

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR FACULTAD DE  
INGENIERÍA  
CARRERA DE: INGENIERÍA EN SISTEMAS**



**Trabajo de Titulación**

**Tema:** Aplicativo móvil de gestión de cobranza para instituciones financieras de Economía Popular y Solidaria del Ecuador.

**AUTOR:**

Tribaldos Ortiz Tulio Esteban

QUITO DM, FEBRERO DE 2025

## **DEDICATORIA**

El presente trabajo de titulación lo dedico, primeramente, a mis padres, quienes con sus enseñanzas y apoyo incondicional han sido mi principal inspiración durante todo este proceso de enseñanza y aprendizaje. A mis queridos abuelos, quienes me enseñaron el valor del esfuerzo y la disciplina, inculcándome valores que me impulsaron a ser mejor cada día. A mi pequeña hermana, que me motiva a ser un ejemplo para ella, recordándome la importancia de luchar por mis metas.

Finalmente, dedico este proyecto a todos amigos y profesores, que contribuyeron de forma significativa en mi formación personal y profesional.

Dedico este trabajo de titulación a mi novia Katerin, por ser mi soporte y mi principal compañera durante todo este tiempo. Gracias a su amor, consejos y aliento he logrado convertirme en una mejor persona y superar los momentos más difíciles. Este logro lo comparto contigo con todo mi cariño y gratitud.

## **AGRADECIMIENTO**

Agradezco a Dios por darme la vida y permitirme cursar mis estudios. A mis padres, por brindarme esta valiosa oportunidad y, con gran cariño y esfuerzo, apoyarme en mi formación personal y profesional.

## **RESUMEN**

El trabajo de titulación describe el desarrollo de un aplicativo móvil para la gestión de cobranza en las instituciones financieras de economía popular y solidaria. Para permitir a los clientes afiliados realizar transacciones, evitando el traslado a las sucursales correspondientes a la organización, de esta manera se elimina la barrera geográfica entre la entidad financiera y el cliente.

El objetivo del aplicativo es englobar las acciones de depósitos, consultas de productos crediticios, registro de depósitos y entrega de recibos en una solución informática que permita a la institución maximizar sus tareas de cobro, evitando que estas se vean limitadas únicamente a las oficinas de la empresa.

# ÍNDICE

DEDICATORIA .....	II
AGRADECIMIENTO .....	III
RESUMEN .....	IV
ÍNDICE .....	V
ÍNDICE DE ILUSTRACIONES, GRÁFICOS Y TABLAS .....	IX
ÍNDICE DE ILUSTRACIONES .....	IX
ÍNDICE DE TABLAS .....	XI
CAPÍTULO I: INTRODUCCIÓN .....	12
1. MARCO DE REFERENCIA .....	12
1.1. Definición del Tema .....	12
1.2. Justificación .....	12
1.3. Planteamiento del Problema .....	13
1.4. Objetivo General .....	13
1.5. Objetivos Específicos .....	13
1.6. Antecedentes .....	14
1.7. Alcance .....	14
CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA .....	16
2. Marco Teórico .....	16
2.1. Importancia de las INSTITUCIONES FINANCIERAS DE EPS .....	16
2.2. Proceso de Recaudación .....	16
2.2.1. Registro de Credenciales del Empleado Encargado de la Recaudación .....	16
2.2.2. Solicitud de Cobro por parte del Cliente .....	17
2.2.3. Identificación del Cliente .....	17
2.2.4. Gestión de Productos .....	17
2.2.5. Advertencia de Cobro por parte de la INSTITUCIÓN FINANCIERA DE EPS .....	17
2.2.6. Generación de Reportes .....	17
2.2.7. Impresión de Comprobantes .....	18
2.3. Terminología Esencial del Negocio .....	18
2.4. Marco Tecnológico .....	19
2.4.1. Sistemas Operativos móviles .....	19
2.4.1.1. Android .....	19
2.4.2. Tecnologías Front-end .....	19
2.4.2.1. Figma .....	20

2.4.2.2. Visual Studio Code.....	20
2.4.2.3. TypeScript.....	20
2.4.2.4. Angular.....	20
2.4.2.5. Ionic .....	20
2.4.2.6. Capacitor .....	20
2.4.2.7. Cordova .....	21
2.4.3. Tecnologías Back-end.....	21
2.4.3.1. HTTP Apache .....	21
2.4.3.2. PHP.....	21
2.4.3.3. SLIM Framework .....	21
2.4.3.4. MD5 .....	21
2.4.3.5. Postman.....	22
2.4.3.6. Apache JMeter.....	22
2.4.3.7. Power Designer .....	22
2.4.3.8. SQL.....	22
2.4.3.9. Informix .....	22
2.4.4. Tecnologías adicionales.....	23
2.4.4.1. JSON .....	23
2.4.4.2. Gradle .....	23
2.4.4.3. Android Studio .....	23
2.4.5. Impresora Bluetooth .....	23
2.5. Proceso de Desarrollo .....	24
2.5.1. Proceso unificado (Unified Process – UP) .....	24
2.5.2. Modelo en Espiral .....	24
2.5.3. Modelo vista controlador (MVC).....	25
2.5.4. Desarrollo Orientado a Objetos.....	25
2.6. Desarrollo Ágil de Software.....	26
2.6.1. Prototipado .....	26
CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA .....	28
3. Levantamiento de Requerimientos .....	28
3.1. Entrevista a Ingeniero en Soluciones Informáticas para INSTITUCIONES FINANCIERAS DE EPS .....	28
3.2. Requerimientos Funcionales .....	31
3.3. Requerimientos No Funcionales .....	32
3.4. Diagrama de Caso de Uso .....	32

3.5. Usuarios del Sistema.....	33
3.5.1. Usuario/Agente Recaudador .....	33
3.5.2. Socio o Cliente .....	34
3.6. Arquitectura de la Aplicación.....	34
3.6.1. Arquitectura Cliente-servidor .....	34
3.7. Componentes Clave de la Arquitectura de la Aplicación.....	35
3.7.1. Capa de presentación (UI/UX).....	35
3.7.2. Capa Lógica del Negocio.....	36
3.7.2.1. Control de Inicio de Sesión .....	36
3.7.2.2. Depósitos en Cuentas de Ahorro .....	36
3.7.2.3. Monto Máximo y Mínimo de Depósitos en Cuentas de Ahorro .....	37
3.7.2.4. Consultas de Productos de Crédito .....	37
3.7.2.5. Validación de Número de Cédula .....	37
3.7.2.6. Reportes .....	38
3.7.2.7. Impresiones Bluetooth .....	39
3.7.3. Funciones Principales .....	39
3.7.3.1. Gestión de Flujo de Datos .....	39
3.7.3.2. Respuesta y Validación de Resultados.....	39
3.7.4. Módulos del Proyecto .....	40
3.7.4.1. Módulo de Depósitos.....	40
3.7.4.2. Módulo de Consultas.....	40
3.7.4.3. Módulo de Reportes .....	40
3.7.4.4. Módulo de Configuración .....	40
3.7.5. Capa de Datos.....	41
3.7.5.1. Base de Datos Relacional .....	41
CAPITULO IV: DESARROLLO E IMPLEMENTACIÓN.....	53
4. Desarrollo de Back-end .....	53
4.1. Arquitectura del Back-end .....	53
4.1.1. Desarrollo de APIs .....	53
4.1.1.1. Rutas Principales .....	53
4.1.2. Manejo de Errores.....	55
4.1.3. Pruebas de Back-end .....	55
4.2. Prototipo de la Interfaz de Usuario.....	55
4.2.1. Refinamiento del Prototipo .....	58
4.3. Funcionalidades del Sistema.....	58

3.8. Desarrollo de Interfaces .....	62
4.4.1. Pantalla de Inicio de Sesión .....	62
4.4.2. Pantalla de Inicio.....	63
3.8.3. Menú de Navegación.....	64
3.8.4. Pantallas de Depósitos.....	64
3.8.5. Pantallas de Consultas.....	65
3.8.6. Pantallas de Reportes .....	66
3.8.7. Pantalla de Configuración .....	67
3.8.8. Recibos o Comprobantes.....	68
4.5. Implementación de la Aplicación.....	69
4.5.1. Pruebas Funcionales .....	69
4.5.2. Pruebas no Funcionales .....	69
4.5.3. Pruebas de Carga.....	70
4.5. Pruebas de Usabilidad.....	70
4.6. Implementación del Aplicativo en Formato APK.....	70
4.7. Credenciales Utilizadas Durante las Pruebas.....	71
4.8. Resultados .....	72
4.9. Resultados de pruebas de usabilidad .....	90
CONCLUSIONES Y RECOMENDACIONES.....	93
5.1. Conclusiones .....	93
5.2. Recomendaciones .....	93
BIBLIOGRAFÍA.....	95
ANEXOS.....	97

# ÍNDICE DE ILUSTRACIONES, GRÁFICOS Y TABLAS

## ÍNDICE DE ILUSTRACIONES

<b>Ilustración 1.</b> Modelo de desarrollo en espiral. ....	25
<b>Ilustración 2.</b> Ciclo de vida Prototipado.....	27
<b>Ilustración 3.</b> Diagrama de caso de uso del aplicativo.....	33
<b>Ilustración 4.</b> Arquitectura de la aplicación. ....	35
<b>Ilustración 5.</b> Validación de cédula de ciudadanía.....	38
<b>Ilustración 6.</b> Base de datos general.....	52
<b>Ilustración 7.</b> Prueba de APIS con Postman.....	55
<b>Ilustración 8.</b> Diseño del prototipo de la interfaz de usuario 1.....	56
<b>Ilustración 9.</b> Diseño del prototipo de la interfaz de usuario 2.....	56
<b>Ilustración 10.</b> Diseño del prototipo de la interfaz de usuario 3.....	57
<b>Ilustración 11.</b> Diseño del prototipo de la interfaz de usuario 4.....	57
<b>Ilustración 12.</b> Pantalla de inicio de sesión. ....	63
<b>Ilustración 13.</b> Página de inicio.....	64
<b>Ilustración 14.</b> Menú de navegación. ....	64
<b>Ilustración 15.</b> Pantallas de depósitos.....	65
<b>Ilustración 16.</b> Pantallas de consultas.....	66
<b>Ilustración 17.</b> Pantallas de reportes.....	67
<b>Ilustración 18.</b> Pantallas de configuración. ....	68
<b>Ilustración 19.</b> Comprobante o recibo del depósito. ....	69
<b>Ilustración 20.</b> Generación del archivo APK firmado. ....	71
<b>Ilustración 21.</b> Instalación de la aplicación.....	78
<b>Ilustración 22.</b> Se muestra el icono funcional del aplicativo. ....	78
<b>Ilustración 23.</b> Apertura del aplicativo. ....	79
<b>Ilustración 24.</b> Inicio de sesión. ....	79
<b>Ilustración 25.</b> Se muestran las pantallas del módulo de depósitos.....	80
<b>Ilustración 26.</b> Recibo correspondiente al depósito correspondiente.....	81
<b>Ilustración 27.</b> Se muestran las pantallas del módulo de consultas.....	82
<b>Ilustración 28.</b> Se muestran las pantallas del módulo de reportes.....	83
<b>Ilustración 29.</b> Se muestran las pantallas del módulo de configuración. ....	84
<b>Ilustración 30.</b> Cierre de sesión desde la página de inicio. ....	85
<b>Ilustración 31.</b> Manejo de excepciones.....	86
<b>Ilustración 32.</b> Cierre de sesión del sistema. ....	87
<b>Ilustración 33.</b> Reporte de rendimiento – API de inicio de sesión.....	88
<b>Ilustración 34.</b> Reporte de rendimiento – API de consulta de información del socio...88	
<b>Ilustración 35.</b> Reporte de rendimiento – API de acceso a las cuentas de ahorro del cliente.....	88
<b>Ilustración 36.</b> Reporte de rendimiento – API de depósitos. ....	88
<b>Ilustración 37.</b> Reporte de rendimiento – API de acceso a los créditos de un cliente. 89	
<b>Ilustración 38.</b> Reporte de rendimiento - API de acceso a la información del crédito. 89	
<b>Ilustración 39.</b> Reporte de rendimiento - API de acceso a la información de la cuota. ....	89
<b>Ilustración 40.</b> Reporte de rendimiento - API de acceso a reportes.....	89
<b>Ilustración 41.</b> Reporte de rendimiento - API de impresión del recibo.....	90

<b>Ilustración 42.</b> Reporte de rendimiento - API de cierre de sesión. ....	90
<b>Ilustración 43.</b> Respuesta al cuestionario de pruebas de usabilidad 1. ....	90
<b>Ilustración 44.</b> Respuesta al cuestionario de pruebas de usabilidad 2. ....	91
<b>Ilustración 45.</b> Respuesta al cuestionario de pruebas de usabilidad 3. ....	91
<b>Ilustración 46.</b> Diccionario de la base de datos. ....	97
<b>Ilustración 47.</b> Código del Back-end. ....	97
<b>Ilustración 48.</b> Código del front-end. ....	98
<b>Ilustración 49.</b> Generación del archivo APK en Android Studio. ....	98
<b>Ilustración 50.</b> Cuestionario post-test para pruebas de usabilidad.....	99

## ÍNDICE DE TABLAS

<b>Tabla 1:</b> Requerimientos funcionales. ....	31
<b>Tabla 2:</b> Requerimientos no funcionales. ....	32
<b>Tabla 3:</b> Tabla usuarios. ....	41
<b>Tabla 4:</b> Empleados vinculados con usuarios. ....	42
<b>Tabla 5:</b> Almacenamiento de acceso de usuario cuando inicia sesión. ....	43
<b>Tabla 6:</b> Datos de clientes. ....	43
<b>Tabla 7:</b> Datos de teléfono de clientes. ....	45
<b>Tabla 8:</b> Datos de las cuentas que posee un cliente. ....	45
<b>Tabla 9:</b> Datos de tipos de cuentas que puede tener una cuenta. ....	46
<b>Tabla 10:</b> Datos de movimientos de una cuenta. ....	47
<b>Tabla 11:</b> Detalles y forma de pago de movimiento en una cuenta. ....	48
<b>Tabla 12:</b> Detalles de depositante de terceros en una cuenta. ....	48
<b>Tabla 13:</b> Detalle de efectivo en movimientos de cuentas. ....	49
<b>Tabla 14:</b> Denominaciones de efectivo. ....	49
<b>Tabla 15:</b> Detalle de denominación de efectivo. ....	49
<b>Tabla 16:</b> Créditos que posee un cliente. ....	50
<b>Tabla 17:</b> Detalles de los créditos que posee un cliente. ....	50
<b>Tabla 18:</b> Detalle de rubros que posee un dividendo. ....	51
<b>Tabla 19:</b> Funcionalidades ejecutadas por el sistema. ....	58
<b>Tabla 20:</b> Funcionalidades ejecutadas por el agente recaudador. ....	61
<b>Tabla 21:</b> Credenciales de los agentes de recaudación. ....	71
<b>Tabla 22:</b> Pruebas Funcionales. ....	72
<b>Tabla 23:</b> Pruebas no funcionales. ....	76
<b>Tabla 24:</b> Pruebas de carga N°1. ....	77
<b>Tabla 25:</b> Pruebas de carga N°2. ....	77
<b>Tabla 26:</b> Dispositivos en los que se instaló la aplicación. ....	87

# **CAPÍTULO I: INTRODUCCIÓN**

## **1. MARCO DE REFERENCIA**

### **1.1. Definición del Tema**

En el actual escenario de la economía popular y solidaria en Ecuador, muchas cooperativas, cajas, bancos comunales, asociaciones y fundaciones de ahorro y crédito enfrentan desafíos para que sus sistemas informáticos les permitan realizar transacciones sin que el alcance geográfico sea un limitante. De ahí surge la necesidad de desarrollar aplicaciones móviles que simplifiquen la gestión de transacciones tanto para las instituciones financieras, sus empleados y sus clientes, ofreciendo mayor accesibilidad y versatilidad en la administración de los servicios financieros.

### **1.2. Justificación**

Para una mejor identificación, las cooperativas, cajas, bancos comunales, asociaciones y fundaciones de ahorro y crédito, en adelante se les denominarán instituciones financieras de economía popular y solidaria – INSTITUCIONES FINANCIERAS DE EPS.

Las INSTITUCIONES FINANCIERAS DE EPS requieren la implementación de un sistema, que simplifique las labores de recaudación de productos de ahorro y gestión de cobranza. Este aplicativo estará dirigida principalmente a los empleados de la INSTITUCIÓN FINANCIERA DE EPS, quienes serán los usuarios principales. El empleado encargado de la recaudación puede agilizar y mejorar la eficiencia en la gestión de los depósitos realizados por los clientes de la INSTITUCIÓN FINANCIERA DE EPS. Además, el sistema habilita la posibilidad de consultar información sobre créditos activos del cliente, incluyendo detalles del valor a pagar de cada una de las cuotas, según la tabla de amortización del crédito. Asimismo, se incluyen reportes de las transacciones diarias realizadas por el agente de recaudación, permitiendo llevar un control de las cuentas y operaciones relacionadas.

Las INSTITUCIONES FINANCIERAS DE EPS están obligadas a desarrollar nuevos productos y servicios orientados a mejorar la satisfacción de sus socios, por lo tanto, el sistema deberá proporcionar a los empleados de la INSTITUCIÓN FINANCIERA DE EPS la posibilidad de realizar diversas transacciones que, a través de la optimización tanto del recurso humano como tecnológico, les eviten a sus clientes tener que desplazarse a una oficina.

### **1.3. Planteamiento del Problema**

Desarrollar una aplicación móvil para INSTITUCIONES FINANCIERAS DE EPS que permita generar nuevos canales de recaudación para transacciones de ahorros y créditos, sin que sus clientes tengan que movilizarse a una oficina, ya que recibirán la visita de los gestores de cobranza, quienes, con un celular y una impresora Bluetooth portátil a través del sistema, realizarán transacciones ya sea en el domicilio, lugar de trabajo o negocio.

### **1.4. Objetivo General**

Implementar un sistema móvil que permita a las instituciones de economía popular y solidaria, agilizar sus procesos de recaudación de depósitos en cuentas de ahorros y créditos, utilizando dispositivos Android con tecnología Bluetooth para imprimir recibos de transacciones.

### **1.5. Objetivos Específicos**

- Analizar la información correspondiente sobre las acciones de gestión de cobranza en INSTITUCIONES FINANCIERAS DE EPS.
- Diseñar un modelo de datos y flujos de trabajo que permitan la implementación del aplicativo móvil.
- Implementar la metodología de prototipado para el desarrollo del software.
- Implementar la funcionalidad de depósitos directos en cuentas de ahorro.
- Implementar la funcionalidad de consulta de cuotas de crédito.
- Implementar la funcionalidad de registro de transacciones o reportes por día.
- Integrar impresoras portátiles con tecnología Bluetooth para la impresión de comprobantes de transacciones.

- Realizar pruebas funcionales, no funcionales, de carga y usabilidad para garantizar la eficiencia del sistema.

## **1.6. Antecedentes**

“Las INSTITUCIONES FINANCIERAS DE EPS, desempeñan un papel fundamental en la inclusión financiera, enfocándose espacialmente en áreas rurales y comunidades desatendidas” (Ramón Arteaga-Macías, 2017) facilitando el alcance a servicios financieros esenciales promoviendo al desarrollo económico local.

Sin embargo, estas instituciones enfrentan desafíos cruciales, como la dispersión geográfica de sus socios o clientes, dificultando la prestación de servicios financieros. Actualmente, las actividades de recaudación en las organizaciones requieren que el cliente se desplace físicamente a las oficinas para realizar depósitos u otras transacciones, generando malestar entre los clientes debido al transporte, tiempo y los posibles contratiempos que puedan surgir dentro de las sucursales. Limitando las capacidades de las INSTITUCIONES FINANCIERAS DE EPS para llegar a las comunidades, restringiendo su alcance.

La implementación de aplicativos móviles dentro de las instituciones financieras, poseen el potencial de cambiar el sector financiero en el Ecuador. Debido a que estas soluciones facilitan las actividades humanas, actuando como un medio entre las instituciones financieras y sus socios, permitiendo interactuar de forma más eficiente con los servicios financieros.

Ofreciendo una plataforma que integra tecnologías como impresoras portátiles y dispositivos móviles, que permiten a los empleados realizar transacciones de forma directa en el domicilio, lugar de trabajo o negocio de un socio o cliente.

## **1.7. Alcance**

El proyecto de desarrollo del sistema móvil tiene como objetivo crear una solución innovadora para INSTITUCIONES FINANCIERAS DE EPS, permitiendo un acceso ilimitado a dispositivos de recaudadores a través de una aplicación Android. Las funcionalidades principales incluyen un inicio de sesión para oficiales de recaudación, consulta en línea de saldos y valores a pagar por créditos, la capacidad de realizar depósitos directos en cuentas de ahorro desde dispositivos móviles, así como la integración con impresoras Bluetooth para la impresión de recibos de transacciones. Se realizarán pruebas exhaustivas para garantizar la seguridad y funcionalidad del sistema, además de

la generación de informes diarios de transacciones para análisis y control financiero. En conjunto, este proyecto busca modernizar y optimizar los procesos de recaudación en INSTITUCIONES FINANCIERAS DE EPS, mejorando la eficiencia operativa y la experiencia del cliente a través de tecnología móvil y herramientas de gestión financiera innovadoras.

## **CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA**

### **2. Marco Teórico**

#### **2.1. Importancia de las INSTITUCIONES FINANCIERAS DE EPS**

Las INSTITUCIONES FINANCIERAS DE EPS, desempeñan un papel fundamental en la inclusión financiera, para ciertas comunidades desatendidas. Estas entidades facilitan el acceso a servicios financieros esenciales, para promover el desarrollo económico, apoyando a la reducción de la pobreza.

“La Superintendencia de Economía Popular y Solidaria (SEPS), al mes de septiembre del año 2021, contabilizó 512 cooperativas de ahorro y crédito y mutualistas a nivel nacional, con más de 8 millones de certificados de aportación” (Sánchez, 2021). reflejando una amplia cobertura dentro del sistema financiero ecuatoriano.

#### **2.2. Proceso de Recaudación**

Se define como un conjunto de actividades que tienen como objetivo la administración de fondos provenientes de pagos, depósitos o contribuciones realizadas por clientes. “El proceso incluye desde la solicitud de pagos, manejo de transacciones, hasta la emisión de comprobantes, garantizando el cumplimiento de las normativas financieras aplicables” (David et al., 2020)

Todas estas acciones pueden concretarse desde las sucursales correspondientes a las instituciones financieras. Sin embargo, es necesario detallar de forma secuencial, cómo se realizan las gestiones de cobranza dentro del aplicativo que se desea implementar.

##### **2.2.1. Registro de Credenciales del Empleado Encargado de la Recaudación**

Para una mejor identificación a los empleados de la institución financiera, encargados de la gestión de cobranza se les denomina agentes de recaudación. La INSTITUCIÓN FINANCIERA DE EPS es la encargada de registrar al agente recaudador dentro del sistema. Solo los agentes con credenciales pueden acceder al aplicativo y

realizar operaciones.

### **2.2.2. Solicitud de Cobro por parte del Cliente**

El cliente o socio solicita que un agente de cobranza se acerque a un punto de encuentro, para realizar acciones de cobro, mediante el número de contacto de atención al cliente, propio de la INSTITUCIÓN FINANCIERA DE EPS.

### **2.2.3. Identificación del Cliente**

Los socios son identificados con su número de cédula, que es validada y, por lo tanto, registrada en el sistema de la INSTITUCIÓN FINANCIERA DE EPS.

### **2.2.4. Gestión de Productos**

El agente recaudador tiene la facultad de consultar y operar sobre los productos que tiene abiertos un cliente, ya sean cuentas de ahorro o cuentas de crédito. Como toda cuenta de ahorro, esta le permite al cliente solicitar depósitos, mientras que el producto de crédito requiere pagos de cuotas mensuales con intereses. El cobro de crédito se descontará directamente de las cuentas de ahorro del socio. Por lo tanto, al agente como se le da acceso a la información del crédito para conocer sus valores a pagar.

### **2.2.5. Advertencia de Cobro por parte de la INSTITUCIÓN FINANCIERA DE EPS**

Sin embargo, cuando los productos de ahorro de un socio afiliado no poseen fondos, y tiene cuotas de crédito por pagar, se procede a contactar al socio o cliente para informarle que se requiere el cobro de la cuota de forma inmediata. Es otro escenario, en donde actúa un agente recaudador, con la finalidad de realizar la gestión de cobranza a dicho socio en el menor tiempo posible.

### **2.2.6. Generación de Reportes**

Se generan reportes diarios con todas las transacciones realizadas, brindando información sobre el cliente y el producto al que se le realizó una transacción. Los reportes sirven como respaldo tanto para el agente como para la institución.

## 2.2.7. Impresión de Comprobantes

Las transacciones se respaldan con un comprobante o recibo físico.

