UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Tema:

APLICACIÓN MÓVIL USANDO EL FRAMEWORK IONIC PARA LA GESTIÓN DE PRESUPUESTOS DE VENTAS EN LA EMPRESA RECTIMA INDUSTRY CIA. LTDA.

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Tecnologías de la Información

ÁREA: Base de datos y Sistemas informáticos

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

AUTOR: Henry David Chimborazo Tubon

TUTOR: Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

Ambato - Ecuador

agosto – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: APLICACIÓN MÓVIL USANDO EL FRAMEWORK IONIC PARA LA GESTIÓN DE PRESUPUESTOS DE VENTAS EN LA EMPRESA RECTIMA INDUSTRY CIA. LTDA., desarrollado bajo la modalidad Proyecto de Investigación por el señor Henry David Chimborazo Tubon, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado 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:
FRANKLIN OSWALDO MAYORGA MAYORGA

Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: APLICACIÓN MÓVIL USANDO EL FRAMEWORK IONIC PARA LA GESTIÓN DE PRESUPUESTOS DE VENTAS EN LA EMPRESA RECTIMA INDUSTRY CIA. 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.

Henry David Chimborazo Tubon

C.C. 1805295258

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.

Henry David Chimborazo Tubon

C.C. 1805295258

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Henry David Chimborazo Tubon, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado APLICACIÓN MÓVIL USANDO EL FRAMEWORK IONIC PARA LA GESTIÓN DE PRESUPUESTOS DE VENTAS EN LA EMPRESA RECTIMA INDUSTRY CIA. 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:
DAVID OMAR GUEVARA AULESTIA

Ing. David Guevara Aulestia, Mg.
PROFESOR CALIFICADOR

Firmado electrónicamente por:
RUBEN EDUARDO NOGALES PORTERO

Ing. Rubén Nogales Portero, Mg.
PROFESOR CALIFICADOR

DEDICATORIA

Dedico el presente trabajo de titulación primeramente a Dios por darme vida, salud y sabiduría en todo el trascurso de mis estudios.

A mis padres y hermano que son mi gran inspiración y modelo a seguir, siempre me apoyaron en los momentos difíciles brindándome su compresión, amor, ayuda, apoyo y ánimo.

A mis tíos y abuelos que me apoyaron en todo el trascurso de mis estudios brindándome palabras de aliento y compañía.

A mis maestros que con sus enseñanzas y ayuda contribuyeron en toda mi formación académica.

Y finalmente a mis amigos y todas aquellas personas que de una u otra manera contribuyeron para el logro de mis objetivos.

Henry David Chimborazo Tubon

AGRADECIMIENTO

En primer lugar, agradezco a Dios por darme vida, salud y sabiduría para culminar exitosamente esta etapa de mi vida.

A mis padres Luis y Emma por ser mi ejemplo a seguir, mi fortaleza y el pilar para cumplir todas mis metas y mi hermano Diego por ser un gran hermano.

A mis tíos Edgar y Flor quienes me ayudaron, apoyaron y ha sido fundamentales durante todo mi periodo académico.

A mi tutor Ing. Franklin Mayorga por su apoyo, conocimiento, consejos y ayuda cuando más la necesitaba durante este trabajo de titulación.

A mi tutor empresarial Ing. Paul Cherres quien fue la persona que me ayudo y contribuyo con todo su conocimiento.

A la Familia Alvarado Arellano al permitirme y apoyarme en la realización de mi trabajo de titulación en la Empresa Rectima Industry Cia. Ltda.

A la Universidad Técnica de Ambato en especial a la Facultad de Ingeniería en Sistemas, Electrónica e Industrial por darme la apertura de poder haber cursado satisfactoriamente la carrera Tecnologías de la Información.

Henry David Chimborazo Tubon

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 3

1.3.1 Multiplataforma 3

1.3.2 Ingeniería de Software 4

1.3.3 Metodologías de Desarrollo 5

1.3.4 Desarrollo de Software 5

1.3.5 Aplicación Móvil 6

1.3.6 Tipos de aplicaciones móvil 7

1.3.7 Sistemas operativos móvil 8

1.3.8 Framework 9

1.3.9 Motores base de datos 11

1.3.10 Planificación Financiera 12

1.3.11 Asignación de Recursos 12

1.3.12 Toma de Decisiones 12

1.3.13 Presupuestos 13

1.4 Objetivos 14

1.4.1 Objetivo general 14

1.4.2 Objetivos específicos 14

CAPÍTULO II. METODOLOGÍA 15

2.1 Materiales 15

2.2 Métodos 19

2.2.1 Modalidad de la investigación 19

2.2.2 Población y muestra 19

2.2.3 Recolección de información 20

2.2.4 Procesamiento y análisis de datos 34

CAPÍTULO III. RESULTADOS Y DISCUSIÓN 39

3.1 Análisis y discusión de los resultados 39

3.1.1 Proceso de planificación de presupuesto en ventas 39

3.1.2 Framework Ionic para el desarrollo móvil 43

3.1.3 Marcos de trabajo usado para el desarrollo del proyecto de investigación 46

3.1.4 Framework Node.js para el desarrollo de Web Services 47

3.1.5 Entorno de desarrollo integrado para el desarrollo del proyecto 50

3.1.6 Gestor de base de datos 52

3.1.7 Metodologías de desarrollo móvil 53

3.2 Desarrollo de la propuesta 57

3.2.1 Fase 1: Exploración 57

3.2.2 Fase 2: Inicialización 62

3.2.3 Fase 3: Producción 105

3.2.4 Fase 4: Estabilización 123

3.2.5 Fase 5: Pruebas 130

3.2.6 Implantar aplicativo móvil 152

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES 161

4.1 Conclusiones 161

4.2 Recomendaciones 162

REFERENCIAS BIBLIOGRÁFICAS 163

ANEXOS 165

ÍNDICE DE TABLAS

Tabla 1. Instrumento elaborado para la entrevista 15

Tabla 2. Población..... 19

Tabla 3. Resultados guía de entrevista jefe de área informática 21

Tabla 4. Confiabilidad Alfa de Cronbach 24

Tabla 5. Confiabilidad Alfa de Cronbach 24

Tabla 6. Datos análisis encuesta..... 35

Tabla 7. Versiones Ionic 45

Tabla 8. Marcos de trabajo..... 46

Tabla 9. Versiones Node.js 49

Tabla 10. Entornos de desarrollo 51

Tabla 11. Metodologías de desarrollo..... 54

Tabla 12. Requerimientos funcionales..... 58

Tabla 13. Requerimientos no funcionales..... 60

Tabla 14. Planificación fases proyecto..... 64

Tabla 15. Presupuesto mensual..... 67

Tabla 16. Presupuesto anual..... 67

Tabla 17. Clientes..... 68

Tabla 18. Productos..... 68

Tabla 19. Promocional por mes 69

Tabla 20. Promocional por familia..... 70

Tabla 21. Promocional por asesor 70

Tabla 22. Datos nuevo cliente 71

Tabla 23. Documentos nuevo cliente 72

Tabla 24. Bonificación por cartera 73

Tabla 25. Presupuesto efectividad 74

Tabla 26. Usuarios Aplicación 74

Tabla 27. Provincias 75

Tabla 28. Procedimientos almacenados 106

Tabla 29. Vistas 111

Tabla 30. Diseño StoryCards 113

Tabla 31. StoryCard registro usuario 113

Tabla 32. StoryCard inicio sesión 114

Tabla 33. StoryCard cambio de contraseña 115

Tabla 34. StoryCard inicio 115

Tabla 35. StoryCard progreso actual 116

Tabla 36. StoryCard comisión por monto 117

Tabla 37. StoryCard comisión por efectividad 117

Tabla 38. StoryCard comisión por rentabilidad 118

Tabla 39. StoryCard bono por monto 119

Tabla 40. StoryCard bono por promocional 119

Tabla 41. StoryCard bono por cartera 120

Tabla 42. StoryCard total progreso 121

Tabla 43. StoryCard creación nuevo cliente 122

Tabla 44. StoryCard ventas clientes 123

Tabla 45. Plantilla para pruebas de aceptación 146

Tabla 46. Prueba aceptación 1 146

Tabla 47. Prueba aceptación 2 146

Tabla 48. Prueba aceptación 3 147

Tabla 49. Prueba aceptación 4 147

Tabla 50. Prueba aceptación 5 148

Tabla 51. Prueba aceptación 6 148

Tabla 52. Prueba aceptación 7 149

Tabla 53. Prueba aceptación 8 149

Tabla 54. Prueba aceptación 9 150

Tabla 55. Prueba aceptación 10 150

Tabla 56. Prueba aceptación 11 150

Tabla 57. Prueba aceptación 12 151

Tabla 58. Prueba aceptación 13 151

Tabla 59. Prueba aceptación 14 152

Tabla 60. Actividades capacitación 157

Tabla 61. Cumplimiento presupuesto antes del aplicativo móvil 158

Tabla 62. Cumplimiento presupuesto después del aplicativo móvil 158

ÍNDICE DE FIGURAS

Figura 1. Calculo Alfa de Cronbach 23

Figura 2. Pregunta 1 25

Figura 3. Pregunta 2 26

Figura 4. Pregunta 3 27

Figura 5. Pregunta 4 28

Figura 6. Pregunta 5 29

Figura 7. Pregunta 6 30

Figura 8. Pregunta 7 31

Figura 9. Pregunta 8 32

Figura 10. Pregunta 9 33

Figura 11. Pregunta 10 34

Figura 12. Estadística respuestas encuesta 36

Figura 13. Secuencia de pasos planificación presupuestos en ventas 38

Figura 14. Proceso planificación de presupuestos 40

Figura 15. Base de datos en SQL Server 66

Figura 16. Diagrama caso de uso de los asesores comerciales 76

Figura 17. Diagrama caso de uso administrador 76

Figura 18. Inicio de sesión 77

Figura 19. Registro Usuario 78

Figura 20. Envió mensaje cambio de contraseña 79

Figura 21. Ingreso y validación código 80

Figura 22. Ingreso nueva contraseña 81

Figura 23. Inicio 82

Figura 24. Menú navegación 83

Figura 25. Progreso diario 84

Figura 26. Comisión bono 85

Figura 27. Comisión efectividad 86

Figura 28. Comisión rentabilidad 87

Figura 29. Bono monto 88

Figura 30. Bono promocional 89

Figura 31. Bono cartera 90

Figura 32. Total progreso alcanzado 91

Figura 33. Bono monto 92

Figura 34. Bono monto 93

Figura 35. Ventas clientes asignados 94

Figura 36. Compras y devoluciones productos 95

Figura 37. Administrador inicio 96

Figura 38. Administrador progreso diario 97

Figura 39. Administrador comisión bono 98

Figura 40. Administrador comisión efectividad 99

Figura 41. Administrador comisión rentabilidad 100

Figura 42. Administrador bono monto 101

Figura 43. Administrador bono por promocional 102

Figura 44. Administrador bono cartera 103

Figura 45. Administrador total progreso 104

Figura 46. Administrador ventas clientes 105

Figura 47. Dependencias del backend 110

Figura 48. Dependencias del frontend 112

Figura 49. Cambio modelo interfaz inicio se sesión 124

Figura 50. Cambio modelo interfaz registro usuario 125

Figura 51. Cambio iconos menú 126

Figura 52. Cambio interfaz progreso actual 127

Figura 53. Cambio interfaz total progreso 128

Figura 54. Cambio interfaz creación cliente 129

Figura 55. Validación campos completos 131

Figura 56. Validación usuario y contraseña correcto 132

Figura 57. Validación ingreso código del asesor comercial 133

Figura 58. Validación existencia código asesor comercial 134

Figura 59. Validación ingreso contraseña 135

Figura 60. Validación usuario ya registrado 136

Figura 61. Validación ingreso código 137

Figura 62. Validación registro usuario 138

Figura 63. Validación ingreso código validación 139

Figura 64. Validación código verificación..... 140

Figura 65. Validación contraseñas iguales..... 141

Figura 66. Validación selección campos fecha modificado..... 142

Figura 67. Validación selección fecha actual..... 143

Figura 68. Validación ingreso datos personales nuevo cliente 144

Figura 69. Validación ingreso documentos nuevo cliente 145

Figura 70. Instalación aplicativo móvil..... 153

Figura 71. Análisis aplicativo móvil 154

Figura 72. Validación análisis aplicativo móvil..... 155

Figura 73. Terminación instalación aplicativo móvil 156

Figura 74. Aplicativo móvil instalado..... 156

Figura 75. Porcentaje cumplimientos ventas antes y después..... 159

Figura 76. Comparación media y desviación estándar..... 160

ÍNDICE DE ANEXOS

Anexo A. Código registro usuario 165

Anexo B. Código inicio sesión..... 168

Anexo C. Código cambio de contraseña 169

Anexo D. Código inicio 173

Anexo E. Código progreso actual 174

Anexo F. Código comisión por monto..... 177

Anexo G. Código comisión por efectividad..... 180

Anexo H. Código comisión por rentabilidad 183

Anexo I. Código bono por monto 185

Anexo J. Código bono por promocional 188

Anexo K. Código bono por cartera 191

Anexo L. Código total progreso..... 193

Anexo M. Código creación nuevo cliente..... 199

Anexo N. Código ventas clientes..... 206

Anexo O. Manual de usuario 211

RESUMEN EJECUTIVO

El presente proyecto de investigación se centra en crear una solución que modernice la gestión y seguimiento del cumplimiento de los presupuestos de ventas en la Empresa Rectima Industry Cia. Ltda. Para lo cual, se implementó un aplicativo móvil utilizando el Framework Ionic que permite a los distintos asesores comerciales acceder a información detallada sobre presupuestos, bonos, comisiones, número de ventas realizadas por cada producto y la creación de nuevos clientes. Se inició con un análisis detallado de los procesos de planificación de presupuestos de ventas, identificando el proceso seguido desde la preparación del pronóstico de ventas hasta el monitoreo y evaluación del presupuesto. Este análisis permitió tener una visión clara y precisa del proceso seguido para la planificación de presupuestos en ventas, lo cual fue fundamental para el desarrollo correcto y preciso del aplicativo móvil. Se utilizó el Framework Ionic para la implementación del aplicativo móvil, aprovechando su capacidad para desarrollar aplicaciones móviles eficientes y crear interfaces de usuario amigables. Además, se empleó Mobile-D, una metodología ágil diseñada específicamente para el desarrollo de aplicativos móviles, lo que facilitó la interacción rápida y la entrega continua del aplicativo desarrollado.

El uso de una metodología ágil combinada con un Framework como Ionic permitió un desarrollo rápido y económico, ofreciendo una solución robusta que cumple con las necesidades específicas de la Empresa Rectima Industry Cia. Ltda. El aplicativo móvil resultante dinamizó el proceso de gestión de los presupuestos de ventas, garantizando una mayor eficacia operativa y mejorando la capacidad de respuesta del equipo de ventas. Este enfoque tecnológico ha proporcionado nuevas formas de acceder a información actualizada mejorando la eficacia operativa y la capacidad de respuesta del equipo de ventas mediante la adopción de aplicaciones móviles desarrolladas con tecnologías modernas y metodologías ágiles.

Palabras clave: Presupuesto, venta, ionic, aplicación móvil, mobile-d

ABSTRACT

The present research project focuses on creating a solution to modernize the management and tracking of sales budget compliance at Rectima Industry Cia. Ltda. Company. To achieve this, a mobile application was implemented using the Ionic Framework, allowing various sales advisors to access detailed information on budgets, bonuses, commissions, the number of sales made per product, and the creation of new clients. The project began with a detailed analysis of the sales budget planning processes, identifying the process followed from the preparation of the sales forecast to the monitoring and evaluation of the budget. This analysis provided a clear and precise vision of the process followed for sales budget planning, which was fundamental for the correct and accurate development of the mobile application. The Ionic Framework was used for the implementation of the mobile application, leveraging its capability to develop efficient mobile applications and create user-friendly interfaces. Additionally, Mobile-D, an agile methodology specifically designed for mobile application development, was employed, facilitating rapid interaction and continuous delivery of the developed application.

The use of an agile methodology combined with a Framework like Ionic allowed for rapid and cost-effective development, offering a robust solution that meets the specific needs of Rectima Industry Cia. Ltda. Company. The resulting mobile application streamlined the sales budget management process, ensuring greater operational efficiency and improving the responsiveness of the sales team. This technological approach has provided new ways to access updated information, enhancing the operational efficiency and responsiveness of the sales team through the adoption of mobile applications developed with modern technologies and agile methodologies.

Keywords: Budget, sale, ionic, mobile application, mobile-d

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

APLICACIÓN MÓVIL USANDO EL FRAMEWORK IONIC PARA LA GESTIÓN DE PRESUPUESTOS DE VENTAS EN LA EMPRESA RECTIMA INDUSTRY CIA. LTDA.

1.1.1 Planteamiento del problema

La economía global está experimentando un declive gradual, con una inflación que se mantiene baja y un crecimiento que se mantiene estable. A pesar de ello, el ritmo de crecimiento sigue siendo moderado, permitiendo que la actividad económica a nivel mundial se mantenga resistente gracias al respaldo de los factores tanto de oferta como de demanda en las principales economías [1]. La economía digital está desempeñando un papel fundamental en las actividades económicas al impulsar la producción. Su influencia se traduce en la disminución de costos mediante la implementación de tecnologías innovadoras en los procesos empresariales, lo que fomenta la innovación en diversos sectores de la economía [2]. Los mercados altamente competitivos donde las condiciones económicas y comerciales están en constante cambio denotan la importancia de tener una gestión eficiente de presupuestos, convirtiéndose en una prioridad para las empresas debido a la creciente competencia y la volatilidad del mercado.

A nivel de Ecuador, las empresas se enfrentan a dinámicas específicas del mercado local que impactan en la gestión de presupuestos. Factores como la estacionalidad de la demanda, la regulación gubernamental, la disponibilidad de recursos y la situación económica del país pueden influir en la capacidad de las empresas para alcanzar sus metas en ventas. En este contexto, las empresas ecuatorianas deben implementar estrategias adaptadas a las condiciones del mercado nacional, para garantizar la gestión efectiva de sus presupuestos, el uso de sistemas eficaces se ha vuelto esencial en el contexto de las empresas nacionales.

En la Empresa Rectima Industry Cia. Ltda. de la ciudad de Ambato la ineficiente comunicación sobre el cumplimiento de ventas ha sido un problema que se presenta de forma frecuente y ha generado disminución en la productividad de los asesores comerciales encargados de realizar las ventas, perdida en oportunidades de ventas, insatisfacción de los clientes y dificultades en la toma de decisiones estratégicas que permitan aumentar los porcentajes de ventas efectivas. El desarrollo de una aplicación móvil abordara los desafíos específicos presentes al momento de obtener información precisa sobre los presupuestos en ventas asignados para los asesores comerciales, todo esto con la meta de mejorar la gestión de cumplimiento presupuestario y la toma de decisiones efectivas.

1.2 Antecedentes investigativos

Después de haber realizado una revisión literaria dentro de los repositorios de algunas Universidades se encontraron varios temas que sirvieron de apoyo para el trabajo propuesto.

A. Josue M. Chicaiza [3] plantea el uso del Ionic como un Framework para la creación de una aplicación para ayudar al E-Turismo de las PYMES del sector gastronómico tradicional, Ionic al ser un Framework multiplataforma permite crear aplicaciones usando como base Angular, React o Vue para la estructura de un proyecto, facilita el manejo de múltiples librerías en beneficio del proyecto de manera flexible y ágil, en conclusión el Framework Ionic genera ventajas en el Desarrollo Móvil que son populares y confiables.

D. Israel A. Sánchez [4] explora las funcionalidades con las que puede contar el Framework Ionic con la finalidad de analizar todas las herramientas que puedan ser orientadas al Desarrollo Móvil en el ámbito alimenticio, a su vez se muestra y demuestra que esta tecnología puede agilitar el desarrollo permitiendo la creación de aplicaciones ligeras que consumen menos recursos, permite la automatización de proceso, una curva de aprendizaje baja dado que utiliza lenguaje de programación Web y posee componentes y plugins listos para ser usados. Estos componentes permitieron que se desarrolle e implante un proyecto de calidad para la difusión de ofertas

alimenticias estudiantiles cumpliendo de manera ágil todos los requerimientos establecidos.

E. Arias D. Adrian [5] presenta la construcción de un Aplicativo Móvil multiplataforma para el control de inventario, los apartados que tiene gran influencia en el éxito del proyecto resaltados son, la construcción de un Aplicativo Móvil de manera sencilla, la utilización de estándares, con una selección y aplicación correcta de una metodología ágil como lo es Scrum permitió tener como resultado una Aplicación de calidad, en conclusión el desarrollo de un Aplicativo Móvil puede ser rápido y sencillo.

C. Arias N. Magaly [6] explica que el presupuesto es una descripción numérica formal de los objetivos que la gestión Empresarial pretende lograr durante un periodo determinado, junto con la implementación de estrategias y resalta la importancia de implementar un plan de marketing para cumplir con los planes presupuestarios. El control presupuestario dentro de una Empresa permite salvaguardar la continuidad de la misma y contribuye a una buena gestión administrativa.

G. Citelly P. Israel [7] menciona la importancia de optimizar el proceso de ventas teniendo un control minucioso que permita tener una gestión clara de los presupuestos, estimular las ventas en el mercado con un palan estratégico y tener un control claro y estructurado de la misma. En conclusión, tener un control efectivo del cumplimiento de presupuestos en las ventas es fundamental para el éxito y la sostenibilidad de cualquier Empresa, permitiendo evaluar el rendimiento real frente a las metas establecidas.

1.3 Fundamentación teórica

1.3.1 Multiplataforma

El enfoque multiplataforma en el desarrollo busca maximizar la eficiencia en términos de costos y beneficios al compartir el mismo código entre las diferentes versiones destinadas a distintas plataformas. Esto conlleva varias ventajas destacables, como una reducción en el tiempo y el costo de desarrollo, así como la capacidad de ofrecer

funcionalidades cercanas a las aplicaciones nativas al tener acceso al hardware del dispositivo. Además, cuenta con entornos de desarrollo robustos que facilitan este proceso. Una de las principales ventajas de los sistemas multiplataforma es su capacidad para ser instalados en cualquier sistema operativo sin necesidad de crear versiones específicas para cada dispositivo [8].

1.3.2 Ingeniería de Software

La ingeniería de software se define como una disciplina que utiliza los fundamentos de la ciencia de la computación y las matemáticas para encontrar soluciones eficaces y rentables a problemas relacionados con el desarrollo y mantenimiento de software [9].

a. Proceso de Desarrollo

El proceso de desarrollo de software se refiere a un marco de trabajo que se emplea para gestionar y guiar la creación de un producto de software.

b. Modelo Tradicional en Cascada

Este modelo es el más sencillo dado que las fases del proceso son ejecutadas de forma secuencial.

c. Modelo en Fuente

Este proceso sigue el modelo en cascada, pero hay algunas fases que dependen de las demás. Por lo tanto, la secuencia no es lineal, sino que tiene un ciclo.

d. Modelo en Espiral

Las actividades que se realizan es este modelo son de forma repetida y producen un aumento.

1.3.3 Metodologías de Desarrollo

Una metodología de desarrollo de software es un enfoque sistemático que organiza, gestiona y administra proyectos para asegurar su éxito. Esta metodología implica una serie de procesos ordenados y estructurados que abarcan desde la concepción de un producto de software hasta su implementación y mantenimiento [10].

a. Tipos de Metodologías

- **Metodología Tradicional**

También conocidas como metodologías pesadas son un conjunto de prácticas y técnicas que se utilizan para desarrollar software de forma disciplinada y sistemática. Estas metodologías se caracterizan por su enfoque detallado y planificado, con el objetivo de crear un software de alta calidad y eficiencia.

- **Metodología Ágil**

Las metodologías ágiles se basan en la adaptabilidad de los procesos de desarrollo. Estas metodologías se caracterizan por entregas frecuentes, una comunicación constante entre clientes y desarrolladores, un enfoque sencillo y la capacidad de adaptarse a los cambios de última hora.

1.3.4 Desarrollo de Software

La administración de proyectos es el proceso de planificar, organizar, dirigir y controlar el desarrollo de un producto o servicio, dentro de los plazos y presupuestos establecidos. Para ello, requiere de recursos humanos, por lo que debe tener en cuenta la organización técnica, las habilidades organizativas y el arte de dirigir un equipo [11].

a. Tipos de Desarrollo

- **Desarrollo Basado en Prototipos**

Los prototipos pueden ser elaborados de diferentes maneras, dependiendo del objetivo que se quiera alcanzar. Pueden ser el resultado de un proceso de desarrollo completo, o pueden ser bocetos simples que se utilizan para explorar ideas. También pueden utilizarse para anticipar el ciclo de vida del proyecto o implementación, o pueden ser parte de un enfoque evolutivo.

- **Desarrollo Interactivo e Incremental**

La construcción inicial de un pequeño fragmento de software que se va expandiendo gradualmente ayuda a los participantes del proceso a identificar y resolver problemas o inconformidades rápidamente, antes de que puedan causar un daño irreparable al proyecto.

- **Desarrollo Ágil**

Los procesos ágiles se caracterizan por su flexibilidad y adaptabilidad. En lugar de planificar todo el proyecto de antemano, los procesos ágiles se centran en obtener feedback de los usuarios a medida que el software se desarrolla. Este feedback se utiliza para ajustar el desarrollo del software en función de las necesidades de los usuarios.

1.3.5 Aplicación Móvil

En la actualidad, las aplicaciones móviles juegan un papel fundamental en la adaptación de los usuarios a la era digital, donde muchas de las actividades cotidianas pueden llevarse a cabo mediante dispositivos móviles inteligentes [12].

a. Cliente

El cliente web o móvil es una herramienta que permite al usuario acceder a los recursos de un servidor web. El usuario utiliza el cliente web o móvil para enviar una solicitud al servidor web, indicando el recurso que desea obtener.

b. Servido

Un servidor web es un programa informático que está siempre disponible para recibir solicitudes de conexión de los clientes web o móviles.

1.3.6 Tipos de aplicaciones móvil

Los diversos tipos de aplicaciones han sido creados para atender necesidades específicas, pero han evolucionado para ofrecer un mejor rendimiento. La incorporación de nuevas tecnologías ha permitido el desarrollo de aplicaciones más robustas y escalables. Además, los procesos de creación de aplicaciones se han simplificado, lo que ha contribuido a un crecimiento constante en el ámbito de las aplicaciones móviles [13].

a. Aplicación Nativa

Las aplicaciones nativas son programas diseñados específicamente para un dispositivo o plataforma particular. Esto significa que operan en el dispositivo sin requerir software adicional, ya que se han creado utilizando el lenguaje de programación propio de esa plataforma. Habitualmente, el término "aplicación nativa" se refiere a aplicaciones móviles, existiendo versiones nativas para cada sistema operativo, como iOS y Android.

b. Subtítulo cuarto nivel

Las aplicaciones híbridas son aplicaciones móviles desarrolladas utilizando lenguajes de programación web como HTML5, CSS o JavaScript, junto con un Framework que adapta la vista web a cualquier pantalla de un dispositivo móvil. En esencia, se trata de aplicaciones diseñadas para funcionar en múltiples sistemas operativos móviles,

como iOS, Android o Windows, lo que elimina la necesidad de crear una aplicación distinta para cada plataforma.

1.3.7 Sistemas operativos móvil

Un sistema operativo móvil es un conjunto de programas que permite las funcionalidades específicas de un teléfono móvil y proporciona servicios a las aplicaciones que se ejecutan en él. Su función principal es ofrecer las herramientas necesarias para que los usuarios puedan manejar y utilizar el dispositivo de manera cómoda y sencilla. La innovación y evolución de los dispositivos electrónicos ha dado lugar a una variedad de sistemas operativos móviles disponibles en la actualidad [14].

a. Android

Android es un sistema operativo móvil creado por la empresa estadounidense Google. Basado en el núcleo de Linux, su objetivo principal ha sido fomentar el uso de un sistema operativo abierto, gratuito, versátil y altamente seguro, desarrollado para dispositivos móviles como teléfonos inteligentes y tabletas.

c. iOS

iOS es un sistema operativo móvil creado por Apple Inc. exclusivamente para sus dispositivos como el iPhone, iPad y iPod Touch. En cuanto al desarrollo de aplicaciones, iOS ofrece una plataforma de desarrollo sólida que permite a los desarrolladores crear aplicaciones para los dispositivos móviles de Apple utilizando lenguajes de programación como Objective-C y Swift. Además, los desarrolladores tienen acceso a diversas herramientas y recursos de desarrollo, incluyendo Xcode, el entorno de desarrollo integrado de Apple.

1.3.8 Framework

Un Framework es una estructura conceptual que ayuda a organizar el desarrollo de software, incluye un lenguaje interpretado, bibliotecas y otros recursos que simplifican la integración de los diversos componentes de un proyecto.

a. Framework frontend Ionic

Es una herramienta gratuita que facilita el desarrollo de aplicaciones móviles, que utiliza herramientas como JavaScript, HTML y CSS para implementar interfaces gráficas [15].

- **Cordova**

Actúa como una capa de ejecución que permite que la aplicación web se comporte como una aplicación nativa. Esto se logra mediante el uso de plugins que proporcionan acceso a las funcionalidades nativas del dispositivo, como la cámara, el GPS y los contactos. Cordova empaqueta la aplicación web en un contenedor nativo para que pueda ser distribuida y ejecutada en plataformas móviles como iOS y Android.

- **Capacitor**

Permite el acceder a las funciones nativas del dispositivo de manera más sencilla y eficiente. Capacitor se integra perfectamente con proyectos Ionic, permitiendo que cualquier proyecto pueda ejecutarse en distintos tipos de plataformas iOS, Android, y Web. Capacitor facilita el desarrollo y la actualización de aplicaciones al proporcionar una mejor compatibilidad con la tecnología web moderna.

- **React**

Utilizado para gestionar la lógica del aplicativo y construir la interfaz de usuario. El Framework Ionic proporciona un conjunto de componentes de interfaz de usuario que se integran con React, permitiendo la creación de aplicativos móviles.

b. Framework backend Node.js

Es un entorno de ejecución multiplataforma basado en el motor V8 de Chrome. Diseñado principalmente para utilizar JavaScript en el servidor, aunque no de forma exclusiva, facilita la creación de aplicaciones de red altamente escalables, como Web Services [16].

- **Web Services**

Es un sistema de software diseñado para soportar la interoperabilidad entre diferentes aplicaciones en una red, permitir que diferentes sistemas se comuniquen entre sí, independientemente del lenguaje de programación o la plataforma en la que están desarrollados y permiten que las aplicaciones intercambien datos y servicios a través de la web.

- **Express**

Facilita la gestión de rutas, el manejo de solicitudes HTTP como GET, POST, PUT, DELETE, etc., la configuración de middleware, y la integración con motores de plantillas para generar respuestas HTML dinámicas. Es una de las herramientas más populares y ampliamente utilizadas para el desarrollo backend.

- **Cors**

Habilita el acceso de recursos a través de dominios diferentes, lo cual es esencial en muchas aplicaciones web o móviles modernas que consumen Web Services desde diferentes orígenes. Este módulo es crucial para solucionar problemas relacionados con la seguridad de las solicitudes web o móviles desde un origen (dominio) diferente al origen donde se aloja el recurso solicitado.

- **Mssql**

Permite ejecutar consultas SQL, almacenar procedimientos, gestionar transacciones y otras operaciones de base de datos directamente desde un proyecto Node.js, utilizado para aplicaciones que requieren una integración sólida con SQL Server como parte de su backend.

- **Helmet**

Configura de manera adecuada encabezados HTTP como Content Security Policy, X-Content-Type-Options, X-Frame-Options, Strict-Transport-Security, entre otros. Estas configuraciones ayudan a prevenir una variedad de ataques web, como captura de clics, ataques de inyección de contenido y ataques de scripting entre sitios.

1.3.9 Motores base de datos

Software que gestiona y manipula los datos en una base de datos, se encarga de realizar todas las tareas relacionadas con la gestión de los datos, incluyendo el almacenamiento, la consulta, la actualización y la eliminación de datos, así como el mantenimiento de la integridad y la seguridad de la base de datos.

a. Base de datos relacionales

Reflejan los eventos del mundo real mediante transacciones que poseen las propiedades de atomicidad, consistencia, aislamiento y durabilidad [17].

- **Sql Server**

Sistema de gestión de bases de datos relacional (RDBMS) desarrollado por Microsoft. Utiliza el lenguaje de consulta estructurado (SQL) para gestionar y manipular datos almacenados en bases de datos relacionales.

b. Base de datos no relacionales

Son sistemas de gestión de bases de datos que no utilizan el modelo de tablas relacionales tradicionales para organizar los datos. En lugar de esto, utilizan varios modelos de datos para almacenar y gestionar información, lo que las hace adecuadas para manejar grandes volúmenes de datos no estructurados o semi estructurados [17].

1.3.10 Planificación Financiera

La planificación financiera se reconoce como un proceso empresarial que refleja la situación económica a través de diversas actividades y cambios. En este contexto, la formulación de la estrategia organizacional implica el diseño de objetivos, la selección de estrategias y políticas, la ejecución de acciones y la implementación de mecanismos de control relacionados con las decisiones de inversión y financiamiento, teniendo en cuenta sus implicaciones tanto a corto como a largo plazo [18].

La gestión eficiente de los recursos financieros resulta crucial para el funcionamiento de una organización, por lo que es fundamental emplear este recurso de manera óptima y maximizar su aprovechamiento en las inversiones de la entidad.

1.3.11 Asignación de Recursos

Se refiere a una estrategia que facilita el logro de los objetivos principales utilizando los recursos disponibles. Estos objetivos deben ser definidos de manera consensuada entre expertos, sociedades científicas y miembros de organizaciones [19].

a. Asignación de recurso financieros

Implica tomar decisiones sobre cómo invertir, gastar y asignar los recursos financieros de la empresa en diferentes áreas y proyectos para alcanzar los objetivos organizacionales.

b. Asignación de recurso humanos

Incluye la contratación, capacitación, desarrollo y asignación de personal para cumplir con las funciones y responsabilidades específicas dentro de una empresa.

1.3.12 Toma de Decisiones

La toma de decisiones se ha categorizado como una de las funciones ejecutivas, que son consideradas como actividades cognitivas superiores ampliamente estudiadas [20].

a. Toma de decisiones programadas

Son aquellas decisiones que se toman de forma repetitiva y rutinaria, para las cuales se han establecido procedimientos o reglas predefinidas.

b. Toma de decisiones no programadas

Son decisiones únicas o infrecuentes para las cuales no existen procedimientos preestablecidos, por lo que requieren un análisis más detallado y juicio por parte de los responsables.

1.3.13 Presupuestos

Un presupuesto es un análisis sistemático que examina tanto el futuro como el presente de un proceso productivo y financiero en una empresa. Este análisis calcula los recursos de entrada y salida, como dinero, tiempo, materiales, uso de maquinaria y espacio, entre otros [21].

El objetivo del presupuesto es proporcionar indicadores financieros que reflejen la cantidad y el costo de los recursos necesarios para desarrollar el producto, abarcando el proceso productivo. Además, debe ofrecer datos concretos sobre la rentabilidad, la utilidad esperada, el flujo de efectivo y otros indicadores financieros.

a. Presupuesto operativo

Proyecta los ingresos y gastos relacionados con las operaciones diarias de la empresa, como ventas, costos de producción, y gastos operativos. Su objetivo es proporcionar una visión detallada de la rentabilidad operativa.

b. Presupuesto financiero

Incluye la planificación de flujos de efectivo, estados de resultados y balances proyectados. Su propósito es asegurar que la empresa tenga suficientes recursos financieros para operar y cumplir con sus obligaciones.

c. Presupuesto de ventas

Proyecta las ventas futuras en términos de unidades y valor monetario. Es fundamental para la planificación de ingresos y para establecer metas de ventas.

d. Presupuesto de producción

Detalla la cantidad de productos que se deben fabricar en un período determinado, considerando las ventas proyectadas y los niveles de inventario. Ayuda a planificar la capacidad de producción y la gestión de inventarios.

1.4 Objetivos

1.4.1 Objetivo general

Implantar una Aplicación Móvil para la gestión de presupuestos de ventas usando el Framework Ionic en la Empresa Rectima Industry Cia. Ltda.

1.4.2 Objetivos específicos

- Analizar los procesos que interviene en la planificación de presupuestos de ventas
- Investigar los veneficios del Framework Ionic en el Desarrollo Móvil
- Desarrollar una Aplicación Móvil para la gestión de presupuestos de ventas con el Framework Ionic en la Empresa Rectima Industry Cia. Ltda.

CAPÍTULO II. METODOLOGÍA

2.1 Materiales

Para recolectar la información del presente proyecto de investigación se aplicó una entrevista dirigida al jefe del área de sistemas y una encuesta con cuestionario dirigido a los asesores comerciales encargado del cumplimiento de presupuestos en ventas.

a. Guía de entrevista aplicada al jefe del área de sistemas

La entrevista que se realizó al jefe del área de sistemas según instrumento de la Tabla 1 tuvo como propósito obtener información sobre los que intervienen en el proceso de planificación de presupuestos en ventas.

Tabla 1. Instrumento elaborado para la entrevista

Datos personales **Apellido y Nombre jefe área de sistemas:** **Institución:** **Cargo:** **Objetivo:**			
N°	**Pregunta**	**Respuesta**	**Observación**
1	**¿Quiénes participan en la elaboración del presupuesto de ventas?**		
2	**¿Cuál es su papel en la elaboración del presupuesto de ventas?**		
3	**¿Qué información se utiliza para elaborar el presupuesto de ventas?**		
4	**¿Podría describir el proceso general que se utiliza para elaborar el presupuesto de ventas?**		
5	**¿Se emplea algún tipo de incentivo por el cumplimiento de los presupuestos?**		
6	**¿Cómo se monitorea el avance del cumplimiento de presupuestos?**		
7	**¿Considera que el proceso actual de comunicación de los cumplimientos es el adecuado?**		
8	**¿Considera que el acceso a la información de los presupuestos es oportuno?**		
Conclusiones:			

b. Encuesta dirigida a los asesores comerciales encargado del cumplimiento de presupuestos en ventas

El instrumento elaborado para la encuesta aplicada a los asesores comerciales tuvo como objetivo conocer la perspectiva que tienen sobre la automatización de procesos para la gestión de presupuestos en ventas.

El cuestionario está formado por diez preguntas que se detallan a continuación:

- **¿Con qué frecuencia recibe información sobre el cumplimiento de sus objetivos en ventas?**
  - Diariamente
  - Semanalmente
  - Mensualmente
  - No recibo información
- **¿En qué formato recibe la información sobre el cumplimiento de ventas?**
  - Informes por correo electrónico
  - Mensajes de texto
  - Reuniones con el gerente de ventas
  - Otros
- **¿Qué tan satisfecho está con la calidad y precisión de la información que recibe?**
  - Muy satisfecho
  - Satisfecho
  - Insatisfecho
  - Muy Insatisfecho

- **¿Con qué frecuencia experimenta problemas con la información que recibe?**
  - Todos los días
  - Ocasionalmente
  - Extrañamente
  - Nunca
- **¿Cuál cree que es el mayor desafío al intentar acceder a información sobre presupuestos?**
  - Lentitud en la entrega de datos
  - Falta de comunicación
  - Falta de claridad en la presentación de datos
  - Ninguno
- **¿Cómo afecta el limitado acceso a la información sobre el cumplimiento de ventas al momento de poder alcanzar sus objetivos comerciales?**
  - Dificulta la identificación de áreas de mejora
  - Afecta la toma de decisiones estratégicas
  - Incrementa el riesgo de errores y malentendidos
  - Otros
- **¿Qué tipo de información considera más importante para tomar decisiones comerciales?**
  - Nivel de ventas de productos
  - Marcas más vendidas
  - Análisis de clientes

- Ninguna

- **¿Qué tan importante considera el uso de tecnología para mejorar la gestión de la información sobre el cumplimiento en ventas?**

- Muy importante
- Importante
- Neutro
- Poco importante

- **¿Considera útil una aplicación móvil para la gestión de presupuestos?**

- Muy útil
- Útil
- Poco útil
- Nada útil

- **¿Considera favorable utilizar una aplicación móvil para una gestión rápida y fácil de información sobre presupuestos de ventas?**

- Muy satisfecho
- Satisfecho
- Insatisfecho
- Muy Insatisfecho

2.2 Métodos

2.2.1 Modalidad de la investigación

La presente investigación se contextualizó en la modalidad de investigación de campo y documental-bibliográfica.

a. Investigación de campo

La investigación se realizó en el lugar de los hechos, donde el investigador recolectó datos a través de encuestas y entrevista. Estos datos se utilizaron para identificar y analizar el problema que se estaba investigando.

b. Investigación Bibliográfica-Documental

La investigación se basó en el análisis de fuentes bibliográficas y documentales, como libros, artículos, revistas, tesis, entre otros, que sirvió para construir una fundamentación teórica sobre temas como ingeniería de software, metodología de desarrollo, planificación financiera, asignación de recursos, toma de decisiones, entre otros

2.2.2 Población y muestra

La población con la cual se trabajó fue el jefe del área de sistemas y los asesores comerciales encargados de las vetas que se detallan en la Tabla 2.

Tabla 2. Población

Población	**Número**	**Porcentaje**
Jefe Área de Sistemas	1	7,14%
Asesores Comerciales	13	92.86%
Total	14	**100%**

El enfoque de la investigación se dirigió a una población que no supero los cien elementos, lo que implico no llevar a cabo muestras representativas, sino que se trabajó con la totalidad de la población.

2.2.3 Recolección de información

La recolección de la información se realizó por medio de una encuesta dirigida a 13 asesores comerciales y por medio de una entrevista que se la aplicó de manera presencial al jefe del área de sistemas para obtener los aspectos importantes de todo el proceso para la planificación de presupuestos en ventas.

a. Resultado entrevista aplicada al jefe del área de sistemas

En esta sección se presentan los resultados obtenidos a partir de una entrevista aplicada al jefe del área de sistemas, los resultados obtenidos que se presentan en la Tabla 3 datos relevantes como las personas que participan en la creación y el proceso seguido para la generación del presupuesto en ventas.

Tabla 3. Resultados guía de entrevista jefe de área informática

Datos personales **Apellido y Nombre jefe área de sistemas:** Cherres Paul **Institución:** Rectima Industry Cia. Ltda. **Cargo:** jefe Área de Sistemas **Objetivo:** Analizar los procesos que intervienen en la planificación del presupuesto de ventas			
N°	**Pregunta**	**Respuesta**	**Observación**
1	**¿Quiénes participan en la elaboración del presupuesto de ventas?**	Las personas que participan en el proceso de elaboración de presupuestos son: Gerente Comercial, Supervisor de Ventas y Gerente General	La elaboración del presupuesto en ventas se realiza por el quipo en su mayoría Gerentes
2	**¿Cuál es su papel en la elaboración del presupuesto de ventas?**	Su papel es recolectar la información de los diferentes presupuestos que serán asignados a los asesores comerciales	El jefe de área informática recibe los presupuestos generados
3	**¿Qué información se utiliza para elaborar el presupuesto de ventas?**	Se utiliza las ventas de los años anteriores, artículos más vendidos y un estudio del mercado	
4	**¿Podría describir el proceso general que se utiliza para elaborar el presupuesto de ventas?**	El proceso inicia con un estudio del mercado, análisis de la situación actual de ventas y proyección de los productos que se puede comercializar en futuro, posteriormente se analizan las ventas realizadas a los clientes que se encuentran registrados en la empresa analizando el nivel de compra. Con esta información se asigna un determinado número de clientes a cada uno de los asesores comerciales dependiendo de las zonas donde se encuentran los clientes con lo cual se obtiene una parte del presupuesto total de cada uno de los asesores comerciales. De la misma manera se agrupan los productos dependiendo de las características para determinar el total del presupuesto por grupos de clientes y productos, de esta forma cada uno de los asesores tienen asignado un presupuesto en ventas.	El proceso seguido tiene varios pasos importantes, el cumplimiento efectivo de estos procesos permiten la generación efectiva de los presupuestos en ventas.
5	**¿Se emplea algún tipo de incentivo por el cumplimiento de los presupuestos?**	Si se emplean incentivos por cumplir de presupuesto asignado en cada uno de los meses el asesor comercial recibe un porcentaje de las ventas que obtuvo durante	Se aplica beneficios por cumplimiento en presupuestos de ventas, estos se

		el mes, siempre y cuando los asesores cumplan con el porcentaje mínimo establecido. Estos incentivos se reconocen por cumplimiento de presupuesto que es equivalente al 95% de cumplimiento, por cumplimiento de promociones que es equivalente al 100% de ventas en productos promocionales y por atención al cliente el cual debe llegar al 50% en atenciones.	otorgan según el porcentaje de cumplimiento alcanzado por mes
6	**¿Cómo se monitorea el avance del cumplimiento de presupuestos?**	Actualmente se tiene implementado un sistema de notificación vía correo electrónico que se genera todos los días a las 7:00 am con información de las ventas realizadas hasta el día anterior, de esta manera se tiene un control del avance	El monitoreo se realiza por medio de notificaciones vía correo electrónico siendo poco efectivo al no dar un seguimiento continuo
7	**¿Considera que el proceso actual de comunicación de los cumplimientos es el adecuado?**	Considera que no porque al momento no se dispone de información adecuada y verídica para la toma acertada de decisiones estratégicas para el veneficio de la Empresa.	
8	**¿Considera que el acceso a la información de los presupuestos es oportuno?**	No, es limitada dado que solo se tiene acceso a la información una vez al día siendo ineficiente e inadecuado para el cumplimiento de los presupuestos.	Se puede observar que no se tiene un correcto acceso a la información
Conclusiones: La generación de los presupuestos en ventas se realiza a través de un grupo de personas que definen los presupuestos a través de un conjunto de procesos, que comienzan en el análisis del mercado, ventas realizadas, asignación de clientes y agrupación de productos, esto permite la creación del presupuesto que será asignado a cada asesor comercial, su cumplimiento efectivo permite la obtención de incentivos que motivan a poder cumplir con los presupuestos asignados, al concluir el proceso de creación de presupuestos se realiza la notificación del presupuesto asignado vía correo electrónico, la entrega de información por este medio tiene limitantes que impactan en el cumplimiento efectivo de los presupuestos asignados a los asesores comerciales		

b. Validación encueta aplicada a los asesores comerciales

El coeficiente de Alfa de Cronbach, una herramienta estadística y psicométrica comúnmente utilizada para evaluar la consistencia interna de un conjunto de ítems en escalas de medición, fue aplicado para validar la confiabilidad de los datos recopilados a través de encuestas a los asesores comerciales.

