



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE ANÁLISIS DE SISTEMAS

SISTEMATIZACIÓN DE LOS PROCESOS DE CONTROL DE INVENTARIO A
TRAVÉS DE UN SOFTWARE ORIENTADO A LA WEB PARA LA EMPRESA
GLOBALCONNECTION DE LA CIUDAD DE QUITO

Proyecto de Investigación Previo a la obtención del título de Tecnólogo en Análisis
de Sistemas

Autor: Luis Clever Pupiales Farinango

Tutor: Ing. Juan Minango

Quito, Abril 2015

Declaratoria

Declaro que la investigación es absolutamente original, autentica, personal, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes. Las ideas, doctrinas, resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.

Luis Clever Pupiales Farinango

CC 1720242914

Contrato de Cesión Sobre Derechos Propiedad Intelectual

Comparecen a la celebración del presente contrato de cesión y transferencia de derechos de propiedad intelectual, por una parte, el estudiante Luis Clever Pupiales Farinango, por sus propios y personales derechos, a quien en lo posterior se le denominará el “CEDENTE”; y, por otra parte, el INSTITUTO SUPERIOR TECNOLÓGICO CORDILLERA, representado por su Rector el Ingeniero Ernesto Flores Córdova, a quien en lo posterior se lo denominará el “CESIONARIO”. Los comparecientes son mayores de edad, domiciliados en esta ciudad de Quito Distrito Metropolitano, hábiles y capaces para contraer derechos y obligaciones, quienes acuerdan al tenor de las siguientes cláusulas:

PRIMERA: ANTECEDENTE.- a) El Cedente dentro del pensum de estudio en la carrera de análisis de sistemas que imparte el Instituto Superior Tecnológico Cordillera, y con el objeto de obtener el título de Tecnólogo en Análisis de Sistemas, el estudiante participa en el proyecto de grado denominado “sistematización de los procesos de control de inventario a través de un software orientado a la web para la empresa Globalconnection de la ciudad de Quito”, el cual incluye la creación y desarrollo del programa de ordenador o software, para lo cual ha implementado los conocimientos adquiridos en su calidad de alumno. b) Por iniciativa y responsabilidad del Instituto Superior Tecnológico Cordillera se desarrolla la creación del programa de ordenador, motivo por el cual se regula de forma clara la cesión de los derechos de autor que genera la obra literaria y que es producto del Proyecto de grado, el mismo que culminado es de plena aplicación técnica, administrativa y de reproducción.

SEGUNDA: CESIÓN Y TRANSFERENCIA.- Con el antecedente indicado, el Cedente libre y voluntariamente cede y transfiere de manera perpetua y gratuita todos los derechos patrimoniales del programa de ordenador descrito en la cláusula anterior a favor del Cesionario, sin reservarse para sí ningún privilegio especial (código fuente, código objeto, diagramas de flujo, planos, manuales de uso, etc.). El Cesionario podrá explotar el programa de ordenador por cualquier medio o procedimiento tal cual lo establece el Artículo 20 de la Ley de Propiedad Intelectual, esto es, realizar, autorizar o prohibir, entre otros: a) La reproducción del programa de ordenador por cualquier forma o procedimiento; b) La comunicación pública del software; c) La distribución pública de ejemplares o copias, la comercialización, arrendamiento o alquiler del programa de ordenador; d) Cualquier transformación o modificación del programa de ordenador; e) La protección y registro en el IEPI el programa de ordenador a nombre del Cesionario; f) Ejercer la protección jurídica del programa de ordenador; g) Los demás

Derechos establecidos en la Ley de Propiedad Intelectual y otros cuerpos legales que normen sobre la cesión de derechos de autor y derechos patrimoniales.

TERCERA: OBLIGACIÓN DEL CEDENTE.- El cedente no podrá transferir a ningún tercero los derechos que conforman la estructura, secuencia y organización del programa de ordenador que es objeto del presente contrato, como tampoco emplearlo o utilizarlo a título personal, ya que siempre se deberá guardar la exclusividad del programa de ordenador a favor del Cesionario

CUARTA: CUANTIA.- La cesión objeto del presente contrato, se realiza a título gratuito y por ende el Cesionario ni sus administradores deben cancelar valor alguno o regalías por este contrato y por los derechos que se derivan del mismo.

QUINTA: PLAZO.- La vigencia del presente contrato es indefinida.

SEXTA: DOMICILIO, JURISDICCIÓN Y COMPETENCIA.- Las partes fijan como su domicilio la ciudad de Quito. Toda controversia o diferencia derivada de éste, será resuelta directamente entre las partes y, si esto no fuere factible, se solicitará la asistencia de un Mediador del Centro de Arbitraje y Mediación de la Cámara de Comercio de Quito. En el evento que el conflicto no fuere resuelto mediante este procedimiento, en el plazo de diez días calendario desde su inicio, pudiendo prorrogarse por mutuo acuerdo este plazo, las partes someterán sus controversias a la resolución de un árbitro, que se sujetará a lo dispuesto en la Ley de Arbitraje y Mediación, al Reglamento del Centro de Arbitraje y Mediación de la Cámara de comercio de Quito, y a las siguientes normas: a) El árbitro será seleccionado conforme a lo establecido en la Ley de Arbitraje y Mediación; b) Las partes renuncian a la jurisdicción ordinaria, se obligan a acatar el laudo arbitral y se comprometen a no interponer ningún tipo de recurso en contra del laudo arbitral; c) Para la ejecución de medidas cautelares, el árbitro está facultado para solicitar el auxilio de los funcionarios públicos, judiciales, policiales y administrativos, sin que sea necesario recurrir a juez ordinario alguno; d) El procedimiento será confidencial Y en derecho; e) El lugar de arbitraje serán las instalaciones del centro de arbitraje y mediación de la Cámara de Comercio de Quito; f) El idioma del arbitraje será el

español; y, g) La reconvencción, caso de haberla, seguirá los mismos procedimientos antes indicados para el juicio principal.

SÉPTIMA: ACEPTACIÓN.- Las partes contratantes aceptan el contenido del presente contrato, por ser hecho en seguridad de sus respectivos intereses.

En aceptación firman a los días del mes de del dos mil quince.

f) _____

f) _____

C.C. N° 1720242914

Instituto Superior Tecnológico Cordillera

Luis Clever Pupiales Farinango

CEDENTE

CESIONARIO

Agradecimiento.

Agradezco dios por darme la salud la sabiduría y a mis profesores

Los que me guiaron durante todo mi proceso de educación

Dedicatoria

A mi madre por ser un pilar fundamental de mi vida,

Hermanas/os por su apoyo infinito.

Índice general

Contenido	Páginas
Declaratoria.....	i
Contrato de Cesión Sobre Derechos Propiedad Intelectual.....	ii
Agradecimiento.....	vi
Dedicatoria.....	vii
Índice general.....	viii
Índice de Tablas.....	xii
Índice de Figuras.....	xvi
Resumen.....	xvii
ABSTRACT.....	xviii
Capítulo I: Antecedentes.....	1
1.01. Contexto.....	1
1.02. Justificación.....	2
1.03. Definición del Problema Central.....	3
Capítulo II Análisis de Involucrados.....	4
2.01. Requerimientos.....	4
2.01.1 Descripción del Sistema Actual.....	4
2.01.2 Visión y Alcance.....	5
2.01.3 Entrevistas.....	5
2.01.4 Matriz de Requerimientos.....	6

2.01.5 Descripción Detallada.....	6
2.02. Mapeo de Involucrados.....	14
2.03.0 Matriz de Involucrados	15
Capítulo III Problemas y Objetivos	18
3.01. Árbol de problemas.	18
3.02. Árbol de Objetivos.	19
3.03. Diagrama de casos de uso.....	20
3.04. Casos de uso de realización.....	23
3.05. Diagrama de secuencias del sistema.	27
3.06. Especificaciones de casos de uso.	29
Capítulo IV: Análisis de Alternativas	31
4.01. Matriz de análisis de alternativas	31
4.02. Matriz de impactos de Objetivos.....	32
4.03. Estándares para el diseño de clases.	33
4.04. Diagrama de clases	33
4.05. Modelo Lógico – Físico	34
4.06. Diagrama de Componentes.....	35
4.07. Diagrama de Estrategias.....	36
4.08. Matriz de Marco lógico.	37
4.09. Vistas Arquitectónicas.....	37
4.01.01. Vista lógica.	37

4.01.02. Vista Física.	37
4.01.03. Vista de Desarrollo.	38
4.01.04. Vista de procesos.	39
Capítulo V: Propuesta	41
5.01. Especificación de estándares de programación	41
5.02. Diseño de interfaces de usuario.	64
5.03. Especificación de pruebas de unidad.....	65
5.04. Especificación de pruebas de aceptación.	69
5.05. Especificación de pruebas de carga.	72
5.06. Configuración del Ambiente mínima /ideal.	74
Capítulo VI Aspectos Administrativos.....	76
6.01. Recursos	76
6.02. Presupuesto.....	76
6.03. Cronograma	77
Capítulo VII: Conclusiones y Recomendaciones.....	78
7.01. Conclusiones	78
7.02. Recomendaciones	78
Bibliografía	79
Anexos	80
Anexo 001	81
Anexo 002	82

Mapeo de Involucrados	82
Anexo 003	83
Anexo 005	85
Matriz de Marco lógico.....	85
Anexo 006	86
Cronograma.....	86
Anexo 007	88
Manual de instalación y configuración básica del sistema	88
Manual de usuario	96
Introducción	96
Ventana de Administración	97
Usuario	97
Auditoria.....	98
Producto.....	100
Manual técnico	102

Índice de Tablas

<i>Contenido</i>	<i>Páginas</i>
Tabla 1.....	3
Matriz de fuerzas T	3
Tabla 2.....	6
Diseño de entrevistas.....	6
Tabla 3.....	7
Creación de usuario nuevo.	7
Tabla 4.....	8
Ingresos de usuarios al sistema.	8
Tabla 5.....	9
Registro de proveedores.....	9
Tabla 6.....	10
Ingreso de mercadería.	10
Tabla 7.....	11
Reporte Stock e inventario	11
Tabla 8.....	12
Reporte de productos por categoría.....	12
Tabla 9.....	13
Reporte de productos.....	13

Tabla 10.....	14
Al Software se accede a través de todo tipo de navegador	14
Tabla 11.....	16
Descripción detallada de los actores directos.....	16
Tabla 12.....	17
Descripción detallada de los actores indirectos.	17
Tabla 13.....	23
Crear usuario en sistema.	23
Tabla 14.....	25
Ingreso de producto al sistema.....	25
Tabla 15.....	26
Egreso de productos.	26
Tabla 16.....	29
Casos de uso crear usuario.	29
Tabla 17.....	29
Casos de uso ingreso de productos a bodega.	29
Tabla 18.....	30
Casos de uso despacho de productos.	30
Tabla 19.....	31
Matriz de análisis.	31

Tabla 20.....	32
Matriz de objetivos del sistema.....	32
Tabla 21.....	66
Módulo de prueba Administración.....	66
Tabla 22.....	67
Módulo de prueba Administración Login.....	67
Tabla 23.....	67
Módulo de prueba Productos.....	67
Tabla 24.....	68
Módulo de prueba Ventas (Egresos).....	68
Tabla 25.....	68
Módulo de prueba Reportes.....	68
Tabla 26.....	70
Módulo de prueba Administración.....	70
Tabla 27.....	70
Módulo de prueba Ingreso de productos.....	70
Tabla 28.....	71
Módulo de prueba Ventas (Egresos).....	71
Tabla 29.....	71
Módulo de prueba Auditoria.....	71

Tabla 30.....	73
Pruebas de servidor	73
Tabla 31.....	74
Interconexión de usuarios	74
Tabla.32.....	76
Recursos utilizados.....	76
Tabla 33.....	77
Recursos Utilizados.....	77

Índice de Figuras

Contenido	Páginas
Figura. 1 Árbol de problemas con sus causas y efectos.....	18
Figura. 2 Componentes finalidad propuesta.	19
Figura. 3 crear usuario en sistema.....	20
Figura. 4 Caso de usos particulares.....	21
Figura. 5 Despacho producto	22
Figura. 6 Crear usuario en el Sistema.	23
Figura.7 Ingreso de productos al sistema.	24
Figura. 8 Egreso de productos de bodega.	25
Figura. 9 Diagrama de ingreso de productos.....	27
Figura. 10 Diagrama de secuencias Despacho producto	28
Figura. 11 Diagrama de componentes del sistema desarrollado	35
Figura. 12 Estrategias tomadas.....	36
Figura. 13 Vista lógica del sistema creado	37
Figura. 14 Se demuestra la vista física del sistema creado.....	38
Figura. 16 Se muestra la ventana de auditoria	64
Figura 17 Ventana de productos.....	65

Resumen

Con el pasar de los años la tecnología ha crecido exponencialmente en todos los campos, es así en la ciencia en la industria en la educación, en la agricultura en el hogar, en tal virtud es así como se al logrado facilitar al ser humano en la realización de varias actividades ya sean en el hogar en el trabajo o en sus momentos de esparcimiento creando así una relación entre tecnología y ser humano muy grata y compleja a la vez, grata por que el ser humano se siente más agradable para consigo mismo por sus avances tecnológicos y técnicos , grandes adelantos que hoy en día lo convierten al ser humano en un ser virtuoso emprendedor y creador de sus propios sueños.

Es por tal razón que gracias a esos sueños y avances tecnológicos hoy creamos una pequeña herramienta que facilita el control e inventario de bodega la misma que se convierte en un pequeño aporte al desarrollo de la empresa Globalconnection y la matriz productiva del país sin dejar de lado, el plan nacional del buen vivir generado por las nuevas políticas de estado vigentes al momento.

ABSTRACT

Over the years technology has grown exponentially in all fields, so in science industry in education, agriculture in the home, such virtue is how I was to successfully facilitated the human being in the several activities whether at home at work or in their leisure time thus creating a relationship between technology and human being pleasant and complex at the same time pleasing that the human being feels nicer to himself for his technological and technical advances, breakthroughs that today make humans virtuoso be an entrepreneur and creator of his own dreams.

It is for this reason that thanks to those dreams and technological advances now create a small tool that facilitates inventory control and warehouse the same becomes a small contribution to the development of the company Globalconnection and productive matrix of the country without leaving The national plan of good living generated by new state policies in effect at the time.

Capítulo I: Antecedentes

1.01. Contexto.

En la actualidad toda industria, empresa está buscando su desarrollo a nivel regional o de país por esta razón muchas de la empresas se han visto obligadas a canalizar recursos humanos y monetarios a las nuevas tecnologías que mantienen la globalización del mundo mismas que hacen de cada departamento de una empresa sea más funcional, eficiente y productiva pero para lograr estas metas como tal los gerentes se han visto en la necesidad de importar dichas herramientas a costos elevados.

Hoy a través de las nuevas políticas de estado como son la nueva matriz productiva se busca fortalecer a la industria, empresas del Ecuador a través de sus propias herramientas técnicas y tecnológicas a costos mucho más factibles y al alcance de todos así también se busca que este desarrollo sea un pilar fundamental para el crecimiento del país y su región.

Si bien es cierto hoy en día en Ecuador muchas de las empresas trabajan con software importado para muchas de sus áreas como son: la administración de bodega, administración de personal área de contabilidad entre otras.

Por tal motivo se elaborara el diseño y construcción de un software para el área de control de inventario de bodegas vinculándonos así a la matriz productiva del Ecuador del año 2014 en la que se plantea de forma clara que el Ecuador hasta antes de esta fecha fue un país consumista importador no solo de productos para la alimentación sino también un importador de software, tecnología y más.

(Desarrollo, planificacion.gob.ec, 2012)

SISTEMATIZACIÓN DE LOS PROCESOS DE CONTROL DE INVENTARIO A TRAVÉS DE UN SOFTWARE ORIENTADO A LA WEB PARA LA EMPRESA GLOBALCONNECTION DE LA CIUDAD DE QUITO

1.02. Justificación

El presente trabajo está desarrollado en base a una necesidad latente de la empresa pero no así de vital importancia para su funcionamiento como tal ya que a pesar de no tener dicha aplicación (software) ha encontrado un punto de equilibrio dentro del mercado y su desarrollo en el país lo que la hacen cada día mejor.

Hoy en día sabemos que la tecnología ha crecido en forma significativa lo que conlleva a que las industrias, empresas accedan a este recurso para mejorar sus procesos y facilitar a los empleados a través de herramientas técnicas y tecnológicas sus niveles de rendimiento y capacidad de trabajo razón por la que se insertara este software en la empresa a partir de su elaboración y finalización la que permitirá mantener una optimización de tiempos y recursos manejando informes detallados de productos así como numero de incidencia de riesgos que tienen los productos tales como pérdidas o una sobrecompra de los mismos de tal manera dicho software ayudara llevando un control de inventario global de los productos que están almacenados en bodegas y de esta forma ayudar con un mejor desarrollo empresarial.

(Desarrollo, planificacion.gob.ec, 2009)

1.03. Definición del Problema Central.

Tabla 1

Matriz de fuerzas T

MATRIZ DE FUERZAS T					
Problemática Agravada	Problema Central				Problemática Resuelta
Perdida de productos y clientes insatisfechos	Reportes de inventario de productos atrasados				Bodega funcional con reportes de productos al día
Fuerzas Impulsadoras	Intensidad	Potencial Cambio	Intensidad	Potencial Cambio	Fuerzas Bloqueadoras
Innovación tecnológica	3	4	4	5	Falta de capacitación
Minimización de riesgos de los Productos (perdidas, daños)	4	5	3	4	Mala administración de recursos tecnológicos y humanos
Implementación del personal necesario	3	4	4	5	Falta de interés de los empleados
Investigación de productos con mayor demanda	4	5	4	5	Inexistencia de un software de control propio
Cumplimiento de los procesos de control de inventario	3	5	3	5	Desconocimiento de los procesos
Cientes satisfechos	4	4			

Análisis de la matriz de fuerzas

Dentro de los datos obtenidos de la matriz de fuerzas T se puede notar que la empresa debe implementar e innovar su sector técnico y tecnológico para mejorar los procesos de control de bodega así como también contar con un personal calificado para sus respectivos departamentos y así mantener sus reportes al día con lo que lograra incrementar su productividad y eficiencia dentro del sector al que está orientada.

Capítulo II Análisis de Involucrados

2.01. Requerimientos.

2.01.1 Descripción del Sistema Actual.

La empresa Globalconnection actualmente se dedica a la venta equipos informáticos como son partes, piezas y suministros, hoy en día lleva su proceso de inventario de bodega en niveles aceptables puesto que se trabaja llevando todo en registros manuales y otros en una herramienta de trabajo como es la hoja de cálculo de Excel que es parte de la suite de Microsoft que se la implementa dentro de los equipos informáticos básicos al momento de la compra de los mismos.

La hoja de cálculo de Excel Es una herramienta muy importante sobre todo para las empresas, ya que en este programa pueden realizar su contabilidad, presupuestos, llevar la facturación, análisis de datos, graficar datos, agendas de clientes/proveedores, todo tipo de base de datos, es así también programa que permite realizar operaciones con números Es útil para realizar desde simples sumas hasta cálculos complejos. etc.

En la actualidad la empresa trabaja todos sus procesos en esta suite informática, si bien es cierto mantiene resultados favorables.

2.01.2 Visión y Alcance

La herramienta desarrollada para la empresa mantendrá sus niveles de control óptimo y será la innovación técnica y tecnológica, ejemplo, desarrollo e innovación dentro del territorio, el software buscara ser líderes y pionero por trabajar con la necesidades propias de la empresa y apegado a la nueva matriz productiva del Ecuador.

La elaboración y construcción del nuevo software informático está apegada a las necesidad y requerimientos de la empresa como son la de mantener su propio software de control de inventario de bodega el cual permitirá mejorar los procesos de la misma así como la optimización de recursos humanos y tecnológicos que mantiene la empresa.

En esta sección detallamos los módulos que se crearan:

- Módulo de mantenimiento
- Módulo de seguridad

2.01.3 Entrevistas

En esta sección desarrollaremos dos modelos de entrevistas que son las que nos facilitaran la ejecución y elaboración del software.

Tabla 2

Diseño de entrevistas

Diseño de entrevistas		
Identificador: 001		
Preguntas	Objetivos	Análisis Posterior
¿Cuál es el proceso que desearía automatizar?	Conocer el proceso que se automatizara	Levantamiento de información necesario para automatizar dicho proceso
¿Qué personas deben tener accesos al sistema?	Controlar el ingreso y salida de datos.	Datos e información segura y confiable.
¿Se debe llevar un control de los proveedores?	Llevar un registro de proveedores	Mantener un listado de los proveedores que abastecen a la empresa a través de la asignación de un
¿Se debe generar reporte de novedades?	Llevar un registro de siniestros de los productos.	Llevar registro de información de casos especiales de los productos
¿Cuántas personas accederán al sistema simultáneamente?	Conocer el número de usuarios que estarán utilizando el sistema constantemente.	El software pueda soportar mayor número de usuarios en línea
¿Qué volumen de información se maneja diariamente?	Conocer la cantidad de información que maneja la empresa por día	Software capaz de soportar mayor número de transacciones por usuarios

2.01.4 Matriz de Requerimientos.

En esta sección se describen los requerimientos básicos que solicita la empresa para su funcionamiento y ejecución del software.

Ver tabla Anexo 001.

2.01.5 Descripción Detallada

Tabla 3

Creación de usuario nuevo.

Descripción del Requerimiento		Estado	Implementación
Registro de usuarios			
Creado por	Luis Pupiales	Actualizado por	Luis Pupiales
Fecha de creación	10 enero 2015	Fecha de actualización	10 enero 2015
Identificador	REF001		
Tipo de requerimiento	Critico	Tipo de requerimiento	Funcional
Datos de entrada	Información del usuario		
Descripción	Ingreso de información detallada por cada usuario		
Datos de salida	Información de usuario actualizada		
Resultados esperados	Actualización de la base de datos		
Origen	Administrador de sistemas		
Dirigido a	Gerencia, Administrador del sistema		
Prioridad	5		
Requerimientos asociados	Ninguno		
ESPECIFICACIONES			
Precondiciones	El ingreso de los usuarios será para asignarles un password con el que ingresarán al sistema		
Pos condiciones	Solo el administrador de sistemas o el responsable del sistema podrá ingresar asignar los password		
Criterios de aceptación			

Análisis.

En esta tabla se describe los principales datos que deberá tener el requerimiento funcional para su ejecución tales como el ingreso de los datos principales del usuario para luego asignarle su respectivo password previo al análisis de gerencia y el administrador del sistema.

Tabla 4

Ingresos de usuarios al sistema.

Descripción del Requerimiento		Estado	Implementación
Ingresos de usuarios al sistema			
Creado por	Luis Pupiales	Actualizado	Luis Pupiales
Fecha de creación	10 enero 2015	por Fecha de actualización	10 enero 2015
Identificador	REF002		
Tipo de requerimiento	Critico	Tipo de requerimiento	Funcional
Datos de entrada	Password de usuarios		
Descripción	Ingreso del password de los usuarios para acceder al sistema		
Datos de salida	Usuario identificado		
Resultados esperados	Ingreso al sistema		
Origen	Gerencia, Administrador del sistema		
Dirigido a	Usuarios del sistema		
Prioridad	5		
Requerimientos asociados	Ninguno		
ESPECIFICACIONES			
Precondiciones	El usuario debe estar ingresado en la base de datos para que pueda acceder al sistema		
Pos condiciones	El usuario debe tener su password		
Criterios de aceptación	Ninguno		

Análisis.

