



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS**

Tema:

“Módulo de Business Intelligence para el Sistema Financiero Financial de la empresa
Sifizsoft-Ambato”

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la
obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

SUBLINEA DE INVESTIGACIÓN: Aplicaciones distribuidas
AUTOR: Juan Fernando Garcés Freire
PROFESOR REVISOR: Ing. Mg. Kléver Renato Urvina Barrionuevo

Ambato – Ecuador

Julio – 2015

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el tema: “**Módulo de Business Intelligence para el Sistema Financiero Financial de la empresa Sifizsoft-Ambato**”, del señor Juan Fernando Garcés Freire, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato julio, 2015

EL TUTOR

Ing. Mg. Kléver Renato Urvina Barrionuevo

AUTORÍA

El presente trabajo de investigación titulado: **“Módulo de Business Intelligence para el Sistema Financiero Financial de la empresa Sifizsoft-Ambato”**, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato julio, 2015

Juan Fernando Garcés Freire

CC: 1803960473

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato julio, 2015

Juan Fernando Garcés Freire

CC: 1803960473

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Edison Álvarez y el Ing. Carlos Núñez, revisó y aprobó el Informe Final del Proyecto de Investigación titulado “**Módulo de Business Intelligence para el Sistema Financiero Financiero de la empresa Sifizsoft-Ambato**”, presentado por el señor Juan Fernando Garcés Freire, de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ing. Mg. Vicente Morales
PRESIDENTE DEL TRIBUNAL

Ing. Edison Álvarez
DOCENTE CALIFICADOR

Ing. Carlos Núñez
DOCENTE CALIFICADOR

DEDICATORIA

El presente trabajo se lo dedico a mi padre Ángel que desde el cielo me cuida y con su bendición me ha dado el coraje para afrontar los retos de todos los días.

A mi madre Beatriz que gracias a su apoyo incondicional y su ejemplo me ha inculcado los valores necesarios para hacer de mí un hombre de bien.

A mis hermanos Hernán, Danilo, Alex y Marlon que más que mi sangre son mis amigos y siempre han estado para apoyarme en momentos de dificultad.

A Cristina que con ese don de mujer tierna y amorosa ha sabido comprenderme y ha sido mi compañera incondicional en duras batallas.

Juan Fernando Garcés Freire.

AGRADECIMIENTO

Quiero agradecer a Dios por darme la vida y por permitirme estar en este momento tan importante de mi carrera profesional.

También agradecer a mi familia que siempre estuvieron a mi lado dándome aliento para no desmayar en mi formación académica y personal.

Y un especial agradecimiento a la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, por haberme permitido formarme académicamente y a sus profesores que desinteresadamente siempre expusieron todo su conocimiento para formar líderes de bien para la sociedad.

Juan Fernando Garcés Freire

TABLA DE CONTENIDO

Portada.....	I
Aprobación del tutor.....	II
Autoría.....	III
Derechos de autor.....	IV
Aprobación de la comisión calificadora.....	V
Dedicatoria.....	VI
Agradecimiento.....	VII
Resumen ejecutivo.....	XV
Abstract.....	XVI
Glosario de términos y acrónimos.....	XVII
Introducción.....	XIX
CAPÍTULO I.....	1
EL PROBLEMA.....	1
1.1. Tema.....	1
1.2. Planteamiento del problema.....	1
1.2.1. Contextualización.....	1
1.3. Delimitación.....	2
1.4. Justificación.....	3
1.5. Objetivos.....	4
1.5.1. Objetivo general.....	4
1.5.2. Objetivos específicos.....	4
CAPÍTULO II.....	5
MARCO TEÓRICO.....	5
2.1. Antecedentes Investigativos.....	5
2.2. Fundamentación Teórica.....	6
2.2.1. Business Intelligence.....	6
2.2.2. Metodología Hefesto.....	6
2.2.3. Comparación de Hefesto con otras metodologías.....	8
2.2.4. Data Warehousing.....	9
2.2.5. Cubo OLAP.....	10
2.2.6. Arquitectura OLAP.....	10
2.2.7. ETL.....	10
2.2.8. Tablas y entidades de dimensiones.....	11
2.2.9. Tablas de Hechos.....	11
2.2.10. Esquemas Dimensionales.....	11
2.2.11. Esquema de Estrella.....	12
2.2.12. Esquema de Copo de Nieve.....	13

2.2.13.	Esquema de Constelación	13
2.2.14.	Key Performance Indicator (KPI).....	14
2.2.15.	Cartera Financiera.....	15
2.2.16.	Depósitos a la Vista	15
2.2.17.	Depósitos a plazo fijo	16
2.2.18.	SQL Server Analysis Services.....	16
2.2.19.	Devexpress	16
2.2.20.	Financial Business System v2.0	17
2.3.	Propuesta de Solución.....	19
CAPÍTULO III		20
METODOLOGÍA.....		20
3.1.	Modalidad de la investigación	20
3.2.	Población y muestra.....	20
3.3.	Recolección de información	20
3.4.	Procesamiento y análisis de datos.....	21
3.5.	Desarrollo del proyecto.....	21
CAPÍTULO IV.....		23
LA PROPUESTA		23
4.1.	Análisis de requerimientos.....	23
4.1.1.	Identificar preguntas.....	23
4.1.2.	Identificar indicadores y perspectivas de análisis.....	24
4.1.3.	Diseñar Modelo Conceptual.....	26
4.1.4.	Definir ámbitos del sistema.....	28
4.2.	Establecimiento de los procesos transaccionales del sistema financiero.....	29
4.2.1.	Determinar Indicadores.....	29
4.2.2.	Establecer correspondencias.....	30
4.2.3.	Determinar el nivel de granularidad.....	39
4.2.4.	Diseñar el modelo conceptual ampliado.....	43
4.3.	Diseño del Modelo Lógico del Data Warehouse.....	45
4.3.1.	Establecer el tipo de modelo lógico a utilizar.....	46
4.3.2.	Crear tablas de dimensiones.....	46
4.3.3.	Construir las tablas de hechos.....	49
4.3.4.	Establecer las uniones correspondientes.....	52
4.3.5.	Creación de los cubos multidimensionales.....	53
4.4.	Desarrollo de los procesos de extracción, transformación y carga de datos....	64
4.4.1.	Diseñar proceso de carga inicial.....	64
4.4.2.	Establecer proceso de actualización de datos.....	74
4.5.	Desarrollo del módulo de Business Intelligence.....	78
4.5.1.	Establecer componentes adecuados para mostrar información.....	79
4.5.2.	Elaboración de la interfaz gráfica.....	81

4.5.3.	Implementación del módulo en el Sistema Financiero Financial.....	86
4.5.4.	Ejecución de Pruebas	89
CAPITULO V	98
CONCLUSIONES Y RECOMENDACIONES	98
5.1.	Conclusiones	98
5.2.	Recomendaciones	98
BIBLIOGRAFÍA O REFERENCIAS	100
ANEXOS	103

ÍNDICE DE FIGURAS

Figura 2.1. Metodología hefesto.....	6
Figura 2.2 esquema de estrella	12
Figura 2.3. Esquema de copo de nieve.	13
Figura 2.4. Esquema de constelación	14
Figura 2.5. Esquema de kpi	14
Figura 2.6. Modelo conceptual cartera	27
Figura 2.7. Modelo conceptual captaciones vista.....	27
Figura 2.8. Modelo conceptual captaciones plazo.....	28
Figura 2.9. Esquema de business intelligence [19]	28
Figura 2.10. Modelo entidad relación proceso de cartera	31
Figura 4.11. Correspondencias cartera	32
Figura 4.12.modelo entidad relación proceso de captaciones vista.....	34
Figura 4.13. Correspondencias captaciones vista.....	35
Figura 4.14.modelo entidad relación proceso de captaciones plaz	37
Figura 4.15. Correspondencias captaciones plazo.....	38
Figura 4.16. Modelo conceptual ampliado - cartera.....	44
Figura 4.17. Modelo conceptual ampliado – captaciones vista.....	44
Figura 4.18. Modelo conceptual ampliado – captaciones plazo.....	45
Figura 4.19. Perspectiva - clientes.....	47
Figura 4.20. Perspectiva – usuario oficial	47
Figura 4.21. Perspectiva – tiempo	47
Figura 4.22. Perspectiva - préstamos.....	48
Figura 4.23. Perspectiva – cuentas	49
Figura 4.24. Perspectiva – depósitos a plazo fijo	49
Figura 4.25. Tabla de hechos – cartera.....	50
Figura 4.26. Tabla de hechos – captaciones vista	51
Figura 4.27. Tabla de hechos – captaciones plazo	51
Figura 4.28. Uniones correspondientes	52

Figura 4.29. Crear nuevo proyecto	57
Figura 4.30. Estructura de proyecto	57
Figura 4.31. Conexiones.....	58
Figura 4.32. Cadena de conexión	58
Figura 4.33. Creación vistas del origen de datos.....	59
Figura 4.34. Selección de tablas y dimensiones	59
Figura 4.35. Finalización del asistente	60
Figura 4.36.diagrama de la vista de datos	60
Figura 4.37.selección tablas de hechos.....	61
Figura 4.38.selección medidas	61
Figura 4.39.selección dimensiones.....	62
Figura 4.40.resumen de creación de los cubos	62
Figura 4.41.jerarquías	62
Figura 4.42.generación kpi	63
Figura 4.43. Proceso cubos financiero.....	63
Figura 4.44. Módulo interfaz gráfica.....	78
Figura 4.45.control pivotgridcontrol	79
Figura 4.46.control chartcontrol	80
Figura 4.47.control popupmenu	80
Figura 4.48.control barcontrol.....	81
Figura 4.49.control conexión motor olap	81
Figura 4.50. Control pivotgridcontrol – agregar campos	82
Figura 4.52. Control pivotgridcontrol – agregar kpi	82
Figura 4.51. Control pivotgridcontrol – agregar grupos.....	82
Figura 4.53. Control chartcontrol- escoger datos	83
Figura 4.54. Diseño de interfaz	83
Figura 4.55. Funcionalidad menú cartera	87
Figura 4.56. Funcionalidad menú vista	88
Figura 4.57. Funcionalidad menú plazo	88
Figura 4.58. Rol menú funcionalidad	88
Figura 4.59. Módulo business intelligence.....	89

Figura 4.60. Tipo de préstamo el monto total entregado en el tiempo	89
Figura 4.61. Tipo de préstamo la cantidad de créditos entregados en el tiempo.....	90
Figura 4.62. Comparativa entre oficinas cartera	90
Figura 4.63. Destino financiero entre dos años	91
Figura 4.64. Oficiales de crédito en el tiempo.....	91
Figura 4.65. Cuentas por oficina y usuario.....	92
Figura 4.66. División política clientes vista	92
Figura 4.67. Cuentas por tipo de cliente.....	93
Figura 4.68. Tipos de depósitos a plazo fijo en el tiempo	93
Figura 4.69. Monto depósitos por ciudad oficinas	94
Figura 4.70. Cantidad depósitos por ciudad oficinas	94
Figura 4.71. Vista preliminar.....	95
Figura 4.72. Guardar diseño	95
Figura 4.73. Abrir diseño.....	96
Figura 4.74. Diseño excel cartera	96
Figura 4.75. Diseño excel plazo	97
Figura 4.76. Diseño excel vista	97

ÍNDICE DE TABLAS

Tabla 2.1: Tabla comparativa de las metodologías para BI.	9
Tabla 4.1 indicadores de los procesos	24
Tabla 4.3 perspectivas - cartera	33
Tabla 4.4 indicadores- cartera.....	33
Tabla 4.5 perspectivas- captaciones vista	36
Tabla 4.6 indicadores- captaciones vista	36
Tabla 4.7. Perspectivas- captaciones plazo.....	38
Tabla 4.8. Indicadores- captaciones plazo	38

RESUMEN EJECUTIVO

En toda empresa se toma decisiones diariamente y todas las decisiones tienen riesgos, por lo tanto para minimizar el riesgo es necesario un adecuado control de la cantidad y calidad de información que en el ámbito gerencial permita un análisis global que determine el rumbo de la misma.

“Business Intelligence” se ha convertido en un concepto necesario de discutir en este entorno y descubrir sus ventajas y la solución que puede dar a la hora de un efectivo control gerencial, transformando los datos en información y la información en conocimiento de forma que se pueda optimizar el proceso de toma de decisiones.

Sifizsoft S.A. es una empresa que desarrolla Software financiero contable y brinda soporte y soluciones informáticas a instituciones financieras en especial a cooperativas de ahorro y crédito con su Sistema Informático Financiero “Financial Business System” que posee una alta gama de funcionalidades que cubren con la mayoría de las necesidades de estas instituciones.

Al momento la empresa no cuenta con un módulo de Business Intelligence con herramientas que permitan una navegación dinámica de la información y pueda brindar el soporte necesario para la toma de decisiones gerenciales.

El presente tema de investigación propone utilizar la Metodología Hefesto para la creación del Data Warehouse utilizando como herramienta de minería de datos el Sistema Gestor de Base de Datos Sql Server y para el desarrollo de la interfaz de usuario componentes DevExpress.

El módulo de Business Intelligence proporcionará un análisis dinámico información actualizada, que permitirá tener un panorama general en los módulos de Cartera con la colocación de créditos, Captaciones Vista con su proceso de apertura de cuentas y Captaciones Plazo con sus depósitos a plazo fijo.

ABSTRACT

In every business decision is taken daily and all decisions have risks, so to minimize the risk requires a proper control of the quantity and quality of information management in the area allow a comprehensive analysis to determine the course of the same.

"Business Intelligence" has become a necessary concept to discuss in this environment and discover its advantages and can provide the solution for effective management control, transforming data into information and information into knowledge so that they can optimize the process decision-making.

Sifizsoft S.A. is a company that develops financial accounting software and provides support and solutions to financial institutions particularly unions credit with financial system "Financial Business System" that has a high range of functionalities that cover the majority of needs of these institutions.

At the moment the company does not have a module of Business Intelligence tools that allow dynamic navigation and can provide the information needed to support management decisions.

Present Research theme propose to use the Hefesto Methodology for creating the Data Warehouse using as a tool for Data Mining The System Manager Database SQL Server and to development User Interface components DevExpress.

Business Intelligence module provides a dynamic and user-friendly with updated information analysis, which will have an overview on the modules Portfolio with the placement of credits, deposits vista with its process of opening savings accounts and Term Deposits with investments or deposits.

GLOSARIO DE TÉRMINOS Y ACRÓNIMOS

Business Intelligence.- Se entiende por Business Intelligence al conjunto de metodologías, aplicaciones, prácticas y capacidades enfocadas a la creación y administración de información que permite tomar mejores decisiones a los usuarios de una organización [7].

Data Warehouse.- Es una colección de datos orientada al negocio, integrada, variante en el tiempo y no volátil para el soporte del proceso de toma de decisiones de la gerencia” [8].

Cubo OLAP.- Es una base de datos multidimensionales, que permiten procesar grandes volúmenes de información, en campos bien definidos, y con un acceso inmediato a los datos para su consulta y posterior análisis.

OLAP.- Define a una tecnología que se basa en el análisis multidimensional de los datos y que le permite al usuario tener una visión más rápida e interactiva de los mismos.

Esquema Multidimensional.- Es aquel que separa físicamente las medidas que cuantifican el negocio de los elementos descriptivos (también llamados dimensiones) que describen y categorizan el negocio [12].

Tablas de Dimensión.- Son las tablas que definen como están los datos organizados lógicamente y proveen el medio para analizar el contexto del negocio. Representan los ejes del cubo, y los aspectos de interés, mediante los cuales el usuario podrá filtrar y manipular la información almacenada en la tabla de hechos.

Tablas de Hechos.- Es la tabla central de un esquema multidimensional que contiene los valores de las medidas de negocio o indicadores de negocio.

Jerarquías.- Una jerarquía representa una relación lógica entre dos o más atributos pertenecientes a un cubo multidimensional; siempre y cuando posean su correspondiente relación “padre-hijo”.

Medidas.- Las Medidas, son datos numéricos de interés primario para los usuarios de un cubo multidimensional.

Proceso ETL.- Es el proceso que organiza el flujo de los datos entre diferentes sistemas en una organización y aporta los métodos y herramientas necesarias para mover datos desde múltiples fuentes a un almacén de datos, reformatearlos, limpiarlos y cargarlos en otra base de datos, data mart o bodega de datos.

SQL.- es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar de forma sencilla información bases de datos, así como hacer cambios en ellas.

Job SQL Server.- Procesos que permiten realizar tareas que se ejecuten según su configuración previa.

Procedimiento Almacenado.- Los procedimientos almacenados son grupos formados por instrucciones SQL y el lenguaje de control de flujo.

Script de base de datos.- Son archivos adicionales que contienen instrucciones SQL o utilidades como SQLCMD que no forman parte de la definición del esquema de base de datos.

INTRODUCCIÓN

El presente trabajo cuyo tema es “Módulo de Business Intelligence para el Sistema Financiero Financial de la empresa Sifizsoft-Ambato” consta de cinco capítulos que se detallan a continuación:

CAPÍTULO I. “El Problema”, en este capítulo se identifica el Problema a investigar que está aconteciendo en el contexto de la realidad, para lo cual delimitamos sus alcances, justificamos debidamente el problema y planteamos los objetivos, los mismos que serán la guía de todo el proyecto.

CAPÍTULO II. “Marco Teórico”, en este capítulo presenta los Antecedentes Investigativos y el fundamento teórico el cual ayuda a comprender de una forma clara el problema, para posteriormente plantear una propuesta de solución al mismo.

CAPÍTULO III. “Marco Metodológico”, se describen las Modalidades de Investigación a seguir para conseguir el resultado inicialmente propuesto, así como también la forma de cómo se recolectará y se procesará la información y finalmente el esquema a aplicar para el Desarrollo del Proyecto.

CAPÍTULO IV. “La Propuesta”, en este capítulo se describe el desarrollo de la propuesta de solución, definiendo los requisitos necesarios ante el problema planteado.

CAPÍTULO V. “Conclusiones y Recomendaciones”, en este capítulo se presenta las conclusiones que obtuvieron después de realizar la presente investigación además de realizar recomendaciones técnicas en beneficio del adecuado funcionamiento del sistema.

CAPÍTULO I

EL PROBLEMA

1.1. Tema

Módulo de Business Intelligence para el Sistema Financiero Financiamiento de la empresa Sifizsoft-Ambato.

1.2. Planteamiento del problema

1.2.1. Contextualización

“Business Intelligence” se ha convertido en uno de los factores más importantes a la hora de explorar y llevar el control absoluto de la información, para gestionarla de la manera más eficaz y entender la situación actual de la empresa, plantear posibles escenarios futuros y tomar decisiones basadas en el conocimiento obtenido.

Hoy en día las entidades financieras cumplen un papel fundamental en el desarrollo económico de las naciones, mediante la captación de recursos económicos de familias, empresas, sector público y su canalización en beneficio de los mismos. Los flujos de información que manejan dichas empresas son cada vez más complejos. A causa de diversos factores que afectan al mercado, hoy en día, organizar adecuadamente la información se vuelve imprescindible para quienes quieran posicionarse y hacerle frente a la competencia eficazmente.[1]

Se realizó una encuesta a las entidades financieras clientes de Sifizsoft en Ecuador, de veintiún instituciones encuestadas dieciséis no cuentan con un módulo de Business Intelligence, generando una desventaja frente a la competencia que cada vez es más eficiente mediante la innovación tecnológica e invirtiendo en sistemas inteligentes que les

permiten conocer su estado actual y tomar decisiones futuras que contribuyan al crecimiento empresarial.

En la provincia de Tungurahua se refleja un gran crecimiento que han tenido las cooperativas de ahorro y crédito en los últimos años, registrándose 167 instituciones financieras en la Superintendencia de Economía Popular y Solidaria al 2015 [2], creando un mercado cada vez más competitivo que exige herramientas efectivas para tomar decisiones, pero la realidad es que en la actualidad son muy pocas las entidades financieras que cuenten con un sistema de inteligencia de negocios. De siete instituciones clientes de Sifizsoft encuestadas, solo una dispone de un módulo de Business Intelligence (Anexo 2).

Sifizsoft es una empresa de desarrollo de software financiero contable escalable que se ajusta las necesidades de las Instituciones Financieras, y brinda soporte personalizado. Así la empresa cuenta con el Sistema Informático Financiero Financial 2.0, que cubre totalmente las necesidades de sus clientes tanto en procesos contables, financieros y de gestión de talento humano, con una tecnología que brinda seguridad, escalabilidad y agilidad en todo momento.

Una de las desventajas que actualmente presenta Financial 2.0, es que no dispone de un módulo de inteligencia de negocios que permita a las instituciones financieras realizar un análisis gerencial de control de los procesos de Cartera, Captaciones Vista y Plazo; y tomar decisiones oportunas con información relevante; esto hace que la empresa pierda terreno frente a sus competidores que han visto en la inteligencia de negocios una herramienta fundamental para mejorar la gestión gerencial y así ganar prestigio en el mercado.

1.3. Delimitación

Área académica: Software

Línea de Investigación: Desarrollo de Software

Sub línea de Investigación: Aplicaciones distribuidas.

Delimitación espacial: Sifizsoft S.A. de la ciudad de Ambato ubicado en la Av. Víctor Hugo y Atahualpa.

Delimitación temporal: La duración del proyecto es de 6 meses a partir de la fecha de aprobación del perfil por parte del Honorable Consejo Directivo de la facultad.

1.4. Justificación

El presente proyecto es de interés del investigador ya que domina la temática y por ser parte de la empresa está involucrado en los procesos financieros y cuenta con el apoyo total de la empresa.

El aporte teórico será importante ya que es un tema de actualidad que contribuirá a mejorar la gestión gerencial de las entidades financieras, para que brinden un mejor servicio a sus clientes y así ser un importante aporte a la comunidad.

La presente investigación toma relevancia para las entidades financieras ya que la información se procesa a través de mecanismos de “Business Intelligence”, de esta forma la alta dirección podrá contar con herramientas suficientes que le permitan una mejor gestión.

Con este nuevo módulo Sifizsoft resolverá el inconveniente de no contar con un módulo de inteligencia de negocios que permita a las instituciones financieras realizar un óptimo análisis gerencial en los módulos de mayor relevancia como son Cartera, Captaciones Vista y Plazo, lo cual le dará mayor prestigio y competitividad en el mercado.

La apertura de la empresa Sifizsoft S.A. y la necesidad que tienen sus clientes de un módulo de sistema de toma de decisiones hacen que sea totalmente factible desarrollar el tema propuesto.

1.5. Objetivos

1.5.1. Objetivo general

Implementar un Módulo de Business Intelligence con componentes DevExpress, para el Sistema Financiero Financial de la empresa Sifizsoft-Ambato.

1.5.2. Objetivos específicos

- Analizar los requerimientos necesarios para el desarrollo de un nuevo módulo de Business Intelligence para el sistema financiero Financial de la empresa Sifizsoft-Ambato.
- Establecer los principales procesos transaccionales del sistema financiero Financial de la empresa Sifizsoft-Ambato.
- Diseñar el modelo lógico del DataWarehouse para el módulo de Business Intelligence del sistema financiero Financial de la empresa Sifizsoft-Ambato.
- Desarrollar los procesos de extracción, transformación y carga de datos para el módulo de Business Intelligence del sistema financiero Financial de la empresa Sifizsoft-Ambato.
- Desarrollar el módulo de Business Intelligence del sistema financiero Financial de la empresa Sifizsoft-Ambato.

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes Investigativos

El “Business Intelligence” es una solución efectiva a los problemas de recuperación de cartera, el cual posee todas las características necesarias para administrar y mejorar los procesos de la empresa, y a la vez reducirá el tiempo en la toma de decisiones a nivel gerencial, para una mejor forma de recuperación de cartera y así tomar decisiones de manera oportuna. [3].

“Se puede concluir que un sistema de inteligencia de negocios puede mejorar los procesos de la empresa, agilizando las ventas y entregando información confiable y oportuna para la alta gerencia y así tomar decisiones ágiles para brindar un mejor servicio a su clientes.” [4].

“La implementación del BIR(Herramienta Business Intelligence) permitió generar informes para analizar el impacto de los créditos y simplificar la tarea emisión de reportes para el Departamento de Crédito, permitiendo ganar tiempo para gestionar otros procesos”[5].

“Un buen diseño del Data Warehouse puede optimizar las consultas reflejadas en tiempos de respuesta obteniendo datos relevantes para el análisis del negocio” [6].

2.2. Fundamentación Teórica

2.2.1. Business Intelligence

Se entiende por Business Intelligence al conjunto de metodologías, aplicaciones, prácticas y capacidades enfocadas a la creación y administración de información que permite tomar mejores decisiones a los usuarios de una organización [7].

2.2.2. Metodología Hefesto

HEFESTO es una metodología, cuya propuesta está fundamentada en una muy amplia investigación, comparación de metodologías existentes y experiencias propias en procesos de confección de almacenes de datos.

La idea principal, es comprender cada paso que se realizará, para no caer en el tedio de tener que seguir un método al pie de la letra sin saber exactamente qué se está haciendo, ni por qué.

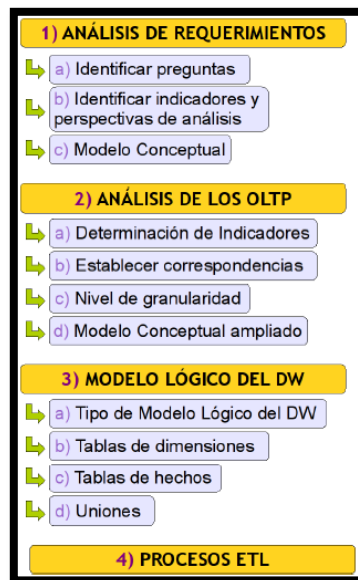


Figura 2.1. Metodología Hefesto.

Fases

Análisis de Requerimientos

Lo primero será identificar los requerimientos del usuario a través de preguntas que expliciten los objetivos de su organización. Luego, se analizarán las estas preguntas a fin de identificar cuáles serán los indicadores y perspectivas que serán tomadas en cuenta para la construcción del DW. Finalmente se confeccionará un modelo conceptual en donde se podrá visualizar el resultado obtenido en este primer paso.

Análisis de los OLTP

Seguidamente, se analizarán las fuentes OLTP para determinar cómo serán calculados los indicadores y para establecer las respectivas correspondencias entre el modelo conceptual creado en el paso anterior y las fuentes de datos.

Luego, se definirán qué campos se incluirán en cada perspectiva. Finalmente, se ampliará el modelo conceptual con la información obtenida en este paso.

Modelo Lógico del DW

A continuación, se confeccionará el modelo lógico de la estructura del DW, teniendo como base el modelo conceptual que ya ha sido creado.

Para ello, primero se definirá el tipo de modelo que se utilizará y luego se llevarán a cabo las acciones propias al caso, para diseñar las tablas de dimensiones y de hechos. Finalmente, se realizarán las uniones pertinentes entre estas tablas.

Procesos ETL

Una vez construido el modelo lógico, se deberá proceder a probarlo con datos, a través de procesos ETL.

Al generar los ETL, se debe tener en cuenta cual es la información que se desea almacenar en el depósito de datos, para ello se pueden establecer condiciones adicionales y restricciones. Estas condiciones deben ser analizadas y llevadas a cabo con mucha prudencia para evitar pérdidas de datos importantes [8].

2.2.3. Comparación de Hefesto con otras metodologías

Bajo el concepto del desarrollo de soluciones de inteligencia de negocio, se tienen diversas guías para la construcción de proyectos BI, tanto para el análisis, diseño, construcción e implementación; pero muy pocas consolidadas como metodologías. Entre las metodologías más conocidas tenemos las documentadas por Ralph Kimball, Bill Inmon, Bernabeu Ricardo Dario (Hefesto) 2 y la propuesta por el instituto SAS (SEMMA). Para seleccionar la metodología a ser implementada en el proyecto de SATB, se realizó un análisis comparativo entre ellas y se representan en la siguiente tabla. [9]

No.	Factores de análisis Puntaje Bueno/Precisa/Alto=3, Semiprecisas/Medio=2, Costoso/No precisa/Bajo=1, Nulo/NO=0,TODOS/SI=1	Ralph Kimball	Bill Inmon	Ricardo Bernabeu (Hefesto)	SAS
1	Flexibilidad	Medio (2)	Alto (3)	Alto (3)	Baja (1)
2	Adaptable sobre cualquier tecnología	SI (1)	SI (1)	SI (1)	NO (0)
3	Afinidad con el sistema actual en desarrollo	Medio (2)	Medio (2)	Alto (3)	Medio (2)
4	Comunicación con el cliente.	Alto (3)	Alto (3)	Alto (3)	Alto (3)
5	Tamaño del Proyecto	Todos (1)	Todos (1)	Pequeño/Mediano (1)	Mediano/Grande (1)
6	Tiempo en el análisis y diseño	Costoso por ser iterativo (1)	Medio una sola vez (2)	Medio una sola vez (2)	Costoso por ser iterativo (1)

7	Tiempo en construcción	Costoso (1)	Medio (2)	Bueno (3)	Medio (2)
8	Etapa de implantación	SI (1)	SI (1)	NO (0)	SI (1)
9	Guías y prácticas se aplican a SQL	SI (1)	SI (1)	SI (1)	Algunas (0)
10	Fácil entendimiento principiantes	NO (0)	NO (0)	SI (1)	NO (0)
11	Revisión Post Implantación	SI (1)	SI (1)	NO (0)	SI (1)
12	Documentación precisa	Precisa (3)	Precisa (3)	Precisa (3)	SemiPrecisa (3)
13	Perspectiva	Estrella (2)	Relacional (1)	Estrella/Cop o Nieve (3)	Estrella (1)
14	Rápido acceso en reportes	Alta (3)	Baja (1)	Media (2)	Alta (3)
15	Más usada en el mundo	Baja (1)	Alta (3)	Alta (3)	Baja (1)
	Total	23	25	29	20

Tabla 2.1: Tabla comparativa de las metodologías para BI.

Una vez realizada la comparación entre las cuatro metodologías podemos seleccionar Hefesto del autor Ricardo Bernabeu como la metodología más compatible para desarrollar el módulo de Business Intelligence para la empresa Sifizsoft. Los factores más predominantes fueron: Afinidad con el sistema actual en desarrollo, perspectiva y su utilización en el mundo.

2.2.4. Data Warehousing

Es el encargado de extraer, transformar, consolidar, integrar y centralizar los datos que la empresa genera en todos los ámbitos de su actividad diaria de negocios (compras, ventas, producción, etc.) y/o información externa relacionada. Permitiendo de esta manera el acceso y exploración de la información requerida, a través de una amplia gama de posibilidades de análisis multivariados, con el objetivo final de dar soporte al proceso de toma de decisiones estratégico y táctico.

Data Warehouse

Un Data Warehouse es una colección de datos orientada al negocio, integrada, variante en el tiempo y no volátil para el soporte del proceso de toma de decisiones de la gerencia” [8].

2.2.5. Cubo OLAP

Es un mecanismo para buscar datos con rapidez y tiempo de respuesta uniforme independientemente de la cantidad de datos en el cubo o la complejidad del procedimiento de búsqueda.

Los cubos son subconjuntos de datos de un almacén de datos, organizado y sumariado dentro de una estructura multidimensional. Los datos se sumarizan de acuerdo a factores de negocio seleccionados, proveyendo el mecanismo para la rápida y uniforme tiempo de respuesta de las complejas consultas [10].

2.2.6. Arquitectura OLAP

La tecnología OLAP tiene por arquitectura típica aquella donde el almacén de datos constituye el repositorio de datos al que acceder a través de las herramientas OLAP.

Esto hace que el acceso sea fácil y eficiente, mediante una vista multidimensional de los datos de las fuentes con una configuración cliente/servidor, y que se puedan dar operaciones de navegación complejas sobre los datos. Sin embargo, también cabe la posibilidad de que las herramientas de explotación OLAP accedan directamente a las fuentes de datos para recoger la información necesaria para el análisis [11].

2.2.7. ETL

Este término viene de inglés de las siglas Extract-Transform-Load que significan Extraer, Transformar y Cargar y se refiere a los datos en una empresa. ETL es el proceso que

organiza el flujo de los datos entre diferentes sistemas en una organización y aporta los métodos y herramientas necesarias para mover datos desde múltiples fuentes a un almacén de datos, reformatearlos, limpiarlos y cargarlos en otra base de datos, data mart o bodega de datos. ETL forma parte de la Inteligencia Empresarial (Business Intelligence), también llamado “Gestión de los Datos” (Data Management) [12].