Fórmula

$$\alpha = \frac{k}{k-1}\left[1 - \frac{\sum S_i^2}{S_t^2}\right]$$

Donde:

k = El número de ítems

∑S_i^2 = Sumatoria de la varianza de los ítems

S_t^2 = Varianza de la suma de los ítems

Para evaluar la confiabilidad interna de las preguntas elaboradas en escalas de Likert, se empleó el coeficiente de Alfa de Cronbach, utilizando la herramienta Excel para realizar el cálculo correspondiente.

	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	Suma
Sujeto 1	4	3	1	1	2	2	3	3	2	2	23
Sujeto 2	4	3	2	2	4	2	4	3	3	3	30
Sujeto 3	3	3	2	2	2	2	3	3	3	3	26
Sujeto 4	4	3	2	2	2	2	3	2	3	3	26
Sujeto 5	4	3	2	2	2	3	4	3	3	3	29
Sujeto 6	4	3	2	2	3	3	3	3	3	3	29
Sujeto 7	4	3	2	2	3	3	4	3	4	3	31
Sujeto 8	4	4	2	1	4	4	4	4	4	4	35
Sujeto 9	4	4	2	2	4	4	4	4	4	4	36
Sujeto 10	4	4	2	2	2	4	4	3	4	4	33
Sujeto 11	4	4	1	1	4	4	4	3	4	4	33
Sujeto 12	4	3	2	2	2	4	4	3	4	4	32
Sujeto 13	4	3	2	1	4	4	3	3	4	4	32
Varianza	0.071	0.213	0.130	0.213	0.840	0.746	0.237	0.225	0.402	0.391	13.006

Formula	Valor
α (Alfa) =	0.815
k (Num. Items) =	10
Vi (Varianza Item) =	3.467455621
Vt (Varianza Total) =	13.00591716

$$\alpha = \frac{k}{k-1}\left[1 - \frac{\sum V_i}{V_t}\right]$$

$$\alpha = \frac{10}{10-1}\left[1 - \frac{3.467}{13.006}\right] = 0.815$$

Figura 1. Calculo Alfa de Cronbach

En la fórmula de Alfa de Cronbach se reemplazó los valores calculados de la siguiente manera:

$$\alpha = \frac{k}{k-1}\left[1 - \frac{\sum S_i^2}{S_t^2}\right]$$

Tabla 4. Confiabilidad Alfa de Cronbach

Ítem	**Valor**
α (Alfa)	0.815
k (Número de Ítem)	10
S_i^2 (Varianza de cada Ítem)	3.467455621
S_t^2 (Varianza total)	13.00591716

Tabla 5. Confiabilidad Alfa de Cronbach

Ítem	**Valor**
Mayor 0.90	Excelente
0.80-0.89	Buena
0.70-0.79	Aceptable
0.60-0.69	Deficiente

Según el coeficiente de Alfa de Cronbach presentado en la Tabla 4, la confiabilidad de los datos es de 0.815, valor que se encuentra dentro del rango bueno Tabla 5 para la validación de los datos.

c. Resultados encuetas aplicada a los asesores comerciales

Primera pregunta: ¿Con qué frecuencia recibe información sobre el cumplimiento de sus objetivos en ventas?

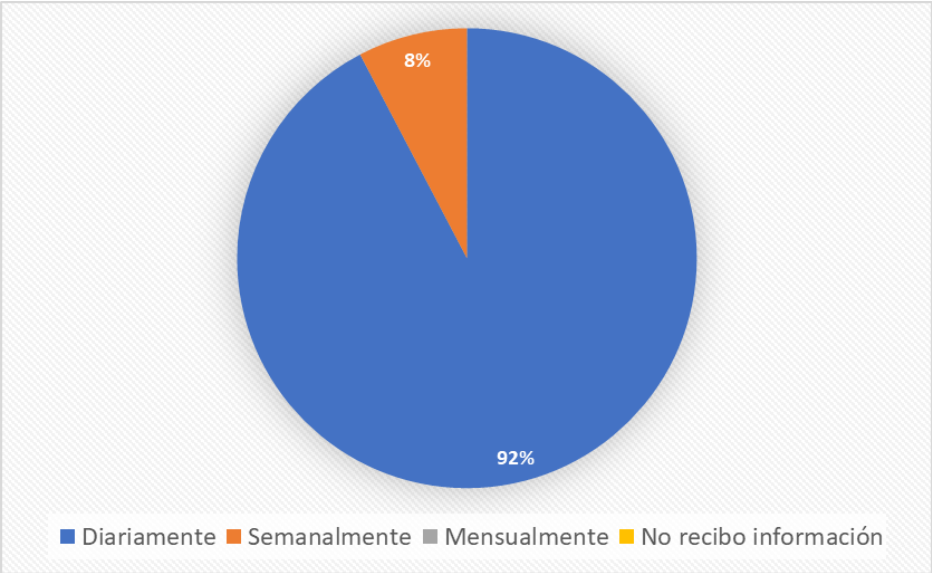


Figura 2. Pregunta 1

Análisis e interpretación de resultados

Como se presenta en la Figura 2, de 13 asesores comerciales encuestados el 92% recibe información sobre su cumplimiento en ventas diariamente, lo que indica una alta frecuencia en la comunicación. En contraste, el 8% recibe esta información de manera semanal, sugiriendo que la frecuencia de actualización de información no es la óptima.

Segunda pregunta: ¿En qué formato recibe la información sobre el cumplimiento de ventas?

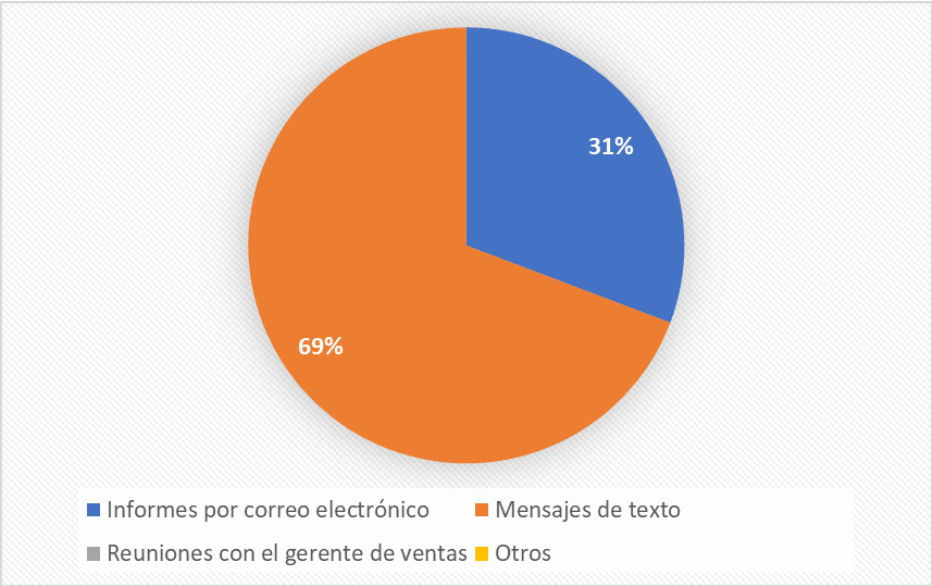


Figura 3. Pregunta 2

Análisis e interpretación de resultados

Como se presenta en la Figura 3, de 13 asesores comerciales encuestados el 69% recibe la información a través de mensajes de texto, indicando que este medio es el canal predominante para la entrega de la información. Por otro lado, el 31% recibe la información por correo electrónico, aunque menos común, este medio también se utiliza para la transmisión de la información. Estas diferencias en las preferencias de comunicación indican la importancia de un enfoque flexible para la entrega de la información.

Tercera pregunta: ¿Qué tan satisfecho está con la calidad y precisión de la información que recibe?

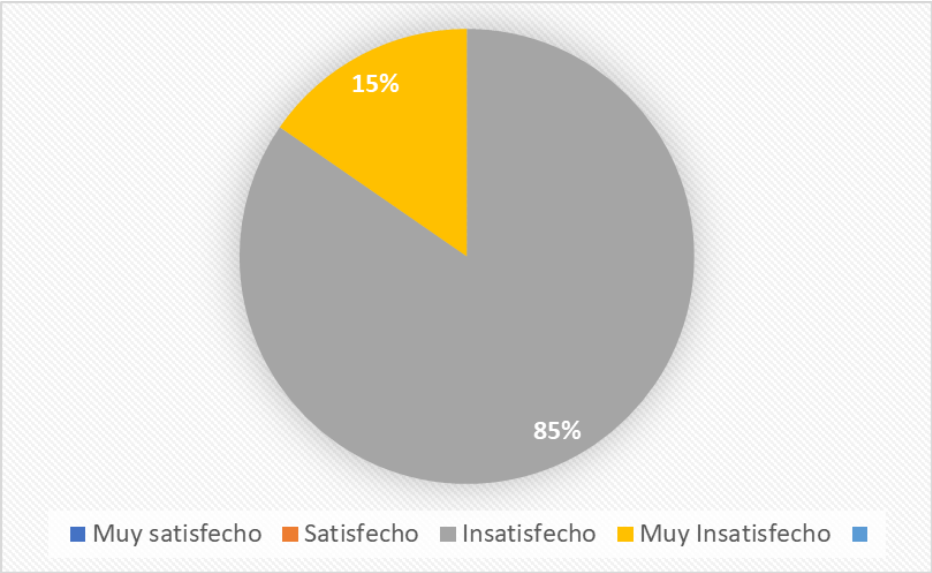


Figura 4. Pregunta 3

Análisis e interpretación de resultados

Como se presenta en la Figura 4, de 13 asesores comerciales encuestados el 85% expresa insatisfacción con la manera en que se presenta la información, lo que sugiere que existe una percepción de inconformidad con la entrega de la información. Además, el 15% restante se encuentra muy insatisfecho con la calidad y precisión de la información, indicando una preocupación mayor sobre la entrega y exactitud de los datos recibidos. La insatisfacción general sugiere la relevancia de una mejorar en la estructura y exactitud de la información proporcionada.

Cuarta pregunta: ¿Con qué frecuencia experimenta problemas con la información que recibe?

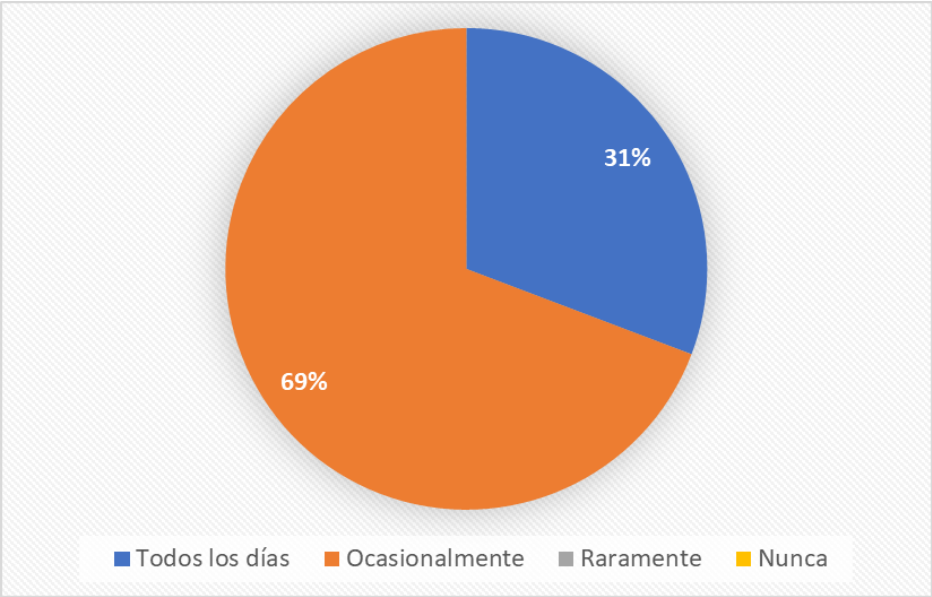


Figura 5. Pregunta 4

Análisis e interpretación de resultados

Como se presenta en la Figura 5, de 13 asesores comerciales encuestados el 69% indica que ocasionalmente presenta problemas con la información, sugiriendo que estos inconvenientes son de manera más frecuentes. En contraste, el 31% reporta que enfrentan inconvenientes todos los días con la información recibida, lo que refleja un problema que persiste de manera diario. La alta incidencia de problemas denota la importancia de abordar los puntos débiles en el flujo de entrega de la información.

Quinta pregunta: ¿Cuál cree que es el mayor desafío al intentar acceder a información sobre presupuestos?

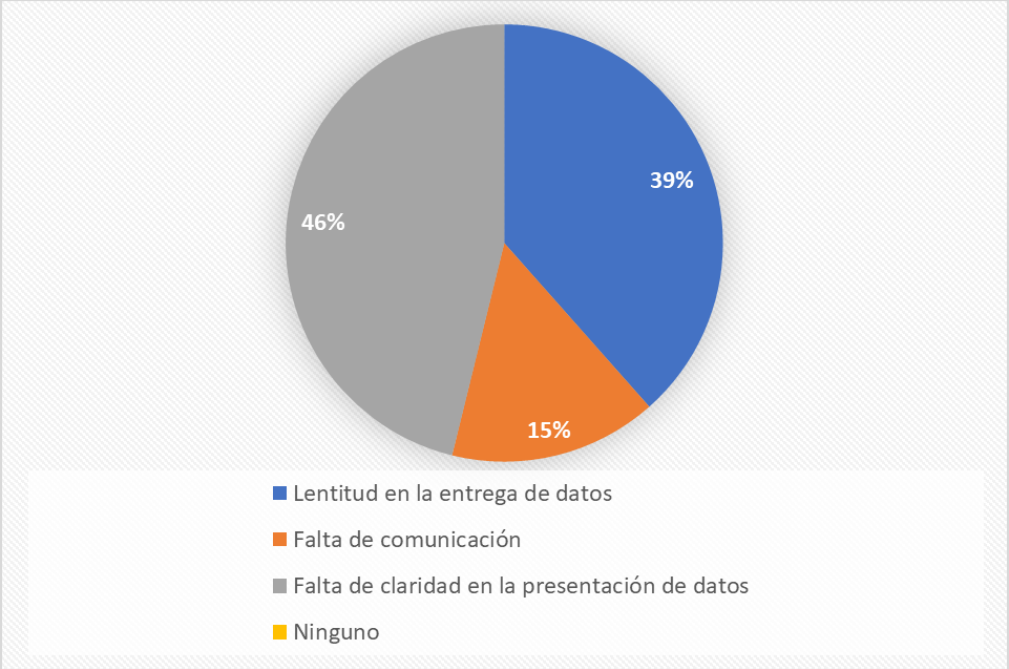


Figura 6. Pregunta 5

Análisis e interpretación de resultados

Como se presenta en la Figura 6, de 13 asesores comerciales encuestados el 46% identifica la falta de claridad en la presentación de los datos como el mayor obstáculo, indicando que una comunicación mal estructurada de la información es un problema significativo. El 39% señala que la lentitud en la entrega de datos representa un desafío importante, sugiriendo que los retrasos en la provisión de la información tengan influencia al acceder a la información de presupuestos. Por otro lado, el 15% menciona la falta de una comunicación efectiva como un desafío, lo que resalta la importancia de una comunicación clara y directa para la presentación de la información. Estos hallazgos subrayan la importancia de abordar estos problemas para optimizar la presentación de la información.

Sexta pregunta: ¿Cómo afecta el limitado acceso a la información sobre el cumplimiento de ventas al momento de poder alcanzar sus objetivos comerciales?

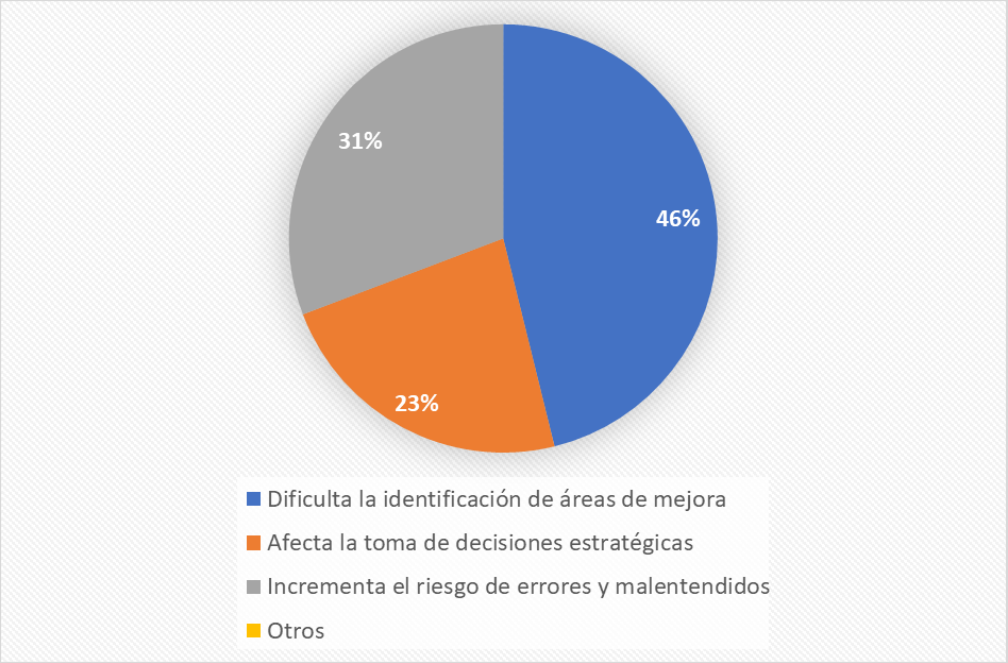


Figura 7. Pregunta 6

Análisis e interpretación de resultados

Como se presenta en la Figura 7, de 13 asesores comerciales encuestados el 46% señala que la dificultad para identificar áreas de mejora es un obstáculo importante, lo que sugiere que al no haber claridad al identificar áreas de mejora influye en la efectividad general. Por otro lado, el 31% identifica el aumento del riesgo de errores o malentendidos como un factor crítico que trascendería en el alcance de objetivos, indicando que la comunicación y la precisión son importantes para evitar fallos. Finalmente, el 23% menciona que la toma de decisiones estratégicas tiene un impacto directo en su capacidad para cumplir objetivos, lo que resalta la necesidad de una buena toma de decisiones fundamentadas y estratégicas para alcanzar los objetivos.

Séptima pregunta: ¿Qué tipo de información considera más importante para tomar decisiones comerciales?

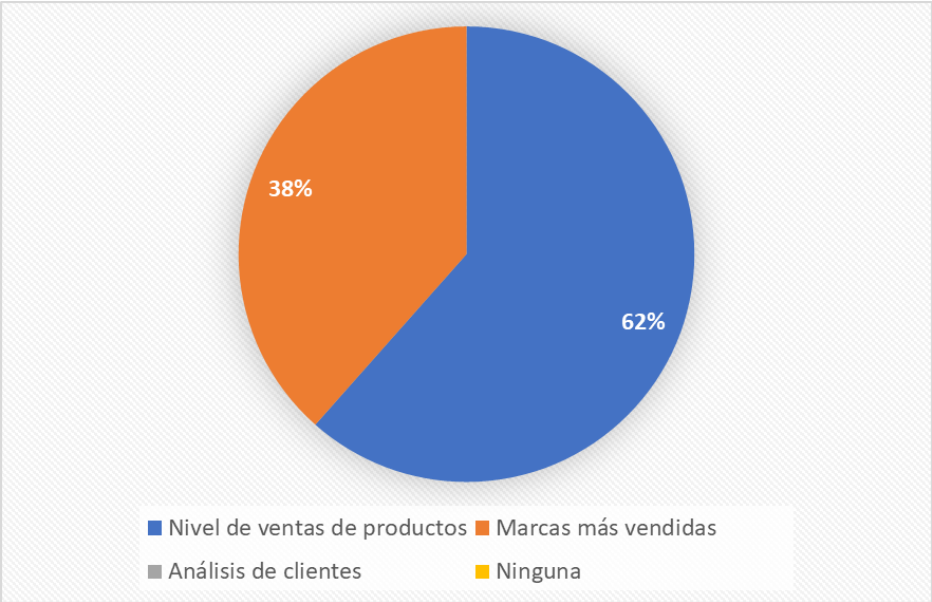


Figura 8. Pregunta 7

Análisis e interpretación de resultados

Como se presenta en la Figura 8, de 13 asesores comerciales encuestados el 62% considera que el nivel de ventas de productos es un factor crucial para la toma de decisiones comerciales. Esto sugiere que el desempeño de ventas es visto como un indicador para guiar las estrategias y decisiones dentro de la empresa. Por otro lado, el 38% de los encuestados destaca la importancia de las marcas más vendidas en la toma de decisiones comerciales, indicando que el éxito de las marcas es importante en las estrategias de negocio.

Octava pregunta: ¿Qué tan importante considera el uso de tecnología para mejorar la gestión de la información sobre el cumplimiento en ventas?

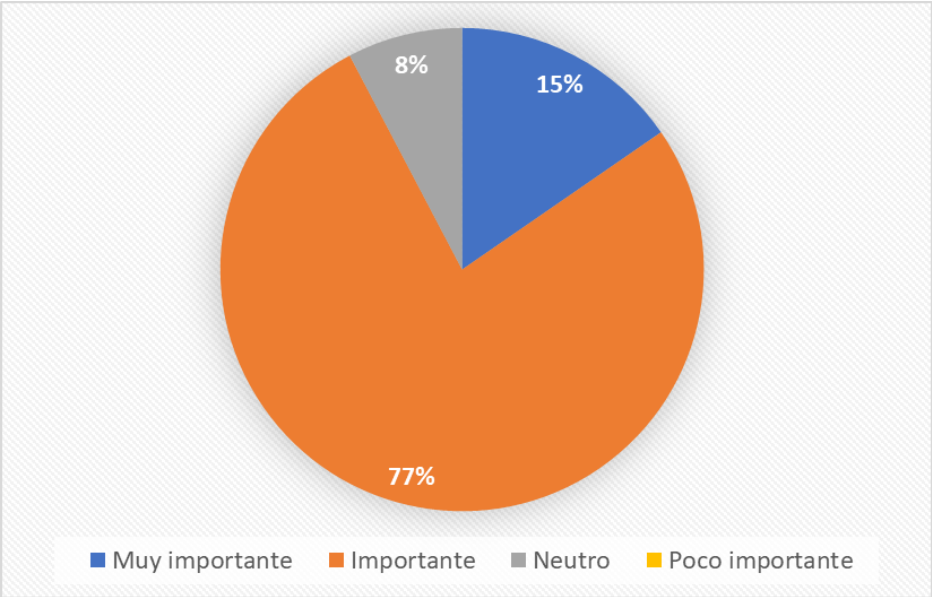


Figura 9. Pregunta 8

Análisis e interpretación de resultados

Como se presenta en la Figura 9, de 13 asesores comerciales encuestados el 77% considera que el uso de tecnología es crucial para gestionar la información relacionada con las ventas, lo que indica una clara apreciación de las herramientas tecnológicas como un facilitador para este proceso. Además, el 15% manifiesta que integrar tecnología en la gestión de cumplimientos es muy importancia, lo que sugiere una inclinación aun mayo a la tecnología y su papel en la optimización de la gestión de presupuestos. Por otro lado, el 8% no afirma ni descarta la implementación de tecnología para la gestión del cumplimiento en ventas, mostrando una menor convicción sobre su impacto.

Novena pregunta: ¿Considera útil una aplicación móvil para la gestión de presupuestos?

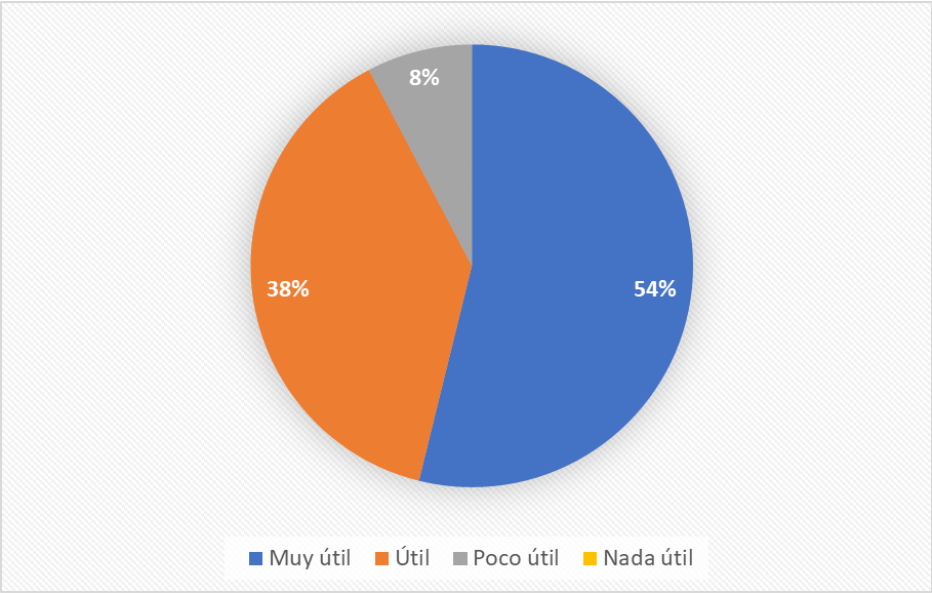


Figura 10. Pregunta 9

Análisis e interpretación de resultados

Como se presenta en la Figura 10, de 13 asesores comerciales encuestados el 54% considera que contar con una aplicación móvil para esta función es muy útil, destacando una preferencia por herramientas móviles para la gestión de presupuestos. Un 38% adicional considera que la iniciativa de desarrollar una aplicación móvil para la gestión de presupuestos es útil, mostrando un grado menor de entusiasmo. En contraste, el 8% opina que una aplicación móvil para la gestión de presupuestos es poco útil, reflejando una perspectiva más escéptica respecto a su relevancia.

Decima pregunta: ¿Considera favorable utilizar una aplicación móvil para una gestión rápida y fácil de información sobre presupuestos de ventas?

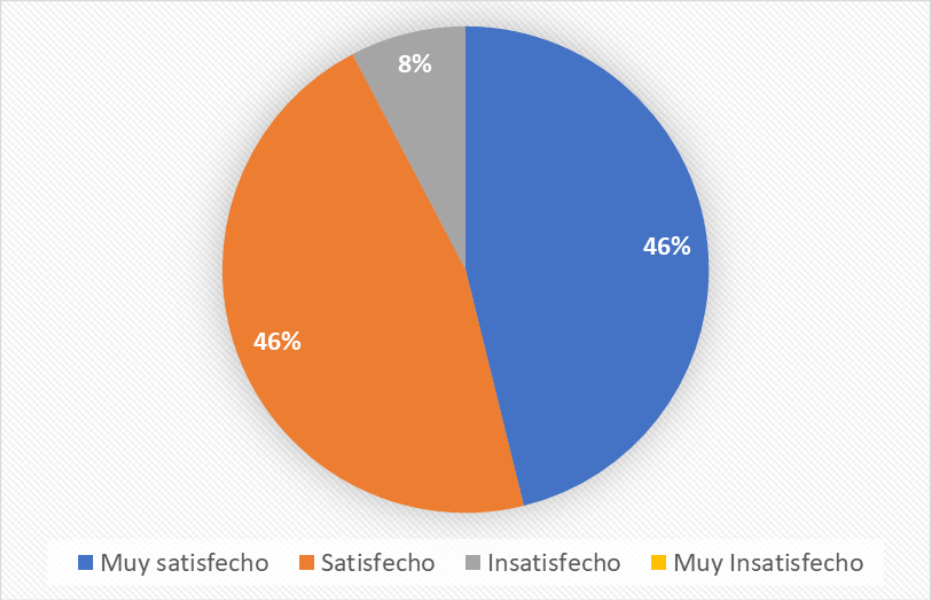


Figura 11. Pregunta 10

Análisis e interpretación de resultados

Como se presenta en la Figura 11, de 13 asesores comerciales encuestados el 92% apoya la idea de utilizar un aplicativo móvil para la gestión de presupuestos. Este nivel de aceptación sugiere que la mayoría valora la capacidad de las aplicaciones móviles para presentar la información de presupuestos de mejor manera. Sin embargo, el 8% restante expresa reservas o inconformidad con la aplicación móvil, evidenciando una minoría que presenta dudas sobre la implementación o funcionalidad de la herramienta. Estos resultados destacan una importante aceptación general de la tecnología móvil para la presentación de presupuestos en ventas.

2.2.4 Procesamiento y análisis de datos

Con los datos recopilados dentro de la Empresa Rectima Industry Cia. Ltda. se realiza el análisis estadístico y de procedimientos dentro de la empresa, estos datos son obtenidos mediante encuesta y entrevista, para la encuesta se representan los resultados mediante un gráfico representativo Figura 12 para su análisis y para el procedimiento

de planificación de presupuestos se representa mediante un gráfico Figura 13 de los pasos a seguidos.

Datos encuesta

Tabla 6. Datos análisis encuesta

Pregunta 1: ¿Con qué frecuencia recibe información sobre el cumplimiento de sus objetivos en ventas?					
Opción 1	Diariamente	**Opción 2**	Semanalmente		
Pregunta 2: ¿En qué formato recibe la información sobre el cumplimiento de ventas?					
Opción 1	Informes por correo electrónico	**Opción 2**	Mensajes de texto		
Pregunta 3: ¿Qué tan satisfecho está con la calidad y precisión de la información que recibe?					
Opción 3	Insatisfecho	**Opción 4**	Muy Insatisfecho		
Pregunta 4: ¿Con qué frecuencia experimenta problemas con la información que recibe?					
Opción 1	Todos los días	**Opción 2**	Ocasionalmente		
Pregunta 5: ¿Cuál cree que es el mayor desafío al intentar acceder a información sobre presupuestos?					
Opción 1	Lentitud en la entrega de datos	**Opción 2**	Falta de comunicación	**Opción 3**	Falta de claridad en la presentación de datos
Pregunta 6: ¿Cómo afecta el limitado acceso a la información sobre el cumplimiento de ventas al momento de poder alcanzar sus objetivos comerciales?					
Opción 1	Dificulta la identificación de áreas de mejora	**Opción 2**	Afecta la toma de decisiones estratégicas	**Opción 3**	Incrementa el riesgo de errores y malentendidos
Pregunta 7: ¿Qué tipo de información considera más importante para tomar decisiones comerciales?					
Opción 1	Nivel de ventas de productos	**Opción 2**	Marcas más vendidas		
Pregunta 8: ¿Qué tan importante considera el uso de tecnología para mejorar la gestión de la información sobre el cumplimiento en ventas?					
Opción 1	Muy importante	**Opción 2**	Importante	**Opción 3**	Neutro
Pregunta 9: ¿Considera útil una aplicación móvil para la gestión de presupuestos?					
Opción 1	Muy útil	**Opción 2**	Útil	**Opción 3**	Poco útil
Pregunta 10: ¿Considera favorable utilizar una aplicación móvil para una gestión rápida y fácil de información sobre presupuestos de ventas?					
Opción 1	Muy satisfecho	**Opción 2**	Satisfecho	**Opción 3**	Insatisfecho

Gráfico encueta

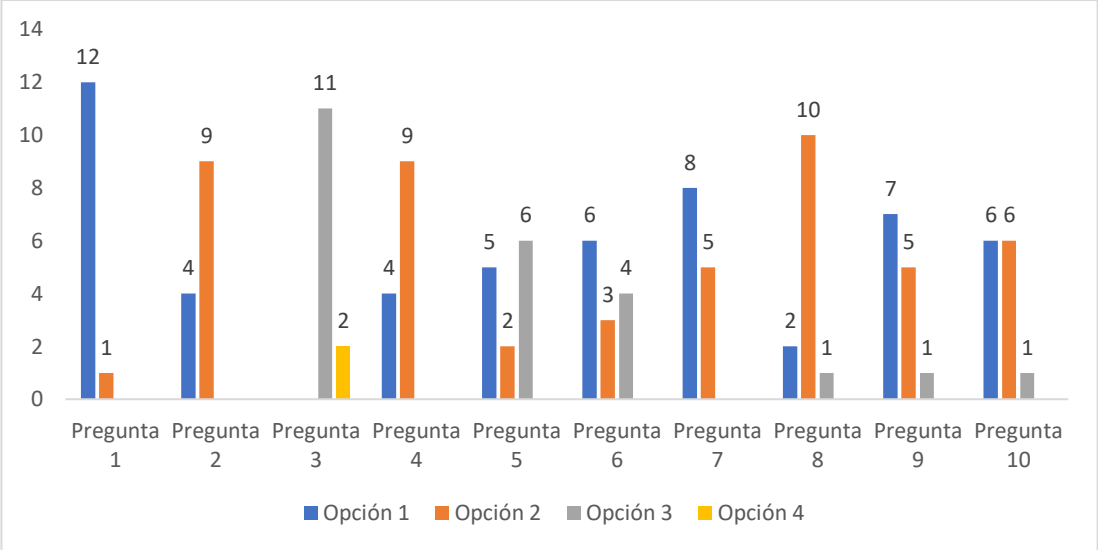


Figura 12. Estadística respuestas encuesta

En relación a los datos se agrupan en función de: frecuencia y calidad de información (Preguntas 1, 2, 3 y 4), desafíos y afectaciones en el cumplimiento de objetivos (Preguntas 5 y 6), importancia de información y tecnología (Preguntas 7 y 8) y utilidad de un aplicativo móvil (Preguntas 9 y 10).

a. Frecuencia y calidad de información

Al recibir información diariamente es positivo, pero la insatisfacción generalizada con la calidad sugiere que la frecuencia de entrega no compensa la falta de precisión.

La insatisfacción y los problemas frecuentes indican que la información no es relevante en su forma actual, lo cual influiría estar negativamente en el cumplimiento de presupuestos.

b. Desafíos y afectaciones en el cumplimiento de objetivos

La falta de claridad es el desafío más frecuente, lo cual sugiere que se debe mejorar la presentación de la información.

La dificultad para identificar áreas de mejora es una de las afectaciones más frecuentes, indicando que la información no actualizada afecta en la toma de decisiones comerciales.

c. Importancia de información y tecnología

La preferencia por el nivel de ventas sugiere que la información detallada es más valiosa para la toma de decisiones comerciales.

La alta valoración del uso de la tecnología destaca una oportunidad para mejorar la gestión de la información a través de herramientas tecnológicas.

d. Utilidad de un aplicativo móvil

La mayor parte de los asesores comerciales destaca el uso de un aplicativo móvil, lo que sugiere una receptividad hacia soluciones digitales para la gestión de presupuestos.

La alta favorabilidad indica que un aplicativo móvil pueden ser una herramienta efectiva para mejorar la rapidez y facilidad en el acceso a la información.

Con el análisis de los datos obtenidos se procesa los siguientes resultados:

- La limitada cantidad de información causa dificultades e influye directamente en el cumplimiento de los objetivos planteados en la asignación de presupuestos, esto impide una correcta toma de decisiones e identificación de áreas de mejora que permita cumplir los objetivos comérciales.

- La entrega de información se realiza por medio de correos electrónicos causando limitantes que impactan en el cumplimiento efectivo de los presupuestos en ventas.

- La tecnología es un medio importante que impacta favorablemente en la mayoría de los asesores comerciales, por lo cual se considera adecuado usar esta herramienta para tener un mejor manejo en la gestión de los presupuestos en ventas.

- Los asesores comerciales de la Empresa Rectima Industry Cia. Ltda. está de acuerdo en la implementación de un aplicativo móvil que gestione el cumplimiento

en presupuestos en ventas que permita tener un mejor control del cumplimiento de sus objetivos comerciales.

Para la entrevista se representa mediante un gráfico Figura 13 la secuencia de pasos a seguir para la planificación de presupuestos en ventas.

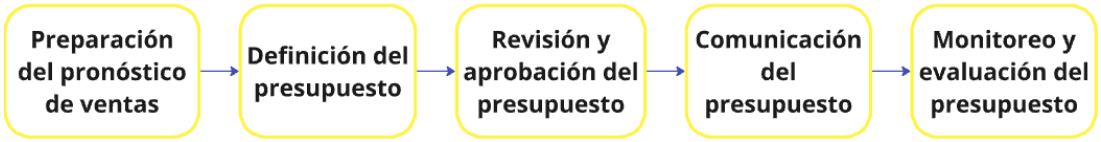


Figura 13. Secuencia de pasos planificación presupuestos en ventas

En base a los datos recolectados en la entrevista siguieren:

- La generación de los presupuestos en ventas se realiza a través de un grupo de personas que definen los presupuestos a través de un conjunto de procesos, el proceso seguido se divide en: preparación del pronóstico de ventas, definición del presupuesto, revisión y aprobación del presupuesto, comunicación del presupuestos y monitoreo y evaluación.

- El cumplimiento efectivo de los presupuestos concede la obtención de un incentivo que es calculado y es otorgado por medio de porcentajes que va desde un 95% a un 100% en cumplimiento según las distintas escalas asignadas en criterio de la Empresa.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados

En esta sección, se examinan las relaciones entre los datos, se discuten las implicaciones teóricas de los resultados, identificando los beneficios de algunos Frameworks para el desarrollo del proyecto, así como el proceso involucrado para la generación del presupuesto en ventas.

3.1.1 Proceso de planificación de presupuesto en ventas

Este proceso es un componente esencial de la gestión empresarial que implica la previsión y asignación de recursos financieros para alcanzar los objetivos de ventas. Se desarrollan estrategias de ventas y se asignan los recursos necesarios para implementarlas, creando un presupuesto detallado que incluye previsiones de ingresos y gastos. Este proceso también involucra el monitoreo y control continuo para asegurar que las actividades de ventas se mantengan alineadas con los objetivos comerciales, permitiendo ajustes oportunos y evaluaciones finales para mejorar futuras planificaciones.

Con los datos obtenidos en el Capítulo 2, se puede plasmar el proceso seguido actual en la Empresa Rectima Industry Cia. Ltda. para la planificación de presupuestos en ventas, este proceso es descrito a detalle medio un diagrama de procesos presentado en la Figura 14.

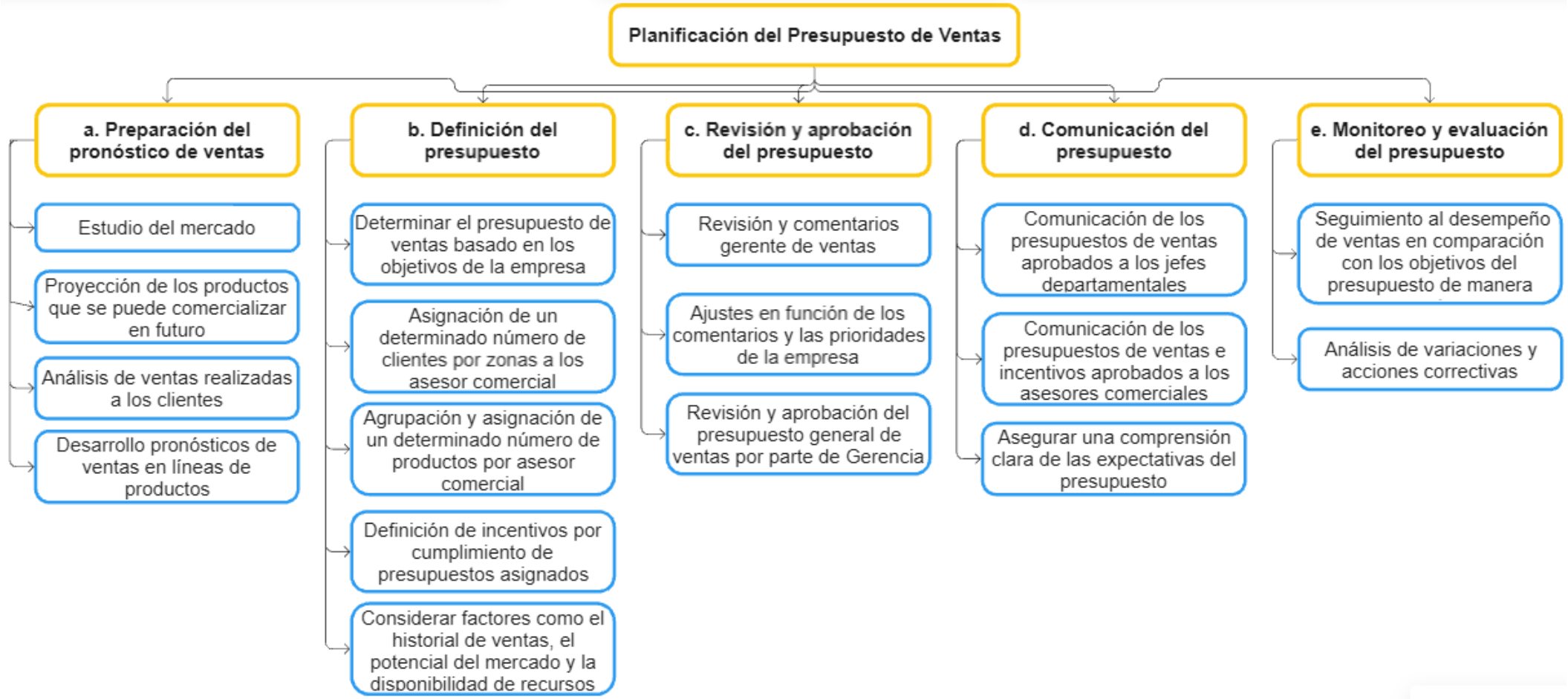


Figura 14. Proceso planificación de presupuestos

En la Figura 14, se presenta el proceso seguido para el desarrollo del presupuesto en ventas en la Empresa detallado a continuación:

a. Preparación del pronóstico de ventas

- **Estudio del mercado:** proceso usado para la recolección, análisis e interpretación del mercado objetivo.

- **Proyección de los productos que se puede comercializar en futuro:** proceso usado para prever y planificar los productos que tendrán éxito en el mercado.

- **Análisis de ventas realizadas a los clientes:** proceso usado para examinar y evaluar los datos de las ventas por cliente, la efectividad y rendimiento de las ventas a los clientes.

- **Desarrollos pronósticos de ventas en líneas de productos:** proceso usado para anticipar la demanda futura y realizar planificaciones para las ventas de productos futuras.

b. Definición del presupuesto

- **Determinar el presupuesto de ventas basado en los objetivos de la empresa:** estrategia usada para establecer proyecciones presupuestarias para la guía de estrategias de ventas.

- **Asignación de un determinado número de clientes por zonas a cada asesor comercial:** estrategia usada para optimizar la eficiencia del equipo de ventas.

- **Agrupación y asignación de un determinado número de productos por asesor comercial:** estrategia usada empleada para maximizar la eficiencia de las ventas y garantizar que los asesores comerciales manejen eficientemente la cartera de productos.

- **Definición de incentivos por cumplimiento de presupuestos asignados:** estrategia usada para motivar a los asesores comerciales a alcanzar y superar sus objetivos.

- **Considerar factores como el historial de ventas, el potencial del mercado y la disponibilidad de recursos:** estrategia usada para una toma de decisiones mas informa y efectiva por parte de la empresa.

c. Revisión y aprobación del presupuesto

- **Revisión y comentarios gerente de ventas:** proceso usado para garantizar la efectividad de las estrategias de ventas.

- **Ajustes en función de los comentarios y las prioridades de la empresa:** proceso usado para asegurar que las estrategias de ventas estén alineadas con los objetivos generales.

- **Revisión y aprobación del presupuesto general de ventas por parte de Gerencia:** proceso usado para garantizar que la proyección inicial este alineada con los objetivos definidos.

d. Comunicación del presupuesto

- **Comunicación de los presupuestos de ventas aprobados a los jefes departamentales:** proceso usado para garantizar que todas las áreas estén alineadas con los objetivos comerciales.

- **Comunicación de los presupuestos de ventas e incentivos aprobados a los asesores comerciales:** proceso usado para incentivar a los asesores comerciales a cumplir sus objetivos.

- **Asegurar una comprensión clara de las expectativas del presupuesto:** proceso usado para expresar de manera clara el presupuesto.

e. Monitoreo y evaluación del presupuesto

- **Seguimiento al desempeño de ventas en comparación con los objetivos del presupuesto de manera regular:** estrategia usada para evaluar si las metas establecidas son alcanzadas y realizar ajustes de ser necesario.

- **Análisis de variaciones y acciones correctivas:** proceso utilizado para identificar discrepancias entre el desempeño real y el desempeño proyectado.

3.1.2 Framework Ionic para el desarrollo móvil

El Framework Ionic es una plataforma de desarrollo de código abierto que ofrece un conjunto de herramientas para construir aplicaciones móviles avanzadas y eficientes. Esta plataforma aprovecha tecnologías web como HTML, CSS y JavaScript, y ofrece integraciones con marcos de desarrollo populares como Angular, React y Vue. Con Ionic, es posible la creación de aplicaciones móviles de alta calidad que funcionan sin problemas en una variedad de dispositivos y sistemas operativos. [22]

Se enfoca en mejorar la experiencia del usuario en el frontend de una aplicación, incluyendo controles de interfaz, interacciones, gestos y animaciones. Es una herramienta fácil de aprender que se puede integrar con otros marcos o bibliotecas populares como Angular, React o Vue. Además, también se puede utilizar de forma independiente, sin necesidad de utilizar ningún marco frontend, simplemente incluyendo un script.

a. Beneficios del Framework Ionic

Ionic ha permitido a diversas empresas alcanzar el éxito en el desarrollo de aplicaciones móviles, como lo evidencian los casos de Sworkit, MarketWatch y JustWatch.

