



INSTITUTO TECNOLÓGICO
“CORDILLERA”

CARRERA DE ANÁLISIS DE SISTEMAS

AUTOMATIZACIÓN DE LOS PROCESOS FINANCIEROS MEDIANTE UN
SISTEMA INTEGRADO DE GESTIÓN FINANCIERA: MÓDULO DE
TESORERÍA

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

Autor: Luis Bolívar Trejo Cruz

Tutor: Lic. Wilson Núñez

Quito, Abril 2015

Acta de Aprobación de Tutor y Lector

En mi calidad de tutor del trabajo sobre el tema: AUTOMATIZACIÓN DE LOS PROCESOS FINANCIEROS MEDIANTE UN SISTEMA INTEGRADO DE GESTION FINANCIERA: MÓDULO DE TESORERÍA, presentado por el ciudadano: **Luis Bolívar Trejo Cruz**, estudiante de la Escuela de Sistemas, considero que dicho informe reúne los requisitos y méritos suficientes para ser sometido a la evaluación por parte del Tribunal de Grado, que el Honorable Consejo de Escuela designe, para su correspondiente estudio y calificación.

Quito, Abril de 2015

Lcdo. Wilson Nuñez

TUTOR

Ing. Hugo Heredia

LECTOR

Declaratoria

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

Luis Bolívar Trejo Cruz

C.C. 040154401-0

Cesión de Derechos

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

PRIMERA: ANTECEDENTE.- a) El Cedente dentro del pensum de estudio en la carrera de análisis de sistemas que imparte el Instituto Superior Tecnológico Cordillera, y con el objeto de obtener el título de Tecnólogo en Análisis de Sistemas, el estudiante participa en el proyecto de grado denominado “AUTOMATIZACIÓN DE LOS PROCESOS FINANCIEROS MEDIANTE UN SISTEMA INTEGRADO DE GESTIÓN FINANCIERA: MÓDULO DE TESORERÍA”, el cual incluye la creación y desarrollo del programa de ordenador o software, para lo cual ha implementado los conocimientos adquiridos en su calidad de alumno. **b)** Por iniciativa y responsabilidad del Instituto Superior Tecnológico Cordillera se desarrolla la creación del programa de ordenador, motivo por el cual se regula de forma clara la cesión de los derechos de autor que genera la obra literaria y que es producto del proyecto de grado, el mismo que culminado es de plena aplicación técnica, administrativa y de reproducción.

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

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

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

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

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

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

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

f) _____

C.C. N° 0401544010

CEDENTE

f) _____

Instituto Superior Tecnológico Cordillera

CESIONARIO

Agradecimiento

Gracias al Ing. Wilson Núñez, por el valioso aporte profesional, técnico y humano brindado en el transcurso de este semestre en favor de la culminación de la memorable carrera que después de mucho sacrificio y esfuerzo se culmina.

A los docentes y compañeros que compartieron conmigo la vida institucional y fueron partícipes de cada experiencia, gracias por permitirme compartir y aprender junto a ustedes.

Dedicatoria

A mi madre Clara Matilde Cruz Lima;

A mi padre Luis Bolívar Trejo Ibarra que en paz descansa;

Ya que gracias a ellos he alcanzado este logro;

A mi hermano;

A todos mis familiares y amigos.

Índice General

Contenido

| | |
|---------------------------------------------|-------|
| Acta de Aprobación de Tutor y Lector | ii |
| Declaratoria | iii |
| Cesión de Derechos | iv |
| Agradecimiento | viii |
| Dedicatoria | ix |
| Índice de figuras | xv |
| Resumen ejecutivo | xviii |
| Introducción | xxi |
| Capítulo I: Antecedentes | 1 |
| 1.01. Contexto | 1 |
| 1.02. Justificación | 1 |
| 1.03 Definición del Problema Central | 2 |
| Capítulo II: Análisis de Involucrados | 6 |
| 2.01. Requerimientos | 6 |
| 2.01.1 Descripción del sistema actual | 6 |
| 2.01.2 Visión y Alcance | 6 |
| 2.01.2.1 Visión | 6 |
| 2.01.2.1 Alcance | 7 |
| 2.01.3 Entrevistas | 7 |
| 2.01.4 Matriz de Requerimientos | 8 |
| 2.01.5 Descripción Detallada | 9 |
| 2.02 Mapeo de Involucrados. | 17 |
| 2.03. Matriz de Involucrados | 18 |
| Capítulo III: Problemas y Objetivos | 19 |
| 3.01. Árbol de Problemas | 19 |
| 3.0.2 Árbol de Objetivos | 20 |
| 3.03. Diagramas de casos de uso | 21 |

| | |
|----------------------------------------------------------|----|
| 3.04. Casos de uso de Realización | 23 |
| 3.05. Diagrama de secuencias del sistema..... | 27 |
| 3.06. Especificación de casos de uso..... | 29 |
| Capítulo IV: Análisis de Alternativas | 31 |
| 4.01. Matriz de Análisis de Alternativas | 31 |
| 4.02. Matriz de Impactos de Objetivos..... | 33 |
| 4.03. Estándares para el Diseño de Clases | 34 |
| 4.04. Diagrama de clases | 35 |
| 4.05. Modelo Lógico – Físico | 37 |
| 4.06. Diagrama de Componentes..... | 38 |
| 4.07. Diagramas de Estrategias | 39 |
| 4.08 . Matriz de Marco Lógico..... | 40 |
| 4.09. Vistas arquitectónicas..... | 41 |
| 4.01.01. Vista lógica | 41 |
| 4.01.02. Vista física | 42 |
| 4.01.03. Vista de desarrollo | 42 |
| Capítulo V: Propuesta | 44 |
| 5.01. Especificación de estándares de programación | 44 |
| 5.01.01. Organización de ficheros | 44 |
| 5.01.02. Fichero fuente Java (.java)..... | 44 |
| 5.01.04. Sentencias de paquete | 46 |
| 5.01.06. Sentencias de importación | 46 |
| 5.01.07. Declaraciones de clases e interfaces | 48 |
| 5.01.08. Sangría | 48 |
| 5.01.09. Longitud de línea | 49 |
| 5.01.10. División de líneas..... | 49 |
| 5.01.11. Comentarios | 50 |
| 5.01.12. Comentarios de implementación | 50 |
| 5.01.13. Comentarios de documentación..... | 51 |
| 5.01.14. Una declaración por línea | 51 |
| 5.01.15. Inicialización..... | 51 |

| | |
|--------------------------------------------------------------------|-----|
| 5.01.16. Localización..... | 51 |
| 5.01.17. Declaración de clases / interfaces | 52 |
| 5.01.18. Sentencias | 53 |
| 5.01.19. Espacios en blanco..... | 54 |
| 5.01.20. Nomenclatura de identificadores | 55 |
| 5.01.21. Paquetes | 56 |
| 5.01.22. Clases e interfaces..... | 57 |
| 5.01.23. Métodos | 57 |
| 5.01.24. Variables | 58 |
| 5.01.25. Constantes..... | 58 |
| 5.01.02.01. Prácticas de programación | 58 |
| 5.01.02.02. Visibilidad de atributos de instancia y de clase | 58 |
| 5.01.02.03. Referencias a miembros de una clase | 59 |
| 5.01.02.04. Constantes..... | 59 |
| 5.01.02.05. Asignación sobre variables | 60 |
| 5.01.02.06. Otras prácticas..... | 61 |
| 5.01.02.07. Documentación: javadoc..... | 62 |
| 5.02. Diseño de Interfaces de Usuario..... | 67 |
| 5.03. Especificación de pruebas de unidad..... | 76 |
| 5.04 Especificación de pruebas de aceptación | 79 |
| 5.05. Especificación de pruebas de carga | 79 |
| 5.06. Configuración del Ambiente mínima/ideal | 82 |
| Capítulo VI: Aspectos Administrativos | 85 |
| 6.01. Recursos | 85 |
| 6.02. Presupuesto..... | 85 |
| Capítulo VII: Conclusiones y Recomendaciones..... | 87 |
| 7.01. Conclusiones | 87 |
| 7.02. Recomendaciones..... | 87 |
| Bibliografía | 120 |

Índice de Tablas

| Título | Página |
|------------------------------------------------------------------------------------------------|---------------|
| Tabla 1 <i>Matriz de fuerzas T.</i> | 2 |
| Tabla 2 <i>Formato de entrevista</i> | 7 |
| Tabla 3 <i>Diseño de requerimientos</i> | 8 |
| Tabla 4 <i>Requerimiento funcional 001</i> | 9 |
| Tabla 5 <i>Requerimiento funcional 002</i> | 10 |
| Tabla 6 <i>Requerimiento funcional 003</i> | 11 |
| Tabla 7 <i>Requerimiento funcional 004</i> | 12 |
| Tabla 8 <i>Requerimiento no funcional 001</i> | 13 |
| Tabla 9 <i>Requerimiento no funcional 002</i> | 14 |
| Tabla 10 <i>Requerimiento no funcional 003</i> | 15 |
| Tabla 11 <i>Requerimiento no funcional 004</i> | 16 |
| Tabla 12 <i>Detalle de interés, involucrados y conflictos percibidos en el proyecto.</i> | 18 |
| Tabla 13 <i>Bancos</i> | 24 |
| Tabla 14 <i>Anulación de cheques</i> | 25 |
| Tabla 15 <i>Cuentas por cobrar</i> | 26 |
| Tabla 16 <i>Cuentas por pagar</i> | 27 |
| Tabla 17 <i>Especifica casos de uso general.</i> | 29 |
| Tabla 18 <i>Especifica caso de uso 002 de cobros.</i> | 29 |
| Tabla 19 <i>Especifica caso de uso 003 de cobros.</i> | 29 |
| Tabla 20 <i>Especifica caso de uso 004 de bancos.</i> | 30 |
| Tabla 21 <i>Análisis de alternativas.</i> | 31 |
| Tabla 22 <i>Matriz de Análisis de Impactos de Objetivos</i> | 33 |

| | |
|--------------------------------------------------------------------------------------------|----|
| Tabla 23 <i>Resumen del proyecto lo que se desea lograr</i> | 40 |
| Tabla 24 <i>Declaración de clases e interfaces</i> | 48 |
| Tabla 25 <i>Ventana principal</i> | 70 |
| Tabla 26 <i>Registro de facturas</i> | 71 |
| Tabla 27 <i>Notas de crédito</i> | 72 |
| Tabla 28 <i>Anticipos proveedores</i> | 73 |
| Tabla 29 <i>Pago a proveedores</i> | 74 |
| Tabla 30 <i>Cuadre de anticipos</i> | 75 |
| Tabla 31 <i>Bancos</i> | 76 |
| Tabla 32 <i>Matriz de pruebas de carga</i> | 77 |
| Tabla 33 <i>Pruebas de Reportes, resultados eficientes</i> | 77 |
| Tabla 34 <i>Pruebas de compilación de Código.</i> | 78 |
| Tabla 35 <i>Pruebas de Almacenamiento de datos en la Base</i> | 78 |
| Tabla 36 <i>Detalle de pruebas de aceptación en la creación de usuarios</i> | 79 |
| Tabla 37 <i>Detalle de un tipo de prueba de carga más baja.</i> | 81 |
| Tabla 38 <i>Detalle de un tipo de prueba de carga con un número mínimo de usuarios</i> ... | 81 |
| Tabla 39 <i>Detalle de un tipo de prueba de carga con un número máximo de usuarios</i> .. | 82 |
| Tabla 40 <i>Se requiere descubrir los límites</i> | 82 |
| Tabla 41 <i>Detalle de Gastos realizados en el Proyecto (Presupuesto)</i> | 85 |

Índice de figuras

| Título | Página |
|-----------------------------------------------------------------------|---------------|
| Figura 1 <i>Mapeo de involucrados</i> | 17 |
| Figura 2 . <i>Árbol de Problemas</i> | 19 |
| Figura 3 <i>Árbol de objetivos</i> | 20 |
| Figura 4 <i>Diagrama de Caso de Uso General 001</i> | 21 |
| Figura 5 <i>Diagrama de Caso de Uso 002</i> | 22 |
| Figura 6 <i>Diagrama de Caso de Uso 003</i> | 22 |
| Figura 7 <i>Diagrama de Caso de Uso 004</i> | 23 |
| Figura 8 <i>Diagrama de Caso de Uso de Realización 001</i> | 23 |
| Figura 9 <i>Diagrama de Caso de Uso de Realización 002</i> | 24 |
| Figura 10. <i>Diagrama de Caso de Uso de Realización 003</i> | 25 |
| Figura 11 <i>Diagrama de Caso de Uso de Realización 004</i> | 26 |
| Figura 12 <i>Transacciones del módulo</i> | 28 |
| Figura 13. <i>Manejo de bancos cuentas por cobra y pagar</i> | 28 |
| Figura 14 <i>Vista detallada de las clases</i> | 36 |
| Figura 15 <i>Vista detallada del modelo lógico del sistema</i> | 37 |
| Figura 16 <i>Diagrama Modelo Físico</i> | 37 |
| Figura 17 <i>Diagrama de Componentes 001</i> | 38 |
| Figura 18 <i>Diagrama de Estrategias</i> | 39 |
| Figura 19 <i>Descripción Lógica a del sistema</i> | 41 |
| Figura 20 <i>Diagrama de Despliegue</i> | 42 |
| Figura 21 <i>Diagrama de Componentes</i> | 42 |
| Figura 22 <i>Descripción física del sistema</i> | 43 |

| | |
|---------------------------------------------------------------------------|----|
| Figura 23 Capas con las que interactúa la interface del usuario | 68 |
| Figura 24 Diagrama general de los pasos para diseñar una interface. | 68 |
| Figura 25 Muestra la página principal de ERP. | 69 |
| Figura 26 Ingreso al sistema. | 69 |
| Figura 27 Muestra el ingreso de facturas al sistema | 70 |
| Figura 28 Muestra el ingreso de las notas de crédito | 72 |
| Figura 29 Muestra el anticipo a proveedores | 73 |
| Figura 30 Muestra el pago a proveedores | 74 |
| Figura 31 Muestra el cuadro de anticipos..... | 75 |
| Figura 32 Muestra el ingreso a bancos..... | 76 |
| Figura 33 Diagrama de secuencia del proceso de pruebas de aceptación..... | 79 |
| Figura 34 Se verifica la versión de sistema operativo | 89 |
| Figura 35 Inicio de instalación de WampServer | 90 |
| Figura 36 Ejecutar como administrador | 90 |
| Figura 37 Paso 1 de instalación..... | 91 |
| Figura 38 Acepta los acuerdos de la licencia | 91 |
| Figura 39 Selección de carpeta de instalación | 92 |
| Figura 40 Opción de icono en el escritorio..... | 92 |
| Figura 41 Inicio de instalación | 93 |
| Figura 42 Muestra el progreso de la instalación | 93 |
| Figura 43 Selecciona el navegador a ser usado..... | 94 |
| Figura 44 Envío de email | 94 |
| Figura 45 Final de instalación de WampServer..... | 95 |

| | |
|----------------------------------------------------------------------------|-----|
| Figura 46 <i>Ejecuta el instalador como administrador</i> | 95 |
| Figura 47 <i>Pantalla de bienvenida</i> | 96 |
| Figura 48 <i>Acuerdo de licencia</i> | 96 |
| Figura 49 <i>Seleccionamos los componentes que necesitamos</i> | 97 |
| Figura 50 <i>Selección de carpeta de instalación</i> | 97 |
| Figura 51 <i>Configuración de puerto usuario y contraseña</i> | 98 |
| Figura 52 <i>Inicio de instalación</i> | 98 |
| Figura 53 <i>Progreso de instalación</i> | 99 |
| Figura 54 <i>Final de instalación Apache Tomcat</i> | 99 |
| Figura 55 <i>Navegador de internet</i> | 100 |
| Figura 56 <i>Ingreso de dirección o URL para abrir el sistema</i> | 100 |
| Figura 57 <i>Pantalla de ingreso al sistema</i> | 101 |
| Figura 58 <i>Muestra la pantalla con todos los módulos</i> | 101 |
| Figura 59 <i>Pantalla en la cual ingresamos usuario y contraseña</i> | 102 |
| Figura 60 <i>Selección de empresa y periodo</i> | 102 |
| Figura 61 <i>Lista las diferentes tareas que necesita el usuario</i> | 103 |
| Figura 62 <i>Muestra la pantalla de ingreso de facturas</i> | 103 |
| Figura 63 <i>Muestra la pantalla de nota de crédito</i> | 104 |
| Figura 64 <i>Muestra la pantalla de pago de anticipos</i> | 104 |
| Figura 65 <i>Muestra la pantalla de pagos</i> | 105 |
| Figura 66 <i>Muestra la pantalla de cuadro de anticipos</i> | 105 |
| Figura 67 <i>Muestra la pantalla de ingreso a bancos</i> | 106 |
| Figura 68 <i>Muestra la pantalla de consulta de retenciones</i> | 106 |

| | |
|-----------------------------------------------------------|-----|
| Figura 69 Muestra la pantalla de cuentas por cobrar | 107 |
| Figura 70 Muestra la pantalla de cuentas por cobrar | 107 |

Resumen ejecutivo

El presente trabajo de titulación propone la implementación de un Módulo de Tesorería el cual es parte de un sistema integrado de gestión financiera en ambiente Web. El origen del trabajo a desarrollarse analizando el capítulo 1, donde se da a conocer una breve explicación de porqué se eligió el tema, su contexto, justificación y realizando la construcción de la matriz de fuerzas T. Determinando las fuerzas que impulsan y bloquean la situación negativa dentro de la organización, capítulo 2 detallaremos el mapa de identificación de los involucrados directos e indirectos, deberá salir del levantamiento de requerimientos y las entrevistas que se realizan a las empresas donde nos permita dar a conocer el alcance de nuestro tema, capítulo 3 determinaremos el árbol de problemas luego de haber generado los requerimientos y de obtener los involucrados, es importante que se organice la información tomando en consideración la causa, problema y efecto que determinan el problema central y posterior al árbol de objetivos que tiene fines y medios que se genera en base de el árbol de problemas ya con esto se podrá tener una visión clara de los problemas que se generan y procesos de la misma pues nos servirá para realizar los diagramas de casos de uso, realización, secuencia, capítulo 4 se determinara el estándar para nombrar, clases, atributos, métodos que serán utilizados para el diseño de clases de la solución y diagrama de clases , modelo lógico físico y demás diagramas que se generan en el modelo UML , capítulo 5 se desarrollara la propuesta estándar de codificación usado para la implementación del producto, y pruebas de aceptación, calidad, capítulo 6 detallaremos el costo del producto y su planificación, capítulo 7 se dará conclusiones y recomendaciones a más de los anexos y manuales de usuario, instalación, técnico.

Abstract

This paper proposes titling implementing a Treasury module which is part of an integrated financial management system in Web environment. The origin of the work to develop analyzing Chapter 1, where it provides a brief explanation of why the topic, context, rationale was chosen and performing the construction of the matrix T. forces determining the forces driving and block the situation negative within the organization, Chapter 2 will detail the map of identifying direct and indirect involved must exit requirements gathering and interviews are conducted to companies which allow us to present the scope of our subject, Chapter 3 will determine the problem tree after having generated the requirements and to get involved, it is important that the information be organized taking into account the cause, problem and effect that determine the center and rear objective tree problem having ends and means that generated based on the problem tree and with this you can have a clear view of the problems that are generated and processes the same as we serve to make diagrams use cases, performance, string, Chapter 4 standard is determined to name, classes, attributes, methods to be used for the design of classes of solution and class diagram, physical logic model and other diagrams that are generated in the UML model, Chapter 5 standard codification proposal was developed used for product implementation and acceptance testing, quality, Chapter 6 will detail the cost of the product and its planning, Chapter 7 will give conclusions and recommendations over the annexes and user manuals, installation technician.

Introducción

El presente estudio tiene por objetivo realizar la implementación de un sistema de información que automatice el control del departamento de tesorería de las empresas Ecuatorianas que sea ágil y seguro, ya que manipula información sensible tanto en cuentas por cobrar como en cuentas por pagar de una empresa, es decir, manejar el rendimiento de ganancias o pérdidas de la misma.

Necesitan automatizar los procesos ya que éstos no se los realiza con efectividad y agilidad. Esto quiere decir que tendremos una interfaz de usuario en donde podremos realizar varias operaciones, verificaciones y consultas de manera detallada.

Actualmente, el departamento de tesorería de las empresas lleva a mano con aplicaciones que ya no están acorde con las necesidades actuales, teniendo como consecuencia equivocaciones al momento que los empresarios realicen la toma de decisiones, dando como consecuencia a la pérdida de tiempo en la verificación de información y toma de decisiones por parte de los empresarios y se genera una carga de trabajo repetitiva para los trabajadores que lo realizan.

Por ello, tras un arduo análisis de la gestión a automatizar, la solución planteada permitirá una buena interacción con la información requerida por los trabajadores y empresarios en forma organizada, transparente y sin pérdida de tiempo. Esta investigación se dará a conocer paso a paso la solución que se podrá efectuar en los procesos financieros del módulo de tesorería.

Capítulo I: Antecedentes

1.01.Contexto

En esta primera parte del trabajo de titulación se sitúa el contexto en el que se ha realizado la investigación y el cuál es el estado actual.

La tesorería es la que se encarga directamente de los movimientos reales del dinero que entra y sale de la empresa, aglutina todas esas gestiones necesarias para conseguirlo fuentes de financiación como los descuentos comerciales, préstamos bancarios, etc.

Además, se encarga de almacenar todos los soportes de las transacciones, incluyendo la emisión diaria de la información sobre los fondos de la empresa, es el área encargada de aplicar todo ese conjunto de procesos administrativos que permiten prevenir los errores en cuanto al manejo de efectivo, la caja y bancos.

En la actualidad en el Ecuador esta importante área dentro de la estructura financiera en una empresa tiene falencias en situaciones tanto de manejo de caja, bancos, cuentas por cobrar y pagar dando como resultado el incumplimiento en los pagos a proveedores atrasos en los cobros de cartera falta de fondos en las cuentas bancarias ya que no se dispone información de manera veraz y oportuna de cobros y pagos que tienen que realizarse.

1.02.Justificación

El módulo de Tesorería es importante dentro de un Sistema de Gestión Financiera ya que cumple con los requerimientos solicitados por el departamento.

Servirá para automatizar los procesos de manejo de caja, bancos, cuentas por pagar y cobrar. De esta manera se tendrá mayor agilidad en el proceso de pagos cobros manejo de cuentas bancarias.

1.03 Definición del Problema Central

Las diferentes empresas en el Ecuador tienen problemas de liquidez, retrasos en pagos a proveedores así como también son deficientes al momento de realizar cobros ya que no disponen de una aplicación que ayude en estos procesos.

Esto motiva la necesidad de tener en cuenta la matriz T en la que se analizará cada uno de los pasos y procesos que se va desarrollando con el objetivo de suplir las deficiencias.

Tabla 1

Matriz de fuerzas T.

| ANÁLISIS DE FUERZAS T | | | | | |
|------------------------------------------------------------------------|--------------------------------------------------------------------------|----|---|----|--------------------------------------------------------------------------------------------------------------------------|
| Situación Empeorada | Situación Actual | | | | Situación Mejorada |
| Quiebre de la empresa por falta de liquidez | Iliquidez retraso en pagos a proveedores, deficiente gestión de cobranza | | | | Recuperación de cartera pago puntual a proveedores, emisión de cheques de acuerdo a la realidad económica de la empresa. |
| Fuerzas Impulsadoras | I | PC | I | PC | Fuerzas Bloqueadoras |
| 1.-Necesidad de un módulo dedicado únicamente a la tesorería. | 5 | 5 | 4 | 2 | 1.-Empresas que llevan el control de tesorería en conjunto con otras actividades. |
| 2.- Disponer de una sola herramienta de consulta y toma de decisiones. | 4 | 4 | 4 | 4 | 2.- Cada departamento lleva su control de forma aislada. |
| 3.-Buen uso del sistema de gestión financiera (módulo de tesorería). | 3 | 3 | 5 | 4 | 3.-Falta de capacitación por parte del personal. |
| 4.-Información consolidada de las cuentas. | 4 | 4 | 5 | 2 | 4.-Falta de cultura para entregar reportes a tiempo. |
| 5.-Nuevas normas por parte de las agencias de control. | 1 | 4 | 4 | 2 | 5.-Desinformación de las empresas acerca de las normas vigentes. |

Nota. Análisis de las fuerzas T, descripción de la situación empeorada, situación actual y situación mejorada.

I = Intensidad: ¿Cuál es el nivel de impacto de la fuerza sobre la problemática?

PC = Potencial de cambio: ¿Cuánto se puede modificar o aprovechar la fuerza para llegar a la situación deseada?

Escala:

1 = Bajo / 2 = Medio bajo / 3 = Medio / 4 = Medio alto / 5 = Alto

Interpretación

Fuerza Impulsadora: - Necesidad de un módulo dedicado únicamente a la tesorería

I = 5 Se define claramente la función del tesorero lo cual hace más eficiente su labor en la empresa.

PC = 5 Motiva al empresario a crear un puesto específico de trabajo.

Fuerza Impulsadora: Disponer de una sola herramienta de consulta y toma de decisiones

I = 4 Se evita tener la información dispersa y en constante riesgo de alterarse o peor aún extraviarse.

PC = 4 Se definen los reportes de manera clara para ser entregados por el departamento.

Fuerza Impulsadora: Buen uso del sistema de gestión financiera (Módulo de Tesorería)

I = 3 Capacitar al personal sobre el uso adecuado del módulo mediante manuales de usuario.

PC = 3 El usuario debe tener el conocimiento acerca del módulo para no tener inconvenientes durante el uso diario.

Fuerza Impulsadora: Información consolidada de las cuentas

I = 4 Se obtiene información actualizada en las cuentas como caja bancos cuentas por cobrar y cuentas por pagar.

PC = 4 Incremento en la rentabilidad de la empresa.

Fuerza Impulsadora: Nuevas normas por parte de las agencias de control

I = 1 Se evidencia la falta de interés de las empresas en usar aplicaciones que cumplan con las normativas vigentes.

PC = 4 Se obtiene una solución informática acorde a las normativas vigentes.

Fuerza Bloqueadora: Empresas que llevan el control de tesorería en conjunto con otras actividades.

I = 4 Estas tareas están a cargo del contador o su auxiliar.

PC = 2 Falta de presupuesto para crear el puesto.

Fuerza Bloqueadora: Cada departamento lleva su control de forma aislada.

I = 4 No existe una fuente de consulta automatizada, hay que pedir información al departamento que la tiene.

PC = 4 Empresas que carecen de infraestructura tecnológica, costos elevados de software.

Fuerza Bloqueadora: Falta de capacitación por parte del personal

I = 5 No existe interés del usuario por aprender a usar nuevas tecnologías.

PC = 4 Con usuarios capacitados se aprovecha al máximo el recurso tecnológico.

Fuerza Bloqueadora: Falta de cultura para entregar reportes a tiempo.

I = 5 Atraso constante de parte de los funcionarios.

PC = 2 La empresa dispondrá de la información el momento que se necesite.

Fuerza Bloqueadora: Desinformación de las empresas acerca de las normas vigentes.

I = 4 Falta de interés por los funcionarios.

PC = 2 La empresa funcionaria de acuerdo a lo que dice la ley.

Capítulo II: Análisis de Involucrados

2.01. Requerimientos

2.01.1 Descripción del sistema actual

En la actualidad las funciones que se realizan en la dependencia de tesorería son de manera manual y aislada, llevando el control en libros de Excel, también lo hacen de manera escrita en cuadernos o libretas de apuntes. Ocasionando muchas veces confusiones, malos entendidos u omisiones al momento de realizar pagos y cobros; esto deriva en los ya conocidos descuadres en caja bancos. A su vez el tesorero lleva la peor parte ya que no está con la caja y bancos cuadrados y deja trabajo pendiente para seguir organizando o cuadrando saldos de días y en algunos casos de meses anteriores.

Debido a estas situaciones el tesorero no puede estar al día con su gestión de cobros pagos y bancos.

2.01.2 Visión y Alcance

2.01.2.1 Visión

El presente proyecto está enfocado en ser una herramienta de apoyo para el departamento de tesorería, y mejorar los procesos de control de bancos caja cuentas por pagar y cobrar mediante una página Web.

El desarrollo del sistema tiene el propósito de tener una plataforma informática la cual permita gestionar de manera ágil los reportes de los bancos las cuentas por cobrar, las cuentas por pagar y la caja de la empresa, teniendo reportes actualizados del efectivo en caja y saldo en los bancos para de esta manera realizar un plan de pagos acorde a la situación financiera de la empresa, así como también reportes de las cuentas por cobrar vencidas así como también las que se encuentran por vencerse.

2.01.2.1 Alcance

El alcance de este proyecto nos permitirá llevar el control de las diferentes cuentas que maneja el módulo de tesorería las cuales, así como también manejar reportes tanto impresos como digitales de los mismos.

Tenemos registro de facturas, registro de notas de crédito, registro de anticipos a proveedores, pagos a proveedores, cuadro de anticipos a proveedores, así como también información de los bancos.

Dentro de los reportes dispondremos de información reporte de retenciones, reporte de cuentas por cobrar, reporte de cuentas por pagar, reporte de anticipos

De esta manera se cumplirá las expectativas de los empresarios cumpliendo con los requerimientos solicitados, aplicando los estándares, normas y políticas de las empresas Ecuatorianas.