## 2.3. Terminología Esencial del Negocio

- **Cuentas de ahorro:** Es un producto financiero que permite a los socios depositar y almacenar dinero, generando intereses. Por lo tanto, solo las cuentas activas pueden recibir depósitos. A
- **Créditos:** Producto financiero en el cual la institución financiera de EPS otorga un préstamo a un socio, quien debe devolverlo en cuotas mensuales. Para ello, los créditos deben estar activos para ser consultados.
- **Tabla de amortización:** Documento que desglosa de forma detallada el cronograma de pagos de un préstamo o crédito, diseñado para facilitar la comprensión del proceso de pago y el saldo pendiente que posee un cliente.
- **Dividendo o cuota:** Un dividendo o cuota se define como una porción específica de una deuda de crédito, que debe ser pagada de forma periódica por el socio asociado a la institución. Cada crédito otorgado se divide en dividendos que incluyen el capital que es parte del monto original del crédito, intereses por costo financiero del uso de crédito, seguro, interés de mora u otros cargos.
- **Rubro:** Un rubro es una categoría que representa partes específicas de los cargos aplicables a un crédito o transacción. Los rubros permiten conocer los detalles del dividendo, brindando información al cliente acerca del capital total a pagar para todas las cuotas.
- **Tasa de interés:** Es un porcentaje aplicado al capital pendiente, para calcular el costo financiero o los intereses de una transacción. El porcentaje se hace presente en cada rubro de la cuota de crédito.
- **Movimientos de cuenta:** Registro detallado de todas las operaciones realizadas en una cuenta o producto financiero.

## **2.4. Marco Tecnológico**

En el ámbito del marco tecnológico, se definen que herramientas han sido utilizadas durante el desarrollo del proyecto, especificando las tecnologías de front-end y back-end que componen nuestro aplicativo móvil. Como, por ejemplo: bases de datos, APIs, lenguajes de programación, herramientas de conexión a base de datos, etc. Adicionalmente, se detallan los procesos y metodologías de desarrollo que fueron implementadas. Cabe mencionar, antes de aclarar las tecnologías es necesario definir las terminologías de lo que son tecnologías front-end y back-end todo dentro de este mismo apartado.

### **2.4.1. Sistemas Operativos móviles**

Se detalla el sistema operativo o software para el cual está destinado el funcionamiento del aplicativo móvil.

#### **2.4.1.1. Android**

“Se destaca como uno de los sistemas operativos móviles más ampliamente adoptados, permitiendo una accesibilidad amplia para varios dispositivos alrededor del mundo” (Miguel, 2025). Es conocido por su capacidad para gestionar aplicaciones y ejecutar procesos en segundo plano. Adicionalmente, cuenta con la integración de servicios proporcionados por Google, permitiendo el desarrollo ilimitado de aplicaciones.

### **2.4.2. Tecnologías Front-end**

“El término front-end hace referencia a la interfaz gráfica o GUI (Graphical User Interface) de cualquier sistema ya sea web o móvil” (Jesús & Caballero, 2016). Por lo tanto, se define como aquello con lo que los usuarios interactúan mediante elementos visuales que facilitan la comprensión del aplicativo, permitiendo un uso más eficiente e intuitivo para el usuario.

En cuanto a tecnología, las herramientas utilizadas permiten la elaboración de la interfaz gráfica. Estas herramientas varían desde frameworks de desarrollo hasta el lenguaje de programación necesario para su ejecución.

### **2.4.2.1. Figma**

“Herramienta de diseño basada en la nube que permite crear, editar y prototipar interfaces de usuario y experiencias de usuario” (Alcaraz, 2024). Ofrece capacidades de colaboración en tiempo real y de fácil acceso desde cualquier dispositivo.

### **2.4.2.2. Visual Studio Code**

“Editor de código fuente, gratuito, capaz de soportar múltiples lenguajes de programación” (Cuadrado, 2022). Permite a los desarrolladores crear aplicaciones de forma eficiente.

### **2.4.2.3. TypeScript**

“Lenguaje de programación de código abierto desarrollado por Microsoft en el año 2012, caracterizado por implementar muchos mecanismos de programación orientada a objetos” (Pucciarelli, 2020).

### **2.4.2.4. Angular**

“Es un framework de código abierto mantenido por Google, utilizado para crear páginas web tipo SPA (Single Page Application). Significa que la página se carga solo al inicio y las actualizaciones se producen sin la tediosa necesidad de recargar la página de forma completa” (Pucciarelli, 2020).

### **2.4.2.5. Ionic**

“Conjunto de herramientas de interfaz de usuario para el desarrollo de aplicaciones móviles de alta calidad, usando tecnologías web como HTML, CSS y JavaScript, integrados a frameworks populares como Angular” (Churata & Fernando, 2017).

### **2.4.2.6. Capacitor**

“Framework multiplataforma creado por Ionic, que permite a los desarrolladores construir aplicaciones para IOS, Android y aplicaciones web.” (Alejandro et al., 2020).

### **2.4.2.7. Cordova**

“Herramienta que permite acceder a las funcionalidades nativas del dispositivo, empaquetando la aplicación para ejecutarse en sistemas operativos móviles.” (Herrera et al., 2021)

### **2.4.3. Tecnologías Back-end**

“El back-end se define como todo aquello que el usuario no visualiza; las tecnologías de back-end se encargan de gestionar la funcionalidad general de la aplicación” (Geovanni & Espino, 2021). Cuando el usuario interactúa con las tecnologías de front-end, esta interacción envía una solicitud al back-end, que devuelve una respuesta desde la base de datos. El back-end interactúa con servidores, microservicios y APIs para procesar y gestionar los datos necesarios para el funcionamiento de la aplicación.

#### **2.4.3.1. HTTP Apache**

“Software de servidor web gratuito y de código abierto. Permite a los propietarios de sitios web alojar contenido ya que es uno de los servidores más antiguos y confiables” (Márquez Díaz et al., 2025)

#### **2.4.3.2. PHP**

“Lenguaje de programación destinado a la creación de aplicaciones y sitios web, favoreciendo la conexión entre servidores e interfaz de usuario” (Llatas Yovera et al., 2017). Entre sus beneficios se encuentra el hecho de ser de código abierto y su constante perfeccionamiento, gracias a su comunidad de desarrolladores.

#### **2.4.3.3. SLIM Framework**

“Microframework de PHP diseñado para la construcción de APIs. Proporciona funciones básicas de enrutamiento, manejo de solicitudes y respuestas, adecuándolo para proyectos que necesitan un despliegue rápido” (Llatas Yovera et al., 2017).

#### **2.4.3.4. MD5**

“MD5 se define como un protocolo criptográfico, basado en una función hash, el

cual verifica que la contraseña que se envía, mediante solicitudes HTTP, inicios de sesión o APIs coincida con la contraseña almacenada en la base de datos” (José Ramírez López & Hernández Audelo, 2020).

#### **2.4.3.5. Postman**

“Herramienta disponible tanto para sistemas operativos como Windows y Mac. Está compuesta por diferentes herramientas y utilidades que permiten realizar distintas tareas dentro del mundo de API REST” (L. Von Chong, 2021) tales como creación de peticiones a APIs, pruebas para validar el comportamiento de APIs y la creación de entornos de trabajo con el uso de variables globales y locales.

#### **2.4.3.6. Apache JMeter**

“Herramienta de código abierto diseñada para realizar pruebas de carga, análisis de rendimiento de aplicaciones, servicios web y demás tecnologías orientadas a la red” (V.Chong, 2025). Esta tecnología fue utilizada para realizar pruebas de carga, para comprobar el rendimiento del aplicativo en situaciones de carga intensa.

#### **2.4.3.7. Power Designer**

“Programa utilizado, en su mayoría para la modelización de datos, permitiendo visualizar, analizar y manipular de forma sencilla y eficiente metadatos, con la finalidad de crear una arquitectura de información eficaz” (Villagómez et al., 2020) . Esta herramienta se usa principalmente para el modelamiento de bases de datos.

#### **2.4.3.8. SQL**

“Lenguaje estándar para el manejo y manipulación de bases de datos relacionales. Se usa para realizar consultas, insertar, actualizar y eliminar datos” (García, 2010). Adicionalmente, es usado para definir la estructura de la base de datos y administrar permisos de acceso.

#### **2.4.3.9. Informix**

“Base de datos de base empresarial caracterizada por su alto rendimiento, siempre activa, altamente escalable, optimizada para cargas de trabajo transaccionales”

(Villagómez et al., 2020)

## **2.4.4. Tecnologías adicionales**

Estas herramientas se utilizaron para la creación y desarrollo del aplicativo en formato APK, para evaluaciones y pruebas, y posteriormente distribuir el aplicativo para el uso de la INSTITUCIÓN FINANCIERA DE EPS.

### **2.4.4.1. JSON**

“Formato ligero de intercambio de datos, basado en una estructura de texto plano organizado en pares o listas” (Bárbara et al., 2024). JSON es ampliamente usado en aplicaciones para transmitir datos entre un cliente y un servidor, gracias a su compatibilidad con una gran cantidad de lenguajes de programación.

### **2.4.4.2. Gradle**

Herramienta que permite la automatización de compilación de código abierto, se centra en la flexibilidad y el rendimiento.

### **2.4.4.3. Android Studio**

“Entorno de desarrollo integrado oficial, utilizado para el desarrollo de aplicaciones para el sistema operativo Android” (Miguel, 2025). Ofrece funciones que mejoran la productividad al momento de desarrollar aplicativos.

## **2.4.5. Impresora Bluetooth**

“Bluetooth se establece como un estándar de comunicación inalámbrica, que permite la transmisión de datos a corta distancia, siendo ideal para conectar dispositivos móviles, auriculares, impresoras, etc.” (José Efrén Pérez Carmona, 2008)

Para el desarrollo del aplicativo y las impresiones de recibos, se utilizó una impresora térmica (MINI Thermal Printer). Su principal característica es su tamaño, permitiendo que esta pueda ser transportada fácilmente de un lugar a otro. Otro de sus beneficios es que utilizan el calor como medio para producir impresiones en papel, a comparación de impresoras que usan tinta para imprimir, esta puede lograr a degradarse

y hacer poco visible su contenido, garantizando durabilidad al momento de generar recibos o comprobantes. Para el uso de esta impresora, es necesario adquirir rollos de papel de cincuenta y ocho milímetros de ancho.

## **2.5. Proceso de Desarrollo**

En este apartado se describe a detalle el proceso de desarrollo de software y las metodologías implementadas durante la elaboración del proyecto.

“Un proceso de desarrollo de software se define como una estructura a implementar con el propósito de facilitar el desarrollo de un producto” (Luis & Méndez, 2013). Por lo tanto, se utilizará el proceso unificado (Unified Process – UP), ya que establece los procedimientos genéricos del proyecto que son adaptados según el contexto del producto a desarrollar.

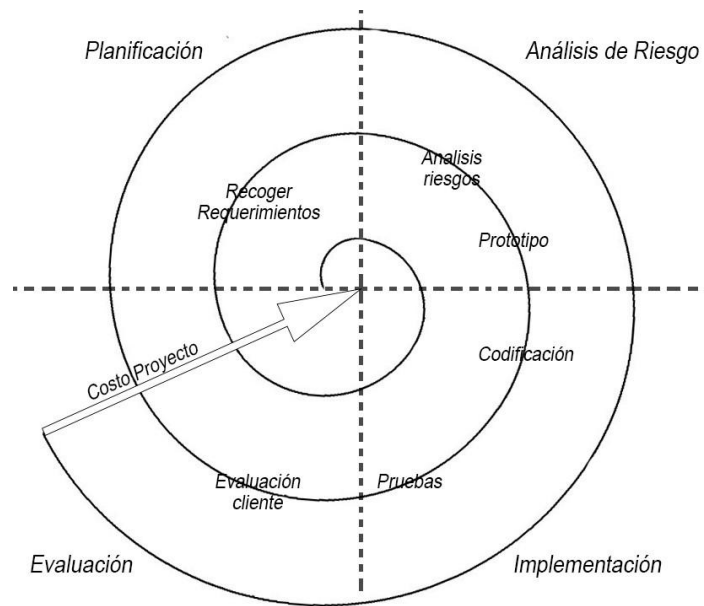
### **2.5.1. Proceso unificado (Unified Process – UP)**

Proceso encargado de transformar los requerimientos de un usuario en un sistema de software. “Funciona como un marco de trabajo que tiene la funcionalidad de ser personalizado para la mayoría de las organizaciones y proyectos específicos” (Luis & Méndez, 2013). El ciclo de vida en el proceso unificado es una implementación del desarrollo en espiral. En donde, el ciclo de vida ordena las tareas en fases o iteraciones.

### **2.5.2. Modelo en Espiral**

Se define como un modelo que combina las metodologías de desarrollo lineal (Cascada) y metodologías iterativas. Su principal beneficio que otorga a los proyectos es minimización de riesgos durante el proceso de desarrollo. Se basa en el refinamiento continuo del producto, ya que en cada fase los productos son extensiones de otros productos. Adicionalmente, el modelo se caracteriza por el énfasis en una fuerte interacción con los usuarios antes de la implementación, aplicada a través de prototipos a lo largo del trabajo.

**Ilustración 1. Modelo de desarrollo en espiral.**



**Nota:** Tomado de análisis y diseño del sistema de gestión académica de postgrados de la universidad de Cuenca (sgap) [Fotografía], Luis & Méndez 2013.

### 2.5.3. Modelo vista controlador (MVC)

Se define como un patrón que surge con el objetivo de reducir el esfuerzo de programación. “El patrón MVC es un paradigma que nos permite dividir las partes que conforman una aplicación, en el modelo, las vistas y el controlador” (Enríquez et al., 2023). Permitiendo la implementación por separado de cada elemento, garantizando la actualización y mantenimiento de nuestro aplicativo de forma sencilla en un corto periodo de tiempo.

### 2.5.4. Desarrollo Orientado a Objetos

Tipo de desarrollo que orienta al analista a encapsular datos y procesos en entidades, que definirán tipos de objetos. Sirve como guía para tener en cuenta el proyecto desde la etapa de análisis y durante todo el diseño en términos de clases y objetos. Durante el proyecto se hace uso del desarrollo orientado objetos, para encapsular los datos y procesos de las INSTITUCIONES FINANCIERAS DE EPS, en clases y objetos, implementando una estructura determinada dentro del sistema, provocando beneficios como reutilización de código y facilidad de implementar características nuevas.

## 2.6. Desarrollo Ágil de Software

Metodologías de desarrollo basadas en un desarrollo iterativo e incremental, donde, los requerimientos evolucionan durante todo el proceso de desarrollo de software. Basado en la filosofía, en donde el desarrollador y usuario entienden en su totalidad el problema y la complejidad del sistema. Se caracteriza por su respuesta al cambio en vez de seguir una estricta planificación.

### 2.6.1. Prototipado

Se define como una técnica que consiste en desarrollar prototipos del sistema, para capturar la información específica acerca de los requerimientos de los usuarios. En la cual se inicia por un prototipo, conocida como una versión en pequeña escala del sistema deseado.

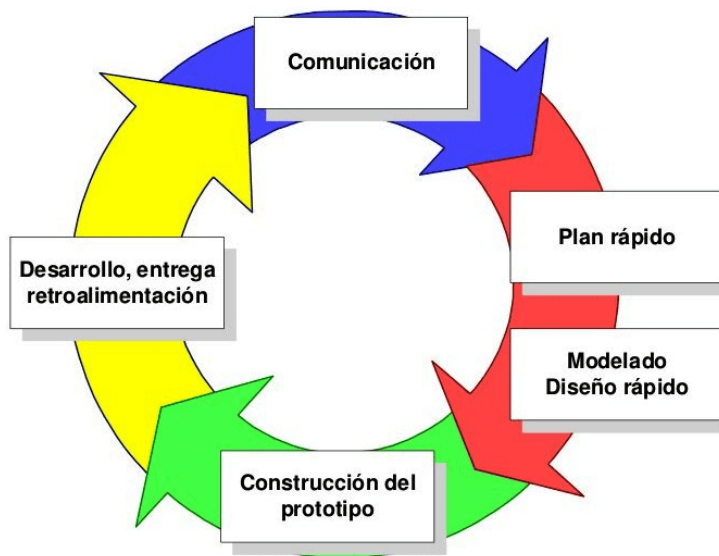
Los prototipos permiten validar los requerimientos capturados, identificar nuevos, evaluar la usabilidad y diseñar pantallas y reportes. El objetivo de esta metodología es conocer la reacción de los usuarios al interactuar con el prototipo del sistema. “Conforme los usuarios interactúan, se recogen sus reacciones mediante observación y entrevistas, con la finalidad de mejorar la especificación de requerimientos e implementarlos en el sistema” (Luis & Méndez, 2013).

Las etapas que conforman al prototipado son las siguientes:

- **Recolección y refinamiento de requisitos:** Etapa en la que se recopila y se detallan los requisitos del sistema. En donde el usuario y el desarrollador colaboran para refinar las funcionalidades del software.
- **Modelado, diseño rápido:** Creación de diseños preliminares, generando una representación visual que permita iniciar una retroalimentación.
- **Construcción del prototipo:** Prototipo inicial del sistema, refleja las características del sistema, permitiendo a los usuarios interactuar con él.
- **Desarrollo, evaluación del prototipo por el cliente:** Evaluación del prototipo por parte de los usuarios, brindando retroalimentación para el producto final.

- **Refinamiento del prototipo:** Se realizan mejoras y ajustes al sistema, basados en la retroalimentación del usuario.
- **Producto final:** Una vez que el sistema cumple con los requisitos y funcionalidades deseadas por el usuario, el software está listo para su respectiva implementación y despliegue.

*Ilustración 2. Ciclo de vida Prototipado.*



**Nota:** Tomado de *El prototipado como herramienta metodológica en la práctica del diseño [Fotografía]*, García Lara & Helle Pesot, 2024.

## **CAPÍTULO III: ANÁLISIS Y DISEÑO DEL SISTEMA**

### **3. Levantamiento de Requerimientos**

Se detallaron las etapas y los procesos que conlleva el proceso de recaudación, pero por detrás existen muchos más requisitos, que forman parte de este proceso englobando, ámbitos como seguridad, disponibilidad, conectividad, etc. Para ello, se requiere la implementación de múltiples funcionalidades, asegurando que el aplicativo cumpla con todos los requerimientos necesarios, para entregar un producto acorde con lo que los usuarios y socios necesitan.

Tomando en cuenta, que la información de las INSTITUCIONES FINANCIERAS DE EPS no es abierta al público, ya que posee información sensible. Para el presente desarrollo, con el objetivo de identificar los requerimientos necesarios se llevará a cabo una entrevista a un ingeniero con experiencia en soluciones informáticas para instituciones de economía popular y solidaria.

#### **3.1. Entrevista a Ingeniero en Soluciones Informáticas para INSTITUCIONES FINANCIERAS DE EPS**

- **¿Qué aspectos considera importantes para un sistema de gestión de cobranza?**
  - El aplicativo debe permitir acceder a los datos tanto las cuentas de ahorro y crédito de los clientes asociados.
  - Garantizar la seguridad de los datos a través de autenticaciones y encriptación.
  - Consolidar las transacciones realizadas por los agentes recaudadores, de manera visual.
  - Controlar accesos y operaciones realizadas por los agentes.
- **¿Cuáles son las necesidades que poseen las INSTITUCIONES FINANCIERAS**

## **DE EPS en cuanto a la gestión de cobranza o recaudación?**

- Los sistemas comerciales suelen ser muy costosos.
- La información que engloba la recaudación es un activo crítico, por ello, se manejan en bases de datos y servidores internos a la institución.
- **¿Cuál es el proceso ideal para gestionar recaudaciones?**
  - El sistema debe contar con un flujo de trabajo eficiente, registrando depósitos, incluyendo información en tiempo real acerca de la transacción realizada.
  - Módulos que faciliten depósitos a cuentas de ahorro, consulta de créditos y sus cuotas mediante a través de la tabla de amortización, emisión de reportes diarios y la integración de dispositivos Bluetooth para la emisión de comprobantes físicos.
- **¿Qué información es necesaria para interactuar con los clientes en el aplicativo?**
  - Número de identificación o cédula.
  - Información de cuentas de ahorro y crédito activos.
- **¿Qué funcionalidades de seguimiento considera importantes para el aplicativo?**
  - Generación de reportes diarios, con información detallada de la transacción.
  - Control y registro de uso de impresoras Bluetooth conectadas.
- **¿Cómo manejaría los depósitos a las cuentas de ahorro del cliente?**
  - El sistema debe habilitar la acreditación directa a las cuentas de ahorro seleccionadas por el cliente.
  - Finalmente, como confirmación de la transacción se imprimirá el comprobante correspondiente.
- **¿Qué aspectos considera importantes para garantizar la seguridad dentro del sistema?**

- Encriptación de contraseñas y datos sensibles mediante algoritmos de protección.
- Control de sesiones y cierre automático del sistema en ocasiones de inactividad.
- **¿Qué estrategia se debe implementar en el sistema para manejar excepciones o ingresar información errónea?**
  - Garantizar una retroalimentación adecuada al usuario, bloqueando del flujo del sistema hasta que se solventa el error.
- **¿Cuál es el proceso de autenticación ideal para los usuarios del sistema?**
  - Acceso al sistema mediante credenciales, como nombre de usuario, contraseña y número telefónico del empleado de la INSTITUCION FINANCIERA DE EPS.
- **¿Qué requisitos adicionales sugiere para implementar en el sistema?**
  - Diseño amigable e intuitivo.
  - La interfaz debe ser adaptable para cualquier dispositivo.
  - Funcionalidades de exploración de reportes.
  - Velocidad al momento de realizar depósitos, consultas, consolidar reportes y configuración de la impresora Bluetooth.
  - Disponibilidad total durante la jornada laboral.
  - Mantener informado al usuario de las acciones que está realizando.
  - Precisión en los rubros correspondientes a las cuotas del crédito.

## 3.2. Requerimientos Funcionales

*Tabla 1: Requerimientos funcionales.*

<b>Requerimientos Funcionales</b>	<b>Descripción</b>
<b>RF0 Ingreso al sistema</b>	El aplicativo contará con el ingreso de nombre de usuario, contraseña encriptada con MD5 y número telefónico para acceder.
<b>RF1 Depósitos</b>	El sistema permitirá realizar depósitos a las cuentas de los clientes de la INSTITUCION FINANCIERA DE EPS.
<b>RF2 Impresión de comprobantes</b>	El sistema permitirá la impresión de recibos de los depósitos realizados e imprimirlos mediante una impresora Bluetooth.
<b>RF3 Consulta de créditos</b>	El sistema permitirá consultar créditos aperturados por el cliente, mostrando detalles como el número de cuotas, fecha de vencimiento y valor a pagar.
<b>RF4 Consulta de cuotas</b>	El sistema permitirá visualizar la información de las cuotas del crédito, incluyendo intereses, días de mora, valor de la mora y el valor total de la cuota.
<b>RF5 Reportes de transacciones</b>	El sistema permitirá visualizar todos los depósitos realizados durante el día.
<b>RF6 Búsqueda de reportes</b>	El sistema permitirá buscar depósitos, a través del nombre del cliente.
<b>RF7 Impresión de reportes</b>	El sistema permitirá la impresión de recibos desde los reportes generados, utilizando la impresora Bluetooth.
<b>RF8 Conexión Bluetooth</b>	El sistema permitirá buscar impresoras Bluetooth cercanas, mostrando el estado

	de conexión, y ofrecer la funcionalidad de conectar y desconectar el aplicativo.
<b>RF9 Manejo de excepciones</b>	El sistema gestionará errores al ingresar información tanto del usuario como el cliente.
<b>RF10 Control de sesiones</b>	El sistema evitará el uso de credenciales en varios dispositivos al mismo tiempo, y cerrará la sesión por inactividad o cuando esta se encuentre en segundo plano.

