



UNIVERSIDAD TÉCNICA DE AMBATO

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

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

TEMA:

**MÓDULO DE PRESUPUESTO FINANCIERO APLICADO A LA PROYECCIÓN
DE CRECIMIENTO EN MATRIZ Y OFICINAS OPERATIVAS DE LA
COOPERATIVA DE AHORRO Y CRÉDITO SAN FRANCISCO LTDA.**

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

LÍNEA DE INVESTIGACIÓN: Aplicaciones distribuidas

AUTOR: Pamela Solange Méndez Pinto

TUTOR: Ing. Dennis Vinicio Chicaiza Castillo, Mg

Ambato – Ecuador

Octubre – 2018

CERTIFICACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el Tema:

“MÓDULO DE PRESUPUESTO FINANCIERO APLICADO A LA PROYECCIÓN DE CRECIMIENTO EN MATRIZ Y OFICINAS OPERATIVAS DE LA COOPERATIVA DE AHORRO Y CRÉDITO SAN FRANCISCO LTDA”, de la señorita, PAMELA SOLANGE MÉNDEZ PINTO 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, enero 09 del 2019

EL TUTOR



Ing. Dennis Vinicio Chicaiza Castillo, Mg

AUTORÍA DEL TRABAJO

El presente trabajo de investigación titulado: MÓDULO DE PRESUPUESTO FINANCIERO APLICADO A LA PROYECCIÓN DE CRECIMIENTO EN MATRIZ Y OFICINAS OPERATIVAS DE LA COOPERATIVA DE AHORRO Y CRÉDITO SAN FRANCISCO LTDA. 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, enero 09 del 2019



Pamela Solange Méndez Pinto

CC: 1804570594

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, enero 09 del 2019

A handwritten signature in blue ink, appearing to read 'Pamela Solange Méndez Pinto', is written over a solid black horizontal line.

Pamela Solange Méndez Pinto

CC: 1804570594

APROBACIÓN DEL TRIBUNAL DE GRADO

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. David Guevara e Ing. Carlos Núñez, revisó y aprobó el Informe Final del trabajo de graduación titulado “MÓDULO DE PRESUPUESTO FINANCIERO APLICADO A LA PROYECCIÓN DE CRECIMIENTO EN MATRIZ Y OFICINAS OPERATIVAS DE LA COOPERATIVA DE AHORRO Y CRÉDITO SAN FRANCISCO LTDA”, presentado por la señorita Pamela Solange Méndez Pinto 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. Elsa Pilar Urrutia Urrutia

PRESIDENTE DEL TRIBUNAL



Ing. Mg. David Guevara

DOCENTE CALIFICADOR



Ing. Mg. Carlos Núñez

DOCENTE CALIFICADOR

DEDICATORIA

*El presente trabajo se lo dedico a mi madre Patricia,
quien gracias a su ejemplo, enseñanzas, valores y amor ha guiado
mis pasos para hacer de mí una mujer de bien.*

*A mi hermano David, que a pesar de su corta edad
siempre me ha enseñado fortaleza ante las adversidades.*

*A mi abuelito Telmo, que desde el cielo
está cuidándome y su recuerdo me ha dado el coraje
para superar cada obstáculo presentado.*

*A Christian que con su paciencia y amor ha sabido apoyarme
en cada decisión tomada y por ser parte esencial en mi vida.*

Pamela Solange Méndez Pinto

AGRADECIMIENTO

*Quiero agradecer a Dios por darme su bendición cada día,
por ser mi fortaleza en el camino recorrido y brindarme la sabiduría para
formarme profesionalmente, permitiéndome así, el cumplimiento
de una de mis metas más anheladas.*

*También quiero extender un sincero agradecimiento
a la Facultad de Ingeniería en Sistemas, Electrónica e Industrial
de la Universidad Técnica de Ambato, por abrirme sus puertas
hacia mi formación académica, por los conocimientos, experiencias
y vivencias brindadas, que han formado mi criterio
como persona y profesional.*

Pamela Solange Méndez Pinto

ÍNDICE GENERAL

PORTADA.....	i
CERTIFICACIÓN DEL TUTOR	ii
AUTORÍA DEL TRABAJO	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
RESUMEN EJECUTIVO	xiii
ABSTRACT.....	xiv
GLOSARIO DE TÉRMINOS Y ACRÓNIMOS.....	xv
INTRODUCCIÓN	xvii
CAPÍTULO I.....	1
EL PROBLEMA	1
1.1. Tema de Investigación.....	1
1.2. Planteamiento del Problema	1
1.3. Delimitación	3
1.3.1. De Contenidos.....	3
1.3.2. Temporal	3
1.4. Justificación.....	3
1.5. Objetivos	4
1.5.1. General	4
1.5.2. Específicos	4
CAPÍTULO II	5
MARCO TEÓRICO.....	5
2.1. Antecedentes Investigativos	5
2.2. Fundamentación Teórica	6
2.2.1. Presupuesto Financiero	6
2.2.2. Metodología SCRUM	7
2.2.3. Comparación de SCRUM con otras metodologías ágiles	8
2.2.4. Entity Framework.....	9
2.2.5. Patrón Modelo-Vista-Controlador	9

2.2.6.	Windows Communication Foundation	10
2.2.7.	Elementos de un servicio en WCF	11
2.2.8.	Smart Client	12
2.2.9.	DevExpress	13
2.2.10.	DevExpress WinForms	13
2.2.11.	Financial Business System v2.0.....	14
2.2.12.	Arquitectura N Capas	15
2.2.13.	Microsoft SQL Server	15
2.2.14.	Plan de Cuentas	15
2.2.15.	Proyecciones	16
2.3.	Propuesta de Solución	25
CAPÍTULO III		26
METODOLOGÍA		26
3.1.	Modalidad de Investigación	26
3.1.1.	Investigación de campo.....	26
3.1.2.	Investigación bibliográfica o documentada	26
3.2.	Población y Muestra	27
3.3.	Recolección de Información.....	27
3.4.	Procesamiento y análisis de datos	27
3.5.	Desarrollo del proyecto	27
CAPÍTULO IV		29
DESARROLLO DE LA PROPUESTA.....		29
4.1.	Análisis de Requerimientos.....	29
4.1.1.	Identificación de necesidades.....	29
4.1.2.	Funcionalidades.....	40
4.2.	Diseño y Arquitectura	47
4.2.1.	Elaboración de diagrama de clases	47
4.2.2.	Elaboración de diagrama de secuencias.....	48
4.2.3.	Elaboración de diagrama de estado.....	52
4.2.4.	Diseño de base de datos	52
4.2.5.	Diagrama de despliegue	55
4.2.6.	Diseño de interfaces	56
4.3.	Programación.....	60
4.3.1.	Desarrollo del sistema.....	60

4.4.	Pruebas	73
4.4.1.	Pruebas de desarrollo	73
4.4.2.	Pruebas de validación.....	74
4.4.3.	Pruebas de aceptación	76
4.5.	Puesta en Producción.....	78
CAPÍTULO V		80
CONCLUSIONES Y RECOMENDACIONES.....		80
5.1.	Conclusiones	80
5.2.	Recomendaciones	81
REFERENCIAS.....		82
ANEXOS		84

ÍNDICE DE TABLAS

Tabla 2.1: Comparativa de metodologías ágiles	8
Tabla 2.2: Búsqueda de coeficiente de correlación.....	22
Tabla 2.3: Resultados – Coeficiente de correlación.....	23
Tabla 2.4: Aplicación del método detallado	24
Tabla 4.5 Especificación Parametrización	41
Tabla 4.6 Especificación Selección de cuentas.....	42
Tabla 4.7 Especificación Porcentajes de distribución.....	43
Tabla 4.8 Especificación Proyección	44
Tabla 4.9 Especificación Visualización de saldos contables	44
Tabla 4.10 Especificación Mayorización.....	45
Tabla 4.11 Especificación Presentación de balances	46
Tabla 4.12 Especificación Comparativa de saldos contables.....	46
Tabla 4.13 PERIODOPRESUPUESTARIO	52
Tabla 4.14 PERIODOPRESUPUESTARIOCTACONTABLE	52
Tabla 4.15 PERIODOPRESUPUESTARIODISTRIBUCION.....	53
Tabla 4.16 CRECIMIENTOPRESUPUESTOANUAL.....	53
Tabla 4.17 DETALLECRECIMIENTOPRESUPUESTOANUAL.....	53
Tabla 4.18 PLANCUENTACONTABLEMAYORIZAR	54
Tabla 4.19: Resultado Pruebas de Aceptación.....	76
Tabla 4.20: Comparativa de resultados obtenidos	77
Tabla 4.21: Puesta en Producción Funcionalidades.....	78
Tabla 4.22: Puesta en producción Menú	78
Tabla 4.23: Puesta en producción Reporte Comparativa.....	79
Tabla 2.24: Puesta en producción Permisos.....	79

ÍNDICE DE GRÁFICOS

Fig. 1 Metodología SCRUM.....	7
Fig. 2 Patrón de arquitectura de software, Modelo-Vista-Controlador.....	10
Fig. 3 Arquitectura Smart Client.....	13
Fig. 4 Arquitectura del sistema financiero	14
Fig. 5 Arquitectura N Capas.....	15
Fig. 6 Criterio de mínimos cuadrados	16
Fig. 7 Dispersión de datos - búsqueda de coeficiente de correlación	23
Fig. 8 Diagrama de actividades	30
Fig. 9 Diagrama de casos de uso.....	41
Fig. 10 Diagrama de clases: modelo conceptual.....	48
Fig. 11 Diagrama de clases - Módulo de presupuesto	48
Fig. 12 Secuencia Parametrización	49
Fig. 13 Secuencia Selección de Cuentas	49
Fig. 14 Secuencia Porcentajes de Distribución	50
Fig. 15 Secuencia Proyecciones	50
Fig. 16 Secuencia Visualización de saldos contables	51
Fig. 17 Secuencia Mayorización	51
Fig. 18 Secuencia Presentación de balances	51
Fig. 19 Secuencia Comparativa de saldos contables.....	51
Fig. 20 Diagrama de estado - Parametrización	52
Fig. 21 Modelo relacional - Módulo de presupuesto	54
Fig. 22 Diagrama de Arquitectura – Módulo de presupuesto	55
Fig. 23 Diseño de Interfaces – Parametrización.....	56
Fig. 24 Diseño de interfaces - Selección de cuentas	56
Fig. 25 Diseño de interfaces - Porcentajes de distribución	57
Fig. 26 Diseño de interfaces – Proyección de valores	57
Fig. 27 Diseño de interfaces - Mayorización de cuentas	58
Fig. 28 Diseño de interfaces - Balance General.....	58
Fig. 29 Diseño de interfaces - Estado de Pérdidas y Ganancias	59
Fig. 30 Diseño de interfaces - Comparativa de saldos.....	59
Fig. 31 Prueba de validación Valores Numéricos.....	75
Fig. 32 Prueba de Validación Valores adecuados.....	75
Fig. 33 Prueba de Validación Datos Reales	75
Fig. 34 Prueba de Validación Presentación de Mensajes.....	76

RESUMEN EJECUTIVO

La cooperativa de ahorro y crédito “San Francisco LTDA.”, se encuentra en constante crecimiento, tanto en el número de personas que se asocian a ella, como en el número de sucursales operativas ubicadas en todo el Ecuador, requiriendo así, personal capacitado y recursos óptimos para brindar un servicio eficientemente.

El creciente desarrollo de tecnología ha causado que esta sea indispensable en acciones y procesos que lleven a cabo entidades financieras, por tal razón la cooperativa “San Francisco LTDA.” tiene la necesidad de sistematizar sus procesos mediante soluciones informáticas que le permitan ahorrar recursos, tanto de tiempo como de esfuerzo.

El sistema financiero que utiliza la entidad se encuentra bajo constantes modificaciones realizadas por el departamento de tecnología de la información, área de desarrollo de software, el mismo carece de una funcionalidad que permita llevar a cabo el respectivo proceso para el cálculo de presupuesto que cada año se calcula.

El principal inconveniente que poseen los directivos de la entidad al no contar con este proceso sistematizado es el retraso en la toma de decisiones para el logro de los objetivos, ya que, dependiendo de los montos presupuestados, se puede saber el nivel de factibilidad de abrir nuevas sucursales o contratar más personal.

Por medio del presente documento se propone la implantación de un módulo de presupuesto financiero para el cálculo anual del mismo, el cual estará adaptado al entorno de desarrollo utilizado por el área para evitar inconvenientes de adaptabilidad y mantener un estándar de programación.

A través de la información presentada por el módulo, se busca brindar a los directivos de la entidad una herramienta efectiva para el manejo de proyecciones de cuentas contables que permita controlar en periodos de tiempo el nivel de cumplimiento propuesto que se les ha asignado a sus empleados.

ABSTRACT

The credit union cooperative "San Francisco LTDA.", is constantly growing, both in the number of people associated with it, as in the number of operational branches located throughout Ecuador, requiring trained personnel and optimal resources to provide a service efficiently.

The growing development of technology has caused it to be indispensable in actions and processes carried out by financial entities, for this reason the cooperative "San Francisco LTDA." Has the need to systematize its processes through computer solutions that allow it to save resources, both of time as of effort.

The financial system used by the entity is under constant changes made by the information technology department, software development area, it lacks functionality that allows carrying out the respective process for calculating the budget that each year is calculated.

The main drawback that the directors of the entity have in not having this systematized process is the delay in making decisions for the achievement of the objectives, since, depending on the budgeted amounts, the level of feasibility of opening can be known new branches or hire more staff.

Through this document we propose the implementation of a financial budget module for the annual calculation thereof, which will be adapted to the development environment used by the area to avoid adaptability problems and maintain a programming standard.

Through the information presented by the module, it seeks to provide the executives of the entity with an effective tool for the management of accounting account projections that allows to control in time periods the level of proposed compliance that has been assigned to its employees.

GLOSARIO DE TÉRMINOS Y ACRÓNIMOS

CMM Capability Maturity Model: El Modelo de Madurez de Capacidades es una herramienta con el cual se evalúan los procesos presentes en una organización.

ASD Adaptive Software Development: Desarrollo adaptativo de software es una metodología que se orienta a la implementación de patrones para llevar a cabo un desarrollo de software.

DSDM Dynamic Systems Development Method: El método de desarrollo de sistemas dinámicos provee un marco de referencia para el desarrollo e implementación de software, se basa en iteraciones que le permiten adaptarse a requerimientos cambiantes tanto en recursos, tiempo y presupuesto.

FDD Feature-Driven Development: El desarrollo basado en funcionalidades es una variante del desarrollo ágil, pero orienta todas sus fases a la usabilidad que el usuario final necesita del sistema.

LD Lean Development: Es una traducción de principios y prácticas de tal manera que puedan ser un apoyo para el área de desarrollo de software al momento de construir un producto de software.

XP eXtreme Programming: La programación extrema se destaca por sus procesos ágiles en el desarrollo de software poniendo énfasis en la adaptabilidad, permitiendo nuevas funcionalidades agregadas en un futuro.

Rich Clients: Hace referencia a un equipo conectado en red que tiene varios recursos instalados en forma local pero también consume en su minoría recursos de otros equipos distribuidos a través de la red.

Thin Clients: Un cliente ligero trabaja dentro la arquitectura de red cliente-servidor, depende en gran medida de los recursos distribuidos en la red ya que su arquitectura no permite almacenarlos localmente.

IMF's: Las Instituciones de Microfinanzas son entidades que tienen como objetivo la introducción de personas independientes y microempresarios de escasos recursos al campo laboral otorgando microcréditos y capacitaciones.

UI (User Interface): La interfaz de usuario es el medio por el cual un usuario puede comunicarse con cualquier tipo de dispositivo electrónico.

Object Management Group: Es un consorcio que se dedica al cuidado y establecimiento de diversos estándares de tecnologías orientadas a objetos, tales como UML, XMI, CORBA y BPMN.

INTRODUCCIÓN

El presente proyecto de investigación cuyo tema es, “MÓDULO DE PRESUPUESTO FINANCIERO APLICADO A LA PROYECCIÓN DE CRECIMIENTO EN MATRIZ Y OFICINAS OPERATIVAS DE LA COOPERATIVA DE AHORRO Y CRÉDITO SAN FRANCISCO LTDA.”, consta de cinco capítulos que se detallan a continuación:

Capítulo I. “El problema”, identifica las necesidades de la entidad dentro de un contexto real para que pueda ser planteado en conformidad con la situación actual, definiendo alcance, justificación y objetivos a cumplir que servirán como guía en el transcurso del proyecto.

Capítulo II. “Marco Teórico”, contiene la recopilación de información teórica necesaria para comprender el problema y poder así sugerir una posible solución.

Capítulo III. “Metodología”, describe las modalidades aplicadas de investigación, el entorno con el cual se va a trabajar y una breve descripción de la manera en cómo estará desarrollado el proyecto.

Capítulo IV. “Desarrollo de la Propuesta”, consta de las actividades llevadas a cabo para el desarrollo e implantación de la propuesta de solución como definición de requerimientos, casos de uso del manejo del módulo, diagramas de secuencia de procesos, diseño de base de datos, diseño de arquitectura, diseño de interfaz gráfica al usuario y pruebas de funcionamiento.

Capítulo V. “Conclusiones y Recomendaciones”, establece las conclusiones que el investigador puede sacar del procedimiento que ha llevado a cabo, las recomendaciones para futuros investigadores que deseen aportar a la funcionalidad del módulo.

Finalmente, se incluye las referencias consultadas y se anexa entrevista, manual de usuario y los documentos técnicos requeridos.

CAPÍTULO I

EL PROBLEMA

1.1. Tema de Investigación

Módulo de Presupuesto Financiero Aplicado a la Proyección de Crecimiento en Matriz y Oficinas Operativas de la Cooperativa De Ahorro y Crédito San Francisco Ltda.

1.2. Planteamiento del Problema

El presupuesto de una organización expresa la forma en que se van a aplicar los recursos disponibles en el futuro para conseguir los objetivos fijados en la estrategia. Por tanto, el presupuesto posee unas características esenciales que lo convierten en una herramienta esencial para la gestión de cualquier organización [1].

En el área de las finanzas, el presupuesto financiero es una de las herramientas que busca incrementar el patrimonio de los accionistas y de la empresa. Si bien el proceso administrativo cuenta con diversas etapas, el presupuesto viene a empatar los objetivos señalados en la planeación del negocio con los acontecimientos que se van llevando a cabo [2].

Un presupuesto adecuadamente planificado es la base para el desarrollo de todas las actividades que se han establecido realizar para la eficiente ejecución presupuestaria, por otra parte siendo la evaluación presupuestaria un factor determinante en el proceso presupuestario, es necesario establecer saldos para poder determinar la ejecución de cada actividad, ya que es susceptible a cambios o reajustes dentro del presupuesto y con ello tomar acciones correctivas en los casos que sean necesarios [3].

El presupuesto está adaptado a los planes de cada una de las circunscripciones territoriales, garantizando la equidad de los recursos, todo lo que se financie con estos recursos, sean programas o proyectos, estarán sujetos a evaluación al final de su vencimiento [4].

La gestión financiera tiene como esencia la adecuada realización del presupuesto de gastos, la misma que realiza mediante las diferentes fases, contando inicialmente con una autorización del gasto y concluyendo el desembolso. La gestión financiera es aquella que convierte a la misión y visión en operaciones monetarias [5].

Si la metodología para confeccionar el presupuesto es la adecuada y ante todo es útil, sencilla y práctica, el sistema será siempre actualizado, mejorado y motivador y suscitará la participación de toda la organización. No olvidemos que la finalidad última de un buen sistema presupuestario no es otra que mejorar los resultados de su organización, facilitar su crecimiento y su supervivencia en el tiempo [1].

En la provincia de Tungurahua se puede reflejar el crecimiento exponencial que han tenido las cooperativas de ahorro y crédito en los últimos años ya en el último informe emitido por la Superintendencia de Economía Popular y Solidaria al año 2015 se registraron un total de 161 instituciones financieras activas bajo su supervisión [6], estos datos, para la Cooperativa de Ahorro y Crédito "San Francisco LTDA." representan una alta competitividad en el mercado al que se dirige, lo cual exige el uso de herramientas efectivas para la toma de decisiones.

SifizSoft S.A. es una empresa que desarrolla Software financiero contable escalable que se ajusta a las necesidades de las Instituciones Financieras [7], entidad creadora del producto comercial Financiamiento 2.0 que fue adquirido por la Cooperativa de Ahorro y Crédito "San Francisco LTDA.", en el año 2016, este software es el principal intermediario que conecta a los socios con los servicios que oferta la cooperativa, por lo tanto, con la compra de su código fuente, es posible agregar características adicionales propias del personal a cargo que permitan satisfacer las necesidades que vayan surgiendo en la entidad con el pasar del tiempo.

Una de las desventajas que presenta Financiamiento 2.0 es que carece de un módulo de presupuesto que le permita a la institución proyectar sus cuentas hacia el próximo año

y tomar mediante un control de cumplimiento las decisiones necesarias y oportunas que permitan seguir generando utilidades, esto hace que la entidad se encuentre en desventaja frente a sus competidores.

El plazo mínimo que rige para el presupuesto es de un ejercicio económico completo, en la Cooperativa de Ahorro y Crédito "San Francisco LTDA.", se toma octubre como fecha mes de inicio y como fecha fin, octubre del siguiente año, en este periodo se ejecuta la utilización de los valores proyectados que en la actualidad son calculados manualmente.

1.3. Delimitación

1.3.1. De Contenidos

Área Académica:

Software

Línea de Investigación:

Desarrollo de Software

Sublíneas de investigación:

Intercambio de Información

1.3.2. Temporal

La duración de la investigación se llevó a cabo en un tiempo de seis meses a partir de la aprobación del proyecto por parte del Honorable Consejo Directivo de la Facultad de Ingeniería en sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

1.4. Justificación

La importancia de esta investigación se fundamenta en que se aborda una temática que afecta al sector empresarial principalmente al entorno financiero. La presente investigación pretende ser un aporte significativo al análisis de balances actuales para otorgar un monto de presupuesto a diferentes actividades en pro de la empresa.

Al contar con un gran número de sucursales en la Cooperativa de Ahorro y Crédito San Francisco Ltda. la cantidad de información que aportan las mismas es alta al momento de establecer distintos escenarios para cada una y obtener así un presupuesto determinado, asignado para llevar a cabo ciertas tareas.

Los parámetros cambiantes y el uso de herramientas convencionales no aportan eficacia a las operaciones que se llevan a cabo para poder obtener un valor aproximado a la realidad que vive la empresa, por lo que se toman decisiones guiadas por la experiencia de quien lo dirige.

1.5. Objetivos

1.5.1. General

Implantar una herramienta informática que permita realizar el cálculo del presupuesto financiero en la Cooperativa de Ahorro y Crédito San Francisco Ltda.

1.5.2. Específicos

- Analizar la situación actual del proceso de cálculo presupuestario.
- Planificar iteraciones basadas en Metodología Scrum para el desarrollo del módulo de cálculo presupuestario.
- Validar el funcionamiento del módulo, comparando los resultados obtenidos del proceso de cálculo presupuestario contra valores reflejados en procesos de años anteriores.
- Desplegar el módulo de cálculo presupuestario en el ambiente de producción de la Cooperativa de Ahorro y Crédito San Francisco Ltda.