2.01.3 Entrevistas

Tabla 2

Formato de entrevista

| Preguntas | Objetivos | Análisis posterior |
|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ¿Dentro de su estructura administrativa tienen un software que ayude a la tesorería? | Determinar si la empresa dispone de aplicativos específicamente dedicados para ese departamento. | Se quiere mejorar el proceso de control en bancos caja cuentas por cobrar y pagar. |
| ¿Cómo se lleva el control de bancos, caja, cuentas por cobrar y cuentas por pagar? | Obtener información detallada de los procesos del área de tesorería. | Adaptar las necesidades en la aplicación propuesta |
| ¿Dispone reportes actualizados de parte del tesorero de su empresa? | Saber si la información está actualizada. | Obtener información veraz en tiempo real |
| ¿Qué espera del módulo de tesorería luego de su implementación? | Cumplir con las expectativas del o los usuarios. | Diseñar una interfaz amigable para el usuario, y este debe funcionar tranquilamente en los navegadores más usados como Chrome, Internet Explorer, Mozilla y Opera. |

Nota: Preguntas realizadas en la entrevista dirigida al personal del nivel directivo o gerencial.

2.01.4 Matriz de Requerimientos

Tabla 3

Diseño de requerimientos

| Identificador | Descripción | Fuente | Prioridad | Tipo | Estado | Usuarios Involucrados |
|--------------------------------------|------------------------------------------------------------|----------------------|-----------|---------|----------------|---------------------------------|
| REQUERIMIENTOS FUNCIONALES | | | | | | |
| RF001 | Control de caja. | Tesorero | Alta | Sistema | En revisión | Cajero Tesorero |
| RF002 | Control de bancos. | Tesorero Contador | Alta | Sistema | En revisión | Tesorero Contador Gerente |
| RF003 | Cuentas por cobrar actualizadas | Tesorero | Alta | Sistema | En revisión | Tesorero Contador Gerente |
| RF004 | Cuentas por pagar actualizadas | Tesorero | Alta | Sistema | En revisión | Tesorero Contador Gerente |
| REQUERIMIENTOS NO FUNCIONALES | | | | | | |
| RNF001 | Interfaz amigable con botones y listas desplegables. | Gerente | Alta | Usuario | En revisión | Gerente |
| RNF002 | Compatibilidad con los diferentes navegadores: | Gerente Tesorero | Medi a | Usuario | En revisión | Gerente Tesorero Cajero |
| RNF003 | Uso de la infraestructura existente en la empresa. | Gerente | Alta | Usuario | En revisión | Gerente Tesorero |
| RNF004 | Reportes en formatos como pdf, xls, xml, así como físicos. | Gerente | Alta | Usuario | En revisión | Tesorero Gerente |

Nota: Matriz en forma detallada del levantamiento del listado de requerimientos funcionales y no funcionales planteados por personal de la institución los cuales son validados y que se utilizan para realizar el sistema informático solicitado, indicando los puntos claves que debe tener el sistema ante el usuario que lo utilizara.

2.01.5 Descripción Detallada

En el siguiente apartado se encuentra de forma detallada en tablas cada uno de los requerimientos funcionales y no funcionales presentados para el desarrollo del sistema planteado.

Requerimientos Funcionales

Tabla 4

Requerimiento funcional 001

| Descripción del Requerimiento: Control de caja. | | Estado | Análisis |
|-------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|------------|
| Creado por: | Luis Trejo | Actualizado por: | Luis Trejo |
| Fecha de Creación: | 09/11/2014 | Fecha de Actualización: | 12/03/2015 |
| Identificador: | RF001 | | |
| Tipo de Requerimiento: | Alto | Tipo de Requerimiento: | Funcional |
| Datos de Entrada: | Ingreso del cobro de las ventas sean en efectivo cheque o transferencia. | | |
| Descripción: | El cajero ingresara el medio de pago con el respectivo número de factura. | | |
| Datos de Salida: | Reportes acorde a la información ingresada. | | |
| Resultados Esperados: | Cuadre de caja diario. | | |
| Origen: | Tesorero | | |
| Dirigido a: | Cajero, Tesorero | | |
| Prioridad: | 4 | | |
| Requerimientos asociados: | RF001 | | |
| ESPECIFICACIÓN: | | | |
| Pre condiciones: | Para cumplir con el requerimiento el cajero y tesorero deben ingresar al sistema | | |
| Post condiciones: | Dentro del sistema el cajero y tesorero podrán ingresar información. Si los usuarios olvidan sus credenciales de acceso pueden solicitar al administrador que genere claves nuevas. | | |
| Criterios de Aceptación: | Ayuda a tener reportes actualizados de las ventas y cobros que se realicen desde caja. | | |

Nota: Descripción detallada del requerimiento funcional 001

Tabla 5

Requerimiento funcional 002

| Descripción del Requerimiento: Control de bancos. | | Estado | | Análisis | |
|---------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|------------|----------|--|
| Creado por: | Luis Trejo | Actualizado por: | Luis Trejo | | |
| Fecha de Creación: | 09/11/2014 | Fecha de Actualización: | 12/03/2015 | | |
| Identificador: | RF002 | | | | |
| Tipo de Requerimiento: | Alto | Tipo de Requerimiento: | Funcional | | |
| Datos de Entrada: | Ingreso de cada una de las transacciones bancarias. | | | | |
| Descripción: | El tesorero identificará cada uno de los movimientos bancarios sean estos cobro o pagos. | | | | |
| Datos de Salida: | Reportes actualizados de los estados bancarios. | | | | |
| Resultados Esperados: | Las cuentas en los bancos con saldos positivos. | | | | |
| Origen: | Tesorero | | | | |
| Dirigido a: | Tesorero, contador, gerente | | | | |
| Prioridad: | 4 | | | | |
| Requerimientos asociados: | RF002 | | | | |
| ESPECIFICACIÓN: | | | | | |
| Pre condiciones: | Tesorero debe ingresar al sistema. Los saldos del banco deben estar iguales tanto en el módulo como en la o las respectivas cuentas bancarias. | | | | |
| Post condiciones: | Visualización de saldos contables y disponibles de las cuentas. | | | | |
| Criterios de Aceptación: | Ayuda a realizar plan de pagos a proveedores. | | | | |

Nota: Descripción detallada del requerimiento funcional 002, para llevar el control en los bancos. En el cual se deberá verificar los datos que deben estar presentes en el sistema para su desarrollo.

Tabla 6

Requerimiento funcional 003

| Descripción del Requerimiento: Cuentas por cobrar actualizadas. | | Estado | Análisis |
|-----------------------------------------------------------------|------------|--------------------------------|------------|
| Creado por: | Luis Trejo | Actualizado por: | Luis Trejo |
| Fecha de Creación: | 09/11/2014 | Fecha de Actualización: | 12/03/2015 |

| | |
|----------------------------------|----------------------------------------------------------------------------------------------------------|
| Identificador: | RF003 |
| Tipo de Requerimiento: | Alto |
| Datos de Entrada: | Ingreso de las cuentas por cobrar de clientes que tienen crédito con la empresa. |
| Descripción: | El tesorero ingresara el tiempo del crédito concedido al cliente con el respectivo número de factura. |
| Datos de Salida: | Reportes con cuentas por cobrar que ya están vencidas o que están por vencerse. |
| Resultados Esperados: | Cobros más eficientes. |
| Origen: | Tesorero |
| Dirigido a: | Tesorero, contador, gerente |
| Prioridad: | 4 |
| Requerimientos asociados: | RF003 |
| ESPECIFICACIÓN: | |
| Pre condiciones: | Para cumplir con el requerimiento el tesorero debe ingresar al sistema |
| Post condiciones: | El tesorero al día en la gestión de cobros. |
| Criterios de Aceptación: | El tesorero puede entregar reportes oportunos de cuentas vencidas y así no declararlas como incobrables. |

Nota: Descripción detallada del requerimiento funcional 003, para llevar el control en cuentas por cobrar.

Tabla 7

Requerimiento funcional 004

| Descripción del Requerimiento: Cuentas por pagar actualizadas. | | Estado | Análisis |
|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|--------------------------------|------------|
| Creado por: | Luis Trejo | Actualizado por: | Luis Trejo |
| Fecha de Creación: | 09/11/2014 | Fecha de Actualización: | 12/03/2015 |
| Identificador: | RF004 | | |
| Tipo de Requerimiento: | Alto | Tipo de Requerimiento: | Funcional |
| Datos de Entrada: | Ingreso de las cuentas por pagar a proveedores de la empresa. | | |
| Descripción: | El tesorero ingresara el detalle de los pagos con su respectivo número de factura. | | |
| Datos de Salida: | Reportes con cuentas por pagar que ya están vencidas o que están por vencerse. | | |
| Resultados Esperados: | Puntualidad en los pagos a proveedores. | | |
| Origen: | Tesorero | | |
| Dirigido a: | Tesorero, contador, gerente | | |
| Prioridad: | 4 | | |
| Requerimientos asociados: | RF004 | | |
| ESPECIFICACIÓN: | | | |
| Pre condiciones: | Para cumplir con el requerimiento el tesorero debe ingresar al sistema | | |
| Post condiciones: | El tesorero al día en la gestión de pagos. | | |
| Criterios de Aceptación: | El tesorero puede entregar reportes oportunos de pagos vencidos y así mantener una buena línea de crédito con los proveedores. | | |

Nota: Descripción detallada del requerimiento funcional 004, para llevar el control en cuentas por pagar. En el cual se deberá verificar los datos que deben estar presentes en el sistema para su desarrollo

Requerimientos No Funcionales.

Tabla 8

Requerimiento no funcional 001

| Descripción del Requerimiento: Interfaz amigable con botones y listas desplegables. | | Estado | Análisis |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|--------------------------------|------------|
| Creado por: | Luis Trejo | Actualizado por: | Luis Trejo |
| Fecha de Creación: | 09/11/2014 | Fecha de Actualización: | 12/03/2005 |
| Identificador: | RNF001 | | |
| Tipo de Requerimiento: | Alto | Tipo de Requerimiento: | Funcional |
| Datos de Entrada: | Usuario debe distinguir con facilidad las funciones de cada botón y lista desplegable para manejar el módulo. | | |
| Descripción: | El sistema dispondrá de un adecuado manual de usuario. | | |
| Datos de Salida: | Usuario debe guardar los cambios que realice en el sistema. | | |
| Resultados Esperados: | Uso adecuado del sistema en un ambiente amigable. | | |
| Origen: | Gerencia | | |
| Dirigido a: | Tesorero, cajero, gerente | | |
| Prioridad: | 4 | | |
| Requerimientos asociados: | RNF001 | | |
| ESPECIFICACIÓN: | | | |
| Pre condiciones: | El usuario recibirá capacitación para el uso del software | | |
| Post condiciones: | Usuario contara con soporte en sitio o vía remota. | | |
| Criterios de Aceptación: | Interfaz amigable con los usuarios. | | |

Nota: Descripción detallada del requerimiento no funcional 001, para el diseño del módulo. En el cual se deberá verificar los datos que deben estar presentes en el sistema para su desarrollo

Tabla 9

Requerimiento no funcional 002

| Descripción del Requerimiento: Compatibilidad con los diferentes navegadores. | | Estado | Análisis |
|-------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|--------------------------------|------------|
| Creado por: | Luis Trejo | Actualizado por: | Luis Trejo |
| Fecha de Creación: | 09/11/2014 | Fecha de Actualización: | 12/03/2015 |
| Identificador: | RNF002 | | |
| Tipo de Requerimiento: | Alto | Tipo de Requerimiento: | Funcional |
| Datos de Entrada: | Usuario debe digitar dirección Web o una dirección Ip para acceder al sistema. | | |
| Descripción: | El sistema funcionara en un ambiente Web. | | |
| Datos de Salida: | Cerrar navegador Web. | | |
| Resultados Esperados: | Uso adecuado del sistema en un ambiente amigable. | | |
| Origen: | Gerencia | | |
| Dirigido a: | Tesorero, cajero, gerente | | |
| Prioridad: | 4 | | |
| Requerimientos asociados: | RNF002 | | |
| ESPECIFICACIÓN: | | | |
| Pre condiciones: | El navegador debe estar actualizado | | |
| Post condiciones: | Posibles fallas debido a factores externos al módulo como caída de la red o del servicio de internet. | | |
| Criterios de Aceptación: | Interfaz amigable con los usuarios. | | |

Nota: Descripción detallada del requerimiento no funcional 002, para el diseño del módulo. En el cual se deberá verificar los datos que deben estar presentes en el sistema para su desarrollo

Tabla 10

Requerimiento no funcional 003

| Descripción del Requerimiento: Uso de la infraestructura existente en la empresa. | | Estado | Análisis |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|--------------------------------|------------|
| Creado por: | Luis Trejo | Actualizado por: | Luis Trejo |
| Fecha de Creación: | 09/11/2014 | Fecha de Actualización: | 12/03/2015 |
| Identificador: | RNF003 | | |
| Tipo de Requerimiento: | Alto | Tipo de Requerimiento: | Funcional |
| Datos de Entrada: | Usuario usará el recurso tecnológico existente. | | |
| Descripción: | Los requerimientos de instalación del módulo estarán acorde a la infraestructura existente. | | |
| Datos de Salida: | La inversión en tecnología será mínima. | | |
| Resultados Esperados: | Uso satisfactorio de los recursos tecnológicos de la empresa. | | |
| Origen: | Gerencia | | |
| Dirigido a: | Tesorero, cajero, gerente | | |
| Prioridad: | 4 | | |
| Requerimientos asociados: | RNF003 | | |
| ESPECIFICACIÓN: | | | |
| Pre condiciones: | Si la infraestructura no es la adecuada el módulo no podría funcionar correctamente | | |
| Post condiciones: | Se sugerirá actualizar la infraestructura tecnológica. | | |
| Criterios de Aceptación: | Disponer de equipos actualizados. | | |

Nota: Descripción detallada del requerimiento no funcional 003, para el diseño del módulo. En el cual se deberá verificar los datos que deben estar presentes en el sistema para su desarrollo

Tabla 11

Requerimiento no funcional 004

| Descripción del Requerimiento: Reportes en formatos como pdf, xls, xml, así como físicos. | | Estado | Análisis |
|-------------------------------------------------------------------------------------------|------------|--------------------------------|------------|
| Creado por: | Luis Trejo | Actualizado por: | Luis Trejo |
| Fecha de Creación: | 09/11/2014 | Fecha de Actualización: | 12/03/2015 |

| | | | |
|----------------------------------|--------------------------------------------------------------------------------------------------------|-------------------------------|-----------|
| Identificador: | RNF004 | | |
| Tipo de Requerimiento: | Alto | Tipo de Requerimiento: | Funcional |
| Datos de Entrada: | Usuario definirá el formato en el que desea el reporte. | | |
| Descripción: | La facilidad de presentar reporte de acuerdo a la necesidad del usuario. | | |
| Datos de Salida: | Programas que usen los formatos requeridos así como también el uso de impresoras. | | |
| Resultados Esperados: | Variedad de opciones al momento de presentar reportes. | | |
| Origen: | Gerencia | | |
| Dirigido a: | Tesorero, cajero, gerente | | |
| Prioridad: | 4 | | |
| Requerimientos asociados: | RNF004 | | |
| ESPECIFICACIÓN: | | | |
| Pre condiciones: | Los computadores deberán tener instalados los programas que permitan la visualización de los reportes. | | |
| Post condiciones: | Verificar si los computadores tienen instalados estos programas. | | |
| Criterios de Aceptación: | Disponer los reportes en el formato solicitado. | | |

Nota: Deberá verificar los datos que deben estar presentes en el sistema para su desarrollo

2.02 Mapeo de Involucrados.

Identificación de involucrados directos e indirectos que intervienen en el negocio.

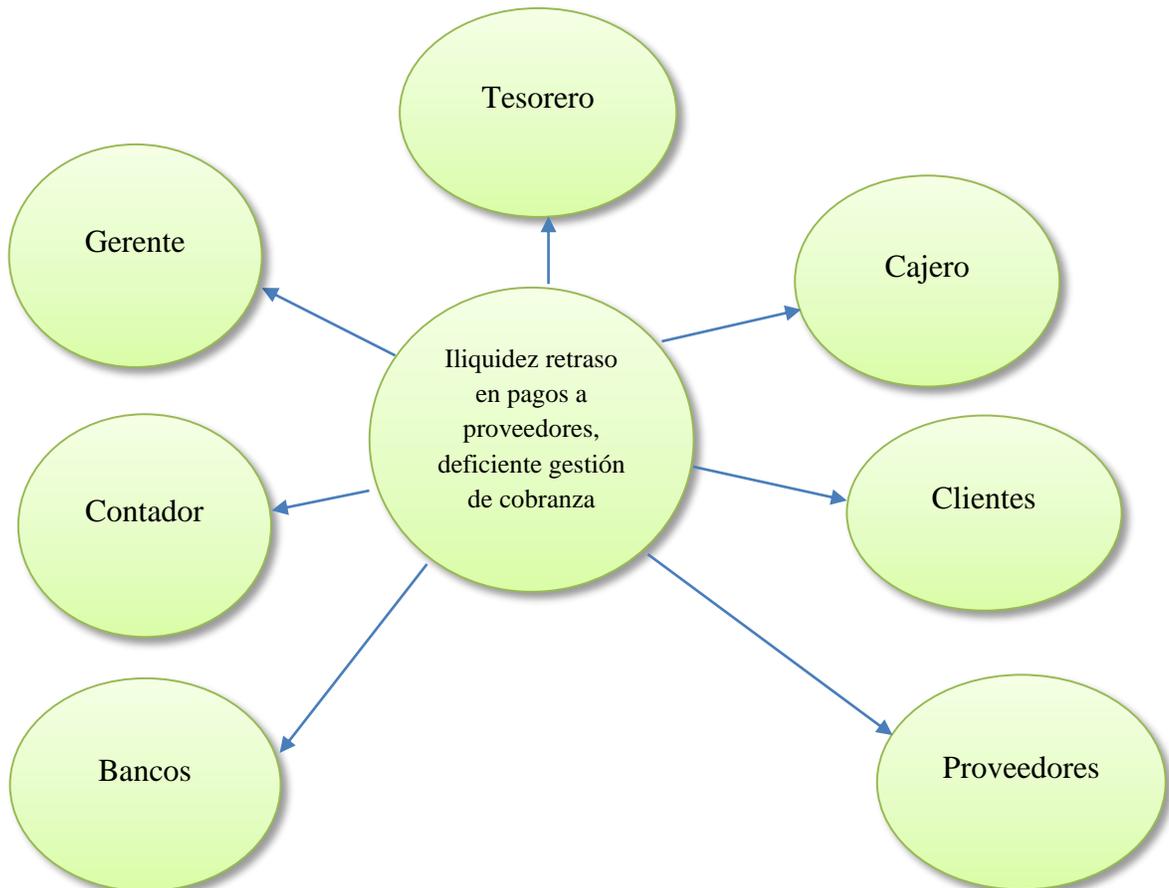


Figura 1 *Mapeo de involucrados*. Descripción de los actores directos e indirectos que influyen en la institución.

2.03. Matriz de Involucrados

Tabla 12

Detalle de interés, involucrados y conflictos percibidos en el proyecto.

| Actores Involucrados | Interés sobre el problema central | Problemas Percibidos | Recursos, Mandatos y Capacidades | Interés sobre el Proyecto | Conflictos Potenciales |
|----------------------|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|-----------------------------------------------------|
| Tesorero | Generar liquidez agilidad en los cobros, evitar retrasos en los pagos a proveedores, tener la caja cuadrada correctamente. | Cuentas por cobrar y pagar vencidas, molestias en los proveedores falta de información actualizada. | Se necesita una herramienta la cual ayude a gestionar las actividades de tesorería con reportes actualizados | Al disponer un módulo de tesorería mejorará el apoyo a las actividades de la empresa. | Temor a usar la herramienta. |
| Cajero | Registrar los ingresos y egresos en tiempo real. | Actualmente se registran ingresos y egresos de manera manual. | Consulta de movimientos realizados desde la caja con reportes | Disminuir el tiempo que toma el cuadro de caja. | Error al digitar los valores de ingresos y egresos. |
| Clientes | Mejorar el servicio al cliente. | Cartera por cobrar vencida. | Se necesita consultar la lista de clientes que tienen sus cuentas vencidas para con la empresa. | El módulo mejorara la gestión de cobranza. | Clientes morosos. |
| Proveedores | Se mantendrá al día en los pagos. | Falta de liquidez por fallas de la gestión de cobranza. | El módulo reportara de manera anticipada los pagos que ya están por vencerse | Se mantendrán abiertas las líneas de crédito con los proveedores. | Iliquidez de la empresa. |
| Bancos | Control sobre las cuentas bancarias. | Escaso control de la cuentas bancarias, emisión de cheques sin fondos | Información del detalle de los cheques emitidos y los saldos | Cuentas bancarias sólidas. | Emisión de cheques sin los fondos que los respalden |
| Contador | Tener reportes los cuales ayuden a su gestión | No disponer de información actualizada. | Dispondrá de información veraz y actualizada | Alivianar la carga de trabajo | Apego a aplicaciones antiguas. |
| Gerente | Se requiere tener reportes apropiados de los estados financieros de la Empresa | No es posible tener reportes rápidos de toda la información que se maneja en la empresa | Es quien necesita mejorar los tiempos de respuesta en el manejo de información de la Empresa | Gran interés en la implementación del sistema, con esto agilizarán notablemente los procesos | Falta de interés hacia las nuevas tecnologías |

Nota Matriz de Involucrados. Descripción clara y precisa del análisis de los involucrados los cuales intervienen directa e indirectamente en la ejecución de los procesos de la institución para el cumplimiento de los objetivos planteados y que se van cubrir con la implementación del presente sistema.

Capítulo III: Problemas y Objetivos

3.01. Árbol de Problemas

Nos permite identificar problemas asociados al módulo de tesorería y así analizar las causas, problema y Efecto, a la situación asociada al tema.

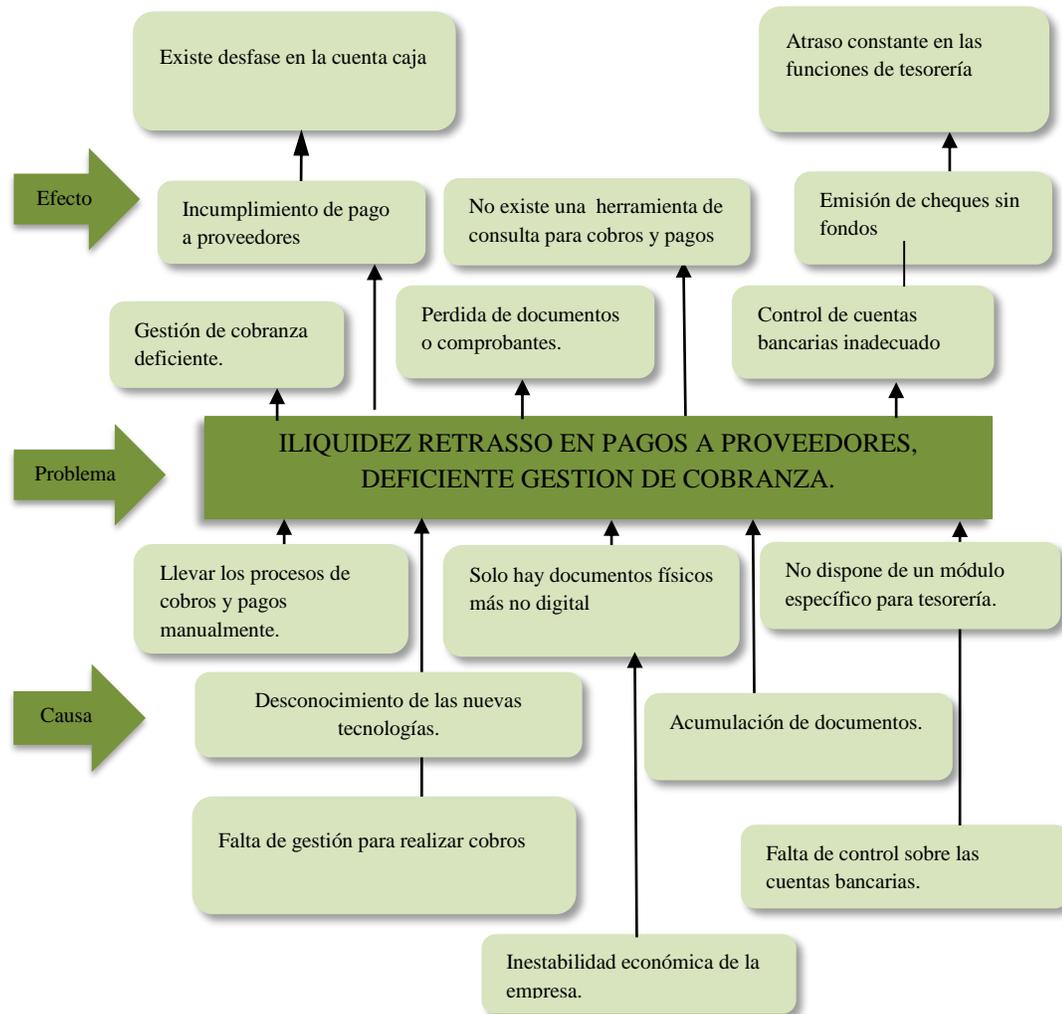


Figura 2 . *Árbol de Problemas*. Descripción detallada de los problemas los cuales motivan al desarrollo del sistema

3.0.2 Árbol de Objetivos

En base al árbol de problemas se revisa cada punto negativo y convertir en objetivo (positivo) realista y deseable., de esta manera las causas son medios y los efectos son fines.

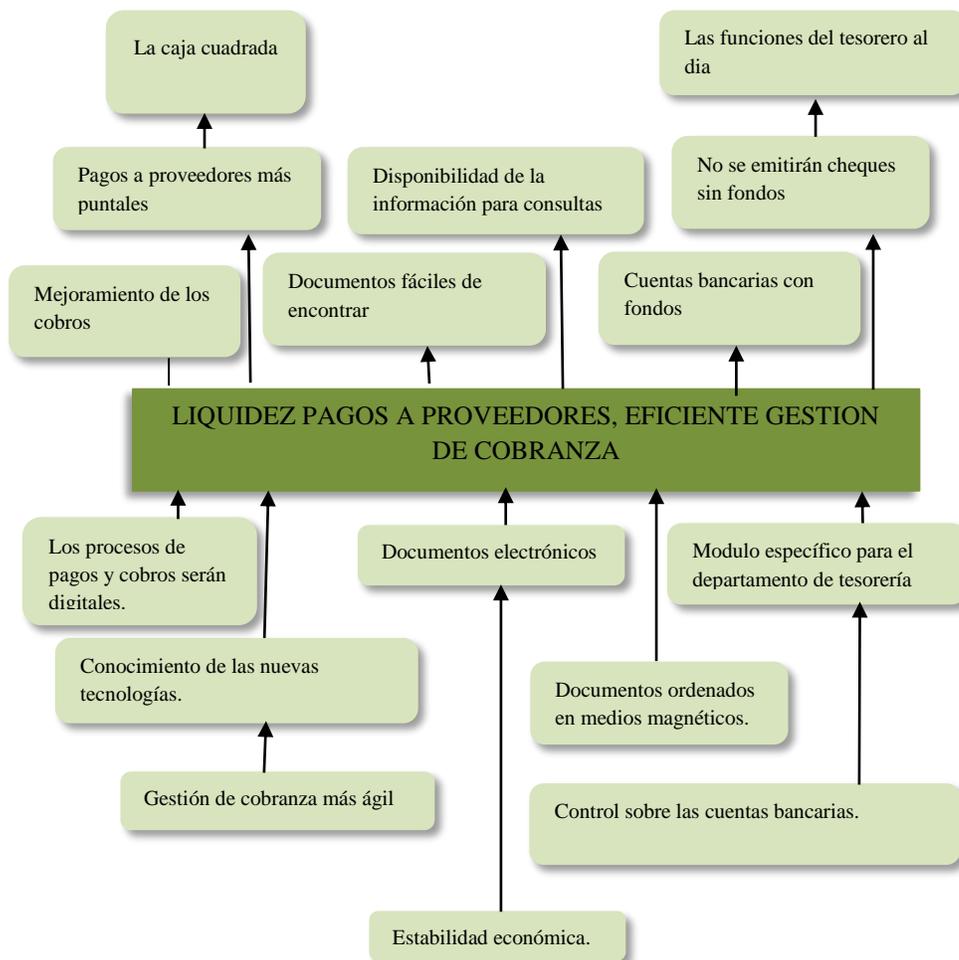


Figura 3 *Árbol de objetivos*. Descripción detallada del árbol de objetivos que deben cumplirse al desarrollar el sistema

3.03. Diagramas de casos de uso

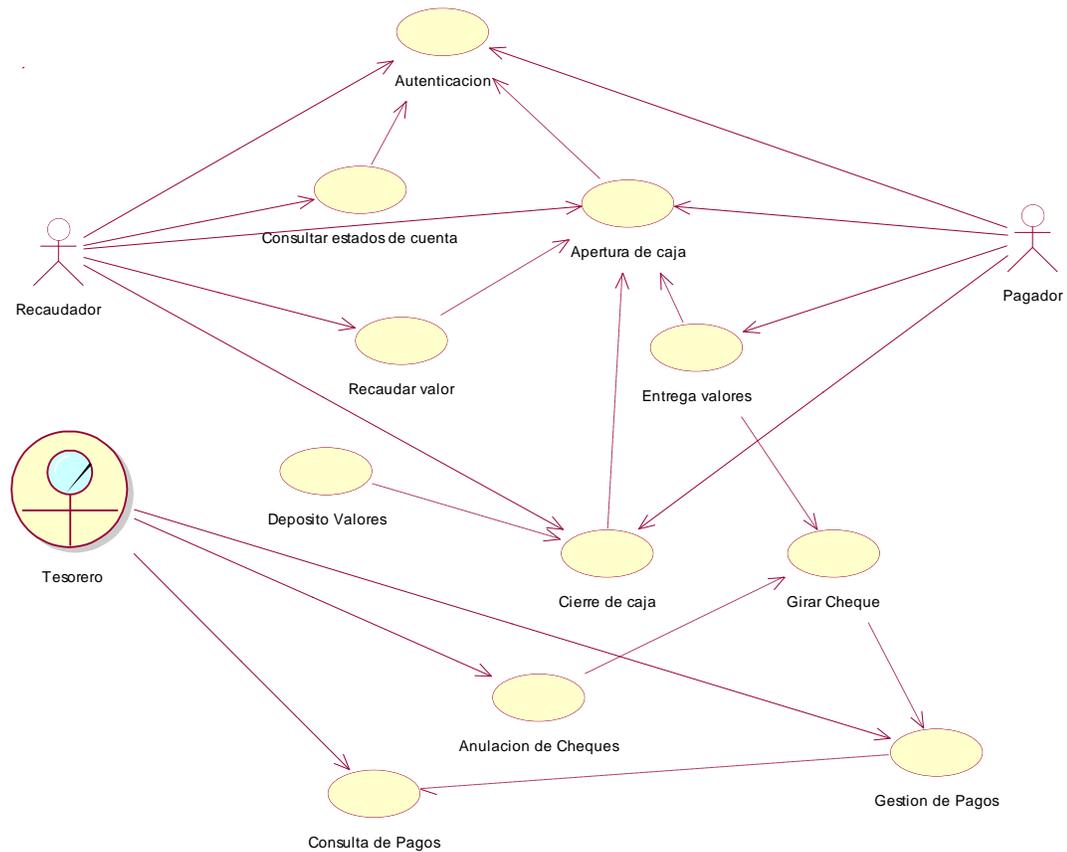


Figura 4 *Diagrama de Caso de Uso General 001*. En la siguiente figura se muestra la vista en detalle del caso de uso general de los autores implicados y sus relaciones en el negocio del sistema. Donde se indica los procesos a ejecutarse. Este diagrama es fundamental ya que nos ayuda a visualizar el funcionamiento en el modelado y organización del sistema.

