

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE SOFTWARE

Tema:

AUTOMATIZACIÓN DE LOS PROCESOS DE LA GESTIÓN DE RIESGO DE LIQUIDEZ ESTRUCTURAL DE LA COOPERATIVA DE AHORRO Y CRÉDITO "SAN MARTÍN" DE TISALEO LTDA

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniera de Software

ÁREA: Gestión de Tecnologías de la Información

LÍNEA DE INVESTIGACIÓN: Tecnologías de la Información y Sistemas de control

AUTOR: Mariela Elizabeth Amanta Manobanda

TUTOR: MSc. Félix Fernández Peña, PhD.

Ambato - Ecuador

agosto – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: AUTOMATIZACIÓN DE LOS PROCESOS DE LA GESTIÓN DE RIESGO DE LIQUIDEZ ESTRUCTURAL DE LA COOPERATIVA DE AHORRO Y CRÉDITO "SAN MARTÍN" DE TISALEO LTDA, desarrollado bajo la modalidad Proyecto de Investigación por la señorita Mariela Elizabeth Amanta Manobanda, estudiante de la Carrera de Ingeniería de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que la estudiante ha sido tutorada durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, agosto 2024.

Firmado electrónicamente por:
FELIX OSCAR
FERNANDEZ PENA

MSc. Félix Fernández Peña, PhD.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: AUTOMATIZACIÓN DE LOS PROCESOS DE LA GESTIÓN DE RIESGO DE LIQUIDEZ ESTRUCTURAL DE LA COOPERATIVA DE AHORRO Y CRÉDITO "SAN MARTÍN" DE TISALEO LTDA es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2024.



Mariela Elizabeth Amanta Manobanda

C.C: 180442443-8

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, agosto 2024.



Mariela Elizabeth Amanta Manobanda

C.C: 180442443-8

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por la señorita Mariela Elizabeth Amanta Manobanda, estudiante de la Carrera de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado AUTOMATIZACIÓN DE LOS PROCESOS DE LA GESTIÓN DE RIESGO DE LIQUIDEZ ESTRUCTURAL DE LA COOPERATIVA DE AHORRO Y CRÉDITO "SAN MARTÍN" DE TISALEO LTDA, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, agosto 2024.



Firmado electrónicamente por:
ELSA PILAR URRUTIA URRUTIA

Ing. Elsa Pilar Urrutia Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL



Firmado electrónicamente por:
CARLOS ISRAEL NUNEZ MIRANDA

Ing. Carlos Nuñez Miranda, Mg.
PROFESOR CALIFICADOR



Firmado electrónicamente por:
DENNIS VINICIO CHICAIZA CASTILLO

Ing. Dennis Chicaiza Castillo, Mg.
PROFESOR CALIFICADOR

DEDICATORIA

La presente investigación está dedicada a mis padres, Eduardo Amanta y Gladys Manobanda, quienes han sido un pilar fundamental en mi vida. Con su apoyo, amor incondicional y sabiduría me han inspirado a alcanzar mis sueños.

A mis hermanos Edisson y Santiago, quienes, con sus consejos y palabras de aliento, han sido una fuente constante de motivación y apoyo a lo largo de todo este proceso.

AGRADECIMIENTO

Agradezco a Dios por la sabiduría que me ha brindado para alcanzar mis sueños.

A la Universidad Técnica de Ambato, a todos los docentes que compartieron su conocimiento a lo largo de mi proceso académico, en especial a mi tutor, Ing. PhD. Félix Fernández Peña, MSc., cuya vasta experiencia y conocimientos han sido fundamentales para alcanzar mi éxito personal y profesional.

Agradezco también a la Cooperativa de Ahorro y Crédito "San Martín" de Tisaleo Ltda., por confiar en mí y brindarme la oportunidad de desarrollar mi proyecto de titulación.

ÍNDICE GENERAL DE CONTENIDOS

PORTADA ..... **i**

APROBACIÓN DEL TUTOR ..... **ii**

AUTORÍA ..... **iii**

DERECHOS DE AUTOR ..... **iv**

APROBACIÓN DEL TRIBUNAL DE GRADO ..... **v**

DEDICATORIA ..... **vi**

AGRADECIMIENTO ..... **vii**

ÍNDICE GENERAL DE CONTENIDOS ..... **viii**

ÍNDICE DE TABLAS ..... **xii**

ÍNDICE DE FIGURAS ..... **xv**

ÍNDICE DE ANEXOS ..... **xix**

RESUMEN EJECUTIVO ..... **xx**

ABSTRACT ..... **xxi**

CAPÍTULO I. MARCO TEÓRICO **1**

1.1 Tema de investigación 1

1.1.1 Planteamiento del problema 1

1.2 Antecedentes investigativos 2

1.3 Fundamentación teórica 10

1.3.1 Base de Datos 10

1.3.2 Framework 12

1.3.3 IDE 13

1.3.4 Lenguaje de Programación 14

1.3.5 Arquitectura de Software 17

1.3.6 Sistema Operativo 19

1.3.7 Metodología Ágil 20

1.3.8 Pruebas de Software 21

1.3.9 Tipo de Investigación 22

1.4 Objetivos 22

1.4.1 Objetivo general 22

1.4.2 Objetivos específicos 22

CAPÍTULO II. METODOLOGÍA 24

2.1 Materiales 24

2.2 Métodos 24

2.2.1 Modalidad de la investigación 24

2.2.2 Población y muestra 25

2.2.3 Recolección de información 26

2.2.4 Procesamiento y análisis de datos 27

CAPÍTULO III. RESULTADOS Y DISCUSIÓN 29

3.1 Análisis y discusión de resultados..... 29

3.1.1 Metodología Ágil para el desarrollo del sistema..... 29

3.1.2 Resultados de la entrevista técnica..... 29

3.1.3 Levantamiento de Requerimientos Funcionales 31

3.1.4 Levantamiento de Requerimientos No Funcionales 32

3.1.5 Levantamiento de Historias de Usuario 33

3.2 Diseño 39

3.2.1 Arquitectura de software del sistema 39

3.2.2 Diseño de la Base de Datos..... 41

3.2.3 Diagramas de Caso de Uso 43

3.2.4 Diagramas de Secuencia 45

3.2.5 Diseño de la Interfaz 55

3.3 Implementación..... 67

3.3.1 Funcionamiento..... 67

3.3.2 Proceso de alimentación de tablas de auditoria..... 79

3.3.3 Procedimientos almacenados para la generación de reportes 100

3.4 Pruebas 117

3.4.1 Pruebas de Funcionalidad 117

3.4.2 Pruebas de Usabilidad..... 121

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES 124

4.1 Conclusiones 124

4.2 Recomendaciones 125

REFERENCIAS BIBLIOGRÁFICAS 126

ANEXOS 129

ÍNDICE DE TABLAS

Tabla 1. Cadenas de búsqueda y resultados 3

Tabla 2. Criterios de inclusión y exclusión 4

Tabla 3. Cuadro comparativo de las herramientas para la base de datos 11

Tabla 4. Cuadro comparativo de las herramientas para el framework 12

Tabla 5. Cuadro comparativo de las herramientas para el IDE 14

Tabla 6. Cuadro comparativo de las herramientas para el Lenguaje de Programación 16

Tabla 7. Cuadro comparativo de las herramientas para la Arquitectura del Software 18

Tabla 8. Cuadro comparativo de las herramientas para el Sistema Operativo 19

Tabla 9. Cuadro comparativo de las herramientas para la Metodología Ágil 21

Tabla 10. Resultados de la entrevista técnica 29

Tabla 11. Historia de usuario 1: Inicio de Sesión 33

Tabla 12. Historia de usuario 2: Cambio de Contraseña 33

Tabla 13. Historia de usuario 3: Gestionar Usuario 34

Tabla 14. Historia de Usuario 4: Gestionar Rol 35

Tabla 15. Historia de usuario 5: Gestionar Menú Sistema 35

Tabla 16. Historia de usuario 6: Gestionar acceso a acciones de Menú por Rol de usuario 36

Tabla 17. Historia de usuario 7: Gestionar Empleado 36

Tabla 18. Historia de usuario 8: Reporte Empleados 36

Tabla 19. Historia de usuario 9: Gestionar feriados 37

Tabla 20. Historia de usuario 10: Reporte de Liquidez Estructural 37

Tabla 21. Historia de usuario 11: Reporte de Volatilidad 38

Tabla 22. Historia de usuario 12: Reporte de Mayores Depositantes 38

Tabla 23. Historia de usuario 13: Reporte de Indicadores 38

Tabla 24. Historia de usuario 14: Reporte de Balances Diarios 39

Tabla 25. Historia de usuario 15: Cerrar Sesión 39

Tabla 26. SP_DFI_CON_MAYORIZAR. Alimentación de la tabla DFI_CONT_BALANCE 79

Tabla 27. SP_DFI_CON_MAYORIZAR_CUENTA. Alimentación de la tabla DFI_CONT_BALANCE 82

Tabla 28. SP_DFI_CON_MAYORIZAR_SUBGRUPO. Alimentación de la tabla DFI_CONT_BALANCE 94

Tabla 29. SP_DFI_CON_MAYORIZAR_GRUPO. Alimentación de la tabla DFI_CONT_BALANCE 95

Tabla 30. Creación de la tabla `URI_LIQ_INDICADOR` e ingreso de datos 97

Tabla 31. Creación de la tabla `URI_LIQ_MAYOR_DEPOSITANTE` e ingreso de datos 99

Tabla 32. SP_URI_LIQ_LIQUIDEZ_ESTRUCTURAL 100

Tabla 33. SP_URI_LIQ_VOLATILIDAD 107

Tabla 34. SP_URI_LIQ_MAYORES_DEPOSITANTES 115

Tabla 35. Pruebas de Funcionalidad 117

Tabla 36. Pruebas utilizando el Sistema de Usabilidad de Software 122

Tabla 37. Cálculos del puntaje SUS..... 122

ÍNDICE DE FIGURAS

Figura 1. Arquitectura de software del sistema para la automatización de procesos. 40

Figura 2. Diseño de la Base de Datos para el sistema 42

Figura 3. Tablas sin relación de la Base de datos 43

Figura 4. Diagrama de Caso de uso del Administrador 44

Figura 5. Diagrama de Caso de uso del Analista de Riesgos 44

Figura 6. Diagrama de secuencia de Inicio de Sesión 45

Figura 7. Diagrama de secuencia de Cambiar Contraseña 46

Figura 8. Diagrama de secuencia Gestionar Usuario 46

Figura 9. Diagrama de secuencia Gestionar Rol 47

Figura 10. Diagrama de secuencia Gestionar Menú Sistema 48

Figura 11. Diagrama de secuencia Acceso a acciones de Menú por Rol 48

Figura 12. Diagrama de secuencia Gestionar Empleado 49

Figura 13. Diagrama de secuencia Reporte Empleados 50

Figura 14. Diagrama de secuencia Gestionar Feriado 50

Figura 15. Diagrama de secuencia Reporte Liquidez Estructural 51

Figura 16. Diagrama de secuencia Reporte Volatilidad 52

Figura 17. Diagrama de secuencia Reporte Mayores Depositantes 52

Figura 18. Diagrama de secuencia Reporte Indicadores 53

Figura 19. Diagrama de secuencia Reporte Balances Diarios 54

Figura 20. Diagrama de secuencia Cerrar Sesión 54

Figura 21. Diseño de la Pantalla de Inicio de Sesión 55

Figura 22. Diseño de la Pantalla Principal 56

Figura 23. Diseño de la Pantalla de Cambio de Contraseña 56

Figura 24. Diseño de la Pantalla de Gestionar Usuarios 57

Figura 25. Diseño de la Pantalla de Crear Usuario 57

Figura 26. Diseño de la Pantalla de Modificar Usuario 58

Figura 27. Diseño de la Pantalla de Rol 58

Figura 28. Diseño de la Pantalla de Menú Sistema 59

Figura 29. Diseño de la Pantalla de Menú Rol 59

Figura 30. Diseño de la Pantalla de Empleado 60

Figura 31. Diseño de la Pantalla de Listado de Empleados 60

Figura 32. Diseño de la Pantalla de Feriados 61

Figura 33. Diseño de la Pantalla de Feriado 61

Figura 34. Diseño de Liquidez Estructual 62

Figura 35. Diseño de Volatilidad - Saldos 62

Figura 36. Diseño de Volatilidad - Variaciones 63

Figura 37. Diseño de Volatilidad – Deviación Estandar 63

Figura 38. Diseño de Volatilidad – Matriz de Correlación 64

Figura 39. Diseño de la Pantalla de Mayores Depositantes 64

Figura 40. Diseño de la pantalla de Limites de Riesgo de Liquidez 65

Figura 41. Diseño de la Pantalla de Medidas de Tendencia Central 65

Figura 42. Diseño de Estado de Situación Final 66

Figura 43. Pantalla de Inicio de Sesión 67

Figura 44. Pantalla Principal del Sistema 67

Figura 45. Pantalla de Cambio de Contraseña 68

Figura 46. Pantalla de Usuarios 68

Figura 47. Pantalla de Crear Usuario 69

Figura 48. Pantalla de Buscar 69

Figura 49. Pantalla de Modificar Usuario 70

Figura 50. Pantalla de Confirmación 70

Figura 51. Pantalla de Rol 71

Figura 52. Pantalla de Menú Sistema 71

Figura 53. Pantalla de Menú Rol 72

Figura 54. Pantalla de Empleado 72

Figura 55. Pantalla de Nomina 73

Figura 56. Pantalla de feriados 73

Figura 57. Pantalla de Feriado 74

Figura 58. Pantalla de Reporte de Liquidez Estructural 74

Figura 59. Pantalla de reportes de Volatilidad – Saldos 75

Figura 60. Pantalla de reportes de Volatilidad – Variaciones 75

Figura 61. Pantalla de reportes de Volatilidad – Desviaciones 76

Figura 62. Pantalla de Volatilidad – Matriz de Correlaciones 76

Figura 63. Pantalla de Mayores Depositantes 77

Figura 64. Pantalla de Indicadores 78

Figura 65. Pantalla de Balances Diarios 78

ÍNDICE DE ANEXOS

Anexo 1. Plantilla para la especificación de requisitos 129

Anexo 2. Respuestas a las preguntas de la entrevista técnica 131

RESUMEN EJECUTIVO

El proceso de la gestión de riesgo de liquidez estructural tiene por objeto el cálculo y registro de indicadores diarios referentes para asegurar que los activos líquidos de una entidad financiera sean suficientes para cubrir los pasivos exigibles en el tiempo y en las cantidades necesarias. El presente proyecto de investigación se centró en su automatización para la Cooperativa de Ahorro y Crédito San Martín. Para el desarrollo del sistema, se utilizó la metodología de desarrollo de software ágil Xtreme Programming (XP) y la arquitectura Cliente – Servidor. El sistema se desarrolló utilizando el lenguaje de programación C# en conjunto con el framework :Net y servidor de base de datos SQL Server, en consonancia con las políticas de la entidad para la que se desarrolló el trabajo. La herramienta propuesta superó las pruebas de funcionamiento. Así mismo, se obtuvo un valor de 80.62 al medir su nivel de usabilidad utilizando el instrumento SUS, aplicado a los usuarios potenciales de la propuesta de solución. Este resultado corrobora el cumplimiento de los requisitos funcionales y no funcionales del usuario. Se concluye que el sistema para la automatización de los procesos de la gestión de riesgo de liquidez estructural implementado permite gestionar de manera satisfactoria el proceso en cuestión, evitando los inconvenientes del tratamiento manual de los datos.

Palabras clave: Gestión de riesgo de liquidez estructura, Xtreme Programming, C#, .Net, SQL Server.

ABSTRACT

The structural liquidity risk management process is aimed at calculating and recording daily indicators to ensure that the liquid assets of a financial institution are sufficient to cover the liabilities due on time and in the necessary amounts. This research project focused on its automation for Cooperativa de Ahorro y Crédito San Martín. For the development of the system, the agile software development methodology Xtreme Programming (XP) and the Client - Server architecture were used. The system was developed using the C# programming language in conjunction with the .Net framework and SQL Server database server, in accordance with the policies of the entity for which the work was developed. The proposed tool passed the performance tests. Likewise, a value of 80.62 was obtained when measuring its usability level using the SUS instrument, applied to the potential users of the proposed solution. This result corroborates the fulfillment of the user's functional and non-functional requirements. It is concluded that the system for the automation of structural liquidity risk management processes implemented allows the process in question to be satisfactorily managed, avoiding the inconveniences of manual data processing.

Keywords: Structural liquidity risk management, Xtreme Programming, C#, .Net, SQL Server.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

AUTOMATIZACIÓN DE LOS PROCESOS DE LA GESTIÓN DE RIESGO DE LIQUIDEZ ESTRUCTURAL DE LA COOPERATIVA DE AHORRO Y CRÉDITO "SAN MARTÍN" DE TISALEO LTDA

1.1.1 Planteamiento del problema

En la publicación [1] se menciona que la gestión de riesgo de liquidez estructural desempeña un papel fundamental en la identificación y prevención de posibles crisis económicas. El riesgo de liquidez, según Ghosh, se origina por desfases en el patrón de vencimiento de los activos y pasivos de una entidad financiera [2]. Este riesgo se puede detectar al evaluar los índices de liquidez estructural. En este contexto, se considera que un manejo inapropiado de la gestión de riesgo de liquidez estructural podría desencadenar una crisis financiera, con efectos perjudiciales en la economía y un debilitamiento de la confianza en el sistema financiero [3].

En la publicación [4] se destaca que el feriado bancario que tuvo lugar del 8 al 12 de marzo de 1999, fue uno de los cinco hitos de la crisis financiera que afectó al país durante el último quinquenio del siglo XX. Esta crisis financiera encuentra sus raíces en la inestabilidad monetaria y financiera que se gestó a partir de 1982 en la periferia del sistema capitalista, vinculada con la crisis de la deuda externa mexicana. Además, otras causas que desembocaron en el feriado bancario están relacionadas con la crisis financiera que se gestó en los primeros meses de 1996. En ese periodo, el Banco Central del Ecuador enfrentó problemas de liquidez que evolucionaron hacia una situación irreversible de insolvencia. Aunque el feriado bancario se percibe como un evento específico y coyuntural no planificado, resulta esencial para comprender la naturaleza de la crisis financiera ecuatoriana de finales del siglo XX.

En el mismo sentido la Cooperativa de Ahorro y Crédito "San Martín" de Tisaleo Ltda. se enfrenta a un desafío crítico debido a la ineficiencia de su sistema de la gestión de riesgo de liquidez estructural. Los cálculos ejecutados en la institución con ayuda de Excel, resultan en procesos lentos y susceptibles a errores. Esto impacta negativamente en la administración estratégica de la liquidez estructural. En la publicación [5] se menciona que, la administración estratégica involucra evaluar la relación entre activos líquidos y pasivos exigibles. Por medio de cálculos realizados con ecuaciones según lo establecido por la Superintendencia de Economía Popular y Solidaria (SEPS). Estos cálculos incluyen tanto la Liquidez Estructural de Primera Línea, la Liquidez Estructural de Segunda Línea que determinan la disponibilidad de recursos altamente líquidos. También la volatilidad que mide la variación de las fuentes de financiamiento de la entidad con el tiempo. En este contexto, es esencial destacar que se debe llevar una buena gestión del riesgo de liquidez estructural, ya que permite protegerse de posibles fluctuaciones que podrían ocasionar pérdidas económicas.

1.2 Antecedentes investigativos

En la presente investigación, se llevó a cabo un análisis de la literatura disponible en diversas bases de datos científicas, tales como: Willey, Google Scholar, Springer y Science Direct. Asimismo, se definieron cadenas de búsqueda basados en palabras clave específicas como: "system", "financial system", "software solution", ”automated tool", "digital platform", "web platform", "software", .automation", "structural liquidity", "structural liquidity management", "liquidity management", çompany", "financial institution", "bank", çooperative". Estas palabras fueron seleccionadas cuidadosamente para garantizar la búsqueda de la revisión de literatura. En la tabla 1 se presentan las cadenas de búsqueda en las revistas en las que se generó la búsqueda y el número de artículos que se encontró.

Tabla 1. Cadenas de búsqueda y resultados.

Fuente: Elaborado por autor

Nº	CADENA	REVISTA	Nº ARTÍCULOS
1	(”system” OR “financial system” OR ”software solution” OR “automated tool” OR “digital platform” OR “web platform” OR “software” OR “automation”) AND (”structural liquidity” OR “structural liquidity management” OR “liquidity management”) AND (”company” OR “financial institution” OR “bank” OR “cooperative”)	Willey	15
		Google Scholar	430
		Springer	23
		Science Direct	9
2	((“financial services” OR “banking”) AND (“technology integration” OR “digital solutions” OR “automation system”) AND (“financial institution” OR “credit union” OR “financial entity”)	Willey	5
		Google Scholar	360
		Springer	7
		Science Direct	2
3	(“liquidity risk” OR “liquidity analysis”) AND (“web-based system” OR “digital platform”) AND (“financial institution”)	Willey	3
		Google Scholar	230
		Springer	15
		Science Direct	7
4	(“structural liquidity” OR “liquidity structure”) AND (“financial institution” OR “bank” OR “credit union”) AND “automation”	Willey	2
		Google Scholar	130
		Springer	23
		Science Direct	5
5	(“structural liquidity” AND “software solution”) OR (“structural liquidity” AND “financial institution” AND “automation”)	Willey	4
		Google Scholar	86
		Springer	24
		Science Direct	7
		Total	**1387**

El total de artículos encontrados en la revisión son 1387. Estos artículos van a ser evaluados por unos criterios de inclusión y exclusión. Los criterios de inclusión y exclusión se muestran en la tabla 2.

Tabla 2. Criterios de inclusión y exclusión.

Fuente: Elaborado por autor

Inclusión	Las investigaciones y soluciones consideradas deben abordar de manera integral la gestión de riesgo de liquidez estructural y la automatización de procesos en entidades financieras.
	Deben ser artículos publicados en revistas científicas, conferencias académicas, pueden ser tesis y libros.
	La fecha de publicación debe situarse entre el año 2018 y la fecha actual.
	Los estudios deben incluir casos prácticos, resultados empíricos o implementaciones reales de sistemas de automatización para respaldar la validez y aplicabilidad de las soluciones propuestas.
Exclusión	Se excluyen investigaciones redactadas en idiomas distintos al inglés o español.
	Se excluyen documentos publicados antes del año 2018.
	Se excluyen investigaciones que se centren únicamente en la teoría sin proporcionar ejemplos prácticos o implementaciones tangibles de sistemas de automatización para la liquidez estructural.
	Excluir trabajos que no presenten metodologías claras o que carezcan de rigor en la recopilación y análisis de datos relacionados con la mejora de la liquidez estructural.

El propósito de esta investigación fue identificar artículos que aporten a la automatización de los procesos de gestión del riesgo de liquidez estructural. Se realizo un análisis de diversas bases de datos científicas. Utilizando la tabla 2 que contienen los criterios de inclusión y exclusión se filtraron los artículos de los cuales se excluye 816 por la poca información que aportan a la investigación, 193 que se desviaban considerablemente del tema central. Además, se descartaron 104 artículos que no abordaban enfoques de automatización, 145 artículos con metodologías que tenían otra perspectiva y 114 estudios que se limitaban al análisis de herramientas sin proporcionar resultados. Como resultado, la investigación culminó con la

identificación de 15 artículos directamente relacionados con el tema, los cuales presentaron resultados sustentados y relevantes para el ámbito de estudio.

En la publicación [6] los autores proponen y desarrollan un sistema web integral para la Cooperativa de Ahorro y Crédito Mujeres Unidas. Utilizando herramientas de desarrollo como Ruby, RubyMine como IDE (Integrated Development Environment), Ruby on Rails como framework de desarrollo. Para la base de datos PostgreSQL como gestor principal y SQL (Structured Query Language) Server como base de datos relacional. Los autores mencionan que se logró una eficiente gestión de procesos judiciales y recursos humanos. Asimismo, como la generación de informes estadísticos sobre créditos, agencias y desempeño social. Además, los autores utilizan React para el diseño de la interfaz lo que garantiza una experiencia interactiva. Por otro lado, la metodología ocupada fue Scrum. También se menciona que se realizaron pruebas de funcionamiento en módulos clave, como cobranzas judiciales y recursos humanos. Las pruebas dieron un resultado del 97.4%, indicando un alto rendimiento. Resultados similares se obtienen en los módulos de créditos, agencias y desempeño social, con un 96.3 %. Las pruebas confirman la eficacia del sistema. Sin embargo, el artículo carece de detalles sobre la investigación y técnicas utilizadas, así como información sobre la población o muestra trabajada, aspectos que podrían fortalecer la comprensión del estudio.

En la publicación [7] los autores proponen la implementación de un sistema web-móvil. El objetivo es mejorar la gestión de ahorros y préstamos en una cooperativa de crédito en Cusco. Utilizando herramientas de desarrollo como MySQL para la base de datos, Angular JS como marco de trabajo, JavaScript para el lenguaje de programación. Además, se menciona que se trabajó con una muestra poblacional de 20 operaciones financieras. La recopilación de datos incluyó información de bancos y datos estadísticos, utilizando técnicas de observación y análisis documental. El autor menciona que la investigación se clasificó como aplicada y de diseño preexperimental. El resultado es la reducción del tiempo promedio de generación de informes financieros. El tiempo va de 36.25 a 11.45 minutos, representando una disminución del 68.41 %. Asimismo, se logró reducir el tiempo de búsqueda de información y el tiempo de procesamiento.

En la publicación [8], se desarrolló e implementó una aplicación web. Esta aplicación se encarga de la identificación, procesamiento e informe de clientes morosos de la Cooperativa de Ahorro y Crédito CREDIFINCA en el distrito de Chivay, Arequipa. Utilizando MySQL como gestor de base de datos. El dataset se conformó mediante la recopilación directa de información de los clientes en situación de morosidad. La aplicación, diseñada para ser utilizada por todos los analistas de crédito de la cooperativa, se construyó con la librería Ajax. La investigación, de tipo aplicado y explicativo correlacional, incluyó datos del índice de morosidad de los años 2019 y 2020. Estos datos fueron recolectados a través de un cuestionario que empleó técnicas de encuesta y análisis cuantitativo utilizando el coeficiente Rho de Spearman. Los resultados indicaron que la aplicación permitió procesar, seleccionar e informar de manera oportuna y confiable la morosidad de la cooperativa.

En la publicación [9], se llevó a cabo la implementación de un módulo de facturación electrónica. Este módulo es adaptable al sistema Morvisoft para la Cooperativa de Ahorro y Crédito San Martín. Utilizando SQL Server Express Edition 2016 como gestor de base de datos. El dataset se formó mediante un objeto SqlDataAdapter y consultas SQL. La población de la investigación incluyó a los empleados de la cooperativa, y el desarrollo se realizó en Visual Studio con C#. La arquitectura del sistema se basó en capas y componentes modulares ejecutándose en un servidor, con enfoque cliente-servidor. La recolección de datos se realizó a través de observación de campo y encuestas. Bajo la metodología XP, los resultados evidenciaron que el módulo optimizó el proceso de generación, firma, envío y autorización de comprobantes electrónicos. Esto facilito la certificación del SRI. La integración al sistema de la cooperativa se logró con pruebas exitosas, incluyendo pruebas de adaptación al sistema Morvisoft.

En la publicación [10], se implementó la aplicación web-móvil en la Cooperativa de Ahorro y Crédito "Kullki Wasi". La aplicación realizara consultas y transferencias de saldo. Utilizando SQL Server como gestor de base de datos y herramientas de análisis como diagramas de UML (Unified Modeling Language). La población de

socios supera los 60000, con una muestra de 202 socios para la encuesta. Los IDEs empleados fueron Visual Studio .Net y Android Studio, junto con los lenguajes C# y Java. La arquitectura de tres capas se implementó en el sistema operativo Android. Los autores mencionan que los resultados destacaron mejoras en la calidad de servicios. Además, se redujo los tiempos de espera de los procesos financieros.

En la publicación [11], se desarrolló e implementó con éxito una aplicación web responsiva. La aplicación quiere facilitar la gestión de análisis en el proceso de obtención de créditos en la Cooperativa de Ahorro y Crédito CREDIL LTDA. La aplicación, creada con bases de datos Informix, involucró a 25 empleados. Se desarrollo utilizando NetBeans como IDE, con los lenguajes de programación Java y HTML (HyperText Markup Language). Guiada por la arquitectura MVC (Model-View-Controller). La aplicación se diseñó utilizando una base de datos existente que fue rediseñada para complementar las tablas necesarias. Las pruebas de operatividad se llevaron a cabo bajo la metodología ágil SCRUM. Los resultados obtenidos son la facilidad en la gestión de datos durante la solicitud de crédito. Esto agiliza el proceso de análisis y permite el acceso desde dispositivos móviles o equipos de escritorio.

En la publicación [12], se llevó a cabo el desarrollo de una aplicación web y móvil. La aplicación se encarga de consulta de saldos y movimientos de cuentas en la Cooperativa de Ahorro y Crédito UNIBLOCK y Servicios Ltda. Utilizando MySQL como base de datos relacional. El proyecto utilizo una población de 11 personas y se basó en el framework Laravel con programación en PHP. La arquitectura adoptada fue de tres capas. Por otro lado, para la recopilación de datos se fundamentó en información sobre las necesidades de los clientes y la factibilidad del desarrollo. Las técnicas empleadas fueron encuestas y observación. Las pruebas de requerimientos funcionales se llevaron a cabo bajo la metodología ágil SCRUM. Los resultados revelaron que la aplicación para consulta de saldos y movimientos optimizó trámites. Además, proporciono información en tiempo real sobre el estado de las cuentas de los clientes.

En la publicación [13], se presenta el desarrollo de una aplicación móvil. La aplicación realiza transferencias directas e interbancarias en la Cooperativa de

Ahorro y Crédito San Martín. Utilizando SQL Server Express como gestor de base de datos. La aplicación se construyó en Android NDK (Native Development Kit) con Android Studio como IDE y Java como lenguaje de programación. Por otro lado, utilizaron la arquitectura MVP (Model-View-Presenter). La investigación se basó en datos actuales de transferencias, tiempos empleados y resultados de transacciones anteriores. Las técnicas fueron observación, entrevistas y encuestas en una combinación de investigación de campo y bibliográfica. Se llevaron a cabo pruebas de aceptación, integración, rendimiento y usabilidad bajo la metodología ágil XP (Extreme Programming). Los resultados destacaron que se pudo realizar transferencias de manera más rápida y sin la necesidad de acudir físicamente a las oficinas de la cooperativa.

En la publicación [14], los autores proponen la implementación de una aplicación móvil y de escritorio. La aplicación pretende facilitar la gestión de préstamos en la empresa CrediServi de Chinandega. La aplicación móvil permitirá a los empleados realizar recaudaciones, almacenar datos y enviar cuotas a una computadora central. La aplicación de escritorio permite al administrador recibir y gestionar estas cuotas. La base de datos utilizada es MySQL. El entorno de desarrollo incluye Android Studio y Visual Studio. El lenguaje de programación utilizado es PHP para la conexión a la base de datos y el desarrollo de la aplicación móvil. Se menciona que los datos del proyecto incluyen información de clientes con financiamiento y datos de recaudaciones. Por último, realizaron pruebas de la funcionalidad de la aplicación.

En la publicación [15], los autores proponen la creación de una herramienta para optimizar la administración eficiente del almacenamiento de información. La base de datos utilizada es MySQL, además, el dataset se formó a través de entrevistas y encuestas a los empleados. El lenguaje de programación que se utilizó es PHP, complementado con HTML5 y JavaScript para el desarrollo del aplicativo web. El autor menciona que los datos del proyecto provienen de registros de usuarios, documentación y reportes de ingreso de documentos. Además, se realizaron pruebas de escritorio para asegurar el correcto funcionamiento del programa y pruebas con datos reales mediante pruebas extras y revisiones. El autor destaca la automatización lograda en la gestión de documentos.

En la publicación [16] se aborda la normativa de la Superintendencia de Bancos. Esta se relaciona con el mantenimiento de un nivel adecuado de liquidez estructural. El autor propone la implementación de un sistema de monitoreo de límites de exposición de riesgo de liquidez en una entidad financiera mediana en Ecuador. La base de datos utilizada se generó a partir de los saldos de cuentas diarios de la entidad. La metodología aplicada fue de carácter explicativo, destinada al análisis del comportamiento y las tendencias de los indicadores de liquidez más relevantes. Para la definición de indicadores, se consideraron datos históricos desde enero de 2010 hasta diciembre de 2014. La información utilizada incluyó estados financieros, operaciones de crédito y saldos de depósitos a la vista y a plazo. Las técnicas cuantitativas, análisis histórico y backtesting fueron empleadas para obtener los límites de exposición de riesgo de liquidez. Los resultados obtenidos no solo proporcionaron límites claros de exposición, sino también revelaron el nivel de fortaleza actual del sector en la gestión del riesgo de liquidez, identificando acciones de mejora para optimizar su administración.

En la publicación [17] se destaca la evaluación de la liquidez estructural como una técnica clave para mejorar la gestión del riesgo de liquidez. La autora señala la importancia de contar con sistemas informáticos eficaces para el monitoreo y proyecciones. La metodología de Valor en Riesgo (VaR) se menciona para el cálculo de la volatilidad. Además, se llevan a cabo visitas in situ a cinco Unidades de Riesgo de cooperativas seleccionadas, analizando 36 variables. La investigación se basa en datos provenientes de estados financieros de cooperativas y normas emitidas por el Comité de Supervisión Bancaria de Basilea y la Superintendencia de Bancos. El empleo de técnicas como revisión documental, entrevistas y encuestas contribuye a la obtención de resultados significativos. Estos resultados permiten identificar áreas clave que requieren fortalecimiento y acciones de mejora. La combinación de diversas técnicas y enfoques revela la complejidad del estudio y destaca la necesidad de estrategias integrales.

1.3 Fundamentación teórica

La Fundamentación Teórica engloba principios conceptuales tanto tecnológicos como financieros, desempeñando un papel fundamental como guía orientadora del proyecto.

1.3.1 Base de Datos

La base de datos actúa como un repositorio organizado y eficiente para el almacenamiento, gestión y recuperación de grandes cantidades de datos. 15 PostgreSQL se distingue por su arquitectura, la cual garantiza la fiabilidad, integridad y corrección de los datos. Además, presenta la ventaja de ser compatible y ejecutarse eficientemente en los principales sistemas operativos. En [6], el autor emplea PostgreSQL como gestor de base de datos para almacenar la información del sistema en tablas relacionales.

MySQL administra bases de datos relacionales de código abierto. En [7], [8] y [15], los autores resaltan su preferencia por este gestor de bases de datos como un entorno seguro para almacenar información, ya que facilita la organización lógica de los datos. En [12], se demuestra la eficiencia de este gestor al colaborar con Xamarin, destacándose por su velocidad, robustez y facilidad de uso. Además, en [14], el autor menciona que gracias a MySQL se logra establecer conexiones a la base de datos de manera sencilla y eficiente.

SQL Server es un sistema de gestión de bases de datos relacionales que almacena y recupera datos solicitados por otras aplicaciones de software. En [9] se utiliza SQL Server Express Edition 2016, y en [10], SQL Server 2008; los dos son gestores de bases de datos gratuitos que pueden ser empleados sin inconvenientes. Sin embargo, en [9], el autor sugiere considerar la versión de pago. Además, en [13], se resalta que SQL Server posee un robusto sistema de gestión de bases de datos relacionales, que opera mediante el lenguaje de consultas Transact-SQL,. Además, menciona que entre sus destacadas funcionalidades se incluye la capacidad para administrar otros servidores de datos y la opción de operar en un modo cliente-servidor.

En [11], el autor destaca a Informix como un sistema de gestión de bases de datos relacionales diseñado para el almacenamiento masivo de datos en tablas virtuales. Reconocido por su robustez y seguridad, este software exhibe una notable capacidad de escalabilidad de datos. Además, menciona que integra un sistema de gestión de bases de datos relacionales basado en SQL de cuarta generación, lo que simplifica la ejecución de consultas en programas de aplicación.

En la tabla 3 se presenta el análisis, en el cual se enfocarán las características de las bases de datos, con el propósito de facilitar la toma de decisiones.

Tabla 3. Cuadro comparativo de las herramientas para la base de datos.
Fuente: Elaborado por autor

Características	PostgreSQL	MySQL	SQL Server	Informix
Cuenta con soporte ACID	X	X	X	X
Permite la gestión de transacciones	X	X	X	X
Permite la creación de Stored Procedures		X	X	
Permite la creación de Triggers	X	X	X	X
Permite la creación de Vistas	X	X	X	X
Permite la replicación de datos	X	X	X	X
Ofrece escalabilidad horizontal y vertical	X	X	X	X
Ofrece soporte XML	X	X	X	X
Permite el almacenamiento, consulta y manipulación de datos en formato JSON	X	X	X	X
Ofrece soporte para datos espaciales	X		X	

En la tabla se muestran las características de las diferentes bases de datos. Gracias a las ventajas que ofrece SQL Server, se ha optado por seleccionar esta herramienta. SQL Server se destaca por su escalabilidad, seguridad y creación de Stored Procedures, asegurando la integridad y confiabilidad de los datos. Además, la empresa indica que sería factible realizar el desarrollo con SQL Server debido a su facilidad de uso y amplia disponibilidad de soporte técnico.

1.3.2 Framework

Los frameworks representan un conjunto integral de conceptos, prácticas y herramientas diseñadas para agilizar el proceso de desarrollo al ofrecer soluciones a problemas específicos.

Ruby on Rails es un marco de aplicación web escrito en el lenguaje de programación Ruby. En [6] se destaca la aplicación del framework de desarrollo ágil Ruby on Rails en el ámbito del back-end, el cual demuestra ser altamente productivo al centrarse en las necesidades de los desarrolladores, facilitando la adopción de mejores prácticas de desarrollo. El autor menciona que permite la creación de sistemas robustos con código limpio, promoviendo así la facilidad de mantenimiento del sistema, además, Ruby on Rails se distingue por su capacidad para integrar módulos y herramientas específicas, simplificando tareas complejas. El autor subraya cómo el framework no solo estructura la lógica y el flujo de la aplicación, sino que también proporciona una solución eficiente para llevar a cabo las tareas más desafiantes.

.NET es un marco gratuito, multiplataforma y de código abierto. En [10] y [14], los autores hacen referencia al framework .NET, el cual es compatible con varios lenguajes de programación y brinda a los desarrolladores la libertad de elegir el lenguaje que mejor se ajuste a sus requerimientos.
Laravel es un marco de desarrollo de código abierto. En [12] se menciona que Laravel se destaca como un framework que incorpora las mejores características de otros frameworks y las integra para consolidarse como una plataforma sólida. Un ejemplo elocuente de esto es Xamarin.

En la tabla 4 se presenta el análisis, en el cual se enfocarán las características de los frameworks, con el propósito de facilitar la toma de decisiones.

Tabla 4. Cuadro comparativo de las herramientas para el framework.

Fuente: Elaborado por autor

Características	Ruby on Rails	.NET	Laravel
Conexión con SQL Server		X	

Permite un desarrollo rápido	X	X	X
Cuenta con una gran comunidad de desarrolladores	X	X	X
Ofrece características de seguridad avanzada	X	X	X
Ofrece escalabilidad horizontal y vertical	X	X	X
Cuenta con soporte de IDE		X	X
Cross-Platform		X	
Permite un sólido soporte para pruebas automatizadas	X	X	X
Compatibilidad con herramientas de integración continua	X	X	X

En la tabla se muestran las características de los diferentes frameworks. Gracias a las ventajas que ofrece .NET, se ha optado por seleccionar esta herramienta. .NET se destaca por su escalabilidad y soporte integral para desarrollo de escritorio. Además, su integración con SQL Server, Visual Studio y otras herramientas de Microsoft permiten un desarrollo más eficiente. La empresa indica que sería factible realizar el desarrollo con .NET.

1.3.3 IDE

Un Entorno de Desarrollo Integrado es un programa que reúne un conjunto de herramientas útiles para los desarrolladores, permitiéndoles agilizar diversas tareas.

RubyMine es un entorno de desarrollo integrado. En [6], el autor menciona que RubyMine proporciona herramientas esenciales para el desarrollo con Ruby. Además, presenta características como autocompletado, generación de controladores y modelos desde una interfaz, integración con diversos lenguajes de marcado y un lanzador de pruebas.

Visual Studio es un entorno de desarrollo integrado creado por Microsoft. En [9], el autor elabora una tabla comparativa entre los dos entornos de desarrollo previstos, NetBeans y Visual Studio. Tras la comparación, el autor se inclina por Visual Studio debido a su capacidad para desarrollar aplicaciones con componentes que permiten verificar su rendimiento. Además, destaca su uso compartido de herramientas y la posibilidad de programar en varios lenguajes. Asimismo, se realiza una comparación entre las versiones de Visual Studio, siendo la opción ganadora Visual Studio

Community 2017. En [10] y [14], otros autores también hacen referencia al uso del IDE Visual Studio, describiéndolo como una herramienta extremadamente potente y gratuita que abarca un conjunto de herramientas multiplataforma.

En la tabla 5 se presenta el análisis, en el cual se enfocarán las características de los IDE's, con el propósito de facilitar la toma de decisiones.

Tabla 5. Cuadro comparativo de las herramientas para el IDE.
Fuente: Elaborado por autor

Características	RubyMine	Visual Studio
Conexión con SQL Server		X
Conexión con .NET		X
Soporte multiplataforma	X	
Herramientas de aprendizaje automático integradas		X
Entorno integrado de desarrollo completo		X
Integración de Git	X	X
Depuración	X	X
Herramienta de productividad	X	X
Soporte de desarrollo en la nube		X

En la tabla se muestran las características de los diferentes IDE's. Gracias a las ventajas que ofrece Visual Studio, se ha optado por seleccionar esta herramienta. Visual Studio se destaca por su compatibilidad con múltiples lenguajes de programación. Además, su integración con .NET y SQL Server, lo convierten en una opción ideal para el desarrollo del proyecto. La empresa indica que sería factible realizar el desarrollo con Visual Studio debido a su facilidad de uso, su interfaz intuitiva.

1.3.4 Lenguaje de Programación

En [11] se menciona que los lenguajes de programación se caracterizan por conjuntos de reglas sintácticas, semánticas y símbolos que definen su estructura y expresan el significado de los elementos.

Ruby es un lenguaje de programación dinámica, de alto nivel y orientado a objetos. En [6], el autor menciona que se utilizará Ruby cómo lenguaje de programación

orientado a objetos, ya que este lenguaje es muy amigable y cuenta con un lenguaje natural, lo que quiere decir que tiene un lenguaje expresivo, este lenguaje se usa para aplicaciones web y es un lenguaje multiplataforma.

Javascript es un lenguaje interpretado, lo que significa que se ejecuta directamente en el navegador sin necesidad de compilación previa. En [7], el autor menciona que se ocupará el lenguaje de programación javascript el cuál es un lenguaje qué favorece al programador ya que facilita las órdenes en la web.

C# es un lenguaje de programación que sigue un modelo de programación orientada a objetos. En [9], el autor elige C# debido a la existencia de un software previamente desarrollado en la cooperativa bajo la plataforma Visual Studio con el mismo lenguaje. La utilización de C#, respaldada por las clases y librerías proporcionadas por el framework, simplifica la creación del formato XML (eXtensible Markup Language) para el envío de comprobantes desde C# mediante un esquema SOAP (Simple Object Access Protocol). El autor mencina que este lenguaje funciona de manera normal, sin inconvenientes aparentes, y mantiene la velocidad del módulo sin afectaciones. Por otro lado, en [10], el autor menciona el uso del lenguaje de programación C# para el desarrollo web. En este sistema, una estructura de C# se configura en clases, donde todas las librerías proporcionadas por el framework están formadas por clases con sus métodos correspondientes. En [14], el autor menciona que C# fue aprobado como estándar por la ECMA (European Computer Manufacturers Association) e ISO (International Organization for Standardization). La sintaxis fundamental de C# se deriva de C/C++, y su modelo de objetos se basa en la plataforma .NET, compartiendo similitudes con Java y a la vez incorporando mejoras inspiradas en otros lenguajes. Aunque C# forma parte de la Plataforma .NET, se distingue claramente como un lenguaje de programación independiente diseñado para generar programas en dicha plataforma. Además, cabe mencionar que C# fue seleccionado para el desarrollo del producto [14], después de un análisis comparativo entre lenguajes como Kotlin, PHP y otros, destacándose por sus características y capacidades para cumplir con los requisitos del proyecto.

Java es reconocido como un lenguaje orientado a objetos, destaca por su potencia y adaptabilidad para la programación de aplicaciones en red. En [10], el autor menciona la elección de Java como lenguaje de programación para aplicaciones móviles. El cual es conocido por ser fácil de usar. El autor menciona que constituye un lenguaje de alto nivel y se distingue por su característica fundamental de ser independiente de la plataforma.

Xamarin es una plataforma de desarrollo que ofrece la posibilidad de escribir y compilar aplicaciones completamente nativas para iOS, Android y Mac, compartiendo un código base desarrollado en el lenguaje C#. En [12] mencionan que utilizaron Xamarin por su capacidad de integrarse con Microsoft Visual Studio, permitiendo así la generación de aplicaciones también para Windows, abarcando Windows RT (RunTime) para tablets y Windows Phone para dispositivos móviles. Además, mencionan la versatilidad en la compatibilidad de plataformas.

PHP es un lenguaje de programación de código abierto ampliamente utilizado para el desarrollo web. En [12] se menciona la relevancia de PHP en este proyecto, además se aprovecha la capacidad multiplataforma, reforzando la importancia estratégica de este lenguaje en el contexto de desarrollo de sistemas modernos y versátiles.

En la tabla 6 se presenta el análisis, en el cual se enfocarán las características de los Lenguajes de Programación, con el propósito de facilitar la toma de decisiones.