2.2.8. Tablas y entidades de dimensiones

Las tablas de dimensiones describen los distintos aspectos de un proceso de negocio. Por ejemplo, si desea determinar los objetivos de ventas, puede almacenar los atributos de dichos objetivos en una tabla de dimensiones.

Las tablas de dimensiones agrupan los datos en la base de datos cuando el negocio crea informes. Por ejemplo, puede agrupar objetivos de ventas por país, producto o minorista, y dichas agrupaciones se almacenarán en tablas de dimensiones[13].

2.2.9. Tablas de Hechos

Una tabla de hechos o una entidad de hecho es una tabla o entidad de un esquema de estrella o copo de nieve que almacena medidas para medir el negocio, como las ventas, el coste de las mercancías o las ganancias.

Las tablas y entidades de hechos agregan medidas o los datos numéricos de un negocio. Para medir los datos de una tabla o entidad de hechos, todas las medidas de una tabla o entidad de hechos debe corresponder al mismo grano [13].

2.2.10. Esquemas Dimensionales

Un esquema dimensional es aquel que separa físicamente las medidas que cuantifican el negocio de los elementos descriptivos (también llamados dimensiones) que describen y categorizan el negocio. El esquema dimensional puede ser físico o lógico. Un esquema

dimensional físico generalmente se representa en forma de esquema de estrella o de copo de nieve, en el que los objetos que contiene son en realidad tablas de base de datos.

El esquema dimensional puede incluso adoptar la forma de una sola tabla o vista, en la que todos los hechos y dimensiones están en columnas distintas de dicha tabla o vista. En un esquema dimensional lógico, los hechos, las medidas y las dimensiones se representan como entidades y atributos independientes a un proveedor de base de datos y, por lo tanto, se pueden transformar en un esquema dimensional físico para cualquier proveedor de base de datos [13].

2.2.11. Esquema de Estrella

Un esquema de estrella es un tipo de esquema de base de datos relacional que consta de una sola tabla de hechos central rodeada de tablas de dimensiones.

En la siguiente figura se muestra un esquema de estrella con una sola tabla de hechos y cuatro tablas de dimensiones. Un esquema de estrella puede tener cualquier número de tablas de dimensiones. Las ramas situadas al final de los enlaces que conectan las tablas indican una relación de muchos a uno entre la tabla de hechos y cada tabla de dimensiones [13].

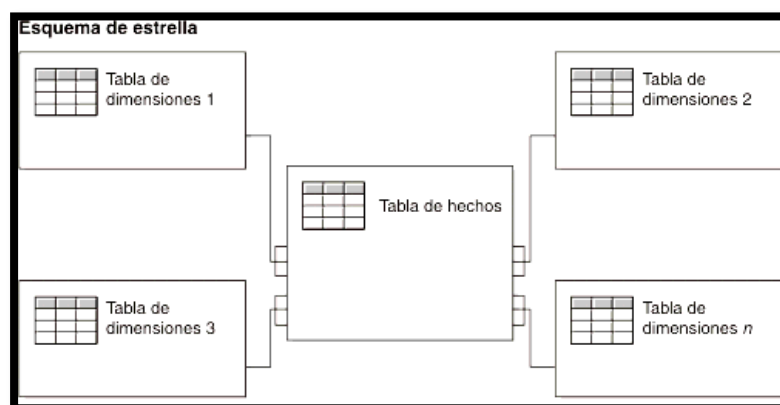


Figura 2.2 Esquema de estrella

2.2.12. Esquema de Copo de Nieve

El esquema de copo de nieve consta de una tabla de hechos que está conectada a muchas tablas de dimensiones, que pueden estar conectadas a otras tablas de dimensiones a través de una relación de muchos a uno.

Las tablas de un esquema de copo de nieve generalmente se normalizan en el tercer formulario de normalización. Cada tabla de dimensiones representa exactamente un nivel en una jerarquía.

En la siguiente figura se muestra un esquema de copo de nieve con dos dimensiones, cada una con tres niveles. Un esquema de copo de nieve puede tener varias dimensiones y cada dimensión puede tener varios niveles [13].

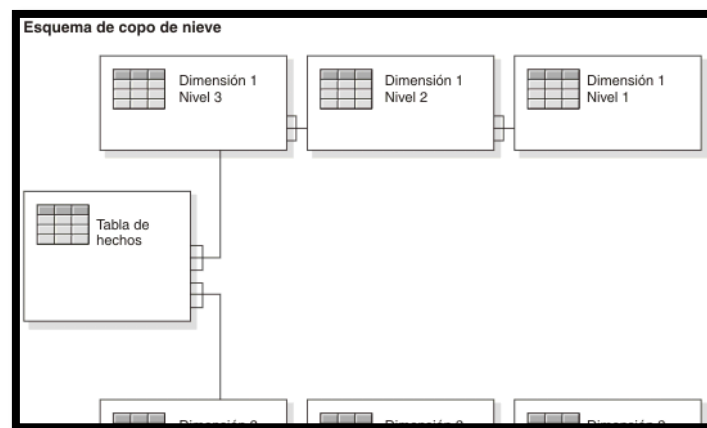


Figura 2.3. Esquema de copo de nieve.

2.2.13. Esquema de Constelación

Un esquema de constelación es una combinación de un esquema de estrella y un esquema de copo de nieve. Los esquemas de constelación son esquemas de copo de nieve en los que sólo algunas de las tablas de dimensiones se han desnormalizado.

El objetivo de los esquemas de constelación es aprovechar las ventajas de los esquemas de estrella y de copo de nieve. Las jerarquías de los esquemas de estrella están

desnormalizadas, mientras que las jerarquías de los esquemas de copo de nieve están normalizadas.

Los esquemas de constelación están normalizados para eliminar las redundancias de las dimensiones [13].

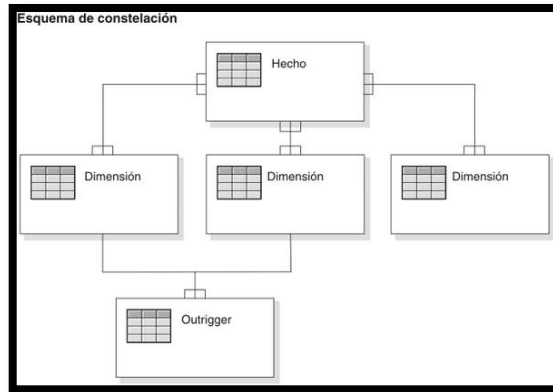


Figura 2.4. Esquema de constelación

2.2.14. Key Performance Indicator (KPI)

Los indicadores clave de rendimiento (KPI) ayudan a las empresas a entender lo bien que se está realizando el trabajo en relación con sus metas y objetivos estratégicos.

En un sentido más amplio, un KPI proporciona la información de rendimiento más importante que permite a las partes interesadas saber si se va por buen camino.

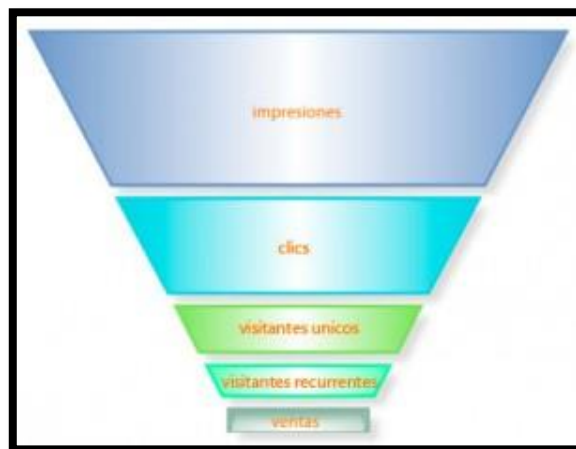


Figura 2.5. Esquema de KPI

Medir lo más importante

Los KPI's sirven para reducir la complejidad del desempeño de una empresa, a un pequeño número de indicadores clave, a fin de hacerla más entendible.

Este es el mismo enfoque que utilizamos en nuestra vida cotidiana. Un ejemplo claro:

Si vamos al médico, para analizar nuestra salud, nuestro doctor utiliza varios indicadores para comprender como nos encontramos, tales como; la presión arterial, nuestro peso, la altura, etc.

Y así es como un doctor es capaz de analizar la complejidad de un cuerpo humano. Pues bien, esto es lo que se trata de conseguir en una empresa para determinar sus KPI's [14].

2.2.15. Cartera Financiera

Comprende los saldos de capital de las operaciones de crédito otorgados por una entidad, bajo las distintas modalidades autorizadas y en función al giro especializado que le corresponde a cada una de ellas, incluye todas las operaciones otorgadas con recursos propios o con fuentes de financiamiento interno o externo, operaciones contingentes pagadas por la institución por incumplimiento de los deudores principales, contratos de arrendamiento mercantil financiero, sobregiros en cuentas corrientes de los clientes, valores por cobrar a tarjetahabientes; así como los montos provisionados para cubrir créditos de dudosa recuperación, de acuerdo a las disposiciones establecidas por la Superintendencia de Bancos [15].

2.2.16. Depósitos a la Vista

Se refiere a los depósitos monetarios de los gobiernos provinciales y locales, sociedades públicas no financieras (empresas), otras sociedades no financieras (empresas) y otros

sectores residentes (hogares) en cuenta corriente, cheques de gerencia y cheques certificados en el BCE, bancos privados y en el Banco Nacional de Fomento [15].

2.2.17. Depósitos a plazo fijo

Registran las obligaciones a cargo de la entidad derivada de la captación de recursos del público exigibles al vencimiento de un período no menor de treinta días, libremente convenido por las partes. Pueden instrumentarse en un título valor, nominativo, a la orden o al portador y pueden ser pagados antes del vencimiento del plazo, previo acuerdo entre el acreedor y el deudor [15].

2.2.18. SQL Server Analysis Services

Microsoft SQL Server Analysis Services (SSAS) ofrece funciones de procesamiento analítico en línea (OLAP) y minería de datos para aplicaciones de Business Intelligence. Analysis Services admite OLAP y permite diseñar, crear y administrar estructuras multidimensionales que contienen datos agregados desde otros orígenes de datos, como bases de datos relacionales.

En el caso de las aplicaciones de minería de datos, Analysis Services permite diseñar, crear y visualizar modelos de minería de datos que se construyen a partir de otros orígenes de datos mediante el uso de una gran variedad de algoritmos de minería de datos estándar del sector [16].

2.2.19. DevExpress

DevExpress es una de las más completas suites de componentes de UI para el desarrollo en todas las plataformas de .NET como Windows Forms, ASP.NET, MVC, Silverlight y Windows 8 XAML.

Ventajas

- Cuenta con controles para todas las plataformas de Microsoft Windows.
- Posee más de 70 controles mediante los cuales se pueden diseñar aplicaciones de alta complejidad.
- La creación de los componentes es semi – automática, DevExpress se encarga de realizar todo el código necesario para la visualización y llenado de los componentes según la plataforma utilizada.
- Permite llenar de manera sencilla cada uno de los componentes con información traída de una conexión de base de datos.
- Mejora el rendimiento de las aplicaciones al optimizar el código de llenado de las vistas.
- Funciona en cualquier explorador. (Aplicaciones Web)
- El tiempo de desarrollo se reduce considerablemente al no tener que teclear todo el código.
- Incluye plantillas predefinidas para varios tipos de aplicaciones.
- Incluye la herramienta Theme Builder, la cual permite editar el estilo de los controles y genera automáticamente la hoja de estilos de la aplicación siguiendo el patrón elegido.
- La documentación que incluye tanto en Visual Studio (descripciones de métodos y parámetros) como externa (documentación de clases y ejemplos) es realmente extensa y útil.
- Incluye video tutoriales y demostraciones sencillas de comprender.
- Es altamente personalizable.
- Cuenta con soporte rápido y eficaz para los problemas que se puedan presentar [17].

2.2.20. Financial Business System v2.0

Sistema financiero administrativo contable de última tecnología desarrollado actualmente con Visual Studio 2013 en un lenguaje C# con tecnología WCF, orientado a gestionar las

operaciones del negocio financiero, con especial atención para Cooperativas de Ahorro y Crédito e Instituciones Financieras [18].

Los módulos que oferta con su Sistema Financiero Financial 2.0 se descompone de la siguiente manera:

Módulos Básicos:

- Seguridad
- Auditoría
- Personas y Clientes
- Contabilidad (en línea e integrada con todos los módulos Financieros)

Módulos de Negocio:

- Captaciones a la Vista
- Captaciones a Plazo
- Crédito
- Cartera
- Cobranzas
- Cajas
- Contabilidad
- Reportes a Organismos de Control A (SRI – COSEDE – UAF- Otros de acuerdo al país donde opera nuestro cliente)

Módulos Opcionales:

- Anexo Transaccional (SRI)
- Interfaz Cajeros Automáticos
- Interfaz Ventanillas Compartidas
- Tesorería
- Portafolio de Inversiones
- Activos Fijos
- Control Presupuestario

- Obligaciones Financieras
- Add-in Excel
- Información Gerencial e Indicadores
- Huellas Dactilares
- Riesgo de Liquidez y Mercado
- Riesgo Operativo
- Lavado de Activos
- Banca Virtual
- Reportes a Organismos de Control B (SBS – SEPS- Otros de acuerdo al país donde opera nuestro cliente)

2.3. Propuesta de Solución

Se implementará un módulo de Business Intelligence en el Sistema Financiero Financial V2.0 que se presentará en el grupo de módulos opcionales, cuyo propósito será brindar información financiera gerencial depurada y actualizada, necesaria para la toma de decisiones en base a datos estadísticos precisos y oportunos.

CAPÍTULO III

METODOLOGÍA

3.1. Modalidad de la investigación

La presente investigación considerará las modalidades detalladas de la siguiente forma:

- De campo porque se acudirá al lugar de los hechos, es decir en la empresa Sifizsoft de la ciudad de Ambato. como también aplicará una entrevista y observación en el mismo campo.
- Bibliográfica-Documental, ya que se sustentará en revistas, libros, internet, tesis, como fuentes de información y para comparar información con el tema a investigar.
- Aplicada porque se ofrecerá una solución eficiente a un problema práctico en base a conocimientos adquiridos.

3.2. Población y muestra

Debido al tipo del proyecto no es necesario población.

3.3. Recolección de información

Para la recolección, procesamiento y análisis de la información se aplicará el método de observación y entrevista, para conocer de mejor manera la problemática que ocurre y poder determinar los procesos a mejorar.

La recolección de información se llevará a cabo en la empresa Sifizsoft de la ciudad de Ambato, se realizará una entrevista al Gerente y una encuesta a una muestra de los clientes de la misma.

También se usará información recolectada a través de libros e internet.

3.4. Procesamiento y análisis de datos

Para el procesamiento y análisis de la información se aplicará los siguientes procedimientos:

1. Organizar y analizar la información obtenida.
2. Identificar las principales áreas funcionales que son determinantes en el éxito o fracaso de las entidades financieras.
3. Identificar la información que será necesaria para una toma de decisiones ágil y oportuna.

3.5. Desarrollo del proyecto

1. Análisis de requerimientos
 - 1.1. Identificar preguntas.
 - 1.2. Identificar indicadores y perspectivas de análisis.
 - 1.3. Diseñar Modelo Conceptual.
 - 1.4. Definir ámbitos del sistema.
 - 1.5. Estimar costos del proyecto.
2. Establecimiento de los procesos transaccionales del sistema financiero
 - 2.1. Determinar Indicadores.
 - 2.2. Establecer correspondencias.
 - 2.3. Determinar el nivel de granularidad.
 - 2.4. Diseñar el modelo conceptual ampliado.
3. Diseño del Modelo Lógico del Data Warehouse
 - 3.1. Establecer el tipo de modelo lógico a utilizar.

- 3.2. Crear tablas de dimensiones.
- 3.3. Construir las tablas de hechos.
- 3.4. Establecer las uniones correspondientes.
- 3.5. Creación de los cubos multidimensionales.
- 4. Desarrollo de los procesos de extracción, transformación y carga de datos
 - 4.1. Diseñar proceso de carga inicial.
 - 4.2. Establecer proceso de actualización de datos.
- 5. Desarrollo del módulo de Business Intelligence
 - 5.1. Establecer componentes gráficos adecuados para mostrar información.
 - 5.2. Elaboración de la interfaz gráfica.
 - 5.3. Implementación del módulo en el Sistema Financiero Financiamiento
 - 5.4. Ejecución de pruebas.

CAPÍTULO IV

LA PROPUESTA

4.1. Análisis de requerimientos

4.1.1. Identificar preguntas.

Se realizó una entrevista (Anexo 1) al gerente de la empresa Sifizsoft con el objetivo de identificar las áreas específicas en los que se requiere obtener información necesaria para la toma de decisiones gerenciales, así se identificó las áreas de Cartera, Captaciones Vista y Captaciones Plazo por ser las de mayor procesamiento de datos y de mayor importancia financiera y contable para las entidades financieras.

A continuación se analizó que información puntual es requerida por cada área para un análisis gerencial:

Cartera:

- Colocación de los préstamos que comprende conocer los clientes que tuvieron préstamos en un determinado tiempo, la cantidad y la sumatoria de los montos de las deudas iniciales; además de saber el tipo de crédito, destino financiero, calificación contable, la oficina en la que se realizó el mismo, el oficial de crédito responsable, el tipo de cliente y la ubicación de su residencia.

Captaciones Vista:

- Se desea conocer la cantidad de cuentas creadas por cliente, tipo de cuenta, oficina, usuario oficial al igual que el tipo de cliente y la ubicación de su residencia.

Captaciones Plazo:

- Para las inversiones se necesita saber la cantidad y el monto total de depósitos a plazo fijo por cliente, oficina, usuario oficial y tipo de depósito.

A continuación se consultó cuáles serían los indicadores que representan cada uno de los procesos, y se obtuvo como respuesta el siguiente resumen.

Cartera	Captaciones Vista	Captaciones Plazo
<ul style="list-style-type: none">• Monto de los Crédito.• Cantidad de créditos.	<ul style="list-style-type: none">• Cantidad de cuentas aperturadas.	<ul style="list-style-type: none">• Monto total de depósitos a plazo fijo.• Cantidad de depósitos a plazo fijo.

Tabla 4.1 Indicadores de los procesos

Como se puede apreciar, las necesidades de información expuestas están acorde a los objetivos y estrategias de la empresa, ya que es precisamente esta información requerida la que proveerá un ámbito para la toma de decisiones, que en este caso permitirá analizar el comportamiento de los clientes sujetos de créditos, de ahorro y de inversiones a los que se pretende satisfacer ampliamente, para así lograr obtener una ventaja competitiva y maximizar las ganancias.

4.1.2. Identificar indicadores y perspectivas de análisis.

Una vez se ha establecido las preguntas claves, se procederá a la descomposición de la información obtenida para descubrir los indicadores que se evaluarán en función de las correspondientes perspectivas de análisis.

A continuación, se analizarán las preguntas obtenidas en el paso anterior y se detallarán cuáles son sus respectivos indicadores y perspectivas.

Cartera

- Cantidad de créditos por cada tipo de préstamo a cada cliente en un tiempo determinado.
- Monto total entregado por cada tipo de préstamo a cada cliente en un tiempo determinado.
- Cantidad de créditos por cada oficial de crédito a cada cliente en un tiempo determinado.
- Monto total entregado por cada oficial de crédito a cada cliente en un tiempo determinado.

Captaciones Vista

- Cantidad de cuentas aperturadas por tipo de cuenta a cada cliente en un tiempo determinado.
- Cantidad de cuentas aperturadas por oficial de Captaciones Vista en un tiempo determinado.

Captaciones Plazo

- Cantidad de depósitos a plazo fijo por cada tipo de depósito a cada cliente en un tiempo determinado.
- Monto total de depósitos a plazo fijo por cada tipo de depósito a cada cliente en un tiempo determinado.
- Cantidad de depósitos a plazo fijo realizados por oficial de Captaciones Plazo en un tiempo determinado.
- Monto total de depósitos a plazo fijo entregados por oficial de Captaciones Plazo en un tiempo determinado.

En síntesis obtendremos los siguientes indicadores y perspectivas para cada módulo:

Cartera

Indicadores

- Cantidad de créditos.

- Montos entregados.

Perspectivas

- Cliente.
- Usuario oficial.
- Tiempo.
- Préstamos.

Captaciones Vista

Indicadores

- Cantidad de cuentas.

Perspectivas

- Cliente.
- Usuario oficial.
- Cuentas.
- Tiempo.

Captaciones Plazo

Indicadores

- Cantidad de Depósitos a Plazo Fijo.
- Monto de Depósitos a Plazo Fijo.

Perspectivas

- Cliente.
- Usuario oficial.
- Depósitos a plazo fijo.
- Tiempo.

4.1.3. Diseñar Modelo Conceptual.

A continuación se construirá un modelo conceptual a partir de los indicadores y perspectivas obtenidas en el paso anterior.

A través de este modelo, se podrá observar con claridad cuáles son los alcances del módulo, para luego poder trabajar sobre ellos, además al poseer un alto nivel de definición de los datos, permite que pueda ser presentado ante los usuarios y explicado con facilidad.

MODELO CONCEPTUAL CARTERA

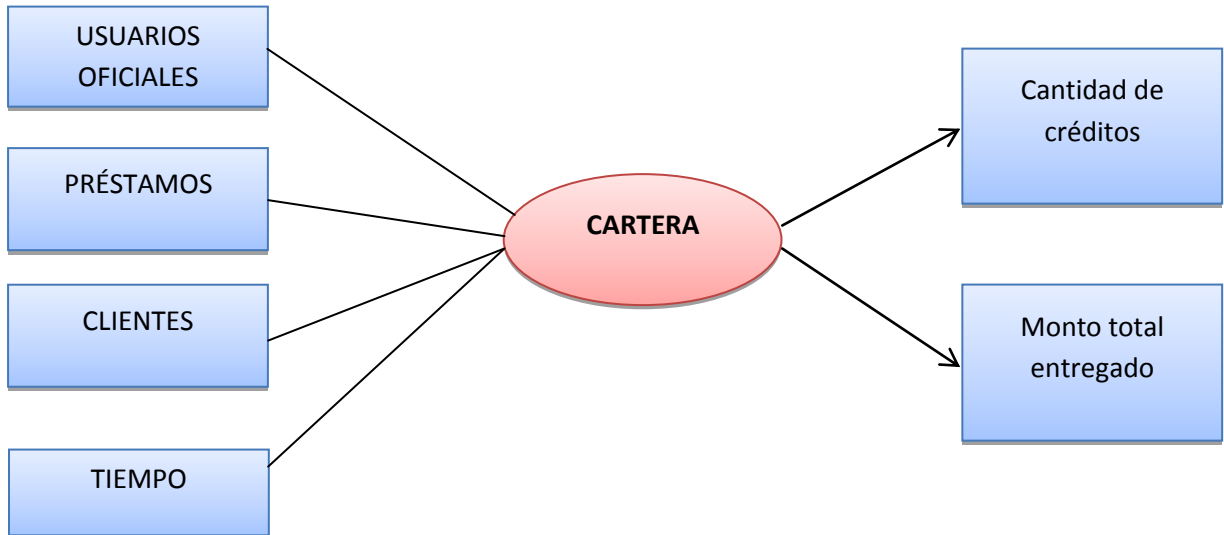


Figura 2.6. Modelo Conceptual Cartera

MODELO CONCEPTUAL CAPTACIONES VISTA

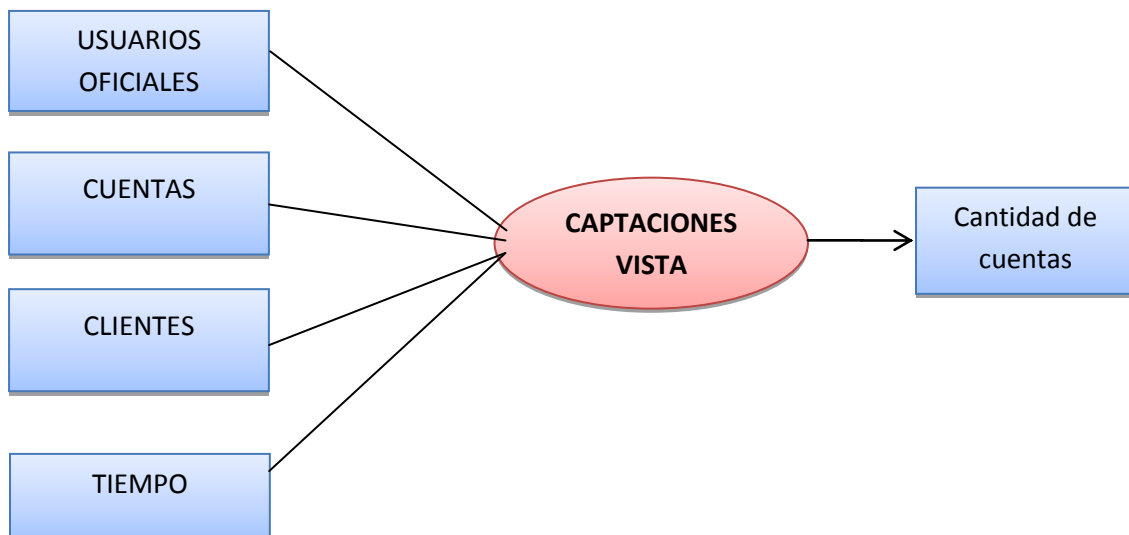


Figura 2.7. Modelo Conceptual Captaciones Vista

MODELO CONCEPTUAL CAPTACIONES PLAZO

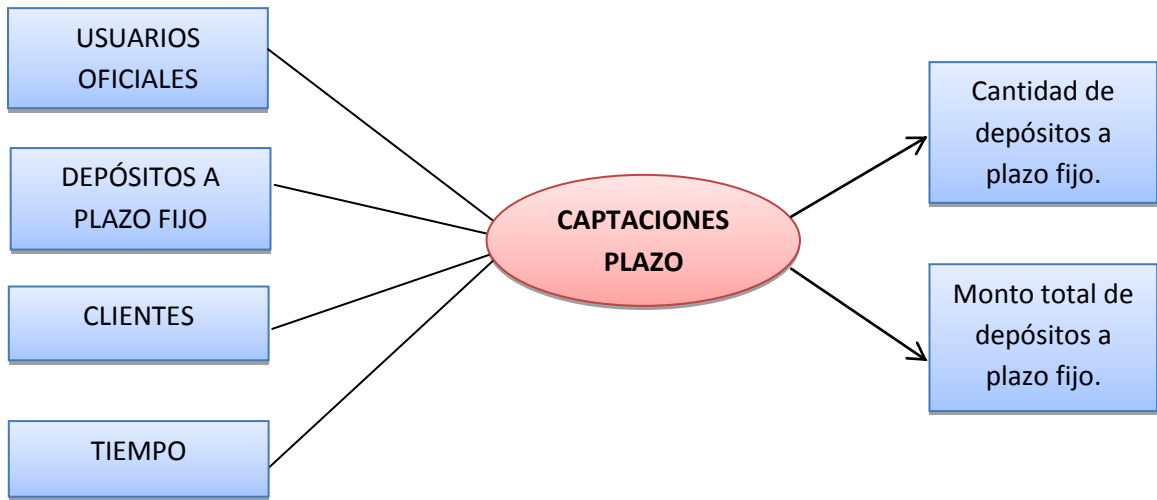


Figura 2.8. Modelo Conceptual Captaciones Plazo

4.1.4. Definir ámbitos del sistema.

La arquitectura se desarrollará en base al siguiente esquema de Business Intelligence.

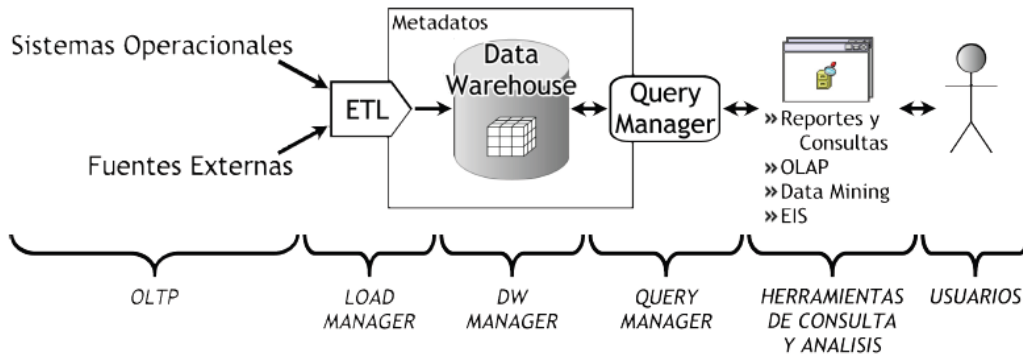


Figura 2.9. Esquema de Business Intelligence [19]

Proceso ETL.- Este proceso se realizará mediante Jobs que ejecutarán diariamente una secuencia de pasos para actualizar el Data Warehouse y procesar los cubos creados por cada módulo.

Data Warehouse.- Este almacén de datos se realizará mediante cubos multidimensionales generados en Analysis Services.

Reportes y consultas.- Los reportes y consultas se realizarán a través de componentes DevExpress que maneja herramientas de consulta de Business Intelligence.

4.2. Establecimiento de los procesos transaccionales del sistema financiero

Se analizó las fuentes de los procesos del Sistema Financiero Financiero v2.0, para determinar cómo serán calculados los indicadores y establecer las respectivas correspondencias entre el modelo conceptual y la base de datos. A continuación se definirá que campos se incluirán en cada perspectiva, y finalmente se obtendrá el modelo conceptual ampliado.

4.2.1. Determinar Indicadores.

Los indicadores se calcularán de la siguiente manera:

Cartera

Indicador: Cantidad de créditos.

- Hechos: Créditos entregados.
- Función de conteo: COUNT.

Aclaración: el indicador “Cantidad de créditos” representa los créditos entregados a un cliente en un tiempo determinado.

Indicador: Monto total entregado

- Hechos: Valor de la deuda inicial.
- Función de sumarización: SUM.

Aclaración: el indicador Monto total entregado será la sumatoria de la deuda inicial contraída por cliente.

Captaciones Vista

Indicador: Cantidad de cuentas.

- Hechos: Cuentas creadas.
- Función de conteo: COUNT.

Aclaración: el indicador “Cantidad de cuentas” representa el número de cuentas entregadas a un cliente en un tiempo determinado.

Captaciones Plazo

Indicador: Cantidad de Depósitos a Plazo Fijo.

- Hechos: Depósitos a plazo fijos.
- Función de conteo: COUNT.

Aclaración: el indicador “Cantidad de Depósitos a Plazo Fijo” los Depósitos a Plazo Fijo realizados por un cliente en un tiempo determinado.

Indicador: Monto Total de Depósito a plazo fijo.

- Hechos: Valor total del DPF.
- Función de sumarización: SUM.

Aclaración: el indicador Valor total del Depósito a plazo fijo será la sumatoria de los montos de los depósitos a plazo fijo realizadas por el cliente.

4.2.2. Establecer correspondencias.

En este punto se analizará la estructura de la base de datos del Sistema Financiero Financiero v2.0 para establecer las respectivas correspondencias con el modelo conceptual.