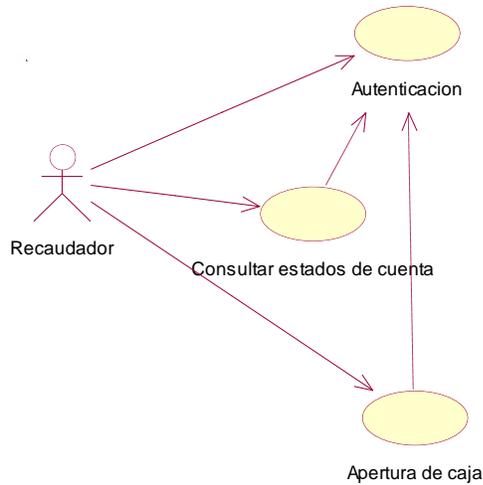


Figura 5 *Diagrama de Caso de Uso 002*. Vista detallada del caso de uso donde el usuario ingresa al módulo y así empezar a usarlo.

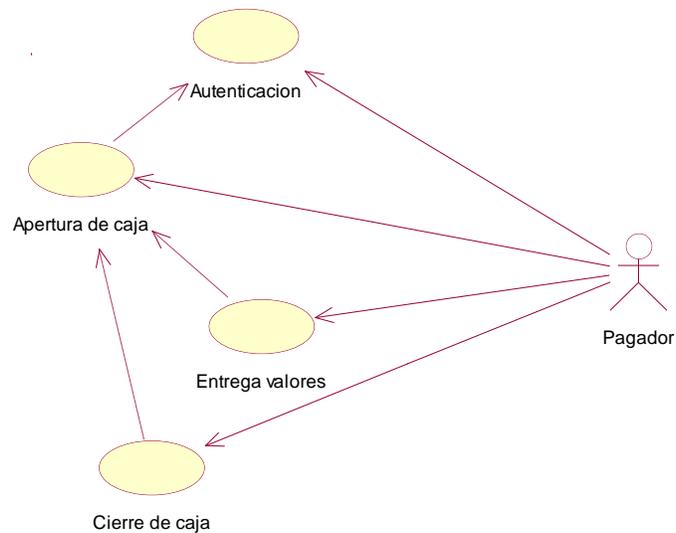


Figura 6 *Diagrama de Caso de Uso 003*. Vista detallada del caso de uso donde el usuario hace el proceso de pagos.

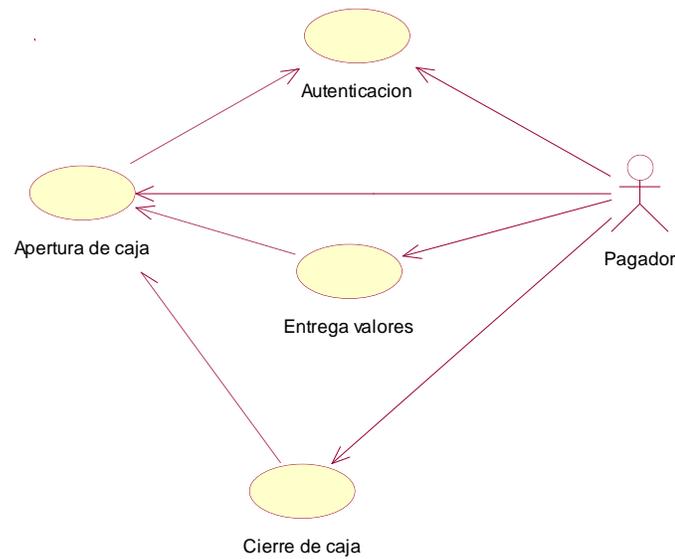


Figura 7 Diagrama de Caso de Uso 004. Vista detallada del caso de uso donde el usuario hace el proceso de cuadre de caja.

3.04. Casos de uso de Realización

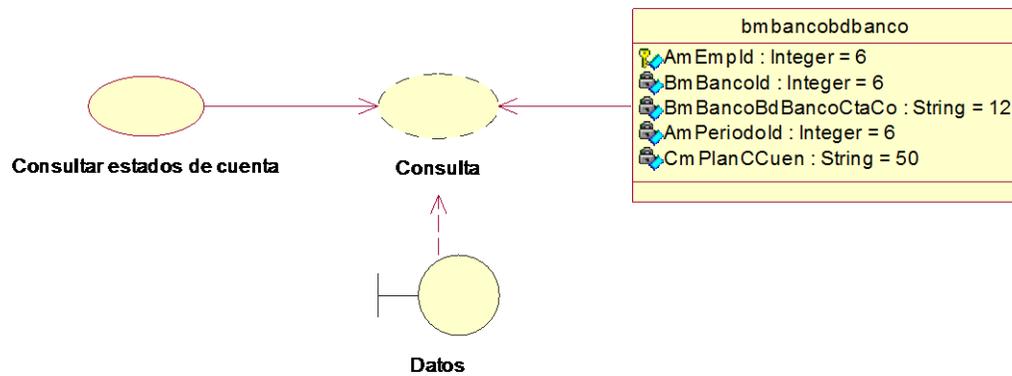


Figura 8 Diagrama de Caso de Uso de Realización 001. Vista detallada del caso de uso de realización para realizar la consulta de las cuentas bancarias.

Tabla 13

Bancos

| | |
|----------------------------------------------------------------------------------------|--------------------------------------------------|
| Nombre | Elaboración de consulta de las cuentas bancarias |
| Identificador | Caso: UC001 |
| Responsabilidades | Consulta de cuentas bancarias |
| Tipo | Sistema |
| Referencias Casos de Usos | Caso: UC001 |
| Referencias Requisitos | RE001 |
| PRECONDICIONES | |
| 1.-Solicitar al contador el ingreso de las cuentas bancarias. | |
| POSCONDICIONES | |
| Muestra nombre del Banco y su número de cuenta respectivo | |
| SALIDAS PANTALLA | |
| Las cuentas bancarias en menú desplegable para seleccionar la cuenta que se va a usar. | |

Nota: Descripción detallada del caso de uso de realización 001, donde verifica las cuentas bancarias.

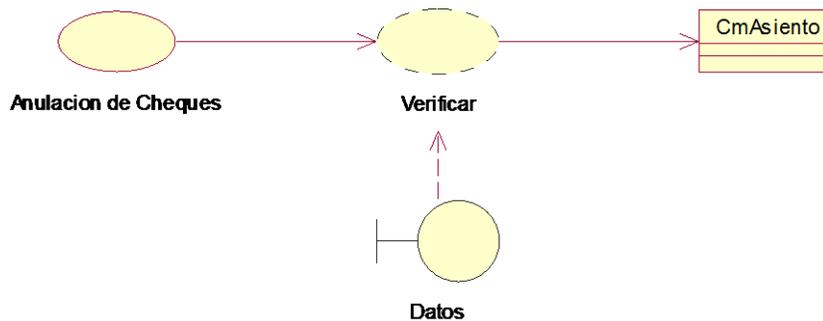


Figura 9 Diagrama de Caso de Uso de Realización 002. Vista detallada del caso de uso de realización para realizar la anulación de cheques.

Tabla 14

Anulación de cheques

| | |
|---------------------------------------------------------------------------------------|---------------------------------------------|
| Nombre | Elaboración de consulta de cheques anulados |
| Identificador | Caso: UC002 |
| Responsabilidades | Consulta de cheques anulados |
| Tipo | Sistema |
| Referencias Casos de Usos | Caso: UC002 |
| Referencias Requisitos | RE002 |
| PRECONDICIONES | |
| 1.-Solicitar el cheque a ser anulado. 1.-Verifica los datos para anular el cheque. | |
| POSCONDICIONES | |
| Guarda el detalle del cheque anulado | |
| SALIDAS PANTALLA | |
| Lista de los cheques anulados por cada banco | |

Nota: Descripción detallada del caso de uso de realización 002, donde se anula cheques.

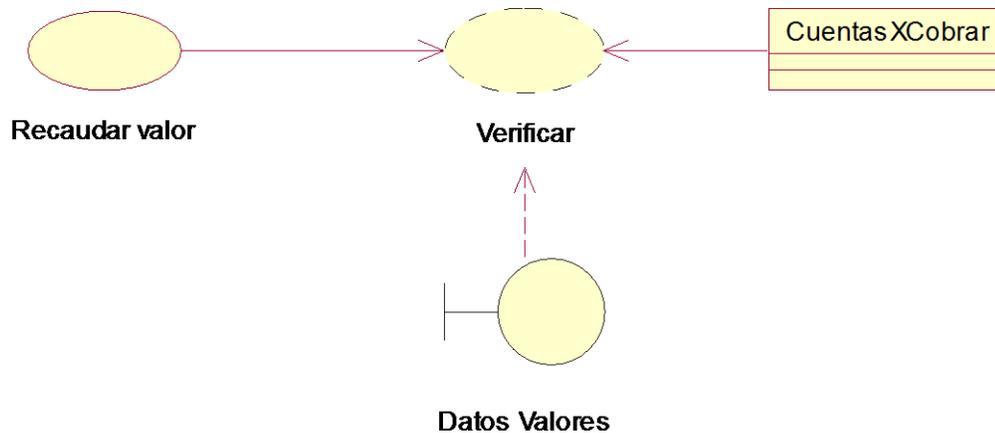


Figura 10. *Diagrama de Caso de Uso de Realización 003. Vista detallada del caso de uso de realización para realizar cobros.*

Tabla 15

Cuentas por cobrar

| | |
|-----------------------------------------------------------------------------|-----------------------------------------------|
| Nombre | Elaboración de consulta de cuentas por cobrar |
| Identificador | Caso: UC003 |
| Responsabilidades | Consulta de cuentas por cobrar |
| Tipo | Sistema |
| Referencias Casos de Usos | Caso: UC003 |
| Referencias Requisitos | RE003 |
| PRECONDICIONES | |
| 1.-Ingreso de las facturas de venta. 1.-Verificar el pago de la factura. | |
| POSCONDICIONES | |
| Guarda el detalle de cada factura con el crédito | |
| SALIDAS PANTALLA | |
| Lista de las facturas pendientes de cobro | |

Nota: Descripción detallada del caso de uso de realización 003, realiza la gestión de cobros

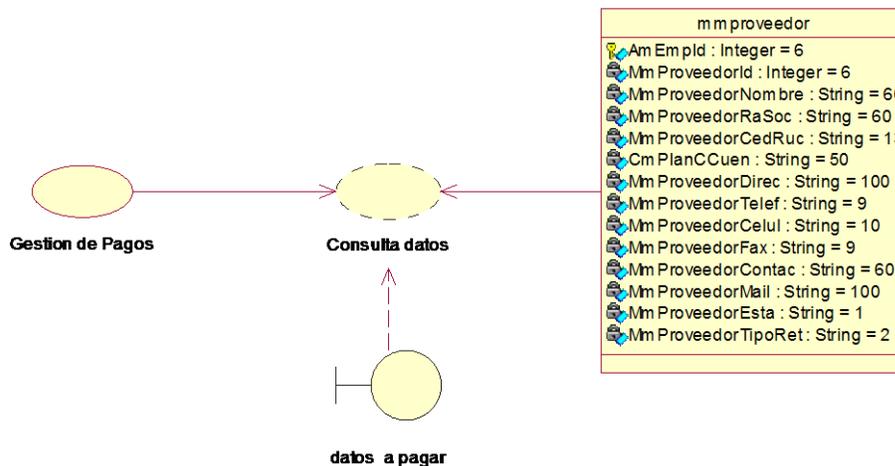


Figura 11 *Diagrama de Caso de Uso de Realización 004. Vista detallada del caso de uso de realización para realizar pagos.*

Tabla 16

Cuentas por pagar

| | |
|--------------------------------------------------------------|----------------------------------------------|
| Nombre | Elaboración de consulta de cuentas por pagar |
| Identificador | Caso: UC004 |
| Responsabilidades | Consulta de cuentas por cobrar |
| Tipo | Sistema |
| Referencias Casos de Usos | Caso: UC004 |
| Referencias Requisitos | RE004 |
| PRECONDICIONES | |
| 1.-Ingreso de las facturas compra. | |
| 1.-Verificar el tiempo de crédito otorgado por el proveedor. | |
| POSTCONDICIONES | |
| Guarda el detalle de cada factura de compra con el crédito | |
| SALIDAS PANTALLA | |
| Lista de las facturas pendientes de pago | |

Nota: Descripción detallada del caso de uso de realización 004, realiza la gestión de pagos

3.05. Diagrama de secuencias del sistema

El siguiente Diagrama de Secuencia (Sequence Diagram) nos permitirá mostrar la interacción de los conjuntos de objetos de la aplicación a desarrollarse a través del tiempo y se modela base al diagrama de casos de uso, aquí se da a conocer los procesos que ejecutarían en el módulo.

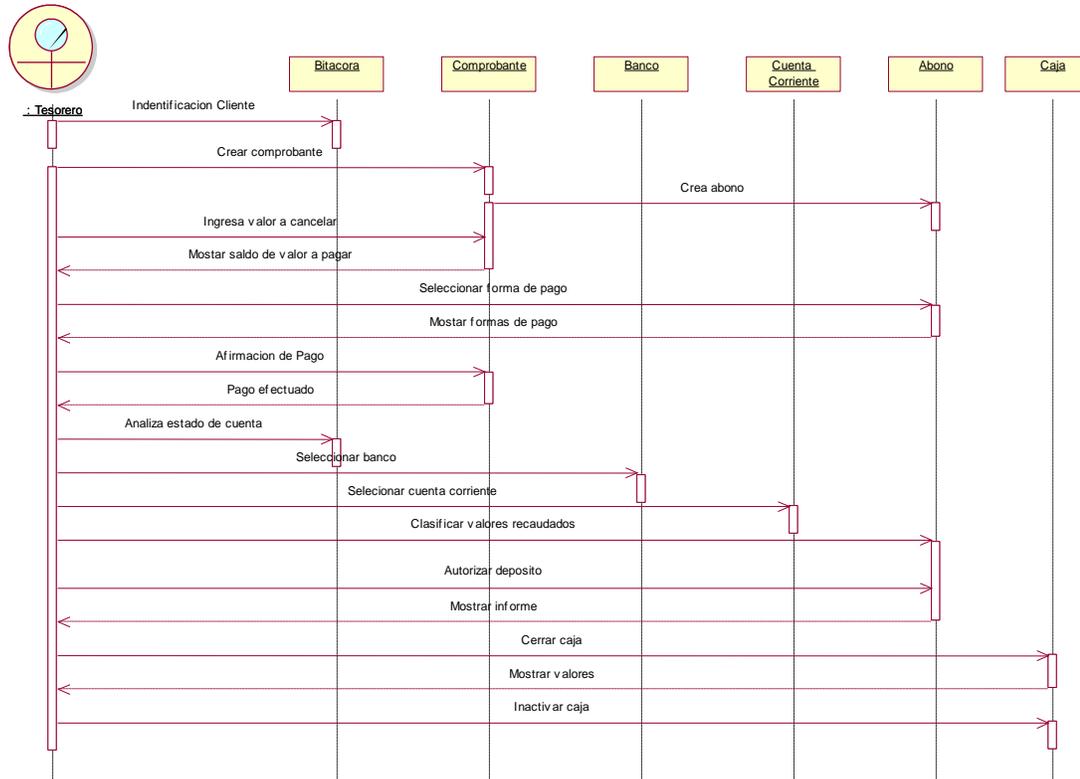


Figura 12 En este proceso el tesorero realiza el control y los reportes de las transacciones que realiza el módulo.

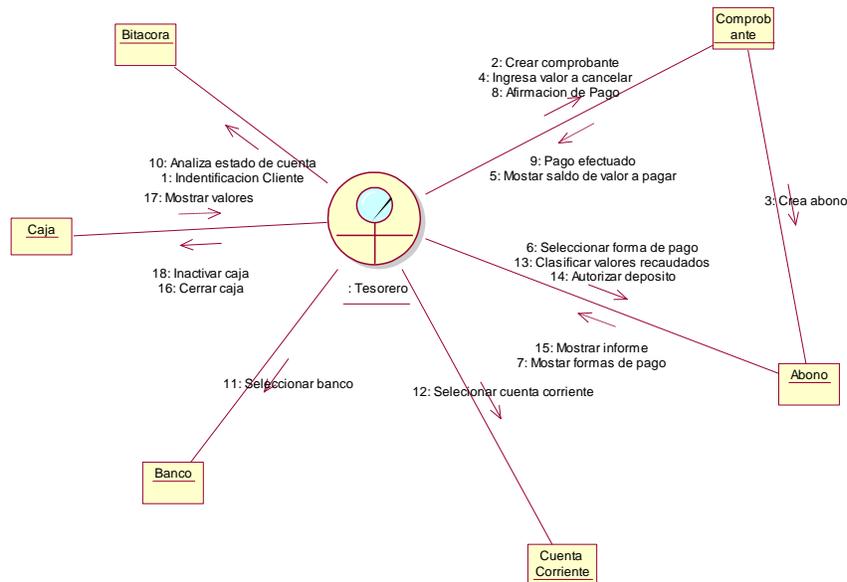


Figura 13. Se detalla la secuencia en la cual el tesorero maneja bancos cuentas por cobrar y cuentas por pagar.

3.06. Especificación de casos de uso

Tabla 17

Especifica casos de uso general.

| | |
|--------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| Caso de uso | Caso de uso general |
| Identificador | Caso: UCR001 |
| CURSO TÍPICO DE EVENTOS | |
| Usuario | Sistema |
| 1.-Ingresa al sistema. 2.-Verificación de datos. 3.-verificación de información para realizar las actividades contables. | Realizar con consultas a la base de datos Select |
| CURSOS ALTERNATIVOS | |
| Para todo este proceso el tesorero debe contar con la información emitida por ventas. | |

Nota: Descripción detallada de la especificación del caso de uso 001, caso de uso general.

Tabla 18

Especifica caso de uso 002 de cobros.

| | |
|--------------------------------------------------------------------------------------------------------------------------|------------------------------------------|
| Caso de uso | Caso de uso 002 |
| Identificador | Caso: UCR002 |
| CURSO TÍPICO DE EVENTOS | |
| Usuario | Sistema |
| 1.-Ingresa al sistema. 2.-Ingresa facturas de venta. 3.-verificación de información para realizar las actividades. | Usuario consulta las cuentas por cobrar. |
| CURSOS ALTERNATIVOS | |
| Para todo este proceso el usuario debe ingresar con sus respectivas credenciales. | |

Nota: Descripción detallada de la especificación del caso de uso 002.

Tabla 19

Especifica caso de uso 003 de cobros.

| | |
|------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| Caso de uso | Caso de uso 003 |
| Identificador | Caso: UCR003 |
| CURSO TÍPICO DE EVENTOS | |
| Usuario | Sistema |
| 1.-Ingresa al sistema. 2.-Verificación de datos. 3.-Ingreso de facturas de compra y anticipos. | Usuario consulta las cuentas por pagar para realizar un plan de pagos. |
| CURSOS ALTERNATIVOS | |
| Para todo este proceso el usuario debe ingresar con sus respectivas credenciales. | |

Nota: Descripción detallada de la especificación del caso de uso 003.

Tabla 20

Especifica caso de uso 004 de bancos.

| | |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------|
| Caso de uso | Caso de uso 004 |
| Identificador | Caso: UCR004 |
| CURSO TÍPICO DE EVENTOS | |
| Usuario | Sistema |
| 1.-Ingresa al sistema. | Usuario consulta los cheques emitidos para ingresarlos al sistema. |
| 2.-Verificación de datos. | |
| 3.-Ingreso de cheques emitidos o transferencias realizadas. | |
| CURSOS ALTERNATIVOS | |
| Para todo este proceso el usuario debe ingresar con sus respectivas credenciales. | |

Nota: Descripción detallada de la especificación del caso de uso 004.

Capítulo IV: Análisis de Alternativas

4.01. Matriz de Análisis de Alternativas

Tabla 21

Análisis de alternativas.

| Objetivos | Impacto sobre el propósito | Factibilidad Técnica | Factibilidad Financiera | Factibilidad Social | Factibilidad Política | Total | Categoría |
|--------------------------------------------------------------|----------------------------|----------------------|-------------------------|---------------------|-----------------------|------------|------------|
| Obtener reportes actualizados de los cobros, pagos y bancos. | 5 | 5 | 3 | 3 | 2 | 18 | Alta |
| Registro de facturas de compra. | 5 | 5 | 3 | 3 | 2 | 18 | Alta |
| Agilizar la gestión de cobros. | 5 | 4 | 4 | 3 | 2 | 18 | Alta |
| Más puntualidad en el pago a proveedores. | 5 | 4 | 3 | 3 | 2 | 17 | Alta |
| Control de las cuentas bancarias y cheques emitidos. | 5 | 5 | 2 | 2 | 2 | 16 | Media Alta |
| TOTAL: | 25 | 23 | 15 | 15 | 10 | 113 | |

Nota: Descripción detallada de la matriz de análisis de alternativas, se identifica las soluciones alternativas; y evaluar la factibilidad técnica, financiera, social y política determinando la estrategia general a ser adoptada en el proyecto.

Escala: 5 => Alto = 25 | 4 => Medio Alto = 20 | 3 => Medio = 15 | 2 => Medio bajo =

10 | 1 => Bajo = 5.

El propósito de la matriz de alternativas es determinar las técnicas que permitan desarrollar las actividades, tomando en cuenta el recurso financiero, minimizando el costo ante el proyecto, amparándose en las políticas y procedimientos entregados por la institución, siendo estas de gran ayuda para automatizar los procesos.

El propósito para que los procesos y procedimientos estén bien definidos es utilizar técnicas que ayuden a agilizar de manera eficiente las políticas internas establecidas, para mantener una organización adecuada.

4.02. Matriz de Impactos de Objetivos

Tabla 22

Matriz de Análisis de Impactos de Objetivos

| Objetivos | Factibilidad de Lograse | Impacto en Género | Impacto Ambiental | Relevancia | Sostenibilidad | Total |
|--------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|---------------------------------------------------------------|-----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|-----------------------|
| | (Alta-Media-Baja) | (Alta-Media-Baja) | (Alta-Media-Baja) | (Alta-Media-Baja) | (Alta-Media-Baja) | Puntos |
| | (4-2-1) | (4-2-1) | (4-2-1) | (4-2-1) | (4-2-1) | |
| Agilidad en la ejecución de procesos y procedimientos. | 4. Los beneficios son mayores que los costos. | 4. Incrementa plazas de trabajo para todos. | 4. Contribuye a proteger el entorno del ambiente físico, minimizando la utilización de papel. | 4. Responde a las expectativas de los beneficiarios desde el nivel gerencial hasta el operacional. | 4. Fortalece la participación de los beneficiarios, capacitación en el manejo operacional. | |
| Definición de procesos y procedimientos por cada puesto de trabajo | 4. Cuenta con financiamiento propio | 4. Aumenta los ingresos | 4. Mejora el entorno social, utilizando tecnología de punta. | 4. Es una prioridad concebida por los funcionarios de la institución. | 4. Fortalece la Organización interna de la Coordinación Nacional de Nutrición. | |
| Crecimiento institucional y estabilidad en puestos de trabajo. | 5. Existe tecnología extranjera y se conoce de su funcionamiento para su adecuada aplicación. | 4. Fortalece la aplicación de la realización de las personas. | 4. Protege el uso de los recursos, humano y material | 5. Los beneficios son deseados por los funcionarios de la institución | 4. Los funcionarios están en posibilidades de aportar medios, materiales e intelectuales | 97 puntos Alta |
| Organización y control de contable | 4. Es aceptable y conveniente para los beneficiarios a las empresas | 3. Mejora el nivel educativo de los funcionarios. | 4. Mejora el entorno cultural, en base al conocimiento informático | 4. Los beneficios son deseados por los funcionarios de la institución | 4. Los funcionarios están en posibilidades de aportar medios, materiales e intelectuales | |
| Asignación de presupuesto | 4. Se cuenta con soporte técnico especializado y gerencia de costos de la Institución | 4. Incremento de valores éticos y morales de funcionarios | 4. Favorece la educación ambiental. | 3. Beneficia a los involucrados indirectos. | 21. Se puede conseguir financiamiento a futuro | |
| Puntos | 21 | 19 | 20 | 20 | 21 | |

Nota: Descripción detallada de la matriz de análisis de impactos de objetivos. En esta parte encontramos de forma detallada los objetivos y sus impactos ante factores que se presentan al realizar el proyecto.

4.03. Estándares para el Diseño de Clases

Clases, Interfaces

Para nombrar a las clases siempre usar sustantivos ejemplo: Autor.

Para las interfaces siempre usar sustantivo + adjetivo ejemplo: CampoSimple o calificativos ejemplo: Editable

El nombre de la clase o interfaz debe:

Estar en singular.

Comenzar con mayúscula.

No estar abreviado.

Ser representativo del objeto que representara, en castellano.

En caso de estar compuesto por más de una palabra, las mismas se concatenaran y cada una de ellas comenzará con mayúsculas.

Atributos

El nombre del atributo debe:

Estar en minúscula.

No estar abreviado.

Ser representativo del mismo, en castellano.

En caso de estar compuesto por más de una palabra, las mismas se concatenaran y a partir de la segunda palabra las mismas comenzarán con mayúsculas.

Ej:

Private String nombre;

Private int cantidadDeHijos;

Métodos

Adoptamos por convención programar en idioma castellano y de manera declarativa.

Esto significa que para “hablarle” a un objeto la forma es:

//Así no! Declarativo!

```
autor.esc_lib();
```

//Así no! En castellano.

```
autor.escribiUnBook();
```

//Así si.

```
autor.escribiUnLibro();
```

El nombre del método debe:

Estar en minúscula.

No estar abreviado.

Ser declarativo, en castellano.

En caso de estar compuesto por más de una palabra, las mismas se concatenaran y cada una de ellas comenzará con mayúsculas.

Ej:

Autor

Libro

Tener una sola tabla controlada por estados.

4.04. Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, orientados a objetos.

4.05. Modelo Lógico – Físico

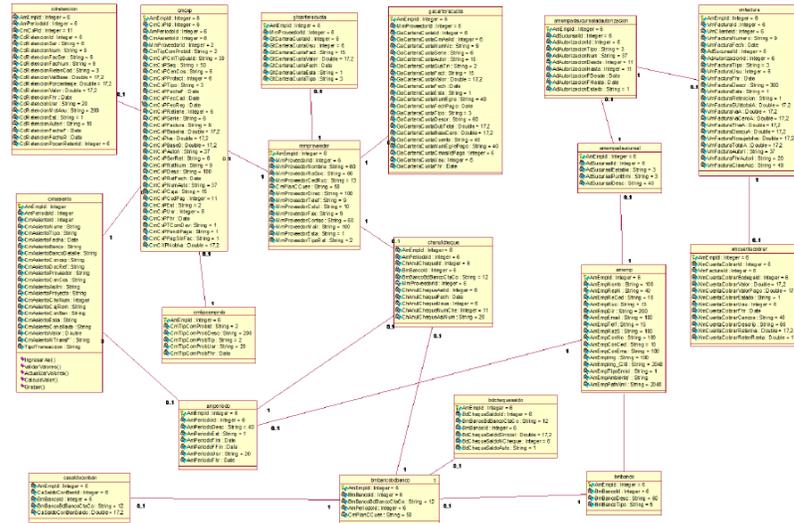


Figura 15 Vista detallada del modelo lógico del sistema.

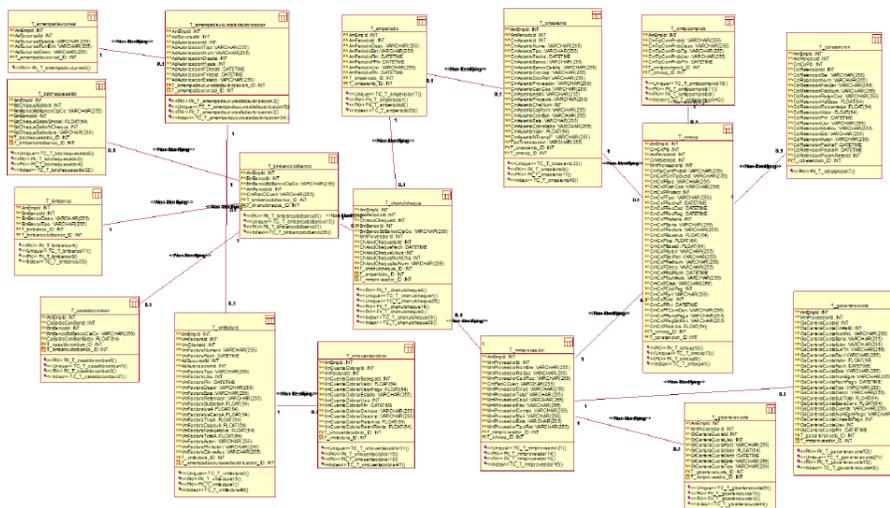


Figura 16 Diagrama Modelo Físico. Vista detallada del modelo físico del sistema y sus conexiones entre clases.