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes Investigativos

La presupuestación financiera es un método para especificar qué se debe hacer para lograr la debida implementación de las estrategias. No se debe considerar como un instrumento que sirve para limitar los egresos, sino más bien como un método que sirve para obtener el uso más rentable y productivo de los recursos de la organización. Los presupuestos financieros son la asignación planificada de los recursos de la empresa, basada en los pronósticos del futuro [8].

El aplicar un correcto cálculo del presupuesto financiero anual en instituciones, ha ayudado a las mismas a una correcta distribución de sus recursos monetarios y físicos.

Magdalena Lemus en su proyecto de investigación orientado a la devolución del impuesto al valor agregado y el presupuesto de la Universidad Técnica de Ambato, expresa que, de acuerdo con la investigación realizada en la Escuela Superior Politécnica del Litoral, se define que, “El presupuesto es básicamente la cuenta en que una institución registra el dinero que espera recaudar y gastar durante un año. Muestra como planea obtener recursos, en caso de que sus gastos sean mayores a sus ingresos”, por lo tanto es un medio para prever y decidir la producción que se va a realizar en un período determinado, así como para asignar formalmente los recursos que esa producción requiere en las actividades de una institución, sector o región siendo su objetivo principal el de comparar a tiempo entre lo presupuestado y los resultados obtenidos [9].

La Revista “Visión de Futuro” en su aporte “*Planeamiento Financiero: Su Importancia Y Contribución Para La Gestión De Las Empresas Cooperativas*”, manifiesta que el planeamiento financiero trae ventajas, como el conocimiento anticipado de todo el presupuesto de consumos e inversiones para el período siguiente, el poner en práctica una gestión administrativa, permite la identificación por parte de la empresa de necesidades como el aclaramiento o la posibilidad de expansión, la viabilidad de su presupuesto para el mercado actuante, la evaluación de posibilidad en la implementación de nuevos proyectos con sus respectivos costos, y la planificación de reservas para inversiones futuras [10].

En la publicación “*Gestión de presupuestos financieros*” realizada por la revista Ciencias Holguín se define que una asignación de recursos que no ha tenido un análisis previo trae como consecuencia que en ocasiones se destinen grandes cantidades de recursos para actividades que no lo requieren, mientras para otras los recursos asignados no son suficientes. El carácter no participativo de este proceso también incide en la poca fundamentación de los presupuestos elaborados y que no se conozcan las actividades que deben ser priorizadas [11].

La herramienta de presupuestación Quoters, es un software comercial que fue evaluado por la contadora general de la cooperativa, la desventaja principal presentada fue el ingreso de datos, la cantidad de información que requería para el cálculo debía extraerse mediante consultas directas a la base de datos y cargar esa información al software mediante un archivo de texto plano por cada mes que debía contener el valor de las cuentas contables de todos los años históricos posibles, adicional la proyección de cuentas podía realizarse hasta el cuarto nivel de subcuentas, limitando así los resultados esperados por lo cual no fue adquirido.

2.2. Fundamentación Teórica

2.2.1. Presupuesto Financiero

Su función principal es ayudar a que los recursos sean asignados de manera correcta, utilizando una proyección matemática calculada para que las entidades estén preparadas para asumir cambios y de esta manera no tener problemas de programación de recursos financieros [4].

2.2.2. Metodología SCRUM

Scrum es una metodología ágil que se basa “en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de la toma de decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo” [12].

Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración [13].

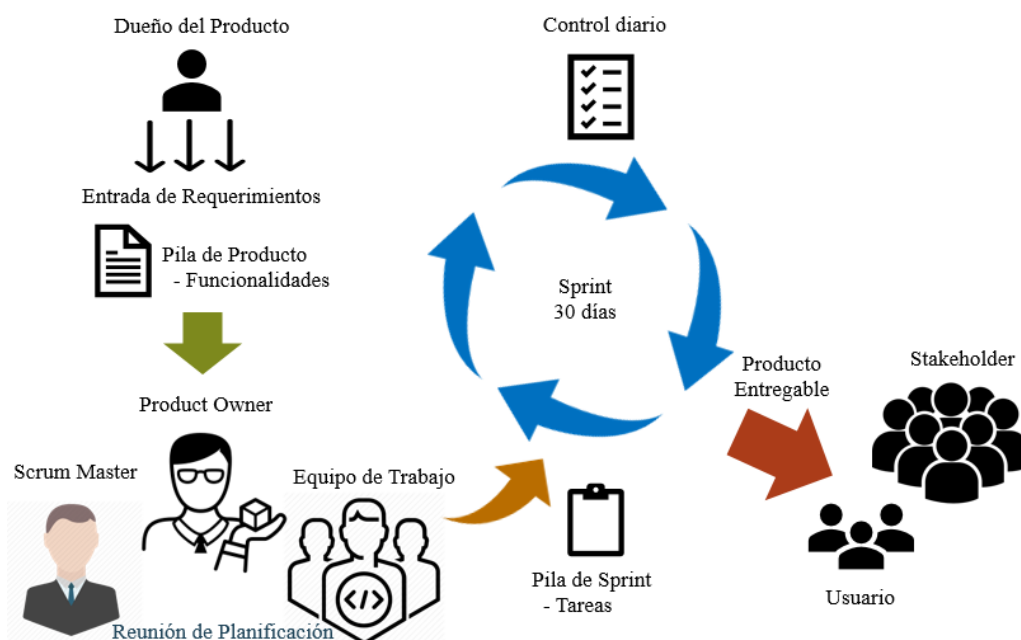


Fig. 1 Metodología SCRUM

En la cooperativa de ahorro y crédito San Francisco Ltda., los requerimientos son creados y analizados en conjunto por el jefe de departamento que solicita el desarrollo, el jefe de departamento de T.I. y el jefe de área de desarrollo, los cuales son planeados por nivel de prioridad, en seguida se procede a documentar las especificaciones

requeridas que serán asignadas a los programadores juntamente con una estimación de tiempo para la planeación del sprint.

Al concluir el tiempo de un sprint, el cual costa de 30 días, el jefe de desarrollo realiza una revisión de cumplimiento de especificaciones a cada uno de los programadores, evaluando así, si los tiempos han sido los adecuados o si durante el desarrollo se presentaron inconvenientes que retrasaron el cumplimiento de estos.

2.2.3. Comparación de SCRUM con otras metodologías ágiles

La siguiente tabla [14], compara las distintas aproximaciones ágiles en base a tres parámetros: vista del sistema como algo cambiante, tener en cuenta la colaboración entre los miembros del equipo y características más específicas de la propia metodología como son simplicidad, excelencia técnica, resultados, adaptabilidad, etc.

Tabla 2.1: Comparativa de metodologías ágiles

	CMM	ASD	Crystal	DSDM	FDD	LD	Scrum	XP
Sistema como algo cambiante	1	5	4	3	3	4	5	5
Colaboración	2	5	5	4	4	4	5	5
Resultados	2	5	5	4	4	4	5	5
Simplicidad	1	4	4	3	5	3	5	5
Adaptabilidad	2	5	5	3	3	4	4	3
Excelencia técnica	4	3	3	4	4	4	3	4
Prácticas de colaboración	2	5	5	4	3	3	4	5
Media CM	2.2	4.4	4.4	3.6	3.8	3.6	4.2	4.4
Media Total	1.7	4.8	4.5	3.6	3.6	3.9	4.7	4.8

Como se puede observar, todas las metodologías ágiles tienen una significativa diferencia en su índice de agilidad respecto a CMM y entre ellas destacan ASD, Scrum y XP como mejores puntuadas con respecto a los parámetros mencionados.

2.2.4. Entity Framework

Como su nombre lo indica, Entity Framework permite trabajar directamente con Entidades que representan su propio esquema sin la necesidad de utilizar los objetos DataReader y DataSet, Entity Framework es un conjunto de tecnologías ADO.NET que ayuda a llenar el espacio entre el desarrollo orientado a objetos y las bases de datos. Esta brecha se conoce comúnmente como "falta de concordancia" y existe porque el mapeo y la organización de las clases no coinciden del todo con la organización de los objetos relacionales. Muchos mapeos han intentado resolver este problema asignando clases y propiedades orientadas a objetos directamente a tablas y columnas, pero esta arquitectura permite que las aplicaciones se escriban en modelos conceptuales y no directamente en la base de datos. Al hacerlo, se ha cerrado la brecha entre la programación orientada a objetos y las bases de datos, lo que permite a los desarrolladores centrarse en la tarea de desarrollar aplicaciones, sin preocuparse por las bases de datos o acceso a datos [15].

2.2.5. Patrón Modelo-Vista-Controlador

El Modelo es el objeto que representa los datos del programa, maneja los datos y controla todas sus transformaciones, no tiene conocimiento específico de los Controladores o de las Vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el Modelo y sus Vistas, y notificar a las Vistas cuando cambia el Modelo.

La Vista es el objeto que maneja la presentación visual de los datos representados por el Modelo, genera una representación visual del Modelo y muestra los datos al usuario. Interactúa preferentemente con el Controlador, pero es posible que trate directamente con el Modelo a través de una referencia al propio Modelo.

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la Vista y el Modelo, cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del Modelo o por alteraciones de la Vista. Interactúa con el Modelo a través de una referencia al propio Modelo [16].

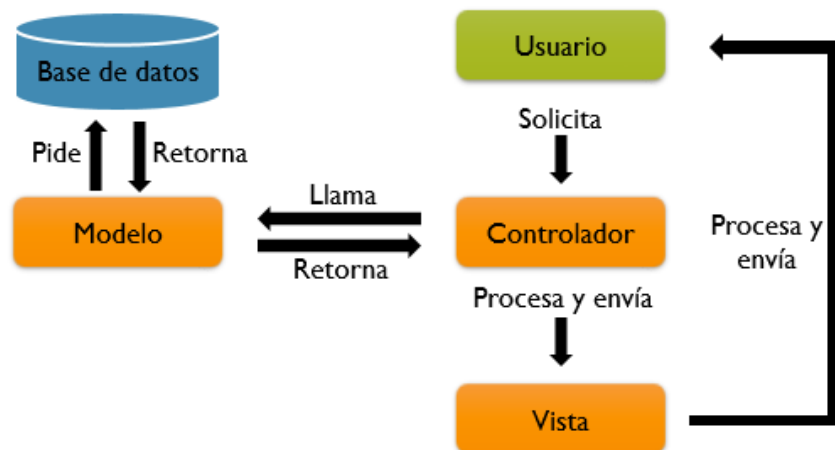


Fig. 2 Patrón de arquitectura de software, Modelo-Vista-Controlador

2.2.6. Windows Communication Foundation

Windows Communication Foundation (WCF) es un marco para crear aplicaciones orientadas a servicios con el cual se puede enviar datos como mensajes asincrónicos desde el *endpoint* de un servicio a otro. El *endpoint* de un servicio puede ser parte de un servicio continuamente disponible hospedado por IIS, o puede ser un servicio alojado en una aplicación. Un *endpoint* puede ser un cliente de un servicio que solicita datos del servicio, sus mensajes pueden ser tan simples como un solo carácter o palabra enviada como XML, o tan compleja como una secuencia de datos binarios [17].

Características:

- **Orientación a servicios:** Como consecuencia del uso de los estándares de WS, WCF le permite crear aplicaciones orientadas a servicios, La arquitectura orientada a servicios (SOA) es la dependencia de los servicios web para enviar y recibir datos.
- **Interoperabilidad:** WCF implementa los estándares del sector modernos para la interoperabilidad de servicios web.
- **Varios modelos de mensajes:** Los mensajes se intercambian mediante uno de los distintos modelos. El más común es el de solicitud/respuesta, en que un extremo solicita datos de otro extremo y el otro extremo responde.
- **Metadatos de servicios:** WCF admite la publicación de metadatos de servicios utilizando los formatos especificados en los estándares de la industria, como WSDL, esquemas XML y WS-policy. Los metadatos se pueden publicar sobre

HTTP y HTTPS, o utilizando el estándar intercambio de metadatos de servicios web.

- **Contratos de datos:** WCF se basa en .Net framework, por tal motivo, también incluye métodos con código sencillo para proporcionar los contratos que desea aplicar. WCF incluye un completo sistema para trabajar con datos de esta manera fácil. Cuando se han creado las clases que representan los datos, el servicio genera automáticamente los metadatos que permiten a los clientes ajustarse a los tipos de datos que se han diseñado.
- **Seguridad:** Es posible cifrar los mensajes para proteger la privacidad, así como obligar a los usuarios a que se autenticuen antes de permitirles recibir mensajes. La seguridad puede implementarse utilizando estándares conocidos como SSL o WS-secure conversation.
- **Transacciones:** Wcf también admite las transacciones que usan uno de los tres modelos de transacción: las transacciones WS-atomic, las API del espacio de nombres system.transactions y coordinador de transacciones distribuidas de Microsoft.
- **Extensibilidad:** La arquitectura de WCF tiene varios puntos de extensibilidad. Si se necesita una función adicional, existen una serie de puntos de entrada que le permiten personalizar el comportamiento de un servicio [18].

2.2.7. Elementos de un servicio en WCF

La identificación o creación de cualquier servicio en WCF se identifica por las siglas:

- Address (Dirección): Específica donde se encuentra ubicado el servicio.
- Binding (Enlace): Define como acceder al servicio.
- Contract(Contrato): Detalla cuales operaciones ofrece el servicio.

En WCF se puede tener un mismo contrato respondiendo a varias formas de enlace, y un mismo enlace para varios contratos. Así mismo, un servicio puede ser alcanzable desde varias ubicaciones. Esta libertad permite crear servicios que interactúen de múltiples formas con diversos clientes, en diferentes plataformas y usando diversas tecnologías, alcanzando una compatibilidad e interoperabilidad sin presentes teniendo en cuenta la facilidad y transparencia del proceso actual.

Los tipos de contratos son los siguientes:

- **Contratos de Servicio:** Este tipo de contrato describe que hace un servicio. Cada contrato de servicio ofrece un conjunto de operaciones denominadas operaciones de servicio. Un servicio debe contener al menos un contrato de servicio y puede tener más de uno. En términos de C# un contrato es un tipo (clase o interfaz) en el que los métodos son las operaciones del servicio. Los atributos `ServiceContract` y `OperationContract` pertenecen a esta categoría.
- **Contrato de Datos:** Las operaciones de un servicio aceptan tipos de datos complejos, es decir estructuras además de los conocidos tipos simples que son multiplataforma por naturaleza (cadenas, enteros, booleanos, etc.) la forma de definir estas estructuras se realiza definiendo contratos de datos. Esto debe hacerse lo más neutral posible a un lenguaje de programación de modo que los dos extremos de una comunicación no dependan de esto. Los atributos `DataContract` y `DataMember` son usados con este propósito.
- **Contratos de Mensaje:** Estos contratos definen como son pasados los mensajes de un extremo a otro de la comunicación. Permitiendo definir como son almacenados los datos en el cuerpo y cabecera del mensaje. El atributo `MessageContract` especifica que la clase en cuestión es un contrato de mensaje. Mientras que los atributos, `MessageHeader` y `MessageBodyMember` marcan que contenido será colocado en la cabecera del mensaje SOAP y cual dentro del cuerpo de este [19].

2.2.8. Smart Client

Un Cliente Inteligente consiste en una aplicación que proporciona al usuario un interfaz Rica para acceder a un sistema basado en servicios web, pero que mantiene las ventajas de los clientes web tradicionales, como la ausencia de instalación en la máquina cliente, el funcionamiento a través de Internet y la actualización automática.

El término Cliente Inteligente tiene la intención de referirse simultáneamente a la captura de los beneficios de un Cliente Liviano (cero instalaciones, autoactualizaciones) y un Cliente Pesado (alta presentación, alta productividad).

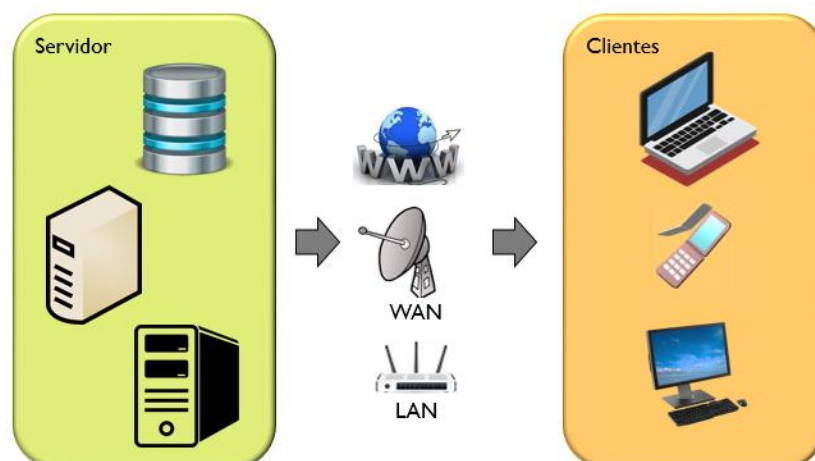


Fig. 3 Arquitectura Smart Client

Las aplicaciones Cliente Inteligente estrecharon las diferencias entre las aplicaciones web y las aplicaciones de escritorio. Otorgan los beneficios de una aplicación web (tales como apoyarse en Internet y ofrecer acceso remoto a los datos) mientras mantienen ese aspecto vivaz y elegante inherente a las aplicaciones de escritorio [20].

2.2.9. DevExpress

DevExpress (Developer Express) es una compañía de desarrollo de software con sede en los Estados Unidos, produce herramientas y componentes de asistencia de codificación para desarrolladores Delphi, C ++ Builder y Microsoft Visual Studio, la mayor parte de su línea de productos son los componentes VCL, .NET WinForms y ASP.NET que replican la interfaz de usuario de las aplicaciones de Microsoft Windows y Microsoft Office [21].

2.2.10. DevExpress WinForms

DevExpress WinForms es una suite de componentes de UI para el desarrollo en todas las plataformas de .NET como Windows Forms, ASP.NET, MVC, Silverlight y Windows 8 XAML, estos controles y componentes se integran al entorno de desarrollo para que el usuario puede utilizarlos directamente desde la barra de herramientas, como si fueran componentes propios.

Controles DevExpress en Visual Studio Toolbox

El instalador registra automáticamente los controles de DevExpress WinForms en la caja de herramientas de Visual Studio. Estos controles tienen nombres de categoría que comienzan con "DX + número de versión".

Menú DevExpress en Visual Studio

El menú de DevExpress Visual Studio es agregado automáticamente por el Instalador Unificado, se utiliza para invocar a los asistentes de DevExpress y acceder a proyectos de utilidad, recursos de soporte técnico y detalles de licencias y/o registro [22].

2.2.11. Financial Business System v2.0

Es un sistema financiero administrativo contable de última tecnología orientado a gestionar las operaciones del negocio financiero, con especial atención para Cooperativas de Ahorro y Crédito e IMF's, creado por Sifizsoft S.A. [7].

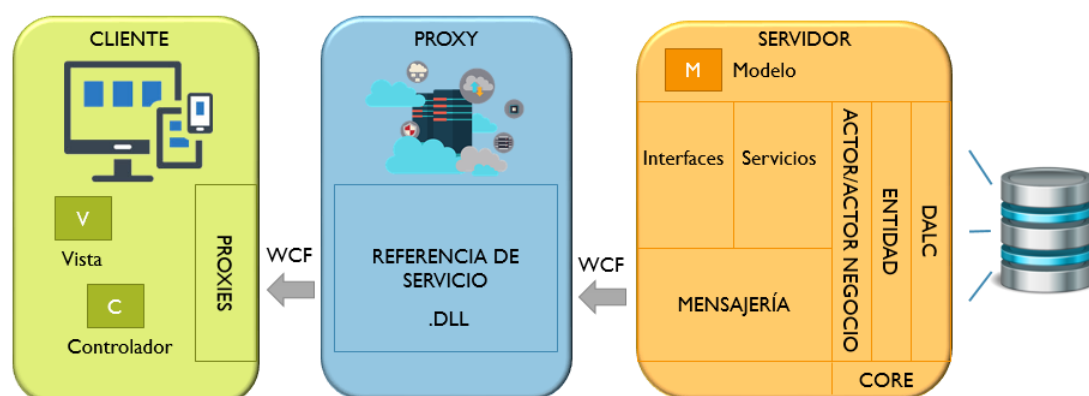


Fig. 4 Arquitectura del sistema financiero

El sistema tiene una arquitectura n capas ya que el servidor se conecta a las demás capas para ser capaz de adaptarse a los diferentes clientes, además, tiene integrada la capa de servicios externos, la cual se utiliza para traer al sistema información que no necesariamente le perteneces a la entidad.

La seguridad del aplicativo está implementada a través del protocolo de comunicación WCF (Windows Communication Foundation), el aplicativo está dividido en tres proyectos individuales que son: FBSServidor, FBSProxies, FBSCliente.

2.2.12. Arquitectura N Capas

El patrón arquitectónico a N capas ayuda a estructurar las aplicaciones que se pueden descomponer en grupos de subtareas en la que cada grupo de subtareas está en un nivel particular de abstracción.

Los sistemas a menudo requieren alguna estructuración horizontal que es ortogonal a su subdivisión vertical. Este es el caso en el que varias operaciones están en el mismo nivel de abstracción, pero son en gran parte independientes entre sí [23].

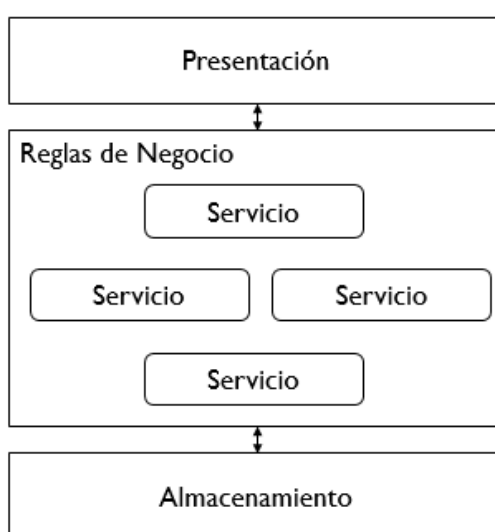


Fig. 5 Arquitectura N Capas

2.2.13. Microsoft SQL Server

Microsoft SQL Server es un sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos [24].

2.2.14. Plan de Cuentas

El plan de cuentas busca la uniformidad en el registro de las operaciones económicas realizadas por las entidades con el fin de permitir la presentación comprensible de la información y una adecuada clasificación de los hechos económicos.

El plan de cuentas está compuesto por un catálogo de cuentas y la descripción y dinámica para la aplicación de estas, las cuales deben observarse en el registro contable de todas las operaciones o transacciones económicas.

El catálogo de cuentas contiene la relación ordenada y clasificada de las clases, grupos, cuentas y subcuentas del activo, pasivo, patrimonio, ingresos, gastos, costo de ventas y costos de producción o de operación, identificadas con un código numérico y su respectiva denominación [25].

