



INSTITUTO TECNOLÓGICO
“CORDILLERA”

CARRERA ANALISIS DE SISTEMAS

CONTROL DE INVENTARIO DE BODEGA Y FACTURACIÓN MEDIANTE
UNA APLICACIÓN WEB PARA LA EMPRESA OCÉANO PACÍFICO

Proyecto de investigación previo a la obtención del título de tecnólogo en Análisis
de Sistemas.

Autor: Alex Darío Cuichán Apunte.

Tutor: Lic. Patricia Garzón Msc.

Quito, Octubre 2014

APROBACION DE TUTOR

En mi calidad de tutor del trabajo sobre el tema: “**CONTROL DE INVENTARIO DE BODEGA Y FACTURACIÓN MEDIANTE UNA APLICACIÓN WEB PARA LA EMPRESA OCÉANO PACÍFICO.**” presentado por el ciudadano: **Alex Darío Cuichán Apunte**, estudiante de la Escuela de Sistemas, considero que dicho informe reúne los requisitos y méritos suficientes para ser sometido a la evaluación por parte del Tribunal de Grado, que el Honorable Consejo de Escuela designe, para su correspondiente estudio y calificación.

Quito Noviembre del 2014

Lic. Patricia Garzón Msc.

TUTOR

Ing. Pablo Aguilera

LECTOR



DECLARATORIA

Declaro que la investigación es absolutamente original, auténtica, 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.

Alex Darío Cuichán Apunte

CC: 1717554339

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 **ALEX DARÍO CUICHÁN APUNTE**, 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 "**CONTROL DE INVENTARIO DE BODEGA Y FACTURACIÓN MEDIANTE UNA APLICACIÓN WEB PARA LA EMPRESA OCÉANO PACÍFICO**", 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 reconvenición, 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 24 días del mes de Octubre del dos mil catorce.

f) _____
C.C. N° 1717554339
Alex Cuichán

f) _____
Instituto Superior Tecnológico Cordillera
CESIONARIO



AGRADECIMIENTOS

El presente trabajo de titulación primeramente me gustaría agradecer a Dios por bendecirme, darme las fuerzas y la sabiduría para llegar hasta donde he llegado.

A mi tutor de tesis, Lic. Patricia Garzón Msc. por su esfuerzo y dedicación, quien con sus conocimientos, su experiencia, su paciencia y su motivación ha logrado en mí que pueda terminar mis estudios con éxito.

A mis compañeros, de clase con quienes hemos compartido experiencias y con quienes hemos superado los retos que nos han puesto en esta lucha.

Al INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA por darme la oportunidad de estudiar y ser un profesional.



DEDICATORIA

A mis padres y hermano, que con dedicación y
Paciencia supieron guiarme por el camino correcto.

A todos mis familiares y amigos

A mis profesores por sus enseñanzas



ÍNDICE GENERAL

Título	pagina
APROBACION DE TUTOR	i
AGRADECIMIENTOS	ii
CONTRATO DE CESIÓN SOBRE DERECHOS PROPIEDAD INTELECTUAL	iii
DEDICATORIA	vii
ÍNDICE GENERAL.....	ix
ÍNDICE DE TABLAS	xii
ÍNDICE DE FIGURAS.....	xiv
RESUMEN EJECUTIVO	xvi
ABSTRACT	xvii
Capítulo I: Antecedentes	1
1.02 Justificación.....	1
1.01 Contexto	1
1.03 Definición del Problema Central.....	2
Capítulo II: Análisis de involucrado	4
2.01 Requerimientos.	4
2.01.01 Descripción del sistema actual	4
2.01.02 Visión y Alcance	4
2.01.03 Entrevista.....	6
2.01.04 Matriz de Requerimiento.....	7
2.01.05 Detalle de requerimientos	8
2.02 Mapeos de Involucrados	15
2.03 Matriz de involucrados.....	16
Capitulo III: Problemas y Objetivos	18
3.01. Árbol de problemas.....	18
3.02. Árbol de objetivos.....	19
3.03. Diagrama de casos de usos.....	20
3.04. Casos de uso de realización	21
3.05 Diagramas de secuencia del sistema	30

3.05 Especificación de casos de uso	34
Capítulo IV: Análisis de alternativas	36
4.01 Matriz de Análisis de Alternativas	36
4.02 Matriz de Impactos de Objetivos	37
4.03 Estándares para el diseño de clases	37
4.04 Diagrama de Clases	38
4.05 Modelo lógico – físico	39
4.06 Diagrama de componentes.	40
4.07 Diagrama de Estrategias.....	41
4.08 Matriz de Marco lógico.....	42
4.09 Vistas Arquitectónicas	42
4.09.01 Vista lógica.....	42
4.09.02 Vista Física.....	43
4.09.03 Vista de Desarrollo.....	43
4.09.04 Vista de Procesos	45
Capítulo V: Propuesta	46
5.01 Especificación de estándares de programación.....	46
5.02 Diseño de interfaces de usuario.....	47
5.03 Especificación de pruebas de unidad	51
5.04 Especificación de pruebas de aceptación	52
5.05 Especificación de pruebas de carga.....	53
5.06 Configuración del Ambiente mínima/ideal.....	54
Capítulo VI: Aspectos Administrativos	55
6.01. Recursos	55
6.02. Presupuesto	56
<i>Presupuesto</i>	56
6.03. Cronograma.....	56
Capítulo VII: Conclusiones y Recomendaciones.....	57
7.01 Conclusiones	57
7.02 Recomendaciones.....	57



ANEXOS	59
A.01 Matriz de Impacto de Objetivos	61
A.02 Estándares para el diseño de clases	62
A.03 Matriz de Marco lógico.....	65
A.04 Cronograma.....	67
A.06 Manual de Usuario	70
A.07 Manual Técnico.....	78
A.08 Manual de instalación.....	106
BIOGRAFIA	121



ÍNDICE DE TABLAS

Título	pagina
Tabla 1_Matriz T.....	3
Tabla 2 Entrevista	6
Tabla 3_Matriz de requerimientos	7
Tabla 4_Detalle de requerimientos RF001	8
Tabla 5_Detalle de requerimientos RF002	9
Tabla 6_Detalle de requerimientos RF003	10
Tabla 7_Detalle de requerimientos RF004	11
Tabla 8_Detalle de requerimientos NRF001.....	12
Tabla 9_Detalle de requerimientos NRF002.....	13
Tabla 10_Detalle de requerimientos NRF003	14
Tabla 11_Matriz de análisis de involucrados.....	16
Tabla 12_Especificación CUR001 Registro de usuario.....	21
Tabla 13_Especificación CUR002 Registro clientes	22
Tabla 14_Especificación CUR003 Registro de productos.....	23
Tabla 15_Especificación CUR04 Control inventario	24
Tabla 16_Especificación CUR005 Alerta de stock.....	25
Tabla 17_Especificación CUR006 Informe de reportes de productos.....	26
Tabla 18 Especificación CUR007 Factura.....	27
Tabla 19_Especificación CUR008 Pago factura.....	28
Tabla 20_Especificación CUR009 Anular factura.....	29
Tabla 21_Especificación CU001 Registro usuarios.....	34
Tabla 22_Especificación CU002 Registro productos	35
Tabla 23_Especificación CU003 Control inventario	35
Tabla 24_Especificación CU004 Facturación.....	35
Tabla 25_Matriz de Análisis de Alternativas.....	36
Tabla 26 Descripción de los estándares de control	46
.....	46



Tabla 27 Ingreso al Sistema	47
Tabla 28 Menú	48
Tabla 29_Agregar Proveedor	49
Tabla 30 Inventario	49
Tabla 31_Administrar producto	50
Tabla 32_Especificación PU001 login.....	51
Tabla 33_Especificación PU002 validaciones	51
Tabla 34_Especificación PA001 Administrador.....	52
Tabla 35_Especificación PA002 Secretaria	52
Tabla 36_Especificación PA003 Bodeguero	53
Tabla 37_Especificación PC001 Carga de usuarios.....	53
Tabla 38_Recursos	55
Tabla 39_Presupuesto del proyecto	56



ÍNDICE DE FIGURAS

Título	pagina
Figura 1: Mapeo de involucrados.....	15
Figura 2: Árbol de Problemas	18
Figura 3: Árbol de Objetivos	19
Figura 4: Diagrama de Caso de Uso General:.....	20
Figura 5: Registro de usuarios	21
Figura 6: Registro clientes	22
Figura 7: Registro de productos	23
Figura 8: Control inventario.....	24
Figura 9: Alerta de stock.....	25
Figura 10: Informe de reportes.....	26
Figura 11: Factura.....	27
Figura 12: Pagar factura.....	28
Figura 13: Anular factura.....	29
Figura 14: Registro de usuarios:	30
Figura 16: Registro producto	31
Figura 17: Control de inventario.....	31
Figura 18: Alerta de stock.....	32
Figura 19: Informes de productos	32
Figura 20: Factura	33
Figura 21: Pagar factura.....	33
Figura 22: Anular factura.....	34
Figura 23: Diagrama de Clases	38
Figura 24: Modelo lógico – físico.....	39
Figura 26: negocio	40
Figura 27: presentación.....	41
Figura 28: diagrama de estrategias.....	41
Figura 29: Vista lógica.....	42
Figura 30: vista física.....	43
Figura 31: Capa acceso de datos	43
Figura 32: Capas lógica del negocio	44



Figura 33: Capa presentación.....	44
Figura 34: Vista de procesos.....	45
Figura 35: login.....	47
Figura 36: Menú principal	48
Figura 37: Ingreso de proveedor	48
Figura 38: ingreso de producto	49
Figura 39: lista de producto	50



RESUMEN EJECUTIVO

El presente proyecto tiene como objetivo general el control inventario de bodega y facturación dentro de la empresa OCEANO PACIFICO ubicado en Quito, parroquia de Cotocollao. En el cual se realizará un análisis de requerimientos y necesidades, que permitan proponer el desarrollo de un sistema web de control de inventario y facturación.

En el capítulo I se puede encontrar el contexto y justificación del proyecto a realizarse y la definición del problema central.

En el capítulo II se detalla el análisis de involucrados de la empresa y podemos ver los requerimientos que necesita dicha empresa

En el capítulo III podemos identificar los problemas y objetivos del tema central.

En el capítulo IV podemos encontrar diferentes formas de solucionar un problema, o lo que es igual, distintos medios para alcanzar un objetivo.

En el capítulo V se encuentra la propuesta para el sistema a realizarse:

En el capítulo VI se detalla los aspectos administrativos como los recursos que fueron utilizados en la construcción del proyecto.

En el capítulo VII se detalla las conclusiones y recomendaciones de todo el trabajo desarrollado.

ABSTRACT

This project has the general objective of warehouse inventory control and billing within the company OCEÁNO PACÍFICO located in Quito, Cotacollao parish. In which an analysis of needs and requirements, allowing propose the development of a web system inventory control and billing will be done.

In chapter I can find the context and rationale for the project to be carried and defining the core problem.

In Chapter II analysis of the company involved detailed requirements and we can see that the company needs .

In Chapter III we can identify the problems and objectives of the central theme.

In Chapter IV we can find different ways to solve a problem, or what is the same, different means to an end.

In chapter V the proposed system is done:

In Chapter VI administrative aspects as resources that were used in the construction of the project is detailed.

In Chapter VII the conclusions and recommendations of all our efforts is detailed.



Capítulo I: Antecedentes

1.02 Justificación

El presente proyecto busca crear una aplicación web que ayude y permita el control de inventarios y facturación para optimizar el proceso de atención al cliente.

Es conveniente desarrollar la herramienta informática e implementarla, porque está orientada a mejorar los problemas que afronta actualmente dicha empresa en todos los procesos de control de productos e incluso de servicio. Esto es para mejorar un desempeño en sus actividades y un mayor aprovechamiento de los recursos humanos, de tiempo, materiales, y financieros.

1.01 Contexto

El control de inventario de bodega y facturación es el proceso de operaciones para definir el costo total del inventario. El control de inventario es una operación principal de una compañía u organización que maneja la recepción, almacenamiento adecuado y distribución de elementos.

Las empresas se encuentran en constantes cambios e innovación en la atención a los clientes que requieren de sus servicios y productos ya que es de suma importancia que el cliente se sienta satisfecho por la atención brindada por los empleados de dicha institución.

Por ejemplo, la información asociada a las características de un producto tales como precio, marca, proveedor, cantidad, etc. se los lleva de forma física, digital y se los guarda en carpetas. Además llevan un registro manuscrito de los productos que posee en inventario con sus respectivos movimientos de entrada y salida.

Un sistema de inventario de bodega y facturación es un conjunto de políticas y controles que supervisan los niveles de inventario y establece cuales son los niveles que debe mantenerse, cuando hay que registrar un producto. Una forma práctica de establecer un sistema de inventario es llevar la cuenta de cada artículo que sale de la empresa y colocar una orden por más existencias cuando los inventarios lleguen a un nivel predeterminado.

Con estos antecedentes y con apoyo de la investigación, es indispensable implementar una herramienta tecnológica automatizada que mejore y controle los procesos de inventario y facturación de una manera rápida, clara y precisa, para facilitar la gestión de los administradores, los mismos que analizan e interpretan los registros que genera el sistema, optimizando del tiempo, y garantizando la integridad de la información.

1.03 Definición del Problema Central

En la siguiente se presenta las situaciones: empeorada, actual y mejorada mediante la cual describiremos los extremos a evitar, la problemática en el presente y lo que buscamos al hacer nuestra investigación. Además describe las fuerzas mediante la cual son calificadas con la intensidad y el potencial de campo en la escala de 1 al 5.

Tabla 1

Matriz T

ANÁLISIS DE FUERZAS T					
Situación Empeorada	Situación Actual				Situación Mejorada
Pérdida económica por no controlar el inventario.	Inadecuado control de inventario de bodega y facturación				Óptimo control de procesos en el inventario y facturación de productos
Fuerzas Impulsadoras	I	PC	I	PC	Fuerzas Bloqueadoras
Proceso de registro de inventario de bodega de forma manual.	3	4	3	4	No llevar control adecuado del ingreso y salida de los productos.
Información de proveedores archivado de manera física.	3	4	2	4	Poca comunicación entre los administradores de la empresa.
Proceso de facturación se realiza manualmente.	4	3	3	4	El control de las facturas físicas lleva tiempo y es susceptible a errores.
Pérdida de información de los artículos.	2	3	2	3	Desinterés en la automatización de tareas y el uso de herramientas informáticas.

Nota: Determinación de las fuerzas bloqueadoras e impulsadoras.
PC = Potencial de Cambio; I = Intensidad

Capítulo II: Análisis de involucrado

2.01 Requerimientos.

2.01.01 Descripción del sistema actual

La empresa Océano Pacífico en la actualidad posee un método de registro de inventario de bodega manualmente. El bodeguero está encargado de llevar el inventario de los repuestos y materiales, como por ejemplo los artículos existentes, entradas y salidas de productos, toda esta información se la almacena en hojas, una vez obtenida la información debe ser registrada en archivos de Excel. A causa de esto la información esta insegura, causa problemas al momento de buscar información, en muchos casos la información es redundante, y no se puedan notar pérdidas de productos de manera rápida y oportuna.

El manejo de la facturación es llevada mediante los talonarios otorgados por el SRI.

El cajero es el encargado de facturar a los clientes. Por lo tanto el registro de las facturas en el área de contabilidad se las lleva en archivos diseñados en Excel. Lo que hace un proceso más difícil. Al momento de registrar facturas anuladas, devoluciones. En este caso también la información a llegado a ser es redundante, es un proceso poco factible, e incómodo. Es un proceso inseguro, ya que no cuentan con los respaldos de los archivos tanto de las bodegas como de los libros del área de contabilidad.

2.01.02 Visión y Alcance

Visión

El sistema se enfocará en automatizar el control de inventario y facturación de bodega tales como la administración de mercadería con la exactitud de la misma, también

controlar sus proveedores por nombres, días de pedidos, entregas, valor, cantidad y ventas diarias los cuales en el día se encuentran actualizados.

Alcance

El resultado que se desea obtener al desarrollar la aplicación es generar informes de bodega y facturación

Módulo de seguridad

En este módulo se restringirá el acceso al sistema de acuerdo al rol de cada usuario.

Módulo de mantenimiento

El sistema podrá actualizar así como también registrar la información de bodega y factura.

Módulo de procesos

Para acceder al sistema cada usuario ingresara su id y contraseña según su rol y cargo ya registrados en el sistema. La manipulación de datos de bodega será registrado en la base de.

En facturación proveerá al usuario de las herramientas para dar un correcto tratamiento a los registros que han sido importados del módulo de interfaz del inventario.

2.01.03 Entrevista

Tabla 2

Entrevista

identificador: 001		
Preguntas	Objetivos	Análisis posterior
¿Cuál es la problemática que buscan solucionar a través del sistema?	Determinar los problemas a solucionar con el sistema informático que se creara	El sistema deberá ser capaz de ingresar cada producto indicando fecha de ingreso y la cantidad de artículos. El sistema debe permitir actualizar la información de la compra de los artículos. El sistema debe permitir consultar modificar el estado de los artículos en venta. La interfaz del sistema deberá ser implementada como una aplicación web. La interfaz debe ser en idioma español
¿Quiénes manejarán el sistema informático que se creara para el control de inventario y facturación?	Determinar quienes tendrán acceso al sistema de inventario y facturación.	Los que utilizaran y tendrán acceso al sistema web son administrador y bodeguero. Cada usuario que desee ingresar al sistema deberá introducir en la página principal un código de usuario y una contraseña la cual será validada por el sistema, dándole acceso o enviándole un mensaje para que introduzca nuevamente sus datos.
¿Cuál es el proceso de facturación?	Determinar cuál es el proceso para facturar un artículo de salida de la empresa.	El sistema deberá generar factura de salida de dicho artículo.

2.01.04 Matriz de Requerimiento

Tabla 3

Matriz de requerimientos

Identificador	Descripción	Fuente	Prioridad	Tipo	Estado	Usuarios Involucrados
REQUERIMIENTOS FUNCIONALES						
RF001	El sistema permitirá ingresar un nuevo registro de compra de producto indicando fecha de ingreso y la cantidad de artículos.	Administrador Bodega Secretaria	Alta	Software	En revisión	Administrador Bodeguero
RF002	El sistema debe permitir actualizar la venta de un producto	Administrador Bodega Secretaria	Alta	Software	En revisión	Administrador Bodeguero secretaria
RF003	El sistema deberá alertar el mínimo de stock de un producto.	Administrador Secretaria	Alta	Software	En revisión	Secretaria
RF004	El sistema deberá generar factura.	bodeguero	Alta	Software	En revisión	bodeguero
NRF001	La interfaz del sistema deberá ser implementada en una aplicación web.	Administrador	Alta	Software	En revisión	Administrador secretaria
NRF002	La interfaz debe ser en idioma español.	Administrador	Alta	Software	En revisión	Administrador
NRF003	Cada usuario que desee ingresar al sistema deberá introducir en la página principal un código de usuario y una contraseña.	Administrador	Alta	Software	En revisión	Administración

2.01.05 Detalle de requerimientos

Tabla 4

Detalle de requerimientos RF001

El sistema deberá ser capaz de ingresar un nuevo registro de compra de producto indicando fecha de ingreso y la cantidad de artículos.

Creado por	Estado	Análisis	
Alex Cuichán	Alex Cuichán	Actualizado por	Alex Cuichán
Fecha de creación	17/07/2014	Fecha de actualización	30/07/2014

Identificador	RF001		
----------------------	--------------	--	--

Estado de requerimiento	critico	Tipo de requerimiento	Funcional
Datos de entrada	1.- código de producto 2.- datos de producto		
Descripción	Una vez ingresado los códigos del producto se procede al almacenamiento y registro de dichos artículos.		
Datos de salida	Registro de bodega.		
Resultados Esperados	Adecuado registro de producto.		
Origen	Administrador		
Dirigido a	Administrador, bodeguero, secretaria		
Prioridad	3		
Requerimientos Asociados	Ninguno		

ESPECIFICACIÓN

Precondiciones	<ul style="list-style-type: none"> - Que se haya realizado la compra - Entrega de producto por parte del proveedor
Pos condiciones	No existe.
Criterios de Aceptación	El usuario podrá establecer el tipo de producto ingresado al sistema.

Tabla 5

Detalle de requerimientos RF002

El sistema debe permitir actualizar la venta de un producto.

Creado por	Estado Alex Cuichán	Análisis Actualizado por	Alex Cuichán
Fecha de creación	17/07/2014	Fecha de actualización	30/07/2014

Identificador	RF002		
Estado de requerimiento	critico	Tipo de requerimiento	Funcional
Datos de entrada	Producto registrado		
Descripción	El sistema debe permitir consultar, modificar, estados de la información de los productos.		
Datos de salida	kardex		
Resultados Esperados	Actualización de venta		
Origen	Bodeguero		
Dirigido a	Administrador, bodeguero		
Prioridad	3		
Requerimientos Asociados	Ninguno		

ESPECIFICACIÓN

Precondiciones	Actualizar venta de producto
Pos condiciones	No existe.
Criterios de Aceptación	Permite la información detallada del tipo de articulo

Tabla 6

Detalle de requerimientos RF003

El sistema deberá alertar el mínimo de stock de un producto.

	Estado	Análisis	
Creado por	Alex Cuichán	Actualizado por	Alex Cuichán
Fecha de creación	17/07/2014	Fecha de actualización	30/07/2014

Identificador	RF003		
Estado de requerimiento	Critico	Tipo de requerimiento	Funcional
Datos de entrada	Mínimo de stock de producto		
Descripción	El sistema deberá alertar cuando un producto esté a punto de agotarse,		
Datos de salida	Alerta de stock		
Resultados Esperados	Actualización de producto		
Origen	Bodeguero		
Dirigido a	Administrador, bodeguero		
Prioridad	3		
Requerimientos Asociados	Ninguno		

ESPECIFICACIÓN

Precondiciones	Producto en existencia
Pos condiciones	No existe.
Criterios de Aceptación	Alerta de stock

Tabla 7

Detalle de requerimientos RF004

El sistema deberá generar factura

Creado por	Estado Alex Cuichán	Análisis Actualizado por Alex Cuichán	Alex Cuichán
Fecha de creación	17/07/2014	Fecha de actualización	30/07/2014

Identificador	RF004		
Estado de requerimiento	Critico	Tipo de requerimiento	Funcional
Datos de entrada	Datos del cliente Detalle del articulo vendido		
Descripción	Generar factura al comprar dicho articulo		
Datos de salida	Factura		
Resultados Esperados	Adecuado proceso de facturación.		
Origen	Cajera		
Dirigido a	Administrador, cajera, secretaria		
Prioridad	3		
Requerimientos Asociados	Ninguno		

ESPECIFICACIÓN

Precondiciones	Para ejecutar el proceso de facturación el cliente deberá ser registrado.
Pos condiciones	No existe
Criterios de Aceptación	Registro de la información personal del cliente y detalle de articulo de compra.

Tabla 8

Detalle de requerimientos NRF001

La interfaz del sistema deberá ser implementada en una aplicación web.

Creado por	Estado Alex Cuichán	Análisis Actualizado por	Alex Cuichán
Fecha de creación	17/07/2014	Fecha de actualización	30/07/2014

Identificador	NRF001		
Estado de requerimiento	critico	Tipo de requerimiento	No Funcional
Datos de entrada	Ingreso al sistema Debe implementarse en una aplicación web para tener acceso al sistema en cualquier parte facilitando así al usuario.		
Descripción			
Datos de salida	Aplicación vía web		
Resultados Esperados	Implementada en la web		
Origen	Administrador		
Dirigido a	Administrador, bodeguero, secretaria, cajero		
Prioridad	3		
Requerimientos Asociados	Ninguno		

ESPECIFICACIÓN

Precondiciones	Para permitir el ingreso al sistema el usuario debe estar registrado
Pos condiciones	No existe.
Criterios de Aceptación	Permitirá el manejo del sistema en cualquier sitio disponible.

Tabla 9

Detalle de requerimientos NRF002

La interfaz debe ser en idioma español.

Creado por	Estado Alex Cuichán	Análisis Actualizado por	Alex Cuichán
Fecha de creación	17/07/2014	Fecha de actualización	30/07/2014

Identificador	NRF002		
Estado de requerimiento	critico	Tipo de requerimiento	No Funcional
Datos de entrada	Ingreso al sistema El interfaz como también el contenido del sistema que se implementara debe ser idioma español.		
Descripción			
Datos de salida	Contenido español		
Resultados Esperados	El contenido debe ser en español		
Origen	Administrador		
Dirigido a	Administrador, secretaria, bodeguero, cajero		
Prioridad	3		
Requerimientos Asociados	Ninguno		

ESPECIFICACIÓN

Precondiciones	Para ejecutar el requerimiento el usuario debe ingresar al sistema
Pos condiciones	No existe.
Criterios de Aceptación	Permite que el usuario que manejara el sistemas le sea entendible.

Tabla 10

Detalle de requerimientos NRF003

Cada usuario que desee ingresar al sistema deberá introducir en la página principal un código de usuario y una contraseña.

Creado por	Estado Alex Cuichán	Análisis Actualizado por Alex Cuichán	
Fecha de creación	17/07/2014	Fecha de actualización	30/07/2014

Identificador	NRF003		
Estado de requerimiento	critico	Tipo de requerimiento	No Funcional
Datos de entrada	Cada uno de los usuarios debe estar registrado en el sistema		
Descripción	1.- Ingresar al sistema 2.- seleccionar su rol 3.- ingresar datos de usuario		
Datos de salida	Ingreso al sistema		
Resultados Esperados	Seguridad en el sistema de acuerdo a el tipo de rol		
Origen	Administrador		
Dirigido a	Administrador, bodeguero, secretaria, cajero		
Prioridad	3		
Requerimientos Asociados	Ninguno		

ESPECIFICACIÓN

Precondiciones	Ingresar al sistema de acuerdo al rol que desempeña
Pos condiciones	
Criterios de Aceptación	Cada usuario que desee ingresar al sistema deberá introducir en la página principal un código de usuario y una contraseña la cual será validada por el sistema, dándole acceso o enviándole un mensaje para que introduzca nuevamente sus datos.

2.02 Mapeos de Involucrados

El análisis de los involucrados nos permite, identificar a aquellas personas y Organizaciones interesadas en el éxito de un proyecto, aquellas que contribuyen o son afectadas por los objetivos del mismo, y a quienes tienen un poder de Influencia sobre los problemas que hay que enfrentar.

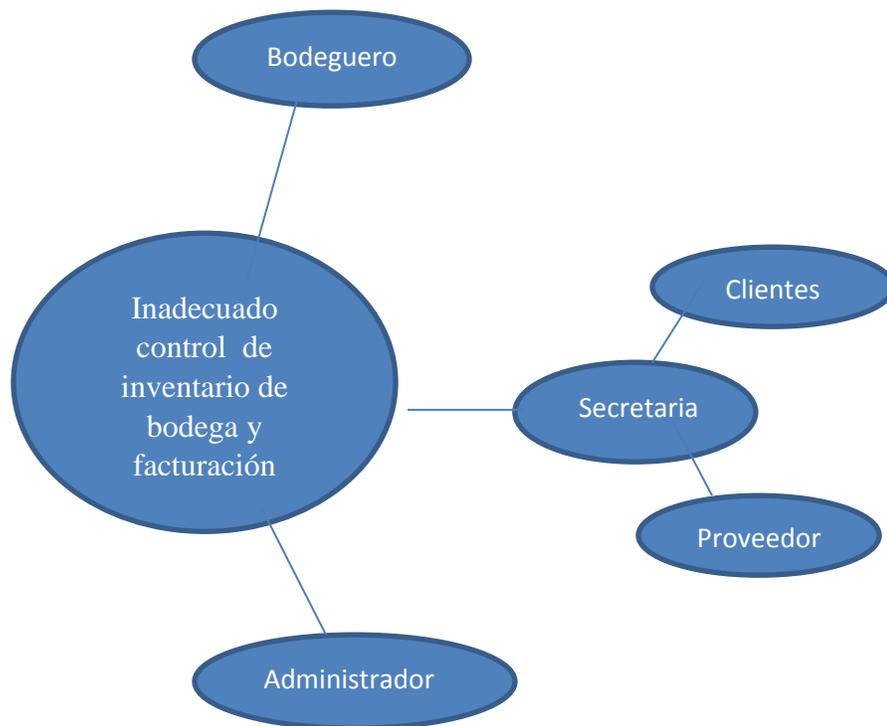


Figura 1: Mapeo de involucrados: El análisis de involucrados es un instrumento esencial que permite identificar a aquellas personas interesadas en un determinado proyecto.

2.03 Matriz de involucrados

Tiene como propósito analizar a los involucrados con su respectivo interés, los problemas percibidos, en el desarrollo de sus actividades sociales a través de los recursos que mantiene el proyecto para comprobar la efectividad de su producción.

Tabla 11

Matriz de análisis de involucrados

Actores Involucrados	Intereses sobre el problema central	Problemas percibidos	Recursos, Mandatos y Capacidades	Intereses sobre el Proyecto	Conflictos Potenciales
Secretaria	Llevar de manera eficaz el proceso de facturación.	Falta de una aplicación para el manejo de información de la factura.	Recurso humano Recurso tecnológico.	Mantener el control adecuado de facturación.	.perdida de información
bodeguero	-Mejorar el registro de ingreso de mercadería -Mejorar control de stock de mercadería	Conflictos al momento de buscar la mercadería ingresada de bodega	Recurso tecnológico.	Mantener un control adecuado de registro de mercadería mediante una aplicación web.	Perdida de mercadería en bodega.
administrador	Mejorar al momento de generar reportes de inventario y facturación	No puede encontrar la información de manera efectiva	Recurso económico Recurso tecnológico	Generar informes y reportes de manera adecuada.	Falta de interés para facilitar un sistema web

Nota: Analiza la matriz de involucrados

En la matriz mostrada anteriormente se analizarán las estrategias que realizarán los diferentes tipos de actores mencionados en el mapeo de involucrados, buscando soluciones a los distintos problemas presentados para el desarrollo del proyecto.

