



ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN.

**“DISEÑO Y DESARROLLO
DE UN SISTEMA WMS (Warehouse Management System)
PARA LA EMPRESA LOGISTECSA BAJO LA METODOLOGÍA MSF”**

Previo a la obtención del Título:

INGENIERO EN SISTEMAS E INFORMÁTICA

Por: ANDRÉS SALVADOR MARCHANT CASTELNUOVO

Sangolquí 2 de Abril del 2012

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por Sr. ANDRÉS SALVADOR MARCHANT CASTELNUOVO, como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

Sangolquí 2 de Abril del 2012

Sr. Ing Mauricio Campaña

DEDICATORIA

Dedico esta Tesis a mi mamá, que no solo me apoyó durante este proceso tan difícil y largo, si no durante toda mi vida.

Andrés Marchant Castelnuovo

AGRADECIMIENTOS

Agradezco al Sr Ing Alexander Verdesoto, por abrirme las puertas de su empresa y apoyarme a realizar este trabajo de una manera amistosa y a la vez profesional.

También quiero agradecer a mi director de tesis, el Sr Ing. Mauricio Campaña quien me guió durante todo el proceso, exigiendo lo mejor de mí para concluir mis objetivos.

No puedo dejar de mencionar y agradecer a mi co-director, el Ing. Germán Ñacato, quien aportó su experiencia a mi trabajo, brindándole así excelencia académica.

Andrés Marchant Castelnovo

ÍNDICE DE CONTENIDOS

	Pág.
RESUMEN	1
CAPÍTULO I - INTRODUCCIÓN	3
1.1 Introducción	3
1.2 Planteamiento del Problema	6
1.3 Justificación	8
1.4 Objetivos	9
1.5 Alcance	9
CAPÍTULO II – FUNDAMENTACIÓN TEÓRICA	13
2.1 Warehouse Management System	13
2.1.1. Definición de un WMS	13
2.1.2. Historia de los WMS	14
2.1.3. Los WMS en la Industria	15
2.2. Metodología Microsoft Solution Framework	19
2.2.1. Definición de MSF	19
2.2.2. Prevención	21
2.2.2.1. Identificación de los objetivos del proyecto	22
2.2.2.2. Identificar los Objetivos del diseño	25
2.2.3. Planificación	26
2.2.4. Desarrollo	27
2.3. Herramientas de Diseño y Desarrollo	37
2.3.1. Microsoft SQL Management Studio 2005.	37

2.3.2.	Microsoft Visio.	43
2.3.3.	Visual Studio C#.	45
2.3.4.	Microsoft Project.	49
3.	CAPÍTULO III- VISIÓN Y PLANIFICACIÓN	50
3.1.	Visión y Alcance	50
3.1.1.	Beneficios, metas, objetivos y restricciones	52
3.1.2.	Casos de Usos	60
3.1.3.	Requerimientos	85
3.1.3.1.	Requerimientos de estándares de Sistema	86
3.1.3.2.	Requerimientos de Negocio	88
3.1.3.3.	Requerimientos de Interoperabilidad	96
3.1.3.4.	Requerimientos de Rendimiento	108
3.1.3.5.	Requerimientos del Sistema	113
3.2.	Planificación	117
3.2.1.	Vista Conceptual	117
3.2.2.	Vista Lógica.	119
3.2.2.1.	Diagrama de Clases	119
3.2.2.2.	Diagramas de casos de uso	120
3.2.3.	Plan de Riesgos	122

3.2.4. Plan de Pruebas	124
CAPÍTULO IV – DESARROLLO Y ESTABILIZACIÓN DEL APLICATIVO	128
4.1 Desarrollo	128
4.1.1 Desarrollo Estructural de la Plataforma WMS	128
4.1.2 Desarrollo de las Interfaces de la Plataforma WMS	129
4.2 Pruebas	138
4.2.1 Prueba de la integridad de los datos	138
4.2.2 Prueba de Funcionalidad	142
4.2.3 Reportes de Pruebas y Bugs	145
4.2.3.1.Pruebas en Core	145
4.2.3.2.Pruebas de Memoria	145
4.3 Código fuente de la Plataforma WMS	146
CAPÍTULO V – CONCLUSIONES Y RECOMENDACIONES	190
5.1.Conclusiones	190
5.2.Recomendaciones	192
REFERENCIAS BIBLIOGRÁFICAS	194
APÉNDICE 1: Biografía	196
LISTADO DE TABLAS	viii
LISTADO DE FIGURAS	ix

LISTADO DE TABLAS

	Pág
TABLA N°2.1.: Diseño Multi Capa	30
TABLA N° 3.2.: Prioridades	97
TABLA N°3.3.: Descripción de Servicios	111
TABLA N°3.4: Requerimientos de Hardware	115
TABLA N°3.5: Requerimientos de Software	116
TABLA N°3.6.: Tabla de Riesgos	123
TABLA N°3.7.: Tabla de Pruebas	124
TABLA N°3.8.: Prueba de Funcionalidad	125
TABLA N°3.9.: Prueba de Interfaz de Usuario	127

LISTADO DE FIGURAS

	Pág.
FIGURA N° 1.1.: Proceso de Diseño MSF	20
FIGURA N° 3. 1.: Escalado en vertical	90
FIGURA N°3.2.: Escalado en horizontal	91
FIGURA N°3.3.: Red de la empresa Logistecsa	98
FIGURA N°3.4.: Diagrama Conceptual	117
FIGURA N°3.5.: Diagrama de Clases	119
FIGURA N°3.6.: Diagrama simplificado de Uso	120
FIGURA N°3.7.: Diagrama de Casos de Uso	121
FIGURA N°4.1.: Diagrama Estructural	128
FIGURA N°4.2.: Prototipo de Pantalla Validación de usuarios	129
FIGURA N°4.3.: Prototipo de Pantalla Ingreso a la plataforma	130
FIGURA N°4.4.: Prototipo de Pantalla Consulta de protocolos	131
FIGURA N°4.5.: Prototipo de Pantalla Búsqueda de productos por posición	132
FIGURA N°4.6.: Prototipo de Pantalla Crear producto	133
FIGURA N°4.7.: Prototipo de Pantalla Modificar producto	134
FIGURA N°4.8.: Prototipo de Pantalla Eliminar producto	135
FIGURA N°4.9.: Prototipo de Pantalla Órdenes	136
FIGURA N°4.10.: Prototipo de Pantalla Cargar Órdenes	137

RESUMEN

Hoy en día, las necesidades de los operadores logísticos aumentan a pasos agigantados, requiriendo un mayor control sobre el espacio y eficiencia en su administración para poder competir en el mercado actual. Implementar un sistema de administración de bodegas (WMS) ayuda eficazmente a conseguir estos objetivos. Por tales necesidades, la empresa Logistecs S.A contrato al Sr Andrés Marchant Castelnuovo para desarrollar el sistema dentro de sus bodegas.

Con el apoyo del Área de Sistemas se establecieron los puntos básicos para empezar el proyecto, también se analizó la metodología a utilizar y se optó por el MSF (Microsoft Solution Framework) ya que ésta presentaba las mejores características para el entorno de la empresa.

Durante el trascurso del proyecto se desarrolló completamente en C#, debido a que la empresa se basa en tecnología Microsoft y el lenguaje es lo suficiente mente dinámico y eficiente para el propósito perseguido.

Se comenzó el proyecto con el establecimiento de los verdaderos requerimientos de la empresa, siguiendo al pie de la letra los protocolos marcados por la metodología, donde se encuentran con varios problemas de procesos internos, los cuales tuvieron que ser reestructurados para que estos cumplan con los requerimientos básicos de eficiencia y eficacia.

Al tener todas las necesidades cubiertas y los pasos de análisis requeridos por el MSF se creó el esqueleto del sistema con la arquitectura multi capa, el cual el sería la base para el WMS a desarrollar.

El aporte de esta tesis es investigativo y de desarrollo. La primera porque se recaudó información sobre los sistemas WMS actuales y los requerimientos internos de la empresa Logisteca. S.A sobre sus procesos internos para la automatización en el sistema desarrollado. La segunda porque se logró concentrar la mayoría de requerimientos dentro de un sistema integral, el cual proveerá información en tiempo real y oportunamente dentro de las bodegas de la empresa.

CAPÍTULO I

INTRODUCCIÓN

1.1 Introducción

El servicio de logística, a nivel mundial, es un concepto que poco a poco fue creciendo y tomando mayor importancia. Para la región Latinoamericana y en Ecuador, la industria espera un significativo crecimiento en un futuro cercano, por lo cual las inversiones en este segmento de mercado son bastantes redituables.

La principal meta al empezar estas operaciones en las empresas de logística es el diverso rango de servicios como son la planificación, almacenaje y transporte. Los servicios deben ser ofrecidos a nivel regional y enfocados en los mercados nacionales. Adicionalmente a este núcleo de servicios, la compañía provee servicios de acondicionamiento y re-embalaje para aumentar ganancias.

La Industria de logística muestra un fuerte crecimiento en el mercado por un aumento de demanda y el crecimiento en los costos operativos. El desarrollo de nuevas estrategias y soluciones de negocios parecen críticas para los nuevos negocios al obtener partes del mercado y sobrevivir en esta industria tan competitiva. La elección de servicios así como el desarrollo de aplicaciones puede ser una estrategia en este campo de negocios. Adicionalmente un buen manejo de costos es críticamente importante para una sólida corriente de ingresos.

Las operaciones de las empresas de logística ofrecen un núcleo de servicios como los siguientes: servicio de transporte logístico, servicio de planificación, servicio de planificación de distribución, servicio de almacenaje, servicio de acondicionamiento, servicio de re-empaque y entregas *just-in-time*.

Un punto importante en este negocio es el desarrollo de nuevas e innovadoras tecnologías para entregar al cliente una diferencia significativa en el servicio. Como un agregado se puede ofrecer un servicio personalizado basándonos en los sistemas informáticos que facilitan estas tareas reduciendo costos y aumentando la productividad.

El rango de productos es seleccionado para proveer crecimientos potenciales. La operación de este negocio requiere un buen conocimiento del mercado así como conceptos de competencia logística, para aumentar la satisfacción del cliente. Sin embargo, es crítico que este concepto de servicios sea ofrecido acompañado de un fuerte enfoque en el costo de administración.

Una meta central de la estrategia de negocio es desarrollar una única identidad corporativa. Esta identidad creará una lealtad del cliente y ayudará a ganar una ventaja competitiva. Por esta razón está planeado así, adicionalmente de la selección de nuevos e interesantes servicios. La inversión requerida para el propósito de este negocio es moderado comparado a otras compañías en la industria.

Con la finalidad de mejorar los procesos de ingreso de productos, su administración dentro de las bodegas y su posterior entrega, la empresa Logistecsa identificó su necesidad de implementar un nuevo sistema de control y administración logística.

La evolución de un sistema de administración de bodegas es similar a otras soluciones informáticas. Inicialmente es un software que controla el movimiento y el almacenamiento de materiales dentro de las bodegas; el principal papel del sistema *Warehouse Management System* (WMS) es incluir la manufactura, administración de transporte, manejo de órdenes y despachos, además de un control completo del sistema de conteo.

El principal propósito de un sistema WMS es controlar el movimiento y almacenamiento de materiales, con un proceso de operación asociado con las transacciones. Un piking directo, un reemplazo de mercadería directo y un putaway directo son claves para un WMS.

La detallada instalación y procesamiento de un WMS puede variar significativamente de una empresa a otra, sin embargo la lógica básica siempre será la misma: la diversidad de ítems, ubicación, cantidad, unidad de medida y la información de la orden para determinar el stock. Dónde recoger y qué secuencia seguir en estas operaciones son generalmente específicas para cada necesidad.

Dado que Logistecsa es una empresa dedicada a la logística de diferentes empresas tiene muy en cuenta la importancia de mantener una buena administración, dentro y fuera de sus bodegas, para poder brindar el mejor y más eficiente servicio posible.

En la práctica, una solución WMS exitosa es generalmente diseñada para funcionar en Hardware, Software y equipos periféricos, con prácticas operativas mejoradas para el manejo de inventario, espacio y equipo, tanto en las bodegas como en los centros de distribución.

La implementación de un sistema WMS permitirá a la compañía incrementar su competitividad reduciendo costos, mejorando servicio al cliente, aumentando la exactitud en el inventario y mejorando así la flexibilidad y respuesta. Además, este sistema mejorará el manejo del inventario en tiempo real, con información actual así como con las órdenes más recientes, envíos y cualquier movimiento dentro de la bodega.

Al no tener un sistema que lleve una contabilidad exacta de los productos existentes, dentro de las miles de posiciones de arriendo que existen en sus bodegas, complica bastante su facturación y administración, ya que se basa en un sistema obsoleto y varios papeles de guías.

Lo que se trata de hacer con este nuevo sistema es integrar todos los procesos existentes en un sistema consolidado para que la administración sea única y la información no se pierda, ni se altere en ningún paso correspondiente a estos procesos.

1.2 Planteamiento del Problema

Desde sus inicios la empresa Logistecsa ha trabajado sin un sistema específico de Administración de Bodegas y práctico para sus labores específicas, lo que ha determinado los altos niveles de errores en el manejo de su mercancía, tanto de surtido como de recibo. Esto implica una falta de control en el material que se recibe, almacena y procesa; provocando surtidos incorrectos y “pérdida” de artículos dentro del mismo almacén. Estas pérdidas no implican que los artículos no estén, sino que no se sabe dónde han sido colocados, afectando directamente a la eficiencia y facturación. Además existe una falta de conocimiento del estado y niveles de inventario exactos en cualquier momento. El inventario solo se realiza cada fin de mes, restando control sobre los procesos encomendados. Este factor implica la carencia de un adecuado manejo de la información

que genera la operación del almacén y por ello no se conoce la situación y estado real del mismo.

Estas falencias llevaron a la empresa a definir la necesidad de implementar una nueva forma de administrar, con los consiguientes métodos, procedimientos, reglas y controles.

Un WMS deberá ayudar a mejorar todos los procesos, ya que no solo controlará los mecanismos y procesos actuales, sino que implementará nuevos y mejorados procesos de manejo administrativo y operativo.

Los actuales sistemas de control y registro tiene varias deficiencias en el procesos actual de un almacén, estos sistemas son muy vulnerables a errores humanos ya que casi toda la información que se almacena se la ingresa manualmente, lo que permite crear incoherencias en la base de datos y futuros problemas tanto para el administrador del sistema como para toda la compañía en general.

El manejo de comprobantes, listas, pedidos y envíos en su mayoría son papeles físicos los cuales producen un gran gasto a la compañía y problemas logísticos a los trabajadores ya que al no manejar una única información se producen ciertos errores al momento de transferir información de un área a otra.

La información en tiempo real, para los administradores y clientes es muy limitada ya que no siempre es real y veraz. También existe un limitado acceso a la información de la bodega ya que no existe una interfaz amigable con el usuario para hacer las consultas deseadas dado que se hacen directamente desde el gestor de bases de datos.

1.3 Justificación

La gerencia de la empresa Logistecsa considera necesario la ejecución del proyecto “Diseño, desarrollo e implementación de un sistema WMS (Warehouse Management System) para la empresa Logistecsa” como una nueva forma de renovación de sus procesos y metodologías actuales de operación y administración.

Un sistema WMS puede reducir el tiempo de espera limitando los movimientos del inventario y mejorando la ubicación de estos, dándole más crédito a un ambiente puntual. Como resultado, la necesidad de un stock seguro se ve reducido, lo cual incrementa el movimiento de mercadería y trabajo.

Adicionalmente, para reducir los requerimientos de un stock permanente, un sistema WMS puede usualmente incrementar la capacidad de las bodegas, al manejar más eficientemente la localización de los ítems en relación a las entregas, despachos, picking y acondicionamiento. Este eficiente aumento de capacidad puede mejorar la productividad, reduciendo significativamente los costos de mantenimiento de mercadería en las bodegas.

La implementación de un WMS a tiempo real significa reducir el papeleo de documentos, tradicionalmente asociado con las operaciones de los mega depósitos. También asegura una oportuna y acertada información del movimiento dentro del depósito. Recibir reportes, boletas de picking , reporte de kardex, lista de paquetes, etc., los cuales son normalmente mantenidos en papeles y copias de archivos, ahora se los puede manejar todos simplemente con una administración eficiente bajo una base de datos.

Este proyecto maneja información confidencial, por lo tanto se anticipa, se acepte que la información no sea entregada como documento requisito de graduación por la característica reservada del proyecto pero para compensar esto, se introducirá información de prueba para que se realicen las exámenes necesarios al sistema.

1.4 Objetivos

Objetivos Generales

Diseñar y Desarrollar un sistema WMS (Warehouse Management System) bajo la metodología MSF (Microsoft Solution Framework) y la plataforma de desarrollo Microsoft .NET C# para la empresa Logistecsa.

Objetivos Específicos

- Revisar y analizar los fundamentos teóricos de los sistemas administrativos de almacenes (WMS).
- Describir las distintas fases de la metodología Microsoft Solution Framework
- Identificar las diferentes herramientas de software para el desarrollo del WMS en la empresa Logistecsa S.A.
- Realizar el análisis de requerimientos y planificación del proyecto de software.
- Realizar el desarrollo y las pruebas del sistema WMS.

1.5 Alcance

Este proyecto pretende realizar una plataforma de manejo de depósitos (WMS-Warehouse Management System) para la empresa Logistecsa, dado que actualmente en el país no existe ningún proveedor de software que cumpla con las necesidades de la empresa.

Además, para el desarrollo de este software se utilizarán herramientas de última tecnología como es el Visual Studio 2008 y el Microsoft Solution Framework, como metodología de elaboración y diseño.

Ingresos

Este permitirá a los usuarios ingresar tanto nueva mercadería, importaciones, transferencia de inventario, como devoluciones de diferentes clientes, haciendo una selección por cada ítem, lote y cliente.

Mediante una interfaz fácil y muy dinámica de usar, controlará que los ingresos sean los más precisos, ya que es en este paso donde la mayoría de usuarios ingresa datos falsos o incoherentes. Además, tendrá un módulo en cual se pueden cargar archivos planos provenientes de otros sistemas, para la integración con el inventario directamente al sistema, sin pasar por un proceso humano.

Generación de Ingresos

En esta parte, situada posterior al ingreso, el sistema generara reportes automáticos con todos los ítems ingresados exitosamente al sistema, el cual podrá utilizarse en las distintas áreas de la empresa y enviárselos a los clientes como confirmación.

Estos reportes tendrán detallada toda la actividad correspondiente con el ingreso de estos productos al sistema.

Inventario

El inventario será manejado por una interfaz totalmente segura que no permita ingresos y egresos erróneos, habrá siempre una autenticación de usuarios para mantener

un registro de los movimientos hechos. Además, en este módulo se podrá manejar e intercambiar posiciones dentro del depósito para mantenerlo actualizado día a día.

Reserva de Inventario

Esta área permitirá a los distintos usuarios crear reservas de los productos almacenados para que no sean tomados como inventarios y se produzcan errores de repetición o sobre pedido. Esta parte es fundamental para el buen funcionamiento de sistema, ya que actualmente presenta grandes errores dentro de la empresa.

Piking

Este proceso dentro del sistema es el área de preparación de pedidos para ser despachados, acá se hará el conteo final, pesaje y confirmación del pedido, antes de que pase al despacho final. Este módulo permitirá controlar el egreso final del inventario y así mantenerlo actualizado.

Generación de entregas

En esta parte del sistema se generará un documento completo donde se describirá todos los detalles de la mercadería saliente del inventario, con destinatarios y un completo resumen de la carga que contiene. Este documento es de vital importancia para la empresa y sus clientes, ya que en base a este se efectúa tanto la entrega como la facturación.

Administración de Usuarios

Este módulo se encargará de manejar los diferentes niveles de acceso dentro de la plataforma, manejando siempre la información de una manera confiable y segura.

Además se registrará cada movimiento que exista dentro del sistema para llevar un control exacto de los usuarios.

Interfaces de modificación

El sistema permitirá a usuarios con sus permisos pertinentes a hacer modificación a la información, registrando siempre los movimientos. Estas correcciones son permitidas solo en casos de error humano.

Proyecto General

Según el acuerdo con la empresa, el proyecto será diseñado y desarrollado por parte del estudiante Andrés Marchant Castelnuovo, entregando en su totalidad con el código, manuales e información técnica según la metodología aprobada (Microsoft Solution Framework).

CAPÍTULO II

FUNDAMENTACIÓN TEÓRICA

2.1 Warehouse Management System

2.1.1 Definición de un WMS

Un sistema WMS (Warehouse Management System) es un software aplicativo que soporta día tras día operaciones de bodegas. El WMS permite una administración centralizada de tareas como el seguimiento de inventario, niveles y posiciones de stock. Un sistema WMS puede funcionar solo o pertenecer a un sistema de planificación de recursos empresariales o bien en sus siglas en inglés ERP (Enterprise Resource Planning).

Al principio los sistemas administrativos de bodegas (WMS) solo podían proveer funciones básicas de posiciones, actualmente las aplicaciones WMS pueden ser tan complejas y manejan tanta información que requieren personal dedicado para manejarlos. Sistemas High-end pueden ser utilizados para el rastreo y observación de rutas como las tecnologías de radio frecuencia o reconocimiento de voz.

No importa que tan simple o complicada pueda ser la aplicación, el objetivo de un WMS se mantiene en proveer administración con la información necesaria para hacer más eficiente el control del movimiento interno de productos dentro de las bodegas.

Una vez que la información fue recolectada, se sincroniza en tiempo real con la base de datos. La base de datos puede proveer diferentes tipos de reportes sobre el estado de la bodega, tanto como de sus productos como de su espacio utilizado.

El objetivo de un WMS es proveer un conjunto de procesos computarizados para manejar el stock y mantener siempre en tiempo real las posiciones tanto libres como ocupadas permitiendo así tener un control eficiente sobre las bodegas.

El WMS no es simplemente un administrador informático dentro de los límites de las bodegas, este es mucho más complejo y va más allá de los límites físicos. Administración de inventario, planificación de inventario, administración de costos, aplicaciones IT y tecnología de comunicaciones están relacionada con el WMS.

2.1.2 Historia de los WMS

Los principios de los WMS se remontan a la creación de graneros para almacenar comida, los cuales históricamente fueron construidos para alimentar durante los tiempos de sequías. Desde que los exploradores europeos empezaron a crear rutas de comercio con otras naciones, las bodegas crecieron y tomaron importancia dentro de estos mercados.

Cuando las vías férreas empezaron a expandirse, la creación de depósitos para el almacenaje de materiales fue necesaria. En 1891, la asociación americana de almaceneros se organizó para competir con las compañías de ferrocarriles por los depósitos de envíos.

El presidente Theodore Roosevelt impulsó considerablemente la Comisión Interestatal de Comercio con el paso de la Ley Hepburn en 1906. El negocio de almacenaje y sus sistemas de administración empezaron a crecer después que el gobierno de Estados Unidos pusiera más restricciones a las ferroviarias.

La segunda guerra mundial impactó directamente en el almacenaje de muchas maneras, incluida la necesidad de incrementar los tamaños de los almacenes y la necesidad de métodos de almacenaje y ubicación más modernos de los productos y materiales.

Mientras la producción en masa crecía por la manufactura del momento, la necesidad de capacidades de almacenamiento crecía proporcionalmente.

Fundada en 1953, Walters Wholesale empezó con un modesto local y logró crecer hasta convertirse en unos de los mayores almacenes de distribución de la industria eléctrica en el sur de California. Para incrementar su eficiencia y acomodar su rápido crecimiento, Walters Wholesale pasó de la forma tradicional de recoger y empacar los productos a un sistema administrativo de almacenamiento, aplicando muchos de los métodos que actualmente se conocen.

Luego de la introducción de la informática en todos los procesos industriales, varias compañías americanas en su mayoría, crearon sistemas básicos de control de inventario donde simplemente se llevaba un registro electrónico de los productos en las bodegas.

Con la revolución tecnológica en los años 70's, varias compañías de desarrollo vieron un gran potencial en el área de logística y almacenaje, donde implementaron software de calidad a todos los procesos de almacenaje y organización de estas empresas. El software creado para esta área del mercado fue evolucionando hasta la fecha de hoy donde se incorporan muchas tecnologías como lectores láser, radio frecuencia y un sin número de aplicativos para crear sistemas administrativos de almacenes (WMS).

2.1.3 Los WMS en la Industria

La industria de logística se encuentra en recuperación desde la recesión que empezó al principio del siglo XXI, provocada en parte por la debacle de las empresas.com quienes eran sus principales clientes. También fue copada por los métodos de distribución como el

just-in-time (justo a tiempo), convirtiendo innecesarios a los grandes almacenes, ya que el producto pasa de la fábrica directamente al cliente.

Hoy en día los operadores de almacenes tratan de convertirse en mucho más que una simple bodega, se están transformando en proveedores logísticos o como en sus siglas 3PLs (Third Party Logistics Providers) los cuales proveen una gama de servicios y funciones. Además de las típicas funciones de almacenaje, hoy en día proveen de múltiples servicios como call centers, servicio de etiquetado y acondicionamiento.

Administración

Las formas más avanzadas de la gestión de tareas en la actividad de un trabajador, es supervisar cómo lleva a cabo cada una de éstas bajo los permisos, prioridades y proximidad física del trabajador a la zona de trabajo. Mientras más grande sea la bodega y más tareas se realicen en el menor tiempo posible, mayor va a ser la productividad dirigidas por un avanzado administrador de tareas dentro de un sistema WMS. En muchos casos, se incrementa ostensiblemente la productividad solo por cambiar el diseño y el flujo del almacén.

Los permisos pueden ser de diferentes formas, estos pueden deberse a que algunos trabajadores no tengan la experiencia necesaria para realizar operaciones dentro del WMS o dentro del almacén. También puede existir el caso de un trabajador que tenga acceso a partes bajas de los racks pero no a las altas, dado a que no sabe utilizar el montacargas. Por lo tanto, dentro del sistema debe haber una buena administración de permisos para que el funcionamiento interno tenga los mejores rendimientos.

Las Prioridades reflejan la importancia de una tarea. Órdenes urgentes, por ejemplo, deben tener una mayor prioridad que órdenes regulares, en el picking debe tener prioridad

sobre pallets put away. En centros de alta distribución con buen manejo administrativo de los procesos en los sistemas, se puede observar que el flujo interno también es eficiente dando los correctos parámetros de funcionamiento.

Productividad

El WMS ayudará a la racionalización de muchos de los procesos y a que se maneje más fácil y eficientemente el ciclo de productividad. La meta general del software logístico es aumentar márgenes de ganancia y reducir los tiempos, para dar al negocio una ventaja competitiva en el mercado. Además, integrar soluciones logísticas que ayudarán a sincronizar la cadena de suministro, ahorrando costos, eliminando ineficiencias e incrementando los ingresos. Con toda esta solución se podrá asegurar órdenes más precisas, inventarios al 100% de exactitud y mejores entregas.

Con un WMS el personal de las bodegas no tiene que gastar tiempo en leer y archivar papeles, los cuales por lo general, son las guías de pedidos, información de productos, etc. Un buen sistema puede ayudar a ahorrarse expansiones innecesarias, determinando una buena ubicación de los productos y haciendo más eficiente la utilización de espacio disponible.

Dado que el sistema mantiene siempre en tiempo real los espacios utilizados, espacios disponibles y espacios nulos, no hay posibilidad de tener un control errado y reduce drásticamente el error humano.

Es muy difícil completar las áreas de marketing o manufactura sin una herramienta logística. Esta involucra la integración de la información, transporte, inventario, almacenaje, manejo de materiales y embalaje. La responsabilidad de la operación logística

es la reubicación geográfica de las materias primas, trabajar en procesos e inventarios de acabado cuando sea necesario, al menor costo posible.

La implementación exitosa de este software llevará a un incremento de la productividad, incrementará la colaboración entre áreas, bajará el tiempo de ingreso/egreso y aumentará la seguridad de la información.

Este WMS será diseñado bajo un marco teórico basado en el MSF y desarrollado en la plataforma Windows Visual Studio 2008. Estas dos herramientas dan al desarrollador una rica contribución al software ya que permiten una integración total con la mayoría de procesos actuales en la compañía y una alta compatibilidad con las bases de datos actualmente utilizadas por la empresa. Esta aplicación de desarrollo toma ventaja de la infraestructura y de las habilidades del personal que actualmente ejercen los procesos de almacenamiento y que llevan haciéndolo durante un largo período de tiempo.

Un sistema WMS es una parte fundamental de los procesos de la cadena de abastecimiento y las principales metas de este sistema serán el movimiento y almacenaje de materiales dentro del almacén y los procesos que son asociados a estas transacciones, incluyendo pedidos, ingresos, putaway y picking. El sistema también dirigirá y optimizará el stock en putway basado en información en tiempo real sobre el estado y su utilización.

El *Warehouse Management System* proveerá un conjunto de procesos computarizados para manejar el ingreso de inventario como los pedidos dentro del almacén, modelar y manejar la representación lógica del espacio físico dentro de las bodegas, administrar el inventario dentro de los almacenes para permitir un nexo perfecto para ordenar el procesamiento y gestión de la logística con el fin de recoger, empacar y enviar los productos fuera de los almacenes.

2.2 Metodología Microsoft Solution Framework

2.2.1 Definición de MSF

El proceso del MSF describe un alto nivel de secuencias de actividades para construir y desplegar soluciones de Innovación Tecnológica (IT). Al alcanzar esta serie de procesos, estos son lo suficientemente flexibles para acomodar una serie de proyectos IT.

MSF combina dos estándares industriales de diseño: el modelo de cascada, el cual enfatiza el logro de sub fases y el modelo espiral, el cual se enfoca en la necesidad de continuidad en el refinamiento de requerimientos y estimaciones para un proyecto.

Un aspecto innovador del proceso MSF es que este cubre el ciclo de vida de un proyecto desde el inicio al despliegue de la solución. Esto ayuda a los equipos de desarrolladores a enfocarse en las bases de la empresa para plasmarlo en el sistema, porque ningún valor será obtenido hasta que la solución no esté desplegada y operativa. Para tener un panorama más claro se puede observar la figura 1.1 que se despliega a continuación sobre el ciclo de vida del MSF.

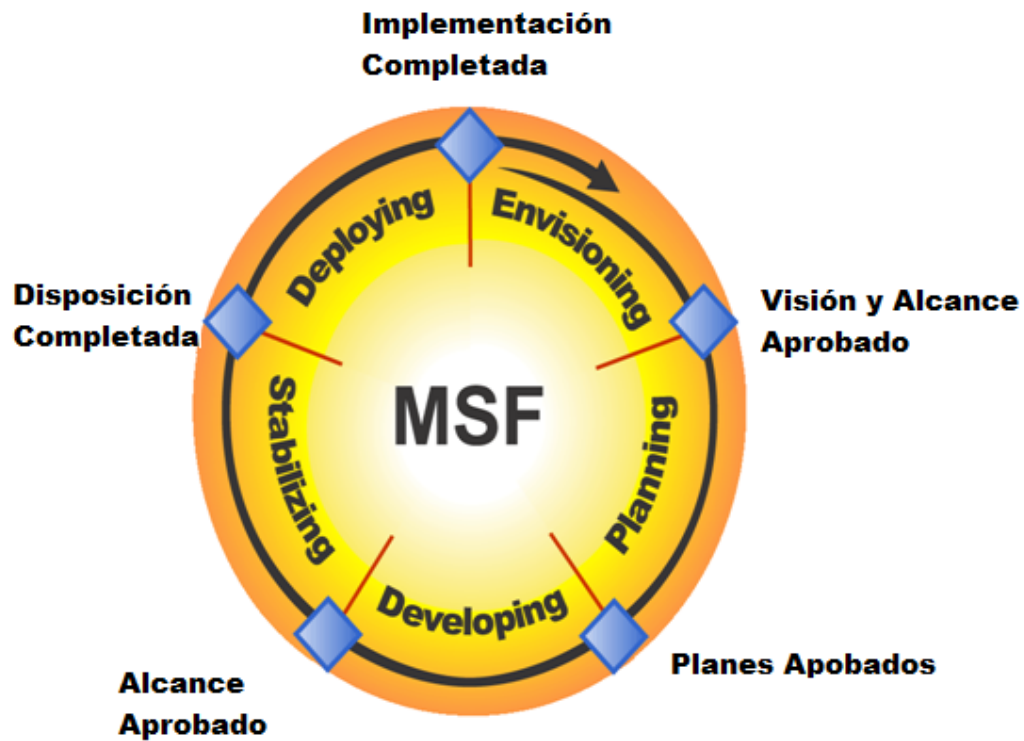


FIGURA N° 1.1: Proceso de Diseño MSF¹

Todo proyecto exitoso sigue una metodología para alcanzar sus objetivos. MSF fue creada para maximizar el éxito de proyectos IT durante todo el ciclo de desarrollo.

A diferencia de una metodología prescriptiva, MSF proporciona un marco flexible y escalable para satisfacer las necesidades de cualquier organización, la guía principal de MSF consiste en principios, modelos, procesos y elementos tecnológicos, los cuales la mayoría de proyectos atraviesan

1. Microsoft (s/f) Microsoft Solutions Framework. En: <http://www.microsoft.com/technet/prodtechnol/exchange/guides/ExMgmtGuide/62fab0e2-a570-4921-a0f1-7fdc6fff2a9.mspx?mfr=true>. Consultado el 3 de noviembre de 2010.

2.2.2 Prevención

En esta sección de la metodología MSF, es un estado temprano de la planificación. El principal objetivo de esta fase es llegar a una visión común entre los desarrolladores y los clientes para satisfacer ambas ópticas del proyecto. Durante el estado Prevención se crean claras definiciones sobre los problemas existentes y cómo van a ser solucionados.

La planificación de estas acciones se realizará en la etapa de Planificación y serán ejecutadas en la etapa de desarrollo. La etapa de Prevención termina en el momento que ambas partes (desarrollador/cliente) logran un acuerdo sobre el alcance del proyecto, definiendo la visión de este, sus objetivos y alcance.

Los puntos clave para el equipo de desarrollo durante la etapa de Prevención son:

- Durante el desarrollo de la visión y alcance se describen los objetivos, limitaciones y bases del proyecto. Asimismo, se explican los límites operativos del desarrollo y la manera en que el equipo atacará los distintos puntos a solucionar, de esta manera quedan definidos los requerimientos tanto técnicos como logísticos del proyecto en cuestión.
- Se documenta la estructura y cómo el equipo manejará el proyecto. Se definen los estándares, procesos, recursos del proyecto y limitantes.
- Se crea una sección de riesgos donde se identifican y se analizan los riesgos que pueda tener el proyecto durante toda su vida de desarrollo. Este análisis incluye la continencia y mitigación de los riesgos para ayudar al proceso de desarrollo a manejarlos.

2.2.2.1 Identificación de los objetivos

Antes que nada es muy importante tener claros los objetivos del proyecto WMS ya que resultan indispensables para el éxito. Los objetivos generales son definidos por los desarrolladores y por los clientes, dado que se debe llegar a un acuerdo entre lo deseado y lo realizable. Para llegar a tal consenso ciertas metas deben ser incluidas:

- Las metas del proyecto, una vez definidas, se usan para crear una visión del proyecto. Esta visión es la clave para tomar las acciones y decisiones necesarias por el equipo durante el desarrollo del WMS.
- Los objetivos son utilizados para crear las limitaciones, especificando los requerimientos que servirán para detallar los planes de planificación y desarrollo.
- Los objetivos sirven también como bases para establecer conflictos y prioridades, que guiarán la toma de decisiones (horarios, opciones, recursos, presupuestos y restricciones entre otras) necesarias en el transcurso del proyecto.

Objetivos del negocio

Los objetivos de negocio son establecidos para tomar ventaja de las oportunidades o para solucionar un problema de negocio. Cada organización tiene objetivos específicos. En el caso de Logistecsa S.A, su objetivo prioritario en el área de informática, almacenaje, picking, despachos y acondicionamiento, es la unificación de información mediante un sistema que les permite tener la información segura, en tiempo real y lo más fácil de comprender. De estos objetivos generales se derivan las metas específicas, las cuales son casi generales en todo negocio:

- Reducir Costos de las plataformas IT.
- Maximizar la ganancia de la inversión.

