



INSTITUTO TECNOLÓGICO
"CORDILLERA"

CARRERA DE SISTEMAS

CONTROL DE COMPRAS DE SUMINISTROS MEDIANTE UN SISTEMA DE
CONTROL GERENCIAL PARA LA EMPRESA CASABACA.S.A

Proyecto de investigación previo a la obtención del título de tecnología Analista de
Sistemas

Autora: Argudo Leroux Nicole Alexandra

Tutor: Ing. Hugo Heredia

Quito, Noviembre 2013



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.

ArgudoLeroux Nicole Alexandra

CC 1725692048

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 **ArgudoLeroux Nicole Alexandra**, 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 compras de suministros mediante un sistema de control gerencial para la empresa CASABACA.S.A.**", 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 14 días del mes de Noviembre del dos mil trece.

f) _____
C.C. N°1725692048
CEDENTE

f) _____
Instituto Superior Tecnológico Cordillera
CESIONARIO



AGRADECIMIENTO

Gracias al Ing. Hugo Heredia por el valioso aporte profesional, técnico y humano brindado en el transcurso de este semestre en favor de la culminación de esta retadora carrera que después de mucho sacrificio voy a culminar.

Gracias a todo el personal docente, mis queridos maestros que al fin verán los frutos de sus enseñanzas.



DEDICATORIA

A mis padres por su apoyo incondicional

A mis hermanos por su calor filial



Índice General

➤ Portada	
➤ Caratula	
➤ Declaración de Aprobación Tutor y Lector	i
➤ Declaración de autoría del estudiante	ii
➤ Declaración de cesión de derechos a la institución	iii
➤ Agradecimiento	v
➤ Dedicatoria	vi
➤ Índice general	vii
➤ Índice de Tablas	xi
➤ Índice de Figuras	xii
➤ Resumen ejecutivo	xiv
➤ Abstract	xv

Capítulo I: Antecedentes

1.01 Contexto	1
1.02 Justificación	2
1.03 Matriz de Análisis de Fuerzas T	5
1.03.01 Análisis de Matriz de Fuerzas	6

Capítulo II: Análisis de Involucrados

2.01 Mapeo de Involucrados	7
2.01.01 Identificar los Involucrados	7

2.01.02 Agrupar por Características Similares	8
2.02 Matriz de Análisis de Involucrados	10
2.02.02 Análisis de Matriz de Involucrados	10

Capítulo III: Problemas y Objetivos

3.01 Árbol de Problemas	11
3.01.02 Análisis de Árbol de Problemas	13
3.02 Árbol de Objetivos	13
3.02.02 Análisis de Árbol de Objetivos	16

Capítulo IV: Análisis de Alternativas

4.01 Matriz de Análisis de Alternativas	17
4.01.02 Análisis de Alternativas	17
4.02 Matriz de Análisis de Impactos de Objetivos	20
4.02.02 Análisis de Matriz de Impacto de Objetivos	20
4.03.- Diagrama de estrategias	20
4.03.01 Análisis estrategias	22
4.04. Matriz Marco Lógico	22
4.05-. Análisis Impacto de objetivos	22

Capítulo V: Propuesta

5.01 Justificación de la propuesta	23
5.02 Análisis y Diseño	23

5.02.01 Diagramas de Casos de Uso y Especificaciones	23
5.02.01.01 Lista de Casos de Uso	23
5.02.01.02 Lista de Actores	24
5.02.01.03 Requerimientos Funcionales	25
5.02.01.04 Requerimientos No Funcionales	26
5.02.01.05 Restricciones	27
5.02.01.06 Diagramas de Casos de Uso	28
5.03.01.06.01 Especificaciones de Casos de Uso	29
5.02.02. Diagrama de casos de uso de realización	32
5.02.03 Diagramas de Secuencia y Colaboración	35
5.02.02.01 Diagramas de Secuencia	35
5.02.04 Diagramas de Colaboración	38
5.02.03 Diagrama de Componentes	41
5.02.04 Diagrama de Clases	41
5.03 Desarrollo	44
5.03.01 Arquitectura del sistema	44
5.03.01.01 Arquitectura Física	44
5.03.01.02 Arquitectura Lógica	47
5.03.02 Estándares de Programación	48
5.03.02.02 Estándares de Base de Datos	48
5.03.02.03 Estándares de Programación	50
5.03.02.04 Parámetros para documentar el código	52
5.03.03 Diseño Interfaz	52



5.04 Pruebas	60
5.04.01 Pruebas Funcionales	62
5.04.02 Pruebas por caso de uso	63
5.04.03 Casos de Prueba	63
Capítulo VI: Aspectos Administrativos	
6.01 Recursos	65
6.01.01 Recursos Materiales y Tecnológicos	65
6.01.02 Recurso Humano	65
6.02 Presupuesto	65
6.03 Cronograma	66
Capítulo VII: Conclusiones y Recomendaciones	
7.01 Conclusiones	67
7.02 Recomendaciones	68
Anexos: A.1	69
Anexo: A2 Manual Técnico	80
Anexo: A3 Manual Usuario	155
Bibliografía	163



Índice de Tablas

Tabla 1: Análisis de fuerzas T	5
Tabla 6 Especificación de caso de uso	29
Tabla 7 Especificación caso de uso Ingreso de suministro o producto	30
Tabla 8 Especificación caso de uso Entrega de suministro o productos	31
Tabla 9 Ejemplo del nombre de clases	49
Tabla 10 Estándares para la programación del software	50



Índice de Figuras

Figura 1	9
Figura 2:	12
Figura 3:	15
Figura 4:	19
Figura 5:	21
Figura 6	28
Figura 7	32
Figura 8	33
Figura 9	34
Figura 10	35
Figura 11	36
Figura 12	37
Figura 13	39
Figura 14	40
Figura 16	41
Figura 17	42
Figura 18	43
Figura 19	45
Figura 20	46
Figura 21	47



Figura 22	53
Figura 23	54
Figura 24	55
Figura 25	56
Figura 26	57
Figura 27	58
Figura 28	59



Resumen ejecutivo

Para incluir la tecnología apropiada y presentar planes de mejora y promover la participación del personal de una organización, para fortalecer la cooperación y el uso de la cadena de suministro, aplicando sus requerimientos de manera apropiada.

Por alentador que parezca alguna toma de decisión, para esto aumentar los niveles de satisfacción del cliente en la gestión de la cadena de suministro de la organización. Además, analizó los beneficiarios indirectos con claridad, proveedores de este grupo, el departamento financiero, que aunque su interferencia en el proyecto no es directa, su participación es de gran importancia que permite identificar su demanda, los intereses, los conflictos de esta manera establecen la asignación de todos los involucrados. Es importante tener en cuenta un seguimiento sobre la falta de control en la adquisición de suministros, una vez que el consumo sea adquirido, el suministro debe tener repotes sobre el grado de abuso y falta de control, este proyecto puede ser un mejor avance en la parte financiera porque ellos pueden revisar informes mensuales analizar esta área consume más y porque consume en exceso en el caso que se darían.

Abstract

To include the appropriate technology and to submit improvement plans promotes the participation of the staff of an organization, to strengthen cooperation and use of supply chain, applying their requirements in an appropriate manner.

By encouraging decision making, for this increase levels of customer satisfaction in the management of the Organization's supply chain. In addition discussed the indirect beneficiaries with clarity, found in this group suppliers, the financial Department, which although its interference in the project is not direct, their participation is of great importance allowing to identify their demand, interests, conflicts in this way establish the mapping of all involved.

It is important to have a follow-up on the lack of control in the procurement of supplies, once consumption should raise some reports about the extent of abuse and lack of control on purchases, this project can be a best advance in the financial part because they may review monthly reports analyzing this area consumes more and because it consumes in excess in the case that would be given.

Capítulo I: Antecedentes

1.01 Contexto

La Cadena productiva es un conjunto de agentes y actividades económicas que intervienen en un proceso productivo, la cadena de suministros se refieren a las actividades y procesos que involucran clientes y proveedores para que la mercancía sea producida y distribuida en las cantidades correctas.

La empresa CASABACA.S.A ubicada en la provincia de Pichincha, en la Zona 9 dentro del plan Nacional de Desarrollo del Buen Vivir desarrolla su actividad comercial en el sector Automotriz, siendo la primera empresa en mantener una oficina en París para importaciones y exportación de mercadería.

Dentro de su actividad comercial se encuentra no sólo la venta y compra, sino que también se ocupa de los seguros de vehículos, venta de accesorios y repuestos, mantenimiento vehicular entre otros.

Para CASABACA.S.A, el cliente es lo primero, por lo que continuarán trabajando arduamente para seguir cumpliendo con el mejor servicio y la mejor atención del mercado.

Las estrategias de mercado obligan a las organizaciones públicas y privadas cerrar sus negocios directamente con sus proveedores y acreedores que le permitan obtener costos mínimos y poder competir, mejorando sus capacidades.

El manejo inadecuado en la cadena de suministros por parte de los diferentes entes de la organización a los cuales se asignan y distribuyen los bienes, no es eficaz producir altos costos, es necesario añadir el desconocimiento de las autoridades y directivos que muy pocas ocasiones conocen el proceso de asignación de suministros a las diferentes áreas.

Con estos antecedentes y en función a la disimilitud que de alguna manera requieren del sustento tecnológico para mejorar y fortalecer su capacidad gerencial es indispensable proporcionar una herramienta que facilite y garantice el desarrollo de sus actividades reduciendo la problemática, reduciendo el tiempo de asignación y control de suministros a las diferentes dependencias de la institución.

1.02. Justificación

El presente proyecto pretende solucionar el manejo inadecuado en la cadena de suministros de la empresa mediante la inclusión de tecnología que permita mejorar la organización y administración de los insumos y artículos.

Por otro lado se observa la falta de estrategias de la empresa que por sus características resulte fácil la adquisición y distribución. Es decir, la inclusión de herramienta de apoyo para la cooperación y difusión en la utilización de los suministros dentro de la empresa.

Al incluir la tecnología apropiada y presentar planes de mejora promueve la participación del personal de una organización, que fortalezca la cooperación y utilización de la cadena de suministros, solicitando sus requerimientos de un modo adecuado.

Según Mario Bonilla, ingeniero ambiental, todas las industrias deben empezar a reciclar. "Esto ya no es una opción. Las condiciones ambientales mundiales son preocupantes.

No podemos continuar indiferentes". Según este especialista, el 90% de la producción mundial de pulpa para papel se basa en la madera. "El consumo mundial por año se calcula en unos 640 millones de metros cúbicos.

Esto equivale a un bosque de más de 2 millones de hectáreas. La situación es realmente preocupante” (“El papel reciclado se transforma en lápices, en Ambato,” n.d.) por mencionar un ejemplo del mal uso que se hace de uno de los elementos de la cadena de suministro.

Por lo anterior la realización del proyecto se justifica claramente por la importancia que tiene en la toma de decisiones de las organizaciones, utilizando un pensamiento sistémico para generar una administración eficiente, obteniendo un impacto en las diferentes áreas de la organización y focalizando una integración y competitividad en el mercado.



1.03 Matriz de Análisis de fuerzas T

Tabla 1

Análisis de fuerzas T

SITUACIÓN EMPEORADA	SITUACIÓN ACTUAL				SITUACIÓN MEJORADA
COSTOS ALTOS ASOCIADOS A SUMINISTROS	MANEJO INADECUADO DE LA CADENA DE SUMINISTROS				INCREMENTAR NIVELES DE SATISFACCIÓN EN EL MANEJO DE SUMINISTROS
FUERZAS IMPULSORAS	I	PC	I	PC	FUERZAS BLOQUEADORAS
1. Mejora en la cooperación funcional	4	4	4	2	1. Procesos poco estructurado
2. Reducción amplia de costos	4	3	3	1	2. Incertidumbres
3. Procedimientos y manuales estructurados	4	3	3	3	3. Costos elevados
4. Mejora en la distribución y control de suministros	4	4	4	2	4. Baja satisfacción del cliente
5. Aumentar la capacidad de toma de decisiones	4	4	4	1	5. Baja cooperación funcional

Nota: I= Intensidad; PC= Potencial de Cambio

1.03.01 Análisis de la Matriz de Fuerzas.

Luego de haber realizado el análisis de fuerzas T, es importante que la organización CASABACA.S.A adopte líneas estratégicas para mejorar la cooperación funcional; reducir los costos. Estableciendo procedimiento y manuales estructurados con el propósito de distribuir y controlar la cadena de suministros.

Fomentando una toma de decisiones, para de esta incrementar los niveles de satisfacción de los clientes en el manejo de la cadena de suministros de la organización de CASABACA.S.A

Capítulo II: Análisis de Involucrados

2.01 Mapeo de involucrados

2.01.01 Identificar los involucrados.

Para poder establecer el mapeo de los involucrados sobre el problema del mal manejo de la cadena de suministros se realizó un taller con la participación de un grupo de personas que se seleccionaron de cada departamento de la organización, los cuales fueron clasificando a los involucrados de acuerdo con dos criterios, el carácter interno de la organización, y la función que desempeña dentro de la cadena de suministro.

Entre los beneficiarios directos del proyecto encontramos al Departamento de Compras, Clientes Internos, que forman parte de la organización, pues sus demandas e intereses que cada uno de ellos tienen afecta directamente al proyectos, así como los posibles impactos que tendría la ejecución del proyectos sobre dichos intereses.

Además se analizó los beneficiarios indirectos con claridad, encontrándose en este grupo los Proveedores, el Departamento Financiero, que si bien su injerencia en el proyecto no es directa, su participación es de mucha importancia permitiendo Identificar su demanda, intereses, conflictos para de esta forma establecer el mapeo de todos los involucrados.

Del resultado del trabajo realizado se determinó que el grupo o grupos objetivos que requiere mayor énfasis y trabajo son el Departamento de Compras y los Clientes Internos que serán calificados considerando su interés y participación en el proyecto.

Como Área Financiera se ha agrupado los siguientes: Gerencia, Departamento de adquisiciones, Departamento de Cobranzas, como Clientes Internos están considerados: Departamento de Sistemas, Proyectos, Repuestos y venta de Vehículos, como Proveedores agrupamos a Dilipa, Infoline, Super Paco y el Departamento de Compras en el todos los coordinadores, así como se muestra a continuación.

2.01.02 Agruparlos por Características Similares.

Departamento Financiero: Gerencia, Departamento Financiero y Departamento Cobranzas.

Departamento Compras: Coordinadores.

Proveedores: Santa María, Súper Paco y Dilipa.

Clientes internos: Departamento de Sistemas, Departamento Proyecto, Departamento Repuestos, área de ventas de vehículos, etc.



Figura 1

Representa el manejo inadecuado de la cadena de suministros tanto involucrados directa o indirectamente

2.02 Matriz de Análisis de Involucrados.

Tabla 2

Análisis de Involucrados

Ver Anexo 66

2.03 Análisis de Involucrados

Al determinar las estrategias tanto de complejidad como de factibilidad, la empresa como tal carece de medidas de control que a su vez las definimos como actores.

Cada actor conlleva a un problema, a un recurso, a un interés o un conflicto y por ultimo a establecer las respectivas alianzas estratégicas, es decir todo el análisis conlleva a un nivel de pérdida económica, no es un alto porcentaje pero con el pasar del tiempo lo llegaría hacer.



Capítulo III: Árbol de problemas y objetivos

3.01 Árbol de Problemas

La identificación y el análisis de problemas es el primer paso para la identificación de los objetivos que regirán en el proyecto. Además, este tipo de análisis es el que se utiliza para la identificación de alternativas de solución a un determinado problema, y constituye, por tanto, la primera fase del proceso de planificación del proyecto de inversión ya sean de interés público o privado

El árbol de problemas es una ayuda importante para entender la problemática a resolver. En él se expresan, en encadenamiento tipo causa/efecto, Ofrece una visión sistémica, permitiendo desglosar o desagregar procesos de pensamientos lógicos de lo general a lo particular las condiciones negativas percibidas por los involucrados en relación con el problema en cuestión.

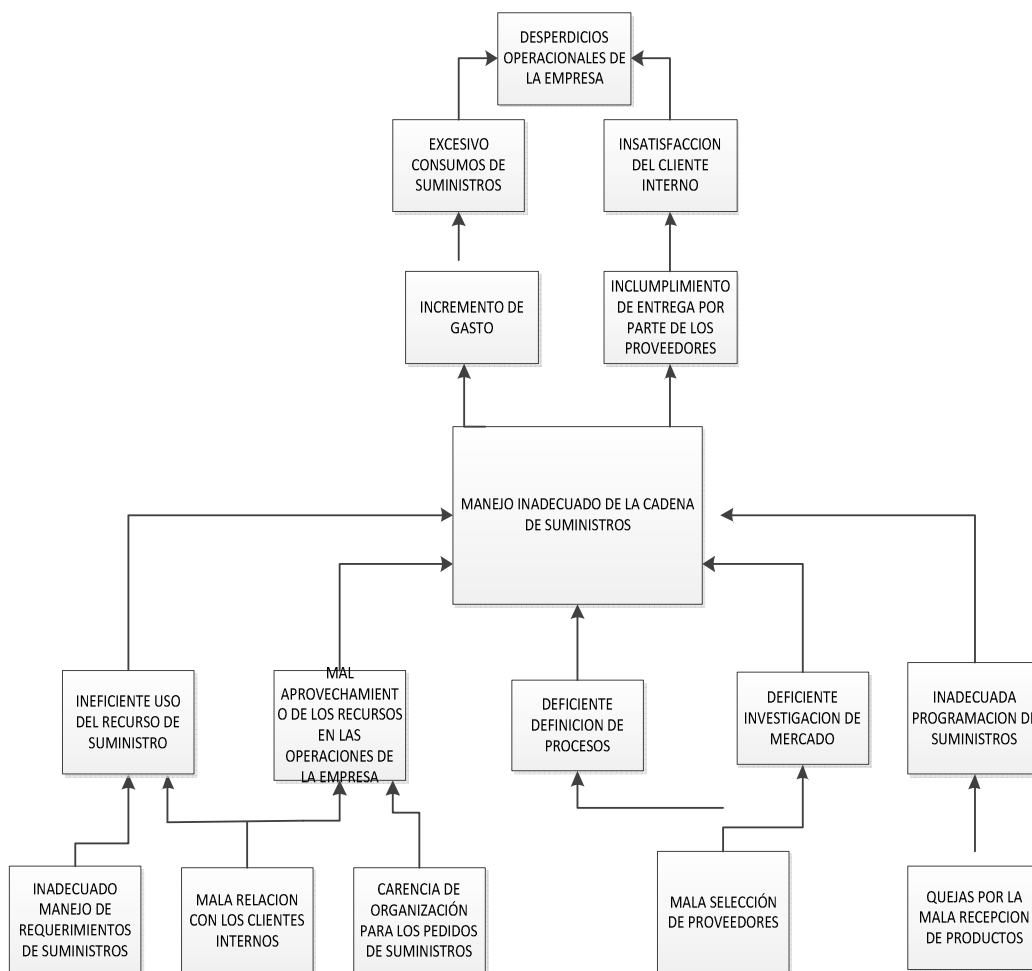


Figura 2

Árbol de problemas representa el manejo erróneo que tiene el usuario y la ...
insatisfacción de la empresa y del empleado

3.01.01 Análisis árbol de problemas.

Para el debido análisis se toma en cuenta el problema central Manejo inadecuado de la cadena de suministros, teniendo claro cuál es el problema principal partimos de las causas (quejas, carencias, manejos inadecuados, etc.) volvemos al problema principal y generamos el análisis correspondiente a la suma de efectos (excesivo, incumplimiento, desperdicios, etc.)

Confirmado el mencionado encadenamiento causa/efecto, se ordenan los problemas principales permitiendo identificar el conjunto de problemas sobre el cual estará enfocado a los objetivos del proyecto. Esta clarificación de la cadena de problemas permite mejorar el proyecto en su ejecución.

3.02 Árbol de Objetivos.

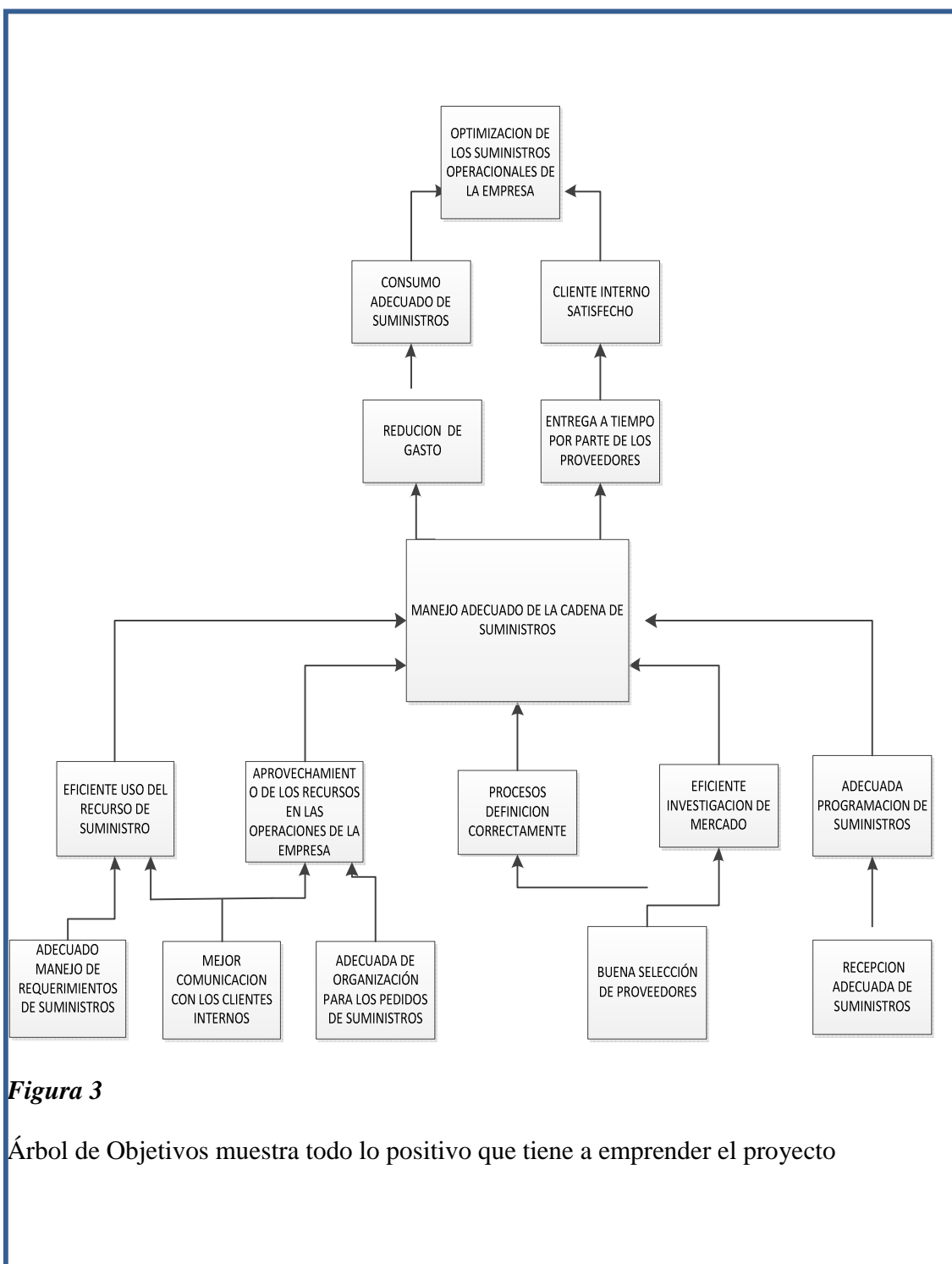
Los problemas de desarrollo identificados en el árbol de problemas se convierten, como soluciones, en objetivos del proyecto como parte de la etapa inicial de diseñar una respuesta.

El propósito es utilizar el árbol de problemas para identificar las posibles soluciones al problema, las cuales podrían ser expresadas como manifestaciones contrarias del mismo. Esto da lugar a la conversión del árbol de problemas en un árbol de objetivos: la secuencia encadenada de abajo hacia arriba de causas-efectos se transforma en un flujo interdependiente de medios-fines. En un árbol de objetivos. Los medios fundamentales se especifican en el nivel inferior: constituyen las raíces del árbol.



Los fines se especifican en la parte superior: son las ramas del árbol. Más propiamente son los objetivos del posible proyecto.