Tabla 6. Cuadro comparativo de las herramientas para el Lenguaje de Programación. Fuente: Elaborado por autor

Características	Ruby	Javascript	C#	Java	Xamarin	PHP
Orientado a objetos			X	X	X	
Tipado estático			X	X	X	
Tipado dinámico	X	X				X
Compilado			X	X	X	
Interpretado	X	X				X
Gestión de memoria			X	X	X	
Multiplataforma		X	X	X	X	X
Amplia comunidad	X	X	X	X	X	X
Integración con SQL Server			X			
Integración con Visual Studio			X			
Desarrollo de aplicaciones de			X			

escritorio						
Robustes y estabilidad			X	X	X	
Facilidad de aprendizaje		X	X			X
Gestión de hilos			X	X	X	
Soporte de LINQ			X			
Soporte de WPF			X			
Soporte de WinForms			X			
Integración con .NET Framework			X			

En la tabla se muestran las características de los diferentes lenguajes de programación. Gracias a las ventajas que ofrece C#, se ha optado por seleccionar este lenguaje. C# se destaca por su tipado estático, lo que ayuda a detectar errores en tiempo de compilación. C# también se integra perfectamente con SQL Server y Visual Studio, proporcionando un entorno de desarrollo eficiente. La empresa indica que sería factible realizar el desarrollo con C#.

1.3.5 Arquitectura de Software

En [8] se menciona que las arquitecturas de software transforman los datos que recibe en un modelo del sistema, definiendo así los componentes del sistema, sus interfaces y sus relaciones.

MVC o Modelo - Vista - Controlador es un patrón de diseño de software que organiza la estructura de un sistema en tres componentes principales. En [6], [11] y [14], los autores optan por la utilización del patrón de arquitectura MVC, ya que obtienen beneficios para la estabilidad del sistema, facilitando la organización y el mantenimiento del código. Los autores mencionan que la arquitectura MVC desglosa el sistema en tres componentes clave:

- **Modelo:** Esta capa se encarga de consultar los datos en la base.
- **Vista:** Es la parte visual donde se representan los datos enviados por el controlador.
- **Controlador:** Actúa como la capa intermedia entre el modelo y la vista, gestionando las solicitudes, solicitando datos al modelo y comunicándolos a la vista.

En los artículos revisados los autores mencionan MVC como arquitectura de software, sin embargo, no se menciona la arquitectura Cliente-Servidor que desempeña un papel importante dentro de las implementaciones de software. La arquitectura Cliente – Servidor es un modelo de diseño de software que se basa en dos tipos de programas: los clientes, que son dispositivos o programas que solicitan y consumen servicios o recursos, y los servidores, que son dispositivos o programas que proporcionan esos servicios o recursos. En este modelo, los clientes envían las solicitudes y los servidores responden con los servicios requeridos. Esta comunicación puede ocurrir a través de una red, utilizando una variedad de protocolos de comunicación, incluso personalizados.

En la tabla 7 se presenta el análisis, en el cual se enfocarán las características de las Arquitecturas de Software, con el propósito de facilitar la toma de decisiones.

Tabla 7. Cuadro comparativo de las herramientas para la Arquitectura del Software.
Fuente: Elaborado por autor

Características	MVC	Cliente - Servidor
Separación de capas	X	X
Escalabilidad	X	X
Facilidad de mantenimiento	X	X
Reusabilidad de componentes	X	X
Seguridad	X	X
Interoperabilidad		X
Centralización		X
Desempeño		X
Flexibilidad		X
Conexión con base de datos		X
Compatibilidad con múltiples plataformas		X

En la tabla se muestran las características de las Arquitecturas de Software, gracias a las características que mantiene la arquitectura Cliente - Servidor, se ha optado por seleccionar esta herramienta ya que cumple con las características adecuadas para el desarrollo del proyecto.

1.3.6 Sistema Operativo

Los sistemas operativos gestionan los recursos o elementos de un dispositivo disponibles en la RAM (Random Access Memory), tarjetas de memoria o discos duros, además, se crean procesos multitarea y gestiona la carga del procesador.

Android es un sistema operativo móvil de código abierto, basado en el kernel de Linux. En [12], [13] y [14] mencionan que utilizaron Adroid debido a su amplia adaptabilidad, desde smartphones hasta televisores y automóviles. Además, mencionan que la estructura abierta de Android fomenta la innovación y la personalización, brindando a los fabricantes la libertad de ajustar la plataforma según sus necesidades.

En los artículos revisados los autores mencionan Android como sistema operativo, sin embargo, no se menciona el sistema operativo Windows que ha desempeñado un papel importante en el desarrollo de aplicaciones de escritorio. Además, su conjunto de herramientas de desarrollo, como Visual Studio, proporciona a los desarrolladores un entorno robusto para crear aplicaciones para el sistema operativo más utilizado en el mundo de la computación personal.

En la tabla 8 se presenta el análisis, en el cual se enfocarán las características de los Sistemas Operativos, con el propósito de facilitar la toma de decisiones.

Tabla 8. Cuadro comparativo de las herramientas para el Sistema Operativo.
Fuente: Elaborado por autor

Características	Android	Windows
Interfaz amigable	X	X
Soporte de aplicaciones Windows		X
Multiplataforma		X
Seguridad	X	X
Estabilidad	X	X
Facilidad de uso	X	X
Soporte de Hardware	X	X
Soporte de juegos	X	X
Soporte de red	X	X
Actualizaciones	X	X
Compatibilidad con software empresarial		X

En la tabla se muestran las características de Los Sistemas Operativos, gracias a las características que mantiene Windows, se ha optado por seleccionar esta herramienta ya que cumple con las características adecuadas para el desarrollo del proyecto.

1.3.7 Metodología Ágil

SCRUM, la metodología ágil mencionada en [6], [11] y [12], ha destacado por facilitar una comunicación efectiva en la obtención de requerimientos funcionales entre la parte administrativa y el equipo de trabajo. El autor destaca la importancia de reuniones constantes donde se visualizan las entregas planificadas o sprints aceptados, particularmente en el contexto del desarrollo de un sistema web para el registro de información relacionada con créditos. El uso de SCRUM, basado en los pilares de transparencia, inspección y adaptación, responde a la necesidad de automatizar procesos y renovar la gestión del buró de crédito de los clientes. Estos pilares aseguran la visibilidad y estándares definidos, identifican variaciones indeseadas y fomentan correcciones, así como promueven la adaptación del grupo de trabajo al entorno. Según el autor en [11], esta metodología no solo garantiza la calidad del software, sino que también contribuye al desarrollo eficiente y sostenible, aumentando la productividad del equipo para lograr un producto final de calidad en el dinámico campo del desarrollo de sistemas informáticos.

XP una metodología de desarrollo de software ágil que busca mejorar la calidad del software y la satisfacción del cliente. En [9] y [13] los autores se concentran en la aplicación efectiva de técnicas de programación, simplicidad, comunicación clara, trabajo en equipo y reutilización de código. Extreme Programming (XP), una forma destacada de esta metodología se distingue por su enfoque en cumplir requerimientos para generar valor al cliente, imponiendo restricciones que guían el desarrollo. Este método ágil, adaptable a equipos de cualquier tamaño, propone ciclos de desarrollo cortos con entregas funcionales al final de cada iteración, marcando una diferencia respecto a otras metodologías.

En la tabla 9 se presenta el análisis, en el cual se enfocarán las características de las Metodologías Agiles, con el propósito de facilitar la toma de decisiones.

Tabla 9. Cuadro comparativo de las herramientas para la Metodología Ágil.

Fuente: Elaborado por autor

Características	SCRUM	XP
Iteraciones cortas	X	X
Retroalimentación continua	X	X
Pruebas frecuentes		X
Refactorización constante		X
Programación en pareja		X
Entrega continua		X
Alta colaboración	X	X
Documentación mínima	X	X
Satisfacción del cliente	X	X
Adaptabilidad	X	X
Mejora continua	X	X
Simplicidad		X

En la tabla se muestran las características de las Metodologías Ágiles, gracias a las características que mantiene XP, se ha optado por seleccionar esta herramienta ya que cumple con las características adecuadas para el desarrollo del proyecto.

1.3.8 Pruebas de Software

Las pruebas garantizan que el software sea de alta calidad y que cumpla los requisitos especificados.

Las pruebas de aceptación son un tipo de prueba que se realiza para evaluar si un sistema cumple con los criterios de aceptación. En [11] el autor destaca la crucial etapa de pruebas en el desarrollo del sistema web para la Cooperativa de Ahorro y Crédito Credil Ltda., subrayando la importancia de determinar si el resultado de funcionalidad propuesto es el más adecuado y cumple con los criterios de mejora del servicio. En este contexto, se enfoca en el análisis de las pruebas de aceptación de las historias construidas en el sprint 3. El autor menciona la importancia atribuida a estas pruebas, ya que su papel es crucial para garantizar la calidad y la efectividad del sistema antes de su implementación definitiva.

En [13], se realizan pruebas adicionales al concluir cada iteración, y la decisión de llevarlo a producción recae en el cliente, según indica el autor. Estas pruebas de aceptación, segmentadas para la aplicación móvil y la de escritorio, se centran en

verificar el cumplimiento de los objetivos establecidos en las historias de usuario. Esto se consigue gracias a una fase de pruebas que evalúa específicamente la "Facilidad de uso percibida" por el grupo de evaluación. Los resultados obtenidos de las pruebas del aplicativo revelan un destacado promedio de utilidad de la aplicación de 4.6/5. Estos resultados indican una alta aceptación y facilidad de uso de la aplicación móvil, demostrando el éxito de las pruebas y la positiva acogida por parte de los usuarios.

1.3.9 Tipo de Investigación

La investigación cuantitativa es un enfoque de investigación que se centra en la recopilación y análisis de datos numéricos para comprender patrones, relaciones y tendencias en fenómenos específicos. En [12] se menciona que este tipo de investigación se basa en la medición objetiva y la aplicación de métodos estadísticos para generalizar resultados a poblaciones más amplias.

1.4 Objetivos

1.4.1 Objetivo general

Implantar un sistema para la Gestión de Riesgo de Liquidez Estructural en la Cooperativa de Ahorro y Crédito "San Martín"de Tisaleo Ltda., mediante la selección adecuada de herramientas de diseño y desarrollo de software, lo cual permita mejorar el nivel de usabilidad del sistema.

1.4.2 Objetivos específicos

- Identificar las herramientas adecuadas para la implementación de la aplicación de Gestión de Riesgo de Liquidez Estructural.
- Desarrollar una aplicación que automatice los procesos de la Gestión de Riesgo de Liquidez Estructural de la Cooperativa de Ahorro y Crédito “San Martín” de Tisaleo LTDA.

- Evaluar el nivel de usabilidad del sistema de Gestión de Riesgo de Liquidez Estructural conforme a las especificaciones proporcionadas por la Cooperativa.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

Los recursos materiales, se examinan evaluando la disponibilidad de las herramientas, equipos y tecnologías necesarias para el proyecto.

- Espacio de trabajo
- Internet
- Electricidad
- Laptop
- Hojas de papel bond
- Lápiz
- Esfero
- Transporte
- Impresora
- Celular

2.2 Métodos

2.2.1 Modalidad de la investigación

El desarrollo de este proyecto se llevará a cabo a través de una metodología de investigación, la cual ofrece el marco estructurado esencial para abordar de manera sistemática los objetivos establecidos. Para el presente proyecto se optó por la siguiente investigación:

Investigación Cuantitativa

La investigación será cuantitativa, ya que busca la obtención de datos precisos y medibles. Este método se distingue por su enfoque objetivo y estructurado, en el cual las variables se cuantifican y se someten a un análisis estadístico. Esto proporcionará una base sólida para la toma de decisiones informadas y presentar resultados obtenidos que se utilizarán para generar informes detallados.

2.2.2 Población y muestra

La población de estudio está conformada por los empleados que forman parte del proceso de la gestión de riesgo de liquidez estructural dentro de la Cooperativa de Ahorro y Crédito "San Martín"de Tisaleo LTDA. La composición de la población se desglosa de la siguiente manera:

- Administrador de Riesgos: 1 persona.
- Analistas de Riesgos: 1 personas.
- Departamento de Tecnología de la Información (TI): 2 personas

Justificación de la Muestra

Administrador de Riesgos: Esta persona sería responsable de supervisar y garantizar que el sistema automatizado funcione correctamente.

Analista de Riesgos: Profesional que realiza análisis en profundidad de la liquidez estructural, así como el cálculo de indicadores clave.

Departamento de Tecnología de la Información: Estos profesionales serán responsables de verificar que el sistema este correctamente desarrollado.

Población para la obtención de requerimientos

Con el propósito de llevar a cabo el proceso de obtención de requerimientos, se ha decidido incluir únicamente al administrador de riesgos y analista de riesgos, estas

personas serán entrevistadas con el fin de obtener información detallada y precisa para el desarrollo del proyecto.

Población para validación y pruebas

Para el proceso de validación y pruebas, se tomará en cuenta al administrador de riesgos, analista de riesgos y las dos personas del departamento de tecnologías. Esta decisión tiene como objetivo abarcar cada faceta y perspectiva relevante, garantizando así una validación integral y sólida de los datos recopilados.

2.2.3 Recolección de información

En el proceso de Recolección de Información, es importante emplear técnicas que permitan la recopilación de datos. La selección y aplicación de estas técnicas son fundamentales para asegurar la calidad y pertinencia de la información recabada. En el proyecto se utilizaron las siguientes técnicas de recolección de datos:

Entrevista Técnica

La entrevista técnica tendrá un análisis cualitativo, donde se profundizará en aspectos detallados para identificar las necesidades y expectativas de cada departamento. Esta técnica se destaca como un componente esencial en el proceso de obtención de información, especialmente en la recopilación de requerimientos. Para evaluar la consistencia y fiabilidad de los resultados, se aplicará una técnica de test y retest. Este método implica la repetición de la entrevista con los mismos participantes en diferentes momentos, permitiendo así validar la coherencia de las respuestas y minimizar posibles sesgos.

Encuesta

Las encuestas son técnicas esenciales para la validación y evaluación de la efectividad de la información recopilada. Para garantizar la validez de la encuesta, se aplicará el método de escala Likert. Estos indicadores estadísticos son esenciales para medir la consistencia interna y la confiabilidad de las respuestas, proporcionando así una base sólida para evaluar la validez de los datos recopilados a través de las encuestas.

Observación directa

La observación directa es una técnica para la recopilación de información, destacándose por su capacidad para obtener datos de manera inmediata y sin intermediarios. Esta técnica se basa en la observación directa de documentos, manuales y el flujo de trabajo manual, con el fin de obtener una comprensión profunda de los procesos operativos, las prácticas cotidianas. A través de esta técnica, se logra identificar de manera efectiva información clave, procedimientos y requerimientos esenciales.

2.2.4 Procesamiento y análisis de datos

El procesamiento y análisis de datos permiten extraer información valiosa para la toma de decisiones informadas. Esta fase es importante para comprender las necesidades y expectativas de los usuarios de cada departamento evaluado, identificar los requisitos del sistema y validar su efectividad.

Entrevista Técnica

La entrevista técnica permitirá una interacción directa con los usuarios finales y demás partes interesadas, brindando la oportunidad de obtener información detallada sobre sus necesidades. Además, se recopilarán datos que abarcan desde los requisitos funcionales hasta los no funcionales. A continuación, se presentan las preguntas que se realizaron en la entrevista técnica:

1. ¿Quiénes son los usuarios principales que usaran el sistema?
2. ¿De dónde se obtiene la información para el proceso de la gestión de riesgo de liquidez estructural?
3. ¿Cuál es el proceso que se realiza después de obtener los datos para la gestión de riesgo de liquidez estructural?
4. ¿Cuáles son los cálculos que se realizan dentro de la liquidez estructural?
5. ¿De dónde sacan las fórmulas o estructuras que siguen para los procesos?

6. ¿Cuáles son los documentos, plantillas, estructuras o fórmulas que establece la SEPS?

7. ¿Cuáles son los datos, reportes o documentos que se obtiene después del proceso?

8. ¿Hay otro documento que sigan para el proceso?

9. ¿Con que sistemas debe integrarse el producto que se realizara?

10. ¿Existen límites para los resultados finales del proceso?

11. ¿Cómo se establecen los límites y alertas para la gestión de riesgo de liquidez?

Encuesta

La utilización de encuestas es esencial para validar la efectividad del sistema propuesto. Los resultados de las encuestas permiten medir la satisfacción del usuario y la utilidad percibida

Observación Directa

La observación directa de documentos, manuales existentes y el flujo de trabajo se recopilan los requerimientos. Esta técnica permite identificar posibles áreas de mejora. La información recopilada mediante la observación directa se integrará con los datos obtenidos de las entrevistas técnicas para solventar todas las necesidades.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de resultados

3.1.1 Metodología Ágil para el desarrollo del sistema

El desarrollo del sistema se llevará a cabo utilizando la metodología ágil XP, seleccionada tras una comparación de herramientas detallada en la tabla 9. Esta metodología es especialmente adecuada para equipos pequeños de desarrollo de software, ya que se enfoca en la entrega de software de alta calidad. A continuación, se presentan las características principales de XP:

- **Satisfacción del cliente:** La interacción constante con el cliente permite priorizar los requerimientos del usuario.
- **Refactorización continua:** El código se refactoriza de manera continua para mantenerlo limpio y eficiente.
- **Simplicidad:** Se busca siempre la solución más simple y directa.
- **Calidad de software:** Las pruebas y la refactorización constantes aseguran un software de alta calidad.

3.1.2 Resultados de la entrevista técnica

Se realizo una entrevista técnica para el desarrollo del sistema para la Gestión de Riesgos de Liquidez Estructural de la Cooperativa de Ahorro y Crédito “San Martín” de Tisaleo Ltda. En la tabla 10 se presentan los resultados de la entrevista realizada.

Tabla 10. Resultados de la entrevista técnica.

Fuente: Elaborado por autor

N°	PREGUNTA	RESULTADO
1	¿Quiénes son los usuarios principales que usaran el sistema?	En el sistema actual, las personas que participan en los procesos son: Administrador de Riesgos y Analista de Riesgos

2	¿De dónde se obtiene la información para el proceso de la gestión de riesgo de liquidez estructural?	El conocimiento de la fuente de información utilizada en el sistema actual que administran es crucial. Los datos requeridos son extraídos del servidor de archivos interno de la empresa. Estos documentos son cargados en el servidor mediante una consulta ejecutada en la base de datos por un analista del departamento de sistemas.
3	¿Cuál es el proceso que se realiza después de obtener los datos para la gestión de riesgo de liquidez estructural?	El proceso inicia con la recolección de datos que luego se organizan en una hoja de cálculo de Excel, donde se eliminan los datos irrelevantes y se integran en una plantilla proporcionada por la SEPS. Esta tarea se repite diariamente, con los datos agregados a la primera hoja con la fecha correspondiente. En una segunda hoja del mismo Excel, se automatiza la extracción de datos de la primera hoja, y se realizan los cálculos necesarios para determinar la liquidez estructural de primera y segunda línea, mientras que en una tercera hoja se calcula la volatilidad. Finalmente, se efectúa el cálculo del 50% de los mayores depositantes, cada uno de los cuales requiere procedimientos y fórmulas específicas. Una vez que ya estén todos los datos en la hoja del Excel para por un proceso de redondeo en otro Excel y en un formato establecido por la SEPS se pegan lo datos y finalmente, se suben los datos a la página de la SEPS.
4	¿Cuáles son los cálculos que se realizan dentro de la liquidez estructural?	Los cálculos que se realizan dentro del proceso de la liquidez estructural son: • Liquidez estructural de primera línea • Liquidez estructural de segunda línea • Volatilidad • 50% de los mayores depositantes
5	¿De dónde sacan las fórmulas o estructuras que siguen para los procesos?	La SEPS es la que establece las estructuras y fórmulas que deben seguir las entidades financieras.
6	¿Cuáles son los documentos, plantillas, estructuras o fórmulas que establece la SEPS?	Los documentos que proporciona la SEPS son en formato xlsx para el proceso de la liquidez estructural, las plantillas son: • Reporte • Redondeo • Formato para subir
7	¿Cuáles son los datos, reportes o documentos que se obtiene después del proceso?	En el documento de Reporte se obtiene lo siguiente: • Liquidez estructural

		• Volatilidad • 50% de los mayores depositantes Los datos del documento de Reporte pasan por el documento de Redondeo y finalmente al documento del formato para subir.
8	¿Hay otro documento que sigan para el proceso?	Los documentos establecidos por la SEPS
9	¿Con que sistemas debe integrarse el producto que se realizara?	El sistema debe integrarse con el Core Financiero de la Cooperativa
10	¿Existen límites para los resultados finales del proceso?	Para los resultados de la liquidez estructural de primera línea y la liquidez estructural de segunda línea se establecen los limites Alto, Medio y Bajo.
11	¿Cómo se establecen los límites y alertas para la gestión de riesgo de liquidez?	Con los porcentajes diarios de los resultados se realizan cálculos anuales para establecer los niveles de riesgo, los cálculos son los siguientes: • Promedio • Moda • Desviación estándar • Datos observados • Máximo • Mínimo
CONCLUSIÓN: Las personas encargadas de la gestión de riesgos en la Cooperativa de Ahorro y Crédito "San Martín" de Tisaleo Ltda. llevan a cabo sus procesos de manera manual. Esto resulta en una inversión considerable de tiempo, obstaculiza los procedimientos y ralentiza la toma de decisiones.		

3.1.3 Levantamiento de Requerimientos Funcionales

Se realizó el levantamiento de requerimientos funcionales. Estos permitirán especificar las acciones que el sistema debe cumplir para satisfacer las necesidades del usuario. Además, debe ser capaz de satisfacer los objetivos para el desarrollo del

sistema. A continuación, se listan los requerimientos funcionales para la implementación del sistema:

- Inicio de Sesión
- Cambio de Contraseña
- Gestionar Usuario
- Gestionar Rol
- Gestionar Menú Sistema
- Gestionar acceso a acciones de Menú por rol de usuario
- Gestionar Empleado
- Reporte Empleados
- Gestionar Feriados
- Reporte de Liquidez Estructural
- Reporte de Volatilidad
- Reporte de Mayores Depositantes
- Reporte de Indicadores
- Reporte de Balances Diarios
- Cerrar Sesión

3.1.4 Levantamiento de Requerimientos No Funcionales

Se realizó el levantamiento de los requerimientos no funcionales. Estos permitirán especificar las características que no se relacionan directamente con la funcionalidad del sistema. Los requerimientos no funcionales son igual de importantes para su desempeño, usabilidad y seguridad. A continuación, se listan los requerimientos no funcionales para la implementación del sistema:

- El sistema debe garantizar la seguridad de los datos
- Debe tener tiempos de respuesta rápidos
- La interfaz de usuario debe ser fácil de usar e intuitivo
- El diseño de la interfaz debe corresponder a las pautas de identidad de la institución.

3.1.5 Levantamiento de Historias de Usuario

Una vez definidas las funcionalidades generales, se presentan las historias de usuario. Las mismas que utilizan un lenguaje natural para una mejor comprensión de lo que necesita el usuario. A continuación, en las siguientes tablas se presentan las historias de usuario.

Tabla 11. Historia de usuario 1: Inicio de Sesión.

Fuente: Elaborado por autor

	Historia de Usuario
Código: H01	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Inicio de Sesión	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Media
Programador Responsable: Mariela Amanta	
Descripción	
El usuario necesita abrir el sistema para que se le muestre la pantalla de inicio de sesión, donde deben ingresar las credenciales para poder ingresar a la pantalla principal del sistema.	
	Criterios de Aceptación
• Cuando el usuario ingrese al sistema se mostrará la pantalla de inicio de sesión, donde debe ingresar el usuario y contraseña, el sistema verificara las credenciales y se mostrará la pantalla principal del sistema. • Cuando el usuario ingrese al sistema se mostrará la pantalla de inicio de sesión, donde debe ingresar el usuario y contraseña, el sistema verificara las credenciales, si el usuario y contraseña son incorrectas, el sistema emitirá un mensaje de alerta.	

Tabla 12. Historia de usuario 2: Cambio de Contraseña.

Fuente: Elaborado por autor

	Historia de Usuario
Código: H02	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Cambio de contraseña	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Alta
Programador Responsable: Mariela Amanta	
Descripción	
En la interfaz de “Cambio de Contraseña” el usuario podrá visualizar los siguientes campos: Contraseña actual, Nueva contraseña, Confirmar Contraseña, que son los campos para poder cambiar la contraseña.	
Criterios de Aceptación	
• Cuando el usuario esté en la pantalla "Cambiar Contraseña", deberá ingresar la contraseña actual, la nueva contraseña que debe tener más de 6 caracteres y combinar números y letras; y la confirmación de la nueva contraseña. Una vez	

completados todos los campos correctamente, se podrá guardar la nueva contraseña. • Cuando el usuario esté en la pantalla "Cambiar Contraseña", deberá ingresar la contraseña actual, la nueva contraseña, si se ingresan menos de 6 caracteres en la nueva contraseña, aparecerá una alerta indicando que debe ingresar más de 6 caracteres. Por otro lado, si la confirmación de la contraseña no coincide con la nueva contraseña, se mostrará una alerta indicando que las contraseñas no coinciden.

Tabla 13. Historia de usuario 3: Gestionar Usuario.

Fuente: Elaborado por autor

	Historia de Usuario
Código: H03	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Gestionar Usuarios	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Media
Programador Responsable: Mariela Amanta	
Descripción	
En la interfaz de “Usuarios” se podrán visualizar los usuarios existentes, se podrá buscarlos por el código de empleado. Además, se podrá Crear, Modificar y Resetear un usuario.	
	Criterios de Aceptación
• Cuando el usuario este en la pantalla “Usuarios” podrá visualizar la tabla con los usuarios que se han creado hasta el momento. • Cuando el usuario este en la pantalla “Usuarios” podrá dirigirse al rectángulo celeste e ingresar el código del empleado, en caso de no saber el código del empleado, el usuario puede presionar la tecla F2 para que se muestre la pantalla de búsqueda por apellidos y nombres, se ingresan los nombres y se escoge el empleado que está buscando. • Cuando el usuario este en la pantalla “Usuarios” y presione el botón “Nuevo” se mostrará el formulario de crear usuario, donde se deben escoger e ingresar los siguientes datos: Departamento, Empleado, Rol, Agencia, Dirección IP, Si es o No Administrador, una vez ingresado los campos correctamente se creará el usuario. • Cuando se encuentre en la pantalla “Usuarios”, en la tabla donde están registrados todos los usuarios, al presionar el botón “Modificar” del usuario que desee, se muestra la pantalla “Modificar usuario” donde se podrán cambiar los campos: Dirección IP, Administrador, Estado, Cerrar Sesión. • Cuando se encuentre en la pantalla “Usuarios”, en la tabla donde están registrados todos los usuarios, al presionar el botón “Resetear” del usuario que desee, se muestra un mensaje de confirmación para resetear la contraseña, poniendo una por defecto.	

Tabla 14. Historia de Usuario 4: Gestionar Rol.

Fuente: Elaborado por autor

Historia de Usuario	
Código: H04	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Gestionar Rol	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Media
Programador Responsable: Mariela Amanta	
Descripción	
En la interfaz de "Rol", el usuario podrá visualizar los roles existentes. Además, se podrá Crear, Modificar y Eliminar un rol.	
Criterios de Aceptación	
• Cuando el usuario este en la pantalla "Rol" podrá visualizar todos los roles que existen hasta ahora • Cuando el usuario este en la pantalla "Rol", en la parte derecha podrá crear un rol nuevo • Cuando el usuario este en la pantalla "Rol", al escoger uno de los roles existentes en la parte derecha de la pantalla se podrá modificar el rol • Cuando el usuario este en la pantalla "Rol" se podrá eliminar un rol	

Tabla 15. Historia de usuario 5: Gestionar Menú Sistema

Fuente: Elaborado por autor

Historia de Usuario	
Código: H05	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Gestionar Menú Sistema	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Media
Programador Responsable: Mariela Amanta	
Descripción	
En la interfaz de "Menú Sistema", el usuario podrá visualizar los menús y los ítems de los menús existentes. Además, se podrá Crear, Modificar y Eliminar un menú.	
Criterios de Aceptación	
• Cuando el usuario este en la pantalla "Menú Sistema" podrá visualizar todos los menús y los ítems del menú que existen hasta ahora • Cuando el usuario este en la pantalla "Menú Sistema", en la parte derecha podrá crear un menú y un ítem del menú • Cuando el usuario este en la pantalla "Menú Sistema", al escoger uno de los menús o los ítems del menú existentes en la parte derecha de la pantalla se podrá modificar el menú • Cuando el usuario este en la pantalla "Menú Sistema" se podrá eliminar un menú o un ítem del menú	

Tabla 16. Historia de usuario 6: Gestionar acceso a acciones de Menú por Rol de usuario.

Fuente: Elaborado por autor

	Historia de Usuario
Código: H06	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Gestionar acceso a acciones de Menú por Rol de usuario	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Media
Programador Responsable: Mariela Amanta	
Descripción	
En la interfaz de "Menú Rol", el usuario podrá asociar el menú con el rol	
	Criterios de Aceptación
• Cuando el usuario este en la pantalla "Menú Rol" podrá visualizar todos los menús y los ítems del menú que existen y podrá asociar con los roles que existen.	

Tabla 17. Historia de usuario 7: Gestionar Empleado.

Fuente: Elaborado por autor

	Historia de Usuario
Código: H07	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Gestionar Empleado	
Prioridad en Negocio: Media	**Riesgo en Desarrollo:** Baja
Programador Responsable: Mariela Amanta	
Descripción	
En la interfaz "Empleado", el usuario podrá Crear y Modificar un empleado	
	Criterios de Aceptación
• Cuando el usuario este en la pantalla "Empleado", al escoger la opción Crear Nuevo el usuario ingresará los datos necesarios en la pestaña empleado, laboral y formación, una vez llenado todos los datos correctamente se creará el nuevo empleado • Cuando el usuario este en la pantalla "Empleado", al escoger la opción Modificar Existente el usuario debe buscar al empleado por el código del empleado o F2 para habilitar la pantalla de búsqueda, una vez que se haya encontrado al empleado se procede a modificar los datos que necesite y se realizan los cambios	

Tabla 18. Historia de usuario 8: Reporte Empleados.

Fuente: Elaborado por autor

	Historia de Usuario
Código: H08	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Reporte Empleados	
Prioridad en Negocio: Media	**Riesgo en Desarrollo:** Baja

Programador Responsable: Mariela Amanta
Descripción
En la interfaz “Nomina”, el usuario podrá visualizar el estado de empleados
Criterios de Aceptación
• Cuando el usuario este en la pantalla “Nomina” podrá visualizar una tabla con el listado de los empleados y todos los datos registrados de cada uno de ellos.

Tabla 19. Historia de usuario 9: Gestionar feriados.

Fuente: Elaborado por autor

	Historia de Usuario
Código: H09	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Gestionar Feriados	
Prioridad en Negocio: Media	**Riesgo en Desarrollo:** Baja
Programador Responsable: Mariela Amanta	
Descripción	
En la interfaz de “Feriados”, el usuario podrá visualizar un calendario de todo el año y la cantidad de feriados que hasta la fecha se han dado. Además, el usuario podrá Crear, Modificar y Eliminar un feriado del calendario.	
Criterios de Aceptación	
• Cuando el usuario este en la pantalla “Feriados” podrá visualizar el calendario de todo el año, al presionar en una fecha se muestra una pantalla donde se puede Crear un nuevo feriado y agregar una descripción. • Cuando el usuario este en la pantalla “Feriados” podrá visualizar el calendario de todo el año, al presionar en una fecha que sea feriado, se muestra una pantalla para Modificar la descripción del feriado. • Cuando el usuario este en la pantalla “Feriados” podrá visualizar el calendario de todo el año, al presionar en una fecha que sea feriado, se muestra una pantalla para Eliminar ese feriado.	

Tabla 20. Historia de usuario 10: Reporte de Liquidez Estructural.

Fuente: Elaborado por autor

	Historia de Usuario
Código: H10	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Reporte de Liquidez Estructural	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Media
Programador Responsable: Mariela Amanta	
Descripción	
En la interfaz de “Liquidez Estructural” el usuario podrá generar y visualizar los reportes para evaluar la Liquidez Estructural de la cooperativa	
Criterios de Aceptación	
• Cuando el usuario este en la pantalla “Liquidez Estructural” al seleccionar una fecha, el sistema generara el reporte de Liquidez Estructural	

Tabla 21. Historia de usuario 11: Reporte de Volatilidad.

Fuente: Elaborado por autor

Historia de Usuario	
Código: H11	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Reporte de Volatilidad	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Media
Programador Responsable: Mariela Amanta	
Descripción	
En la interfaz de "Volatilidad" el usuario podrá generar y visualizar los reportes para evaluar la Volatilidad de la cooperativa	
Criterios de Aceptación	
• Cuando el usuario este en la pantalla "Volatilidad" al seleccionar una fecha, el sistema generara el reporte de Volatilidad	

Tabla 22. Historia de usuario 12: Reporte de Mayores Depositantes.

Fuente: Elaborado por autor

Historia de Usuario	
Código: H12	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Reporte de Mayores Depositantes	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Media
Programador Responsable: Mariela Amanta	
Descripción	
En la interfaz de "Mayores Depositantes" el usuario podrá generar y visualizar los reportes para evaluar los Mayores Depositantes de la cooperativa	
Criterios de Aceptación	
• Cuando el usuario este en la pantalla "Mayores Depositantes" al seleccionar una fecha, el sistema generara el reporte de Mayores Depositantes	

Tabla 23. Historia de usuario 13: Reporte de Indicadores.

Fuente: Elaborado por autor

Historia de Usuario	
Código: H13	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Reporte de Indicadores	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Media
Programador Responsable: Mariela Amanta	
Descripción	
En la interfaz de "Indicadores" el usuario podrá generar y visualizar los reportes de los indicadores de la cooperativa	
Criterios de Aceptación	
• Cuando el usuario este en la pantalla "Indicadores" al seleccionar una fecha, el sistema generara el reporte de los indicadores	

Tabla 24. Historia de usuario 14: Reporte de Balances Diarios.

Fuente: Elaborado por autor

	Historia de Usuario
Código: H14	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Reporte de Balances Diarios	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Media
Programador Responsable: Mariela Amanta	
Descripción	
En la interfaz de "Balances Diarios" el usuario podrá generar y visualizar los reportes de los Balances Diarios de la cooperativa	
	Criterios de Aceptación
• Cuando el usuario este en la pantalla "Balances Diarios" al seleccionar una fecha, el sistema generara el reporte de los balances diarios	

Tabla 25. Historia de usuario 15: Cerrar Sesión.

Fuente: Elaborado por autor

	Historia de Usuario
Código: H15	**Usuario:** Administrador de Riesgos Analista de Riesgos
Nombre de la Historia: Cerrar Sesión	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Baja
Programador Responsable: Mariela Amanta	
Descripción	
Como usuario, quiero cerrar sesión de manera segura para proteger mi cuenta.	
	Criterios de Aceptación
• Cuando el usuario presione la opción Cerrar Sesión se le mostrará un mensaje de confirmación y se precederá a cerrar la sesión con éxito	

3.2 Diseño

3.2.1 Arquitectura de software del sistema

El sistema de escritorio para la automatización de los procesos de gestión de riesgos de liquidez estructural de la Cooperativa de Ahorro y Crédito "San Martín" de Tisaleo Ltda. se desarrolló utilizando el modelo Cliente-Servidor. En este modelo de diseño de software, las tareas se distribuyen entre clientes y servidores. Los

servidores son responsables de proporcionar los servicios, mientras que los clientes los consumen.

Los componentes principales del sistema incluyen el cliente, el servidor y la red. El cliente es el sistema o interfaz de usuario que solicita los servicios al servidor. Por otro lado, el servidor es la parte del sistema que recibe las solicitudes de servicios. La red es el medio a través del cual los clientes y servidores se comunican, facilitando la interacción entre ambos.

En el caso de la Cooperativa de Ahorro y Crédito "San Martín" de Tisaleo Ltda., el sistema de escritorio se conecta a dos servidores. El primero es el Servidor Principal, que es donde se encuentra el sistema y la base de datos. El segundo servidor es el Servidor de Archivos, encargado de almacenar y gestionar los documentos y datos necesarios. La estructura se muestra en la figura 1.

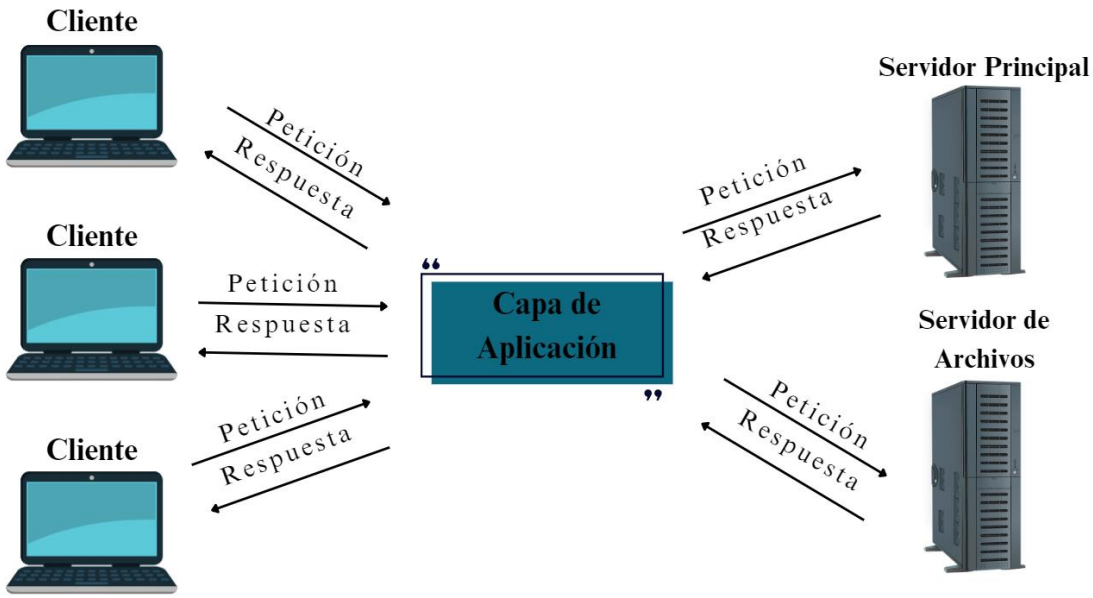


Figura 1. Arquitectura de software del sistema para la automatización de procesos.

Fuente: Elaborado por autor

3.2.2 Diseño de la Base de Datos

La base de datos que maneja la Cooperativa de Ahorro y Crédito “San Martín” de Tisaleo Ltda. es extensa y sigue una nomenclatura establecida para la creación de tablas. Para el desarrollo del sistema de automatización de los procesos de gestión de riesgo de liquidez estructural, se utilizan 16 tablas, que son:

- COR_FIN_USUARIO
- COR_FIN_ESTADO_USUARIO
- DTH_EMP_EMPLEADO
- DTH_EMP_ESTADO_EMPLEADO
- DTH_EMP_CARGO
- DTH_EMP_DEPARTAMENTO
- COR_FIN_AUDITORIA_INICIO_SESION
- COR_FIN_ESTADO_SESION
- COR_FIN_ROL
- COR_FIN_MENU_ROL
- COR_FIN_MENU
- COR_FIN_AGENCIA
- COR_FIN_ESTADO_AGENCIA
- DFI_CONT_BALANCE
- URI_LIQ_INDICADOR
- URI_LIQ_MAYOR_DEPOSITANTE

Los atributos de cada tabla y sus relaciones se muestran en la figura 2.

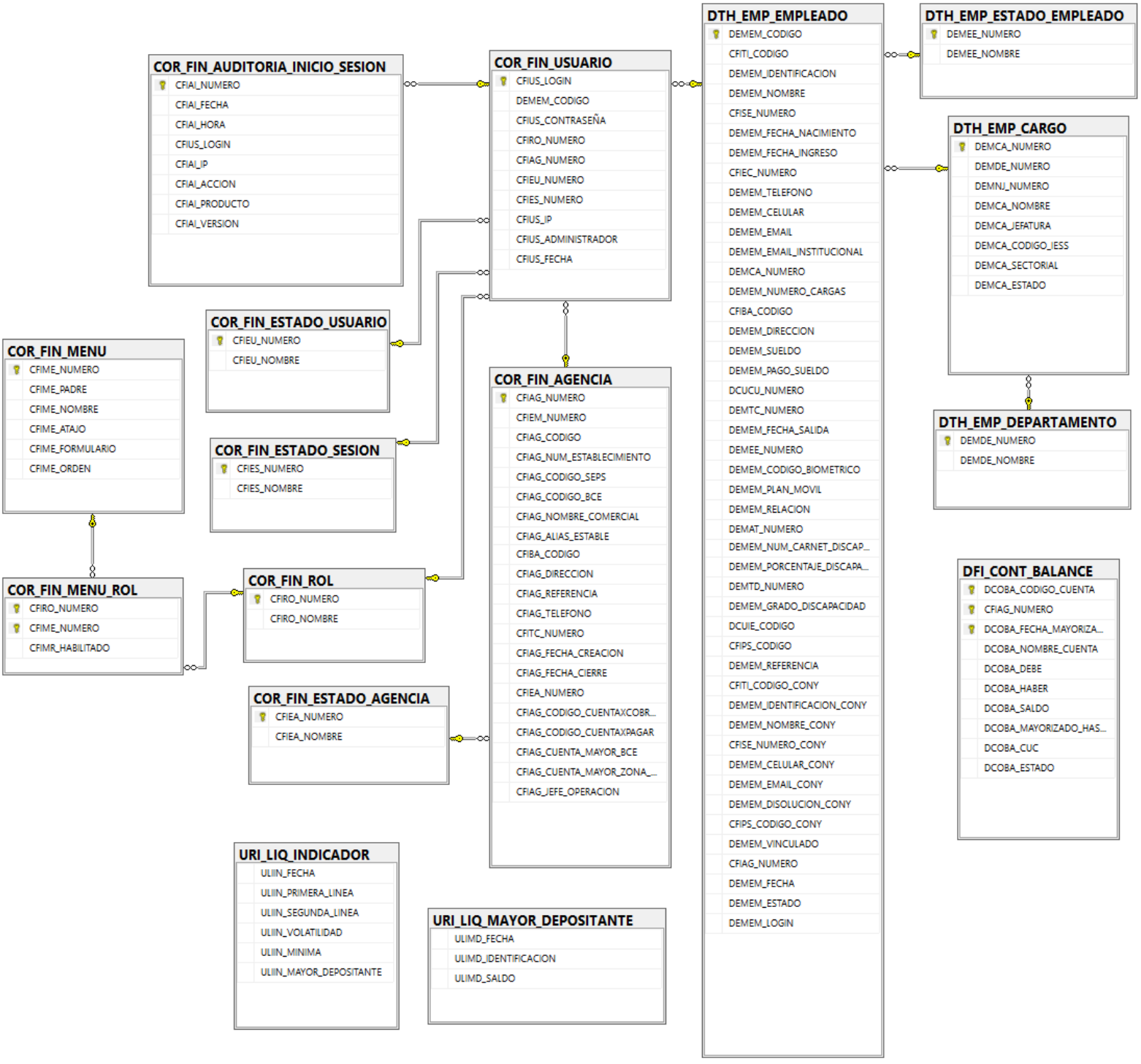


Figura 2. Diseño de la Base de Datos para el sistema.

Fuente: Elaborado por autor

Por otro lado, dentro de la base de datos se cuenta con tres tablas que no tienen relaciones con otras tablas, que son: DFI_CONT_BALANCE, URI_LIQ_INDICADOR Y URI_LIQ_MAYOR_DEPOSITANTE. Estas tablas están aisladas ya que contienen datos calculados a partir de otras tablas. Además, están diseñadas para almacenar información, de modo que, al consultar los datos necesarios para los reportes, el proceso no sea tan tardío. Las tablas se presentan en la figura 3.

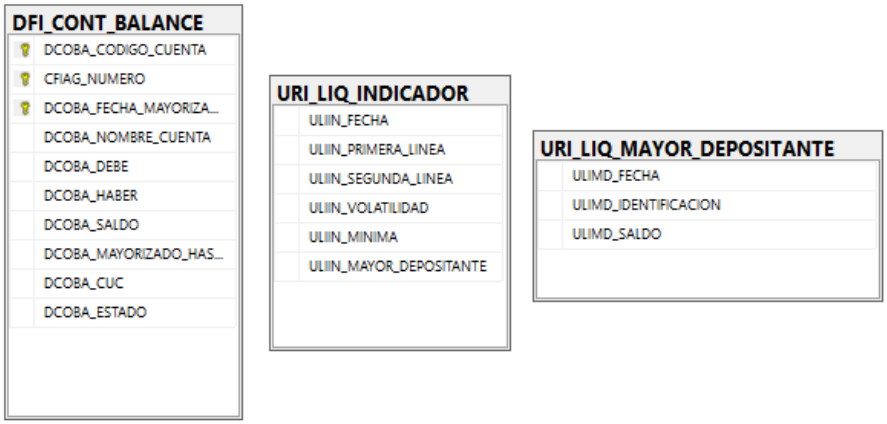


Figura 3. Tablas sin relación de la Base de datos.

Fuente: Elaborado por autor

3.2.3 Diagramas de Caso de Uso

Los diagramas de caso de uso ilustran los diferentes escenarios de interacción entre los usuarios y el sistema. Estos diagramas permiten comprender como los usuarios pueden realizar diferentes tareas y procesos dentro del sistema. A continuación, en las figuras 4 y 5 se presenta el diagrama para el Administrador de Riesgos y para el Analista de Riesgos.