Sworkit, una popular aplicación de fitness, ha utilizado Ionic para ofrecer rutinas de entrenamiento personalizadas a millones de usuarios, aprovechando la capacidad de la plataforma para mantener una única base de código para iOS y Android reducido los tiempos de desarrollo y así como los costos.

MarketWatch, una importante fuente de información financiera, ha mejorado su capacidad para entregar datos en tiempo real a través de una aplicación móvil optimizada con Ionic, proporcionando a sus usuarios acceso inmediato a noticias y análisis del mercado.

JustWatch ha utilizado Ionic para desarrollar una aplicación que facilita la búsqueda y el descubrimiento de contenidos en múltiples servicios de streaming, gracias a la capacidad de la plataforma para ofrecer una experiencia de usuario uniforme y rápida en diferentes dispositivos.

Estos casos de uso resaltan los beneficios que el Framework Ionic ofrece para el desarrollo de aplicaciones móviles y web gracias a sus características, al ser una plataforma de código abierto, proporciona la libertad de modificar y mejorar el código según se requiera. Al contar con conocimientos en tecnologías web, la curva de aprendizaje es baja debido a la utilización de HTML, CSS y JavaScript dentro del Framework. Ionic incorpora características tanto de Angular, React y Vue que permiten desarrollar aplicaciones móviles de manera eficiente, cuenta con la capacidad de crear aplicaciones para iOS, Android y la Web utilizando una única base de código permitiendo optimiza el tiempo y los recursos de desarrollo, cuenta con una comunidad de desarrolladores activa y una amplia variedad de recursos en su documentación en línea, lo que facilita la resolución de problemas y el aprendizaje continuo y gracias a los plugins de Cordova y Capacitor, Ionic proporciona acceso a las características nativas del dispositivo, ampliando las capacidades de la aplicación.

b. Versiones de Ionic

La Tabla 7 muestra un análisis comparativo de las versiones existentes del Framework con la finalidad de poder seleccionar la versión que mejor se adapte a las necesidades del proyecto de investigación.

Tabla 7. Versiones Ionic

Versión	Fecha de lanzamiento	Fin Soporte	Estado	Versión React Requerida	Ventajas	Desventajas	Soporte para plataformas	Rendimiento	Curva de aprendizaje	Documentación	Licencia
Ionic 6	Diciembre 8, 2021	Marzo 29, 2024	Fin Soporte	v17+	Integración de Capacitor, Web Components y Rendimiento Mejorado	Mayor curva de aprendizaje, aumento del tamaño de la aplicación y falta de soporte	Android 5.0+ y iOS 13.0+	Bueno	Avanzado	Completa	MIT
Ionic 7	Marzo 29, 2023	Abril 17, 2025	Mantenimiento	v17+	Implementación Capacitor Core, Rollup y Mejora en la interfaz de usuario	Posibles cambios en la API y posibles actualizaciones de dependencias	Android 5.1+ y iOS 14.0+	Muy Bueno	Avanzado	Completa y Actualizada	MIT
Ionic 8	Abril 17, 2024	TBD	Activo	v17+	Integración con Stencil, Mejora en TypeScript	Mayores dependencias Stencil, posible complejidad	Android 5.1+ y iOS 15.0+	Muy Bueno	Avanzado	Completa y Actualizada	MIT

Después de analizar las características técnicas del Framework Ionic, se tomó la decisión de utilizar la versión Ionic 7. Esta elección se basó en consideraciones como las ventajas y desventajas del Framework, la calidad de la documentación disponible, el rendimiento general, el soporte para múltiples plataformas y es la versión que mejor se adaptó a los requerimientos técnicos del proyecto, aunque Ionic 7 puede tener una curva de aprendizaje avanzada, también ofrece una amplia disponibilidad de componentes visuales, mejoras significativas y una sólida compatibilidad con dispositivos móviles para su implementación. Además, se consideró que, a pesar de no ser la versión más reciente de Ionic, sigue siendo una opción robusta y estable para el desarrollo de aplicaciones móviles multiplataforma.

3.1.3 Marcos de trabajo usado para el desarrollo del proyecto de investigación

La Tabla 8 muestra un análisis de los marcos de trabajo disponibles para el Framework Ionic con la finalidad de poder seleccionar el que mejor se adapte a las necesidades del proyecto de investigación.

Tabla 8. Marcos de trabajo

Marco de trabajo	**Lenguaje**	**Curva de Aprendizaje**	**Popularidad**	**Rendimiento**
Angular	TypeScript	Avanzado	Usado por: Microsoft, Google y IBM	Alto
React	TypeScript	Moderada	Usado por: Facebook, Netflix, PayPal y Uber	Alto
Vue	TypeScript	Avanzada	Usado por: Alibaba, Huawei y Tencent	Alto

Se optó por utilizar React como el marco de trabajo para el desarrollo en conjunto con el Framework Ionic, principalmente debido a su integración con TypeScript, esto podía representar un desafío adicional en términos de curva de aprendizaje, pero se consideró que React por ser un ecosistemas más grande y maduro en componentes reutilizable, además al ser un Framework diseñada específicamente para el desarrollo de aplicaciones móviles, se presentó como la opción más beneficioso para el desarrollo del proyecto.

3.1.4 Framework Node.js para el desarrollo de Web Services

Node.js es un entorno JavaScript de código abierto y multiplataforma que ha ganado popularidad en una amplia gama de proyectos. En Node.js, las aplicaciones se ejecutan en un solo proceso, evitando la creación de un nuevo hilo para cada solicitud. Además, Node.js ofrece una biblioteca estándar con primitivas asíncronas que permiten que el código JavaScript se ejecute sin bloquear, lo que significa que las operaciones para la trasferencia de un conjunto de datos (E/S) y las llamadas a funciones externas no detienen la ejecución del programa. En general, las bibliotecas de Node.js están diseñadas utilizando paradigmas no bloqueantes, lo que hace que el bloqueo del código sea poco común. [23]

El servidor está programado para escuchar en un puerto y nombre de host especifico. Una vez que el servidor está activo, se invoca una función de devolución de llamada que informa el estado del servidor. Cada vez que se recibe una nueva solicitud, se desencadena el evento "request", el cual proporciona dos objetos: una solicitud y una respuesta. Estos dos objetos son fundamentales para manejar las llamadas HTTP.

a. Beneficios del Framework Node.js

Node.js ha trascendido en el éxito de aplicaciones de alto perfil como Netflix, LinkedIn y PayPal.

Netflix utiliza Node.js para gestionar su gran volumen de tráfico y mejorar la eficiencia de su plataforma de streaming, permitiendo tiempos de carga más rápidos y una experiencia de usuario fluida con la capacidad de Node.js para manejar operaciones en tiempo real.

LinkedIn ha integrado Node.js en su backend para optimizar el manejo de grandes volúmenes de datos y ofrecer un rendimiento ágil en su red social profesional. Esto ha permitido facilitar una mejor navegación.

PayPal ha empleado Node.js para mejorar la velocidad y la eficiencia de sus servicios de pago en línea, permitiendo el manejar un alto número de transacciones simultáneas

con baja latencia, mejorando la experiencia del usuario y acelerando el tiempo de desarrollo.

Con la revisión de los casos de uso éxitos de Node.js se plasman los beneficios que lo convierten en una opción atractiva para el desarrollo de aplicaciones modernas. Su rendimiento y escalabilidad son destacados debido a su arquitectura basada en eventos y su motor de JavaScript, que permite manejar un gran número de conexiones simultáneas. La facilidad de aprendizaje es clave, ya que al tener conocimiento en el manejo de JavaScript la curva de aprendizaje se reduce, el ecosistema rico y extenso de Node.js, alimentado por el gestor de paquetes npm, proporciona acceso a una vasta colección de módulos y herramientas que facilitan el desarrollo. La plataforma también permite un desarrollo rápido, gracias a su modelo de programación asíncrono y no bloqueante, especialmente beneficioso para aplicaciones en tiempo real, como chats, plataformas de colaboración o informativas, donde la velocidad y la capacidad de respuesta son cruciales, además del bajo consumo de recursos, lo que ayuda a reducir los costos operativos y mejorar el rendimiento en entornos con recursos limitados. La comunidad grande y activa de Node.js garantiza un constante flujo de soporte, actualizaciones y mejoras, facilitando la resolución de posibles inconvenientes que se puedan producir en el desarrollo.

b. Versiones de Node.js

La Tabla 9 muestra un análisis comparativo de las versiones existentes del Framework con la finalidad de poder seleccionar la versión que mejor se adapte a las necesidades del proyecto de investigación.

Tabla 9. Versiones Node.js

Versión	Fecha de lanzamiento	Estado	Versión npm Requerida	Ventajas	Desventajas	Soporte para plataformas	Rendimiento	Curva de aprendizaje	Documentación	Licencia
Node.js 20	Abril 18, 2023	Soporta a largo plazo estable (LTS)	v10.5.0	Implementación de Capacitor Core que reduce el tamaño de la aplicación.	Posible impacto en la compatibilidad con algunas APIs nativas.	Android, iOS, Web, Windows, macOS	Muy Bueno	Moderada	Buena	MIT
Node.js 21	Octubre 17, 2023	Ultimas actualizaciones (Current)	v10.5.0	Introducción de Capacitor Plugin, simplificando la integración de funcionalidades nativas.	Posible aumento en el número de plugins.	Android, iOS, Web, Windows, macOS	Muy Bueno	Moderada	Buena	MIT
Node.js 22	Abril 24, 2024	Ultimas actualizaciones no estables (Current)	v10.5.1	Compilador de componentes web que permite crear componentes más eficientes.	Mayor dependencia de Stencil.	Android, iOS, Web, Windows, macOS	Muy Bueno	Avanzado	En desarrollo	MIT

Después de evaluar las características técnicas del Framework Node.js, se optó por la versión 21 como la más adecuada para el proyecto. Esta decisión se basó en varios factores, como las ventajas y desventajas, la calidad de la documentación disponible, el rendimiento general y el soporte para múltiples plataformas, además de ser una versión actualizada, confiable y con soporte. Todos estos atributos permitieron que la versión seleccionada se ajuste mejor a los requisitos técnicos del proyecto. Aunque no era la versión más reciente de Node.js, se consideró una opción sólida y estable para el proyecto.

3.1.5 Entorno de desarrollo integrado para el desarrollo del proyecto

La elección del Entorno de Desarrollo Integrado (IDE) es un aspecto fundamental durante la fase inicial de la planificación de proyectos de desarrollo de software. Los IDEs mencionados en la tabla son los más populares para el desarrollo tanto móvil como web. No obstante, hay IDEs que se adaptan mejor a las necesidades específicas de ciertos proyectos o que simplemente son más ampliamente utilizados. A continuación, se proporciona en la Tabla 10 una comparación que detalla las principales características de distintos IDEs con el objetivo de poder seleccionar el que mejor se adapte a los requerimientos del proyecto de investigación.

Tabla 10. Entornos de desarrollo

IDE	**Fecha de lanzamiento**	**Plataforma**	**Lenguajes de programación compatibles**	**Características destacadas**	**Ventajas**	**Desventajas**
Visual Studio Code (VS Code)	Febrero de 2015	Multiplataforma (Windows, macOS, Linux)	Más de 70, incluyendo JavaScript, Python, C++, Java, C#, PHP, Go, y muchos más	• Código abierto y gratuito • Extensible con una gran cantidad de extensiones • Interfaz personalizable • Integración con Git y otros sistemas de control de versiones • Depuración avanzada	• Ligero y rápido • Ideal para principiantes y desarrolladores ocasionales • Gran comunidad de usuarios • Amplia gama de extensiones disponibles	• Algunas extensiones pueden ralentizar el IDE • Menos funcionalidades integradas que otros IDEs Curva de aprendizaje para usar algunas extensiones complejas
WebStorm	Octubre de 2010	Multiplataforma (Windows, macOS, Linux)	JavaScript, HTML, CSS, TypeScript, Node.js, React, Angular, Vue.js, Java, Python, PHP, Ruby, C++, y muchos más	• Potentes herramientas de desarrollo web • Soporte para Frameworks como React, Angular y Vue.js • Refactorización y análisis de código avanzados	• Entorno de desarrollo web completo con herramientas integradas • Ideal para desarrolladores web profesionales • Soporte para Frameworks modernos y populares	• Licencia de pago puede ser costosa para algunos usuarios • Puede ser abrumador para principiantes debido a la gran cantidad de funcionalidades
IntelliJ IDEA	Enero de 2000	Multiplataforma (Windows, macOS, Linux)	Java, Kotlin, Scala, Groovy, C++, C#, Python, JavaScript, PHP, Ruby, y muchos más	• Amplio conjunto de herramientas para el desarrollo de Java • Potentes capacidades de refactorización y análisis de código	• Entorno de desarrollo Java completo con potentes herramientas • Ideal para desarrolladores Java profesionales	• Licencia de pago puede ser costosa para algunos usuarios • Puede ser pesado y consumir muchos recursos del sistema

La selección de Visual Studio Code (VS Code) como el entorno de desarrollo integrado para proyectos se fundamenta en su versatilidad y ventajas. VS Code es multiplataforma, funcionando en Windows, macOS y Linux, lo que garantiza flexibilidad en el entorno de desarrollo. Soporta más de 70 lenguajes de programación, incluidos JavaScript, Python, C++, Java, C# y PHP, adaptándose a diversas necesidades de desarrollo. Su naturaleza de código abierto y gratuita fomenta una gran comunidad que contribuye a su mejora constante y ofrece una amplia gama de extensiones para personalizar y ampliar el IDE. Con una interfaz personalizable, un rendimiento ligero y rápido, proporciona una experiencia de desarrollo eficiente y accesible tanto para principiantes como para desarrolladores experimentados. Estos factores hacen de VS Code una herramienta ideal para optimizar el flujo de trabajo tanto en la construcción de aplicativos Web como Móviles, esta última desarrollo del presente proyecto.

3.1.6 Gestor de base de datos

La elección de SQL Server como base de datos para el desarrollo del proyecto se justifica dado que SQL Server ha sido una plataforma confiable y robusta que se ha utilizado de manera consistente en todos los sistemas dentro de la Empresa Rectima Industry Cia. Ltda. durante muchos años.

Además, SQL Server ofrece un conjunto completo de herramientas y características que contribuyeron en las necesidades específicas del proyecto. Su escalabilidad permitió manejar grandes volúmenes de datos y garantizar un rendimiento óptimo en el aplicativo. La seguridad integrada y las capacidades de cumplimiento normativo brindaron tranquilidad en la protección de datos críticos y cumpliendo con los requisitos de regulación.

Otro factor importante es el soporte técnico y la comunidad de usuarios que respaldan SQL Server. Con un amplio conjunto de recursos en línea y documentación detallada, cualquier problema que pudo surgir fue abordado de manera efectiva y oportuna.

3.1.7 Metodologías de desarrollo móvil

Las metodologías ágiles son altamente valoradas en el desarrollo móvil debido a su flexibilidad y capacidad para adaptarse a un entorno de mercado en constante evolución. Estas metodologías ofrecen un enfoque iterativo que permite a los equipos responder de manera ágil a cambios en las expectativas de los usuarios, nuevas tecnologías y requisitos de los sistemas operativos móviles. En este contexto, se presentan en la Tabla 11 diversas metodologías ágiles para el desarrollo de proyectos, con el fin de seleccionar una metodología adecuada que se adapte al desarrollo del presente proyecto.

Tabla 11. Metodologías de desarrollo

Metodologías	**Fases**	**Magnitud del proyecto**	**Tamaño de equipo**	**Roles**	**Características**	**Valor para el cliente**
Scrum	Dividido en esprints de 1 a 4 semanas. Cada sprint tiene un objetivo específico.	Pequeño a grande.	5 a 9 miembros.	Scrum Master: Facilita el proceso Scrum. Product Owner: Define la visión del producto y prioriza las funcionalidades. Team: Desarrolla el producto.	Entrega incremental: El producto se entrega en partes funcionales al final de cada sprint. Adaptación al cambio: Se pueden realizar cambios a los requisitos durante el proyecto. Trabajo en equipo: Promueve la colaboración y comunicación entre los miembros del equipo.	Entrega temprana de funcionalidades: El cliente puede usar el producto y dar retroalimentación desde el principio. Mayor satisfacción del cliente: El cliente está más involucrado en el desarrollo del producto.
Extreme Programming (XP)	Ciclos de desarrollo cortos con fases de planificación, diseño, codificación y pruebas.	Pequeño a mediano.	4 a 12 miembros.	Coach: Guía al equipo en las prácticas de XP. Cliente: Define los requerimientos del producto. Desarrolladores: Desarrollan el producto. Tester: Testeo del producto. Gestor: gestor del proyecto.	Entrega de valor temprano: Se busca entregar valor al cliente lo antes posible. Calidad: Se pone énfasis en la calidad del código y las pruebas. Comunicación: Se fomenta la comunicación constante entre el equipo y el cliente.	Producto de alta calidad: El cliente obtiene un producto que cumple con sus expectativas.

Lean	Enfoque en el flujo de trabajo continuo, sin fases definidas.	Pequeño a grande.	Variable, no definido	Gerente de proyecto: Lidera el proyecto. Ingenieros: Desarrollan el producto. Equipo de soporte: Brinda soporte al equipo.	Eliminación de desperdicios: Se busca identificar y eliminar actividades que no agregan valor. Flujo de trabajo continuo: Se trabaja en un flujo continuo de desarrollo, sin interrupciones. Entrega rápida: Se busca entregar el producto lo antes posible.	Producto que se adapta a sus necesidades: El producto se puede adaptar a las necesidades cambiantes del cliente.
Mobile-D	Exploración, Inicialización, Producción, Estabilización y Prueba y fin del sistema.	Pequeño a mediano.	5 a 9 miembros.	No definido	Entrega incremental: El producto se entrega en partes funcionales al final de cada sprint. Adaptación al cambio: Se pueden realizar cambios a los requisitos durante el proyecto. Trabajo en equipo: Promueve la colaboración y comunicación entre los miembros del equipo.	Entrega rápida de un producto de calidad: El cliente obtiene un producto de calidad rápidamente.
Kanban	Solicitud de tareas. Elección de tareas. Desarrollo Pruebas Finalización	Pequeño a grande.	Variable, no definido	No definido	Flujo de trabajo continuo: Se trabaja en un flujo continuo de desarrollo, sin interrupciones. Entrega rápida: Se busca entregar el producto lo antes posible. Enfoque en el cliente: Se pone énfasis en satisfacer las necesidades del cliente.	Entrega rápida de un producto que satisface sus necesidades: El cliente obtiene un producto que cumple con sus expectativas rápidamente.

La selección de la metodología Mobile-D para el proyecto se fundamenta en su estructura y características adaptadas a proyectos de magnitud pequeña a mediana. Mobile-D proporciona fases claramente definidas y una entrega incremental, permitiendo que el producto se entregue en partes funcionales al final de cada sprint, lo que facilita la validación continua y la adaptación rápida a cambios. Su enfoque en la adaptación al cambio permite ajustar los requisitos durante el desarrollo, mientras que la promoción del trabajo en equipo asegura una colaboración eficaz y una comunicación fluida entre los miembros del equipo, esta metodología ofrece valor al cliente al proporcionar una entrega rápida de un producto de calidad, garantizando que el cliente obtenga un producto funcional y refinado en intervalos regulares. Esta combinación de flexibilidad, colaboración y entrega continua de valor hace de Mobile-D la opción ideal para el desarrollo del presente proyectos que requiere una gestión eficiente y una respuesta ágil a las necesidades.

El propósito de la metodología Mobile-D es proporcionar un marco de trabajo ágil específicamente diseñado para el desarrollo de aplicaciones móviles. Busca optimizar el proceso de desarrollo al adaptar las prácticas ágiles a los desafíos y requisitos únicos del entorno móvil. Mobile-D se centra en la entrega temprana y frecuente de funcionalidades, la colaboración cercana entre los miembros del equipo y los stakeholders, la retroalimentación continua de los usuarios y la capacidad de respuesta a los cambios en el mercado y las tecnologías móviles. Su objetivo es mejorar la eficiencia, la calidad y la satisfacción del cliente en el desarrollo de aplicaciones móviles.

a. Fase de la metodología Mobile-D

- **Exploración:** En esta fase inicial, se define el alcance del proyecto, los objetivos comerciales y las necesidades de los usuarios. Se realizan estudios de mercado, se identifican las funcionalidades clave y se crean prototipos de baja fidelidad para visualizar la interfaz. Se establece el presupuesto, el cronograma y los criterios de éxito del proyecto.

- **Inicialización:** Se establece el entorno de desarrollo, se definen los roles y responsabilidades del equipo y se planifica la primera iteración de desarrollo. Se seleccionan las herramientas y tecnologías, se crea un plan detallado para la

primera iteración, se definen los procesos de control de versiones y gestión de errores, se establecen métricas para medir el progreso y se capacita al equipo en las herramientas y metodologías a utilizar.

- **Producción:** El desarrollo se lleva a cabo en iteraciones cortas, implementando las funcionalidades clave en cada una. Se realizan pruebas unitarias y de integración de manera regular, se obtiene retroalimentación del equipo y de los usuarios, y se adapta el plan de desarrollo según sea necesario.

- **Estabilización:** Se corrigen errores, se mejora el rendimiento y se prepara la aplicación para su lanzamiento. Se realizan pruebas exhaustivas, se optimiza el rendimiento, se prepara la documentación y los materiales de capacitación, y se planifica la estrategia de lanzamiento y marketing.

- **Prueba y fin del Sistema:** Se realizan pruebas de aceptación final con usuarios reales para recopilar su retroalimentación e implementar las correcciones y mejoras finales. Se documenta el proyecto, se celebran los logros del equipo y se finaliza el proyecto.

3.2 Desarrollo de la propuesta

Para el desarrollo del proyecto se examinaron las diferentes fases de la metodología Mobile-D, que está enfocada en el desarrollo de aplicativos móviles en plazos cortos y con equipos reducidos.

3.2.1 Fase 1: Exploración

En la etapa inicial de exploración, se sentaron las bases para una comprensión integral del proyecto. Se identificaron y analizaron los interesados (stakeholders) claves, asegurando su involucramiento activo desde el principio. Se estableció con precisión el alcance del proyecto, definiendo claramente sus límites y objetivos. Además, se realizó una selección de las herramientas de desarrollo, asegurando que fueran las más adecuadas para las necesidades específicas del aplicativo.

a. Definición interesados (stakeholders)

Usuarios: Asesores comerciales.

Desarrollador: Equipo de trabajo para el desarrollo del aplicativo.

b. Definición alcance del proyecto

Para la definición del alcance del proyecto, se identificaron los procesos llevados a cabo para la planificación de presupuestos de ventas y la favorable aceptación del uso de un aplicativo móvil recopilados en el Capítulo dos, se levantaron los requerimientos funcionales que describen las funcionalidades específicas necesarias y se determinaron los requerimientos no funcionales que definen cómo debe comportarse el sistema, estableciendo estándares de rendimiento, usabilidad, compatibilidad, seguridad y escalabilidad. Esta integración permitió que el diseño del sistema se alineara con las expectativas de los usuarios finales y las metas estratégicas de la empresa, facilitando un flujo de trabajo efectivo y una experiencia de usuario óptima. Estos requerimientos son definidos en la Tabla 12 y Tabla 13.

- **Requisitos funcionales**

Tabla 12. Requerimientos funcionales

Número Requerimiento	**Requerimientos**	**Descripción**
Requerimientos del usuario		
RF1	Ingreso en el sistema	El usuario debe poder ingresar en el sistema
RF2	Registro en el sistema	El usuario debe poder registrarse en el sistema
RF3	Cambio de contraseña	El usuario debe poder cambiar su contraseña
RF4	Revisión información general	El usuario debe poder visualizar el valor de sus veneficios, bonificaciones y las principales promocionales
RF5	Revisión del progreso	El usuario debe poder visualizar en porcentajes la meta a ser alcanzada, el número de clientes por cada detalle, la cantidad que se lleva actualmente en número y en porcentaje

RF6	Revisión de los beneficios por monto	El usuario debe poder visualizar los montos calculados con el detalle del valor asignado y el valor cumplido actualmente, así como el bono ganado
RF7	Revisión de los beneficios por efectividad	El usuario debe poder visualizar los valores correspondientes a la efectividad, detalle 100% y efectivizado, así como el porcentaje alcanzado por cada apartado
RF8	Revisión de los beneficios por rentabilidad	El usuario debe poder visualizar la rentabilidad, comisión y porcentaje con el valor obtenido
RF9	Revisión de los beneficios por bono monto	El usuario debe poder visualizar su venta, presupuesto y el porcentaje alcanzado, así como el valor obtenido
RF10	Revisión de los beneficios por promocionales	El usuario debe poder visualizar total ve venta, valor presupuesto y porcentaje por promocional
RF11	Revisión de los beneficios por cartera	El usuario debe poder visualizar a detalle la efectividad, detalle mayor del 100% y la efectividad, así como el valor obtenido
RF12	Revisión valores totales	El usuario debe poder visualizar un reporte general con todos los valores calculados y porcentajes tanto de progreso actual en el cumplimento del presupuesto y progreso en obtención, estos valores se deben detallar de manera comprensible, los datos serán del avance actual por cada día
RF13	Revisión ventas a clientes	El usuario debe poder visualizar un listado de los clientes asignados y un reporte de los productos adquirido por cada uno de los clientes
Requerimientos del Sistema		
RF13	Cálculo de la efectividad	El sistema debe calcular la efectividad de cada asesor comercial, para lo cual se requiere calcular por número de clientes atendidos, por número de nuevos clientes y por clientes que realicen una recompra por cada asesor comercial
RF14	Cálculo de la rentabilidad	El sistema debe calcular la rentabilidad con el detalle objetivo, la rentabilidad actual y en caso de cumplir con el objetivo se le asigne al asesor un valore de comisión
RF15	Cálculo bonificación	El sistema debe calcular la bonificación para el cumplimiento en promocionales, se debe detallar el valor de la venta, el valor del presupuesto y el porcentaje alcanzado, si cumple con los criterios establecidos se asigne el valor de la comisión al asesor

RF16	Cálculo bonificación por cumplimiento	El sistema debe calcular la bonificación para el cumplimiento en bono por cartera se detalla por tres segmentos, por efectividad donde se calcule valor de cobranza, valor del presupuesto y porcentaje alcanzado, por detalle 100% se calcule el total de cobranza realizado, el valore de la cartere y el porcentaje alcanzado y por efectivizado donde se calcule el valor de la cobranza dentro de plazo, el valor fuera de plazo y el porcentaje alcanzado
RF17	Creación de clientes	El sistema debe permitir la creación de nuevos clientes, poder ingresar todos los datos requeridos del cliente y poder subir fotos de los documentos necesarios
RF18	Valor total de los veneficios	El sistema debe calcular el valor total de los beneficios que pueden llegar a obtener por cada asesor comercial

- **Requisitos no funcionales**

Tabla 13. Requerimientos no funcionales

Número Requerimiento	**Requerimientos**	**Descripción**
RNF1	Rendimiento	Tiempos de carga rápidos para una experiencia de usuario fluida
RNF2	Usabilidad	Interfaz de usuario intuitiva, fácil de usar y elementos de la interfaz con un tamaño adecuado para facilitar su interacción en dispositivos móviles
RNF3	Compatibilidad	Aplicación compatible con diferentes gamas de dispositivos móviles
RNF4	Seguridad	Protección contra acceso no autorizado, aautenticación para el acceso al aplicativo móvil y cifrado de contraseñas
RNF5	Escalabilidad	Diseñar de la arquitectura de la aplicación que permita la adaptación fácil a cambios en los requisitos y la inclusión de nuevas funcionalidades

c. Limitaciones del proyecto

El presente proyecto presenta las siguientes limitaciones:

- Para el uso efectivo del aplicativo móvil es necesario una conexión a internet.
- El aplicativo móvil es compatible con la versión 5.1 o superior en Android y el iOS la versión compatible es a partir de la 14.0 o superior.
- El aplicativo móvil tiene como objetivo principal informar los presupuestos en ventas y el progreso del cumplimiento actual del presupuesto en ventas.

d. Establecimiento de dependencias

El aplicativo móvil está destinado al uso de los asesores comerciales en cualquier región del Ecuador.

e. Establecimiento del proyecto de investigación

Para el desarrollo del proyecto de investigación se utilizaron las siguientes herramientas:

- **Ionic**

Framework que permite la creación de aplicativos móviles de fácil personalización.

- **React**

Framework que cuenta con una amplia gama de componentes de interfaz de usuarios que se adaptan a los estándares de diseño de plataformas móviles.

- **Node.js**

Framework que permite un desarrollo rápido y ágil de aplicativos móviles del lado del backend gracias a su arquitectura asíncrona y basada en eventos.

- **Capacitor**

Herramienta que permite aprovechar la familiaridad y versatilidad de la tecnología Web como HTML, CSS y JavaScript, proporciona acceso a las funcionalidades nativas del dispositivo móvil.

- **Cordova**

Herramienta de desarrollar de aplicaciones móviles, que permite utilizar tecnología web para crear aplicaciones multiplataforma con acceso a funcionalidades nativas del dispositivo.

- **SQL Server**

Herramienta que permite la gestión y almacenamiento de grandes volúmenes de datos, crucial para el manejo de la información confidencial de la Empresa.

- **Axios**

Herramienta flexible para realizar solicitudes HTTP en aplicaciones JavaScript.

- **Visual Studio Code**

Editor de código disponible para los principales sistemas operativos y permite la creación y gestión de proyectos móviles.

3.2.2 Fase 2: Inicialización

Se plantean los recursos necesarios para la ejecución del desarrollo y diseño del aplicativo móvil.

a. Configuración ambiente de desarrollo

Se detalla a continuación las herramientas usadas e instaladas para el desarrollo del aplicativo móvil:

- Framework: Ionic
- Lenguaje de programación: React
- Gestor base de datos: SQL Server
- Entorno de ejecución servidor: Node.js

b. Preparación ambiente de desarrollo

Se configuro e instalo las siguientes herramientas:

- Ionic v7.2.0
- React v18.2.0
- SQL Server v19.2.56.2
- Visual Studio Code v1.85.1
- Capacitor v5.6.0
- Cordova v4.0.0
- Axios v1.6.5
- Node.js v21.5.0
- Android Studio v17.0.6

c. Planificación fases

En la Tabla 14 se realiza la división del proyecto en etapas manejables y secuenciales que faciliten el manejo de un enfoque ágil, permitiendo la entrega incremental del proyecto.

Tabla 14. Planificación fases proyecto

Fases	**Ítems**	**Descripción**
Exploración	Iteración 0	Definición interesados Definición alcance del proyecto y limitaciones Establecimientos del proyecto y establecimiento del proyecto
Iniciación	Iteración 1	Configuración y preparación ambiente de desarrollo Elaboración del modelo del aplicativo móvil Elaboración de la base de datos Elaboración diagramas de uso Elaboración de prototipado de la aplicación
Producción	Iteración 2	Registro del usuario Desarrollo del módulo para el registro de los usuarios Registro storycard Pruebas funcionamiento
	Iteración 3	Inicio de sesión Desarrollo del módulo para el ingreso de los usuarios registrados Registro storycard Pruebas funcionamiento
	Iteración 4	Cambio de contraseña Desarrollo del módulo para el cambio de la contraseña por parte de los usuarios Ventana para ingreso de correo y generación de código Ventana validación de código enviado por correo Ventana para el ingreso de la nueva contraseña Registro storycard Pruebas funcionamiento
	Iteración 5	Inicio Desarrollo del módulo de bienvenida al aplicativo Vista de presupuesto en bonificaciones Vista publicidad en promocionales Registro storycard Pruebas funcionamiento
	Iteración 6	Progreso hoy Desarrollo de la vista del total en ventas, presupuesto, porcentaje alcanzado, cumplimiento, efectividad, rentabilidad, bonos y número de nuevos clientes Registro storycard Pruebas funcionamiento

	Iteración 7	Comisión monto Desarrollo de la vista con el detalle de la venta neta, presupuesto, valor comisión y porcentaje a cumplir Registro storycard Pruebas funcionamiento
	Iteración 8	Comisión efectividad Desarrollo de la vista con el detalle y valor alcanzado Clientes atendidos con el total, venta cliente y alcance Nuevo cliente con número clientes nuevos, venta y alcance Cliente recompra con numero de recompra, venta y alcance Registro storycard Pruebas funcionamiento
	Iteración 9	Comisión rentabilidad Desarrollo de la vista con el detalle del valor alcanzado, porcentaje y rentabilidad objetivo Registro storycard Pruebas funcionamiento
	Iteración 10	Bono monto Desarrollo de la vista con el detalle del valor a alcanzar, valor obtenido, venta, presupuesto y porcentaje cumplimiento Registro storycard Pruebas funcionamiento
	Iteración 11	Bono promocional Desarrollo de la vista con el detalle del porcentaje de comisión, valor obtenido, venta promoción y presupuesto promoción Registro storycard Pruebas funcionamiento
	Iteración 12	Bono cartera Desarrollo de la vista con el detalle de la efectiva cobranza y cumplimiento presupuesto Por detalle mayor al 100% con la venta cobrada y valor cartera Por detalle de efectivizado con la cobranza en plazo y cobranza fuera de plazo Registro storycard Pruebas funcionamiento
	Iteración 13	Detalle total incentivos Desarrollo de la vista con el total de los montos alcanzados en comisiones y bonos.

		Detalle del monto, efectividad, rentabilidad, montos y el total alcanzado Descarga de archivo Pdf con los valores obtenidos durante el mes Registro storycard Pruebas funcionamiento
	Iteración 14	Creación clientes Desarrollo del módulo para la creación de nuevos clientes, ingreso de datos y documentación del cliente Registro storycard Pruebas funcionamiento
	Iteración 15	Ventas clientes/productos Desarrollo del módulo para la revisión y vista de clientes asignados y cantidad de productos adquiridos por cada uno Registro storycard Pruebas funcionamiento
Estabilización	Iteración 16	Corrección errores en el código desarrollado
	Iteración 17	Corrección errores interfaz de usuario
Pruebas	Iteración 18	Pruebas aplicativo móvil y comprobación correcto funcionamiento según los criterios definidos

d. Diseño base de datos

SQL Sever es un sistema gestor de base de datos relacional con la capacidad de manejar grandes volúmenes de datos eficientemente con un conjunto de herramientas administrativas que agilizan su gestión.

- **Base de datos**

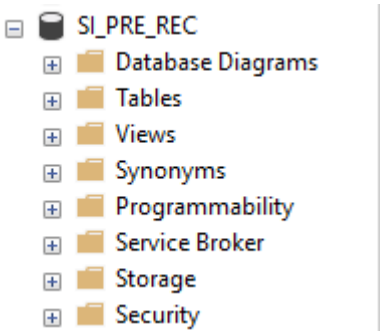


Figura 15. Base de datos en SQL Server

- **Estructura base de datos SGRGP**

Se presenta la estructura de cada tabla creada en la base de datos SQL Server.

Tabla 15. Presupuesto mensual

Campo	**Tipo dato**	**Descripción**
PRE_VenMen_ID	number	Almacenar el registro del valor único que identifica la venta mensual
PRES_Vendedor_ID	number	Almacenar el registro del código único del asesor
MES_ID	number	Almacenar el registro del mes de ingreso
PRES_M_VendedorInicial	decimal	Almacenar el registro del valor del presupuesto inicial
PRES_M_VendedorProyectado	decimal	Almacenar el registro del primer valor mensual proyectado
PRES_M_VendedorProyectado2	decimal	Almacenar el registro del segundo valor mensual proyectado
PRES_M_VendedorProyectado3	decimal	Almacenar el registro del tercer valor mensual proyectado

Tabla 16. Presupuesto anual

Campo	**Tipo dato**	**Descripción**
PRES_Vendedor_ID	number	Almacenar el registro del valor único que identifica el presupuesto en venta anual
ANIO_ID	number	Almacenar el registro del código único del asesor
VEN_Codigo	string	Almacenar el registro del mes de ingreso
PRES_Vendedor_Anual	decimal	Almacenar el registro del valor del presupuesto inicial
PRES_Vendedor_Anual_Proy	decimal	Almacenar el registro del primer valor anual proyectado
PRES_Vendedor_Anual_Proy2	decimal	Almacenar el registro del segundo valor anual proyectado
PRES_Vendedor_Anual_Proy3	decimal	Almacenar el registro del tercer valor anual proyectado

Tabla 17. Clientes

Campo	Tipo dato	Descripción
CLI_RUC	string	Almacenar el registro del ruc del cliente
CLI_Nombre	string	Almacenar el registro del nombre del cliente
CLI_RepresLegal	string	Almacenar el registro de la representación legal del cliente
CLI_Direccion	string	Almacenar el registro de la dirección del cliente
CLI_Telefono	string	Almacenar el registro del teléfono del cliente
CLI_CondicionPago	string	Almacenar el registro de la condición de pago del cliente
CLI_NivelPrecio	string	Almacenar el registro del nivel de precios por cliente
CLI_Cupo	decimal	Almacenar el registro del cupo asignado al cliente
Clasif_Ventas	string	Almacenar el registro de la clasificación de ventas
Clasif_Cobranzas	string	Almacenar el registro de la clasificación de cobranza
VEN_Codigo	string	Almacenar el registro del código único del asesor
CLI_Tipologia	string	Almacenar el registro de la tipología del cliente
VEN_Nombre	string	Almacenar el registro del nombre del asesor

Tabla 18. Productos

Campo	Tipo dato	Descripción
PROD_ID	string	Almacenar el registro del valor único que identifica cada producto
PROD_ID_N	string	Almacenar el registro del código asignado a cada producto
PROD_Nombre	string	Almacenar el registro del nombre de cada producto
PROD_Cantidad	number	Almacenar el registro de la cantidad presente por producto
PROD_CIF	decimal	Almacenar el registro del valor de cada cifra del producto
LIN_ID	number	Almacenar el registro del código línea del producto
LIN_ID_N	string	Almacenar el registro del número línea del producto

PROD_Marca	string	Almacenar el registro de la marca del producto
MarcaVehi	string	Almacenar el registro de la marca vehículo que pertenece el producto
TipoABC	string	Almacenar el registro del tipo de ABC de producto
PROD_Amortiguador	number	Almacenar el registro por amortiguador
Disponibilidad	string	Almacenar el registro de la disponibilidad del producto
cantDisponibilidad	number	Almacenar el registro de la cantidad disponibilidad del producto
stock	number	Almacenar el registro del total existencia disponible
Familia	string	Almacenar el registro de la familia a la que pertenece el producto
LinCompras	string	Almacenar el registro de la línea adquirida de cada producto
FAM_ID	number	Almacenar el registro del código por familia del producto
comprador	string	Almacenar el registro la referencia de la compra del producto

Tabla 19. Promocional por mes

Campo	**Tipo dato**	**Descripción**
PRES_PROMOCION_ID	number	Almacenar el registro de código único que identifica la promocional
ANIO_ID	number	Almacenar el registro del año del ingreso de la promocional
MES_ID	number	Almacenar el registro del mes ingresado la promocional
PRES_Vendedor_Anual	decimal	Almacenar el registro del valor de la venta anual
PRES_Vendedor_Anual_Proy	decimal	Almacenar el registro del primer valor proyectado de la promocional por mes
PRES_Vendedor_Anual_Proy2	decimal	Almacenar el registro del segundo valor proyectado de la promocional por mes
PRES_Vendedor_Anual_Proy3	decimal	Almacenar el registro de tercer valor proyectado de la promocional por mes

Tabla 20. Promocional por familia

Campo	Tipo dato	Descripción
PRE_ProFa_ID	number	Almacenar el registro del código único de la promocional por familia
PRES_Promocion_ID	number	Almacenar el registro del código del presupuesto al que pertenece
Familia_NOM	string	Almacenar el registro del nombre de la familia
PRE_ProFaInicial	decimal	Almacenar el registro del valor promedio inicial de la promocional por familia
PRE_ProFaProyectado	decimal	Almacenar el registro del primer valor proyectado de la promocional por familia
PRE_ProFaProyectado2	decimal	Almacenar el registro del segundo valor proyectado de la promocional por familia
PRE_ProFaProyectado	decimal	Almacenar el registro de tercer valor proyectado de la promocional por familia

Tabla 21. Promocional por asesor

Campo	Tipo dato	Descripción
PRE_ProVen_ID	number	Almacenar el registro del código único de la promocional por asesor
PRES_Promocion_Ven_ID	number	Almacenar el registro del código del presupuesto en ventas al que pertenece
Familia_Nom_Ven	string	Almacenar el registro del nombre de la familia por asesor
PRE_ProFaInicial_Ven	decimal	Almacenar el registro del valor promedio inicial de la promocional por asesor
PRE_ProFaProyectado_Ven	decimal	Almacenar el registro del primer valor proyectado de la promocional por asesor
PRE_ProFaProyectado2_Ven	decimal	Almacenar el registro del segundo valor proyectado de la promocional por asesor
PRE_ProFaProyectado_Ven	decimal	Almacenar el registro de tercer valor proyectado de la promocional por asesor

Tabla 22. Datos nuevo cliente

Campo	Tipo dato	Descripción
ID	number	Almacenar el registro del código único del nuevo cliente
COD_VENDEDOR	string	Almacenar el registro del código de identificación del asesor
RUC_CED	string	Almacenar el registro del ruc o cédula del nuevo cliente
NUM_CARPETA	string	Almacenar el registro del número carpeta asociada al que pertenece el nuevo cliente
EXTRAN	number	Almacenar el registro en caso de ser un cliente del extranjero
NOM_COM	string	Almacenar el registro del nombre comercial asociado al nuevo cliente
RAZ_SOC	string	Almacenar el registro de la razón social del nuevo cliente
DIRECCION	string	Almacenar el registro de la dirección del nuevo cliente
TELEFONO	string	Almacenar el registro de número de teléfono del nuevo cliente
PAIS	string	Almacenar el registro del país al que pertenece el nuevo cliente
CIUDAD	string	Almacenar el registro de la ciudad a la que pertenece el nuevo cliente
PARROQUIA	string	Almacenar el registro de la parroquia a la que pertenece el nuevo cliente
PROVINCIA	string	Almacenar el registro de la provincia a la que pertenece el nuevo cliente
COR_ELEC	string	Almacenar el registro del correo electrónico del nuevo cliente
NOM_CONT	string	Almacenar el registro del nombre comercial del nuevo cliente
CUPO	string	Almacenar el registro del cupo seleccionado del nuevo cliente
PLAZO	string	Almacenar el registro del plazo seleccionado del nevo cliente
NIV_PRECIO	string	Almacenar el registro el nivel precio seleccionado por el nuevo cliente
RELACION	string	Almacenar el registro de la relación a la que pertenece el nuevo cliente

FECHA	datetime	Almacenar la fecha de ingreso del nuevo cliente
FECHA_AUT	datetime	Almacenar la fecha de autorización del nuevo cliente
USUARIO_AUT	string	Almacenar el registro del usuario que autoriza al nuevo cliente
ESTADO	number	Almacenar el estado del registro del nuevo cliente

Tabla 23. Documentos nuevo cliente

Campo	**Tipo dato**	**Descripción**
Id	number	Almacenar el registro del código único del documento del nuevo cliente
Id_INFO_CLI	number	Almacenar el registro de la clave foránea de la tabla de datos del nuevo cliente
RUC_CED	string	Almacenar el registro del ruc o cédula del nuevo cliente
D_CED	string	Almacenar la copia digital de la cédula de identidad
D_PAP_VOT	string	Almacenar la copia digital de la papeleta de votación
D_RUC	string	Almacenar la copia digital del ruc
D_P_SERV	string	Almacenar la copia digital un servició básico
D_DECLA	string	Almacenar la copia digital de la declaración juramentada
D_CERT_BANC	string	Almacenar la copia digital del certificado bancario
D_REF_PER	string	Almacenar la copia digital de una referencia personal
D_REF_COM	string	Almacenar la copia digital de una referencia comercial
D_LET_CAM	string	Almacenar la copia digital de la letra de cambio
D_DEC_AUT_FIR	string	Almacenar la copia digital de la autorización firmada
FECHA	datatime	Almacenar la fecha de registro del nuevo cliente
FECHA_AUT	datatime	Almacenar la fecha autorización registro nuevo cliente
USUARIO_AUT	string	Almacenar el registro del usuario que autoriza al nuevo cliente
ESTADO	number	Almacenar el estado del registro del nuevo cliente

Tabla 24. Bonificación por cartera

Campo	Tipo dato	Descripción
ID	number	Almacenar el registro del código único para la bonificación por cartera
FECHA	datetime	Almacenar la fecha de registro de la bonificación
COD_VENDEDOR	string	Almacenar el registro del código de identificación del asesor
NOM_VENDEDOR	string	Almacenar el registro del nombre del asesor
COBROS_NETO	decimal	Almacenar el registro del valor de la cobranza realizada
PRESUP_COBRO	decimal	Almacenar el registro del valor del presupuesto de la cobranza
EFECTI_COBRO	decimal	Almacenar el registro de la cobranza efectivizada
OBJET_EFECTI	decimal	Almacenar el registro del valor objetivo de la cobranza
BONO_EFECTI	decimal	Almacenar el registro del valor objetivo del bono
BONO_EFECTI_OBTEN	decimal	Almacenar el registro del valor del bono objetivo de la cobranza
VTS_MES_ANTES	decimal	Almacenar el registro del valor de las ventas del mes anterior
DIEZVTAS_ANTES	decimal	Almacenar el registro del valor de las ventas diez días anteriores
VTS_MES_ACTUAL	decimal	Almacenar el registro de la cantidad de la venta actual
TOTAL_VENTAS	decimal	Almacenar el registro de la cantidad de venta realizada
FACTURAS	decimal	Almacenar el registro de la cantidad de facturas ingresadas
CHQ_PROTESTADOS	decimal	Almacenar el registro de la cantidad de cheques protestados
CHQ_POST_VENCIDOS	decimal	Almacenar el registro de la cantidad de cheques vencidos
CARTERA_TOTAL	decimal	Almacenar el registro del valor de la cartera
DIFERENCIA	decimal	Almacenar el registro de la diferencia obtenida
PARTI_DEFICIT	decimal	Almacenar el registro del valor del déficit
CUMPLIM_CARTERA	decimal	Almacenar el registro del porcentaje de cumplimiento en cartera

OBJETIVO_CARTER	decimal	Almacenar el registro del porcentaje objetivo de la cartera
BONO_CARTERA	decimal	Almacenar el registro del valor del bono por cartera
BONO_CARTERA_OBTEN	decimal	Almacenar el registro del valor objetivo del bono por cartera
COBRO_DENTR_PLAZO	decimal	Almacenar el registro de la cantidad de cobranza dentro de plazo
COBRO_FUERA_PLAZO	decimal	Almacenar el registro de la cantidad de cobranza fuera de plazo
CUMPLIM_COBROS	decimal	Almacenar el registro del valor del cumplimiento por cobros
OBJETIVO_COBROS	decimal	Almacenar el registro del valor objetivo del cumplimiento de los cobros
BONO_COBROS	decimal	Almacenar el registro del valor del bono del cumplimiento por cobros
BONO_COBROS_OBTEN	decimal	Almacenar el registro del valor objetivo del bono por cobros

Tabla 25. Presupuesto efectividad

Campo	**Tipo dato**	**Descripción**
ID	number	Almacenar el registro del código único del presupuesto por efectividad
FECHA	datetime	Almacenar la fecha del registro
COD_VENDEDOR	string	Almacenar el registro del código de identificación del asesor
NOM_VENDEDOR	string	Almacenar el registro del nombre del asesor
CLIENT_ASIGN	number	Almacenar el registro de la cantidad de clientes asignado
CLIENT_ATEND	number	Almacenar el registro de la cantidad de clientes atendidos
EFECTIVIDAD	decimal	Almacenar el registro del valor de la efectividad

Tabla 26. Usuarios Aplicación

Campo	**Tipo dato**	**Descripción**
ID	number	Almacenar el registro del código único para el usuario
USUARIO	string	Almacenar el registro del usuario

NOMBRE	string	Almacenar el registro del nombre del usuario
DEPARTAMENTO	string	Almacenar el registro del departamento al que pertenece el usuario
CLAVE	string	Almacenar el registro de la contraseña del usuario
FECHA	datetime	Almacenar la fecha de registro del usuario
CORREO	string	Almacenar el registro del correo electrónico del usuario
ESTADO	number	Almacenar el estado del usuario si esta activo o no

Tabla 27. Provincias

Campo	**Tipo dato**	**Descripción**
PROV_ID	number	Almacenar el registro del código único de la provincia
CODIGO_PROV	number	Almacenar el registro del código de la provincia
NOMBRE_PROV	string	Almacenar el registro del nombre de la provincia

e. Diagrama casos de uso

Los diagramas de casos de uso tienen como objetivo comprender la interacción entre los usuarios y el sistema, enfocándose en cómo el sistema responde a las acciones del usuario. A continuación, en la Figura 14 y Figura 15 se presentan los diagramas de casos de uso que se han creado según el rol que desempeña cada usuario.