El árbol de objetivos es un procedimiento metodológico que permite: Describir la situación futura que prevalecerá una vez resuelto los problemas, identificar y clasificar los objetivos por orden de importancia y visualizar en un diagrama las relaciones medios-fines.





3.02 .02 Análisis Árbol de Objetivos.

Mediante el análisis de objetivos se obtiene un modelo que explica el proceso que es necesario seguir para lograr un determinado cambio en la Empresa que resuelva el problema central. Siendo esencial este análisis para fomentar el cambio con un respaldo tomando en cuenta diferentes parámetros e incorporar nuevas alternativas de control y manejo de suministros.

Teniendo en cuenta el principal objetivo manejo adecuado de la cadena de suministros, tomamos como punto de partida técnicas para identificar el curso distinto de acción, que pueden llegar a ser estrategias de proyecto, evaluar estas posibles opciones y determinar la estrategia a ser adoptada.

Excluir objetivos, esto nos ayuda a quitar opciones que son descartables o no factibles para llegar a seleccionar las alternativas quiere decir las mejores estrategias del proyecto como tal.

Capítulo IV: Análisis de Alternativas

4.01 Matriz de análisis de Alternativas

Se establece una Matriz de análisis de la disponibilidad de los recursos humanos, financieros y técnicos para complementar el alcance del proyecto

Tabla 3.

Análisis de Alternativas

Ver Anexo 70

4.01.01 Análisis de Alternativas

Hemos concluido que es factible elaborar el siguiente proyecto, la factibilidad del cliente interno como de proveedores y los diferentes escenarios son fundamentales para la ejecución de dicha propuesta ya que generando un resultado positivo hemos creado varios parámetros sólidos en base a nuestro objetivo general.

Al momento analizamos el impacto positivo direccionándonos hacia los objetivos encontramos la factibilidad del proyecto en lo técnico, financiero, social y político. Los valores se determinan entre medio, bajo y alto la balanza es de 0 a 18 de 0-6 es bajo de 7-13 medio y 14-18 alto. Generando una estabilidad para determinar si es posible culminar o no el proyecto

4.02 Matriz de Análisis de Impacto de los Objetivos

Esta Matriz analiza los diferentes componentes que se va a ejecutar en el sistema
Analizar el aspecto de eventual dependencia tecnológica que se deriva de la
transferencia de equipos.

Es conveniente indicar las diferentes alternativas que se han contemplado y exponer
las razones por las que se ha llegado a la opción definitiva.

Obteniendo en esta matriz las metas, estrategias, propósitos, indicadores de
propósitos fines y sus indicadores de finalidad con respecto al problema central del
proyecto del cual sacaremos lo positivo a realizarse.

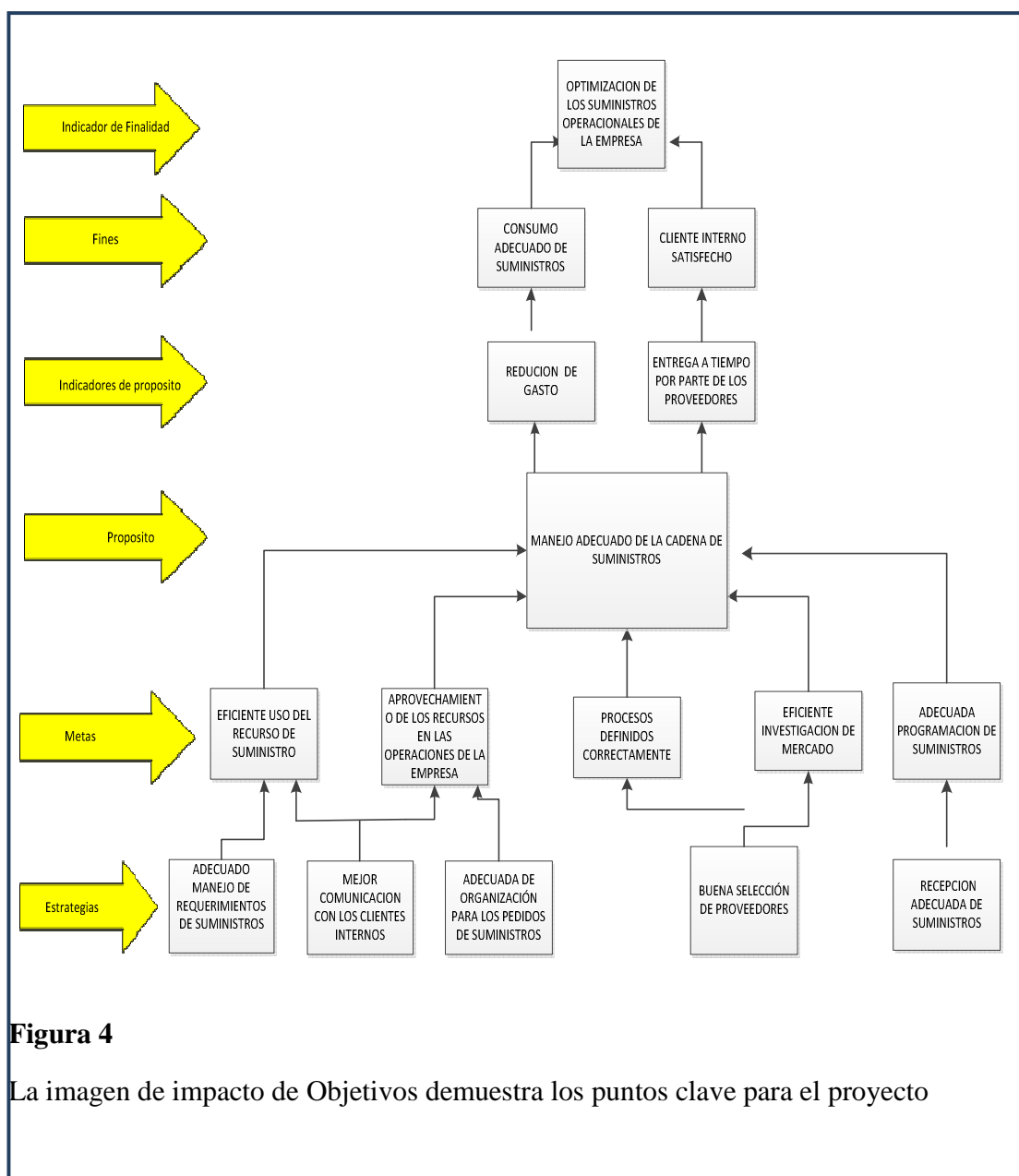


Figura 4

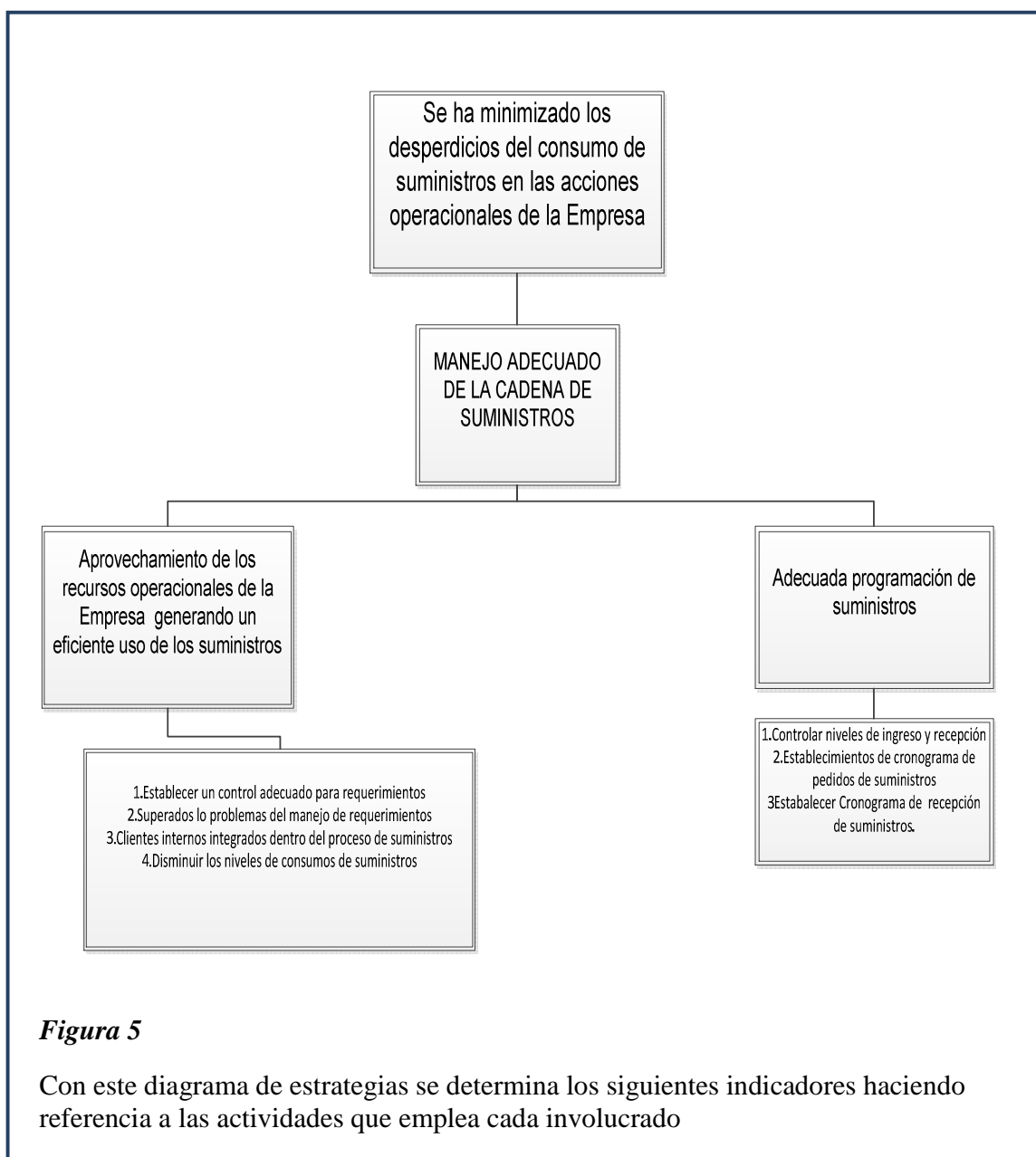
La imagen de impacto de Objetivos demuestra los puntos clave para el proyecto

4.02.02 Análisis de Impacto de los Objetivos.-

Este análisis define las interdependencias específicas de las funciones de negocio y los vínculos clave existentes entre los recursos de TI. Crea una panorámica de sus vulnerabilidades y requisitos de la empresa.

4.03.- Diagrama de estrategias

Este diagrama especifica las estrategias para dar solución al problema central y sus diferentes patrones.





4.03.01 Análisis estrategias.-

Para determinar el diagrama de estrategias centradas en los diferentes análisis de matrices para la definición de los indicadores e identificar a las diferentes actividades para obtener un mejor enfoque con respecto al tema centra

4.04. Matriz Marco Lógico

Es un instrumento básico que facilita el diseño, la ejecución, el seguimiento y la evaluación de las acciones. Se trata de una matriz de cuatro filas por cuatro columnas con las siguientes características

Tabla 4

Matriz de Marco Lógico

Ver Anexo 73

4.05-. Análisis Impacto de objetivos

Es una herramienta para verificar la participación de los clientes internos o externos de la empresa, fortaleciendo de este modo la organización.

Tabla 5

Análisis Impacto de objetivos

Ver Anexo 75



Capítulo V: Propuesta

5.01 Justificación de la propuesta.

El Documento como tal, a continuación define los requerimientos funcionales y no funcionales del módulo de compras de suministros. De forma que nos indicará todas las funcionalidades del sistema, y de igual forma las restricciones.

El alcance se definirá con la determinación de casos de uso de suministros, con respecto a la manipulación en el ingreso de requerimientos, generando con su respectivo ingreso, modificación, eliminación e ingreso de productos, así como de la emisión de reportes e información relativa a productos de suministros.

Este sistema estará o se definirá con procesos técnicos los cuales serán viables, contempla las funcionalidades que prestará al usuario dentro del contexto que implica el uso del Sistema Automatizado de Suministros (Control de suministros de compras), el consta del siguiente módulo: Ingreso de Requerimientos, Ingreso de Usuarios e Ingreso de Productos de Suministros.

5.02. Análisis y Diseño

5.02.01 Diagrama de casos de uso y especificaciones

5.02.01.01 LISTA DE CASOS DE USO

Para el control de requerimiento de suministros, se han identificado los casos de uso descritos a continuación:

U.C.1.- Ingresar Suministros

Este icono le permite acceder al ingreso de datos del suministro y listado de suministros

U.C.2.-Ingresar Productos Suministros.

Este icono le permite acceder a los datos de la tabla de los diferentes productos que tendrá en el sistema tanto suministros de compras, como suministros de oficina

U.C.3.- Ingresar a Iconos Requerimientos de compras.

Esta opción le permite ingresar a los requerimientos mensuales puede ingresar un nuevo requerimiento o analizar el estado en que se encuentra el requerimientos

5.02.01.02 - LISTA DE ACTORES

Los actores descritos a continuación son aquellos que por su importancia en el módulo de compras, tiene una alta consecuencia en la información y serán estos los principales.

Coordinador de compras

Este actor está encargado de verificar todos los requerimientos ingresados por todos los usuarios de la empresa, verifica en el sistema el ingreso de los suministros y envía por mail el requerimiento al proveedor, luego el proveedor entrega a compras todos los insumos de cafetería y oficina, el



coordinador como tal va personalmente a entregar a cada usuario luego el usuario hace la recepción en el sistema

Usuario

Actor que ingresará su requerimiento de suministros en el sistema mensualmente pondrá cantidad de suministros que necesita, luego llegara el coordinador de compras con los insumos y en el sistema ingresara la recepción de todos los productos en el caso que llegara a faltar se ingresara en el sistema como pendiente o en proceso y se ingresa en observaciones la novedad. En caso que llegara a sobrar se ingresa la observación y se finaliza el requerimiento

Administrador

Actor que mantendrá y apoyará al sistema, para verificar si está existiendo una coherencia entre los datos ingresados de requerimientos en el módulo de compras.

5.02.01.03 Requerimientos funcionales

El sistema de compras dispondrá de los siguientes requerimientos funcionales, que se detalla a continuación:

RF01.- El Sistema de compras, permite al empleado ingresar al sistema con usuario y password para el ingreso de requerimientos.



RF02.- El Sistema de compras, proporciona al inicio de la página web la opción al usuario para ingresar requerimiento de suministro.

RF03.- El Sistema de compras, permite al usuario Administrador que ingrese datos de los productos nuevos en suministros

RF04.- El Sistema de compras, entregara información de requerimiento requerimientos ingresado en general al área de compras.

RF05.- El Sistema de compras, proporcionará información del módulo de Requerimientos a los usuarios

RF06.- El Sistema de compras, permitirá obtener reportes de los requerimientos ingresados.

RF07.- El Sistema de compras, permitirá al usuario cerrar el requerimiento al momento de la entrega del producto

5.02.01.04 Requerimientos no funcionales

NRF01.- El Sistema de compras, no requiere que el usuario ingrese valores en productos.

NRF02.- El Sistema de compras, no permite a los usuarios modificar estados del requerimiento.

NRF03.- El Sistema de compras, no permite al área de compras modificar requerimientos ingresados por cualquier usuario.

NRF04.- El Sistema de compras, no permite que el usuario valide información de otros usuarios

NRF05.- El Sistema de compras, no permite que el usuario ingresar datos en tablas de Administradores.

5.02.01.05 Restricciones

En el modelamiento del sistema de requerimiento de compras se utilizarán los diagramas de UML, lenguaje que cuenta con elementos gráficos, que describirá lo que el sistema es capaz de hacer. Para dicho modelamiento se empleará la herramienta de software Rational Rose, el cual nos provee tanto herramientas como información para poder generar los Casos de Uso y para la codificación del Sistema de suministros de compras se trabajará con Visual Studio 2010 el cual permite que el sistema sea desarrollado en 3 capas, la arquitectura utilizada será configurada en cliente servidor.

5.02.01.06 Diagramas de casos de uso

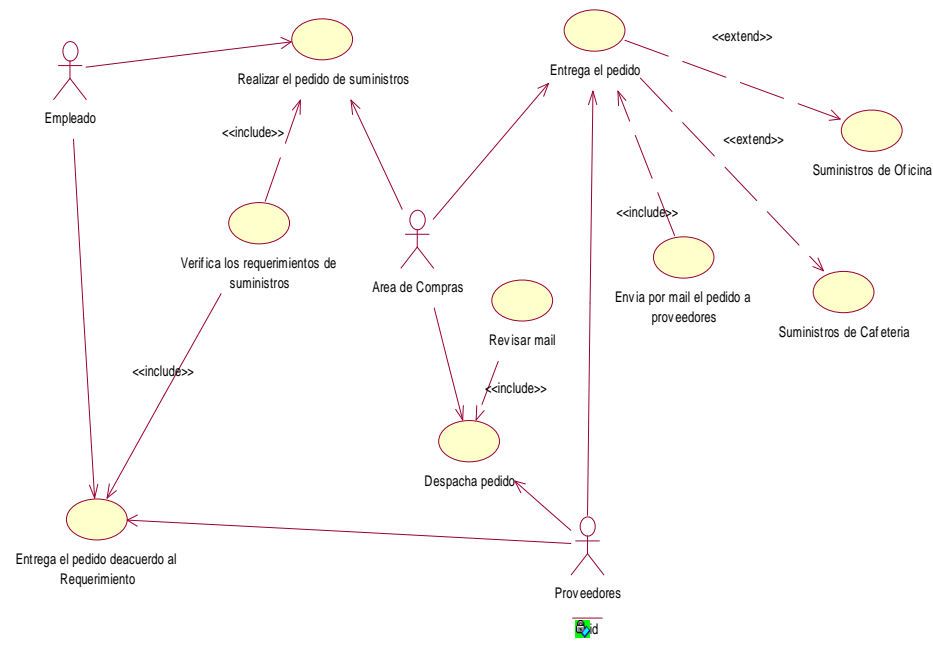


Figura 6

El modelo de casos de uso describe la funcionalidad del proceso a ser implementado en el sistema de control de suministros. Un caso de uso representa una unidad discreta de interacción entre un usuario y el sistema. Cada caso de uso tiene una descripción que describe la acción que formara parte en el sistema propuesto. Un caso de uso puede "incluir" la funcionalidad de otro caso de uso o "extender" a otro caso de uso con su propio comportamiento, como lo indica la figura.



5.02.01.06.01 Especificaciones de casos de uso

Tabla 6

Especificación de caso de uso ingreso de requerimiento.

Caso de Uso	: Ingreso del requerimiento
Id	: U.C.1
Actores	: Empleado
Precondiciones:	
1.- Ingreso a Sistema de compras/RequerimientoNuevo.	
Flujo de Eventos:	
1.- El caso de uso inicia cuando el usuario ingresa al sistema de Suministros de compras	
2.- El usuario al ingresar a requerimiento tiene la opción de pedido en esa opción ingresa un nuevo pedido y verifica el registro del pedido al guardar los datos y arroja la lista de datos de suministros.	
3.- Termina Proceso de Ingreso de Suministro	
Flujos Alterativos:	
1.- El usuario en cualquier momento puede salir de la pantalla de suministros	



Tabla 7
Especificación caso de uso Ingreso de suministro o productos

Caso de Uso	: Ingreso de Suministro o Productos
Id	: U.C.2
Actores	: Proveedores
Precondiciones:	
1.- El proveedor no tiene acceso al sistemas pero Compras envía un mail en general de todos los productos que necesita	
Flujo de Eventos:	
1.- El proveedor recibe un mail de los productos los verifica con el personal de compras los productos del requerimiento.	
2.- El caso de uso inicia cuando el coordinador ingresa/SuministrosLista” a verificar en general todos los productos	
2.- Luego entrega a cada usuario los productos (El proceso refleja pendiente).	
3.- El caso de uso termina	
Flujos Alternativos:	
1.- El usuario en cualquier momento puede salir de la pantalla requerimientoLista.	
2.- El caso de uso termina	



Tabla 8
Especificación caso de uso Entrega de suministro o productos

Caso de Uso	: Entrega de productos (Compras a Usuario)
Id	: U.C.3
Actores	: Compras Usuarios
Precondiciones:	
1.- Ingreso al sistema de compras/RequerimientoLista.	
Flujo de Eventos:	
1.- Compras envía un mail al proveedor con los requerimientos en general 2.- El caso de uso inicia cuando Compras llega a la entrega del producto el usuario ingresa al sistema para verificar los productos requeridos ingresa el estado de recibidos/RequerimientoPedido” 3.- El caso de uso termina	
Flujos Alternativos:	
1.- El usuario en cualquier momento puede salir de la pantalla requerimientoEstado 2.- El caso de uso termina	

5.02.02. Diagrama de casos de uso de realización

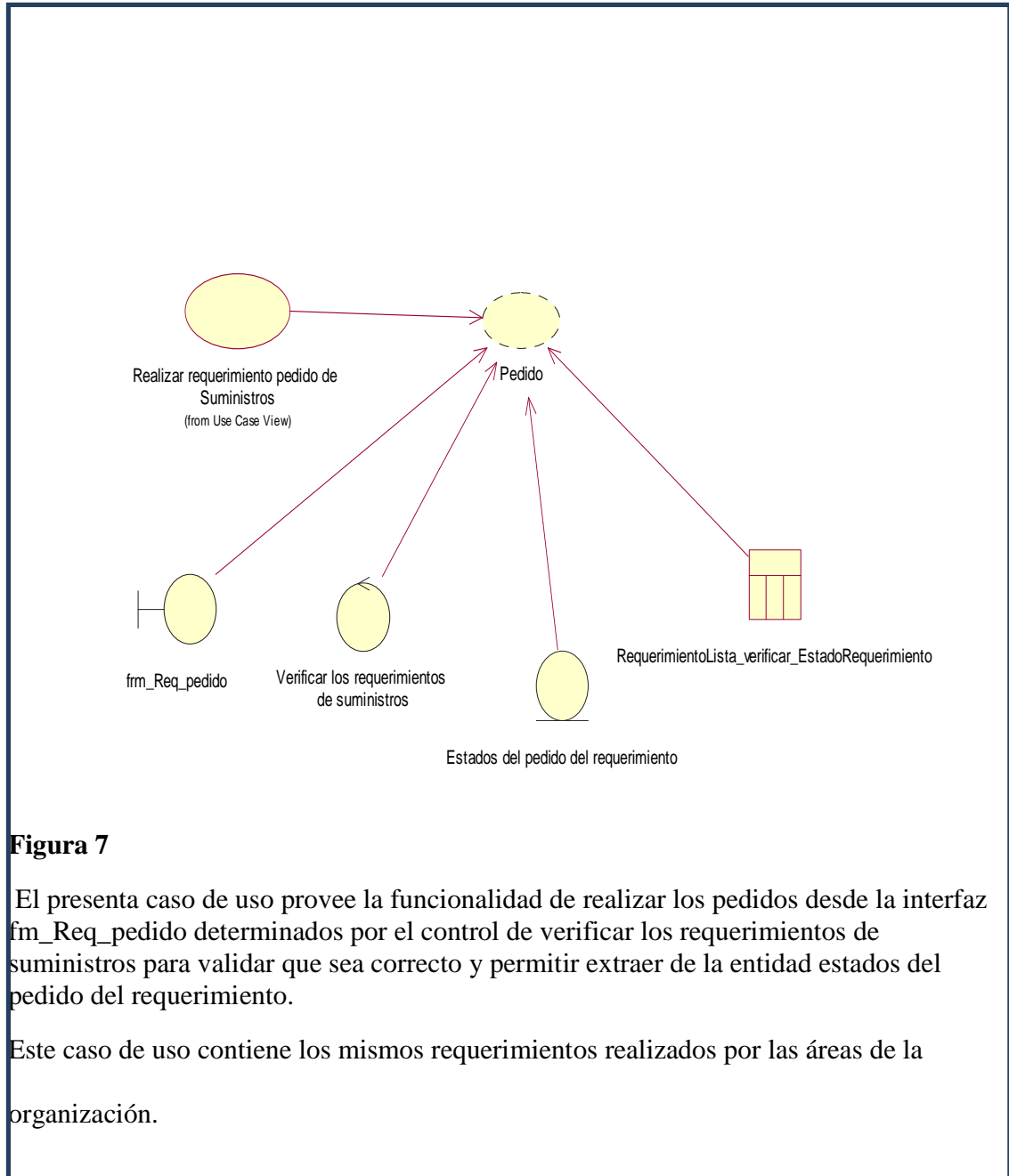
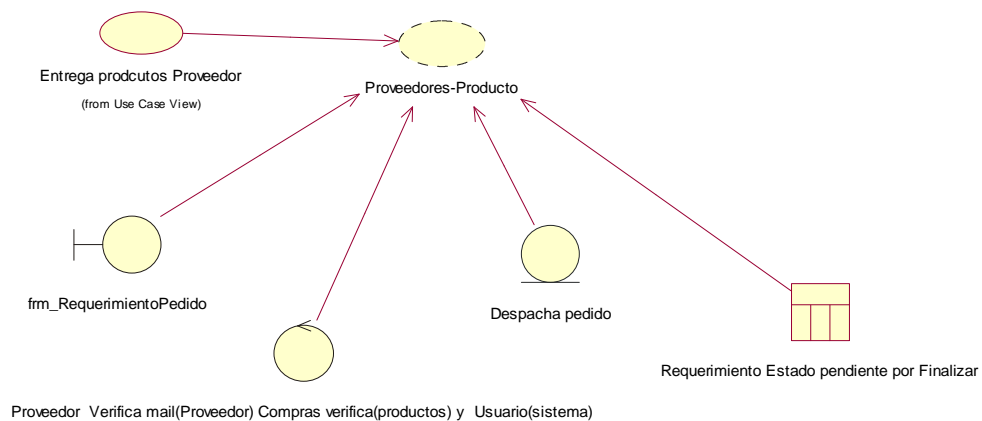


