



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES

ARAGÓN

“DISEÑO E IMPLEMENTACIÓN DEL SISTEMA PARA EL
CONTROL DE PRODUCTO NO CONFORME”

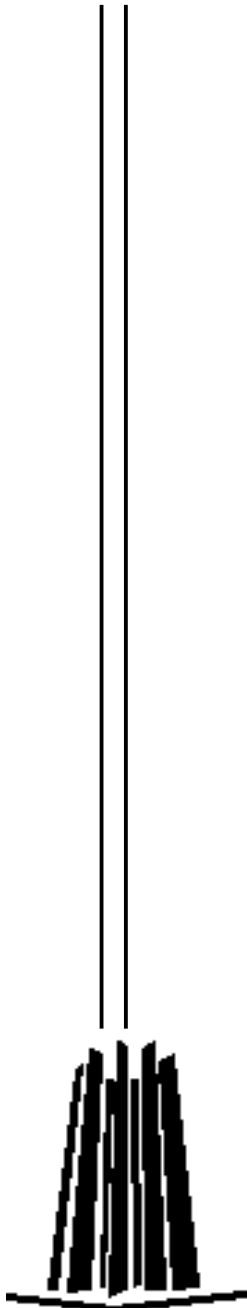
DESARROLLO DE UN CASO PRÁCTICO

PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A :

RAÚL ACOSTA DÍAZ



Junio 2011



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ÍNDICE

ÍNDICE DE FIGURAS	1
INTRODUCCIÓN	5
CAPÍTULO I	
1. Análisis del proceso de producto no conforme	6
1.1. Objetivo	6
1.2. Alcance	6
1.3. Antecedentes	6
1.4. Definición del problema	7
1.5. Diagrama de flujo	8
1.6. Descripción de los módulos	10
CAPÍTULO II	
2. Diseño del sistema	15
2.1. Modelado del sistema	15
2.2. Lenguaje Unificado de Modelado (UML)	17
2.3. El Proceso Racional Unificado (RUP)	18
2.4. Descripción de las actividades	21
2.4.1. Fase de inicio	21
2.4.2. Fase de elaboración	21
2.4.2.1. Identificar actores	22
2.4.2.2. Diagramas de paquetes	22
2.4.2.3. Identificar casos de uso	24
2.4.2.4. Desarrollo de los casos de uso	34
2.4.2.5. Diagramas de actividades	48
2.4.3. Fase de construcción	70
2.5. Diseño de base de datos	72
2.5.1. Base de datos relacionales	72
2.5.2. Normalización y diseño de la base de datos	73
2.5.3. Diagrama físico de base de datos del sistema	74
CAPÍTULO III	
3. Desarrollo de la aplicación	79
3.1. Lenguaje de programación	79
3.1.1. Microsoft .NET	79
3.1.2. .NET Framework	80

3.1.3. Ajax Control Tool Kit	80
3.1.4. Itext Sharp	81
3.1.5. JQuery	81
3.1.6. Web Service	82
3.1.7. HTML estático	83
3.2. Interfaz gráfica del sistema	83
3.3. Herramientas de desarrollo	89
3.3.1. Visual Studio	89
3.3.2. Aplicaciones Web en plataforma .NET	89
3.3.3. Desarrollo diagramas UML con Visio	91
3.3.4. Modelado base de datos con Erwin	92
3.4. Servidor Web	93
3.4.1. IIS	93
3.4.2. SQL Server	94
CAPÍTULO IV	
4. Implantación del sistema	97
4.1. Descripción del proceso de implementación	97
4.1.1. Pruebas finales	100
4.1.2. Migración de base de datos a productivo	102
4.1.3. Instalación del módulo WEB	105
4.1.4. Configuración de la página	107
4.1.5. Configuración DTS para el envío de correo	112
4.1.6. Liberación del sistema a productivo con los usuarios	115
CONCLUSIONES	117
GLOSARIO	118
REFERENCIAS BIBLIOGRÁFICAS	121
BIBLIOGRAFIA	122
MESOGRAFIA	123
ANEXO I	125

ÍNDICE DE FIGURAS

1.1. Proceso producto no conforme	9
2.1. Fases e Iteraciones de la Metodología RUP	20
2.2. Actores del sistema	22
2.3. Caso de uso para la operación nuevo	24
2.4. Caso de uso para la operación editar	25
2.5. Caso de uso para la operación usuarios	25
2.6. Caso de uso para la operación almacén	26
2.7. Caso de uso para la operación material sirve	26
2.8. Caso de uso para la operación aplica rechazo	27
2.9. Caso de uso para la operación sorteó o retrabajó	28
2.10. Caso de uso para la operación regresa origen	29
2.11. Caso de uso para la operación buscar	30
2.12. Caso de uso para la operación informe	31
2.13. Caso de uso para la operación panel	32
2.14. Caso de uso para la operación PO	32
2.15. Caso de uso para la operación comprador	33
2.16. Caso de uso para la operación consulta proveedor	33
2.17. Descripción del caso de uso crear un nuevo rechazo	35
2.18. Descripción del caso de uso agregar número de PO	36
2.19. Descripción del caso de uso bloquear material	37
2.20. Descripción del caso de uso consultar si el proveedor está de acuerdo con el rechazo	38
2.21. Descripción del caso de uso indicar si se resorteó o retrabajó	39
2.22. Descripción del caso de uso agregar hoja de aceptado	40
2.23. Descripción del caso de uso desbloquear material	41
2.24. Descripción del caso de uso agregar folio Good Receipt	42
2.25. Descripción del caso de uso agregar folio Invoice Receipt	43
2.26. Descripción del caso de uso proveer NC y/o SCRAP	44
2.27. Descripción del caso de uso agregar número de pedimento y documentos	45
2.28. Descripción del caso de uso validar si aplica el rechazo	46
2.29. Descripción del caso de uso validar si el material sirve	47
2.30. Diagrama que representa el flujo general de la pantalla PO	48
2.31. Diagrama que representa el flujo que se sigue para realizar el proceso de login	49
2.32. Diagrama que representa el flujo que se sigue para asignar en el menú lo que visualizara el usuario de almacén	50

2.33. Diagrama que representa el flujo que se sigue para asignar en el menú lo que visualizara el usuario de calidad	51
2.34. Diagrama que representa el flujo que se sigue para asignar en el menú lo que visualizara el usuario de finanzas	52
2.35. Diagrama que representa el flujo que se sigue para asignar en el menú lo que visualizara el usuario de compras	53
2.36. Diagrama que representa el flujo que se sigue para asignar en el menú lo que visualizara el usuario de ingeniería	54
2.37. Diagrama que representa el flujo que se sigue para asignar en el menú lo que visualizara el usuario de IMPO/EXPO	55
2.38. Diagrama que representa el flujo que se sigue para asignar en el menú lo que visualizara el usuario de almacén	56
2.39. Diagrama que representa el flujo general de la pantalla de usuarios	57
2.40. Diagrama que representa el flujo general de la pantalla de panel	57
2.41. Diagrama que representa el flujo general de la pantalla editar	58
2.42. Diagrama que representa el flujo general de la pantalla del informe	59
2.43. Diagrama que representa el flujo general de la pantalla de buscar	60
2.44. Diagrama que representa el flujo general de la pantalla material sirve	61
2.45. Diagrama que representa el flujo general de la pantalla regresa a origen	62
2.46. Diagrama que representa el flujo general de la pantalla aplica rechazo	63
2.47. Diagrama que representa el flujo general de la pantalla sorteo o retrabajó	64
2.48. Diagrama que representa el flujo general de varios procesos	65
2.49. Diagrama que representa el flujo general de la pantalla del comprador	66
2.50. Diagrama que representa el flujo general de la pantalla consulta proveedor	66
2.51. Diagrama que representa el flujo general de la pantalla nuevo rechazo	67
2.52. Diagrama que representa el flujo general de la pantalla login	68
2.53. Diagrama que representa el flujo general de la pantalla almacén	69
2.54. Diagrama de entidades de la base de datos	76
2.55. Diagrama físico de la base de datos	77
3.1 HTML estático: Configuración típica de una "aplicación web" que se limita a ofrecer la información almacenada en páginas HTML a las que el usuario final accede desde su navegador web	83
3.2 Login de usuario	84
3.3 Captura y edición de usuarios nuevos/existentes	84
3.4 Capturar de nuevo folio para el producto no conforme	85
3.5 Captura del número de PO	85
3.6 Bloqueo de material	85

3.7 Consulta proveedor	85
3.8 Indicar disposición	86
3.9 Proveer NC y/o SCRAP	86
3.10 Almacén captura folio	86
3.11 Cuentas por pagar captura folio	86
3.12 Disposición IMPO-EXPO	86
3.13 Agregar peso y tipo de material	87
3.14 IMPO-EXPO	87
3.15 Se tramita SCRAP	87
3.16 Almacén valida si se tramita el regreso del material	87
3.17 Piezas buenas y malas	88
3.18 Desbloqueo de almacén	88
3.19 Documento autorización de materiales descontar	88
3.20 Validación calidad/ingeniería	88
3.21. La plataforma .NET: Los formularios ASP.NET y los formularios Windows son las dos alternativas principales de las que dispone el programador para crear las interfaces de usuario de sus aplicaciones	90
3.22. Eventos en ASP.NET: La respuesta de la aplicación web se obtiene como resultado de ejecutar los manejadores de eventos asociados a los controles incluidos en la interfaz de usuario	90
3.23. ASP.NET se encarga de mostrar los formularios web de la forma que resulte más adecuada para el navegador que utilice el usuario final de la aplicación	91
3.24. Arquitectura cliente/servidor	95
4.1. Aplicación Web de ASP.NET	97
4.2. Creación de una aplicación Web de ASP.NET	98
4.3. Creación de un directorio virtual en IIS	99
4.4. Asistente para importación/exportación	102
4.5. Selector de tablas y vistas	103
4.6. Transformación de columnas	104
4.7. Inicio ejecución del proceso	104
4.8. Estructura sitio web	105
4.9. Cliente FTP	106
4.10. Archivos contenidos en un sitio web	107
4.11. Internet Information Services en Windows 2000	107
4.12. El cuadro de diálogo propiedades en Internet Information Services para la carpeta rechazos, que no es un directorio virtual	108
4.13. El cuadro de diálogo propiedades en Internet Information Services para la carpeta	

rechazos después de haberla convertido en un directorio virtual	109
4.14. La pestaña documentos del cuadro de diálogo propiedades	109
4.15. La pestaña seguridad de directorios del cuadro de diálogo propiedades	110
4.16. Cuadro de diálogo de métodos de autenticación	110
4.17. Cuadro de diálogo de Configuración de aplicación	111
4.18. Ejecutar DTS	112
4.19. Configuración nombre DTS	113
4.20. Configuración del proceso a ejecutar	113
4.21. Termino de configuración proceso	114
4.22. Configuración en la programación del DTS	115

INTRODUCCIÓN

Un producto es un objeto que puede ser comprado o vendido, y que responde a una necesidad propia de un grupo de consumidores.

Durante la creación o ensamble de un producto, existe un área de calidad la cual se encarga de controlar la calidad de los productos que se elaboran para garantizar la satisfacción de sus clientes. Por tal motivo dicha área es la encargada de establecer procesos de inspección que aseguren un producto de buena calidad. Estos procesos identifican defectos como: dimensiones incorrectas, diferencias en color, daños visibles, mala aplicación de esmalte y/o grietas en las piezas, por mencionar algunos.

Los procesos de análisis e inspección son aplicados tanto en el almacén como en el ensamble, con el fin de identificar el producto no conforme resultado de un proceso que no cumplió con los requisitos especificados.

Por tanto las áreas deben asegurarse de que el producto que no sea conforme con los requisitos, se identifique y controle para prevenir su uso o entrega no intencional. Los controles, las responsabilidades y autoridades relacionadas con el tratamiento del producto no conforme estarán definidos dependiendo del problema.

El área correspondiente debe tratar los productos no conformes mediante una o más de las siguientes maneras:

- a) Tomando acciones para eliminar la no conformidad detectada.
- b) Autorizando su uso, liberación o aceptación.
- c) Tomando acciones para impedir su uso o aplicación originalmente previsto.

Actualmente el proceso para identificar y solucionar el producto no conforme tiende a provocar errores, pérdida, retrasos y obtención incorrecta de la información, debido a que la mayor parte de esta es manejada en papel y tiene que ser distribuida a distintas áreas para dar una solución efectiva. Por tales motivos se llega a tener un descontento por parte de los distribuidores por que muchas veces por estos inconvenientes su material es retenido, provocando perdidas tanto para ellos como para la empresa ya que los tiempos de entrega del producto final se ven detenidos por la ineficiencia de su proceso.

Por los anteriores motivos es importante crear un sistema integral que cumpla con las especificaciones necesarias para el mejoramiento del proceso de producto no conforme, el cual permita:

- Agilizar y optimizar los tiempos de respuesta, captura y distribución a las distintas área.

- Proporcionar de manera eficiente información en lo que compete al proceso y seguimiento que se le da a cada rechazo.
- Mantener un histórico de los distribuidores con el fin de mejorar el producto y evitar la continuidad de defectos comunes.
- Evitar pérdida de información y el retrabajó que conlleva iniciar el proceso desde el comienzo o la incorrecta liberación del producto.

El presente trabajo tiene como objetivo desarrollar e implementar un sistema que cubra las necesidades actuales necesarias para el mejoramiento del proceso de producto no conforme.

Con las herramientas y la metodología a utilizar se pretende generar una base de datos integral que cubra con las necesidades del sistema así como la creación de módulos independientes que realicen las funciones necesarias para procesar la información generada de un rechazo, permitiendo tener acceso sólo a los datos que le competen a cada área o comprador, generación de un módulo de recordatorio, el cual permitirá mantener a los usuarios actualizados sobre sus actividades pendientes en el sistema.

Con la finalidad de automatizar y mejorar el proceso se documenta cada parte del requerimiento en cuatro capítulos los cuales desglosan de manera consecutiva los procesos llevados a cabo y la información necesaria en cada modulo, permitiendo el manejo óptimo de la información y eliminando toda redundancia que pudiera afectar al sistema.

CAPÍTULO 1: Explica en detalle el procedimiento que se lleva a cabo para el producto no conforme, indicando así las áreas involucradas, personal y documentos que se requieren para dar un seguimiento de manera efectiva, permitiendo dar un introducción sobre conceptos y el flujo de la información.

CAPÍTULO 2: Se analiza la estructura del sistema, permitiendo así plantear y diseñar los formularios que formaran parte del sistema, se plasmarán las interacciones entre el usuario y el sistema mediante los diagramas UML¹. También se diseñara la base de datos de acuerdo a los formularios existentes y a las necesidades que se requieran al momento de extraer la información.

CAPÍTULO 3: Se realizará como tal el diseño de la aplicación y la estructura que tendrá la base de datos. Se dará una introducción al gestor de base de datos y el lenguaje de programación utilizado, así como sus herramientas existentes a este lenguaje las cuales nos permitirán tener un mejor manejo de la información y comodidad para el usuario.

¹ UML (*Unified Modeling Language*, Lenguaje Unificado de Modelado)

CAPÍTULO 4: Recopilación de los resultados obtenidos en el punto 3, lo que permitirá llevar a cabo las pruebas pertinentes para poder migrar tanto la base de datos como el sistema, permitiendo llevarlo a producción para su utilización por parte del usuario.

CAPÍTULO I

ANÁLISIS DEL PROCESO DE PRODUCTO NO CONFORME

1. Análisis del proceso de producto no conforme

1.1. Objetivo

Establecer los lineamientos que deben ser aplicados para la identificación y tratamiento de los servicios no conformes aplicables a los procesos definidos en el alcance del sistema de gestión de la calidad, así como establecer la secuencia de actividades para asegurar que los productos que no sean conformes con los requisitos especificados, se identifiquen y controlen para prevenir el uso del material.

1.2. Alcance

Aplicable para el sistema de gestión de la calidad, y los procesos involucrados en el seguimiento de producto no conforme, así como para los procesos gobernadores y de soporte relacionados en punto de contacto, para el personal perteneciente a la organización.

1.3. Antecedentes

Se define como materia prima todos los elementos que se incluyen en la elaboración de un producto o servicio. La materia prima es todo aquel elemento que se transforma e incorpora en un producto final. Un producto terminado tiene incluido una serie de elementos y subproductos, que mediante un proceso de transformación permitieron la confección del producto final.

Las materias primas críticas son inspeccionadas para garantizar que cumplen las especificaciones de compra. Los controles se realizarán de acuerdo con las instrucciones de inspección establecidas, y de su resultado se emitirá el correspondiente informe de verificación de materia prima.

El director de manufactura establecerá las instrucciones de inspección necesarias para la verificación de las materias primas críticas, estas podrán ser tanto en almacén como en ensamble. El resultado de la inspección se reflejará en el informe de verificación de materia prima.

En caso de que se detecte algún producto no conforme, el director de manufactura complementará el informe de no conformidad y se realizara el procedimiento correspondiente.

El director de calidad, a la vista de los recibos de factura y los rechazos, mantendrá un histórico del suministro de cada proveedor. Cuando a juicio del director de calidad los resultados de la supervisión no sean aceptables, se dará de baja al proveedor de la lista, con comunicación al director compras y al director de manufactura.

La materia prima no conforme y el producto terminado comprado para la venta no conforme se identifican como un rechazo y/o producto no conforme.

Si se trata de producto no conforme el supervisor de la calidad junto con el jefe de aseguramiento de la calidad y el comprador analizan la no conformidad para estudiar la clase de disposición que se le dará al mismo, ya sea reproceso, rechazo, reparación usando como esta o reclasificado como.

1.4. Definición del problema

Actualmente el proceso de rechazo conlleva una serie de pasos los cuales tienen como objetivo clasificar el producto y darle un seguimiento pero al ser manual se presentan los siguientes problemas:

- ✓ No se tiene un seguimiento de los rechazos.
- ✓ No se sabe con certeza si un rechazo ya ha finalizado.
- ✓ Existe una pérdida de tiempo al realizar los rechazos, ya que se requieren de autorizaciones por parte de distintas áreas las cuales se encuentran en diferentes departamentos y el trasladarse implica el traspapelado, pérdida o retraso en la autorización.
- ✓ El seguimiento de los rechazos es enviado cada semana a todas las áreas, durante la recolección se dan cuenta que existen variaciones en la información, lo que impide tenerla actualizada diariamente.
- ✓ Al existir un desorden en el seguimiento de los rechazos, la pérdida de información es muy frecuente lo que retrasa la liberación del material.
- ✓ No existe ningún tipo de búsqueda de rechazos, ni documentos que muestren el seguimiento.
- ✓ Existen retrasos en la obtención de la información.
- ✓ El material es retenido por mucho tiempo en almacén, provocando pérdidas al proveedor y en el producto final.
- ✓ No se tiene información veraz ni completa para realizar la inspección de proveedores, provocando una constante amenaza para la calidad del producto.

1.5. Diagrama de flujo

En la figura 1.1 se muestra el diagrama de flujo que describe la secuencia de actividades a seguir para el control de producto no conforme.

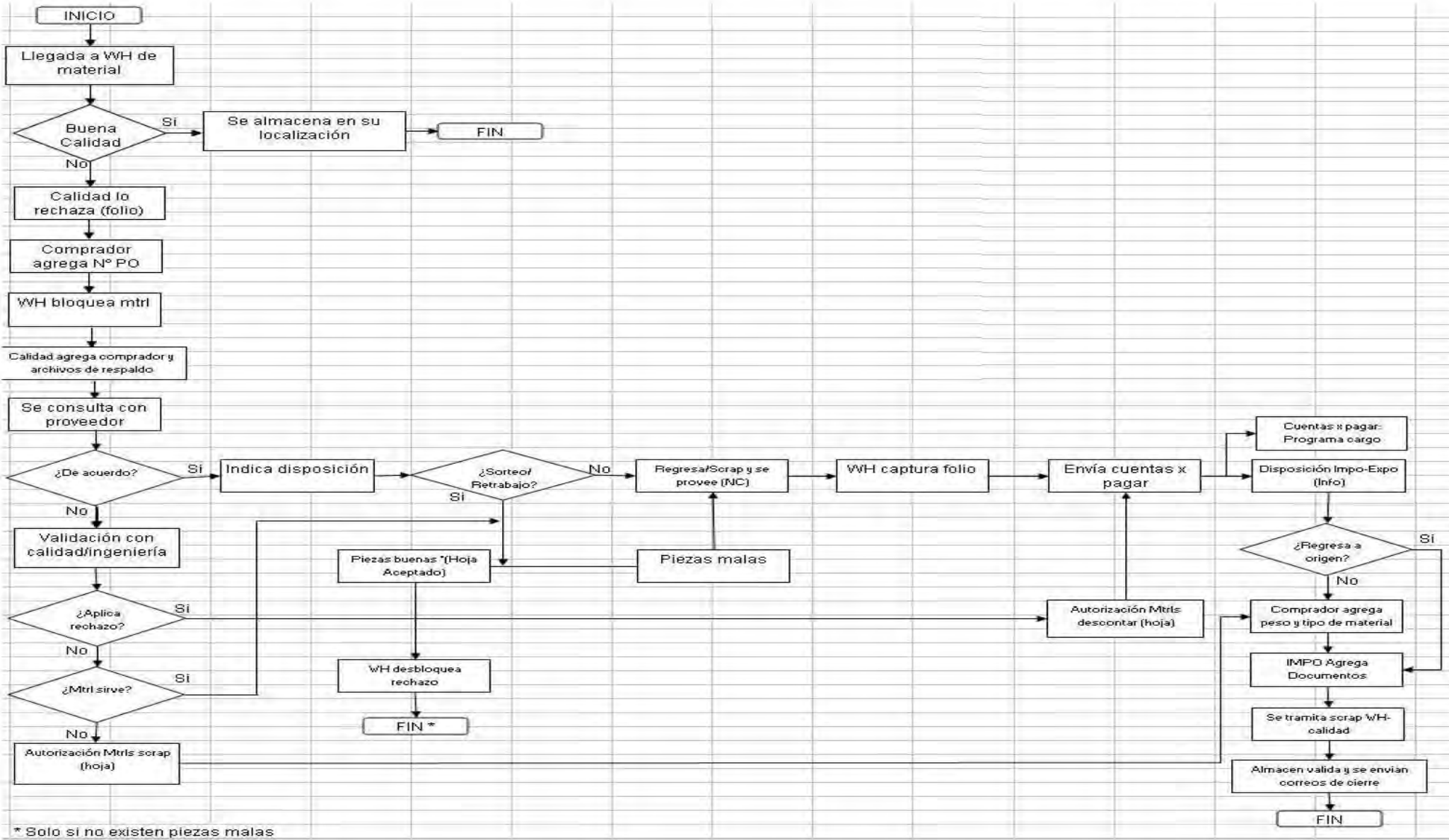


Figura 1.1. Proceso producto no conforme.

1.6. Descripción del proceso.

1. Recepción de material: El personal involucrado en este proceso es el encargado de recibir toda la materia prima de nuestro producto/servicio, la cual será llevada a almacén, o al área correspondiente para su utilización o inspección.
2. Detectar producto y/o servicio no conforme. Un producto y/o servicio no conforme es aquel que no cumple con los requisitos establecidos en cada proceso. El producto y/o servicio no conforme puede ser detectado durante o después de la ejecución del proceso por el personal involucrado en los procesos, los clientes o usuarios (queja) y como resultado de una auditoria.
3. Generar folio: Al recibir la notificación de producto no conforme, el líder de proceso debe registrar el producto no conforme en el formato F-SGC-11-83, que incluye:

Número de factura
Fecha ingreso materia
Fecha rechazo
Vendedor
Número de parte
Número de piezas
Número de documento
Descripción de rechazo
Partes buenas
Partes defectuosas
Resultado de la inspección
Inspector
Lugar inspección

4. Agregar número de PO². La persona encargada de generar el rechazo, es la indicada para notificar al comprador que se ha generado una no conformidad, la cual deberá asignar el número de PO correspondiente.

²PO (Purchase Order).- Orden de compra generada al momento de comprar el producto y/o material.

5. Bloqueo de material. El encargado de almacén bloqueará el material correspondiente dependiendo de la información proporcionada por el comprador la cual deberá contener los siguientes datos:

Folio
Cantidad
Número de factura
Número de parte
Número de proveedor

6. Consulta proveedor. Una vez que el comprador hace recolección de la información producida en la no conformidad, éste deberá dialogar con su proveedor todos los datos recabados, pretendiendo llegar a las causas y posible solución conforme al análisis de lo recabado, permitiendo evaluar la situación y poder dar el seguimiento correspondiente.
7. Indicar disposición. Una vez determinada la causa raíz del problema, los responsables de proceso definen el tratamiento a seguir para el producto no conforme, dando el seguimiento con forme a la siguiente información:

No se resorteo/retrabajo: Indica que el material tiene bases justificables para ser manejado como producto no conforme.

Nota: Todos los rechazos generados en este punto deberán agregar la información correspondiente a los puntos VIII, IX Y X

Se resorteo/retrabajo: Existe una corrección total o parcial del producto y proseguirán con lo descrito en el punto XVI.

8. Proveer NC³ y/o SCRAP⁴. El comprador responsable del rechazo tendrá que proporcionar al encargado la nota de crédito.

Nota: Todos los rechazos cuyo origen sea que no se resorteo o se retrabajo, tendrán que agregar tanto NC como SCRAP dado que se ha generado un material inservible.

9. Almacén captura folio. Almacén proporciona el número GR⁵ en base al número de folio proporcionado por el comprador.

³ NC (Nota de Crédito). Es un documento por el cual una empresa comunica a su cliente haberle disminuido su deuda.

⁴ SCRAP. Hoja generada por el área correspondiente, la cual contendrá información respecto al desperdicio que se genero de un

⁵ GR (Good Receipt): Número proporcionado por la factura que se genera al momento de rechazar algún material.

10. Cuentas por pagar captura folio. Se proporciona el número IR⁶ al folio, proporcionado por el comprador, con esto se da por enterado al área para iniciar los trámites correspondientes para la cancelación y devolución del material al proveedor.

Nota: El orden al momento de agregar la información de los puntos VIII, IX Y X puede ser aleatorio por lo que al momento de tener la información correspondiente en cada punto se continuará con el punto XI.

11. Disposición IMPO-EXPO. Se pone en la bandeja del usuario para validar si procede o no el rechazo.
12. Agregar peso y tipo de material. El comprador agregará el peso y el tipo de material correspondiente al rechazo generado.
13. IMPO-EXPO agrega documentos. Se agregarán los documentos correspondientes, generados en el paso X.
14. Se tramita SCRAP. El área correspondiente generará la hoja, la cual será proporcionada al comprador.
15. Almacén valida. Se validará toda la información, así como los documentos correspondientes para liberar el material.

Nota: Todos los rechazos que pasen por este punto, se les realizará una revisión para verificar si pasaron por el punto XVI, de confirmarse su paso por este punto se verifican las hojas de aceptado para validar si existieron piezas rechazadas, de ser verdadero se verificara que se halla hecho el desbloqueo pertinente. Al termino se dará por finalizado el proceso de lo contrario el cierre estará a cargo del punto XVII.

16. Piezas buenas y malas. El comprador ha decidido que si se resorteó o se retrabajó, en base a esto el área correspondiente con la documentación y revisión del material generara la(s) hoja(s) de aceptado, permitiendo obtenerlas de forma parcial o no, con esto se indica que podrán generarse dos hojas especificando el número de piezas aceptadas y rechazadas en cada una.

⁶ IR (Invoice Receipt): Número proporcionado por la factura que se genera al momento de recibir algún material.

Nota: Todos los rechazos generados en este punto deberán agregar la información correspondiente a los puntos VIII, IX Y X.

17. Desbloqueo de almacén. Con base en las hojas de aceptado proporcionadas por el área de calidad, las cuales contendrán el número de piezas aceptadas y/o rechazadas, se procederá hacer el desbloqueo correspondiente de material, para seguir con su uso.

Nota: En caso de que sólo exista una hoja de aceptado y no se hayan encontrado piezas malas, al igual si existen rechazos procedentes del punto XV, se terminara el proceso en este punto.

18. Documento autorización de materiales descontar. El rechazo aplica por lo que el área correspondiente procederá a proporcionar la hoja de los materiales rechazados y se continuará con el procedimiento descrito en el punto X.
19. Validación calidad/ingeniería. El proveedor no estuvo de acuerdo con el material rechazado por lo que las áreas correspondientes realizarán una evaluación del material, haciendo las comparaciones pertinentes para llegar a una conclusión proporcionando dos respuestas las cuales serán compradas con base en la siguiente tabla 1.1 para dar una respuesta:

Calidad	Ingeniería	Respuesta
Aplica	Aplica	Aplica
Aplica	No Aplica	No Aplica
No Aplica	Aplica	No Aplica
No Aplica	No Aplica	No Aplica

Tabla 1.1. Validación de respuestas.

Nota: Si la respuesta de ingeniería es no aplica, este tendrá que agregar documentos de respaldo para validar dicha respuesta.

20. Hoja SCRAP. El material es considerado como inservible por lo cual se deberá agregar por parte del área correspondiente la hoja SCRAP y se continuara con lo dicho en el paso XII.

CAPÍTULO II
DISEÑO DEL SISTEMA

2. Diseño del sistema

2.1. Modelado del sistema

“El modelado de sistemas está jugando un rol cada vez más importante ya que permite obtener de manera abstracta la vista de dicho sistema, facilitando así la comunicación de ideas entre clientes y usuarios además de que se reduce el problema que se está estudiando, centrándose sólo en un aspecto a la vez.”⁷

Por otra parte para desarrollar un software de calidad y duradero es necesario idear una solida base que pueda ser flexible al cambio, para ello el modelado es una técnica de ingeniería probada y aceptada; no sólo en la construcción de un gran edificio, un avión, una represa hidroeléctrica, sino que debe serlo también en ingeniería del software.

El modelo es una simplificación de la realidad y nos proporciona planos detallados del sistema, así como una visión global de este, dicho modelo puede ser estructural destacando la organización del sistema o puede ser de comportamiento, resaltando la dinámica.

Gracias al modelado podemos conseguir cuatro objetivos:

1. Nos ayudará a visualizar cómo es y cómo se quiere que sea el sistema, permitiendo obtener un panorama de este.
2. Nos permite especificar su estructura y comportamiento para poder entender de manera general el funcionamiento.
3. Nos proporciona las plantillas que nos guiarán en la construcción del sistema para facilitar el trabajo en cada formulario.
4. Documentar las decisiones que hemos adoptado para cada situación.

“En el modelado existen cuatro principios los cuales nos ayudan a entender el porqué es necesario modelar un sistema”.⁸

- a) La forma como vemos el problema y la elección de los modelos a crear tendrá una profunda influencia sobre cómo acotemos el problema y le damos solución al mismo.

Para poder entender bien el sistema es necesario visualizarlo como objetos los cuales estarán compuestos por abstracciones de la realidad y de la solución del

problema, este es precisamente el paradigma al que le apuesta UML⁹: el modelo orientado a objetos.

Con base en la forma en que se visualiza el problema se tendrán que elegir bien los modelos, porque los modelos adecuados podrán facilitar el entendimiento de los problemas de desarrollo más difíciles, ofreciéndonos una mejor comprensión que simplemente no podríamos obtener de otra manera; los modelos erróneos desorientarán, haciendo que se centren en cuestiones irrelevantes.

- b) Todo modelo puede ser expresado a diferentes niveles de precisión.

La precisión se puede ver desde dos ópticas:

La primera es el grado de detalle con que se representa un modelo; por ejemplo, si lo que se desea es razonar acerca de los requerimientos del sistema con un cliente o usuario final, se puede elaborar un diagrama de casos de uso de forma que al ser parte del análisis nos ayudan a describir qué es lo que el sistema debe hacer. Los casos de uso son el qué hace el sistema desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente.

La segunda forma de ver la precisión de un modelo se refiere al nivel de abstracción, es decir, a los detalles y la vista (porción del sistema o realidad) que presenta un modelo al usuario; por ejemplo, en el sistema de rechazos maneja hojas de aceptado que realizan los usuarios, los diagramas de secuencias permiten visualizar el comportamiento que tiene permitiendo tener una gran precisión en los detalles de este proceso.

“Los mejores tipos de modelos son aquellos que permiten elegir el grado de detalle, dependiendo de quién está viendo el sistema y por qué necesita verlo”.¹⁰

- c) Los mejores modelos están ligados a la realidad.

Todos los modelos simplifican la realidad; el truco está en asegurarse de que las simplificaciones no enmascaran ningún detalle importante.

Una de las formas de representar la realidad, son los diagramas de casos de uso ya que documentan el comportamiento de un sistema desde el punto de vista del

⁹ UML (Unified Modeling Language, Lenguaje Unificado de Modelado).

usuario. Por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar.

- d) Un único modelo no es suficiente. Cualquier sistema no trivial se aborda mejor a través de un pequeño conjunto de modelos casi independientes

La idea es tener modelos que se pueden construir y estudiar separadamente, pero que aún así, estén interrelacionados, es por eso que existen varios diagramas en UML permitiendo modelar diferentes aspectos del sistema, desde las vistas lógicas y físicas del sistema hasta los aspectos dinámicos, estáticos y funcionales del mismo.

2.2. Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan.

UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware, y organizaciones del mundo real. En el caso del sistema de rechazos es un sistema web orientado a objetos, con múltiples usuarios de distintas áreas las cuales tendrán accesos a módulos separados para poder seguir el proceso que tendrá un producto no conforme.

Los elementos de UML se muestran mediante diagramas que presentan múltiples vistas del sistema, ese conjunto de vistas son conocidos como modelos.

UML dispone de varios diagramas los cuales pueden representar distintos aspectos del sistema así como tener un mejor entendimiento de cómo funcionaría el sistema, además de poder detectar subprocesos que tal vez no se verían sin un buen modelamiento.

Para el sistema de rechazos se dispondrán de los siguientes diagramas.

- ✓ Diagramas de paquetes. Un diagrama de paquetes muestra cómo nuestro sistema estará dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica del sistema, nos ayudará a desmenuzar el sistema y poder identificar de manera rápida a que módulos tendrá acceso cada área.
- ✓ Casos de uso. Nos especificarán el comportamiento del sistema o de una parte, y es una descripción de un conjunto de secuencias de acciones, incluyendo

variantes, que ejecutará el sistema para producir un resultado observable de valor para un actor.

- ✓ Diagramas de actividades. Nos ayuda a representar paso a paso los flujos de trabajo del sistema para modelar una secuencia de acciones y condiciones tomadas dentro de los casos de uso que integran el sistema.

2.3. El Proceso Racional Unificado (RUP)

En el desarrollo de software no existen proyectos que sean iguales. Cada uno tiene distintas prioridades, requerimientos y tecnologías diferentes. Sin embargo, en todos los proyectos, se debe minimizar el riesgo, garantizar la predictibilidad de los resultados y entregar a tiempo software de calidad. Rational Unified Process, o RUP, es una plataforma flexible de procesos de desarrollo de software que ayuda brindando guías consistentes y personalizadas de procesos para todo el equipo de proyecto.

RUP describe cómo utilizar de forma correcta las reglas de negocio y procedimientos probados en el desarrollo de software. El proceso captura varias de las mejores prácticas en el desarrollo moderno de software en una forma que es aplicable para un amplio rango de proyectos y organizaciones. Es una guía de cómo utilizar de manera efectiva UML.

Provee a cada miembro del equipo fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas de desarrollo. Crea y mantiene modelos, en lugar de enfocarse en la producción de una gran cantidad de papeles de documentación.

Una de las principales características de RUP es la flexibilidad de desarrollar iterativamente. Rational Unified Process organiza el proyecto en término de disciplinas y fases, consistiendo cada una en una o más iteraciones. La aproximación iterativa ayuda a mitigar los riesgos en forma temprana y continua, con un progreso demostrable y frecuentes release ejecutables. Además provee un entorno de proceso de desarrollo configurable basado en estándares; permite tener claro y accesible el proceso de desarrollo que se sigue y que este sea configurado a las necesidades del proyecto.

RUP maneja 6 principios claves de los cuales sólo se utilizarán 5 ya que uno de ellos se enfoca a la colaboración entre equipos para el desarrollo del sistema, pero para este sólo se involucra una persona, es por eso que se omite éste y a continuación se detalla cada principio:

1. Adaptación del proceso

El proceso se debe adaptar a las necesidades propias del proyecto.

2. Balancear prioridades

Debido a que el proyecto involucra distintas áreas es necesario encontrar un balance que satisfaga las expectativas y requerimientos de los diversos inversores, tomando en cuenta que los recursos son limitados.

3. Demostrar valor iterativamente

El proyecto se analizará en etapas iteradas, dado que son 5 áreas involucradas, es imprescindible analizar la opinión de los involucrados así como la estabilidad y calidad del sistema y por consiguiente revisar todos los requerimientos y nuevas necesidades e inquietudes que llegasen a surgir.

4. Elevar el nivel de abstracción

Es necesario entender al 100% los requerimientos y funcionamiento del sistema, debido a esto es importante acompañar el desarrollo con representaciones visuales de arquitectura, por ejemplo UML nos proporcionara diagramas tales como:

- ✓ Diagrama de paquetes
- ✓ Diagramas de caso de uso
- ✓ Diagrama de actividades
- ✓ Diagrama de secuencia
- ✓ Diagramas de clase

Los cuales nos permiten representar de forma gráfica el funcionamiento y usabilidad del sistema.

5. Enfocarse en la calidad

La calidad es la representación de todo el trabajo al final del desarrollo, por tal motivo es importante analizar durante todo el proceso y al final de las iteraciones que todo este en perfecto funcionamiento, para poder tener un software de calidad y funcional.

RUP propone un ciclo de desarrollo el cual consta de cuatro fases.

- ✓ Inicio

Busca evaluar el proyecto, en este punto se decide si se continúa o no con el proyecto en función del impacto que tendrá, riesgos y los imperativos económicos, se determinan los principales casos de uso y se realizará el primer diagrama de flujo el cual permitirá mostrara a grandes rasgos el funcionamiento del sistema.

✓ **Elaboración**

Permite construir la arquitectura del sistema, permitiendo realizar un plan de proyecto, se concluye la elaboración y se conocen finalmente las exigencias del proyecto y su arquitectura.

✓ **Construcción**

Corresponde al desarrollo de software de la arquitectura determinado durante la fase de elaboración, dando como resultado un sistema totalmente operativo, eficiente y el manual de usuario.

✓ **Transición**

Comprende la instalación del software y la distribución de la liga al cliente y se capacita a los usuarios. En este punto surgen nuevas peticiones y modificaciones los cuales serán analizados para su corrección y/o edición.