Reducir Costos de las plataformas IT

Para reducir los costos de un sistema no solo es prioritario el precio de desarrollo o la compra de Hardware, sino también el costo de entrenar a los usuarios, mantener y darle mantenimiento al sistema WMS. Para evaluar la reducción de costos del sistema hay varios factores a considerar:

- **El Costo del Hardware:** Evaluar el costo de servidores, equipo de redes los cuales serán necesarios para el buen funcionamiento del WMS dentro de la empresa y sus sucursales tales como access points, routers, nics y otros. Además de las adecuaciones necesarias en el actual data center para los nuevos equipos.
- **El Costo de Software:** el costo de los programas de desarrollo como el Visual Studio .NET 2008, sistemas operativos de servicios como el Windows Server 2005, sin contar las terminales con sus respectivos sistemas operativos basados en Windows S.O.
- **La Fiabilidad del sistema:** si el sistema WMS falla o colapsa, el costo de recuperarlo o arreglarlo subiría.
- **Los costos de mantenimiento:** La administración y mantenimiento del sistema es uno de los factores más costosos para cualquier organización con un departamento IT.

Maximizar la ganancia

Para estimar las ganancias de un proyecto de desarrollo de un nuevo sistema de almacenaje (Warehouse Management System) se deben tomar las siguientes consideraciones:

- **Creación de nuevas Oportunidades de Negocio:** La creación de un nuevo sistema WMS genera nuevas capacidades tecnológicas a la empresa y conlleva a mejores capacidades productivas.
- **Oportunidades para mejorar la Productividad:** Al tener un sistema más avanzado que permite controlar la información y tener un acceso en tiempo real de todas las funciones de la bodega, se genera automáticamente un aumento considerable en la productividad y eficacia de los procesos internos.
- **Fidelidad del WMS:** Al tener un sistema WMS más confiable, aumenta la precisión de las operaciones, bajando el tiempo de recuperación o arreglo de errores, generando menor costo para la empresa Logistecsa.

2.2.2.2 Identificar los Objetivos del Diseño

Los objetivos del diseño se enfocan más en los atributos de una solución al problema planteado. Cada organización tiene objetivos específicos únicos, aunque existen ciertas metas generalizadas para todas las empresas:

- **Alta Disponibilidad y Fiabilidad:** La solución debe procesar diferentes acciones sin interrupciones, hasta en momentos de falla. El sistema WMS se basará en una plataforma SQL Server, la cual permite la cauterización en diferentes grupos conectados. Si existe alguna falla en los sistemas operativos, hardware o en el mismo WMS SQL Server puede ser configurado para fallar sobre otro nodo, minimizando así el tiempo de apagado.
- **Alta Seguridad:** En el negocio, la seguridad de la información es vital, lanzado sobre “Trustworthy Computing Initiative”, Windows Server 2005 fue diseñado con énfasis en la seguridad. Además, dentro de la plataforma WMS existirán diferentes niveles de seguridad, dependiendo el usuario y su nivel de acceso.
- **Gran Desempeño y Estabilidad:** El sistema debe desempeñarse en un nivel aceptable, aun cuando el número de usuarios aumente. Bajo la plataforma Sql Server y un diseño multicapa, la aplicación desarrollada en C# debe cumplir con un gran desempeño dentro de la empresa, creando la posibilidad de aumentar la carga en cualquier momento y con un mínimo costo de implementación.
- **Fácil Operación:** El sistema debe proveer una administración fácil y eficiente al momento de instalar, administrar y mantener durante todo su periodo de instalación. Por esta razón se diseña el sistema con objetos Windows y

semejantes para darle al personal mayor comodidad al momento de migrar al sistema WMS. La mayoría de los empleados dentro de Logistecsa S.A están acostumbrados a trabajar sobre ambiente Windows.

2.2.3 Planificación

En la fase de planificación, el equipo define cómo va a operar, qué va a crear, cómo lo va a hacer y cuándo lo va a hacer. Durante esta etapa la visión inicial y el concepto de la solución definidas en la etapa anterior, van tomando una forma práctica de implementación con diseños y planes de ejecución.

La fase concluye cuando el equipo determina que los planes definen exitosamente la construcción del proyecto, además de la aprobación de patrocinadores y accionistas tanto en el plan como en cronograma.

La conclusión formal de la planificación se da cuando se ejecuta el punto de Aprobación de la Planificación del Proyecto.

Los puntos claves para la fase de planificación son:

- **Especificaciones de funcionalidad.** La función de estas especificaciones es crear un repositorio virtual del proyecto, donde los parámetros y necesidades se juntan de la fase de planificación. Este repositorio es el resultado de las actividades durante el diseño conceptual, lógico y físico de la fase de planificación.

Este repositorio puede incluir modelos tales como diagramas de casos, escenario de usos, lista de características, modelos gráficos de interfaces, diseño de base de datos y muchas cosas más.

- **Plan maestro de Proyecto** El plan maestro del proyecto es una colección de planes individuales que direcciona las tareas a realizarse por cada uno de los roles de funcionalidad definidos en las especificaciones de funcionalidad. En el plan maestro se documentan las estrategias del equipo, que se usan para completar el trabajo. El concepto de la solución que el equipo desarrolla en la fase de Prevención provee un acercamiento de alto nivel, el cual es desarrollado en detalle en la fase de planificación.
- **Cronograma Maestro del Proyecto** El cronograma principal sincroniza la elaboración y procesos dentro del proyecto. Un cronograma bien planteado da una vista general del proyecto y sus primeros pasos a ejecutar.

2.2.4 Desarrollo

Introducción y Metas

Para un proyecto de desarrollo de un WMS, la fase de desarrollo es el momento en que el equipo crea sus componentes como la codificación y la infraestructura así como la documentación. Normalmente este trabajo consiste en tomar todos los parámetros y diseños de las anteriores fases y crear un sistema bajo esta metodología.

El trabajo de desarrollo y pruebas son los puntos principales de esta fase. Algunas tareas de desarrollo pueden continuar hasta la fase de Estabilización, en respuesta a algunos resultados de pruebas efectuadas al sistema.

Uno de los primeros pasos para desarrollar el sistema WMS para la empresa Logistecsa S.A es crear una instancia de SQL Server, que proveerá los contenedores para desarrollar los esquemas y los objetos para las necesidades del proyecto. Al desarrollar una base de datos hay que enfocarse en la arquitectura utilizada para ser lo más eficiente posible.

Las dos implementaciones más comunes son las OLTP (Online Transaction Processing) y las DSS (Decision Support Systems). La OLTP se caracteriza por un gran número de pequeñas transacciones que frecuentemente añaden o modifican información. Los sistemas DSS, por otro lado, contienen grandes volúmenes de información, en su mayoría información estática importada desde bases OLTP, sistemas de archivos u otros.

“OLTP” es la sigla en inglés de Procesamiento de Transacciones En Línea (On Line Transaction Processing), es un tipo de sistemas que facilitan y administran aplicaciones transaccionales, usualmente para entrada de datos y recuperación y procesamiento de transacciones (gestor transaccional). Los paquetes de software para OLTP se basan en la arquitectura cliente-servidor, ya que suelen ser utilizados por empresas con una red informática distribuida.

El término resulta ambiguo, ya que puede entenderse "transacción" en el contexto de las "transacciones computacionales" o de las "transacciones en bases de datos". También podría entenderse en términos de transacciones de negocios o comerciales. OLTP también se ha utilizado para referirse a la transformación en la que el sistema responde de inmediato a las peticiones del usuario. Un cajero automático de un banco es un ejemplo de una aplicación de procesamiento de transacciones comerciales.

La tecnología OLTP se utiliza en innumerables aplicaciones, como en banca electrónica, procesamiento de pedidos, comercio electrónico, supermercados o industria.”²

“Debido a que hay muchos enfoques para la toma de decisiones y debido a la amplia gama de ámbitos en los cuales se toman las decisiones, el concepto de **sistema de apoyo a las decisiones (DSS** por sus siglas en inglés Decision Support System) es muy amplio. Un **DSS** puede adoptar muchas formas diferentes. En general, podemos decir que un DSS es un sistema informático utilizado para servir de apoyo, más que automatizar, el proceso de toma de decisiones. La decisión es una elección entre alternativas basadas en estimaciones de los valores de esas alternativas. El apoyo a una decisión significa ayudar a las personas que trabajan solas o en grupo a reunir inteligencia, generar alternativas y tomar decisiones. Apoyar el proceso de toma de decisión implica el apoyo a la estimación, la evaluación y/o la comparación de alternativas. En la práctica, las referencias a DSS suelen ser referencias a aplicaciones informáticas que realizan una función de apoyo”³

Desarrollo de la solución, diseño y arquitectura.

La fase de planificación del modelo del proceso de MSF incluye tres procesos de diseño: Conceptual, lógico y físico. Estos tres procesos no se desarrollan en paralelo. En cambio, su inicio y puntos de finalización están juntos. Estos procesos son dependientes los unos de los otros. El diseño lógico es dependiente del diseño conceptual y el diseño físico es dependiente del diseño lógico. Ver tabla 2.1.

² Wikipedia, la enciclopedia libre (s/f). OLTP. En: <http://es.wikipedia.org/wiki/OLTP>. Consultado el 3 de noviembre de 2010.

³ Alter, S. L. (1980). Decision support systems: current practice and continuing challenges. Reading, Mass., Addison-Wesley Pub Consultado el 5 de noviembre de 2010

TABLA N° 2.1.: Diseño Multi Capa

Tipo diseño	Perspectiva	Propósito
Diseño Conceptual	Enfoca el problema desde la perspectiva del usuario y del negocio.	Define el problema y la solución en términos de los escenarios de uso y requerimientos específicos.
Diseño Lógico	Enfoca el problema desde la perspectiva del equipo de diseño y arquitectura	Define la solución como un orden lógico de objetos y servicios cooperativos
Diseño Físico	Enfoca el problema desde la perspectiva de los desarrolladores.	Define los componentes de la solución y tecnologías.

Construcción del Diseño Conceptual

El primer proceso de diseño en la fase de planificación es el diseño conceptual. Una vez que se ha completado el diseño conceptual se utiliza en la creación de procesos de diseño físico y lógico.

El diseño conceptual es el proceso de recopilación, análisis y priorización del negocio y la perspectiva del usuario del problema y la solución. Y luego la creación de un alto nivel de representación de la solución en la forma de los requerimientos detallados. EL equipo reúne los requerimientos de alto nivel durante la fase de avisoreamiento y los documenta en la forma de uso de casos y de escenarios. Mientras se produce el diseño conceptual el equipo refina estos requerimientos.

Desarrollo de los Requerimientos detallados

Un diseño conceptual graba los requerimientos del sistema en términos de los siguientes requisitos:

- Requerimientos del negocio
- Requerimientos del usuario
- Requerimientos del sistema
- Requerimientos de la operación

Requisitos del negocio

Estos describen las necesidades de la organización y esperanzas de solución. Estos requisitos existen en el nivel de administración de toma de decisiones y provee el contexto en el cual la solución operará.

Algunos ejemplos incluyen:

- El diseño del sistema WMS para la empresa Logistecsa S.A debe completarse en el marco de tiempo establecido.
- Las reglas y políticas de negocio existentes deben mantenerse en el nuevo sistema.
- El esfuerzo requiere que el desarrollo justifique el costo incurrido.
- El nivel existente de desempeño y funcionalidad debe mantenerse después del despliegue, incluyendo el recovery, disponibilidad y escalabilidad.

Requisitos del usuario

Estos definen el aspecto no funcional de la interacción del usuario con la solución. Ayudarán a determinar la interfaz del usuario y las expectativas de desempeño de la solución en términos de su confiabilidad, disponibilidad y accesibilidad. Además, los

requerimientos del usuario servirán para identificar el entrenamiento que los usuarios necesitarán para usar efectivamente la solución.

Algunos ejemplos de requisitos del usuario incluyen:

- El usuario debe ser capaz de usar la misma interfaz con la aplicación WMS que el sistema operativo en uso, para minimizar el entrenamiento del usuario.
- El tiempo que toma recopilar la información al nuevo sistema de un usuario debe disminuir.
- Debe haber manuales del usuario comprensibles para el nuevo sistema. Los manuales de entrenamiento deben ser considerados también para los nuevos usuarios.

Requisitos del sistema

Los requisitos del sistema especifican las transacciones detalladas y su secuencia en el sistema. Los requisitos ayudarán al equipo del proyecto a definir cómo interactuará la nueva solución con los sistemas existentes. Algunos ejemplos incluyen:

- Después de la migración el Nuevo sistema debería continuar inter operando con aplicaciones como hizo el sistema con plataforma UNIX anterior.
- El sistema no deberá requerir alguna otra credencial de usuario que no sean las que se dan al autenticarse a la red corporativa.
- El sistema debería soportar usuarios remotos e internos.

Requisitos operacionales

Estos describen lo que la solución debe entregar para maximizar la operabilidad y mejorar la entrega del servicio con tiempo y riesgos reducidos. Algunos ejemplos incluyen:

- El sistema debe permitir a los administradores ejecutar sus tareas de ambos lados, remotamente e in-situ.
- El sistema debería ser capaz de recuperarse de fallas críticas sin impactos mayores y dentro de los niveles de servicio.
- La base de datos debe incluir un proceso para administrar el tráfico total y el tiempo de respuesta dentro de los niveles de servicio establecidos.
- El sistema debe ser capaz de manejar los niveles variables de carga y transacciones del usuario. Además, el sitio debería estar designado de manera que pueda ser modificado y actualizado sin afectar el desempeño y disponibilidad.

Crear el Diseño de la Solución

El diseño lógico es el segundo paso en la creación de la solución de diseño. Después de identificar las necesidades del negocio y del usuario en el diseño conceptual, el diseño lógico define cómo las diferentes partes de la solución serán coordinadas y trabajarán juntas. El diseño lógico define las partes del sistema, provee un marco de trabajo para mantener todas las partes del sistema unidas, e ilustra cómo el sistema interactúa con los usuarios y con otros sistemas.

Mientras es creado el diseño lógico, el equipo toma en cuenta todos los requisitos del negocio, usuario, operacionales y de sistema que establecen la necesidad de seguridad, auditoría, ingreso, escalabilidad, administración de estado, manejo del error, licenciamiento, globalización, arquitectura de aplicaciones, integración con otros sistemas, etc.

Los objetos, servicios, atributos y relaciones pueden obtenerse de los diferentes escenarios de uso, que fueron desarrollados en la fase de diseño conceptual.

El diseño lógico representa una visión lógica del sistema completo. Equipos individuales como aquellos para el desarrollo de la interfaz del usuario (IU), el desarrollo de la base de datos y de aplicaciones, pueden tomar esta representación y comenzar a construir diseños detallados por su dominio cuando se está haciendo el diseño físico.

Interfaz para usuarios de alto nivel y diseño de base de datos

La lista de objetos y servicios da al desarrollador una idea del tipo de funcionalidad esperada por los usuarios. El desarrollador puede usar esta información para diseñar elementos de interfaz del usuario como botones, campos de texto, ítems de menú, etc.

Similarmente, el objeto y atributo de la información del diseño lógico puede ser utilizado para desarrollar una base de datos inicial, equivalente a la base de datos de del WMS anterior, siempre y cuando esta sea útil y se acople al sistema nuevo.

Construcción del Diseño Físico

El diseño físico es el proceso de descripción de los componentes, servicios y tecnologías de la solución desde la perspectiva del desarrollador. El objetivo del diseño es proveer claridad y detalle para el programador, a fin de desarrollar sus componentes, ya sean la interfaz de usuario, lógica de negocios, base de datos o infraestructura.

EL diseño físico debería incluir:

- Definición de clase de las aplicaciones.
- Esquema de base de datos para la solución.
- Despliegue base del modelo que provee:
 - La topología de la red, mostrando locaciones de hardware e interconexiones.

- Los datos y componentes de la tecnología, lo que indica las locaciones de los componentes de la solución, servicios, y almacenamiento de datos en relación a la topología de red.
- Especificaciones de componentes que incluyen la estructura interna de los componentes e interfaces de los componentes.
- Modelos de programación que identifican las guías de implementación para conexiones compartidas, manejo de errores, seguridad, y documentación codificada.

El diagrama del diseño físico permitirá ver cómo todos los componentes se conectan juntos. Para completarse, incluirá elementos de fuera, como firewalls o conexiones de red que no están directamente relacionadas a la solución. Esto ayudará a probar la seguridad del sistema y podrá mostrar potenciales cuellos de botella, o resaltar puntos de falla.

Desarrollo de los planes de proyecto

El diseño de la solución que fue creado en la fase de planteamiento es usado como la base para crear el plan del proyecto. Los planes de proyecto incluyen:

- Plan de desarrollo
- Plan de estabilización (incluye plan de prueba y plan piloto)

Adicionalmente a los planes mencionados arriba, debe crearse un checklist de entrenamiento que identificará las habilidades requeridas y las existentes.

Planificación Pre-Instalación

Para instalar el software SQL Server y la base de datos hay que seguir estos simples pasos:

- **Subdirectorios independientes.** Los archivos deben estar separados por categorías e instanciados para minimizar los efectos y mejorar la navegación. Cuando se nombran las instancias de SQL Server al crearse, los directorios asociados son creados con nombres instanciados para propósitos de identificación.
- **Integridad del directorio home.** Mantener el Servidor Sql separada de la información. Esto permite al software ser movido o borrado sin afectar la aplicación.
- **Separación de bases de datos.** La información guardada del sistema en la base de datos Master debe estar separada de la demás bases de datos.
- **Separación del contenido de tablas.** Cada instancia de SQL Server viene con un sistema de bases de datos **master**, **msdb** y **tempdb**.

Instalación

Las opciones de instalación deben ser determinadas antes de instalar SQL Server.

- **Instanciar nombres.** SQL Server tiene la capacidad para instanciar por default donde estas toman el nombre del servidor en la red. También pueden ser creadas instancias múltiples con asignación de nombres.
- **Librerías de Red.** SQL Servers puede comunicarse usando varias redes y protocolos TCP/IP. Para minimizar la utilización de recursos, solo es necesario configurar los puertos extras al TCP/IP.
- **Modo de autenticación.** Las opciones dentro de SQL para autenticar usuarios es tanto por las cuentas de Windows o un usuario privado.⁴

⁴ Microsoft (2008) Microsoft Tech Net. Installing SQL Server. En: <http://technet.microsoft.com/en-us/library/ms143516%28SQL.90%29.aspx>. Consultado el 4 de mayo de 2010.

2.3 Herramientas de Diseño y Desarrollo

2.3.1 Microsoft SQL Management Studio 2005

SQL Server Management Studio es un entorno integrado para obtener acceso a todos los componentes de SQL Server, así como para configurarlos y administrarlos. SQL Server Management Studio combina un amplio grupo de herramientas gráficas con un editor de texto enriquecido, para ofrecer acceso a SQL Server a los programadores y administradores, sin importar su nivel de especialización.

SQL Server Management Studio combina las funciones del Administrador corporativo y el Analizador de consultas, herramientas incluidas en versiones anteriores de SQL Server, en un único entorno. Además, SQL Server Management Studio proporciona un entorno para administrar Analysis Services, Integration Services, Reporting Services y XQuery. Este entorno ofrece a los programadores una experiencia familiar y proporciona a los administradores de bases de datos una herramienta única para realizar sus tareas con la facilidad de las herramientas gráficas y una experiencia de secuencias de comandos enriquecida.

SQL Server Management Studio incluye las siguientes características generales:

- Compatibilidad con la mayoría de las tareas administrativas de SQL Server 2005 y SQL Server 2000.

- Un entorno único integrado para administración y edición de SQL Server Database Engine (Motor de base de datos de SQL Server).
- Nuevos cuadros de diálogo para la administración de objetos de SQL Server Database Engine (Motor de base de datos de SQL Server), Analysis Services, Reporting Services, Notification Services y SQL Server 2005 Compact Edition, lo que permite ejecutar las acciones inmediatamente, enviarlas a un editor de código o escribirlas en secuencias de comandos para ejecutarlas posteriormente.
- Cuadros de diálogo no modales y de tamaño variable que permiten obtener acceso a varias herramientas mientras un cuadro de diálogo está abierto.
- Un cuadro de diálogo común de programación que permite realizar acciones de los cuadros de diálogo de administración en otro momento.
- Exportación e importación del registro de servidor de SQL Server Management Studio desde un entorno de Management Studio a otro.
- Guardado o impresión de archivos de plan de presentación XML o de interbloqueo generados por el Analizador de SQL Server, revisión posterior o envío a los administradores para su análisis.

- Un nuevo cuadro de mensaje de error e informativo que presenta mucha más información, permite enviar a Microsoft un comentario sobre los mensajes, copiar mensajes en el Portapapeles y enviar fácilmente los mensajes por correo electrónico al equipo de soporte.
- Un explorador Web integrado para una rápida exploración de MSDN o la Ayuda en pantalla.⁵

Secuencias de comandos

El Editor de código de SQL Server Management Studio contiene editores de secuencias de comandos integrados para crear secuencias de comandos Transact-SQL, MDX, DMX, XML/A y XML. Ofrece las características siguientes:

- Ayuda dinámica para el acceso inmediato a la información relevante mientras se trabaja.
- Un amplio conjunto de plantillas y la posibilidad de crear plantillas personalizadas.
- Compatibilidad con la escritura y modificación de consultas o secuencias de comandos sin necesidad de conexión a un servidor.

⁵ MSDN (s/f) **Using SQL Server Management Studio**. En: <http://msdn.microsoft.com/en-us/library/ms174173.aspx>. Consultado el 15 Agosto del 2009

- Compatibilidad con secuencias de comandos para consultas y secuencias de comandos SQLCMD.
- Una nueva interfaz para ver resultados XML.
- Control de código fuente integrado para proyectos de secuencias de comandos y soluciones compatibles con el almacenamiento y la conservación de copias de secuencias de comandos a medida que evolucionan.
- Compatibilidad de Microsoft IntelliSense con instrucciones MDX.

Explorador de objetos

El Explorador de objetos de SQL Server Management Studio es una herramienta integrada para ver y administrar objetos en todo tipo de servidores. Ofrece las características siguientes:

- Filtrado por todo o parte de un nombre, esquema o fecha.
- Llenado asincrónico de objetos, con la posibilidad de filtrar objetos según sus metadatos.
- Acceso al Agente SQL Server en los servidores de réplica para administración.

Administrar servidores con SQL Server Management Studio

Microsoft SQL Server Management Studio es un potente cliente integrado diseñado para satisfacer los requisitos de administración de servidores del administrador de SQL Server. En Management Studio, las tareas administrativas se realizan mediante el Explorador de objetos, que permite conectarse a cualquier servidor de la familia de SQL Server y examinar de forma gráfica su contenido. Un servidor puede ser una instancia de Database Engine (Motor de base de datos), Analysis Services, Reporting Services, Integration Services o SQL Server 2005 Compact Edition.

Utilizar Management Studio para:

- Registrar servidores.
- Conectarse a una instancia de Database Engine (Motor de base de datos), SSAS, SSRS, SSIS o SQL Server 2005 Compact Edition.
- Configurar las propiedades del servidor.
- Administrar la base de datos y objetos de SSAS, tales como cubos, dimensiones y ensamblados.
- Crear objetos, tales como bases de datos, tablas, cubos, usuarios de base de datos e inicios de sesión.
- Administrar archivos y grupos de archivos.
- Adjuntar o separar bases de datos.
- Iniciar herramientas de secuencias de comandos.
- Administrar la seguridad.
- Ver registros del sistema.
- Supervisar la actividad actual.
- Configurar la réplica.

- Administrar índices de texto.

Desarrollar soluciones y proyectos en SQL Server Management Studio

Las soluciones y los proyectos de secuencias de comandos son contenedores que los programadores utilizan para organizar archivos relacionados en Microsoft SQL Server Management Studio, administre las soluciones y los proyectos de secuencias de comandos mediante el Explorador de soluciones.

Puede utilizar Management Studio como plataforma de desarrollo de secuencias de comandos de SQL Server, Analysis Services y SQL Server 2005 Compact Edition. También para desarrollar secuencias de comandos para bases de datos relacionales y multidimensionales, y todo tipo de consultas. La experiencia de desarrollo de Management Studio se ha mejorado con la mezcla enriquecida de eficaces editores de código. Mediante Management Studio puede hacer lo siguiente:

- Escribir consultas y secuencias de comandos para permitir los procesos de producción.
- Agregar información de conexión y otros archivos asociados al proyecto.
- Guardar las consultas y las secuencias de comandos junto con sus conexiones en un proyecto.
- Organizar los proyectos de secuencias de comandos en un único contenedor denominado solución.
- Guardar la solución en una base de datos de Microsoft Visual SourceSafe (VSS) u otro proveedor de control de código fuente de otro fabricante para

realizar un seguimiento de los cambios de desarrollo y administrar el ciclo vital.⁶

2.3.2 Microsoft Visio

Las herramientas de diagramación avanzada de Visio permiten simplificar las tareas complejas con elementos visuales dinámicos basados en datos.

La aplicación permite la utilización de plantillas profesionales y modernas dando un estilo más organizado y formal a los diagramas y flujogramas destinados a proyectos como este. En el diseño del WMS se utilizaron algunos diagramas de casos de uso para conceptualizar las acciones y procesos dentro de la empresa Logisteca S.A.

La simplicidad, las formas basadas en datos y el uso compartido en la Web convierten a Visio en una de las formas más eficaces de ver y comprender información importante.

Simplicidad

Visio fomenta el crecimiento empresarial y la eficacia operativa mediante la presentación de la gestión y de las TI y su infraestructura, con herramientas avanzadas de gestión de procesos y sencilla función de una navegación y creación de diagramas.

⁶ Microsoft (2008) Microsoft Tech Net. Administrar servidores con SQL Server Management Studio. En: <http://technet.microsoft.com/es-es/library/ms173793%28SQL.90%29.aspx>. Consultado el 18 de Septiembre del 2009

Conexión

Visio revela valoraciones y tendencias importantes, por medio del análisis de la información compleja y la vinculación de los datos, sistemas y proceso. Se pueden conectar diagramas de Visio con los datos y procesos en tiempo real. La visualización de los datos en tiempo real en los diagramas es muy importante para aplicar variables y cambios en todos los momentos

Posibilidad de compartir

Visio fomenta el cumplimiento, la eficacia y el trabajo en equipo, compartiendo complejos sistemas de TI mediante diagramas actualizables y conectados a los datos que están en la Red. Todo ello, gracias a Visio Services para SharePoint

SharePoint

“Microsoft SharePoint, también conocido como Microsoft SharePoint Products and Technologies, es una colección de productos y elementos de software que incluye, entre una selección cada vez mayor de componentes, funciones de colaboración, basado en el Explorador web, módulos de administración de proceso, módulos de búsqueda y una plataforma de administración de documento .

SharePoint puede utilizarse para sitios de web host que acceda a espacios de trabajo compartidos, almacenes de información y documentos, así como para alojar aplicaciones definidas como los wikis y blogs. Todos los usuarios pueden manipular los controles

propietarios llamados "web parts" o interactuar con piezas de contenido, como listas y bibliotecas de documentos”⁷.

Programación de Aplicaciones

Con Visio se puede desarrollar una visión completa de la información más representativa en un único diagrama actualizado de las necesidades para el proyecto.

- Se puede conectar fácilmente los diagramas a una o más fuentes, incluyendo Microsoft Excel, SQL Server y las listas de servicios de SharePoint.
- Los datos en tiempo real se los puede ver en forma de diagrama, según las condiciones que se definan, por medio de gráficos ilustrativos, como el color, los iconos, los símbolos y las barras de datos.⁸ [7]

2.3.3 Visual Studio C#

“C#” es un lenguaje con seguridad de tipos y orientado a objetos, sencillo pero eficaz, que permite que los programadores creen una gran variedad de aplicaciones. Junto a .NET Framework, Visual C# 2008 permite la creación de aplicaciones para Windows, servicios web, herramientas de bases de datos, componentes, controles, etc.

⁷ Christina Torode (s/f) SharePoint alternatives seek to fill in the gaps. TECH TARGET. En: <http://searchcio-midmarket.techtarget.com/tip/SharePoint-alternatives-seek-to-fill-in-the-gaps>. Consultado el 18 de Septiembre del 2009

⁸ VISIO.Microsoft (s/f) En: <http://visiotoolbox.com/2010/es/business-intelligence.aspx>. Consultado el 19 de Septiembre del 2009

C# es un lenguaje orientado a objetos elegante y con seguridad de tipos que permite a los desarrolladores compilar diversas aplicaciones sólidas y seguras que se ejecutan en .NET Framework. Se puede utilizar C# para crear aplicaciones cliente de Windows tradicionales, servicios Web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos, y mucho, mucho más. Visual C# 2010 proporciona un editor de código avanzado, cómodos diseñadores de interfaz de usuario, depurador integrado y numerosas herramientas más para facilitar el desarrollo de aplicaciones basadas en la versión 4.0 del lenguaje C# y la versión 4 de .NET Framework.

La sintaxis de C# es muy expresiva, pero también es sencilla y fácil de aprender. La sintaxis de C# basada en signos de llave podrá ser reconocida inmediatamente por cualquier persona familiarizada con C, C++ o Java. Los desarrolladores que conocen cualquiera de estos lenguajes pueden empezar a trabajar de forma productiva en C# en un plazo muy breve. La sintaxis de C# simplifica muchas de las complejidades de C++ y proporciona características eficaces tales como tipos de valor que admiten valores NULL, enumeraciones, delegados, expresiones lambda y acceso directo a memoria, que no se encuentran en Java. C# admite métodos y tipos genéricos, que proporcionan mayor rendimiento y seguridad de tipos, e iteradores, que permiten a los implementadores de clases de colección definir comportamientos de iteración personalizados que el código cliente puede utilizar fácilmente. Las expresiones Language-Integrated Query (LINQ) convierten la consulta fuertemente tipada en una construcción de lenguaje de primera clase.

Como lenguaje orientado a objetos, C# admite los conceptos de encapsulación, herencia y polimorfismo. Todas las variables y métodos, incluido el método Main que es el punto de entrada de la aplicación, se encapsulan dentro de definiciones de clase. Una clase

puede heredar directamente de una clase primaria, pero puede implementar cualquier número de interfaces. Los métodos que reemplazan a los métodos virtuales en una clase primaria requieren la palabra clave `override` como medio para evitar redefiniciones accidentales. En C#, una struct es como una clase sencilla; es un tipo asignado en la pila que puede implementar interfaces pero que no admite la herencia.

Además de estos principios básicos orientados a objetos, C# facilita el desarrollo de componentes de software a través de varias construcciones de lenguaje innovadoras, entre las que se incluyen las siguientes:

- Firmas de métodos encapsulados denominadas delegados, que habilitan notificaciones de eventos con seguridad de tipos.
- Propiedades, que actúan como descriptores de acceso para variables miembro privadas.
- Atributos, que proporcionan metadatos declarativos sobre tipos en tiempo de ejecución.
- Comentarios en línea de documentación XML.
- Language-Integrated Query (LINQ) que proporciona funciones de consulta integradas en una gran variedad de orígenes de datos.

Si necesita interactuar con otro software de Windows, como objetos COM o archivos DLL nativos de Win32, podrá hacerlo en C# mediante un proceso denominado "interoperabilidad". La interoperabilidad habilita los programas de C# para que puedan realizar prácticamente las mismas tareas que una aplicación C++ nativa. C# admite incluso el uso de punteros y el concepto de código "no seguro" en los casos en que el acceso directo a la memoria es totalmente crítico.

El proceso de compilación de C# es simple en comparación con el de C y C++, y es más flexible que en Java. No hay archivos de encabezado independientes, ni se requiere que los métodos y los tipos se declaren en un orden determinado. Un archivo de código fuente de C# puede definir cualquier número de clases, structs, interfaces y eventos.

Arquitectura

Los programas de C# se ejecutan en .NET Framework, un componente que forma parte de Windows y que incluye un sistema de ejecución virtual denominado Common Language Runtime (CLR) y un conjunto unificado de bibliotecas de clases. CLR es la implementación comercial de Microsoft de CLI (Common Language Infrastructure), un estándar internacional que constituye la base para crear entornos de ejecución y desarrollo en los que los lenguajes y las bibliotecas trabajan juntos sin ningún problema.

El código fuente escrito en C# se compila en un lenguaje intermedio (IL) conforme con la especificación CLI. El código de lenguaje intermedio y recursos tales como mapas de bits y cadenas se almacenan en disco en un archivo ejecutable denominado ensamblado, cuya extensión es .exe o .dll generalmente. Un ensamblado contiene un manifiesto que proporciona información sobre los tipos, la versión, la referencia cultural y los requisitos de seguridad del ensamblado.

Cuando se ejecuta un programa de C#, el ensamblado se carga en CLR, con lo que se pueden realizar diversas acciones en función de la información del manifiesto. A continuación, si se cumplen los requisitos de seguridad, CLR realiza una compilación Just In Time (JIT) para convertir el código de lenguaje intermedio en instrucciones máquina nativo. CLR también proporciona otros servicios relacionados con la recolección automática de elementos no utilizados, el control de excepciones y la administración de recursos. El código ejecutado por CLR se denomina algunas veces "código administrado",

en contraposición al "código no administrado" que se compila en lenguaje máquina nativo destinado a un sistema específico. En el diagrama siguiente se muestran las relaciones en tiempo de compilación y tiempo de ejecución de los archivos de código fuente de C#, las bibliotecas de clases de .NET Framework, los ensamblados y CLR.

La interoperabilidad del lenguaje es una característica clave de .NET Framework. Como el código de lenguaje intermedio generado por el compilador de C# cumple la especificación de tipos común (CTS), este código generado en C# puede interactuar con el código generado en las versiones .NET de Visual Basic, Visual C++ o cualquiera de los más de 20 lenguajes conformes a CTS. Un único ensamblado puede contener varios módulos escritos en diferentes lenguajes .NET, y los tipos admiten referencias entre sí como si estuvieran escritos en el mismo lenguaje.

Además de los servicios en tiempo de ejecución, .NET Framework también incluye una amplia biblioteca de más de 4.000 clases organizadas en espacios de nombres que proporcionan una gran variedad de funciones útiles para la entrada y salida de archivos, la manipulación de cadenas, el análisis XML, los controles de los formularios Windows Forms y muchas tareas más. La aplicación de C# típica utiliza continuamente la biblioteca de clases de .NET Framework para el tratamiento de las tareas comunes de "infraestructura".⁹ [8]

⁹ MSDN (s/f) Lo nuevo en el depurador de Visual Studio. En: <http://msdn.microsoft.com/es-EC/library/01xdt7cs.aspx>. Consultado el 21 de Septiembre del 2009

2.3.4 Microsoft Project

Microsoft Project (o MSP) es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.

CAPÍTULO III

VISIÓN Y PLANIFICACIÓN

3.1 Visión y Alcance

La Visión del Warehouse Management System es básicamente entregar a la empresa la información correcta y en el momento preciso, la estrategia a seguir contiene 3 etapas fundamentales: 1. Software Colaborativo, 2. Software Informativo y 3. Software servicial, las cuales explicaré detalladamente dentro del punto Visión Conceptual.

Alcance

Para la mayoría de los negocios relacionados con almacenes y bodegas, los sistemas WMS son críticos como herramienta administrativa y de control que permite a sus empleados producir mejores resultados. Mientras que el ambiente de trabajo este en constante cambio y los requerimientos de seguridad estén siempre presentes, las organizaciones estarán en búsqueda de soluciones que ayuden superar estos desafíos. Con el sistema WMS desarrollado para la empresa Logistecsa, los usuarios obtendrán una herramienta de avanzada que necesitan, además de un aumento en la eficiencia operacional requerida por los administradores del negocio. Adicionalmente, el sistema WMS proveerá acceso a la información requerida en cualquier momento y en cualquier lugar deseado.

Eficiencia Operacional

El sistema WMS desarrollado para la empresa Logistecsa facilitará la eficiencia operacional a través de las capacidades de optimización de hardware y de redes, las cuales a la vez ayudará la productividad y a la administración de los almacenes.