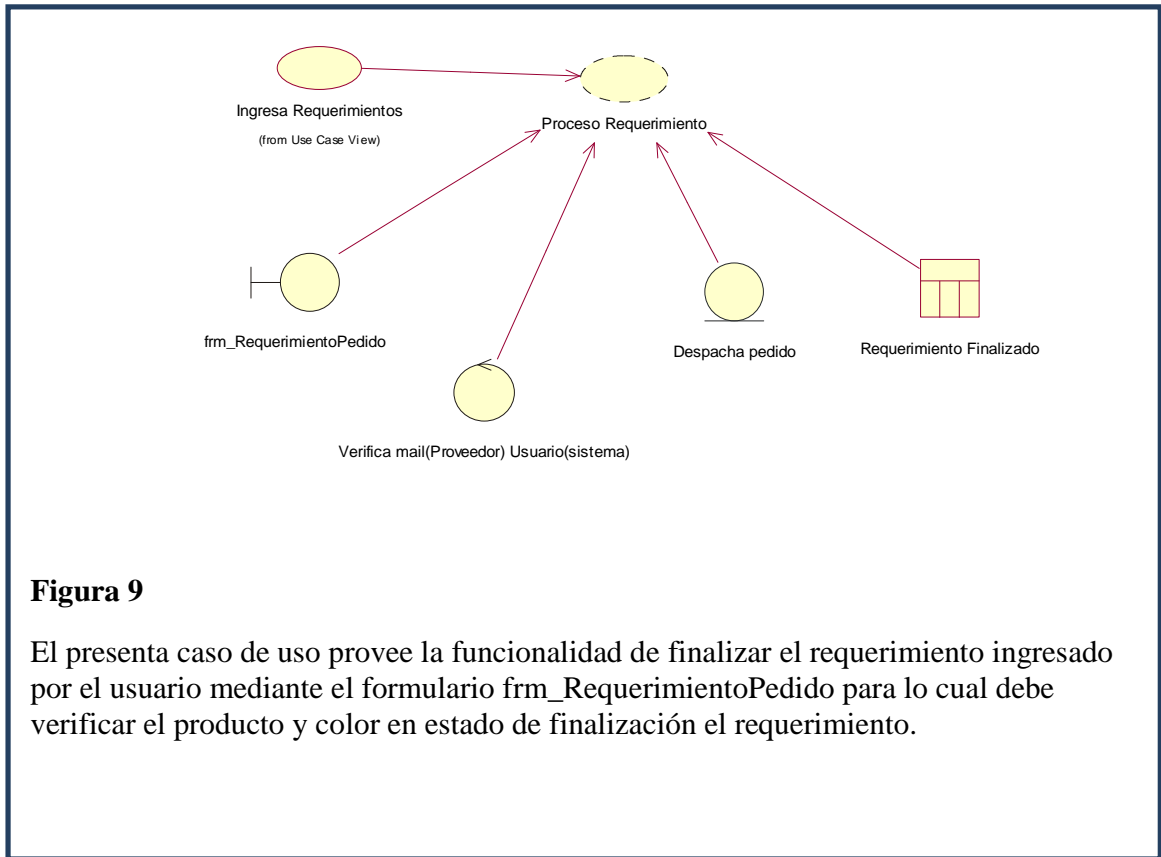
Figura 7

El presente caso de uso provee la funcionalidad de realizar los pedidos desde la interfaz fm_Req_pedido determinados por el control de verificar los requerimientos de suministros para validar que sea correcto y permitir extraer de la entidad estados del pedido del requerimiento.

Este caso de uso contiene los mismos requerimientos realizados por las áreas de la organización.

**Figura 8**

El presente caso de uso provee la funcionalidad de registrar la entrega de los pedidos que el proveedor realice, mediante la interfaz frm_RequerimientoPedido para el efecto deberá pasar por el control de verificación de mail que el proveedor debe realizar antes de la entrega del suministro luego el proveedor debe registrar el despacho del pedido y almacenarlo en la clase requerimiento estado pendiente donde guardará el estado del producto entregado por parte del usuario.

**Figura 9**

El presente caso de uso provee la funcionalidad de finalizar el requerimiento ingresado por el usuario mediante el formulario frm_RequerimientoPedido para lo cual debe verificar el producto y color en estado de finalización el requerimiento.

5.02.03 Diagrama de secuencia

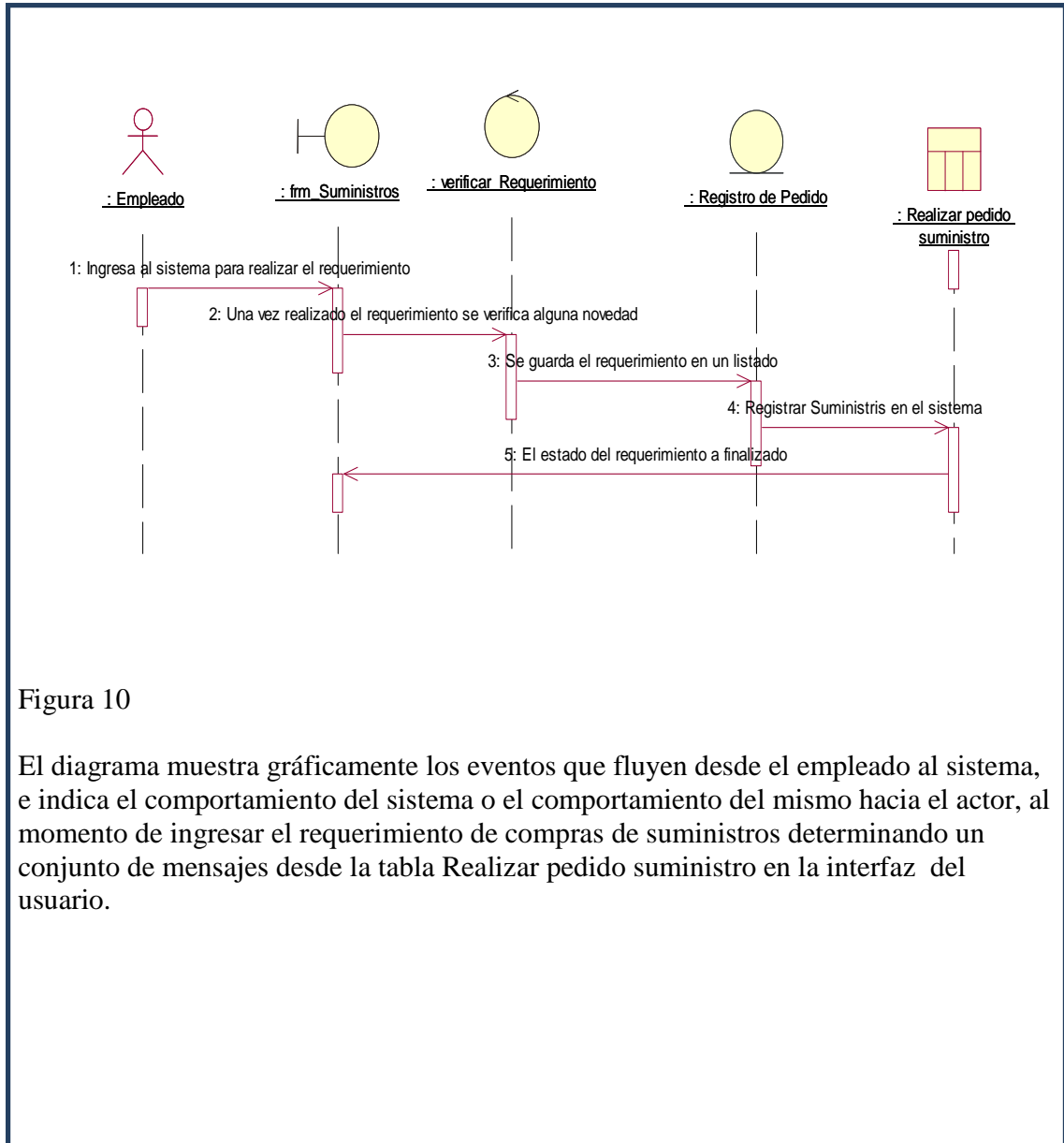


Figura 10

El diagrama muestra gráficamente los eventos que fluyen desde el empleado al sistema, e indica el comportamiento del sistema o el comportamiento del mismo hacia el actor, al momento de ingresar el requerimiento de compras de suministros determinando un conjunto de mensajes desde la tabla Realizar pedido suministro en la interfaz del usuario.



Diagrama de Secuencia Proveedores

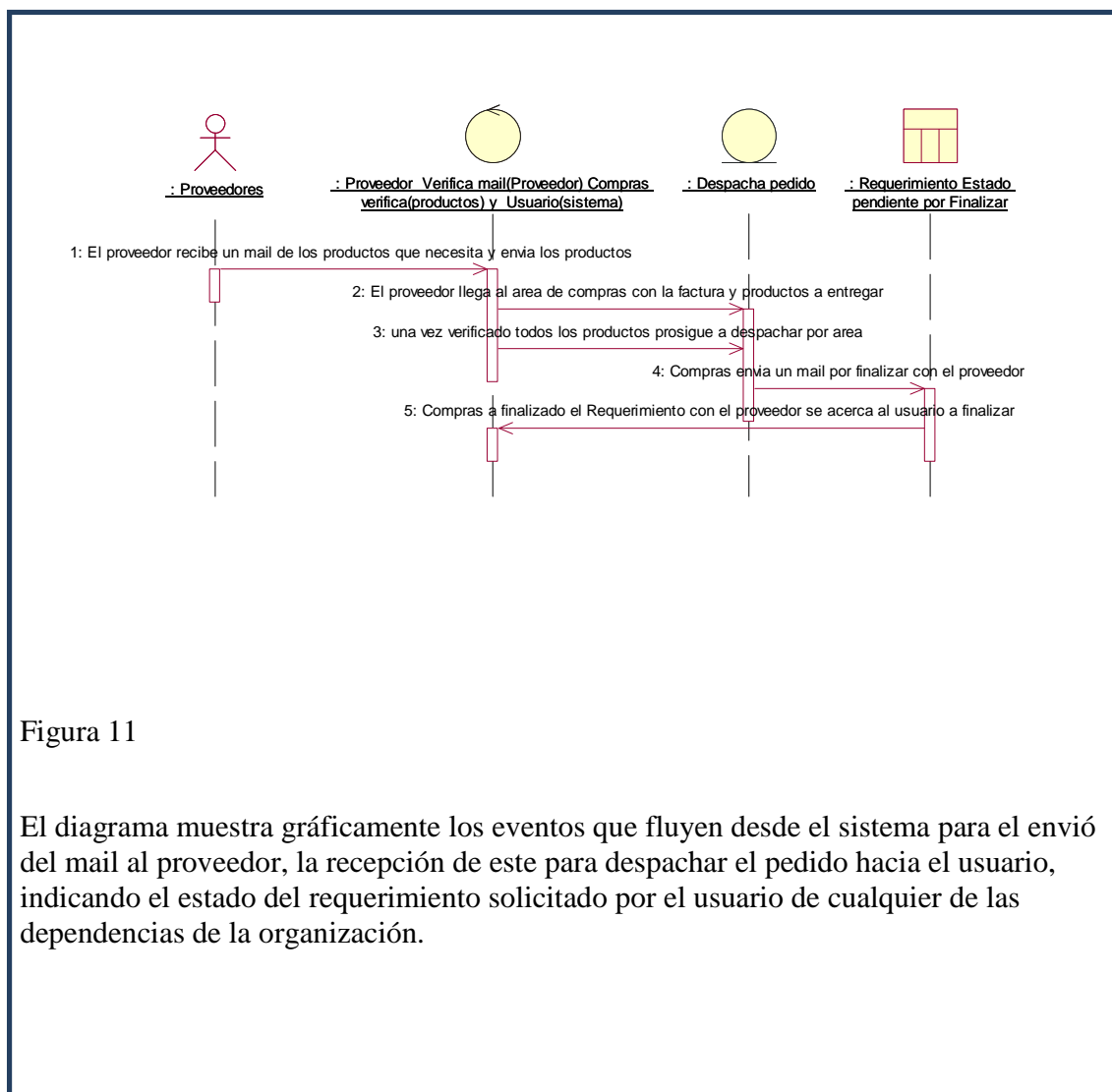


Figura 11

El diagrama muestra gráficamente los eventos que fluyen desde el sistema para el envío del mail al proveedor, la recepción de este para despachar el pedido hacia el usuario, indicando el estado del requerimiento solicitado por el usuario de cualquier de las dependencias de la organización.



Diagrama de Secuencia Finalizar Requerimiento

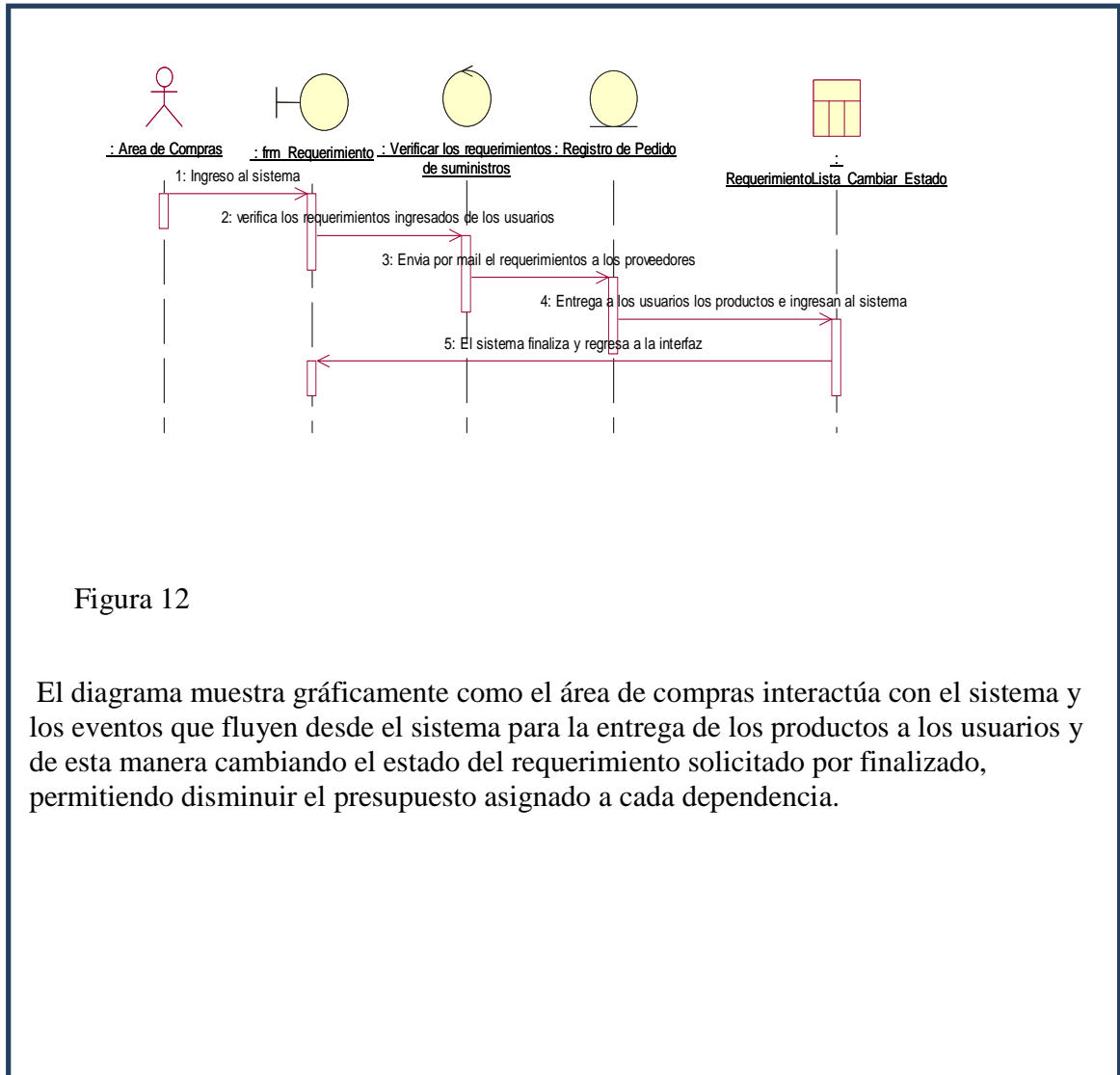


Figura 12

El diagrama muestra gráficamente como el área de compras interactúa con el sistema y los eventos que fluyen desde el sistema para la entrega de los productos a los usuarios y de esta manera cambiando el estado del requerimiento solicitado por finalizado, permitiendo disminuir el presupuesto asignado a cada dependencia.

5.02.04 Diagrama de colaboración

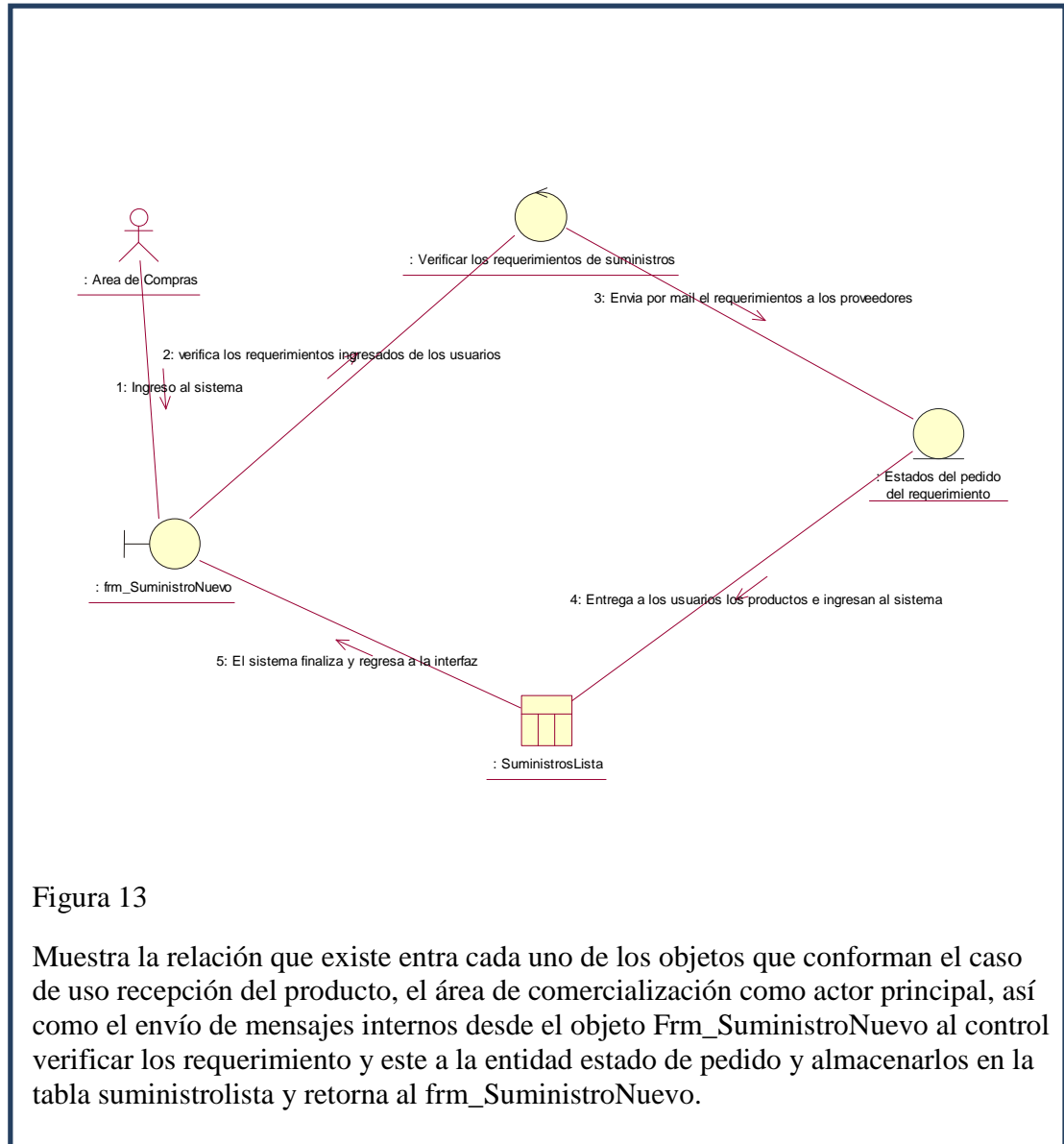
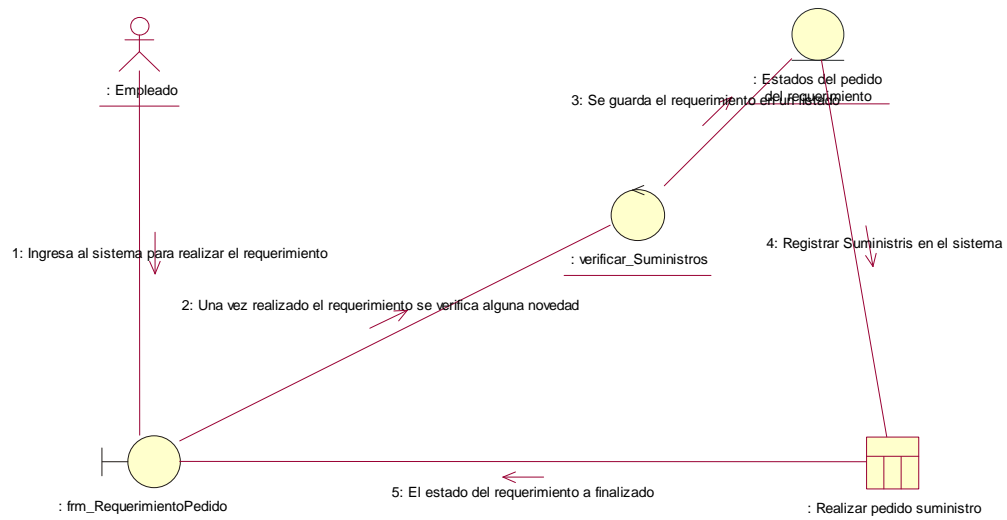


Figura 13

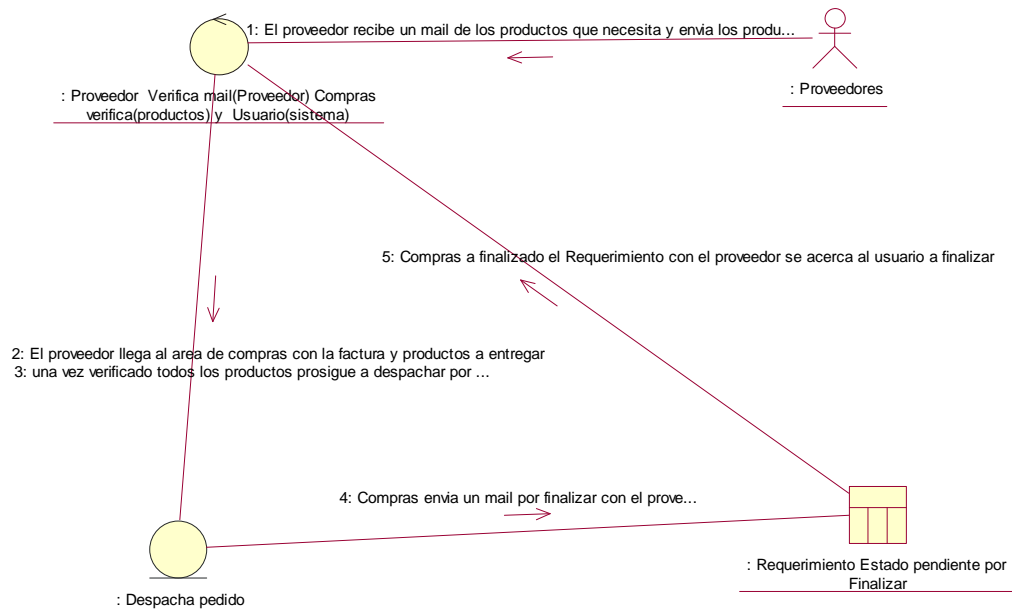
Muestra la relación que existe entre cada uno de los objetos que conforman el caso de uso recepción del producto, el área de comercialización como actor principal, así como el envío de mensajes internos desde el objeto Frm_SuministroNuevo al control verificar los requerimientos y este a la entidad estado de pedido y almacenarlos en la tabla suministrosLista y retorna al frm_SuministroNuevo.

