

APOYO EN EL CAMBIO DE PLATAFORMA: LENGUAJE DE DESARROLLO,
BASE DE DATOS Y FUNCIONALIDADES; PARA EL SISTEMA DE
CONTABILIDAD DE LA EMPRESA EVACC LTDA.

CARLOS FERNANDO VEGA REYES

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2013

APOYO EN EL CAMBIO DE PLATAFORMA: LENGUAJE DE DESARROLLO,
BASE DE DATOS Y FUNCIONALIDADES; PARA EL SISTEMA DE
CONTABILIDAD DE LA EMPRESA EVACC LTDA.

CARLOS FERNANDO VEGA REYES

TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE
INGENIERO DE SISTEMAS

DIRECTOR

Ing. CARLOS ADOLFO BELTRAN CASTRO

TUTOR

Ing. CARLOS FELIPE REYES CONTRERAS

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA

2013

DEDICATORIA

Dedico este esfuerzo al Señor Jesucristo, único creador de todo cuanto existe y quien ha puesto en mí, aliento de gracia y sabiduría para sobrellevar todo lo que se ha puesto en mi camino. Igualmente a mis Padres Fernando y Mery quienes han dedicado su vida entera a formarme como un verdadero ejemplo y nunca escatimaron esfuerzos para que todo esto fuera posible; de la misma manera que mi hermosa novia Erika a quien le debo tantos momentos gratos y especiales, por ser mi apoyo incondicional y caminar conmigo de la mano durante tanto tiempo. A mis amigos y familiares que nunca perdieron su fe en mí.

AGRADECIMIENTOS

“Y nosotros, pueblo tuyo... te daremos gracias para siempre; a todas las generaciones hablaremos de tu alabanza”

<<Salmos, 79:13 (RVR1960)>>

Un caluroso y sentido abrazo para mis padres que han sido mis pilares fundamentales en el desarrollo de mi educación, a ellos infinitas gracias por acompañarme y apoyarme en todo momento.

Uno de los más grandes regalos que la vida me ha entregado, mi hermosa novia Erika, a ti mil gracias por ser esa compañera incondicional que luchó conmigo de la mano durante estos hermosos años.

El respaldo y la formación brindada por todos mis docentes universitarios, de quienes adquirí invaluable enseñanzas son motivo de agradecimiento por parte mía; La disposición del profesor Carlos Felipe Reyes quien comedidamente fue el tutor del presente proyecto y que permitió que su empresa estuviese por algún tiempo siendo centro de academia a mi favor, de la misma manera que agradezco al profesor Carlos Beltrán quien dio su guía para un avance exitoso de esta obra.

Por último, mi más profundo sentimiento de gratitud a todas aquellas personas que directa o indirectamente me apoyaron y que hicieron realidad este sueño que hoy se materializa; un listado increíble de personas increíbles que demuestra lo más valioso que tengo en la vida: mis amigos.

TABLA DE CONTENIDO

INTRODUCCIÓN	14
1. OBJETIVOS	16
1.1 GENERAL:.....	16
1.2 ESPECÍFICOS:.....	16
2. PLANTEAMIENTO Y JUSTIFICACION DEL PROBLEMA	17
3. DESCRIPCION DE LA EMPRESA	18
3.1 Descripción.	18
3.2 Misión	18
3.3 Visión	18
3.4 Valores.....	19
3.5 Estructura Organizacional.....	19
4. IMPACTO.....	20
5. MARCO METODOLÓGICO	21
5.1 Metodologías de desarrollo de software.	21
5.2 Selección de la metodología a implementar.	33
6. MARCO TEÓRICO DE REFERENCIA	34
6.1 Fundamento teórico general.	34
6.1.1 Sistemas de Información.....	34
6.1.2 Plataforma .NET	37
6.1.3 Herramientas de Desarrollo - Visual Studio .NET	43
6.1.4 Bases de Datos.....	45
6.1.5 Contabilidad Colombiana vigente	49
6.2 Fundamento teórico específico:	51
6.2.1 Contabilidad por partida doble	51
6.2.2 Imputación Contable	53
6.2.3 Librerías dinámicas o Librerías de enlace dinámico:	54
6.2.4 Clases y Objetos.....	56
6.2.5 PostgreSQL	57
6.2.6 NPGSQL.....	60
7. DESARROLLO DE LA PRÁCTICA	62
7.1 Antecedentes	62

7.1.1 Descripción de 'Contar' en su versión anterior.....	62
7.1.2 Base de datos:	69
7.2 Levantamiento de requerimientos.....	69
7.2.1 Documentación.....	69
7.2.2 Diseño de la base de datos.....	74
7.3 Módulos a desarrollar	76
7.3.1 Descripción.....	76
7.3.2 Objetivos.....	77
7.3.3 Requisitos funcionales y no funcionales	79
7.4 Ejecución y codificación.....	86
7.4.1 Refactorización de código e implementación de algoritmos	86
7.4.2 Proceso de diseño	91
7.5 Entrega de la herramienta software	93
7.5.1 Acoplamiento del proyecto de ejecución.....	93
7.5.2 Pruebas de montaje.....	94
7.5.3 Inconvenientes presentados durante las pruebas.....	96
7.5.4 Documentación de entrega.....	98
CONCLUSIONES	99
RECOMENDACIONES.....	101
BIBLIOGRAFÍA.....	102
ANEXOS.....	103

TABLA DE FIGURAS

Figura 1. Estructura organizacional EVACC Ltda	19
Figura 2. Modelo de ciclo de desarrollo DRA.....	23
Figura 3. Fases de desarrollo – Metodología XP	24
Figura 5. Modelo evolutivo de desarrollo de software	27
Figura 7. Modelo de Proceso unificado	28
Figura 8. Detalle Gráfico Metodología SCRUM, Ficha Sinóptica	33
Figura 9. Arquitectura del .NET Framework.....	41
Figura 10. Modelo de ejecución del CLR	43
Figura 11. Imagen simplificada de un sistema de base de datos	45
Figura 12. Arquitectura PostgreSQL	58
Figura 13. Logo inicial aplicativo anterior ‘Contar’	63
Figura 14. Formulario maestro contenedor, aplicativo anterior ‘Contar’	63
Figura 15. Distribución de los iconos de navegación.....	64
Figura 16. Entrega de informes	65
Figura 17. Error en formulario de búsqueda módulo PUC	66
Figura 18. Defectos de usabilidad en el módulo de centros de costo	67
Figura 19. Diagrama de casos de uso, Módulo Plan único de cuentas	79
Figura 20. Diagrama de casos de uso, Módulo Tipo de comprobante.....	79
Figura 21. Diagrama de casos de uso, Módulo Terceros	80
Figura 22. Diagrama de casos de uso, Módulo Centros de costos.....	80
Figura 23. Diagrama de casos de uso, Módulo Bancos.....	80
Figura 24. Diagrama de casos de uso, Módulo Gestión de comprobantes.....	81
Figura 25. Diagrama de casos de uso, Módulo Diseño de informes lineales...	81
Figura 26. Diagrama de casos de uso, Módulo Utilidades.....	81
Figura 27. Nuevo logo de la herramienta software	92
Figura 28. Características del equipo de prueba de instalación numero 1	95
Figura 29. Características del equipo de prueba de instalación numero 2	95
Figura 30. Características del equipo de prueba de instalación numero 3	96
Figura 31. Informe de error por inicio de sesión secundario	97
Figura 32. Informe de error por falta de paquete de lenguaje	97

LISTA DE TABLAS

<i>Tabla 1. Representación de la técnica contable: Partida doble</i>	<i>52</i>
<i>Tabla 2. Designación de procesos para la estandarización de entidades</i>	<i>70</i>
<i>Tabla 3. Nombre de las tablas pertenecientes a la nueva base de datos del sistema según nueva convención.</i>	<i>75</i>
<i>Tabla 4. Objetivos de módulos desarrollados</i>	<i>77</i>
<i>Tabla 5. Relación de métodos abreviados para funciones</i>	<i>90</i>

LISTA DE ANEXOS

		Página
ANEXO A.	EVALUACION DEL RENDIMIENTO DE LA APLICACIÓN CONTAR EN SU VERSION ANTERIOR (Evaluación realizada con la herramienta “Análisis de Rendimiento” de Visual Studio 2010)	104
ANEXO B.	LISTADO DE TABLAS EXISTENTES EN LA BASE DE DATOS DE LA HERRAMIENTA CONTAR EN SU VERSION ANTERIOR	108
ANEXO C.	ACTA DE COMPROMISO DE ENTREGA DE HERRAMIENTA SOFTWARE CONTAR EN SU VERSION 3.0	119
ANEXO D.	LISTADO DE TABLAS DEFINITIVAS CORRESPONDIENTES A LA BASE DE DATOS DE LA HERRAMIENTA CONTAR EN SU VERSION 3.0 IMPLEMENTADAS EN POSTGRESQL	121
ANEXO E.	DIAGRAMA RELACIONAL DE LA BASE DE DATOS IMPLEMENTADA EN POSTGRESQL	127
ANEXO F.	DISTRIBUCION DEL NUEVO MENU DENTRO DE LA BARRA DE COMANDOS DEL FORMULARIO PRINCIPAL	129
ANEXO G.	PRUEBAS DE MONTAJE REALIZADAS EN DIFERENTES EQUIPOS	134
ANEXO H.	DOCUMENTO DE ENTREGA DE LA HERRAMIENTA CONTAR EN SU VERSION 3.0	136

RESUMEN

Título: APOYO EN EL CAMBIO DE PLATAFORMA: LENGUAJE DE DESARROLLO, BASE DE DATOS Y FUNCIONALIDADES; PARA EL SISTEMA DE CONTABILIDAD DE LA EMPRESA EVACC LTDA.*

Autor: Carlos Fernando Vega Reyes**

Palabras claves: contabilidad, funcionalidad, reingeniería, CONTAR – herramienta software – PostgreSQL como gestor de base de datos, NPGSQL

Descripción: la renovación de los procesos en toda organización se hace cada vez más necesaria teniendo en cuenta el avance de la tecnología y los retos del mundo actual. La forma en la que necesitamos acceder a nuestra información cada día juega un papel muy importante; es gracias a esto que a través de este proyecto se plantea el inicio de una refactorización del sistema contable de la empresa EVACC Ltda. La herramienta software “CONTAR” como parte vital de la administración anteriormente se encontraba elaborada en lenguajes de programación poco eficientes y bajo algoritmos de elevado consumo de recursos de manera que se hace necesaria la implementación de una nueva plataforma cuyos entornos presenten exigencias más altas. De la misma manera se contaban con las bases de datos implementadas en un software comercial como lo es ACCESS ®; un producto de la compañía Microsoft ® lo cual presenta algunas desventajas en comparación al desarrollo en plataformas de software libre como el gestor de base de datos PostgreSQL que se encuentra liberado bajo la licencia de software libre permisiva BSD (Berkeley Software Distribution).

El sistema contable de la empresa EVACC Ltda presentaba algunas deficiencias en su desarrollo funcional; tales como la falta de informes, búsquedas ágiles e implementaciones en múltiples clientes. Con el inicio de labores que se plantea en este proyecto se pretenden reducir dichas deficiencias y de la misma forma aportar al proceso evolutivo organizacional de EVACC Ltda; además de la mejora en la utilización de algoritmos más eficientes, se pretende dejar un esquema comprensible pero robusto de los procesos contables sistematizados que son necesarios para cada una de las tareas propias de dicha empresa.

* Proyecto de grado

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e informática, Director: Carlos Adolfo Beltrán, Tutor: Carlos Felipe Reyes

ABSTRACT

Title: SUPPORT ON THE PLATFORM CHANGE: DEVELOPMENT LANGUAGE, DATABASE AND FUNCTIONALITIES FOR THE ACCOUNTING SYSTEM OF EVACC LTDA ENTERPRISE.*

Author: CARLOS FERNANDO VEGA REYES.**

Keywords: accounting, functionalities, reengineering, CONTAR -software tool- PostgreSQL as database manager, NPGSQL.

The renovation of the process in all organization is increasingly necessary if contemplate the advance of technology and the challenges of the actual society. The way that people need for accessing to information everyday play an important role; because of that, through this project propose the beginning of the accounting system's EVACC Ltda enterprise re-factorization. "CONTAR" software tool as a vital part of the administration, previously was made in not efficient program languages and algorithm of lofty resource consumption so that is necessary the implementation of a new platform whose environment have more lofty requirements. Of same form, the database was implemented in commercial software such ACCESS™: a Microsoft™ company product, its present some disadvantages in comparison of development in free software platforms such the PostgreSQL database manager that is liberate under the permissive BSD (Berkeley Software Distribution) free software license.

The accounting system of EVACC Ltda enterprise had some deficient in his functional development; such lack of reports, ágile searches and implementations in multiple clients. With the beginning of tasks previously enounced, this project pretend minimize those deficient also contribute to the process organizational evolutionary of EVACC Ltda; as well, the improvement in the usage of algorithms more efficient, is intended to make a apprehensible but hardy schema of the systematized accounting process that are necessary for each tasks own of the enterprise.

* Degree Work.

** Faculty of physical and mechanical engineering. School of Systems and Computer engineering. Director: Ing Carlos Adolfo Beltrán Tuthor: Ing Carlos Felipe Reyes.

INTRODUCCION

El patrimonio de una empresa se encuentra a merced de los procesos contables que allí se lleven; un buen manejo administrativo y contable de sus recursos puede llevar a cualquier negocio a la cima del éxito o a la quiebra por su mal manejo; este concepto va de la mano con el apoyo que la tecnología actual le puede brindar a cualquier organización, si una empresa decide sistematizar sus procesos está haciendo una de las mejores jugadas administrativas que le favorecerán a futuro. Cada una de las dependencias de una estructura organizacional requiere y, tal vez, dependa de otras, es por esto que EVACC Ltda ha tomado la decisión de aplicarle procesos de reingeniería a su área contable como soporte a un proceso de incursión a los nuevos desafíos que el mundo actual le propone. La empresa cuenta con algún tiempo en el mercado del software, periodo durante el cual ha conservado su información en software de manejos pesados, la interfaz fue desarrollada con el lenguaje de programación visual 6.0 –cuyo modelo no es orientado a objetos sino a eventos y su estructura funcional es cliente – servidor; su motor de base de datos era MS Access® –un producto de la compañía Microsoft®- que presenta bastantes limitaciones en cuanto a licencias de uso ya que es un software no comercial, de manera que se hacía cada vez más urgente una actualización de la herramienta, así que bajo un riguroso estudio fue posible deducir que la mejor opción para su nueva implementación era el uso de una interfaz desarrollada bajo un lenguaje 100% orientado a objetos y que su motor de base de datos fuese ágil, libre y seguro; convirtiéndose en los pilares fundamentales de la reingeniería de este proceso.

Otro factor importante es la usabilidad de la herramienta software; aquí se tienen en cuenta aspectos como lo visual y lo adaptativo de la herramienta, el usuario final es quien debe sentirse cómodo con el trabajo que estamos desarrollando ya que es él quien lo ha de dominar la mayor parte del tiempo. 'Contar' en su codificación actual está siguiendo patrones más estilizados, con figuras e imágenes pequeñas y poco vistosas para no interferir con la funcionalidad esencial de la misma: que es la contabilidad, pero al mismo tiempo se espera explotar al máximo los recursos visuales que el entorno Visual Studio® ofrece para el desarrollo de interfaces y junto a esto su accesibilidad por teclado; la cual también sigue los estándares internos propuestos por la empresa.

Simultánea e internamente se están empleando métricas y patrones de código que permitan un resultado más preciso, ágil y con menos uso de recursos hardware; se ha tomado como apoyo a un proveedor de datos relativamente nuevo en el mercado que se conoce como NPGSQL que es una librería dinámica creada con el fin de ser implementada en cualquier interfaz sin la necesidad de tomar recursos desde el propio ordenador donde se encuentre la aplicación sino que por el contrario todo se haga desde el mismo gestor de base de datos (contrario a lo que ocurría con los antiguos ODBC); esto se hace teniendo en cuenta que la cobertura del proyecto actual no se extiende más allá de la estandarización con la que actualmente se trabaja en la empresa y se espera que se continúe con el desarrollo total del sistema. La base de datos se encuentra regularizada y diseñada en un 100%; donde cada una de sus entidades o tablas poseen su respectiva clave primaria y de igual manera han sido creadas con sus correspondientes tipos de datos siempre teniendo presente la algoritmia que requerirán a futuro dichos campos.

El entregable más significativo de este proyecto pretende ser la nueva estandarización de código fuente ya que la migración se está realizando de un lenguaje que se encontraba orientado a eventos a uno que se encuentra bajo el paradigma de la orientación a objetos y esto lo hace más moldeable, más acoplable y más descifrable para el programador; al mismo tiempo que la base de datos implementada en el gestor PostgreSQL la cual ofrece una gran variedad de opciones al momento de manipular bases de datos de gran tamaño.

1. OBJETIVOS

1.1 GENERAL:

Apoyar el proceso de reingeniería de la primera fase para la actualización del sistema contable CONTAR desarrollado y suministrado por la empresa EVACC Ltda.

1.2. ESPECÍFICOS:

- 1.2.1 Conocer los estándares de codificación, documentación y diseño utilizados por la empresa.
- 1.2.2 Conocer el estado actual del sistema precisando sus alcances, debilidades y fortalezas.
- 1.2.3 Conocer la estructura funcional y de base de datos del sistema actual.
- 1.2.4 Definir los nuevos requerimientos y alcances.
- 1.2.5 Definir la nueva estructura funcional según los nuevos requerimientos, eliminando funcionalidades ya no requeridas.
- 1.2.6 Diseñar el nuevo modelo de datos.
- 1.2.7 Adaptar las funcionalidades existentes a la nueva plataforma según la estructura funcional definida.
- 1.2.8 Apoyar la documentación del proceso según los estándares de la empresa.

2. PLANTEAMIENTO Y JUSTIFICACION DEL PROBLEMA

Un sistema de información sólido es la base para que cualquier empresa pueda consolidarse en el mercado y posicionarse en la cima, de manera que un fallo en él puede ocasionar grandes pérdidas e incluso puede llevar a un fracaso financiero y en casos más extremos la iliquidez financiera, es decir su muerte, económicamente hablando. Las columnas que funcionan como soporte vital a este sistema son las herramientas software que prestan soporte al manejo de datos e información; de manera que, si hay un problema en alguna de esas columnas, corre un gran riesgo la estabilidad de la compañía; es decir: si el software no cumple con los requisitos para los cuales fue diseñado, el funcionamiento general de la empresa se verá afectado y es ahí donde aplicamos nuestros conocimientos los integrantes del equipo de desarrollo y gestión de proyectos (donde muy cordialmente las directivas de EVACC Ltda me asignaron un lugar, el cual pertenece al área de desarrollo).

Anteriormente se habían presentado en la manipulación de la herramienta contable CONTAR¹ en su versión 1.22.47 algunos pormenores relacionados a la manipulación de registros tales como fallos en el cruce, adición, eliminación, anulación y/o edición de comprobantes, de la misma manera que se encontraban problemas de usabilidad de la interfaz y un alto grado de demora en la entrega de resultados por parte de la herramienta, simultáneamente no era posible realizar búsquedas en 'tiempo real'² a petición del usuario, todo esto en base a métricas que anteriormente las directivas del área de desarrollo de EVACC Ltda habían establecido.

De la misma manera se calificaron estándares sobre los aspectos visuales y como resultado se obtuvo que la interfaz actual de los formularios e informes no se encuentran acordes a la calidad que nuevas herramientas de desarrollo pueden ofrecer y que por lo tanto era necesaria una refactorización de los componentes con los cuales el cliente – quien, a final de cuentas, es nuestro más importante objetivo – interactuaba diariamente. Para esta nueva versión de CONTAR¹ que aquí se implementa se tiene en cuenta también la disposición de las ventanas dentro del formulario contenedor a fin de ofrecer al usuario un entorno más organizado, agradable y fácil de manejar.

¹ Herramienta software desarrollada, soportada, implementada y distribuida por EVACC Ltda de tipo contable con distribución privada y bajo entera autorización de sus respectivos propietarios.

² Se hace referencia a la terminología 'tiempo real' cuando la respuesta ofrecida por la algoritmia es tendiente a cero y no se requiere de más click para su respectiva obtención.

3. DESCRIPCION DE LA EMPRESA

Tomando como base que la modalidad del actual proyecto consiste en una práctica empresarial, a continuación se presenta una breve descripción de la firma que ha hecho posible la realización de esta labor.

La información que aquí se publica es de uso estrictamente académico y tiene como finalidad dar soporte a la documentación del presente escrito; cualquier intento de plagio o copia total o parcial dará lugar a las sanciones que las leyes legales colombiana estipulen para tal infracción.

3.1 Descripción.

Entornos Virtuales de Aprendizaje Limitada, EVACC Ltda es una empresa que busca contribuir a fomentar el uso de las Tecnologías de Información y Comunicaciones prácticamente en cualquier empresa y con un énfasis especial en instituciones educativas; desarrollándose en un mercado versátil por más de 5 años con un amplio respaldo de solidez y garantía en nuestros productos. Las áreas de acción de la compañía se centran en la capacitación de gran cantidad de software existentes en el mercado; además del desarrollo de software a la medida para sus clientes.

3.2 Misión

Contribuir a la incorporación de tecnologías de información y comunicación en las empresas mediante el desarrollo de software, el uso de software como servicio, comercialización de software y recursos tecnológicos, la implementación de soluciones tecnológicas que faciliten la ejecución de procesos y la gestión de información.

Adicionalmente, EVACC Ltda ofrece los servicios de consultoría, asesoría y capacitación a cualquier empresa.

En el sector educativo buscamos facilitar a las instituciones la incorporación de las TIC a su gestión y al proceso enseñanza aprendizaje.

3.3 Visión

Para el año 2014 seremos líderes en la implementación de las TIC como apoyo al proceso de enseñanza y aprendizaje; así consolidaremos nuestros productos software hacia la gestión de la pequeña y mediana empresa.

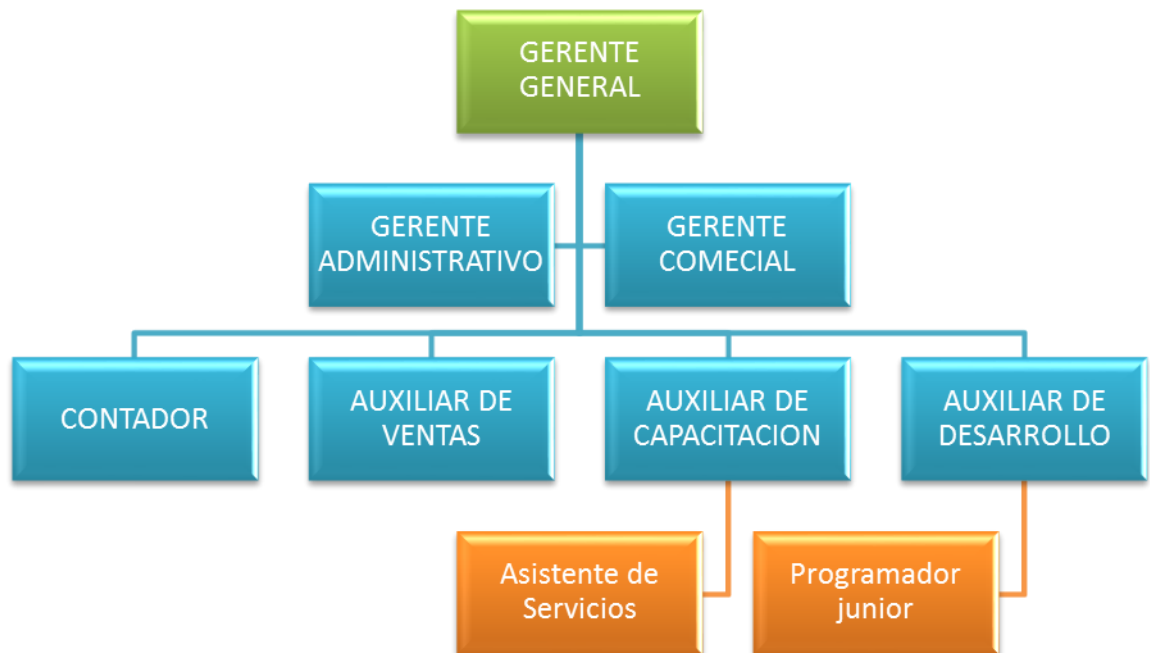
Seremos una empresa confiable en consultoría y asesoría en el uso de tecnologías informáticas.

3.4 Valores

- § Trabajar en equipo para lograr los objetivos que se propongan en un ambiente de cordialidad, confianza y respeto mutuo.
- § Facilitar la aplicación de las capacidades de cada individuo para afrontar los retos que se presenten en cumplimiento de sus objetivos.
- § Buscar soluciones creativas e innovadoras para el cumplimiento de los objetivos planteados.
- § Dar libertad para expresar sugerencias y reclamos. La administración de EVACC Ltda evalúa y responde de forma justa.
- § Promover el desarrollo de personal permanente en forma integral, tanto profesional como humana.
- § Proveer servicios oportunos de la más alta calidad que sean consistentes con las necesidades y preferencias de los clientes, logrando la satisfacción de sus necesidades.

3.5 Estructura Organizacional

Figura 1. Estructura organizacional EVACC Ltda



Fuente: El Autor

4. IMPACTO

Basándonos en la premisa del valor que actualmente tiene el tiempo para cualquier organización, es posible afirmar que perder segundos es vital para cualquier compañía, un manejo inadecuado de recursos puede generar pérdidas incalculables y un trabajador que no produzca lo suficiente puede llegar a convertirse en un problema económico para la empresa. El proceso de 're-factorizar' la herramienta software con la que trabaja el área contable implica un reto para EVACC Ltda, no solamente por los alcances del proyecto mismo sino porque se cuenta con la firme convicción de una mejora en el rendimiento de su parte contable y a su vez una reducción de los costes ocasionados por fallas en el sistema anterior o la demora en los procesos que el mismo lleva a cabo.

Será posible ver reflejados los logros de este proyecto en el momento en que la empresa realice sus consultas o transacciones con mayor fluidez, con mayor seguridad al confirmar procesos y la eficiencia al trabajar en un entorno visual más agradable; todo esto gracias a la reingeniería aplicada a la codificación que la empresa usa como estándar.

Asimismo se espera que aquellos proyectos que continúen con los objetivos de la presente actividad puedan llevar una guía de los estándares que se traen anteriormente y los nuevos que se encuentran implementando en esta labor, cambios necesarios para la culminación del cambio total de plataforma. Conjuntamente se espera facilitar la documentación necesaria para que las métricas de codificación y la estandarización visual y seguimiento de normatividades de la empresa continúen siendo aplicadas al resto de la migración.

Es de recordar que una aplicación de estas debe ser puesta en marcha en múltiples maquinas conectadas en red; por lo tanto, se resalta el valor agregado que tiene la codificación en el lenguaje VisualBasic .NET ya que su implementación en computadores cliente y el uso de algoritmos inteligentes para el acceso a la base de datos de forma simultánea brindan la confianza de una reducción significativa en la perdida de información por acceso simultaneo a dicha base. De manera que al contar con una codificación ágil es posible acceder desde cualquier punto de la red hasta el servidor donde se encuentre instalado el gestor de PostgreSQL y en un trabajo conjunto lograr el ingreso sin tener que interferir en los procesos de otros usuarios.

5. MARCO METODOLÓGICO

La cobertura de la práctica que aquí se describe abarca una mínima parte del gran trabajo que requiere EVACC Ltda en su proyecto general, de tal forma que el presente desarrollo actúa como fundamento y base para futuros colaboradores del mismo; es por esto que se requiere aún más detalle en la metodología a usar. Cuando un ideal de esta envergadura se planea, se debe hacer pensando en el futuro, es por esto que en el presente capítulo se describirán las pautas informativas y los progresos metodológicos que se han llevado a cabo durante la ejecución de la labor que en este escrito se menciona; todo con el fin de mantener como legado este documento para que sirva de soporte a las generaciones venideras que pretendan proseguir con su culminación.

Inicialmente se exponen las metodologías contempladas para su realización, de cada una de ellas se describe de forma global: sus orígenes, aplicabilidades en cualquier proyecto software, ventajas y desventajas de su uso, y seguido esto un breve resumen de lo concluido para concluir con los motivos considerados para optar por una metodología en específico.

5.1. Metodologías de desarrollo de software.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. En el desarrollo de proyectos informáticos es necesario establecer una metodología y un plan de trabajo adecuados que permitan obtener los resultados enmarcados en un cronograma y un presupuesto previamente programados. En este punto radica la importancia de definir y aplicar una de ellas en la construcción de una aplicación Web o cualquier otra aplicación basada en el computador, ya que actúan como plano de apoyo y punto de referencia para hacer más efectivo el trabajo en equipo, cada vez más común dado que las aplicaciones son cada vez de mayor envergadura y la aparición de las tecnologías web demandan además el empleo de una mayor cantidad de métodos y herramientas de seguridad.


Sin embargo la elección de la metodología adecuada debe tener en cuenta los siguientes factores:

- § El tipo de sistema a desarrollar (de Gestión, Científico, de Tiempo real, a la medida o genérico).
- § Alcance y envergadura de la aplicación.
- § Forma de trabajo del equipo de desarrollo.
- § El tipo de cliente o usuarios.
- § La volatilidad de los requisitos.

Cualquiera sea el modelo de ciclo de vida seleccionado, debe cumplir con los siguientes requisitos:

- § Detallar el procedimiento a seguir y el orden de las etapas de desarrollo.
- § Determinar los criterios que señalen la culminación de una etapa.

Actualmente, existen diferentes tipos de metodologías o procesos que buscan satisfacer el desarrollo de proyectos informáticos, llamados también modelos de ciclos de vida del producto software, entre los que se encuentran: Modelos secuenciales: cascada pura, desarrollo rápido de aplicaciones (DRA); y Modelos evolutivos: prototipado evolutivo, el modelo en espiral, programación Extrema (XP: Xtreme Programming), Proceso Unificado (RUP) y dentro del marco ágil SCRUM; los cuales poseen características específicas que buscan adaptarse a distintos proyectos. A continuación se describen brevemente:

 **DRA:** Desarrollo rápido de aplicaciones. El desarrollo Rápido de Aplicaciones (DRA) (Rapid Application Development RAD) es un modelo de proceso del desarrollo del software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto. DRA es una adaptación a “Alta velocidad” en el que se logra el desarrollo rápido utilizando un enfoque de construcción basado en componentes. Si se comprenden bien los requisitos y se limita el ámbito del proyecto, el proceso DRA permite al equipo de desarrollo crear un “sistema completamente funcional” dentro de periodos cortos de tiempo.

Cuando se utiliza principalmente para aplicaciones de sistemas de información, el enfoque DRA comprende las siguientes fases:

- § Modelado de gestión: diseñar la interrelación entre las funciones y la información generada por ellas para apoyar el proceso de gestión.
- § Modelo de datos: descripción detallada de los objetos, atributos y relaciones de la información definida en el modelado de gestión.
- § Modelado de proceso: los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos. Es la comunicación entre los objetos.
- § Generación de aplicaciones: el DRA se enfoca en la construcción de componentes reutilizables y la reutilización de módulos de otros programas.
- § Pruebas de entrega: Como el proceso DRA enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben probar los componentes nuevos y se deben ejercitar todas las interfaces a fondo.

Si el proceso de gestión puede modularse entonces la aplicación es candidato a utilizar la metodología DRA. Así pues cada una de las funciones puede ser afrontada por un equipo diferente y ser integradas en un solo conjunto.

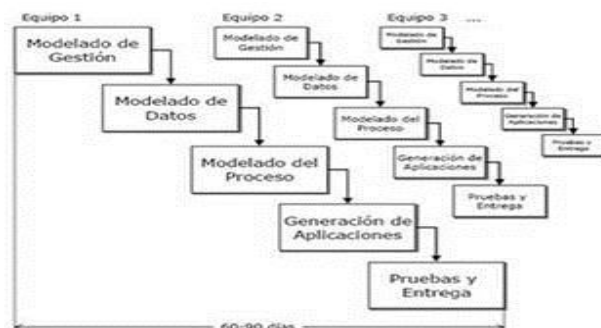
🚦 Ventajas.

- § DRA enfatiza el desarrollo de componentes de programas reutilizables. La reutilización es la piedra angular de las tecnologías de objetos, y se encuentra en el modelo de proceso de ensamblaje.
- § El tiempo de desarrollo que se puede obtener con esta metodología va de los 60 a los 90 días convirtiéndose en una alternativa adecuada para pequeños proyectos al interior de las organizaciones que requieran aplicaciones a la medida.

🚦 Desventajas

- § Para aplicaciones a gran escala por módulos el enfoque DRA tiene el inconveniente de necesitar tantos grupos de desarrollo como módulos se definan.
- § Esta metodología demanda ingenieros y clientes comprometidos con el ritmo del trabajo de esta metodología, de lo contrario el proyecto fracasará.
- § No todos los tipos de aplicaciones son apropiados para DRA. Si un sistema no se puede modular adecuadamente la construcción de los componentes necesarios para DRA será problemático.
- § DRA no es adecuado cuando los riesgos técnicos son altos. Esto ocurre cuando una nueva aplicación hace uso de tecnologías nuevas, o cuando el nuevo software requiere un alto grado de interoperabilidad con programas de computadora ya existentes.

Figura 2. Modelo de ciclo de desarrollo DRA³

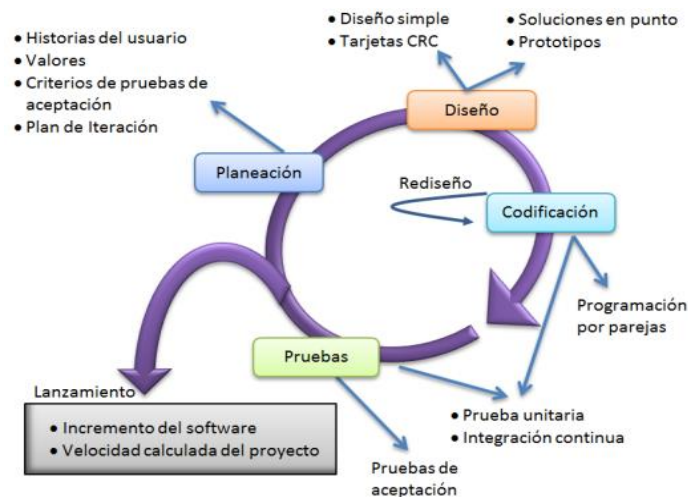


Fuente: El Autor

³ Tomado de http://3.bp.blogspot.com/_S5W0bf4uECM/TKKqWvLTW2I/AAAAAAAAAw/EF7ESijlefW/s400/Figura+4+El+modelo+de+Ciclo+de+Vida+RAD.jpg Derechos Reservados.