En esta tabla se describe como el usuario accederá al sistema luego de su calificación y aprobación de password generado por gerencia y el administrador del sistema.

Tabla 5

Registro de proveedores.

Descripción del Requerimiento		Estado	Implementación
Registro de proveedores			
Creado por	Luis Pupiales	Actualizado por	Luis Pupiales
Fecha de creación	10 enero 2015	Fecha de actualización	10 enero 2015
Identificador	REF003		
Tipo de requerimiento	Crítico	Tipo de requerimiento	Funcional
Datos de entrada	Información del proveedor		
Descripción	Ingreso de información del proveedor nombres dirección ruc		
Datos de salida	Proveedor activo		
Resultados esperados	Base de datos actualizada		
Origen	Gerencia, Departamento de compras		
Dirigido a	Usuarios del sistema		
Prioridad	5		
Requerimientos asociados	Ninguno		
ESPECIFICACIONES			
Precondiciones	El usuario debe estar ingresado en la base de datos para que pueda acceder al sistema		
Pos condiciones	El usuario debe tener su password activo		
Criterios de aceptación	Ninguno		

Análisis.

En esta tabla se describe como ingresara el proveedor a la la base de datos para luego realizar las transacciones respectivas.

Tabla 6*Ingreso de mercadería.*

Descripción del Requerimiento		Estado	Implementación
Ingreso de mercadería			
Creado por	Luis Pupiales	Actualizado por	Luis Pupiales
Fecha de creación	10 enero 2015	Fecha de actualización	10 enero 2015
Identificador	REF004		
Tipo de requerimiento	Critico	Tipo de requerimiento	Funcional
Datos de entrada	Información de los productos		
Descripción	Ingreso detallado de los productos nombre ubicaciones cantidades modelo tipo		
Datos de salida	Información completa de los productos		
Resultados esperados	Base de datos actualizada.		
Origen	Bodega.		
Dirigido a	Usuarios del sistema.		
Prioridad	5		
Requerimientos asociados	Ninguno		
ESPECIFICACIONES			
Precondiciones	El personal encargado de bodega a través de su respectivo password ingresara al sistema he ingresara los productos que a sus instalaciones arriben desde las diferentes bodegas de los proveedores.		
Pos condiciones	Toda mercadería que llegue se ingresará a bodegas a través del sistema.		
Criterios de aceptación	Ninguno		

Análisis.

En esta tabla se describe como ingresara el producto a las bodegas y a través del mecanismo que se debe ingresar.

Tabla 7

Reporte Stock e inventario

Descripción del Requerimiento		Estado	Implementación
Reporte Stock e inventario			
Creado por	Luis Pupiales	Actualizado por	Luis Pupiales
Fecha de creación	10 enero 2015	Fecha de actualización	10 enero 2015
Identificador	REF005		
Tipo de requerimiento	Critico	Tipo de requerimiento	Funcional
Datos de entrada	Ingreso al sistema		
Descripción	Detalle de stock de productos en bodegas		
Datos de salida	Cantidades de los productos		
Resultados esperados	Revisión y actualización de la base de datos.		
Origen	Gerencia		
Dirigido a	Usuarios del sistema		
Prioridad	5		
Requerimientos asociados	Ninguno		
ESPECIFICACIONES			
Precondiciones	Para que el usuario pueda ver los niveles de stock es necesario que lo haga desde el sistema		
Pos condiciones	El usuario puede mirar los reportes solo si tiene password		
Criterios de aceptación	Ninguno		

Análisis.

En esta tabla se describe como el usuario puede ver los niveles de stock de las bodegas matriz.

Tabla 8*Reporte de productos por categoría*

Descripción del Requerimiento		Estado	Análisis
Reporte de productos por categoría			
Creado por	Luis Pupiales	Actualizado	Luis Pupiales
Fecha de creación	10 enero 2015	por Fecha de actualización	10 enero 2015
Identificador	RNF001		
Tipo de requerimiento	No Critico	Tipo de requerimiento	No Funcional
Datos de entrada	Ingreso al sistema		
Descripción	En el sistema se debe encontrar un reporte por categoría de productos.		
Datos de salida	Reportes		
Resultados esperados	Detalle de productos por categoría		
Origen	Sistema		
Dirigido a	Usuarios del sistema.		
Prioridad	2		
Requerimientos asociados	Ninguno		
ESPECIFICACIONES			
Precondiciones	Productos deben estar separados por categorías.		
Pos condiciones	Ninguno		
Criterios de aceptación	Ninguno		

Análisis.

En esta tabla se describe los reportes que al usuario le agradaría obtener dentro del sistema.

Tabla 9

Reporte de productos

Descripción del Requerimiento		Estado	Análisis
Reporte de productos con mayor demanda			
Creado por	Luis Pupiales	Actualizado por	Luis Pupiales
Fecha de creación	10 enero 2015	Fecha de actualización	10 enero 2015
Identificador	RNF002		
Tipo de requerimiento	No Critico	Tipo de requerimiento	No Funcional
Datos de entrada	Ingreso al sistema		
Descripción	Observar un detalle de los productos con mayor demanda		
Datos de salida	Reporte		
Resultados esperados	Productos de mayor demanda.		
Origen	Sistema		
Dirigido a	Usuarios del sistema		
Prioridad	2		
Requerimientos asociados	Ninguno		
ESPECIFICACIONES			
Precondiciones	Ninguno		
Pos condiciones	Ninguno		
Criterios de aceptación	Ninguno		

Análisis.

En esta tabla se describe los productos con mayor demanda que serían vitales para el departamento de marketing si lo hubiere.

Tabla 10

Al Software se accede a través de todo tipo de navegador

Descripción del Requerimiento	Estado	Análisis
Al Software se accede a través de todo tipo de navegador.		
Creado por	Luis Pupiales	Actualizado por Luis Pupiales
Fecha de creación	10 enero 2015	Fecha de actualización 10 enero 2015
Identificador	RNF003	
Tipo de requerimiento	No Critico	Tipo de requerimiento No Funcional
Datos de entrada	URL página.	
Descripción	Ingresar la url del sistema al que se accederá.	
Datos de salida	En pantalla la página principal del sistema	
Resultados esperados	En pantalla venta principal del login de usuarios al sistema para ingresar al mismo.	
Origen	Sistema	
Dirigido a	Usuarios del sistema	
Prioridad	1	
Requerimientos asociados	Ninguno	
ESPECIFICACIONES		
Precondiciones	La url del sistema debe ser verídica	
Pos condiciones	Acceso al internet.	
Criterios de aceptación	Ninguno	

Análisis.

En esta tabla se describe en que navegador se ejecuta el software creado.

2.02. Mapeo de Involucrados.

En esta sección se detallan todos los actores que estarán involucrados en la elaboración del software.

Ver anexo 002

2.03.0 Matriz de Involucrados

En esta sección realizaremos un detalle de los involucrados con la elaboración del software de inventario de bodega así como los actores responsables del cumplimiento de los procesos que se enmararon en la imagen anterior.

Ver anexo 003

Análisis de la Matriz de Involucrados

En esta sección apreciaremos los principales actores responsables de las actividades dentro de la empresa para mantener la bodega en óptimas condiciones, con reportes al día así como los procesos por los que debe pasar un producto, que son de vital importancia para mejorar su productividad.

Tabla 11

Descripción detallada de los actores directos.

Actores Involucrados	Intereses sobre el problema central	Problemas Percibidos	Recursos, Mandatos y Capacidades	Intereses sobre el Proyecto	Conflictos Potenciales
Gerente general	Decidir y hace cumplir las normas y procesos para los empleados.	El proceso de la administración creado por la empresa no se lo ejecuta paso a paso.	Mantienen un proceso de administración creado por la empresa.	Acceder a la información línea	Ninguno
Empleado responsable de bodega.	Llevar de mejor manera su control e inventario.	No se lleva el control de bodega adecuado por falta de herramientas tecnológicas acordes.	Mantienen un proceso de administración creado por la empresa.	Acceder a la información línea	Ninguno
Departamento de ventas.	Conocer los stocks de los productos para su facturación y entrega a los clientes.	Demora en la entrega de los productos e inexistencia de productos en bodega.	Mantienen un proceso de administración creado por la empresa.	Acceder a la información línea	Ninguno
Departamento de contabilidad.	Pagos a los proveedores previo a las compras realizadas	Demora en pago a los proveedores por no tener información actualizada.	Mantienen un proceso de administración creado por la empresa.	Acceder a la información línea	Ninguno
Departamento de compras.	Realizar los pedidos a los proveedores a tiempo.	Las compras se la realizan luego de no tener existencias en bodega.	Mantienen un proceso por la empresa.	Acceder a la información línea	Ninguno
Desarrollador.	Desarrollar el software e implementar.	No existe un software adecuado para sus procesos de inventario.	Mantienen un proceso creado por la empresa.	Acceder a la información línea	Ninguno

Tabla 12

Descripción detallada de los actores indirectos.

Actores Involucrados	Intereses sobre el problema central	Problemas Percibidos	Recursos, Mandatos y Capacidades	Intereses sobre el Proyecto	Conflictos Potenciales
Clientes.	Ninguno	Demora en las entregas de los productos.	Ninguno	Ninguno	Ninguno
Proveedores.	Ninguno	Demora en el pago sus ventas.	Ninguno	Ninguno	Ninguno

Capítulo III Problemas y Objetivos

3.01. Árbol de problemas.

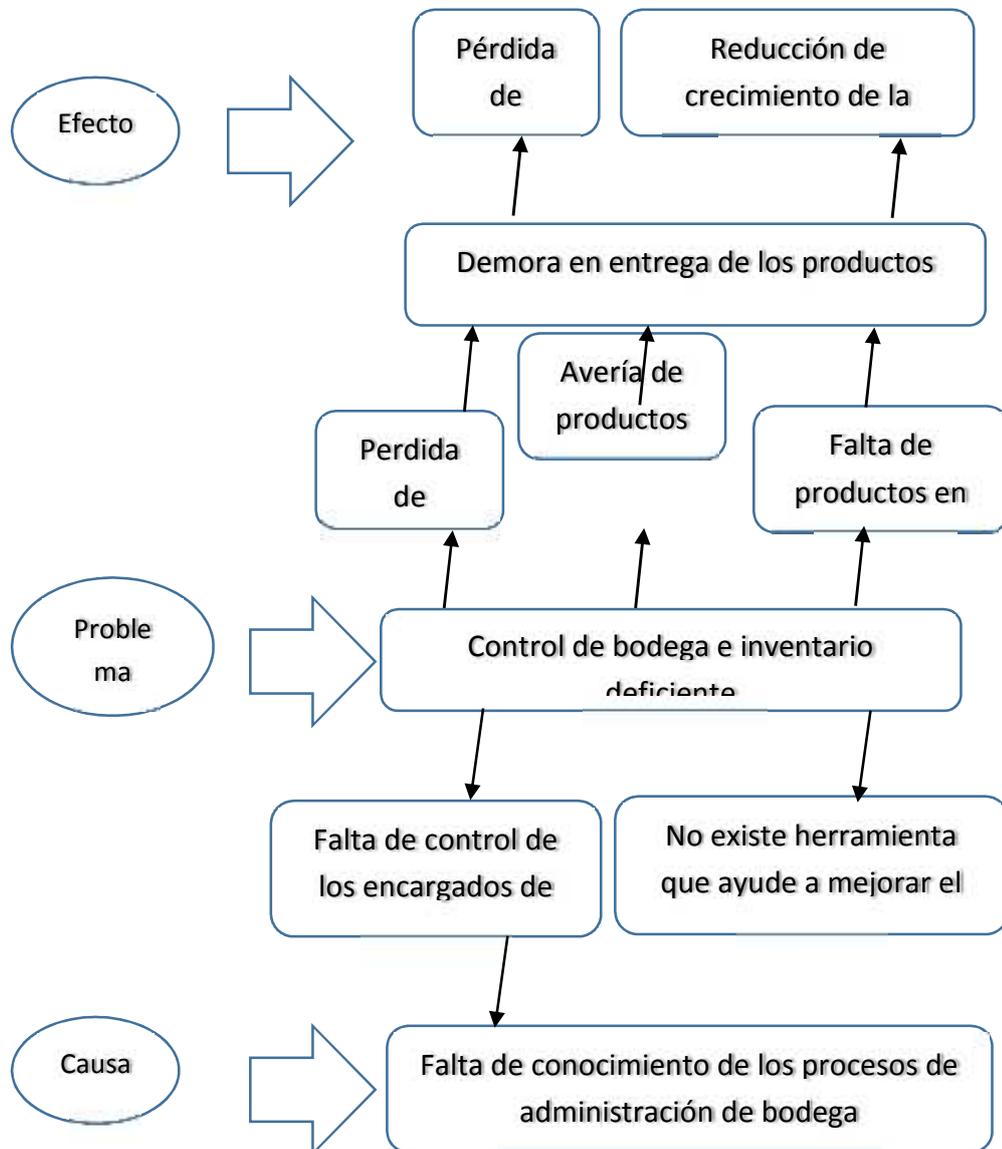


Figura. 1 Árbol de problemas con sus causas y efectos.

En esta figura se refleja la problemática que conlleva a la construcción del software de control de bodega e inventario tomando en cuenta dos aspectos fundamentales como son las causas y los efectos que estas repercuten en la elaboración del mismo.

3.02. Árbol de Objetivos.

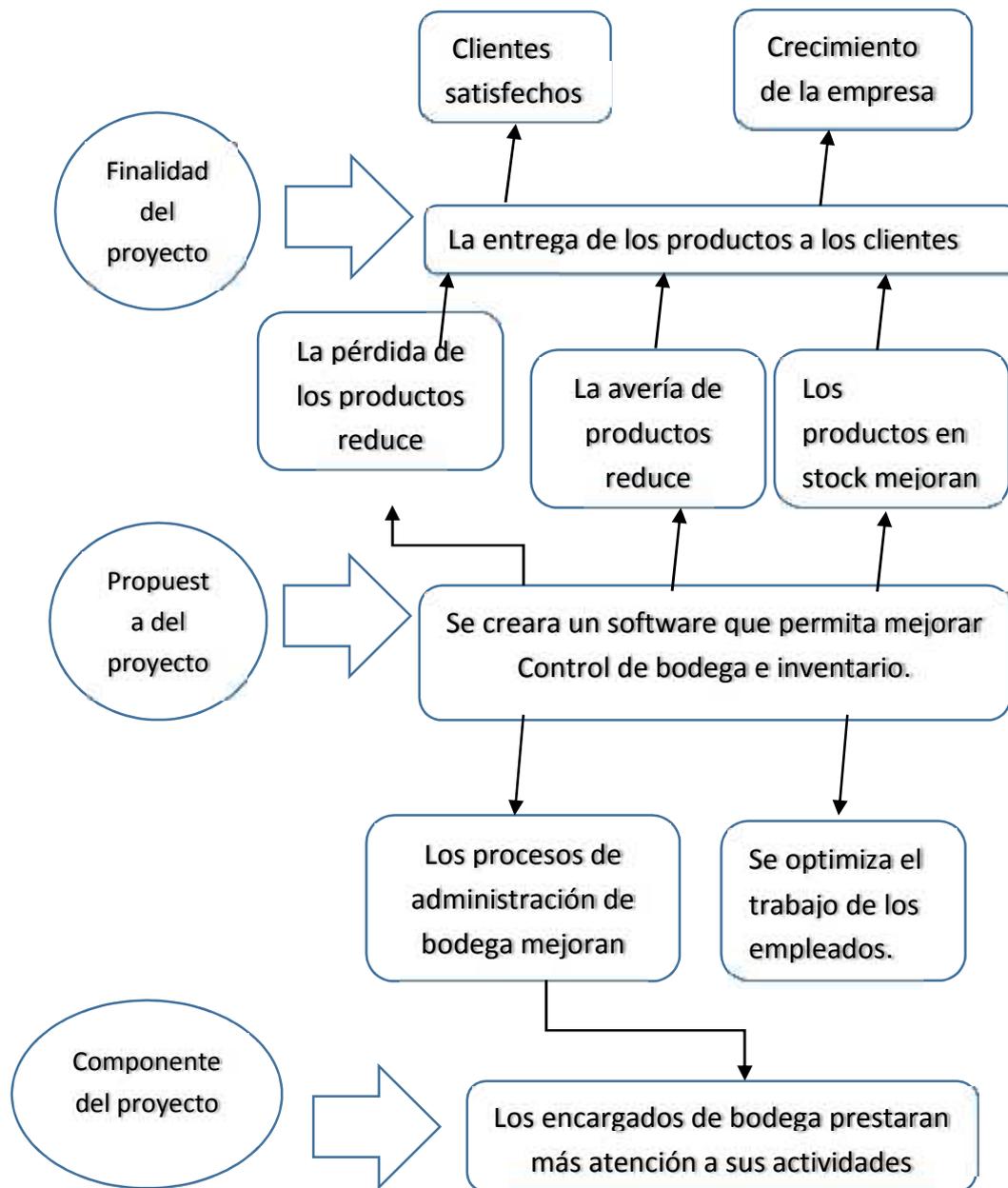


Figura. 2 Componentes finalidad propuesta.

Análisis

Se describen los principales componentes y sus diferentes soluciones con respecto a la producción del software.

3.03. Diagrama de casos de uso.

En esta sección se detallan gráficamente los casos de uso de los procesos que se sigue en la administración y control de bodega.

Diagrama casos de uso crear usuario (responsable de bodega)

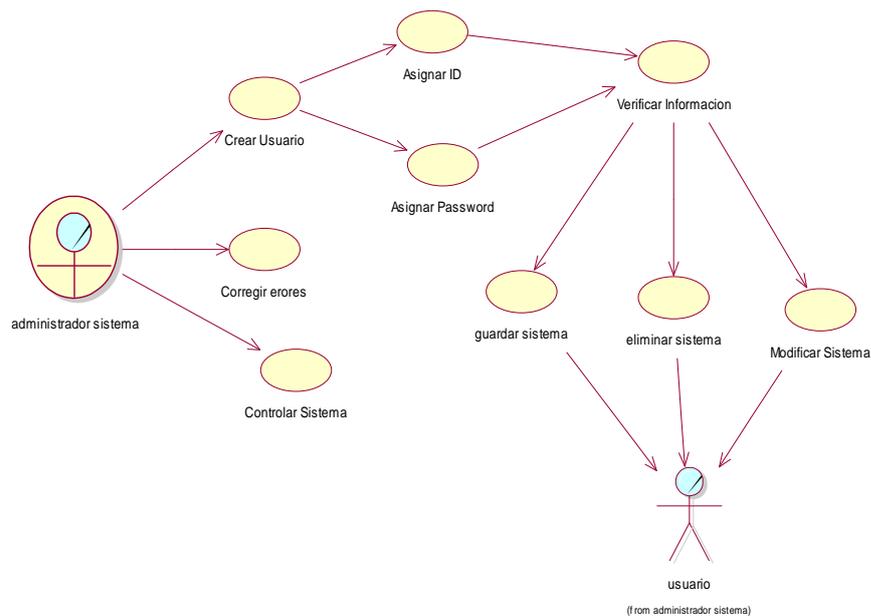


Figura. 3 crear usuario en sistema

Análisis.

En esta imagen podemos observar cómo se crea al usuario del sistema o responsable de bodega mismo que trabajara con el sistema durante su estadía en la empresa.

Diagrama de casos de uso ingresar producto.

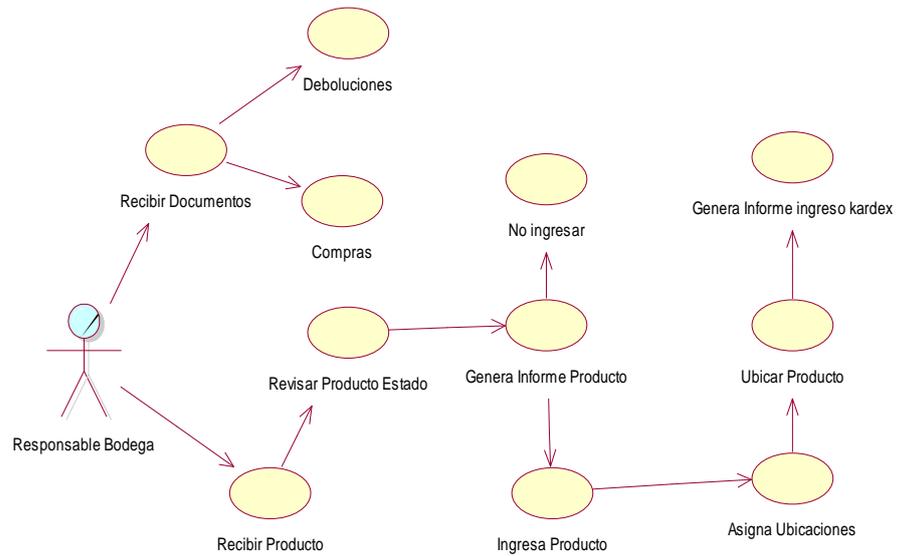


Figura. 4 Caso de usos particulares

Análisis.

En esta imagen describimos los casos de uso que se generan por parte del Sistema Central al cual el usuario ingresara para realizará las diferentes transacciones, peticiones información necesaria que se utilizara dentro del mismo como son el ingreso del producto y sus respectivas ubicaciones.

3.04. Casos de uso de realización.

Caso: CU001

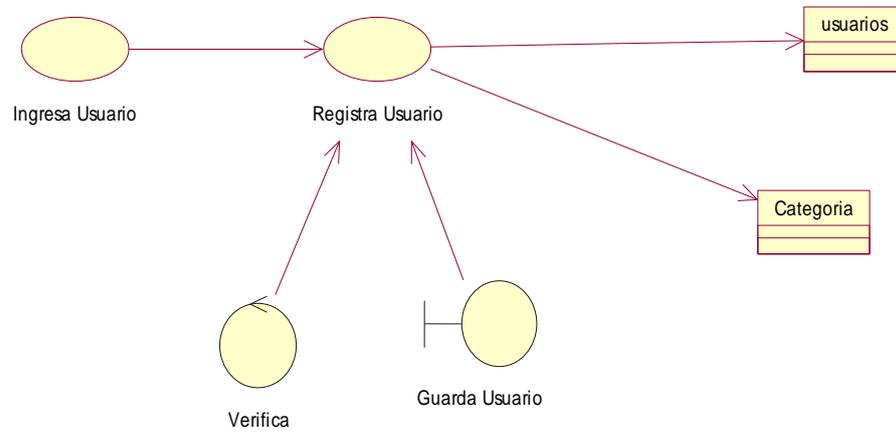


Figura. 6 Crear usuario en el Sistema.

Tabla 13

Crear usuario en sistema.