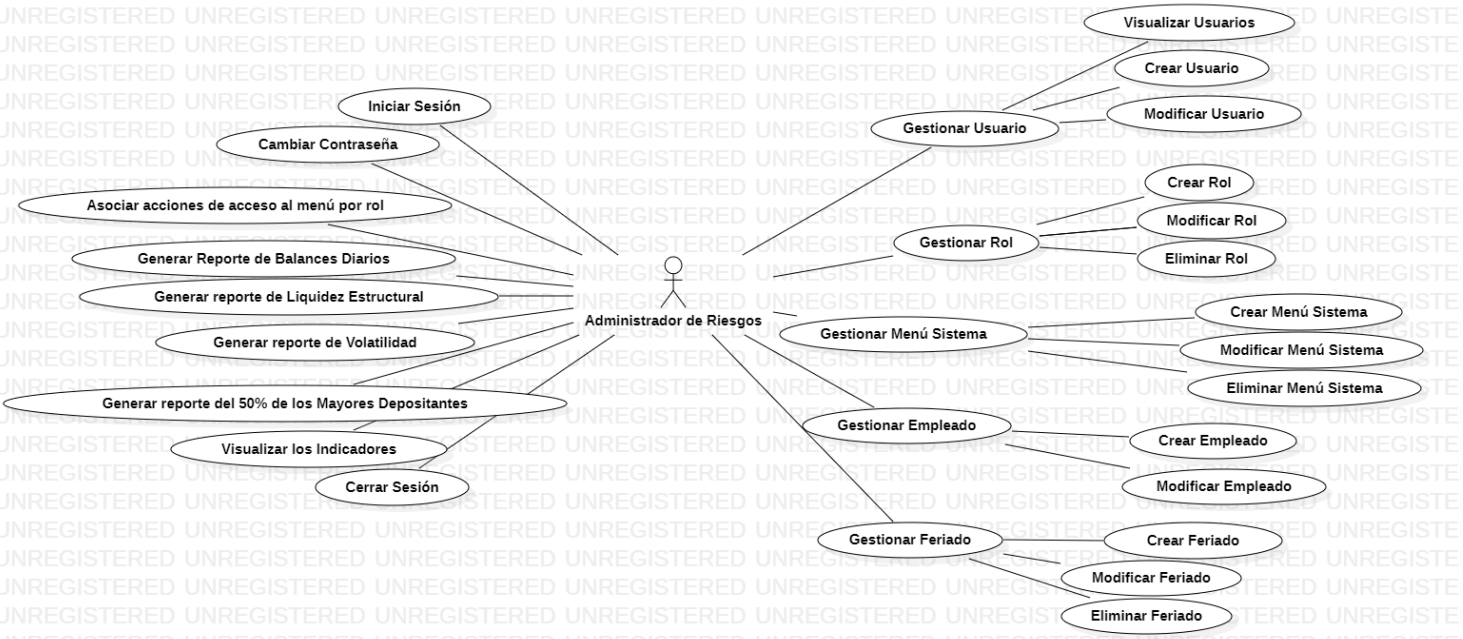


Figura 4. Diagrama de Caso de uso del Administrador.

Fuente: Elaborado por autor

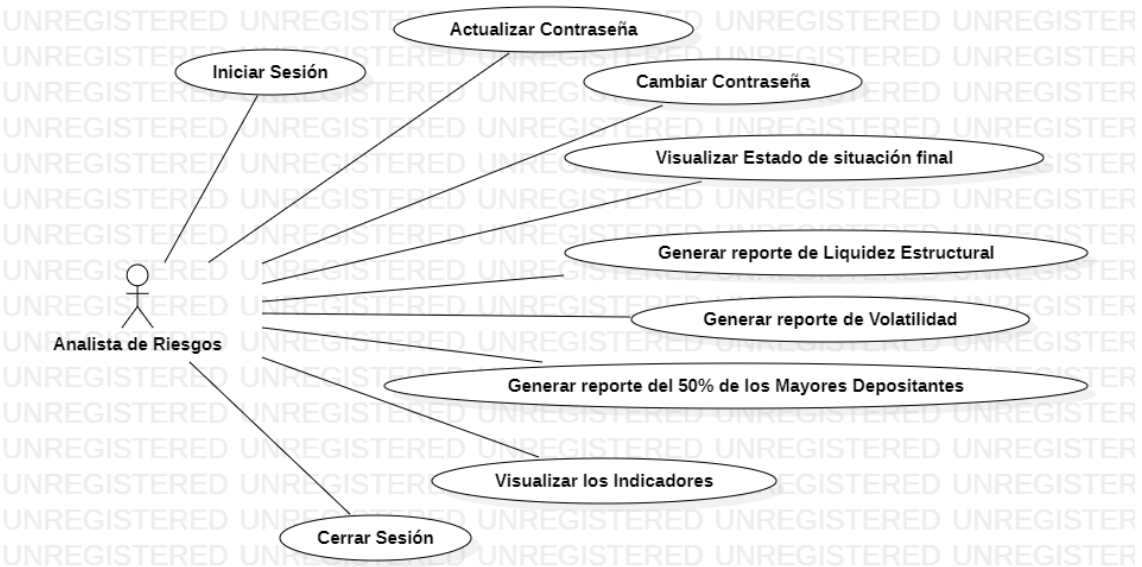


Figura 5. Diagrama de Caso de uso del Analista de Riesgos.

Fuente: Elaborado por autor

3.2.4 Diagramas de Secuencia

Los diagramas de secuencia ilustran la interacción temporal que se da entre los componentes y los actores del sistema. Estos diagramas permiten visualizar los eventos y mensajes en el orden en que ocurren, proporcionando el conocimiento necesario de cómo se llevan a cabo los procesos dentro del sistema.

En la figura 6 se presenta el diagrama de secuencia para el Inicio de Sesión, el proceso comienza cuando el usuario ingresa sus credenciales en la interfaz. Estos datos se envían a la base de datos para su búsqueda y validación, asegurando que las credenciales estén correctamente digitalizadas. Además, se verifica el rol al que pertenece el usuario. Una vez completada la validación, se habilitan las opciones correspondientes según el rol del usuario y se muestra la pantalla principal.

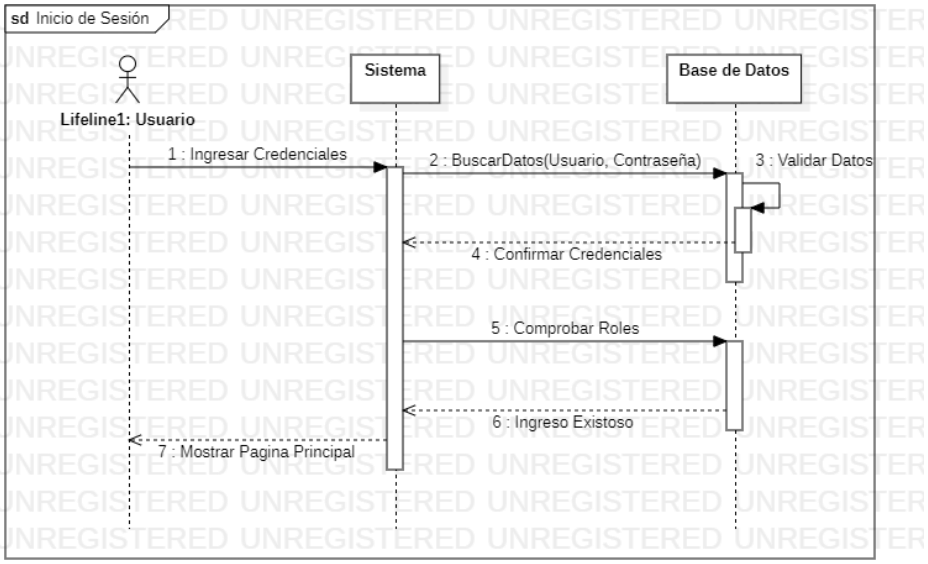


Figura 6. Diagrama de secuencia de Inicio de Sesión

Fuente: Elaborado por autor

En la figura 7 se presenta el diagrama de secuencia para el Cambio de Contraseña, el proceso comienza cuando el usuario selecciona la opción de "Cambio de Contraseña". El sistema muestra un formulario donde el usuario debe ingresar los datos requeridos. Tras ingresar los datos correctamente, el sistema los envía para su validación en la base de datos. Una vez validados, los cambios se aplican con éxito.

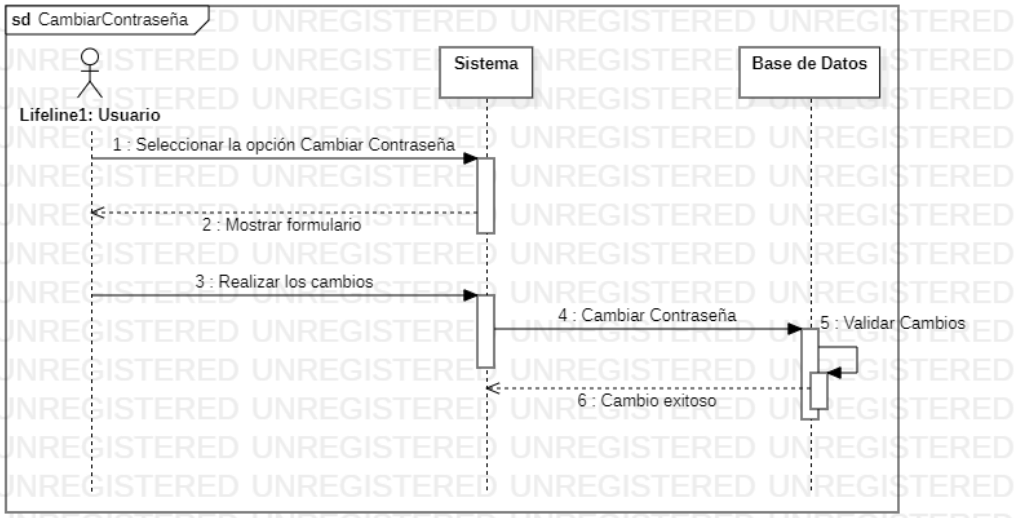


Figura 7. Diagrama de secuencia de Cambiar Contraseña

Fuente: Elaborado por autor

En la figura 8 se presenta el diagrama de secuencia para Gestionar Usuario, el proceso comienza cuando el usuario selecciona la opción “Usuarios”. El sistema muestra la pantalla donde se podrá crear, buscar, modificar o resetear un usuario, al seleccionar la opción que quiera realizar se le mostrará la interfaz acorde a la opción que escogió, y el usuario deberá ingresar o seleccionar los datos correctamente para que se validen y se pueda realizar el proceso con éxito.

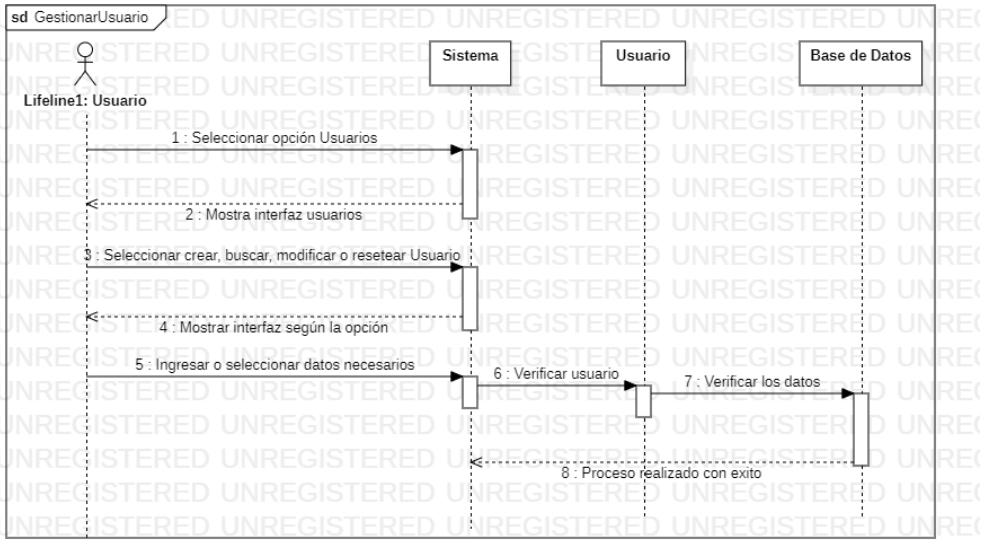


Figura 8. Diagrama de secuencia Gestionar Usuario

Fuente: Elaborado por autor

En la figura 9 se presenta el diagrama de secuencia para Gestionar Rol, el proceso comienza cuando el usuario selecciona la opción “Rol”. El sistema muestra la pantalla donde el usuario podrá ingresar o seleccionar un rol. Después el sistema mostrara las opciones Guardar o Elominar. La acción que se ejecute dependerá de la opción que escogió. Siempre y cuando haya ingresado los datos correctamente.

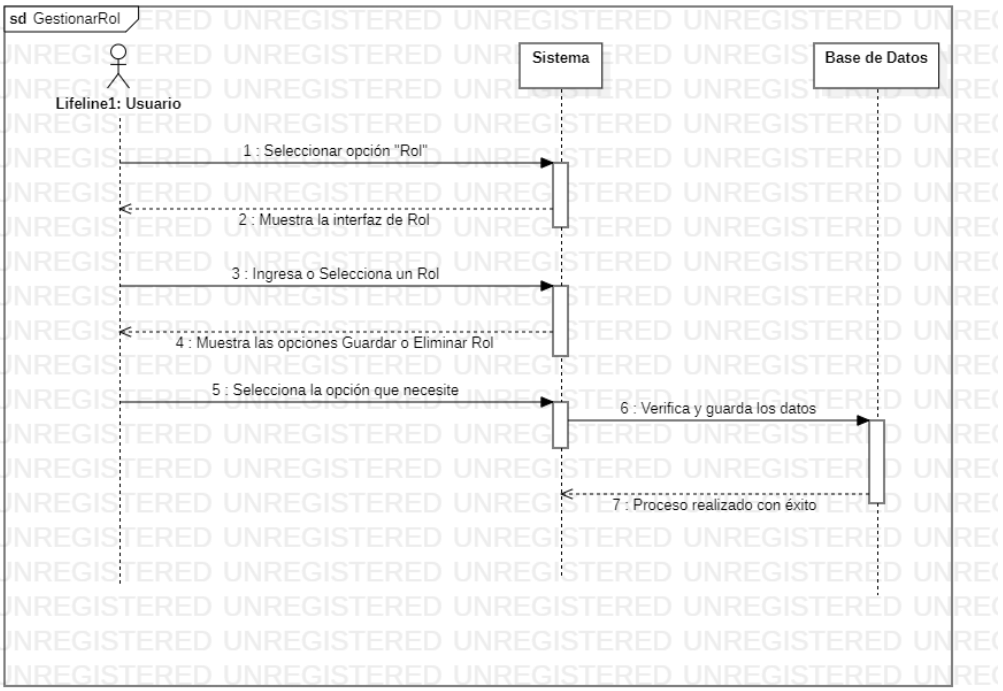


Figura 9. Diagrama de secuencia Gestionar Rol

Fuente: Elaborado por autor

En la figura 10 se presenta el diagrama de secuencia para Gestionar Menú Sistema, el proceso comienza cuando el usuario selecciona la opción “Menú Sistema”. El sistema muestra la pantalla donde el usuario podrá ingresar o seleccionar un menú. Después, el sistema mostrara las opciones Guardar o Elominar. La acción que se ejecute dependerá de la opción que escogió. Siempre y cuando haya ingresado los datos correctamente.

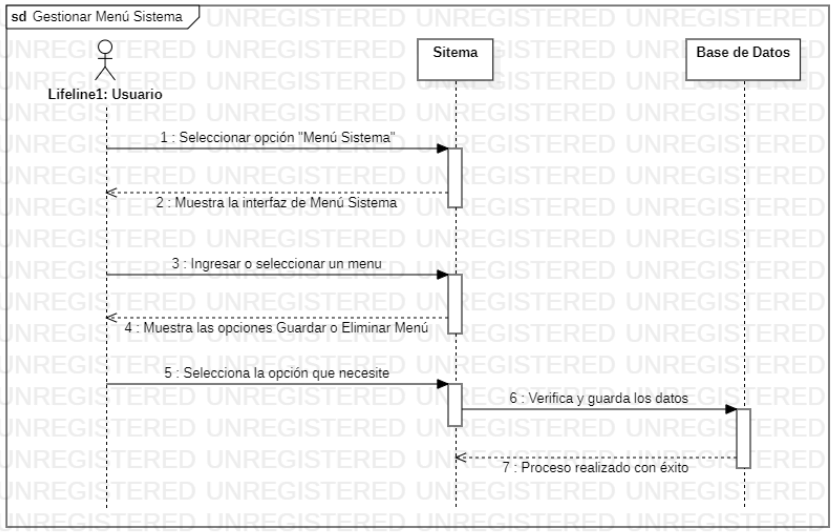


Figura 10. Diagrama de secuencia Gestionar Menú Sistema

Fuente: Elaborado por autor

En la figura 11 se presenta el diagrama de secuencia para Gestionar el acceso a acciones de Menú por Rol de usuario. El proceso comienza cuando el usuario selecciona la opción “Menú Rol”. El sistema muestra la pantalla donde el usuario podrá seleccionar un menú y un rol. Después, el sistema mostrara las opciones si está Habilitado o no y Guardar. Finalmente, el sistema relaciona el menú con el rol seleccionado.

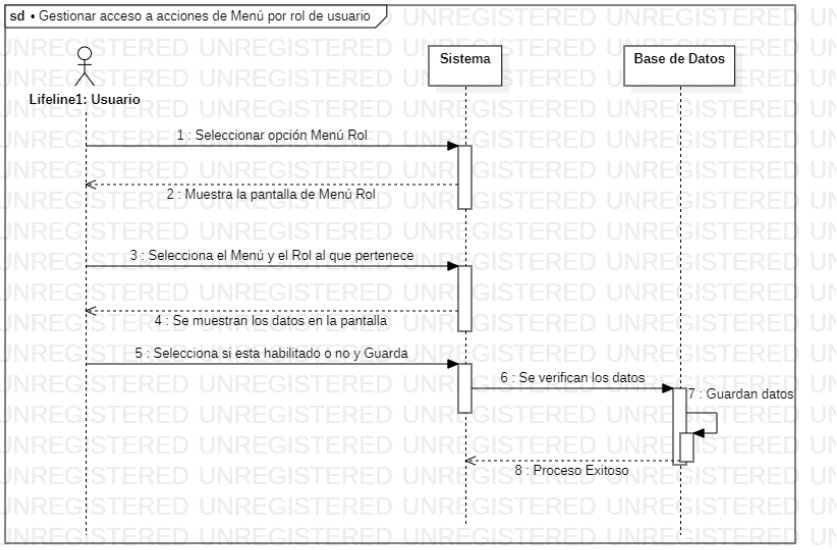


Figura 11. Diagrama de secuencia Acceso a acciones de Menú por Rol

Fuente: Elaborado por autor

En la figura 12 se presenta el diagrama de secuencia para Gestionar Empleado. El proceso comienza cuando el usuario selecciona la opción “Empleado”. El sistema muestra la pantalla donde el usuario podrá Crear o Modificar un empleado. Al seleccionar una de las opciones, el usuario deberá ingresar o modificar los datos y guardarlos.

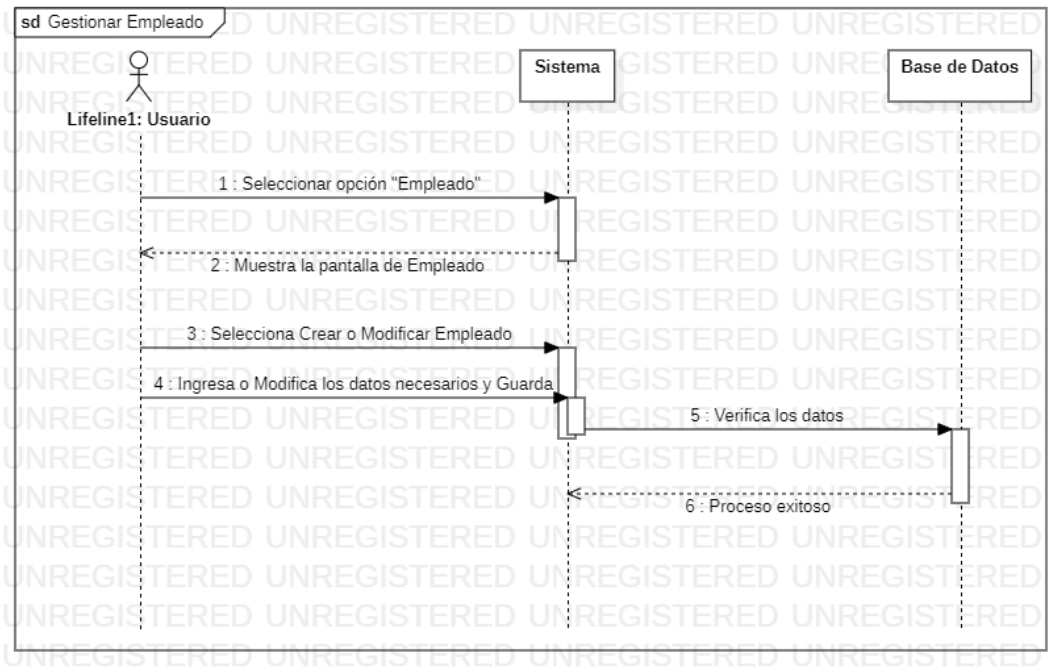


Figura 12. Diagrama de secuencia Gestionar Empleado

Fuente: Elaborado por autor

En la figura 13 se presenta el diagrama de secuencia de Reporte de Empleados. El proceso comienza cuando el usuario selecciona la opción “Nómina”. Se consultan los datos y se muestra el reporte de empleados. Además, el usuario podrá filtrar los datos por Estado o Nómina. Al seleccionar uno de los filtros, se consultarán los datos y se presentara el reporte de Empleados.

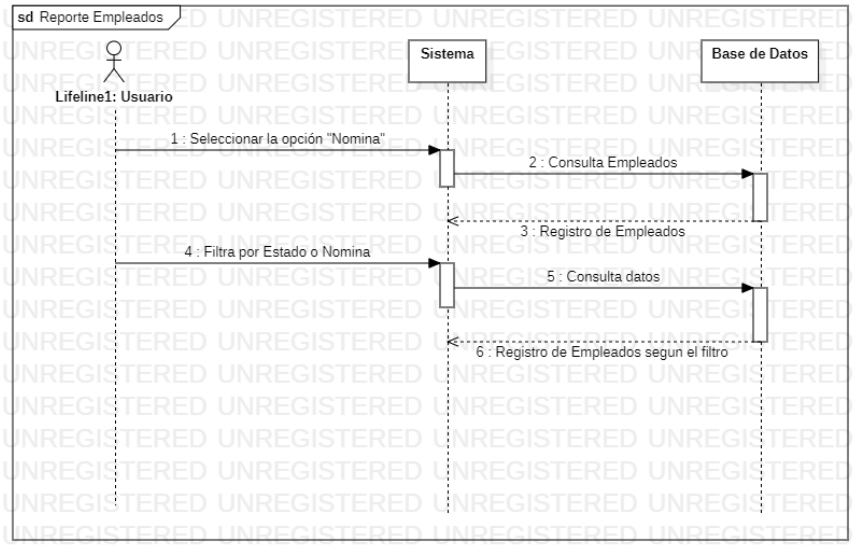


Figura 13. Diagrama de secuencia Reporte Empleados

Fuente: Elaborado por autor

En la figura 14 se presenta el diagrama de secuencia para Gestionar Feriado. El proceso comienza cuando el usuario selecciona la opción “Feriado”. El sistema muestra la pantalla de feriados donde está un calendario. Al seleccionar una de las fechas, el sistema muestra una pantalla donde se podrá Crear, Modificar o Eliminar un feriado. El usuario podrá realizar la acción que desee y se guardaran los datos.

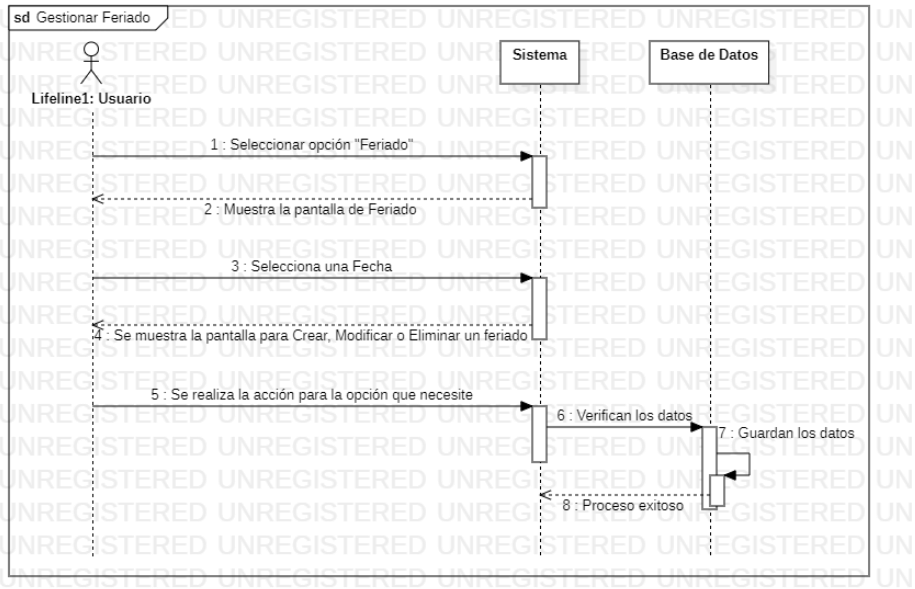


Figura 14. Diagrama de secuencia Gestionar Feriado

Fuente: Elaborado por autor

En la figura 15 se presenta el diagrama de secuencia de Reporte de Liquidez Estructural. El proceso comienza cuando el usuario selecciona la opción “Liquidez Estructural”. El sistema muestra la pantalla donde se presentará el reporte. Al seleccionar la fecha que desee internamente se ejecutara el SP para el reporte, se devolverán los datos calculados y se presentara el reporte de Liquidez Estructural.

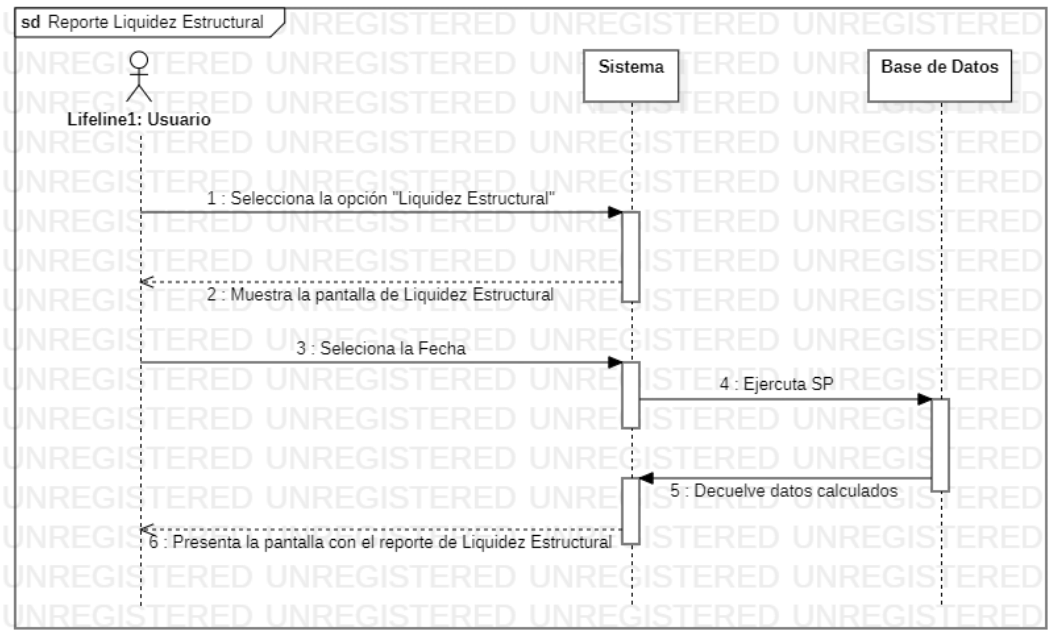


Figura 15. Diagrama de secuencia Reporte Liquidez Estructural

Fuente: Elaborado por autor

En la figura 16 se presenta el diagrama de secuencia de Reporte de Volatilidad. El proceso comienza cuando el usuario selecciona la opción “Volatilidad”. El sistema muestra la pantalla donde se presentará el reporte. Al seleccionar la fecha que desee, internamente se ejecutara el SP para el reporte, se devolverán los datos calculados y se presentara el reporte de Volatilidad.

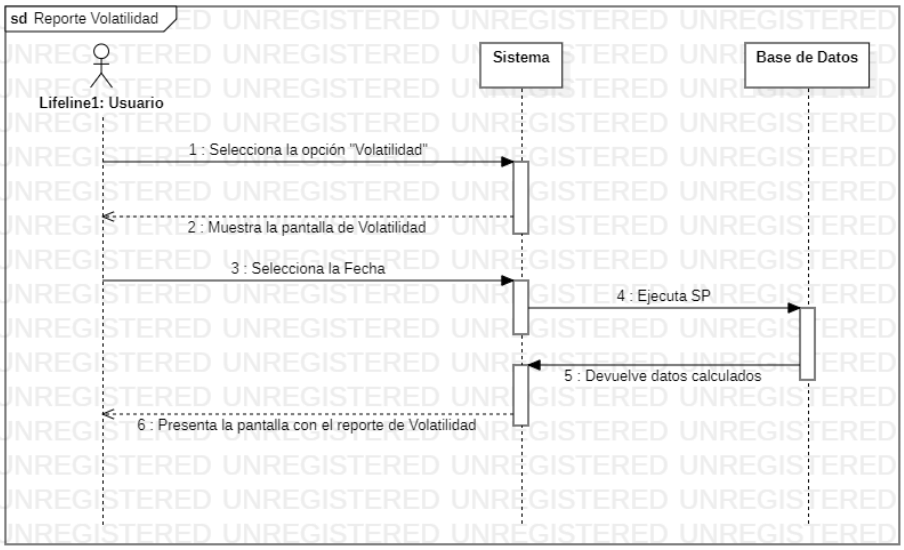


Figura 16. Diagrama de secuencia Reporte Volatilidad

Fuente: Elaborado por autor

En la figura 17 se presenta el diagrama de secuencia de Reporte de Mayores Depositantes. El proceso comienza cuando el usuario selecciona la opción “Mayores Depositantes”. El sistema muestra la pantalla donde se presentará el reporte. Al seleccionar la fecha que desee internamente se ejecutara el SP para el reporte, se devolverán los datos calculados y se presentara el reporte de Mayores Depositantes.

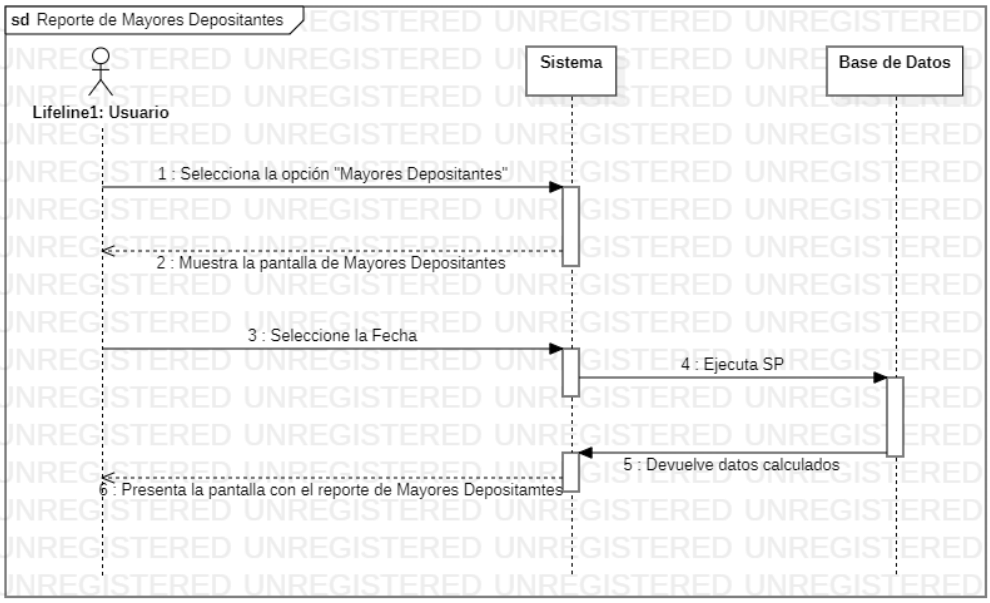


Figura 17. Diagrama de secuencia Reporte Mayores Depositantes

Fuente: Elaborado por autor

En la figura 18 se presenta el diagrama de secuencia de Reporte de Indicadores. El proceso comienza cuando el usuario selecciona la opción “Indicadores”. El sistema muestra la pantalla donde se presentará el reporte. Al seleccionar la fecha que desee, internamente se ejecutara el SP para el reporte, se devolverán los datos calculados y se presentara el reporte de Indicadores.

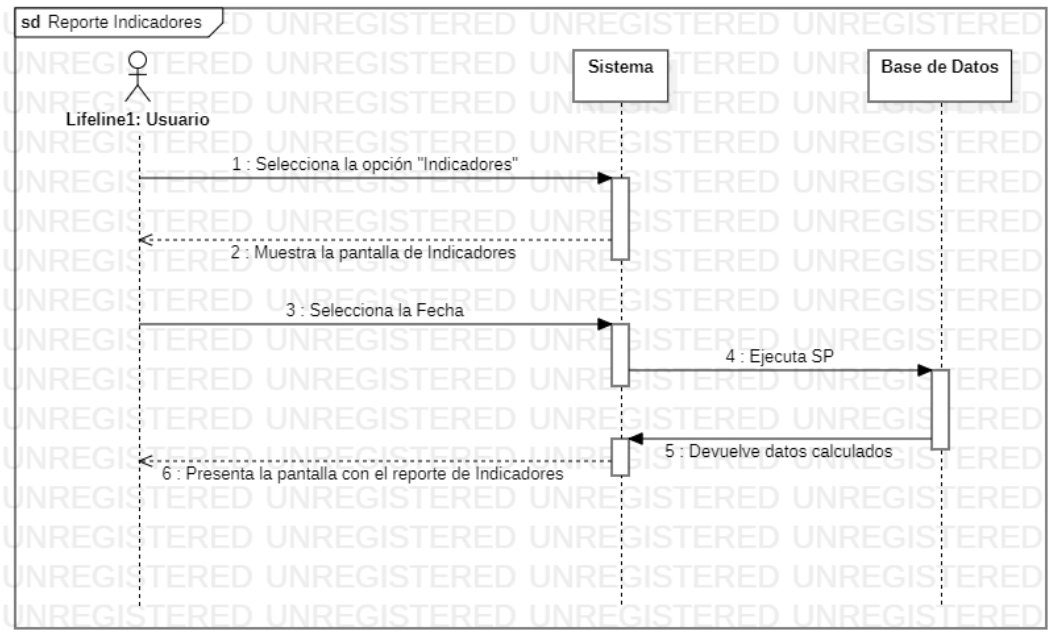


Figura 18. Diagrama de secuencia Reporte Indicadores

Fuente: Elaborado por autor

En la figura 19 se presenta el diagrama de secuencia de Reporte de Balances Diarios. El proceso comienza cuando el usuario selecciona la opción “Balances Diarios”. El sistema muestra la pantalla donde se presentará el reporte. Al seleccionar la fecha que desee internamente se ejecutara el SP para el reporte, se devolverán los datos calculados y se presentara el reporte de Balances Diarios.

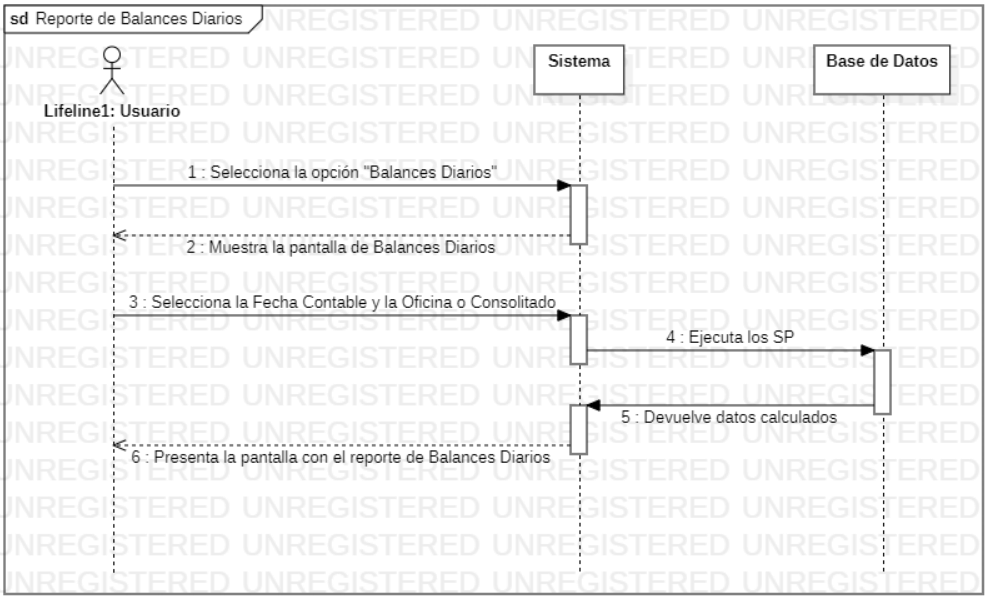


Figura 19. Diagrama de secuencia Reporte Balances Diarios

Fuente: Elaborado por autor

En la figura 20 se presenta el diagrama de secuencias de Cerrar Sesión. El usuario selecciona la opción cerrar sesión, el sistema emite un mensaje de confirmación. Una vez que el usuario confirme el mensaje, el sistema finalizara la sesión.

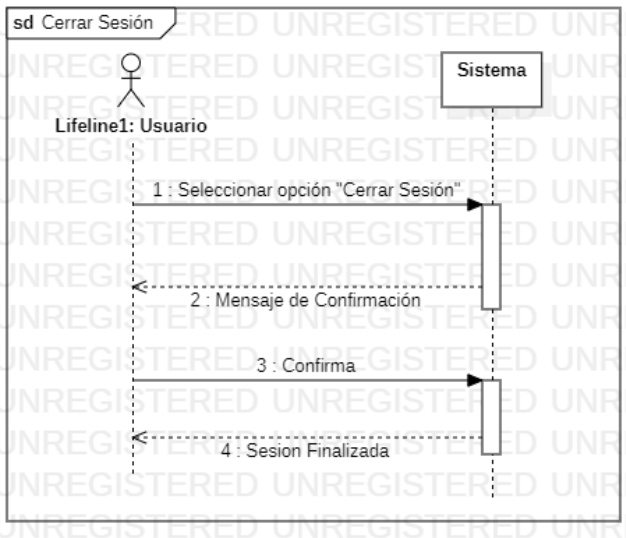


Figura 20. Diagrama de secuencia Cerrar Sesión

Fuente: Elaborado por autor

3.2.5 Diseño de la Interfaz

Se presenta el diseño de la interfaz del sistema, que detalla la disposición visual de los elementos del sistema. Los componentes y colores utilizados en el diseño forman parte de los requerimientos no funcionales del usuario, ya que deben representar simbólicamente a la institución. Estos elementos deben alinearse con la identidad visual de la organización, utilizando colores y componentes que indiquen su pertenencia a la institución. Cada una de las pantallas ha sido diseñada por el autor del presente proyecto.

Inicio de Sesión

En la figura 21 se presenta la pantalla de inicio de sesión donde el usuario podrá ingresar el Usuario y Contraseña que se le haya otorgado.



Figura 21. Diseño de la Pantalla de Inicio de Sesión.

Fuente: Elaborado por autor

Pantalla Principal

En la figura 22 se presentar la Pantalla Principal. Una vez el que usuario ingresa las credenciales en la pantalla de inicio de sesión se muestra la pantalla principal donde podrá visualizar todas las opciones a las que tiene acceso.



Figura 22. Diseño de la Pantalla Principal

Fuente: Elaborado por autor

Cambiar Contraseña

En la figura 23 se presenta la pantalla Cambiar Contraseña, el usuario deberá ingresar la Contraseña Actual, una Nueva Contraseña y Confirmar la Contraseña, una vez ingresado los datos correspondientes se podrá cambiar la contraseña correctamente.

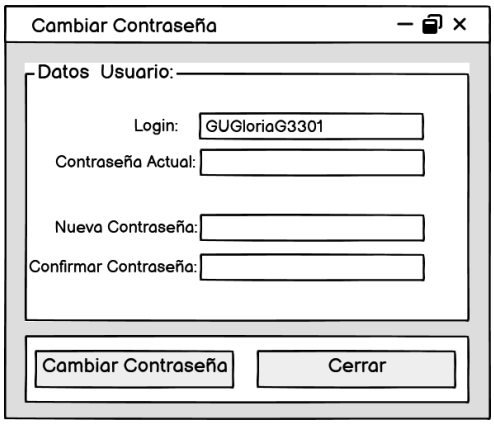


Figura 23. Diseño de la Pantalla de Cambio de Contraseña

Fuente: Elaborado por autor

Usuarios

En la figura 24 se presenta la pantalla Usuarios, el usuario podrá crea, buscar, modificar y resetear un usuario, además, podrá seleccionar filtros como: Estado, Sesión y Administrador.

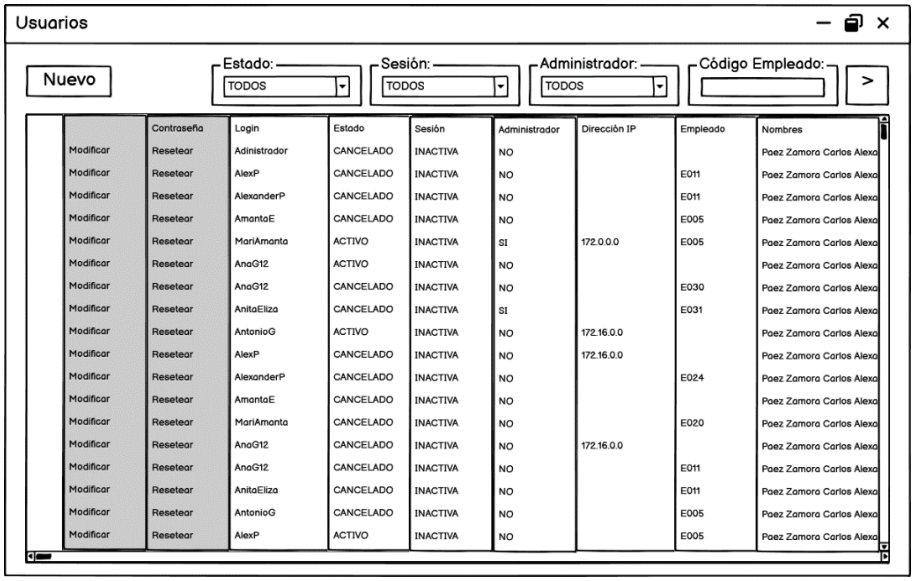


Figura 24. Diseño de la Pantalla de Gestionar Usuarios.

Fuente: Elaborado por autor

Crear Usuario

En la figura 25 se presenta la pantalla Crear Usuario, el usuario deberá ingresar y seleccionar los campos establecidos en la siguiente pantalla, como: Departamento, Empleado, Rol, Agencia, Dirección IP, si es o no Administrador, la contraseña ya se le establece por defecto.

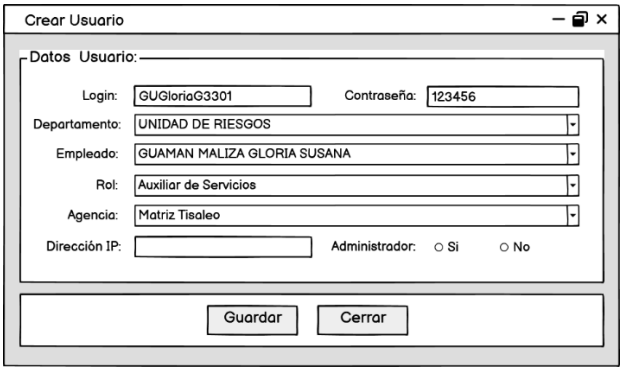


Figura 25. Diseño de la Pantalla de Crear Usuario.

Fuente: Elaborado por autor

Modificar Usuario

En la figura 26 se presenta la pantalla Modificar Usuario, el usuario podrá modificar la Dirección IP, si es o no Administrador, el Estado y Cerrar la Sesión.

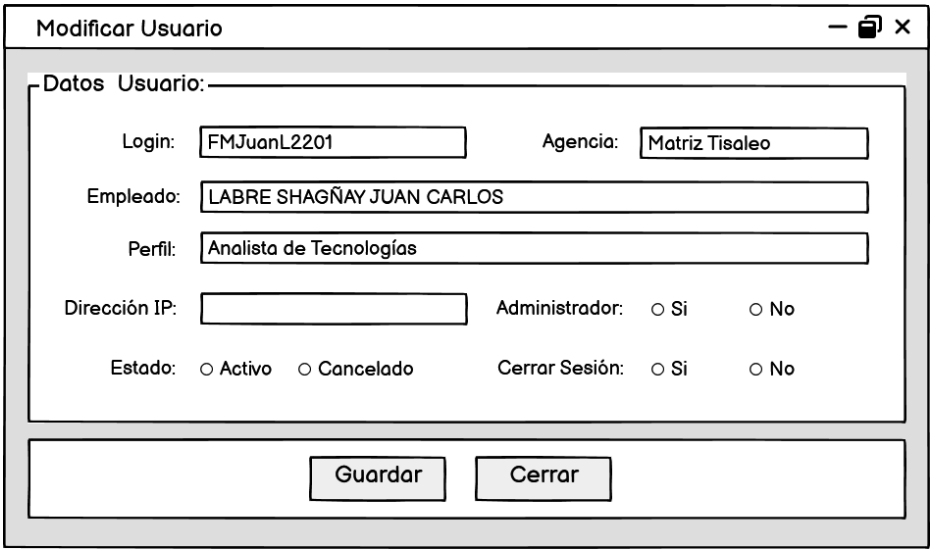


Figura 26. Diseño de la Pantalla de Modificar Usuario.

Fuente: Elaborado por autor

Rol

En la figura 27 se presenta la pantalla Rol, el usuario podrá crear un rol, seleccionar un rol para modificarlo y eliminar un rol.