✚ **XP:** Extreme Programming. Metodología de desarrollo software formulada por Kent Beck en 1999, hace parte de las llamadas “Metodologías ágiles” se caracteriza por enfocarse a la adaptabilidad (aceptar los cambios) y en ser altamente utilizada en proyectos a corto plazo con poca disponibilidad de equipos y principalmente cuyo cronograma se encuentra atrasado. Una de las características más notorias de esta metodología es contar con el cliente como parte del grupo de desarrollo.

Figura 3. Fases de desarrollo – Metodología XP⁴



El ciclo de vida de XP consiste en seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

- § Fase I. Exploración: en esta fase los clientes ilustran a grandes rasgos las funciones y procesos del usuario que son relevantes para la primera entrega del producto, paralelamente el equipo de desarrollo se familiariza con las tecnologías y herramientas a utilizar y se construye un primer prototipo. Esta fase puede llevar desde unas semanas hasta un par de meses.
- § Fase II. Planificación de la Entrega: en esta fase el cliente prioriza las diferentes funciones y procesos esbozados en la etapa anterior de tal manera que los programadores realizan una estimación del esfuerzo y tiempo necesario. Se toman acuerdos sobre el contenido y alcance. Conjuntamente con el cliente se establece el cronograma de la primera entrega.

⁴ Tomado de: <http://4.bp.blogspot.com/-Y5E-AVd3iYc/UCF0rs83otI/AAAAAAAAALc/BVPIWYVi33U/s1600/5.png> Derechos reservados

- § Fase III. Iteraciones: esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores.
- § Fase IV. Producción: en esta fase se requieren pruebas adicionales, revisiones de rendimiento y reducción del tiempo de las iteraciones de tres a una semana. Se debe reevaluar el contenido del proyecto para incorporar nuevos elementos obtenidos de la refinación de esta fase. Se documentan las sugerencias para su posterior implementación (por ejemplo, en la fase de mantenimiento).
- § Fase V. Mantenimiento: se realizan tareas de soporte al cliente mientras la primera versión se encuentra en producción manteniendo el sistema en funcionamiento mientras se desarrollan nuevas iteraciones. Deben incluirse nuevos elementos al equipo de trabajo en esta etapa.
- § Fase VI. Muerte del Proyecto: se requiere que las necesidades y expectativas del cliente sean satisfechas, ampliando estas necesidades a los ámbitos de rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios a la arquitectura. La muerte del proyecto también se presenta cuando la aplicación no presenta los resultados esperados por el cliente o cuando no se cuenta con presupuesto para mantenerla.


Ventajas.

- § Se cuenta con el cliente en cada paso del desarrollo permitiendo así que se cumplan sus expectativas a la par de que se puede realimentar de las ideas del equipo y presentar alternativas que repercutan en pos de la mejora del sistema.
- § La planificación es iterativa permitiendo que la interacción con el cliente permita adaptar el proyecto a los cambios que van surgiendo dentro de la organización y que ineludiblemente repercuten en el diseño inicial.

- § Una de las características más revolucionarias y controversiales es la del trabajo en pares, el cual consiste en que dos programadores trabajen en un mismo equipo lo cual, según esta metodología, reduce los tiempos de ocio, reduce las omisiones y disminuye la negligencia.
- § Otra de las características de resaltar es el concepto de refactorización, éste es básicamente el proceso de mantener el código en buen estado, modificándolo activamente para que conserve claridad y sencillez, disminuyendo notoriamente los costos desorbitados del mantenimiento, modificación y ampliación de aplicaciones ya existentes. La refactorización no sólo sirve para mantener el código legible y sencillo: también se utiliza cuando resulta conveniente modificar código existente para hacer más fácil implementar nueva funcionalidad.

Desventajas

- § Esta metodología funciona en su máxima expresión en proyectos que demanden un grupo de no más de 20 personas, para proyectos de mayor envergadura las líneas de comunicación se vuelven demasiado complejas.
- § Trabaja mejor con interacción cara a cara. Difícil con equipos grandes.

 **DESARROLLO EVOLUTIVO:** El modelo de desarrollo evolutivo (algunas veces denominado como prototipado evolutivo) construye una serie de grandes versiones sucesivas del producto. Este modelo asume que los requerimientos no son completamente conocidos al inicio del proyecto.

El objetivo es desarrollar una implementación inicial e ir refinándola hasta conseguir el sistema adecuado incluyendo en ésta solo los requerimientos que se encuentren bien comprendidos. Esta primera implementación se les entrega a los usuarios, los cuales lo usan y proveen retroalimentación al equipo de trabajo. El sistema evoluciona según las nuevas propuestas del cliente.

Este tipo de metodología es adecuado para:

- § Para sistemas interactivos pequeños o de tamaño mediano.
- § Para partes de sistemas grandes (Ej. la interfaz de usuario).
- § Para sistemas con vida corta.

Figura 5. Modelo evolutivo de desarrollo de software ⁵



Las fases de esta metodología son:

- § Análisis de requisitos del sistema.
- § Análisis de requisitos del software.
- § Diseño, desarrollo e implementación del prototipo.
- § Prueba del prototipo.
- § Refinamiento iterativo del prototipo.
- § Refinamiento de las especificaciones del prototipo.
- § Diseño e implementación del sistema final.
- § Explotación (u operación) y mantenimiento.

🚩 **Ventajas.**

- § La especificación se desarrolla de forma creciente.
- § El diseño y construcción del prototipo permite evaluar hasta qué punto se han comprendido realmente las necesidades y requisitos del cliente.
- § Si al evaluar el primer prototipo se encuentra que no se escogieron bien las herramientas o arquitectura del proyecto se puede tomar correcciones tempranamente.

🚩 **Desventajas.**

- § Hay que documentar cada versión del sistema
- § Los sistemas tienen una estructura deficiente
- § Se requieren herramientas y técnicas especiales (p.e. conocimientos en lenguajes para el prototipado rápidos)

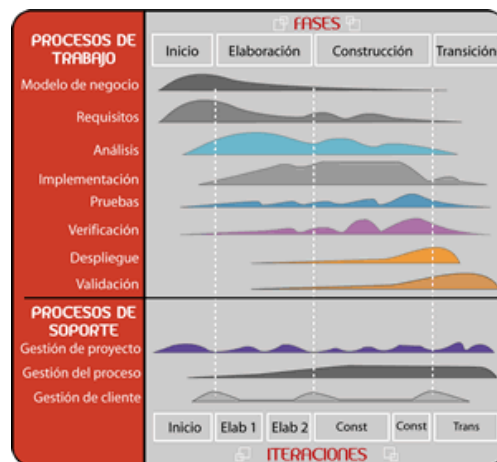
⁵ Tomado de : <http://3.bp.blogspot.com/-xk7JyluoBAw/Talui5F0YCI/AAAAAAAAAC8/WMIbjRb4iC4/s1600/evolutivo01.jpg>

✚ **PROCESO UNIFICADO** (Rational Process Unified). El proceso unificado está basado en componentes y se sostiene sobre tres ideas básicas: casos de uso, arquitectura y desarrollo iterativo e incremental. Para hacer que estas ideas funcionen, se necesita un proceso, que tenga en cuenta ciclos, fases, flujos de trabajo, gestión del riesgo, control de calidad, gestión de proyecto y control de la configuración.

El Proceso Unificado divide el proceso de desarrollo en ciclos, donde se obtiene una nueva versión del producto al final de cada ciclo. Cada ciclo se divide en cuatro Fases: Inicio, Elaboración, Construcción, y Transición. Cada una de estas fases concluye con un hito bien definido donde deben tomarse decisiones respecto al proyecto como la reestructuración del cronograma de trabajo. Cada una de estas fases se divide a su vez en iteraciones.

Cada iteración sigue la estructura de un pequeño ciclo de vida en cascada, pasando a través de los cinco flujos de trabajo fundamentales: requisitos, análisis, diseño, implementación y prueba. En la iteración también incluye la planificación que precede a los flujos de trabajo y la evaluación que va detrás de ellos.

Figura 7. Modelo de Proceso unificado ⁶



✚ **Ventajas.**

- § El proceso unificado es un marco de trabajo genérico.
- § Reduce el costo de riesgos.
- § Presenta signos visibles de progreso rápidamente.
- § Proporciona funcionalidad al cliente antes de entregar todo el proyecto.
- § Se entregan prestaciones importantes al inicio.
- § Proporciona suficiente control de gestión.

⁶ Tomado de: http://2.bp.blogspot.com/_mV6AdT5Kv-0/TPcNgUekPn/AAAAAAAAABA/ExjU9mRhf6E/s1600/Metodologia.gif

✚ Desventajas.

- § Requiere mayor planificación.
- § Requiere mayor control de gestión.
- § Se necesita definir su arquitectura antes de dar inicio a las iteraciones de construcción.

✚ **SCRUM:** asume que el proceso de desarrollo de software es complicado e impredecible y lo trata como una caja negra en lugar de un proceso teórico totalmente definido. Esta es una de las mayores diferencias entre Scrum y otras metodologías de desarrollo de software.

SCRUM es un “proceso ágil” que nos permite enfocarnos en la generación de valor en el mínimo tiempo; además nos permite tener un control continuo sobre el estado actual de nuestro software y junto a esto el cliente puede establecer las prioridades ya que el equipo se auto-organiza para determinar la mejor forma de entregar los requerimientos con más prioridad.

Cada Semana acaba con software entregable que se puede publicar o bien seguir mejorando.

Scrum se aplica bajo una filosofía de trabajo que no sólo implica al desarrollador sino también al cliente dando prioridad a los individuos y las interacciones sobre los procesos y las tareas, prefiriendo el software funcional sobre la excesiva documentación, promocionando la colaboración con el cliente en lugar de la negociación de contratos y sobre todo teniendo capacidad de respuesta sobre los cambios en lugar de seguir estrictamente una planificación.

✚ Creación de equipos.

Scrum considera que un equipo de desarrollo debe funcionar como un equipo deportivo, cada miembro del equipo trabaja de forma independiente, pero hacia el mismo objetivo. Scrum sugiere que un equipo tiene un máximo de 6 a 7 miembros. El facilitador de equipo se llama el "Scrum master". Su trabajo es implementar y gestionar el proceso Scrum en el proyecto. El equipo de Scrum en su conjunto define las prácticas, las reuniones y la terminología de SCRUM, y por su parte el Scrum Master asegura el cumplimiento de estas "normas". El Scrum master desempeña un papel facilitador y su autoridad es sobre todo indirecta: centra la mayor parte de su tiempo en la gestión de interferencias externas y la solución de los obstáculos externos o "bloqueantes" que no pueden ser resueltos por el equipo Scrum.

Creación de documentos

Se trabaja en base a 3 tipos de documentos:

- § Producto: Actúa como un repositorio de los requisitos para una publicación específica. Se suelen contemplar los requisitos de alto nivel.
- § Lanzamiento: Son los requerimientos obtenidos del documento anterior pero ya identificados y priorizados para la próxima versión.
- § Sprint 'Backlog': Es un documento detallado donde se describe la forma en la que el equipo va a implementar los requisitos durante el siguiente sprint. Las tareas se dividen en horas donde ninguna tarea supera las 16 horas de duración. Las tareas en el sprint nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca oportuno.

La segmentación del proyecto

Todo el proyecto se divide en períodos de tiempo con una duración máxima de 4 semanas. Un período se denomina Sprint y cada equipo recibe una cartera de pedidos a ejecutar en un Sprint determinado.

Reuniones Scrum

Durante cada sprint, el equipo lleva a cabo varias reuniones Scrum tanto como sea posible; durante el día de trabajo y con una duración de no más de 30 minutos. La reunión tiene como objetivo la interacción continua entre el Scrum master y el resto del equipo, donde la conversación se centra en las respuestas a las siguientes tres preguntas:

- ¿Qué has hecho desde la última reunión de Scrum?
- ¿Hay algo que haya impedido tu trabajo?
- ¿Qué planeas hacer entre ahora y la próxima reunión Scrum?

En esencia, cada uno de los miembros del equipo se limita a responder cada una de estas preguntas y por tanto, las responsabilidades del Scrum master en base a esto son:

- § Planificar reuniones justo después de cada sesión de trabajo.
- § Toma de decisiones inmediatas en caso de ser necesario eliminar obstáculos que impidan el progreso del proyecto.
- § Señalar los obstáculos que deben ser resueltos externamente a la reunión a fin de no extenderse demasiado innecesariamente.

Fases

El proceso de desarrollo de esta metodología se basa teóricamente en una evolución continua, todas las fases están estructuradas de la misma forma pero sus resultados individuales cambian con cada entrega, de manera que a continuación se describirá con detalle el proceso del cual consta cada una.

Se toma como base que dentro de cada sprint se lleva a cabo el desarrollo del software, de manera que es posible sub-dividir cada etapa en las siguientes actividades: *Elaborar, Integrar, Revisar, Ajustar*.

Aunque no se lleva una secuencia estricta, es decir: pueden existir productos que requieran primero de un desarrollo, luego ser integrado para ser revisado, existen otros que solo requieran ser revisados y/o integrados, etc; siempre se obtendrá un entregable de cada etapa.

En cada una de las entregas de los sprint es necesaria una revisión del mismo; durante esta revisión, el software desarrollado en el Sprint anterior se chequea a fin de hallar imperfectos y si es necesario añadirsele nuevos ítems. El grupo de revisores pueden ser: las partes interesadas del proyecto, gestores, desarrolladores, en ocasiones los clientes, ventas e incluso marketing.

Este procedimiento se repite hasta que el producto se considera listo para su distribución por los participantes en el proyecto y cuando se ve terminado se procede a realizar un cierre final donde dicho producto se prepara para el lanzamiento y la distribución final.

Cierre

Corresponde a las actividades de revisiones finales, donde se verifica que el proyecto se encuentre terminado de acuerdo al plan desarrollado con anterioridad. Cuando se le ha realizado un depurado minucioso al software, y se confirma que la meta se ha logrado, el área de marketing procede a promocionar el producto terminado; de manera que al concluir esta actividad se dará por finalizado el proyecto y se asumirá como cerrado.

Desventajas.

- § En general, dificultad de aplicación en grandes proyectos.
- § Se requiere de un director altamente calificado en la dirección de proyectos, experto en la metodología que monitorice su cumplimiento.
- § Parte del paradigma consiste en usar el método “tal cual”, evitando adaptarlo a la empresa (supuestamente esto saca a la luz problemas en

la metodología de desarrollo existente). Claro, si por otro lado arrancamos con el método ya adaptado a la empresa, es posible que no estemos aprovechando las ventajas de la metodología ágil.

- § Plantea un problema si el desarrollo está restringido por una fecha de entrega y un precio de entrega cerrados por contrato
- § Presupone que los requerimientos cambian, pero no de forma que el cliente acepte un diseño funcional/técnico.
- § Presupone que el equipo está muy formado y motivado
- § Presupone que el cliente está muy involucrado en el desarrollo, participa de forma activa y continua, y revisa frecuentemente el avance de la funcionalidad conforme salen a la luz los sprints. Esto sin embargo no parece producirse en la mayoría de nuestros proyectos: el cliente participa, pero no hasta el punto de dedicar tiempo y recursos para revisar pequeños avances en el desarrollo.

- § Presupone que el cliente no exige ni necesita toda la documentación que manejan actualmente las empresas y que las diversas normativas internacionales requieren.

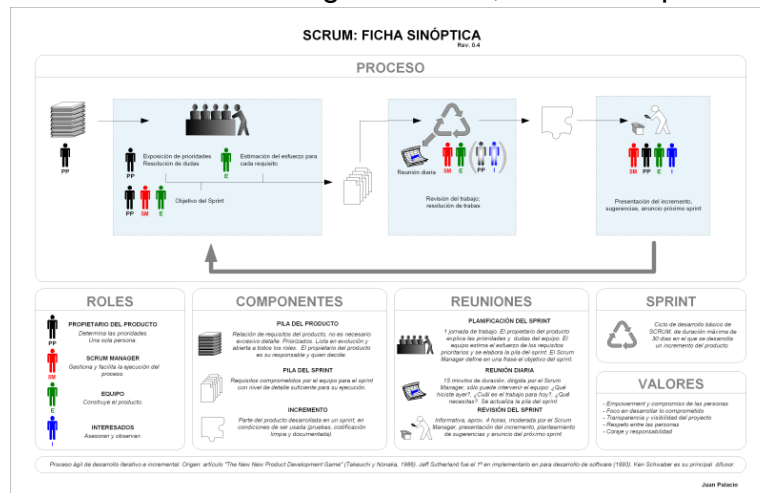
Ventajas.

- § Aumento de velocidad durante el desarrollo: Scrum tiene como objetivo reducir el tiempo de desarrollo pues sólo trabaja sobre las características que proporcionan un valor de negocio obvio para el usuario final.
- § Flexibilidad: La metodología anticipa los cambios durante el desarrollo debido a que el proceso de desarrollo es tan repetitivo, lo que permite hacer grandes cambios estructurales durante el desarrollo. Scrum también adopta un enfoque muy orientado a objetos para la construcción de software.
- § Desarrollador Independiente: Los desarrolladores son mucho más independientes en la metodología y son libres de tomar más decisiones acerca de un determinado proyecto; esto puede ayudar a aumentar la productividad de los equipos de desarrollo de software. Además, el enfoque en la iteración y los hitos disminuyen la cantidad de planificación logrando que los desarrolladores puedan pasar más tiempo de codificación.
- § Menos Gestión: Los gerentes tienen la responsabilidad y mucho menos en el enfoque de Scrum para el desarrollo de software. Las empresas pueden ahorrar dinero en la gestión de los administradores si el enfoque

es seguido por esta metodología. Esto dará lugar a un producto más económico, pues disminuye conjuntamente el tiempo de desarrollo.

Además, con la metodología Scrum es más fácil comprobar el progreso del proyecto ya que las reuniones frecuentes son parte de la misma.

Figura 8. Detalle Gráfico Metodología SCRUM, Ficha Sinóptica ⁷



5.2 Selección de la metodología a implementar.

Con base en el entorno donde se desarrolló la práctica y los procesos de gestión que sigue internamente EVACC Ltda, luego de hacer un minucioso estudio se determinó que la metodología a implementar para la ejecución del proyecto fue SCRUM por tener características adaptables a los requerimientos que se tenían. Siendo una metodología ágil, facilita la evolución del prototipado de la herramienta software y ofrece la documentación necesaria para tener como guía en el proceso de documentación del mismo. De la misma forma se pretende trabajar con una metodología que sea progresiva y que evite al máximo la posibilidad de riesgos que puedan presentarse pero que a su vez permita prever lo que está por venir; teniendo en cuenta que la cobertura del proyecto actual abarca únicamente la documentación estándar de código fuente y el establecimiento de algunos estándares para futuras inclusiones y/o refactorizaciones del mismo.

El grupo de trabajo jugó también un papel fundamental al momento de esta selección ya que se cuenta con poca cantidad de personal pero con una extensa experiencia en la gestión de proyectos; de tal forma que la metodología se adecua a la forma de trabajo con la que se cuenta en la compañía y ofrece pautas para el correcto trabajo en equipo.

⁷ Tomado de: <http://www.omitsis.com/scrum-como-metodologia-de-desarrollo>

6 MARCO TEÓRICO DE REFERENCIA

La competitividad actual del mercado obliga a una empresa a exigirse aún más, por lo tanto es necesaria la actualización de su estructura informática, un sistema de información sólido es la base fundamental para que la organización cumpla con sus objetivos misionales y por ende los visionales; esto se hace posible con documentación e información relacionada con: sistemas de información, Bases de datos, Lenguajes de programación, sistemas de contabilidad en forma global.

De una forma más específica es posible soportar el proceso en base a información relacionada a POSTGRESQL, Visual Basic .NET, librerías dinámicas y conceptos generales sobre contabilidad colombiana.

Los temas anteriormente mencionados comprenden un marco conceptual bastante amplio pero esencial y necesario al momento de llevar a cabo este proyecto.

6.1 Fundamento teórico general.

6.1.1 Sistemas de Información

La palabra “Sistema” se encuentra en todas partes, lo cual ha vuelto el concepto tan amplio como su misma utilidad y aplicabilidad; Su esencia está en el tratamiento de la información; a fin de aclarar conceptos me permito relacionar algunos enunciados que expertos en el tema exponen en sus obras: “Un sistema de información es un conjunto de elementos interrelacionados con el propósito de prestar atención a las demandas de información de una organización, para elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones”.⁸

“Conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Teniendo muy en cuenta el equipo computacional necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema.”⁹

Lo anterior ofrece un panorama de la extensión que puede llegar a cubrir un Sistema de información, lo cual tomaremos como base para indicar que los alcances de este proyecto pueden llegar mucho más allá de lo que se plantea. Así es que al modificar la herramienta contable estamos aportando en una pequeña parte para que el sistema de información de la empresa que adquiere el software pueda contar con una herramienta útil al momento de respaldar parte de su información y mejore los procesos que en él intervienen.

⁸ Juan Peña Llopis, 2006

⁹ J Cesqar Moraga Peralta 2008

Para abordar un proyecto del enfoque y alcance del que es objeto el presente documento, es fundamental contar con la participación conceptual de los Sistemas de Información (SI), entendidos en el contexto, como un conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa y que van en la misma dirección con los objetivos y la misión de esta, abarcando además del software, el hardware necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema y se benefician de este. En esta línea de ideas un Sistema de Información (SI) es un conjunto de componentes interrelacionados para recolectar (entrada), manipular (procesamiento), disseminar datos e información (salida) y para proveer un mecanismo de retroalimentación útil en el cumplimiento de un objetivo.

Un sistema de información realiza entonces cuatro actividades básicas: entrada, procesamiento, salida de información y retroalimentación.

- ✚ **ENTRADA:** En sistemas de información, es la actividad que consiste en recopilar y capturar datos primarios. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos o información que provienen o son tomados de otros sistemas o módulos.
- ✚ **PROCESAMIENTO:** En sistemas de información, supone la conversión o transformación de datos en salidas útiles. Esto puede implicar ejecutar cálculos, realizar comparaciones y adoptar acciones alternas, y el almacenamiento de datos para su uso posterior.
- ✚ **SALIDA:** En sistemas de información, la salida implica producir la información útil, por lo general, en forma de documentos y/o reportes. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo.
- ✚ **RETROALIMENTACIÓN:** En sistemas de información, es la salida que se utiliza para efectuar cambios en actividades de entrada y procesamiento.

Acorde a esta división del trabajo algunos autores establecen, de forma análoga, una categorización de los SI, tomando como criterio los niveles de jerarquía organizacional. Así, Cohen¹⁰ y Asín¹¹ (2000: 8-12) plantean la tipología de la siguiente manera:

¹⁰ Más información en: http://highered.mcgraw-hill.com/sites/9701026586/information_center_view0/enrique_asin_lares.html

Sistemas transaccionales: enfocados a los procesos operacionales (donde se crea valor) con el objetivo de automatizarlos; suelen ser los primeros a implantar en una organización conformándose como la base de la información productiva utilizada posteriormente y, normalmente, se adquieren como paquetes de aplicación disponibles en el mercado.

Sistemas de apoyo a las decisiones: constituyen la plataforma de información e intentan integrar los sistemas transaccionales más relevantes de la empresa, y generar información para ser utilizada por los mandos intermedios y la alta administración (toma de decisiones); caracterizados por menos requerimientos de datos, significativo procesamiento interno e información de salida (2). Dada su función como apoyo a las decisiones, que involucra el criterio de los gerentes para el uso de la información generada, no se presentan evidencias únicas de sus retribuciones y es difícil justificar la inversión en este tipo de sistemas. Según los autores suelen ser “interactivos y amigables, con altos estándares de diseño gráfico y visual”; no coincidiendo con ellos al afirmar que pueden ser desarrollados por el decisor (usuario final), sin participación del personal de informática (3).

Sistemas estratégicos: relativos a la interacción con el entorno para responder a los cambios, con el propósito de buscar ventajas competitivas y establecer relaciones con proveedores, distribuidores y clientes externos o internos (crear relaciones). Suelen ser desarrollados por personal de la organización.

En consecuencia, la asociación entre los diversos niveles de una organización está delimitada por las funciones que cumplen cada uno de ellos; así, el nivel operativo consolida la información para el nivel gerencial, siendo ésta utilizada para decidir sobre las alternativas y apoyar los posibles cursos de acción a largo plazo, que son presentados al nivel estratégico, con el fin de tomar la decisión más adecuada. Sin embargo, la dinámica gerencial impide la rigurosidad en el cumplimiento de dichas funciones.

Aun cuando, para algunos autores, los principales elementos estructurales en la anatomía de un sistema de información para la administración son: captación de datos, procesamiento, análisis, toma de decisiones, control y retroalimentación; es indudable que la mayoría de los sistemas no están necesariamente integrados, aunque sí interrelacionados, bien sea en forma directa como respuesta a los requerimientos de diseño y desarrollo, o en forma indirecta debido a la comunicación formal o informal que se produce en torno a ellos. Esto representa que no siempre los responsables del uso de la información reconocen las potencialidades de los sistemas y su contribución al

¹¹ Más información en: http://highered.mcgraw-hill.com/sites/9701026586/information_center_view0/daniel_cohen_karen.html

control de gestión, situación característica del nivel medio de la organización como gestores en la toma de decisiones.

La segmentación del trabajo planteada por diversos gurús en el tema en su dimensión de toma de decisiones, incluyen tres tipos de trabajadores: a) decisores, responsables como último recurso de tomar la decisión, b) participantes, quienes se involucran en la toma de decisión y c) ejecutores, responsables de aplicar la decisión. Tomando en cuenta la tipología basada en niveles de gestión, se evidencia que en los sistemas de información de apoyo a las decisiones sus usuarios se convierten en decisores en su nivel y participantes en las decisiones del nivel superior. Estos tipos de sistemas se conciben como aquellos que permiten realizar la gestión de la empresa, tomando en cuenta la asignación y control de los recursos disponibles para lograr los objetivos establecidos.

No obstante, estas y otras clasificaciones, la mayoría formuladas desde la perspectiva empresarial, normalmente, omiten la información destinada a entes externos, la cual se considera fundamental para las organizaciones de servicios públicos promocionando la participación de sus beneficiarios.

6.1.2 Plataforma .NET

Para conocer el funcionamiento de la plataforma .NET es necesario conocer su filosofía, sus bases y orígenes, también es necesario conocer su motor, el Framework .NET, el cual es el pilar de las herramientas de desarrollo.

La iniciativa de .NET surgió como la convergencia de varios objetivos distintos pero estrechamente unidos, el más importante de los cuales es necesidad de superar las limitaciones del modelo de programación COM y de encontrar un paradigma común de programación para las aplicaciones Cliente-Servidor y las de Internet.

A pesar del gran avance que ha experimentado la industria del software en los últimos años en términos de herramientas de programación disponibles, tales como lenguajes de desarrollo rápido de aplicaciones (RAD), depuradores y generadores de código; escribir aplicaciones robustas para el ámbito empresarial sigue siendo un trabajo complicado, para explicar esto se puede decir que la plataforma de Windows ha evolucionado caóticamente en los últimos años. Se ha tenido, por ejemplo, hasta tres modelos de programación distintos para producir aplicaciones para el manejo de gráficos (GDI, DirectX y OpenGL), siendo cada modelo completamente distinto a los demás. También se han producido diferentes modelos de programación para acceder a las bases de datos (DAO, RDO, ADO), siendo cada uno de ellos incompatibles con los demás.

Además el propio Windows se presentaba en diferentes formas (Windows 95, 98, Me, NT, 2000, CE, XP y Vista). No todas las versiones son compatibles con todas las funciones, y el resultado neto es que los diseñadores ignoraban, con frecuencia, determinadas funciones de Windows con la intención de que sus aplicaciones pudieran ser utilizadas por la mayoría de los usuarios.

La seguridad es otra área de preocupación que afecta en gran medida a los diseñadores de las plataformas Windows, en el sentido que deben optar por diferentes niveles de seguridad.

- § **Defectos COM**, el Modelo de Objetos de Componentes (COM), ha demostrado ser lo suficientemente versátil como para convertirse en la infraestructura de numerosas aplicaciones distribuidas y empresariales, pero al mismo tiempo, ha demostrado ser excesivamente complejo y peor aún, las aplicaciones COM son inherentemente frágiles porque dependen fuertemente de la información almacenada en el Registro del sistema, que tiene una merecida reputación de poder ser corrompido con facilidad.

Las aplicaciones COM padecen otro serio problema, el control de versiones; cada versión más moderna de un componente COM sobrescribe las versiones más antiguas, por lo que, por ejemplo no podrá tener varias versiones de ADO en el mismo equipo. En la práctica aplicaciones que utilizaban versiones anteriores pueden dejar de repente de funcionar debido a las pequeñas incompatibilidades entre estas versiones.

- § **Interoperabilidad entre lenguajes**, la plataforma Windows dificulta enormemente la integración de piezas de código escritas en diferentes lenguajes, porque cada lenguaje cuenta con sus propios convenios de llamada y limitaciones.

Como integrar diferentes lenguajes resulta bastante difícil, muchos diseñadores de software prefieren utilizar un único lenguaje, perdiéndose así las ventajas particulares que cada lenguaje puede ofrecer. Por ejemplo, Visual Basic resulta bastante más productivo para aplicaciones empresariales y de base de datos y, apresa de todo, muchos diseñadores prefieren seguir utilizando C++ debido a su mayor potencia y flexibilidad. Si la integración de lenguajes fuera más sencilla, se podrían desarrollar las distintas partes de una aplicación de gran tamaño utilizando el lenguaje que fuera más apropiado para cada una de ellas y de esta forma el desarrollo de software sería más barato en términos de tiempo y dinero.

Debido a estos problemas que presentan las herramientas de desarrollo de software actuales, la solución que ofrece Microsoft.NET es la siguiente:

§ **Potenciar las plataformas Windows**, Microsoft.NET ofrece una visión orientada a objeto del sistema operativo Windows e incluye cientos de clases que encapsulan los objetos más importantes del núcleo de Windows. Por ejemplo, una parte de .NET denominada GDI+ contiene todos los objetos que muestran gráficos en una ventana. Dependiendo de la versión específica de Windows, estos objetos pueden utilizar funciones GDI, DirectX o incluso OpenGL, pero desde el punto de vista del diseñador lo importante es que solo existe un modelo de programación.

La seguridad está construida directamente sobre .NET por lo que no tendrá que preocuparse demasiado del modelo de seguridad del sistema operativo. La seguridad de .NET es superior a la seguridad de Windows y el administrador puede, entre otras cosas, garantizar o revocar el derecho a acceder al Registro o al sistema de archivos de determinadas aplicaciones. Este modelo de seguridad es independiente de la versión específica de Windows sobre la que se ejecute la aplicación.

Las aplicaciones y componentes .NET son también inherentemente más seguras que los componentes COM y las aplicaciones Windows “al viejo estilo”. Por ejemplo, los componentes .NET con una firma digital y utilizar una tecnología como Authenticode para permitir que los usuarios sepan quién es el autor de la aplicación.

§ **.NET como mejora de COM**, se puede pensar en .NET como la siguiente generación de COM, ya que estos dos modelos de programación le permiten resolver más o menos los mismos problemas. Sin embargo entre bastidores, son completamente distintos y .NET es superior a COM en muchos aspectos.

En primer lugar, una aplicación .NET puede constar de uno o más Ensamblados. Cada ensamblado suele ser un archivo ejecutable .EXE o DLL. Los ensamblados también pueden incluir otros archivos, tales como DLL adicionales, o .HTML, .gif o cualquier otro archivo no ejecutable. Los ensamblados son una unidad de control de versiones y de implementación en el sentido de que todos los archivos contenidos en el ensamblado tienen el mismo número de versión y no podrá instalar un ensamblado cuyos archivos estén marcados con diferentes números de versiones. Normalmente, una aplicación utiliza varios ensamblados externos, incluyendo aquellos que pertenecen al propio .NET Framework.

.NET no utiliza el Registro para almacenar cualquier información sobre ensamblados (aunque sigue utilizando el Registro para almacenar información sobre el propio .NET Framework). En general toda la información relacionada

con una aplicación y con los componentes que utiliza se encuentra almacenada en los archivos de configuración contenidos en el directorio principal de la aplicación. Los archivos de configuración utilizan la sintaxis XML para almacenar datos de forma jerárquica y, por tanto, son más flexibles que los viejos archivos .INI.

La forma en que .NET resuelve el problema de versiones es el siguiente: existen dos tipos de componentes .NET, los privados y los compartidos. Los componentes privados se encuentran almacenados en el directorio principal de la aplicación (o en uno de sus subdirectorios) y no son visibles para otras aplicaciones. De esta forma, cada aplicación utiliza sus propios componentes privados y las distintas aplicaciones pueden utilizar diferentes versiones de estos componentes porque se encuentran almacenados en diferentes directorios.

Por otro lado, los componentes compartidos son visibles a todas las aplicaciones .NET. Se almacenan normalmente en un almacén central denominado Cache de ensamblados global (GAC) localizado en el directorio C:/Windows/Assembly. Cada componente contenido en el GAC se encuentra almacenado en un directorio independiente y en el mismo equipo pueden coexistir diferentes versiones del mismo componente compartido. Cada aplicación compilada con determinada versión y de un determinado componente continuara funcionando correctamente incluso si el usuario instala una versión más moderna (o más antigua) del mismo componente.

§ **Todos los lenguajes de .NET son iguales**, .NET desplaza la mayoría de su funcionalidad desde el lenguaje de programación al propio .NET Framework. Por ejemplo, .NET Framework dispone de clases para abrir, leer y escribir archivos de texto, de la misma manera, también tiene clases para crear ventanas y controles, por lo que no existe ningún motivo para que estas funciones estén incluidas en los propios lenguajes de programación. Entonces todos los lenguajes de programación .NET pueden utilizar estas clases y, por tanto, todos los lenguajes tendrán la misma capacidad de crear aplicaciones con una muy buena interfaz.