Nombre	Crear usuarios en sistema
Identificador	CU001
Responsabilidades	Ingreso al sistema a través de la inserción de id y password para la creación y validación de usuario.
Tipo	Sistema
Referencias de casos de uso	CU001
Referencias requisitos	RF001
PRECONDICIONES	
Asignar un usuario para el sistema	
Determinar el estado del usuario en el sistema	
POSCONDICIONES	
El administrador del sistema debe estar registrado para ingresar a los usuarios	
SALIDAS PANTALLA	
Interfaz del administrador con varios campos para ingresar información detallada de cada Usuario	

Análisis:

En esta sección se describe los pasos que el usuario debe hacer para ingresar al sistema y manipular como tal la herramienta, siempre y cuando este registrado en la base de datos matriz de la empresa.

Caso: CU002

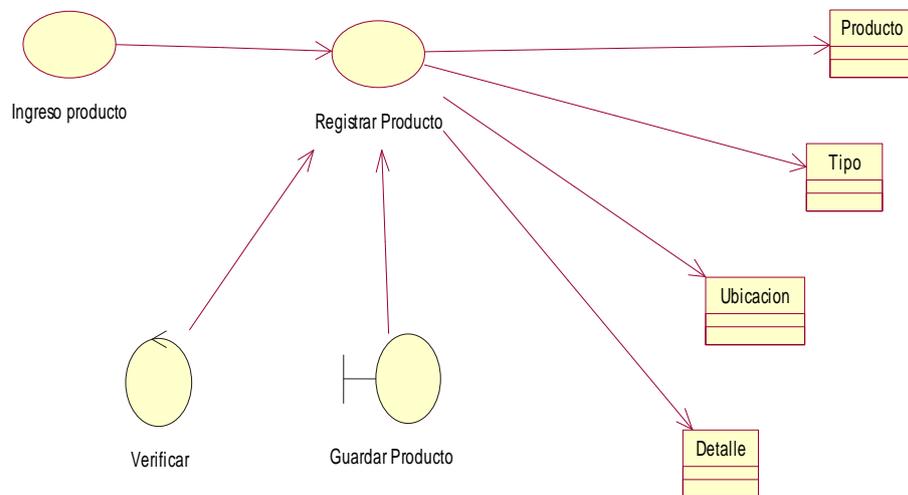


Figura.7 Ingreso de productos al sistema.

Tabla 14

Ingreso de producto al sistema

Nombre	Ingreso de productos al sistema
Identificador	CU002
Responsabilidades	Se ingresan los productos con sus respectivos detalles
Tipo	Sistema
Referencias de casos de uso	CU002
Referencias requisitos	RF002
PRECONDICIONES	
Ingresar información del producto que arriba a las bodegas Determinar el estado, la cantidad de producto que ingresa a bodega	
POSCONDICIONES	
El empleado responsable debe estar activo en el sistema para ingresar la información correspondiente.	
SALIDAS PANTALLA	
Interfaz de ingreso de productos con varios campos para ingresar información detallada de cada producto y sus respectivos detalles.	

Caso: CU003

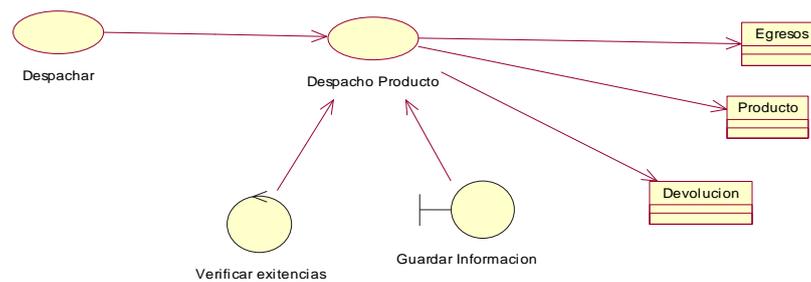


Figura. 8 Egreso de productos de bodega.

Tabla 15

Egreso de productos.

Nombre	Egreso de productos
Identificador	CU003
Responsabilidades	Detallar el egreso de los productos
Tipo	Sistema
Referencias de casos de uso	CU003
Referencias requisitos	RF003
PRECONDICIONES	
El empleado responsable debe estar activo en el sistema para ingresar la información correspondiente.	
POSCONDICIONES	
El responsable de bodega debe estar registrado para dar el seguimiento correspondiente a los productos	
SALIDAS PANTALLA	
Interfaz del administrador con varios campos para ingresar información detallada de cada Usuario	

3.05. Diagrama de secuencias del sistema.

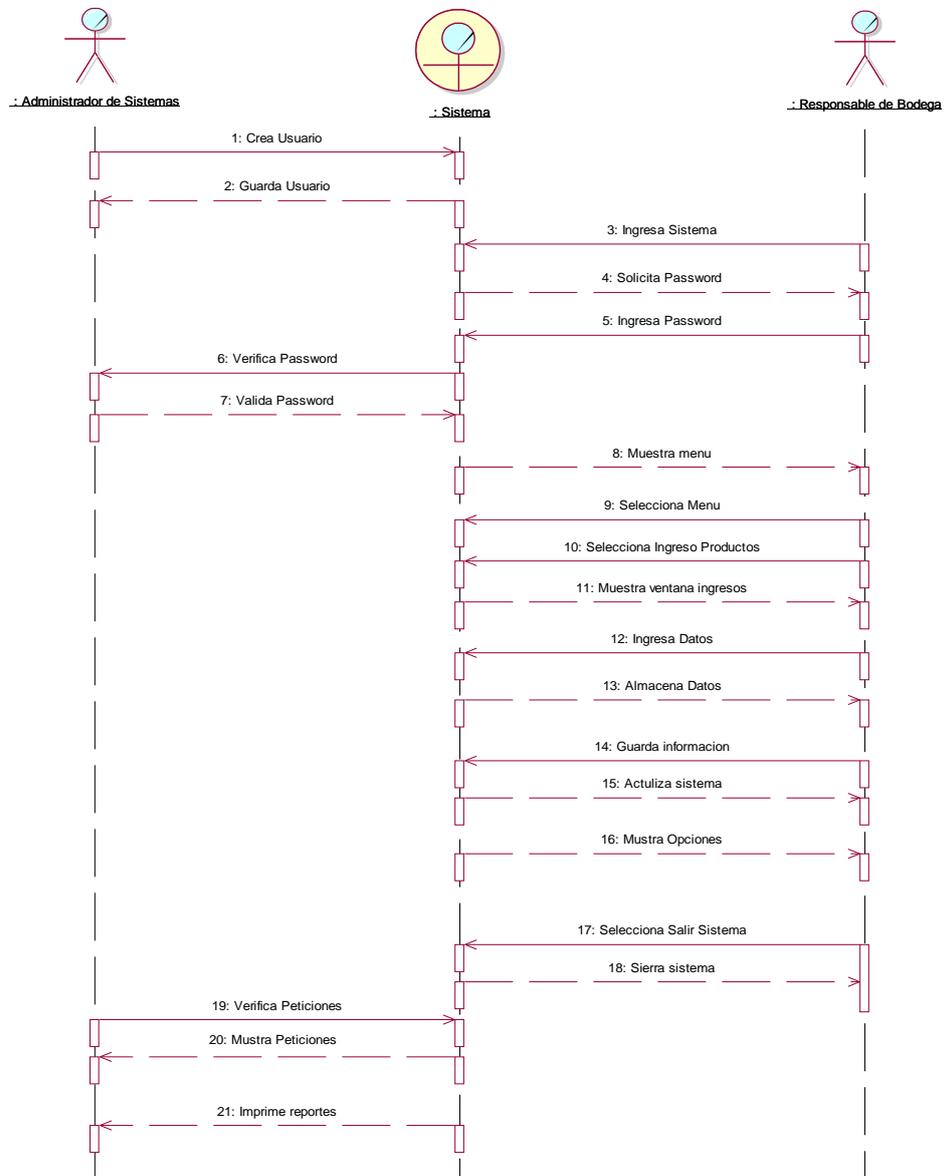


Figura. 9 Diagrama de ingreso de productos.

Diagrama de secuencias Despacho producto

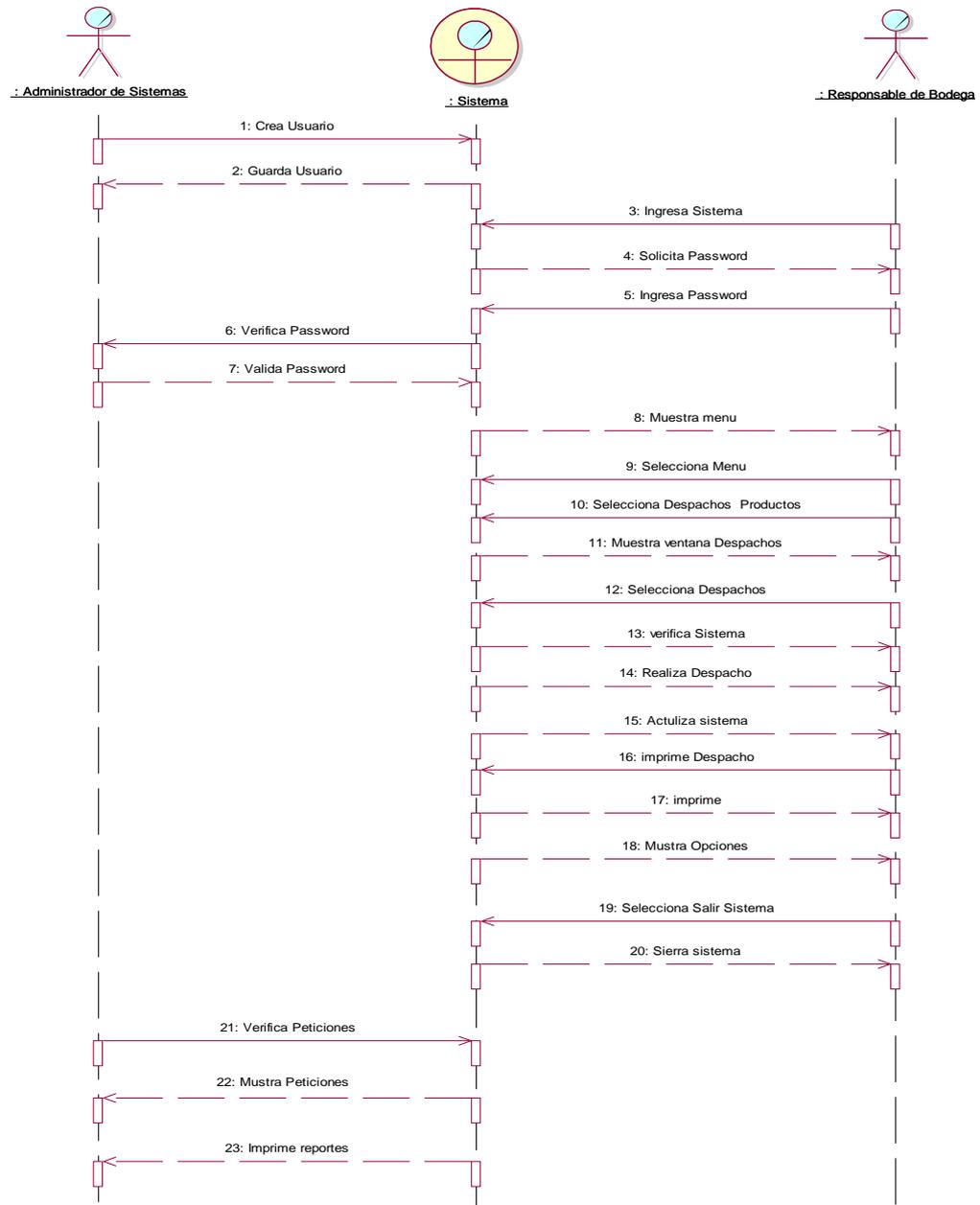


Figura. 10 Diagrama de secuencias Despacho producto

3.06. Especificaciones de casos de uso.

Tabla 16

Casos de uso crear usuario.

Casos de Uso	Crear usuario
Identificador	CU001
CURSO TIPICO DE EVENTOS	
Usuario	Sistema
Administrador de sistema ingresa al sistema y asigna un ID y PASSWORD a los empleados.	Valida y verifica información de los empleados para luego actualizar la información en la base.
CURSOS ALTERNATIVOS	
Valoración de los datos	

Tabla 17

Casos de uso ingreso de productos a bodega.

Casos de uso	Ingreso de productos a bodega
Identificador	CU002
CURSO TIPICO DE EVENTOS	
Usuario	Sistema
Ingresa al sistema y luego carga información detallada de cada producto que llega a bodegas previo a las compras realizadas	Sistema verifica información y valida datos de los productos para posterior actualizar y almacenar en la base.
CURSOS ALTERNATIVOS	
Valoración de los datos	

Tabla 18

Casos de uso despacho de productos.

Casos de Uso	Despacho de Productos
Identificador	CU003
CURSO TIPICO DE EVENTOS	
Usuario	Sistema
Responsable ingresa al sistema para llenar el formulario de salida de productos	Valida información de salida de los productos y actualiza la base de datos.
CURSOS ALTERNATIVOS	
Valoración de los datos	

Capítulo IV: Análisis de Alternativas

4.01. Matriz de análisis de alternativas

Tabla 19

Matriz de análisis.

Variables Estrategias	Necesidades de recursos			Factibilidad Política	Duración del Proyecto	Puntaje	Prioridad
	Humanos	Técnicos	Financieros				
Se asigna recursos para mejorar el departamento de tecnología	3	5	5	4	5	22	Segunda
Se capacita al personal sobre las tic	4	4	5	4	4	21	Tercera
Se fomenta una cultura tecnológica	4	4	3	3	4	18	Cuarta
Se asigna personal adecuado al departamento de tecnología	5	4	5	4	5	23	Primera
Total	16	17	18	15	18	84	

Análisis.

En la tabla 17 se especifican los principales objetivos que se ejecutarán dentro de las instalaciones de la empresa con motivo de cambiar su forma de manejar los respectivos documentos e informes de las diferentes labores que ahí se realizan.

Todos los valores que se muestran en la tabla están descritos en un rango de 1 a 4 siendo, 1=bajo, 2= medio bajo, 3 = medio, 4= medio alto, 5= alto para las respectivas calificaciones dentro de la misma.

4.02. Matriz de impactos de Objetivos.

Tabla 20

Matriz de objetivos del sistema

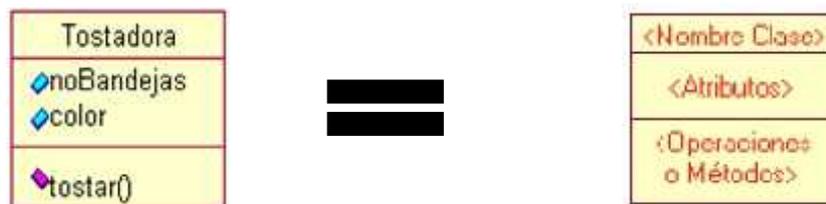
Objetivos	Factibilidad de lograrse	Impacto de Genero	Impacto Ambiental	Relevancia	Sostenibilidad
Se asigna recursos para mejora la tecnología	Los recursos que se asignan se los hace tomando en cuenta sus beneficios a corto mediano y largo plazo	Se trabaja en igualdad de género según políticas de la empresa y normas del ministerio de relaciones laborales.	Los equipos informáticos que se emplearan estarán acorde a las necesidades de los empleados	Se mejora el entorno de trabajo de los empleados	La empresa mantendrá el software debido a sus múltiples beneficios.
Se capacita al personal sobre las tic	Se cuenta con personal calificado para la realización de talleres.	Se trabaja en igualdad de género según políticas de la empresa y normas del ministerio de relaciones laborales.	El personal de la empresa aprenderá a controlar de mejor manera los recursos informáticos	El empleado es más eficiente ante cualquier dificultad relacionada con la informática	La capacitación del empleado se la realizara acorde a sus necesidades
Se fomenta una cultura tecnológica	Los grandes avances tecnológicos se miran día a día en los medios de información	Se trabaja en igualdad de género según políticas de la empresa y normas del ministerio de relaciones laborales.	El empleado a de mejorar sus conocimientos en avances técnicos y tecnológicos	El empleado es capaz de conocer y elegir los respectivos elementos informáticos que lo ayuden a mejorar su nivel de vida	El personal es más conocedor de las nuevas tecnologías
Se asigna personal adecuado al departamento de tecnología	Existe personal que está dispuesto a trabajar por un bien común entre empresa y empleado	Se trabaja en igualdad de género según políticas de la empresa y normas del ministerio de relacione	Se minimizara los riegos de contaminación por mala manipulación y eliminación de equipos informáticos	El responsable ayudara a mejorar los procesos de trabajo a través de la implementación de recursos necesarios.	La empresa a través del talento humano calificar al personal

4.03. Estándares para el diseño de clases.

Para poder comprender con exactitud que es el diseño de una clase es fundamental conocer que es una clase detallando así lo siguiente.

Que es una clase.

Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. Una clase implementa una o más interfaces. Gráficamente se representa como un rectángulo que incluye su nombre, sus atributos y sus operaciones.



En las figuras 5 y 6 podemos notar la estructura con que la realizaremos el modelado de la clase del sistema que por lo general es el más eficiente y efectivo al momento del diseño del mismo.

Una clase está compuesta por tres partes principales como son:

Nombre de la clase

Atributos de la clase

Operación de la clase

4.04. Diagrama de clases

Ver anexo 003

Análisis.

En el diagrama de clases se muestra las relaciones que existen entre las clases así como sus atributos y sus respectivas operaciones que actuarán para generar la base de datos que utilizaremos en la elaboración del software.

4.05. Modelo Lógico – Físico

Ver Anexo 004

Análisis

En esta figura se muestra principalmente los requisitos funcionales, lo que el sistema debe brindar en términos de servicios a sus usuarios. Aquí se aplican los principios de abstracción, encapsulación y herencia. Esta descomposición no sólo se hace para potenciar el análisis funcional, sino también sirve para identificar mecanismos y elementos de diseño comunes a diversas partes del sistema.

4.06. Diagrama de Componentes

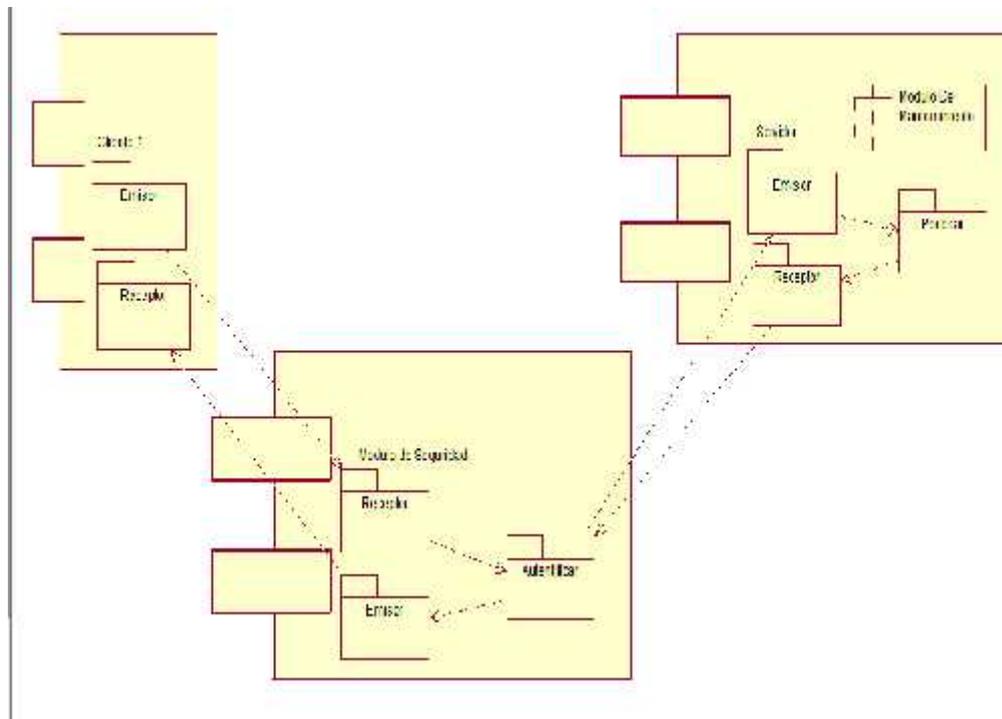


Figura. 11 Diagrama de componentes del sistema desarrollado

Análisis.

En esta figura se ve el diagrama de componentes que representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de Componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

4.07. Diagrama de Estrategias.

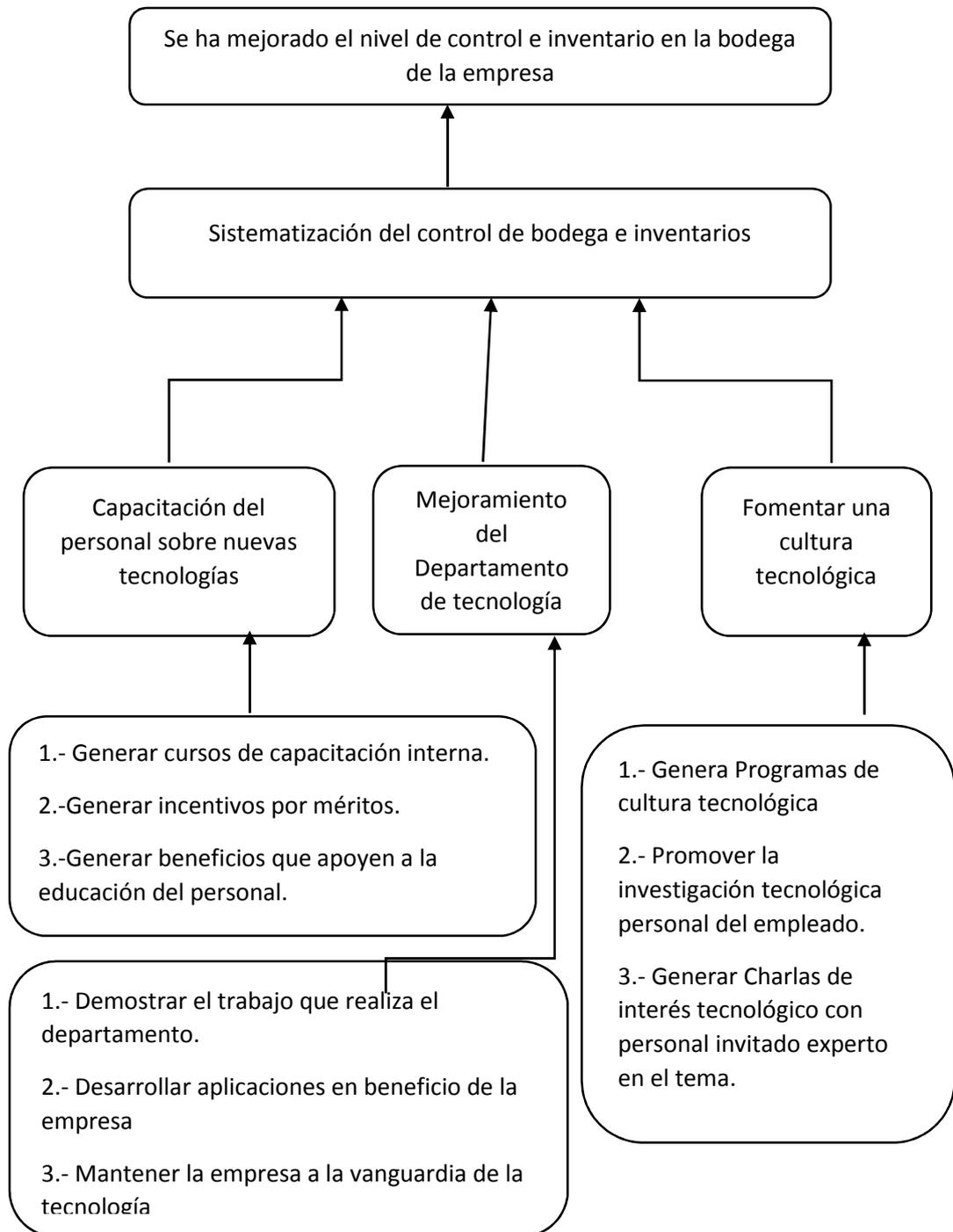


Figura. 12 Estrategias tomadas.