Correspondencias de Cartera

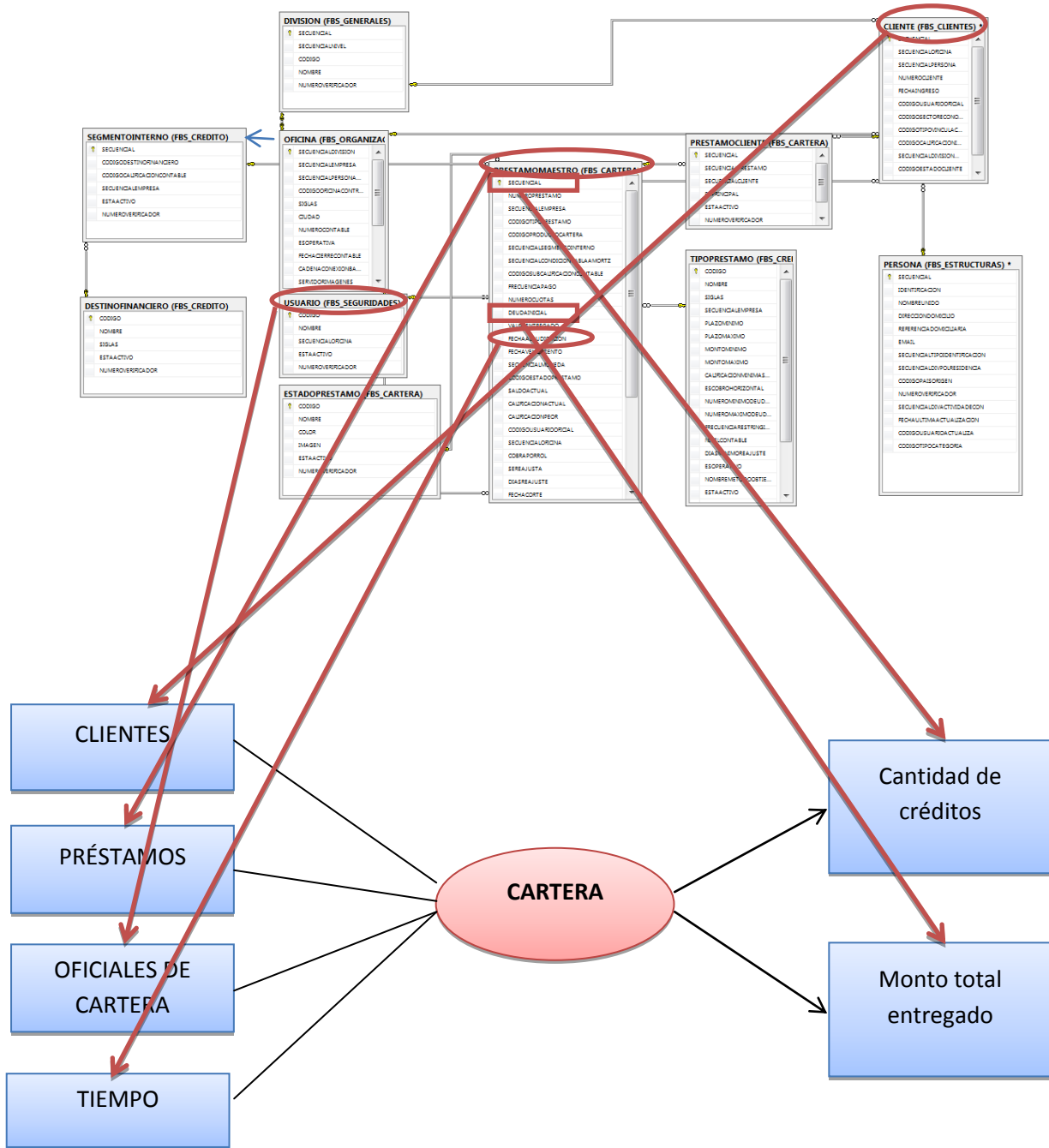


Figura 4.11. Correspondencias Cartera

Las relaciones identificadas fueron las siguientes:

TABLA	PERSPECTIVA
CLIENTE	Clientes
PRESTAMOMAESTRO	Préstamos
USUARIO	Oficiales de Crédito
FECHAADJUDICACION	Tiempo

Tabla 4.3 Perspectivas - Cartera

CAMPO	INDICADOR
SECUENCIAL(PRESTAMOMAESTRO)	Cantidad de créditos
DEUDAINICIAL(PRESTAMOMAESTRO)	Monto Total Entregado

Tabla 4.4 Indicadores- Cartera

Modelo Entidad Relación Proceso de Captaciones Vista



Figura 4.12. Modelo Entidad Relación Proceso de Captaciones Vista

Correspondencias de Captaciones Vista

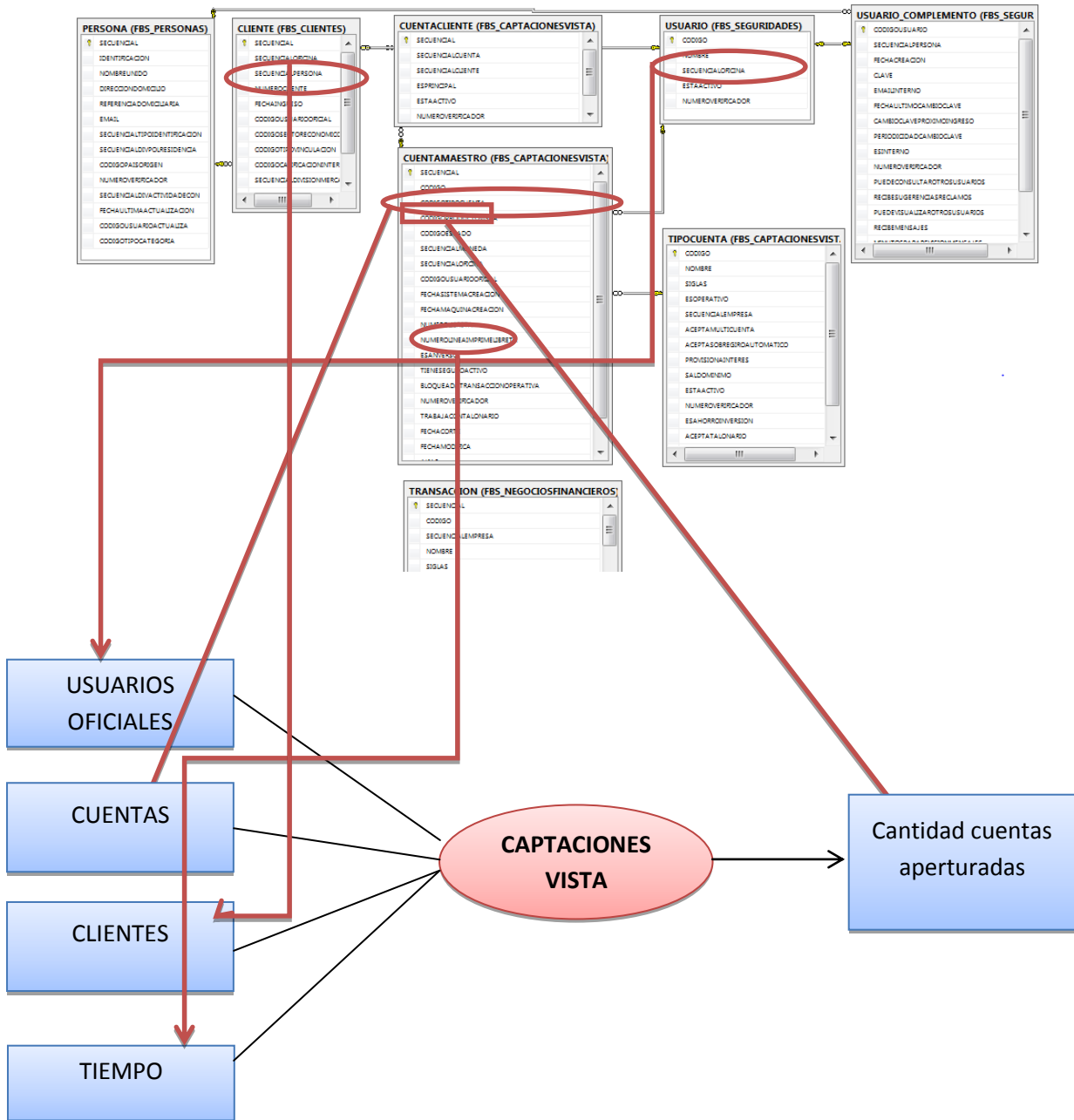


Figura 4.13. Correspondencias Captaciones Vista

Relaciones Identificadas

TABLA	PERSPECTIVA
CLIENTE	Clientes
CUENTAMAESTRO	Cuentas
USUARIO	Usuarios Oficiales
FECHACREACION	Tiempo

Tabla 4.5 Perspectivas– Captaciones Vista

CAMPO	INDICADOR
SECUENCIAL(CUENTAMAESTRO)	Cantidad

Tabla 4.6 Indicadores– Captaciones Vista

Correspondencias de Captaciones Plazo

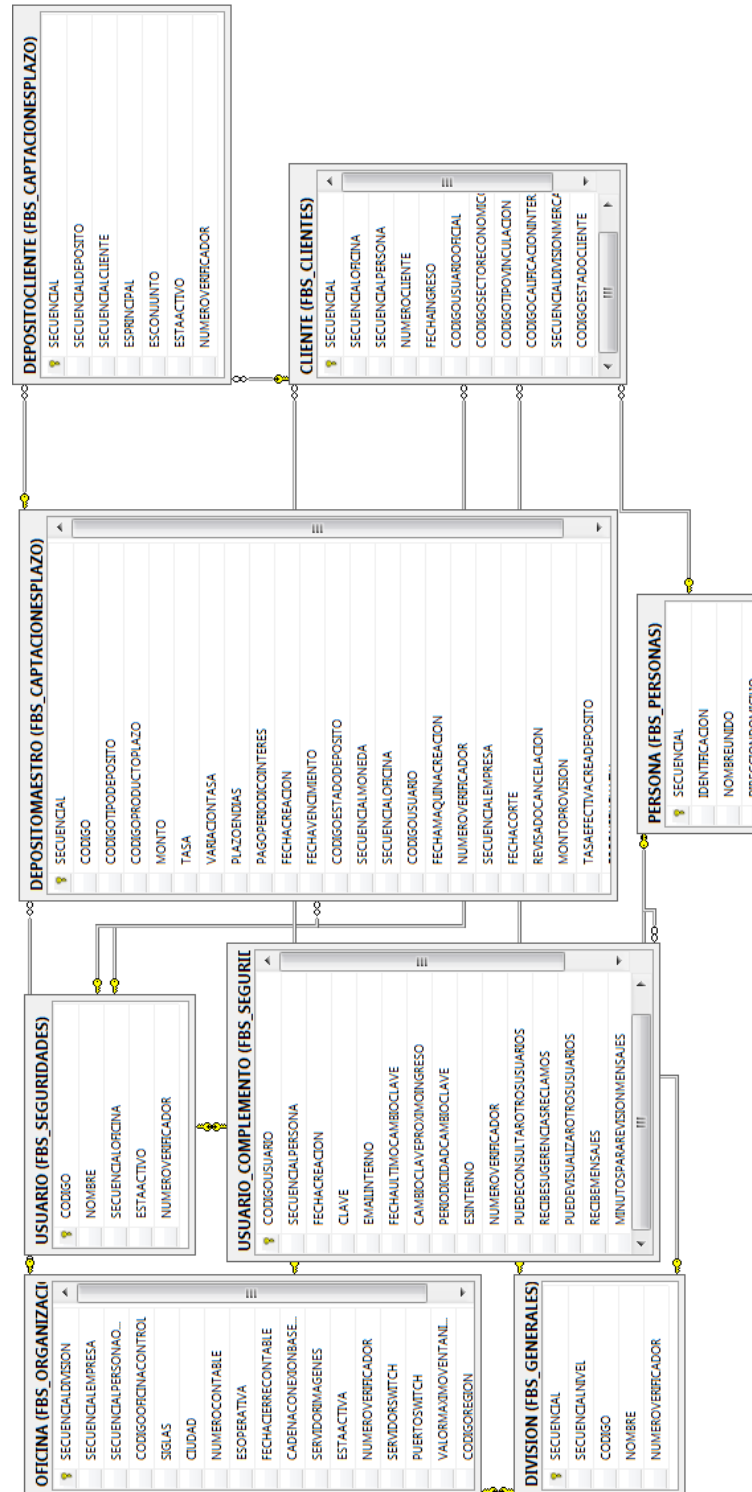


Figura 4.14. Modelo Entidad Relación Proceso de Captaciones Plazo

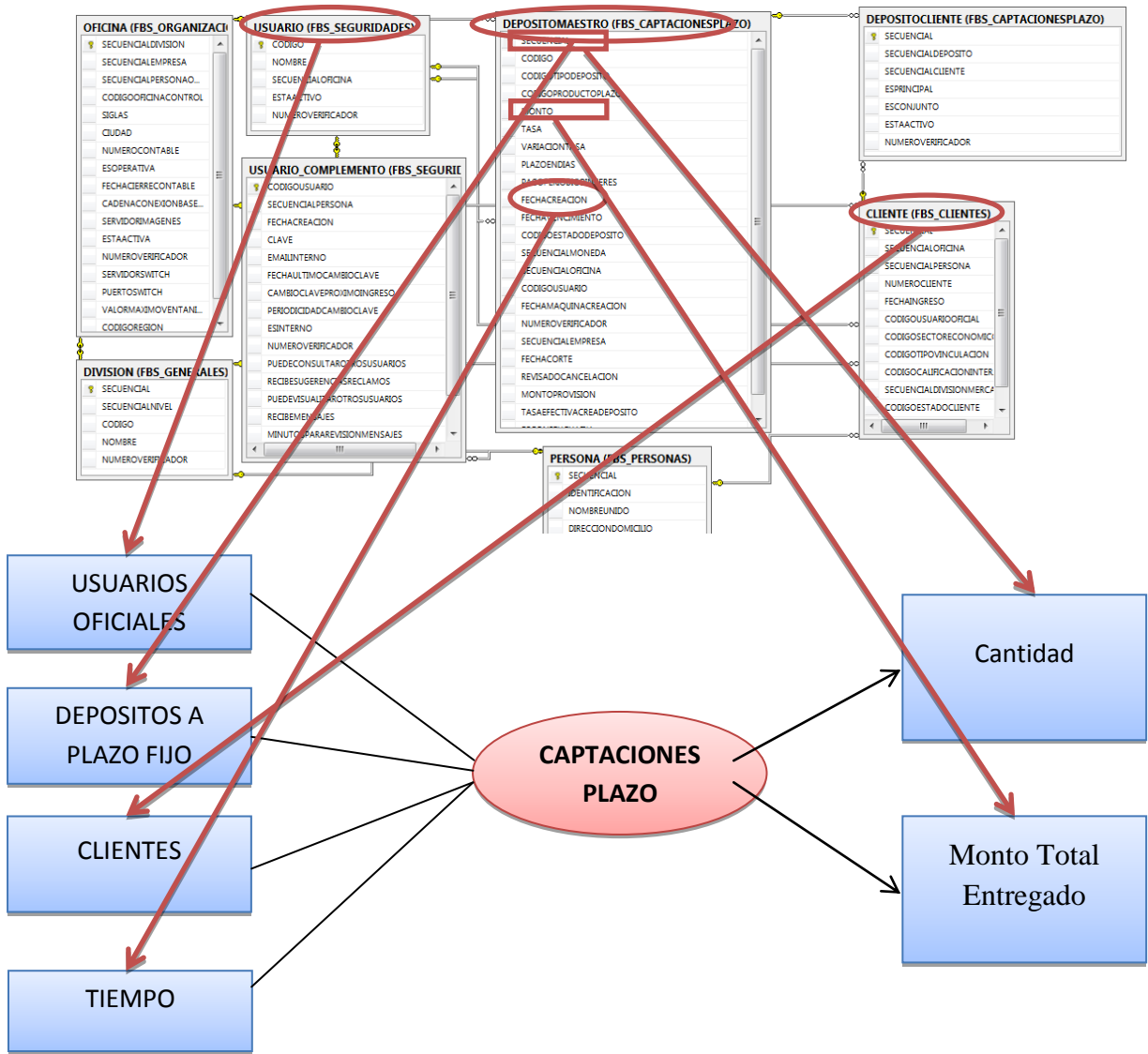


Figura 4.15. Correspondencias Captaciones Plazo

Relaciones Identificadas

TABLA	PERSPECTIVA
CLIENTE	Clientes
DEPOSITOMAESTRO	Depósitos a Plazo Fijo
USUARIO	Usuarios Oficiales
FECHACREACION	Tiempo

Tabla 4.7. Perspectivas– Captaciones Plazo

CAMPO	INDICADOR
SECUENCIAL(DEPOSITOMAESTRO)	Cantidad
MONTO(DEPOSITOMAESTRO)	Monto Total Entregado

Tabla 4.8. Indicadores– Captaciones Plazo

4.2.3. Determinar el nivel de granularidad.

A continuación se examinaron y se seleccionó los campos que contendrá cada perspectiva, ya que será a través de estos por los que se manipularán y filtrarán los indicadores.

Perspectivas Comunes

- Con respecto a la perspectiva Clientes, los datos disponibles son los siguientes:
 - **SECUENCIAL.-** Es un número entero secuencial que representa la clave primaria del cliente.
 - **SECUENCIALPERSONA.-** Clave foránea que indica la información de la persona a la que pertenece el cliente.
 - **NUMERO.-** Representa el código único que se asigna secuencialmente a un cliente.
 - **IDENTIFICACION.-** Indica la identificación que cada cliente ya sea cédula o ruc.
 - **NOMBRE.-** Representa el nombre del cliente.
 - **TIPO.-** Indica el tipo de cliente, persona natural, jurídica u organización.
 - **PROVINCIA.-** Indica la provincia de residencia del cliente.
 - **CANTON.-** Indica el cantón de residencia del cliente.
 - **PARROQUIA.-** Indica la parroquia de residencia del cliente.

- Con respecto a la perspectiva Usuarios Oficiales, los datos disponibles son los siguientes:
 - **CODIGO.-** Este campo será de tipo texto y contiene la clave primaria de los usuarios oficiales de crédito, captaciones vista e inversiones.
 - **NOMBRE.-** Contiene el nombre del usuario oficial.
 - **OFICINA.-** Representa la oficina a la que pertenece el usuario.

- Con respecto a la perspectiva "Tiempo", que es la que determinará la granularidad del depósito de datos, los datos que se emplearán serán los siguientes:
 - Año.

- Trimestre.
- Mes.
- Nombre del día.

Cartera

- Con respecto a la perspectiva Préstamos, los datos disponibles son los siguientes:
 - **SECUENCIAL.-** Este campo es de tipo entero y representa la clave primaria del préstamo.
 - **TIOPRESTAMO.-** Contiene el nombre asignado al tipo de préstamo.
 - **ESTADO.-** Indica el estado actual del préstamo.
 - **OFICINA.-** Nombre de la oficina donde se generó el crédito.
 - **CIUDAD.-** Nombre de la ciudad donde se entregó el crédito.
 - **CALIFICACIONCONTABLE.-** Indica la calificación actual en la que se encuentra el crédito.
 - **DESTINOFINANCIERO.-** Es el destino al que va dirigido el crédito, de acuerdo a su segmento interno.
 - **NUMEROPRESTAMO.-** Es un código único asignado a cada préstamo para su identificación y consultas en el sistema.
 - **DEUDAINICIAL.-** Representa el monto total entregado del préstamo.
 - **SALDOACTUAL.-** Indica el saldo pendiente de pago que tiene el préstamo.
 - **CODIGOUSUARIO.-** Representa el código del usuario que creó el préstamo en el sistema.

Captaciones Vista

- Con respecto a la perspectiva de Captaciones Vista, los datos disponibles son los siguientes:
 - **SECUENCIAL.-** Este campo será de tipo entero y contiene la clave primaria de las cuentas de captaciones vista.
 - **CODIGO.-** Este campo es de tipo texto y representa unívocamente a una cuenta.

- **TIPOCUENTA.-** Contiene el nombre asignado al tipo de cuenta.
- **OFICINA.-** Nombre de la oficina donde se creó la cuenta.
- **CIUDAD.-** Representa la ciudad de la oficina donde se creó la cuenta.
- **FECHACREACION.-** Indica la fecha de apertura de la cuenta.
- **ESTADO.-** Representa el estado actual de cada cuenta.
- **NUMEROLIBRETA.-** Indica el número de libreta que corresponde al documento que se entrega para imprimir los movimientos realizados en las cuentas.
- **CODIGOUSUARIO.-** Representa el código del usuario que creó el depósito la cuenta en el sistema.

Captaciones Plazo

- Con respecto a la perspectiva de Depósitos a Plazo Fijo, los datos disponibles son los siguientes:
 - **SECUENCIAL.-** Este campo será de tipo entero y contiene la clave primaria de las inversiones.
 - **CODIGO.-** Representa un código único del depósito a plazo fijo que se genera el momento que se crea el mismo.
 - **TIPODEPOSITO.-** Contiene el nombre asignado al tipo de inversión.
 - **OFICINA.-** Nombre de la oficina donde se generó el crédito.
 - **CIUDAD.-** Nombre de la ciudad donde se entregó el crédito.
 - **MONTO.-** Indica el monto entregado para el depósito a plazo fijo requerido.
 - **FECHACREACION.-** Fecha de creación del depósito a plazo fijo.
 - **TASAEFFECTIVACREADEPOSITO.-** Representa la tasa de interés con la que se creó el depósito a plazo fijo.
 - **CODIGOUSUARIO.-** Representa el código del usuario que creó el depósito a plazo fijo en el sistema.

Una vez que se recolectó toda la información pertinente y se consultó con el usuario cuales eran los datos que consideraban de interés para analizar los indicadores ya expuestos, los resultados obtenidos fueron los siguientes:

Perspectivas Comunes

- Con respecto a la perspectiva Clientes, los datos requeridos son los siguientes:
 - Nombre
 - Tipo
 - Provincia
 - Cantón
 - Parroquia
- Con respecto a la perspectiva Usuarios Oficiales, los datos requeridos son los siguientes:
 - Código
 - Nombre
- Con respecto a la perspectiva "Tiempo", los datos requeridos son los siguientes:
 - Año.
 - Trimestre.
 - Mes.
 - Nombre del día.

Cartera

- Con respecto a la perspectiva Préstamos, los datos requeridos son los siguientes:
 - Número
 - Tipo
 - Estado
 - Oficina
 - Ciudad
 - Calificación Contable

- Destino Financiero

Captaciones Vista

- Con respecto a la perspectiva de Captaciones Vista, los datos requeridos son los siguientes:
 - Tipo
 - Oficina
 - Ciudad

Captaciones Plazo

- Con respecto a la perspectiva de Depósitos a Plazo Fijo, los datos disponibles son los siguientes:
 - Tipo
 - Oficina
 - Ciudad

4.2.4. Diseñar el modelo conceptual ampliado.

En este paso, y con el fin de graficar los resultados obtenidos en los pasos anteriores, se ampliará el modelo conceptual colocando bajo cada perspectiva los campos elegidos y bajo cada indicador su respectiva fórmula de cálculo para cada módulo:

Cartera

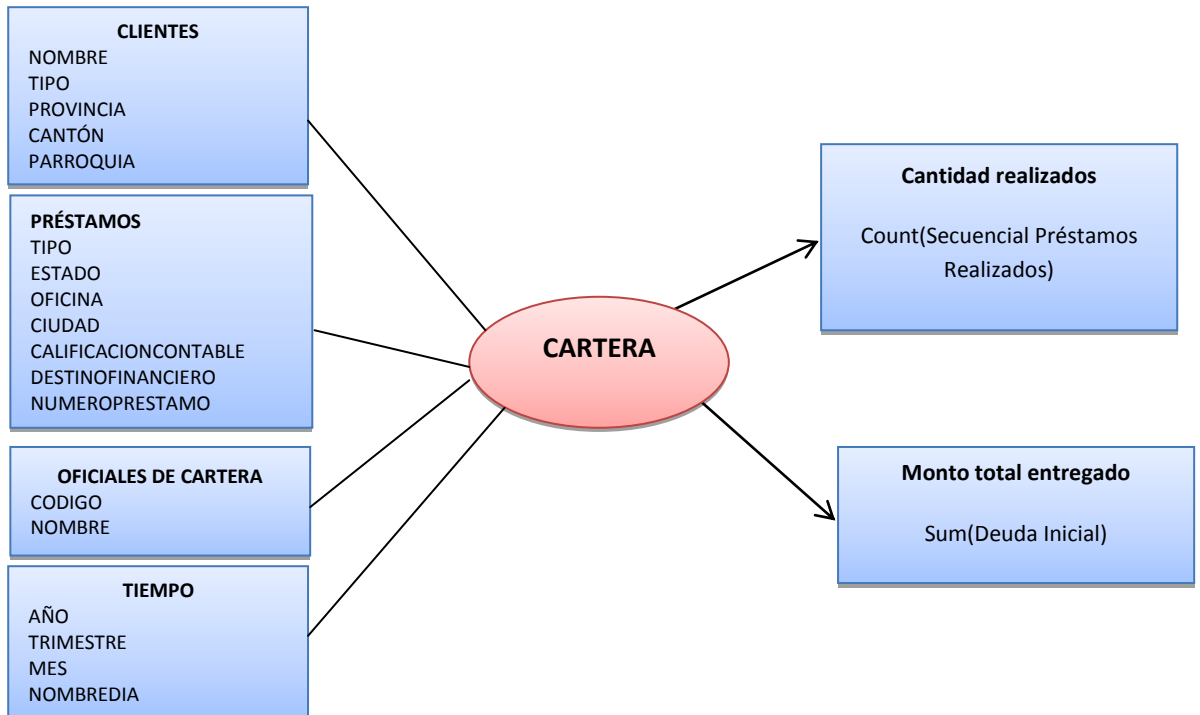


Figura 4.16. Modelo conceptual ampliado - Cartera

Captaciones Vista

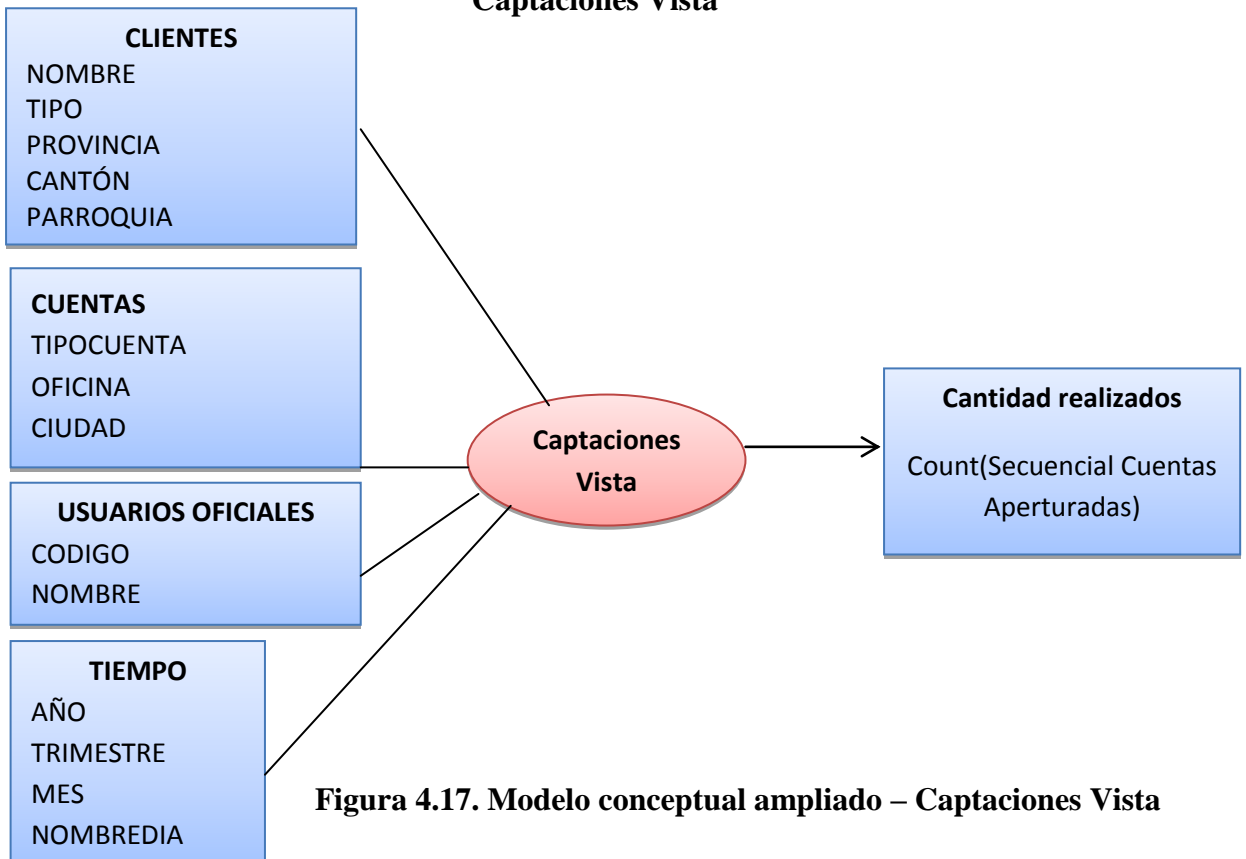


Figura 4.17. Modelo conceptual ampliado – Captaciones Vista

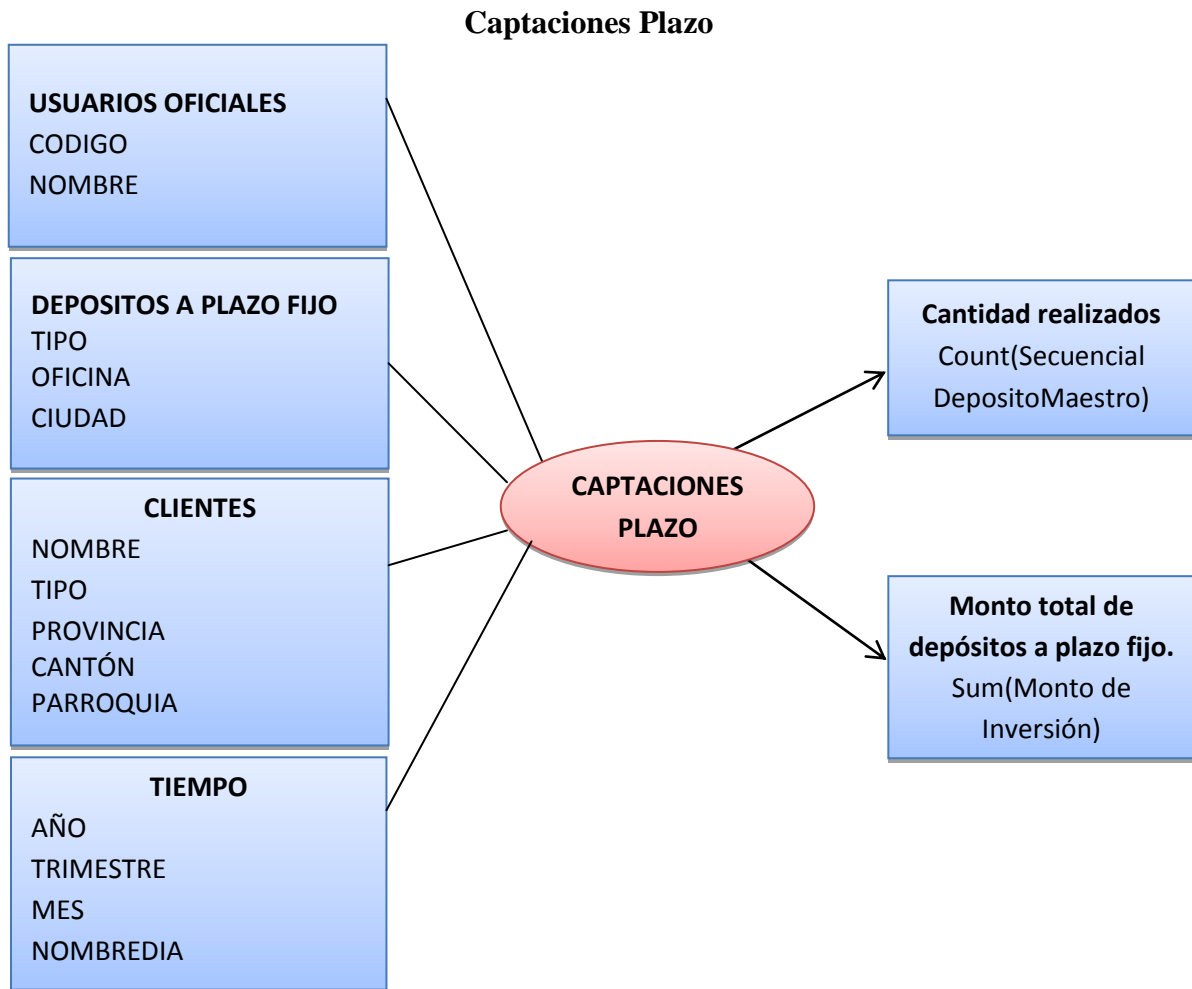


Figura 4.18. Modelo conceptual ampliado – Captaciones Plazo

4.3. Diseño del Modelo Lógico del Data Warehouse.

A continuación, se confeccionará el modelo lógico de la estructura del DW, teniendo como base el modelo conceptual que ya ha sido creado. Para ello, primero se definirá el tipo de modelo que se utilizará y luego se llevarán a cabo las acciones propias al caso, para diseñar las tablas de dimensiones y de hechos. Finalmente, se realizarán las uniones pertinentes entre estas tablas.

4.3.1. Establecer el tipo de modelo lógico a utilizar.

El esquema que se utilizará será en constelación, debido a sus características y diferencias con los otros esquemas siendo su principal ventaja la eliminación de redundancia de información y por ende un óptimo uso del espacio físico.