*Nota: Elaboración propia.*

### 3.3. Requerimientos No Funcionales

*Tabla 2: Requerimientos no funcionales.*

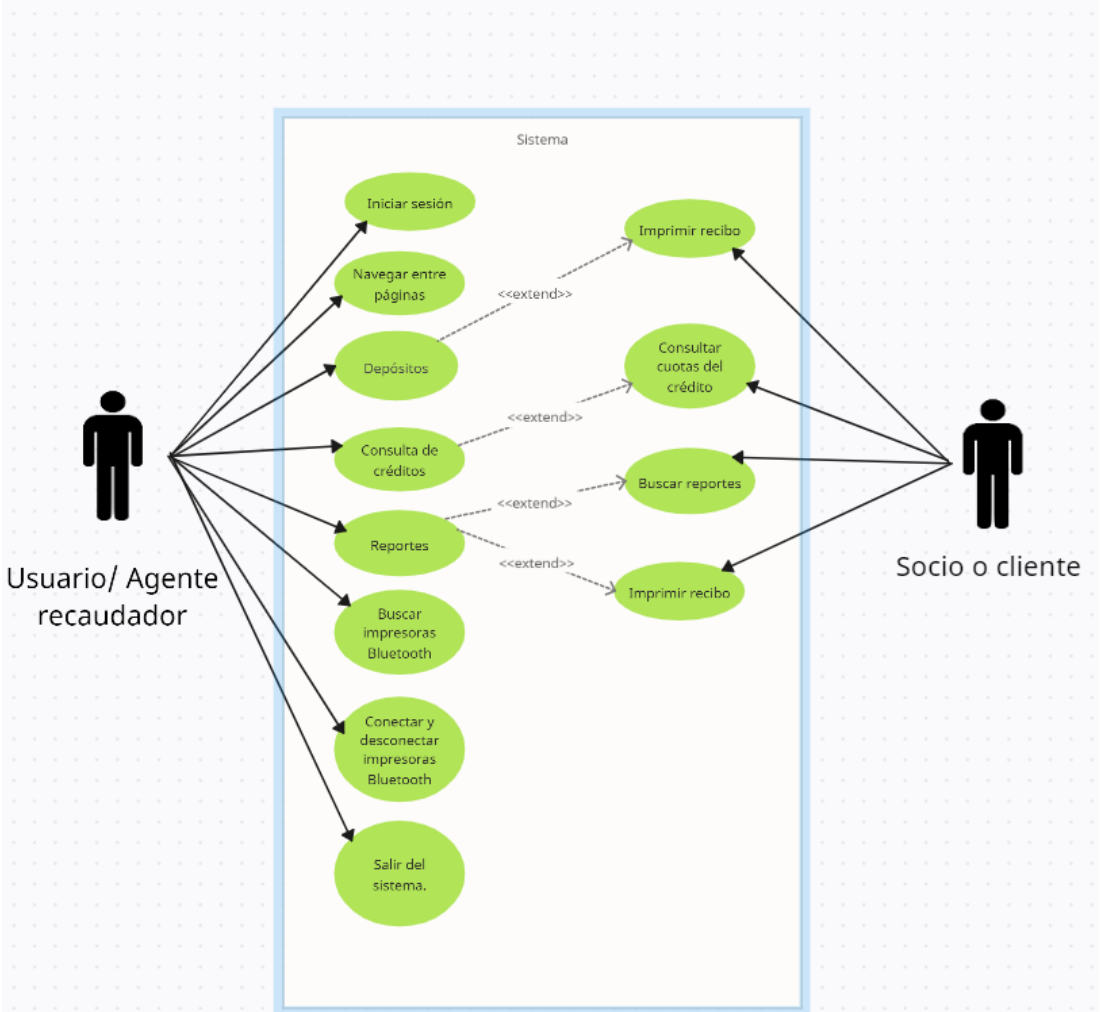
<b>Requerimientos No Funcionales</b>	<b>Descripción</b>
<b>RNF0 Precisión</b>	Garantizar la exactitud en los cálculos de las cuotas relacionadas con intereses, días de mora y valores adicionales.
<b>RNF1 Visibilidad</b>	El sistema debe informar al usuario en todo momento, acerca de las acciones que este concretando.
<b>RNF2 Disponibilidad</b>	El sistema debe estar disponible, para realizar operaciones durante la jornada laboral.
<b>RNF3 Adaptabilidad</b>	Interfaz intuitiva, amigable y adaptable para cualquier dispositivo.

*Nota: Elaboración propia.*

### 3.4. Diagrama de Caso de Uso

“Representación gráfica que describe las interacciones entre usuarios y el sistema” (Zapata & Carlos, 2005). El diagrama ilustra las funcionalidades principales que el aplicativo ofrece, mostrando como se relacionan con cada una de estas. En este caso, se destacan los usuarios principales (agente recaudador y cliente), participando de forma activa usando el aplicativo para concretar la tarea de gestión de cobranza.

**Ilustración 3.** Diagrama de caso de uso del aplicativo.



**Nota:** Elaboración propia.

### 3.5. Usuarios del Sistema

#### 3.5.1. Usuario/Agente Recaudador

Cumple la función de usuario principal del sistema, el mismo que es empleado de la INSTITUCIÓN FINANCIERA DE EPS, encargado de realizar la tarea de recaudación. Por lo tanto, el agente debe aproximarse al lugar donde se encuentre el socio que requiera el pago de un producto de crédito, que será concretado a través de sus cuentas de ahorro.

### **3.5.2. Socio o Cliente**

El socio se define como aquella persona afiliada a una INSTITUCIÓN FINANCIERA DE EPS, por lo tanto, como socio tiene permiso a aperturar cuentas de ahorro y créditos. El cliente tiene la opción de solicitar al agente recaudador realizar depósitos a las cuentas de ahorro y consultas para conocer el estado del crédito, así también, conocer el estado de las cuotas respectivas. Adicionalmente, el socio se le entrega el recibo de la transacción realizada.

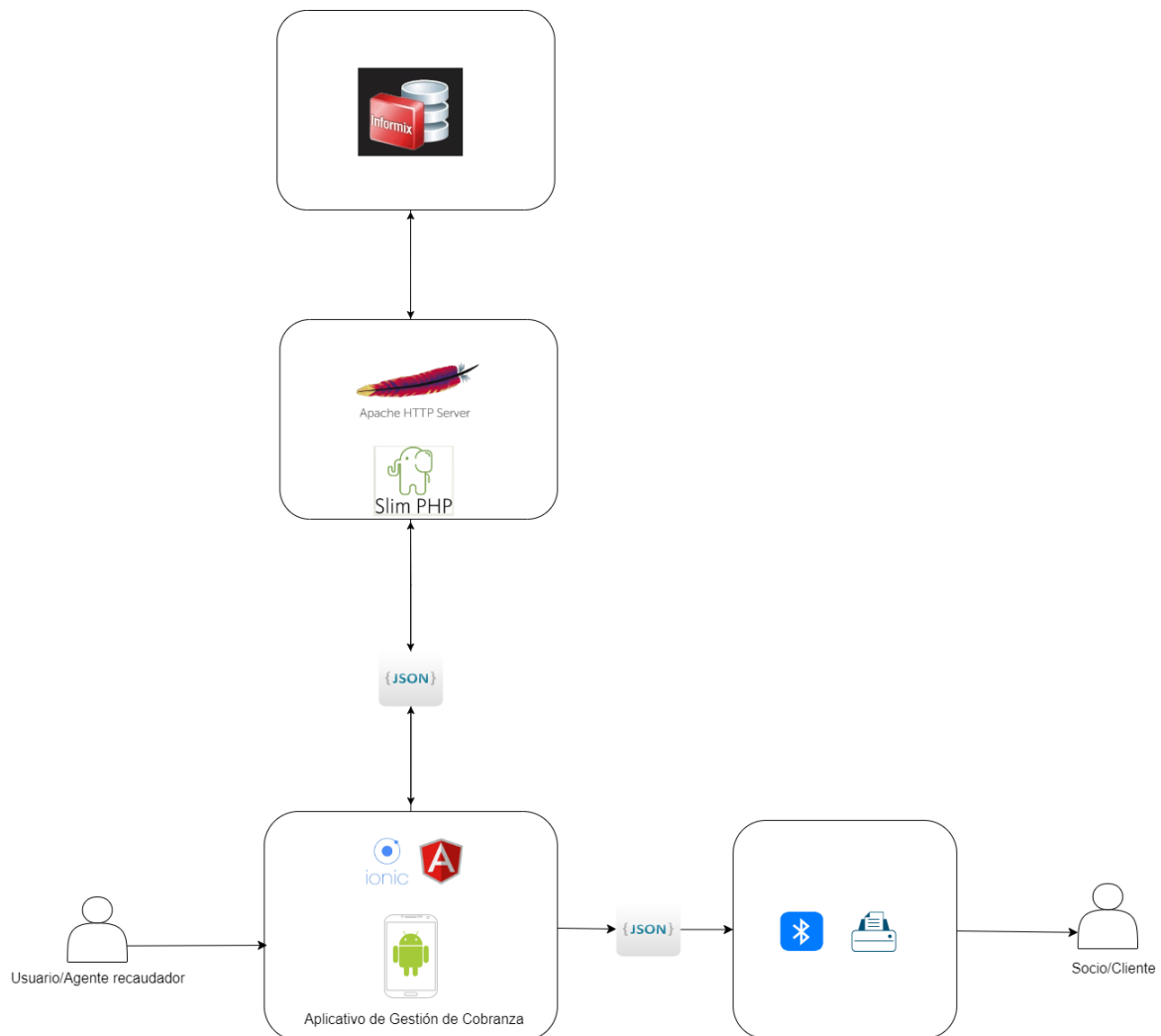
## **3.6. Arquitectura de la Aplicación**

“Se define como la estructura fundamental de un sistema de software, en donde se incluyen sus componentes principales, tecnologías y las interacciones entre ellos” (González Campo & Lozano Oviedo, 2020). Guiando en el desarrollo de la aplicación asegurando que cumpla con los requisitos funcionales y no funcionales.

### **3.6.1. Arquitectura Cliente-servidor**

“La presente arquitectura es un modelo de diseño en el que las tareas se dividen entre proveedores de servicios o servidores y los solicitantes en este caso los usuarios” (Nelson Cecchi et al., 2017). En el presente proyecto, este tipo de arquitectura se adapta a la propuesta del aplicativo, debido a las constantes solicitudes de información realizadas por el agente recaudador, con la finalidad de obtener respuestas desde el servidor y presentar la información acorde a la tarea de gestión de cobranza.

**Ilustración 4.** Arquitectura de la aplicación.



**Nota:** Elaboración propia.

### 3.7. Componentes Clave de la Arquitectura de la Aplicación

En el presente apartado se definen cuáles son los componentes más importantes, que conforman el proyecto, entre los cuales se destacan la capa de presentación, capa del negocio y la capa de datos. Dichos componentes son la estructura de nuestro aplicativo, permitiendo su funcionamiento.

#### 3.7.1. Capa de presentación (UI/UX)

Componente encargado de presentar la interfaz de usuario y la experiencia del usuario. Para el contexto del proyecto, la función principal de la capa de presentación es mostrar la información del socio afiliado a la INSTITUCIÓN FINANCIERA DE EPS de forma

intuitiva, de modo que el agente recaudador pueda manejar las interacciones e información del socio.

Para la presentación se hace uso de la herramienta o framework, de Angular/Ionic para el diseño de las pantallas principales.

- **Herramientas utilizadas:** Angular/Ionic.

### **3.7.2. Capa Lógica del Negocio**

Gestiona la lógica central del sistema, en la cual se definen las reglas, conceptos y procesos asociados al ejercicio de la recaudación. En este apartado, se garantiza la operatividad del sistema en base a las normativas de la INSTITUCIÓN FINANCIERA DE EPS y sus políticas regulatorias.

#### **3.7.2.1. Control de Inicio de Sesión**

La institución financiera asigna al agente recaudador credenciales únicas, en este caso nombre de usuario y contraseña encriptada con MD5. Las credenciales son exclusivas para el agente, garantizando que solo él pueda acceder al sistema. El control de inicio de sesión se encarga de regular que el agente no pueda iniciar sesión en varios dispositivos simultáneamente, forzando el acceso a un único dispositivo. Asimismo, el sistema valida las credenciales, verificando que coincidan con los datos registrados en la base de datos garantizando seguridad y evitar accesos no permitidos.

#### **3.7.2.2. Depósitos en Cuentas de Ahorro**

El socio afiliado a la INSTITUCIÓN FINANCIERA de EPS tiene la facultad de solicitar depósitos sin la necesidad de acercarse a las oficinas de la propia institución. Como se mencionó anteriormente, estos reciben la visita del agente recaudador para realizar el depósito a través del aplicativo. Al final de cada transacción se le entregara al cliente el comprobante correspondiente.

### **3.7.2.3. Monto Máximo y Mínimo de Depósitos en Cuentas de Ahorro**

El socio tiene permitido solamente hacer depósitos de un valor mínimo de un dólar, mientras que el monto máximo es de cinco mil dólares. Con la finalidad, de cumplir con regulaciones específicas, para prevenir actividades ilícitas, en donde la más importante es el lavado de dinero. Cualquier persona que desea hacer este tipo de transacciones tendrá que justificar la proveniencia del dinero y, por lo tanto, depositar el dinero en la sede física de la INSTITUCIÓN FINANCIERA DE EPS.

### **3.7.2.4. Consultas de Productos de Crédito**

El socio también puede poseer créditos, en las cuales deberá cumplir con el pago de las cuotas de este. Mediante el aplicativo, el agente recaudador le da a conocer información específica acerca del crédito y sus cuotas.

### **3.7.2.5. Validación de Número de Cédula**

La primera regla y el estándar principal, para acceder a la información de cualquier producto financiero que pertenezca al socio, es validar que el número de cédula correspondiente sea un número válido que lo identifica. Por ello, a lo largo de los años se llevó a cabo un algoritmo para cumplir con dicha regla:

- La cédula ecuatoriana se conforma por los primeros dígitos que corresponden a la provincia donde fue adquirida, por lo tanto, los primeros dígitos, no serán mayores a veinticuatro ni menores a cero.
- El tercer dígito es un número menor a seis (0, 1, 2, 3, 4, 5).
- Los siguientes hasta el noveno dígito, son números consecutivos.
- Los números en posición impar son multiplicados por dos, mientras que los números en posición par son multiplicados por uno.
- Si el resultado del producto entre los números es mayor o igual a diez se le resta nueve.

- Se suma todos los resultados del producto.
- A la suma de todos los resultados, se le resta la decena superior.
- Si el resultado de la resta coincide con el ultimo digito de la cédula de ciudadanía, la cédula es correcta y pertenece al estado ecuatoriano.

La cédula será ingresada y validada cada vez que se requiera realizar un depósito o una consulta de crédito, obligando a que el verdadero propietario de los productos financieros solicite dichas operaciones.

**Ilustración 5. Validación de cédula de ciudadanía.**

Se multiplica cada coeficiente con cada digito de la cedula sin tomar en cuenta el decimo digito

Si el resultado de algunas de las multiplicaciones es mayor o igual a 10 se le resta 9

COEFICIENTES	2	1	2	1	2	1	2	1	2		
SE MULTIPLICA	*	*	*	*	*	*	*	*	*		
CEDULA	1	7	1	0	0	3	4	0	6		
RESULTADO	2	7	2	0	0	3	8	0	12		
									-9		
SUMAR	=									25	

Se suman todos los resultados

Luego a la suma de todos los resultados le restamos de la decena superior

$30 - 25 = 5$

Nota: Si el resultado de la suma de todos los resultados menos la decena superior es igual a 10 el decimo digito es 0

Numero Cedula : 1710034065

Si todo salio bien el ultimo digito tiene que ser igual a el resultado de la resta

**Nota:** Tomado de *Cómo validar cédula y RUC en Ecuador [Fotografía] Medium, Suarez 2017.*

**3.7.2.6. Reportes**

El agente recaudador y la INSTITUCIÓN tendrán acceso a reportes de todas las transacciones que el agente recaudador ha realizado durante el día, generando un respaldo. Adicionalmente, se otorga la funcionalidad de imprimir el comprobante de una transacción determinada.

### **3.7.2.7. Impresiones Bluetooth**

El agente cumple con la tarea de entregar un recibo físico, mediante la impresión con una impresora conectada a través de Bluetooth. De la misma manera, se pueden realizar impresiones de las transacciones incluidas en los reportes.

### **3.7.3. Funciones Principales**

#### **3.7.3.1. Gestión de Flujo de Datos**

Se obtiene y procesa información de los depósitos, consultas y reportes mediante las solicitudes que se realizan desde el front-end, usando las APIs elaboradas con SLIM para cada uno de los módulos, las cuales se dirigen al servidor Apache y, finalmente hacia la base de datos Informix.

Para garantizar la seguridad en la obtención de datos, se implementa una autenticación mediante un token generado con MD5. El token es usado para validar cada solicitud realizada al servidor, enviándolo como parámetro en cada API. Esto permite que solo las solicitudes autorizadas accedan a la información.

#### **3.7.3.2. Respuesta y Validación de Resultados**

Los resultados se obtienen desde la base de datos mediante APIs alojadas en el servidor de Apache. La información se recupera en formato JSON, mediante la librería HttpClient para realizar solicitudes Http facilitando la comunicación entre agente recaudador y el servidor.

- **Herramientas utilizadas:**

- HTTP Apache.
- SLIM framework.
- Informix.
- Postman.

### **3.7.4. Módulos del Proyecto**

En este apartado se definen los módulos que conforman este proyecto, los cuales proporcionan funcionalidad a nuestro aplicativo móvil. El aplicativo cuenta con un número de cuatro módulos entre los cuales se encuentran: módulo de depósitos, consultas, reportes y configuración.

#### **3.7.4.1. Módulo de Depósitos**

En este módulo se lleva a cabo la tarea de depósitos a las cuentas de ahorro del socio. Una vez completada la transacción, el módulo permite imprimir el comprobante mediante el enlace entre el aplicativo móvil y la impresora Bluetooth.

#### **3.7.4.2. Módulo de Consultas**

Este módulo permite conocer información acerca de los créditos otorgados al socio afiliado. Además, muestra datos de las cuotas de crédito como fecha de vencimiento, estado (si la cuota ya fue pagada o sigue activa), el valor a pagar. También se incluyen detalles sobre intereses, seguro, interés de mora, días de mora y el valor total a pagar.

#### **3.7.4.3. Módulo de Reportes**

Módulo donde se registran todas las transacciones que realiza el agente recaudador durante el día. Además, permite imprimir el comprobante de las transacciones realizadas previamente.

#### **3.7.4.4. Módulo de Configuración**

El módulo permite verificar el estado de conexión de nuestro aplicativo con la impresora Bluetooth, mostrando si está conectada o desconectada. Además, facilita la búsqueda de dispositivos bluetooth cercanos, en caso de que se requiera conectarse a otra impresora. El agente recaudador puede administrar la conexión dependiendo si desea conectar o desconectar el dispositivo.

### 3.7.5. Capa de Datos

La capa de datos se encarga de gestionar el proceso de almacenamiento, procesamiento y manipulación de la información. Para este propósito, se utiliza el servicio de Informix como la base de datos del sistema.

#### 3.7.5.1. Base de Datos Relacional

“Una base de datos relacional se define como un sistema de almacenamiento que organiza la información en tablas relacionadas entre sí, permitiendo almacenar, gestionar y consultar grandes volúmenes de información” (Valverde et al., 2024). El uso de este modelo en el proyecto garantiza la integridad y consistencia de los datos, facilitando la ejecución de consultas complejas para gestionar las operaciones del sistema. A continuación, se presentan cada una de las tablas que conforman la base de datos, para su posterior modelamiento.

- **Herramientas utilizadas:**
  - Informix.
  - Power Designer.

**Tabla 3:** Tabla usuarios.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
usua_cod_usua	Identificador único de usuario.	Serial.
usua_cod_empl	Identificador único de usuario (empleado de la INSTITUCION FINANCIERA DE EPS).	Número entero.
usua_cod_perf	Código de perfil de cada usuario.	Cadena de texto
usua_nom_usua	Nombre de usuario.	Cadena de texto
usua_passwd	Contraseña de acceso.	Cadena de texto.
usua_fec_uac	Fecha de acceso.	Fecha.

usua_num_agen	Número identificador de agente recaudador.	Número entero.
usua_ban_usua	Bandera para conocer si el usuario se encuentra activo.	Número entero.
usua_imp_pred	Descripción de la impresora predeterminada.	Cadena de texto.

**Tabla 4:** Empleados vinculados con usuarios.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
empl_cod_empl	Identificador único de empleados.	Serial.
empl_cod_clie	Identificador único de empleado vinculado con un usuario.	Número entero.
empl_fec_ingr	Fecha de ingreso al sistema.	Fecha.
empl_fec_sali	Fecha de salida del sistema.	Fecha.
empl_sue_empl	Sueldo del empleado.	Decimal.
empl_num_iess	Número de afiliación al IEES.	Cadena de texto.
empl_obs_empl	Observaciones del empleado.	Cadena de texto.
empl_cod_ofic	Código de oficina.	Número entero.
empl_cod_rlab	Relación laboral.	Número entero.
empl_tit_empl	Título profesional.	Cadena de texto.
empl_cod_taud	Tipo de auditor.	Número entero.
empl_ban_ases	Asesor de crédito.	Número entero.
empl_tfp_pago	Tipo de pago.	Número entero.
empl_ban_bgal	Beneficiario.	Número entero.
empl_ban_enca	Si posee enfermedad catastrófica.	Número entero.
empl_cod_disc	Tipo de discapacidad.	Número entero.
empl_por_disc	Porcentaje de discapacidad.	Número decimal.

empl_tid_disc	Descripción de discapacidad.	Cadena de texto.
empl_ide_disc	Identificación de familiar por discapacidad.	Cadena de texto.
empl_cod_tcem	Tiempo contrato del empleado.	Número entero.

**Tabla 5:** Almacenamiento de acceso de usuario cuando inicia sesión.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
acce_cod_bbdd	Código de acceso a la base de datos.	Número entero.
acce_cod_usua	Identificador único de usuario.	Número entero.

**Tabla 6:** Datos de clientes.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
clie_cod_clie	Identificador único de clientes.	Serial.
clie_cod_tcli	Tipo de cliente.	Número entero.
clie_num_clie	Numero de cliente.	Número entero.
clie_cod_ofic	Código oficina.	Número entero.
clie_cod_tide	Tipo de identificación.	Cadena de texto.
clie_ide_clie	Número de identificación.	Cadena de texto.
clie_ape_clie	Apellido de cliente.	Cadena de texto.
clie_nom_clie	Nombre de cliente.	Cadena de texto.
clie_fec_nac	Fecha de nacimiento.	Fecha.
clie_cod_sect	Código de sector.	Número entero.
clie_dir_domi	Dirección domicilio.	Cadena de texto.
clie_fec_uac	Fecha de última actualización de datos.	Fecha.
clie_fec_ingr	Fecha de ingreso.	Fecha.
clie_fec_sali	Fecha de salida.	Fecha.
clie_ema_clie	Correo electrónico de cliente.	Cadena de texto.

clie_nat_juri	Cliente natural o jurídico.	Número entero.
clie_est_clie	Estado de cliente.	Número entero.
clie_rep_lega	Nombre de representante legal.	Cadena de texto.
clie_ide_repr	Identificación de representante legal.	Cadena de texto.
clie_tide_repr	Tipo de identificación del representante legal.	Cadena de texto.
clie_cod_pais	Código de país.	Número entero.
clie_cod_usua	Usuario que registro al cliente.	Número entero.
clie_ref_dire	Referencia de dirección.	Cadena de texto.
clie_cod_cocu	Tipo de ocupación.	Número entero.
clie_cod_disc	Estado de discapacidad.	Número entero.
clie_cod_tviv	Tipo de vivienda.	Número entero.
clie_val_vivi	Valor de la vivienda.	Número decimal.
clie_num_carg	Cargas familiares.	Número entero.
clie_cod_tcsr	Tipo de socio representante.	Número entero.
clie_cod_auid	Autoidentificación.	Número entero.
clie_cod_prof	Código de profesión.	Número entero.
clie_ban_peps	Persona políticamente expuesta.	Número entero.
clie_cod_tvin	Tipo de vinculación.	Número entero.
clie_ban_grup	Bandera de pertenencia o no a un grupo.	Número entero.
clie_cod_grup	Código de grupo.	Número entero.
clie_cod_tres	Tipo de residencia.	Número entero.
clie_pai_resi	País de residencia.	Número entero.
clie_hue_dact	Huella dactilar.	Cadena de texto.
clie_ban_pdpe	Consentimiento de datos personales.	Número entero.

**Tabla 7:** Datos de teléfono de clientes.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
tele_cod_tele	Identificador único de teléfono de cliente.	Serial.
tele_cod_clie	Código de cliente.	Número entero.
tele_cod_ttel	Tipo de teléfono.	Número entero.
tele_cod_utel	Operadora.	Número entero.
tele_num_tele	Número de teléfono.	Cadena de texto.
tele_ban_prin	Teléfono principal.	Número entero.

**Tabla 8:** Datos de las cuentas que posee un cliente.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
dpvi_cod_dpvi	Identificador único de la cuenta de un cliente.	Serial.
dpvi_cod_tcdv	Código de tipo de cuenta.	Número entero.
dpvi_cod_tfir	Tipo de firma.	Número entero.
dpvi_num_firm	Número de firmas.	Número entero.
dpvi_cod_eacd	Estado de la cuenta.	Número entero.
dpvi_cod_ofic	Código de oficina.	Número entero.
dpvi_num_dpvi	Número de cuenta.	Cadena de texto.
dpvi_cod_clie	Código del cliente al que le pertenece la cuenta.	Número entero.
dpvi_fec_inic	Fecha de apertura de cuenta.	Fecha.
dpvi_ofi_cier	Oficina en la que se cierra la cuenta.	Número entero.
dpvi_fec_cier	Fecha de cierre de la cuenta.	Fecha.
dpvi_sal_disp	Saldo disponible.	Número decimal.
dpvi_sal_cont	Saldo contable.	Número decimal.
dpvi_aho_mini	Ahorro mínimo.	Número decimal.
dpvi_fec_acti	Fecha de activación de la cuenta.	Fecha.

dpvi_fec_venc	Fecha de vencimiento de la cuenta.	Fecha.
dpvi_dia_plaz	Plazo en días de la cuenta.	Número entero.
dpvi_tas_dpvi	Tasa de la cuenta.	Número decimal.
dpvi_dpv_debi	Cuenta a debitar el ahorro programado.	Número entero.
dpvi_usu_inac	Código de usuario que inactivo la cuenta.	Número entero.
dpvi_cod_intv	Intervalo de pago de interés.	Cadena de texto.
dpvi_cod_ccdv	Código de detalle cierre de cuenta.	Número entero.
dpvi_cod_sect	Código de sector.	Número entero.
dpvi_nom_dpvi	Nombre completo del socio.	Cadena de texto.
dpvi_ban_prin	Bandera de permiso de impresión.	Número entero.