**Figura 14**

Muestra la relación que existe entre cada uno de los objetos que conforman el caso de uso recepción del producto, el empleado como actor principal, así como el envío de mensajes internos desde el objeto Frm_RequerimientoPedido al control verificar_Suministro y almacenarlos en la tabla Realizar pedido Suministro y retorna al Frm_RequerimientoPedido donde el actor puede ver el comportamiento mediante un mensaje del estado.

Figura 15

Muestra la relación que existe entre cada uno de los objetos que conforman el caso de uso entrega del pedido, el proveedor como actor principal, así como el envío de mensajes internos desde el proveedor donde verifica el mail para proceder a mostrar la información en la entidad despacho pedido, considerando que se registra el estado



pendiente para proceder a finalizar el proceso verificando la entrega para la cancelación correspondiente.



5.02.05 Diagrama de componentes

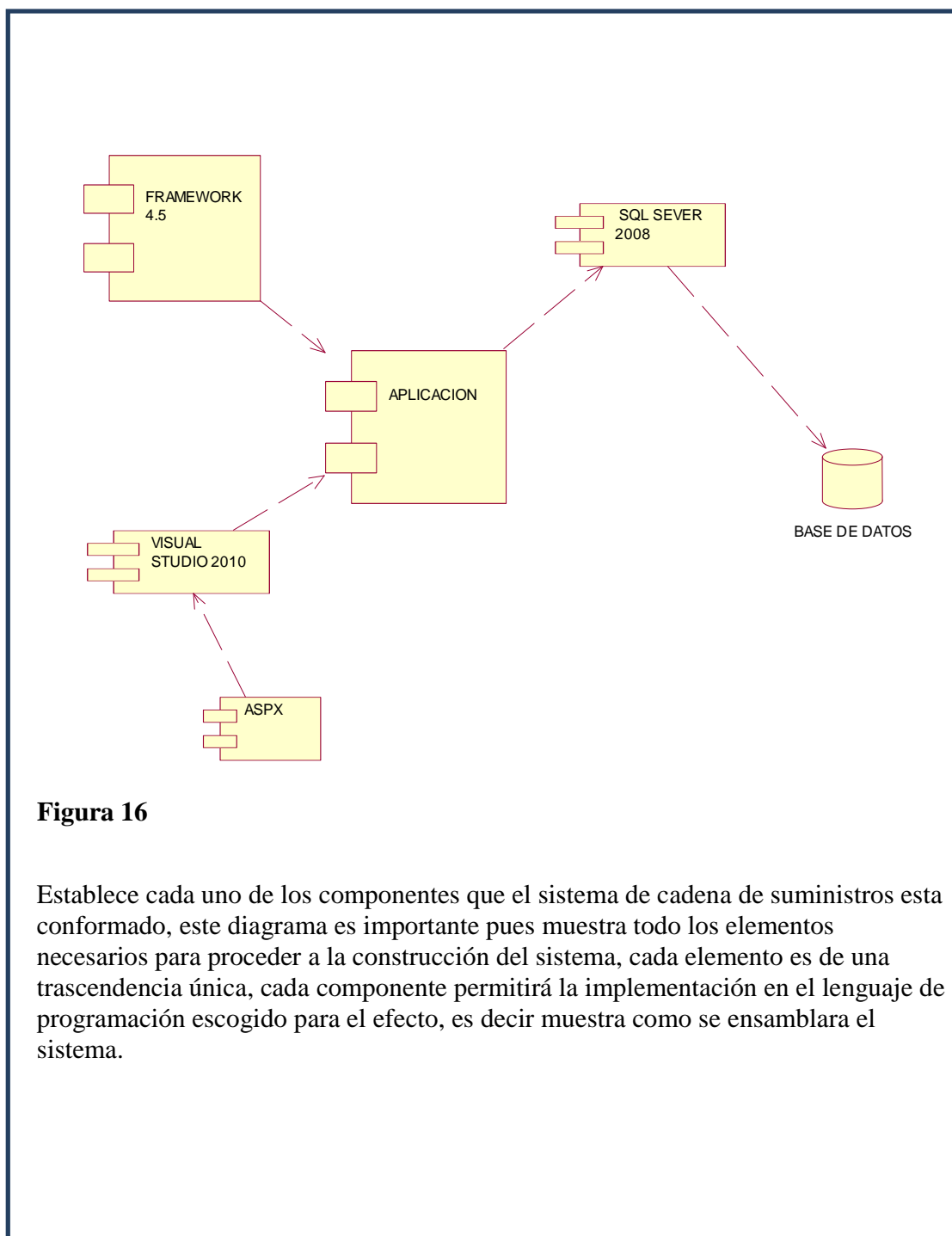


Figura 16

Establece cada uno de los componentes que el sistema de cadena de suministros esta conformado, este diagrama es importante pues muestra todo los elementos necesarios para proceder a la construcción del sistema, cada elemento es de una trascendencia única, cada componente permitirá la implementación en el lenguaje de programación escogido para el efecto, es decir muestra como se ensamblara el sistema.

5.02.06 Diagrama de clases

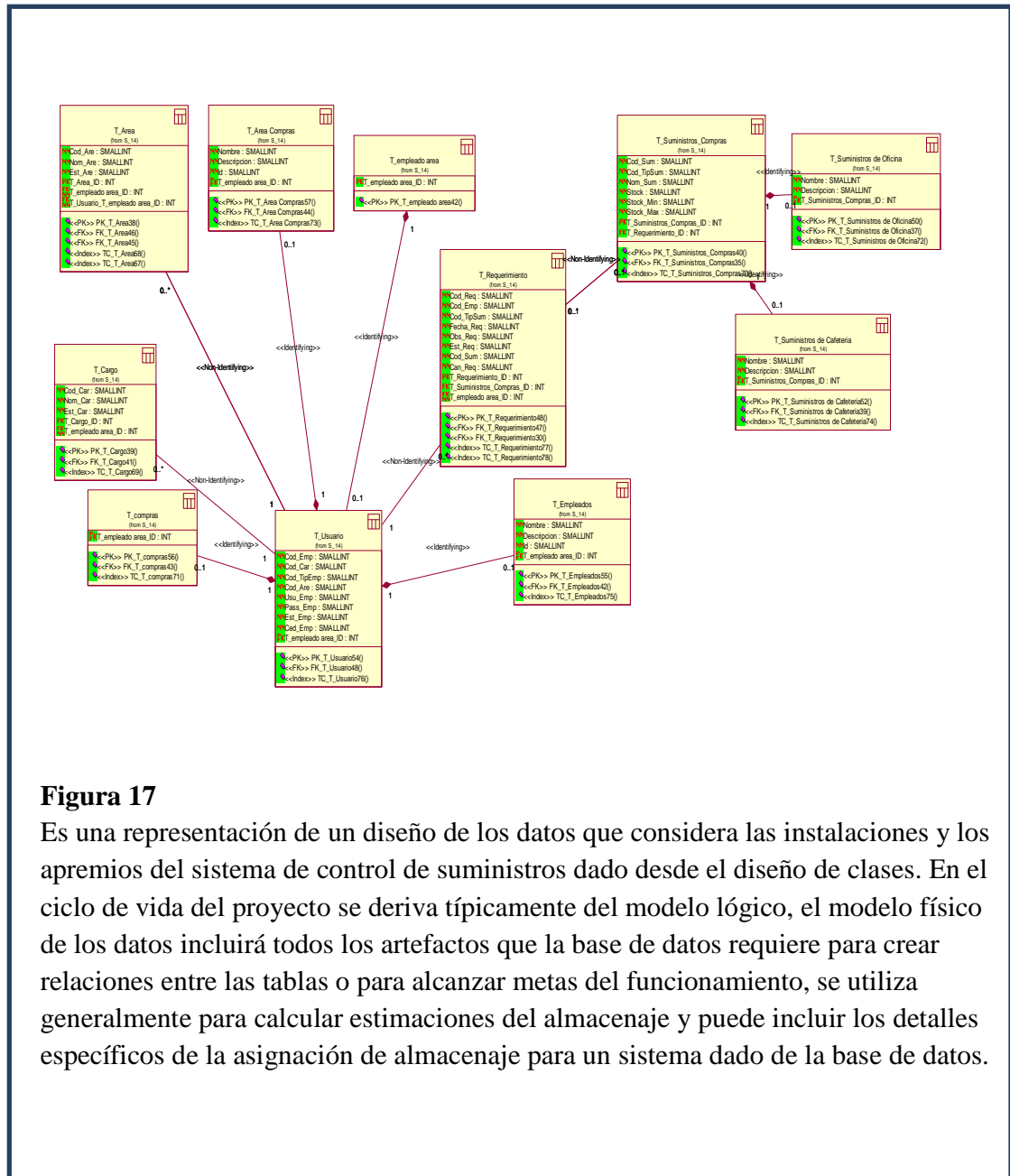
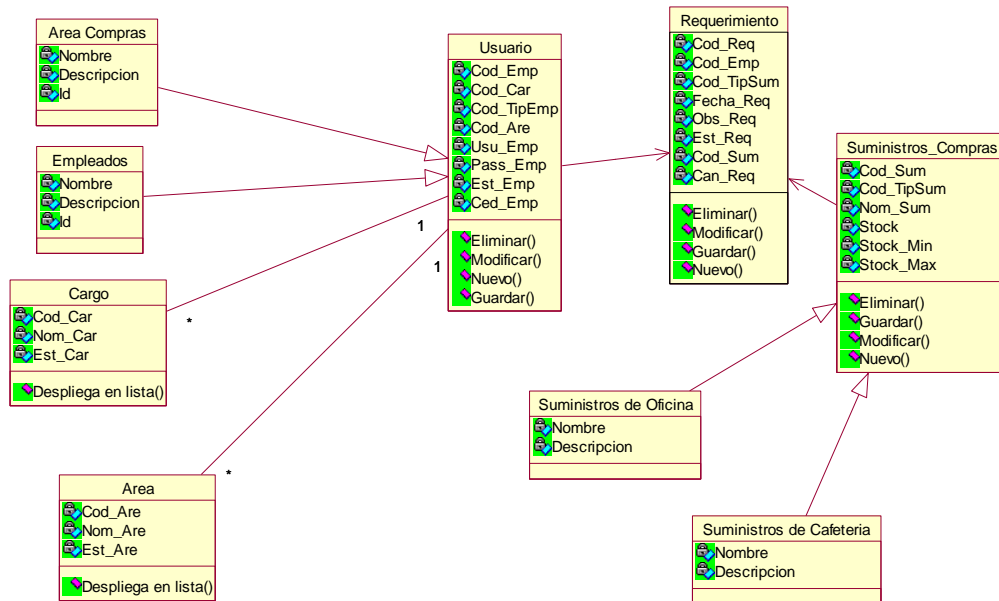


Figura 17

Es una representación de un diseño de los datos que considera las instalaciones y los apremios del sistema de control de suministros dado desde el diseño de clases. En el ciclo de vida del proyecto se deriva típicamente del modelo lógico, el modelo físico de los datos incluirá todos los artefactos que la base de datos requiere para crear relaciones entre las tablas o para alcanzar metas del funcionamiento, se utiliza generalmente para calcular estimaciones del almacenaje y puede incluir los detalles específicos de la asignación de almacenaje para un sistema dado de la base de datos.

**Figura 18**

El Modelo lógico es el modelo principal de diseño y análisis para el sistema de control de suministros, en él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal para el sistema de compras de suministros para satisfacer los detalles de la implementación.



5.03 Desarrollo

5.03.01 Arquitectura del sistema

El diseño arquitectónico que se presenta en este proyecto corresponde a los macros componentes con que deberá contar nuestra aplicación.

En la literatura vigentes relacionadas con la fase de diseño de software existen un sin número de técnicas y lenguajes para especificar esta fase. No obstante, nosotros consideraremos que el diseño arquitectónico para nuestra aplicación está conformada por: Arquitectura física, Arquitectura lógica.

5.03.01.01.- Arquitectura Física

En la arquitectura física, tratamos de expresar cuales son los componentes físicos que participan en nuestra solución y la relación existente entre ellos. Entre los componentes físicos que componen nuestro sistema podemos encontrar: Un servidor Web, un servidor de aplicaciones, un servidor de datos.

El servidor Web a utilizar será un servidor apache, él se encargará de contener todas las páginas de nuestro sistema. Por otro lado, se espera que este servidor resida en un servidor.

El servidor de aplicaciones corresponderá a la herramienta de configuración de las páginas web del sistema de suministros.

El servidor da corresponderá al motor de base de datos SQL Server, donde residirán todos los datos de nuestro sistema.

Para tener una visión general de los componentes y las interrelaciones que existen entre ellos podemos observar las siguientes figuras:

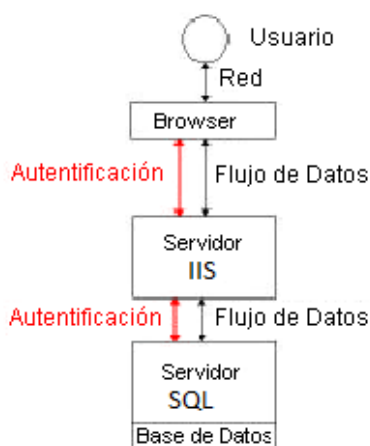
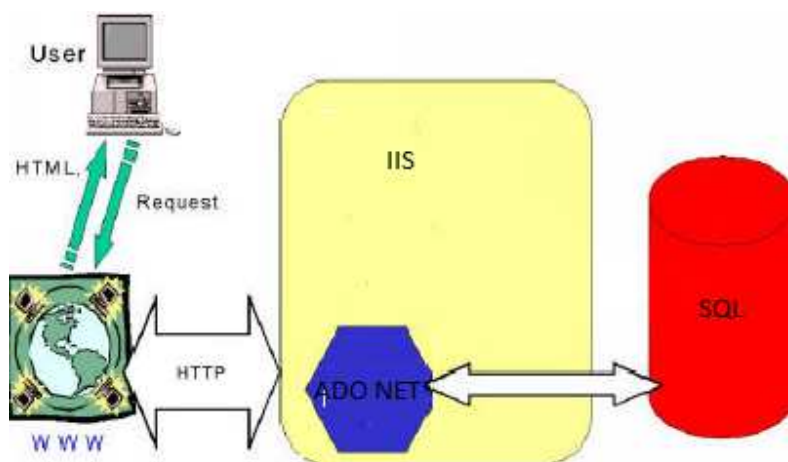


Figura 19

La figura expresa claramente cómo se trabajará en el desarrollo y ejecución de la aplicación, mostrando claramente como es el flujo de la información desde la conexión de los usuarios del área que necesita el requerimiento, pasando por la autenticación y obtener una respuesta satisfactoria desde el motor de base de datos. Como se explica en el documento “Sistema de Control de Suministros”, el lenguaje y el motor de base de datos se encuentran previamente definidos.

**Figura 20**

La figura número 5.03.01.01.02 trata de reflejar de una manera intuitiva el funcionamiento General de nuestro sistema. No obstante, y de manera más específica trataremos de abordar el problema enfocado en una arquitectura de tres capas, donde los servicios de datos y objetos residirán en el motor de base de datos SQL, ya que este motor cuenta con dicha capacidad.

5.03.01.02.- Arquitectura Lógica

La arquitectura lógica expresa cuáles son los componentes lógicos (subsistemas, o macro-funciones) que participan en nuestra solución, y la relación entre ellos.

La especificación de esta arquitectura, es similar a la arquitectura. Física. Se especifican actores y relaciones entre ellos.

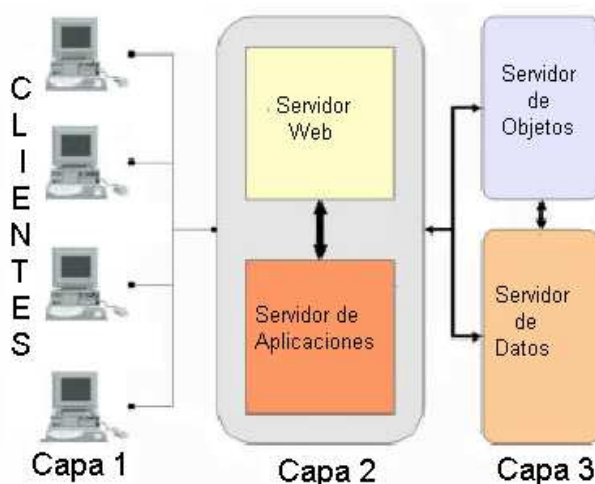


Figura 21

Muestra gráficamente como está estructurada la arquitectura lógica del sistema empezando desde la capa 1 que representa al cliente, la capa 2 a la configuración de los servidores web, y de aplicaciones para concluir con los servidores de objeto y de datos.

En los diagramas que expresan tanto la arquitectura lógica como la física, se puede utilizar casi cualquier simbología que clarifique el escenario. En nuestro caso



particular la arquitectura lógica estará definida utilizando los casos de uso utilizados en el Sistema de control de suministros. Además, en dicho documento se puede encontrar una descripción detallada de cada uno de los casos de usos más importantes.

5.03.02 Estándares

Los estándares en el desarrollo del sistema de control de compras, serán usados de forma paramétrica.

Esto nos indica que el sistema tendrá por objeto tener un control en los gastos de suministros de cafetería y oficina, pudiendo ser utilizado en cualquier área de la empresa.

La parametrización, nos servirá poder manejar

- Departamentos o áreas de la organización
- Tipos de suministros según será el caso.

5.03.02.02 Estándares de Base de Datos

Para un diseño correcto del sistema de control de suministros se ha establecido estándares tanto para la base de datos como para la programación agrupados en tablas como se muestra a continuación:

Tabla 9

Ejemplo del nombre de clases

Ver Anexos 77

5.03.02.03 Estándares de Programación

Tabla 10

Ver Anexos 78

5.03.02.04 Parámetros para documentar el código

Cada clase o módulo que se realice dentro del sistema, se encontrará debidamente comentado de una forma general de lo que hace dicho proceso. La siguiente codificación será necesaria para lo antes descrito, y no se eliminará ninguna información anterior, por ejemplo:

Autor: Sistema de Suministros

Versión: 1.0

Modificado por: Nicole Argudo

Fecha Inicio: 15/Ene/2013

Fecha Modificación: 01/Nov/2013

Descripción: Se realiza una validación adicional con respecto al ingreso de la cédula del empleado, se tiene una función que recibe como parámetro el número y devuelve si está correcta o no.

5.03.03 Diseño interfaz

5.03.03 Diseño interfaz



SISTEMA CONTROL DE COMPRAS MATERIAL DE OFICINA

REGISTRO

Nombre Usuario
jc_flores

clave usuario
ingresar la clave

INGRESO



Not a member yet? [Join us](#)

Figura 22

Esta pantalla indica el inicio de sesión de todos los usuarios tanto Administrador como Usuario. El usuario que van a ingresar es las dos primeras iniciales del nombre seguidas de un guion bajo y el apellido



Figura 23

El menú que se indica a continuación es usuario la pantalla que se muestra es de un ingreso del requerimiento. Podemos observar donde está el icono  seleccione empleado se despliega una página adicional donde el usuario elige su nombre, seleccionamos Tipo suministro y los productos se generan de inmediato, debemos colocar observación y cantidad seleccionamos este icono  se calcula el valor directo y agregamos los productos.

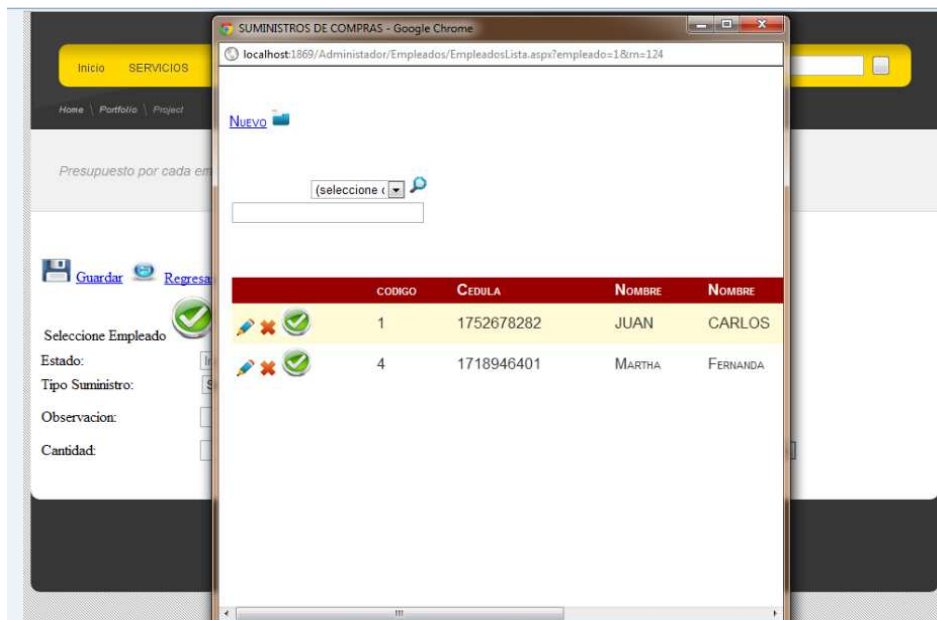


Figura 24

Al momento que agregamos los productos se genera una lista detallando los productos que escogimos la fecha, el valor, el presupuesto y el presupuesto asignado a cada área



Presupuesto por cada empleado

Datos Guardados

Guardar Regresar

Seleccione Empleado

Nombre: JUAN Apellido: FLORES

Estado: Ingresado

Tipo Suministro: Suministros de Oficina

Observación: Perfecto Fecha: 03/11/2013

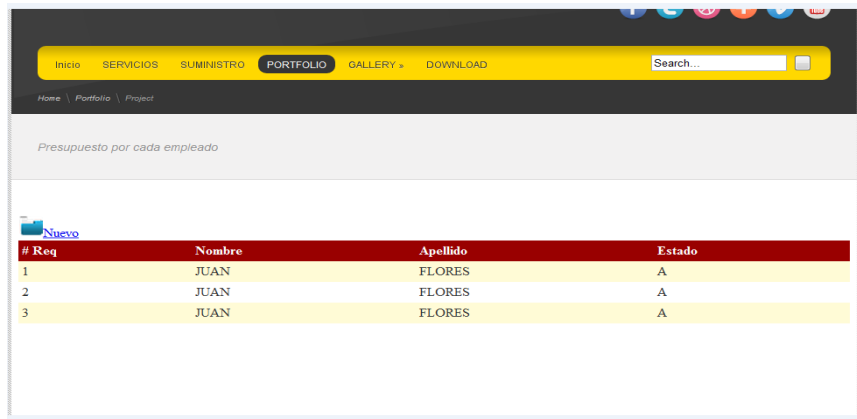
Cantidad: 1 Producto: 150,00 Cartucho A150

[Agregar Producto a Lista](#)

Tipo Pro	Producto	Fecha	Valor
F			
✗ Suministros de Oficina	Lapiz	03/11/2013	9,00
			Presupuesto Asignado: 300,00
F			
✗ Suministros de Oficina	Cartucho A150	03/11/2013	150,00
			PRESUPUESTO 159,00

Figura 25

Al momento que ingresa los datos debe guardar el requerimiento



The screenshot shows a web application interface. At the top, there is a navigation bar with links: Inicio, SERVICIOS, SUMINISTRO, PORTFOLIO, GALLERY, and DOWNLOAD. A search bar is also present. Below the navigation bar, there is a breadcrumb trail: Home > Portfolio > Project. The main content area has a heading "Presupuesto por cada empleado". Below this, there is a table with the following data:

# Req	Nombre	Apellido	Estado
1	JUAN	FLORES	A
2	JUAN	FLORES	A
3	JUAN	FLORES	A

Figura 26

En esta pantalla muestra el detalle de los requerimientos al guardarlos



Figura 27

Menú del Administrador



Figura 28

En el Menú de Administradores podemos visualizar las siguientes opciones:

Como por ejemplo ingreso de empleados áreas, productos, valores, etc.