- **Asesor Comercial**

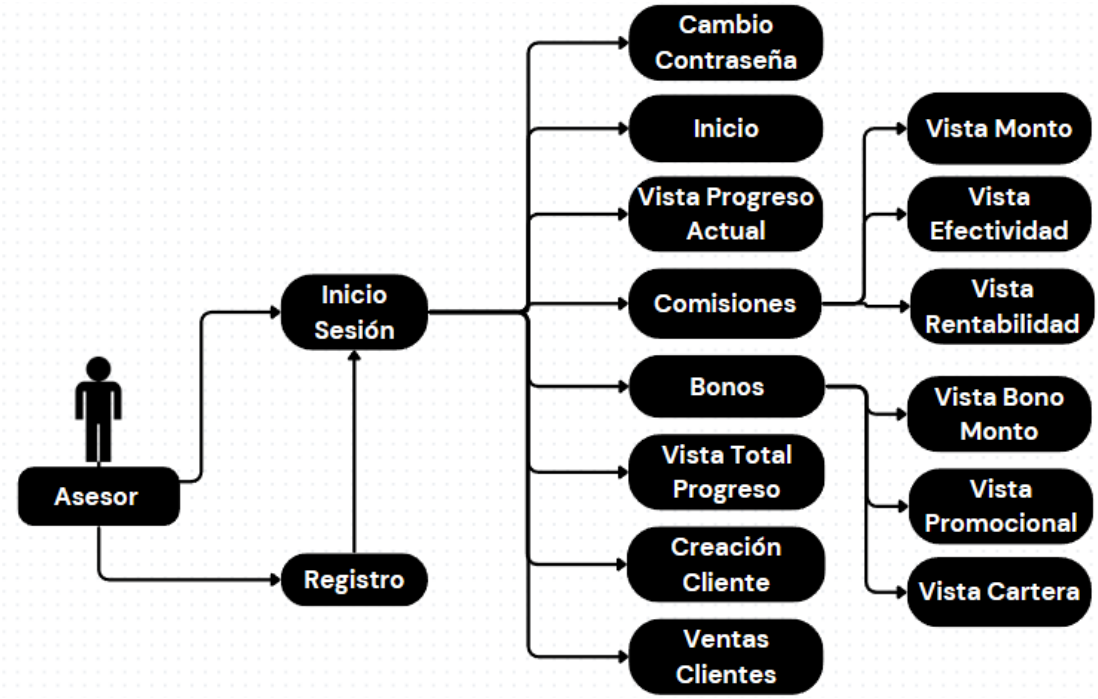


Figura 16. Diagrama caso de uso de los asesores comerciales

- **Personal Administrativo**

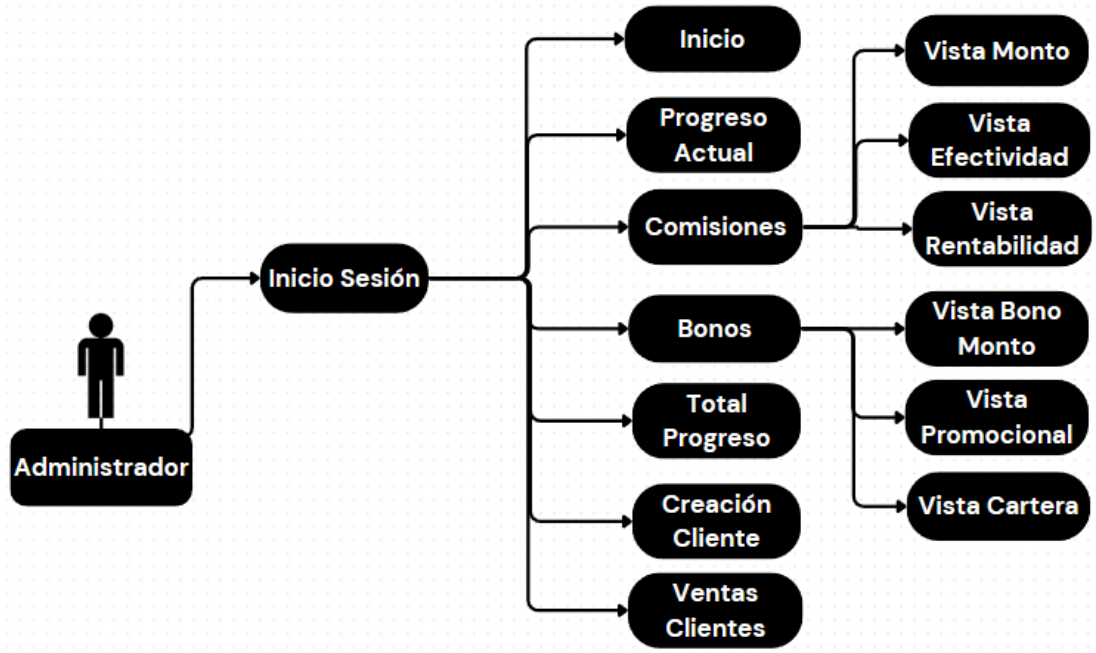


Figura 17. Diagrama caso de uso administrador

f. Prototipo aplicativo móvil

Los prototipos o maquetas son fundamentales en el proceso de desarrollo de aplicaciones móviles, ya que posibilitan la visualización previa del diseño y la funcionalidad de la aplicación antes de su implementación, con este enfoque se utilizó la herramienta Figma para la creación del prototipado. Esta práctica contribuye a asegurar que la aplicación satisfaga las necesidades y expectativas del usuario.

Prototipo para asesores comerciales.

- **Inicio Sesión**

En la interfaz de inicio de sesión Figura 18 el usuario puede realizar tres acciones posibles, iniciar sesión y acceder al sistema, realizar el registro como nuevo usuario y poder cambiar su contraseña en caso de olvido.

Figura 18. Inicio de sesión

- **Registro de usuario**

En la pantalla de registro de usuario Figura 19 el nuevo usuario podrá a través de su código podrá realizar una búsqueda de sus datos personales, los mismos se cargarán automáticamente al ingresar un código efectivo, se permite el ingreso de una nueva contraseña y el registro efectivo para el ingreso al aplicativo. Si el usuario no es parte de la Empresa no se podrá registrar al ser un aplicativo específico de la Empresa.

Figura 19. Registro Usuario

- **Cambio contraseña**

En la interfaz de cambio de contraseña Figura 20 se permite el restablecer una contraseña olvidada a los usuarios que se encuentren registrados en el sistema.

La primera acción permitida es ingresar el correo electrónico empresarial registrado en el aplicativo a donde se enviará un código de verificación y validades para realizar un cambio de contraseña.

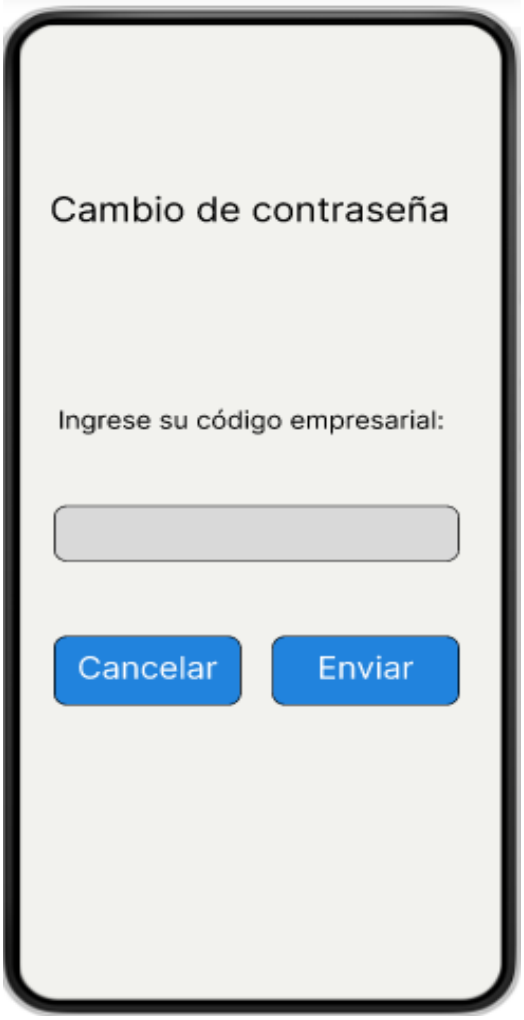


Figura 20. Envió mensaje cambio de contraseña

La segunda acción Figura 21 permite el ingreso del código generado por el aplicativo y enviado al correo del usuario que requiere el cambio de contraseña, se valida que sea correcto y se permite el avance a la siguiente acción.

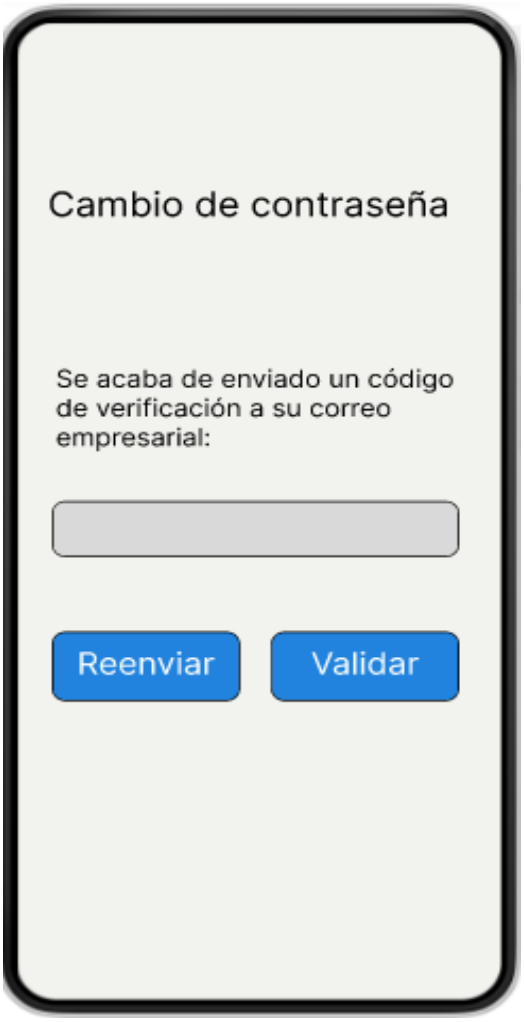


Figura 21. Ingreso y validación código

La tercera acción Figura 22 permite el ingreso de una nueva contraseña al usuario, una vez ingresada se guarda la nueva contraseña y se permite el ingreso al sistema con la nueva contraseña ingresada.

Figura 22. Ingreso nueva contraseña

- **Inicio**

En la interfaz de inicio Figura 23 se presenta la bienvenida al aplicativo móvil, valor a obtener por cumplimiento de objetivos y las principales promociones que se presentan como disponibles a aplicar para las ventas.

Figura 23. Inicio

- **Menú**

En el menú de navegación Figura 24 se presentan las distintas opciones disponibles con las que cuenta en aplicativo y permite el ingreso a las distintas interfaces disponibles.

Figura 24. Menú navegación

- **Progreso diario**

En esta interfaz Figura 25 se reflejan los valores obtenidos por cada usuario del progreso que se va obteniendo por cada día, el progreso está dividido por porcentaje alcanzado, valor del presupuesto asignado y venta actual, así como un detalle desplegado mediante una tabla del progreso en cumplimiento de la meta asignada.

Figura 25. Progreso diario

- **Comisión por bono**

En esta interfaz Figura 26 se detalla los valores y metas a cumplir por lo cual se puede llegar a obtener esta comisión, se divide en valor alcanzado, porcentaje de la comisión y un detalle desglosado mediante una tabla.

Figura 26. Comisión bono

- **Comisión por efectividad**

En esta interfaz Figura 27 se detalla la consigna por la cual se puede obtener una comisión por concepto de efectividad, se divide el valor alcanzado, porcentaje que se puede obtener y el detalle por medio de tablas con los conceptos a tomados en cuenta para la comisión.

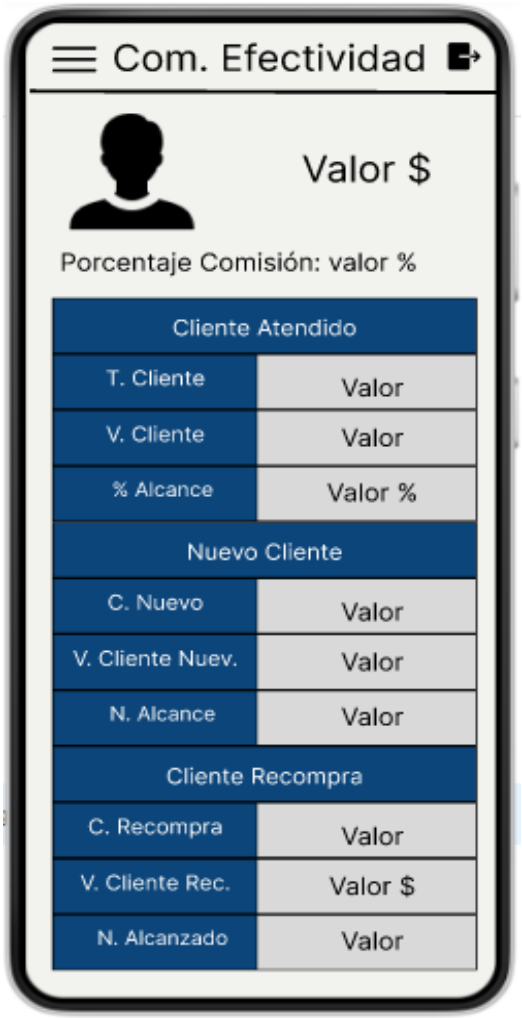


Figura 27. Comisión efectividad

- **Comisión por rentabilidad**

En esta interfaz Figura 28 se detalla la consigna por la cual se puede obtener una comisión por concepto de rentabilidad, se divide el valor alcanzado, porcentaje que se puede obtener y el detalle por medio de una tabla con los conceptos a tomados en cuenta para la obtención de la comisión.

Figura 28. Comisión rentabilidad

- **Bono por monto**

En esta interfaz Figura 29 se detalla la consigna por la cual se puede obtener un bono por el monto, se divide con el valor alcanzado, total del bono que se puede obtener y el detalle por medio de una tabla con los conceptos a tomados en cuenta para el bono.

Figura 29. Bono monto

- **Bono por promocional**

En esta interfaz Figura 30 se detalla la consigna por la cual se puede obtener un bono por concepto de promocionales, se divide en el valor alcanzado, valor que se puede obtener al cumplir los parámetros que se detalle por medio de una tabla con los conceptos a tomados en cuenta para la obtención del bono.

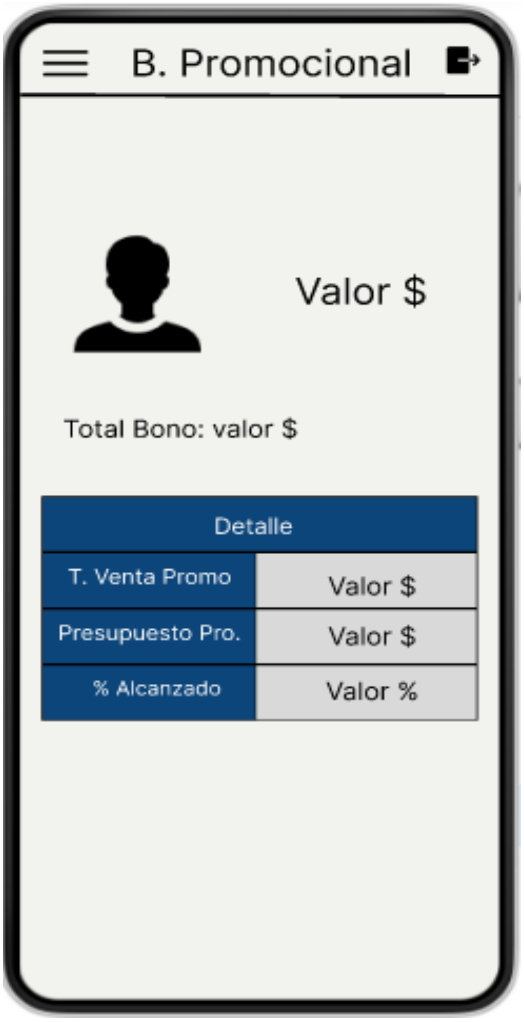


Figura 30. Bono promocional

- **Bono por cartera**

En esta interfaz Figura 31 se detalla las consignas por las cuales se puede obtener un bono por concepto de cartea, se divide por el valor alcanzado y el detalle por medio de tablas con los conceptos a tomados en cuenta para la obtención del bono.

Figura 31. Bono cartera

- **Total progreso alcanzado**

En esta interfaz Figura 32 se detalla los valores obtenidos y el porcentaje alcanzado en concepto de las comisiones y bonos que los usuarios al obtenido durante el mes, se permite realizar una búsqueda en concepto del mes y el año, todos estos valores se detallan por medio de una tabla, a su vez se permite la descarga en formato Pdf de estos datos.

Figura 32. Total progreso alcanzado

- **Creación nuevo usuario**

En esta interfaz Figura 33 se permite la creación de un nuevo cliente, se permite el ingreso de los datos personales de un cliente.

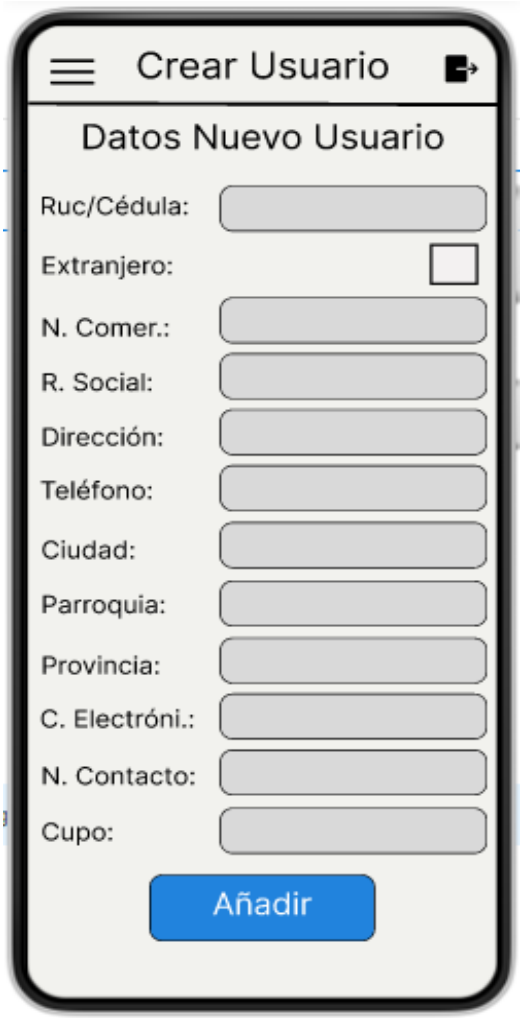


Figura 33. Bono monto

En esta segunda sección Figura 34 para la creación del nuevo cliente se permite el ingreso de la documentación requerida del nuevo cliente y se permite el registro del mismo.

Figura 34. Bono monto

- **Ventas Clientes**

En esta interfaz Figura 35 se permite la visualización de un listado con las ventas de los clientes asignados por cada asesor comercial.

Figura 35. Ventas clientes asignados

En esta segunda sección Figura 36 se visualiza un reporte gráfico y un listado con las cantidades de productos que se ha adquirido y devuelto por cada cliente.

Figura 36. Compras y devoluciones productos

Prototipo para usuario administrador.

- **Inicio Administrador**

En esta interfaz Figura 37 se da la bienvenida al usuario administrador.

Figura 37. Administrador inicio

- **Administración progreso diario**

En esta interfaz Figura 38 el usuario administrador puede revisar las estadísticas del progreso en cumplimiento por cada asesor comercial, para lo cual se permite realizar búsquedas en concepto de asesor comercial, mes y año.

Figura 38. Administrador progreso diario

- **Administración comisión por bono**

En esta interfaz Figura 39 el usuario administrador puede revisar los valores alcanzados en comisiones por bono por parte de los asesores comerciales, se permite la selección y búsqueda del asesor comercial, mes y año.

Figura 39. Administrador comisión bono

- **Administración comisión por efectividad**

En esta interfaz Figura 40 el usuario administrador puede revisar los valores alcanzados en comisiones por efectividad por parte de los asesores comerciales, se permite la selección y búsqueda del asesor comercial, mes y año.

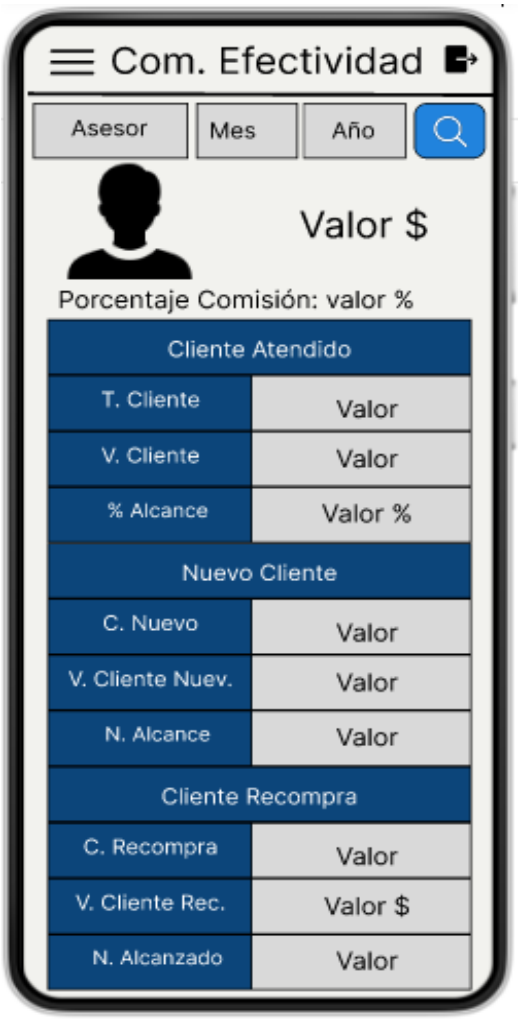


Figura 40. Administrador comisión efectividad

- **Administración comisión por rentabilidad**

En esta interfaz Figura 41 el usuario administrador puede revisar los valores alcanzados en comisiones por rentabilidad por parte de los asesores comerciales, se permite la selección y búsqueda del asesor comercial, mes y año.

Figura 41. Administrador comisión rentabilidad

- **Administración bono por monto**

En esta interfaz Figura 42 el usuario administrador puede revisar los valores del bono alcanzados por concepto de monto por parte de los asesores comerciales, se permite la selección y búsqueda del asesor comercial, mes y año.

Figura 42. Administrador bono monto

- **Administración bono por promocional**

En esta interfaz Figura 43 el usuario administrador puede revisar los valores del bono alcanzados por concepto de promocional por parte de los asesores comerciales, se permite la selección y búsqueda del asesor comercial, mes y año.

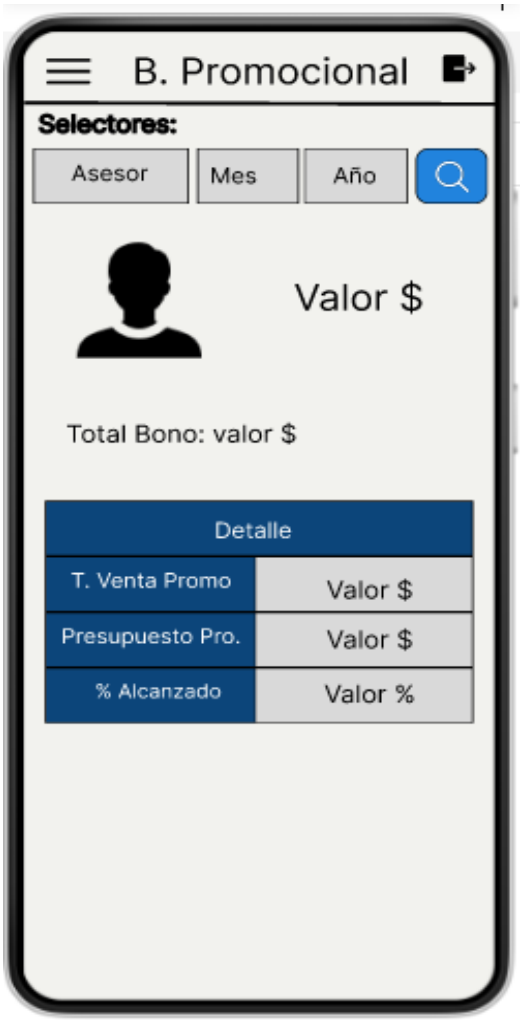


Figura 43. Administrador bono por promocional

- **Administración bono por cartera**

En esta interfaz Figura 44 el usuario administrador puede revisar los valores del bono alcanzados por concepto de cartera por parte de los asesores comerciales, se permite la selección y búsqueda del asesor comercial, mes y año.

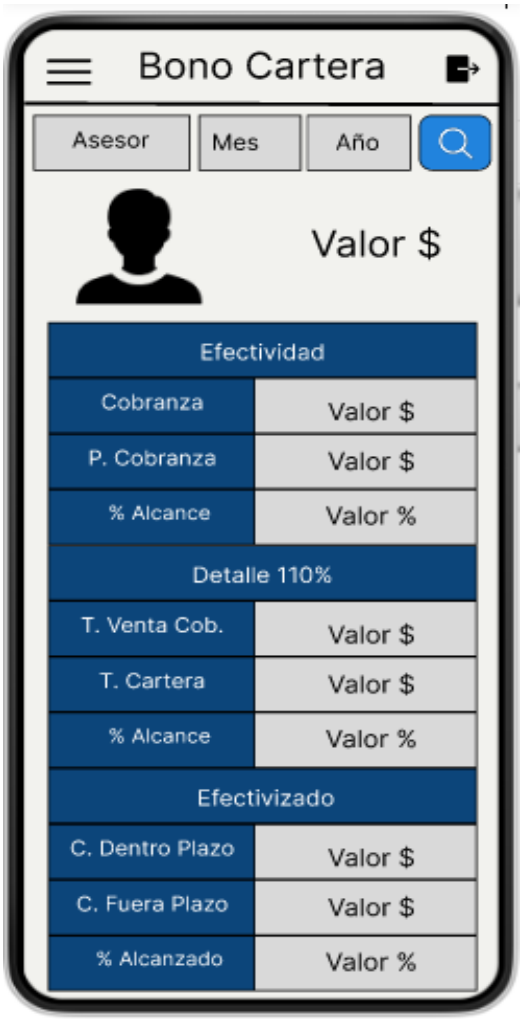


Figura 44. Administrador bono cartera

- **Administración total progreso**

En esta interfaz Figura 45 el usuario administrador puede revisar el valor total alcanzado por cada asesor comercial en concepto de bonos y comisiones, se permite la búsqueda a través del asesor comercial, mes y año.

Figura 45. Administrador total progreso

- **Administración ventas clientes**

En esta interfaz Figura 46 el usuario administrador puede revisar los valores totales en ventas a clientes realizadas a través de una búsqueda por cada asesor comercial.

Figura 46. Administrador ventas clientes

3.2.3 Fase 3: Producción

En esta fase se realizó el desarrollo del aplicativo móvil tomando en consideración las interacciones y requerimientos que fueron definidos en la fase 1 y 2.

a. Backend

Node.js está enfocado para el desarrollo del lado backend, para el proyecto se utilizó esta herramienta para la creación de un web services por el cual se permite realizar

peticiones que permita obtener, insertar y actualizar según sea el requerimiento por cada módulo.

- **Procedimientos almacenados**

Para la interacción entre la base de datos y el web services se generó procedimientos almacenados que faciliten la consulta y obtención de datos de das distintas tablas generadas para el proyecto, a su vez también son utilizados para la obtención de datos específicos desde el sistema contable utilizado por la Empresa y son esenciales en el proceso de generación de presupuestos, comisiones y bonos dentro del proyecto, a continuación, en la Tabla 28 se describe cómo funcionan cada uno.

Tabla 28. Procedimientos almacenados

Procedimiento creado	**Descripción**
dbo.SP_LOGIN_APP_VALIDARLOGIN	Procedimiento utilizado para validar el usuario y contraseña ingresado por parte del usuario
dbo.SP_APP_BUSCAR_VENDEDORES	Procedimiento utilizado para realizar una busque del historial de vendedores registrados en la Empresa que permita obtener datos personales usados para el registro de un nuevo usuario en el aplicativo
dbo.SP_VENT_APP_INSERTARUSUARIO	Procedimiento utilizado para la creación de un nuevo usuario en el aplicativo
dbo.SP_VENT_APP_PRESUPUESTOSDIA	Procedimiento utilizado para la extracción del resumen de los indicadores: código de asesores comerciales, mes, año, presupuesto de venta, ventas realizadas hasta la fecha, porcentaje cumplimiento de presupuesto, porcentaje objetivo de efectividad, porcentaje efectivo alcanzado, porcentaje objetivo de rentabilidad, porcentaje obtenido de rentabilidad, porcentaje para cumplimiento de bono por ventas realizadas, porcentaje alcanzado para bono por cumplimiento de ventas, porcentaje para cumplimiento de bono de promocionales, porcentaje alcanzado para bono de promocionales, numero de objetivo de clientes nuevos y total clientes nuevos creados La información necesaria es extraída del sistema contable presente en la Empresa, esta información es ingresada en las tablas creadas en la fase 1, una vez ingresados los datos se obtiene el presupuesto y se lo compara con el valor de las ventas que se va realizando por cada día, obteniendo de esta

	manera el porcentaje de cumplimiento de presupuesto, este cálculo se realiza por cada interacción del procedimiento, considerando todas las ventas que se van generando durante el día. Para el cumplimiento de la efectividad se considera el número total de clientes asignados a cada uno de los asesores comerciales, se compara con los clientes que realizaron compras en el día y permite la obtención del porcentaje del cumplimiento Para el cálculo de la rentabilidad se considera los costos de cada uno de los productos con los precios facturados por cada cliente, de esta manera se obtiene el porcentaje de la rentabilidad alcanzado Para el cálculo del bono por cumplimiento se considera el presupuesto de cada uno de los asesores comerciales con las ventas realizadas por cada día, de esta manera se obtiene el porcentaje del bono por cumplimiento Para el cálculo del bono por promocionales se considera el presupuesto de los productos en promoción y las ventas realizas de estos productos, de esta manera se obtiene el porcentaje del bono por promocionales Para el cálculo del bono por creación de clientes nuevos se requiere la creación de tres clientes por mes, de esta manera se obtiene esta bonificación
dbo.SP_VENT_APP_INFODIACOMIMONTO	Procedimiento utilizado para calcular las ventas que se realizan durante el trascurso del día, esta información se extra del sistema contable de la Empresa, se realiza la suma de las ventas, restando las devoluciones y sin considerar los artículos de servicio. El presupuesto se obtiene de la tabla de presupuestos mensuales, el porcentaje de la comisión es el 0.25% de las ventas realizadas por cada día y el porcentaje objetivo para obtener la bonificación es el 90% del cumplimiento
dbo.SP_VENT_APP_INFODIACOMIEFECTI	Procedimiento utilizado para el cálculo de la efectividad, se considera los clientes asignados a cada uno de los asesores comerciales y el número de clientes que realizaron compras con la fecha actual, el porcentaje de la comisión es igual al 0.25% de la venta obtenida con la fecha actual, dividida de la siguiente manera: El 0.12% se asigna cuando el asesor comercial llega al 80% de atención de los clientes asignados El 0.06% se obtiene calculando la creación de tres clientes nuevos al mes

	El 0.06% se obtiene por las ventas realizadas en el mes a los clientes que se crearon el mes anterior.
dbo.SP_VENT_APP_INFODIACOMIRENTAB	Procedimiento utilizado para obtener el costo de los artículos vendidos y el valor de la venta desde el sistema contable de la Empresa, con lo cual se calcula el porcentaje de rentabilidad el cual debe ser igual o superior al 25% para que el asesor comercial obtenga su comisión que equivale al 0.25% de las ventas obtenidas
dbo.SP_VENT_APP_INFODIABONOMONTO	Procedimiento utilizado para calcular las ventas que se realizan durante el trascurso del día, esta información se extra del sistema contable de la Empresa, sumando las ventas, restando las devoluciones, sin considerar los artículos de servicio. El presupuesto se obtiene de la tabla de presupuestos mensuales y el bono es de 100 dólares cuando el asesor comercial alcanzo el 95% de su cumplimiento
dbo.SP_VENT_APP_INFODIABONOPROMO	Procedimiento utilizado para calcular las ventas que se realizan durante el trascurso del día, esta información se extra del sistema contable de la Empresa, sumando las ventas, restando las devoluciones, sin considerar los artículos de servicio y cuyos productos estén dentro de las promociones del mes. El presupuesto se obtiene de la tabla de promocional por mes y el bono es de 120 dólares cuando el asesor comercial alcanzo el 100% de la venta de los productos en promoción
dbo.SP_VENT_APP_INFODIABONOCOBRAR	Procedimiento utilizado para el cálculo de la bonificación por respaldo de cartea, se considera las ventas realizadas por día y los pagos realizados, la bonificación se divide en tres partes: Por efectividad, se otorga una bonificación de 100 dólares por las ventas realizadas en efectivo y cuyo porcentaje debe ser del 40% Por alcance del 110%, se le otorga al asesor comercial una bonificación de 50 dólares cuando tiene su cartera al día, es decir la morosidad no debe superar un 10% de las ventas por cada mes Por efectivizado, se otorga al asesor comercial una bonificación de 50 dólares cuando cumple con los dos puntos anteriores
dbo.SP_VENT_APP_INFODIACOMITOTAL	Procedimiento utilizado para calcular el total de los valores de las comisiones por moto, efectividad y rentabilidad, bonos por monto, promocionales y cartera, así como el valor total que es la suma de todos los valores anteriores

dbo.SP_VENT_APP_INFOIDEALCOMI	Procedimiento utilizado para calcular el monto máximo que puede obtener un asesor comercial considerando que el mismo ha llegado al máximo de los indicadores como lo son, se cumplió con el 100% en ventas, el 100% en efectividad, el porcentaje mínimo de rentabilidad y el 100% en los bonos, con lo cual se suma cada una de las bonificaciones obteniendo el máximo que puede llegar a ganar por mes
dbo.SP_VENT_APP_LISTADOVENDEDORES	Procedimiento utilizado para obtener los datos de cada asesor comercial
dbo.SP_VENT_APP_LISTADOPROVINCIAS	Procedimiento utilizado para obtener los datos de provincias
dbo.SP_VENT_APP_CONSUL_VERIF_CLIENTE	Procedimiento utilizado para registrar los datos de un nuevo cliente
dbo.SP_VENT_APP_CORREOCAMBIOCLAVE	Procedimiento utilizado para enviar un correo al usuario con un código de verificación para permitir el cambio de contraseña
dbo.SP_VENT_APP_CAMBIOCLAVE	Procedimiento utilizado para realizar la actualización de la contraseña de un usuario
dbo.SP_VENT_APP_CORREOCAMBIOCLAVE	Procedimiento utilizado para el envío de correo electrónico a los distintos usuarios, se arma el formato del correo, se realizar él envió del correo electrónico a la cuenta de Outlook de Microsoft Office 365 y en el mensaje se adjunta el código generado con el cual se valida la identidad del usuario.
dbo.SP_VENT_APP_INFOCLEINTE	Procedimiento utilizado para obtener el valor de las ventas y devoluciones realizadas durante los últimos 6 meses, se suma el total de las ventas y se resta las devoluciones, todo se realiza por cada cliente asignado a un usuario.
dbo.SP_VENT_APP_INFOCLIENTEPRODUCTOS	Procedimiento utilizado para obtener la cantidad de productos adquiridos y devueltos de los últimos 6 meses, se suman la cantidad de productos vendidos y se restan las devoluciones por cada cliente.

Los procedimientos generados fueron utilizados en el web services creado para le aplicativo móvil, estos procedimientos son ejecutados por medio de peticiones de tipo get, post y put, estos métodos sirven para obtener, enviar y actualizar distintos datos en la base de datos, utilizando un manejo por medio de asincronismo que principalmente permite ejecutar el código en segundo plano y evita el bloque de la ejecución del resto del mismo.

- **Dependencias usadas**

Para el desarrollo del proyecto se implementan dependencias que permiten la construcción del backend abordando requisitos específicos del proyecto, a continuación, en la Figura 47 se presentan las dependencias utilizadas.

Figura 47. Dependencias del backend

b. Frontend

Para el frontend se utilizó el framework Ionic con React que permite la creación de interfaces fáciles de personalizar y adaptar a los requerimientos específicos del proyecto.

- **Vista**

Las vistas que fueron desarrolladas para el aplicativo móvil están enfocadas para la interacción y visualización de los usuarios, esto permite que los usuarios puedan tener una visualización clara de los módulos definidos anteriormente, a continuación, en la Tabla 29 se presenta un listado de vista presentes en el aplicativo móvil.

Tabla 29. Vistas

Función	**Vista**
Ingreso aplicativo	Login.tsx
Inicio	Home.tsx
Registro	CrearCuenta.tsx
Cambio contraseña	CambioContrasena.tsx
Progreso por día	DatosDia.tsx
Comisión por concepto de bono	Monto.tsx
Comisión por concepto de efectividad	Efectividad.tsx
Comisión por concepto de rentabilidad	Rentabilidad.tsx
Bonificación por concepto de monto	BonoMonto.tsx
Bonificación por concepto de promocional	Promocional.tsx
Bonificación por concepto de cartera	Cartera.tsx
Total progreso comisión y bonificación	TotalHistorial.tsx
Creación nuevo cliente	CrearCliente.tsx
Ventas Clientes	VentaCliente.tsx

- **Dependencias usadas en el desarrollo del aplicativo**

Para el desarrollo del proyecto se implementan dependencias que permiten la construcción y manejo de capacidades nativas en el aplicativo móvil del frontend abordando requisitos específicos y necesarios del proyecto, a continuación, en la Figura 48 se presentan las dependencias utilizadas.

```
"dependencies": {
  "@capacitor-community/fcm": "^5.0.1",
  "@capacitor/android": "5.6.0",
  "@capacitor/app": "5.0.6",
  "@capacitor/camera": "^5.0.9",
  "@capacitor/core": "^5.6.0",
  "@capacitor/haptics": "5.0.6",
  "@capacitor/ios": "5.6.0",
  "@capacitor/keyboard": "5.0.7",
  "@capacitor/push-notifications": "^5.1.1",
  "@capacitor/status-bar": "5.0.6",
  "@ionic-native/file": "^5.36.0",
  "@ionic-native/file-opener": "^5.36.0",
  "@ionic/pwa-elements": "^3.2.2",
  "@ionic/react": "^7.0.0",
  "@ionic/react-router": "^7.0.0",
  "@types/react-router": "^5.1.20",
  "@types/react-router-dom": "^5.3.3",
  "axios": "^1.6.5",
  "bootstrap": "^5.3.2",
  "cordova-plugin-file": "^8.0.1",
  "cordova-plugin-file-opener2": "^4.0.0",
  "html2canvas": "^1.4.1",
  "ionicons": "^7.2.2",
  "jspdf": "^2.5.1",
  "md5": "^2.3.0",
  "react": "^18.2.0",
  "react-dom": "^18.2.0",
  "react-hooks": "^1.0.1",
  "react-router": "^5.3.4",
  "react-router-dom": "^5.3.4",
  "recharts": "^2.12.7"
}
```

Figura 48. Dependencias del frontend

c. StoryCards (Tarjetas de historia)

Los StoryCards son un elemento clave en la metodología Mobile-D, diseñadas para documentar las historias de usuario de una aplicación móvil de manera detallada y precisa. Estas tarjetas contienen una descripción completa de cada historia de usuario, ofreciendo un registro exhaustivo de los requisitos y las funcionalidades que se deben implementar en la aplicación móvil.

A continuación, en la Tabla 30 se presente el modelo utilizado para la creación de StoryCards en el proyecto.

Tabla 30. Diseño StoryCards

N°	**Tipo**	**Dificultad**		**Esfuerzo/Horas**		**Importancia**
		Antes	**Después**	**Antes**	**Después**	
#	Nuevo Fijo Mejora	Simple Regular Complicado	Simple Regular Complicado	#	#	Bajo Intermedio Alto
Descripción						
Fecha	Estado	Comentario				

La tabla está dividida por, el número donde se indica el identificador único del StoryCard, en el campo tipo donde se elige una opción (nuevo, fijo, mejora) que describe el tipo de funcionalidad que se va a implementar. Por lo general, durante esta etapa, la opción por defecto es "nuevo". La dificultad se refiere al nivel de complejidad asociado con la funcionalidad, tanto antes como después de su implementación, y se puede clasificar como simple, regular o complicado. El campo esfuerzo indica la estimación y el tiempo real empleado para desarrollar la funcionalidad, expresado en horas. La importancia refleja el impacto que la actividad tendrá en el proyecto, y en la descripción se detalla la acción que debe cumplir la funcionalidad.

- **StoryCard registro usuario**

Iteración 2: Desarrollo del módulo para el registro de los usuarios.

Tabla 31. StoryCard registro usuario

N°	**Tipo**	**Dificultad**		**Esfuerzo/Horas**		**Importancia**
		Antes	**Después**	**Antes**	**Después**	
1	Nuevo	Regular	Regular	8	10	Alto
Descripción	Los usuarios pueden crearse una cuentan para ingresas al aplicativo.					
Fecha	Estado	Comentario				
11/03/2024	Definición	Sin comentarios				
18/03/2024	Implementación	Sin comentarios				
18/03/2024	Ejecución	Sin comentarios				
19/03/2024	Verificación	Sin comentarios				

En el Anexo A se detalla el código empleado para el registro de un nuevo usuario en el aplicativo, donde se realiza una petición al Web Services de validación del código único del asesor comercial, si el código es válido se cargan los datos personales del asesor y se permite la creación de la nueva cuenta, si el código no es válido se presenta un mensaje de advertencia.

- **StoryCard inicio sesión**

Iteración 3. Desarrollo del módulo para el ingreso de los usuarios registrados.