También se ha realizado el respectivo análisis de involucrados, los cuales muestran intereses sobre el problema central, generando diferentes interrogantes en cada uno de dichos involucrados permitiendo encontrar cuales son los puntos favorables y no favorables para su mejor desempeño dentro de la empresa, y así llegar a cumplir su objetivo, que es poder solucionar el problema central antes mencionado.

Capítulo III: Problemas y Objetivos

3.01. Árbol de problemas.

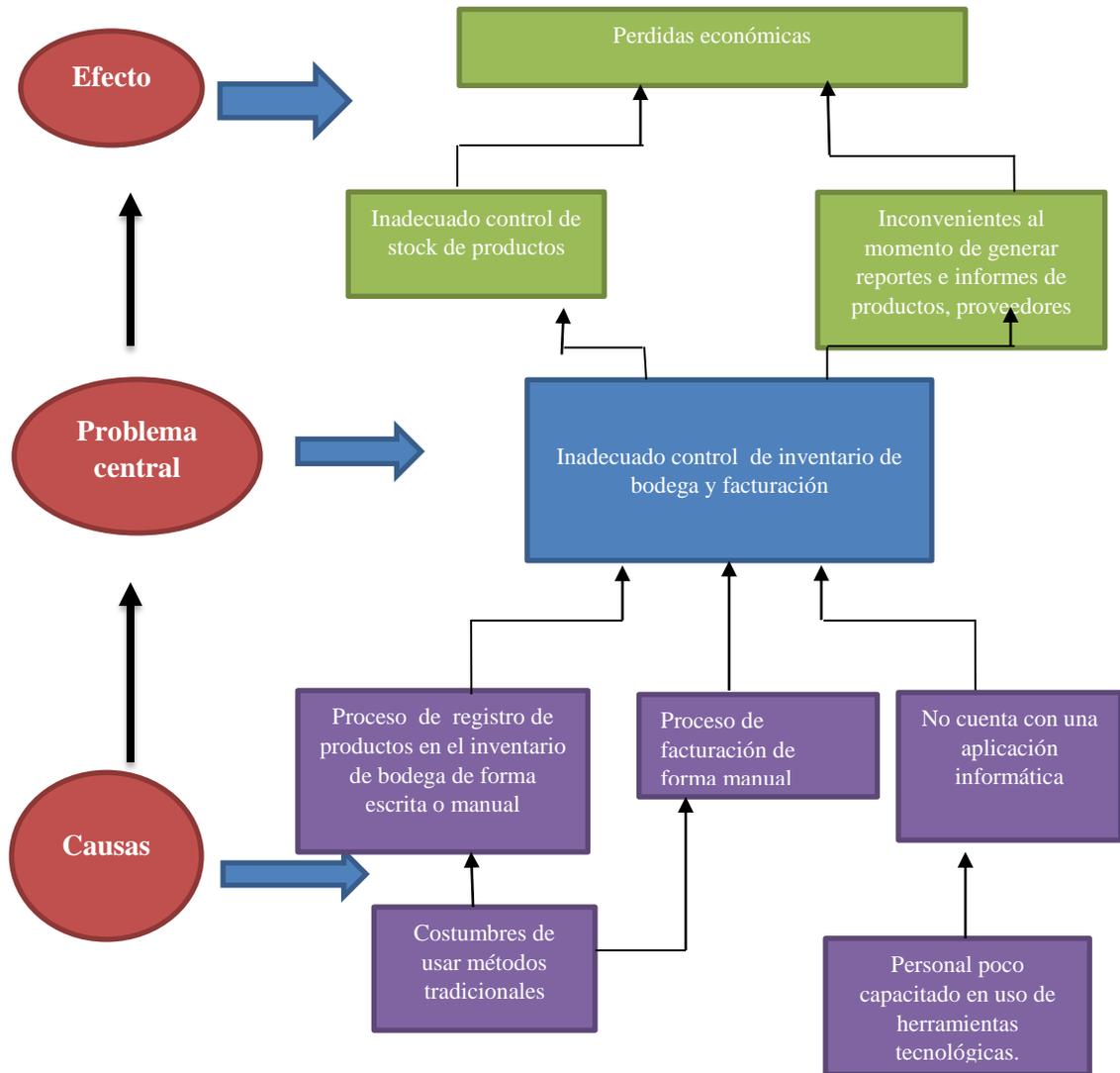


Figura 2: Árbol de Problemas: Es importante porque nos ayuda a entender problemática a resolver. En él se expresan causa y efecto, las situaciones negativas observadas por los involucrados en relación con el problema en cuestión.

3.02. Árbol de objetivos.

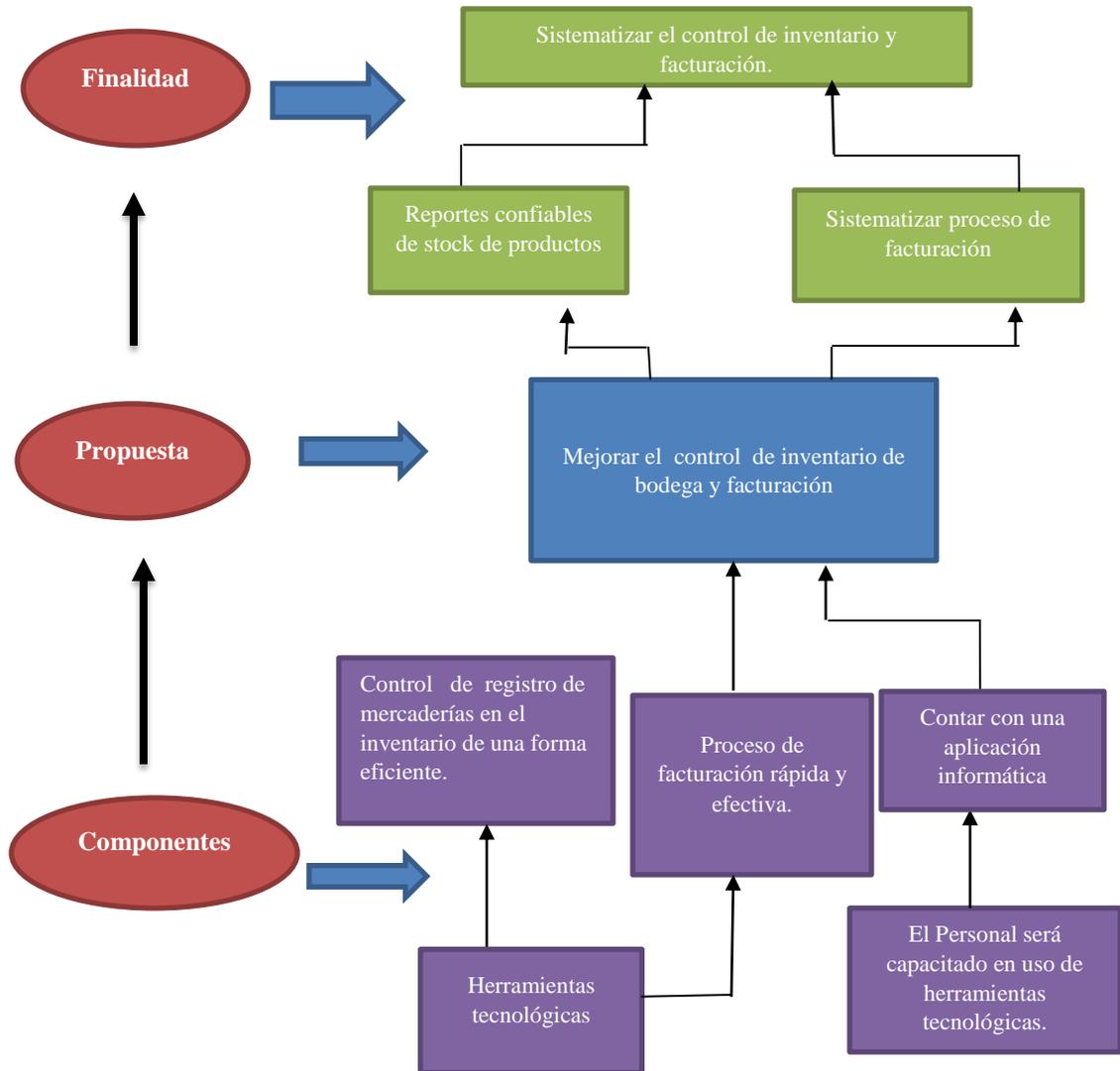


Figura 3: Árbol de Objetivos: Facilita la determinación de un objetivo del proyecto, como también permite; identificar los objetivos y alternativas para la realización del proyecto y describir la situación futura que será lograda cuando se resuelvan los problemas identificados anteriormente.

3.03. Diagrama de casos de usos.

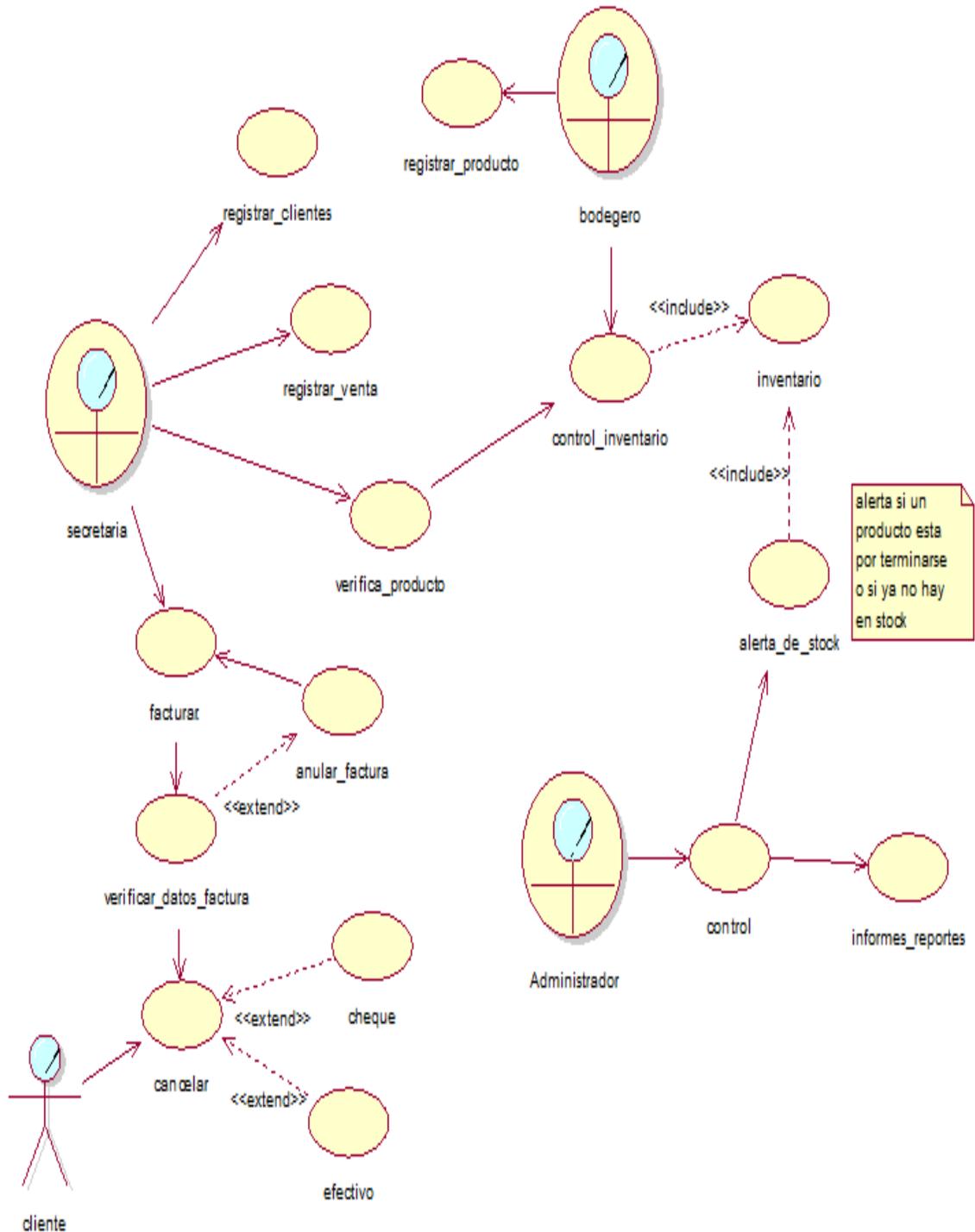


Figura 4: Diagrama de Caso de Uso General: representa los diferentes actores, pasos y acciones los cuales se realizan en la empresa para llevar a cabo una actividad o proceso.

3.04. Casos de uso de realización

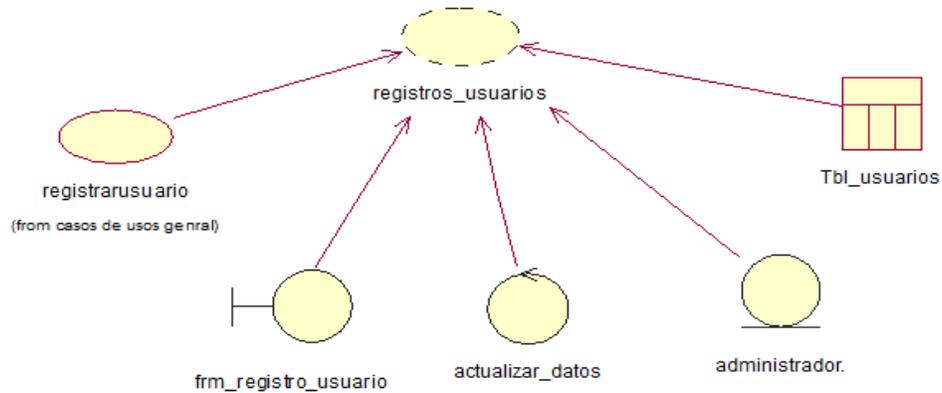


Figura 5: Registro de usuarios

Tabla 12

Especificación CUR001 Registro de usuario

NOMBRE	Registro de usuario
IDENTIFICADOR	CUR001
RESPONSABILIDADES	Ingresar datos de usuario
TIPO	sistema
REFERENCIAS	
CASOS DE USO	Casos de uso general
REFERENCIAS	
REQUISITOS	NRF003
PRECONDICIONES	Debe acceder a un navegador, ingresar al sistema y seleccionar el rol de usuario. Previamente debe estar registrado en la base de datos del sistema
POSCONDICIONES	
	Validar datos de usuario
SALIDAS PANTALLA	
	Ingreso al sistema

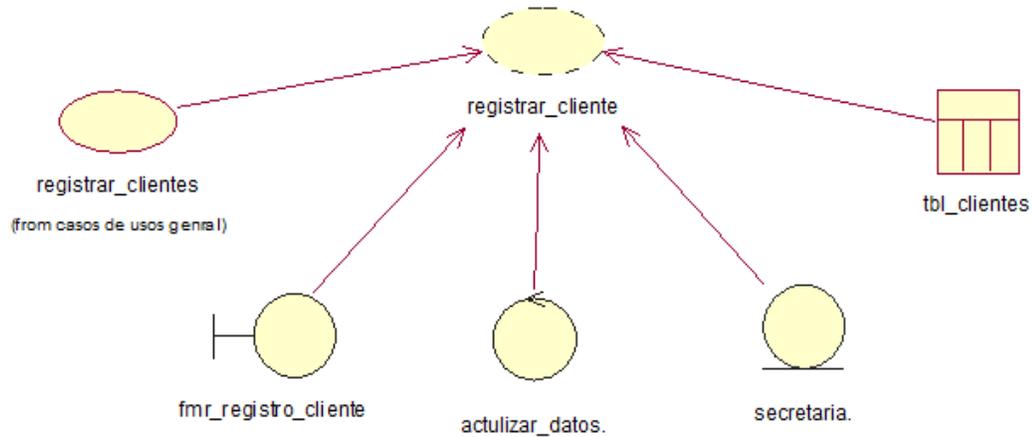


Figura 6: Registro clientes

Tabla 13

Especificación CUR002 Registro clientes

NOMBRE	Registro clientes
IDENTIFICADOR	CUR002
RESPONSABILIDADES	Ingresar datos de cliente
TIPO	Sistema
REFERENCIAS	
CASOS DE USO	Casos de general
REFERENCIAS	
REQUISITOS	NRF003
PRECONDICIONES	Ingresar datos del cliente en caso de ser nuevo, caso contrario se lo encontrara en el sistema y se actualizara los datos.
POSCONDICIONES	
	Validar cedula
SALIDAS PANTALLA	
	Guardar información

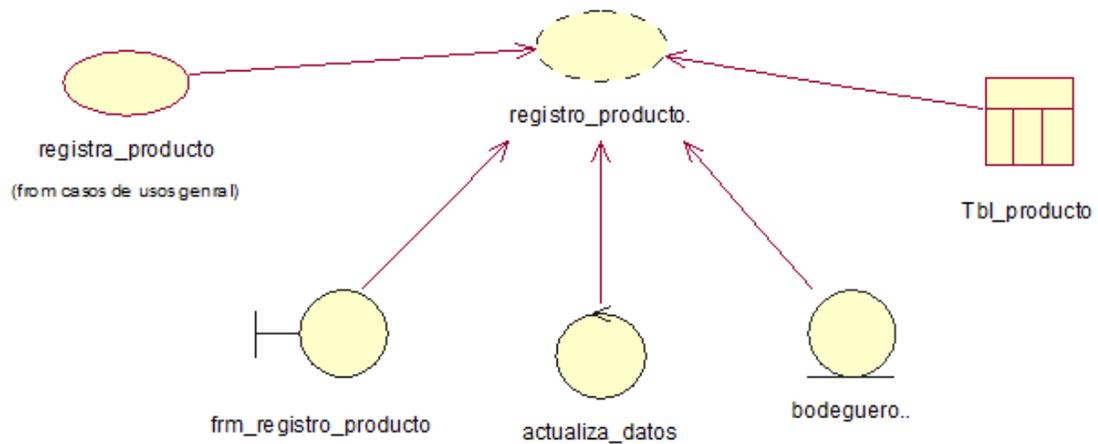


Figura 7: Registro de productos

Tabla 14

Especificación CUR003 Registro de productos

NOMBRE	Registro de productos
IDENTIFICADOR	CUR003
RESPONSABILIDADES	Ingresar datos de producto
TIPO	Sistema
REFERENCIAS	
CASOS DE USO	Casos de uso general
REFERENCIAS	
REQUISITOS	RF001
PRECONDICIONES	Registrar producto Guardar producto
POSCONDICIONES	Validar stock de producto
SALIDAS PANTALLA	Guardar información

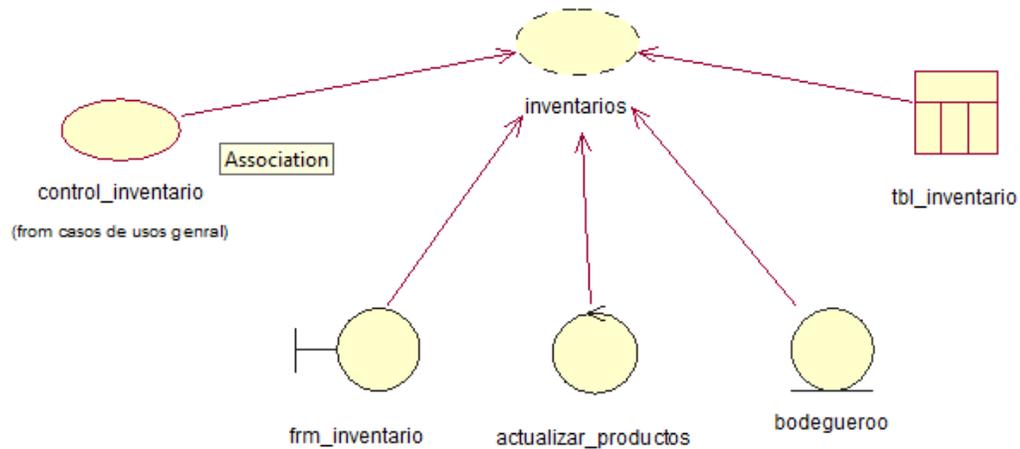


Figura 8: Control inventario

Tabla 15

Especificación CUR04 Control inventario

NOMBRE	Control inventario
IDENTIFICADOR	CUR004
RESPONSABILIDADES	Agregar la información de los productos de entrada y salida
TIPO	Sistema
REFERENCIAS	
CASOS DE USO	Casos de uso general
REFERENCIAS	
REQUISITOS	RF001, RF002
PRECONDICIONES	Se realiza un control del inventario de todos los productos. Disponer de un computador. Mantener la información actualizada y ordenada.
POSCONDICIONES	Guardar información de productos en una base de datos
SALIDAS PANTALLA	
	Ingreso al sistema

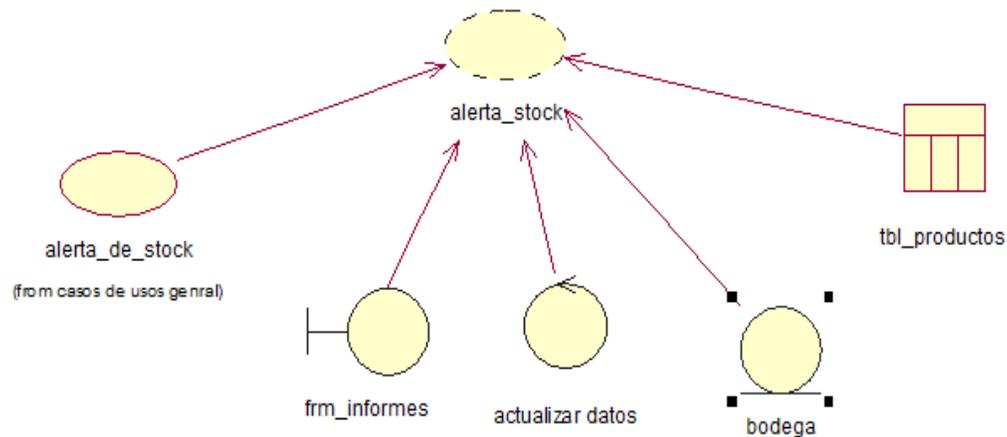


Figura 9: Alerta de stock:

Tabla 16

Especificación CUR005 Alerta de stock

NOMBRE	Alerta de stock
IDENTIFICADOR	CUR005
RESPONSABILIDADES	Realizar alerta de stock de un producto
TIPO	Sistema
REFERENCIAS	
CASOS DE USO	Casos de uso general
REFERENCIAS	
REQUISITOS	RF003
PRECONDICIONES	Consultar existencia de productos. Verificar datos de productos. Mantener la información actualizada y ordenada.
POSCONDICIONES	Mensaje de alerta de stock mínimo en el sistema
SALIDAS PANTALLA	Mensaje de error

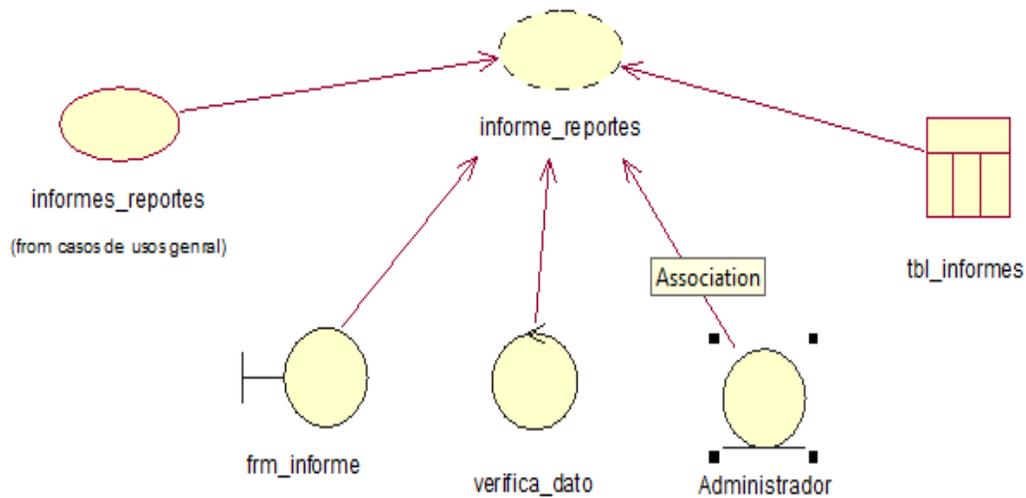


Figura 10: Informe de reportes

Tabla 17

Especificación CUR006 Informe de reportes de productos

NOMBRE	Informe de reportes de productos
IDENTIFICADOR	CUR006
RESPONSABILIDADES	Realizar reportes de los productos en stock
TIPO	Sistema
REFERENCIAS	
CASOS DE USO	Casos de uso general
REFERENCIAS	
REQUISITOS	RF001, RF003
PRECONDICIONES	Los informes deben ser entregados al administrador en periodos mensuales. Los informes deben presentarse de forma detallada y especifica.
POSCONDICIONES	Ingresar las fechas de consulta para la generación de los informes.
SALIDAS PANTALLA	Imprimir informes

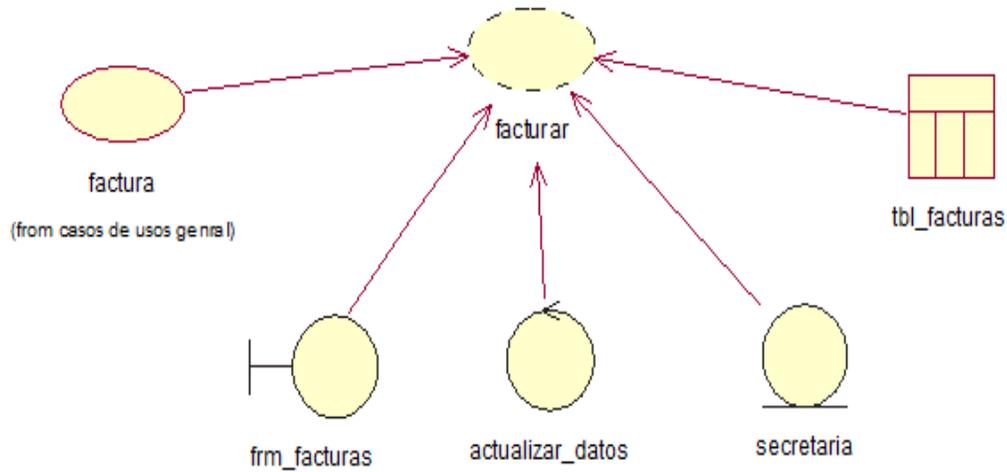


Figura 11: Factura:

Tabla 18

Especificación CUR007 Factura

NOMBRE	Factura
IDENTIFICADOR	CUR007
RESPONSABILIDADES	Generar la venta efectuada
TIPO	Sistema
REFERENCIAS	
CASOS DE USO	Casos de uso general
REFERENCIAS	
REQUISITOS	RF004
PRECONDICIONES	Los datos de cliente deben de ser correctos. La factura debe ser entregada al cliente. La información debe ser actualizada.
POSCONDICIONES	En caso de que se registre incorrectamente la información, el sistema permitirá a la secretaria la corrección de la misma.
SALIDAS PANTALLA	factura

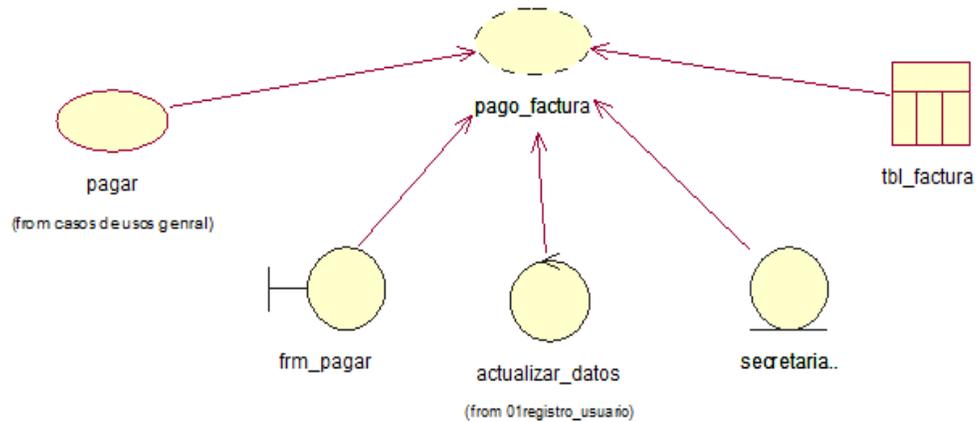


Figura 12: Pagar factura

Tabla 19

Especificación CUR008 Pago factura

NOMBRE	Pago factura
IDENTIFICADOR	CUR008
RESPONSABILIDADES	Proceso de pago de factura
TIPO	Sistema
REFERENCIAS	
CASOS DE USO	Casos de uso general
REFERENCIAS	
REQUISITOS	RF004
PRECONDICIONES	El cliente realiza el pago de su factura. Se verifica si los datos son correctos. La información debe ser actualizada.
POSCONDICIONES	En caso de que no pueda pagar en efectivo, el cliente realizará su pago con cheque.
SALIDAS PANTALLA	Factura