Algunas de las características que incluye este sistema son las siguientes:

- **Compatibilidad con Windows Server 2008 y Windows Vista.** El sistema WMS corre sobre Microsoft's next-generation server platforms, incluyendo Windows Server 2005 SP2 y Windows Server 2008. El sistema terminal permite correr también sobre Windows Vista y Windows Server 2008.
- **Aumentar la productividad de la Administración.** El sistema WMS ofrecerá toda una plataforma de consultas para el nivel Gerencial, permitiendo así un manejo mucho más efectivo y con información en tiempo real, además en una futura actualización, el sistema permitirá crear consultas web a través de una plataforma en ASPX ofreciendo toda una nueva gama de consultas tanto para gerencia como para clientes interesados en mantenerse siempre informados sobre la transacciones y estados de sus productos.
- **Extensibilidad de la Plataforma.** La aplicación de escritorios es fácilmente extensible a servicios web para habilitar un sin números de escenarios webs y soportar varios tipos APIs Open Source.

3.1.1 Beneficios, metas, objetivos y restricciones

Análisis de Beneficios

Concepto de Logística

“La logística es definida por la RAE como el conjunto de medios y métodos necesarios para llevar a cabo la organización de una empresa, o de un servicio, especialmente de distribución.”¹⁰

“En el ámbito empresarial existen múltiples definiciones del término logística que ha evolucionado desde la logística militar hasta el concepto contemporáneo del arte y la técnica que se ocupa de la organización de los flujos de mercancías, energía e información.

La Logística es fundamental para el comercio. Las actividades logísticas son el puente entre la producción y los mercados que están separados por el tiempo y la distancia.

La logística empresarial cubre la gestión y la planificación de las actividades de los departamentos de compras, producción, transporte, almacenaje, manutención y distribución”¹¹.

“Para el profesor Ronald H. Ballou la logística empresarial es «todo movimiento y almacenamiento que facilite el flujo de productos desde el punto de compra de los materiales hasta el punto de consumo, así como los flujos de información que se ponen en

¹⁰ Real Academia Española (2001). Diccionario de la Lengua Española, Vigésima segunda edición edición, Madrid (España)

¹¹ *Wikipedia, la enciclopedia libre (s/f) Logística empresarial: definiciones. En: http://es.wikipedia.org/wiki/Logistica#Log.C3.ADstica_empresarial:_definiciones. Consultado el 29 de Septiembre del 2009*

marcha, con el fin de dar al consumidor el nivel de servicio adecuado a un coste razonable”¹²

”El profesor Lambert integra el término logística en otro más general y la define como la parte de la gestión de la cadena logística (Supply Chain Management (SCM)) que planifica, implementa y controla el flujo eficiente y efectivo de materiales y el almacenamiento de productos, así como la información asociada desde el punto de origen hasta el de consumo con el objeto de satisfacer las necesidades de los consumidores.”¹³

“La logística es aquella parte de la gestión de la Cadena de Abastecimientos que planifica, implementa y controla el flujo -hacia atrás y adelante- y el almacenamiento eficaz y eficiente de los bienes, servicios e información relacionada desde el punto de origen al punto de consumo con el objetivo de satisfacer los requerimientos de los consumidores.”¹⁴

Es muy difícil completar las áreas de marketing o manufactura sin una herramienta logística. Esta involucra la integración de la información, transporte, inventario, almacenaje, manejo de materiales y embalaje. La responsabilidad de la operación logística es la reubicación geográfica de las materias primas, trabajar en procesos e inventarios de acabado cuando sea necesario, al menor costo posible.

El WMS ayudará a la racionalización de muchos de los procesos y al ciclo de productividad a que se maneje más fácil y eficiente. La meta general del software logístico es aumentar márgenes de ganancia y reducir los tiempos para dar al negocio una ventaja competitiva en el mercado. Además integrar soluciones logísticas que ayudarán a

¹² Ballou, Ronald H. (1999). Business Logistics Management, Cuarta edición edición, Upper Saddle River (Nueva Jersey, EE.UU.): Prentice-Hall International, Inc

¹³ D.M. Lambert, M. Cooper, J. D. Pagh (1998) Supply Chain Management: Implementation Issues and Research Opportunities, The International Journal of Logistics Management. *Vol.9, n° 2, pp 1-19*

¹⁴ Council of Supply Chain Management Professionals, CSCMP

sincronizar la cadena de suministro, ahorrando costos, eliminando ineficiencias e incrementando los ingresos. Con toda esta solución se podrán asegurar órdenes más precisas, inventarios al 100% de exactitud y mejores entregas.

La implementación exitosa de este software llevará a un incremento de la productividad, incrementará la colaboración entre áreas, bajará el tiempo de ingreso/egreso y aumentará la seguridad de la información.

El WMS ofrecerá las siguientes facilidades:

- Aplicaciones de escritorio interactivas
- Colaboración
- Facilidad de adaptación al software
- Reducción de entrenamiento del personal
- Interfaces Windows.
- Bases de datos unificadas.
- Sistema sectorizado

Este WMS será diseñado bajo un marco teórico basado en el MSF y desarrollado en la plataforma Windows Visual Studio 2008. Estas dos herramientas dan al desarrollador una rica contribución al software ya que permiten una integración total con la mayoría de procesos actuales en la compañía y una alta compatibilidad con las bases de datos actualmente utilizadas por la empresa. Esta aplicación de desarrollo toma ventaja de la infraestructura y de las habilidades del personal que actualmente ya ejercen los procesos de almacenamiento durante un periodo largo de tiempo.

METAS

Un sistema WMS es una parte fundamental de los procesos de la cadena de abastecimiento y una de las principales metas de este sistema será el movimiento y almacenaje de materiales dentro del almacén y los procesos que son asociados a estas transacciones, incluyendo pedidos, ingresos, putaway y picking. El sistema también dirigirá y optimizará el stock en putway basado en información en tiempo real sobre el estado y su utilización.

El *warehouse management system* proveerá un conjunto de procesos computarizados para manejar el ingreso de inventario como los pedidos dentro del almacén, modelar y manejar la representación lógica del espacio físico dentro de las bodegas, administrar el inventario dentro de los almacenes para permitir un nexo perfecto para ordenar el procesamiento y gestión de la logística, con el fin de: recoger, empaçar y enviar los productos fuera de los almacenes.

Metas	Prioridad
WMS Área de servicio	Alta
Proveer un ambiente colaborativo entre empleados, la información necesaria estará almacenada y actualizada en tiempo real.	
Propagar Información a todos los usuarios del WMS	
Proveer visibilidad a los Directores y Gerentes de área para analizar la información dentro de sus bodegas así tomar decisiones se volverá mucho más fácil y eficiente.	

WMS Ingreso de Mercadería**Alta**

Facilitar a los empleados el ingreso de mercadería proveyendo una interfaz amigable y bien informativa

Proveer de procesos claros los cuales guiarán al empleado a un ingreso exitoso

Producir menos información detallada al empleado pero ingresar mayor información automática en la base datos

Crear varios controles de ingreso para mejorar y asegurar la información en el sistema

WMS Control de Bodega**Alto**

Interfaces comunes para una proliferación de ambientes

Crear accesos para futuras aplicaciones web de consultas

Proveer información dinámica sobre el estado de cada posición del almacén para crear un servicio más eficiente para la empresa

Crear controles de stock e inventario

WMS Picking**Alto**

Crear secuencias de rutas más eficientes dándole la oportunidad a los empleados de aumentar su eficiencia y disminuirles trabajo

Proveer una Interfaz dinámica e interactiva con el empleado al momento de buscar productos

Crear controles de reserva de mercadería para eliminar sobre pedidos a los mismos productos

Crear los picking automáticamente eliminando el registro manual y futuras verificaciones

Crear automáticamente los documentos de egreso facilitando la comunicación entre área y haciendo más eficiente la comunicación y envío de documentación.

WMS Entregas**Medio**

Proveer de información actual y veraz, mantener el inventario al día y actualizado con todos los cambios efectuados

Mantener siempre un registro de movimientos dentro y fuera de la bodega en el kardex

Enviar información esencial a distintos servicios externos como consultas y administración general.

Permitir colaboración y comunicación entre departamentos aumentando la eficiencia en las diferentes transacciones.

OBJETIVOS:

Objetivo General

Diseñar bajo la metodología MSF (Microsoft Solution Framework) y desarrollar bajo la tecnología Microsoft .NET C# un sistema WMS (Warehouse Management System) para ingresar, administrar y almacenar la información de los procesos operativos dentro de las bodegas en la empresa Logistecsa.

Objetivos Específicos

- Diseñar y crear una nueva línea de operaciones dentro de las bodegas basada en los procesos usualmente implementados.
- Desarrollar un nuevo proceso de recibo de inventario y generación de reportes para facilitar las operaciones y reducir el papeleo físico.
- Crear una interfaz gráfica amigable con el usuario y tatar de que esta controle la mayoría de errores producidos por el personal
- Publicar la menor cantidad de información detallada al usuario e ingresar a la base de datos la mayor cantidad de información automática
- Limitar el acceso de información a usuarios de bajo nivel para reducir información errónea dentro de la base de datos.
- Implementar en el nuevo sistema una conexión con los sistemas externos de los clientes por el cual, el recibo de inventario sea de forma automática por medio de un

archivo plano, así se evita el mínimo contacto humano con la información y el almacenaje en bodegas del inventario, conservando su integridad y seguridad.

- Crear un módulo de intercambio de ubicación que permita registrar tanto las posiciones actuales como las futuras, haciendo que el movimiento de pallets se registre y organice de la forma más eficiente, ahorrando tiempo y aumentando la productividad.
- Elaborar un módulo de reserva de inventario para todo lo que es el área de picking, el cual tendrá como finalidad mantener íntegro el inventario y efectivizar el despacho de inventario.
- Crear un espacio donde se pueda ingresar los productos a acondicionamiento sacándolos del inventario general y luego volviéndolos a ingresar según el tipo de producto que regrese.

Supuestos

En el desarrollo de software, generalmente crea supuestos en la interpretación de requerimientos del sistema, decisiones de diseño, dominio operacional, ambiente, características de entrada y salida de información y otros factores durante la implementación del software. Estos supuestos suelen ser rara vez documentados y mucho menos validados por las personas que tiene el conocimiento para verificar su adecuación.

Adicionalmente, el negocio, la parte legal y el ambiente operativo siempre van a estar cambiando, como también el software, dejando así sin validez los supuestos anteriormente validados.

Los estados de supuestos que los Warehouse Management Systems deben permitir un completo control sobre los recursos de almacén de productos terminados, al mismo tiempo incrementar la eficiencia en el rastreo del producto dentro de las bodegas.

3.1.2 Casos de Usos

Análisis de Uso

- Ingresar al sistema como usuario Auxiliar
- Autenticar su usuario y cambiar información básica de su perfil
- Ingresar en la plataforma a la bodega del cliente designado
- Verificar y consultar información sobre pedidos (historial de productos despachados)
- Consultar posiciones de productos
- Consultar pickings y productos reservados
- Verificar estados

Usuario 1 (Gerencia): Este tipo de usuario será muy parecido al Usuario 0 ya que la plataforma estará inhabilitada casi en su totalidad permitiéndole a este usuario hacer las siguientes operaciones de consultas:

- Ingresar al sistema como usuario Gerente
- Autenticar su usuario y cambiar información básica de su perfil
- Ingresar a la plataforma al área de todos los clientes en el sistema
- Verificar y consultar información sobre pedidos
- Consultar posiciones de productos
- Consultar pickings y productos reservados

- Verificar estados
- Crear análisis financieros y estimaciones de facturación
- Verificar documentación y comunicación entre las áreas

Usuario 2 (Supervisor): Este usuario es el que proveerá y cargará información al sistema, tendrá un acceso casi total a la plataforma con ciertas restricciones de seguridad. Este usuario podrá efectuar cambios y modificaciones a la información del sistema. Las siguientes operaciones estarán habilitadas para este usuario:

- Autenticar su usuario y cambiar información básica de su perfil
- Ingresar a la plataforma del cliente designado
- Registrar en el sistema los Ingresos
- Crear productos nuevos en el sistema
- Eliminar productos en el sistema
- Cargar archivos planos de los clientes
- Crear y enviar documentos de ingreso
- Registrar y Ordenar los productos en distintas posiciones
- Enviar productos a Acondicionamiento
- Cambiar de estado a los productos
- Ingresar Pedidos
- Crear lista de Picking
- Modificar lista de Picking
- Crear lista de Incineración
- Modificación de lista de Incineración
- Creación y Envío de documento de entrega
- Verificar y consultar información sobre pedidos

- Consultar posiciones de productos
- Consultar pickings y productos reservados
- Verificar estados

Usuario 3 (Jefe de Bodegas): El Usuario 2 tendrá la capacidad y el control total de la plataforma creando permisos para los diferentes trabajadores. Además, administrará y designará los espacios correspondientes para cada uno de los productos de los clientes.

- Autenticar su usuario y cambiar información básica de su perfil
- Ingresar a la plataforma de todos los clientes
- Registrar en el sistema los Ingresos
- Crear productos nuevos en el sistema
- Eliminar productos en el sistema
- Cargar archivos planos de los clientes
- Crear y enviar documentos de ingreso
- Registrar y Ordenar los productos en distintas posiciones
- Registrar las diferentes posiciones para un determinado cliente
- Crear permisos para los diferentes trabajadores
- Enviar productos a Acondicionamiento
- Cambiar de estado a los productos
- Ingresar Pedidos
- Crear lista de Picking
- Modificar lista de Picking
- Crear lista de Incineración
- Modificación de lista de Incineración
- Creación y Envío de documento de entrega

- Verificar y consultar información sobre pedidos
- Consultar posiciones de productos
- Consultar pickings y productos reservados
- Verificar estados
- Crear análisis financieros y estimaciones de facturación
- Verificar documentación y comunicación entre las áreas

Perfiles de Usuarios

Usuario 0 (Auxiliar) : Este usuario tiene la posibilidad de hacer una serie de consultas dentro de la plataforma, como la verificación de ordenes de despacho, estado de los productos, posición de los productos, envíos, devoluciones, etc. La plataforma le brindará un escenario de consulta muy dinámico y eficiente para reducir el tiempo de consulta y entregar la mayor cantidad posible de información veraz y en tiempo real sobre todas las actividades involucradas dentro de los almacenes.

Un punto muy importante es que el usuario solo podrá verificar y hacer consultas de los productos del cliente que esté designado, así se involucrará con el área que le corresponde, manteniendo una confidencialidad y un buen desarrollo de la información.

Usuario 1 (Gerencia) : El usuario 1 es la persona que puede consultar todo relacionado con la plataforma, tiene acceso a la consulta de todas las transacciones creadas por este sistema, además tiene ciertas consultas como costos por posición, costos semanales, costos mensuales y anuales. Además, tiene un control sobre todos los clientes a diferencia del usuario 0 que solo tiene acceso a la información del cliente designado. En esta sección de consultas tendrá un sin número de opciones de consulta para entregar la información más

eficaz y veraz posible, obviamente teniendo en cuenta el factor de tiempo ya que todas las consultas serán con datos a tiempo real.

Usuario 2 (Supervisor): El usuario 1 va a ser el personaje con mayor interacción con el sistema ya que este va ser tanto el controlador como el suministrador de información a la plataforma. Este usuario va a ser el responsable de registrar cada movimiento de mercadería que se haga dentro del almacén.

Al ingresar productos tendrá la posibilidad de selección los productos de una lista existente, o crearlo en el caso de que no existan, en la base de datos para una futura reutilización. También el usuario podrá cambiar de lugar la mercadería para facilitar el proceso de picking a su gusto y esto se registrará automáticamente en el sistema de kardex que se maneja paralelamente a la plataforma a utilizarse.

El usuario 1 tendrá restricciones únicamente en el acceso a la mercadería de diferentes clientes. Cada supervisor podrá trabajar y hacer los cambios deseados sobre los productos del cliente asignado para que no haya problemas entre las distintas áreas. Además, tendrá una libertad de posiciones dentro del almacén restringida al área designada por el jefe de bodegas, es decir que el jefe de bodegas designara ciertos racks y posiciones dentro del almacén para ser utilizados por cada supervisor, dentro de esta área determinada el supervisor podrá hacer los cambios que necesite sin ninguna restricción.

Usuario 3 (Jefe de Bodegas): El jefe de bodegas tendrá todos los permisos del Usuario 2 (Supervisor) con ciertos permisos adicionales. Dentro de las actividades del jefe de bodega existe la función de asignar varios espacios dentro del almacén a diferentes cliente y a la vez a cada cliente se le asigna un supervisor y un grupo de auxiliares.

La asignación de personal a cada área de la bodega es responsabilidad del jefe de bodegas ya que en este procesos se limitará el acceso al personal solo al área correspondiente.

También tendrá a cargo la creación de perfiles de usuarios en el sistema, ya que este proceso es muy delicado y tiene que ser supervisado por el jefe de bodegas. El usuario tendrá la posibilidad de darle el nivel de acceso al usuario que quiera y piense que sea necesario, así que este proceso queda totalmente dispuesto al criterio del usuario. Además el usuario podrá remplazar usuarios o eliminarlos del sistema por conveniencia de los procesos actualmente existentes, ya que se presenta una gran rotación de personal temporal dentro de la bodega.

Escenarios de Usos

En esta sección se describirá cada caso de uso del sistema, cómo este se relaciona con los distintos actores y sus relaciones especificadas.

Caso de Uso 1: Inicialización de Usuarios

Descripción	El caso de Inicialización de usuarios es el primer módulo del sistema, al cual todos los usuarios tienen que pasar para determinar su nivel de injerencia en el sistema.
Identificador de Escenario	U1
Fecha	01/10/09
Revisado	Andrés Marchant C
Actores	Usuario 0, Usuario 1, Usuario 2, Usuario 3
Pre-condiciones	No hay condiciones previas.
Acciones	<ul style="list-style-type: none">• Visualización de la pantalla “Login”• Ingresar los campos de usuario y clave• Validar usuario y contraseña• Ingresar al sistema con los permisos correspondientes activos
Post-condiciones	<ul style="list-style-type: none">• El usuario y la contraseña deben corresponder con el perfil de usuario
Includes	Ninguno
Extends	Consultas, Ingresos, Despachos y Administración

Caso de Uso 2: Consultas Productos

Descripción	En este caso se harán distintos tipos de consultas al sistema sobre los productos, en sus estados, posiciones, fechas de caducidad, lotes, etc.
Identificador de Escenario	C1
Fecha	01/10/09
Revisado	Andrés Marchant C, Ing. Alexander Verdesoto
Actores	Usuario 0, Usuario 1 ,Usuario 2, Usuario 3
Pre-condiciones	Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta.
Acciones	<ul style="list-style-type: none"> • Ingresar al menú de opciones de Consulta • Seleccionar Consultas de Productos • Consultar los diferentes estados de un producto • Consultar el lugar específico de un producto para el posterior rastreo • Consultar la disponibilidad de un producto y su cantidad • Consultar su lote • Consultar fecha de expiración de un producto • Consultar el cliente de ese producto (Usuario 1, Usuario 3) • Consultar costes de permanencia en bodega (Usuario 1, Usuario 3)
Post-condiciones	Ninguna
Includes	Caso de Uso 1: Identificación de Usuarios
Extends	Ninguno

Caso de Uso 3: Consulta Personal

Descripción	Este módulo será una herramienta de administración tanto de recursos humanos como de logística, permitiendo a diferentes encargados revisar y mantenerse al día mediante un kardex eficiente y a tiempo real.
Identificador de Escenario	C2
Fecha	01/10/09
Revisado	Andrés Marchant C
Actores	Usuario 1, Usuario 2 ,Usuario 3
Pre-condiciones	Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta.
Acciones	<ul style="list-style-type: none"> • Ingresar al menú de opciones de Consulta • Seleccionar Consultas de Personal • Consultar el Perfil de un Usuario • Consultar encargados de un o unos productos determinados • Consultar responsable de movimientos • Consultar responsables de Ingresos • Consultar responsables de Egresos • Consultar autores de pickings • Consultar encargados de reservas • Consultar responsables de movimientos existentes de un o unos productos • Consultar historiales personales
Post-condiciones	<ul style="list-style-type: none"> • Ninguno
Includes	Ninguno
Extends	Ninguno

Caso de Uso 4: Consulta de Espacio

Descripción	Este módulo será una herramienta de administración de los espacios existentes en las bodegas. Podremos ver de una manera dinámica los espacios existentes, tanto ocupados como libres, y hacer los análisis necesarios.
Identificador de Escenario	C3
Fecha	01/10/09
Revisado	Andrés Marchant C, Ing. Alexander Verdesoto
Actores	Usuario 1, Usuario 2 ,Usuario 3
Pre-condiciones	Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta.
Acciones	<ul style="list-style-type: none">• Ingresar al menú de opciones de Consulta• Seleccionar Consultas de Espacio• Consultar un o unos espacios determinados• Consultar espacios utilizados• Consultar espacios libres• Consultar utilización por Cliente (Usuario 1 y Usuario 3)• Consultar historial de los espacios
Post-condiciones	<ul style="list-style-type: none">• Ninguno
Includes	Ninguno
Extends	Caso de Uso 1

Caso de Uso 5: Ingresos Importaciones/Transferencias

Descripción	Este módulo de ingresos es uno de los más importantes ya que proveerá la mayoría de información del producto
Identificador de Escenario	I1
Fecha	01/10/09
Revisado	Andrés Marchant C , Ing. Alexander Verdesoto
Actores	Usuario 2, Usuario 3
Pre-condiciones	Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 2 o 3.
Acciones	<ul style="list-style-type: none">• Ingresar al sistema con los permisos correspondientes activos• Ingresar al módulo activo de Ingresos de Importaciones/Transferencias• Seleccionar las características del producto• Llenar los campos requeridos como cantidad, fecha caducidad, lote , posición a ocupar, etc.• Guardar ingreso
Post-condiciones	La información debe ser real y veraz
Includes	Caso de Uso 1: Identificación de Usuarios
Extends	Ninguno

Caso de Uso 6: Modificar Importaciones/Transferencias

Descripción	Este módulo permite modificar ingresos anteriormente creados
Identificador de Escenario	I2
Fecha	01/10/09
Revisado	Andrés Marchant C, Ing. Alexander Verdesoto
Actores	Usuario 2, Usuario 3
Pre-condiciones	Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 2 o 3. Además el sistema tiene que tener creado ingresos anteriormente
Acciones	<ul style="list-style-type: none">• Ingresar al sistema con los permisos correspondientes activos• Ingresar al módulo activo de Ingresos Modificar Importaciones/Transferencias• Seleccionar el ingreso a modificar• Llenar los campos requeridos como cantidad, fecha caducidad, lote, posición a ocupar, etc.• Validar que no existe otro ingreso similar en el sistema• Guardar ingreso
Post-condiciones	<ul style="list-style-type: none">• La información debe ser real y veraz• Debe existir un ingreso para poder modificarlo
Includes	Caso de Uso 4: Ingresos
Extends	Caso de Uso 5: Ingresos Importaciones/Transferencias

Caso de Uso 7: Eliminar Importaciones/Transferencias

Descripción	Este módulo permite eliminar ingresos actuales creados. Permitiendo así corregir errores humanos.
Identificador de Escenario	I3
Fecha	01/10/09
Revisado	Andrés Marchant C, Ing. Alexander Verdesoto
Actores	Usuario 2, Usuario 3
Pre-condiciones	Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 2 o 3. Además el sistema tiene que tener creado ingresos actuales y sin ser usados para otras actividades
Acciones	<ul style="list-style-type: none"> • Ingresar al sistema con los permisos correspondientes activos • Ingresar al módulo activo de Ingresos Eliminar Importaciones/Transferencias • Seleccionar el ingreso a Eliminar • Validar que no se uso este ingreso en el sistema • Guardar cambio
Post-condiciones	Ninguna
Includes	Caso de Uso 9: Administrar Productos Crear
Extends	Caso de Uso 5: Ingresos Importaciones/Transferencias

Caso de Uso 8: Ingresos Importaciones/Transferencias (Automática)

Descripción	Este módulo de ingresos es muy parecido al I1 a diferencia que todo el proceso se lo hace automáticamente desde un archivo plano enviado por los clientes con toda la información requerida.
Identificador de Escenario	I4
Fecha	01/10/09
Revisado	Andrés Marchant C, Ing. Alexander Verdesoto
Actores	Usuario 2, Usuario 3
Pre-condiciones	<ul style="list-style-type: none"> • Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 2 o 3. • El archivo plano debe tener la estructura determinada
Acciones	<ul style="list-style-type: none"> • Ingresar al sistema con los permisos correspondientes activos • Ingresar al módulo activo de Ingresos de Importaciones/Transferencias • Activar el ingreso automático • Cargar Archivo • Validar Información • Guardar ingreso
Post-condiciones	<ul style="list-style-type: none"> • La información debe ser real y veraz
Includes	Caso de Uso 1: Identificación de Usuarios Caso de Uso 6: Ingresos Importaciones/Transferencias
Extends	Ninguno

Caso de Uso 9: Administrar Productos Crear

Descripción	En este caso permitirá al usuario crear productos nuevos que no estén en la base de datos, permitiendo así una expansión del stock actual
Identificador de Escenario	P1
Fecha	01/10/09
Revisado	Andrés Marchant C, Ing. Alexander Verdesoto
Actores	Usuario 2, Usuario 3
Pre-condiciones	Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 2 o 3 e ingresar por el módulo de Administración de Productos.
Acciones	<ul style="list-style-type: none">• Ingresar al sistema con los permisos correspondientes activos• Ingresar al módulo activo de Productos• Seleccionar Crear Producto Nuevo• Seleccionar las características del producto• Llenar los campos requeridos• Validar que no exista ya en la base de datos• Guardar Producto
Post-condiciones	<ul style="list-style-type: none">• No debe existir un producto similar
Includes	Caso de Uso 6: Crear Productos
Extends	Ninguna

Caso de Uso 10: Administrar Productos Modificar

Descripción	En este caso permitirá al usuario modificar productos que estén en la base de datos, permitiendo así una corrección del inventario.
Identificador de Escenario	P2
Fecha	01/10/09
Revisado	Andrés Marchant C, Ing. Alexander Verdesoto
Actores	Usuario 2, Usuario 3
Pre-condiciones	Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 2 o 3 e ingresar por el módulo de Administración de Productos.
Acciones	<ul style="list-style-type: none">• Ingresar al sistema con los permisos correspondientes activos• Ingresar al módulo activo de Productos• Seleccionar Modificar Producto• Seleccionar el producto a cambiar.• Llenar los campos requeridos• Validar que no exista ya en la base de datos• Guardar Producto
Post-condiciones	<ul style="list-style-type: none">• No debe existir un producto similar
Includes	Ninguno
Extends	Caso de Uso 8: Administrar Productos Crear

Caso de Uso 11: Administrar Productos Eliminar

Descripción	Este módulo permite eliminar productos actualmente creados. Permitiendo así corregir errores humanos.
Identificador de Escenario	P3
Fecha	01/10/09
Revisado	Andrés Marchant C, Ing. Alexander Verdesoto
Actores	Usuario 2, Usuario 3
Pre-condiciones	Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 2 o 3. Además el sistema tiene que tener creados productos actuales y sin ser usados para otras actividades en la plataforma.
Acciones	<ul style="list-style-type: none"> • Ingresar al sistema con los permisos correspondientes activos • Ingresar al módulo activo de Administrar Productos • Seleccionar el ingreso a Eliminar • Validar que no se usó este ingreso en el sistema • Guardar cambio
Post-condiciones	Ninguna
Includes	Caso de Uso 8: Administrar Productos Crear
Extends	Caso de Uso 5: Ingresos Importaciones/Transferencias

Caso de Uso 12: Administrar Empleados Crear

Descripción	En este caso permitirá al Usuario 3 crear un perfil de usuario, el cual podrá darle los permisos necesarios para su normal desempeño a las diferentes etapas del sistema.
Identificador de Escenario	E1
Fecha	01/10/09
Revisado	Andrés Marchant C, Ing. Alexander Verdesoto
Actores	Usuario 3
Pre-condiciones	Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 3 e ingresar por el módulo de Administración de Personal.
Acciones	<ul style="list-style-type: none">• Ingresar al sistema con los permisos correspondientes activos• Ingresar al módulo activo de Administración de Personal• Seleccionar Crear Usuario Nuevo• Seleccionar las características del Usuario• Llenar los campos requeridos• Validar que no exista ya en la base de datos• Guardar Producto
Post-condiciones	<ul style="list-style-type: none">• No debe existir un producto similar
Includes	Caso de Uso 1: Identificación de Usuarios
Extends	Ninguna

Caso de Uso 12: Administrar Empleados Modificar

Descripción	En este caso permitirá al Usuario 3 modificar empleados que estén en la base de datos, permitiendo así una corrección del personal.
Identificador de Escenario	E2
Fecha	01/10/09
Revisado	Andrés Marchant C, Ing. Alexander Verdesoto
Actores	Usuario 3
Pre-condiciones	Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 3 e ingresar por el módulo de Administración de Personal.
Acciones	<ul style="list-style-type: none">• Ingresar al sistema con los permisos correspondientes activos• Ingresar al módulo activo de Administración de Personal• Seleccionar Modificar Personal• Seleccionar el empleado a cambiar.• Llenar los campos requeridos• Validar que no exista uno similar ya en la base de datos• Guardar Empleado
Post-condiciones	<ul style="list-style-type: none">• No debe duplicar el personal• No debe tener permisos que no le corresponde
Includes	Ninguno
Extends	Caso de Uso 11: Administrar Empleados Crear

Caso de Uso 13: Administrar Empleados Eliminar

Descripción	Este módulo permite eliminar personal actualmente creados. Permitiendo así corregir errores humanos.
Identificador de Escenario	P3
Fecha	01/10/09
Revisado	Andrés Marchant C, Ing. Alexander Verdesoto
Actores	Usuario 3
Pre- condiciones	Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 3. Además el sistema tiene que tener empleados creados actualmente y sin ser usados por otras actividades en la plataforma.
Acciones	<ul style="list-style-type: none">• Ingresar al sistema con los permisos correspondientes activos• Ingresar al módulo activo de Administrar Personal• Seleccionar el ingreso a Eliminar Personal• Validar que no se usó este ingreso en el sistema• Guardar cambio
Post- condiciones	Ninguna
Includes	Caso de Uso 11: Administrar Empleados Crear
Extends	Ninguno

Caso de Uso 14: Cambio de Ubicación

Descripción	Este es un pequeño módulo en el cual permite a los Usuarios 2 y 3 hacer cambio de posiciones de los productos para hacer más eficiente la distribución de espacio dentro del almacén.
Identificador de Escenario	U1
Fecha	02/10/09
Revisado	Andrés Marchant C, Ing. Alexander Verdesoto
Actores	Usuario 2, Usuario 3
Pre-condiciones	<ul style="list-style-type: none"> • Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 2 o 3. • Debe existir los productos activos y disponibles en el inventario • Las nuevas posiciones deben estar libres para hacer las transacciones.
Acciones	<ul style="list-style-type: none"> • Ingresar al sistema con los permisos correspondientes activos • Ingresar al módulo de Movimiento • Seleccionar los productos a cambiar • Llenar los campos requeridos como nueva posición, etc. • Guardar cambios
Post-condiciones	Ninguna
Includes	Caso de Uso 1: Identificación de Usuarios
Extends	Ninguno

Caso de Uso 15: Reserva Picking

Descripción	Este caso se usa para reservar los productos de los pedidos que llegan, permitiendo al trabajador reservar en el inventario y hacer el picking de los productos más tranquilo y eficiente.
Identificador de Escenario	P1
Fecha	02/10/09
Revisado	Andrés Marchant C
Actores	Usuario 2, Usuario 3
Pre-condiciones	<ul style="list-style-type: none"> • Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 2 o 3. • Debe existir los productos activos y disponibles en el inventario
Acciones	<ul style="list-style-type: none"> • Ingresar al sistema con los permisos correspondientes activos • Ingresar al módulo de Picking • Seleccionar Reserva • Seleccionar todos los productos a reservar con sus cantidades respectivas • Seleccionar el lote (se mostraran por orden de caducidad) • Reservar Productos
Post-condiciones	Se creará una lista de picking con la mejor ruta posible
Includes	Caso de Uso 1: Identificación de Usuarios
Extends	Caso de Uso N: Picking

Caso de Uso 16: Reserva Picking Eliminar

Descripción	Este caso se usa para eliminar las reservas de los productos de los pedidos que llegan, permitiendo al trabajador activar los productos anteriormente reservados en el inventario.
Identificador de Escenario	P2
Fecha	02/10/09
Revisado	Andrés Marchant C
Actores	Usuario 2, Usuario 3
Pre-condiciones	<ul style="list-style-type: none"> • Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 2 o 3. • Debe existir los productos reservados y en reserva de picking
Acciones	<ul style="list-style-type: none"> • Ingresar al sistema con los permisos correspondientes activos • Ingresar al módulo de Picking • Seleccionar eliminar Reserva • Seleccionar la reserva picking a eliminar • Eliminar
Post-condiciones	Ninguna
Includes	Caso de Uso 1: Identificación de Usuarios Caso de Uso 16: Reserva picking
Extends	Ninguno

Caso de Uso 17: Picking

Descripción	Este módulo permitirá al usuario chequear de nuevo los productos seleccionados y aceptar la lista de reserva de picking convirtiéndose en el documento de entrega para la empresa.
Identificador de Escenario	P3
Fecha	02/10/09
Revisado	Andrés Marchant C
Actores	Usuario 2, Usuario 3
Pre-condiciones	<ul style="list-style-type: none">• Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 2 o 3.• Debe existir los productos reservados y listos para picking
Acciones	<ul style="list-style-type: none">• Ingresar al sistema con los permisos correspondientes activos• Ingresar al módulo de Picking• Seleccionar Reserva (en el caso de cerrar cesión y reiniciar procesos)• Verificar todos los productos reservados con sus cantidades respectivas• Seleccionar Despachar
Post-condiciones	Se crea el documento de entrega y se guarda en la base de datos para posteriores condiciones
Includes	Caso de Uso 1: Identificación de Usuarios
Extends	Caso de Uso 16: Reserva Picking

Caso de Uso 18: Eliminar Picking

Descripción	Este módulo permitirá al usuario eliminar un picking anteriormente creado, devolviendo todos los productos al inventario.
Identificador de Escenario	P4
Fecha	02/10/09
Revisado	Andrés Marchant C, Ing. Alexander Verdesoto
Actores	Usuario 2, Usuario 3
Pre-condiciones	<ul style="list-style-type: none"> • Para que exista este estado tiene que pasar primero por el módulo de identificación de usuario para determinar el alcance de su cuenta e identificarse como Usuario 2 o 3. • Debe existir el picking listo para despachar • No debe haber ningún picking posterior ni despachado
Acciones	<ul style="list-style-type: none"> • Ingresar al sistema con los permisos correspondientes activos • Ingresar al módulo de Picking • Seleccionar Eliminar • Buscar el picking a eliminar • Seleccionar Eliminar
Post-condiciones	Se anula el documento de entrega pero se conserva en el sistema Se ingresa de nuevo los productos al sistema
Includes	Caso de Uso 1: Identificación de Usuarios Caso de Uso 5: Ingresos Importaciones/Transferencias
Extends	Caso de Uso 18: Picking

3.1.3 Requerimientos

“En el proceso conceptualizar los análisis de requerimientos incluye tres tipos de actividades:

Conseguir los requisitos: la tarea de comunicación con los clientes y usuarios para determinar cuáles son sus requerimientos. Esto en ciertas ocasiones se llama recolección de requerimientos.

- Análisis de requerimientos: determinar si los requerimientos son poco claros, incompletos, ambiguos o contradictorios, y luego resolver esos problemas.
- Registro de requerimientos: Los requerimientos pueden ser documentados en varios formatos como el documento de lenguaje natural, caso de usos, historias de usuarios o procesos de especificación.

El análisis de requerimientos puede ser un largo y duro proceso durante el cual muchas y delicadas habilidades psicológicas están involucradas. Un nuevo sistema cambia el ambiente y la relación entre las personas, por eso es importante identificar todos los usuarios, tomar sus necesidades en cuenta y asegurar que entiendan las implicaciones de este nuevo sistema. El analista puede utilizar varias técnicas para obtener los requerimientos de los clientes. Históricamente, esto ha incluido mantener entrevistas, grupos de trabajo y crear listas de requerimientos. Técnicas más modernas incluyen prototipos y utilización de casos de uso.