4.08. Matriz de Marco lógico.

En esta sección se describe el marco lógico del proyecto que se está desarrollando

Ver anexo 005

4.09. Vistas Arquitectónicas.

4.01.01. Vista lógica.

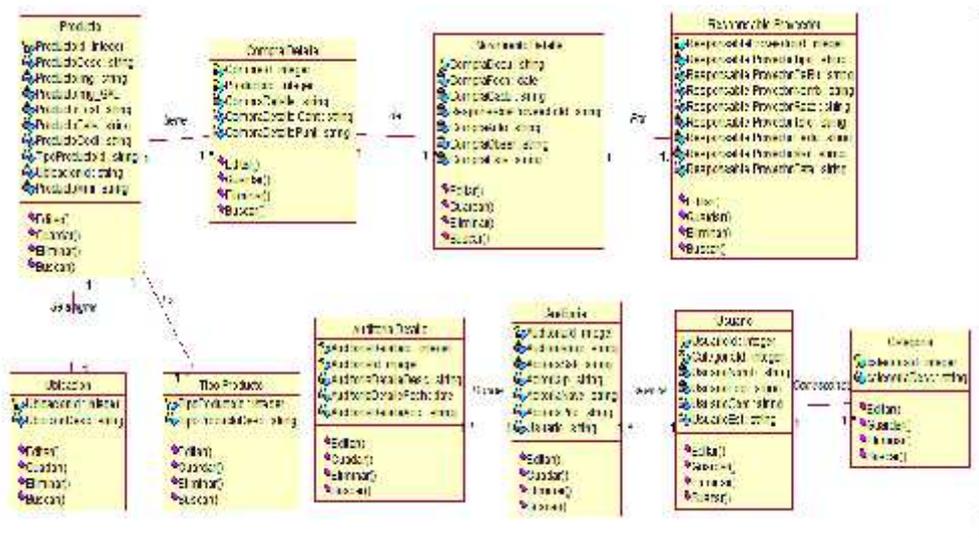


Figura. 13 Vista lógica del sistema creado

4.01.02. Vista Física.

Ver anexo 001

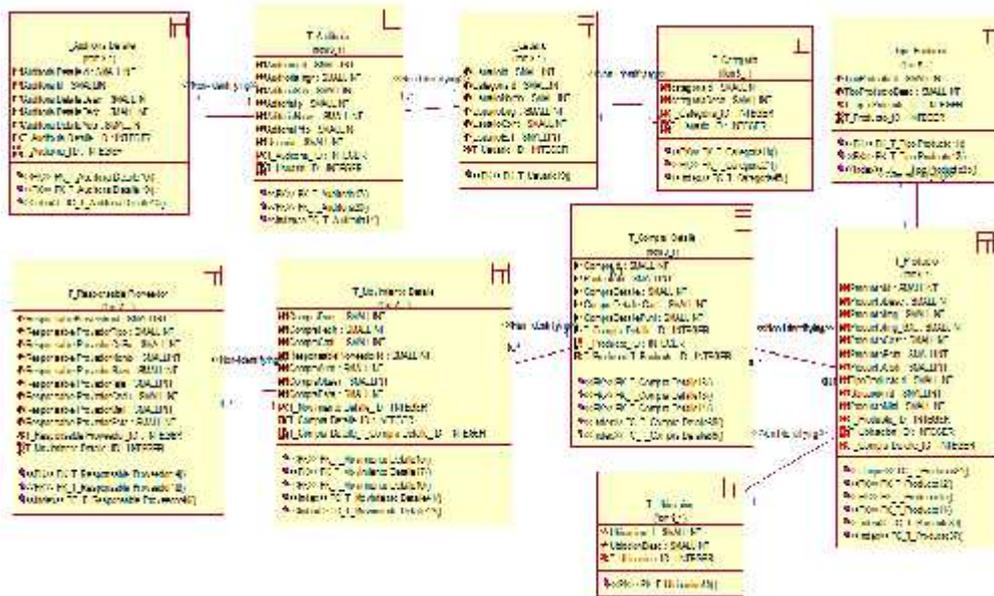


Figura. 14 Se demuestra la vista física del sistema creado.

4.01.03. Vista de Desarrollo.

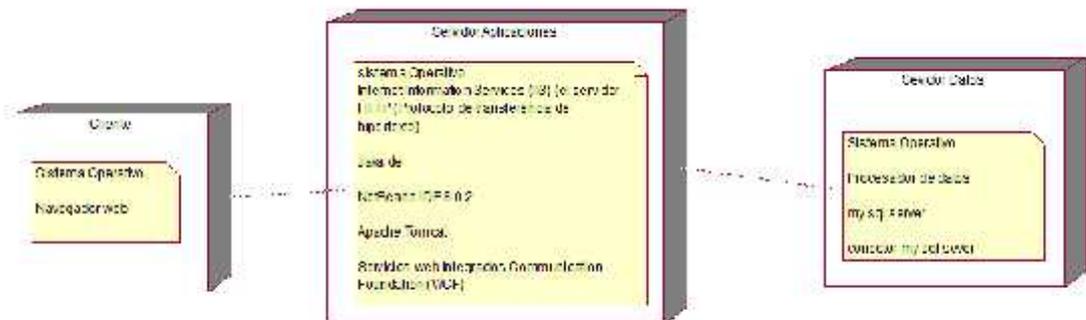


Figura. 15 Se demuestra los procesos básicos que se ejecutarán

Análisis

En esta figura se muestra como se centra en la organización real de los módulos de software en el ambiente de desarrollo.

El software se empaqueta en partes pequeñas que pueden ser bibliotecas o subsistemas que son desarrollados por uno o un grupo de desarrolladores.

Los subsistemas se organizan en una jerarquía de capas, cada una brinda una interfaz estrecha y bien definida hacia las capas superiores.

4.01.04. Vista de procesos.

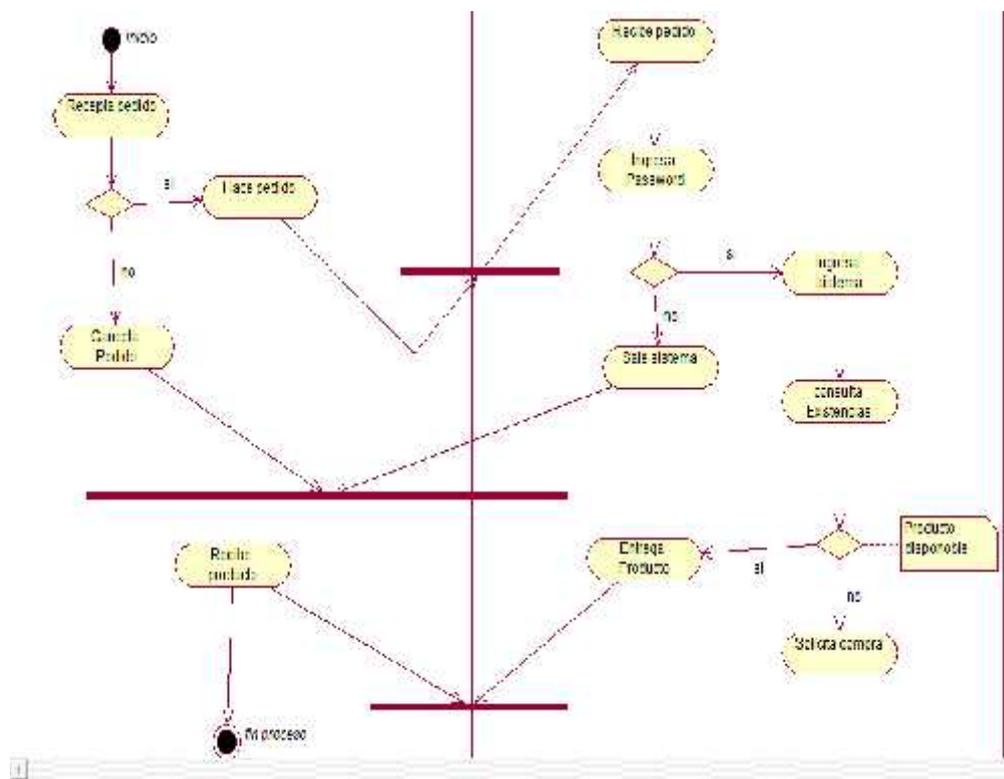


Figura. 16 Se describen los procesos del sistema

Análisis

Se tratan los aspectos de concurrencia y distribución, integridad del sistema, y tolerancia a fallos.

Puede ser descrita como un conjunto de redes lógicas de procesos que son ejecutados de forma independiente, y distribuidos a lo largo de varios recursos de hardware conectados mediante un bus o a una red de datos.

Capítulo V: Propuesta

5.01. Especificación de estándares de programación

El software control de inventario esta creado en una plataforma de desarrollo denominado NetBeans IDE 8.0 y sus componentes necesarios como son el jdk para Windows Seven de 64 bits, así también la implantación y configuración de Apache Tomcat que Es un servidor de internet (especializado en aplicaciones Java). Que hace a una computadora sea capaz de almacenar los sitios Web que creas (en simple HTML o en Java) y enviarlos a las personas que se conecten a él como una página que puedan ver en sus Navegadores (Internet Exporer, FireFox, etc.).

My sql server

Estándares de programación java

Organización de ficheros

Las clases en Java se agrupan en paquetes. Estos paquetes se deben organizar de manera jerárquica.

Dentro del paquete principal las clases se organizarán en subpaquetes en función del área, organismo o sección del Ayuntamiento al que pertenezca el código desarrollado. Por ejemplo, si estamos desarrollando un servicio web de inscripción a un curso de programación Java del IMFE las clases de dicho servicio se incluirían en el paquete "eu.malaga.imfe.webservices.cursojava" o similar.

Un fichero consta de secciones que deben estar separadas por líneas en blanco y comentarios opcionales que identifiquen cada sección.

Deben evitarse los ficheros de gran tamaño que contengan más de 1000 líneas. En

ocasiones, este tamaño excesivo provoca que la clase no encapsule un comportamiento claramente definido, albergando una gran cantidad de métodos que realizan tareas funcional o conceptualmente heterogéneas.

Fichero fuente Java (.java)

Cada fichero fuente Java debe contener una única clase o interfaz pública. El nombre del fichero tiene que coincidir con el nombre de la clase. Cuando existan varias clases privadas asociadas funcionalmente a una clase pública, podrán colocarse en el mismo fichero fuente que la clase pública. La clase pública debe estar situada en primer lugar dentro del fichero fuente.

En todo fichero fuente Java distinguimos las siguientes secciones:

Comentarios de inicio.

Sentencia de paquete.

Sentencias de importación.

Declaraciones de clases e interfaces.

Comentarios de inicio

Todo fichero fuente debe comenzar con un comentario que incluya el nombre de la clase, información sobre la versión del código, la fecha y el copyright. El copyright indica la propiedad legal del código, el ámbito de distribución, el uso para el que fue desarrollado y su modificación.

Dentro de estos comentarios iniciales podrían incluirse adicionalmente comentarios sobre los cambios efectuados sobre dicho fichero (mejora, incidencia, error, etc.).

Estos comentarios son opcionales si los ficheros están bajo un sistema de control de versiones bien documentado, en caso contrario se recomienda su uso. Estos comentarios constituyen el historial de cambios del fichero. Este historial es único para cada fichero y permitirá conocer rápidamente el estado y la evolución que ha tenido el fichero desde su origen.

A continuación se muestra un comentario de inicio para la clase "JceSecurity.java".

```
/*  
  
* @(#)JceSecurity.java 1.50 04/04/14  
  
*  
  
* Copyright 2004 Sun Microsystems, Inc. All rights reserved.  
  
* SUN PROPRIETARY/CONFIDENTIAL. Use is subject to license terms.  
  
*/  
  
/**  
  
* This class instantiates implementations of JCE engine classes from  
  
* providers registered with the java.security.Security object.  
  
*  
  
* @author Jan Luehe  
  
* @author Sharon Liu  
  
* @version 1.50, 04/14/04  
  
* @since 1.4  
  
*/
```

Sentencias de paquete

La primera línea no comentada de un fichero fuente debe ser la sentencia de paquete, que indica el paquete al que pertenece(n) la(s) clase(s) incluida(s) en el

fichero fuente. Por ejemplo,

```
package javax.crypto;
```

Sentencias de importación

Tras la declaración del paquete se incluirán las sentencias de importación de los paquetes necesarios. Esta importación de paquetes obligatorios seguirá el siguiente orden:

Paquetes del JDK de java.

Paquetes de utilidades no pertenecientes al JDK de Java, de frameworks de desarrollo o de proyectos opensource tales como apache, hibernate, springframework, etc.

Paquetes desarrollados para el Ayuntamiento de Málaga.

Paquetes de la aplicación.

Se recomienda minimizar en la medida de lo posible el uso de importaciones del tipo "package.*", pues dificultan la comprensión de las dependencias existentes entre las clases utilizadas por la aplicación. En caso contrario, se recomienda utilizar comentarios de línea tras la importación.

```
import java.io.*; // BufferedReader, PrintWriter, FileInputStream, File
```

```
import java.util.ArrayList;
```

```
import org.apache.log4j.Logger;
```

```
import org.apache.lucene.analysis.Analyzer;
```

```
import es.provincia.organismo.corporativas.atlas.vo.AgendaVO;
```

import es.provincia.organismo.atlas.vo.AnuncioVO;

import es.provincia.organismo.atlas.vo.OrganigramaVO;

Declaraciones de clases e interfaces	Descripción
La siguiente tabla describe los elementos que componen la declaración de una clase o interfaz, así como el orden en el que deben estar situados.	
Elementos de declaración de una clase / interfaz	
Comentario de documentación de la clase/interfaz /** ... */	Permite describir la clase/interfaz desarrollada. Necesario para generar la documentación de la api mediante javadoc.
Sentencia class / interface	
Comentario de implementación de la clase/interfaz, si es necesario /* ... */	Este comentario incluye cualquier información que no pueda incluirse en el comentario de documentación de la clase/interfaz.
Variables de clase (estáticas)	En primer lugar las variables de clase públicas (public), después las protegidas (protected), posteriormente las de nivel de paquete (sin modificador), y por último las privadas (private).
Variables de instancia	Primero las públicas (public), después las protegidas (protected), luego las de nivel de paquete (sin modificador), y finalmente las privadas (private).
Constructores	
Métodos	Deben agruparse por funcionalidad en lugar de agruparse por ámbito o accesibilidad. Por ejemplo, un método privado puede estar situado entre dos métodos públicos. El objetivo es desarrollar código fácil de leer y comprender.

Cuando una expresión ocupe más de una línea, esta se podrá romper o dividir en función de los siguientes criterios,

Tras una coma.

Antes de un operador.

Se recomienda las rupturas de nivel superior a las de nivel inferior.

Alinear la nueva línea con el inicio de la expresión al mismo nivel que la línea anterior.

Si las reglas anteriores generan código poco comprensible, entonces estableceremos tabulaciones de 8 espacios.

Ejemplos:

```
unMetodo(expresionLarga1, expresionLarga 2, expresionLarga 3,  
          expresionLarga 4, expresionLarga 5);
```

```
if ((condicion1 && condicion2)  
    || (condicion3 && condicion4)  
    ||!(condicion5 && condicion6)) {  
    unMetodo();  
}
```

Comentarios

Distinguimos dos tipos de comentarios: los comentarios de implementación y los de documentación.

Comentarios de implementación

Estos comentarios se utilizan para describir el código ("el cómo"), y en ellos se incluye información relacionada con la implementación, tales como descripción de la función de variables locales, fases lógicas de ejecución de un método, captura de excepciones, etc.

Distinguimos tres tipos de comentarios de implementación:

Comentarios de bloque:

Permiten la descripción de ficheros, clases, bloques, estructuras de datos y algoritmos.

/*

* Esto es un comentario

* de bloque

*/

Comentarios de línea:

Son comentarios cortos localizados en una sola línea y tabulados al mismo nivel que el código que describen. Si ocupa más de una línea se utilizará un comentario de bloque. Deben estar precedidos por una línea en blanco.

/* Esto es un comentario de línea */

// Esto es otro comentario de línea

Comentario a final de línea

Comentario situado al final de una sentencia de código y en la misma línea.

```
int contador = 4 + 10; // Inicialización del contador
```

```
contador++; /* Incrementamos el contador */
```

Comentarios de documentación

Los comentarios de documentación, también denominados "comentarios javadoc", se utilizan para describir la especificación del código, desde un punto de vista independiente de la implementación, de forma que pueda ser consultada por

desarrolladores que probablemente no tengan acceso al código fuente.

El apartado 2 de este documento describe el uso de comentarios de documentación.

Declaraciones

Una declaración por línea

Se recomienda el uso de una declaración por línea, promoviendo así el uso de comentarios. Ejemplo,

```
int idUnidad; // Identificador de la unidad organizativa  
String[] funciones; // Funciones de la unidad
```

Inicialización

Toda variable local tendrá que ser inicializada en el momento de su declaración, salvo que su valor inicial dependa de algún valor que tenga que ser calculado previamente.

```
int idUnidad = 1;  
String[] funciones = { "Administración", "Intervención", "Gestión" };
```

Localización

Las declaraciones deben situarse al principio de cada bloque principal en el que se utilicen, y nunca en el momento de su uso.

```
public void unMetodo() {  
    int contador = 0; // inicio del método
```

```
...  
}
```

La única excepción a esta regla son los índices de los bucles "for", ya que, en Java, pueden incluirse dentro de la propia sentencia "for".

```
for (int i=0; contador<10; i++) {  
...  
}
```

Se debe evitar el uso de declaraciones que oculten a otras declaraciones de ámbito superior.

```
int contador = 0; // Inicio del método  
public void unMetodo() {  
  
if (condicion) {  
    int contador = 2; // ¡¡ EVITAR !!  
...  
}  
...  
}
```

Declaración de clases / interfaces

Durante el desarrollo de clases / interfaces se deben seguir las siguientes reglas de formateo:

No incluir ningún espacio entre el nombre del método y el paréntesis inicial del listado de parámetros.

El carácter inicio de bloque ("{"") debe aparecer al final de la línea que contiene la sentencia de declaración.

El carácter fin de bloque ("}") se sitúa en una nueva línea tabulada al mismo nivel que su correspondiente sentencia de inicio de bloque, excepto cuando la sentencia sea nula, en tal caso se situará detrás de "{".

Los métodos se separarán entre sí mediante una línea en blanco.

```
public classe ClaseEjemplo extends Object {  
  
    int variable1;  
  
    int variable2;  
  
    public ClaseEjemplo() {  
  
        variable1 = 0;  
  
        variable2 = 1;  
  
    }  
  
    ...  
  
}
```

Sentencias

Cada línea debe contener como máximo una sentencia. Ejemplo,

```
int contador++;  
  
int variable--;
```

Las sentencias pertenecientes a un bloque de código estarán tabuladas un nivel más a la derecha con respecto a la sentencia que las contiene.

El carácter inicio de bloque "{" debe situarse al final de la línea que inicia el bloque.

El carácter final de bloque "}" debe situarse en una nueva línea tras la última línea del bloque y alineada con respecto al primer carácter de dicho bloque.

Todas las sentencias de un bloque deben encerrarse entre llaves "{ ... }", aunque el bloque conste de una única sentencia. Esta práctica permite añadir código sin cometer errores accidentalmente al olvidar añadir las llaves. Ejemplo,

```
if (condicion) {  
    variable++;  
}
```

La sentencia "try/catch" siempre debe tener el formato siguiente,

```
try {  
    sentencias;  
} catch (ClaseException e) {  
    sentencias;  
}
```

En el bloque "catch" siempre se imprimirá una traza de error indicando el tipo de excepción generada y posteriormente se elevará dicha excepción al código invocante, salvo que la lógica de ejecución de la aplicación no lo requiera.

Siempre se utilizará el bloque "finally" para liberar recursos y para imprimir trazas de monitorización de fin de ejecución.

```
try {
```

```
sentencias;  
  
} catch (ClaseException e) {  
  
    sentencias;  
  
} finally {  
  
    sentencias;  
  
}
```

Espacios en blanco

Las líneas y espacios en blanco mejoran la legibilidad del código permitiendo identificar las secciones de código relacionadas lógicamente.

Se utilizarán espacios en blanco en los siguientes casos:

Entre una palabra clave y un paréntesis. Esto permite que se distingan las llamadas a métodos de las palabras clave. Por ejemplo:

```
while (true) {  
  
    ...  
  
}
```

Tras cada coma en un listado de argumentos. Por ejemplo:

```
objeto.unMetodo(a, b, c);
```

Para separar un operador binario de sus operandos, excepto en el caso del operador ("."). Nunca se utilizarán espacios entre los operadores unarios (p.e., "++" o "--") y sus operandos. Por ejemplo:

```
a += b + c;
```

```
a = (a + b) / (c + d);
```

```
contador++;
```

Para separar las expresiones incluidas en la sentencia "for". Por ejemplo:

for (expresion1; expresion2; expresion3)

Al realizar el moldeo o "casting" de clases. Ejemplo:

Unidad unidad = (Unidad) objeto;

Nomenclatura de identificadores

Las convenciones de nombres de identificadores permiten que los programas sean más fáciles de leer y por tanto más comprensibles. También proporcionan información sobre la función que desempeña el identificador dentro del código, es decir, si es una constante, una variable, una clase o un paquete, entre otros.

Paquetes

Se escribirán siempre en letras minúsculas para evitar que entren en conflicto con los nombres de clases o interfaces. El prefijo del paquete siempre corresponderá a un nombre de dominio de primer nivel, tal como: es, eu, org, com, net, etc.

El resto de componentes del paquete se nombrarán de acuerdo a las normas internas de organización de la empresa: departamento, proyecto, máquina, sección, organismo, área, etc.

Generalmente se suele utilizar el nombre de dominio de Internet en orden inverso.

Cuando dicho nombre contenga un carácter "-", este se sustituirá por el carácter "_".

Ejemplos:

es.provincia.organismo1.festivaldecine

es.provincia.organismo2.vivienda

es.provincia.organismo3.juventud

es.provincia.organismo3.formacion

es.provincia.organismo3.gestionturistica

java.util.ArrayList

java.util.Date

java.util.Properties

javax.servlet.http.HttpServletRequest

javax.servlet.http.HttpServletResponse

Clases e interfaces

Los nombres de clases deben ser sustantivos y deben tener la primera letra en mayúsculas. Si el nombre es compuesto, cada palabra componente deberá comenzar con mayúsculas.