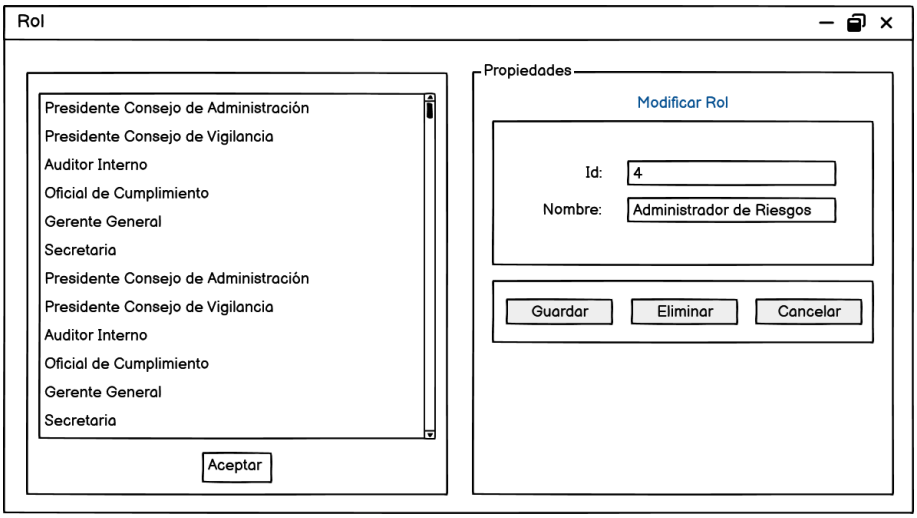


Figura 27. Diseño de la Pantalla de Rol.

Fuente: Elaborado por autor

Menú Sistema

En la figura 28 se presenta la pantalla Menú Sistema, el usuario podrá crear, seleccionar un menú o ítem para modificarlo y eliminar un menú o ítem del sistema.

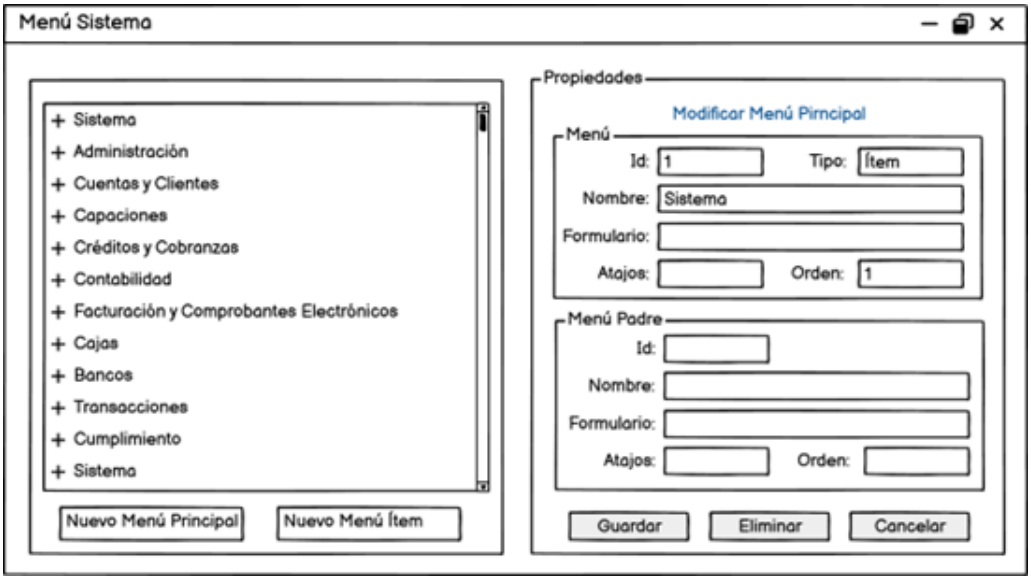


Figura 28. Diseño de la Pantalla de Menú Sistema.

Fuente: Elaborado por autor

Menú Rol

En la figura 29 se presenta la pantalla Menú Rol, el usuario podrá asociar el menú con el rol para que cada usuario tenga un menú personalizado según el rol que desempeña.

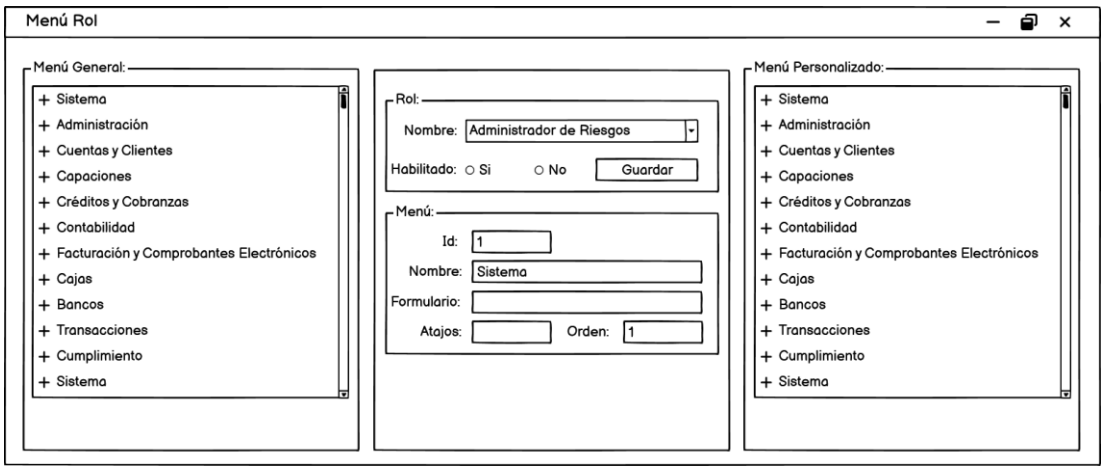


Figura 29. Diseño de la Pantalla de Menú Rol.

Fuente: Elaborado por autor

Empleado

En la figura 30 se presenta la pantalla de Empleado, el usuario podrá crear o modificar un empleado, al ingresar los datos necesarios.

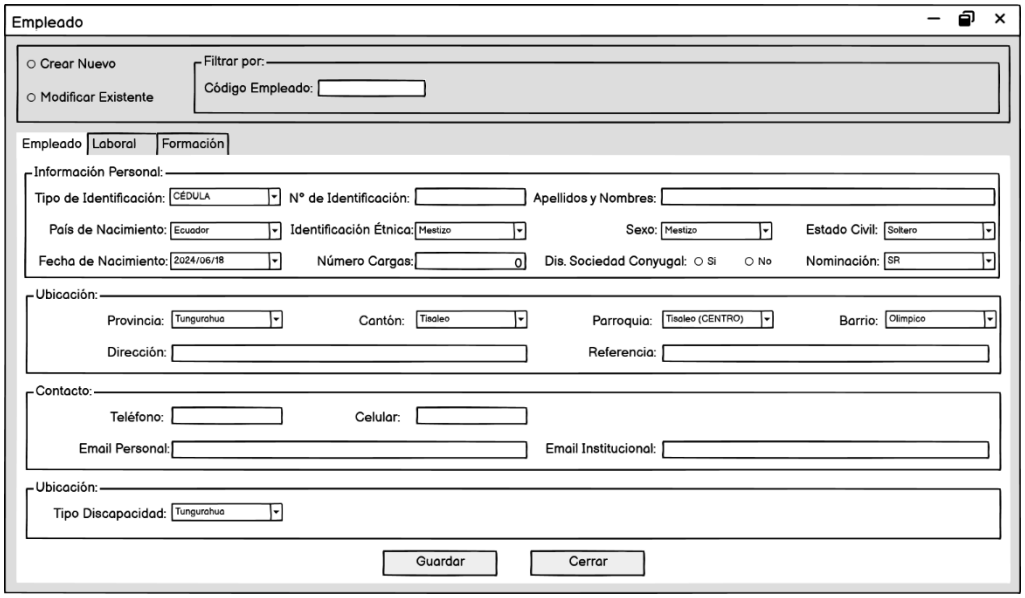
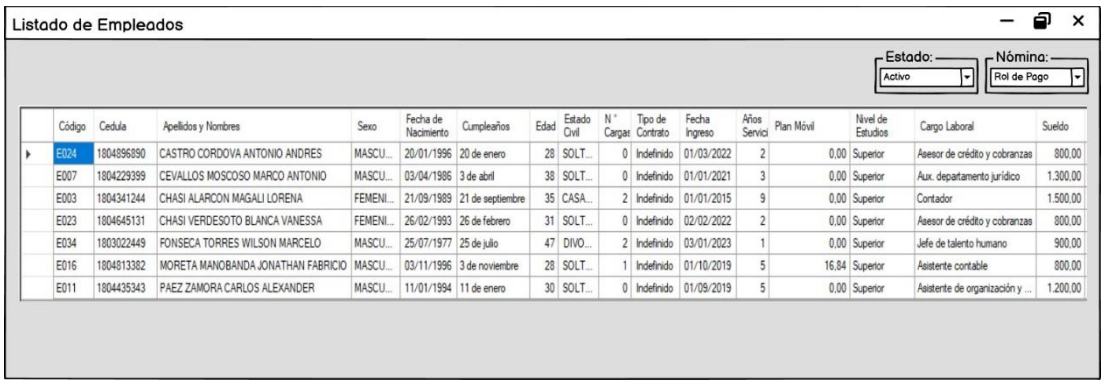


Figura 30. Diseño de la Pantalla de Empleado

Fuente: Elaborado por autor

Nómina

En la figura 31 se presenta la pantalla del Listado de Empleados, el usuario podrá visualizar la lista de los empleados con sus datos.



Código	Cedula	Apellidos y Nombres	Sexo	Fecha de Nacimiento	Cumpleaños	Edad	Estado Civil	N° Cargas	Tipo de Contrato	Fecha Ingreso	Años Servicio	Plan Móvil	Nivel de Estudios	Cargo Laboral	Sueldo
E024	1804896890	CASTRO CORDOVA ANTONIO ANDRES	MASCU...	20/01/1996	20 de enero	28	SOLT...	0	Indefinido	01/03/2022	2	0,00	Superior	Asesor de crédito y cobranzas	800,00
E007	1804229399	CEVALLOS MOSCOSO MARCO ANTONIO	MASCU...	03/04/1986	3 de abril	38	SOLT...	0	Indefinido	01/01/2021	3	0,00	Superior	Aux. departamento jurídico	1.300,00
E003	1804341244	CHASI ALARCON MAGALI LORENA	FEMENI...	21/09/1989	21 de septiembre	35	CASA...	2	Indefinido	01/01/2015	9	0,00	Superior	Contador	1.500,00
E023	1804645131	CHASI VERDESOTO BLANCA VANESSA	FEMENI...	26/02/1993	26 de febrero	31	SOLT...	0	Indefinido	02/02/2022	2	0,00	Superior	Asesor de crédito y cobranzas	800,00
E034	1803022449	FONSECA TORRES WILSON MARCELO	MASCU...	25/07/1977	25 de julio	47	DIVO...	2	Indefinido	03/01/2023	1	0,00	Superior	Jefe de talento humano	900,00
E016	1804813382	MORETA MANOBANDA JONATHAN FABRICIO	MASCU...	03/11/1996	3 de noviembre	28	SOLT...	1	Indefinido	01/10/2019	5	16,84	Superior	Asistente contable	800,00
E011	1804435343	PAEZ ZAMORA CARLOS ALEXANDER	MASCU...	11/01/1994	11 de enero	30	SOLT...	0	Indefinido	01/09/2019	5	0,00	Superior	Asistente de organización y ...	1.200,00

Figura 31. Diseño de la Pantalla de Listado de Empleados

Fuente: Elaborado por autor

Feriados

En la figura 32 se presenta la pantalla de Feriados, el usuario podrá visualizar la cantidad de feriados que existe en el periodo.

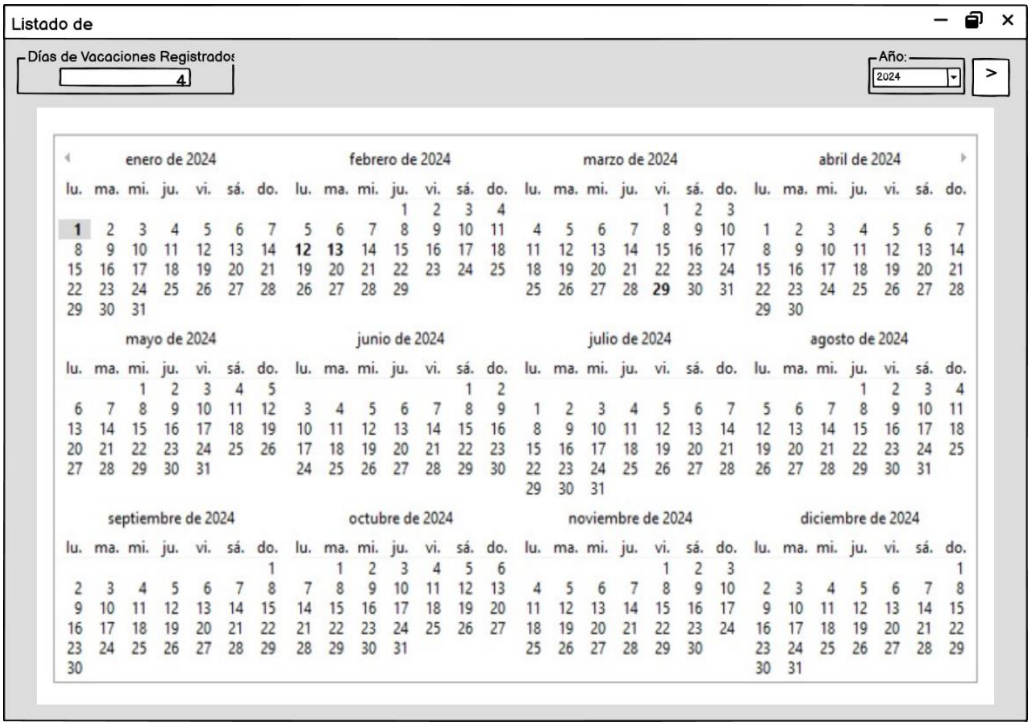


Figura 32. Diseño de la Pantalla de Feriados

Fuente: Elaborado por autor

Feriado

En la figura 33 se presenta la pantalla de Feriado, el usuario podrá crear, modificar y eliminar un feriado en la fecha seleccionada, además, podrá ingresar una descripción.

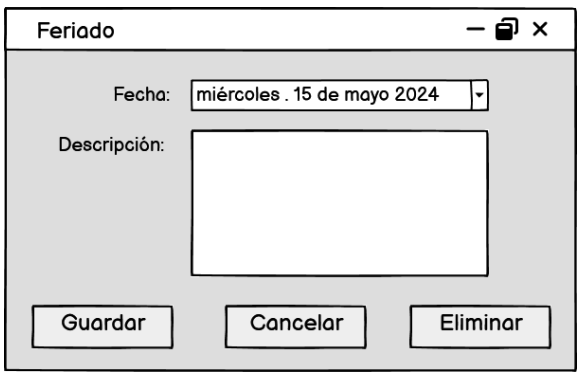


Figura 33. Diseño de la Pantalla de Feriado.

Fuente: Elaborado por autor

Liquidez Estructural

En la figura 34 se presenta la pantalla de Liquidez Estructural, el usuario podrá generar el reporte de la Liquidez Estructural de la fecha que el desee, se escoge una fecha y se presentan los valores de esa semana.

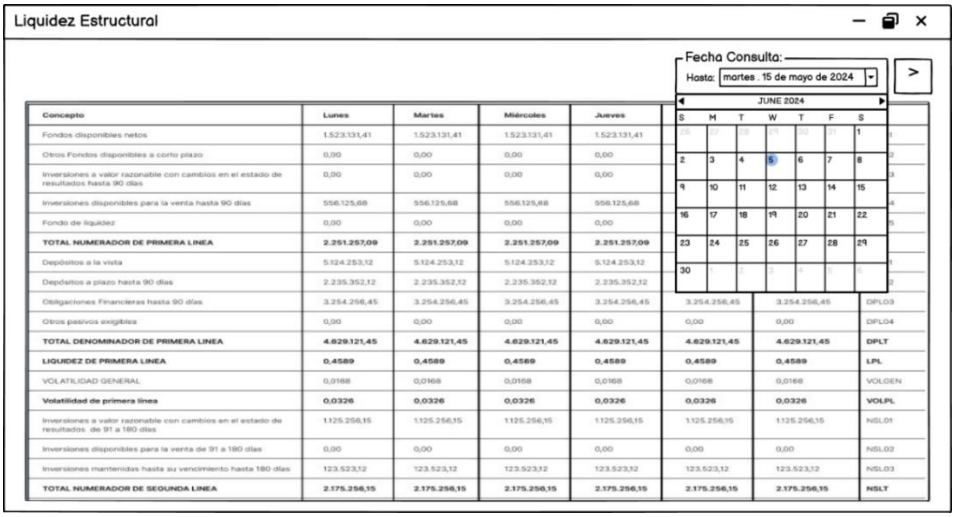


Figura 34. Diseño de Liquidez Estructual.

Fuente: Elaborado por autor

Volatilidad – Saldos

En la figura 35 se presenta la pantalla Volatilidad – Saldos, el usuario podrá generar el reporte de los Saldos de la Volatilidad al seleccionar una fecha.

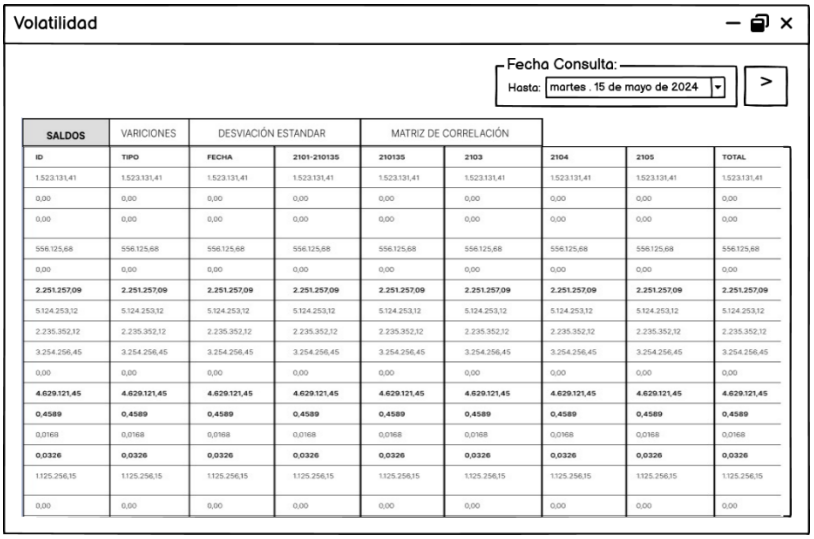


Figura 35. Diseño de Volatilidad - Saldos.

Fuente: Elaborado por autor

Volatilidad – Variaciones

En la figura 36 se presenta la pantalla Volatilidad – Variaciones, el usuario podrá generar el reporte de las Variaciones de la Volatilidad al seleccionar una fecha.

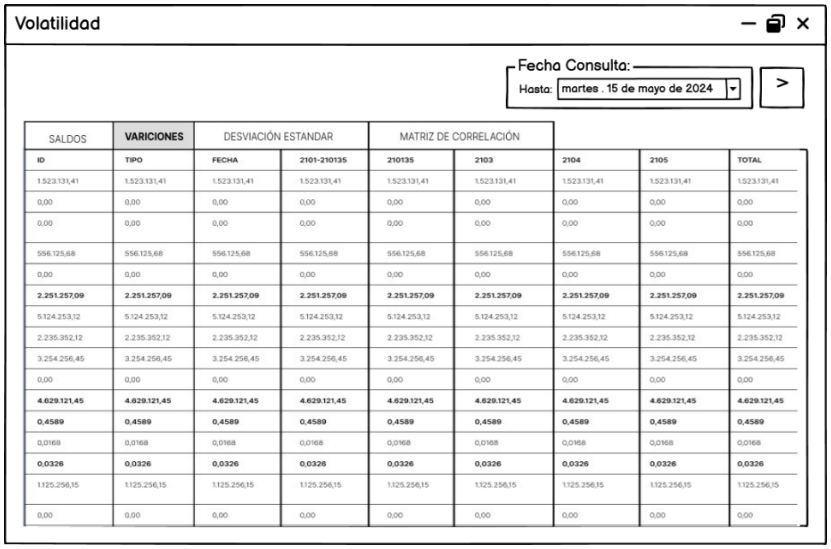


Figura 36. Diseño de Volatilidad - Variaciones.

Fuente: Elaborado por autor

Volatilidad – Desviación Estándar

En la figura 37 se presenta la pantalla Volatilidad – Desviación Estándar, el usuario podrá generar el reporte de la Desviación Estándar de la Volatilidad al seleccionar una fecha.

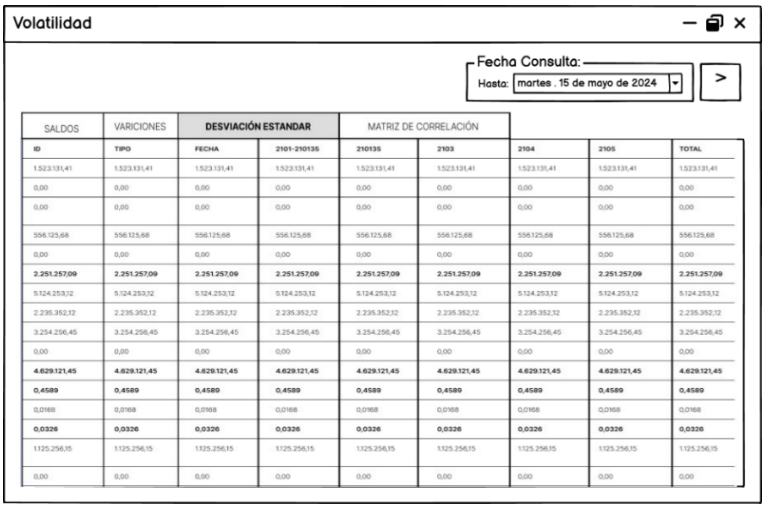


Figura 37. Diseño de Volatilidad – Deviación Estandar.

Fuente: Elaborado por autor

Volatilidad – Matriz de Correlación

En la figura 38 se presenta la pantalla Volatilidad – Matriz de Correlación, el usuario podrá generar el reporte de la Matriz de Correlación y realizar los cálculos necesarios de la Volatilidad al seleccionar una fecha.

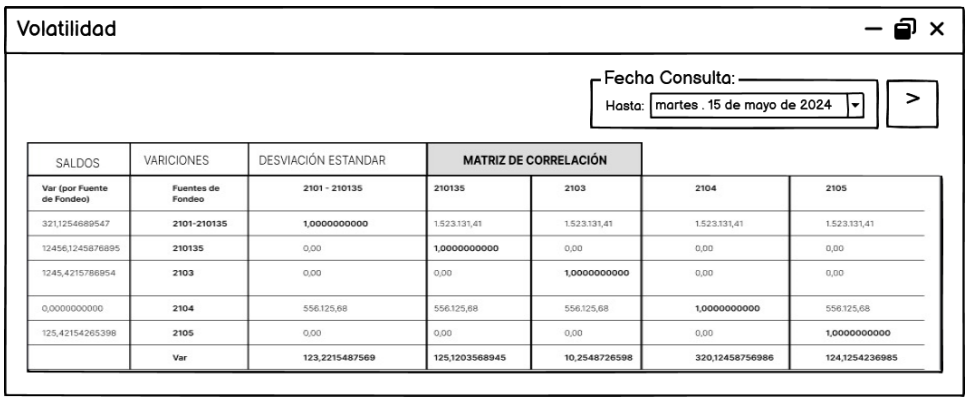


Figura 38. Diseño de Volatilidad – Matriz de Correlación.

Fuente: Elaborado por autor

Mayores Depositantes

En la figura 39 se presenta la pantalla de Mayores Depositantes, el usuario podrá generar el reporte de los Mayores Depositantes, el cual podrá filtrarse por la fecha.

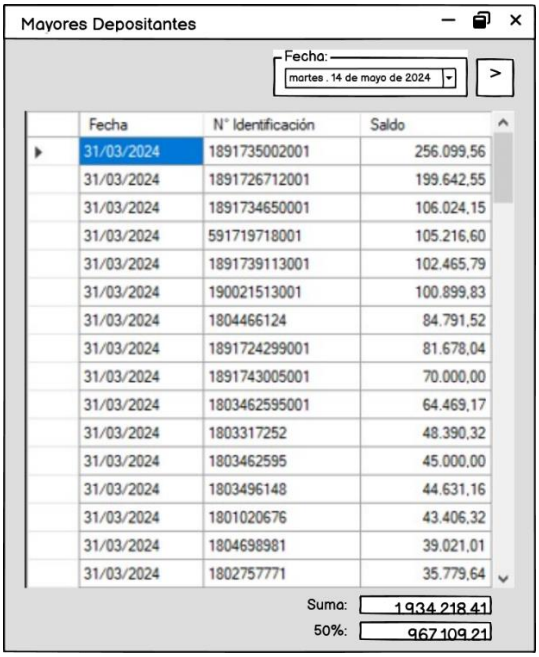


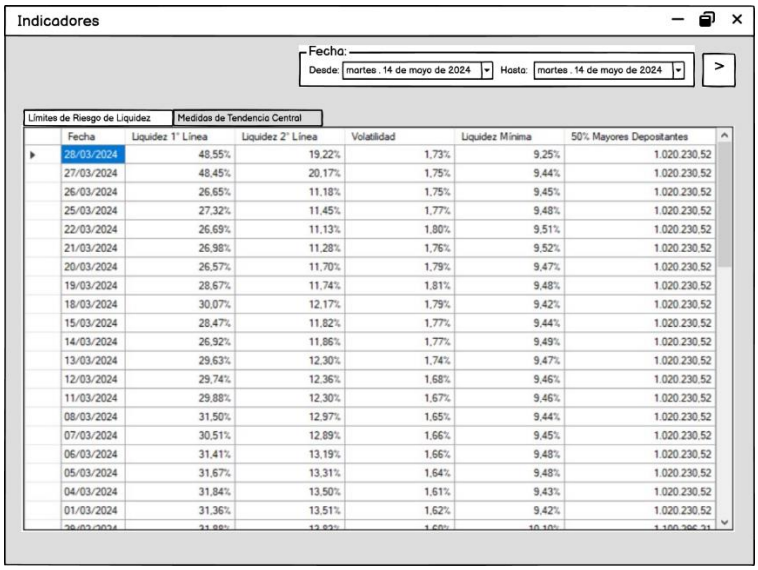
Figura 39. Diseño de la Pantalla de Mayores Depositantes

Fuente: Elaborado por autor

Indicadores

En las figuras 40 y 41 se presentan las pantallas de los Indicadores, los usuarios podrán generar el reporte de los límites de riesgo de liquidez y las medidas de tendencia central para generar los indicadores.

Indicadores – Limites de Riesgo de Liquidez

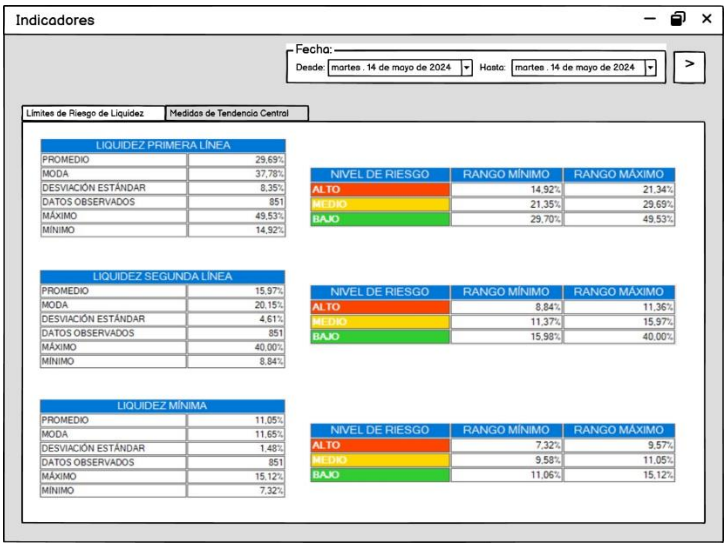


Fecha	Liquidez 1° Línea	Liquidez 2° Línea	Volatilidad	Liquidez Mínima	50% Mayores Depositantes
28/03/2024	48,55%	19,22%	1,73%	9,25%	1.020.230,52
27/03/2024	48,45%	20,17%	1,75%	9,44%	1.020.230,52
26/03/2024	26,65%	11,18%	1,75%	9,45%	1.020.230,52
25/03/2024	27,32%	11,45%	1,77%	9,48%	1.020.230,52
22/03/2024	26,69%	11,13%	1,80%	9,51%	1.020.230,52
21/03/2024	26,98%	11,28%	1,76%	9,52%	1.020.230,52
20/03/2024	26,57%	11,70%	1,79%	9,47%	1.020.230,52
19/03/2024	28,67%	11,74%	1,81%	9,48%	1.020.230,52
18/03/2024	30,07%	12,17%	1,79%	9,42%	1.020.230,52
15/03/2024	28,47%	11,82%	1,77%	9,44%	1.020.230,52
14/03/2024	26,92%	11,86%	1,77%	9,49%	1.020.230,52
13/03/2024	29,63%	12,30%	1,74%	9,47%	1.020.230,52
12/03/2024	29,74%	12,36%	1,68%	9,46%	1.020.230,52
11/03/2024	29,88%	12,30%	1,67%	9,46%	1.020.230,52
08/03/2024	31,50%	12,97%	1,65%	9,44%	1.020.230,52
07/03/2024	30,51%	12,89%	1,66%	9,45%	1.020.230,52
06/03/2024	31,41%	13,19%	1,66%	9,48%	1.020.230,52
05/03/2024	31,67%	13,31%	1,64%	9,48%	1.020.230,52
04/03/2024	31,84%	13,50%	1,61%	9,43%	1.020.230,52
01/03/2024	31,36%	13,51%	1,62%	9,42%	1.020.230,52

Figura 40. Diseño de la pantalla de Limites de Riesgo de Liquidez

Fuente: Elaborado por autor

Indicadores – Medidas de Tendencia Central



LIQUIDEZ PRIMERA LÍNEA	
PROMEDIO	29,69%
MODA	37,78%
DESVIACIÓN ESTÁNDAR	8,35%
DATOS OBSERVADOS	851
MÁXIMO	49,53%
MÍNIMO	14,92%

NIVEL DE RIESGO	RANGO MÍNIMO	RANGO MÁXIMO
ALTO	14,92%	21,34%
MEDIO	21,35%	29,69%
BAJO	29,70%	49,53%

LIQUIDEZ SEGUNDA LÍNEA	
PROMEDIO	15,97%
MODA	20,15%
DESVIACIÓN ESTÁNDAR	4,61%
DATOS OBSERVADOS	851
MÁXIMO	40,00%
MÍNIMO	8,84%

NIVEL DE RIESGO	RANGO MÍNIMO	RANGO MÁXIMO
ALTO	8,84%	11,36%
MEDIO	11,37%	15,97%
BAJO	15,98%	40,00%

LIQUIDEZ MÍNIMA	
PROMEDIO	11,05%
MODA	11,65%
DESVIACIÓN ESTÁNDAR	1,48%
DATOS OBSERVADOS	851
MÁXIMO	15,12%
MÍNIMO	7,32%

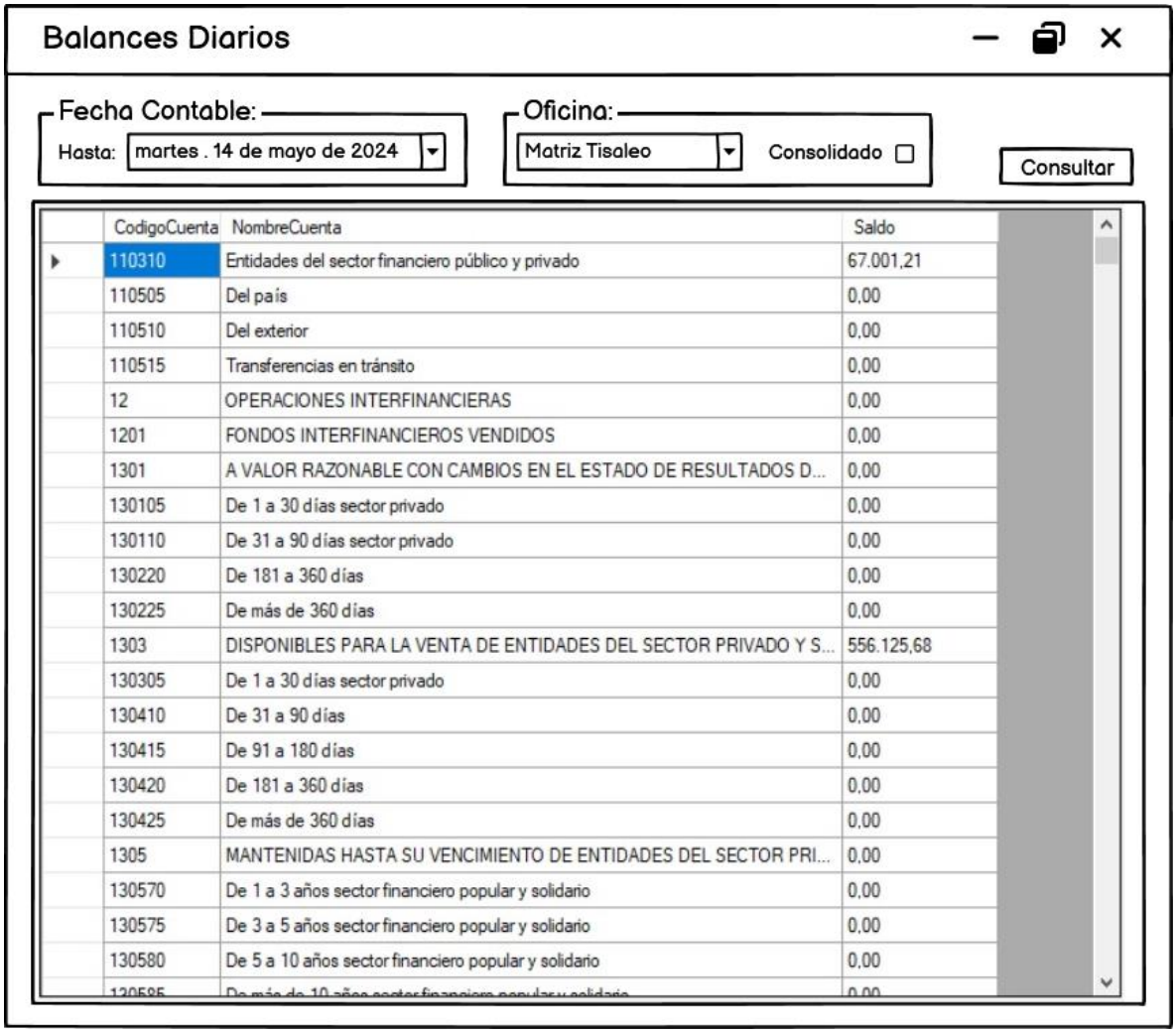
NIVEL DE RIESGO	RANGO MÍNIMO	RANGO MÁXIMO
ALTO	7,32%	9,57%
MEDIO	9,58%	11,05%
BAJO	11,06%	15,12%

Figura 41. Diseño de la Pantalla de Medidas de Tendencia Central

Fuente: Elaborado por autor

Balances Diarios

En la figura 42 se presenta la pantalla de Balances Diarios, el usuario podrá generar el reporte de los Balances Diarios, el cual podrá filtrarse por la fecha y/o por oficina.



	CodigoCuenta	NombreCuenta	Saldo
▶	110310	Entidades del sector financiero público y privado	67.001,21
	110505	Del país	0,00
	110510	Del exterior	0,00
	110515	Transferencias en tránsito	0,00
	12	OPERACIONES INTERFINANCIERAS	0,00
	1201	FONDOS INTERFINANCIEROS VENDIDOS	0,00
	1301	A VALOR RAZONABLE CON CAMBIOS EN EL ESTADO DE RESULTADOS D...	0,00
	130105	De 1 a 30 días sector privado	0,00
	130110	De 31 a 90 días sector privado	0,00
	130220	De 181 a 360 días	0,00
	130225	De más de 360 días	0,00
	1303	DISPONIBLES PARA LA VENTA DE ENTIDADES DEL SECTOR PRIVADO Y S...	556.125,68
	130305	De 1 a 30 días sector privado	0,00
	130410	De 31 a 90 días	0,00
	130415	De 91 a 180 días	0,00
	130420	De 181 a 360 días	0,00
	130425	De más de 360 días	0,00
	1305	MANTENIDAS HASTA SU VENCIMIENTO DE ENTIDADES DEL SECTOR PRI...	0,00
	130570	De 1 a 3 años sector financiero popular y solidario	0,00
	130575	De 3 a 5 años sector financiero popular y solidario	0,00
	130580	De 5 a 10 años sector financiero popular y solidario	0,00
	130585	De más de 10 años sector financiero popular y solidario	0,00

Figura 42. Diseño de Estado de Situación Final.

Fuente: Elaborado por autor

3.3 Implementación

3.3.1 Funcionamiento

En primera instancia, el usuario deberá ingresar al sistema por medio de la pantalla de inicio de sesión, ingresando sus credenciales. La interfaz se muestra en la figura 43.



Figura 43. Pantalla de Inicio de Sesión.

Fuente: Elaborado por autor

Una vez que se haya ingresado correctamente las credenciales, se mostrará la pantalla principal del sistema como en la figura 44, como administrador podrá realizar configuraciones y realizar los reportes de Liquidez Estructural.



Figura 44. Pantalla Principal del Sistema.

Fuente: Elaborado por autor

Dentro del sistema se podrá cambiar de contraseña ingresando a Sistema > Cambio Contraseña como se muestra en la figura 45, donde el usuario deberá ingresar todos los datos necesarios para realizar el cambio.

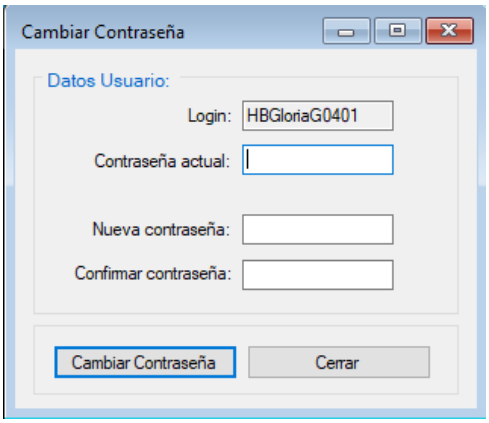


Figura 45. Pantalla de Cambio de Contraseña.

Fuente: Elaborado por autor

El usuario administrador podrá ingresar a Administración > Usuarios. En la figura 46 se presenta la pantalla de Usuarios en la que se podrá crear, buscar, modificar y resetear un usuario si así lo desea.

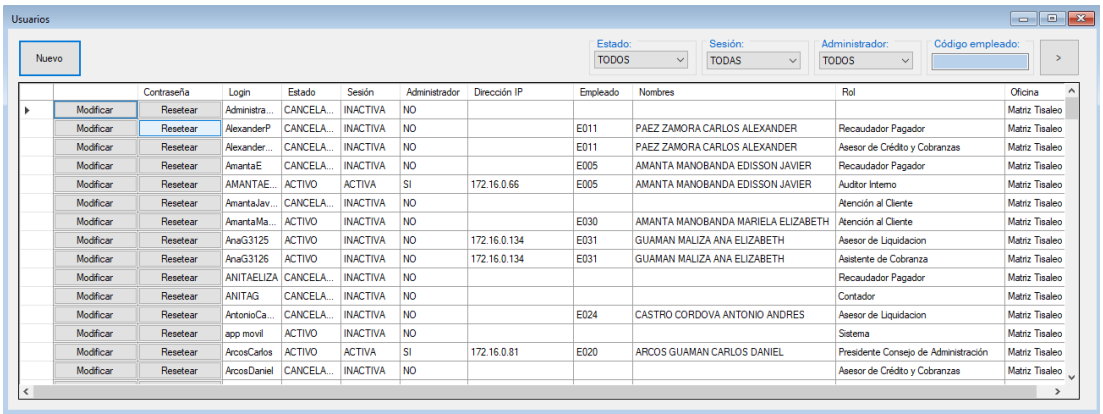


Figura 46. Pantalla de Usuarios.

Fuente: Elaborado por autor

Para crear un usuario en la pantalla de Usuarios deberá presionar el botón “Nuevo”, se mostrará la pantalla con el formulario para crea un usuario como en la figura 47.

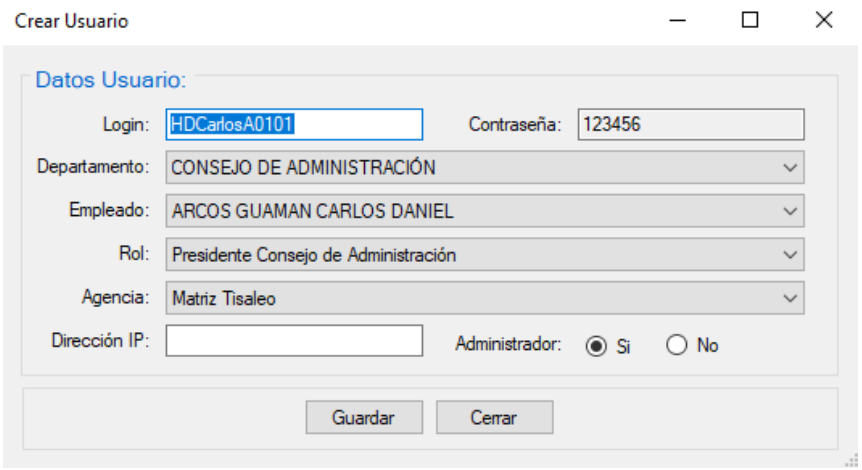


Figura 47. Pantalla de Crear Usuario.

Fuente: Elaborado por autor

Para buscar un usuario en la pantalla de Usuarios se podrá realizar ingresando en el código del empleado en el cuadro celeste, en caso de no saber el código del empleado, se puede presionar la tecla F2 para buscar por nombres o apellidos en la pantalla que se muestra en la figura 48.

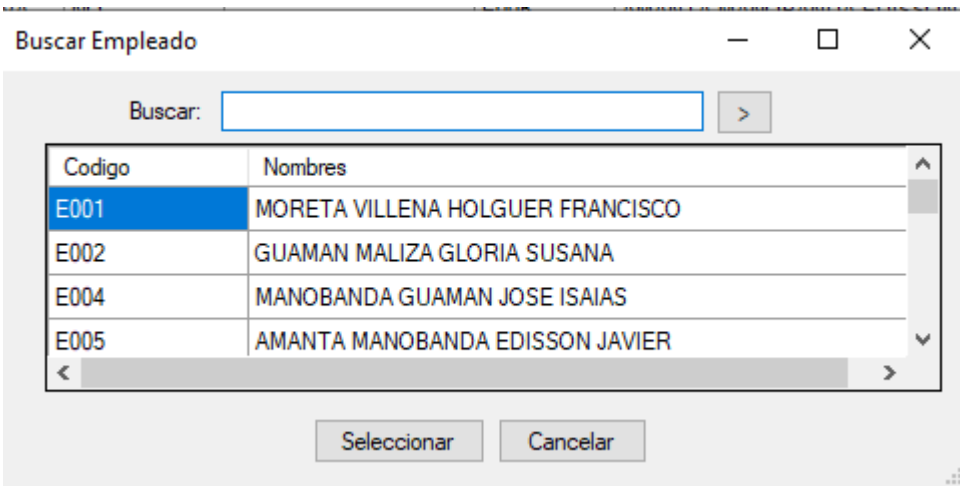


Figura 48. Pantalla de Buscar.

Fuente: Elaborado por autor

Para modificar un usuario en la pantalla de Usuarios se debe presionar el botón "Modificar" que consta para cada uno de los empleados existentes, se mostrara la pantalla "Modificar Usuario" como en la figura 49, donde los campos login, Agencia, Empleado y Perfil estarán boqueados, y los campos que se podrán editar serán: Dirección IP, Administrador, estado y Cerrar Sesión.

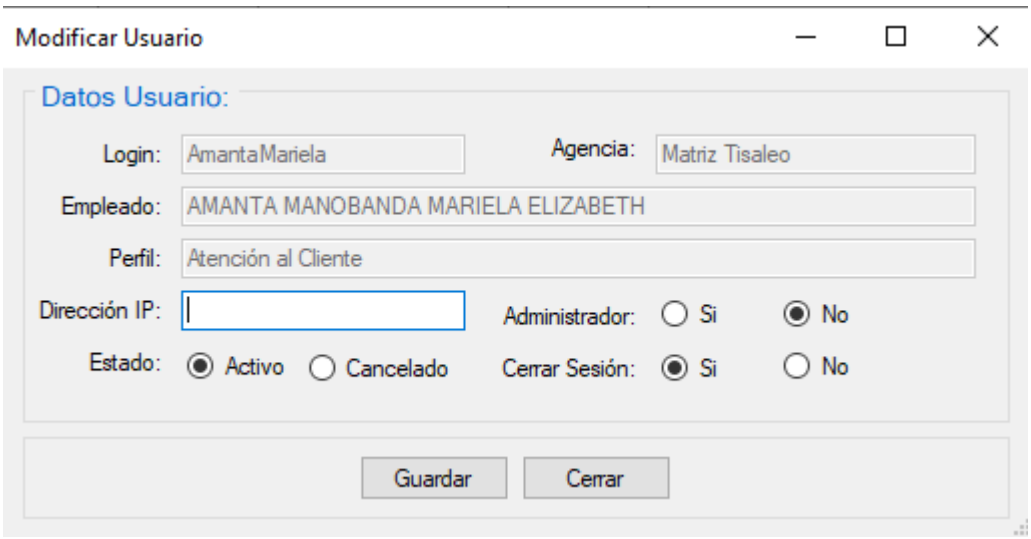


Figura 49. Pantalla de Modificar Usuario.