4.06. Diagrama de Componentes

Este diagrama representa los componentes de un software y muestra las dependencias entre estos. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de componentes prevalecen en el campo de la arquitectura de software pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

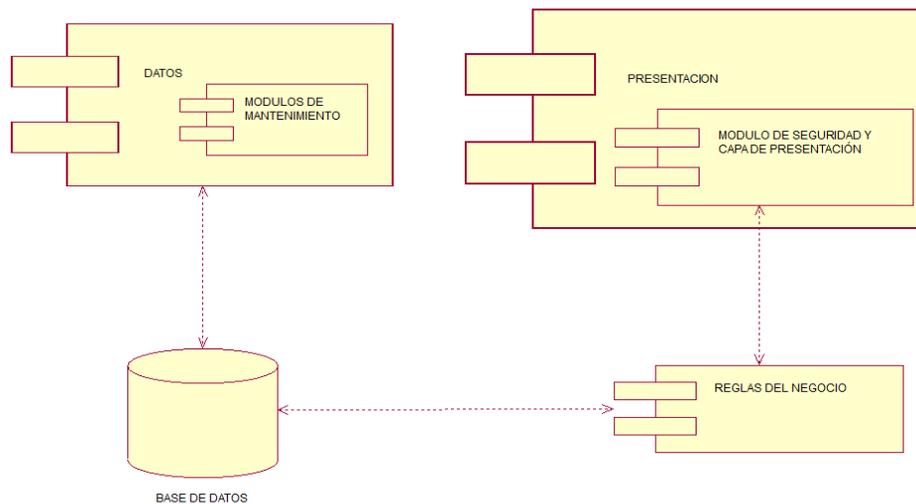


Figura 17 *Diagrama de Componentes 001*. Vista detallada del diagrama de componentes que se utilizan en el desarrollo y funcionamiento del sistema.

4.07. Diagramas de Estrategias

El diagrama de estrategias tiene como finalidad establecer la estructura y alcance de las mismas, vincular cada estrategia a todos los objetivos que pueden ser trabajados de forma articulada, determinar los límites del proyecto y determinar los objetivos específicos.

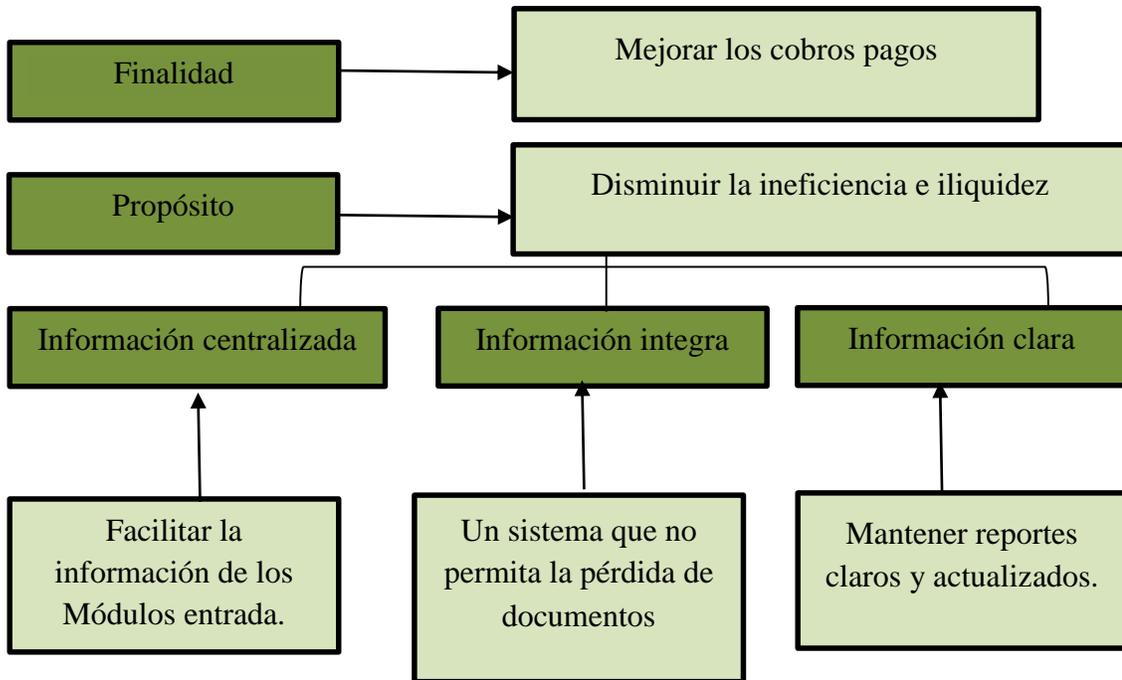


Figura 18 *Diagrama de Estrategias*. Vista detallada del diagrama de estrategias a utilizar en el sistema.

4.08. Matriz de Marco Lógico

La presente matriz de marco lógico nos permite resumir en un solo documento que es lo que se desea lograr en el proyecto, cómo se alcanzará el propósito y sus componentes, qué factores externos son indispensables para el éxito del proyecto, cómo se pretende medir el éxito de los objetivos, resultados y qué recursos son los necesarios para la ejecución de nuestro proyecto.

Tabla 23

Resumen del proyecto lo que se desea lograr.

| Resumen narrativo de objetos | Indicadores | Medios de verificación | Supuestos |
|------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| Fin: Mejorar la gestión de cobros y pagos | Reportes actualizados de cuentas por cobrar y pagar. | Fácil obtener los datos de proporcionan los módulos del sistema. | Capacitación al personal de la empresa con la nueva aplicación. |
| Propósito: Disminuir el tiempo en los procesos para la obtención de reportes a los inversionistas. | Prevenir pérdidas. | Optimizar recursos y mejorar los procesos. | Mayor bienestar de los usuarios |
| Componentes: 1. Sistema amigable para utilizar y atractiva a la vista del usuario. | Realizar ensayos de calidad a la aplicación con distintos usuarios para verificar la complejidad. | Realizar modelos que muestre las mejoras que se está logrando | . |
| Actividad: 1. Analizar cómo funciona el proceso. 2. Levantar requerimientos. 3. Realizar BDD y desarrollar la aplicación. | Tener claras las reglas del negocio para realizar un buen funcionamiento de la aplicación. | Documentar todo los cambios del sistema. | El tiempo es demasiado corto para desarrollar la aplicación en su totalidad por distintos motivos. |

Nota: Descripción del resumen del proyecto.

4.09. Vistas arquitectónicas

4.01.01. Vista lógica

Se aplican los principios de abstracción, encapsulación y herencia. Esta descomposición no sólo se hace para potenciar el análisis funcional, sino también sirve para identificar mecanismos y elementos de diseño.

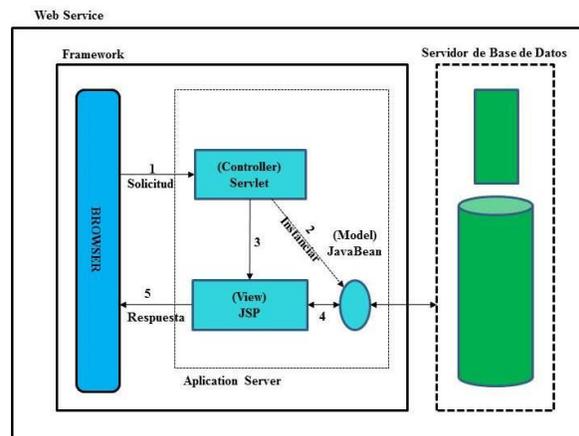


Figura 19 Descripción Lógica a del sistema.

FUENTE: HECTOR C

4.01.02. Vista física

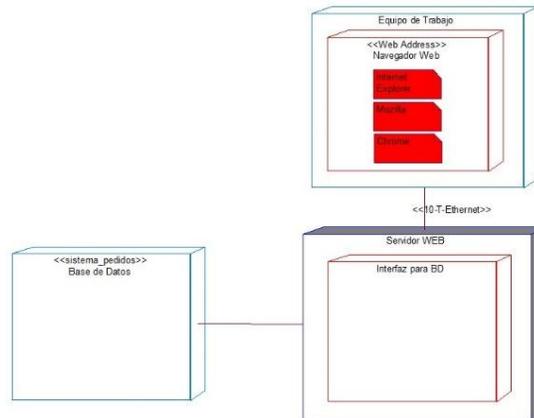


Figura 20 Vista física del diseño de la solución (Diagrama de Despliegue). Sacado del diagrama de despliegue para visualizar la forma física como está estructurado el sistema

FUENTE: HECTOR C

4.01.03. Vista de desarrollo

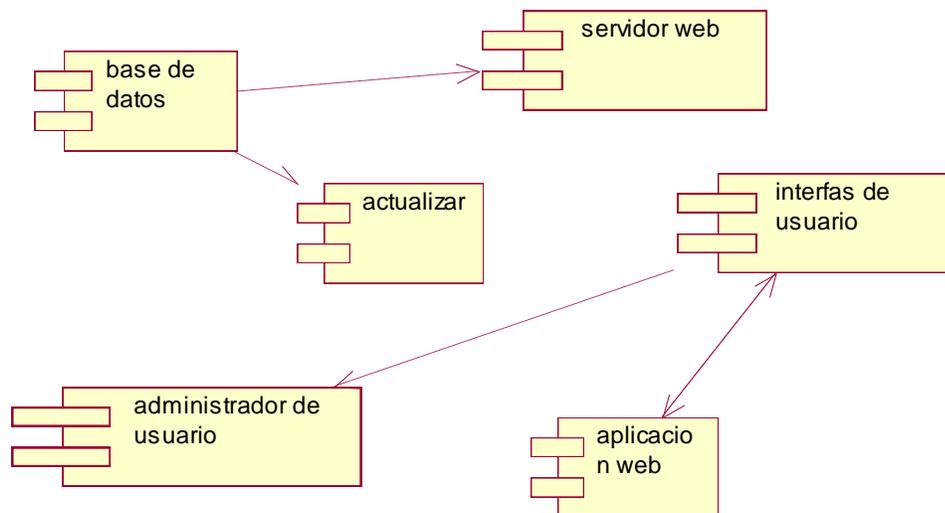


Figura 21 Vista de desarrollo de la solución (Diagrama de Componentes). Sacado del diagrama de componentes para poder determinar los componentes que intervienen en el funcionamiento del sistema.

4.01.04. Vista de procesos

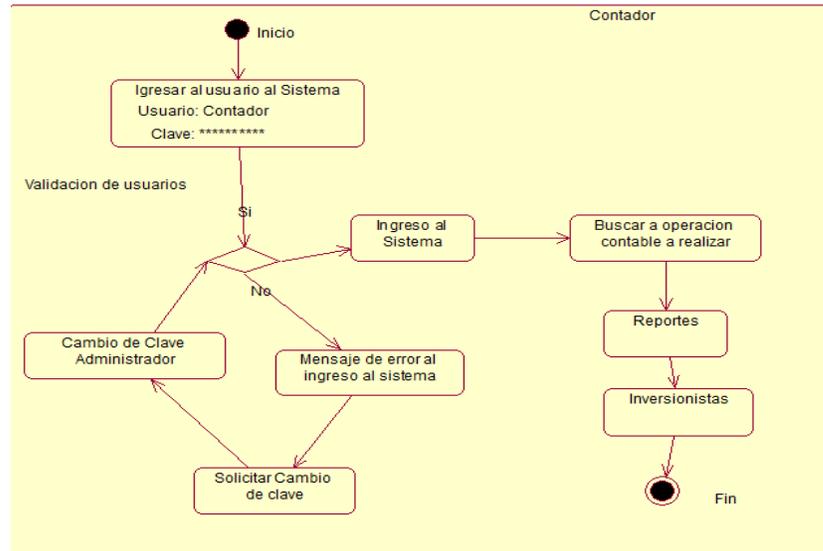


Figura 22 Descripción física del sistema

Capítulo V: Propuesta

5.01. Especificación de estándares de programación

5.01.01. Organización de ficheros

Las clases en Java se agrupan en paquetes. Estos paquetes se deben organizar de manera jerárquica, de forma que todo código desarrollado tendrá que estar incluido dentro de un paquete. Dentro del paquete principal las clases se organizarán en subpaquetes en función del área, organismo o sección del código desarrollado. Por ejemplo, si estamos desarrollando un servicio web de inscripción a un curso de programación Java las clases de dicho servicio se incluirían en un paquete. Un fichero consta de secciones que deben estar separadas por líneas en blanco y comentarios opcionales que identifiquen cada sección. Deben evitarse los ficheros de gran tamaño que contengan más de 1000 líneas. En ocasiones, este tamaño excesivo provoca que la clase no encapsule un comportamiento claramente definido, albergando una gran cantidad de métodos que realizan tareas funcional o conceptualmente heterogéneas. Ejemplos mas adelante.

5.01.02. Fichero fuente Java (.java)

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

En todo fichero fuente Java distinguimos las siguientes secciones:

-
- Comentarios de inicio.
 - Sentencia de paquete.
 - Sentencias de importación.
 - Declaraciones de clases e interfaces.

5.01.03. Comentarios de inicio

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

Dentro de estos comentarios iniciales podrían incluirse adicionalmente comentarios sobre los cambios efectuados sobre dicho fichero (mejora, incidencia, error, etc.). Estos comentarios son opcionales si los ficheros están bajo un sistema de control de versiones bien documentado, en caso contrario se recomienda su uso. Estos comentarios constituyen el historial de cambios del fichero. Este historial es único para cada fichero y permitirá conocer rápidamente el estado y la evolución que ha tenido el fichero desde su origen.

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

```
/*  
  
* @ (#) JceSecurity.java 1.50 04/04/14  
  
*  
* Copyright 2004 Sun Microsystems, Inc. All rights reserved.  
* SUN PROPRIETARY/CONFIDENTIAL. Use is subject to license terms.  
  
*/
```

/**

* This class instantiates implementations of JCE engine classes from

* providers registered with the java.security.Security object.

** @author Jan Luehe

* @author Sharon Liu

* @version 1.50, 04/14/04

* @since 1.4

*/

5.01.04. Sentencias de paquete

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

Por ejemplo,

```
package javax.crypto;
```

5.01.06. Sentencias de importación

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

- Paquetes del JDK de java.
- Paquetes de utilidades no pertenecientes al JDK de Java, de frameworks de desarrollo o de proyectos opensource tales como apache, hibernate, pringframework, etc.
- Paquetes desarrollados para el Ayuntamiento de Málaga.

-
- Paquetes de la aplicación.

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

```
import java.io.*; // BufferedReader, PrintWriter, FileInputStream, File  
  
import java.util.ArrayList;  
  
import org.apache.log4j.Logger;  
  
import org.apache.lucene.analysis.Analyzer;  
  
import es.provincia.organismo.corporativas.atlas.vo.AgendaVO;  
  
import es.provincia.organismo.atlas.vo.AnuncioVO;  
  
import es.provincia.organismo.atlas.vo.OrganigramaVO;
```

5.01.07. Declaraciones de clases e interfaces

La siguiente tabla describe los elementos que componen la declaración de una clase o interfaz, así como el orden en el que deben estar situados.

Tabla 24

Declaración de clases e interfaces

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

Nota: Descripción detallada de la declaración de clases e interfaces.

5.01.08. Sangría

Como norma general se establecen 4 caracteres como unidad de sangría. Los entornos de desarrollo integrado (IDE) más populares, tales como Eclipse o NetBeans, incluyen facilidades para formatear código Java.

5.01.09. Longitud de línea

La longitud de línea no debe superar los 80 caracteres por motivos de visualización e impresión.

5.01.10. División de líneas

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

- Tras una coma.
- Antes de un operador.
- Se recomienda las rupturas de nivel superior a las de nivel inferior.
- Alinear la nueva línea con el inicio de la expresión al mismo nivel que la línea anterior.
- Si las reglas anteriores generan código poco comprensible, entonces estableceremos tabulaciones de 8 espacios.

Ejemplos:

```
unMetodo(expresionLarga1, expresionLarga 2, expresionLarga 3,  
expresionLarga 4, expresionLarga 5);  
  
if ((condicion1 && condicion2)  
|| (condicion3 && condicion4)  
||!(condicion5 && condicion6)) {  
unMetodo();  
}
```

5.01.11. Comentarios

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

5.01.12. Comentarios de implementación

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

Distinguimos tres tipos de comentarios de implementación:

- Comentarios de bloque:

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

```
/*  
  
* Esto es un comentario  
  
* de bloque  
  
*/
```

- Comentarios de línea:

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

Deben estar precedidos por una línea en blanco.

```
/* Esto es un comentario de línea */  
  
//Esto es otro comentario de línea
```

- Comentario a final de línea

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

```
int contador = 4 + 10; //Inicialización del contador  
  
contador++; /* Incrementamos el contador */
```

5.01.13. Comentarios de documentación

Los comentarios de documentación, también denominados "comentarios javadoc", se utilizan para describir la especificación del código, desde un punto de vista independiente de la implementación, de forma que pueda ser consultada por desarrolladores que probablemente no tengan acceso al código fuente.

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

5.01.14. Una declaración por línea

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

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

5.01.15. Inicialización

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

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

5.01.16. Localización

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

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

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

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

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

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

5.01.17. Declaración de clases / interfaces

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

- No incluir ningún espacio entre el nombre del método y el paréntesis inicial del listado de parámetros.
- El carácter inicio de bloque ("{") debe aparecer al final de la línea que contiene la sentencia de declaración.
- El carácter fin de bloque ("} ") se sitúa en una nueva línea tabulada al mismo nivel que su correspondiente sentencia de inicio de bloque, excepto cuando la sentencia sea nula, en tal caso se situará detrás de "{".
- Los métodos se separarán entre sí mediante una línea en blanco.

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

5.01.18. Sentencias

Cada línea debe contener como máximo una sentencia. Ejemplo,

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

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

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

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

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

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

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

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

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

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

5.01.19. Espacios en blanco

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

Se utilizarán espacios en blanco en los siguientes casos:

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

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

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

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

Para separar un operador binario de sus operandos, excepto en el caso del operador (".").

Nunca se utilizarán espacios entre los operadores unarios (p.e., "++" o "--") y sus operandos. Por ejemplo:

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

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

```
contador++;
```

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

```
for (expresion1; expresion2; expresion3)
```

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

```
Unidad unidad = (Unidad) objeto;
```

5.01.20. Nomenclatura de identificadores

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

5.01.21. Paquetes

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

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

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

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

Ejemplos:

es.provincia.organismo1.festivaldecine

es.provincia.organismo2.vivienda

es.provincia.organismo3.juventud

es.provincia.organismo3.formacion

es.provincia.organismo3.gestionturistica

java.util.ArrayList

java.util.Date

java.util.Properties

javax.servlet.http.HttpServletRequest

javax.servlet.http.HttpServletResponse

5.01.22. Clases e interfaces

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

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

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

```
class Ciudadano
```

```
class OrganigramaDAO
```

```
class AgendaService
```

```
class IAgendaService
```

5.01.23. Métodos

Los métodos deben ser verbos escritos en minúsculas. Cuando el método esté compuesto por varias palabras cada una de ellas tendrá la primera letra en mayúsculas.

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

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

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

5.01.24. Variables

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

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

Unidad unidad;

Agenda agenda;

Tramite tramite;

5.01.25. Constantes

Todos los nombres de constantes tendrán que escribirse en mayúsculas. Cuando los nombres de constantes sean compuestos las palabras se separarán entre sí mediante el carácter de subrayado "_".

```
int LONGITUD_MAXIMA;
```

```
int LONGITUD_MINIMA;
```

5.01.02.01. Prácticas de programación

5.01.02.02. Visibilidad de atributos de instancia y de clase

Los atributos de instancia y de clase serán siempre privados, excepto cuando tengan que ser visibles en subclases herederas, en tales casos serán declarados como protegidos.

El acceso a los atributos de una clase se realizará por medio de los métodos "get" y "set" correspondientes, incluso cuando el acceso a dichos atributos se realice en los métodos miembros de la clase.

```
public class Unidad {  
  
    private int id;  
  
    private String nombre;  
  
    public void actualizaUnidad(Unidad unidad) {  
  
        this.setId(unidad.getId());  
  
        this.setNombre(unidad.getNombre());  
  
    }  
  
}
```

5.01.02.03. Referencias a miembros de una clase

Evitar el uso de objetos para acceder a los miembros de una clase (atributos y métodos estáticos). Utilizaremos en su lugar el nombre de la clase. Por ejemplo:

```
metodoUtilidad(); // Acceso desde la propia clase estática  
  
ClaseUtilidad.metodoUtilidad(); // Acceso común desde cualquier clase
```

5.01.02.04. Constantes

Los valores constantes (literales) nunca aparecerán directamente en el código. Para designar dichos valores se utilizarán constantes escritas en mayúsculas y se declararán, según su ámbito de uso, o bien en una Clase de constantes creada para tal efecto, o bien en la clase donde sean utilizadas.

```
// Uso incorrecto  
  
codigoErrorUsuarioNoEncontrado = 1;
```

```
switch (error) {  
  
case codigoErrorUsuarioNoEncontrado:  
  
...}  
  
// Uso correcto  
  
public final int CODIGOERROR_USUARIONOENCONTRADO = 1;  
  
...  
  
switch (error) {  
  
case CODIDOGERROR_USUARIONOENCONTRADO:  
  
...  
  
}
```

5.01.02.05. Asignación sobre variables

Se deben evitar las asignaciones de un mismo valor sobre múltiples variables en una misma sentencia, ya que dichas sentencias suelen ser difíciles de leer.

```
int a = b = c = 2; // Evitar
```

No utilizar el operador de asignación en aquellos lugares donde sea susceptible de confusión con el operador de igualdad. Por ejemplo:

```
// INCORRECTO
```

```
if ((c = d++) == 0) { }
```

```
// CORRECTO
```

```
c = d++;
```

```
if (c == 0) { }
```

No utilizar asignaciones embebidas o anidadas. Ejemplo:

```
c = (c = 3) + 4 + d; // Evitar
```

debería escribirse

$c = 3;$

$c = c + 4 + d;$

5.01.02.06. Otras prácticas

Paréntesis

Es una buena práctica el uso de paréntesis en expresiones que incluyan distintos tipos de operadores para evitar problemas de precedencia de operadores. Aunque la precedencia de operadores nos pueda parecer clara, debemos asumir que otros programadores no tengan un conocimiento exhaustivo sobre las reglas de precedencia.

```
if (w == x && y == z) // INCORRECTO
```

```
if ((w == x) && (y == z)) // CORRECTO
```

Valores de retorno

Los valores de retorno tendrán que ser simples y comprensibles, de acuerdo al propósito y comportamiento del objeto en el que se utilicen.

```
// INCORRECTO
```

```
public boolean esProgramador(Empleado emp) {  
    if (emp.getRol().equals(ROL_PROGRAMADOR)) {  
        return true;  
    } else {  
        return false;  
    }  
}
```

```
// CORRECTO
```

```
public boolean esProgramador(Empleado emp) {  
  
    boolean esUnProgramador = false;  
  
    if (emp.getRol().equals(ROL_PROGRAMADOR)) {  
  
        esUnProgramador = true;  
  
    }  
  
    return esUnProgramador;  
  
}
```

Expresiones en el operador condicional ternario

Toda expresión compuesta, por uno o más operadores binarios, situada en la parte condicional del operador ternario deberá ir entre paréntesis. Ejemplo:

```
(x >= y) ? x : y;
```

Comentarios especiales (TODO, FIXME, XXX)

Utilizaremos XXX para comentar aquella porción de código que, aunque no tenga mal funcionamiento, requiera modificaciones. Usaremos FIXME para señalar un bloque de código erróneo que no funciona. Emplearemos TODO para comentar posibles mejoras de código, como puedan ser las debidas a optimizaciones, actualizaciones o refactorizaciones.

5.01.02.07. Documentación: javadoc

Se aconseja, como buena práctica de programación, incluir en la entrega de la aplicación la documentación de los ficheros fuente de todas las clases. Dicha documentación será generada por la herramienta "javadoc".

La herramienta "javadoc" construirá la documentación a partir de los comentarios (incluidos en las clases) encerrados entre los caracteres "/*" y "*/". Distinguimos tres

tipos de comentarios javadoc, en función del elemento al que preceden: de clase, de variable y de método.

Dentro de los comentarios "javadoc" podremos incluir código html y etiquetas especiales de documentación. Estas etiquetas de documentación comienzan con el símbolo "@", se sitúan al inicio de línea del comentario y nos permiten incluir información específica de nuestra aplicación de una forma estándar.

Como norma general utilizaremos las siguientes etiquetas:

- @author Nombre
- Añade información sobre el autor o autores del código.
- @version InformacionVersion
- Permite incluir información sobre la versión y fecha del código.
- @param NombreParametro Descripción

Inserta el parámetro especificado y su descripción en la sección "Parameters:" de la documentación del método en el que se incluya. Estas etiquetas deben aparecer en el mismo orden en el que aparezcan los parámetros especificados del método. Este tag no puede utilizarse en comentarios de clase, interfaz o campo. Las descripciones deben ser breves.

- @return Descripción

Inserta la descripción indicada en la sección "Returns:" de la documentación del método. Este tag debe aparecer en los comentarios de documentación de todos los métodos, salvo en los constructores y en aquellos que no devuelvan ningún valor (void).

- @throws NombreClase Descripción

Añade el bloque de comentario "Throws:" incluyendo el nombre y la descripción de la excepción especificada. Todo comentario de documentación de un método debe contener un tag "@throws" por cada una de las excepciones que pueda elevar. La descripción de la excepción puede ser tan corta o larga como sea necesario y debe explicar el motivo o motivos que la originan.

- @see Referencia

Permite incluir en la documentación la sección de comentario "See also:", conteniendo la referencia indicada. Puede aparecer en cualquier tipo de comentario "javadoc". Nos permite hacer referencias a la documentación de otras clases o métodos.

- @deprecated Explicación

Esta etiqueta indica que la clase, interfaz, método o campo está obsoleto y que no debe utilizarse, y que dicho elemento posiblemente desaparecerá en futuras versiones.

"javadoc" añade el comentario "Deprecated" en la documentación e incluye el texto explicativo indicado tras la etiqueta. Dicho texto debería incluir una sugerencia o referencia sobre la clase o método sustituto del elemento "deprecado".

- @since Version

Se utiliza para especificar cuándo se ha añadido a la API la clase, interfaz, método o campo. Debería incluirse el número de versión u otro tipo de información.

El siguiente ejemplo muestra los tres tipos de comentarios "javadoc",

```
/**
```

```
* UnidadOrganizativa.java:
```

```
*
```

```
* Clase que muestra ejemplos de comentarios de documentación de código.
```

```
* @author jlflorido
* @version 1.0, 05/08/2008
* @see documento "Normas de programación v1.0"
* @since jdk 5.0
*/

public class UnidadOrganizativa extends PoolDAO {

    /** Trazas de la aplicación */

    private Logger log = Logger.getLogger(UnidadOrganizativa.class);

    /** Identificador de la unidad organizativa */

    private int id;

    /** Nombre de la unidad organizativa */

    private String nombre;

    /** Obtiene el identificador de esta unidad organizativa */

    public int getId() {

        return id;

    }

    /** Establece el identificador de esta unidad organizativa */

    public void setId(int id) {

        this.id = id;

    }

    /** Obtiene el nombre de esta unidad organizativa */

    public String getNombre() {

        return nombre;

    }

}
```

```
}

/** Establece el nombre de esta unidad organizativa */

public void setNombre(String nombre) {

    this.nombre = nombre;

}

/**

 * Inserta la unidad organizativa en el sistema.

 *

 * @param unidad Unidad organizativa a insertar

 * @throws Exception Excepción elevada durante el proceso de inserción

 */

public void insertarUnidad(UnidadOrganizativa unidad) throws Exception{

    log.debug("-> insertarUnidad(UnidadOrganizativa unidad)");

    Connection conn = null;

    PreparedStatement pstmt = null;

    StringBuffer sqlSb = null;

    try {

        conn = this.dameConexion();

        sqlSb = new StringBuffer("")

        .append("INSERT INTO ORG.UNIDAD_ORGANIZATIVA ")

        .append("(ID, NOMBRE) VALUES (?, ?)");

        pstmt = conn.prepareStatement(sqlSb.toString());

        pstmt.setInt(1, unidad.getId());

    }

}
```

```
pstmt.setString(2, unidad.getNombre());  
  
pstmt.executeUpdate();  
  
} catch (Exception e) {  
  
log.error("Error: error al insertar la unidad. " +  
"Descripción:" + e.getMessage(), e);  
  
throw e;  
  
} finally {  
  
log.debug("<- insertarUnidad(UnidadOrganizativa unidad)");  
  
}  
  
}  
  
}
```

(Foundations, 2014)

5.02. Diseño de Interfaces de Usuario

El diseño de la interfaz de usuario es el proceso de determinar los distintos componentes, tanto de hardware como de software, sus características y su disposición, que se utilizarán para interactuar con una serie de usuarios determinados en un medio ambiente determinado.