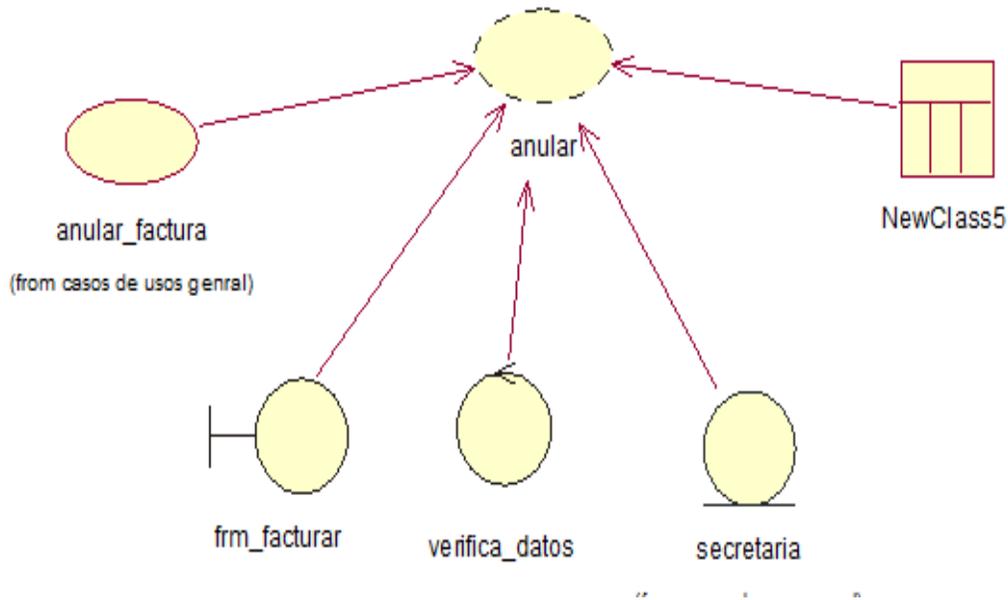


Figura 13: Anular factura

Tabla 20

Especificación CUR009 Anular factura

NOMBRE	Anular factura
IDENTIFICADOR	CUR009
RESPONSABILIDADES	Anulación de factura
TIPO	Sistema
REFERENCIAS	
CASOS DE USO	Casos de uso general
REFERENCIAS	
REQUISITOS	RF004
PRECONDICIONES	El cliente solicita la anulación de su factura. Se verifica si los datos son correctos y si su pedido es procedente.
POSCONDICIONES	Anulación de factura
SALIDAS PANTALLA	Factura

3.05 Diagramas de secuencia del sistema

3.05.01 Registro de usuarios

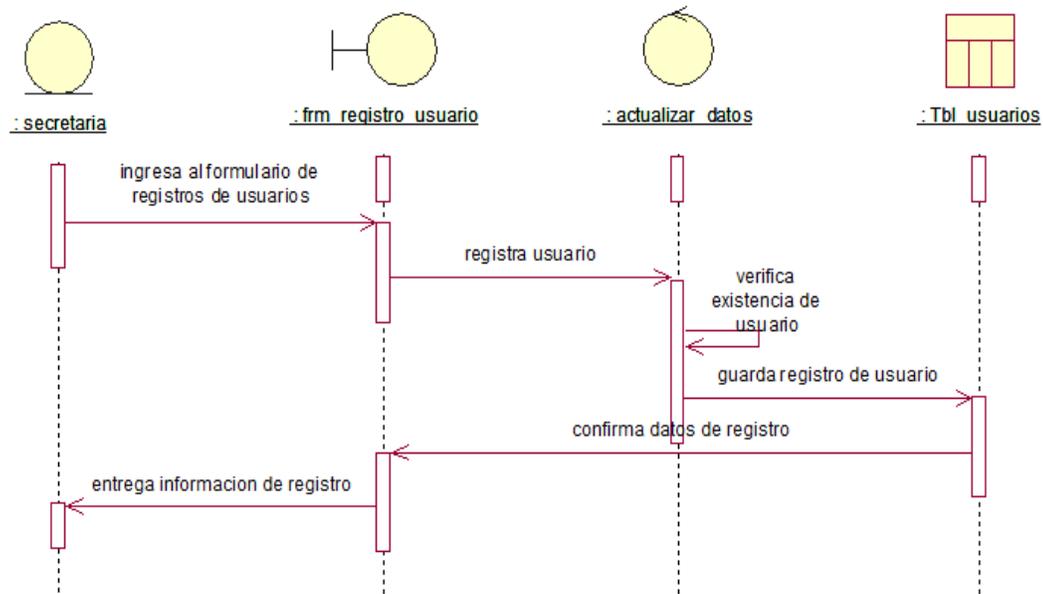


Figura 14: Registro de usuarios:

3.05.02 Registro cliente

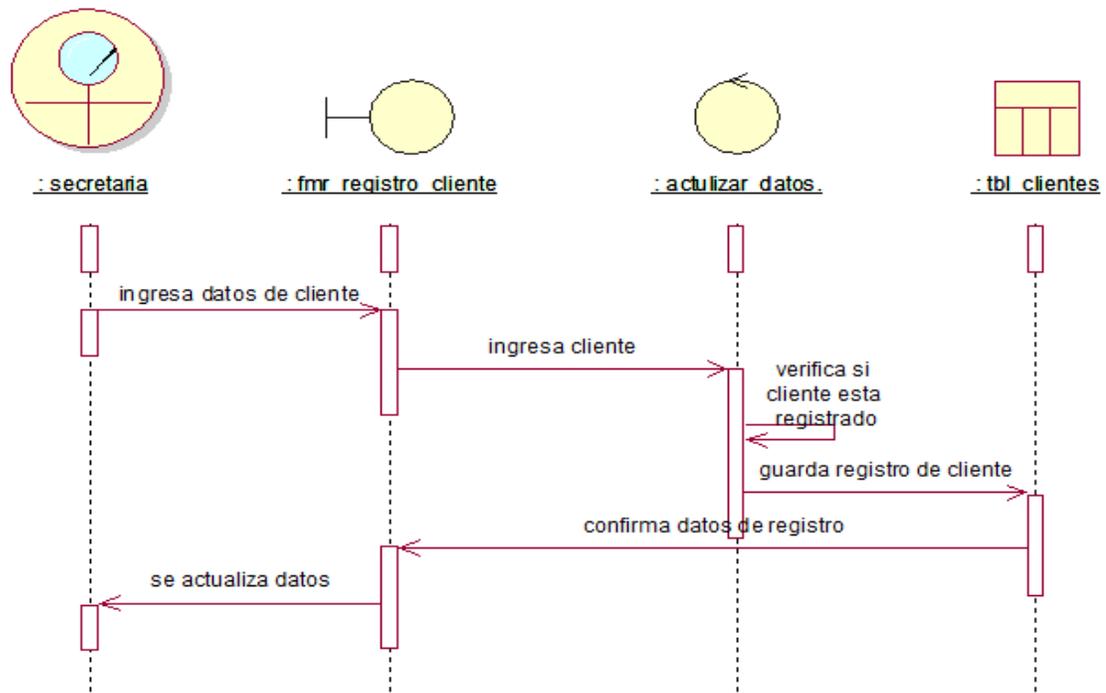


Figura 15: Registro cliente

3.05.03 Registro producto

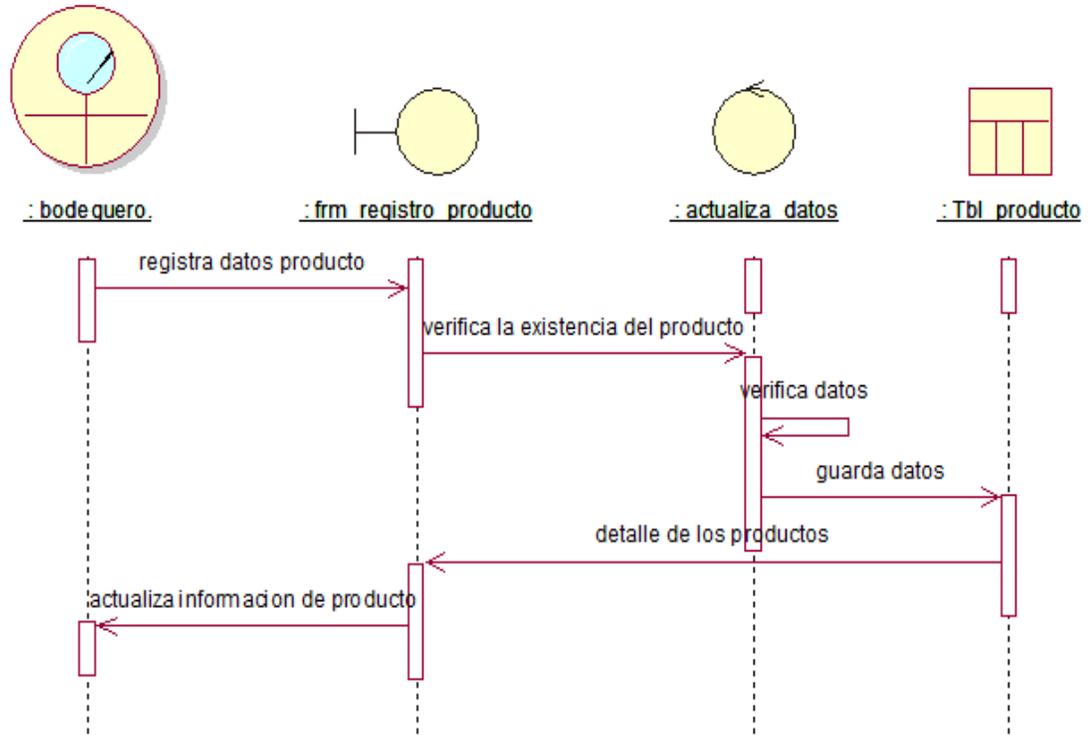


Figura 16: Registro producto

3.05.04 Control de inventario

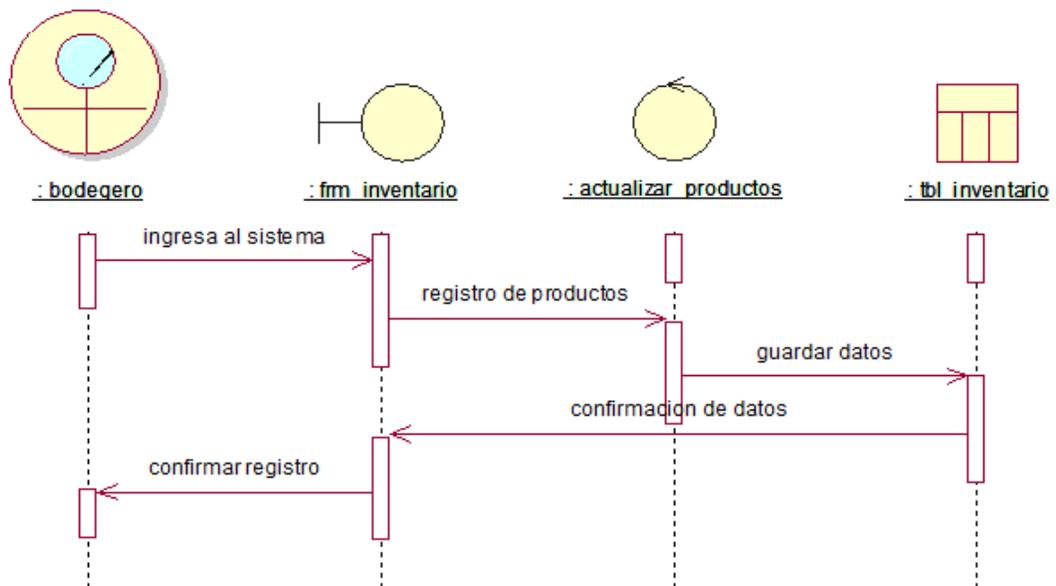


Figura 17: Control de inventario

3.05.05 Alerta de stock

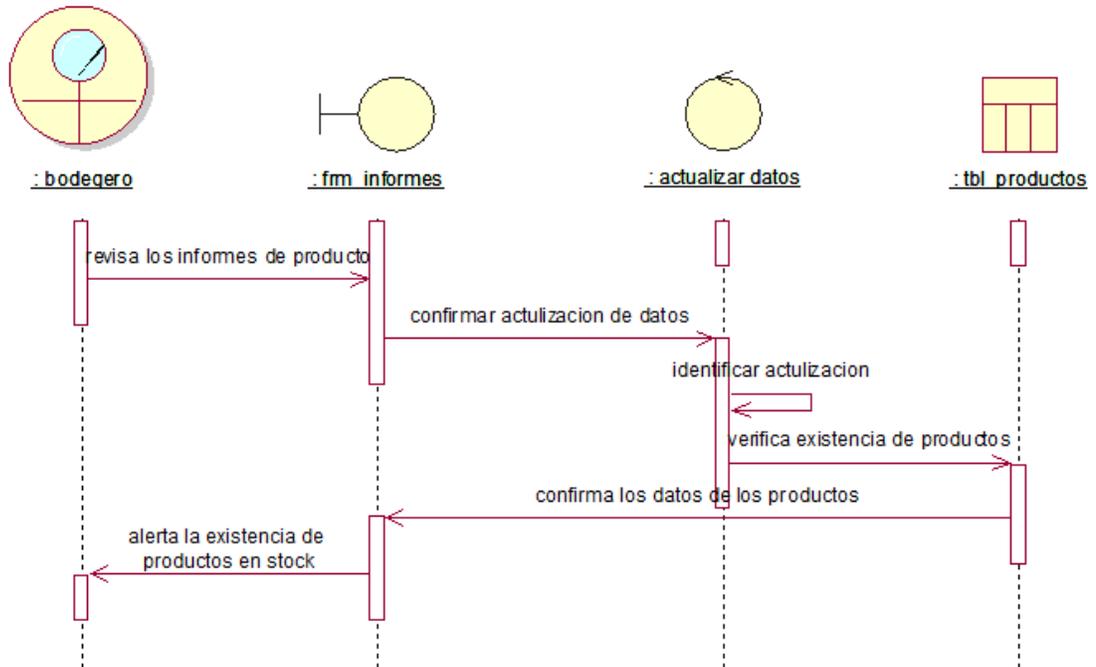


Figura 18: Alerta de stock

3.05.06 Informes de productos

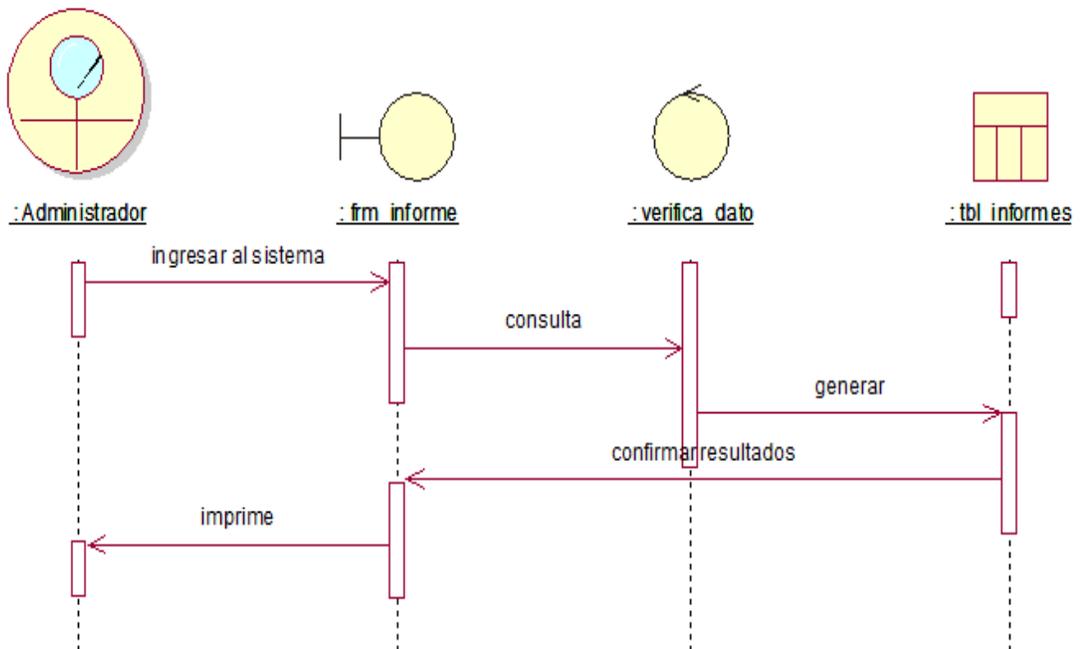


Figura 19: Informes de productos

3.05.07 Factura

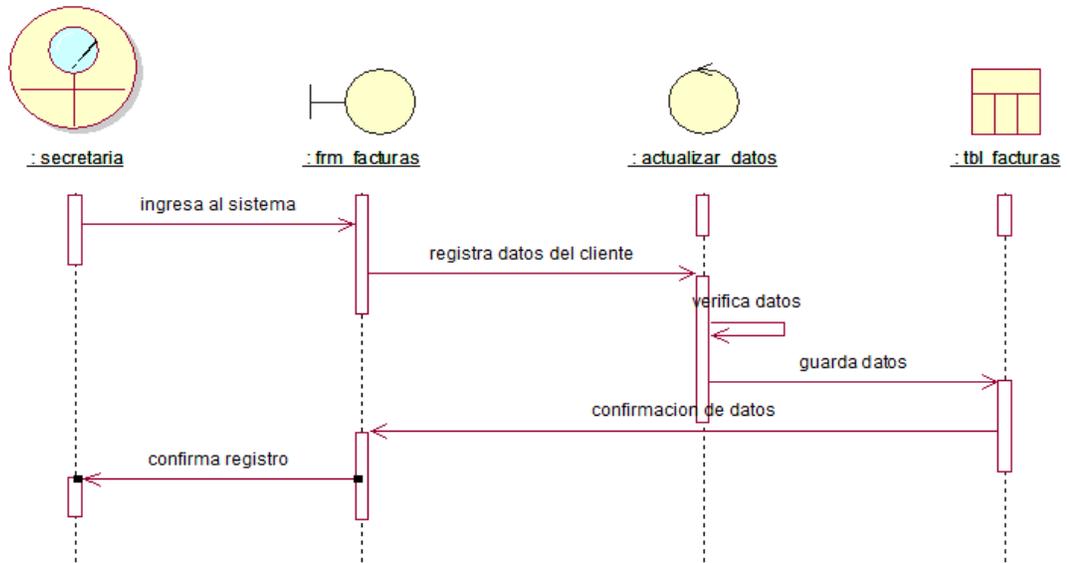


Figura 20: Factura

3.05.08 Pagar factura

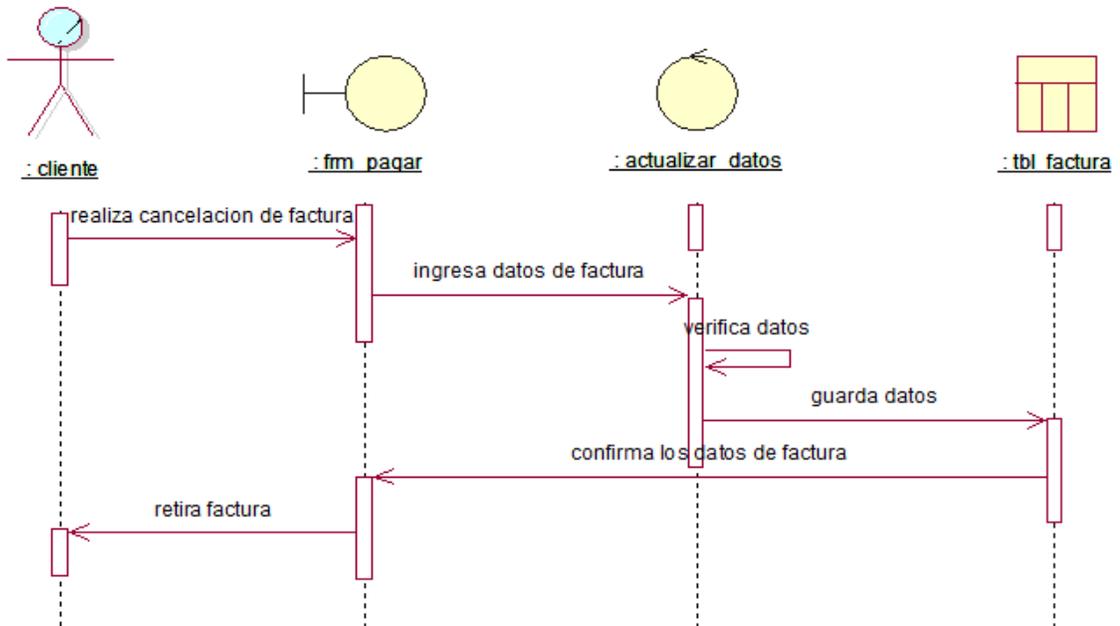


Figura 21: Pagar factura

3.05.09 Anular factura

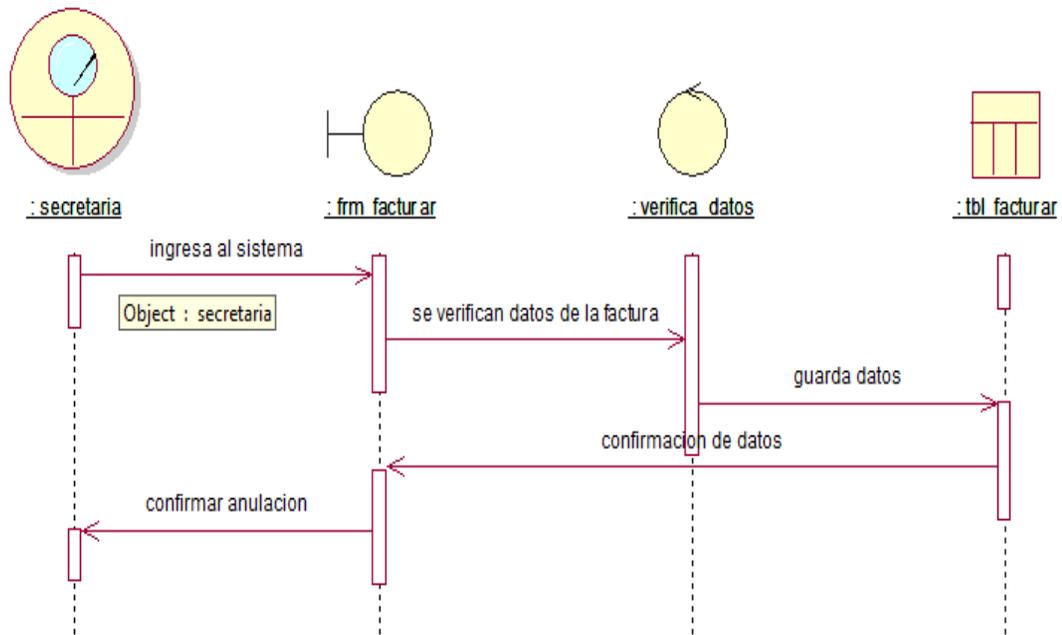


Figura 22: Anular factura

3.05 Especificación de casos de uso

Tabla 21

Especificación CU001 Registro usuarios

Caso de usos	Registro usuarios
Identificador	CU001
CURSO TIPICO DE EVENTOS	
USUARIO	SISTEMA
Ingresa al sistema	El administrador ingresa datos de empleados y proporciona un user y password
CURSOS ALTERNATIVOS	
El usuario registrado en caso de olvidar su password deberá informar al administrador.	

Tabla 22

Especificación CU002 Registro productos

Caso de usos	Registro productos
Identificador	CU002
CURSO TIPICO DE EVENTOS	
USUARIO	SISTEMA
Ingres a sistema Recibe productos	El bodeguero registrará información de mercadería en el sistema
CURSOS ALTERNATIVOS	
ninguno	

Tabla 23

Especificación CU003 Control inventario

Caso de usos	Control inventario
Identificador	CU003
CURSO TIPICO DE EVENTOS	
USUARIO	SISTEMA
Ingres a al sistema El bodeguero podrá realizar actualizaciones de información de productos ingresados	El bodeguero registrará información de mercadería en el sistema y se procederá a actualizar datos.
CURSOS ALTERNATIVOS	
ninguno	

Tabla 24

Especificación CU004 Facturación

Caso de usos	Facturación
Identificador	CU004
CURSO TIPICO DE EVENTOS	
USUARIO	SISTEMA
Ingres a datos de cliente y producto	Se ingresara la cédula de cliente en caso de no constar en el sistemas se podrá ingresar sus datos y guardar en la base de datos.
CURSOS ALTERNATIVOS	
ninguno	

Capítulo IV: Análisis de alternativas

4.01 Matriz de Análisis de Alternativas

Tabla 25

Matriz de Análisis de Alternativas

Objetivos	Impact. del propósito	Factib. Técnica	Factib. Financ.	Factib. Social	Factib. política	Total	Categoría
Sistematizar el proceso de inventario y facturación.	4	3	4	2	4	17	Alto
Mejorar el control de inventario y facturación.	4	4	4	3	2	17	Alto
Desarrollar e implementar sistema informático	4	4	4	2	2	16	Alto
El Personal será capacitado en uso de herramientas tecnológicas.	4	3	4	3	3	17	Alto
TOTAL	16	14	16	10	11	67	

Nota: Matriz de Análisis de Alternativas

4.02 Matriz de Impactos de Objetivos

Posteriormente de realizar un análisis de cada objetivo se verificó la factibilidad los mismos que interpretan la incidencia de los impactos que genera los propósitos y metas, tiene como finalidad categorizar las propuestas y posteriormente comparar los objetivos que se encuentran involucrados en el proyecto. (Ver Anexo A.01)

4.03 Estándares para el diseño de clases

UML es un lenguaje estándar que sirve para escribir los *planos del software*, puede utilizarse para visualizar, especificar, construir y documentar todos los artefactos que componen un sistema con gran cantidad de software. UML puede usarse para modelar desde sistemas de información hasta aplicaciones distribuidas basadas en Web, pasando por sistemas empotrados de tiempo real. (Ver anexo A.02)

4.04 Diagrama de Clases

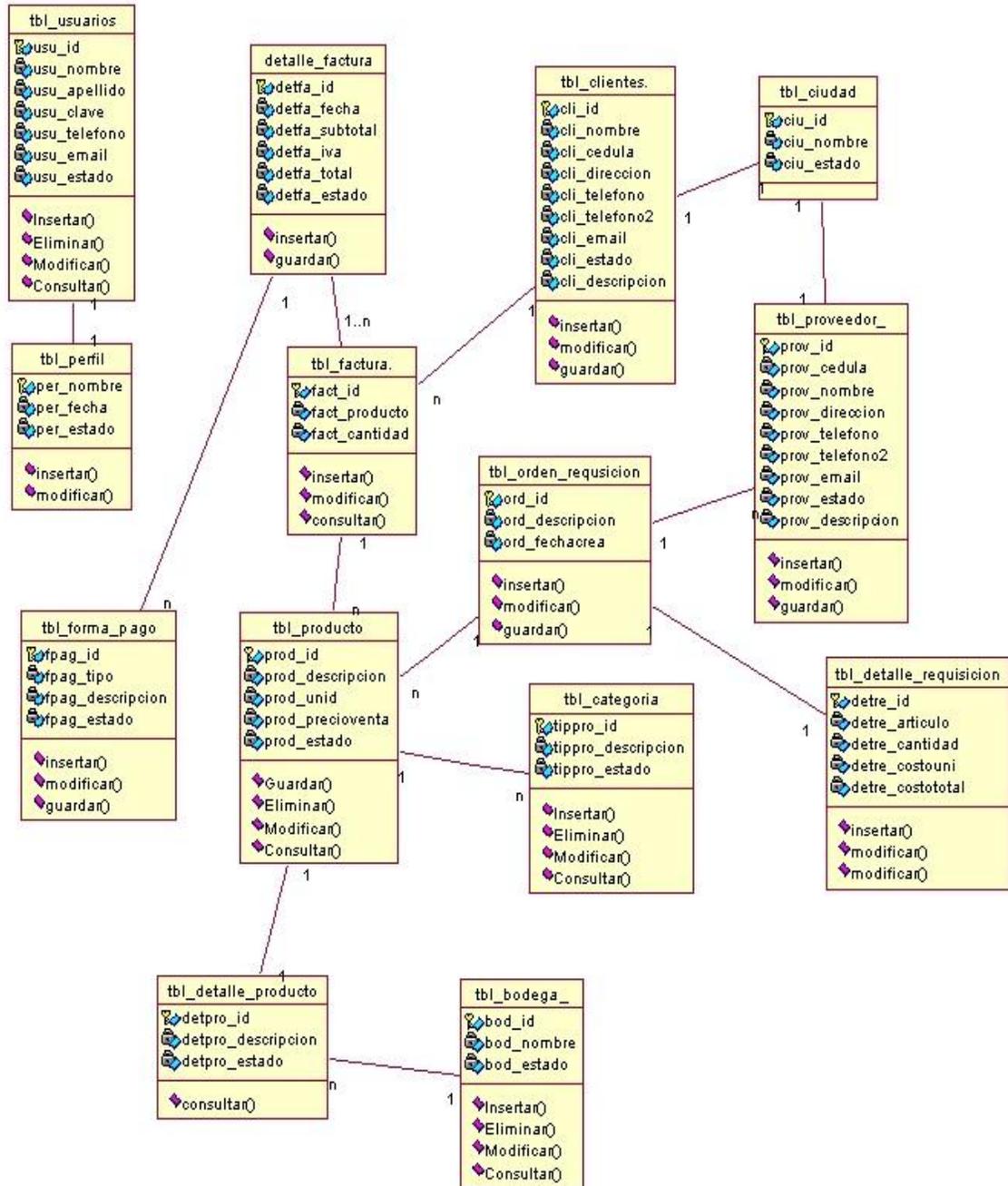


Figura 23: Diagrama de Clases: en el diagrama de Clases se representa la estructura del sistema que se va a utilizar (la base de datos del sistema con sus tablas, atributos y procesos).