.NET Framework

El .NET Framework (traducido como “Marco de Trabajo”) es el componente fundamental de la plataforma Microsoft .NET, necesario tanto para poder desarrollar aplicaciones como para poder ejecutarlas luego en entornos de prueba o producción. El .NET Framework tiene tres variantes principales, todas descargables gratuitamente desde internet.

§ **.NET Framework Redistributable Package:** este es el mínimo componente de la plataforma .NET que se necesita para poder ejecutar aplicaciones. Normalmente esta es la variante que se instala en los entornos productivos, una vez que el desarrollo y las pruebas de la aplicación han finalizado.

Está compuesto por:

- § El entorno de ejecución de la plataforma .NET
- § Las bibliotecas de funcionalidad reutilizable

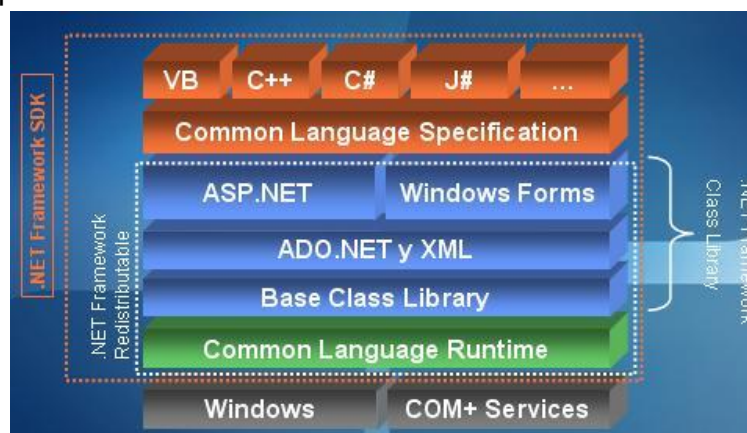
§ **.NET Framework SDK:** Esta versión contiene herramientas de desarrollo de línea de comandos (compiladores, depuradores, entre otros), documentación de referencia, ejemplos y manuales para desarrolladores de aplicaciones. Normalmente esta variante se instala en los entornos de desarrollo de aplicaciones, y es más útil a los programadores que a los usuarios finales. Para poder instalar la versión SDK (Software Development Kit) es necesario instalar previamente el Redistributable Package.

§ **.NET Compact Framework:** esta es una versión reducida del .NET Framework Redistributable, especialmente pensada para ser instalada en dispositivos móviles como Pocket PC y Smartphone.

El .NET Framework puede ser instalado en cualquier sistema operativo de la familia Windows superior a Windows 98.

🚦 ARQUITECTURA DEL .NET FRAMEWORK

Figura 9. Arquitectura del .NET Framework



Fuente: El Autor

En la figura se pueden apreciar las distintas partes que componen al .NET Framework, incluidas el entorno de ejecución de aplicaciones (CLR, en verde), el conjunto de bibliotecas de funcionalidad reutilizable (.NET Framework Class Library, en azul) y los compiladores y herramientas de desarrollo para los lenguajes .NET (en rojo). Todos estos componentes se ubican por encima de la familia de sistemas operativos Windows.

Dentro del conjunto de la .NET Framework Class Library se distinguen 4 subcomponentes principales:

- § **La Base Class Library** (BCL-Biblioteca de Clases Base), que contiene la funcionalidad más comúnmente utilizada para el desarrollo de todo tipo de aplicaciones. Algunos ejemplos de la funcionalidad provista por la BCL son el manejo de colecciones, cadenas de texto, entrada/salida, threading o hilos, operaciones matemáticas y dibujos 2D.
- § **ADO.NET**, que contiene un conjunto de clases que permiten interactuar con bases de datos relacionales y documentos XML como repositorios de información persistente.
- § **ASP.NET**, que constituye la tecnología dentro del .NET Framework para construir aplicaciones con interfaz de usuario Web (es decir, aplicaciones cuya lógica se encuentra centralizada en uno o varios servidores y que los clientes pueden acceder usando un browser o navegador mediante una serie de protocolos y estándares como HTTP y HTML).
- § **Windows Forms** (o simplemente WinForms), que constituye la tecnología dentro del .NET Framework que permite crear aplicaciones con interfaz de usuario basada en formularios y ventanas Windows de funcionalidad rica y que se ejecutan directamente en los clientes.
- § **Common Language Runtime (CLR)**: actúa como un motor de ejecución de aplicaciones y componentes manejados; tiene las siguientes características:
 - Compilación Just-In-Time (JIT), se encarga de compilar las aplicaciones .NET a código de maquina nativo para el sistema operativo y la plataforma de hardware en la que se está ejecutando.
 - Gestión automática de memoria (Garbage Collector).
 - Gestión de errores consistente (Excepciones).
 - Ejecución basada en componentes (Assemblies), todas las aplicaciones .NET son empaquetadas en componentes reutilizables denominados genéricamente Asemblies.
 - Gestión de Seguridad.
 - Multithreading, provee un entorno de ejecución multi-hilos por sobre las capacidades del Sistema Operativo.

§ **Base Class Library (BCL):** es la Biblioteca de Clases del .NET Framework, que provee cientos de tipos básicos (clases e interfaces principalmente) orientados a objetos, extensibles mediante herencia. Los tipos están organizados en jerarquías lógicas de nombres, denominadas NAMESPACEs. Los tipos son independientes del lenguaje de desarrollo.

✚ **FUNCIONAMIENTO DEL .NET FRAMEWORK:** Cuando se crea una aplicación Windows en algún lenguaje compatible con la plataforma .NET, se puede utilizar cualquiera de los servicios que la biblioteca de clases de .NET provee. Cuando se compila una aplicación, se crea un código intermedio llamado MSIL. Este código es independiente de la plataforma de hardware. Una vez compilado, el ejecutor de lenguaje común administra la ejecución de la aplicación.

Figura 10. Modelo de ejecución del CLR



Fuente: El Autor

Uno de los subsistemas del Common Language Runtime se llama compilación JIT, que transforma el código intermedio MSIL en código de máquina en el sistema donde la aplicación se va a ejecutar. Esta compilación a lenguaje de máquina se hace en el momento de ejecución del código. La aplicación sin embargo puede interactuar con otras aplicaciones .NET y servicios independientemente del lenguaje en que fueron desarrollados.

6.1.3 Herramientas de Desarrollo - Visual Studio .NET

Visual Studio es la herramienta de desarrollo por excelencia de la plataforma .NET, siendo una herramienta única que permite desarrollar cualquier tipo de aplicación (Web, Windows, de Consola, para dispositivos móviles, para Microsoft Office, de Bases de Datos y mas) en cualquiera de los lenguajes .NET provistos por Microsoft (C#, VB.NET, C++.NET y J#).

Visual Studio .NET es el entorno RAD (“Rapid Application Development” o Desarrollo Rápido de Aplicaciones), que permite utilizar .NET Framework para desarrollar las aplicaciones de una forma rápida y visual, incluyendo características de desarrollo como el IntelliSense. Visual Studio .NET utiliza .NET Framework.

Todas las aplicaciones que se realicen con .NET, serán desarrollos orientados a objetos. Este cambio es especialmente problemático para los desarrolladores que estaban acostumbrados a trabajar con Visual Basic 6 sin tener conocimientos sobre la orientación a objetos. La problemática llega porque será necesario cambiar el esquema de trabajo que se llevaba a cabo cuando se trabajaba con Visual Basic.

Dentro del marco o infraestructura de desarrollo .NET (*.NET Framework*) conviven diferentes lenguajes de desarrollo; sin embargo, todos comparten una serie de características que son idénticas para cada uno de ellos. La más importante es que comparten el mismo entorno de trabajo, .NET Framework contiene todo lo necesario para programar, compilar y ejecutar las aplicaciones.

VISUAL BASIC .NET

Es la tecnología que permite hacer aplicaciones Cliente-Servidor basado en interfaces de usuario graficas GUI. Visual Basic .NET proporciona las siguientes características:

- § Visual Basic .NET es ahora un lenguaje Orientado a Objetos, con cualidades similares a C++, conservando su especialidad: la manipulación y creación de componentes.
- § Podemos crear clases, derivándolas de otras así estén escritas en Visual Basic .NET o en otros lenguajes .NET como el C#
- § También es un lenguaje que cuenta con la capacidad de sobrecargar métodos de tal forma que pueden existir múltiples versiones de un mismo método, cada uno tomando una lista de parámetros distinta.
- § Se ha añadido un control estructurado de excepciones moderno, dando paso a la superación del arcaico On Error GoTo. Existen operadores aritméticos y lógicos abreviados, inicialización de variables durante la declaración, mayor consistencia de tipos de datos, etc.
- § No solo se puede crear aplicaciones para Windows y componentes para ellas, también se puede utilizar Visual Basic .NET para producir aplicaciones de Consola, servicios de Windows, interfaces Web, componentes de Servidor ASP .NET y Servicios Web.

6.1.4 Bases de Datos

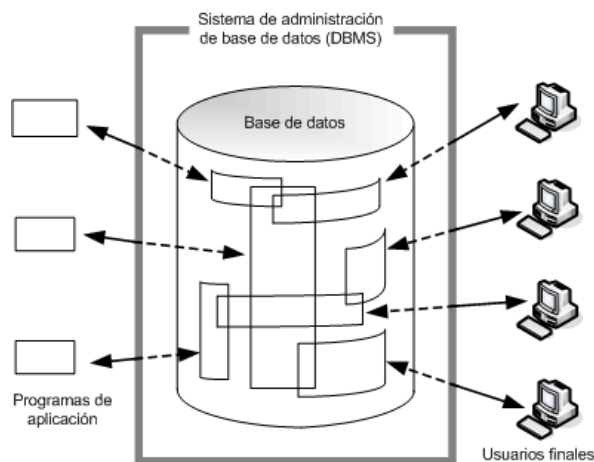
Los sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de sistema gestor de bases de datos y 'Data Base Management System', su expresión inglesa.

Un servidor de base de datos es por tanto el equipo en el que corre un sistema de gestión de bases de datos. Durante la realización de la práctica se brindó apoyo al diseño y construcción de aplicaciones distribuidas en el framework .net con el motor de bases de datos postgresql y para el banco de datos de terceros se cuenta con ms access.

DEFINICIÓN DE UNA BASE DE DATOS

Una Base de datos es un conjunto de archivos almacenados de forma *integrada y compartida*, cuya finalidad es almacenar información y permitir a los usuarios recuperar y actualizar esa información con base en peticiones; este manejo implica el uso de un software administrados DBMS (Sistema de Administración de Base de Datos).

Figura 11. Imagen simplificada de un sistema de base de datos¹²



¹² Fuente: Introducción a los Sistemas de bases de datos. Date C.J. – Prentice Hall, 2001 Séptima edición.

NIVELES DE UNA BASE DE DATOS

- § **Nivel Interno:** Está constituido por el conjunto de bloques de discos, con sus diferentes registros y sus respectivas direcciones, apuntadores, contadores y datos.
El nivel interno es el interactúa directamente con el sistema operativo (el servidos de disco) y a la tarjeta controladora del disco, por lo cual es un esquema físico binario, escondido en la intimidad del hardware.
- § **Nivel Conceptual:** Corresponde a la percepción de datos común a todos los usuarios de una organización. Por lo tanto se ve como un conjunto universal de entidades vinculadas entre sí, y cada una con sus propios atributos.
La apariencia de este nivel salta a la vista pero es generalmente amorfo y sin existencia física.
- § **Nivel Externo:** Corresponde a la percepción de datos que individualmente o en grupo tienen los usuarios de una misma área funcional de una organización.
Es una vista o subconjunto del conjunto universal, que describe únicamente la parte de los datos de interés para cada usuario y que, tampoco tiene existencia física.

Cada vista puede omitir uno o más registros, atributos o relaciones del esquema conceptual, o también cambiar su orden.

PROPIEDADES DE UNA BASE DE DATOS

- § **Independencia de los Datos:** La independencia de los datos es un objetivo principal de los sistemas de Base de Datos, se define como la inmunidad de las aplicaciones a cambios en la representación física y en la técnica de acceso; lo que implica que las aplicaciones involucradas no dependan de ninguna representación física o técnica de accesos en particular.
Es la capacidad de modificar los esquemas interno o conceptual sin causar cambios en los programas (que usan esquemas externos).
La independencia entre lo externo y lo interno se llama “Independencia física” y, entre lo externo y lo conceptual “Independencia lógica”. La independencia lógica sirve para el mantenimiento pero fomenta la separación entre la estructura y la funcionalidad organizacional.
- § **Integridad:** La integridad es el hecho de asegurar que los datos de una base de datos estén correctos. El control centralizado de la base de datos puede ayudar a evitar inconsistencia en la información, permitiendo que el

administrador de la base de datos simplemente las restricciones de integridad o reglas del negocio.

Los datos, apuntadores, direcciones, enlaces, índices y contadores, deben mantenerse siempre incorruptibles.

La corrupción puede aparecer por fallas del hardware, defectos del software, actualizaciones incompletas o inválidas, o la violación a las reglas de integridad referencial.

§ **Seguridad:** Los datos deben estar protegidos contra accesos no autorizados, y además reservados en diferentes rangos de permisividad para accesos autorizados.

Al tener la completa jurisdicción sobre la base de datos, el DBA puede asegurar que el único medio de acceso a la base de datos sea a través de los canales adecuados y por lo tanto puede definir las reglas o restricciones de seguridad que serán verificadas siempre que intente acceder a los datos. Es posible establecer diferentes restricciones para cada tipo de acceso (recuperación, inserción, eliminación y actualización) para cada parte de la información de la base de datos.

§ **Control de Redundancia:** Es la propiedad por medio de la cual, la repetición de datos es mínima y es controlable.

El proceso de normalización busca precisamente reducir al máximo la redundancia de datos, aunque nunca al 100%.

§ **Consistencia:** Es la propiedad que impide las contradicciones entre datos.

Si la redundancia estuviera controlada, un mismo dato repetido en lugares diferentes podría tener valores diferentes, y por tanto inconsistentes. Resulta claro que una base de datos en un estado inconsistente es capaz de proporcionar a sus usuarios información incorrecta o contradictoria.

§ **Control de Concurrencia:** Es la capacidad para ejercer un riguroso control en la ejecución simultánea de transacciones con el fin de proteger la consistencia de las actualizaciones y consultas a la Base de datos.

§ **Auditoría:** Es el examen de la Base de datos y su entorno, a fin de comprobar que se ajusta a lo establecido. Éste examen no es sólo a posteriori sino que también es a priori.

§ **Respaldo y Recuperación:** Es la facilidad para obtener copias de la Base de datos (backups de respaldo) que permitan recuperar la Base de datos ante cualquier falla.

§ **Rutas de Acceso:** Es la facilidad para obtener diferentes rutas de acceso, por medio de claves primarias y secundarias, pudiendo obtener respuesta a diferentes consultas aplicando diversos criterios de búsqueda.

🚦 **DEFINICIÓN DE SISTEMA DE ADMINISTRACIÓN DE BASE DE DATOS (DBMS)**

Un DBMS es el software que maneja todo acceso a la base de datos y tiene las siguientes funciones:

- § **Definición de datos:** es la creación de objetos en la base de datos a partir de sentencias. En otras palabras el DBMS incluye entre sus componentes un compilador DDL (Lenguaje de Definición de Datos).
- § **Manipulación de Datos:** es el manejo de peticiones para recuperar, actualizar o eliminar datos existentes en la base de datos o agregar nuevos datos a ésta. En otras palabras el DBMS incluye un componente compilador DML (Lenguaje de Manejo de Datos).
- § **Optimización y ejecución:** las peticiones DML son procesadas por el componente optimizador, cuya finalidad es determinar una forma eficiente de implementar la petición.
- § **Seguridad e Integridad:** su función es vigilar las peticiones del usuario y rechazar todo intento de violar las restricciones de seguridad e integridad definidas por el DBA.
- § **Recuperación de datos y concurrencia:** el componente del DBMS denominado administrados de transacciones o monitos de procesamiento de transacciones (monitor PT) establece ciertos controles de recuperación y concurrencia.
- § **Diccionario de Datos:** el diccionario de datos es el conjunto de datos que describen la BD, y contiene:
 - La descripción de todos los esquemas (externo, conceptual e interno).
 - La descripción de todos los campos (o atributos), registros (o tuplas) y referencias cruzadas entre los registros de varios archivos (o tablas, bárreles o entidades).
 - Los códigos de autorización y seguridad de los datos y sus redefiniciones con las que puedan ser referidos con nombres diferentes y en programas diferentes.

Para llevar a cabo su función de forma óptima, un DBMS se apoya en el Administrador de Base de Datos (DBA).

ADMINISTRADOR DE BASE DE DATOS (DBA)

El Administrador de Datos (DA) es la persona que toma las decisiones de estrategia y política con respecto a los datos de la empresa y el DBA es la persona que proporciona el apoyo técnico necesario para implementar dichas decisiones. Por lo tanto, el DBA es el responsable del control general del sistema a nivel técnico. Algunas de las tareas del DBA son las siguientes:

- § Definir el Esquema Conceptual.
- § Definir el Esquema Interno.
- § Establecer un Enlace con los Usuarios.
- § Definir las restricciones de Seguridad y de Integridad.
- § Definir las políticas de vaciado y recarga.

6.1.5 Contabilidad Colombiana vigente

Otro de los factores que inciden en el manejo interno de una organización es su área contable, por tal motivo es indispensable reconocer conceptos propios de la rama y su funcionamiento en sí. La definición según algunos expertos es: “La ciencia y técnica que enseña a recopilar, clasificar y registrar de una forma sistemática y estructural, las operaciones mercantiles realizadas por una empresa, con el fin de producir informes que, analizados e interpretados, permitan planear, controlar y tomar decisiones sobre la actividad de la empresa”¹³, de manera que contabilidad se refiere al correcto manejo de la parte financiera de la organización y busca ofrecerle una herramienta que permita, bajo estándares, un reporte completo del manejo de sus activos y pasivos que han sido transados durante un lapso de tiempo determinados.

La contabilidad como ciencia

La Contabilidad puede definirse como una ciencia instrumental que se ocupa de sistematizar y generalizar los métodos contables, al tiempo que investiga la consecución de otros nuevos.

Es una ciencia instrumental porque el objeto de su estudio consiste en instrumentos o métodos. La ciencia de la contabilidad se dirige a la construcción de un cuerpo metodológico, que proporcione una herramienta de análisis a otras ciencias, particularmente la economía de la empresa.

Debido a esto último, hay una clara subordinación de la contabilidad respecto a la economía, y también en gran medida al derecho.

¹³ Tomado de: <http://www.gerencie.com/contabilidad.html>

Las obligaciones y derechos que integran el patrimonio de la empresa y su vida entera en suma, están minuciosamente relacionados por el ordenamiento jurídico.

Muchas anotaciones contables se hallan obligatoriamente prefijadas por la ley que concede importancia probatoria e impone requisitos formales a la manera de llevar la teneduría de libros.

Objetivos de la contabilidad

- § Obtener en cualquier momento una información ordenada y sistemática sobre el desenvolvimiento económico y financiero de la empresa.
- § Establecer en términos monetarios, la cuantía de los bienes, deudas y el patrimonio que posee la empresa.
- § Llevar un control de todos los ingresos y egresos.
- § Facilitar la planeación, ya que no solamente da a conocer los efectos de una operación mercantil, sino que permite prever situaciones futuras.
- § Determinar las utilidades o pérdidas obtenidas al finalizar el ciclo contable.
- § Servir de fuente fidedigna de información ante terceros.

Toda esta información acompañada de una herramienta software que facilite su almacenaje y procesamiento se convierten en el brazo fuerte de una compañía sólida y de igual manera permite avanzar cuando se es interpretada de manera adecuada por quien corresponde; es decir: los reportes contables entregan al representante de una compañía la información necesaria para tomar decisiones que afectaran el futuro de la misma, así pues esta metodología es la base para cumplir con los objetivos visionales de cualquier empresa. En Colombia se presentan aspectos legales relacionados con la contabilidad que a pesar de estar fuera del contexto de este documento nos es necesario exponer debido a que forma parte de la finalidad de la herramienta en sí: existe el decreto 2649 de 1993 que expone el marco jurídico relacionado con los conceptos referentes a una contabilidad legal y aquellas normas a cumplir para que su validez como prueba sea efectiva, esto es oficio de aquella persona que asume el nombre de Contador que según la ley 43 de 1990 es aquella persona que ejerce dicha profesión con el título requerido para la misma y posee las facultades legales para hacerlo, reglamentando así sus funciones para el ejercicio de la profesión. De la misma manera se encuentran reglamentadas las cuentas, subcuentas y grupos a utilizar como referencia para el proceso contable en dicho marco; a este documento se le conoce como el Plan Único de Cuentas Nacional¹⁴.

¹⁴ Más información en www.puc.com.co

6.2 Fundamento teórico específico:

Junto a los conceptos que se han detallado anteriormente, es necesario enfocarnos en algunos temas que pueden ser de utilidad para adquirir conceptos que se requerirán más adelante durante el desarrollo de la práctica. Algunos temas requieren de una profundización mayor o de una aplicabilidad específica a cualquier tema en particular, es por tanto que a continuación se señalaran de acuerdo al tema en el que se pretendan implementar.

Se pretende exponer en el presente documento un glosario de conceptos que puedan abrir paso al lector del mismo y colaboren en el proceso de facilitar la comprensión de los mismos, de manera que se detallaran dichos conceptos temáticamente.

6.2.1. Contabilidad por partida doble

Hacer referencia a una contabilidad por partida doble significa dar a entender el sistema contable que la empresa está usando; dicho sistema es utilizado para reflejar todas las transacciones financieras.

Es la base de la contabilidad actual y consiste en un movimiento contable que afecta a un mínimo de dos asientos o cuentas, un débito y un crédito y, además, tiene que haber la misma cantidad de débitos que de créditos; en otras palabras, débito menos crédito ha de ser cero, para que haya equilibrio en la contabilidad.

La partida doble nace de un hecho económico entre dos partes y se basa en la teoría de "no hay deudor sin acreedor", es decir, que si se compra un producto quien lo compre será el deudor y el proveedor será el acreedor.

Claro está que estos principios no son tan obvios en todos los casos, como por ejemplo en movimientos internos de la empresa en los que se retira efectivo de la caja para consignarlo a un banco; en esta caso, se cumple otro de los principios elementales de la partida doble que señala que: una cuenta da al tiempo que otra cuenta recibe, dándose la contrapartida, como característica propia de la partida doble, puesto que sin esta no podría existir equilibrio. No pueden haber un movimiento que solo afecte cuentas de una misma naturaleza; si una cuenta se afecta en su lado positivo, la otra se debe afectar en su lado negativo.

La partida doble se compone de una cuenta, la cual debe tener un nombre, un código y un valor negativo y otro positivo. El valor negativo corresponde a un crédito y el valor positivo corresponde a un débito. Su funcionamiento consiste en registrar un valor en por lo menos dos cuentas: el valor debe ser igual en una y otra cuenta; en una cuenta se registra su débito y en la otra el crédito.

En algunos casos se pueden afectar más de dos cuentas, pero aun así, los débitos y los créditos de las cuentas afectadas deben ser iguales, lo que es lo mismo, si se restan los créditos de los débitos el resultado debe ser cero.

Esto permite conservar el equilibrio que también profesa la ecuación patrimonial. Si los débitos representan lo que se tiene, los créditos representan lo que se debe. Lo que se tiene es el activo y lo que se debe puede ser patrimonio y/o pasivo. De ahí la ecuación patrimonial (Activo es igual a pasivo más patrimonio, o activo menos pasivo es igual a patrimonio).

La técnica contable de hoy día, está fundamentada en la partida doble, la cual se basa en el aumento y disminución de las diferentes cuentas que hacen parte del sistema contable del ente económico.

Esta técnica la podemos representar de la siguiente manera:

Tabla 1. Representación de la técnica contable: Partida doble

Débitos	Créditos
Aumento de activos	Disminuciones del Activo
Disminuciones del Pasivo	Aumento de pasivos
Disminuciones del Patrimonio	Aumento de patrimonio

Los siguientes son algunos de los principios fundamentales de la partida doble:

- § Quien recibe es deudor, quien entrega es acreedor.
- § No hay deudor sin acreedor, ni acreedor sin deudor. (Los recursos no surgen por si solos, debe proceder de algún lado)
- § Todo valor que entra debe ser igual al valor que sale. (Es la base del equilibrio de la ecuación patrimonial)
- § Todo valor que entra por una cuenta debe salir por la misma cuenta. (Una cuenta así como se crea se debe también eliminar de la misma forma).

La partida doble, equivocadamente se cree que fue ideada por Lucas Pacciolo, pero fue en realidad un monje de la orden franciscana quien en el siglo XV en la obra "*Summa de Aritmética, Geometría, Proportioni et Proportionalita*" hace referencia al "*Método Veneciano*" de teneduría de libros que posteriormente pasaría a llamarse Método de la partida doble.

6.2.2. Imputación Contable

Es posible obtener una pequeña definición del concepto si nos referimos a los procesos contables que involucra, de tal manera que cuando se hace referencia al término se dice que es una operación consistente en asignar o atribuir determinados costes a un producto o centro de costes concreto, cualquiera que sea el criterio o clave de distribución que se utilice. De igual forma se crean conceptos paralelos que permiten indicar que es una afectación de una anotación o de una operación a la cuenta, al servicio o a la función a la que pertenece.

Utilización

La Contabilidad de costes por procesos permite la imputación de gastos generales de centros de coste, en primer lugar en los procesos empresariales y, posteriormente, en los objetos de coste. Esto hace posible el análisis de los gastos generales por centro de coste (como por ejemplo área funcional o área de responsabilidad) y por proceso empresarial. La Contabilidad de centros de coste indica dónde se produjeron los costes, mientras que la Contabilidad de costes por procesos indica las actividades (de la empresa) que los produjeron.

Integración

Los gastos generales se transfieren desde la *Gestión financiera* y se asignan en primer lugar a los centros de coste de la Contabilidad de centros de coste; si se utiliza la Contabilidad de costes por procesos, los gastos generales se transferirán desde la Contabilidad de centros de coste a los procesos empresariales de la contabilidad de costes por procesos.

A partir de la Contabilidad de centros de coste o la Contabilidad de costes por procesos, los gastos generales se transfieren al “Controlling” de costes del producto. Desde la Contabilidad de objetos de coste es posible transferir los costes a la Gestión financiera, por ejemplo, para valorar inventarios de productos terminados o no terminados, así como transferirlos a la Contabilidad de centros de beneficio y a la Cuenta de resultados. Los gastos generales no asignados a ningún objeto de coste también se pueden transferir directamente desde la Contabilidad de centros de coste o desde la Contabilidad de costes por procesos a la Cuenta de resultados (como es el caso de los costes de ventas y marketing).

Condiciones previas

La Contabilidad de costes por procesos imputa los gastos generales a los objetos de referencia (en este caso: objetos de coste), en base a los parámetros definidos en un esquema de procesos. En el amoldamiento para el control de costes del producto en la contabilidad de objetos de coste, se definen esquemas de procesos y la determinación de esquema de procesos con las siguientes características:

En algunos sistemas es posible utilizar la contabilidad de costes por procesos para todos los objetos de coste.

- § Órdenes de fabricación
- § Órdenes de proceso
- § Colectores de costes de producción
- § Nodos de objetos de coste de una jerarquía de objetos de coste
- § Posiciones de documento comercial

La Contabilidad de costes por procesos puede utilizarse tanto para los datos plan como para los datos reales. Mientras que la planificación de centros de costes representa el nodo de partida, los datos reales se componen de las contabilizaciones actuales en centros de coste.

La medición manual de los recursos utilizados por procesos (base de referencia) y de las magnitudes de referencia representa una gran cantidad de trabajo. No obstante, los datos necesarios con frecuencia ya están disponibles como estadísticas y pueden tomarse del SIL (Sistema de información para logística).

También se puede implementar la Contabilidad de costes por procesos como un sistema de contabilidad de costes estadístico paralelo. En este caso no existe cargo operacional de los objetos de coste.

6.2.3. Librerías dinámicas o Librerías de enlace dinámico:

Existen algunos tipos de archivos que se desarrollan con el fin de ser ejecutados bajo otros entornos y servir de apoyo a los procesos para los cuales fueron creados; entre los más comunes se encuentran las librerías de enlace dinámico, cuyas características principales se citaran a continuación.

Como base se empezara por definir lo que es un archivo DLL: Un archivo DLL es una biblioteca que contiene el código y los datos que se pueden utilizar por más de un programa al mismo tiempo. Por ejemplo, en sistemas operativos Windows, el archivo Comdlg32 DLL realiza funciones relacionadas con el cuadro de diálogo. Por lo tanto, cada programa puede utilizar la funcionalidad

contenida en este archivo DLL para implementar un cuadro de diálogo 'Abrir' . Esto ayuda a promover la reutilización de código y el uso eficaz de la memoria.

Mediante el uso de un archivo DLL, un programa puede ser llevado a módulos en componentes separados. Por ejemplo, un programa de contabilidad puede ser vendido por módulos. Cada módulo se puede cargar en el programa principal en tiempo de ejecución si se instala ese módulo. Dado que los módulos son independientes, el tiempo de carga del programa es más rápido y sólo se carga un módulo cuando se solicita esa funcionalidad.

Además, las actualizaciones son más fáciles de aplicar a cada módulo sin afectar a otras partes del programa. Por ejemplo, puede tener un programa de nóminas, y los tipos impositivos cambian cada año. Cuando estos cambios se aíslan en un archivo DLL, puede aplicar una actualización sin necesidad de generar o volver a instalar todo el programa.

Ventajas DLL

La siguiente lista describe algunas de las ventajas que se proporcionan cuando un programa utiliza un archivo DLL:

- § Utiliza menos recursos: Cuando varios programas utilizan la misma biblioteca de funciones, un archivo DLL puede reducir la duplicación de código que se carga en el disco y en la memoria física. Esto puede influir en gran medida en el rendimiento no sólo el programa que se ejecuta en primer plano, sino también de otros programas que se ejecutan en el sistema operativo Windows.
- § Promueve la arquitectura modular: Un archivo DLL ayuda a promover el desarrollo de programas modulares. Esto le ayudará a desarrollar programas grandes que requieren varias versiones de idioma o un programa que requiere la arquitectura modular. Un ejemplo de un programa modular es un programa de contabilidad que tiene muchos módulos que se pueden cargar dinámicamente en tiempo de ejecución.
- § Instalación e implementación de las aceleraciones: Cuando una función dentro de un archivo DLL necesita una actualización o una solución, la implementación e instalación de la DLL no requiere que el programa deba volver a vincularse con el archivo DLL. Además, si varios programas utilizan la misma DLL, estos programas se beneficiarán de la actualización o la revisión. Este problema puede producirse con más frecuencia cuando se utiliza un archivo DLL de terceros que se actualiza con regularidad o que se fija para dar por sentado que la disponibilidad del mismo es abierta.

Exportar funciones DLL

También es posible exportar funciones declaradas siempre y cuando el archivo ofrezca los permisos suficientes; de esta forma no será necesario llamar la librería completa sino que solamente se hará uso de la función que el programa cliente lo requiera, esto se logra mediante la implementación de la palabra clave reservada: "EXPORTS".

6.2.4. Clases y Objetos

Teniendo en cuenta que el paradigma de la programación bajo el cual se trabaja este proyecto es POO lo cual quiere decir: 'Programación Orientada a Objetos', es necesario puntualizar en algunos detalles y/o conceptos que se encuentran demarcados por esta filosofía. Los conceptos que se mencionarán a continuación se encuentran fuertemente ligados al entorno en el cual se desarrollara la aplicación y hacen parte del conocimiento requerido para codificar la algoritmia a implementar en la herramienta software.

Una "Clase" es una entidad estática que se asemeja a la clasificación por tipos en un mundo real; dicha clase define la forma y el comportamiento de sus objetos. Dentro de ella se encuentran dos formas básicas de componentes: los atributos y los métodos.

- § **Atributo:** son variables que codifican el estado de un objeto; es decir: son los elementos unitarios que referencian las particularidades de cada entidad que posee las características de dicha clase.
- § **Método:** procedimientos que describen la acción que se debe realizar en un determinado momento, conforman el cuerpo de la clase y se encargan de establecer patrones de funcionamiento para todos sus objetos derivados.

Un "Objeto" es una instancia de una clase; es decir: todo elemento derivado de una clase que cumpla con todas las características y particularidades que la misma propone, siendo una representación 'virtual' de dicha clase.

La ideología del manejo en clases es dar la posibilidad de contar con características individuales (propias de cada objeto) pero manteniendo un esquema común con todos los que son de su tipo (aquellos objetos que pertenecen a la misma clase tendrán iguales todas las cualidades impuestas por la clase en común).

Para este tipo de situaciones se recurren a ejemplos ilustrativos que aclaren el significado de cada uno de los conceptos mencionados. Entre las representaciones más visuales se puede enunciar la siguiente:

Se puede tomar como símil a una “Clase” el concepto de “Automóvil”, el cual tiene algunas características propias de un vehículo (atributos) como por ejemplo: tiene cuatro llantas, posee una estructura metálica de soporte, cuenta con puertas, etc. El vehículo además realiza ciertas acciones propias del mismo (métodos), como por ejemplo: avanza hacia adelante, retrocede, toca la bocina, frena, etc. Pero como es lógico, no todos los vehículos son iguales, entonces podemos decir que existen diferentes tipos (instancias / objetos) de vehículos que además de cumplir con las cualidades de serlo, se diferencian entre sí por algunos detalles pero aun así no dejan de ser vehículos. Un automóvil policial, una ambulancia, una camioneta, un camión son tipos de vehículos que cumplen con las características anteriormente mencionadas pero en su particularidad poseen detalles que los identifican de los demás, convirtiéndolos así en “Objetos” de la “Clase” automóvil.