2.2.15. Proyecciones

Modelos De Regresión Con Dos Variables

1. Método de mínimos cuadrados ordinarios (MCO)

Este método se atribuye a Carl Friedrich Gauss, matemático alemán, presenta propiedades estadísticas muy atractivas que lo han convertido en uno de los más eficaces y populares del análisis de regresión, su fórmula es:

$$\sum \hat{u}_i^2 = f(\hat{\beta}_1, \hat{\beta}_2)$$

Se representa de la siguiente manera:

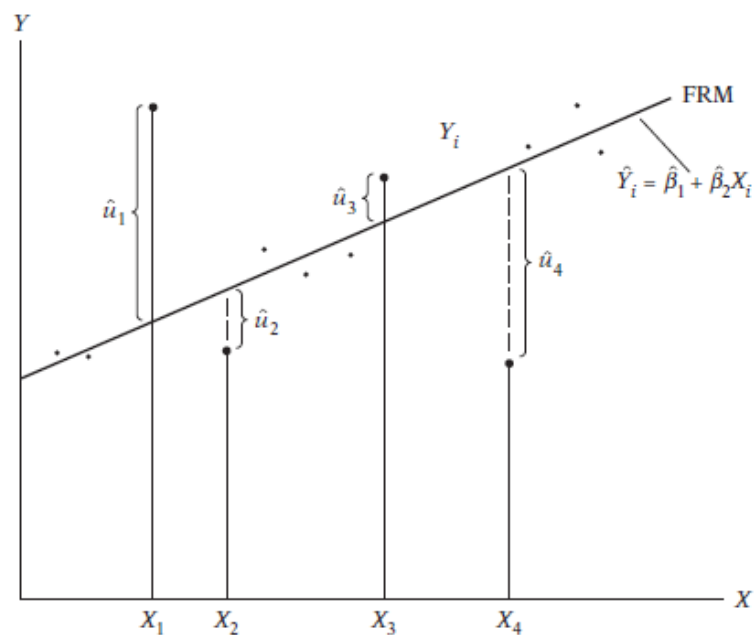


Fig. 6 Criterio de mínimos cuadrados

Este criterio, aunque es intuitivo, no es recomendable aplicarlo en este caso a cuentas contables, ya que, como se ve en la figura presenta un alto índice de dispersión hipotético.

2. Modelo clásico de regresión lineal

El modelo de Gauss, modelo clásico o estándar de regresión lineal (MCRL), es el cimiento de la mayor parte de la teoría econométrica, el mismo ha servido de base para derivaciones de regresiones lineales y plantea siete supuestos que son:

- **Supuesto 1:**

Modelo de regresión lineal: El modelo de regresión es lineal en los parámetros, aunque puede o no ser lineal en las variables:

$$Y_i = \beta_1 + \beta_2 X_i + u_i$$

- **Supuesto 2:**

Valores fijos de X , o valores de X independientes del término de error: Los valores que toma la regresión X pueden considerarse fijos en muestras repetidas (el caso de la regresión fija), o haber sido muestreados junto con la variable dependiente Y (el caso de la regresión estocástica).

$$\text{cov}(X_i, u_i) = 0$$

- **Supuesto 3:**

El valor medio de la perturbación u_i es igual a cero: Dado el valor de X_i , la media o el valor esperado del término de perturbación aleatoria u_i es cero. Simbólicamente, tenemos que:

$$E(u_i | X_i) = 0$$

O, si X no es estocástica:

$$E(u_i) = 0$$

- **Supuesto 4:**

Homoscedasticidad o varianza constante de u_i : La varianza del término de error, o de perturbación, es la misma sin importar el valor de X . Simbólicamente, tenemos que:

$$\text{var}(u_i) = E[u_i - E(u_i|X_i)]^2$$

$$\text{var}(u_i) = E(u_i^2 | X_i), \text{ por el supuesto 3}$$

$$\text{var}(u_i) = E(u_i^2), \text{ si } X_i \text{ son variables no estocásticas}$$

$$\text{var}(u_i) = \sigma^2$$

Donde var significa varianza.

- **Supuesto 5:**

No hay autocorrelación entre las perturbaciones: Dados dos valores cualesquiera de X , X_i y X_j ($i \neq j$), la correlación entre dos u_i y u_j cualesquiera ($i \neq j$) es cero. En pocas palabras, estas observaciones se muestrean de manera independiente. Simbólicamente:

$$\begin{aligned} cov(u_i, u_j | X_i, X_j) &= 0 \\ cov(u_i, u_j) &= 0, \text{ si } X \text{ no es estocástica} \end{aligned}$$

- **Supuesto 6:**

El número de observaciones n debe ser mayor que el número de parámetros por estimar: Sucesivamente, el número de observaciones n debe ser mayor que el número de variables explicativas.

- **Supuesto 7:**

La naturaleza de las variables X : No todos los valores X en una muestra determinada deben ser iguales. Técnicamente, $var(X)$ debe ser un número positivo. Además, no puede haber valores atípicos de la variable X , es decir, valores muy grandes en relación con el resto de las observaciones.

Modelo Clásico De Regresión Lineal Normal (MCRLN)

La llamada teoría clásica de la inferencia estadística consta de dos ramas, a saber: estimación y pruebas de hipótesis, con los supuestos del modelo clásico de regresión lineal (MCRL) se demuestra que los estimadores de dichos parámetros, $\hat{\beta}_1$, $\hat{\beta}_2$ y $\hat{\sigma}^2$, satisfacen varias propiedades estadísticas deseables, como el insesgamiento, la varianza mínima, etc.

Como las variables mencionadas son aleatorias es necesario averiguar sus distribuciones de probabilidad, pues sin conocerlas no es posible relacionarlas con sus valores verdaderos.

1. Distribución de probabilidad de las perturbaciones u_i

Para encontrar las distribuciones de probabilidad de los estimadores:

$$\hat{B}_2 = \sum k_i Y_i$$

Donde:

$$k_i = x_i / \sum x_i^2$$

Se supone que las X son fijas, o no estocásticas, debido a que se trata de un análisis de regresión condicional (condicional en los valores fijos de X_i), por lo cual la fórmula se plantea de la siguiente manera:

$$\hat{\beta}_2 = \sum k_i(\beta_1 + \beta_2 X_i + u_i)$$

2. Supuesto de normalidad de u_i

El modelo clásico de regresión lineal normal supone que cada u_i está normalmente distribuida con:

Media: $E(u_i) = 0$

Varianza: $E[u_i - E(u_i)]^2 = E(u_i^2) = \sigma^2$

$cov(u_i, u_j): E\{[(u_i - E(u_i))][u_j - E(u_j)]\} = E(u_i u_j) = 0; i \neq j$

3. Propiedades de los estimadores según el supuesto de normalidad

Si se supone que u_i sigue una distribución normal los estimadores de MCO tienen las siguientes propiedades:

- Son insesgados.
- Tienen varianza mínima en combinación con 1, esto significa que son estimadores insesgados con varianza mínima, o eficientes.
- Presentan consistencia; es decir, a medida que el tamaño de la muestra aumenta indefinidamente, los estimadores convergen hacia sus verdaderos valores poblacionales.
- $\hat{\beta}_1$ (al ser una función lineal de u_i) está normalmente distribuida con

$$E(\hat{\beta}_1) = \beta_1$$

- $\hat{\beta}_2$ (al ser una función lineal de u_i) está normalmente distribuida con

$$E(\hat{\beta}_2) = \beta_2$$

Modelo De Regresión Múltiple

La adición de variables conduce al análisis de los modelos de regresión múltiple, es decir, modelos en los cuales la variable dependiente, o regresada, Y , depende de dos o más variables explicativas, o regresoras, como todo modelo, tiene funciones que derivan del mismo para ajustarse a los tipos de datos y son:

1. Modelo con tres variables

Se generaliza la función de regresión poblacional para concluir su fórmula de la siguiente manera:

$$Y_i = \beta_1 + \beta_2 X_{2i} + \beta_3 X_{3i} + u_i$$

2. Ecuación de regresión múltiple

Con los supuestos del modelo de regresión clásico, se cumple que, al tomar la esperanza condicional de Y en ambos lados obtenemos:

$$E(Y_i | X_{2i}, X_{3i}) = \beta_1 + \beta_2 X_{2i} + \beta_3 X_{3i}$$

Modelos De Regresión No Lineales

- **Modelos de regresión intrínsecamente lineales e intrínsecamente no lineales**

Si un modelo es no lineal en los parámetros, se trata de un modelo de regresión no lineal (en los parámetros), sin importar que las variables de tal modelo sean lineales o no, sin embargo, algunos modelos pueden parecer no lineales en los parámetros, pero ser inherente o intrínsecamente lineales.

Se rige por la función de distribución de probabilidad logística:

$$\ln \frac{1 - Y_i}{Y_i} = \beta_1 + \beta_2 X_i + u_i$$

- **Estimación de modelos de regresión lineales y no lineales**

Se aplica esta estimación para resaltar la diferencia entre modelos de regresión lineales y no lineales:

$$Y_i = \beta_1 + \beta_2 X_i + u_i$$

$$Y_i = \beta_1 e^{\beta_2 X_i} + u_i$$

Modelo De Regresión De Respuesta Cualitativa

En todos los modelos de regresión ya considerados, se dio a conocer que, en la regresión, la variable dependiente o la variable de respuesta Y era cuantitativa, mientras que las variables explicativas podían ser cuantitativas o cualitativas, o una mezcla de las dos.

La regresión es una variable binaria, o dicótoma. La investigación en economía del trabajo indica que la PFL es una función de la tasa de desempleo, tasa del salario promedio, escolaridad, ingreso familiar, etcétera.

Existen cuatro métodos para crear un modelo de probabilidad para una variable de respuesta binaria y son:

1. Modelo lineal de probabilidad (MLP)

La esperanza condicional del modelo se interpreta como la probabilidad condicional de Y_i .

En general, la esperanza de una variable aleatoria Bernoulli está dada por la probabilidad de que esa variable sea igual a 1.

$$E(Y_i | X_i) = \beta_1 + \beta_2 X_i = P_i$$

2. Modelo logit

A esta ecuación de estimación se la conoce como función de distribución logística la cual es muy común en análisis de fenómenos de crecimiento, como población, PNB, oferta monetaria, etc.

$$L_i = \ln \frac{P_i}{1 - P_i} = \beta_1 + \beta_2 X_i + u_i$$

3. Modelo probit

Para explicar el comportamiento de una variable dependiente dicótoma es preciso utilizar una función de distribución acumulativa (FDA) seleccionada apropiadamente. El modelo logit utiliza una función logística acumulativa, pero no es la única aplicable.

$$P_i = P(Y = 1 | X) = P(I_i^* \leq I_i) = P(Z_i \leq \beta_1 + \beta_2 X_i) = F(\beta_1 + \beta_2 X_i)$$

4. Modelo tobit

Una extensión del modelo probit es el modelo tobit, desarrollado por James Tobin, economista laureado con el Nobel, el modelo tobit también se conoce como modelo de regresión censurada. Algunos autores los llaman modelos de regresión con variable dependiente limitada debido a la restricción impuesta sobre los valores tomados por la variable regresada.

$$Y_i = \beta_1 + \beta_2 X_i + u_i \quad [26]$$

Definición de datos

Para saber qué método de proyección es el adecuado para ser aplicado en el módulo de presupuesto financiero, se aplicó el coeficiente de correlación lineal, el cual nos ayuda a saber si existe una relación directa entre los datos.

Tabla 2.2: Búsqueda de coeficiente de correlación

	x	y	1	2			
OFICINA	AÑO ANTERIOR	MES ANTERIOR	x - media x	y - media y	1 ²	2 ²	1*2
MATRIZ	10399.04	1638.23	6673.26	-925.55	44532356.88	856651.5709	-6176464.48
SALCEDO	23383.34	889	19657.56	-1674.78	386419541	2804903.915	-32922176.1
PUYO	2224.91	799	-1500.87	-1764.78	2252620.236	3114465.167	2648718.041
PILLARO	1040	1222.5	-2685.78	-1341.28	7213431.171	1799044.745	3602399.956
IZAMBA	1845.66	1888.11	-1880.12	-675.67	3534863.089	456536.35	1270351.72
MACAS	540	1040	-3725.78	-2563.78	13881460.14	6572992.177	9552105.993
PELILEO	9838.47	2089	6112.69	-474.78	37364940.43	225420.5463	-2902210.41
TENA	565.56	320	-3160.22	-2563.78	9987010.408	6572992.177	8102131.897
LATACUNGA	200	21067	-3525.78	18503.22	12431146.88	342368975.1	-65238324.7
OFICINA SUR	8832.31	7696.63	5106.53	5132.85	26076616.39	26346100.5	26211012.11
CEVALLOS	4194.69	3293.61	468.91	729.83	219873.6266	532644.9147	342220.0594
TISALEO	7558.3	2187	3832.52	-376.78	14688185.35	141966.7379	-1444033.85
SAQUISILÍ	2723	452	-3725.78	-2111.78	13881460.14	4459634.775	7868052.006
EL QUINCHE	44	230	-3725.78	-2563.78	13881460.14	6572992.177	9552105.993
SANGOLQUÍ	320	4324.83	-3725.78	1761.05	13881460.14	3101280.419	-6561272.78
CONOCOTO	249	715	-3476.78	-1848.78	12088021.13	3418005.003	6427823.636
ALANGASÍ	200	200	-3725.78	-2363.78	13881460.14	5587478.282	8806949.361
VILLAFLORES	34	250	-3725.78	-2313.78	13881460.14	5353599.808	8620660.204
PUENGASÍ	458.6	1354.45	-3267.18	-2563.78	10674485.79	6572992.177	8376354.313
SUMA:	70789.88	48711.91			650771853.2	426858676.5	-13863597.1
N:	19						
MEDIA:	3725.78	2563.78					

Los datos utilizados son saldos reales tomados de una cuenta contable que consta el plan de cuentas de la empresa, fueron proporcionados por la contadora general y toma como mes de referencia octubre 2017 para campo mes anterior y diciembre 2016 para campo año anterior.

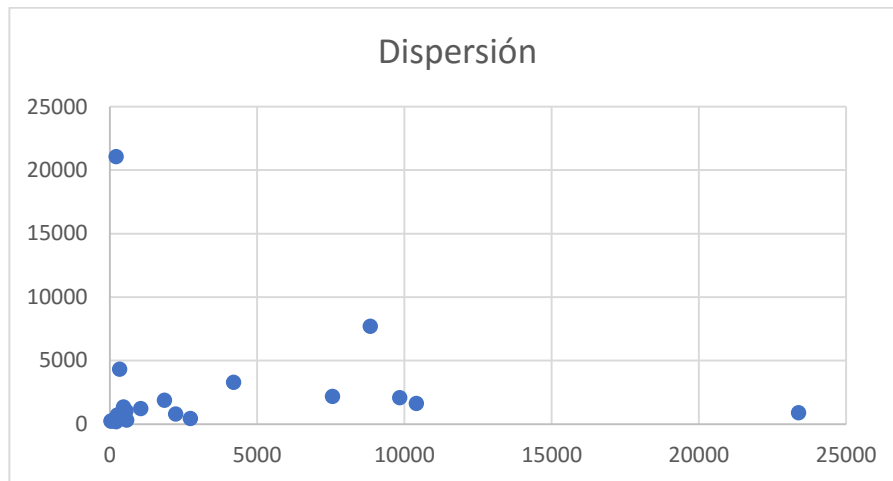


Fig. 7 Dispersión de datos - búsqueda de coeficiente de correlación

Por medio del gráfico de dispersión, se puede observar que los datos son altamente dispersos, por lo que el resultado del coeficiente de correlación será cercano a cero, lo cual representa que los datos tienen un índice bajo de correlación.

Tabla 2.3: Resultados – Coeficiente de correlación

	Resultado	Fórmula
Covarianza	-1647052.84	$cov(X, Y) = \frac{\sum(Xi - \bar{X})(Yi - \bar{Y})}{N}$
Desviación x	5754.03	$\sigma = \sqrt{\frac{\sum(Xi - \bar{X})^2}{N}}$
Desviación y	4670.04	$\sigma = \sqrt{\frac{\sum(Yi - \bar{Y})^2}{N}}$
Coeficiente de correlación:	-0.061	$r = \frac{\sum(Xi - \bar{X})(Yi - \bar{Y})}{\sqrt{\sum(Xi - \bar{X})^2 \sum(Yi - \bar{Y})^2}}$

Método aplicado

Con el resultado obtenido se decide aplicar el modelo clásico de regresión lineal supuesto 1, el cual ha sido analizado juntamente con la Eco. Alejandra Cuesta docente de la Facultad de Contabilidad y Auditoría en la carrera de Economía para plasmar el modelo en pasos detallados, para que el mismo pueda ser introducido en el módulo de presupuesto financiero.

Tabla 2.4: Aplicación del método detallado

OBS	Y	X	$X_i (X - X_{MEDIA})$	$y_i (Y - Y_{MEDIA})$	$x_i^2 (X - X_{MEDIA})^2$	$y_i x_i (X - X_{MEDIA})(Y - Y_{MEDIA})$
1	262,810.45	1.00	-0.5000	18,966.2400	0.2500	-9,483.120
2	224,877.97	2.00	0.5000	-18,966.2400	0.2500	-9,483.120
TOTAL	487,688.42	3.00	0.00	0.00	0.5000	-18,966.2400

Yesti	Y-Yesti	(Y-Yesti) ²	X ²	Y ²	y _i ²
262810.45	0.00	0.00	1.00	69,069,332,629.20	359,718,259.73
224877.97	0.00	0.00	4.00	50,570,101,391.32	359,718,259.73
487,688.42	0.00	0.00	5.00	119,639,434,020.52	719,436,519.47

X MEDIA: 1.50

Y MEDIA: 243,844.21

La tabla 2.4 representa los cálculos previos necesarios para llevar a cabo una correcta proyección, el valor asignado a Y representa el saldo de la cuenta contable a fecha cierre 31 de octubre de los años 2016 y 2017, mientras que el valor de X es un secuencial por cada año transcurrido, cada valor obtenido será reemplazado en las siguientes fórmulas para poder obtener una proyección final.

Cálculo de Betas

$$\hat{\beta}_2 = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sum(X_i - \bar{X})^2} = -37,932.48$$

$$\hat{\beta}_2 = \frac{\sum X_i Y_i}{\sum X_i^2} = -37,932.48$$

$$\hat{\beta}_1 = \bar{Y} - \hat{\beta}_2 \bar{X} = 300,742.93$$

Coefficiente de determinación r^2

$$r^2 = 1 - \frac{\sum \hat{u}_i^2}{\sum (Y_i - \bar{Y})^2} = 100.00\%$$

Coefficiente de determinación r^2 ajustado

$$r_a^2 = 1 - (1 - r^2) \frac{n-1}{gl} = 100.00\%$$

Fórmula por aplicar:

$$\hat{Y}_i = \hat{\beta}_1 + \hat{\beta}_2 X$$

$\hat{Y}_i =$	$\hat{\beta}_1$	$\hat{\beta}_2$	X
	300742.93	-37932.4800000	3
$Y_3 =$	186,945.49		

El valor Y3, es la proyección buscada del nuevo saldo que tendrá una cuenta contable al año especificado, en este caso, se posee 2 históricos, los cuales interfieren directamente en el cálculo del tercer año. (El resultado representa un monto en dólares)

2.3. Propuesta de Solución

Implantar un módulo de presupuesto financiero que permita centralizar los respectivos saldos de cuentas contables para presentar una proyección matemática de su valor con la finalidad de alcanzar una administración adecuada del dinero disponible y lograr una oportuna toma de decisiones.

CAPÍTULO III

METODOLOGÍA

3.1. Modalidad de Investigación

La presente investigación se considera de tipo aplicada debido a que busca la utilización de los conocimientos adquiridos a lo largo de la carrera universitaria para dar solución práctica a los problemas existentes dentro de la empresa, con la finalidad de obtener una mejora de tiempo y esfuerzo en el proceso de presupuestación de la “Cooperativa de Ahorro y Crédito San Francisco LTDA.”.

3.1.1. Investigación de campo

Se considera esta modalidad ya que la investigación se realiza en su totalidad dentro de las instalaciones de la empresa “Cooperativa de Ahorro y Crédito San Francisco LTDA.”, para estar de forma directa en contacto con la realidad en el lugar de los hechos, con el fin de recolectar información sobre el problema investigado. Las técnicas por utilizar son: entrevista y observación.

3.1.2. Investigación bibliográfica o documentada

Se considera esta modalidad por la fundamentación de los conceptos utilizados que proporcionan un sustento mediante estudios realizados en trabajos referentes al tema de presupuesto y proyecciones de autores reconocidos y, además, recurre a diferentes fuentes obtenidas de libros, artículos científicos, tesis desarrolladas en Universidades para profundizar enfoques con respecto al tema de la investigación.

3.2. Población y Muestra

Para realizar la investigación se trabajó con la “Cooperativa de Ahorro y Crédito San Francisco LTDA.”, se aplicó una entrevista al Ing. Diego Torres jefe del departamento de TI., y al Ing. Jairo Bejarano jefe del área de desarrollo, quienes poseen a su cargo el análisis de nuevas funcionalidades para ser implantadas al sistema financiero.

3.3. Recolección de Información

La técnica aplicada en la recolección de información es la entrevista, la cual fue aplicada utilizando la guía de entrevista como instrumento, estos datos contribuyeron al correcto análisis de la información requerida para el desarrollo del proyecto.

3.4. Procesamiento y análisis de datos

Una vez obtenida la información, para el procesamiento y análisis se aplican los siguientes pasos:

- Organizar y revisar la información obtenida
- Identificar el criterio bajo el cual se registrarán las proyecciones
- Separar los datos que serán necesarios para el cumplimiento de los requisitos

3.5. Desarrollo del proyecto

Las acciones realizadas para el desarrollo del presente proyecto de llevaron a cabo de la siguiente manera:

1. Análisis
 - a. Estudio de la de la situación actual de la institución y sus políticas.
 - b. Establecer requerimientos.
 - c. Determinar límites de acceso a información y firma de acta de confidencialidad.
2. Diseño
 - a. Conocimiento de la estructura del sistema que maneja la institución.
 - b. Creación de esquema de base de datos.
 - c. Elaboración de interfaces.

3. Construcción
 - a. Elaboración de clases y objetos.
 - b. Desarrollo del módulo integral de presupuestos.
 - c. Definición de permisos y horarios de acceso al módulo.
4. Validación y pruebas
 - a. Verificación de correcto funcionamiento.
 - b. Ejecución en entorno de pruebas.
5. Producción
 - a. Despliegue del módulo de presupuestos en ambiente de producción.
 - b. Elaboración del manual de implantación.
 - c. Elaboración de manual de usuario.

CAPÍTULO IV

DESARROLLO DE LA PROPUESTA

4.1. Análisis de Requerimientos

4.1.1. Identificación de necesidades

Aplicación de entrevista

Para la revisión de la situación actual se realizó una entrevista (Anexo1) al jefe de departamento de TI de la Cooperativa de Ahorro y Crédito “San Francisco LTDA.”.