Los nombres serán simples y descriptivos. Debe evitarse el uso de acrónimos o abreviaturas, salvo en aquellos casos en los que dicha abreviatura sea más utilizada que la palabra que representa (URL, HTTP, etc.).

Las interfaces se nombrarán siguiendo los mismos criterios que los indicados para las clases. Como norma general toda interfaz se nombrará con el prefijo "I" para diferenciarla de la clase que la implementa (que tendrá el mismo nombre sin el prefijo "I").

class Ciudadano

class OrganigramaDAO

class AgendaService

class IAgendaService

Métodos

Los métodos deben ser verbos escritos en minúsculas. Cuando el método esté

compuesto por varias palabras cada una de ellas tendrá la primera letra en mayúsculas.

```
public void insertaUnidad(Unidad unidad);
```

```
public void eliminaAgenda(Agenda agenda);
```

```
public void actualizaTramite(Tramite tramite)
```

Variables

Las variables se escribirán siempre en minúsculas. Las variables compuestas tendrán la primera letra de cada palabra componente en mayúsculas.

Las variables nunca podrán comenzar con el carácter "_" o "\$". Los nombres de variables deben ser cortos y sus significados tienen que expresar con suficiente claridad la función que desempeñan en el código. Debe evitarse el uso de nombres de variables con un sólo carácter, excepto para variables temporales.

```
Unidad unidad;
```

```
Agenda agenda;
```

```
Tramite tramite;
```

Constantes

Todos los nombres de constantes tendrán que escribirse en mayúsculas. Cuando los

nombres de constantes sean compuestos las palabras se separarán entre sí mediante el carácter de subrayado "_".

```
int LONGITUD_MAXIMA;
```

```
int LONGITUD_MINIMA;
```

(fundations, 2010)

Apache Tomcat

Introducción al Servidor de Aplicaciones Tomcat de Apache

Tomcat es un contenedor de Servlets con un entorno JSP. Un contenedor de Servlets es un shell de ejecución que maneja e invoca servlets por cuenta del usuario.

Podemos dividir los contenedores de Servlets en:

Contenedores de Servlets Stand-alone (Independientes)

Estos son una parte integral del servidor web. Este es el caso cuando usando un servidor web basado en Java, por ejemplo, el contenedor de servlets es parte de JavaWebServer (actualmente sustituido por **iPlanet**). Este el modo por defecto usado por Tomcat.

Sin embargo, la mayoría de los servidores, no están basados en Java, los que nos lleva los dos siguientes tipos de contenedores:

Contenedores de Servlets dentro-de-Proceso

El contenedor Servlet es una combinación de un plugin para el servidor web y una implementación de contenedor Java. El plugin del servidor web abre una JVM (Máquina Virtual Java) dentro del espacio de direcciones del servidor web y permite que el contenedor Java se ejecute en él. Si una cierta petición debería ejecutar un

servlet, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando JNI). Un contenedor de este tipo es adecuado para servidores multi-thread de un sólo proceso y proporciona un buen rendimiento pero está limitado en escalabilidad

Contenedores de Servlets fuera-de-proceso

El contenedor Servlet es una combinación de un plugin para el servidor web y una implementación de contenedor Java que se ejecuta en una JVM fuera del servidor web. El plugin del servidor web y el JVM del contenedor Java se comunican usando algún mecanismo IPC (normalmente sockets TCP/IP). Si una cierta petición debería ejecutar un servlet, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando IPCs). El tiempo de respuesta en este tipo de contenedores no es tan bueno como el anterior, pero obtiene mejores rendimientos en otras cosas (escalabilidad, estabilidad, etc.).

Tomcat puede utilizarse como un contenedor solitario (principalmente para desarrollo y depuración) o como plugin para un servidor web existente (actualmente se soporan los servidores Apache, IIS y Netscape). Esto significa que siempre que despluguemos Tomcat tendremos que decidir cómo usarlo, y, si seleccionamos las opciones 2 o 3, también necesitaremos instalar un adaptador de servidor web

Arrancar y Parar Tomcat

Arrancamos y paramos Tomcat usando los scripts que hay en el directorio **bin**:

Para arrancar Tomcat ejecutamos:

- Sobre UNIX:
- bin/startup.sh
- Sobre Win32:
- bin\startup

Para parar Tomcat ejecutamos:

- Sobre UNIX:
- bin/shutdown.sh
- Sobre Win32:

- bin\shutdown

La Estructura de Directorios de Tomcat

Asumiendo que hemos descomprimido la distribución binaria de Tomcat deberíamos tener la siguiente estructura de directorios:

Nombre de Directorio	Descripción
bin	Contiene los scripts de arrancar/parar
conf	Contiene varios ficheros de configuración incluyendo server.xml (el fichero de configuración principal de Tomcat) y web.xml que configura los valores por defecto para las distintas aplicaciones desplegadas en Tomcat.
doc	Contiene varia documentación sobre Tomcat (Este manual, en inglés).
lib	Contiene varios ficheros jar que son utilizados por Tomcat. Sobre UNIX, cualquier fichero de este directorio se añade al classpath de Tomcat.
logs	Aquí es donde Tomcat sitúa los ficheros de diario.
src	Los ficheros fuentes del API Servlet. ¡No te excites, todavía! Estoa son sólo los interfaces vacíos y las clases abstractas que debería implementar cualquier contenedor de servlets.
webapps	Contiene aplicaciones Web de Ejemplo.

Adicionalmente podemos, o Tomcat creará, los siguientes directorios:

Ficheros de Configuración de Tomcat

La configuración de Tomcat se basa en dos ficheros:

1. server.xml - El fichero de configuración global de Tomcat.
2. web.xml - Configura los distintos contextos en Tomcat.

Esta sección trata la forma de utilizar estos ficheros. No vamos a cubrir las interioridades de web.xml, esto se cubre en profundidad en la especificación del API Servlet. En su lugar cubriremos el contenido de server.xml y discutiremos el uso de web.xml en el contexto de Tomcat.

server.xml

server.xml es el fichero de configuración principal de Tomcat. Sirve para dos objetivos:

1. Proporcionar configuración inicial para los componentes de Tomcat.

2. Especifica la estructura de Tomcat, lo que significa, permitir que Tomcat arranque y se construya a sí mismo ejemplarizando los componentes especificados en server.xml.

Los elementos más importantes de server.xml se describen en la siguiente tabla:

Se puede encontrar información adicional dentro del fichero server.xml.

Configurar Tomcat para Cooperar con Apache Web Server

Hasta ahora no hemos explicado Tomcat como un plugin, en su lugar lo hemos considerado como un contenedor independiente y hemos explicado cómo usarlo. Sin embargo, hay algunos problemas:

1. Tomcat no es tan rápido como Apache cuando sirve páginas estáticas.
2. Tomcat no es tan configurable como Apache.
3. Tomcat no es tan robusto como Apache.
4. Hay mucho sites que llavan mucho tiempo de investigación sobre ciertos servidores web, por ejemplo, sites que usan scripts CGI o módulos perl o php...

No podemos asumir que todos ellos quieran abandonar dichas tecnologías.

Por todas estas razones es recomendable que las sites del mundo real usen un servidor web, como Apache, para servir el contenido estático de la site, y usen Tomcat como un plugin para Servlets/JSP.

No vamos a cubrir las diferentes configuraciones en profundidad, en su lugar:

1. Cubriremos el comportamiento fundamental de un servidor web.
2. Explicaremos la configuración que necesitamos.
3. Demostraremos esto sobre Apache.

Operación del Servidor Web

En resumidas cuentas un servidor web está esperando peticiones de un cliente HTTP.

Cuando estas peticiones llegan el servidor hace lo que sea necesario para servir las peticiones proporcionando el contenido necesario. Añadirle un contenedor de servlets podría cambiar de alguna forma este comportamiento. Ahora el servidor Web también necesita realizar lo siguiente:

- Cargar la librería del adaptador del contenedor de servlets e inicializarlo (antes de servir peticiones).
- Cuando llega una petición, necesita chequear para ver si una cierta petición pertenece a un servlet, si es así necesita permitir que el adaptador tome el control y lo maneje.

Por otro lado el adaptador necesita saber qué peticiones va a servir, usualmente basándose en algún patrón de la URL requerida, y dónde dirigir estas peticiones.

Las cosas son incluso más complejas cuando el usuario quiere seleccionar una configuración que use hosts virtuales, o cuando quieren que múltiples desarrolladores trabajen en el mismo servidor web pero en distintos contenedores de Servlets. Cubriremos estos dos casos en las secciones avanzadas.

Configuraciones necesarias

La configuración más obvia en la que uno puede pensar es en la identidad de las URLs servlet que están bajo la responsabilidad del contenedor de servlets. Esto está claro, alguien debe conocer qué peticiones transmitir al contenedor de servlets...

Todavía hay algunos ítems de configuración adicionales que deberíamos proporcionar a la combinación web-server/servlet-container:

- Necesitamos proporcionar la configuración sobre los procesos Tomcat disponibles y sobre los puertos/host TCP/IP sobre los que éstos están escuchando.

- Necesitamos decirle al servidor web la localización de la librería adaptador (para que pueda cargarla en la arrancada).
- Necesitamos seleccionar la información interna del adaptador sobre cuando log guardar, etc.

Toda esta información debe aparecer en el fichero de configuración del servidor web, o en un fichero de configuración privado usado por el adaptador. La siguiente sección explicará cómo se puede implementar esta configuración en Apache.

(Esteban, 1998)

MySQL

El sistema de base de datos operacional MySQL es hoy en día uno de los más importantes en lo que hace al diseño y programación de base de datos de tipo relacional. Cuenta con millones de aplicaciones y aparece en el mundo informático como una de las más utilizadas por usuarios del medio. El programa MySQL se usa como servidor a través del cual pueden conectarse múltiples usuarios y utilizarlo al mismo tiempo.

La historia del MySQL (cuya sigla en inglés se traslada a My Structured Query Language o Lenguaje de Consulta Estructurado) se remite a principios de la década de 1980. Programadores de IBM lo desarrollaron para contar con un código de programación que permitiera generar múltiples y extendidas bases de datos para empresas y organizaciones de diferente tipo. Desde esta época numerosas versiones han surgido y muchas de ellas fueron de gran importancia. Hoy en día MySQL es desarrollado por la empresa Sun Microsystems.

Una de las características más interesantes de MySQL es que permite recurrir a bases de datos multiusuario a través de la web y en diferentes lenguajes de programación

que se adaptan a diferentes necesidades y requerimientos. Por otro lado, MySQL es conocida por desarrollar alta velocidad en la búsqueda de datos e información, a diferencia de sistemas anteriores. Las plataformas que utiliza son de variado tipo y entre ellas podemos mencionar LAMP, MAMP, SAMP, BAMP y WAMP (aplicables a Mac, Windows, Linux, BSD, Open Solaris, Perl y Python entre otras). (melvinbladimir5, 2010)

Acciones Básicas

Para crear y utilizar una base de datos encontramos las siguientes opciones en el menú de MySQL:

- Crear una base de datos.
- Crear una tabla.
- Introducir datos en la tabla.
- Recuperar datos desde la tabla de varias maneras.
- Emplear múltiples tablas.

Crear una Base de Datos.

Se la puede realizar mediante la sentencia SHOW

Ejemplo:

```
Mysql> SHOW DATABASES;
```

Para crear una base de datos propia que evite que los datos colocados en ella sean fácilmente borrados por otra persona que disponga de acceso es necesario pedir

permiso de administrador para realizar esta operación crear una base de datos como por ejemplo el nombre “manager” a través de un comando:

SISTEMATIZACIÓN DE LOS PROCESOS DE CONTROL DE INVENTARIO A TRAVÉS DE UN SOFTWARE ORIENTADO A LA WEB PARA LA EMPRESA GLOBALCONNECTION DE LA CIUDAD DE QUITO

```
Mysql> GRANT ALL ON managerie.* TO 'su_nombre_mysql'@' su  
_host_cliente';
```

Extraer datos desde una base existente.

Se la realiza a través de la sentencia SELEC la que se usa para extraer información desde una tabla la sintaxis genera de esta es:

SELEC **seleccionar_esto**

FROM **desde_tabla**

WHERE **condiciones;**

En este comando “seleccionar _esto “Es lo que se quiere ver.

“ Desde_tabla” Indica la tabla donde están los datos a recuperar.

La cláusula WHERE es opcional. Si está presente, representa las condiciones que deben cumplir los registros para considerarse resultados válidos.

5.02. Diseño de interfaces de usuario.

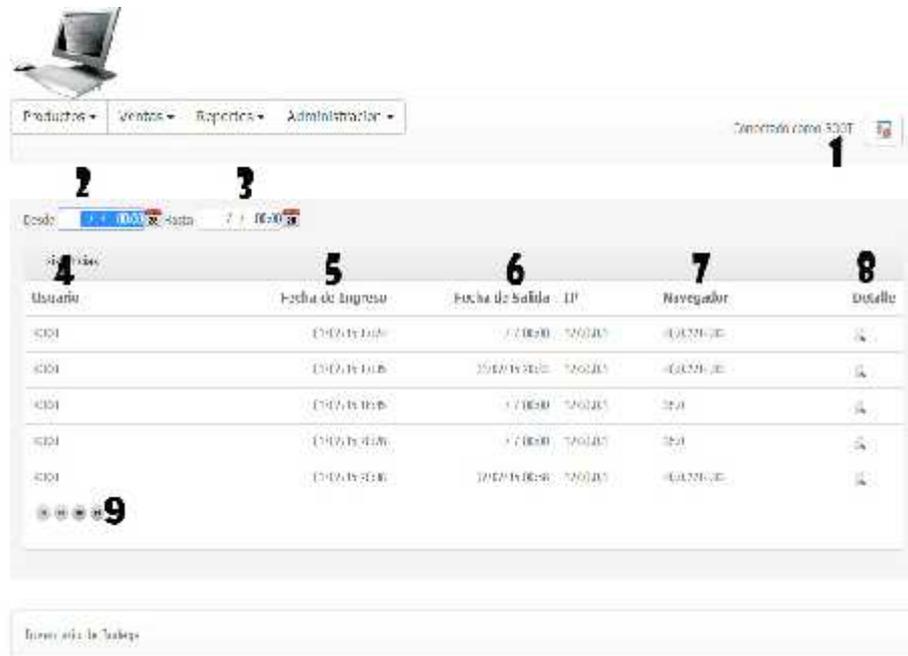


Figura. 16 Se muestra la ventana de auditoria

En la figura. 16 cada número corresponde a una acción que el software ejecuta cómo se menciona a continuación.

- 1.- Tipo de usuario en sistema
- 2.- Fecha desde cuando se realiza la auditoria
- 3.- Fecha de finalización de la auditoria
- 4.- Tipo de usuario al que se realiza la auditoria
- 5.- Fecha que ingreso al sistema
- 6.- Fecha de salida de y dirección ip auditable
- 7.- Detalle de modificaciones en sistema
- 8.- Detalle para imprimir
- 9.- Botón para pasar las paginas

Figura 17 Ventana de productos

En la figura. 17 cada número corresponde a una acción que el software ejecuta cómo se menciona a continuación.

- 1.- Campo para ingreso nombre producto
- 2.- Campo para ingreso de imagen producto
- 3.- Campo para número de existencias mínima en bodega
- 4.- Campo de selección tipo producto
- 5.- Campo de selección de ubicación de producto dentro de las bodegas
- 6.- Botón guardar (actualizar la base)
- 7.- Botón cancelar proceso

5.03. Especificación de pruebas de unidad.

En programación, una prueba unitaria o de unidad es una forma de comprobar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada

uno de los módulos funcione correctamente por separado para así poder asegurar el correcto funcionamiento del sistema o subsistema en cuestión.

Metodología RUP

El Rational Unified Process o Proceso Unificado de Racional. Es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta y de mayor calidad para satisfacer las necesidades de los usuarios que tienen un cumplimiento al final dentro de un límite de tiempo y presupuesto previsible. Es una metodología de desarrollo iterativo que es enfocada hacia “diagramas de los casos de uso, y manejo de los riesgos y el manejo de la arquitectura” como tal.

El RUP mejora la productividad del equipo ya que permite que cada miembro del grupo sin importar su responsabilidad específica pueda acceder a la misma base de datos incluyendo sus conocimientos. Esto hace que todos compartan el mismo lenguaje, la misma visión y el mismo proceso acerca de cómo desarrollar un software.

Tabla 21

Módulo de prueba Administración.

Identificador de prueba	IDPU001
Método a Probar	Prueba dinámica(RUP)
Objetivo de la Prueba	Conocer si el software realiza las respectivas validaciones de los datos que se ingresan al sistema
Datos de Entrada	Información del personal responsable de bodega
Resultados Esperados	1.- Validar los diferentes datos que ingresan al sistema así como la cedula del empleado entre otros.
Comentarios	El código fuente debe estar estructurado de tal forma que se capas de validar los datos que se ingresan al sistema.

Tabla 22*Módulo de prueba Administración Login*

Identificador de prueba	IDPU002
Método a Probar	Prueba dinámica(RUP)
Objetivo de la Prueba	Conocer si el sistema realiza las diferentes validaciones de usuarios que ingresan al sistema.
Datos de Entrada	
Password	
Resultados Esperados	
1.- Si el usuario ingresa un password diferente al asignado el usuario no ingresa al sistema. 2.- Si el usuario ingresa un password correcto ingresa al sistema y continúa con el proceso.	
Comentarios	
Antes de realizar dicha verificación se creara al usuario y se le asignara una un password al cual el software validara.	

Tabla 23*Módulo de prueba Productos*

Identificador de prueba	IDPU003
Método a Probar	Prueba dinámica(RUP)
Objetivo de la Prueba	Conocer si almacena y valida la información del producto que se ingresa al sistema.
Datos de Entrada	
Información detallada de los productos que ingresan a bodega.	
Resultados Esperados	
Validar y guardar información de los productos que se ingresan al sistema. Tales como nombre tipo material, etc.	
Comentarios	
El empleado responsable de ingresar dicha información debe tomar en cuenta todos los datos que el sistema requiere para seguir con el debido proceso.	

Tabla 24*Módulo de prueba Ventas (Egresos)*

Identificador de prueba	IDPU004
Método a Probar	Prueba dinámica(RUP)
Objetivo de la Prueba	Conocer si el sistema genera y registra las salidas de productos y los actualiza en base.
Datos de Entrada	
Ingreso de pedidos	
Resultados Esperados	
1.- Confirma niveles de productos en stock. 2.-Contabiliza y registra los egresos de bodega.	
Comentarios	
El empleado responsable de despachar el producto debe tomar en cuenta todos los datos que el sistema requiere para seguir con el debido proceso.	

Tabla 25*Módulo de prueba Reportes*

Identificador de prueba	IDPU005
Método a Probar	Prueba dinámica(RUP)
Objetivo de la Prueba	Conocer si se generan los reportes de los productos para posterior su impresión.
Datos de Entrada	
Acepta petición de consulta de los productos de la bodega por parte de los usuarios del sistema	
Resultados Esperados	
Generar informe de los productos, niveles de stock precios ,etc.	
Comentarios	
El empleado responsable podrá imprimir los reportes de las respectivas consultas.	

Análisis

En estas tablas se determina el funcionamiento del software acorde a la metodología, Como es la programación orientada a objetos, en capas o metodología RUP que es la que facilita el trabajo ya que se lo puede realizar por secciones o módulos.

5.04. Especificación de pruebas de aceptación.

Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Estas pruebas no se realizan durante el desarrollo, pues sería impresentable al cliente; sino que se realizan sobre el producto terminado e integrado o pudiera ser una versión del producto o una iteración funcionad pactada previamente con el cliente.

La experiencia muestra que aún después del más cuidadoso proceso de pruebas por parte del desarrollador, quedan una serie de errores que sólo aparecen cuando el cliente comienza a usarlo

Una prueba de aceptación puede ir desde un informal caso de prueba hasta la ejecución sistemática de una serie de pruebas bien planificadas. De hecho, las pruebas de aceptación pueden tener lugar a lo largo de semanas o meses, descubriendo así errores latentes o escondidos que pueden ir degradando el funcionamiento del sistema. Estas pruebas son muy importantes, ya que definen el paso nuevas fases del proyecto cómo el despliegue y mantenimiento.

Tabla 26*Módulo de prueba Administración*

Identificador de la prueba	IDPA001
Caso de uso	Administrador sistema
Tipo de usuario	Administrador
Objetivo de la Prueba	Demostrar la creación de los usuarios y la asignación de sus respectivos password
Secuencia de Eventos	
<ol style="list-style-type: none"> 1.- Ingresar al sistema. 2.- Ingresa información de nuevo usuario. 3.- Asignación de una ID y un password. 4.- Guardar información en la base 	
Resultados esperados	
<ol style="list-style-type: none"> 1.- Crear usuario 2.- Validar usuario y password 3.- Actualizar base 	
Comentarios	
El administrador del sistema será el encargado de verificar la funcionabilidad del sistema	
Estado	Aceptado

Tabla 27*Módulo de prueba Ingreso de productos*

Identificador de la prueba	IDPA002
Caso de uso	Ingreso Productos
Tipo de usuario	Responsable bodega
Objetivo de la Prueba	Demostrar como ingresan los datos de cada producto en detalles en el sistema
Secuencia de Eventos	
<ol style="list-style-type: none"> 1.- ingresa tipo producto. 2.- ingresa producto nombre. 3.- ingresa material producto. 4.- ingresa ubicación producto 5.- ingresa proveedor (si existe) 6.- guarda información. 	
Resultados esperados	
<ol style="list-style-type: none"> 1.- Muestra en pantalla los productos con sus respectivos detalles 2.- Guarda actualiza la base. 	
Comentarios	
Se debe llenar todos los campos que el sistema solicita para continuar con el proceso de ingreso de los productos.	
Estado	Aceptado

Tabla 28*Módulo de prueba Ventas (Egresos)*

Identificador de la prueba	IDPA003
Caso de uso	Egreso Productos
Tipo de usuario	Responsable de Bodega
Objetivo de la Prueba	Demostrar en sistema los niveles de stock e inventario general
Secuencia de Eventos	
1.- Generar un egreso de producto 2.- verificar información en pantalla del productos seleccionado.	
Resultados esperados	
Información detallada del producto seleccionado con sus respectivas actualizaciones en base.	
Comentarios	
El responsable de bodega completara toda la información que el sistema solicite para continuar con el resto del proceso.	
Estado	Aceptado

Tabla 29*Módulo de prueba Auditoria*

Identificador de la prueba	IDPA004
Caso de uso	Auditoria
Tipo de usuario	Administrador
Objetivo de la Prueba	Conocer los movimiento, detalles de peticiones que el responsable de bodega realiza
Secuencia de Eventos	
1.- Ingresar al sistema en modo administrador 2.- Ingresar a auditoria 3.- Seleccionar fecha de peticiones 4.- Generar un informe (imprimir)	
Resultados esperados	
Mostrar en pantalla las diferentes peticiones que le responsable de bodega realiza dentro del sistema durante un periodo comprendido de entre desde y hasta.	
Comentarios	
El responsable de auditoria deberá ingresar al sistema en modo administrador para poder verificar dicha información.	
Estado	Aceptado

5.05. Especificación de pruebas de carga.

La carga de trabajo se refiere a la capacidad máxima que tiene un servidor web (hardware y software), para atender a un conjunto de usuarios de manera simultánea. Por ello, las actividades de esta etapa tienen relación con comprobar, de manera anticipada, el funcionamiento que tendrá el servidor del Sitio Web cuando esté en plena operación.