Fuente: Elaborado por autor

Para resetear un usuario en la pantalla de Usuarios se debe presionar el botón "Resetear" que conta para cada uno de los empleados existentes, se mostrará una pantalla de confirmación como en la figura 50 para saber si quiere o no resetear la contraseña y se pondrá una por defecto.

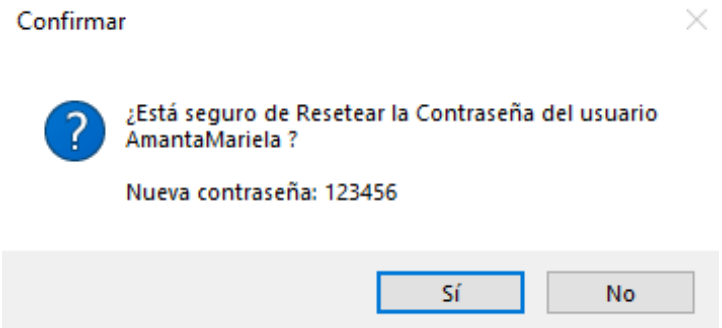


Figura 50. Pantalla de Confirmación.

Fuente: Elaborado por autor

Como usuario administrador podrá ingresar a Administración > Rol, en la pantalla Rol el usuario podrá crear, modificar y eliminar un rol como se muestra en la figura 51.

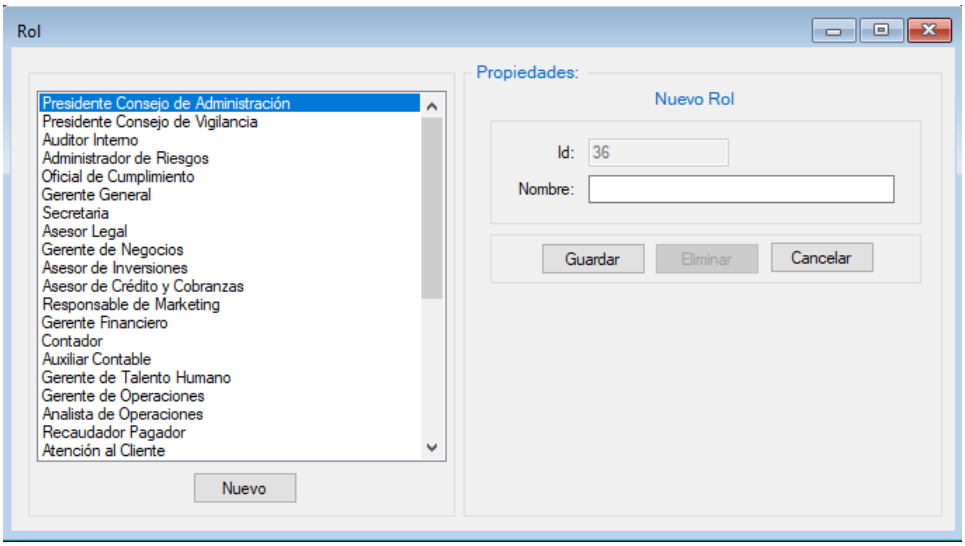


Figura 51. Pantalla de Rol.

Fuente: Elaborado por autor

También como parte de las configuraciones, el usuario podrá ingresar a Administración > Menú Sistema, en la pantalla Menú Sistema el administrador podrá crear, modificar y eliminar un menú o un ítem del menú como se muestra en la figura 52.

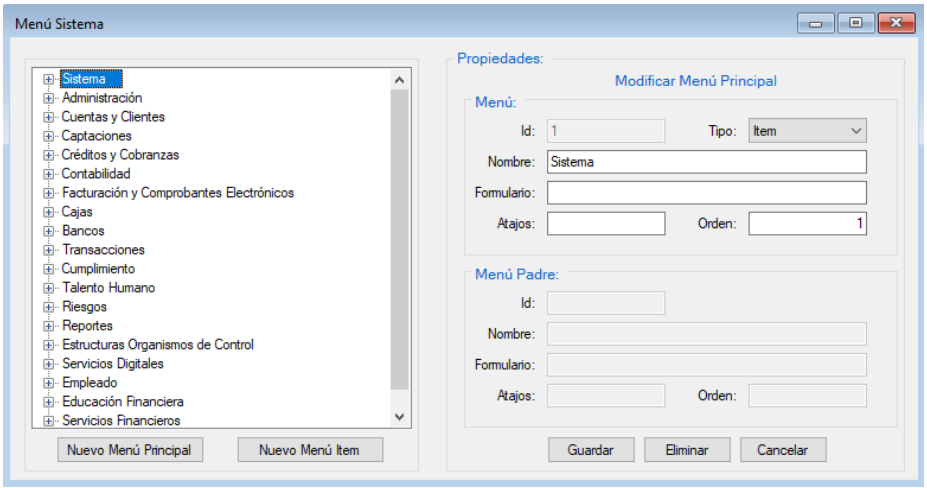


Figura 52. Pantalla de Menú Sistema.

Fuente: Elaborado por autor

Una vez creado el rol y el menú dentro del sistema se podrá ingresar a Administración > Menú Rol como se muestra en la figura 53, donde el usuario podrá relacionar el menú o los ítems del menú con el rol para crear un menú personalizado para cada uno de los role existentes.

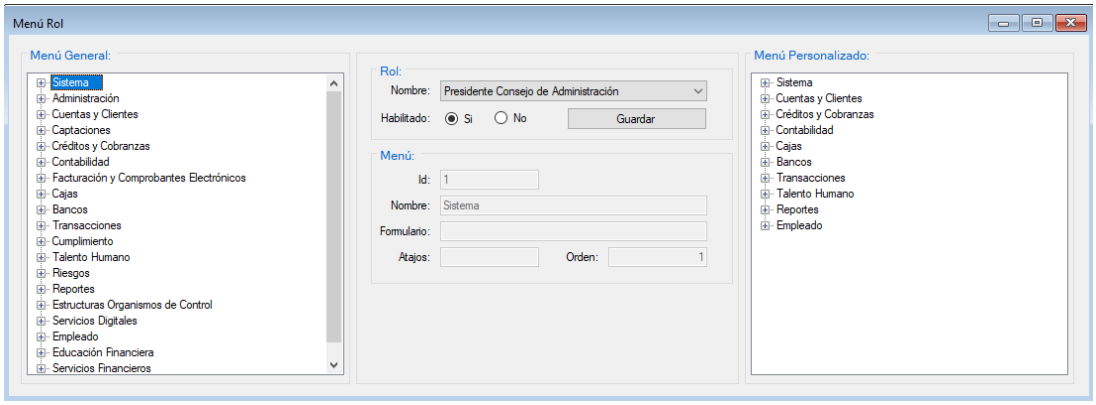


Figura 53. Pantalla de Menú Rol.

Fuente: Elaborado por autor

El usuario El usuario administrador también podrá gestionar los empleados ingresando a Talento Humano > Empleado como se muestra en la figura 54, en la pantalla de Empleado se podrá Crear o Modificar un empleado con todos los datos necesarios.

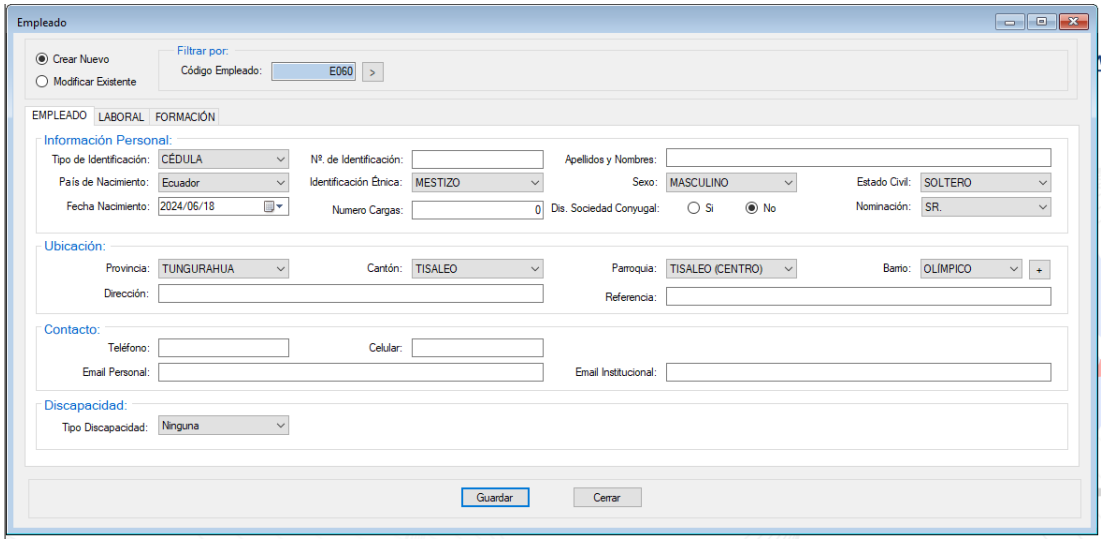
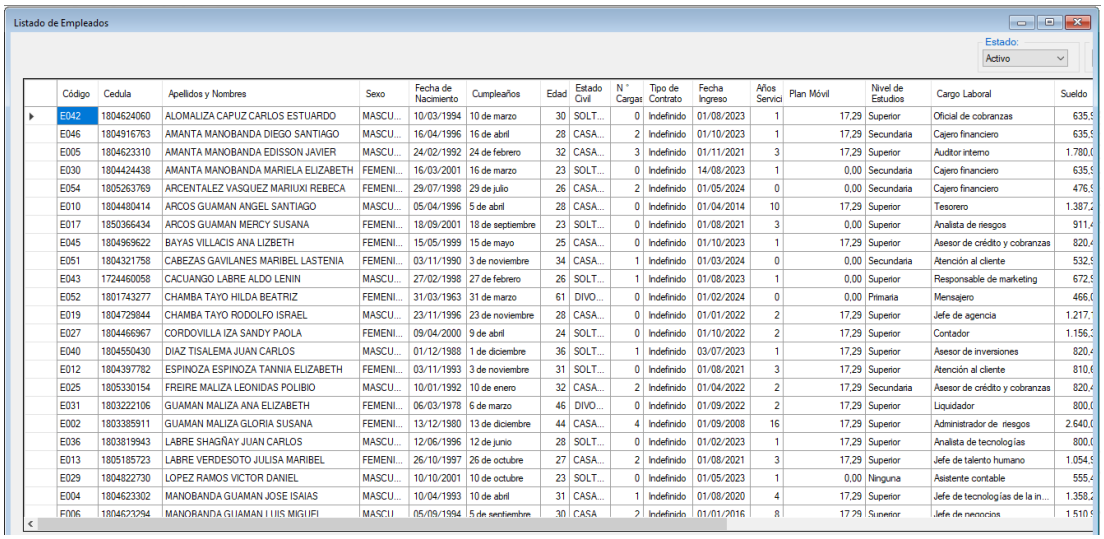


Figura 54. Pantalla de Empleado

Fuente: Elaborador por autor

El usuario administrador también podrá visualizar la nómina de los empleados ingresando a Talento Humano > Nomina como se muestra en la figura 55, en la pantalla de Listado de Empleados se podrá visualizar los datos de los Empleados.



Código	Cedula	Apellidos y Nombres	Sexo	Fecha de Nacimiento	Cumpleaños	Edad	Estado Civil	N ° Cargas	Tipo de Contrato	Fecha Ingreso	Años Servici	Plan Móvil	Nivel de Estudios	Cargo Laboral	Sueldo
E042	1804624060	ALOMALIZA CAPUZ CARLOS ESTUARDO	MASCU...	10/03/1994	10 de marzo	30	SOLT...	0	Indefinido	01/08/2023	1	17,29	Superior	Oficial de cobranzas	635.9
E046	1804916763	AMANTA MANOBANDA DIEGO SANTIAGO	MASCU...	16/04/1996	16 de abril	28	CASA...	2	Indefinido	01/10/2023	1	17,29	Secundaria	Cajero financiero	635.9
E005	1804623310	AMANTA MANOBANDA EDISSON JAVIER	MASCU...	24/02/1992	24 de febrero	32	CASA...	3	Indefinido	01/11/2021	3	17,29	Superior	Auditor interno	1.780.(
E030	1804424438	AMANTA MANOBANDA MARIELA ELIZABETH	FEMENI...	16/03/2001	16 de marzo	23	SOLT...	0	Indefinido	14/08/2023	1	0,00	Secundaria	Cajero financiero	635.9
E054	1805263769	ARCENTALEZ VASQUEZ MARIUXI REBECA	FEMENI...	29/07/1998	29 de julio	26	CASA...	2	Indefinido	01/05/2024	0	0,00	Secundaria	Cajero financiero	476.9
E010	1804480414	ARCOS GUAMAN ANGEL SANTIAGO	MASCU...	05/04/1996	5 de abril	28	CASA...	0	Indefinido	01/04/2014	10	17,29	Superior	Tesorero	1.387.2
E017	1850366434	ARCOS GUAMAN MERCY SUSANA	FEMENI...	18/09/2001	18 de septiembre	23	SOLT...	0	Indefinido	01/08/2021	3	0,00	Superior	Analista de riesgos	911.4
E045	1804969622	BAYAS VILLACIS ANA LIZBETH	FEMENI...	15/05/1999	15 de mayo	25	CASA...	0	Indefinido	01/10/2023	1	17,29	Superior	Asesor de crédito y cobranzas	820.4
E051	1804321758	CABEZAS GAVILANES MARIBEL LASTENIA	FEMENI...	03/11/1990	3 de noviembre	34	CASA...	1	Indefinido	01/03/2024	0	0,00	Secundaria	Atención al cliente	532.9
E043	1724460058	CACUANGO LABRE ALDO LENIN	MASCU...	27/02/1998	27 de febrero	26	SOLT...	1	Indefinido	01/08/2023	1	0,00	Superior	Responsable de marketing	672.9
E052	1801743277	CHAMBA TAYO HILDA BEATRIZ	FEMENI...	31/03/1963	31 de marzo	61	DIVO...	0	Indefinido	01/02/2024	0	0,00	Primaria	Mensajero	466.0
E019	1804729844	CHAMBA TAYO RODOLFO ISRAEL	MASCU...	23/11/1996	23 de noviembre	28	CASA...	0	Indefinido	01/01/2022	2	17,29	Superior	Jefe de agencia	1.217.7
E027	1804466967	CORDOVILLA IZA SANDY PAOLA	FEMENI...	09/04/2000	9 de abril	24	SOLT...	0	Indefinido	01/10/2022	2	17,29	Superior	Contador	1.156.2
E040	1804550430	DIAZ TISALEMA JUAN CARLOS	MASCU...	01/12/1988	1 de diciembre	36	SOLT...	1	Indefinido	03/07/2023	1	17,29	Superior	Asesor de inversiones	820.4
E012	1804397782	ESPINOZA ESPINOZA TANNIA ELIZABETH	FEMENI...	03/11/1993	3 de noviembre	31	SOLT...	0	Indefinido	01/08/2021	3	17,29	Superior	Atención al cliente	810.0
E025	1805330154	FREIRE MALIZA LEONIDAS POLIBIO	MASCU...	10/01/1992	10 de enero	32	CASA...	2	Indefinido	01/04/2022	2	17,29	Secundaria	Asesor de crédito y cobranzas	820.4
E031	1803222106	GUAMAN MALIZA ANA ELIZABETH	FEMENI...	06/03/1978	6 de marzo	46	DIVO...	0	Indefinido	01/09/2022	2	17,29	Superior	Liquidador	800.0
E002	1803385911	GUAMAN MALIZA GLORIA SUSANA	FEMENI...	13/12/1980	13 de diciembre	44	CASA...	4	Indefinido	01/09/2008	16	17,29	Superior	Administrador de riesgos	2.640.0
E036	1803819943	LABRE SHAGÑAY JUAN CARLOS	MASCU...	12/06/1996	12 de junio	28	SOLT...	0	Indefinido	01/02/2023	1	17,29	Superior	Analista de tecnologías	800.0
E013	1805185723	LABRE VERDESOTO JULISA MARIBEL	FEMENI...	26/10/1997	26 de octubre	27	CASA...	2	Indefinido	01/08/2021	3	17,29	Superior	Jefe de talento humano	1.054.9
E029	1804822730	LOPEZ RAMOS VICTOR DANIEL	MASCU...	10/10/2001	10 de octubre	23	SOLT...	0	Indefinido	01/05/2023	1	0,00	Ninguna	Asistente contable	555.4
E004	1804623302	MANOBANDA GUAMAN JOSE ISAIAS	MASCU...	10/04/1993	10 de abril	31	CASA...	1	Indefinido	01/08/2020	4	17,29	Superior	Jefe de tecnologías de la in...	1.358.2
E006	1804623294	MANOBANDA GUAMAN LUIS MIGUEL	MASCU...	05/09/1994	5 de septiembre	30	CASA...	2	Indefinido	01/01/2016	8	17,29	Superior	Jefe de negocios	1.510.9

Figura 55. Pantalla de Nomina

Fuente: Elaborador por autor

El usuario administrador también podrá gestionar los feriados ingresando a Talento Humano > Feriados como se muestra en la figura 56, en la pantalla de Feriados se podrá visualizar cuantos feriados habido en el transcurso del año, adema se podrá ingresar un nuevo feriado.

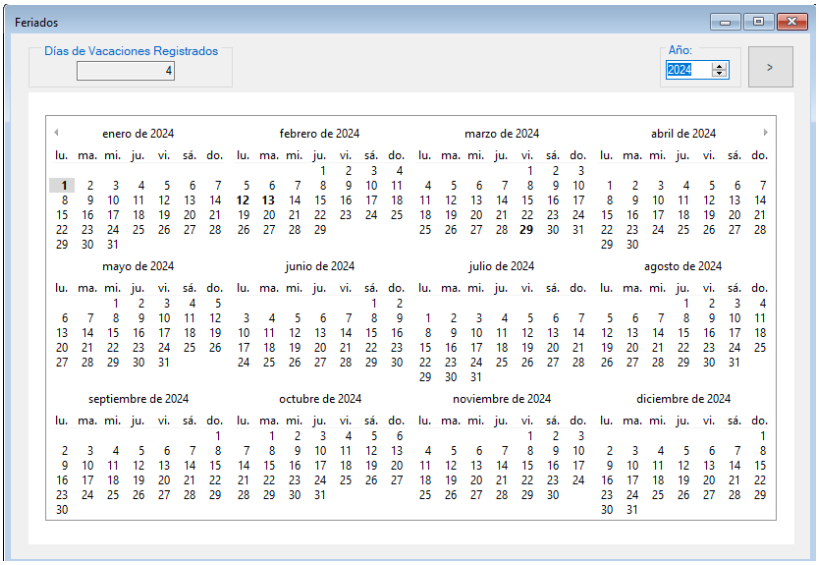


Figura 56. Pantalla de feriados.

Fuente: Elaborado por autor

Al seleccionar un día dentro de la pantalla Feriados, se mostrará una pantalla para crear, modificar o eliminar un feriado como se muestra en la figura 57, además, se podrá ingresar una descripción para ese día.

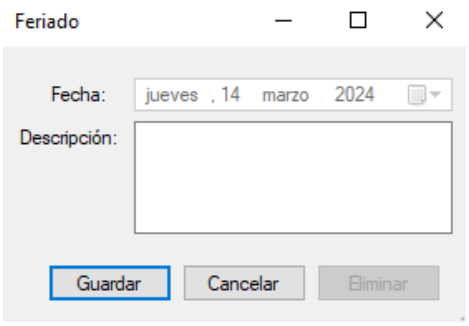
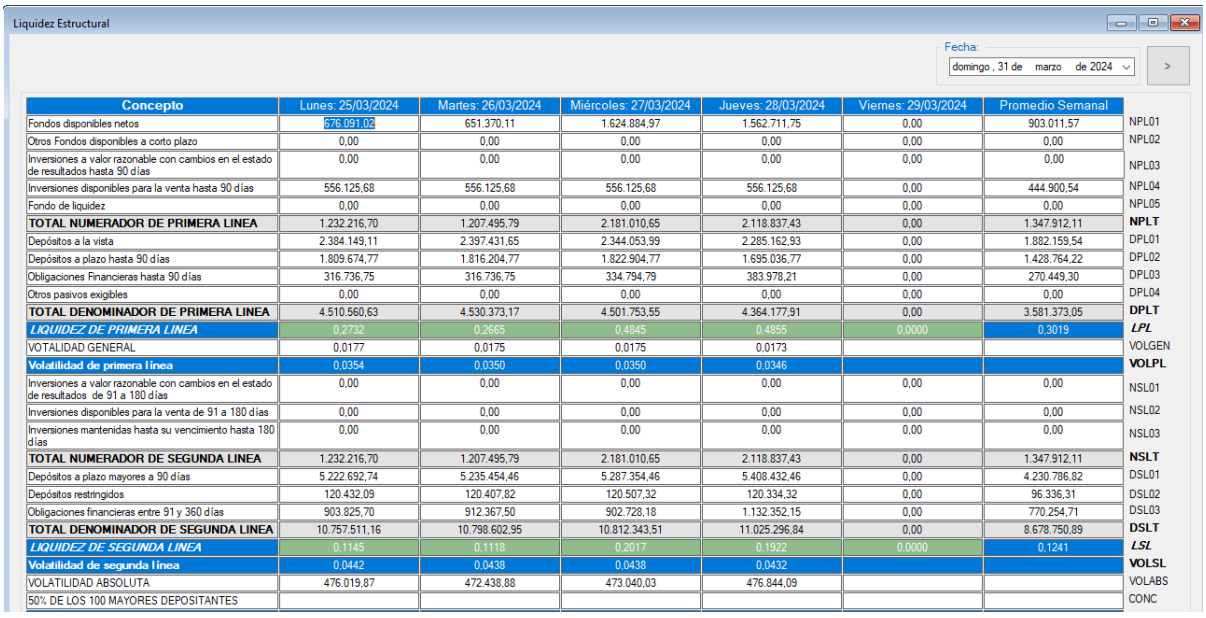


Figura 57. Pantalla de Feriado.

Fuente: Elaborado por autor

Como parte principal se tienen los reportes para que el usuario pueda evaluar la Liquidez estructural, al ingresar a Riesgos > Liquidez Estructural como se muestra en la figura 58, el usuario podrá generar el reporte de Liquidez Estructural, con una fecha seleccionada.



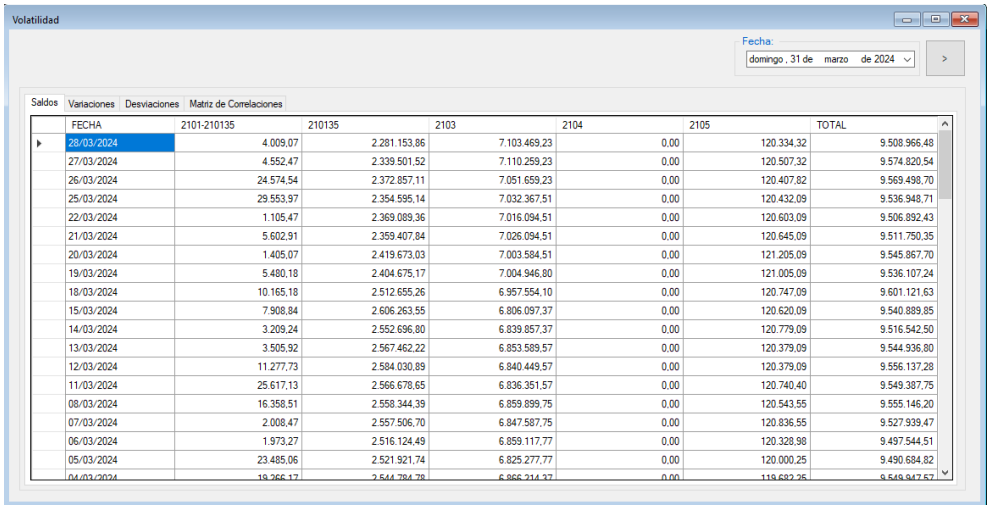
Concepto	Lunes: 25/03/2024	Martes: 26/03/2024	Miércoles: 27/03/2024	Jueves: 28/03/2024	Viernes: 29/03/2024	Promedio Semanal	
Fondos disponibles netos	676.091,02	651.370,11	1.624.884,97	1.562.711,75	0,00	903.011,57	NPL01
Otros Fondos disponibles a corto plazo	0,00	0,00	0,00	0,00	0,00	0,00	NPL02
Inversiones a valor razonable con cambios en el estado de resultados hasta 90 días	0,00	0,00	0,00	0,00	0,00	0,00	NPL03
Inversiones disponibles para la venta hasta 90 días	556.125,68	556.125,68	556.125,68	556.125,68	0,00	444.900,54	NPL04
Fondo de liquidez	0,00	0,00	0,00	0,00	0,00	0,00	NPL05
TOTAL NUMERADOR DE PRIMERA LINEA	1.232.216,70	1.207.495,79	2.181.010,65	2.118.837,43	0,00	1.347.912,11	**NPLT**
Depósitos a la vista	2.384.149,11	2.397.431,65	2.344.053,99	2.285.162,93	0,00	1.882.159,54	DPL01
Depósitos a plazo hasta 90 días	1.809.674,77	1.816.204,77	1.822.904,77	1.695.036,77	0,00	1.428.764,22	DPL02
Obligaciones Financieras hasta 90 días	316.736,75	316.736,75	334.794,79	383.978,21	0,00	270.449,30	DPL03
Otros pasivos exigibles	0,00	0,00	0,00	0,00	0,00	0,00	DPL04
TOTAL DENOMINADOR DE PRIMERA LINEA	4.510.560,63	4.530.373,17	4.501.753,55	4.364.177,91	0,00	3.581.373,05	**DPLT**
LIQUIDEZ DE PRIMERA LINEA	0,2732	0,2665	0,4845	0,4855	0,0000	0,3019	***LPL***
VOTALIDAD GENERAL	0,0177	0,0175	0,0175	0,0173			VOLGEN
Volatilidad de primera linea	0,0354	0,0350	0,0350	0,0346			**VOLPL**
Inversiones a valor razonable con cambios en el estado de resultados de 91 a 180 días	0,00	0,00	0,00	0,00	0,00	0,00	NSL01
Inversiones disponibles para la venta de 91 a 180 días	0,00	0,00	0,00	0,00	0,00	0,00	NSL02
Inversiones mantenidas hasta su vencimiento hasta 180 días	0,00	0,00	0,00	0,00	0,00	0,00	NSL03
TOTAL NUMERADOR DE SEGUNDA LINEA	1.232.216,70	1.207.495,79	2.181.010,65	2.118.837,43	0,00	1.347.912,11	**NSLT**
Depósitos a plazo mayores a 90 días	5.222.692,74	5.235.454,46	5.287.354,46	5.408.432,46	0,00	4.230.786,82	DSL01
Depósitos restringidos	120.432,09	120.407,82	120.507,32	120.334,32	0,00	96.336,31	DSL02
Obligaciones financieras entre 91 y 360 días	903.825,70	912.367,50	902.728,18	1.132.352,15	0,00	770.254,71	DSL03
TOTAL DENOMINADOR DE SEGUNDA LINEA	10.757.511,16	10.798.602,95	10.812.343,51	11.025.296,84	0,00	8.678.750,89	**DSLT**
LIQUIDEZ DE SEGUNDA LINEA	0,1145	0,1118	0,2017	0,1922	0,0000	0,1241	***LSL***
Volatilidad de segunda linea	0,0442	0,0438	0,0438	0,0432			**VOLSL**
VOLATILIDAD ABSOLUTA	476.019,87	472.438,88	473.040,03	476.844,09			VOLABS
50% DE LOS 100 MAYORES DEPOSITANTES							CONC

Figura 58. Pantalla de Reporte de Liquidez Estructural.

Fuente: Elaborado por autor

Al ingresar a Riesgos > Volatilidad se podrá generar los 4 reportes que forman parte de la Volatilidad, como se muestran en la figura 59, figura 60, figura 61 y figura 62.

Saldos

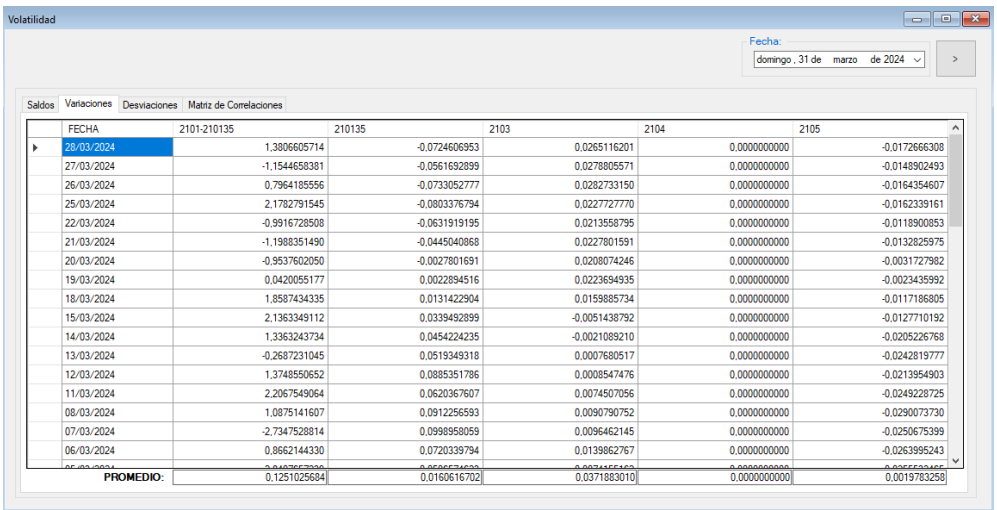


FECHA	2101-210135	210135	2103	2104	2105	TOTAL
28/03/2024	4.009,07	2.281.153,86	7.103.469,23	0,00	120.334,32	9.508.966,48
27/03/2024	4.552,47	2.339.501,52	7.110.259,23	0,00	120.507,32	9.574.820,54
26/03/2024	24.574,54	2.372.857,11	7.051.659,23	0,00	120.407,82	9.569.498,70
25/03/2024	29.553,97	2.354.595,14	7.032.367,51	0,00	120.432,09	9.536.948,71
22/03/2024	1.105,47	2.369.089,36	7.016.094,51	0,00	120.603,09	9.506.892,43
21/03/2024	5.602,91	2.359.407,84	7.026.094,51	0,00	120.645,09	9.511.750,35
20/03/2024	1.405,07	2.419.673,03	7.003.584,51	0,00	121.205,09	9.545.867,70
19/03/2024	5.480,18	2.404.675,17	7.004.946,80	0,00	121.005,09	9.536.107,24
18/03/2024	10.165,18	2.512.655,26	6.957.554,10	0,00	120.747,09	9.601.121,63
15/03/2024	7.908,84	2.606.263,55	6.806.097,37	0,00	120.620,09	9.540.889,85
14/03/2024	3.209,24	2.552.696,80	6.839.857,37	0,00	120.779,09	9.516.542,50
13/03/2024	3.505,92	2.567.462,22	6.853.589,57	0,00	120.379,09	9.544.936,80
12/03/2024	11.277,73	2.584.030,89	6.840.449,57	0,00	120.379,09	9.556.137,28
11/03/2024	25.617,13	2.566.678,65	6.836.351,57	0,00	120.740,40	9.549.387,75
08/03/2024	16.358,51	2.558.344,39	6.859.899,75	0,00	120.543,55	9.555.146,20
07/03/2024	2.008,47	2.557.506,70	6.847.587,75	0,00	120.836,55	9.527.939,47
06/03/2024	1.973,27	2.516.124,49	6.859.117,77	0,00	120.328,98	9.497.544,51
05/03/2024	23.485,06	2.521.921,74	6.825.277,77	0,00	120.000,25	9.490.684,82
04/03/2024	19.266,17	2.544.784,78	6.866.214,37	0,00	119.682,25	9.549.947,57

Figura 59. Pantalla de reportes de Volatilidad – Saldos.

Fuente: Elaborado por autor

Variaciones

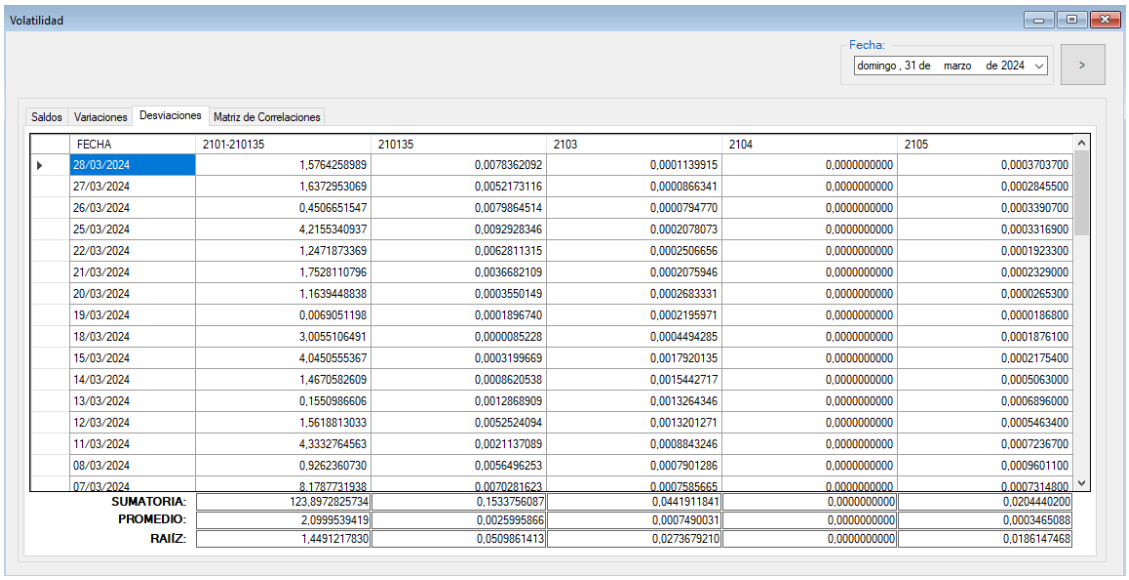


FECHA	2101-210135	210135	2103	2104	2105
28/03/2024	1,3806605714	-0,0724606953	0,0265116201	0,0000000000	-0,0172666308
27/03/2024	-1,1544658381	-0,0561692899	0,0278805571	0,0000000000	-0,0148902493
26/03/2024	0,7964185556	-0,0733052777	0,0282733150	0,0000000000	-0,0164354607
25/03/2024	2,1782791545	-0,0803376794	0,0227727770	0,0000000000	-0,0162339161
22/03/2024	-0,9916728508	-0,0631919195	0,0213558795	0,0000000000	-0,0118900853
21/03/2024	-1,1988351490	-0,0445040868	0,0227801591	0,0000000000	-0,0132825975
20/03/2024	-0,9537602050	-0,0027801691	0,0208074246	0,0000000000	-0,0031727982
19/03/2024	0,0420055177	0,0022894516	0,0223694935	0,0000000000	-0,0023435992
18/03/2024	1,8587434335	0,0131422904	0,0159885734	0,0000000000	-0,0117186805
15/03/2024	2,1363349112	0,0339492899	-0,0051438792	0,0000000000	-0,0127710192
14/03/2024	1,3363243734	0,0454224235	-0,0021089210	0,0000000000	-0,0205226768
13/03/2024	-0,2687231045	0,0519349318	0,0007680517	0,0000000000	-0,0242819777
12/03/2024	1,3748550652	0,0885351786	0,0008547476	0,0000000000	-0,0213954903
11/03/2024	2,2067549064	0,0620367607	0,0074507056	0,0000000000	-0,0249228725
08/03/2024	1,0875141607	0,0912256593	0,0090790752	0,0000000000	-0,0290073730
07/03/2024	-2,7347528814	0,0998958059	0,0096462145	0,0000000000	-0,0250675399
06/03/2024	0,8662144330	0,0720339794	0,0139862767	0,0000000000	-0,0263995243
PROMEDIO:	0,1251025684	0,0160616702	0,0371883010	0,0000000000	0,0019783258

Figura 60. Pantalla de reportes de Volatilidad – Variaciones.

Fuente: Elaborado por autor

Desviaciones

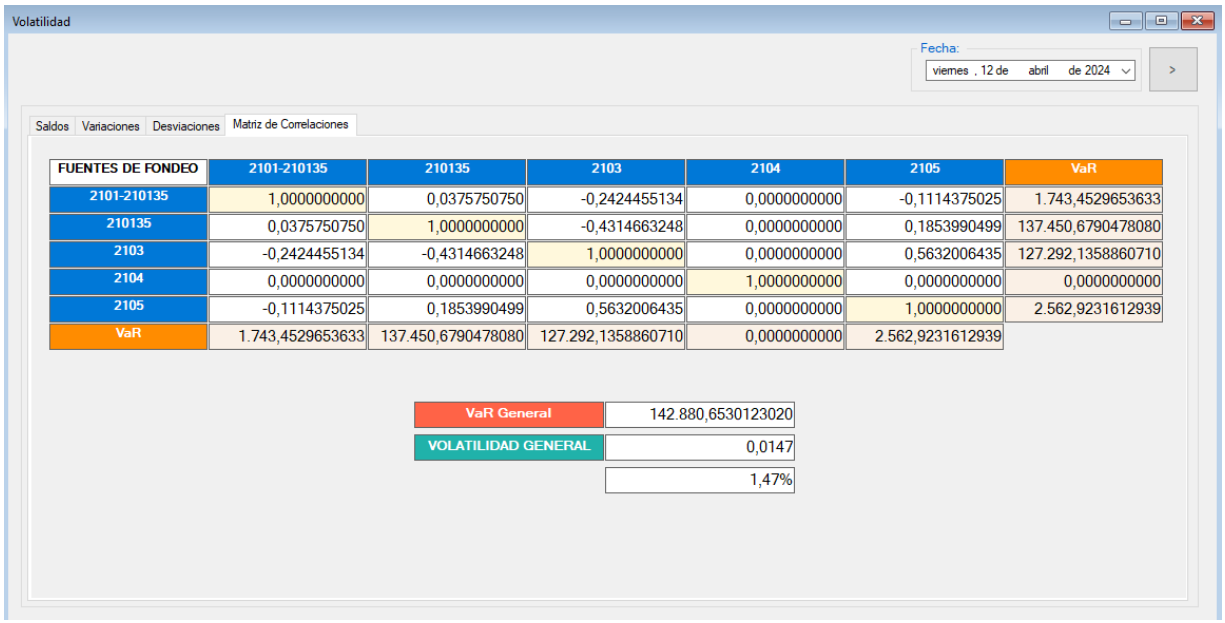


FECHA	2101-210135	210135	2103	2104	2105
28/03/2024	1,5764258989	0,0078362092	0,0001139915	0,0000000000	0,0003703700
27/03/2024	1,6372953069	0,0052173116	0,0000866341	0,0000000000	0,0002845500
26/03/2024	0,4506651547	0,0079864514	0,0000794770	0,0000000000	0,0003390700
25/03/2024	4,2155340937	0,0092928346	0,0002078073	0,0000000000	0,0003316900
22/03/2024	1,2471873369	0,0062811315	0,0002506656	0,0000000000	0,0001923300
21/03/2024	1,7528110796	0,0036682109	0,0002075946	0,0000000000	0,0002329000
20/03/2024	1,1639448838	0,0003550149	0,0002683331	0,0000000000	0,0000265300
19/03/2024	0,0069051198	0,0001896740	0,0002195971	0,0000000000	0,0000186800
18/03/2024	3,0055106491	0,0000085228	0,0004494285	0,0000000000	0,0001876100
15/03/2024	4,0450555367	0,0003199669	0,0017920135	0,0000000000	0,0002175400
14/03/2024	1,4670582609	0,0008620538	0,0015442717	0,0000000000	0,0005063000
13/03/2024	0,1550986606	0,0012868909	0,0013264346	0,0000000000	0,0006896000
12/03/2024	1,5618813033	0,0052524094	0,0013201271	0,0000000000	0,0005463400
11/03/2024	4,3332764563	0,0021137089	0,0008843246	0,0000000000	0,0007236700
08/03/2024	0,9262360730	0,0056496253	0,0007901286	0,0000000000	0,0009601100
07/03/2024	8,1787731938	0,0070281623	0,0007585665	0,0000000000	0,0007314800
SUMATORIA:	123,8972825734	0,1533756087	0,0441911841	0,0000000000	0,0204440200
PROMEDIO:	2,0999539419	0,0025995866	0,0007490031	0,0000000000	0,0003465088
RAÍZ:	1,4491217830	0,0509861413	0,0273679210	0,0000000000	0,0186147468

Figura 61. Pantalla de reportes de Volatilidad – Desviaciones.

Fuente: Elaborado por autor

Matriz de Correlaciones



FUENTES DE FONDEO	2101-210135	210135	2103	2104	2105	VaR
2101-210135	1,0000000000	0,0375750750	-0,2424455134	0,0000000000	-0,1114375025	1.743,4529653633
210135	0,0375750750	1,0000000000	-0,4314663248	0,0000000000	0,1853990499	137.450,6790478080
2103	-0,2424455134	-0,4314663248	1,0000000000	0,0000000000	0,5632006435	127.292,1358860710
2104	0,0000000000	0,0000000000	0,0000000000	1,0000000000	0,0000000000	0,0000000000
2105	-0,1114375025	0,1853990499	0,5632006435	0,0000000000	1,0000000000	2.562,9231612939
VaR	1.743,4529653633	137.450,6790478080	127.292,1358860710	0,0000000000	2.562,9231612939	

VaR General	142.880,6530123020
VOLATILIDAD GENERAL	0,0147
	1,47%

Figura 62. Pantalla de Volatilidad – Matriz de Correlaciones.

Fuente: Elaborado por autor

Se tiene el reporte de los Mayores Depositantes al ingresar a Riesgos > Mayores Depositantes como se muestra en la Figura 63, el usuario podrá generar el reporte según la fecha que necesite para los reportes de Mayores Depositantes.

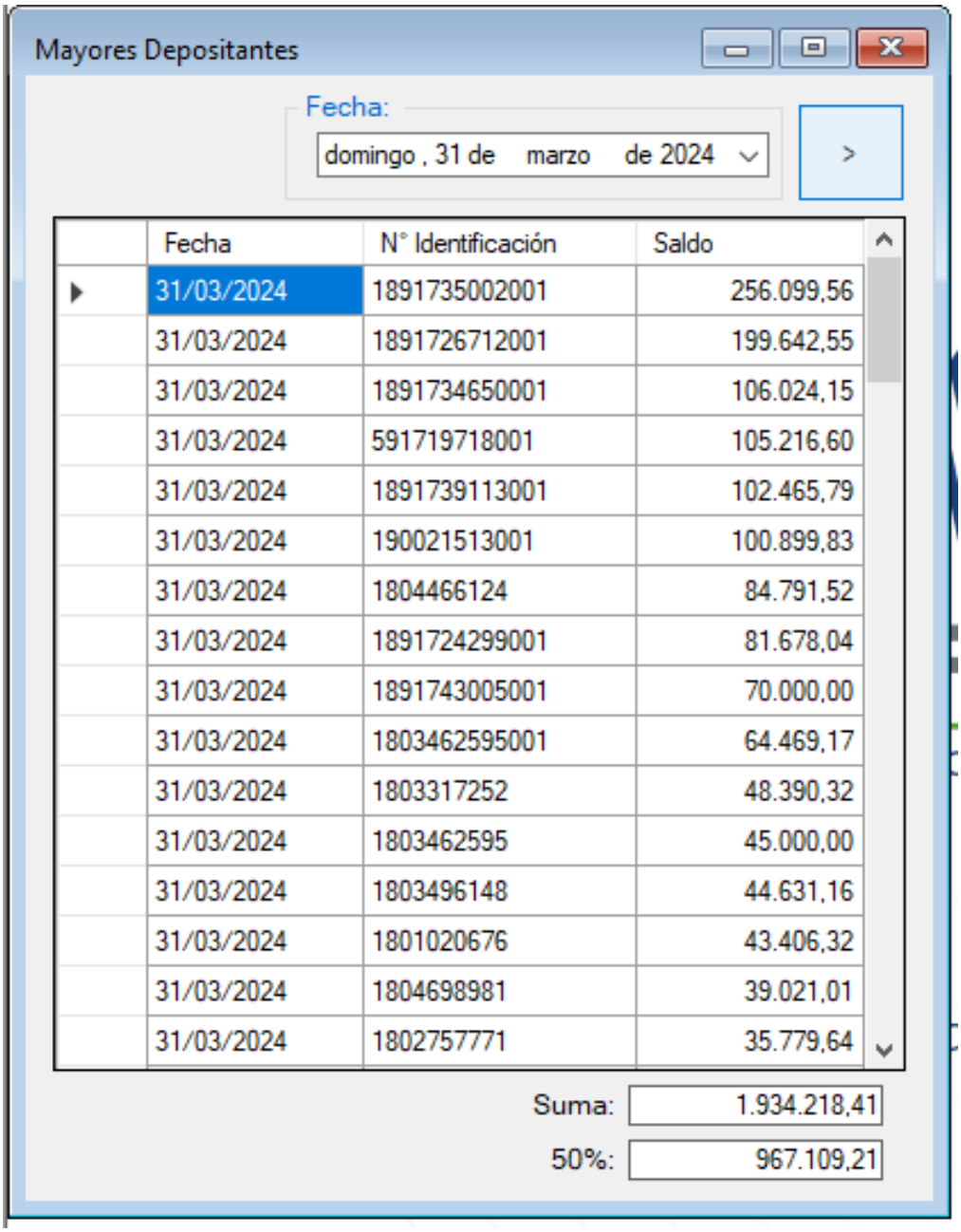
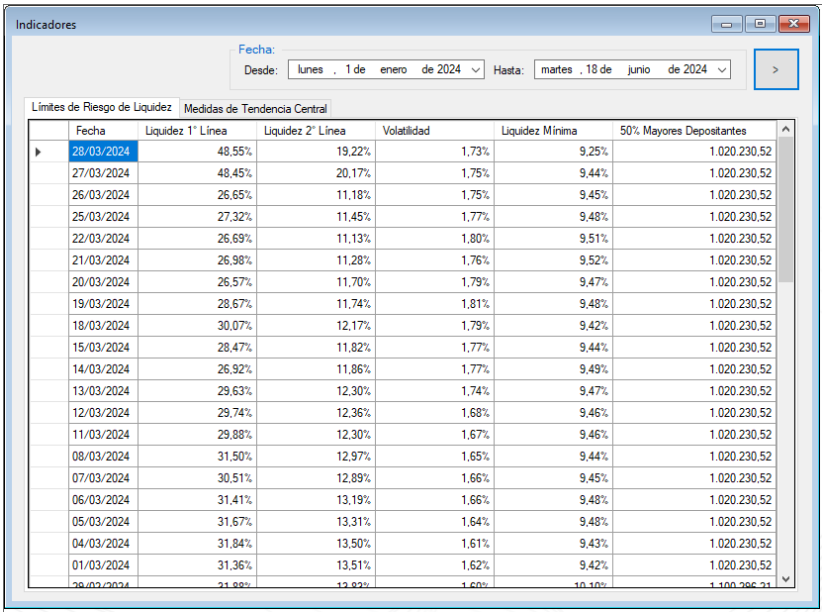


Figura 63. Pantalla de Mayores Depositantes

Fuente: Elaborado por autor

También se tiene el reporte de los Indicadores al ingresar a Riesgos > Indicadores como se muestra en la figura 64, el usuario podrá generar el reporte según las fechas que necesite para los Indicadores.