De los datos obtenidos en la entrevista se puede concluir que el proceso de cálculo de presupuesto anual requiere automatizarse para que pueda realizarse en menos tiempo y con menor esfuerzo, por esa razón se evaluó el proceso actual de la empresa de la siguiente manera:

- **Personal Involucrado:** Contadora general, Gestor de base de datos.
- **Descripción:** El proceso es llevado manualmente en hojas de cálculo, las cuales se llenan de información extraída de la base de datos, se considera información del plan de cuentas, dividido por sus 33 agencias operativas (a fecha 08/2018), sueldos de personal, proveedores, eventos, agasajos, comisiones de los miembros de la asamblea e imprevistos, cada uno de estos grupos de datos afectan al archivo principal de cuentas para que se logre valores aceptables para el transcurso del año siguiente. (Anexo 2)
- **Tiempo de conclusión:** 15 días.
- **Conclusión:** Proceso no óptimo

Observación de procesos

Mediante la observación de actividades en el área de investigación, se pudo identificar el proceso que se lleva a cabo para coordinar la implementación de nuevos requerimientos solicitados por los diferentes departamentos al área de desarrollo, de esta manera se ha podido estructurar el procedimiento a llevar a cabo para la construcción e implantación de una solución informática al problema planteado en la aplicación de la entrevista.

Se utilizó una guía de observación como herramienta para el análisis de procesos, la cual se adjunta como anexo (Anexo 3). A continuación, se detalla el proceso mediante el diagrama de actividades:

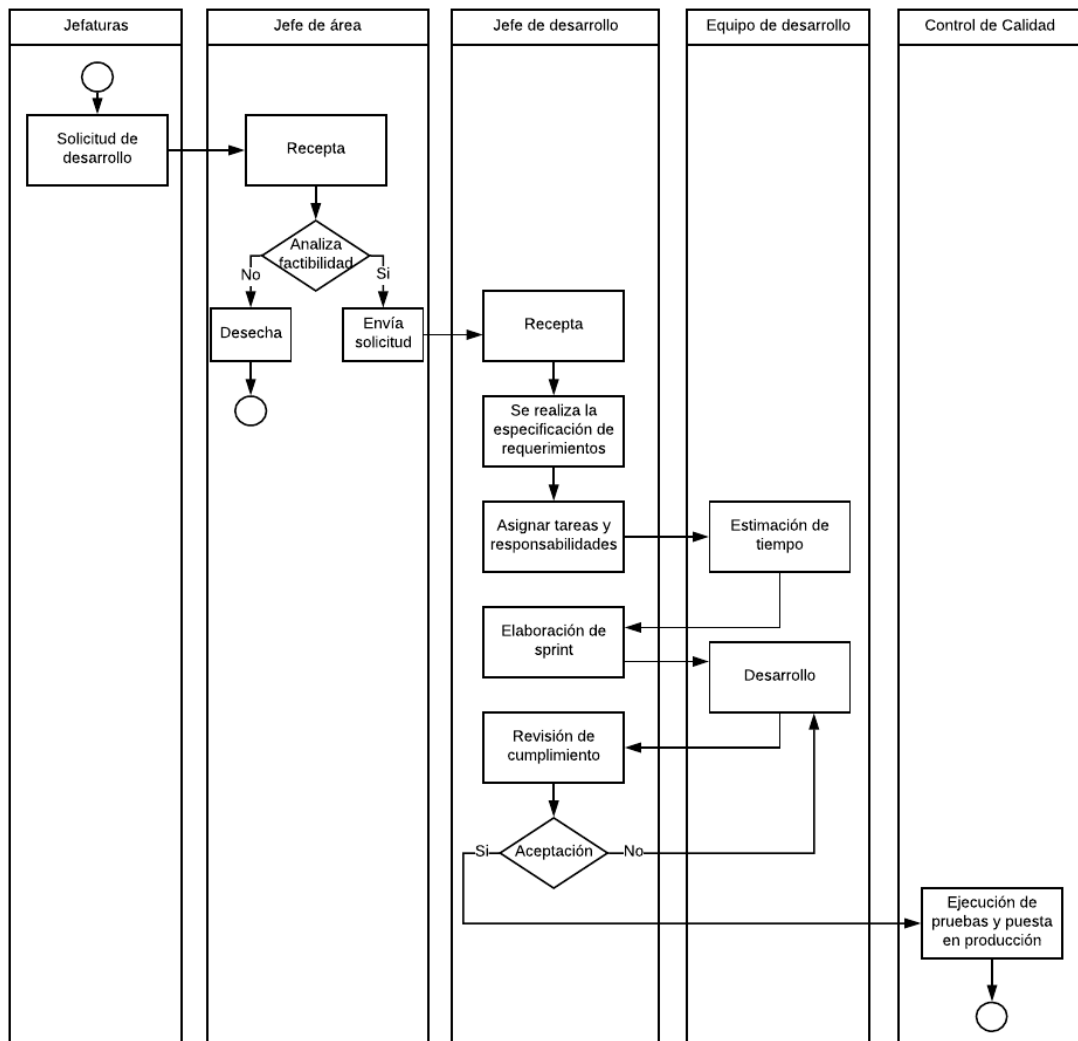


Fig. 8 Diagrama de actividades

Levantamiento y especificación de requisitos según documento de especificación de requerimientos establecido por el jefe de desarrollo

1. Introducción

En el presente documento se encuentra la especificación de requerimientos para la implantación del módulo de presupuesto financiero aplicado a la proyección de crecimiento en la Cooperativa de Ahorro y Crédito “San Francisco Ltda.”, el mismo que fue desarrollado y documentado por el jefe de desarrollo de la entidad en su tesis sobre “Implementación de un marco de trabajo basada en la norma ISO/IEC 12207 y la metodología ágil SCRUM para el mejoramiento de la calidad de los productos de software” [27].

1.1. Propósito

El objetivo principal de este documento es la redacción en forma clara y concreta de las necesidades del cliente en términos del software que se va a realizar. La correcta documentación de este será una guía para validar e inspeccionar la construcción del software en cada una de sus etapas.

1.2. Alcance

El módulo se desarrollará con el fin de mejorar el proceso de cálculo de presupuesto financiero anual de la empresa, manteniendo la asertividad y confianza del mismo. El sistema permitirá:

- Visualizar gráficamente la información base conformada por el saldo de las cuentas contables de años anteriores en cada una de las sucursales.
- Seleccionar la cantidad de años históricos a mostrar en la información base.
- Elegir las cuentas contables que intervendrán directamente con el resultado final del cálculo a mostrar.

1.3. Personal involucrado

Nombre	Jairo Andrés Bejarano Montesdeoca
Rol	Líder de proyecto
Categoría profesional	Analista funcional

Responsabilidad	Aprobación de funcionalidades
Información de contacto	jairo.bejarano@coac-sanfra.com

Nombre	Pamela Solange Méndez Pinto
Rol	Analista desarrollador
Categoría profesional	Estudiante
Responsabilidad	Análisis y especificación de requerimientos, diseño de la arquitectura, programación del módulo
Información de contacto	pamela.mendez@coac-sanfra.com

2. Descripción general

2.1. Perspectiva del producto

El módulo funcionará dentro del aplicativo financiero existente en la Cooperativa de Ahorro y Crédito “San Francisco Ltda.”, por lo que su desarrollo se llevará a cabo dentro de la arquitectura que el mismo maneja y este a su vez será el proveedor de los datos necesarios para que el módulo cumpla con su función.

2.2. Funcionalidad del producto

Las principales funciones que el sistema debe brindar son:

- Administrar información referente al estado de cuentas reflejadas en el balance general y estado de resultados de años pasados.
- Permitir seleccionar las cuentas que estarán involucradas en el proceso.
- Ser integrado a una arquitectura ya establecida.
- Brindar funcionalidades para exportar los resultados a diferentes formatos.

2.3. Características del usuario

El producto por construir va dirigido a dos tipos de usuarios principales, el primero, el usuario que posee conocimientos avanzados en sistemas de información, procesos de ingeniería y gestión de pruebas, en segundo lugar,

el usuario que posee conocimiento sobre un área específica distinta al manejo de sistemas de información, el cual permite verificar la funcionalidad y usabilidad del módulo.

2.4. Restricciones

2.4.1. Restricción de herramienta de desarrollo

Microsoft Visual Studio 2013

2.4.2. Restricción de diseño

Adaptación al formato de pantallas del sistema Financiamiento 2.0

2.4.3. Restricción de metodología

SCRUM

2.4.4. Restricciones de software

Sistema operativo Windows 10 con usuario administrador

2.4.5. Restricciones de hardware

Equipo proporcionado por el área de soporte de la entidad

2.5. Supuestos y dependencias

Los requerimientos del sistema no cambiarán durante el desarrollo de la aplicación.

3. Requisitos específicos

3.1. Requisitos

3.1.1. Requisitos funcionales

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Parametrización
Características:	El módulo permitirá al usuario definir los indicadores iniciales de partida.
Descripción del requerimiento:	Parámetros que representan el comienzo de un periodo
Requerimiento no funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Selección de cuentas
Características:	El usuario podrá definir una por una, las cuentas que se requieren.
Descripción del requerimiento:	El módulo será capaz de presentar en una lista todas las cuentas existentes.
Requerimiento no funcional:	<ul style="list-style-type: none"> • RNF04 • RNF06
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Porcentajes de distribución
Características:	El usuario podrá definir los porcentajes de cumplimiento trimestral
Descripción del requerimiento:	Los valores serán utilizados para el seguimiento de los montos presupuestados.
Requerimiento no funcional:	<ul style="list-style-type: none"> • RNF02
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Proyecciones
Características:	El módulo permitirá al usuario realizar una proyección lineal para obtener un valor futuro
Descripción del requerimiento:	El módulo será capaz de consultar los valores anteriores de las cuentas seleccionadas
Requerimiento no funcional:	<ul style="list-style-type: none"> • RNF01
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Visualización de saldos contables
Características:	El módulo permitirá al usuario tener un gráfico estadístico de los saldos contables.
Descripción del requerimiento:	El módulo será capaz de consultar los valores históricos de una cuenta específica en una determinada sucursal.
Requerimiento no funcional:	<ul style="list-style-type: none"> • RNF03
Prioridad del requerimiento:	
Alta	

Identificación del requerimiento:	RF06
Nombre del Requerimiento:	Mayorización
Características:	El módulo permitirá al usuario ejecutar procesos tanto de mayorización como de la creación de la oficina consolidado.
Descripción del requerimiento:	El módulo será capaz de recorrer el plan contable para asignar los valores de las cuentas de movimiento a sus antecesoras.
Requerimiento no funcional:	<ul style="list-style-type: none"> • RNF05
Prioridad del requerimiento:	
Alta	

Identificación del requerimiento:	RF07
Nombre del Requerimiento:	Presentación de balances
Características:	El módulo permitirá al usuario visualizar tanto el balance general como el estado de pérdidas y ganancias.
Descripción del requerimiento:	Los valores de los balances representarán los datos de proyección, no sus valores actuales.

Requerimiento no funcional:	<ul style="list-style-type: none"> • RNF06 • RNF07
Prioridad del requerimiento:	
Alta	

Identificación del requerimiento:	RF08
Nombre del Requerimiento:	Comparativa de saldos contables
Características:	El módulo permitirá al usuario visualizar un reporte general de todas las cuentas en dos períodos de tiempo.
Descripción del requerimiento:	Los datos reales de las cuentas y los valores presupuestados serán presentados en un documento comparativo tomando en cuenta los porcentajes establecidos para cada trimestre.
Requerimiento no funcional:	<ul style="list-style-type: none"> • RNF06 • RNF07
Prioridad del requerimiento:	
Alta	

3.1.2. Requisitos no funcionales

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Interfaz de presentación de datos
Características:	El módulo presentará una interfaz específica para la visualización de tablas.
Descripción del requerimiento:	La interfaz será sencilla y adaptada al sistema informático que maneja la empresa.
Prioridad del requerimiento:	
Alta	

Identificación del requerimiento:	RNF02
Nombre del Requerimiento:	Interfaz de ingreso de datos

Características:	El módulo presentará una interfaz específica para el ingreso de datos al sistema.
Descripción del requerimiento:	Recepción de datos de tipo numérico que serán ingresados en forma manual.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF03
Nombre del Requerimiento:	Interfaz de presentación de gráficos
Características:	El módulo presentará una interfaz específica para la visualización de gráficos.
Descripción del requerimiento:	La interfaz será sencilla y adaptada al sistema informático que maneja la empresa.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF04
Nombre del Requerimiento:	Interfaz de presentación de plan contable
Características:	El módulo presentará una interfaz específica para la visualización del plan contable.
Descripción del requerimiento:	La interfaz será sencilla y adaptada al sistema informático que maneja la empresa.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF05
Nombre del Requerimiento:	Procesos en segundo plano
Características:	Los procesos que el módulo presenta deberán tener un método asíncronico para su ejecución
Descripción del requerimiento:	Permite que mientras se ejecute el proceso las demás funciones estén disponibles al usuario.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF06
Nombre del Requerimiento:	Formatos de cuentas
Características:	El módulo deberá presentar las cuentas contables con el sufijo correspondiente al código contable de las mismas
Descripción del requerimiento:	El formato permite la facilidad de distinción por parte del usuario
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF07
Nombre del Requerimiento:	Interfaz de presentación de reportes
Características:	El módulo presentará una interfaz específica para la visualización de reportes
Descripción del requerimiento:	La interfaz será sencilla y adaptada al sistema informático que maneja la empresa.
Prioridad del requerimiento: Alta	

3.2. Descripción de funcionalidades

- Parametrización:
 - En el menú: Parámetros > Presupuesto > Periodo Presupuestario, se presenta una tabla en la que se presentan todos los periodos presupuestarios creados (en el caso de existir), y permite añadir uno nuevo, únicamente si todos los periodos presentados están en estado: cerrado.
 - Al tener creado el periodo presupuestario a utilizar se le asignan las cuentas que van a intervenir en el cálculo de las proyecciones para conformar el presupuesto futuro, las cuales se presentan del plan de

cuentas general para que el usuario decida cuales son las que desea remover, en el caso de tener un periodo presupuestario al que con anterioridad se le hayan asignado cuentas, se podrá duplicar las mismas en la creación del nuevo periodo.

- Es necesario fijar también los valores de distribución que serán utilizados para controlar los niveles de cumplimiento a medida que va avanzando el año presupuestado.

- Proceso
 - Se debe presentar una pantalla de proyección de cuentas basada en los parámetros ya establecidos para presentar únicamente las cuentas catalogadas como: de movimiento, las cuales se seleccionan una por una para permitir al usuario ingresar un porcentaje fijo de crecimiento que afectará esa cuenta en todas las oficinas, o elegir la opción de regresión para que el sistema automáticamente consulte los saldos finales de esa cuenta para poder determinar un valor futuro, así mismo, que afectará a todas las oficinas. (cada una de las cuentas mostradas deben ser proyectadas para obtener una mayor certeza en el monto de proyección)
 - Se debe permitir y controlar el proceso de mayorización, el cual debe tomar los saldos de cada una de las cuentas presupuestadas y las dirige a sus respectivas cuentas superiores, esto permite la acumulación de los saldos a una sola cuenta principal totalizada.
 - Se debe permitir la creación de una oficina consolidado que se realiza después de culminada la mayorización para tomar los valores ya calculados y reunirlos en un solo plan de cuentas
 - Se debe presentar el balance general de los valores proyectados que permita visualizar los datos por oficina o por su consolidado.
 - Se debe presentar el estado de pérdidas y ganancias que permita visualizar los datos por oficina o por su consolidado.

3.3. Reportes

- Se debe crear un reporte que indique los siguientes filtros: Oficina (incluido consolidado) y las firmas de las diferentes autoridades que aprueban los datos expuestos. El reporte debe mostrar tanto el balance general como el estado de pérdidas y ganancias en archivos separados.
- Se debe presentar una comparativa de saldos, que tenga los siguientes filtros: Oficina (incluido consolidado), fecha de corte, trimestre y año del presupuesto. El reporte debe mostrar los valores reales y el valor presupuestado en el trimestre seleccionado tomando en cuenta los parámetros de distribución, los cuales permitirán al usuario controlar el nivel de cumplimiento de los valores proyectados al año que transcurre.

4.1.2. Funcionalidades

Diagrama de casos de uso

Representa en forma gráfica la manera como un usuario(actor), interactúa con el sistema a desarrollar, especificando qué personas tendrán acceso a cada una de las funciones del sistema.

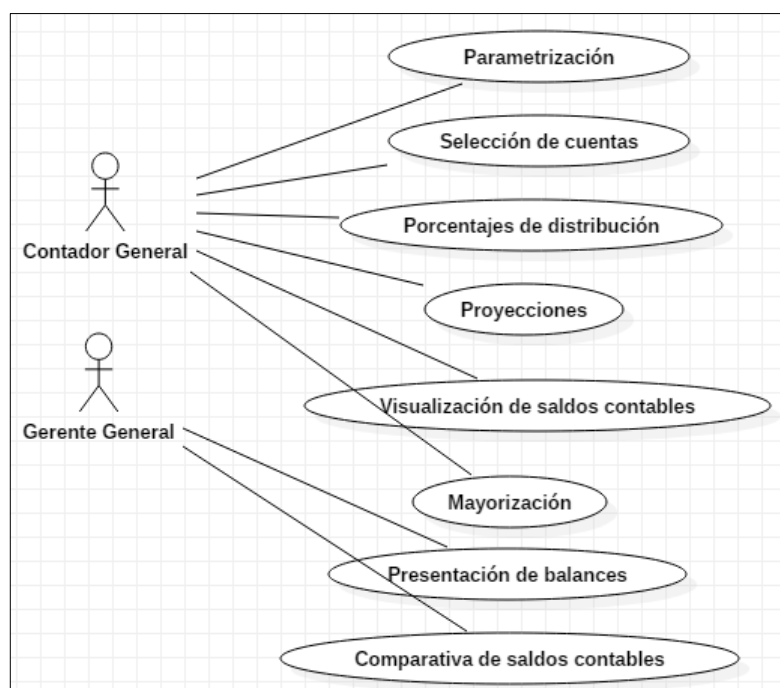


Fig. 9 Diagrama de casos de uso

Especificaciones de Casos de Uso

Tabla 4.5 Especificación Parametrización

Caso de Uso: Parametrización
<p>Descripción:</p> <ul style="list-style-type: none"> • Consultar, agregar, actualizar los datos iniciales del módulo.
<p>Actores:</p> <ul style="list-style-type: none"> • Usuario con el rol: Contador General.
<p>Precondiciones:</p> <ul style="list-style-type: none"> • El usuario debe constar en nómina y encontrarse dentro del horario permitido para su rol.
<p>Flujo Normal:</p> <ul style="list-style-type: none"> • Ingresar a la opción Parámetros – Presupuestos – Periodo Presupuesto <ul style="list-style-type: none"> ○ Se muestran registros previos en el caso de existir • Pulsar el botón Agregar <ul style="list-style-type: none"> ○ Aparece un recuadro inferior para el ingreso de datos ○ Pulsar el botón Guardar

- Pulsar el botón Editar
 - Aparece un recuadro inferior para la actualización de datos
 - Pulsar el botón Guardar

Flujo Alternativo:

- El sistema verifica que los campos ingresados sean válidos.
 - Si los datos son incorrectos muestra un mensaje de error.

Pos condiciones:

- Registros almacenados, actualizados, consultados.

Tabla 4.6 Especificación Selección de cuentas

Caso de Uso: Selección de cuentas
<p>Descripción:</p> <ul style="list-style-type: none"> • Consultar, definir, eliminar cuentas contables a intervenir. <p>Actores:</p> <ul style="list-style-type: none"> • Usuario con el rol: Contador General. <p>Precondiciones:</p> <ul style="list-style-type: none"> • El usuario debe constar en nómina y encontrarse dentro del horario permitido para su rol. <p>Flujo Normal:</p> <ul style="list-style-type: none"> • Ingresar a la opción Parámetros – Presupuestos – Periodo Presupuesto <ul style="list-style-type: none"> ○ Pulsar el botón Plan Contable ○ En la parte inferior se muestra el plan de cuentas asociado al periodo presupuestario activo, en el caso de no aún no definirlo aparecerá el plan de cuentas completo • Las cuentas contables que no sean eliminadas serán las que se definan <ul style="list-style-type: none"> ○ Pulsar el botón Guardar • Pulsar el botón Eliminar <ul style="list-style-type: none"> ○ Pulsar el botón Guardar <p>Flujo Alternativo:</p>

- El sistema verifica que los campos ingresados sean válidos.
 - Si los datos son incorrectos muestra un mensaje de error.

Pos condiciones:

- Registros definidos, descartados, consultados.

Tabla 4.7 Especificación Porcentajes de distribución

Caso de Uso: Porcentajes de distribución
<p>Descripción:</p> <ul style="list-style-type: none"> • Definir, actualizar valores trimestrales de cumplimiento. <p>Actores:</p> <ul style="list-style-type: none"> • Usuario con el rol: Contador General. <p>Precondiciones:</p> <ul style="list-style-type: none"> • El usuario debe constar en nómina y encontrarse dentro del horario permitido para su rol. <p>Flujo Normal:</p> <ul style="list-style-type: none"> • Ingresar a la opción Parámetros – Presupuestos – Periodo Presupuesto <ul style="list-style-type: none"> ○ Pulsar el botón Distribución ○ Aparece un recuadro inferior para el ingreso de datos ○ Pulsar el botón Guardar • Pulsar el botón Distribución <ul style="list-style-type: none"> ○ Aparece un recuadro inferior para la actualización de datos ○ Pulsar el botón Guardar <p>Flujo Alternativo:</p> <ul style="list-style-type: none"> • El sistema verifica que los campos ingresados sean válidos. <ul style="list-style-type: none"> ○ Si los datos son incorrectos muestra un mensaje de error. <p>Pos condiciones:</p> <ul style="list-style-type: none"> • Registros almacenados, actualizados.

Tabla 4.8 Especificación Proyección

Caso de Uso: Proyecciones
<p>Descripción:</p> <ul style="list-style-type: none"> • Proyección de los valores de las cuentas contables. <p>Actores:</p> <ul style="list-style-type: none"> • Usuario con el rol: Contador General. <p>Precondiciones:</p> <ul style="list-style-type: none"> • El usuario debe constar en nómina y encontrarse dentro del horario permitido para su rol. <p>Flujo Normal:</p> <ul style="list-style-type: none"> • Ingresar a la opción Presupuestos – Indicadores Cuentas <ul style="list-style-type: none"> ○ Se listan para selección individual las cuentas contables definidas ○ Definir las fechas de corte ○ Escoger método de proyección ○ Pulsar el botón Ejecutar ○ Pulsar el botón Guardar <p>Flujo Alternativo:</p> <ul style="list-style-type: none"> • El sistema verifica que los campos ingresados sean válidos. <ul style="list-style-type: none"> ○ Si los datos son incorrectos muestra un mensaje de error. <p>Pos condiciones:</p> <ul style="list-style-type: none"> • Registros almacenados.

Tabla 4.9 Especificación Visualización de saldos contables

Caso de Uso: Visualización de saldos contables
<p>Descripción:</p> <ul style="list-style-type: none"> • Consultar valores previos de cuentas contables. <p>Actores:</p> <ul style="list-style-type: none"> • Usuario con el rol: Contador General. <p>Precondiciones:</p>

- El usuario debe constar en nómina y encontrarse dentro del horario permitido para su rol.