6.2.5. PostgreSQL

Antecedentes

- § La implementación del DBMS POSTGRES empezó en 1996 en la Universidad de Berkeley en California por el profesor Michael Stonebraker.
- § PostgreSQL es una herramienta de código abierto descendiente del original “Ingres” de Berkeley.
- § La creciente comunidad de usuarios demandó gran cantidad de esfuerzo en soporte y mantenimiento, que debería ser invertido en Investigación. Por lo cual Berkeley terminó oficialmente el proyecto en 1993.
- § En 1994, Andrew Yu y Jolly Chen le adicionaron un intérprete SQL. Postgres95 fue lanzado en la Web para que encontrara su propio camino. Fue escrito en C ANSI y se le realizaron algunas mejoras (funciones agregadas, group by, libpg, psql, libpgtcl, objetos grandes).
- § En 1996, se cambió el nombre PostgreSQL y se empezó a numerar desde la versión 6.0.

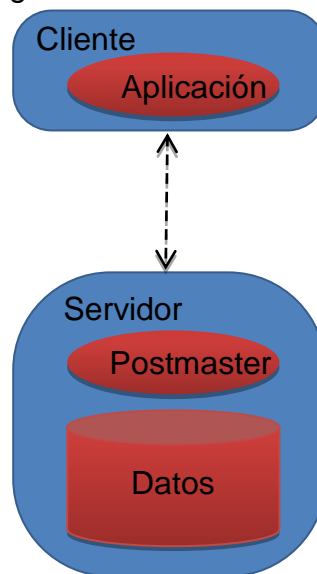
Arquitectura

Las bases de la arquitectura del sistema PostgreSQL, entendiendo como las partes de Postgres, usa un modelo cliente/servidor conocido como “proceso por usuario”.

Una sesión PostgreSQL consiste en los siguientes procesos cooperativos:

- § **Un proceso servidor:** maneja los archivos de base de datos, acepta conexiones a la base de datos y permite acciones sobre estos archivos. El proceso servidor es llamado **Postmaster**.
- § **Las aplicaciones de usuario final (frontEnd):** permiten realizar operaciones sobre la base de datos. Pueden ser de diversa naturaleza: una herramienta de texto o consola, una aplicación gráfica, o una interfaz Web.

Figura 12. Arquitectura PostgreSQL



Fuente: EL Autor

Un único Postmaster controla una colección de bases de datos dadas en un único host. Debido a esto una colección de bases de datos se suele llamar una instalación o un sitio.

Las aplicaciones frontEnd que quieren acceder a una determinada base de datos hacen llamadas a la librería y ésta envía peticiones de usuario a través de la red al Postmaster (en donde se establece una conexión), el cual en respuesta inicia un nuevo proceso en el servidor (backend).

Características

Las principales mejoras en PostgreSQL incluyen:

- § Los bloqueos de tabla han sido sustituidos por el control de concurrencia multi-versión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite

copias de seguridad en caliente desde pg_dump mientras la base de datos permanece disponible para consultas.

- § Se han implementado importantes características del motor de datos, incluyendo sub consultas, valores por defecto, restricciones a valores en los campos (constraints) y disparadores (triggers).
- § Se han añadido funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, forzado de tipos cadena, literales, conversión de tipos y entrada de enteros binarios y hexadecimales.
- § Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.
- § La velocidad del código del motor de datos ha sido incrementada aproximadamente en un 20 – 40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fue lanzada.

Aislamiento transaccional

El estándar ANSI/ISO SQL define cuatro niveles de aislamiento transaccional en función de tres hechos que deben ser tenidos en cuenta entre transacciones concurrentes. Estos hechos no deseados son:

- § **Lecturas “sucias”**, una transacción lee datos escritos por una transacción no esperada, no confirmada (committed).
- § **Lecturas no repetibles**, una transacción vuelve a leer datos que previamente había leído y encuentra que han sido modificados por una transacción confirmada.
- § **Lecturas “fantasma”**, una transacción vuelve a ejecutar una consulta, devolviendo un conjunto de fila que satisfacen una condición de búsqueda y encuentra que otras filas que satisfacen la condición han sido insertadas por otra transacción confirmada.

Bloqueos

Postgres ofrece varios modos de bloqueo para controlar el acceso concurrente a los datos en tablas. Algunos de estos modos de bloqueo los adquiere Postgres automáticamente antes de la ejecución de una declaración, mientras que otros son proporcionados para ser usados por las aplicaciones.

Todos los modos de bloqueo (excepto para AccessShareLock) adquiridos en una transacción se mantienen hasta la duración de la transacción.

Existen dos tipos de bloqueos:

- § Bloqueos a nivel de tabla
- § Bloqueos a nivel de registro.

Índices

Internamente Postgres almacena los datos en archivos del sistema operativo; cada tabla tiene su propio archivo.

Un índice es un archivo separado ordenado por una o más columnas, contiene apuntadores al archivo de datos permitiendo rápido acceso a la tabla.

Clúster

El comando clúster reordena físicamente una tabla para que coincida con el orden del índice. Este comando es de gran valor cuando el desempeño es crítico y cuando existen muchos valores duplicados.

La tabla es realmente copiada a una tabla temporal en el orden indicado, luego renombra la nueva con el nombre de la original. Por esta razón, todos los permisos sobre la tabla se pierden.

Vacuum

Cuando Postgres actualiza un registro, mantiene una copia del original y agrega uno nuevo. El registro original es marcado como borrado. De manera similar los registros eliminados con marcados como borrados.

El comando VACUUM remueve los registros marcados como borrados del archivo. Se debe correr VACUUM con cierta frecuencia sobre tablas que tienen gran movimiento.

Es posible correr el comando sobre una tabla o sobre toda la Base de datos.


6.2.6. NPGSQL

Definición:

Npgsql es un proveedor de datos .NET para el servidor PostgreSQL, que permite la interacción entre cualquier aplicación cliente y el servidor. Cumple funciones específicas para todo entorno y se encuentra activamente desarrollado en base a los lineamientos especificados en la documentación .NET mencionada en el apartado anterior de este capítulo.

Para hacer uso de Npgsql es necesario que el servidor Postgres mantenga activo el servicio de 'escucha' de conexiones TCP / IP puesto que es el medio por el cual se interconecta la librería.

Básicamente el proveedor de datos funciona bajo el mismo ideal de una librería dinámica, de la cual se comentan sus características en la sección 6.2.1 del presente documento.

 **Procedimiento para su instalación:** para trabajar con las funcionalidades de este proveedor es necesario chequear que cumpla con los requisitos externos anteriormente mencionados, además de ello se deben seguir los siguientes pasos para crear una conectividad exitosa entre el framework de Visual Studio .NET y el gestor PostgreSQL.

- § Se accede a la página de descarga de la librería¹⁵ donde se encontrará un archivo comprimido en formato .ZIP.
- § Se descomprimen las librerías Npgsql y Mono.Security y se adicionan a la carpeta debug del proyecto .NET.
- § Dentro de la interfaz del proyecto abierto en Visual Studio se selecciona la opción agregar referencia del menú herramientas y se cargan las librerías descomprimidas anteriormente.
- § Llamar a la clase Npgsql por medio de la palabra reservada 'IMPORTS'.

En este momento ya es posible acceder a todos los métodos de la librería de la misma forma que se accede a cualquier otra perteneciente al entorno.

¹⁵ http://pgfoundry.org/frs/?group_id=1000140

7. DESARROLLO DE LA PRÁCTICA


El presente capítulo describe las actividades realizadas durante la práctica que contribuyeron a la consecución de los objetivos propuestos en el plan de trabajo. Es de resaltar que las labores desempeñadas estaban acordes al cargo designado dentro de la organización y no se encontraban ligadas al resultado del proyecto desarrollado sino al apoyo en los procesos de su implementación, construcción y puesta en marcha, acorde al título de la práctica.

7.1. Antecedentes

Para el desarrollo de la presente práctica se ha tomado como base un aplicativo contable desarrollado anteriormente por EVACC Ltda al que han nombrado como 'Contar' y que fue implementado en un lenguaje orientado a eventos como se mencionaba anteriormente; la base de datos de la cual se alimentaba este aplicativo se encuentra bajo ACCESS de Microsoft lo que converge en un tiempo mayor a la hora de acceder a los datos contenidos en la misma. De la misma forma la algoritmia usada para el acceso a los datos no era la óptima y esto impedía un funcionamiento eficaz de la aplicación.

7.1.1 Descripción de 'Contar' en su versión anterior.

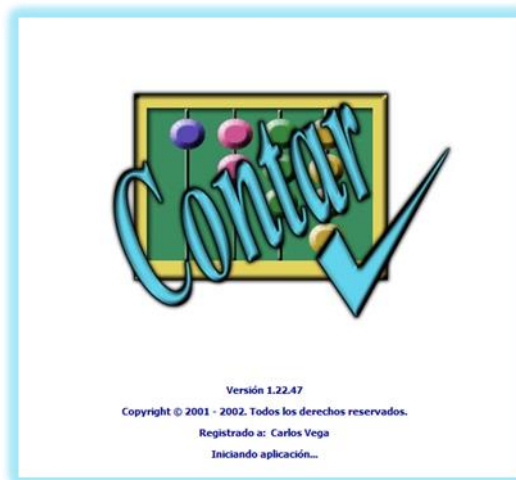
Por reglamentaciones internas de EVACC Ltda ha sido imposible la publicación del código fuente de la aplicación anterior, por tal motivo esta sección se centrará en los aspectos visuales, los problemas de accesibilidad de la misma y la organización de su base de datos.

 **ASPECTOS VISUALES:** la interfaz de una herramienta software debe cumplir con los requisitos necesarios mínimos que puedan hacer de su uso algo agradable y a la vez debe impactar para causar la sensación de estar trabajando con algo realmente útil y funcional, todo esto se logra con la combinación perfecta de colores a usar, la distribución y localización de botones y accesos directos y la mayor cantidad de información útil que se le pueda brindar al usuario; es decir, guiarle a través de su uso y alertar cuando algo no se está haciendo bien. A continuación se expondrá con detalle las particularidades de la herramienta mencionada.

§ El diseño del logotipo que identificaba la herramienta es un ábaco, así es y se seguirá manteniendo, teniendo en cuenta que la finalidad de representarse por medio de esta figura es hacer alusión al primer objeto que se usó para “contar”, pero visualmente no era posible identificar este mensaje ya que la combinación de colores no lo facilitaba, en apartados posteriores es posible detallar la nueva apariencia que se le ha asignado y los cambios que se le han realizado.

A continuación se presenta la imagen del formulario inicial de la aplicación que se usaba como presentación.

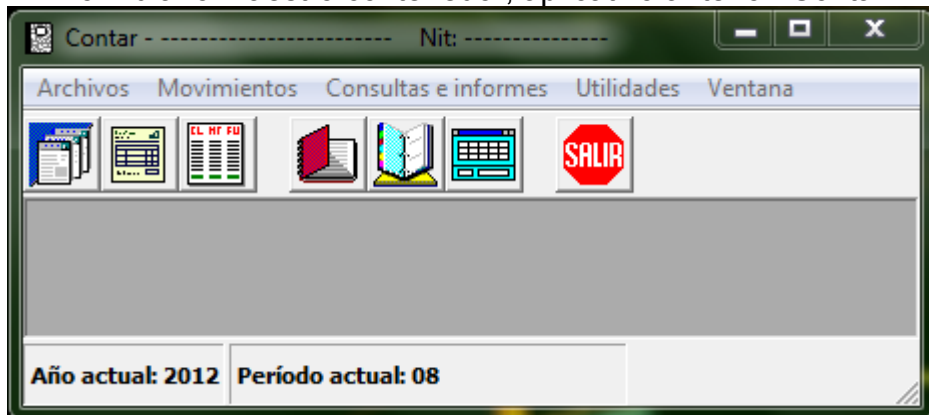
Figura 13. Logo inicial aplicativo anterior ‘Contar’



Fuente: El Autor

La aplicación consta de un formulario principal que juega el papel de 'MdiContainer maestro' dentro del cual se ejecutan los demás sub-formularios del aplicativo. La siguiente imagen muestra el estado inicial del formulario tras su ejecución.

Figura 14. Formulario maestro contenedor, aplicativo anterior 'Contar'



Fuente: El Autor

En la figura se observa la iconología usada para los accesos directos a cada uno de los módulos y su dificultad para ser interpretados y/o correlacionados a su módulo destino. De la misma manera presentan una desventaja en relación a su agrupación, ya que los botones que permiten el acceso a libros generales se encuentran antes del botón que permite el acceso a los saldos por cuenta y después de los botones de reportes de saldos acumulados; esto puede malinterpretarse por parte del usuario, haciéndole creer que los libros generales pertenecen a la sección de informes, cuando eso no es cierto.

Figura 15. Distribución de los iconos de navegación.



Fuente: El Autor

En el gráfico expuesto se puede detallar la forma en la que se pretendía que el usuario navegara a través de los registros, como se puede ver en la imagen, en cada uno de los sub-formularios se encontraba esta barra de navegación que contiene en su respectivo orden los iconos de:

- § *Añadir nuevo registro (a)*: botón que permite adicionar un registro, de acuerdo al módulo del cual pertenezca cada formulario.
- § *Editar el registro actual (b)*: botón que permite realizarle cambios al registro que actualmente se muestra en pantalla.
- § *Eliminar el registro actual (c)*: botón que permite la eliminación del registro que actualmente se muestra en pantalla.
- § *Guardar cambios (d)*: botón que permite guardar los cambios realizados al formulario; ya sea cuando se crea un nuevo registro o cuando se edita uno existente.
- § *Imprimir (e)*: botón opcional según el formulario donde se encuentre el usuario, permite crear y/o pre visualizar los informes que puedan ser generados por cada formulario.
- § *Buscar (f)*: botón que permite la apertura de un formulario de búsqueda de acuerdo al módulo que se esté trabajando.
- § *Salir (g)*: Botón que genera el cierre del formulario.
- § *Primer registro (h)*: botón que permite la visualización en pantalla del primer ítem de la tabla de registros en la que se encuentre trabajando el formulario.
- § *Siguiente registro (i)*: botón que avanza una posición en el listado de registros y lo muestra en pantalla.

- § *Registro anterior (j)*: botón que retrocede una posición en el listado de registros y lo muestra en pantalla.
- § *Ultimo registro (k)*: botón que permite la visualización en pantalla del último ítem de la tabla de registros en la que se encuentre trabajando el formulario.

Una de las inconsistencias fundamentales que presenta esta aplicación es que no se tuvo en cuenta la interrelación con el usuario y su apoyo en la caza de errores, es decir: no se limita al usuario de acuerdo a la actividad relacionada que esté llevando a cabo. Los botones que aquí se muestran permanecían habilitados en todo momento y eso se convirtió en la causa principal de muchas complicaciones que se presentaron en cuestiones de procesos.

De acuerdo a los nuevos estándares establecidos para el desarrollo de la aplicación en su siguiente versión, se pretende que este tipo de incidencias ya no afecten más, ni a desarrolladores, ni a usuarios finales. Este tipo de limitaciones impuestas se comentaran más adelante en este mismo apartado.

Figura 16. Entrega de informes

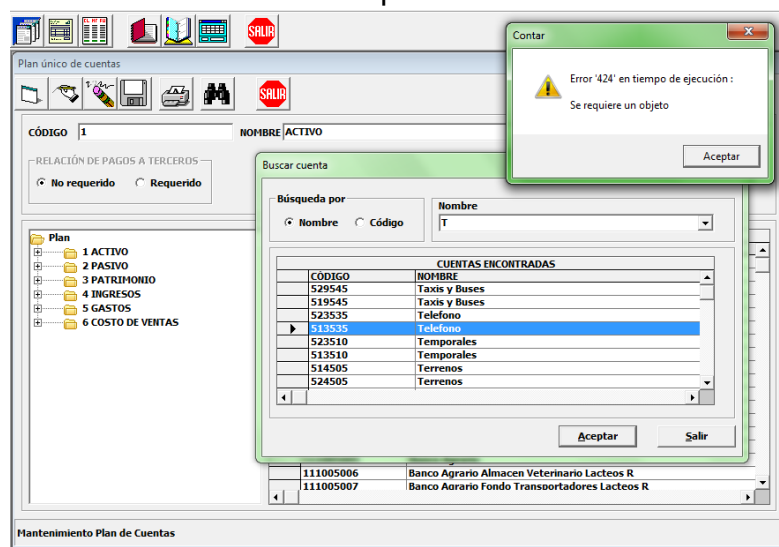
BALANCE DE COMPROBACIÓN		AÑO	PERÍODO	FECHA EMISIÓN:	
		2012	8	23-abr-2013	
CÓDIGO	NOMBRE CUENTA	SALDO ANTERIOR	DÉBITO ACTUAL	CRÉDITO ACTUAL	NUEVO SALDO
1	ACTIVO	1,071,030,864.21	375,920,088.00	375,919,088.00	1,071,031,864.21

Fuente: El Autor

Para la elaboración de informes se cuenta con una plantilla genérica como la que se expone en la imagen anterior, como se puede ver es un esbozo bastante básico y no resalta los detalles de los valores importantes de cada informe; de la misma forma – y como se venía comentando – la distribución de botones tampoco es muy versátil para que el usuario final se sienta cómodo con ella. Para ser un informe lineal que no requiere de mucho detalle se puede observar que aún le faltan distribuciones de espacio y un mayor aprovechamiento de resaltes.

✚ **ACCESSIBILIDAD:** la parte externa de una herramienta software es importante pero debe ir altamente ligada a su facilidad de acceso, si se encuentra que alguna interfaz a la vista es agradable pero es dispendioso poder acceder a cualquiera de sus componentes, dicho elemento no está cumpliendo con sus objetivos a cabalidad; en la herramienta con la que actualmente se cuenta se pueden ver algunos fallos en la restricciones de acciones, y en contraparte se permiten accesos a procesos que no deberían ser por el estado en el que se encuentran; es decir: se debe invalidar una acción cuando otra está realizando un procedimiento que afecte a la primera, como en el ejemplo de guardar y editar: no es posible editar un registro si el usuario se encuentra realizando el ingreso de uno nuevo, y de la misma manera solo es posible acceder al guardado de un registro cuando se ha editado o cuando se crean registros nuevos, pero no en otros caso. Para documentar lo que se ha expuesto anteriormente me permito relacionar a continuación un conjunto de imágenes que demuestran algunos de los procesos que presentan fallas con la versión 1.22.47 de contar.

Figura 17. Error en formulario de búsqueda módulo PUC



Fuente: El Autor

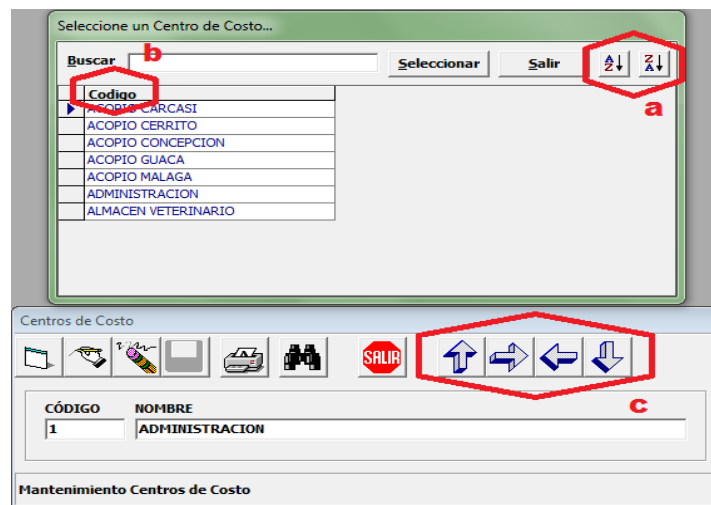
Lo que se detalla en la imagen es un ejemplo claro de las fallas que se presentan cuando se le permiten acciones al usuario que perjudican otros procedimientos, para este caso particular se encontró un problema de selección; en el formulario búsqueda de cuentas del puc se tiene un cuadro combinado donde es posible realizar búsquedas por medio de una lista desplegable pero no se tuvo la precaución de impedir al usuario que escribiera un texto que se encontrara fuera de los datos de origen, y si eso ocurre que lo informa por algún medio visible. La imagen expone lo sucedido en el momento en que se introduce al cuadro de texto una cadena de texto que no coincide con las descripciones de ninguna cuenta en el puc de la empresa y seguido

esto el usuario presiona el botón 'Aceptar'. Es también evidente la falta de precauciones en la métrica del código y el uso de la estructura "TRY – CATCH" de la cual se comentan algunos detalles en el marco teórico del presente documento. La falla es de tal consecuencia que la aplicación colapsa por completo y se cierran todos los formularios – incluso el principal –.

Algo muy similar sucede cuando este formulario se pretende cerrar sin seleccionar alguna cuenta; la interfaz muestra un cuadro de mensaje de error con el texto: "3633: Can't Load DLL, MSJTER35.DLL", que es un error propio de la aplicación y es generado por el mal uso de la codificación en la algoritmia.

Simultáneamente es notorio el error que se comete al mantener activos los cuadros de texto de código y descripción de la cuenta puc cuando no se le encuentra realizando ninguna acción a ese registro en particular; según los estándares para el diseño de software¹⁶, una caja de texto que el usuario ve como habilitada para su edición debe ser editable pues de lo contrario se puede cometer el desacierto de confundir al usuario y así arruinar la finalidad de la herramienta.

Figura 18. Defectos de usabilidad en el módulo de centros de costo



Fuente: El Autor

Otro de los factores que se tuvo en cuenta fue la presencia de botones sin funcionalidad alguna; en formularios como el de búsqueda de centros de costo (ordenamiento ascendente y descendente)^(a) el cual se publica en la figura anterior, que no surten ningún efecto en la ordenación de la tabla que se despliega en la parte inferior del mismo.

¹⁶ Véase: Estándares de desarrollo de software, Editorial Mediterráneo, Numeral 5.2 P 14 disponible en : http://www.ejie.net/documentos/Anexos_PBT/Estandares%20de%20desarrollo%20software.v2.0.pdf

A su vez se puede vislumbrar la incoherencia que presenta el título de la columna ^(b) con los datos o tipos de datos que contiene y esto demuestra un descuido que puede confundir al usuario de manera que es una práctica no recomendada en el desarrollo de software.

En la navegación por registros se contaba con una colección de iconos en forma de flechas^(c) apuntando hacia las 4 direcciones de un plano cartesiano; así que dichas flechas indicaban (en el orden de la figura):

- § Hacia arriba: desplazarse al primer registro.
- § Hacia la derecha: avanzar un registro.
- § Hacia la izquierda: retroceder un registro.
- § Hacia abajo: ir al último registro.

Este tipo de convenciones resultan siendo incomprensibles para el usuario final ya que pretender una navegación entre registros se asume como lineal y no dimensional, de manera que un desarrollador puede pretender que tales signos se comprendan de acuerdo a un paradigma que el usuario puede llegar a malinterpretar; este tipo de imperfectos pueden solucionarse cuando se siguen normas internacionales para el desarrollo de productos software con aplicabilidades comerciales.

Dentro del marco de las propiedades intrínsecas de la aplicación, es necesario resaltar aspectos como la necesidad que se tenía en esta versión de una búsqueda ágil, inteligente y acoplable a las necesidades el usuario, ya que dentro de los formularios de búsqueda de cada módulo era necesario ingresar coincidencias exactas acorde a los criterios para la misma, esto conllevaba a un resultado insuficiente y frustrante para el usuario final.

En el proceso de diseño, el equipo de desarrollo debe tener en cuenta aspectos importantes como las opciones que la herramienta brindará a posibles indecisiones por parte del cliente, en este aplicativo se ve la ausencia de este tipo de 'beneficios' puesto que, tal como se puede observar en las dos últimas imágenes, el usuario no cuenta con la opción de rehusarse a ejecutar el proceso que inicialmente él mismo solicitó; es decir: si tomo la decisión de guardar o eliminar, la única forma de abortar el proceso es salir del formulario y volver a llamarlo, lo cual genera pérdidas de tiempo, clicks e hilos de procesos.

7.1.2. Base de datos:

Como se ha comentado en apartes anteriores de este mismo documento, el gestor de base de datos con el cual trabajaba la aplicación era MS Access, es necesario comentar que las tablas se encontraban libres de procedimientos de correlación, estructuración, modelado y normalización, junto a esto se presenta el gran problema de la baja tasa de velocidad cuando se interconectan interfaces desarrolladas con un paradigma orientado a eventos y un gestor de tipo comercial como lo es Access que tiene un tiempo de respuesta elevado y un uso de procesador demasiado elevado, tal como se puede deducir del ANEXO A

Las tablas contenidas en el gestor anterior se encuentran relacionadas en el ANEXO B donde se detallan los nombres y tipos de campos de cada una. La versión del gestor en el cual se encontraba la base de datos era inferior a 2002 por lo cual fue necesario realizarle una actualización para que permitiera ser editada durante todo el proceso y poder así llevar a cabo pruebas de montaje y el desarrollo total de la práctica.

7.2 Levantamiento de requerimientos.

El proyecto principal asignado fue la reconstrucción de código fuente de la interfaz que anteriormente había sido desarrollada en un lenguaje orientado a eventos como lo es visual 6 y la reestructuración de la base de datos que se encontraba establecida en ACCESS a fin de ser implantada en el gestor PostgreSQL para su trabajo en conjunto.

La herramienta seleccionada para este desarrollo fue establecida en las condiciones de EVACC Ltda basada en criterios propios de lo que consideraron eran requisitos fundamentales en el nuevo sistema: confiabilidad, escalabilidad y tecnología de vanguardia. En su concepto estas características las cumple la plataforma .NET de Microsoft® y su herramienta Visual Studio 2010 Ultimate®. La aplicación ha sido designada con el nombre que su predecesora conservaba 'Contar' a lo cual se le adiciona la versión que se pretende iniciar: 3.0.

7.2.1. Documentación.

En esta sección del escrito se procederá a describir la documentación correspondiente al procedimiento que fue necesario llevar a cabo para cumplir los objetivos propuestos, además de los acontecimientos que ocurrieron durante su ejecución, teniendo en cuenta la modalidad del proyecto.

Se inicia el proceso de crear los parámetros que se utilizaran en la estandarización de las técnicas a utilizar para el nombramiento de funcionalidades y estructuras de código fuente, así como la asignación de los nombres de tablas de las bases de datos; tal como se menciona en la sección 7.2.2 del documento actual. A continuación se relaciona una tabla donde se especifica el tipo de entidad y el procedimiento o técnica a seguir, de manera que puedan derivarse de allí los formatos usados para dicho fin.

Tabla 2. Designación de procesos para la estandarización de entidades

ENTIDAD (Componente)	PROCEDIMIENTO (Técnica)	EJEMPLO
Tabla de base de datos	Se nombrará anteponiéndole las tres letras fijas 'tbl' a una descripción breve del nombre que representa los datos contenidos en la tabla. Siendo ésta una sola cadena de caracteres ininterrumpidos de donde se le extraen las vocales intermedias y solo se mantienen visibles las últimas vocales sonoras de cada palabra.	Una tabla que almacena nombres de proveedores debe titularse : tblNmbrePrvdres
Campo perteneciente a una tabla de base de datos	Su nombre está dado por la tabla a la cual pertenece de manera que al igual que el estándar anterior, al nombre del campo se le anteponen las tres primeras letras del nombre de la tabla sin incluir vocales intermedias y siguiendo los mismos pasos del proceso anteriormente expuesto. Los tipos de datos de un campo se asignaran de acuerdo a las búsquedas que se generen a partir de los valores que allí se almacenen; es decir, de ser necesario debe almacenarse un	El campo que almacena el código de un proveedor en una tabla llamada Nombre de proveedores (que según el esquema debe ser llamada: tblNmbrePrvdres) será llamado: nmbcdgo. Si al tener almacenada la secuencia de un código que me representa un proveedor, debo concatenarlo con un

	campo cuyo contenido es numérico como varchar() siempre y cuando su proceso de búsqueda asuma este valor como una cadena de texto.	identificador de tipo texto que identifica el almacén de dicho proveedor es necesario almacenar el campo código como texto para poder así formar una cadena de tipo texto.
Formularios	En la creación de un nuevo formulario le debe ser asignado el prefijo 'Frm' a su nombre elaborado con los mismos parámetros de los puntos anteriores. Es decir, eliminando vocales intermedias y conservando únicamente la última vocal sonora siendo todo el nombre una sola palabra y cuyas mayúsculas identifican cada una de ellas.	En el momento de crear el formulario registros de clientes se le debe nombrar como 'FrmRgstrosClntes'
Imágenes	El nombre asignado cumple los parámetros anteriormente expuestos y se debe anteponer el prefijo 'Img', en caso de ser icono se adiciona 'lco' al principio de la cadena.	Si se espera incluir una imagen de un gráfico estadístico se nombra a la imagen como: 'ImgGrfcoEstdstco' y/o por otra parte se desea adicionar un icono a los recursos de registros tendrá como nombre: 'lcoRgstros'
Cuadros de texto	Su nombre está dado por la sigla 'Txt' más el nombre corto según los mismos parámetros enunciados inicialmente.	Para nombrar un cuadro de texto que recibe información del nombre de un usuario se debe asignarle 'TxtNmbreUsuario' como nombre.

Combos de búsqueda	Su nombre está dado por la sigla 'Cmb' más el nombre corto según los mismos parámetros enunciados inicialmente.	El nombre de un combo que contiene un listado de productos será 'CmbPrdctos'
Cuadros de chequeo	Su nombre está dado por la sigla 'Chk' más el nombre corto según los mismos parámetros enunciados inicialmente.	Un cuadro que permite seleccionar si un campo ventas es visible o no será llamado: 'ChkVntas'
Radio Button u opciones de selección	Su nombre está dado por la sigla 'Opc' más el nombre corto según los mismos parámetros enunciados inicialmente.	Un botón de opción que me permite elegir configuraciones tendrá como nombre 'OpcCnfgrcnes'
Etiqueta de texto	Su nombre está dado por la sigla 'Lbl' más el nombre corto según los mismos parámetros enunciados inicialmente.	Una etiqueta que muestra el título de un informe debe ser llamada: 'LblTtloInfrme'
Botón	Su nombre está dado por la sigla 'Btn' más el nombre corto según los mismos parámetros enunciados inicialmente.	Un botón que confirma los cambios hechos en el formulario es posible nombrarlo como: 'BtnAcptar'
Informe o lista imprimible	Su nombre está dado por la sigla 'Inf' más el nombre corto según los mismos parámetros enunciados inicialmente.	El documento generado después de llamar un listado de ventas y se guarda como informe se nombra: 'InfLstdoVntas'
Reporte de procesos – No imprimible	Su nombre está dado por la sigla 'Rpt' más el nombre corto según los mismos parámetros enunciados inicialmente.	Luego de realizar el ajuste de liquidación de nómina y generar un resultado parcial no imprimible del mismo, se nombrara: 'RptLqdcionNmna'

Clases Adicionales	Todo fragmento de código anexo que no se relacione con una interfaz o formulario debe ser nombrado en inglés bajo los criterios anteriormente mencionados y debe ser descrito por medio de comentarios en el interior de su contenido a fin de facilitar la implementación en otros módulos.	Para la clase que controla la conectividad con la base de datos de proveedores se debe contemplar el nombre: 'PrvdersCnction'
Funciones y procedimientos	Todo procedimiento que retorne un valor o no conservara su nombre en español donde describa puntual pero reducidamente las acciones que realiza; si posee más de una palabra, éstas serán separadas por un guion bajo. Si requiere argumentos de entrada, estos serán de nombres cortos y sin caracteres adicionales.	Una función que solicita una entrada numérica y luego de validar si es positivo o no, arroja un resultado de verificación booleano de la siguiente manera: 'Es_positivo(Byref num As Double) As Boolean'

Con las bases estipuladas se procede a iniciar la elaboración del documento que se deja como constancia de compromiso de entrega a EVACC Ltda, el cual contiene implícitamente los requisitos mencionados a continuación y se puede encontrar en el ANEXO C adjunto a este documento.

- § Árbol de menús
- § Disposición de pantallas
- § Modelo de informes
- § Diseño de datos
- § Requerimientos establecidos
- § Especificaciones de programación.

De esta manera se inicia la labor de reingeniería que se pretende fundamentar con el proceso total de la práctica; en este punto se definen aspectos del software como el diseño del nuevo logo y el nombre que se le asigna a la aplicación: CONTAR v3.0 lo cual se describe con más detalle en la sección 7.4.2 de este documento.

Para la planeación de ejecución y montaje del aplicativo se establecen los requisitos mínimos de hardware y software a tener en cuenta en el momento de la instalación y posterior ejecución: la plataforma en la cual se ha desarrollado este programa es cien por ciento compatible con sistemas operativos Windows en versiones superiores a Windows XP sin diferencia de arquitectura de procesadores y cuyas velocidades no bajen de los 1.2 GigaHertz, bajo la premisa de contar con una conexión a internet en algunos casos; respecto a la memoria se sugiere un valor no menor a un (1) gigabyte y un espacio en disco de por lo menos 100 megabytes, esto con el fin de dedicar recursos suficientes para la ejecución sin contratiempos.

7.2.2. Diseño de la base de datos

Para el diseño de la base de datos con la cual se trabajó durante la fase actual del proyecto fue necesario tener en cuenta los módulos que se pretendían desarrollar durante la ejecución total del proyecto mismo, de manera que en este aspecto se contó con el apoyo del área de desarrolladores de EVACC Ltda y con la asesoría del contador encargado; en el presente subcapítulo se describe enteramente lo relacionado a lo que concierne con los objetivos de la práctica.

El gestor de base de datos usado es PostgreSQL, una herramienta de distribución libre bajo la licencia BSD¹⁷, con un potente sistema de indexación de tablas y una interfaz altamente amigable. Para su desarrollo fue necesaria la intervención del tutor quien dirige este trabajo y quien de igual manera dispuso que éste fuera el diseño final de la misma.

Inicialmente y como se comenta en la sección 7.2.1 del presente escrito, se implementó una nueva estandarización para aplicar en la creación de la nueva base de datos; En la TABLA01 se detallan las convenciones a utilizar para el nombramiento de las tablas y la asignación de los estándares en su respectivo nombre.