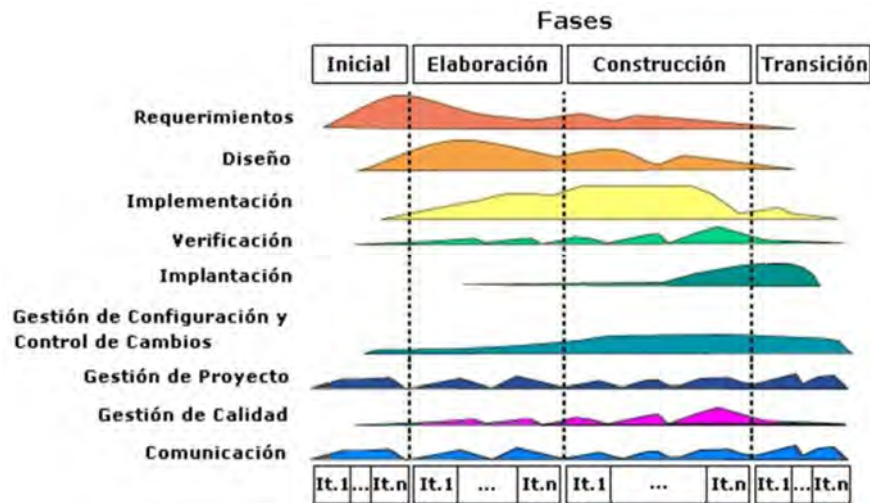


Figura 2.1. Fases e Iteraciones de la Metodología RUP.

2.4. Descripción de las actividades

2.4.1. Fase de inicio

En la fase de inicio las iteraciones son más relevantes y se les pone mayor énfasis a las actividades que modelarán los requisitos del sistema, en esta fase se obtendrá una lista de lo que se requiere que realice el sistema. Para asegurarnos de que un requisito es claro, debemos saber qué persona lo ha propuesto y por qué lo ha propuesto, qué es lo que cree exactamente que vamos a hacer en nuestro programa cuando nos ha dicho ese requisito.

El equipo encargado de levantar los requerimientos realizará una junta en la cual se involucraran las 5 áreas para conocer el proceso.

- ✓ Entender el problema actual, permitiendo identificar el objetivo y las mejoras para cada área.
- ✓ Entender la estructura y la dinámica en conjunto e individual, dado que el proceso depende de distintos involucrados.
- ✓ Identificar posibles redundancias en el proceso.
- ✓ Entender el proceso con un caso real para poder proponer mejoras y evitar posibles omisiones de pasos durante el rechazo.
- ✓ Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común.
- ✓ Estimar tiempo de desarrollo del sistema.
- ✓ Se identifican todos los casos de uso.
- ✓ Se analizan todos los requerimientos levantados en el primer capítulo, se identifican dudas o mejoras y se comentan con el área o experto involucrado.

2.4.2. Fase de elaboración

En la fase de elaboración, las iteraciones se orientan al desarrollo con base en la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo del negocio (refinamiento), análisis, diseño y una parte de implementación orientado a la base de la arquitectura.

Durante el análisis vamos a definir más claramente qué es lo que va a hacer nuestro programa y para ello se dividirá en 5 pasos los cuales nos ayudarán a darle continuidad a nuestro sistema.

2.4.2.1. Identificar actores

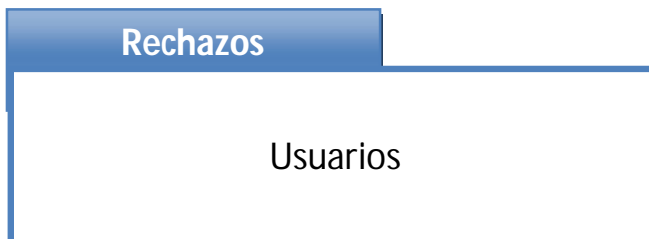
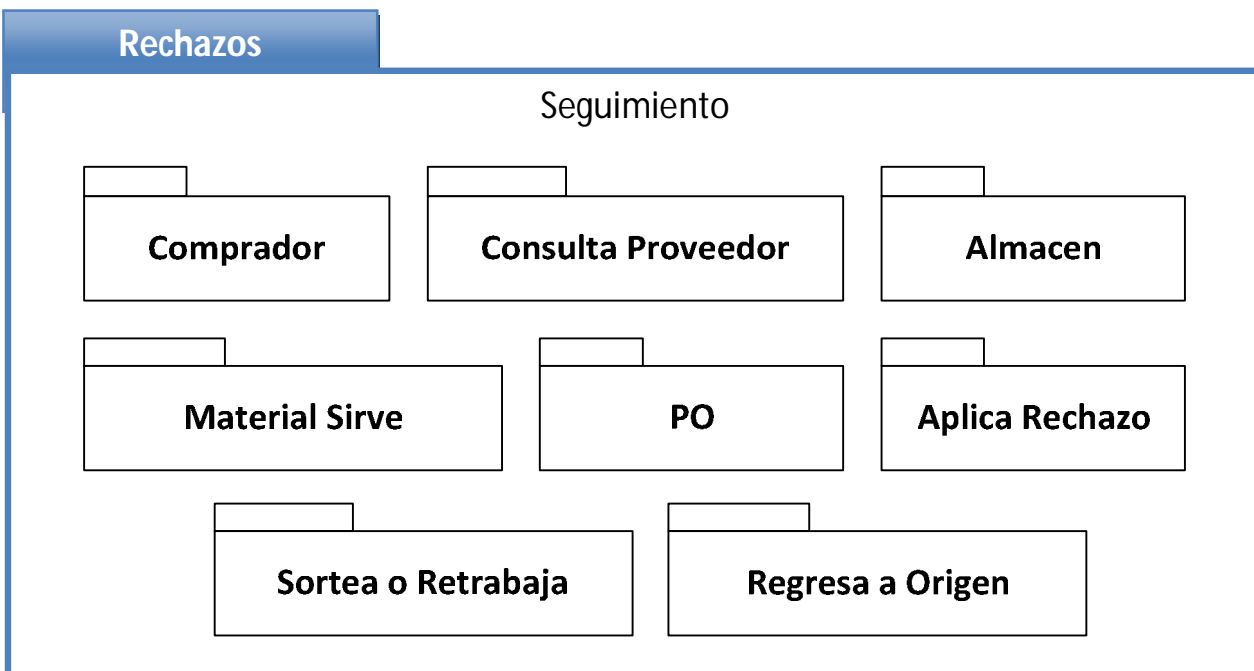
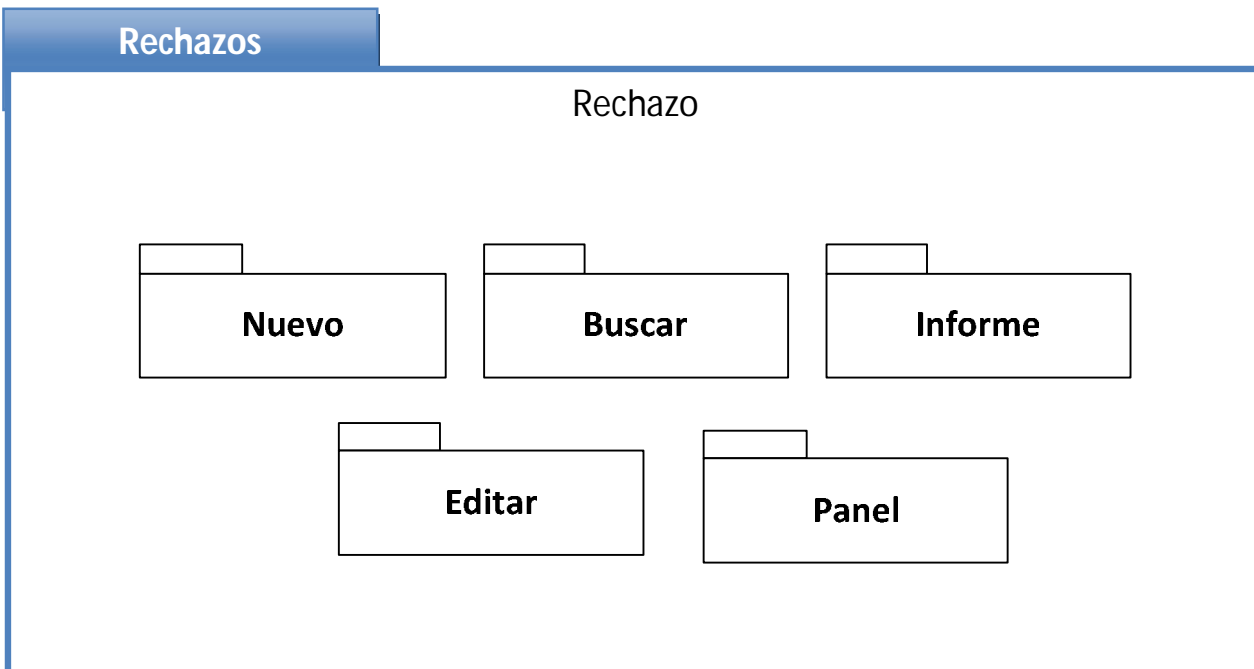
Un actor del sistema es quien interactúa o hace uso del sistema, es decir todas aquellas áreas que tendrán acceso al sistema de rechazos. La correcta identificación de los actores nos ayudará a poder definir el número de roles y sus respectivos privilegios para poder garantizar que cada uno tenga acceso solo a lo que le compete. La figura 2.2 muestra los actores con los que cuenta el sistema.



Figura 2.2. Actores del sistema.

2.4.2.2. Diagramas de paquetes

Los diagramas de paquetes nos ayudan a identificar los apartados que tendrá el sistema y las funciones que realizarán. Permitiendo agrupar los procesos de acuerdo a su funcionalidad.



2.4.2.3. Identificar casos de uso

Los casos de uso nos muestra la relación que existe entre los actores y los casos de uso del sistema. Representan la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa. Estos representan también el sistema como una caja rectangular con el nombre en su interior. Los casos de uso están en el interior de la caja del sistema, y los actores fuera, y cada actor está unido a los casos de uso en los que participa mediante una línea.

En los diagramas de casos de uso los muñecos son los actores y los óvalos son los documentos de casos de uso. Por lo cual se dibuja un muñeco por actor y un ovalo por cada caso de uso y se enlazan con líneas cuando existe una relación entre ellos. Con esto se consigue una visión general de cómo los diferentes actores interactúan con los distintos casos de uno.

En las siguientes figuras se muestran los casos de uso que representan el sistema.

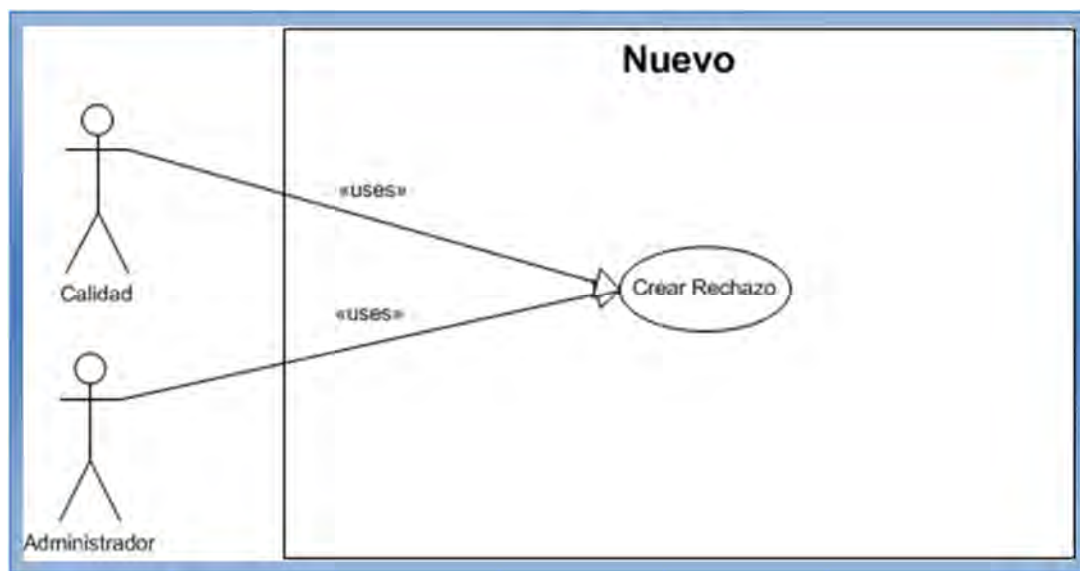


Figura 2.3. Caso de uso para la operación nuevo.

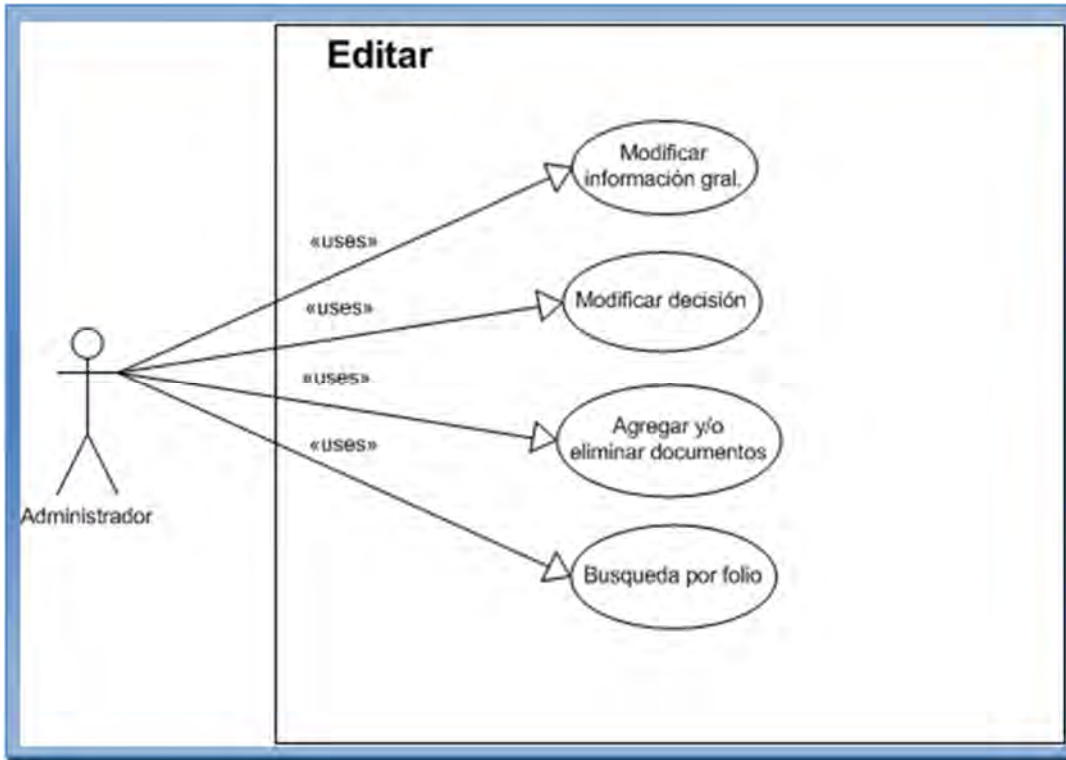


Figura 2.4. Caso de uso para la operación editar.

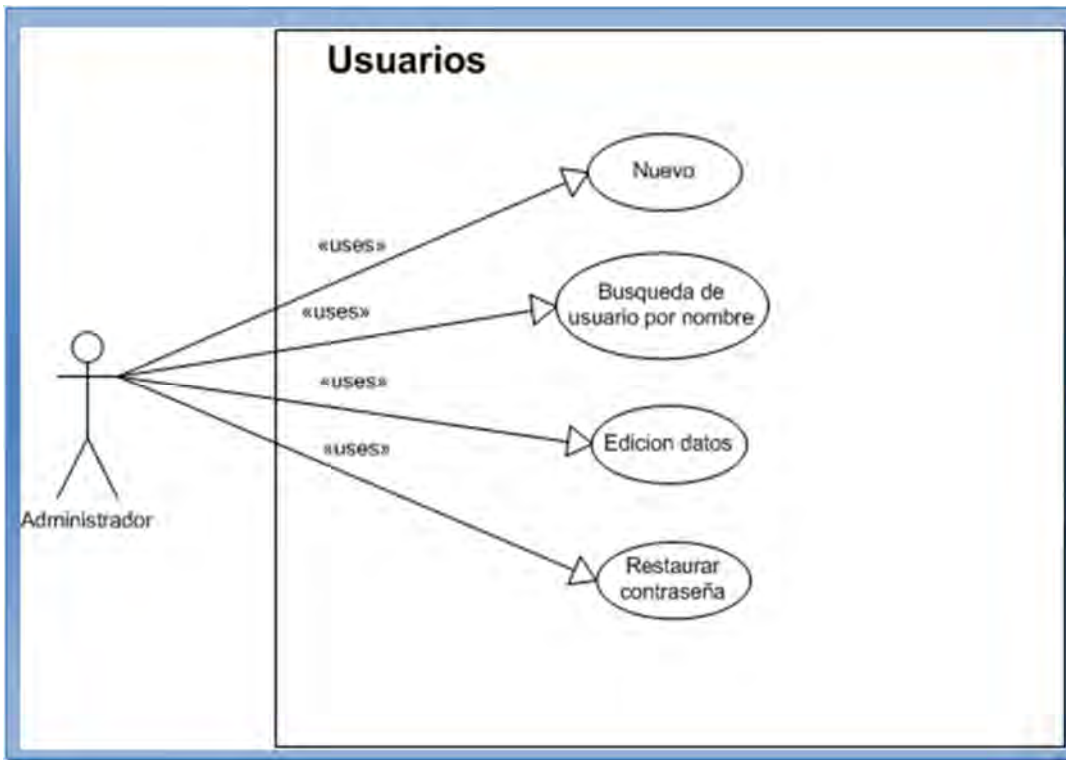


Figura 2.5. Caso de uso para la operación usuarios.

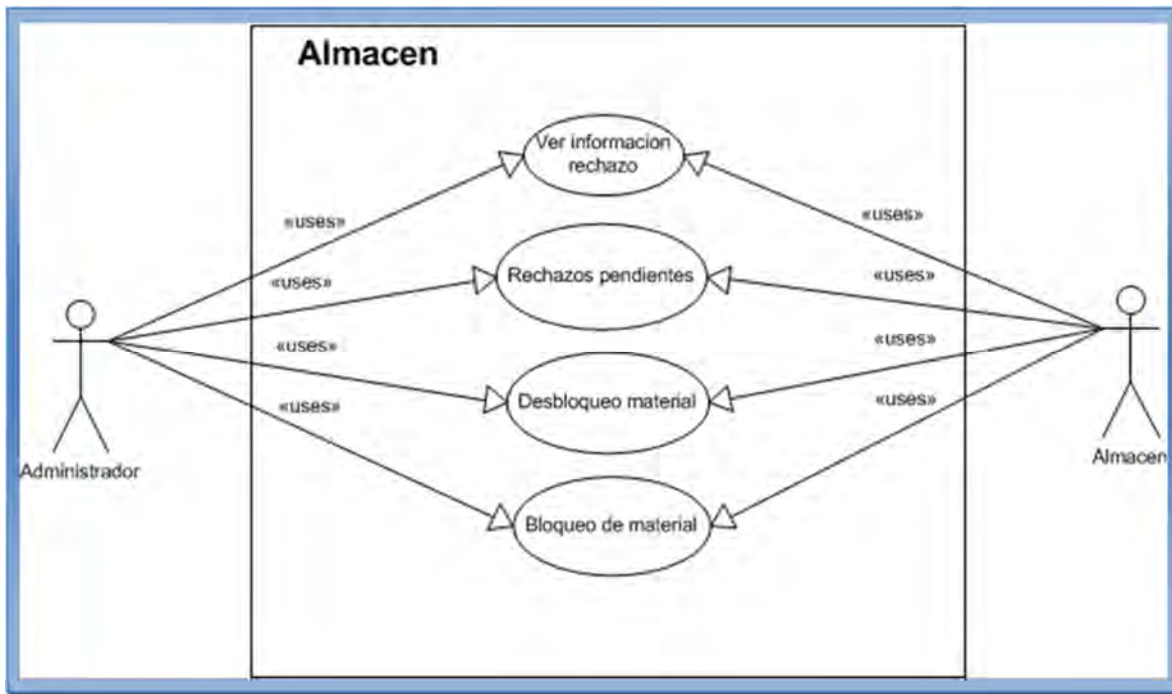


Figura 2.6. Caso de uso para la operación almacén.

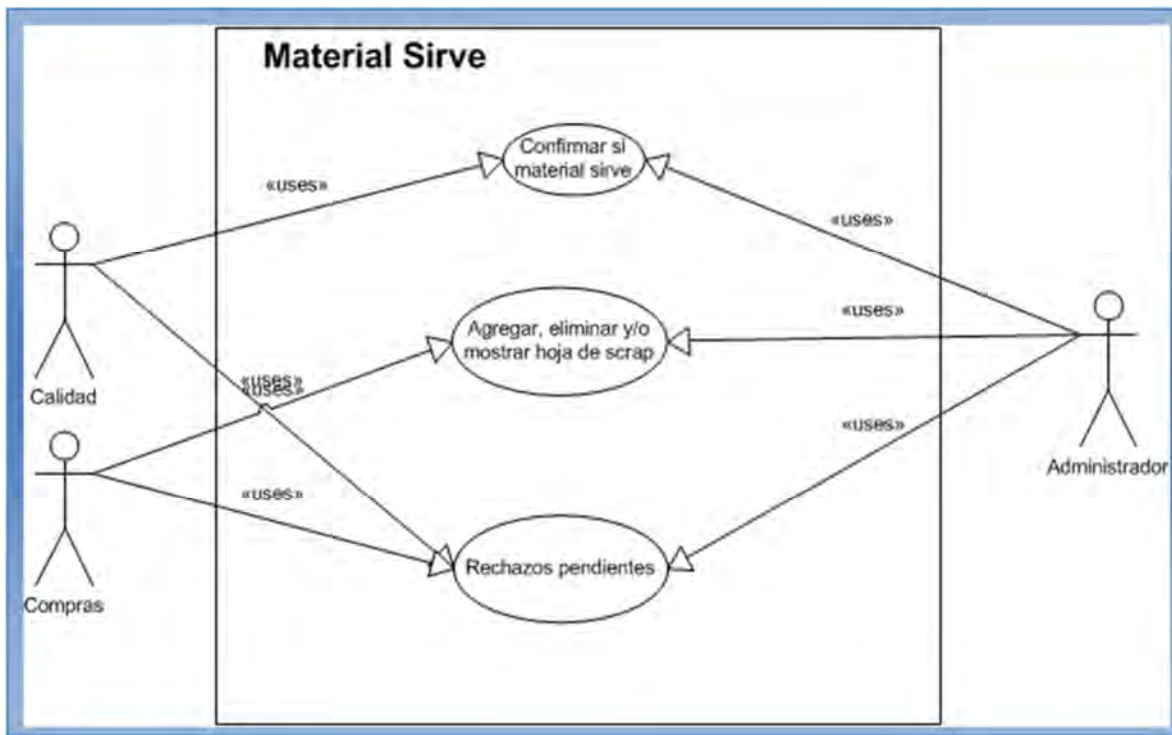


Figura 2.7. Caso de uso para la operación material sirve.

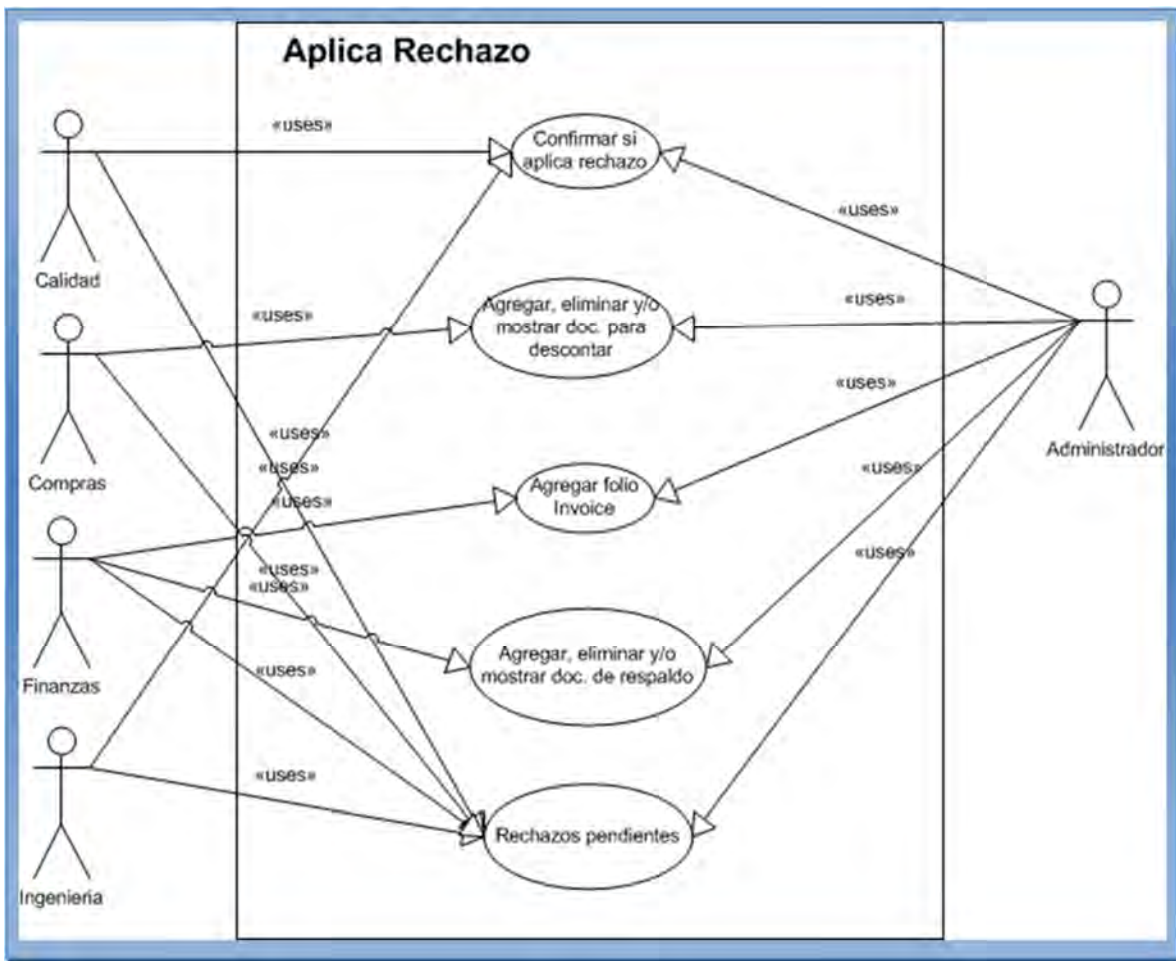


Figura 2.8. Caso de uso para la operación aplica rechazo.

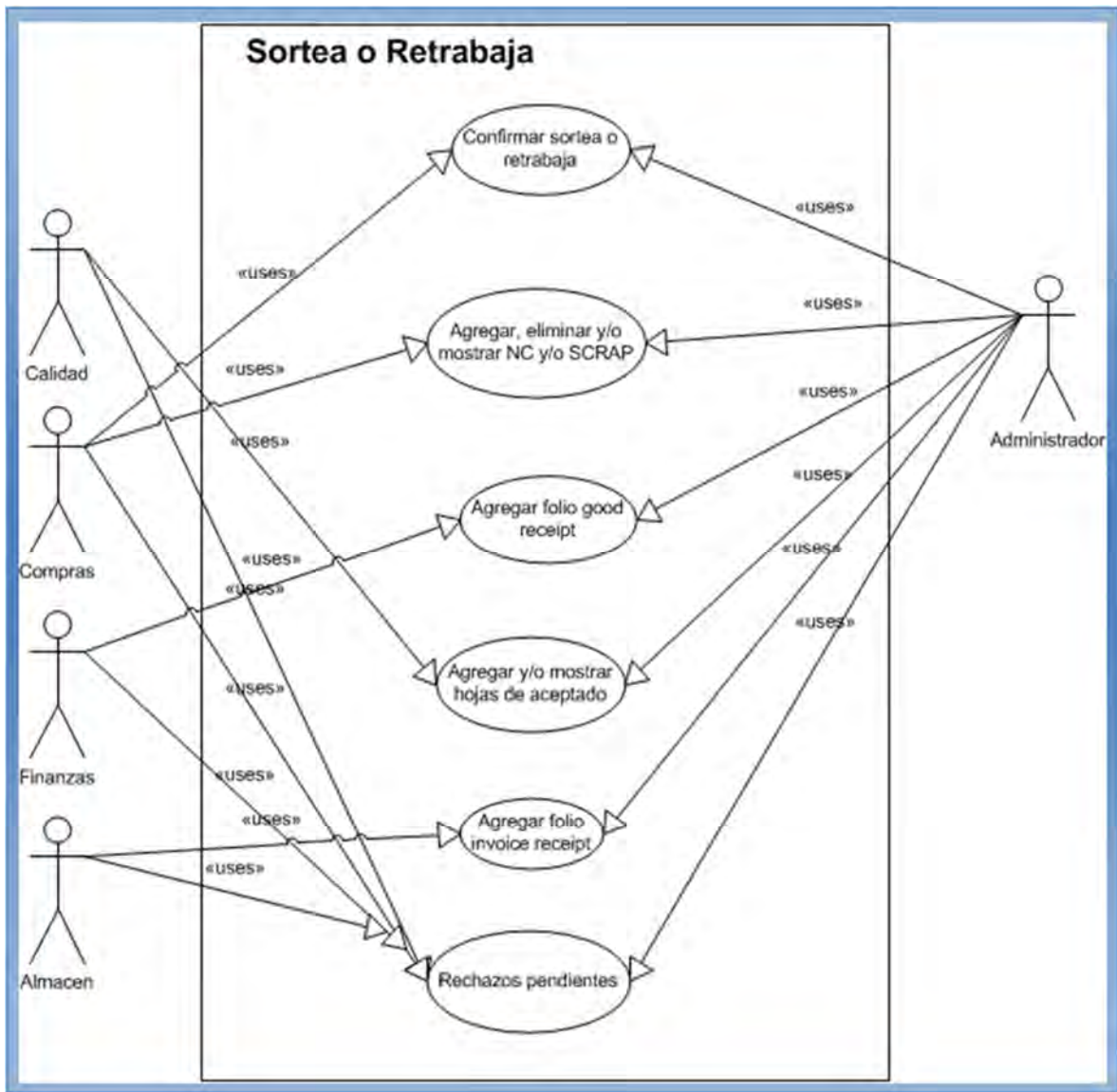


Figura 2.9. Caso de uso para la operación sorteó o retrabajó.

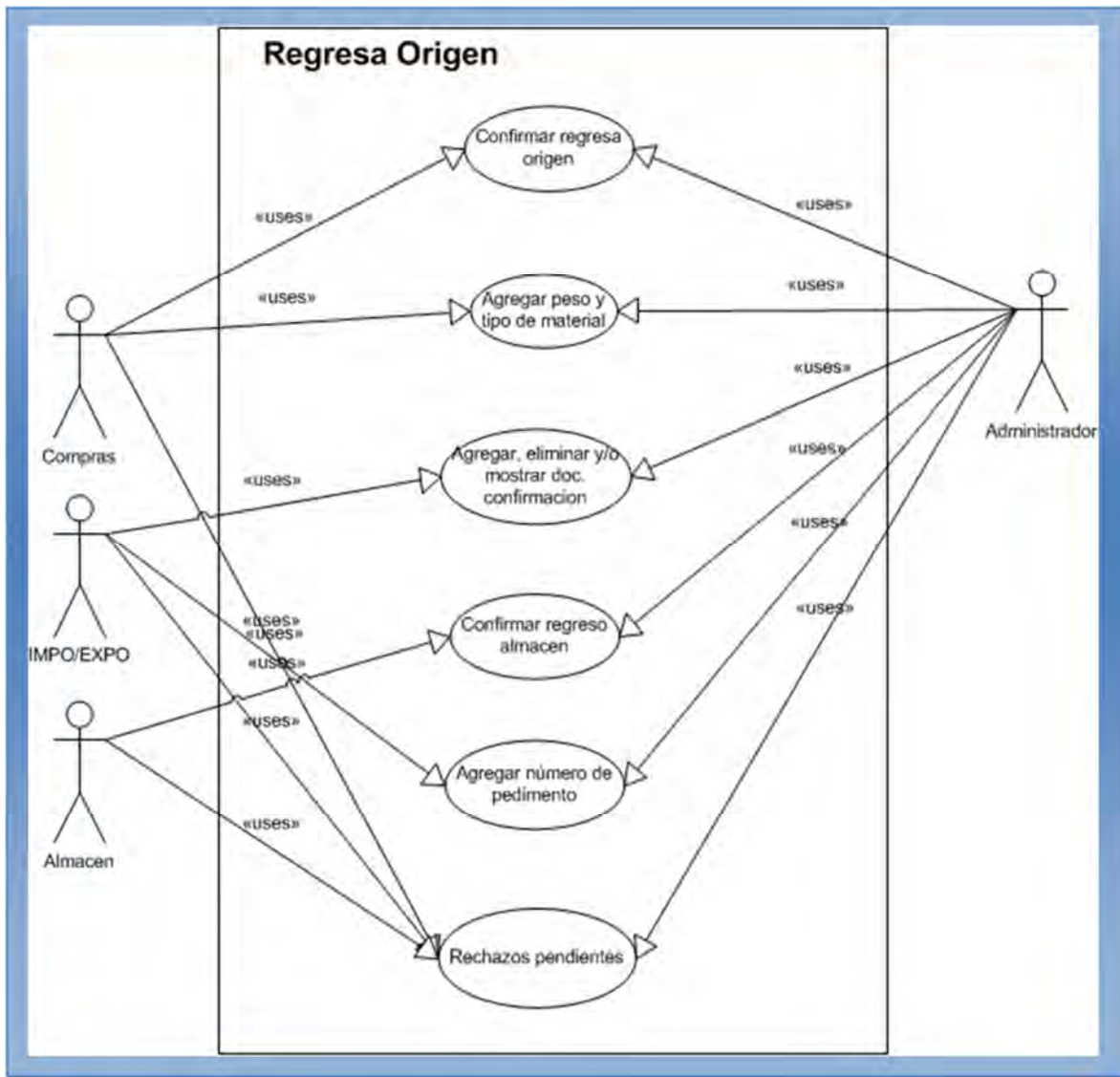


Figura 2.10. Caso de uso para la operación regresa origen.

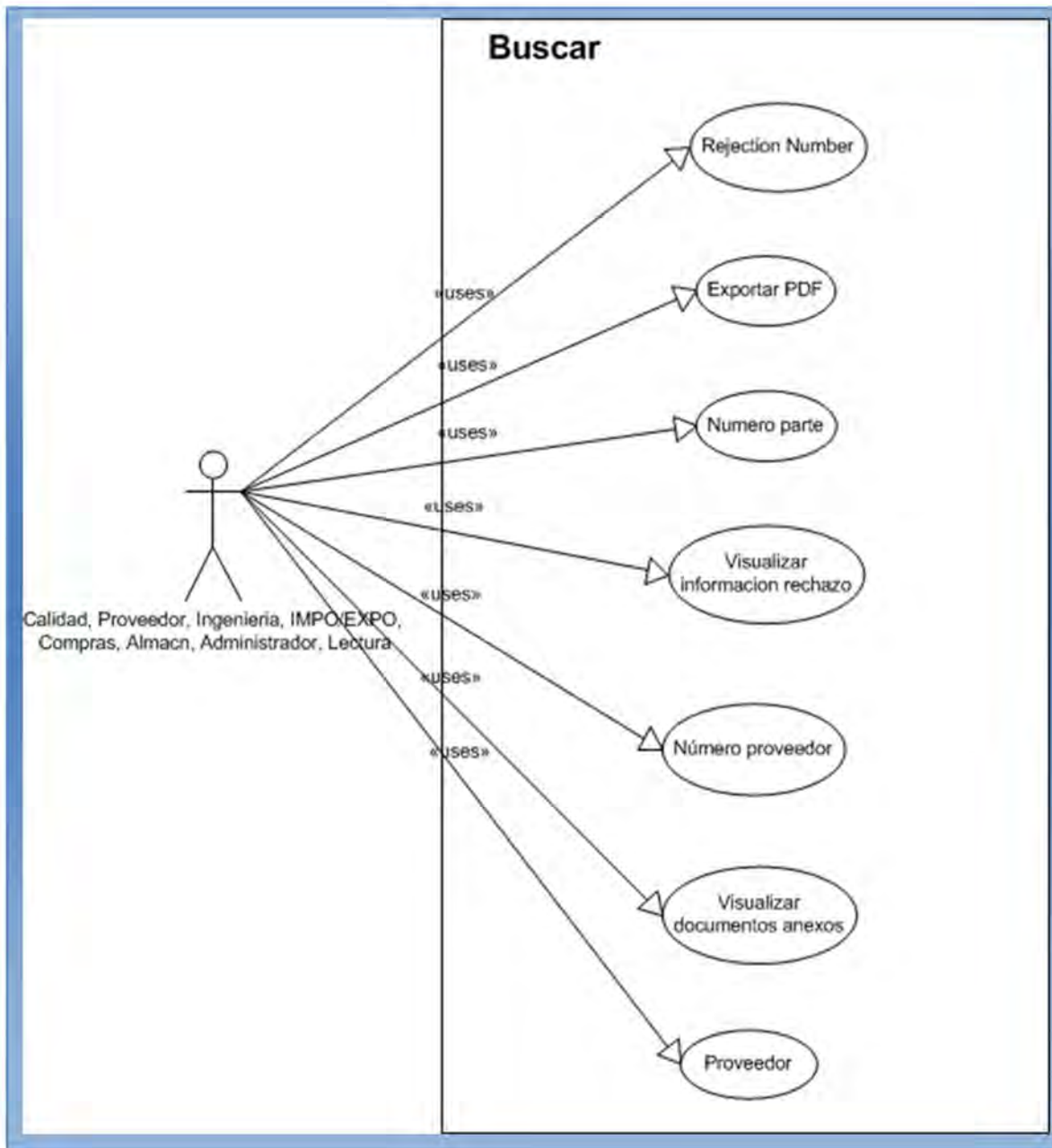


Figura 2.11. Caso de uso para la operación buscar.