Flujo Normal:

- Ingresar a la opción Presupuestos – Indicadores Cuentas
 - Pulsar el botón Ejecutar
 - Aparecerán todas las sucursales pertenecientes a la entidad
 - Pulsar sobre el nombre
 - Aparecerá la gráfica con los valores a consultar

Flujo Alternativo:

- El sistema verifica que existan los datos previos solicitados.

Pos condiciones:

- Registros consultados.

Tabla 4.10 Especificación Mayorización

Caso de Uso: Mayorización
<p>Descripción:</p> <ul style="list-style-type: none"> • Asignación de saldos contables a sus respectivas cuentas superiores. <p>Actores:</p> <ul style="list-style-type: none"> • Usuario con el rol: Contador General. <p>Precondiciones:</p> <ul style="list-style-type: none"> • El usuario debe constar en nómina y encontrarse dentro del horario permitido para su rol. <p>Flujo Normal:</p> <ul style="list-style-type: none"> • Ingresar a la opción Presupuestos – Mayorización Cuentas <ul style="list-style-type: none"> ○ Pulsar el botón Ejecutar <p>Flujo Alternativo:</p> <ul style="list-style-type: none"> • El sistema verifica que no existan un proceso previo de mayorización. <p>Pos condiciones:</p> <ul style="list-style-type: none"> • Registros almacenados.

Tabla 4.11 Especificación Presentación de balances

Caso de Uso: Presentación de balances
<p>Descripción:</p> <ul style="list-style-type: none"> • Presentación de los datos proyectados <p>Actores:</p> <ul style="list-style-type: none"> • Usuario con el rol: Gerente General, Contador General. <p>Precondiciones:</p> <ul style="list-style-type: none"> • El usuario debe constar en nómina y encontrarse dentro del horario permitido para su rol. <p>Flujo Normal:</p> <ul style="list-style-type: none"> • Ingresar a la opción Presupuestos – Reporte Balance General <ul style="list-style-type: none"> ○ Pulsar el botón Ejecutar • Ingresar a la opción Presupuestos – Reporte Pérdidas y Ganancias <ul style="list-style-type: none"> ○ Pulsar el botón Ejecutar <p>Flujo Alternativo:</p> <ul style="list-style-type: none"> • El sistema verifica que existan los datos previos solicitados. <p>Pos condiciones:</p> <ul style="list-style-type: none"> • Registros consultados.

Tabla 4.12 Especificación Comparativa de saldos contables

Caso de Uso: Comparativa de saldos contables
<p>Descripción:</p> <ul style="list-style-type: none"> • Presentación de los valores proyectados para cada trimestre junto con los valores reales. <p>Actores:</p> <ul style="list-style-type: none"> • Usuario con el rol: Gerente General, Contador General. <p>Precondiciones:</p> <ul style="list-style-type: none"> • El usuario debe constar en nómina y encontrarse dentro del horario permitido para su rol. <p>Flujo Normal:</p>

- Ingresar a la opción Reportes
 - Aparecerá una lista de los módulos del sistema
 - Seleccionar Presupuestos
 - Aparecerán los reportes disponibles para el módulo elegido
 - Seleccionar Comparación Presupuestaria
 - Pulsar el botón Ejecutar
 - Aparecerá el reporte comparativo

Flujo Alternativo:

- El sistema verifica que existan los datos previos solicitados.

Pos condiciones:

- Registros consultados.

4.2. Diseño y Arquitectura

El lenguaje UML es un estándar OMG (Object Management Group) diseñado para visualizar, especificar, construir y documentar software orientado a objetos [28].

Este estándar de representación de esquemas nos permite mostrar información específica de cómo un sistema trabaja, o en este caso, de cómo se desea que trabaje. Los diagramas UML utilizados en el diseño del módulo son: casos de uso, actividades, clases, secuencias, estados. Además, se especifica el diseño de arquitectura y diseño de interfaces.

4.2.1. Elaboración de diagrama de clases

El siguiente diagrama describe la estructura del módulo a nivel de sus clases, atributos y como se relacionan unas con otras.

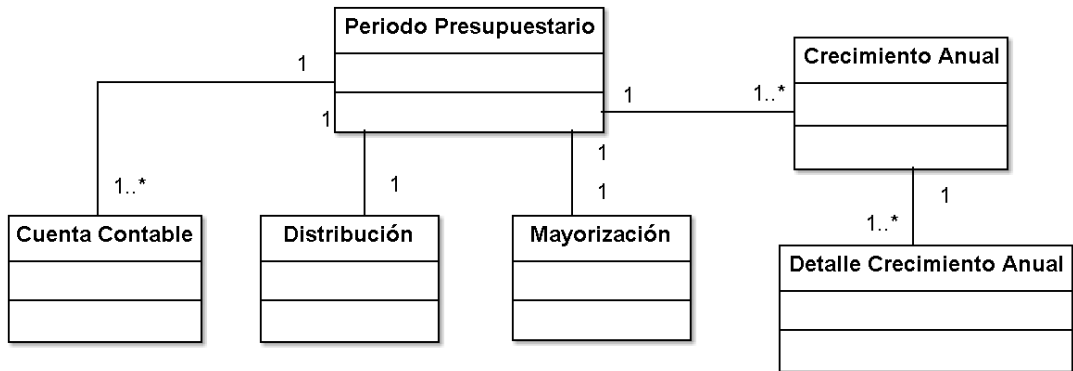


Fig. 10 Diagrama de clases: modelo conceptual

Diagrama detallado

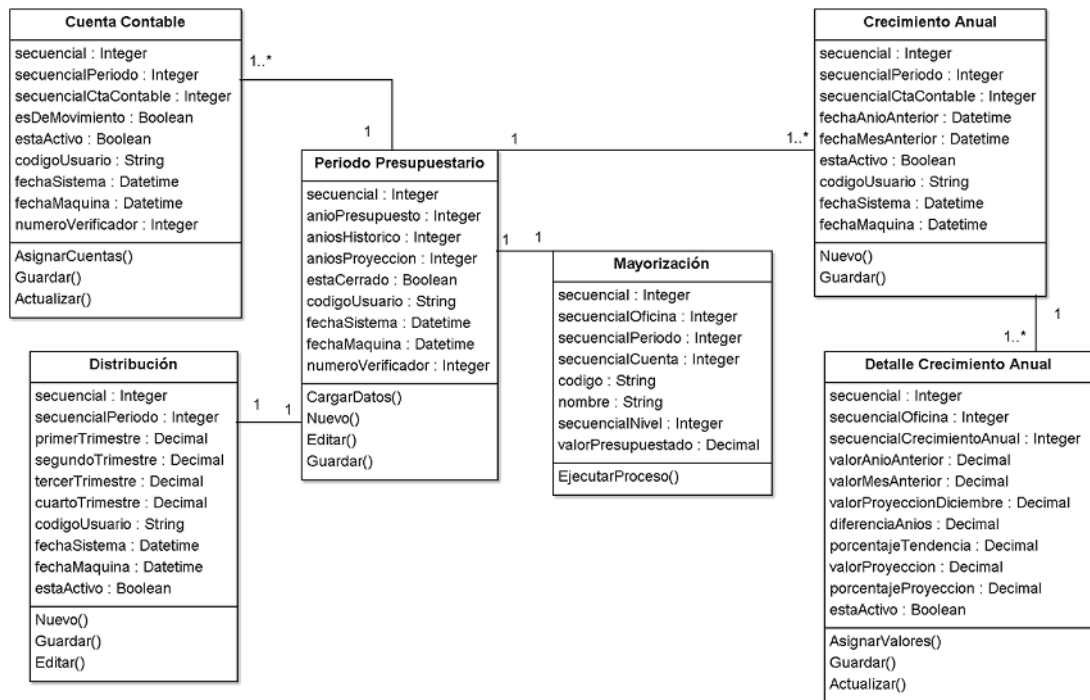


Fig. 11 Diagrama de clases - Módulo de presupuesto

4.2.2. Elaboración de diagrama de secuencias

El diagrama de secuencia representa cada una de las interacciones de un objeto en un sistema a través del tiempo, se incluye la forma en cómo viaja la información hasta llegar a ser presentada visualmente al usuario final, además, muestra la relación con cada una de las clases a implementar.

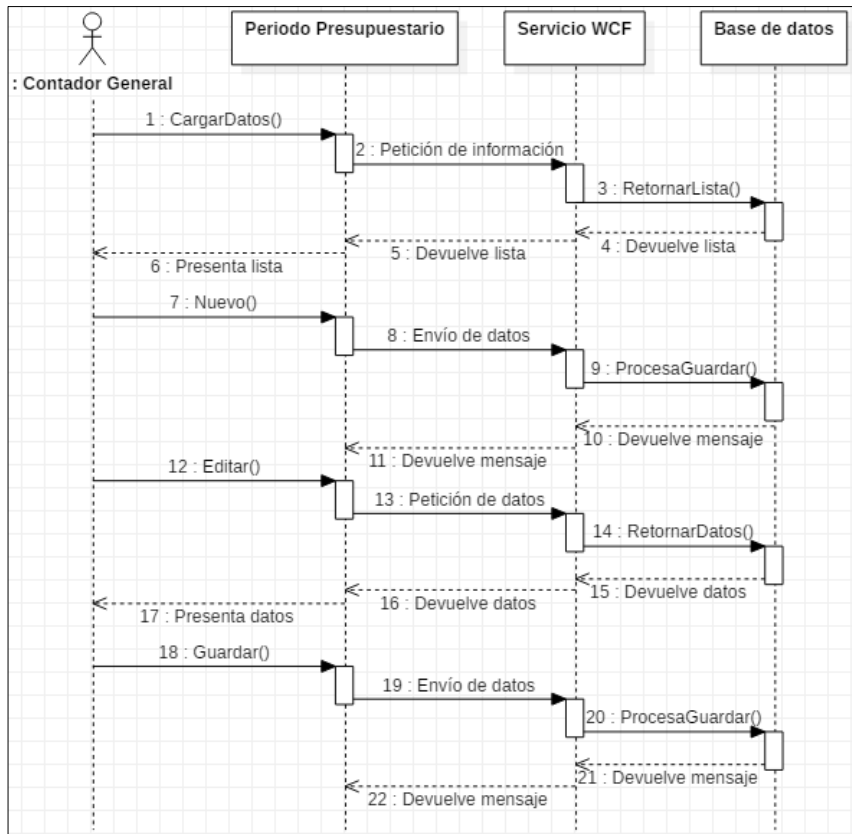


Fig. 12 Secuencia Parametrización

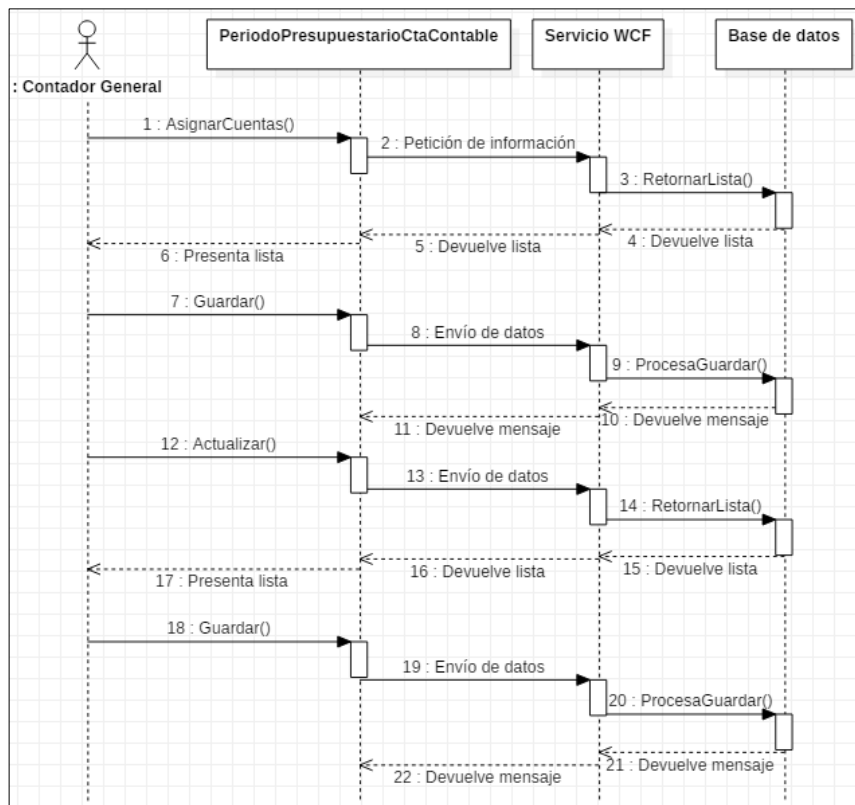


Fig. 13 Secuencia Selección de Cuentas

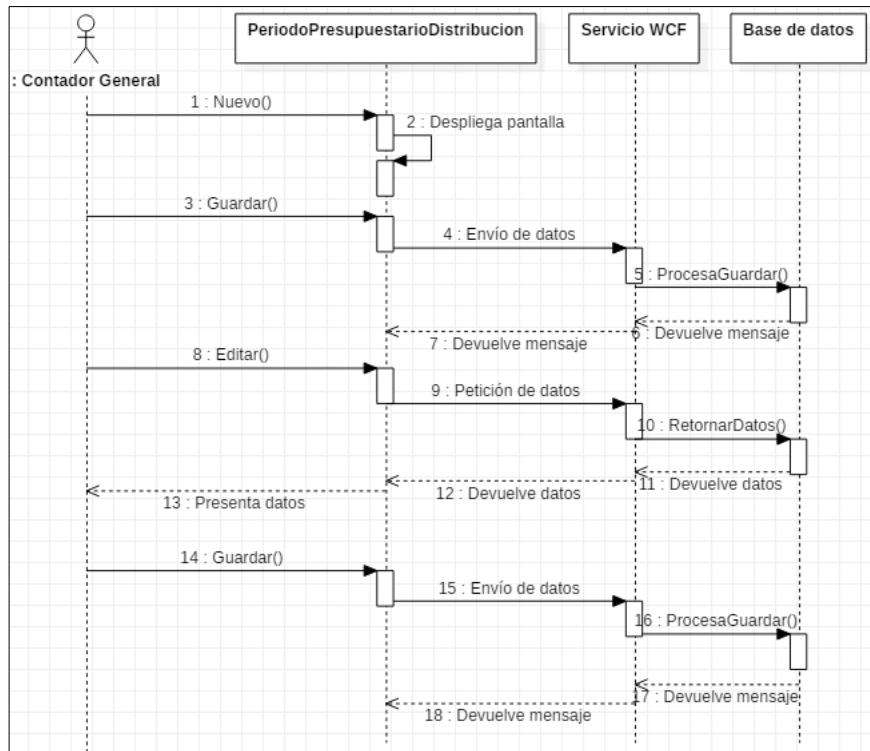


Fig. 14 Secuencia Porcentajes de Distribución

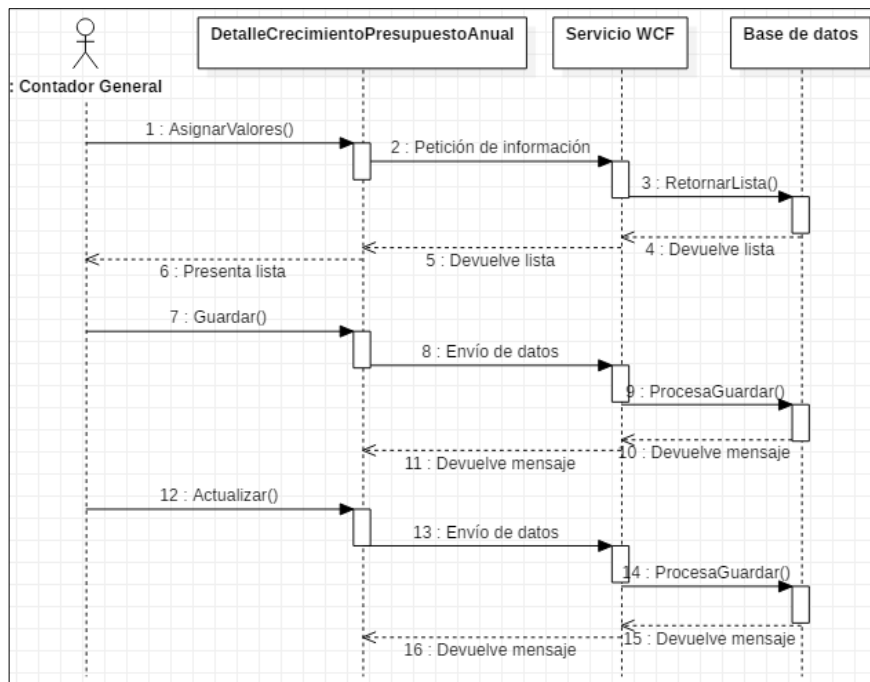


Fig. 15 Secuencia Proyecciones

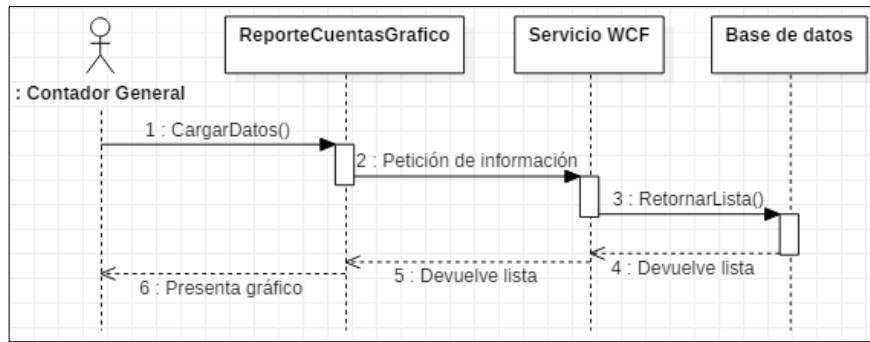


Fig. 16 Secuencia Visualización de saldos contables

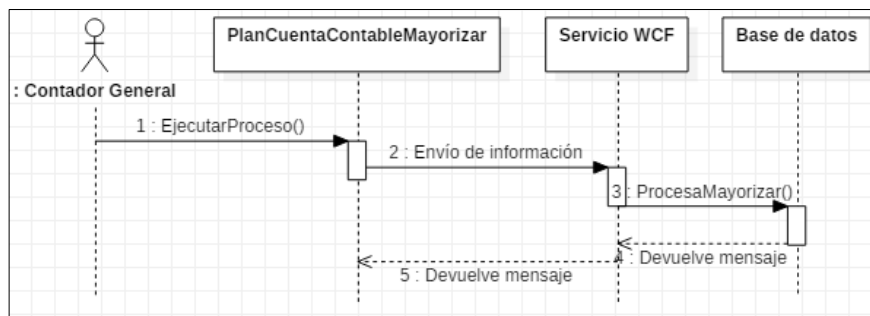


Fig. 17 Secuencia Mayorización

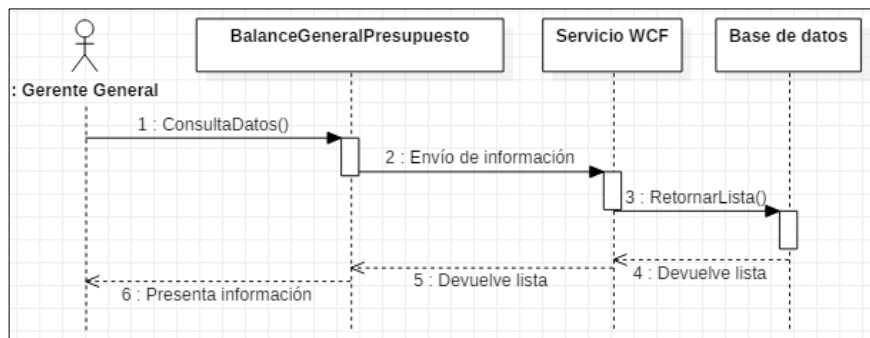


Fig. 18 Secuencia Presentación de balances

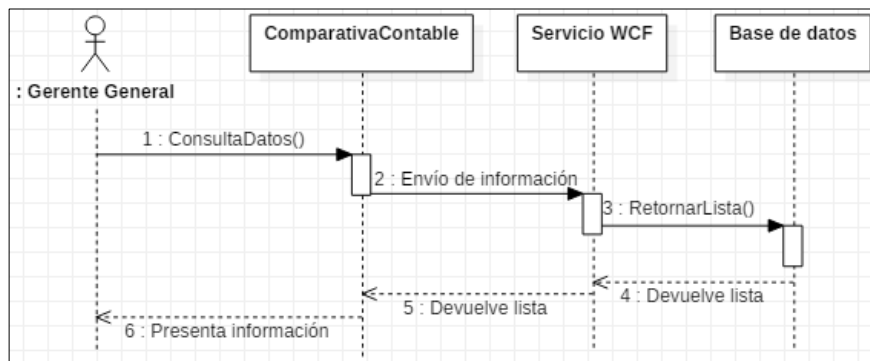


Fig. 19 Secuencia Comparativa de saldos contables

4.2.3. Elaboración de diagrama de estado

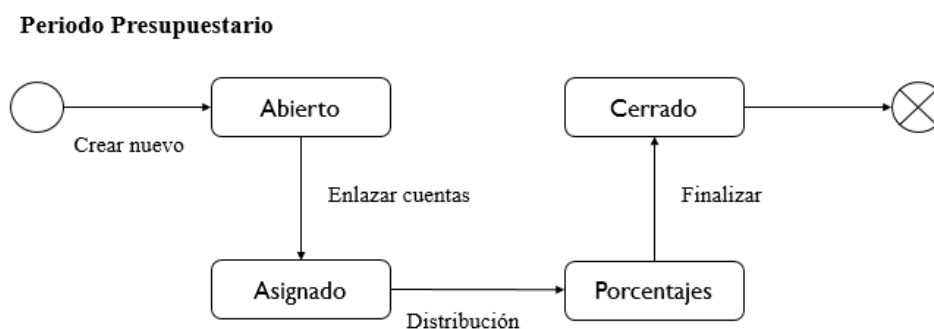


Fig. 20 Diagrama de estado - Parametrización

4.2.4. Diseño de base de datos

- Esquema: FBS_PRESUPUESTOS

Tabla 4.13 PERIODOPRESUPUESTARIO

Campo	Tipo de Dato	Descripción
SECUENCIAL	int	Contiene el secuencial del registro
ANIOPRESUPUESTO	int	Año en el que se realiza el presupuesto
ANIOSHISTORICO	int	Número de años anteriores a considerar
ANIOSPROYECCION	int	Número de años a proyectar
ESTACERRADO	bit	Campo de estado que indica el presupuesto activo
CODIGOUSUARIO	nvarchar(25)	Almacena el código del usuario que guarda la información
FECHASISTEMA	datetime	Fecha del sistema al momento del almacenamiento
FECHAMAQUINA	datetime	Fecha de la máquina al momento del almacenamiento
NUMEROVERIFICADOR	int	Número de veces que la información ha sido modificada