4.05 Modelo lógico – físico

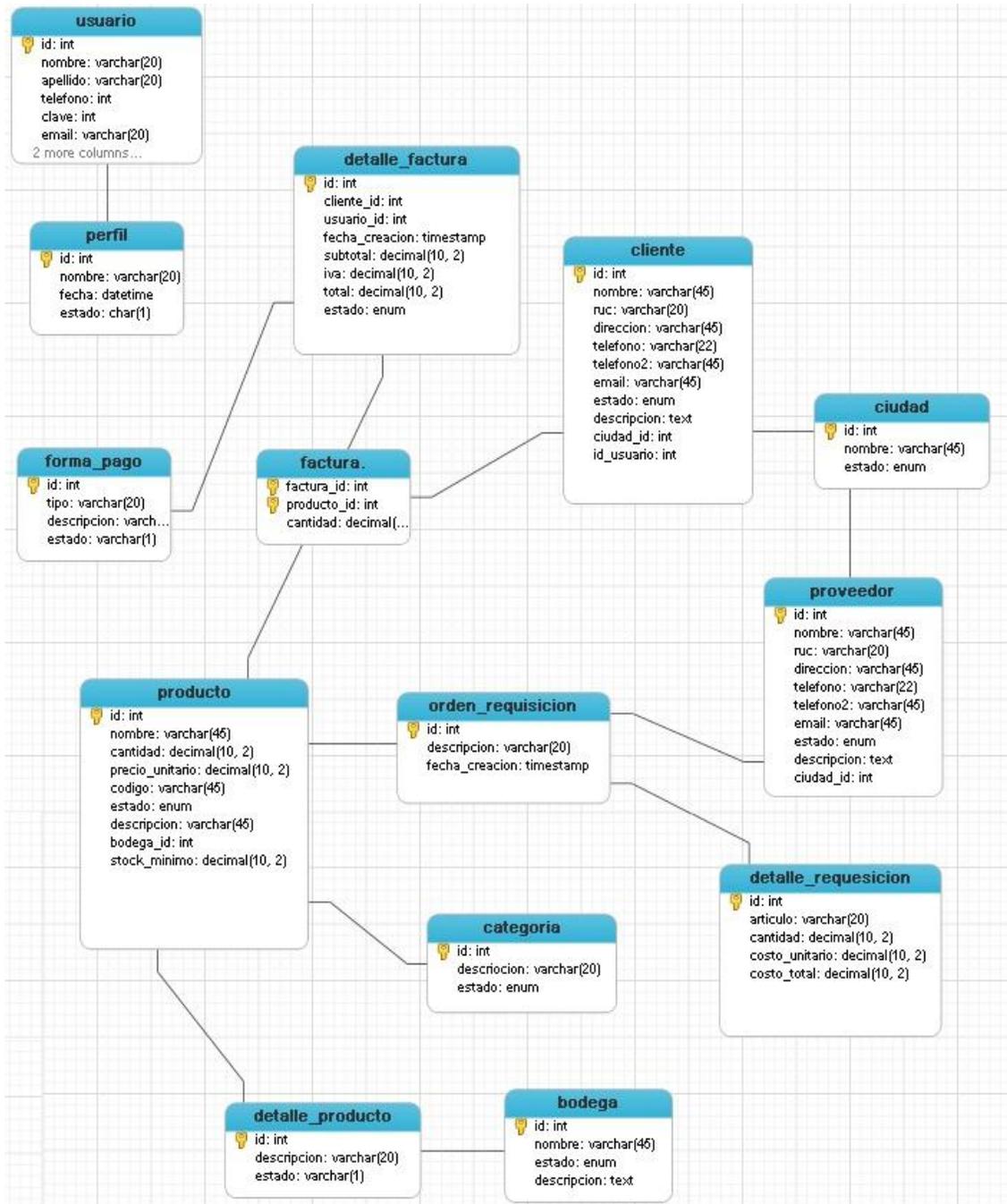


Figura 24: Modelo lógico – físico: el diagrama lógico y físico es una visión de la estructura del sistema para lograr su funcionalidad. Y visualiza la estructura de la base de datos a utilizar con sus campos y sus respectivos tipos de datos.

4.06 Diagrama de componentes.

4.06.01 Capa acceso de datos

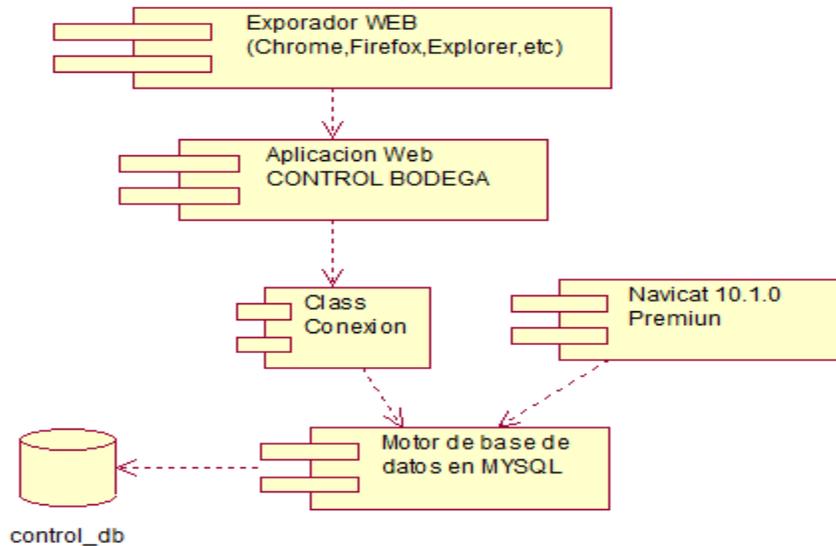


Figura 25: capa datos: en este diagrama representa los componentes de la capa Acceso a Datos

4.06.02 Capa lógica del negocio

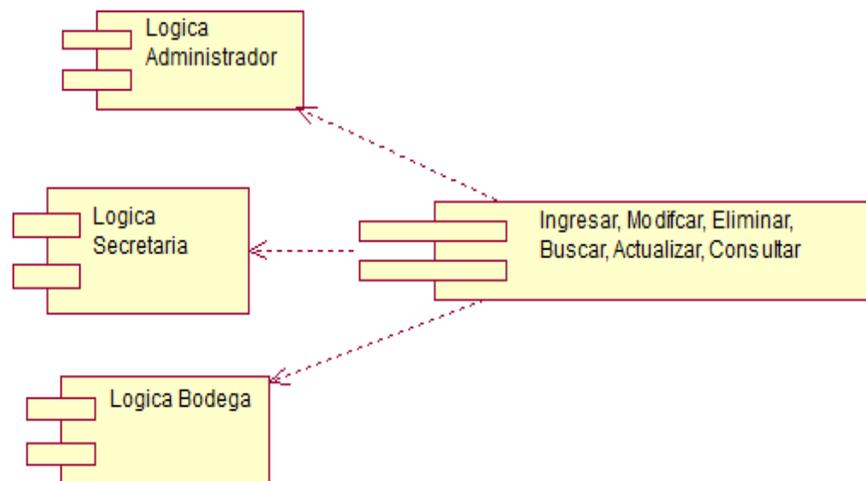


Figura 26: negocio: en este diagrama se visualiza las lógicas que se utilizarán para el sistema como es la de, administrador, secretaria y bodega.

4.06.03 Capa presentación

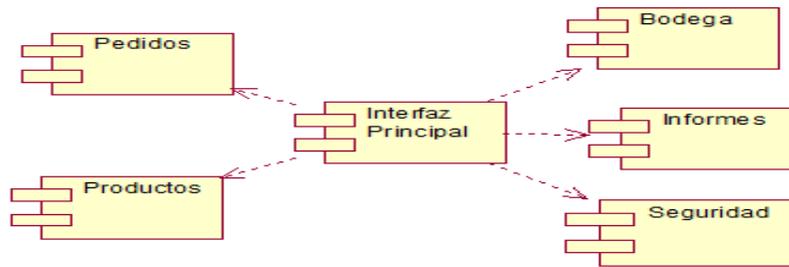


Figura 27: presentación: este diagrama representa los componentes de la capa de Presentación.

4.07 Diagrama de Estrategias.

Este Diagrama toma en cuenta el árbol de problemas y objetivos con la finalidad de mostrar las actividades que se van a realizar dentro de la solución al problema, identificando la finalidad, el propósito y componentes del mismo.

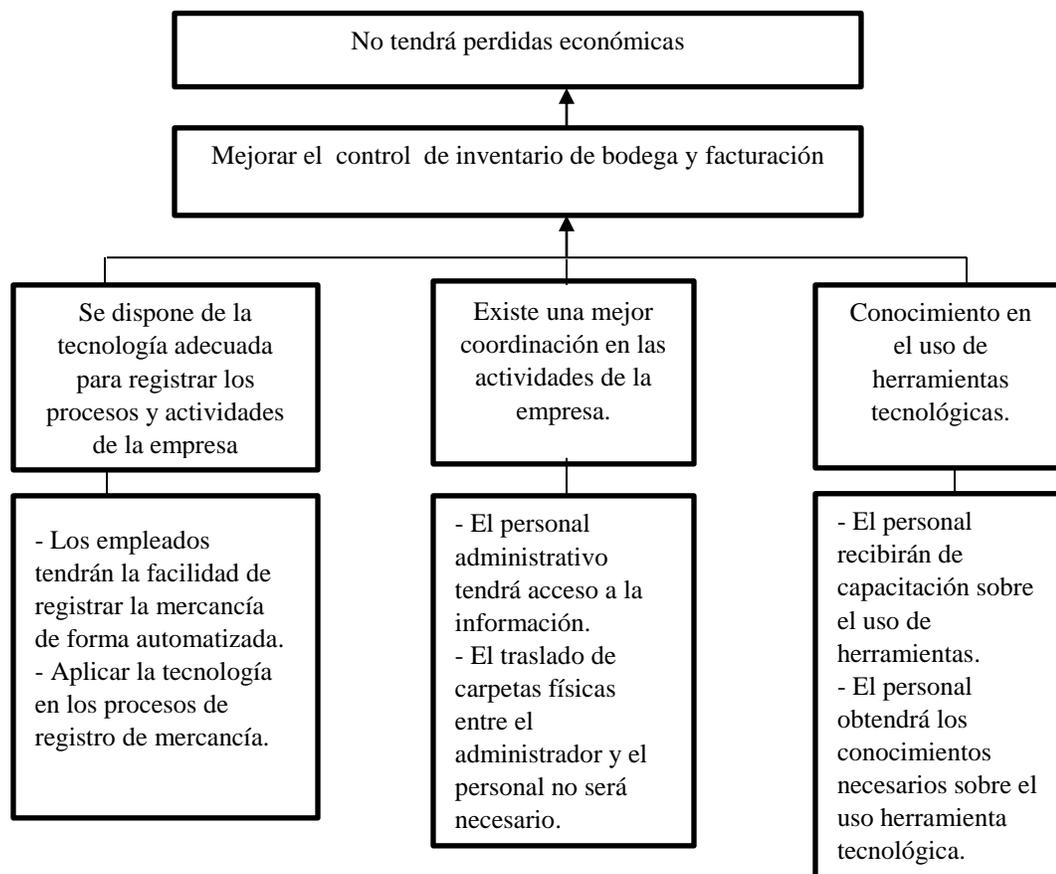


Figura 28: diagrama de estrategias: En este diagrama se detallan las actividades que se van a realizar para mantener el seguimiento y control actualizado de inventarios.

4.08 Matriz de Marco lógico.

En esta matriz permite describir los resultados anunciados en el proyecto, analizando los indicadores, medios de verificación y supuestos del proyecto, mostrando el estado inicial del problema y como se ha ido desarrollando las soluciones, además se analiza los elementos de las posibles dificultades que se pueden dar. (Ver anexo A. 03).

4.09 Vistas Arquitectónicas

4.09.01 Vista lógica

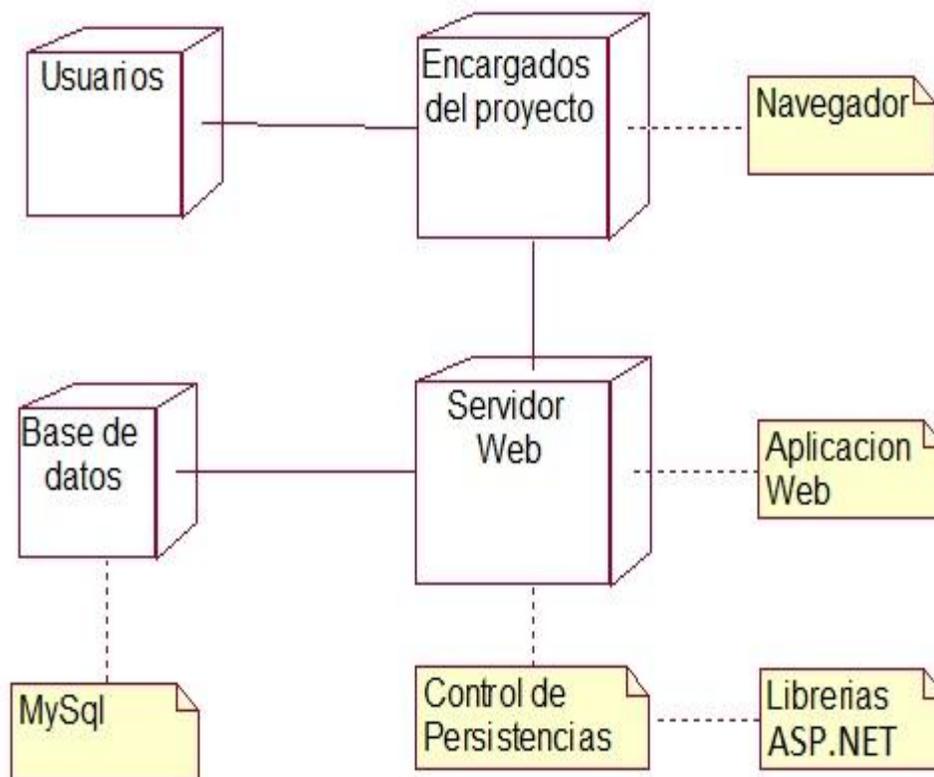


Figura 29: Vista lógica: esta vista nos sirve para soportar los requisitos funcionales del sistema, o sea, los servicios que el sistema debe proporcionar y comprende las clases, interfaces y colaboraciones.

4.09.02 Vista Física

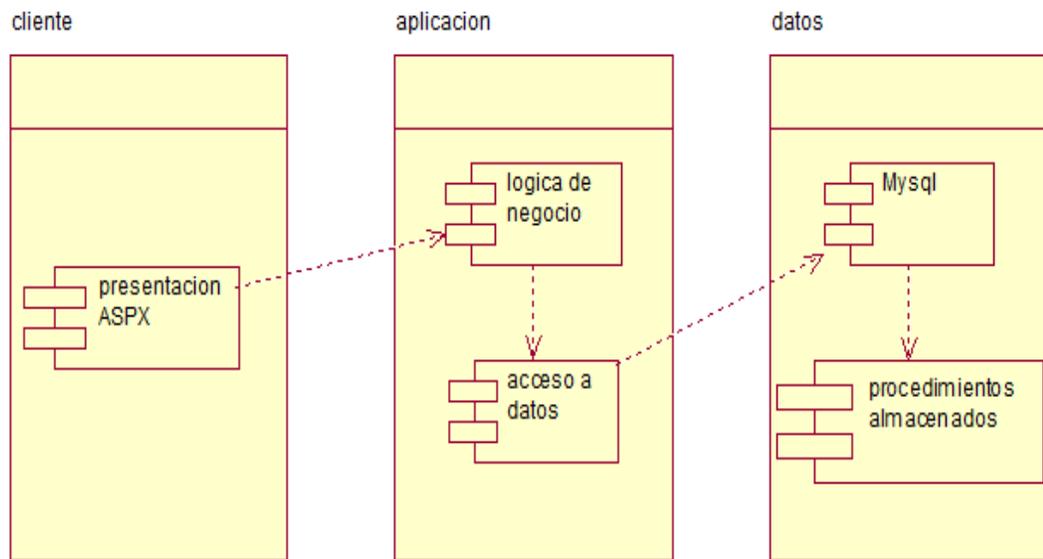


Figura 30: vista física: del diseño de la solución (diagrama de bloques)

4.09.03 Vista de Desarrollo

Capa acceso de datos

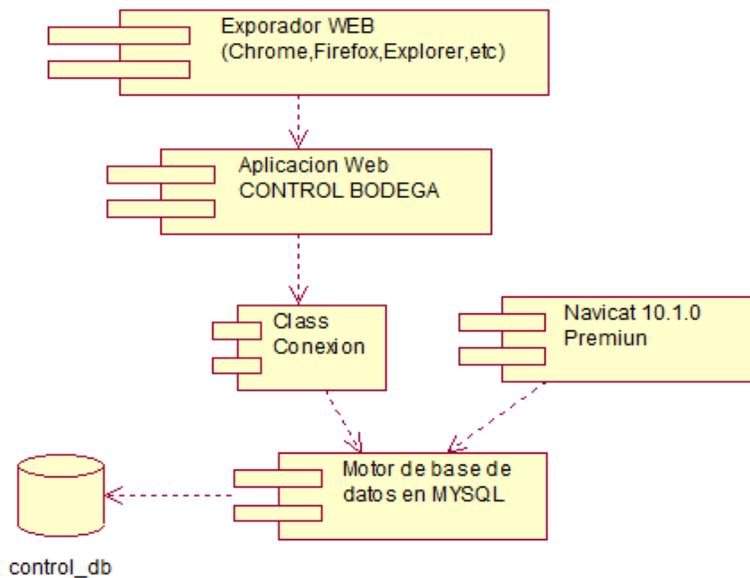


Figura 31: Capa acceso de datos

Capas lógica del negocio

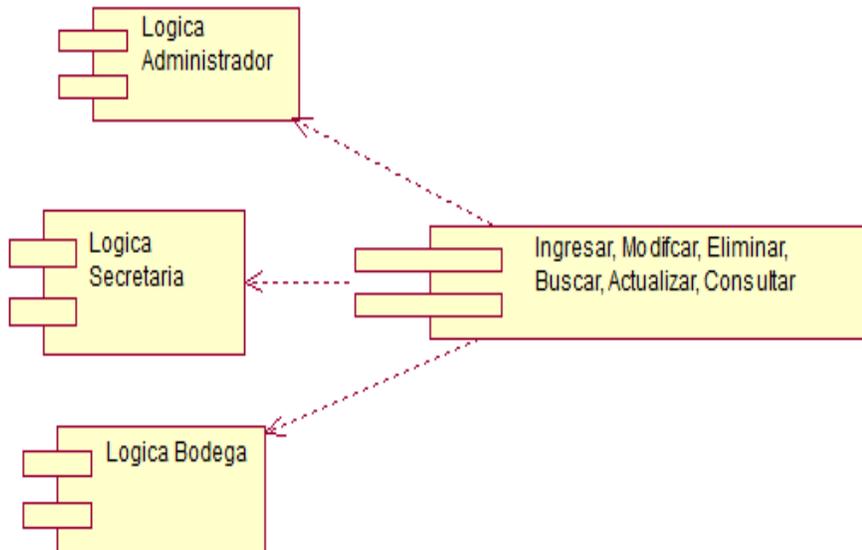


Figura 32: Capas lógicas del negocio

Capa presentación

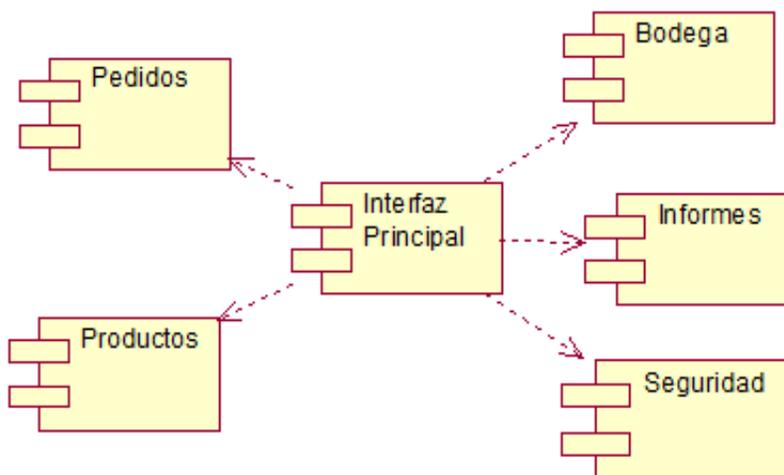


Figura 33: Capa presentación

4.09.04 Vista de Procesos

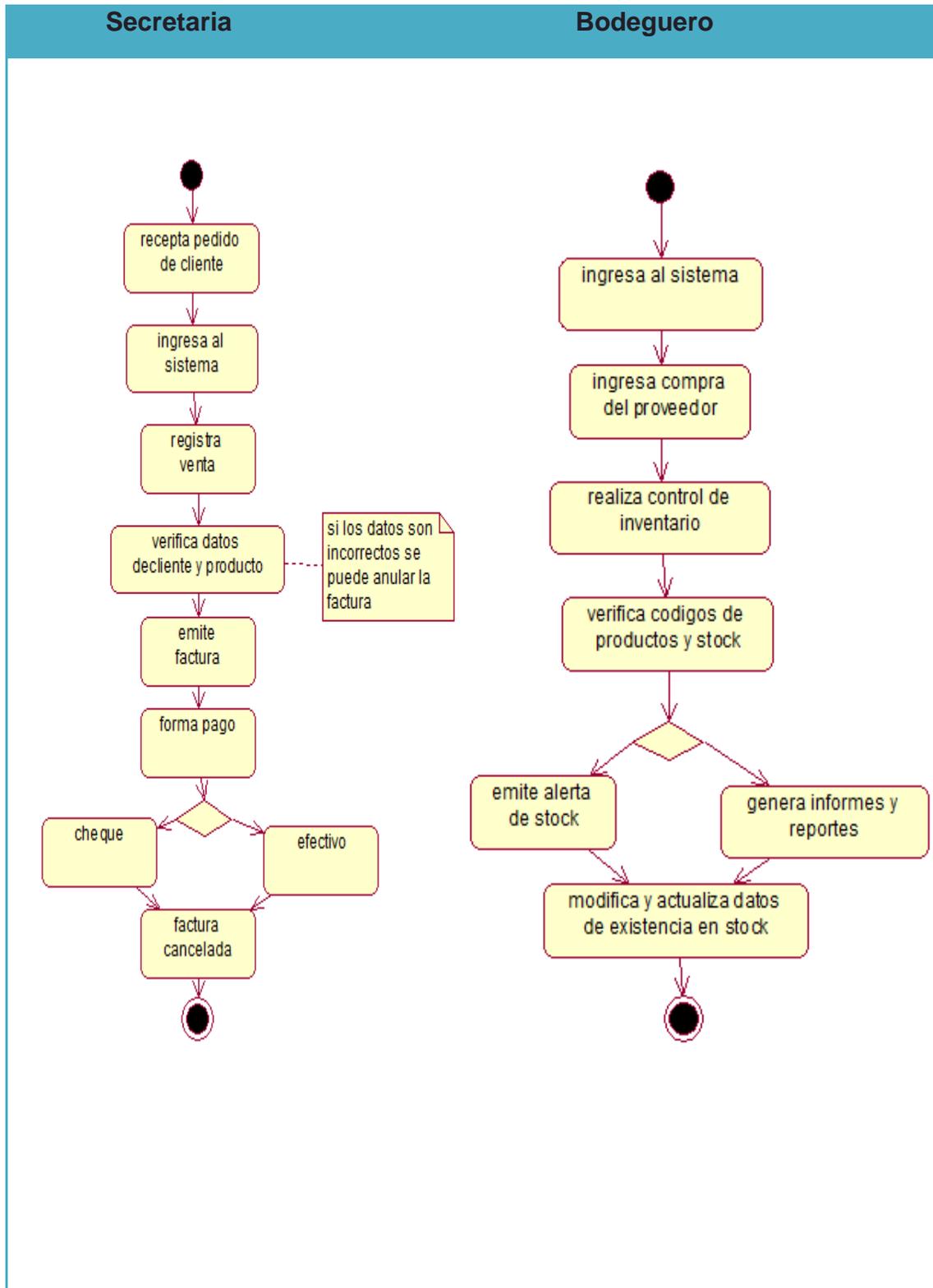


Figura 34: Vista de procesos: diseño de la solución del sistema dividido en agrupaciones

Capítulo V: Propuesta

5.01 Especificación de estándares de programación

Un estándar de programación es una forma de normalizar la programación de tal manera que al trabajar en un proyecto cualquiera, las personas involucradas en el mismo tengan acceso y comprenda el código que se generó. En otras palabras define la escritura y organización del código fuente de un programa.

Tabla 26

Descripción de los estándares de control

Tipo de control	Prefijo	Ejemplo
Boolean	bln	Bln1
Byte	Bty	bty
Currency	Cur	Cur1
Double	Dbl	Dbl1
Date	Dtm	Dtm1
Float	Sng	Sng1
Integer	Int	Inta
Object	Obl	Obla
String	Str	Stra
Variant	Vnt	vnta
Label	lbl	lblnombre
Textbox	txt	txtapellido
DataList	dtl	Dtl1
Repeater	rep	reptom
Checkbox	chk	chknom
RadioButton	rbt	Rbtgenero
RadioButtonlist	rbl	Rbl1
Image	img	imgnom

Panel	pan	pan
Table	tbl	Tbl1
Menú	Mnu	Mnumaster
Customvalidator	Cuv	Cuvdato
Image	Img	imgfoto
Fileupload	Fup	fuparchivo
Confirmbuttonextender	Cbe	cbeeliminar

5.02 Diseño de interfaces de usuario

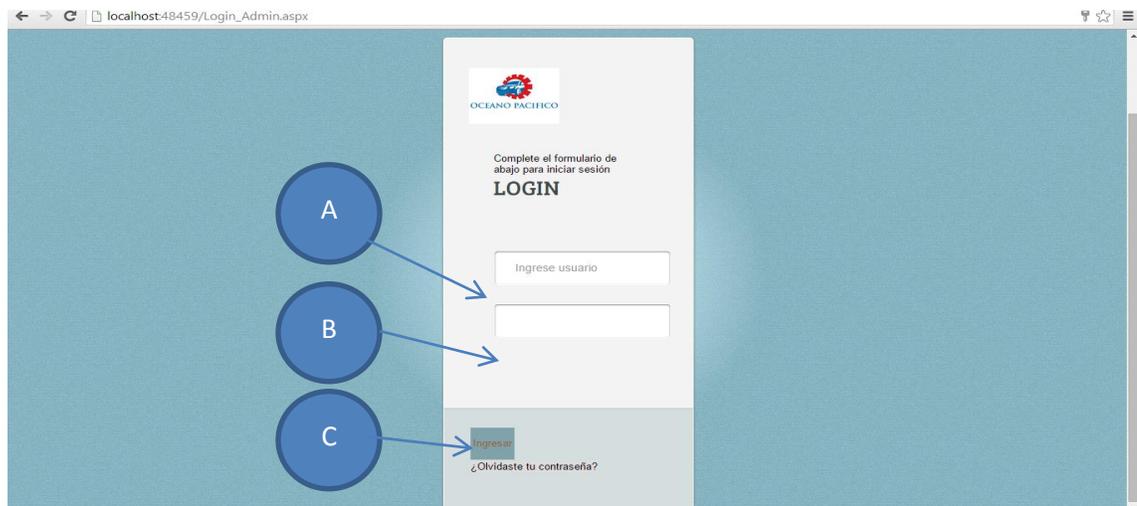


Figura 35: login: el usuario ingresa al sistema

Tabla 27

Ingreso al Sistema

Ítem	Representación	Descripción
A	Caja de Texto	Usuario
B	Caja de Texto	Contraseña
C	Botón	Ingreso al Sistema

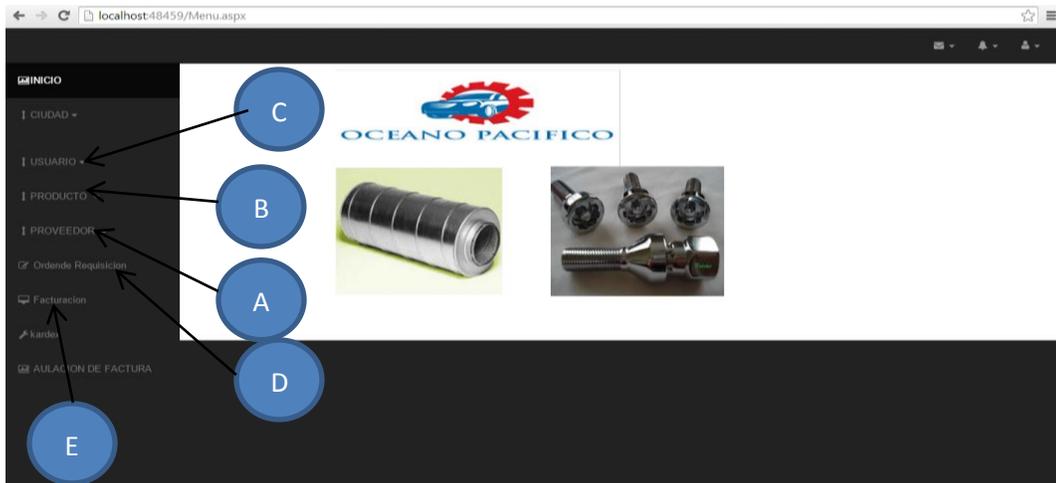


Figura 36: Menú principal: visualización del menú del sistema

Tabla 28

Menú

Ítem	Representación	Descripción
A	Botón	Ingreso al formulario proveedores
B	Botón	Ingreso al formulario inventario
C	Botón	Ingreso al formulario usuario
D	Botón	Ingreso al formulario orden de requisición
E	Botón	Ingreso al formulario facturación



Figura 37: Ingreso de proveedor: creación de nuevo proveedor

Tabla 29

Agregar Proveedor

Ítem	Representación	Descripción
A	Nuevo Proveedor	Agregar Nuevo Proveedor
B	Registros	Campos de Registros
C	Botón	Crear y Cancelar