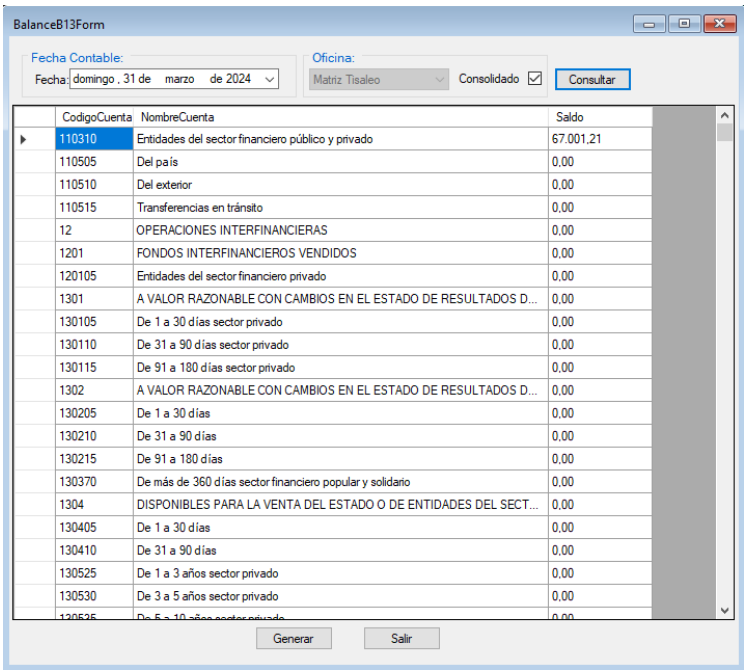


Fecha	Liquidez 1° Línea	Liquidez 2° Línea	Volatilidad	Liquidez Mínima	50% Mayores Depositantes
28/03/2024	48,55%	19,22%	1,73%	9,25%	1.020.230,52
27/03/2024	48,45%	20,17%	1,75%	9,44%	1.020.230,52
26/03/2024	26,65%	11,18%	1,75%	9,45%	1.020.230,52
25/03/2024	27,32%	11,45%	1,77%	9,48%	1.020.230,52
22/03/2024	26,69%	11,13%	1,80%	9,51%	1.020.230,52
21/03/2024	26,98%	11,28%	1,76%	9,52%	1.020.230,52
20/03/2024	26,57%	11,70%	1,79%	9,47%	1.020.230,52
19/03/2024	28,67%	11,74%	1,81%	9,48%	1.020.230,52
18/03/2024	30,07%	12,17%	1,79%	9,42%	1.020.230,52
15/03/2024	28,47%	11,82%	1,77%	9,44%	1.020.230,52
14/03/2024	26,92%	11,86%	1,77%	9,49%	1.020.230,52
13/03/2024	29,63%	12,30%	1,74%	9,47%	1.020.230,52
12/03/2024	29,74%	12,36%	1,68%	9,46%	1.020.230,52
11/03/2024	29,88%	12,30%	1,67%	9,46%	1.020.230,52
08/03/2024	31,50%	12,97%	1,65%	9,44%	1.020.230,52
07/03/2024	30,51%	12,89%	1,66%	9,45%	1.020.230,52
06/03/2024	31,41%	13,19%	1,66%	9,48%	1.020.230,52
05/03/2024	31,67%	13,31%	1,64%	9,48%	1.020.230,52
04/03/2024	31,84%	13,50%	1,61%	9,43%	1.020.230,52
01/03/2024	31,36%	13,51%	1,62%	9,42%	1.020.230,52

Figura 64. Pantalla de Indicadores

Fuente: Elaborado por autor

Por otro lado, al ingresar a Estructuras Organismos de Control > B13 – Balances Diarios se podrá visualizar la pantalla de Balances Diarios como se muestra en la figura 65, el usuario podrá generar el reporte del cual se obtienen los datos para los reportes de la Liquide Estructural.



CodigoCuenta	NombreCuenta	Saldo
110310	Entidades del sector financiero público y privado	67.001,21
110505	Del país	0,00
110510	Del exterior	0,00
110515	Transferencias en tránsito	0,00
12	OPERACIONES INTERFINANCIERAS	0,00
1201	FONDOS INTERFINANCIEROS VENDIDOS	0,00
120105	Entidades del sector financiero privado	0,00
1301	A VALOR RAZONABLE CON CAMBIOS EN EL ESTADO DE RESULTADOS D...	0,00
130105	De 1 a 30 días sector privado	0,00
130110	De 31 a 90 días sector privado	0,00
130115	De 91 a 180 días sector privado	0,00
1302	A VALOR RAZONABLE CON CAMBIOS EN EL ESTADO DE RESULTADOS D...	0,00
130205	De 1 a 30 días	0,00
130210	De 31 a 90 días	0,00
130215	De 91 a 180 días	0,00
130370	De más de 360 días sector financiero popular y solidario	0,00
1304	DISPONIBLES PARA LA VENTA DEL ESTADO O DE ENTIDADES DEL SECT...	0,00
130405	De 1 a 30 días	0,00
130410	De 31 a 90 días	0,00
130525	De 1 a 3 años sector privado	0,00
130530	De 3 a 5 años sector privado	0,00

Figura 65. Pantalla de Balances Diarios.

Fuente: Elaborado por autor

3.3.2 Proceso de alimentación de tablas de auditoria

En la tabla 26 se presenta el código para alimentar la tabla `DFI_CONT_BALANCE` se ejecuta inicialmente el procedimiento almacenado ` [dbo].[SP_DFI_CON_MAYORIZAR] `. Este procedimiento almacenado actualiza y mayoriza las cuentas contables en la base de datos. Primero, verifica y añade las cuentas del catálogo único al balance de cada agencia y del consolidado si no están presentes. Luego, ejecuta procedimientos de mayorización para cuentas transaccionales, subgrupos y grupos, asegurando que todas las cuentas reflejen las transacciones hasta la fecha actual. Adicionalmente, inicializa las cuentas de intersucursal con valores cero para mantener el balance correcto. Este proceso garantiza la integridad y actualización de los registros contables tanto a nivel de agencias individuales como del consolidado.

Tabla 26. SP_DFI_CON_MAYORIZAR. Alimentación de la tabla DFI_CONT_BALANCE

Fuente: Elaborado por autor

```
ALTER PROCEDURE [dbo].[SP_DFI_CON_MAYORIZAR]


AS
BEGIN

	declare @codigoAgencia int,
			@CodigoCuenta varchar(20),
			@codCuenta varchar(20),
			@NombreCuenta varchar(200),
			@existeCuenta int,
			@hoy date,
			@estado bit,
			@Cuc bit,
			@TotalDebe money,
			@TotalHaber money,
			@TotalSaldo money;

	declare @CuentasAuxiliares table(
		CodigoCuenta varchar(10),
		CodigoAgencia int,
		NombreCuenta varchar(200),
		Debe money,
		Haber money,
		Saldo money,
		CUC bit)

	DECLARE @CodCuentaCursor varchar(20),
			@codCuentaAux varchar(20);
```

```
	--Se debe verificar que existan las cuentas contables en su respectiva
agencia en el balance o sino se deben agregar.
	set @hoy = GETDATE();

	DECLARE Agencias CURSOR FOR
	SELECT CFIAG_CODIGO
	FROM COR_FIN_AGENCIA;

	OPEN Agencias;
	FETCH NEXT FROM Agencias
	INTO @codigoAgencia;

	WHILE @@FETCH_STATUS = 0
	BEGIN
		--Primero verifico el C.U.C para ver si todas las cuentas
existentes estan agregadas al balance
		DECLARE CatalogoUnicoCuenta CURSOR FOR
		SELECT CodigoCuenta
		FROM CatalogoUnicoCuentas
		WHERE Periodo = 2024;

		OPEN CatalogoUnicoCuenta;
		FETCH NEXT FROM CatalogoUnicoCuenta
		INTO @codCuenta;

		WHILE @@FETCH_STATUS = 0
		BEGIN

			select @existeCuenta = count(*) from Balance where
CodigoCuenta = @codCuenta and CodigoAgencia = @codigoAgencia and
FechaMayorizado = @hoy;

			if (@existeCuenta = 0)
			begin
				select @cuc = CUC, @estado = Estado, @nombreCuenta
= NombreCuenta    from CatalogoUnicoCuentas where CodigoCuenta = @codCuenta;

				INSERT INTO Balance (CodigoCuenta, CodigoAgencia,
FechaMayorizado, NombreCuenta, Debe, Haber, Saldo, MayorizadoHasta, CUC,
Estado)
				VALUES(@codCuenta, @codigoAgencia, @hoy,
@nombreCuenta, 0, 0, 0, 0, @cuc, @estado)
			end

			FETCH NEXT FROM CatalogoUnicoCuenta
			INTO @codCuenta;

		END;
		CLOSE CatalogoUnicoCuenta;
		DEALLOCATE CatalogoUnicoCuenta;

		--Mayorizar las cuentas transaccionales
		EXEC SPMayorizarCuenta
			@codAgencia = @codigoAgencia, @fechaHasta = @hoy;

		--Mayorizar el grupo
		EXEC SPMayorizarSubGrupo
			@codAgencia = @codigoAgencia, @fechaHasta = @hoy;

		--Mayorizar cuenta principal
```

```
                EXEC SPMayorizarGrupo
                        @codAgencia = @codigoAgencia, @fechaHasta = @hoy;

                FETCH NEXT FROM Agencias
                INTO @codigoAgencia;
        END;
        CLOSE Agencias;
        DEALLOCATE Agencias;

        --Inicializo las cuentas de intersucursal
         UPDATE Balance
        SET Debe = 0, Haber = 0, Saldo = 0
        FROM CatalogoUnicoCuentas
        WHERE Balance.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
                AND CatalogoUnicoCuentas.ManejaIntersucursal = 1
                AND Balance.CodigoAgencia = 0;

        -- Mayorizo el consolidado del balance.

        DECLARE CatalogoUnicoCuenta CURSOR FOR
        SELECT CodigoCuenta
        FROM CatalogoUnicoCuentas
        WHERE Periodo = 2024;

        OPEN CatalogoUnicoCuenta;
        FETCH NEXT FROM CatalogoUnicoCuenta
        INTO @codCuenta;

        WHILE @@FETCH_STATUS = 0
        BEGIN

                select @existeCuenta = count(*) from Balance where CodigoCuenta
= @codCuenta and CodigoAgencia = 0 and FechaMayorizado = @hoy;

                if (@existeCuenta = 0)
                begin
                        select @cuc = CUC, @estado = Estado, @nombreCuenta =
NombreCuenta    from CatalogoUnicoCuentas where CodigoCuenta = @codCuenta;

                        INSERT INTO Balance (CodigoCuenta, CodigoAgencia,
FechaMayorizado, NombreCuenta, Debe, Haber, Saldo, MayorizadoHasta, CUC,
Estado)
                        VALUES(@codCuenta, 0, @hoy, @nombreCuenta, 0, 0, 0, 0,
@cuc, @estado)
                end

                FETCH NEXT FROM CatalogoUnicoCuenta
                INTO @codCuenta;

        END;
        CLOSE CatalogoUnicoCuenta;
        DEALLOCATE CatalogoUnicoCuenta;

        --Mayorizar las cuentas transaccionales
        EXEC SP_DFI_CON_MAYORIZAR_CUENTA
                        @codAgencia = 0, @fechaHasta = @hoy;

        --Mayorizar el grupo
        EXEC SP_DFI_CON_MAYORIZAR_SUBGRUPO
                        @codAgencia = 0, @fechaHasta = @hoy;
```

```
        --Mayorizar cuenta principal
        EXEC SP_DFI_CON_MAYORIZAR_GRUPO
                        @codAgencia = 0, @fechaHasta = @hoy;

END
```

Después de ejecutar el procedimiento almacenado principal para llenar la tabla `DFI_CONT_BALANCE`, se procede a ejecutar los procedimientos almacenados adicionales que se especifican en el procedimiento principal `[dbo].[SP_DFI_CON_MAYORIZAR]`.

SP_DFI_CON_MAYORIZAR_CUENTA

En la tabla 27 se presenta el código del procedimiento almacenado [dbo].[SP_DFI_CON_MAYORIZAR_CUENTA] se encarga de mayorizar cuentas contables hasta una fecha específica para una agencia dada o de forma consolidada. Si se especifica una agencia, inserta en una tabla temporal las cuentas con sus respectivos débitos, créditos y saldos. Si no se especifica, verifica el balance de cuentas intersucursal y, si cuadra, actualiza las cuentas regulares y balancea las cuentas intersucursal a cero. Realiza diferentes inserciones y cálculos dependiendo de las características de las cuentas y la fecha dada, asegurando que se respeten las restricciones y condiciones establecidas, como el manejo intersucursal y los periodos contables específicos.

Tabla 27. SP_DFI_CON_MAYORIZAR_CUENTA. Alimentación de la tabla DFI_CONT_BALANCE

Fuente: Elaborado por autor

```
ALTER PROCEDURE [dbo].[SP_DFI_CON_MAYORIZAR_CUENTA]

        @codAgencia int,
        @fechaHasta date
AS
BEGIN

        declare @CantFilas int,
                        @FilaActual int,
                        @Debe money,
                        @Haber money,
                        @Saldo money,
                        @CodigoCuenta varchar(20),
                        @cuentasPorCobrar money, --19;
```

```
                @cuentasPorPagar money;          --29

        declare @CuentasAuxiliares table(
                fila INT IDENTITY (1,1) PRIMARY KEY NOT NULL, --campo para recorrer las
filas
                CodigoCuenta varchar(10),
                CodigoAgencia int,
                NombreCuenta varchar(200),
                Debe money,
                Haber money,
                Saldo money,
                CUC bit)


        --Verifico si manda a mayorizar agencias o consolidado.

        if(@codAgencia != 0)
        begin
                INSERT INTO @CuentasAuxiliares
                SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia,
CatalogoUnicoCuentas.NombreCuenta,SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0
END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber,
SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) - SUM(CASE WHEN
DebeHaber = 'H' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                 FROM DiarioTransaccion INNER JOIN
                                Diario ON DiarioTransaccion.NumeroRegistro =
Diario.NumeroRegistro INNER JOIN
                                CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta =
CatalogoUnicoCuentas.CodigoCuenta
                 WHERE (Diario.FechaContable<=@fechaHasta) and (Diario.CodigoAgencia =
@codAgencia) AND CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                                        FROM CatalogoUnicoCuentas
                                        WHERE CatalogoUnicoCuentas.CodigoCuenta LIKE
'1%' and LEN(CatalogoUnicoCuentas.CodigoCuenta) >= 4) AND CatalogoUnicoCuentas.Periodo =
2024
                 GROUP BY CatalogoUnicoCuentas.CodigoCuenta,
CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                 ORDER BY CodigoCuenta asc;

                INSERT INTO @CuentasAuxiliares
                SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia,
CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0
END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber,
SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) - SUM(CASE WHEN
DebeHaber = 'D' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                 FROM DiarioTransaccion INNER JOIN
                                Diario ON DiarioTransaccion.NumeroRegistro =
Diario.NumeroRegistro INNER JOIN
                                CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta =
CatalogoUnicoCuentas.CodigoCuenta
                 WHERE (Diario.FechaContable<=@fechaHasta) and (Diario.CodigoAgencia =
@codAgencia) AND CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                                        FROM CatalogoUnicoCuentas
                                        WHERE Estado = 1 and
CatalogoUnicoCuentas.CodigoCuenta LIKE '2%' and LEN(CatalogoUnicoCuentas.CodigoCuenta)
>= 4) AND CatalogoUnicoCuentas.Periodo = 2024
                 GROUP BY CatalogoUnicoCuentas.CodigoCuenta,
CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                 ORDER BY CodigoCuenta asc;
```

```
INSERT INTO @CuentasAuxiliares
SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia, CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
FROM DiarioTransaccion INNER JOIN
Diario ON DiarioTransaccion.NumeroRegistro = Diario.NumeroRegistro INNER JOIN
CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
WHERE (Diario.FechaContable<=@fechaHasta) and (Diario.CodigoAgencia = @codAgencia) AND CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
FROM CatalogoUnicoCuentas
WHERE Estado = 1 and CatalogoUnicoCuentas.CodigoCuenta LIKE '3%' and LEN(CatalogoUnicoCuentas.CodigoCuenta) >= 4) AND CatalogoUnicoCuentas.Periodo = 2024
GROUP BY CatalogoUnicoCuentas.CodigoCuenta, CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
ORDER BY CodigoCuenta asc;


if(@fechaHasta = '2023-12-31')
begin
INSERT INTO @CuentasAuxiliares
SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia, CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
FROM DiarioTransaccion INNER JOIN
Diario ON DiarioTransaccion.NumeroRegistro = Diario.NumeroRegistro INNER JOIN
CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
WHERE (Diario.FechaContable<=@fechaHasta) and (Diario.CodigoAgencia = @codAgencia) AND CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
FROM CatalogoUnicoCuentas
WHERE Estado = 1 and CatalogoUnicoCuentas.CodigoCuenta LIKE '4%' and LEN(CatalogoUnicoCuentas.CodigoCuenta) >= 4) AND CatalogoUnicoCuentas.Periodo = 2024 AND (DiarioTransaccion.NumeroRegistro != '2391957' AND DiarioTransaccion.NumeroRegistro != '2391958')
GROUP BY CatalogoUnicoCuentas.CodigoCuenta, CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
ORDER BY CodigoCuenta asc;

INSERT INTO @CuentasAuxiliares
SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia, CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
FROM DiarioTransaccion INNER JOIN
Diario ON DiarioTransaccion.NumeroRegistro = Diario.NumeroRegistro INNER JOIN
CatalogoUnicoCuentas on
```

```
DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
                                WHERE (Diario.FechaContable<=@fechaHasta) and (Diario.CodigoAgencia = @codAgencia) and CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                                                                FROM CatalogoUnicoCuentas
                                                                WHERE Estado = 1 and CatalogoUnicoCuentas.CodigoCuenta LIKE '5%' and LEN(CatalogoUnicoCuentas.CodigoCuenta) >= 4) AND CatalogoUnicoCuentas.Periodo = 2024 AND (DiarioTransaccion.NumeroRegistro != '2391957' AND DiarioTransaccion.NumeroRegistro != '2391958')
                                GROUP BY CatalogoUnicoCuentas.CodigoCuenta, CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                                ORDER BY CodigoCuenta asc;
            end
            else
            begin
                INSERT INTO @CuentasAuxiliares
                SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia, CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                                FROM DiarioTransaccion INNER JOIN
                                        Diario ON DiarioTransaccion.NumeroRegistro = Diario.NumeroRegistro INNER JOIN
                                        CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
                                WHERE (Diario.FechaContable<=@fechaHasta) and (Diario.CodigoAgencia = @codAgencia) AND CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                                                                FROM CatalogoUnicoCuentas
                                                                WHERE Estado = 1 and CatalogoUnicoCuentas.CodigoCuenta LIKE '4%' and LEN(CatalogoUnicoCuentas.CodigoCuenta) >= 4) AND CatalogoUnicoCuentas.Periodo = 2024
                                GROUP BY CatalogoUnicoCuentas.CodigoCuenta, CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                                ORDER BY CodigoCuenta asc;

                INSERT INTO @CuentasAuxiliares
                SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia, CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                                FROM DiarioTransaccion INNER JOIN
                                        Diario ON DiarioTransaccion.NumeroRegistro = Diario.NumeroRegistro INNER JOIN
                                        CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
                                WHERE (Diario.FechaContable<=@fechaHasta) and (Diario.CodigoAgencia = @codAgencia) and CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                                                                FROM CatalogoUnicoCuentas
                                                                WHERE Estado = 1 and CatalogoUnicoCuentas.CodigoCuenta LIKE '5%' and LEN(CatalogoUnicoCuentas.CodigoCuenta) >= 4) AND CatalogoUnicoCuentas.Periodo = 2024
                                GROUP BY CatalogoUnicoCuentas.CodigoCuenta, CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                                ORDER BY CodigoCuenta asc;
            end
```

```
INSERT INTO @CuentasAuxiliares
SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia, CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
FROM DiarioTransaccion INNER JOIN
Diario ON DiarioTransaccion.NumeroRegistro = Diario.NumeroRegistro INNER JOIN
CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
WHERE (Diario.FechaContable<=@fechaHasta) and (Diario.CodigoAgencia = @codAgencia) AND CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
FROM CatalogoUnicoCuentas
WHERE Estado = 1 and CatalogoUnicoCuentas.CodigoCuenta LIKE '71%' and LEN(CatalogoUnicoCuentas.CodigoCuenta) >= 4) AND CatalogoUnicoCuentas.Periodo = 2024
GROUP BY CatalogoUnicoCuentas.CodigoCuenta, CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
ORDER BY CodigoCuenta asc;

INSERT INTO @CuentasAuxiliares
SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia, CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
FROM DiarioTransaccion INNER JOIN
Diario ON DiarioTransaccion.NumeroRegistro = Diario.NumeroRegistro INNER JOIN
CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
WHERE (Diario.FechaContable<=@fechaHasta) and (Diario.CodigoAgencia = @codAgencia) AND CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
FROM CatalogoUnicoCuentas
WHERE Estado = 1 and CatalogoUnicoCuentas.CodigoCuenta LIKE '72%' and LEN(CatalogoUnicoCuentas.CodigoCuenta) >= 4) AND CatalogoUnicoCuentas.Periodo = 2024
GROUP BY CatalogoUnicoCuentas.CodigoCuenta, CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
ORDER BY CodigoCuenta asc;

INSERT INTO @CuentasAuxiliares
SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia, CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
FROM DiarioTransaccion INNER JOIN
Diario ON DiarioTransaccion.NumeroRegistro = Diario.NumeroRegistro INNER JOIN
CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
WHERE (Diario.FechaContable<=@fechaHasta) and (Diario.CodigoAgencia = @codAgencia) AND CatalogoUnicoCuentas.CodigoCuenta IN (SELECT
```

```
CodigoCuenta
                                        FROM CatalogoUnicoCuentas
                                        WHERE Estado = 1 and
CatalogoUnicoCuentas.CodigoCuenta LIKE '73%' and LEN(CatalogoUnicoCuentas.CodigoCuenta)
>= 4)  AND CatalogoUnicoCuentas.Periodo = 2024
                        GROUP BY CatalogoUnicoCuentas.CodigoCuenta,
CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                        ORDER BY CodigoCuenta asc;

                INSERT INTO @CuentasAuxiliares
                SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia,
CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0
END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber,
SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) - SUM(CASE WHEN
DebeHaber = 'D' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                        FROM DiarioTransaccion INNER JOIN
                                Diario ON DiarioTransaccion.NumeroRegistro =
Diario.NumeroRegistro INNER JOIN
                                CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta =
CatalogoUnicoCuentas.CodigoCuenta
                        WHERE (Diario.FechaContable<=@fechaHasta) and
(Diario.CodigoAgencia = @codAgencia) AND CatalogoUnicoCuentas.CodigoCuenta IN (SELECT
CodigoCuenta
                                        FROM CatalogoUnicoCuentas
                                        WHERE Estado = 1 and
CatalogoUnicoCuentas.CodigoCuenta LIKE '74%' and LEN(CatalogoUnicoCuentas.CodigoCuenta)
>= 4)  AND CatalogoUnicoCuentas.Periodo = 2024
                        GROUP BY CatalogoUnicoCuentas.CodigoCuenta,
CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                        ORDER BY CodigoCuenta asc;


        end
        else
        begin
                --Verifico si cuadran las cuentas transaccionales intersucursales

                SELECT @cuentasPorCobrar = (SUM(CASE WHEN DebeHaber = 'D' THEN
monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END))
                 FROM DiarioTransaccion INNER JOIN
                                Diario ON DiarioTransaccion.NumeroRegistro =
Diario.NumeroRegistro INNER JOIN
                                CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta =
CatalogoUnicoCuentas.CodigoCuenta
                 WHERE (Diario.FechaContable<=@fechaHasta) and
CatalogoUnicoCuentas.CodigoCuenta IN (SELECT  CodigoCuenta
                                        FROM CatalogoUnicoCuentas
                                        WHERE Estado = 1 and
CatalogoUnicoCuentas.CodigoCuenta LIKE '190805__' and
LEN(CatalogoUnicoCuentas.CodigoCuenta) >= 4)  AND CatalogoUnicoCuentas.Periodo = 2024;

                SELECT @cuentasPorPagar = (SUM(CASE WHEN DebeHaber = 'H' THEN
monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END))
                 FROM DiarioTransaccion INNER JOIN
                                Diario ON DiarioTransaccion.NumeroRegistro =
Diario.NumeroRegistro INNER JOIN
                                CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta =
CatalogoUnicoCuentas.CodigoCuenta
                 WHERE (Diario.FechaContable<=@fechaHasta) and
```

```
CatalogoUnicoCuentas.CodigoCuenta IN (SELECT  CodigoCuenta
                                      FROM  CatalogoUnicoCuentas
                                      WHERE  Estado  =  1  and
CatalogoUnicoCuentas.CodigoCuenta  LIKE  '290805__'  and
LEN(CatalogoUnicoCuentas.CodigoCuenta) >= 4)  AND CatalogoUnicoCuentas.Periodo = 2024;


            if(@cuentasPorCobrar = @cuentasPorPagar)
            begin
                  INSERT INTO @CuentasAuxiliares
                  SELECT  CatalogoUnicoCuentas.CodigoCuenta,  @codAgencia,
CatalogoUnicoCuentas.NombreCuenta,SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0
END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber,
SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) - SUM(CASE WHEN
DebeHaber = 'H' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                  FROM DiarioTransaccion INNER JOIN
                                    Diario  ON  DiarioTransaccion.NumeroRegistro  =
Diario.NumeroRegistro INNER JOIN
                                    CatalogoUnicoCuentas  on
DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
                  WHERE  (Diario.FechaContable<=@fechaHasta)  and
(CatalogoUnicoCuentas.ManejaIntersucursal  !=  1)  AND  CatalogoUnicoCuentas.CodigoCuenta  IN
(SELECT  CodigoCuenta
                                          FROM  CatalogoUnicoCuentas
                                          WHERE  CatalogoUnicoCuentas.CodigoCuenta
LIKE  '1%'  and  LEN(CatalogoUnicoCuentas.CodigoCuenta)  >=  4)  AND
CatalogoUnicoCuentas.Periodo = 2024
                  GROUP  BY  CatalogoUnicoCuentas.CodigoCuenta,
CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                  ORDER BY CodigoCuenta asc;

                  INSERT INTO @CuentasAuxiliares
                  SELECT  CatalogoUnicoCuentas.CodigoCuenta,  @codAgencia,
CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0
END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber,
SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) - SUM(CASE WHEN
DebeHaber = 'D' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                  FROM DiarioTransaccion INNER JOIN
                                    Diario  ON  DiarioTransaccion.NumeroRegistro  =
Diario.NumeroRegistro INNER JOIN
                                    CatalogoUnicoCuentas  on
DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
                  WHERE  (Diario.FechaContable<=@fechaHasta)  and
(CatalogoUnicoCuentas.ManejaIntersucursal  !=  1)  AND  CatalogoUnicoCuentas.CodigoCuenta  IN
(SELECT  CodigoCuenta
                                          FROM  CatalogoUnicoCuentas
                                          WHERE  Estado  =  1  and
CatalogoUnicoCuentas.CodigoCuenta LIKE '2%' and LEN(CatalogoUnicoCuentas.CodigoCuenta)
>= 4)  AND CatalogoUnicoCuentas.Periodo = 2024
                  GROUP  BY  CatalogoUnicoCuentas.CodigoCuenta,
CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                  ORDER BY CodigoCuenta asc;

                  UPDATE Balance
                  SET Debe = 0, Haber = 0, Saldo = 0
                  FROM CatalogoUnicoCuentas
                  WHERE Balance.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
                                    AND CatalogoUnicoCuentas.ManejaIntersucursal = 1
                                    AND Balance.CodigoAgencia = @codAgencia
```

```
                                        AND Balance.FechaMayorizado = @fechaHasta;
                end
                else
                begin
                        INSERT INTO @CuentasAuxiliares
                        SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia, CatalogoUnicoCuentas.NombreCuenta,SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                        FROM DiarioTransaccion INNER JOIN
                                Diario ON DiarioTransaccion.NumeroRegistro = Diario.NumeroRegistro INNER JOIN
                                CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
                        WHERE (Diario.FechaContable<=@fechaHasta) and CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                                FROM CatalogoUnicoCuentas
                                WHERE CatalogoUnicoCuentas.CodigoCuenta LIKE '1%' and LEN(CatalogoUnicoCuentas.CodigoCuenta) >= 4) AND CatalogoUnicoCuentas.Periodo = 2024
                        GROUP BY CatalogoUnicoCuentas.CodigoCuenta, CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                        ORDER BY CodigoCuenta asc;

                        INSERT INTO @CuentasAuxiliares
                        SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia, CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                        FROM DiarioTransaccion INNER JOIN
                                Diario ON DiarioTransaccion.NumeroRegistro = Diario.NumeroRegistro INNER JOIN
                                CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
                        WHERE (Diario.FechaContable<=@fechaHasta) and CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                                FROM CatalogoUnicoCuentas
                                WHERE Estado = 1 and CatalogoUnicoCuentas.CodigoCuenta LIKE '2%' and LEN(CatalogoUnicoCuentas.CodigoCuenta) >= 4) AND CatalogoUnicoCuentas.Periodo = 2024
                        GROUP BY CatalogoUnicoCuentas.CodigoCuenta, CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                        ORDER BY CodigoCuenta asc;
                end

                ---------------------------------------------------------------------------
                ---------------------------------------------------------------------------
                ----------MAYORIZO LAS CUENTAS 3, 4, 5, 6, 7----------------------------------
                ---------------------------------------------------------------------------
                ---------------------------------------------------------------------------

                INSERT INTO @CuentasAuxiliares
                SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia, CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
```

```
                    FROM DiarioTransaccion INNER JOIN
                        Diario ON DiarioTransaccion.NumeroRegistro =
Diario.NumeroRegistro INNER JOIN
                        CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta =
CatalogoUnicoCuentas.CodigoCuenta
                    WHERE (Diario.FechaContable<=@fechaHasta) and
CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                        FROM CatalogoUnicoCuentas
                        WHERE Estado = 1 and
CatalogoUnicoCuentas.CodigoCuenta LIKE '3%' and LEN(CatalogoUnicoCuentas.CodigoCuenta)
>= 4) AND CatalogoUnicoCuentas.Periodo = 2024
                    GROUP BY CatalogoUnicoCuentas.CodigoCuenta,
CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                    ORDER BY CodigoCuenta asc;


        --Para mayorizar el cierre del periodo no debo tomar en cuenta los asientos del
cierre de matriz y agencias

        if(@fechaHasta = '2023-12-31')
        begin
            INSERT INTO @CuentasAuxiliares
            SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia,
CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0
END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber,
SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) - SUM(CASE WHEN
DebeHaber = 'H' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                FROM DiarioTransaccion INNER JOIN
                    Diario ON DiarioTransaccion.NumeroRegistro =
Diario.NumeroRegistro INNER JOIN
                    CatalogoUnicoCuentas on
DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
                WHERE (Diario.FechaContable<=@fechaHasta) and
CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                    FROM CatalogoUnicoCuentas
                    WHERE Estado = 1 and
CatalogoUnicoCuentas.CodigoCuenta LIKE '4%' and LEN(CatalogoUnicoCuentas.CodigoCuenta)
>= 4) AND CatalogoUnicoCuentas.Periodo = 2024 AND (DiarioTransaccion.NumeroRegistro !=
'2391957' AND DiarioTransaccion.NumeroRegistro != '2391958')
                GROUP BY CatalogoUnicoCuentas.CodigoCuenta,
CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                ORDER BY CodigoCuenta asc;

            INSERT INTO @CuentasAuxiliares
            SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia,
CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0
END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber,
SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) - SUM(CASE WHEN
DebeHaber = 'D' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                FROM DiarioTransaccion INNER JOIN
                    Diario ON DiarioTransaccion.NumeroRegistro =
Diario.NumeroRegistro INNER JOIN
                    CatalogoUnicoCuentas on
DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
                WHERE (Diario.FechaContable<=@fechaHasta) and
CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                    FROM CatalogoUnicoCuentas
                    WHERE Estado = 1 and
CatalogoUnicoCuentas.CodigoCuenta LIKE '5%' and LEN(CatalogoUnicoCuentas.CodigoCuenta)
```

```
>= 4)  AND CatalogoUnicoCuentas.Periodo = 2024  AND (DiarioTransaccion.NumeroRegistro != '2391957' AND DiarioTransaccion.NumeroRegistro != '2391958')
                                    GROUP BY CatalogoUnicoCuentas.CodigoCuenta, CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                                    ORDER BY CodigoCuenta asc;
                        end
                else
                begin
                        INSERT INTO @CuentasAuxiliares
                        SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia, CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                                    FROM DiarioTransaccion INNER JOIN
                                            Diario ON DiarioTransaccion.NumeroRegistro = Diario.NumeroRegistro INNER JOIN
                                            CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
                                    WHERE (Diario.FechaContable<=@fechaHasta) and CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                                            FROM CatalogoUnicoCuentas
                                            WHERE Estado = 1 and CatalogoUnicoCuentas.CodigoCuenta LIKE '4%' and LEN(CatalogoUnicoCuentas.CodigoCuenta) >= 4)  AND CatalogoUnicoCuentas.Periodo = 2024
                                    GROUP BY CatalogoUnicoCuentas.CodigoCuenta, CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                                    ORDER BY CodigoCuenta asc;

                        INSERT INTO @CuentasAuxiliares
                        SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia, CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) - SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END)  AS Saldo, CatalogoUnicoCuentas.CUC
                                    FROM DiarioTransaccion INNER JOIN
                                            Diario ON DiarioTransaccion.NumeroRegistro = Diario.NumeroRegistro INNER JOIN
                                            CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta = CatalogoUnicoCuentas.CodigoCuenta
                                    WHERE (Diario.FechaContable<=@fechaHasta) and CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                                            FROM CatalogoUnicoCuentas
                                            WHERE Estado = 1 and CatalogoUnicoCuentas.CodigoCuenta LIKE '5%' and LEN(CatalogoUnicoCuentas.CodigoCuenta) >= 4)  AND CatalogoUnicoCuentas.Periodo = 2024
                                    GROUP BY CatalogoUnicoCuentas.CodigoCuenta, CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                                    ORDER BY CodigoCuenta asc;
                end



                INSERT INTO @CuentasAuxiliares
                SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia, CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) - SUM(CASE WHEN
```

```
DebeHaber = 'H' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                        FROM DiarioTransaccion INNER JOIN
                                Diario ON DiarioTransaccion.NumeroRegistro =
Diario.NumeroRegistro INNER JOIN
                                CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta =
CatalogoUnicoCuentas.CodigoCuenta
                        WHERE (Diario.FechaContable<=@fechaHasta) and
CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                                FROM CatalogoUnicoCuentas
                                WHERE Estado = 1 and
CatalogoUnicoCuentas.CodigoCuenta LIKE '71%' and LEN(CatalogoUnicoCuentas.CodigoCuenta)
>= 4) AND CatalogoUnicoCuentas.Periodo = 2024
                        GROUP BY CatalogoUnicoCuentas.CodigoCuenta,
CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                        ORDER BY CodigoCuenta asc;

                        INSERT INTO @CuentasAuxiliares
                SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia,
CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0
END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber,
SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) - SUM(CASE WHEN
DebeHaber = 'D' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                        FROM DiarioTransaccion INNER JOIN
                                Diario ON DiarioTransaccion.NumeroRegistro =
Diario.NumeroRegistro INNER JOIN
                                CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta =
CatalogoUnicoCuentas.CodigoCuenta
                        WHERE (Diario.FechaContable<=@fechaHasta) and
CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                                FROM CatalogoUnicoCuentas
                                WHERE Estado = 1 and
CatalogoUnicoCuentas.CodigoCuenta LIKE '72%' and LEN(CatalogoUnicoCuentas.CodigoCuenta)
>= 4) AND CatalogoUnicoCuentas.Periodo = 2024
                        GROUP BY CatalogoUnicoCuentas.CodigoCuenta,
CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                        ORDER BY CodigoCuenta asc;

                        INSERT INTO @CuentasAuxiliares
                SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia,
CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0
END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber,
SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0 END) - SUM(CASE WHEN
DebeHaber = 'H' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                        FROM DiarioTransaccion INNER JOIN
                                Diario ON DiarioTransaccion.NumeroRegistro =
Diario.NumeroRegistro INNER JOIN
                                CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta =
CatalogoUnicoCuentas.CodigoCuenta
                        WHERE (Diario.FechaContable<=@fechaHasta) and
CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                                FROM CatalogoUnicoCuentas
                                WHERE Estado = 1 and
CatalogoUnicoCuentas.CodigoCuenta LIKE '73%' and LEN(CatalogoUnicoCuentas.CodigoCuenta)
>= 4) AND CatalogoUnicoCuentas.Periodo = 2024
                        GROUP BY CatalogoUnicoCuentas.CodigoCuenta,
CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                        ORDER BY CodigoCuenta asc;

                        INSERT INTO @CuentasAuxiliares
```

```
                SELECT CatalogoUnicoCuentas.CodigoCuenta, @codAgencia,
CatalogoUnicoCuentas.NombreCuenta, SUM(CASE WHEN DebeHaber = 'D' THEN monto ELSE 0
END) AS Debe, SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) AS Haber,
SUM(CASE WHEN DebeHaber = 'H' THEN monto ELSE 0 END) - SUM(CASE WHEN
DebeHaber = 'D' THEN monto ELSE 0 END) AS Saldo, CatalogoUnicoCuentas.CUC
                        FROM DiarioTransaccion INNER JOIN
                                Diario ON DiarioTransaccion.NumeroRegistro =
Diario.NumeroRegistro INNER JOIN
                                CatalogoUnicoCuentas on DiarioTransaccion.CodigoCuenta =
CatalogoUnicoCuentas.CodigoCuenta
                        WHERE (Diario.FechaContable<=@fechaHasta) and
CatalogoUnicoCuentas.CodigoCuenta IN (SELECT CodigoCuenta
                                        FROM CatalogoUnicoCuentas
                                        WHERE Estado = 1 and
CatalogoUnicoCuentas.CodigoCuenta LIKE '74%' and LEN(CatalogoUnicoCuentas.CodigoCuenta)
>= 4) AND CatalogoUnicoCuentas.Periodo = 2024
                        GROUP BY CatalogoUnicoCuentas.CodigoCuenta,
CatalogoUnicoCuentas.NombreCuenta, CatalogoUnicoCuentas.CUC
                        ORDER BY CodigoCuenta asc;

        end

        select @CantFilas=COUNT(fila) FROM @CuentasAuxiliares;

        select @FilaActual = 1
        -- Recorro todas las cuentas para poder editar en el balance
        WHILE @FilaActual <= @CantFilas
        BEGIN
                -- Selecciono los datos de la cuenta a consultar para su posterior uso.
                select @CodigoCuenta = x.CodigoCuenta, @Debe = X.Debe, @Haber = X.Haber,
@Saldo = X.Saldo
                from @CuentasAuxiliares as x
                where fila=@FilaActual;

                UPDATE [dbo].[Balance]
                SET
                        [Debe] = @Debe,
                        [Haber] = @Haber,
                        [Saldo] = @Saldo
                WHERE (CodigoAgencia = @codAgencia)
                  AND (CodigoCuenta = @CodigoCuenta)
                  AND (FechaMayorizado = @fechaHasta);

                select @FilaActual = @FilaActual + 1
        END


END
```

SP_DFI_CON_MAYORIZAR_SUBGRUPO

En la tabla 28 se presenta el código del procedimiento almacenado `SP_DFI_CON_MAYORIZAR_SUBGRUPO`. Este SP recorre las cuentas auxiliares del catálogo de cuentas del año 2024, calculando los totales de 'Debe',

'Haber' y 'Saldo' para cada cuenta auxiliar, filtrados por la agencia y la fecha proporcionadas. Luego actualiza la tabla ` DFI_CONT_BALANCE` con estos totales para cada cuenta principal correspondiente.

Tabla 28. SP_DFI_CON_MAYORIZAR_SUBGRUPO. Alimentación de la tabla DFI_CONT_BALANCE

Fuente: Elaborado por autor

```
ALTER PROCEDURE [dbo].[SP_DFI_CON_MAYORIZAR_SUBGRUPO]

	@codAgencia int,
	@FechaHasta date
AS
BEGIN

	declare @TotalDebe money,
			@TotalHaber money,
			@TotalSaldo money;

	DECLARE @CodCuentaCursor varchar(20),
			@codCuentaAux varchar(20);

	--Recorro las cuentas auxiliares para obtener valores para sacar
los valores
	DECLARE CuentasGrupos CURSOR FOR
	SELECT	CodigoCuenta
	FROM	dbo.CatalogoUnicoCuentas
	WHERE	(CodigoCuenta LIKE '__')  AND
CatalogoUnicoCuentas.Periodo = 2024
	ORDER BY CodigoCuenta ASC;

	OPEN CuentasGrupos;

	FETCH NEXT FROM CuentasGrupos
	INTO @CodCuentaCursor;

	WHILE @@FETCH_STATUS = 0
	BEGIN

		set @codCuentaAux = @CodCuentaCursor + '__';

		select @TotalDebe = cast(SUM(Debe) as  decimal(16,2)),
@TotalHaber = cast(SUM(Haber) as  decimal(16,2)), @TotalSaldo =
cast(SUM(Saldo) as  decimal(16,2))
		from Balance
		where (CodigoCuenta LIKE @codCuentaAux) and (CodigoAgencia =
@codAgencia) and (FechaMayorizado = @FechaHasta);


		if (@TotalDebe IS NULL)
			Set @TotalDebe = 0;

		if (@TotalHaber IS NULL)
			Set @TotalHaber = 0;
```

```
            if (@TotalSaldo IS NULL)
                  Set @TotalSaldo = 0;

            UPDATE [dbo].[Balance]
            SET
                    [Debe] = @TotalDebe,
                    [Haber] = @TotalHaber,
                    [Saldo] = @TotalSaldo
            WHERE (CodigoAgencia = @codAgencia) AND (CodigoCuenta =
@CodCuentaCursor) and (FechaMayorizado = @FechaHasta);


            FETCH NEXT FROM CuentasGrupos
            INTO @CodCuentaCursor;

      END;

      CLOSE CuentasGrupos;
      DEALLOCATE CuentasGrupos;



END
```

SP_DFI_CON_MAYORIZAR_GRUPO

En la tabla 29 se presenta el código del procedimiento almacenado `SP_DFI_CON_MAYORIZAR_GRUPO` recorre las cuentas del catálogo de cuentas del año 2024 con un solo carácter, calculando los totales de 'Debe', 'Haber' y 'Saldo' para cada grupo de cuentas auxiliar, filtrados por la agencia y la fecha proporcionadas. Luego, actualiza la tabla ` DFI_CONT_BALANCE` con estos totales para cada cuenta principal correspondiente.

Tabla 29. SP_DFI_CON_MAYORIZAR_GRUPO. Alimentación de la tabla DFI_CONT_BALANCE

Fuente: Elaborado por autor

```
ALTER PROCEDURE [dbo].[SP_DFI_CON_MAYORIZAR_GRUPO]

       @codAgencia int,
       @FechaHasta date
AS
BEGIN

      declare @TotalDebe money,
                  @TotalHaber money,
                  @TotalSaldo money;
```

```
	DECLARE @CodCuentaCursor varchar(20),
			@codCuentaAux varchar(20);

	--Recorro las cuentas auxiliares para obtener valores para sacar
los valores
	DECLARE CuentasGrupos CURSOR FOR
	SELECT	 CodigoCuenta
	FROM	   dbo.CatalogoUnicoCuentas
	WHERE	(CodigoCuenta LIKE '_')  AND CatalogoUnicoCuentas.Periodo
= 2024
	ORDER BY CodigoCuenta ASC;

	OPEN CuentasGrupos;

	FETCH NEXT FROM CuentasGrupos
	INTO @CodCuentaCursor;

	WHILE @@FETCH_STATUS = 0
	BEGIN

		set @codCuentaAux = @CodCuentaCursor + '_';

		select @TotalDebe = cast(SUM(Debe) as  decimal(16,2)),
@TotalHaber = cast(SUM(Haber) as  decimal(16,2)), @TotalSaldo =
cast(SUM(Saldo) as  decimal(16,2))
		from Balance
		where (CodigoCuenta LIKE @codCuentaAux) and (CodigoAgencia =
@codAgencia) and (FechaMayorizado = @FechaHasta)


		if (@TotalDebe IS NULL)
			Set @TotalDebe = 0;

		if (@TotalHaber IS NULL)
			Set @TotalHaber = 0;

		if (@TotalSaldo IS NULL)
			Set @TotalSaldo = 0;

		UPDATE [dbo].[Balance]
		SET
			[Debe] = @TotalDebe,
			[Haber] = @TotalHaber,
			[Saldo] = @TotalSaldo
		WHERE (CodigoAgencia = @codAgencia) AND (CodigoCuenta =
@CodCuentaCursor) AND (FechaMayorizado = @FechaHasta);


		FETCH NEXT FROM CuentasGrupos
		INTO @CodCuentaCursor;

	END;

	CLOSE CuentasGrupos;
	DEALLOCATE CuentasGrupos;

END
```

En la tabla 30 se presenta el código de cómo se ingresaron los datos en la tabla `URI_LIQ_INDICADOR`. Primero, se define una tabla llamada [dbo].[URI_LIQ_INDICADOR] con columnas para almacenar fechas (ULIIN_FECHA), valores decimales (ULIIN_PRIMERA_LINEA, ULIIN_SEGUNDA_LINEA, ULIIN_MINIMA) y valores de dinero (ULIIN_VOLATILIDAD, ULIIN_MAYOR_DEPOSITANTE). Los datos son insertados en esta tabla con fechas y valores específicos para cada columna.

