1 Requisitos de rendimiento	12
3.3.2 Seguridad	12
3.3.3 Fiabilidad	13
3.3.4 Disponibilidad	13
3.3.5 Mantenibilidad	13

Especificación de Requisitos de Software

3.3.6	Portabilidad	13
3.4	Otros requisitos	13
4	APÉNDICES	13

1 Introducción

El documento contiene las especificaciones de los requisitos del software para el sistema de gestión de inventario para la microempresa Calidad Textil. Los requerimientos solicitados por el cliente engloban varios procesos como el de compras, productos, clientes, etc.

La estructura de este documento está hecha con base al formato IEEE 830.

1.1 Propósito

El objetivo de este documento ERS es definir de manera específica los alcances y restricciones que tendrá el sistema de gestión de inventario para la microempresa Calidad Textil.

- Se desea realizar enfocando su uso y dirección al equipo de desarrollo del software.
- Este será una guía para el desarrollo de cada uno de los módulos que integrarán el sistema informático.

1.2 Alcance

El sistema de gestión de inventario para la microempresa Calidad Textil tendrá las siguientes funciones:

- Administración de catálogos maestros para administración de información base (proveedores, productos, clientes etc.)
- Contará con módulo de compras, el cual permitirá la captura de la adquisición de mercancías solicitada al proveedor y surtimiento al almacén.
- Administrará las existencias de los productos en sus respectivos almacenes, permitiendo el registro de transacciones operativas para flujos de inventarios.
- Relacionarán todos los datos para obtención de informes de movimientos, estadístico y evaluación del almacén.
- El sistema se regirá mediante la seguridad de información con roles por usuarios.

1.3 Personal involucrado

Nombre	Kelvin Olgún Beltrán
Rol	Product Owner
Categoría profesional	PLCC
Responsabilidades	Comunicación de las necesidades a cubrir por el software, indagar los flujos operativos a realizar en el sistema.
Información de contacto	Kelvin270470@gmail.com

Nombre	Patlán Cadena José Ulises
Rol	Scrum Máster
Categoría profesional	PLCC
Responsabilidades	Seguimiento al proyecto en conjunto con el producto owner,

Especificación de Requisitos de Software

	eliminación de obstáculos en la ejecución del proyecto, gestionar el orden de ejecución de las actividades.
Información de contacto	pacaulises@gmail.com

Nombre	Carlos Chávez Bautista
Rol	Team Member
Categoría profesional	PLCC
Responsabilidades	Ejecución de diagramas UML, ejecución de los requerimientos, calidad y liberación del software.
Información de contacto	cachaba@gmail.com

Nombre	Dolly María Colín López
Rol	Team Member
Categoría profesional	PLCC
Responsabilidades	Ejecución de diagramas UML, ejecución de los requerimientos, calidad y liberación del software.
Información de contacto	domacolopez@gmail.com

Nombre	Erick Ricardo Martínez Hernández
Rol	Team Member
Categoría profesional	LCC
Responsabilidades	Ejecución de diagramas UML, ejecución de los requerimientos, calidad y liberación del software.
Información de contacto	erickrmh@gmail.com

1.4 Definiciones, acrónimos y abreviaturas

Nombre	Descripción
Usuario	Persona que usará el sistema para gestionar procesos
ERS	Especificación de Requisitos Software
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
Dead Line	Plazo para desarrollo de software

1.5 Referencias

Referencia	Título	Fecha	Autor
IEEE	Standard IEEE 830	1998	IEEE
ISOFT	Ingeniería del Software 7ma edición	2010	Roger S. Pressman

1.6 Resumen

Este documento consta de tres secciones:

- **Primera sección:** Se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.
- **Segunda sección:** Se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, los

datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.

- **Tercera sección:** Aquella en la que se definen detalladamente los requisitos que debe satisfacer el sistema.

2 Descripción general

2.1 Perspectiva del producto

El sistema de gestión de inventarios funcionará con una arquitectura en monolito, este término se utiliza para describir sistemas donde la aplicación y la base de datos están combinadas o integradas en una única entidad, por tanto, es un enfoque arquitectónico en el desarrollo de software en el que todos los componentes y funcionalidades de una aplicación se encuentran dentro de un solo código base.

En este tipo de sistemas, la aplicación y la base de datos suelen estar acopladas y desplegadas juntas en un mismo entorno.