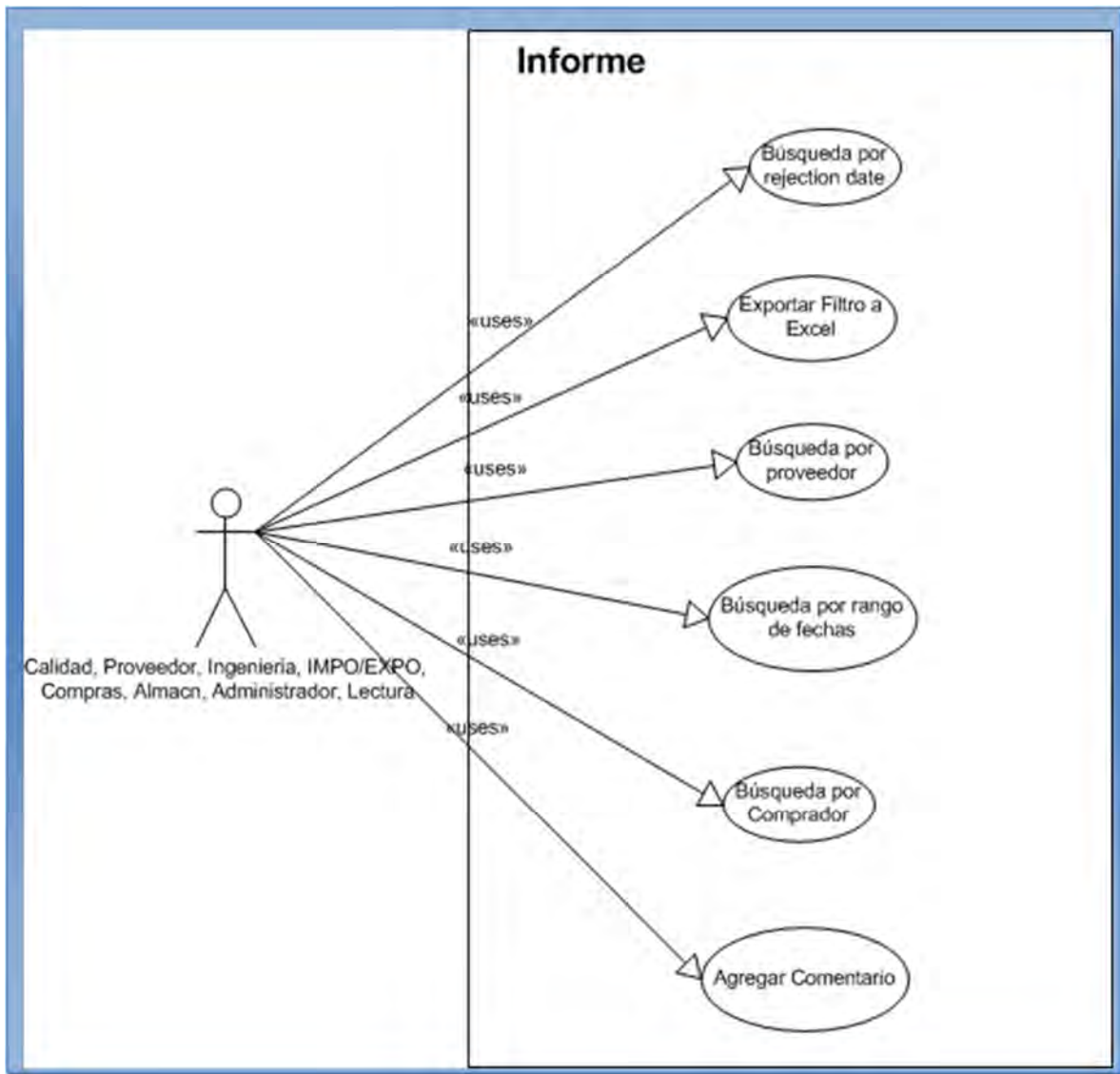


Figura 2.12. Caso de uso para la operación informe.

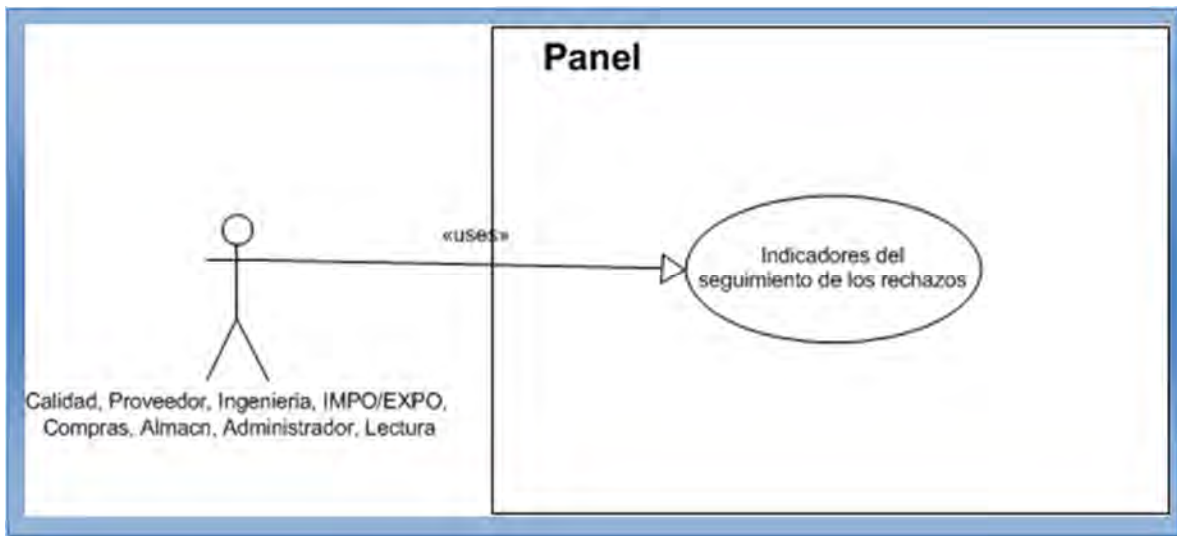


Figura 2.13. Caso de uso para la operación panel.

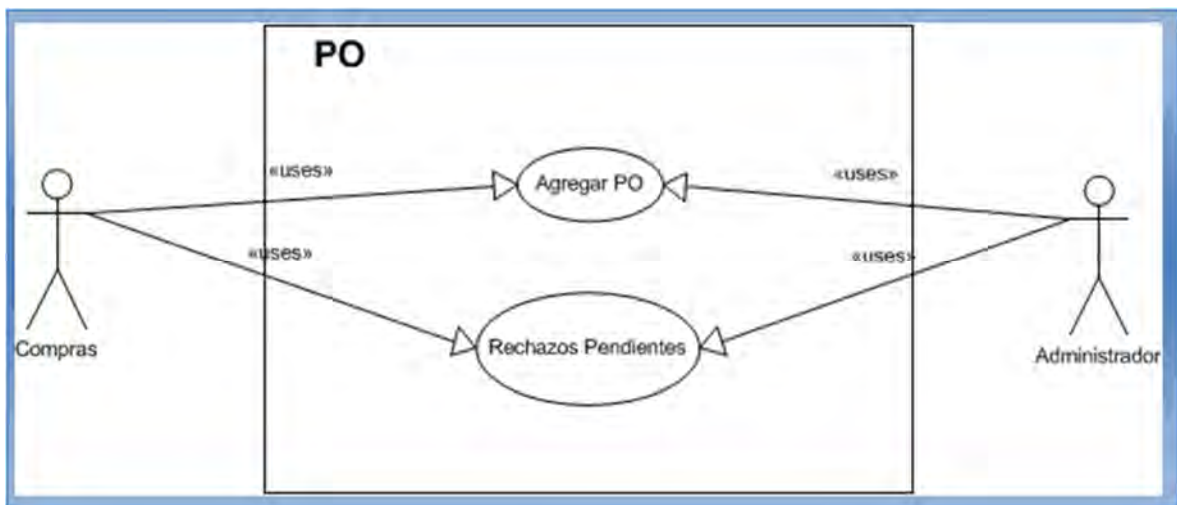


Figura 2.14. Caso de uso para la operación PO.

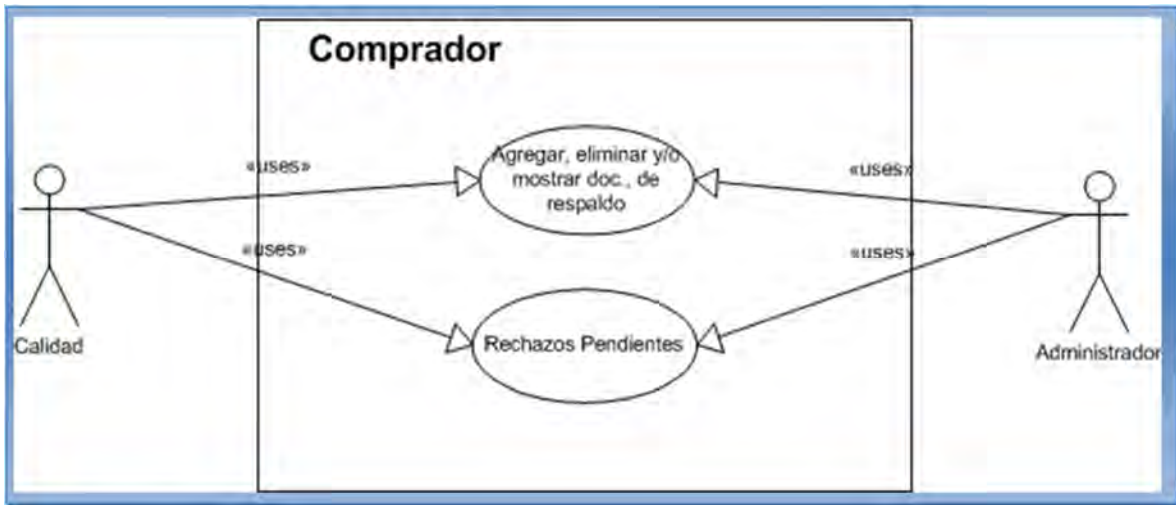


Figura 2.15. Caso de uso para la operación comprador.

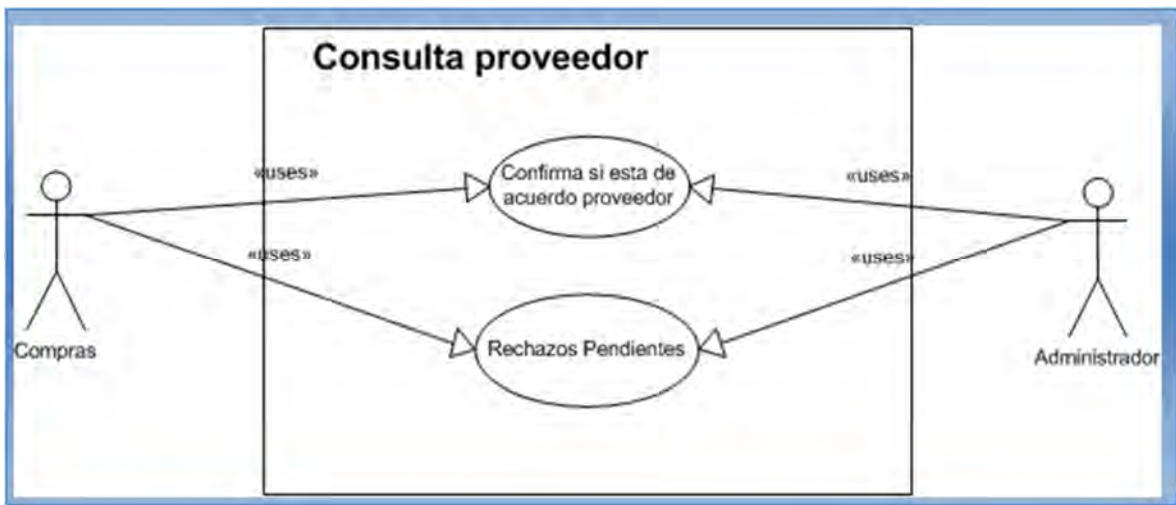


Figura 2.16. Caso de uso para la operación consulta proveedor.

2.4.2.4. Desarrollo de los casos de uso

El desarrollo de los casos de usos forman parte del análisis nos ayuda a describir qué es lo que es sistema debe hacer. Los casos de uso son qué hace el sistema desde el punto de vista del usuario. Es decir, se explica la forma de interactuar entre el sistema y el usuario.

Saltándome los campos evidentes como nombre, autor, fecha y descripción; los actores son aquellos que interactúan con el sistema. Las precondiciones son los hechos que se han de cumplir para que el flujo de evento se pueda llevar a cabo. Luego tenemos el flujo de eventos, que corresponde a la ejecución normal y exitosa del caso de uso. Los flujos alternativos son los que nos permiten indicar qué es lo que hace el sistema en los casos menos frecuentes e inesperados. Las Pos condiciones son los hechos que se ha de cumplir si el flujo de eventos normal se ha ejecutado correctamente. Por último, las excepciones es aquel flujo que nos permite controlar los errores que llegasen a ocurrir durante la operación.

A continuación se presentan una serie de figuras que describen el desarrollo de los casos de uso y la forma de interactuar entre el sistema y el usuario.

Nombre:	Crear un nuevo rechazo
Autor:	Josué Pérez
Descripción:	Permite generar un nuevo rechazo dentro del sistema
Actores:	Usuario de calidad Logeado
Precondiciones:	El usuario debe haberse Logeado en el sistema
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el menú rechazos --> nuevo. 2. El sistema muestra el formato a llenar para ingresar un nuevo rechazo mostrando un conjunto de cajas de texto, listas desplegables y cajas de texto autocompletables. 3. El actor llena el formato con los datos del rechazo. 4. El sistema comprueba la validez de los datos y los almacena. 5. El sistema envía un correo al área de compras, indicándole que se ha generado un nuevo rechazo con el folio asignado para este.
Flujo Alternativo:	El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al actor de ello permitiéndole que lo corrija.
Pos condiciones:	Se agregó el Rechazo # 0000, Se envió notificación a Compras
Excepciones:	<ol style="list-style-type: none"> 1. Si el sistema no almacena el rechazo, este almacenara el error y enviará una notificación indicando que ha surgido un error, con el código 0000 favor de notificar al área de sistemas.

Figura 2.17. Descripción del caso de uso crear un nuevo rechazo.

Nombre:	Agregar número PO
Autor:	Josué Pérez
Descripción: Permite agregar el numero de PO	
Actores: Usuario de compras logeado	
Precondiciones: El usuario debe haberse logeado en el sistema. Debe existir un rechazo en la bandeja de PO. El usuario de be haber recibido un correo indicándole que debe agregar el numero de PO al folio 0000.	
Flujo Normal: <ol style="list-style-type: none"> 1. El usuario pulsa sobre el menú seguimiento --> PO. 2. El sistema muestra todos los rechazos pendientes de ingresar el número de PO. 3. El actor pulsa sobre el botón PO del rechazo. 4. El sistema muestra una caja de texto para ingresar la información. 5. El sistema comprueba la validez de los datos y los almacena. 6. El sistema envía un correo a almacén indicándole que se ha agregado numero de PO, y que debe de bloquear el material con la siguiente información: Folio Rechazo, Cantidad, Número de Factura, Número de Parte y Número de Proveedor 	
Flujo Alternativo: El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al actor de ello permitiéndole que lo corrija.	
Pos condiciones: Se quita el rechazo de la bandeja de compras	
Excepciones: <ol style="list-style-type: none"> 1. Si el sistema no almacena el número de PO, se indica al usuario que ha surgido un error al almacenar el valor, favor de intentar más tarde 	

Figura 2.18. Descripción del caso de uso agregar número de PO.

Nombre:	Bloquear material
Autor:	Josué Pérez
Descripción: Permite bloquear el material, rechazado por calidad.	
Actores: Usuario de almacén Logeado	
Precondiciones: El usuario debe haberse logeado en el sistema. Debe existir un rechazo pendiente por bloquear material. El usuario recibe un correo indicándole que debe bloquear el material del numero de rechazo 0000.	
Flujo Normal: <ol style="list-style-type: none"> 1. El usuario pulsa sobre el menú seguimiento --> Almacén. 2. El sistema muestra todos los rechazos pendientes por bloquear o desbloquear. 3. El actor pulsa sobre el botón bloquear del rechazo. 4. El sistema bloquea las piezas indicadas. 5. El sistema envía un correo a compras indicándole que valide si esta de acuerdo con el rechazo 	
Flujo Alternativo: El sistema comprueba la validez de los datos, si los datos no son correctos, se avisa al actor de ello permitiéndole que lo corrija.	
Pos condiciones: Se quita el rechazo de la bandeja de calidad	
Excepciones: <ol style="list-style-type: none"> 1. Si el sistema no almacena la respuesta, se indica al usuario que ha surgido un error al almacenar el valor, favor de intentar más tarde 	

Figura 2.19. Descripción del caso de uso bloquear material.

Nombre:	Consultar si el proveedor esta de acuerdo con el rechazo
Autor:	Josué Pérez
Descripción:	Permite validar si el proveedor está de acuerdo con el rechazo del Material.
Actores:	Usuario de compras logeado
Precondiciones:	El usuario debe haberse logeado en el sistema Debe existir un rechazo en la bandeja de consulta proveedor. El usuario recibe un correo indicándole que él tiene un rechazo pendiente por validar si está de acuerdo el proveedor del rechazo.
Flujo Normal:	<ol style="list-style-type: none"> 1. El actor pulsa sobre el menú seguimiento --> Consulta Proveedor. 2. El sistema muestra todos los rechazos pendientes por validar si está de acuerdo el proveedor. 3. El actor pulsa sobre el icono de paloma para registrar su conformidad o el icono de tache para expresar su inconformidad. 4. El sistema almacena la información.
Flujo Alternativo:	No existe un flujo alternativo para esta acción
Pos condiciones:	Si el actor pulsa el icono de paloma se envía un correo a compras indicándole si se resorteá o retrabajá. Si el actor pulsa el icono de tache se envía un correo a calidad e ingeniería indicándole que capture si aplica o no el rechazo. Se quita el rechazo de la bandeja consulta proveedor.
Excepciones:	<ol style="list-style-type: none"> 1. Si el sistema no almacena la respuesta, se indica al usuario que ha surgido un error al almacenar el valor, favor de intentar más tarde

Figura 2.20. Descripción del caso de uso consultar si el proveedor está de acuerdo con el rechazo.

Nombre:	Indicar si se resorteá o retrabajá el material
Autor:	Josué Pérez
Descripción:	Permite indicar la disposición del material.
Actores:	Usuario de compras Logeado
Precondiciones:	El usuario debe haberse logeado en el sistema Debe existir un rechazo en la bandeja de consulta proveedor. El usuario recibe un correo indicándole que él tiene un rechazo por indicar disposición
Flujo Normal:	<ol style="list-style-type: none"> 1. El actor pulsa sobre el menú seguimiento --> Sortea o Retrabaja. 2. El sistema muestra todos los rechazos pendientes para indicar disposición. 3. El actor pulsa sobre el icono de paloma para registrar su conformidad o el icono de tache para expresar su inconformidad. 4. El sistema almacena la información.
Flujo Alternativo:	No existe un flujo alternativo para esta acción
Pos condiciones:	Si el actor pulsa el icono de paloma se envía un correo a compras indicándole si se resorteá o retrabajá. Si el actor pulsa el icono de tache se envía un correo a calidad e ingeniería indicándole que capture si aplica o no el rechazo.
Excepciones:	<ol style="list-style-type: none"> 1. Si el sistema no almacena la respuesta, se indica al usuario que ha surgido un error al almacenar el valor, favor de intentar más tarde

Figura 2.21. Descripción del caso de uso indicar si se resorteá o retrabajá.

Nombre:	Agregar hoja de aceptado
Autor:	Josué Pérez
Descripción: Permite agregar las hoja(s) de aceptado.	
Actores: Usuario de calidad logeado	
Precondiciones: El usuario debe haberse logeado en el sistema Debe existir un rechazo en la bandeja de sortea o retrabajá. El usuario recibe un correo indicándole que él tiene un rechazo por agregar hojas de aceptado	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor pulsa sobre el menú seguimiento --> Sortea o Retrabaja. 2. El sistema muestra todos los rechazos pendientes para agregar hojas de aceptado. 3. El actor pulsa sobre el botón # Piezas. 4. Se muestra el formato a llenar, mostrando fecha, proveedor, número de parte, factura y folio afectado y habilita cajas de texto para ingresar cantidad aceptada y cantidad rechazada, además se habilitan combos para indicar el responsable de calidad y bodega. 5. Se habilita checkbox que indica si la hoja de aceptado es parcial o no. 6. El sistema valida y almacena la información. 	
Flujo Alternativo: Si el actor indica hoja de aceptado parcial, permite ingresar una nueva hoja de lo contrario valida y almacena información	
Pos condiciones: Se envía email a almacén, indicándole el número de piezas aceptadas. Se envía email a compras, indicándole el número de piezas rechazadas y solicitando la hoja de SCRAP y/o NC	
Excepciones: <ol style="list-style-type: none"> 1. Si el sistema no almacena las hojas de aceptado, se indica al usuario que ha surgido un error al almacenar los datos, favor de proporcionar el código de error 0000 al área de sistemas. 	

Figura 2.22. Descripción del caso de uso agregar hoja de aceptado.

Nombre:	Desbloquear material
Autor:	Josué Pérez
Descripción: Permite desbloquear el material, aceptado por calidad.	
Actores: Usuario de almacén Logeado	
Precondiciones: El usuario debe haberse logeado en el sistema. Debe existir un rechazo pendiente por desbloquear material. El usuario recibe un correo indicándole que debe desbloquear el material del numero de rechazo 0000.	
Flujo Normal: <ol style="list-style-type: none"> 1. El usuario pulsa sobre el menú seguimiento --> Almacén. 2. El sistema muestra todos los rechazos pendientes por bloquear o desbloquear. 3. El actor pulsa sobre el botón desbloquear del rechazo. 4. El sistema desbloquea las piezas indicadas. 	
Flujo Alternativo: El sistema comprueba que no existan más desbloques de material del folio o rechazos pendientes de lo contrario cierra el folio y envía correo a las áreas involucradas de que el rechazo ha finalizado.	
Pos condiciones: Indica al actor el número de piezas aceptadas y rechazadas. Se quita el rechazo de la bandeja de almacén.	
Excepciones: <ol style="list-style-type: none"> 1. Si el sistema no almacena la respuesta, se indica al usuario que ha surgido un error al almacenar el valor, favor de intentar mas tarde. 	

Figura 2.23. Descripción del caso de uso desbloquear material.

Nombre:	Agregar folio Good Receipt.
Autor:	Josué Pérez
Descripción:	
Permite agregar el folio Good Receipt	
Actores:	
Usuario de almacén logeado	
Precondiciones:	
El usuario debe haberse logeado en el sistema. Debe existir una hoja de aceptado con material rechazado. El usuario recibe un correo indicándole que existe material rechazado para el folio 0000.	
Flujo Normal:	
<ol style="list-style-type: none"> 1. El usuario pulsa sobre el menú seguimiento --> Sortea o retrabajá. 2. El sistema muestra todos los rechazos pendientes por agregar folio Good Receipt. 3. El actor pulsa sobre el botón folio del rechazo. 4. El sistema muestra la fecha y una caja de texto para proporcionar el folio. 5. El sistema muestra los documentos de SCRAP y la NC. 6. El sistema comprueba la validez de los datos y los almacena. 7. Envía email con la información que hace falta para el folio. 	
Flujo Alternativo:	
El sistema comprueba que no exista el folio Good Receipt, de lo contrario indicara la información faltante y enviará correo a las áreas involucradas.	
Pos condiciones:	
Indica al actor los documentos o folios faltantes. Se quita el rechazo de la bandeja de sortea o retrabajá.	
Excepciones:	
<ol style="list-style-type: none"> 1. Si el sistema no almacena la respuesta, se indica al usuario que ha surgido un error al almacenar el valor, favor de intentar mas tarde. 	

Figura 2.24. Descripción del caso de uso agregar folio Good Receipt.

Nombre:	Agregar folio Invoice Receipt.
Autor:	Josué Pérez
Descripción: Permite agregar el folio Invoice Receipt.	
Actores: Usuario de finanzas Logeado	
Precondiciones: El usuario debe haberse logeado en el sistema. Debe existir una hoja de aceptado con material rechazado. El usuario recibe un correo indicándole que existe material rechazado para el folio 0000.	
Flujo Normal: <ol style="list-style-type: none"> 1. El usuario pulsa sobre el menú seguimiento --> Sortea o retrabajá 2. El sistema muestra todos los rechazos pendientes por agregar folio Invoice Receipt. 3. El actor pulsa sobre el botón folio del rechazo. 4. El sistema muestra la fecha y una caja de texto para proporcionar el folio. 5. El sistema muestra los documentos de SCRAP y la NC 6. El sistema comprueba la validez de los datos y los almacena. 7. Envía email con la información que hace falta para el folio. 	
Flujo Alternativo: El sistema comprueba que no exista el folio Invoice Receipt, de lo contrario indicara la información faltante y enviará correo a las áreas involucradas.	
Pos condiciones: Indica al actor los documentos o folios faltantes. Se quita el rechazo de la bandeja de sortea o retrabajá.	
Excepciones: <ol style="list-style-type: none"> 1. Si el sistema no almacena la respuesta, se indica al usuario que ha surgido un error al almacenar el valor, favor de intentar mas tarde. 	

Figura 2.25. Descripción del caso de uso agregar folio Invoice Receipt.

Nombre:	Proveer NC y/o SCRAP.
Autor:	Josué Pérez
Descripción: Permite agregar los documentos de la NC y/o SCRAP	
Actores: Usuario de compras logeado	
Precondiciones: El usuario debe haberse logeado en el sistema. Debe existir una hoja de aceptado con material rechazado. El usuario recibe un correo indicándole que existe material rechazado para el folio 0000 o que se encuentra pendiente la carga de NC.	
Flujo Normal: <ol style="list-style-type: none"> 1. El usuario pulsa sobre el menú seguimiento --> Sorteá o retrabajá 2. El sistema muestra todos los rechazos pendientes por proveer NC. 3. El actor pulsa sobre el botón proveer NC del rechazo. 4. El sistema muestra un selector de archivos y dos botones para agregar o quitar documentos. 5. El sistema comprueba la validez de los datos y los almacena. 6. El sistema muestra los documentos de SCRAP y la NC 7. Envía email con la información que hace falta para el folio. 	
Flujo Alternativo: El sistema comprueba que no exista el folio Invoice Receipt, de lo contrario indicara la información faltante y enviará correo a las áreas involucradas.	
Pos condiciones: Indica al actor los documentos o folios faltantes. Se envía email a IMPO\EXPO Se quita el rechazo de la bandeja de sorteá o retrabajá.	
Excepciones: <ol style="list-style-type: none"> 1. Si el sistema no almacena el documento, se indica al usuario que ha surgido un error al almacenar el valor, favor de intentar mas tarde. 	

Figura 2.26. Descripción del caso de uso proveer NC y/o SCRAP.

Nombre:	Agregar número de pedimento y documentos
Autor:	Josué Pérez
Descripción:	Permite agregar el numero de pedimento y los documentos
Actores:	Usuario de IMPO/EXPO Logeado
Precondiciones:	<p>El usuario debe haberse logeado en el sistema. El usuario recibe un correo indicándole que existe folio pendiente por agregar numero de pedimento y documentos.</p>
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el menú seguimiento --> Regresa a Origen 2. El sistema muestra todos los rechazos pendientes por proveer numero de Pedimento. 3. El actor pulsa sobre el botón # de Pedimento del rechazo. 4. El sistema muestra una caja de texto para agregar el numero de pedimento y un selector de archivos con dos botones para agregar o quitar documentos. 5. El sistema comprueba la validez de los datos y los almacena. 6. El sistema muestra los documentos asignados. 7. Envía email con la información que hace falta para el folio.
Flujo Alternativo:	<p>El sistema comprueba que no exista el numero de pedimento, de lo contrario indicara que la información se ha agregado.</p>
Pos condiciones:	<p>Se envía email a almacén para indicar el retorno del material Se quita el rechazo de la bandeja de sortea o retrabajá.</p>
Excepciones:	<ol style="list-style-type: none"> 1. Si el sistema no almacena la respuesta, se indica al usuario que ha surgido un error al almacenar el valor, favor de intentar mas tarde. 2. Si el sistema no almacena el documento, se indica al usuario que ha surgido un error al almacenar el valor, favor de intentar mas tarde.

Figura 2.27. Descripción del caso de uso agregar número de pedimento y documentos.

Nombre:	Validar si aplica el rechazo
Autor:	Josué Pérez
Descripción: Permite verificar si el rechazo procede o no	
Actores: Usuario de Calidad o Ingeniería logeado	
Precondiciones: El usuario debe haberse logeado en el sistema. El usuario recibe un correo indicándole que existe folio pendiente por validar si aplica o no el rechazo.	
Flujo Normal: <ol style="list-style-type: none"> 1. El usuario pulsa sobre el menú seguimiento --> Aplica Rechazo 2. El sistema muestra todos los rechazos pendientes por proveer si aplica el rechazo. 3. El actor pulsa sobre el link aplica del rechazo. 4. El sistema muestra dos botones para indicar presionar un si o No. 5. El sistema comprueba la validez de los datos y los almacena. 6. Envía email con la información que hace falta para el folio. 	
Flujo Alternativo: El sistema valida que calidad o ingeniería hayan registrado su respuesta de lo contrario espera la respuesta de ambos.	
Pos condiciones: Si aplica el rechazo se envía correo a compras para agregar hojas para descontar material. Si no aplica el rechazo se envía un correo a calidad para que valide si sirve o no el material. Se quita el folio de la bandeja de aplica el rechazo.	
Excepciones: <ol style="list-style-type: none"> 1. Si el sistema no almacena la respuesta, se indica al usuario que ha surgido un error al almacenar el valor, favor de intentar mas tarde. 	

Figura 2.28. Descripción del caso de uso validar si aplica el rechazo.

Nombre:	Validar si el material sirve
Autor:	Josué Pérez
Descripción:	Permite verificar si el material sirve
Actores:	Usuario de Calidad Logeado
Precondiciones:	El usuario debe haberse logeado en el sistema. El usuario recibe un correo indicándole que existe folio pendiente por validar si el material sirve.
Flujo Normal:	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el menú seguimiento --> Material Sirve 2. El sistema muestra todos los rechazos pendientes por validar si el material sirve. 3. El actor pulsa sobre la imagen de paloma para indicar que sirve el material o la imagen de tacha para indicar que no sirve el material. 4. El sistema comprueba la validez de los datos y los almacena.
Flujo Alternativo:	Si el material no sirve espera hoja de SCRAP proporcionada por calidad.
Pos condiciones:	Si el material sirve envía email a calidad para que ingrese hoja(s) de aceptado. Se quita el folio de la bandeja de material sirve.
Excepciones:	<ol style="list-style-type: none"> 1. Si el sistema no almacena la respuesta, se indica al usuario que ha surgido un error al almacenar el valor, favor de intentar mas tarde.

Figura 2.29. Descripción del caso de uso validar si el material sirve.

2.4.2.5. Diagramas de actividades

Muestran el flujo de control entre actividades, es decir es aquel que muestra las operaciones que se pasan entre los objetos en su interacción.

Este diagrama es usado para mostrar la secuencia de actividades, ya que como comentamos muestran el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad. La versatilidad de estos diagramas nos permite detallar situaciones donde el proceso paralelo puede ocurrir en la ejecución de algunas las actividades.

En las siguientes figuras se muestran las actividades más importantes del sistema.

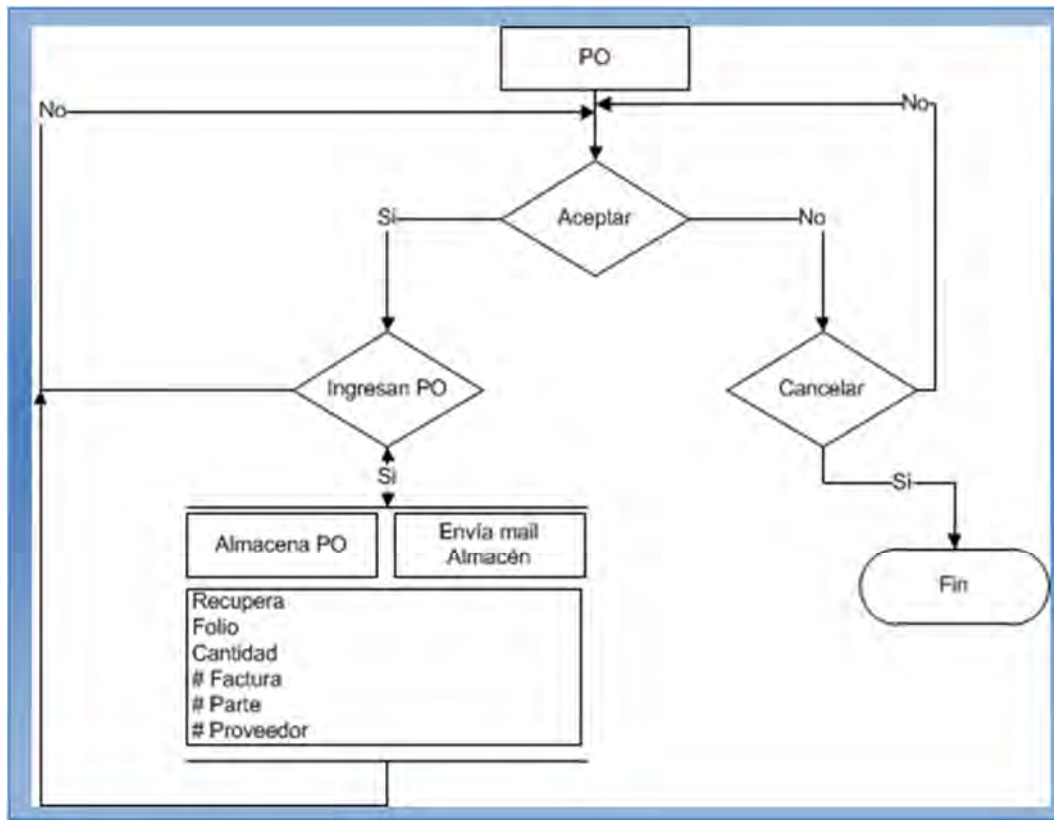


Figura 2.30. Diagrama que representa el flujo general de la pantalla PO.

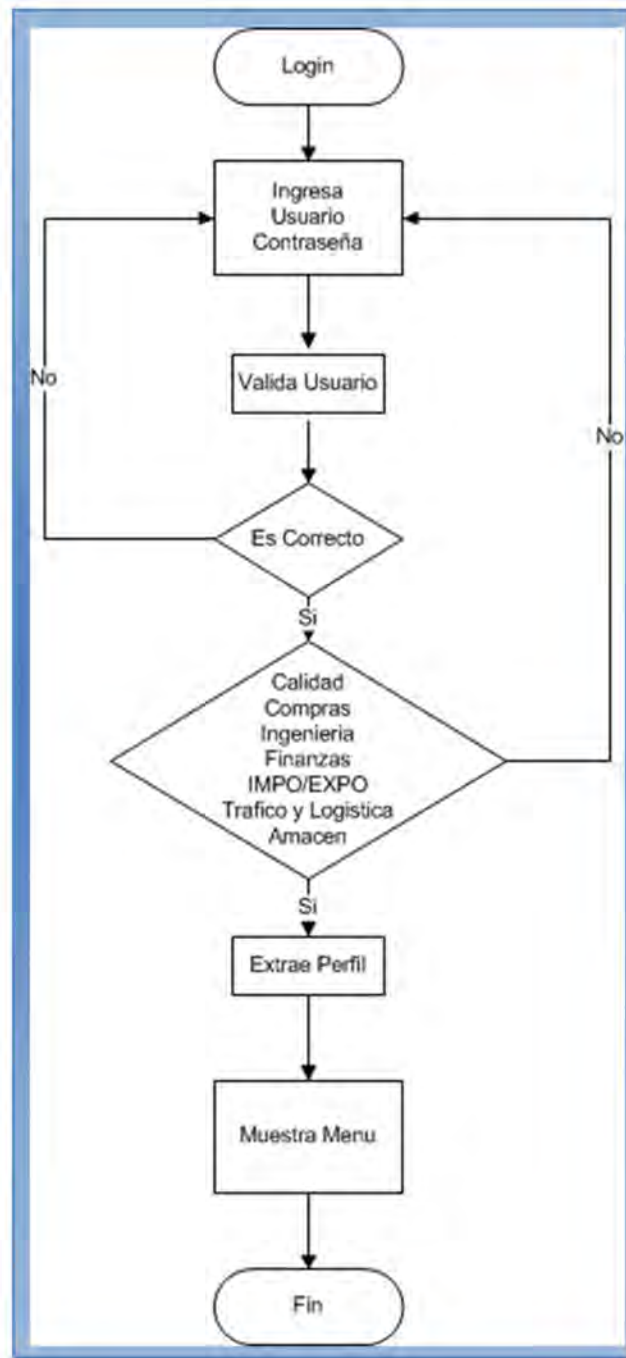


Figura 2.31. Diagrama que representa el flujo que se sigue para realizar el proceso de login.

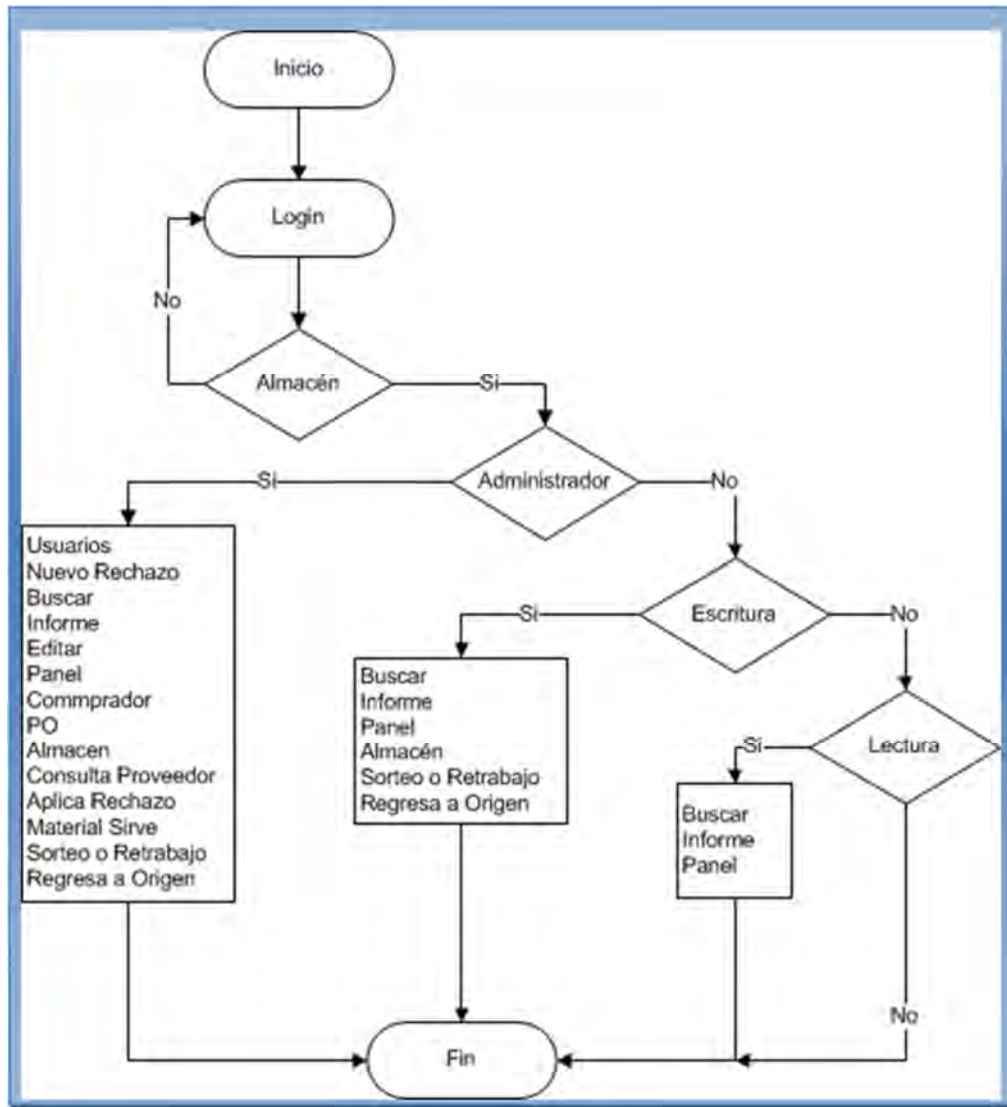


Figura 2.32. Diagrama que representa el flujo que se sigue para asignar en el menú lo que visualizará el usuario de almacén.