4.3.2. Crear tablas de dimensiones.

En este paso se diseñó las tablas de dimensiones que formaran parte del DW. Para ello se tomará cada perspectiva con sus campos relacionados y se realizará el siguiente proceso:

- Se elegirá un nombre que identifique la tabla de dimensión.
- Se añadirá un campo que represente su clave principal con nombre ID que al momento de referenciarlo, el nombre de la clave foránea identificará la dimensión a la que se hace referencia.
- Se redefinirán los nombres de los campos si es que no son lo suficientemente intuitivos.

Perspectivas Comunes

Perspectiva “Clientes”:

- La nueva tabla de dimensión tendrá el nombre “DIMCLIENTE”.
- Se le agregará una clave principal con el nombre “ID”.
- Un campo NOMBRE.
- Un campo PROVINCIA.
- Un campo CANTON.
- Un campo PARROQUIA.
- Además se agregará un campo TIPO.

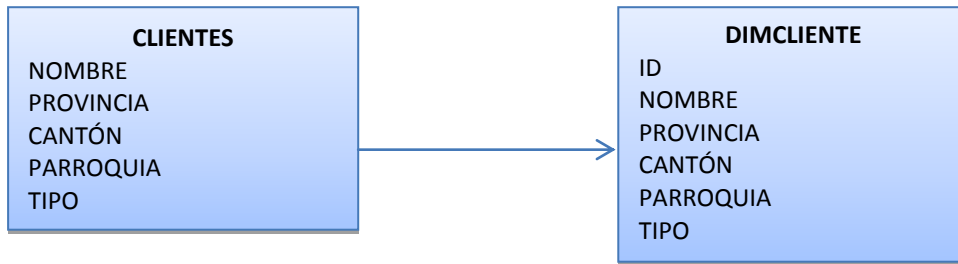


Figura 4.19. Perspectiva - Clientes

Perspectiva “Usuario Oficial”:

- La nueva tabla de dimensión tendrá el nombre “DIMOFICIAL”.
- Se le agregará una clave principal con el nombre “ID” que cambiará por el campo CODIGO.
- El nombre del campo NOMBRE no será cambiado.



Figura 4.20. Perspectiva – Usuario Oficial

Perspectiva “Tiempo”:

- La nueva tabla de dimensión tendrá el nombre “DIMFECHA”.
- Se le agregará una clave principal con el nombre “ID” que se formará con el año mes y el día.
- El nombre del campo AÑO será cambiado por ANIO.
- El resto de cambios no se cambiará de nombre.

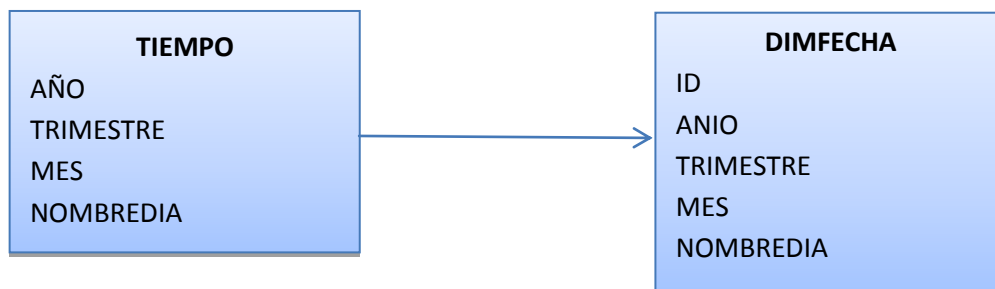


Figura 4.21. Perspectiva – Tiempo

Cartera

Perspectiva “Préstamos”:

- La nueva tabla de dimensión tendrá el nombre “DIMPRESTAMO”.
- Se le agregará una clave principal con el nombre “ID”.
- El nombre del campo TIPO no será cambiado.
- El nombre del campo ESTADO no será cambiado.
- El nombre del campo OFICINA no será cambiado.
- El nombre del campo CIUDAD no será cambiado.
- El nombre del campo CALIFICACIONCONTABLE no será cambiado.
- El nombre del campo DESTINOFINANCIERO no será cambiado.
- El nombre del campo NUMEROPRESTAMO no será cambiado.

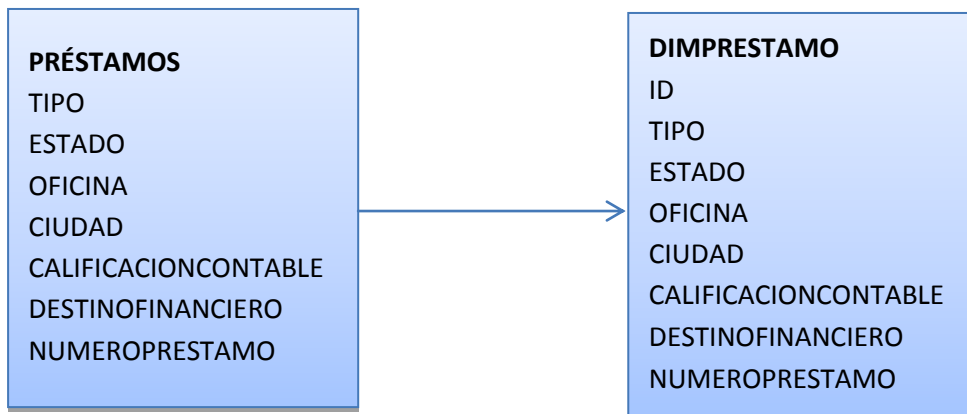


Figura 4.22. Perspectiva - Préstamos

Captaciones Vista

Perspectiva “Cuentas”:

- La nueva tabla de dimensión tendrá el nombre “DIMCUENTA”.
- Se le agregará una clave principal con el nombre “ID”.
- El nombre del campo CODIGO no será cambiado.
- El nombre del campo TIPOCUENTA no será cambiado.
- El nombre del campo OFICINA no será cambiado.
- El nombre del campo CIUDAD no será cambiado.
- El nombre del campo FECHACREACION no será cambiado.

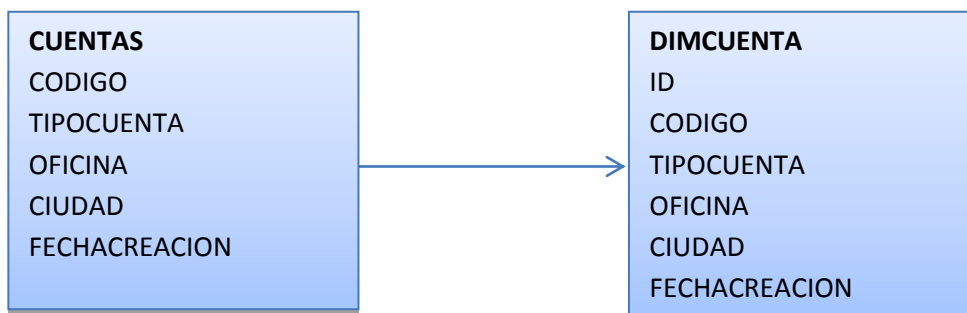


Figura 4.23. Perspectiva – Cuentas

Captaciones Plazo

Perspectiva “Depósitos a Plazo Fijo”:

- La nueva tabla de dimensión tendrá el nombre “DIMDEPOSITOMAESTRO”.
- Se le agregará una clave principal con el nombre “ID”.
- El nombre del campo TIPO no será cambiado.
- El nombre del campo OFICINA no será cambiado.
- El nombre del campo CIUDAD no será cambiado.



Figura 4.24. Perspectiva – Depósitos a Plazo Fijo

4.3.3. Construir las tablas de hechos.

En este paso, se definirán las tablas de hechos, que son las que contendrán los hechos a través de los cuales se construirán los indicadores de estudio.

Para el esquema en constelación, se realizará lo siguiente:

- Se le asignará un nombre a las tablas de hechos que represente la información analizada.

- Se definirá su clave primaria, que se compone de la combinación de las claves primarias de cada tabla de dimensión relacionada.
- Se crearán tantos campos de hechos como indicadores se hayan definido en el modelo conceptual y se les asignará los mismos nombres que estos. En caso que se prefiera, podrán ser nombrados de cualquier otro modo.

A continuación, se confeccionará las siguientes tablas de hechos por cada módulo:

Cartera

- La tabla de hechos tendrá el nombre de “HPRESTAMO”.
- La clave primaria contendrá las llaves primarias de las tablas de dimensiones antes definidas: ID (DIMCLIENTE), ID (DIMPRESTAMO), ID (DIMFECHA) e ID (DIMOFICIAL).
- Se creará dos hechos que corresponden a los indicadores antes presentados y que serán renombrados: Cantidad realizados por “CANTIDAD” y Monto total entregado por “MONTO”.

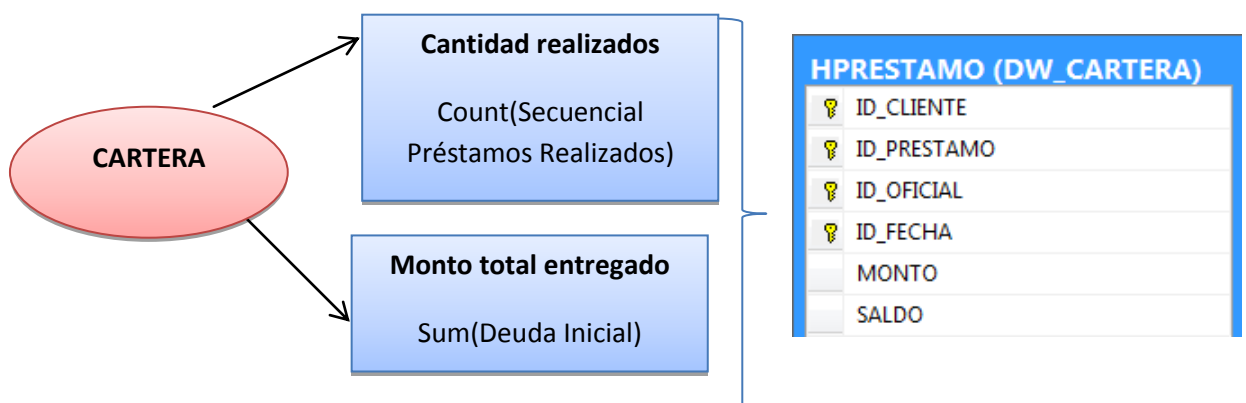


Figura 4.25. Tabla de hechos – cartera

Captaciones Vista

- La tabla de hechos tendrá el nombre de “HCUENTA”.

- La clave primaria contendrá las llaves primarias de las tablas de dimensiones antes definidas: ID (DIMCLIENTE), ID (DIMCUENTA), ID (DIMFECHA) e ID (DIMOFICIAL).
- Se creará dos hechos que corresponden a los indicadores antes presentados y que serán renombrados: Cantidad realizados por “CANTIDAD”.

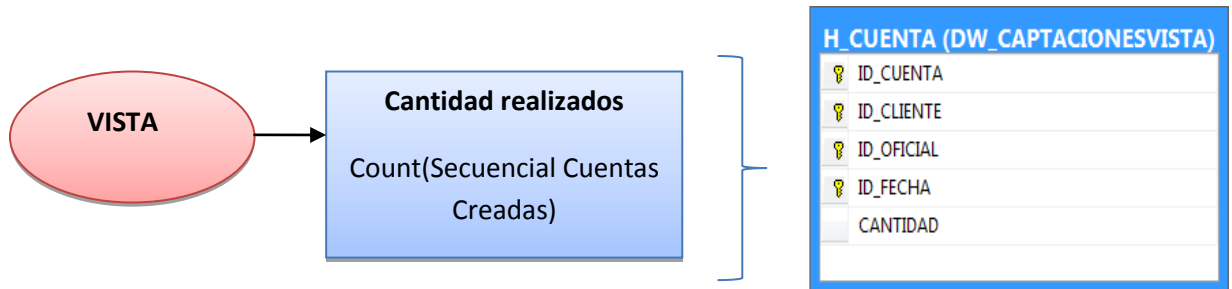


Figura 4.26. Tabla de hechos – Captaciones Vista

- La tabla de hechos tendrá el nombre de “HDEPOSITOMAESTRO”.
- La clave primaria contendrá las llaves primarias de las tablas de dimensiones antes definidas: ID (DIMCLIENTE), ID (DIMDEPOSITOMAESTRO), ID (DIMFECHA) e ID (DIMOFICIAL).
- Se creará dos hechos que corresponden a los indicadores antes presentados y que serán renombrados: Cantidad realizados por “CANTIDAD” y monto total de inversión “MONTO”.

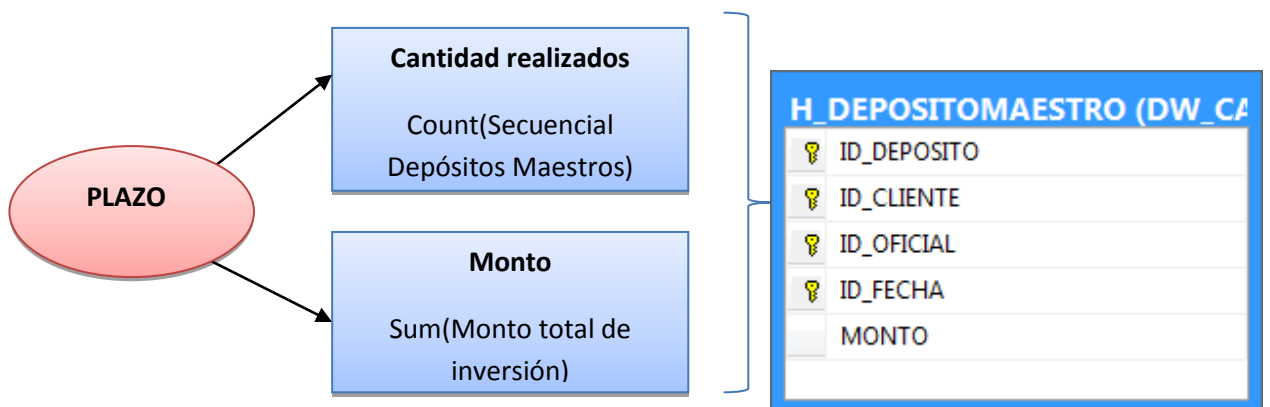


Figura 4.27. Tabla de hechos – Captaciones Plazo

4.3.4. Establecer las uniones correspondientes.

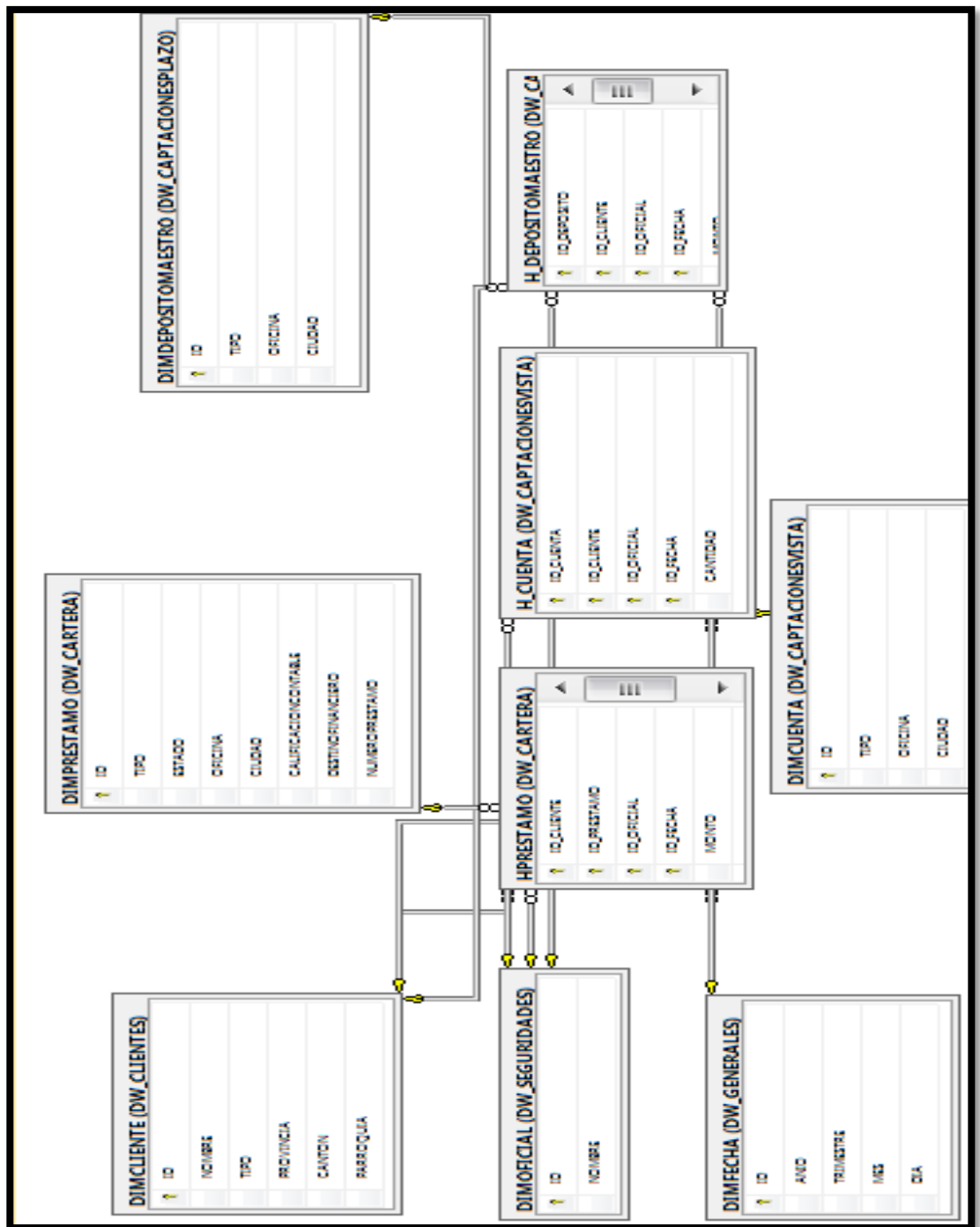


Figura 4.28. Uniones correspondientes

4.3.5. Creación de los cubos multidimensionales.

A continuación se desarrolló el Data Warehouse y se creó las dimensiones y la tabla de hechos que conformarán el mismo.

Dimensiones

- **DIMCLIENTE**

```
CREATE TABLE [DW_CLIENTES].[DIMCLIENTE] (
    [ID] [int] NOT NULL,
    [NOMBRE] [nvarchar](200) NOT NULL,
    [TIPO] [nvarchar](50) NOT NULL,
    [PROVINCIA] [nvarchar](50) NULL,
    [CANTON] [nvarchar](50) NULL,
    [PARROQUIA] [nvarchar](50) NULL,
    CONSTRAINT [PK_DIM_CLIENTE] PRIMARY KEY CLUSTERED
    (
        [ID] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
```

- **DIMOFICIAL**

```
CREATE TABLE [DW_SEGURIDADES].[DIMOFICIAL] (
    [ID] [nvarchar](20) NOT NULL,
    [NOMBRE] [nvarchar](200) NOT NULL,
    CONSTRAINT [PK_DIM_OFICIAL] PRIMARY KEY CLUSTERED
    (
        [ID] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
```

- **DIMFECHA**

```
CREATE TABLE [DW_GENERALES].[DIMFECHA] (
    [ID] [nvarchar](10) NOT NULL,
    [ANIO] [nvarchar](10) NOT NULL,
    [TRIMESTRE] [nvarchar](20) NOT NULL,
    [MES] [nvarchar](20) NOT NULL,
    [DIA] [nvarchar](20) NOT NULL,
    CONSTRAINT [PK_DIMFECHA] PRIMARY KEY CLUSTERED
    (
        [ID] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
```

- **DIMPRESTAMO**

```
CREATE TABLE [DW_CARTERA].[DIMPRESTAMO] (
    [ID] [int] NOT NULL,
```



```

        [TIPO] [nvarchar](50) NOT NULL,
        [ESTADO] [nvarchar](50) NOT NULL,
        [OFICINA] [nvarchar](200) NOT NULL,
        [CIUDAD] [nvarchar](30) NOT NULL,
        [CALIFICACIONCONTABLE] [nvarchar](50) NULL,
        [DESTINOFINANCIERO] [nvarchar](80) NULL,
        [NUMEROPRESTAMO] [nvarchar](20) NULL,
CONSTRAINT [PK_DIM_PRESTAMO] PRIMARY KEY CLUSTERED
(
        [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]GO

```

- **DIMCUENTA**

```

CREATE TABLE [DW_CAPTACIONESVISTA].[DIMCUENTA] (
        [ID] [int] NOT NULL,
        [TIPO] [nvarchar](200) NOT NULL,
        [OFICINA] [nvarchar](200) NOT NULL,
        [CIUDAD] [nvarchar](200) NOT NULL,
        CONSTRAINT [PK_DIMCUENTA] PRIMARY KEY CLUSTERED
(
        [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

- **DIMDEPOSITO**

```

CREATE TABLE [DW_CAPTACIONESPLAZO].[DIMDEPOSITOMAESTRO] (
        [ID] [int] NOT NULL,
        [TIPO] [nvarchar](50) NOT NULL,
        [OFICINA] [nvarchar](200) NOT NULL,
        [CIUDAD] [nvarchar](50) NOT NULL,
        CONSTRAINT [PK_DIM_DEPOSITOMAESTRO] PRIMARY KEY CLUSTERED
(
        [ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

Tablas de hechos

- **HPRESTAMO**

```

CREATE TABLE [DW_CARTERA].[HPRESTAMO] (
        [ID_CLIENTE] [int] NOT NULL,
        [ID_PRESTAMO] [int] NOT NULL,
        [ID_OFICIAL] [nvarchar](20) NOT NULL,
        [ID_FECHA] [nvarchar](10) NOT NULL,
        [MONTO] [decimal](18, 2) NOT NULL,
        [SALDO] [decimal](18, 2) NULL,
CONSTRAINT [PK_HPRESTAMO] PRIMARY KEY CLUSTERED
(
        [ID_CLIENTE] ASC,
        [ID_PRESTAMO] ASC,
        [ID_OFICIAL] ASC,
        [ID_FECHA] ASC
)

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [DW_CARTERA].[HPRESTAMO] WITH CHECK ADD CONSTRAINT
[FK_HPRESTAMO_DIMCLIENTE] FOREIGN KEY([ID_CLIENTE])
REFERENCES [DW_CLIENTES].[DIMCLIENTE] ([ID])
GO

ALTER TABLE [DW_CARTERA].[HPRESTAMO] CHECK CONSTRAINT [FK_HPRESTAMO_DIMCLIENTE]
GO

ALTER TABLE [DW_CARTERA].[HPRESTAMO] WITH CHECK ADD CONSTRAINT
[FK_HPRESTAMO_DIMFECHA] FOREIGN KEY([ID_FECHA])
REFERENCES [DW_GENERALES].[DIMFECHA] ([ID])
GO

ALTER TABLE [DW_CARTERA].[HPRESTAMO] CHECK CONSTRAINT [FK_HPRESTAMO_DIMFECHA]
GO

ALTER TABLE [DW_CARTERA].[HPRESTAMO] WITH CHECK ADD CONSTRAINT
[FK_HPRESTAMO_DIMOFICIAL] FOREIGN KEY([ID_OFICIAL])
REFERENCES [DW_SEGURIDADES].[DIMOFICIAL] ([ID])
GO

ALTER TABLE [DW_CARTERA].[HPRESTAMO] CHECK CONSTRAINT [FK_HPRESTAMO_DIMOFICIAL]
GO

ALTER TABLE [DW_CARTERA].[HPRESTAMO] WITH CHECK ADD CONSTRAINT
[FK_HPRESTAMO_DIMPRESTAMO] FOREIGN KEY([ID_PRESTAMO])
REFERENCES [DW_CARTERA].[DIMPRESTAMO] ([ID])
GO

ALTER TABLE [DW_CARTERA].[HPRESTAMO] CHECK CONSTRAINT [FK_HPRESTAMO_DIMPRESTAMO]
GO

```

- **HCUENTA**

```

CREATE TABLE [DW_CAPTACIONESVISTA].[HCUENTA] (
    [ID_CUENTA] [int] NOT NULL,
    [ID_CLIENTE] [int] NOT NULL,
    [ID_OFICIAL] [nvarchar](20) NOT NULL,
    [ID_FECHA] [nvarchar](10) NOT NULL,
    [CANTIDAD] [int] NULL,
    CONSTRAINT [PK_H_CUENTA] PRIMARY KEY CLUSTERED
(
    [ID_CUENTA] ASC,
    [ID_CLIENTE] ASC,
    [ID_OFICIAL] ASC,
    [ID_FECHA] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [DW_CAPTACIONESVISTA].[HCUENTA] WITH CHECK ADD CONSTRAINT
[FK_H_CUENTA_DIMCLIENTE] FOREIGN KEY([ID_CLIENTE])
REFERENCES [DW_CLIENTES].[DIMCLIENTE] ([ID])
GO

ALTER TABLE [DW_CAPTACIONESVISTA].[HCUENTA] CHECK CONSTRAINT
[FK_H_CUENTA_DIMCLIENTE]
GO

ALTER TABLE [DW_CAPTACIONESVISTA].[HCUENTA] WITH CHECK ADD CONSTRAINT
[FK_H_CUENTA_DIMCUENTA] FOREIGN KEY([ID_CUENTA])
REFERENCES [DW_CAPTACIONESVISTA].[DIMCUENTA] ([ID])
GO

```

```

ALTER TABLE [DW_CAPTACIONESVISTA].[HCUENTA] CHECK CONSTRAINT [FK_H_CUENTA_DIMCUENTA]
GO

ALTER TABLE [DW_CAPTACIONESVISTA].[HCUENTA] WITH CHECK ADD CONSTRAINT
[FK_H_CUENTA_DIMFECHA] FOREIGN KEY([ID_FECHA])
REFERENCES [DW_GENERALES].[DIMFECHA] ([ID])
GO

ALTER TABLE [DW_CAPTACIONESVISTA].[HCUENTA] CHECK CONSTRAINT [FK_H_CUENTA_DIMFECHA]
GO

ALTER TABLE [DW_CAPTACIONESVISTA].[HCUENTA] WITH CHECK ADD CONSTRAINT
[FK_H_CUENTA_DIMOFICIAL] FOREIGN KEY([ID_OFICIAL])
REFERENCES [DW_SEGURIDADES].[DIMOFICIAL] ([ID])
GO

ALTER TABLE [DW_CAPTACIONESVISTA].[HCUENTA] CHECK CONSTRAINT
[FK_H_CUENTA_DIMOFICIAL]
GO

```

- **HDEPOSITOMAESTRO**

```

CREATE TABLE [DW_CAPTACIONESPLAZO].[HDEPOSITOMAESTRO] (
    [ID_DEPOSITO] [int] NOT NULL,
    [ID_CLIENTE] [int] NOT NULL,
    [ID_OFICIAL] [nvarchar] (20) NOT NULL,
    [ID_FECHA] [nvarchar] (10) NOT NULL,
    [MONTO] [decimal] (18, 2) NOT NULL,
    CONSTRAINT [PK_H_DEPOSITOMAESTRO] PRIMARY KEY CLUSTERED
(
    [ID_DEPOSITO] ASC,
    [ID_CLIENTE] ASC,
    [ID_OFICIAL] ASC,
    [ID_FECHA] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

ALTER TABLE [DW_CAPTACIONESPLAZO].[HDEPOSITOMAESTRO] WITH CHECK ADD CONSTRAINT
[FK_H_DEPOSITOMAESTRO_DIMCLIENTE] FOREIGN KEY([ID_CLIENTE])
REFERENCES [DW_CLIENTES].[DIMCLIENTE] ([ID])
GO

ALTER TABLE [DW_CAPTACIONESPLAZO].[HDEPOSITOMAESTRO] CHECK CONSTRAINT
[FK_H_DEPOSITOMAESTRO_DIMCLIENTE]
GO

ALTER TABLE [DW_CAPTACIONESPLAZO].[HDEPOSITOMAESTRO] WITH CHECK ADD CONSTRAINT
[FK_H_DEPOSITOMAESTRO_DIMDEPOSITOMAESTRO] FOREIGN KEY([ID_DEPOSITO])
REFERENCES [DW_CAPTACIONESPLAZO].[DIMDEPOSITOMAESTRO] ([ID])
GO

ALTER TABLE [DW_CAPTACIONESPLAZO].[HDEPOSITOMAESTRO] CHECK CONSTRAINT
[FK_H_DEPOSITOMAESTRO_DIMDEPOSITOMAESTRO]
GO

ALTER TABLE [DW_CAPTACIONESPLAZO].[HDEPOSITOMAESTRO] WITH CHECK ADD CONSTRAINT
[FK_H_DEPOSITOMAESTRO_DIMFECHA] FOREIGN KEY([ID_FECHA])
REFERENCES [DW_GENERALES].[DIMFECHA] ([ID])
GO

ALTER TABLE [DW_CAPTACIONESPLAZO].[HDEPOSITOMAESTRO] CHECK CONSTRAINT
[FK_H_DEPOSITOMAESTRO_DIMFECHA]
GO

ALTER TABLE [DW_CAPTACIONESPLAZO].[HDEPOSITOMAESTRO] WITH CHECK ADD CONSTRAINT
[FK_H_DEPOSITOMAESTRO_DIMOFICIAL] FOREIGN KEY([ID_OFICIAL])

```

```

REFERENCES [DW_SEGURIDADES].[DIMOFICIAL] ([ID])
GO

ALTER TABLE [DW_CAPTACIONESPLAZO].[HDEPOSITOMAESTRO] CHECK CONSTRAINT
[FK_H_DEPOSITOMAESTRO_DIMOFICIAL]
GO

```

Creación Cubos en Analysis Services

A continuación creamos los cubos multidimensionales por cada esquema con la ayuda de la herramienta Analysis Services de SQLSERVER 2012 siguiendo los siguientes pasos:

1.- **Crear un nuevo Proyecto.-** Se creó un nuevo Proyecto Multidimensional y de Minería de Datos que se llamará Cubo BI.

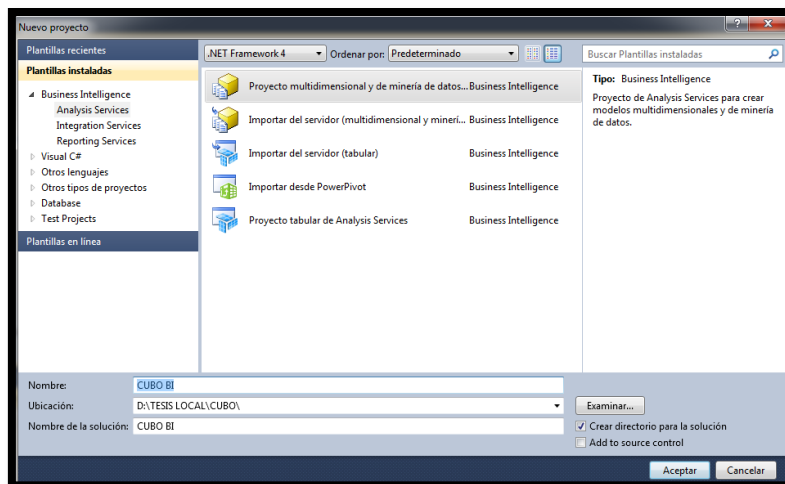


Figura 4.29. Crear nuevo proyecto

Se generó la siguiente estructura sobre la que vamos a trabajar.

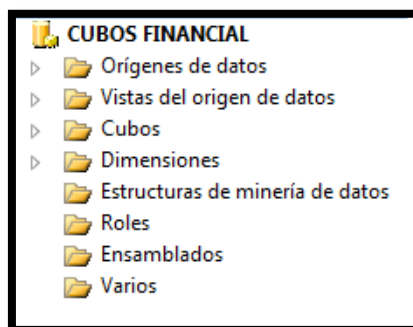


Figura 4.30. Estructura de proyecto

2.- Creación Origen de Datos.- Se creó la conexión con la base de datos en la que se encuentra nuestras tablas de dimensión y de hechos.