La técnica que se establece para el nombramiento de las tablas consiste en nombrar las mismas según un proceso de eliminación de vocales intermedias excepto la última pero anteponiéndole el prefijo 'tbl' a su nombre, es decir, si tomamos como ejemplo la tabla 'Personas', según las reglas anteriormente expuestas debe ser nombrada como Prsnas y adicionándole el prefijo tbl obtenemos como resultado 'tblPrsnas'; esto nos arroja una convención estándar que es ajustable a muchas de las funcionalidades que más adelante se verán en este documento.

Bajo el mismo principio, los campos pertenecientes a cada una de las tablas, deberían ser nombrados con el mismo estándar que dichas tablas, con la salvedad de anteponer (en algunos casos puntuales) las tres primeras letras

¹⁷ Licencia de software libre permisiva; para más información consultar : http://es.wikipedia.org/wiki/Licencia_BSD

que identificasen a la tabla; un ejemplo para esto sería: si uno de los campos pertenecientes a la tabla 'Personas' contiene el nombre de dicha persona, lo ideal sería nombrar el campo como 'PerNombre', pero solo cuando sea necesario.¹⁸ En caso contrario su identificador sería 'Nombre'.

Tabla 03: Nombre de las tablas pertenecientes a la nueva base de datos del sistema según nueva convención.

NOMBRE	CONVENCIÓN
Comprobante	cmprbnte
Detalle de Comprobante	dtllecmpbnte
Periodo	tblprdo
Detalle de bancos	tblbncosdtlle
Concepto de cuenta	tblcncptocnta
Concepto DIAN	tblcncptodian
Cuentas pérdidas y ganancias	tblcntaspyg
Centros de costo	tblcntrocsto
Certificado retención en la fuente	tblcrtfcdortfnte
Detalle certificado rete fuente	tblcrtfcdortfntedtlle
Definición	tbldfncion
Formato DIAN	tblfrmtodian
Puc	puc ¹⁹
Periodos	tblprdos
Comprobante anulado	dtllecmpbnteanlido
Organización	tblorgnzcion
Proceso	tblprcso
Saldos de terceros	tblsldostrcro
Saldos de terceros al cierre	tblsldostrcrocre
Tipo usuario	tbltpousrio
Tipo proceso de usuario	tbltpousrioprcso
Usuario	tblusrio
Plantilla comprobante	tblplntllacmpbnte
Plantilla detalle comprobante	tblplntllacmpbntedtlle
Tipo comprobante	tbltpocmpbnte
Tipo de cuenta	tbltpocnta
Bancos	tblbncos
Tipo procesos	tbltpoprcso

¹⁸ Si existe otra tabla que se llame usuarios y contenga un campo llamado nombre (haciendo referencia al nombre del usuario) puede generar conflictos cuando se tengan que realizar procesos de consulta entre las dos tablas.

¹⁹ A la tabla central del plan único de cuentas no se le adicionan prefijos para hacer más fácil su lectura/escritura.

En el anexo D se detalla una lista con las tablas pertenecientes a la base de datos, donde se especifica los campos que cada una de ellas contiene y de igual manera el tipo de datos que componen cada una, junto con la capacidad máxima de posiciones a almacenar para los casos en que a dichos campos se les asigna la propiedad 'tipo' como 'CharacterVargin()'; esto con el fin de corroborar el procedimiento anteriormente comentado.

Para la obtención del listado de tablas se empleó sintaxis SQL, ya que la herramienta presta este tipo de ayudas y se explotó el recurso que permite acceder a los esquemas de las bases de datos; la instrucción utilizada para tal fin se detalla a continuación:

```
“SELECT table_name AS Nombre_Tabla , column_name AS Nombre_Columna , is_nullable AS NULO, data_type AS Tipo, character_maximum_length AS MAX FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME IN (SELECT table_name FROM information_schema.tables where table_type =‘BASE TABLE’ AND table_schema=‘public’)”
```

Paralelamente se desarrolló un esquema de relaciones entre estas tablas donde se hallaron los enlaces que lo permitían, de igual forma se encuentran normalizadas hasta la tercera forma normal en caso de ser posible y se les asignan sus respectivas claves primarias de acuerdo al campo de control de cada una. En el ANEXO E se encuentra el diagrama que describe lo anteriormente mencionado.

7.3. Módulos a desarrollar

La interfaz consta de diversos módulos de acuerdo a los movimientos contables que se planean hacer, de manera que el alcance de este proyecto se estructuró al diseño, implementación y puesta en marcha de determinadas unidades que se encuentran contenidas dentro del mismo.

De igual forma se expone la documentación elaborada para la estandarización de procesos. Y los criterios tenidos en cuenta al momento del desarrollo de los contenidos que aquí se mencionan.

7.3.1. Descripción.

- § **Plan Único de Cuentas:** Sección para ingresar, eliminar, editar y buscar cuentas pertenecientes al plan de cuentas interno de la empresa.
- § **Tipo Comprobante:** Sección para ingresar, eliminar editar, buscar los tipos de comprobantes contables que se manejan en la empresa además de la impresión de un listado total de los mismos.

- § **Terceros:** Acceso a la aplicación que controla la base de datos con información sobre terceros.
- § **Centros de Costo:** Sección para ingresar, eliminar, editar y buscar centros de costo.
- § **Bancos:** Sección para ingresar, eliminar, editar y buscar información relacionada con los bancos con los que se trabajan en la empresa y además incluye una sección para ingresar, eliminar, editar y buscar las cuentas correspondientes a cada uno de los bancos.
- § **Gestión de comprobantes:** Una de las unidades modulares más importantes en la administración contable; sección donde se pueden realizar movimientos y transacciones con los comprobantes.
- § **Diseño de informes lineales:** componente adicional con finalidad informativa, donde se muestran listados sin una lista subyacente dependiente (teniendo en cuenta que los reportes de más de un nivel se encuentran dentro del marco de un nuevo proyecto consecuente al presente).
- § **Utilidades:** Sección para la configuración y/o establecimiento de las rutas de acceso al aplicativo que controla la base de datos de terceros y junto a esto las cadenas de conexión a las bases de datos.

7.3.2. Objetivos.

A continuación se presenta una tabla con los objetivos del sistema para cada uno de los módulos mencionados anteriormente, donde se especifica la finalidad de cada uno de ellos, lo que se logra con la implementación del mismo dentro de la herramienta, además de sus características esenciales..

Tabla 04 Objetivos de módulos desarrollados

Módulo	Objetivos
Plan Único de Cuentas	<ul style="list-style-type: none"> § Gestionar los procesos asociados a las cuentas y subcuentas que componen el plan de cuentas de acuerdo a lo que estipulan las normas contables internas, las que a su vez se rigen por la normatividad colombiana. § Se ajusta al esquema del plan único de cuentas nacional²⁰
Tipo Comprobante	<ul style="list-style-type: none"> § Administrar los diferentes tipos de comprobantes con los cuales se realizan las transacciones diarias. § Asigna la secuencia de cada uno de los tipos.

²⁰ Más información en: www.puc.com.co

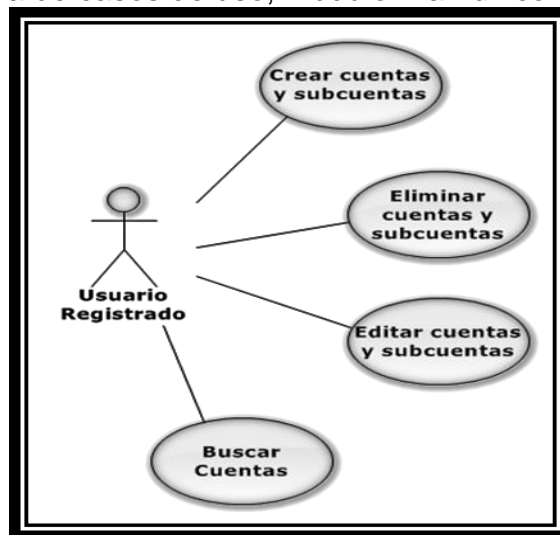
Terceros	<ul style="list-style-type: none"> § Módulo de interrelación entre aplicativos. § Acceso a ejecutables externos.
Centros de costo	<ul style="list-style-type: none"> § Es el encargado de administrar las dependencias organizativas de la empresa que añaden beneficio o utilidad indirectamente. § Pretende ser la herramienta para una distribución por sectores y obtener así una contabilidad más organizada
Bancos	<ul style="list-style-type: none"> § Unidad de manejo de las entidades bancarias y por ende las cuentas que a los mismos les han sido asignadas § Establece la numeración que corresponde a la expedición de los cheques de acuerdo al banco.
Gestión de comprobantes	<ul style="list-style-type: none"> § Componente central del ejercicio donde se desarrollan los procesos de actividades con comprobantes y además es posible controlar su estado. § Su control es vital para el manejo de los libros contables (diario, mayor y balances, etc) ya que aquí se gestionan los movimientos unitarios de la empresa.
Diseño de informes lineales	<ul style="list-style-type: none"> § Sección donde predomina el aspecto visual, procesa los datos recibidos de otras dependencias y los lista en pantalla con formatos lineales preestablecidos. § No genera un entorno de interfaz particular sino que por el contrario se busca hacerse cada vez más global y de tal forma encajar en todos los sub-módulos de la aplicación.
Utilidades	<ul style="list-style-type: none"> § Herramienta administrativa con fines de ajustes de configuración que le permita al usuario tener el control de los recursos externos a los que la aplicación pueda acceder y se consideren de dominio público. § Aunque se solicita precaución al usuario en el momento de modificar la información que aquí se maneja, de igual forma se ofrece al usuario posibilidades de verificación de procesos.

7.3.3. Requisitos funcionales y no funcionales

Para la implementación del aplicativo se deben tener en cuenta los procedimientos que cada uno de los módulos deben seguir; para soportar este proceso se generan a continuación los diagramas de casos de usos de acuerdo a los estándares definidos por UML.

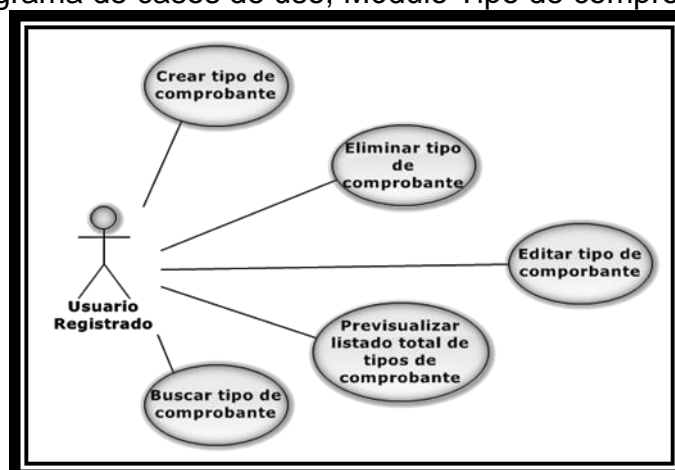
Cabe resaltar que el módulo de control de usuarios y permisos no se encuentra contemplado en el marco del presente proyecto, de manera que se asumen todos los usuarios como un mismo actor al cual se le asignó el nombre de: "Usuario Registrado" y cuenta con accesos a todos los recursos permitidos hasta el momento.

Figura 19. Diagrama de casos de uso, Módulo Plan único de cuentas



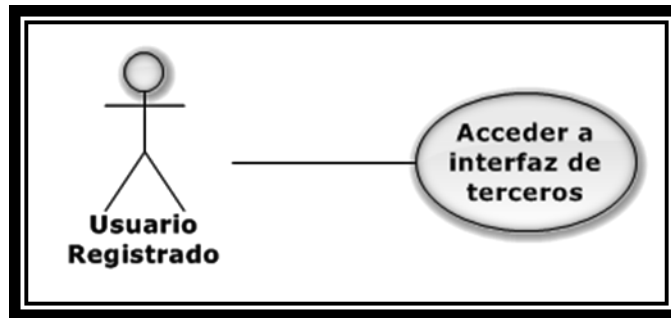
Fuente: El Autor

Figura 20. Diagrama de casos de uso, Módulo Tipo de comprobante



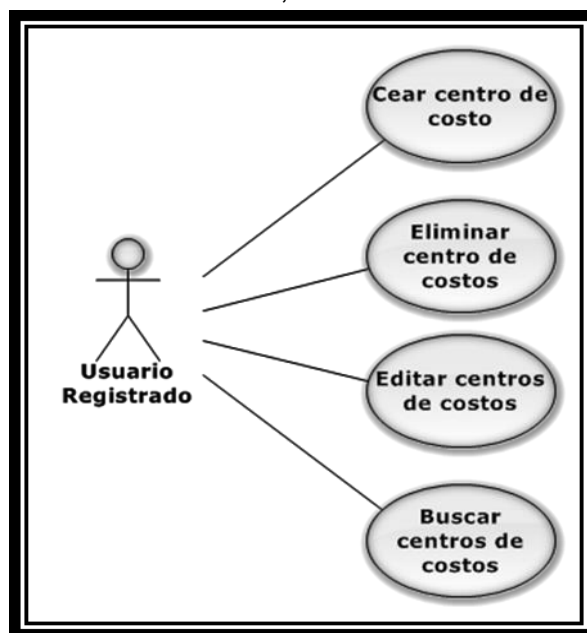
Fuente: El Autor

Figura 21. Diagrama de casos de uso, Módulo Terceros



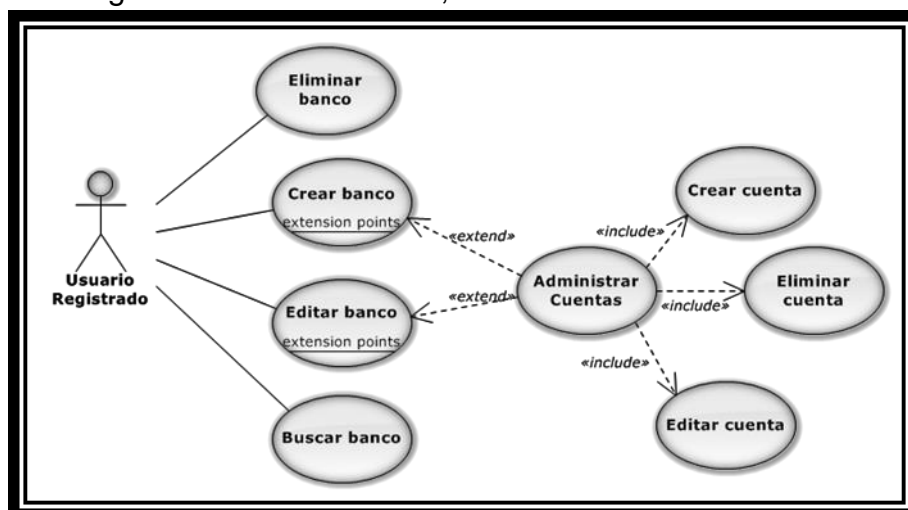
Fuente: El Autor

Figura 22. Diagrama de casos de uso, Módulo Centros de costos



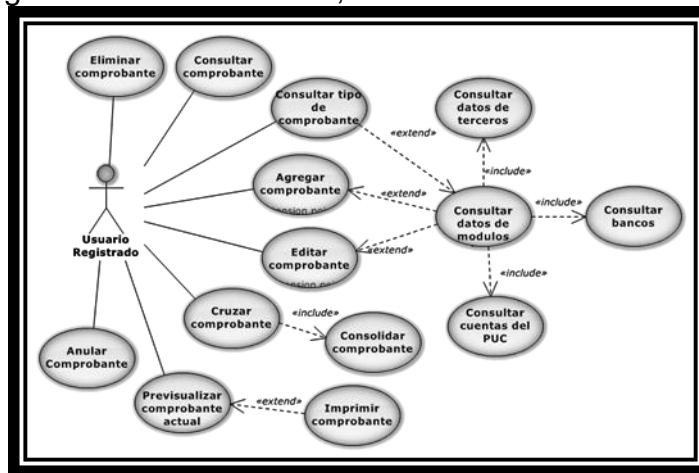
Fuente: El Autor

Figura 23. Diagrama de casos de uso, Módulo Bancos



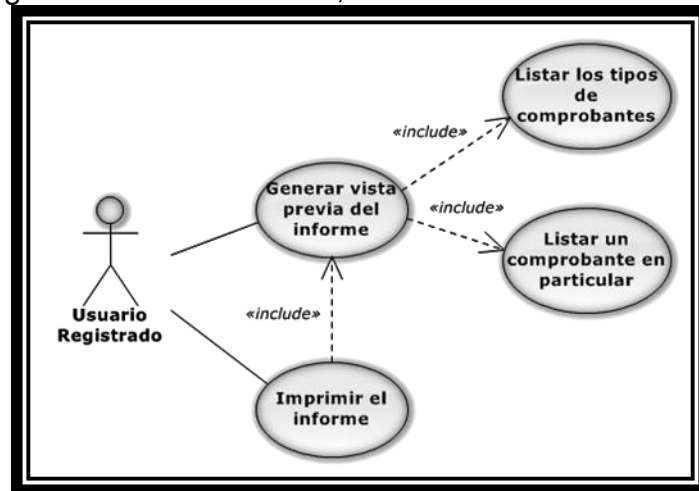
Fuente: El Autor

Figura 24. Diagrama de casos de uso, Módulo Gestión de comprobantes



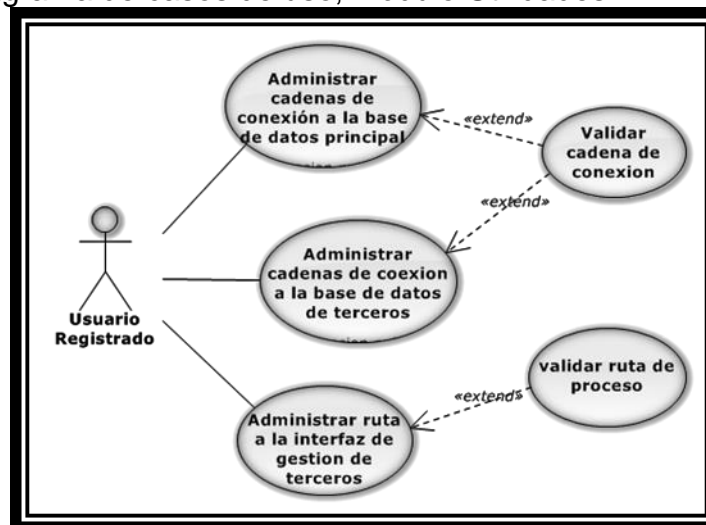
Fuente: El Autor

Figura 25. Diagrama de casos de uso, Módulo Diseño de informes lineales



Fuente: El Autor

Figura 26. Diagrama de casos de uso, Módulo Utilidades



Fuente: El Autor

Con base en los diagramas anteriores es posible extraer un listado acorde a cada módulo de requerimientos funcionales del aplicativo que se da como entregable del proyecto actual; la lista a continuación fue presentada ante el equipo de desarrollo y el asistente tutor acompañante del presente, donde todos estuvieron de acuerdo con que así fuese.

LISTADO DE REQUERIMIENTOS FUNCIONALES DEL APLICATIVO.

- § La carga de la aplicación se hará por medio de un formulario inicial que contenga el nuevo logo de la herramienta.
- § Los módulos se ejecutaran en un formulario principal tipo MDI-Container que facilitara la navegación de formularios y contendrá leyendas informativas de los estados y periodos actuales, así como una barra de iconos de accesos directos a dichos formularios.
- § El usuario estará en la capacidad de crear, modificar y/o eliminar las cuentas y subcuentas del plan único de cuentas de acuerdo a las siguientes restricciones:
 1. Será posible eliminar cuentas que posean subcuentas asignadas bajo una advertencia de confirmación.
 2. No será posible eliminar cuentas que posean registros asociados en el módulo de comprobantes.
 3. La creación de cuentas es entera responsabilidad legal del usuario, teniendo en cuenta que este aspecto es controlado por la parte contable de la empresa.
 4. La estructura de conformación de cuentas se rige al plan único de cuentas nacional, es decir:
 - a. Un (1) dígito para las cuentas de primer nivel
 - b. Dos (2) dígitos para las cuentas de segundo nivel
 - c. Cuatro (4) dígitos para cuentas de tercer nivel
 - d. Seis (6) dígitos para cuentas de cuarto nivel
 - e. Nueve (9) dígitos para cuentas de quinto nivel
 - f. Doce (12) dígitos para cuentas de sexto y último nivel.
 5. La edición de cualquier cuenta solo permitirá la modificación de la cadena de texto que la describe y la exigencia de pago a terceros.
 6. Se tendrá un acceso al formulario patrón de búsqueda con el fin de filtrar/seleccionar los campos requeridos por el usuario.
- § El usuario estará en la capacidad de crear, modificar y/o eliminar los tipos de comprobantes que prefiera además de contar con la posibilidad de generar un listado imprimible y/o previsualizable de todos los tipos de comprobantes existentes de acuerdo a las siguientes restricciones:
 - 1 No es posible eliminar tipos de comprobantes que tengan comprobantes elaborados bajo el mismo.

- 2 Se tendrá un acceso al formulario patrón de búsqueda con el fin de filtrar/seleccionar los campos requeridos por el usuario.
- § El usuario estará en la capacidad de acceder a la interfaz de terceros.
- § El usuario estará en la capacidad de crear, modificar y/o eliminar los centros de costo con los cuales cuenta la empresa de acuerdo a las siguientes restricciones:
- 3 No es posible eliminar centros de costo que se encuentren relacionados en algún detalle de algún comprobante.
 - 4 La edición de cualquier centro de costo solo permitirá la modificación de la cadena de texto que lo describe.
- § El usuario estará en la capacidad de crear, modificar y/o eliminar la información relacionada a los bancos con los cuales trabaja la empresa y de igual manera contará con la opción de manipular las cuentas asociadas a los mismos de acuerdo a las siguientes restricciones:
- 5 No es posible eliminar bancos que se encuentren relacionados en detalles de comprobantes.
 - 6 Será posible modificar tanto el código como la descripción de un banco siempre y cuando el código nuevo no se encuentre asignado a otro banco ni esté relacionado en los detalles de algún comprobante.
 - 7 No existe un límite de cuentas a asignar a cada banco
 - 8 Para cada banco se contará con la opción de crear, modificar o eliminar sus cuentas si:
 - a La cuenta no existe
 - b La cuenta posee un registro asociado en el PUC
 - c La cuenta trae consigo una secuencia.
 - 9 Se tendrá acceso al formulario patrón de búsqueda con el fin de filtrar/seleccionar los campos requeridos por el usuario.
- § El usuario contará con la posibilidad de crear, modificar, eliminar, cruzar y/o anular comprobantes además de contar con la posibilidad de generar un informe pre visualizado del mismo y una sección de búsqueda de acuerdo a las siguientes restricciones:
- 10 En la creación de un comprobante nuevo el usuario contará con la posibilidad de consultar los tipos de datos existentes.
 - 11 La interfaz asignará el número de secuencia siguiente pero le permitirá al usuario su modificación.
 - 12 La fecha de creación del comprobante se encontrará dentro del periodo actual a menos que el usuario decida lo contrario.
 - 13 Para la edición del detalle se tendrán en cuenta aspectos como:
 - a El usuario tendrá la oportunidad de digitar en cada registro la cuenta que desee o seleccionarla de un filtro de búsqueda.

- b Los valores numéricos se mostraran / almacenaran con formato doble, dos unidades decimales y separados por miles.
 - c Las cuentas del PUC seleccionadas que requieran terceros permitirán la visualización de una nueva columna para su digitación.
 - d La consolidación de cada detalle es requerida para su almacenaje y/o edición.
 - e El usuario contara con la posibilidad de eliminar registros; lo cual invalidara cualquier consolidación pre – hecha.
 - f No será posible adicionar al detalle cuentas por parte del usuario que contengan menos de 6 dígitos.
 - g Cada registro contara alternativamente con una etiqueta de descripción de acuerdo al movimiento.
 - h La consolidación se encargara de mayorizar cuentas hacia los débitos o créditos.
- 14 En la modificación de terceros asociados al comprobante se tendrán en cuenta los siguientes aspectos:
- a Los bancos se consultaran en un formulario patrón de búsqueda para tal fin.
 - b La identificación y descripción del tercero se consultara de la base de datos de terceros y será accesible por medio de un formulario patrón de búsqueda.
- 15 El usuario contara con la posibilidad de importar detalles de otros comprobantes según lo requiera.
- 16 El comprobante será anulado solamente si cumple en su totalidad las siguientes condiciones:
- a El comprobante es el último de la secuencia en su tipo.
 - b El comprobante se encuentra abierto.
- 17 El comprobante será cruzado solamente si cumple en su totalidad las siguientes condiciones:
- a El comprobante no se encuentra anulado
 - b El comprobante posee sumas iguales; es decir: el valor del débito es igual al valor del crédito.
 - c El comprobante se encuentra consolidado
 - d El comprobante pertenece al periodo activo.
 - e El comprobante se encuentra abierto
- 18 El comprobante será eliminado solamente cuando es el último en la secuencia de su tipo.
- 19 El usuario contara con la posibilidad de generar un documento para visualizar en pantalla y opcionalmente imprimirlo que

contenga todos los datos correspondientes al comprobante en pantalla.

20 Se tendrá acceso al formulario patrón de búsqueda con el fin de filtrar/seleccionar los campos requeridos por el usuario. Dicho formulario contara con las siguientes especificaciones:

- a Opción de filtrar comprobantes por tipo
- b Opción de filtrar comprobantes en un rango de fechas no cerrado
- c Búsqueda de comprobante por número.

§ El usuario contara con la capacidad de generar informes a fin de pre visualizarlos y según se requiera navegar entre las distintas partes del documento, con la facultad de controlar su nivel de zoom y poder enviar dicho documento a la impresora de selección.

§ El usuario estará en la capacidad de modificar la información relacionada a las conexiones externas a bases de datos principales y de terceros, así como la ruta para el acceso a la interfaz de manejo de terceros, generando la opción de validar la cadena que se selecciona.

Para su implementación en la interfaz se diseñó un diagrama organizacional donde se puede ver con más detalle la distribución de los mismos dentro del conjunto de menús de la barra de comandos; disponible en el ANEXO F

REQUERIMIENTOS NO FUNCIONALES

Además de los detalles mencionados anteriormente, se buscan algunas características en particular para el montaje y ejecución del sistema que permitan el correcto funcionamiento del mismo. Estas características además de ser externas son inherentes al sistema, se convierten en fundamentales y se limitan al entorno en el cual se mueve la herramienta.

Para una ejecución exitosa del aplicativo se recomienda que sea instalado en un sistema operativo Windows® con procesador de alta capacidad y un amplio espacio en memoria virtual y moderada capacidad en disco duro; junto a esto se espera que la herramienta se acompañe de una impresora comercial, una pantalla de resolución media y que se cuente con la capacitación adecuada para el dominio de la misma.

Por parte del usuario se espera que se cuente con conocimientos en herramientas informáticas desarrolladas en plataformas .NET, además de conocimientos en procesos contables y dominio de la metodología que se implementa por parte de EVACC Ltda, a lo que se le conoce como reglas del negocio.

7.4. Ejecución y codificación

En este párrafo del documento se exponen las métricas de código que se utilizaron para la elaboración de la herramienta software, además del proceso necesario para la ejecución de la implantación de la herramienta, junto con las pruebas de montaje realizadas y las mejoras visuales que fueron requeridas por parte del equipo desarrollador de la práctica.

7.4.1. Refactorización de código e implementación de algoritmos

Siguiendo los estándares anteriormente planteados se procede a diseñar las bases de la herramienta software contar v3.0 en colaboración con el equipo de desarrollo de la empresa EVACC Ltda. A continuación se describe con detalle la metodología usada para su elaboración.



CREACIÓN DE CLASES ANEXAS:

Para los métodos implementados de consulta, modificación, eliminación e inserción de registros de las tablas de la base de datos se creó una clase independiente dentro del mismo proyecto que fue nombrada como 'ConexToDB' la cual recibe como argumento un identificador entero corto que identifica la base de datos con la cual se desea trabajar; es decir: cero (0) para realizar procedimientos en PostgreSQL y uno (1) para trabajar con MS Access teniendo en cuenta que estas son las dos bases de datos con las que se relaciona la interfaz actualmente. Se puntualiza en la relación que se tiene entre dicho identificador y la idea de asociar en un futuro más gestores de bases de datos al proyecto.

La clase mencionada consta de tres procedimientos principales asignados con permisos públicos debido a su necesidad de ser invocada desde clases no amigas.

1. **Constructor** (Llamado mediante la palabra reservada NEW): Es el encargado de crear una nueva instancia de la clase. Recibe como argumento:
 - a. El identificador Id de tipo entero corto que indica la base de datos con la cual se desea trabajar.
2. **Obtener_Tabla**: función que retorna una estructura 'DataTable' en caso de obtener un resultado valido, y la estructura vacía 'Nothing' en caso de no obtener un resultado valido. Recibe como argumentos:
 - a. La cadena de texto *InstruccionSQL* de tipo String que contiene la instrucción en lenguaje SQL con la cual se busca acceder a los datos.

- b. La cadena de texto *NombreTabla* de tipo string (Opcional) que sirve para asignarle el nombre a la tabla resultante.
 - c. El valor *IndiceLlavePrimaria* de tipo entero (Opcional) que indica la posición de la columna que contiene la llave primaria; cuando recibe menos uno (-1), la tabla resultante no contiene llave primaria. Por defecto es 0.
3. **Modificar_Registro:** función de tipo booleano que retorna verdadero en caso de obtener un resultado positivo en el momento de modificar cualquier registro de la tabla y falso en caso contrario. Recibe como argumento:
- a. La cadena de texto *InstruccionSQL* de tipo String que contiene la instrucción en lenguaje SQL con la cual se busca modificar los datos.
4. **Testeo:** función de tipo booleano que retorna un valor verdadero en caso de obtener un resultado positivo al momento de realizar una prueba de conectividad con el gestor de base de datos y falso en caso contrario. Esta función no requiere de argumentos.

En esta instancia es necesario importar las librerías dinámicas Npgsql y System.Data.OleDb por medio de la palabra reservada 'IMPORTS'.

De la misma forma, para la implementación del proceso de pre visualización se diseñó una clase que permite manipular los objetos necesarios para tal fin, requiere que inicialmente se implemente la instrucción 'OPTION Explicit ON' que obliga al compilador a declarar las variables antes de su uso y no permite la conversión de cadenas de diferente tipo; además es necesario invocar a las librerías System.Drawing y System.Drawing.Printing, esta clase se encuentra altamente ligada a la barra de herramientas de tipo ToolStrip denominada BndImprimir del formulario principal tipo container, la cual contiene los botones base para desplazamiento entre páginas, un selector de zoom, un botón de impresión y un botón de cerrado del documento. Para ser llamada es necesario que se le asignen valores a sus miembros tales como: documento a imprimir, tabla de la cual se obtienen los datos, control de vista preliminar que contiene las configuraciones de impresión del documento y las fuentes que se requieran para mostrar. Se aclara que el documento que recibe debe tener asignado el 'Handler' (procedimiento) de impresión desde la clase en la cual es creado. En su mayoría los eventos de los que consta esta clase son adheridos a los botones de BndImprimir en tiempo de ejecución a través del procedimiento llamado mostrar, el cual es el único de carácter público y es el encargado de llevar a cabo la labor de cargar el documento en el container principal y asignar los eventos mencionados a los botones.

ESTANDARIZACIÓN DE CÓDIGO EN LOS FORMULARIOS.

La codificación para los formularios depende de la función de cada uno pero se procuró trabajar bajo los estándares que el equipo de desarrollo de la empresa sugería. Teniendo en cuenta que en su mayoría, cada formulario proviene de una tabla en la base de datos, se inicia todo formulario cargando una entidad tipo DataTable llamada TablaDatos y designada con su respectivo constructor, de manera que al iniciar la carga de cualquier formulario se procede a crear una instancia 'co' de la clase ConexToDB() y llenar los datos de la tabla con los resultados que arroja el procedimiento co.Obtener_Tabla que recibe la consulta SQL: "SELECT * FROM" como parámetro.

Adicionalmente se implementa (en la mayoría de los casos) un procedimiento llamado 'mostrar_registro' que recibe como argumentos un identificador id de tipo entero que representara el índice del registro a mostrar de acuerdo al contenido de la tabla TablaDatos la cual contiene previamente cargada la información. Este procedimiento es llamado cada vez que se desea visualizar un conjunto de registros en pantalla y se guía por el contenido de la llave primaria, la cual ha sido designada desde el procedimiento de su creación gracias al tercer parámetro que se le envía a la función mostrar_registro.

El manejo por medio de entidades de este tipo hace que su despliegue sea mucho más limpio y más ágil, un algoritmo de ese estilo permite obtener acceso a los datos de una forma más dinámica, de manera que el paso a la base de datos solamente se realiza cada vez que se carga el formulario específico; lo desfavorable del método es la recurrencia, es decir: cuando el aplicativo está siendo manipulado en varias máquinas cliente y la base de datos se encuentra en un solo servidor lo cual, por accesos temporalmente simultáneos, hace que se corra el riesgo de una interpretación de datos equivocada; por ejemplo: la carga de un formulario F (por consiguiente de la TablaDatos F) en el computador A se realiza en un tiempo denominado 't1A' y su permanencia en el sistema con F cargado es de un lapso de n minutos, si en un computador B igualmente se carga el formulario F (con su respectiva TablaDatos F) y se le realizan cambios a la tabla en un periodo menor a esos n minutos luego del tiempo 't1A', sucederá que los cambios que se realicen en la computadora A no estarán contenidos en la tabla de B hasta el momento en que B recargue de nuevo el formulario F ni los cambios que se realicen en B serán visibles para A hasta que este último no recargue su formulario F; esto conllevaría a un sinnúmero de inconvenientes con el manejo de registros pero como solución a esto el equipo organizacional de la empresa establece una directiva como plan de contingencia para reducir este tipo de errores. De

acuerdo a las máquinas que se encuentren operando con la aplicación se designaran labores específicas a cada una de ellas según lo determine el director del montaje, es decir: si en la computadora A y en la computadora B se tiene implementada la herramienta, se pueden asignar labores de procesos con comprobantes a la computadora A y diligenciamiento de libros contables a la computadora B de manera que la posibilidad de generarse un error por acceso recurrente se vea disminuida al mínimo valor posible, convirtiéndose así esta opción en una regla del negocio.

Para el estado de edición del formulario se hace uso de un identificador numérico denominado 'sw' cuyo tipo de datos es Integer y cumple con las siguientes características.