5.04 Pruebas

El propósito de este apartado es planificar, estructurar y documentar la planificación de las pruebas de aceptación del sistema a realizar, así como la estrategia a utilizar para su ejecución.

Luego de finalizar las pruebas de sistema, el programa se encuentra completamente ensamblado, y se han encontrado y corregido los errores entre los módulos, métodos, clases y objetos. En este punto se comienza con la etapa de las pruebas de validación de requerimientos más conocida como pruebas de aceptación. Éstas se enfocan en las acciones que realiza el usuario además de las salidas del sistema que puedan ser reconocidas por él; dichas acciones y salidas engloban las expectativas del usuario, y están definidas en las especificaciones de los requerimientos del software.

Las pruebas de aceptación, se realizan a los requerimientos funcionales, y a los no-funcionales como facilidad de uso, recuperación, eficiencia, entre otros; y se pretende lograr: corrección, vale decir, carencia de ambigüedad; completitud, es decir, especificación completa y clara del problema; y por último pero no menos importante, consistencia, quiere decir, que no haya requisitos contradictorios.

Las pruebas que a continuación se detalla pretenden dar una visión general sobre las actividades a realizar; sobre las pruebas consideradas; además de una explicación global que se consideró para la realización de los documentos a entregar, ya que darán una mayor información relacionada a la evaluación y reportes de este tipo de pruebas.



El objetivo principal de las pruebas unitarias del sistema será el de establecer un nivel de confianza que permitirá asegurar la aceptación del sistema por los usuarios (Empleado, áreas de compras) en las posteriores pruebas de aceptación.

Se probará que la aplicación cumpla con los requerimientos que fueron especificados previamente, verificando que se cumple satisfactoriamente con las funcionalidades y características necesarias para que los usuarios satisfagan estos, enfocados en las siguientes áreas:

- Funcionalidad especificada en el sistema de compras de suministros.
- Manejo de los datos y transacciones involucradas en las funcionalidades del punto anterior.
- Rendimiento al ejecutar las funcionalidades del primer punto.

Las categorías asignadas a los resultados de prueba en un Caso de Prueba son:

1. **Éxito:** El resultado de la prueba es conforme al resultado esperado.
2. **Aceptable:** El resultado de la prueba indica que el sistema difiere de la especificación aceptada pero es aceptable, no son necesarios cambios en la aplicación.
3. **Tolerable:** El resultado de la prueba es incorrecto, la aplicación en prueba trabaja y podría ser aceptada, pero la falla deberá ser rectificada en el periodo de tiempo acordado.
4. **Intolerable:** El resultado de la prueba es incorrecto, y la falla debe ser corregida antes de concluir la fase de prueba.



5. **Error:** El resultado de la prueba observado es correcto, pero el resultado esperado de acuerdo a los scripts de prueba son incorrectos

Las categorías de los resultados de pruebas que son consideradas para aceptar la aplicación y pasar con éxito la fase de prueba. Identificamos los siguientes criterios los cuales deben ser evaluados progresivamente.

- Requerimientos de Prueba: ¿Todos los requerimientos del sistema han sido probados?
- Pruebas Cubiertas: ¿Todas las partes del software han sido probadas, incluyendo manejo de errores?
- Medida de Casos de Prueba: ¿Cuántos Casos de Prueba han sido planeados, diseñados, implementados, ejecutados y pasaron con éxito o falla?
- Defectos detectados en Casos de Prueba: Es importante tener un ratio de los defectos encontrados en los casos de prueba, y de los defectos corregidos y mantenidos.

5.04.01 Pruebas Funcionales

- Revisar la implementación del caso de uso. **Ingreso del requerimiento**
- Revisar la implementación del caso de uso Ingreso de suministro.
- Revisar la implementación del caso de uso entrega de productos.

5.04.02 Pruebas por Caso de Uso

Para las pruebas de casos de uso se probarán en el siguiente orden: Entrega de producto, Ingreso de suministro, Ingreso del requerimiento donde se probara la correcta implementación de los flujos de eventos y flujos alternativos

5.04.03 Casos de Prueba

SISTEMA DE COMPRAS				
Caso de Uso	UC_1 Realizar pedido de suministro			
Caso de Prueba	CP_01 Ingreso de suministro			
Actor	Empleado			
	Área de Compras			
Pre Condiciones:	El Empleado debe estar registrado en el sistema			
	La Lista de solicitud de suministro debe estar disponible			
	El sistema debe notificar al empleado en la interfaz del menú principal.			
Propósito:	ingresar el requerimiento de suministro de oficina o de cafetería			
Escenario	CP_01_E01: Comprobar la correcta Generación del pedido de Requerimiento			
	Sec.	Actividad	Clase de equivalencia	Resultado Esperado
	1	Datos de la solicitud de requerimiento: Datos del empleado : Área de destino: Datos del Requerimiento: Detalle del requerimiento:	Válida	Visualiza los datos seleccionados en los respectivos campos de acuerdo al conjunto de datos ingresados.
	2	Solicita " Consultar Solicitud de requerimiento"	Válida	Muestra un mensaje : "Datos del orden de Requerimiento a generar".



SISTEMA DE COMPRAS

Caso de Uso	UC_3 Despacho de producto			
Caso de Prueba	CP_03 Despacho de producto			
Actor	Área de compras			
	Proveedor			
Pre Condiciones:	El actor área de compras debe estar registrado en el sistemas			
	La Lista de solicitud de suministro debe estar disponible			
	El sistema debe permitir al actor área de compras verificar que la entrega del producto se haya dado.			
Propósito:	constatar la entrega del suministro de oficina o cafetería			
Escenario	CP_03_E01: Despacho y cambio de estado			
	Sec.	Actividad	Clase de equivalencia	Resultado Esperado
	1	El proveedor despacha el suministros de acuerdo a los requerimientos :	Válida	El actor área de compras visualiza el cambio de estado de la solicitud ingresada por el empleado.

SISTEMA DE COMPRAS

Caso de Uso	UC_2 Entregar suministro según requerimiento			
Caso de Prueba	CP_02 Entrega de suministro			
Actor	Empleado			
	Proveedor			
Pre Condiciones:	El Empleado debe estar registrado en el sistemas			
	La Lista de solicitud de suministro debe estar disponible			
	El sistema debe notificar al proveedor vía email el requerimiento			
Propósito:	Caminar el estado de los suministros de oficina o de cafetería			
Escenario	CP_02_E01: Comprobar la correcta entrega del envío de mail			
	Sec.	Actividad	Clase de equivalencia	Resultado Esperado



1	Datos de la solicitud de email Datos del empleado : Área de destino: Datos del Requerimiento: Detalle del requerimiento:	Válida	Visualiza los datos seleccionados en los respectivos campos de acuerdo al conjunto de datos ingresados n el email del proveedor.
---	--	--------	--

Capítulo VI: Aspectos Administrativos

6.01 Recursos

Los recursos que obtuvimos para la implementación del sistema fueron los siguientes:

Recursos Materiales y Tecnológicos	Laptops, Maquinas de escritorio, servidores, etc.
	Microsoft Visual Studio 2010, MySQL Server 2008, Internet.
Recursos Humanos	Todos los empleados que conforman la empresa.
	Personas externas a la organización (Proveedores).



6.02 Presupuesto

Para analizar el presupuesto que se necesita en la inversión de una nueva herramienta para el uso exclusivo no fue necesario tomar en cuenta la parte de costos

por mantenimientos mensuales o anuales ya que dentro de la Empresa contamos con un área de desarrollo y un área técnica. Aparte de estas áreas contamos con servidores, UPS y claro está un DATACENTER.

El sistema como tal es una herramienta costo- beneficio, esto se refiere a que los usuarios podrán tener control de sus compras mensuales y la empresa lograra reducir tanto gasto innecesario e incrementaran beneficios.

6.03 Cronograma



		MESES				Mayo				Junio				Julio				Agosto				Septiembre				Octubre				Noviembre							
		SEMANAS				1ra	2da	3ra	4ta	1ra	2da	3ra	4ta	1ra	2da	3ra	4ta	1ra	2da	3ra	4ta	1ra	2da	3ra	4ta	1ra	2da	3ra	4ta								
DETALLES DEL DESARROLLO																																					
Planteamiento del Proyecto																																					
Aprobación del Proyecto																																					
Desarrollo del Portal Fase I																																					
Pruebas I																																					
Desarrollo del Portal Fase II																																					
Pruebas II																																					
Desarrollo del Portal Fase III																																					
Pruebas III																																					
Desarrollo del Portal Fase V (FINAL)																																					
Pruebas V																																					
Aprobación del Portal por el Director de Carrera																																					
Montaje del Portal en el Servidor																																					
Pruebas en el Servidor																																					
Publicación del Sistema de compras																																					

Capítulo VII: Conclusiones y Recomendaciones

7.01. Conclusiones

Una vez realizada la investigación y trabajando con los datos se puede exponer un criterio que se obtuvo acerca del departamento de compras y el departamento de sistemas para trabajar en un seguimiento de control de compras ya que es una función muy importante para controlar la mala manipulación de gastos directamente en el consumo de suministros.

Es importante tener un seguimiento en la falta de control en la adquisición de suministros, una vez hecha el consumo se debería plantear algunos reportes sobre el grado de abuso y descontrol en las compras, con este proyecto se podrá tener un mejor

avance en la parte financiera porque podrán revisar reportes mensuales analizando con esto que área consume más y porque consume en exceso en el caso que se diera.

Una vez implementado este sistema el área de soporte como de compras estará preparado para cualquier inconveniente sea del sistema o con productos innovadores al mercado, se estará actualizando todos estos productos en el sistema para evitar molestias al momento de adquirir algún suministro en especial.

7.02. Recomendaciones

Para evitar esta ambigüedad, es necesario elaborar, a partir de la investigación, una definición y clasificación clara de los tipos de suministros de compras tanto de oficina como de cafetería adaptada a los objetivos curriculares de la empresa o mejor dicho lo que la empresa quiere como tal.

Existe una escasez de resultados duraderos de proyectos sobre suministros de compras por parte de las Empresas de Quito

Para realizar un análisis minucioso en el levantamiento de información, siendo fundamental la toma de decisiones sobre el desarrollo de la aplicación, ya que esto



almacena información para el área financiera obteniendo mejores resultados para nuevas tomas estratégicas de la empresa como por ejemplo comparar gastos del año anterior con el año actual.



Anexos A.01

A.01.01

Tabla 2

Análisis de Involucrados

Actores	Intereses	Problemas Percibidos	Recursos y mandatos	Intereses en el proyecto	Conflicto y alianzas
Proveedores	Mejorar el sistema de suministros, Innovación centrada en el cliente interno	Bajo interés en el control de suministros. Deficiente formación de un coordinador	Proporcionar una tasa de interés bajo para la empresa. Cotizaciones con ofertas muy convenientes	Corregir el manejo de pedidos de suministros	Conflicto al generar el pago a los proveedores en las fechas designadas según la Empresa
Clientes Internos	Mejorar la forma en que desarrollan sus pedidos mensuales y ajustarse a un	Crisis Inicialmente jerárquica, sin apoyo de los colaboradores. Dificultad de	Su participación y cooperación es imprescindible Interés en mejorar la	La innovación se lleve a cabo para optimizar la productividad en tiempo y	Aprobación luego de presentar el proyecto ante la junta Administrativa



	valor adecuado de gasto mensual sin sobrepasar los valores estimados	cambiar procesos. Insuficiencia de tiempo para participar en mejoras de procesos. Inconformidad en los procesos de suministros, Se incrementa la mejor utilización de los suministros	utilización de los suministros. Formalmente comprometidos en los temas de utilización de suministros, pero en algunos casos no se puede constatar este compromiso. Garantizar al acceso de los recursos de la cadena de suministros	dinero para mejorar la inconformidad de suministros	con la aceptación de socios y empleados
Área Financiera	Otorgar una mejor automatización	Altos niveles en gastos por diferentes	Capacidad de Aprobar el Financiamiento.	Corregir la mala planificación	Conflicto con los clientes



	para incrementar niveles económicos	áreas de la empresa. Altos niveles de consumo. Malos hábitos de utilización. Incremento de los costos de compra	Capacidad de contratación de una compra de suministro. Disposición a colaborar.	de pedidos en las áreas. Minimizar tiempo y costos.	internos por las normas de regularización en los usos de suministros. Proveedores que están acostumbrados a la entrega oportuna de los suministros Departamento de compras por elevados consumos de suministros
Departamento de Compras	Automatizar los procesos de la cadena de suministros.	Altos índices en los costos. Mala planificación	Control máximo en la aprobación de órdenes de	Mejorar los procesos donde el sistema	Conflicto con todas las áreas por los cambios en las



	Disminuir el gasto del presupuesto de la cadena de suministros. Mejorar los tiempos de respuesta en la entrega de suministros en los clientes internos	de insumos en las diferentes áreas. Mala coordinación con proveedores.	compras, Planificación con proveedores. La cooperación y participación es imprescindible. Infraestructura institucional.	influya para minimizar los costos de consumos. Que la reforma se lleve a cabo para mejorar las condiciones y usos de la cadena de suministros por parte de los clientes internos. Gran interés en la reforma de la cadena de suministros pero con los recursos	políticas de adjudicación de suministros con los clientes internos que están mal acostumbrados a la mala utilización de suministros y a no apoyar a las buenas prácticas de consumo.
--	--	--	--	--	--



				adecuados. Apoyar siempre y cuando no se requiera incremento en el presupuesto del departamento	
--	--	--	--	---	--



A.01.02

Tabla 3.

Análisis de Alternativas

MATRIZ DE ANALISIS DE ALTERNATIVAS							
OBJETIVOS	PROPOSITO	IMPACTO SOBRE EL	TECNICA	FINANCIERA	SOCIAL	POLITICA	TOTAL
							CA
							TE
							GO
							RI
							A

Los

departamentos

utilizan

eficientemente

los recursos

suministros

dotados

Medio



los clientes internos deberán aprovechar los recursos en las operaciones de la empresa	3	3	2	2	2	12	Medio
--	---	---	---	---	---	----	-------

El proceso

de

requerimien

tos de

suministros

son

definidos

bajo

políticas

organizacio

nales

3	2	2	2	2	11	Medio
---	---	---	---	---	----	-------



Investigación								Medio
mercado eficiente								
para la selección	2	2	3	3		3	13	
de								
proveedores								
Requerimientos								
ingresados								
adecuadamente		2	3	3	2	2	12	Medio
nente para la								
programación								
de								
suministros								
0 - 6	Bajo	13	12	13	11	11		
7 - 13	Medio							
14-18	Alto							



A.01.03

Tabla 4

Matriz de Marco Lógico

Resumen Narrativo	Indicadores	Medios de Verificación	Supuestos
FIN DEL PROYECTOS Optimización de los suministros operacionales de la empresa	Porcentaje de reducción de gastos en los suministros. Reducción de tiempo de entrega del suministro por parte de los proveedores. Reducción de la insatisfacción de los clientes internos.	Reporte del departamento de tesorería para analizar mensualmente gastos de suministros de las diferentes áreas de la empresa. Reportes de análisis de control de tiempo en las entregas. Encuestas mensuales a responsables de cada área para determinar el nivel	Resistencia al cambio por parte del personal.



		de satisfacción del cliente interno	
PROPOSITO DEL PROYECTO Manejo adecuado de la cadena de suministros	Manejo adecuado de la cadena de suministros	conclusiones emitidas por parte de empleados internos	Mayor desinterés en el uso de la aplicación informática.
COMPONENTES DEL PROYECTO C1.Requisitos de suministros por área. C2.Control de entrega y recepción de suministros C3. Programación de suministros de oficina	Números de requerimientos por fecha de corte. Contabilizar el número de requerimientos máximo por área.	Estadísticas por parte del departamento Financiero al obtener todas las encuestas por área y persona.	Capacitación para las diferentes áreas de la empresa se mantiene estable y coordinadas sin perder interés por la retroalimentación (feedback)
Actividades del proyecto por cada componente detallado se debe detallar Actividades	Para disminución de gastos tendremos reportes que servirá al área	Los requisitos de cada uno de las áreas serán analizadas en el impacto de cualquier toma de	Los procesos serán actualizados hacia el personal con capacitaciones si es el caso.



	financiera	decisión	



A.01.04

Tabla 5

Análisis Impacto de objetivos

Análisis Impacto de objetivos						
	Factibilidad de Lograse	Impacto en Género	Impacto Ambiental	Relevancia	Sostenibilidad	Total
Los departamentos utilizan eficientemente los recursos de suministros dotados	* Se incrementara una mejor comunicación con los clientes internos	Se incremente la participación de clientes internos	Contribuye a proteger el entorno físico	Responde a las expectativas de los beneficiarios	Fortalece a la Organización	63
	*Se utilizara de mejor forma los requerimientos de suministros	Incrementa los ingresos disminuyendo gastos innecesarios	Mejora el entorno social.	Los beneficios son deseados por los beneficiarios	Fortalece la participación de los beneficiarios	0-21 BAJO



Los clientes internos						
	deberán	*Adecuada organización	Incrementa los niveles de consumo en el control de suministros	Mejora el entorno cultural	Es una prioridad sentida por los beneficiarios	Fortalece la selección de proveedores
	aprovechar los recursos en las operaciones de la empresa					
						22-42 MEDIO
Investigación de mercado eficiente para la selección de proveedores	Se cuenta con una buena selección de proveedores	Fortalece las políticas determinadas para un control de requerimientos de consumo	Protege el uso de los recursos.		mejora la recepción de suministros	43-63 ALTO
Requerimientos ingresados adecuadamente para la programación de suministros	Existe adecuada recepción de suministros		Favorece la educación ambiental			
	15	12	15	9	12	



A.01.05

Tabla 9

Ejemplo del nombre de clases

Tipo	Descripción	Nombre
Base de Datos	Para el nombre de las clases se utilizarán mayúsculas y minúsculas, no se aceptan caracteres especiales ni espacios en blanco ni nulos.	SUMINISTROS_COMPRAS
Clases	El nombre de las clases que hace de referencia Deberán ser escritas en mayúsculas y minúsculas. No se utilizarán caracteres nulos.	Usuario Empleados
Atributos	para describir los Atributos se tomarán las tres primeras letras del nombre del atributo seguido de un guion bajo,	Cod_Req Cod_Car Ced_Emp



	las tres primeras letras del nombre de la clase	
Métodos	Para nombrar a los métodos se utilizarán	Eliminar()
	mayúsculas y minúsculas, se admiten espacios en	Modificar()
	blanco pero no caracteres especiales, seguidos de dos	Nuevo()
	paréntesis	Guardar()

A.01.05

Tabla 10

Estándares para la programación del software

Tipo	Descripción	Nombre
Variable	nombre completo de variables en referencia a su utilización	Codigo
		Nombre
		Apellido
Clase	Deberá estar escrito por el nombre de las tablas seguida por un guion bajo, y por el	Requerimiento_Logica Area_Logica



	nombre completo en minúsculas de la clase.	
Método	Nombre del método deberá estar escrito la primera con mayúscula y lo demás con minúsculas.	Guardar Modificación Eliminar
Biblioteca de Clases	El nombre deberá estar escrito por sus iniciales en mayúsculas y lo correspondiente en minúsculas sin espacios en blanco.	AccesoDatos LogicaNegocio Administrador
Objetos	El nombre de los objetos deberán estar escritos la primera con mayúscula y en minúsculas seguido por un guion bajo y el nombre completo del objeto referido a la clase perteneciente.	NuevoEmpleado



etiquetas, cajas de textos, tablas de datos, botones, listas desplegables, calendarios.	Todos los objetos mencionados sus nombres deberán sus nombres deberán estar compuesto por la abreviatura del tipo seguido por un guión bajo y el nombre completo de la referencia a utilizarse.	lbl_Suministros txt_Nombre hdf_Empleado
Formularios Web	Los formularios deberán estar escritos por el nombre de la tabla seguido por el nombre completo de su referencia a utilizarse.	RequerimientoNuevo EmpleadoLista



Anexo 2: Manual Técnico

A.02.01 INTRODUCCION

El presente documento se constituye en una herramienta de soporte y ayuda para los Ingenieros Desarrolladores, Administradores, Operadores y en general para cualquier persona interesada en el concepto Técnico del Sistema control de compras de suministro.

En el manual técnico se detalla parte de la estructura de la programación realizada. Con la finalidad que el departamento técnico pueda comprender la lógica de programación empleada. Además de cómo fueron estructurados los datos para la creación de las tablas de la base de datos.

A.02.02 Objetivo del manual

Proporcionar un resumen de la composición técnica de cada módulo y pantalla, así como los detalles técnicos de manera clara mediante la descripción y gráficos del sistema, para que los usuarios técnicos tengan una mayor visión de la estructuración y funcionamiento del sistema.

A.02.03 Base de datos

A continuación se lista el script de la base de datos con la cual se conecta y maneja el sistema.