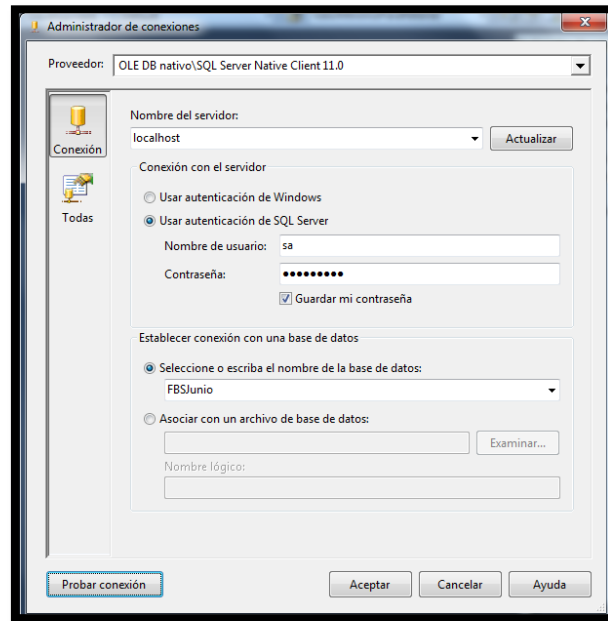


Figura 4.31. Conexiones

A continuación se muestra la cadena de conexión resultante.

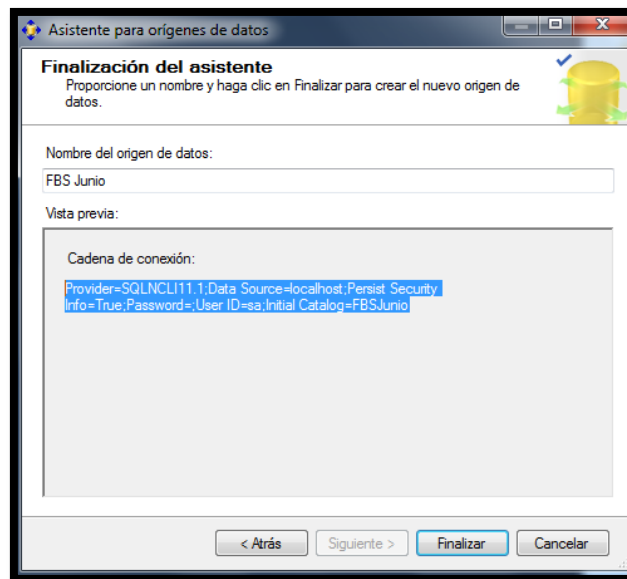


Figura 4.32. Cadena de conexión

3.- Creación Vistas del Origen de Datos.- Se creó una vista del origen de datos que servirá para escoger las tablas necesarias para crear el cubo, siguiendo los siguientes pasos.

- Se selecciona el origen de datos creado anteriormente.

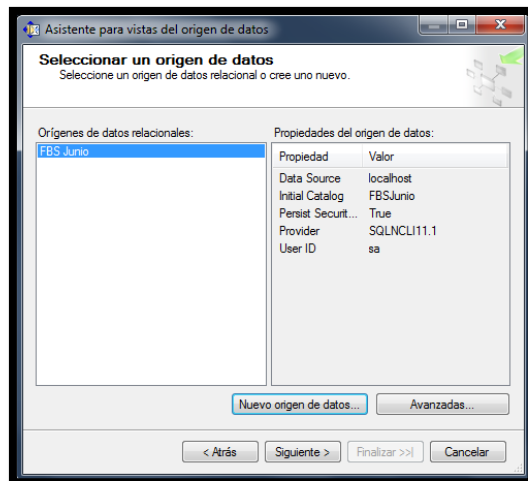


Figura 4.33. Creación Vistas del Origen de Datos

- Escoger las tablas de dimensiones y de hechos que se utilizarán para generar el cubo.

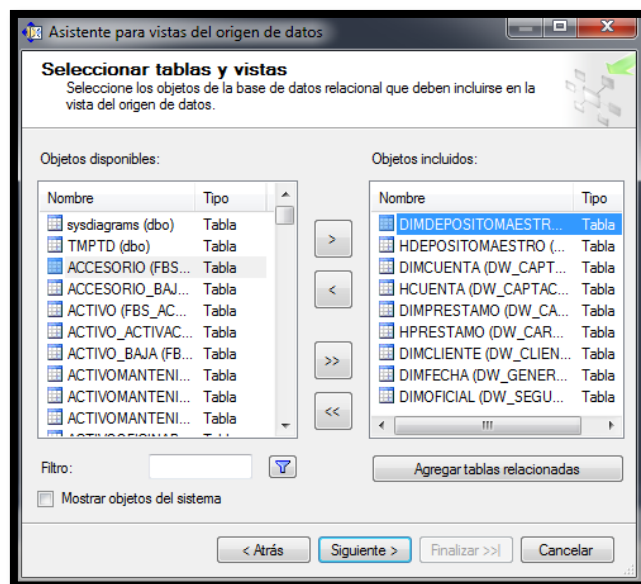


Figura 4.34. Selección de tablas y dimensiones

Al final nos muestra un listado con las dimensiones y las tablas de hechos con los que vamos podemos formar los cubos.

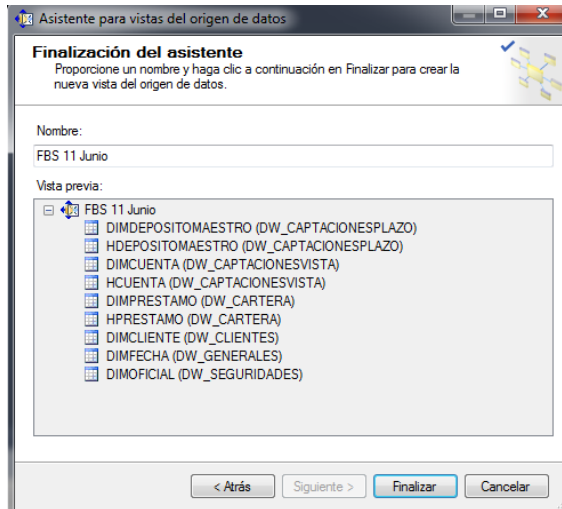


Figura 4.35. Finalización del asistente

A continuación se visualiza el diagrama de la vista de datos creados en este paso.

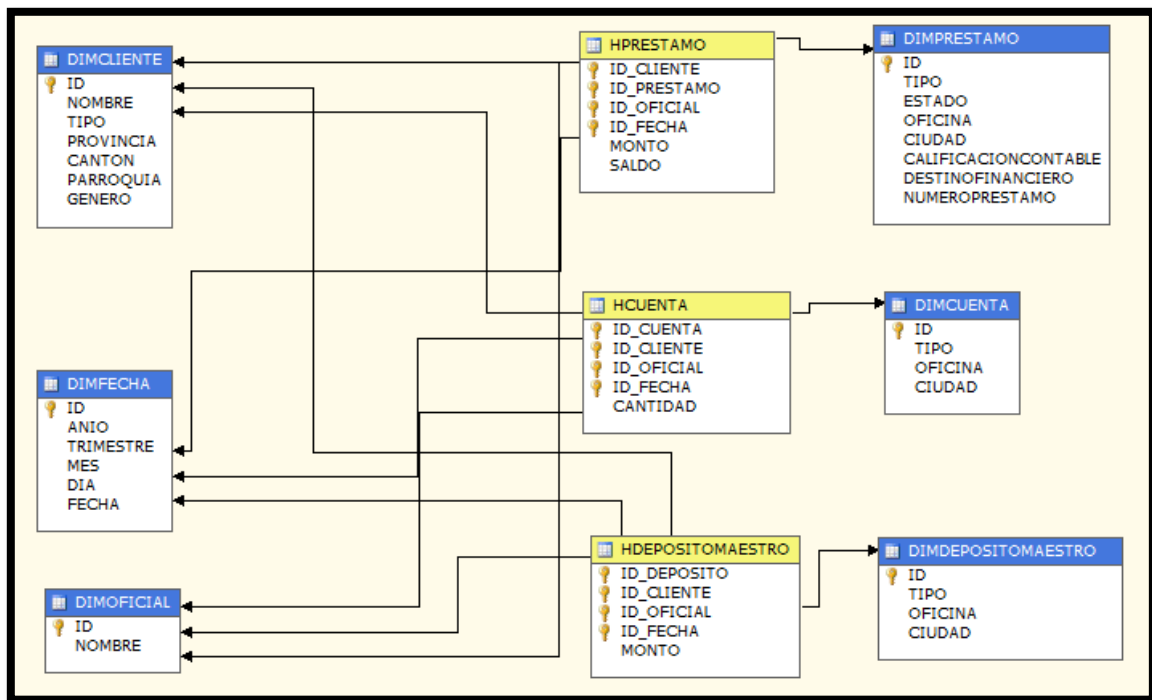


Figura 4.36. Diagrama de la vista de datos

4.- Creación de Cubos.- Se creó un cubo por cada esquema en base a las tablas creadas en la vista de datos generada anteriormente:

Cartera

- Seleccionar las tablas de hechos que contienen los indicadores de cada módulo.

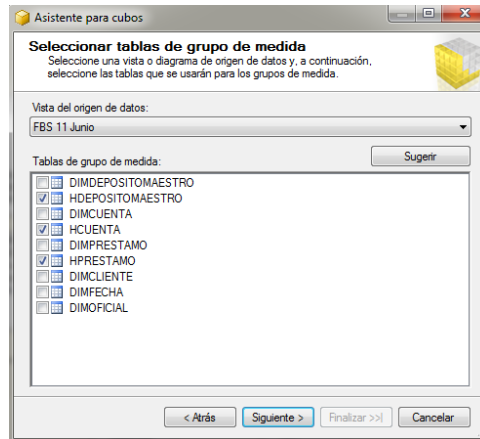


Figura 4.37. Selección tablas de hechos

- Seleccionar las medidas en base a las cuales se va a consultar con las distintas dimensiones.

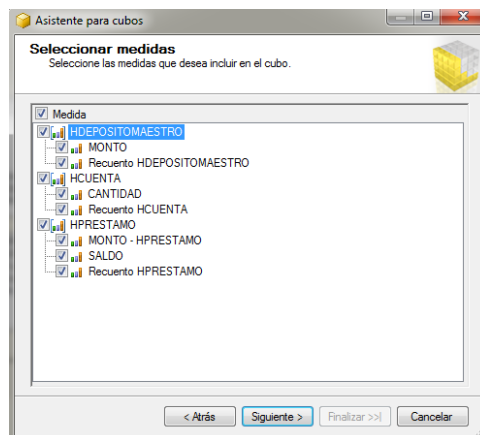


Figura 4.38. Selección medidas

- Seleccionar las dimensiones pre-establecidas en el Data Warehouse para crear los cubos por cada esquema.

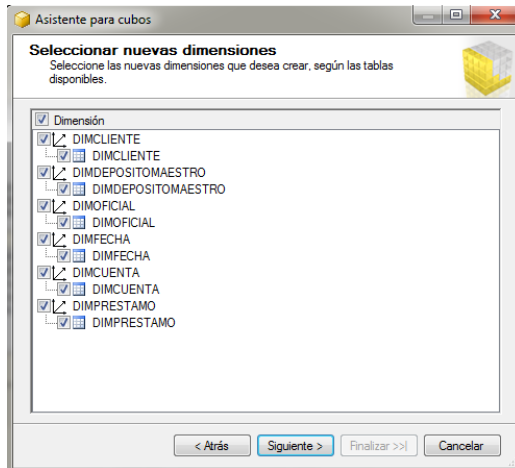


Figura 4.39. Selección dimensiones

- Al final se obtendrá las medidas y las dimensiones diferenciadas por carpetas así:

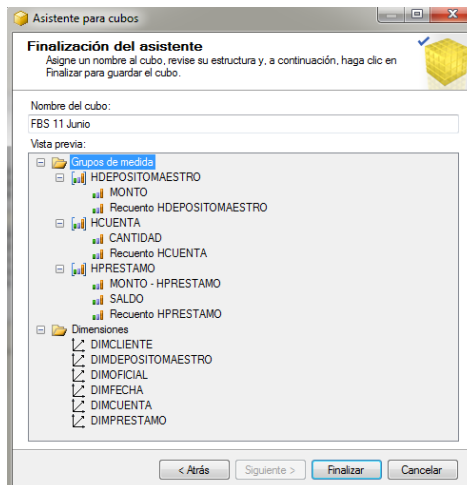


Figura 4.40. Resumen de creación de los cubos

- Seguido de este paso se creará las jerarquías para las fechas y la división política de la residencia del cliente así:

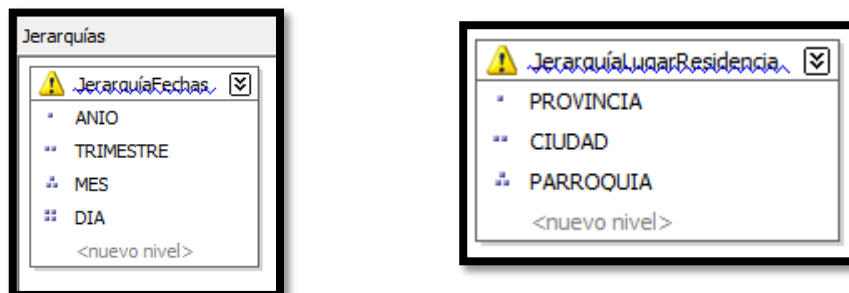


Figura 4.41. Jerarquías

- Se creará un KPI en el cubo de cartera que presente alertas en base al abono del préstamo indicando un valor neutro en un cincuenta por ciento de abono del préstamo a partir de ahí los valores mayores indicarán alertas negativas caso contrario alertas negativas.

Figura 4.42. Generación KPI

- A continuación será necesario implementar toda la solución y luego procesar cada uno de los cubos creados.

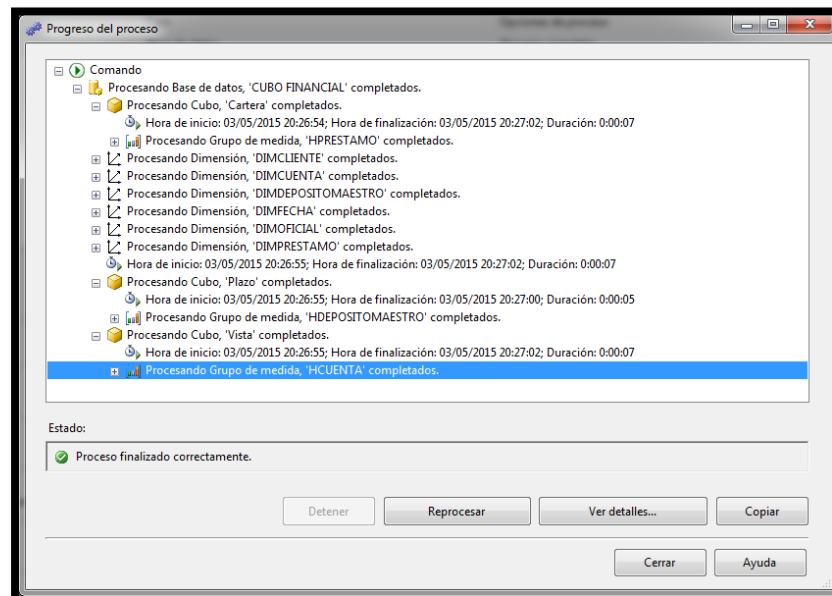


Figura 4.43. Proceso Cubos Financial

4.4. Desarrollo de los procesos de extracción, transformación y carga de datos

4.4.1. Diseñar proceso de carga inicial.

Para el proceso inicial de carga del Data Warehouse se realizaron procedimientos almacenados cuya funcionalidad consiste en ejecutar una consulta y de acuerdo a su respectivo proceso seleccionar los datos e insertarlos en las correspondientes tablas de dimensiones.

Así se definió los procedimientos separados por módulo:

- **Cartera**

```
CREATE PROCEDURE [dbo].[CREADWCARTERA]
AS
BEGIN

    SET NOCOUNT ON;

    DECLARE @SECUENCIALPRESTAMO INT;
    DECLARE @NUMEROPRESTAMO NVARCHAR(20);
    DECLARE @FECHAADJUDICACION DATE;
    DECLARE @CALIFICACIONCONTABLE NVARCHAR(50);
    DECLARE @DESTINOFINANCIERO NVARCHAR(80);
    DECLARE @OFICINAPRESTAMO NVARCHAR(200);
    DECLARE @OFICIALCREDITO NVARCHAR(200);
    DECLARE @CIUDADOFICINA NVARCHAR(30);
    DECLARE @SECUENCIALCLIENTE INT;
    DECLARE @CLIENTE NVARCHAR(200);
    DECLARE @IDCLIENTE NVARCHAR(20);
    DECLARE @TIPOCLIENTE NVARCHAR(50);
    DECLARE @ESTADOPRESTAMO NVARCHAR(50);
    DECLARE @TIPOPRESTAMO NVARCHAR(50);
    DECLARE @CODIGOOFICIAL NVARCHAR(20);
    DECLARE @DEUDAINICIAL DECIMAL(18,2);
    DECLARE @SECUENCIALDIVPOLRESIDENCIA INT;

    DECLARE @IDFECHA NVARCHAR(10);
    DECLARE @TRIMESTRE NVARCHAR(20);
    DECLARE @MES NVARCHAR(20);
    DECLARE @DIA NVARCHAR(20);
    DECLARE @SALDO DECIMAL(18,2);

    DECLARE DWCR01 CURSOR SCROLL FOR
    SELECT PM.SECUENCIAL SECUENCIALPRESTAMO,
           PM.NUMEROPRESTAMO,
           PM.FECHAADJUDICACION,
           CC.NOMBRE CALIFICACIONCONTABLE,
           DF.NOMBRE DESTINOFINANCIERO,
           D.NOMBRE OFICINAPRESTAMO,
           PO.NOMBREUNIDO OFICIALCREDITO,
           O.CIUDAD CIUDADOFICINA,
           C.SECUENCIAL SECUENCIALCLIENTE,
           PEC.NOMBREUNIDO CLIENTE,
           PEC.IDENTIFICACION IDCLIENTE,
           CASE WHEN
             TI.PARAPERSONANATURAL=1
```

```

        THEN
        'PERSONAS NATURALES'
        ELSE
        'PERSONAS JURIDICAS'
        END TIPOCLIENTE,
        EP.NOMBRE ESTADOPRESTAMO,
        TP.NOMBRE TIPOPRESTAMO
        U.CODIGO CODIGOOFICIAL ,
        PM.DEUDAINICIAL,
        PO.SECUENCIALDIVPOLRESIDENCIA,
        PM.SALDOACTUAL SALDO
FROM [REDACTED] PM
        INNER JOIN [REDACTED] SI ON
SI.SECUENCIAL=PM.SECUENCIALSEGMENTOINTERNO
        INNER JOIN [REDACTED] CC ON
CC.CODIGO=SI.CODIGOCALIFICACIONCONTABLE
        INNER JOIN [REDACTED] DF ON
DF.CODIGO=SI.CODIGODESTINOFINANCIERO
        INNER JOIN [REDACTED] O ON
O.SECUENCIALDIVISION=PM.SECUENCIALOFICINA
        INNER JOIN [REDACTED] D ON D.SECUENCIAL=O.SECUENCIALDIVISION
        INNER JOIN [REDACTED] U ON U.CODIGO =
PM.CODIGOUSUARIOOFICIAL
        INNER JOIN [REDACTED] UC ON
UC.CODIGOUSUARIO=U.CODIGO
        INNER JOIN [REDACTED] PO ON PO.SECUENCIAL=UC.SECUENCIALPERSONA
        INNER JOIN [REDACTED] PC ON
PC.SECUENCIALPRESTAMO=PM.SECUENCIAL
        INNER JOIN [REDACTED] C ON C.SECUENCIAL=PC.SECUENCIALCLIENTE
        INNER JOIN [REDACTED] PEC ON PEC.SECUENCIAL=C.SECUENCIALPERSONA
        INNER JOIN [REDACTED] TI ON
TI.SECUENCIAL=PEC.SECUENCIALTIPOIDENTIFICACION
        INNER JOIN [REDACTED] EP ON EP.CODIGO =
PM.CODIGOESTADOPRESTAMO
        INNER JOIN [REDACTED] TP ON TP.CODIGO=PM.CODIGOTIPOPRESTAMO

OPEN DWCR01
FETCH FIRST FROM DWCR01
INTO @SECUENCIALPRESTAMO,
@NUMEROPRESTAMO,
@FECHAADJUDICACION,
@CALIFICACIONCONTABLE,
@DESTINOFINANCIERO,
@OFICINAPRESTAMO,
@OFICIALCREDITO,
@CIUDADOFICINA,
@SECUENCIALCLIENTE,
@CLIENTE,
@IDCLIENTE,
@TIPOCLIENTE,
@ESTADOPRESTAMO,
@TIPOPRESTAMO,
@CODIGOOFICIAL,
@DEUDAINICIAL,
@SECUENCIALDIVPOLRESIDENCIA,
@SALDO

WHILE (@@FETCH_STATUS = 0)
BEGIN
--Dimensión Préstamo
IF (ISNULL((SELECT DM.ID
FROM DW_CARTERA.DIMPRESTAMO DM
WHERE DM.ID = @SECUENCIALPRESTAMO), 0)=0)
        BEGIN
                INSERT INTO DW_CARTERA.DIMPRESTAMO
                VALUES (@SECUENCIALPRESTAMO, @TIPOPRESTAMO, @ESTADOPRESTAMO, @OFICINAPRESTAMO, @CIUDA
                DOFICINA,
                @CALIFICACIONCONTABLE, @DESTINOFINANCIERO, @NUMEROPRESTAMO);
        END
END

```

```

--Dimensión Oficial
--Inserta un Nuevo Oficial
IF(ISNULL((SELECT TOP(1) DMO.ID
            FROM DW_SEGURIDADES.DIMOFICIAL DMO
            WHERE DMO.ID = @CODIGOOFICIAL), 'NE')='NE')
BEGIN
    INSERT INTO DW_SEGURIDADES.DIMOFICIAL
    VALUES (@CODIGOOFICIAL, @OFICIALCREDITO);
END

--Actualiza oficial
IF(ISNULL((SELECT TOP(1) DM.ID
            FROM DW_SEGURIDADES.DIMOFICIAL DM
            WHERE DM.ID = @CODIGOOFICIAL), 'NE')<>'NE')
BEGIN
    IF(ISNULL((SELECT TOP(1) DMO.ID
                FROM DW_SEGURIDADES.DIMOFICIAL DMO
                WHERE DMO.NOMBRE = @OFICIALCREDITO
                AND DMO.ID= @CODIGOOFICIAL), 'NE')='NE')
    BEGIN
        UPDATE DW CARTERA.DIMOFICIAL
        SET NOMBRE = @OFICIALCREDITO
        WHERE ID=@CODIGOOFICIAL
    END
END

--Dimensión Cliente
--Inserta un Nuevo Cliente
IF(ISNULL((SELECT TOP(1) DM.ID
            FROM DW_CLIENTES.DIMCLIENTE DM
            WHERE DM.ID = @SECUENCIALCLIENTE), 0)=0)
BEGIN
    INSERT INTO DW_CLIENTES.DIMCLIENTE
    VALUES (@SECUENCIALCLIENTE, @CLIENTE, @TIPOCLIENTE, FBS_GENERALES.FN_OBTENERPROVINCI
    APORSECDIVISION(@SECUENCIALDIVPOLRESIDENCIA),
    FBS_GENERALES.FN_OBTENERCANTONPORSECDIVISION(@SECUENCIALDIVPOLRESIDENCIA),
    FBS_GENERALES.FN_OBTENERPARROQUIAPORSECDIVISION(@SECUENCIALDIVPOLRESIDENCIA);
END

--Actualiza Cliente
IF(ISNULL((SELECT TOP(1) DM.ID
            FROM DW_CLIENTES.DIMCLIENTE DM
            WHERE DM.ID = @SECUENCIALCLIENTE), 0)<>0)
BEGIN
    IF(ISNULL((SELECT TOP(1) DM.NOMBRE
                FROM DW_CLIENTES.DIMCLIENTE DM
                WHERE DM.NOMBRE = @CLIENTE
                AND DM.ID = @SECUENCIALCLIENTE), 'NE')='NE')
    BEGIN
        UPDATE DW_CLIENTES.DIMCLIENTE
        SET NOMBRE = @CLIENTE
        WHERE ID=@SECUENCIALCLIENTE
    END
END

--Inserta table dimfecha
SET @IDFECHA= CAST(YEAR(@FECHAADJUDICACION) AS nvarchar) + Replace ( str
( datepart(mm, @FECHAADJUDICACION), 2), ' ', '0' ) + Replace ( str ( datepart(mm,
@FECHAADJUDICACION), 2), ' ', '0' ) ;
SET @TRIMESTRE= CASE WHEN DATEPART(qq,@FECHAADJUDICACION)=1
                    THEN
                    'PRIMER TRIMESTRE'
                    ELSE
                    CASE WHEN
DATEPART (qq, @FECHAADJUDICACION)=2
                    THEN

```

```

'SEGUNDO TRIMESTRE'
ELSE
CASE WHEN
DATEPART (qq, @FECHAADJUDICACION) =3
THEN
'TERCER TRIMESTRE'
ELSE
CASE WHEN
DATEPART (qq, @FECHAADJUDICACION) =4
THEN
'CUARTO TRIMESTRE'
END
END
END;

SET LANGUAGE spanish;
SET @MES=DATENAME (month, @FECHAADJUDICACION) ;

SET @DIA=DATENAME (WEEKDAY, @FECHAADJUDICACION) ;

IF (ISNULL ((SELECT TOP (1) DF.ID
FROM DW_GENERALES.DIMFECHA DF
WHERE DF.ID=@IDFECHA), 0) =0)
BEGIN
INSERT INTO DW_GENERALES.DIMFECHA
VALUES (@IDFECHA, YEAR (@FECHAADJUDICACION), @TRIMESTRE, @MES, @DIA) ;
END

--Inserta tabla de hechos
IF (ISNULL ((SELECT TOP (1) HP.ID_PRESTAMO
FROM DW_CARTERA.HPRESTAMO HP
WHERE HP.ID_PRESTAMO=@SECUENCIALPRESTAMO
AND HP.ID_CLIENTE=@SECUENCIALCLIENTE
AND HP.ID_OFICIAL=@CODIGOOFICIAL
AND HP.ID_FECHA=@IDFECHA), 0) =0)
BEGIN
INSERT INTO DW_CARTERA.HPRESTAMO
VALUES (@SECUENCIALCLIENTE, @SECUENCIALPRESTAMO, @CODIGOOFICIAL, @IDFECHA, @DEUDAINICIAL, @SALDO) ;
END

FETCH NEXT FROM DWCR01
INTO @SECUENCIALPRESTAMO,
@NUMEROPRESTAMO,
@FECHAADJUDICACION,
@CALIFICACIONCONTABLE,
@DESTINOFINANCIERO,
@OFICINAPRESTAMO,
@OFICIALCREDITO,
@CIUDADOFICINA,
@SECUENCIALCLIENTE,
@CLIENTE,
@IDCLIENTE,
@TIPOCLIENTE,
@ESTADOPRESTAMO,
@TIPOPRESTAMO,
@CODIGOOFICIAL,
@DEUDAINICIAL,
@SECUENCIALDIVPOLRESIDENCIA,
@SALDO
END
CLOSE DWCR01
DEALLOCATE DWCR01

END

```

- Vista

```

CREATE PROCEDURE [dbo].[CREADWVISTA]

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    DECLARE @SECUENCIALCUENTA INT;
    DECLARE @OFICINACUENTA NVARCHAR(200);
    DECLARE @OFICIAL NVARCHAR(200);
    DECLARE @CIUDADOFICINA NVARCHAR(30);
    DECLARE @SECUENCIALCLIENTE INT;
    DECLARE @CLIENTE NVARCHAR(200);
    DECLARE @IDCLIENTE NVARCHAR(20);
    DECLARE @TIPOCLIENTE NVARCHAR(50);
    DECLARE @TIPOCUENTA NVARCHAR(50);
    DECLARE @CODIGOOFICIAL NVARCHAR(20);
    DECLARE @SECUENCIALDIVPOLRESIDENCIA INT;
    DECLARE @FECHACREACION DATETIME;

    DECLARE @IDFECHA NVARCHAR(10);
    DECLARE @TRIMESTRE NVARCHAR(20);
    DECLARE @MES NVARCHAR(20);
    DECLARE @DIA NVARCHAR(20);

    DECLARE DWCR02 CURSOR SCROLL FOR
    SELECT CM.SECUENCIAL SECUENCIALCUENTA,
           D.NOMBRE OFICINACUENTA,
           PO.NOMBREUNIDO OFICIAL,
           O.CIUDAD CIUDADOFICINA,
           C.SECUENCIAL SECUENCIALCLIENTE,
           PEC.NOMBREUNIDO CLIENTE,
           PEC.IDENTIFICACION IDCLIENTE,
           CASE WHEN
                TI.PARAPERSONANATURAL=1
            THEN
                'PERSONAS NATURALES'
            ELSE
                'PERSONAS JURIDICAS'
            END TIPOCLIENTE,
           TC.NOMBRE TIPOCUENTA ,
           U.CODIGO CODIGOOFICIAL ,
           PO.SECUENCIALDIVPOLRESIDENCIA,
           CM.FECHASISTEMACREACION FECHACREACION
    FROM [REDACTED] CM
    INNER JOIN [REDACTED] O ON
    O.SECUENCIALDIVISION=CM.SECUENCIALOFICINA
    INNER JOIN [REDACTED] D ON
    D.SECUENCIAL=O.SECUENCIALDIVISION
    INNER JOIN [REDACTED] U ON U.CODIGO =
    CM.CODIGOSUARIOOFICIAL
    INNER JOIN [REDACTED] UC ON
    UC.CODIGOSUARIO=U.CODIGO
    INNER JOIN [REDACTED] PO ON
    PO.SECUENCIAL=UC.SECUENCIALPERSONA
    INNER JOIN [REDACTED] CC ON
    CC.SECUENCIALCUENTA=CM.SECUENCIAL
    INNER JOIN [REDACTED] C ON
    C.SECUENCIAL=CC.SECUENCIALCLIENTE
    INNER JOIN [REDACTED] PEC ON
    PEC.SECUENCIAL=C.SECUENCIALPERSONA
    INNER JOIN [REDACTED] TI ON
    TI.SECUENCIAL=PEC.SECUENCIALTIPOIDENTIFICACION
    INNER JOIN [REDACTED] TC ON
    TC.CODIGO=CM.CODIGOTIPOCUENTA

    OPEN DWCR02

```

```

FETCH FIRST FROM DWCR02
INTO @SECUENCIALCUENTA,
@OFICINACUENTA,
@OFICIAL,
@CIUDADOFICINA,
@SECUENCIALCLIENTE,
@CLIENTE,
@IDCLIENTE,
@TIPOCLIENTE,
@TIPOCUENTA,
@CODIGOOFICIAL,
@SECUENCIALDIVPOLRESIDENCIA,
@FECHACREACION

WHILE (@@FETCH_STATUS = 0)
BEGIN
--Dimensión Préstamo
IF(ISNULL((SELECT DM.ID
FROM DW_CAPTACIONESVISTA.DIMCUENTA DM
WHERE DM.ID = @SECUENCIALCUENTA),0)=0)
BEGIN
INSERT INTO DW_CAPTACIONESVISTA.DIMCUENTA
VALUES(@SECUENCIALCUENTA,@TIPOCUENTA,@OFICINACUENTA,@CIUDADOFICINA);
END

--Dimensión Oficial
--Inserta un Nuevo Oficial
IF(ISNULL((SELECT TOP(1) DMO.ID
FROM DW_SEGURIDADES.DIMOFICIAL DMO
WHERE DMO.ID = @CODIGOOFICIAL),'NE')='NE')
BEGIN
INSERT INTO DW_SEGURIDADES.DIMOFICIAL
VALUES(@CODIGOOFICIAL,@OFICIAL);
END

--Actualiza oficial
IF(ISNULL((SELECT TOP(1) DM.ID
FROM DW_SEGURIDADES.DIMOFICIAL DM
WHERE DM.ID = @CODIGOOFICIAL),'NE')<>'NE')
BEGIN
IF(ISNULL((SELECT TOP(1) DMO.ID
FROM DW_SEGURIDADES.DIMOFICIAL DMO
WHERE DMO.NOMBRE = @OFICIAL
AND DMO.ID= @CODIGOOFICIAL),'NE')='NE')
BEGIN
UPDATE DW_SEGURIDADES.DIMOFICIAL
SET NOMBRE = @OFICIAL
WHERE ID=@CODIGOOFICIAL
END
END

--Dimensión Cliente
--Inserta un Nuevo Cliente
IF(ISNULL((SELECT TOP(1) DM.ID
FROM DW_CLIENTES.DIMCLIENTE DM
WHERE DM.ID = @SECUENCIALCLIENTE),0)=0)
BEGIN
INSERT INTO DW_CLIENTES.DIMCLIENTE
VALUES(@SECUENCIALCLIENTE,@CLIENTE,@TIPOCLIENTE,FBS_GENERALES.FN_OBTENER
PROVINCIAPORSECDIVISION(@SECUENCIALDIVPOLRESIDENCIA),
FBS_GENERALES.FN_OBTENERCANTONPORSECDIVISION(@SECUENCIALDIVPOLRESIDENCIA
),
FBS_GENERALES.FN_OBTENERPARROQUIAPORSECDIVISION(@SECUENCIALDIVPOLRESIDEN
CIA));
END

--Actualiza Cliente