**Tabla 9:** Datos de tipos de cuentas que puede tener una cuenta.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
tcdv_cod_tcdv	Identificador único de tipo de producto o cuenta.	Serial.
tcdv_cod_tcap	Código de descripción de tipo de producto.	Número entero.
tcdv_cod_mone	Código de moneda con la que se maneja el producto.	Número entero.
tcdv_cod_fcinv	Código de forma de pago.	Número entero.
tcdv_cod_ttpf	Tasa.	Número entero.
tcdv_fec_crea	Fecha de creación del producto.	Fecha.
tcdv_fec_udev	Fecha de última provisión.	Fecha.
tcdv_cod_atcd	Código de producto.	Número entero.
tcdv_des_tcdv	Nombre de producto.	Cadena de texto.
tcdv_aho_mini	Ahorro mínimo del producto.	Número decimal.

tcdv_cco_acti	Cuenta contable activa.	Número entero.
tcdv_cco_prov	Cuenta contable de la provisión.	Número entero.
tcdv_cco_gast	Cuenta contable de gasto.	Número entero.
tcdv_cco_inac	Cuenta contable inactiva.	Número entero.
tcdv_cco_dxcf	Cuenta contable de depósitos por confirmar.	Número entero.
tcdv_cal_dias	Calculo días.	Número entero.
tcdv_fec_upgi	Ultima acreditación de intereses.	Fecha.
tcdv_ctr_trab	Número de dígitos de cuenta.	Número entero.
tcdv_val_inic	Valor inicial del producto.	Número decimal.
tcdv_bse_inac	Estado de autorización de inactivación de cuentas.	Número entero.
tcdv_est_tcdv	Estado del tipo de producto.	Número entero.
tcdv_usu_modi	Usuario que modifiko el producto.	Número entero.
tcdv_fec_modi	Fecha de modificación de producto.	Fecha.
tcdv_cod_tpdv	Tipo de depósito.	Número entero.

**Tabla 10:** Datos de movimientos de una cuenta.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
mcdv_cod_mcdv	Identificador único de movimientos de cuentas.	Serial.
mcdv_num_mcdv	Número de la papeleta.	Cadena de texto.
mcdv_cod_dpvi	Código del producto.	Número entero.
mcdv_cod_caja	Código de caja.	Cadena de texto.
mcdv_fec_mcdv	Fecha de movimiento.	Fecha.
mcdv_ctr_impr	Bandera de impresión de movimiento.	Número de entero.
mcdv_cod_pape	Código de papeleta contable.	Número entero.

mcdv_det_mcdv	Detalle del movimiento.	Cadena de texto.
mcdv_val_mcdv	Valor del movimiento.	Número decimal.
mcdv_ban_deta	Detalle de efectivo.	Número entero.
mcdv_num_cheq	Número de cheque.	Número entero.
mcdv_cod_tran	Código de transacción.	Número entero.
mcdv_cod_orec	Código de orden de emisión.	Número entero.
mcdv_cod_tndd	Tipo de movimiento (nota de crédito, nota de débito).	Número entero.
mcdv_cod_ofic	Código de oficina.	Número entero.
mcdv_ban_iva	Bandera de generación de IVA.	Número entero.

**Tabla 11:** Detalles y forma de pago de movimiento en una cuenta.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
ddte_cod_mcdv	Código del movimiento de cuenta.	Número entero.
ddte_cod_tfpq	Forma de pago.	Número entero.
ddte_val_ddte	Valor del pago.	Número decimal.
ddte_efe_dias	Días de efectivización.	Número entero.

**Tabla 12:** Detalles de depositante de terceros en una cuenta.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
ddep_cod_ddep	Identificador único de detalles de depositantes de terceros en una cuenta.	Serial.
ddep_cod_mcdv	Código del movimiento de cuenta.	Número entero.
ddep_fec_ddep	Fecha de depósito.	Fecha.
ddep_val_ddep	Valor del depósito.	Número decimal.
ddep_nom_ddep	Nombre del depositante.	Cadena de texto.
ddep_num_docu	Número de documento del depositante.	Cadena de texto.
ddep_obs_ddep	Observación del depósito.	Cadena de texto.

**Tabla 13:** Detalle de efectivo en movimientos de cuentas.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
deef_cod_deef	Identificador único de detalles de efectivo	Serial.
deef_cod_modu	Código de modulo donde se genera la transacción.	Número entero.
deef_cod_movi	Código de movimiento.	Número entero.
deef_cod_mone	Código de moneda.	Número entero.
deef_cod_caja	Código de caja.	Cadena de texto.
deef_fec_deef	Fecha de movimiento.	Fecha.
deef_cod_tasi	Tipo de transacción.	Cadena de texto.
deef_cod_pape	Código de papeleta.	Número entero.
deef_cod_ofic	Código de oficina.	Número entero.

**Tabla 14:** Denominaciones de efectivo.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
camb_cod_camb	Identificador único de denominación de efectivo.	Serial.
camb_cod_tdin	Bandera de monedas o billetes.	Número entero.
camb_val_camb	Denominación de billetes y monedas.	Número decimal.
camb_val_seri	Bandera de registro de serial de billetes.	Número entero.

**Tabla 15:** Detalle de denominación de efectivo.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
defe_cod_defe	Identificador único de detalle de denominación de efectivo.	Serial.
defe_cod_deef	Código del detalle de movimiento de efectivo.	Número entero.
defe_cod_camb	Código de la denominación de efectivo.	Número entero.

defe_num_defe	Número monedas o billetes.	Número entero.
---------------	----------------------------	----------------

**Tabla 16:** Créditos que posee un cliente.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
cred_num_cred	Identificador único de créditos de un cliente.	Serial.
cred_num_soli	Número de solicitud.	Cadena de texto.
cred_num_paga	Número de pagare.	Cadena de texto.
cred_cod_pape	Código de papeleta contable.	Número entero.
cred_fec_inic	Fecha de creación de crédito.	Fecha.
cred_tip_ccon	Cuenta contable.	Cadena de texto.
cred_cap_cred	Capital concedido.	Número decimal.
cred_tas_cred	Tasa de crédito.	Número decimal.
cred_fec_venc	Fecha de vencimiento de crédito.	Fecha.
cred_cod_intv	Intervalo de crédito.	Cadena de texto.
cred_cod_ecre	Estado de crédito.	Número entero.
cred_cod_dtcr	Destino financiero.	Número entero.
cred_cod_ofi	Código de oficina.	Número entero.
cred_cod_clie	Código de cliente.	Número entero.

**Tabla 17:** Detalles de los créditos que posee un cliente.

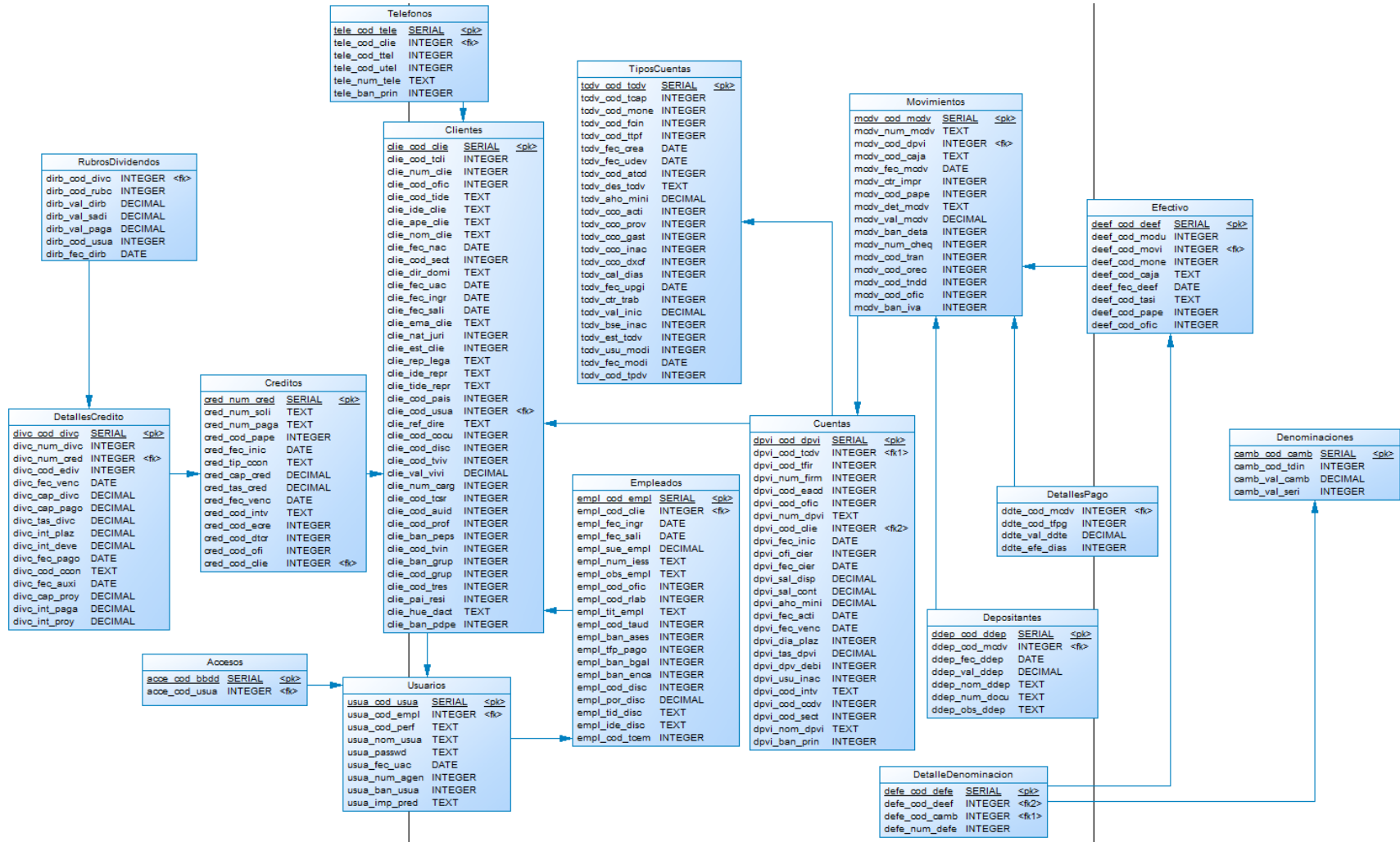
<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
divc_cod_divc	Identificador único de detalles de crédito.	Serial.
divc_num_divc	Número de dividendo.	Número entero.
divc_num_cred	Número de crédito.	Número entero.
divc_cod_ediv	Estado del dividendo.	Número entero.
divc_fec_venc	Fecha de vencimiento.	Fecha.
divc_cap_divc	Capital activo.	Número decimal.
divc_cap_pago	Capital pagado.	Número decimal.
divc_tas_divc	Tasa de interés.	Número decimal.

divc_int_plaz	Interés activo.	Número decimal.
divc_int_deve	Interés devengado.	Número decimal.
divc_fec_pago	Fecha de pago.	Fecha.
divc_cod_ccon	Cuenta contable del dividendo.	Cadena de texto.
divc_fec_auxi	Fecha auxiliar.	Fecha.
divc_cap_proy	Capital inicial proyectado.	Número decimal.
divc_int_paga	Interés pagado.	Número decimal.
divc_int_proy	Interés inicial proyectado.	Número decimal.

**Tabla 18:** Detalle de rubros que posee un dividendo.

<b>Campo</b>	<b>Descripción</b>	<b>Tipo</b>
dirb_cod_divc	Identificador único de dividendo.	Número entero.
dirb_cod_rubc	Código de rubro.	Número entero.
dirb_val_dirb	Valor activo del rubro.	Número decimal.
dirb_val_sadi	Valor original del rubro.	Número decimal.
dirb_val_paga	Valor pagado.	Número decimal.
dirb_cod_usua	Código de usuario.	Número entero.
dirb_fec_dirb	Fecha carga rubro.	Fecha.

**Ilustración 6.** Base de datos general.



**Nota:** Modelo físico de la base de datos.

## CAPITULO IV: DESARROLLO E IMPLEMENTACIÓN

### 4. Desarrollo de Back-end

En este apartado se detalla la estructura y el desarrollo del back-end junto con la prueba de las APIs para la entrega de la información almacenada dentro de la base de datos de Informix, para posteriormente comenzar con la implementación de funcionalidades dentro del sistema o aplicativo.

#### 4.1. Arquitectura del Back-end

El back-end fue diseñado usando el patrón MVC (Modelo Vista Controlador), permitiendo separar la lógica del negocio, interacciones con la base de datos y la administración de solicitudes HTTP. El backend se estructura de la siguiente forma:

- **Routes:** Se definen las rutas de APIS y acciones asociadas.
- **Controllers:** Se establece la lógica del negocio, controlando la interacción entre rutas y modelos.
- **Models:** Maneja las consultas y operaciones con la base de datos.
- **Middlewares:** Seguridad, autenticación y manejo de excepciones.
- **Public:** Contiene el archivo index.php, como punto de entrada del sistema.

##### 4.1.1. Desarrollo de APIs

Las APIs se definen como mecanismos que permiten a los componentes de un software comunicarse entre sí mediante un conjunto de definiciones y protocolos. Son consideradas como un contrato de servicio entre dos aplicaciones, especificando cómo se comunican entre si mediante solicitudes y respuestas.

###### 4.1.1.1. Rutas Principales

- **Login (POST /api/public/movil/login):** Autenticar al usuario mediante su nombre de usuario, contraseña y número de teléfono.
- **Logout (POST /api/public/movil/logon):** Cierra la sesión del usuario, mediante él envió de un identificador único de inicio de sesión.

- **Consulta socio (POST /api/public/movil/consultasocio):** Accede a la información del socio afiliado a la institución financiera, mediante el ingreso de su número de cédula.
- **Escoge cuenta (POST /api/public/movil/escogecuenta):** Accede a las cuentas de ahorro que posee un cliente, a través del identificador único del cliente.
- **Deposito (POST /api/public/movil/deposito):** Acredita el dinero dentro de las cuentas de ahorro, utilizando el identificador único de la cuenta seleccionada y el valor que se ingresado en el aplicativo.
- **Consulta crédito (POST /api/public/movil/consultacreditos):** Accede a los productos de crédito que posee un cliente, mediante el identificador único correspondiente al cliente.
- **Consulta tabla (POST /api/public/movil/consultatabla):** Accede a la información del crédito, enviando el identificador del crédito seleccionado, mostrando todas sus cuotas.
- **Consulta cuota (POST /api/public/movil/consultacuota):** Accede a la información de la cuota de crédito, mediante el identificador de crédito y el número de la cuota seleccionada. Sin embargo, solo se puede acceder a las cuotas que aún permanecen activas.
- **Reportes (POST /api/public/movil/reportes/movimientos):** Accede a las transacciones realizadas durante el día, enviando como parámetro la fecha actual.
- **Recibo (POST /api/public/movil/recibo):** Permite el acceso a los datos de las transacciones, para ser impresas enviando un parámetro de autorización para concretar la impresión del recibo.

## 4.1.2. Manejo de Errores

Para manejar los errores, se realizó de forma centralizada a través de bloques “try-catch”, capturando los errores durante la ejecución de cada método correspondiente a cada API. Asegurando que cualquier excepción sea interceptada y procesada adecuadamente, generando las respuestas.

## 4.1.3. Pruebas de Back-end

Comprobando que el desarrollo del backend cumple con los resultados esperados, se realizaron pruebas de envío de parámetros, mediante el uso de la herramienta Postman. Verificando que el back-end cumpla con la devolución de resultados y respuestas, en formato JSON y con la autenticación en base a la generación de tokens con MD5.

### *Ilustración 7. Prueba de APIS con Postman.*

The screenshot shows a Postman interface for a REST client. The URL is `www.sitetrior.com/inmaculada/public/movil/consultacreditos`. The request body is set to 'form-data' and contains the following parameters:

Key	Value	Description
<input checked="" type="checkbox"/> idsesion	Text 3	
<input checked="" type="checkbox"/> fechaservidor	Text 2024-08-14 18:00:00	
<input checked="" type="checkbox"/> token	Text 1dc807dfec7cbcf950f77003453b805	
<input checked="" type="checkbox"/> codigo	Text 21	

The response status is 200 OK, Time: 10.67 s, Size: 574 B. The response body is shown in the 'Raw' tab as a JSON object:

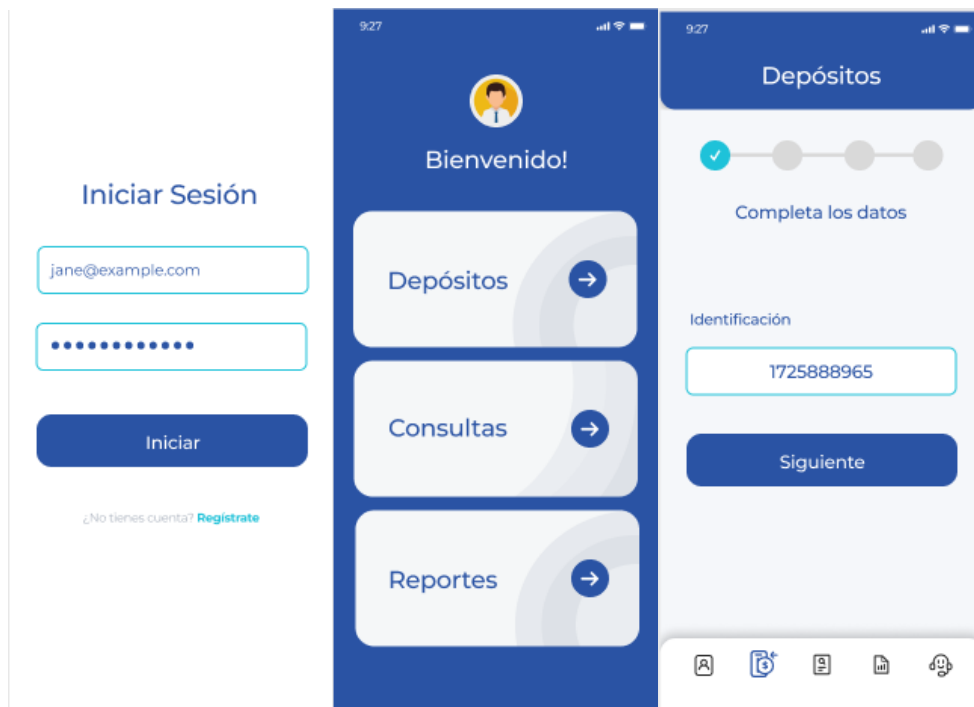
```
1 {"result":{"idsesion":"3","fechaservidor":"2024-8-14 12:23:41","listado":[{"idcredito":"1810","descripcion":"CONSUMO  
2 ORDINARIO","saldo":"607.64","fechainicio":"2022-07-19","capital":"5600.00"}]},{"codigoResultado":"000","mensajeResultado":"Datos","token":"1dc807dfec7cbcf950f77003453b805"}
```

**Nota:** Comprobación de funcionamiento de API Consulta crédito.

## 4.2. Prototipo de la Interfaz de Usuario

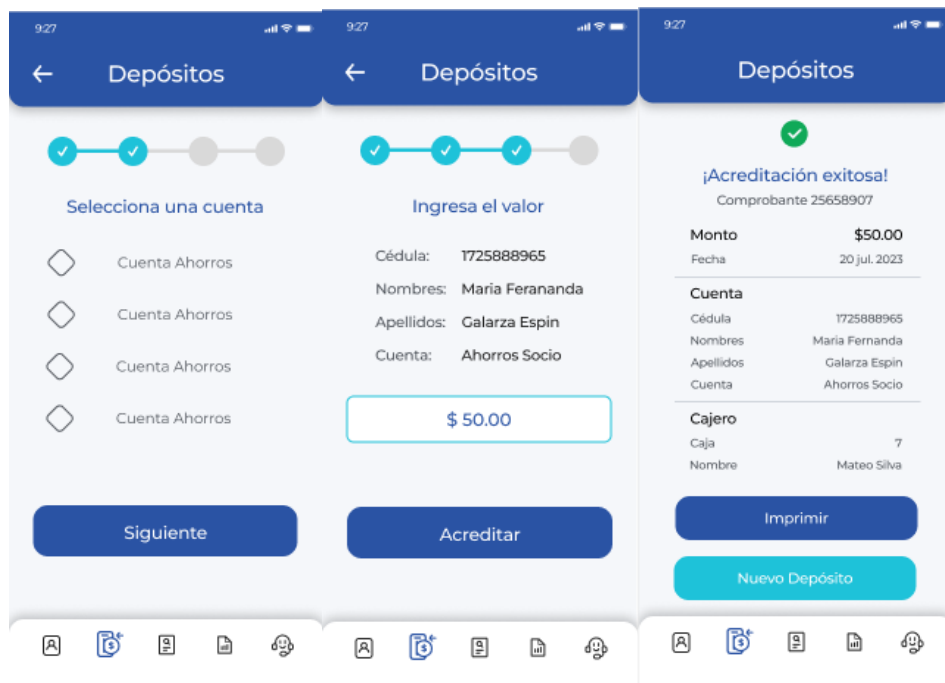
Para el diseño preliminar de la interfaz de usuario se hizo uso de la herramienta Figma, que se destaca por ser una gran herramienta de edición de gráficos y es una de las plataformas más utilizadas para diseño UX/UI.

**Ilustración 8.** Diseño del prototipo de la interfaz de usuario 1.



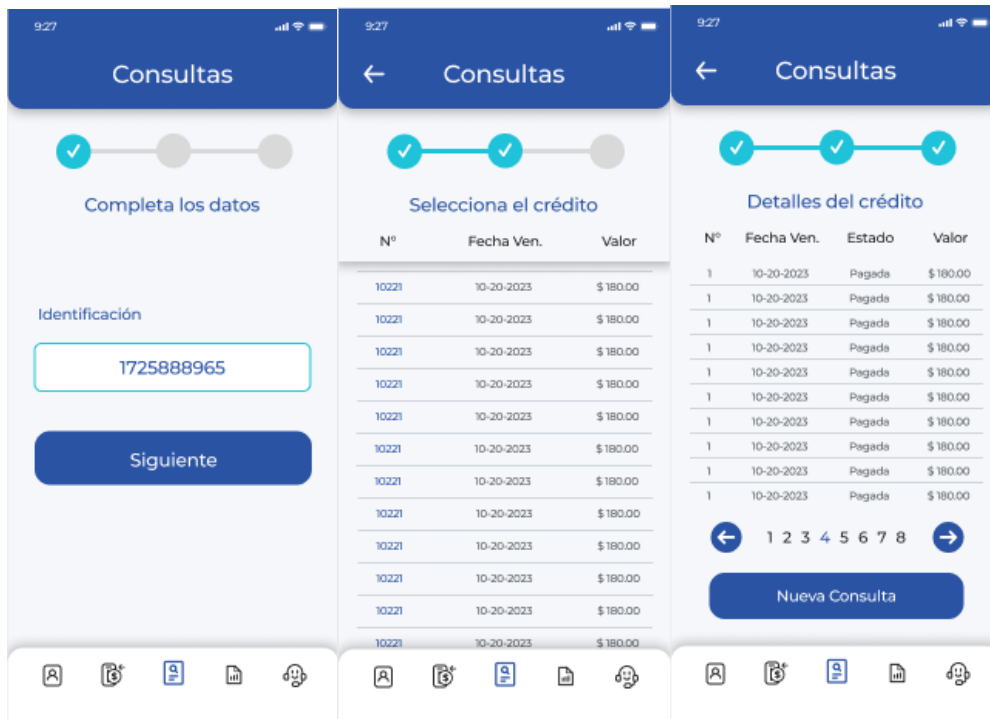
**Nota:** El gráfico representa una de las propuestas en un diseño preliminar que se usó para el desarrollo de esta aplicación.

**Ilustración 9.** Diseño del prototipo de la interfaz de usuario 2.



**Nota:** El gráfico representa una de las propuestas en un diseño preliminar que se usó para el desarrollo de esta aplicación.

**Ilustración 10.** Diseño del prototipo de la interfaz de usuario 3.



**Nota:** El gráfico representa una de las propuestas en un diseño preliminar que se usó para el desarrollo de esta aplicación.