Las pruebas en este caso consisten en simular una carga de trabajo similar y superior a la que tendrá cuando el sitio esté funcionando, con el fin de detectar si el software instalado (programas y aplicaciones) cumple con los requerimientos de muchos usuarios simultáneos y también si el hardware (servidor y el equipamiento computacional de redes y enlace que lo conecta a Internet) es capaz de soportar la cantidad de visitas esperadas.

Es importante considerar que si el servidor está en las dependencias de un tercero que entrega el servicio de alojamiento del Sitio Web (hosting), se le debe solicitar a dicho proveedor un informe en que dé a conocer las características de carga de la solución de hardware y software sobre la cual funciona el Sitio Web de la institución.

Hay diversos software en el mercado que está orientados a este tipo de simulaciones, todos los cuales ofrecen características similares. Entre los datos más relevantes que es posible obtener se cuenta:

Tiempo de acceso de los usuarios a los datos

Volumen de datos y ancho de banda utilizado

Archivos solicitados y tiempos usados en transferencia de datos

Tiempo de espera de los usuarios tras hacer un clic

Tiempo de respuesta a clicks de usuarios

Niveles de error existentes tras clicks de usuarios

Como se puede apreciar del listado anterior, los reportes que se obtienen a través de esta vía se refieren a tiempos de acceso que tienen los usuarios que acceden al Sitio Web y la degradación que ocurre en los servicios cuando aumenta el volumen de visitantes concurrentes.

Tabla 30

Pruebas de servidor

Identificador de la Prueba	IDPC001
Tipo de prueba	Simulación de ingreso de información por lotes cantidades máximas por día
Objetivo de la Prueba	Conocer si la base de datos soporta la subida masiva de información
Descripción	Ingresar la información de los productos por lotes o por segmentos
Resultados Esperados	La base de datos soporte dichas pruebas de sobrecarga de información
Comentarios	La base de datos debe soportar la subida excesiva de carga de información por que a futuro crecerá en forma exponencial.

Tabla 31

Interconexión de usuarios

Identificador de la Prueba	IDPC001
Tipo de prueba	Simulación de usuarios conectados al sistema simultáneamente
Objetivo de la Prueba	Conocer si el sistema soporta el mayor número de usuarios conectado y ejecutando peticiones al servidor
Descripción	Conectar el mayor número de usuarios en una tiempo establecido
Resultados Esperados	El sistema soporte el mayor número de usuarios realizando peticiones en tiempo real
Comentarios	El sistema debe ser capaz de soportar el mayor número de usuarios y petición de los mismos en tiempo real debido al crecimiento que la empresa generara en un tiempo próximo.

5.06. Configuración del Ambiente mínima /ideal.

Los requerimientos básicos son el mínimo valor con el cual el sistema puede operar al 100% de su capacidad no así quiere decir que es un estándar de requerimientos.

A continuación especificamos los requerimientos básicos de:

Servidor

hardware	software
<ul style="list-style-type: none"> • Monitor • Teclado • Mouse • CPU • Dual core 3.0 GHZ • Memoria ram de 4 GB • Disco duro de 500 GB • Dvd writer • Tarjeta de red 	<ul style="list-style-type: none"> • Sistema operativo • MYSQL Server (wamp) • Apache tomcat • Navicat premium

Cientes

hardware	Software
<ul style="list-style-type: none">• Monitor• Teclado• Mause• CPU• Intel celeron GHZ• Memoria ram de 512 MB• Disco duro de 320 GB• Tarjeta de red	<ul style="list-style-type: none">• Sistema operativo• Navegador• Paquete office

Configuraciones de los equipos

Servidor

Cliente

Ver Anexo 007 manuales

Capítulo VI Aspectos Administrativos

Aspectos Administrativos

6.01. Recursos

En esta sección describiremos todos los recursos que utilizamos para la elaboración del software de control de inventario

Tabla.32

Recursos utilizados.

Recursos Humanos	Recursos Tecnológico	Recursos Vario
<ul style="list-style-type: none">• Desarrollador• Tutor• Lector	<ul style="list-style-type: none">• Computador• Impresora	<ul style="list-style-type: none">• Resma de papel• Luz• Teléfono• Internet• Movilización

6.02. Presupuesto

Elaborar un presupuesto permite a las empresas, los gobiernos, las organizaciones privadas o las familias establecer prioridades y evaluar la consecución de sus objetivos.

Tabla 33*Recursos Utilizados*

Recursos Humanos	Monto 1	Recursos tecnológico	Monto 2	Recursos Vario	Monto 3
Desarrollador según tabla sectorial 2015	381,65 USD	Computador 4 horas diarias 6/m	0,75 por hora	Resma de papel	4,50
				Luz	8,00
				Teléfono	30,00
				Impresiones	75,00
				Anillado	12,00
				Internet	45,00
				Empastado	50,00
Totales	2.289,90		360		224,50

Total suma

montos 1+2+3

2874,40 USD**6.03. Cronograma**

En esta sección se detalla paso a paso las diferentes actividades que se realizaron durante la ejecución del software.

Ver anexo 006

Capítulo VII: Conclusiones y Recomendaciones

7.01. Conclusiones

- Se ha desarrollado un software de control e inventario orientado a la web con total satisfacción luego de varios días de dedicación y esfuerzo.
- El software que se creó es una herramienta que facilitará al empleado responsable de bodega a controlar los productos que ingresan en las instalaciones.

7.02. Recomendaciones

- Para generar un software es necesario conocer la plataforma de desarrollo sobre la cual se construirá el mismo, así también sus herramientas necesarias.
- Para la elaboración de un software es necesario establecer comunicación con las personas que trabajaran directamente e indirectamente en la organización.

Bibliografía

Lenguaje Unificado de Modelado UML. 16 de 08 de 2012. 06 de 03 de 2014

Mónica Urigüen, Ph.D. Un Futuro Planificado Estratégicamente: Aplicación de las

Herramientas de Calidad. 10 de 08 de 2007. 12 de 04 de 2014

El Proceso Unificado de Desarrollo de Software. 25 de 10 de 2008. 20 de 03 de 2014

Gutiérrez, Javier. Especificación e implementación de casos de prueba. 26 de 02 de 2009. 17 de 03 de 2014

Que es un lenguaje de programación. 15 de 06 de 2010. 14 de 03 de 2014

<http://www.javiergarzas.com/2012/03/herramientas-para-pruebas-software.html>

http://es.wikipedia.org/wiki/Diagrama_de_componentes

Desarrollo, S. N. (05 de Noviembre de 2009). *planificacion.gob.ec*. Obtenido de http://www.planificacion.gob.ec/wp-content/uploads/downloads/2012/07/Plan_Nacional_para_el_Buen_Vivir.pdf

Desarrollo, S. N. (05 de Noviembre de 2012). *planificacion.gob.ec*. Obtenido de

http://www.planificacion.gob.ec/wp-content/uploads/downloads/2013/01/matriz_productiva_WEBtodo.pdf

http://www.planificacion.gob.ec/wp-content/uploads/downloads/2013/01/matriz_productiva_WEBtodo.pdf

Esteban, E. a. (8 de junio de 1998). *programacion net*. Obtenido de

http://programacion.net/articulo/tomcat_-_introduccion_134

fundations, j. (22 de julio de 2010). *javafundation.com*. Obtenido de

[http://javafundations.blogspot.com/2010/07/java-estandares-de-](http://javafundations.blogspot.com/2010/07/java-estandares-de-programacion.html#1_estandares)

[programacion.html#1_estandares](http://javafundations.blogspot.com/2010/07/java-estandares-de-programacion.html#1_estandares)

melvinbladimir5. (11 de febrero de 2010). *my sql*. Obtenido de

Anexos

Anexo 001

Matriz de requerimientos. Funcionales y no funcionales

Matriz de requerimientos

Identificador	Descripción	Fuente	Prioridad	Tipo	Estado	Usuarios Involucrados
---------------	-------------	--------	-----------	------	--------	-----------------------

Requerimientos funcionales

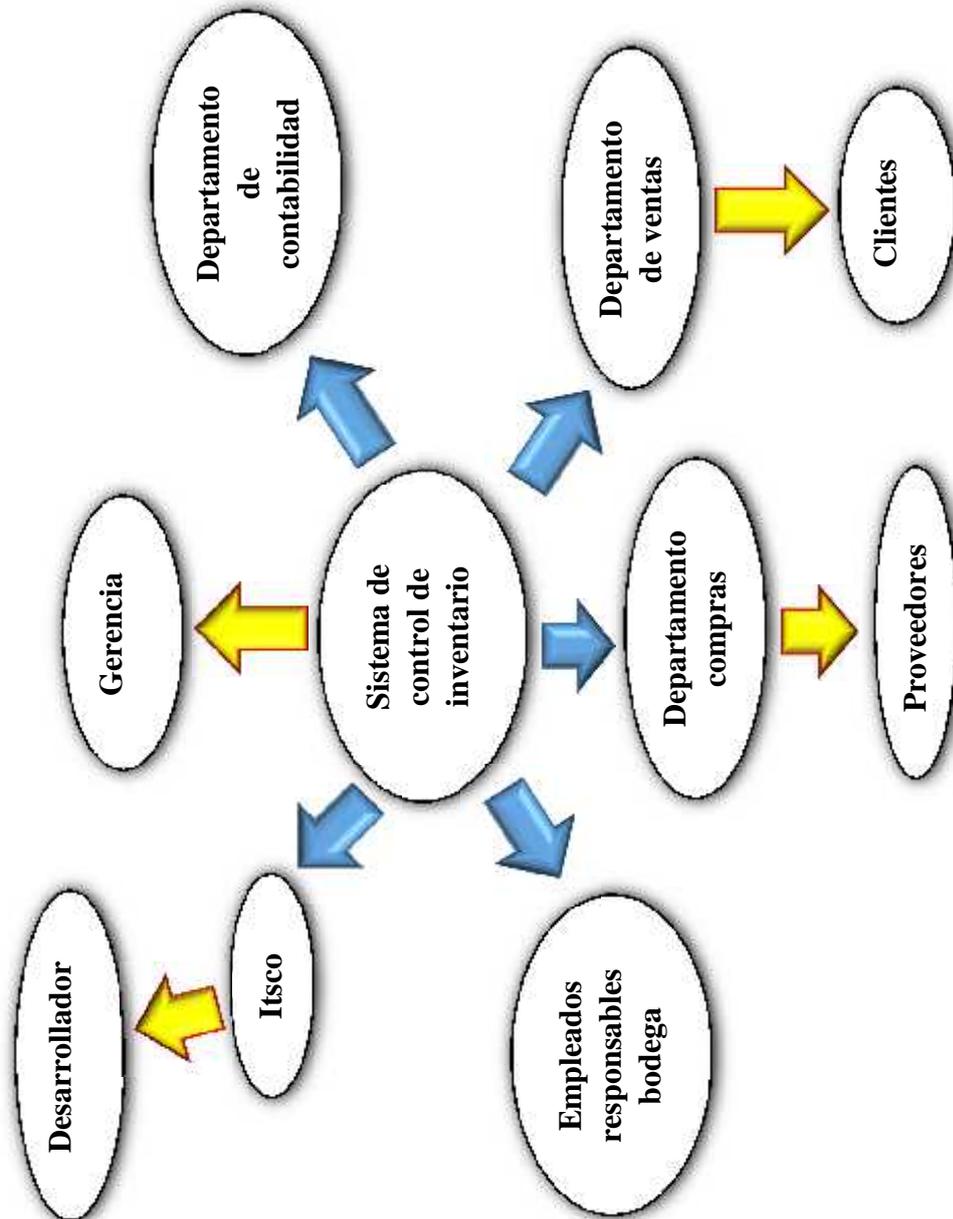
REF001	Registro de usuarios	Gerencia, Administrador sistema.	Alta	Sistema	Valido	Gerente, Departamento de contabilidad, empleados responsables de bodega
REF002	Ingresos de usuarios al sistema	Gerencia, Administrador sistema.	Alta	Sistema	Valido	Gerente
REF003	Registro de proveedores	Gerencia, Departamento compras.	Alta	Sistema	Valido	Gerente, departamento de contabilidad.
REF004	Ingreso de mercadería	Gerencia, Departamento compras.	Alta	Sistema	Valido	Gerente, personal responsable de bodega,
REF005	Reporte Stock de inventario	Gerencia, Departamento ventas.	Alta	Sistema	Valido	Gerente, personal responsable de bodega, departamento de compras, ventas

Requerimientos no funcionales.

RNF001	Reporte de productos por categoría	Gerencia; Empleados bodega, departamento ventas.	Baja	Sistema	Revisión	Gerente, Empleados de bodega
RNF002	Reporte de productos con mayor demanda	Gerencia; departamento de ventas.	Baja	Sistema	Revisión	Gerente, Empleados de bodega
REF003	Al Software se accede a través de todo tipo de navegador	Gerencia;	Baja	Sistema	Revisión	Gerente, Empleados de bodega

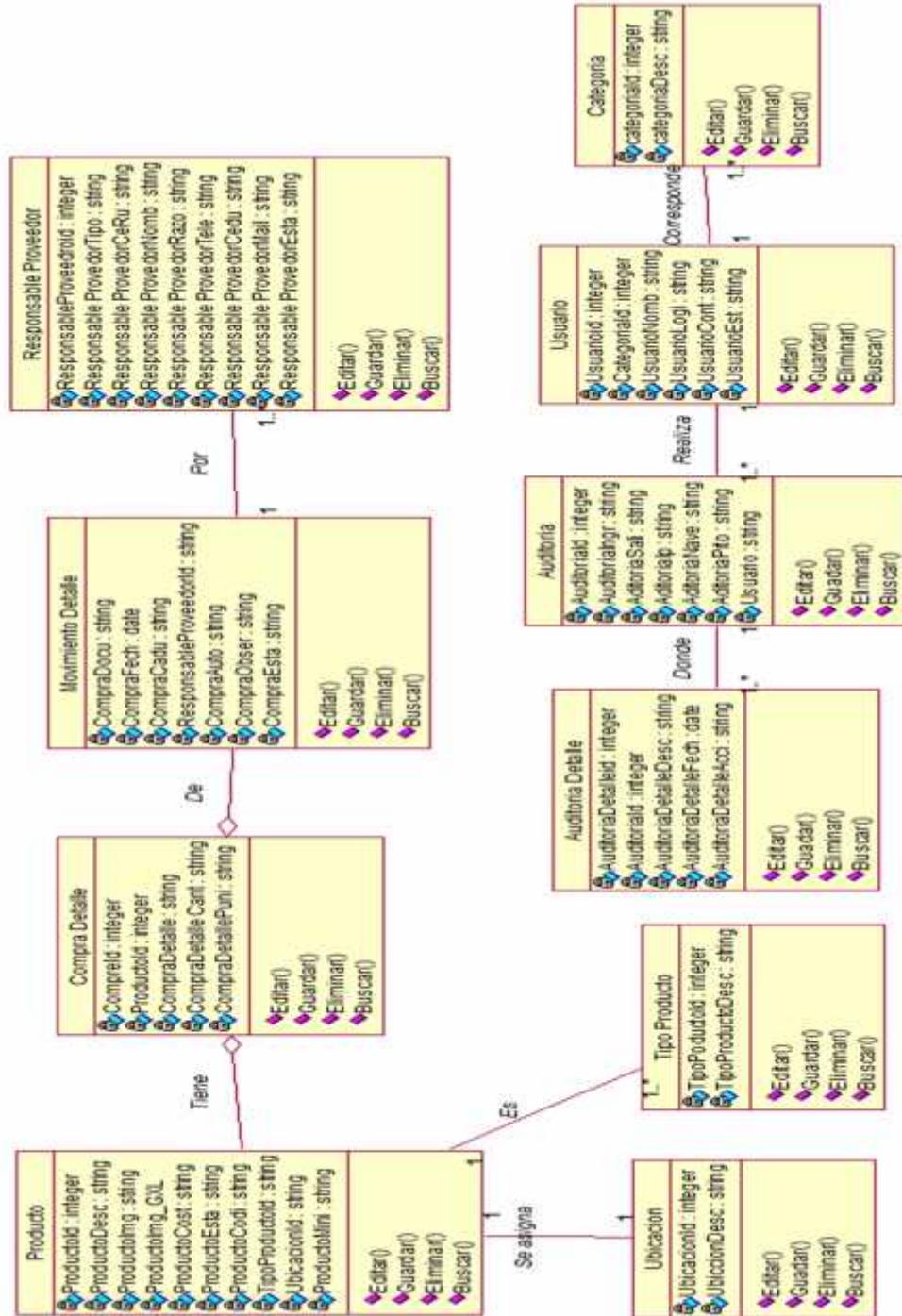
Anexo 002

Mapeo de Involucrados



Anexo 003

Diagrama de clases



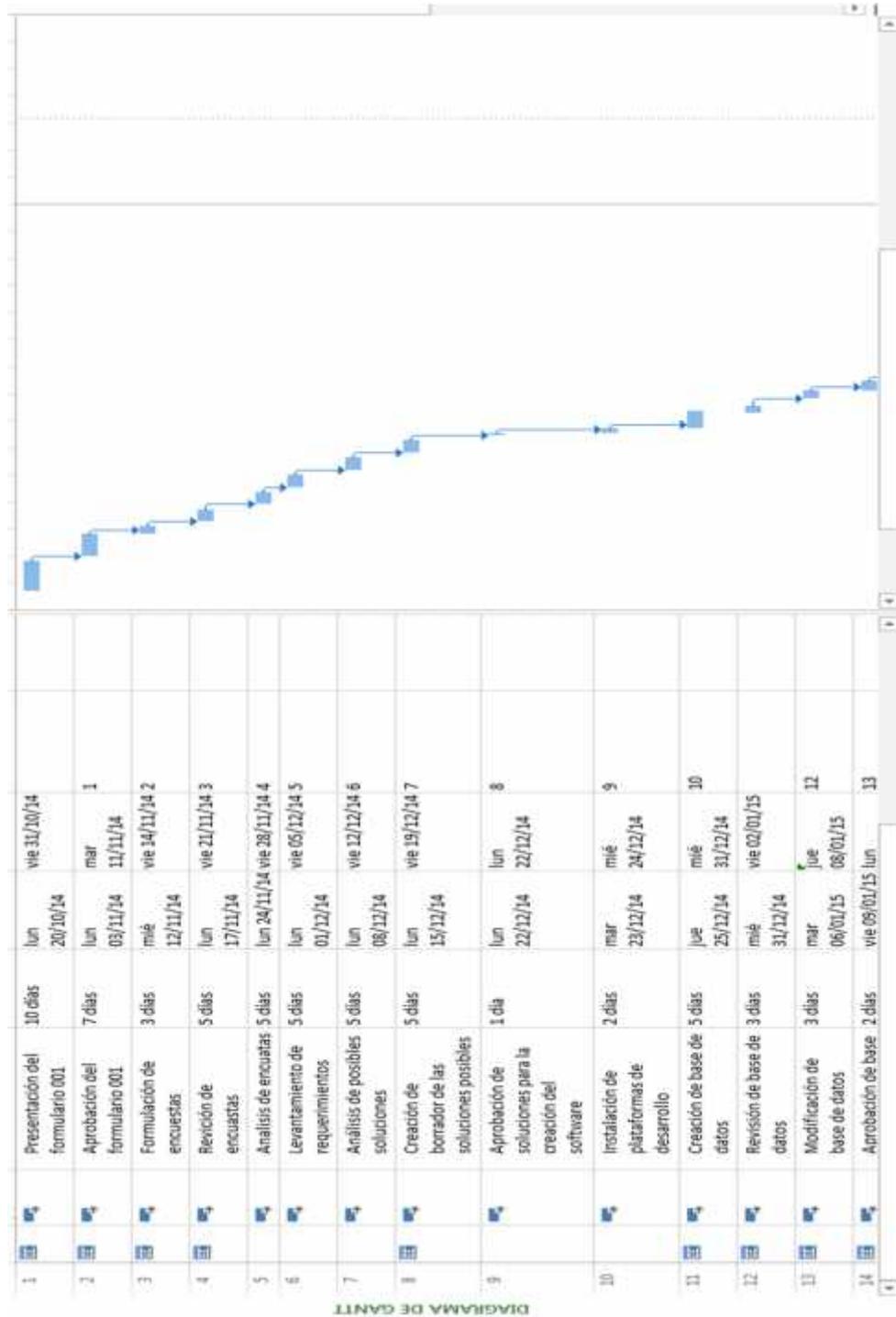
Anexo 005

Matriz de Marco lógico.

	Indicadores	Medios de Verificación	Supuestos
Finalidad Contribuir de forma adecuada en el desarrollo y mejoramiento de los procesos de administración de bodega así como un desarrollo técnico y tecnológico.	El proceso de administración de bodegas estará evaluado según las normas INEN (instituto ecuatoriano de normalización)	Se lo realizara a través del manual técnico que mantiene el INEN para el control e inventario de bodegas	Los procesos de admiración de bodega no mejoran con la implantación de una software Para el desarrollo de una empresa no es necesaria la implantación de software.
Propósito A través del software se reduciría notablemente la pérdida de las partes y piezas de los equipos informáticos	Se generara informes detallados por periodos que comprenden un antes y después de para cotejar dicha información y valorar los resultados.	Se lo realizara a través de cuadros estadísticos. Se tomara en cuenta las sugerencias del personal.	El software es de vital importancia dentro de una empresa por que ayuda a no perder información.
Componentes El software facilita la generación y control de bodega. El personal de la empresa que está relacionado más de cerca con el software conocerá que la importancia de la tecnología en la actualidad.	El personal responsable de manipular dicho software será encuestado y generara su utilidad y valor del software	Se lo realizara a través de cuadros estadísticos. Se tomara en cuenta las sugerencias del personal.	La tecnología es vital en una empresa para su desarrollo económico y social
Actividades Capacitación del personal. Valoración del software por personal responsable para su ejecución.	El personal de la empresa será evaluado según su desempeño laboral.	A través de la participación del personal en los diferentes departamentos.	Un personal que conoce de tecnología tiene mayores oportunidades de ascender e una empresa.

Anexo 006

Cronograma



Manual de Instalación

Anexo 007

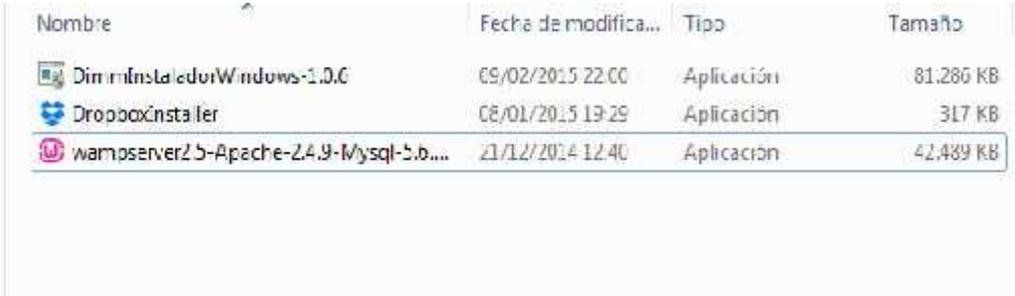
Manual de instalación y configuración básica del sistema

En esta sección se detallan las configuraciones básicas y la instalación del respectivo software para el funcionamiento óptimo de la a aplicación realizada.