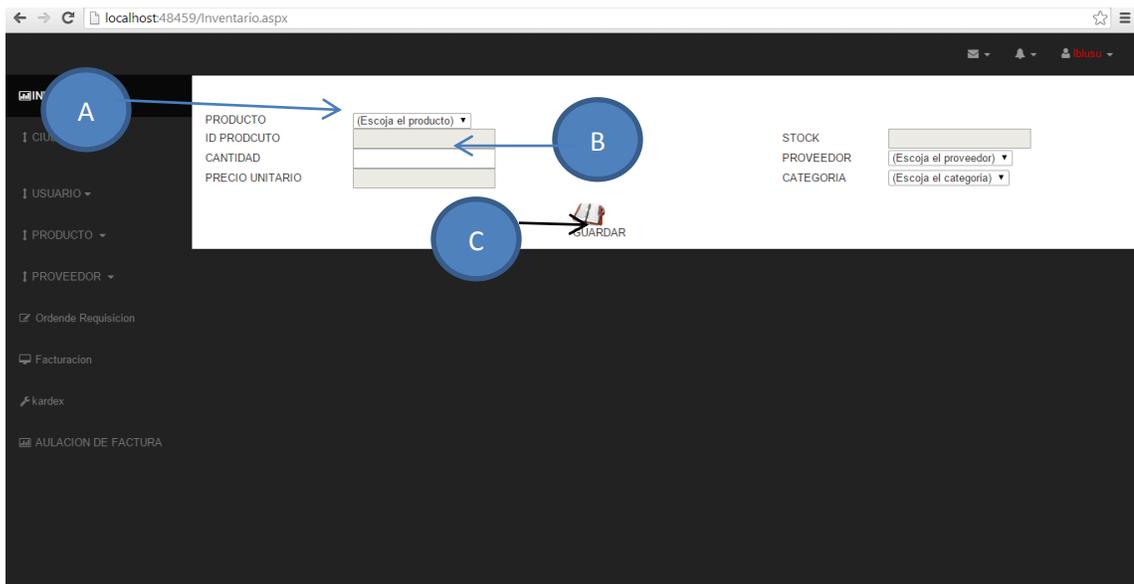


Figura 38: ingreso de producto: registro de producto nuevo

Tabla 30

Inventario

Ítem	Representación	Descripción
A	Nuevo producto	Agregar nuevo producto
B	Registros	Campos de Registros
C	Botón	Crear y Cancelar

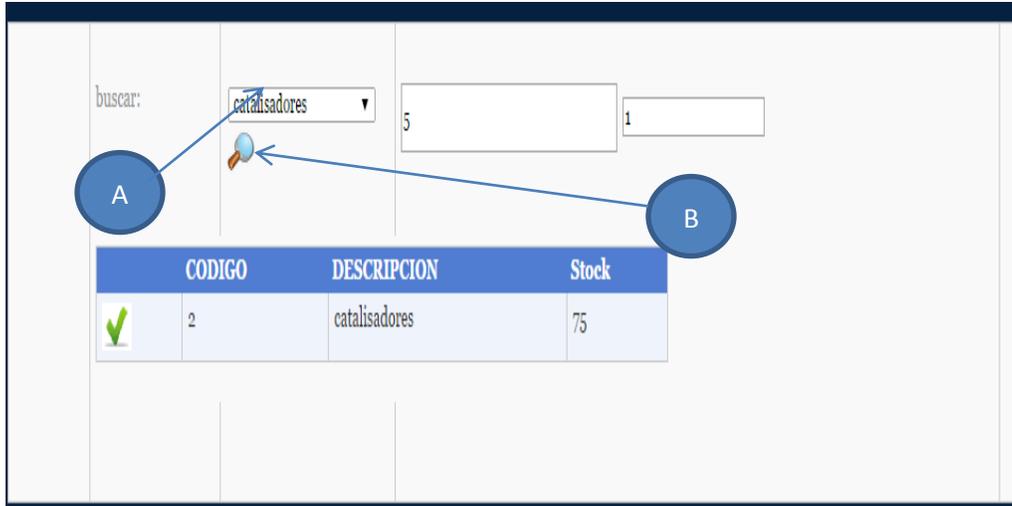


Figura 39: lista de producto: lista de productos en existencia

Tabla 31

Administrar producto

Ítem	Representación	Descripción
A	Lista de productos	Lista de producto
B	Búsqueda	Sistema de búsqueda

5.03 Especificación de pruebas de unidad

Tabla 32

Especificación PU001 login

Identificador de la Prueba	PU001
Método a Probar	login
Objetivo de la Prueba	Verificar que los usuarios ingresen correctamente sus datos en la interfaz de ingreso
Datos de Entrada	
1. User 2. Password	
Resultado Esperado	
El sistema debe emitir una alerta que faltan de llenar los campos requeridos	
Comentario	
Advierte correctamente que los campos están vacíos	

Tabla 33

Especificación PU002 validaciones

Identificador de la Prueba	PU002
Método a Probar	Validación de cedula
Objetivo de la Prueba	Evidenciar que el método validación de la cedula compruebe que el numero ingresado sea correcto
Datos de Entrada	
Numero de cedula	
Resultado Esperado	
El numero ingresado sea correcto Que posea un extensión de 10 dígitos Que no permita ingresar letras	
Comentario	
ninguno	

5.04 Especificación de pruebas de aceptación

Tabla 34

Especificación PA001 Administrador

Identificador de la prueba	PA001
Caso de uso	CU- general
Tipo de usuario	Administrador
Objetivo de la prueba	Registrar los roles de usuarios que operan el sistema
Secuencia de eventos	Ingrese datos de usuario Asignar permisos y roles Asignar login y password
Resultados esperados	Registrar tipo usuario Dar acceso a un usuario al sistema
Comentarios	Crear roles, permisos y restricciones a usuarios
Estado: Aceptado	

Tabla 35

Especificación PA002 Secretaria

Identificador de la prueba	PA002
Caso de uso	CU- general
Tipo de usuario	secretaria
Objetivo de la prueba	Emitir factura de venta
Secuencia de eventos	Ingrese datos de cliente Ingresar información de productos (cantidad, descripción y precio)
Resultados esperados	Emitir factura de venta para cancelación de cliente
Comentarios	Debe ser registrado datos requeridos para emitir factura de venta
Estado: Aceptado	

Tabla 36

Especificación PA003 Bodeguero

Identificador de la prueba	PA003
Caso de uso	CU- general
Tipo de usuario	Bodeguero
Objetivo de la prueba	Ingresar y modificar productos en inventario
Secuencia de eventos	Ingresar/modificar datos de producto Emitir informes y/o reportes de stock
Resultados esperados	Mantener una información actualizada de productos existentes en inventario
Comentarios	Conservar ordenadamente datos de todos los productos en inventario
Estado: Aceptado	

5.05 Especificación de pruebas de carga

Se carga a la base de datos con el objetivo de verificar que no exista una sobrecarga en la información y exista persistencia en la base de datos.

Tabla 37

Especificación PC001 Carga de usuarios

Identificador de la prueba	PC001
Tipo de prueba	Carga excesiva de usuarios
Objetivo de la prueba	ingresar 2 usuarios conectados al sistema
Descripción	Se realiza el ingreso de dos usuarios conectados a la vez al sistema
Resultados esperados	El sistema soporte el tráfico de usuarios cuando están conectados 2 usuarios
Comentarios	La prueba fue un éxito se comprobó la estabilidad del sistema con dos o más usuarios conectados

5.06 Configuración del Ambiente mínima/ideal

Requerimientos de Hardware

Equipo Servidor

- Procesador: Intel Core 2 Duo de 2 Ghz o 3 Ghz
- Memoria Ram 4 GB Disco duro con 100 Gb libres para datos.

Equipo Cliente

- Procesador: Intel Core 2 Duo de 2 Ghz
- Memoria Ram 4 Gb
- Disco Duro: 520 Gb

Requerimientos de Software

Visual Studio 2012

Mysql

Equipo Cliente

Sistema Operativo

Microsoft Windows 7 o superior

Browser

- Internet Explorer
- Mozilla Firefox
- Google Chrome

Capítulo VI: Aspectos Administrativos

6.01. Recursos

Tabla 38

Recursos

HUMANO	NOMBRE	ACTIVIDAD	RESPONSABILIDAD
Tutor	Ing. Patricia Garzón	Director del Proyecto	Guía e instruye al desarrollador del proyecto
Administrador	Sr. Galo Cruz	Autoriza la elaboración del Sistema	Toma decisiones
Empleados	Empleados	Proporciona Información	Atiende a los clientes
Recursos Físicos	Equipo de Computo		
	Impresora		
Recursos Económicos	Alimentos-Refrigerios		
	Transporte		
Recursos Renovables	Hojas		
	Energía		
	Anillados		
Recurso Didáctico	Internet		
Otros Recursos	Seminarios		
	Tutorías		

Nota: Determina y analiza los recursos necesarios.

6.02. Presupuesto

Tabla 39

Presupuesto del proyecto

CANT.	DESCRIPCIÓN	VALOR UNITARIO	VALOR TOTAL
540	Impresiones Color	\$ 0,25	\$ 135,00
540	Impresiones B/N	\$ 0,10	\$ 54,00
1	Materiales de Oficina	\$ 20,00	\$ 20,00
320	Transporte	\$ 0,25	\$ 80,00
2	Anillados	\$ 2,50	\$ 5,00
6	Internet	\$ 21,00	\$ 126,00
1	Seminarios	\$ 520,00	\$ 520,00
1	Tutorías	\$ 200,00	\$ 200,00
		TOTAL	\$ 1.140,00

Nota: determina el presupuesto para el proyecto

6.03. Cronograma

Es una descripción específica de las actividades y del tiempo que se va a emplear para la ejecución del proyecto. Se debe organizar el trabajo en fechas probables, para saber cuánto tiempo requerirá elaborar el trabajo definitivo. Para su presentación se utilizan generalmente diagramas, lo que permite visualizar mejor el tiempo de cada actividad, y sobre todo en aquellos casos en que hay varias actividades en un mismo tiempo.

(Ver anexo A.04).

Capítulo VII: Conclusiones y Recomendaciones

7.01 Conclusiones

- Con la implementación del sistema lograremos la optimización en el control de inventarios y facturación a través de los módulos con los que cuenta el sistema los cuales permiten el funcionamiento de la misma.
- Los administradores tendrán la opción de realizar la búsqueda de la información que genera el sistema y agiliza los procesos operacionales que se realizan en la actualidad, aportando nuevas estrategias en el desarrollo de las actividades.
- El sistema permitirá un control de cada cliente, producto y bodega para facilitar la información con eficiencia y rapidez.
- Esta Empresa contará con el aporte tecnológico beneficiando a sus clientes, empleados y a las personas aledañas del sector.

7.02 Recomendaciones

- El personal y los administradores tiene la obligación de revisar los manuales para el buen desempeño del sistema.
- Los errores que se presenten al momento de la manipulación del sistema se deberá reportar al personal técnico para la solución inmediata.

- Es necesario que los equipos tecnológicos que contarán el personal debe estar actualizados con la última tecnología para el desempeño de sus funciones.
- El acceso a los mantenimientos del sistema deben ser exclusivos del administrador, con el propósito de mantener la persistencia y la integridad de los datos, permitiendo que la seguridad del software se encuentre en un estado óptimo.



ANEXOS



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE ANALISIS DE SISTEMAS

CONTROL DE INVENTARIO DE BODEGA Y FACTURACION MEDIANTE UNA
APLICACIÓN WEB PARA LA EMPRESA OCEANO PACIFICO

ANEXOS

Autor: Alex Darío Cuichán Apunte.

Tutor: Lic. Patricia Garzón Msc.

Quito, Octubre 2014

A.01 Matriz de Impacto de Objetivos

Anexo 1

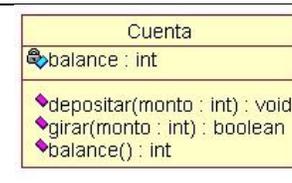
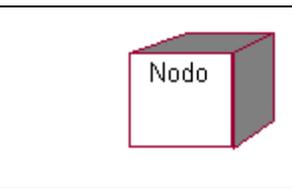
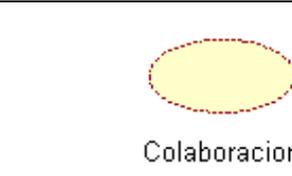
Matriz de impacto de objetivos

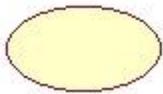
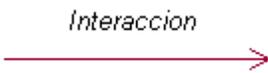
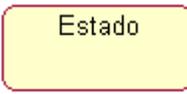
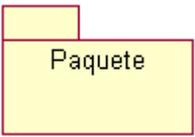
Objetivos	Factibilidad de Lograse	Impacto en Género	Impacto Ambiental	Relevancia	Sostenibilidad	T O T A L
Sistematizar el proceso de inventario y facturación	- Se cuenta con el apoyo total por parte de la administración	- Ninguno el género es indistinto	- Contribuye a proteger el entorno físico.	- Responde a las expectativas de los beneficiarios	- Fortalece la participación de los empleados.	
Mejorar el control de inventario y facturación.	- Se cuenta con la infraestructura adecuada	- Ninguno el género es indistinto	- Protege el uso de los recursos.	- Es una prioridad sentida por los beneficiarios	- Fortalece la Organización local	
Desarrollar e implementar sistema informático.	- Cuenta con financiamiento por parte de la empresa.	- Ninguno el género es indistinto	- Favorece al entorno ambiental.	- Es una prioridad para la gerencia	- la Empresa respalda el financiamiento del proyecto a futuro	
El personal será capacitado en uso del sistema informático.	- Mejora el entorno social con los empleados.	- Ninguno el género es indistinto	- Mejora el entorno social.	- Se utilizará la infraestructura existente	- Los usuarios están en posibilidad de adoptar el sistema.	
Total	19 puntos	0 puntos	20 puntos	16 puntos	20 puntos	75

A.02 Estándares para el diseño de clases

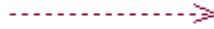
Anexo 2

Estándares para el diseño de clases

Clase	 <p>tbl_producto (from diagrama_de_clases)</p> <p>prod_id prod_descripcion prod_unid prod_precioventa prod_estado</p> <p>Guardar() Eliminar() Modificar() Consultar()</p>	Se define los atributos y los métodos de una serie de objetos. Todos los objetos de esta clase. Las clases implementan una o más interfaces.
Interfaz	 <p>Interfaz</p>	Permite emplear un círculo para representar las interfaces, aunque lo más normal es emplear la clase con el nombre en cursiva.
Atributos	 <p>Cuenta</p> <p>balance : int</p> <p>depositar(monto : int) : void girar(monto : int) : boolean balance() : int</p>	Los atributos o características de una Clase pueden ser de tres tipos, public, private, protected
Atributo Public		Indica que el atributo será visible tanto dentro como fuera de la clase.
Atributo privado		Indica que el atributo solo será accesible desde dentro de la clase solo con sus métodos
Atributo Protected		Indica que el atributo no será accesible desde la fuera de la clase, pero si podrá ser accedido por métodos de la clase además de las subclases que se deriven.
Nodo	 <p>Nodo</p>	Elemento físico que existe en tiempo de ejecución y representa un recurso computacional con capacidad de procesar.
Colaboración	 <p>Colaboracion</p>	Define una interacción entre elementos que cooperan para proporcionar un comportamiento mayor

Casos de uso	 Caso de uso	Es la descripción de un conjunto de acciones que un sistema ejecuta y que produce un determinado resultado que es de interés para un actor particular.
Diagrama de Componentes	 Comp	Se define una parte física y reemplazable de un sistema que conforma con un conjunto de interfaces y proporciona la implementación de dicho conjunto.
Interacción	 Interaccion	Comprende un conjunto de mensajes que se intercambian entre un conjunto de objetos, para cumplir un objetivo específico.
Estado	 Estado	Especifica la secuencia de estados por los que pasa un objeto o una interacción, en respuesta a eventos.
Paquete	 Paquete	Se define para organizar otros elementos en grupos.
Nota	 Nota	Se define partes explicativa de UML, que puede describir textualmente cualquier aspecto del modelo.
Asociación	 * 0..1	Se define una relación estructural que resume un conjunto de enlaces que son conexiones entre objetos.
Generalización	 	Es una relación en la que el elemento generalizado puede ser distribuido por cualquiera de los elementos hijos, ya que comparten su estructura y comportamiento.

Dependencia



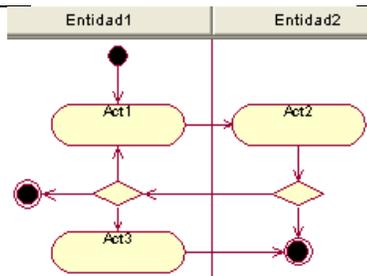
Es una relación entre dos elementos, tal que un cambio en uno puede afectar al otro.

Realización



Es una relación que implica que la parte realizada cumple con una serie de especificaciones propuestas por la clase realizada.

Actividades



Se define un tipo especial de diagrama de estados que muestra el flujo de actividades dentro de un sistema.

Fuente: <http://www.monografias.com/trabajos28/proyecto-uml/proyecto-uml.shtml>

A.03 Matriz de Marco lógico.

Anexo 3

Matriz de marco lógico

RESUMEN NARRATIVO	INDICADORES	MEDIOS DE VERIFICACION	SUPUESTOS
FINALIDAD			
Sistematización del proceso de inventario y facturación.	-En el tiempo de uso del sistema se espera que incremente un 60% la satisfacción de las nuevas estrategias de registro y control.	Sera controlado en reportes	El administrador y el personal de la empresa no tendrán dificultad en la generación de informes , inventarios y facturación
PROPÓSITO DEL PROYECTO			
Mejorar la calidad de control actualizado de registro de productos, inventarios, facturación	- Aumento de un 70% en el manejo eficaz en los procesos y actividades de la empresa.	Registros de actividades realizadas.	- Se logra optimizar el tiempo de actividades y procesos de gestión y registros.
COMPONENTES DEL PROYECTO			
-Se dispone de la tecnología adecuada para registrar la información de los productos.	- El acceso a la información aumentara 90% con la implementación del sistema tecnológico.		Los Administradores y el personal asistente optaron con nuevos conocimientos y están dispuestos a utilizar la tecnología mediante la cual van a realizar sus labores.
- Existe una mejor coordinación en el	-Aumento de un 95% en la coordinación de los procesos y actividades de la empresa.		

personal.

ACTIVIDADES DEL PROYECTO

-desarrollar un sistema que mejore el proceso.

-Agilizar el proceso de registro de productos y facturación.

-Los empleados podrán tener acceso a la información desde cualquier lugar.

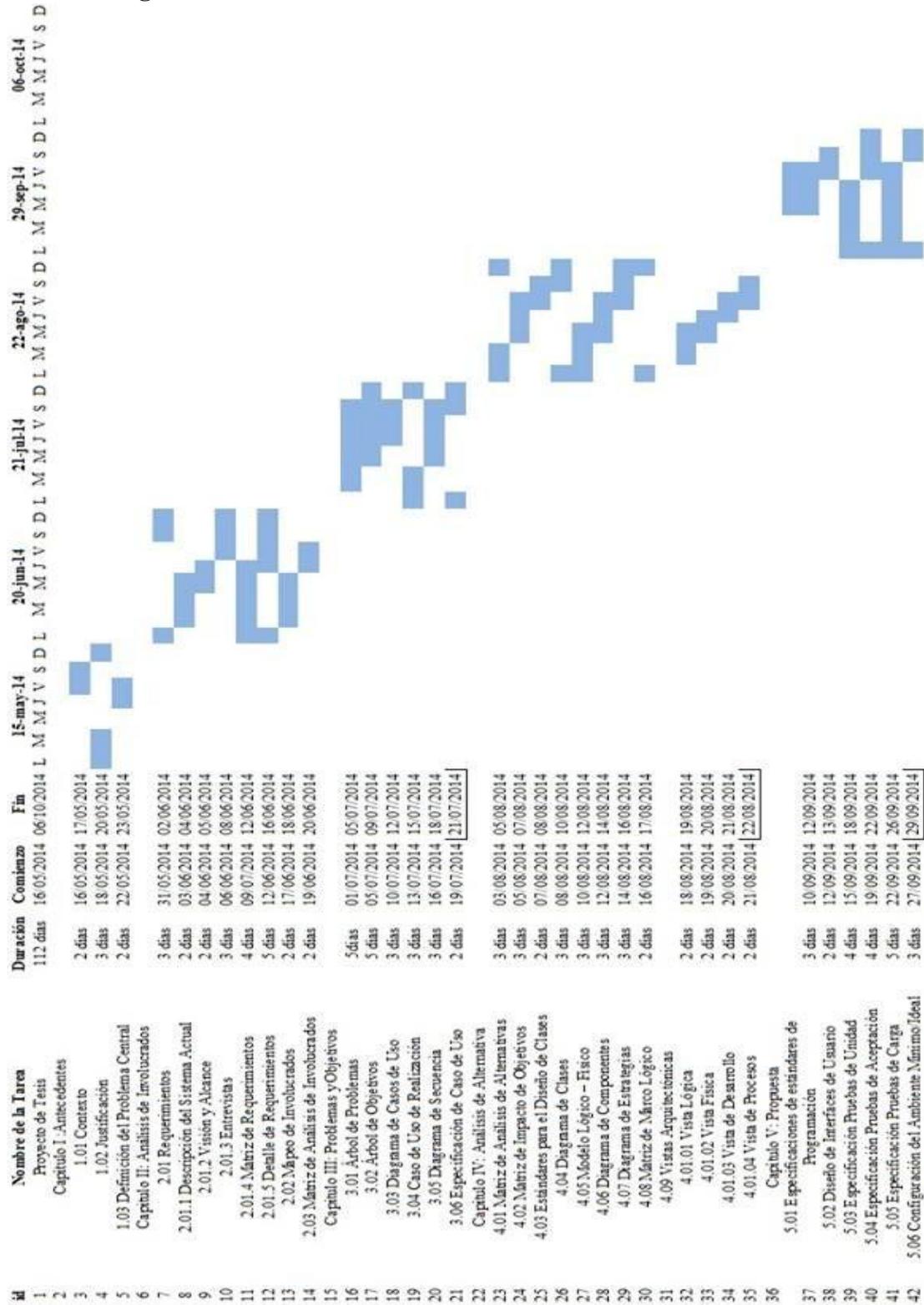
-El personal obtendrá los conocimientos necesarios sobre el uso de la herramienta.

-La implementación del proyecto cuenta con todo el respaldo necesario.

Las necesidades que requiere el desarrollador del sistema los requerimientos se basan en entorno y en función mediante el cual el proyecto se encuentra en la fase de desarrollo

Conjeturar que el personal encargado se encuentra 100% capacitado para manipular el nuevo sistema y las nuevas estrategias de organización en el control de inventario de bodega y facturación.

A.04 Cronograma



Anexo 4: Cronograma de actividades



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE ANALISIS DE SISTEMAS

CONTROL DE INVENTARIO DE BODEGA Y FACTURACIÓN MEDIANTE UNA
APLICACIÓN WEB PARA LA EMPRESA OCÉANO PACÍFICO

MANUAL DE USUARIO

Autor: Alex Darío Cuichán Apunte.

Tutor: Lic. Patricia Garzón Msc.

Quito, Octubre 2014

INDICE MANUAL DE USUARIO

Titulo	pagina
Figura 1: Ingreso Al Sistema:	70
Figura 2: Administrador	70
Figura 3: Crear nuevo usuario.....	71
Figura 4: Secretaria	71
Figura 5: Crear Proveedor.....	72
Figura 6: Bodeguero:	72
Figura 7: Nuevo Producto	73
Figura 8: crear orden requisición	73
Figura 9: crear factura:	74
Figura 10: lista de productos:.....	74
Figura 11: Kardex	75

A.06 Manual de Usuario

Login

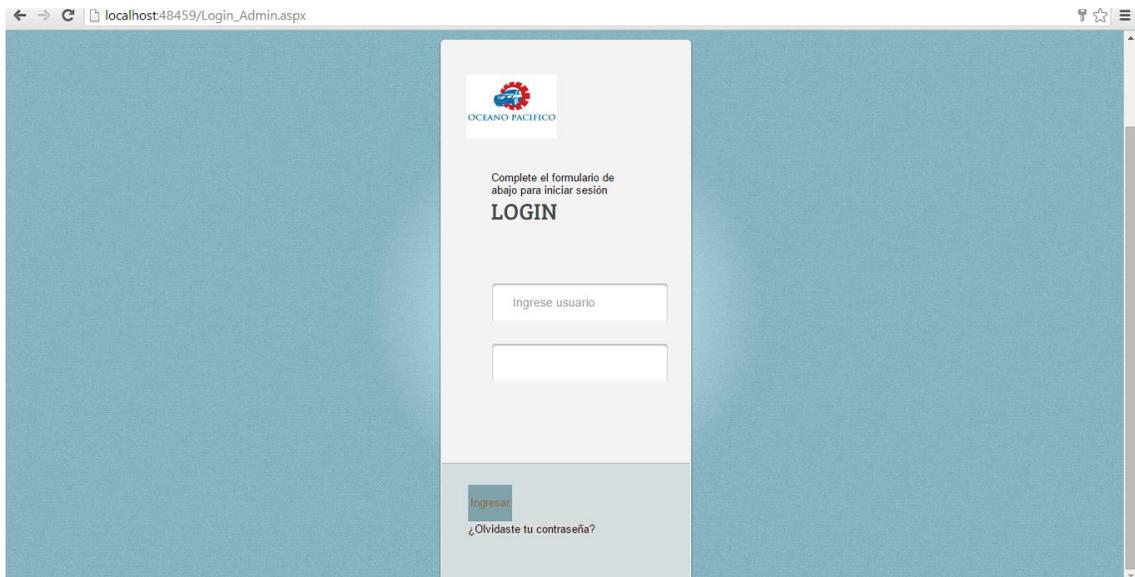


Figura 1: Ingreso Al Sistema: El cliente deberá ingresar usuario y contraseña para ingresar al sistema. Luego de haber ingresado sus datos deberá dar clic en ingresar .

Administrador



Figura 2: Administrador: El administrador tendrá el acceso a todas las ventanas del sistema.

Ingreso de nuevo usuario

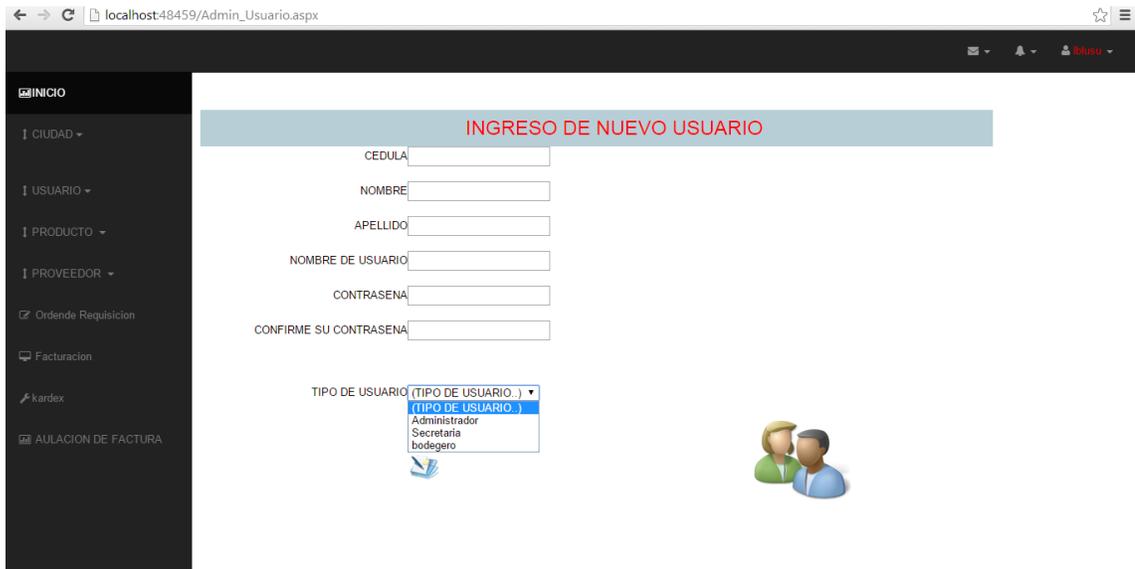
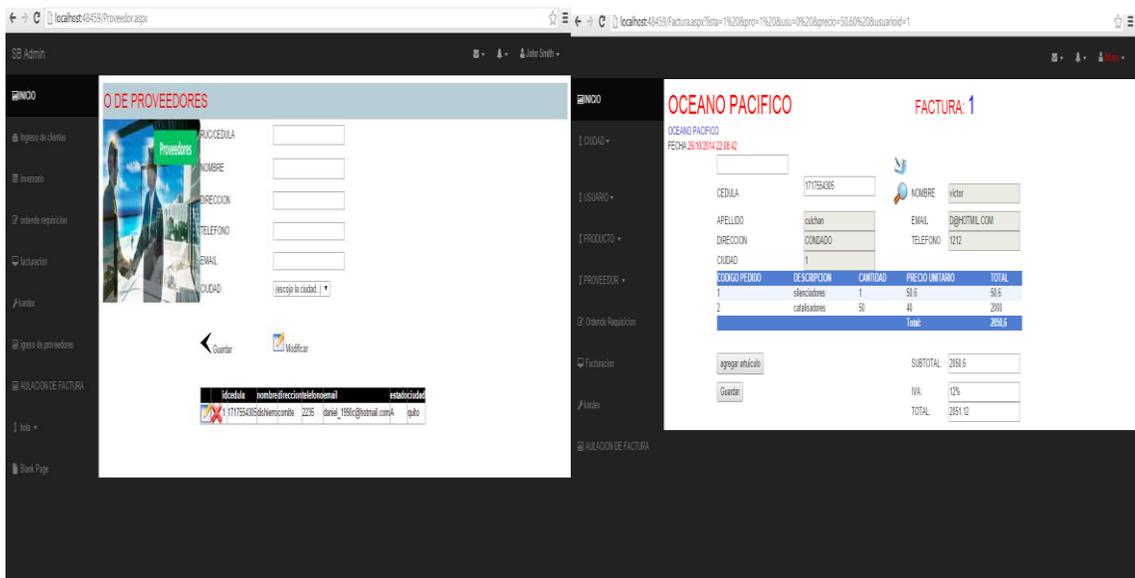


Figura 3: Crear nuevo usuario: se crea nuevo empleado de acuerdo a su rol que desempeña en la empresa

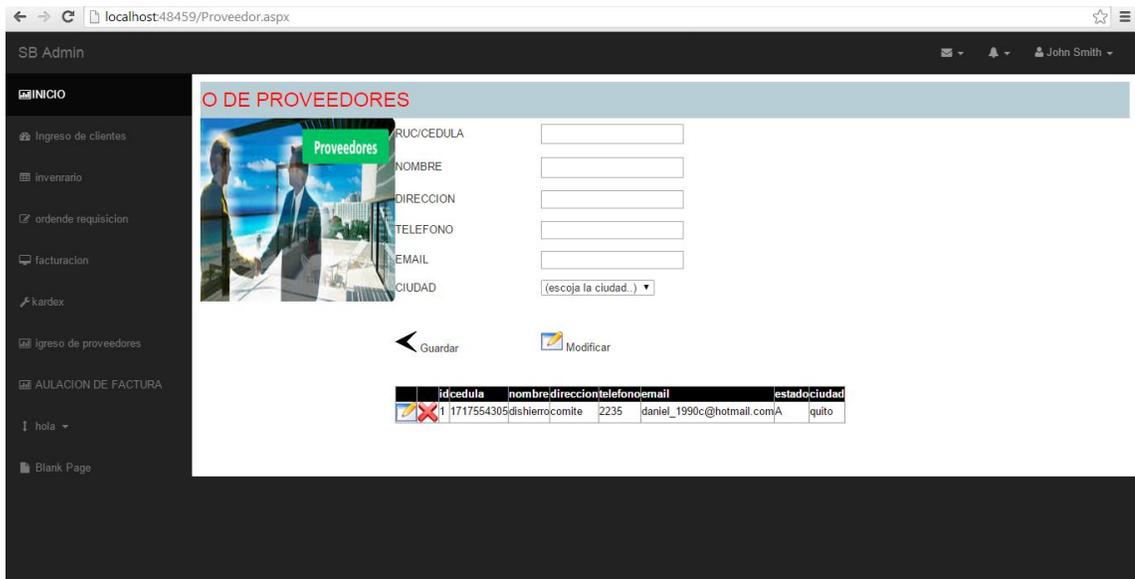
Secretaria



ORDENO PEDIDO	DESCRIPCION	CANTIDAD	PRECIO UNITARIO	TOTAL
1	alicates	1	50.6	50.6
2	catifes	50	40	2000
Total:				2050.6

Figura 4: Secretaria: se encarga de ingreso proveedor y facturación

Ingreso de proveedores



idcedula	nombre	direccion	telefono	email	estado	ciudad
1	1717554305	dishierocomite	2235	daniel_1990@hotmail.com	A	quito

Figura 5: Crear Proveedor: En esta ventana se realizará la creación de proveedor llenando los respectivos registros (nombre, ruc, dirección, teléfono, teléfono2, email, estado, descripción, ciudad).