**Ilustración 11.** Diseño del prototipo de la interfaz de usuario 4.



**Nota:** El gráfico representa una de las propuestas en un diseño preliminar que se usó para el desarrollo de esta aplicación.

Estos borradores representan los primeros prototipos de las pantallas del aplicativo, incluyendo sus módulos y subprocessos. El diseño del aplicativo se ha desarrollado con la finalidad de lograr una estética visual clara, para el agente recaudador. Utilizando elementos propios del framework Ionic que incluyen iconos, cartas, marcadores y ventanas modales. El prototipo de la interfaz el diseño fue reestructurándose, conforme la retroalimentación del ingeniero entrevistado anteriormente y las necesidades del usuario principal.

#### 4.2.1. Refinamiento del Prototipo

En base a la retroalimentación brindada por parte del ingeniero entrevistado en el levantamiento de requerimientos, se reestructuró el diseño para cumplir con los requerimientos adicionales mencionados, los cambios se implementaron durante la fase de desarrollo de la interfaz. A continuación, se describen dichos requerimientos:

- La página de inicio debe contar con el botón de acceso al módulo de configuración y el botón de salida del sistema.
- Implementar la pantalla de selección del crédito y de consulta de cuotas, ya que no se incluyó en el prototipo inicial.
- Implementación del módulo de configuración, debido a que no estaba presente en el prototipo.

#### 4.3. Funcionalidades del Sistema

En este apartado se detallan las funcionalidades que debe cumplir el sistema, incluyendo las funcionalidades realizadas por el agente recaudador. Estas funcionalidades son detalladas a continuación en las siguientes tablas:

**Tabla 19:** *Funcionalidades ejecutadas por el sistema.*

Funcionalidad	Nombre	Descripción
FS1	Validación de formularios.	El sistema debe permitir el accionar de botones y demás acciones, solamente si el usuario ingresa la información correcta, caso

		contrario, se da paso a la retroalimentación para el usuario.
FS2	Validación de credenciales e información de clientes.	El sistema realiza validaciones de las credenciales del agente y número de identificación del cliente, antes de mandarlas como parámetros, a las APIS para recibir la información de créditos, consultas y reportes.
FS3	Encriptación de credenciales con MD5.	El sistema encripta la contraseña del agente recaudador con MD5, para verificar que coincida con la contraseña presente en la base de datos.
FS4	Visualización de la información de las cuentas de ahorro.	El sistema permite visualizar la información de todas las cuentas de ahorro, que un cliente pueda tener activas.
FS5	Acreditación de dinero en cuentas de ahorro.	La aplicación se encarga de acreditar el dinero deseado por el socio, a la cuenta de ahorro que el desee.
FS6	Consulta de créditos.	El sistema accede a la información de los productos de crédito del socio afiliado. Solamente, si este posee créditos activos.
FS7	Visualización de la información del crédito.	El sistema permite visualizar las cuotas del crédito, dentro de una tabla,

		en donde se muestran el número de cuotas, su fecha de vencimiento y su estado (si esta activa o pagada).
FS8	Visualización de la información de la cuota de crédito.	El sistema permite visualizar a detalle dentro de una tabla, la información adicional de las cuotas que se mantienen activas.
FS9	Visualización de reportes.	El sistema permite visualizar los reportes, de las transacciones realizadas durante el día.
FS10	Búsqueda de reportes.	El sistema permite buscar los reportes realizados, mediante la digitación del nombre del cliente o socio.
FS11	Búsqueda de dispositivos Bluetooth.	El sistema permite buscar dispositivos Bluetooth.
FS12	Visualización de estado de conexión con el dispositivo Bluetooth	El sistema permite visualizar la conexión entre el aplicativo y el dispositivo Bluetooth.
FS13	Impresión de recibos mediante impresora Bluetooth.	Una vez finalizado el depósito, el sistema imprime un recibo o comprobante mediante la impresora Bluetooth. De la misma forma se pueden realizar impresiones desde el módulo de reportes, de ciertas transacciones realizadas durante el día.
FS14	Navegación entre módulos.	Navegación entre módulos, mediante el menú colocado

		en la parte inferior de cada módulo.
FS15	Manejo de excepciones.	Cuando existan problemas de ingreso de datos, el sistema dará retroalimentación al usuario informando el error y bloqueando el flujo hasta que el error desaparezca.
FS16	Sesión expirada.	Se cuenta el tiempo de inactividad dentro del sistema una vez se cumpla el tiempo, la sesión expira cerrando el sistema. Adicionalmente, la sesión se cierra, cuando el aplicativo se encuentra en segundo plano o se fuerce su cierre.

**Tabla 20:** Funcionalidades ejecutadas por el agente recaudador.

<b>Funcionalidad</b>	<b>Nombre</b>	<b>Descripción</b>
FA1	Seleccionar cuentas de ahorro.	Anteriormente mencionado, el agente recaudador puede seleccionar la cuenta de ahorro que el usuario decida, para realizar depósitos.
FA2	Confirmación de acreditación.	El agente recaudador tiene la facultad de confirmar la acreditación de saldo o efectivo o cancelarla respectivamente.
FA3	Selección de créditos.	El agente recaudador selecciona el crédito, del

		que se desea obtener información.
FA4	Conexión con el dispositivo Bluetooth.	Mediante una lista de dispositivos, el agente recaudador selecciona la impresora con la que se imprimirán los comprobantes.
FA5	Desconexión del dispositivo Bluetooth.	El agente recaudador puede desconectar el aplicativo mediante un botón.
FA6	Cerrar sesión.	El agente recaudador, puede cerrar la sesión desde la página de inicio mediante el botón de salida.


### **3.8. Desarrollo de Interfaces**

En esta etapa se trabaja en la interfaz definitiva que tendrá nuestro aplicativo, una vez familiarizados con los componentes que ofrece el framework de Angular/Ionic. Otorgando al agente recaudador, pantallas que mejor se adecuen para presentar e ingresar la información correspondiente, cumpliendo con la tarea de gestión de recaudación.

#### **4.4.1. Pantalla de Inicio de Sesión**

Al iniciar el aplicativo, se mostrará la pantalla de inicio de sesión, donde se da paso al ingreso de credenciales para acceder al aplicativo y utilizar el sistema de forma definitiva.

**Ilustración 12.** Pantalla de inicio de sesión.



"Administración Financiera Cooperativa"

Iniciar Sesión

Usuario

---

Contraseña

---

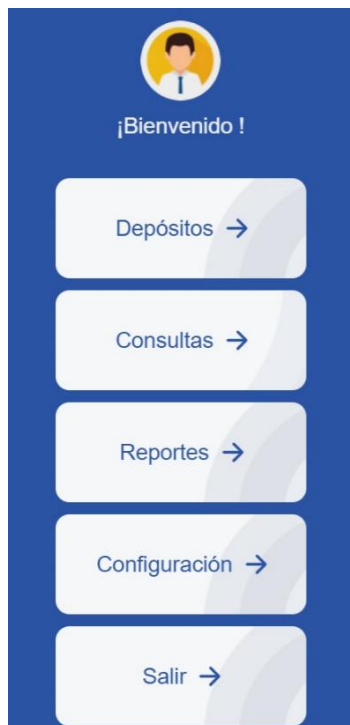
Ingresar

**Nota:** *Elaboración propia.*

#### **4.4.2. Pantalla de Inicio**

Una vez que el agente recaudador inicia sesión se presenta la pantalla de inicio en donde se muestran cada uno de los botones que brindan acceso a cada uno de los módulos del aplicativo. Dando la facilidad al agente recaudador de acceder de forma rápida a la tarea que desea realizar.

**Ilustración 13.** Página de inicio.



**Nota:** Elaboración propia.

### 3.8.3. Menú de Navegación

El menú de navegación se muestra en cada uno de los módulos que conforman el aplicativo, permitiendo al agente cambiar entre módulos con facilidad incluyendo la página de inicio. Además, el menú incorpora una estética visual con iconos, propios del framework Ionic, en el cual se desarrolló el aplicativo.

**Ilustración 14.** Menú de navegación.



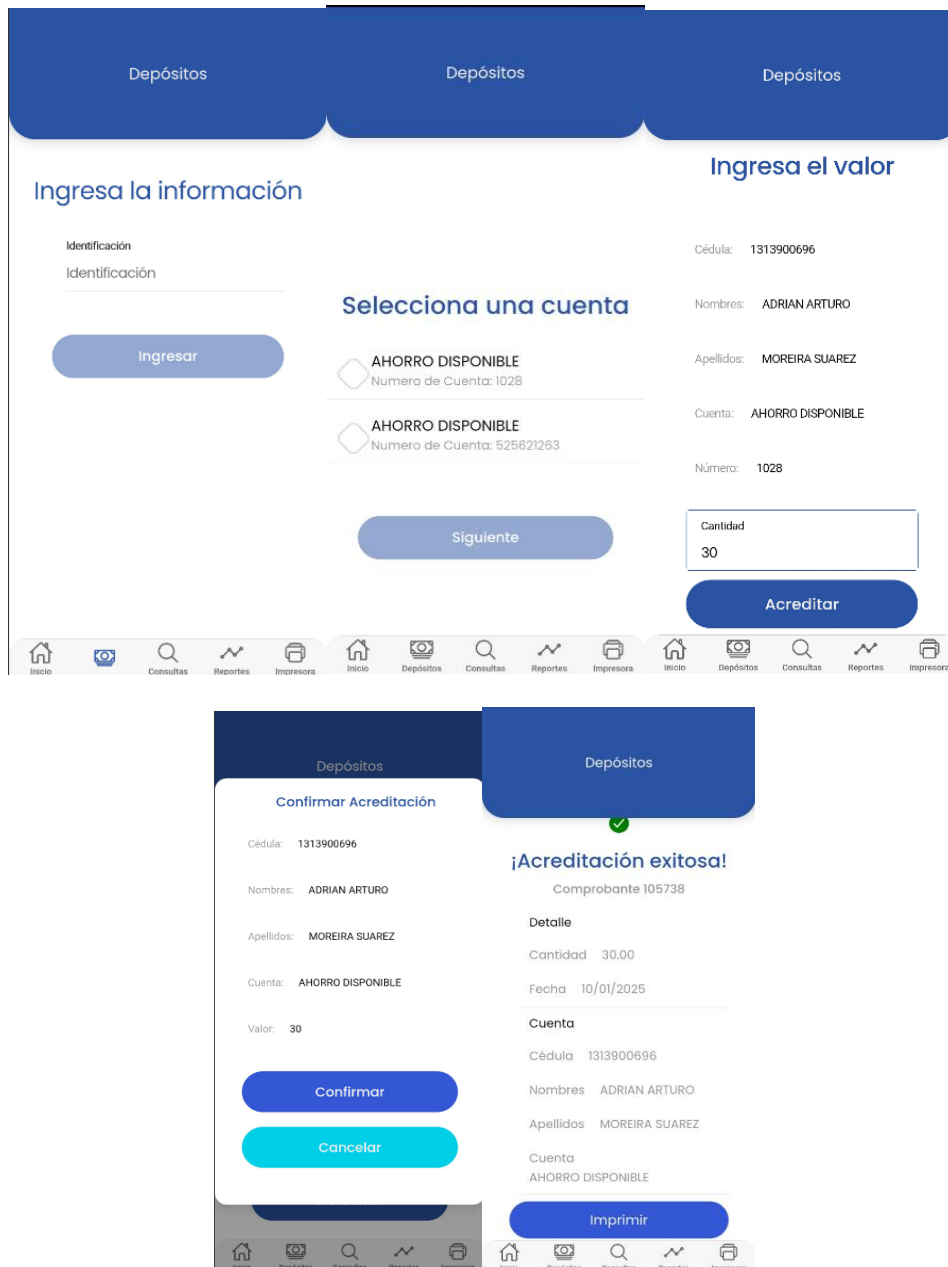
**Nota:** Elaboración propia.

### 3.8.4. Pantallas de Depósitos

Una vez que el agente haya accedido al módulo de depósitos mediante la pantalla principal o el menú de navegación, se presentan los formularios y componentes que muestran la información de las cuentas de ahorro y la información de la transacción que se está realizando. Cabe mencionar que, sí la información ingresada es incorrecta, no se

cumplirá con el flujo del sistema y, por lo tanto, no se mostrarán las pantallas correspondientes a dicho módulo.

**Ilustración 15.** Pantallas de depósitos.



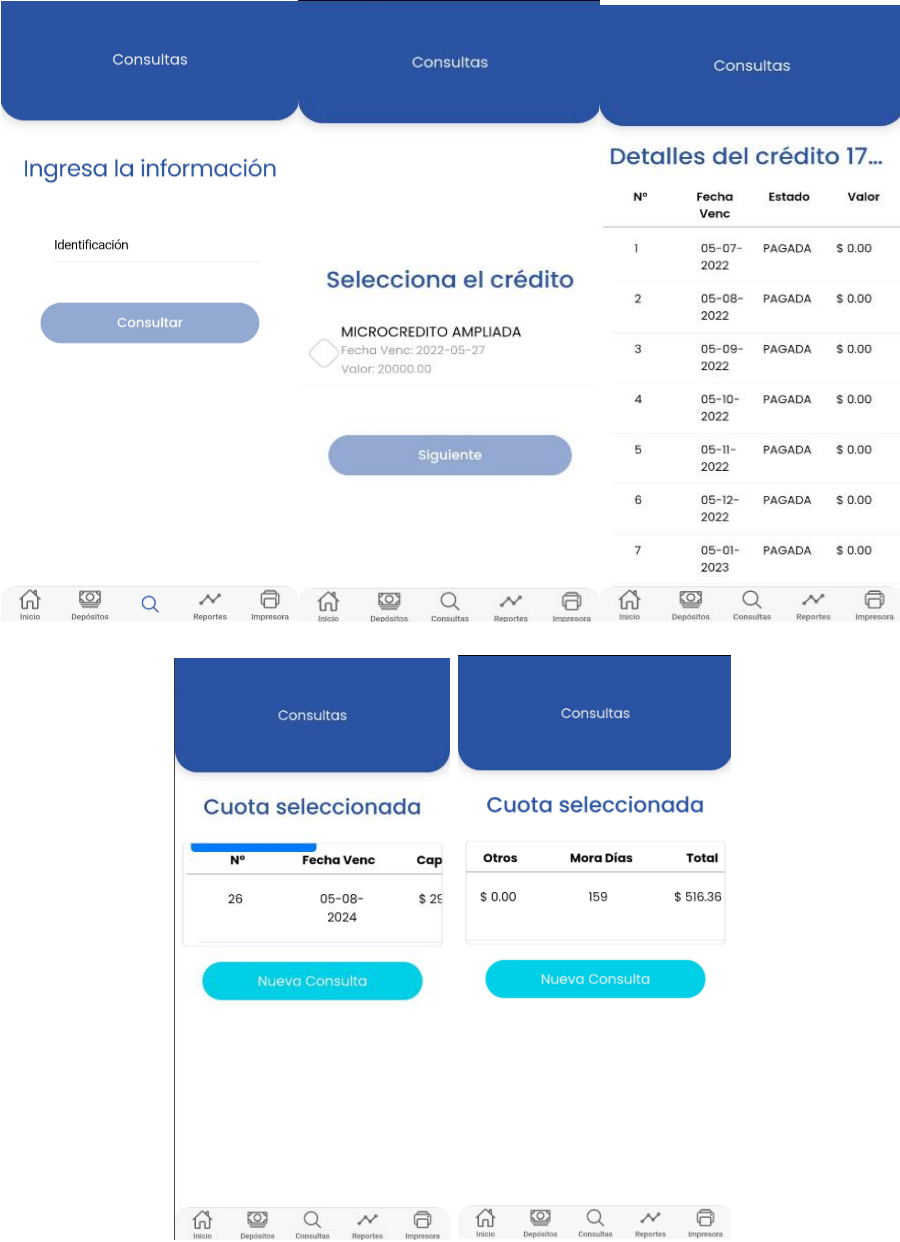
**Nota:** Elaboración propia.

### 3.8.5. Pantallas de Consultas

De la misma forma como en las páginas de depósitos, se presentan los componentes que permiten ingresar y mostrar información correspondiente a los créditos

del cliente. En comparación con la transacción de depósitos, las consultas de crédito no son muy extensas y no requieren de muchas pantallas. Sin embargo, si la información ingresada es incorrecta no se podrá continuar con el proceso del módulo de consultas.

**Ilustración 16. Pantallas de consultas.**



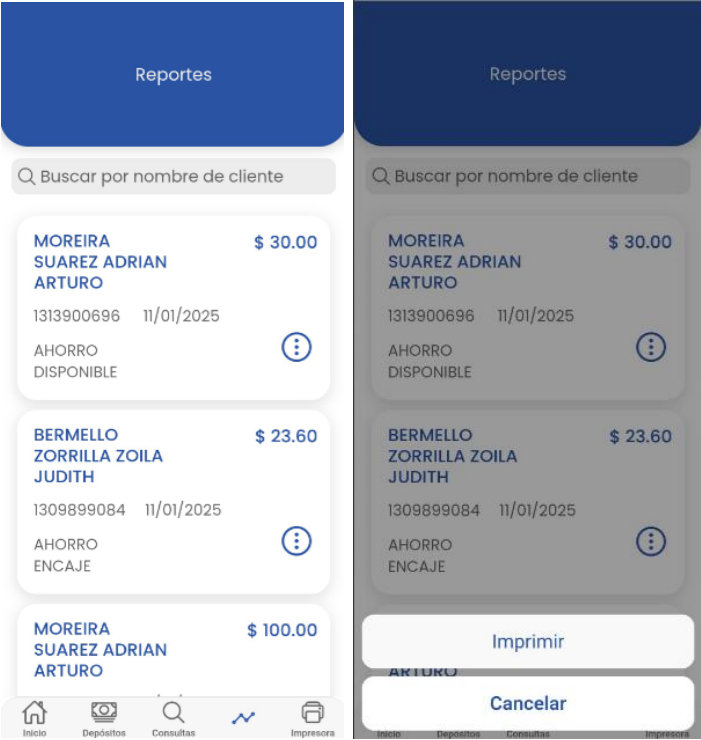
**Nota:** Elaboración propia.

**3.8.6. Pantallas de Reportes**

En cuanto a los reportes, se presenta la información de todos los depósitos realizados por el agente, a través de componentes como tarjetas o "cards". Donde se

muestra el nombre del cliente, número de cédula, número de cuenta, fecha y valor del depósito. Adicionalmente, se implementa un botón que permite imprimir el comprobante de cada transacción correspondiente.

**Ilustración 17.** Pantallas de reportes.

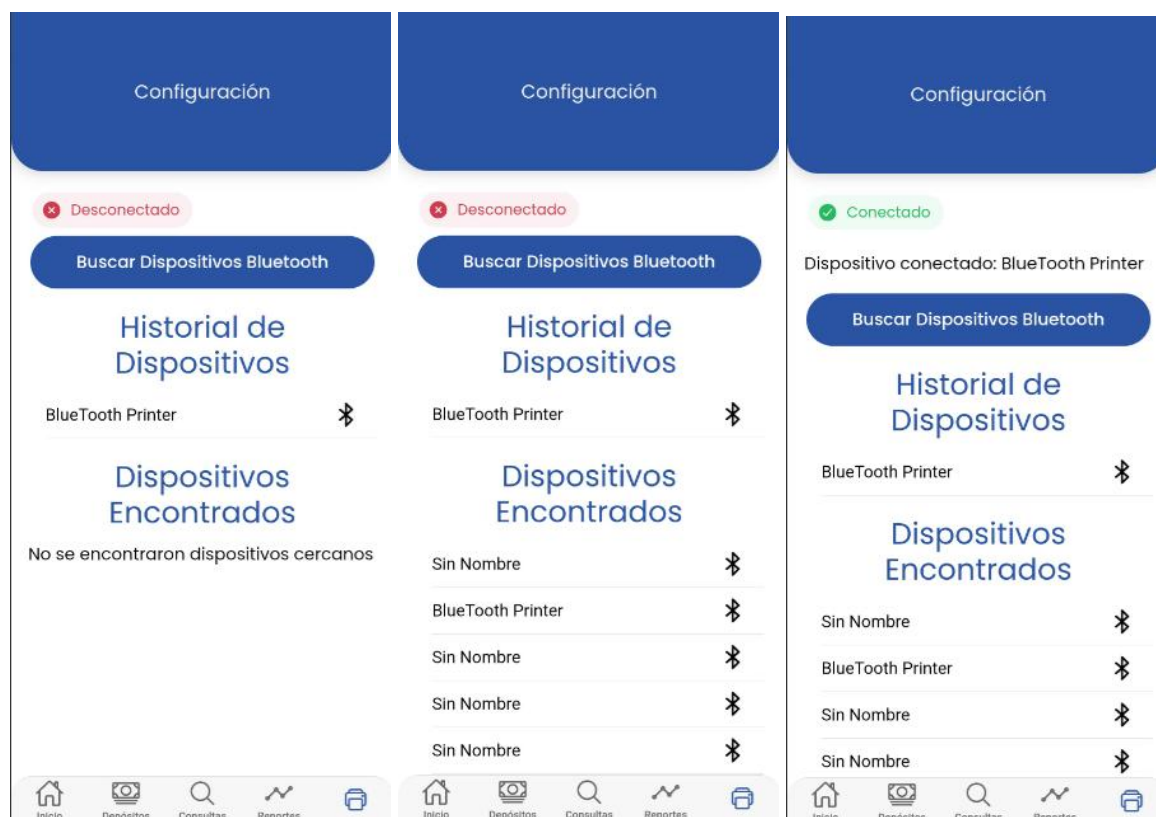


**Nota:** Elaboración propia.

### 3.8.7. Pantalla de Configuración

Para la configuración, se muestra el estado de conexión Bluetooth con la impresora, adicionando la funcionalidad de buscar dispositivos Bluetooth cercanos, guardar los dispositivos conectados anteriormente y desactivar la conexión. Los dispositivos que se encuentran se visualizarán dentro de una tabla.

**Ilustración 18.** Pantallas de configuración.

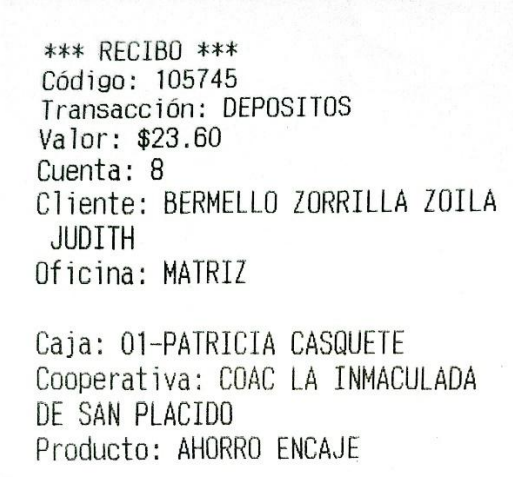


**Nota:** Elaboración propia.

### 3.8.8. Recibos o Comprobantes

Para finalizar con la fase de desarrollo, es necesario implementar el contenido que se mostrara dentro del recibo. Para ello, se implementa un formato que incluye información como el código de comprobante, el valor depositado, nombre del cliente propietario del producto de ahorro, el nombre del agente recaudador y finalmente el nombre de la cooperativa.

**Ilustración 19.** Comprobante o recibo del depósito.



\*\*\* RECIBO \*\*\*  
Código: 105745  
Transacción: DEPOSITOS  
Valor: \$23.60  
Cuenta: 8  
Cliente: BERMELLO ZORRILLA ZOILA  
JUDITH  
Oficina: MATRIZ

Caja: 01-PATRICIA CASQUETE  
Cooperativa: COAC LA INMACULADA  
DE SAN PLACIDO  
Producto: AHORRO ENCAJE

**Nota:** Elaboración propia.

## 4.5. Implementación de la Aplicación

En este presente apartado se describe el enfoque para transformar el diseño y requerimientos del aplicativo en un producto listo para su uso por los usuarios a quienes está destinado. Para ello, se realiza la combinación de distintos procesos y la ejecución de rigurosas pruebas para asegurar la excelencia y la eficacia del aplicativo.

### 4.5.1. Pruebas Funcionales

Las pruebas de funcionalidad se enfocan en verificar el adecuado rendimiento y desempeño de las funciones de la aplicación, garantizando su correcto funcionamiento conforme según lo estimado. Estas pruebas aseguran la calidad y la correcta implementación del sistema para cumplir con su propósito. Esta etapa nos permite identificar errores y problemas antes de la implementación del software.

### 4.5.2. Pruebas no Funcionales

Al igual que las pruebas funcionales, estas evalúan aspectos del software que no están directamente relacionados con su funcionalidad. Estas pruebas abordan áreas esenciales como la eficiencia, seguridad, usabilidad y compatibilidad de software. Garantizando que el aplicativo cumpla con los requisitos en términos de rendimiento, seguridad, experiencia de usuario, etc.