Instalando wampserver (My SQL SERVER)

Pasos para su instalación y ejecución

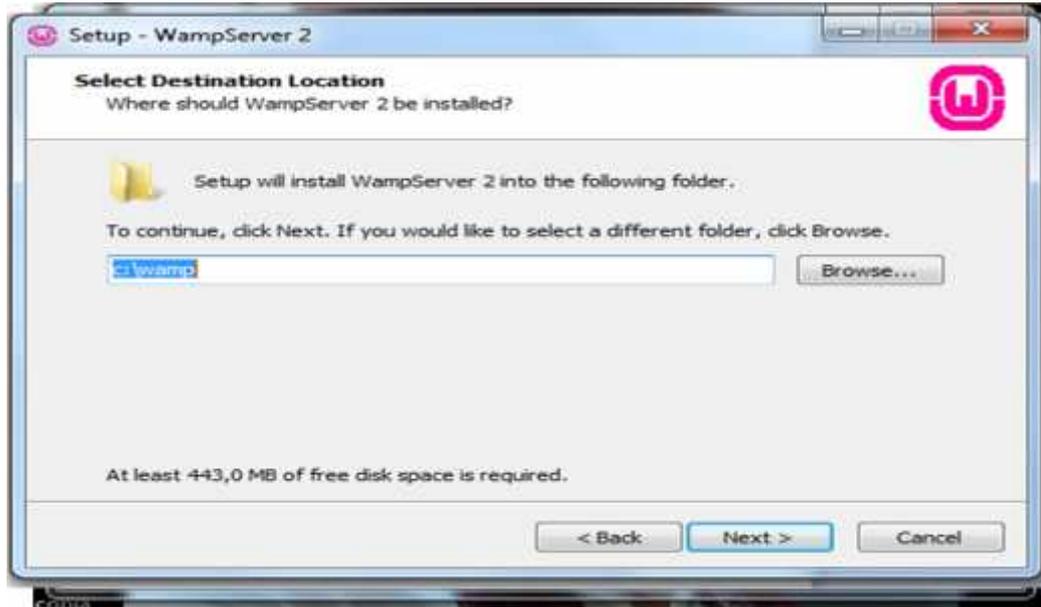
1.- descargara la aplicación wampserver



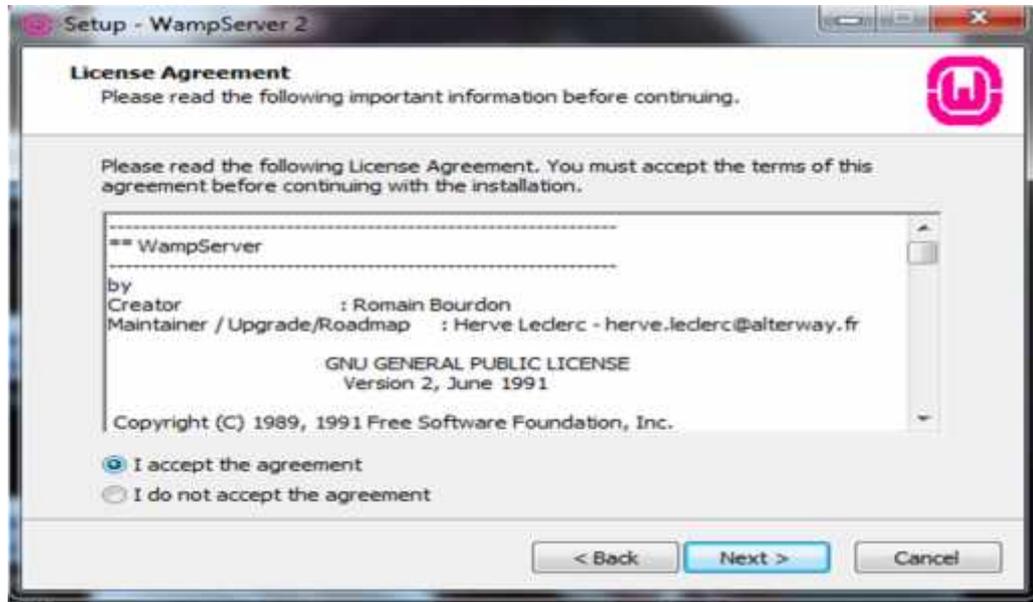
Nombre	Fecha de modifica...	Tipo	Tamaño
DiminInstaladorWindows-1.0.0	09/02/2015 22:00	Aplicación	81.286 KB
DropboxInstalier	08/01/2015 19:29	Aplicación	317 KB
wampserver2.5-Apache-2.4.9-MySQL-5.6...	21/12/2014 12:40	Aplicación	42.489 KB

2.- Sobre el icono de wampserver dar doble clic y se despliega la siguiente ventana.

Clic en next

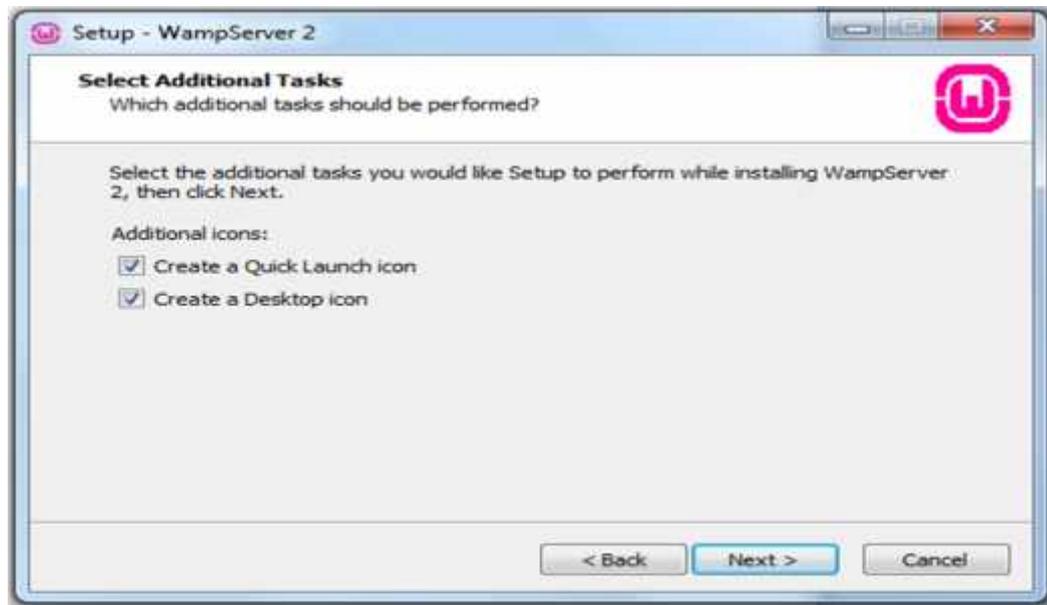


3.- Aparecerá la siguiente ventana, aceptamos los términos y condiciones para posterior dar clic en next

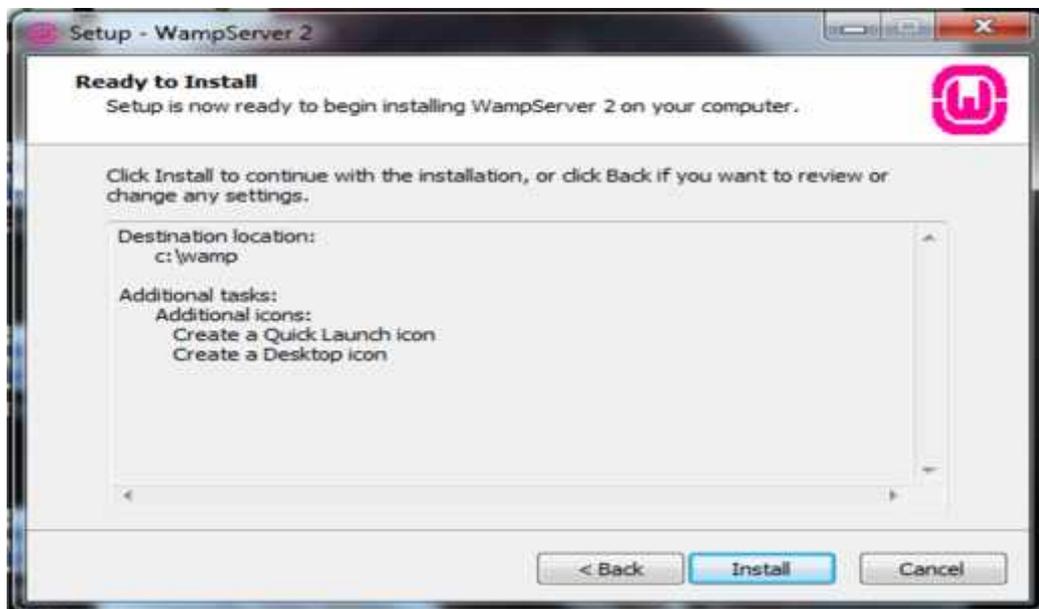


4.-. Luego muestra la siguiente ventana, aquí seleccionamos la carpeta donde instalaremos dicha aplicación y sus datos generales. Luego damos clic en next

5.- Creamos los accesos directos seleccionando en la siguiente pestaña, posterior a esto damos clic en next.



6.- Llegamos a la ventana de instalación y empezamos a utilizar, clic en install



Instalación de apache tomcat

Descargar el programa apache tomcat tomado en cuenta a versión de sistema operativo en el acula se instalara dicha aplicación.

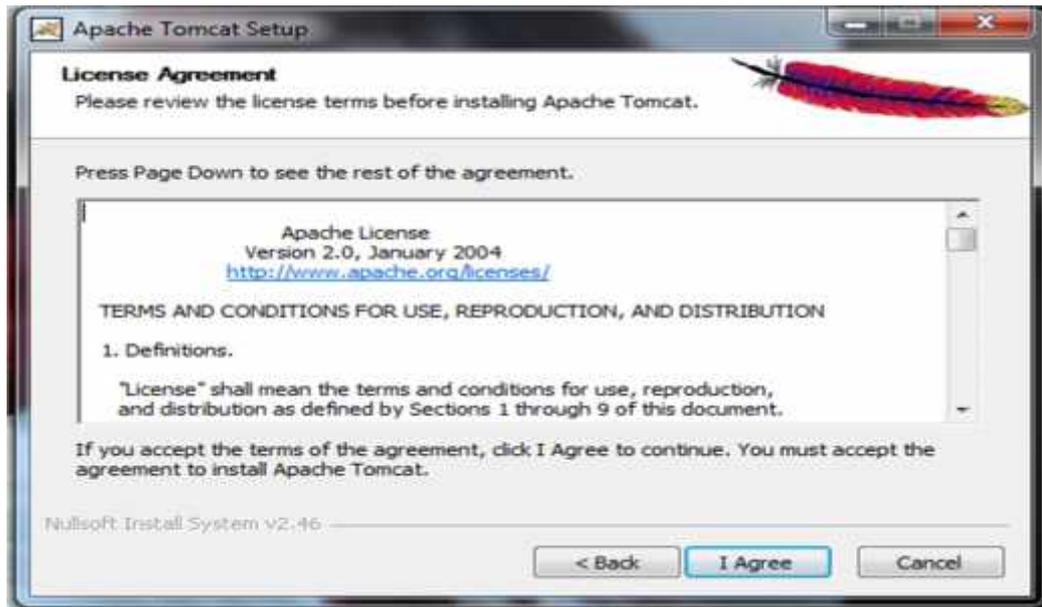
1.- doble clic sobre el archivo apache tomcat



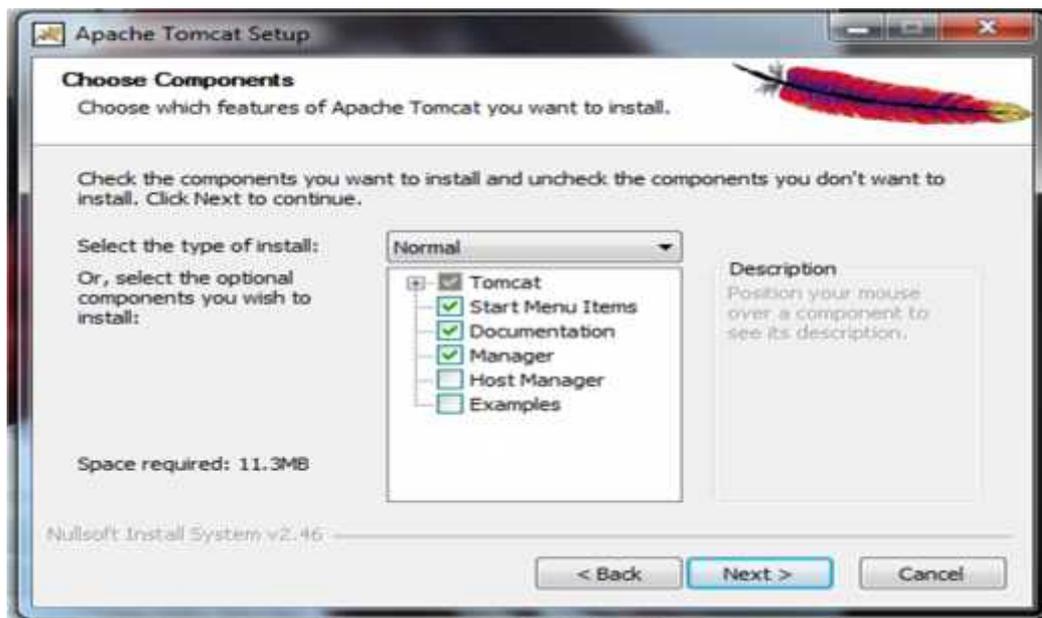
2.- Aprese la siguiente ventana, clic en next



3.- En esta ventana aceptamos los términos y condiciones del software

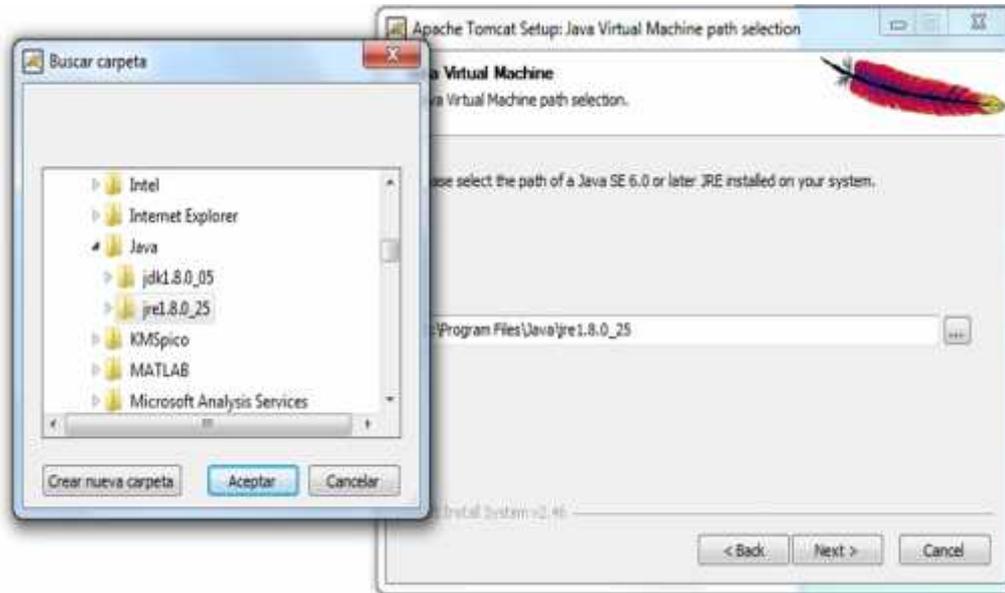


4.- Seleccionamos las opciones en que trabajaremos para posterior dar clic en next

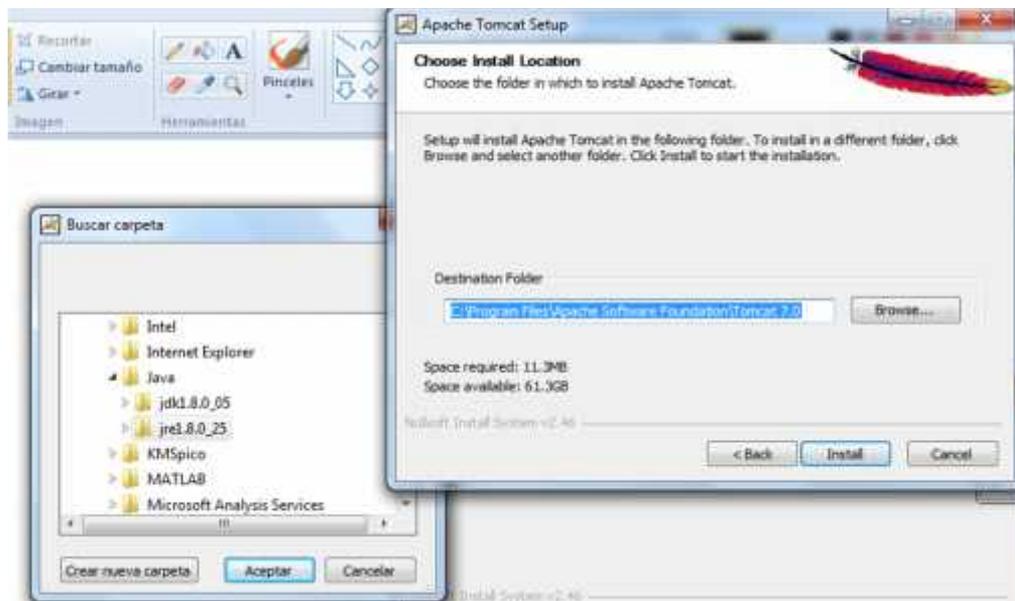


5.- en esta ventana ingresaremos los datos con los que trabajaremos para nuestro caso ingresaremos solo el user name y el password ya que lo utilizaremos como servidor interno, posterior damos clic en next.

6.- Ubicamos la carpeta donde instalamos el jdk presionamos next.



7.- Instalamos el software en la carpeta donde instalamos anteriormente el jdk



8.- Una vez instalado este es el panel de configuraciones del apache donde cargaremos nuestra aplicación.

9.- Ingresamos a la configuración utilizando el user name que ingresamos así como el password que ingresamos al instalar el software.



10.- verificamos que nuestra aplicación este ya en el servidor y luego la arrancamos para su funcionamiento.



Manual de Usuario

Manual de usuario

Introducción



Bienvenido al manual de usuarios del software de administración de bodega de la empresa Globalconnection año 2015.

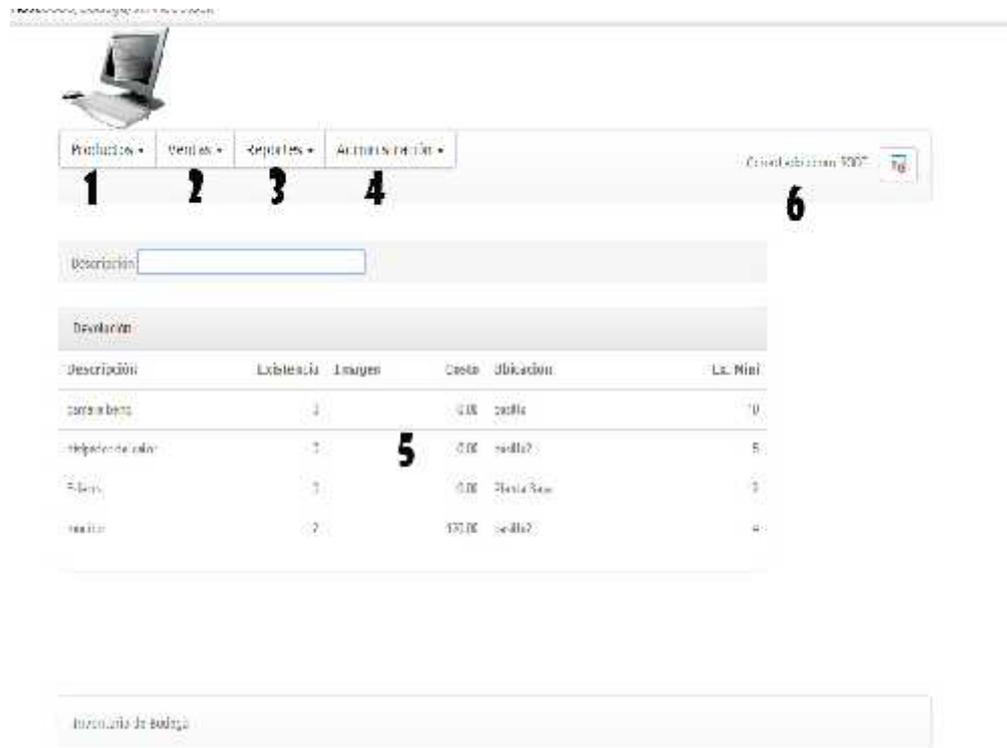
El software de administración de bodega facilita la búsqueda, organización de la información correspondiente a los productos que se manejan en el interior de las instalaciones optimizando el tiempo de respuesta para los usuarios.

Es importante leer este manual antes de comenzar a usar el software, ya que aquí se encuentra toda la información necesaria para el usuario que debe tener en cuenta, a fin de lograr la ejecución correcta de la misma, cabe destacar que es primordial leer este manual si se utiliza por primera vez este software.

El manual incluye los pasos para realizar la utilización del software de administración de bodega.

A continuación se describen cada una de las interfaces con sus respectivos botones y acciones que cumplen dentro del sistema.

Ventana de Administración



En la imagen cada número corresponde a una acción que el software ejecuta cómo se menciona a continuación.