Tabla 32. StoryCard inicio sesión

N°	**Tipo**	**Dificultad**		**Esfuerzo/Horas**		**Importancia**
		Antes	**Después**	**Antes**	**Después**	
2	Nuevo	Regular	Regular	8	11	Alto
Descripción	Los usuarios pueden iniciar sesión si cuentan con una cuenta creada.					
Fecha	Estado	Comentario				
11/03/2024	Definición	Sin comentarios				
20/03/2024	Implementación	Sin comentarios				
20/03/2024	Ejecución	Sin comentarios				
21/03/2024	Verificación	Sin comentarios				

En el Anexo B se detalla el código empleado para el manejo de la funcionalidad y validad de inicio de sesión, donde se realiza una petición al Web Services si la respuesta es favorable se permite el acceso al sistema de lo contrario se envía un mensaje de advertencia de credenciales incorrectas.

- **StoryCard cambio de contraseña**

Iteración 4: Desarrollo del módulo para el cambio de la contraseña por parte de los usuarios.

Tabla 33. StoryCard cambio de contraseña

N°	Tipo	Dificultad		Esfuerzo/Horas		Importancia
		Antes	Después	Antes	Después	
3	Nuevo	Regular	Regular	12	15	Alto
Descripción	Los usuarios pueden realizar un cambio de contraseña si lo requieren.					
Fecha	Estado	Comentario				
11/03/2024	Definición	Sin comentarios				
22/03/2024	Implementación	Sin comentarios				
23/03/2024	Ejecución	Sin comentarios				
25/03/2024	Verificación	Sin comentarios				

En el Anexo C se detalla el código empleado para el cambio de contraseña, se requiere el código del vendedor y se realiza una petición al Web Services para validar el código ingresado si es válido se envía un correo electrónico con un código generado, se valida el código si es correcto se permite el cambio de contraseña.

- **StoryCard inicio**

Iteración 5: Desarrollo del módulo de bienvenida al aplicativo.

Tabla 34. StoryCard inicio

N°	Tipo	Dificultad		Esfuerzo/Horas		Importancia
		Antes	Después	Antes	Después	
4	Nuevo	Simple	Regular	5	8	Alto
Descripción	Los usuarios pueden ingresar a la página de inicio ver promocionales y presupuesto en bonificaciones.					
Fecha	Estado	Comentario				
11/03/2024	Definición	Sin comentarios				
26/03/2024	Implementación	Sin comentarios				
26/03/2024	Ejecución	Sin comentarios				
27/03/2024	Verificación	Sin comentarios				

En el Anexo D se detalla el código empleado para la página de inicio, donde se realiza una petición al Web Services para obtener las imágenes de promocionales y para obtener el valor total del presupuesto en bonificaciones.

- **StoryCard progreso actual**

Iteración 6: Desarrollo de la vista del total en ventas, presupuesto, porcentaje alcanzado, cumplimiento, efectividad, rentabilidad, bonos y número de nuevos clientes.

Tabla 35. StoryCard progreso actual

N°	**Tipo**	**Dificultad**		**Esfuerzo/Horas**		**Importancia**
		Antes	**Después**	**Antes**	**Después**	
5	Nuevo	Complicado	Regular	18	24	Alto
Descripción	Los usuarios pueden ingresar a la página de progreso actual, se detalla el total en ventas, presupuesto, porcentaje alcanzado, cumplimiento, efectividad, rentabilidad, bonos y número de nuevos clientes.					
Fecha	Estado	Comentario				
11/03/2024	Definición	Sin comentarios				
01/04/2024	Implementación	Sin comentarios				
03/04/2024	Ejecución	Sin comentarios				
04/04/2024	Verificación	Sin comentarios				

En el Anexo E se detalla el código empleado para la vista de progreso actual, donde se realiza una petición al Web Services con la fecha y el código de cada asesor comercial, si la respuesta retorna datos se almacenan en variables que se usan para mostrar los valores, si no es afirmativa la respuesta se muestra un mensaje de advertencia.

- **StoryCard comisión por monto**

Iteración 7: Desarrollo de la vista con el detalle de la venta neta, presupuesto, valor comisión y porcentaje a cumplir.

Tabla 36. StoryCard comisión por monto

N°	Tipo	Dificultad		Esfuerzo/Horas		Importancia
		Antes	Después	Antes	Después	
6	Nuevo	Regular	Regular	10	14	Alto
Descripción	Los usuarios pueden ingresar a la página de comisión por monto, se detalla la venta neta, presupuesto, valor comisión y porcentaje a cumplir.					
Fecha	Estado	Comentario				
11/03/2024	Definición	Sin comentarios				
05/04/2024	Implementación	Sin comentarios				
05/04/2024	Ejecución	Sin comentarios				
06/04/2024	Verificación	Sin comentarios				

En el Anexo F se detalla el código empleado para la vista de comisión por monto, donde se realiza una petición al Web Services con la fecha y el código de cada asesor comercial, si la respuesta retorna datos se almacenan en variables que se usan para mostrar los valores, si no es afirmativa la respuesta se muestra un mensaje de advertencia.

- **StoryCard comisión por efectividad**

Iteración 8: Desarrollo de la vista con el detalle y valor alcanzado, clientes atendidos con el total, venta cliente y alcance, nuevo cliente con número clientes nuevos, venta, alcance y cliente recompra con numero de recompra, venta y alcance.

Tabla 37. StoryCard comisión por efectividad

N°	Tipo	Dificultad		Esfuerzo/Horas		Importancia
		Antes	Después	Antes	Después	
7	Nuevo	Complicado	Regular	15	20	Alto
Descripción	Los usuarios pueden ingresar a la página de comisión por efectividad, se detalla el valor alcanzado, clientes atendidos, venta cliente y alcance, nuevo cliente con número clientes nuevos, venta, alcance y cliente recompra con numero de recompra, venta y alcance.					
Fecha	Estado	Comentario				
11/03/2024	Definición	Sin comentarios				
08/04/2024	Implementación	Sin comentarios				
08/04/2024	Ejecución	Sin comentarios				
10/04/2024	Verificación	Sin comentarios				

En el Anexo G se detalla el código empleado para la vista de comisión por efectividad, donde se realiza una petición al Web Services con la fecha y el código de cada asesor comercial, si la respuesta retorna datos se almacenan en variables que se usan para mostrar estos valores, si no es afirmativa la respuesta se muestra un mensaje de advertencia.

- **StoryCard comisión por rentabilidad**

Iteración 9: Desarrollo de la vista con el detalle del valor alcanzado, porcentaje y rentabilidad objetivo.

Tabla 38. StoryCard comisión por rentabilidad

N°	**Tipo**	**Dificultad**		**Esfuerzo/Horas**		**Importancia**
		Antes	**Después**	**Antes**	**Después**	
8	Nuevo	Regular	Regular	8	12	Alto
Descripción	Los usuarios pueden ingresar a la página de comisión por rentabilidad, se detalla el valor alcanzado, porcentaje y rentabilidad objetivo.					
Fecha	Estado	Comentario				
11/03/2024	Definición	Sin comentarios				
11/04/2024	Implementación	Sin comentarios				
11/04/2024	Ejecución	Sin comentarios				
12/04/2024	Verificación	Sin comentarios				

En el Anexo H se detalla el código empleado para la vista de comisión por rentabilidad, donde se realiza una petición al Web Services con la fecha y el código de cada asesor comercial, si la respuesta retorna datos se almacenan en variables que se usan para mostrar estos valores, si no es afirmativa la respuesta se muestra un mensaje de advertencia.

- **StoryCard bono por monto**

Iteración 10: Desarrollo de la vista con el detalle del valor a alcanzar, valor obtenido, venta, presupuesto y porcentaje cumplimiento.

Tabla 39. StoryCard bono por monto

N°	Tipo	Dificultad		Esfuerzo/Horas		Importancia
		Antes	Después	Antes	Después	
9	Nuevo	Regular	Regular	8	12	Alto
Descripción	Los usuarios pueden ingresar a la página de bono por monto, se detalla el valor a alcanzar, valor obtenido, venta, presupuesto y porcentaje cumplimiento.					
Fecha	Estado	Comentario				
11/03/2024	Definición	Sin comentarios				
13/04/2024	Implementación	Sin comentarios				
13/04/2024	Ejecución	Sin comentarios				
15/04/2024	Verificación	Sin comentarios				

En el Anexo I se detalla el código empleado para la vista de bono por monto, donde se realiza una petición al Web Services con la fecha y el código de cada asesor comercial, si la respuesta retorna datos se almacenan en variables que se usan para mostrar estos valores, si no es afirmativa la respuesta se muestra un mensaje de advertencia.

- **StoryCard bono por promocional**

Iteración 11: Desarrollo de la vista con el detalle del porcentaje de comisión, valor obtenido, venta promoción y presupuesto promoción.

Tabla 40. StoryCard bono por promocional

N°	Tipo	Dificultad		Esfuerzo/Horas		Importancia
		Antes	Después	Antes	Después	
10	Nuevo	Regular	Regular	10	12	Alto
Descripción	Los usuarios pueden ingresar a la página de bono por promocional, se detalla el porcentaje de comisión, valor obtenido, venta promoción y presupuesto promoción.					
Fecha	Estado	Comentario				
11/03/2024	Definición	Sin comentarios				
16/04/2024	Implementación	Sin comentarios				
16/04/2024	Ejecución	Sin comentarios				
17/04/2024	Verificación	Sin comentarios				

En el Anexo J se detalla el código empleado para la vista de bono por promocional, donde se realiza una petición al Web Services con la fecha y el código de cada asesor comercial, si la respuesta retorna datos se almacenan en variables que se usan para mostrar estos valores, si no es afirmativa la respuesta se muestra un mensaje de advertencia.

- **StoryCard bono por cartera**

Iteración 12: Desarrollo de la vista con el detalle de la efectiva cobranza y cumplimiento presupuesto, por detalle mayor al 110% con la venta cobrada y valor cartera y por detalle de efectivizado con la cobranza en plazo y cobranza fuera de plazo.

Tabla 41. StoryCard bono por cartera

N°	**Tipo**	**Dificultad**		**Esfuerzo/Horas**		**Importancia**
		Antes	**Después**	**Antes**	**Después**	
11	Nuevo	Regular	Regular	12	15	Alto
Descripción	Los usuarios pueden ingresar a la página de bono por cartea, se detalla la efectiva cobranza y cumplimiento presupuesto, el detalle mayor al 110% con la venta cobrada y valor cartera y el detalle de efectivizado con la cobranza en plazo y cobranza fuera de plazo.					
Fecha	Estado	Comentario				
11/03/2024	Definición	Sin comentarios				
18/04/2024	Implementación	Sin comentarios				
18/04/2024	Ejecución	Sin comentarios				
20/04/2024	Verificación	Sin comentarios				

En el Anexo K se detalla el código empleado para la vista de bono por cartera, donde se realiza una petición al Web Services con la fecha y el código de cada asesor comercial, si la respuesta retorna datos se almacenan en variables que se usan para mostrar estos valores, si no es afirmativa la respuesta se muestra un mensaje de advertencia.

- **StoryCard total progreso**

Iteración 13: Desarrollo de la vista con el total de los montos alcanzados en comisiones, bonos, detalle de la comisión por monto, efectividad, rentabilidad, bono por montos, el total alcanzado y permitir la descarga de archivo Pdf con los valores obtenidos durante el mes.

Tabla 42. StoryCard total progreso

N°	**Tipo**	**Dificultad**		**Esfuerzo/Horas**		**Importancia**
		Antes	**Después**	**Antes**	**Después**	
12	Nuevo	Complicado	Regular	20	35	Alto
Descripción	Los usuarios pueden ingresar a la página del total en progreso, se detalla el total de los montos alcanzados en comisiones, bonos, detalle de la comisión por monto, efectividad, rentabilidad, bono por montos, el total alcanzado y permitir la descarga de archivo pdf con los valores obtenidos durante el mes.					
Fecha	Estado	Comentario				
11/03/2024	Definición	Sin comentarios				
22/04/2024	Implementación	Sin comentarios				
25/04/2024	Ejecución	Sin comentarios				
27/04/2024	Verificación	Sin comentarios				

En el Anexo L se detalla el código empleado para la vista del total del progreso alcanzado por los asesores comerciales, donde se realiza una petición al Web Services con la fecha y el código de cada asesor comercial, si la respuesta retorna datos se almacenan en variables que se usan para mostrar estos valores, si no es afirmativa la respuesta se muestra un mensaje de advertencia, realizar una petición al Web Services con la fecha y el código del hacer comercial, si la petición es afirmativa se almacenan los datos en variables, se usa estas variables para crear un nuevo archivo Pdf que se descarga de forma automática, si la petición no es afirmativa se muestra un mensaje de advertencia.

- **StoryCard creación nuevo cliente**

Iteración 14: Desarrollo del módulo para la creación de nuevos clientes, ingreso de datos y documentación del cliente.

Tabla 43. StoryCard creación nuevo cliente

N°	Tipo	Dificultad		Esfuerzo/Horas		Importancia
		Antes	Después	Antes	Después	
13	Nuevo	Complicado	Regular	18	25	Alto
Descripción	Los usuarios pueden registrar nuevos clientes, ingreso de datos y documentos personales.					
Fecha	Estado	Comentario				
11/03/2024	Definición	Sin comentarios				
29/04/2024	Implementación	Sin comentarios				
30/04/2024	Ejecución	Sin comentarios				
01/05/2024	Verificación	Sin comentarios				

En el Anexo M se detalla el código empleado para la creación de nuevos usuarios, se divide por dos secciones, la primera si el cliente compra a contado solo se requiere los datos personales y dos documentos, la segunda si el cliente compra por plazos se requiere los datos personales y todos los papeles requeridos, al obtener los valores se realiza una petición la Web Services si la petición retorta un OK se presenta un mensaje de confirmación de creación correcta del clientes, si la respuesta no es favorable se presenta un mensaje de advertencia.

- **StoryCard ventas clientes**

Iteración 15: Desarrollo del módulo para la revisión y vista de las ventas a los clientes por asesor comercial, así como productos vendidos a los clientes.

Tabla 44. StoryCard ventas clientes

N°	Tipo	Dificultad		Esfuerzo/Horas		Importancia
		Antes	Después	Antes	Después	
14	Nuevo	Complicado	Regular	20	35	Alto
Descripción	Los usuarios pueden ingresar a la página de ventas por clientes para la revisión de clientes y productos vendidos a cada cliente.					
Fecha	Estado	Comentario				
11/03/2024	Definición	Sin comentarios				
02/05/2024	Implementación	Sin comentarios				
07/05/2024	Ejecución	Sin comentarios				
09/05/2024	Verificación	Sin comentarios				

En el Anexo N se detalla el código utilizado para la generación de un listado por ventas a clientes de parte de cada asesor comercial, así como los productos comprados por los clientes, se realiza una petición al Web Services se obtiene los datos y almacenan en variables, estos datos son utilizados para la visualización, en caso de no tener una respuesta favorable en la petición se envía un mensaje de advertencia.

3.2.4 Fase 4: Estabilización

En esta fase del desarrollo del aplicativo se realizan ajustes necesarios para garantizar que el aplicativo móvil se mantenga estable y se minimicen los fallos que se puedan producir, garantizando que la funcionalidad del aplicativo se la adecuada, que todos los procesos estén integrados y alineados con las interacciones anteriormente definidas. En este marco se realizaron las siguientes modificaciones en la interfaz y código del aplicativo móvil.

Cambios en la interfaz

- En el modelo de la interfaz de inicio se sesión, Figura 49 se modifica el estilo de los labels para el ingreso de datos, diseño del botón y cambio de color del fondo de la interfaz.

Figura 49. Cambio modelo interfaz inicio se sesión

- En el modelo de la interfaz de registro de usuario, Figura 50 se cambia la distribución de los componentes y se cambia el fondo de la interfaz.

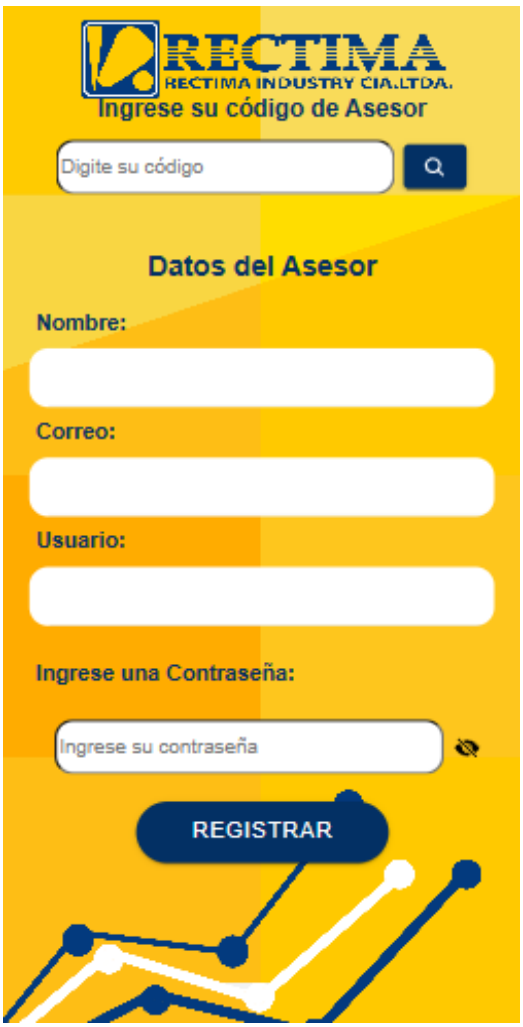


Figura 50. Cambio modelo interfaz registro usuario

- Nuevos iconos representativos del menú de opciones, en la Figura 51 se colocan iconos más representativos por cada sesión del menú.

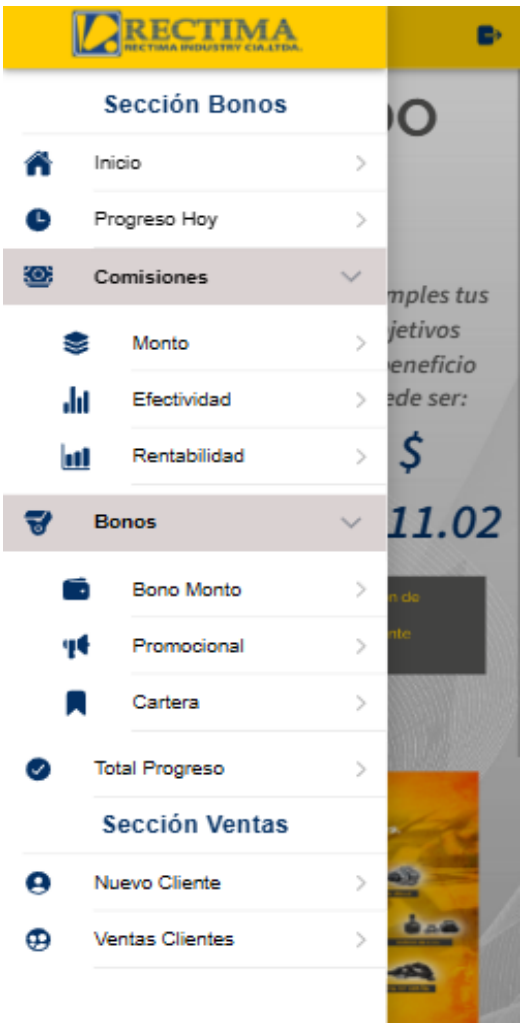


Figura 51. Cambio iconos menú

- En la interfaz del progreso actual, Figura 52 se corrige la distribución de los elementos en la tabla, tamaño de letra y tamaño de imagen del asesor comercial.

Figura 52. Cambio interfaz progreso actual

- En la interfaz por total de progreso, Figura 53 se modifica el orden de las filas, tamaño de letra de valor total obtenido y el tamaño de la tabla.

Figura 53. Cambio interfaz total progreso

- En el modelo de los labels para el ingreso de datos, Figura 54 se modifica el color de los botones de selección para el ingreso de documentos personales, se coloca dos colores que represente los documentos obligatorios y los documentos opcionales.

Figura 54. Cambio interfaz creación cliente

a. Cambios en el código

- En el menú de opciones se corrige el error de submenús desplegados al iniciar sesión en el aplicativo, se implementa el submenú con valores falso cuando se ingrese por primera vez al aplicativo y después de una interacción por parte del usuario en el submenú el mismo se despliegue u oculte.

- Se implementa una nueva lógica que restringa la navegación después de cerrado sesión, se implementa un protector de rutas que verifique si el usuario está dentro del aplicativo y si ha cerrado sesión, esto da mayor seguridad y protección de rutas.

- En el módulo de comisión por efectividad, se cambia los valores de la tabla de nuevos clientes con valores de clientes recompra y se cambia los valores de numéricos a monetarios en dólares.

- Para la distribución del ingreso de datos personales y documentos personales de ingreso de un nuevo cliente, se presenta en primer lugar el apartado para el ingreso de datos personales, completados los campos se oculta esta sección y se habilita la sesión para el ingreso de documentos personales.

3.2.5 Fase 5: Pruebas

En la fase final de la metodología Mobile-D, se realizaron pruebas para asegurar que la aplicación funcione correctamente y cumpla con las necesidades de los interesados, esto permitió entregar una versión final del aplicativo móvil que es estable y funciona.

Prueba de funcionalidad

El aplicativo móvil fue revisado y probado por el jefe del área de sistemas y por una parte de los asesores comercial, para garantizar un aplicativo estable y funcional, se realizaron las siguientes pruebas de funcionalidad.

- **Inicio de sesión**

Caso de prueba 1: Inicio de sesión sin los campos completos.

El aplicativo controla Figura 55 que se completen los dos campos requeridos como lo son usuario y contraseña para el ingreso al aplicativo.

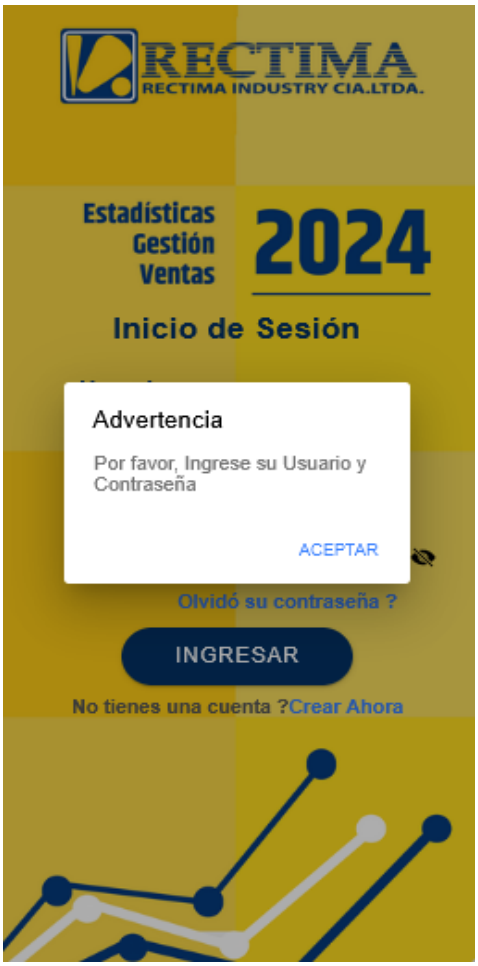


Figura 55. Validación campos completos

Caso de prueba 2: Credenciales incorrectas.

El aplicativo controla Figura 56 que el usuario tenga una cuenta creada con un usuario y contraseña válida para permitir el ingreso al aplicativo.

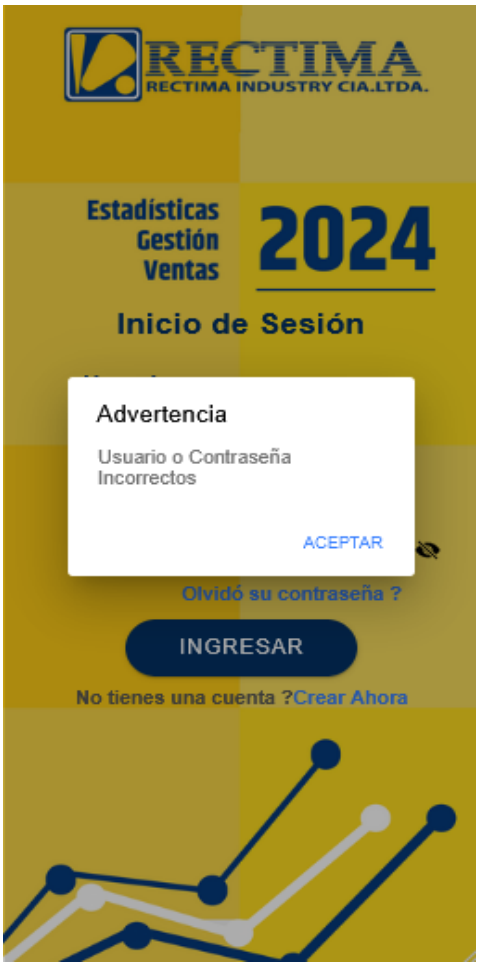


Figura 56. Validación usuario y contraseña correcto

- **Creación nuevo usuario**

Caso de prueba 1: Búsqueda de asesor comercial.

El aplicativo controla Figura 57 que el usuario ingrese un código para realizar una búsqueda de sus datos personales.

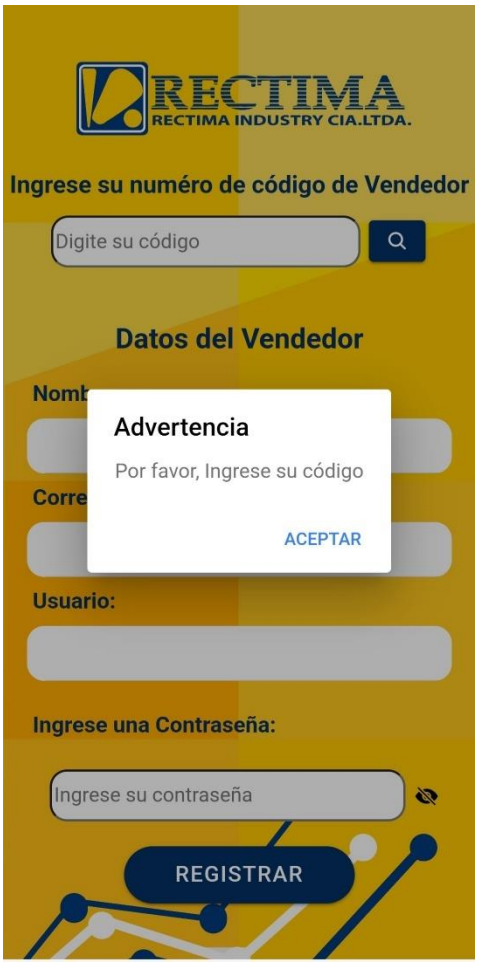


Figura 57. Validación ingreso código del asesor comercial

Caso de prueba 2: Validación existencia código asesor comercial.

El aplicativo controla Figura 58 que el código único de un asesor comercial exista para permitir la creación de una nueva cuenta.

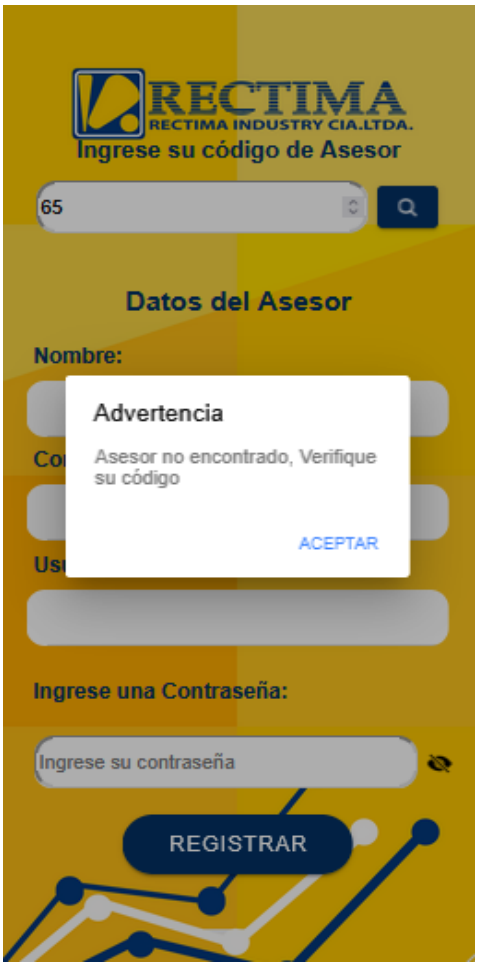


Figura 58. Validación existencia código asesor comercial

Caso de prueba 3: Validación ingreso contraseña.

El aplicativo móvil controla Figura 59 que el usuario digite una nueva contraseña para permitir la creación de la nueva cuenta.

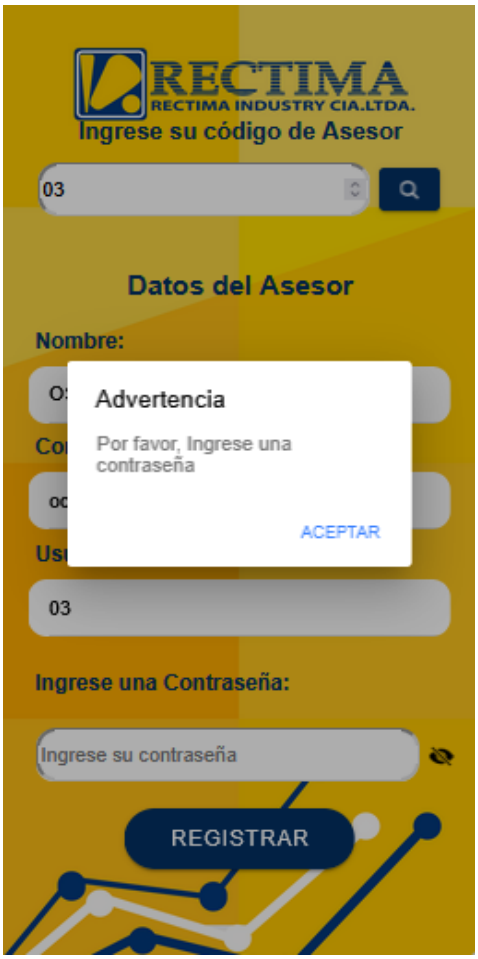


Figura 59. Validación ingreso contraseña

Caso de prueba 4: Usuario ya registrado.

El aplicativo móvil controla Figura 60 que los usuarios ya registrados no puedan volver a crear una cuenta si ya se encuentran registrados.

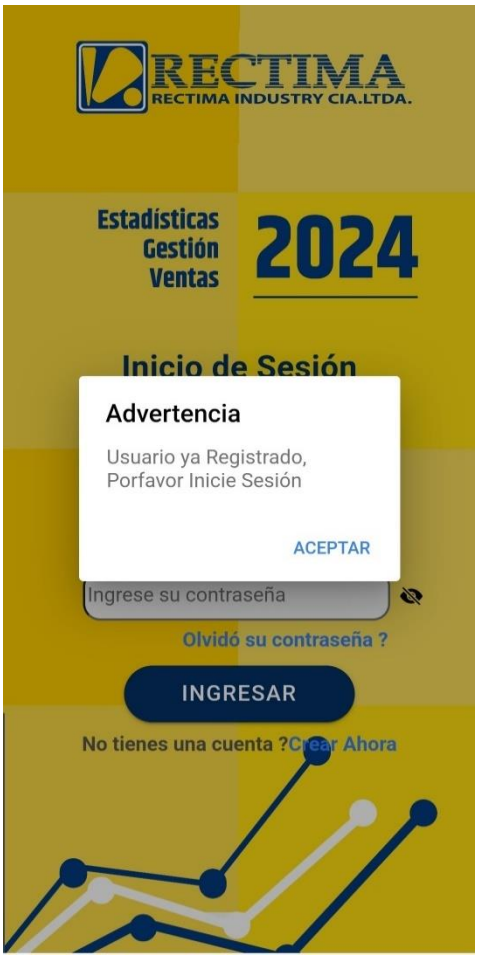


Figura 60. Validación usuario ya registrado

- **Cambio de contraseña**

Caso de prueba 1: Ingreso código asesor comercial.

El aplicativo móvil controla Figura 61 que se digite un código de asesor comercial para continuar con el proceso de cambio de contraseña.

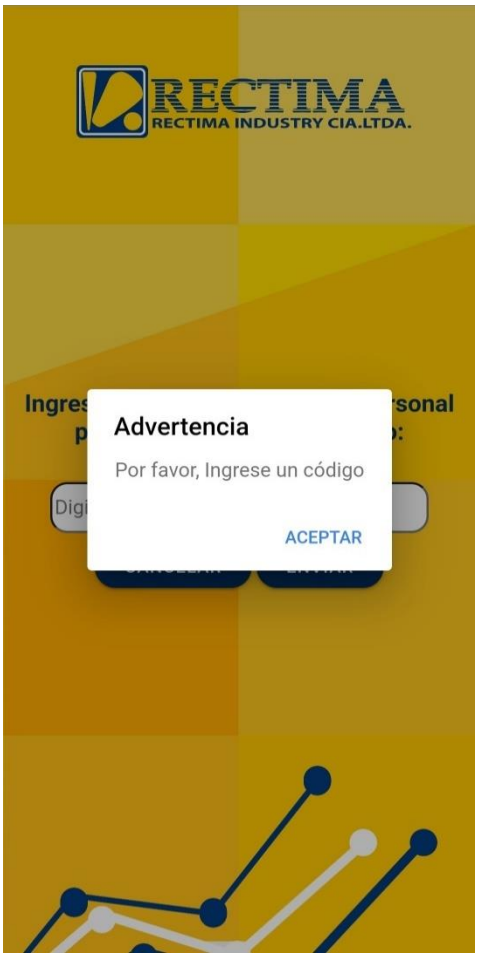


Figura 61. Validación ingreso código

Caso de prueba 2: Usuario registrado para cambio de contraseña.

El aplicativo móvil controla Figura 62 que el usuario se encuentre registrado para permitir el cambio de contraseña.

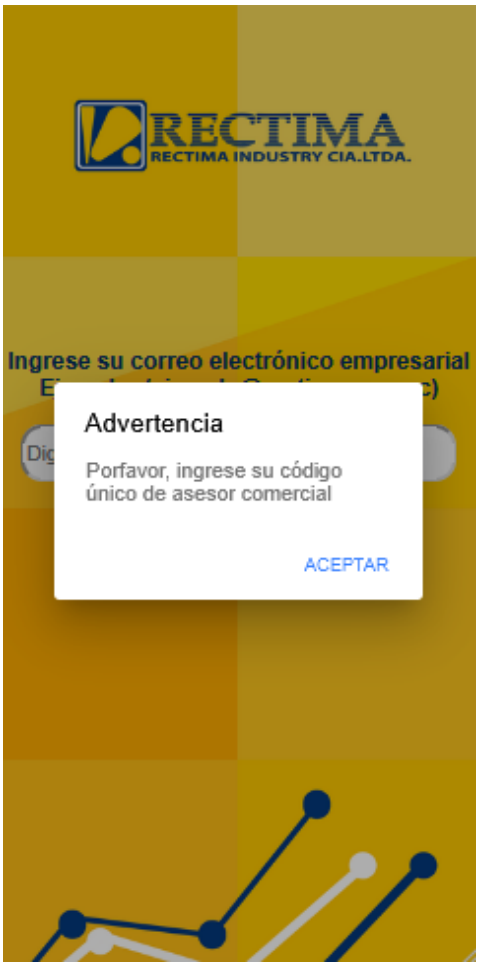


Figura 62. Validación registro usuario

Caso de prueba 3: Ingreso código de verificación.

El aplicativo móvil controla Figura 63 que el usuario ingrese el código de verificación que se envía al correo personal de cada asesor comercial.

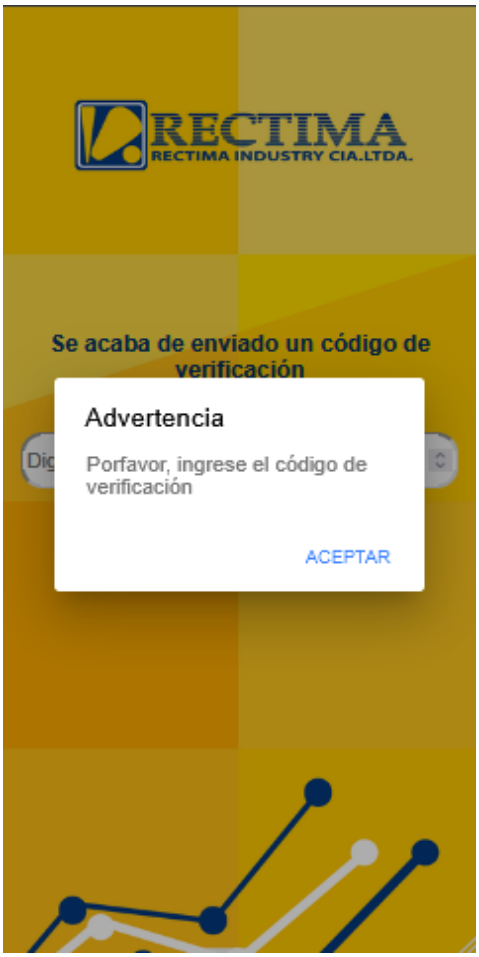


Figura 63. Validación ingreso código validación

Caso de prueba 4: Verificación código verificación.

El aplicativo móvil controla Figura 64 que el código ingresado por el usuario sea igual al código presente dentro del sistema.

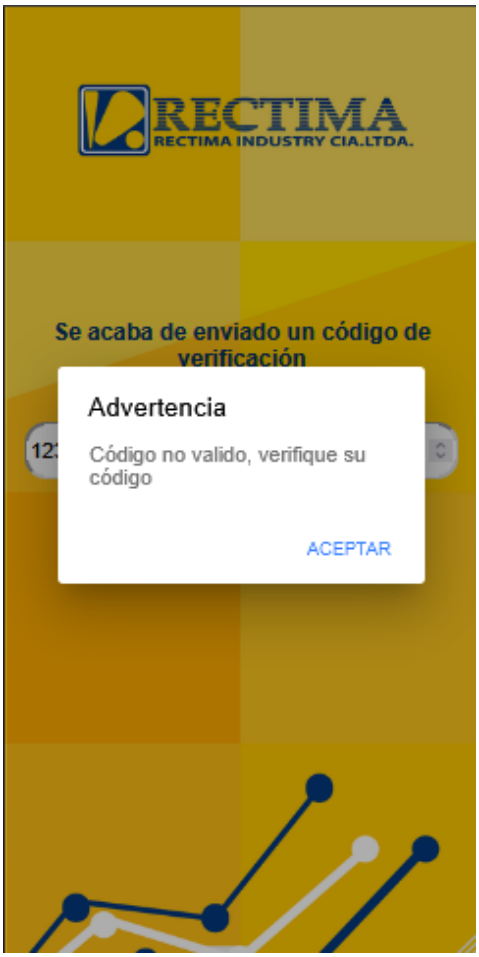


Figura 64. Validación código verificación

Caso de prueba 5: Contraseñas iguales.

El aplicativo móvil controla Figura 65 que la contraseña principal sea igual que la contraseña de verificación para permitir el cambio de contraseña.

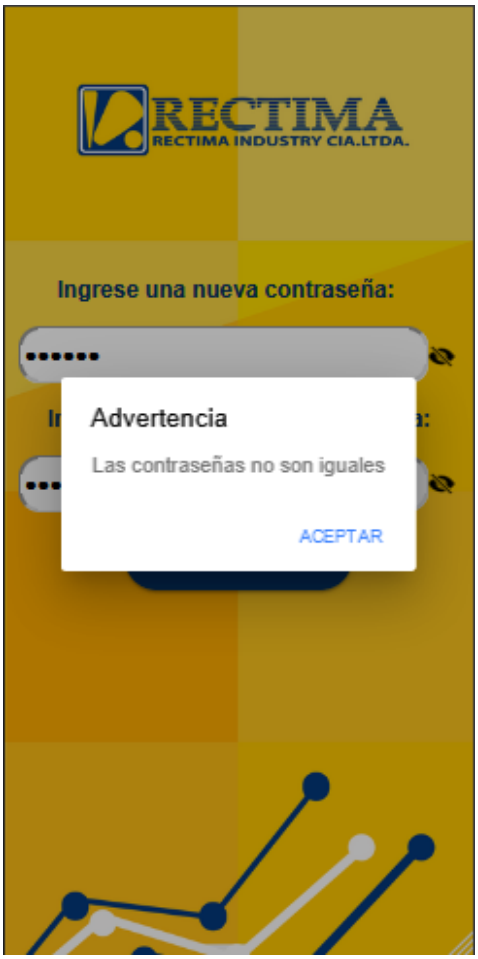


Figura 65. Validación contraseñas iguales

- **Total progreso**

Caso de prueba 1: Selección campos de fecha antes de realizar consulta.

El aplicativo móvil no controlaba que el usuario seleccione una fecha antes de realizar una consulta del total en progreso actual.

Por lo cual se realizó un cambio que permita controlar la selección efectiva de los datos de la fecha antes de poder realizar una búsqueda.

Después de realizar el cambio en el código el aplicativo móvil controla Figura 66 que el usuario seleccione una fecha antes de realizar una consulta del total en progreso actual.

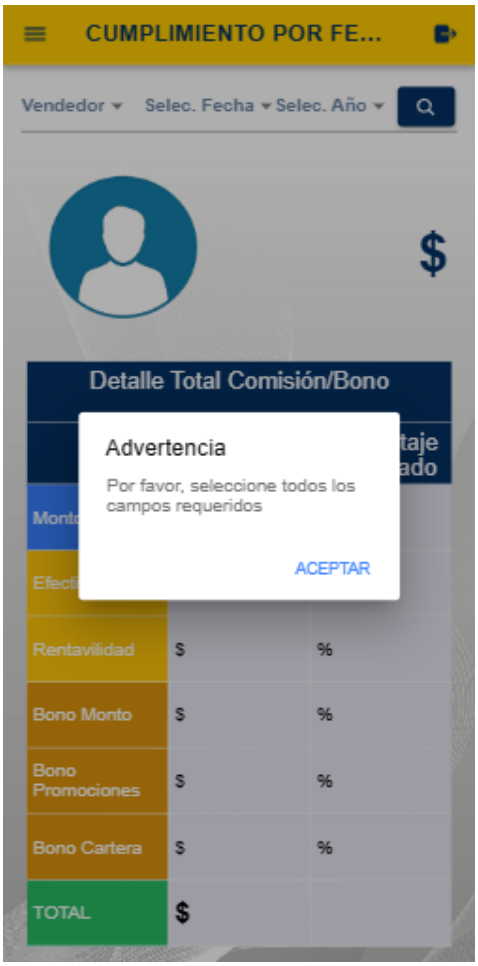


Figura 66. Validación selección campos fecha modificado

Caso de prueba 2: Selección fecha actual.

El aplicativo móvil controla Figura 67 que el usuario seleccione una fecha que este acorde al año presente.

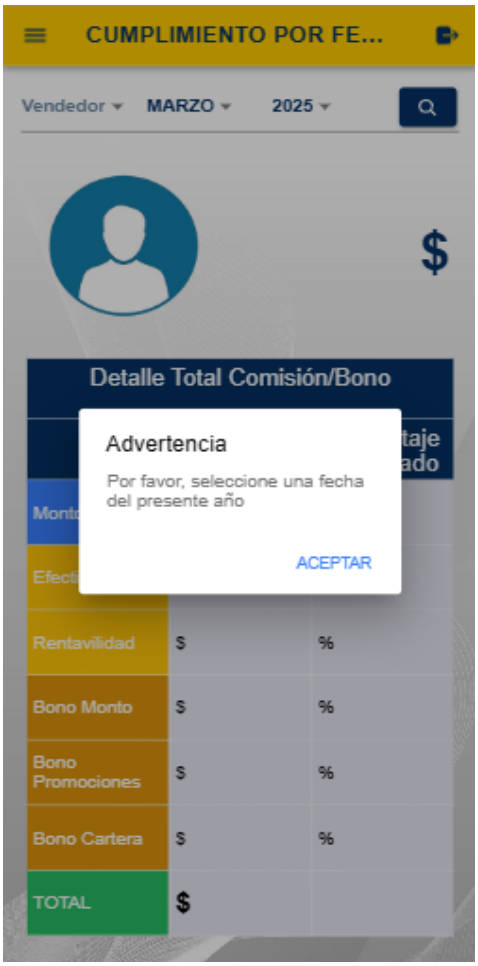


Figura 67. Validación selección fecha actual

- **Creación nuevo cliente**

Caso de prueba 1: Ingreso de datos personales requeridos.

El aplicativo móvil controla Figura 68 que el usuario ingrese todos los campos requeridos de los datos personales del nuevo cliente para continuar con el proceso de creación del nuevo cliente.

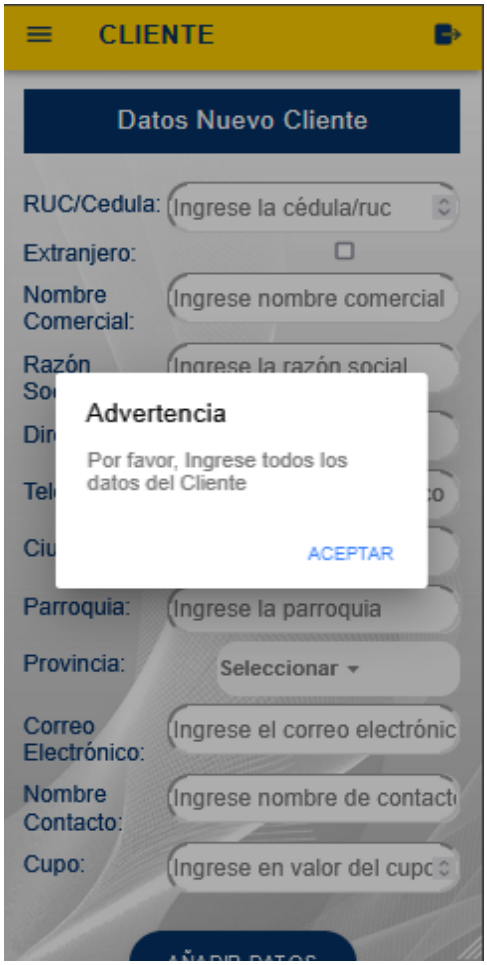


Figura 68. Validación ingreso datos personales nuevo cliente

Caso de prueba 2: Ingreso de documentos requeridos.