```



```

        IF (ISNULL((SELECT TOP(1) DM.ID
                    FROM DW_CLIENTES.DIMCLIENTE DM
                    WHERE DM.ID = @SECUENCIALCLIENTE), 0) <> 0)
    BEGIN
        IF (ISNULL((SELECT TOP(1) DM.NOMBRE
                    FROM DW_CLIENTES.DIMCLIENTE DM
                    WHERE DM.NOMBRE = @CLIENTE
                    AND DM.ID = @SECUENCIALCLIENTE), 'NE') = 'NE')
        BEGIN
            UPDATE DW_CLIENTES.DIMCLIENTE
            SET NOMBRE = @CLIENTE
            WHERE ID=@SECUENCIALCLIENTE
        END
    END

    --Inserta table dimfecha
    SET @IDFECHA= CAST(YEAR(@FECHACREACION) AS nvarchar) + Replace (
    str ( datepart(mm, @FECHACREACION), 2), ' ', '0' ) + Replace ( str (
    datepart(mm, @FECHACREACION), 2), ' ', '0' ) ;
    SET @TRIMESTRE= CASE WHEN DATEPART(qq, @FECHACREACION)=1
        THEN
        'PRIMER TRIMESTRE'
    ELSE
        CASE WHEN
            DATEPART(qq, @FECHACREACION)=2
                THEN
                'SEGUNDO TRIMESTRE'
            ELSE
                CASE WHEN
                    DATEPART(qq, @FECHACREACION)=3
                        THEN
                        'TERCER TRIMESTRE'
                    ELSE
                        CASE WHEN
                            DATEPART(qq, @FECHACREACION)=4
                                THEN
                                'CUARTO
                                TRIMESTRE'
                            END
                        END
                    END
                END
            END
        END
    END;

    SET LANGUAGE spanish;
    SET @MES=DATENAME(month, @FECHACREACION);

    SET @DIA=DATENAME(WEEKDAY, @FECHACREACION);

    IF (ISNULL((SELECT TOP(1) DF.ID
                FROM DW_GENERALES.DIMFECHA DF
                WHERE DF.ID=@IDFECHA), 0)=0)
    BEGIN
        INSERT INTO DW_GENERALES.DIMFECHA
        VALUES (@IDFECHA, YEAR(@FECHACREACION), @TRIMESTRE, @MES, @DIA);
    END

    --Inserta tabla de hechos
    IF (ISNULL((SELECT TOP(1) HC.ID_CUENTA
                FROM DW_CAPTACIONESVISTA.H_CUENTA HC
                WHERE HC.ID_CUENTA=@SECUENCIALCUENTA
                AND HC.ID_CLIENTE=@SECUENCIALCLIENTE
                AND HC.ID_OFICIAL=@CODIGOOFICIAL
                AND HC.ID_FECHA=@IDFECHA), 0)=0)
    BEGIN
        INSERT INTO DW_CAPTACIONESVISTA.H_CUENTA
        VALUES (@SECUENCIALCUENTA, @SECUENCIALCLIENTE, @CODIGOOFICIAL, @IDFECHA);
    END

    FETCH NEXT FROM DWCR02
    INTO @SECUENCIALCUENTA,

```

```

        @OFICINACUENTA,
        @OFICIAL,
        @CIUDADOFICINA,
        @SECUENCIALCLIENTE,
        @CLIENTE,
        @IDCLIENTE,
        @TIPOCLIENTE,
        @TIPOCUENTA,
        @CODIGOOFICIAL,
        @SECUENCIALDIVPOLRESIDENCIA,
        @FECHACREACION

    END

    CLOSE DWCR02
    DEALLOCATE DWCR02

END

```

- **Plazo**

```

CREATE PROCEDURE [dbo].[CREADWPLAZO]

AS
BEGIN

    SET NOCOUNT ON;

    DECLARE @SECUENCIALDEPOSITO INT;
    DECLARE @OFICINADEPOSITO NVARCHAR(200);
    DECLARE @OFICIAL NVARCHAR(200);
    DECLARE @CIUDADOFICINA NVARCHAR(30);
    DECLARE @SECUENCIALCLIENTE INT;
    DECLARE @CLIENTE NVARCHAR(200);
    DECLARE @IDCLIENTE NVARCHAR(20);
    DECLARE @TIPOCLIENTE NVARCHAR(50);
    DECLARE @TIPOCUENTA NVARCHAR(50);
    DECLARE @CODIGOOFICIAL NVARCHAR(20);
    DECLARE @SECUENCIALDIVPOLRESIDENCIA INT;
    DECLARE @FECHACREACION DATETIME;
    DECLARE @MONTO DECIMAL(18,2);

    DECLARE @IDFECHA NVARCHAR(10);
    DECLARE @TRIMESTRE NVARCHAR(20);
    DECLARE @MES NVARCHAR(20);
    DECLARE @DIA NVARCHAR(20);

    DECLARE DWCR02 CURSOR SCROLL FOR
    SELECT DM.SECUENCIAL SECUENCIALDEPOSITO,
           D.NOMBRE OFICINADEPOSITO,
           PO.NOMBREUNIDO OFICIAL,
           O.CIUDAD CIUDADOFICINA,
           C.SECUENCIAL SECUENCIALCLIENTE,
           PEC.NOMBREUNIDO CLIENTE,
           PEC.IDENTIFICACION IDCLIENTE,
           CASE WHEN
                TI.PARAPERSONANATURAL=1
            THEN
                'PERSONAS NATURALES'
            ELSE
                'PERSONAS JURIDICAS'
            END TIPOCLIENTE,
           TD.NOMBRE TIPOCUENTA ,
           U.CODIGO CODIGOOFICIAL ,
           PO.SECUENCIALDIVPOLRESIDENCIA,

```

```

DM.FECHACREACION FECHACREACION,
DM.MONTO MONTO

FROM [REDACTED] DM
INNER JOIN [REDACTED] O ON
O.SECUENCIALDIVISION=DM.SECUENCIALOFICINA
INNER JOIN [REDACTED] D ON
D.SECUENCIAL=O.SECUENCIALDIVISION
INNER JOIN [REDACTED] U ON U.CODIGO = DM.CODIGOUSUARIO
INNER JOIN [REDACTED] UC ON
UC.CODIGOUSUARIO=U.CODIGO
INNER JOIN [REDACTED] PO ON
PO.SECUENCIAL=UC.SECUENCIALPERSONA
INNER JOIN [REDACTED] DC ON
DC.SECUENCIALDEPOSITO=DM.SECUENCIAL
INNER JOIN [REDACTED] C ON
C.SECUENCIAL=DC.SECUENCIALCLIENTE
INNER JOIN [REDACTED] PEC ON
PEC.SECUENCIAL=C.SECUENCIALPERSONA
INNER JOIN [REDACTED] TI ON
TI.SECUENCIAL=PEC.SECUENCIALTIPOIDENTIFICACION
INNER JOIN [REDACTED] TD ON
TD.CODIGO=DM.CODIGOTIPODEPOSITO

OPEN DWCR02
FETCH FIRST FROM DWCR02
INTO @SECUENCIALDEPOSITO,
@OFICINADEPOSITO,
@OFICIAL,
@CIUDADOFICINA,
@SECUENCIALCLIENTE,
@CLIENTE,
@IDCLIENTE,
@TIPOCLIENTE,
@TIPOCUENTA,
@CODIGOOFICIAL,
@SECUENCIALDIVPOLRESIDENCIA,
@FECHACREACION,
@MONTO

WHILE (@@FETCH_STATUS = 0)
BEGIN
--Dimensión Depósito
IF (ISNULL((SELECT DM.ID
FROM DW_CAPTACIONESPLAZO.DIMDEPOSITOMAESTRO DM
WHERE DM.ID = @SECUENCIALDEPOSITO),0)=0)
BEGIN
INSERT INTO DW_CAPTACIONESPLAZO.DIMDEPOSITOMAESTRO
VALUES (@SECUENCIALDEPOSITO,@TIPOCUENTA,@OFICINADEPOSITO,@CIUDADOFICINA);
END

--Dimensión Oficial
--Inserta un Nuevo Oficial
IF (ISNULL((SELECT TOP(1) DMO.ID
FROM DW_SEGURIDADES.DIMOFICIAL DMO
WHERE DMO.ID = @CODIGOOFICIAL), 'NE')='NE')
BEGIN
INSERT INTO DW_SEGURIDADES.DIMOFICIAL
VALUES (@CODIGOOFICIAL,@OFICIAL);
END

--Actualiza oficial
IF (ISNULL((SELECT TOP(1) DM.ID
FROM DW_SEGURIDADES.DIMOFICIAL DM
WHERE DM.ID = @CODIGOOFICIAL), 'NE') <> 'NE')
BEGIN
IF (ISNULL((SELECT TOP(1) DMO.ID
FROM DW_SEGURIDADES.DIMOFICIAL DMO
WHERE DMO.NOMBRE = @OFICIAL
AND DMO.ID= @CODIGOOFICIAL), 'NE')='NE')
BEGIN

```

```

UPDATE DW_SEGURIDADES.DIMOFICIAL
SET NOMBRE = @OFICIAL
WHERE ID=@CODIGOOFICIAL
END

END

--Dimensión Cliente
--Inserta un Nuevo Cliente
IF(ISNULL((SELECT TOP(1) DM.ID
FROM DW_CLIENTES.DIMCLIENTE DM
WHERE DM.ID = @SECUENCIALCLIENTE),0)=0)

BEGIN
INSERT INTO DW_CLIENTES.DIMCLIENTE

VALUES (@SECUENCIALCLIENTE,@CLIENTE,@TIPOCLIENTE,FBS_GENERALES.FN_OBTENERPROV
INCIAPORSECDIVISION(@SECUENCIALDIVPOLRESIDENCIA),

FBS_GENERALES.FN_OBTENERCANTONPORSECDIVISION(@SECUENCIALDIVPOLRESIDENCIA,
FBS_GENERALES.FN_OBTENERPARROQUIAPORSECDIVISION(@SECUENCIALDIVPOLRESIDENCIA
));
END

--Actualiza Cliente
IF(ISNULL((SELECT TOP(1) DM.ID
FROM DW_CLIENTES.DIMCLIENTE DM
WHERE DM.ID = @SECUENCIALCLIENTE),0)>0)

BEGIN
IF(ISNULL((SELECT TOP(1) DM.NOMBRE
FROM DW_CLIENTES.DIMCLIENTE DM
WHERE DM.NOMBRE = @CLIENTE
AND DM.ID = @SECUENCIALCLIENTE),'NE')='NE')
BEGIN
UPDATE DW_CLIENTES.DIMCLIENTE
SET NOMBRE = @CLIENTE
WHERE ID=@SECUENCIALCLIENTE
END
END

END

--Inserta table dimfecha
SET @IDFECHA= CAST(YEAR(@FECHACREACION) AS nvarchar) + Replace ( str
( datepart(mm, @FECHACREACION), 2), ' ', '0' ) + Replace ( str ( datepart(mm,
@FECHACREACION), 2), ' ', '0' ) ;
SET @TRIMESTRE= CASE WHEN DATEPART (qq,@FECHACREACION)=1
THEN
'PRIMER TRIMESTRE'
ELSE
CASE WHEN
DATEPART (qq, @FECHACREACION) =2
THEN
'SEGUNDO TRIMESTRE'
ELSE
CASE WHEN
DATEPART (qq, @FECHACREACION) =3
THEN
'TERCER TRIMESTRE'
ELSE
CASE WHEN
DATEPART (qq, @FECHACREACION) =4
THEN
'CUARTO
TRIMESTRE'
END
END
END
END;

SET LANGUAGE spanish;
SET @MES=DATENAME(month,@FECHACREACION);

SET @DIA=DATENAME(WEEKDAY,@FECHACREACION);

```

```

        IF (ISNULL((SELECT TOP(1) DF.ID
                    FROM DW_GENERALES.DIMFECHA DF
                    WHERE DF.ID=@IDFECHA), 0) = 0)
    BEGIN
        INSERT INTO DW_GENERALES.DIMFECHA
        VALUES (@IDFECHA, YEAR (@FECHACREACION), @TRIMESTRE, @MES, @DIA);
    END

    --Inserta tabla de hechos
    IF (ISNULL((SELECT TOP(1) HD.ID_DEPOSITO
                FROM DW_CAPTACIONESPLAZO.HDEPOSITOMAESTRO HD
                WHERE HD.ID_DEPOSITO=@SECUENCIALDEPOSITO
                AND HD.ID_CLIENTE=@SECUENCIALCLIENTE
                AND HD.ID_OFICIAL=@CODIGOOFICIAL
                AND HD.ID_FECHA=@IDFECHA), 0) = 0)
    BEGIN
        INSERT INTO DW_CAPTACIONESPLAZO.HDEPOSITOMAESTRO
        VALUES (@SECUENCIALDEPOSITO, @SECUENCIALCLIENTE, @CODIGOOFICIAL, @IDFECHA, @MONTO
    );
    END

    FETCH NEXT FROM DWCR02
    INTO @SECUENCIALDEPOSITO,
    @OFICINADEPOSITO,
    @OFICIAL,
    @CIUDADOFICINA,
    @SECUENCIALCLIENTE,
    @CLIENTE,
    @IDCLIENTE,
    @TIPOCLIENTE,
    @TIPOCUENTA,
    @CODIGOOFICIAL,
    @SECUENCIALDIVPOLRESIDENCIA,
    @FECHACREACION,
    @MONTO

    END

    CLOSE DWCR02
    DEALLOCATE DWCR02

END

```

4.4.2. Establecer proceso de actualización de datos.

La actualización de los datos se los realizará diariamente en la noche en un horario para no saturar la base de datos, mediante Jobs.

El Job tendrá cuatro pasos establecidos, los tres primeros ejecutan los procedimientos almacenados de Cartera, Plazo y Vista para crear el Data Warehouse y el último procesará el cubo creado para actualizar la información.

A continuación se presenta el script para crear el Job indicado:

```

USE [msdb]
GO
BEGIN TRANSACTION
DECLARE @ReturnCode INT
SELECT @ReturnCode = 0
IF NOT EXISTS (SELECT name FROM msdb.dbo.syscategories WHERE name=N'[Uncategorized (Local)]' AND category_class=1)
BEGIN
EXEC @ReturnCode = msdb.dbo.sp_add_category @class=N'JOB', @type=N'LOCAL', @name=N'[Uncategorized (Local)]'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback

END

DECLARE @jobId BINARY(16)
EXEC @ReturnCode =  msdb.dbo.sp_add_job @job_name=N'ACTUALIZACIONDW',
        @enabled=1,
        @notify_level_eventlog=0,
        @notify_level_email=1,
        @notify_level_netsend=0,
        @notify_level_page=0,
        @delete_level=0,
        @description=N'No hay ninguna descripción.',
        @category_name=N'[Uncategorized (Local)]',
        @owner_login_name=N'sa',
        @notify_email_operator_name=N'Notificación', @job_id = @jobId OUTPUT
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
/***** Object: Step [PASO UNO CARGA DW CARTERA]  Script Date: 04/05/2015 22:36:38 *****/
EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId, @step_name=N'PASO UNO CARGA DW CARTERA',
        @step_id=1,
        @cmdexec_success_code=0,
        @on_success_action=3,
        @on_success_step_id=0,
        @on_fail_action=2,
        @on_fail_step_id=0,
        @retry_attempts=0,
        @retry_interval=0,
        @os_run_priority=0, @subsystem=N'TSQL',
        @command=N'DECLARE          @return_value int

EXEC      @return_value = [dbo].[CREADWCARTERA]

SELECT      "Return Value" = @return_value

GO',
        @database_name=N'FBS_11Junio',
        @flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
/***** Object: Step [PASO DOS CARGA DW VISTA]  Script Date: 04/05/2015 22:36:38 *****/
EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId, @step_name=N'PASO DOS CARGA DW VISTA',
        @step_id=2,
        @cmdexec_success_code=0,
        @on_success_action=3,
        @on_success_step_id=0,
        @on_fail_action=2,
        @on_fail_step_id=0,
        @retry_attempts=0,
        @retry_interval=0,
        @os_run_priority=0, @subsystem=N'TSQL',
        @command=N'DECLARE          @return_value int

EXEC      @return_value = [dbo].[CREADWVISTA]

SELECT      "Return Value" = @return_value

GO',
        @database_name=N'FBS_11Junio',

```

```

        @flags=0
    IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
    /***** Object: Step [PASO TRES CARGA DW PLAZO]  Script Date: 04/05/2015 22:36:39 *****/
    EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId, @step_name=N'PASO TRES CARGA DW PLAZO',
        @step_id=3,
        @cmdexec_success_code=0,
        @on_success_action=3,
        @on_success_step_id=0,
        @on_fail_action=2,
        @on_fail_step_id=0,
        @retry_attempts=0,
        @retry_interval=0,
        @os_run_priority=0, @subsystem=N'TSQL',
        @command=N'DECLARE          @return_value int

EXEC      @return_value = [dbo].[CREADWPLAZO]

SELECT    "Return Value" = @return_value

GO',
        @database_name=N'FBS_11Junio',
        @flags=0
    IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
    /***** Object: Step [PROCESA CUBO CARTERA]  Script Date: 04/05/2015 22:36:39 *****/
    EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId, @step_name=N'PROCESA CUBO CARTERA',
        @step_id=4,
        @cmdexec_success_code=0,
        @on_success_action=3,
        @on_success_step_id=0,
        @on_fail_action=2,
        @on_fail_step_id=0,
        @retry_attempts=0,
        @retry_interval=0,
        @os_run_priority=0, @subsystem=N'ANALYSISCOMMAND',
        @command=N'<Batch xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">
        <Process xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:ddl2="http://schemas.microsoft.com/analysiservices/2003/engine/2"
        xmlns:ddl2_2="http://schemas.microsoft.com/analysiservices/2003/engine/2/2"
        xmlns:ddl100_100="http://schemas.microsoft.com/analysiservices/2008/engine/100/100"
        xmlns:ddl200="http://schemas.microsoft.com/analysiservices/2010/engine/200"
        xmlns:ddl200_200="http://schemas.microsoft.com/analysiservices/2010/engine/200/200"
        xmlns:ddl300="http://schemas.microsoft.com/analysiservices/2011/engine/300"
        xmlns:ddl300_300="http://schemas.microsoft.com/analysiservices/2011/engine/300/300">
        <Object>
        <DatabaseID>CUBO FINANCIAL</DatabaseID>
        <CubeID>FBS 11 Junio</CubeID>
        </Object>
        <Type>ProcessFull</Type>
        <WriteBackTableCreation>UseExisting</WriteBackTableCreation>
        </Process>
        </Batch>',
        @server=N'DELLFERNANDO\SQL2012',
        @database_name=N'master',
        @flags=0
    IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
    /***** Object: Step [PROCESA CUBO PLAZO]  Script Date: 04/05/2015 22:36:39 *****/
    EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId, @step_name=N'PROCESA CUBO PLAZO',
        @step_id=5,
        @cmdexec_success_code=0,
        @on_success_action=3,
        @on_success_step_id=0,
        @on_fail_action=2,
        @on_fail_step_id=0,
        @retry_attempts=0,
        @retry_interval=0,
        @os_run_priority=0, @subsystem=N'ANALYSISCOMMAND',
        @command=N'<Batch xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">

```

```

<Process xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ddl2="http://schemas.microsoft.com/analysiservices/2003/engine/2"
xmlns:ddl2_2="http://schemas.microsoft.com/analysiservices/2003/engine/2/2"
xmlns:ddl100_100="http://schemas.microsoft.com/analysiservices/2008/engine/100/100"
xmlns:ddl200="http://schemas.microsoft.com/analysiservices/2010/engine/200"
xmlns:ddl200_200="http://schemas.microsoft.com/analysiservices/2010/engine/200/200"
xmlns:ddl300="http://schemas.microsoft.com/analysiservices/2011/engine/300"
xmlns:ddl300_300="http://schemas.microsoft.com/analysiservices/2011/engine/300/300">
  <Object>
    <DatabaseID>CUBO FINANCIAL</DatabaseID>
    <CubeID>FBS 11 Junio 2</CubeID>
  </Object>
  <Type>ProcessFull</Type>
  <WriteBackTableCreation>UseExisting</WriteBackTableCreation>
</Process>
</Batch>',
        @server=N'localhost\SQL2012',
        @database_name=N'master',
        @flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
/***** Object: Step [PROCESA CUBO VISTA] Script Date: 04/05/2015 22:36:39 *****/
EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId, @step_name=N'PROCESA CUBO VISTA',
        @step_id=6,
        @cmdexec_success_code=0,
        @on_success_action=1,
        @on_success_step_id=0,
        @on_fail_action=2,
        @on_fail_step_id=0,
        @retry_attempts=0,
        @retry_interval=0,
        @os_run_priority=0, @subsystem=N'ANALYSISCOMMAND',
        @command=N'<Batch xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">
  <Process xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ddl2="http://schemas.microsoft.com/analysiservices/2003/engine/2"
xmlns:ddl2_2="http://schemas.microsoft.com/analysiservices/2003/engine/2/2"
xmlns:ddl100_100="http://schemas.microsoft.com/analysiservices/2008/engine/100/100"
xmlns:ddl200="http://schemas.microsoft.com/analysiservices/2010/engine/200"
xmlns:ddl200_200="http://schemas.microsoft.com/analysiservices/2010/engine/200/200"
xmlns:ddl300="http://schemas.microsoft.com/analysiservices/2011/engine/300"
xmlns:ddl300_300="http://schemas.microsoft.com/analysiservices/2011/engine/300/300">
  <Object>
    <DatabaseID>CUBO FINANCIAL</DatabaseID>
    <CubeID>FBS 11 Junio 1</CubeID>
  </Object>
  <Type>ProcessFull</Type>
  <WriteBackTableCreation>UseExisting</WriteBackTableCreation>
</Process>
</Batch>',
        @server=N'localhost\SQL2012',
        @database_name=N'master',
        @flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_update_job @job_id = @jobId, @start_step_id = 1
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_add_jobschedule @job_id=@jobId, @name=N'NUEVA PROGRAMACIÓN',
        @enabled=0,
        @freq_type=4,
        @freq_interval=1,
        @freq_subday_type=1,
        @freq_subday_interval=5,
        @freq_relative_interval=0,
        @freq_recurrence_factor=0,
        @active_start_date=20150426,
        @active_end_date=99991231,
        @active_start_time=213500,
        @active_end_time=235959,
        @schedule_uid=N'e3b1c9f8-75fe-4396-8918-31a50d5da898'

```



```

IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_add_jobserver @job_id = @jobId, @server_name = N'(local)'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
COMMIT TRANSACTION
GOTO EndSave
QuitWithRollback:
    IF (@@TRANSCOUNT > 0) ROLLBACK TRANSACTION
EndSave:

GO

```

4.5. Desarrollo del módulo de Business Intelligence

Para el diseño del Front End o interfaz de usuario final se seguirá la arquitectura establecida por el Sistema Financiero Financiamiento v2.0 para la creación de los formularios de visualización con la secuencia siguiente.

- Escoger el módulo en el que se va a implementar la interfaz gráfica y la carpeta de consulta de datos.

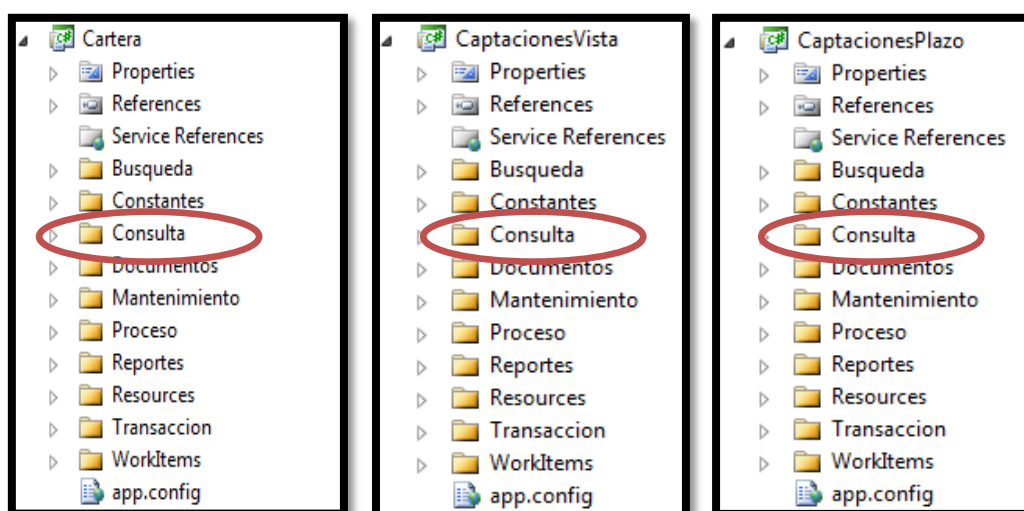


Figura 4.44. Módulo Interfaz Gráfica

- Crear un nuevo componente que será un control del tipo UserControl como base contenedor de los controles de consulta con los siguientes nombres:
 - ConsultaCarteraBI_Vista.
 - ConsultaVistaBI_Vista

- ConsultaPlazoBI_Vista
- A continuación se creará las clases de presentación donde se programará los métodos necesarios para el funcionamiento de los diferentes controles.
 - ConsultaCarteraBI_Presentador.
 - ConsultaVistaBI_Presentador
 - ConsultaPlazoBI_Presentador

4.5.1. Establecer componentes adecuados para mostrar información

Para la visualización de la información se escogieron componentes DevExpress que permitirá la navegación de los datos en todas sus dimensiones para ser analizados de acuerdo a los diferentes criterios requeridos y fueron los siguientes:

PivotGridControl

Este componente permite conectarse con el motor OLAP de Analysis Services para acceder al cubo multidimensional realizado y escoger las diferentes dimensiones y los diferentes criterios de análisis.

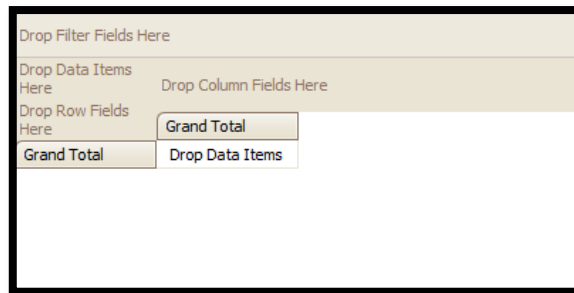


Figura 4.45.Control PivotGridControl

ChartControl

Este componente permitirá tener una visión gráfica del análisis gerencial o de inteligencia de negocios que se realicen en base a los criterios escogidos en el PivotGridControl.

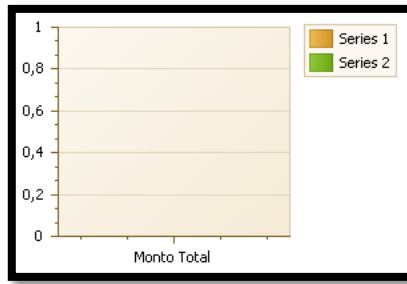


Figura 4.46.Control ChartControl

PopupMenu

Con este componente se presentarán opciones adicionales que permitirá realizar las siguientes opciones sobre el gráfico.

- Mostrar u ocultar una leyenda general informativa.
- Rotar el gráfico 90 grados.
- Escoger el tipo de etiqueta que puede ser solo el valor, el valor y el argumento de selección o ninguna.
- Seleccionar el tipo de gráfico con el que necesitamos realizar el análisis.
- Mostrar u ocultar los ejes además de rotar 45 grados la leyenda para una mejor visualización.

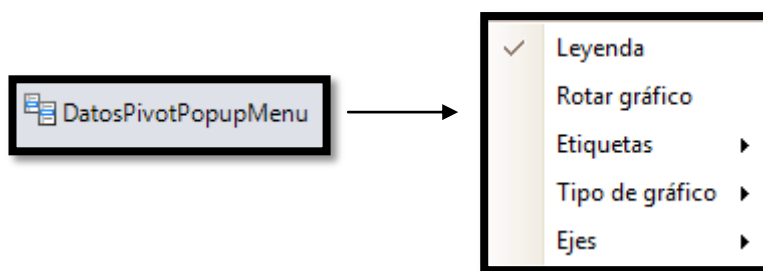


Figura 4.47.Control PopupMenu

Bar

En este componente se ubicará opciones para exportar y guardar el diseño gerencial creado; además de la opción de salir y la visualización de la ruta de la clase de programación.



Figura 4.48.Control BarControl

4.5.2. Elaboración de la interfaz gráfica

Para la elaboración de la interfaz gráfica se escogió una forma de Windows como panel para ingresar todos los componentes necesarios; luego de lo cual iniciaremos con la configuración del PivotGridControl de la siguiente manera:

- Configurar la conexión del motor OLAP indicando la instancia del servidor de base de datos, el nombre del proyecto realizado en el Analisis Services y el nombre del cubo realizado.

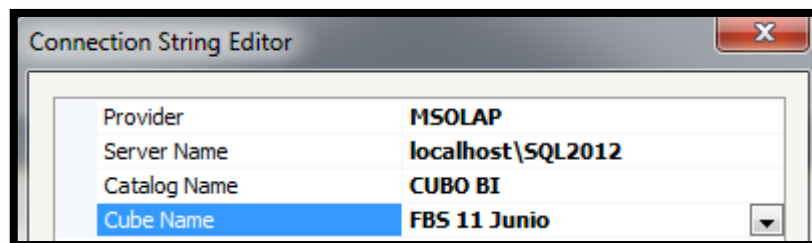


Figura 4.49.Control Conexión motor OLAP

- Escoger las perspectivas de análisis junto con sus indicadores para configurar sus etiquetas y su forma de visualización.

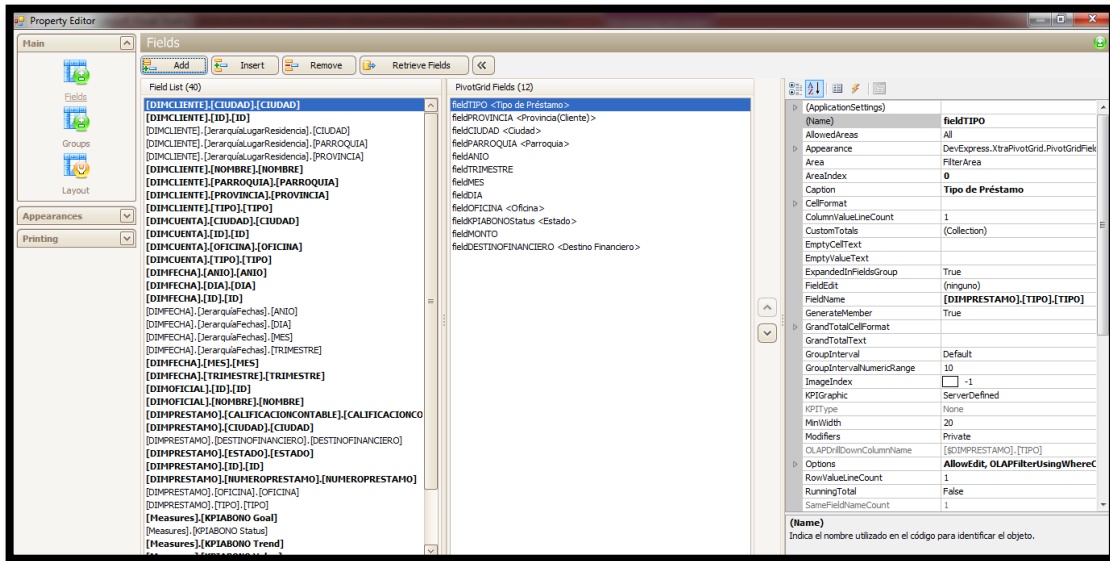


Figura 4.50. Control PivotGridControl – Agregar campos

- Agregar las jerarquías de tiempo y de ubicación geográfica.