Bodeguero

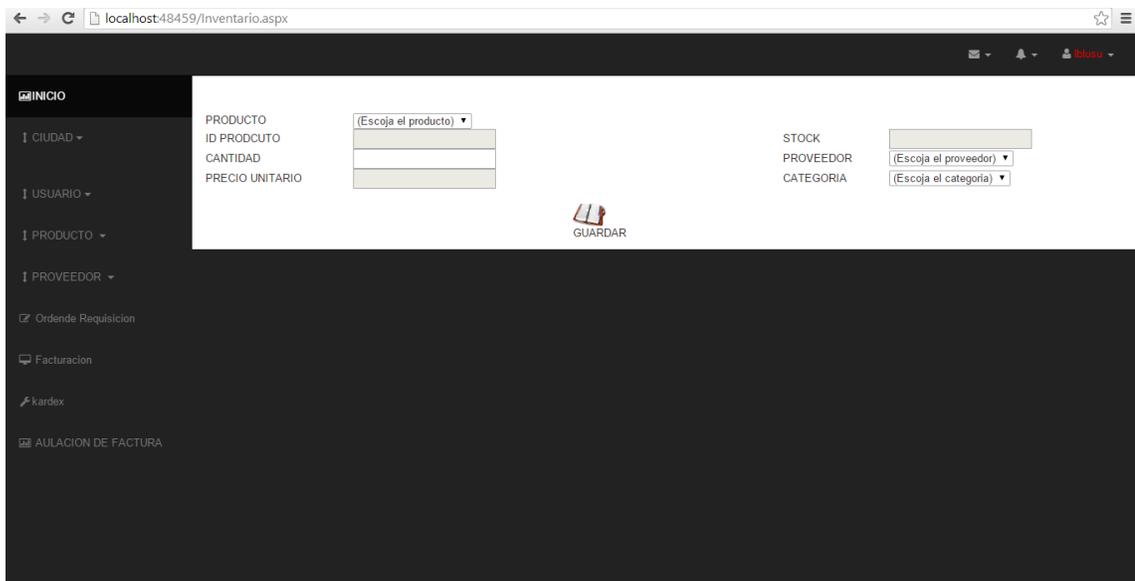


Figura 6: Bodeguero: el bodeguero podrá ver opciones de su cargo e ingresar producto nuevo.

Ingreso de producto para el inventario

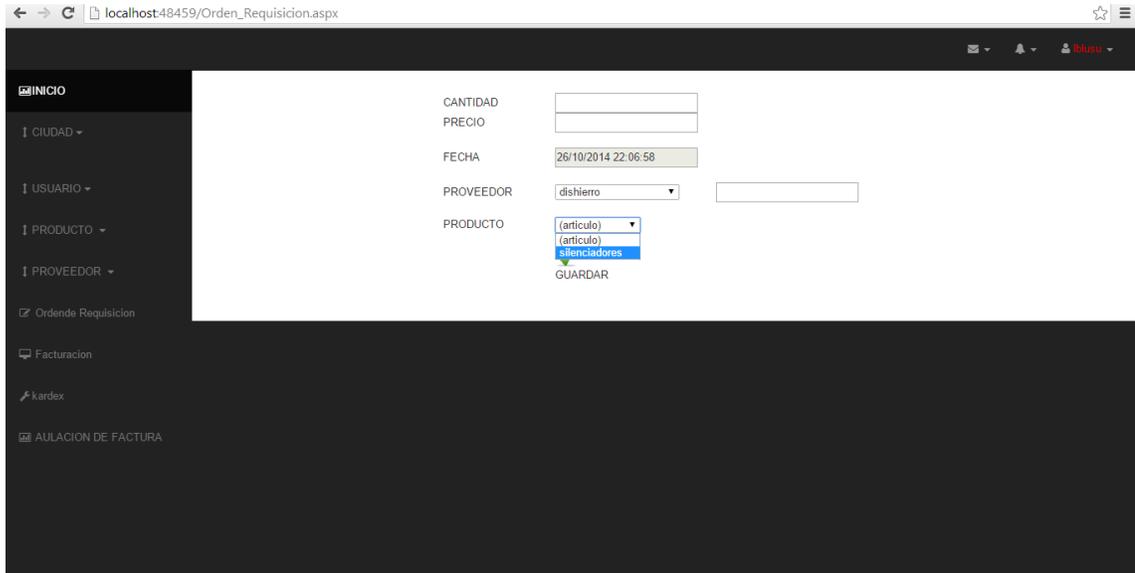


Figura 7: Nuevo Producto: El usuario registrar un nuevo producto llenando los registros necesarios para la información de dicho producto, seleccionar una imagen del producto y su respectivo proveedor.

Orden Requisición

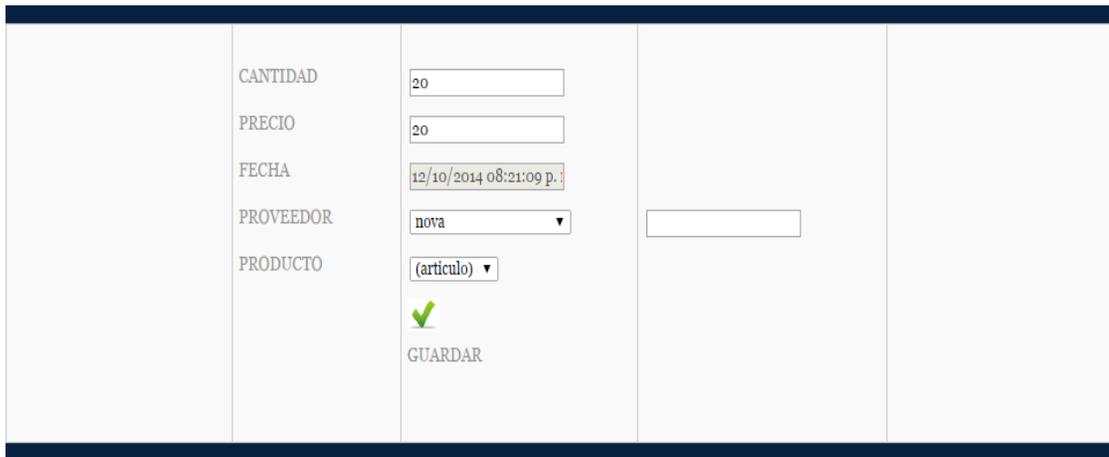


Figura 8: crear orden requisición Ingreso de artículo de requisición para el envío del producto al proveedor

Factura del cliente

OCEANO PACIFICO
FACTURA:

OCEANO PACIFICO
FECHA: 12/10/2014 08:23:11 p. m.

<input type="text" value="1"/>	<input type="text" value="1717554305"/>		<input type="text" value="victor"/>
CECULA			NOMBRE
APELLIDO	<input type="text" value="cuichan"/>		EMAIL
DIRECCION	<input type="text" value="D@HOTMAIL.COM"/>		CONDADO
CIUDAD	<input type="text" value="1"/>		TELEFONO
			<input type="text" value="252"/>

CODIGO PEDIDO	DESCRIPCION	CANTIDAD	PRECIO UNITARIO	TOTAL
2	catalisadores	5	2	10
2	catalisadores	5	2	10
2	catalisadores	5	2	10
1	silenciadores	1	2	2
2	catalisadores	5	2	10
Total:				42

SUBTOTAL:

IVA:

TOTAL:

Figura 9: crear factura: Se procede a crear factura con datos de cliente y información de producto.

Lista productos

buscar:



	CODIGO	DESCRIPCION	Stock
	2	catalisadores	75

Figura 10: lista de productos: Lista de productos en existencia para añadir a la factura.

Kardex del producto

← → localhost:12921/Kkardex.aspx

KARDEX

PROVEEDOR:

PRODUCTO:

ID	Fecha	Detalle	Cantidad	Precio	Total	Cantidad	Precio	Total	Cantidad	Precio	Total
1	12/10/2014 01:46:45 a. m.	Saldo inicial	50	2.00	100	0	0.00	0	50	2.00	100
3	12/10/2014 01:57:23 a. m.	ordende requisicion	50	2.00	100	0	0.00	0	100	2.00	200
4	12/10/2014 01:58:46 a. m.	ordende requisicion	25	55.00	1375	0	0.00	0	125	55.00	6875
6	12/10/2014 02:04:46 a. m.	Venta de factura	0	0.00	0	1	2.00	2	124	2.00	248



Figura 11: Kardex: Se podrá ver la entrada, salida y existencias de los productos.



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE ANALISIS DE SISTEMAS

CONTROL DE INVENTARIO DE BODEGA Y FACTURACIÓN MEDIANTE UNA
APLICACIÓN WEB PARA LA EMPRESA OCÉANO PACÍFICO

MANUAL TECNICO

Autor: Alex Darío Cuichán Apunte.

Tutor: Lic. Patricia Garzón Msc.

Quito, Octubre 2014

INDICE DE MANUAL TECNICO

Título	pagina
A.07.01 Login del sistema	78
A.07.02 Menu Administrador	80
A.07.03 Proveedor	81
A.07.04 Ingreso de nuevo empleado.....	83
A.07.05 Inventario	85
A.07.06 Orden de Requisición.....	90
A.07.07 Factura.....	94
A.07.08 Kardex.....	101

A.07 Manual Técnico

Facilitar la estructura técnica de cada módulo, igualmente como los detalles técnicos de forma clara mediante la descripción del sistema, para que los usuarios técnicos obtengan una mayor visión de la estructura y funcionamiento del sistema.

A.07.01 Login del sistema

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Logica_Negocio;
using Tbl_Datos;
namespace Facturacion_Inventario
{
    public partial class Login_Admin1 : System.Web.UI.Page
    {
        Logica_Tiusuario tipusu = new Logica_Tiusuario();
        protected void Page_Load(object sender, EventArgs e)
        {
            if(!IsPostBack)
            {
                cargartipousuario();
            }
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            ingre();
        }
        private void ingre()
        {
            string nombre = (input1.Value);
            string codigo;
            // int cedula = Convert.ToInt32(HF_cedula.Value);
            bool existe;

            Logica_Login Login_Usu = new Logica_Login();
            if (input1.Value == "")
            {
                string script = "alert('Ingreso email...');";
                ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
script, true);
                return;
            }
            if (input2.Value == "")
            {
                string script = "alert('Ingreso clave...');";
                ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
script, true);
                return;
            }
        }
    }
}
```



```
}
existe = Login_Usu.Existe_Usuario(input1.Value);

// existe = Logica_Login.ac.ExisteUsuario(ingreso());

if (existe)
{
    //int tusu = 2;
    existe = Login_Usu.autoidentificarusuario(input1.Value,
input2.Value, Convert.ToInt32(ddltip.SelectedValue));
    int nn = Convert.ToInt32(ddltip.SelectedValue);
    if (existe)
    {

        //string usuario = input1.Value;
        //Session["Usuario"] = usuario;

        switch (nn)
        {
            case 1:
                Response.Redirect("~/Administrador.aspx", true);

                //case 2:
                //    string usuario1 = input1.Value;
                //    Session["nom"] = usuario1;
                //    string codd =
(Login_Usu.Obtenerid(input1.Value, input2.Value).Rows[0]["usu_id"].ToString());

                //    Response.Redirect("~/DatosVehiculo.aspx?cod=" +
codd, true);

                break;
            case 2:
                Response.Redirect("~/Secretaria.aspx", true);

                break;
            case 3:

                Response.Redirect("~/Bodegero.aspx", true);

                break;

        }
    }
    else
    {
        string script = "alert('Contraseña incorrecta...');";
        ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script, true);
        return;
    }
}
else
```



```
        {
            //Lbl_UsuMensaje.Text = "CONTRASENA INCORRECTA";
            string script = "alert('Usuario incorreto...')";
            ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script, true);
            return;
        }
    }

    private void cargartipousuario()
    {
        List<tbl_tipousuario> perlista = new List<tbl_tipousuario>();
        perlista = tipusu.ObtenerTipousuario();
        perlista.Insert(0, new tbl_tipousuario() { TIPUSU_DESCRIPCION =
"(ESCJA EL TIPO DE USUARIO..)"});
        ddltip.DataSource = perlista;
        ddltip.DataTextField = "TIPUSU_DESCRIPCION";
        ddltip.DataValueField = "TIPUSU_ID";
        ddltip.DataBind();
    }
}
}
```

A.07.02 Menu Administrador

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Acceso_Datos;
using Logica_Negocio;

namespace Facturacion_Inventario
{
    public partial class Administrador1 : System.Web.UI.Page
    {
        // logi
        protected void Page_Load(object sender, EventArgs e)
        {
            if(!IsPostBack)
            {
                // int Pedidoanterior =
                Logica_Administrador.Accesomysql.ultimo_registro();
            }
        }

        protected void ImageButton2_Click(object sender, ImageClickEventArgs e)
        {
            Response.Redirect("~/Admin_Usuario.aspx");
        }
    }
}
```



```
    }  
  
    protected void ImageButton1_Click(object sender, ImageClickEventArgs e)  
    {  
        Response.Redirect("~/Proveedor.aspx");  
    }  
  
    protected void ImageButton3_Click(object sender, ImageClickEventArgs e)  
    {  
        Response.Redirect("~/Factura.aspx");  
    }  
  
    protected void ImageButton5_Click(object sender, ImageClickEventArgs e)  
    {  
        Response.Redirect("~/Inventario.aspx");  
    }  
  
    protected void ImageButton4_Click(object sender, ImageClickEventArgs e)  
    {  
        Response.Redirect("~/Orden_Adquisicion.aspx");  
    }  
  
    protected void ImageButton6_Click(object sender, ImageClickEventArgs e)  
    {  
        Response.Redirect("~/Kkardex.aspx");  
    }  
    }  
}
```

A.07.03 Proveedor

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
using Tbl_Datos;  
using Logica_Negocio;  
  
namespace Facturacion_Inventario  
{  
    public partial class Proveedor1 : System.Web.UI.Page  
    {  
        Logica_ciudad tiper = new Logica_ciudad();  
        Logica_Proveedor provee = new Logica_Proveedor();  
        bool existe;  
        private tbl_proveedor _reinfo = new tbl_proveedor();  
        private tbl_proveedor _re { get; set; }  
  
        protected void Page_Load(object sender, EventArgs e)  
        {  
            if(!IsPostBack)  
            {
```



```
        cargarciudad();
    }
}
private void cargarciudad()
{
    List<tbl_ciudad> perlista = new List<tbl_ciudad>();
    perlista = tiper.Obtenerciudad();
    perlista.Insert(0, new tbl_ciudad() { CIU_NOMBRE = "(escoja la
ciudad..)" });
    ddlciudad.DataSource = perlista;
    ddlciudad.DataTextField = "CIU_NOMBRE";
    ddlciudad.DataValueField = "CIUD_ID";
    ddlciudad.DataBind();
}
protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
    guardar();
}
private void guardar()
{
    if (Lbl_Mensajes.Text == "Este usuario ya existe ingrese otro o
Inicie sesion<br>")
    {
        Lbl_Mensajes.Text = "";
        // Lbl_Mensajes.Text = "Ingrese otro email q no este en la base
de datos";
        string script = "alert('Email ya existe');";
        ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
script, true);
        return;
    }
    else
    {
        if (Hff_Id.Value == "0")
        {
            if (ddlciudad.SelectedIndex == 0)
            {
                string script = "alert('ingrese ciudad..')";
                ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script, true);
                return;
            }
            existe = provee.InsertProveedor(ingreso());
            string scriptt = "alert('PROVEEDOR GUARDADO')";
            ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", scriptt, true);
            return;
        }
        else
        {
        }
    }
}
}
```



```
private tbl_proveedor ingreso()//metodo
{
    // Hff_Id.Value = Convert.ToString(3340);
    _reinfo.PROVEE_ID= Convert.ToInt32(Hff_Id.Value);
    _reinfo.PROVEE_NOMBRE = (txtnombre.Text);

    _reinfo.PROVEE_CEDULA =Convert.ToInt32(txtruc.Text);
    _reinfo.PROVEE_DIRECCION = txtdireccion.Text;
    _reinfo.PROVEE_TELEFONO =Convert.ToInt32(txtelefono.Text);

    _reinfo.PROVEE_EMAIL = txtemail.Text;
    _reinfo.PROVEE_ESTADO = (Convert.ToChar("A"));

    _reinfo.CIUD_ID = Convert.ToInt32(ddlciudad.SelectedValue);

    return _reinfo;
}
}
```

A.07.04 Ingreso de nuevo empleado

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Logica_Negocio;
using Tbl_Datos;

namespace Facturacion_Inventario
{
    public partial class Admin_Usuario1 : System.Web.UI.Page
    {
        Logica_Tiusuario tipusu = new Logica_Tiusuario();
        private tbl_UsuAdmin _reinfo = new tbl_UsuAdmin();
        private tbl_UsuAdmin _re { get; set; }
        Logica_UsuarioAdmin logusu = new Logica_UsuarioAdmin();

        bool existe;

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                cargartipousuario();
            }
        }

        protected void Imgbtn_Guardar_Click(object sender, ImageClickEventArgs e)
        {

```



```
        Guardar();
    }
    private void Guardar()
    {
        if (Hff_Id.Value == "0")
        {
            if (ddltipusu.SelectedIndex == 0)
            {
                string script = "alert('Ingreso Tipo de Usuario...');";
                ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script, true);
                return;
            }
            else
            {
                existe = logusu.InsertarUsuarioAdmin(ingreso());
                string scriptT = "alert('USUARIO GUARDADO...');";
                ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", scriptT, true);
                // Response.Redirect("Factura.aspx?cof=" +
txtcedula.Text.Trim() + txt_nombre.Text.Trim() + "&codd=" +
txtapellido.Text.Trim() + "&coddd=" + txt_emai.Text.Trim() + "&codddd=" +
txt_direccion.Text.Trim() + "&coddddd=" + txt_telefon.Text.Trim() + "
&coddddddd=" + ddlciudad.SelectedValue.Trim(), true);

                return;
            }
        }
        else
        {
        }
    }

    private tbl_UsuAdmin ingreso()//metodo
    {

        // Hff_Id.Value = Convert.ToString(3340);
        _reinfo.ADNUSU_ID= Convert.ToInt32(Hff_Id.Value);
        _reinfo.ADNUSU_NOMBRE= (txt_nombre.Text);

        _reinfo.ADNUSU_APELLIDO = txtapellido.Text;
        _reinfo.ADNUSU_USUARIO = txt_usuario.Text;
        _reinfo.ADNUSU_CLAVE= Txt_clave.Text;

        _reinfo.ADNUSU_DIRECCION = txt_direccion.Text;
        _reinfo.ADNUSU_TELEFONO = Convert.ToInt32(txt_telefono.Text);

        _reinfo.ADNUSU_ESTADO = (Convert.ToChar("A"));

        _reinfo.TIPUSU_ID = Convert.ToInt32(ddltipusu.SelectedValue);

        return _reinfo;
    }
}
```



```
    }

    private void cargartipousuario()
    {
        List<tbl_tipousuario> perlista = new List<tbl_tipousuario>();
        perlista = tipusu.ObtenerTipousuario();
        perlista.Insert(0, new tbl_tipousuario() { TIPUSU_DESCRIPCION = "(TIPO
DE USUARIO..)" });
        ddltipusu.DataSource = perlista;
        ddltipusu.DataTextField = "TIPUSU_DESCRIPCION";
        ddltipusu.DataValueField = "TIPUSU_ID";
        ddltipusu.DataBind();
    }
}
}
```

A.07.05 Inventario

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Logica_Negocio;
using Tbl_Datos;

namespace Facturacion_Inventario
{
    public partial class Inventario : System.Web.UI.Page
    {
        Logica_Proveedor logpro = new Logica_Proveedor();
        Logica_Producto_Nombre logproducto = new Logica_Producto_Nombre();
        Logica_Inventario loginv = new Logica_Inventario();
        Logica_Kardex karde = new Logica_Kardex();
        Logica_Kardexx kar = new Logica_Kardexx();

        private tbl_producto _reinfo = new tbl_producto();
        private tbl_kardex _reinfoKardex = new tbl_kardex();

        private DateTime fechasistema;
        private string id;
        private tbl_producto _re { get; set; }

        bool existe, existeec;
        private string detalle;
        protected void Page_Load(object sender, EventArgs e)
        {
            if(!IsPostBack)
            {
                CargarProveedor();
                CargarProducto();
            }
        }
    }
}
```



```
private void CargarProducto()
{
    List<tbl_proucto_Nombre> ubilista = new List<tbl_proucto_Nombre>();
    ubilista = logproducto.Obtenernombreproveedor();
    ubilista.Insert(0, new tbl_proucto_Nombre() { PROD_DESCRIPCION =
"(Escoja el producto)" });
    ddlpro.DataSource = ubilista;
    ddlpro.DataTextField = "PROD_DESCRIPCION";
    ddlpro.DataValueField = "PROD_ID";
    ddlpro.DataBind();
}

private void CargarProveedor()
{
    List<tbl_ObtenerProveedor> ubilista = new
List<tbl_ObtenerProveedor>();
    ubilista = logpro.Obtenernombreprovedor();
    ubilista.Insert(0, new tbl_ObtenerProveedor() { PROVEE_NOMBRE =
"(Escoja el proveedor)" });
    ddlprove.DataSource = ubilista;
    ddlprove.DataTextField = "PROVEE_NOMBRE";
    ddlprove.DataValueField = "PROVEE_ID";
    ddlprove.DataBind();
}

protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
    guardar();
}

private void guardar()
{
    if (txtdesc.Text == "")
    {
        string script = "alert('Ingrese la cantidad');";
        ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
script, true);
        return;
    }
    else if (txtcant.Text == "")
    {
        string script = "alert('ingrese la cantidad...');";
        ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script, true);
        return;
    }
    else if(txtstock.Text=="")
    {
        string scriptt = "alert('ingrese el stock');";
        ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", scriptt, true);
    }
}
```



```
        return;
    }
    //else if (ddlprove.SelectedValue== "")
    //{
    //    string scriptt = "alert('ingrese el proveedor');";
    //    ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", scriptt, true);
    //    return;
    //}

    existe = loginv.InsertarInve(ingreso());
    // existe = (kar.Existe_id(ddlpro.SelectedValue));

    existe = karde.InsertarInve((ingresoKarde()));

    string scripttt = "alert('PRODUCTO GUARDADO');";
    ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", scripttt, true);
    return;
}

private tbl_producto ingreso()//metodo
{
    // Hff_Id.Value = Convert.ToString(3340);
    _reinfo.PROD_ID = Convert.ToInt32(Hff_Id.Value);
    fechasistema = (DateTime.Now);
    // fechasistema =Convert.ToInt32(DateTime.Now.ToString("ddMMyyyy"));

    _reinfo.PROD_FECHA=Convert.ToDateTime(Convert.ToString(fechasistema));

    _reinfo.PROD_DESCRIPCION = (txtdesc.Text);
    _reinfo.PROD_CANTIDAD = Convert.ToInt32(txtcant.Text);

    _reinfo.PROD_PRECIOUNITARIO = Convert.ToInt32(txtprecio.Text);
    _reinfo.PROD_STOCK_MIN =Convert.ToInt32(txtstock.Text);
    _reinfo.PROD_STOCK = Convert.ToInt32(txtstock.Text);

    _reinfo.PROD_ESTADO = (Convert.ToChar("A"));

    _reinfo.PROVEE_ID=Convert.ToInt32(ddlprove.SelectedValue);
    _reinfo.USU_ID = (Convert.ToChar("1"));

    return _reinfo;
}
private tbl_kardex ingresoKarde()//metodo
{
    // Hff_Id.Value = Convert.ToString(3340);
    _reinfoKardex.KARD_ID= Convert.ToInt32(Hff_Id.Value);
    fechasistema = (DateTime.Now);
    // fechasistema =Convert.ToInt32(DateTime.Now.ToString("ddMMyyyy"));
```



```
        _reinfoKardex.KARD_FECHA=
Convert.ToDateTime(Convert.ToString(fechasistema));

        if(ddlpro.SelectedValue=="0")
        {
            detalle = "Saldo inicial";
            _reinfoKardex.KARD_DETALLE = Convert.ToString(detalle);

            _reinfoKardex.KARD_ENTCANTIDA =
Convert.ToInt32(txtcant.Text);
            _reinfoKardex.KARD_ENTPRECIO =
Convert.ToInt32(txtprecio.Text);

            int a = Convert.ToInt32(txtcant.Text) *
Convert.ToInt32(txtprecio.Text);

            _reinfoKardex.KARD_ENTTOTAL = Convert.ToInt32(a);

            if (ddlpro.SelectedValue != "0")
            {
                id =
(kar.ObtenerKardexEntras(Convert.ToInt32(ddlpro.SelectedValue)).Rows[0]["kard_exi
catidad"].ToString());

            }
            else
            {
                id = Convert.ToString(kar.secuencial1());
            }

            // int u = Convert.ToInt32(id) +
Convert.ToInt32(txtcant.Text);

            _reinfoKardex.KARD_EXICATIDAD =
Convert.ToInt32(txtcant.Text);

            _reinfoKardex.KARD_EXIPRECIO =
Convert.ToInt32(txtprecio.Text);

            int aa = Convert.ToInt32(txtcant.Text) *
Convert.ToInt32(txtprecio.Text);

            _reinfoKardex.KARD_EXITOTAL = (aa);

            _reinfoKardex.FACT_ID = Convert.ToInt32(a);

        }
        else
        {
            detalle = "ordende requisicion";
```



```
_reinfoKardex.KARD_DETALLE = Convert.ToString(detalle);

_reinfoKardex.KARD_ENTCANTIDA = Convert.ToInt32(txtcant.Text);
_reinfoKardex.KARD_ENTPRECIO = Convert.ToInt32(txtprecio.Text);

int a = Convert.ToInt32(txtcant.Text) *
Convert.ToInt32(txtprecio.Text);

_reinfoKardex.KARD_ENTTOTAL = Convert.ToInt32(a);

// string id = Convert.ToString(kar.secuencial1());
string id =
(kar.ObtenerKardexEntras(Convert.ToInt32(ddlpro.SelectedValue)).Rows[0]["kard_exi
catidad"].ToString());

int u = Convert.ToInt32(id) + Convert.ToInt32(txtcant.Text);

_reinfoKardex.KARD_EXICATIDAD = Convert.ToInt32(u);
_reinfoKardex.KARD_EXIPRECIO = Convert.ToInt32(txtprecio.Text);

int aa = Convert.ToInt32(u) * Convert.ToInt32(txtprecio.Text);