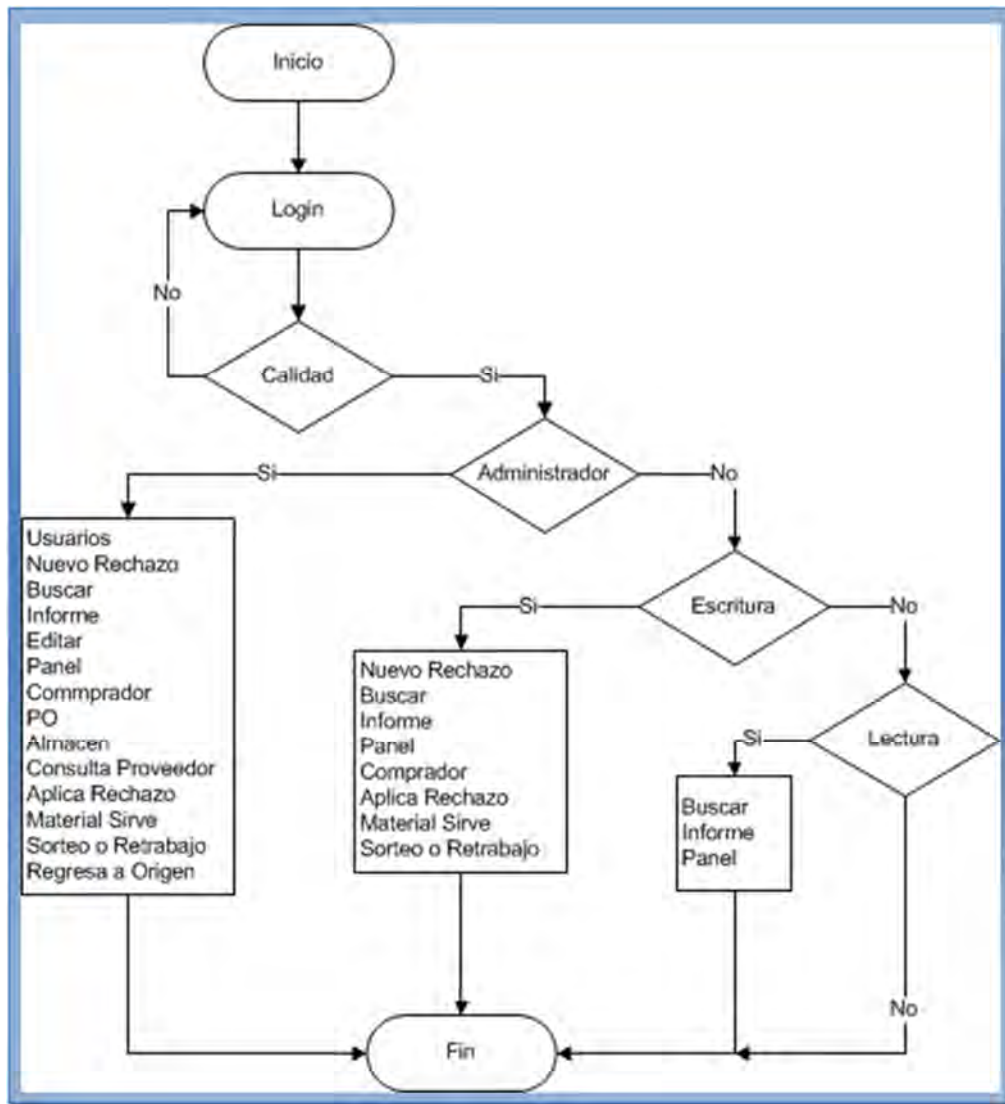


Figura 2.33. Diagrama que representa el flujo que se sigue para asignar en el menú lo que visualizará el usuario de calidad.

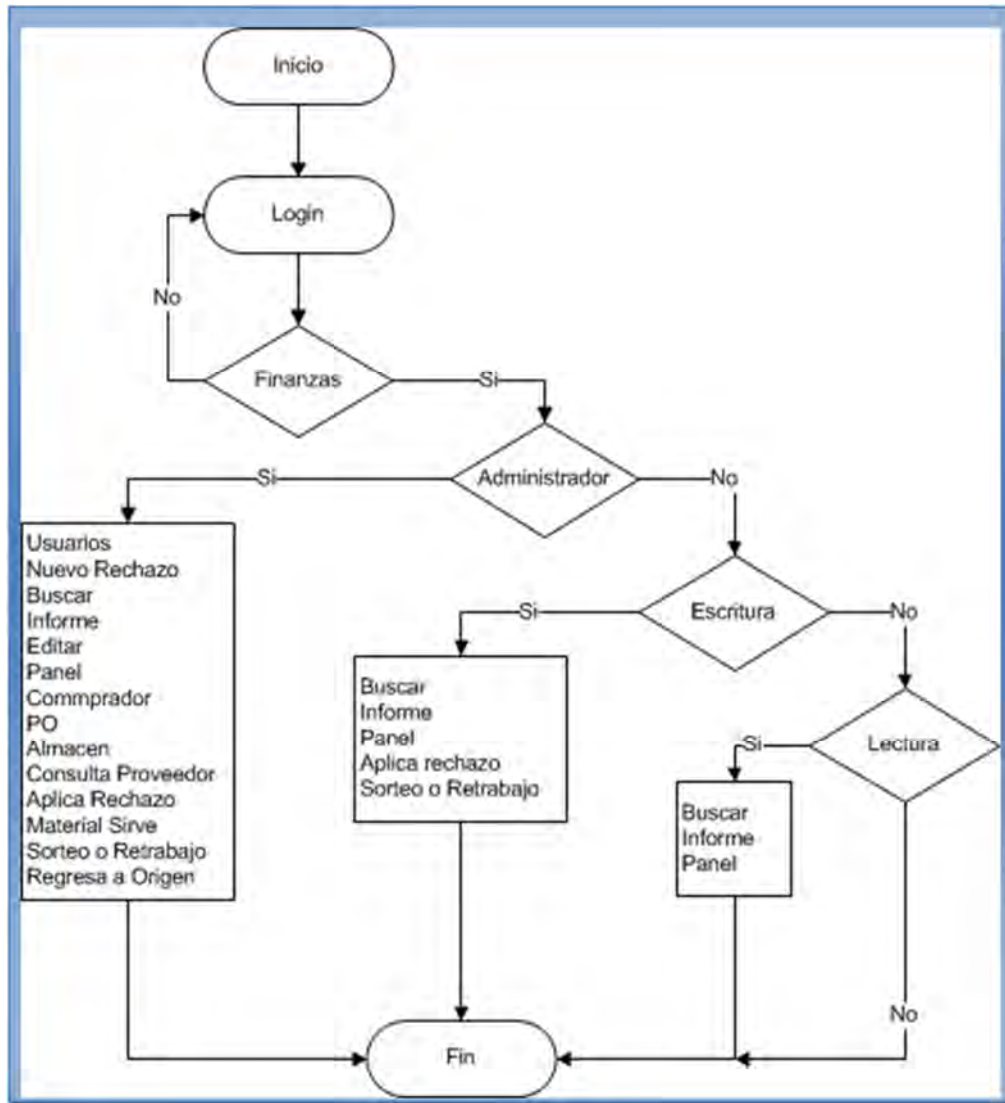


Figura 2.34. Diagrama que representa el flujo que se sigue para asignar en el menú lo que visualizará el usuario de finanzas.

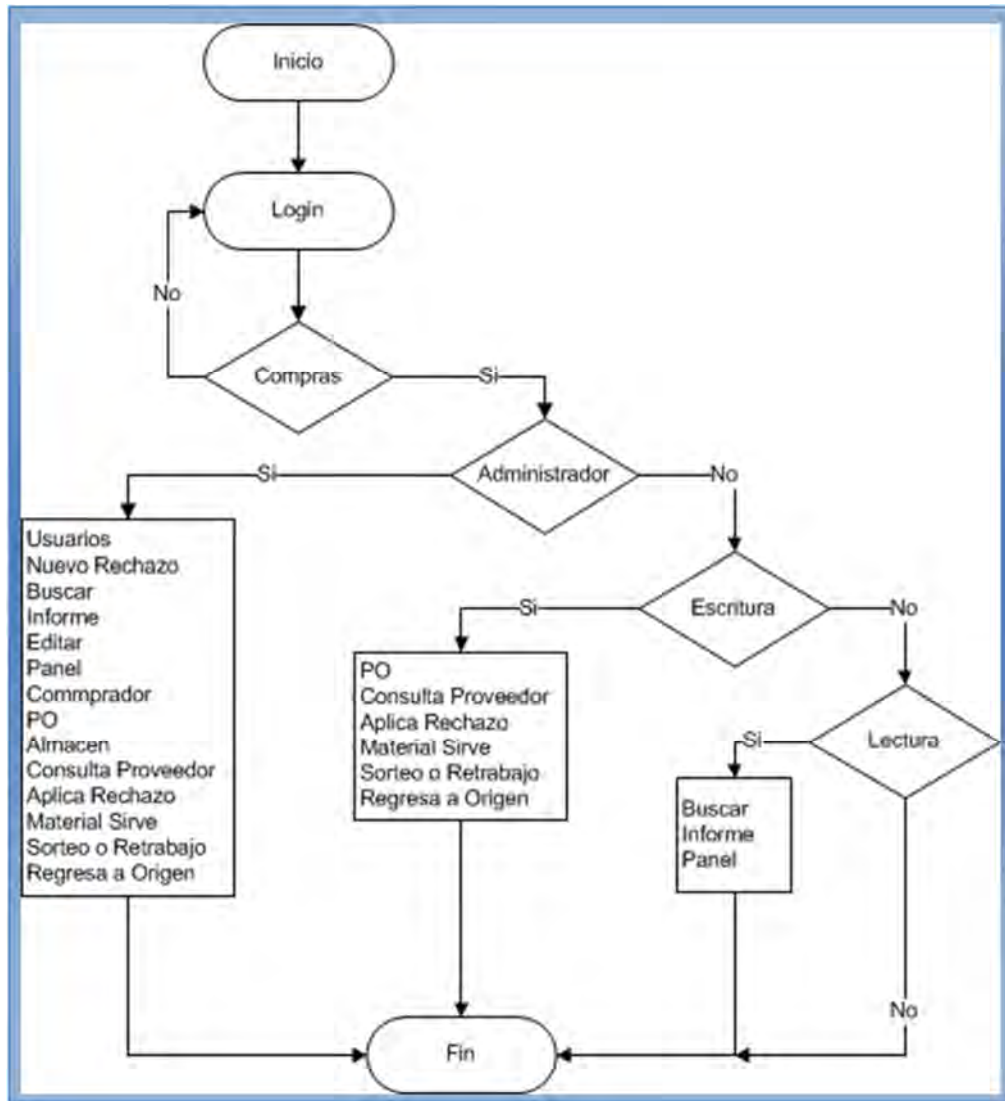


Figura 2.35. Diagrama que representa el flujo que se sigue para asignar en el menú lo que visualizará el usuario de compras.

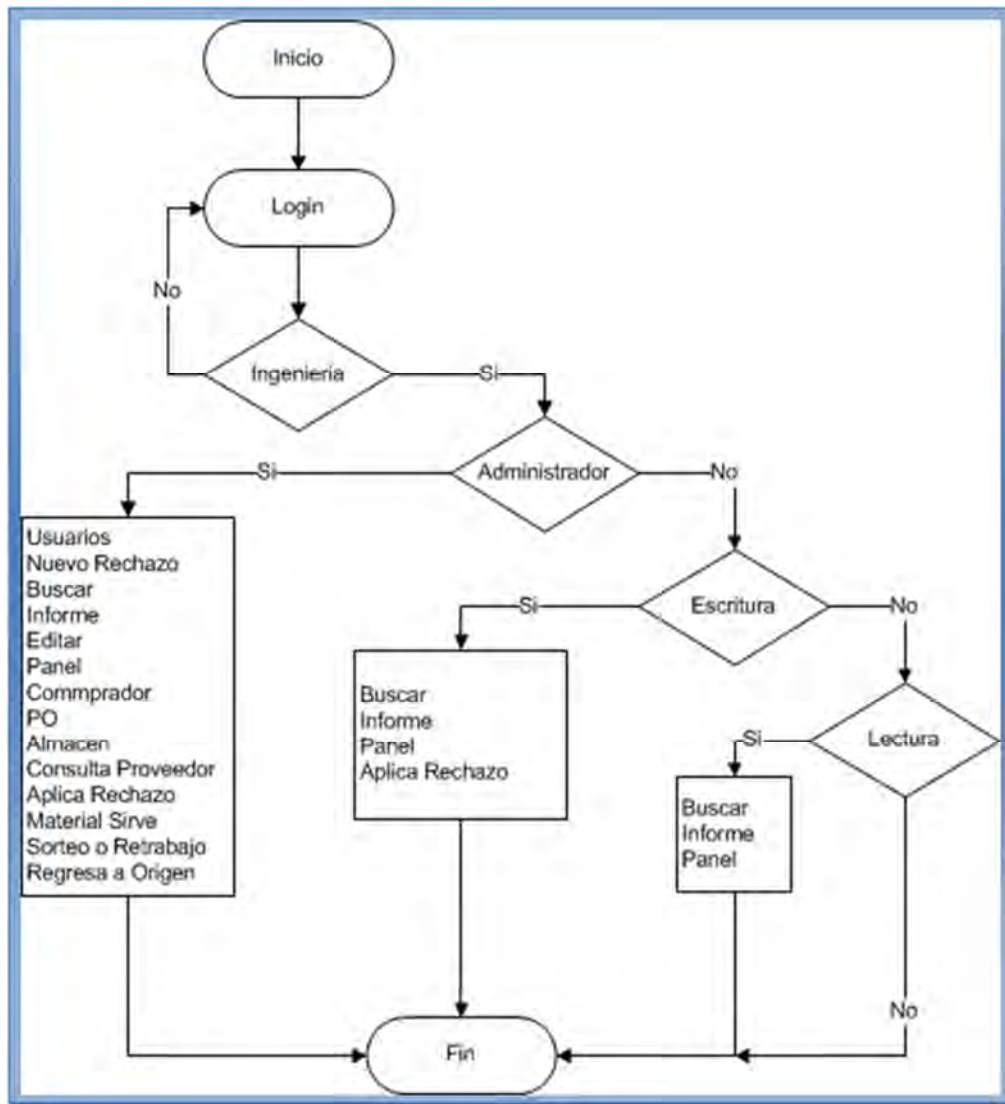


Figura 2.36. Diagrama que representa el flujo que se sigue para asignar en el menú lo que visualizará el usuario de ingeniería.

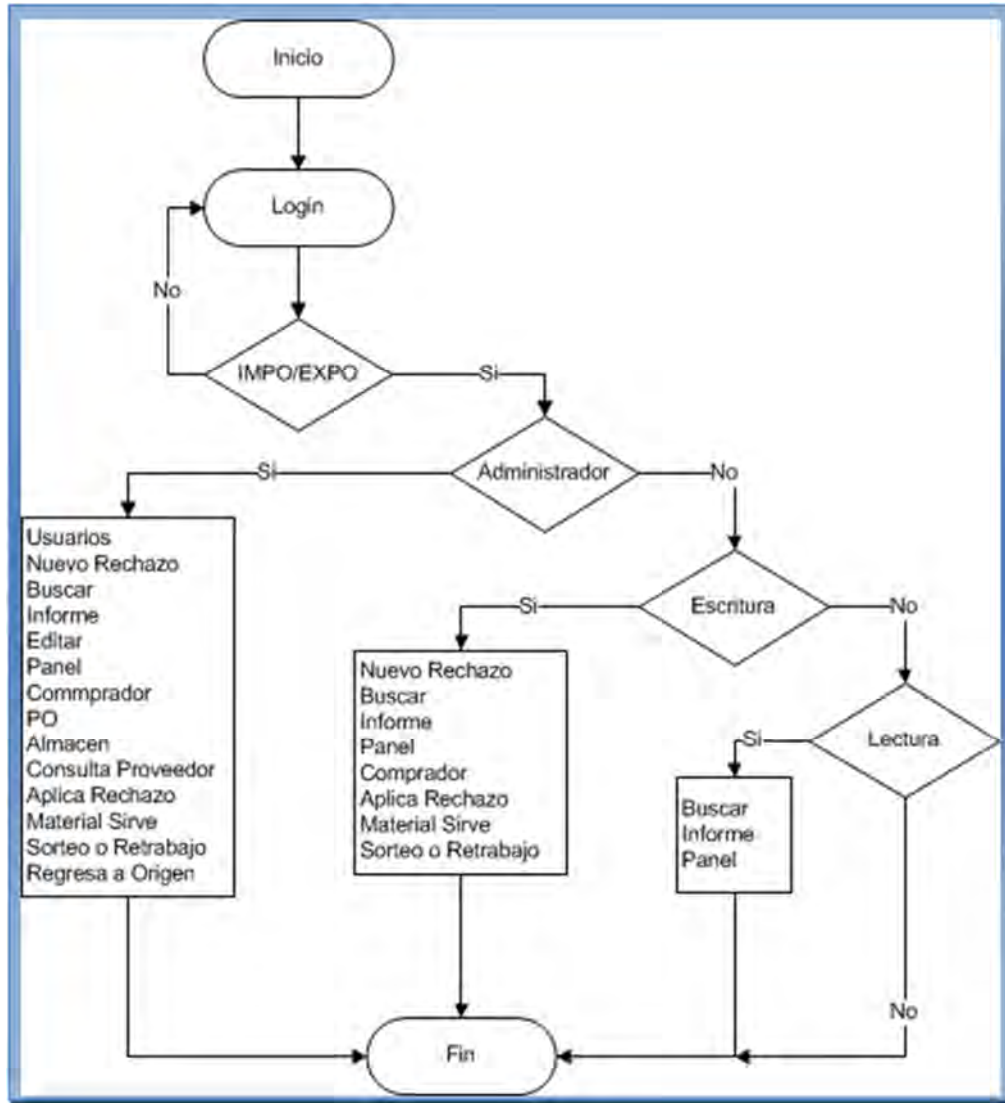


Figura 2.37. Diagrama que representa el flujo que se sigue para asignar en el menú lo que visualizará el usuario de IMPO/EXPO.

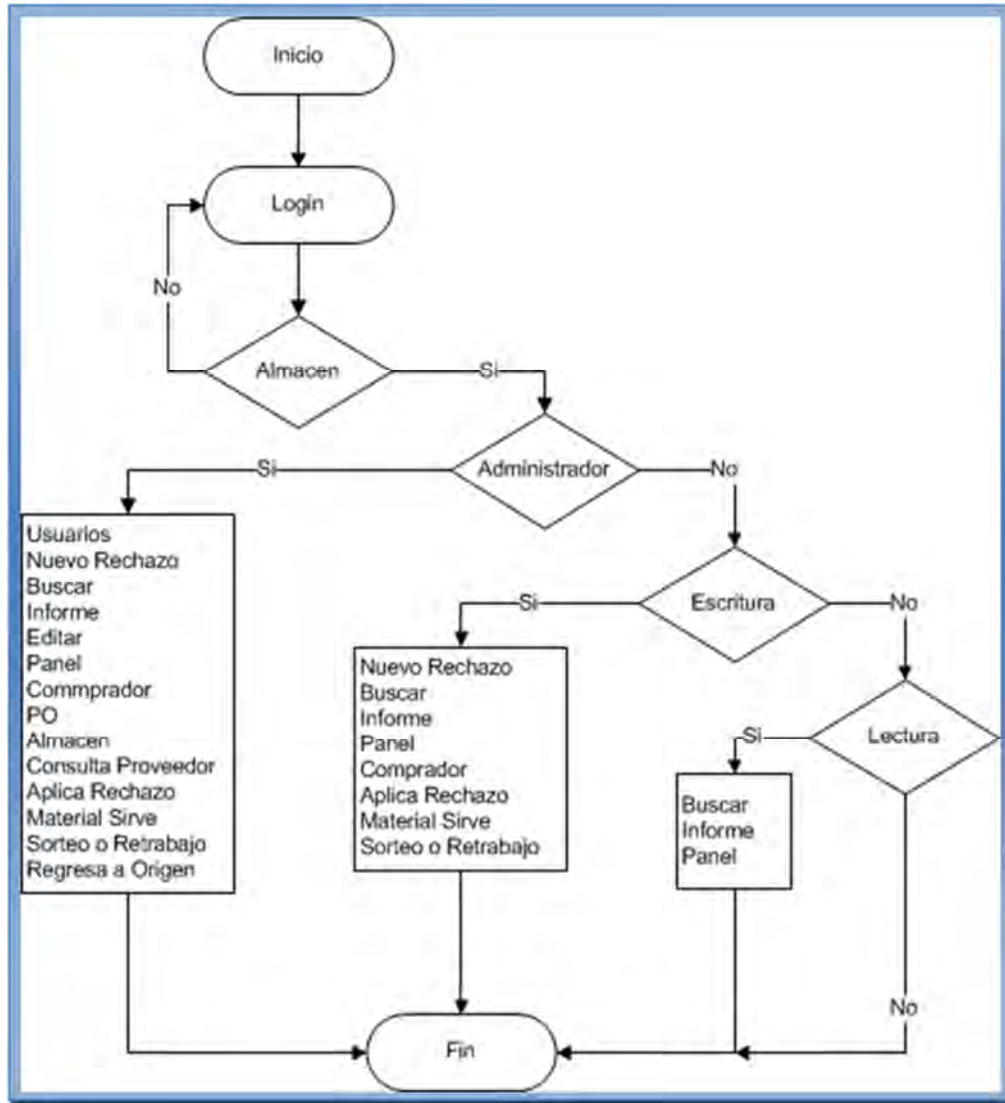


Figura 2.38. Diagrama que representa el flujo que se sigue para asignar en el menú lo que visualizará el usuario de almacén.

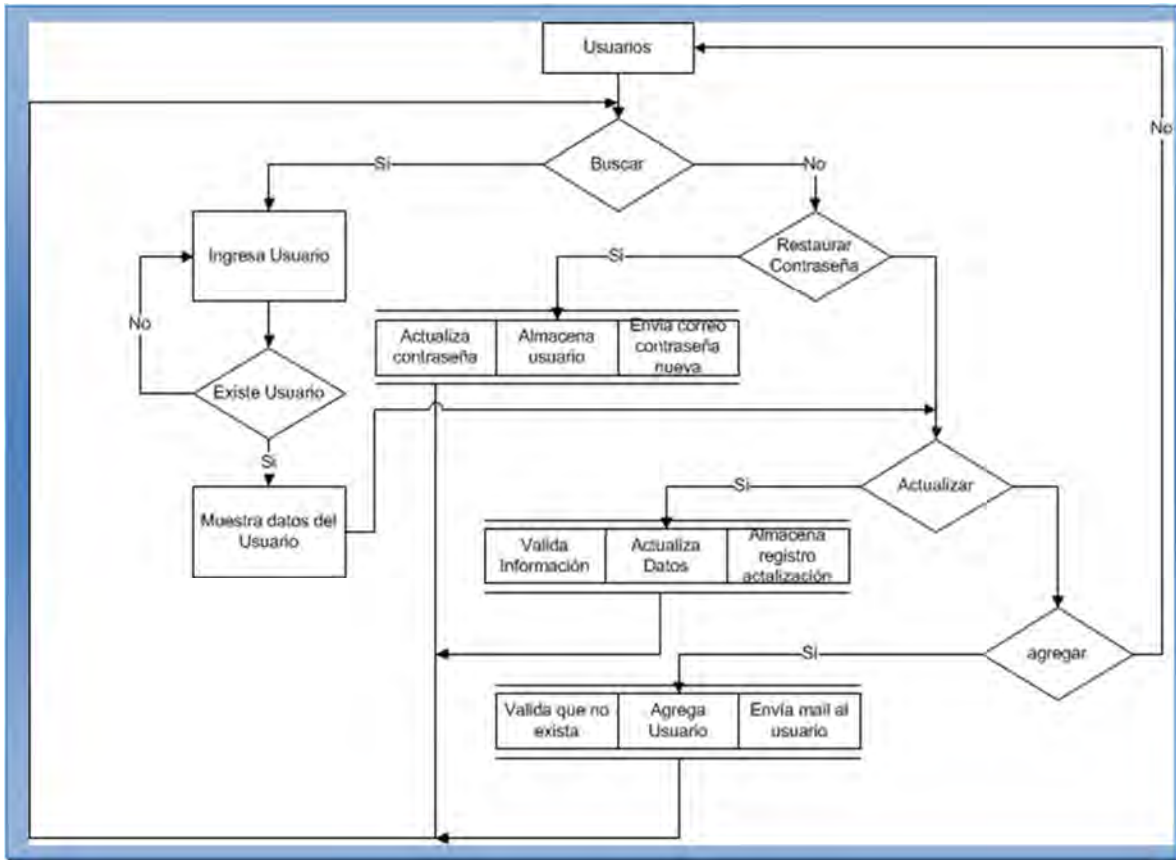


Figura 2.39. Diagrama que representa el flujo general de la pantalla de usuarios.



Figura 2.40. Diagrama que representa el flujo general de la pantalla de panel.

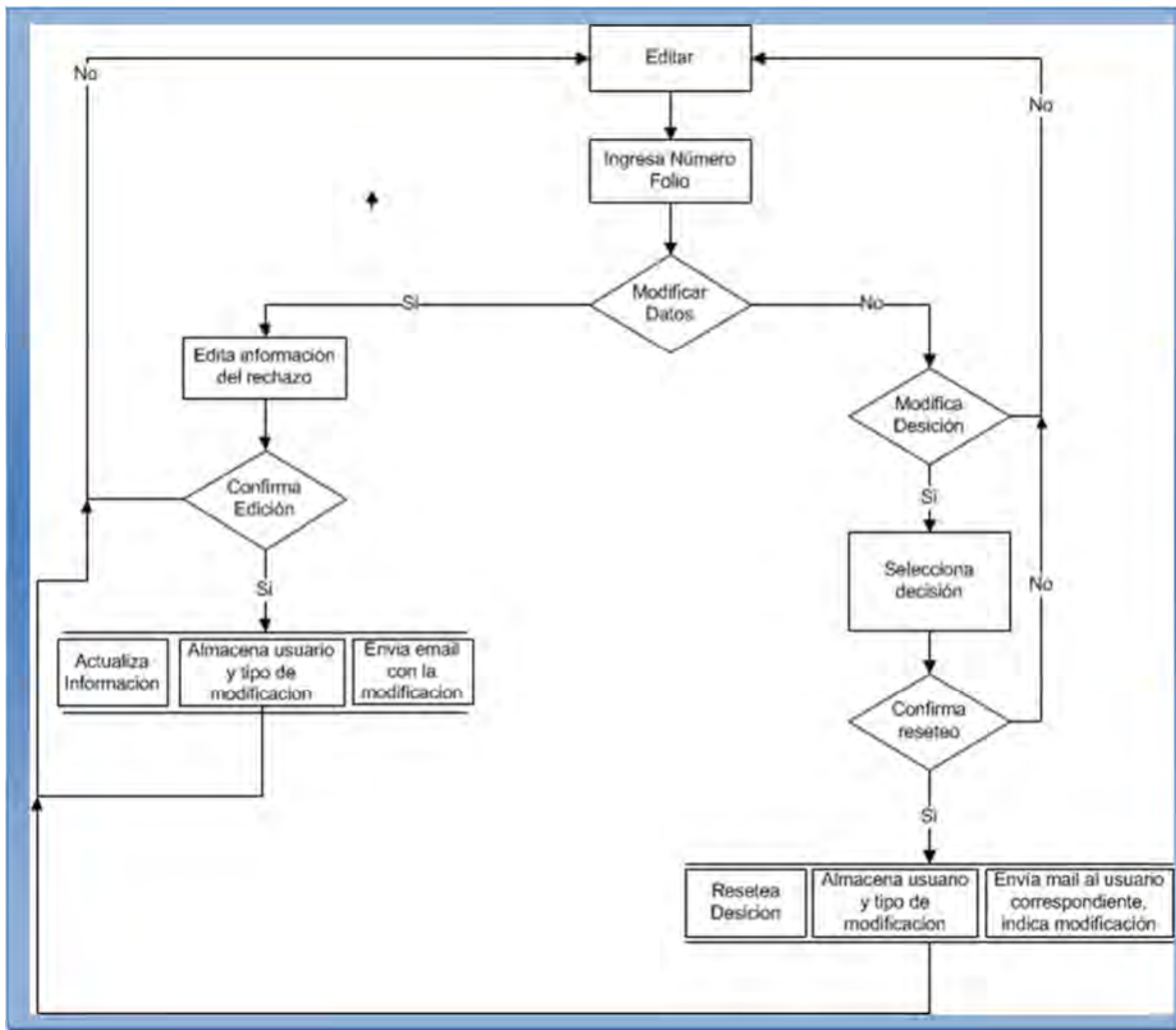


Figura 2.41. Diagrama que representa el flujo general de la pantalla editar.

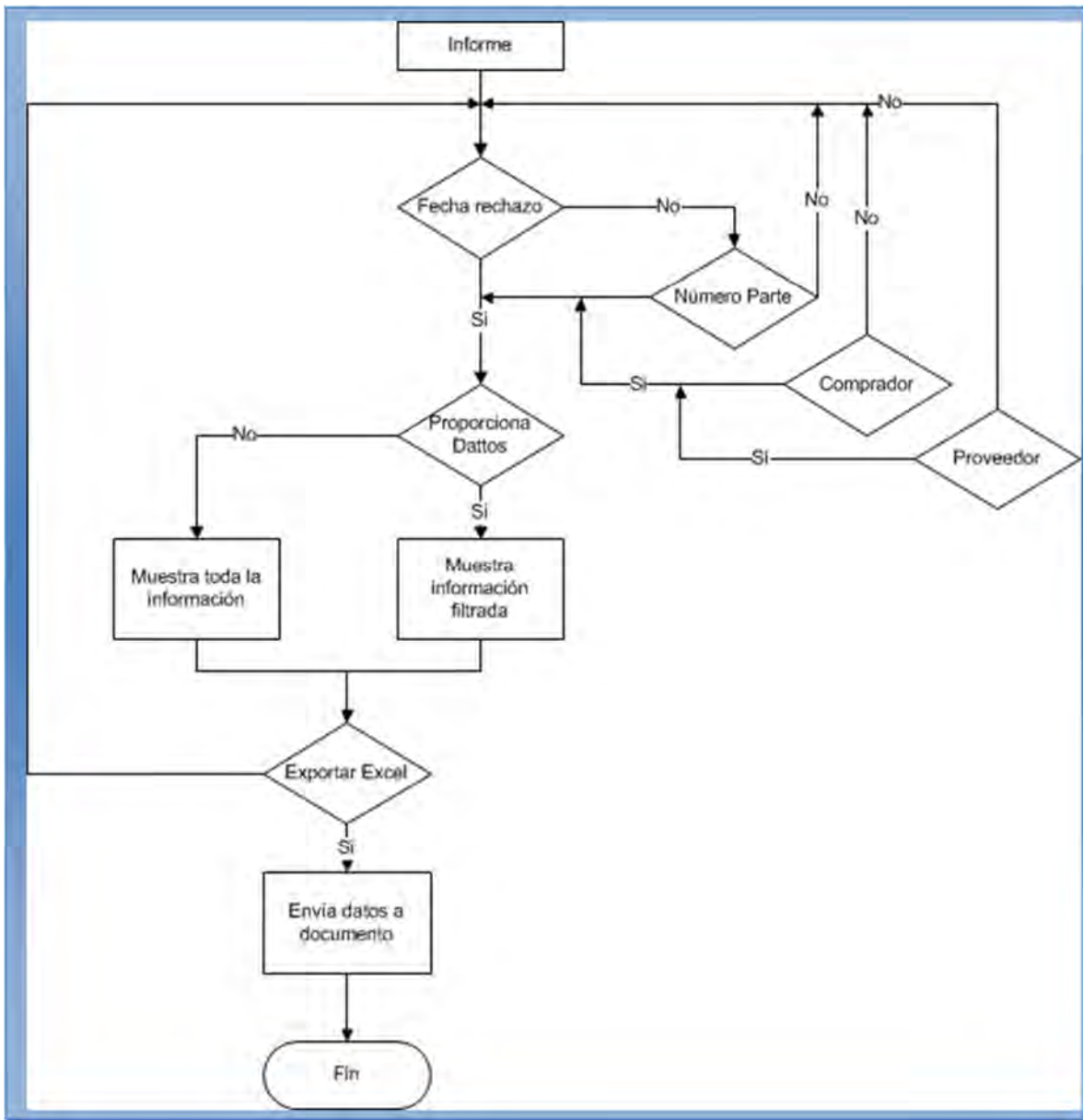


Figura 2.42. Diagrama que representa el flujo general de la pantalla del informe.

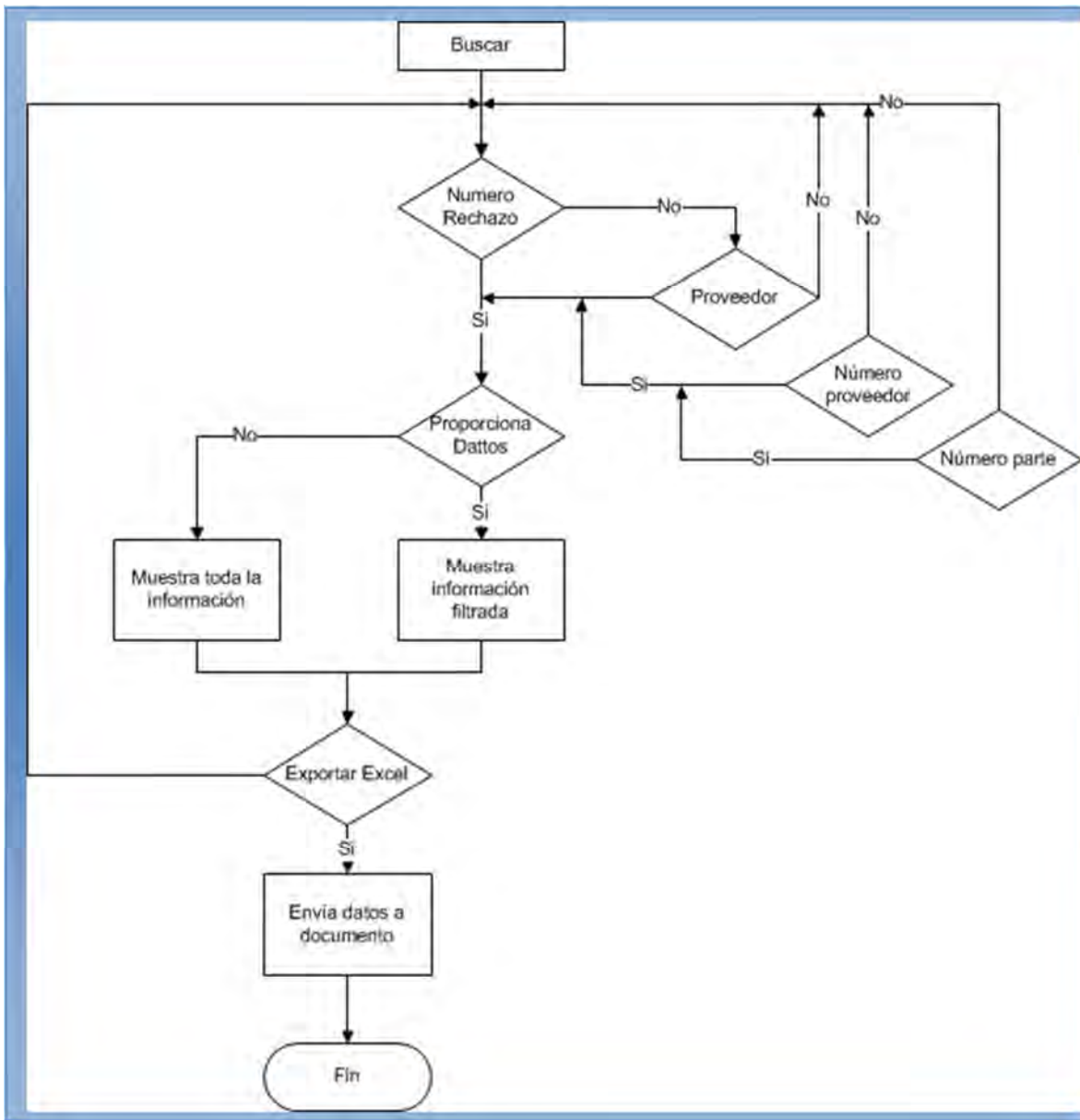


Figura 2.43. Diagrama que representa el flujo general de la pantalla de buscar.

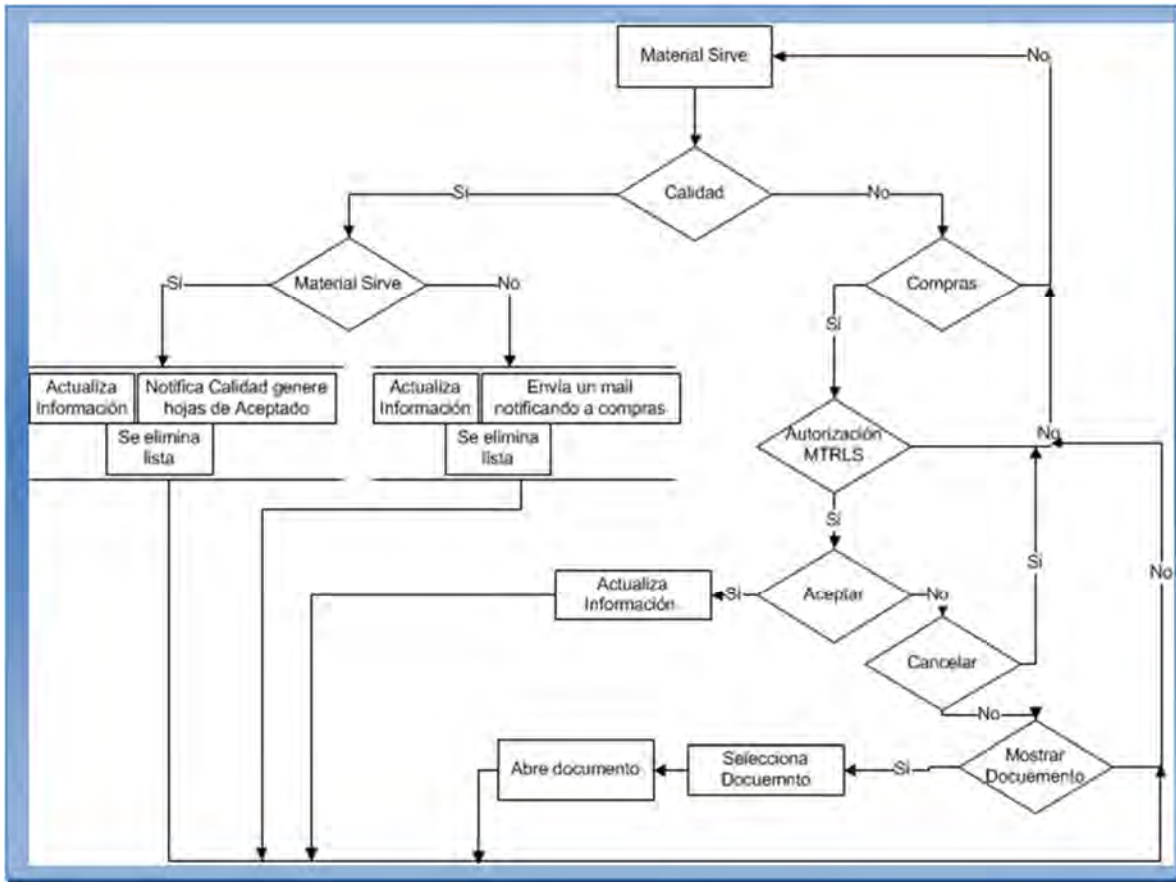


Figura 2.44. Diagrama que representa el flujo general de la pantalla material sirve.