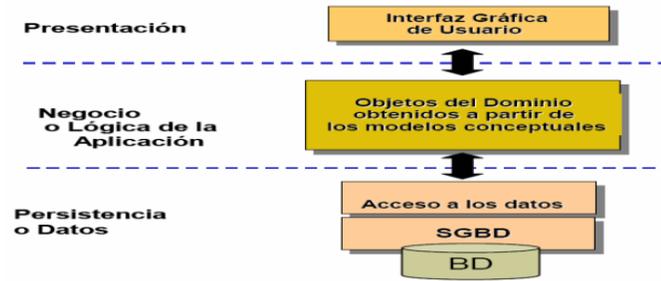


Figura 23 Capas con las que interactúa la interface del usuario

FUENTE: GARCIA O.

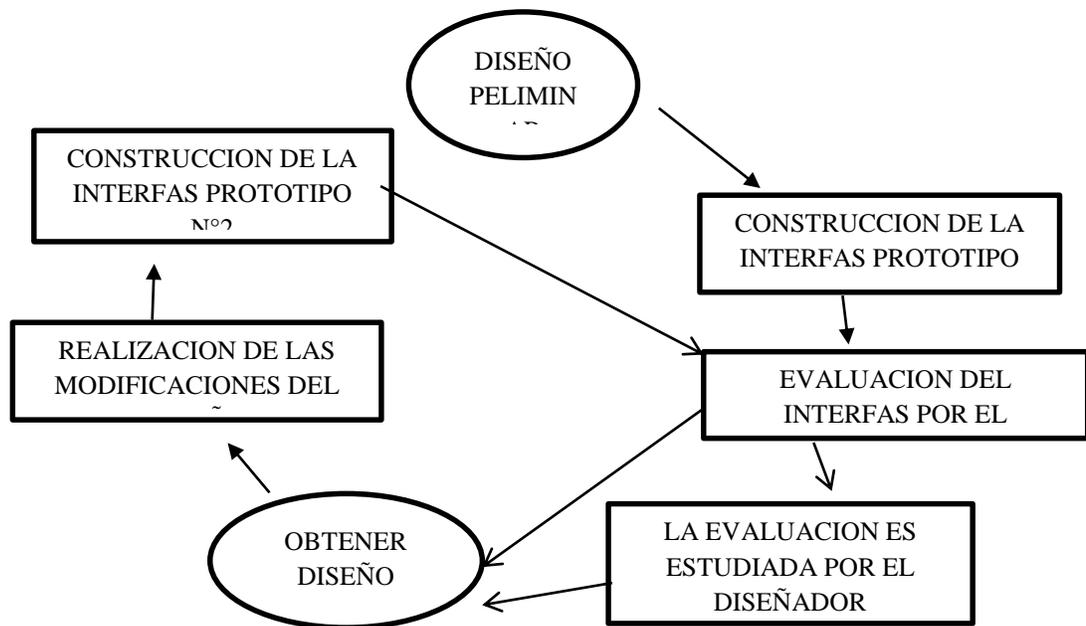


Figura 24 Diagrama general de los pasos para diseñar una interfaz.



Figura 25 Muestra la página principal de ERP.



Figura 26 *Ingreso al sistema*. Vista detallada de los elementos y botones que muestra la pantalla de usuario para el ingreso y registro en el sistema.

Tabla 25

Ventana principal.

| Ítem | Representación | Descripción | Nomenclatura de programación |
|------|----------------|----------------------|------------------------------|
| A | Menú | Selección de modulo | menmenuprin |
| B | Botón | Salir | btnsalir |
| C | Etiqueta | Proyecto | lblproyecto |
| D | Lista | Periodo | ddlpedido |
| E | Lista | Nombre de la Empresa | ddlnomempresa |
| F | Imagen | Logo de la Empresa | Imglogo |
| G | Etiqueta | Menú | Lblmenu |

Nota: Descripción detallada de elementos y botones que muestra la pantalla principal del sistema ERP.

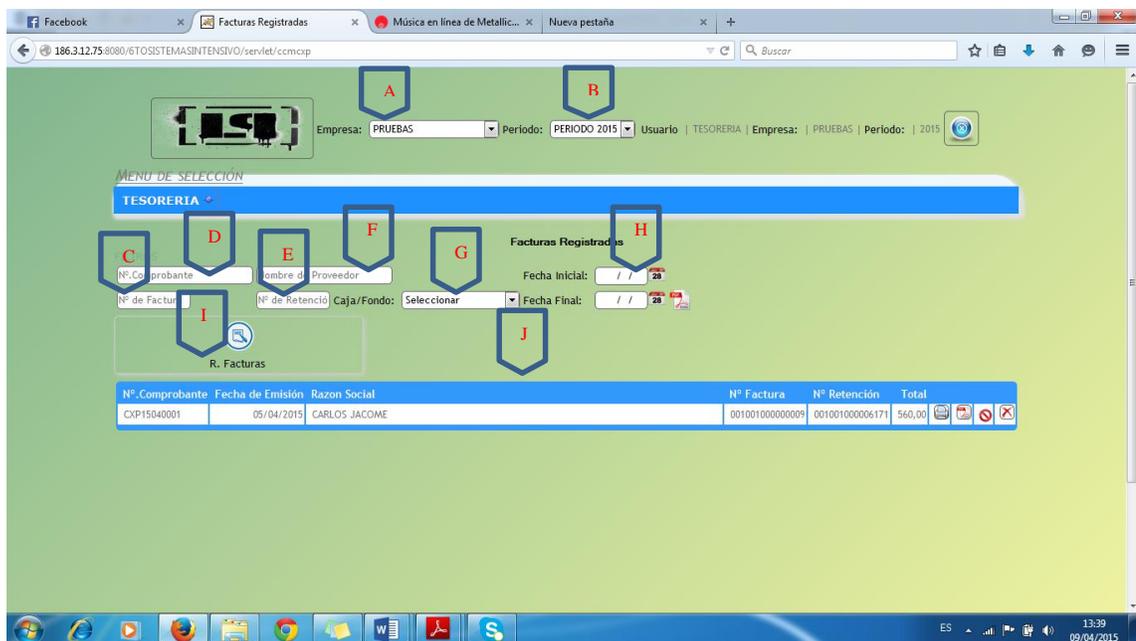


Figura 27 Muestra el ingreso de facturas al sistema

Tabla 26

Registro de facturas

| Ítem | Representación | Descripción | Nomenclatura de programación |
|----------|----------------|------------------------------|------------------------------|
| A | Lista | Selección de empresa | listnomempresa |
| B | Lista | Selección de periodo | listperiodo |
| C | Texto | Número de factura | txtnumfactura |
| D | Texto | Número de comprobante | txtnumcomprobante |
| E | Texto | Número de retención | txtnumretencion |
| F | Texto | Nombre del proveedor | txtnomproveedor |
| G | Lista | Tipo de pago | listcajfondo |
| H | Texto | Fecha inicial / Fecha final | txtfecha |
| I | Botón | Guardar factura | Btnguafactura |
| J | Tabla | Muestra la factura ingresada | Tblmuefactura |

Nota: Descripción detallada de elementos y botones que se usan en el registro de facturas.

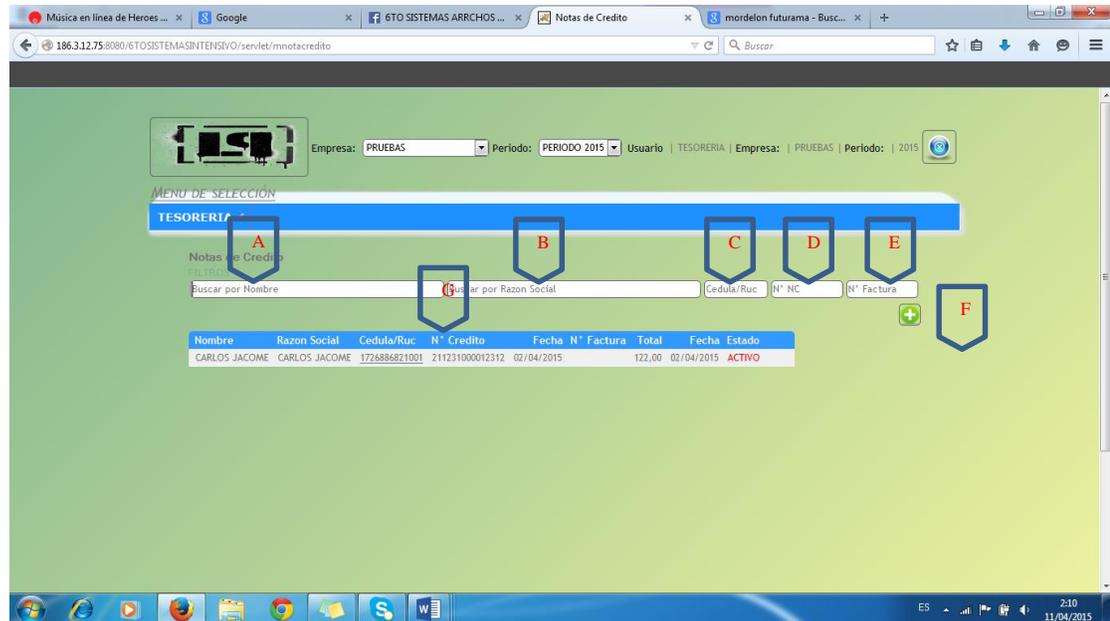


Figura 28 Muestra el ingreso de las notas de crédito

Tabla 27

Notas de crédito

| Ítem | Representación | Descripción | Nomenclatura de programación |
|------|----------------|------------------------------|------------------------------|
| A | Texto | Busca por nombre | txtnombre |
| B | Texto | Busca por razón social | txtrazsocial |
| C | Texto | Número de cédula | txtciruc |
| D | Texto | Número de comprobante | txtnumcomprobante |
| E | Texto | Número de factura | txtnumfactura |
| F | Botón | Crea nueva nota de crédito | btnnotcredito |
| G | Tabla | Muestra las notas de crédito | tblnotcredito |

Nota: Descripción detallada de elementos y botones que se usan en la nota de crédito.

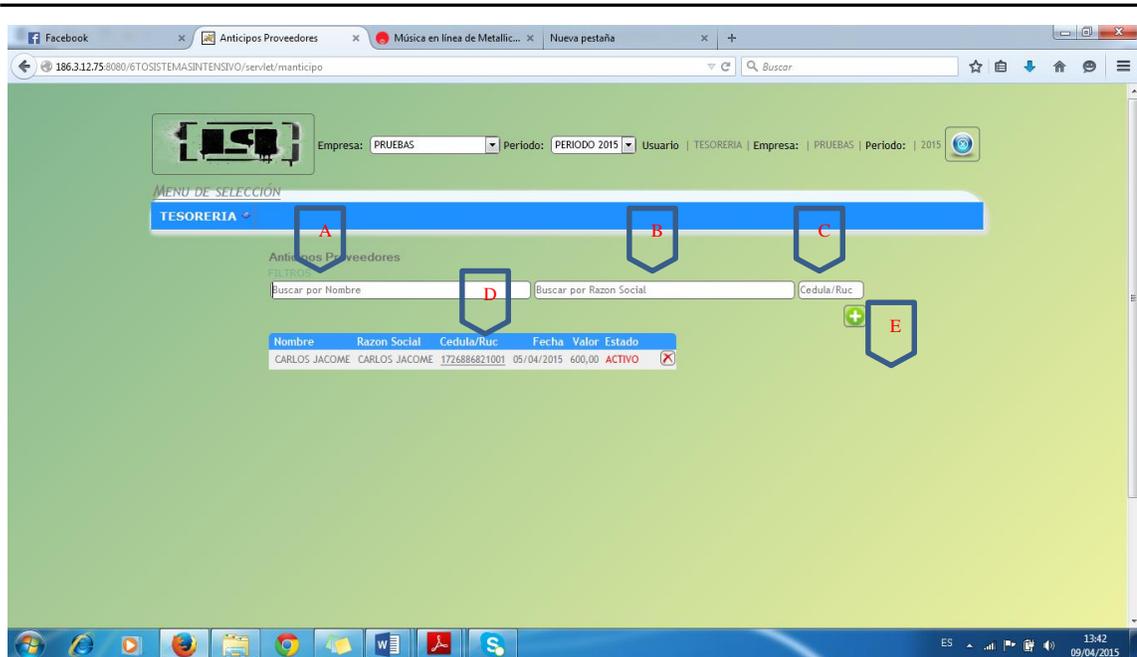


Figura 29 Muestra el anticipo a proveedores

Tabla 28

Anticipos proveedores

| Ítem | Representación | Descripción | Nomenclatura de programación |
|------|----------------|------------------------------|------------------------------|
| A | Texto | Busca por nombre | txtnombre |
| B | Texto | Busca por razón social | txtrazsocial |
| C | Texto | Número de cédula | txtnumcedula |
| D | Tabla | Muestra anticipos realizados | tblanticipos |
| E | Botón | Nuevo anticipo | btnanticipo |

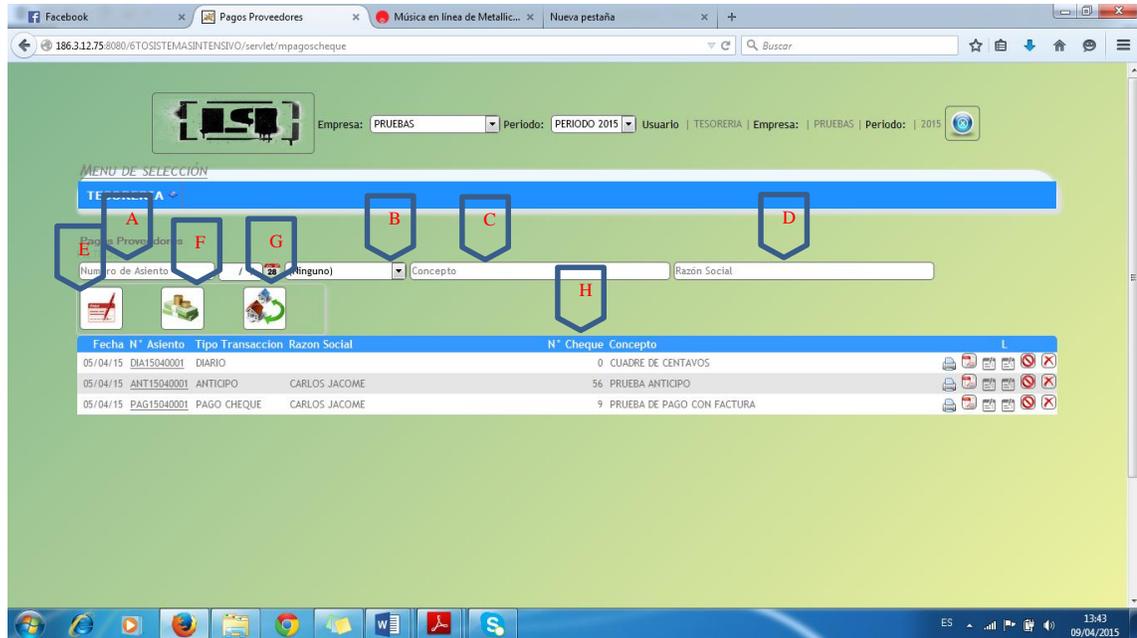


Figura 30 Muestra el pago a proveedores

Tabla 29

Pago a proveedores

| Ítem | Representación | Descripción | Nomenclatura de programación |
|------|----------------|------------------------------|------------------------------|
| A | Texto | Número de asiento | txtnumasiento |
| B | Lista | Modo de pago | listmodpago |
| C | Texto | Concepto | txtconcepto |
| D | Texto | Razón social | txtrazsocial |
| E | Botón | Nuevo pago con cheque | btnpagcheque |
| F | Botón | Nuevo pago con caja | btnpagcaja |
| G | Botón | Nuevo pago con transferencia | btnpagtransferencia |
| H | Tabla | Muestra pagos realizados | Tblpagos |

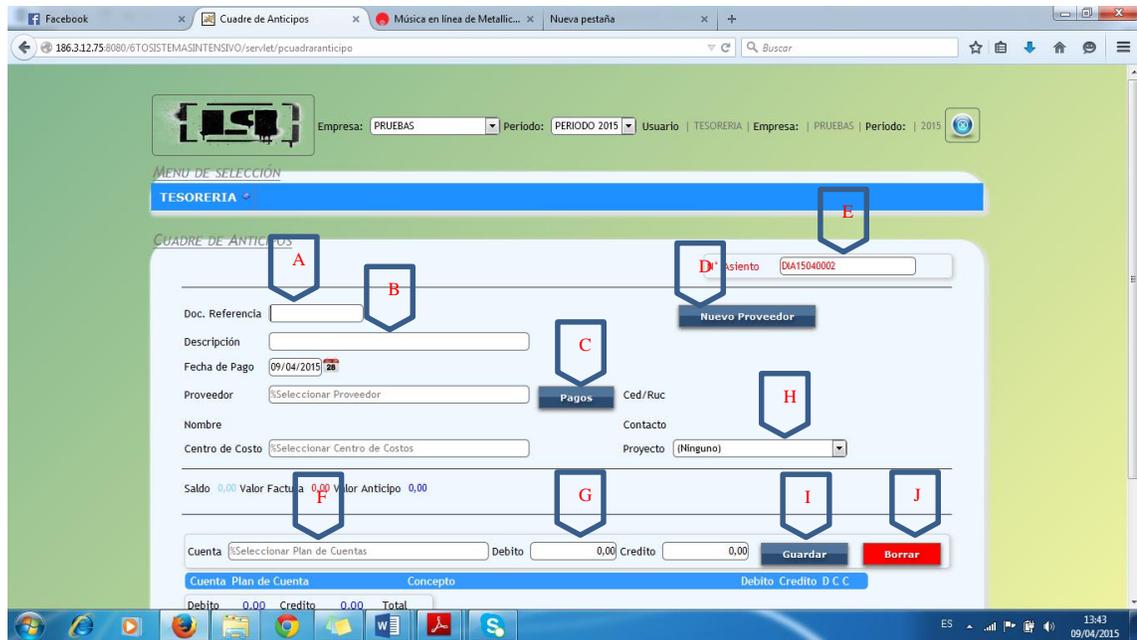


Figura 31 Muestra el cuadro de anticipos

Tabla 30

Cuadre de anticipos

| Ítem | Representación | Descripción | Nomenclatura de programación |
|------|----------------|-------------------------|------------------------------|
| A | Texto | Documento de referencia | txtdocreferencia |
| B | Texto | Descripción | txtdescripcion |
| C | Botón | Pago | btnpago |
| D | Botón | Nuevo proveedor | btnnueproveedor |
| E | Texto | Número de asiento | txtnumasiento |
| F | Texto | Cuenta | txtcuenta |
| G | Texto | Debito | txtdebito |
| H | Lista | Proyecto | listpro |
| I | Botón | Guardar | btnguardar |
| J | Botón | Borrar | btnborrar |

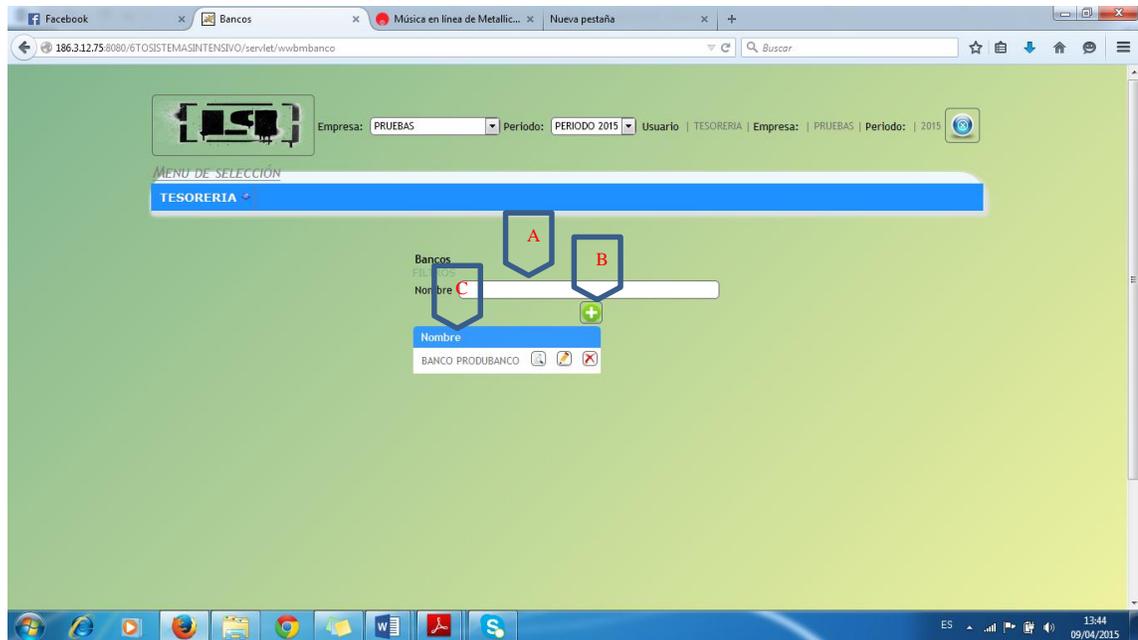


Figura 32 Muestra el ingreso a bancos

Tabla 31

Bancos

| Ítem | Representación | Descripción | Nomenclatura de programación |
|------|----------------|-------------------------------|------------------------------|
| A | Texto | Nombre | txtnombre |
| B | Botón | Nueva cuenta bancaria | btnnuebanco |
| C | Tabla | Muestra las cuentas bancarias | tbltabancarias |

5.03. Especificación de pruebas de unidad

Para realizar esta prueba es necesario ejecutarla sobre una sola tabla la misma que es la que tiene más transacción dentro del proyecto, la tabla "Tbl_usu" con un soporte mínimo de 50 clientes. Se medirá la persistencia de la tabla y de la programación para así obtener la persistencia de la transacción del sistema

Tabla 32

Matriz de pruebas de carga

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Identificador de la Prueba: | PC001 |
| Tipo de prueba | Simulación de desempeño con un número alto de registros en la base de datos/ Simulación de desempeño con un número alto de peticiones concurrentes |
| Objetivo de la Prueba: | Determinar que la interfaz de usuario cumple con lo requerido por el usuario e ingresar tantos datos como sea posible para medir la persistencia de la tabla. |
| Descripción | |
| Como usuario administrador crear una cierta cantidad de usuarios de reservación, proveedores utilizando la página web, ingresando un mínimo de 50 registros | |
| Resultados esperado | |
| Los registros ingresados en tabla se mantienen sin intermitencia y trabajan sin problemas | |
| Comentarios | |
| La ventana es aceptable y amigable con el usuario. | |

Tabla 33

Pruebas de Reportes, resultados eficientes

| | |
|---------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| Identificador de la Prueba: | PRU_UNI002 |
| Método a Probar | Reportes |
| Objetivo de la Prueba | Verificar que todos los resultados esperados en el proceso sean los adecuados y correctos. |
| Datos de Entrada: | |
| Datos Generales en los diferentes Formularios | |
| Resultados Esperados | |
| Corregir errores posibles al generar reportes | |
| Comentarios | |
| Se realiza pruebas en todos los reportes encontrando errores únicamente de ortografía | |

Tabla 34

Pruebas de compilación de Código.

| | |
|--------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| Identificador de la Prueba: | PRU_UNI003 |
| Método a Probar | Pruebas de Código - Compilación |
| Objetivo de la Prueba | Evaluar los resultados obtenidos y analizar los errores del código encontrados |
| Datos de Entrada: | |
| Compilación proceso paso a paso. | |
| Resultados Esperados | |
| Mantener el sistema en ejecución, corregir y evitar errores al compilar. | |
| Comentarios | |

Tabla 35

Pruebas de Almacenamiento de datos en la Base.

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| Identificador de la Prueba: | PRU_UNI004 |
| Método a Probar | Almacenamiento de datos en la base |
| Objetivo de la Prueba | Evaluar si los datos ingresados en los mantenimientos están siendo manejados de la manera adecuada. |
| Datos de Entrada: | |
| Datos generales en todos los formularios de Mantenimiento | |
| Resultados Esperados | |
| Obtener datos consistentes y coherentes. | |
| Comentarios | |
| Con esta prueba inclusive se verifica encriptación de contraseñas y que los valores sean reales al presentar en los diferentes reportes | |

5.04 Especificación de pruebas de aceptación

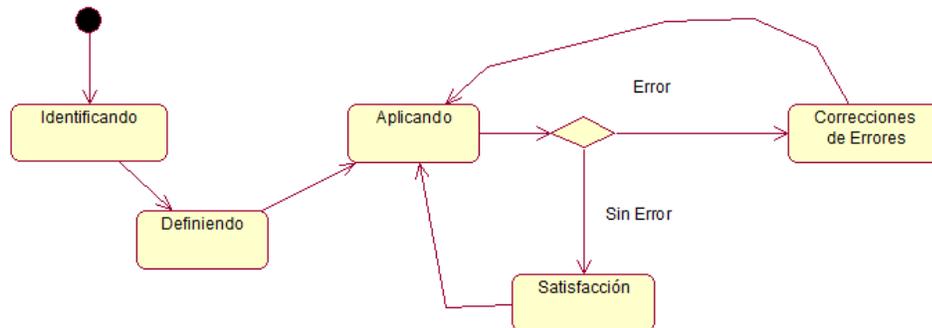


Figura 33 Diagrama de secuencia del proceso de pruebas de aceptación

Tabla 36

Detalle de pruebas de aceptación en la creación de usuarios

| | |
|------------------------------------|----------------------------------------------------------------------------------------------------|
| Identificador de la Prueba: | PRU_ACE001 |
| Caso de Uso | CU001 |
| Tipo de Usuario | Administrador |
| Objetivo de la Prueba | Probar el funcionamiento del proceso general de Usuarios. |
| Secuencia de Eventos | Login de usuario, ingresar Usuarios, Ejecutar cambios Eliminar. |
| Resultados Esperados | Que no tenga inconsistencias con respecto a guardados, validaciones y seguridad en la información. |
| Comentarios | Se realiza las pruebas en el sistema ingresando un nuevo usuario y dando mantenimiento a la tabla. |
| Estado Aceptado/No aceptado | Aceptado |

5.05. Especificación de pruebas de carga

Estas pruebas de producto se pueden realizar tanto en las plataformas de prueba del desarrollo como, opcionalmente, en la plataforma de producción del cliente. En

cualquier caso, el resultado obtenido consiste en una serie de informes que reflejan el rendimiento del sistema en distintos escenarios.

Ha de tenerse en cuenta que en estas pruebas se presentan factores que pueden influir en los resultados obtenidos como la topología de red, la configuración de los servidores...

Estas pruebas no pretenden optimizar todos estos factores sino sólo medir el rendimiento de las aplicaciones entregadas en su ubicación establecida.

Prueba normal. Permite constituir los tiempos medios de respuesta cuando sólo un usuario está conectado a la aplicación.

Esta prueba pretende establecer una referencia futura para posteriores comparaciones así como medir unitariamente el software entregado.

Prueba con número mínimo de usuarios. Se realizan las pruebas del sistema con el número de usuarios mínimos concurrentes establecido.

Prueba con número máximo de usuarios. Se realizan las pruebas del sistema con el número de usuarios máximo concurrentes establecido.

Prueba de número máximo soportado de usuarios. Se busca encontrar cuál es el límite del sistema ya es un diseñado para que sea confiable y eficiente.

Tabla 37

Detalle de un tipo de prueba de carga más baja.

| | |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Identificador de la Prueba: | PRCA001 |
| Tipo de Prueba | Prueba normal (Prueba de Carga) |
| Objetivo de la Prueba | Establecer los tiempos medios de respuesta cuando sólo un usuario está conectado en la aplicación. |
| Descripción: | |
| | Esta prueba pretende establecer una referencia futura para establecer la eficiencia cuando varios usuarios estén en uso del sistema y para posteriores comparaciones del sistema. |
| Resultados Esperados | |
| | Hacer que los procesos del sistema sean óptimos y tengan buenos tiempos de respuesta. |
| Comentarios | |
| | Ninguna |

Tabla 38

Detalle de un tipo de prueba de carga con un número mínimo de usuarios

| | |
|------------------------------------|--------------------------------------------------------------------------------------------------|
| Identificador de la Prueba: | PRCA002 |
| Tipo de Prueba | Prueba con número mínimo de usuarios |
| Objetivo de la Prueba | Conocer si los procesos se están efectuando con normalidad y sin problemas con algunos usuarios. |
| Descripción: | |
| | Se realizan las pruebas del sistema con el número de usuarios mínimos. |
| Resultados Esperados | |
| | Validar la funcionalidad del sistema con un mínimo de usuarios. |
| Comentarios | |

Tabla 39

Detalle de un tipo de prueba de carga con un número máximo de usuarios

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| Identificador de la Prueba: | PRCA003 |
| Tipo de Prueba | Prueba con número máximo de usuarios |
| Objetivo de la Prueba | Establecer los tiempos de respuesta cuando una gran cantidad de usuarios están conectados a la aplicación. |
| Descripción: | |
| Se realizan las pruebas del sistema con el número de usuarios máximo y verificar el estado de resultados en el menor tiempo. | |
| Resultados Esperados | |
| Hacer que los procesos del sistema sean óptimos y tengan buenos tiempos de respuesta cuando tengamos una gran cantidad de usuarios conectados. | |
| Comentarios | |
| Ninguna | |

Tabla 40

Se requiere descubrir los límites

| | |
|---------------------------------------------------------------------------------------|------------------------------------------------------|
| Identificador de la Prueba: | PRCA004 |
| Tipo de Prueba | Prueba de número máximo soportado de usuarios |
| Objetivo de la Prueba | Se busca encontrar cuál es el límite del sistema. |
| Descripción: | |
| Se realizan las pruebas del sistema con el número de usuarios para encontrar errores. | |
| Resultados Esperados | |
| Encontrar los límites que tiene el sistema en sobrecarga. | |
| Comentarios | |
| Ninguna | |

5.06. Configuración del Ambiente mínima/ideal

Creo que está bastante claro que las computadoras consumen energía y la energía generada por fuentes contaminantes puede ser un efecto directo de las computadoras. No

obstante, un estudio del impacto ambiental de estos equipos debe ir más allá. Los materiales usados para su construcción, metales y plásticos, minerales, etc. fueron extraídos del medio ambiente y esta actividad con seguridad afecta el medio ambiente. Luego, el transporte de los componentes hacia las fábricas donde se ensamblan los equipos generan contaminación por la quema de combustibles para los vehículos. Luego, una vez ensamblado el equipo se transporta a los sitios donde serán utilizados. Muchos países importan esos equipos y los traen de muy lejos, y aparece nuevamente la contaminación por el transporte.