El aplicativo móvil controla Figura 69 que el usuario ingrese todos los documentos requeridos del nuevo cliente para poder realizar la creación del cliente.

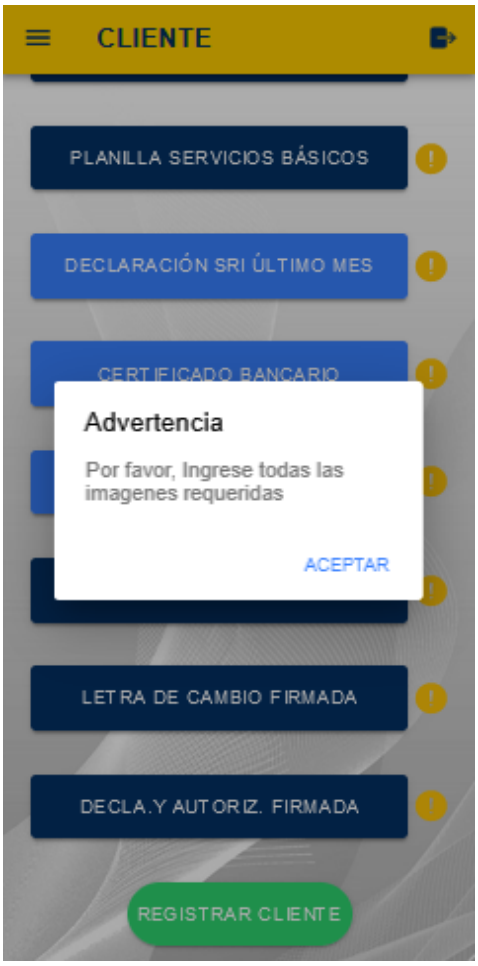


Figura 69. Validación ingreso documentos nuevo cliente

a. Pruebas de aceptación

Estas pruebas se realizaron con la gerente de la empresa, el jefe del área de sistemas y asesores comerciales para simular un entorno de producción y asegurar que la aplicación satisface las necesidades y expectativas de los usuarios. A continuación, en la Tabla 45 se presente el modelo utilizado para las pruebas de aceptación.

Tabla 45. Plantilla para pruebas de aceptación

Pruebas de aceptación	**Número: #**
N° de StoryCard: #	
Nombre:	
Descripción de la actividad:	
Condiciones de ejecución e interfaz:	
Resultado esperado:	
Resultado de la prueba:	

Tabla 46. Prueba aceptación 1

Pruebas de aceptación	**Número: 1**
N° de StoryCard: 1	
Nombre: Registro Usuario	
Descripción de la actividad: La función permite al usuario asesor el registro como un nuevo usuario del aplicativo móvil en caso de que se use por primera vez y el usuario cuente con un código de asesor comercial asignado.	
Condiciones de ejecución e interfaz: En la interfaz gráfica el usuario asesor puede ingresar su código, realizar una búsqueda de sus datos personales y crear una nueva contraseña que le permita el ingreso al aplicativo móvil.	
Resultado esperado: Si el usuario asesor ingresa un código valido se cargan sus datos personales, se permite el ingreso de una contraseña y se crea la nueva cuenta.	
Resultado de la prueba: Satisfactoria	

Tabla 47. Prueba aceptación 2

Pruebas de aceptación	**Número: 2**
N° de StoryCard: 2	
Nombre: Inicio de sesión	
Descripción de la actividad: La función permite al usuario acceder a los distintos módulos del aplicativo móvil.	
Condiciones de ejecución e interfaz: En la interfaz gráfica el usuario administrador o asesor puede ingresar un usuario y contraseña, se debe presionar en el botón de ingresar para que el aplicativo verifique las credenciales ingresadas, en caso de ser correctas se permita el acceso a la página de inicio del aplicativo móvil.	
Resultado esperado: Si el usuario administrador o asesor ingresa un usuario y contraseña correcto el aplicativo móvil automáticamente lo redirecciona a la página de inicio.	
Resultado de la prueba: Satisfactoria	

Tabla 48. Prueba aceptación 3

Pruebas de aceptación	**Número: 3**
N° de StoryCard: 3	
Nombre: Cambio de contraseña	
Descripción de la actividad: La función permite al usuario administrador o asesor cambiar su contraseña en caso de olvido o perdida.	
Condiciones de ejecución e interfaz: En la interfaz gráfica el usuario administrador o asesor puede ingresar su código o valor asignado para el usuario administrador, el código se valida y se realiza él envió de un correo electrónico con un código de verificación que se requiere para validar antes de permitir el cambio de contraseña, validado el código se permite el ingreso de una nueva contraseña que el aplicativo actualiza y redirige automáticamente a la página de inicio de sesión.	
Resultado esperado: Si el usuario administrador o asesor ingresa un código o valor asignado para el usuario administrador valido, un código de verificación valido y una contraseña valida se permite el cambio de contraseña.	
Resultado de la prueba: Insatisfactorio	

En esta prueba de aceptación la validación del código de verificación ingresado no se validaba de forma correcta lo que no permitía continuar al cambio de contraseña.

Para solventar el problema se cambión el tipo de almacenado del código de verificación pasando de una variable simple a una declaración de estado que permita tener un valor inicial que puede ser actualizado sin que el valor se pierda, todo esto dentro del código del aplicativo móvil.

Después del cambio realizado en el código el cambio de contraseña se puedo realizar de forma satisfactoria.

Tabla 49. Prueba aceptación 4

Pruebas de aceptación	**Número: 4**
N° de StoryCard: 4	
Nombre: Visualización **i**nicio	
Descripción de la actividad: La función permite al usuario administrador o asesor visualizar promocionales y presupuesto en bonificaciones.	
Condiciones de ejecución e interfaz: En la interfaz gráfica el usuario asesor puede revisar promocionales y presupuesto en bonificaciones, para el usuario administrador se presenta un mensaje de bienvenida y las imágenes de las promocionales.	
Resultado esperado: Si el usuario asesor ha ingresado satisfactoriamente al aplicativo móvil se mostrará imágenes de promocionales y el valor del presupuesto	

en bonificaciones, para el usuario administrador se mostrará una bienvenida e imágenes de promocionales.
Resultado de la prueba: Satisfactoria

Tabla 50. Prueba aceptación 5

Pruebas de aceptación	**Número: 5**
N° de StoryCard: 5	
Nombre: Visualización progreso actual	
Descripción de la actividad: La función permite al usuario administrador o asesor visualizar el detalla del total en ventas, presupuesto, porcentaje alcanzado, cumplimiento, efectividad, rentabilidad, bonos y número de nuevos clientes, el usuario administrador puede seleccionar al asesor, mes y año.	
Condiciones de ejecución e interfaz: En la interfaz gráfica el usuario administrador o asesor puede visualizar los valores en total en ventas, presupuesto, porcentaje alcanzado, cumplimiento, efectividad, rentabilidad, bonos y número de nuevos clientes, el usuario administrador puede revisar estos valores por cada año, mes y asesor comercial.	
Resultado esperado: Si el usuario administrador o asesor comercial han ingresado satisfactoriamente al aplicativo móvil se mostrará el valor del total en ventas, presupuesto, porcentaje alcanzado, cumplimiento, efectividad, rentabilidad, bonos y número de nuevos clientes, adicionalmente al usuario administrador se le permite seleccionar el nombre del asesor comercial que requiera consultar el valor alcanzado en progreso actual.	
Resultado de la prueba: Satisfactoria	

Tabla 51. Prueba aceptación 6

Pruebas de aceptación	**Número: 6**
N° de StoryCard: 6	
Nombre: Comisión por monto	
Descripción de la actividad: La función permite al usuario administrador o asesor visualizar el detalla de la venta neta, presupuesto, valor comisión y porcentaje a cumplir, el usuario administrador puede seleccionar al asesor, mes y año.	
Condiciones de ejecución e interfaz: En la interfaz gráfica el usuario administrador o asesor puede visualizar los valores de la venta neta, presupuesto, valor comisión y porcentaje a cumplir, el usuario administrador puede revisar estos valores por cada año, mes y asesor comercial.	
Resultado esperado: Si el usuario administrador o asesor comercial han ingresado satisfactoriamente al aplicativo móvil se mostrará el valor de la venta neta, presupuesto, valor comisión y porcentaje a cumplir, adicionalmente al usuario administrador se le permite seleccionar el nombre del asesor comercial que requiera consultar el valor alcanzado en comisión por monto.	
Resultado de la prueba: Satisfactoria	

Tabla 52. Prueba aceptación 7

Pruebas de aceptación	**Número: 7**
N° de StoryCard: 7	
Nombre: Comisión por efectividad	
Descripción de la actividad: La función permite al usuario administrador o asesor visualizar el detalla del valor alcanzado, clientes atendidos, venta cliente y alcance, nuevo cliente con número clientes nuevos, venta, alcance y cliente recompra con numero de recompra, venta y alcance.	
Condiciones de ejecución e interfaz: En la interfaz gráfica el usuario administrador o asesor puede visualizar los valores de clientes atendidos, venta cliente y alcance, nuevo cliente con número clientes nuevos, venta, alcance y cliente recompra con numero de recompra, venta y alcance, el usuario administrador puede revisar estos valores por cada año, mes y asesor comercial.	
Resultado esperado: Si el usuario administrador o asesor comercial han ingresado satisfactoriamente al aplicativo móvil se mostrará el valor alcanzado, clientes atendidos, venta cliente y alcance, nuevo cliente con número clientes nuevos, venta, alcance y cliente recompra con numero de recompra, venta y alcance, adicionalmente al usuario administrador se le permite seleccionar el nombre del asesor comercial que requiera consultar el valor alcanzado en comisión por efectividad.	
Resultado de la prueba: Satisfactoria	

Tabla 53. Prueba aceptación 8

Pruebas de aceptación	**Número: 8**
N° de StoryCard: 8	
Nombre: Comisión por rentabilidad	
Descripción de la actividad: La función permite al usuario administrador o asesor visualizar el detalla del valor alcanzado, porcentaje y rentabilidad objetivo.	
Condiciones de ejecución e interfaz: En la interfaz gráfica el usuario administrador o asesor puede visualizar el valor alcanzado, porcentaje y rentabilidad objetivo, el usuario administrador puede revisar estos valores por cada año, mes y asesor comercial.	
Resultado esperado: Si el usuario administrador o asesor comercial han ingresado satisfactoriamente al aplicativo móvil se mostrará el valor alcanzado, porcentaje y rentabilidad objetivo, adicionalmente al usuario administrador se le permite seleccionar el nombre del asesor comercial que requiera consultar el valor del progreso en bono por rentabilidad.	
Resultado de la prueba: Satisfactoria	

Tabla 54. Prueba aceptación 9

Pruebas de aceptación	**Número: 9**
N° de StoryCard: 9	
Nombre: Bono por monto	
Descripción de la actividad: La función permite al usuario administrador o asesor visualizar el detalla del valor obtenido, venta, presupuesto y porcentaje cumplimiento.	
Condiciones de ejecución e interfaz: En la interfaz gráfica el usuario administrador o asesor puede visualizar el valor obtenido, venta, presupuesto y porcentaje cumplimiento, el usuario administrador puede revisar estos valores por cada año, mes y asesor comercial.	
Resultado esperado: Si el usuario administrador o asesor comercial han ingresado satisfactoriamente al aplicativo móvil se mostrará el valor obtenido, venta, presupuesto y porcentaje cumplimiento, adicionalmente al usuario administrador se le permite seleccionar el nombre del asesor comercial que requiera consultar el valor alcanzado en bono por monto.	
Resultado de la prueba: Satisfactoria	

Tabla 55. Prueba aceptación 10

Pruebas de aceptación	**Número: 10**
N° de StoryCard: 10	
Nombre: Bono por promocional	
Descripción de la actividad: La función permite al usuario administrador o asesor visualizar el detalla del porcentaje de comisión, valor obtenido, venta promoción y presupuesto promoción.	
Condiciones de ejecución e interfaz: En la interfaz gráfica el usuario administrador o asesor puede visualizar el porcentaje de comisión, valor obtenido, venta promoción y presupuesto promoción, el usuario administrador puede revisar estos valores por cada año, mes y asesor comercial.	
Resultado esperado: Si el usuario administrador o asesor comercial han ingresado satisfactoriamente al aplicativo móvil se mostrará el porcentaje de comisión, valor obtenido, venta promoción y presupuesto promoción, adicionalmente al usuario administrador se le permite seleccionar el nombre del asesor comercial que requiera consultar el valor alcanzado en bono por promocional.	
Resultado de la prueba: Satisfactoria	

Tabla 56. Prueba aceptación 11

Pruebas de aceptación	**Número: 11**
N° de StoryCard: 11	
Nombre: Bono por cartera	
Descripción de la actividad: La función permite al usuario administrador o asesor visualizar el detalla de la efectiva en cobranza, cumplimiento presupuesto, detalle mayor al 110% con la venta cobrada, valor cartera y detalle de efectivizado con la cobranza en plazo y cobranza fuera de plazo.	

Condiciones de ejecución e interfaz: En la interfaz gráfica el usuario administrador o asesor puede visualizar la efectiva en cobranza, cumplimiento presupuesto, detalle mayor al 110% con la venta cobrada, valor cartera y detalle de efectivizado con la cobranza en plazo y cobranza fuera de plazo, el usuario administrador puede revisar estos valores por cada año, mes y asesor comercial.
Resultado esperado: Si el usuario administrador o asesor comercial han ingresado satisfactoriamente al aplicativo móvil se mostrará la efectiva en cobranza, cumplimiento presupuesto, detalle mayor al 110% con la venta cobrada, valor cartera y detalle de efectivizado con la cobranza en plazo y cobranza fuera de plazo, adicionalmente al usuario administrador se le permite seleccionar el nombre del asesor comercial que requiera consultar el valor alcanzado en bono por cartera.
Resultado de la prueba: Satisfactoria

Tabla 57. Prueba aceptación 12

Pruebas de aceptación	**Número: 12**
N° de StoryCard: 12	
Nombre: Total progreso	
Descripción de la actividad: La función permite al usuario administrador o asesor visualizar el detalla del total de montos alcanzados en comisiones, bonos, comisión por monto, efectividad, rentabilidad, bono por montos, total alcanzado y permitir la descarga de archivo pdf con los valores obtenidos durante el mes.	
Condiciones de ejecución e interfaz: En la interfaz gráfica el usuario administrador o asesor puede visualizar el total de montos alcanzados en comisiones, bonos, comisión por monto, efectividad, rentabilidad, bono por montos, total alcanzado y permitir la descarga de archivo pdf con los valores obtenidos durante el mes, el usuario administrador puede revisar estos valores por cada año, mes y asesor comercial.	
Resultado esperado: Si el usuario administrador o asesor comercial han ingresado satisfactoriamente al aplicativo móvil se mostrará el total de montos alcanzados en comisiones, bonos, comisión por monto, efectividad, rentabilidad, bono por montos, total alcanzado y se permite la descarga de archivo pdf con los valores obtenidos durante el mes, adicionalmente al usuario administrador se le permite seleccionar el nombre del asesor comercial que requiera consultar el valor alcanzado en total de progreso comisiones.	
Resultado de la prueba: Satisfactoria	

Tabla 58. Prueba aceptación 13

Pruebas de aceptación	**Número: 13**
N° de StoryCard: 13	
Nombre: Creación nuevo cliente	
Descripción de la actividad: La función permite al usuario asesor registrar nuevos clientes, el ingreso de datos y documentos personales del nuevo cliente.	
Condiciones de ejecución e interfaz: En la interfaz gráfica el usuario asesor puede registrar nuevos clientes, ingresar datos y documentos personales del nuevo cliente.	

Resultado esperado: Si el usuario asesor comercial ha ingresado satisfactoriamente al aplicativo móvil se permite el registrar de datos y documentos personales del nuevo cliente.
Resultado de la prueba: Satisfactoria

Tabla 59. Prueba aceptación 14

Pruebas de aceptación	**Número: 14**
N° de StoryCard: 14	
Nombre: Ventas Clientes	
Descripción de la actividad: La función permite al usuario administrador o asesor comercial visualizar la venta por cada cliente y los productos vendidos por cada cliente.	
Condiciones de ejecución e interfaz: En la interfaz gráfica el usuario administrador o asesor puede visualizar las ventas por cada cliente y los productos vendidos a cada cliente, el usuario administrador puede revisar estos valores por cada asesor comercial.	
Resultado esperado: Si el usuario administrador o asesor comercial han ingresado satisfactoriamente al aplicativo móvil se mostrará las ventas por cada cliente y los productos comprados por cada cliente, adicionalmente al usuario administrador se le permite seleccionar el nombre del asesor comercial que requiera consultar las ventas por cada cliente.	
Resultado de la prueba: Satisfactoria	

3.2.6 Implantar aplicativo móvil

Una vez concluido el desarrollo de las fases de la metodología Mobile-D, se implanto el aplicativo móvil en la Empresa Rectima Industry Cia. Ltda.

- **Instalación**

Para que los usuarios puedan acceder al aplicativo móvil se procedió a subir el aplicativo de instalación en Amazon AWS S3 el cual está disponible en el siguiente link.

- **Link descarga aplicativo móvil**

https://aplicativoventas.s3.us-east-2.amazonaws.com/VentasApp.apk

Una vez descargado el archivo se procede a instalar el aplicativo móvil en los distintos dispositivos móviles de cada asesor comercial.

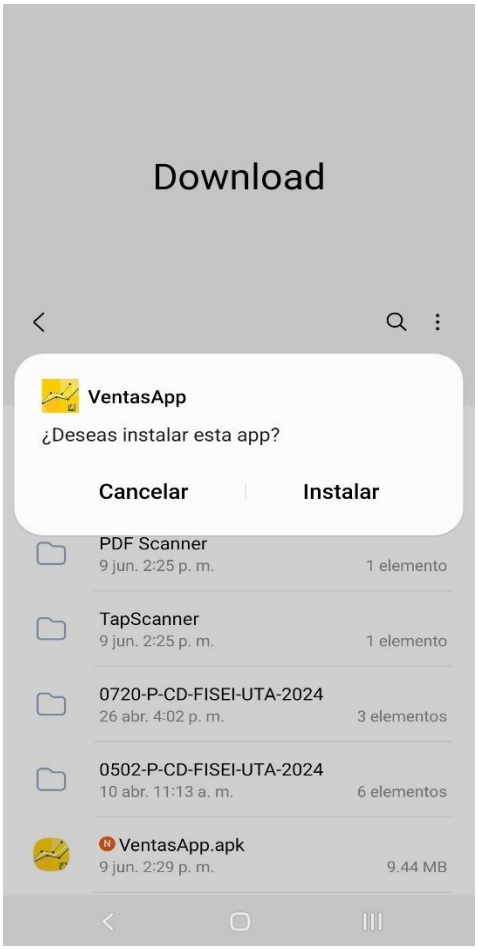


Figura 70. Instalación aplicativo móvil

Antes de la instalación los dispositivos realizan un análisis de seguridad del aplicativo móvil.

Figura 71. Análisis aplicativo móvil

Superado el análisis previo del aplicativo móvil se permite la instalación en el dispositivo móvil.

Figura 72. Validación análisis aplicativo móvil

Instalación correcta del aplicativo móvil que ya permite su uso.

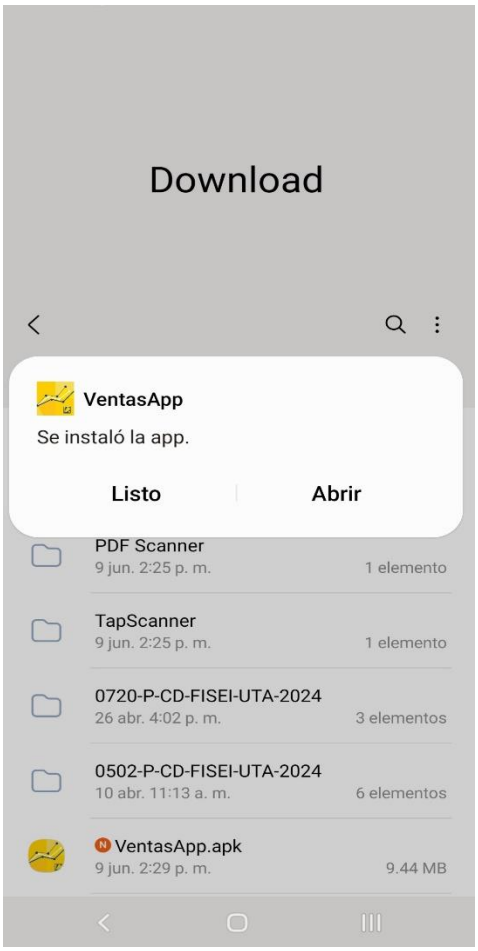


Figura 73. Terminación instalación aplicativo móvil

Figura 74. Aplicativo móvil instalado

- **Capacitación**

Para finalizar la implementación del aplicativo móvil se realizó una capacitación a los asesores comerciales, en donde se explicó el funcionamiento a detalle del aplicativo móvil, el cronograma de capacitación se detalla a continuación en la Tabla 60.

Capacitador: Henry Chimborazo

Usuarios: Asesores comerciales

Fecha: 11/05/2024

Tabla 60. Actividades capacitación

Actividad	**Descripción**
Inicio de sesión	Presentación interfaz y proceso para el ingreso al aplicativo
Crear usuario	Presentación interfaz y proceso para la creación de un nuevo usuario
Cambio de contraseña	Presentación interfaz y proceso para el cambio de contraseña
Inicio	Presentación interfaz y elementos presentes
Progreso actual	Presentación interfaz, explicación valores y elementos presentes
Comisión por monto	Presentación interfaz, explicación valores y elementos presentes
Comisión por efectividad	Presentación interfaz, explicación valores y elementos presentes
Comisión por Rentabilidad	Presentación interfaz, explicación valores y elementos presentes
Bono por monto	Presentación interfaz, explicación valores y elementos presentes

Bono por promocional	Presentación interfaz, explicación valores y elementos presentes
Bono por cartera	Presentación interfaz, explicación valores y elementos presentes
Total progreso	Presentación interfaz, explicación valores y elementos presentes
Nuevo cliente	Presentación interfaz y proceso para creación de un nuevo cliente
Ventas Clientes	Presentación interfaz, explicación valores y gráficos presentes

Las capacitaciones se complementan con el manual de usuario presente en el Anexo O.

- **Estadísticas aplicativo móvil**

En la Tabla 61 se presentan los valores obtenidos en cumplimientos de presupuestos durante los meses Enero, Febrero y Marzo y en la Tabla 62 se presentan los valores obtenidos durante los meses Abril, Mayo y Junio.

Tabla 61. Cumplimiento presupuesto antes del aplicativo móvil

Meses	Presupuesto	Venta Real	Porcentaje Alcanzado
Enero	$830,598.08	$696,463.00	83.85%
Febrero	$826,836.55	$679,735.90	82.21%
Marzo	$753,535.26	$620,639.75	82.36%

Tabla 62. Cumplimiento presupuesto después del aplicativo móvil

Meses	Presupuesto	Venta Real	Porcentaje Alcanzado
Abril	$872,456.10	$782,322.15	89.67%
Mayo	$906,800.26	$832,821.00	91.84%
Junio	$853,910.12	$799,959.41	93.68%

En la Figura 75 se plasman los valores porcentuales de cumplimiento en ventas obtenidos durante los meses de Enero a Junio, generando una comparación entre los valores de los tres primeros meses antes de la implantación del aplicativo móvil y los tres meses después de la implantación del aplicativo móvil.

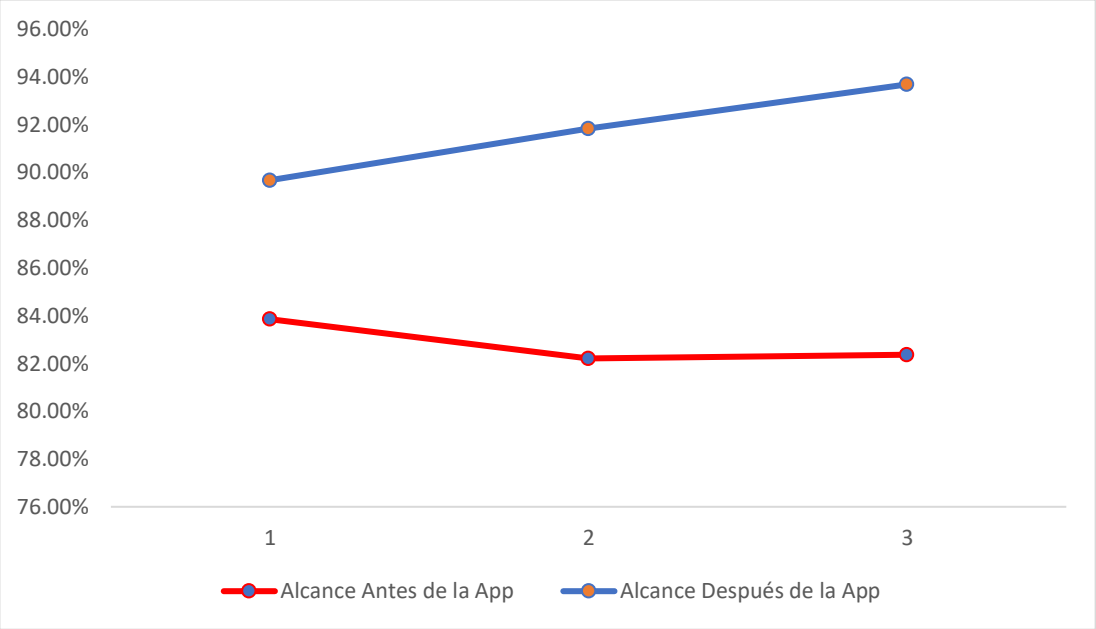


Figura 75. Porcentaje cumplimientos ventas antes y después

Al realizar la comparación de las tendencias del porcentaje en cumplimiento antes y después de la implantación del aplicativo móvil se refleja un aporte positivo en el incremento del cumplimiento en presupuesto de ventas.

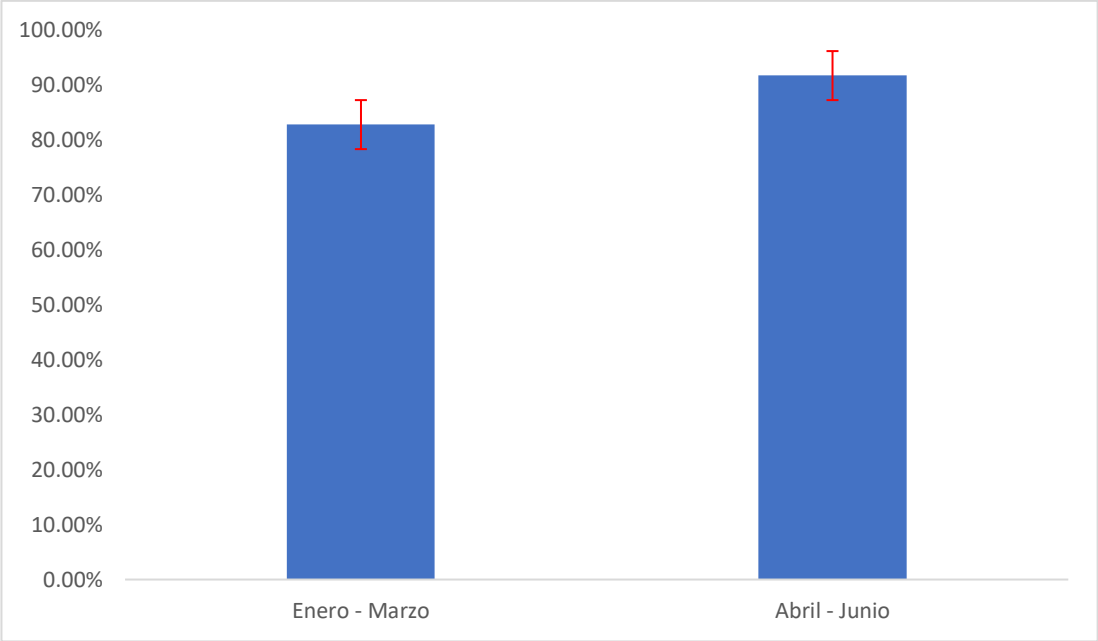


Figura 76. Comparación media y desviación estándar

En la Figura 76 se muestran las medidas de dos grupos tanto de los meses Enero a Marzo con una media de 82.81% y una desviación estándar de 0.009066605 y de los meses Abril a Junio con un media de 91.73% y una desviación estándar de 0.020072618, con estos valores se calcula los intervalos de confianza tanto positivos como negativos con un nivel de confianza del 95%, para el primer grupo el intervalo de confianza positivo refleja un 83.05% contra un intervalo de confianza negativo del 82.57%, para el segundo grupo el intervalo de confianza positivo refleja un 92.26% contra un intervalo de confianza negativo del 91.20%, las diferencias en las medidas y la ausencia de solapamiento indican una referencia estadística significativa entre los grupos y refuerza el aporte que el aplicativo móvil para gestión y comunicación de presupuestos ha tenido en la Empresa Rectima Industry Cia. Ltda.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Al identificar y evaluar cada etapa del proceso de planificación en presupuestos de ventas, desde la recopilación de datos, la proyección de ventas, asignación de recursos y el monitoreo de desempeño, se logra una visión integral que facilita la creación del aplicativo. Este enfoque no solo gestiona los presupuestos en ventas si no que permite una toma de decisiones comerciales más informada que permite optimizar los recurso y tiempo, además de potenciar la capacidad de respuesta a cambios en el mercado.

- Al explorar las ventajas que ofrece Ionic, como su capacidad para crear aplicaciones multiplataforma, su extensa biblioteca de componentes y su integración con herramientas modernas, se puede determinar cómo estas características pueden optimizar el proceso y mejorar la calidad en el desarrollo de aplicativos móviles.

- El Framework Ionic proporciona una base sólida para el desarrollo de una aplicación móvil robusta y eficiente. Esto permite aprovechar al máximo las capacidades de Ionic, asegurando que la aplicación no solo cumpla con los requisitos técnicos, sino que también ofrezca una buena experiencia de usuario mediante los requisitos no funcionales, contribuyendo al éxito del proyecto de gestión de presupuestos de ventas.

- La implantación de un aplicativo móvil enfocado en la gestión y comunicación de presupuestos en ventas favorece a la Empresa Rectima Industry Cia. Ltda., al proporcionar información en tiempo real que facilita la comunicación de los datos necesarios sobre presupuestos, comisiones, bonificaciones y ventas que ha contribuido para el aumento en el porcentaje en cumplimiento de ventas, pasando de un promedio de un 82.81% a un 91.73% evidenciado un impacto positivo de la tecnología móvil en la gestión de presupuestos.

4.2 Recomendaciones

- Se sugiere implementar indicadores de desempeño para monitorear la eficiencia y efectividad de los procesos de planificación, la capacitación continua del personal, siempre implementar las mejores prácticas de planificación de presupuestos, revisiones y ajustes continuos en función de los resultados obtenidos y los cambios en el mercado.

- Es importante evaluar características clave de Ionic, como su capacidad para crear aplicaciones multiplataforma, el rendimiento de las aplicaciones, la facilidad de uso y la integración con otras tecnologías y herramientas de desarrollo. Comparar Ionic con otros Frameworks móviles proporciona una perspectiva más amplia y ayudar a determinar si es la mejor opción para el desarrollo de un proyecto.

- Es recomendable utilizar una interfaz web complementaria para la creación de clientes que funcione en paralelo con el aplicativo móvil desarrollado. Esto permite que los usuarios gestionen la información desde cualquier dispositivo con acceso a internet, facilitando el registro de nuevos datos. La funcionalidad web puede ser especialmente útil para los equipos de ventas que trabajan desde una oficina o un entorno con acceso a una computadora.

- Se recomienda programar actualizaciones regulares para agregar nuevas características al aplicativo móvil, monitorear el rendimiento en tiempo real, gestionar la seguridad, ofrecer soporte continuo a los usuarios y recopilar comentarios para identificar áreas de mejora futuras.

REFERENCIAS BIBLIOGRÁFICAS

[1] P.-O. Gourinchas, «La economía mundial se prepara para un aterrizaje suave, pero persisten los riesgos,» 2024.

[2] R. B. López, «La Economía Digital y Desarrollo Económico,» 2023.

[3] M. C. A. Josue, *Aplicación multiplataforma con Framework Ionic para apoyar el e-turismo de las Pymes del sector gastronómico tradicional de la ciudad de Ambato.*, 2023.

[4] A. S. D. Israel, *Aplicación móvil utilizando el framework Ionic para la difusión de la oferta alimenticia estudiantil de los restaurantes aledaños a la Universidad Técnica de Ambato,* 2023.

[5] C. R. C. David y E. A. D. Adrian, *Aplicación móvil multiplataforma para el control de inventario en la Boutique Nicoleth, utilizando la plataforma Xamarin,* 2023.

[6] C. A. N. Magaly, *Plan de marketing de la lotización Flores de Oriente ubicada en el cantón Lago Agrio,* 2022.

[7] G. C. P. Israel, «Aplicación web-movil para la gestión de ventas y marketing de la empresa "Segindsa",» 2023.

[8] B. G. P. Andreina y P. C. L. Maritza, *"Desarrollo de una aplicación móvil multiplataforma para la gestión de la información de la compañía de tricimotos Contrilamana",* 2023.

[9] D. Ramos, R. Noriega, J. R. Laínez, A. Durango y I. T. C. Academy, Curso de Ingeniería de Software : 2ª Edición, Createspace Independent Publishing Platform, p. 340.

[10] V. Arias, «Plan de mejora para la gestión de la información del direccionamiento estratégico del Ministerio de Relaciones Exteriores y su Fondo Rotatorio,» de *Tesis de maestría*, 2021.

[11] M. E. Gabriel y P. Julián, «Metodologías de desarrollo de software».

[12] J. R. M. Ríos, J. A. Honores-Tapia, N. Pedreira-Souto y H. P. Pardo-León, Comparativa de metodologías de desarrollo de aplicaciones móviles, vol. 10, Area de Innovacion y Desarrollo, S.L. 3 Ciencias, 2021.

[13] V. Avilés y M. Valeria, *ESTUDIO COMPARATIVO DE LAS APLICACIONES MÓVILES HÍBRIDASDESARROLLADAS CON EL LENGUAJE DE PROGRAMACIÓNJAVASCRIPT, Y LAS APLICACIONES MÓVILES NATIVAS,* 2022.

[14] C. Gavilánez y D. Marisol, *Análisis comparativo entre sistemas operativos de dispositivos móviles ANDROID, IPHONE OS y MIUI,* 2023.

[15] M. A. Z. A. JHONATAN DAVID CHOTO, *DESARROLLO DE UNA APLICACIÓN MÓVIL INFORMATIVADE RUTAS PARA LA COMPAÑÍA DE TRANSPORTE TORRESREVELO S.A. EMPLEANDO EL FRAMEWORK IONIC,* 2020.

[16] H. Molina, J. Ignacio, G. Castro y D. Stefanie, *Análisis comparativo de Jamstack VS NODE.JS en el desarrollo de páginas y aplicaciones web,* 2021.

[17] W. C. Esquivel y G. L. Sevilla, *Paralelismos entre bases de datos relacionales y norelacionales (un enfoque en seguridad),* 2021.

[18] F. I. M. L. y R. I. Rodríguez Ponce, «Gestión de la información para analizar la actividad gerencial y la planificación financiera de Portoparques,» vol. 16, May 2023.

[19] P. Burdiles y A. Pommier, «EL TRIAJE EN PANDEMIA: FUNDAMENTOS ÉTICOS PARA LA ASIGNACIÓN DE RECURSOS DE SOPORTE VITAL AVANZADO EN ESCENARIOS DE ESCASEZ,» *Revista Médica Clínica Las Condes,* vol. 32, pp. 61-74, January 2021.

[20] L. Zampetti, «Toma de Decisiones, Funciones Ejecutivas y Emoción: Una revisión de los modelos teóricos,» *Analogías Del Comportamiento,* vol. 1, p. 53–63, 2023.

[21] A. Chero Fernández, *PRESUPUESTOS,* 2020.

[22] «Ionic Framework,» [En línea]. Available: https://ionicframework.com/docs/v7/.

[23] «Node.js,» [En línea]. Available: https://nodejs.org/docs/latest/api/.

ANEXOS

Anexo A. Código registro usuario

```
const CrearCuenta: React.FC = () => {
  const history = useHistory();
  let [showLoading, setShowLoading] = useState(false);
  let [showLoadingBus, setShowLoading2] = useState(false);
  let [codigo, setCodigo] = useState('');
  let [password, setPassword] = useState('');
  let [showPassword, setShowPassword] = useState(false);
  let [showAlert, setShowAlert] = useState(false);
  let [showAlertPass, setShowAlertPass] = useState(false);
  let [showAlertCre, setShowAlertCre] = useState(false);
  let [alertMessage, setAlertMessage] = useState('');
  let [errorPassword, setErrorPassword] = useState('');
  let [tipoUsuario, setTipoUsuario] = useState('Seleccionar');
  let url = 'https://siav.rectima.com.ec:3001/'
  let [datosUsuario, setDatosUsuario] = useState({
    VEN_Codigo: '',
    VEN_Nombre: '',
    correo: ''
  })
  let toggleShowPassword = () => {
    setShowPassword(!showPassword);
  }
  // Función para manejar el cambio de la selección
  let handleTipoUsuario = (e: CustomEvent) => {
    let selectedValue = e.detail.value;
    setTipoUsuario(selectedValue)
  }
  // Funciones para obtener valores individuales
  const obtenerNombre = () => {
    return datosUsuario.VEN_Nombre;
  }

  const obtenerCorreo = () => {
    return datosUsuario.correo;
  }

  const obtenerUsuario = () => {
    return datosUsuario.VEN_Codigo;
  }
  let validarPassword = (e: any) => {
    let value = e.target.value;
    // Expresión regular para validar la contraseña
    const regex = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-
Za-z\d@$!%*?&]{5,}$/;
    // Verificar si la longitud es menor que 10 o si no son todos
números
    if (regex.test(value) || value === '') {
      setErrorPassword('');
    } else {
```

```
      setErrorPassword('La contraseña debe contener mínimo cinco
caracteres, una letra mayúscula, minúscula, un número y un
carácter');
    }
    setPassword(value);
  };

  let handleLogin = async () => {
    if (!codigo) {
      setAlertMessage('Por favor, ingrese su código empresarial');
      setShowAlert(true);
      return;
    }
    setShowLoading2(true);
    try {
      // Realizar la petición al servidor
      let response = await
axios.post('http://siav.rectima.com.ec:3001/buscarUsuario', {
        codigo
      });
      // Verificar la respuesta del servidor
      if (response.data.success) {
        // Obtención de datos exitosamente
        let { VEN_Codigo, VEN_Nombre, correo } = response.data.user;
        setDatosUsuario({
          VEN_Codigo,
          VEN_Nombre,
          correo
        })
      } else {
        // Inicio de sesión fallido, mostrar modal de alerta
        setAlertMessage('Asesor no encontrado, por favor verifique
su código e intentelo de nuevo')
        setShowAlert(true)
      }
    } catch (error: any) {
      // Manejar errores de la petición
      console.error('Error en la petición:', error.message)
      setAlertMessage('Asesor no encontrado, por favor verifique su
código e intentelo de nuevo')
      setShowAlert(true)
    }
    finally {
      setShowLoading2(false) // Ocultar el componente IonLoading al
finalizar la petición
    }
  }

  let handleRegistro = async () => {
    if (obtenerNombre() == '') {
      setShowLoading(false)
      setAlertMessage('Por favor, primero obtenga sus datos con su
código empresarial')
      setShowAlert(true)
      return
    }
```

```
    try {
      if (!password) {
        setShowLoading(false)
        setAlertMessage('Por favor ingrese una contraseña')
        setShowAlert(true)
        return
      }
      setShowLoading(true)
      // Realizar la petición al servidor
      let response = await
axios.post('http://siav.rectima.com.ec:3001/comprobarUsuario', {
        codigo
      })
      // Verificar la respuesta del servidor
      if (response.data.success) {
        // Inicio de sesión fallido, mostrar modal de alerta
        setShowLoading(false)
        setAlertMessage('Usuario ya registrado, por favor inicie
sesión')
        setShowAlert(true)
        history.push('/login')
      }
    } catch (error: any) {
      try {
        let NOMBRE = obtenerNombre()
        let CORREO = obtenerCorreo()
        let USUARIO = obtenerUsuario()
        let Password = md5(password)
        tipoUsuario = 'Asesor'
        let responseAdd = await
axios.put(url, {
          USUARIO, NOMBRE, Password, CORREO, tipoUsuario
        })
        if (responseAdd.data.success) {
          // Creación correcta
          setShowLoading(false)
          setAlertMessage('Usuario registrado correctamente')
          setShowAlert(true)
          history.push('/login')
        }
      } catch (error: any) {
        console.error('Error en la petición:', error.message)
        setShowLoading(false)
        setAlertMessage('Se produjo un error al crear la nueva
cuenta intentelo de nuevo mas tarde')
        setShowAlert(true)
        history.push('/login')
      }
    }

  }
}
```

Anexo B. Código inicio sesión

```
const LoginPage: React.FC = () => {
  const history = useHistory()
  let [showLoading, setShowLoading] = useState(false)
  let [showPassword, setShowPassword] = useState(false)
  let [username, setUsername] = useState('')
  let [password, setPassword] = useState('')
  let [showAlert, setShowAlert] = useState(false)
  let url = 'https://siav.rectima.com.ec:3001/'
  let [alertMessage, setAlertMessage] = useState('')
  let { setUsersname, setNombre, setTipoUsuario } = useUser()

  let toggleShowPassword = () => {
    setShowPassword(!showPassword);
  }

  let handleLogin = async () => {
    try {
      if (!username || !password) {
        setAlertMessage('Por favor, Ingrese su Usuario y
Contraseña')
        setShowAlert(true)
        return
      }
      setShowLoading(true)
      // Realizar la petición al servidor
      let Password = md5(password)
      let response = await axios.post(url, {
        username,
        Password,
      });
      // Verificar la respuesta del servidor
      if (response.data.success) {
        // Inicio de sesión exitoso, redirigir a otra página
        let { usuario, tipo_usuario, nombre } = response.data.user;
        let { token } = response.data.token;
        // Almacenar el token en el almacenamiento local del
navegador
        localStorage.setItem('token', token)
        setUsersname(usuario)
        setNombre(nombre)
        setTipoUsuario(tipo_usuario)
        // Limpiar los campos después de un inicio de sesión exitoso
        setUsername('')
        setPassword('')
        //Redireccionar a la página de inicio
        history.replace('/home')
      } else {
        // Inicio de sesión fallido, mostrar modal de alerta
        setAlertMessage('Usuario o contraseña incorrecto');
        setShowAlert(true)
      }
    } catch (error: any) {
```

```
      // Manejar errores de la petición
      console.error('Error inicio de sesión:', error.message);
      setAlertMessage('Usuario o contraseña incorrectos');
      setShowAlert(true)
    }
    finally {
      setShowLoading(false) // Ocultar el componente IonLoading al
finalizar la petición
    }
  }
}
```