Donde sea necesario, el analista empleará una combinación de estos tres métodos para establecer los requerimientos exactos de los usuarios y accionistas, para que el sistema

llegue a cumplir las necesidades del negocio. La ingeniería de requerimientos es una sub disciplina de la ingeniería de sistemas e ingeniería de software, que se refiere a la determinación de los objetivos, funciones y a las limitaciones de hardware y software de sistemas. En algunos modelos de ciclo de vida, el proceso de requerimientos empieza con el estudio de factibilidad, el cual conlleva a un reporte de factibilidad. Si este estudio de factibilidad sugiere que el producto debe ser desarrollado, los análisis de requerimientos pueden empezar.”¹⁵

3.1.3.1 Requerimientos de estándares de Sistema

Clientes

SW existente

- Windows XP Professional Service Pack 2 o superior
- .Net Framework 3.5
- Internet Explorer 6.0 o Superior o Mozilla 2.0.
- Office 2003 con (*Microsoft Office Compatibility Pack for Word, Excel, and PowerPoint 2007 file formats*). O Superior
- Resolución de pantalla 1024 x 768

Servidor de Aplicaciones

SW existente

- Windows Server 2008 R2
- Internet Information Services 6.0
- Microsoft .Net Framework 2.0

¹⁵ Phillip A. Laplante (2007) What Every Engineer Should Know about Software Engineering. Taylor & Francis Group, Boca Raton, FL, USA

- Microsoft .Net Framework 3.0
- Microsoft .Net Framework 3.5
- Microsoft .net framework 2.0 Service Pack 2
- Microsoft .net framework 3.0 Service Pack 2
- Microsoft .net framework 3.5 (kb953595)
- Microsoft .net framework 3.5 SP1

Servidor de base de datos

SW existente

- SQLServer 2005 Enterprise editionSP2 Español

Hardware existente

LOGI12

Información General

- Memoria RAM 8GB
- 8 Procesadores

Unidades de Disco

- C: Total Espacio 80GB , Espacio Libre 50.6GB
- D: Total Espacio 56.7GB, Espacio Libre 23.6GB

LOGI13

Información General

- Memoria RAM 8GB
- 4 Procesadores

Unidades de Disco

- C: Total Espacio 33.9GB , Espacio Libre 20.9GB
- D: Total Espacio 67.8GB, Espacio Libre 65.6GB

3.1.3.2 Requerimientos de Negocio

Introducción

En esta sección se describirán los requerimientos para el diseño y el desarrollo del Warehouse Management System para la empresa Logistecsa.

El documento de requerimientos intenta informar cómo se va a planificar el diseño y el desarrollo en el período determinado, varios componentes de esta plataforma de servicio serán implementados por primera vez en este ambiente de trabajo como es el de la empresa Logistecsa.

Escalabilidad para Necesidades Empresariales

“La escalabilidad debe formar parte del proceso de diseño porque no es una característica separada que se pueda agregar después. Al igual que con otras funciones de aplicación, las decisiones que se tomen durante las primeras fases de diseño y codificación determinarán en gran medida la escalabilidad de la aplicación.

La escalabilidad de una aplicación requiere una pertenencia equilibrada entre dos dominios distintos, software y hardware. Puede avanzar grandes pasos que aumenten la escalabilidad de un dominio sólo para sabotarlos cometiendo errores en el otro. Por ejemplo, la creación de un grupo de servidores Web con equilibrio de carga no beneficiará

una aplicación Web que se ha diseñado para ejecutarse un solo equipo. De igual modo, el diseño de una aplicación altamente escalable y su implementación en equipos conectados a una red con poco ancho de banda no controlará bien las cargas pesadas cuando se sature el tráfico en la red”.

Puesto que la escalabilidad no es un problema de diseño de las aplicaciones independientes, aquí se tratan las aplicaciones distribuidas. Las aplicaciones distribuidas están también un paso más allá de las tradicionales aplicaciones de cliente-servidor. Las aplicaciones distribuidas son aplicaciones que están diseñadas como aplicaciones de n niveles. La arquitectura de estas aplicaciones distribuidas favorece el diseño de aplicaciones escalables compartiendo recursos, como bases de datos y componentes empresariales.

Escalar en vertical

El escalado en vertical es el término que más se utiliza para lograr escalabilidad utilizando software mejor, más rápido y más caro. El escalado incluye agregar más memoria, procesadores más rápidos o, simplemente, migrar la aplicación a un único equipo más potente. Normalmente, este método permite un aumento en la capacidad sin requerir cambios en el código fuente. Desde el punto de vista administrativo, las cosas permanecen igual puesto que sigue habiendo un único equipo que administrar. Como se observa en la figura 3.1 se aprecia el ciclo de escalabilidad.

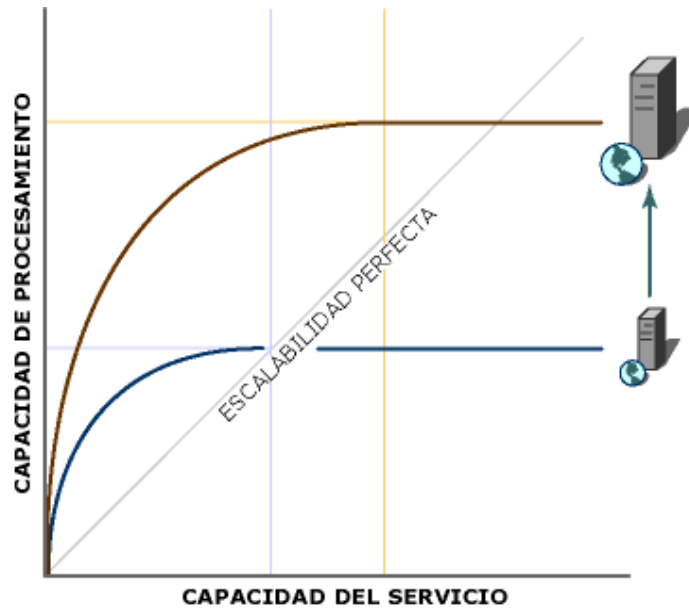


FIGURA N° 3.1: Escalado en vertical¹⁶.

Escalar en horizontal

Otra alternativa es escalar en horizontal. Escalar en horizontal aprovecha el ahorro que supone utilizar el hardware de PC activo para distribuir la carga de procesamiento en más de un servidor. Aunque el escalado en horizontal se logra utilizando muchos equipos, la colección funciona esencialmente como un único equipo. Al dedicar varios equipos a una tarea común, mejora la tolerancia de errores de la aplicación. Por supuesto, desde el punto de vista del administrador, escalar en horizontal presenta un desafío mayor de administración debido al mayor número de equipos. »¹⁷ Ver figura 3.2.

¹⁶: [http://i.msdn.microsoft.com/Aa292203.vxscalabilityscaleup\(es-es,VS.71\).gif](http://i.msdn.microsoft.com/Aa292203.vxscalabilityscaleup(es-es,VS.71).gif).

¹⁷ MSDN- (2003) Información general sobre escalabilidad. En: [http://msdn.microsoft.com/es-es/library/aa292203\(VS.71\).aspx](http://msdn.microsoft.com/es-es/library/aa292203(VS.71).aspx). Consultado el 4 de Noviembre del 2009

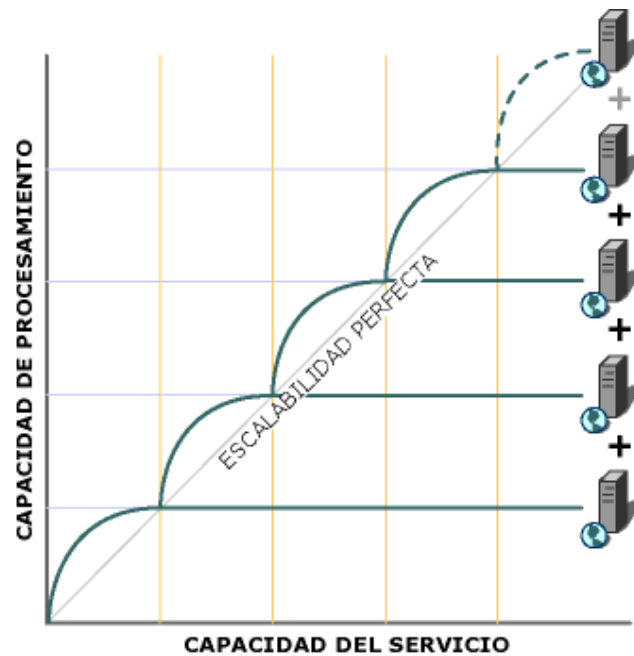


FIGURA N° 3.2: Escalado en horizontal.¹⁸

Escalabilidad es una meta muy importante para muchos proyectos de desarrollo de software porque sin la escalabilidad, el éxito en el proyecto puede verse amenazado por un pobre desempeño al final de este. Es esencial que el crecimiento base del cliente pueda ser manejado por el sistema, además que este aumento de tráfico y el usuario no se encuentre con un rendimiento inaceptable del sistema.

En este proyecto se va a requerir un diseño bien implementado para permitir la futura escalabilidad del sistema para que se acople a nuevas tecnologías y métodos de utilización, para este análisis dividimos en dos factores de escalabilidad, por hardware y software.

¹⁸ Microsoft,2007, Scalabilitys and Scale En : [http://i.msdn.microsoft.com/Aa292203.vxscalabilityscaleup\(es-es.VS.71\).gif](http://i.msdn.microsoft.com/Aa292203.vxscalabilityscaleup(es-es.VS.71).gif).

Escalabilidad de Hardware

Escalabilidad de hardware se refiere primordialmente a la adición de servidores o piezas a un sistema establecido. Añadir servidores a un sistema que está sobrecargado, incrementando sus conexiones o capacidades, puede mejorar significativamente su desempeño. El número de servidores requerido por un sistema está determinado por la tasa de transferencia de la conexión, el tipo de información almacenada y el número de clientes conectados concurrentemente.

Otro tipo de escalabilidad más sencilla requiere una simple actualización de componentes al servidor en uso. Añadiendo ciertas piezas como más memoria RAM o capacidad de almacenamiento, se puede arreglar de una forma económica la mayoría de problemas en el desempeño de la aplicación.

Para la aplicación WMS creada para la empresa Logistecsa se estima que a futuro se necesitará mayor capacidad de almacenamiento en los servidores como requerimiento básico para el buen funcionamiento de la plataforma. Como el sistema maneja mucha información dentro de este mismo no es necesario invertir en conexiones de alta velocidad, además los usuarios conectados actualmente y a futuro no entorpecerán con el buen funcionamiento del sistema y su configuración actual.

El sistema está diseñado de una cierta forma que permitirá la implementación de nueva tecnología en el área de bodegas para hacer más eficiente los procesos de ingresos, almacenaje y egresos.

Dentro de estas tecnologías existe enlaces wireless con dispositivos de lectura de código de barras, etiquetas RIF de radio frecuencia y un sin números de tecnologías disponibles a implementar a esta plataforma, la cual permite una escalabilidad total hacia estas tecnologías permitiendo una plataforma muy estable para la empresa.

Escalabilidad de Software

Con el fin de lograr el éxito en la integridad de la información, las aplicaciones inteligentes deben ser escalables. Las aplicaciones deben funcionar continuamente eficientes a pesar de que se impongan nuevas condiciones como el incremento de volúmenes de información o cambios de requerimientos de usuarios. Para completar el proceso de escalabilidad, ambas infraestructuras: tecnología y arquitectura deben ser consideradas.

Para este proyecto se consideró el diseño y desarrollo de una plataforma de 3 capas para mejorar la eficiencia de los procesos dentro de la estructura, además permitiéndonos entre otras cosas una escalabilidad muy eficiente, ya que la mayoría de componentes están desarrollados por módulos, facilitando así futuros cambios o agregados.

La solución de 3 capas provee un soporte técnico con una reducción de tiempo y costos mucho más eficaz. Además de la usual ventaja del software modular con las interfaces bien definidas, la arquitectura de de 3 capas permite que cualquiera de sus capas puedan ser actualizadas o remplazadas independientemente por un nuevo requerimiento de usuario o tecnológico. Por ejemplo, si se cambia de sistema operativo o se actualiza a una nueva versión, solo cambiaría la parte de presentación, manteniendo a todo el sistema íntegro y funcional.

Seguridad

Mejorar el proceso de desarrollo de software y la creación de software de mejor calidad son formas de mejorar la seguridad del software, principalmente hay que identificar los componentes del software crítico que controlan las funciones principales de seguridad y poner especial atención a estos durante todos los procesos, tanto de desarrollo

como de pruebas. Esta estrategia ayuda a enfocarse hacia los recursos de seguridad en las aéreas más vulnerables.

Sistema de cuentas

Una de las razones por la cual las aplicaciones son violadas, es porque son instaladas con privilegios de cuenta que normalmente un usuario no debería tener. Usualmente los sistemas tienen una serie de privilegios para diferentes usuarios, así como accesos al sistema e ingresos a servidores, llamar a componentes COM+, hacer consultas a registros clasificados, etc.

La manera más común de instalar una aplicación es darle privilegios administrativos, en efecto, cualquier código que corra la aplicación será como administrador. Los usuarios tienen la capacidad de hacer cualquier cosa que quieran con el sistema, sin restricciones. Estas cosas son perfectas para que se produzcan ataques (accidentales o provocados) a la integridad de la información del sistema, produciendo grandes errores en el buen funcionamiento de este, incluso poniendo en peligro hasta la facturación de la empresa, dado que este es uno de los procesos más importante de cualquier negocio.

Cuando se desarrollan aplicaciones y se instalan, es importante considerar los mínimos privilegios que la aplicación debe tener y cómo estos pueden ser minimizados. Uno de los principales desafíos con respecto a minimizar privilegios dentro del sistema es entender cuáles son.

Dentro del WMS de la empresa Logistecsa se va a crear toda una estructura de permisos para los diferentes usuarios de las áreas relacionadas al sistema. Se creará 4 tipos diferentes de usuarios con diferentes alcances. Además, se llevará un registro de todos los movimientos de los usuarios dentro del sistema para mejorar la seguridad y el control.

Ningún usuario tendrá control completo sobre el servidor y la información, asegurando así la máxima seguridad posible.

Creación de Capas

Una de las técnicas de diseño para mejorar la seguridad en el sistema es crear múltiples capas en la arquitectura, para proteger la integridad del sistema. Diseñar aplicaciones desde el principio, entendiendo que hay varias capas entre la aplicación y el usuario, da un acceso a herramientas de infraestructura que ayuda a proteger la aplicación.

Un Gateway aplicativo puede inspeccionar y comprobar cada llamada al sistema y prevenir esos que pueden potencialmente crear problemas. Cuando se empieza a diseñar, se pueden probar ciertas interacciones e identificar las áreas donde el Gateway permita el tráfico de información.

Planificar desde el principio para desarrollar un software seguro generalmente resulta en una aplicación segura. Sin embargo, hay que aplicar todo el tiempo de desarrollo e implementación las técnicas de seguridad para mantener y asegurar un sistema protegido y seguro.

En esta aplicación se diseñará procesos de 3 capas para mantener la seguridad de la mejor manera posible, haciendo que las funciones tengan un Gateway el cual se asegure de la correcta información a enviar. Además, se crearán clases específica para cada módulo, creando así una sectorización más importante dentro del sistema, manteniendo la información separada por área.

Sobrecarga de Buffer

Una de las mayores causas de violaciones de seguridad en las aplicaciones informáticas es por la sobrecarga de buffer. Una sobrecarga es simplemente donde el código se ubica en una pequeña porción de memoria para los parámetros, luego la función recibe más información de la que esperaba ocasionando que algún código maliciosos pueda ser corrido en el servidor, dañando información vital para el buen funcionamiento de la plataforma.

Hoy en día la mayoría de proyectos trabajan en lenguajes que se encargan automáticamente de la sobrecarga de buffer. Trabajar con Visual Studio C# significa que no se tiene que preocupar por la sobrecarga interna del buffer, ya que esta aplicación lo maneja muy efectivamente. Por estas razones, para todos los casos prácticos, es imposible desbordar el buffet dentro de los límites de la aplicación.

El desarrollo de esta aplicación será en base a la herramienta Visual Studio .NET C#, el cual es un lenguaje robusto y muy seguro para este tipo de aplicaciones. Gracias a esa herramienta y un diseño de modelos y capas, la seguridad e integridad de toda la plataforma está muy segura para la exposición que va a tener frente las diferentes amenazas existentes en este ámbito laboral.

3.1.3.3 Requerimientos de Interoperabilidad

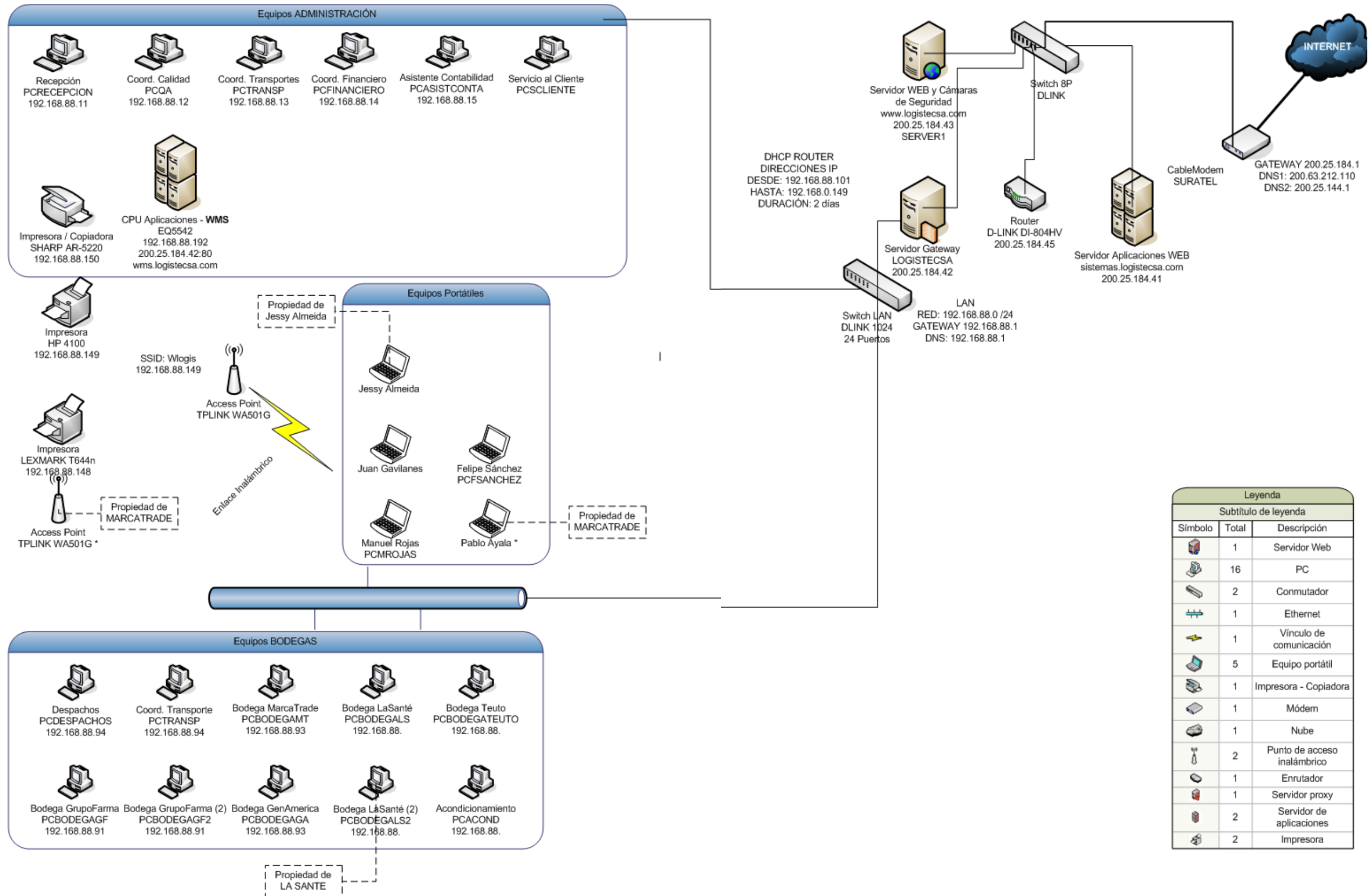
En la tabla 3.2 se puede ver las relaciones entre la interoperabilidad del sistema y su prioridad.

TABLA N° 3.2.: Prioridades

Interoperabilidad	Prioridad
WMS Base de Datos	Alta
El WMS tiene que trabajar completamente con los protocolos SQL	
El sistema WMS debe tener una interoperabilidad total con el servidor de bases de datos existente en la empresa	
El sistema debe poder autenticarse con las conexiones seguras del servidor de datos.	
WMS Conexión	Alta
Se debe tener una interoperabilidad completa con los protocolos TCP/IP para las transacciones.	
El sistema debe poder ampliarse para recibir consultas externas y sistemas TMS (transportation management system) instalado actualmente en la empresa.	
WMS Impresión	Media
El sistema debe poder trabajar en conjunto con sistemas de impresión instalado en la empresa para presentar los diferentes documentos que este genera	
El WMS debe tener una interoperabilidad con los procesadores de palabras como Word y Excel para transmitir estadísticas y datos a los diferentes usuarios del sistema.	

Impacto sobre la Red

FIGURA N° 3.3.: Estructura de la Red Logistecsa S.A



Leyenda		
Subtítulo de leyenda		
Símbolo	Total	Descripción
	1	Servidor Web
	16	PC
	2	Conmutador
	1	Ethernet
	1	Vínculo de comunicación
	5	Equipo portátil
	1	Impresora - Copiadora
	1	Módem
	1	Nube
	2	Punto de acceso inalámbrico
	1	Enrutador
	1	Servidor proxy
	2	Servidor de aplicaciones
	2	Impresora

3.1.3.4 Requerimientos de Rendimiento

Al momento de identificar y cuantificar los requerimientos de rendimiento, es muy importante identificar las razones atrás de cada requerimiento. Esto es parte de un proceso general de planificación.

Concepto Operacional

La proliferación de nueva tecnología en los almacenes ha hecho mucho más fácil de rastrear el movimiento de inventario, permitiendo así a las compañías mejorar las cifras de eficiencia, productividad y aciertos en el envío de órdenes. El problema actual es que la mayoría de procesos se los hace manualmente y con un sistema muy básico dentro de la empresa Logistecsa. Esta tecnología desarrollada a medida ayudará a los empleados a crear rutinas más eficientes, documentos digitales, órdenes y entregas semiautomáticas, dándole más tiempo al empleado para otras operaciones.

Mantenimiento de Software

El mantenimiento tanto de software como de hardware es muy importante para que el sistema continúe durante mucho tiempo con sus operaciones normales. El mantenimiento es la modificación de un producto de software después de la entrega para corregir ciertos errores, mejorar efectividad u otros atributos del software y adaptar el producto a un nuevo ambiente o requerimientos dispuesto por el plan de negocios.

Pigoski captura las necesidades para empezar el mantenimiento cuando el desarrollo empieza en una nueva definición:

“El mantenimiento de software es una actividad totalmente requerida para proveer un soporte de costo-beneficio al sistema de software. Las actividades

son desarrolladas durante la etapa de pre-entrega así como en la etapa de post-entrega. La actividad de pre-entrega incluye una planificación para la operación de post-entrega, operaciones, portabilidad y determinación logística. La actividad de Post-entrega incluye modificación de software, entrenamiento y un help desk operativo ”¹⁹

IEEE define el mantenimiento en cuatro categorías muy descriptivas para enfocar de una mejor manera estos conceptos y sus procesos correspondientes:

*“**Mantenimiento Correctivo:** modificación de reacción de un producto de software realizado después de la entrega para corregir los defectos descubiertos.*

***Mantenimiento adaptativo:** La modificación de un producto de software al cabo de su entrega para mantener el programa usable en un ambiente que es modificado.*

***Mantenimiento perfectivo:** La modificación de un producto de software hecha después de la entrega para mejorar su desempeño y su mantenibilidad.*

***Mantenimiento de emergencia:** mantenimiento correctivo no programado para mantener un sistema operativo. ”²⁰*

¹⁹ Pigoski, T. M. (1997) Practical Software Maintenance – Best Practices for Managing Your Software Investment. John Wiley & Sons. New York..NY.

²⁰ IEEE Std. 1219-1998 (1998) Standard for Software Maintenance. IEEE Computer Society Press. Los Alamitos. CA.

Mantenimiento de Software

Existen actualmente algunos requerimientos internacionales de mantenimiento de software que se tomaran en cuenta al momento de desarrollo de la aplicación WMS para la empresa Logistecsa.

El proceso de implementación contiene la preparación y la transición de las actividades del software, como la concepción y creación de un plan de mantenimiento, la preparación para manejar los problemas identificados durante el desarrollo y el seguimiento en la configuración del producto.

El proceso de aceptación de la modificación, debe chequearse con el individuo que hizo la petición del mantenimiento para asegurarse que la solución creada para ese mantenimiento sea la correcta y no haya problemas entre las distintas áreas.

El procesos de migración como es el cambio de plataforma o infraestructura es excepcional y no es parte de las tareas de mantenimiento diario. Si el software debe ser implementado sobre una nueva plataforma sin cambios en su funcionalidad, este procesos debe ser usado y mantenido por los desarrolladores del sistema para asegurar su perfecta funcionalidad.

En el caso del sistema WMS siempre se podrá migrar a otra plataforma Windows siempre y cuando esta pueda trabajar con el framework 3.5, ya que es la base de este sistema de almacenamiento y seguimiento de la empresa Logistecsa.

De vez en cuando se debe hacer un proceso de depuración de la base de datos con queries correspondientes para limpiar y corregir pequeños errores cometidos por los usuarios, ya que el sistema no es 100% seguro a corregir errores dados por el alto nivel de manipulación humana.

Requerimientos de rendimiento

La siguiente sección se listará los diferentes requerimientos de rendimiento para el WMS de la empresa Logistecsa. Estos requerimientos fueron sugeridos y aprobados por el administrador de sistemas y el administrador del proyecto.

En esta tabla 3.3 se describe a detalle cada sección del sistema con su descripción.

TABLA N° 3.3.: Descripción de Servicios

<i>Sección</i>	<i>Descripción</i>
<i>Niveles de servicio</i>	<ul style="list-style-type: none">• El servicio debe estar siempre en línea y disponible para hacer las diferentes transacciones requeridas por la empresa• En casos de emergencia se podría desconectar el servicio momentáneamente dado que todos los servicios se pueden retrasar períodos pequeños de tiempo hasta que el sistema este en línea.• Es permisible que por períodos de tiempo no muy seguidos el sistema este lento, dado que puede haber una sobrecarga de conexiones, pero siempre tiene que mantenerse en línea sin caerse.
<i>Reactivación</i>	<ul style="list-style-type: none">• <i>El sistema</i> tiene que estar en línea segundos después de que el servidor se

	restablezca, después de una falla energética
<i>Fiabilidad y Disponibilidad</i>	<ul style="list-style-type: none"> • La fiabilidad del sistema tiene que superar un 85% de la utilización total • El sistema tiene que ingresar a un 100% la integridad de los datos, ya que se manejan inventarios y la facturación de la empresa depende de estos directamente. • El sistema tiene que estar completamente activo y disponible las 24 horas del día, siempre y cuando el servidor y demás servicios alrededor del sistema WMS estén disponibles
<i>Mejoramiento</i>	<ul style="list-style-type: none"> • El sistema debe tener la posibilidad de aumentar módulos como un TMS (Transportation Management System) o incorporar con este un ERP • El sistema debe poder agregar sistemas de información por código de barras o IRP
<i>Software</i>	<ul style="list-style-type: none"> • Este sistema debe poder ser implementado bajo una plataforma Windows, dado que todo el sistema de la empresa Logistecsa se basa en software • Conexiones seguras a través de un servidor Windows 2005 y Microsoft SQL server 2000 • S.O Windows con compatibilidad de

	framework 3.5
<i>Hardware</i>	<ul style="list-style-type: none"> • El sistema debe funcionar bajo el sistema PC a partir de un procesador Intel P 4 o su equivalente en Amd con una memoria mínima de 512 en RAM y 8 megas de video • Debe tener una NIC FastEthernet o wireless 802.11 b/g para conectarse al sistema dentro de la red LAN

3.1.3.5 Requerimientos del Sistema

Para mantener la comunicación y entendimiento, los requerimientos deben ser documentados. Esto significa que son escritos con el propósito de eliminar ambigüedades tanto para los diseñadores como para los usuarios y clientes.

Las especificaciones técnicas son muy importantes ya que permitirá ver el efecto de este nuevo sistema en el entorno actual de la empresa y ver las diferentes posibilidades de tener cruces con sistemas ya implementados en la empresa Logistecsa.

Hay que analizar también la posibilidad de usar equipos y materiales ya existentes en la empresa o si se necesita comprar nuevos medios para implementar el sistema WMS y ponerlo en marcha.

Sistemas y Servicios de dependencias

Los sistemas informáticos siempre necesitan diferentes tipos de dependencias para poder ser utilizados correctamente. Hoy en día las computadoras son usadas como

herramientas multifuncionales para diferentes tareas como: comunicaciones, manipulación de imágenes, procesadores de palabras o para producir software. Muchos sub sistemas son necesarios para hacer estos diferentes procesos. Esto depende directamente de la utilidad que le dé el usuario y como este quiera configurarlo.

Las dependencias de este sistema pueden ser agrupadas en dos categorías: las dependencias de hardware y software.

Dependencias de Hardware

Cualquiera parte de computadora pertenece a esta categoría. Por ejemplo la tarjeta madre, disco duro, scanner, impresora o modem.

Para este sistema WMS se necesitara una compleja red de datos ya implementada en la empresa, donde intervienen las diferentes terminales a utilizar, los NIC's de cada computador conectado, cableado estructurado, una serie de switch que interconectan los diferentes sectores de la planta y los servidores requeridos para este proyecto.

A continuación una lista de requerimientos de hardware básicos para el funcionamiento del sistema en la tabla 3.4:

TABLA N° 3.4.: Requerimientos de Hardware

Tipo	Procesador	Descripción	Capacidad	N° Núcleos	Existente
Cliente					
Terminales Clientes	Intel/ Amd	Capacidad de procesamiento mayor a 2.3 Ghz	>2.3	1	Si
Servidor					
Servidor BDD	Intel	Servidores HP	>2.5	4	Si

Sistema Operativo

Un sistema operativo es un software de sistema, es decir, un conjunto de programas de computación destinados a realizar muchas tareas entre las que destaca la administración de los dispositivos periféricos.

TABLA N° 3.5.: Requerimientos de Software

Tipo	Fabricante	Descripción	Versión	Num. Licencias	Free
Servidor					
Sistema Operativo	Microsoft	Windows	2005 R2 Enterprise Edition SP2	1	No
Base de Datos	Microsoft	SQL Server	2005 Enterprise edition SP2 Español10i	1	No
Software Base	Microsoft	Framework	2.0	1	Si
	Microsoft	Framework	3.0	1	Si
	Microsoft	Framework	3.5	1	Si
	Microsoft	Framework	2.0Service Pack 2	1	Si
	Microsoft	Framework	3.0Service Pack 2	1	Si
	Microsoft	Framework	3.5(kb953595)	1	Si
	Microsoft	Framework	3.5 SP1	1	Si
Cliente					
Sistema Operativo	Microsoft	Windows	2008 R2	10	No
Base de Datos					
Software Base	Microsoft	Office	2010	2	No

3.2 Planificación

3.2.1 Vista Conceptual.

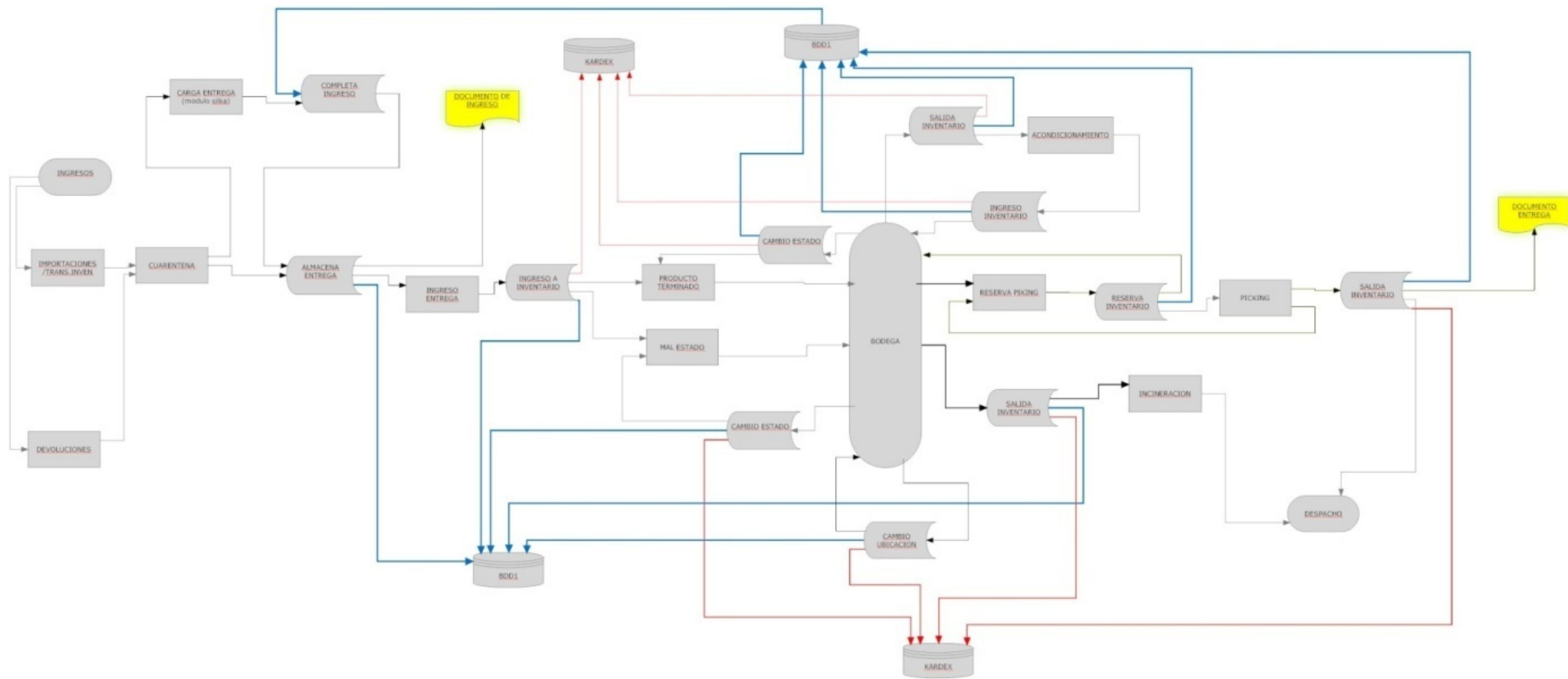


FIGURA N° 3.4.: Diagrama Conceptual

Visual Estudio 2008 y el Framework 3.5 permitirán a los desarrolladores crear rápidamente aplicaciones eficientes, seguras y reducidas en costo. Esta herramienta permite a las organizaciones crear rápidamente aplicaciones redituables y manejables a otro nivel como es el caso de Logistecsa y sus necesidades específicas sobre una herramienta de manejo de inventario y procesos internos dentro de sus bodegas.

Por estas razones los directivos del área de informática determinaron que la mejor opción era desarrollar la aplicación WMS bajo esta estructura y el lenguaje C# como base a este desarrollo.