Cuando se inicia cualquier formulario sw se encuentra con un valor de cero (0), en caso de realizar una acción de adición de registros, esta variable toma el valor de uno (1) y de la misma forma, si se entra en un modo de edición, esta variable asumirá el valor de dos (2); por lo tanto es esta variable la que indica el tipo de permisos que se le pueden brindar al usuario según la acción que seleccione.

Los permisos de una acción están dados así:

- § Cuando se está ingresando un nuevo registro es imposible eliminar y/o editar, solo se tiene la posibilidad de guardar los cambios o deshacer en caso de retornar al estado anterior sin guardar.
- § Cuando se está editando es imposible agregar un nuevo registro, ni tampoco se puede eliminar el registro actual. Las opciones disponibles para el usuario en ese momento se limitan a guardar los cambios o deshacer los cambios y retornar sin guardar.
- § Los procesos de guardar y deshacer solo se encuentran activos cuando el usuario decide añadir o editar registros.

El identificador sw también funciona como una bandera de paso para determinar la acción a ejecutar en caso de solicitar el guardado de cambios hechos en el formulario. Si se guardan cambios mientras sw tiene un valor de (1) se procede a realizar la creación de un nuevo registro pero si el valor de esta variable es uno (1) se procede a verificar el registro sobre el cual se están haciendo los cambios y seguido esto aplicarle las transformaciones que el cliente le ha hecho. Cabe aclarar que luego de finalizar el proceso de guardado con éxito se debe restablecer el valor de sw a cero (0).

El identificador sw puede tomar valores superiores a 2 cuando se desean implementar estados con permisos diferentes a los ya mencionados; esto dependerá de los ajustes y/o requerimientos que el programador tenga como base para desarrollar.

Pensando en la comodidad del cliente se han establecido una convención de teclas para la adición, eliminación, modificación, impresión búsqueda y navegación entre registros (cuando estas opciones se encuentren disponibles), esta notación se detalla a continuación:

Tabla 5. Relación de métodos abreviados para funciones

Icono	Acción	Conjunto de teclas (método abreviado)
	Asignar posición que se desea visualizar en pantalla (cuadro de texto)	F2 (tecla de función)
	Visualizar el último registro de la tabla	Fin
	Visualizar el siguiente registro en relación al que actualmente se muestra en pantalla	Av Pág.
	Visualizar el registro anterior en relación al que actualmente se muestra en pantalla	Re Pág.
	Visualizar el primer registro de la tabla	Inicio
	Guardar los cambios realizados, ya sea por ingreso de nuevo registro o por edición de un registro existente.	Enter
	Añadir un nuevo conjunto de campos o registro a la tabla.	Ctrl + A
	Eliminar el registro que actualmente se ve en pantalla.	Suprimir
	Editar el registro que actualmente se ven en pantalla.	Ctrl + E
	Deshacer la acción solicitada, ya sea adicionar nuevo registro o editar registro actual.	Ctrl + D
	Imprimir la selección (De acuerdo al formulario en el cual se encuentre)	Ctrl + P
	Invocar al formulario patrón de búsqueda de acuerdo al formulario actual.	Ctrl + B
	Cerrar el formulario actual.	Escape

²¹ Aplica también para el botón de cierre incluido en cualquier formulario, propio de las formas del sistema operativo Windows®

Nota: los iconos de navegación se inhabilitan según la posición que se encuentre visualizada en ese momento en el formulario; es decir: cuando se visualiza el último registro se inhabilitaran los iconos correspondientes a siguiente y último registro de la misma manera que se inhabilitan los iconos de primer y anterior registro cuando se encuentra en pantalla la información correspondiente al primer registro de la tabla.

Adicionalmente se especifica que los campos correspondientes a valores ingresados por el usuario se encuentran validados en la medida de lo posible, teniendo en cuenta que existen valores que no corresponden a una máscara de entrada previsible.

7.4.2. Proceso de diseño

En el aspecto visual también se le realizaron mejoras representativas, los diseños fueron elaborados bajo la guía del tutor asignado. Se tomaron como referencias algunos estándares pero predominó la idea particular del grupo de desarrollo de la empresa.

Toda la esquematización y uso de imágenes se encuentran cubiertas por la licencia de libre distribución GPL y algunos por la licencia Creative Commons que permite su uso de manera comercial sin restricciones. De la misma manera los iconos que fueron objeto de aplicación para esta labor se encuentran cubiertos bajo las licencias anteriormente mencionadas.

En cuanto a la ubicación de los comandos se implanto un estándar que determina la posición de los mismos, comúnmente dispuestos sobre una barra de herramientas acoplada en la parte inferior del formulario y extendidos en todo su ancho.

De preferencia se espera que los botones de navegación (si son requeridos) se encuentren ubicados de izquierda a derecha y en su respectivo orden, así:

1. Primer registro
2. Registro anterior
3. Separador
4. Cuadro de texto para posición
5. Etiqueta de contador
6. separador
7. Siguiente registro
8. Último registro.
9. Separador

De aquí en adelante los botones de comando se enlazan al margen derecho del formulario y su ubicación de derecha a izquierda en su respectivo orden debe ser:

10. Búsqueda
11. Pre visualizar para imprimir
12. Deshacer
13. Guardar
14. Separador
15. Editar
16. Eliminar
17. Guardar
18. Separador
19. Acciones adicionales.

Esto le permite al usuario final contar siempre con los mismos comandos en las mismas posiciones y de esta forma no se verá obligado a reacomodarse por cada nuevo formulario al que acceda.

En el formulario principal también se ha dispuesto de una barra con iconos que permiten un acceso de forma más directa a cada uno de los módulos más importantes²² y de un tamaño mayor para su realce visual.

ELABORACIÓN DEL NUEVO LOGO.

Para culminar con las mejoras en el diseño se determinó la creación de un nuevo logotipo que hiciera parte de la identificación visual de la herramienta software pero que a su vez fuese posible la implementación en toda documentación que al mismo se refiriera. A continuación se presenta el diseño elaborado, el cual fue aprobado por la junta directiva de EVACC Ltda y aprobado para su uso.

Figura 27. Nuevo logo de la herramienta software



Fuente: El Autor

²² La disposición de los módulos seleccionados para asignarles un acceso directo fue determinado por el grupo de desarrollo en base a reglas del negocio.

El nuevo diseño hace referencia en forma abstracta a un ábaco, como símbolo de la primera herramienta utilizada para 'contar' (conservando el ideal inicial) pero que en este momento se encuentra cubierto por una nueva ideología, por nuevos principios y destacado por los diversos ambientes en los cuales puede desarrollarse el aplicativo, sus colores predominante son el azul en tres tonalidades haciendo alusión al color que identifica a la herramienta de desarrollo Visual Studio. Al logotipo lo acompaña el nombre de la herramienta desarrollada y junto a ella la versión a implementar que fue elaborado con un tipo de letra cuyo impacto fuera mayor y diera una noción de modernismo.

7.5. Entrega de la herramienta software

Luego de culminar el proceso de codificación y estructuración visual de la interfaz de la herramienta se procede a llevar a cabo la documentación respectiva para la entrega del producto desarrollado por medio de esta práctica. A continuación se describe el proceso que fue necesario realizar para verificar el correcto funcionamiento del aplicativo y posterior entrega ante la empresa interesada EVACC Ltda.

7.5.1. Acoplamiento del proyecto de ejecución

Una herramienta bastante útil que ofrece Visual Studio es la generación de proyectos que, de forma paralela, llevan a cabo la labor de publicar todo el contenido de otros proyectos dentro de un solo paquete instalable. Dentro de las propiedades de este tipo de proyectos se encuentra la posibilidad de realizar algunos ajustes que se llevarán a cabo en el momento de instalar el proyecto principal al cual se encienden asociados; dichos ajustes son, en su mayoría, facilidades que se le pueden brindar al cliente cuando la herramienta sea instalada en la computadora destino.

Entre las opciones que el proyecto instalador ofrece se encuentran:

- § Establecimiento de los nombres internos del aplicativo
- § Configuraciones de nombres de propietarios y/o desarrolladores
- § Verificaciones de versiones anteriormente instaladas
- § Directorio de almacenaje para ficheros propios de la aplicación.
- § Modificaciones a realizar en el editor de registros de la maquina cliente
- § Generación del paquete instalable (extensión *.msi) propio del sistema operativo Windows para paquetes re-codificables.
- § Establecer nuevas asociaciones de archivo para la maquina destino.
- § Adjuntar archivos necesarios para la instalación y/o ejecución del aplicativo.
- § Otras funcionalidades específicas.

De igual manera es posible controlar la creación de carpetas y manejo de archivos dentro de la maquina cliente al momento de instalar; genera un archivo final con extensión *.msi que funciona de la mano con Windows installer y le da el trato propio de cualquier proyecto de tal envergadura.

El proyecto instalador se encarga de compilar el código fuente y empaquetarlo así como lo hace con los recursos agregados al mismo tales como imágenes, íconos, textos, etc, con el fin de hacer más fácil el traslado de la aplicación.

7.5.2. Pruebas de montaje

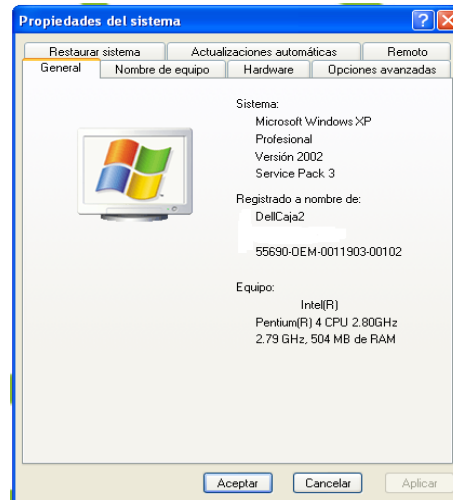
Luego de contar con el paquete de instalación completo se procedió a realizar las pruebas de montaje en dos máquinas con diferentes características internas a fin de evaluar el rendimiento, los ajustes visuales y documentar los posibles contratiempos que llegasen a ocurrir durante dicho proceso.

Dentro del conjunto de herramientas necesarias para realizar estas pruebas se encuentran:

- § Paquete instalable (*.msi) generado por el proyecto instalador de la herramienta visual studio.
- § Paquete ejecutable del gestor de base de datos PostgreSQL en su versión más reciente cuyo uso es libre.
- § Archivo backup generado por PostgreSQL como respaldo de la creación de la base de datos.
- § Cadenas de conexión para su inclusión por medio del menú utilidades de la aplicación.

Como primera opción se realizó la prueba en un equipo con un procesador marca Intel modelo Pentium 4, con una velocidad de 2.8 Ghz, y 500Mb de capacidad en la memoria volátil. El equipo cuenta con un sistema operativo Windows® XP en su versión Profesional 2002 service pack 3. A continuación se presenta una imagen tomada de las características mencionadas.

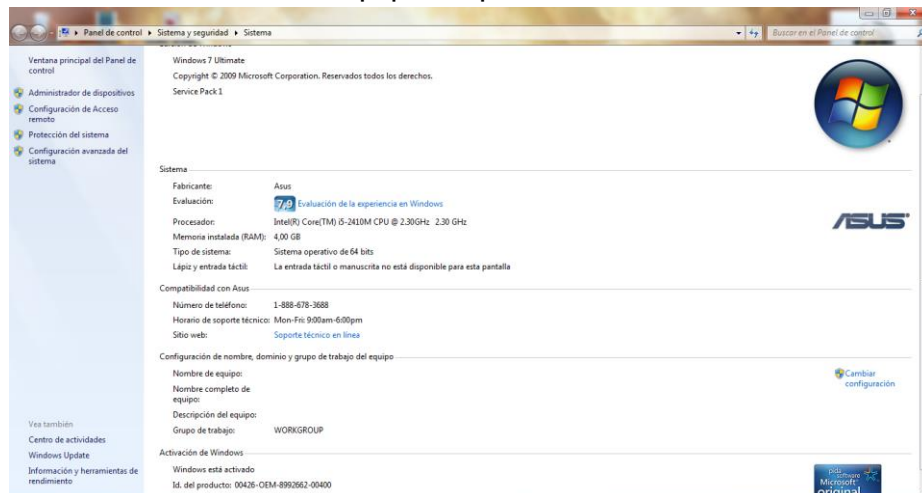
Figura 28. Características del equipo de prueba de instalación numero 1



Fuente: EL Autor

En la segunda oportunidad se procedió a montar el aplicativo en un computador con procesador Intel core i5 2410M con una velocidad de 2.3 GigaHertz y 4 gigabytes asignadas en la memoria RAM. El sistema operativo con el cual cuenta es un Windows 7 versión Ultimate y trabaja bajo una arquitectura de 64 bits. La siguiente es una figura que muestra las características mencionadas y fue tomada como captura de pantalla.

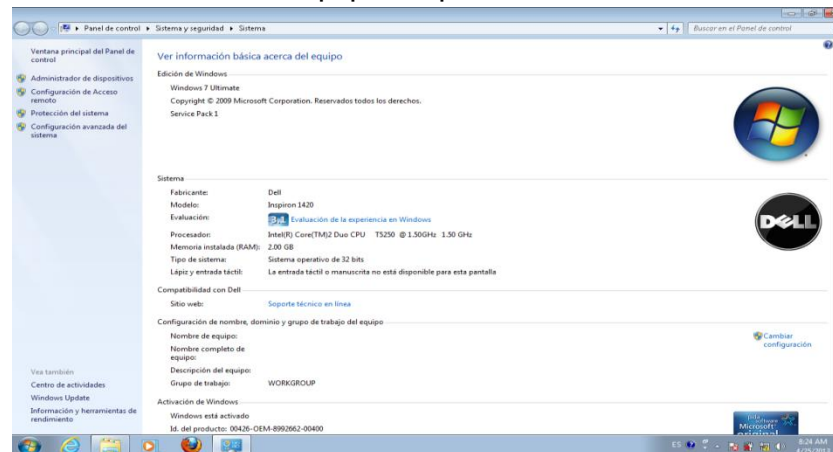
Figura 29. Características del equipo de prueba de instalación numero 2



Fuente: El Autor

Para el tercer y último ejercicio de instalación se contó con una computadora cuyo procesador es de marca Intel referencia Core 2 Duo T5250 que cuenta con una velocidad de 1.5 GigaHertz y dispone de 2 gigabytes para la memoria de acceso aleatorio RAM. Esta máquina se encuentra funcionando con Windows Seven en su versión Ultimate como sistema operativo y trabaja bajo una arquitectura de 32 bits. La siguiente es una imagen correspondiente a la ventana de información del sistema de dicho computador.

Figura 30. Características del equipo de prueba de instalación numero 3



Fuente: El Autor

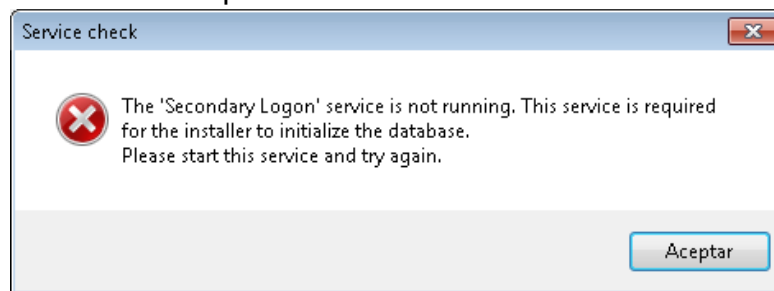
La implementación del sistema se efectuó de la misma manera para los tres equipos y el procedimiento fue exitoso en todos los casos. En el ANEXO G se detalla paso a paso el procedimiento realizado en cada una de las máquinas.

7.5.3. Inconvenientes presentados durante las pruebas

Tomamos como base que el procedimiento de instalación se ejecuta siempre de la misma manera, independientemente de cual sea la plataforma en la cual se esté implementando la herramienta; es decir: siempre se siguen los mismos pasos de instalación sin importar las características del sistema operativo. De tal manera que siguiendo dichos pasos, se han encontrado cierto tipo de conflictos que no permitieron el desarrollo normal de la instalación, a continuación se mencionarán los detalles concernientes a las dificultades mencionadas.

En las máquinas cuyo sistema operativo es Windows 7, teniendo en cuenta que el instalador de PostgreSQL accede al servicio de inicio de sesión secundario de Windows, y que este servicio comúnmente no se encuentra activo, fue necesario recurrir al administrador de servicios de Windows; al cual es posible acceder por medio del Panel de control, luego seleccionando herramientas administrativas y después la opción de Servicios; donde se mostrara en pantalla una lista con todos los servicios del sistema operativo; allí se selecciona el ítem que tiene como nombre: "Inicio de sesión secundario" ó "Secondary Logon" según el idioma en el cual se tenga configurado el sistema operativo. Este servicio es necesario mantenerlo como iniciado con el fin de permitir el acceso del gestor de base de datos a los procesos de inicio de sesión del sistema. La instalación se vio interrumpida por el estado actual de este servicio, de manera que el asistente arroja un mensaje de advertencia del cual se ha tomado una captura de pantalla y se expone a continuación.

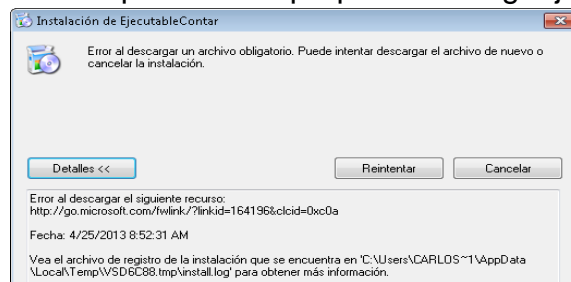
Figura 31. Informe de error por inicio de sesión secundario



Fuente: El Autor

Teniendo en cuenta que en ninguna de las maquinas donde se realizó la prueba existían aplicativos desarrollados con el Framework 4.0, tales computadoras no contenían el paquete de lenguaje para idioma español. Este paquete fue necesario descargarlo desde su página oficial²³ de Microsoft® y seguir las instrucciones que allí se mencionan. Debido a la imperiosa necesidad de contar con este archivo, la instalación de la aplicación se vio interrumpida arrojando un mensaje de alerta; a continuación se expone una captura de pantalla con el mensaje generado.

Figura 32. Informe de error por falta de paquete de lenguaje

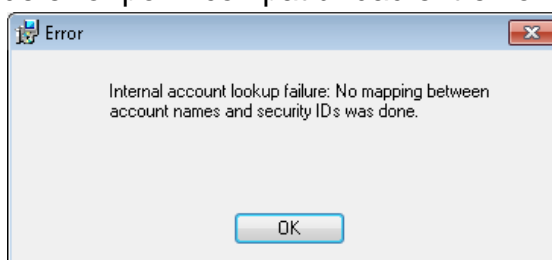


Fuente: El Autor

En uno de los equipos con sistema operativo Windows 7 se presentó un error particular; la sesión de usuario mediante la cual se estaba instalado el proyecto contenía un nombre con más de 15 caracteres, en el momento en el cual el instalador de PostgreSQL se encontraba creando la nueva cuenta de usuario, la instalación se vio interrumpida por un mensaje que anunciaba una incompatibilidad entre los nombres de las cuentas. La información al respecto contenida en la web no se presenta de forma muy clara, debido a que tal error no es muy común entre este tipo de instalaciones, de manera que con base en información recopilada de algunos foros de usuarios se procedió a modificar el nombre de usuario de la sesión actual por uno más corto, lo cual permitió continuar con éxito el proceso de instalación. A continuación se presenta una imagen como captura del mensaje desplegado.

²³ Información e instalación del paquete de idioma español necesario para sistemas operativos Windows, disponible en: <http://www.microsoft.com/es-es/download/details.aspx?id=23067>

Figura 33. Mensaje de error por incompatibilidad entre nombres de cuentas.



Fuente: El Autor

Seguido esto la instalación concluyó exitosamente.

7.5.4. Documentación de entrega

Luego de realizar las respectivas pruebas de montaje se procedió a tener una junta con el grupo de desarrollo de EVACC Ltda, donde se acordaron los puntos a tener en cuenta para la elaboración del acta de entrega del aplicativo desarrollado; dicho documento fue elaborado con el fin de soportar físicamente el proceso realizado, de manera que se entrega como adjunto al presente escrito y a la documentación anexa correspondiente.

Cabe aclarar que el producto entregado corresponde únicamente a los alcances del presente proyecto, de tal manera que la empresa es entera propietaria del contenido elaborado; por lo cual, no se estipulan tipos ni cantidades de licencias de funcionamiento.

El formato con el cual se hace entrega de la herramienta software se encuentra en el ANEXO H del presente escrito. Allí se especifican los ítems que hacen parte del disco donde se han compactado todos los archivos necesarios para su correcto funcionamiento.

El personal de la empresa se encuentra a gusto con los resultados obtenidos y en muestra de su aprobación proceden a diligenciar el formato de culminación de la práctica realizada con su compañía. Las relaciones comerciales, laborales y personales quedan abiertas como gesto de gratitud recíproco y con miras a un futuro próspero para cada una de las partes.

CONCLUSIONES

Gracias al ejercicio de llevar a cabo esta labor se han adquirido importantes experiencias; el objetivo de las mismas se ha culminado con gran esfuerzo y dedicación, por lo cual se deduce lo siguiente:

- § El diseño y puesta en marcha de un software contable genera un impacto de gran solidez y confianza a cualquier compañía; su implementación como parte vital del sistema de la información de la empresa se ve reflejada en los avances que registran con relación a sus modelos internos organizacionales.
- § Una estandarización de procesos representa el marco fundamental para un seguimiento continuo y progresivo del presente proyecto; se convierte en el primer paso para nuevos integrantes del equipo y deja su huella en la estructuración del mismo.
- § La colaboración y apoyo en la renovación del sistema contable de EVACC Ltda genera en la calidad del código fuente implementado un importante progreso para futuras tecnologías, es acoplable y moldeable a plataformas de nueva generación; su estructura modular le facilitará la actualización mientras se siga versionando la herramienta.
- § La innovación en materia de diseño visual le permitirá a la herramienta software Contar 3.0 disponer de un espacio en las herramientas a la vanguardia en este campo; lo anterior podrá verse reflejado en la valorización económica que se presentara cuando el proyecto total sea culminado y el producto salga al mercado.
- § El mayor objetivo de la presente labor fue centrada pensando en el cliente final, su satisfacción es nuestro principal reto y eso conlleva a un trabajo organizado y un desarrollo altamente efectivo en materia de adaptabilidad y un amplio factor de ajuste a los requerimientos del cliente.
- § La tecnología .NET es una excelente herramienta para la construcción de aplicaciones software; facilita las labores de programación, se acopla a las metodologías de desarrollo de aplicaciones actuales y cuenta con una amplia gama de información en plataformas web tales como MSDN o Microsoft su página comercial, además de ser la más comentada en foros de ayuda de su tipo.

- § El motor de base de datos PostgreSQL es una aplicación excelente para el manejo de los datos, su solidez y versatilidad hacen de la herramienta un gestor altamente confiable y moldeable pero sin perder las características esenciales de seguridad y rapidez; el manejo de los datos en este ambiente se torna agradable y sencillo, genera confianza tanto en sus transacciones como en el almacenaje de su contenido.

- § La experiencia de contar con un equipo de trabajo tan sólido como el de EVACC Ltda ha hecho posible ampliar mis conocimientos en materias de trato con el cliente, dominio de áreas contables, antes desconocidas y me ha formado como profesional en un ambiente empresarial donde se requiere esfuerzo, constancia y disciplina para sacar adelante los objetivos que se proponen sin dejar de lado un excelente compañerismo y un espectacular ambiente laboral.

- § Es cada vez más notoria la influencia que tienen los avances tecnológicos en la administración y organización del dinero, por lo tanto es evidente la necesidad de enfatizar cada día más en la seguridad de las herramientas software, temas como la contabilidad hacen parte de uno de los ítems más valiosos para una compañía; EVACC Ltda ha puesto gran parte de su información en manos de esta labor y de igual manera ha depositado su confianza en las manos del encargado de este proyecto, gestión que compromete a guardar estrictamente las normas internas de dominio de la información de la empresa.

- § La selección de una adecuada metodología para el desarrollo de proyectos permite una excelente administración de los recursos necesarios para la elaboración del mismo, la gestión del tiempo, del dinero y de los alcances se hacen más fácil cuando se tiene instaurado un cronograma y se sigue un plan preestablecido; a pesar de los inconvenientes presentados durante el ejercicio de esta labor fue posible una entrega oportuna y sin repercusiones negativas para ninguna de las partes.

RECOMENDACIONES

El proceso del trabajo en equipo junto con el personal de desarrollo de EVACC Ltda fue una experiencia de grandes aportes personales; para posibles mejoras en la continuación del proyecto que se inicia con esta práctica sugiero los siguientes procesos:

- § Proseguir con la intención de culminar el objetivo general de finalizar el sistema de información contable, no olvidar los ideales de atención al cliente ni la gestión humana que tanto requiere el área de desarrollo.
- § Continuar bajo los estándares establecidos para la codificación de algoritmos y el uso de metodologías ágiles para llevar a cabo los proyectos de herramientas software.
- § Bajo la importancia de un apoyo adicional, se hace necesaria la implementación de publicidad referente al producto terminado por medios audiovisuales y de igual manera vía web para que se incremente la tasa de clientes potenciales y de esta formar generar mayores ingresos a la empresa.
- § Incursionar en el mundo de las aplicaciones móviles, teniendo como base la cantidad incremental de dispositivos inteligentes que actualmente se encuentran en circulación, es posible abrir mercados apuntando hacia ese horizonte y poder así ofrecer a los clientes finales un 'plus' que lleve al producto más allá de lo esperado.
- § Simultáneamente al punto anterior, se sugiere contemplar la posibilidad de ofrecer al cliente final un almacenamiento de sus datos por medio del Cloud Computing a fin de convertir la herramienta en un producto menos estático y más adaptable a los usuarios que requieren desplazarse continuamente.
- § Mantener la herramienta actualizada en materia de aspectos contables; soportándose en la variabilidad de las leyes y normativas colombianas que rigen este concepto, a fin de llevar un control riguroso de los procesos necesarios para la ejecución de cada acción y no poner en riesgo aspectos de tipo legal.

BIBLIOGRAFÍA

BALENA, Francesco. Programación avanzada con Microsoft Visual Basic .NET, Mc Graw Hill Interamericana, 2003.

BLANCO, Luis Miguel. Programación en Visual Basic .NET, Grupo EIDOS 2002.

CATALDI, Zulma. Metodología de diseño, desarrollo y evaluación de software educativo. Tesis de Magister en informática. (Versión Resumida), Facultad de Informática, UNLP, 2002.

DATE, C.J. Introducción a los sistemas de bases de datos, Séptima edición, Prentice Hall, 2001.

GASTEIZ, Vitoria. Estándares de desarrollo de software, consultado en:

http://www.industria.ejgv.euskadi.net/r44-ikus0004/es/contenidos/informacion/anexos_pbt_ejie/es_0214/adjuntos/Estandares%20de%20desarrollo%20software.v2.0.pdf

GOMEZ, Ronald. Ciclo de vida de un sistema de información, Disponible en:

<http://mundoinformatico2012.wordpress.com/ciclo-de-vida-de-un-sistema-informatico/>

LOCKHART, Thomas. PostgreSQL Programmer's Guide. The PostgreSQL Global Development Group, USA 1998.

MOMJIAN, Bruce. PostgreSQL Programmer's Guide. The PostgreSQL Global Development Group, USA 1998.

ORTEGA PAREDES, José Gabriel. Sinopsis del libro: CONTABILIDAD. Paradigma de reconstrucción a través del giro informático disponible en:

<https://www.morebooks.de/store/es/book/contabilidad/isbn/978-3-8484-5034-3>

Referenciado desde: <http://www.cienciadelcontabilidad.com/>

PRESSMAN, Roger S. Ingeniería del Software – un enfoque práctico, Edición No 5, España, Mc Graw Hill, 2002.

Reglamentación contabilidad colombiana, Decreto 2649 de 1993 consultada en: <http://www.gerencie.com/principios-de-contabilidad-en-colombia.html>

ANEXOS

ANEXO A

EVALUACION DEL RENDIMIENTO DE LA APLICACIÓN CONTAR EN SU VERSION ANTERIOR

Evaluación realizada con la herramienta “Análisis de Rendimiento” de Visual Studio 2010

Soportado en la funcionalidad que presta la herramienta “Análisis de rendimiento”²⁴ se presenta a continuación un esquema gráfico del desempeño de la maquina en el momento de la ejecución de determinadas tareas; aquí se presenta un paralelo que indica el comportamiento del reloj del procesador durante cada una de ellas, de tal manera que el primer cuadro represente la acción realizada por la herramienta en su versión anterior y el segundo cuadro represente la misma acción pero realizada por la herramienta en la versión 3.0, la cual es objeto de estudio en el proyecto actual.

En cada uno de los reportes se puede observar la gráfica de rendimiento del procesador donde se le asigna al eje horizontal el tiempo de uso del mismo medido en segundos, mientras que sobre el eje vertical se encuentra una escala de 0 a 100 que indica el porcentaje que está siendo usado en cada instante de tiempo. La metodología básica usada en este tipo de informes es la selección de muestras recolectadas de un llamado a la pila de instrucciones enviadas hacia el procesador de la maquina por la aplicación; de tal forma que dentro de los informes se pueden observar los porcentajes de las muestras que llaman a las funciones secundarias (Inclusivas) y de las que no lo hacen (Exclusivas)²⁵. La diferenciación entre aplicaciones se puede ver en cada informe por las funciones a las que tiene acceso; es decir: para la aplicación contar en su versión anterior no es posible acceder más que a las librerías DLL que la componen, en contraparte, a la aplicación en su versión actual le es posible acceder a todas las funciones disponibles por estar en proceso de edición.

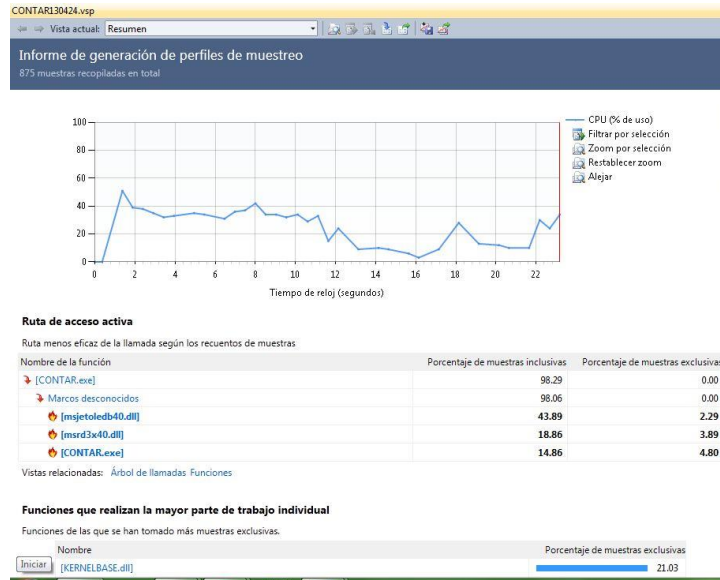
Para este proceso de análisis se tomaron muestras aleatorias de tiempo donde cada aplicación fue medida realizando las mismas acciones; en cada uno de los detalles se describen las acciones que se realizaron para cada muestra.

²⁴ Para más información e interpretación de resultados dirigirse a: <http://msdn.microsoft.com/es-es/magazine/cc337887.aspx>

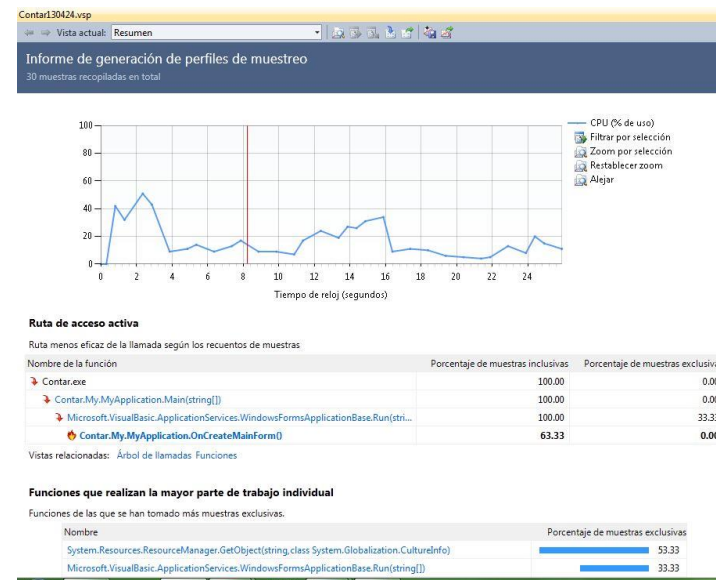
²⁵ Información sobre los datos de muestreo en: [http://msdn.microsoft.com/es-co/library/ms242753\(v=vs.90\).aspx](http://msdn.microsoft.com/es-co/library/ms242753(v=vs.90).aspx)

Muestra Número 1 Ejecución inicial de la aplicación

Aplicación en la versión anterior (23 seg)



Aplicación en la versión 3.0 (26 seg)



Actividad: Ejecución de la aplicación, despliegue del formulario inicial, cierre.

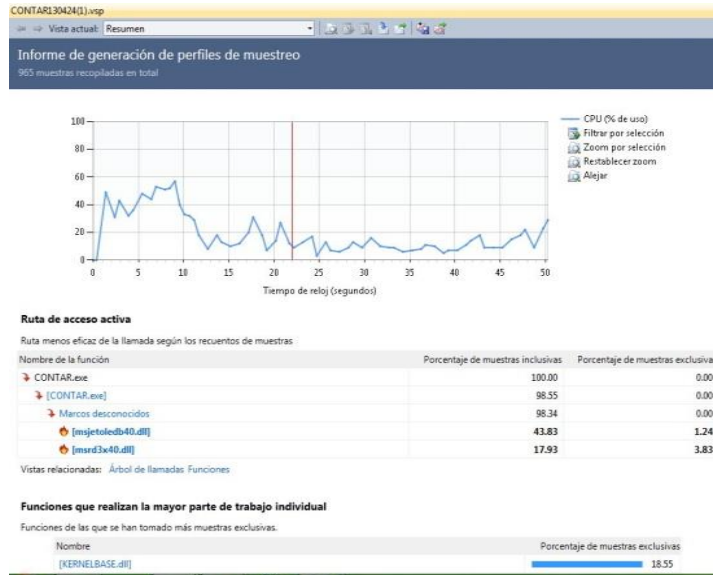
Análisis: es posible ver que luego de la normalización (4 segundos aproximadamente) la aplicación en desarrollo tiende a consumir menos recursos de procesador, mientras que la aplicación anterior tiene su pico más bajo alrededor de los 16 segundos y su consumo promedio se eleva por encima del 15% mientras que 'Contar 3.0' permanece relativamente estable por debajo del 15%.