Tabla 4.14 PERIODOPRESUPUESTARIOCTACONTABLE

Campo	Tipo de Dato	Descripción
SECUENCIAL	int	Contiene el secuencial del registro
SECUENCIALPERIODO	int	Campo relacionado con tabla PeriodoPresupuestario
SECUENCIALCTACONTABLE	int	Secuencial de la cuenta contable
ESDEMOVIMIENTO	bit	Indica el tipo de cuenta contable
ESTAACTIVO	bit	Campo de estado que indica si la cuenta es considerada
CODIGOUSUARIO	nvarchar(25)	Almacena el código del usuario que guarda la información
FECHASISTEMA	datetime	Fecha del sistema al momento del almacenamiento
FECHAMAQUINA	datetime	Fecha de la máquina al momento del almacenamiento
NUMEROVERIFICADOR	int	Número de veces que la información ha sido modificada

Tabla 4.15 PERIODOPRESUPUESTARIODISTRIBUCION

Campo	Tipo de Dato	Descripción
SECUENCIAL	int	Contiene el secuencial del registro
SECUENCIALPERIODOPRESUPUESTARIO	int	Campo relación con tabla PeriodoPresupuesto
PRIMERTRIMESTRE	decimal(4,2)	Representa el porcentaje de cumplimiento
SEGUNDOTRIMESTRE	decimal(4,2)	Representa el porcentaje de cumplimiento
TERCERTRIMESTRE	decimal(4,2)	Representa el porcentaje de cumplimiento
CUARTOTRIMESTRE	decimal(4,2)	Representa el porcentaje de cumplimiento
CODIGOUSUARIO	nvarchar(25)	Código del usuario que guarda la información
FECHASISTEMA	datetime	Fecha del sistema al almacenar los datos
FECHAMAQUINA	datetime	Fecha de la máquina al almacenar los datos
ESTAATIVO	bit	Indica si los valores ingresados se consideran

Tabla 4.16 CRECIMIENTOPRESUPUESTOANUAL

Campo	Tipo de Dato	Descripción
SECUENCIAL	int	Contiene el secuencial del registro
SECUENCIALPERIODOPRESUPUESTARIO	int	Campo relación con tabla PeriodoPresupuesto
SECUENCIALCTACONTABLE	int	Secuencial de la cuenta contable
FECHAANIOANTERIOR	datetime	Fecha corte a año anterior
FECHAMESANTERIOR	datetime	Fecha corte a mes anterior
ESTAATIVO	bit	Indica si los valores serán considerados
CODIGOUSUARIO	nvarchar(25)	Código del usuario que guarda la información
FECHASISTEMA	datetime	Fecha del sistema al almacenar los datos
FECHAMAQUINA	datetime	Fecha de la máquina al almacenar los datos

Tabla 4.17 DETALLECRECIMIENTOPRESUPUESTOANUAL

Campo	Tipo de Dato	Descripción
SECUENCIAL	int	Contiene el secuencial del registro
SECUENCIALOFICINA	int	Secuencial de sucursal u oficina
SECUENCIALCRECIMIENTOPRESUPUESTOANUAL	int	Secuencial de relación con cabecera
VALORANIOANTERIOR	decimal(18,2)	Saldo de la cuenta a año anterior
VALORMESANTERIOR	decimal(18,2)	Saldo de la cuenta a mes anterior
VALORPROYECCIONDICIEMBRE	decimal(18,2)	Valor calculado estáticamente
DIFERENCIAANIOS	decimal(18,2)	Diferencia diciembre - año anterior
PORCENTAJETENDENCIA	decimal(18,2)	Indica la tendencia de la cuenta
VALORPROYECCION	decimal(18,2)	Valor proyectado en dólares
PORCENTAJEPROYECCION	decimal(18,2)	Porcentaje de variación entre años
ESTAATIVO	bit	Indica si los valores son considerados

4.2.5. Diagrama de despliegue

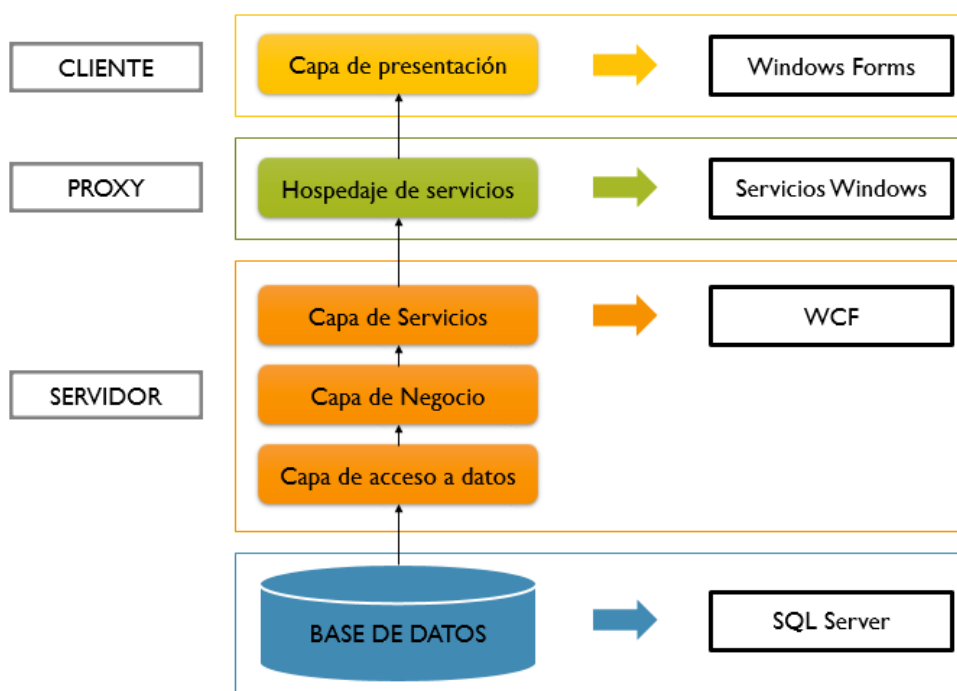


Fig. 22 Diagrama de Arquitectura – Módulo de presupuesto

El diagrama representa en forma gráfica la manera en cómo el módulo va a ser construido, su objetivo principal es el de modelar la arquitectura que tiene un software en tiempo de ejecución.

En la base de datos se encuentran las tablas que contendrán información de las cuentas a intervenir en el proceso, más todos los otros datos que el usuario requiera para su entendimiento.

La fase Servidor contiene toda la abstracción del acceso a los datos y se define los diferentes comportamientos que las entidades y objetos pueden tener para cada necesidad del módulo.

La fase Proxy aloja los servicios creados por el servidor y lanzados mediante una interfaz para que puedan ser accedidos y consumidos desde el cliente.

La fase Cliente representa la parte visual que interactúa directamente con el usuario final, en él se hace el llamado de los métodos para poder comunicarse con objetos y datos y presentarlos mediante petición.

4.2.6. Diseño de interfaces

El contenido presentado al usuario final debe estar adaptado al sistema financiero de la cooperativa, por lo cual las pantallas se estructuraron de la siguiente manera:

- Parametrización

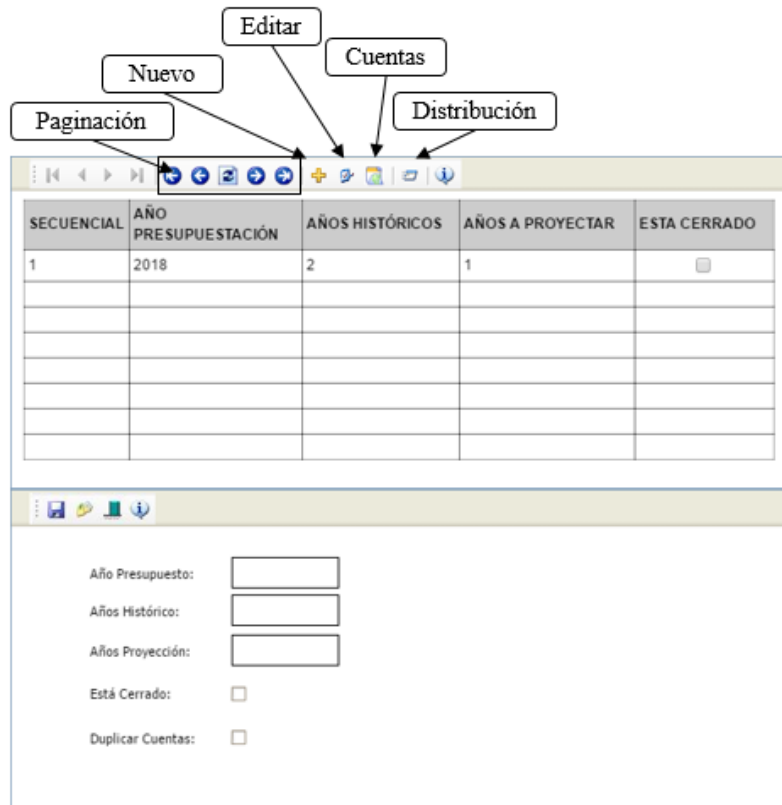


Fig. 23 Diseño de Interfaces – Parametrización

- Selección de Cuentas

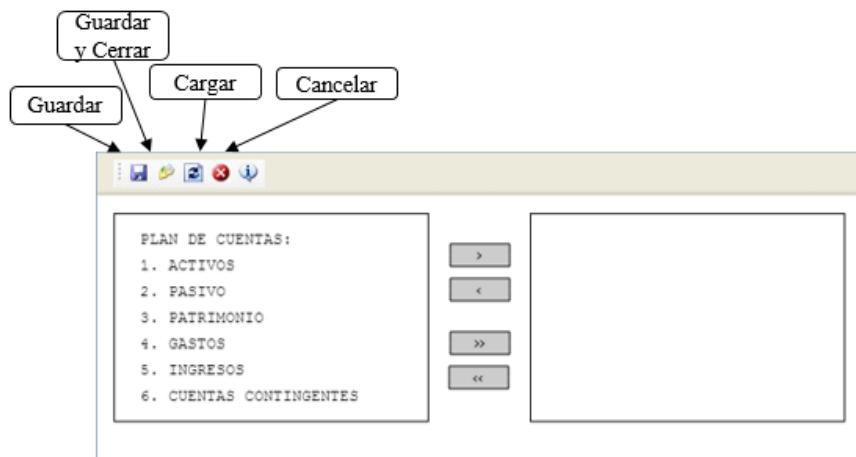


Fig. 24 Diseño de interfaces - Selección de cuentas

- **Porcentajes de distribución**

SECUENCIAL	AÑO PRESUPUESTACIÓN	AÑOS HISTÓRICOS	AÑOS A PROYECTAR	ESTA CERRADO
1	2018	2	1	<input type="checkbox"/>

Primer Trimestre:
 Segundo Trimestre:
 Tercer Trimestre:
 Cuarto Trimestre:

Fig. 25 Diseño de interfaces - Porcentajes de distribución

- **Proyección de valores**

Ejecutar
 Buscar Gráfico

Oficina: TODOS Cuenta: 11010501 - CAJA GENERAL

Fecha Año Anterior: 31/12/2017 Fecha Mes Anterior: 31/12/2017

Fijar Porcentaje Método de Regresión

OFICINA	AÑO ANTERIOR	MES ANTERIOR	PROYECCIÓN DICIEMBRE	DIFERENCIA AÑOS	% TENDENCIA	VALOR PROYECCIÓN	% PROYECCIÓN
MATRIZ	224,877.97	246,985.83	987,943.32	763,065.35	339.32	0.00	0.00
SALCEDO	106,284.11	114,642.19	458,568.76	352,284.65	331.46	0.00	0.00
PUYO	175,472.42	100,793.17	403,172.68	227,700.26	129.76	0.00	0.00
PILLARO	89,858.74	151,049.69	604,198.76	514,340.02	572.39	0.00	0.00
IZAMBA	83,189.80	33,131.60	132,526.40	49,336.60	59.31	0.00	0.00
MACAS	45,236.62	58,831.85	235,327.40	190,090.78	420.21	0.00	0.00
PELILEO	163,358.55	72,993.79	291,975.16	128,616.61	78.73	0.00	0.00
TENA	60,847.51	73,785.25	295,141.00	234,293.49	385.05	0.00	0.00
LATACUNGA	36,010.26	49,133.97	196,535.88	160,525.62	445.78	0.00	0.00
OFICINA SUR	95,917.50	95,246.72	380,986.88	285,069.38	297.20	0.00	0.00
CEVALLOS	20,662.80	43,143.29	172,573.16	151,910.36	735.19	0.00	0.00
TISALEO	19,353.89	33,422.35	133,689.40	114,335.51	590.76	0.00	0.00
SAQUISILI	18,277.87	18,496.71	73,986.84	55,708.97	304.79	0.00	0.00
EL QUINCHE	37,657.93	42,832.54	171,330.16	133,672.23	354.96	0.00	0.00

Fig. 26 Diseño de interfaces – Proyección de valores

- **Mayorización de Cuentas**

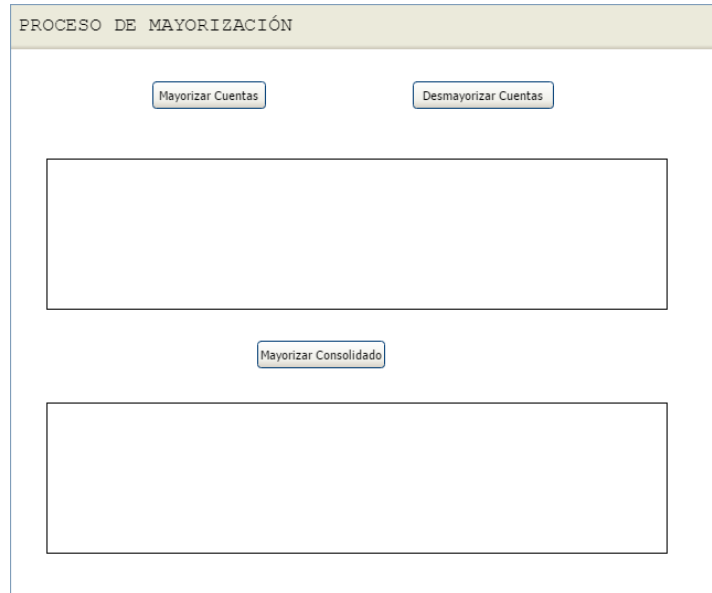


Fig. 27 Diseño de interfaces - Mayorización de cuentas

- **Balance General**

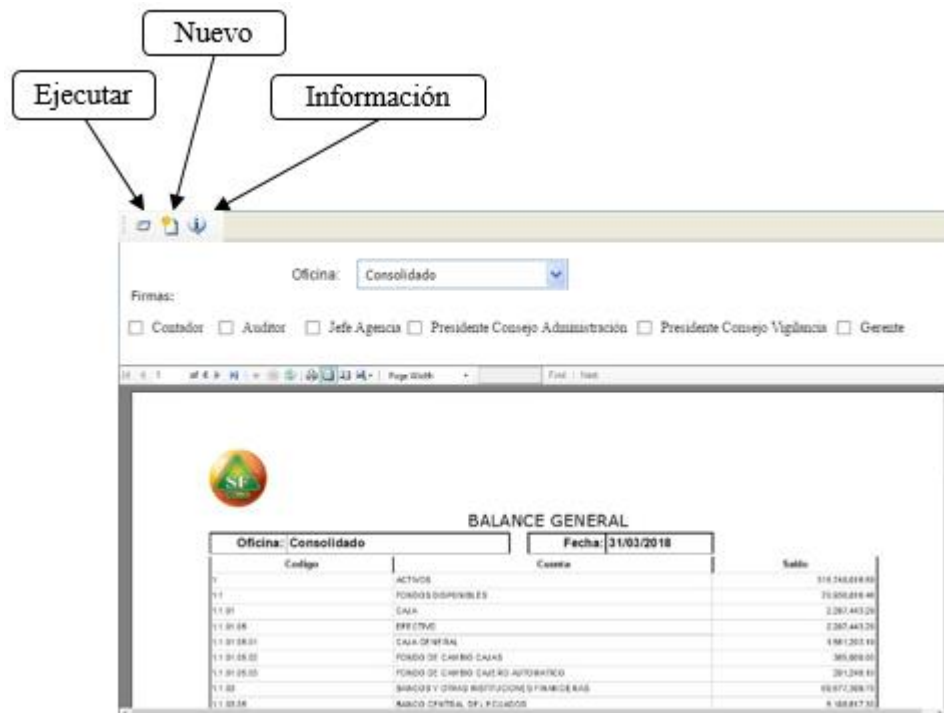


Fig. 28 Diseño de interfaces - Balance General

- Estado de Pérdidas y Ganancias

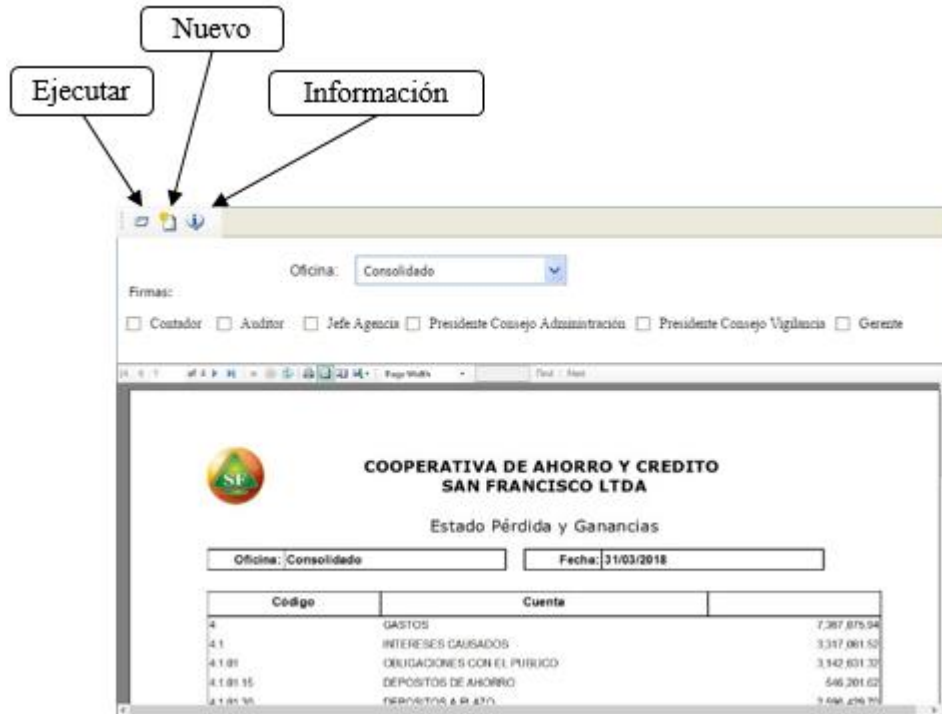


Fig. 29 Diseño de interfaces - Estado de Pérdidas y Ganancias

- Comparativa de saldos



Fig. 30 Diseño de interfaces - Comparativa de saldos

4.3. Programación

4.3.1. Desarrollo del sistema

La arquitectura a nivel de código del módulo de presupuesto financiero presenta la siguiente estructura:

La solución **FBSServidor** contiene los siguientes proyectos:

- **FBSCore**: Es el proyecto que se encarga de realizar la comunicación con la base de datos, el proyecto está dividido en las siguientes carpetas:
 - **DALC**: Son las clases en las que se encuentra la interacción directa con la base de datos.
 - **Entidad**: Es la conversión del modelo de base de datos a clases.
 - **Actor**: Son las acciones básicas que se ejecutan sobre la base de datos en función de clases, por ejemplo: guardar, actualizar, borrar, listar.
 - **Actor Negocio**: Son las acciones específicas que el desarrollador necesita ejecutar sobre la base de datos, en conforme a los requerimientos.
- **FBSMensajería**: Contiene las clases que sirven para intercambiar información entre el servidor, el proxy y el cliente.
- **FBSServicios**: En él se encuentra alojada toda la lógica de negocio el mismo que contiene los web services y las interfaces respectivas para cada uno.
 - **Interfaces**: Son los métodos que se exponen para ser consumidos.

La solución **FBSProxies** se encarga de realizar el vínculo con el servidor a través de las referencias web y las transforma en .dll

En la solución **FBSCliente** se encuentran todas las interfaces gráficas del aplicativo financiero, para lo cual existe una vista donde se incrustan todos los componentes gráficos y un controlador que es una clase que maneja todos los eventos que se ejecutan sobre la vista.

La arquitectura lógica de programación utilizada para la implementación del módulo consiste en 5 capas:

- Capa de Objetos de Negocio.

- Capa de Acceso a Datos.
- Capa de Servicios WCF.
- Capa de Proxies
- Capa de Presentación.

Las clases públicas contenidas en la capa de objetos de negocio representan cada una de las tablas del módulo de presupuesto almacenadas en la base de datos. En esta clase también constan las propiedades de cada campo.

FRAGMENTO DE CÓDIGO

Fuente: DetalleCrecimientoPresupuestoAnual_Entidad.cs

```
public partial class DetalleCrecimientoPresupuestoAnual:ClaseBase
{
    /// <summary>
    /// Atributo privado de secuencial
    /// </summary>
    private int _secuencial;

    /// <summary>
    /// Atributo privado de valorAnioAnterior
    /// </summary>
    private decimal _valorAnioAnterior;

    /// <summary>
    /// Atributo privado de valorMesAnterior
    /// </summary>
    private decimal _valorMesAnterior;

    /// <summary>
    /// Atributo privado de valorProyeccionDiciembre
    /// </summary>
    private decimal _valorProyeccionDiciembre;

    /// <summary>
    /// Atributo privado de diferenciaAnios
    /// </summary>
    private decimal _diferenciaAnios;

    /// <summary>
    /// Atributo privado de porcentajeTendencia
    /// </summary>
    private decimal _porcentajeTendencia;

    /// <summary>
    /// Atributo privado de valorProyeccion
    /// </summary>
    private decimal _valorProyeccion;

    /// <summary>
    /// Atributo privado de porcentajeProyeccion
    /// </summary>
    private decimal _porcentajeProyeccion;
}
```

```

    /// <summary>
    /// Atributo privado de estaActivo
    /// </summary>
    private bool _estaActivo;

    //Atributos Foraneos
    /// <summary>
    /// CAMPO SECUENCIALOFICINA CREADO AUTOMATICAMENTE
    /// </summary>
    private int _secuencialOficina;

    /// <summary>
    /// CAMPO SECUENCIALCRECIMIENTOPRESUPUESTOANUAL CREADO AUTOMATICAMENTE
    /// </summary>
    private int _secuencialCrecimientoPresupuestoAnual;

    /// <summary>
    /// Atributo privado de la entidad Oficina
    /// </summary>
    private Oficina _oficina;

    /// <summary>
    /// Atributo privado de la entidad CrecimientoPresupuestoAnual
    /// </summary>
    private CrecimientoPresupuestoAnual _crecimientoPresupuestoAnual;

    #region Propiedades

    /// <summary>
    /// CAMPO SECUENCIAL
    /// </summary>
    public int Secuencial
    {
        get { return this._secuencial; }
    }

    /// <summary>
    /// CAMPO VALORANIOANTERIOR
    /// </summary>
    public decimal ValorAnioAnterior
    {
        get { return this._valorAnioAnterior; }
        set { this._valorAnioAnterior=value; }
    }

    /// <summary>
    /// CAMPO VALORMESANTERIOR
    /// </summary>
    public decimal ValorMesAnterior
    {
        get { return this._valorMesAnterior; }
        set { this._valorMesAnterior=value; }
    }

    /// <summary>
    /// CAMPO VALORPROYECCIONDICIEMBRE
    /// </summary>
    public decimal ValorProyeccionDiciembre
    {
        get { return this._valorProyeccionDiciembre; }
        set { this._valorProyeccionDiciembre=value; }
    }

```

```

        /// <summary>
        /// CAMPO DIFERENCIAANIOS
        /// </summary>
        public decimal DiferenciaAnios
        {
            get { return this._diferenciaAnios; }
            set { this._diferenciaAnios=value; }
        }

        /// <summary>
        /// CAMPO PORCENTAJETENDENCIA
        /// </summary>
        public decimal PorcentajeTendencia
        {
            get { return this._porcentajeTendencia; }
            set { this._porcentajeTendencia=value; }
        }

        /// <summary>
        /// CAMPO VALORPROYECCION
        /// </summary>
        public decimal ValorProyeccion
        {
            get { return this._valorProyeccion; }
            set { this._valorProyeccion=value; }
        }

        /// <summary>
        /// CAMPO PORCENTAJEPROYECCION
        /// </summary>
        public decimal PorcentajeProyeccion
        {
            get { return this._porcentajeProyeccion; }
            set { this._porcentajeProyeccion=value; }
        }

        /// <summary>
        /// CAMPO ESTAACTIVO
        /// </summary>
        public bool EstaActivo
        {
            get { return this._estaActivo; }
            set { this._estaActivo=value; }
        }

        #endregion
    }
}

```

La capa de acceso a datos contiene todos los métodos necesarios de consulta, inserción, actualización y eliminación de la información que contiene el módulo de presupuesto financiero, los cuales permiten interactuar con la base de datos, a través de las clases creadas en la capa de objetos del negocio.