Anexo C. Código cambio de contraseña

```
const CambiarContrasena: React.FC = () => {
  const history = useHistory();
  let [showLoading, setShowLoading] = useState(false);
  let [showLoadingBus, setShowLoading2] = useState(false);
  let [correoElectronico, setCorreo] = useState('');
  let [Asesor, setCodogoAsesor] = useState('');
  let [codigo, setCodigo] = useState('');
  let [password, setPassword] = useState('');
  let [passwordCheck, setPasswordCheck] = useState('');
  let [showPassword, setShowPassword] = useState(false);
  let [showPasswordVali, setShowPasswordVali] = useState(false);
  let [showAlert, setShowAlert] = useState(false);
  let [showAlertPass, setShowAlertPass] = useState(false);
  let [showAlertCre, setShowAlertCre] = useState(false);
  let [alertMessage, setAlertMessage] = useState('');
  let [showDelivert, setShowDelivery] = useState(true);
  let [showCode, setShowCode] = useState(false);
  let [changePassword, setChangePassword] = useState(false);
  let [errorCorreo, setErrorCorreo] = useState('');
  let [habilitado, setHabilitado] = useState(true);
  let [contador, setContador] = useState(1);
  let url = 'https://siav.rectima.com.ec:3001/'
  let [passwordValida, setPasswordValida] = useState(false);
  let [codigoGenerado, setCodigoGenerado] = useState('');
  let formatoCorreo = ''

  let mostrarCorreoFormato = (correo: string) => {
    // Dividir el correo en el nombre de usuario y el dominio
    let [nombreUsuario, dominio] = correo.split('@');

    // Obtener las iniciales del nombre de usuario
    let iniciales = nombreUsuario.charAt(0).toUpperCase() + '.....';

    // Formar el correo en el formato deseado
    let correoFormateado = `${iniciales}@${dominio}`;

    return correoFormateado;
  };

  // Función para generar un código aleatorio de 6 dígitos
  let generarCodigoAleatorio = () => {
```

```
    let codigo = '';
    const digitos = '1234567890';

    // Mezclar los dígitos para evitar repeticiones
    let digitosAleatorios = digitos.split('').sort(() =>
Math.random() - 0.5);

    // Tomar los primeros 6 dígitos
    for (let i = 0; i < 6; i++) {
      codigo += digitosAleatorios[i];
    }

    return codigo;
  }

  let toggleDelivery = async (option: number) => {

    if (option == 0) {
      if (!Asesor) {
        setAlertMessage('Por favor ingrese un código empresarial')
        setShowAlert(true)
        return
      }
      try {
        // Realizar la petición al servidor
        let response = await axios.post(url, {
          Asesor
        })
        // Verificar la respuesta del servidor
        if (response.data.success) {
            let correoValue = response.data.message.CORREO;
            correoElectronico = correoValue
            //setCorreo(correoValue)
            let codigoNuevo = generarCodigoAleatorio()
            codigoGenerado = codigoNuevo
            setCodigoGenerado(codigoNuevo)
          try {
            let response = await axios.post(url, {
              Asesor,codigoGenerado
            })
            if (response.status === 200) {
              setAlertMessage('Mensaje enviado')
               setShowAlert(true)
              formatoCorreo =
mostrarCorreoFormato(correoElectronico)
              setShowDelivery(false)
              setShowCode(true)
            }else{
              setAlertMessage('Se produjo un error al enviar el
correo vuelva a intentarlo de nuevo más tarde')
               setShowAlert(true)
            }
          } catch (error: any) {
            console.log('Error mensaje: '+error)
            setAlertMessage('Se produjo un error al enviar el correo
vuelva a intentarlo de nuevo más tarde')
```

```
            setShowAlert(true)
          }
        }else{
          //alert('Porfavor ingrese un correo valido')
          setAlertMessage('Por favor ingrese un código valido')
          setShowAlert(true)
        }
      } catch (error: any) {
        console.log('Error: '+error)
        setAlertMessage('Se produjo un error al enviar el correo
vuelva a intentarlo de nuevo más tarde')
        setShowAlert(true)
      }
    }else if(option == 1){
      setShowDelivery(true)
      setShowCode(false)
      setHabilitado(!habilitado)
    }
  }

  let toggleChangePassword = () => {
    if (codigo == codigoGenerado) {
        setShowCode(false)
        setChangePassword(true)
        setCodigoGenerado('')
    }else{
       setContador(contador + 1);
      if (contador < 3) {
        setAlertMessage('Código no valido por favor verifique el
código ingresado')
         setShowAlert(true)
      }else{
        setAlertMessage('Código no valido reinténtelo de nuevo')
        setShowAlert(true)
        setCodigoGenerado('')
        setContador(1)
        setHabilitado(!habilitado)
      }

    }
  }

  let handleCambiarClave = async () => {
    if (!password || !passwordCheck) {
      setAlertMessage('Por favor, Ingrese una contraseña')
      setShowAlert(true)
      return
    }

    if (password != passwordCheck) {
      setAlertMessage('Por favor revise las contraseñas ingresadas
no son iguales')
      setShowAlert(true)
    }else{
      try {
        let Password = md5(password)
```

```
        let response = await axios.put(url, {
          Asesor,Password
        })
        if (response.status === 200) {
          alert('Se cambio la contraseña correctamente !!')
          history.push('/login')
        }else{
          setAlertMessage('Se ha producido un error no se ha podido
realizar el cambiar de la contraseña inténtelo más tarde')
          setShowAlert(true)
          history.push('/login')
        }
      } catch (error: any) {
        setAlertMessage('Se ha producido un error al cambiar la
contraseña intentelo de nuevo más tarde')
        setShowAlert(true)
        history.push('/login')
      }
    }
  }

  let validarCorreo = (codAsesor: string) => {
    const patronCorreo = /^[0-9]+$/;
    if (patronCorreo.test(codAsesor)) {
      setErrorCorreo('');
    } else {
      setErrorCorreo('Por favor ingrese un código valido');
    }
  };

  let handleChangeCorreo = (e: React.ChangeEvent<HTMLInputElement>)
=> {
    let nuevoAsesor = e.target.value;
    setCodogoAsesor(nuevoAsesor);
    if (nuevoAsesor.trim() === '') {
      setErrorCorreo('');
    }else{
      validarCorreo(nuevoAsesor);
    }
  };

  let validarPassword = (value: any) => {
    // Expresión regular para validar la contraseña
    const regex = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-
Za-z\d@$!%*?&]{5,}$/;
    return regex.test(value);
  };

  let handleChange = (e: any) => {
    let valor = e.target.value;
    setPassword(valor);
    setPasswordValida(validarPassword(valor));
  };

  let toggleShowPassword = () => {
    setShowPassword(!showPassword);
```

```
  }

  let toggleShowPasswordVali = () => {
    setShowPasswordVali(!showPasswordVali);
  }
}
```

Anexo D. Código inicio

```
const Home: React.FC = () => {

  const history = useHistory()
  let { username, tipo_usuario, nombre } = useUser()
  let [currentSlide, setCurrentSlide] = useState(0)
  let [fadeClass, setFadeClass] = useState('fade-in')
  let [alertMessage, setAlertMessage] = useState('')
  let [showAlertSesion, setShowAlertSesion] = useState(false)
  let fechaActual = new Date()
  let formatoFecha = 'yyyy/MM/dd'
  let url = 'https://siav.rectima.com.ec:3001/'
  let fechaFormateada = format(fechaActual, formatoFecha)
  let slides =
['https://siav.rectima.com.ec:81/imageapp/promos/promo1.png',

'https://siav.rectima.com.ec:81/imageapp/promos/promo2.png',

'https://siav.rectima.com.ec:81/imageapp/promos/promo3.png']
  let [datosUsuario, setDatosUsuario] = useState('');

  useEffect(() => {
    let interval = setInterval(() => {
      setFadeClass('fade-out'); // Iniciar la transición de salida
      setTimeout(() => {
        setCurrentSlide((prevSlide) => (prevSlide + 1) %
slides.length);
        setFadeClass('fade-in'); // Iniciar la transición de entrada
después de un breve retraso
      }, 1000);
    }, 5000);
    if (tipo_usuario != 'Administrador') {
      cargarPresupuesto();
    }
    return () => {
      // Limpiar el intervalo al salir del componente o cambiar de
vista
      clearInterval(interval)
    }
  }, [slides.length]);

  const handleLogout = () => {
    // Lógica para cerrar sesión
    // Eliminar el token del localStorage
    localStorage.removeItem('token')
    const location = { pathname: '/login' };
    history.replace(location)
```

```
    history.push(location)
    window.location.reload()

  }

  let cerrarSesionInactivo = async () => {
    setShowAlertSesion(false)
    localStorage.removeItem('token')
    const location = { pathname: '/login' };
    history.replace(location)
    history.push(location)
    window.location.reload()
  }

  let cargarPresupuesto = async () => {
    //Comprobar si el usuario tiene un valor
    if(username == null || username == ''){
    // Mensaje de advertencia
      setAlertMessage('Su sesión ha expirado. Por favor, vuelva a
iniciar sesión')
      setShowAlertSesion(true)
    }

    try {
      // Realizar la petición al servidor
      let response = await axios.post(url, {
        fechaFormateada,
        username,
      });
      // Verificar la respuesta del servidor
      if (response.data.success) {
        let TOTAL_IDEAL = response.data.user[0].TOTAL_IDEAL
        setDatosUsuario(TOTAL_IDEAL)
      } else {
        // Inicio de sesión fallido
        console.log('Error de inicio de sesión')
      }
    } catch (error: any) {
      // Manejar errores de la petición
      console.error('Se produjo un error al obtener el presupuesto
intentelo de nuevo más tarde', error.message);
    }
  }
}
```

Anexo E. Código progreso actual

```
const DatosDia: React.FC = () => {
    const history = useHistory()
    let { username, tipo_usuario } = useUser()
    let fechaActual = new Date()
    let [showAlert, setShowAlert] = useState(false)
    let [alertMessage, setAlertMessage] = useState('')
    let [showLoading, setShowLoading] = useState(false) //cambio
```

```
    let [vendedorSeleccionado, setVendedorSeleccionado] =
useState('')
    let [showAlertSesion, setShowAlertSesion] = useState(false)
    let formatoFecha = 'yyyy/MM/dd'
    let [mes, setMesSeleccionado] = useState('')
    let [anio, setAnioSeleccionado] = useState('')
    let url = 'https://siav.rectima.com.ec:3001/'
    let [datosVendedor, setDatosVendedor] = useState<Vendedor[]>([])
    let fechaFormateada = format(fechaActual, formatoFecha)
    let rutaImagen =
`https://siav.rectima.com.ec:81/imageapp/vendedores/${username}.png`
    let [datosUsuario, setDatosUsuario] = useState({
        MES: '',anio: '',CUMPLIMIENTO_HOY: '',CUMPLIMIENTO_OBJ: '',
CUMPLIMIENTO_VTA: '',
        PRESUPUESTO: '',VENTA_HOY: '',CLIE_ASIGN: '',CLIE_ATEND:
'',EFECTI_OBJ: '',EFECTI_VTA: '',
        RENTA_OBJ: '',RENTA_VTA: '',BONO1_OBJ: '',BONO1_VTA:
'',BONO2_OBJ: '',BONO2_VTA: '',CLTN_OBJ: '',CLTN_VTA: ''
    })

    useEffect(() => {
        // Esta función se ejecutará al cargar la página
        if (tipo_usuario != 'Administrador') {
            handleCargaDatos()
        }else{
            handleListaVendedor()
        }
    }, []);

    const handleLogout = () => {
        // Lógica para cerrar sesión
        // Eliminar el token del localStorage
        localStorage.removeItem('token')
        const location = { pathname: '/login' }
        history.replace(location)
        history.push(location)
        window.location.reload()
    }
    let handleSeleccionMes = (event: CustomEvent) => {
        setMesSeleccionado(event.detail.value)
    }

    let handleSeleccionAnio = (event: CustomEvent) => {
        setAnioSeleccionado(event.detail.value)
    }

    let handleSeleccionVendedor = (event: CustomEvent) => {
        setVendedorSeleccionado(event.detail.value)
    }

    let handleRefresh = (event: CustomEvent<RefresherEventDetail>)
=> {
        setTimeout(() => {
            // Llamada a cargar datos
            handleCargaDatos()
            event.detail.complete()
```

```
        })
    }

    let cerrarSesionInactivo = async () => {
        setShowAlertSesion(false)
        localStorage.removeItem('token')
        const location = { pathname: '/login' };
        history.replace(location)
        history.push(location)
        window.location.reload()
      }

    const handleCargaDatos = async () => {
        //Comprobar si el usuario tiene un valor
        if(username == null || username == ''){
            // Mensaje de advertencia
              setAlertMessage('Su sesión ha expirado. Por favor,
vuelva a iniciar sesión')
              setShowAlertSesion(true)
            }

        try {
            setShowLoading(true)
            if (tipo_usuario === 'Administrador') {
                fechaFormateada = anio+'/'+mes+'/01'
                username = vendedorSeleccionado
            }
            console.log('Dos veces !')
            // Realizar la petición al servidor
            let response = await axios.post(url, {
                fechaFormateada,
                username,
            });
            // Verificar la respuesta del servidor
            if (response.data.success) {
                // Inicio de sesión exitoso, redirigir a otra página
                let { MES, anio, CUMPLIMIENTO_HOY, CUMPLIMIENTO_VTA,
CUMPLIMIENTO_OBJ, PRESUPUESTO, VENTA_HOY, CLIE_ASIGN, CLIE_ATEND,
EFECTI_OBJ, EFECTI_VTA,
                    RENTA_OBJ, RENTA_VTA, BONO1_OBJ, BONO1_VTA,
BONO2_OBJ, BONO2_VTA, CLTN_OBJ, CLTN_VTA } = response.data.user;
                setDatosUsuario({

MES,anio,CUMPLIMIENTO_HOY,CUMPLIMIENTO_OBJ,CUMPLIMIENTO_VTA,PRESUPUE
STO,VENTA_HOY,

CLIE_ASIGN,CLIE_ATEND,EFECTI_OBJ,EFECTI_VTA,RENTA_OBJ,RENTA_VTA,BONO
1_OBJ,BONO1_VTA,BONO2_OBJ,
                    BONO2_VTA,CLTN_OBJ,CLTN_VTA
                })
                // Ocultar el componente IonLoading
                setShowLoading(false);
            } else {
                // Inicio de sesión fallido, mostrar modal de alerta
                setAlertMessage('Error al obtener los datos
requeridos');
```

```
                setShowAlert(true);
            }
        } catch (error: any) {
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message);
            setAlertMessage('Se produjo un error al obtener los
datos requeridos intentelo de nuevo más tarde');
            setShowAlert(true);
        } finally {
            // Asegurar que el componente IonLoading se oculte
incluso en caso de error
            setShowLoading(false);
        }
    }

    let handleListaVendedor = async () => {
        try {
            // Realizar la petición al servidor
            let response = await axios.get(url);
            // Verificar la respuesta del servidor
            if (response.data.success) {
                // Cargar datos
               setDatosVendedor(response.data.user)
            } else {
                // Inicio de sesión fallido, mostrar modal de alerta
                setAlertMessage('Error al obtener los datos
requeridos');
                setShowAlert(true);
            }
        } catch (error: any) {
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message);
            setAlertMessage('Se produjo un error al obtener datos
del listado intentelo de nuevo más tarde');
            setShowAlert(true);
        }
    }
}
```

Anexo F. Código comisión por monto

```
const Monto: React.FC = () => {
    interface Vendedor {
        USUARIO: string;
        NOMBRE: string;
    }
    const history = useHistory()
    let { username, tipo_usuario } = useUser()
    let fechaActual = new Date()
    let [showAlert, setShowAlert] = useState(false)
    let [alertMessage, setAlertMessage] = useState('')
    let [showLoading, setShowLoading] = useState(false)
    let formatoFecha = 'yyyy/MM/dd'
    let url = 'https://siav.rectima.com.ec:3001/'
```

```
    let [vendedorSeleccionado, setVendedorSeleccionado] =
useState('')
    let [showAlertSesion, setShowAlertSesion] = useState(false)
    let [mes, setMesSeleccionado] = useState('')
    let [anio, setAnioSeleccionado] = useState('')
    let [datosVendedor, setDatosVendedor] = useState<Vendedor[]>([])
    let fechaFormateada = format(fechaActual, formatoFecha)
    let rutaImagen =
`https://siav.rectima.com.ec:81/imageapp/vendedores/${username}.png`
    let [datosUsuario, setDatosUsuario] = useState({
        MES: '',ANIO: '',VENTA_NETA: '',PRESUPUESTO:
'',CUMPLIMIENTO: '',COMISION: '',PORCE_COMISION: ''
    })

    useEffect(() => {
        if (tipo_usuario != 'Administrador') {
            handleCargaDatos()
        }else{
            handleListaVendedor()
        }
    }, [])

    let handleSeleccionMes = (event: CustomEvent) => {
        setMesSeleccionado(event.detail.value)
    }

    let handleSeleccionAnio = (event: CustomEvent) => {
        setAnioSeleccionado(event.detail.value)
    }

    let handleSeleccionVendedor = (event: CustomEvent) => {
        setVendedorSeleccionado(event.detail.value)
    }
    let handleLogout = () => {
        // Lógica para cerrar sesión
        // Eliminar el token del localStorage
        localStorage.removeItem('token')
        const location = { pathname: '/login' }
        history.replace(location)
        history.push(location)
        window.location.reload()
    }

    let handleRefresh = (event: CustomEvent<RefresherEventDetail>)
=> {
        setTimeout(() => {
            // Llamada a cargar datos
            handleCargaDatos()
            event.detail.complete()
        })
    }

    let cerrarSesionInactivo = async () => {
        setShowAlertSesion(false)
        localStorage.removeItem('token')
        const location = { pathname: '/login' };
```

```
        history.replace(location)
        history.push(location)
        window.location.reload()
      }

    let handleCargaDatos = async () => {
        //Comprobar si el usuario tiene un valor
        if(username == null || username == ''){
        // Mensaje de advertencia
          setAlertMessage('Su sesión ha expirado. Por favor, vuelva
a iniciar sesión')
          setShowAlertSesion(true)
        }

        try {
            setShowLoading(true)
            if (tipo_usuario === 'Administrador') {
                fechaFormateada = anio+'/'+mes+'/01'
                username = vendedorSeleccionado
            }
            // Realizar la petición al servidor
            let response = await axios.post(url, {
                fechaFormateada,
                username,
            });
            // Verificar la respuesta del servidor
            if (response.data.success) {
                // Inicio de sesión exitoso, redirigir a otra página
                let { MES, ANIO, VENTA_NETA, PRESUPUESTO,
CUMPLIMIENTO, COMISION,
                    PORCE_COMISION} = response.data.user;
                setDatosUsuario({

MES,ANIO,VENTA_NETA,PRESUPUESTO,CUMPLIMIENTO,COMISION,PORCE_COMISION
                })
            } else {
                // Inicio de sesión fallido, mostrar modal de alerta
                setAlertMessage('Error al obtener los datos
requeridos')
                setShowAlert(true)
            }
        } catch (error: any) {
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message)
            setAlertMessage('Se produjo un error al obtener los
datos requeridos intentelo de nuevo más tarde')
            setShowAlert(true)
        } finally {
            setShowLoading(false)
        }
    }

    let handleListaVendedor = async () => {
        try {
            // Realizar la petición al servidor
            let response = await axios.get(url);
```

```
            // Verificar la respuesta del servidor
            if (response.data.success) {
                // Cargar datos
               setDatosVendedor(response.data.user)
            } else {
                // Inicio de sesión fallido, mostrar modal de alerta
                setAlertMessage('Error al obtener los datos
requeridos');
                setShowAlert(true);
            }
        } catch (error: any) {
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message);
            setAlertMessage('Se produjo un error al obtener datos
del listado intentelo de nuevo más tarde');
            setShowAlert(true);
        }
    }
}
```

Anexo G. Código comisión por efectividad

```
const Efectividad: React.FC = () => {
    interface Vendedor {
        USUARIO: string;
        NOMBRE: string;
    }
    let history = useHistory()
    let { username, tipo_usuario } = useUser()
    let fechaActual = new Date()
    let [showAlert, setShowAlert] = useState(false)
    let [alertMessage, setAlertMessage] = useState('')
    let [showLoading, setShowLoading] = useState(false)
    let formatoFecha = 'yyyy/MM/dd'
    let url = 'https://siav.rectima.com.ec:3001/'
    let [vendedorSeleccionado, setVendedorSeleccionado] =
useState('')
    let [showAlertSesion, setShowAlertSesion] = useState(false)
    let [mes, setMesSeleccionado] = useState('')
    let [anio, setAnioSeleccionado] = useState('')
    let [datosVendedor, setDatosVendedor] = useState<Vendedor[]>([])
    let fechaFormateada = format(fechaActual, formatoFecha)
    let rutaImagen =
`https://siav.rectima.com.ec:81/imageapp/vendedores/${username}.png`
    let [datosUsuario, setDatosUsuario] = useState({
        MES: '',ANIO: '',TOTAL_CLT: '',CLT_ATENDIDO:
'',CUMPLIMIE_ATENCION: '',CLI_NEW: '',
        VTA_CLI_NEW: '',CUMPLIMIENTO_NEW: '',CLI_NEW_REC:
'',VTA_CLI_NEW_REC: '',CUMPLIMIENTO_REC: '',
        COMISION: '',PORCE_COMISION: ''
    })

    useEffect(() => {
        if (tipo_usuario != 'Administrador') {
            handleCargaDatos()
```

```
        }else{
            handleListaVendedor()
        }
    }, [])

    let handleSeleccionMes = (event: CustomEvent) => {
        setMesSeleccionado(event.detail.value)
    }

    let handleSeleccionAnio = (event: CustomEvent) => {
        setAnioSeleccionado(event.detail.value)
    }

    let handleSeleccionVendedor = (event: CustomEvent) => {
        setVendedorSeleccionado(event.detail.value)
    }

    let handleLogout = () => {
        // Lógica para cerrar sesión
        // Eliminar el token del localStorage
        localStorage.removeItem('token')
        const location = { pathname: '/login' };
        history.replace(location)
        history.push(location)
        window.location.reload()
    };

    let handleRefresh = (event: CustomEvent<RefresherEventDetail>)
=> {
        setTimeout(() => {
            // Llamada a cargar datos
            handleCargaDatos()
            event.detail.complete()
        })
    }

    let cerrarSesionInactivo = async () => {
        setShowAlertSesion(false)
        localStorage.removeItem('token')
        const location = { pathname: '/login' };
        history.replace(location)
        history.push(location)
        window.location.reload()
      }

    let handleCargaDatos = async () => {
            //Comprobar si el usuario tiene un valor
            if(username == null || username == ''){
                // Mensaje de advertencia
                setAlertMessage('Su sesión ha expirado. Por favor,
vuelva a iniciar sesión')
                setShowAlertSesion(true)
            }

        try {
            setShowLoading(true)
```

```
            if (tipo_usuario === 'Administrador') {
                fechaFormateada = anio+'/'+mes+'/01'
                username = vendedorSeleccionado
            }
            // Realizar la petición al servidor
            let response = await axios.post(url, {
                fechaFormateada,
                username,
            });
            // Verificar la respuesta del servidor
            if (response.data.success) {
                // Inicio de sesión exitoso, redirigir a otra página
                let { MES, ANIO, TOTAL_CLT, CLT_ATENDIDO,
CUMPLIMIE_ATENCION, CLI_NEW, VTA_CLI_NEW,
                    CUMPLIMIENTO_NEW, CLI_NEW_REC, VTA_CLI_NEW_REC,
CUMPLIMIENTO_REC, COMISION, PORCE_COMISION } = response.data.user;
                setDatosUsuario({

MES,ANIO,TOTAL_CLT,CLT_ATENDIDO,CUMPLIMIE_ATENCION,CLI_NEW,VTA_CLI_N
EW,

CUMPLIMIENTO_NEW,CLI_NEW_REC,VTA_CLI_NEW_REC,CUMPLIMIENTO_REC,COMISI
ON,PORCE_COMISION
                })
            } else {
                // Inicio de sesión fallido, mostrar modal de alerta
                setAlertMessage('Error al obtener los datos
requeridos')
                setShowAlert(true)
            }
        } catch (error: any) {
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message)
            setAlertMessage('Se produjo un error al obtener los
datos requeridos intentelo de nuevo más tarde')
            setShowAlert(true)
        } finally {
            setShowLoading(false)
        }
    }

    let handleListaVendedor = async () => {
        try {
            // Realizar la petición al servidor
            let response = await axios.get(url);
            // Verificar la respuesta del servidor
            if (response.data.success) {
                // Cargar datos
               setDatosVendedor(response.data.user)
            } else {
                // Inicio de sesión fallido, mostrar modal de alerta
                setAlertMessage('Error al obtener los datos
requeridos');
                setShowAlert(true);
            }
        } catch (error: any) {
```

```
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message);
            setAlertMessage('Se produjo un error al obtener datos
del listado intentelo de nuevo más tarde');
            setShowAlert(true);
        }
    }
}
```

Anexo H. Código comisión por rentabilidad

```
const Rentabilidad: React.FC = () => {
    interface Vendedor {
        USUARIO: string;
        NOMBRE: string;
    }
    const history = useHistory()
    let { username, tipo_usuario } = useUser()
    let fechaActual = new Date()
    let [showAlert, setShowAlert] = useState(false)
    let [alertMessage, setAlertMessage] = useState('')
    let [showLoading, setShowLoading] = useState(false)
    let formatoFecha = 'yyyy/MM/dd'
    let url = 'https://siav.rectima.com.ec:3001/'
    let fechaFormateada = format(fechaActual, formatoFecha)
    let [vendedorSeleccionado, setVendedorSeleccionado] =
useState('')
    let [showAlertSesion, setShowAlertSesion] = useState(false)
    let [mes, setMesSeleccionado] = useState('')
    let [anio, setAnioSeleccionado] = useState('')
    let [datosVendedor, setDatosVendedor] = useState<Vendedor[]>([])
    let rutaImagen =
`https://siav.rectima.com.ec:81/imageapp/vendedores/${username}.png`
    let [datosUsuario, setDatosUsuario] = useState({
        MES: '',ANIO: '',PORCE_ALCANZADO: '',PORCE_COMISION:
'',COMISION: '',RENTA_OBJETIVO: ''
    })

    useEffect(() => {
        if (tipo_usuario != 'Administrador') {
            handleCargaDatos()
        }else{
            handleListaVendedor()
        }
    }, [])

    let handleSeleccionMes = (event: CustomEvent) => {
        setMesSeleccionado(event.detail.value)
    }

    let handleSeleccionAnio = (event: CustomEvent) => {
        setAnioSeleccionado(event.detail.value)
    }

    let handleSeleccionVendedor = (event: CustomEvent) => {
```

```
        setVendedorSeleccionado(event.detail.value)
    }

    let handleLogout = () => {
        // Lógica para cerrar sesión
        // Eliminar el token del localStorage
        localStorage.removeItem('token')
        const location = { pathname: '/login' };
        history.replace(location)
        history.push(location)
        window.location.reload()
    }

    let handleRefresh = (event: CustomEvent<RefresherEventDetail>)
=> {
        setTimeout(() => {
            // Llamada a cargar datos
            handleCargaDatos()
            event.detail.complete()
        })
    }

    let cerrarSesionInactivo = async () => {
        setShowAlertSesion(false)
        localStorage.removeItem('token')
        const location = { pathname: '/login' };
        history.replace(location)
        history.push(location)
        window.location.reload()
      }

    let handleCargaDatos = async () => {
        //Comprobar si el usuario tiene un valor
        if(username == null || username == ''){
            // Mensaje de advertencia
            setAlertMessage('Su sesión ha expirado. Por favor,
vuelva a iniciar sesión')
            setShowAlertSesion(true)
        }

        try {
            setShowLoading(true)
            if (tipo_usuario === 'Administrador') {
                fechaFormateada = anio+'/'+mes+'/01'
                username = vendedorSeleccionado
            }
            // Realizar la petición al servidor
            let response = await axios.post(url, {
                fechaFormateada,
                username,
            })
            // Verificar la respuesta del servidor
            if (response.data.success) {
                // Inicio de sesión exitoso, redirigir a otra página
                let { MES, ANIO, PORCE_ALCANZADO, PORCE_COMISION,
COMISION, RENTA_OBJETIVO} = response.data.user
```

```
                setDatosUsuario({

MES,ANIO,PORCE_ALCANZADO,PORCE_COMISION,COMISION,RENTA_OBJETIVO
                })
            } else {
                // Inicio de sesión fallido, mostrar modal de alerta
                setAlertMessage('Error al obtener los datos
requeridos')
                setShowAlert(true)
            }
        } catch (error: any) {
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message)
            setAlertMessage('Se produjo un error al obtener los
datos requeridos intentelo de nuevo más tarde')
            setShowAlert(true)
        } finally {
            setShowLoading(false)
        }
    }

    let handleListaVendedor = async () => {
        try {
            // Realizar la petición al servidor
            let response = await axios.get(url);
            // Verificar la respuesta del servidor
            if (response.data.success) {
                // Cargar datos
               setDatosVendedor(response.data.user)
            } else {
                // Inicio de sesión fallido, mostrar modal de alerta
                setAlertMessage('Error al obtener los datos
requeridos');
                setShowAlert(true);
            }
        } catch (error: any) {
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message);
            setAlertMessage('Se produjo un error al obtener datos
del listado intentelo de nuevo mas tarde');
            setShowAlert(true);
        }
    }
}
```

Anexo I. Código bono por monto

```
const BonoMonto: React.FC = () => {
    interface Vendedor {
        USUARIO: string;
        NOMBRE: string;
    }
    const history = useHistory()
    let { username,tipo_usuario } = useUser()
    let fechaActual = new Date()
```

```
    let [showAlert, setShowAlert] = useState(false)
    let [alertMessage, setAlertMessage] = useState('')
    let [showLoading, setShowLoading] = useState(false)
    let formatoFecha = 'yyyy/MM/dd'
    let url = 'https://siav.rectima.com.ec:3001/'
    let fechaFormateada = format(fechaActual, formatoFecha)
    let [vendedorSeleccionado, setVendedorSeleccionado] =
useState('')
    let [showAlertSesion, setShowAlertSesion] = useState(false)
    let [mes, setMesSeleccionado] = useState('')
    let [anio, setAnioSeleccionado] = useState('')
    let [datosVendedor, setDatosVendedor] = useState<Vendedor[]>([])
    let [totalExcedente, setExcedente] = useState(0)
    let rutaImagen =
`https://siav.rectima.com.ec:81/imageapp/vendedores/${username}.png`
    let [datosUsuario, setDatosUsuario] = useState({
        MES: '',ANIO: '',VENTA_NETA: '',PRESUPUESTO:
'',CUMPLIMIENTO: '',BONO: '',COMISION: ''
    })

    useEffect(() => {
        if (tipo_usuario != 'Administrador') {
            handleCargaDatos()
        }else{
            handleListaVendedor()
        }
    }, [])

    let handleSeleccionMes = (event: CustomEvent) => {
        setMesSeleccionado(event.detail.value)
    }

    let handleSeleccionAnio = (event: CustomEvent) => {
        setAnioSeleccionado(event.detail.value)
    }

    let handleSeleccionVendedor = (event: CustomEvent) => {
        setVendedorSeleccionado(event.detail.value)
    }

    let handleLogout = () => {
        // Lógica para cerrar sesión
        // Eliminar el token del localStorage
        localStorage.removeItem('token')
        const location = { pathname: '/login' };
        history.replace(location)
        history.push(location)
        window.location.reload()
    }

    let handleRefresh = (event: CustomEvent<RefresherEventDetail>)
=> {
        setTimeout(() => {
            // Llamada a cargar datos
            handleCargaDatos()
            event.detail.complete()
```

```
        })
    }

    let cerrarSesionInactivo = async () => {
        setShowAlertSesion(false)
        localStorage.removeItem('token')
        const location = { pathname: '/login' };
        history.replace(location)
        history.push(location)
        window.location.reload()
    }

    let handleCargaDatos = async () => {

        //Comprobar si el usuario tiene un valor
        if(username == null || username == ''){
            // Mensaje de advertencia
            setAlertMessage('Su sesión ha expirado. Por favor,
vuelva a iniciar sesión')
            setShowAlertSesion(true)
        }

        try {
            setShowLoading(true)
            if (tipo_usuario === 'Administrador') {
                fechaFormateada = anio+'/'+mes+'/01'
                username = vendedorSeleccionado
            }
            // Realizar la petición al servidor
            let response = await axios.post(url, {
                fechaFormateada,
                username,
            })
            // Verificar la respuesta del servidor
            if (response.data.success) {
                // Inicio de sesión exitoso, redirigir a otra página
                let { MES, ANIO, VENTA_NETA, PRESUPUESTO,
CUMPLIMIENTO, BONO, COMISION
                } = response.data.user
                setDatosUsuario({

MES,ANIO,VENTA_NETA,PRESUPUESTO,CUMPLIMIENTO,BONO,COMISION
                })
                let valExcedente
                let ventaNetaNumerica =
parseFloat(VENTA_NETA.replace(',', ''));
                let presupuestoNumerico =
parseFloat(PRESUPUESTO.replace(',', ''));
                if (VENTA_NETA > PRESUPUESTO) {
                    valExcedente = ventaNetaNumerica -
presupuestoNumerico
                    valExcedente =
parseFloat(valExcedente.toFixed(2))
                }else{
                    valExcedente = 0
                }
```

```
                    setExcedente(valExcedente)
                } else {
                    // Inicio de sesión fallido, mostrar modal de alerta
                    setAlertMessage('Error al obtener los datos
requeridos')
                    setShowAlert(true)
                }
            } catch (error: any) {
                // Manejar errores de la petición
                console.error('Error en la petición:', error.message)
                setAlertMessage('Se produjo un error al obtener los
datos requeridos intentelo de nuevo más tarde')
                setShowAlert(true)
            } finally {
                setShowLoading(false)
            }
        }

        let handleListaVendedor = async () => {
            try {
                // Realizar la petición al servidor
                let response = await axios.get(url);
                // Verificar la respuesta del servidor
                if (response.data.success) {
                    // Cargar datos
                   setDatosVendedor(response.data.user)
                } else {
                    // Inicio de sesión fallido, mostrar modal de alerta
                    setAlertMessage('Error al obtener los datos
Requeridos');
                    setShowAlert(true);
                }
            } catch (error: any) {
                // Manejar errores de la petición
                console.error('Error en la petición:', error.message);
                setAlertMessage('Se produjo un error al obtener datos
del listado intentelo de nuevo más tarde');
                setShowAlert(true);
            }
        }
}
```

Anexo J. Código bono por promocional

```
const Promocional: React.FC = () => {
    interface Vendedor {
        USUARIO: string;
        NOMBRE: string;
    }
    const history = useHistory()
    let { username, tipo_usuario } = useUser()
    let fechaActual = new Date()
    let [showAlert, setShowAlert] = useState(false)
    let [alertMessage, setAlertMessage] = useState('')
    let [showLoading, setShowLoading] = useState(false)
```

```
    let formatoFecha = 'yyyy/MM/dd'
    let url = 'https://siav.rectima.com.ec:3001/'
    let fechaFormateada = format(fechaActual, formatoFecha)
    let [vendedorSeleccionado, setVendedorSeleccionado] =
useState('')
    let [showAlertSesion, setShowAlertSesion] = useState(false)
    let [mes, setMesSeleccionado] = useState('')
    let [anio, setAnioSeleccionado] = useState('')
    let [datosVendedor, setDatosVendedor] = useState<Vendedor[]>([])
    let rutaImagen =
`https://siav.rectima.com.ec:81/imageapp/vendedores/${username}.png`
    let [datosUsuario, setDatosUsuario] = useState({
        MES: '',ANIO: '',PRESU_PROMO: '',VENTA_PROMO:
'',CUMPLIMIENTO: '',BONO_PROMO: '',COMISION: ''
    })

    useEffect(() => {
        if (tipo_usuario != 'Administrador') {
            handleCargaDatos()
        }else{
            handleListaVendedor()
        }
    }, [])

    let handleSeleccionMes = (event: CustomEvent) => {
        setMesSeleccionado(event.detail.value)
    }

    let handleSeleccionAnio = (event: CustomEvent) => {
        setAnioSeleccionado(event.detail.value)
    }

    let handleSeleccionVendedor = (event: CustomEvent) => {
        setVendedorSeleccionado(event.detail.value)
    }

    let handleLogout = () => {
        // Lógica para cerrar sesión
        // Eliminar el token del localStorage
        localStorage.removeItem('token')
        const location = { pathname: '/login' };
        history.replace(location)
        history.push(location)
        window.location.reload()
    }

    let handleRefresh = (event: CustomEvent<RefresherEventDetail>)
=> {
        setTimeout(() => {
            // Llamada a cargar datos
            handleCargaDatos()
            event.detail.complete()
        })
    }

    let cerrarSesionInactivo = async () => {
```

```
        setShowAlertSesion(false)
        localStorage.removeItem('token')
        const location = { pathname: '/login' };
        history.replace(location)
        history.push(location)
        window.location.reload()
      }

    let handleCargaDatos = async () => {

        //Comprobar si el usuario tiene un valor
        if(username == null || username == ''){
            // Mensaje de advertencia
            setAlertMessage('Su sesión ha expirado. Por favor,
vuelva a iniciar sesión')
            setShowAlertSesion(true)
        }
        try {
            setShowLoading(true)
            if (tipo_usuario === 'Administrador') {
                fechaFormateada = anio+'/'+mes+'/01'
                username = vendedorSeleccionado
            }
            // Realizar la petición al servidor
            let response = await axios.post(url, {
                fechaFormateada,
                username,
            });
            // Verificar la respuesta del servidor
            if (response.data.success) {
                // Inicio de sesión exitoso, redirigir a otra página
                let { MES, ANIO, PRESU_PROMO, VENTA_PROMO,
CUMPLIMIENTO, BONO_PROMO, COMISION
                } = response.data.user
                setDatosUsuario({
                    MES,ANIO,PRESU_PROMO, VENTA_PROMO, CUMPLIMIENTO,
BONO_PROMO, COMISION
                })
            } else {
                // Inicio de sesión fallido, mostrar modal de alerta
                setAlertMessage('Error al obtener los datos
requeridos')
                setShowAlert(true)
            }
        } catch (error: any) {
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message)
            setAlertMessage('Se produjo un error al obtener los
datos requeridos intentelo de nuevo más tarde')
            setShowAlert(true)
        } finally {
            setShowLoading(false)
        }
    }

    let handleListaVendedor = async () => {
```

```
        try {
            // Realizar la petición al servidor
            let response = await axios.get(url);
            // Verificar la respuesta del servidor
            if (response.data.success) {
                // Cargar datos
               setDatosVendedor(response.data.user)
            } else {
                // Inicio de sesión fallido, mostrar modal de alerta
                setAlertMessage('Error al obtener los datos
requeridos');
                setShowAlert(true);
            }
        } catch (error: any) {
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message);
            setAlertMessage('Se produjo un error al obtener datos
del listado intentelo de nuevo más tarde');
            setShowAlert(true);
        }
    }
}
```

Anexo K. Código bono por cartera

```
const Cartera: React.FC = () => {
    interface Vendedor {
        USUARIO: string;
        NOMBRE: string;
    }
    const history = useHistory()
    let { username, tipo_usuario } = useUser()
    let fechaActual = new Date()
    let [showAlert, setShowAlert] = useState(false)
    let [alertMessage, setAlertMessage] = useState('')
    let [showLoading, setShowLoading] = useState(false)
    let formatoFecha = 'yyyy/MM/dd'
    let fechaFormateada = format(fechaActual, formatoFecha)
    let url = 'https://siav.rectima.com.ec:3001/'
    let [vendedorSeleccionado, setVendedorSeleccionado] =
useState('')
    let [showAlertSesion, setShowAlertSesion] = useState(false)
    let [mes, setMesSeleccionado] = useState('')
    let [anio, setAnioSeleccionado] = useState('')
    let [datosVendedor, setDatosVendedor] = useState<Vendedor[]>([])
    let rutaImagen =
`https://siav.rectima.com.ec:81/imageapp/vendedores/${username}.png`
    let [datosUsuario, setDatosUsuario] = useState({
        MES: '',ANIO: '',PRESUP_COBRAR: '',COBROS:
'',CUMPLIMIENTO_EFECT: '',TOTAL_VTA: '',
        CARTERA: '',CUMPLIMIENTO_CARTERA: '',TOTAL_VTA_EFEC:
'',CARTERA_EFEC: '',CUMPLIMIENTO_CARTERA_EFEC: '',
        COMISION: ''
    })
```

```
    useEffect(() => {
        if (tipo_usuario != 'Administrador') {
            handleCargaDatos()
        }else{
            handleListaVendedor()
        }
    }, [])

    let handleSeleccionMes = (event: CustomEvent) => {
        setMesSeleccionado(event.detail.value)
    }

    let handleSeleccionAnio = (event: CustomEvent) => {
        setAnioSeleccionado(event.detail.value)
    }

    let handleSeleccionVendedor = (event: CustomEvent) => {
        setVendedorSeleccionado(event.detail.value)
    }

    let handleLogout = () => {
        // Lógica para cerrar sesión
        // Eliminar el token del localStorage
        localStorage.removeItem('token')
        const location = { pathname: '/login' };
        history.replace(location)
        history.push(location)
        window.location.reload()
    }
    let handleRefresh = (event: CustomEvent<RefresherEventDetail>)
=> {
        setTimeout(() => {
            // Llamada a cargar datos
            handleCargaDatos()
            event.detail.complete()
        })
    }

    let cerrarSesionInactivo = async () => {
        setShowAlertSesion(false)
        localStorage.removeItem('token')
        const location = { pathname: '/login' };
        history.replace(location)
        history.push(location)
        window.location.reload()
    }

    let handleCargaDatos = async () => {

        //Comprobar si el usuario tiene un valor
        if(username == null || username == ''){
            // Mensaje de advertencia
            setAlertMessage('Su sesión ha expirado. Por favor,
vuelva a iniciar sesión')
            setShowAlertSesion(true)
        }
```

```
        try {
            setShowLoading(true)
            if (tipo_usuario === 'Administrador') {
                fechaFormateada = anio+'/'+mes+'/01'
                username = vendedorSeleccionado
            }
            // Realizar la petición al servidor
            let response = await axios.post(url, {
                fechaFormateada,
                username,
            })
            // Verificar la respuesta del servidor
            if (response.data.success) {
                // Inicio de sesión exitoso, redirigir a otra página
                let { MES, ANIO, PRESUP_COBRAR, COBROS,
CUMPLIMIENTO_EFECT, TOTAL_VTA, CARTERA, CUMPLIMIENTO_CARTERA,
                    TOTAL_VTA_EFEC, CARTERA_EFEC,
CUMPLIMIENTO_CARTERA_EFEC, COMISION} = response.data.user
                setDatosUsuario({

MES,ANIO,PRESUP_COBRAR,COBROS,CUMPLIMIENTO_EFECT,TOTAL_VTA,CARTERA,

CUMPLIMIENTO_CARTERA,TOTAL_VTA_EFEC,CARTERA_EFEC,CUMPLIMIENTO_CARTER
A_EFEC,COMISION
                })
            } else {
                // Inicio de sesión fallido, mostrar modal de alerta
                setAlertMessage('Error al obtener los datos
requeridos')
                setShowAlert(true)
            }
        } catch (error: any) {
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message)
            setAlertMessage('Se produjo un error al obtener los
datos requeridos intentelo de nuevo más tarde')
            setShowAlert(true)
        } finally {
            setShowLoading(false)
        }
    }
}
```

Anexo L. Código total progreso

```
const TotalComosionB: React.FC = () => {
    interface Vendedor {
        USUARIO: string;
        NOMBRE: string;
    }
    const history = useHistory()
    let { username, tipo_usuario } = useUser()
    let url = 'https://siav.rectima.com.ec:3001/'
    let [showAlert, setShowAlert] = useState(false)
    let [alertMessage, setAlertMessage] = useState('')
```

```
    let [showLoading, setShowLoading] = useState(false)
    let [showLoadingPdf, setShowLoadingPdf] = useState(false)
    let [showAlertCre, setShowAlertCre] = useState(false)
    let [mes, setMesSeleccionado] = useState('')
    let [anio, setAnioSeleccionado] = useState('')
    let [htmlContent, setHtmlContent] = useState('')
    //let formatoFecha = 'yyyy/MM/dd'
    let [mostrarGenerarPdf, setMostrarGenerarPdf] = useState(false)
    let [vendedorSeleccionado, setVendedorSeleccionado] =
useState('')
    let [showAlertSesion, setShowAlertSesion] = useState(false)
    let [datosVendedor, setDatosVendedor] = useState<Vendedor[]>([])
    //let fechaFormateada = format(fechaActual, formatoFecha)
    let rutaImagen =
`https://siav.rectima.com.ec:81/imageapp/vendedores/${username}.png`
    //let imagenReporte = `src/assets/logo_rectima.png`
    let [datosUsuario, setDatosUsuario] = useState({
        COMI_MONTO: '', COMI_RENTA: '', COMI_EFECTIV: '',
BONO_MONTO: '', BONO_PROMO: '',
        BONO_CARTERA: '', TOTAL_COMI:'', CUMPLI_MONTO: '',
PORCE_RENTABILIDA: '', CUMPLIMIENTO_ALCA: '',
        CUMPLIMI_BONO1: '', CUMPLIMI_BONO2: '', CUMPLIMI_CARTE: ''
    })

    useEffect(() => {
        if (tipo_usuario === 'Administrador') {
            handleListaVendedor()
        }
    }, []);

    let handleLogout = () => {
        // Lógica para cerrar sesión
        // Eliminar el token del localStorage
        localStorage.removeItem('token')
        const location = { pathname: '/login' }
        history.replace(location)
        history.push(location)
        window.location.reload()
    }
    let handleSeleccionMes = (event: CustomEvent) => {
        setMesSeleccionado(event.detail.value)
    }

    let handleSeleccionAnio = (event: CustomEvent) => {
        setAnioSeleccionado(event.detail.value)
    }

    let handleSeleccionVendedor = (event: CustomEvent) => {
        setVendedorSeleccionado(event.detail.value)
    }

    let handleRefresh = (event: CustomEvent<RefresherEventDetail>)
=> {
        setTimeout(() => {
            // Llamada a cargar datos
            handleCargaDatos()
```

```
            event.detail.complete()
        })
    }

    let handleGenerarReporte = async () => {
        let htmlGenerar = ''
        let asesor

        if (mes == '' || anio == '') {
            setAlertMessage('Por favor, seleccione un mes y año
antes de generar el pdf')
            setShowAlertCre(true)
            return
        }
        if(tipo_usuario === 'Administrador'){
            asesor = vendedorSeleccionado
        }else{
            asesor = username
        }
        setShowLoadingPdf(true)
        try {
            let response = await axios.post(url, {
                mes, anio, asesor
                })
                if (response.data.success) {
                    let totalRol =
parseFloat(response.data.user[0].comision)+parseFloat(response.data.
user[0].comision_cliente)+parseFloat(response.data.user[0].comi_cli_
nue)+

parseFloat(response.data.user[0].comi_cli_rec)+parseFloat(response.d
ata.user[0].comi_rentabilidad)+parseFloat(response.data.user[0].bono
1_comi)+

parseFloat(response.data.user[0].porce_comi_promo)+parseFloat(respon
se.data.user[0].COMI_COBRA_EFEC)+parseFloat(response.data.user[0].co
mi_cartera)+
                    parseFloat(response.data.user[0].COMI_PLAZO)

                    setTimeout(() => {setHtmlContent(htmlGenerar);
                      });
setTimeout(async () => { // Esperar a que se genere el HTML
                        let pdf = new jsPDF('p', 'mm', 'a4');
                         // Convertir las imágenes externas en
base64
                        pdf.html(htmlGenerar, {
                                callback: () => {
                                    let fechaActual = new
Date().toISOString().split('T')[0];

//pdf.save('reporte_'+username+'_app_'+fechaActual+'.pdf');
                                    if (tipo_usuario ===
'Administrador') {
                                        username =
vendedorSeleccionado
                                    }
```

```
                                        let fileName =
'reporte_'+username+'_app_'+fechaActual+'.pdf';
                                        let pdfData =
pdf.output('blob');

File.writeFile(File.externalRootDirectory + '/Download', fileName,
pdfData, { replace: true })
                                        .then(() => {
                                            return FileOpener.open(

File.externalRootDirectory + "/Download/" + fileName,
                                                "application/pdf"
                                            );
                                      })
                                        .catch(error => {
                                        alert('Se produjo un error al
guardar el archivo intentelo de nuevo mas tarde '+error.message);
                                        });
                                        setMostrarGenerarPdf(false)
                                      },
                                      x:17,
                                      y:4,
                                      html2canvas: {scale:0.27}
                                    }
                                    );
                            });
                            setShowLoadingPdf(false);
                    } else {
                    setShowLoadingPdf(false);
                    // Manejar errores de la petición
                    setAlertMessage('Se produjo un error al obtener los
datos')
                    setShowAlert(true)
                    }
            } catch (error: any) {
                setShowLoadingPdf(false);
                // Manejar errores de la petición
                console.error('Error interno:', error.message)
                setAlertMessage('Se produjo un error al generar el
archivo pdf intentelo de nuevo más tarde')
                setShowAlert(true)
            }
        }

        let cerrarSesionInactivo = async () => {
            setShowAlertSesion(false)
            localStorage.removeItem('token')
            const location = { pathname: '/login' };
            history.replace(location)
            history.push(location)
            window.location.reload()
        }

        let handleCargaDatos = async () => {
            let fechaActual = new Date();
            let anioActual = fechaActual.getFullYear();
```

```
          //Comprobar si el usuario tiene un valor
        if(username == null || username == ''){
            // Mensaje de advertencia
            setAlertMessage('Su sesión ha expirado. Por favor,
vuelva a iniciar sesión')
            setShowAlertSesion(true)
        }

        try {
            if (tipo_usuario === 'Administrador') {

                if(anio != anioActual.toString()){
                    setAlertMessage('Por favor, seleccione una fecha
del año actual')
                    setShowAlertCre(true)
                    return
                }
                if (mes == '' || anio == '' || vendedorSeleccionado
== '') {
                    setAlertMessage('Por favor, seleccione todos los
campos requeridos')
                    setShowAlertCre(true)
                    return
                }else{
                    username = vendedorSeleccionado
                }
            }

            if(mes == '' || anio == ''){
                setAlertMessage('Por favor, seleccione un mes y
año')
                setShowAlertCre(true)
                return
            }


            if(anio != anioActual.toString()){
                setAlertMessage('Por favor, seleccione una fecha del
año actual')
                setShowAlertCre(true)
                return
            }
            // Realizar la petición al servidor
            setShowLoading(true)

            let fechaFormateada = anio+'/'+mes+'/01'
            let opcion = 1
            let response = await axios.post(url, {
                fechaFormateada,
                username,
                opcion
            })
            // Verificar la respuesta del servidor
            if (response.data.success) {
```

```
                let { COMI_MONTO, COMI_RENTA, COMI_EFECTIV,
BONO_MONTO, BONO_PROMO, BONO_CARTERA,
                    TOTAL_COMI, CUMPLI_MONTO, PORCE_RENTABILIDA,
CUMPLIMIENTO_ALCA, CUMPLIMI_BONO1,
                    CUMPLIMI_BONO2, CUMPLIMI_CARTE} =
response.data.user
                setDatosUsuario({

COMI_MONTO,COMI_RENTA,COMI_EFECTIV,BONO_MONTO,BONO_PROMO,BONO_CARTER
A,
                    TOTAL_COMI, CUMPLI_MONTO, PORCE_RENTABILIDA,
CUMPLIMIENTO_ALCA, CUMPLIMI_BONO1,
                    CUMPLIMI_BONO2, CUMPLIMI_CARTE
                })
                if (tipo_usuario != 'Supervisor') {
                    setMostrarGenerarPdf(true)
                }
            } else {
                setShowLoading(false)
                // Consulta fallida, mostrar modal de alerta
                setAlertMessage('Error al obtener los datos
requeridos')
                setShowAlert(true)
            }
        } catch (error: any) {
            setShowLoading(false)
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message)
            setAlertMessage('Se produjo un error al obtener los
datos requeridos intentelo de nuevo más tarde')
            setShowAlert(true)
        } finally {
            setShowLoading(false)
        }
    }

    let handleListaVendedor = async () => {
        try {
            // Realizar la petición al servidor
            let response = await axios.get(url);
            // Verificar la respuesta del servidor
            if (response.data.success) {
                // Cargar datos
               setDatosVendedor(response.data.user)
            } else {
                // Inicio de sesión fallido, mostrar modal de alerta
                setAlertMessage('Error al obtener los datos
requeridos');
                setShowAlert(true);
            }
        } catch (error: any) {
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message);
            setAlertMessage('Se produjo un error al obtener datos
del listado intentelo de nuevo más tarde');
            setShowAlert(true);
```

```
        }
    }
}
```