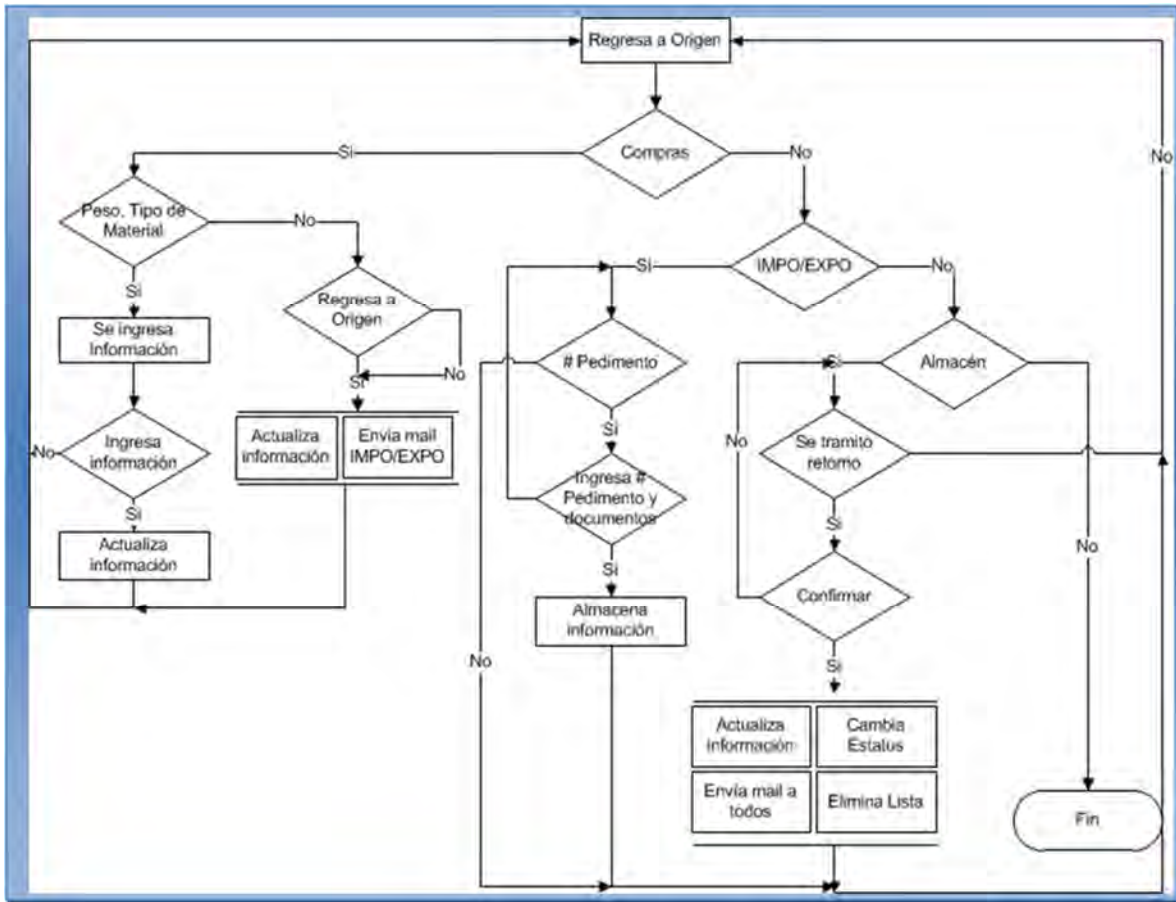


Figura 2.45. Diagrama que representa el flujo general de la pantalla regresa a origen.

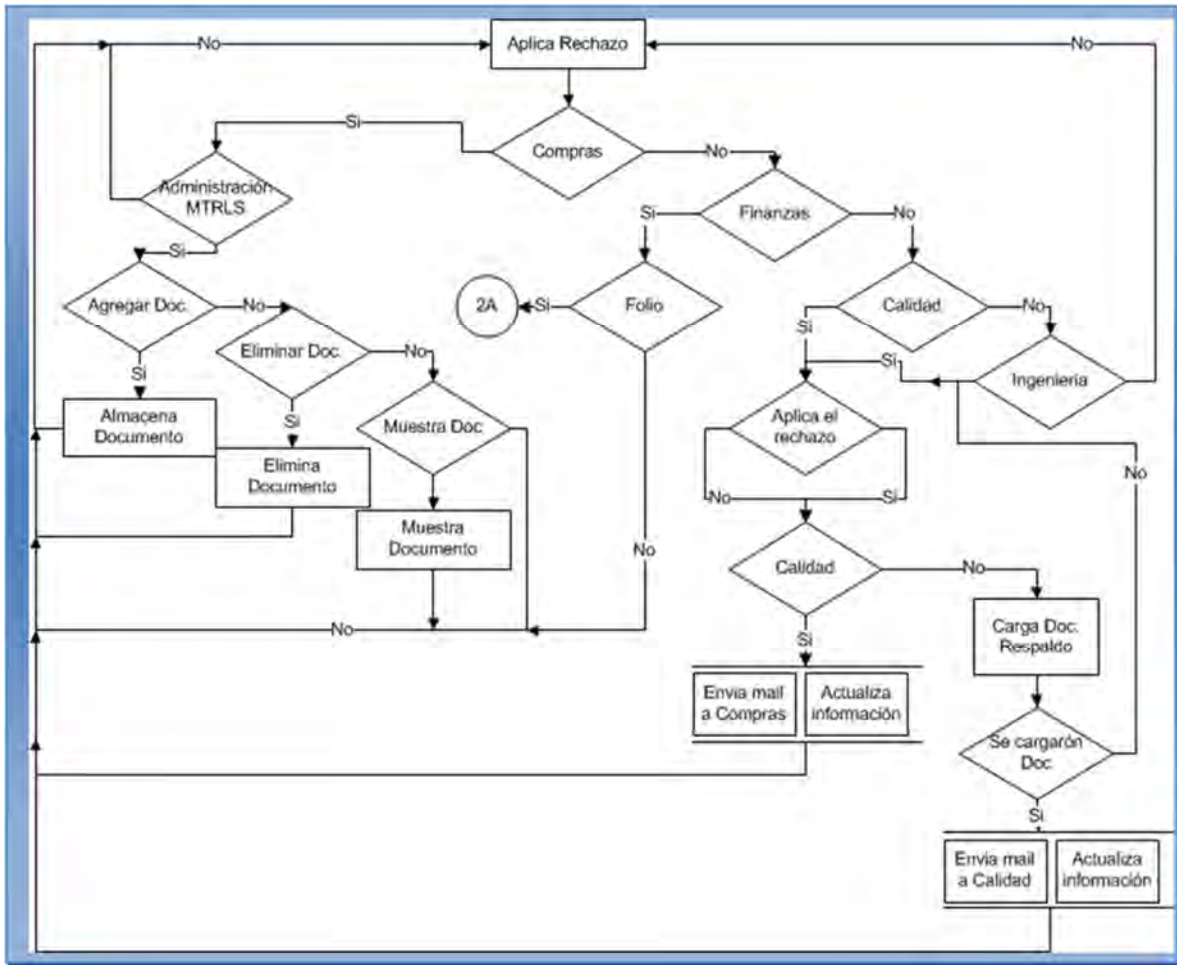


Figura 2.46. Diagrama que representa el flujo general de la pantalla aplica rechazo.

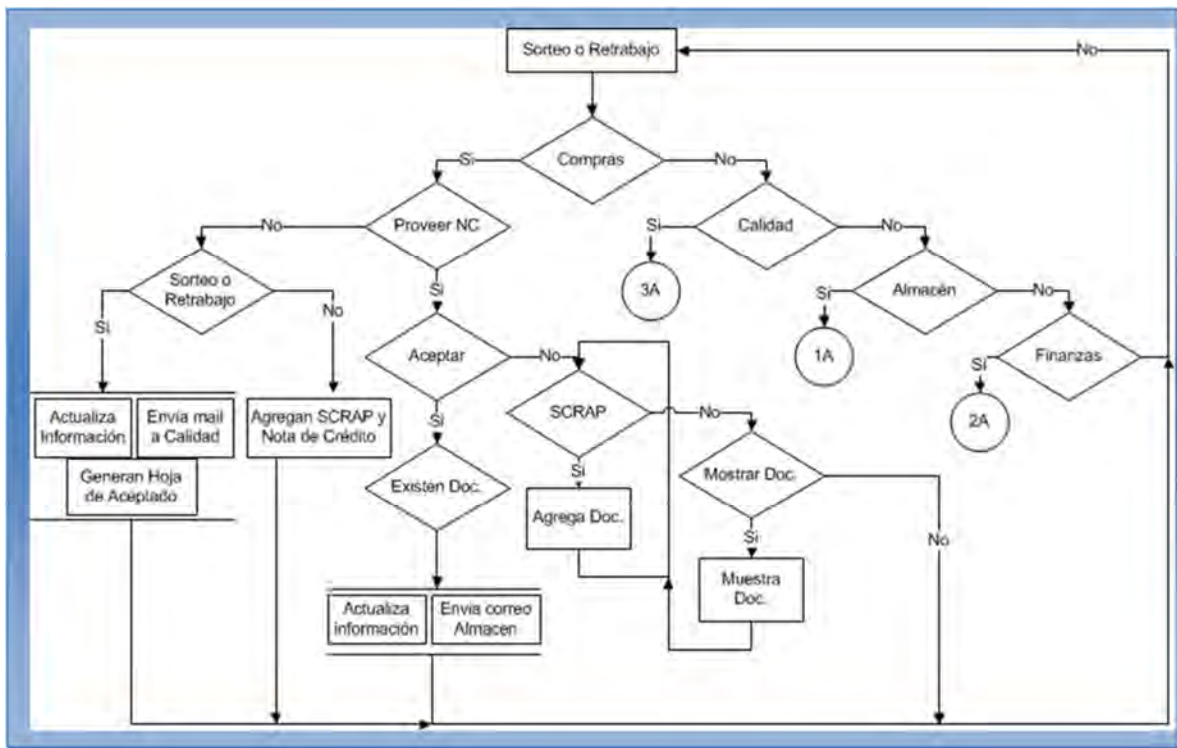


Figura 2.47. Diagrama que representa el flujo general de la pantalla sorteo o retrabajó.

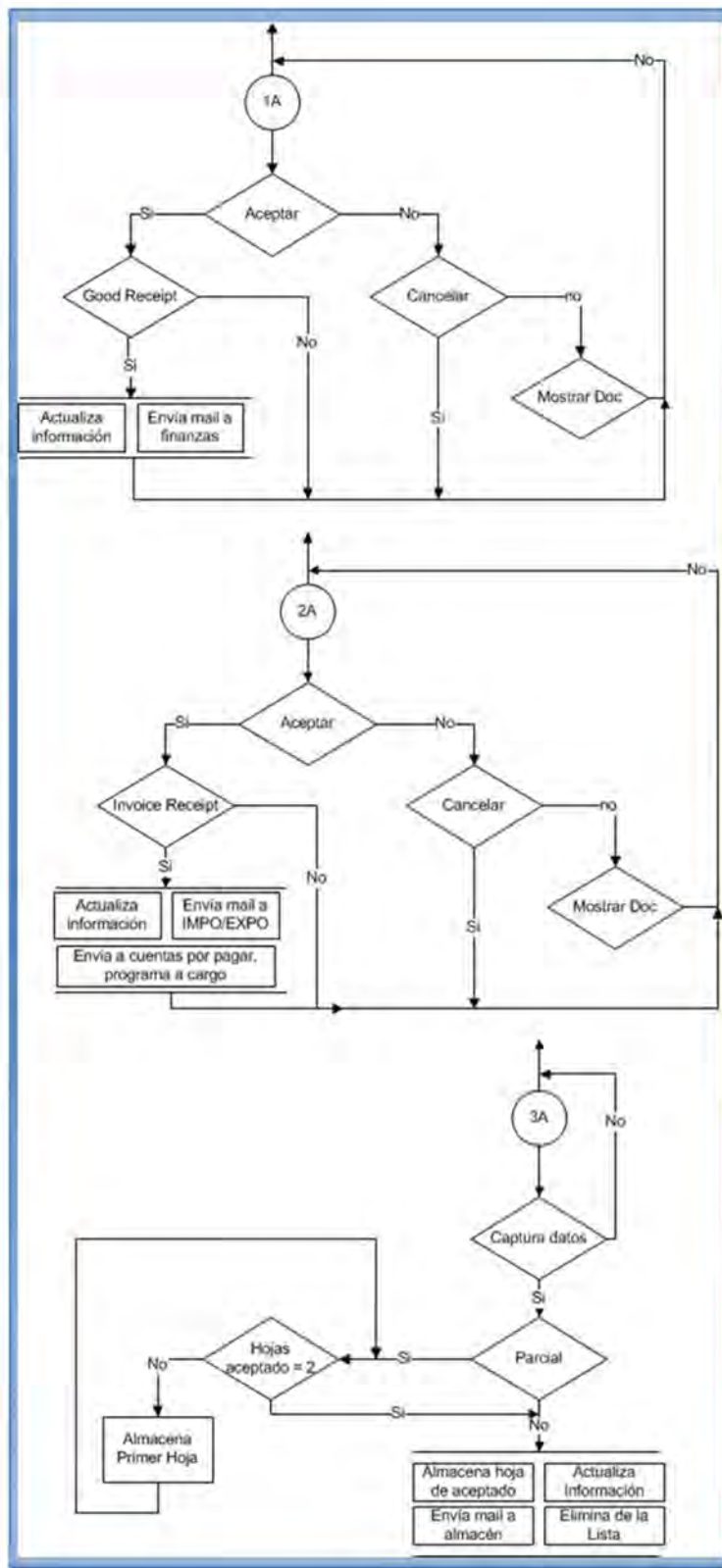


Figura 2.48. Diagrama que representa el flujo general de varios procesos.

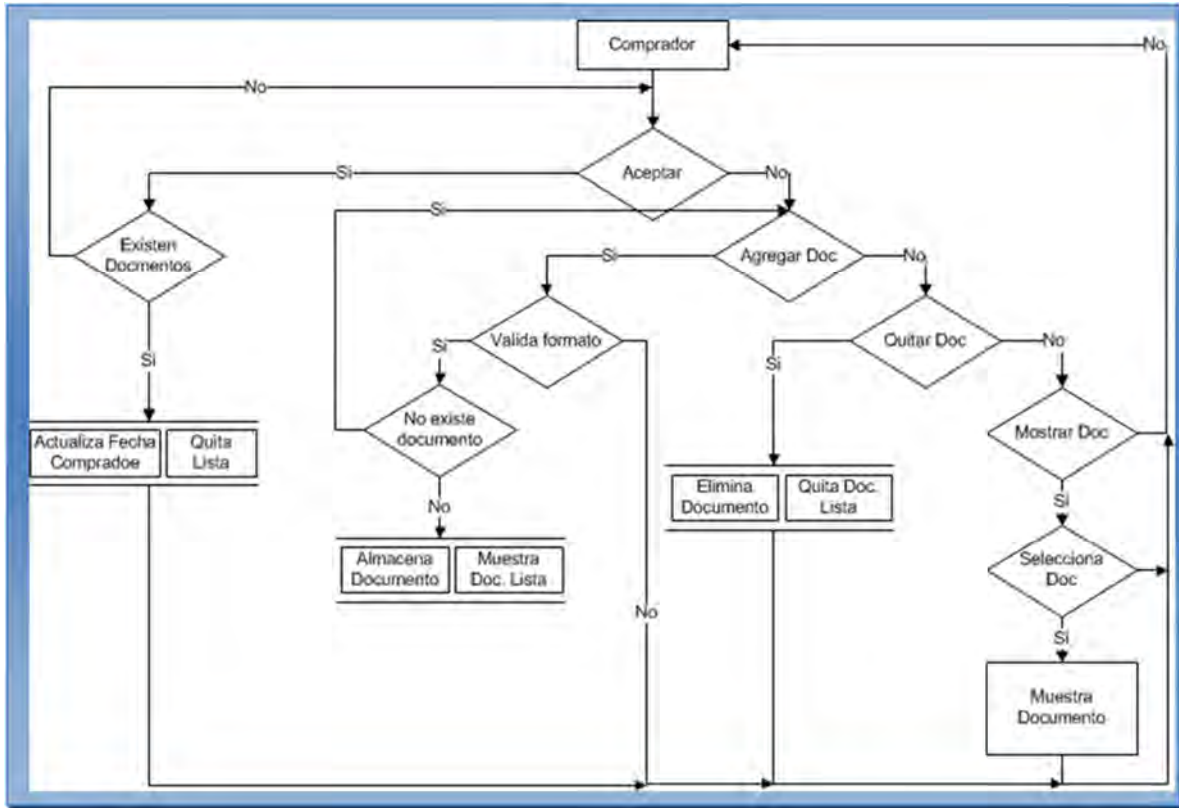


Figura 2.49. Diagrama que representa el flujo general de la pantalla del comprador.

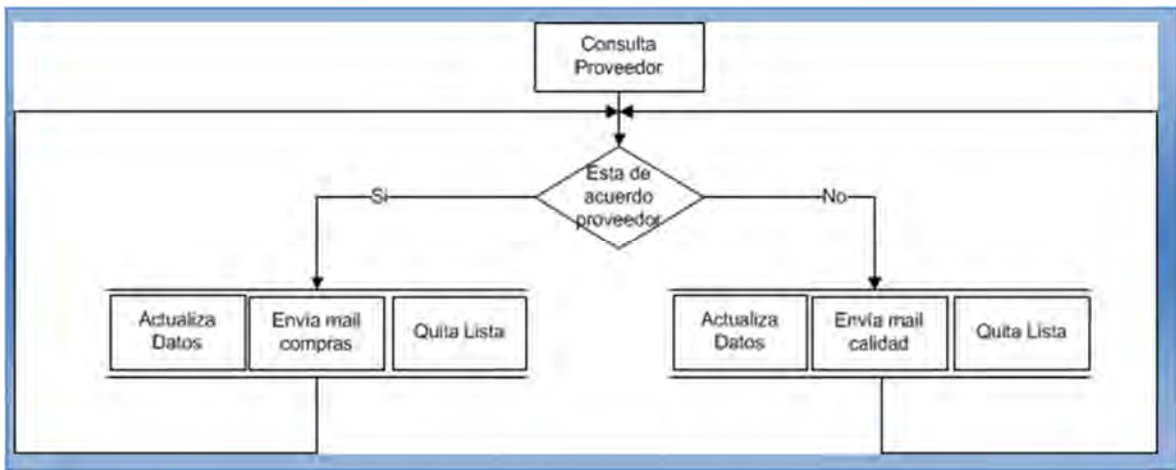


Figura 2.50. Diagrama que representa el flujo general de la pantalla consulta proveedor.

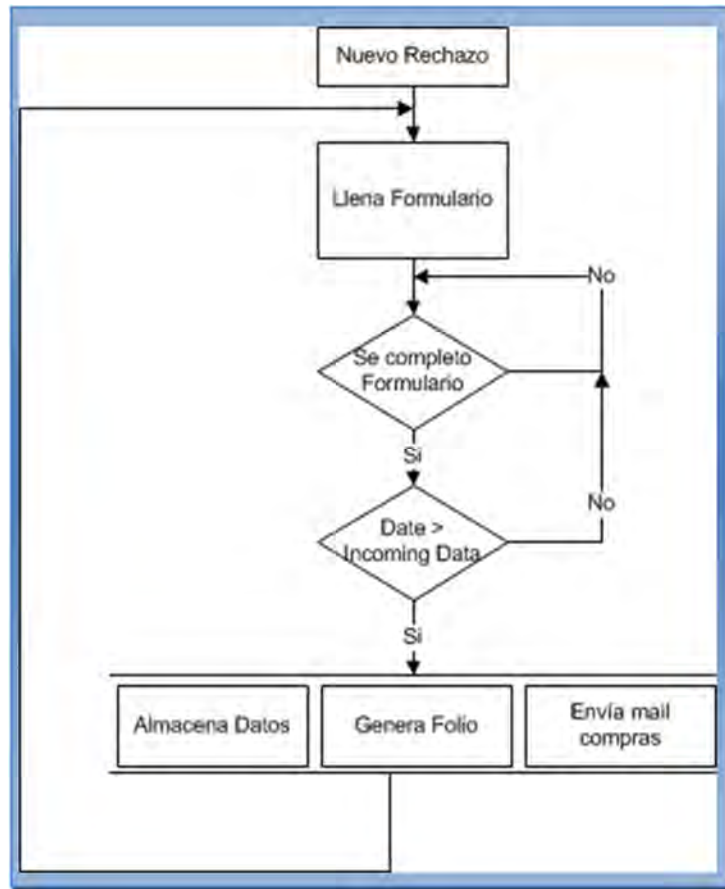


Figura 2.51. Diagrama que representa el flujo general de la pantalla nuevo rechazo.

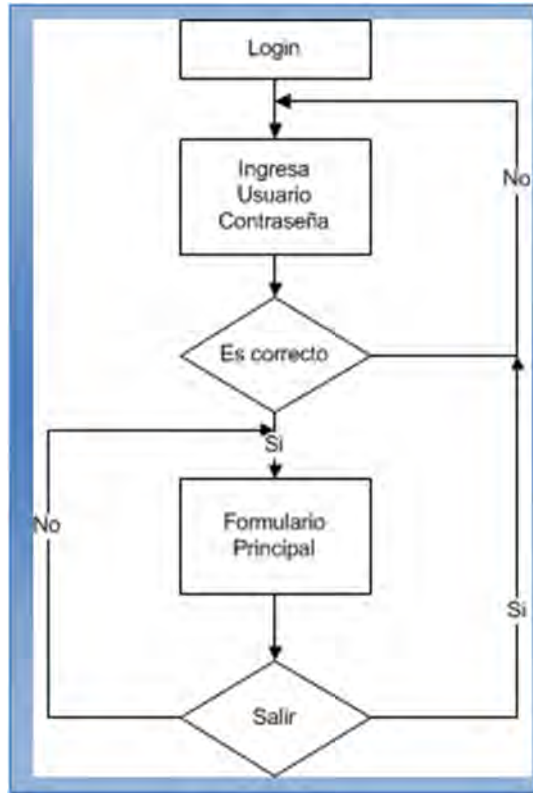


Figura 2.52. Diagrama que representa el flujo general de la pantalla login.

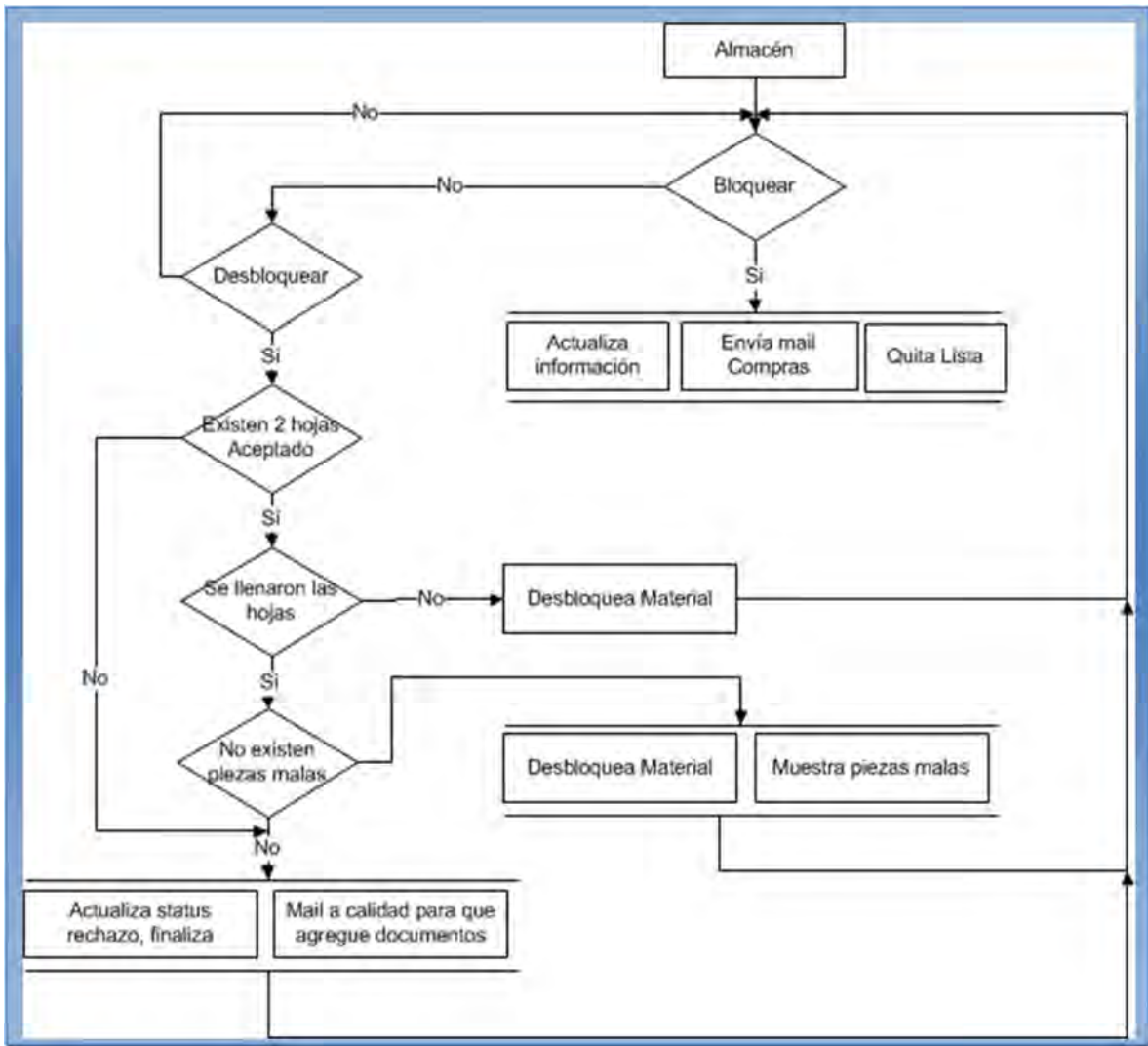


Figura 2.53. Diagrama que representa el flujo general de la pantalla almacén.

2.4.3. Fase de construcción

Implementación

El sistema se desarrolla gradualmente en forma incremental e iterativo de tal manera que el diseño, la programación, pruebas, implementación y documentación se lleva a cabo simultáneamente, con esto es posible tener un control sobre la versión o el producto terminado en cada iteración.

Para la fase de implementación se presentaran los siguientes aspectos.

- ✓ Lineamientos para la instalación e implementación

Para la instalación del sistema se tomaran las siguientes secciones.

- Distribución de la aplicación.
Dado a que el sistema es un sitio web, solo es necesario enviar por correo a todos los involucrados la liga donde podrán tener acceso al sistema.
- Requerimientos del equipo
Estaciones de trabajo
 - Procesador Pentium III
 - Disco duro de 20 Gb.
 - 128 Mb en memoria RAM¹¹
 - Explorer 6 o superior.
 - Mozilla 3 o superior.
 - Safari 3 o superior.
- Requerimiento mínimos y pasos para la Instalación del software
Servidor
 - Windows NT Ver. 4.0 o superior Instalado en servidor.
 - IIS 6
 - Framework 3.5 o superior
 - Crystal Report*Servidor de base de datos*
 - Procesador Pentium IV de 1.5 GHz.
 - Disco duro de 40 Gb.

¹¹ (RAM) Random Access Memory. Memoria de acceso aleatorio.

256 en memoria RAM

SQL server 2005

Pasos para la instalación del sitio web

Descomprimir archivo rechazos.zip en C:\inetpub\wwwroot

Entrar al administrador Integretion Services.

Crear un directorio virtual.

Asignar framework 3.5.

Modificar cadena de conexión con los del servidor de base de datos.

Actualizar cambios.

- ✓ Manual de usuario.

El manual del usuario tiene todas las instrucciones para la utilización del software que hemos desarrollado. Debido a que este muestra todas las interfaces gráficas con el usuario, se decidió ubicarlo en la pantalla de login. De esta manera permitimos que el usuario pueda tener acceso a la documentación del manual en cualquier momento o pueda imprimirlo previa descarga.

Pruebas

En este apartado es necesario evaluar la calidad del sistema en todo momento permitiendo realizar mejores significativas no solo al final de proyecto sino durante todo el desarrollo del sistema, para ello se hará enfoque en los siguientes puntos:

- ✓ Cualquier error será documentado, para poderle dar un seguimiento.
- ✓ Se solicitará una opinión a cada encargado sobre el funcionamiento y calidad del sistema.
- ✓ Validar con cada área, cada una de las funciones del producto según lo diseñado.
- ✓ Validar que los requisitos tengan su apropiada implementación.

Distribución

Esta actividad tiene como objetivo producir con éxito distribuciones del producto y distribuirlo a los usuarios. Las actividades implicadas incluyen:

- ✓ Validar la correcta ejecución de la página en el servidor así como los complementos instalados.
- ✓ Realizar pruebas de visualización de reporte y la funcionalidad para exportar los documentos.
- ✓ Distribuir la liga del sitio web a todos los involucrados.
- ✓ Validar la correcta visualización en todos los exploradores.
- ✓ Validar la integridad y almacenamiento de los datos.
- ✓ Capacitar al personal y resolver dudas que pudieran surgir.

2.5. Diseño de base de datos

2.5.1. Bases de datos relacionales

Las bases de datos relacionales son aquellas en las que la información se almacena en tablas (filas y columnas de datos) y se pueden realizar búsquedas utilizando los datos de columnas específicas de una o varias tablas, pudiendo encontrar datos en distintas tablas.

En una base de datos relacional, las filas representan registros y las columnas representan campos. Al realizar las búsquedas, una base de datos relacional hace coincidir la información de un campo de una tabla con información en el campo correspondiente de otra tabla y con ello produce una tercera tabla que combina los datos solicitados de ambas tablas.

Por ejemplo se tiene un idRechazo, vendor, tipoMaterial y otra tabla contiene idRechazo, statusCalidad, statusCompras , una base de datos relacional hace coincidir el campo idRechazo de las dos tablas para encontrar información.

Una base de datos relacional es una base de datos basada en un modelo relacional. Estrictamente hablando el término se refiere a una colección específica de datos pero a menudo es usado como sinónimo del software usado para gestionar esa colección de datos. Ese software se conoce como sistema gestor de base de datos relacional o RDBMS¹².

Con base en la definición y al sistema se utilizara una base de datos relacional ya que se contarán con distintas tablas las cuales contendrán seguimientos, rechazos,

¹² (RDBMS) Relational Database Management System. Sistema de gestión de bases de datos relacionales.

documentos, proveedores y sku los cuales al realizar los cruces correspondientes nos devolverán la información de manera rápida y fiable.

2.5.2. Normalización y diseño de base de datos

Diseño

Antes de diseñar los objetos que formarán la base de datos, es importante invertir algún tiempo en diseñar la base de datos. Un buen diseño de la base de datos es la pieza clave para crear una base de datos que realice las operaciones que desee de una forma efectiva, precisa y eficaz.

Pasos para el diseño de la base de rechazos.

1. Finalidad de la base de datos.- Se determina la finalidad de la base de datos con base en los requerimientos levantados, de ahí se sabe que información se requerirá obtener de la base se puede determinar sobre qué asuntos necesita almacenar hechos (las tablas) y qué hechos necesita almacenar sobre cada asunto (los campos de las tablas).
2. Determinar tablas.- En base a los reportes que se requieren y la información que se requiere almacenar, se determina el numero de tablas, para el sistema de rechazos se cuentan con 3 tablas principales rejection, seguimiento y hojas de aceptado, con estas se desencadenan una serie de tablas las cuales serán catálogos que alimentaran a las tablas principales.

Una tabla no debe contener información duplicada y la información no debe duplicarse entre las tablas. Cuando cada elemento de información está almacenado en una tabla, se actualiza en un solo lugar. Esto resulta más eficiente y elimina la posibilidad de que existan entradas duplicadas que contengan información diferente.

3. Identificadores únicos para cada tabla.- cada tabla de la base de datos deberá incluir un campo o un conjunto de campos que identifiquen de forma exclusiva cada registro individual de la tabla.

4. Determinar las relaciones entre las tablas.- es necesario definir relaciones entre las tablas, para garantizar la integridad de los datos y evitar la pérdida de información por un mal manejo.

Normalización

La normalización es el proceso de organizar los datos de una base de datos. Se incluye la creación de tablas y el establecimiento de relaciones entre ellas según reglas diseñadas tanto para proteger los datos como para hacer que la base de datos sea más flexible al eliminar la redundancia y las dependencias incoherentes.

Los datos redundantes provocan que se desperdicie espacio de disco y crean problemas de mantenimiento. Si hay que cambiar datos que existen en más de un lugar, se deben cambiar de la misma forma exactamente en todas sus ubicaciones. Un cambio en la información de un rechazo es mucho más fácil de implementar si los datos sólo se almacenan en la tabla rechazos y no en algún otro lugar de la base de datos.

Para la base de rechazos se realizará la siguiente normalización.

- ✓ Se eliminarán los grupos repetidos de cada tabla.
- ✓ Se creará una tabla independiente para cada conjunto de datos relacionados.
- ✓ Existirá una clave principal para cada tabla relacionada.
- ✓ Se crearán catálogos independientes para administrar la tabla de rechazo.
- ✓ Se asignarán las claves foráneas correspondientes.
- ✓ Eliminación de claves que no existan en la tabla de rechazo.

Con lo anterior se cumplen las tres formas de normalización de una base.

2.5.3. Diagrama físico de base de datos del sistema

Es necesario definir las entidades a cualquier objeto, real o abstracto, que existe en un contexto determinado o puede llegar a existir y del cual deseamos guardar la información por ello es necesario desarrollar un diagrama de entidades el cual permita conocer el número de objetos a definir, la figura 2.54 nos permite visualizar dichos objetos.

Un esquema físico permite la descripción de la implementación de una base de datos, define las estructuras de almacenamiento y los métodos utilizados para tener un acceso eficiente. Permite visualizar las claves principales y foráneas, definiendo de manera correcta las entidades que son consideradas tablas y los atributos que vienen a ser los campos. Figura 2.55

Modelo Rechazos

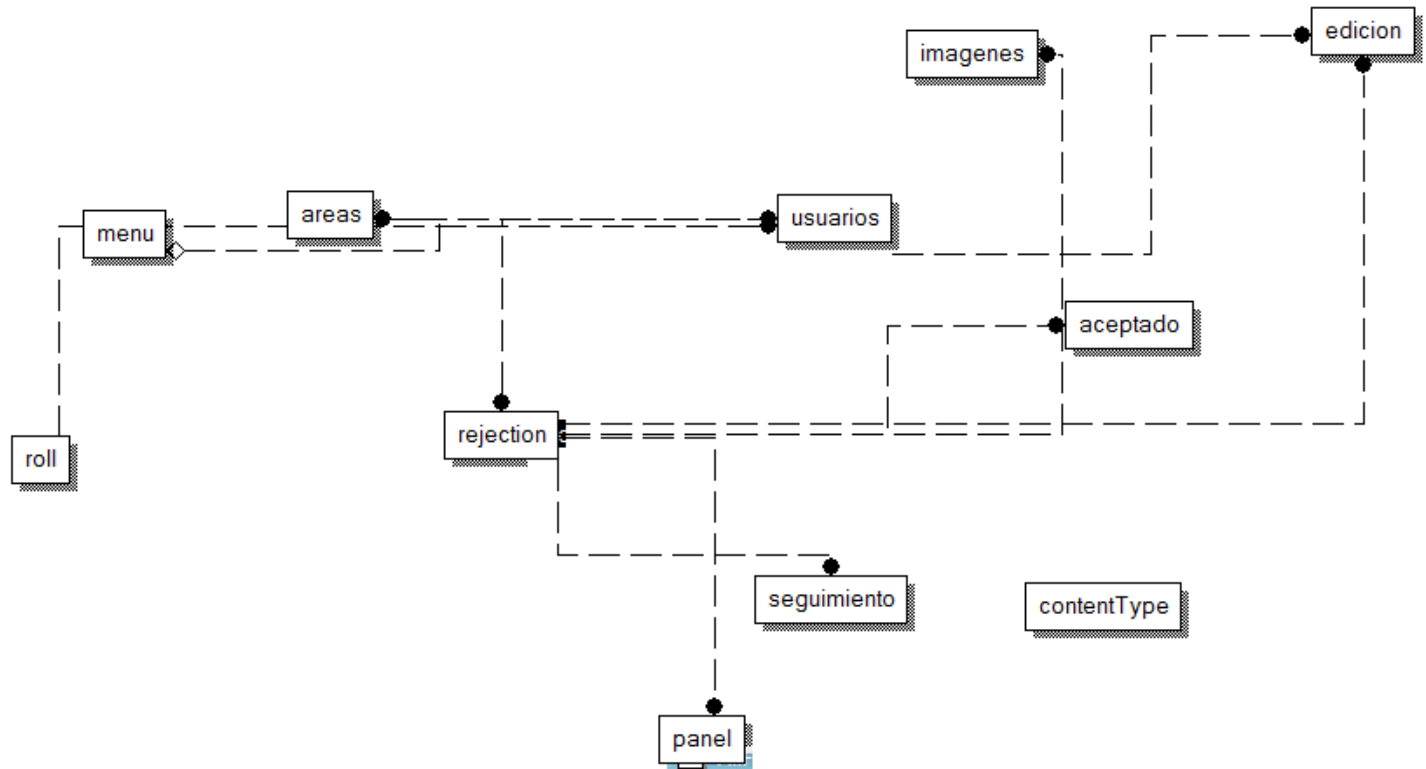


Figura 2.54. Diagrama de entidades de la base de datos.