Resultado: en la partida inicial, la versión 3.0 del aplicativo consume menos recursos de procesador.

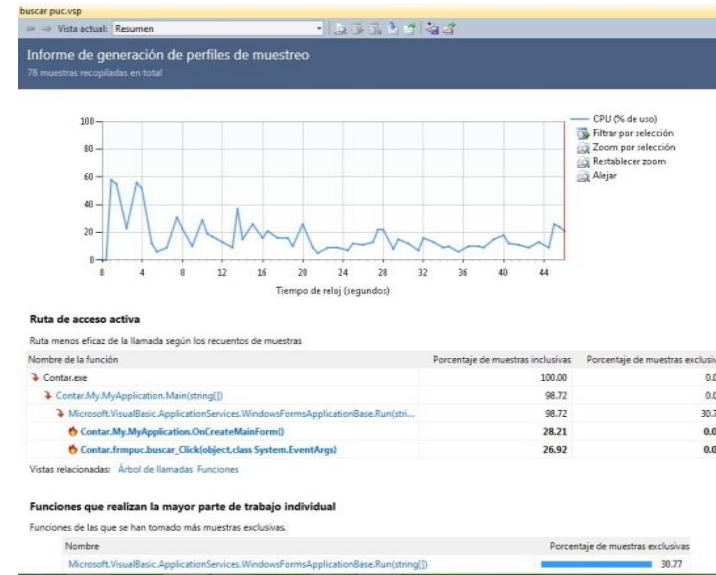
Muestra Número 2

Apertura del formulario del Plan único de cuentas

Aplicación en la versión anterior (50 seg)



Aplicación en la versión 3.0 (46 seg)



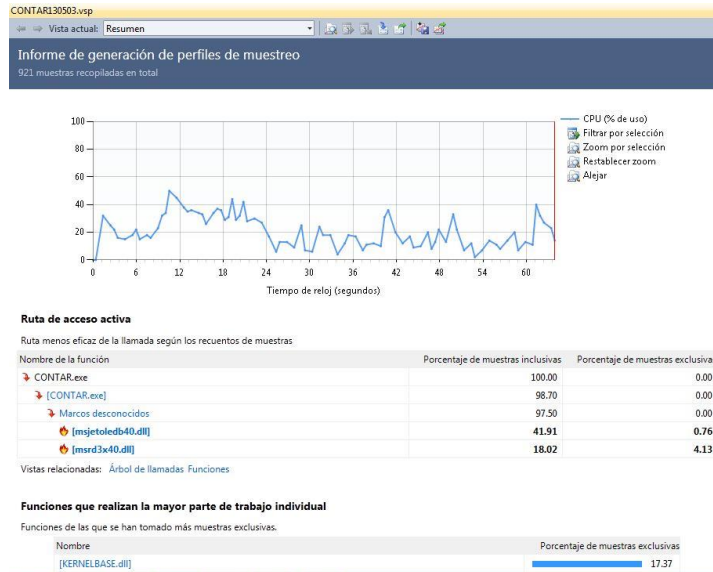
Actividad: Ejecución de la aplicación, despliegue del formulario inicial, solicitud del formulario del plan único de cuentas, consulta de una cuenta, desplazamiento dentro del árbol, cierre del formulario puc y cierre de la aplicación.

Análisis: las dos aplicaciones tienden a una normalización que promedia el 10% pero la versión anterior exige más esfuerzo al momento de la consulta, mientras que la nueva versión muestra picos muy pronunciados pero de corto tiempo, lo cual indica que la nueva algoritmia permite un esfuerzo inicial moderado (mientras se accede a la base de datos) pero luego gestiona los procesos con más rapidez y estabilidad.

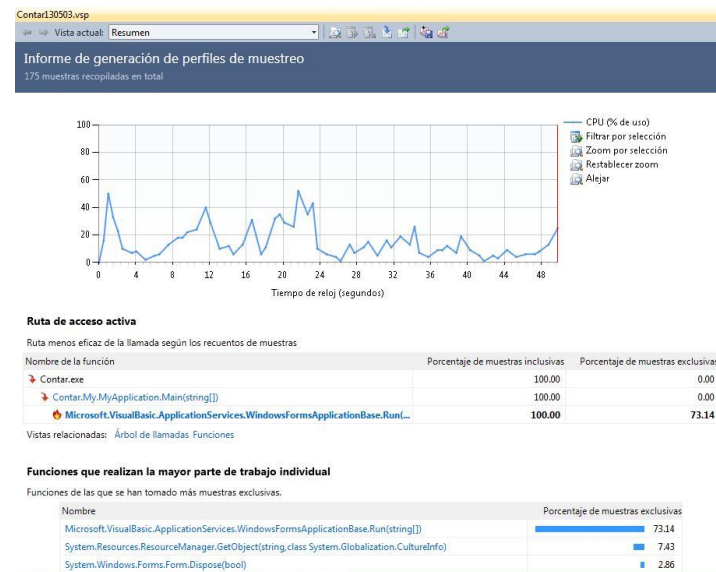
Resultado: la algoritmia utilizada en la nueva versión implica el mismo esfuerzo que la versión anterior pero en un periodo más corto de tiempo, las búsquedas se realizan con más fluidez en la nueva plataforma y se estabilizan con tendencia similar.

Muestra Número 3 Apertura de múltiples formularios simultáneamente.

Aplicación en la versión anterior (64 seg)



Aplicación en la versión 3.0 (50 seg)



Actividad: Ejecución de la aplicación, despliegue de formularios de puc, centros de costo, bancos y tipos de comprobantes de manera simultánea, cierre de cada formulario, cierre de la aplicación.

Análisis: el comportamiento de apertura de formularios simultáneamente genera un mayor consumo de recursos en la nueva versión, pero los picos presentados en la gráfica luego de los 24 segundos demuestran la estabilidad del sistema que brinda la algoritmia implementada en contar 3.0 contra las variaciones que se ven en la gráfica de la versión anterior.

Resultado: el consumo de recursos es máximo cuando se despliega cada formulario, pero la normalización del sistema es más notoria en la nueva plataforma; de igual manera se presentan puntos de cero (0) consumo con mayor frecuencia en la situación de la nueva implementación.

ANEXO B

LISTADO DE TABLAS EXISTENTES EN LA BASE DE DATOS DE LA HERRAMIENTA CONTAR EN SU VERSION ANTERIOR

Tabla: Blnce

Propiedades

DateCreated:	14/06/2000	LastUpdated:	15/06/2000
OrderByOn:	Falso	RecordCount:	0
Updatable:	Verdadero		

Columnas

Nombre	Tipo	Tamaño
cnta1	Texto	1
cnta2	Texto	2
cnta3	Texto	4
Nmbre	Texto	50
sldo	Moneda	8

Tabla: BlncePrba

Propiedades

DateCreated:	21/05/2000	LastUpdated:	18/10/2003
OrderByOn:	Falso	RecordCount:	0
Updatable:	Verdadero		

Columnas

Nombre	Tipo	Tamaño
Ano	Texto	4
Prdo	Texto	2
PucCdgo	Texto	20
PucNmbre	Texto	50
SldoAntrior	Moneda	8
PucSldoDbtoMesAct	Moneda	8
PucSldoCrdtoMesAct	Moneda	8
NvoSldo	Moneda	8

Tabla: BlnceRep

Propiedades

DateCreated:	01/10/2000	LastUpdated:	01/10/2000
OrderByOn:	Falso	RecordCount:	75
Updatable:	Verdadero		

Columnas

Nombre	Tipo	Tamaño
cnta1	Texto	1
cnta2	Texto	2
cnta3	Texto	4
Nmbre1	Texto	50
Nmbre2	Texto	50
Nmbre3	Texto	50
sldo	Moneda	8
ANO	Texto	4
PRDO	Texto	2
Lynda	Texto	50
vlor	Moneda	8

Tabla: Cmprbnte

Propiedades

DateCreated:	23/04/2000	LastUpdated:	24/04/2013
OrderBy:	Cmprbnte.CmpTpo, Cmprbnte.CmpNmro	OrderByOn:	Verdadero
RecordCount:	38857	Updatable:	Verdadero

Columnas

Nombre	Tipo	Tamaño
CmpTpo	Texto	4
CmpNmro	Texto	8

CmpFcha	Fecha/Hora	8
CmpEntdad	Texto	14
CmpChque	Texto	20
CmpBnco	Texto	20
CmpEstdo	Texto	1
CmpCncpto	Memo	-
CmpDbto	Moneda	8
CmpCrdto	Moneda	8
CmpNmbreEntdad	Texto	50
CmpVlorChque	Moneda	8
CmpCntaBnco	Texto	15
EsCrre	Entero	2
CdgoBnco	Texto	4

Tabla: Cntas

Propiedades

DateCreated: 25/02/2002 LastUpdated: 18/10/2003
 OrderByOn: Verdadero RecordCount: 1833
 Updatable: Verdadero

Columnas

Nombre	Tipo	Tamaño
CntaOld	Texto	20
CntaNew	Texto	20

Tabla: DtleCmprbnte

Propiedades

DateCreated: 23/04/2000 LastUpdated: 24/04/2013
 OrderByOn: Falso RecordCount: 183840
 Updatable: Verdadero

Columnas

Nombre	Tipo	Tamaño
DetTpo	Texto	4
DetNmro	Texto	8
DetCnta	Texto	20
Detscncia	Entero largo	4
DetDscrpcion	Texto	100
DetDbePrcial	Moneda	8
DetHberPrcial	Moneda	8
DetDbe	Moneda	8
DetHber	Moneda	8
DetCncpto	Memo	-
DetNit	Texto	15
CdgoCtroCsto	Entero	2
DetOrden	Entero	2
NmroDcmnto	Texto	20
FchaDcmnto	Fecha/Hora	8
TpoDcmnto	Texto	1

Tabla: DtleCmprbnteANLDO

Propiedades

DateCreated: 27/02/2008 LastUpdated: 27/02/2008
 RecordCount: 5839 Updatable: Verdadero

Columnas

Nombre	Tipo	Tamaño
DetTpo	Texto	4
DetNmro	Texto	8
DetCnta	Texto	20
Detscncia	Entero largo	4
DetDscrpcion	Texto	100
DetDbePrcial	Doble	8
DetHberPrcial	Doble	8
DetDbe	Doble	8
DetHber	Doble	8
DetCncpto	Memo	-
DetNit	Texto	15
CdgoCtroCsto	Entero largo	4
DetOrden	Entero largo	4

Tabla: EqvInciasPuc

Propiedades

DateCreated: 25/02/2002 LastUpdated: 18/10/2003
OrderBy: EqvInciasPuc.PucNmbreNvo OrderByOn: Verdadero
RecordCount: 2715 Updatable: Verdadero

Columnas

	Tipo	Tamaño
Nombre	Texto	20
PucCdgoAnt	Texto	100
PucNmbreAnt	Texto	20
PucCdgoNvo	Texto	100
PucNmbreNvo	Moneda	8
SldoDbto	Moneda	8

Tabla: Errores de conversión

Propiedades

DateCreated: 24/04/2013 LastUpdated: 24/04/2013
RecordCount: 1 Updatable: Verdadero

Columnas

	Tipo	Tamaño
Nombre	Texto	255
Tipo de objeto	Texto	255
Nombre de objeto	Memo	-

Tabla: LbroDrioRep

Propiedades

DateCreated: 05/07/2000 LastUpdated: 05/07/2000
RecordCount: 0 Updatable: Verdadero

Columnas

	Tipo	Tamaño
Nombre	Texto	255
cnta	Fecha/Hora	8
fcha1	Fecha/Hora	8
fcha2	Texto	20
DetCnta	Texto	4
CmpTpo	Texto	8
CmpNmro	Moneda	8
DetDbePrcial	Moneda	8

Tabla: MvmntosRsmen

Propiedades

DateCreated: 09/05/2000 LastUpdated: 19/07/2000
OrderByOn: Falso RecordCount: 0
Updatable: Verdadero

Columnas

	Tipo	Tamaño
Nombre	Texto	4
CmpTpo	Texto	8
CmpNmro	Fecha/Hora	8
CmpFcha	Memo	-
CmpCncpto	Texto	20
DetCnta	Texto	50
DetDscrpcion	Moneda	8
Debe	Moneda	8
Haber	Texto	20

Tabla: Prdos

Propiedades

DateCreated: 21/04/2000 LastUpdated: 18/10/2003
OrderByOn: Falso RecordCount: 77
Updatable: Verdadero

Columnas

	Tipo	Tamaño
Nombre	Texto	4
Ano	Texto	2
Prdo	Texto	1
Estdo	Texto	1

Tabla: Prspsto

Página: 11

Propiedades

DateCreated: 24/10/2000
 OrderByOn: Falso
 Updatable: Verdadero

LastUpdated: 18/10/2003
 RecordCount: 0

Columnas

	Tipo	Tamaño
Nombre	Texto	20
Codigo	Texto	50
EnP	Doble	8
EnR	Doble	8
FebP	Doble	8
FebR	Doble	8
MarP	Doble	8
MarR	Doble	8
AbrP	Doble	8
AbrR	Doble	8
MayP	Doble	8
MayR	Doble	8
JunP	Doble	8
JunR	Doble	8
JulP	Doble	8
JulR	Doble	8
AgoP	Doble	8
AgoR	Doble	8
SepP	Doble	8
SepR	Doble	8
OctP	Doble	8
OctR	Doble	8
NovP	Doble	8
NovR	Doble	8
DicP	Doble	8
DicR	Doble	8

Tabla: Puc

Propiedades

DateCreated: 23/03/2000
 OrderByOn: Falso
 Updatable: Verdadero

LastUpdated: 24/04/2013
 RecordCount: 647

Columnas

	Tipo	Tamaño
Nombre	Texto	20
PucCdgo	Texto	50
PucNmbre	Moneda	8
PucSlidoDbtoMesAnt	Moneda	8
PucSlidoCrdtoMesAnt	Moneda	8
PucSlidoDbtoMesAct	Moneda	8
PucSlidoCrdtoMesAct	Moneda	8
PucAjsteDbtoMesAnt	Moneda	8
PucAjsteCrdtoMesAnt	Moneda	8
PucAjsteDbtoMesAct	Moneda	8
PucAjsteCrdtoMesAct	Moneda	8
PucCntroCstos	Texto	1
PucPgosTrcros	Texto	1
PucAjstes	Texto	1
PucTrcro	Texto	50
ScnciaChque	Entero largo	4
TrcroCrre	Entero largo	4
NitTrcroCrre	Texto	15

Tabla: RprteCrze

Propiedades

DateCreated: 21/05/2000
 OrderByOn: Falso
 Updatable: Verdadero

LastUpdated: 21/05/2000
 RecordCount: 394

Columnas

	Tipo	Tamaño
Nombre	Texto	4
Tpo	Texto	8
Nmro	Fecha/Hora	8
Fcha		

Obsrvcion	Texto	100
-----------	-------	-----

Tabla: SlidoMyresRep

Propiedades

DateCreated:	13/06/2000	LastUpdated:	13/06/2000
OrderByOn:	Falso	RecordCount:	0
Updatable:	Verdadero		

Columnas

Nombre	Tipo	Tamaño
Cnta	Texto	20
Nmbre	Texto	50
SubCnta	Texto	20
SubNmbre	Texto	50
Slido	Moneda	8
Ano	Texto	4
Prdo	Texto	4

Tabla: TblBncos

Propiedades

DateCreated:	27/02/2008	LastUpdated:	27/02/2008
RecordCount:	3	Updatable:	Verdadero

Columnas

Nombre	Tipo	Tamaño
CdgoBnco	Entero largo	4
DscrpcionBnco	Texto	100

Tabla: TblBncosDtlle

Propiedades

DateCreated:	27/02/2008	LastUpdated:	21/02/2009
RecordCount:	6	Updatable:	Verdadero

Columnas

Nombre	Tipo	Tamaño
CdgoBnco	Entero largo	4
NroCnta	Texto	30
PucCdgo	Texto	20
Scncia	Texto	20
CdgoFrmto	Entero largo	4

Tabla: TblCncptoCnta

Propiedades

DateCreated:	02/04/2009	LastUpdated:	24/04/2013
RecordCount:	273	Updatable:	Verdadero

Columnas

Nombre	Tipo	Tamaño
NmroCncpto	Entero largo	4
PucCdgo	Texto	20

Tabla: TblCncptoDian

Propiedades

DateCreated:	02/04/2009	LastUpdated:	24/04/2013
RecordCount:	50	Updatable:	Verdadero

Columnas

Nombre	Tipo	Tamaño
NmroCncpto	Entero largo	4
DscrpcionCncpto	Texto	50
NmroFrmto	Entero largo	4

Tabla: TblCntaInhblltda

Propiedades

DateCreated:	04/10/2003	LastUpdated:	18/10/2003
OrderByOn:	Falso	RecordCount:	0
Updatable:	Verdadero		

Columnas

Nombre	Tipo	Tamaño
PucCdgo	Texto	20
PucNmbre	Texto	50

Tabla: TblCntasPyG

Propiedades

DateCreated: 16/04/2004
OrderByOn: Falso
Updatable: Verdadero

LastUpdated: 16/04/2004
RecordCount: 93

Columnas

	Tipo	Tamaño
Nombre	Entero	2
Orden	Entero	2
Grpo	Entero	2
PucCdgo	Texto	20

Tabla: TblCntrcCsto

Propiedades

DateCreated: 13/11/2007
RecordCount: 7

LastUpdated: 13/11/2007
Updatable: Verdadero

Columnas

	Tipo	Tamaño
Nombre	Entero	2
CdgoCtroCsto	Entero	2
DscrpcionCntrcCsto	Texto	100

Tabla: TblCrtfcdoRtfnte

Propiedades

DateCreated: 20/10/2006
RecordCount: 0

LastUpdated: 24/04/2013
Updatable: Verdadero

Columnas

	Tipo	Tamaño
Nombre	Entero largo	4
CdgoCrtfcdo	Texto	4
AnoGrvble	Texto	100
Prdo	Texto	15
Nit	Texto	100
Nmbre	Moneda	8
Ttal	Texto	100
Cnsgndo	Texto	200
TtalLtras	Texto	200

Tabla: TblCrtfcdoRtfnteDtll

Propiedades

DateCreated: 20/10/2006
RecordCount: 0

LastUpdated: 24/04/2013
Updatable: Verdadero

Columnas

	Tipo	Tamaño
Nombre	Entero largo	4
CdgoCrtfcdo	Texto	100
Cncpto	Doble	8
Prcntje	Moneda	8
Bse	Moneda	8
VlorRtndo	Moneda	8

Tabla: TblDfncion

Propiedades

DateCreated: 01/01/2009
RecordCount: 0

LastUpdated: 01/01/2009
Updatable: Verdadero

Columnas

	Tipo	Tamaño
Nombre	Texto	4
CdgoDef	Texto	20
NmbreDef	Texto	50
DescDef	Texto	1
TpoVlorDef	Texto	50
VlorDef	Texto	50

Tabla: TblFrmtoCheque

Propiedades

DateCreated: 21/02/2009
RecordCount: 1

LastUpdated: 21/02/2009
Updatable: Verdadero

Columnas

	Tipo	Tamaño
Nombre	Entero largo	4
CdgoFrmto	Entero largo	4

Dscrpcion	Texto	100
Obsrvcion	Texto	100

Tabla: TblFrmtoDian

Propiedades

DateCreated: 02/04/2009 LastUpdated: 24/04/2013
RecordCount: 12 Updatable: Verdadero

Columnas

Nombre	Tipo	Tamaño
NmroFrmto	Entero largo	4
DscrpcionFrmto	Texto	50

Tabla: TblOrgnzcion

Propiedades

DateCreated: 17/03/2004 LastUpdated: 01/11/2011
OrderByOn: Falso RecordCount: 1
Updatable: Verdadero

Columnas

Nombre	Tipo	Tamaño
NitOrg	Texto	20
NmbreOrg	Texto	100
Cbza1Org	Texto	200
Cbza2Org	Texto	200
Cbza3Org	Texto	200
Cbza4Org	Texto	200
Pie1Org	Texto	200
Pie2Org	Texto	200
Pie3Org	Texto	200
Pie4Org	Texto	200
RprsntnteOrg	Texto	200
CdlaRprsntnteOrg	Texto	20
DrccionOrg	Texto	100
TlfnOrg	Texto	50
EmailOrg	Texto	200
CdadOrg	Texto	200
FrmRcibiCnfrme	Entero	2
Abvtra	Texto	20
PrfjoCmprbnte	Texto	1
TrcroExtrno	Entero	2
Aux1	Entero	2
Aux2	Entero	2
RtaPrsnas	Texto	255
Frmtochque	Entero largo	4
cntrocsto	Entero largo	4

Tabla: TblPntllacmprbnte

Propiedades

DateCreated: 13/06/2012 LastUpdated: 24/04/2013
RecordCount: 0 Updatable: Verdadero

Columnas

Nombre	Tipo	Tamaño
TpoCdgo	Texto	4
Dscrpcion	Texto	50
Nit	Texto	15
Estdo	Entero largo	4

Tabla: TblPrcso

Propiedades

DateCreated: 27/02/2008 LastUpdated: 24/04/2013
RecordCount: 0 Updatable: Verdadero

Columnas

Nombre	Tipo	Tamaño
CdgoPrc	Texto	4
NmbrePrc	Texto	50
NmbreFrmPrc	Texto	50
DscrpcionPrc	Memo	-
VldarPrc	Entero largo	4

Tabla: TblPucCnfcop

Propiedades

DateCreated: 20/10/2006
RecordCount: 0

LastUpdated: 20/10/2006
Updatable: Verdadero

Columnas

	Tipo	Tamaño
Nombre		
Cdgo	Texto	20
Dscrpcion	Texto	200
Slido	Moneda	8

Tabla: TblPyG

Propiedades

DateCreated: 10/08/2004
OrderByOn: Falso
Updatable: Verdadero

LastUpdated: 19/07/2012
RecordCount: 126

Columnas

	Tipo	Tamaño
Nombre		
Orden	Entero largo	4
Grpo1	Texto	1
Dscrp1	Texto	50
Ttal1	Moneda	8
Grpo2	Texto	2
Dscrp2	Texto	50
Ttal2	Moneda	8
Grpo3	Texto	4
Dscrp3	Texto	50
Ttal3	Moneda	8
Grpo4	Texto	6
Dscrp4	Texto	50
Ttal4	Moneda	8
Grpo5	Texto	9
Dscrp5	Texto	50
Ttal5	Moneda	8
Grpo6	Texto	12
Dscrp6	Texto	50
Ttal6	Moneda	8
cncptoordes	Texto	50

Tabla: TblSldosTrcro

Propiedades

DateCreated: 13/11/2007
RecordCount: 18199

LastUpdated: 09/01/2008
Updatable: Verdadero

Columnas

	Tipo	Tamaño
Nombre		
NroIdPrsna	Texto	20
PucCdgo	Texto	20
Dbe00	Moneda	8
Hbe00	Moneda	8
Dbe01	Moneda	8
Hbe01	Moneda	8
Dbe02	Moneda	8
Hbe02	Moneda	8
Dbe03	Moneda	8
Hbe03	Moneda	8
Dbe04	Moneda	8
Hbe04	Moneda	8
Dbe05	Moneda	8
Hbe05	Moneda	8
Dbe06	Moneda	8
Hbe06	Moneda	8
Dbe07	Moneda	8
Hbe07	Moneda	8
Dbe08	Moneda	8
Hbe08	Moneda	8
Dbe09	Moneda	8
Hbe09	Moneda	8
Dbe10	Moneda	8

Hbe10	Moneda	8
Dbel1	Moneda	8
Hbe12	Moneda	8
Dbel3	Moneda	8
Hbe13	Moneda	8
Dbel4	Moneda	8
Hbe14	Moneda	8
Dbel5	Moneda	8
Hbe15	Moneda	8
Dbel6	Moneda	8

Tabla: TblSldosTrcro

Hbe16	Moneda	8
Dbel7	Moneda	8
Hbe17	Moneda	8
Hbe11	Moneda	8
Dbel2	Moneda	8

Tabla: TblSldosTrcroCrre

Propiedades

DateCreated: 29/02/2008 LastUpdated: 29/02/2008
RecordCount: 18679 Updatable: Verdadero

Columnas

Nombre	Tipo	Tamaño
NroIdPrsna	Texto	20
PucCdgo	Texto	20
Dbel00	Moneda	8
Hbe00	Moneda	8
Dbel01	Moneda	8
Hbe01	Moneda	8
Dbel02	Moneda	8
Hbe02	Moneda	8
Dbel03	Moneda	8
Hbe03	Moneda	8
Dbel04	Moneda	8
Hbe04	Moneda	8
Dbel05	Moneda	8
Hbe05	Moneda	8
Dbel06	Moneda	8
Hbe06	Moneda	8
Dbel07	Moneda	8
Hbe07	Moneda	8
Dbel08	Moneda	8
Hbe08	Moneda	8
Dbel09	Moneda	8
Hbe09	Moneda	8
Dbel10	Moneda	8
Hbe10	Moneda	8
Dbel11	Moneda	8
Hbe11	Moneda	8
Dbel12	Moneda	8
Hbe12	Moneda	8
Dbel13	Moneda	8
Hbe13	Moneda	8
Dbel14	Moneda	8
Hbe14	Moneda	8
Dbel15	Moneda	8

Tabla: TblSldosTrcroCrre

Hbe15	Moneda	8
Dbel6	Moneda	8
Hbe16	Moneda	8
Dbel7	Moneda	8
Hbe17	Moneda	8

Tabla: TblTpoUsrio

Propiedades

DateCreated: 27/02/2008 LastUpdated: 24/04/2013

RecordCount: 0 Updatable: Verdadero

Columnas

Nombre	Tipo	Tamaño
CdgoTpo	Texto	3
NmbreTpo	Texto	50

Tabla: TblTpoUsrioPrcso

Propiedades

DateCreated: 27/02/2008 LastUpdated: 24/04/2013

RecordCount: 0 Updatable: Verdadero

Columnas

Nombre	Tipo	Tamaño
CdgoTpo	Texto	3
CdgoPrc	Texto	4

Tabla: TblTrcroMvmnto

Propiedades

DateCreated: 14/02/2004 LastUpdated: 14/02/2004

OrderByOn: Falso RecordCount: 0

Updatable: Verdadero

Columnas

Nombre	Tipo	Tamaño
CmpTpo	Texto	4
CmpNmro	Texto	8
NIT	Texto	15
DetCnta	Texto	20
Detscncia	Texto	2
DetDbePrcial	Moneda	8
DetHberPrcial	Moneda	8

Tabla: TblUsrio

Propiedades

DateCreated: 27/02/2008 LastUpdated: 24/04/2013

RecordCount: 0 Updatable: Verdadero

Columnas

Nombre	Tipo	Tamaño
CdgoUsu	Texto	10
NmbreUsu	Texto	50
ClveUsu	Texto	10
FchaCrcionUsu	Fecha/Hora	8
FchaUltmoUsu	Fecha/Hora	8
Estdousu	Texto	1
CdgoPrc	Texto	4
CdgoTpo	Texto	3

Tabla: tpo_prcso

Propiedades

DateCreated: 03/03/2001 LastUpdated: 24/04/2013

OrderByOn: Falso RecordCount: 0

Updatable: Verdadero

Columnas

Nombre	Tipo	Tamaño
tppr_codtpo	Texto	3
tppr_codprcso	Texto	4

Tabla: TpoCmprbnte

Propiedades

DateCreated: 23/03/2000 LastUpdated: 24/04/2013

OrderByOn: Falso RecordCount: 14

Updatable: Verdadero

Columnas

Nombre	Tipo	Tamaño
TpoCdgo	Texto	4
TpoNmbre	Texto	50
TpoScncial	Texto	8
TpoChque	Texto	1
TpoCnta	Texto	15

TpoMov	Entero	2
EsCrre	Entero largo	4
Autmtco	Entero largo	4

Tabla: TpoCnta

Propiedades

DateCreated:	27/05/2000	LastUpdated:	27/05/2000
OrderByOn:	Falso	RecordCount:	7
Updatable:	Verdadero		

Columnas

Nombre	Tipo	Tamaño	
Cnta	Texto		1
Mdo	Texto		1

Tabla: Trnscion

Propiedades

DateCreated:	25/02/2002	LastUpdated:	18/10/2003
OrderByOn:	Falso	RecordCount:	0
Updatable:	Verdadero		

Columnas

Nombre	Tipo	Tamaño	
Ano	Texto		4
Estdo	Texto		1
Trsldo	Sí/No		1

ANEXO C

ACTA DE COMPROMISO DE ENTREGA DE HERRAMIENTA SOFTWARE CONTAR EN SU VERSION 3.0



COMPROMISO DE ENTREGA

Acordado entre las partes: **EVACC LTDA** compañía desarrolladora de software y en su representación el Ingeniero **CARLOS FELIPE REYES CONTRERAS** con cédula de ciudadanía **13'354.870** de Pamplona, quien de aquí en adelante se denominará LA EMPRESA y **CARLOS FERNANDO VEGA REYES** con cédula de ciudadanía **74'380.498** de Duitama; estudiante activo de la Universidad Industrial de Santander, que actualmente ejerce labores como estudiante en proceso de práctica, quien de aquí en adelante se denominará como EL ESTUDIANTE; donde se especifican las condiciones necesarias para el desarrollo del aplicativo software cuyo nombre ha sido denominado como CONTAR en su versión 3.0 bajo las siguientes cláusulas:

1. EL ESTUDIANTE se compromete a prestar los servicios de apoyo en el cambio de plataforma de la herramienta software CONTAR en su versión 3.0 y el modelado de la base de datos de acuerdo a las especificaciones que LA EMPRESA determine, siempre y cuando se encuentren dentro de los requerimientos estipulados por la práctica empresarial que actualmente se encuentra desarrollando.
2. Los requerimientos y alcances del producto se encuentran limitados como primera instancia a los requisitos a los que dé a lugar la práctica empresarial que actualmente desarrolla EL ESTUDIANTE, y como segunda instancia a las directivas que profese el grupo de desarrollo de LA EMPRESA, dando por terminado el actual convenio en caso que alguna de las partes determine incidir en faltas a las cláusulas que aquí se enuncian.
3. Las labores que EL ESTUDIANTE realice en favor de LA EMPRESA serán de entera disposición de la misma según ambas partes lo estipulen; toda actividad de codificación que requiere la terminación del producto se encontrará bajo la dirección del grupo de desarrollo de LA EMPRESA y será aprobada por la directiva organizacional de la misma en reuniones que de común acuerdo se programen.
4. Se contará con el apoyo de un documento escrito en el que se definan los procedimientos a realizar para la entrega del producto final donde se especifica la codificación a utilizar, la metodología de implementación, el modelado de datos y la disposición de ventanas del mismo. Dicho documento será implementado como proyecto de grado a favor de EL ESTUDIANTE el cual se compromete a presentar una copia del mismo y facilitarla para su archivo con fines enteramente informativos.
5. El producto terminado será avalado por las directivas de LA EMPRESA siempre y cuando cumpla con las condiciones de montaje del reglamento interno de la misma; información que será suministrada a EL ESTUDIANTE en la medida en que ambas partes lo requieran, con la prerrequisito expreso que dicha información sea entregada veraz, efectiva y oportunamente.

Carrera 31 # 40-19 Apto 302 Celular: 3167981105 Teléfono: 6321835
Bucaramanga



ENTORNOS VIRTUALES DE APRENDIZAJE LTDA

6. La metodología a implementar durante el proyecto actual será determinada en su totalidad por EL ESTUDIANTE y ejecutada luego de darse a conocer en las reuniones dispuestas para tal fin; siendo entera responsabilidad de LA EMPRESA su ejecución y puesta en marcha y de igual forma las acciones legales a las que haya lugar para el desarrollo de las mismas.

7. El vencimiento de presente convenio se encuentra dado por la finalización del periodo de práctica que EL ESTUDIANTE se encuentra desarrollando en LA EMPRESA; de tal forma que se dará por terminado todo vínculo establecido entre las dos partes una vez se haya culminado el objetivo de dicha práctica.

En Bucaramanga a 22 de octubre de 2012, en constancia de lo anterior firman:

CARLOS FELIPE REYES CONTRERAS
Representante de EVACC Ltda.
C.C. 13354870

CARLOS FERNANDO VEGA REYES
Estudiante en proceso de práctica.
C.C. 74380498

Carrera 31 # 40-19 Apto 302 Celular: 3167981105 Teléfono: 6321835
Bucaramanga

ANEXO D

LISTADO DE TABLAS DEFINITIVAS CORRESPONDIENTES A LA BASE DE DATOS DE LA HERRAMIENTA CONTAR EN SU VERSION 3.0 IMPLEMENTADAS EN POSTGRESQL

A continuación se presenta una tabla que detalla la estructura implementada en el gestor de base de datos PostgreSQL donde se menciona el nombre de cada campo y su tabla correspondiente, especificando para cada uno si permite valores nulos o no, además del tipo de datos que admite y en caso de ser tipo 'character varying' la cantidad máxima de caracteres que pueden incorporarse.