Anexo M. Código creación nuevo cliente

```
const CrearCliente: React.FC = () => {
  let { takePhoto } = usePhotoGallery();
    interface Provincia {
      CODIGO_PROV: string;
      NOMBRE_PROV: string;
  }
  const history = useHistory();
  let [showLoadingCliente, setShowLoadingCliente] = useState(false);
  let { username, nombre } = useUser()
  let [cedula, setCedula] = useState('');
  let [extranjero, setExtranjero] = useState(0);
  let [comercial, setComercial] = useState('');
  let [razon, setRazon] = useState('');
  let [direccion, setDireccion] = useState('');
  let [telefono, setTelefono] = useState('');
  let [ciudad, setCiudad] = useState('');
  let [parroquia, setParroquia] = useState('');
  let [cupo, setCupo] = useState('');
  let [correo, setCorreo] = useState('');
  let [contacto, setContacto] = useState('');
  let url = 'https://siav.rectima.com.ec:3001/'
  let [datosProvincia, setDatosProvincia] =
useState<Provincia[]>([])
  let [proviSeleccion, setProvinSeleccion] = useState('')
  let [showCampos, setShowCampos] = useState(true);
  let [showCupo, setShowCupo] = useState(false);
  let [plazoPago, setPlazoPago] = useState('Seleccionar');
  let [campoImagenes, setShowImagenes] = useState(false);
  let [colorBoton, setColorBoton] = useState('secondary');

  let [photoCedula, setPhotoCedula] = useState('');
  let [photoPapeleta, setPhotoPapeleta] = useState('');
  let [photoRuc, setPhotoRuc] = useState('');
  let [photoSerBas, setPhotoSerBas] = useState('');
  let [photoSRI, setPhotoSRI] = useState('');
  let [photoCerBanco, setPhotoCerBanco] = useState('');
  let [photoRefPerso, setPhotoRefPerso] = useState('');
  let [photoRefComer, setPhotoRefComer] = useState('');
  let [photoLetra, setPhotoLetra] = useState('');
  let [photoDeclaracion, setPhotoDeclaracion] = useState('');
  let [showAlertSesion, setShowAlertSesion] = useState(false)

  let [showAlert, setShowAlert] = useState(false);
  let [showAlertPass, setShowAlertPass] = useState(false);
  let [showAlertCre, setShowAlertCre] = useState(false);
  let [alertMessage, setAlertMessage] = useState('');
  let [errorCedula, setErrorCedula] = useState('');
  let [errorTelefono, setErrorTelefono] = useState('');
  let [errorCorreo, setErrorCorreo] = useState('');
```

```
  let [errorCupo, setErrorCupo] = useState('');
  useEffect(() => {
    //cargar provincia
    handleListaProvincias()
}, []);

  let handleLogout = () => {
    // Lógica para cerrar sesión
    username = ''
    nombre = ''
     // Eliminar el token del localStorage
    localStorage.removeItem('token')
    const location = { pathname: '/login' };
    history.replace(location)
    history.push(location)
    window.location.reload()
}


//Validar cedula/ruc
  let handleChangeCedula = (e: any) => {
    let value = e.target.value;
    let cedulaRegex = /^[0-9]{10,13}$/;
    // Verificar si la longitud es menor que 10 o si no son todos
números
    if (cedulaRegex.test(value) || value === '') {
      setErrorCedula('');
    } else {
      setErrorCedula('La cédula/ruc debe ser numérico tener mínimo
10 y maximo 13 caracteres');
    }
    setCedula(value);
  };

  //Validar telefono
  let handleChangeTelefono = (e: any) => {
    let value = e.target.value;
    let telefonoRegex = /^0[0-9]{8}$/;
    // Verificar si la longitud es menor que 10 o si no son todos
números
    if (telefonoRegex.test(value) || value === '') {
      setErrorTelefono('');
    } else {
      setErrorTelefono('El número de teléfono debe empezar con 0,
tener mínimo 10 caracteres y ser numérico');
    }
    setTelefono(value);
  };

  //Validar correo
  let handleChangeCorreo = (e: any) => {
    let value = e.target.value;
    let emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
    // Verificar si la longitud es menor que 10 o si no son todos
números
    if (emailRegex.test(value) || value === '') {
```

```
      setErrorCorreo('');
    } else {
      setErrorCorreo('Ingrese una dirección de correo electrónico
válida');
    }
    setCorreo(value);
  };

  //Validar correo
  let handleChangeCupo = (e: any) => {
    let value = e.target.value;
    let cupoRegex = /^[0-9]*$/;
    // Verificar si la longitud es menor que 10 o si no son todos
números
    if (cupoRegex.test(value) || value === '') {
      setErrorCupo('');
    } else {
      setErrorCupo('Ingrese un valor numérico para el cupo');
    }
    setCupo(value);
  };

   // Función para cambiar la visibilidad de todos los campos
   let toggleCampos = () => {
    if (cedula == '' || comercial == '' || razon == '' || direccion
== '' || telefono == '' || ciudad == '' || parroquia == '' || correo
== '' || contacto == '' || proviSeleccion == '') {
      setAlertMessage('Por favor, Ingrese todos los datos requeridos
del nuevo cliente');
      setShowAlert(true)
    }else{
      setShowCampos(!showCampos)
      setShowCupo(!showCupo)

      // Cambiar el color del botón según el estado de mostrarCampos
      if (!showCampos) {
        setColorBoton('secondary');
      } else {
        setColorBoton('success');
      }
    }

  }

  // Función para manejar el cambio de la selección
  let handlePlazoPago = (e: CustomEvent) => {
    let selectedValue = e.detail.value;
    setPlazoPago(selectedValue)
    //Mostra el campo adicional
    setShowImagenes(selectedValue === 'CONTADO' || selectedValue ===
'8 DIAS' || selectedValue === '15 DIAS')
  }
  //Funcion para cambiar valor campo extranjero 0 | 1
  const handleCheckboxChange = (e:
React.ChangeEvent<HTMLInputElement>) => {
    const isChecked = e.target.checked;
```

```
    const value = isChecked ? 1 : 0;
    setExtranjero(value);
  };

  //Función para tomar la foto y almacenarla en cada boton
  let handleTakePhoto = async (valores: number) => {
        // Llama a la función takePhoto del hook
        let photoPath = await takePhoto();
    try {
      if (photoPath) {
        switch (valores) {
          case 1:
            setPhotoCedula(photoPath)
            break;
          case 2:
            setPhotoPapeleta(photoPath)
            break;
          case 3:
            setPhotoRuc(photoPath)
            break;
          case 4:
           setPhotoSerBas(photoPath)
            break;
          case 5:
           setPhotoSRI(photoPath)
            break;
          case 6:
            setPhotoCerBanco(photoPath)
            break;
          case 7:
            setPhotoRefPerso(photoPath)
            break;
          case 8:
              setPhotoRefComer(photoPath)
            break;
          case 9:
            setPhotoLetra(photoPath)
            break;
          case 10:
            setPhotoDeclaracion(photoPath)
            break;
          default:
            break;
        }
      }

    } catch (error) {
      console.error('Error al tomar la foto:', error);
    }
  };
 //Función para eliminar la foto de los campos
  let elimiarFoto = (valor: number) =>{
    switch (valor) {
      case 1:
        setPhotoCedula('')
        break;
```

```
        case 2:
          setPhotoPapeleta('')
          break;
        case 3:
          setPhotoRuc('')
          break;
        case 4:
         setPhotoSerBas('')
          break;
        case 5:
         setPhotoSRI('')
          break;
        case 6:
          setPhotoCerBanco('')
          break;
        case 7:
          setPhotoRefPerso('')
          break;
        case 8:
            setPhotoRefComer('')
          break;
        case 9:
          setPhotoLetra('')
          break;
        case 10:
          setPhotoDeclaracion('')
          break;
        default:
          break;
      }
    }
    // Funciones para obtener valores individuales
    let handleSeleccionProvincia = (event: CustomEvent) => {
      setProvinSeleccion(event.detail.value)

    }

    let cerrarSesionInactivo = async () => {
      setShowAlertSesion(false)
      localStorage.removeItem('token')
      const location = { pathname: '/login' };
      history.replace(location)
      history.push(location)
      window.location.reload()
    }

    //Para obtener el listado de las provincias
    let handleListaProvincias = async () => {
       //Comprobar si el usuario tiene un valor
       if(username == null || username == ''){
        // Mensaje de advertencia
          setAlertMessage('Su sesión ha expirado. Por favor, vuelva a
iniciar sesión')
          setShowAlertSesion(true)
        }
```

```
    try {
        // Realizar la petición al servidor
        let response = await axios.get(url);
        // Verificar la respuesta del servidor
        if (response.data.success) {
            // Cargar datos
           setDatosProvincia(response.data.user)
        } else {
            // Inicio de sesión fallido, mostrar modal de alerta
            setAlertMessage('Error al obtener los datos de las
Provicias');
            setShowAlert(true);
        }
    } catch (error: any) {
        // Manejar errores de la petición
        console.error('Error en la petición:', error.message);
        setAlertMessage('Se produjo un error al obtener los datos
del selector');
        setShowAlert(true);
    }
}
//Accion para limpiar los campos despues de la creación del cliente
const handleAceptarClick = () => {
   //Limpiar los campos despues de insertarlso en la base de datos
   setCedula('')
   setExtranjero(0);
   setComercial('');
   setRazon('');
   setDireccion('');
   setTelefono('');
   setCiudad('');
   setParroquia('');
   setCupo('');
   setCorreo('');
   setContacto('');
   setProvinSeleccion('')
   setShowCampos(true);
   setShowCupo(false);
   setPlazoPago('Seleccionar');
   setColorBoton('secondary');
   setPhotoCedula('');
   setPhotoPapeleta('');
   setPhotoRuc('');
   setPhotoSerBas('');
   setPhotoSRI('');
   setPhotoCerBanco('');
   setPhotoRefPerso('');
   setPhotoRefComer('');
   setPhotoLetra('');
   setPhotoDeclaracion('');

};

//Funcion para registrar al nuevo cliente
let registrarCliente = async () => {
  if (plazoPago == 'CONTADO') {
```

```
    if (!photoCedula || !photoRuc) {
      setAlertMessage('Por favor, Ingrese las imágenes requeridas');
      setShowAlert(true);
      return;
    }
    setShowLoadingCliente(true);
   try{
    let response = await axios.post(url, {
      username, cedula, extranjero, comercial, razon, direccion,
telefono, ciudad, parroquia, correo, contacto, proviSeleccion, cupo,
      plazoPago, photoCedula, photoPapeleta, photoRuc, photoSerBas,
photoSRI, photoCerBanco, photoRefPerso, photoRefComer, photoLetra,
      photoDeclaracion
    });
    setShowLoadingCliente(false);
    if (response.data.success) {
      let respuesta = response.data.user[0].SALIDA;
      if (respuesta == 'OK') {
        setAlertMessage(`Cliente registrado correctamente \n\n
Nombre: ${razon}`)
        setShowAlertCre(true)
      }else{
        setAlertMessage(`${respuesta}`);
        setShowAlertCre(true)
      }

    }else{
      setShowLoadingCliente(false);
      setAlertMessage('Error al registrar el nuevo cliente');
      setShowAlertPass(true);
    }
   }catch(error: any){
    // Manejar errores de la petición
    setShowLoadingCliente(false);
    setAlertMessage('Se produjo un error al registrar el nuevo
cliente intentelo de nuevo más tarde '+error.message);
    setShowAlertPass(true);
   }

  }else{
    if (!photoCedula || !photoPapeleta || !photoRuc || !photoSerBas
|| !photoRefComer || !photoLetra || !photoDeclaracion) {
      setAlertMessage('Por favor, Ingrese todas las imagenes
requeridas');
      setShowAlert(true);
      return;
    }
    setShowLoadingCliente(true);
    try{
     let response = await axios.post(url, {
       username, cedula, extranjero, comercial, razon, direccion,
telefono, ciudad, parroquia, correo, contacto, proviSeleccion, cupo,
       plazoPago, photoCedula, photoPapeleta, photoRuc, photoSerBas,
photoSRI, photoCerBanco, photoRefPerso, photoRefComer, photoLetra,
       photoDeclaracion
     });
```

```
      setShowLoadingCliente(false);
      if (response.data.success) {
        let respuesta = response.data.user[0].SALIDA;
        if (respuesta == 'OK') {
          setAlertMessage(`Cliente Registrado Correctamente \n\n
Nombre: ${razon}`)
          setShowAlertCre(true)
        }else{
          setAlertMessage(`${respuesta}`);
          setShowAlertCre(true)
        }

      }else{
        setShowLoadingCliente(false);
        setAlertMessage('Error al registrar el nuevo cliente');
        setShowAlertPass(true);
      }
     }catch(error: any){
      // Manejar errores de la petición
      setShowLoadingCliente(false);
      setAlertMessage('Se produjo un error al registrar el nuevo
cliente inténtelo de nuevo más tarde '+error.message);
      setShowAlertPass(true);
     }
   }
}
}
```

Anexo N. Código ventas clientes

```
const ReporteCliente: React.FC = () => {
    interface Vendedor {
        USUARIO: string;
        NOMBRE: string;
    }
    interface UserData {
        RUC: string;
        CLIENTE: string;
        VENTAS: number;
        DEVOLUCIONES: number;
    }
    interface UserDataProductos {
        PRODUCTO: string;
        DESCRIPCION: string;
        VENTAS: number;
        DEVOLUCIONES: number;
    }

    let history = useHistory()
    let { username, nombre, tipo_usuario } = useUser()
    let [showAlert, setShowAlert] = useState(false)
    let [showLoadingProductos, setShowLoadingProductos] =
useState(false)
    let [alertMessage, setAlertMessage] = useState('')
    let [showLoading, setShowLoading] = useState(false)
```

```
    let [vendedorSeleccionado, setVendedorSeleccionado] =
useState('')
    let [showAlertSesion, setShowAlertSesion] = useState(false)
    let [datosVendedor, setDatosVendedor] = useState<Vendedor[]>([])
    let  rutaImagen = ''
    let url = 'https://siav.rectima.com.ec:3001/'
    if (tipo_usuario != 'Supervisor') {
        rutaImagen =
`https://siav.rectima.com.ec:81/imageapp/vendedores/${username}.png`
    }else{
        rutaImagen =
`https://siav.rectima.com.ec:81/imageapp/vendedores/200.png`
    }

    let [data, setData] = useState<UserData[]>([]);
    let [dataPro, setDataPro] = useState<UserDataProductos[]>([]);
    let [currentPage, setCurrentPage] = useState(1);
    let [currentPagePro, setCurrentPro] = useState(1);
    let itemsPerPage = 10; // Número de elementos por página
    let itemsPerPagePro = 5;
    let [searchTerm, setSearchTerm] = useState('');
    let [showModal, setShowModal] = useState(false);
    let [clienteProducto, setClienteProducto] = useState('')
    let [filteredData, setFilteredData] = useState(data);

    useEffect(() => {
        if (tipo_usuario == 'Asesor') {
            cargarClientesAsignados()
        }else{
            handleListaVendedor()
        }
    }, [])

    useEffect(() => {
        filtrarDatos(searchTerm);
    }, [searchTerm, data])

    let handleLogout = () => {
        // Lógica para cerrar sesión
        // Eliminar el token del localStorage
        localStorage.removeItem('token')
        const location = { pathname: '/login' };
        history.replace(location)
        history.push(location)
        window.location.reload()
    };

    let handleSeleccionVendedor = (event: CustomEvent) => {
        setVendedorSeleccionado(event.detail.value)
    }

    let handleRefresh = (event: CustomEvent<RefresherEventDetail>)
=> {
        setTimeout(() => {
            // Llamada a cargar datos
            cargarClientesAsignados()
```

```
            event.detail.complete()
        })
    }

    let cerrarSesionInactivo = async () => {
        setShowAlertSesion(false)
        localStorage.removeItem('token')
        const location = { pathname: '/login' };
        history.replace(location)
        history.push(location)
        window.location.reload()
      }

    let cargarClientesAsignados = async () => {
            //Comprobar si el usuario tiene un valor
            if(username == null || username == ''){
                // Mensaje de advertencia
                setAlertMessage('Su sesión ha expirado. Por favor,
vuelva a iniciar sesión')
                setShowAlertSesion(true)
            }

        try {
            setShowLoading(true)
            if (tipo_usuario != 'Asesor') {
                username = vendedorSeleccionado
            }
            let asesor = username
            // Realizar la petición al servidor
            let response = await axios.post(url, {
                asesor
            });

            // Verificar la respuesta del servidor
            if (response.data.success) {
                //Almacenar datos obtenidos
                setData(response.data.user);

            } else {
                // Inicio de sesión fallido, mostrar modal de alerta
                setAlertMessage('Se produjo un error al obtener
clientes asignados, intentelo de nuevo')
                setShowAlert(true)
            }
        } catch (error: any) {
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message)
            setAlertMessage('Se produjo un error al obtener los
clientes intentelo de nuevo más tarde')
            setShowAlert(true)
        } finally {
            setShowLoading(false)
        }
    }

    let handleListaVendedor = async () => {
```

```
        try {
            // Realizar la petición al servidor
            let response = await axios.get(url);
            // Verificar la respuesta del servidor
            if (response.data.success) {
                // Cargar datos
               setDatosVendedor(response.data.user)
            } else {
                // Inicio de sesión fallido, mostrar modal de alerta
                setAlertMessage('Error al obtener los datos
requeridos');
                setShowAlert(true);
            }
        } catch (error: any) {
            // Manejar errores de la petición
            console.error('Error en la petición:', error.message);
            setAlertMessage('Se produjo un error al obtener datos
del listado intentelo de nuevo más tarde');
            setShowAlert(true);
        }
    }

    let filteredDataPro = dataPro.filter(itemPro =>

itemPro.DESCRIPCION.toLowerCase().includes(searchTerm.toLowerCase())
    );

    let handleInputChange = (e: any) => {
        let value = e.target.value
        setSearchTerm(value)
        searchTerm = value
        filtrarDatos(value)
    }

    let filtrarDatos = (searchTerm: any) => {
        let filtered = data.filter(item =>

item.CLIENTE.toLowerCase().includes(searchTerm.toLowerCase())
        );
        setFilteredData(filtered)
    }

    // Calcular el índice del primer y último elemento clientes
    let lastIndex = currentPage * itemsPerPage
    let firstIndex = lastIndex - itemsPerPage
    let maxButtonsToShow = 5
    let halfMaxButtonsToShow = Math.floor(maxButtonsToShow / 2)
    // Calcular el índice del primer y último elemento productos
    let lastIndexPro = currentPagePro * itemsPerPagePro
    let firstIndexPro = lastIndexPro - itemsPerPagePro

    // Para calcular el valor del paginador clientes
    let startPage = Math.max(1, currentPage - halfMaxButtonsToShow);
    let endPage = Math.min(Math.ceil(filteredData.length /
itemsPerPage), startPage + maxButtonsToShow - 1);
    // Para calcular el valor del paginador clientes productos
```

```
    let startPagePro = Math.max(1, currentPage -
halfMaxButtonsToShow);
    let endPagePro = Math.min(Math.ceil(filteredData.length /
itemsPerPagePro), startPagePro + maxButtonsToShow - 1);
    // Para asignar los vaores clientes
    let currentData = filteredData.slice(firstIndex, lastIndex)
    // Para asignar los vaores clientes
    let currentDataPro = filteredDataPro.slice(firstIndexPro,
lastIndexPro)
    //Para paginar por cada 5 items
    let renderPaginationButtons = () => {
        let buttons = []

        for (let i = startPage; i <= endPage; i++) {
            buttons.push(
                <IonButton color={currentPage === i ? 'primary' :
'secondary'} key={i} onClick={() => handlePageChange(i)}
className={currentPage === i ? 'active' : ''}>
                    {i}
                </IonButton>
            );
        }

        return buttons;
    };

    // Función para cambiar de página
    let handlePageChange = (pageNumber: number) => {
        setCurrentPage(pageNumber);
    };
    let handlePageChangePro = (pageNumberPro: number) => {
        setCurrentPro(pageNumberPro);
    };
    let chartRef = useRef<HTMLCanvasElement>(null);
    //Funcion para buscar productos por ruc/cedula cliente
    let buscarProductosCliente = async (ruc: any, cliente: any) => {

        setShowLoadingProductos(true)
       try {
        let response = await axios.post(url, {
            ruc
        });
        // Verificar la respuesta del servidor
        if (response.data.success) {
            //Almacenar datos obtenidos
            dataPro = []
            setDataPro(response.data.user)
            setClienteProducto(cliente)
            setShowLoadingProductos(false)
            setShowModal(true)
        } else {
            // Inicio de sesión fallido, mostrar modal de alerta
            setShowLoadingProductos(false)
            setAlertMessage('Se produjo un error al obtener los
productos, intentelo de nuevo')
            setShowAlert(true)
```

```
            }
        } catch (error: any) {
            //Manejo de errores
            setShowLoadingProductos(false)
            console.error('Error en la petición:', error.message)
            setAlertMessage('Se produjo un error al obtener los
productos intentelo de nuevo más tarde')
            setShowAlert(true)
        }
    }

    //CerrarModal
    let cerrarModal = () => {
        setShowModal(false)
    }
}
```

Anexo O. Manual de usuario

- **Objetivo**

El presente manual está diseñado para ayudar a los usuarios a comprender todas las características y funcionalidades del aplicativo móvil, así como a proporcionar instrucciones paso a paso sobre cómo realizar diferentes tareas dentro del aplicativo móvil.

- **Funcionalidad**

El aplicativo móvil esta desarrollada y enfocada en la revisión en tiempo real del progreso en comisiones y bonos que los asesores comerciales puedan alcanzar durante el trascurso de cada mes, así como un apartado de ventas donde los asesores comerciales pueden crear nuevos clientes y revisar reportes de valores en ventas a clientes, así como la cantidad de productos adquiridos por cada cliente.

La funcionalidad se enfoca en:

Una pantalla de inicio de sesión.

Una pantalla de creación nuevo usuario.

Una pantalla para el cambio de contraseña.

Una pantalla de inicio con las promocionales e información.

Una pantalla con la descripción del avance en bonos y comisiones actual.

Una pantalla con la descripción del avance en comisiones por monto.

Una pantalla con la descripción del avance en comisiones por efectividad.

Una pantalla con la descripción del avance en comisiones por rentabilidad.

Una pantalla con la descripción del avance en bonos por monto.

Una pantalla con la descripción del avance en bonos por promocional.

Una pantalla con la descripción del avance en bonos por cartera.

Una pantalla con el resumen del total de bonos y comisiones alcanzados actual.

Una pantalla para el ingreso y creación de nuevos clientes.

Una pantalla para la revisión del valor en ventas por clientes y la cantidad de productos adquiridos por cada cliente.

- **Navegación y descripción del aplicativo móvil**

Pantalla inicio de sesión

Al ingresar al aplicativo móvil lo primero que se muestra es el apartado de inicio de sesión, donde el usuario debe ingresar su usuario y contraseña en caso de estar registrado en el aplicativo móvil.

Pantalla de registro

En caso de no contar con una cuenta el usuario puede realizar el registro ingresando al apartado **Crear Ahora** presente en la pantalla de inicio de sesión, en esta pantalla se mostrará los campos requeridos para el registro.

Pantalla cambio de contraseña

En este apartado se permite el cambio de contraseña en caso de olvido o perdida, para el cambio de contraseña es necesario el ingreso del código empresarial personal y contar con un correo electrónico empresarial asignado para validar la identidad del usuario que requiere el cambio de contraseña.

Ingreso código personal empresarial

Validación identidad por código enviado a correo personal del usuario

Ingreso de una nueva contraseña

Pantalla de inicio

Se presenta una bienvenida, los datos de comisión y alcance de cada usuario, frases motivacionales y las promocionales disponibles.

Menú de navegación

En el menú de navegación se muestran todos los campos disponibles dentro del aplicativo móvil, el menú está compuesto por Inicio, Progreso Hoy, Comisiones por monto, efectividad y rentabilidad, Bonos por monto, promocional y cartera, Total Progreso, Nuevo Cliente y Ventas Clientes.

Pantalla progreso hoy

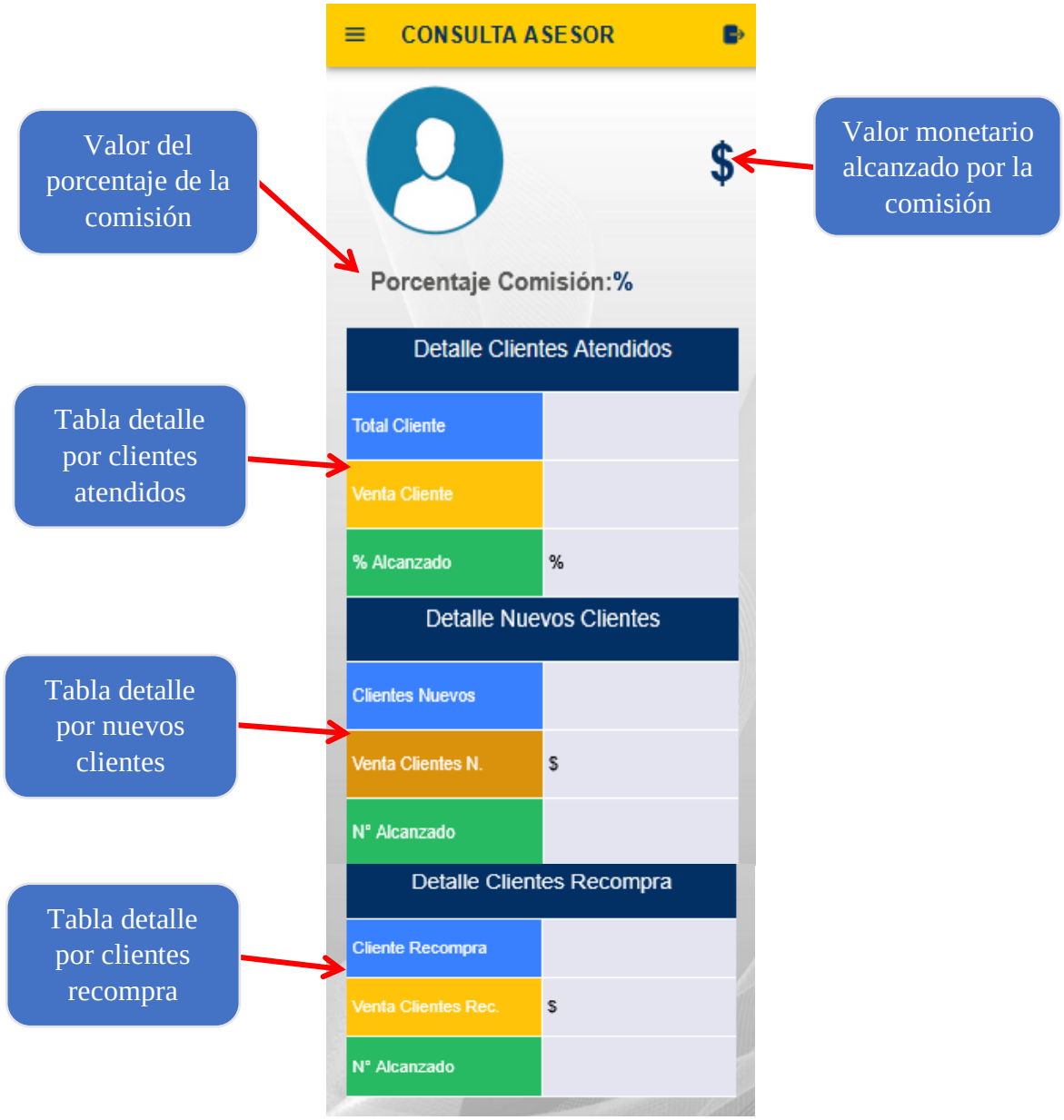
En este módulo se detalla la venta real, la meta a cumplir durante cada mes, se coloca los valores reales de alcance y porcentaje cumplido con el cumplimiento, efectividad, rentabilidad, bono cumplimiento, bono promocional y clientes nuevos.

Pantalla comisión por monto

En este módulo se describe a detalle los criterios para la obtención de la comisión, se muestra el valor alcanzado, el porcentaje necesario para cumplir el monto, la venta neta, presupuesto y el porcentaje del cumplimiento.

Pantalla comisión por efectividad

En este módulo se describe a detalle la composición para la obtención de la comisión, se muestra el valor alcanzado, el porcentaje necesario para cumplir la efectividad y el detalle de clientes atendidos, nuevos clientes y clientes recompra.



Pantalla comisión por rentabilidad

En este módulo se describe a detalle la composición para la obtención de la comisión, se muestra el valor alcanzado y el porcentaje necesario para cumplir la rentabilidad, se describe el detalle del porcentaje de la rentabilidad, rentabilidad objetivo y comisión.

Pantalla bono por monto

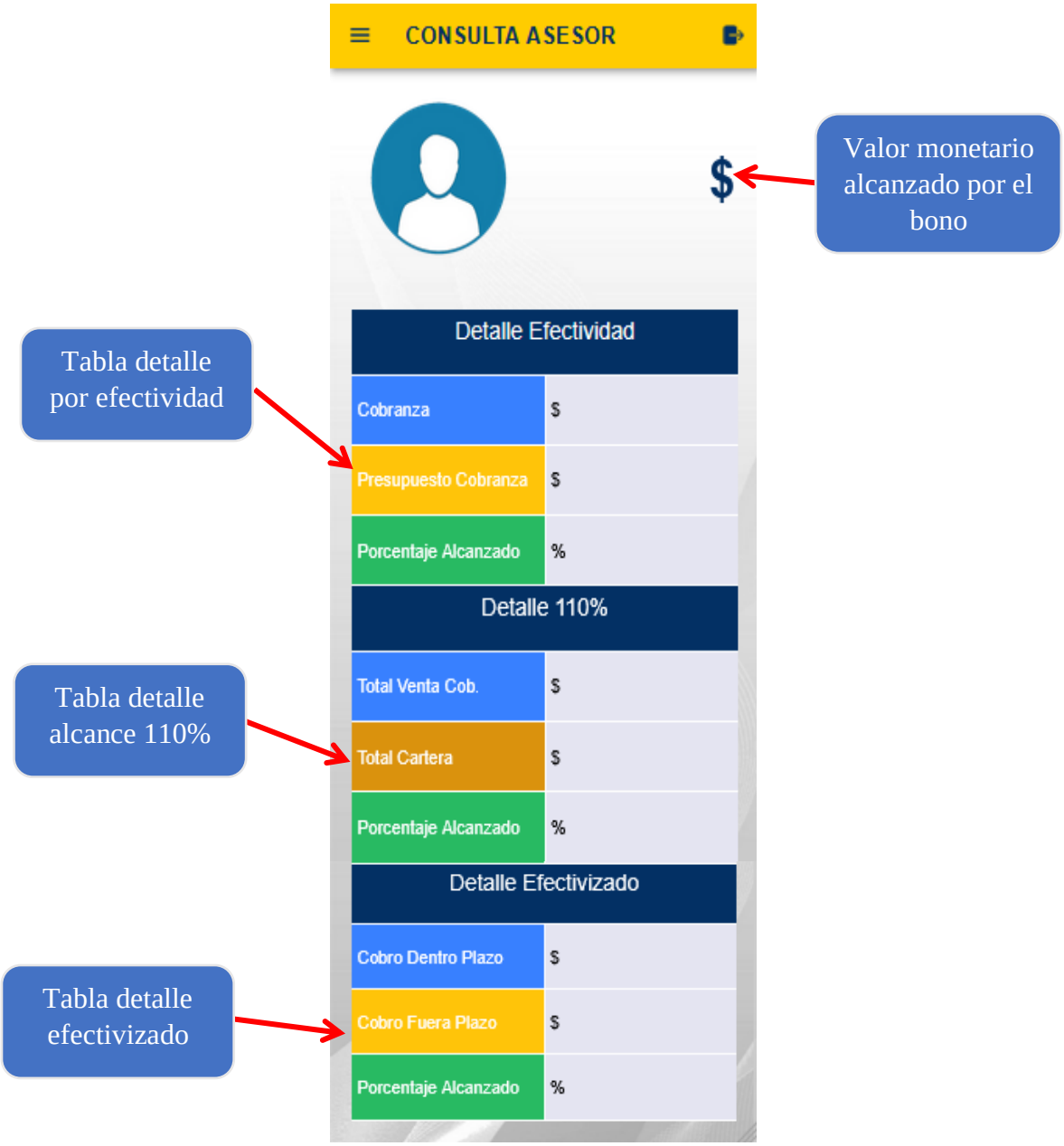
En este módulo se describe a detalle la composición para la obtención del bono, se presenta el valor alcanzado, el total del bono que se puede obtener, el detalle del total de la venta, presupuesto y porcentaje alcanzado.

Pantalla bono por promocional

En este módulo se describe a detalle la composición para la obtención del bono, se presenta el valor alcanzado y el total del bono que se puede obtener, el detalle del total de la venta promocional, presupuesto promocional y porcentaje alcanzado.

Pantalla bono por cartera

En este módulo se describe a detalle la composición para la obtención del bono, se presenta el valor alcanzado y el total del bono que se puede obtener, se detalle por efectividad, detalle 110% y detalle efectividad.



Pantalla total progreso

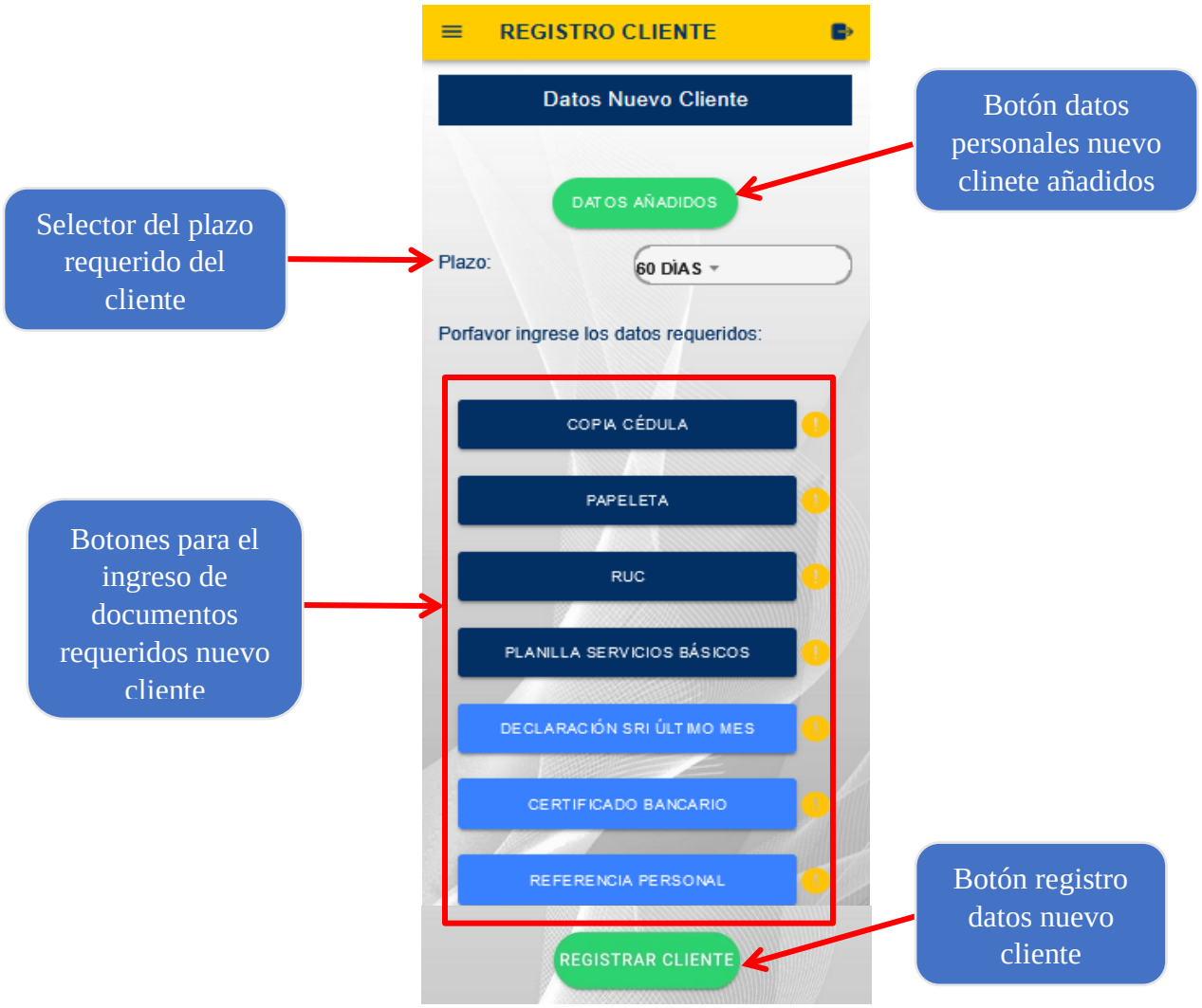
En este módulo se detalla un resumen del total de bonos y comisiones que cada usuario ha obtenido actualmente, se detalla el monto, efectividad, rentabilidad, bono monto, bono promociones, bono cartera y valor total.

Pantalla creación nuevo cliente

En este módulo se permite el ingreso de datos personales y documentación para la creación de un nuevo cliente.

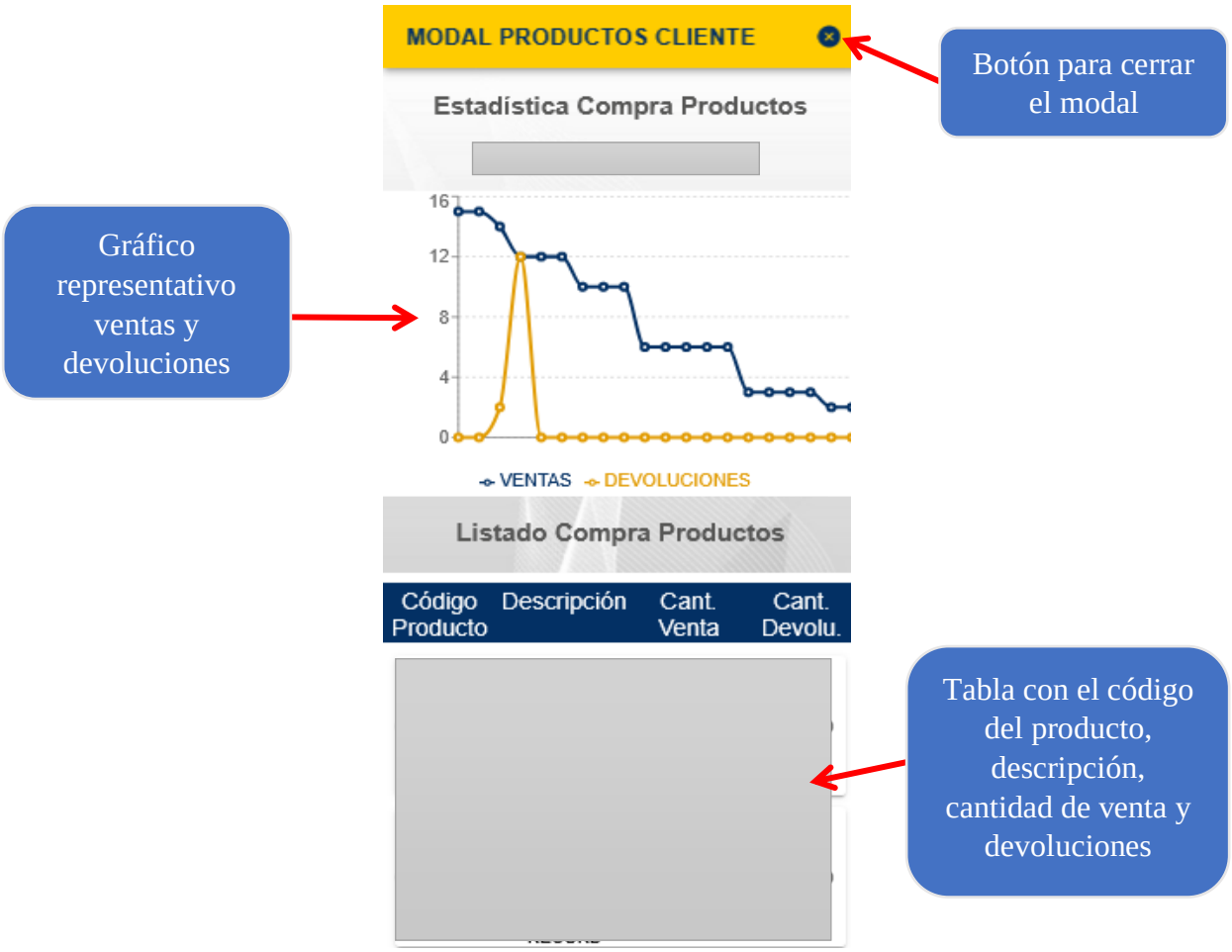
Ingreso datos personales del nuevo cliente.

Ingreso documentación nuevo cliente.



Pantalla ventas clientes

En este módulo se muestran los valores de ventas y devoluciones de los clientes por cada usuario, así como un cuadro estadístico con la cantidad de los productos vendidos y devueltos por cada cliente seleccionado.

Valor ventas y devoluciones por cliente.

Cantidad de ventas y devoluciones de productos.