3.2.3. Vista Lógica

3.2.3.1. Diagrama de Clases

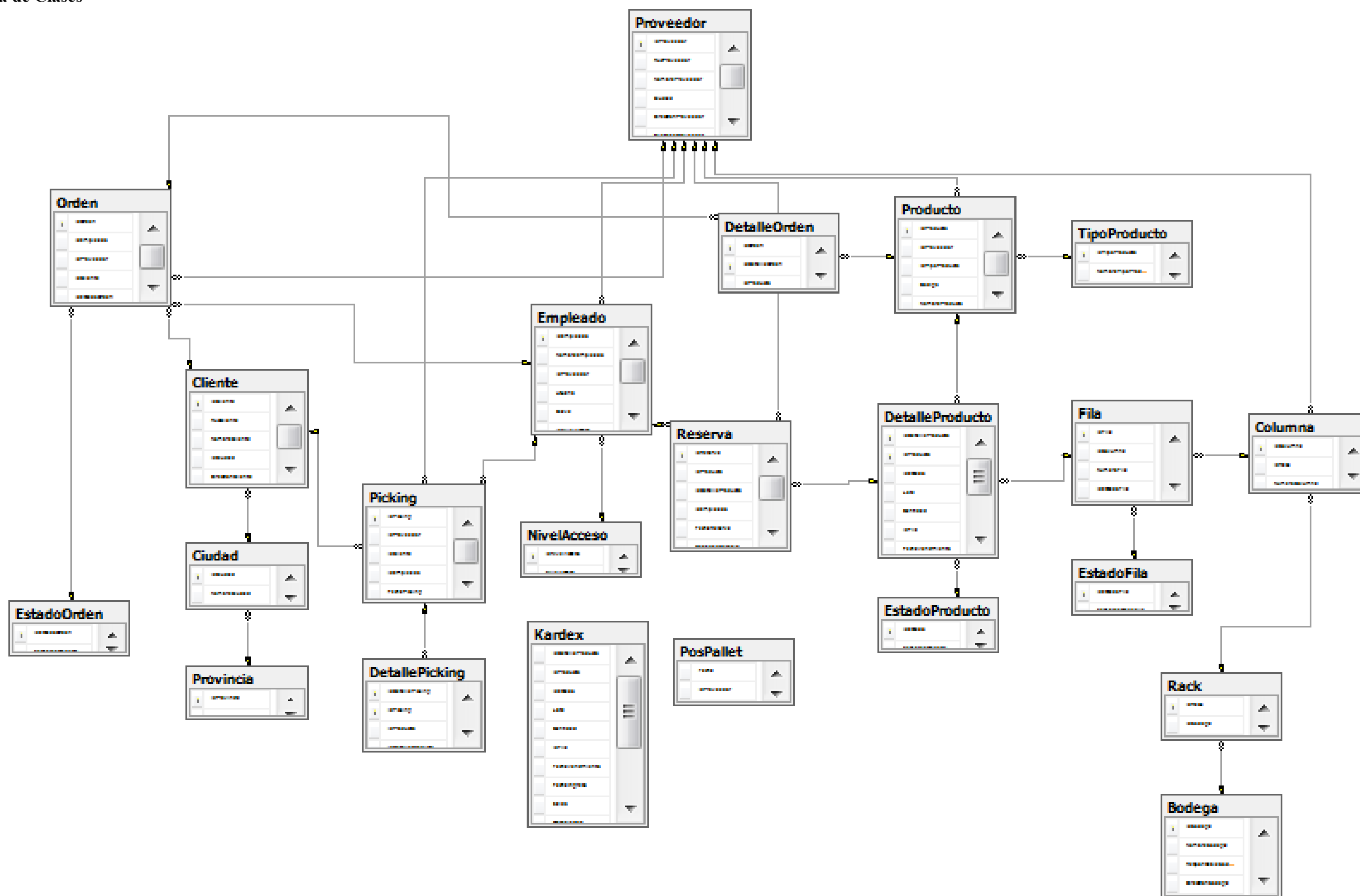


FIGURA N° 3.5.: Diagrama de Clases

3.2.2.2. Diagrama de Casos de Uso

En la siguiente figura 3.7 se aprecia el diagrama simplificado de uso, donde se encuentra el sistema WMS como contenedor de secuencias entre usuarios, lógica e información y la base de datos (BDD)

Diagrama Simplificado de Uso

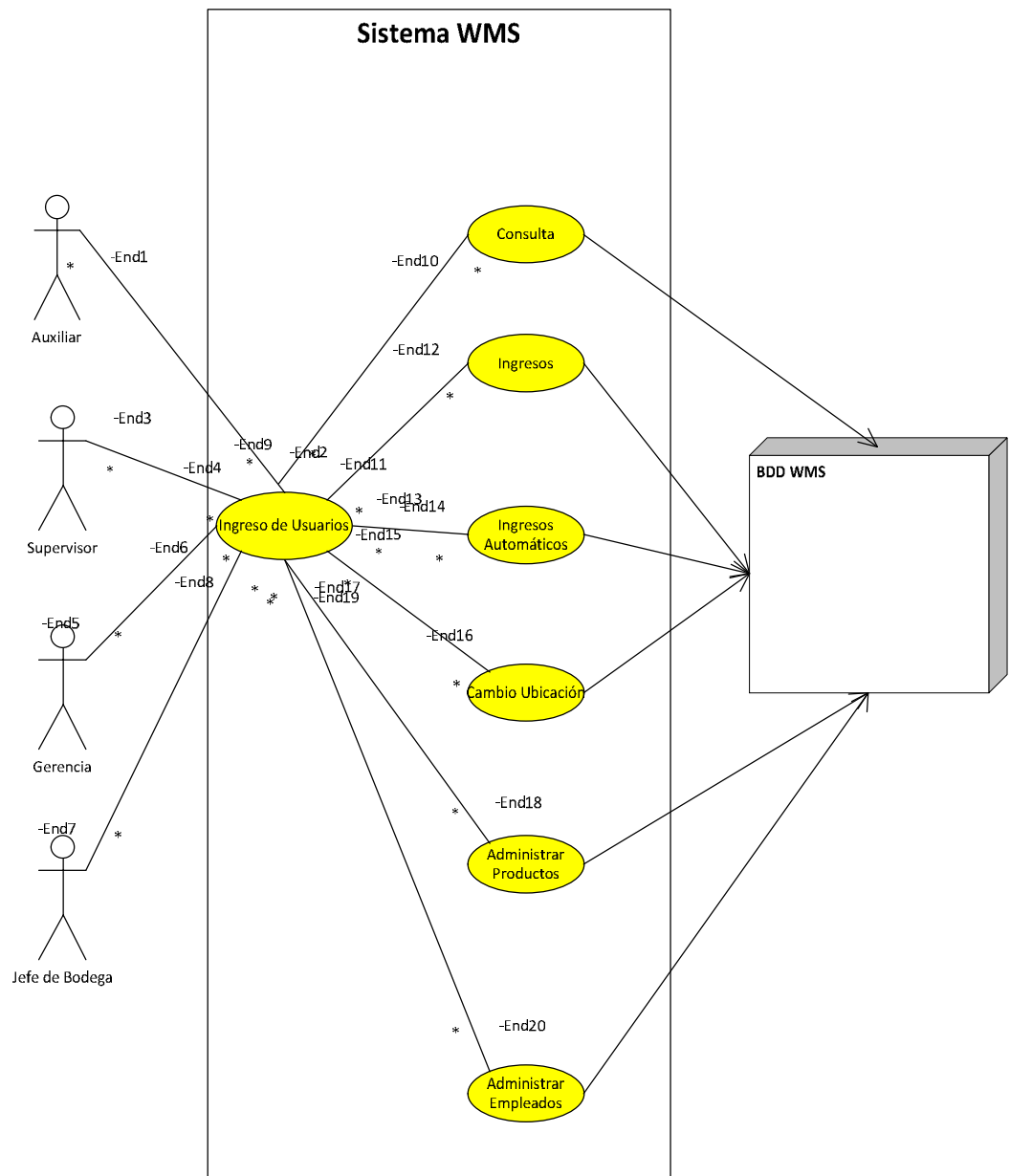


FIGURA N°3.6.: Diagrama simplificado de Uso

Diagrama de Casos de Uso

En este diagrama (figura 3.7) se ve en detalle cada interacción del sistema WMS con cada usuario (nivel) y sus diferentes ejecuciones con la base de datos

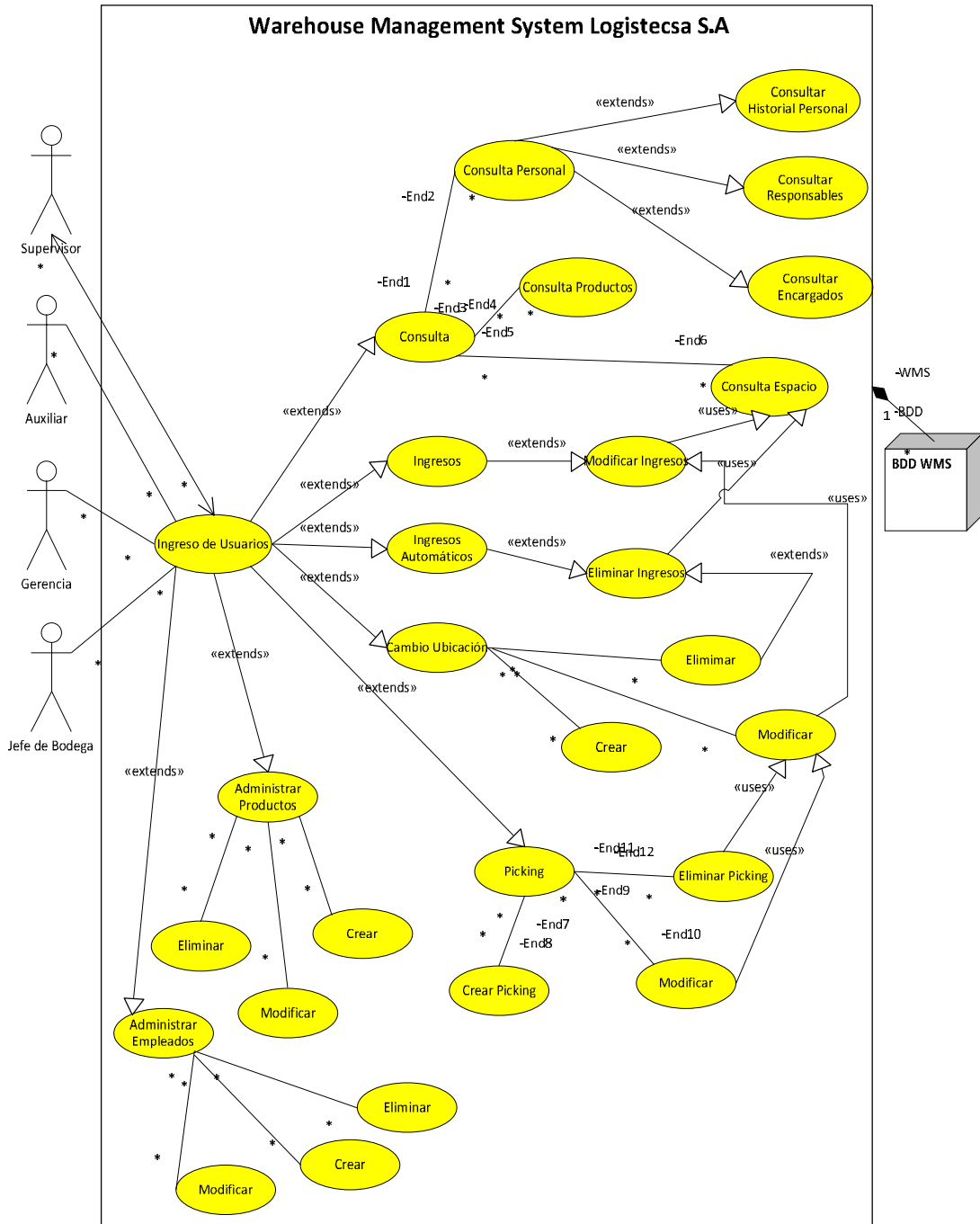


FIGURA N° 3.7.: Diagrama de Casos de Uso

3.2.3. Plan de Riesgos

Para el buen desarrollo del sistema WMS en la empresa Logistecsa S.A. es necesario identificar de manera eficaz los riesgos del proyecto. El seguimiento continuo de los riesgos, permite que se ejecuten los planes de contingencia planificados para todos los riesgos planteados a tiempo.

Plan General de Riesgo

En el plan general de riesgos se puede observar cada riesgo planteado, administrarlos de manera eficaz y ejecutar su solución, también se da un seguimiento continuo al estado de cada uno de estos y actualizaciones a tal documento.

A continuación se presentará el cuadro de riesgos identificando los riesgos del proyecto WMS.

Matriz de Riesgos

Probabilidad: 1-Baja	Impacto: 1 - Baja
2- Media	2 - Medio
3 - Alta	3 – Alta

TABLA N°3.6.: Tabla de Riesgos

Descripción / Condición	Consecuencia	Probabilidad	Impacto	Exposición	Estrategia Mitigación	Contingencia
Los entregables de cada fase no puedan ser revisados en el tiempo establecido en el cronograma por los responsables de aceptación de los entregables	Retraso en el proyecto.	2	3	6	Entregas parciales de los avances con anticipación a la fecha de entrega del documento final.	Calendarizar previamente la revisión/aprobación de los documentos
El retraso en la entrega información y/o recursos solicitada para desarrollo del proyecto	Retraso en el proyecto.	2	3	6	Solicitar la información con anticipación	Reprogramar las actividades a través de un Control de Cambios
Que no haya continuidad en los jefes de cada área dentro de Logistecsa	Que la calidad del proyecto sea afectada y que el proyecto no concluya en el tiempo establecido	1	3	3	Se entregara al programador un acta de los pasos y requerimientos del sistema	Los pasos a seguir se aseguran dentro de la empresa para que futuros cambios no lo afecten
Que los requerimientos del alcance de la solución sean mayores a las acordadas en este proyecto	Nivel de insatisfacción de los usuarios de la solución	1	3	3	Comunicación adecuada sobre el alcance del proyecto	No aplica
El servicio eléctrico por causas externas no esté disponible	Suspensión de las actividades	1	3	3	La empresa cuenta con un generador Kholer de alta gama	Se trabajara siempre con la pila de la laptop conectada para casos así seguir trabajando sin cortes
El hardware no está disponible para el ambiente de desarrollo	Retraso en la implementación	3	2	6	Solicitud de antemano de los equipos	Se seguirá trabajando localmente hasta que estén las condiciones adecuadas para el desarrollo
Afectar a los usuarios en sus labores del día a día	Afectar la productividad de los usuarios	3	3	9	Tener definido un proceso de rollback	Crear un ambiente aislado para evitar afectar las actividades de los usuarios
La no disponibilidad de un ambiente de producción.	Retraso en el proyecto	1	3	3	Hacer seguimiento a la preparación del servidor	Se evaluará en la fecha antes de implementar en producción
Demora en la entrega de requerimiento por parte de Logistecsa	Retraso en el proyecto	1	3	3	Solicitar el requerimiento con anticipación	Reprogramar las actividades dentro del cronograma
Retrasos provocados por el Estudiante	Retraso en el proyecto	3	3	9	Solicitar apoyo del departamento de sistemas	Reprogramar las actividades dentro del cronograma

3.2.4. Plan de Pruebas

Prueba de la integridad de los datos y de la base de datos

- Las bases de datos y los procesos de la base de datos se deberían probar como subsistema independiente. Esta exanimación deberá probar los subsistemas dentro de la interfaz del usuario del objetivo-de-prueba como la interfaz a los datos.

TABLA N°3.7. : Tabla de Pruebas

Objetivo de Prueba:	<ul style="list-style-type: none">• Que la clase de conexión, y la escritura de datos sean correctas
Técnica:	<ul style="list-style-type: none">• Probar cada query de inserción de datos, verificando su validez en la base de datos.• Examinar en la base de datos si los parámetros enviados se insertaron y guardaron válidamente por cada query y a la vez probar si el mensaje de exitoso se devuelve.
Herramientas requeridas	<ul style="list-style-type: none">• La técnica requiere las siguientes herramientas:• Herramienta de automatización del script de prueba• Herramientas de recuperación y respaldo• Herramientas y utilidades SQL de base de datos• Scripts de generación de datos

Criterios de Terminación:	<ul style="list-style-type: none"> • Los datos enviados deben ser exactos en la base según como se hayan guardado
Consideraciones Especiales:	<ul style="list-style-type: none"> • Las pruebas necesitan estar aisladas del ambiente de producción y sin acceso para los otros usuarios • Los procesos deben ser llamados manualmente.

Prueba de Funcionalidad

- La prueba de funcionalidad sirve para controlar si todos los botones, accesos a usuarios y limitantes del sistema están funcionando como los requerimientos especifican. Este tipo de prueba se las hace sobre una prueba de caja negra; para controlar la integración de los componentes, clases y funciones con la aplicación y su GUI (interfaz grafica de usuario)

TABLA N°3.8 Prueba de Funcionalidad

Objetivo de Prueba:	<ul style="list-style-type: none"> • La función del sistema está adecuada, la interacción entre diversos componentes y cómo estos manejan los datos
---------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------

Técnica:	<ul style="list-style-type: none"> • Insertar datos erróneos para ver cómo el sistema valida. • Adquirir todos los mensajes de error para comparar y ver si están válidos. • Determinar si cada regla de negocio se aplica correctamente.
Herramientas Requeridas:	<ul style="list-style-type: none"> • La técnica requiere las herramientas siguientes: • Herramienta de debug (compilador) • Herramientas de monitoreo- • Herramientas y utilidades SQL de base de datos
Criterios de Terminación:	<ul style="list-style-type: none"> • Se logró el objetivo de la prueba al tener todos los datos válidos y validaciones disponibles.
Consideraciones Especiales:	<ul style="list-style-type: none"> • Tener el lenguaje de la interfaz como predeterminada para que no se ingresen caracteres inexistentes.

Prueba de Interfaz del Usuario

- La prueba de interfaz de usuario se lleva a cabo para determinar si la interfaz utilizada es amigable con el usuario y si todas las opciones son accesibles de manera eficiente por el usuario final.

TABLA N°3.9 Prueba de Interfaz de Usuario

Objetivo de Prueba:	<ul style="list-style-type: none"> • Verificar lo siguiente: • La navegación a través del objetivo-de-prueba refleja correctamente en las funciones y los requerimientos, incluyendo ventana-a-ventana, campo-a-campo, y el uso de los métodos de acceso (tabuladores, movimientos del ratón, claves de aceleración). • Objetos y características de la ventana, tales como menús, tamaño, posición, estado y foco se conforman con los estándares.
Técnica:	<ul style="list-style-type: none"> • Cree o modifique las pruebas para que cada ventana verifique los estados apropiados de navegación y del objeto para cada ventana y los objetos de uso.
Herramientas requeridas	<ul style="list-style-type: none"> • Se requiere el compilador de C# .NET
Criterios de Terminación:	<ul style="list-style-type: none"> • La técnica apoya la prueba de cada pantalla o ventana principal que será usada extensamente por el usuario final.
Consideraciones Especiales:	<ul style="list-style-type: none"> • Se requiere tener una resolución estándar de 1024 x 760 ya que esta es el requerimiento del cliente

CAPÍTULO IV

DESARROLLO Y ESTABILIZACIÓN DEL APLICATIVO

4.1 Desarrollo

4.1.1 Desarrollo Estructural de la Plataforma WMS

En esta figura se puede observar un diagrama de estructuras del sistema con sus diferentes procesos y etapas en cada nivel y correlatividad entre estas.

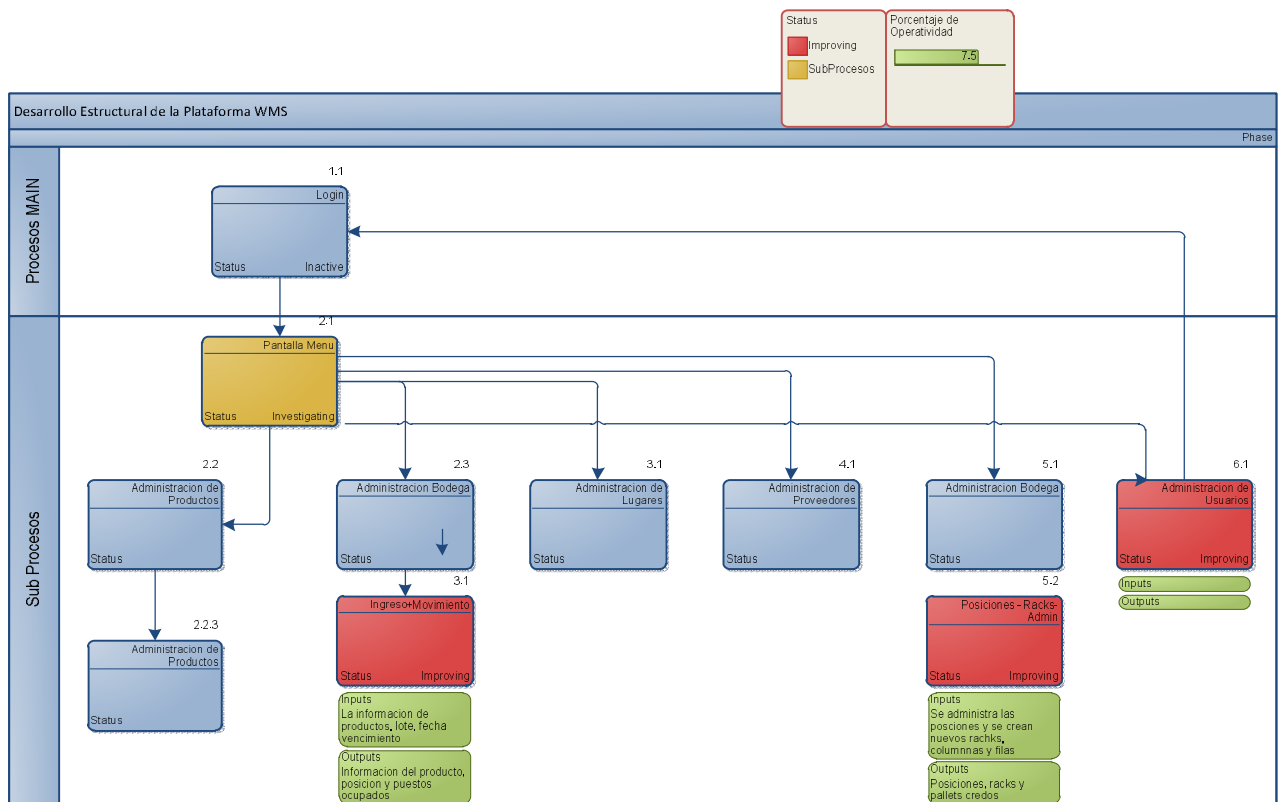


FIGURA N°4.1.: Diagrama Estructural

4.1.2 Desarrollo de las Interfaces de la Plataforma WMS

Prototipos de Pantalla

La Figura 4.2 muestra el prototipo de pantalla de ingreso al sistema donde el usuario ingresa sus credenciales de acceso.

Pt1 :	Validación de Usuarios
Descripción:	Ingreso al sistema WMS por la intranet
Casos de uso o RF asociado:	CU1



FIGURA N° 4.2.: Prototipo de Pantalla Validación de usuarios

En la figura 4.3 se puede observar el menú principal de la aplicación wms donde se expande a las demás propiedades.

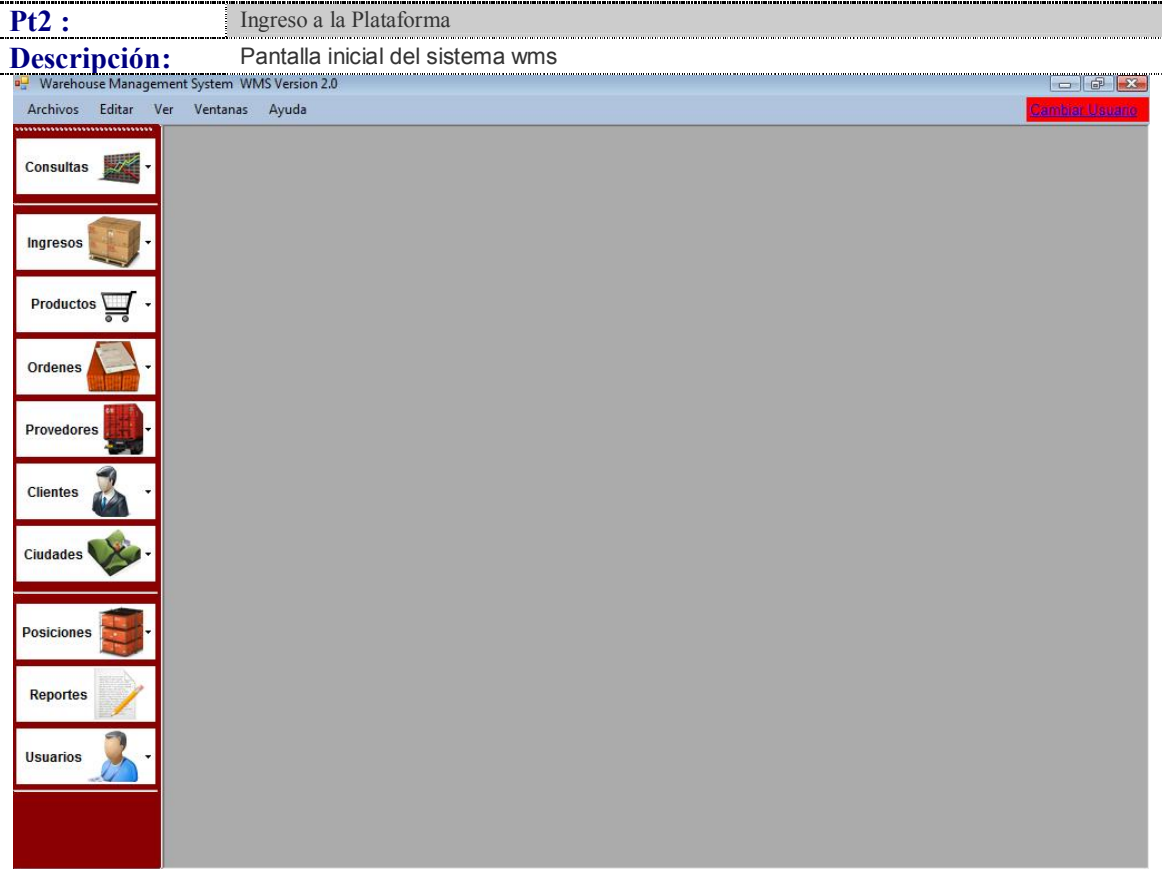


FIGURA N° 4.3.: Prototipo de Pantalla Ingreso a la plataforma

En la figura 4.4 se muestra la pantalla de consulta de productos por bodega y proveedor haciendo referencia a búsquedas.

Pt3 : Consulta de Productos

Descripción: Se consulta los productos por bodega y proveedor

Casos de uso o RF asociado: CU2

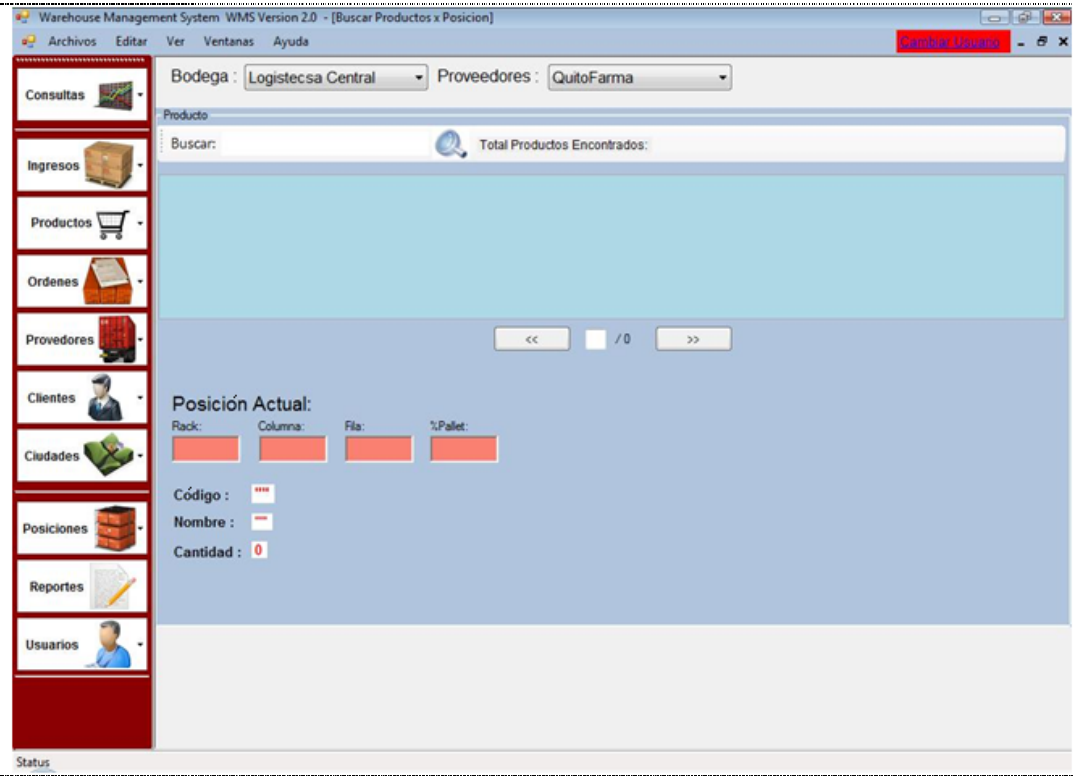


FIGURA N° 4.4.: Prototipo de Pantalla Consulta de protocolos

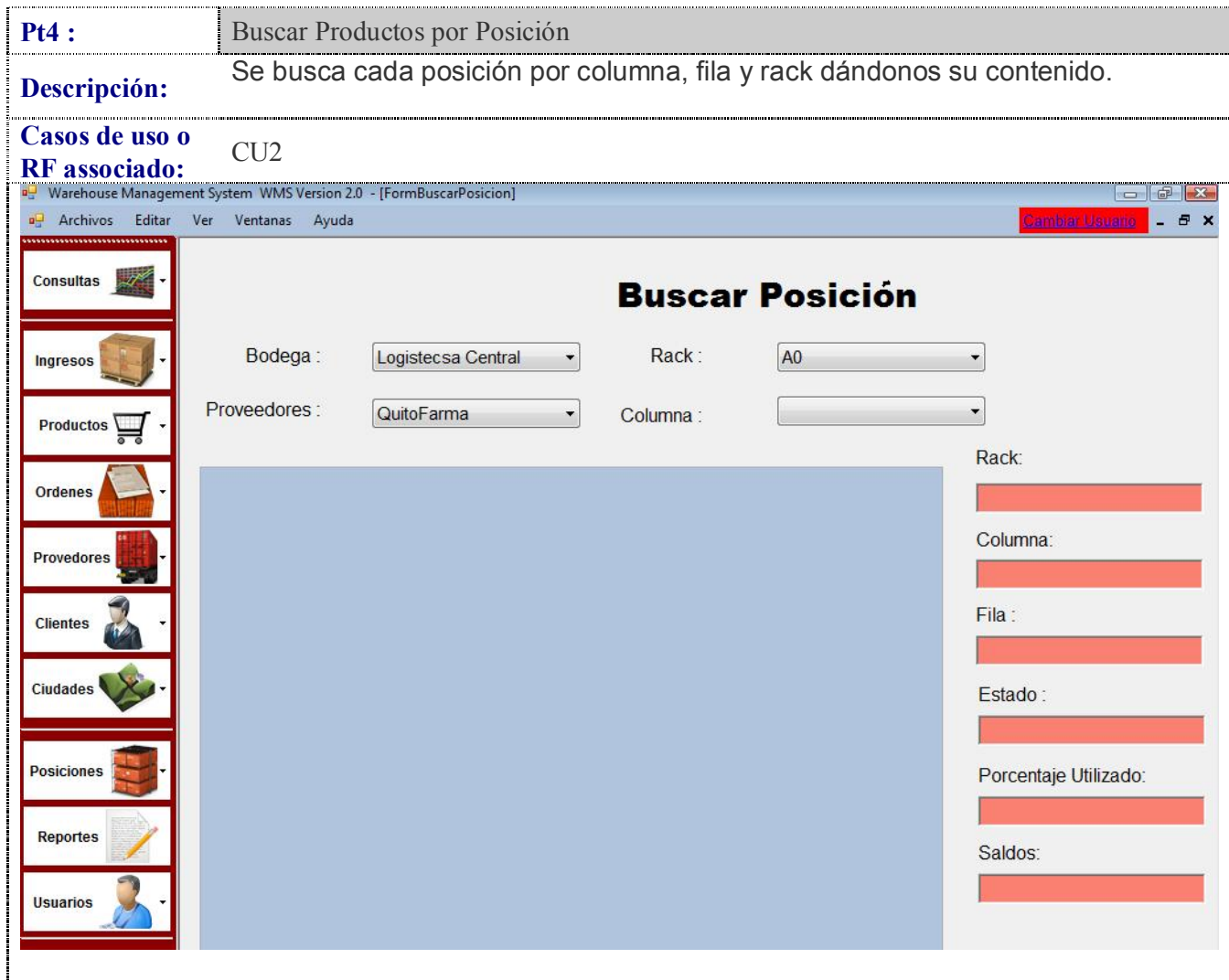


FIGURA N° 4.5.: Prototipo de Pantalla Búsqueda de productos por posición

Pt5 : Crear Producto

Descripción:

Pantalla para Crear Nuevos Productos dentro del sistema

Casos de uso o RF asociado:

CU9

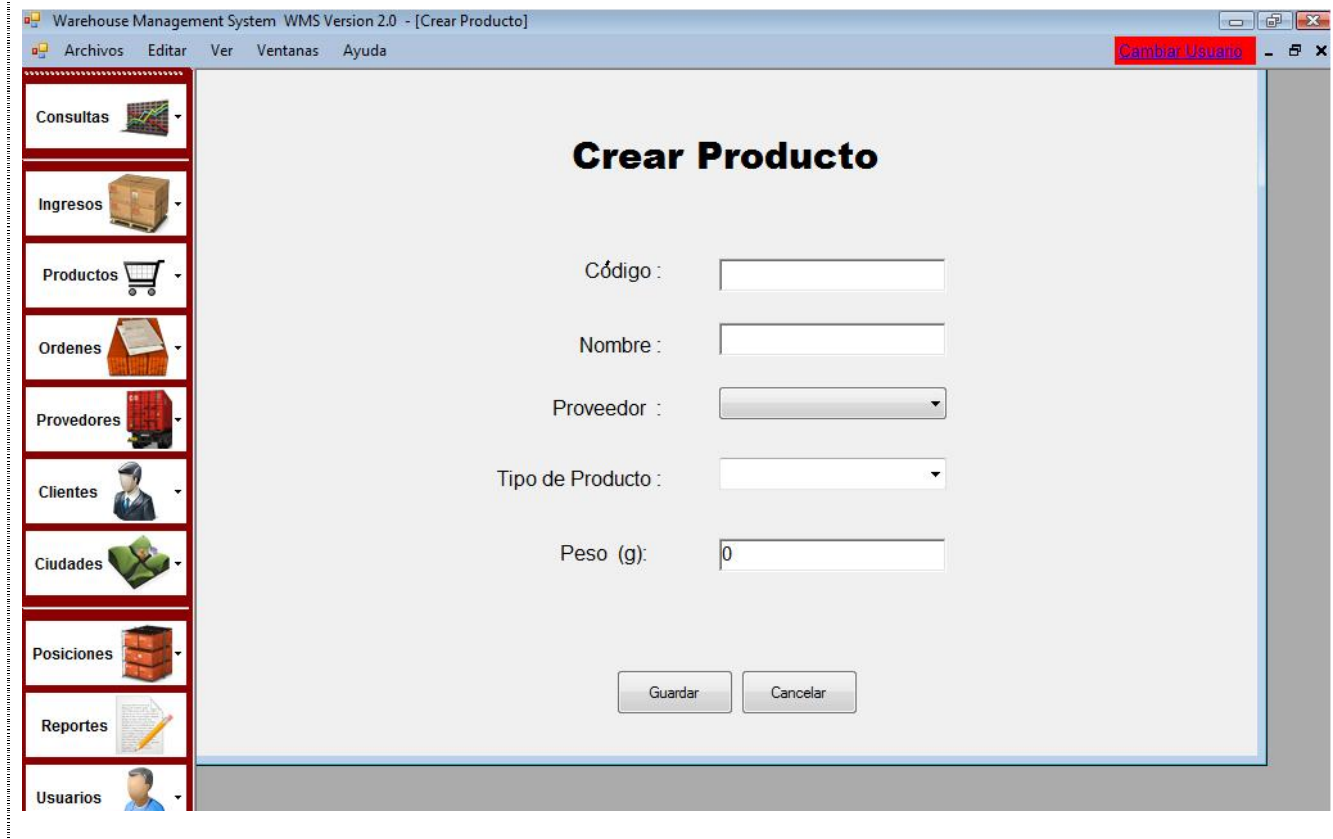


FIGURA N° 4.6.: Prototipo de Pantalla Crear producto

Pt6 : Modificar Producto

Descripción: Pantalla para Modificar Productos existentes dentro del sistema wms

Casos de uso o RF asociado: CU10

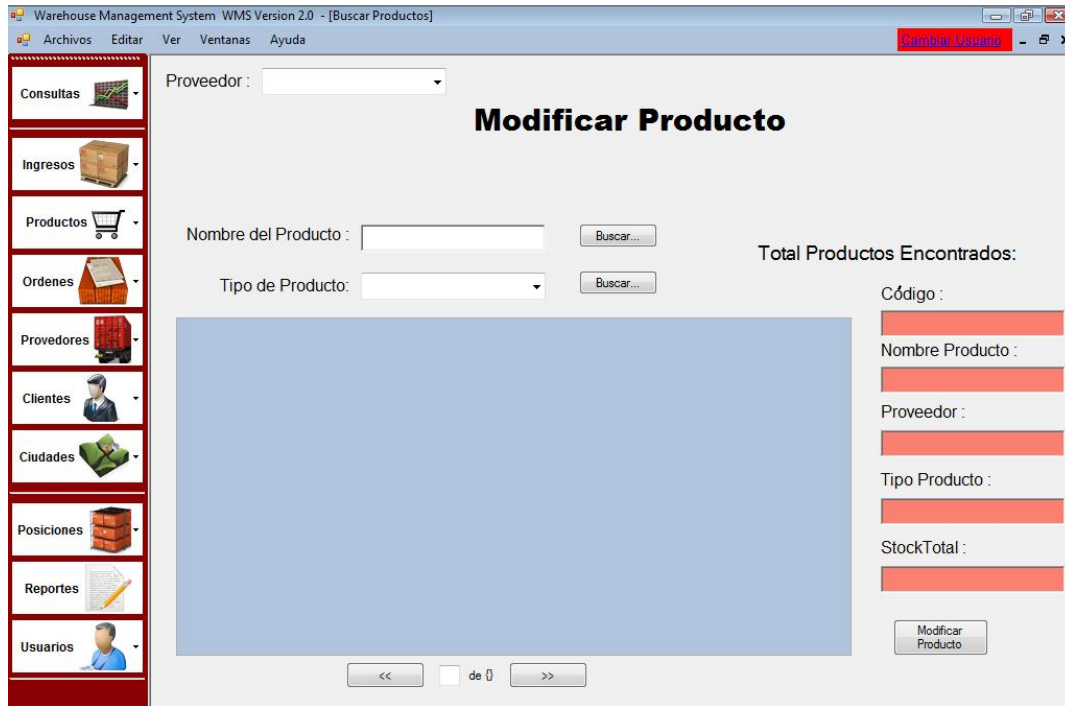


FIGURA N° 4.7.: Prototipo de Pantalla Modificar producto

Pt7 :	Eliminar Producto
Descripción:	Pantalla para Eliminar Productos existentes dentro del sistema wms si haber sido usados en cualquier tipo de transacción
Casos de uso o RF asociado:	CU11