Para casos las Aplicaciones desarrollados en un ambiente cliente servidor, un cuarto de máquinas envuelve los siguientes factores a tomar en consideración: localización, diseño, hardware, software, fuente de energía, temperatura, humedad, recuperación de desastres, seguridad. El planeamiento adecuado seguro de la localización y el diseño particular son los primeros pasos para crear un ambiente seguro.

Otras consideraciones a tener en cuenta son los insumos básicos con lo que la sala deservidores debe contar (enchufes, espacio, disponibilidad de red).

Un punto importante a considerar en cuanto al ambiente de la sala de servidores, es la humedad. Altos niveles de humedad pueden causar condensación y bajos niveles pueden causar electrostática. Adicionalmente, la sala de servidores debe contar con detectores de humo y agua, así como también UPS para proteger a los equipos de cortes/picos de electricidad.

El ambiente mínimo ideal para que nuestra aplicación en la web corra efectivamente sería:

-
- Contratar un Web Hosting confiable y que permanezca siempre en línea.
 - Contratar en la oficina un servicio de Internet que sea siempre estable.
 - Contratar un ancho de banda adecuada para tener una navegación óptima.

Mantener la infraestructura de la oficina en buenas condiciones.

Capítulo VI: Aspectos Administrativos

6.01. Recursos

- Empresas Ecuatorianas.
- Personal Administrativo de la Empresa
- Inversionistas
- Tutor del Proyecto
- Un Ordenador de buenas características.
- Manuales.
- MySql.
- Rational Rose
- Java NetBeans.
- Power Designer

6.02. Presupuesto

Tabla 41

Detalle de Gastos realizados en el Proyecto (Presupuesto)

| Descripción de Gastos | Cantidad | P. Unitario | Sub Total |
|---------------------------|-----------|-------------|-----------|
| Tinta de Impresión | 3 | 5\$ | 15\$ |
| Hojas | 3 Resmas | 3.30\$ | 9,90\$ |
| Lapiceros | 2 | 0,55\$ | 1.10\$ |
| Movilidad | 44 Pasaje | 0,25\$ | 11\$ |
| Teléfono llamadas | 3 Horas | 0,25\$ | 45\$ |
| Impresiones | 800 | 0,05\$ | 40\$ |

| | | | |
|---------------------|-----|--------|-------|
| Fotocopias | 500 | 0,03\$ | 15\$ |
| Alimentación | 90 | 3,50\$ | 315\$ |
| Total | | | 452\$ |

Capítulo VII: Conclusiones y Recomendaciones

7.01. Conclusiones

El Proyecto de Tesis planteado en este documento nos ha brindado la oportunidad de desarrollar una Aplicación Web de Servicio empresarial, así como efectuar los conocimientos en el análisis y desarrollo de Software logrados a lo largo de nuestra carrera, de estudio progresando nuestra investigación y análisis.

Se ha cumplido satisfactoriamente el desarrollo de este proyecto, que permitirá las empresas Ecuatorianas llevar un mejor control de sus estados financieros

La principal ventaja al desarrollar esta aplicación orientada al web para la tarea financiera, es que no tiene que tener recursos tecnológicos exageradamente costosos. Se diseñó una interface amigable para los usuarios involucrados.

7.02. Recomendaciones

- Antes del manejo de la aplicación debe leer la documentación.
- Para un correcto funcionamiento del sistema deben ser capacitados cada uno de los involucrados en la empresa.
- La persona encargada en administrar del sistema debe tener conocimientos en sistemas.
- Realizar verificaciones de funcionamiento del sistema.
- Capacitar al área de sistemas para mejor los rendimientos del sistema y la empresa.

ANEXOS

ANEXO A

Manual de instalación

Instalación de WampServer.

1.- Verificar si el sistema Operativo es de 64 o 32 Bits.

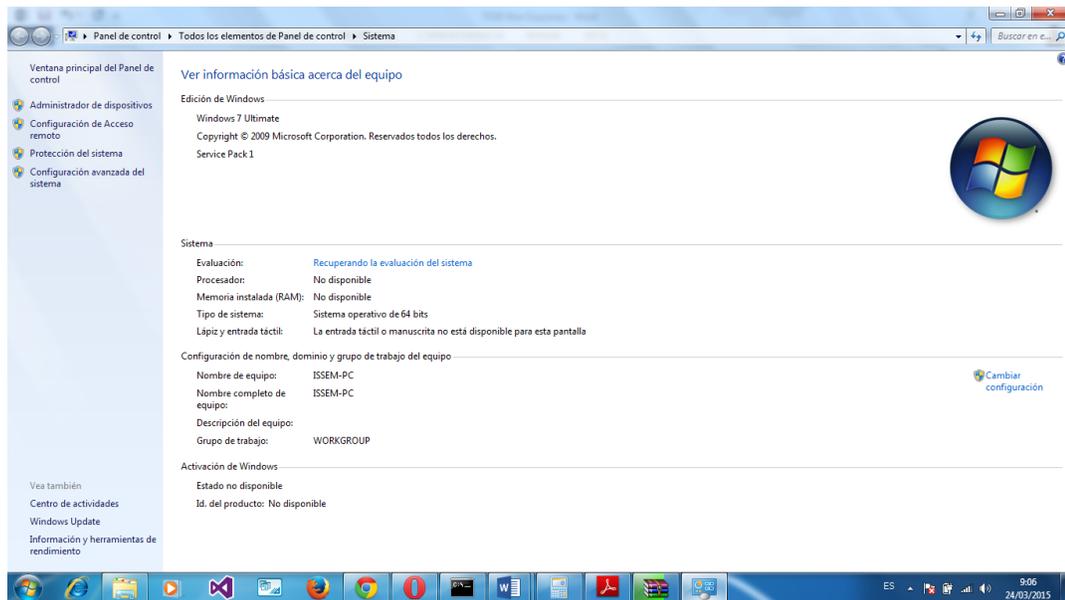


Figura 34 Se verifica la versión de sistema operativo

2.- Dar clic derecho en el instalador WampServer Ejecutar como administrador

(Dependiendo si su sistema es de 32 o 64 bis)

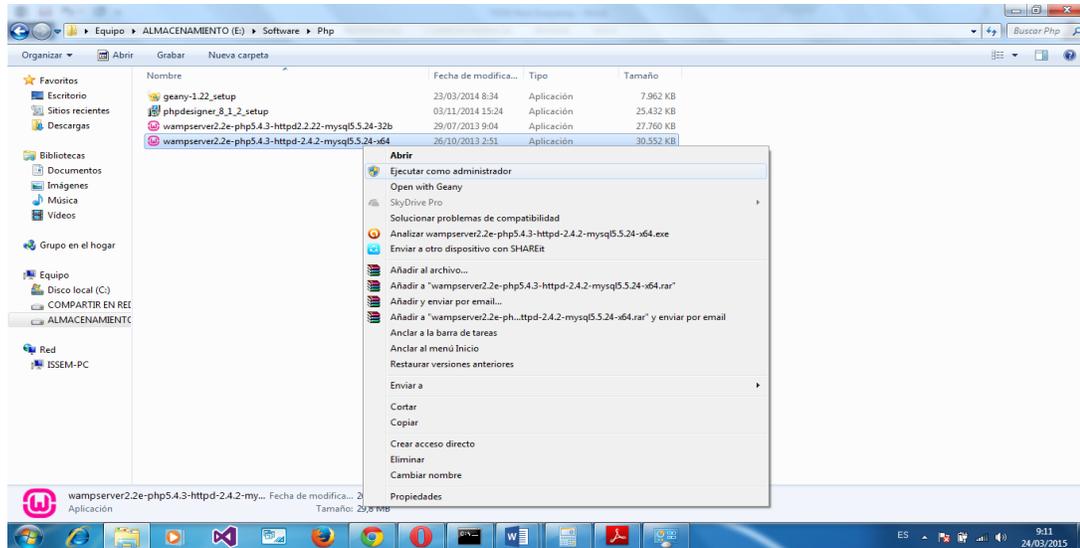


Figura 35 Inicio de instalación de WampServer

3.- Dar clic en Ejecutar.

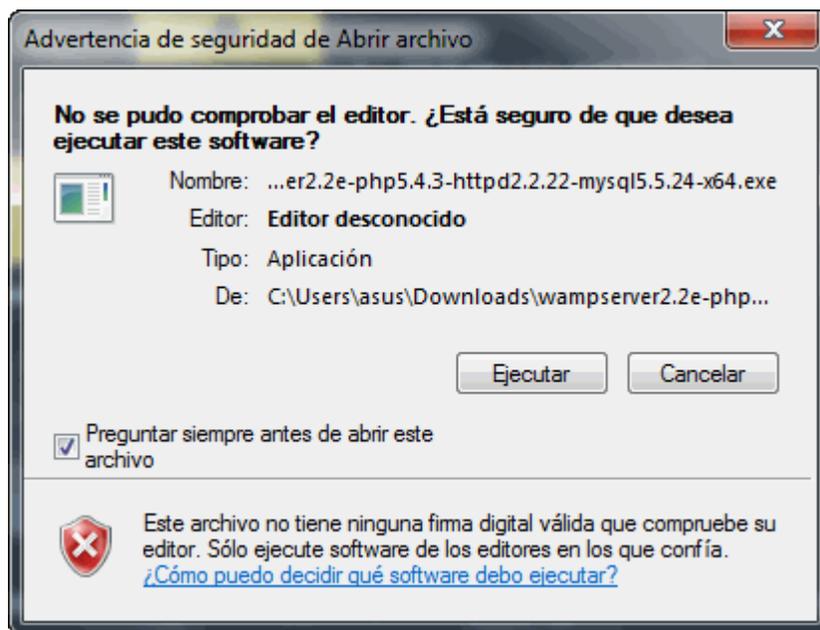


Figura 36 Ejecutar como administrador

4.- Luego desplegara una ventana de bienvenida, dar clic en siguiente (Next).

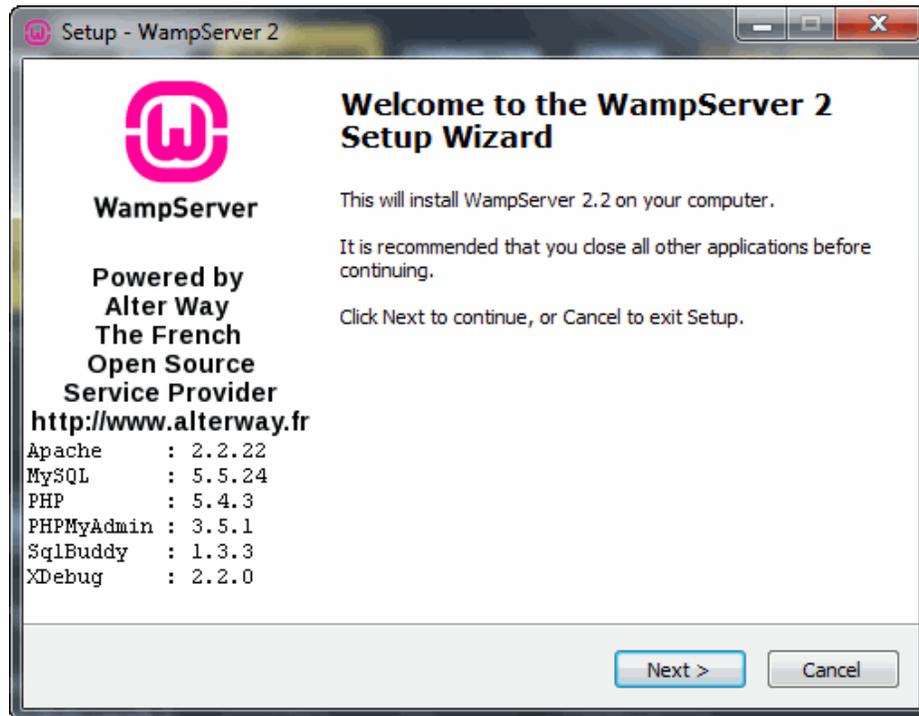


Figura 37 Paso 1 de instalación

5.- Luego aceptar los términos de Licencia y siguiente (Next).

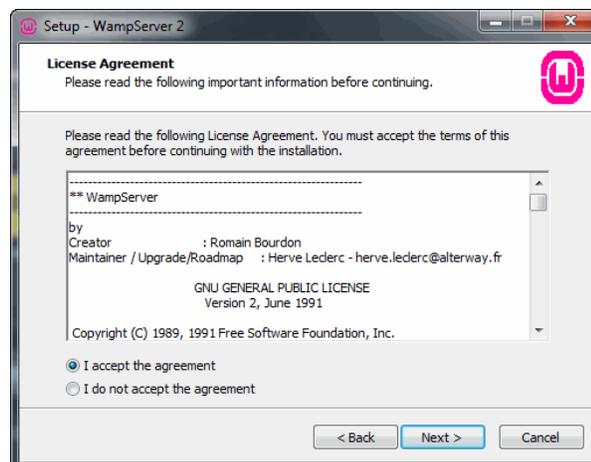


Figura 38 Acepta los acuerdos de la licencia

6.- Luego ubicar la instalación del WampServer (opcional).

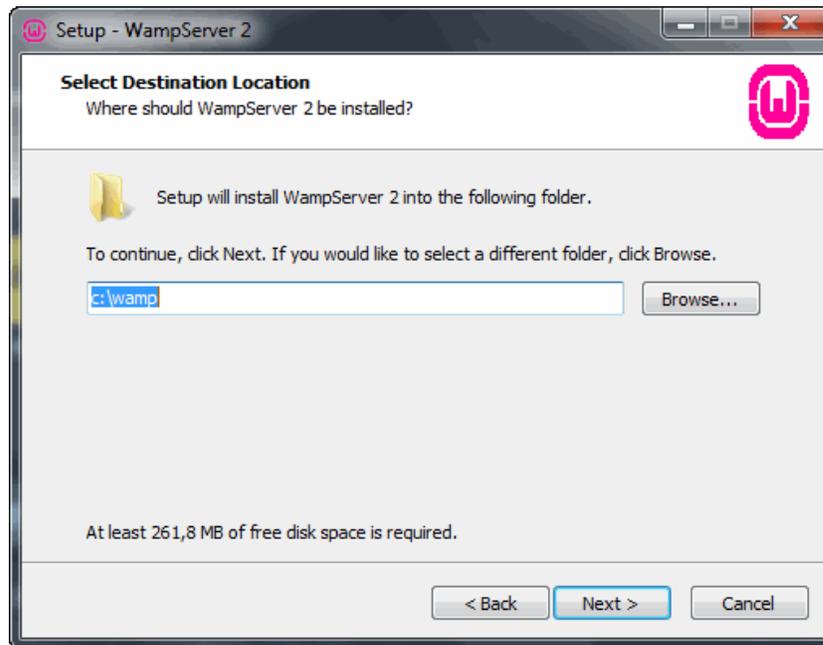


Figura 39 Selección de carpeta de instalación

7.- Si se desea un icono en el escritorio seleccionarlo y dar en siguiente (Next).

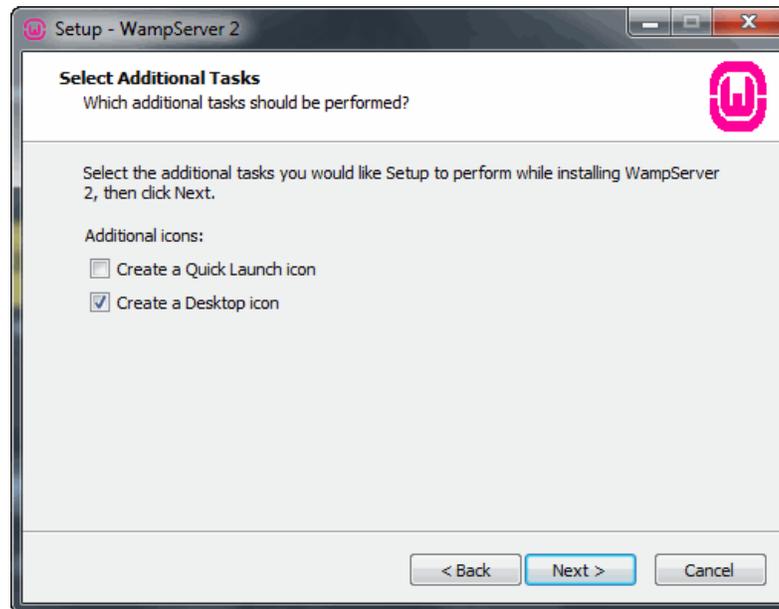


Figura 40 Opción de icono en el escritorio

8.- Dar clic en Instalar (Install)

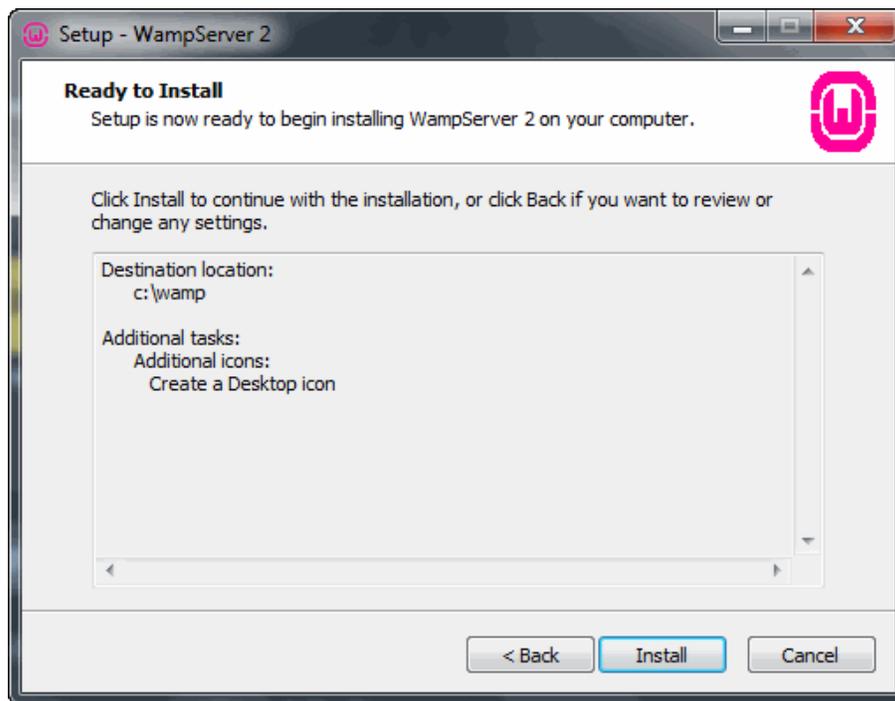


Figura 41 Inicio de instalación

9.- Esperar que se instale correctamente la espera es mínima.

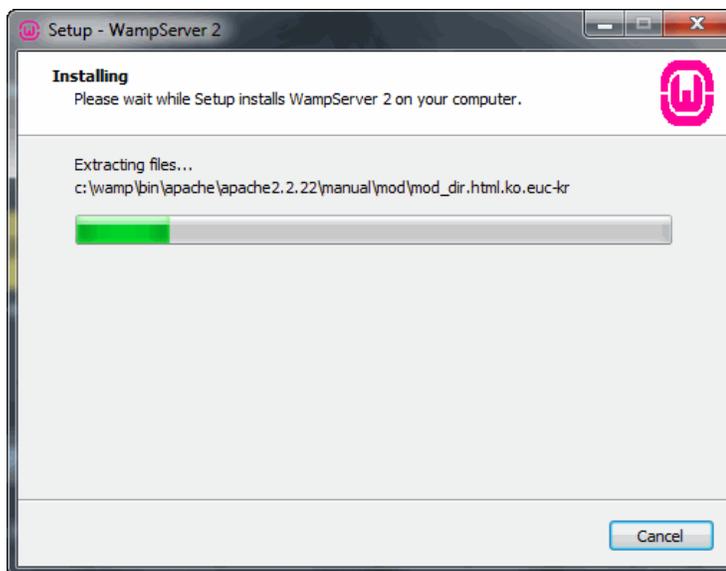


Figura 42 Muestra el progreso de la instalación

10.- Luego desplegara una ventana donde se puede elegir el navegador que será utilizado para la ejecución de la aplicación.

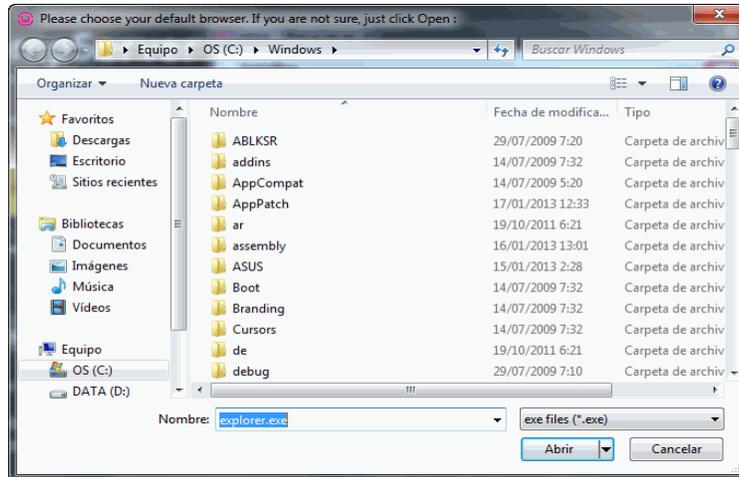


Figura 43 *Selecciona el navegador a ser usado*

11.- Esta ventana se configura el envío de email a PHP luego de ingresar dar clic en siguiente (Next).

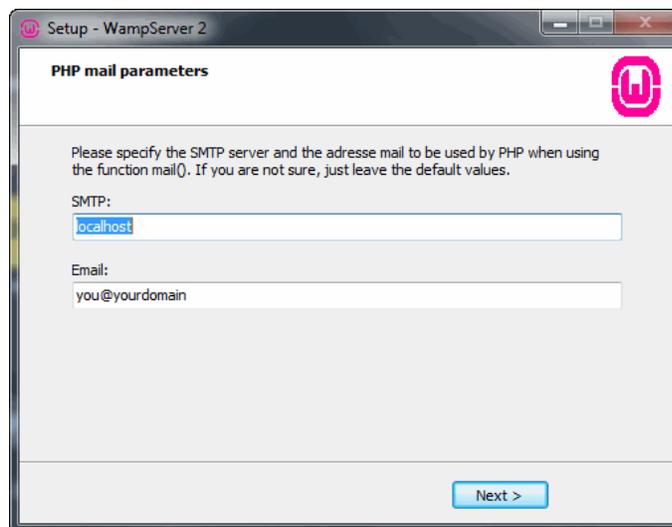


Figura 44 *Envío de email*

12.- Esta ventana es de finalización de la instalación del WampServer si se desea ejecutar el WampServer der clic en ejecutar WampServer (Lauch WampServer) y Finalizar (Finish).

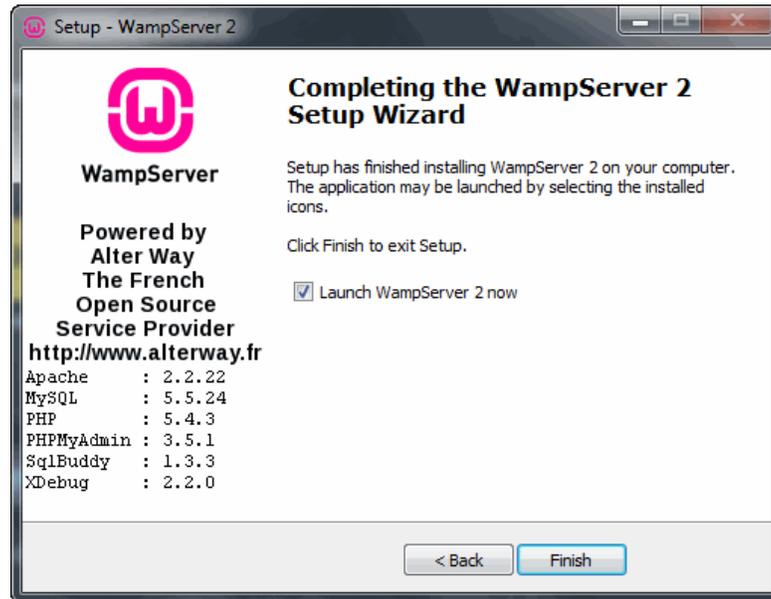


Figura 45 Final de instalación de WampServer

Instalación de Apache Tomcat.

1.-Dar clic derecho en el instalador ApacheTomcat Ejecutar como administrador.

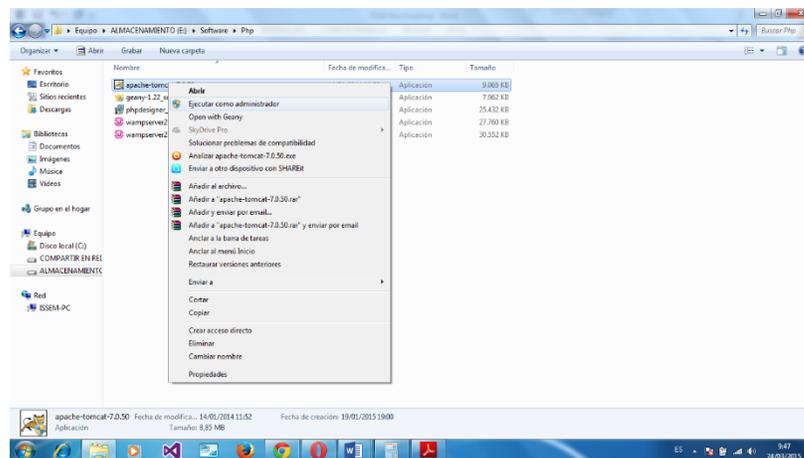


Figura 46 Ejecuta el instalador como administrador

2.- Luego desplegara una ventana de bienvenida, dar clic en siguiente (Next).



Figura 47 Pantalla de bienvenida

3.- Luego aceptar los términos de Licencia y siguiente (Next).

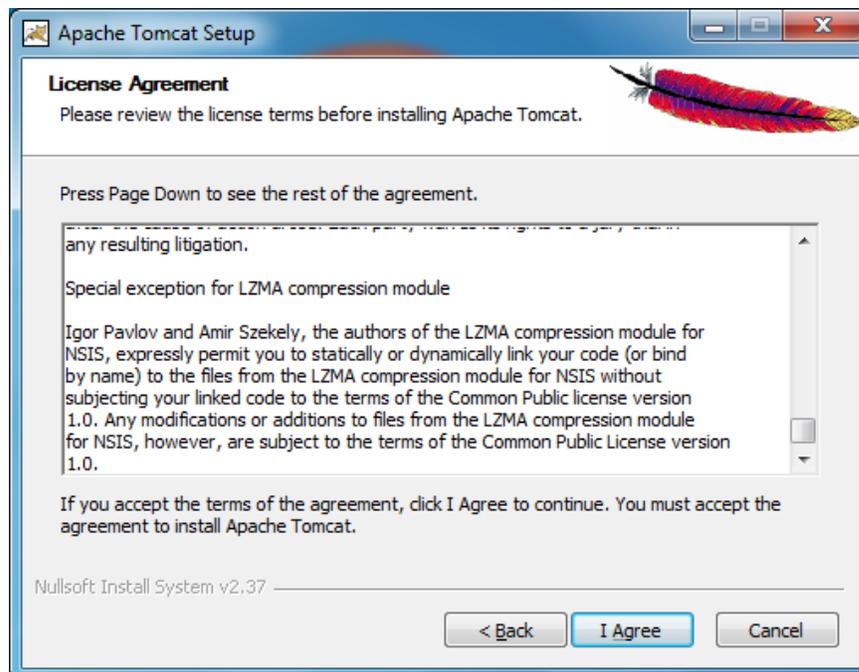


Figura 48 Acuerdo de licencia

4.- Luego no pedirá si deseamos instalar todos los complementos de Apache Tomcat
Seleccionamos todo y siguiente (Next).

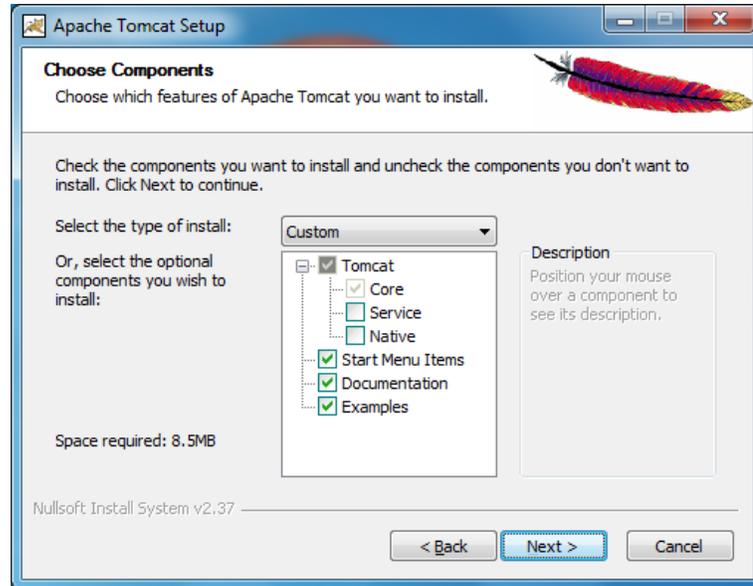


Figura 49 Seleccionamos los componentes que necesitamos

5.- Luego ubicar la instalación del Apache Tomcat (opcional).