_reinfoKardex.KARD_EXITOTAL = (aa);

_reinfoKardex.FACT_ID = Convert.ToInt32(a);
}

if(ddlpro.SelectedValue=="0")
{
int ultimo = loginv.ultimo();
if(ultimo==0)
{
_reinfoKardex.PROD_ID = Convert.ToInt32("1");
}
else
{
_reinfoKardex.PROD_ID = Convert.ToInt32(ultimo);
}
}
else
{
_reinfoKardex.PROD_ID = Convert.ToInt32(ddlpro.SelectedValue);
}
}
```

```

        return _reinfoKardex;
    }

}
}

```

A.07.06 Orden de Requisición

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Tbl_Datos;
using Logica_Negocio;
using System.Net.Mail;
using System.Net;
using ControlParking;
namespace Facturacion_Inventario
{
    public partial class Orden_Adquisicion1 : System.Web.UI.Page
    {
        bool existe;
        private DateTime fechasistema;
        private string fechasistema1;
        Logica_Proveedor logpro = new Logica_Proveedor();
        Logica_OrednAdquisicion logob = new Logica_OrednAdquisicion();
        private tbl_OrdenAdquisicion _reinfo = new tbl_OrdenAdquisicion();

        protected void Page_Load(object sender, EventArgs e)
        {
            if(!IsPostBack)
            {
                MultiView1.ActiveViewIndex = 1;
                cargarfecha();
                CargarProveedor();
            }
        }

        protected void ImageButton2_Click(object sender, ImageClickEventArgs e)
        {
            guardar();
        }
        private void guardar()
        {
            if(txtcantidad.Text=="")
            {
                string script = "alert('Ingrese la cantidad');";
                ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script, true);
                return;
            }
        }
    }
}

```



```
        else if(txtprecio.Text=="")
        {
            string script = "alert('Ingrese el precio');";
            ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script, true);
            return;
        }
        else if (ddlprovee.SelectedValue== "")
        {
            string script = "alert('Ingrese el proveedor');";
            ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script, true);
            return;
        }
        else if (ddlproducto.SelectedValue == "")
        {
            string script = "alert('Ingrese el producto');";
            ScriptManager.RegisterStartupScript(this, typeof(Page),
"alerta", script, true);
            return;
        }
    }

    else
    {
        existe = logob.Insertarrequisicion(ingreso());

        if(existe)
        {
            MultiView1.ActiveViewIndex = 0;
            conectar();
        }
    }
}
private void conectar()
{
    int codigo =
(int.Parse(logob.ObtenerUltimoRegistro().Rows[0]["secuencial"].ToString()));

    GridView1.DataSource = logob.ObtenerDatosorden(codigo);

    GridView1.DataBind();
}
private tbl_OrdenAdquisicion ingreso()//metodo
{
    Hff_Id.Value = Convert.ToString(Hff_Id.Value);
    _reinfo.ORDEREQUI_ID= Convert.ToInt32(Hff_Id.Value);
    _reinfo.PROVEE_ID = Convert.ToInt32(ddlprovee.SelectedValue);
}
```



```
_reinfo.PRODPROVE_ID=Convert.ToInt32(ddlproducto.SelectedValue);
_reinfo.ORDEADQ_CANTIDAD = Convert.ToInt32(txtcantidad.Text);
_reinfo.ORDEADQ_PRECIO=Convert.ToDecimal(txtprecio.Text);
fechasistema =(DateTime.Now);

_reinfo.ORDEADQ_FECHA_CRECION=Convert.ToDateTime(fechasistema);

return _reinfo;
}

private void cargarfecha()
{
    fechasistema = (DateTime.Now);
    txtfecha.Text =Convert.ToString(fechasistema);
}

private void CargarProveedor()
{
    List<tbl_ObtenerProveedor> ubilista = new
List<tbl_ObtenerProveedor>();
    ubilista = logpro.Obtenernombreprovedor();
    ubilista.Insert(0, new tbl_ObtenerProveedor() { PROVEE_NOMBRE =
"(Escoja el proveedor)" });
    ddlprovee.DataSource = ubilista;
    ddlprovee.DataTextField = "PROVEE_NOMBRE";
    ddlprovee.DataValueField = "PROVEE_ID";
    ddlprovee.DataBind();
}

protected void ddlprovee_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        int codubi = int.Parse(ddlprovee.SelectedValue);
        List<tbl_ProdProveedor> pro = new List<tbl_ProdProveedor>();

        pro = Logica_ObtenerNomProd._lista(codubi);
        pro.Insert(0, new tbl_ProdProveedor() { PRODPROVE_NOMBRE =
"(articulo)" });
        ddlproducto.DataSource = pro;
        ddlproducto.DataTextField = "PRODPROVE_NOMBRE";
        ddlproducto.DataValueField = "PRODPROVE_ID";
        ddlproducto.DataBind();
    }
    catch (Exception)
    {
        throw;
    }
}

protected void ddlproducto_SelectedIndexChanged(object sender, EventArgs
e)
{
}
}
```



```
protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
{
    GridViewRow row =GridView1.SelectedRow;
    Hff_Id.Value = GridView1.SelectedRow.Cells[0].Text.ToString();
    ddlprovee.Text = GridView1.SelectedRow.Cells[1].Text.ToString();
    ddlproducto.Text = GridView1.SelectedRow.Cells[2].Text.ToString();
    txtcantidad.Text = GridView1.SelectedRow.Cells[3].Text.ToString();
    txtprecio.Text = GridView1.SelectedRow.Cells[4].Text.ToString();
    txtfecha.Text = GridView1.SelectedRow.Cells[5].Text.ToString();

    //ddlusuario.Text = Dgv_Reserva.SelectedRow.Cells[5].Text.ToString();
    if (Hff_Id.Value != null || Hff_Id.Value != null)
    {
        MultiView1.ActiveViewIndex = 1;
    }
}

protected void Button1_Click(object sender, EventArgs e)
{
    int a =Convert.ToInt32(ddlprovee.SelectedValue);

    string nombre =
((logob.Obtenercantidad(a)).Rows[0]["secuencia"].ToString());
    string precio =
((logob.Obtenerprecio(a)).Rows[0]["secuencia"].ToString());
    // string proveedor =
((logob.Obtenerproveedor(a)).Rows[0]["provee_nombre"].ToString());
    string producto =
((logob.Obtenerproducto(a)).Rows[0]["prod_descripcion"].ToString());

    fechasisistema1 =Convert.ToString(DateTime.Now);

    string email =
((logob.Obteneremail(a)).Rows[0]["provee_email"].ToString());

    SendMail m = new SendMail("daniel_1990c@hotmail.com",
"nmvdca19902012");
    m.sendMail(email, nombre,precio,fechasisistema1,producto);
    string scriptt = "alert('EL PEDIDO FUE ENVIADO EXITOSAMENTE...');";
    ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
scriptt, true);
    return;
}

protected void txtcantidad_TextChanged(object sender, EventArgs e)
{
}
}
```

A.07.07 Factura

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Logica_Negocio.Utilidades;
using Logica_Negocio;
using MySql.Data.MySqlClient;
using System.Data;
using Tbl_Datos;
namespace Facturacion_Inventario
{
    public partial class Factura1 : System.Web.UI.Page
    {
        public static MySqlConnection conexion = new
        MySqlConnection("server=localhost;User Id=root;database=facturacionnew");
        private static MySqlCommand comando;
        private static MySQLTransaction transaccion;
        private DateTime fechasistema;
        Logica_Kardex karde = new Logica_Kardex();
        Logica_Kardexx kar = new Logica_Kardexx();

        private decimal total = 0;
        private tbl_Factura _reinfo = new tbl_Factura();
        private tbl_Factura _re { get; set; }
        private tbl_kardex _reinfoKardex = new tbl_kardex();

        Logica_Factura logfac = new Logica_Factura();
        Logica_DetalleFactura logpeido = new Logica_DetalleFactura();
        bool existe, existe2;
        bool exi;
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Request.QueryString["usuarioid"] == null ||
                Request.QueryString["lista"] == null || Request.QueryString["cof"] == null ||
                Request.QueryString["cod"] == null || Request.QueryString["codd"] == null ||
                Request.QueryString["coddd"] == null || Request.QueryString["codddd"] == null ||
                Request.QueryString["coddddd"] == null || Request.QueryString["coddddddd"] ==
                null)
            {

            }

            else
            {

                txtced.Text = "" + Request.QueryString["cof"];

                txtnom.Text = "" + Request.QueryString["cod"];
                txtape.Text = "" + Request.QueryString["codd"];
                txtemail.Text = "" + Request.QueryString["coddd"];
                txtdire.Text = "" + Request.QueryString["codddd"];
                txttele.Text = "" + Request.QueryString["coddddd"];
                txtciudad.Text = "" + Request.QueryString["coddddddd"];
            }
        }
    }
}

```



```
    }

    // txtced.Text = "" + Request.QueryString["cof"];
    hfced.Value = "" + Request.QueryString["usu"];
    hfprecio.Value = "" + Request.QueryString["precio"];
    hfusuarioid.Value = "" + Request.QueryString["usuarioid"];
    HiddenField1.Value = "" + Request.QueryString["lista"];
    HiddenField2.Value = "" + Request.QueryString["pro"];
    // hfced.Value = txtced.Text;
    // Button1.Attributes.Add("onclick", "javascript:placas()");//
    // ImageButton2.Attributes.Add("onclick", "javascript:cliente()");//
    if (HiddenField2.Value != "")
    {
        if (!IsPostBack)
        {
            // guardar();
            Label1.Text = Convert.ToString(DateTime.Now);
            guarda();
            buscarNombre();
            conectar();

            cargarDatos();
        }
    }

    Label1.Text = Convert.ToString(DateTime.Now);

}
private void cargarDatos()
{
    //MySqlDataAdapter Adp = new MySqlDataAdapter("SELECT
    ((fact_valorunitario * fact_cantidad)) AS total, tbl_factura.fact_id,
    tbl_factura.fact_fecha, tbl_factura.fact_cantidad,
    tbl_factura.fact_valorunitario,usu_id,tbl_factura.fact_iva FROM tbl_factura WHERE
    usu_id='" + id + "'", conexion);
    //DataSet ds = new DataSet();
    //Adp.Fill(ds);

    //GridView1.DataSource = ds;
    //GridView1.DataBind();
}
private void guarda()
{
    exi = logfac.InsertarFactura(ingreso());
    exi = karde.Insertarnve(ingresokardex());
}
private tbl_Factura ingreso()//metodo
{
    // Hff_Id.Value = Convert.ToString(3340);
    _reinfo.FACT_ID= Convert.ToInt32(Hff_Id.Value);
}
```



```
fechasistema = (DateTime.Now);

_reinfo.FACT_FECHA = Convert.ToDateTime(fechasistema);

_reinfo.FACT_CANTIDAD =Convert.ToInt32(HiddenField2.Value);
_reinfo.FACT_VALORUNITARIO =Convert.ToDecimal(hfprecio.Value);

int iva = 12;
_reinfo.FACT_IVA = iva;

_reinfo.FACT_ESTADO = Convert.ToChar("A");

// int a = logfac.secuencial();

// string coddd =
(logfac.Obtenerid(Convert.ToInt32(hfcdd.Value)).Rows[0]["usu_id"].ToString());

_reinfo.USU_ID = Convert.ToInt32(hfusuarioid.Value);
_reinfo.PROD_ID= Convert.ToInt32(HiddenField1.Value);

return _reinfo;
}
private tbl_kardex ingresokardex()//metodo
{

_reinfoKardex.KARD_ID = Convert.ToInt32(Hff_Id.Value);
fechasistema = (DateTime.Now);

_reinfoKardex.KARD_FECHA= Convert.ToDateTime(fechasistema);

string detalle = "Venta de factura";
_reinfoKardex.KARD_DETALLE = detalle;
_reinfoKardex.KARD_SALCATIDAD = Convert.ToInt32(HiddenField2.Value);
_reinfoKardex.KARD_SALPRECIO = Convert.ToInt32(hfprecio.Value);

int a = Convert.ToInt32(HiddenField2.Value) *
Convert.ToInt32(hfprecio.Value);

_reinfoKardex.KARD_SALTOTAL = (a);

//string id = Convert.ToString(kar.secuencial1());
//txtid.Text = id;
string id =
(kar.ObtenerKardexEntras(Convert.ToInt32(HiddenField1.Value)).Rows[0]["kard_exica
tidad"].ToString());

int nueva =Convert.ToInt32( id) - Convert.ToInt32(HiddenField2.Value);

_reinfoKardex.KARD_EXICATIDAD = Convert.ToInt32(nueva);
_reinfoKardex.KARD_EXIPRECIO = Convert.ToInt32(hfprecio.Value);
```



```
        int nuevatotal = Convert.ToInt32(nueva) *
Convert.ToInt32(hfprecio.Value);

        _reinfoKardex.KARD_EXITOTAL = (nuevatotal);

        _reinfoKardex.PROD_ID = Convert.ToInt32(HiddenField1.Value);

        return _reinfoKardex;
    }
    //private void entradas()
    //{

        //    GridView1.DataSource =
kar.ObtenerKardex(Convert.ToInt32(HiddenField1.Value));
        //    GridView1.DataBind();

        //}
protected void Button2_Click(object sender, EventArgs e)
{

}

protected void TextBox1_TextChanged(object sender, EventArgs e)
{

    validarcedula();
}
private void validarcedula()
{
    string cedula = txtced.Text;

    Validar_Cedula Validacion = new Validar_Cedula();

    if (Validacion.CedulaCorrecta(cedula))
    {

        //if (Convert.ToString(logusu.Existe_Cedula(cedula)))
        //{
        //    mensaje += "Este usuario ya existe ingrese otro o Inicie
sesion<br>";
        //}

        lblmensaje.Text = "";
        txtnom.Focus();
    }
    else
    {
        string script = "alert('Cedula Incorrecta Ingrese de
Nuevo...');";
        ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
script, true);
    }
}
```



```
        return;
    }
}

protected void ImageButton1_Click(object sender, ImageClickEventArgs e)
{
    buscarNombre();
}
private void buscarNombre()
{
    existe = logfac.Existe_Usuario(txtced.Text);

    if (existe)
    {
        string id =
(logfac.Obtenerid(Convert.ToInt32(txtced.Text)).Rows[0]["usu_id"].ToString());
        lblidfactura.Text = id;
        // txtid.Text = id;

        string codd =
(logfac.ObtenerNombre(txtced.Text).Rows[0]["usu_nombre"].ToString());
        txtnom.Text = codd;
        string codd1 =
(logfac.ObtenerApellido(txtced.Text).Rows[0]["usu_apellido"].ToString());
        txtape.Text = codd1;
        string codd2 =
(logfac.ObtenerEmail(txtced.Text).Rows[0]["usu_email"].ToString());
        txtemail.Text = codd2;
        string codd3 =
(logfac.Obtenerdireccion(txtced.Text).Rows[0]["usu_direccion"].ToString());
        txtdire.Text = codd3;
        string codd4 =
(logfac.ObtenerTelefono(txtced.Text).Rows[0]["usu_telefono"].ToString());
        txttele.Text = codd4;
        string codd5 =
(logfac.ObtenerCiudad(txtced.Text).Rows[0]["ciud_id"].ToString());
        txtciudad.Text = codd5;
        txtiva.Text = "12%";
        return;
    }

    existe = logfac.Existe_Usuario1(hfusuarioid.Value);

    if(existe)
    {
        string id =
(logfac.Obtenerid1(Convert.ToInt32(hfusuarioid.Value)).Rows[0]["usu_id"].ToString
());
        lblidfactura.Text = id;
        // txtid.Text = id;
        string cedula =
(logfac.Obtenercedula(Convert.ToInt32(hfusuarioid.Value)).Rows[0]["usu_cedula"].T
oString());
        txtced.Text = cedula;
    }
}
```



```
        string codd =
(logfac.ObtenerNombre1(hfusuarioid.Value).Rows[0]["usu_nombre"].ToString());
        txtnom.Text = codd;
        string codd1 =
(logfac.ObtenerApellido1(hfusuarioid.Value).Rows[0]["usu_apellido"].ToString());
        txtape.Text = codd1;
        string codd2 =
(logfac.ObtenerEmail1(hfusuarioid.Value).Rows[0]["usu_email"].ToString());
        txtemail.Text = codd2;
        string codd3 =
(logfac.Obtenerdireccion1(hfusuarioid.Value).Rows[0]["usu_direccion"].ToString())
;
        txtdire.Text = codd3;
        string codd4 =
(logfac.ObtenerTelefono1(hfusuarioid.Value).Rows[0]["usu_telefono"].ToString());
        txttele.Text = codd4;
        string codd5 =
(logfac.ObtenerCiudad1(hfusuarioid.Value).Rows[0]["ciud_id"].ToString());
        txtciudad.Text = codd5;
        txtiva.Text = "12%";
    }

    else
    {
        string script = "alert('Usuario no existe...');";
        ScriptManager.RegisterStartupScript(this, typeof(Page), "alerta",
script, true);
        return;
    }
}

protected void txtced_DataBinding(object sender, EventArgs e)
{
    validarcedula();
}

protected void TextBox6_TextChanged(object sender, EventArgs e)
{
}

protected void Button1_Click(object sender, EventArgs e)
{
    int a =Convert.ToInt32(lblidfactura.Text);
    Response.Redirect("Lista_Producto.aspx?u=" + a.ToString().Trim(),
true);
}

protected void ImageButton2_Click(object sender, ImageClickEventArgs e)
{
    Response.Redirect("Cliente.aspx", true);
}

private void conectar()
{
```



```
// GridView1.DataSource =
logpeido.ObtenerPedido(Convert.ToInt32(HiddenField1.Value));

// GridView1.DataSource =
logpeido.ObtenerPedido(Convert.ToInt32(hfced.Value));
GridView1.DataSource =
logpeido.ObtenerPedido(Convert.ToInt32(hfusuarioid.Value));

GridView1.DataBind();
}
protected void GridView1_RowDataBound(object sender, GridViewRowEventArgs
e)
{
    if (e.Row.RowType == DataControlRowType.DataRow)
    {
        total += Convert.ToDecimal(DataBinder.Eval(e.Row.DataItem,
"TOTAL"));
    }
    if (e.Row.RowType == DataControlRowType.Footer)
    {
        Label Label2 = (Label)e.Row.FindControl("Label4");
        e.Row.Cells[3].Text = "Total:";
        e.Row.Cells[4].Text = total.ToString();
        e.Row.Cells[4].HorizontalAlign = HorizontalAlign.Right;
        e.Row.Font.Bold = true;
        txtsubtotal.Text =Convert.ToString(total);
    }
}

//private void guardar()
//{{

//    MySqlDataAdapter Adp = new MySqlDataAdapter("SELECT SUM(
(prod_precio * pedido_cantidad) + extras_precio ) AS total,
tbl_producto.prod_nombre, tbl_usuario.usuario_nombre, tbl_pedido.pedido_id,
tbl_pedido.pedido_cantidad, tbl_producto.prod_precio,
tbl_pedido.deta_observacion, tbl_pedido.reserva_id, tbl_extras.extras_precio,
tbl_extras.extras_descripcion FROM tbl_pedido INNER JOIN tbl_producto ON
tbl_pedido.prod_id = tbl_producto.prod_id INNER JOIN tbl_reservacion ON
tbl_pedido.reserva_id = tbl_reservacion.reserva_id INNER JOIN tbl_usuario ON
tbl_reservacion.usuario_id = tbl_usuario.usuario_id INNER JOIN tbl_extras ON
tbl_pedido.extras_id = tbl_extras.extras_id WHERE tbl_usuario.usuario_id='" +
txtced.Text + "'GROUP BY pedido_id", conexion);
//    DataSet ds = new DataSet();
//    Adp.Fill(ds);
//    GridView1.DataSource = ds;
//    GridView1.DataBind();

//}}
protected void GridView1_RowDataBound1(object sender,
GridViewRowEventArgs e)
{
    if (e.Row.RowType == DataControlRowType.DataRow)
```



```
        {
            total += Convert.ToDecimal(DataBinder.Eval(e.Row.DataItem,
"TOTAL"));
        }
        if (e.Row.RowType == DataControlRowType.Footer)
        {
            Label Label2 = (Label)e.Row.FindControl("Label4");
            e.Row.Cells[3].Text = "Total:";
            e.Row.Cells[4].Text = total.ToString();
            e.Row.Cells[4].HorizontalAlign = HorizontalAlign.Right;
            e.Row.Font.Bold = true;
            txtsubtotal.Text = Convert.ToString(total);
            int iva =Convert.ToInt32("0.12");
            txttotal.Text =Convert.ToString(Convert.ToInt32(txtsubtotal.Text)
* iva);
        }
    }
}
```

A.07.08 Kardex

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Tbl_Datos;
using Logica_Negocio;
namespace Facturacion_Inventario
{
    public partial class FACTURAA : System.Web.UI.Page
    {
        bool existe;
        private DateTime fechasisistema;
        private string fechasisistema1;
        private int codubi;
        private decimal total = 0;

        Logica_Proveedor logpro = new Logica_Proveedor();
        Logica_OrednAdquisicion logob = new Logica_OrednAdquisicion();
        Logica_Kardex kar = new Logica_Kardex();
        private tbl_OrdenAdquisicion _reinfo = new tbl_OrdenAdquisicion();

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                CargarProveedor();

                return;
            }
        }
    }
}
```



```
// conectar();
}

protected void Button1_Click(object sender, EventArgs e)
{
    //if()
    //{

    entradas();
    //salidas();
    //dos();
    //}
    //else
    //{

    //}
}
private void CargarProveedor()
{
    List<tbl_ObtenerProveedor> ubilista = new
List<tbl_ObtenerProveedor>();
    ubilista = logpro.Obtenernombreproveedor();
    ubilista.Insert(0, new tbl_ObtenerProveedor() { PROVEE_NOMBRE =
"(Escoja el proveedor)" });
    DropDownList1.DataSource = ubilista;
    DropDownList1.DataTextField = "PROVEE_NOMBRE";
    DropDownList1.DataValueField = "PROVEE_ID";
    DropDownList1.DataBind();
}

protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)
{
    try
    {
        codubi = int.Parse(DropDownList1.SelectedValue);
        List<tbl_producto_Nombre> pro = new List<tbl_producto_Nombre>();

        pro = Logica_Producto_Nombre._lista(codubi);
        pro.Insert(0, new tbl_producto_Nombre() { PROD_DESCRIPCION =
"(articulo)" });
        DropDownList2.DataSource = pro;
        DropDownList2.DataTextField = "PROD_DESCRIPCION";
        DropDownList2.DataValueField = "PROD_ID";
        DropDownList2.DataBind();
    }
    catch (Exception)
    {

        throw;
    }
}
private void entradas()
{
    hfproducto.Value = DropDownList2.SelectedValue;
}
```



```
        GridView1.DataSource =  
kar.ObtenerKardex(Convert.ToInt32(hfproducto.Value));  
        GridView1.DataBind();  
  
    }  
  
    protected void GridView1_RowDataBound(object sender, GridViewRowEventArgs  
e)  
    {  
  
        protected void Button2_Click(object sender, EventArgs e)  
        {  
            Response.Redirect("Administrador.aspx", true);  
        }  
    }  
}
```



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE ANÁLISIS DE SISTEMAS

CONTROL DE INVENTARIO DE BODEGA Y FACTURACIÓN MEDIANTE UNA
APLICACIÓN WEB PARA LA EMPRESA OCÉANO PACÍFICO

MANUAL DE INSTALACIÓN

Autor: Alex Darío Cuichán Apunte.

Tutor: Lic. Patricia Garzón Msc.

Quito, Octubre 2014

INDICE MANUAL DE HERRAMIENTAS

Título	pagina
Figura 1: Instalación de WampServer.....	106
Figura 2: Términos de Licencia de WampServer	106
Figura 3: Directorio de Instalación de WampServer.....	107
Figura 4: Accesos Directos de WampServer	107
Figura 5: Install de WampServer	108
Figura 6: Procesos de Instalación de WampServer.....	108
Figura 7: Explorador de Firefox por defecto para WampServer.....	109
Figura 8: Explorador de Windows como browser para el WampServer.....	109
Figura 9: Servidor SMTP y un Email de WampServer.....	110
Figura 10: Completando la Instalación de WampServer.	110
Figura 11: Icono del WampServer en la Barra de Tareas	111
Figura 12: Cambio de Idioma en el WampServer.....	111
Figura 13: Inicio del Servicio de WampServer.....	111
Figura 14: Pantalla de Presentación de WampServer	112
Figura 15: Cambio de Puerto	112
Figura 16: Texto ID	113
Figura 17: Cambio por puerto 82.....	113
Figura 18: Cambio en localhost	113
Figura 19: Cambio texto de Puerto	114
Figura 20: Explorador	114
Figura 21: inicio instalación.....	115
Figura 22: Asistente de instalación	116
Figura 23: términos de licencia.....	116
Figura 24: ruta instalación	116
Figura 25: lenguaje de programación.....	117
Figura 26: proceso de instalación.....	118
Figura 27: instalación correcta.....	118
Figura 28: reiniciar equipo.....	118
Figura 29: encendido equipo.....	119
Figura 30: iniciar visual studio.....	119
Figura 31: fin instalación	120

A.08 Manual de instalación

Instalación del Servidor de Base de Datos

En el nuestro proyecto el servidor de base de datos será el Mysql para lo cual utilizaremos en WampServer.

En la pantalla de bienvenida presionamos next

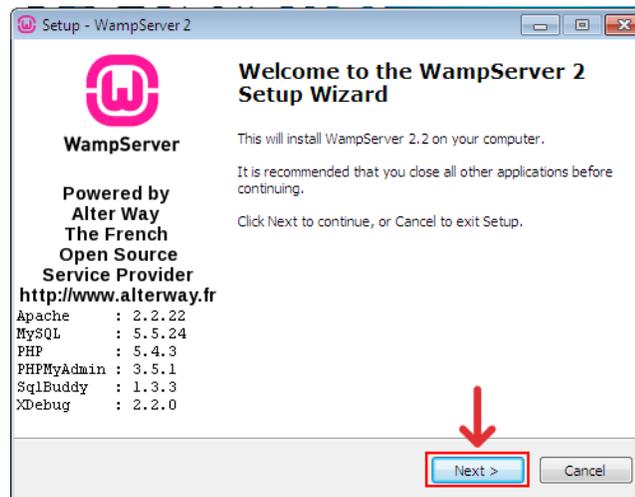


Figura 1: Instalación de WampServer

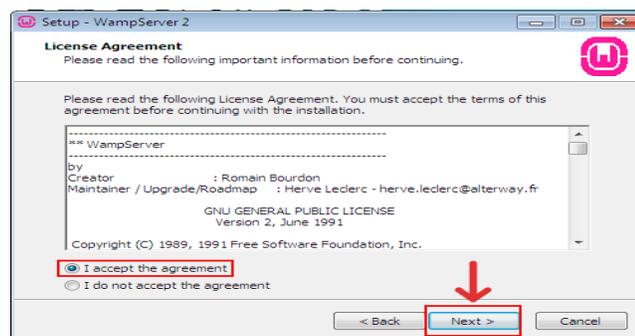


Figura 2: Términos de Licencia de WampServer

Seleccionamos el directorio de la instalación y presionamos next.

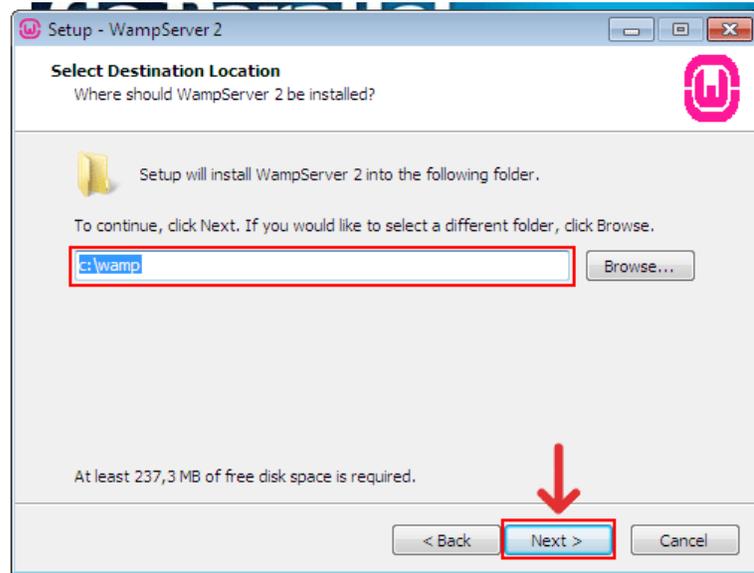


Figura 3: Directorio de Instalación de WampServer

Escogemos los accesos directos deseados y presionamos next.