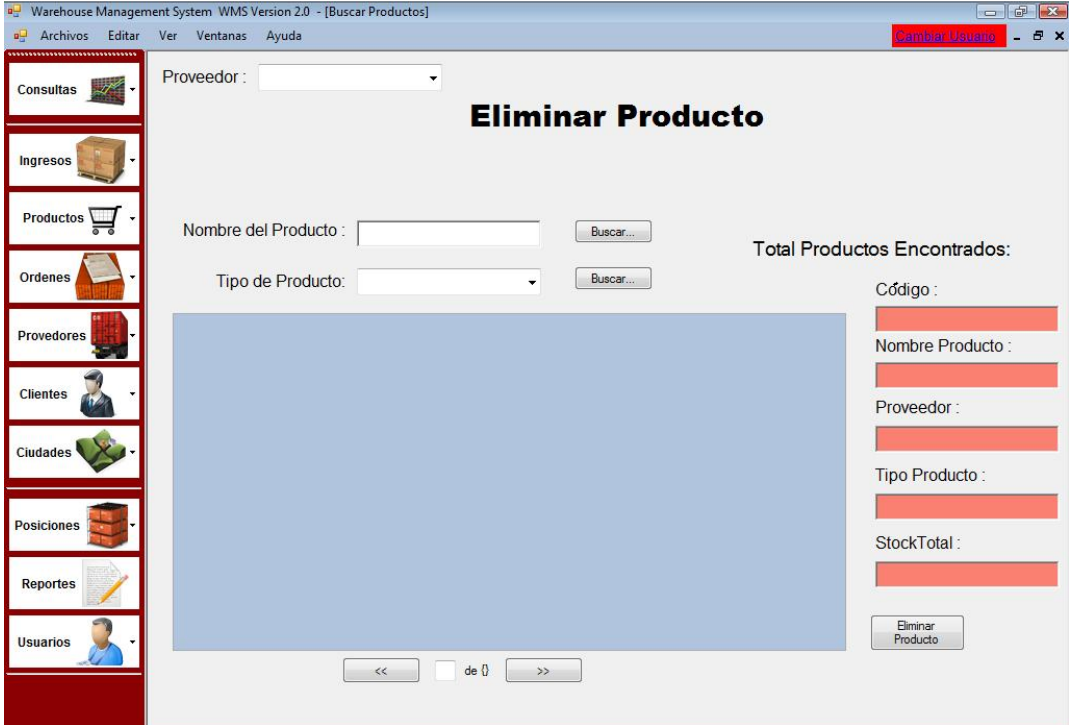


FIGURA N° 4.8.: Prototipo de Pantalla Eliminar producto

Pt8 :	Órdenes
Descripción:	Crea ordenes dentro del sistema, cargando información tanto del usuario como de la BDD central.
Casos de uso o RF asociado:	CU5

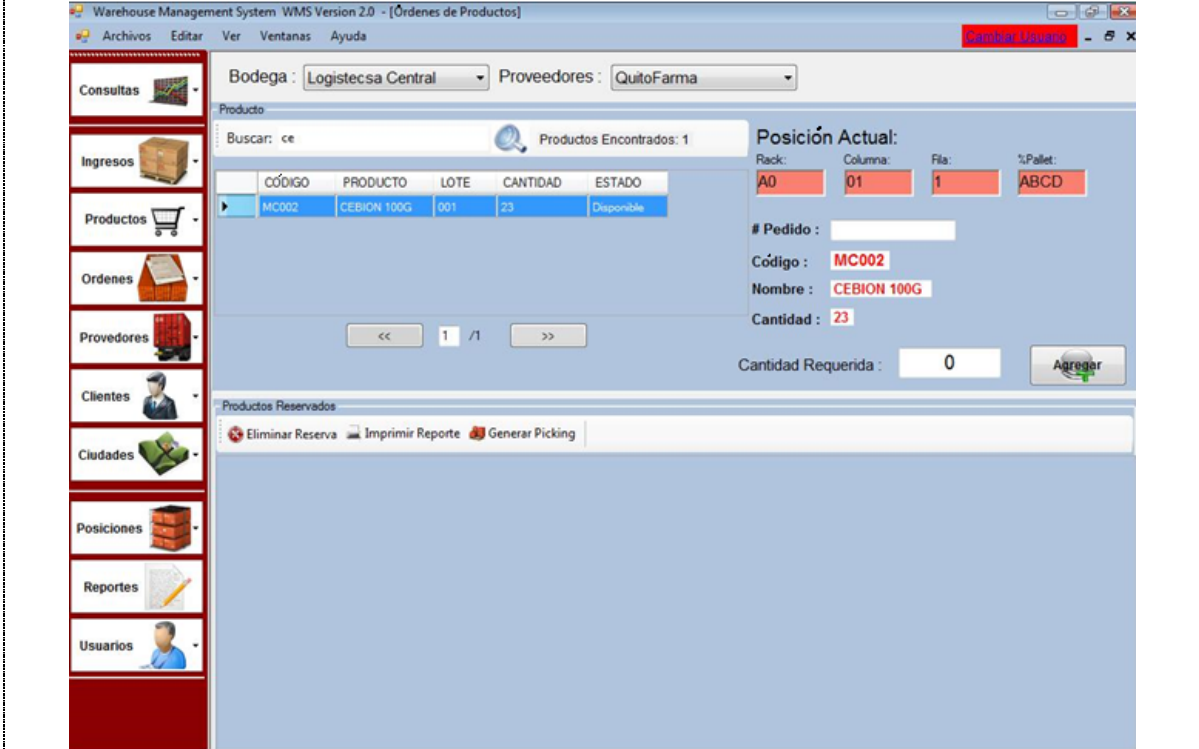


FIGURA N° 4.9.: Prototipo de Pantalla Órdenes

Pt9:	Cargar Ordenes
Descripción:	Modulo dependiente de Pt8 en el cual se puede cargar ordenes no establecidas
Casos de uso o RF asociado:	CU6

FIGURA N° 4.10.: Prototipo de Pantalla Cargar Órdenes

4.2. Pruebas

4.2.1. Prueba de la integridad de los datos

Para este capítulo se estableció una muestra de los datos a insertar de cada clase del sistema WMS para la demostración, ya que cada query de inserción se probó unitariamente durante el proceso de desarrollo.

Query:

```
public void IngresarCiudad(ECiudades aux)
{
    base.ExecuteNonQuery("insert into Ciudad values (NEWID(),'" +
aux.get2NombreCiudad() + "',''" + aux.get3idProvincia()+ "')");
}
```

Inserción:

Ciudad	idProvincia
Quito	1

Query:

```
public void IngresarCliente(ECliente aux)
{
    base.ExecuteNonQuery("insert into Cliente values (NEWID(),'" +
aux.get2RucCliente() + "',''" + aux.get3NombreCliente() + "',''" +
+aux.get4idCiudad()+ "',''" +aux.get5DireccionCliente()+ "',''" +
+aux.get6telefonoCliente()+ "',''" +aux.get7Representate()+ "')");
}
```

Inserción:

	idCliente	RucCliente	NombreCliente	idCiudad	DireccionCliente	telefonoCliente
1	26CA7896-20A5-4D59-B83F-C82A3B6257CE	1709533572	FYBECCA	6119C649-F697-474F-BC41-6B2F6573622D	AV 10 DE AGOSTO	34545454

Query:

```
public void IngresarEmpleado(EEmpleado aux)
{
    base.ExecuteNonQuery("insert into Empleado values (NEWID(),'" +
+ aux.get1Nombre() + "',''" + aux.get2idProveedor() + " ',''" +
aux.get3Usuario() + " ',''" + aux.get4Clave() + " ',''" +
aux.get5idNivelAcceso() + " ',''" + aux.get6Correo() + " ',''" +
aux.get7Celular() + "')");
}
```

}

Inserción:

idEmpleado	NombreEmpleado	idProveedor	Usuario	Clave	idNivelAcceso	Correo	Celular
6416FA6D-ADBA-4490-B36F-37E2E62C9C0E	Andres Marchant	443F5821-94AF-4035-9B54-1FD05D28BE8C	amarchant	as	5	amarchant@logistecsa.com	084533261

Query:

```
public void InsertarPicking(EPicking Pick)
{
    base.ExecuteQuery("Insert into Picking
values ('"+Pick.get1idPicking()+"', '"+ Pick.get2idProveedores() +
''," + Pick.get3idCliente() + "','"+ Pick.get4idEmpleado() + "','"+
Pick.get5FechaPicking() + "','"+ Pick.get6NumeroPedido() +
''," +Pick.get7Bultos()+")");
}
```

Inserción:

idPicking	idProveedor	idCliente	idEmpleado	FechaPicking	NumeroPedido	Bultos
8D501AF8-5EDD-4365-ADD3-173FD0852FD1	443F5821-94AF-4035-9B54-1FD05D28BE8C	26CA7896-20A5-4D59-B83F-C82A386257CE	6416FA6D-ADBA-4490-B36F-37E2E62C9C0E	10/24/2011 3:34:44 PM	5466	2

Query:

```
public void InsertarDetallePicking(EPicking Pick)
{
    base.ExecuteQuery("Insert into DetallePicking
values (NEWID(), '"+Pick.get1idPicking()+"', '"+Pick.getBidProducto()+"',
'"+ Pick.getCidDetalleProducto()+"', '"+Pick.getDCantidad()+")");
    base.ExecuteQuery("Insert into Kardex
(idDetalleProducto, idProducto, idEstado, Lote, Cantidad, idFila, FechaVenci-
miento, FechaIngreso, Saldo, Observacion, FechaSalida, Operacion, idEmpleado
, FechaOperacion) Select
idDetalleProducto, idProducto, idEstado, Lote, '"+Pick.getDCantidad()+"', idF-
ila, FechaVencimiento, FechaIngreso, Saldo, Observacion, '"+
Pick.get5FechaPicking()+"' as FechaSalida, 'Salida de Producto' as
Operacion, '"+Pick.get4idEmpleado()+"' as idEmpleado, '"+
Pick.get5FechaPicking()+"' as FechaOperacion from DetalleProducto d
where d.idProducto='"+ Pick.getBidProducto() + "' and
d.idDetalleProducto='"+ Pick.getCidDetalleProducto() + "'");
    base.ExecuteQuery("Delete Reserva where idReserva='"+
Pick.getIdReserva() + "'");
    base.ExecuteQuery("Delete DetalleProducto where
idDetalleProducto='"+ Pick.getCidDetalleProducto() + "' and
idProducto='"+ Pick.getBidProducto() + "' and Cantidad=0");
}
```

Inserción:

idDetallePicking	idPicking	idProducto	idDetalleProducto	cantidad
C097396F-60FE-4C2A-9053-B2C062D08A0D	8D501AF8-5EDD-4365-ADD3-173FD0852FD1	FE076B54-28C6-4324-95D5-C31D1B59FF91	A89627A1-9548-431C-BDC4-7C5948E85B05	50

Query:

```
public void InsertaReserva(EReserva Res, int Cant)
```

```

    {
        base.ExecuteNonQuery("Insert into Reserva
values (NEWID(), '"+Res.get2idProducto()+"', '"+Res.get3idDetallePro()+"'
, '"+Res.get4idEmpleado()+"', '"+
Res.get5FechaReserva()+"', '"+Res.get6CantidadReserva()+"', '"+Res.get7Num
eroPedido()+"', '"+Res.get8idProveedor()+"'");
        base.ExecuteNonQuery("Update DetalleProducto set Cantidad="
+ (Cant - Res.get6CantidadReserva()) + " where idDetalleProducto='" +
Res.get3idDetallePro() + "' and
idProducto='"+Res.get2idProducto()+"'");
    }

```

Inserción:

Query:

```

public void InsertarPosicion(EPosicion aux)
{
    String aux2;
    base.ExecuteNonQuery("insert into Rack values (NEWID(), '" +
aux.getidBodega() + "', '" + aux.getNombreRack() + "'");
    aux2 = ObtenerRack(aux.getNombreRack());
    for (int i = 0; i < aux.getNumeroColumna(); i++)
    {
        if (i < 9)
        {
            base.ExecuteNonQuery("insert into Columna
values (NEWID(), '" + aux2 + "', '0" + (i + 1).ToString() + "', '" +
aux.getidProveedor() + "'");
        }
        else
        {
            base.ExecuteNonQuery("insert into Columna
values (NEWID(), '" + aux2 + "', '" + (i + 1).ToString() + "', '" +
aux.getidProveedor() + "'");
        }
    }
    for (int i = 0; i < aux.getNumeroColumna(); i++)
    {
        for (int j = 0; j < aux.getNumeroFila(); j++)
        {
            if (i < 9)
            {
                base.ExecuteNonQuery("insert into Fila
values (NEWID(), '" + ObtenerColumna("0" + (i + 1).ToString(), aux2) +
"', '" + (j + 1).ToString() + "', 1, 0, '");
            }
            else
            {
                base.ExecuteNonQuery("insert into Fila
values (NEWID(), '" + ObtenerColumna((i + 1).ToString(), aux2) + "', '" +
(j + 1).ToString() + "', 1, 0, '");
            }
        }
    }
}

```

Inserción:

	idRack	idBodega	NombreRack
1	F7582DFB-EED5-43D1-BE91-F71BAD72862B	6EAE9BD6-F4DC-4233-B04E-5768B97BD916	A0

	idColumna	idRack	NumeroColumna	idProveedor
1	39AD1CD9-94F3-439A-8C8B-3787C624DCC5	F7582DFB-EED5-43D1-BE91-F71BAD72862B	01	443F5821-94AF-4035-9B54-1FD05D2BBE8C
2	0AD4F832-4C59-429C-B876-9A36DF7997EC	F7582DFB-EED5-43D1-BE91-F71BAD72862B	02	443F5821-94AF-4035-9B54-1FD05D2BBE8C
3	8AA23558-28D4-4FF0-A9CA-20022B242A19	F7582DFB-EED5-43D1-BE91-F71BAD72862B	03	443F5821-94AF-4035-9B54-1FD05D2BBE8C
4	1002AAAD-7F12-432F-A772-8381425EFF91	F7582DFB-EED5-43D1-BE91-F71BAD72862B	04	443F5821-94AF-4035-9B54-1FD05D2BBE8C
5	092CF057-6086-4426-BBB3-9A0622CFC511	F7582DFB-EED5-43D1-BE91-F71BAD72862B	05	443F5821-94AF-4035-9B54-1FD05D2BBE8C

	idFila	idColumna	NumeroFila	idEstadoFila	PorcentajeUtilizado	Saldo
1	3298C40F-6B71-4A56-9E4A-597F4845EE32	0AD4F832-4C59-429C-B876-9A36DF7997EC	1	1	0	
2	468D7293-9E8F-4686-B7E4-8C80F94BC458	39AD1CD9-94F3-439A-8C8B-3787C624DCC5	1	1	0	
3	2BD753C5-EE0C-4CB3-9451-9BE11E10D0B0	1002AAAD-7F12-432F-A772-8381425EFF91	1	1	0	
4	6E5285DE-88BD-4CE5-A4CD-DE22B83FA538	8AA23558-28D4-4FF0-A9CA-20022B242A19	1	1	0	
5	E0DD3893-BEB6-4747-95EE-D889747479B8	092CF057-6086-4426-BBB3-9A0622CFC511	1	1	0	
6	74B463B0-06B8-4885-9A69-D5B18DAE6A23	092CF057-6086-4426-BBB3-9A0622CFC511	10	1	0	
7	D76CDB47-7AAD-42FE-BFAA-F9EC9D678245	0AD4F832-4C59-429C-B876-9A36DF7997EC	10	1	0	
8	E2DD137-D083-4F2F-B10E-4B56C1BD854C	1002AAAD-7F12-432F-A772-8381425EFF91	10	1	0	
9	D25DA718-D773-469C-B414-4249585EC426	39AD1CD9-94F3-439A-8C8B-3787C624DCC5	10	1	0	
10	F4C50ACF-36B0-4BCB-B4B0-425E5E87285B	8AA23558-28D4-4FF0-A9CA-20022B242A19	10	1	0	
11	FF853F3E-1900-48B4-ABC1-57803988C918	39AD1CD9-94F3-439A-8C8B-3787C624DCC5	11	1	0	
12	7E0ECF64-3DC6-4748-809D-1330252EC69C	8AA23558-28D4-4FF0-A9CA-20022B242A19	11	1	0	
13	6857BBAA-F701-475C-A4FC-24279E2CADE2	092CF057-6086-4426-BBB3-9A0622CFC511	11	1	0	
14	27710D01-4734-4FB0-B7BF-2E6B151FDC45	1002AAAD-7F12-432F-A772-8381425EFF91	11	1	0	
15	AB8F26BE-D5E4-4A35-8F4C-B54883A23A85	0AD4F832-4C59-429C-B876-9A36DF7997EC	11	1	0	
16	89500DD7-B48B-442B-B1A7-BCB1A98DEE11	8AA23558-28D4-4FF0-A9CA-20022B242A19	12	1	0	
17	51022F8E-3AF3-4C67-8164-C023C30073BC	39AD1CD9-94F3-439A-8C8B-3787C624DCC5	12	1	0	
18	CE311007-D4CE-4138-A668-8F6BFD600E0	092CF057-6086-4426-BBB3-9A0622CFC511	12	1	0	
19	9D15711C-25F5-499D-92F6-86332FFBA07F	0AD4F832-4C59-429C-B876-9A36DF7997EC	12	1	0	
20	DEE717B8-3916-45BC-BB55-09DDE4E0CC76	1002AAAD-7F12-432F-A772-8381425EFF91	12	1	0	
21	8E378F02-BC11-4C88-9F92-4E76122293EB	8AA23558-28D4-4FF0-A9CA-20022B242A19	13	1	0	
22	06D2072E-69A2-4F77-A858-475AB6EB407E	0AD4F832-4C59-429C-B876-9A36DF7997EC	13	1	0	
23	3DD96A17-4B74-401C-97A8-5E043429E4F5	092CF057-6086-4426-BBB3-9A0622CFC511	13	1	0	
24	FF27FC5-254D-444F-8D60-9280FE39989D	39AD1CD9-94F3-439A-8C8B-3787C624DCC5	13	1	0	

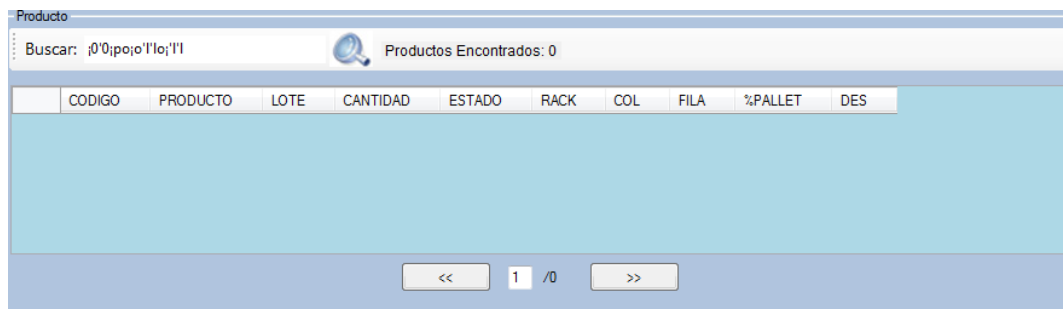
✔ Query executed successfully.

4.2.2. Prueba de Funcionalidad

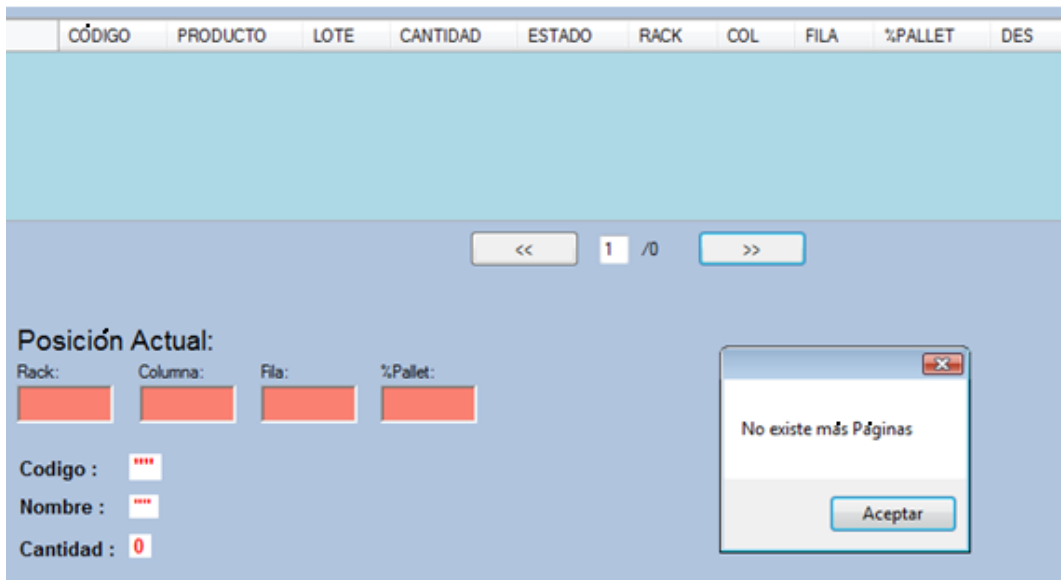
Ingreso fallido al sistema por clave incorrecta



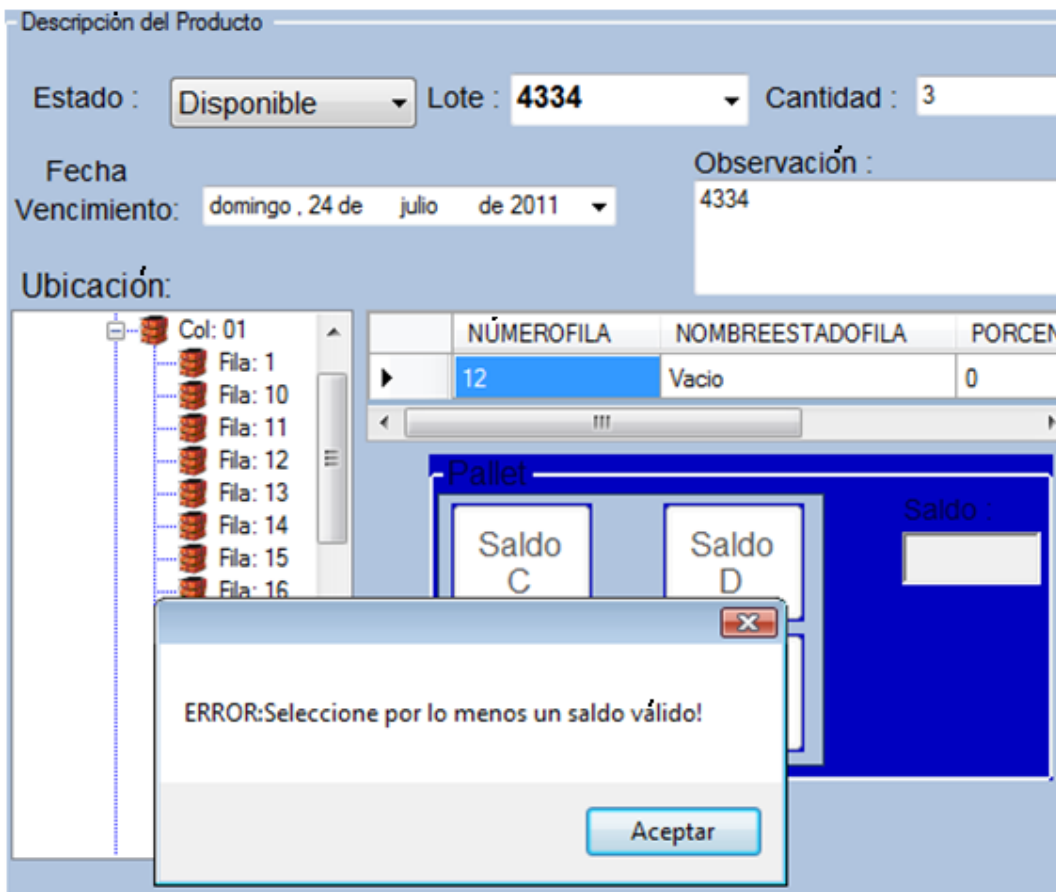
Se ingreso caracteres especiales en la búsqueda y el sistema los eliminó según la lista especificada de caracteres especiales no autorizados.



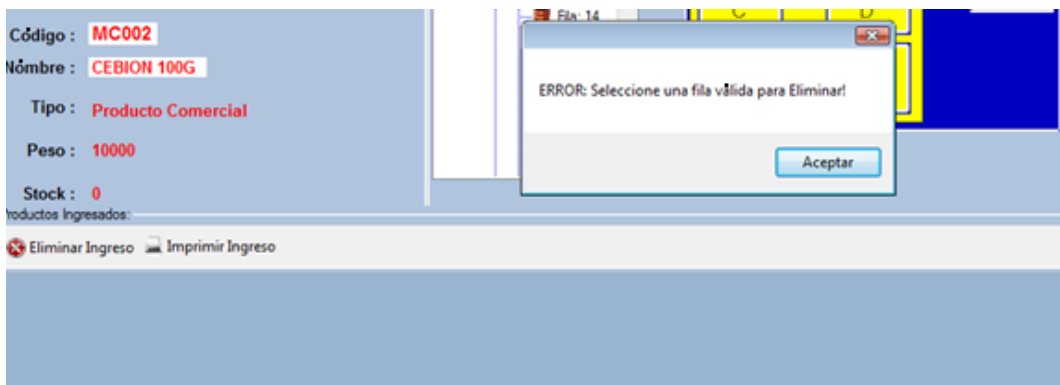
Se solicitaron más páginas de las existentes dentro de la búsqueda



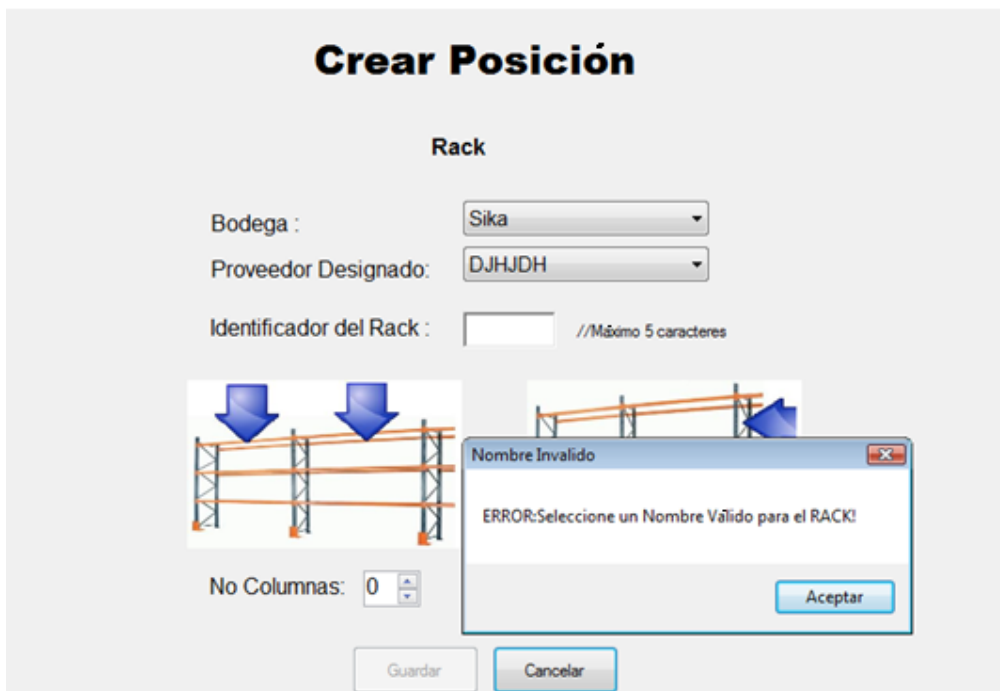
Al no seleccionar un saldo válido se niega el ingreso de datos



Al eliminar una columna inexistente el sistema alerta al usuario

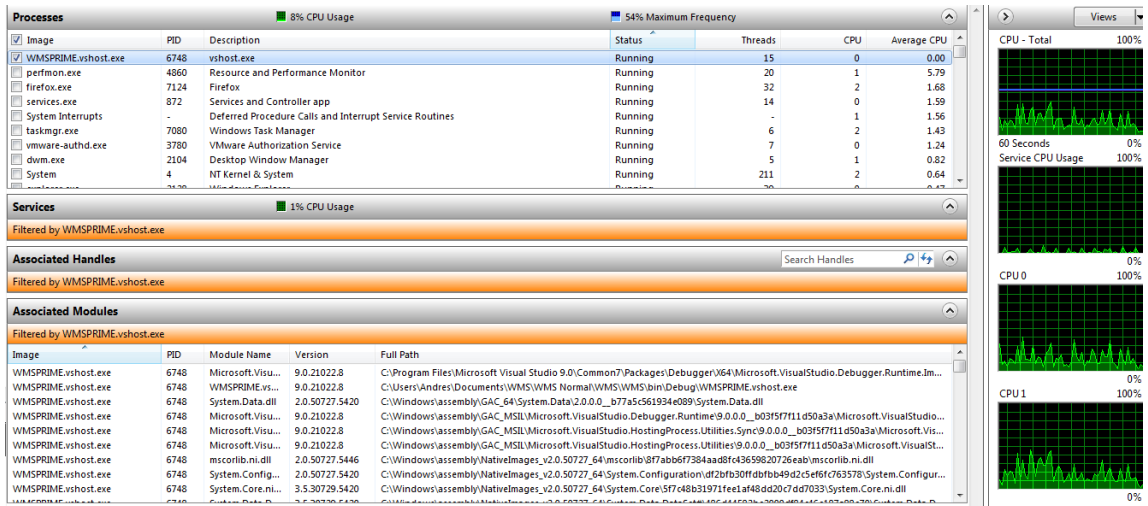


Quando se crea un rack sin nombre el sistema alerta al usuario y no le permite ingresar datos falsos

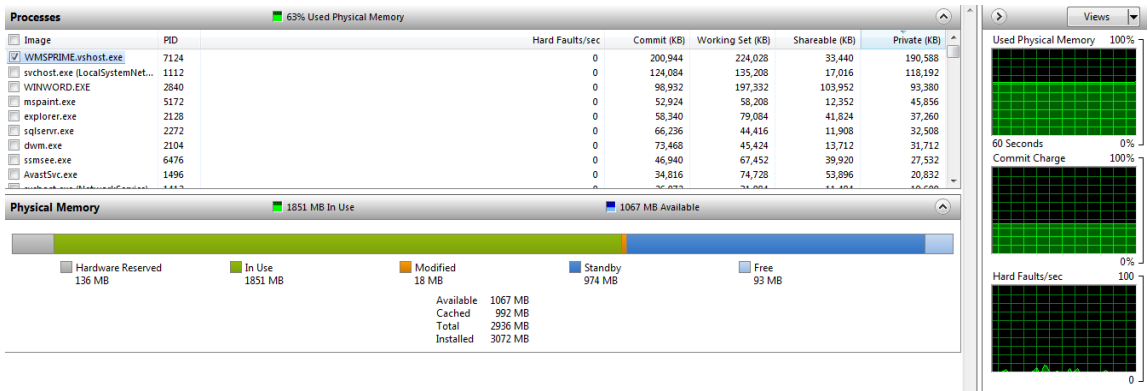


4.2.3. Reportes de Pruebas y Bugs

4.2.3.1. Pruebas en Core



4.2.3.2. Pruebas de Memoria



4.3. Código fuente de la Plataforma WMS

WMSPRIME

```
namespace WMSPRIME.Ciudad
{
    public partial class FormBuscarCiudad : Form
    {
        AdminCiudades admin = new AdminCiudades();
        String id;
        public FormBuscarCiudad()
        {
            InitializeComponent();
        }

        private void FormBuscarCiudad_Load(object sender, EventArgs e)
        {
            try
            {
                foreach (DataRow dr in
this.admin.RecuperarProvincias().Tables[0].Rows)
                {
this.comboBoxProvincia.Items.Add(dr["NombreProvincia"].ToString());
                }
            }
            catch (Exception ed)
            {
                MessageBox.Show("Error" + ed);
            }
        }

        private void dataGridBciudad_RowHeaderMouseClick(object
sender, DataGridViewCellEventArgs e)
        {
            if (e.RowIndex > -1)
            {
                this.richBoxNombre.Text =
this.dataGridBciudad.Rows[e.RowIndex].Cells["Ciudad"].Value.ToString()
;
                this.richBoxprovincia.Text =
this.dataGridBciudad.Rows[e.RowIndex].Cells["Provincia"].Value.ToStrin
g();
                id =
this.dataGridBciudad.Rows[e.RowIndex].Cells["idCiudad"].Value.ToString
();
            }
            else
            {
                MessageBox.Show("Seleccione una fila Valida!");
            }
        }
    }
}
```

```

private void buttonBuscar_Click(object sender, EventArgs e)
{
    Auxiliar aut = new Auxiliar();
    if (aut.autenticacion(richBoxbuscar.Text) == "")
    {
        MessageBox.Show("ERROR: Ingrese un patron de
Busqueda!", "Busqueda en Blanco");
    }
    else
    {
        dataGridBciudad.DataSource =
admin.BuscarCiudad(aut.autenticacion(richBoxbuscar.Text)).Tables[0];
        this.dataGridBciudad.Columns["idCiudad"].Visible =
false;
    }
}

private void buttonBProv_Click(object sender, EventArgs e)
{
    try
    {
        this.dataGridBciudad.DataSource =
admin.BuscarProvincia(this.comboBoxProvincia.SelectedItem.ToString()).
Tables[0];

        this.dataGridBciudad.Columns["idCiudad"].Visible =
false;

    }
    catch
    {
        MessageBox.Show("ERROR:Seleccione una Provincia");
    }
}

private void buttonElim_Click(object sender, EventArgs e)
{
    try
    {
        if (MessageBox.Show("Esta seguro de eliminar la ciudad
" + this.richBoxNombre.Text + "?", "Eliminar",
MessageBoxButtons.OKCancel, MessageBoxIcon.Exclamation) ==
DialogResult.OK)
        {
            admin.EliminarCiudad(id);
            MessageBox.Show("Eliminacion Correcta");
            this.dataGridBciudad.DataSource =
admin.RecuperarCiudades().Tables[0];
            this.dataGridBciudad.Columns["idCiudad"].Visible =
false;
        }
    }
    catch