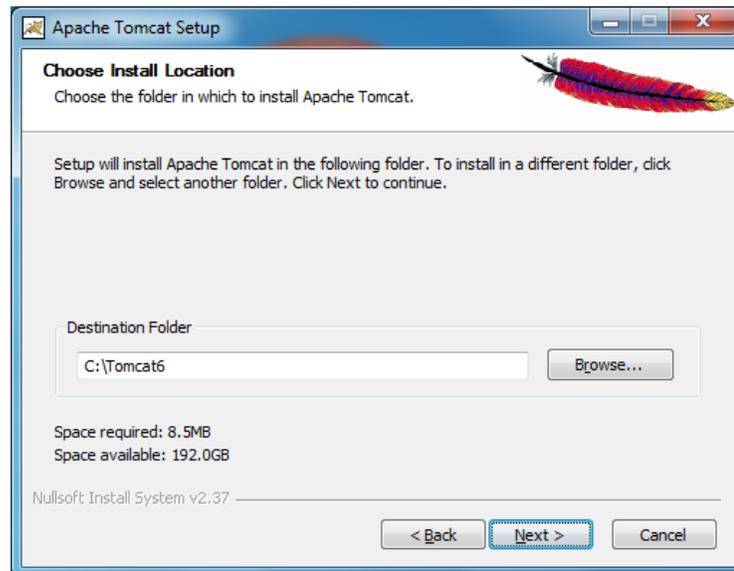


Figura 50 Selección de carpeta de instalación

6.- Aquí debemos si deseamos cambiar el puerto se lo puede realizar

HTTP/1.1 connector Port aquí ponemos el puerto que queramos o se prefiera ejemplo 8080 – 50 – 80, también se debe ingresar el usuario y password, dar clic en siguiente (Next).

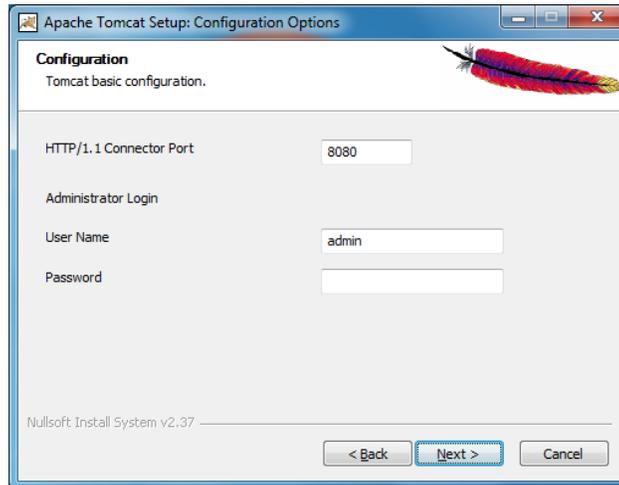


Figura 51 Configuración de puerto usuario y contraseña

7.- Luego ubicar la instalación del Apache Tomcat (opcional)

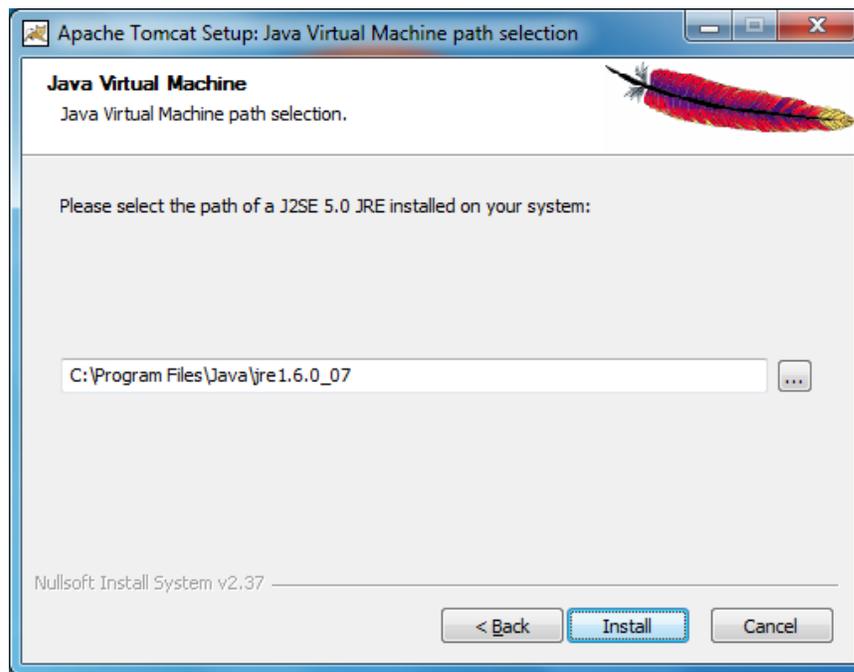


Figura 52 Inicio de instalación

8.- Esperar que se instale correctamente.

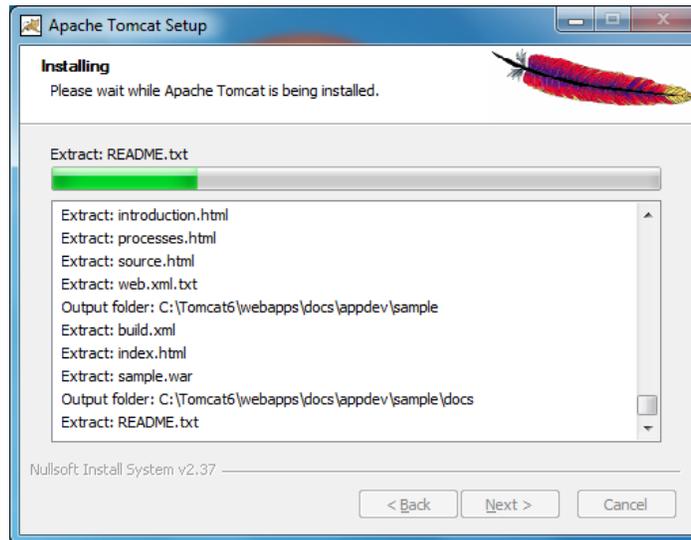


Figura 53 Progreso de instalación

9.- Esta ventana es de finalización de la instalación del Apache Tomcat si se desea ejecutar el Apache Tomcat seleccionar ejecutar Apache Tomcat (Run Apache Tomcat) y Finalizar (Finish).

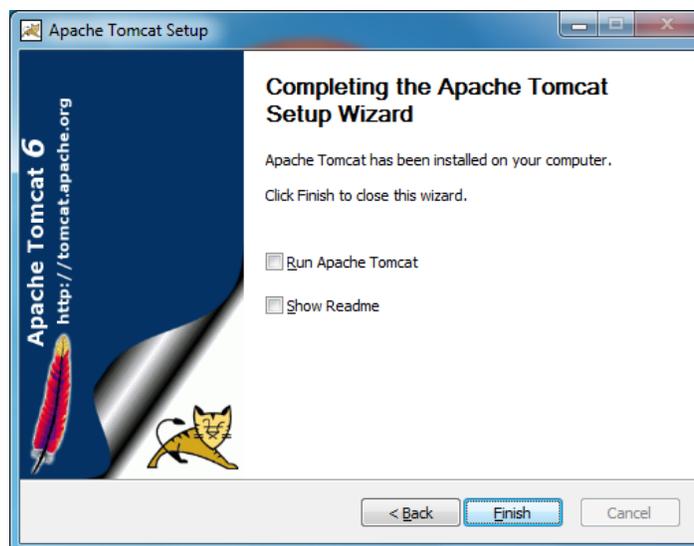


Figura 54 Final de instalación Apache Tomcat

Manual de usuario

1.- Ingresar al navegador.

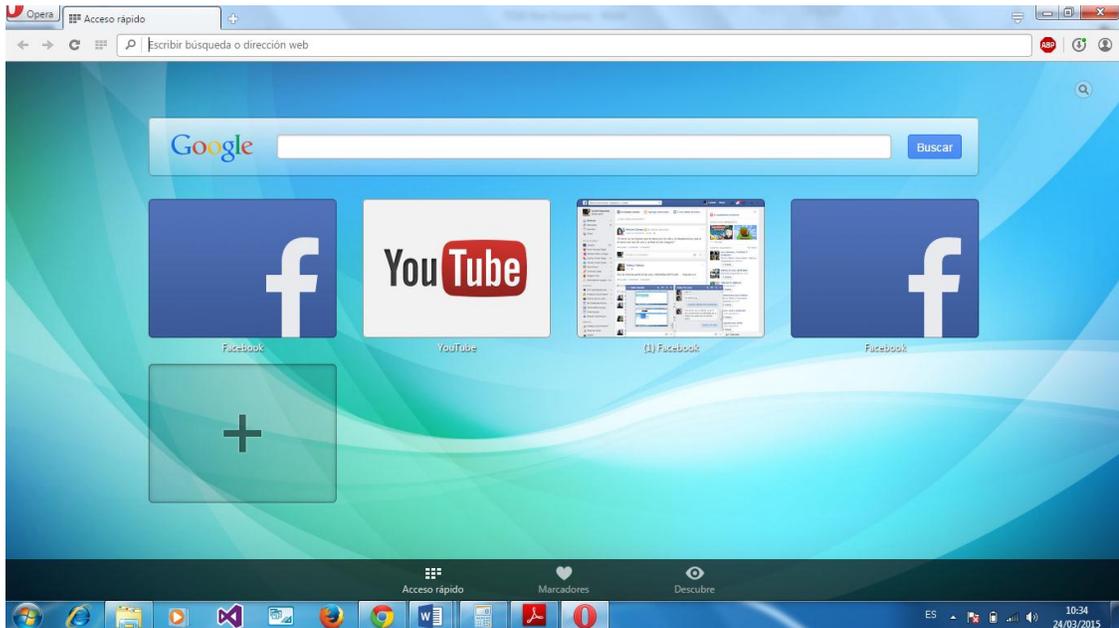


Figura 55 Navegador de internet

2.- Ingresar al URL correspondiente.

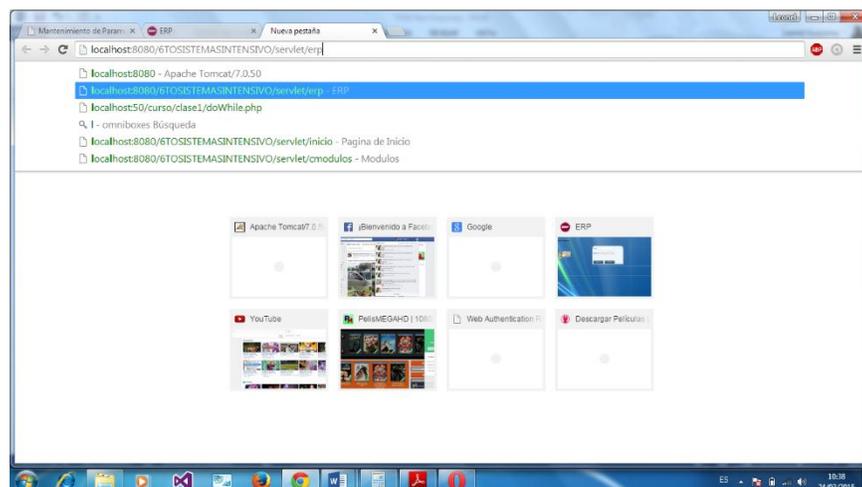


Figura 56 Ingreso de dirección o URL para abrir el sistema

3.- Dar clic en el botón INGRESAR.



Figura 57 Pantalla de ingreso al sistema

4.- Ingresar al módulo correspondiente en nuestro caso Modulo de Tesorería.

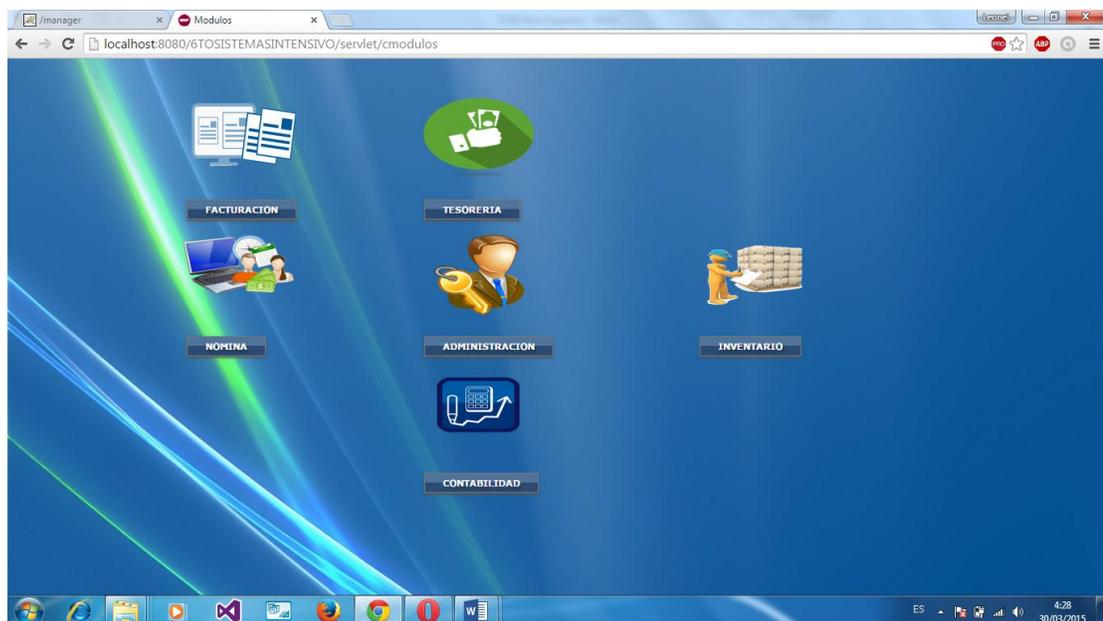


Figura 58 Muestra la pantalla con todos los módulos

5.- Ingresar el usuario y contraseña y dar clic en ingresar.

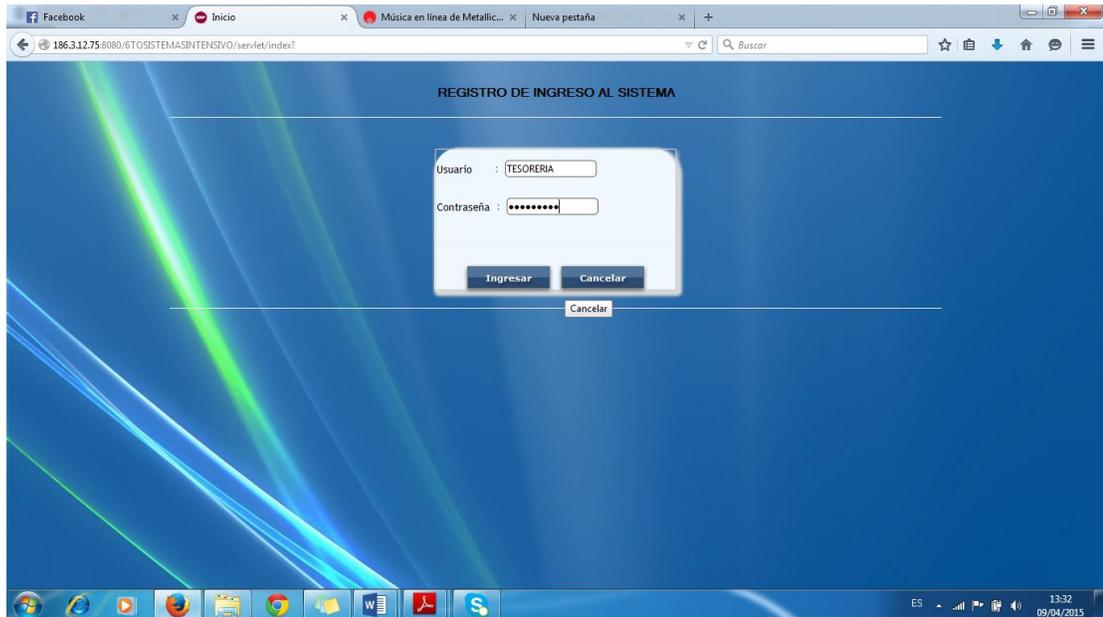


Figura 59 Pantalla en la cual ingresamos usuario y contraseña

6.- seleccionar la Empresa, Periodo y damos F5

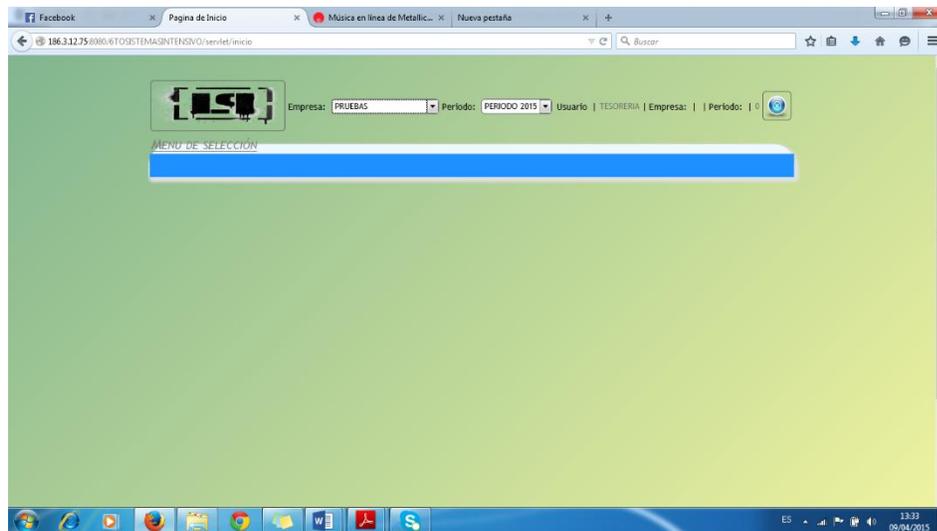


Figura 60 Selección de empresa y periodo

7.- Desplegara el menú de nuestro módulo

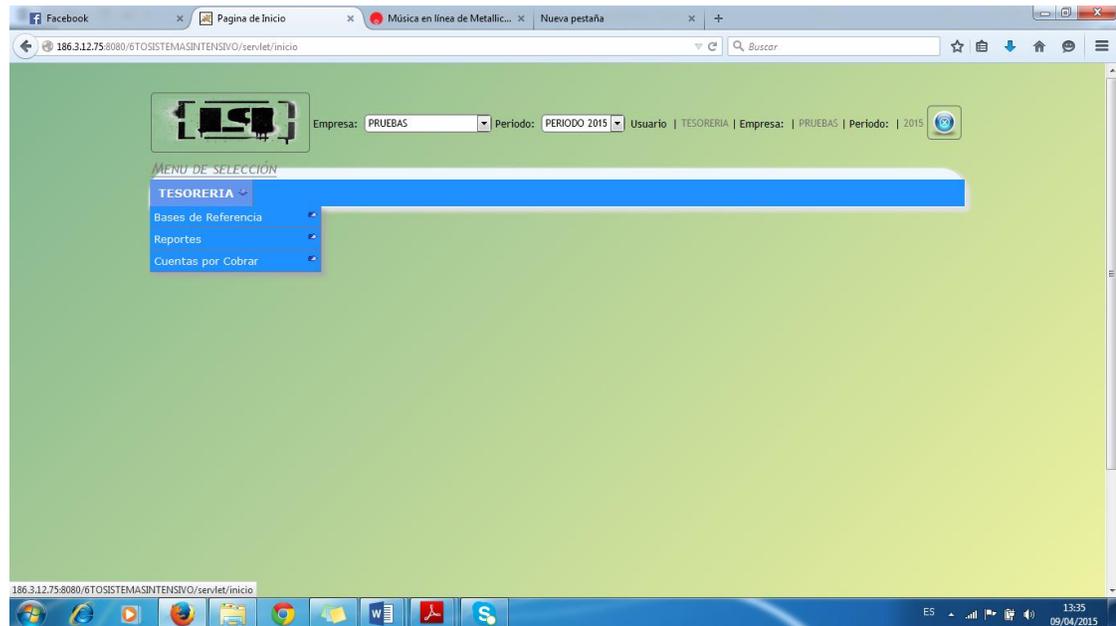


Figura 61 Lista las diferentes tareas que necesita el usuario

8.- Ingreso a registro de facturas

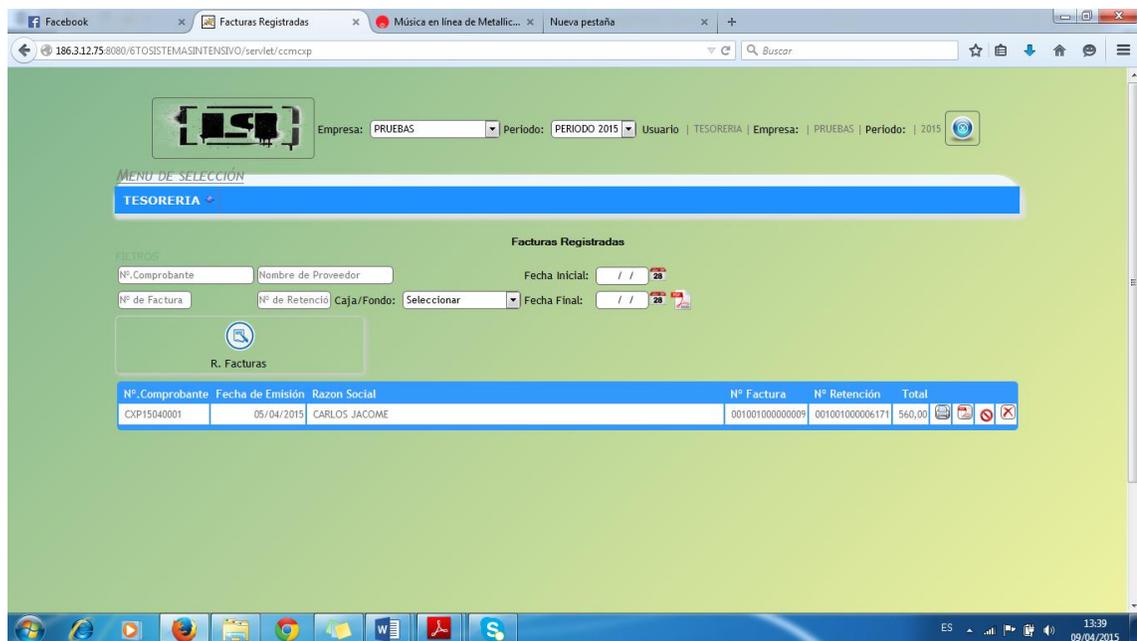


Figura 62 Muestra la pantalla de ingreso de facturas

9.-Ingreso a notas de crédito

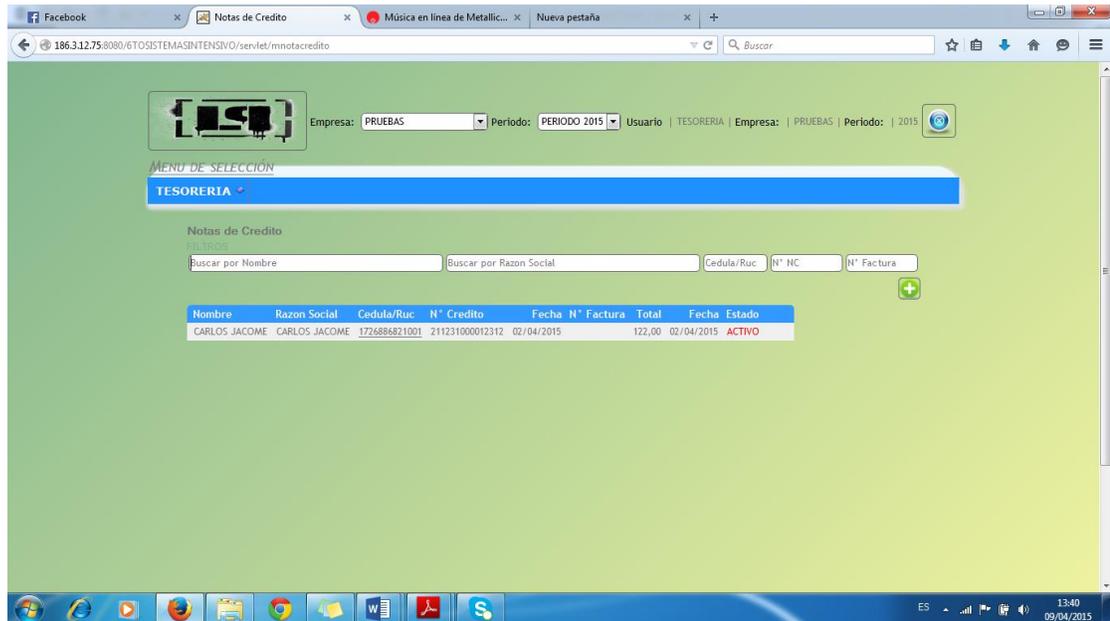


Figura 63 Muestra la pantalla de nota de crédito

10.- Ingreso a anticipos

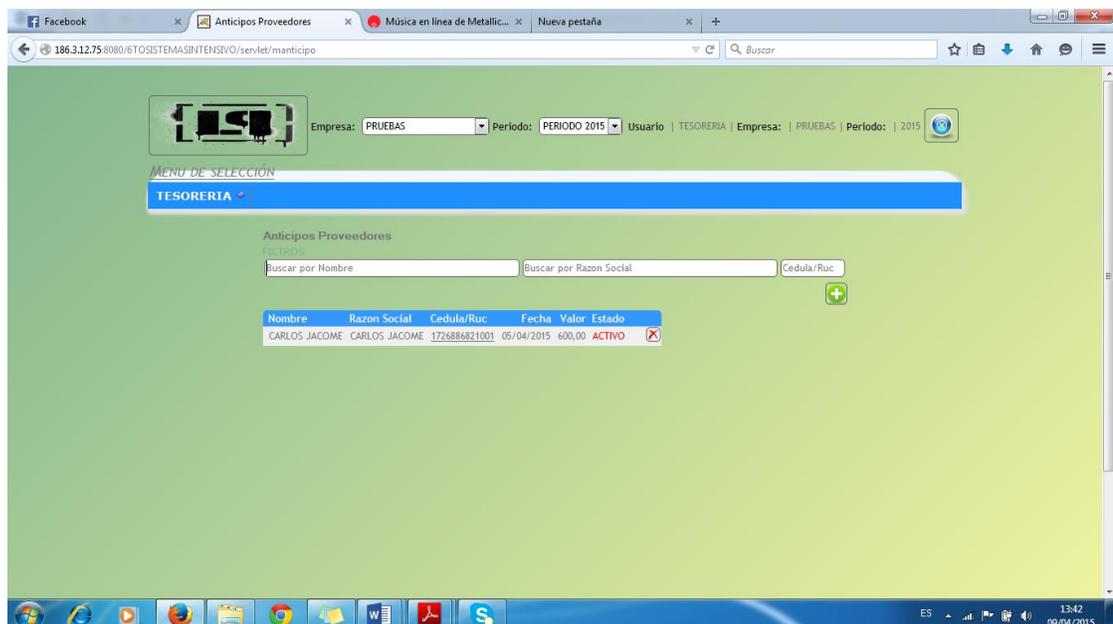


Figura 64 Muestra la pantalla de pago de anticipos

11.- Ingreso a Pagos.

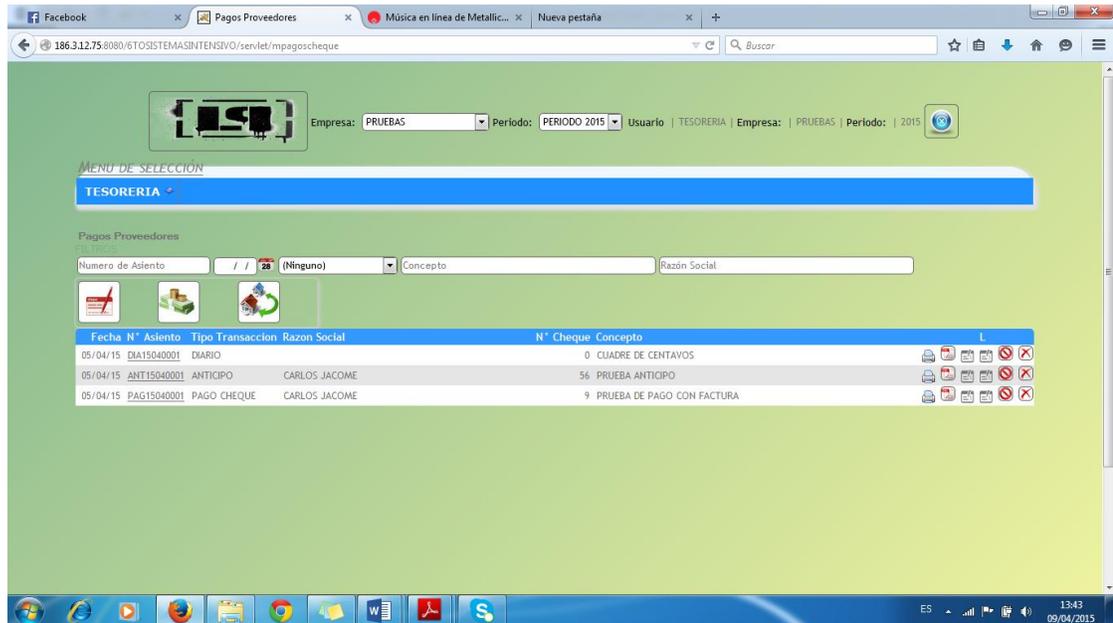


Figura 65 Muestra la pantalla de pagos

12.- Ingreso a cuadro de anticipos

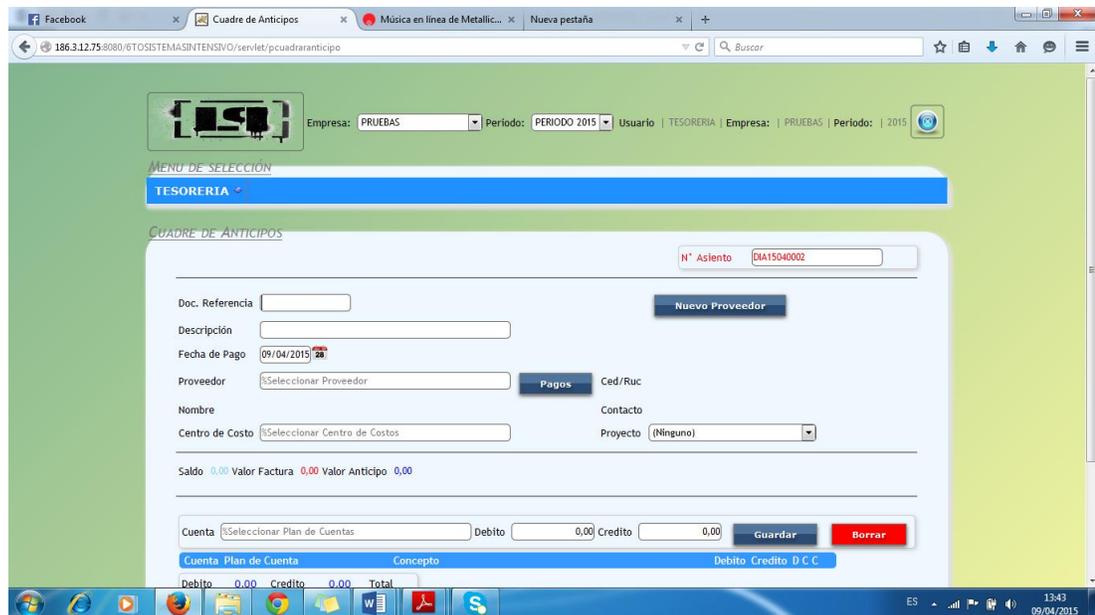


Figura 66 Muestra la pantalla de cuadro de anticipos

13.- Ingreso a bancos.