- 1.- Boton para ingresar datos de los productos
- 2.- Boton para seleccionar salidas de productos
- 3.- Boton para ver reportes de los Productos
- 4.- Boton de seleccion de tipo usuario
- 5.- Sector de formularios del sistema
- 6.- Tipo de usuario que utiliza el sistema

Usuario

The image shows a web application interface for user registration. At the top, there is a navigation bar with menu items: 'Productos', 'Ventas', 'Reportes', and 'Administración'. To the right of the navigation bar, it says 'Conectado como ADMIN' with a user icon. Below the navigation bar, the title 'Usuario' is displayed. The registration form consists of several fields and buttons:

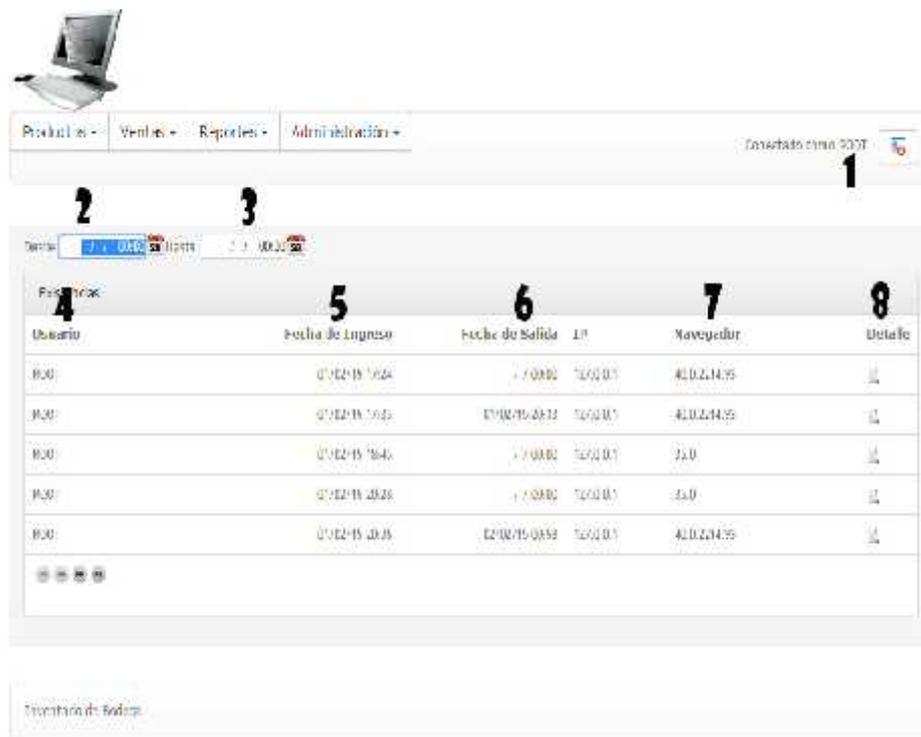
- 1**: A text input field for 'Nombre' (Name).
- 2**: A dropdown menu for 'Categoría' (Category) with 'ADMINISTRADOR' selected.
- 3**: A text input field for 'Login'.
- 4**: A text input field for 'Contraseña' (Password) with the example 'Ejemplo: Ejemplo@1234'.
- 5**: A text input field for 'Confirmación' (Confirmation) with the example 'Ejemplo: Ejemplo@1234'.
- 6**: A blue button labeled 'Guardar' (Save).
- 7**: A red button labeled 'Cancelar' (Cancel).

Below the form, there is a button labeled 'Inventario de Boleas'.

En la imagen cada número corresponde a una acción que el software ejecuta cómo se menciona a continuación.

- 1.- Campo para ingreso de nombre usuarios
- 2.- Campo selección de tipo de usuario
- 3.- Campo para ingreso de login de usuario
- 4.- Campo ingreso contraseña
- 5.- Campo para confirmación contraseña
- 6.- Botón guardar datos (actualizar base de datos)
- 7.- Botón cancelar

Auditoria



En la imagen cada número corresponde a una acción que el software ejecuta cómo se menciona a continuación.

- 1.- Tipo de usuario en sistema
- 2.- Fecha desde cuando se realiza la auditoria
- 3.- Fecha de finalización de la auditoria
- 4.- Tipo de usuario al que se realiza la auditoria
- 5.- Fecha que ingreso al sistema
- 6.- Fecha de salida de y dirección ip auditable
- 7.- Detalle de modificaciones en sistema
- 8.- Detalle para imprimir

Producto

Productos - Ventas - Reportes - Administración - Guardar (modo Ajax)

Producto

Descripción **1**

Existencia: 0

Imagen **2**

Costo: 0,00

Existencia Mínima **3**

Referencia:

Tipo **4** interno

Ubicación pasillo **5**

6 Guardar **7** Cancelar

En la imagen cada número corresponde a una acción que el software ejecuta cómo se menciona a continuación.

- 1.- Campo para ingreso nombre producto
- 2.- Campo para ingreso de imagen producto
- 3.- Campo para número de existencias mínima en bodega
- 4.- Campo de selección tipo producto
- 5.- Campo de selección de ubicación de producto dentro de las bodegas
- 6.- Botón guardar (actualizar la base)
- 7.- Botón cancelar proceso

Manual

Técnico

Manual técnico

Introducción



Bienvenido al manual de técnico del software de administración de bodega de la empresa Globalconnection año 2015.

Es importante leer este manual antes de comenzar a usar el software, ya que aquí se encuentra toda la información necesaria para el responsable de los mantenimientos del mismo y a su vez los datos principales que se debe tener en cuenta, a fin de lograr la ejecución correcta de la misma,

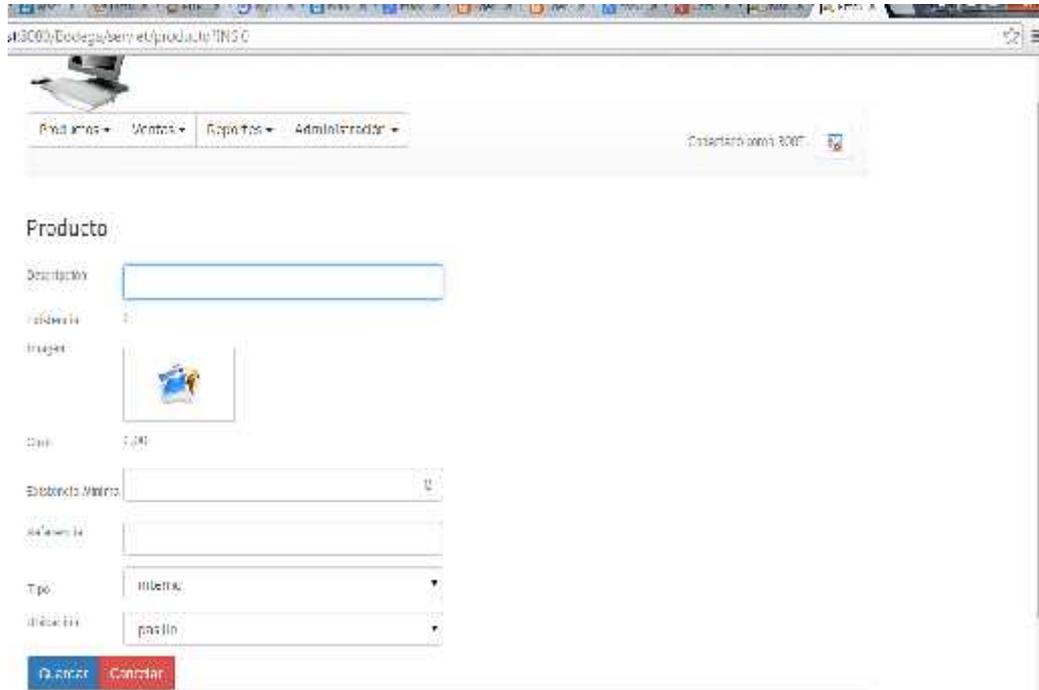
El manual incluye los pasos, los métodos y la estructura con que se a creado cada una de las vistas de usuarios así como las conexiones que se generan entre la base y el sistema en cuestión.

Este manual está orientado al personal responsable del mantenimiento del sistema así también al personal que esta capacitado en plataformas de desarrollo. Para una posterior modificación de las mismas a fin de mejorar, implementar nuevos requerimientos de la empresa.

A continuación se especificara una de las interfaces así como su respectiva programación

Interface ingreso de producto

Imagen



A continuación se detalla el código fuente de esta imagen desde la creación de la clase hasta las vistas generadas.

Código fuente clase producto

```

gx.evt.autoSkip = !1;

function producto() {

    this.ServerClass = "Producto";

    this.PackageName = "";

    this.setObjectType("trn");

    this.hasEnterEvent = !0;

    this.SetStandaloneVars = function() {

```

```
this.AV13ProductoId = gx.fn.getIntegerValue("vPRODUCTOID", ".");

this.A19ProductoId = gx.fn.getIntegerValue("PRODUCTOID", ".");

this.AV14Insert_TipoProductoId =
gx.fn.getIntegerValue("vINSERT_TIPOPRODUCTOID", ".");

this.AV11Insert_UbicacionId =
gx.fn.getIntegerValue("vINSERT_UBICACIONID", ".");

this.AV15mensaje = gx.fn.getControlValue("vMENSAJE");

this.AV16f = gx.fn.getControlValue("vF");

this.A40000ProductoImg_GXI =
gx.fn.getControlValue("PRODUCTOIMG_GXI");

this.A24ProductoEsta = gx.fn.getControlValue("PRODUCTOESTA");

this.AV17Pgmname = gx.fn.getControlValue("vPGMNAME");

this.Gx_mode = gx.fn.getControlValue("vMODE")

};

this.Valid_Productodesc = function() {

    gx.ajax.validSrvEvt("dyncall", "valid_Productodesc", ["gx.O.A19ProductoId",
"gx.O.A20ProductoDesc", "gx.O.AV15mensaje", "gx.O.AV16f"], ["AV15mensaje",
"AV16f"]);

    return!0

};

this.Valid_Productomini = function() {
```

```
try {  
  
    var a = gx.util.balloon.getNew("PRODUCTOMINI");  
  
    this.AnyError = 0;  
  
    if (0 == this.A28ProductoMini)  
  
        try {  
  
            a.setError("La existencia minima es requerida"), this.AnyError = 1  
  
        } catch (b) {  
  
        }  
  
    } catch (c) {  
  
    }  
  
    try {  
  
        return null == a ? !0 : a.show()  
  
    } catch (d) {  
  
    }  
  
    return!0  
  
};  
  
this.Valid_Tipoproductoid = function() {  
  
    gx.ajax.validSrvEvt("dyncall", "valid_Tipoproductoid",  
["gx.O.A27TipoProductoId"], []);  
  
    return!0  
}
```

```
};

this.Valid_Ubicacionid = function() {

    gx.ajax.validSrvEvt("dyncall", "valid_Ubicacionid", ["gx.O.A26UbicacionId"],
[]);

    return!0

};

this.e12012_client = function() {

    this.executeServerEvent("AFTER TRN",

        !0)

};

this.e13011_client = function() {

    this.executeServerEvent("ENTER", !0)

};

this.e14011_client = function() {

    this.executeServerEvent("CANCEL", !0)

};

this.GXValidFnc = [];

this.GXCtrlIds = [2, 5, 6, 12, 15, 17, 20, 22, 25, 27, 30, 32, 35, 37, 40, 42, 45, 47,
50, 52];

this.GXLastCtrlId = 52;
```

```
this.GXValidFnc[2] = {fld: "TABLEMAIN", grid: 0};

this.GXValidFnc[5] = {fld: "GROUPODATA", grid: 0};

this.GXValidFnc[6] = {fld: "TABLE1", grid: 0};

this.GXValidFnc[12] = {fld: "TABLE2", grid: 0};

this.GXValidFnc[15] = {fld: "TEXTBLOCKPRODUCTODESC", format: 0, grid:
0};

this.GXValidFnc[17] =

    {lvl: 0, type: "svchar", len: 40, dec: 0, sign: !1, ro: 0, grid: 0, gxgrid: null,
fnc: this.Valid_Productodesc, isvalid: null, fld: "PRODUCTODESC", gxz:
"Z20ProductoDesc", gxold: "O20ProductoDesc", gxvar: "A20ProductoDesc", ucs:
[], op: [17], ip: [17], nacdep: [], ctrltype: "edit", v2v: function(a) {

    gx.O.A20ProductoDesc = a

    }, v2z: function(a) {

    gx.O.Z20ProductoDesc = a

    }, v2c: function() {

    gx.fn.setControlValue("PRODUCTODESC", gx.O.A20ProductoDesc,
0)

    }, c2v: function() {

    gx.O.A20ProductoDesc = this.val()

    }, val: function() {

    return gx.fn.getControlValue("PRODUCTODESC")
```

```
    },  
  
    nac: function() {  
  
        return !1  
  
    }  
});  
  
this.GXValidFnc[20] = {fld: "TEXTBLOCKPRODUCTOEXIT", format: 0, grid:  
0};  
  
this.GXValidFnc[22] = {lvl: 0, type: "int", len: 4, dec: 0, sign: !1, pic: "ZZZ9", ro:  
1, grid: 0, gxgrid: null, fnc: null, isvalid: null, rgrid: [], fld: "PRODUCTOEXIT",  
gxz: "Z21ProductoExit", gxold: "O21ProductoExit", gxvar: "A21ProductoExit", ucs:  
[], op: [], ip: [], nacdep: [], ctrltype: "edit", v2v: function(a) {  
  
    gx.O.A21ProductoExit = gx.num.intval(a)  
  
    }, v2z: function(a) {  
  
    gx.O.Z21ProductoExit = gx.num.intval(a)  
  
    }, v2c: function() {  
  
    gx.fn.setControlValue("PRODUCTOEXIT",  
  
    gx.O.A21ProductoExit, 0)  
  
    }, c2v: function() {  
  
    gx.O.A21ProductoExit = this.val()  
  
    }, val: function() {  
  
    return gx.fn.getIntegerValue("PRODUCTOEXIT", ".")  
  
    }, nac: function() {
```

```
return !1

});

this.GXValidFnc[25] = {fld: "TEXTBLOCKPRODUCTOIMG", format: 0, grid:
0};

this.GXValidFnc[27] = {lvl: 0, type: "bits", len: 1024, dec: 0, sign: !1, ro: 0, grid:
0, gxgrid: null, fnc: null, isvalid: null, fld: "PRODUCTOIMG", gxz:
"Z22ProductoImg", gxold: "O22ProductoImg", gxvar: "A22ProductoImg", ucs: [],
op: [], ip: [], nacdep: [], ctrltype: "edit", v2v: function(a) {

    gx.O.A22ProductoImg =

    a

}, v2z: function(a) {

    gx.O.Z22ProductoImg = a

}, v2c: function() {

    gx.fn.setMultimediaValue("PRODUCTOIMG", gx.O.A22ProductoImg,
gx.O.A40000ProductoImg_GXI)

}, c2v: function() {

    gx.O.A40000ProductoImg_GXI = this.val_GXI();

    gx.O.A22ProductoImg = this.val()

}, val: function() {

    return gx.fn.getBlobValue("PRODUCTOIMG")

}, val_GXI: function() {
```

```
return gx.fn.getControlValue("PRODUCTOIMG_GXI")

}, gxvar_GXI: "A40000ProductoImg_GXI", nac: function() {

return!1

});

this.GXValidFnc[30] = {fld: "TEXTBLOCKPRODUCTOCOST", format: 0, grid:

0};

this.GXValidFnc[32] =

{lvl: 0, type: "decimal", len: 16, dec: 2, sign: !1, pic:

"ZZZZZZZZZZZZ9.99", ro: 1, grid: 0, gxgrid: null, fnc: null, invalid: null, fld:

"PRODUCTOCOST", gxz: "Z23ProductoCost", gxold: "O23ProductoCost", gxvar:

"A23ProductoCost", ucs: [], op: [], ip: [], nacdep: [], ctrltype: "edit", v2v: function(a)

{

gx.O.A23ProductoCost = gx.fn.toDecimalValue(a, ",", ".")

}, v2z: function(a) {

gx.O.Z23ProductoCost = gx.fn.toDecimalValue(a, ".", ",")

}, v2c: function() {

gx.fn.setDecimalValue("PRODUCTOCOST", gx.O.A23ProductoCost,

2, ",")

}, c2v: function() {

gx.O.A23ProductoCost =

this.val()

}
```

```
    }, val: function() {  
  
        return gx.fn.getDecimalValue("PRODUCTOCOST", ".", ",")  
  
    }, nac: function() {  
  
        return !1  
  
    }  
});  
  
this.GXValidFnc[35] = {fld: "TEXTBLOCKPRODUCTOCOST2", format: 0,  
grid: 0};  
  
this.GXValidFnc[37] = {lvl: 0, type: "int", len: 4, dec: 0, sign: !1, pic: "ZZZ9", ro:  
0, grid: 0, gxgrid: null, fnc: this.Valid_Productomini, isvalid: null, rgrid: [], fld:  
"PRODUCTOMINI", gxz: "Z28ProductoMini", gxold: "O28ProductoMini", gxvar:  
"A28ProductoMini", ucs: [], op: [37], ip: [37], nacdep: [], ctrltype: "edit", v2v:  
function(a) {  
  
    gx.O.A28ProductoMini = gx.num.intval(a)  
  
    },  
  
v2z: function(a) {  
  
    gx.O.Z28ProductoMini = gx.num.intval(a)  
  
    }, v2c: function() {  
  
    gx.fn.setControlValue("PRODUCTOMINI", gx.O.A28ProductoMini, 0)  
  
    }, c2v: function() {  
  
    gx.O.A28ProductoMini = this.val()  
  
    }, val: function() {
```

```
return gx.fn.getIntegerValue("PRODUCTOMINI", ".")

}, nac: function() {

    return !1

});

this.GXValidFnc[40] = {fld: "TEXTBLOCKPRODUCTOCODI", format: 0, grid:
0};

this.GXValidFnc[42] = {lvl: 0, type: "svchar", len: 40, dec: 0, sign: !1, ro: 0, grid:
0, gxgrid: null, fnc: null, isvalid: null, fld: "PRODUCTOCODI", gxz:
"Z25ProductoCodi", gxold: "O25ProductoCodi",

    gxvar: "A25ProductoCodi", ucs: [], op: [], ip: [], nacdep: [], ctrltype: "edit",
v2v: function(a) {

    gx.O.A25ProductoCodi = a

    }, v2z: function(a) {

    gx.O.Z25ProductoCodi = a

    }, v2c: function() {

    gx.fn.setControlValue("PRODUCTOCODI", gx.O.A25ProductoCodi, 0)

    }, c2v: function() {

    gx.O.A25ProductoCodi = this.val()

    }, val: function() {

    return gx.fn.getControlValue("PRODUCTOCODI")

    }, nac: function() {
```

```
return!l

});

this.GXValidFnc[45] = {fld: "TEXTBLOCKPRODUCTOCODI2", format: 0,
grid: 0};

this.GXValidFnc[47] = {lvl: 0, type: "int", len: 4, dec: 0, sign: !1, pic: "ZZZ9", ro:
0,

grid: 0, gxgrid: null, fnc: this.Valid_Tipoproductoid, isvalid: null, rgrid: [], fld:
"TIPOPRODUCTOID", gxz: "Z27TipoProductoId", gxold: "O27TipoProductoId",
gxvar: "A27TipoProductoId", ucs: [], op: [], ip: [47], nacdep: [], ctrltype:
"dyncombo", v2v: function(a) {

    gx.O.A27TipoProductoId = gx.num.intval(a)

}, v2z: function(a) {

    gx.O.Z27TipoProductoId = gx.num.intval(a)

}, v2c: function() {

    gx.fn.setComboBoxValue("TIPOPRODUCTOID",
gx.O.A27TipoProductoId)

}, c2v: function() {

    gx.O.A27TipoProductoId = this.val()

}, val: function() {

    return gx.fn.getIntegerValue("TIPOPRODUCTOID",
".")
```

```
    }, nac: function() {  
  
        return "INS" == this.Gx_mode && 0 != this.AV14Insert_TipoProductoId  
  
    }  
});  
  
this.GXValidFnc[50] = { fld: "TEXTBLOCKUBICACIONID", format: 0, grid:  
0};  
  
this.GXValidFnc[52] = { lvl: 0, type: "int", len: 4, dec: 0, sign: !1, pic: "ZZZ9", ro:  
0, grid: 0, gxgrid: null, fnc: this.Valid_Ubicacionid, isvalid: null, rgrid: [], fld:  
"UBICACIONID", gxz: "Z26UbicacionId", gxold: "O26UbicacionId", gxvar:  
"A26UbicacionId", ucs: [], op: [], ip: [52], nacdep: [], ctrltype: "dyncombo", v2v:  
function(a) {  
  
    gx.O.A26UbicacionId = gx.num.intval(a)  
  
    }, v2z: function(a) {  
  
    gx.O.Z26UbicacionId =  
  
        gx.num.intval(a)  
  
    }, v2c: function() {  
  
    gx.fn.setComboBoxValue("UBICACIONID", gx.O.A26UbicacionId)  
  
    }, c2v: function() {  
  
    gx.O.A26UbicacionId = this.val()  
  
    }, val: function() {  
  
    return gx.fn.getIntegerValue("UBICACIONID", ".")  
  
    }, nac: function() {
```

```
return "INS" == this.Gx_mode && 0 != this.AV11Insert_UbicacionId

});

this.O20ProductoDesc = this.Z20ProductoDesc = this.A20ProductoDesc = "";

this.O21ProductoExit = this.Z21ProductoExit = this.A21ProductoExit = 0;

this.O22ProductoImg = this.Z22ProductoImg = this.A22ProductoImg =

this.A40000ProductoImg_GXI = "";

this.O28ProductoMini =

    this.Z28ProductoMini = this.A28ProductoMini = this.O23ProductoCost =

this.Z23ProductoCost = this.A23ProductoCost = 0;

this.O25ProductoCodi = this.Z25ProductoCodi = this.A25ProductoCodi = "";

this.O26UbicacionId = this.Z26UbicacionId = this.A26UbicacionId =

this.O27TipoProductoId = this.Z27TipoProductoId = this.A27TipoProductoId = 0;

this.AV17Pgmname = this.A40000ProductoImg_GXI = "";

this.AV8IsAuthorized = !1;

this.AV9TrnContext = {};

this.AV18GXV1 = this.AV11Insert_UbicacionId =

this.AV14Insert_TipoProductoId = 0;

this.AV12TrnContextAtt = {};

this.AV7Context =

    {};
```

```
this.AV13ProductoId = 0;
```

```
this.AV10WebSession = {};  
  
this.A19ProductoId = 0;  
  
this.Gx_mode = this.A24ProductoEsta = this.AV16f = this.AV15mensaje = "";  
  
this.ServerEvents = ["e12012_client", "e13011_client", "e14011_client"];  
  
this.EnterCtrl = ["BTN_ENTER"];  
  
this.VarControlMap.AV13ProductoId = {id: "vPRODUCTOID", grid: 0, type:  
"int"};  
  
this.VarControlMap.A19ProductoId = {id: "PRODUCTOID", grid: 0, type: "int"};  
  
this.VarControlMap.AV14Insert_TipoProductoId = {id:  
"vINSERT_TIPOPDUCTOID", grid: 0, type: "int"};  
  
this.VarControlMap.AV11Insert_UbicacionId =  
  
    {id: "vINSERT_UBICACIONID", grid: 0, type: "int"};  
  
this.VarControlMap.AV15mensaje = {id: "vMENSAJE", grid: 0, type: "svchar"};  
  
this.VarControlMap.AV16f = {id: "vF", grid: 0, type: "svchar"};  
  
this.VarControlMap.A40000ProductoImg_GXI = {id: "PRODUCTOIMG_GXI",  
grid: 0, type: "svchar"};  
  
this.VarControlMap.A24ProductoEsta = {id: "PRODUCTOESTA", grid: 0, type:  
"char"};  
  
this.VarControlMap.AV17Pgmmname = {id: "vPGMNAME", grid: 0, type:  
"char"};  
  
this.VarControlMap.Gx_mode = {id: "vMODE", grid: 0, type: "char"};
```

```
this.SetStandaloneVars()  
  
}  
  
producto.prototype = new gx.GxObject;  
  
gx.setParentObj(new producto);
```

Análisis

Este código corresponde la interface de ingreso de productos en bodega así como los métodos y los controles que se utilizaron para la misma