FRAGMENTO DE CÓDIGO

Fuente: DetalleCrecimientoPresupuestoAnual_Actor.cs

```
/// <summary>
    /// Devuelve una entidad DetalleCrecimientoPresupuestoAnual por
    Secuencial sin verificar existencia
    /// </summary>
    /// <param name="secuencial">CAMPO SECUENCIAL CREADO
    AUTOMATICAMENTE</param>
    /// <param name="soloActivo">Solo si el registro esta activo</param>
    /// <returns>DetalleCrecimientoPresupuestoAnual</returns>
    public static DetalleCrecimientoPresupuestoAnual
    DamePorSecuencialSinVerificarExistencia(int secuencial, bool soloActivo)
    {
        DetalleCrecimientoPresupuestoAnual
    detalleCrecimientoPresupuestoAnual=null;
        try
        {
            detalleCrecimientoPresupuestoAnual =
    DetalleCrecimientoPresupuestoAnualDALC.DamePorSecuencialSinVerificarExistencia(secu
    encial);
            if(detalleCrecimientoPresupuestoAnual==null)
                return detalleCrecimientoPresupuestoAnual;
            if(soloActivo &&
    !detalleCrecimientoPresupuestoAnual.EstaActivo)
                throw new
    ActorException("G00008|DetalleCrecimientoPresupuestoAnual no se encuentra
    activo(a)");
        }
        catch(Exception ex)
        {
            if (ExceptionPolicy.HandleException(ex, "Politica
    Actor"))
                throw;
        }
        return detalleCrecimientoPresupuestoAnual;
    }
}
```

Mensajería:

```
[DataContract]
public class DetalleCrecimientoPresupuestoAnualMSE
{
    private int _secuencial;
    private int _secuencialOficina;
    private int _secuencialCrecimientoPresupuestoAnual;
    private decimal _valorAnioAnterior;
    private decimal _valorMesAnterior;
    private decimal _valorProyeccionDiciembre;
    private decimal _diferenciaAnios;
    private decimal _porcentajeTendencia;
    private decimal _valorProyeccion;
    private decimal _porcentajeProyeccion;
    private bool _estaActivo;

    [DataMember]
    public int Secuencial
    {
        get { return this._secuencial; }
    }
}
```

```

        set { this._secuencial = value; }
    }
    [DataMember]
    public int SecuencialOficina
    {
        get { return this._secuencialOficina; }
        set { this._secuencialOficina = value; }
    }
    [DataMember]
    public int SecuencialCrecimientoPresupuestoAnual
    {
        get { return this._secuencialCrecimientoPresupuestoAnual; }
        set { this._secuencialCrecimientoPresupuestoAnual = value; }
    }
    [DataMember]
    public decimal ValorAnioAnterior
    {
        get { return this._valorAnioAnterior; }
        set { this._valorAnioAnterior = value; }
    }
    [DataMember]
    public decimal ValorMesAnterior
    {
        get { return this._valorMesAnterior; }
        set { this._valorMesAnterior = value; }
    }
    [DataMember]
    public decimal ValorProyeccionDiciembre
    {
        get { return this._valorProyeccionDiciembre; }
        set { this._valorProyeccionDiciembre = value; }
    }
    [DataMember]
    public decimal DiferenciaAnios
    {
        get { return this._diferenciaAnios; }
        set { this._diferenciaAnios = value; }
    }
    [DataMember]
    public decimal PorcentajeTendencia
    {
        get { return this._porcentajeTendencia; }
        set { this._porcentajeTendencia = value; }
    }
    [DataMember]
    public decimal ValorProyeccion
    {
        get { return this._valorProyeccion; }
        set { this._valorProyeccion = value; }
    }
    [DataMember]
    public decimal PorcentajeProyeccion
    {
        get { return this._porcentajeProyeccion; }
        set { this._porcentajeProyeccion = value; }
    }
    [DataMember]
    public bool EstaActivo
    {
        get { return this._estaActivo; }
    }

```

```

        set { this._estaActivo = value; }
    }

    public DetalleCrecimientoPresupuestoAnualMSE() { }

    public DetalleCrecimientoPresupuestoAnualMSE(
        int secuencial,
        int secuencialOficina,
        int secuencialCrecimientoPresupuestoAnual,
        decimal valorAnioAnterior,
        decimal valorMesAnterior,
        decimal valorProyeccionDiciembre,
        decimal diferenciaAnios,
        decimal porcentajeTendencia,
        decimal valorProyeccion,
        decimal porcentajeProyeccion,
        bool estaActivo)
    {
        this._secuencial = secuencial;
        this._secuencialOficina = secuencialOficina;
        this._secuencialCrecimientoPresupuestoAnual =
secuencialCrecimientoPresupuestoAnual;
        this._valorAnioAnterior = valorAnioAnterior;
        this._valorMesAnterior = valorMesAnterior;
        this._valorProyeccionDiciembre = valorProyeccionDiciembre;
        this._diferenciaAnios = diferenciaAnios;
        this._porcentajeTendencia = porcentajeTendencia;
        this._valorProyeccion = valorProyeccion;
        this._porcentajeProyeccion = porcentajeProyeccion;
        this._estaActivo = estaActivo;
    }
}

```

En la capa de servicios se encuentran los métodos que serán llamados en la capa de presentación, al usar esta capa, se aplica el contrato de servicio para la comunicación, lo que significa que el contenido de un método en esta clase puede cambiar cuantas veces el usuario o el desarrollador lo requieran, sin afectar el llamado en el cliente porque al exponer un servicio mediante una interfaz, se puede acceder a ellos directamente como si se encontraran en la misma solución o en un mismo namespace.

FRAGMENTO DE CÓDIGO

Fuente: DetalleCrecimientoPresupuestoAnualWS.cs

```

//Devuelve los saldos anteriores de la cuenta seleccionada
public ReporteHistoricoCuentasLista
DevuelveInformacionCuentas(InformacionPresupuestoME mensajeEntrada)
{
    ReporteHistoricoCuentasLista mensajeSalida = null;
    try
    {
        List<ReporteHistoricoCuentasMS> listaReporteHistoricoCuentasMS =
new List<ReporteHistoricoCuentasMS>();
    }
}

```

```

        Dictionary<string, ReporteHistoricoCuentasMS> _diccionarioDatos =
new Dictionary<string, ReporteHistoricoCuentasMS>();

        PeriodoPresupuestarioWS periodoPresupuestarioWS = new
PeriodoPresupuestarioWS();
        periodoPresupuestarioWS.settings = settings;

        PeriodoPresupuestarioMSE periodoPresupuestarioMSE =
periodoPresupuestarioWS.DevuelvePeriodoPresupuestarioAbierto();

        if (periodoPresupuestarioMSE.AniosHistorico > 0)
        {
            PlanCuentaContable_Vista cuenta =
PlanCuentaContable_Vista.DameUnoPorSecuencial(mensajeEntrada.SecuencialCuenta);
            if (cuenta.Codigo.Substring(0, 1) == "1" ||
cuenta.Codigo.Substring(0, 1) == "2" || cuenta.Codigo.Substring(0, 1) == "3")
            {
                if (cuenta.Codigo.Substring(0, 1) == "1")
                {
                    for (int i = periodoPresupuestarioMSE.AniosHistorico; i
> 0; i--)
                    {
                        DateTime fecha = new DateTime(DateTime.Now.Year -
i, 12, 31);

                        ReporteCuentas_DataSet.InformacionCuentasDataTable
cuentaAnterior = ReporteCuentasInformacion.DameReporteCuentas(
                            mensajeEntrada.SecuencialCuenta,
                            mensajeEntrada.SecuencialOficina,
                            fecha);

                        ReporteHistoricoCuentasMS reporteHistoricoCuentasMS
= new ReporteHistoricoCuentasMS();
                        reporteHistoricoCuentasMS.SecuencialCuentaContable
= cuentaAnterior[0].secuencialCuentaContable;
                        reporteHistoricoCuentasMS.Codigo =
cuentaAnterior[0].codigo;
                        reporteHistoricoCuentasMS.Nombre =
cuentaAnterior[0].nombre;
                        reporteHistoricoCuentasMS.SecuencialOficina =
cuentaAnterior[0].secuencialOficina;
                        reporteHistoricoCuentasMS.Total =
cuentaAnterior[0].total;
                        reporteHistoricoCuentasMS.Fecha =
cuentaAnterior[0].fechaTotal.Substring(0, 10);

                        listaReporteHistoricoCuentasMS.Add(reporteHistoricoCuentasMS);
                    }
                }
                else
                {
                    for (int i = periodoPresupuestarioMSE.AniosHistorico; i
> 0; i--)
                    {
                        DateTime fecha = new DateTime(DateTime.Now.Year -
i, 12, 31);

                        ReporteCuentas_DataSet.InformacionCuentasDataTable
cuentaAnterior = ReporteCuentasInformacion.DameReporteCuentas(
                            mensajeEntrada.SecuencialCuenta,
                            mensajeEntrada.SecuencialOficina,
                            fecha);
                    }
                }
            }
        }

```

```

ReporteHistoricoCuentasMS reporteHistoricoCuentasMS
= new ReporteHistoricoCuentasMS();
reporteHistoricoCuentasMS.SecuencialCuentaContable
= cuentaAnterior[0].secuencialCuentaContable;
reporteHistoricoCuentasMS.Codigo =
cuentaAnterior[0].codigo;
reporteHistoricoCuentasMS.Nombre =
cuentaAnterior[0].nombre;
reporteHistoricoCuentasMS.SecuencialOficina =
cuentaAnterior[0].secuencialOficina;
reporteHistoricoCuentasMS.Total =
cuentaAnterior[0].total * -1;
reporteHistoricoCuentasMS.Fecha =
cuentaAnterior[0].fechaTotal.Substring(0, 10);

listaReporteHistoricoCuentasMS.Add(reporteHistoricoCuentasMS);
    }
}
//Cuentas 4 y 5
else
{
    if (cuenta.Codigo.Substring(0, 1) == "4")
    {
        for (int i = periodoPresupuestarioMSE.AniosHistorico; i
> 0; i--)
        {
            DateTime fecha = new DateTime(DateTime.Now.Year -
i, 12, 31);

            SaldoPerdidaGanancia_DataSet.SaldoCuentaPGDataTable
cuentaAnterior = SaldoPerdidaGananciaInformacion.DameSaldoPerdidaGanancia(
                mensajeEntrada.SecuencialOficina,
                mensajeEntrada.SecuencialCuenta,
                fecha);

            ReporteHistoricoCuentasMS reporteHistoricoCuentasMS
= new ReporteHistoricoCuentasMS();
            reporteHistoricoCuentasMS.SecuencialOficina =
cuentaAnterior[0].secuencialOficina;
            reporteHistoricoCuentasMS.Total =
cuentaAnterior[0].valor;
            reporteHistoricoCuentasMS.Fecha =
cuentaAnterior[0].fecha.ToString().Substring(0, 10);

            listaReporteHistoricoCuentasMS.Add(reporteHistoricoCuentasMS);
                }
            }
        else
        {
            for (int i = periodoPresupuestarioMSE.AniosHistorico; i
> 0; i--)
            {
                DateTime fecha = new DateTime(DateTime.Now.Year -
i, 12, 31);

                SaldoPerdidaGanancia_DataSet.SaldoCuentaPGDataTable
cuentaAnterior = SaldoPerdidaGananciaInformacion.DameSaldoPerdidaGanancia(
                    mensajeEntrada.SecuencialOficina,
                    mensajeEntrada.SecuencialCuenta,

```



```

        [SoapHeader("settings", typeof(MessageSettings), Direction =
SoapHeaderDirection.In)]
        [FaultContractAttribute(typeof(MessageError), ProtectionLevel =
ProtectionLevel.EncryptAndSign)]
        [OperationContract]
        ValorProyeccionLista DevuelveProyeccionDatos(ProyeccionME mensajeEntrada);
    }

```

El servicio WCF es publicado y alojado en la capa de proxies para que el cliente pueda acceder a ellos mediante una instancia.

FRAGMENTO DE CÓDIGO

Fuente: DetalleCrecimientoPresupuestoAnual.svc

```

<%@ServiceHostLanguage="C#"           Debug="true"ervice="FBS_PROXY.Presupuestos.
DetalleCrecimientoPresupuestoAnual "%>

```

Finalmente, la capa de presentación contiene:

La instancia del servicio WCF que permitirá acceder a los métodos

FRAGMENTO DE CÓDIGO

Fuente: DetalleCrecimientoPresupuestoAnualProxy.cs

```

public class DetalleCrecimientoPresupuestoAnualProxy
{
    public static DetalleCrecimientoPresupuestoAnualWSClient Devuelve()
    {
        EndpointAddress direccion = WCF_Comun.devuelveEndPoint("Presupuestos",
"DetalleCrecimientoPresupuestoAnualWS");
        DetalleCrecimientoPresupuestoAnualWSClient ws = new
DetalleCrecimientoPresupuestoAnualWSClient(FBS_Bindings.Binding(), direccion);

        ws.Endpoint.Behaviors.Add(new CustomBehavior()); MessageSettings
settings = new MessageSettings();
        settings.EquipoCliente = System.Net.Dns.GetHostName();
        settings.FechaSistemaCliente = VariablesGenerales.FechaSistema;
        settings.Empresa = VariablesGenerales.SecuencialEmpresa;
        settings.UserName = System.Threading.Thread.CurrentPrincipal.Identity.Name;
        settings.Password = FBSProxies.ClientUsernameTokenManager.Pwd;
        settings.FechaCliente = DateTime.Now; settings.RolesCliente =
VariablesGenerales.NombresRol;
        ws.InnerChannel.SetHeader("settings", settings);

        return ws;
    }
}

```

Y los controles gráficos de tipo vista-controlador, que permite que los elementos visuales contenidos en los formularios tengan diferente comportamiento dependiendo

de la clase a la que se haga referencia, con lo cual se optimiza el uso de recursos y tiempo al implementar requerimientos adicionales o cambios.

FRAGMENTO DE CÓDIGO

Fuente: CrecimientoPresupuestarioAnual_Presentador.cs

```
class CrecimientoPresupuestarioAnual_Presentador : BasePresentador
{
    CrecimientoPresupuestarioAnual_Vista _vista;
    WorkItem _workItem;
    int _secuencialCuenta;
    DateTime _fechaAnioAnterior;
    DateTime _fechaMesAnterior;
    decimal _porcentajeCrecimiento;

    public
    CrecimientoPresupuestarioAnual_Presentador(CrecimientoPresupuestarioAnual_Vista
    vista, WorkItem workItem, int secuencialOficina, int secuencialCuenta, DateTime
    fechaAnioAnterior, DateTime fechaMesAnterior, decimal porcentajeCrecimiento)
    {
        this._vista = vista;
        this._workItem = workItem;

        this._secuencialCuenta = secuencialCuenta;
        this._fechaAnioAnterior = fechaAnioAnterior;
        this._fechaMesAnterior = fechaMesAnterior;
        this._porcentajeCrecimiento = porcentajeCrecimiento;

        InformacionPresupuestoME presupuesto = new InformacionPresupuestoME();
        presupuesto.SecuencialOficina = secuencialOficina;
        presupuesto.SecuencialCuenta = secuencialCuenta;
        presupuesto.FechaAnioAnterior = fechaAnioAnterior;
        presupuesto.FechaMesAnterior = fechaMesAnterior;

        vista.parametrosDetalleCrecimientoMSBindingSource.DataSource = new
        List<ParametrosDetalleCrecimientoMS>(DetalleCrecimientoPresupuestoAnualProxy.Devuel
        ve().DevuelveDatosCrecimientoPresupuestoAnual(presupuesto).Lista);

        vista.GuardarItem.Click += new EventHandler(Guardar_Click);
        vista.SalirItem.Click += new EventHandler(Cerrar_Click);
        vista.InformacionItem.Click += new EventHandler(InformacionItem_Click);
        vista.crecimientoGridView.CellEndEdit += new
        DataGridViewCellEventHandler(CrecimientoGridView_CellEndEdit);
        vista.Load += new EventHandler(CrecimientoPresupuestarioAnual_Load);
        vista.crecimientoGridView.CellContentClick +=
        crecimientoGridView_CellContentClick;
        PeriodoPresupuestarioMSE periodoPresupuesto =
        PeriodoPresupuestarioProxy.Devuelve().DevuelvePeriodoPresupuestarioAbierto();
        vista.periodoPresupuestoLabel.Text = "Periodo Presupuestario: " +
        periodoPresupuesto.AnioPresupuesto.ToString();
    }

    void Guardar_Click(object sender, EventArgs e)
    {
        try
        {
            ParametrosCabeceraDetalleME parametrosCabeceraDetalleME = new
            ParametrosCabeceraDetalleME();
```



```

        DetalleCrecimientoPresupuestoListaME listaME = new
DetalleCrecimientoPresupuestoListaME();

List<FBSProxies.Presupuestos.CrecimientoPresupuestoAnual.DetalleCrecimientoPresupue
stoAnualMSE> detalleCrecimientoListaME = new
List<FBSProxies.Presupuestos.CrecimientoPresupuestoAnual.DetalleCrecimientoPresupue
stoAnualMSE>();

FBSProxies.Presupuestos.CrecimientoPresupuestoAnual.DetalleCrecimientoPresupuestoAn
ualMSE me = null;

        if (_vista.crecimientoGridView.RowCount != 0)
        {
            CrecimientoPresupuestoAnualMSE crecimientoPresupuestoAnualMSE =
new CrecimientoPresupuestoAnualMSE();

                crecimientoPresupuestoAnualMSE.SecuencialCtaContable =
this._secuencialCuenta;
                crecimientoPresupuestoAnualMSE.FechaAnioAnterior =
this._fechaAnioAnterior;
                crecimientoPresupuestoAnualMSE.FechaMesAnterior =
this._fechaMesAnterior;
                crecimientoPresupuestoAnualMSE.EstaActivo = true;

                parametrosCabeceraDetalleME.CrecimientoPresupuestoME =
crecimientoPresupuestoAnualMSE;

                _vista.parametrosDetalleCrecimientoMSBindingSource.EndEdit();

                foreach (ParametrosDetalleCrecimientoMS item in
(List<ParametrosDetalleCrecimientoMS>)_vista.parametrosDetalleCrecimientoMSBindingS
ource.DataSource)
                {
                    me = new
FBSProxies.Presupuestos.CrecimientoPresupuestoAnual.DetalleCrecimientoPresupuestoAn
ualMSE();

                        me.Secuencial = item.SecuencialDetalle;
                        me.SecuencialOficina = item.SecuencialOficina;
                        me.ValorAnioAnterior = item.ValorAnioAnterior;
                        me.ValorMesAnterior = item.ValorMesAnterior;
                        me.ValorProyeccionDiciembre =
item.ValorProyeccionDiciembre;
                        me.DiferenciaAnios = item.DiferenciaAnios;
                        me.PorcentajeTendencia = item.PorcentajeTendencia;
                        me.ValorProyeccion = item.ValorProyeccion;
                        me.PorcentajeProyeccion = item.PorcentajeProyeccion;
                        me.EstaActivo = true;

                                detalleCrecimientoListaME.Add(me);
                }
                listaME.Lista = detalleCrecimientoListaME.ToArray();

parametrosCabeceraDetalleME.DetalleCrecimientoPresupuestoListaME = listaME;

        }

CrecimientoPresupuestoAnualProxy.Devuelve().ProcesaGuardar(parametrosCabeceraDetall
eME);

this._workItem.Workspaces[FBSComun.Constantes.WorkspaceNames.Consulta.ListaWorkspac
e].Close(this._vista);
        this._workItem.SmartParts.Remove(this._vista);

```

```

    }
    catch (Exception errorExc) {
FBSGeneralException.AdministrarError(VariablesGenerales.NombreEmpresa,
VariablesGenerales.NombreOficina, VariablesGenerales.NombreUsuario,
VariablesGenerales.FechaSistema, VariablesGenerales.NombreEmpresa, "", "Disculpe
hubo un error en Financial Business System v2.0", errorExc); }
    }
}

```

4.4. Pruebas

4.4.1. Pruebas de desarrollo

Al finalizar la implementación del módulo, existen un número de pruebas que deben ser llevadas a cabo con la finalidad de detectar posibles errores que se hayan cometido en el transcurso.

Pruebas de Caja Blanca

Las pruebas de caja blanca se realizan a nivel de software para comprobar la correcta funcional de los métodos establecidos.