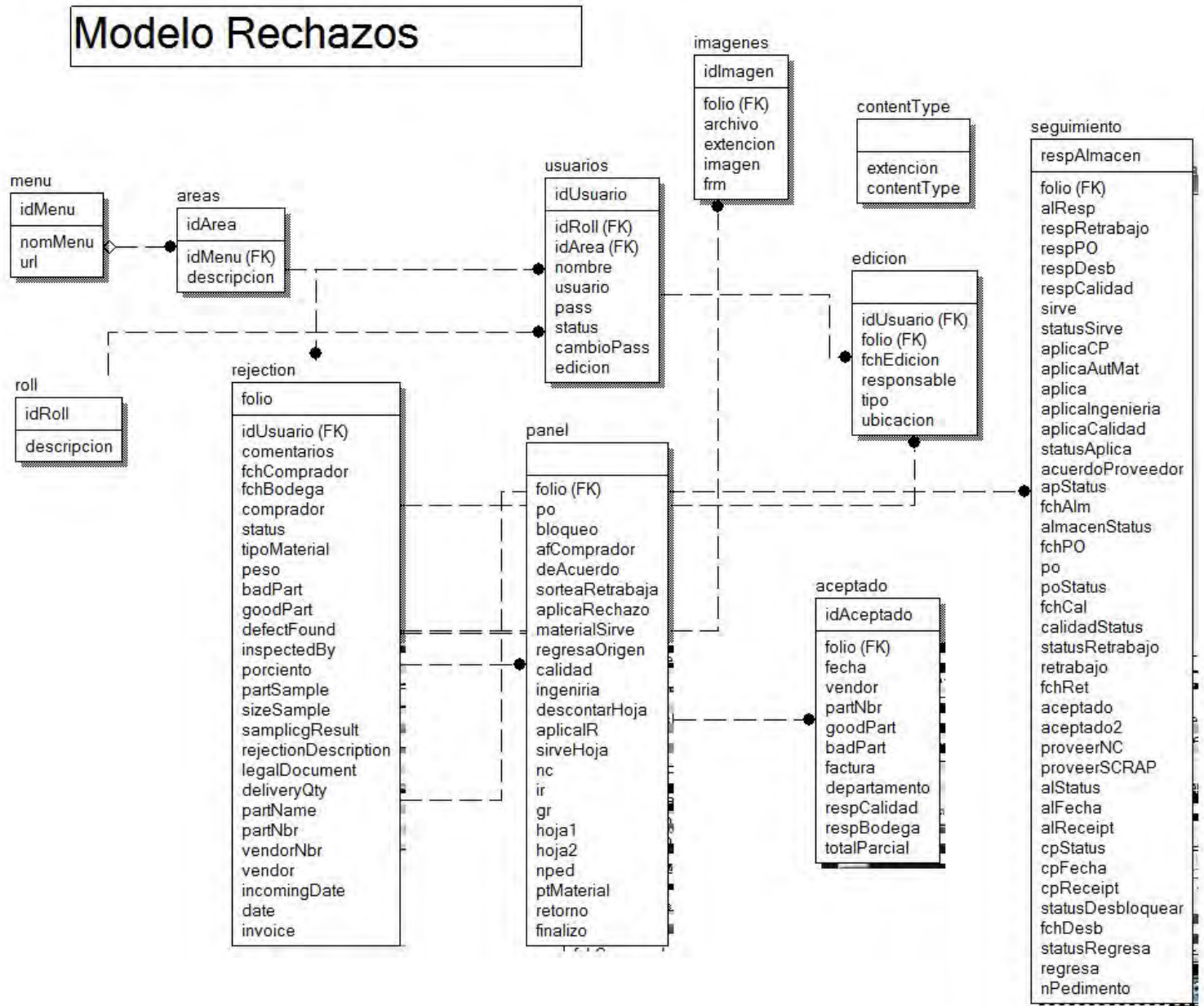


Figura 2.55. Diagrama físico de la base de datos.

CAPÍTULO III
DESARROLLO DE LA APLICACIÓN

3. Desarrollo de la Aplicación

3.1. Lenguaje de programación

“Un lenguaje de programación es un lenguaje artificial que puede ser usado para controlar el comportamiento de una máquina, en una forma legible tanto para la maquina como para el ser humano. Estos se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas. ”¹³

3.1.1. Microsoft .NET¹⁴

“Microsoft.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años con el objetivo de obtener una plataforma sencilla y potente para distribuir el software en forma de servicios que puedan ser suministrados remotamente y que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, lenguaje de programación y modelo de componentes con los que hayan sido desarrollados.”¹⁵

La utilización del .NET tiene la ventaja de que reúne en una misma plataforma un conjunto interesante de características, como independencia de plataforma, independencia de lenguaje, soporte de bases de datos, soporte para XML¹⁶, servicios Web y aplicaciones Web, entre otras. De entre todas estas se destacan dos, las cuales son muy importantes y son la razón por la cual se optó por el desarrollo en esta plataforma.

- ✓ Independencia del lenguaje
Gracias a la independencia del lenguaje es posible desarrollar aplicaciones en distintos lenguajes dentro de la misma plataforma .NET, lo que conlleva a que una aplicación pueda tener diferentes partes desarrolladas en distintos lenguajes y poderse comunicar entre si de forma transparente. Permitiendo poder reutilizar código, ya que las clases desarrolladas dentro del proyecto podrán ser reutilizadas en uno o varios proyectos independientemente del lenguaje.

- ✓ Independencia de plataforma

¹⁴ NET (NETwork, internet)

¹⁶ XML (*eXtensible Markup Language*, lenguaje de marcas extensible).

Al igual que en Java, el código .NET no se compila a código máquina, sino a un código en un formato intermedio, independiente de la plataforma. Esto permite la aplicación se ejecute de modo distribuido, a lo largo de todo Internet, y que sea accesible desde múltiples dispositivos.

3.1.2. .NET Framework

NET framework es el corazón de la tecnología .NET. Es el marco de trabajo y ejecución común a toda la tecnología .NET. Es por lo tanto un elemento indispensable dentro de la tecnología .NET.

Gracias al Framework es posible realizar las aplicaciones en distintos lenguajes como C#, VB.NET y J#, por mencionar algunos.

El framework incluye soluciones en áreas como: la interfaz de usuario, acceso a datos, conectividad a bases de datos, criptografía, desarrollo de aplicaciones web, algoritmos numéricos y comunicación de redes.

3.1.3. Ajax Control Tool Kit

Es un conjunto de controles ya desarrollados y listos para ser usados que permiten utilizar la tecnología de asp .net ajax. Estos controles están disponibles en el cuadro de controles del usuario listos para arrastrar y usar.

Las principales características son:

- ✓ Dichos controles permiten realizar llamadas a servicios web para obtener información de manera eficiente sin tener que volver a recargar la página, lo que da un mayor performance.
- ✓ Actualizaciones parciales de la página.
- ✓ Compatibilidad con distintos exploradores.
- ✓ Mayor eficacia, porque las partes importantes del proceso de una página web se realizan en el explorador.

3.1.4. Itext Sharp

iText es una biblioteca Open Source para crear y manipular archivos PDF¹⁷, RTF¹⁸, y HTML¹⁹ en .NET. Fue escrita por Bruno Lowagie, Paulo Soares, y otros; está distribuida bajo la Mozilla Public License con la LGPL²⁰ como licencia alternativa.

Dicha herramienta permite de manera fácil y rápida exportar un documento a múltiples formatos, o múltiples instancias del mismo formato. Su fácil instalación y manejabilidad hizo que fuera una de las mejores opciones a elegir para exportar la información de los rechazos.

“Más recientemente, ha sido extendida a una biblioteca PDF de propósito general, capaz de rellenar formularios, mover páginas de un PDF a otro, y otras cosas. Estas extensiones son a menudo mutuamente excluyentes. Una clase te permite rellenar en formularios, mientras una clase diferente e incompatible hace posible copiar páginas de un PDF a otro.”²¹

3.1.5. JQuery

jQuery es un framework Javascript, el cual sirve como base para la programación avanzada de aplicaciones, que aporta una serie de funciones o códigos para realizar tareas habituales.

Una de las desventajas que se tiene con JavaScript es que se tienen que realizar distintas funciones para cada explorador, para hacer una u otra cosa dependiendo de si es Internet Explorer, Firefox, Opera, etc., jQuery no tiene ese inconveniente puesto que “implementa una serie de clases (de programación orientada a objetos) que nos permiten programar sin preocuparnos del navegador con el que nos está visitando el usuario, ya que funcionan de igual forma en todas las plataformas más habituales.”²²

Así pues, este framework Javascript, ofrece una infraestructura con la que se trabaja con facilidad para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, con jQuery obtendremos ayuda en la creación de interfaces de usuario,

¹⁷ PDF (*Portable Document Format*, formato de documento portátil).

¹⁸ RTF (*Rich Text Format*, formato de texto enriquecido).

¹⁹ HTML (*HyperText Markup Language*, lenguaje de marcado de hipertexto).

²⁰ LGPL (*Lesser General Public License*, Licencia Pública General Reducida).

efectos dinámicos en los reportes, autocompletables en el formato de nuevo rechazo, manipulación de CSS²³, etc.

Sus principales funcionalidades son:

- ✓ Se puede manipular el DOM²⁴ con menos código, es decir con menos código se pueden realizar más cosas, obviamente además de facilitar la manipulación de la pagina con consultas más sencillas conllevara a páginas más ligeras.
- ✓ Permite realizar aplicaciones AJAX mas ligeras y sencillas a través del uso de funciones que permiten obtener datos del servidor codificados en JSON sin necesidad de refrescar la pagina usando una petición Get XMLHttpRequest.
- ✓ Asignación de funciones en los eventos de los input que realizan tareas simultaneas de manera fácil y con menos código.
- ✓ Validaciones más precisas y fáciles de implementar.

3.1.6. Web Services

“Un web service es un conjunto de protocolos cuya utilidad específica es permitir el intercambio de información entre aplicaciones, aun en el caso de que éstas estén programadas sobre tecnologías y lenguajes diferentes.”²⁵

Una de las principales ventajas de los web services es que la comunicación entre distintas aplicaciones resulta independiente de la plataforma con la que estén desarrolladas las aplicaciones que se comunican por este medio, por lo que supone un modelo de interacción de gran utilidad, ya que permite enlazar nuevos desarrollos con los anteriormente existentes, sin que la tecnología de los más antiguos condicione a los desarrollos más recientes.

Esto resulta interesante ya que actualmente American Standard aglutinan un parque de aplicaciones muy variado, que de este modo pueden hacer trabajar en colaboración con un coste mucho más reducido de lo que supondría acometer migraciones completas.

²³ CSS (*Cascading Style Sheets*, hojas de estilo en cascada).

²⁴ DOM (*Document Object Model*, Modelo de Objetos del Documento).

3.1.7. HTML Estático

HTML es el lenguaje que define a las páginas web, se trata de un conjunto de etiquetas que sirven para definir los distintos elementos que componen una página web, como son:

- ✓ **<INPUT>**: define cajas de texto, botones, checkbox y radio, por mencionar algunos.
- ✓ **<TABLE>**: es una tabla y dentro de esta tenemos filas **<TR>** y celdas **<TD>**
- ✓ **<A>**: define los enlaces.
- ✓ ****: los textos dentro de esta etiqueta se estructuran en listas. Mediante el uso de **** definimos cada guión dentro de la lista, y usando **** en lugar de **** tendremos listas ordenadas.
- ✓ ****: imágenes.

Con las etiquetas anteriores es posible construir un sitio web que puede ser suficiente para muchas aplicaciones, en ocasiones necesitaremos que el contenido de nuestra página web se genere dinámicamente en función de las necesidades y deseos de los usuarios, para esto se utilizará la tecnología que proporciona ASP.NET, la cual permitirá generar documentos de manera dinámica.

Éste es el principio de funcionamiento de las aplicaciones web.



Figura 3.1 HTML estático: Configuración típica de una "aplicación web" que se limita a ofrecer la información almacenada en páginas HTML a las que el usuario final accede desde su navegador web.

3.2. Interfaz Gráfica del Sistema

La interfaz de un sistema es la forma en que se presentará al usuario los datos además de que es la interacción entre las personas y la computadora, tiene como objetivo ser

amigable y fácil de utilizar para inducir al usuario a que la utilice, para ello se presentan las siguientes pantallas que conformaran al sistema.



Figura 3.2 Login de usuario.



Figura 3.3 Captura y edición de usuarios nuevos/existentes.

REJECTION REPORT			
		Rejection Nbr:	40
		Invoice Nbr:	<input type="text"/>
Data Rejection:			
Date:	<input type="text"/>	Incoming Date:	<input type="text"/>
Vendor:	run		
Part Nbr:	RUNNER CORPORATION		
Delivery Qty:	000113	Part Name:	
	000116	Legal Document	<input type="text"/>
	000117		
	000117-0020A		
Rejection Description:	<input type="text"/>		
Good Part:	<input type="text"/>	Bad Part:	<input type="text"/>
Sampling Result(C/U=%):	100/200=50%	Inspected by	Lorena Sánchez
Where were defect found:	Seleccione una Opción		
Aceptar			

Figura 3.4 Capturar de nuevo folio para el producto no conforme.

NUMERO DE PO	
PO:	<input type="text" value="123456"/>
<input type="button" value="Aceptar"/> <input type="button" value="Cancelar"/>	

Figura 3.5 Captura del número de PO.

45	SECORH CONSULTORES EJECUTIVOS	000117-0020A	description	Ver	<input type="button" value="Bloquear"/>
----	-------------------------------	--------------	-------------	---------------------	---

Figura 3.6 Bloqueo de material.

45	SECORH CONSULTORES EJECUTIVOS	000117-0020A	description	Ver	✓	✗
----	-------------------------------	--------------	-------------	---------------------	---	---

Figura 3.7 Consulta proveedor.

45	SECORH CONSULTORES EJECUTIVOS	000117- 0020A	description	Ver	✓	✗
----	----------------------------------	------------------	-------------	---------------------	---	---

Figura 3.8 Indicar disposición.

PROVEER NC

Examinar...

>>>

<<<

Hoja de aceptado 0423.jpg

Aceptar

Scrap >>>

Figura 3.9 Proveer NC y/o SCRAP.

CAPTURA FOLIO

Fecha:

08/03/2010

Good Receipt:

Aceptar

Cancelar

>>>

Figura 3.10 Almacén captura folio.

FOLIO DE CARGO

Fecha:

08/03/2010

Invoice Receipt:

Aceptar

Cancelar

>>>

Figura 3.11 Cuentas por pagar captura folio.

46	HABINSA TRANSPORTADORES Y ACCESORIO	M907004- 295	Prueba de Rechazo	Ver	✓	✗
----	---	-----------------	-------------------	---------------------	---	---

Figura 3.12 Disposición IMPO-EXPO.

The screenshot shows a form with a green header bar containing the text "PESO Y TIPO MATERIAL". Below the header, there is a field for "Folio:" with the value "44". Underneath, there are two input fields: "Peso:" and "Tipo de Material:". At the bottom of the form, there are two buttons: "Aceptar" and "Cancelar".

Figura 3.13 Agregar peso y tipo de material.

The screenshot displays a form with a green header. The first row contains "Folio" with the value "0000". The second row has "Número de Pedimento:" followed by an empty input field with a red asterisk. Below this, there is a row with an empty input field, a button labeled "Examinar...", two arrow buttons ">>>" and "<<<", and another empty input field containing the text "Sin enlazar". The bottom row features three buttons: "ACEPTAR", "CANCELAR", and "MOSTRAR DOC.", all set against a green background.

Figura 3.14 IMPO-EXPO

The screenshot shows a dialog box with a green background and a dashed border. The title is "Se tramita SCRAP". Below the title, there are two buttons: "Si" and "No".

Figura 3.15 Se tramita SCRAP.

The screenshot shows a dialog box with a green background. The title is "¿Se tramita retorno?". Below the title, there are two buttons: "Si" and "No".

Figura 3.16 Almacén valida si se tramita el regreso del material.

Figura 3.17 Piezas buenas y malas.

46	HABINSA TRANSPORTADORES Y ACCESORIO	M907004- 295	Prueba de Rechazo	Ver	Desbloquear
----	---	-----------------	-------------------	---------------------	-------------

Figura 3.18 Desbloqueo de almacén.

Figura 3.19 Documento autorización de materiales descontar.

Figura 3.20 Validación calidad/ingeniería.

3.3. Herramientas de Desarrollo

3.3.1. Visual Studio

“Visual Studio es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones Web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C++, Visual C# y Visual J# utilizan el mismo Entorno de Desarrollo Integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y Servicios Web XML.”²⁶

Dicho software nos proporciona un completo conjunto de herramientas para la creación de aplicaciones de escritorio y web empresariales cuya distribución, administración y mantenimiento se realiza de manera rápida y fácil, permitiendo dar soluciones integrales para cada empresa. Aparte de generar aplicaciones de escritorio de alto rendimiento, se pueden utilizar las eficaces herramientas de desarrollo basado en componentes y otras tecnologías de Visual Studio para simplificar el diseño, desarrollo e implementación en equipo de soluciones empresariales.

3.3.2. Aplicaciones Web en plataforma .NET

“ASP.NET es el nombre con el que se conoce la parte de la plataforma .NET que permite el desarrollo y ejecución tanto de aplicaciones web como de servicios web. Igual que sucedía en ASP, ASP.NET se ejecuta en el servidor. En ASP.NET, no obstante, las aplicaciones web se suelen desarrollar utilizando formularios web, que están diseñados para hacer la creación de aplicaciones web tan sencilla como la programación en Visual Basic.”²⁷

ASP.NET se integra a la plataforma de .NET, los formularios para Windows y los formularios ASP.NET forman parte de las herramientas con las que se pueden construir interfaces de usuarios.

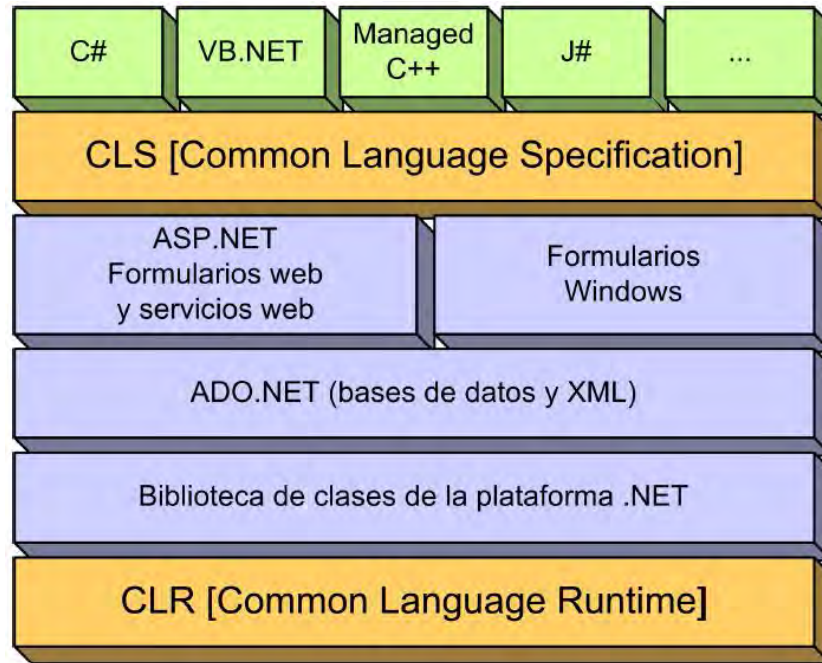


Figura 3.21. La plataforma .NET: Los formularios ASP.NET y los formularios Windows son las dos alternativas principales de las que dispone el programador para crear las interfaces de usuario de sus aplicaciones.

Igual que sucede en el caso de los formularios Windows, la programación en ASP.NET está basado en el uso de controles y eventos. Las páginas ASP.NET, en vez de aceptar datos de entrada y generar su salida en HTML, implementan su funcionalidad en fragmentos de código que se ejecutan como respuesta a eventos asociados a los controles de la interfaz con los que puede interactuar el usuario. Dicha forma de funcionar le proporciona un mayor nivel de conceptualización con lo que se requiere menos código y permite crear aplicaciones más modulares, entendibles y de fácil mantenimiento.

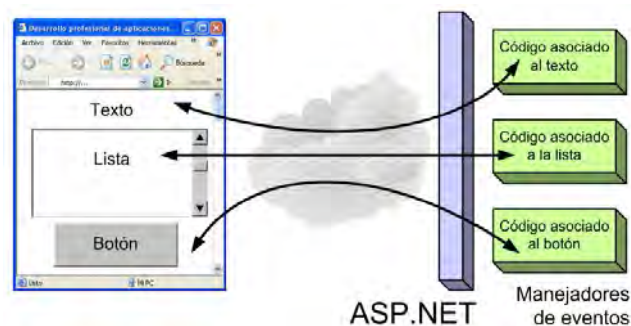


Figura 3.22. Eventos en ASP.NET: La respuesta de la aplicación web se obtiene como resultado de ejecutar los manejadores de eventos asociados a los controles incluidos en la interfaz de usuario.

Una de las características destacables de ASP.NET es que todas las etiquetas que se introducen en la página HTML para incluir controles en la interfaz del usuario son independientes del HTML que después se genera para construir la interfaz de usuario que le llega al navegador. Es el caso, por ejemplo, de los controles `<asp:Label...>` y `<asp:Button...>`. ASP.NET es el encargado de convertir cualquier etiqueta en el fragmento de HTML que resulte más adecuado para mostrar los controles en función del navegador web que utilice el usuario de nuestra aplicación. De esta forma, ASP.NET garantiza la compatibilidad de los controles de nuestra aplicación web con distintos navegadores, permitiendo despreocuparse de las diferencias existentes entre los diferentes navegadores web que puede emplear el usuario final para acceder a nuestra aplicación.

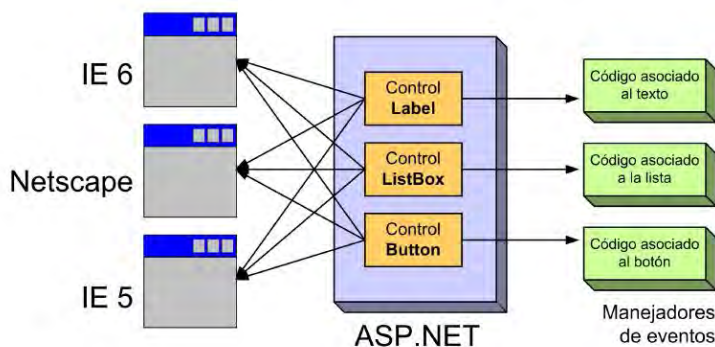


Figura 3.23. ASP.NET se encarga de mostrar los formularios web de la forma que resulte más adecuada para el navegador que utilice el usuario final de la aplicación.

3.3.3. Desarrollo diagramas UML con Visio

UML se compone de un grupo de especificaciones de notación que van encaminadas a la programación orientada a objetos, las cuales están compuesta por distintos diagramas, que representan las diferentes etapas del desarrollo de un proyecto de software.

“Microsoft Visio es un software de dibujo vectorial para Microsoft Windows. Visio comenzó a formar parte de los productos de Microsoft cuando fue adquirida la compañía Visio en el año 2000.”²⁸

Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, y más.

Aunque originalmente apuntaba a ser una aplicación para dibujo técnico para el campo de Ingeniería y Arquitectura; con añadidos para desarrollar diagramas de negocios, su adquisición por Microsoft implicó drásticos cambios de directrices de tal forma que a partir de la versión de Visio para Microsoft Office 2003 el desarrollo de diagramas para negocios pasó de añadido a ser el núcleo central de negocio, minimizando las funciones para desarrollo de planos de Ingeniería y Arquitectura que se habían mantenido como principales hasta antes de la compra. Una prueba de ello es la desaparición de la función "property line" tan útil para trabajos de agrimensura y localización de puntos por radiación, así como el suprimir la característica de forma de fantasma que facilitaba la ubicación de los objetos en dibujos técnicos.

3.3.4. Modelado base de datos con ERwin

"AllFusion ERwin Data Modeler es una herramienta de diseño de base de datos que ayuda a los usuarios a diseñar, generar y mantener alta calidad de las aplicaciones de base de datos de alta performance. " ²⁹

Erwin permiten visualizar la estructura, elementos clave y el diseño optimizado de la base de datos, desde los requerimientos de un modelo lógico y reglas de negocio que definen la base, a un modelo físico optimizado para las características propias de estas.

Permite generar tablas, líneas de procedimientos almacenados y disparadores de forma automática pudiendo tener una base de datos más integrada y controlada con la información y procedimientos que esta realizará. Su tecnología de "comparación completa" permite el desarrollo iterativo, de forma tal que los modelos están siempre sincronizados con la base de datos. Al integrarse con entornos de desarrollo líderes, AllFusion ERwin Data Modeler también acelera la creación de aplicaciones centralizadas en datos.

La mejor gestión de la información empieza por un diseño óptimo de las bases de datos. Mediante AllFusion ERwin Data Modeler, se pueden visualizar estructuras complejas de datos y activos de información corporativa, así como establecer estándares de gestión de datos para toda la empresa. Permite automatizar de forma

inteligente procesos de diseño y sincronizar el modelo con el diseño de bases de datos.

Características generales

- ✓ Aumenta la productividad.
- ✓ Comunica en forma más efectiva.
- ✓ Permite modificaciones a las estructuras de forma fácil, sin perder la integridad de esta.
- ✓ Tecnología transformable.
- ✓ Administra grandes modelos.
- ✓ Comparación completa.
- ✓ Genera diseños de base de datos.
- ✓ Diseña almacenes de datos y mercados de datos.

3.4. Servidor Web

3.4.1. IIS

“Internet Information Server (IIS), es una serie de servicios para las computadoras que funcionan con Windows. Originalmente era parte del Option Pack para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP³⁰, SMTP³¹, NNTP³² y HTTP³³/HTTPS^{34, 35}.”

Dicho servicio convierte a una computadora en un servidor de internet o intranet, es decir los que tienen instalado este servicio pueden publicar páginas web tanto local como remotamente.

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

³⁰ FTP (*File Transfer Protocol*, Protocolo de Transferencia de Archivos).

³¹ SMTP (*Simple Mail Transfer Protocol*, Protocolo Simple de Transferencia de Correo).

³² NNTP (*Network News Transport Protocol*, protocolo para la transferencia de noticias en red).

³³ HTTP (*HyperText Transfer Protocol*, Protocolo de transferencia de hipertexto).

³⁴ HTTPS (*HyperText Transfer Protocol Secure*, Protocolo seguro de transferencia de hipertexto).

Entre sus principales características se tienen:

- ✓ Reiniciar IIS: es posible reiniciar los servicios de Internet sin la necesidad de reiniciar el equipo.
- ✓ Mensajes de error personalizados mejorados: Es posible enviar mensajes personalizados a los usuarios cuando se producen errores de HTTP en los sitios Web, proporcionando un mejor control sobre dichos inconvenientes y darles un tratamiento eficaz.
- ✓ Opciones de configuración: Se pueden establecer los permisos para las operaciones Web de Lectura, Escritura, Ejecución y Secuencia de comandos en el nivel de sitios, directorios o archivos, permitiendo un mejor control sobre los sitios web.
- ✓ Almacenamiento de certificados: proporciona un único punto de entrada que le permite almacenar, realizar copias de seguridad y configurar certificados de servidor.
- ✓ Comunicaciones seguras: “Secure Sockets Layer (SSL) 3.0 y Seguridad de capa de transporte (TLS) proporcionan una forma segura para intercambiar información entre clientes y servidores”³⁶. Además nos permite comprobar quien es el cliente antes de que el usuario inicie sesión en el servidor, permitiendo llevar un mejor control sobre los accesos.

3.4.2. SQL Server

“SQL Server es un conjunto de objetos eficientemente almacenados. Los objetos donde se almacena la información se denominan tablas, y éstas a su vez están compuestas de filas y columnas. En el centro de SQL Server está el motor de SQL Server, el cual procesa los comandos de la base de datos. Los procesos se ejecutan dentro del sistema operativo y entienden únicamente de conexiones y de sentencias SQL.”³⁷

Es necesario realizar una buena instalación y configuración de SQL Server ya que con esto se garantiza que las consultas que se realicen mediante sentencias SQL obtengan un tiempo de respuesta óptimo además de que la memoria y el CPU se aprovechen al máximo.

SQL Server utiliza el lenguaje Transact-SQL para enviar peticiones tanto de consultas, inserciones, modificaciones, y de borrado a las tablas, así como otras peticiones que el sistema enviar para solicitar datos.

Existen dos motores muy importantes en SQL Server:

- ✓ El motor relacional: Incluye los componentes necesarios para la consulta de datos.
- ✓ El motor de almacenaje: Gestiona el almacenaje físico de los datos y la actualización de los datos en disco.

SQL Server se encarga de administrar bases de datos relacionales basadas en la arquitectura Cliente / Servidor (RDBMS: Relational Database Management System).

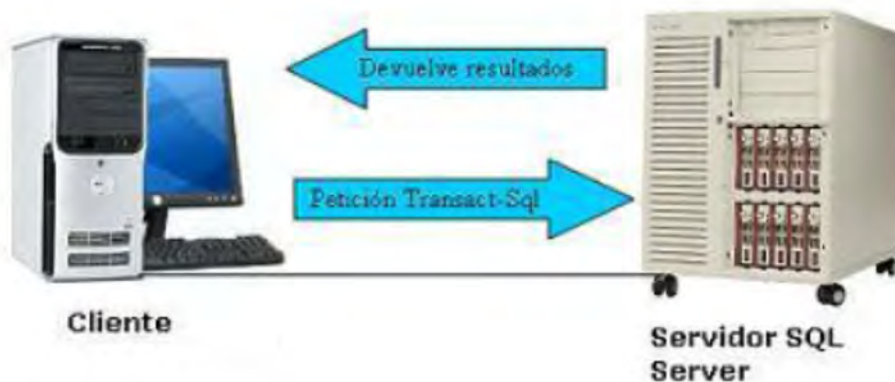


Figura 3.24. Arquitectura cliente/servidor.

SQL Server utiliza la arquitectura Cliente / Servidor para coordinar el trabajo entre el equipo cliente y el equipo servidor. Dependiendo del tipo de aplicación que se quiera programar dependerá la repartición de la carga de trabajo entre el cliente y el servidor.

El equipo cliente se suele encargar de la parte lógica y de mostrar la información al usuario que realiza una petición.

El equipo servidor SQL Server, se encarga de administrar la base de datos, de gestionar los recursos del servidor. (CPU, memoria, etc.), y por supuesto de resolver y devolver en forma de solución la petición realizada por el cliente.

CAPÍTULO IV
INPLANTACIÓN DEL SISTEMA

4. Implantación del Sistema

4.1. Descripción del proceso de Implementación

La situación se plantea para las siguientes condiciones:

- ✓ Windows 2003 Server
- ✓ IIS 6.0
- ✓ Sitio web .NET 3.5

En la figura 4.1 se muestra un ejemplo de los distintos ficheros que pueden existir en una aplicación Web de ASP.Net.

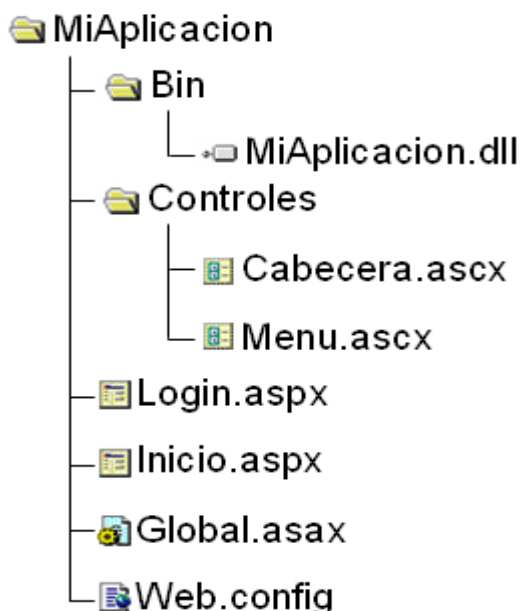


Figura 4.1. Aplicación Web de ASP.Net.

Una vez creada la aplicación, el código del servidor se ensambla en un archivo .dll situado en la carpeta Bin de la aplicación Web. Por tanto, una vez realizada la compilación, los ficheros de código (.vb ó .cs) ya no son necesarios para la ejecución de la aplicación ya que están ensamblados en la dll y es aconsejable quitarlos del servidor para que no se pueda acceder desde el exterior a su contenido.

Para terminar, se va a crear una aplicación Web de tipo ASP.Net e instalarla en un servidor Web con IIS. El primer paso es crear la aplicación Web, para ello se entra en Visual Studio .Net y en el menú 'Archivo' se selecciona 'Nuevo proyecto'. Aquí se debe elegir uno de los lenguajes disponibles y seleccionar 'Aplicación Web ASP.Net'.

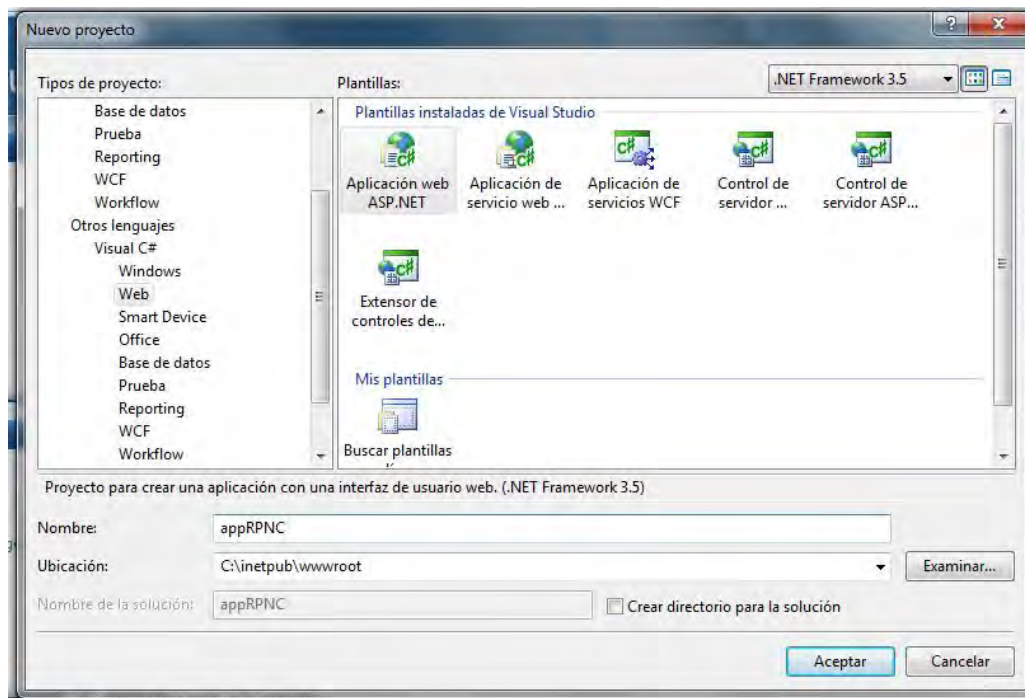


Figura 4.2. Creación de una aplicación Web de ASP.Net.

De forma automática, al crear un nuevo proyecto Web, Visual Studio .Net crea un directorio virtual en el IIS y lo asocia con la aplicación Web. Si se ha instalado IIS con la configuración por defecto, el sitio Web predeterminado (local host) será 'c:\inetpub\wwwroot'.

En el caso de que se tuviera una aplicación Web de ASP.Net ya creada y se desee instalar en un servidor Web, se debe copiar la carpeta con la aplicación en el servidor Web y asociarla manualmente a un directorio virtual. Para ello, dentro de IIS se selecciona el elemento de 'Sitio Web predeterminado' y pulsando con el botón derecho se selecciona la opción: 'Nuevo' > 'Directorio virtual' donde mediante un asistente se asocia la carpeta de la aplicación Web a un directorio virtual en el servidor.

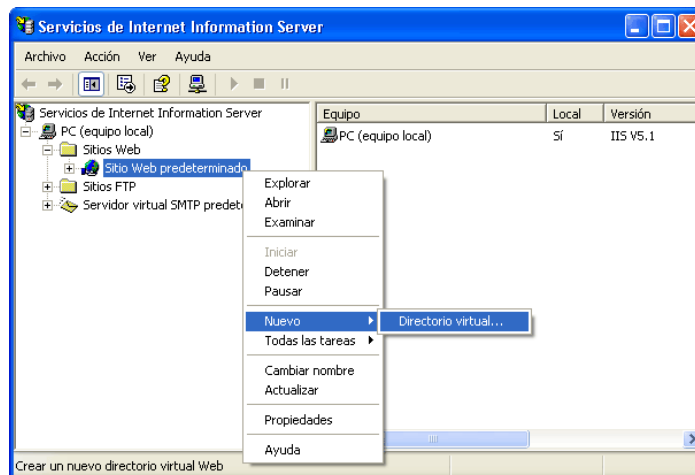


Figura 4.3. Creación de un directorio virtual en IIS.

Directorios y permisos

- ✓ Si se coloca el sitio fuera del directorio por defecto (C:\inetpub\wwwroot) tendrá que darle al directorio del sitio los mismos permisos que tenga C:\inetpub\wwwroot.
- ✓ Hay que tener en cuenta que el directorio del sitio tiene que tener los mismos permisos que C:\inetpub\wwwroot, por lo que se puede dar la situación que al mover el directorio del sitio desde otro lugar dichos permisos no se hereden automáticamente. En ese caso habría que desmarcar y marcar la opción de heredar permisos y comprobar que se actualicen.

La tabla 4.1 muestra basecamente los permisos requeridos para los directorios que vayan a contener los sitios web.

Cuenta o grupo	Permisos
Grupo EQUIPO\Administradores	Control total
Usuario EQUIPO\IUSR_EQUIPO <i>Cuenta para el acceso anónimo a los sitios web</i>	Mostrar el contenido de la carpeta
Grupo EQUIPO\IIS_WPG <i>Cuentas de usuario de identidad para procesos asociados con los grupos de aplicaciones</i>	Lectura y ejecución
	Mostrar el contenido de la carpeta
	Leer
Grupo EQUIPO\RED	Mostrar el contenido de

	la carpeta
Grupo EQUIPO\Servicio de red (IIS 6.0)	Leer y Ejecutar
	Mostrar contenido de carpeta
	Lectura
	Escritura
Grupo EQUIPO\SYSTEM	Control total
Grupo EQUIPO\Usuarios del dominio	Control total

Tabla 4.1. Permisos en los directorios.

Para probar que la aplicación Web funciona correctamente se debe compilar primero en Visual Studio .Net y posteriormente acceder a la aplicación mediante el navegador:

[http://\[Nombre del servidor\]/\[directorio virtual\]/\[página\]](http://[Nombre del servidor]/[directorio virtual]/[página])

4.1.1. Pruebas finales

Las pruebas finales juegan un papel fundamental en el desarrollo de un sitio web así como su mantenimiento a largo plazo. Mientras más pequeños sean los sitios web, especialmente aquellos con presupuestos más limitados, que no sea necesario seguir los procedimientos de pruebas formales que se requieren a gran escala, cada sitio tiene que ser probado a fondo para asegurarse de que es libre de errores, sea amigable para el usuario, accesible y compatible con las normas.