2.2 Funcionalidad del producto

- El sistema tendrá funciones tales como administración de catálogos maestros para administración de información base (proveedores, productos, clientes etc.)
- Contará con módulo de compras, el cual permitirá la captura de la adquisición de mercancías solicitada al proveedor y surtimiento al almacén.
- Administrará las existencias de los productos en sus respectivos almacenes, permitiendo el registro de transacciones operativas para flujos de inventarios.
- Relacionarán todos los datos para obtención de informes de movimientos, estadístico y evaluación del almacén.
- El sistema se registrará mediante la seguridad de información con roles por usuarios.

2.3 Características de los usuarios

Tipo de usuario	Administrador del Sistema
Formación	Formación en Ciencias de la computación, sistemas, informática o afín
Habilidades	Administración y Organización
Actividades	Control y manejo de sistema en general

Tipo de usuario	Administrador de Datos Maestros
Formación	Formación en Ciencias de la computación, sistemas, informática o afín

Especificación de Requisitos de Software

Habilidades	Administración y Organización
Actividades	Responsable de la actualización de catálogos, datos maestros en el aplicativo garantizando que la información sea confiable

Tipo de usuario	Analista y Administrador de Negocio
Formación	Administrador de Empresas o carrera afín
Habilidades	Administración y Organización
Actividades	Análisis de reportes de información y tomas de decisión

2.4 Restricciones

Requerimientos mínimos recomendados para la fase de desarrollo:

- 8 GB de memoria RAM.
- Procesador 3.5 GHz.
- 4 GB de almacenamiento libre.

2.5 Suposiciones y dependencias

- Se asume que los requisitos aquí descritos son estables.
- Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados para garantizar una ejecución correcta.

2.6 Evolución previsible del sistema

- Enriquecer el módulo de compras al implementar la funcionalidad de ordenes de compras y cuentas por pagar.
- Implementar un módulo para facturación.

3 Requisitos específicos

Requisito Funcional 1.

Número de requisito	RF01
Nombre de requisito	Autenticación de Usuarios
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	El ingreso al sistema será mediante una pantalla de login donde se requiere el nombre del usuario y una contraseña. La pantalla de conexión deberá contar con la cadena de conexión a la instancia indicando el nombre de la base de datos a conectar; los datos de la cadena de conexión serán implícitos y no dependientes de captura por el usuario.

Especificación de Requisitos de Software

Requisito Funcional 2.

Número de requisito	RF02
Nombre de requisito	Registrar Usuarios
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	El sistema permitirá al usuario (Administrador) registrarse y este deberá proporcionar datos como: Usuario y contraseña.

Requisito Funcional 3.

Número de requisito	RF03
Nombre de requisito	Datos Maestros
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	La estructura de la DB contará con tablas en la DB de tipo catálogo, las cuales tendrán la capacidad de albergar datos maestros, mismos que serán ocupados como medios de captura por cada transacción registrada.

Requisito Funcional 4.

Número de requisito	RF04
Nombre de requisito	Operación
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	Todas las operaciones del día a día serán registradas en tablas del sistema denominadas transaccionales mediante el uso de transacciones las cuales contarán con validaciones y ejecuciones por cada transición que tenga acceso el usuario

Requisito Funcional 5.

Número de requisito	RF05
Nombre de requisito	Consultar Información
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	El sistema ofrecerá al usuario información general acerca de las ventas.

Requisito Funcional 6.

Número de requisito	RF06
Nombre de requisito	Gestionar Reportes
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	Permite al administrador imprimir reportes de las operaciones de inventarios, mediante parámetros de búsqueda, realizadas con detalles y totales.

Especificación de Requisitos de Software

Requisito Funcional 7.

Número de requisito	RF07
Nombre de requisito	Roles de Seguridad
Tipo	Restricción
Prioridad del requisito	Alta/Esencial
Descripción	El sistema contará con un repositorio de usuarios, a cada uno de ellos le será asignado un rol dentro del sistema, el cual será los objetos a los cuales tendrá acceso y las transacciones que podrá ejecutar.

Requisito Funcional 8.

Número de requisito	RF08
Nombre de requisito	Transacción de Ordenes de Compras
Tipo	Requisito
Prioridad del requisito	Media/Deseado
Descripción	En el funcionamiento transaccional, el usuario tendrá la opción de capturar órdenes de compra a proveedor, las condiciones comerciales deberán tomarse previamente del catálogo de proveedores correspondientes al módulo de Datos Maestros.

Requisito Funcional 9.

Número de requisito	RF09
Nombre de requisito	Transacción de Compras
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	Posterior al flujo de la orden de compra, el responsable del área de almacén podrá realizar la recepción total o parcial del contenido de la orden, afectando con ello el inventario en piso y obteniendo como resultado el costo promedio.

Requisito Funcional 10.