Tabla 30. Creación de la tabla `URI_LIQ_INDICADOR` e ingreso de datos.

Fuente: Elaborado por autor

```
CREATE TABLE [dbo].[URI_LIQ_INDICADOR](
	[ULIIN_FECHA] [date] NOT NULL,
	[ULIIN_PRIMERA_LINEA] [decimal](18, 15) NOT NULL,
	[ULIIN_SEGUNDA_LINEA] [decimal](18, 15) NOT NULL,
	[ULIIN_VOLATILIDAD] [decimal](18, 4) NOT NULL,
	[ULIIN_MINIMA] [decimal](18, 15) NOT NULL,
	[ULIIN_MAYOR_DEPOSITANTE] [money] NOT NULL
) ON [PRIMARY]
GO
/****** Object:  Table [dbo].[URI_LIQ_MAYOR_DEPOSITANTE]    Script Date:
05/06/2024 0:09:42 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[URI_LIQ_MAYOR_DEPOSITANTE](
	[ULIMD_FECHA] [date] NOT NULL,
	[ULIMD_IDENTIFICACION] [varchar](15) NOT NULL,
	[ULIMD_SALDO] [money] NOT NULL
) ON [PRIMARY]
GO
INSERT [dbo].[URI_LIQ_INDICADOR] ([ULIIN_FECHA], [ULIIN_PRIMERA_LINEA],
[ULIIN_SEGUNDA_LINEA], [ULIIN_VOLATILIDAD], [ULIIN_MINIMA],
[ULIIN_MAYOR_DEPOSITANTE]) VALUES (CAST(N'2021-03-22' AS Date),
CAST(0.329735989256252 AS Decimal(18, 15)), CAST(0.134086852901965 AS
Decimal(18, 15)), CAST(0.0269 AS Decimal(18, 4)), CAST(0.092128055070465
AS Decimal(18, 15)), 700967.3550)
GO
INSERT [dbo].[URI_LIQ_INDICADOR] ([ULIIN_FECHA], [ULIIN_PRIMERA_LINEA],
[ULIIN_SEGUNDA_LINEA], [ULIIN_VOLATILIDAD], [ULIIN_MINIMA],
[ULIIN_MAYOR_DEPOSITANTE]) VALUES (CAST(N'2021-03-23' AS Date),
CAST(0.333103893511342 AS Decimal(18, 15)), CAST(0.134758345011881 AS
Decimal(18, 15)), CAST(0.0269 AS Decimal(18, 4)), CAST(0.092256807211601
AS Decimal(18, 15)), 704746.3700)
GO
INSERT [dbo].[URI_LIQ_INDICADOR] ([ULIIN_FECHA], [ULIIN_PRIMERA_LINEA],
[ULIIN_SEGUNDA_LINEA], [ULIIN_VOLATILIDAD], [ULIIN_MINIMA],
[ULIIN_MAYOR_DEPOSITANTE]) VALUES (CAST(N'2021-03-24' AS Date),
CAST(0.325909567692361 AS Decimal(18, 15)), CAST(0.132552844222639 AS
Decimal(18, 15)), CAST(0.0269 AS Decimal(18, 4)), CAST(0.092066683739836
```

```
AS Decimal(18, 15)), 704225.2350)
GO
INSERT [dbo].[URI_LIQ_INDICADOR] ([ULIIN_FECHA], [ULIIN_PRIMERA_LINEA],
[ULIIN_SEGUNDA_LINEA], [ULIIN_VOLATILIDAD], [ULIIN_MINIMA],
[ULIIN_MAYOR_DEPOSITANTE]) VALUES (CAST(N'2021-03-25' AS Date),
CAST(0.332719839125103 AS Decimal(18, 15)), CAST(0.133174800228595 AS
Decimal(18, 15)), CAST(0.0269 AS Decimal(18, 4)), CAST(0.089928637525519
AS Decimal(18, 15)), 681514.4600)
GO
INSERT [dbo].[URI_LIQ_INDICADOR] ([ULIIN_FECHA], [ULIIN_PRIMERA_LINEA],
[ULIIN_SEGUNDA_LINEA], [ULIIN_VOLATILIDAD], [ULIIN_MINIMA],
[ULIIN_MAYOR_DEPOSITANTE]) VALUES (CAST(N'2021-03-26' AS Date),
CAST(0.318764134065608 AS Decimal(18, 15)), CAST(0.128545525186829 AS
Decimal(18, 15)), CAST(0.0269 AS Decimal(18, 4)), CAST(0.090016399951155
AS Decimal(18, 15)), 678763.5500)
GO
INSERT [dbo].[URI_LIQ_INDICADOR] ([ULIIN_FECHA], [ULIIN_PRIMERA_LINEA],
[ULIIN_SEGUNDA_LINEA], [ULIIN_VOLATILIDAD], [ULIIN_MINIMA],
[ULIIN_MAYOR_DEPOSITANTE]) VALUES (CAST(N'2021-03-29' AS Date),
CAST(0.333138545464942 AS Decimal(18, 15)), CAST(0.124558812272897 AS
Decimal(18, 15)), CAST(0.0265 AS Decimal(18, 4)), CAST(0.092150272265006
AS Decimal(18, 15)), 678074.2250)
GO
INSERT [dbo].[URI_LIQ_INDICADOR] ([ULIIN_FECHA], [ULIIN_PRIMERA_LINEA],
[ULIIN_SEGUNDA_LINEA], [ULIIN_VOLATILIDAD], [ULIIN_MINIMA],
[ULIIN_MAYOR_DEPOSITANTE]) VALUES (CAST(N'2021-03-30' AS Date),
CAST(0.321759596670223 AS Decimal(18, 15)), CAST(0.123083654531388 AS
Decimal(18, 15)), CAST(0.0265 AS Decimal(18, 4)), CAST(0.077387833076539
AS Decimal(18, 15)), 564535.8550)
GO
INSERT [dbo].[URI_LIQ_INDICADOR] ([ULIIN_FECHA], [ULIIN_PRIMERA_LINEA],
[ULIIN_SEGUNDA_LINEA], [ULIIN_VOLATILIDAD], [ULIIN_MINIMA],
[ULIIN_MAYOR_DEPOSITANTE]) VALUES (CAST(N'2021-03-31' AS Date),
CAST(0.296354211445000 AS Decimal(18, 15)), CAST(0.117890249824964 AS
Decimal(18, 15)), CAST(0.0265 AS Decimal(18, 4)), CAST(0.091472666910478
AS Decimal(18, 15)), 669908.3050)
GO
INSERT [dbo].[URI_LIQ_INDICADOR] ([ULIIN_FECHA], [ULIIN_PRIMERA_LINEA],
[ULIIN_SEGUNDA_LINEA], [ULIIN_VOLATILIDAD], [ULIIN_MINIMA],
[ULIIN_MAYOR_DEPOSITANTE]) VALUES (CAST(N'2021-04-01' AS Date),
CAST(0.290505092232633 AS Decimal(18, 15)), CAST(0.115857548663513 AS
Decimal(18, 15)), CAST(0.0265 AS Decimal(18, 4)), CAST(0.081551646880699
AS Decimal(18, 15)), 595692.6150)
GO
INSERT [dbo].[URI_LIQ_INDICADOR] ([ULIIN_FECHA], [ULIIN_PRIMERA_LINEA],
[ULIIN_SEGUNDA_LINEA], [ULIIN_VOLATILIDAD], [ULIIN_MINIMA],
[ULIIN_MAYOR_DEPOSITANTE]) VALUES (CAST(N'2021-04-05' AS Date),
CAST(0.252429458214849 AS Decimal(18, 15)), CAST(0.108758861508095 AS
Decimal(18, 15)), CAST(0.0258 AS Decimal(18, 4)), CAST(0.082664024801796
AS Decimal(18, 15)), 597965.9000)
GO
INSERT [dbo].[URI_LIQ_INDICADOR] ([ULIIN_FECHA], [ULIIN_PRIMERA_LINEA],
[ULIIN_SEGUNDA_LINEA], [ULIIN_VOLATILIDAD], [ULIIN_MINIMA],
[ULIIN_MAYOR_DEPOSITANTE]) VALUES (CAST(N'2021-04-06' AS Date),
CAST(0.248181202801285 AS Decimal(18, 15)), CAST(0.105458747971791 AS
Decimal(18, 15)), CAST(0.0256 AS Decimal(18, 4)), CAST(0.086802019932657
AS Decimal(18, 15)), 643248.8200)
GO
```

En la tabla 31 se presenta el código de cómo se ingresaron los datos en la tabla `URI_LIQ_MAYOR_DEPOSITANTE`. Esta tabla tiene tres columnas: ULIMD_FECHA, ULIMD_IDENTIFICACION, y ULIMD_SALDO. Posteriormente, se realizan múltiples inserciones de datos en esta tabla.

Tabla 31. Creación de la tabla `URI_LIQ_MAYOR_DEPOSITANTE` e ingreso de datos.

Fuente: Elaborado por autor

```
CREATE TABLE [dbo].[URI_LIQ_MAYOR_DEPOSITANTE](
	[ULIMD_FECHA] [date] NOT NULL,
	[ULIMD_IDENTIFICACION] [varchar](15) NOT NULL,
	[ULIMD_SALDO] [money] NOT NULL
) ON [PRIMARY]
GO
INSERT [dbo].[URI_LIQ_MAYOR_DEPOSITANTE] ([ULIMD_FECHA],
[ULIMD_IDENTIFICACION], [ULIMD_SALDO]) VALUES (CAST(N'2024-03-31' AS
Date), N'1891735002001', 256099.5600)
GO
INSERT [dbo].[URI_LIQ_MAYOR_DEPOSITANTE] ([ULIMD_FECHA],
[ULIMD_IDENTIFICACION], [ULIMD_SALDO]) VALUES (CAST(N'2024-03-31' AS
Date), N'1891726712001', 199642.5500)
GO
INSERT [dbo].[URI_LIQ_MAYOR_DEPOSITANTE] ([ULIMD_FECHA],
[ULIMD_IDENTIFICACION], [ULIMD_SALDO]) VALUES (CAST(N'2024-03-31' AS
Date), N'1891734650001', 106024.1500)
GO
INSERT [dbo].[URI_LIQ_MAYOR_DEPOSITANTE] ([ULIMD_FECHA],
[ULIMD_IDENTIFICACION], [ULIMD_SALDO]) VALUES (CAST(N'2024-03-31' AS
Date), N'591719718001', 105216.6000)
GO
INSERT [dbo].[URI_LIQ_MAYOR_DEPOSITANTE] ([ULIMD_FECHA],
[ULIMD_IDENTIFICACION], [ULIMD_SALDO]) VALUES (CAST(N'2024-03-31' AS
Date), N'1891739113001', 102465.7900)
GO
INSERT [dbo].[URI_LIQ_MAYOR_DEPOSITANTE] ([ULIMD_FECHA],
[ULIMD_IDENTIFICACION], [ULIMD_SALDO]) VALUES (CAST(N'2024-03-31' AS
Date), N'190021513001', 100899.8300)
GO
INSERT [dbo].[URI_LIQ_MAYOR_DEPOSITANTE] ([ULIMD_FECHA],
[ULIMD_IDENTIFICACION], [ULIMD_SALDO]) VALUES (CAST(N'2024-03-31' AS
Date), N'1804466124', 84791.5200)
GO
INSERT [dbo].[URI_LIQ_MAYOR_DEPOSITANTE] ([ULIMD_FECHA],
[ULIMD_IDENTIFICACION], [ULIMD_SALDO]) VALUES (CAST(N'2024-03-31' AS
Date), N'1891724299001', 81678.0400)
GO
INSERT [dbo].[URI_LIQ_MAYOR_DEPOSITANTE] ([ULIMD_FECHA],
[ULIMD_IDENTIFICACION], [ULIMD_SALDO]) VALUES (CAST(N'2024-03-31' AS
Date), N'1891743005001', 70000.0000)
```

```
GO
INSERT [dbo].[URI_LIQ_MAYOR_DEPOSITANTE] ([ULIMD_FECHA],
[ULIMD_IDENTIFICACION], [ULIMD_SALDO]) VALUES (CAST(N'2024-03-31' AS
Date), N'1803462595001', 64469.1700)
GO
INSERT [dbo].[URI_LIQ_MAYOR_DEPOSITANTE] ([ULIMD_FECHA],
[ULIMD_IDENTIFICACION], [ULIMD_SALDO]) VALUES (CAST(N'2024-03-31' AS
Date), N'1803317252', 48390.3200)
GO
INSERT [dbo].[URI_LIQ_MAYOR_DEPOSITANTE] ([ULIMD_FECHA],
[ULIMD_IDENTIFICACION], [ULIMD_SALDO]) VALUES (CAST(N'2024-03-31' AS
Date), N'1803462595', 45000.0000)
GO
```

3.3.3 Procedimientos almacenados para la generación de reportes

En la tabla 32 se presenta el código del procedimiento almacenado `SP_URI_LIQ_LIQUIDEZ_ESTRUCTURAL`. Este procedimiento maneja el cálculo y ajuste diario de saldos y liquidez estructural para diversas categorías financieras a lo largo de una semana laboral estándar. Utilizando tablas temporales, actualiza los saldos según el tipo de cuenta y día de la semana. Además, se va ajustando la volatilidad general del balance para reflejar la estabilidad financiera diaria y semanal.

Tabla 32. SP_URI_LIQ_LIQUIDEZ_ESTRUCTURAL.

Fuente: Elaborado por autor

```
ALTER PROCEDURE [dbo].[SP_URI_LIQ_LIQUIDEZ_ESTRUCTURAL]
        @fechaInicio date,
        @fechaFin date
AS
BEGIN
        DECLARE @fecha date;
        DECLARE @dias int;
        SET @fecha = @fechaInicio;
        SET @dias = 1;

        DECLARE @LiquidezEstructural AS TABLE(
                Orden int,
                Tipo varchar(10),
                Lunes float,
                Martes float,
                Miercoles float,
                Jueves float,
                Viernes float
        );
```

```
	DECLARE @NPL_LUNES float, @NPL_MARTES float, @NPL_MIERCOLES float,
@NPL_JUEVES float, @NPL_VIERNES float;
	DECLARE @DPL_LUNES float, @DPL_MARTES float, @DPL_MIERCOLES float, @DPL_JUEVES
float, @DPL_VIERNES float;
	DECLARE @DPL02_LUNES float, @DPL02_MARTES float, @DPL02_MIERCOLES float,
@DPL02_JUEVES float, @DPL02_VIERNES float;

	DECLARE @NSL_LUNES float, @NSL_MARTES float, @NSL_MIERCOLES float, @NSL_JUEVES
float, @NSL_VIERNES float;
	DECLARE @DSL_LUNES float, @DSL_MARTES float, @DSL_MIERCOLES float, @DSL_JUEVES
float, @DSL_VIERNES float;

	DECLARE		@VOLATILIDAD_GENERAL float,
		@VaR_GENERAL float;

	INSERT INTO @LiquidezEstructural
	VALUES(1, 'NPL01',0,0,0,0,0);
	INSERT INTO @LiquidezEstructural
	VALUES(2, 'NPL02',0,0,0,0,0);
	INSERT INTO @LiquidezEstructural
	VALUES(3, 'NPL03',0,0,0,0,0);
	INSERT INTO @LiquidezEstructural
	VALUES(4, 'NPL04',0,0,0,0,0);
	INSERT INTO @LiquidezEstructural
	VALUES(5, 'NPL05',0,0,0,0,0);


	INSERT INTO @LiquidezEstructural
	VALUES(7, 'DPL01',0,0,0,0,0);
	INSERT INTO @LiquidezEstructural
	VALUES(8, 'DPL02',0,0,0,0,0);
	INSERT INTO @LiquidezEstructural
	VALUES(9, 'DPL03',0,0,0,0,0);
	INSERT INTO @LiquidezEstructural
	VALUES(10, 'DPL04',0,0,0,0,0);

	INSERT INTO @LiquidezEstructural
	VALUES(15, 'NSL01',0,0,0,0,0);
	INSERT INTO @LiquidezEstructural
	VALUES(16, 'NSL02',0,0,0,0,0);
	INSERT INTO @LiquidezEstructural
	VALUES(17, 'NSL03',0,0,0,0,0);

	INSERT INTO @LiquidezEstructural
	VALUES(19, 'DSL01',0,0,0,0,0);
	INSERT INTO @LiquidezEstructural
	VALUES(22, 'DSL02',0,0,0,0,0);
	INSERT INTO @LiquidezEstructural
	VALUES(23, 'DSL03',0,0,0,0,0);

	INSERT INTO @LiquidezEstructural
	VALUES(13, 'VOLGEN', 0, 0, 0, 0, 0);

	WHILE @dias <= 5
	BEGIN
		WITH BalanceDiarioGeneral AS (
```

```
SELECT CodigoCuenta
,CASE
WHEN CodigoCuenta = '1105' OR CodigoCuenta = '2201' OR CodigoCuenta = '2102' OR CodigoCuenta = '2202' THEN -Saldo
ELSE Saldo
END Saldo
,CASE
WHEN CodigoCuenta = '11' OR CodigoCuenta = '1105' THEN 'NPL01'
WHEN CodigoCuenta = '1201' OR CodigoCuenta = '2201' OR CodigoCuenta = '1202' OR CodigoCuenta = '2102' OR CodigoCuenta = '2202' THEN 'NPL02'
WHEN CodigoCuenta = '130105' OR CodigoCuenta = '130110' OR CodigoCuenta = '130150' OR CodigoCuenta = '130155' OR CodigoCuenta = '130205' OR CodigoCuenta = '130210' THEN 'NPL03'
WHEN CodigoCuenta = '130305' OR CodigoCuenta = '130310' OR CodigoCuenta = '130350' OR CodigoCuenta = '130355' OR CodigoCuenta = '130405' OR CodigoCuenta = '130410' THEN 'NPL04'
WHEN CodigoCuenta = '190286' THEN 'NPL05'
WHEN CodigoCuenta = '2101' THEN 'DPL01'
WHEN CodigoCuenta = '210305' OR CodigoCuenta = '210310' THEN 'DPL02'
WHEN
CodigoCuenta = '2601' OR
CodigoCuenta = '260205' OR CodigoCuenta = '260210' OR CodigoCuenta = '260250' OR CodigoCuenta = '260255' OR
CodigoCuenta = '260305' OR CodigoCuenta = '260310' OR
CodigoCuenta = '260405' OR CodigoCuenta = '260410' OR CodigoCuenta = '260450' OR CodigoCuenta = '260455' OR
CodigoCuenta = '260605' OR CodigoCuenta = '260610' OR
CodigoCuenta = '260705' OR CodigoCuenta = '260710' OR
CodigoCuenta = '260905' OR CodigoCuenta = '260910' OR
CodigoCuenta = '261005' OR CodigoCuenta = '261010' OR CodigoCuenta = '261090' THEN 'DPL03'
WHEN CodigoCuenta = '23' OR CodigoCuenta = '24' OR CodigoCuenta = '27' OR CodigoCuenta = '2903' THEN 'DPL04'
WHEN CodigoCuenta = '130115' OR CodigoCuenta = '130160' OR CodigoCuenta = '130215' THEN 'NSL01'
WHEN CodigoCuenta = '130315' OR CodigoCuenta = '130360' OR CodigoCuenta = '130415' THEN 'NSL02'
WHEN CodigoCuenta = '130505' OR CodigoCuenta = '130510' OR CodigoCuenta = '130515' OR CodigoCuenta = '130550' OR CodigoCuenta = '130555' OR
CodigoCuenta = '130560' OR CodigoCuenta = '130605' OR CodigoCuenta = '130610' OR CodigoCuenta = '130615' THEN 'NSL03'
WHEN CodigoCuenta = '2103' THEN 'DSL01'
WHEN CodigoCuenta = '2105' THEN 'DSL02'
WHEN CodigoCuenta = '260215' OR CodigoCuenta = '260220' OR CodigoCuenta = '260260' OR CodigoCuenta = '260265' OR CodigoCuenta = '260315' OR
CodigoCuenta = '260320' OR CodigoCuenta = '260415' OR CodigoCuenta = '260420' OR CodigoCuenta = '260460' OR CodigoCuenta = '260465' OR
CodigoCuenta = '260615' OR CodigoCuenta = '260620' OR CodigoCuenta = '260715' OR CodigoCuenta = '260720' OR CodigoCuenta = '260915' OR CodigoCuenta = '260920' THEN 'DSL03'
```

```
                                    END Tipo
                            FROM Balance
                            WHERE FechaMayorizado = @fecha  AND CodigoAgencia = 0 AND
                            (
                                    CodigoCuenta = '11' OR CodigoCuenta = '1105' OR
                                    CodigoCuenta = '1201' OR CodigoCuenta = '2201' OR
CodigoCuenta = '1202' OR CodigoCuenta = '2102' OR CodigoCuenta = '2202' OR
                                    CodigoCuenta = '130105' OR CodigoCuenta = '130110' OR
CodigoCuenta = '130150' OR CodigoCuenta = '130155' OR CodigoCuenta = '130205'  OR
CodigoCuenta = '130210' OR
                                    CodigoCuenta = '130305' OR CodigoCuenta = '130310' OR
CodigoCuenta = '130350' OR CodigoCuenta = '130355' OR CodigoCuenta = '130405'  OR
CodigoCuenta = '130410' OR
                                    CodigoCuenta = '190286' OR
                                    CodigoCuenta = '2101' OR
                                    CodigoCuenta = '210305' OR CodigoCuenta = '210310' OR
                                    CodigoCuenta = '2601' OR
                                    CodigoCuenta = '260205' OR CodigoCuenta = '260210' OR
CodigoCuenta = '260250' OR CodigoCuenta = '260255' OR
                                    CodigoCuenta = '260305' OR CodigoCuenta = '260310' OR
                                    CodigoCuenta = '260405' OR CodigoCuenta = '260410' OR
CodigoCuenta = '260450' OR CodigoCuenta = '260455' OR
                                    CodigoCuenta = '260605' OR CodigoCuenta = '260610' OR
                                    CodigoCuenta = '260705' OR CodigoCuenta = '260710' OR
                                    CodigoCuenta = '260905' OR CodigoCuenta = '260910' OR
                                    CodigoCuenta = '261005' OR CodigoCuenta = '261010' OR
CodigoCuenta = '261090' OR
                                    CodigoCuenta = '23' OR CodigoCuenta = '24' OR CodigoCuenta =
'27' OR CodigoCuenta = '2903' OR
                                    CodigoCuenta = '130115' OR CodigoCuenta = '130160' OR
CodigoCuenta = '130215' OR
                                    CodigoCuenta = '130315' OR CodigoCuenta = '130360' OR
CodigoCuenta = '130415' OR
                                    CodigoCuenta = '130505' OR CodigoCuenta = '130510' OR
CodigoCuenta = '130515' OR CodigoCuenta = '130550' OR CodigoCuenta = '130555' OR
                                    CodigoCuenta = '130560' OR CodigoCuenta = '130605' OR
CodigoCuenta = '130610' OR CodigoCuenta = '130615' OR
                                    CodigoCuenta = '2103' OR
                                    CodigoCuenta = '2105' OR
                                    CodigoCuenta = '260215' OR CodigoCuenta = '260220' OR
CodigoCuenta = '260260' OR CodigoCuenta = '260265' OR CodigoCuenta = '260315' OR
                                    CodigoCuenta = '260320' OR CodigoCuenta = '260415' OR
CodigoCuenta = '260420' OR CodigoCuenta = '260460' OR CodigoCuenta = '260465' OR
                                    CodigoCuenta = '260615' OR CodigoCuenta = '260620' OR
CodigoCuenta = '260715' OR CodigoCuenta = '260720' OR CodigoCuenta = '260915' OR
CodigoCuenta = '260920'
                             )
                    ),
                    BalanceDiario AS (
                            SELECT Tipo, SUM(Saldo) Valor
                            FROM BalanceDiarioGeneral
                            WHERE Tipo IS NOT NULL
                            GROUP BY Tipo
                    )
                    UPDATE LiquidezEstructural
                    SET
```

```
                        LiquidezEstructural.Lunes = CASE WHEN @dias = 1 THEN
BalanceDiario.Valor ELSE LiquidezEstructural.Lunes END,
                        LiquidezEstructural.Martes = CASE WHEN @dias = 2 THEN
BalanceDiario.Valor ELSE LiquidezEstructural.Martes END,
                        LiquidezEstructural.Miercoles = CASE WHEN @dias = 3 THEN
BalanceDiario.Valor ELSE LiquidezEstructural.Miercoles END,
                        LiquidezEstructural.Jueves = CASE WHEN @dias = 4 THEN
BalanceDiario.Valor ELSE LiquidezEstructural.Jueves END,
                        LiquidezEstructural.Viernes = CASE WHEN @dias = 5 THEN
BalanceDiario.Valor ELSE LiquidezEstructural.Viernes END
                FROM @LiquidezEstructural LiquidezEstructural
                        INNER JOIN BalanceDiario ON LiquidezEstructural.Tipo =
BalanceDiario.Tipo

                EXEC    [dbo].[SP_URI_LIQ_VOLATILIDAD]
                @fechaFin = @fecha,
                @imprimirTablas = 0,
                @OUT_VaR_GENERAL = @VaR_GENERAL OUTPUT,
                @OUT_VOLATILIDAD_GENERAL = @VOLATILIDAD_GENERAL OUTPUT

                UPDATE LiquidezEstructural
                SET
                        LiquidezEstructural.Lunes = CASE WHEN @dias = 1 THEN
@VOLATILIDAD_GENERAL ELSE LiquidezEstructural.Lunes END,
                        LiquidezEstructural.Martes = CASE WHEN @dias = 2 THEN
@VOLATILIDAD_GENERAL ELSE LiquidezEstructural.Martes END,
                        LiquidezEstructural.Miercoles = CASE WHEN @dias = 3 THEN
@VOLATILIDAD_GENERAL ELSE LiquidezEstructural.Miercoles END,
                        LiquidezEstructural.Jueves = CASE WHEN @dias = 4 THEN
@VOLATILIDAD_GENERAL ELSE LiquidezEstructural.Jueves END,
                        LiquidezEstructural.Viernes = CASE WHEN @dias = 5 THEN
@VOLATILIDAD_GENERAL ELSE LiquidezEstructural.Viernes END
                FROM @LiquidezEstructural LiquidezEstructural
                WHERE Tipo = 'VOLGEN'

                SET @fecha = DATEADD(DAY, @dias, @fechaInicio);
                SET @dias = @dias + 1;
        END

        INSERT INTO @LiquidezEstructural
        SELECT 6, 'NPLT', SUM(Lunes), SUM(Martes), SUM(Miercoles), SUM(Jueves),
        SUM(Viernes)
        FROM @LiquidezEstructural
        WHERE Tipo = 'NPL01' OR Tipo = 'NPL02' OR Tipo = 'NPL03' OR Tipo = 'NPL04' OR Tipo =
'NPL05';

        INSERT INTO @LiquidezEstructural
        SELECT 11, 'DPLT', SUM(Lunes), SUM(Martes), SUM(Miercoles), SUM(Jueves),
        SUM(Viernes)
        FROM @LiquidezEstructural
        WHERE Tipo = 'DPL01' OR Tipo = 'DPL02' OR Tipo = 'DPL03' OR Tipo = 'DPL04';

        SELECT @NPL_LUNES = Lunes, @NPL_MARTES = Martes, @NPL_MIERCOLES = Miercoles,
@NPL_JUEVES = Jueves, @NPL_VIERNES = Viernes
        FROM @LiquidezEstructural
        WHERE Tipo = 'NPLT';
```

```
	SELECT @DPL_LUNES = Lunes, @DPL_MARTES = Martes, @DPL_MIERCOLES = Miercoles,
@DPL_JUEVES = Jueves, @DPL_VIERNES = Viernes
	FROM @LiquidezEstructural
	WHERE Tipo = 'DPLT';

	INSERT INTO @LiquidezEstructural
	VALUES(
		12,
		'LPL',
		CASE WHEN @DPL_LUNES = 0 THEN 0 ELSE @NPL_LUNES/@DPL_LUNES END,
		CASE WHEN @DPL_MARTES = 0 THEN 0 ELSE @NPL_MARTES/@DPL_MARTES
END,
		CASE WHEN @DPL_MIERCOLES = 0 THEN 0 ELSE
@NPL_MIERCOLES/@DPL_MIERCOLES END,
		CASE WHEN @DPL_JUEVES = 0 THEN 0 ELSE @NPL_JUEVES/@DPL_JUEVES END,
		CASE WHEN @DPL_VIERNES = 0 THEN 0 ELSE @NPL_VIERNES/@DPL_VIERNES
END
	);


	INSERT INTO @LiquidezEstructural
	SELECT
		14, 'VOLPL', Lunes*2, Martes*2, Miercoles*2, Jueves*2, Viernes*2
	FROM @LiquidezEstructural
	WHERE Tipo = 'VOLGEN'

	INSERT INTO @LiquidezEstructural
	SELECT 18, 'NSLT', SUM(Lunes), SUM(Martes), SUM(Miercoles), SUM(Jueves),
	SUM(Viernes)
	FROM @LiquidezEstructural
	WHERE Tipo = 'NSL01' OR Tipo = 'NSL02' OR Tipo = 'NSL03' OR Tipo = 'NPLT';

	SELECT @DPL02_LUNES = Lunes, @DPL02_MARTES = Martes, @DPL02_MIERCOLES =
Miercoles, @DPL02_JUEVES = Jueves, @DPL02_VIERNES = Viernes
	FROM @LiquidezEstructural
	WHERE Tipo = 'DPL02';

	UPDATE @LiquidezEstructural
	SET Lunes = LUNES - @DPL02_LUNES,
		Martes = Martes - @DPL02_MARTES,
		Miercoles = Miercoles - @DPL02_MIERCOLES,
		Jueves = Jueves - @DPL02_JUEVES,
		Viernes = Viernes - @DPL02_VIERNES
	WHERE Tipo = 'DSL01';

	INSERT INTO @LiquidezEstructural
	SELECT 24, 'DSLT', SUM(Lunes), SUM(Martes), SUM(Miercoles), SUM(Jueves),
	SUM(Viernes)
	FROM @LiquidezEstructural
	WHERE Tipo = 'DSL01' OR Tipo = 'DSL02' OR Tipo = 'DSL03' OR Tipo = 'DPLT';

	SELECT @NSL_LUNES = Lunes, @NSL_MARTES = Martes, @NSL_MIERCOLES = Miercoles,
@NSL_JUEVES = Jueves, @NSL_VIERNES = Viernes
	FROM @LiquidezEstructural
	WHERE Tipo = 'NSLT';
```

```
	SELECT @DSL_LUNES = Lunes, @DSL_MARTES = Martes, @DSL_MIERCOLES = Miercoles,
@DSL_JUEVES = Jueves, @DSL_VIERNES = Viernes
	FROM @LiquidezEstructural
	WHERE Tipo = 'DSLT';

	INSERT INTO @LiquidezEstructural
	VALUES(
		25,
		'LSL',
		CASE WHEN @DSL_LUNES = 0 THEN 0 ELSE @NSL_LUNES/@DSL_LUNES END,
		CASE WHEN @DSL_MARTES  = 0 THEN 0 ELSE @NSL_MARTES/@DSL_MARTES
END,
		CASE WHEN @DSL_MIERCOLES  = 0 THEN 0 ELSE
@NSL_MIERCOLES/@DSL_MIERCOLES END,
		CASE WHEN @DSL_JUEVES  = 0 THEN 0 ELSE @NSL_JUEVES/@DSL_JUEVES END,
		CASE WHEN @DSL_VIERNES  = 0 THEN 0 ELSE @NSL_VIERNES/@DSL_VIERNES
END
	);

	INSERT INTO @LiquidezEstructural
	SELECT
		26,
		'VOLSL',
		Lunes*2.5,
		Martes*2.5,
		Miercoles*2.5,
		Jueves*2.5,
		Viernes*2.5
	FROM @LiquidezEstructural
	WHERE Tipo = 'VOLGEN'

	INSERT INTO @LiquidezEstructural
	SELECT
		27,
		'VOLABS',
		@DSL_LUNES*Lunes,
		@DSL_MARTES*Martes,
		@DSL_MIERCOLES*Miercoles,
		@DSL_JUEVES*Jueves,
		@DSL_VIERNES*Viernes
	FROM @LiquidezEstructural
	WHERE Tipo = 'VOLSL';

	SELECT
		CASE
			WHEN Tipo='LPL' OR Tipo='VOLGEN' OR Tipo='VOLPL' OR Tipo='LSL' OR
Tipo='VOLSL'   THEN CAST(ROUND(Lunes,4) AS decimal(18,4))
			ELSE CAST(ROUND(Lunes,2) AS decimal(18,2))
		END Lunes,
		CASE
			WHEN Tipo='LPL' OR Tipo='VOLGEN' OR Tipo='VOLPL' OR Tipo='LSL' OR
Tipo='VOLSL'   THEN CAST(ROUND(Martes,4) AS decimal(18,4))
			ELSE CAST(ROUND(Martes,2) AS decimal(18,2))
		END Martes,
		CASE
```

```
                        WHEN Tipo='LPL' OR Tipo='VOLGEN' OR Tipo='VOLPL' OR Tipo='LSL' OR
Tipo='VOLSL'    THEN CAST(ROUND(Miercoles,4) AS decimal(18,4))
                        ELSE CAST(ROUND(Miercoles,2) AS decimal(18,2))
                    END Miercoles,
                    CASE
                        WHEN Tipo='LPL' OR Tipo='VOLGEN' OR Tipo='VOLPL' OR Tipo='LSL' OR
Tipo='VOLSL'    THEN CAST(ROUND(Jueves,4) AS decimal(18,4))
                        ELSE CAST(ROUND(Jueves,2) AS decimal(18,2))
                    END Jueves,
                    CASE
                        WHEN Tipo='LPL' OR Tipo='VOLGEN' OR Tipo='VOLPL' OR Tipo='LSL' OR
Tipo='VOLSL'    THEN ROUND(Viernes,4)
                        ELSE CAST(ROUND(Viernes,2) AS decimal(18,2))
                    END Viernes,
                    CASE
                        WHEN Tipo='LPL' OR Tipo='VOLGEN' OR Tipo='VOLPL' OR Tipo='LSL' OR
Tipo='VOLSL'    THEN CAST(ROUND((Lunes + Martes + Miercoles + Jueves + Viernes)/5,4) AS
decimal(18,4))
                        ELSE CAST(ROUND((Lunes + Martes + Miercoles + Jueves + Viernes)/5,2)
AS decimal(18,2))
                    END
                    Promedio,
                    Tipo
        FROM @LiquidezEstructural
        ORDER BY Orden
END
```

En la tabla 33 se presenta el código del procedimiento almacenado `SP_URI_LIQ_VOLATILIDAD`. En este procedimiento se calcula y registra varios indicadores clave relacionados con la volatilidad financiera. Utiliza tablas temporales para almacenar datos intermedios y realiza cálculos sobre saldos financieros diarios de diferentes categorías. Incluye operaciones para determinar variaciones, desviaciones estándar, y correlaciones entre estos saldos. Finalmente, calcula y devuelve el VaR (Valor en Riesgo) general y la volatilidad general, que son almacenados en una tabla `@General`.

Tabla 33. SP_URI_LIQ_VOLATILIDAD.