```

```

        {
            MessageBox.Show("Error: No se puede eliminar !
Seleccione una Ciudad
Valida", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    private void buttonMod_Click(object sender, EventArgs e)
    {
        if (this.richBoxNombre.Text == "")
        {
            MessageBox.Show("ERROR: Seleccione una ciudad");
        }
        else
        {
            FormCrearCiudad crearciudad = new FormCrearCiudad();

            crearciudad.buttonModificar.Visible = true;
            crearciudad.EGuardarbutton.Visible = false;
            crearciudad.label2.Text = "Modificar Ciudad";
            crearciudad.richBoxNombre.Text =
this.richBoxNombre.Text;
            crearciudad.id = this.id;
            crearciudad.Text = "Modificar Ciudad";
            crearciudad.Show();
            this.Close();
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        this.Close();
    }
}

```

```

namespace WMSPRIME.Clientes
{
    public partial class FormCrearCliente : Form
    {
        Auxiliar auxi = new Auxiliar();
        public ECliente cliente = new ECliente();
        AdminCiudades aux2 = new AdminCiudades();
        AdminClientes admin = new AdminClientes();
        String[] cod = new String[100];
        String[] provincia = new String[30];

        public FormCrearCliente()
        {
            InitializeComponent();
        }

        private void EGuardarbutton_Click(object sender, EventArgs e)
        {
            try
            {

```

```

cliente.set2RucCliente(auxi.autenticacion(this.richBoxRuc.Text));

cliente.set3NombreCliente(auxi.autenticacion(this.richBoxNombre.Text));

cliente.set4idCiudad(cod[this.comboBoxCiudad.SelectedIndex]);

cliente.set5DireccionCliente(auxi.autenticacion(this.richBoxDireccion.Text));

cliente.set6telefonoCliente(auxi.autenticacion(this.richBoxTelefono.Text));

cliente.set7Representate(auxi.autenticacion(this.richBoxRepresenten.Text));

        admin.InsertarCliente(cliente);
        MessageBox.Show("Ingreso Exitoso");
        this.Close();
    }
    catch
    {
        MessageBox.Show("Error: Ingrese datos Validos !",
"ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void FormCrearCliente_Load(object sender, EventArgs e)
{
    int i = 0;
    foreach (DataRow dr in
this.aux2.RecuperarProvincias().Tables[0].Rows)
    {

this.comboBoxProvincia.Items.Add(dr["NombreProvincia"].ToString());
        provincia[i] = dr["idProvincia"].ToString();
        i++;
    }
    comboBoxCiudad.Visible = true;
    PCiudad.Visible = true;
    this.richBoxRuc.Text = cliente.get2RucCliente();
    this.richBoxNombre.Text = cliente.get3NombreCliente();
    this.richBoxTelefono.Text = cliente.get6telefonoCliente();
    this.richBoxDireccion.Text =
cliente.get5DireccionCliente();
    this.richBoxRepresenten.Text = cliente.get7Representate();

}

private void ECancelbutton_Click(object sender, EventArgs e)
{
    this.Close();
}

private void boton1_Click(object sender, EventArgs e)
{
    cliente.set2RucCliente(auxi.autenticacion(this.richBoxRuc.Text));

```

```

cliente.set3NombreCliente (aux1.autenticacion(this.richBoxNombre.Text
));

cliente.set4idCiudad (cod[this.comboBoxCiudad.SelectedIndex]);

cliente.set5DireccionCliente (aux1.autenticacion(this.richBoxDireccion.
Text));

cliente.set6telefonoCliente (aux1.autenticacion(this.richBoxTelefono.
Text));

cliente.set7Representate (aux1.autenticacion(this.richBoxRepresenten. Tex
t));
        admin.UpdateCliente (cliente);
        MessageBox.Show ("Modificacion Exitosa");
        this.Close ();
    }

    private void richBoxRuc_KeyPress (object sender,
KeyPressEventArgs e)
    {
        if (char.IsDigit (e.KeyChar) || char.IsControl (e.KeyChar))
        {
            e.Handled = false;
        }
        else
        {
            e.Handled = true;
        }
    }

    private void comboBoxProvincia_SelectedIndexChanged (object
sender, EventArgs e)
    {
        comboBoxCiudad.Items.Clear ();
        int i = 0;
        foreach (DataRow dr in
this.aux2.RecuperarCiudades (provincia[comboBoxProvincia.SelectedIndex]
).Tables[0].Rows)
        {

this.comboBoxCiudad.Items.Add (dr["Ciudad"].ToString ());
            cod[i] = dr["idCiudad"].ToString ();
            i++;
        }
    }
}
namespace WMSPRIME.Empleado
{
    public partial class FormCrearEmpleado : Form
    {
        AdminProvedores aux = new AdminProvedores ();
        AdminEmpleado aux2 = new AdminEmpleado ();
        String[] cod = new String[1000];
        public EEmpleado emp = new EEmpleado ();
        Auxiliar aut = new Auxiliar ();
    }
}

```

```

public FormCrearEmpleado()
{
    InitializeComponent();
}

private void FormCrearEmpleado_Load(object sender, EventArgs
e)
{
    try
    {
        int i = 0;
        foreach (DataRow dr in
this.aux.RecuperarProvedores().Tables[0].Rows)
        {

this.ProveedorcomboBox.Items.Add(dr["Ruc"].ToString() + " " +
dr["Nombre"].ToString());
            cod[i] = dr["idProveedor"].ToString();
            i++;
        }
        this.richBoxCelular.Text = emp.get7Celular();
        this.richBoxClave.Text = emp.get4Clave();
        this.richBoxmail.Text = emp.get6Correo();
        this.richBoxnombre.Text = emp.get1Nombre();
        this.richBoxUsuario.Text = emp.get3Usuario();

    }
    catch (Exception ed)
    {
        MessageBox.Show("Seleccione un Proveedor" + ed);
    }
}

private void EGuardarbutton_Click(object sender, EventArgs e)
{
    try
    {
emp.set1Nombre (aut.autenticacion(this.richBoxnombre.Text));
emp.set2idProveedor (cod[this.ProveedorcomboBox.SelectedIndex]);
emp.set3Usuario (aut.autenticacion(this.richBoxUsuario.Text));
emp.set4Clave (aut.autenticacion(this.richBoxClave.Text));
        if (radioButton1.Checked)
        {
            emp.set5idNivelAcceso(1);
        }
        if (radioButton2.Checked)
        {
            emp.set5idNivelAcceso(2);
        }
        if (radioButton3.Checked)
        {
            emp.set5idNivelAcceso(3);
        }
    }
}

```



```

        if (radioButton4.Checked)
        {
            emp.set5idNivelAcceso(4);
        }
        if (radioButton5.Checked)
        {
            emp.set5idNivelAcceso(5);
        }

emp.set6Correo (aut.autenticacion(this.richBoxmail.Text));

emp.set7Celular (aut.autenticacion(this.richBoxCelular.Text));
aux2.InsertaEmpleado (emp);

        MessageBox.Show("INGRESO CORRECTO");
        this.Close();

    }
    catch
    {
        MessageBox.Show("Error Ingrese datos correctos" ,
        "Ingreso Fallido", MessageBoxButtons.OK);
    }
}

private void ECancelbutton_Click(object sender, EventArgs e)
{
    this.Close();
}

private void buttonG_Click(object sender, EventArgs e)
{
    try
    {

emp.set1Nombre (aut.autenticacion(this.richBoxnombre.Text));

emp.set2idProveedor (cod[this.ProveedorcomboBox.SelectedIndex]);

emp.set3Usuario (aut.autenticacion(this.richBoxUsuario.Text));

emp.set4Clave (aut.autenticacion(this.richBoxClave.Text));
        if (radioButton1.Checked)
        {
            emp.set5idNivelAcceso(1);
        }
        if (radioButton2.Checked)
        {
            emp.set5idNivelAcceso(2);
        }
        if (radioButton3.Checked)
        {
            emp.set5idNivelAcceso(3);
        }
        if (radioButton4.Checked)
        {
            emp.set5idNivelAcceso(4);
        }
        if (radioButton5.Checked)

```

```

        {
            emp.set5idNivelAcceso(5);
        }

emp.set6Correo (aut.autenticacion(this.richBoxmail.Text));

emp.set7Celular (aut.autenticacion(this.richBoxCelular.Text));
    aux2.ActualizarEmpleado (emp);
    MessageBox.Show("Modificacion Exitosa");
    this.Close();
    }
    catch
    {
        MessageBox.Show("ERROR: Modificacion Fallida");
    }
}

private void richBoxCelular_KeyPress (object sender,
KeyPressEventArgs e)
{
    if (char.IsDigit(e.KeyChar) || char.IsControl(e.KeyChar))
    {
        e.Handled = false;
    }
    else
    {
        e.Handled = true;
    }
}
}
}

```

```

namespace WMSPRIME.Ingreso
{
    public partial class FormIngreso : Form
    {
        Auxiliar auxi = new Auxiliar();
        AdminPosiciones aux = new AdminPosiciones();
        AdminProvedores prov = new AdminProvedores();
        AdminProducto prod = new AdminProducto();
        EProducto producto = new EProducto();
        EPosicion posicioningreso = new EPosicion();
        EPosicion posicion = new EPosicion();
        String[] cod = new String[100];
        String[] prove = new String[1000];
        String[] estadopr = new String[10];
        DataTable DTfila = new DataTable();
        DataTable DTingreso = new DataTable();
        DataTable tabla = new DataTable();
        Auxiliar aut = new Auxiliar();
        public String idProv;
        public String idUser;
        public String NivelAc;
        int numero = 0;
        int row = -1;
        String comboid;
        String idproducto;
        String idDetallePro;
    }
}

```

```

public FormIngreso()
{
    InitializeComponent();
}
public void CensoBus()
{
    this.dataGridBProv.Columns["idTipoProducto"].Visible =
false;
    this.dataGridBProv.Columns["idProducto"].Visible = false;
    this.dataGridBProv.Columns["idProveedor"].Visible = false;
    this.dataGridBProv.Columns["Tipo"].Visible = false;
    this.dataGridBProv.Columns["Peso"].Visible = false;
    this.dataGridBProv.Columns["Proveedor"].Visible = false;
}
public void Censo()
{
    // this.dataGridPosi.Columns["idColumna"].Visible = false;
    this.dataGridPosi.Columns["idFila"].Visible = false;
    this.dataGridPosi.Columns["idEstadoFila"].Visible = false;
    this.dataGridPosi.Columns["idEstadoFila"].Visible = false;
}
public void CensoIngreso()
{
    this.dataGridIngreso.Columns["idDetalleProducto"].Visible
= false;
    this.dataGridIngreso.Columns["idFila"].Visible = false;
    this.dataGridIngreso.Columns["idProducto"].Visible =
false;
    this.dataGridIngreso.Columns["SaldoUtilizado"].Visible =
false;
}
public EPosicion FillPos(EPosicion Epos)
{
    switch (Epos.getSaldo().Length)
    {
        case 0:
            Epos.setEstadoFila("1");
            Epos.setPUtilizado("0");

            break;

        case 1:

            Epos.setEstadoFila("2");
            Epos.setPUtilizado("25");

            break;

        case 2:
            Epos.setEstadoFila("3");
            Epos.setPUtilizado("50");

            break;

        case 3:
            Epos.setEstadoFila("4");
            Epos.setPUtilizado("75");

            break;
    }
}

```

```

        case 4:
            Epos.setEstadoFila("5");
            Epos.setPUtilizado("100");

            break;
        }
        return Epos;
    }
    public void desactivar(String aux)
    {
        if (aux == "")
        {
            this.buttonall.BackColor = Color.LightSteelBlue;
            this.buttonA.BackColor = Color.White;
            this.buttonB.BackColor = Color.White;
            this.buttonC.BackColor = Color.White;
            this.buttonD.BackColor = Color.White;
            buttonA.Enabled = true;
            buttonB.Enabled = true;
            buttonC.Enabled = true;
            buttonD.Enabled = true;
        }
        else
        {
            for (int index = 0; index < aux.Length; index++)
            {

                switch (aux[index])
                {

                    case 'A':
                        buttonA.BackColor = Color.Red;
                        buttonA.Enabled = false;

                        break;
                    case 'B':
                        buttonB.BackColor = Color.Red;
                        buttonB.Enabled = false;
                        break;
                    case 'C':
                        buttonC.BackColor = Color.Red;
                        buttonC.Enabled = false;
                        break;
                    case 'D':
                        buttonD.BackColor = Color.Red;
                        buttonD.Enabled = false;
                        break;
                    default:

                        break;

                }
            }
        }
    }
}
private void fillDataGrid_dtgBrowse()
{
    DataTable tablaAux = new DataTable();

```

```

try
{
    Cursor = Cursors.WaitCursor;
    tablaAux.Columns.Add("idProducto");
    tablaAux.Columns.Add("idProveedor");
    tablaAux.Columns.Add("idTipoProducto");
    tablaAux.Columns.Add("codigo");
    tablaAux.Columns.Add("Producto");
    tablaAux.Columns.Add("Proveedor");
    tablaAux.Columns.Add("Tipo");
    tablaAux.Columns.Add("Peso");
    tablaAux.Columns.Add("Stock");

    this.dataGridBProv.DataSource = tablaAux;

    for (int i = 0; i < 50; i++)
    {
        if (numero < tabla.Rows.Count)
        {
            tablaAux.LoadDataRow(tabla.Rows[numero].ItemArray, false);
            numero++;
        }
        else
        {
            break;
        }
    }

    }
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    Cursor = Cursors.Default;
}
}

private void comboBoxBodega_SelectedIndexChanged(object
sender, EventArgs e)
{
    groupBoxproducto.Visible = true;

    try
    {
        comboBoxProve.Items.Clear();

        int i = 0;
        foreach (DataRow dr in
this.prov.RecuperarProvedores().Tables[0].Rows)
        {

            this.comboBoxProve.Items.Add(dr["Nombre"].ToString());
            prove[i] = dr["idProveedor"].ToString();
            i++;
        }
    }
}

```

```

    }
    if (int.Parse(NivelAc) > 3)
    {
        this.comboBoxProve.Visible = true;
        label3.Visible = true;
    }
}
catch (Exception ed)
{
    MessageBox.Show("Error" + ed);
}
}

private void FormIngreso_Load(object sender, EventArgs e)
{
    try
    {
        int i = 0;
        foreach (DataRow dr in
this.aux.RecuperarBodega().Tables[0].Rows)
        {

this.comboBoxBodega.Items.Add(dr["NombreBodega"].ToString());
            cod[i] = dr["idBodega"].ToString();
            i++;
        }
        i = 0;
        foreach (DataRow dr in
this.prod.RecuperarEstadoProductos().Tables[0].Rows)
        {

this.comboBoxEstado.Items.Add(dr["Nombre"].ToString());
            estadopr[i] = dr["Codigo"].ToString();
            i++;
        }
    }
    catch (Exception ed)
    {
        MessageBox.Show("Error" + ed);
    }
}

private void comboBoxProve_SelectedIndexChanged(object sender,
EventArgs e)
{
    idProv= prove[this.comboBoxProve.SelectedIndex];
    dataGridBProv.Columns.Clear();
    groupBox1.Visible = false;
}

private void dataGridBProv_RowHeaderMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
{
    treeView1.Nodes.Clear();
    TreeNode nodopadre = new TreeNode("Racks");
    treeView1.Nodes.Add(nodopadre);
    groupBox1.Visible = true;
}

```

```

        comboBoxLote.Text = "";
        textBoxCantidad.Text = "0";
        if (e.RowIndex > -1)
        {
            foreach (DataRow dr in
this.aux.RecuperarRack(cod[this.comboBoxBodega.SelectedIndex], this.idP
rov).Tables[0].Rows)
            {
                TreeNode newrack = new
TreeNode(dr["Nombre"].ToString());
                foreach (DataRow dc in
this.aux.RecuperarColumnas(dr["idRack"].ToString(),
this.idProv).Tables[0].Rows)
                {
                    TreeNode newcolumn = new TreeNode("Col:
"+dc["NumeroColumna"].ToString());

                    //newcolumn.ImageIndex = imageList1.Images[1];
                    foreach (DataRow df in
this.aux.RecuperarFilasEstado(dr["idRack"].ToString(), dc["idColumna"].
ToString(), idProv).Tables[0].Rows)
                    {
                        TreeNode newfila = new TreeNode("Fila: "
+df["NumeroFila"].ToString());
                        newfila.Tag = df["idFila"].ToString();
                        newcolumn.Nodes.Add(newfila);

                    }
                    newrack.Nodes.Add(newcolumn);
                }
                nodopadre.Nodes.Add(newrack);
            }

            try
            {

                producto.set1idProducto(this.dataGridBProv.Rows[e.RowIndex].Cells["idP
roducto"].Value.ToString());

                producto.set2idProveedor(this.dataGridBProv.Rows[e.RowIndex].Cells["id
Proveedor"].Value.ToString());

                producto.set3idTipoProducto(int.Parse(this.dataGridBProv.Rows[e.RowInd
ex].Cells["idTipoProducto"].Value.ToString()));

                producto.set4codigo(this.dataGridBProv.Rows[e.RowIndex].Cells["codigo"
].Value.ToString());
                this.labelCodigo.Text = producto.get4codigo();

                producto.set5NombreProducto(this.dataGridBProv.Rows[e.RowIndex].Cells[
"Producto"].Value.ToString());
                this.labelNombre.Text = producto.get5NombreProducto();

                producto.set6Peso(float.Parse(this.dataGridBProv.Rows[e.RowIndex].Cell
s["Peso"].Value.ToString()));
                this.labelPeso.Text = producto.get6Peso().ToString();

                producto.set7StockTotal(int.Parse(this.dataGridBProv.Rows[e.RowIndex].
Cells["Stock"].Value.ToString()));
            }
        }
    }
}

```

```

        this.labelStock.Text =
producto.get7StockTotal().ToString();
        this.labeltipo.Text =
this.dataGridBProv.Rows[e.RowIndex].Cells["Tipo"].Value.ToString();

        comboBoxLote.Items.Clear();
        foreach (DataRow dr in
this.prod.RecuperarLotes(producto.get1idProducto()).Tables[0].Rows)
        {

this.comboBoxLote.Items.Add(dr["Lote"].ToString());

        }
    }
    catch
    {
        MessageBox.Show("ERROR: Conexion Fallida!");
    }
}

private void treeView1_NodeMouseDoubleClick(object sender,
TreeNodeMouseClickEventArgs e)
{
    if (e.Node.Tag == null)
    {
        MessageBox.Show("Seleccione una fila valida");
    }
    else
    {
        groupBox3.Enabled = true;
        this.dataGridPosi.DataSource =
aux.RecuperarFila(e.Node.Tag.ToString()).Tables[0];
        Censo();
        String a =
dataGridPosi.Rows[0].Cells["Saldo"].Value.ToString();
        this.richBoxsaldo.Text = "";
        desactivar(a);

    }
}

private void buttonC_Click(object sender, EventArgs e)
{
    if (buttonC.Enabled)
    {
        if (buttonC.BackColor == Color.Yellow)
        {
            this.buttonC.BackColor = Color.White;
            String a = richBoxsaldo.Text;

            richBoxsaldo.Text = Regex.Replace(a, "C", "");
        }
        else
        {
            this.buttonC.BackColor = Color.Yellow;
            richBoxsaldo.Text = richBoxsaldo.Text + "C";
        }
    }
}

```



```

    }
}

private void buttonA_Click(object sender, EventArgs e)
{
    if (buttonA.Enabled)
    {
        if (buttonA.BackColor == Color.Yellow)
        {
            this.buttonA.BackColor = Color.White;
            String a = richBoxsaldo.Text;

            richBoxsaldo.Text = Regex.Replace(a, "A", "");
        }
        else
        {
            this.buttonA.BackColor = Color.Yellow;
            richBoxsaldo.Text = richBoxsaldo.Text + "A";
        }
    }
}

private void buttonB_Click(object sender, EventArgs e)
{
    if (buttonB.Enabled)
    {
        if (buttonB.BackColor == Color.Yellow)
        {
            this.buttonB.BackColor = Color.White;
            String a = richBoxsaldo.Text;

            richBoxsaldo.Text = Regex.Replace(a, "B", "");
        }
        else
        {
            this.buttonB.BackColor = Color.Yellow;
            richBoxsaldo.Text = richBoxsaldo.Text + "B";
        }
    }
}

private void buttonD_Click(object sender, EventArgs e)
{
    if (buttonD.Enabled)
    {
        if (buttonD.BackColor == Color.Yellow)
        {
            this.buttonD.BackColor = Color.White;
            String a = richBoxsaldo.Text;

            richBoxsaldo.Text = Regex.Replace(a, "D", "");
        }
        else
        {
            this.buttonD.BackColor = Color.Yellow;

```

```

        richBoxsaldo.Text = richBoxsaldo.Text + "D";
    }
}

private void buttonall_Click(object sender, EventArgs e)
{
    if (buttonall.Enabled)
    {
        if (buttonA.Enabled && buttonB.Enabled &&
buttonC.Enabled && buttonD.Enabled)
        {
            if (this.buttonall.BackColor == Color.Yellow)
            {
                this.buttonall.BackColor =
Color.LightSteelBlue;
                this.buttonA.BackColor = Color.White;
                this.buttonB.BackColor = Color.White;
                this.buttonC.BackColor = Color.White;
                this.buttonD.BackColor = Color.White;
                buttonA.Enabled = true;
                buttonB.Enabled = true;
                buttonC.Enabled = true;
                buttonD.Enabled = true;
                String a = richBoxsaldo.Text;

                richBoxsaldo.Text = Regex.Replace(a, "ABCD",
""");
            }
            else
            {
                this.buttonall.BackColor = Color.Yellow;
                this.buttonA.BackColor = Color.Yellow;
                this.buttonB.BackColor = Color.Yellow;
                this.buttonC.BackColor = Color.Yellow;
                this.buttonD.BackColor = Color.Yellow;
                buttonA.Enabled = false;
                buttonB.Enabled = false;
                buttonC.Enabled = false;
                buttonD.Enabled = false;
                richBoxsaldo.Text = "ABCD";
            }
        }
        else
        {
            MessageBox.Show("ERROR: No puede seleccionar todo
el pallet, Saldos Utilizados!");
        }
    }
}

private void textBox1_KeyPress(object sender,
KeyPressEventArgs e)
{
    if (char.IsDigit(e.KeyChar) || char.IsControl(e.KeyChar))
    {
        e.Handled = false;
    }
}

```

```

        }
        else
        {
            e.Handled = true;
        }
    }

    private void toolStripButton1_Click(object sender, EventArgs
e)
    {
        //boton eliminar ingreso
    }

    private void dataGridIngreso_RowHeaderMouseClick(object
sender, DataGridViewCellEventArgs e)
    {
        row = e.RowIndex;
        if (row > -1)
        {
            try
            {
                idproducto =
this.dataGridIngreso.Rows[row].Cells["idProducto"].Value.ToString();
                idDetallePro =
this.dataGridIngreso.Rows[row].Cells["idDetalleProducto"].Value.ToStri
ng();

                posicioningreso.setidFila(this.dataGridIngreso.Rows[row].Cells["idFila
"].Value.ToString());
                String saldopro =
this.dataGridIngreso.Rows[row].Cells["SaldoProducto"].Value.ToString()
.TrimEnd();
                String saldopos =
this.dataGridIngreso.Rows[row].Cells["SaldoUtilizado"].Value.ToString(
).TrimEnd();

                for (int i = 0; i < saldopro.Length; i++)
                {
                    saldopos=
Regex.Replace(saldopos, (saldopro[i]).ToString(), "");
                }
                posicioningreso.setSaldo(saldopos);
                posicioningreso = FillPos(posicioningreso);

            }
            catch
            {

            }

        }
        else
        {
        }
    }

```

```

    }

    private void dataGridBProv_RowHeaderMouseClick(object sender,
DataGridViewCellEventArgs e)
    {

        if (e.RowIndex > -1)
        {

            producto.setIidProducto(this.dataGridBProv.Rows[e.RowIndex].Cells["idP
roducto"].Value.ToString());

            producto.setAidDetalleProducto(this.dataGridBProv.Rows[e.RowIndex].Cel
ls["idDetalleProducto"].Value.ToString());

            producto.setFidFila(this.dataGridBProv.Rows[e.RowIndex].Cells["idFila"
].Value.ToString());

            producto.set2idProveedor(this.dataGridBProv.Rows[e.RowIndex].Cells["id
Proveedor"].Value.ToString());
                this.labelCodigo.Text =
this.dataGridBProv.Rows[e.RowIndex].Cells["Codigo"].Value.ToString();
                this.labelNombre.Text =
this.dataGridBProv.Rows[e.RowIndex].Cells["Producto"].Value.ToString()
;
                this.labelCantidad.Text =
this.dataGridBProv.Rows[e.RowIndex].Cells["Cantidad"].Value.ToString()
;

                richBoxRack.Text =
this.dataGridBProv.Rows[e.RowIndex].Cells["Rack"].Value.ToString();
                richBoxCol.Text =
this.dataGridBProv.Rows[e.RowIndex].Cells["Col"].Value.ToString();
                richBoxFila.Text =
this.dataGridBProv.Rows[e.RowIndex].Cells["Fila"].Value.ToString();
                richBoxPallet.Text =
this.dataGridBProv.Rows[e.RowIndex].Cells["%Pallet"].Value.ToString();

//posicionanterior.setidFila(dataGridBProv.Rows[e.RowIndex].Cells["idF
ila"].Value.ToString());
                //posicionanterior.setSaldo(richBoxPallet.Text);

        }
        else
        {
            MessageBox.Show("Seleccione una fila Valida");
        }
    }

    private void boton1_Click(object sender, EventArgs e)
    {
        if (textBoxCantidad.Text.Length > 9 ||
int.Parse(labelCantidad.Text) < int.Parse(textBoxCantidad.Text) ||
textBoxCantidad.Text == "0" || textBoxCantidad.Text == "")
        {
            MessageBox.Show("Error: Seleccione una cantidad igual
o menor a la disponible!", "ERROR", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }
}

```

```

    }
    else
    {
        if (textBoxPedido.Text == "")
        {
            MessageBox.Show("Error: Ingrese un numero de
Pedido Valido!", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        else
        {
            try
            {

                pick.set2idProducto(producto.get1idProducto());

                pick.set3idDetallePro(producto.getAidDetalleProducto());
                pick.set4idEmpleado(idUser);
                pick.set5FechaReserva(DateTime.Now);

                pick.set6CantidadReserva(int.Parse(textBoxCantidad.Text));

                pick.set7NumeroPedido(aut.autenticacion(textBoxPedido.Text));
                pick.set8idProveedor(producto.get2idProveedor());
                Adminpick.IngresarReserva(pick,
int.Parse(labelCantidad.Text));
                MessageBox.Show("Ingreso Exitoso");
                tabla2 = Adminpick.BuscarReservaEmp(idUser,
aut.autenticacion(textBoxPedido.Text));
                dataGridReserva.DataSource = tabla2;
                Censo();
                dataGridBProv.DataSource = "";
                textBoxCantidad.Text = "0";

            }
            catch
            {
                MessageBox.Show("Error: Seleccione un producto
Valido !", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}

private void textBoxCantidad_KeyPress(object sender,
KeyPressEventArgs e)
{
    if (char.IsDigit(e.KeyChar) || char.IsControl(e.KeyChar))
    {
        e.Handled = false;
    }
    else
    {
        e.Handled = true;
    }
}

private void dataGridReserva_RowHeaderMouseClick(object
sender, DataGridViewCellEventArgs e)
{

```

```

        rowIndex = e.RowIndex;
        if (rowIndex > -1)
        {
            try
            {
pick.set1idReserva (this.dataGridReserva.Rows[rowIndex].Cells["idReserva"].Value.ToString());

pick.set2idProducto (this.dataGridReserva.Rows[rowIndex].Cells["idProducto"].Value.ToString());

pick.set3idDetallePro (this.dataGridReserva.Rows[rowIndex].Cells["idDetalleProducto"].Value.ToString());

pick.set6CantidadReserva (int.Parse (this.dataGridReserva.Rows[rowIndex].Cells["Cantidad Reservada"].Value.ToString()));

            }
            catch
            {
                MessageBox.Show("Error: Seleccione una fila Valida !", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }

    private void tSButtonEliminar_Click(object sender, EventArgs e)
    {
        if (rowIndex > -1)
        {
            Adminpick.RestablecerReserva (pick);
            MessageBox.Show("Eliminacion Correcta !", "OK", MessageBoxButtons.OK);
            tabla = Adminpick.BuscarReservaEmp (idUser, aut.autenticacion(textBoxPedido.Text));
            dataGridReserva.DataSource = tabla;
            Censo();
            rowIndex = -1;
        }
        else
        {
            MessageBox.Show("Error: No se Puede Eliminar, elija un fila validad !", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    private void toolStripButton6_Click(object sender, EventArgs e)
    {
        FormPicking NewPick = new FormPicking();
        NewPick.idProv = idProv;
        NewPick.idUser = idUser;
        NewPick.NivelAc = NivelAc;
        NewPick.DTpick = tabla2;
    }
}

```

```

        NewPick.Show();
    }

    private void tSComboBoxPedido_SelectedIndexChanged(object
sender, EventArgs e)
    {
        try
        {
            tabla2 = Adminpick.BuscarReservaEmp(idUser,
tSComboBoxPedido.SelectedItem.ToString());
            dataGridReserva.DataSource = tabla2;
            Censo();
        }
        catch
        {
            MessageBox.Show("Error: Seleccione un # de Pedido
Valido !", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    private void button4_Click(object sender, EventArgs e)
    {
        if (tabla.Rows.Count / 50 < 1)
        {
            MessageBox.Show("No existe mas Paginas");
        }
        else
        {
            if (numero < tabla.Rows.Count)
            {
                button3.Enabled = true;
                fillDataGrid_dtgBrowse();
                Censo();
                if ((numero % 50) == 0)
                {
                    textBoxPag.Text = (numero / 50).ToString();
                }
                else
                {
                    textBoxPag.Text = ((numero / 50) +
1).ToString();
                }
            }
            else
            {
                button4.Enabled = false;
            }
        }
    }

    private void button3_Click(object sender, EventArgs e)
    {

```

```

        if (numero % 50 != 0)
        {
            numero = (numero - (numero % 50));
        }
        if (numero > 0)
        {
            numero = numero - 100;
            if (numero < 0)
            {
                numero = 0;
            }
            button4.Enabled = true;
            fillDataGrid_dtgBrowse();
            Censo();

            if ((numero % 50) == 0)
            {
                textBoxPag.Text = (numero / 50).ToString();
            }
            else
            {
                textBoxPag.Text = ((numero / 50) + 1).ToString();
            }

            if (numero < 0)
            {
                numero = 0;
            }

        }
        else
        {
            button3.Enabled = false;
        }
    }

    private void toolStripButton2_Click(object sender, EventArgs
e)
    {
        if (dataGridReserva.Rows.Count > 0)
        {
            String msg;
            this.saveFileDialog1.DefaultExt = "xls";
            saveFileDialog1.Filter = "XLS file (*.xls)|*.xls|All
files (*.*)|*.*";
            saveFileDialog1.RestoreDirectory = true;
            saveFileDialog1.Title = "GUARDAR REPORTE?";
            saveFileDialog1.InitialDirectory = @"C:/";
            if (saveFileDialog1.ShowDialog() !=
DialogResult.Cancel)
            {
                this.Cursor = Cursors.WaitCursor;
                msg = aut.AbrirConsultaExcel(this.dataGridReserva,
saveFileDialog1.FileName, "A5", "ReporteOrdenes");
                this.Cursor = Cursors.Default;
                MessageBox.Show(msg);
            }
        }
    }
}

```



```

        else
        {
            MessageBox.Show("ERROR: No existe informacion!");
        }
    }
}

namespace WMSPRIME.Ordenes
{
    public partial class FormPicking : Form
    {
        AdminPicking Adminpick = new AdminPicking();
        AdminClientes cli = new AdminClientes();
        ECliente cliente = new ECliente();
        EPicking pick= new EPicking();
        Auxiliar aut = new Auxiliar();
        DataTable tabla = new DataTable();
        //int rowIndex = -1;
        public String idProv;
        public String idUser;
        public String NivelAc;
        public DataTable DTpick = new DataTable();
        public FormPicking()
        {
            InitializeComponent();
        }

        private void toolStripButton5_Click(object sender, EventArgs
e)
        {
            if (TSTextBoxBuscar.Text == "")
            {
                MessageBox.Show("Ingrese una busqueda Valida");
            }
            else
            {
                try
                {
                    this.dataGridBProv.DataSource =
cli.BuscarCliente(aut.autenticacion(this.TSTextBoxBuscar.Text)).Tabl
es[0];
                    this.dataGridBProv.Columns["id"].Visible = false;
                    this.dataGridBProv.Columns["idCiudad"].Visible =
false;
                    //this.dataGridBProv.Columns["Ciudad"].Visible =
false;
                    //this.dataGridBProv.Columns["Provincia"].Visible
= false;
                    //this.dataGridBProv.Columns["Telefono"].Visible =
false;

                }
                catch

```

```

        {
            MessageBox.Show("Error: Ingrese Caracteres Validos
!", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

private void FormPicking_Load(object sender, EventArgs e)
{
    this.dataGridReserva.DataSource = DTpick;
    if (DTpick.Rows.Count > 0)
    {
        this.dataGridReserva.Columns["idReserva"].Visible =
false;
        this.dataGridReserva.Columns["idProducto"].Visible =
false;
        this.dataGridReserva.Columns["idEmpleado"].Visible =
false;
        this.dataGridReserva.Columns["idDetalleProducto"].Visible = false;
    }
}

private void dataGridBProv_RowHeaderMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
{
    if (e.RowIndex > -1)
    {
        cliente.set1idCliente(this.dataGridBProv.Rows[e.RowIndex].Cells["id"].
Value.ToString());

        cliente.set2RucCliente(this.dataGridBProv.Rows[e.RowIndex].Cells["Ruc"
].Value.ToString());
        richBoxRuc.Text = cliente.get2RucCliente();

        cliente.set3NombreCliente(this.dataGridBProv.Rows[e.RowIndex].Cells["N
ombre"].Value.ToString());
        richBoxNomCli.Text = cliente.get3NombreCliente();

        cliente.set4idCiudad(this.dataGridBProv.Rows[e.RowIndex].Cells["idCiud
ad"].Value.ToString());
        richBoxCiu.Text =
this.dataGridBProv.Rows[e.RowIndex].Cells["Ciudad"].Value.ToString();

        cliente.set5DireccionCliente(this.dataGridBProv.Rows[e.RowIndex].Cells
["Direccion"].Value.ToString());
        richBoxDir.Text = cliente.get5DireccionCliente();

        cliente.set6telefonoCliente(this.dataGridBProv.Rows[e.RowIndex].Cells[
"telefono"].Value.ToString());

        cliente.set7Representate(this.dataGridBProv.Rows[e.RowIndex].Cells["Re
presentante"].Value.ToString());
        groupBox1.Visible = true;
    }
}
}