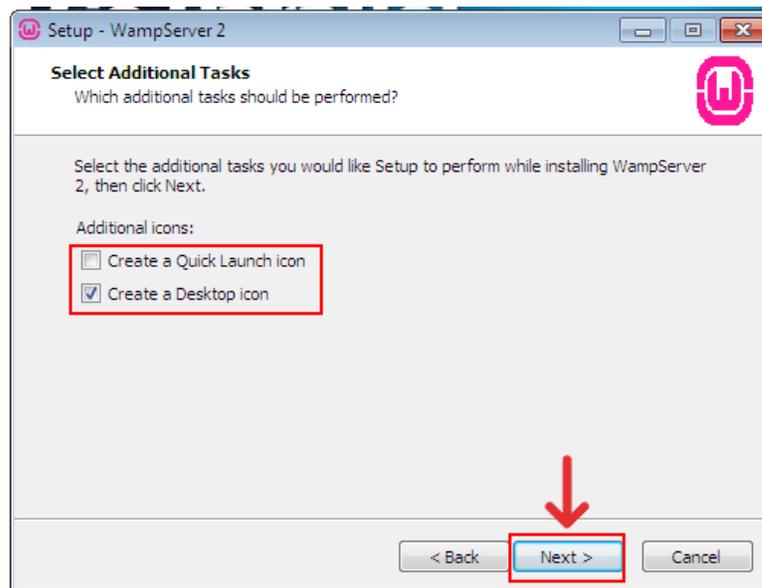


Figura 4: Accesos Directos de WampServer

Presionamos Install

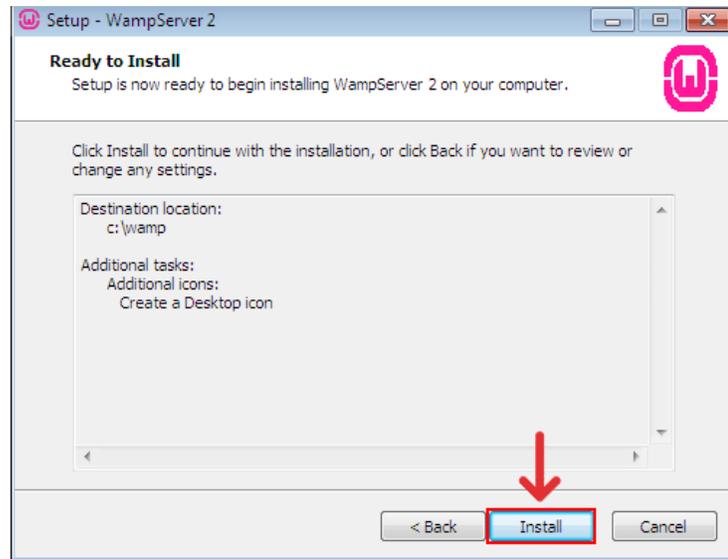


Figura 5: Install de WampServer

Esperamos a que el proceso de instalación finalice.

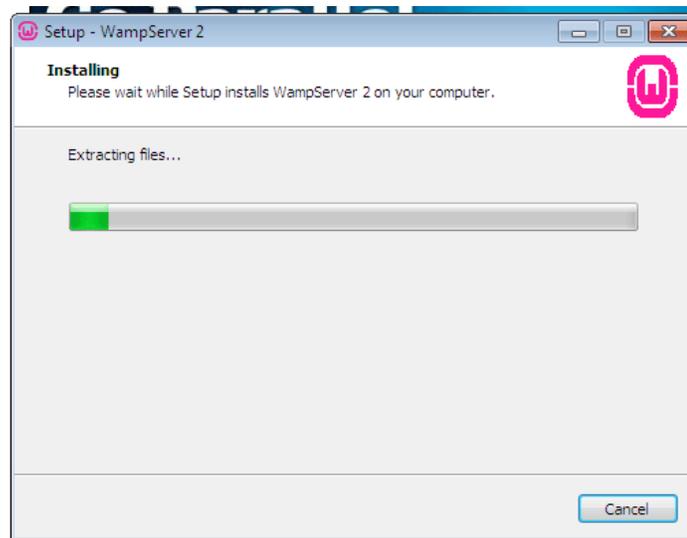


Figura 6: Procesos de Instalación de WampServer

Si queremos que nuestro Firefox sea el explorador web por defecto de WampServer presionamos Si caso contrario No.

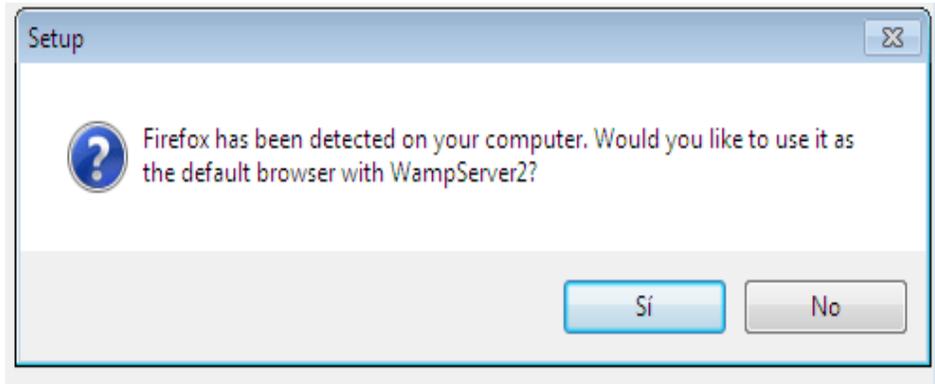


Figura 7: Explorador de Firefox por defecto para WampServer

En caso de que no encuentre ningún explorador browser lo pondrá como por defecto el explorer.exe, simplemente presionamos en abrir.

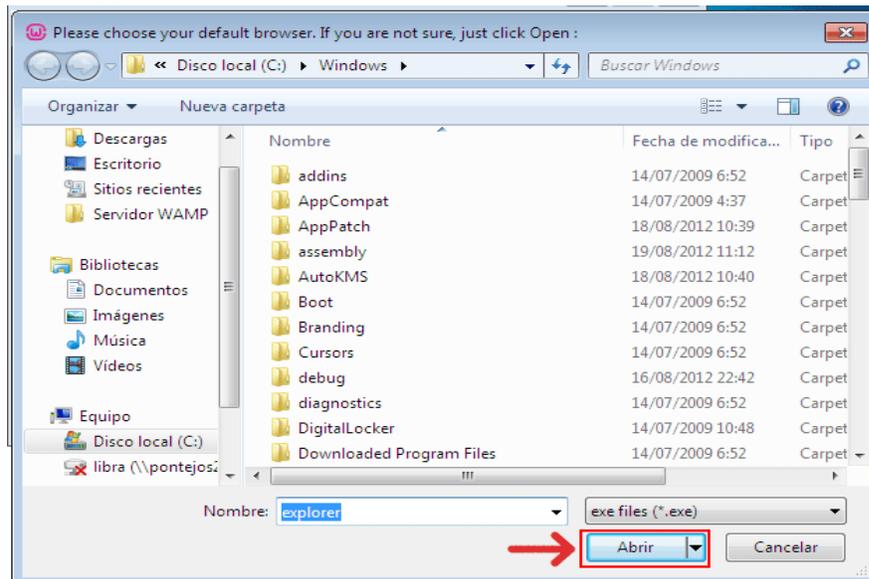


Figura 8: Explorador de Windows como browser para el WampServer

Digitamos el nombre del servidor SMTP y el Email si nuestro servidor es local dejamos los campos tal como nos muestra y presionamos next.

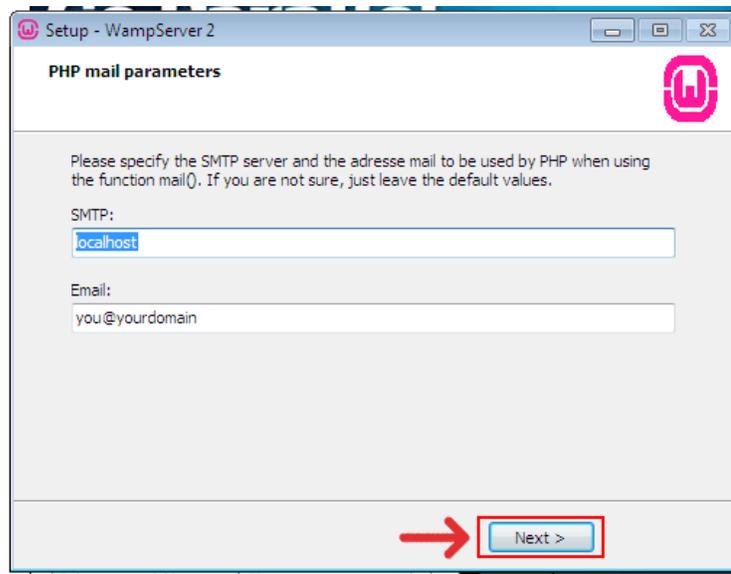


Figura 9: Servidor SMTP y un Email de WampServer

Una vez completada la instalación activamos el inicio automático y presionamos finish.

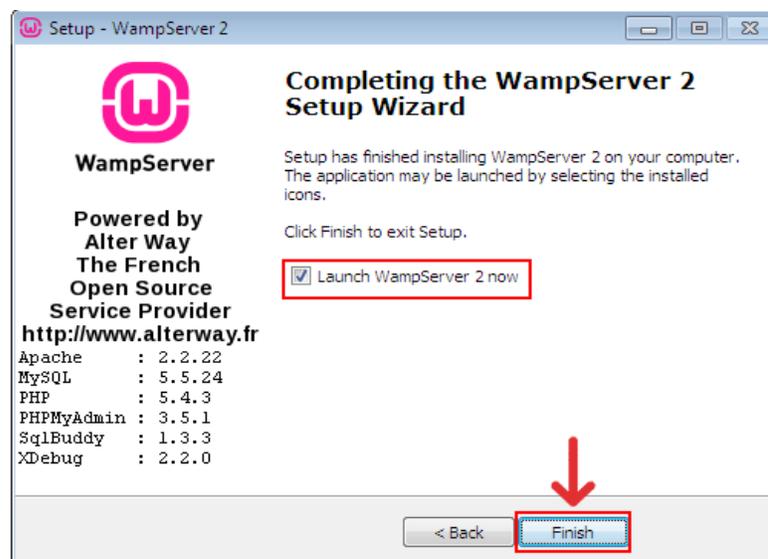


Figura 10: Completando la Instalación de WampServer.



Figura 11: Icono del WampServer en la Barra de Tareas

Para cambiar el idioma nos dirigimos a Language y luego escogemos en idioma deseado.

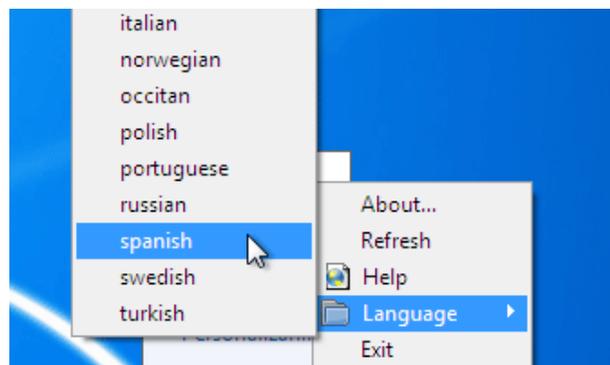


Figura 12: Cambio de Idioma en el WampServer

Para iniciar el servicio presionamos Encender.

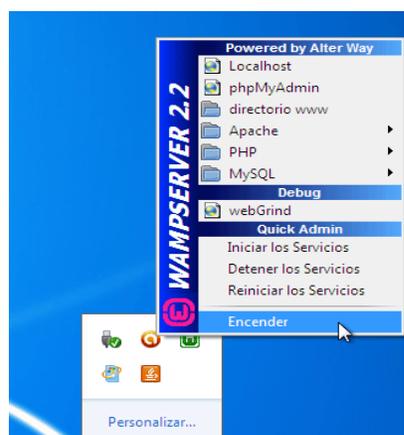


Figura 13: Inicio del Servicio de WampServer

Nos dirigimos al explorador web y escribimos localhost en la barra de direcciones y obtendremos ya nuestro WampServer instalado.

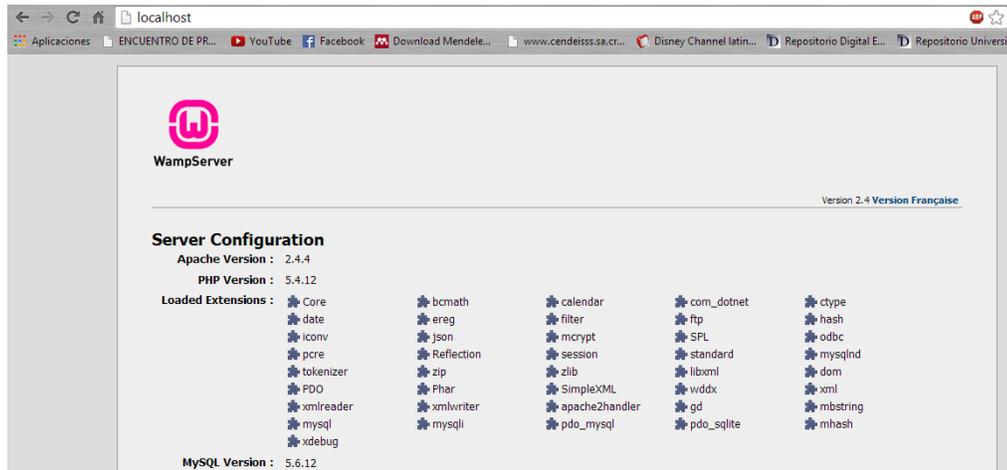


Figura 14: Pantalla de Presentación de WampServer

Cambiar Puerto en wampserver

Primeramente ejecutamos el servidor y le hacemos clic en Apache->httpd.conf tal como se muestra en la siguiente imagen:

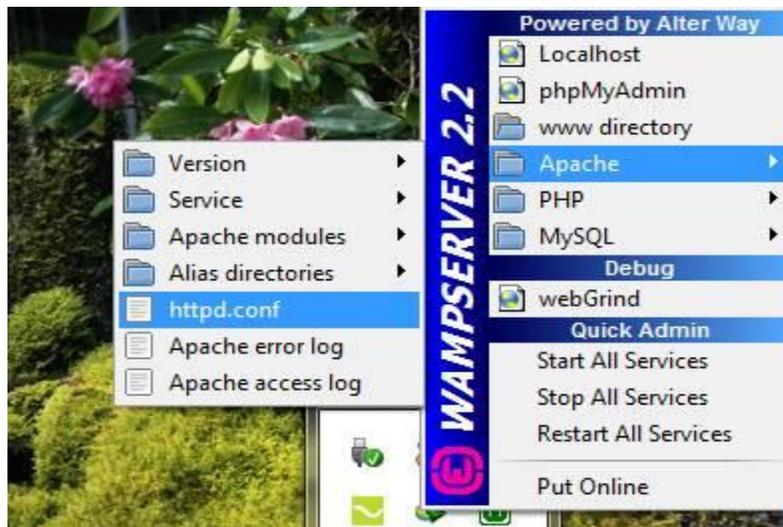
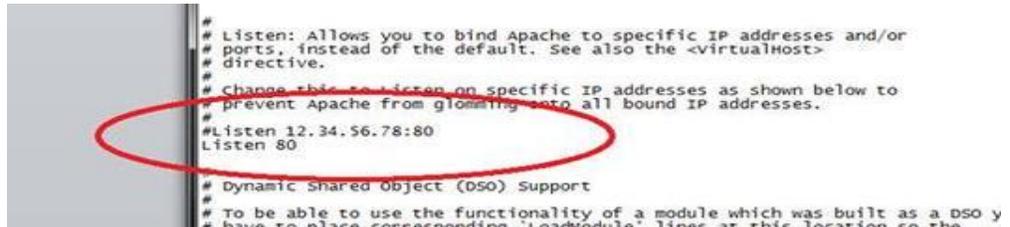


Figura 15: Cambio de Puerto

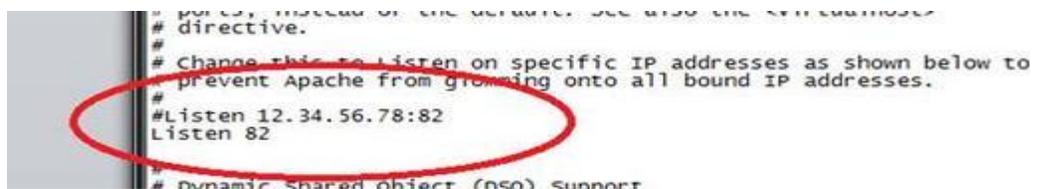
Nos abrirá un documento de texto buscamos el siguiente párrafo:



```
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <virtualHost>
# directive.
# Change this to listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#Listen 12.34.56.78:80
Listen 80
#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO y
# have to place corresponding 'LoadModule' lines at this location so the
```

Figura 16: Texto ID

Y donde dice 80 le cambiamos por 82 y debe quedarnos así:



```
# ports, instead of the default. See also the <virtualHost>
# directive.
# Change this to listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:82
Listen 82
#
# Dynamic Shared Object (DSO) Support
```

Figura 17: Cambio por puerto 82

Ahora más abajo hay otra línea la cual cambiamos de la misma forma como se muestra en la siguiente imagen cambiar de 80 a 82:

```
option: "Powered by
: "Localhost:82";
: "phpMyAdmin"; Ac
```

Figura 18: Cambio en localhost

Ahora cerramos y cuando nos pregunte guardar los cambios le decimos que sí.

Ahora nos vamos a "C:\wamp" y abrimos wampmanager.ini con notepad++ y buscamos

la línea:288, 298 y 296 y reemplazamos **localhost** por **localhost: 80** en las tres líneas de be quedarnos así:

```
Type: separator: Caption: "Powered by Alter Way"
Type: item: Caption: "Localhost:80": Action: run: FileName: "C:/Program Files/Mozilla Firefox/firefox.exe"; Parameters: "http://localhost:80/"; Glyph: 5
Type: item: Caption: "phpMyAdmin": Action: run: FileName: "C:/Program Files/Mozilla Firefox/firefox.exe"; Parameters: "http://localhost:80/phpmyadmin/"; Glyph:
Type: item: Caption: "www directory": Action: shellExecute: FileName: "c:/wamp/www/"; Glyph: 1
Type: submenu: Caption: "Apache": SubMenu: apacheMenu: Glyph: 3
Type: submenu: Caption: "PHP": SubMenu: phpMenu: Glyph: 3
Type: submenu: Caption: "MySQL": SubMenu: mysqlMenu: Glyph: 3
Type: separator: Caption: "Debug"
Type: item: Caption: "Client XDebug": Glyph: 6; Action: run: FileName: "c:/wamp/tools/xdebug.exe"
Type: item: Caption: "webGrind": Action: run: FileName: "C:/Program Files/Mozilla Firefox/firefox.exe"; Parameters: "http://localhost:80/webgrind/"; Glyph: 5
Type: separator: Caption: "Quick Admin"
```

Figura 19: Cambio texto de Puerto

Son en cuatro lugares en los cuales se debe cambiar como se muestra en la imagen siguiente, recuerda solo aumentar dos puntos y escribir 80 como se muestra en la imagen de arriba.

Después cerramos y guardamos los cambios y volvemos a reiniciar WAMP SERVER recuerda que el explorador tienes que usar ahora ya no localhost si no localhost: 80 como se muestra en la siguiente imagen:



Figura 20: Explorador

Y listo ya tienes acceso a tu servidor local.

Instalación Visual Studio 2010



Figura 21: inicio instalación: Iniciamos el setup de Visual Basic 2010 para la instalación. Pulsaremos en "Instalar Microsoft Visual Studio 2010"

Asistente de instalación Visual Basic 2010



Figura 22: Asistente de instalación: Desmarcaremos "Sí, enviar a Microsoft Corporation información sobre la instalación" (si no queremos enviar esta información) y pulsaremos "Siguiente".

Términos de Licencia



Figura 23: términos de licencia: Leeremos los términos de licencia, si estamos de acuerdo marcaremos "He leído los términos de la licencia y los acepto". Pulsaremos "Siguiente" para continuar.

Ruta de Instalación Visual Studio

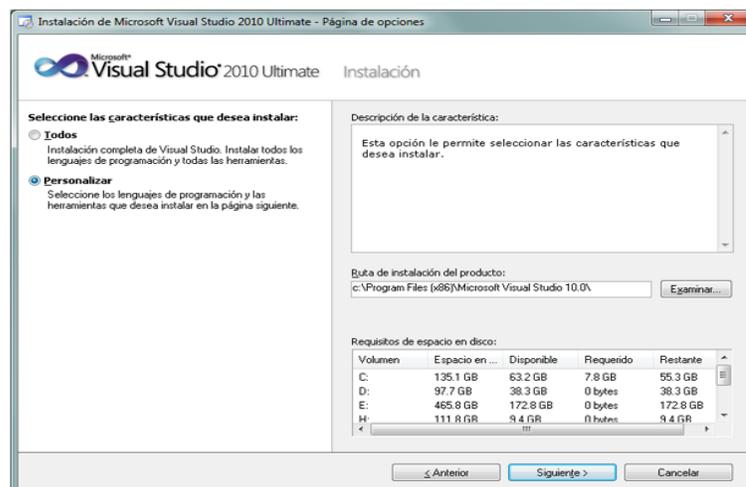


Figura 24: ruta instalación: A continuación marcaremos "Personalizar", para seleccionar la "Ruta de instalación del producto" indicaremos la unidad y carpeta de destino de la instalación.

Seleccionar Lenguajes de Programación

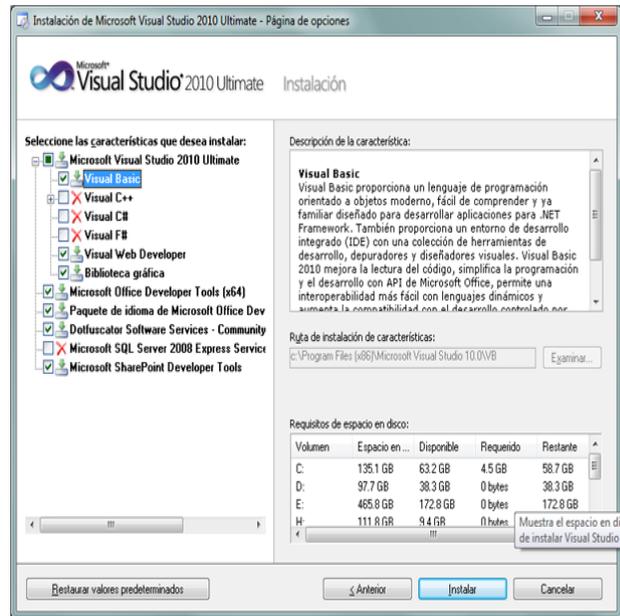


Figura 25: lenguaje de programación: Una vez seleccionadas las características a instalar pulsaremos en el botón "Instalar".

Proceso de Instalación

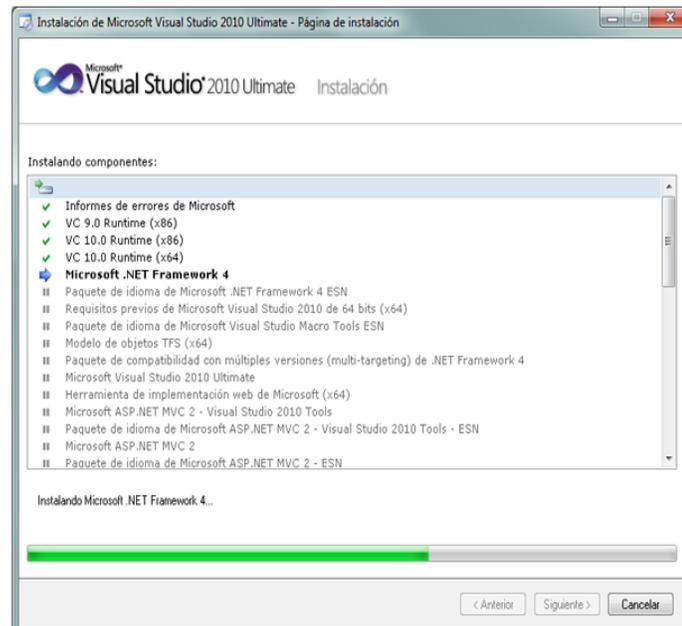


Figura 26: proceso de instalación: Esperamos que el proceso de instalación termine.

Instalación Correcta



Figura 27: instalación correcta: Tras finalizar la instalación nos mostrará la siguiente ventana, con el texto "Correcto. Se instaló Visual Studio 2010 y finalizó el programa de instalación". Pulsaremos "Finalizar"

Reiniciar Equipo

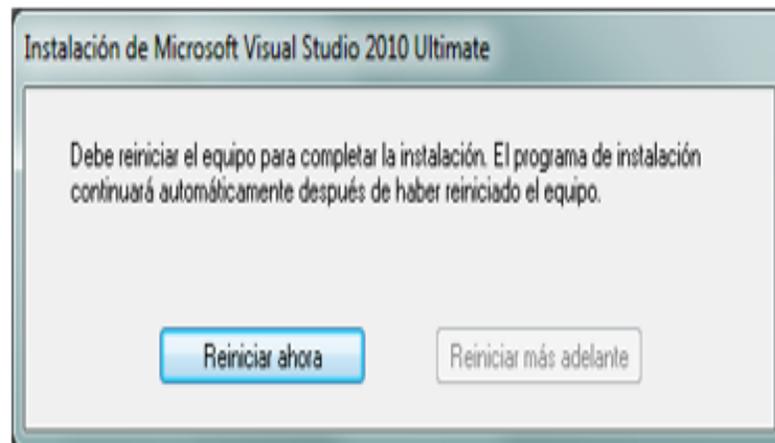


Figura 28: reiniciar equipo: Tras el proceso de instalación puede que pida reiniciar el equipo.

Luego de Haber Reiniciado el Equipo

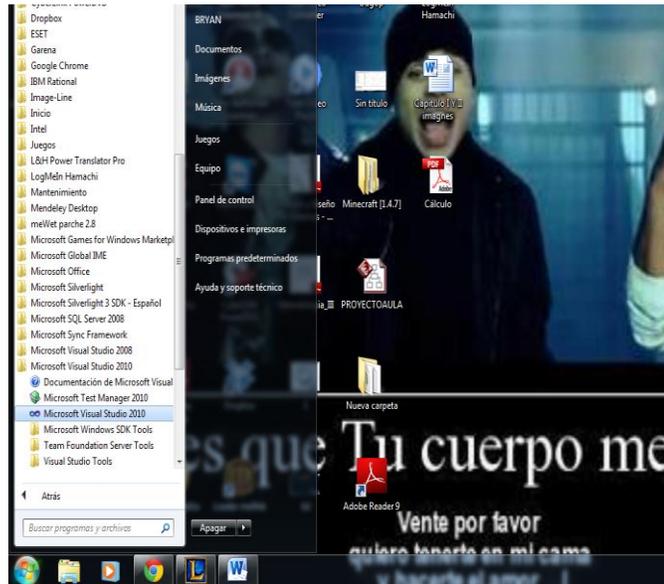


Figura 29: encendido equipo: Desde el botón "Iniciar" - "Todos los programas" podremos ver el grupo de programas "Microsoft Visual Studio 2010".

Iniciar por Primera vez Visual Studio

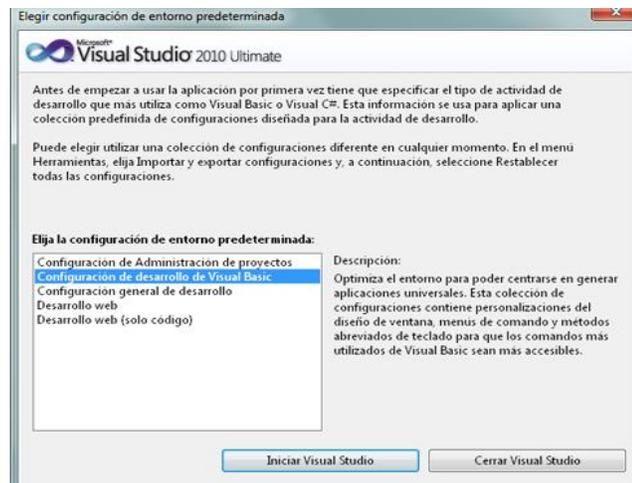


Figura 30: iniciar visual studio Tras iniciar por primera vez Microsoft Visual Studio 2010, seleccionaremos la configuración de entorno predeterminada, por ejemplo "Configuración de desarrollo de Visual Basic".

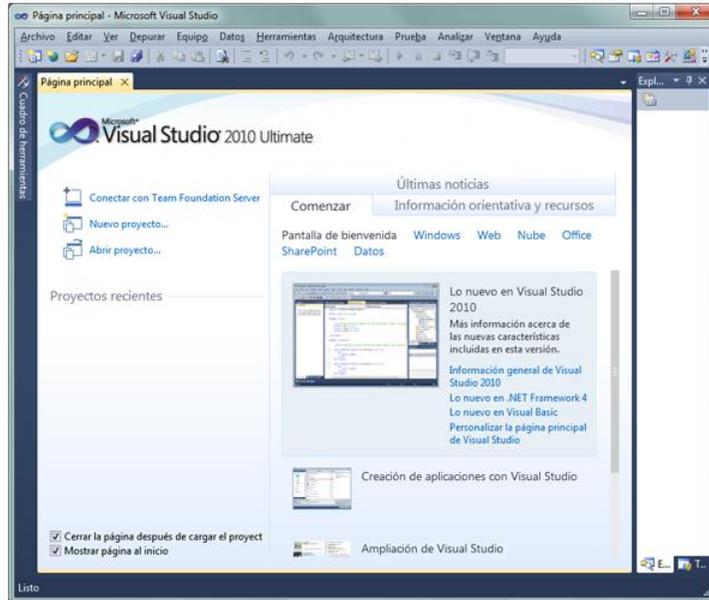


Figura 31: fin instalación: Y ya podremos disponer de este entorno de programación.

BIOGRAFIA:

<http://www.itssvirtual.edu.ec/sites/sistemas/node/34>

<http://www.monografias.com/trabajos11/conin/conin.shtml>

<http://www.lokad.com/es/definicion-control-de-inventario>

<http://www.elinventariomagazine.com/>

<http://es.wikipedia.org/wiki/Factura>

<http://www.micnt.com.ec/index.php/facturacion>

http://www.gabilos.com/cursos/curso_de_contabilidad/curso_contabilidad.html?3_el_libro_de_inventarios.html

<http://es.slideshare.net/rachela78/contabilidad-bsica>

<http://www.monografias.com/trabajos5/insof/insof.shtml>

<http://www.monografias.com/trabajos5/insof/insof.shtml>

<http://es.wikipedia.org/wiki/ASP.NET>

<http://www.visualstudio.com/es-es/visual-studio-homepage-vs.aspx>

<http://sourceforge.net/projects/wampserver/>