USE [SuministrosBD]

GO



/***** Object: Table [dbo].[TIPO_SUMINISTROS] Script Date: 11/10/2013

20:24:02 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[TIPO_SUMINISTROS](

[COD_TIPSUM] [int] NOT NULL,

[NOM_TIPSUM] [char](50) NULL,

[EST_TIPSUM] [char](10) NULL,

CONSTRAINT [PK_TIPO_SUMINISTROS] PRIMARY KEY NONCLUSTERED

(

[COD_TIPSUM] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,

IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,

ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI_PADDING OFF

GO



```
INSERT [dbo].[TIPO_SUMINISTROS] ([COD_TIPSUM], [NOM_TIPSUM],  
[EST_TIPSUM]) VALUES (1, N'Suministros de Oficina', N'A')  
INSERT [dbo].[TIPO_SUMINISTROS] ([COD_TIPSUM], [NOM_TIPSUM],  
[EST_TIPSUM]) VALUES (2, N'Suministros de Cafeteria', N'A')  
/***** Object: Table [dbo].[TIPO_EMPLEADO]  Script Date: 11/10/2013 20:24:02  
*****/  
  
SET ANSI_NULLS ON  
  
GO  
  
SET QUOTED_IDENTIFIER ON  
  
GO  
  
SET ANSI_PADDING ON  
  
GO  
  
CREATE TABLE [dbo].[TIPO_EMPLEADO](  
    [COD_TIPEMP] [int] NOT NULL,  
    [NOM_TIPEMP] [char](50) NULL,  
    [EST_TIPEMP] [char](10) NULL,  
    CONSTRAINT [PK_TIPO_EMPLEADO] PRIMARY KEY NONCLUSTERED  
(  
    [COD_TIPEMP] ASC  
)  
WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
ON [PRIMARY]
```



```
GO

SET ANSI_PADDING OFF

GO

INSERT [dbo].[TIPO_EMPLEADO] ([COD_TIPEMP], [NOM_TIPEMP],
[EST_TIPEMP]) VALUES (1, N'ADMINISTRADOR', N'A
')

INSERT [dbo].[TIPO_EMPLEADO] ([COD_TIPEMP], [NOM_TIPEMP],
[EST_TIPEMP]) VALUES (2, N'USUARIO', N'A')

/***** Object: Table [dbo].[CARGO] Script Date: 11/10/2013 20:24:02 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[CARGO](
    [COD_CAR] [int] IDENTITY(1,1) NOT NULL,
    [NOM_CAR] [char](50) NULL,
    [EST_CAR] [char](10) NULL,
    CONSTRAINT [PK_CARGO] PRIMARY KEY CLUSTERED
(
    [COD_CAR] ASC
```



```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
  
) ON [PRIMARY]  
  
GO  
  
SET ANSI_PADDING OFF  
  
GO  
  
SET IDENTITY_INSERT [dbo].[CARGO] ON  
  
INSERT [dbo].[CARGO] ([COD_CAR], [NOM_CAR], [EST_CAR]) VALUES (1,  
N'ASISTENTE DE COMPRAS', N'A')  
  
INSERT [dbo].[CARGO] ([COD_CAR], [NOM_CAR], [EST_CAR]) VALUES (2,  
N'SOPORTE A USUARIOS', N'A')  
  
INSERT [dbo].[CARGO] ([COD_CAR], [NOM_CAR], [EST_CAR]) VALUES (3,  
N'COORDINADOR', N'A')  
  
SET IDENTITY_INSERT [dbo].[CARGO] OFF  
  
/***** Object: Table [dbo].[AREA] Script Date: 11/10/2013 20:24:02 *****/  
  
SET ANSI_NULLS ON  
  
GO  
  
SET QUOTED_IDENTIFIER ON  
  
GO  
  
SET ANSI_PADDING ON  
  
GO  
  
CREATE TABLE [dbo].[AREA](
```



```
[COD_ARE] [int] NOT NULL,  
[NOM_ARE] [char](50) NULL,  
[EST_ARE] [char](50) NULL,  
CONSTRAINT [PK_AREA] PRIMARY KEY NONCLUSTERED  
(  
    [COD_ARE] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]  
GO  
SET ANSI_PADDING OFF  
GO  
INSERT [dbo].[AREA] ([COD_ARE], [NOM_ARE], [EST_ARE]) VALUES (1,  
N'SISTEMAS', N'A')  
INSERT [dbo].[AREA] ([COD_ARE], [NOM_ARE], [EST_ARE]) VALUES (2,  
N'COMPRAS', N'A')  
/***** Object: Table [dbo].[EMPLEADOS]    Script Date: 11/10/2013 20:24:02  
*****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO
```




SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[EMPLEADOS](

[COD_EMP] [int] NOT NULL,

[COD_CAR] [int] NULL,

[COD_TIPEMP] [int] NULL,

[COD_ARE] [int] NULL,

[NOM_EMP] [char](50) NULL,

[NOM1_EMP] [char](50) NULL,

[APE_EMP] [char](50) NULL,

[APE1_EMP] [char](50) NULL,

[USU_EMP] [varchar](30) NULL,

[PASS_EMP] [varchar](8) NULL,

[EST_EMP] [char](10) NOT NULL,

[CED_EMP] [varchar](15) NULL,

[PRES_EMP] [decimal](18, 2) NULL,

[PRES_ACT_EMP] [decimal](18, 2) NULL,

CONSTRAINT [PK_EMPLADOS] PRIMARY KEY NONCLUSTERED

(

[COD_EMP] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,

IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,

ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]



) ON [PRIMARY]

GO

SET ANSI_PADDING OFF

GO

```
INSERT [dbo].[EMPLEADOS] ([COD_EMP], [COD_CAR], [COD_TIPEMP],  
[COD_ARE], [NOM_EMP], [NOM1_EMP], [APE_EMP], [APE1_EMP],  
[USU_EMP], [PASS_EMP], [EST_EMP], [CED_EMP], [PRES_EMP],  
[PRES_ACT_EMP]) VALUES (1, 1, 2, 1, N'JUAN',
```

```
N'CARLOS', N'FLORES',
```

```
N'PANTOJA', N'jc_flores', N'123', N'A',
```

```
N'1752678282', CAST(300.00 AS Decimal(18, 2)), CAST(300.00 AS Decimal(18, 2)))
```

```
INSERT [dbo].[EMPLEADOS] ([COD_EMP], [COD_CAR], [COD_TIPEMP],  
[COD_ARE], [NOM_EMP], [NOM1_EMP], [APE_EMP], [APE1_EMP],  
[USU_EMP], [PASS_EMP], [EST_EMP], [CED_EMP], [PRES_EMP],  
[PRES_ACT_EMP]) VALUES (2, 1, 1, 1, N'Nicole',
```

```
N'Alexandra', N'Argudo', N'Leroux
```

```
', N'na_argudo', N'123', N'E', N'1786789239', NULL, NULL)
```

```
INSERT [dbo].[EMPLEADOS] ([COD_EMP], [COD_CAR], [COD_TIPEMP],  
[COD_ARE], [NOM_EMP], [NOM1_EMP], [APE_EMP], [APE1_EMP],  
[USU_EMP], [PASS_EMP], [EST_EMP], [CED_EMP], [PRES_EMP],  
[PRES_ACT_EMP]) VALUES (4, 1, 1, 2, N'Martha',
```

```
N'Fernanda', N'Noquez', N'Tapia
```

```
', N'mf_noquez', N'123', N'A', N'1718946401', NULL, NULL)
```



/***** Object: Table [dbo].[SUMINISTROS_COMPRAS] Script Date: 11/10/2013

20:24:02 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

SET ANSI_PADDING ON

GO

CREATE TABLE [dbo].[SUMINISTROS_COMPRAS](

[COD_SUM] [int] NOT NULL,

[COD_TIPSUM] [int] NULL,

[NOM_SUM] [char](50) NULL,

[PRECIO] [decimal](18, 2) NULL,

[IVA] [decimal](18, 2) NULL,

[TOTAL] [decimal](18, 2) NULL,

[EST_SUM] [char](10) NULL,

CONSTRAINT [PK_SUMINISTROS] PRIMARY KEY NONCLUSTERED

(

[COD_SUM] ASC

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,

IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,

ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]



GO

SET ANSI_PADDING OFF

GO

```
INSERT [dbo].[SUMINISTROS_COMPRAS] ([COD_SUM], [COD_TIPSUM],  
[NOM_SUM], [PRECIO], [IVA], [TOTAL], [EST_SUM]) VALUES (1, 1, N'Resmas  
de Papel', CAST(1.50 AS Decimal(18, 2)), CAST(0.18 AS  
Decimal(18, 2)), CAST(1.68 AS Decimal(18, 2)), N'A')
```

```
INSERT [dbo].[SUMINISTROS_COMPRAS] ([COD_SUM], [COD_TIPSUM],  
[NOM_SUM], [PRECIO], [IVA], [TOTAL], [EST_SUM]) VALUES (2, 2, N'Te  
aromatico', CAST(2.00 AS Decimal(18, 2)), CAST(0.24 AS  
Decimal(18, 2)), CAST(2.24 AS Decimal(18, 2)), N'A')
```

```
INSERT [dbo].[SUMINISTROS_COMPRAS] ([COD_SUM], [COD_TIPSUM],  
[NOM_SUM], [PRECIO], [IVA], [TOTAL], [EST_SUM]) VALUES (3, 1, N'Esferos  
' , CAST(1.50 AS Decimal(18, 2)), CAST(0.18 AS Decimal(18, 2)), CAST(1.68 AS  
Decimal(18, 2)), N'A')
```

```
INSERT [dbo].[SUMINISTROS_COMPRAS] ([COD_SUM], [COD_TIPSUM],  
[NOM_SUM], [PRECIO], [IVA], [TOTAL], [EST_SUM]) VALUES (4, 1, N'Lapiz  
' , CAST(0.75 AS Decimal(18, 2)), CAST(0.09 AS Decimal(18, 2)), CAST(0.84 AS  
Decimal(18, 2)), N'A')
```

```
INSERT [dbo].[SUMINISTROS_COMPRAS] ([COD_SUM], [COD_TIPSUM],  
[NOM_SUM], [PRECIO], [IVA], [TOTAL], [EST_SUM]) VALUES (5, 1, N'Cartucho  
A150', CAST(150.00 AS Decimal(18, 2)), CAST(18.00 AS  
Decimal(18, 2)), CAST(168.00 AS Decimal(18, 2)), N'A')
```

```
INSERT [dbo].[SUMINISTROS_COMPRAS] ([COD_SUM], [COD_TIPSUM],  
[NOM_SUM], [PRECIO], [IVA], [TOTAL], [EST_SUM]) VALUES (6, 1, N'mouse  
inalambrico', CAST(13.00 AS Decimal(18, 2)), CAST(1.56 AS  
Decimal(18, 2)), CAST(14.56 AS Decimal(18, 2)), N'A')
```

```
INSERT [dbo].[SUMINISTROS_COMPRAS] ([COD_SUM], [COD_TIPSUM],  
[NOM_SUM], [PRECIO], [IVA], [TOTAL], [EST_SUM]) VALUES (7, 1, N'Cartucho  
A145', CAST(150.00 AS Decimal(18, 2)), CAST(18.00 AS  
Decimal(18, 2)), CAST(168.00 AS Decimal(18, 2)), N'A')
```

```
INSERT [dbo].[SUMINISTROS_COMPRAS] ([COD_SUM], [COD_TIPSUM],  
[NOM_SUM], [PRECIO], [IVA], [TOTAL], [EST_SUM]) VALUES (8, 2, N'Azucar  
100g', CAST(3.00 AS Decimal(18, 2)), CAST(0.36 AS  
Decimal(18, 2)), CAST(3.36 AS Decimal(18, 2)), N'A')
```

```
INSERT [dbo].[SUMINISTROS_COMPRAS] ([COD_SUM], [COD_TIPSUM],  
[NOM_SUM], [PRECIO], [IVA], [TOTAL], [EST_SUM]) VALUES (9, 2, N'Cafe 500g  
' , CAST(4.00 AS Decimal(18, 2)), CAST(0.48 AS Decimal(18, 2)), CAST(4.48 AS  
Decimal(18, 2)), N'A')
```

```
/***** Object: Table [dbo].[REQUERIMIENTO] Script Date: 11/10/2013 20:24:02
```

```
*****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
SET ANSI_PADDING ON
```



GO

```
CREATE TABLE [dbo].[REQUERIMIENTO](  
    [COD_REQ] [int] NOT NULL,  
    [COD_EMP] [int] NULL,  
    [COD_TIPSUM] [int] NULL,  
    [FEC_REQ] [datetime] NULL,  
    [OBS_REQ] [varchar](50) NULL,  
    [EST_REQ] [varchar](50) NULL,  
  
    CONSTRAINT [PK_REQUERIMIENTO] PRIMARY KEY NONCLUSTERED  
  
    (  
        [COD_REQ] ASC  
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
    ) ON [PRIMARY]
```

GO

SET ANSI_PADDING OFF

GO

```
INSERT [dbo].[REQUERIMIENTO] ([COD_REQ], [COD_EMP], [COD_TIPSUM],  
    [FEC_REQ], [OBS_REQ], [EST_REQ]) VALUES (1, 1, 1,  
    CAST(0x0000A24C00000000 AS DateTime), N'Pedido mes Septiembre', N'A    ')
```

```
INSERT [dbo].[REQUERIMIENTO] ([COD_REQ], [COD_EMP], [COD_TIPSUM],  
[FEC_REQ], [OBS_REQ], [EST_REQ]) VALUES (2, 1, 1,  
CAST(0x0000A26B00000000 AS DateTime), N'Perfecto', N'A')
```

```
INSERT [dbo].[REQUERIMIENTO] ([COD_REQ], [COD_EMP], [COD_TIPSUM],  
[FEC_REQ], [OBS_REQ], [EST_REQ]) VALUES (3, 1, 1,  
CAST(0x0000A26B00000000 AS DateTime), N'Perfecto', N'A')
```

```
/***** Object: Table [dbo].[PRESUPUESTO]   Script Date: 11/10/2013 20:24:02
```

```
*****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
SET ANSI_PADDING ON
```

```
GO
```

```
CREATE TABLE [dbo].[PRESUPUESTO](  
    [ID_PRE] [int] NULL,  
    [COD_EMP] [int] NULL,  
    [COD_SUM] [int] NULL,  
    [VALPRE] [decimal](18, 2) NULL,  
    [COD_REQ] [int] NULL,  
    [CAN_PRE] [decimal](18, 0) NOT NULL,  
    [PRE_ESTADO] [char](1) NULL  
) ON [PRIMARY]
```



GO

SET ANSI_PADDING OFF

GO

```
INSERT [dbo].[PRESUPUESTO] ([ID_PRE], [COD_EMP], [COD_SUM], [VALPRE],  
[COD_REQ], [CAN_PRE], [PRE_ESTADO]) VALUES (1, 1, 4, CAST(150.00 AS  
Decimal(18, 2)), 2, CAST(11 AS Decimal(18, 0)), N'0')
```

```
INSERT [dbo].[PRESUPUESTO] ([ID_PRE], [COD_EMP], [COD_SUM], [VALPRE],  
[COD_REQ], [CAN_PRE], [PRE_ESTADO]) VALUES (2, 1, 5, CAST(150.00 AS  
Decimal(18, 2)), 2, CAST(1 AS Decimal(18, 0)), N'0')
```

```
INSERT [dbo].[PRESUPUESTO] ([ID_PRE], [COD_EMP], [COD_SUM], [VALPRE],  
[COD_REQ], [CAN_PRE], [PRE_ESTADO]) VALUES (3, 1, 4, CAST(150.00 AS  
Decimal(18, 2)), 3, CAST(12 AS Decimal(18, 0)), N'0')
```

```
INSERT [dbo].[PRESUPUESTO] ([ID_PRE], [COD_EMP], [COD_SUM], [VALPRE],  
[COD_REQ], [CAN_PRE], [PRE_ESTADO]) VALUES (4, 1, 5, CAST(150.00 AS  
Decimal(18, 2)), 3, CAST(1 AS Decimal(18, 0)), N'0')
```

/***** Object: ForeignKey [FK_EMPLEADO_RELATIONS_AREA] Script Date:
11/10/2013 20:24:02 *****/

```
ALTER TABLE [dbo].[EMPLEADOS] WITH CHECK ADD CONSTRAINT  
[FK_EMPLEADO_RELATIONS_AREA] FOREIGN KEY([COD_ARE])  
REFERENCES [dbo].[AREA] ([COD_ARE])
```

GO

```
ALTER TABLE [dbo].[EMPLEADOS] CHECK CONSTRAINT  
[FK_EMPLEADO_RELATIONS_AREA]
```




GO

/***** Object: ForeignKey [FK_EMPLEADO_RELATIONS_CARGO] Script

Date: 11/10/2013 20:24:02 *****/

ALTER TABLE [dbo].[EMPLEADOS] WITH CHECK ADD CONSTRAINT

[FK_EMPLEADO_RELATIONS_CARGO] FOREIGN KEY([COD_CAR])

REFERENCES [dbo].[CARGO] ([COD_CAR])

GO

ALTER TABLE [dbo].[EMPLEADOS] CHECK CONSTRAINT

[FK_EMPLEADO_RELATIONS_CARGO]

GO

/***** Object: ForeignKey [FK_EMPLEADO_RELATIONS_TIPO_EMP] Script

Date: 11/10/2013 20:24:02 *****/

ALTER TABLE [dbo].[EMPLEADOS] WITH CHECK ADD CONSTRAINT

[FK_EMPLEADO_RELATIONS_TIPO_EMP] FOREIGN KEY([COD_TIPEMP])

REFERENCES [dbo].[TIPO_EMPLEADO] ([COD_TIPEMP])

GO

ALTER TABLE [dbo].[EMPLEADOS] CHECK CONSTRAINT

[FK_EMPLEADO_RELATIONS_TIPO_EMP]

GO

/***** Object: ForeignKey [FK_PRESUPUESTO_REQUERIMIENTO] Script

Date: 11/10/2013 20:24:02 *****/

ALTER TABLE [dbo].[PRESUPUESTO] WITH CHECK ADD CONSTRAINT

[FK_PRESUPUESTO_REQUERIMIENTO] FOREIGN KEY([COD_REQ])

```
REFERENCES [dbo].[REQUERIMIENTO] ([COD_REQ])
```

```
GO
```

```
ALTER TABLE [dbo].[PRESUPUESTO] CHECK CONSTRAINT
```

```
[FK_PRESUPUESTO_REQUERIMIENTO]
```

```
GO
```

```
/***** Object: ForeignKey [FK_PRESUPUESTO_SUMINISTROS_COMPRAS]
```

```
Script Date: 11/10/2013 20:24:02 *****/
```

```
ALTER TABLE [dbo].[PRESUPUESTO] WITH CHECK ADD CONSTRAINT
```

```
[FK_PRESUPUESTO_SUMINISTROS_COMPRAS] FOREIGN KEY([COD_SUM])
```

```
REFERENCES [dbo].[SUMINISTROS_COMPRAS] ([COD_SUM])
```

```
GO
```

```
ALTER TABLE [dbo].[PRESUPUESTO] CHECK CONSTRAINT
```

```
[FK_PRESUPUESTO_SUMINISTROS_COMPRAS]
```

```
GO
```

```
/***** Object: ForeignKey [FK_REQUERIM_RELATIONS_EMPLEADO] Script
```

```
Date: 11/10/2013 20:24:02 *****/
```

```
ALTER TABLE [dbo].[REQUERIMIENTO] WITH CHECK ADD CONSTRAINT
```

```
[FK_REQUERIM_RELATIONS_EMPLEADO] FOREIGN KEY([COD_EMP])
```

```
REFERENCES [dbo].[EMPLEADOS] ([COD_EMP])
```

```
GO
```

```
ALTER TABLE [dbo].[REQUERIMIENTO] CHECK CONSTRAINT
```

```
[FK_REQUERIM_RELATIONS_EMPLEADO]
```

```
GO
```



/***** Object: ForeignKey [FK_SUMINIST_RELATIONS_TIPO_SUM] Script

Date: 11/10/2013 20:24:02 *****/

```
ALTER TABLE [dbo].[SUMINISTROS_COMPRAS] WITH CHECK ADD
CONSTRAINT [FK_SUMINIST_RELATIONS_TIPO_SUM] FOREIGN
KEY([COD_TIPSUM])
REFERENCES [dbo].[TIPO_SUMINISTROS] ([COD_TIPSUM])
GO

ALTER TABLE [dbo].[SUMINISTROS_COMPRAS] CHECK CONSTRAINT
[FK_SUMINIST_RELATIONS_TIPO_SUM]
GO
```

A.02.04 Lógica de negocio

A.02.04.01 Cargo Lógica

`namespace` LogicaNegocio

```
{

    public class CargoLogica
    {

        public static SuministrosBDDDataContext dc = new SuministrosBDDDataContext();

        public static List<CARGO> ObtenerCargo()
        {
            var lista = dc.CARGO.Where(cargo => cargo.EST_CAR == "A");

            return lista.ToList();
        }

        public static List<CARGO> ObtenerCargoId(int id)
```



```
{  
  
    var lista = dc.CARGO.Where(cargo => cargo.EST_CAR == "A" &  
cargo.COD_CAR.Equals(id));  
  
    return lista.ToList();  
  
}  
  
public static List<CARGO> ObtenerCargoNombre(string Nombre)  
  
{  
  
    var lista = dc.CARGO.Where(cargo => cargo.EST_CAR == "A" &  
cargo.NOM_CAR.Equals(Nombre));  
  
    return lista.ToList();  
  
}  
  
public static CARGO ObtenerCargoIdinfo(int id)  
  
{  
  
    var lista = dc.CARGO.FirstOrDefault(cargo => cargo.EST_CAR == "A" &  
cargo.COD_CAR.Equals(id));  
  
    return lista;  
  
}  
  
private static int Id_Codigo()  
  
{  
  
    try  
  
    {  
  
        return dc.CARGO.Max(cargo => cargo.COD_CAR + 1);  
  
    }  
  
}
```



```
        catch (Exception)

        {

            return 1;

        }

    }

    public static void Save(CARGO CargoInfo)

    {

        try

        {

            CargoInfo.EST_CAR = "A";

            CargoInfo.COD_CAR = Id_Codigo();

            //Validacion(ProductoInfo);

            dc.CARGO.InsertOnSubmit(CargoInfo);

            dc.SubmitChanges();

        }

        catch (Exception ex)

        {

            throw new ArgumentException("Datos No Guardados: </br>" + ex.Message);

        }

    }

    public static void Modify(CARGO CargoInfo)

    {

        try

        {
```



```
//Validacion(ProductoInfo);

dc.SubmitChanges();

}

catch (Exception ex)

{

    throw new ArgumentException("Datos No Modificados: </br>" +

ex.Message);

}

}

public static void Delete(CARGO CargoInfo)

{

    try

    {

        CargoInfo.EST_CAR = "E";

        dc.SubmitChanges();

    }

    catch (Exception ex)

    {

        throw new ArgumentException("Datos No Eliminados: </br>" + ex.Message);

    }

}

} } }
```

A.02.04.02 Empleado Lógica

```
namespace LogicaNegocio
```



```
{  
  
    public class EmpleadosLogica  
    {  
  
        public static SuministrosBDDDataContext dc = new SuministrosBDDDataContext();  
  
        public static List<EMPLEADOS> ObtenerEmpleadosLista()  
        {  
  
            var lista = dc.EMPLEADOS.Where(empleado => empleado.EST_EMP == "A");  
            return lista.ToList();  
        }  
  
        public static List<EMPLEADOS> ObtenerListaEmpleadosXId(int idEmp)  
        {  
  
            var lisAlu = dc.EMPLEADOS.Where(empleado => empleado.EST_EMP ==  
            "A" &  
  
                empleado.COD_EMP.Equals(idEmp));  
  
            return lisAlu.ToList();  
        }  
  
        public static List<EMPLEADOS> ObtenerEmpleadosXCedula(string Ced)  
        {  
  
            var lista = dc.EMPLEADOS.Where(empleado => empleado.EST_EMP == "A"  
            & empleado.CED_EMP.Equals(Ced));  
  
            return lista.ToList();  
        }  
    }  
}
```



```
//info//

public static EMPLEADOS ObtenerEmpleadosXCedInfo(string Ced)
{
    var empleado = dc.EMPLEADOS.FirstOrDefault(emp => emp.EST_EMP ==
    "A" & emp.CED_EMP.Equals(Ced));

    return empleado;
}

public static EMPLEADOS ObtenerEmpleadosXCodInfo(int cod)
{
    var empleado = dc.EMPLEADOS.FirstOrDefault(emp => emp.EST_EMP ==
    "A" & emp.COD_EMP.Equals(cod));

    return empleado;
} //login//

public static bool AutenticarEmpleado(string USU_EMPlogin, string
PASS_EMPlogin)
{
    var existe = dc.EMPLEADOS.Any(emp => emp.EST_EMP == "A" &
    emp.USU_EMP.Equals(USU_EMPlogin) &
    emp.PASS_EMP.Equals(PASS_EMPlogin));

    return existe;
}
```




```
public static EMPLEADOS UsuarioActual(string USU_EMPlogin, string
PASS_EMPlogin)
{
    var Actual =dc.EMPLEADOS.FirstOrDefault(emp => emp.EST_EMP == "A" &
emp.USU_EMP.Equals(USU_EMPlogin) &
emp.PASS_EMP.Equals(PASS_EMPlogin));

    return Actual;
}

private static int Id_AutoIncrementableEmpleados()
{
    try
    {
        return dc.EMPLEADOS.Max(empleados => empleados.COD_EMP + 1);
    }
    catch (Exception ex)
    {
        return 1;
    }
}

public static void save(EMPLEADOS EmpleadoInfo)
{
    try
    {
```



```
dc.ExecuteCommand("insert into EMPLEADOS  
(COD_EMP,CED_EMP,COD_CAR,COD_TIPEMP,COD_ARE, " +  
  
"NOM_EMP,NOM1_EMP,APE_EMP,APE1_EMP,EST_EMP" +  
") values ({0},{1},{2},{3},{4},{5},{6},{7},{8},{9}) ", new object[]  
  
{  
EmpleadoInfo.COD_EMP = Id_AutoIncrementableEmpleados(),  
EmpleadoInfo.CED_EMP,  
EmpleadoInfo.COD_CAR,  
EmpleadoInfo.COD_TIPEMP,  
EmpleadoInfo.COD_ARE,  
EmpleadoInfo.NOM_EMP,  
EmpleadoInfo.NOM1_EMP,  
EmpleadoInfo.APE_EMP,  
EmpleadoInfo.APE1_EMP,  
EmpleadoInfo.EST_EMP="A"  
});  
}  
catch (Exception ex)  
{  
    throw new ArgumentException("Datos No Guardados </br>" + ex.Message);  
}  
}
```



```
public static void modify(EMPLEADOS EmpleadoInfo)
{
    try
    {
        dc.ExecuteCommand("Update EMPLEADOS set
COD_CAR={0},COD_TIPEMP={1},COD_ARE={2},NOM_EMP={3},NOM1_EMP=
{4},APE_EMP={5},APE1_EMP={6},EST_EMP={7},CED_EMP={8} where
COD_EMP={9}", new object[]
{
    EmpleadoInfo.CED_EMP,
    EmpleadoInfo.COD_CAR,
    EmpleadoInfo.COD_TIPEMP,
    EmpleadoInfo.COD_ARE,
    EmpleadoInfo.NOM_EMP,
    EmpleadoInfo.NOM1_EMP,
    EmpleadoInfo.APE_EMP,
    EmpleadoInfo.APE1_EMP,
    EmpleadoInfo.EST_EMP="A",
    EmpleadoInfo.CED_EMP
});
    }
    catch (Exception ex)
    {