### **4.5.3. Pruebas de Carga**

Estas pruebas permiten analizar la capacidad de respuesta de un sistema frente situaciones de carga intensa, simulando el comportamiento de múltiples usuarios accediendo al aplicativo simultáneamente. Su objetivo es descubrir problemas en el rendimiento y otros posibles inconvenientes, garantizando un rendimiento óptimo del sistema y asegurando una experiencia fluida para los usuarios en condiciones de carga excesiva.

### **4.5. Pruebas de Usabilidad**

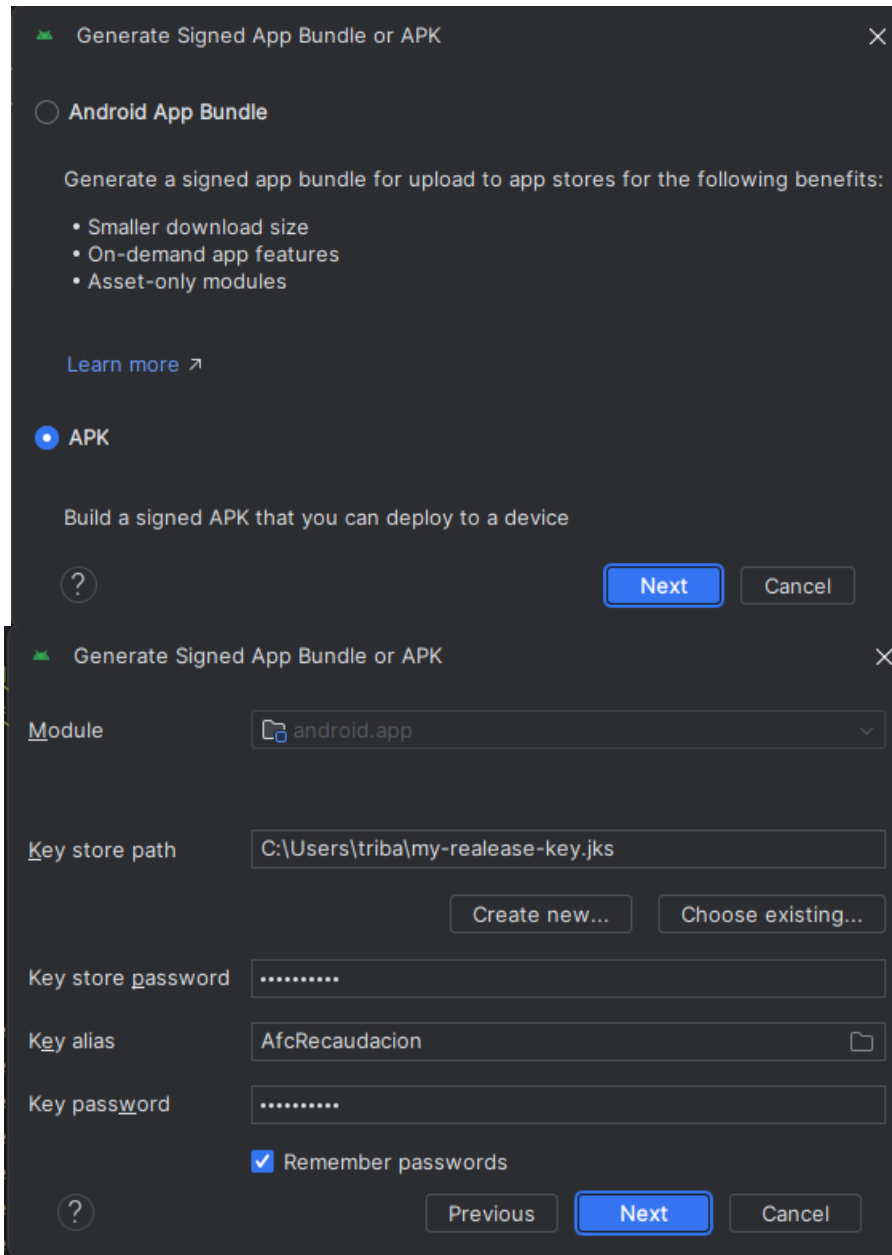
Estas pruebas evalúan si el aplicativo es agradable y fácil de usar para todos los usuarios. A diferencia de las pruebas anteriores, estas se centran en medir la facilidad de uso del sistema. Basándose en la retroalimentación del usuario, se identifican aspectos y oportunidades de mejora para versiones futuras de la aplicación. Para registrar estas pruebas, se implementó un cuestionario que los usuarios de prueba fueron concretando una vez finalizada su experiencia con el aplicativo.

### **4.6. Implementación del Aplicativo en Formato APK**

“El formato APK (Android Application Package), se los conoce a los archivos ejecutables de instalación que se han diseñado para dispositivos con sistema operativo Android, permitiendo la descarga de aplicaciones y su ejecución” (Miguel, 2025). Para la transformación del aplicativo en dicho formato, el framework Angular/Ionic permite sincronizar los cambios realizados durante todo el proceso de desarrollo en una carpeta Android, que posteriormente mediante el uso de Android Studio se realiza la transformación del aplicativo con su respectiva firma.

La firma digital es un proceso obligatorio que asegura la integridad de un archivo APK. Se considera como un certificado criptográfico que verifica que la aplicación es proveniente de un desarrollador legítimo. Además, garantiza que el APK no ha sido alterado una vez que haya sido firmada. El estándar de Android exige que todas sus aplicaciones estén firmadas antes de ser instaladas en el dispositivo.

**Ilustración 20.** Generación del archivo APK firmado.



## 4.7. Credenciales Utilizadas Durante las Pruebas

**Tabla 21:** Credenciales de los agentes de recaudación.

Usuario	Contraseña	Número telefónico
FACILITO	MOLINO23	0990378338
PATRICIA	MOLINO23	0939203784

**Nota:** Estas credenciales fueron utilizadas con el propósito de realizar las pruebas correspondientes al sistema.

## 4.8. Resultados

Se utilizó la tabla 22 para las pruebas de funcionalidad, la tabla 23 para las pruebas no funcionales, la tabla 24 y 25 para realizar las pruebas de carga. Estas pruebas permiten determinar si el sistema presenta fallos o factores que afecten su funcionamiento y la experiencia del usuario. Adicionalmente, proporcionan una estructura para el registro y análisis de resultados, facilitando la detección de fallos y toma de decisiones para mejorar la calidad del software.

Para las pruebas de carga, se hizo uso de la herramienta Apache JMeter, permitiendo evaluar el sistema bajo situaciones de alta concurrencia, mediante la simulación de múltiples usuarios enviando solicitudes a las APIs del sistema.

- **Herramientas utilizadas:**
  - Apache JMeter.

**Tabla 22:** Pruebas Funcionales.

Caso de prueba	Descripción	Entrada	Resultado.	Resultado esperado	¿Pasa la prueba?
CP-001	Instalación del aplicativo.	Aplicativo en formato APK.	La aplicación se instala de forma correcta.	Ilustración 21 e ilustración 22.	Pasó
CP-002	Verificar si la aplicación se puede abrir.	Ícono que representa al aplicativo.	La pantalla de inicio de sesión se muestra sin errores.	Ilustración 23.	Pasó
CP-003	Acceso al aplicativo una vez iniciada la sesión.	Credenciales de inicio de sesión.	Se muestra la pantalla de inicio de la aplicación.	Ilustración 24.	Pasó.
CP-004	Acceso a módulo de depósitos.	Pulsar en el botón de depósitos en	El formulario de ingreso de cédula en el	Ilustración 25.	Pasó.

		la página de inicio del aplicativo o menú de navegación.	módulo de depósitos se muestra correctamente.		
CP-005	Acceso a productos de ahorro del cliente.	Ingreso de cédula en el módulo de depósitos.	Se muestra la información de productos de ahorro.	Ilustración 25.	Pasó.
CP-006	Ingreso del valor a depositar y confirmación de acreditación.	Selección de un producto de ahorro y el valor a depositar.	Se muestra la página que confirma la transacción.	Ilustración 25.	Pasó.
CP-007	Impresión de recibo del depósito.	Pulsar el botón de imprimir recibo.	Se imprime el recibo de la transacción correspondiente.	Ilustración 26.	Pasó.
CP-008	Acceso a módulo de consultas.	Pulsar en el botón de consultas de la página de inicio del aplicativo o en el menú de navegación.	El formulario de ingreso de cédula en el módulo de consultas se muestra correctamente.	Ilustración 27.	Pasó.
CP-009	Acceso a productos de crédito del cliente.	Ingreso de cédula en el módulo de consultas.	Se muestra la información de productos de ahorro.	Ilustración 27.	Pasó.
CP-010	Visualización de cuotas de crédito.	Selección de un producto de crédito.	Se muestra la tabla con todas las cuotas que	Ilustración 27.	Pasó.

			conforman el crédito.		
CP-011	Visualización de información cuotas de crédito.	Selección de una cuota de crédito activa.	Se muestra la tabla con toda la información de la cuota correspondiente.	Ilustración 27.	Pasó.
CP-012	Visualización de reportes.	Pulsar en el botón de reportes de la página de inicio del aplicativo o menú de navegación.	Se muestran todos los reportes realizados en el día.	Ilustración 28.	Pasó.
CP-013	Búsqueda de reportes.	Ingreso del nombre de un cliente.	Se muestran todos los reportes del socio o cliente buscado.	Ilustración 28.	Pasó.
CP-014	Impresión de un reporte específico.	Pulsar el botón de impresión.	Se imprime el recibo de la transacción correspondiente.	Ilustración 28.	Pasó.
CP-015	Acceso a módulo de configuración.	Pulsar en el botón de configuración de la página de inicio o menú de navegación.	Se muestra la página del módulo de configuración.	Ilustración 29.	Pasó.
CP-016	Búsqueda de dispositivos bluetooth.	Pulsar en el botón de búsqueda, de	Se muestra la tabla con los dispositivos	Ilustración 29.	Pasó.

		dispositivos bluetooth.	Bluetooth encontrados.		
CP-017	Conexión con la impresora Bluetooth.	Pulsar en la impresora Bluetooth con el que se desea enlazar.	Se muestra la conexión establecida con el dispositivo Bluetooth.	Ilustración 29.	Pasó.
CP-018	Registro histórico de dispositivo Bluetooth.	El dispositivo conectado se almacena en el historial de dispositivos.	Se muestra el dispositivo registrado en el historial de dispositivos.	Ilustración 29.	Pasó.
CP-019	Cierre de sesión desde la página de inicio.	Pulsar el botón de salir, desde la pantalla de inicio.	Se muestra la página del inicio de sesión, indicando que se ha cerrado con éxito el sistema.	Ilustración 30.	Pasó.
CP-020	Manejo de excepciones.	Ingreso incorrecto de datos.	Bloqueo del flujo del sistema y se muestra la retroalimentación al usuario.	Ilustración 31.	Pasó.
CP-021	Cierre del sistema por tiempo de inactividad.	Inactividad dentro del sistema.	Cierre de la sesión del usuario.	Ilustración 32.	Pasó.
CP-022	Cierre forzado del sistema, cuando el aplicativo se muestra en	Cierre del aplicativo.	Se muestra nuevamente el inicio de sesión.	Ilustración 32.	Pasó.

	segundo plano.				
--	----------------	--	--	--	--

**Nota:** Se muestran las pruebas funcionales que se realizaron.

**Tabla 23:** Pruebas no funcionales.

Caso de prueba	Tipo de prueba	Resultado esperado	Resultado real	¿Pasa la prueba?
CPNF-001	Compatibilidad	Capacidad de la aplicación para funcionar en diferentes dispositivos con diferentes versiones del sistema operativo Android.	Tabla 26.	Pasó
CPNF - 002	Seguridad	Capacidad de la aplicación para funcionar en diferentes dispositivos con parches de seguridad.	Tabla 26.	Pasó
CPNF - 003	Disponibilidad	Capacidad del aplicativo, para estar disponible en todo momento.	Tabla 26.	Pasó.

**Nota:** Se muestran las pruebas no funcionales que se realizaron.

**Tabla 24:** Pruebas de carga N°1.

Parámetro	API de inicio de sesión.	API de consulta de información del socio.	API de acceso a las cuentas de ahorro del cliente.	API de depósitos.	API de acceso a los créditos de un cliente.
Número de hilos (Usuarios)	500	500	500	500	500
Número de peticiones	500	500	500	500	500
Porcentaje de error	0,00%	0,00%	0,00%	0,00%	0,00%
Throughput (peticiones/seg)	2,5/sec	13,0/sec	25,6/sec	30,0/sec	20,5/sec
Ilustración	Ilustración 33	Ilustración 34	Ilustración 35	Ilustración 36	Ilustración 37
¿Pasa la prueba?	Pasó.	Pasó.	Pasó.	Pasó.	Pasó.

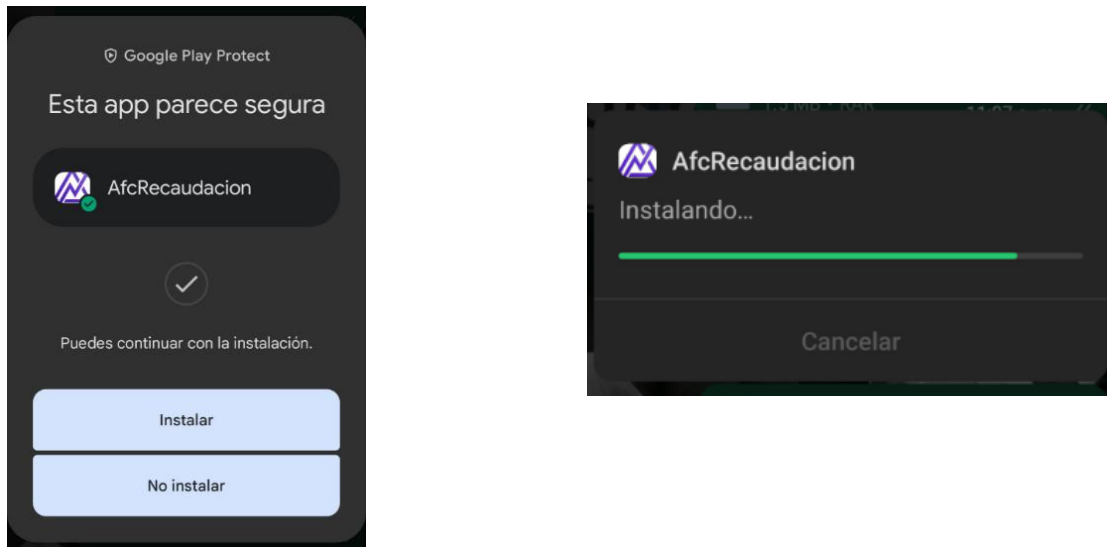
**Nota:** Se muestran las pruebas de carga que se realizaron.

**Tabla 25:** Pruebas de carga N°2.

Parámetro	API de acceso a la información del crédito.	API de acceso a la información de la cuota.	API de acceso a reportes.	API de impresión del recibo.	API de cierre de sesión.
Número de hilos (Usuarios)	500	500	500	500	500
Número de peticiones	500	500	500	500	500
Porcentaje de error	0,00%	0,00%	0,00%	0,00%	0,00%
Throughput (peticiones/seg)	24,5/sec	6,4/sec	17,4/sec	14,3/sec	18,4/sec
Ilustración	Ilustración 38	Ilustración 39	Ilustración 40	Ilustración 41	Ilustración 42
¿Pasa la prueba?	Pasó.	Pasó.	Pasó.	Pasó.	Pasó.

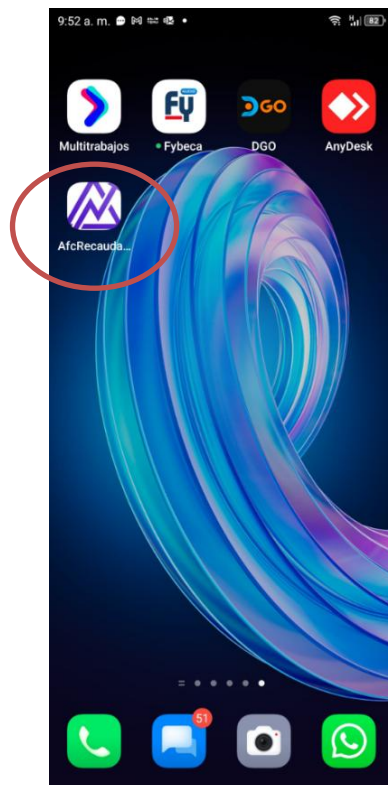
**Nota:** Se muestran las pruebas de carga que se realizaron.

**Ilustración 21.** Instalación de la aplicación.



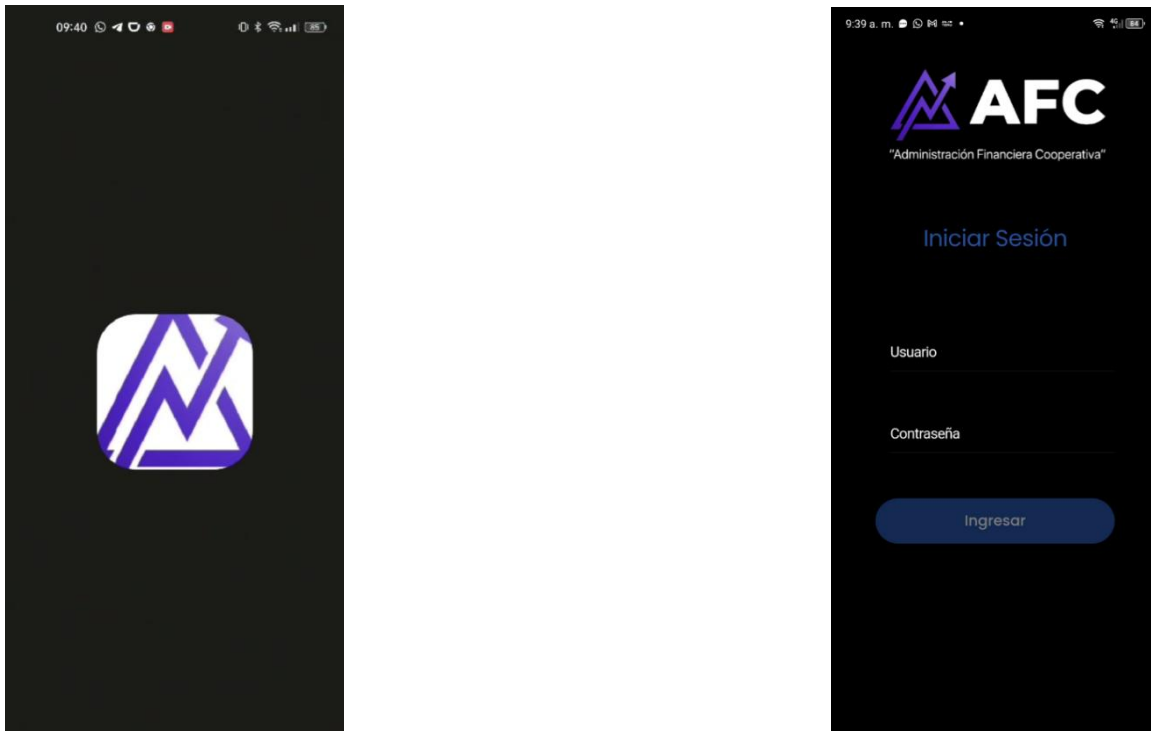
**Nota:** En la presente figura, se muestra la instalación del aplicativo para proceder con su apertura dentro del dispositivo.

**Ilustración 22.** Se muestra el icono funcional del aplicativo.



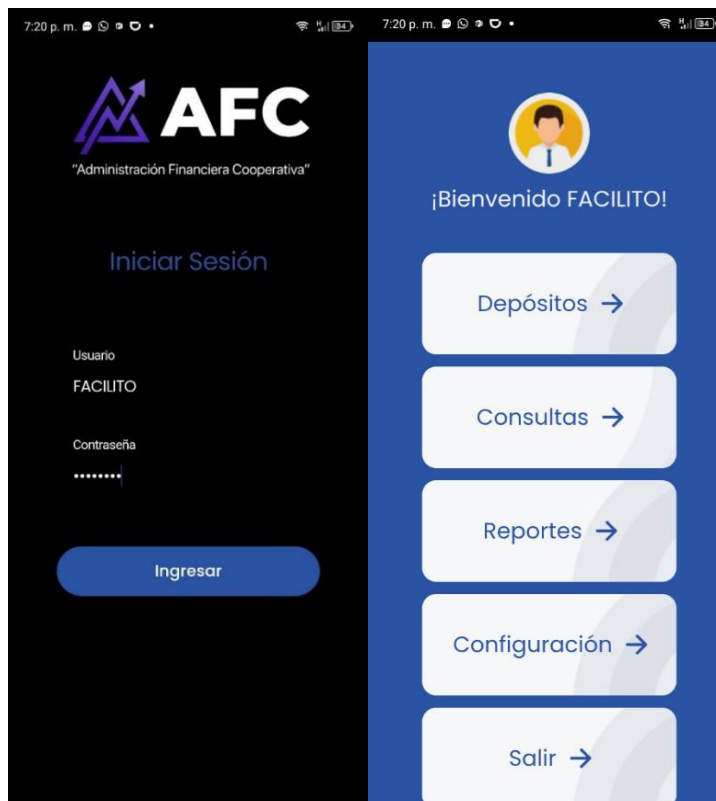
**Nota:** En la presente figura, se muestra el aplicativo dentro de las pantallas del dispositivo móvil.

**Ilustración 23.** Apertura del aplicativo.



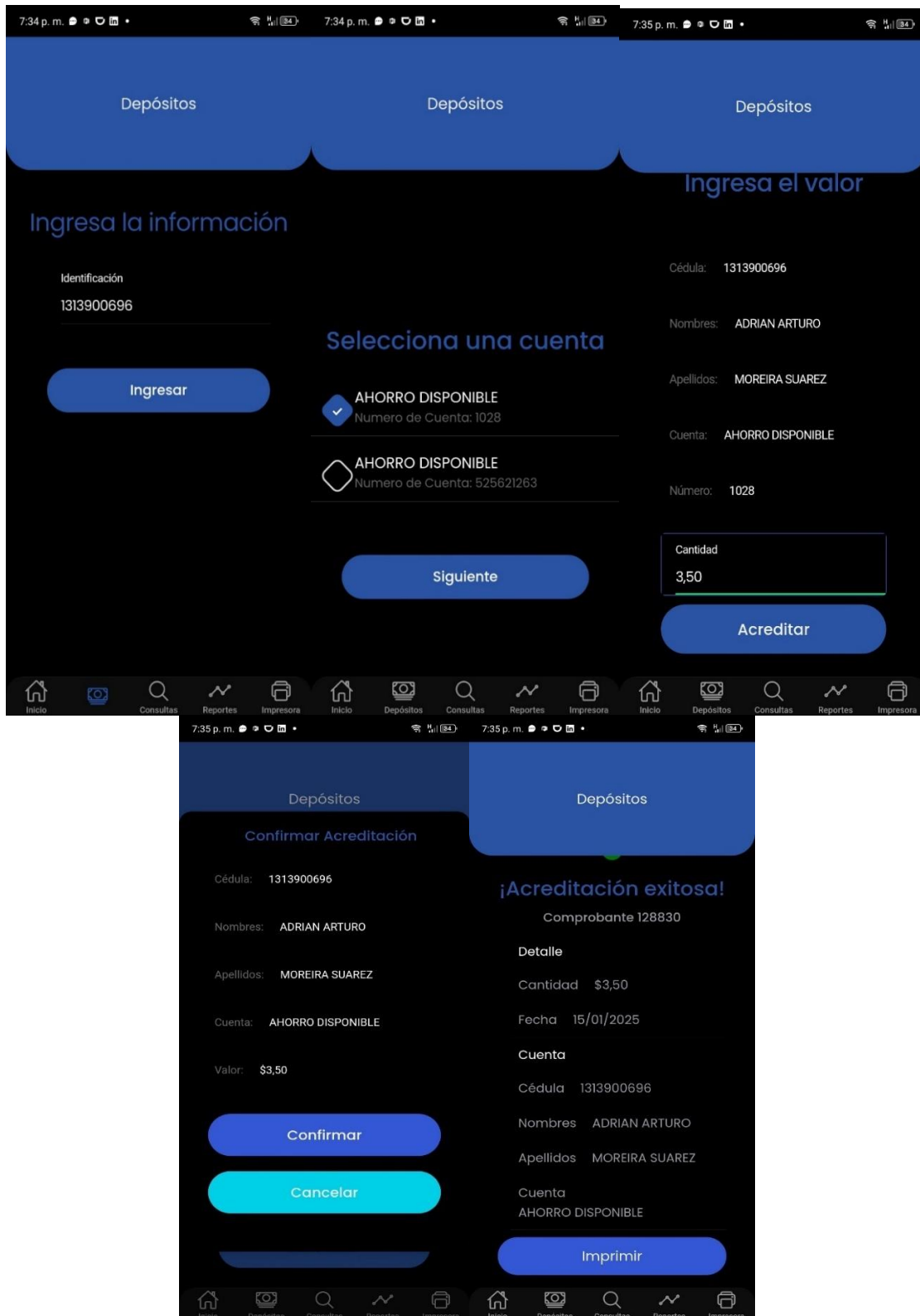
**Nota:** Se muestra la apertura exitosa del aplicativo en un dispositivo físico con sistema Android.