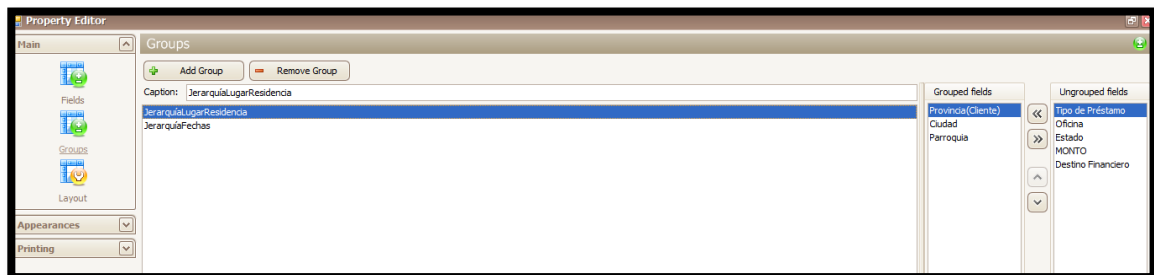


Figura 4.51. Control PivotGridControl – agregar grupos

- Indicar el campo KPI creado y agregar la siguiente propiedad que permite escoger el tipo de gráfico que indicará el estado del KPI.

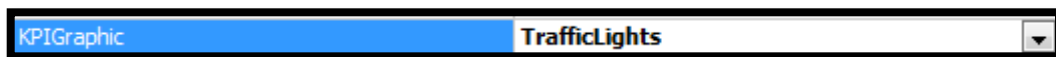


Figura 4.52. Control PivotGridControl – Agregar KPI

El siguiente paso será configurar las propiedades para el componente ChartControl indicando la fuente de donde tomará la información para establecer su naturaleza y diseño:



Figura 4.53. Control ChartControl- Escoger datos

Al fin de la selección de los respectivos componentes y editar sus correspondientes propiedades obtenemos el siguiente diseño:

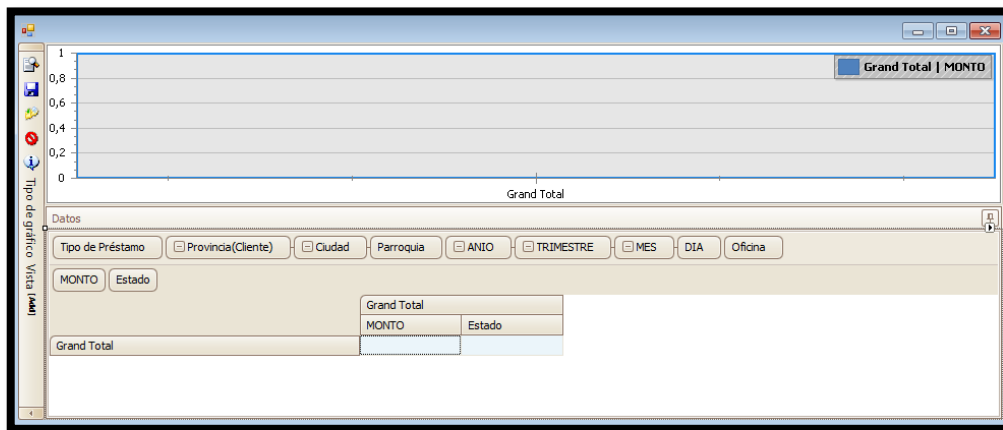


Figura 4.54. Diseño de interfaz

El código para el funcionamiento de los componentes se realizaron con una programación orientada a objetos y utilizando un lenguaje de programación C#, a continuación se presenta el código de los principales métodos desarrollados.

Guardar Modelo

```
void GuardarModelo(string ruta, DataSet datos)
{
    try
    {
        string rutaXmlPivot = ruta.Replace("ger", "xml");
        MemoryStreampivotStream = new MemoryStream();
        _vista.datosPivotGridControl.SaveLayoutToXml(rutaXmlPivot);

        MemoryStreamchartStream = new MemoryStream();
        _vista.chartControl.SaveToStream(chartStream);
    }
}
```

```

string lineas = _vista.verCompactaCheckItem.Checked.ToString()
+ "\r\n" + _vista.totalFilasBarCheckItem.Checked.ToString()
+ "\r\n" + _vista.totalColumnasBarCheckItem.Checked.ToString()
+ "\r\n" + _vista.verLeyendaBarCheckItem.Checked.ToString()
+ "\r\n" + _vista.rotarGraficoBarCheckItem.Checked.ToString()
+ "\r\n" + _vista.VerValorBarCheckItem.Checked.ToString()
+ "\r\n" + _vista.VerArgumentoYValorBarCheckItem.Checked.ToString()
+ "\r\n" + _vista.ocultarEtiquetaBarCheckItem.Checked.ToString()
+ "\r\n" + _vista.barraCheckItem.Checked.ToString()
+ "\r\n" + _vista.barraApiladaCheckItem.Checked.ToString()
+ "\r\n" + _vista.lineaCheckItem.Checked.ToString()
+ "\r\n" + _vista.pastelBarCheckItem.Checked.ToString()
+ "\r\n" + _vista.ocultarEjeXBarCheckItem.Checked.ToString()
+ "\r\n" + _vista.ocultarEjeYBarCheckItem.Checked.ToString()
+ "\r\n" + _vista.rotarEjeXBarCheckItem.Checked.ToString()
+ "\r\n" + _vista.rotarEjeYBarCheckItem.Checked.ToString();

//MemoryStreamdatosStream = new MemoryStream();
//datos.WriteXml(datosStream, XmlWriteMode.WriteSchema);

List<FBSComun.Utilidades.Sistema.Archivo> archivos = new
List<FBSComun.Utilidades.Sistema.Archivo>();
archivos.Add(new FBSComun.Utilidades.Sistema.Archivo("pivotStream.001",
pivotStream.GetBuffer()));
archivos.Add(new FBSComun.Utilidades.Sistema.Archivo("chartStream.002",
chartStream.GetBuffer()));
archivos.Add(new FBSComun.Utilidades.Sistema.Archivo("lineasStream.003",
Encoding.ASCII.GetBytes(lineas)));

//archivos.Add(new FBSComun.Utilidades.Sistema.Archivo("datos.004",
datosStream.GetBuffer()));
FBSComun.Utilidades.Sistema.ComprimirArchivos(ruta, archivos);
pivotStream.Close();
chartStream.Close();
//datosStream.Close();
}
catch (Exception ex)
{
MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error, MessageBoxDefaultButton.Button1);
}
}

```

Recuperar Modelo

```

voidRecuperarModelo()
{
_vista.guardarGraficoSsaveFileDialog.Filter = "Sifizsoft - Modelos de
Gerencia(*.ger)|*.ger";
_vista.recuperarGraficoOpenFileDialog.CheckFileExists = true;
_vista.guardarGraficoSsaveFileDialog.CheckPathExists = true;

if (_vista.recuperarGraficoOpenFileDialog.ShowDialog() == DialogResult.OK)
{
CargaDeCaracteristicasComponentes(_vista.recuperarGraficoOpenFileDialog.FileName);
}
}

```

Vista Preliminar

```

private void VistaPreliminar()
{
PrintingSystem printingSystem1 = new PrintingSystem();
CompositeLink cl = new CompositeLink(printingSystem1);

cl.PaperKind = System.Drawing.Printing.PaperKind.A4;
cl.Landscape = true;
printingSystem1.Links.Add(cl);
}

```

```

PrintableComponentLinkpclChart = new PrintableComponentLink();
PrintableComponentLinkpclPivot = new PrintableComponentLink();
PrintableComponentLinkpclPanel = new PrintableComponentLink();
    _vista.datosPivotGridControl.OptionsView.ShowFilterHeaders = false;

pclChart.Component = _vista.chartControl;
pclPivot.Component = _vista.datosPivotGridControl;

cl.Links.AddRange(new object[] { pclChart, pclPivot });
cl.ShowPreviewDialog();

    printingSystem1.ShowPrintStatusDialog = false;

    _vista.datosPivotGridControl.OptionsView.ShowFilterHeaders = true;
}

```

WorkItem

A continuación se creará una clase de configuración para invocar las ventanas codificadas desde el menú del Sistema Financiero y tendrá el siguiente código:

```

public class ConsultaGerencialBI : GeneralWorkItem
{
    public void Run(IWorkspace workspace, string titulo, string ruta)
    {
        try
        {
            this._workspace = workspace;
            this._vista = (GeneralContenidoVista)this.Items[titulo];
            if (_vista == null)
            {
                this._vista = new GeneralContenidoVista();
                this._vista.Ruta(ruta);
                this._vista.Load += new EventHandler(_vista_Load);
                this._vista.SalirItem.Click += new EventHandler(SalirItem_Click);
                this.SmartParts.Add(this._vista, titulo);
                SmartPartInfoinf = new SmartPartInfo();
                inf.Title = titulo;
                workspace.Show(this._vista, inf);
            }
            else
            {
                workspace.Activate(this._vista);
            }
            catch (Exception err)
            {
                if (err.InnerException == null)
                {
                    MessageBox.Show(err.Message, "Error al abrir la opción del menú", MessageBoxButtons.OK,
                    MessageBoxIcon.Error, MessageBoxDefaultButton.Button1);
                }
                else
                {
                    MessageBox.Show(err.InnerException.Message, "Error al abrir la opción del menú",
                    MessageBoxButtons.OK, MessageBoxIcon.Error, MessageBoxDefaultButton.Button1);
                }
            }
        }

        void SalirItem_Click(object sender, EventArgs e)
        {
            this.Cerrar();
        }

        void _vista_Load(object sender, EventArgs e)
        {
            try
            {
                ConsultaCarteraBI_Vista vista = this.SmartParts.AddNew<ConsultaCarteraBI_Vista>();
                ConsultaCarteraBI_Presentador presentador = new ConsultaCarteraBI_Presentador(vista, this);
            }
            catch { }
        }
    }
}

```



```
this.SmartParts.Add(vista, "ConsultaCarteraBI_Vista");
this.Items.Add(presentador, "ConsultaCarteraBI_Presentador");
this.Workspaces[WorkspaceNames.GeneralContenidoWorkspace].Show(vista);
    }
    catch (Exception err)
    {
        throw err;
    }
}
}
```

4.5.3. Implementación del módulo en el Sistema Financiero Financiamiento

Para realizar la implementación del módulo BI en el sistema financiero Financiamiento será necesario establecer los prerequisites tanto de software como de hardware para el ambiente de desarrollo y de producción descritos a continuación.

Requisitos de Hardware

A continuación se describe los requisitos de hardware tanto para el servidor como para los clientes considerando los equipos disponibles por la empresa.

Estación Cliente

- Procesador Intel i7.
- Disco duro de 500 Gb.
- Ram 8 Gb.

Estación Servidor

- Procesador Intel i7.
- Disco duro de 2Tb.
- Ram 32 Gb.

Requisitos de Software

Sistema Operativo

- Windows 7 u 8.

Herramientas de Desarrollo de Programación.

- Visual Studio 2010.
- Suite de Componentes Dev Express.

Gestor de Base de Datos

- SQL Server Estándar 2012.

Herramientas Complementarias

- Analysis Services.

Con los prerrequisitos disponibles el paso siguiente fue subir las funcionalidades de consulta BI creadas al menú del sistema financiero Financiera de la siguiente manera:

- Crear una funcionalidad en el módulo correspondiente:

Cartera

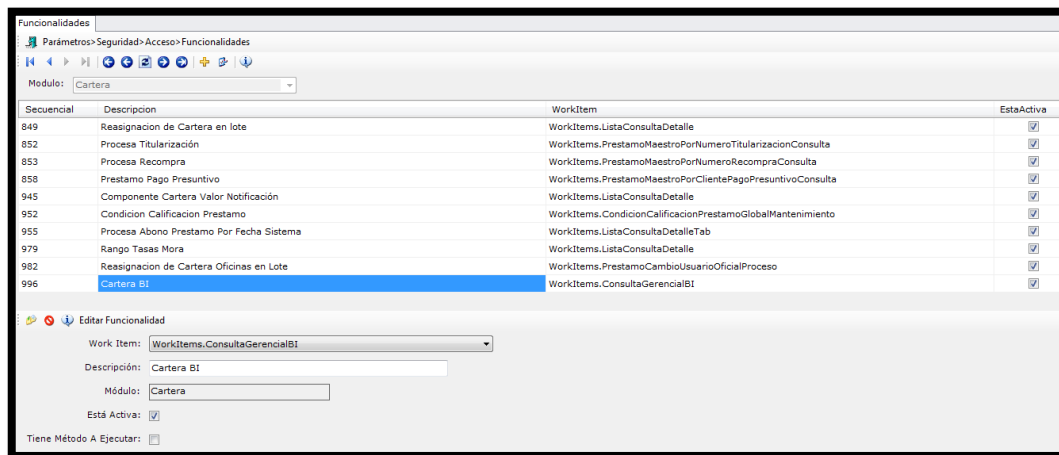


Figura 4.55. Funcionalidad Menú Cartera

Vista

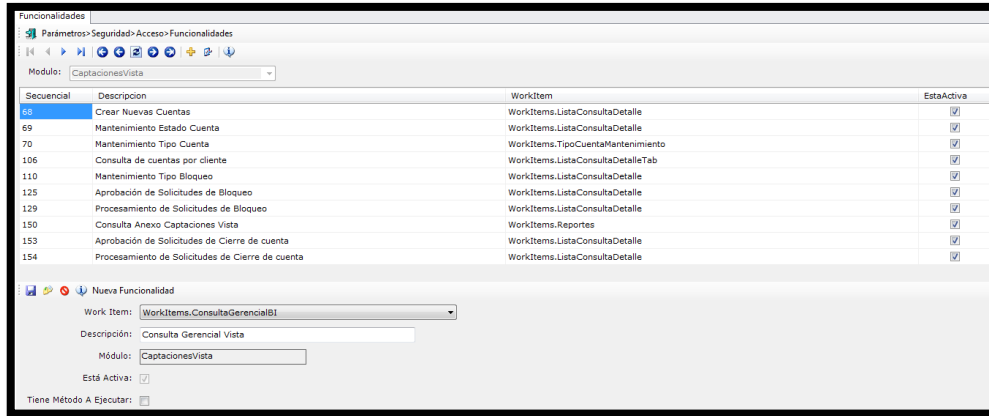


Figura 4.56. Funcionalidad Menú Vista

Plazo

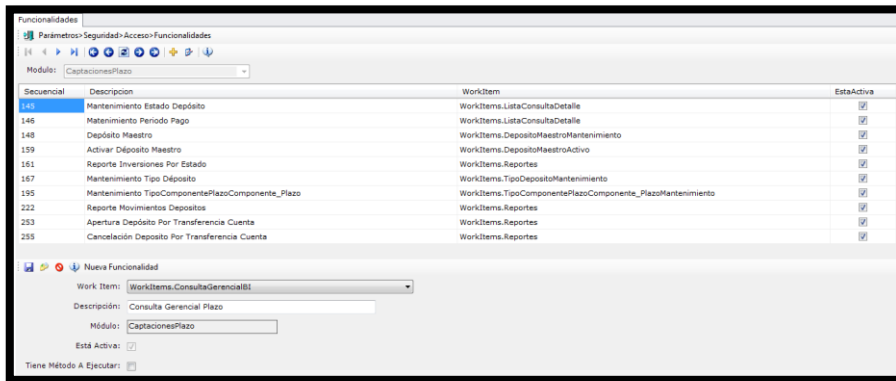


Figura 4.57. Funcionalidad Menú Plazo

- A continuación se asignará el rol de usuario que restringirá el tipo de usuario que tendrán acceso al módulo de Business Intelligence.

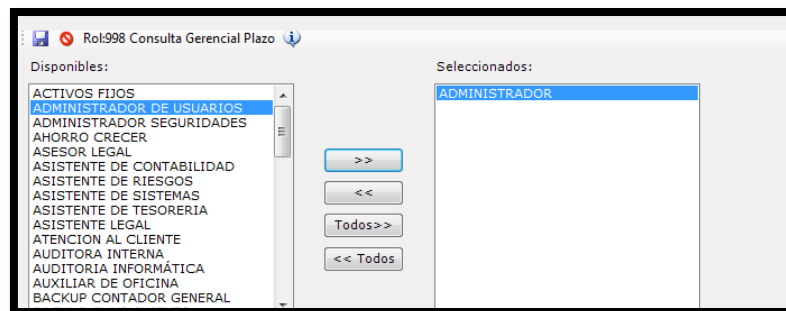


Figura 4.58. Rol Menú Funcionalidad

- Así finalmente obtendremos el siguiente menú.



Figura 4.59. Módulo Business Intelligence

4.5.4. Ejecución de Pruebas

Las pruebas se concentraron en realizar una consulta de datos dinámica que responda los cuestionamientos descritos en el análisis de requerimientos por cada módulo.

Cartera

- Consultar por tipo de préstamo el monto total entregado en el tiempo.

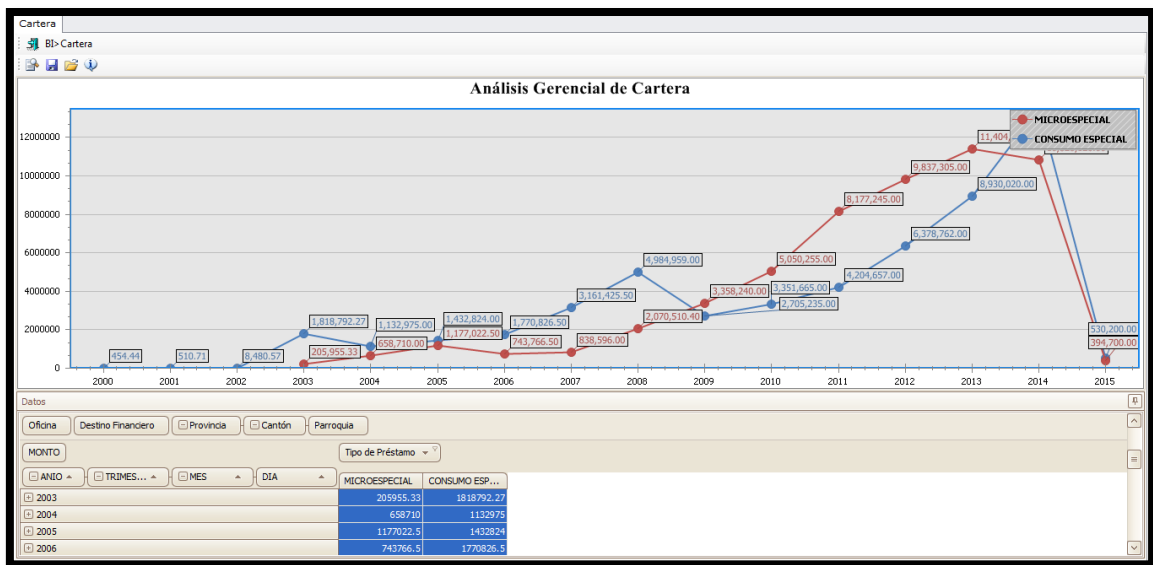


Figura 4.60. Tipo de préstamo el monto total entregado en el tiempo

- Consultar por tipo de préstamo la cantidad de créditos entregados en el tiempo.

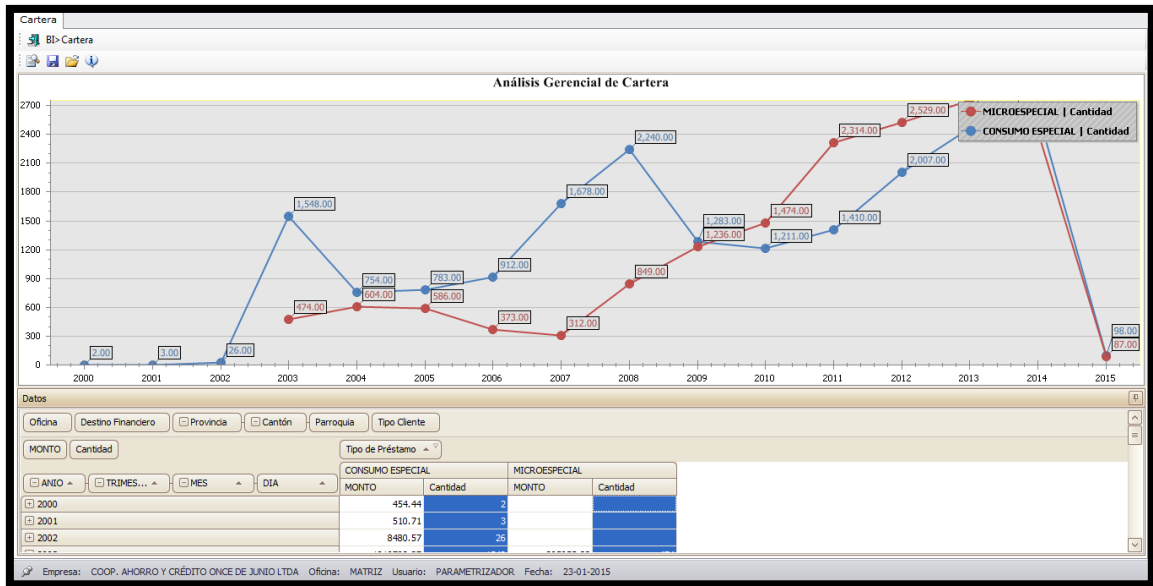


Figura 4.61. Tipo de préstamo la cantidad de créditos entregados en el tiempo

- Comparativa entre oficinas de la cantidad de préstamos entregado en los trimestres de un determinado año.

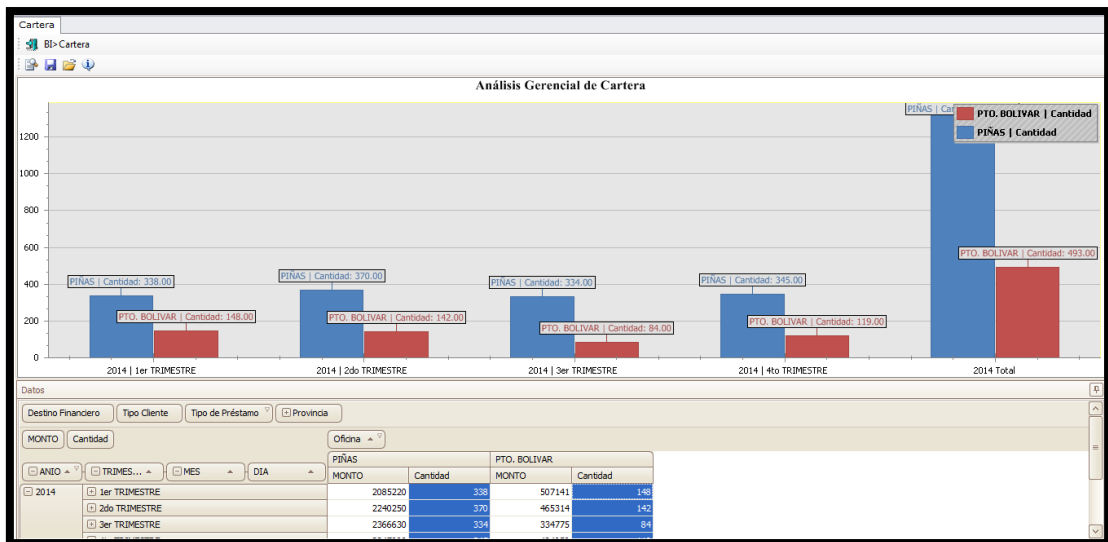


Figura 4.62. Comparativa entre oficinas cartera

- Los montos entregados por destino financiero entre dos años.

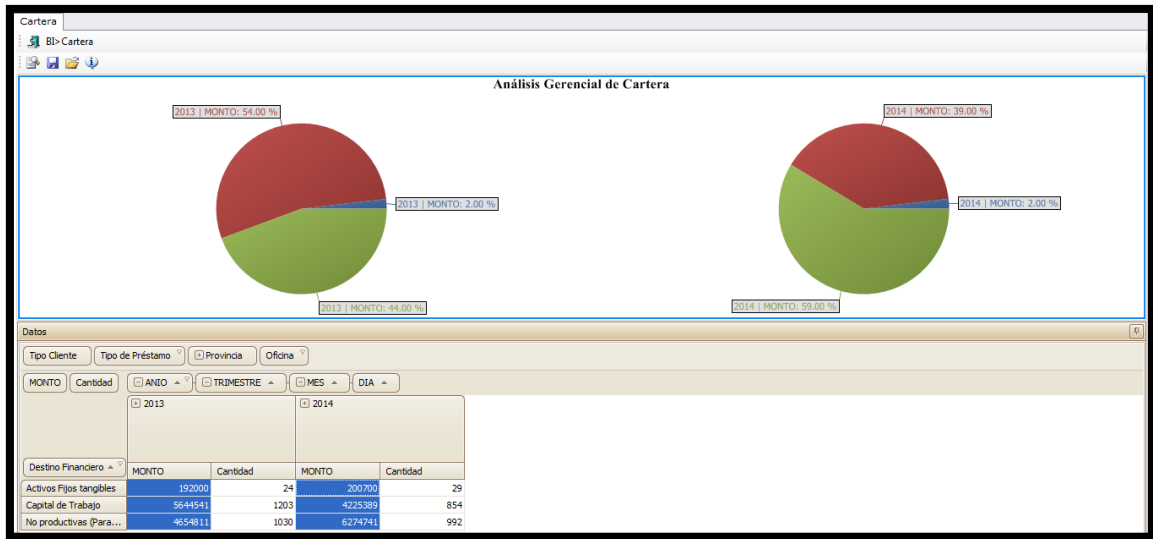


Figura 4.63. Destino Financiero entre dos años

- Cantidad de créditos realizados por un número de oficiales en el tiempo.

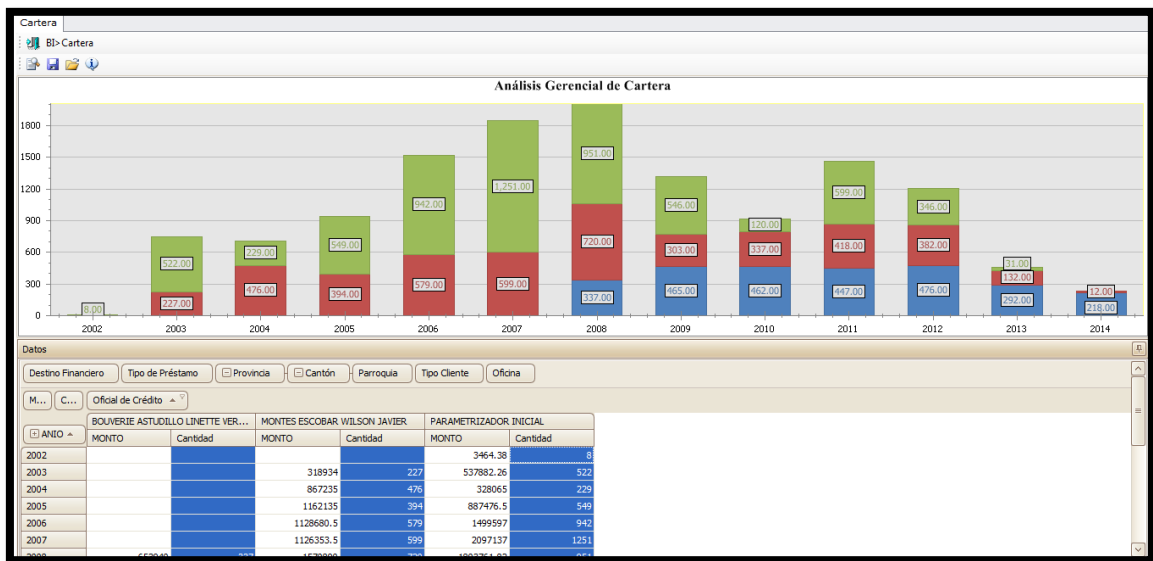


Figura 4.64. Oficiales de Crédito en el tiempo

Captaciones Vista

- Cantidad de cuentas por oficina y usuario.

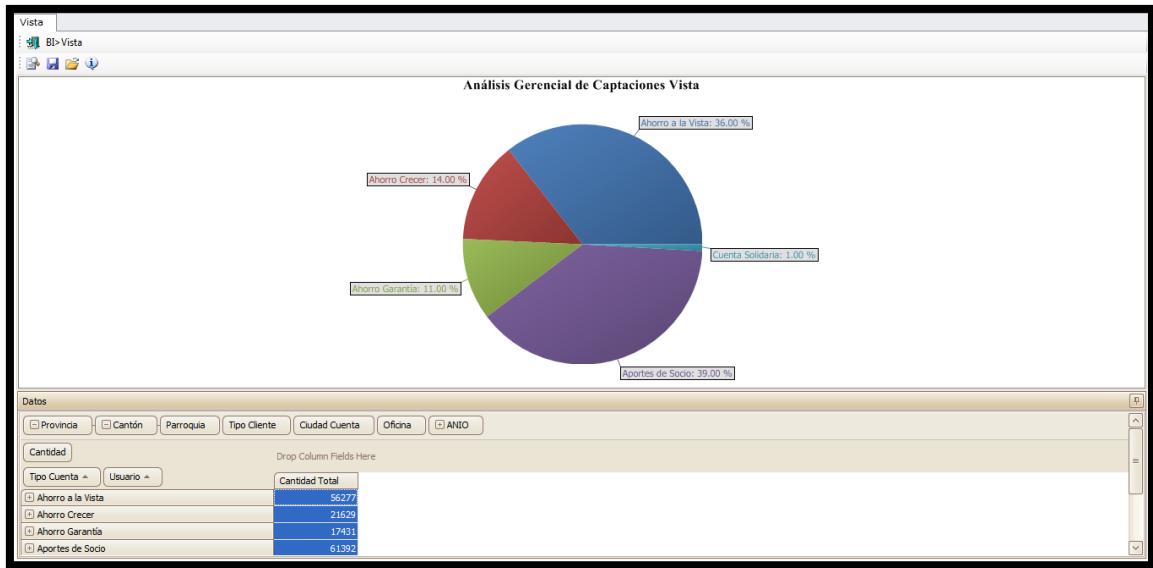


Figura 4.65. Cuentas por oficina y usuario

- Las cuentas captadas por división política de residencia de los clientes.

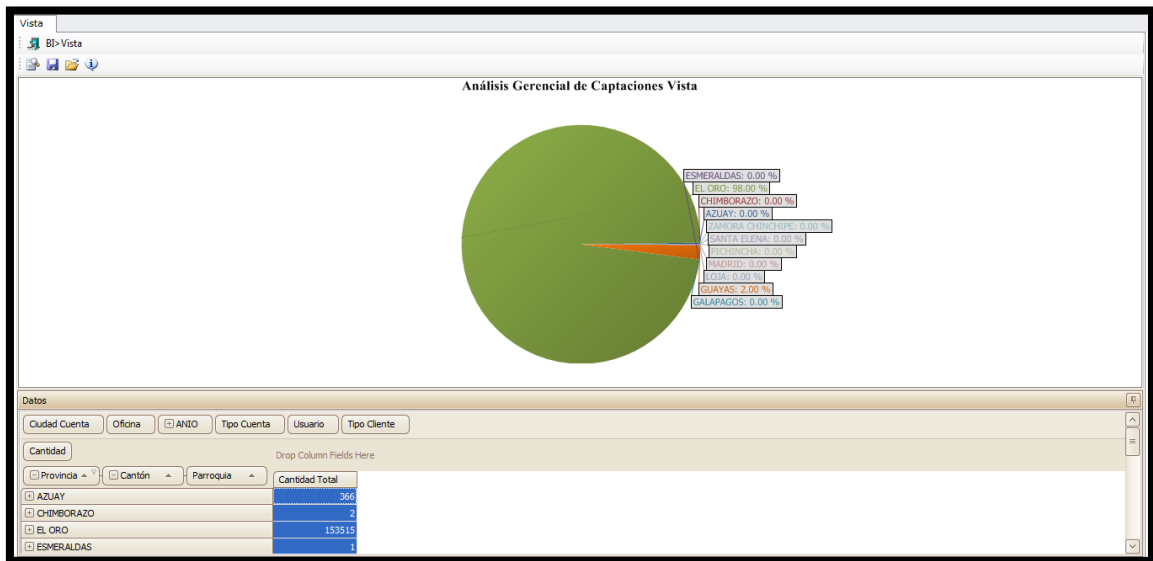


Figura 4.66. División Política Clientes Vista

- Cuentas por tipo de cliente.