```



```
        throw new ArgumentException("Datos Modificados </br>" + ex.Message);
    }
}

public static void NoModify(EMPLEADOS EmpleadoInfo)
{
    try
    {
        dc.SubmitChanges();
    }
    catch (Exception ex)
    {
        throw new ArgumentException("No se puede modificar los datos" +
ex.Message);
    }
}

public static void Delete(EMPLEADOS EmpleadoInfo)
{
    try
    {
        EmpleadoInfo.EST_EMP = "E";
        dc.SubmitChanges();
    }
    catch (Exception ex)
```



```
{  
    throw new ArgumentException("no se puede eliminar los datos" +  
ex.Message);  
}  
}  
  
public static bool existe(string nombre, string pass)  
{  
    var exist = dc.EMPLEADOS.Any(existe => existe.EST_EMP == "A" &  
existe.USU_EMP.Equals(nombre)  
        & existe.PASS_EMP.Equals(pass));  
  
    return exist;}  
  
public static EMPLEADOS EmpleadoActual(string nombre, string pass)  
{  
    var exist = dc.EMPLEADOS.FirstOrDefault(existe => existe.EST_EMP == "A"  
& existe.USU_EMP.Equals(nombre)  
        & existe.PASS_EMP.Equals(pass));  
  
    return exist;  
}  
  
public static bool CedulaCorrecta(string txtcedula)  
{  
    int Esnumero;  
  
    var total = 0;  
  
    const int LongitudCedula = 10;
```



```
int[] coeficientes = { 2, 1, 2, 1, 2, 1, 2, 1, 2 };

if (int.TryParse(txtcedula, out Esnumero) && txtcedula.Length ==
LongitudCedula)
{
    var provincia = Convert.ToInt32(string.Concat(txtcedula[0], txtcedula[1],
string.Empty));

    var digitoTres = Convert.ToInt32(txtcedula[2] + string.Empty);

    if (provincia > 0 && provincia <= 24)
    {
        var digitoverificador = Convert.ToInt32(txtcedula[9] + string.Empty);

        for (var i = 0; i < coeficientes.Length; i++)
        {
            var valor = Convert.ToInt32(coeficientes[i] + string.Empty) *
Convert.ToInt32(txtcedula[i] + string.Empty);

            total = valor >= 10 ? total + (valor - 9) : total + valor;

        }

        var digitoverificadorobtenido = total >= 10 ? (total % 10) != 0 ? 10 - (total
% 10) : (total % 10) : total;

        return digitoverificadorobtenido == digitoverificador;

    }

}

return false;

}
```



```
public static List<EMPLEADOS> ObtenerEmpleadosXNombre(string Nombre)
{
    var lista = dc.EMPLEADOS.Where(empleado => empleado.EST_EMP == "A"
& empleado.NOM_EMP.StartsWith(Nombre));

    return lista.ToList();
}

public static List<EMPLEADOS> ObtenerEmpleadosXNombre1(string Nombre1)
{
    var lista = dc.EMPLEADOS.Where(empleado => empleado.EST_EMP == "A"
& empleado.NOM1_EMP.StartsWith(Nombre1));

    return lista.ToList();
}

public static List<EMPLEADOS> ObtenerEmpleadosXApellido(string Apellido)
{
    var lista = dc.EMPLEADOS.Where(empleado => empleado.EST_EMP == "A"
& empleado.APE_EMP.StartsWith(Apellido));

    return lista.ToList();
}

public static List<EMPLEADOS> ObtenerEmpleadosXApellido1(string
Apellido1)
{
    var lista = dc.EMPLEADOS.Where(empleado => empleado.EST_EMP == "A"
& empleado.APE1_EMP.StartsWith(Apellido1));
```



```
return lista.ToList();
```

```
} } }
```

A.02.04.03 Lógica Suministros

```
namespace LogicaNegocio
```

```
{
```

```
public class Logica_Suministro
```

```
{
```

```
public static SuministrosBDDDataContext dc = new SuministrosBDDDataContext();
```

```
public static List<SUMINISTROS_COMPRAS> ObtenerListaSuministros()
```

```
{
```

```
var SumLis = dc.SUMINISTROS_COMPRAS.Where(sum => sum.EST_SUM  
== "A");
```

```
return SumLis.ToList();
```

```
}
```

```
public static List<SUMINISTROS_COMPRAS>
```

```
ObtenerListaSuministrosXTipo(int tipo)
```

```
{
```

```
var SumLis = dc.SUMINISTROS_COMPRAS.Where(sum => sum.EST_SUM  
== "A" & sum.COD_TIPSUM.Equals(tipo));
```

```
return SumLis.ToList();
```

```
}
```

```
public static List<SUMINISTROS_COMPRAS> ObtenerListaSuministrosXCod(int
```

```
Cod)
```




```
{  
    var Sumlis = dc.SUMINISTROS_COMPRAS.Where(sum => sum.EST_SUM ==  
    "A" &  
        sum.COD_SUM.Equals(Cod));  
    return Sumlis.ToList();  
}  
  
Public static List<SUMINISTROS_COMPRAS>  
ObtenerListaSuministrosXNombre(string nombre)  
{  
    var SumNom = dc.SUMINISTROS_COMPRAS.Where(sum => sum.EST_SUM  
    == "A" &  
        sum.NOM_SUM.Equals(nombre));  
    return SumNom.ToList();  
}  
  
public static SUMINISTROS_COMPRAS ObtenerSuministrosInfoXCod(int Cod)  
{  
    var infoSum = dc.SUMINISTROS_COMPRAS.FirstOrDefault(sum =>  
sum.EST_SUM == "A" &  
        sum.COD_SUM.Equals(Cod));  
    return infoSum;  
}  
  
private static int Id_AutoIncrementableSuministro()
```



```
{  
    try  
    {  
        return dc.SUMINISTROS_COMPRAS.Max(sum => sum.COD_SUM + 1);  
    }  
    catch (Exception ex)  
    {  
        return 1;  
    }  
}  
  
public static void ModificarSuministros(SUMINISTROS_COMPRAS infoSum)  
{  
    try  
    {  
        infoSum.EST_SUM = "A";  
        dc.ExecuteCommand("Update SUMINISTROS_COMPRAS Set  
COD_TIPSUM={0},NOM_SUM={1}" +  
        ",PRECIO={2},IVA={3},TOTAL={4},EST_SUM={5} Where  
COD_SUM={6}", new object[]  
        {  
            infoSum.COD_TIPSUM,  
            infoSum.NOM_SUM,  
            infoSum.PRECIO,  
            infoSum.IVA,
```



```
        infoSum.TOTAL,
        infoSum.EST_SUM,
        infoSum.COD_SUM
    });

    dc.Refresh(RefreshMode.OverwriteCurrentValues, infoSum);
}

catch (Exception ex)
{
    throw new ArgumentException("Error al modificar los datos<p>" +
ex.Message);
}
}

public static void Delete(SUMINISTROS_COMPRAS infoSum)
{
    try
    {
        infoSum.EST_SUM = "E";
        dc.SubmitChanges();
    }

    catch (Exception ex)
    {
        throw new ArgumentException("no se puede eliminar los datos" +
ex.Message);
    }
}
```



```
}  
  
}  
  
public static void save(SUMINISTROS_COMPRAS infoSum)  
{  
    try  
    {  
        dc.ExecuteCommand("Insert into  
SUMINISTROS_COMPRAS(COD_SUM,COD_TIPSUM,NOM_SUM,PRECIO," +  
        "IVA,TOTAL,EST_SUM)" +  
        "values ({0},{1},{2},{3},{4},{5},{6})", new object[]  
        {  
            infoSum.COD_SUM = Id_AutoIncrementableSuministro(),  
            infoSum.COD_TIPSUM,  
            infoSum.NOM_SUM,  
            infoSum.PRECIO,  
            infoSum.IVA,  
            infoSum.TOTAL,  
            infoSum.EST_SUM = "A"  
        });  
    }  
    catch (Exception ex)  
    {  
        throw new ArgumentException("Datos No Guardados </br>" + ex.Message);  
    }  
}
```



```
    }  
}  
  
public static void NoModify(SUMINISTROS_COMPRAS infoSum)  
{  
    try  
    {  
        dc.SubmitChanges();  
    }  
    catch (Exception ex)  
    {  
        throw new ArgumentException("No se puede modificar los datos" +  
ex.Message);  
    } } }
```

A.02.04.04 Presupuesto Lógica

```
namespace LogicaNegocio  
{  
    public class PresupuestoLogica  
    {  
        public static SuministrosBDDDataContext dc = new SuministrosBDDDataContext();  
        public static List<PRESUPUESTO> ObtenerPresupuestoXIdrequer(int  
idrequerimiento)  
    {
```



```
var lista = dc.PRESUPUESTO.Where(p => p.PRE_ESTADO == 'A' &
p.COD_REQ.Equals(idrequerimiento));

return lista.ToList();
}

public static List<PRESUPUESTO> ObtenerPresupuestoXIdPRE(int idPRE)
{
var lista = dc.PRESUPUESTO.Where(p => p.PRE_ESTADO == 'A' &
p.ID_PRE.Equals(idPRE));

return lista.ToList();
}

public static int Id_Codigo()
{
try
{
return dc.PRESUPUESTO.Max(p => Convert.ToInt32(p.ID_PRE + 1));
}

catch (Exception)
{
return 1;
}
}

public static void Save(PRESUPUESTO Presupuestoinfo)
{
```



```
try
{
    dc.ExecuteCommand("Insert into PRESUPUESTO(ID_PRE, COD_EMP,
COD_SUM, VALPRE, COD_REQ, CAN_PRE, PRE_ESTADO)" +
    "values({0},{1},{2},{3},{4},{5},{6})", new object[]
    {
        Presupuestoinfo.ID_PRE = Id_Codigo(),
        Presupuestoinfo.COD_EMP,
        Presupuestoinfo.COD_SUM,
        Presupuestoinfo.VALPRE,
        Presupuestoinfo.COD_REQ,
        Presupuestoinfo.CAN_PRE,
        Presupuestoinfo.PRE_ESTADO == 'A'
    }
    );
}

catch (Exception ex)
{
    throw new ArgumentException("Datos no Guardados<br>" + ex.Message);
}

}

public static void Modify(PRESUPUESTO Presupuestoinfo)
{
    try
    {
```



```
dc.ExecuteCommand("Update PRESUPUESTO set COD_EMP={0},  
COD_SUM={1}, VALPRE={2}, COD_REQ={3}. CAN_PRE={4},  
PRE_ESTADO={5} where ID_PRE={6}", new object[]  
  
    {  
        Presupuestoinfo.COD_EMP,  
        Presupuestoinfo.COD_SUM,  
        Presupuestoinfo.VALPRE,  
        Presupuestoinfo.COD_REQ,  
        Presupuestoinfo.CAN_PRE,  
        Presupuestoinfo.PRE_ESTADO,  
        Presupuestoinfo.ID_PRE  
    }  
    );  
}  
  
catch (Exception ex)  
  
    {  
        throw new ArgumentException("Datos no Modificados<br>" + ex.Message);  
    }  
}  
  
public static void Delete(PRESUPUESTO Presupuestoinfo)  
  
    {  
        try  
  
        {  
  
            dc.ExecuteCommand("Update PRESUPUESTO set PRE_ESTADO={0}  
where ID_PRE={1}", new object[]
```




```
{  
    Presupuestoinfo.PRE_ESTADO = 'E',  
    Presupuestoinfo.ID_PRE  
}  
);  
}  
catch (Exception ex)  
{  
    throw new ArgumentException("Datos no Eliminados<br>" + ex.Message);  
}    }    }}
```

A.02.04.05 Requerimiento Lógica

```
namespace LogicaNegocio  
{  
    public class RequerimientoLogica  
    {  
        public static SuministrosBDDDataContext dc = new SuministrosBDDDataContext();  
        public static List<REQUERIMIENTO> ObtenerRequerimientoLista()  
        {  
            var lista = dc.REQUERIMIENTO.Where(req => req.EST_REQ == "A");  
            return lista.ToList();  
        }  
    }  
}
```



```
public static List<REQUERIMIENTO> ObtenerListaRequerimientoXId(int
idReq)
{
    var lista = dc.REQUERIMIENTO.Where(req => req.EST_REQ == "A" &
        req.COD_REQ.Equals(idReq));

    return lista.ToList();
}

//public static List<REQUERIMIENTO> ObtenerListaCantidadXId(int idCant)
//{
//    var Lista = dc.REQUERIMIENTO.Where(req => req.EST_REQ == "A" &
req.CAN_REQ.Equals(idCant));
//    return Lista.ToList();
//}
//Info

public static REQUERIMIENTO ObtenerRequerimientoXIdInfo(int id)
{
    var requere = dc.REQUERIMIENTO.FirstOrDefault(req => req.EST_REQ ==
"A" & req.COD_REQ.Equals(id));

    return requere;
}

//public static REQUERIMIENTO ObtenerCantidadXIdInfo(int id)
//{
```



```
// var reque = dc.REQUERIMIENTO.FirstOrDefault(req => req.EST_REQ ==  
"A" & req.CAN_REQ.Equals(id));  
  
// return reque;  
//}  
  
private static int Id_AutoIncrementableRequerimiento()  
{  
    try  
    {  
        return dc.REQUERIMIENTO.Max(req => req.COD_REQ + 1);  
    }  
    catch (Exception ex)  
    {  
        return 1;  
    }  
}  
  
Public static int save(REQUERIMIENTO RequerInfo)  
{  
    try  
    {  
        dc.ExecuteCommand("insert into REQUERIMIENTO  
(COD_REQ,COD_EMP,COD_TIPSUM,FEC_REQ, " +  
        "OBS_REQ,EST_REQ" +  
        ") values ({0},{1},{2},{3},{4},{5}) ", new object[]  
        {
```



```
RequerInfo.COD_REQ = Id_AutoIncrementableRequerimiento(),
    RequerInfo.COD_EMP,
    RequerInfo.COD_TIPSUM,
    RequerInfo.FEC_REQ,
    RequerInfo.OBS_REQ,
    RequerInfo.EST_REQ="A"
    });

return RequerInfo.COD_REQ;
}

catch (Exception ex)
{
    throw new ArgumentException("Datos No Guardados </br>" + ex.Message);
}
}

public static void modify(REQUERIMIENTO RequerInfo)
{
    try
    {
        dc.ExecuteCommand("Update REQUERIMIENTO set
COD_EMP={0},COD_TIPSUM={1},FEC_REQ={2},OBS_REQ={3},EST_REQ={4}
where COD_REQ={5}", new object[]
{

```



```
RequerInfo.COD_EMP,
    RequerInfo.COD_TIPSUM,
    RequerInfo.FEC_REQ,
    RequerInfo.OBS_REQ,
    RequerInfo.EST_REQ="A",
    RequerInfo.COD_REQ
    });
}
catch (Exception ex)
{
    throw new ArgumentException("Datos Modificados </br>" + ex.Message);
}
}
public static void NoModify(REQUERIMIENTO RequerInfo)
{
    try
    {
        dc.SubmitChanges();
    }
    catch (Exception ex)
    {
        throw new ArgumentException("No se puede modificar los datos" +
ex.Message);
```



```
    }  
}  
  
public static void Delete(REQUERIMIENTO RequerInfo)  
{  
    try  
    {  
        RequerInfo.EST_REQ = "E";  
        dc.SubmitChanges();  
    }  
    catch (Exception ex)  
    {  
        throw new ArgumentException("no se puede eliminar los datos" +  
ex.Message);  
    }  
}  
}
```

A.02.04.06 Tipo Empleado Logica

```
namespace LogicaNegocio
```

```
{  
  
    public class TipoEmpleadosLogica
```



```
{  
  
    public static SuministrosBDDDataContext dc = new SuministrosBDDDataContext();  
  
    public static List<TIPO_EMPLEADO> obtenerTipoEmpleado()  
    {  
  
        var lista = dc.TIPO_EMPLEADO.Where(tipoempleado =>  
tipoempleado.EST_TIPEMP == "A");  
  
        return lista.ToList();  
    }  
  
    public static List<TIPO_EMPLEADO> ObtenerTipoEmpleadoId(int id)  
    {  
  
        var lista = dc.TIPO_EMPLEADO.Where(tipoempleado =>  
tipoempleado.EST_TIPEMP == "A" & tipoempleado.COD_TIPEMP.Equals(id));  
  
        return lista.ToList();  
    }  
  
    public static List<TIPO_EMPLEADO> ObtenerTipoEmpleadoXNombre(string  
Nombre)  
    {  
  
        var lista = dc.TIPO_EMPLEADO.Where(tipoempleado =>  
tipoempleado.EST_TIPEMP == "A" &  
tipoempleado.NOM_TIPEMP.Equals(Nombre));  
  
        return lista.ToList();  
    }  
}
```



```
public static TIPO_EMPLEADO ObtenerTipoEmpleadoIdinfo(int id)
{
    var lista = dc.TIPO_EMPLEADO.FirstOrDefault(tipoempleado =>
tipoempleado.EST_TIPEMP == "A" & tipoempleado.COD_TIPEMP.Equals(id));

    return lista;
}

private static int Id_Codigo()
{
    try
    {
        return dc.TIPO_EMPLEADO.Max(tipoempleado =>
tipoempleado.COD_TIPEMP + 1);
    }
    catch (Exception)
    {
        return 1;
    }
}

public static void Save(TIPO_EMPLEADO TipoEmpleadoInfo)
{
    try
    {
        TipoEmpleadoInfo.EST_TIPEMP = "A";
    }
}
```




```
TipoEmpleadoInfo.COD_TIPEMP = Id_Codigo();

//Validacion(ProductoInfo);

dc.TIPO_EMPLEADO.InsertOnSubmit(TipoEmpleadoInfo);

dc.SubmitChanges();

}

catch (Exception ex)

{

    throw new ArgumentException("Datos No Guardados: </br>" + ex.Message);

}

}

public static void Modify(TIPO_EMPLEADO TipoEmpleadoInfo)

{

    try

    {

        //Validacion(ProductoInfo);

        dc.SubmitChanges();

    }

    catch (Exception ex)

    {

        throw new ArgumentException("Datos No Modificados: </br>" +

ex.Message);

    }

}
```



```
public static void Delete(TIPO_EMPLEADO TipoEmpleadoInfo)
{
    try
    {
        TipoEmpleadoInfo.EST_TIPEMP = "E";
        dc.SubmitChanges();
    }
    catch (Exception ex)
    {
        throw new ArgumentException("Datos No Eliminados: </br>" + ex.Message);
    }
}
}
```

A.02.04.07 Tipo Suministro Logica

```
namespace LogicaNegocio
```

```
{
    public class TipoSuministros_Logica
    {
        public static SuministrosBDDDataContext dc = new SuministrosBDDDataContext();
        public static List<TIPO_SUMINISTROS> ObtenerSuministros()
        {
```



```
        var lista = dc.TIPO_SUMINISTROS.Where(tipoSum =>
tipoSum.EST_TIPSUM == "A");

        return lista.ToList();
    }

    public static List<TIPO_SUMINISTROS> ObtenerTipoSuministroId(int id)
    {
        var lista = dc.TIPO_SUMINISTROS.Where(tipoSum =>
tipoSum.EST_TIPSUM == "A" & tipoSum.COD_TIPSUM.Equals(id));

        return lista.ToList();
    }

    public static List<TIPO_SUMINISTROS> ObtenerTipoSuministroXNombre(string
Nombre)
    {
        var lista = dc.TIPO_SUMINISTROS.Where(tipoSum =>
tipoSum.EST_TIPSUM == "A" & tipoSum.NOM_TIPSUM.Equals(Nombre));

        return lista.ToList();
    }

    public static TIPO_SUMINISTROS ObtenerTipoSuministroXIdinfo(int cod)
    {
        var infoSum = dc.TIPO_SUMINISTROS.FirstOrDefault(tipoSum =>
tipoSum.EST_TIPSUM == "A" & tipoSum.COD_TIPSUM.Equals(cod));

        return infoSum;
    }
}
```



```
private static int Id_Codigo()
{
    try
    {
        return dc.TIPO_SUMINISTROS.Max(tipoSum => tipoSum.COD_TIPSUM +
1);
    }
    catch (Exception)
    {
        return 1;
    }
}

public static void Save(TIPO_SUMINISTROS infoSum)
{
    try
    {
        infoSum.EST_TIPSUM = "A";
        infoSum.COD_TIPSUM = Id_Codigo();
        //Validacion(ProductoInfo);
        dc.TIPO_SUMINISTROS.InsertOnSubmit(infoSum);
        dc.SubmitChanges();
    }
    catch (Exception ex)
```



```
{

    throw new ArgumentException("Datos No Guardados: </br>" + ex.Message);

}

}

public static void Modify(TIPO_SUMINISTROS infoSum)
{
    try
    {
        //Validacion(ProductoInfo);

        dc.SubmitChanges();
    }

    catch (Exception ex)
    {
        throw new ArgumentException("Datos No Modificados: </br>" +
ex.Message);
    }
}

public static void Delete(TIPO_SUMINISTROS infoSum)
{
    try
    {
        infoSum.EST_TIPSUM = "E";
```



```
dc.SubmitChanges();  
  
}  
  
catch (Exception ex)  
{  
  
    throw new ArgumentException("Datos No Eliminados: </br>" + ex.Message);  
  
}  
  
}  
  
}  
  
}
```

A.02.05 Capa de datos

```
Private void general_load(object sender, eventargs e)  
{  
  
    Base_datos.opendb();  
  
    If (var.r == true)  
  
    {  
  
        MessageBox.show("conectado a sql", "");  
  
    }  
  
    Else  
  
    {  
  
        Var.mensaje = "no se ha establecido la \n";  
  
        Var.mensaje = var.mensaje + "conección con la base \n";  
  
        Var.mensaje = var.mensaje + "de datos \n";  
  
        MessageBox.show(var.mensaje, "conexion",
```



```
messageboxbuttons.ok, messageboxicon.error);} }
```

A.02.06 Programación de ventanas.

A continuación se detalla el código de la ventana Requerimientos del sistema.