**Ilustración 24.** Inicio de sesión.



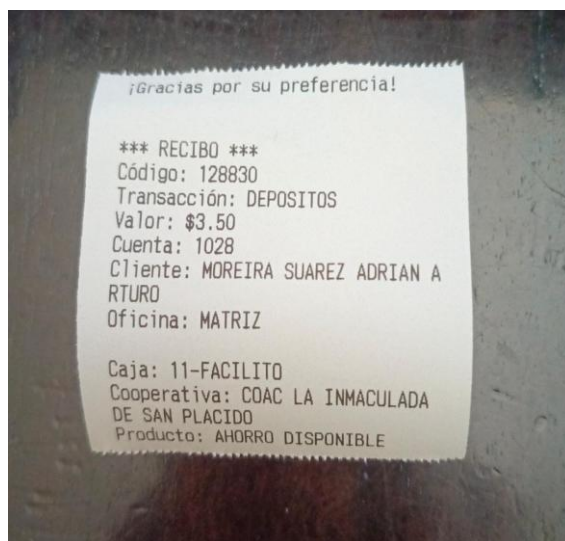
**Nota:** Se muestra el inicio de sesión del sistema relajando que fue exitoso, accediendo la página de inicio del aplicativo.

**Ilustración 25.** Se muestran las pantallas del módulo de depósitos.



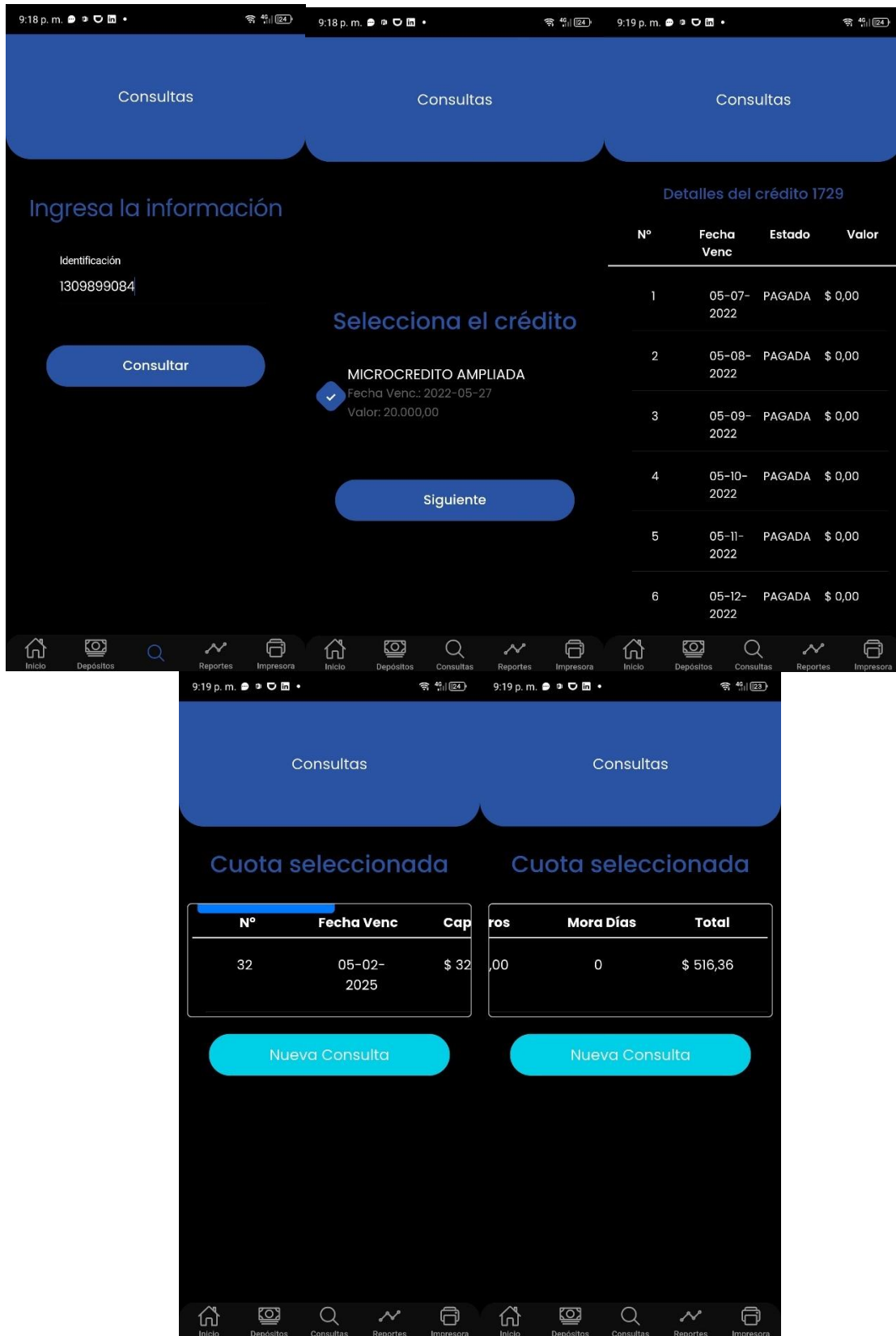
**Nota:** Se muestran las pantallas de depósitos siguiendo su flujo principal en el dispositivo.

**Ilustración 26.** Recibo correspondiente al depósito correspondiente.



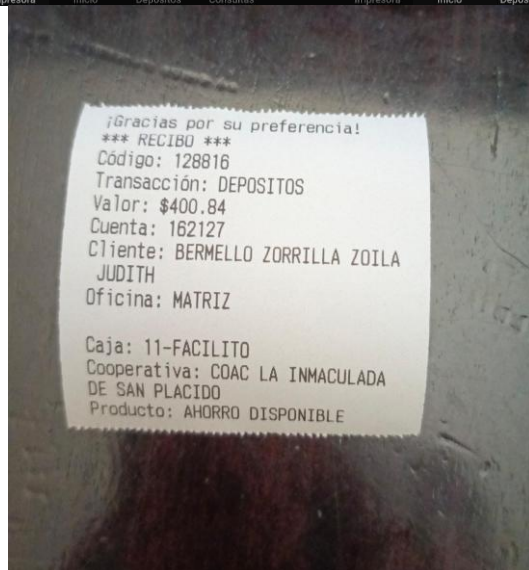
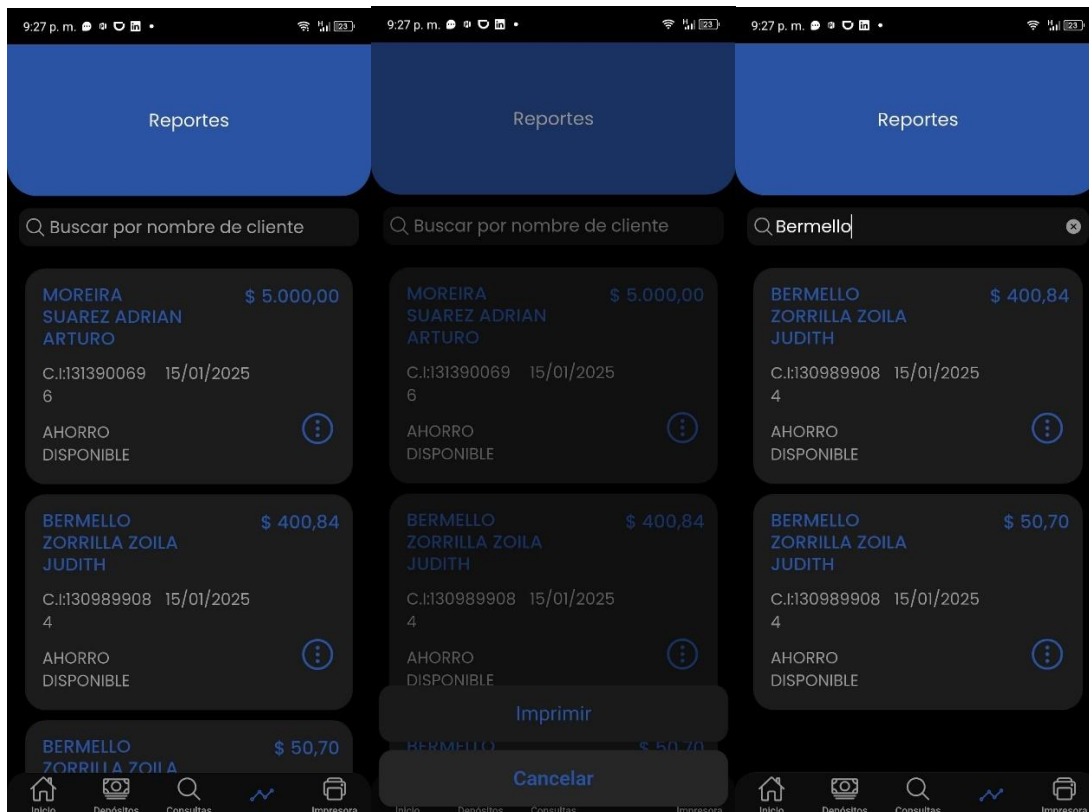
**Nota:** Se muestra el comprobante físico, finalizando el flujo de depósitos.

**Ilustración 27.** Se muestran las pantallas del módulo de consultas.



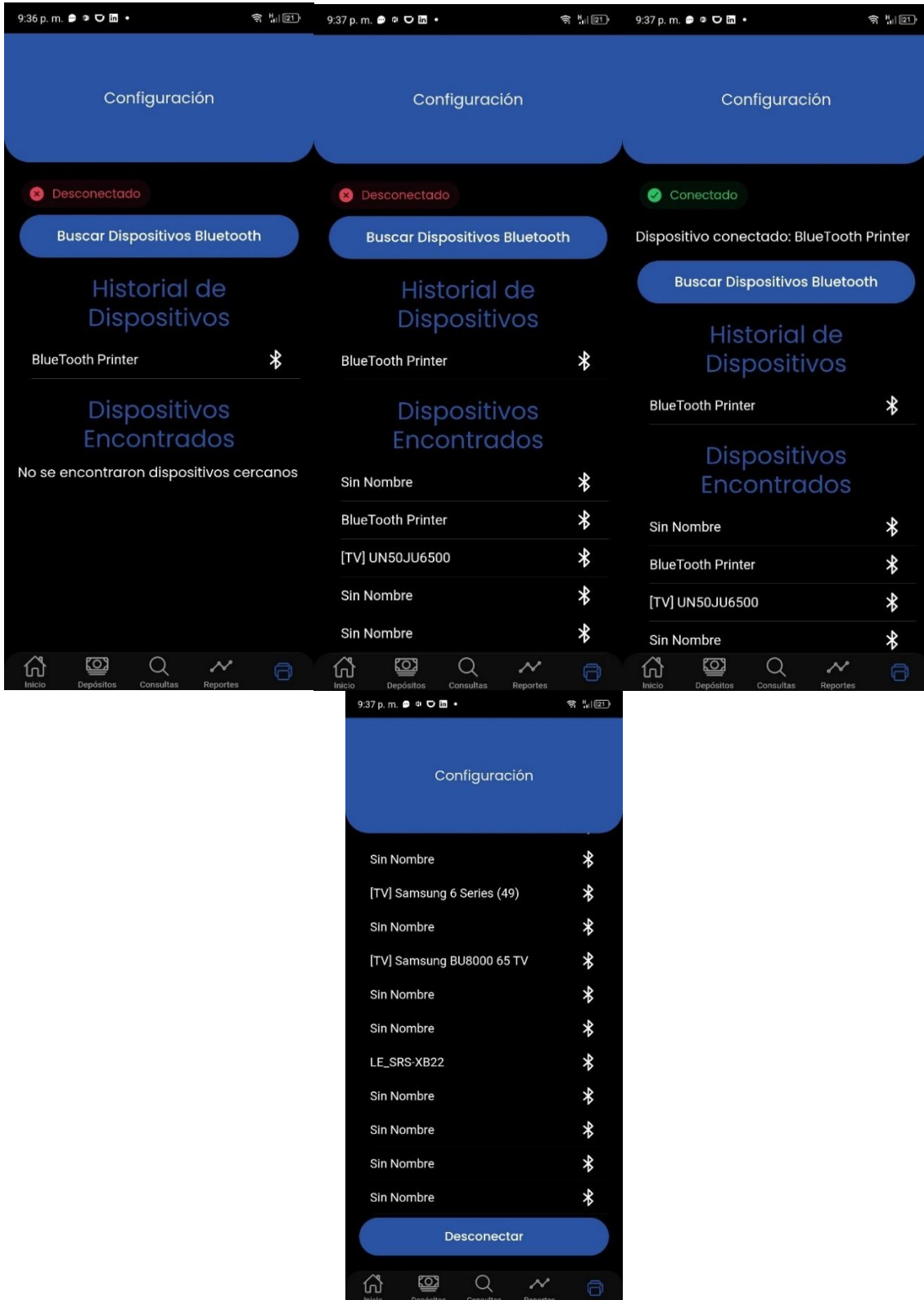
**Nota:** Se muestran las pantallas de consultas siguiendo su flujo principal en el dispositivo.

**Ilustración 28.** Se muestran las pantallas del módulo de reportes.



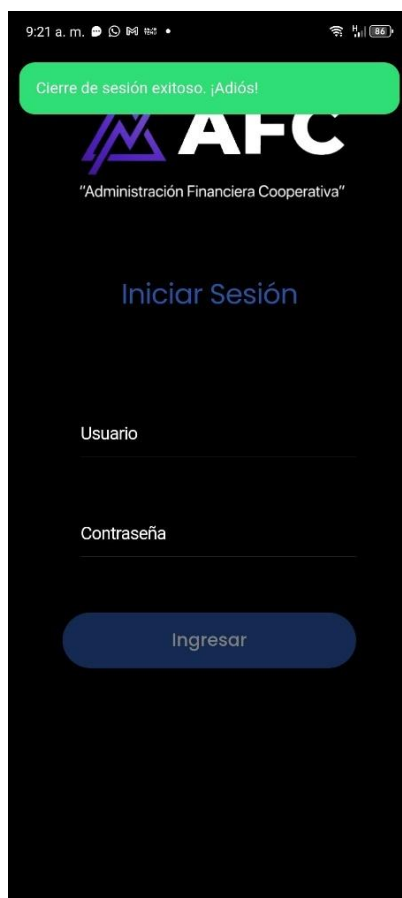
**Nota:** Se muestran las pantallas de consultas, siguiendo su flujo principal en el dispositivo.

**Ilustración 29.** Se muestran las pantallas del módulo de configuración.



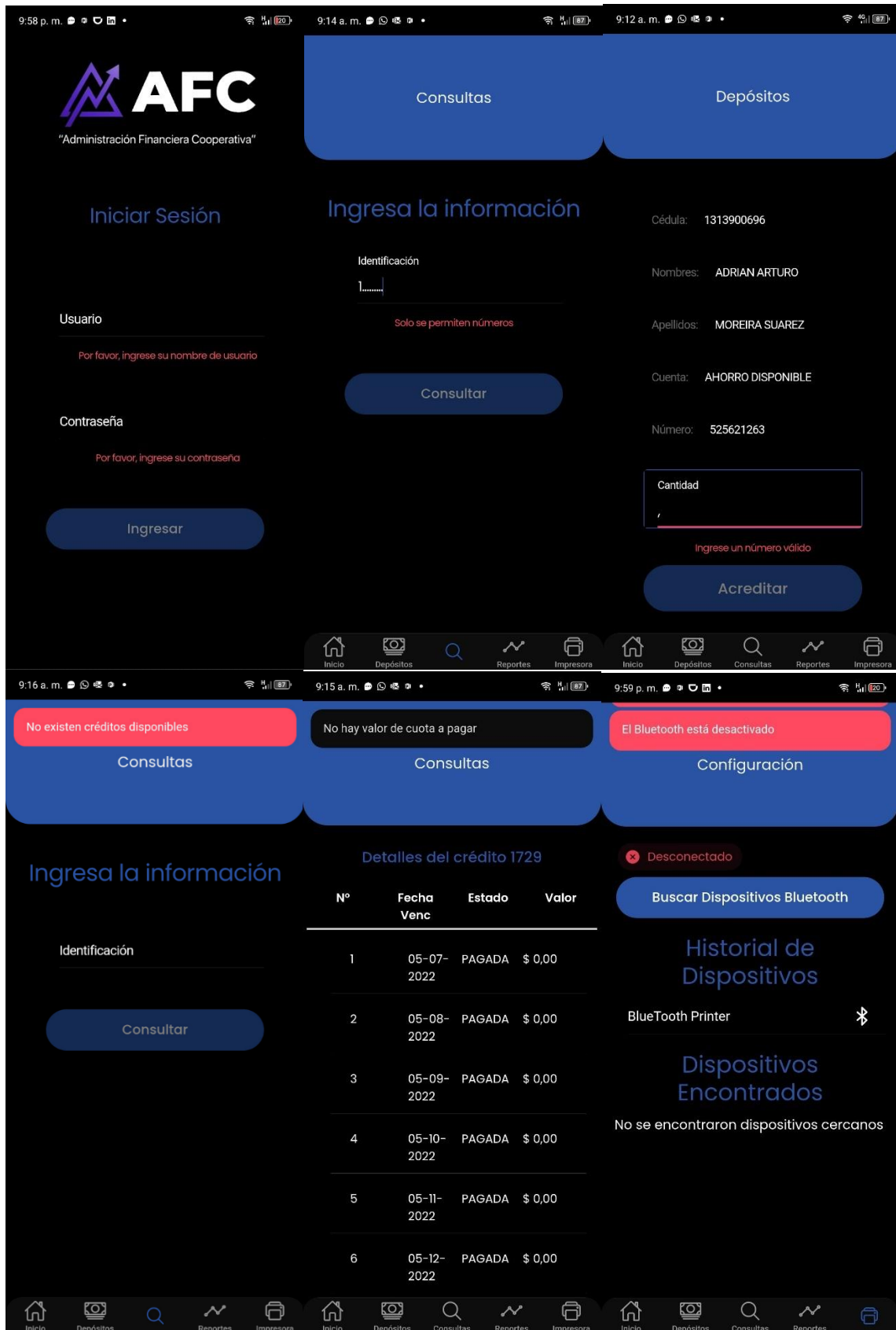
**Nota:** Se muestran las pantallas de configuración siguiendo su flujo principal en el dispositivo.

**Ilustración 30.** Cierre de sesión desde la página de inicio.



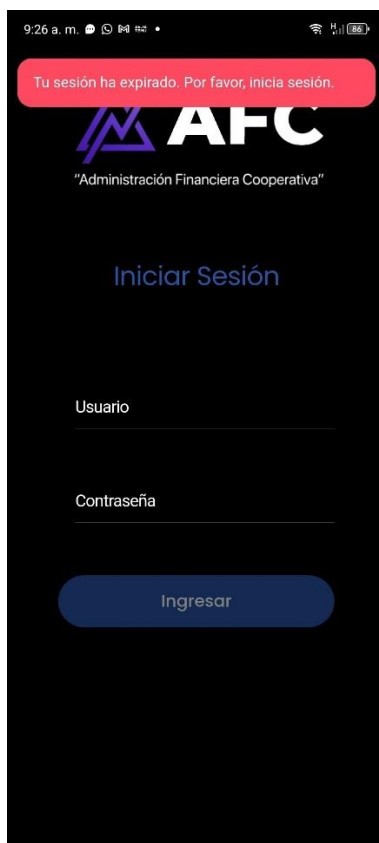
**Nota:** Se muestra el mensaje de confirmación del cierre de sesión.

**Ilustración 31. Manejo de excepciones.**



**Nota:** Se muestran los errores que puede presentar el sistema, brindando retroalimentación al usuario para que pueda solventarlos.

**Ilustración 32.** Cierre de sesión del sistema.



**Nota:** Se presenta el mensaje de cierre de sesión por inactividad o cuando la aplicación pase a segundo plano.

**Tabla 26:** Dispositivos en los que se instaló la aplicación.

Dispositivo	Versión de Android	Versión de parche de seguridad Android
Infinix Note 30 Pro	Android 14	2024-12-05
Redmi Note 11 Pro	Android 14	2024-12-15
Samsung Galxy A15	Android 14	2024-12-10
HONOR 90 Lite	Android 14	2024-12-01
Redmi 12C	Android 13	2024-10-15
Realme 8 Pro	Android 13	2024-04-05
Realme 6 Pro	Android 11	2023-06-05
Redmi 9A	Android 10	2023-02-01
Huawei P30	Android 10	2022-05-10
Samsung Galaxy S9	Android 9.0 Pie	2020-12-01

**Ilustración 33.** Reporte de rendimiento – API de inicio de sesión.

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	500	39318	1339	195850	63100.16	0.00%	2.5/sec	1.06	0.87	427.0
TOTAL	500	39318	1339	195850	63100.16	0.00%	2.5/sec	1.06	0.87	427.0

**Nota:** Reporte de 500 peticiones a la API de inicio de sesión.

**Ilustración 34.** Reporte de rendimiento – API de consulta de información del socio.

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	500	19569	5542	37615	7815.43	0.00%	13.0/sec	8.12	3.94	639.5
TOTAL	500	19569	5542	37615	7815.43	0.00%	13.0/sec	8.12	3.94	639.5

**Nota:** Reporte de 500 peticiones a la API de consulta de información del socio.

**Ilustración 35.** Reporte de rendimiento – API de acceso a las cuentas de ahorro del cliente.

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	500	8974	427	18569	5354.29	0.00%	25.6/sec	30.21	7.20	1208.0
TOTAL	500	8974	427	18569	5354.29	0.00%	25.6/sec	30.21	7.20	1208.0

**Nota:** Reporte de 500 peticiones a la API de acceso a las cuentas de ahorro del cliente.

**Ilustración 36.** Reporte de rendimiento – API de depósitos.

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	500	892	531	16375	1435.48	0.00%	30.0/sec	43.46	10.03	1468.4
TOTAL	500	892	531	16375	1435.48	0.00%	30.0/sec	43.46	10.03	1468.4

**Nota:** Reporte de 500 peticiones a la API de depósitos.

**Ilustración 37.** Reporte de rendimiento – API de acceso a los créditos de un cliente.

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	500	438658	19165	3023503	1038746.40	0.00%	20.5/sec	35.30	11.20	2638.2
TOTAL	500	438658	19165	3023503	1038746.40	0.00%	20.5/sec	35.30	11.20	2638.2

**Nota:** Reporte de 500 peticiones a la API de acceso a los créditos de un cliente.

**Ilustración 38.** Reporte de rendimiento - API de acceso a la información del crédito.

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	500	19789	19204	20347	295.19	0.00%	24.5/sec	53.78	20.03	2249.0
TOTAL	500	19789	19204	20347	295.19	0.00%	24.5/sec	53.78	20.03	2249.0

**Nota:** Reporte de 500 peticiones a la API de acceso a la información del crédito.

**Ilustración 39.** Reporte de rendimiento - API de acceso a la información de la cuota.

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	500	22426	19200	78723	9550.92	0.00%	6.4/sec	16.47	6.20	2655.8
TOTAL	500	22426	19200	78723	9550.92	0.00%	6.4/sec	16.47	6.20	2655.8

**Nota:** Reporte de 500 peticiones a la API de acceso a la información de la cuota.

**Ilustración 40.** Reporte de rendimiento - API de acceso a reportes.

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename:  Browse... Log/Display Only:  Errors  Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	500	21034	19209	28693	1452.97	0.00%	17.4/sec	45.22	30.25	2657.3
TOTAL	500	21034	19209	28693	1452.97	0.00%	17.4/sec	45.22	30.25	2657.3

**Nota:** Reporte de 500 peticiones a la API de acceso a reportes.

**Ilustración 41.** Reporte de rendimiento - API de impresión del recibo.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	500	21658	19186	34866	3167.89	0.00%	14.3/sec	37.03	26.05	2644.5
TOTAL	500	21658	19186	34866	3167.89	0.00%	14.3/sec	37.03	26.05	2644.5

**Nota:** Reporte de 500 peticiones a la API de impresión del recibo.

**Ilustración 42.** Reporte de rendimiento - API de cierre de sesión.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	500	21769	19180	27210	2009.70	0.00%	18.4/sec	47.27	23.60	2635.4
TOTAL	500	21769	19180	27210	2009.70	0.00%	18.4/sec	47.27	23.60	2635.4

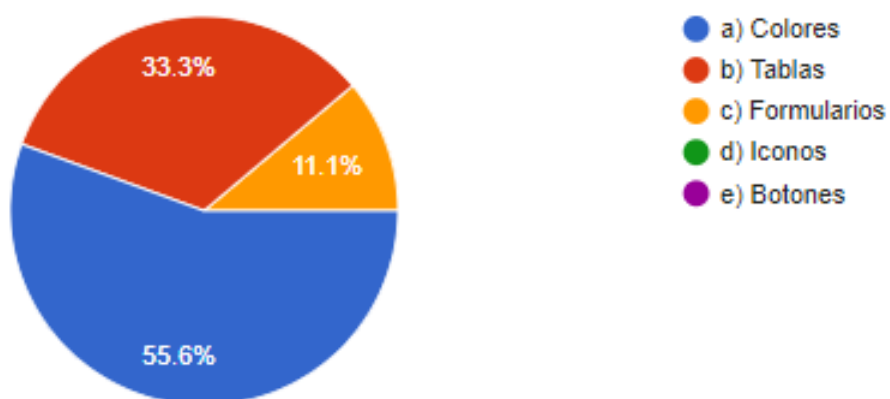
**Nota:** Reporte de 500 peticiones a la API de cierre de sesión.