Métodos de recuperación de datos

1	PeriodoPresupuestario periodoPresupuestario = PeriodoPresupuestarioActor.DamePorSecuencial(mensajeEntrada.Secuencial, false);
2	PeriodoPresupuestario periodoPresupuestario = PeriodoPresupuestarioActor.DameRegistroAbiertoSinVerificarExistencia();
3	mensajeSalida = new CuentaItemLista(listCuentaContableItem.ToArray());
4	mensajeSalida = new DameInformacionConRequisitosMS(this.DevuelveItemListaPorSecuencialPeriodoPresupuestario(new ParametrosPeriodoPresupuestoME(settings.Empresa, periodoPresupuestarioMSE.Secuencial)), oficinaWS.DevuelveItemListaPorSecuencialEmpresaEstaActivaConTodos(new DevuelveItemListaPorSecuencialEmpresaEstaActivoME(settings.Empresa, true, false, false)));
5	mensajeSalida = new DameInformacionConRequisitosMS(this.DevuelveItemListaPorSecuencialPeriodoPresupuestario(new ParametrosPeriodoPresupuestoME(settings.Empresa, periodoPresupuestarioMSE.Secuencial)), oficinaWS.DevuelveItemListaPorSecuencialEmpresaEstaActivaConTodos(new DevuelveItemListaPorSecuencialEmpresaEstaActivoME(settings.Empresa, true, false, false)));
6	CrecimientoPresupuestoAnual_DataSet.InformacionCrecimientoDataTable cuentaCorteMes = CrecimientoPresupuestoInformacion.DameCierreCuentas(mensajeEntrada.SecuencialCuenta, item.secuencialOficina, mensajeEntrada.FechaMesAnterior);

Todos los métodos definidos presentaron información correcta validada con el gestor de base de datos quien realizó consultas específicas para comprobar que los valores presentados por el módulo se encuentren acordes a la realidad.

Conversión al grafo de flujo:

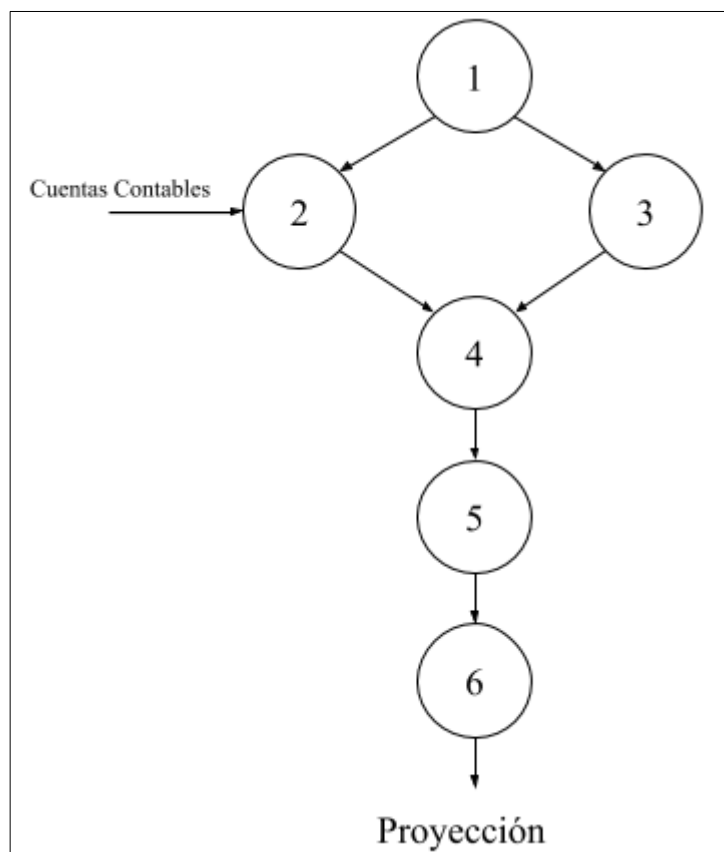


Fig. 31 Grafo de flujo

Camino básico: 1-2-3-4-5-6

Camino alternativo: 1-2-4-5-6

4.4.2. Pruebas de validación

Se realiza el ingreso aleatorio de datos en cada una de las funcionalidades con la finalidad de restringir casi es su totalidad la digitación errónea y evitar que el usuario cometa errores innecesarios. Las pruebas que se realizaron fueron:

- Ingreso de datos.
- Comprobación de llamado al formulario requerido.
- Verificación de datos presentados.

Nuevo Dato Periodo Presupuestario

Año Presupuesto: SFHAKFHSHKAD

Años Histórico: FASDKFJKDS Información incorrecta

Años Proyección: LDSAKFSDKJ

Esta Cerrado:

Duplicar Cuentas:

Fig. 32 Prueba de validación Valores Numéricos

Nuevo Dato Periodo Presupuestario

Año Presupuesto: 201564

Años Histórico: 56841 Información incorrecta

Años Proyección: 4564874

Esta Cerrado:

Duplicar Cuentas:

Fig. 33 Prueba de Validación Valores adecuados

Financial Business System 2.0

Cajas Captaciones Préstamos Contabilidad Activos Fijos Seguridad Tesorería Nomina Proceso en Lote Parámetros Recaudaciones Reportes

Periodo Presupuesto Indicadores Cuentas Indicadores Cuentas

Presupuestos > Indicadores Cuentas

Oficina: Todos Cuenta: 21013501 - CUENTAS ACTIVAS

Fecha Año Anterior: 31/12/2017 Fecha Mes Anterior: 31/07/2018

Fijar Porcentaje Método de Regresión

Periodo Presupuestario: 2017

Oficina	Año Anterior	Mes Anterior	Proyección Diciembre	Diferencia Años	% Tendencia	Valor Proyección	% Proyección
MATRIZ	18,598,205.84	29,059,347.63	49,816,024.51	31,217,818.67	167.85	55,286,031.90	10.98
SALCEDO	10,819,504.30	11,344,663.50	19,447,994.57	8,628,490.27	79.75	23,049,633.05	18.52
BUSO	5,625,342.57	5,304,665.44	9,093,712.18	3,468,369.61	61.66	9,884,456.66	8.70
PILLARO	8,828,906.26	9,279,014.90	15,906,882.69	7,077,976.43	80.17	18,976,715.64	19.30
IZAMBA	3,619,729.15	4,120,346.11	7,063,450.47	3,443,721.32	95.14	8,403,128.58	18.97
MACAS	1,057,760.89	1,227,729.49	2,104,679.13	1,046,918.24	98.97	2,473,970.48	17.55
PELLILO	6,006,729.48	6,145,815.84	10,535,684.30	4,528,954.82	75.40	12,888,193.11	22.33
TENA	1,648,548.20	1,678,215.66	2,876,941.13	1,228,392.93	74.51	3,268,194.76	13.60
LATACUNGA	2,057,708.39	2,237,653.14	3,835,976.81	1,778,268.42	86.42	4,760,466.47	24.10
OFICINA SUB	3,471,183.00	3,890,150.24	6,668,828.98	3,197,645.98	92.12	7,832,708.42	17.45
CEVALLOS	2,444,405.65	2,680,597.92	4,595,310.72	2,150,905.07	87.99	5,518,288.34	20.09
TISALEO	2,627,733.50	2,753,754.73	4,720,722.39	2,092,988.89	79.65	5,580,526.69	18.21
SAQUISILI	791,948.25	931,475.96	1,596,815.93	804,867.68	101.63	1,992,764.25	24.80
EL QUINCHE	627,095.57	586,112.50	1,004,764.29	377,668.72	60.23	1,272,844.86	26.68
SANGOLOUÍ	685,220.82	788,030.43	1,350,909.31	665,688.49	97.15	1,643,989.32	21.70
CONOCOTO	1,103,051.65	1,027,548.65	1,761,511.97	658,460.32	59.69	2,218,920.88	25.97
ALANGASÍ	810,703.27	818,501.02	1,403,144.61	592,441.34	73.08	1,700,255.43	21.17
VILLAFLORA	1,068,525.87	1,061,962.11	1,820,506.47	751,980.60	70.38	2,243,743.47	23.25
PUENGASÍ	568,881.71	676,803.11	1,160,233.90	591,352.19	103.95	1,381,809.07	19.10

Empresa: 02AGO2018 Oficina: MATRIZ Usuario: ADMINISTRADOR DEL SISTEMA Fecha: 07/08/2018

Fig. 34 Prueba de Validación Datos Reales

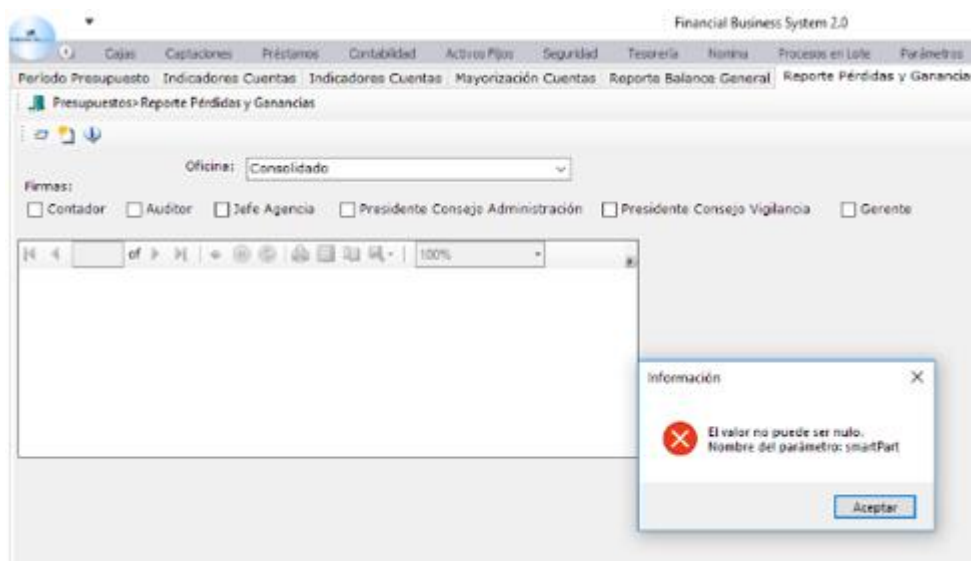


Fig. 35 Prueba de Validación Presentación de Mensajes

4.4.3. Pruebas de aceptación

Esta prueba tiene como finalidad validar que el módulo cuente con los requisitos básicos de funcionamiento que los directivos de la entidad esperen tener del mismo para que puedan dar por aprobado y finalizado el desarrollo.

Parámetros:

- Actividad: Módulo de presupuesto
- Número de veces que se repite el proceso: 1
- Criticidad de afectación al negocio: Media
- Responsable: Pamela Méndez
- Fecha de Prueba: 25/07/2018

Tabla 4.19: Resultado Pruebas de Aceptación

N°	Nombre del caso de prueba	Pre-requisitos/ Asunciones"	Valores Ingresados	Resultado Obtenido
1	Crear un nuevo período presupuestario	Parámetros > Presupuestos > Período Presupuesto	2016; 2; 1	Información almacenada
2	Asignar cuentas contables para el presupuesto	Parámetros > Presupuestos > Período Presupuesto	Cuentas contables de movimiento	Información almacenada
3	Distribución presupuestaria	Parámetros > Presupuestos > Período Presupuesto	20; 25; 25; 30	Porcentaje de cumplimiento trimestral

4	Revisar los indicadores de las cuentas del presupuesto	Presupuestos > Indicadores Cuentas	Ejecutar proceso	Indicadores de las cuentas contables
5	Mayorización de las cuentas del presupuesto	Presupuestos > Mayorización cuentas	Ejecutar proceso	Mayorización
6	Revisión reporte Balance General	Presupuestos > Reporte Balance General	Generar consulta	Reporte Balance General
7	Revisión reporte de pérdidas y ganancias	Presupuestos > Reporte Pérdidas y Ganancias	Generar consulta	Reporte Pérdidas y Ganancias
8	Revisión reporte de comparación presupuestaria	Reportes > Reportes > Presupuestos > Comparación Presupuestaria	Generar consulta	Reporte Comparativo

Estado de los casos de prueba: Exitosos.

Comparativa de presupuestación al año 2017

Tabla 4.20: Comparativa de resultados obtenidos

Cuenta Contable N°	Proceso Manual	Módulo
1.1.01.05 EFECTIVO	\$ 223,800.00	\$ 256,264.99
2.1.01 DEPOSITO A LA VISTA	\$ 102,500.00	\$ 109,937.38
3.3 RESERVAS	\$ 40,000,000.00	\$ 53,265,377.44
4.1.03 OBLIGACIONES FINANCIERAS	\$ 26,800.00	\$ 40,351.31
5.6.90.21 SERVICIO MÉDICO ODONTOLÓGICO	\$ 10,000.00	\$ 9,882.60

Para la comparación de valores proyectados se ha considerado una subcuenta de cada una de las cuentas principales que abarca el plan de cuentas de la empresa. Estas cuentas al considerarse para el cálculo de presupuesto anual tienen un valor en dinero cada año, el cual registrará a la empresa para decisiones como contratación de personal, reubicamiento del mismo y apertura de nuevas agencias.

Por medio de la tabla 4.20 se puede comprobar que los valores que el módulo de presupuesto financiero arroja no están alejados de la realidad, ya que, posee una variación moderada con aquellos valores que mediante los cálculos manuales usados anteriormente se obtenía, logrando la aceptación de resultados por parte de contabilidad y gerencia general.

Estado de prueba de aceptación: Exitosa

4.5. Puesta en Producción

La fase final de puesta en producción o implantación de software documenta las especificaciones del proceso de adaptación que siguió el módulo de presupuesto financiero para incorporarse al sistema en producción y llegar a ser funcional dentro de la institución.

Para integrar las funciones de presupuesto financiero al sistema Financiamiento 2.0 se requiere las especificaciones detalladas a continuación para que se pueda asignar la llamada al método dentro de la opción correspondiente ya que la creación del menú se lo realiza dentro del sistema en funcionamiento y únicamente con un usuario que cuente con el rol de Administrador para que los cambios puedan realizarse.

- Módulo: Presupuestos

Tabla 4.21: Puesta en Producción Funcionalidades

Nombre	WorkItem	Método a ejecutar
Periodo Presupuesto	ListaConsultaDetalle	InstanciaPeriodoPresupuestario
Indicadores Cuentas	Reportes	InstanciaReporteHistoricoCuentas
Mayorización Cuentas	MayorizacionPresupuesto	
Balance General Reporte	Reportes	InstanciaReporteBalanceGeneralPresupuesto
Pérdidas y Ganancias Reporte	Reportes	InstanciaReportePerdidasGananciasPresupuesto

Tabla 4.22: Puesta en producción Menú

Ruta	Funcionalidad
Parámetros > Presupuesto > Periodo Presupuesto	Periodo Presupuesto
Parámetros > Presupuesto > Indicadores Cuentas	Indicadores Cuentas
Parámetros > Presupuesto > Mayorización Cuentas	Mayorización Cuentas
Parámetros > Presupuesto > Balance General Reporte	Balance General Reporte
Parámetros > Presupuesto > Pérdidas y Ganancias Reporte	Pérdidas y Ganancias Reporte

Tabla 4.23: Puesta en producción Reporte Comparativa

Descripción	Path
Comparación Presupuestaria	/Presupuestos/ComparacionPresupuestaria_Report

Tabla 2.24: Puesta en producción Permisos

Funcionalidad	Rol
Periodo Presupuesto	Contador General
Indicadores Cuentas	Contador General
Mayorización Cuentas	Contador General
Balance General Reporte	Gerente General
Pérdidas y Ganancias Reporte	Gerente General

Al concluir este procedimiento, el menú se actualizará automáticamente en el próximo ingreso y será visible y funcional para aquellos usuarios que poseen los roles definidos en los permisos.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- El método para llevar a cabo el proceso de presupuestación que tenía la cooperativa no apartaba en eficacia ni precisión a los resultados presentados, por lo que la toma de decisión debía basarse totalmente en la experiencia de su dirigente, con la sistematización de este proceso, el monto destinado para nuevos proyectos, actividades y/o personal podrá contrastarse con lo empírico.
- El desarrollo del módulo de presupuesto financiero en conjunto con la metodología ágil SCRUM ha permitido que el mismo tenga características de adaptabilidad e integración ya que cada fase de revisión ha permitido la detección oportuna de errores para proceder a su respectiva corrección, documentando a su vez el modelado visual del software para que sea comprensible al usuario final mediante diagramado UML.
- La arquitectura N capas utilizada para la implantación del módulo permite que nuevas tareas o desarrollos adicionales puedan ser accedidos y asignados en forma distribuida, pues cada capa posee su entorno de programación, acceso a datos y presentación.
- Al implementar una tecnología basada en servicios como la que provee WCF permite que el módulo de presupuesto financiero sea escalable y tenga alto índice

de adaptabilidad para su ejecución en diferentes entornos, ya sea tecnología móvil, web o de escritorio.

- La sistematización de información del proceso de cálculo del presupuesto financiero de la cooperativa de ahorro y crédito “San Francisco LTDA.”, ha brindado a la empresa una visión general de los recursos con los que cuenta y las metas proyectadas para poder utilizar tiempo y esfuerzo al logro de las mismas.

5.2. Recomendaciones

- Se recomienda a futuros investigadores que planeen mejorar la funcionalidad del módulo, desarrollar un modelo matemático para implementarlo en el proceso de proyección de cuentas contables, ya que estas están sujetas a cambios inesperados no habituales, los cuales no están considerados.
- Es recomendable la aplicación de la tecnología entity framework para el desarrollo de software porque este abarca toda la arquitectura lógica y física necesaria para que un sistema sea adaptable, escalable y seguro.
- Para un correcto funcionamiento de despliegue del módulo se recomienda que la versión de .NET Framework se establezca en 3.0 o superiores, puesto que ciertas librerías utilizadas para la visualización gráfica de saldos contables por oficina son propias de esa versión.
- Se recomienda a la institución realizar una migración de datos ya que el módulo se alimenta directamente de información proporcionada por el sistema al cual pertenece y este contiene datos institucionales desde 2016, año en el cual la cooperativa comenzó a usarlo, el módulo al recibir más datos históricos le permitirá tener mayor asertividad en sus proyecciones.
- Es recomendable que el administrador de base de datos programe replicaciones automáticas cada cierto periodo de tiempo para que sirvan como respaldo de información en el caso de una incorrecta ejecución manual del proceso.

REFERENCIAS

- [1] L. Muñiz Gonzalez, «Planificación, elaboración, implantación y seguimiento del presupuesto,» de *Control presupuestario*, PROFIT, 2009.
- [2] U. T. d. S. T. Monterrey, *Presupuesto financiero*, México, 2012.
- [3] UNIVERSIDAD TÉCNICA DE AMBATO y M. C. León Flores, *EL PROCESO DE CONTRATACIÓN PÚBLICA DEL GOBIERNO AUTÓNOMO DESCENTRALIZADO DEL MUNICIPIO DE AMBATO EN LA EJECUCIÓN PRESUPUESTARIA*, Ambato, 2017.
- [4] UNIVERSIDAD TÉCNICA DE AMBATO y M. C. Cobo Vayas, *La planificación institucional y el presupuesto del proceso de contratación pública del Gobierno Autónomo Descentralizado Municipalidad de Ambato (GADMA)*, Ambato, 2017.
- [5] UNIVERSIDAD TÉCNICA DE AMBATO y L. E. Mejía Guevara, *La gestión de cobro de los tributos en el presupuesto del Gobierno Autónomo Descentralizado Municipal de Tisaleo*, Ambato, 2017.
- [6] Superintendencia de Economía Popular y Solidaria, «Lista de Organizaciones Supervisadas por la SEPS,» 2015.
- [7] SifizSoft S.A., «www.sifizsoft.com,» [En línea]. Available: <https://www.sifizsoft.com/>.
- [8] B. De La Hoz Suárez, M. A. Ferrer y A. De La Hoz Suárez, «Indicadores de rentabilidad: herramientas para la toma de decisiones financieras en hoteles de categoría media ubicados en Maracaibo,» *RCS Revista de Ciencias Sociales*, vol. XIV, nº 1, 2008.
- [9] UNIVERSIDAD TÉCNICA DE AMBATO y M. A. Lemus Robalino, *LA DEVOLUCIÓN DEL IMPUESTO AL VALOR AGREGADO Y EL PRESUPUESTO DE LA UNIVERSIDAD TÉCNICA DE AMBATO*, Ambato, 2014.
- [10] V. K. Reisdorfer, S. R. Koschewska y N. G. Salla, «PLANEAMIENTO FINANCIERO: SU IMPORTANCIA Y CONTRIBUCIÓN PARA LA GESTIÓN DE LAS EMPRESAS COOPERATIVAS,» *Revista Científica "Visión de Futuro"*, vol. 3, nº 1, pp. 1-11, 2005.
- [11] M. Vete y F. Makubikua, «Gestión de presupuestos financieros: Determinación de prioridades en el Hospital Central,» *Ciencias Holguín*, vol. XXI, nº 3, pp. 1-16, 2015.
- [12] K. Schwaber y J. Sutherland, *La Guía Definitiva de Scrum: Las reglas del Juego*, 2013.
- [13] K. Schwaber, M. Beedle y R. Martin, *Agile Software Development with SCRUM*, Prentice Hall, 2001.
- [14] J. Highsmith, *Agile Software Development Ecosystems*, Addison-Wesley, 2002.
- [15] S. Klein, *Pro Entity Framework 4.0*, Apress, 2010.

- [16] Y. Fernández Romero y Y. Díaz González, «Patrón Modelo-Vista-Controlador,» *Telemática*, vol. 11, nº 1, p. 4, 2012.
- [17] Microsoft, «Aplicaciones orientadas a servicios con WCF,» [En línea]. Available: <https://docs.microsoft.com/es-mx/dotnet/framework/wcf/whats-wcf>.
- [18] UNIVERSIDAD NACIONAL DE CHIMBORAZO y J. L. Klinger Córdova, *ANÁLISIS ESTADÍSTICO COMPARATIVO ENTRE AXIS2 Y WCF PARA MEJORAR LA INTEGRACIÓN DE APLICACIONES EN EL SISTEMA DE CONTROL DOCENTE EN LA ESCUELA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN DE LA UNACH*, Riobamba, 2016.
- [19] UNIVERSIDAD TÉCNICA DE AMBATO y M. C. Resabala Caicedo, *Sistema Informático para el control del mantenimiento técnico de equipos médicos del Hospital IESS - Ambato*, 2015.
- [20] SifizSoft S.A., *Financial Business System 2.0*, Quito.
- [21] Component, «ComponentSource,» [En línea]. Available: <https://www.componentsource.com/brand/devexpress>.
- [22] DevExpress, «DevExpress Documentation,» [En línea]. Available: <https://documentation.devexpress.com/WindowsForms/7889/What-s-Installed>.
- [23] Universidad César Vallejo y L. Cárdenas Escalante, *El patrón de arquitectura n-capas con orientación al dominio como solución en el diseño de aplicaciones empresariales*, Trujillo, 2013.
- [24] M. S. Server, «Developer Network,» [En línea]. Available: <https://msdn.microsoft.com/es-es/library/bb545450.aspx>.
- [25] L. R. Uribe Medina, «Plan de cuentas para sistemas contables en NIIF,» Ecoe Ediciones, 2016.
- [26] D. N. GUJARATI y D. C. Porter, *Econometría*.
- [27] Universidad de las Fuerzas Armadas ESPE y J. A. Bejarano Montesdeoca, *Implementación de un marco de trabajo basada en la Norma ISO /IEC 12207 y la Metodología ágil Scrum para el mejoramiento de la calidad de los productos de software en la Cooperativa de Ahorro y Crédito El Sagrario Ltda.*, Latacunga, 2015.
- [28] G. Booch, J. Rumbaugh y I. Jacobson, *UML - El Lenguaje Unificado de Modelado*, Madrid, 1999.

ANEXOS