Using system;

Requerimiento Lista

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Web;
```

```
using System.Web.UI;
```

```
using System.Web.UI.WebControls;
```

```
using AccesoDatos;
```

```
using LogicaNegocio;
```

```
using System.Data.Linq;
```

```
namespace SUMINISTROS.Administador.Requerimientos
```

```
{
```

```
    public partial class RequerimientoLista : System.Web.UI.Page
```

```
    {
```

```
        List<REQUERIMIENTO> RequerimientoList = new List<REQUERIMIENTO>();
```

```
        REQUERIMIENTO RequerInfo = new REQUERIMIENTO();
```

```
        protected void Page_Load(object sender, EventArgs e)
```



```
{  
    if (!IsPostBack)  
    {  
        CargarRequerimiento();  
    }  
}  
  
private void CargarRequerimiento()  
{  
    RequerimientoList = RequerimientoLogica.ObtenerRequerimientoLista();  
    if (RequerimientoList != null)  
    {  
        grd_Requerimiento.DataSource = RequerimientoList;  
        grd_Requerimiento.DataBind();  
    }  
}  
  
private void CargarDgr(List<REQUERIMIENTO> Lista)  
{  
    if (Lista != null & Lista.Count > 0)  
    {  
        grd_Requerimiento.DataSource = Lista;  
        grd_Requerimiento.DataBind();  
    }  
}
```




```
private void CargarFormulario()
{
    try
    {
        RequerimientoList = new List<REQUERIMIENTO>();
        RequerimientoList = RequerimientoLogica.ObtenerRequerimientoLista();
        CargarDgr(RequerimientoList);
    }
    catch (Exception)
    {
        throw;
    }
}

protected void ImgNuevo_Click(object sender, ImageClickEventArgs e)

Response.Redirect("~/Administrador/Requerimientos/RequerimientosNuevo.aspx",
true);

}

protected void grd_Requerimiento_ItemCommand(object source,
DataGridCommandEventArgs e)
{
    try
```



```
{  
  
    int COD_REQ =  
  
    Convert.ToInt32(((Label)e.Item.Cells[1].FindControl("lblRequerimiento")).Text);  
  
    if (e.CommandName == "editar")  
    {  
  
        Response.Redirect("~/Administrador/Requerimientos/RequerimientosNuevo.aspx?COD  
_REQ=" + COD_REQ.ToString(), true);  
  
    }  
  
    if (e.CommandName == "borrar")  
    {  
  
        RequerInfo = new REQUERIMIENTO();  
  
        RequerInfo =  
  
        RequerimientoLogica.ObtenerRequerimientoXIdInfo(COD_REQ);  
  
        RequerimientoLogica.Delete(RequerInfo);  
  
        CargarFormulario();  
  
    }  
}  
  
catch (Exception)  
  
    {  
    }  
}  
}
```



```
// protected void grd_Requerimiento_SelectedIndexChanged(object sender,
EventArgs e)

// {

// }

}
```

Requerimiento Nuevo

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using AccesoDatos;

using LogicaNegocio;

using System.Data.Linq;

namespace SUMINISTROS.Administador

{

    public partial class RequerimientosNuevo : System.Web.UI.Page

    {

        private REQUERIMIENTO RequerimientosInfo;

        REQUERIMIENTO Userinfo = new REQUERIMIENTO();

        List<EMPLEADOS> EmpleadoLista = new List<EMPLEADOS>();

    }

}
```



```
List<TIPO_SUMINISTROS> TipoSuministroLista = new  
List<TIPO_SUMINISTROS>();  
  
List<SUMINISTROS_COMPRAS> SuministrosLista = new  
List<SUMINISTROS_COMPRAS>();  
  
REQUERIMIENTO RequerInfo = new REQUERIMIENTO();  
  
private int COD_REQ;  
  
protected void Page_Load(object sender, EventArgs e)  
{  
    if (!IsPostBack)  
    {  
        txt_Fecha.Text = DateTime.Now.ToShortDateString();  
        txt_Fecha.Enabled = false;  
        ddl_Estado.SelectedIndex = 0;  
        ddl_Estado.Enabled = false;  
        //CargarUsuario();  
        // CargarTipoSuministro();  
        int codigo = 0;  
        CargarTipoSuministro();  
        if (Request["COD_REQ"] != null)  
        {  
            codigo = Convert.ToInt32(Request["COD_REQ"]);  
            // txtcodigo.Text = COD_REQ.ToString();  
            if (codigo > 0)
```



```
{  
    // txtUsuario.Visible = true;  
    txt_producto.Visible = true;  
    //txtTipoSuministros.Visible = true;  
    // CargarUsuario();  
  
    // CargarTipoSuministro();  
    CargarForm(codigo);  
}  
}  
}  
}  
  
private void CargarForm(int id)  
{  
    REQUERIMIENTO RequerInfo = new REQUERIMIENTO();  
    RequerInfo = RequerimientoLogica.ObtenerRequerimientoXIdInfo(id);  
    if (RequerInfo.EST_REQ == null)  
    {  
    }  
  
    // txtcodigo.Text = RequerInfo.COD_REQ;  
  
    //ddl_Usuario.SelectedValue =  
    RequerInfo.EMPLEADOS.COD_EMP.ToString();  
  
    ddl_TipoSuministro.SelectedValue = RequerInfo.COD_TIPSUM.ToString();  
}
```



```
// txtfecha.Text = RequerInfo.FEC_REQ.Equals();

txtObservacion.Text = RequerInfo.OBS_REQ;

ddl_Estado.SelectedValue = RequerInfo.EST_REQ.ToString();

}

//private void CargarUsuario()

//{

//  EmpleadoLista = new List<EMPLEADOS>();

//  EmpleadoLista = EmpleadosLogica.ObtenerEmpleadosLista();

//  EmpleadoLista.Insert(0, new EMPLEADOS() { NOM_EMP = "Seleccione
Usuario" });

//  ddl_Usuario.DataSource = EmpleadoLista;

//  ddl_Usuario.DataTextField = "NOM_EMP";

//  ddl_Usuario.DataValueField = "COD_EMP";

//  ddl_Usuario.DataBind();

//}

private void CargarTipoSuministro()

{

TipoSuministroLista = new List<TIPO_SUMINISTROS>();

TipoSuministroLista = TipoSuministros_Logica.ObtenerSuministros();

TipoSuministroLista.Insert(0, new TIPO_SUMINISTROS() { NOM_TIPSUM =
"Seleccione Tipo de Suministr" });

ddl_TipoSuministro.DataSource = TipoSuministroLista;

ddl_TipoSuministro.DataTextField = "NOM_TIPSUM";
```



```
ddl_TipoSuministro.DataValueField = "COD_TIPSUM";

ddl_TipoSuministro.DataBind();

}

private void CargarSuministros(int tipo)
{
    SuministrosLista = new List<SUMINISTROS_COMPRAS>();

    SuministrosLista = Logica_Suministro.ObtenerListaSuministrosXTipo(tipo);

    SuministrosLista.Insert(0, new SUMINISTROS_COMPRAS() { NOM_SUM =
"Seleccione Producto" });

    ddl_producto.DataSource = SuministrosLista;

    ddl_producto.DataTextField = "NOM_SUM";

    ddl_producto.DataValueField = "COD_SUM";

    ddl_producto.DataBind();

}

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

protected void imgFecha_Click(object sender, ImageClickEventArgs e)
{
}

protected void ddl_TipoSuministro_SelectedIndexChanged(object sender,
EventArgs e) {

    try

    {

        int codtiposum = Convert.ToInt32(ddl_TipoSuministro.SelectedValue);
```



```
CargarSuministros(codtiposum);

}

catch (Exception)

{

}

}

protected void imgCalculo_Click(object sender, ImageClickEventArgs e)

{

    try

    {

        if (txtCantidad.Text != "")

        {

            decimal cant = Convert.ToDecimal(txtCantidad.Text);

            decimal precio = Convert.ToDecimal(hdf_precio.Value);

            decimal tt = (precio * cant);

            txt_producto.Text = tt.ToString();

        }

    }

    catch (Exception)

    {

    }

}

protected void ddl_producto_SelectedIndexChanged(object sender, EventArgs e)
```




```
{  
  
    try  
  
    {  
  
        int codigo = Convert.ToInt32(ddl_producto.Selected.Value);  
  
        SUMINISTROS_COMPRAS suministros = new  
SUMINISTROS_COMPRAS();  
  
        suministros = Logica_Suministro.ObtenerSuministrosInfoXCod(codigo);  
  
        if (suministros != null || suministros.COD_SUM > 0)  
  
        {  
  
            //txtiva.Text = suministros.IVA.ToString();  
  
            hdf_precio.Value = suministros.PRECIO.ToString();  
  
            txtCantidad.Text = "";  
  
            txt_producto.Text = "";  
  
        }  
  
    }  
  
    catch (Exception)  
  
    {  
  
    }  
  
}  
  
List<REQUERIMIENTO> ListaRequerimientos = new  
List<REQUERIMIENTO>();  
  
private void CargarLista(List<PresubuestoInfo> Lista)  
  
{
```



```
if (Lista != null)
{
    dgrLista.DataSource = Lista;
    dgrLista.DataBind();
}
}

private List<PRESUPUESTO> RequerimientosListawp
{
    get
    {
        if (Session["Requerimientos"] == null)
        {
            Session["Requerimientos"] = new List<PRESUPUESTO>();
        }

        return Session["Requerimientos"] as List<PRESUPUESTO>;
    }

    set
    {
        ViewState["Requerimientos"] = value;
    }
}

private List<PresubuestoInfo> PresubuestoInfoLiSTA
```



```
{  
  
    get  
  
    {  
  
        if (ViewState["__PresubuestoInfo"] == null)  
  
        {  
  
            ViewState["__PresubuestoInfo"] = new List<PresubuestoInfo>();  
  
        }  
  
        return ViewState["__PresubuestoInfo"] as List<PresubuestoInfo>;  
  
    }  
  
    set  
  
    {  
  
        ViewState["__PresubuestoInfo"] = value;  
  
    }  
  
}  
  
SUMINISTROS_COMPRAS smc = new SUMINISTROS_COMPRAS();  
  
private void Guardar()  
  
{  
  
    try  
  
    {  
  
        REQUERIMIENTO RequerimientoInfo = new REQUERIMIENTO();  
  
        //RequerimientoInfo.CAN_REQ = Convert.ToDecimal(txtCantidad.Text);  
  
    }  
  
}
```



```
RequerimientoInfo.COD_EMP = hdf_IdEmpleado.Value == "" ? 0:
Convert.ToInt32(hdf_IdEmpleado.Value);

RequerimientoInfo.COD_TIPSUM =
Convert.ToInt32(ddl_TipoSuministro.SelectedValue);

RequerimientoInfo.EST_REQ = ddl_Estado.SelectedValue;

RequerimientoInfo.FEC_REQ = Convert.ToDateTime(txt_Fecha.Text);

RequerimientoInfo.OBS_REQ = txtObservacion.Text;

Valida(RequerimientoInfo);

//hdf_codrequerimiento.Value =

RequerimientoLogica.save(RequerimientoInfo).ToString();

ProcesAddList( hdf_codrequerimiento.Value == "" ? 0 :
Convert.ToInt32(hdf_codrequerimiento.Value));

}

catch (Exception ex)

{

    lbl_Mensaje.Text = ex.Message;

    lbl_Mensaje.Visible = true;

}

}

private void Valida(REQUERIMIENTO RequerimientoInfo)

{

    string sms = "";
```



```
if (RequerimientoInfo.COD_EMP == 0)
{
    sms += "Ingresar Empleado</br>";
}

if (RequerimientoInfo.COD_TIPSUM == 0)
{
    sms += "Ingresar Tipo de Producto</br>";
}

if (sms != "")
{
    throw new ArgumentException(sms);
}
}

private void GuardarDetalle(int IdRequerimiento)
{
    try
    {
        if (PresubuestoInfo.Count > 0)
        {
            foreach (PresubuestoInfo item in PresubuestoInfo)
            {
                PRESUPUESTO Presupuestoinfo = new PRESUPUESTO();
                Presupuestoinfo.CAN_PRE = item.CANT_PRE;
                Presupuestoinfo.COD_EMP = Convert.ToInt32(hdf_IdEmpleado.Value);
                Presupuestoinfo.COD_REQ = IdRequerimiento;
                Presupuestoinfo.VALPRE = Convert.ToDecimal(txt_producto.Text);
            }
        }
    }
}
```



```
        Presupuestoinfo.COD_SUM = item.COD_SUM;
        PresupuestoLogica.Save(Presupuestoinfo);
    }
}
lbl_Mensaje.Text = "Datos Guardados";
lbl_Mensaje.Visible = true;

}
catch (Exception ex)
{
    lbl_Mensaje.Text = ex.Message;
    lbl_Mensaje.Visible = true;
}
}
private void GuardarBase()
{
    try
    {
        REQUERIMIENTO RequerimientoInfo = new REQUERIMIENTO();
        //RequerimientoInfo.CAN_REQ = Convert.ToDecimal(txtCantidad.Text);
        RequerimientoInfo.COD_EMP = Convert.ToInt32(hdf_IdEmpleado.Value);
        //RequerimientoInfo.COD_SUM =
        Convert.ToInt32(ddl_producto.SelectedValue);
        RequerimientoInfo.COD_TIPSUM =
        Convert.ToInt32(ddl_TipoSuministro.SelectedValue);
        RequerimientoInfo.EST_REQ = ddl_Estado.SelectedValue;
        RequerimientoInfo.FEC_REQ = Convert.ToDateTime(txt_Fecha.Text);
        RequerimientoInfo.OBS_REQ = txtObservacion.Text;
        hdf_codrequerimiento.Value =
        RequerimientoLogica.save(RequerimientoInfo).ToString();
        GuardarDetalle(Convert.ToInt32(hdf_codrequerimiento.Value));
    }
    catch (Exception ex)
    {
        lbl_Mensaje.Text = ex.Message;
        lbl_Mensaje.Visible = true;
    }
}

private void ProcesAddLIst(int idreq)
{
    try
    {
```



```
PresubuestoInfo PresupuestoInfo = new PresubuestoInfo();
PresupuestoInfo.COD_EMP = Convert.ToInt32(hdf_IdEmpleado.Value);
PresupuestoInfo.ID_PRE = 0;
PresupuestoInfo.COD_REQ =
idreq;//Convert.ToInt32(hdf_codrequerimiento.Value);
PresupuestoInfo.VALPRE = txt_producto.Text == "" ? 0 :
Convert.ToDecimal(txt_producto.Text);
SUMINISTROS_COMPRAS suministros = new
SUMINISTROS_COMPRAS();
suministros =
Logica_Suministro.ObtenerSuministrosInfoXCod(Convert.ToInt32(ddl_producto.Select
edValue));
PresupuestoInfo.COD_SUM = suministros.COD_SUM;
PresupuestoInfo.TIOPRO =
suministros.TIPO_SUMINISTROS.NOM_TIPSUM;
PresupuestoInfo.PRO = suministros.NOM_SUM;
PresupuestoInfo.FECHA = txt_Fecha.Text;
PresupuestoInfo.CANT_PRE = txtCantidad.Text == "" ? 0 :
Convert.ToDecimal(txtCantidad.Text);
ValidarDetalle(PresupuestoInfo);
PresubuestoInfoLiSTA.Add(PresupuestoInfo);
CargarLista(PresubuestoInfoLiSTA);
decimal total = 0;
foreach (PresubuestoInfo item in PresubuestoInfoLiSTA)
{
    total += item.VALPRE;
}
lbl_Presupuesto.Text = total.ToString();
//validar Empleado
EMPLEADOS empleadoinfo = new EMPLEADOS();
empleadoinfo =
EmpleadosLogica.ObtenerEmpleadosXCodInfo(Convert.ToInt32(hdf_IdEmpleado.Valu
e));
id_Preasig.Text = empleadoinfo.PRES_EMP.ToString();
if (empleadoinfo.PRES_EMP > total)
{
    lbl_Presupuesto.Text = total.ToString();
}
else
{
    lbl_Mensaje.Text = "El presupuesto Asignado sobrepasa!!!";
    lbl_Mensaje.Visible = true;
}
}
```



```
        catch (Exception ex)
        {
            lbl_Mensaje.Text = ex.Message;
            lbl_Mensaje.Visible = true;
        }
    }
    private void ValidarDetalle(PresubuestoInfo PresupuestoInfo)
    {
        string sms = "";
        if (PresupuestoInfo.CANT_PRE == 0)
        {
            sms += "Ingresar la Cantidad del Producto<br>";
        }
        if (PresupuestoInfo.COD_SUM == 0)
        {
            sms += "Ingresar el Producto<br>";
        }
        if (txt_producto.Text == "" )
        {
            sms += "No se ha generado el total<br>";
        }
        if (sms != "")
        {
            throw new ArgumentException(sms);
        }
    }
    protected void guardar_Click(object sender, ImageClickEventArgs e)
    { GuardarBase(); }
    protected void lnkgrabar_Click(object sender, EventArgs e)
    { GuardarBase(); }
    protected void btnAdd_Click(object sender, EventArgs e)
    {
        Guardar();
    }
    protected void txtiva_TextChanged(object sender, EventArgs e)
    {}
    protected void dgrLista_ItemCommand(object source,
    DataGridCommandEventArgs e)
    {
        try
        {
            int index = e.Item.ItemIndex;
            dgrLista.SelectedIndex = index;
        }
        catch (Exception)
        {
            throw;
        }
    }
}
```




Anexo 3: Manual Usuario

A.03.01 INTRODUCCION

El **SISTEMA DE CONTROL DE SUMINISTROS DE COMPRAS** es una aplicación que permite facilitar el ingreso de requerimientos para cada área y tener un mejor control de los pedidos.

El sistema permite ingresar datos de Empleados, ingreso de requerimientos y de productos según la ventana que necesite además nos permitirá generar dos tipos de reportes por cedula de empleado o por fecha.

A.03.02 REQUISITOS

La aplicación es independiente de la plataforma, siendo compatible con los sistemas operativos: Windows XP, vista, 7. Para su uso no es necesario la instalación de otras herramientas de software.

A.03.03 ACCESO A LA APLICACIÓN

Al ejecutar el sistema, automáticamente se visualizará la ventana de la Figura 22, esta es la pantalla de autenticación y bienvenida del Sistema permite que solo puedan acceder los usuarios establecidos. Si se es usuario del sistema, se deberá registrar en los campos mostrados en pantalla (usuario y clave), no es necesario buscar en el ingreso al sistema si es usuario o Administrador ya que solo editando su información el sistema genera las ventanas tanto de administrador o usuario, de esta forma se ingresará en el sistema, tanto el destino como los privilegios del rol que ocupe dentro de la aplicación.

Figura 22. Pantalla autenticación del usuario



A.03.04 ROLES.

El rol es la clasificación que tienen los usuarios dentro del sistema, de manera que así se controla el trabajo de cada usuario dando determinados privilegios a los mismos según el rol.

- **Administrador:** Posee todos los privilegios dentro del sistema, generalmente son sus creadores y/o personas capaces de manejar toda su información y en caso de que fuera necesario la modificación del mismo.
- **Usuario:** son usuarios encargados de ingresar el requerimiento. Generalmente este rol lo ocupa el personal de la Oficina Central.

A.03.05 MENU PRINCIPALE

En esta barra constan algunas opciones para que el usuario pueda utilizarlos

Servicio, y suministros.



A.03.06 REQUERIMIENTOS LISTA

El usuario debe ingresar en esta opción para poder generar el requerimiento de compras, una vez generado el requerimiento puedo observar la lista de requerimientos

[Nuevo](#)

# Req	Nombre	Apellido	Estado
1	JUAN	FLORES	A
2	JUAN	FLORES	A
3	JUAN	FLORES	A

A.03.07 REQUERIMIENTOS NUEVO

Una vez ingresando en el icono o la palabra NUEVO, se despliega una ventana debe ingresar cada opción



 [Guardar](#)  [Regresar](#)

 Seleccione Empleado

Nombre Apellido

Estado:


Tipo Suministro:


Observacion: Fecha: 

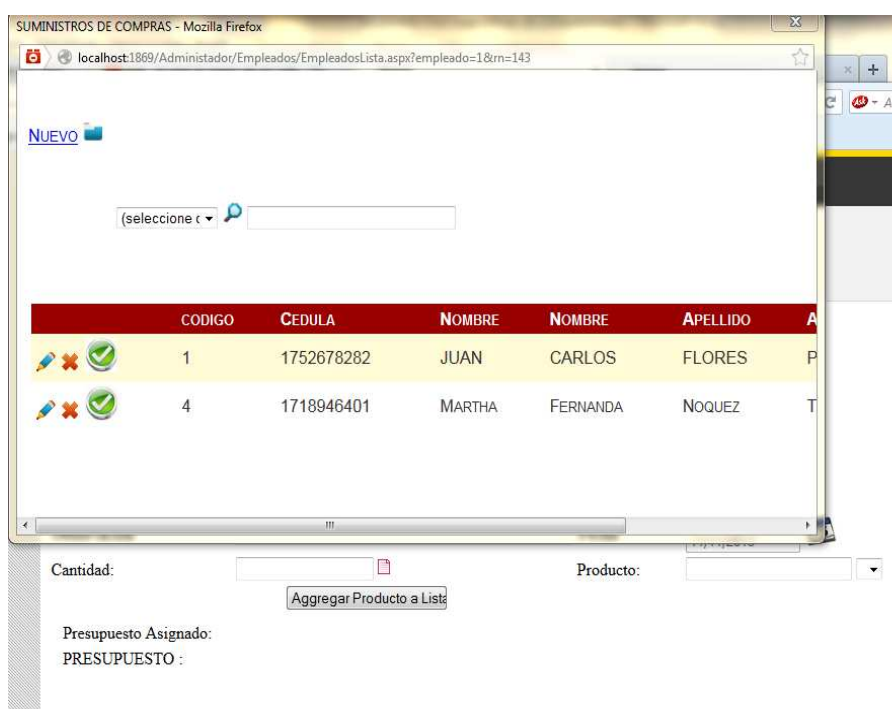
Cantidad:  Producto:



Presupuesto Asignado:
PRESUPUESTO :


A.03.08 SELECCIÓN DE EMPLEADO

Podemos observar donde está el icono  seleccione empleado se despliega una página adicional donde el usuario elige su nombre, seleccionamos Tipo suministro y los

productos se generan de inmediato, debemos colocar observación y cantidad
seleccionamos este icono  se calcula el valor directo y agregamos los productos.



	CODIGO	CEDULA	NOMBRE	NOMBRE	APELLIDO	A
	1	1752678282	JUAN	CARLOS	FLORES	P
	4	1718946401	MARTHA	FERNANDA	NOQUEZ	T

Cantidad: 

Producto:

Presupuesto Asignado:
PRESUPUESTO :

A.03.09 GUARDAR REQUERIMIENTO

Al momento que ingresa los datos debe guardar el requerimiento, donde podemos verificar el presupuesto que consta cada usuario y un detalle.



Presupuesto por cada empleado

Datos Guardados
 Guardar Regresar

Seleccione Empleado Nombre: JUAN Apellido: FLORES

Estado: Ingresado

Tipo Suministro: Suministros de Oficina

Observación: Perfecto Fecha: 03/11/2013

Cantidad: 1 Producto: 150,00 Cartucho A150

Agregar Producto a Lista

Tipo Pro	Producto	Fecha	Valor
F	Suministros de Oficina	Lapiz	03/11/2013 9,00
F	Suministros de Oficina	Cartucho A150	03/11/2013 150,00

Presupuesto Asignado: 300,00
PRESUPUESTO 159,00

A.03.10 DETALE GENERADO

Este detalle es verificado por cada usuario para evitar caer nuevamente en un mal manejo de consumo de suministros

Tipo Pro	Producto	Fecha	Valor
✗ F Suministros de Oficina	Lapiz	11/11/2013	2,25
✗ F Suministros de Oficina	Cartucho A150	11/11/2013	300,00

Presupuesto Asignado:
PRESUPUESTO : 302,25

A.03.11 GUARDAR SIN NINGUN DATO



La cadena de entrada no tiene el formato correcto.



Guardar



Regresar

A.03.12 GENERAR EL REQUERIMIENTO

Al momento de generar el requerimiento sobrepasa el presupuesto asignado no se guardara y tendrá que quitar elementos asignados.

El sistema le informa que el monto sobrepasa el presupuesto que asignan a cada área.



Al lado izquierdo tenemos esta imagen — se utiliza para eliminar lo que se agrego de productos.



El presupuesto Asignado sobrepasa!!



[Guardar](#)



[Regresar](#)

Seleccione Empleado



Nombre

Apellido

Estado:

Tipo Suministro:

Observacion:

Fecha

Cantidad:



Procedimiento

[Agregar Producto a Lista](#)

Tipo Pro	Producto	Fecha	Valor
✗ F Suministros de Oficina	Lapiz	11/11/2013	2,25
✗ F Suministros de Oficina	Cartucho A150	11/11/2013	300,00

Presupuesto

Asignado:

PRESUPUESTO : **302,25**

A.03.13 REQUERIMIENTO INGRESADO EXITOSAMENTE

Una vez ingresado el requerimiento podremos verificarlos en la lista de requerimientos donde tendremos nuestro nombre el estado y el número de requerimiento.

[Nuevo](#)

# Req	Nombre	Apellido	Estado
1	JUAN	FLORES	A
2	JUAN	FLORES	A
3	JUAN	FLORES	A



Bibliografía

House, Ernie R. (1992). Gran política, pequeña política. *Cuadernos de Pedagogía*, 202, 51-57.

Druker, Peter F.(1980). Dirección dinámica de Empresas. New Jersey: International Business Institute

Ballou, R (2004) Logística: Administración de la cadena de suministros Pearson Educación.

<http://books.google.com.mx/books?id=ii5xqLQ5VLgC&printsec=frontcover&dq=>