	<b>GESTIÓN DE SERVICIOS ACADÉMICOS Y BIBLIOTECARIOS</b>		<b>CÓDIGO</b>	FO-GS-15	
			<b>VERSIÓN</b>	02	
	<b>ESQUEMA HOJA DE RESUMEN</b>			<b>FECHA</b>	03/04/2017
				<b>PÁGINA</b>	1 de 1
<b>ELABORÓ</b>		<b>REVISÓ</b>	<b>APROBÓ</b>		
Jefe División de Biblioteca		Equipo Operativo de Calidad	Líder de Calidad		

## RESUMEN TRABAJO DE GRADO

AUTOR:

NOMBRES(S): MOISES ENRIQUE

APELLIDOS: VILLAMIZAR CABALLERO

FACULTAD: INGENIERÍA

PLAN DE ESTUDIO: INGENIERÍA ELECTRÓNICA

DIRECTOR:

NOMBRES(S): ELMER ALEJANDRO

APELLIDOS: PARADA PRIETO.

CODIRECTOR:

NOMBRES(S): JAIR ANGEL

APELLIDOS: NOVA ACEVEDO.

TÍTULO DEL TRABAJO (TESIS): DISEÑO Y PROGRAMACIÓN DE APLICATIVO MÓVIL DE RECARGA DE TARJETA NFC PARA LA EMPRESA RECAUDO BOGOTÁ SAS

RESUMEN

Este proyecto de grado tiene como propósito el desarrollo de una aplicación Mobile para un dispositivo pos con sistema operativo Android, por medio del cual se interactúa a través de esta aplicación con los distintos sensores y sistema del datafono, con el fin de realizar lectura y escritura de datos en tarjetas NFC y conectando con servidores de recaudos Bogotá para la verificación de datos y solicitud de recargas de cada tarjeta, esto sirve para el uso del transporte urbano en la ciudad de Bogotá.

PALABRAS CLAVE: DATAFONO, ANDROID, APLICACIÓN MOVIL,SENSORES, TARJETAS NFC

PÁGINAS: 69

PLANOS: 0

ILUSTRACIONES: 30

CD ROOM: 0

DISEÑO Y PROGRAMACIÓN DE APLICATIVO MÓVIL DE RECARGA DE  
TARJETA NFC PARA LA EMPRESA RECAUDO BOGOTÁ SAS

MOISES ENRIQUE VILLAMIZAR CABALLERO

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER  
FACULTAD DE INGENIERÍA  
PLAN DE ESTUDIOS DE INGENIERÍA ELECTRÓNICA  
SAN JOSÉ DE CÚCUTA

2021

DISEÑO Y PROGRAMACIÓN DE APLICATIVO MÓVIL DE RECARGA DE  
TARJETA NFC PARA LA EMPRESA RECAUDO BOGOTÁ SAS

MOISES ENRIQUE VILLAMIZAR CABALLERO

DIRECTOR:

MSc. IE. ELMER ALEJANDRO PARADA P.

CODIRECTOR

ING. JAIR ANGEL NOVA ACEVEDO

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERÍA

PLAN DE ESTUDIOS DE INGENIERÍA ELECTRÓNICA

SAN JOSÉ DE CÚCUTA

2021

## ACTA DE SUSTENTACIÓN DE UN TRABAJO DE GRADO

Fecha: CÚCUTA, 02 DE NOVIEMBRE DE 2021

Hora: 16:00

Lugar: MODALIDAD REMOTA SINCRÓNICO

Plan de Estudios: INGENIERÍA ELECTRÓNICA

Título de la Tesis: "DISEÑO Y PROGRAMACIÓN DE APLICATIVO MÓVIL DE RECARGA DE TARJETA NFC PARA LA EMPRESA RECAUDO BOGOTÁ S.A.S."

Jurados: IE. ES<sup>p</sup> LUIS FERNANDO BUSTOS MÁRQUEZ  
IE. MSc MARIO JOAQUIN ILLERA BUSTOS

Director: IE. MSc. ELMER ALEJANDRO PARADA PRIETO

Codirector: Ing. JAIR ANGEL NOVA ACEVEDO

Nombre del Estudiante	Código	Calificación
MOISÉS ENRIQUE VILLAMIZAR CABALLERO	1161326	4.4 (Cuatro, cuatro)

### APROBADA

  
\_\_\_\_\_  
LUIS FERNANDO BUSTOS MÁRQUEZ

  
\_\_\_\_\_  
MARIO JOAQUÍN ILLERA BUSTOS

  
\_\_\_\_\_  
ANGELO JOSEPH SOTO VERGEL  
Coordinador Comité Curricular  
Ingeniería Electrónica

## *Dedicatoria*

*Dedicado a Dios, el dador de la vida,  
la sabiduría y mi motivación para continuar.  
por ser la luz que alumbra siempre mi camino  
y por permitirme cumplir este propósito de ser*

*Ingeniero electrónico*

*Moisés Villamizar.*

## **Agradecimientos**

Primeramente, agradecer a Dios quien me da la oportunidad y el privilegio de poder materializar todos mis sueños.