Número de requisito	RF10
Nombre de requisito	Transacción de Inventarios
Tipo	Requisito
Prioridad del requisito	Alta/Esencial
Descripción	Una vez ingresado el producto, el sistema contará con transacciones para manejo de existencias bajo los cuales el usuario podrá capturar si es una entrada o salida, clasificando el motivo que la origina.

3.1 Requisitos comunes de los interfaces

La interfaz con el usuario consistirá en un conjunto de ventanas con botones, listas y campos de textos. Ésta deberá ser construida específicamente para el sistema propuesto.

3.1.1 Interfaces de usuario

Deberá ser una interfaz amigable e intuitiva con colores claros y botones que guíen el funcionamiento del mismo.

3.1.2 Interfaces de hardware

Para un óptimo desempeño del sistema será importante que el equipo de cómputo donde se instalará el software cuente con las siguientes características:

- Teclado, mouse e impresora
- Velocidad mínima del CPU 2.6 GHz
- Memoria RAM de 8 Gb
- Pantalla con resolución mínima 1920 * 1080 pixeles vertical a color
- Capacidad WiFi / Fast ethernet
- Disco duro mayor a 100 GB

3.1.3 Interfaces de software

Es requerido que el equipo cuente con:

- Arquitectura del sistema a 64 bits
- Sistema operativo mínimo Windows 10 o superior
- Componentes Net.Framework 4.5 o superior

3.1.4 Interfaces de comunicación

No aplica.

3.2 Requisitos funcionales

3.2.1 Requisito funcional 1 Autenticación de usuarios:

Deberán identificarse para poder acceder al sistema, este podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad.

3.2.2 Requisito funcional 2 Registrar usuarios: El sistema permitirá al usuario (administrador) registrarse. El

usuario debe proporcionar datos como: Usuario y contraseña.

3.2.3 Requisito funcional 3 Datos maestros: El usuario deberá dar de alta el catálogo de proveedores junto a sus condiciones comerciales, catálogo de productos y su clasificación. El usuario contará con permisos a los catálogos de crear, modificar y eliminar.

3.2.4 Requisito funcional 4 Operación: Todas las operaciones del día a día serán registradas en tablas del sistema denominadas transaccionales mediante el uso de transacciones las cuales contarán con validaciones y ejecuciones por cada transición que tenga acceso el usuario.

3.2.5 Requisito funcional 5 Consultar información: El sistema ofrecerá al usuario información general acerca de las ventas.

- Consultar informes estadísticos: Muestra información general sobre las entradas y salidas de mercancías.
- Consultar operaciones: Muestra información a detalle por cada movimiento operativo de inventarios registrado en el sistema.
- Consultar proveedores: Muestra datos de cada proveedor junto a sus condiciones comerciales como precios y descuentos.
- Consultar existencias: Muestra en tiempo real el inventario disponible para ser ocupado en cada movimiento.

3.2.6 Requisito funcional Gestionar Reportes: Permite al administrador imprimir reportes de las operaciones de inventarios, mediante parámetros de búsqueda, realizadas con detalles y totales.

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

- Asegurar que al realizar un proceso o consulta no se vea afectado el desempeño de la base de datos ni el tráfico de la red.

3.3.2 Seguridad

- Asegurar a los distintos usuarios un buen desempeño, la seguridad y confiabilidad del sistema. Es decir, la información y registros almacenados, podrán ser actualizados y consultados de manera simultánea sin que esto afecte el tiempo de respuesta.

Especificación de Requisitos de Software

- Deberá dar garantía de que toda la información y datos que se manejan son seguros y están bien resguardados de agentes externos.
- Deberá facilitar y controlar el acceso a la información al personal autorizado.

3.3.3 Fiabilidad

- El sistema deberá tener una interfaz de uso intuitiva y sencilla.
- La interfaz de usuario deberá ajustarse a las características institucionales, dentro de la cual estará incorporado el sistema.

3.3.4 Disponibilidad

- La disponibilidad del sistema deberá ser continua con un nivel de servicio para los usuarios 24/7, garantizando un esquema adecuado que permita la posible falla en cualquiera de sus componentes, contar con un plan de contingencia.

3.3.5 Mantenibilidad

- El sistema deberá disponer de documentación fácilmente actualizable que permita realizar operaciones de mantenimiento.

3.3.6 Portabilidad

- El sistema deberá ser implantado bajo la plataforma de Windows y Mac, considerando las especificaciones técnicas de hardware y software.

3.4 Otros requisitos

- Sin requisitos adicionales identificados.

4 Apéndices

- Sin información adicional.