La fase de pruebas se realizará en 6 fases, las cuales identificarán posibles errores.

1) Prueba de Funcionalidad:

En este punto se revisan varios aspectos como son: pruebas a todos los enlaces existentes en la web, validación de conexión a bases de datos, formularios utilizados en las páginas web para enviar o recibir información de usuario, las pruebas de cookies.

Se revisan todos los enlaces:

- ✓ Se validan que todos los enlaces disponibles sean visibles y vistos por su correspondiente perfil
- ✓ Se analizan de todos los enlaces internos.
- ✓ Se validan los vínculos de salto en las páginas.

- ✓ Se prueban los enlaces utilizados para enviar el correo a los usuarios de administración o de otro tipo de páginas web.

Las cookies:

- ✓ Se Comprueba que no se de acceso si no existe una cookie registrada con información del usuario.
- ✓ Se valida que se almacene la información correctamente en esta.
- ✓ Se revisa el correcto acceso a esta para poder definir correctamente el perfil.

Base de datos de prueba:

Se comprueba que todas las consultas de la base de datos se ejecutan correctamente, se recuperan los datos y se actualiza correctamente.

2) Las pruebas de usabilidad:

Prueba para la navegación:

Las pruebas de usabilidad incluyen:

- ✓ El sitio web se puede usar de manera fácil.
- ✓ Las instrucciones se indican con claridad.
- ✓ El menú principal se encuentra siempre en cada página.

3) Pruebas de la interfaz:

Las principales interfaces son:

- ✓ Se valida la comunicación correcta entre todos los servidores.
- ✓ Si llega a ocurrir un error este se maneja de manera adecuada para los usuarios.

4) Prueba de compatibilidad:

- ✓ Se valida la compatibilidad con los navegadores (mozilla 2.1 +, explorer 6.0 + y safari 3.1 +).
- ✓ Se valida las opciones de impresión.

5) Control del rendimiento:

- ✓ Se valida con pruebas de estrés el funcionamiento correcto de la aplicación así como el almacenamiento y modificación de la información.

6) Pruebas de Seguridad:

- ✓ Se prueba pegando URL interna directamente en la barra de direcciones del navegador sin necesidad de registrarse, las páginas internas no se abren.
- ✓ Se prueban algunas entradas no válidas en los campos de entrada como nombre de usuario de inicio de sesión, contraseña, cajas de entrada de texto. El sistema captura dichos errores y manda notificaciones indicando la irregularidad en dichos campos.
- ✓ Se valida que los directorios web o archivos no deben ser accesibles directamente a menos que reciba opción de descarga.

4.1.2. Migración de base de datos a productivo

Para realizar el movimiento entre servidores se realizará por medio de un DTS.

DTS. Data Transformation Services.

El primer paso a seguir es truncar todas las tablas de ambiente de pruebas.

En el menú del directorio de SQL-Server, aparece una aplicación que se llama Asistente para importación y exportación de base de datos que se basa en DTS y que con un sencillo asistente basado en pasos, permite configurar esta información.



Figura 4.4. Asistente para importación/exportación.

La figura 4.4, es la segunda del asistente, pero es exactamente igual que la primera y sirve para identificar origen y destino del movimiento, ahí se pueden crear base de datos nuevas, o mover los datos a una base de datos existente, utilizando los modelos de autenticación habituales. Inmediatamente después de esta nos aparecerá una pantalla en la que tendremos tres opciones.

En esas opciones, elegiremos copiar Tablas y vistas, o bien elegir una consulta específica para elegir los datos a transferir o bien transferir objetos entre servidores SQL-server. Para esta última opción hace falta tener instalado al menos el *Service pack 2 de SQL* y algunas veces hay ciertos problemas con el modo de licenciamiento para poder realizar esta operación. Posteriormente aparece una pantalla en la que podemos elegir que objetos migrar y con qué condiciones, incluso puede personalizarse el mecanismo de migración.

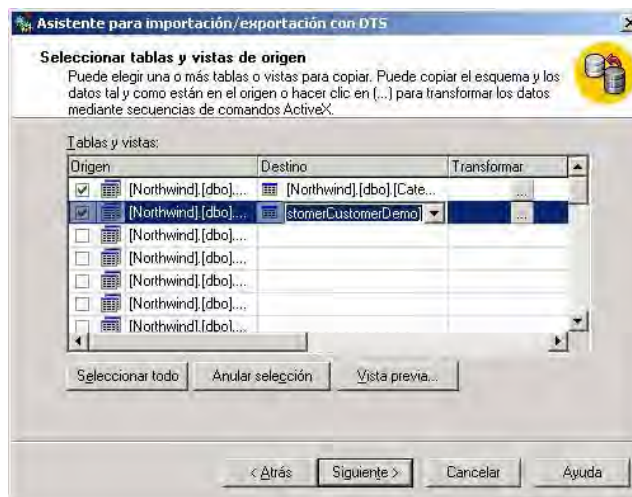


Figura 4.5. Selector de tablas y vistas.

LA figura 4.5 muestra la opción para transformar, se eligen distintas opciones, que sirven para transformar tipos de datos, asignar variables e incluso puede usarse un lenguaje de script para migrar los datos.

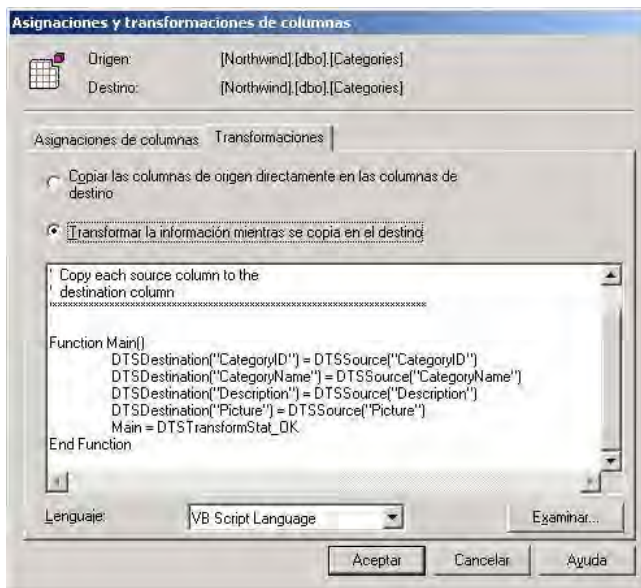


Figura 4.6. Transformación de columnas.

LA figura 4.6 muestra la forma en como se transfiere la información, en nuestro caso como no se tiene información no será necesario configurar este paso dado que no existen registros.

Después puede programarse el proceso para que se ejecute de forma inmediata o bien para que se guarde el proceso y se ejecute inmediatamente o bien se ponga a la espera del momento elegido.

También elegimos la forma de almacenamiento del paquete DTS bien en Visual Basic, bien en el propio server. En la siguiente figura se ve de forma clara como funciona esta cuestión.

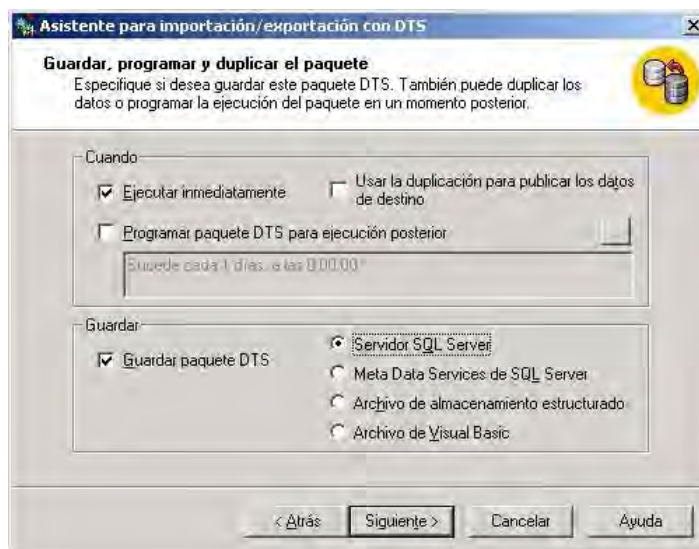


Figura 4.7. Inicio ejecución del proceso.

LA figura 4.7 muestra como se lanza el proceso que se realiza en paralelo mediante bulk insert, por lo que la inserción de filas es muy rápida.

4.1.3. Instalación del módulo WEB

Todo proyecto web en asp.net se puede componer de múltiples elementos (figura 4.8), pero al final tendremos que mostrarlo al mundo por medio de Internet.

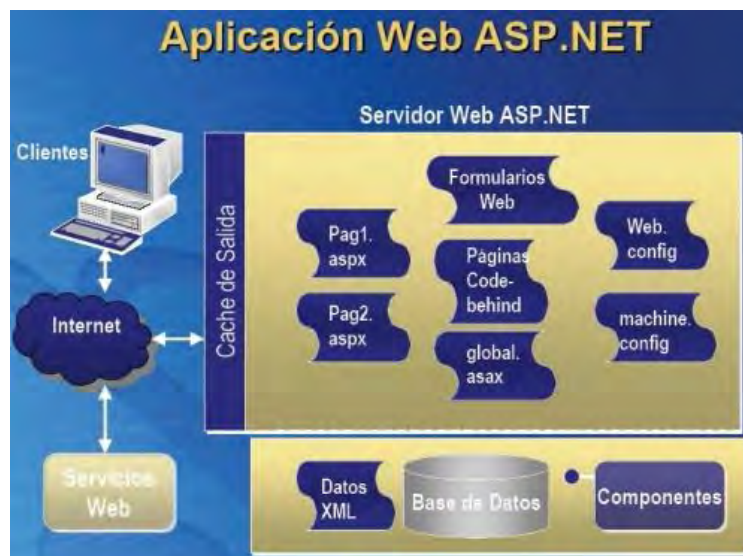


Figura 4.8. Estructura sitio web.

Como sabemos cuándo creamos un proyecto asp.net se crea un directorio virtual en el que se irán guardando todos los elementos que incorporemos al proyecto.

Para subir nuestro proyecto web a nuestro espacio web o hosting podemos utilizar cualquier cliente ftp (figura 4.9).

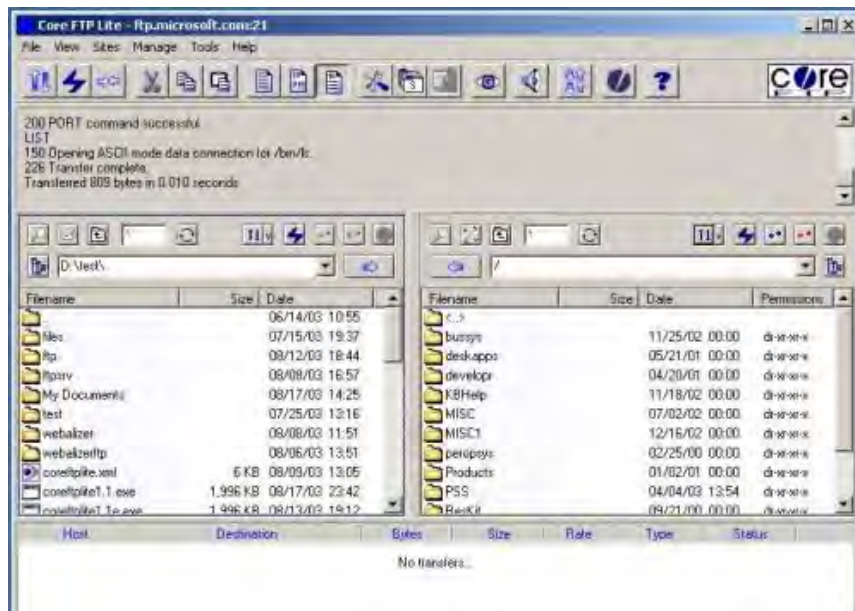


Figura 4.9. Cliente FTP.

Es obligatorio crear en el archivo raíz de nuestro hosting una carpeta llamada bin y colocar en ella la dll citada.

Los archivos básicos que debemos tener en nuestro espacio web son: la carpeta bin citada, los archivos .aspx correspondientes a las paginas, Global.asax que sirve para el control general de la aplicación, Web.config donde se establece la configuración de la aplicación. Además pueden aparecer los siguientes archivos: .asmx que son los servicios Web XML, páginas .htm ó .html, archivos .css de hojas de estilo CSS, ver figura 4.10.

Teniendo todos los archivos se realiza la copia al servidor y se procede a validar para corroborar que todo funcione correctamente.

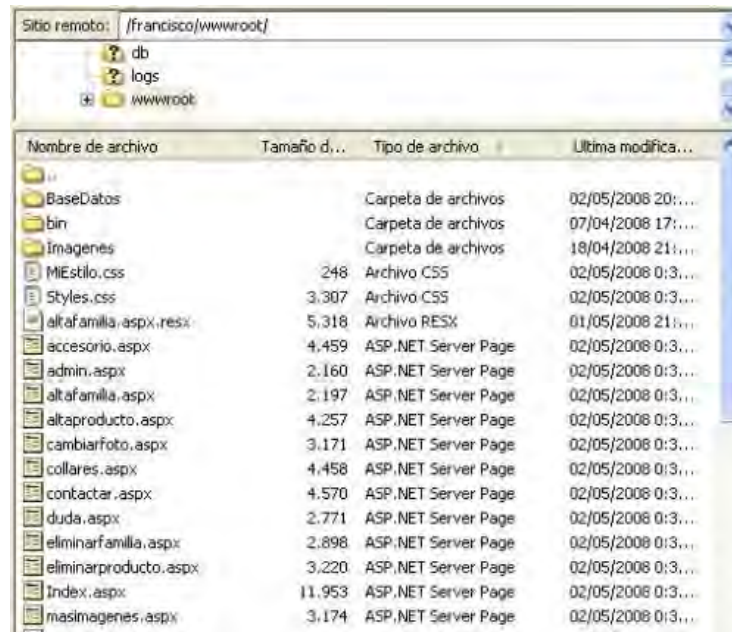


Figura 4.10. Archivos contenidos en un sitio web.

4.1.4. Configuración de la página

Para la aplicación, crearemos una carpeta llamada rechazos en C:\inetpub\wwwroot. Después de crear la carpeta, se abre la consola de Internet Information Services haciendo doble clic en Administrador de servicios de Internet en Herramientas Administrativas. Internet Information Services se muestra en la figura 4.11.

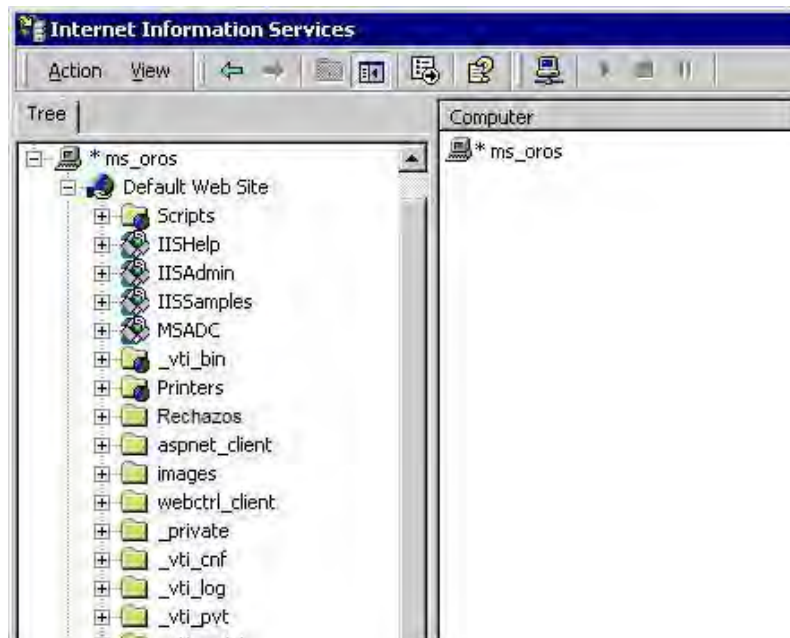


Figura 4.11. Internet Information Services en Windows 2000.

Como se ve en el panel izquierdo de Internet Information Services, la carpeta que añadimos, rechazos, se muestra como una carpeta normal debajo de sitio Web predeterminado. Encima de rechazos se pueden ver carpetas con varios iconos. El icono junto a IISHelp y otras entradas indica que estas carpetas están configuradas como carpetas virtuales.

Si se hace clic con el botón derecho en rechazos y elegimos las Propiedades del menú de accesos directos, el cuadro de diálogo Propiedades aparecerá, tal y como se muestra en la figura 4.12.

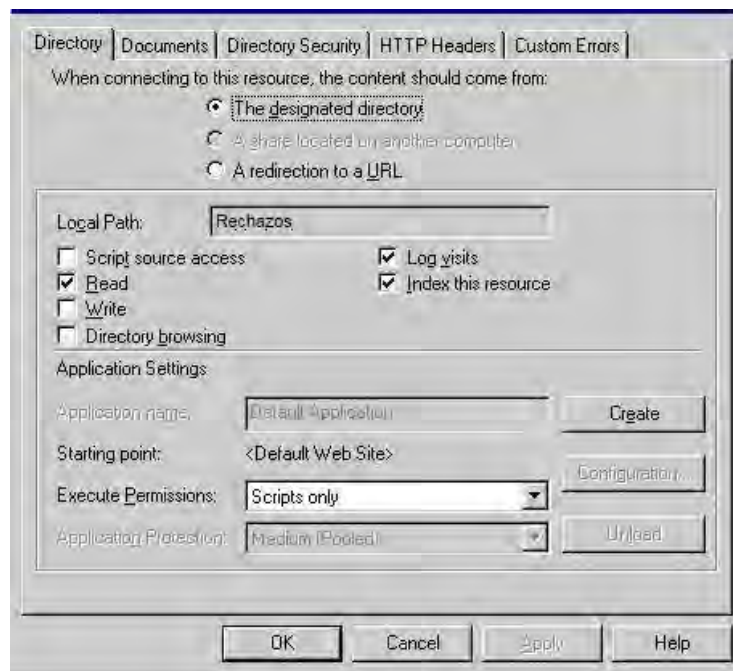


Figura 4.12. El cuadro de diálogo propiedades en Internet Information Services para la carpeta rechazos, que no es un directorio virtual.

La figura 4.12 muestra el cuadro de diálogo Propiedades tal y como aparece para un directorio que no se ha configurado como directorio virtual. Para hacer que esta carpeta sea un directorio virtual para IIS, se procede a dar clic en el botón Crear de la pestaña Directorio. La configuración de la pestaña se modificará tal y como se muestra en la figura 4.13.

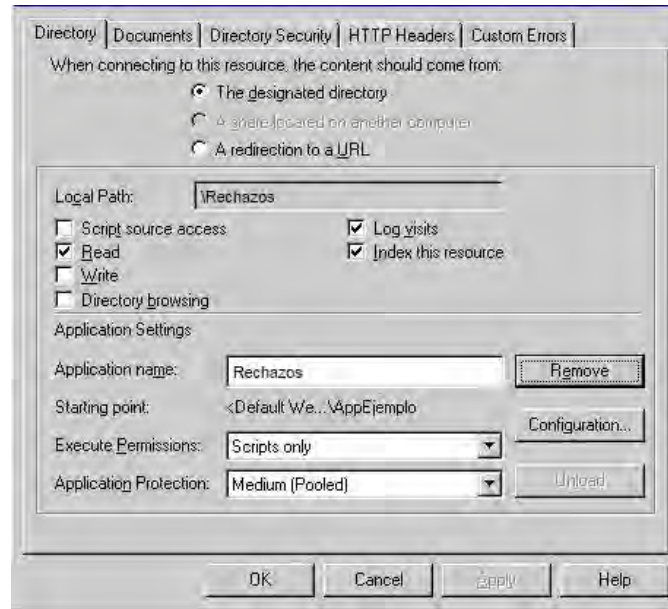


Figura 4.13. El cuadro de diálogo propiedades en Internet Information Services para la carpeta rechazos después de haberla convertido en un directorio virtual.

La pestaña Documentos del cuadro de diálogo se muestra en la figura 4.14.

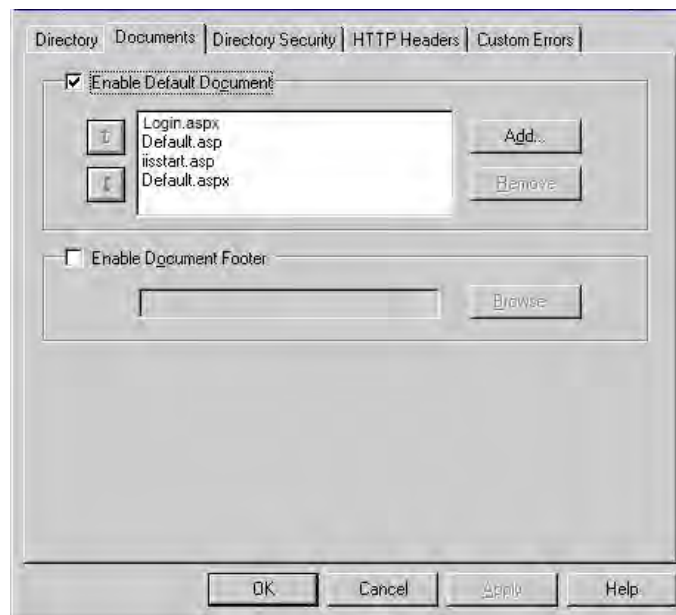


Figura 4.14. La pestaña documentos del cuadro de diálogo propiedades.

Se asignará el documento login.aspx el cual será asignado por defecto en la pestaña Documentos, cuando el usuario va a la carpeta y especifica únicamente el nombre de la carpeta (por ejemplo, http://localhost/Rechazos), ese documento se mostrará.

La pestaña Seguridad de directorios del cuadro de diálogo Propiedades se muestra en la figura 4.15.

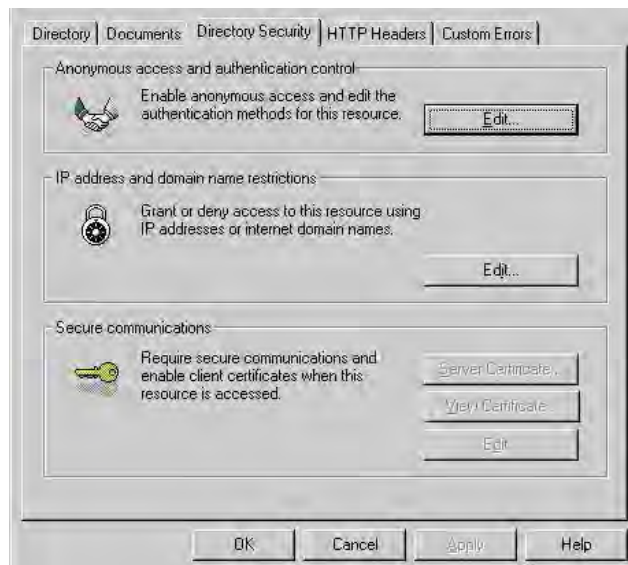


Figura 4.15. La pestaña seguridad de directorios del cuadro de diálogo propiedades.

La sección de Control de Autenticación y Acceso Anónimo de esta pestaña permitirá controlar cómo se autentican los usuarios en el directorio virtual. Si se hace clic en el botón Modificar se muestra el cuadro de diálogo Métodos de Autenticación, mostrado en la figura 4.16.



Figura 4.16. Cuadro de diálogo de métodos de autenticación.

Se tienen pocas opciones disponibles pero, básicamente, este cuadro de diálogo te permite hacer un par de cosas: o permitir el acceso anónimo al directorio virtual o exigir que los usuarios sean usuarios de Windows.

Para este sistema la selección no se hace, ya que todos los usuarios tendrán acceso a la aplicación sin necesidad de que suceda ninguna autenticación.

El botón de configuración de la pestaña Directorio del cuadro de diálogo Propiedades de Rechazos (figura 4.13) permite controlar ciertos aspectos de la aplicación así como resolver problemas con la instalación ASP.NET. Cuando hace clic en Configuración, aparece el cuadro de diálogo Configuración de Aplicación, como se muestra en la figura 4.17.

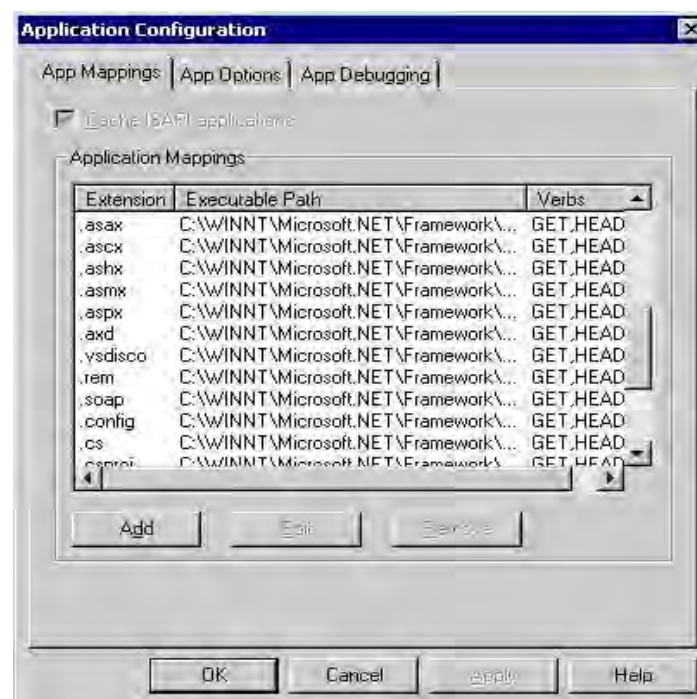


Figura 4.17. Cuadro de diálogo de Configuración de aplicación.

Al desplazarse hacia abajo en la lista de correspondencias con aplicaciones de forma que se puedan ver las extensiones empleadas por ASP.NET. En este cuadro de

diálogo, puede añadir o editar la ruta del ejecutable asociada con cada extensión de archivo.

4.1.5. Configuración DTS para el envío de correo

Para esta aplicación se desarrollo un DTS el cual tiene como finalidad enviar notificaciones respecto a los rechazos que tienen pendientes para elevar el tiempo de respuesta en cada uno.

El primer paso es iniciar el asistente para la configuración de un nuevo trabajo el cual ejecutara el DTS periódicamente (figura 4.18).



Figura 4.18. Ejecutar DTS.

Se asigna el nombre al nuevo trabajo (figura 4.19).

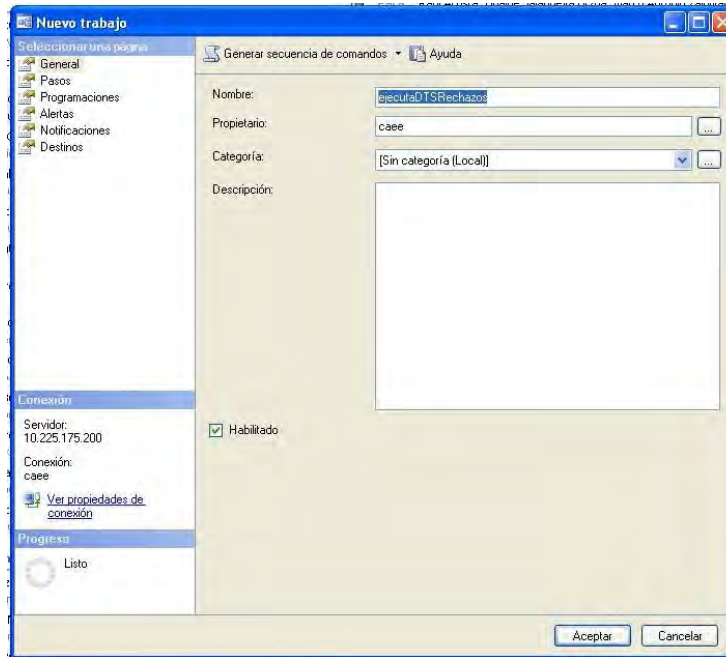


Figura 4.19. Configuración nombre DTS.

Se crea un nuevo paso, el cual tendrá como finalidad ejecutar el DTS desde SQL, se establecerá un nombre, el tipo de lenguaje que se utilizar, la base de datos a la cual se hará referencia y el comando (figura 4.20).

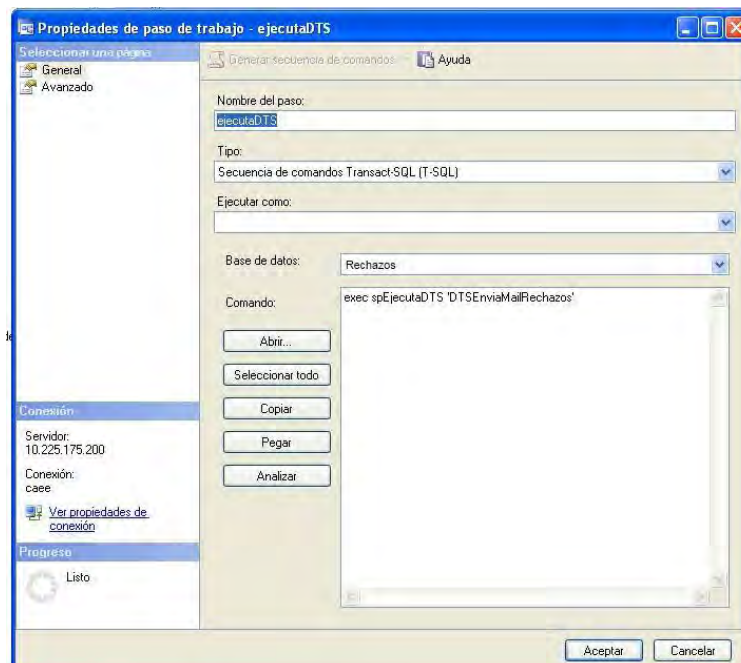


Figura 4.20. Configuración del proceso a ejecutar.

Al término se da click en aceptar y se visualizara la información del paso creado (figura 4.21).

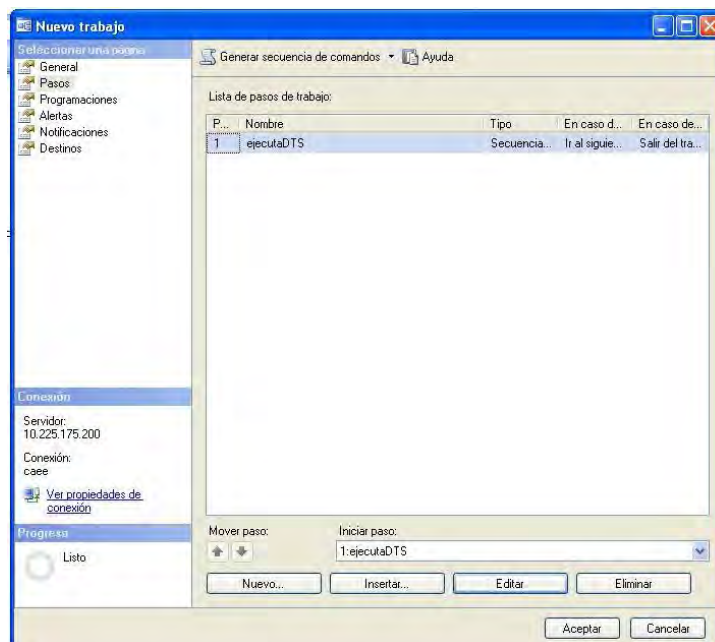


Figura 4.21. Termino de configuración proceso.

Terminado este paso se procede a ir a la pestaña de programaciones en la cual se configurarán la siguiente información (figura 4.22).

- ✓ Nombre.
- ✓ Frecuencia con que sucede (diario).
- ✓ Repetición (1 vez al día).
- ✓ La hora en que se ejecutara (10:00 am, 12:00 am y 04:00 pm).

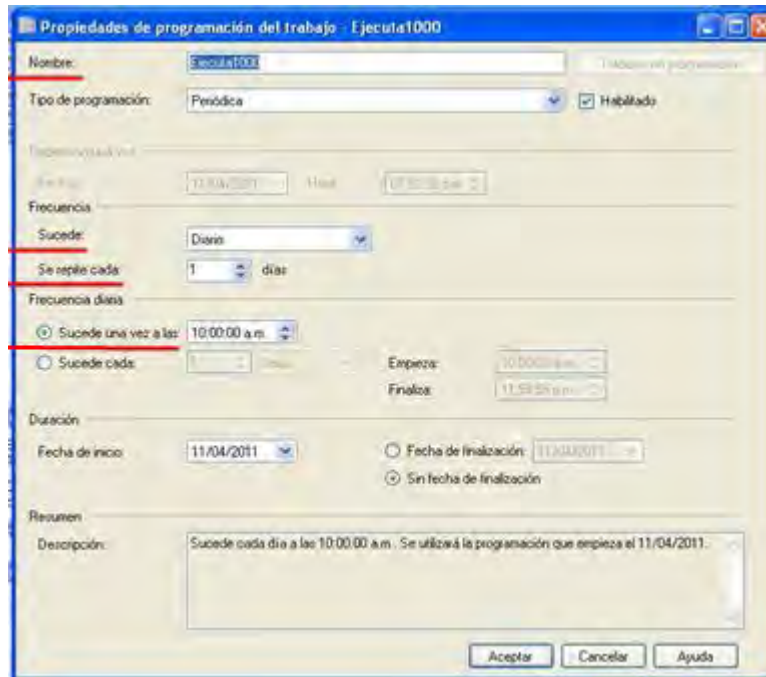


Figura 4.22. Configuración en la programación del DTS.

Con esto se finaliza la configuración, se guardan los cambios, se finaliza, y en automático se lanzarán los procesos en las horas establecidas.

4.1.6. Liberación del sistema a productivo con los usuarios

Una vez configurada la página y el DTS se libera el sistema a todos los involucrados para hacerles conocer el sistema y se realiza la presentación en 3 fases

1. Presentación del sistema.- Se reúne a todas las áreas, se les hace saber del sistema, además de que se les explica el funcionamiento básico de la aplicación, así como los perfiles existentes y los menús que visualizarán cada uno.

Se realiza una presentación con información general de cada formulario pretendiendo describir su funcionamiento y su lógica para el posterior manejo.

2. Primeras pruebas con un caso real.- Se captura la información de distintos rechazos en distintas situaciones con la finalidad de que el usuario conozca la interfaz, se asocie a esta y se resuelvan posibles dudas respecto al funcionamiento.

Se realiza el seguimiento de cada caso siguiendo el flujo que seguirá y se valida que cada área reciba la información correspondiente a cada folio y se valida a detalle todo el proceso.

3. Solicitar los usuarios y dar por terminada la aplicación.- Se solicitan todos los usuarios que tendrán acceso a la aplicación y sus perfiles, para realizar una carga masiva, en ese momento en automático se notificará al usuario que se ha dado de alta y se le enviará una contraseña aleatoria la cual tendrá que modificar al primer ingreso de la aplicación, terminado este paso se libera por completo el sistema.

CONCLUSIONES

Gracias al desarrollo de este caso práctico se pudo implementar un sistema integral que cumplió con las especificaciones señaladas con el fin de mejorar el proceso de producto no conforme, permitiendo agilizar y optimizar los procesos que involucran el flujo de éste, así como una considerable mejoría en los siguientes puntos:

- ✓ Disminución de tiempos al momento de crear el rechazo.
- ✓ Mejora en el seguimiento de cada rechazo.
- ✓ Optimización de búsquedas por distintos rubros.
- ✓ Mejor control de documentos generados en cada rechazo.
- ✓ Seguridad de la integridad de los datos.
- ✓ Generación de reportes íntegros los cuales reflejaran información útil para todas las áreas involucradas
- ✓ Mejora en la captura de nuevos rechazos, así como las autorizaciones implicadas en las áreas.

Durante el desarrollo y documentación del software, me pude dar cuenta que la programación es uno de los pilares fundamentales de la ingeniería en computación, al cual se le dedican muchas horas de esfuerzo para poder entregar un sistema con calidad y funcional para los usuarios.

En la actualidad la tecnología va avanzando, van apareciendo nuevas soluciones, nuevas formas de programación, nuevos lenguajes y un sin fin de herramientas que intentan realizar el trabajo del desarrollo un poco más fácil y novedoso, con controles cada día mas potentes, atractivos, funcionales y lo más importante es que van encaminados a ser multiplataforma, permitiendo así poder desarrollar de manera universal y poder llegar a distintos usuarios sin importar las características del sistema que estos utilicen.

Por otra parte el desarrollo de páginas web hoy en día se ha vuelto un factor importante ya que cada vez se encuentran diferentes usos, tanto personales como profesionales, la necesidad de estar conectado la mayor parte del tiempo se incrementa, con el fin de dar información actualizada, llegar a un número mayor de personas, poder realizar cambios sin la necesidad de actualizar cada máquina, captar más clientes y poder tener un contacto permanente con clientes.

GLOSARIO

Rechazo: Producto que no cumple con las especificaciones, no pueden ser recuperados, por lo que es desechado y puesto a disposición del comprador.

Reproceso: Producto que puede ser reprocesado para satisfacer las especificaciones.

Reparación: Producto que requiere la realización de una operación especial, diferentes a las del plan de calidad, para adecuarlo a las especificaciones.

Usado como esta: Esta disposición se toma cuando la desviación en el producto o dimensiones del componente no afecta su funcionamiento.

Reclasificado como: Producto que puede ser identificado y usado como otro tipo de producto

Producto no conforme: Resultado de un proceso que no cumple con los requisitos especificados.

Conformidad: Cumplimiento de un requisito.

No Conformidad: Incumplimiento de un requisito.

Acción Correctiva: Acción tomada para eliminar la causa de una no conformidad detectada u otra situación indeseable.

Corrección: Acción tomada para eliminar una no conformidad detectada.

Nota 1: Una corrección puede ser, por ejemplo, un reproceso o una reclasificación.

Inspección: Evaluación de la conformidad por medio de la observación y dictamen, acompañada cuando sea apropiado por medición, ensayo/prueba o comparación con patrones.

Liberación: Autorización para proseguir con la siguiente etapa de un proceso

Proceso: Conjunto de actividades mutuamente relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados

Proveedor: Organización o persona que proporciona u producto.

Servicio: Son actividades, beneficios, o satisfacciones que se ofrecen en venta al consumidor; su producción puede o no puede vincularse a un producto físico. Por lo general, se producen y se consumen al mismo tiempo, no pueden almacenarse.

Número PO (Purchase Order): Es el número generado por el documento de orden de compra.

Archivos de Respaldo o Documentos: conjunto de documentos imprescindibles para una posterior justificación de las decisiones tomadas.

Desperdicio: Todo aquel material sobrante o defectuoso, obtenido de cualquier proceso.

Número IR (Invoice Receipt): Proporcionado por la factura que se genera al momento de recibir algún material.

Número GR (Good Receipt): Proporcionado por la factura que se genera al momento de recibir algún material.

NC (Nota de Crédito): Es un documento por el cual una empresa comunica a su cliente haberle disminuido su deuda o que ha registrado a su favor cierta cantidad por el motivo que ella expresa, esta se emite para rebajar el valor de una factura.