A mi esposa, por ser un apoyo durante toda esta etapa y por la fuerza que me ha brindado para continuar.

A mis Padres y abuela, quienes fueron los que me impulsaron a seguir mis ideales y por toda la ayuda que me brindaron en todo este tiempo.

A la Universidad Francisco de Paula Santander por ser mi alma mater, formarme como profesional y ser la institución que me permite escalar un peldaño más.

A todo el cuerpo docente del programa ingeniería electrónica en especial a mi tutor Elmer Alejandro Parada por brindar todo su conocimiento para hacer grandes profesionales.

Y a todas las personas que de una u otra forma hicieron parte de este proceso de formación.

A todos mil y mil gracias

## Tabla de contenido

Introducción	10
1. Título	11
2. Planteamiento Del Problema	12
3. Justificación	14
3.1 Beneficios Tecnológicos	14
3.2 Beneficios Económicos	14
3.3 Beneficios Sociales	15
4. Alcances	16
4.1 Tipo De Proyecto	16
4.2 Resultados Obtenidos	16
5. Limitaciones Y Delimitaciones	17
5.1 Limitaciones	17
5.2 Delimitaciones	17
6. Objetivos	18
6.1 Objetivo General	18
6.2 Objetivos Específicos	18
7. Marco Referencial	19
7.1 Antecedentes	19
7.2 Marco Teórico	21
7.3 Marco Legal	28
8. Diseño Metodológico	30
8.1 Cronograma	36
8.2 Presupuesto	37
9. Resultados	41
9.1 Análisis de requerimientos	41
9.2 Diseño de interfaces	42
9.3 Implementación de librerías	46
9.4 Conexión con Apicore	48
9.5 Configuración de inicialización	48
9.6 Desarrollo de modulo para configuración	51
9.7 Lógica para consumo de servicios	57
9.8 Configuración de respuestas al usuario	59
9.9 Testing y solución de incidencias	62
10. Conclusiones	66
11. Referencias	68

## Lista de tablas

Tabla 1.	Cronograma de actividades	36
Tabla 2.	Presupuesto global del proyecto por fuentes de financiación.	37
Tabla 3.	Descripción de los gastos de personal (en miles de \$).	38
Tabla 4.	Descripción de los equipos de uso propio (en miles de \$).	39
Tabla 5.	Descripción de materiales y suministros	40
Tabla 6.	Mensajes de usuario.	61
Tabla 7.	Tabla de pruebas.	62
Tabla 8.	Prueba de tiempos.	65



## Lista de figuras

Figura 1.	NEWPOSS K2.	22
Figura 2.	Logo Android	23
Figura 3.	Diagrama de flujo funcional.	30
Figura 4.	Flujo para recarga exitosa.	34
Figura 5.	Partes del aplicativo.	35
Figura 6.	Diagrama de flujo del proyecto.	41
Figura 7.	Inicio Clave.	42
Figura 8.	Parámetros inicialización.	43
Figura 9.	Pantalla de bloqueo.	43
Figura 10.	Barra de progreso.	44
Figura 11.	Mensaje informativo.	44
Figura 12.	Tarjeta de configuración y de usuario.	45
Figura 13.	Menú.	45
Figura 14.	Interfaces del menú.	46
Figura 15.	Librerías del Gradle.	47
Figura 16.	Diagrama secuencial	48
Figura 17.	SIM y SAM.	49
Figura 18.	Demo.	50
Figura 19.	Secuencia de gráficos.	51
Figura 20.	Módulo de configuración.	52
Figura 21.	Configuración del sistema Fecha-Hora.	53
Figura 22.	Interfase parámetros operativos.	53
Figura 23.	Interfase saldo SAM.	54
Figura 24.	Interfase Backup.	54
Figura 25.	Interfase eliminar archivos.	55
Figura 26.	Ajustes del sistema.	55
Figura 27.	Modo kiosko.	56
Figura 28.	Interfase parámetros de red.	56
Figura 29.	Secuencia de servicios.	57
Figura 30.	Secuencia de transacción.	60

## Introducción

La ingeniería es una ciencia que se encarga de darle soluciones a problemas que se detectan en su entorno, en el caso de la movilidad se ha creado sistemas de transporte que permite la movilización de los usuarios a través de buses en la ciudad de manera más eficaz y ordenada como lo es el SITP gracias a su estructura vial.