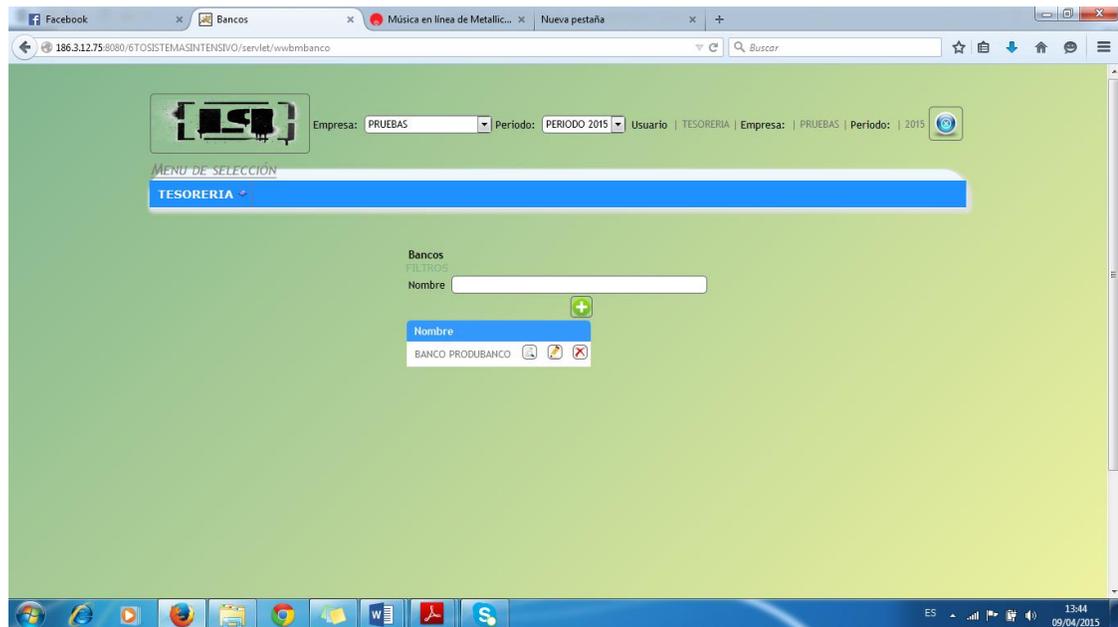


Figura 67 Muestra la pantalla de ingreso a bancos

14.- Consulta de retenciones.

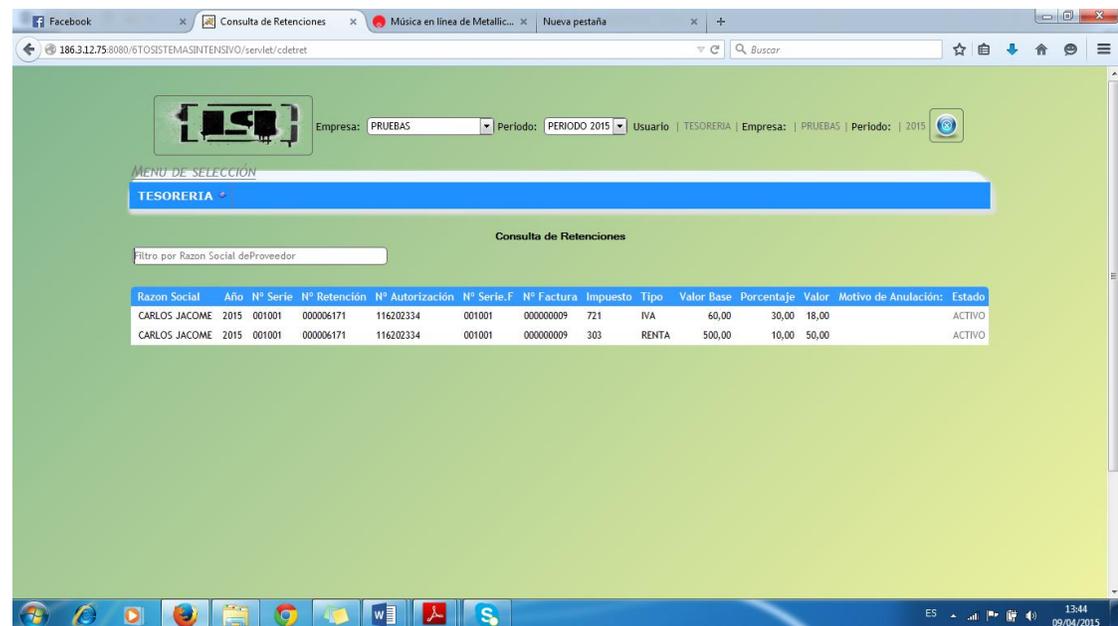


Figura 68 Muestra la pantalla de consulta de retenciones

15.- Ingreso a cuentas por cobrar o cartera.

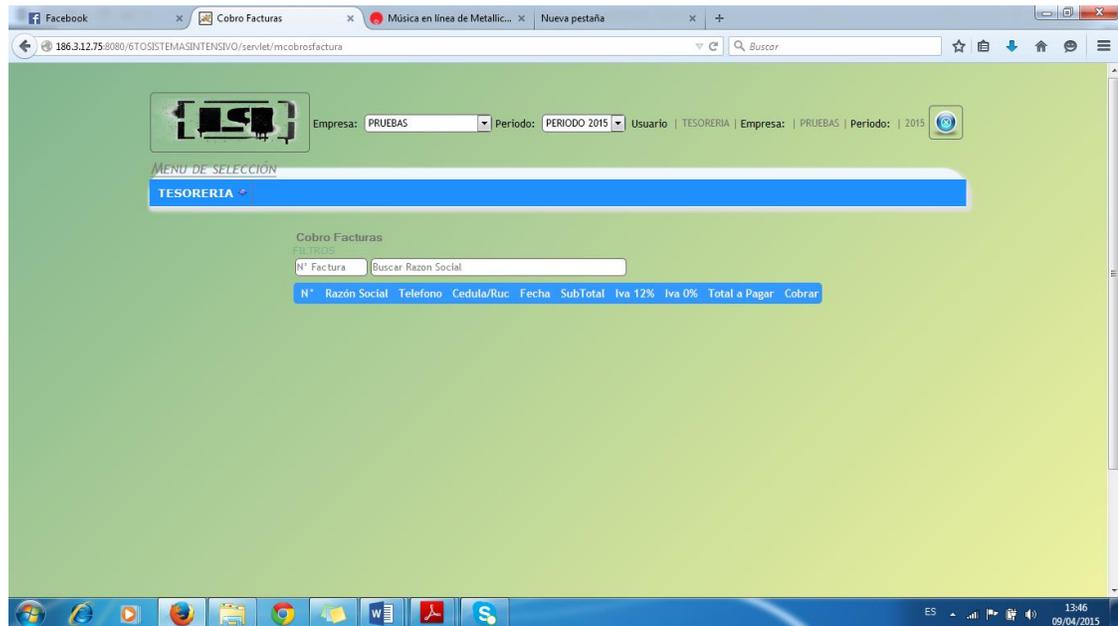


Figura 69 Muestra la pantalla de cuentas por cobrar

16.- Reporte de cobros realizados

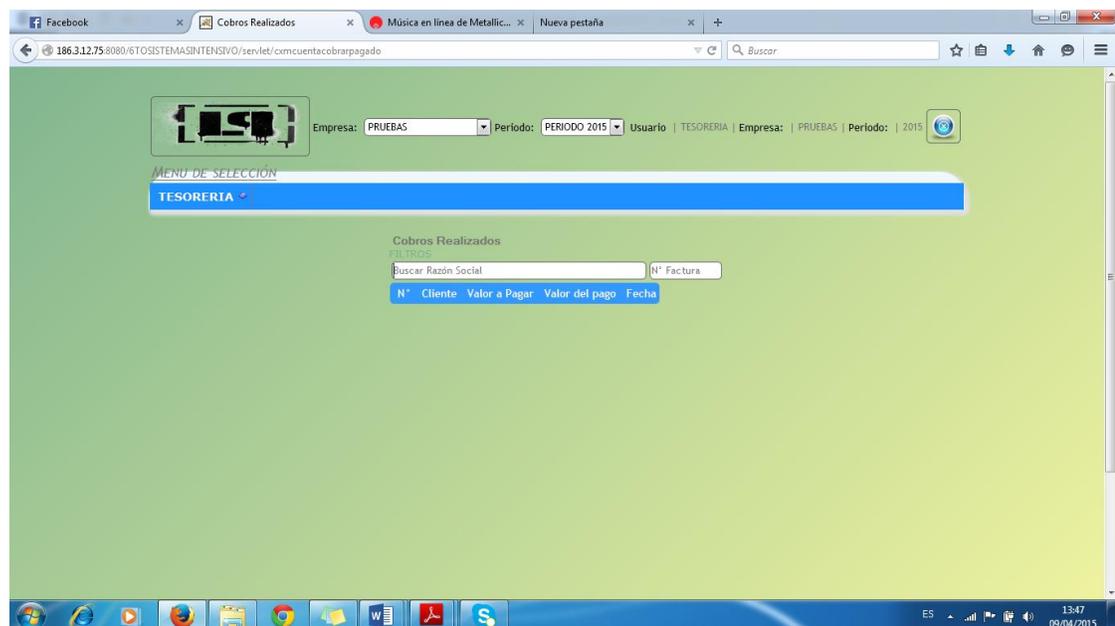


Figura 70 Muestra la pantalla de cuentas por cobrar

Manual Técnico

Especificar el nombre de las tablas de la base de datos y las clases de programación.

1.-La primera letra será en mayúsculas y definirá el Módulo principal en que se usa la tabla.

1. A.- Administración
2. B.- Bancos
3. C.- Contabilidad
4. D.- Smart Devices
5. E.- Ingeniería
6. F.- Activos Fijos
7. G.- Ctas. Por Pagar
8. I.- Inventario
9. N.- Nómina
10. M.- Compras
11. O.- Importaciones
12. P.- Producción
13. Q.- Calidad
14. R.- Reabastecimiento.
15. S.- Planificación
16. T.- Costos
17. V.- Ventas
18. W.- Control de Procesos

19. X.- Ctas.por Cobrar.

1. La segunda letra en minúsculas especifica el Tipo de Tabla que se está

definiendo en la Transacción.:

1. m.- Tablas Maestras.
2. c.- Cabeceras
3. d.- Detalles
4. a.- Acumulados
5. h.- Históricos
6. t.- Temporales.
7. p.- Parámetros.

SQL de base de datos Modulo tesoreria

```
-----  
-- Table structure for amemp  
-----  
  
DROP TABLE IF EXISTS `amemp`;  
  
CREATE TABLE `amemp` (  
  `AmEmpId` smallint(6) NOT NULL AUTO_INCREMENT,  
  `AmEmpNomb` varchar(100) NOT NULL,  
  `AmEmpRepN` varchar(40) NOT NULL,  
  `AmEmpReCed` char(10) NOT NULL,  
  `AmEmpRuc` char(13) NOT NULL,  
  `AmEmpDir` varchar(200) NOT NULL,  
  `AmEmpEmail` varchar(100) NOT NULL,
```

```
`AmEmpTelf` varchar(15) NOT NULL,  
`AmEmpRazS` varchar(100) NOT NULL,  
`AmEmpConNo` varchar(100) NOT NULL,  
`AmEmpConCed` char(10) NOT NULL,  
`AmEmpConEma` varchar(100) NOT NULL,  
`AmEmpImg` longblob NOT NULL,  
`AmEmpImg_GXI` varchar(2048) DEFAULT NULL,  
PRIMARY KEY (`AmEmpId`),  
KEY `UAMEMP` (`AmEmpNomb`) USING BTREE  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;  
  
-----  
-- Records of amemp  
-----  
  
INSERT INTO `amemp` VALUES ('1', 'PRUEVA', 'PRUEVA', 'PRUEVA', 'PRUEVA',  
'PRUEVA', 'PRUEVA', 'PRUEVA', 'PRUEVA', 'PRUEVA', 'PRUEVA', 'PRUEVA',  
0x505255455641, 'PRUEVA');  
  
-----  
-- Table structure for amperiodo  
-----  
  
DROP TABLE IF EXISTS `amperiodo`;  
  
CREATE TABLE `amperiodo` (  
  `AmEmpId` smallint(6) NOT NULL,  
  `AmPeriodoId` smallint(6) NOT NULL,
```

```
`AmPeriodoDesc` varchar(40) NOT NULL,  
`AmPeriodoEst` char(1) NOT NULL,  
`AmPeriodoFIni` date NOT NULL,  
`AmPeriodoFFin` date NOT NULL,  
`AmPeriodoUsr` varchar(20) NOT NULL,  
`AmPeriodoFhr` datetime NOT NULL,  
PRIMARY KEY (`AmEmpId`,`AmPeriodoId`),  
KEY `UAMPERIODO1` (`AmEmpId`,`AmPeriodoDesc`) USING BTREE,  
CONSTRAINT `amperiodo_ibfk_1` FOREIGN KEY (`AmEmpId`) REFERENCES  
`amemp` (`AmEmpId`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
-----  
-- Records of amperiodo  
-----  
  
INSERT INTO `amperiodo` VALUES ('1', '2015', 'PRUEVA', 'A', '2015-01-01', '2015-  
12-31', '1', '0000-00-00 00:00:00');  
  
-----  
-- Table structure for cdasiento  
-----  
  
DROP TABLE IF EXISTS `cdasiento`;  
  
CREATE TABLE `cdasiento` (  
  `AmEmpId` smallint(6) NOT NULL,  
  `AmPeriodoId` smallint(6) NOT NULL,
```

```
`CdAsientoId` smallint(6) NOT NULL,  
`CmAsientoId` smallint(6) NOT NULL,  
`CdAsientoCred` decimal(17,2) NOT NULL,  
`CdAsientoDebi` decimal(17,2) NOT NULL,  
`CdAsientoValor` decimal(17,2) NOT NULL,  
`CdAsientoEst` char(1) NOT NULL,  
`CmPlanCCuen` varchar(50) NOT NULL,  
`CdAsientoFech` date NOT NULL,  
PRIMARY KEY (`AmEmpId`,`AmPeriodoId`,`CdAsientoId`),  
KEY `ICDASIENTO1` (`AmEmpId`,`AmPeriodoId`,`CmAsientoId`) USING BTREE,  
KEY `UCDASIENTO` (`CdAsientoId`,`AmEmpId`) USING BTREE,  
KEY `UCDASIENTO1`  
(`AmEmpId`,`AmPeriodoId`,`CmPlanCCuen`,`CdAsientoFech`) USING BTREE,  
CONSTRAINT `cdasiento_ibfk_1` FOREIGN KEY (`AmEmpId`,`AmPeriodoId`,  
`CmAsientoId`) REFERENCES `cmasiento` (`AmEmpId`,`AmPeriodoId`,  
`CmAsientoId`),  
CONSTRAINT `cdasiento_ibfk_2` FOREIGN KEY (`AmEmpId`,`AmPeriodoId`,  
`CmPlanCCuen`) REFERENCES `cmplanc` (`AmEmpId`,`AmPeriodoId`,  
`CmPlanCCuen`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
-----  
-- Records of cdasiento  
-----
```

-- Table structure for cdatosretencion

```
DROP TABLE IF EXISTS `cdatosretencion`;  
  
CREATE TABLE `cdatosretencion` (  
  `AmEmpId` smallint(6) NOT NULL,  
  `CDatosRetencionId` smallint(6) NOT NULL,  
  `CDatosRetencionSerie` char(6) NOT NULL,  
  `CDatosRetencionAutoriz` varchar(10) NOT NULL,  
  `CDatosRetencionFechAuto` date NOT NULL,  
  `CDatosRetencionFechCadu` date NOT NULL,  
  `CDatosRetencionDesd` int(11) NOT NULL,  
  `CDatosRetencionHast` int(11) NOT NULL,  
  `CDatosRetencionUsr` varchar(20) NOT NULL,  
  `CDatosRetencionFhr` datetime NOT NULL,  
  `CDatosRetencionEst` char(1) DEFAULT NULL,  
  `CDatosRetencionNumUsa` int(11) NOT NULL,  
  PRIMARY KEY (`AmEmpId`, `CDatosRetencionId`),  
  KEY `UCDATOSRETENCION` (`CDatosRetencionId`, `AmEmpId`) USING BTREE,  
  KEY `UCDATOSRETENCION1` (`CDatosRetencionSerie`, `AmEmpId`) USING  
BTREE,  
  KEY `UCDATOSRETENCION2`  
(`CDatosRetencionFechCadu`, `CDatosRetencionHast`) USING BTREE,
```

```
CONSTRAINT `empresa` FOREIGN KEY (`AmEmpId`) REFERENCES `amemp`
(`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----

-- Records of cdatosretencion

-----

-----

-- Table structure for cmasiento

-----

DROP TABLE IF EXISTS `cmasiento`;

CREATE TABLE `cmasiento` (
  `AmEmpId` smallint(6) NOT NULL,
  `AmPeriodoId` smallint(6) NOT NULL,
  `CmAsientoId` smallint(6) NOT NULL,
  `CmAsientoNume` varchar(30) NOT NULL,
  `CmAsientoTipo` char(3) NOT NULL,
  `CmAsientoFecha` date NOT NULL,
  `CmAsientoBanco` smallint(6) NOT NULL,
  `CmAsientoBancoDetalle` varchar(12) NOT NULL,
  `CmAsientoConcep` varchar(60) NOT NULL,
  `CmAsientoDocRef` char(20) NOT NULL,
  `CmAsientoProveedor` smallint(6) NOT NULL,
  `CmAsientoCenCos` varchar(50) NOT NULL,
```

```
`CmAsientoAsiIni` char(1) NOT NULL,  
`CmAsientoProyecto` smallint(6) NOT NULL,  
`CmAsientoCheNum` int(11) NOT NULL,  
`CmAsientoCajRom` char(3) NOT NULL,  
`CmAsientoConBan` char(1) NOT NULL,  
`CmAsientoEsta` char(1) NOT NULL,  
`CmAsientoConciliado` char(1) NOT NULL,  
`CmAsientoValor` decimal(17,2) NOT NULL,  
`CmAsientoNTransF` varchar(40) NOT NULL,  
`TipoTransaccion` char(3) NOT NULL,  
  
PRIMARY KEY (`AmEmpId`,`AmPeriodoId`,`CmAsientoId`),  
KEY `UCMASIENTO` (`CmAsientoCheNum`) USING BTREE,  
KEY `UCMASIENTO1` (`CmAsientoNume`) USING BTREE,  
CONSTRAINT `cmasiento_ibfk_1` FOREIGN KEY (`AmEmpId`,`AmPeriodoId`)  
REFERENCES `amperiodo` (`AmEmpId`,`AmPeriodoId`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
-----  
-- Records of cmasiento  
-----  
-----  
  
-- Table structure for cmplanc  
-----  
  
DROP TABLE IF EXISTS `cmplanc`;
```

```
CREATE TABLE `cmplanc` (  
  `AmEmpId` smallint(6) NOT NULL,  
  `AmPeriodoId` smallint(6) NOT NULL,  
  `CmPlanCCuen` varchar(50) NOT NULL,  
  `CmPlanCDesc` varchar(200) NOT NULL,  
  `CmPlanCTCue` char(1) NOT NULL,  
  `CmPlanCCueP` varchar(50) NOT NULL,  
  `CmPlanCNive` smallint(6) NOT NULL,  
  `CmPlanCMask` smallint(6) NOT NULL,  
  `CmPlanCNat` char(1) DEFAULT NULL,  
  `CmPlanCUsr` varchar(20) NOT NULL,  
  `CmPlanCFhr` datetime NOT NULL,  
  PRIMARY KEY (`AmEmpId`, `AmPeriodoId`, `CmPlanCCuen`),  
  KEY `UCMPLANC` (`AmEmpId`, `CmPlanCCuen`, `AmPeriodoId`) USING BTREE,  
  CONSTRAINT `cmplanc_ibfk_1` FOREIGN KEY (`AmEmpId`, `AmPeriodoId`)  
  REFERENCES `amperiodo` (`AmEmpId`, `AmPeriodoId`)  
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
-----  
  
-- Records of cmplanc  
  
-----  
  
-----  
  
-- Table structure for cmtipsus  
  
-----
```

```
DROP TABLE IF EXISTS `cmtipsus`;

CREATE TABLE `cmtipsus` (
  `AmEmpId` smallint(6) NOT NULL,
  `CmTipSusId` char(3) NOT NULL,
  `CmTipSusDesc` varchar(200) NOT NULL,
  `CmTipSusCCuen` varchar(50) NOT NULL,
  `CmTipSusCCuenAuX` varchar(50) NOT NULL,
  `CmTipSusUsr` varchar(20) NOT NULL,
  `CmTipSusFhr` datetime NOT NULL,
  PRIMARY KEY (`AmEmpId`,`CmTipSusId`),
  CONSTRAINT `cmtipsus_ibfk_1` FOREIGN KEY (`AmEmpId`) REFERENCES
`amemp` (`AmEmpId`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----
-- Records of cmtipsus
-----
-----
-- Table structure for cpgrupo
-----

DROP TABLE IF EXISTS `cpgrupo`;

CREATE TABLE `cpgrupo` (
  `AmEmpId` smallint(6) NOT NULL,
  `CpGrupoId` smallint(6) NOT NULL,
```

```
`CpGrupoDesc` varchar(100) NOT NULL,  
`CpGrupoUsr` varchar(20) NOT NULL,  
`CpGrupoFhr` datetime NOT NULL,  
PRIMARY KEY (`AmEmpId`,`CpGrupoId`),  
KEY `UCPGRUPO` (`CpGrupoDesc`,`AmEmpId`,`CpGrupoId`) USING BTREE,  
CONSTRAINT `cpgrupo_ibfk_1` FOREIGN KEY (`AmEmpId`) REFERENCES  
`amemp` (`AmEmpId`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
-----  
-- Records of cpgrupo  
-----  
  
-----  
-- Table structure for cporcenreten  
-----  
  
DROP TABLE IF EXISTS `cporcenreten`;  
CREATE TABLE `cporcenreten` (  
  `AmEmpId` smallint(6) NOT NULL,  
  `CPorcenRetenId` smallint(6) NOT NULL,  
  `AmPeriodoId` smallint(6) NOT NULL,  
  `CmPlanCCuen` varchar(50) NOT NULL,  
  `CPorcenRetenDesc` varchar(200) NOT NULL,  
  `CPorcenRetenPorcent` decimal(5,2) NOT NULL,  
  `CPorcenRetenTipo` char(1) NOT NULL,
```

```
`CPorcenRetenTipoBs` char(1) NOT NULL,  
`CPorcenRetenEst` char(1) NOT NULL,  
`CPorcenRetenFech` date NOT NULL,  
`CPorcenRetenCod` char(3) NOT NULL,  
PRIMARY KEY (`AmEmpId`,`CPorcenRetenId`,`AmPeriodoId`),  
KEY `ICPORCENRETEN1` (`AmEmpId`,`AmPeriodoId`,`CmPlanCCuen`) USING  
BTREE,  
CONSTRAINT `cporcenreten_ibfk_1` FOREIGN KEY (`AmEmpId`,`AmPeriodoId`,  
`CmPlanCCuen`) REFERENCES `cmplanc` (`AmEmpId`,`AmPeriodoId`,  
`CmPlanCCuen`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Bibliografía

Barcés, M. E. (01 de 03 de 2015). *Patrones de Arquitectura Web*. Obtenido de Patrones de Arquitectura Web: <http://chavezbarces.blogspot.com/2012/06/normal-0-21-false-false-false-es-pe-x.html>

Contable, E. (11 de 12 de 2014). *Ecuador Contable*. Obtenido de Ecuador Contable: <http://ecuadorcontable.com>

España, A. (13 de 03 de 2015). *Estandares de programación*. Obtenido de Estandares de programación: <http://sistemas.mag.go.cr/SoporteTecnico/Est%C3%A1ndares%20de%20Sistemas.pdf>

finanzas, M. d. (12 de 12 de 2014). *Ministerio de finanzas*. Obtenido de Ministerio de finanzas: <http://www.finanzas.gob.ec>

Flores, B. (12 de 01 de 2015). *Java Foundations*. Obtenido de Java Foundations: <http://javafoundations.blogspot.com/2010/07/java-estandares-de-programacion.html>

García, C. (26 de 12 de 2014). *Monografias.com*. Obtenido de Monografias.com: <http://www.monografias.com/trabajos81/ensayo-contabilidad/ensayo-contabilidad.shtml>

Hernesto, H. (03 de 02 de 2015). *Pruebas de Aceptación* . Obtenido de Pruebas de Aceptación : • <http://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&c>

d=11&cad=rja&uact=8&ved=0CFcQFjAK&url=http%3A%2F%2Fwww.inf.utfs
m.cl%2F~visconti%2Fxp%2FPruebas_Aceptacion_2.doc&ei=7_D7U871CKTes
ATYrYGgCg&usg=AFQjCNF1m1WY3su7nRrJlsXIzCmL8HnllA&sig2=yWj

Melez, J. (02 de 02 de 2015). *Ingeniería de Software*. Obtenido de Ingeniería de
Software: http://www.codecompiling.net/files/slides/IS_clase_04_diseno_UI.pdf

Peña, A. (01 de 01 de 2015). *Estandades de programacion*. Obtenido de Estandades de
programacion:
[https://docs.google.com/document/d/1UjUZJpNO9rc4uN98JKteih4JpoeY3lamjy
WwxDKqcIg/edit?pli=1%20%E2%80%A2%20http://www.fing.edu.uy/tecnoinf/
mvd/cursos/ingsoft/material/teorico/is06-DisenioIU.pdf](https://docs.google.com/document/d/1UjUZJpNO9rc4uN98JKteih4JpoeY3lamjyWwxDKqcIg/edit?pli=1%20%E2%80%A2%20http://www.fing.edu.uy/tecnoinf/mvd/cursos/ingsoft/material/teorico/is06-DisenioIU.pdf)

Públicas, M. d. (02 de 03 de 2015). *Ministerio de Administraciones Públicas*. Obtenido
de Ministerio de Administraciones Públicas :
[https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&c
d=14&cad=rja&uact=8&ved=0CHIQFjAN&url=https%3A%2F%2Fadministraci
onelectronica.gob.es%2Fpae_Home%2Fdms%2Fpae_Home%2Fdocumentos%2
FDocumentacion%2FMetodologias-y-guias%2FMetricav3%2FMETRICA](https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&frm=1&source=web&cd=14&cad=rja&uact=8&ved=0CHIQFjAN&url=https%3A%2F%2Fadministracionelectronica.gob.es%2Fpae_Home%2Fdms%2Fpae_Home%2Fdocumentos%2FDocumentacion%2FMetodologias-y-guias%2FMetricav3%2FMETRICA)

Rodriguez, R. A. (02 de 02 de 2015). *Mi Baúl XD*. Obtenido de Mi Baúl XD:
<https://javashi.wordpress.com/acerca-de/>

UML. (01 de 01 de 2015). *OMG*. Obtenido de OMG: <http://www.uml.org>

Velez, J. (03 de 02 de 2015). *NetBeansInstalacion*. Obtenido de NetBeansInstalacion:
<http://joanpaon.wordpress.com/2013/05/21/instalacion-de-netbeans-7-3/>

Wikipedia. (01 de 01 de 2015). *Wikipedia*. Obtenido de Wikipedia:

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

HECTOR, C. (04 de ENERO de 2014). *mgkid.com*. Obtenido de imgkid.com:

<http://imgkid.com/estate-planning-diagram.shtml>

García, O. G. (02 de Febrero de 2015). *unhiloenlared.blogspot.com/*. Obtenido de

unhiloenlared.blogspot.com/: <http://unhiloenlared.blogspot.com/2012/05/diseño-de-interfaces-prototipos-o.html>