SCRAP: Hoja generada por el área correspondiente, la cual contendrá información respecto al desperdicio que se genero de un rechazo.

.NET: Es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

AJAX: es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Framework: En términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

HTML: es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>).

XML: es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

DOM: es esencialmente una interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.

REFERENCIAS BIBLIOGRÁFICAS

7. Paul Harmon y Mark Watson (1998). *Entendiendo UML: La guía del desarrollador, con una aplicación java basada en web*. Morgan Kauffman Publishers, Inc. Recuperado el 14 de enero de 2012. Disponible en www.mkp.com/books_catalog/1-55860-465-0.asp.
8. BOOCH Grady et al (1999). *El Lenguaje Unificado de Modelado (1^{ra} Ed.)*. Editorial Addison Wesley.
10. Craig, Larman (2002). *UML y Patrones Una introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado (2^{da} Ed.)*. Editorial Prentice Hall.
13. C. Louden, Kenneth (2^{da} Ed.)(2004). *Lenguajes de programación: principios y practicas (pp. 20-69)*. Editorial Cengage Learning Editores.
15. Foro Devjoker (2011). *Microsoft .NET*. Recuperado el 26 Julio de 2011. Recuperado en <http://www.devjoker.com/html/Introducci%C3%B3n-a-NET.html>.
21. El guille (2010). *Funcionalidad de iTextSharp usando Visual C#*. Recuperado el 26 de julio de 2011. Disponible en http://www.elguille.info/colabora/NET2005/Percynet_Funcionalidad_itextsharp_con_csharp.htm
22. Álvarez, Miguel Angel (2009). *Introducción a jQuery*. Recuperado el 12 de Octubre 2011. Disponible en <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>
25. Business Development Software (2006). *Web Services: interconexión de aplicaciones web*. Recuperado el 25 de Octubre 2011. Recuperado en <http://www.proyectosbds.com/servicios-tecnologicos-para-instituciones-publicas/web-services-interconexion-de-aplicaciones-web/96/>
26. Golomb, Ernesto (2007). *Introducción a Visual Studio*. Recuperado el 12 de Octubre de 2011. Recuperado en <http://msdn.microsoft.com/es-es/library/6b6b1f4%28v=vs.80%29.aspx>
27. Fernando Berzal, Francisco José Cortijo y Juan Carlos Cubrero (1^{ra} Ed.) (2003). *Desarrollo profesional de aplicaciones web con asp.net (pp 36-56)*.
28. SlideShare (2010). *Microsoft visio*. Recuperado el 25 de Octubre 2011. Recuperado en <http://www.slideshare.net/cikamelomicrosoft-visio-4502404>
29. Post, José Antonio (2011). *Erwin Data Modeler*. 28 de Octubre 2011. Recuperado en www.taringa.net/posts/taringa/.../Erwin--Data-Modeler-v-7_1.html
35. Ramirez, Edgar (2008). *IIS (Internet Information Server)*. 25 de Octubre 2011. Recuperado en <http://edgarramirez.wordpress.com/2008/08/19/iis-internet-informati-on-server/>
36. EcuRed (2005). *IIS*. 25 de Octubre 2011. Recuperado en <http://www.ecured.cu/index.php/IIS>
37. Flores, Antonio (1^{ra} Ed.) (2006), *Introducción a SQL Server (pp 12-20)*.

BIBLIOGRAFIA

- ✓ **Alarcón**, Raúl (1^{ra} Ed.)(2000). *Diseño Orientado a Objeto usando UML*. Editorial Eidos.
- ✓ **Berzal**, Fernando (1^{ra} Ed.) (2003). *Desarrollo profesional de aplicaciones web con asp.net*.
- ✓ **Booch**, Grady (1999). *El Lenguaje Unificado de Modelado (1^{ra} Ed.)*. Editorial Addison Wesley.
- ✓ **C. Louden**, Kenneth (2^{da} Ed.)(2004). *Lenguajes de programación: principios y practicas*. Editorial Cengage Learning Editores.
- ✓ **Craig**, Larman (2002). *UML y Patrones Una introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado (2^{da} Ed.)*. Editorial Prentice Hall.
- ✓ **Flores**, Antonio (1^{ra} Ed.) (2006), *Introducción a SQL Server*.
- ✓ **Fowler**, Martin (2011). *Patterns of Enterprise Application Architecture*. Editorial Pearson Education.
- ✓ **Hohpe**, Gregor (2011). *Enterprise Integration*. Editorial Pearson Education
- ✓ **Tuya**, Javier (1^{ra} Ed.)(2005). *Técnicas cuantitativas para la gestión en la ingeniería del software*. Editorial Editores.

MESOGRAFIA

- ✓ **Alegsa** (2006). *¿Qué es un lenguaje de programación?*. 23 de Octubre del 2011. Disponible en: <http://www.alegsonline.com/art/11.php>
- ✓ **Álvarez, Miguel Angel** (2009). *Qué es jQuery, para qué sirve y qué ventajas tiene el utilizar este framework Javascript*. 24 de Octubre del 2011. Disponible en: <http://www.desarrolloweb.com/articulos/introduccion-jquery.html>
- ✓ **Angelfire** (2003). *Base de datos relacional*. 12 Octubre del 2011. Disponible en: <http://www.angelfire.com/crazy3/eirita/Investigacion.1.htm>
- ✓ **Barranquillo** (2005), *UML*. 28 de Octubre de 2011. Disponible en: <http://barranquillo.ucaldas.edu.co/rgarcia/ingsoft/UML.htm>
- ✓ **Bermeo Pérez, Fabián** (2010). *Metodología RUP - desarrollo de software de calidad*. 24 de Noviembre del 2011. Disponible en: <http://fabianbermeop.blogspot.com/2010/12/metodologia-rup-desarrollo-de-software.html>
- ✓ **BDS - Business Development Software S.L** (2005). *Web Services: interconexión de aplicaciones web*. 24 de Octubre del 2011. Disponible en: <http://www.proyectosbds.com/servicios-tecnologicos-para-instituciones-publicas/web-services-interconexion-de-aplicaciones-web/96/>
- ✓ **Chuidiang** (2007). *Metodología de desarrollo software*. 24 de Noviembre del 2011. Disponible en: <http://www.chuidiang.com/ood/metodologia/metodologia.php>
- ✓ **Cubero, Juan Carlos** (2007). *Desarrollo Profesional de Aplicaciones Web con ASP.NET*. 24 de Octubre del 2011. Disponible en: <http://elvex.ugr.es/decsai/csharp/pdf/web/web-book-b5.pdf>
- ✓ **ENI** (2003). *UML 2: Iniciación, ejemplos y ejercicios corregidos*. 12 de Noviembre de 2011. Disponible en: http://books.google.com.mx/books?id=drst-kOWZWSc&pg=PA24&lpg=PA24&dq=principios+rup&source=bl&ots=opd24wTHx_&sig=U6dB6BXC9LShrKIVDwzZCRm6EYo&hl=es&ei=2hTsTa3nL466sQPSptnvDQ&sa=X&oi=book_result&ct=result&resnum=5&ved=0CC0Q6AEwBDgU#v=onepage&q=principios%20rup&f=false
- ✓ **eumed.net** (2005). *Porque utilizar RUP para desarrollar aplicaciones web*. 12 de Noviembre de 2011. Disponible en: <http://www.eumed.net/libros/2009c/584/Por%20que%20utilizar%20RUP%20para%20desarrollar%20aplicaciones%20web.htm>
- ✓ **González Seco, José Antonio** (2006). *Microsoft .NET*. 23 de Octubre del 2011. Disponible en: <http://www.devjoker.com/html/Introducci%C3%B3n-a-NET.html>
- ✓ **Gracia, Joaquin** (2003). *UML: Casos de Uso*. 23 de Noviembre del 2011. Disponible en: <http://www.ingenierosoftware.com/analisisydiseno/casosdeuso.php>

- ✓ **Microsoft** (2011). *Introducción a Visual Studio*. 24 de octubre del 2011. Disponible en: <http://msdn.microsoft.com/es-es/library/6x6bk1f4%28v=vs.80%29.aspx>
- ✓ **Monografías** (2008). *Diseño y Modelación de un Proyecto de Software. Utilizando el lenguaje UML*. 24 de Octubre del 2011. Disponible en: <http://www.monografias.com/trabajos24/software-uml/software-uml.shtml>
- ✓ **Quiroz Martinez, Diego** (2005). *Proceso Software dirigido por las pruebas*. 28 de Octubre 2011. Disponible en: <http://es.scribd.com/doc/56480382/2/II-Modelado-de-Software-con-UML>.
- ✓ **Reyes Paredes, Percy** (2005). *Funcionalidad de iTextSharp usando Visual C#*. 24 de Octubre del 2011. Disponible en: http://www.elguille.info/colabora/NET2005/Percynet_Funcionalidad_itextsharp_con_csharp.htm
- ✓ **Salinas Caro, Patricio** (2002). *Casos de Uso*. 24 de Noviembre del 2011. Disponible en: <http://www.dcc.uchile.cl/~psalinas/uml/casosuso.html>
- ✓ **Softtlan** (2007). *Tutorial de casos de uso pasó*. 23 de Noviembre del 2011. Disponible en: <http://softtlan.blogspot.com/2007/03/tutorial-de-casos-de-uso-paso.html>
- ✓ **Yaqui** (2002). *El Proceso Unificado de Desarrollo de Software (RUP)*. 23 de Noviembre del 2011. Disponible en: <http://yaqui.mxl.uabc.mx/~molquin/as/RUP.htm>

ANEXO I

1. Código

1.1. Login del sistema (login.aspx)

El siguiente código muestra las funciones que están contenidas en la pantalla del login para llevar a cabo la autenticación del usuario y la descarga del manual de usuario

```
Imports System.Data.SqlClient
Imports System.IO
```

```
Partial Public Class Login
    Inherits System.Web.UI.Page
```

```
    'Evento que valida los datos del usuario
    Protected Sub btnAceptar_Click(ByVal sender As Object, ByVal e As EventArgs) Handles
    btnAceptar.Click
        If Page.IsValid = True Then
            Try
                'Llamamos a la funcion que realizará la validacion del usuario
                ds = validaUsuario(txtUsuario.Text, txtContraseña.Text)
                'Validamos que exista una tabla con los datos del usuario
                If ds.Tables(0).Rows.Count > 0 Then
                    'El usuario existe, creamos una cookie para almacenar los valores del usuario
                    cookie = New HttpCookie("rechazo")
                    'Agregamos los valores que contendra la cookie
                    cookie.Values.Add("id_user", ds.Tables("Login").Rows(0).Item("id_user"))
                    cookie.Values.Add("Nombre", ds.Tables("Login").Rows(0).Item("nombre"))
                    cookie.Values.Add("Area", ds.Tables("Login").Rows(0).Item("idarea"))
                    cookie.Values.Add("NomArea", ds.Tables("Login").Rows(0).Item("descripcion"))
                    cookie.Values.Add("roll", ds.Tables("Login").Rows(0).Item("idroll"))
                    cookie.Values.Add("cambioPass", ds.Tables("Login").Rows(0).Item("cambioPass"))
                    cookie.Values.Add("edicion", ds.Tables("Login").Rows(0).Item("edicion"))
                    'Registramos la cookie en el explorador
                    Response.AppendCookie(cookie)
                    'Limpiamos el data set
                    ds.Tables.Remove("Login")
                    'enviamos a la pagina de inicio
                    Response.Redirect("frmPrincipal.aspx")
                Else
                    'Si no existe el usuario enviamos una leyenda
                    lblMensaje.Text = "Nombre de Usuario o Contraseña Invalido"
                    'Limpiamos el data set
                    ds.Tables.Clear()
                End If
            Catch ex As Exception
                'Si llegase a surgir un error durante la ejecucion se llama a la clase cerror para
                almacenar
                'la descripcion del error y la pagina donde surgio, el cual de volvera un codigo de error al
                usuario
                'para su rapida identificacion
                Dim ce As cError = New cError()
                Dim cError As String
                cError = ce.almacenaError(ex.Message.ToString(), "login.aspx")
            End Try
        End If
    End Sub
End Class
```

```
        lblMensaje.Text = "Comuníquese con su Administrador y proporcione el siguiente código  
de error " + cError  
    End Try  
    End If  
End Sub
```

```
'Evento que se ejecuta cuando el usuario desea realizar una nueva consulta  
Protected Sub btnCancelar_Click(ByVal sender As Object, ByVal e As EventArgs) Handles  
btnCancelar.Click  
    'Limpiamos los valores contenidos en los controles  
    txtContraseña.Text = ""  
    txtUsuario.Text = ""  
    lblMensaje.Text = ""  
End Sub
```

```
'Evento que se ejecuta cuando el usuario desea descargar el manual de usuario  
Protected Sub btnManual_Click(ByVal sender As Object, ByVal e As EventArgs) Handles  
btnManual.Click  
    'Se inicia la descarga del manual de usuario  
    Pup("999999999", "Manual Usuario Rechazos.doc", "MANUAL")  
End Sub
```

```
'Función que configura los datos para iniciar la descarga del manual de usuario  
Sub Pup(ByVal rejection As String, ByVal lbx As String, ByVal frm As String)  
    Dim scp As String  
    'Variable que define las características que tendrá el popup que descargará el manual de  
usuario  
    Dim opc As String = "toolbar=0,resizable=1"  
    scp = "<script language=JavaScript>" & vbCrLf  
    scp += "var vt=window.open('PopUp.aspx?a=" & lbx & "&r=" & rejection & "&f=" & frm & "',''" &  
opc & ");" & vbCrLf  
    scp += vbCrLf  
    scp += "<" & "/script">  
    'registramos el script en la página para que se ejecute dicho script  
    If (Not Me.IsStartupScriptRegistered("Startup5")) Then  
        Me.RegisterStartupScript("Startup5", scp)  
    End If  
End Sub
```

```
'Función que se ejecuta cada vez que se realiza la carga de una página  
Protected Sub Page_Load(ByVal sender As Object, ByVal e As EventArgs) Handles Me.Load  
    'Validamos que no se realice un postback en la página  
    If Not Page.IsPostBack Then  
        'Buscamos todos los documentos generados en las consultas de rechazos  
        Dim dirs As String() = Directory.GetFiles("C:\inetpub\wwwroot\Rechazos\Temporales\  
*.pdf")  
        Dim dir As String  
        'Recorremos los archivos recuperados  
        For Each dir In dirs  
            Try  
                'Eliminamos archivo  
                File.Delete(dir)  
            Catch ex As Exception  
                'El archivo está en uso y no se puede eliminar  
            End Try  
        Next  
    End If  
End Sub
```

```
Next
End If
cierraconexion()
End Sub

'Funcion que realiza la validacion de un u suario
Function validaUsuario(ByVal usuario As String, ByVal contraseña As String) As DataSet
    'Creamos los objetos que utilizaremos para realizar la consulta
    Dim sqlcmd As SqlCommand = Nothing
    Dim sqladp As SqlDataAdapter = Nothing
    Dim sqlcnx As SqlConnection = Nothing
    'Creamos un uevo data set para almacenar los datos devueltos por la consulta
    Dim dsu As DataSet = New DataSet

    Try
        'Asignamos la cadena de conexion al objeto creado
        sqlcnx = New
SqlConnection(ConfigurationManager.ConnectionStrings("sqlcnx").ConnectionString)
        'Creamos un nuevo comando para asignar el store y proporcionar los parametros
        sqlcmd = New SqlCommand()
        sqlcmd.CommandText = "spValidaUsuario"
        sqlcmd.CommandType = CommandType.StoredProcedure
        sqlcmd.Parameters.Add("@usuario", SqlDbType.VarChar).Value = usuario
        sqlcmd.Parameters.Add("@contraseña", SqlDbType.VarChar).Value = contraseña
        sqlcmd.Connection = sqlcnx
        'Asignamos el command al adapter, el cual recuperara los datos devueltos por el query
        sqladp = New SqlDataAdapter(sqlcmd)
        'Guardamos los resultados devueltos en un data set, para posteriormente manipular los
datos
        sqladp.Fill(dsu, "usuario")
        sqlcnx.Close()

        Return dsu

    Catch ex As Exception
        'Si llegase a surgir un error durante la ejecucion se llama a la clase cerrar para almacenar
        'la descripcion del error y la pagina donde surgio, el cual de volvera un codigo de error al
usuario
        'para su rapida identificacion
        Dim ce As cError = New cError()
        Dim cError As String
        cError = ce.almacenaError(ex.Message.ToString(), "login.aspx")
        sqlcnx.Close()
        Return dsu
    End Try

End Function
End Class
```

1.2. Validación de la creación correcta de la cookie en las páginas web.

El siguiente código muestra la forma en que se valida la creación de la cookie para poder tener acceso a la pagina solicitada.

```
'Codigo que se ejecuta cuando se carga la pagina
```

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    'Validamos que no se este realizando un post back
    If Not Page.IsPostBack Then
        'Creamos una variable, la cual contendra el identificador del usuario
        Dim user As String
        'Llamamos a la funcion checa galleta, pasando como parametros la cookie y el valor
        'al que queremos acceder
        user = cc.ChecaGalleta(Request.Cookies("rechazo"), "id_user")
        'Validamos si la variable se encuentra vacia, realizamos un redirect a la pantalla de login
        If user = "" Then
            Response.Redirect("login.aspx")
        End If
    End If
End Sub
```

```
'funcion para verificar la Cookie que regresa parametros de tipo cadena
Public Function ChecaGalleta(ByVal NomGalleta As HttpCookie, ByVal Campo As String) As String
    'Validamos que este vacia o null la cookie
    If Not IsNothing(NomGalleta) Then
        'Recuperamos el valor solicitado
        Return CStr(NomGalleta.Item(Campo))
    Else
        'No existe la cookie, por tanto regresamos un valor vacio
        Return ""
    End If
End Function
```

1.3. Asignación del menú.

El siguiente código muestra la forma en que se asigna el menú dependiendo de roll que tenga cada usuario

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    If Not Page.IsPostBack Then
        'Validamos exist una cookie creda
        If clas1.ChecaGalleta(Request.Cookies("rechazo"), "id_user") = "" Then
            Response.Redirect("login.aspx")
        End If

        'Asignamos en una etiqueta el nombre del area que accesa al sistema
        lblArea.Text = clas1.ChecaGalleta(Request.Cookies("rechazo"), "NomArea")
        'Asignamos a una etiqueta el nombre del usuario que ingreso
        lblUsuario.Text = clas1.ChecaGalleta(Request.Cookies("rechazo"), "Nombre")
        Dim frm_menu As HtmlControl = Me.FindControl("f1")

        'Validamos el roll que tiene el usuario
        Select Case clas1.ChecaGalleta(Request.Cookies("rechazo"), "roll")
            Case 1
                'Administrador
                'validamos si tiene configurado la visibilidad del menu como administrador (todo),
                chkAdministrador.Visible = True
                If Request.QueryString("v") = "1" Then
                    chkAdministrador.Checked = True
                End If
            End Select
    End If
End Sub
```

```

Else
    'de lo contrario solo vera el menu que tiene cualquier usuario con el
    'perfil asignado
    frm_menu.Attributes("src") = "frmPanel.aspx"
    Dim r As New Roll
    'Definimos el menu que visualizara el usuario
    r.menu(Menu1, clas1.ChecaGalleta(Request.Cookies("rechazo"), "Area"))
    modificaCookie(2)
End If
Case 2
    'Escritura
    If Not Request.QueryString("v") Is Nothing Then
        chkAdministrador.Visible = True
    End If
    frm_menu.Attributes("src") = "frmPanel.aspx"
    Dim r As New Roll
    'Definimos el menu que visualizara el usuario
    r.menu(Menu1, clas1.ChecaGalleta(Request.Cookies("rechazo"), "Area"))
Case 3
    'Lectura
    'Eliminamos lo que no podra visualizar
    Menu1.Items.Remove(Menu1.FindItem("Usuarios"))
    Menu1.Items.Remove(Menu1.FindItem("Seguimiento"))
    Menu1.Items.Item(0).ChildItems.Remove(Menu1.FindItem("Rechazo/Nuevo"))
    frm_menu.Attributes("src") = "frmPanel.aspx"
End Select

'Si tiene activado el cambio de passwor redirigimos a la pagina que realizará
'dicho cambio
If clas1.ChecaGalleta(Request.Cookies("rechazo"), "cambioPass") = "True" Then
    frm_menu.Attributes("src") = "frmAltaUsuarios.aspx?id=" &
clas1.ChecaGalleta(Request.Cookies("rechazo"), "id_user")
    'Response.Redirect("frmAltaUsuarios.aspx")
End If

'Si el usuario tiene privilegios de edicion, habilitamos el menu
If Not clas1.ChecaGalleta(Request.Cookies("rechazo"), "edicion") = True Then
    Menu1.Items.Item(0).ChildItems.Remove(Menu1.FindItem("Rechazo/Editar"))
End If
End If
End Sub

Public Sub menu(ByVal men As Menu, ByVal area As String)
    Dim item As String
    'Creamos el query que devolvera a que menu tiene acceso dependiendo del area
    query = "SELECT cer, uno,dos,tres,cua,cin,sei,sie,och,nue,die,onc,doc,trec,cat " & _
"FROM Areas WHERE idarea = " & area
    sql_adp = New SqlDataAdapter(query, sqlcnx)
    sql_adp.Fill(ds, "ROLL")
    sqlcnx.Close()
    men.Items.Remove(men.FindItem("Usuarios"))
    'Recorremos las 14 posiciones existentes contenidas en el menu
    For i = 0 To 14
        'Validamos que el valor devuelto no sea nulo
        If IsDBNull(ds.Tables("ROLL").Rows(0).Item(i)) Then
            'recuperamos el nombre del item

```



```
    item = EstMenu(i)
    'Validamos si contiene alguna de estas palabras, para proceder a eliminar el control
    If item.Contains("Seguimiento") = True Then
        men.Items.Item(1).ChildItems.Remove(men.FindItem(item))
    Elseif item.Contains("Rechazo") = True Then
        men.Items.Item(0).ChildItems.Remove(men.FindItem(item))
    Else
        'De lo contrario se quedara visible
    End If
End If
Next
ds.Tables.Remove("ROLL")
If Not area = "1" Then
    men.Items.Item(1).ChildItems.Remove(men.FindItem("Seguimiento/Comprador"))
End If
End Sub
```

'Funcion que convierte el numero de item en el valor del identificador del menu

```
Public Function EstMenu(ByVal item As Integer) As String
    Select Case item
        Case 0
            Return "Usuarios"
        Case 1
            Return "Rechazo/Nuevo"
        Case 2
            Return "Rechazo/Buscar"
        Case 3
            Return "Seguimiento/Almacen"
        Case 4
            Return "Seguimiento/Consulta Proveedor"
        Case 5
            Return "Seguimiento/Aplica Rechazo"
        Case 6
            Return "Autorizacion Mtrls"
        Case 7
            Return "Cuentas por Pagar"
        Case 8
            Return "Seguimiento/Material Sirve"
        Case 9
            Return "Seguimiento/Sortea o Retrabaja"
        Case 10
            Return "# Piezas"
        Case 11
            Return "Proveer NC"
        Case 12
            Return "Cuentas por Pagar"
        Case 13
            Return "Seguimiento/Regresa a Origen"
        Case 14
            Return "Seguimiento/PO"
    End Select
End Function
```

1.4. Visualización de contenido dependiendo del área

El siguiente código filtra los campos que serán visualizados dependiendo del área que ingrese

```
'Filtra las columnas de un tabla(grid) dependiendo del area
Public Sub grid(ByVal nomarea As String, ByVal grid As GridView)
    Dim j As Integer
    'Creamos el query que selecciona las areas existentes en el sistema
    query = "SELECT descripcion FROM areas "
    sql_adp = New SqlDataAdapter(query, sqlcnx)
    'Recuperamos los datos del query
    sql_adp.Fill(ds, "ROLL")
    sqlcnx.Close()
    'Recorremos la filas de la tabla que recuperamos
    For j = 0 To ds.Tables("ROLL").Rows.Count - 1
        'Asignamos en una variable el nombre del area
        Dim ar As String = ds.Tables("ROLL").Rows(j)("descripcion")
        'Si la variable del area es igual a nom area no la ocultamos
        If ar = nomarea Then
        Else
            'Recorremos las columnas de la tabla(grid)
            For i = 0 To grid.Columns.Count - 1
                'Si la columna contiene el nombre del area la ocultamos
                If grid.Columns(i).HeaderText.Equals(ar) Then
                    grid.Columns(i).Visible = False
                End If
            Next
        End If
    Next
    'eliminamos la tabla creada
    ds.Tables.Remove("ROLL")
End Sub
```

1.5. Envió de correos

El siguiente código muestra la forma en que se envía el correo, dependiendo de los usuarios a los cuales se les enviara.

```
Imports System.Data.SqlClient
Imports System.Web.Mail

Public Class enviaemail
    Public Function mail(ByVal recibe As String, ByVal aviso As String, ByVal rechazo As String) As String
        'validamos a quien se les enviaron los correos, para recuperarlos
        If recibe = "Todos" Then
            'Creamos el query para seleccionar a todos los usuarios
            query = "SELECT u.usuario, u.nombre FROM areas a, usuarios u " & _
                "WHERE u.idarea = a.idarea and u.status = '1'"
        ElseIf recibe = "Compras" Then
            'De la lista de compradores recuperamos los usuarios
            query = "SELECT comprador FROM rejection WHERE folio = " & rechazo
        Else
            'Se recuperan los usuarios dependiendo del area que recibe
            query = "SELECT u.usuario, u.nombre FROM areas a, usuarios u " & _
                "WHERE a.descripcion = " & recibe & " and u.idarea = a.idarea and u.status = '1' and u.idroll in (1,2)"
        End If
        Dim ret As String
        Dim usuarios As DataTable
```

```

sql_adp = New SqlDataAdapter(query, sqlcnx)
'recuperamos los datos en el data set (ds)
sql_adp.Fill(ds, "enviar")
usuarios = ds.Tables("enviar")
sqlcnx.Close()
ds.Tables.Remove("enviar")
'Dim u As String = usuarios.Rows(i).Item(0) 'usuario
'Dim n As String = usuarios.Rows(i).Item(1) 'nombre
Dim email As New System.Net.Mail.MailMessage
email = New System.Net.Mail.MailMessage()
email.From = New System.Net.Mail.MailAddress("rechazos@AmericanStandard.com")
'enviamos copias ocultas para validar que se realizan los envios de los correos
email.Bcc.Add("con_acostar@americanstandard.com")
email.Bcc.Add("enriquezn@americanstandard.com")
"-->envio a usuarios
Dim u As String
'recorremos la tabla con los usuareios que se recuperaron
For i As Integer = 0 To usuarios.Rows.Count - 1
    u = usuarios.Rows(i).Item(0) 'usuario
    'Asignamos el correo del usuario
    email.Bcc.Add("" & u & "@AmericanStandard.com")
Next
'-->fin envio a usuarios
email.Subject = "¡Aviso!"
email.IsBodyHtml = True
'Asignamos el contenido del correo, el cual sera el codigo html
email.Body = "<html>" & _
    "<head>" & _
    "<STYLE TYPE=""text/css"">" & _
    ".description()" & _
    "{" & _
    " font-size:10pt;" & _
    " font-weight:400;" & _
    " font-family:arial, sans-serif;" & _
    "}" & _
    "</STYLE>" & _
    "<Font Face = ""Verdana"" Color = #7C5828 Size=""2"">" & _
    "<body>" & _
    "<div class= description >" & _
    "    Saludos <B>" & recibe & ". </B><BR><BR>" & _
    "</B>" & aviso & _
    "    </B><BR><BR>" & _
    "<A href=""http://" & System.Net.Dns.GetHostName & "/rechazos/login.aspx"">Acceso
al Sistema</A><BR>" & _
    "</div>" & _
    "</body>" & _
    "</html>"

'Establecemos la prioridad del correo
email.Priority = System.Net.Mail.MailPriority.Normal

'Asignamos el servicio que realizará el envio del correo
Dim smtp As New System.Net.Mail.SmtpClient
'smtp.Host = "170.205.28.46"
smtp.Host = "170.205.4.168"
smtp.EnableSsl = False

```

```

Try
    'Enviamos el correo
    smtp.Send(email)
    Try
        ret = " Se envió notificación a " & recibe
    Catch ex As Exception
        'Si surge un error almacenamos la informacion para darle seguimiento
        Dim ce As cError = New cError()
        ce.almacenaError(ex.Message.ToString, "Envío de correo")
        ret = " Error al enviar el correo"
    Finally
        sqlcnx.Close()
    End Try
Catch ex As Exception
    Dim ce As cError = New cError()
    ce.almacenaError(ex.Message.ToString, "Envío de correo")
End Try
Return ret
End Function
End Class

```

1.6. Creación del PDF con los datos del rechazo

```

Dim PdfTable As New PdfPTable(4)
Dim PdfTable1 As New PdfPTable(4)
Dim PdfPCell As PdfPCell = Nothing
Dim font8 As Font = FontFactory.GetFont("VERDENNA", 10)

Public Sub GenerarPDF(ByVal HTM As String)
    'Variable que almacena los parametros de configuracion de la hoja del pdf
    Dim oDoc As New iTextSharp.text.Document(iTextSharp.text.PageSize.LETTER, 10, 10, 42,
35)
    'Nueva instancia para crear un nuevo documento pdf
    Dim pdfw As iTextSharp.text.pdf.PdfWriter
    'Variable que almacena la ruta donde se almacenara el documento
    Dim NombreArchivo As String = "C:\ejemplo.pdf"
    Try
        pdfw = PdfWriter.GetInstance(oDoc, New FileStream(NombreArchivo, _
        FileMode.Create, FileAccess.Write, FileShare.None))
        'Apertura del documento.
        oDoc.Open()
        Dim font8 As Font = FontFactory.GetFont("ARIAL", 12)
        'Write some content
        Dim paragraph As New Paragraph("Using ITextsharp I am going to show how to create
simple table in PDF document ")
        'Creando una instancia de la tabla de PDF y especificar el número de columna de esa tabla
        Dim PdfTable As New PdfPTable(4)
        Dim PdfPCell As PdfPCell = Nothing
        'Agregamos los titulos de la tabla en el pdf, asignando nombre, color, alineacion y numero
de columnas colapsadas
        PdfPCell = New PdfPCell(New Phrase(New Chunk(" INFORME ", font8)))
        PdfPCell.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY
        PdfPCell.HorizontalAlignment = Element.ALIGN_CENTER
        PdfPCell.Colspan = 4
        PdfTable.AddCell(PdfPCell)
    End Try
End Sub

```

```

PdfPCell = New PdfPCell(New Phrase(New Chunk("Rejection Nbr: Iblrejection", font8)))
PdfPCell.HorizontalAlignment = Element.ALIGN_RIGHT
PdfPCell.Colspan = 4
PdfTable.AddCell(PdfPCell)
PdfPCell = New PdfPCell(New Phrase(New Chunk("Invoice Nbr: txtinvoice", font8)))
PdfPCell.HorizontalAlignment = Element.ALIGN_RIGHT
PdfPCell.Colspan = 4
PdfTable.AddCell(PdfPCell)
PdfPCell = New PdfPCell(New Phrase(New Chunk(" DATA REJECTION ", font8)))
PdfPCell.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY
PdfPCell.HorizontalAlignment = Element.ALIGN_CENTER
PdfPCell.Colspan = 4
PdfTable.AddCell(PdfPCell)
Dim str As String = tablas(HTM)
str = str.Replace(", ", ", ")
str = str.Replace(", ", ", ")
str = str.Replace(", ", ", ")
str = str.Replace(", ", ", ")
str = str.Replace(", ", ", ")
str = str.Replace(", ", ", ")
str = str.Replace(", ", ", ")
Dim longitud As Integer = Len(str)
Dim siguiente, palabra As String
Dim contador As Integer = 0
Dim cabre, ccierra As Boolean
For l = 1 To longitud
    siguiente = Mid(str, l, 1)
    If siguiente = "," Then
        cabre = True
        contador += 1
    End If
    If contador = 1 Then
        If cabre = True Then
            palabra = palabra & siguiente
            palabra = palabra.Replace(", ", "")
            'Dependiendo la palabra contenida, crearemos la celda con la informacion
            Select Case palabra
                Case "INFORME"
                    If Mid(str, l + 1, 1) = "," Then
                        Else
                            palabra = ""
                        End If
                    Case "DATA REJECTION"
                        If Mid(str, l + 1, 1) = "," Then
                            l = l + 1
                            palabra = ""
                        Else
                            palabra = ""
                        End If
            PdfPCell = New PdfPCell(New Phrase(New Chunk(" DATA REJECTION ",
font8)))

            PdfPCell.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY
            PdfPCell.HorizontalAlignment = Element.ALIGN_CENTER
            PdfPCell.Colspan = 4
            PdfTable.AddCell(PdfPCell)

            Case "SORTEA O RETRABAJA"

```

```

        If Mid(str, l + 1, 1) = "," Then
            l = l + 1
            palabra = ""
        Else
            palabra = ""
        End If
        PdfPCell = New PdfPCell(New Phrase(New Chunk("  SORTEA O
RETRABAJA ", font8)))
        PdfPCell.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY
        PdfPCell.HorizontalAlignment = Element.ALIGN_CENTER
        PdfPCell.Colspan = 4
        PdfTable.AddCell(PdfPCell)
        Case "HOJA ACEPTADO"
            If Mid(str, l + 1, 1) = "," Then
                l = l + 1
                palabra = ""
            Else
                palabra = ""
            End If
            PdfPCell = New PdfPCell(New Phrase(New Chunk("  HOJA ACEPTADO ",
font8)))
            PdfPCell.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY
            PdfPCell.HorizontalAlignment = Element.ALIGN_CENTER
            PdfPCell.Colspan = 4
            PdfTable.AddCell(PdfPCell)
            Case "SIRVE EL MATERIAL "
                If Mid(str, l + 1, 1) = "," Then
                    l = l + 1
                    palabra = ""
                Else
                    palabra = ""
                End If
                PdfPCell = New PdfPCell(New Phrase(New Chunk("  SIRVE EL MATERIAL ",
font8)))
                PdfPCell.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY
                PdfPCell.HorizontalAlignment = Element.ALIGN_CENTER
                PdfPCell.Colspan = 4
                PdfTable.AddCell(PdfPCell)
                Case "APLICA RECHAZO"
                    If Mid(str, l + 1, 1) = "," Then
                        l = l + 1
                        palabra = ""
                    Else
                        palabra = ""
                    End If
                    PdfPCell = New PdfPCell(New Phrase(New Chunk("  APLICA RECHAZO ",
font8)))
                    PdfPCell.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY
                    PdfPCell.HorizontalAlignment = Element.ALIGN_CENTER
                    PdfPCell.Colspan = 4
                    PdfTable.AddCell(PdfPCell)
                    Case "IMPO / EXPO"
                        If Mid(str, l + 1, 1) = "," Then
                            l = l + 1
                            palabra = ""
                        Else

```

```

        palabra = ""
    End If
    PdfPCell = New PdfPCell(New Phrase(New Chunk(" IMPO / EXPO ", font8)))
    PdfPCell.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY
    PdfPCell.HorizontalAlignment = Element.ALIGN_CENTER
    PdfPCell.Colspan = 4
    PdfTable.AddCell(PdfPCell)
End Select
If siguiente = "." Then
    palabra = palabra.Replace(".", "")
    PdfPCell = New PdfPCell(New Phrase(New Chunk(palabra, font8)))
    PdfPCell.HorizontalAlignment = Element.ALIGN_CENTER
    PdfTable.AddCell(PdfPCell)
    palabra = ""
End If
End If
Elseif contador = 2 Then
    contador = 0
    palabra = palabra & siguiente
    palabra = palabra.Replace(", ", "")
    If palabra = "" Then
        PdfPCell = New PdfPCell(New Phrase(New Chunk("", font8)))
        PdfPCell.BackgroundColor = iTextSharp.text.BaseColor.LIGHT_GRAY
        PdfPCell.HorizontalAlignment = Element.ALIGN_CENTER
        PdfPCell.Colspan = 4
        PdfTable.AddCell(PdfPCell)
        contador = 1
        palabra = ""
    Else
        PdfPCell = New PdfPCell(New Phrase(New Chunk(palabra, font8)))
        PdfPCell.HorizontalAlignment = Element.ALIGN_CENTER
        PdfTable.AddCell(PdfPCell)
        contador = 1
        palabra = ""
    End If
End If
Next I
PdfTable.SpacingBefore = 15.0F
'Dar un poco de espacio después del texto o que se pueden superponer
oDoc.Add(paragraph)
'añadir el párrafo en el documento
oDoc.Add(PdfTable)
'Forzamos vaciamiento del buffer.
pdfw.Flush()
'Cerramos el documento.
oDoc.Close()
Catch ex As Exception
'Si hubo una excepcion y el archivo existe ...
If File.Exists(NombreArchivo) Then
    'Cerramos el documento si esta abierto.
    'Y asi desbloqueamos el archivo para su eliminacion.
    If oDoc.IsOpen Then oDoc.Close()
    '... lo eliminamos de disco.
    File.Delete(NombreArchivo)
End If
Throw New Exception("Error al generar archivo PDF (" & ex.Message & ")")

```

```
Finally
    pdfw = Nothing
    oDoc = Nothing
End Try
End Sub
```

1.7. Recuperar documentos contenidos en el rechazo

El siguiente código muestra la forma en que se recuperan los documentos almacenados en la base de datos.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
    'validamos que no se realice un postback
    If Not Page.IsPostBack Then
        'recuperamos la variable enviada via url, y la validamos
        If Request.QueryString("f") = "Nada" Then
            'Si la variable se recupera vacia no se recupera documento alguno
            Response.Expires = 0
            Response.Buffer = True
            Response.Clear()
            Response.ContentType = "application/pdf"
            Response.WriteFile(direccion)
            direccion = ""
            Response.End()
        Else
            'Si recuperamos algun valor en la variable contruimos e query que recuperara la imagen
            que se almaceno en l base
            Response.ContentType = Nothing
            query = "SELECT i.extension, i.imagen, c.contenttype FROM imagenes i, contentType c"
            & _
                " where archivo = " & Request.QueryString("a") & " and rejectionnbr = " &
Request.QueryString("r") & " and frm = " & Request.QueryString("f") & " and c.extencion =
i.extension"
            sql_adp = New SqlDataAdapter(query, sqlcnx)
            'recuperamos los datos del query
            sql_adp.Fill(ds, "imagen")
            sqlcnx.Close()
            'recuperamos la imagen contenida en los datos recuperados y lo convertimos en byte,
para asignarlo
            'a la variable
            Dim bytelImage As Byte() = CType(ds.Tables("imagen").Rows(0)("imagen"), Byte())
            'asignamos la aplocacion con la cual se abra el documento
            Response.ContentType = ds.Tables("imagen").Rows(0)("contenttype")
            'eliminamos la tabla que se creo
            ds.Tables.Remove("imagen")
            Response.Expires = 0
            Response.Buffer = True
            Response.Clear()
            'guardamos la imagen y la abrimos
            Response.BinaryWrite(bytelImage)
            Response.End()
        End If
    End If
End Sub
```