```

```

        private void tsButtonEliminar_Click(object sender, EventArgs
e)
        {
            if (MessageBox.Show("Esta Seguro Eliminar este Picking?",
"Advertencia", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning) ==
DialogResult.OK)
            {
                this.Close();
            }
        }

        private void toolStripButton6_Click(object sender, EventArgs
e)
        {
            try
            {
                if (DTpick.Rows.Count > 0)
                {
                    if (numericUpDown1.Value > 0)
                    {
                        FormOrdenNueva ord = new FormOrdenNueva();
                        System.Guid idDeta = System.Guid.NewGuid();
                        pick.set1idPicking(idDeta.ToString());
                        pick.set2idProveedores(idProv);
                        pick.set3idCliente(cliente.get1idCliente());
                        pick.set4idEmpleado(idUser);
                        pick.set5FechaPicking(DateTime.Now);

pick.set6NumeroPedido(DTpick.Rows[0]["#Pedido"].ToString());

pick.set7Bultos(Convert.ToInt32(numericUpDown1.Value));
                        Adminpick.IngresarPicking(pick);

                        foreach (DataRow dr in DTpick.Rows)
                        {

pick.setBidProducto(dr["idProducto"].ToString());

pick.setCidDetalleProducto(dr["idDetalleProducto"].ToString());
                            pick.setDCantidad(int.Parse(dr["Cantidad
Reservada"].ToString()));

pick.setEidReserva(dr["idReserva"].ToString());
                            Adminpick.IngresarDetallePicking(pick);

                                }
                                ord.tabla2.Clear();
                                DTpick.Clear();
                                MessageBox.Show("Salida Exitosa");
                                ord.dataGridReserva.DataSource = "";
                                this.Close();
                            }
                        else
                        {
                            MessageBox.Show("ERROR: Elija la Cantidad de
Bultos!!", "Datos Invalidos");
                        }
                    }
                }
            }
        }
    }
}

```



```

int numero3 = 0;
int[] posiciones = new int[6];
public String idProv;
public String idUser;
public String NivelAc;

public FormConsultas()
{
    InitializeComponent();
}
public void Censo()
{
    this.dataGridBProducto.Columns["idProducto"].Visible =
false;
    this.dataGridBProducto.Columns["idTipoProducto"].Visible =
false;
    this.dataGridBProducto.Columns["idProveedor"].Visible =
false;
}
public void Censo2()
{
    this.dataGridBProv.Columns["idProducto"].Visible = false;
    this.dataGridBProv.Columns["idDetalleProducto"].Visible =
false;
    this.dataGridBProv.Columns["idEstado"].Visible = false;
}
private void fillDataTipo()
{
    DataTable tablaAux = new DataTable();

    try
    {
        Cursor = Cursors.WaitCursor;
        tablaAux.Columns.Add("codigo");
        tablaAux.Columns.Add("Producto");
        tablaAux.Columns.Add("Proveedor");
        tablaAux.Columns.Add("Tipo");
        tablaAux.Columns.Add("Lote");
        tablaAux.Columns.Add("FechaIngreso");
        tablaAux.Columns.Add("fechaVencimineto");
        tablaAux.Columns.Add("Peso");
        tablaAux.Columns.Add("Stock");
        tablaAux.Columns.Add("Des");

        dataGridReporte1.DataSource = tablaAux;

        for (int i = 0; i < 50; i++)
        {
            if (numero3 < tabla3.Rows.Count)
            {

```

```

tablaAux.LoadDataRow(tabla3.Rows[numero3].ItemArray, false);
        numero3++;
    }
    else
    {
        break;
    }
}

}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    Cursor = Cursors.Default;
}
}

private void boton1_Click(object sender, EventArgs e)
{

    buttonder2.Enabled = true;
    buttoniz2.Enabled = true;
    try
    {
        if (checkBox1.Checked)
        {
            tabla3 =
adminp.BuscarProductosCortaFecha(idProv).Tables[0];

        }
        else
        {
            tabla3 =
adminp.BuscarProductos(cod[comboBoxBodega.SelectedIndex], idProv,
tipo[comboBoxidProducto.SelectedIndex],
comboBoxStock.SelectedIndex).Tables[0];

        }
        if ((tabla3.Rows.Count % 50) == 0)
        {
            labelTotal3.Text = "/" + (tabla3.Rows.Count /
50).ToString();
        }
        else
        {
            labelTotal3.Text = "/" + ((tabla3.Rows.Count / 50)
+ 1).ToString();
        }
        if (tabla3.Rows.Count > 0)
        {
            textBoxPag3.Text = "1";
        }
        else
        {

```

```

        textBoxPag3.Text = "0";
    }
    if (tabla3.Rows.Count > 50)
    {
        fillDataTipo();
    }
    else
    {
        this.dataGridReportel.DataSource = tabla3;
    }
}
catch
{
    MessageBox.Show("ERROR: Ingrege una busqueda
Valida!!", "Error de Busqueda");
}
}
private void fillDataGrid_dtgBrowse()
{
    DataTable tablaAux = new DataTable();

    try
    {
        Cursor = Cursors.WaitCursor;
        tablaAux.Columns.Add("idProducto");
        tablaAux.Columns.Add("idProveedor");
        tablaAux.Columns.Add("idTipoProducto");
        tablaAux.Columns.Add("codigo");
        tablaAux.Columns.Add("Producto");
        tablaAux.Columns.Add("Proveedor");
        tablaAux.Columns.Add("Tipo");
        tablaAux.Columns.Add("Peso");
        tablaAux.Columns.Add("Stock");

        dataGridBProducto.DataSource = tablaAux;

        for (int i = 0; i < 50; i++)
        {
            if (numero < tabla.Rows.Count)
            {
                tablaAux.LoadDataRow(tabla.Rows[numero].ItemArray, false);
                numero++;
            }
            else
            {
                break;
            }
        }
    }
    catch (Exception ex)
    {

```

```

        MessageBox.Show(ex.Message);
    }
    finally
    {
        Cursor = Cursors.Default;
    }
}

private void FormConsultas_Load(object sender, EventArgs e)
{
    try
    {
        int i = 0;
        foreach (DataRow dr in
this.aux.RecuperarBodega().Tables[0].Rows)
        {

this.comboBoxBodega.Items.Add(dr["NombreBodega"].ToString());

this.comboBoxBod.Items.Add(dr["NombreBodega"].ToString());

            cod[i] = dr["idBodega"].ToString();
            i++;
        }
        comboBoxBod.Items.Add("Todas");
        i = 0;
        foreach (DataRow dr in
this.adminp.RecuperarTipos().Tables[0].Rows)
        {

this.comboBoxidProducto.Items.Add(dr["Nombre"].ToString());
            this.tipo[i] = dr["Codigo"].ToString();

            i++;
        }
        DataRow fila =
aux.RecuperarPosicionesActuales().Tables[0].Rows[0];
        posiciones[0] = int.Parse(fila["Total"].ToString());
        posiciones[1] = int.Parse(fila["LLeno"].ToString());
        posiciones[2] = int.Parse(fila["Vacio"].ToString());
        //=====progresbar llena actual=====

        progressBarUtiA.Maximum=posiciones[0];
        progressBarUtiA.Minimum = 0;
        progressBarUtiA.Value = posiciones[1];
        labelutiliA.Text = "POSICIONES UTILIZADAS: " +
posiciones[1].ToString();
        labelUtilAP.Text = ((int)(((float)posiciones[1] /
posiciones[0]) * 100)).ToString() + "%";
        //=====progresbar vacias actual=====

        progressBarLibreA.Maximum = posiciones[0];
        progressBarLibreA.Minimum = 0;
        progressBarLibreA.Value = posiciones[2];
    }
}

```



```

        labellibA.Text = "POSICIONES LIBRES: " +
posiciones[2];
        labeltotalA.Text = "POSICIONES TOTALES: " +
posiciones[0].ToString();
        this.labelLibAP.Text =
Math.Round((((decimal)posiciones[2] / posiciones[0]) *
100)).ToString() + "%";

    }
    catch
    {
        MessageBox.Show("ERROR: Problemas de Coneccion!");
    }

}

private void comboBoxBodega_SelectedIndexChanged(object
sender, EventArgs e)
{
    try
    {
        comboBoxProve.Items.Clear();
        comboBoxProvAP.Items.Clear();

        int i = 0;
        foreach (DataRow dr in
this.prov.RecuperarProvedores().Tables[0].Rows)
        {

this.comboBoxProve.Items.Add(dr["Nombre"].ToString());

this.comboBoxProvAP.Items.Add(dr["Nombre"].ToString());
            prove[i] = dr["idProveedor"].ToString();
            i++;
        }
        label3.Visible = true;
        comboBoxProve.Visible = true;
    }
    catch
    {
        MessageBox.Show("Error");
    }
}

private void comboBoxProve_SelectedIndexChanged(object sender,
EventArgs e)
{
    idProv= prove[comboBoxProve.SelectedIndex];
    tabControll.Visible = true;
}

private void toolStripButton6_Click(object sender, EventArgs
e)
{
    String msg;

```

```

        this.saveFileDialog1.DefaultExt= "xls";
        saveFileDialog1.Filter = "XLS file (*.xls)|*.xls|All files
(*.*)|*.*";
        saveFileDialog1.RestoreDirectory = true;
        saveFileDialog1.Title = "GUARDAR REPORTE?";
        saveFileDialog1.InitialDirectory = @"C:/";
        if (saveFileDialog1.ShowDialog() != DialogResult.Cancel)
        {
            this.Cursor = Cursors.WaitCursor;
            msg=auxi.AbrirConsultaExcel(dataGridReportel,
saveFileDialog1.FileName, "A5", "ReporteProductos");
            MessageBox.Show(msg, "GENERAR REPORTE");
            this.Cursor = Cursors.Default;
            MessageBox.Show(msg);
        }
    }

    private void comboBox1_SelectedIndexChanged(object sender,
EventArgs e)
    {
        try
        {
            DataRow fila =
aux.RecuperarPosicionesAcProve (prove[comboBoxProvAP.SelectedIndex]).Ta
bles[0].Rows[0];
            posiciones[3] = int.Parse(fila["Total"].ToString());
            posiciones[4] = int.Parse(fila["LLeno"].ToString());
            posiciones[5] = int.Parse(fila["Vacio"].ToString());
            //=====progresbar llena actual
Provee=====

            this.progressBar1.Maximum = posiciones[3];
            this.progressBar1.Minimum = 0;
            this.progressBar1.Value = posiciones[4];
            this.label11.Text = "POSICIONES UTILIZADAS: " +
posiciones[4].ToString();
            this.labelUtilAPee.Text = ((int)((float)posiciones[4]
/ posiciones[3]) * 100).ToString() + "%";
            //=====progresbar vacias actual
Proveee=====

            this.progressBar2.Maximum = posiciones[3];
            this.progressBar2.Minimum = 0;
            this.progressBar2.Value = posiciones[5];
            this.label12.Text = "POSICIONES LIBRES: " +
posiciones[5];
            this.label13.Text = "POSICIONES TOTALES: " +
posiciones[3].ToString();
            this.labelLibAPee.Text =
Math.Round(((decimal)posiciones[5] / posiciones[3]) *
100).ToString() + "%";
        }
        catch
        {
            MessageBox.Show("No se encontro posiciones de ese
Proveedor!");
        }
    }

```

```

    }

    private void button2_Click(object sender, EventArgs e)
    {
        if (auxi.autenticacion(richBoxbuscar.Text) == "")
        {
            MessageBox.Show("ERROR: Seleccione un Parametro de
Busqueda!");
        }
        else
        {
            buttoniz1.Enabled = true;
            buttondel1.Enabled = true;
            try
            {
                numero = 0;
                tabla =
adminp.BuscarProductos(auxi.autenticacion(this.richBoxbuscar.Text),
prove[comboBoxProve.SelectedIndex]).Tables[0];

                label9.Text = "Productos Encontrados: " +
tabla.Rows.Count.ToString();
                if ((tabla.Rows.Count % 50) == 0)
                {
                    label8.Text = "/" + (tabla.Rows.Count /
50).ToString();
                }
                else
                {
                    label8.Text = "/" + ((tabla.Rows.Count / 50) +
1).ToString();
                }
                textBox2.Text = "1";
                if (tabla.Rows.Count > 50)
                {
                    fillDataGrid_dtgBrowse();
                    Censo();
                }
                else
                {
                    this.dataGridBProducto.DataSource = tabla;
                    Censo();
                }
                //=====
            }
            catch
            {
                MessageBox.Show("Error: Ingrese Caracteres Validos
!", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}

```

```

private void button5_Click(object sender, EventArgs e)
{
    if (rowheader1 > -1)
    {
        tabla2 = pick.BuscarKardex(producto.getlidProducto());

        FormReport xForm;
        foreach (Form childForm in this.MdiChildren)
        {
            if (childForm.GetType() == typeof(FormReport))
            {
                childForm.Focus();
                return;
            }
        }
        xForm = new FormReport();
        xForm.tabla = tabla2;
        xForm.Show();
    }
    else
    {
        MessageBox.Show("Seleccione un Producto!");
    }
}

private void button4_Click(object sender, EventArgs e)
{
    if (tabla3.Rows.Count / 50 < 1)
    {
        MessageBox.Show("No existe mas Paginas");
    }
    else
    {
        if (numero3 < tabla3.Rows.Count)
        {
            buttoniz2.Enabled = true;
            fillDataTipo();

            if ((numero % 50) == 0)
            {
                textBoxPag3.Text = (numero3 / 50).ToString();
            }
            else
            {
                textBoxPag3.Text = ((numero3 / 50) +
1).ToString();
            }
        }
        else
        {
            buttonder2.Enabled = false;
        }
    }
}

private void button5_Click_1(object sender, EventArgs e)
{

```

```

        if (dateTimePicker1.Value > dateTimePicker2.Value)
        {
            MessageBox.Show("Seleccione una diferencia de fecha
Valida!");
        }
        else
        {
            dataGridReporte2.DataSource =
pick.BuscarKardex(idProv, dateTimePicker1.Value.ToShortDateString(),
dateTimePicker2.Value.ToShortDateString());
        }
    }

    private void buttonder4_Click(object sender, EventArgs e)
    {

    }

    private void buttoniz2_Click(object sender, EventArgs e)
    {
        if (numero3 % 50 != 0)
        {
            numero3 = (numero3 - (numero3 % 50));
        }
        if (numero3 > 0)
        {
            numero3 = numero3 - 100;
            if (numero3 < 0)
            {
                numero3 = 0;
            }
            buttonder2.Enabled = true;
            fillDataTipo();

            if ((numero3 % 50) == 0)
            {
                textBoxPag3.Text = (numero3 / 50).ToString();
            }
            else
            {
                textBoxPag3.Text = ((numero3 / 50) +
1).ToString();
            }

            if (numero3 < 0)
            {
                numero3 = 0;
            }

        }
        else
        {
            buttoniz2.Enabled = false;
        }
    }
}

```

```

e)         private void toolStripButton1_Click(object sender, EventArgs
{
            String msg;
            this.saveFileDialog1.DefaultExt = "xls";
            saveFileDialog1.Filter = "XLS file (*.xls)|*.xls|All
files (*.*)|*.*";
            saveFileDialog1.RestoreDirectory = true;
            saveFileDialog1.Title = "GUARDAR REPORTE?";
            saveFileDialog1.InitialDirectory = @"C:/";
            if (saveFileDialog1.ShowDialog() != DialogResult.Cancel)
            {
                this.Cursor = Cursors.WaitCursor;
                msg = auxi.AbrirConsultaExcel(this.dataGridReportel,
saveFileDialog1.FileName, "A5", "ReporteProductos");
                this.Cursor = Cursors.Default;
                MessageBox.Show(msg);
            }
        }
    }
}

```

```

namespace WMSPRIME
{
    public partial class WMSmain : Form
    {
        public String NivelUsuario;
        public String CodigoUsuario;
        public String CodigoProvee;

        FormCrearProducto crearproducto = new FormCrearProducto();

        public WMSmain()
        {
            InitializeComponent();
        }

        private void ShowNewForm(object sender, EventArgs e)
        {
        }

        private void OpenFile(object sender, EventArgs e)
        {
            OpenFileDialog openFileDialog = new OpenFileDialog();
            openFileDialog.InitialDirectory =
Environment.GetFolderPath(Environment.SpecialFolder.Personal);
            openFileDialog.Filter = "Text Files (*.txt)|*.txt|All
Files (*.*)|*.*";
            if (openFileDialog.ShowDialog(this) == DialogResult.OK)
            {
                string FileName = openFileDialog.FileName;
            }
        }

        private void SaveAsToolStripMenuItem_Click(object sender,
EventArgs e)

```

```

    {
        SaveFileDialog saveFileDialog = new SaveFileDialog();
        saveFileDialog.InitialDirectory =
Environment.GetFolderPath(Environment.SpecialFolder.Personal);
        saveFileDialog.Filter = "Text Files (*.txt)|*.txt|All
Files (*.*)|*.*";
        if (saveFileDialog.ShowDialog(this) == DialogResult.OK)
        {
            string FileName = saveFileDialog.FileName;
        }
    }

    private void ExitToolsStripMenuItem_Click(object sender,
EventArgs e)
    {
        this.Close();
    }

    private void CutToolStripMenuItem_Click(object sender,
EventArgs e)
    {
    }

    private void CopyToolStripMenuItem_Click(object sender,
EventArgs e)
    {
    }

    private void PasteToolStripMenuItem_Click(object sender,
EventArgs e)
    {
    }

    //private void ToolBarToolStripMenuItem_Click(object sender,
EventArgs e)
    //{
    //    toolStrip.Visible = toolBarToolStripMenuItem.Checked;
    //}

    private void StatusBarToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        statusStrip.Visible = statusBarToolStripMenuItem.Checked;
    }

    private void CascadeToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        LayoutMdi(MdiLayout.Cascade);
    }

    private void TileVerticalToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        LayoutMdi(MdiLayout.TileVertical);
    }

    private void TileHorizontalToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        LayoutMdi(MdiLayout.TileHorizontal);
    }

```

```

    }

    private void ArrangeIconsToolStripMenuItem_Click(object
sender, EventArgs e)
    {
        LayoutMdi (MdiLayout.ArrangeIcons);
    }

    private void CloseAllToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        foreach (Form childForm in MdiChildren)
        {
            childForm.Close();
        }
    }

    private void CrearEmpleadoStripMenu_Click(object sender,
EventArgs e)
    {
        FormCrearEmpleado xForm;
        foreach (Form childForm in this.MdiChildren)
        {
            if (childForm.GetType() == typeof(FormCrearEmpleado))
            {
                childForm.Focus();
                return;
            }
        }
        xForm = new FormCrearEmpleado();
        xForm.MdiParent = this;
        xForm.Show();

    }

    private void toolStripMenuItem18_Click(object sender,
EventArgs e)
    {
        FormCrearProvedor xForm;
        foreach (Form childForm in this.MdiChildren)
        {
            if (childForm.GetType() == typeof(FormCrearProvedor))
            {
                childForm.Focus();
                return;
            }
        }
        xForm = new FormCrearProvedor();
        xForm.MdiParent = this;
        xForm.Show();

    }

    private void nuevaOrdenToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        FormCrearCiudad xForm;
        foreach (Form childForm in this.MdiChildren)
        {

```



```

        if (childForm.GetType() == typeof(FormCrearCiudad))
        {
            childForm.Focus();
            return;
        }
    }
    xForm = new FormCrearCiudad();
    xForm.MdiParent = this;
    xForm.Show();

}

e) private void toolStripMenuItem5_Click(object sender, EventArgs
    {
        FormCrearProducto xForm;
        foreach (Form childForm in this.MdiChildren)
        {
            if (childForm.GetType() == typeof(FormCrearProducto))
            {
                childForm.Focus();
                return;
            }
        }
        xForm = new FormCrearProducto();
        xForm.MdiParent = this;
        xForm.idProv =CodigoProvee;
        xForm.idUser =CodigoUsuario;
        xForm.NivelAc = NivelUsuario;

        xForm.Show();

    }

e) private void toolStripMenuItem7_Click(object sender, EventArgs
    {
        FormCrearPosicion xForm;
        foreach (Form childForm in this.MdiChildren)
        {
            if (childForm.GetType() == typeof(FormCrearPosicion))
            {
                childForm.Focus();
                return;
            }
        }
        xForm = new FormCrearPosicion();
        xForm.MdiParent = this;
        xForm.Show();

    }

private void cargarOrdenToolStripMenuItem_Click(object sender,
EventArgs e)
    {
        FormBuscarCiudad xForm;

```

```

        foreach (Form childForm in this.MdiChildren)
        {
            if (childForm.GetType() == typeof(FormBuscarCiudad))
            {
                childForm.Focus();
                return;
            }
        }
        xForm = new FormBuscarCiudad();
        xForm.MdiParent = this;
        xForm.Show();
    }

    private void toolStripMenuItem16_Click(object sender,
EventArgs e)
    {
        FormBuscarCiudad xForm;
        foreach (Form childForm in this.MdiChildren)
        {
            if (childForm.GetType() == typeof(FormBuscarCiudad))
            {
                childForm.Focus();
                return;
            }
        }
        xForm = new FormBuscarCiudad();
        xForm.MdiParent = this;
        xForm.buttonElim.Visible = true;
        xForm.label2.Text = "Eliminar Ciudad";
        xForm.Text = "Eliminar Ciudad";
        xForm.Show();
    }

    private void toolStripMenuItem3_Click(object sender, EventArgs
e)
    {
        FormBuscarCiudad xForm;
        foreach (Form childForm in this.MdiChildren)
        {
            if (childForm.GetType() == typeof(FormBuscarCiudad))
            {
                childForm.Focus();
                return;
            }
        }
        xForm = new FormBuscarCiudad();
        xForm.MdiParent = this;
        xForm.buttonMod.Visible = true;
        xForm.label2.Text = "Modificar Ciudad";
        xForm.Text = "Modificar Ciudad";
        xForm.Show();
    }
}

```

```

        private void toolStripMenuItem17_Click(object sender,
EventArgs e)
        {
            FormBuscarProveedor xForm;
            foreach (Form childForm in this.MdiChildren)
            {
                if (childForm.GetType() ==
typeof(FormBuscarProveedor))
                {
                    childForm.Focus();
                    return;
                }
            }
            xForm = new FormBuscarProveedor();
            xForm.MdiParent = this;
            xForm.Show();

        }

e) private void toolStripMenuItem4_Click(object sender, EventArgs
{
    FormBuscarProducto xForm;
    foreach (Form childForm in this.MdiChildren)
    {
        if (childForm.GetType() == typeof(FormBuscarProducto))
        {
            childForm.Focus();
            return;
        }
    }
    xForm = new FormBuscarProducto();
    xForm.MdiParent = this;
    xForm.idProv =CodigoProvee;
    xForm.idUser =CodigoUsuario;
    xForm.NivelAc = NivelUsuario;
    xForm.Show();

}

private void toolStripMenuItem12_Click(object sender,
EventArgs e)
{
    FormBuscarProducto xForm;
    foreach (Form childForm in this.MdiChildren)
    {
        if (childForm.GetType() == typeof(FormBuscarProducto))
        {
            childForm.Focus();
            return;
        }
    }
    xForm = new FormBuscarProducto();
    xForm.MdiParent = this;
    xForm.buttonElim.Visible = true;
    xForm.idProv =CodigoProvee;
    xForm.idUser =CodigoUsuario;
    xForm.NivelAc = NivelUsuario;
    xForm.label2.Text = "Eliminar Producto";
}

```

```

        xForm.Show();
    }

    private void toolStripMenuItem25_Click(object sender,
EventArgs e)
    {
        FormBuscarProducto xForm;
        foreach (Form childForm in this.MdiChildren)
        {
            if (childForm.GetType() == typeof(FormBuscarProducto))
            {
                childForm.Focus();
                return;
            }
        }
        xForm = new FormBuscarProducto();
        xForm.MdiParent = this;
        xForm.buttonMod.Visible = true;
        xForm.idProv = CodigoProvee;
        xForm.idUser = CodigoUsuario;
        xForm.NivelAc = NivelUsuario;
        xForm.label2.Text = "Modificar Producto";
        xForm.Show();

    }

    private void toolStripMenuItem26_Click(object sender,
EventArgs e)
    {
        FormEstado xForm;
        foreach (Form childForm in this.MdiChildren)
        {
            if (childForm.GetType() == typeof(FormEstado))
            {
                childForm.Focus();
                return;
            }
        }
        xForm = new FormEstado();
        xForm.MdiParent = this;
        xForm.Show();

    }

    private void toolStripMenuItem6_Click(object sender, EventArgs
e)
    {
        FormTipoProducto xForm;
        foreach (Form childForm in this.MdiChildren)
        {
            if (childForm.GetType() == typeof(FormTipoProducto))
            {
                childForm.Focus();
                return;
            }
        }
        xForm = new FormTipoProducto();
        xForm.MdiParent = this;
    }

```

```

        xForm.Show();
    }

    private void toolStripMenuItem22_Click(object sender,
EventArgs e)
    {
        FormCrearCliente xForm;
        foreach (Form childForm in this.MdiChildren)
        {
            if (childForm.GetType() == typeof(FormCrearCliente))
            {
                childForm.Focus();
                return;
            }
        }
        xForm = new FormCrearCliente();
        xForm.MdiParent = this;
        xForm.Show();
    }

    private void toolStripMenuItem21_Click(object sender,
EventArgs e)
    {
        FormBuscarCliente xForm;
        foreach (Form childForm in this.MdiChildren)
        {
            if (childForm.GetType() == typeof(FormBuscarCliente))
            {
                childForm.Focus();
                return;
            }
        }
        xForm = new FormBuscarCliente();
        xForm.MdiParent = this;
        xForm.Show();
    }

    private void toolStripMenuItem23_Click(object sender,
EventArgs e)
    {
        FormBuscarCliente xForm;
        foreach (Form childForm in this.MdiChildren)
        {
            if (childForm.GetType() == typeof(FormBuscarCliente))
            {
                childForm.Focus();
                return;
            }
        }
        xForm = new FormBuscarCliente();
        xForm.MdiParent = this;
        xForm.label2.Text = "Eliminar Cliente";
    }

```

```

        xForm.Text = "Eliminar Cliente";
        xForm.buttonElim.Visible = true;
        xForm.Show();

    }

private void toolStripMenuItem2_Click(object sender, EventArgs
{
    FormBuscarPosicion xForm;
    foreach (Form childForm in this.MdiChildren)
    {
        if (childForm.GetType() == typeof(FormBuscarPosicion))
        {
            childForm.Focus();
            return;
        }
    }
    xForm = new FormBuscarPosicion();
    xForm.MdiParent = this;
    xForm.idProv = CodigoProvee;
    xForm.idUser = CodigoUsuario;
    xForm.NivelAc = NivelUsuario;
    xForm.Show();
}

private void toolStripMenuItem30_Click(object sender,
EventArgs e)
{
    FormBuscarProPos xForm;
    foreach (Form childForm in this.MdiChildren)
    {
        if (childForm.GetType() == typeof(FormBuscarProPos))
        {
            childForm.Focus();
            return;
        }
    }
    xForm = new FormBuscarProPos();
    xForm.MdiParent = this;
    xForm.idProv = CodigoProvee;
    xForm.idUser = CodigoUsuario;
    xForm.NivelAc = NivelUsuario;
    xForm.Show();
}
}

```

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Las conclusiones que se extraen del documento van dirigidas al mundo empresarial de la logística que esté interesado en el desarrollo de sistemas WMS para el manejo integral de productos.

Este título hace un pequeño resumen de las conclusiones más relevantes que se han presentado durante el desarrollo de este proyecto tanto de investigación como de desarrollo a lo largo de toda la tesis.

5.1.1. Aportación de la Tesis

El aporte de esta tesis es investigativo y de desarrollo. Investigativo porque se recaudó información sobre los sistemas WMS actuales y los requerimientos internos de la empresa Logistecsa. S.A sobre sus procesos internos para la automatización en el sistema desarrollado. De desarrollo porque se logró concentrar la mayoría de requerimientos dentro de un sistema integral, el cual proveerá información en tiempo real y oportuna dentro de las bodegas de la empresa.

A continuación se sintetiza los resultados obtenidos de los objetivos planteados a lo largo de esta tesis:

- Al revisar y analizar los fundamentos de los diferentes sistemas administrativos o bien llamados WMS, se llegó a la conclusión que se

necesitaba elaborar un sistema más pequeño a los existentes, ya que estos requieren de gran capacitación y una fuerte suma de inversión por lo que un sistema a medida resulta una opción más eficiente.

- Al diseñar bajo la metodología MSF la plataforma WMS de la empresa Logistecsa, se logró establecer la metodología exitosamente y seguirla paso por paso, cumpliendo los hitos propuestos tanto por la empresa como el desarrollador. Obtener los diferentes requerimientos del sistema dentro de la empresa Logistecsa, se alcanzó gracias a un exhaustivo levantamiento de requerimientos mediante la metodología MSF y sus técnicas, además fue fundamental la ayuda recibida por parte del personal de la empresa que facilitó enormemente el cumplimiento de este objetivo.
- Al utilizar la herramienta C# .NET para el desarrollo, se contó con una gran dinámica al momento de programar y facilitó enormemente el trabajo dado por su potente framework 3.5.
- Crear una interfaz gráfica amigable con el usuario y tratar de que esta controle la mayoría de errores producidos por el personal, se logró al aplicar las buenas prácticas de programación, un diseño simple y muy lógico para el usuario.
- Limitar el acceso de información a usuarios de bajo nivel para reducir información errónea dentro de la base de datos, diseñando una estructura multi capa se restringió al máximo el manejo de información a los usuarios, dándole a la información mayor seguridad y control.

5.1.2. Respecto a la Propuesta del WMS

El sistema WMS busca ser un medio de control y ejecución de las diferentes áreas de una empresa de logística, brindando seguridad y eficiencia para los procesos involucrados.

La conclusión del plan es recalcar la importancia de invertir para el futuro y no sólo en las áreas tradicionales de inversión, como nuevo personal o más espacio en hangares. En este sentido la empresa Logistecsa S.A deberá invertir en su infraestructura informática, dándole un papel más protagónico a los sistemas de control y gestión como es el WMS.

5.2. Recomendaciones

Estas recomendaciones están dirigidas a la empresa Logistecsa S.A sobre la administración de bodegas y a futuros desarrolladores que quieran sumergirse en el tema logístico de los WMS.

Actualmente los sistemas de wms están en etapas muy avanzadas a nivel mundial, los cuales son herramientas muy poderosas pero a la vez muy complicadas y costosas para el funcionamiento de los actuales sistemas de logística en el país, por lo que un sistema a medida o por lo menos de menor escala sería la solución perfecta para este medio

La empresa Logistecsa S.A debe empezar a darle mayor importancia a los sistemas informáticos, ya que hoy en día existen muchas propuestas para la mejora del negocio, pensando siempre en un avance continuo en el área de sistemas ya que esta brinda soporte a todas las áreas de la empresa.

Al desarrollar esta aplicación en .Net se logró entregar un producto de calidad y eficaz pero para sacarle mejor provecho se podría realizar una versión web, cargando menos el ambiente visual y aumentando el despliegue por la web.

Para situar a la innovación como un arma competitiva y una herramienta de crecimiento de la empresa, es importante aprovechar la capacidad tecnológica de sus recursos humanos, potenciando el liderazgo del Director IT.

REFERENCIAS BIBLIOGRÁFICAS

Libros:

Alter, S. L. (1980). Decision support systems: current practice and continuing challenges. Reading, Mass., Addison-Wesley Pub.

Ballou, R. H. (1999). *Business Logistics Management*, Cuarta edición edición, Upper Saddle River. Nueva Jersey, EE.UU. Prentice-Hall International Inc

Council of Supply Chain Management Professionals, CSCMP, IW, USA

IEEE Std. 1219-1998 (1998) Standard for Software Maintenance. Los Alamitos.EEUU.

IEEE Computer Society Press.

Lambert, D.M.; Cooper, M.; Pagh J. D. (1998) *Supply Chain Management: Implementation Issues and Research Opportunities*, The International Journal of Logistics Management. Vol.9, n° 2, pp 1-19.PE, USA

Phillip A. Laplante (2007) What Every Engineer Should Know about Software Engineering, Taylor & Francis Group, Boca Raton, FL, USA

Pigoski, T. M. (1997) Practical Software Maintenance – Best Practices for Managing Your Software Investment. John Wiley & Sons. New York..NY.

Real Academia Española (2001). Diccionario de la Lengua Española, Vigésima segunda edición edición, Madrid (España)

Internet:

Microsoft (s/f) Microsoft Solutions Framework. En: <http://www.microsoft.com/technet/prodtechnol/exchange/guides/ExMgmtGuide/62fab0e2-a570-4921-a0f1-7fdc6fff2a9.msp?mfr=true>. Consultado el 3 de noviembre de 2010.

(2008a) Microsoft Tech Net. Installing SQL Server. En: <http://technet.microsoft.com/en-us/library/ms143516%28SQL.90%29.aspx>. Consultado el 4 de mayo de 2010.

_____ (2008b) Microsoft Tech Net. Administrar servidores con SQL Server Management Studio. En: <http://technet.microsoft.com/es-es/library/ms173793%28SQL.90%29.aspx>.

MSDN (s/f) Using SQL Server Management Studio. En: <http://msdn.microsoft.com/en-us/library/ms174173.aspx>.

_____ (s/f) Lo nuevo en el depurador de Visual Studio. En: <http://msdn.microsoft.com/es-EC/library/01xd7cs.aspx>.

_____ (2003) Información general sobre escalabilidad. En: [http://msdn.microsoft.com/es-es/library/aa292203\(VS.71\).aspx](http://msdn.microsoft.com/es-es/library/aa292203(VS.71).aspx).

Torode, C. (s/f) SharePoint alternatives seek to fill in the gaps. TECH TARGET. En: <http://searchcio-midmarket.techtarget.com/tip/SharePoint-alternatives-seek-to-fill-in-the-gaps>.

VISIO.Microsoft (s/f) En: <http://visiotoolbox.com/2010/es/business-intelligence.aspx>.

Wikipedia, la enciclopedia libre (s/f). OLTP. En: <http://es.wikipedia.org/wiki/OLTP>. Consultado el 3 de noviembre de 2010.

_____ Logística empresarial: definiciones. En: http://es.wikipedia.org/wiki/Logística#Log.C3.ADstica_empresarial:_definiciones.

[http://i.msdn.microsoft.com/Aa292203.vxscalabilityscaleup\(es-es,VS.71\).gif](http://i.msdn.microsoft.com/Aa292203.vxscalabilityscaleup(es-es,VS.71).gif).

¹ [http://i.msdn.microsoft.com/Aa292203.vxscalabilityscaleup\(es-es,VS.71\).gif](http://i.msdn.microsoft.com/Aa292203.vxscalabilityscaleup(es-es,VS.71).gif).

APÉNDICE 1

BIOGRAFÍA

El Sr. Andrés Salvador Marchant Castelnuovo, nació en Quito el 20 de abril de 1984, cursó sus estudios escolares en varias instituciones académicas de la capital, se graduó en la escuela “El SAUCE” en el año de 1996.

Sus estudios secundarios los realizó en la Unidad Educativa “LICEO DEL VALLE” y se graduó de bachiller nacional con especialidad en químico biólogo, en el año 2002.

Ese mismo año ingresó a la Facultad de Ingeniería en Sistemas de la P.U.C.E, donde cursó hasta tercer año. En el 2005 ingresa a la ESCUELA POLITECNICA DEL EJERCITO donde continúa sus estudios en la carrera Ingeniería de Sistemas y Computación.

La carrera se vio pospuesta durante un semestre al aceptar un voluntariado en la Estación Científica CHARLES DARWIN, donde obtuvo importantes experiencias profesionales y vivenciales. Durante este período realiza su instrucción en buceo.

Gracias al sistema educativo de la ESPE y a la modalidad a distancia, MED, logró continuar con algunas materias desde GALAPAGOS continuando así su carrera profesional.

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADA(O) POR

ANDRÉS SALVADOR MARCHANT CASTELNUOVO

DIRECTOR DE LA CARRERA

Sr. Ing Mauricio Campaña

Lugar y fecha: Sangolquí 2 de Abril del 2012