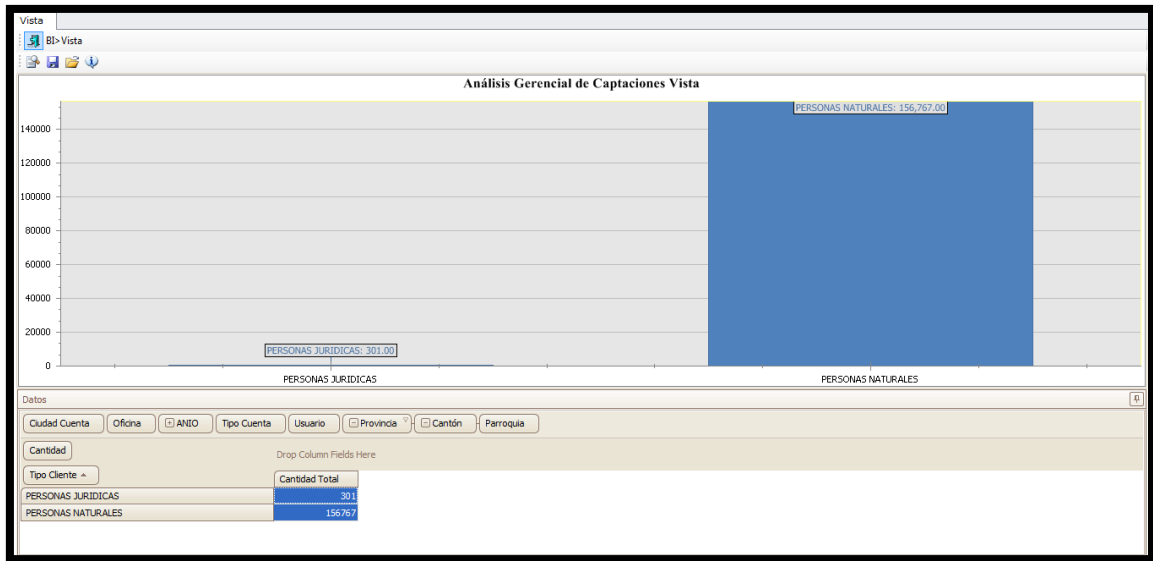


Figura 4.67. Cuentas por tipo de cliente

Captaciones Plazo

- Tipos de depósitos a plazo fijo en el tiempo.

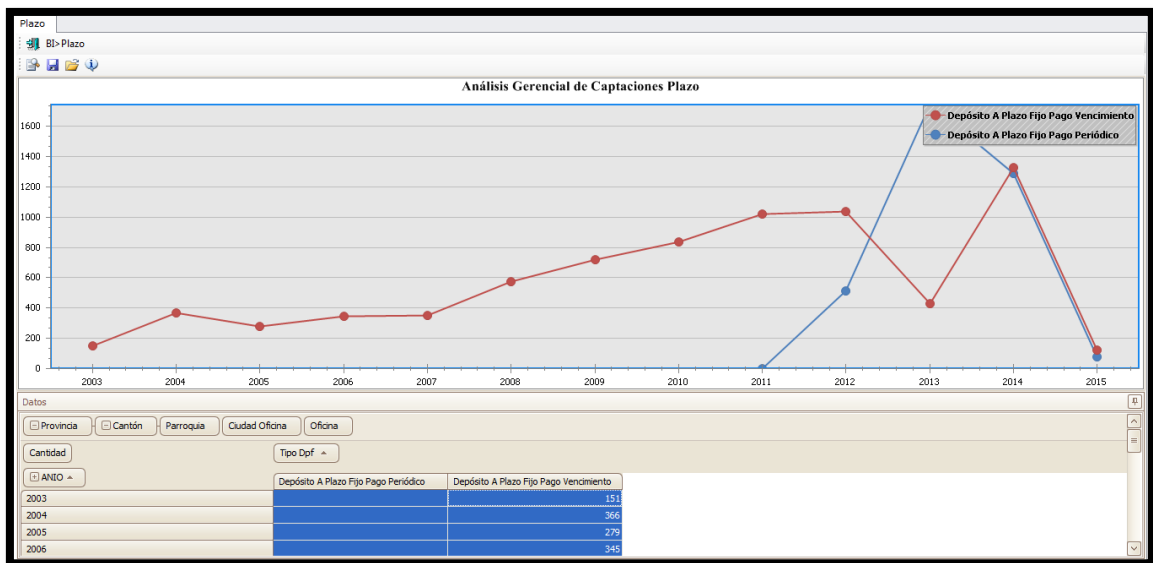


Figura 4.68. Tipos de depósitos a plazo fijo en el tiempo

- Monto total de depósitos por ciudad de todas las oficinas.

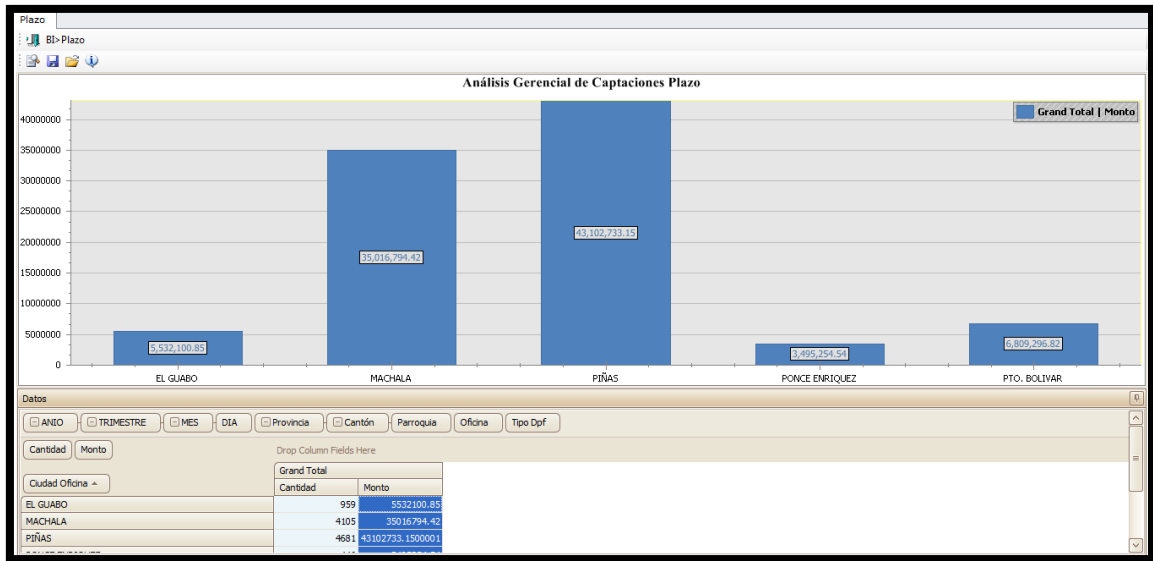


Figura 4.69. Monto Depósitos por Ciudad Oficinas

- Cantidad total de depósitos por ciudad de todas las oficinas.

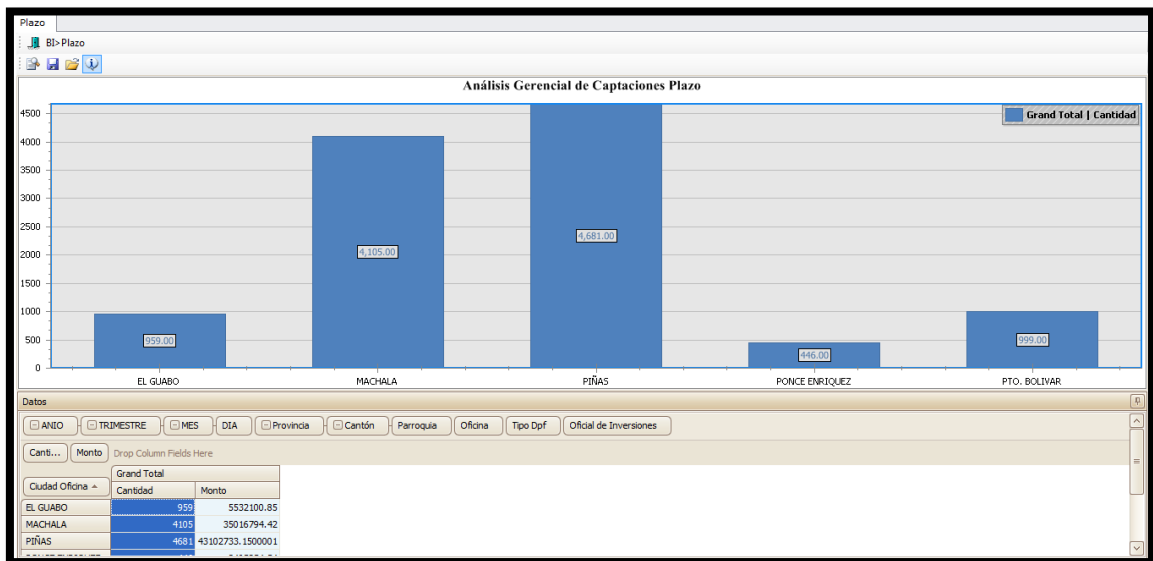


Figura 4.70. Cantidad Depósitos por Ciudad Oficinas

Después de verificar que todas las consultas se realizaron correctamente se procedió a verificar el correcto funcionamiento del resto de componentes.

- **Vista Preliminar** 

En cada consulta se ejecutó la vista preliminar que muestra una ventana con un componente el cual indica la consulta gerencial a manera de documento, el mismo que se puede imprimir, exportar en diferentes formatos o enviar directamente por mail.

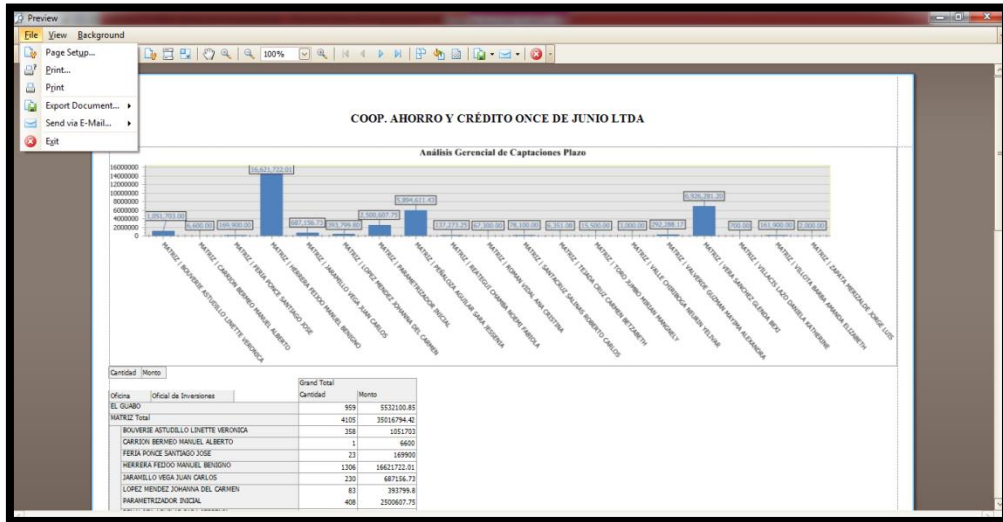


Figura 4.71. Vista Preliminar

- Guardar Modelo 

En cada consulta realizada se guardó el modelo al finalizar el diseño de la consulta en una carpeta así:

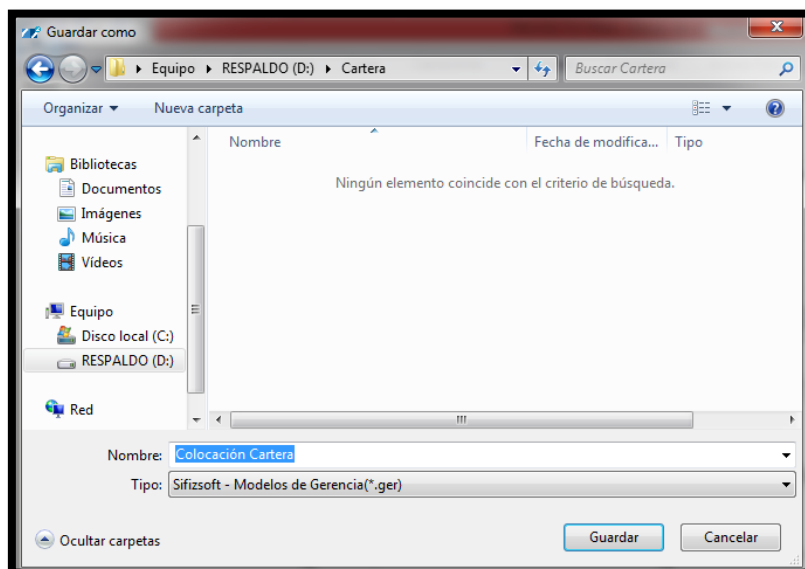


Figura 4.72. Guardar Diseño

- Abrir Diseño 

A la vez se abrió el diseño de cada modelo para verificar que se guardó correctamente.

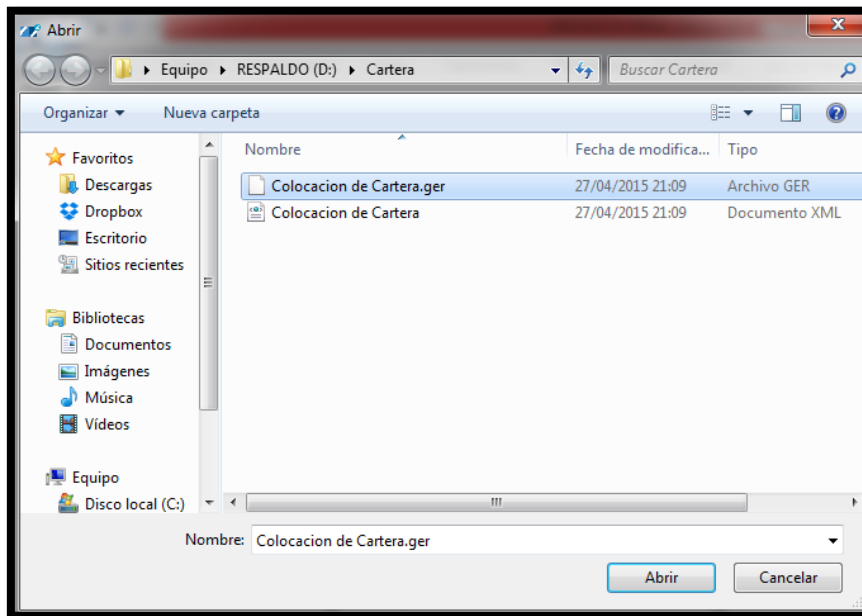


Figura 4.73. Abrir Diseño

Adicionalmente se realizaron pruebas con Excel para probar la conexión de los Cubos OLAP y su funcionalidad.

- Cartera

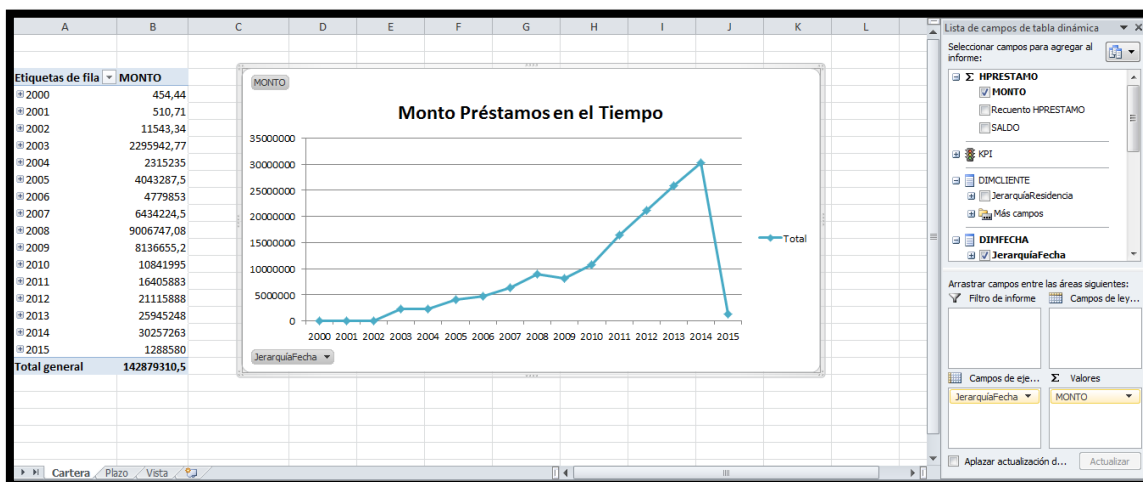


Figura 4.74. Diseño Excel Cartera

- Plazo

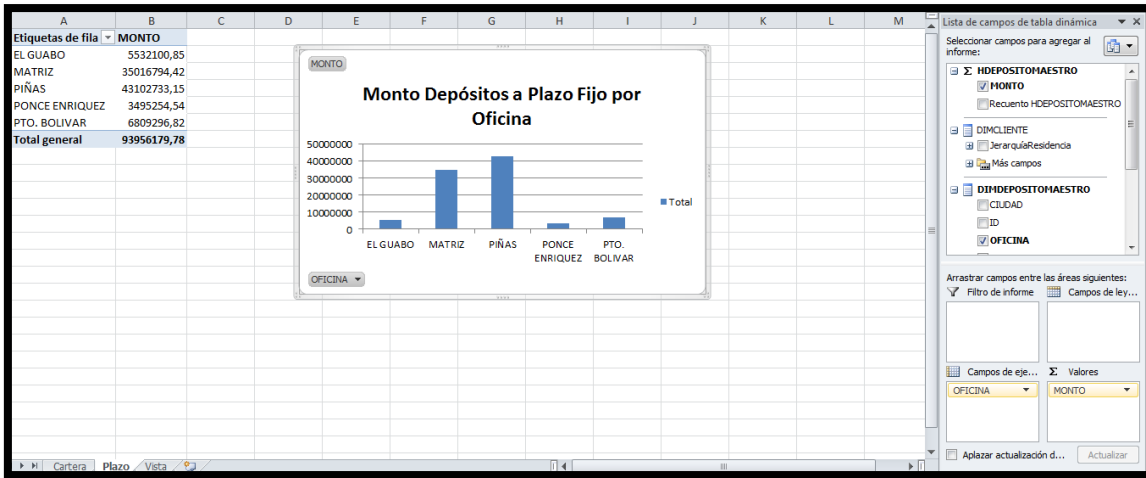


Figura 4.75. Diseño Excel Plazo

- Vista

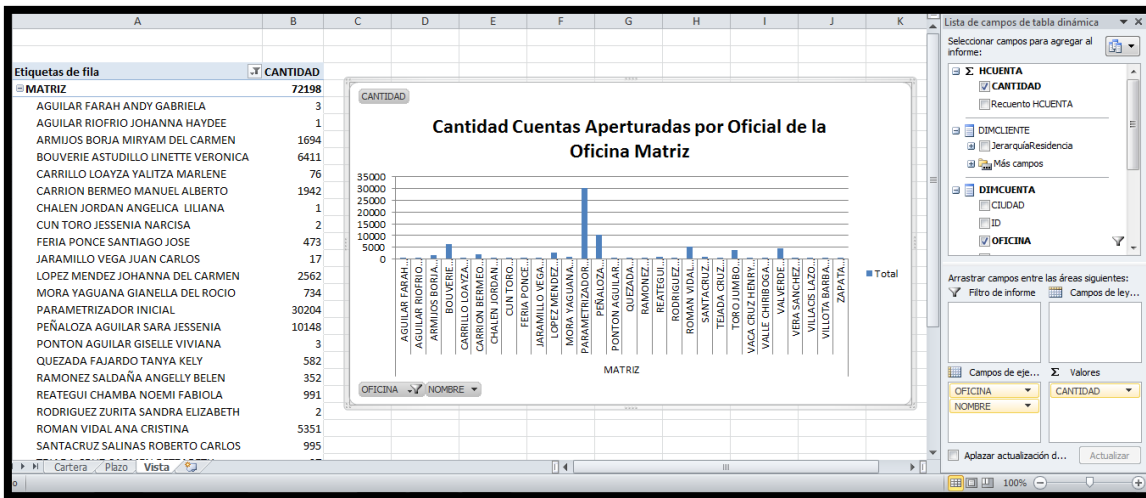


Figura 4.76. Diseño Excel Vista

CAPITULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Los componentes DevExpress escogidos permitieron una conexión directa con el Motor Analítico en Línea de Sql Server y por ende con el cubo multidimensional creado en Analysis Services, así la carga inicial de datos en el front end es óptima y con tiempos mínimos de respuesta.
- El diseño utilizado brinda opciones de visualizar información tabulada, con cuantificadores reales en una tabla pivot y gráficamente con cuadros estadísticos que en conjunto proveen un adecuado análisis gerencial.
- La utilización de jobs permitió realizar las tareas ETL de manera automática y con un monitoreo diario que servirá como contingencia en caso de fallo del proceso.
- La utilización de componentes de windows permitió implementar directamente el módulo BI en el Sistema Financiamiento sin necesidad de aplicaciones externas, brindando seguridad y confiabilidad en los usuarios.

5.2. Recomendaciones

- Para el desarrollo de una Interfaz Business Intelligence de Windows se recomienda utilizar componentes DevExpress ya que permiten una consulta dinámica de los datos y una conexión directa con el motor OLAP de Sql Server.
- Se recomienda se realice un plan de soporte técnico diario del módulo, designando una persona responsable que verifique que los servicios Sql Server se encuentren iniciados y que se realice el proceso de actualización de datos correctamente.

- Es recomendable permitir el acceso al módulo, únicamente a los usuarios que les corresponda realizar un análisis gerencial de la empresa.
- Se recomienda que los modelos de análisis se los guarde en un lugar adecuado y con un nombre descriptivo que denote el tipo de análisis realizado.
- Es aconsejable la difusión del nuevo módulo indicando sus ventajas a todos los clientes de la empresa Sifizsoft S.A. e incluirlo en el sistema base para nuevas presentaciones del mismo.

Bibliografía o referencias

[1]Superintendencia de Bancos y Seguros del Ecuador, http://www.sbs.gob.ec/medios/PORTALDOCS/downloads/articulos_financieros/Estudios%20Tecnicos/2012/AT32_2012.pdf [Internet], 2013.

[2]Superintendencia de Economía Popular y Solidaria, " Listado de Organizaciones Supervisadas por la SEPS", <http://www.seps.gob.ec/web/guest/consulta-roeps> [Internet], 2015.

[3] Ing. Carlos Israel Núñez Miranda autor del proyecto "Business Intelligence para manejo y recuperación de cartera en la empresa FERRIARCOS", http://repo.uta.edu.ec/bitstream/handle/123456789/2355/Tesis_t709mbd.pdf?sequence=1 [Internet], 2012.

[4] Ing. Neris Adalberto Falcón Rodríguez “Desarrollo de una solución de Inteligencia de Negocios en el manejo de estadísticas de control en la venta de repuestos de la empresa Talleres Ambamazda S.A. de la ciudad de Ambato”, http://repo.uta.edu.ec/bitstream/handle/123456789/3008/Tesis_t781mbd.pdf?sequence=1 [Internet], 2012.

[5] Ormaza Bailón Antonia Monserrate y Robles Quintero Cesar Augusto “Análisis, Diseño e Implementación de una Herramienta Informática para la Toma de Decisión en el departamento de Crédito empleando metodologías de Business Intelligence en la cooperativa de ahorro y crédito 15 de Abril de la ciudad de”, <http://repositorio.utm.edu.ec/bitstream/123456789/5107/1/TESIS%20DE%20INGENIERIA%20DE%20SISTEMAS%20BI.pdf> [Internet], 2014.

[6] Boada Byron y TituañaAlvaro “Desarrollo de una aplicación de business intelligence (BI) para la empresa Empaqplast”, <http://repositorio.espe.edu.ec/bitstream/21000/5819/1/T-ESPE-034058.pdf> [Internet], 2014.

[7] Josep Curto Díaz, INTRODUCCION AL BUSINESS INTELLIGENCE, Santiago de Cuba, 2010.

[8] Ing. Bernabeu Ricardo Dario, DATA WAREHOUSING: Investigación y Sistematización de Conceptos – HEFESTO: Metodología propia para la Construcción de unDataWarehouse, Córdoba-Argentina, 2009.

[9] Ximena Lozada Peñafiel, Holger Cruz Tamayo, Washington Pérez Argudo, Andrés de la Torre Díaz, “ANÁLISIS, DISEÑO, CONSTRUCCIÓN E IMPLEMENTACIÓN DE UN DATA WAREHOUSE PARA TOMA DE DECISIONES Y CONSTRUCCIÓN DE LOS KPI, PARA LA EMPRESA KRONOSCONSULTING CIA LTDA”, [Internet], <http://repositorio.espe.edu.ec/bitstream/21000/7942/1/AC-SI-ESPE-047750.pdf>, 2013

[10] Alfredo Martínez Orol, “OLAP y el Diseño de cubos”, <http://www.gestiopolis.com/canales8/ger/olap-online-analytic-processing.htm> [Internet], 2007.

[11] Ing. Juan Carlos Trujillo, Diseño y Explotación de Almacén de Base de Datos, San Vicente (Alicante), 2010.

[12] TL-Tools.Info,”Business Intelligence - Almacenes de Datos - ETL”, http://etl-tools.info/es/bi/proceso_etl.htm [Internet], 2014.

[13] IBM, Esquemas Dimensionales, http://www-01.ibm.com/support/knowledgecenter/SS9UM9_7.5.3/com.ibm.datatools.dimensionai.ui.doc/topics/c_dm_dimschemas.html [Internet], 2013.

[14] Zumo de Marketing, “Key Performance Indicator (KPI)”, <http://zumodemarketing.com/que-es-el-key-performance-indicator-kpi/> [Internet], 2014.

[15] Banco Central del Ecuador, "Metodología de la Información Estadística Mensual", <http://contenido.bce.fin.ec/documentos/PublicacionesNotas/Catalogo/IEMensual/metodologia/METODOLOGIA3RAed.pdf> [Internet], 2011.

[16] Microsoft, "SQL Server Analysis Services", [http://technet.microsoft.com/es-es/library/ms175609\(v=sql.90\).aspx](http://technet.microsoft.com/es-es/library/ms175609(v=sql.90).aspx) [Internet], 2012.

[17] Dawcons Enterprise solution, "DevExpress, el mejor aliado en el desarrollo", <http://dawconsblog.blogspot.com/2014/04/devexpress-el-mejor-aliado-en-el.html> [Internet], 2013.

[18] Sifizsoft, "Financial", <http://www.sifizsoft.com/financial-business-system-v2-0> [Internet], 2015.

[19] DATAPRIX, "Business Intelligence Solutions", <http://www.dataprix.net/en/blogs/respinosamilla/business-intelligence-solution>. [Internet], 2015.

Anexos

Anexo 1

Entrevista

Objetivo de la Entrevista: Descubrir la importancia de implementar un módulo de Business Intelligence en el Sistema Financiero Financial Business System e identificar las áreas más importantes para aplicar dicho módulo; además de establecer un panorama que indicará la naturaleza de las consultas gerenciales.

Entrevistado: Ing. Santiago Álvarez Gerente General de la empresa Sifizsoft S.A.

1.- ¿Tiene conocimiento usted sobre Business Intelligence?

SI

NO

2.- ¿El sistema financiero Financial Business System cuenta con un módulo de inteligencia de negocios?

SI

NO

3.- ¿Cree usted que un módulo de business intelligence beneficiaría a la consulta gerencial en el sistema Financial Business System?

SI

NO

4.- ¿Considera usted que las consultas y reportes que presenta el sistema actualmente es suficiente para un análisis dinámico detallado para la gerencia?

SI

NO

Porque: Porque son reportes estáticos que no permiten navegar entre los datos y para cada perspectiva de análisis será necesario un nuevo requerimiento para realizar un nuevo reporte.

5.- De los siguientes módulos cuales considera los más importante para que sean sujeto de un análisis gerencial.

Cartera	<input type="checkbox"/>	Captaciones Plazo	<input type="checkbox"/>
Captaciones Vista	<input checked="" type="checkbox"/>	Tesorería	<input type="checkbox"/>
Obligaciones Financieras	<input type="checkbox"/>	Lavado de Activos	<input type="checkbox"/>
Activos Fijos	<input type="checkbox"/>	Nómina	<input type="checkbox"/>
Contabilidad	<input type="checkbox"/>	Seguridad	<input type="checkbox"/>

6.- Describa para cada módulo escogido en la pregunta anterior la perspectiva del análisis que se requeriría para un análisis financiero gerencial.

Cartera.- Para cartera se necesita la colocación de los préstamos indicando el monto y la cantidad, será necesario conocer el tipo de cliente la ubicación geográfica, la oficina el usuario, del préstamo será necesario conocer el tipo de préstamo, la calificación contable y el destino financiero.

Captaciones Vista.- Lo importante es conocer el número de cuentas aperturadas, el tipo, la oficina y el usuario que realizó las captaciones así como el tipo de cliente y la ubicación geográfica.

Captaciones Plazo.- Para este módulo se requiere los mismos datos para el cliente que los módulos anteriores al igual que el usuario y conocer el tipo de depósitos y la oficina donde se realizó la inversión.

Anexo 2

Encuesta Módulo de Business Intelligence

El objetivo de la presente encuesta es conocer la necesidad de un módulo de business intelligence en una entidad financiera.

*Obligatorio

Existencia *

¿Su empresa cuenta con un módulo de business intelligence?

- SI
 NO

Módulos Necesarios *

De los siguientes módulos de negocio escoja en cuáles cree ud debería enfocarse un análisis de business intelligence.

- Captaciones a la Vista
 Captaciones a Plazo
 Crédito
 Cartera
 Cobranza
 Contabilidad
 Cajas
 Reportes a Organismos de Control

Indique el nombre de la entidad financiera a la que pertenece. *

Resultados

Existencia	Módulos Necesarios	Entidad Financiera	Provincia
NO	Captaciones a la Vista, Captaciones a Plazo, Cartera, Reportes a Organismos de Control	Firesa	Guayas
SI	Captaciones a la Vista, Captaciones a Plazo, Cartera	El Sagrario	Tungurahua
NO	Captaciones a Plazo, Crédito, Cartera, Reportes a Organismos de Control	Cooperativa Politécnica Ltda.	Pichincha
NO	Captaciones a la Vista, Captaciones a Plazo, Cartera	Cooperativa La Floresta	Tungurahua
SI	Captaciones a Plazo, Crédito, Cartera, Contabilidad	VFE	Pichincha
NO	Crédito, Cartera, Contabilidad	Cooperativa de Ahorro y Credit Tena Ltda	Napo
NO	Captaciones a Plazo, Crédito	CACPEY	Loja
NO	Captaciones a la Vista, Captaciones a Plazo,	CACPET	Tungurahua

	Crédito, Cartera, Contabilidad		
SI	Captaciones a la Vista, Captaciones a Plazo, Crédito, Cartera	Mutualista Ambato	Tungurahua
NO	Captaciones a Plazo, Crédito, Cartera, Cobranza	Cooperativa Acción Imbaburapak Ltda.	Imbabura
NO	Captaciones a la Vista, Captaciones a Plazo, Crédito, Cartera	Lucha Campesina	Guayas
NO	Crédito, Cartera, Contabilidad, Cajas	INSOTEC	Pichincha
NO	Captaciones a la Vista, Captaciones a Plazo, Cartera, Contabilidad	Coac San Alfonso	Tungurahua
NO	Captaciones a Plazo, Cartera	Fundación Alternativa	Pichincha
NO	Captaciones a la Vista, Crédito, Cartera, Contabilidad	FUNED VisionFund OPDF	Guayas
SI	Captaciones a la Vista, Captaciones a Plazo, Cartera, Contabilidad, Cajas, Reportes a Organismos de Control	FAMA OPDF	Pichincha
NO	Captaciones a la Vista, Crédito, Cartera	CACPECO	Cotopaxi
NO	Cartera, Cobranza, Contabilidad	Cooperativa Fortuna	Loja
NO	Captaciones a la Vista, Captaciones a Plazo, Crédito, Cartera, Cobranza	Cooperativa Pablo Muñoz Vega	Tulcán
NO	Captaciones a Plazo, Crédito, Cartera, Cobranza	Cooperativa Acción Imbaburapak Ltda.	Imbabura
SI	Cartera, Cobranza, Contabilidad, Reportes a Organismos de Control	PILARH OPDF	Guayas
NO	Cartera	kullki wasi	Tungurahua
NO	Captaciones a la Vista, Captaciones a Plazo, Crédito	CACPECO	Cotopaxi
NO	Cartera, Cobranza, Contabilidad	EL SAGRARIO	Tungurahua
NO	Cartera, Cobranza, Contabilidad	SACAMBATO	Tungurahua
NO	Captaciones a la Vista, Captaciones a Plazo, Cartera, Contabilidad	San Francisco	Tungurahua
NO	Captaciones a la Vista, Captaciones a Plazo, Crédito, Cartera	Kullki Wasi	Tungurahua