<i>Nombre de la tabla</i>	<i>Nombre del campo</i>	<i>NULL</i>	<i>Tipo de campo</i>	<i>MAX</i>
tblcmprbnte	cmptpo	NO	character varying	4
tblcmprbnte	cmpnmro	NO	character varying	8
tblcmprbnte	cmpfcha	YES	timestamp with time zone	
tblcmprbnte	cmpentdad	YES	character varying	14
tblcmprbnte	cmpchque	YES	character varying	20
tblcmprbnte	cmpbnco	YES	character varying	20
tblcmprbnte	cmpestdo	YES	character varying	1
tblcmprbnte	cmpcncpto	YES	text	
tblcmprbnte	cmpdbto	YES	double precision	
tblcmprbnte	cmpcrdto	YES	double precision	
tblcmprbnte	cmpnmbreentdad	YES	character varying	50
tblcmprbnte	cmpvlorchque	YES	double precision	
tblcmprbnte	cmpcntabnco	YES	character varying	15
tblcmprbnte	escrre	YES	smallint	
tblcmprbnte	cdgobnco	YES	character varying	4
tbldtllecprbnte	dettpo	NO	character varying	4
tbldtllecprbnte	detnmro	NO	character varying	8
tbldtllecprbnte	detcnta	NO	character varying	20
tbldtllecprbnte	detscncia	NO	integer	
tbldtllecprbnte	detdscrpcion	YES	character varying	100
tbldtllecprbnte	detdbeprcial	YES	double precision	
tbldtllecprbnte	dethberprcial	YES	double precision	
tbldtllecprbnte	detdbe	YES	double precision	
tbldtllecprbnte	dethber	YES	double precision	
tbldtllecprbnte	detcncpto	YES	text	
tbldtllecprbnte	detnit	YES	character varying	15
tbldtllecprbnte	cdgoctrocsto	YES	smallint	
tbldtllecprbnte	detorden	YES	smallint	
tbldtllecprbnte	nmrodcmnto	YES	character varying	20
tbldtllecprbnte	fchadcmnto	YES	timestamp with time zone	
tbldtllecprbnte	tpodcmnto	YES	character varying	1
tbldtllecprbnteanldo	dettpo	NO	character varying	4

tblDtllecMprbnteanldo	detnmro	NO	character varying	8
tblDtllecMprbnteanldo	detcnta	NO	character varying	20
tblDtllecMprbnteanldo	detscncia	NO	integer	
tblDtllecMprbnteanldo	detdsrpcion	YES	character varying	100
tblDtllecMprbnteanldo	detdbeprcial	YES	double precision	
tblDtllecMprbnteanldo	dethberprcial	YES	double precision	
tblDtllecMprbnteanldo	detdbe	YES	double precision	
tblDtllecMprbnteanldo	dethber	YES	double precision	
tblDtllecMprbnteanldo	detcncpto	YES	text	
tblDtllecMprbnteanldo	detnit	YES	character varying	15
tblDtllecMprbnteanldo	cdgoctrocsto	YES	integer	
tblDtllecMprbnteanldo	detorden	YES	integer	
tblDtllecMprbnteanldo	nmrodcmnto	YES	character varying	20
tblDtllecMprbnteanldo	fchadcmnto	YES	timestamp with time zone	
tblDtllecMprbnteanldo	tpodcmnto	YES	character varying	1
tblprdo	pucdgo	NO	character varying	20
tblprdo	pucnmbre	YES	character varying	50
tblprdo	pucsloddbtomesant	YES	double precision	
tblprdo	pucslodcrdtomesant	YES	double precision	
tblprdo	pucsloddbtomesact	YES	double precision	
tblprdo	pucslodcrdtomesact	YES	double precision	
tblprdo	pucajstedbtomesant	YES	double precision	
tblprdo	pucajstecrdtomesant	YES	double precision	
tblprdo	pucajstedbtomesact	YES	double precision	
tblprdo	pucajstecrdtomesact	YES	double precision	
tblprdo	pucntrocstos	YES	character varying	1
tblprdo	pucpgostrcros	YES	character varying	1
tblprdo	pucajstes	YES	character varying	1
tblprdo	puctrcro	YES	character varying	50
tblprdo	scnciachque	YES	integer	
tblprdo	trcrocre	YES	integer	
tblprdo	nittcrocre	YES	character varying	15
tblprdos	ano	NO	character varying	4
tblprdos	prdo	NO	character varying	2
tblprdos	estdo	YES	character varying	1
tblprdos	esactvo	YES	character varying	1
tblpuc	pucdgo	NO	character varying	20
tblpuc	pucnmbre	YES	character varying	100
tblpuc	pucsloddbtomesant	YES	double precision	
tblpuc	pucslodcrdtomesant	YES	double precision	
tblpuc	pucsloddbtomesact	YES	double precision	
tblpuc	pucslodcrdtomesact	YES	double precision	
tblpuc	pucajstedbtomesant	YES	double precision	
tblpuc	pucajstecrdtomesant	YES	double precision	
tblpuc	pucajstedbtomesact	YES	double precision	
tblpuc	pucajstecrdtomesact	YES	double precision	
tblpuc	pucntrocstos	YES	character varying	1
tblpuc	pucpgostrcros	YES	character varying	1

tblpuc	pucajstes	YES	character varying	1
tblpuc	puctrcro	YES	character varying	50
tblpuc	scnciachque	YES	integer	
tblpuc	trcrocre	YES	integer	
tblpuc	nittcrocre	YES	character varying	15
tblbncosdtlle	cdgobnco	NO	integer	
tblbncosdtlle	nrocnta	NO	character varying	30
tblbncosdtlle	pucdgo	YES	character varying	20
tblbncosdtlle	scncia	YES	character varying	20
tblbncosdtlle	cdgofrmto	YES	integer	
tblcncptocnta	nmrocncpto	NO	integer	
tblcncptocnta	pucdgo	YES	character varying	20
tblcncptodian	nmrocncpto	NO	integer	
tblcncptodian	dscrpcioncncpto	YES	character varying	50
tblcncptodian	nmrofrmto	YES	integer	
tblcntaspyg	orden	YES	smallint	
tblcntaspyg	grpo	YES	smallint	
tblcntaspyg	pucdgo	YES	character varying	20
tblcntrocsto	cdgoctrocsto	NO	smallint	
tblcntrocsto	dscrpcioncntrocsto	YES	character varying	100
tblcrtfcdortfnte	cdgocrtfcdo	NO	integer	
tblcrtfcdortfnte	anogrvble	YES	character varying	4
tblcrtfcdortfnte	prdo	YES	character varying	100
tblcrtfcdortfnte	nit	YES	character varying	15
tblcrtfcdortfnte	nmbre	YES	character varying	100
tblcrtfcdortfnte	ttal	YES	double precision	
tblcrtfcdortfnte	cnsngndo	YES	character varying	100
tblcrtfcdortfnte	ttalltras	YES	character varying	200
tblcrtfcdortfntedtllle	cdgocrtfcdo	NO	integer	
tblcrtfcdortfntedtllle	cncpto	YES	character varying	100
tblcrtfcdortfntedtllle	prcntje	YES	double precision	
tblcrtfcdortfntedtllle	bse	YES	double precision	
tblcrtfcdortfntedtllle	vlorrndo	YES	double precision	
tbldfncion	cdgodef	YES	character varying	4
tbldfncion	nmbredef	YES	character varying	20
tbldfncion	descdef	YES	character varying	50
tbldfncion	tpovlordef	YES	character varying	1
tbldfncion	vlordef	YES	character varying	50
tblfrmtodian	nmrofrmto	YES	integer	
tblfrmtodian	dscrpcionfrmto	YES	character varying	50
tblorgnzcion	nitorg	YES	character varying	20
tblorgnzcion	nmbreorg	YES	character varying	100
tblorgnzcion	cbza1org	YES	character varying	200
tblorgnzcion	cbza2org	YES	character varying	200
tblorgnzcion	cbza3org	YES	character varying	200
tblorgnzcion	cbza4org	YES	character varying	200
tblorgnzcion	pie1org	YES	character varying	200
tblorgnzcion	pie2org	YES	character varying	200

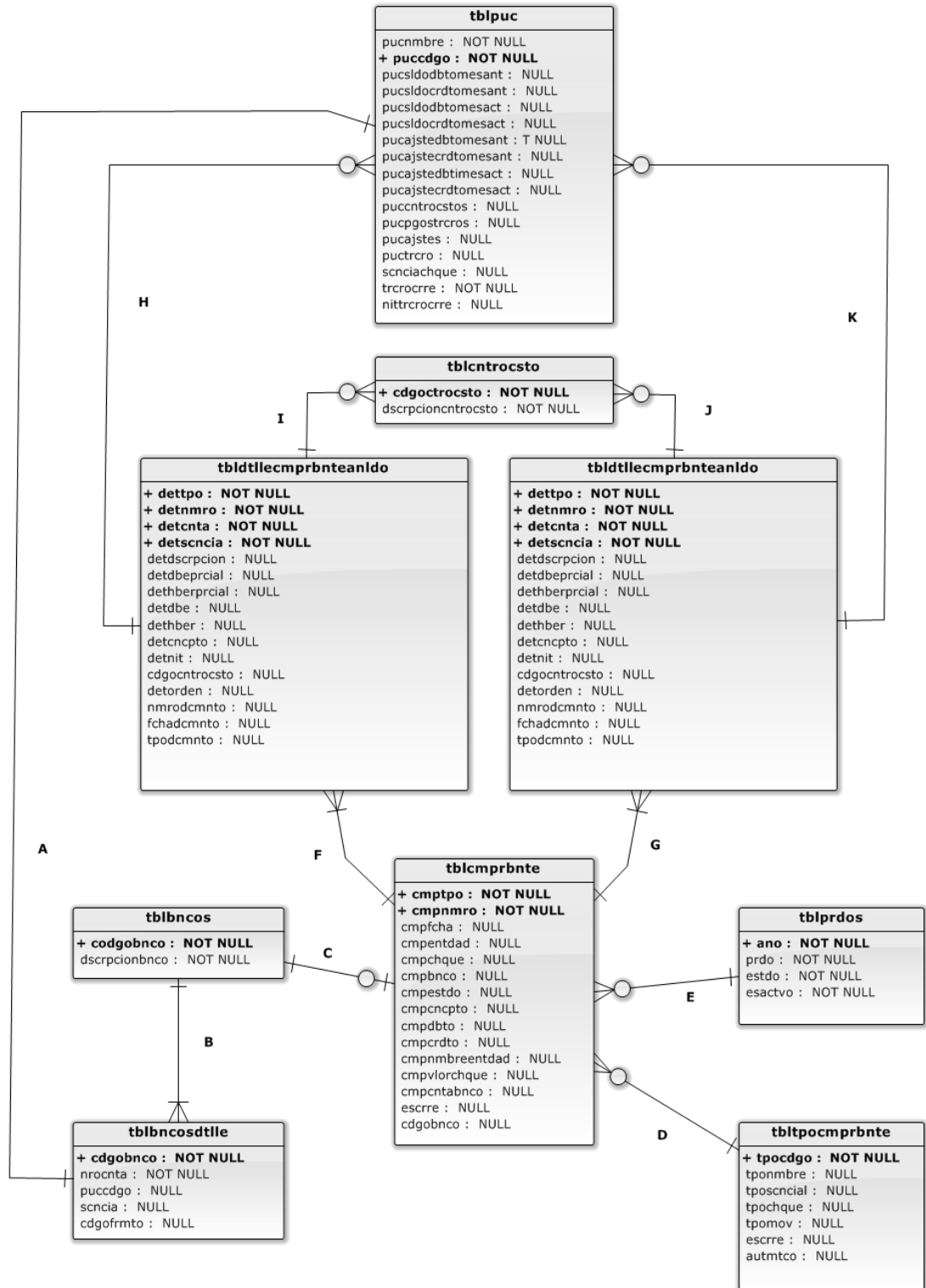
tblorgnzcion	pie3org	YES	character varying	200
tblorgnzcion	pie4org	YES	character varying	200
tblorgnzcion	rprsntnteorg	YES	character varying	200
tblorgnzcion	cdlarprsntnteorg	YES	character varying	20
tblorgnzcion	drccionorg	YES	character varying	100
tblorgnzcion	tlfnoorg	YES	character varying	50
tblorgnzcion	emailorg	YES	character varying	200
tblorgnzcion	cdadorg	YES	character varying	200
tblorgnzcion	frmasrcbicnfrme	YES	smallint	
tblorgnzcion	abrvtra	YES	character varying	20
tblorgnzcion	prfjocmprbnte	YES	character varying	1
tblorgnzcion	trcroextrno	YES	smallint	
tblorgnzcion	aux1	YES	smallint	
tblorgnzcion	aux2	YES	smallint	
tblorgnzcion	rtaprsnas	YES	character varying	255
tblorgnzcion	frmtochque	YES	integer	
tblorgnzcion	cntrocsto	YES	integer	
tblprcso	cdgoprc	NO	character varying	4
tblprcso	nmbreprc	YES	character varying	50
tblprcso	nmbrefrmprc	YES	character varying	50
tblprcso	dscrpcionprc	YES	text	
tblprcso	vldarprc	YES	integer	
tblsldostrcro	nroidprsna	NO	character varying	20
tblsldostrcro	puccdgo	NO	character varying	20
tblsldostrcro	dbe00	YES	double precision	
tblsldostrcro	hbe00	YES	double precision	
tblsldostrcro	dbe01	YES	double precision	
tblsldostrcro	hbe01	YES	double precision	
tblsldostrcro	dbe02	YES	double precision	
tblsldostrcro	hbe02	YES	double precision	
tblsldostrcro	dbe03	YES	double precision	
tblsldostrcro	hbe03	YES	double precision	
tblsldostrcro	dbe04	YES	double precision	
tblsldostrcro	hbe04	YES	double precision	
tblsldostrcro	dbe05	YES	double precision	
tblsldostrcro	hbe05	YES	double precision	
tblsldostrcro	dbe06	YES	double precision	
tblsldostrcro	hbe06	YES	double precision	
tblsldostrcro	dbe07	YES	double precision	
tblsldostrcro	hbe07	YES	double precision	
tblsldostrcro	dbe08	YES	double precision	
tblsldostrcro	hbe08	YES	double precision	
tblsldostrcro	dbe09	YES	double precision	
tblsldostrcro	hbe09	YES	double precision	
tblsldostrcro	dbe10	YES	double precision	
tblsldostrcro	hbe10	YES	double precision	
tblsldostrcro	dbe11	YES	double precision	
tblsldostrcro	hbe12	YES	double precision	

tblsldostrcro	dbe13	YES	double precision	
tblsldostrcro	hbe13	YES	double precision	
tblsldostrcro	dbe14	YES	double precision	
tblsldostrcro	hbe14	YES	double precision	
tblsldostrcro	dbe15	YES	double precision	
tblsldostrcro	hbe15	YES	double precision	
tblsldostrcro	dbe16	YES	double precision	
tblsldostrcro	hbe16	YES	double precision	
tblsldostrcro	dbe17	YES	double precision	
tblsldostrcro	hbe17	YES	double precision	
tblsldostrcro	hbe11	YES	double precision	
tblsldostrcro	dbe12	YES	double precision	
tblsldostrcrocrre	nroidprsna	NO	character varying	20
tblsldostrcrocrre	puccdgo	NO	character varying	20
tblsldostrcrocrre	dbe00	YES	double precision	
tblsldostrcrocrre	hbe00	YES	double precision	
tblsldostrcrocrre	dbe01	YES	double precision	
tblsldostrcrocrre	hbe01	YES	double precision	
tblsldostrcrocrre	dbe02	YES	double precision	
tblsldostrcrocrre	hbe02	YES	double precision	
tblsldostrcrocrre	dbe03	YES	double precision	
tblsldostrcrocrre	hbe03	YES	double precision	
tblsldostrcrocrre	dbe04	YES	double precision	
tblsldostrcrocrre	hbe04	YES	double precision	
tblsldostrcrocrre	dbe05	YES	double precision	
tblsldostrcrocrre	hbe05	YES	double precision	
tblsldostrcrocrre	dbe06	YES	double precision	
tblsldostrcrocrre	hbe06	YES	double precision	
tblsldostrcrocrre	dbe07	YES	double precision	
tblsldostrcrocrre	hbe07	YES	double precision	
tblsldostrcrocrre	dbe08	YES	double precision	
tblsldostrcrocrre	hbe08	YES	double precision	
tblsldostrcrocrre	dbe09	YES	double precision	
tblsldostrcrocrre	hbe09	YES	double precision	
tblsldostrcrocrre	dbe10	YES	double precision	
tblsldostrcrocrre	hbe10	YES	double precision	
tblsldostrcrocrre	dbe11	YES	double precision	
tblsldostrcrocrre	hbe11	YES	double precision	
tblsldostrcrocrre	dbe12	YES	double precision	
tblsldostrcrocrre	hbe12	YES	double precision	
tblsldostrcrocrre	dbe13	YES	double precision	
tblsldostrcrocrre	hbe13	YES	double precision	
tblsldostrcrocrre	dbe14	YES	double precision	
tblsldostrcrocrre	hbe14	YES	double precision	
tblsldostrcrocrre	dbe15	YES	double precision	
tblsldostrcrocrre	hbe15	YES	double precision	
tblsldostrcrocrre	dbe16	YES	double precision	
tblsldostrcrocrre	hbe16	YES	double precision	

tblsldostrcrocrre	dbe17	YES	double precision	
tblsldostrcrocrre	hbe17	YES	double precision	
tbltpousrio	cdgotpo	NO	character varying	3
tbltpousrio	nmbretpo	YES	character varying	50
tbltpousrioprcso	cdgotpo	NO	character varying	3
tbltpousrioprcso	cdgoprc	NO	character varying	4
tblusrio	cdgousu	NO	character varying	10
tblusrio	nmbreusu	YES	character varying	50
tblusrio	clveusu	YES	character varying	10
tblusrio	fchacrcionusu	YES	timestamp with time zone	
tblusrio	fchaultmousu	YES	timestamp with time zone	
tblusrio	estdousu	YES	character varying	1
tblusrio	cdgoprc	YES	character varying	4
tblusrio	cdgotpo	YES	character varying	3
tblplntllacmprbnte	tpocdgo	YES	character varying	4
tblplntllacmprbnte	dscrpcion	YES	character varying	50
tblplntllacmprbnte	nit	YES	character varying	15
tblplntllacmprbnte	estdo	YES	integer	
tblplntllacmprbntedtile	tpocdgo	YES	character varying	4
tblplntllacmprbntedtile	orden	YES	integer	
tblplntllacmprbntedtile	dscrpcion	YES	character varying	50
tblplntllacmprbntedtile	puccdgo	YES	character varying	20
tblplntllacmprbntedtile	dbtocrdto	YES	integer	
tblplntllacmprbntedtile	valor	YES	double precision	
tbltpocmprbnte	tpocdgo	NO	character varying	4
tbltpocmprbnte	tponmbre	YES	character varying	50
tbltpocmprbnte	tposcncial	YES	character varying	8
tbltpocmprbnte	tpochque	YES	character varying	1
tbltpocmprbnte	tpomov	YES	smallint	
tbltpocmprbnte	escrre	YES	integer	
tbltpocmprbnte	autmtco	YES	integer	
tbltpocnta	cnta	NO	character varying	1
tbltpocnta	mdo	YES	character varying	1
tblbncos	cdgobnco	NO	integer	
tblbncos	dscrpcionbnco	YES	character varying	100
tbltpo_prcso	tppr_codtpo	NO	character varying	3
tbltpo_prcso	tppr_codprcso	NO	character varying	4

ANEXO E

DIAGRAMA RELACIONAL DE LA BASE DE DATOS IMPLEMENTADA EN POSTGRESQL



DETALLES

- § **Relación A (1-1):** A cada cuenta relacionada con un banco debe asignársele una única clave en el plan único de cuentas.
- § **Relación B (1-1,n):** Cada banco debe contener al menos una cuenta asociada a sí mismo sin límite de cuentas.
- § **Relación C (1-0,n):** En cada comprobante se debe relacionar un banco aunque no es necesaria su inclusión en cada registro.
- § **Relación D (1-0,n):** Por cada tipo de comprobante es posible crear cualquier cantidad de comprobantes de dicho tipo o también pueden no existir comprobantes de tal tipo.
- § **Relación E (1-0,n):** La fecha de cada comprobante puede o no estar condicionada por un único registro de la tabla periodos
- § **Relación F (1-1,n):** Todo comprobante debe contener al menos un registro de detalle, no existe un límite de los mismos.
- § **Relación G (1-1,n):** Todo comprobante anulado debe contener al menos un registro de detalle, no existe un límite de los mismos.
- § **Relación H (1-0,n):** Para cada línea del detalle de un comprobante es posible incluir una o ninguna cuenta del puc.
- § **Relación I (1-0,n):** Para cada línea del detalle de un comprobante anulado es posible incluir una o ninguna cuenta del puc.
- § **Relación J (1-0,n):** Para cada línea del detalle de un comprobante es posible incluir uno o ninguno de los centros de costo disponibles.
- § **Relación K (1-0,n):** Para cada línea del detalle de un comprobante anulado es posible incluir uno o ninguno de los centros de costo disponibles.

De acuerdo a lo reglamentado por el grupo de desarrollo de EVACC Ltda se han restringido las relaciones de las demás tablas en el modelo de datos debido a posibles cambios a futuro hasta el momento en que sean requeridas para su implementación.

ANEXO F

DISTRIBUCION DEL NUEVO MENU DENTRO DE LA BARRA DE COMANDOS DEL FORMULARIO PRINCIPAL

Para el diseño del menú principal de la aplicación, el grupo desarrollador solicitó un esquema grafico que permitiera la fácil y rápida identificación de cada uno de los menús y submenús de tal forma que su formato estuviese abierto a expansiones y/o modificaciones, teniendo en cuenta que la cobertura del proyecto actual no abarca el desarrollo de todos los módulos de la herramienta.

DRIAGRAMA ORGANIZACIONAL DE MENU POR NIVELES HERRAMIENTA SOFTWARE CONTAR VERSION 3.0



Carlos Fernando Vega

Menú Principal



Archivos

Plan Único de Cuentas

Tipo Comprobante

Terceros

Centros de Costo

Bancos

Salir

Menú Principal

Eventos

Gestión de Comprobantes

Cruce de comprobantes

Procesos

Cierre mensual

Reiniciar Periodo Actual

Reversar periodos

Cierre anual

Reversar cierre anual

Menú Principal

Consultas e informes (*)

Balance de Comprobación	Balance de comprobación general Balance de comprobación con terceros
Libros Oficiales	Libro Diario Mayor y Balances Inventarios y Balances
Libros Auxiliares	Por cuenta Por cuenta y NIT Por tercero Por centro de costo
Libros Generales	Balance General Estado de resultados Estado de resultados por periodo
Otros Informes	Formatos DIAN PUC NITs Estados especiales Certificado de retención Paginación Libros oficiales
Análisis Financiero	Comparativo

(*) La estructura se propone como prototipo para la continuación del proyecto en su siguiente fase; el grupo desarrollador de la empresa no la asume como estructura definitiva debido a los posibles cambios que puedan generarse en la implementación de los módulos correspondientes.

Menú Principal

Utilidades

Cadenas de
conexión.

Base de datos principal

Base de datos terceros

Ejecutable Terceros

Reorganizar Comprobantes

Organización
(*)

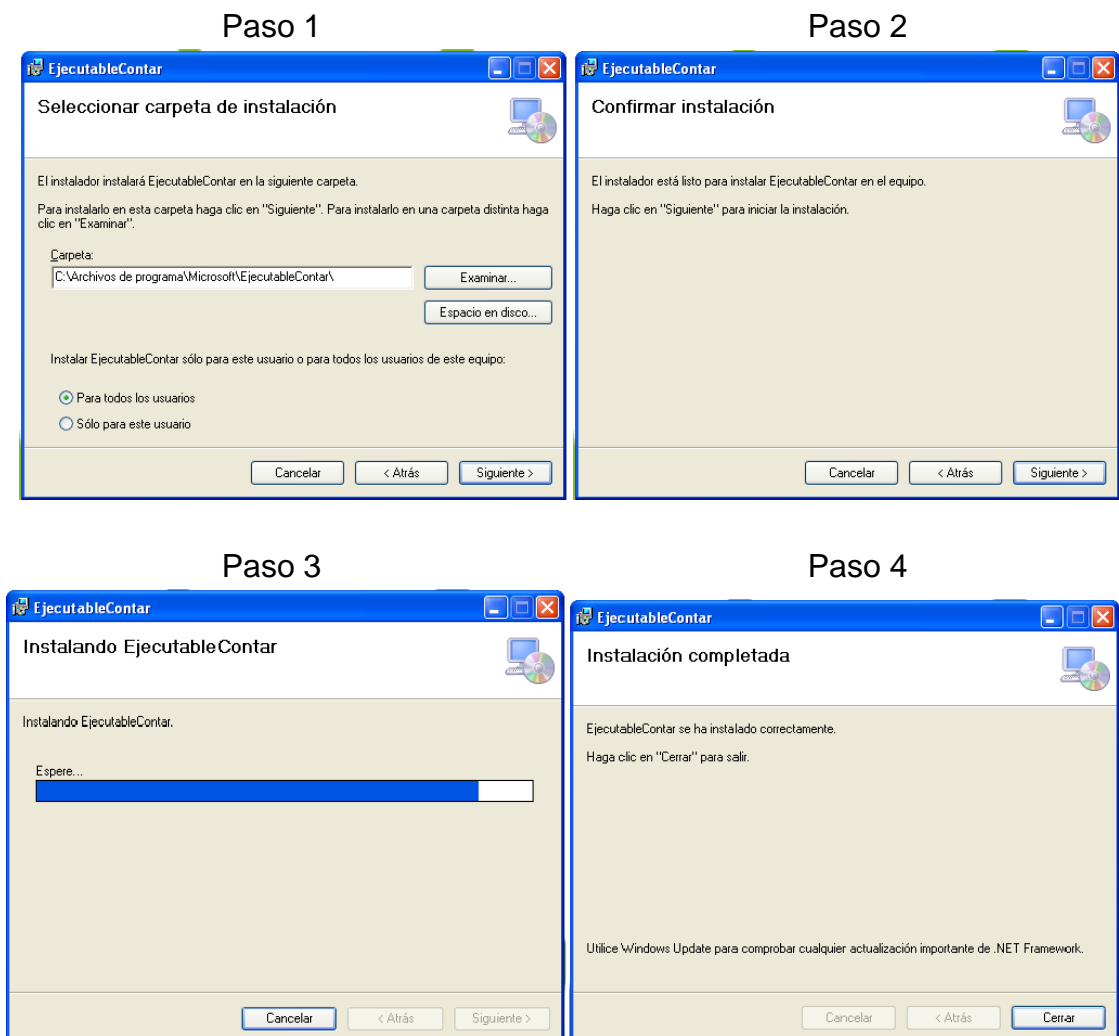
(*) Para la implementación de este menú se requiere del módulo de usuarios y las validaciones con contraseñas, lo cual no se encuentra dentro de los alcances del presente proyecto, de manera que se establece el módulo como definitivo pero su implementación se posterga para futuras versiones de la herramienta.

ANEXO G

PRUEBAS DE MONTAJE REALIZADAS EN DIFERENTES EQUIPOS

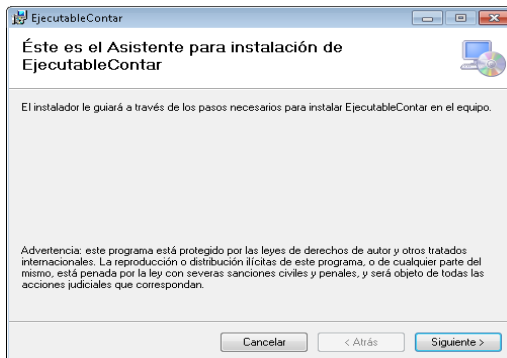
Gracias al aporte de la empresa EVACC Ltda fue posible contar con tres máquinas para realizar las pruebas de montaje de la aplicación en las mismas; a continuación se presentan la evidencias graficas del procedimiento realizado. Algunas capturas de pantalla presentan desperfectos por problemas de rendimiento propios de cada computadora, mas sin embargo la información contenida no pierde validez por la ocurrencia de dicho inconveniente.

Equipo número 1.

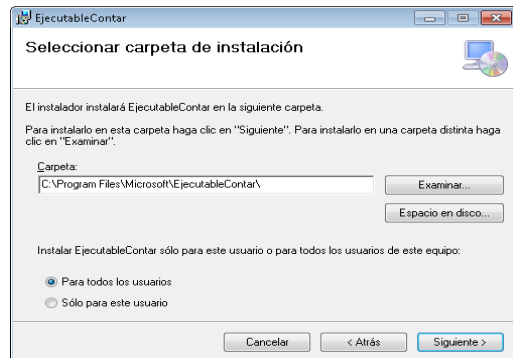


Equipo número 2 y número 3.

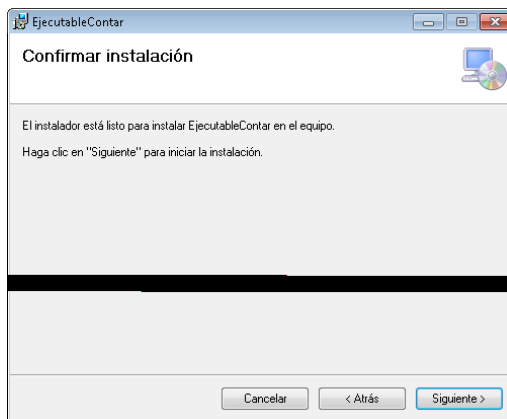
Paso 01



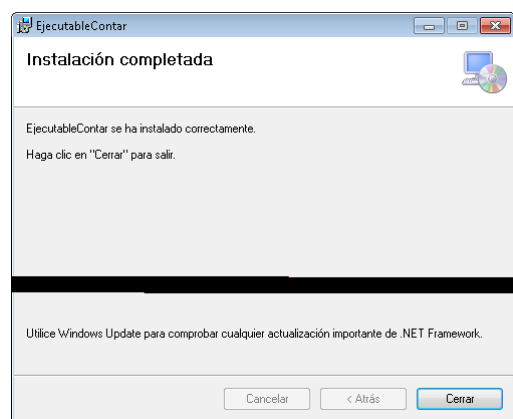
Paso 02



Paso 03

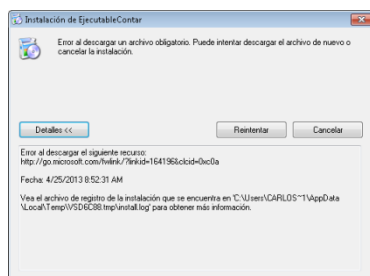


Paso 04

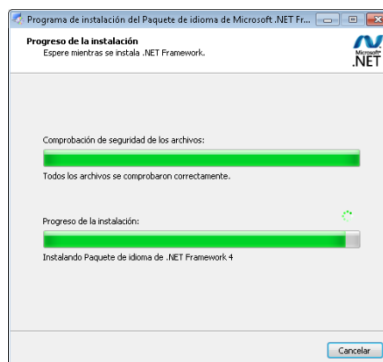


Para estas máquinas fue necesaria la instalación de un paquete de lenguaje de la plataforma .NET.

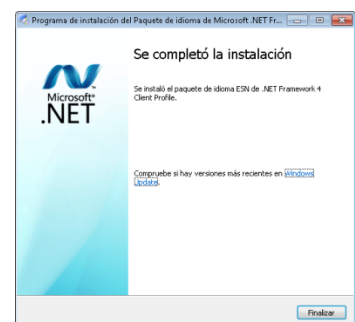
A



B



C



- A: Error indicando la falta del paquete de lenguaje
- B: Proceso de instalación del paquete de lenguaje
- C: Confirmación de la instalación exitosamente.

ANEXO H

DOCUMENTO DE ENTREGA DE LA HERRAMIENTA CONTAR EN SU VERSION 3.0

Bucaramanga, 5 de marzo de 2013

Señores
EVACC Ltda.
Carlos Felipe Reyes Contreras
Representante legal.

Asunto: Entrega del producto software Contar v3.0

Por medio de la presente me dirijo a ustedes con el fin de soportar por medio del presente documento la **ENTREGA** del producto software en la versión 3.0 de acuerdo a los requisitos establecidos para su elaboración. Me permito recordar que la documentación del mismo se encuentra soportada en el documento usado como proyecto de grado donde se especifican aspectos como: la metodología usada para el desarrollo de la práctica, estandarización de código, modelado de datos y disposición de formularios que hacen parte de la herramienta entregada en sus manos.

De igual manera me permito describir el conjunto de archivos que hacen parte del disco de instalación del aplicativo, los cuales se encuentran dispuestos de la siguiente forma:

En el directorio raíz se encuentran tres carpetas con los nombres de *BD* (haciendo referencia a la base de datos), *Debug* (haciendo referencia a la depuración del programa) y *Docs* (la cual contiene los documentos adicionales requeridos); junto a ellos el archivo de autoarranque cuyo nombre es: *Autorun.ini* que contiene la codificación necesaria para la ejecución automática del mismo y el re direccionamiento de los accesos a otras opciones de acuerdo a configuraciones del sistema operativo de montaje.

El contenido de las carpetas está dispuesto así:

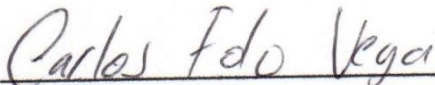
En la carpeta *BD* se encuentra el paquete ejecutable de *PostgreSQL* en la versión 8.3 y un archivo de extensión *.bat* llamado *gestor*, que es el encargado de adicionar la ejecución de dicho paquete al menú contextual de la unidad.

En la carpeta *Debug* se encuentra el paquete instalable del aplicativo 'Contar' en su versión 3.0 junto a los archivos necesarios para su implementación (Generado por la suite Visual Studio 2010).

Por último, en la carpeta *Docs* se encuentra un back-up de la base de datos con información necesaria para la restauración y/o soporte de la misma que lleva el nombre de *base_datos_inicial.backup*, junto a este archivo se encuentran dos videos en formato mp4 que funcionan como video-tutoriales acerca de la instalación y utilización de la herramienta llamados *video1* y *video2* respectivamente; de la misma manera se hallan dos archivos con extensión *.bat* que gracias a su codificación, son los encargados de adicionar la funcionalidad de pre-visualizar los videos anteriormente mencionados al menú contextual de la unidad de cd; estos archivos se llaman *runvid1* y *runvid2* respectivamente.

La firma del presente escrito se asume como la aceptación de los resultados obtenidos y se verifica el cumplimiento a cabalidad con las clausulas expuestas para la realización de tal fin.

Cordialmente.



Carlos Fernando Vega Reyes
C.C. 74.380498