## 4.9. Resultados de pruebas de usabilidad

En este apartado se detallan aquellas preguntas y respuestas del cuestionario, que proporcionan retroalimentación al sistema, permitiendo tomar decisiones para mejorar aspectos que resultaron complicados para el usuario. Las preguntas que generan retroalimentación fueron las siguientes:

**¿Qué aspecto del diseño de la aplicación consideras que necesita una mejora?**

**Ilustración 43.** Respuesta al cuestionario de pruebas de usabilidad 1.

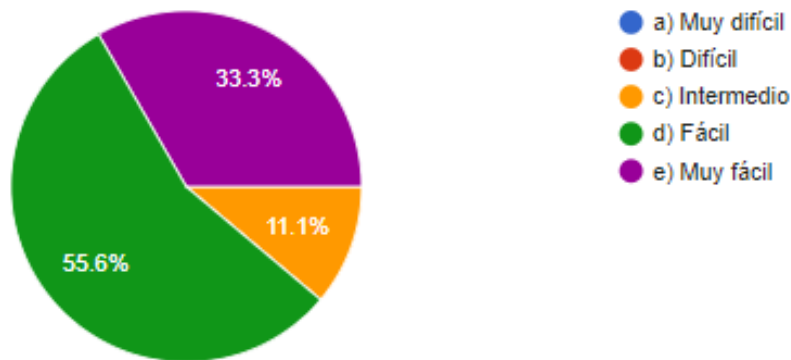


**Nota:** Tomado de cuestionario Post-Test para aplicación de recaudación [Fotografía], elaboración propia.

- **Oportunidad de mejora:** Basándome en las respuestas de los usuarios, los aspectos visuales menos agradables para el usuario son los colores, tablas y formularios. Por ello, para una futura oportunidad de mejora, es necesario entrevistar a agentes recaudadores para ajustar el aplicativo a colores más adecuados, rediseñar los formularios de entrada y finalmente reestructurar el diseño de las tablas de amortización para los créditos y sus cuotas.

**¿Qué tan fácil te resultó navegar por la aplicación y encontrar los módulos que buscabas?**

*Ilustración 44. Respuesta al cuestionario de pruebas de usabilidad 2.*

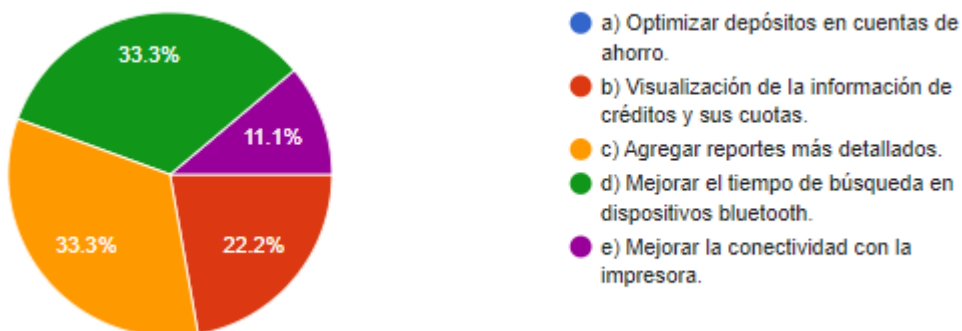


*Nota: Tomado de cuestionario Post-Test para aplicación de recaudación [Fotografía], elaboración propia.*

- **Oportunidad de mejora:** No existieron demasiados inconvenientes en cuanto a la navegación. Sin embargo, se recomienda un tipo de navegación más dinámica, para acceder a los módulos del aplicativo. Esto podría incluir el uso de gestos, reduciendo la dependencia de un menú de navegación.

**¿Qué área consideras que necesita más atención en futuras versiones de la aplicación?**

*Ilustración 45. Respuesta al cuestionario de pruebas de usabilidad 3.*



*Nota: Tomado de cuestionario Post-Test para aplicación de recaudación [Fotografía], elaboración propia.*

- **Oportunidad de mejora:** El área donde se requiere más atención es el tiempo de búsqueda de dispositivos, ya que esta suele extenderse en un lapso de diez segundos. En otras versiones del aplicativo esta debe durar cinco segundos. Además, se requieren reportes con más información acerca de las transacciones realizadas. Asimismo, es necesario mejorar el módulo de créditos y sus tablas, tal como se mencionó anteriormente.

## CONCLUSIONES Y RECOMENDACIONES

### 5.1. Conclusiones

- Se desarrollo un aplicativo móvil destinado a las INSTITUCIONES FINANCIERAS DE EPS, para facilitar el proceso de gestión de cobranza. Con la finalidad de automatizar las tareas correspondientes a dicho proceso, como la validación de depósitos, consultas de créditos, reportes y la impresión de comprobantes.
- Gracias al uso de distintas tecnologías, el aplicativo asegura la compatibilidad con diferentes dispositivos móviles que operan con el sistema operativo Android. Presentándose como un beneficio para las INSTUCIONES FINANCIERAS DE EPS, permitiendo a los agentes de recaudación usar sus propios dispositivos, reduciendo la necesidad de inversión en hardware.
- El uso de la metodología de prototipado durante la elaboración del proyecto permitió una construcción iterativa del sistema, permitiendo la validación temprana de funcionalidades y una retroalimentación constante. Asegurando que el aplicativo final se ajuste a los requerimientos y necesidades operativas de las INSTITUCIONES FINANCIERAS DE EPS.
- El aplicativo implementado proporciona a las INSTITUCIONES FINANCIERAS DE EPS una solución que permite a los socios gestionar sus productos financieros, como cuentas de ahorro y créditos, desde cualquier localización, descartando las limitaciones geográficas que afectan la gestión de cobranza. Adicionalmente, influye en la experiencia del cliente, al momento de brindar acceso a servicios financieros que antes eran inaccesibles para muchos socios, ampliando el alcance de las instituciones fortaleciendo la inclusión financiera.

### 5.2. Recomendaciones

- Realizar entrevistas a agentes recaudadores para recolectar requerimientos, con la finalidad de mejorar aspectos como interfaz, funcionalidades, ingreso de datos, etc. Para poder implementarlas dentro del aplicativo, y así ofrecer una mejor experiencia de usuario, buscando una mejor satisfacción del agente recaudador al momento de realizar acciones de gestión de cobranza.
- Implementar y proponer soluciones que faciliten la entrega de recibos y

comprobantes, para no solo contar con un comprobante físico, sino también contar con elementos de factura electrónica, notificaciones en tiempo real sobre transacciones para que estas sean entregadas tanto a la institución como al socio.

- Capacitar constantemente al agente recaudador en el uso de las funcionalidades del aplicativo, para que pueda concretar las tareas de gestión de cobranza de forma exitosa y en el menor tiempo posible. Reduciendo los errores al proporcionar las indicaciones necesarias para aprovechar al máximo las capacidades del sistema.
- Implementar más opciones de búsqueda y ordenamiento de reportes para una mejor gestión del registro de transacciones, permitiendo filtrar por criterios como el monto, número de cédula del cliente, fecha, etc. Facilitando el acceso a información específica de manera rápida y precisa.

## BIBLIOGRAFÍA

- Alcaraz, R. (2024). *Guía de uso de Figma*. <https://www.figma.com/>.
- Alejandro, D., Bernal, G., Topográfico, I., Sebastián, J., Vega, R., Directora, I. A., María, P. A., & Sevillano, L. (2020). *DESARROLLO DE UNA APLICACIÓN MOVIL “LAND” PARA LA CAPTURA DE VERTICES MEDIANTE LA METODOLOGIA CORINE LAND COVER* Autores.
- Bárbara, M., González, M., Yosvany, I., Perdomo, G., Luis, I. J., & Abreu, R. (2024). *Componente para búsquedas dinámicas mediante estructuras JSON en PostgreSQL para los proyectos de CEGEL*.
- Chong, V. (2025). *Calidad del Software Aizprua, Saulo; Ortega Amable*.
- Chong, L. Von. (2021). *Calidad del Software una Perspectiva Continua Software Quality a Continuous Perspective*. <http://miar.ub.edu/issn/2304->
- Cuadrado, G. C. (22 de Julio de 2022). Visual Studio Code: Editor de código para desarrolladores. Obtenido de OpenWebinars: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- Churata, H., & Fernando, L. (2017). *DESARROLLO DE APLICACIONES MÓVILES HÍBRIDAS CON IONIC* (Vol. 2). [https://commons.wikimedia.org/wiki/File:Ionic\\_](https://commons.wikimedia.org/wiki/File:Ionic_)
- David, M., García, M., María, E. E., & Tomalá, Y. (2020). *4 | No 18 | septiembre-enero*. <http://revistamapa.org/index.php/es>
- en José Efrén Pérez Carmona, M. C. (2008). *“TECNOLOGÍA BLUETOOTH” Tesis para obtener el grado de Ingeniero en Comunicaciones y Electrónica. PRESENTA: C. SERGIO ALBERTO GONZALEZ VERGARA MÉXICO DF, AGOSTO DE 2008. INSTITUTO POLITÉCNICO NACIONAL ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA ELÉCTRICA DEPARTAMENTO DE TITULACIÓN*.
- Enríquez, F., Fierro, S., Flores, B., Imbaquingo Esparza, D., & Michelena, J. (2023). Impacto del patrón modelo vista controlador (MVC) en la seguridad, interoperabilidad y usabilidad de un sistema informático durante su ciclo de vida. *EASI: Ingeniería y Ciencias Aplicadas En La Industria*, 2(1), 11–16. <https://doi.org/10.53591/easi.v2i1.2043>
- Geovanni, M., & Espino, F. (2024). *Corrección de anillo de entradas/salidas y pruebas de antenna y ERC para la definición del flujo de diseño del primer chip con tecnología nanométrica desarrollado en Guatemala*.
- González Campo, C. H., & Lozano Oviedo, J. (2020). Propuesta para la definición de la arquitectura empresarial. *Dimensión Empresarial*, 18((1)). [https://doi.org/10.15665/dem.v18i\(1\).2109](https://doi.org/10.15665/dem.v18i(1).2109)
- Herrera, S. I., Clusella, M. M., Zuaín, S. S., Rocabado Moreno, S. H., Cheein, D., Leturia, F., & Trejo, S. (2024). *Aplicaciones híbridas multiplataforma para Turismo Religioso*.
- Informix, I. (2024). IBM® Informix es una base de datos integrable y de alto rendimiento que permite integrar series temporales y datos espaciales de SQL, NoSQL y JSON. Está disponible en IBM® Cloud Pak for Data. Obtenido de IBM Informix: <https://www.ibm.com/es-es/products/informix>
- Jesús, J., & Caballero, V. (2016). *Modelo de procesos para el desarrollo del front-end de aplicaciones web*.
- José Ramírez López, C., & Hernández Audelo, L. (2020). *Diseño e Implementación de una Función Hash Basada en Caos*.
- Llatas Yovera, B., Humberto Bach Rojas Herrera, L., Jhampier Asesor, A., Ayala, M., & Enrique, A. (2017). *FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO*.
- Luis, J., & Méndez, C. (2013). *UNIVERSIDAD DE CUENCA FACULTAD DE INGENIERÍA ESCUELA DE INFORMÁTICA*.
- Márquez Díaz, J., Sampedro, L., & Vargas, F. (2025). *Instalación y configuración de Apache, un servidor Web gratis*.

- Miguel, R. M. (2025). Android Desarrollo de aplicaciones. Obtenido de Android: [https://books.google.es/books?hl=es&lr=&id=OI5dEAAAQBAJ&oi=fnd&pg=PA3&dq=Android+definicion&ots=NmiMkoUj\\_p&sig=k5WJEnbj4A5he0UKbOuhtNwQhJE#v=onepage&q&f=false](https://books.google.es/books?hl=es&lr=&id=OI5dEAAAQBAJ&oi=fnd&pg=PA3&dq=Android+definicion&ots=NmiMkoUj_p&sig=k5WJEnbj4A5he0UKbOuhtNwQhJE#v=onepage&q&f=false)
- Nelson Cecchi, C., Andrea Braun, L., & Alejandro, G. (2017). *Una Arquitectura Cliente-Servidor para Modelado Conceptual Asistido por Razonamiento Automático*.
- Pucciarelli, L. (2020). Angular TypeScript - Arquitectura Instalación - Directivas y Bindings Forms - Ruteo y más. Obtenido de Google Libros: [https://books.google.es/books?hl=es&lr=&id=6aHTDwAAQBAJ&oi=fnd&pg=PA5&dq=typescript+definicion&ots=MI8OeUNLOy&sig=D8Z\\_C5YJFEaa7jCX7aal0CEkCT8#v=onepage&q=typescript%20definicion&f=false](https://books.google.es/books?hl=es&lr=&id=6aHTDwAAQBAJ&oi=fnd&pg=PA5&dq=typescript+definicion&ots=MI8OeUNLOy&sig=D8Z_C5YJFEaa7jCX7aal0CEkCT8#v=onepage&q=typescript%20definicion&f=false)
- Ramón Arteaga-Macías, I. S. (2017). *Economía popular y solidaria, República del Ecuador Popular and solidary economy, Republic of Ecuador Economía popular e solidária, República do Equador*. 3, 533–542. <https://doi.org/10.23857/dom.cien.pocaip.2017.3.4.oct.533-542>
- Sánchez, A. M. (2021). Universidad Técnica de Ambato. Obtenido de ECONOMÍA POPULAR Y SOLIDARIA EN: [https://obest.uta.edu.ec/wp-content/uploads/2021/10/Economia-popular-y-solidaria-Ecuador.pdf?utm\\_source=chatgpt.com](https://obest.uta.edu.ec/wp-content/uploads/2021/10/Economia-popular-y-solidaria-Ecuador.pdf?utm_source=chatgpt.com)
- Suarez, B. (29 de Mayo de 2017). Cómo validar cédula y RUC en Ecuador. Obtenido de Medium: <https://medium.com/@bryansuarez/cómo-validar-cédula-y-ruc-en-ecuador-b62c5666186f>
- Sql, O. A., & García, A. E. (2010). *Manual Práctico de SQL*.
- Valverde, V., Portalanza, N., & Mora, P. (2024). ANÁLISIS DESCRIPTIVO DE BASE DE DATOS RELACIONAL Y NO RELACIONAL Para citar este artículo puede utilizar el siguiente formato. <https://www.eumed.net/rev/atlante/2019/06/base-datos-relacional.html>
- Villagomez, G., Álvarez, H., Vivanco, D., & Echeverría, F. (2020). *Implementación de la Migración de la Base de Datos del Sistema Canopus de Informix 9.4C a Oracle 10g*.
- Zapata, C. M., & Carlos, A. Á. (2005). CONVERSION OF PROCESSES DIAGRAMS IN USE CASE DIAGRAMS USING ATOM 3. *Año*, 72, 103–113.

# ANEXOS

## ANEXO A: Diccionario de datos.

**Ilustración 46.** Diccionario de la base de datos.

The screenshot shows the Sybase PowerDesigner Physical Data Model HTML Report. The left sidebar contains a tree view of the model structure, including Model information, PDM Diagrams, and Tables. The main content area is titled 'Tables' and 'Model level tables', showing a 'List of tables' with two columns: 'Name' and 'Code'. The table lists various database tables such as Accesos, Clientes, Creditos, Cuentas, Denominaciones, Depositantes, DetallesDenominacion, DetallesCredito, DetallesPago, Efectivo, Empleados, Movimientos, RubrosDividendos, Telefonos, TiposCuentas, and Usuarios.

Name	Code
Accesos	Accesos
Clientes	Clientes
Creditos	Creditos
Cuentas	Cuentas
Denominaciones	Denominaciones
Depositantas	Depositantas
DetallesDenominacion	DetallesDenominacion
DetallesCredito	DetallesCredito
DetallesPago	DetallesPago
Efectivo	Efectivo
Empleados	Empleados
Movimientos	Movimientos
RubrosDividendos	RubrosDividendos
Telefonos	Telefonos
TiposCuentas	TiposCuentas
Usuarios	Usuarios

## ANEXO B: Back-end.

**Ilustración 47.** Código del Back-end.

```
<?php
//namespace App\Model;

class MovilApp_AFOMOVIL
{
    public function getRecibo($data){
        try
        {
            $registros = Array();

            $response = new Response();
            $fechahora = date("Y-n-j H:i:s", strtotime(CoopData::fechahoraservidor()->FECHA));

            if (isset($data['autorizacion']) and isset($data['idesion']) and isset($data['token'])){
                $Mov = new MovimientosData();
                $datos = $Mov->getMovimientoRecibo($data['autorizacion']);
                if (isset($datos)){
                    foreach($datos as $dato)
                    {
                        $registro["codigomcdv"] = $dato->CODIGOMCDV;
                        $registro["transaccion"] = $dato->TRANSACCION;
                        $registro["valor"] = $dato->VALOR;
                        $registro["cuenta"] = $dato->CUENTA;
                        $registro["numero"] = $dato->NUMERO;
                        $registro["oficina"] = $dato->OFICINA;
                        $registro["caja"] = $dato->CAJA;
                        $registro["coope"] = $dato->COOPE;
                        $registro["cedula"] = $dato->CEBULA;
                        $registro["nombres"] = $dato->NOMBRES;
                        $registro["producto"] = $dato->PRODUCTO;
                        $registro["numcliente"] = $dato->NUMCLIENTE;
                    }
                    $registros[] = $registro;
                    $response->result["idesion"] = $data['idesion'];
                    $response->result["fechaservidor"] = $fechahora;
                    $response->result["recibo"] = $registros;
                    $response->SetResponse("000", "Recibo Ok", $data['token']);
                }
            }
            else{
                $response->SetResponse("404", "Datos Incorrectos", null);
            }
        }
        else{
            $response->SetResponse("404", "Datos No Autorizados", null);
        }
        error_log(json_encode($response). "F:". $fechahora. " IP: ". $_SERVER['REMOTE_ADDR']. "\n", 3, $_SERVER['DOCUMENT_ROOT']. "/logs/movil". date('dmY'). ".log");
        return $response;
    }
}
```

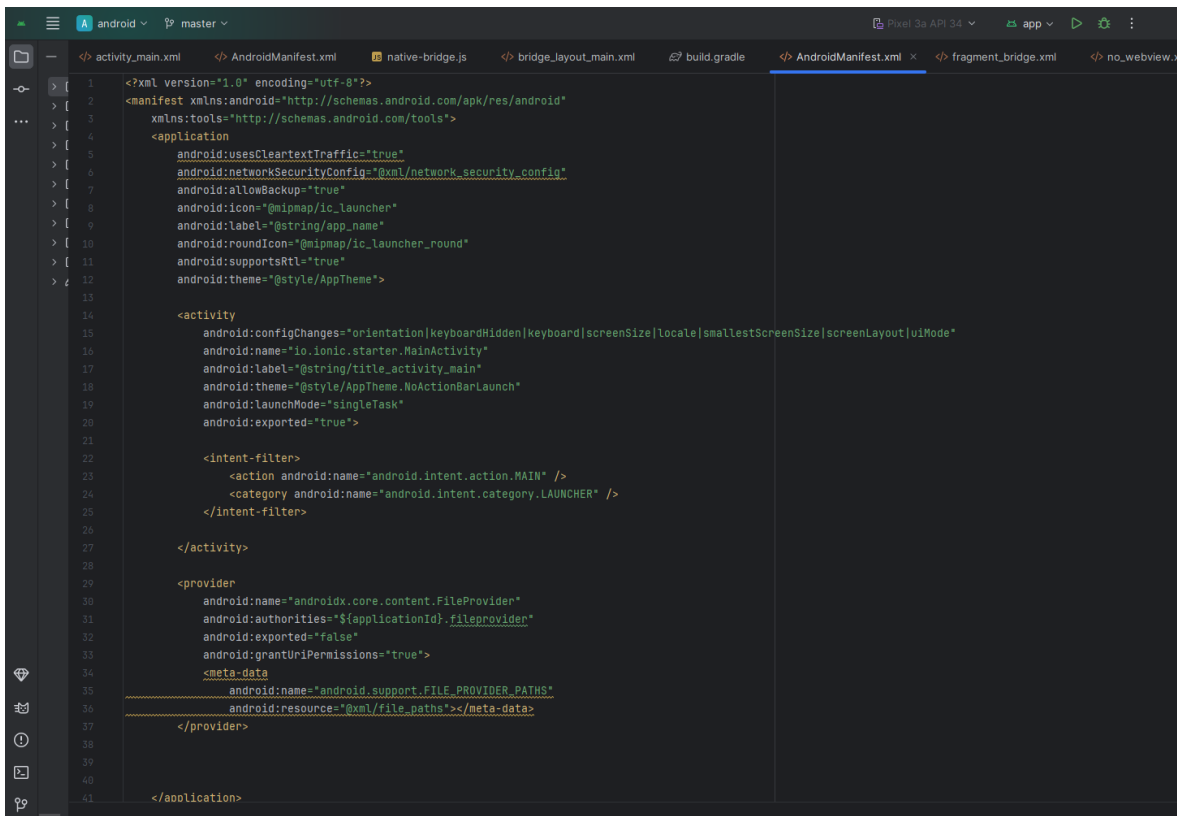
## ANEXO C: Front-end.

## Ilustración 48. Código del front-end.

```
AlcRecaudacion > src > app > login > login.page.html > ion-content:ion-padding
Tulio162, last month | 1 author | Tulio162
1 <ion-content [fullscreen]="true" class="ion-padding">
2   <ion-title size="large" class="login-title">Iniciar Sesión</ion-title>
3
4   <form [formGroup]="loginForm" (ngSubmit)="login()">
5     <ion-item fill="outline" class="custom-input" [class.invalid]="loginForm.get('username')?.invalid && loginForm.get('username')?.touched">
6       <ion-label position="floating">Usuario</ion-label>
7       <ion-input
8         formControlName="username"
9         placeholder="Nombre de usuario"
10        type="text"
11        (ionInput)="convertToUpper($event)">
12     </ion-input>
13   </ion-item>
14   <div *ngIf="loginForm.get('username')?.invalid && (loginForm.get('username')?.dirty || loginForm.get('username')?.touched)" class="error-message">
15     <small *ngIf="loginForm.get('username')?.errors?.['required']">Por favor, ingrese su nombre de usuario</small>
16     <small *ngIf="loginForm.get('username')?.errors?.['pattern']">No se permiten números ni caracteres especiales</small>
17   </div>
18
19   <ion-item fill="outline" class="custom-input" [class.invalid]="loginForm.get('userpass')?.invalid && loginForm.get('userpass')?.touched">
20     <ion-label position="floating">Contraseña</ion-label>
21     <ion-input
22       formControlName="userpass"
23       placeholder="*****"
24       type="password">
25   </ion-input>
26 </ion-item>
27 <div *ngIf="loginForm.get('userpass')?.invalid && (loginForm.get('userpass')?.dirty || loginForm.get('userpass')?.touched)" class="error-message">
28   <small *ngIf="loginForm.get('userpass')?.errors?.['required']">Por favor, ingrese su contraseña</small>
29   <small *ngIf="loginForm.get('userpass')?.errors?.['minlength']">Mínimo 6 caracteres</small>
30 </div>
31 <ion-button expand="full" type="submit" class="custom-button" shape="round" [disabled]="!loginForm.valid" style="--background: var(--ion-color-blue);">Ingresar</ion-button>
32 </form>
33 </ion-content> Tulio162, last month • Se agrega el Proyecto
34
```

## Anexo D: Generación de archivo APK.

### Ilustración 49. Generación del archivo APK en Android Studio.



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <application
        android:usesCleartextTraffic="true"
        android:networkSecurityConfig="@xml/network_security_config"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|locale|smallestScreenSize|screenLayout|uiMode"
            android:name="io.ionic.starter.MainActivity"
            android:label="@string/title_activity_main"
            android:theme="@style/AppTheme.NoActionBarLaunch"
            android:launchMode="singleTask"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <provider
            android:name="androidx.core.content.FileProvider"
            android:authorities="${applicationId}.fileprovider"
            android:exported="false"
            android:grantUriPermissions="true">
            <meta-data
                android:name="android.support.FILE_PROVIDER_PATHS"
                android:resource="@xml/file_paths"></meta-data>
        </provider>
    </application>
```

**Anexo E: Cuestionario de pruebas de usabilidad.**

**Ilustración 50. Cuestionario post-test para pruebas de usabilidad.**