Fuente: Elaborado por autor

```
ALTER PROCEDURE [dbo].[SP_URI_LIQ_VOLATILIDAD]
        @fechaFin date,
        @imprimirTablas bit,
        @OUT_VaR_GENERAL float OUT,
        @OUT_VOLATILIDAD_GENERAL float OUT
AS
BEGIN
DECLARE  @fecha date;
        DECLARE @dias int;
        SET @dias = 1;
```

```
	DECLARE @Volatilidad AS TABLE(
		[ID] int IDENTITY(1,1) PRIMARY KEY,
		[TIPO] varchar(50),
		[FECHA] date,
		[2101-210135] float,
		[210135] float,
		[2103] float,
		[2104] float,
		[2105] float,
		[TOTAL] float
	);

	WITH Fechas AS		(
		SELECT TOP 90 FechaMayorizado
		FROM Balance
		WHERE FechaMayorizado <= @fechaFin AND FechaMayorizado NOT IN (
			SELECT FechaFeriado FROM Feriado
		) AND DATENAME(WEEKDAY,FechaMayorizado) != 'Sábado' AND
DATENAME(WEEKDAY,FechaMayorizado) != 'Domingo'
		GROUP BY FechaMayorizado
		ORDER BY FechaMayorizado DESC
	), BalanceGeneral AS (
		SELECT
			CodigoCuenta, Balance.FechaMayorizado, Saldo
		FROM Balance
			INNER JOIN Fechas ON Balance.FechaMayorizado =
Fechas.FechaMayorizado
		WHERE CodigoAgencia = 0 AND
				(
				CodigoCuenta = '2101' OR
				CodigoCuenta = '210135' OR
				CodigoCuenta = '2103' OR
				CodigoCuenta = '2104' OR
				CodigoCuenta = '2105'
				)
	), SaldosSeparado AS (
		SELECT *
		FROM (
			SELECT *
			FROM BalanceGeneral
		) AS BalanceEnFilas PIVOT (
			SUM(Saldo) FOR CodigoCuenta IN (
					[2101],
					[210135],
					[2103],
					[2104],
					[2105]
					)
		) AS BalanceEnColumnas
	)
	INSERT INTO @Volatilidad
	SELECT 'SALDOS' [TIPO], [FechaMayorizado], [2101]-[210135] [2101-210135],
[210135], [2103], [2104], [2105], (([2101]-[210135]) + [210135] + [2103] + [2104] + [2105])
[TOTAL]
	FROM SaldosSeparado
	ORDER BY [FechaMayorizado];

	WITH SaldosVaraciones AS (
```

```
		SELECT TOP 60
			[FECHA],
			[2101-210135] [Actual 2101-210135],
			[210135] [Actual 210135],
			[2103] [Actual 2103],
			[2104] [Actual 2104],
			[2105] [Actual 2105],
			(SELECT SaldoAnterior.[2101-210135] FROM @Volatilidad SaldoAnterior WHERE SaldoAnterior.ID = SaldoActual.ID - 30) [Anterior 2101-210135],
			(SELECT SaldoAnterior.[210135] FROM @Volatilidad SaldoAnterior WHERE SaldoAnterior.ID = SaldoActual.ID - 30) [Anterior 210135],
			(SELECT SaldoAnterior.[2103] FROM @Volatilidad SaldoAnterior WHERE SaldoAnterior.ID = SaldoActual.ID - 30) [Anterior 2103],
			(SELECT SaldoAnterior.[2104] FROM @Volatilidad SaldoAnterior WHERE SaldoAnterior.ID = SaldoActual.ID - 30) [Anterior 2104],
			(SELECT SaldoAnterior.[2105] FROM @Volatilidad SaldoAnterior WHERE SaldoAnterior.ID = SaldoActual.ID - 30) [Anterior 2105]
		FROM @Volatilidad SaldoActual
		ORDER BY FECHA DESC
	), Variaciones AS (
		SELECT
			FECHA [FECHA],
			CASE
				WHEN [Actual 2101-210135] = 0 AND [Anterior 2101-210135] = 0 THEN 0
				WHEN [Actual 2101-210135] = 0 AND [Anterior 2101-210135] > 0 THEN -CAST(ROUND(LOG(2),10) AS decimal(18,10))
				WHEN [Actual 2101-210135] > 0 AND [Anterior 2101-210135] = 0 THEN CAST(ROUND(LOG(2),10) AS decimal(18,10))
				ELSE CAST(ROUND(LOG(CAST(ROUND([Actual 2101-210135]/[Anterior 2101-210135],10) AS decimal(18,10))),10) AS decimal(18,10))
			END [2101-210135],
			CASE
				WHEN [Actual 210135] = 0 AND [Anterior 210135] = 0 THEN 0
				WHEN [Actual 210135] = 0 AND [Anterior 210135] > 0 THEN -CAST(ROUND(LOG(2),10) AS decimal(18,10))
				WHEN [Actual 210135] > 0 AND [Anterior 210135] = 0 THEN CAST(ROUND(LOG(2),10) AS decimal(18,10))
				ELSE CAST(ROUND(LOG(CAST(ROUND([Actual 210135]/[Anterior 210135],10) AS decimal(18,10))),10) AS decimal(18,10))
			END [210135],
			CASE
				WHEN [Actual 2103] = 0 AND [Anterior 2103] = 0 THEN 0
				WHEN [Actual 2103] = 0 AND [Anterior 2103] > 0 THEN -CAST(ROUND(LOG(2),10) AS decimal(18,10))
				WHEN [Actual 2103] > 0 AND [Anterior 2103] = 0 THEN CAST(ROUND(LOG(2),10) AS decimal(18,10))
				ELSE CAST(ROUND(LOG(CAST(ROUND([Actual 2103]/[Anterior 2103],10) AS decimal(18,10))),10) AS decimal(18,10))
			END [2103],
			CASE
				WHEN [Actual 2104] = 0 AND [Anterior 2104] = 0 THEN 0
				WHEN [Actual 2104] = 0 AND [Anterior 2104] > 0 THEN -CAST(ROUND(LOG(2),10) AS decimal(18,10))
				WHEN [Actual 2104] > 0 AND [Anterior 2104] = 0 THEN CAST(ROUND(LOG(2),10) AS decimal(18,10))
				ELSE CAST(ROUND(LOG(CAST(ROUND([Actual 2104]/[Anterior 2104],10) AS decimal(18,10))),10) AS decimal(18,10))
```

```
                        END [2104],
                        CASE
                                WHEN [Actual 2105] = 0 AND [Anterior 2105] = 0 THEN 0
                                WHEN [Actual 2105] = 0 AND [Anterior 2105] > 0 THEN -
CAST(ROUND(LOG(2),10) AS decimal(18,10))
                                WHEN [Actual 2105] > 0 AND [Anterior 2105] = 0 THEN
CAST(ROUND(LOG(2),10) AS decimal(18,10))
                                ELSE CAST(ROUND(LOG(CAST(ROUND([Actual
2105]/[Anterior 2105],10) AS decimal(18,10))),10) AS decimal(18,10))
                        END [2105]
                FROM SaldosVaraciones
        )
        INSERT INTO @Volatilidad
        SELECT
                'VARIACIONES' [TIPO], [FECHA], [2101-210135], [210135], [2103], [2104],
[2105], NULL
        FROM Variaciones
        ORDER BY [FECHA];

        DECLARE
                @AVG_2101_210135 float,
                @AVG_210135 float,
                @AVG_2103 float,
                @AVG_2104 float,
                @AVG_2105 float;

        SELECT
                @AVG_2101_210135 = CAST(ROUND(CAST(ROUND(SUM([2101-
210135]),10) AS decimal(18,10))/60,10) AS decimal(18,10)),
                @AVG_210135 = CAST(ROUND(CAST(ROUND(SUM([210135]),10) AS
decimal(18,10))/60,10) AS decimal(18,10)),
                @AVG_2103 = CAST(ROUND(CAST(ROUND(SUM([2103]),10) AS
decimal(18,10))/60,10) AS decimal(18,10)),
                @AVG_2104 = CAST(ROUND(CAST(ROUND(SUM([2104]),10) AS
decimal(18,10))/60,10) AS decimal(18,10)),
                @AVG_2105 = CAST(ROUND(CAST(ROUND(SUM([2105]),10) AS
decimal(18,10))/60,10) AS decimal(18,10))
        FROM @Volatilidad
        WHERE [TIPO] = 'VARIACIONES';

        INSERT INTO @Volatilidad
        VALUES('AVG VARIA.', NULL, @AVG_2101_210135, @AVG_210135, @AVG_2103,
@AVG_2104, @AVG_2105, NULL);

        INSERT INTO @Volatilidad
        SELECT 'DESVIACIONES' [TIPO],[FECHA],
                CAST(ROUND(POWER([2101-210135]-@AVG_2101_210135,2),10) AS
decimal(18,10)),
                CAST(ROUND(POWER([210135]-@AVG_210135,2),10) AS decimal(18,10)),
                CAST(ROUND(POWER([2103]-@AVG_2103,2),10) AS decimal(18,10)),
                CAST(ROUND(POWER([2104]-@AVG_2104,2),10) AS decimal(18,10)),
                CAST(ROUND(POWER([2105]-@AVG_2105,2), 8) AS decimal(18,10)),
                NULL
        FROM @Volatilidad
        WHERE [TIPO] = 'VARIACIONES'
        ORDER BY FECHA;

        DECLARE
                @SUM_2101_210135 float,
```

```
		@SUM_210135 float,
		@SUM_2103 float,
		@SUM_2104 float,
		@SUM_2105 float;

	SELECT
		@SUM_2101_210135 = CAST(ROUND(SUM([2101-210135]),10) AS
decimal(18,10)),
		@SUM_210135 = CAST(ROUND(SUM([210135]),10) AS decimal(18,10)),
		@SUM_2103 = CAST(ROUND(SUM([2103]),10) AS decimal(18,10)),
		@SUM_2104 = CAST(ROUND(SUM([2104]),10) AS decimal(18,10)),
		@SUM_2105 = CAST(ROUND(SUM([2105]), 8) AS decimal(18, 8))
	FROM @Volatilidad
	WHERE [TIPO] = 'DESVIACIONES';

	INSERT INTO @Volatilidad
	VALUES('SUM DESVIA.', NULL, @SUM_2101_210135, @SUM_210135, @SUM_2103,
@SUM_2104, @SUM_2105, NULL);

	INSERT INTO @Volatilidad
	VALUES(
		'AVG DESVIA.',
		NULL,
		CAST(ROUND(@SUM_2101_210135/59,10) AS decimal(18,10)),
		CAST(ROUND(@SUM_210135/59,10) AS decimal(18,10)),
		CAST(ROUND(@SUM_2103/59,10) AS decimal(18,10)),
		CAST(ROUND(@SUM_2104/59,10) AS decimal(18,10)),
		CAST(ROUND(@SUM_2105/59,10) AS decimal(18,10)),
		NULL);

	INSERT INTO @Volatilidad
	SELECT
		'SQRT DESVIA.',
		@fechaFin,
		CAST(ROUND(SQRT([2101-210135]),10) AS decimal(18,10)),
		CAST(ROUND(SQRT([210135]),10) AS decimal(18,10)),
		CAST(ROUND(SQRT([2103]),10) AS decimal(18,10)),
		CAST(ROUND(SQRT([2104]),10) AS decimal(18,10)),
		CAST(ROUND(SQRT([2105]),10) AS decimal(18,10)),
		NULL
	FROM @Volatilidad
	WHERE [TIPO] = 'AVG DESVIA.';

	INSERT INTO @Volatilidad
	SELECT
		'VAR POR FF',
		Saldo.[FECHA],
		CAST(ROUND(Sqrt_Desvia.[2101-210135]*Saldo.[2101-210135],10) AS
decimal(18,10)),
		CAST(ROUND(Sqrt_Desvia.[210135]*Saldo.[210135],10) AS decimal(18,10)),
		CAST(ROUND(Sqrt_Desvia.[2103]*Saldo.[2103],10) AS decimal(18,10)),
		CAST(ROUND(Sqrt_Desvia.[2104]*Saldo.[2104],10) AS decimal(18,10)),
		CAST(ROUND(Sqrt_Desvia.[2105]*Saldo.[2105],10) AS decimal(18,10)),
		NULL
	FROM @Volatilidad Sqrt_Desvia
		INNER JOIN @Volatilidad Saldo ON Sqrt_Desvia.FECHA = Saldo.FECHA
	WHERE Sqrt_Desvia.[TIPO] = 'SQRT DESVIA.' AND Saldo.TIPO = 'SALDOS' AND
Saldo.FECHA = @fechaFin
```

```
	IF @imprimirTablas = 1
	BEGIN
		SELECT *
		FROM @Volatilidad
		WHERE TIPO = 'SALDOS'
		ORDER BY FECHA DESC;

		SELECT *
		FROM @Volatilidad
		WHERE TIPO = 'VARIACIONES'
		ORDER BY FECHA DESC;

		SELECT *
		FROM @Volatilidad
		WHERE TIPO = 'AVG VARIA.';

		SELECT *
		FROM @Volatilidad
		WHERE TIPO = 'DESVIACIONES'
		ORDER BY FECHA DESC;

		SELECT *
		FROM @Volatilidad
		WHERE TIPO = 'SUM DESVIA.';

		SELECT *
		FROM @Volatilidad
		WHERE TIPO = 'AVG DESVIA.';

		SELECT *
		FROM @Volatilidad
		WHERE TIPO = 'SQRT DESVIA.';

		SELECT *
		FROM @Volatilidad
		WHERE TIPO = 'VAR POR FF';
	END

	DECLARE @MatrizCorrelaciones TABLE(
		[FuenteFondeo] varchar(20),
		[2101-210135] float,
		[210135] float,
		[2103] float,
		[2104] float,
		[2105] float
	);

	DECLARE @General TABLE(
		[VaR_General] float,
		[VOLATILIDAD_GENERAL] float
	);

	DECLARE
		@a float, @b float, @c float, @d float, @e float, @f float, @g float, @h float, @i
float, @j float,
		@VaR_2101_210135 float, @VaR_210135 float, @VaR_2103 float, @VaR_2104
float, @VaR_2105 float;
	SELECT
		@VaR_2101_210135 = [2101-210135], @VaR_210135 = [210135], @VaR_2103
```

```
= [2103], @VaR_2104 = [2104], @VaR_2105 = [2105]
	FROM @Volatilidad
	WHERE [TIPO] = 'VAR POR FF';

	SELECT
		@a = IIF(SQRT(SUM(POWER(([2101-210135] - @AVG_2101_210135),2)) *
SUM(POWER(([210135] - @AVG_210135),2))) = 0, 0, ROUND(SUM(([2101-210135] -
@AVG_2101_210135) * ([210135] - @AVG_210135)) / SQRT(SUM(POWER(([2101-210135] -
@AVG_2101_210135),2)) * SUM(POWER(([210135] - @AVG_210135),2))),10)),
		@b = IIF(SQRT(SUM(POWER(([2101-210135] - @AVG_2101_210135),2)) *
SUM(POWER(([2103] - @AVG_2103),2))) = 0, 0, ROUND(SUM(([2101-210135] -
@AVG_2101_210135) * ([2103] - @AVG_2103)) / SQRT(SUM(POWER(([2101-210135] -
@AVG_2101_210135),2)) * SUM(POWER(([2103] - @AVG_2103),2))),10)),
		@c = IIF(SQRT(SUM(POWER(([2101-210135] - @AVG_2101_210135),2)) *
SUM(POWER(([2104] - @AVG_2104),2))) = 0, 0, ROUND(SUM(([2101-210135] -
@AVG_2101_210135) * ([2104] - @AVG_2104)) / SQRT(SUM(POWER(([2101-210135] -
@AVG_2101_210135),2)) * SUM(POWER(([2104] - @AVG_2104),2))),10)),
		@d = IIF(SQRT(SUM(POWER(([2101-210135] - @AVG_2101_210135),2)) *
SUM(POWER(([2105] - @AVG_2105),2))) = 0, 0, ROUND(SUM(([2101-210135] -
@AVG_2101_210135) * ([2105] - @AVG_2105)) / SQRT(SUM(POWER(([2101-210135] -
@AVG_2101_210135),2)) * SUM(POWER(([2105] - @AVG_2105),2))),10)),
		@e = IIF(SQRT(SUM(POWER(([210135] - @AVG_210135),2)) *
SUM(POWER(([2103] - @AVG_2103),2))) = 0, 0, ROUND(SUM(([210135] - @AVG_210135) *
([2103] - @AVG_2103)) / SQRT(SUM(POWER(([210135] - @AVG_210135),2)) *
SUM(POWER(([2103] - @AVG_2103),2))),10)),
		@f = IIF(SQRT(SUM(POWER(([210135] - @AVG_210135),2)) *
SUM(POWER(([2104] - @AVG_2104),2))) = 0, 0, ROUND(SUM(([210135] - @AVG_210135) *
([2104] - @AVG_2104)) / SQRT(SUM(POWER(([210135] - @AVG_210135),2)) *
SUM(POWER(([2104] - @AVG_2104),2))),10)),
		@g = IIF(SQRT(SUM(POWER(([210135] - @AVG_210135),2)) *
SUM(POWER(([2105] - @AVG_2105),2))) = 0, 0, ROUND(SUM(([210135] - @AVG_210135) *
([2105] - @AVG_2105)) / SQRT(SUM(POWER(([210135] - @AVG_210135),2)) *
SUM(POWER(([2105] - @AVG_2105),2))),10)),
		@h = IIF(SQRT(SUM(POWER(([2103] - @AVG_2103),2)) *
SUM(POWER(([2104] - @AVG_2104),2))) = 0, 0, ROUND(SUM(([2103] - @AVG_2103) *
([2104] - @AVG_2104)) / SQRT(SUM(POWER(([2103] - @AVG_2103),2)) *
SUM(POWER(([2104] - @AVG_2104),2))),10)),
		@i = IIF(SQRT(SUM(POWER(([2103] - @AVG_2103),2)) *
SUM(POWER(([2105] - @AVG_2105),2))) = 0, 0, ROUND(SUM(([2103] - @AVG_2103) *
([2105] - @AVG_2105)) / SQRT(SUM(POWER(([2103] - @AVG_2103),2)) *
SUM(POWER(([2105] - @AVG_2105),2))),10)),
		@j = IIF(SQRT(SUM(POWER(([2104] - @AVG_2104),2)) *
SUM(POWER(([2105] - @AVG_2105),2))) = 0, 0, ROUND(SUM(([2104] - @AVG_2104) *
([2105] - @AVG_2105)) / SQRT(SUM(POWER(([2104] - @AVG_2104),2)) *
SUM(POWER(([2105] - @AVG_2105),2))),10))
	FROM @Volatilidad
	WHERE [TIPO] = 'VARIACIONES';

	INSERT INTO @MatrizCorrelaciones
	VALUES('21001-210135', 1, @a, @b, @c, @d);
	INSERT INTO @MatrizCorrelaciones
	VALUES('210135', @a, 1, @e, @f, @g);
	INSERT INTO @MatrizCorrelaciones
	VALUES('2103', @b, @e, 1, @h, @i);
	INSERT INTO @MatrizCorrelaciones
	VALUES('2104', @c, @f, @h, 1, @j);
	INSERT INTO @MatrizCorrelaciones
	VALUES('2105', @d, @g, @i, @j, 1);
```

```
	DECLARE @Var_GENERAL float, @TOTAL_ULTIMO_DIA float;
	SELECT
		@Var_GENERAL = SQRT((@VaR_2101_210135*1
	+@VaR_210135*@a		+@VaR_2103*@b		+@VaR_2104*@c
	+@VaR_2105*@d) * @VaR_2101_210135 +
		(@VaR_2101_210135*@a	+@VaR_210135*1
	+@VaR_2103*@e		+@VaR_2104*@f	+@VaR_2105*@g)	*
@VaR_210135 +
		(@VaR_2101_210135*@b	+@VaR_210135*@e
	+@VaR_2103*1	+@VaR_2104*@h	+@VaR_2105*@i) * @VaR_2103 +
		(@VaR_2101_210135*@c	+@VaR_210135*@f
	+@VaR_2103*@h		+@VaR_2104*1 +@VaR_2105*@j) * @VaR_2104 +
		(@VaR_2101_210135*@d	+@VaR_210135*@g
	+@VaR_2103*@i		+@VaR_2104*@j	+@VaR_2105*1	)	*
@VaR_2105);

	SET @OUT_VaR_GENERAL = @Var_GENERAL;
	SELECT @TOTAL_ULTIMO_DIA = TOTAL FROM @Volatilidad WHERE FECHA =
@fechaFin AND TIPO = 'SALDOS';
	SET		@OUT_VOLATILIDAD_GENERAL		=
CAST(ROUND(@Var_GENERAL/@TOTAL_ULTIMO_DIA,4) AS decimal(18,4));

	INSERT INTO @General
	VALUES(@OUT_VaR_GENERAL, @OUT_VOLATILIDAD_GENERAL);

	IF @imprimirTablas = 1
	BEGIN
		SELECT *
		FROM @MatrizCorrelaciones;

		SELECT *
		FROM @General;
	END
END
```

En la table 34 se presenta el código del procedimiento almacenado `SP_URI_LIQ_MAYORES_DEPOSITANTES`. Este procedimiento realiza varias operaciones esenciales para identificar y almacenar información sobre los mayores depositantes basados en varias condiciones de cuentas financieras. Primero, elimina cualquier registro existente para la fecha actual. Luego, utiliza una tabla temporal `@MayoresDespositantes` para recopilar datos de cuentas activas e inactivas, cuentas con encaje mayor que cero, cuentas con saldo pendiente mayor que cero, depósitos a plazo fijo pendientes y ahorros programados pendientes, filtrando por ciertos criterios como tipo de cuenta y número de cuenta de persona.

Posteriormente, con una consulta utilizando la tabla temporal `@MayoresDespositantes`, agrupa los datos por identificación de persona, suma los saldos y selecciona los 100 mayores saldos en orden descendente. Luego, inserta estos resultados en la tabla `URI_LIQ_MAYOR_DEPOSITANTE`, junto con la fecha actual, filtrando específicamente para excluir ciertos códigos de cuenta de depósitos a plazo fijo y agrupando nuevamente por identificación para asegurar la integridad de los datos.

Tabla 34. SP_URI_LIQ_MAYORES_DEPOSITANTES

Fuente: Elaborado por autor

```
ALTER PROCEDURE [dbo].[SP_URI_LIQ_MAYORES_DEPOSITANTES]
AS
BEGIN
	DECLARE @hoy date;

	SET @hoy = GETDATE();

	--Elimina los registros existentes en la fecha actual
	--	En el caso de que se haya ejecutado el SP en la
	--	mañana y se vuelva a ejecutar en la tarde, debe
	--	guardar los valores de la última ejecución
	DELETE FROM URI_LIQ_MAYOR_DEPOSITANTE WHERE ULIMD_FECHA = @hoy;

	DECLARE @MayoresDespositantes AS TABLE (
		NumeroCuenta bigint,
		Identificacion varchar(15),
		Nombre varchar(150),
		Saldo money,
		CodigoCuenta varchar(20)
	);

	--Consulta las cuentas con estado: ACTIVA - INACTIVA
	INSERT INTO @MayoresDespositantes
	SELECT DCUCU.DCUCU_NUMERO, DCUPE.DCUPE_IDENTIFICACION,
DCUPE.DCUPE_NOMBRE, DCUCU.DCUCU_SALDO_DISPONIBLE, '250105'
	FROM DNE_CUE_CUENTA DCUCU
		INNER JOIN DNE_CUE_CUENTA_PERSONA DCUCP ON DCUCU.DCUCU_NUMERO
= DCUCP.DCUCU_NUMERO
		INNER JOIN DNE_CUE_PERSONA DCUPE ON DCUCP.DCUPE_NUMERO =
DCUPE.DCUPE_NUMERO
	WHERE
		(DCUCU.DCUEC_NUMERO = 1 OR DCUCU.DCUEC_NUMERO = 2)
		AND DCUCP.DCUVI_NUMERO = 1;

	--Consulta las cuentas con ENCAJE > 0
	INSERT INTO @MayoresDespositantes
	SELECT DCUCU.DCUCU_NUMERO, DCUPE.DCUPE_IDENTIFICACION,
DCUPE.DCUPE_NOMBRE, DCUCU.DCUCU_ENCAJE, '250155'
```

```
	FROM DNE_CUE_CUENTA DCUCU
		INNER JOIN DNE_CUE_CUENTA_PERSONA DCUCP ON DCUCU.DCUCU_NUMERO
= DCUCP.DCUCU_NUMERO
		INNER JOIN DNE_CUE_PERSONA DCUPE ON DCUCP.DCUPE_NUMERO =
DCUPE.DCUPE_NUMERO
	WHERE
		(DCUCU.DCUEC_NUMERO = 1 OR DCUCU.DCUEC_NUMERO = 2)
		AND DCUCP.DCUVI_NUMERO = 1
		AND DCUCU.DCUCU_ENCAJE > 0;

	--Consulta las cuentas con SALDO PENDIENTE > 0
	INSERT INTO @MayoresDespositantes
	SELECT DCUCU.DCUCU_NUMERO, DCUPE.DCUPE_IDENTIFICACION,
DCUPE.DCUPE_NOMBRE, DCUCU.DCUCU_SALDO_PENDIENTE, '250155'
	FROM DNE_CUE_CUENTA DCUCU
		INNER JOIN DNE_CUE_CUENTA_PERSONA DCUCP ON DCUCU.DCUCU_NUMERO
= DCUCP.DCUCU_NUMERO
		INNER JOIN DNE_CUE_PERSONA DCUPE ON DCUCP.DCUPE_NUMERO =
DCUPE.DCUPE_NUMERO
	WHERE
		(DCUCU.DCUEC_NUMERO = 1 OR DCUCU.DCUEC_NUMERO = 2)
		AND DCUCP.DCUVI_NUMERO = 1
		AND DCUCU.DCUCU_SALDO_PENDIENTE > 0;

	--Consulta los depositos a plazo fijo con estado: PENDIENTE
	INSERT INTO @MayoresDespositantes
	SELECT DCUCU.DCUCU_NUMERO, DCUPE.DCUPE_IDENTIFICACION,
DCUPE.DCUPE_NOMBRE, DINDP.DINDP_MONTO, DINDP.DINDP_CUENTA_MAYOR
	FROM DNE_INV_DEPOSITO_PLAZO_FIJO DINDP
		INNER JOIN DNE_CUE_CUENTA DCUCU ON DINDP.DCUCU_NUMERO =
DCUCU.DCUCU_NUMERO
		INNER JOIN DNE_CUE_CUENTA_PERSONA DCUCP ON DCUCU.DCUCU_NUMERO
= DCUCP.DCUCU_NUMERO
		INNER JOIN DNE_CUE_PERSONA DCUPE ON DCUCP.DCUPE_NUMERO =
DCUPE.DCUPE_NUMERO
	WHERE
		(DCUCU.DCUEC_NUMERO = 1 OR DCUCU.DCUEC_NUMERO = 2)
		AND DCUCP.DCUVI_NUMERO = 1
		AND DINDP.DINEP_NUMERO = 1;

	--Consulta los ahorros programados con estado: PENDIENTE
	INSERT INTO @MayoresDespositantes
	SELECT DCUCU.DCUCU_NUMERO, DCUPE.DCUPE_IDENTIFICACION,
DCUPE.DCUPE_NOMBRE, DINAP.DINAP_SALDO, DINAP.DINAP_CUENTA_MAYOR
	FROM DNE_INV_AHORRO_PROGRAMADO DINAP
		INNER JOIN DNE_CUE_CUENTA DCUCU ON DINAP.DCUCU_NUMERO =
DCUCU.DCUCU_NUMERO
		INNER JOIN DNE_CUE_CUENTA_PERSONA DCUCP ON DCUCU.DCUCU_NUMERO
= DCUCP.DCUCU_NUMERO
		INNER JOIN DNE_CUE_PERSONA DCUPE ON DCUCP.DCUPE_NUMERO =
DCUPE.DCUPE_NUMERO
	WHERE
		(DCUCU.DCUEC_NUMERO = 1 OR DCUCU.DCUEC_NUMERO = 2)
		AND DCUCP.DCUVI_NUMERO = 1
		AND DINAP.DINEA_NUMERO = 1;

	WITH Mayores AS (
		--Agrupa por numero de identificacion y suma el saldo
```

```
            --Ordena por la sumatoria del saldo desde el mayor al menor
            --Obtiene los 100 primeros registros según el ordenmaiento
            SELECT TOP 100 Identificacion, SUM(Saldo) SaldoFinal
            FROM @MayoresDespositantes
            GROUP BY Identificacion
            ORDER BY SaldoFinal DESC
      )
      --Identifica a los 100 mayores depositantes de la consulta inicial
      --Elimina los depositos a plazo fijo mayores a 180 dias
      --Agrupa por numero de identificacion y suma el saldo
      --Inserta los registros en la tabla Mayores Depositantes
      INSERT INTO URI_LIQ_MAYOR_DEPOSITANTE
      SELECT @hoy Fecha,MayoresTodos.Identificacion Identificacion,
SUM(Saldo) Saldo
      FROM @MayoresDespositantes MayoresTodos
            INNER JOIN Mayores ON MayoresTodos.Identificacion =
Mayores.Identificacion
      WHERE CodigoCuenta != '210315'
            AND CodigoCuenta != '210320'
            AND CodigoCuenta != '210325'
      GROUP BY MayoresTodos.Identificacion
      ORDER BY Saldo DESC;
END
```

3.4 Pruebas

3.4.1 **Pruebas de Funcionalidad**

Se realizaron pruebas para verificar que cada funcionalidad del sistema opere de acuerdo con las especificaciones. A continuación, en la tabla 35 se presentan las pruebas de funcionalidad.

Tabla 35. Pruebas de Funcionalidad.

Fuente: Elaborado por autor

Casos de Uso	Pantalla	Datos de Entrada	Resultado	Cumplimiento
Inicio de Sesión	Pantalla [43]	Usuario Contraseña	Acceso a la Pantalla Principal del sistema	Si
Cambio de Contraseña	Pantalla [45]	Contraseña actual Nueva contraseña Confirmar contraseña	Cambio de contraseña con éxito	Si
Buscar Usuario	Pantalla [46], Pantalla [48]	Código empleado Apellidos y Nombres	Lista de usuarios que coinciden según el criterio de búsqueda	Si
Crear Usuario	Pantalla [47]	Departamento Empleado Rol	Usuario creado con la información correspondiente	Si

		Agencia Dirección IP Administrador		
Modificar usuario	Pantalla [49]	Dirección IP Administrador Estado Cerrar Sesión	Modificación del campo seleccionado del usuario	Si
Resetear Usuario	Pantalla [50]		Usuario reseteado con una contraseña por defecto	Si
Crear Rol	Pantalla [51]	Nombre de rol	Rol creado con la información correspondiente	Si
Modificar Rol	Pantalla [51]	Nombre de rol	Modificación del campo seleccionado del rol	Si
Eliminar Rol	Pantalla [51]		Rol eliminado con éxito	Si
Crear Menú Sistema	Pantalla [52]	Tipo Nombre Formulario Atajos Orden	Menú sistema creado con la información correspondiente	Si
Modificar Menú Sistema	Pantalla [52]	Tipo Nombre Formulario Atajos Orden	Modificación del campo seleccionado del menú	Si
Eliminar Menú Sistema	Pantalla [52]		Menú eliminado con éxito	Si
Gestionar acceso a acciones de Menú por Rol de usuario	Pantalla [53]	Menú disponible Nombre rol Habilitado	Acceso al menú según el rol de usuario asignado	Si
Crear Empleado	Pantalla [54]	Tipo de identificación N° de identificación Apellidos y Nombres País de Nacimiento Identificación Étnica Sexo Estado Civil Fecha Nacimiento Numero Cargas Dis. Sociedad Conyugal Nominación Provincia Canto Parroquia Barrio Dirección Referencia Teléfono Celular Email Personal Email Institucional	Empleado creado con la información correspondiente	Si

		Tipo Discapacidad Tipo Contrato Relación Cargo Laboral Código Biométrico Estado Trabajador Fecha Ingreso Vinculado Oficina Pago Nómina Número Cuenta Pago Sueldo Plan Móvil Formación Académica		
Modificar Empleado	Pantalla [54]	Tipo de identificación N° de identificación Apellidos y Nombres País de Nacimiento Identificación Étnica Sexo Estado Civil Fecha Nacimiento Numero Cargas Dis. Sociedad Conyugal Nominación Provincia Canto Parroquia Barrio Dirección Referencia Teléfono Celular Email Personal Email Institucional Tipo Discapacidad Tipo Contrato Relación Cargo Laboral Código Biométrico Estado Trabajador Fecha Ingreso Vinculado Oficina Pago Nómina Número Cuenta Pago Sueldo Plan Móvil Formación Académica	Modificación de los campos seleccionados del empleado	Si
Visualizar Nomina	Pantalla [55]		Lista de empleados con puntos registrados	Si

Empleados			en la creación del empleado	
Crear Feriado	Pantalla [56], Pantalla [57]	Selección de la fecha Descripción	Feriado creado con los datos correspondientes	Si
Modificar Feriado	Pantalla [56], Pantalla [57]	Descripción	Modificación de la descripción del feriado	Si
Eliminar Feriado	Pantalla [57]		Feriado eliminado con éxito	Si
Visualizar Reporte de Liquidez Estructural	Pantalla [58]	Seleccionar fecha	Listado de los datos y cálculos correspondientes a la Liquidez Estructural	Si
Visualizar Reporte de Volatilidad	Pantalla [59], Pantalla [60], Pantalla [61], Pantalla [62]	Seleccionar fecha	Listado de los datos y cálculos correspondientes a la Volatilidad	Si
Visualizar Reporte de Mayores Depositantes	Pantalla [63]	Seleccionar fecha	Listado de los datos y cálculos correspondientes a los Mayores Depositantes	Si
Visualizar Reporte de Indicadores	Pantalla [64]	Seleccionar fecha	Listado de los datos y cálculos correspondientes a los Indicadores	Si
Visualizar Reporte de Balances Diarios	Pantalla [65]	Seleccionar fecha	Listado de los datos y cálculos correspondientes a los Balances Diarios	Si
Cerrar Sesión			Sesión finalizada con éxito	Si

3.4.2 Pruebas de Usabilidad

Se realizaron pruebas utilizando el Sistema de Usabilidad de Software (SUS) para evaluar la usabilidad del sistema. Este sistema consiste en un cuestionario de 10 preguntas después de haber usado el sistema. Las preguntas son:

1. Creo que me gustaría utilizar este sistema con frecuencia
2. Encontré el sistema innecesariamente complejo
3. Pensé que el sistema era fácil de usar
4. Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema
5. Encontré que las diversas funciones de este sistema estaban bien integradas
6. Pensé que había demasiada inconsistencia en este sistema
7. Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente
8. Encontré el sistema muy complicado de usar
9. Me sentí muy seguro usando el sistema
10. Necesitaba aprender muchas cosas antes de empezar con este sistema

Las respuestas de cada enunciado siguen el método de la Escala de Likert, como se muestra a continuación:

1. Totalmente en desacuerdo
2. En desacuerdo
3. Neutro
4. De acuerdo
5. Totalmente de acuerdo

En la tabla 36 se presentan las pruebas de usabilidad realizadas utilizando el SUS.

Tabla 36. Pruebas utilizando el Sistema de Usabilidad de Software.

Fuente. Elaborador por autor

N°	Preguntas	Administrador de Riesgos	Analista de Riesgos	Jefe de Tecnologías	Analista de Tecnologías
1	Creo que me gustaría utilizar este sistema con frecuencia	4	5	4	4
2	Encontré el sistema innecesariamente complejo	2	3	2	2
3	Pensé que el sistema era fácil de usar	2	1	3	3
4	Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema	2	3	2	1
5	Encontré que las diversas funciones de este sistema estaban bien integradas	4	4	4	5
6	Pensé que había demasiada inconsistencia en este sistema	2	1	1	1
7	Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente	5	4	5	5
8	Encontré el sistema muy complicado de usar	1	2	1	1
9	Me sentí muy seguro usando el sistema	4	5	4	5
10	Necesitaba aprender muchas cosas antes de empezar con este sistema	1	1	1	1

En la tabla 37 se presentan los cálculos del puntaje SUS para cada usuario.

Tabla 37. Cálculos del puntaje SUS.

Fuente: Elaborado por autor

Usuarios	Cálculos del Puntaje SUS
Administrador de Riesgos	Ítems positivos: (4-1) + (2-1) + (4-1) + (5-1) + (4-1) = 14 Ítems negativos: (5-2) + (5-2) + (5-2) + (5-1) + (5-1) = 17 Total: 14 + 17 = 31 SUS Score: 31 * 2.5 = 77.5
Analista de Riesgos	Ítems positivos: (5-1) + (1-1) + (4-1) + (4-1) + (5-1) = 14 Ítems negativos: (5-3) + (5-3) + (5-1) + (5-2) + (5-1) = 15 Total: 14 + 15 = 29 SUS Score: 29 * 2.5 = 72.5
Jefe de Tecnologías	Ítems positivos: (4-1) + (3-1) + (4-1) + (5-1) + (4-1) = 15 Ítems negativos: (5-2) + (5-2) + (5-1) + (5-1) + (5-1) = 18 Total: 15 + 18 = 33

	SUS Score: 33 * 2.5 = 82.5
Analista de Tecnologías	Ítems positivos: (4-1) + (3-1) + (5-1) + (5-1) + (5-1) = 17 Ítems negativos: (5-2) + (5-1) + (5-1) + (5-1) + (5-1) = 19 Total: 17 + 19 = 36 SUS Score: 36 * 2.5 = 90
Promedio del Puntaje SUS: (77.5 + 72.5 + 82.5 + 90) / 4 = 80.62	

Como resultado de las pruebas de usabilidad, el Sistema de Usabilidad del Software arrojó una puntuación de 80.62, lo cual indica que el sistema tiene una buena usabilidad. Este puntaje indica que los usuarios encuentran el sistema intuitivo y fácil de usar, cumpliendo así con los estándares esperados para la interacción y satisfacción del usuario.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Los objetivos del proyecto se cumplieron de manera satisfactoria. Se identificaron las herramientas necesarias para el desarrollo del sistema por medio de tablas comparativas que destacaron las características esenciales para el desarrollo. Además, se tomó en cuenta la petición de la empresa respecto a las herramientas establecidas para el desarrollo del sistema. Como resultado, se decidió utilizar SQL Server para la base de datos, el framework .NET, el IDE Visual Studio y el lenguaje de programación C#. Por otro lado, gracias al análisis de las tablas comparativas se eligió la arquitectura cliente-servidor y la metodología XP.

El desarrollo del sistema para la automatización de los procesos de gestión de riesgo de liquidez estructural se realizó utilizando las herramientas previamente establecidas por la Cooperativa de Ahorro y Crédito “San Martín” de Tisaleo Ltda. Se recopiló información detallada para identificar los requerimientos funcionales y no funcionales del usuario. Con base a esta información, se diseñaron y desarrollaron funcionalidades e interfaces necesarias del sistema. Priorizando su seguridad y usabilidad. Como resultado, se logró automatizar la generación de reportes necesarios para la gestión de riesgo de liquidez estructural. Además, el sistema incluye funcionalidades para la gestión de usuarios, empleados, roles, menús y feriados. Por otro lado, los controles de acceso garantizan que solo los usuarios autorizados puedan acceder a información sensible, asegurando así la integridad y confidencialidad de los datos gestionados.

Finalmente, para la validación del sistema se llevaron a cabo pruebas de funcionalidad y usabilidad. En las pruebas de funcionalidad, se verifico cada una de las pantallas del sistema, ingresando datos de entrada y confirmando que los resultados obtenidos fueran los esperados. Para las pruebas de usabilidad, se utilizó el Sistema de Usabilidad del Software (SUS). Se realizaron encuestas a las cuatro personas implicadas: administrador de riesgos, analista de riesgos, jefe de tecnologías y analista de tecnología. El resultado fue una puntuación de 80.62, lo que

evidencia que los usuarios encuentran el sistema intuitivo y fácil de usar. Los encuestados destacaron la facilidad de navegación por las distintas funciones y la claridad de la interfaz, lo que demostró que la experiencia del usuario fue positiva.

4.2 Recomendaciones

- Dentro de la Cooperativa de Ahorro y Crédito “San Martín” de Tisaleo Ltda., se deben realizar las respectivas capacitaciones continuas para el uso del sistema. Esto garantizará que todos los usuarios puedan utilizar eficientemente las nuevas herramientas.
- Realizar evaluaciones periódicas del impacto de la automatización en la gestión del riesgo de liquidez. Esto ayudará a cuantificar los beneficios y justificar futuras inversiones en tecnología y automatización.
- Considerar realiza un manual que documente todos los aspectos del sistema, desde la instalación hasta el funcionamiento diario y la solución de problemas.

REFERENCIAS BIBLIOGRÁFICAS

[1] V. R. dos S. Cardoso, L. A. Campos, J. A. Dantas, y O. R. de Medeiros, «Factors associated with the structural liquidity of banks in Brazil», *Rev. Contab. Financ.*, vol. 30, pp. 252-267, feb. 2019, doi: 10.1590/1808-057x201806350.

[2] «Liquidity Risk Management», en *Managing Risks in Commercial and Retail Banking*, John Wiley & Sons, Ltd, 2012, pp. 293-326. doi: 10.1002/9781119199250.ch17.

[3] R. Ruozi y P. Ferrari, «Liquidity Risk Management in Banks: Economic and Regulatory Issues», en *Liquidity Risk Management in Banks: Economic and Regulatory Issues*, R. Ruozi y P. Ferrari, Eds., en SpringerBriefs in Finance. , Berlin, Heidelberg: Springer, 2013, pp. 1-54. doi: 10.1007/978-3-642-29581-2_1.

[4] «49Revista Uruguaya de Historia Económica - Año X - No. 17 / julio de 2020Ecuador: La economía política del feriado bancario (8-12 DE MARZO DE 1999)», *Rev. Urug. Hist. Econ.*, vol. 17, n.º 17, oct. 2020, doi: 10.47003/RUHE/10.17.03.

[5] X. Sánchez Mayorga y J. C. Millán Solarte, «Medición del riesgo de liquidez. Una aplicación en el sector cooperativo», *Entramado*, vol. 8, n.º 1, pp. 90-98, ene. 2012.

[6] S. J. López Carranco, «Desarrollo de un sistema web para la gestión y monitoreo de los procesos judiciales y de recursos humanos, con informes estadísticos de créditos, agencias y desempeño social para la cooperativa de ahorro y crédito mujeres unidas mediante la utilización del Framework Ruby On Rails y la Librería Reactjs», bachelorThesis, 2018. Accedido: 10 de enero de 2024. [En línea]. Disponible en: http://repositorio.utn.edu.ec/handle/123456789/8643

[7] E. A. Muñoz Gutierrez, «Sistema Web-Móvil para la Gestión de Ahorros y Préstamos en la Cooperativa de Ahorro y Crédito Livitaca, Cusco 2023», *Repos. Inst. - UCV*, 2023, Accedido: 10 de enero de 2024. [En línea]. Disponible en: https://repositorio.ucv.edu.pe/handle/20.500.12692/118395

[8] H. A. Mamani Herrera, «Desarrollo de Aplicación Web para Reporte de Clientes Morosos de la Cooperativa de Ahorro y Crédito Credifinca, Distrito de Chivay, 2020», *Univ. Priv. Telesup - UTELESUP*, 2020, Accedido: 10 de enero de 2024. [En línea]. Disponible en: https://repositorio.utelesup.edu.pe/handle/UTELESUP/1910

[9] E. J. Nata Ichina, «Implementación del módulo de facturación electrónica adaptable al sistema morvisoft para la Cooperativa Ahorro y Crédito San Martín.», bachelorThesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería en Sistemas Computacionales e Informáticos, 2018. Accedido: 10 de enero de 2024. [En línea]. Disponible en: https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/28571

[10] J. A. Caizabanda Jerez, «Aplicación Web-Móvil para la gestión de consultas financieras de la Cooperativa de Ahorro y Crédito" Kullki Wasi"», B.S. thesis, 2018. Accedido: 10 de enero de 2024. [En línea]. Disponible en: https://dspace.uniandes.edu.ec/handle/123456789/8159

[11] P. O. Mamarandi Toapanta, «Desarrollo e implementación de una aplicación web responsivo, que facilite la gestión de análisis, dentro del proceso de obtención de un crédito, en la Cooperativa de Ahorro y Crédito “Credil” Ltda.», bachelorThesis, Escuela Superior Politécnica de Chimborazo, 2018. Accedido: 10 de enero de 2024. [En línea]. Disponible en: http://dspace.espoch.edu.ec/handle/123456789/9273

[12] L. R. Mendoza Vera, «Desarrollo e implementación de una aplicación web y móvil para la consulta de saldos y movimientos de las cuentas de los clientes de la cooperativa de ahorro y crédito Uniblock y servicios ltda», bachelorThesis, Ecuador: Latacunga: Universidad Técnica de Cotopaxi (UTC)., 2019. Accedido: 10 de enero de 2024. [En línea]. Disponible en: http://localhost/handle/27000/5324

[13] W. R. Tisalema Tisalema, «Aplicación móvil para realizar transferencias directas e interbancarias en la Cooperativa de Ahorro y Crédito San Martín de Tisaleo Ltda», bachelorThesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería en Sistemas Computacionales e Informática, 2020. Accedido: 10 de enero de 2024.

[En línea]. Disponible en: https://repositorio.uta.edu.ec:8443/jspui/handle/123456789/31310

[14] M. de la C. Narváez Argeñal, «Desarrollo de un software para la gestión de préstamos (mayo-septiembre 2019)».

[15] Y. M. Baque García, «IMPLEMENTACIÓN DE UN APLICATIVO WEB PARA LA GESTIÓN INTERNA DE LA COOPERATIVA DE AHORRO Y CRÉDITO "POR EL PAN Y EL AGUA"», bachelorThesis, Jipijapa-Unesum, 2023. Accedido: 10 de enero de 2024. [En línea]. Disponible en: http://repositorio.unesum.edu.ec/handle/53000/4745

[16] J. P. Arévalo Cuenca, «Implementación de un sistema de monitoreo de límites de exposición de riesgos de liquidez en una entidad financiera mediana del Ecuador», masterThesis, 2018. Accedido: 10 de enero de 2024. [En línea]. Disponible en: http://dspace.utpl.edu.ec/jspui/handle/20.500.11962/22195

[17] G. I. Idrovo Armijos y P. M. Quizhpi Sinchi, «Estudio de los modelos y/o herramientas para la administración del riesgo de liquidez y establecimiento de mejoras en su administración, aplicado en las Cooperativas de Ahorro y Crédito del Segmento 2 de la Zona del Austro», masterThesis, Universidad del Azuay, 2020. Accedido: 10 de enero de 2024. [En línea]. Disponible en: http://dspace.uazuay.edu.ec/handle/datos/10040

ANEXOS

Anexo 1. Plantilla para la especificación de requisitos

Introducción

El presente documento es una Especificación de Requisitos de Software (ERS) para la Automatización de los procesos de la gestión de riesgo de liquidez estructural de la Cooperativa de Ahorro y Crédito "San Martín" de Tisaleo Ltda. Las especificaciones se han estructurado basándose en el estándar ANSI/IEEE 830.

Propósito

Este documento tiene como propósito determinar las especificaciones funcionales y no funcionales para el desarrollo del sistema de gestión de riesgo de liquidez estructural de la Cooperativa de Ahorro y Crédito "San Martín" de Tisaleo Ltda.

Personal Involucrado

Nombre	Mariela Amanta
Rol	Analista, diseñador y programador
Categoría Profesional	Software
Responsabilidad	Análisis de información, diseño y programación
Información del contacto	marielaamanta@gmail.com

Definiciones, acrónimos y abreviaturas

Nombre	**Descripción**
Usuario	Persona que usara el sistema
ERS	Especificación de Requisitos de Software
RF	Requerimientos Funcionales
RNF	Requerimientos No Funcionales
SEPS	Superintendencia de Economía Popular y Solidaria

Referencias

Título del Documento	Referencia
Estándar IEEE 830	IEEE

Resumen

El presente documento comienza con una introducción a la especificación de requisitos de software, seguida de una descripción general del sistema para comprender sus principales funciones. Finalmente, se detallan los requerimientos funcionales y no funcionales.

Descripción General

Perspectiva del Producto

El sistema de los procesos de la gestión de riesgo de liquidez estructural será un producto de escritorio, lo que permitirá un acceso rápido y eficaz, además se integrará con el sistema de la Cooperativa.

Características de los Usuarios

Tipo de usuario	Administrador de Riesgos
Formación	Licenciada Contable
Actividades	Procesos en la liquidez estructural

Tipo de usuario	Analista de Riesgos
Formación	Licenciada Contable
Actividades	Procesos en la liquidez estructural

Restricciones

- El sistema debe contar con un diseño e implementación simples.
- Debe ser un sistema de escritorio.
- Las herramientas para el sistema deben ser las que especifique la empresa.

Suposiciones y Dependencias

- Se asume que los requisitos descritos en el documento son estables
- Los equipos donde se ejecute el sistema deberán cumplir con requisitos que garanticen la ejecución correcta del sistema.

Anexo 2. Respuestas a las preguntas de la entrevista técnica

1. ¿Quiénes son los usuarios principales que usaran el sistema?

   Bueno las personas que usaremos el sistema es la ingeniera Gloria que es Administradora de Riesgos y mi persona que soy Analista de Riesgos, pero en este caso más lo uso yo, como Analista me encargo de todo ese proceso de la liquidez estructural

2. ¿De dónde se obtiene la información para el proceso de la gestión de riesgo de liquidez estructural?

   Para obtener la información para el proceso tengo que ingresar al servidor de archivos interno de la cooperativa, después me voy a la carpeta de Riesgos donde se encuentra el archivo que sube el ingeniero Wilson con la información que necesitamos.

3. ¿Cuál es el proceso que se realiza después de obtener los datos para la gestión de riesgo de liquidez estructural?

   Una vez que se haya obtenido la información de la carpeta del servidor de archivos se procede a descargar este archivo, se copia toda la información, abrimos una hoja de Excel nueva y pegamos todos los datos, después borramos la información que no necesitemos y dejamos solo lo que vamos a utilizar.

   Después vamos nuevamente al servidor de archivos a la ruta Riesgos/Administración de Riesgos/Liquidez/2024/Liquidez Estructural Mensual/03.LIQUIDEZ_MARZO_TISALEO_2024.xlsm, este es uno de los archivos que nos da la SEPS, este Excel tiene cinco diferentes hojas, en la primera vamos a pegar la información qué copiamos de la hoja de Excel en

blanco, la información la vamos a pegar según el día que estemos realizando ese proceso, por ejemplo si estamos realizando el miércoles sería 20 si vamos a realizar el jueves sería 21 entonces le pegamos justo bajo la fecha que nos toca, una vez que ya pegamos la información debajo de la fecha que corresponde en la primera hoja nos vamos a la segunda hoja y aquí automáticamente los datos se obtienen de la primera hoja que ya pegamos, automáticamente se van realizando los cálculos los cuales nos van a dar los resultados de la liquidez de primera línea y la liquidez de segunda línea, ya que tengamos los resultados de la liquidez de primera línea y los resultados de la liquidez de segunda línea se va a tornar en un color entre verde, amarillo o rojo que son los indicadores o límites establecidos, en este caso se puso verde que sería un límite bajo, en caso de ponerse amarillo sería medio y en caso de ponerse rojo sería alto, esto nos ayuda a saber el nivel de riesgo en el que nos encontramos.

La tercera hoja del Excel tenemos los cálculos para la volatilidad en el cual se realiza una serie de pasos que ya le voy a dar una copia para que sepa cómo es ese proceso, mientras tanto aquí de igual forma se realiza un proceso manual para obtener los resultados de la volatilidad, una vez que obtengamos los resultados de la volatilidad vemos a copiarlos, los resultados están en la parte de abajo, copiamos los resultados y los pegamos en la segunda hoja en la parte donde nos pide la volatilidad.

Como paso final de la liquidez estructural tenemos que sacar el 50% de los 100 mayores depositantes para lo cual vamos a sacar la información del servidor de archivos interno de la cooperativa en la misma carpeta de riesgos vamos a encontrar los datos para los mayores depositantes, en este caso el documento nos proporciona los datos que vamos a ir copiando a continuación: los depósitos restringidos, depósitos a la vista y los depósitos programados, todos estos depósitos los copiamos y los pegamos en una hoja de Excel y después vamos a abrir una hoja dinámica dónde vamos a colocar los filtros y así obtendremos los mayores depositantes después escogeremos los 100 primeros los ordenaremos de mayor a menor y sacaremos el total, de ese total vamos a sacar el 50% que es el valor que vamos a pegar en la

segunda hoja donde habíamos pegado la volatilidad, este cálculo del 50% de los mayores depositantes se va a sacar una sola vez al mes y ese resultado lo vamos a ir pegando en todos los días del mes ya que los anteriores cálculos que realizamos de la liquidez estructural de primera línea, la liquidez estructural de segunda línea y la volatilidad sí se realizan diariamente.

Cuando ya tengamos la segunda hoja completa con los cuatro valores que representa la liquidez estructural vamos a abrir un segundo documento de Excel que nos proporciona las SEPS el cuál es de redondeo, aquí toda la información de la segunda hoja del primer Excel lo vamos a copiar y lo vamos a pegar, este documento nos ayuda netamente a redondear los valores para que el informe final este con los redondeos que quiere la SEPS.

Finalmente, La SEP nos proporciona un tercer documento de Excel que es del formato final en el que debemos plasmar los datos que obtuvimos del redondeo del segundo documento de Excel, aquí solo vamos a traer los datos y automáticamente se pone en el formato que dispone la SEPS. Una vez que ya tengamos el formato final ingresamos a la página y procedemos a subir el documento.

4. ¿Cuáles son los cálculos que se realizan dentro de la liquidez estructural?

   Dentro de la liquidez estructural como ya observamos vamos a obtener cálculos como: la liquidez de estructural de primera línea, liquidez estructural de segunda línea, la volatilidad y el 50% de los mayores depositantes.

5. ¿De dónde sacan las fórmulas o estructuras que siguen para los procesos?

   La superintendencia de economía popular y solidaria es quien nos facilita los documentos con las fórmulas que debemos seguir para el proceso de la liquidez estructural

6. ¿Cuáles son los documentos, plantillas, estructuras o fórmulas que establece la SEPS?

Los documentos que nos proporciona son tres: el documento de reporte, el documento de redondeo y el documento con el formato final para subir

7. ¿Cuáles son los datos, reportes o documentos que se obtiene después del proceso?

   El documento final que se obtiene muestra los resultados de la liquidez estructural de primera línea, la liquidez estructural de segunda línea, la volatilidad y el 50% de los mayores depositantes

8. ¿Hay otro documento que sigan para el proceso?

   Los tres documentos mencionados son los que se utilizan para el proceso de la liquidez estructural

9. ¿Con que sistemas debe integrarse el producto que se realizara?

   El producto que va a desarrollar debería integrarse con el Core Financiero de la Cooperativa

10. ¿Existen límites para los resultados finales del proceso?

    Sí, como ya se mencionó existen límites establecidos para los resultados de la liquidez estructural de primera línea y la liquidez estructural de segunda línea

11. ¿Cómo se establecen los límites y alertas para la gestión de riesgo de liquidez?

    Desde que subimos de segmento empezamos a tener los indicadores de liquidez o sea los límites que se establecen para la liquidez estructural de primera línea y segunda línea, los límites se establecen tomando en cuenta los porcentajes de los indicadores del año anterior, se realiza un promedio, sacamos la moda, la desviación estándar, los máximos y los mínimos, y según los resultados se establecen los límites de los indicadores para la liquidez estructural.