Por lo tanto, para su uso es necesario el pago de este servicio, lo que requiere un sistema de pago rápido y seguro que garantice eficacia y comodidad a la hora de realizarlo. En este trabajo se presenta un dispositivo móvil para pagos, en específico el NEWPOS K2, que a través de la detección y lectura de una tarjeta NFC que contiene un microcontrolador, permite enviar información encriptada, por medio de una aplicación Móvil llamada APICORE, que corre en segundo plano y desarrollado por la empresa recaudos Bogotá SAS. Del mismo modo, se encarga de detectar de manera segura y validar la información de esa tarjeta ofreciendo los datos necesarios para que la aplicación Android desarrollada en este trabajo de grado pueda establecer conexión por medio de la API con el servidor para realizar los pagos. En ese sentido, el fin del proyecto es establecer una conexión segura entre el dispositivo NEWPOS K2, el APICORE y el servidor web de recaudos por medio de una aplicación móvil en Android para recargar saldo a la tarjeta NFC.

## **1. Título**

DISEÑO Y PROGRAMACIÓN DE APLICATIVO MÓVIL DE RECARGA DE  
TARJETA NFC PARA LA EMPRESA RECAUDO BOGOTÁ SAS

## 2. Planteamiento Del Problema

Para la actualización y sofisticación del sistema de transporte urbano de la ciudad de Bogotá, El operador Recaudos Bogotá junto al SITP han buscado la implementación de mejores tecnologías para la integración de diferentes medios de pago para el servicio del SITP, con el fin de facilitar la recarga de las tarjetas por parte del usuario. Actualmente recaudos cuentan con 5287 puntos de recargo activos distribuidos a lo largo de la ciudad. (Transmilenio S.A, 2021)

Como hay un horario para recargar en todos los puntos, hay congestión en horas pico, la inseguridad en ciertas partes de la ciudad dificulta el acceso a estos puntos, en algunos establecimientos la conectividad a internet de los dispositivos presenta dificultades, incluso en horas valle, muchos usuarios se les dificulta apartar tiempo y esfuerzo para tomar carretera e ir hacia un punto de recargo ya que tienen otras obligaciones por cumplir, por consiguiente, es evidente que se necesita un mejor sistema, eficaz y ágil por el cual se satisfagan estas necesidades. (Oficina de Análisis de Información y Estudios Estratégicos, 2019)

Buscando una solución se implementó el sistema de pagos de forma virtual, a través de una plataforma de pago como lo es PSE (Pagos Seguros en Línea) que cobra \$600 por transacción o por la plataforma TuyaPay que es gratuito. Para esto primero se debe ingresar al portal de pagos antes mencionado, crear una cuenta para realizar la transacción hacia la tarjeta. Una vez se obtiene la cuenta se procede a ingresar los datos de la tarjeta ya adquirida, se ingresa el monto deseado y listo, finaliza la transacción. Solo se deberá salir a actualizar el estado de la ficha el día que se necesite tomar el servicio para actualizar el estado de la tarjeta.

Se evidencia por parte de los usuarios del servicio del SITP, que las horas de difícil acceso de estos puntos se encuentran entre las 5 a.m. a 8 a.m. y de 5 p.m. a 7 p.m. ya que corresponde a las horas de mayor flujo de usuarios que toman el servicio de Transmilenio, este comienza a funcionar desde 4:30 am hasta las 10:00 pm, sin embargo, en los puntos de recargo el tiempo por transacción en la recarga a la tarjeta tu llave puede ser de 2 min a 4 min, dependiendo de la acción que realiza el operario (recibe la tarjeta, revisa si tiene recarga web, si no, solicita el monto a recargar, pide el efectivo, realiza la recarga y entrega vueltas si es necesario).

Actualmente, el problema radica en que la demanda creciente por el servicio, desborda la capacidad de los puntos de recargos instalados, por esto, se desea aumentar su cantidad al igual que su eficiencia por medio del uso de otros mecanismos compatibles con la tecnología actual, por consiguiente, se genera la siguiente pregunta:

¿Cómo se pueden integrar más dispositivos autónomos para puntos de recarga que descongestionen las horas pico las estaciones del SITP?

### **3. Justificación**

Este proyecto tiene como propósito aportar a la problemática que viven miles de usuarios diariamente para recargar su tarjeta Tullave, es por esto que la empresa implementara 500 dispositivos Newposs K2 con este aplicativo, distribuyéndolo estratégicamente en la ciudad de Bogotá aumentando los puntos de recargo, de esta manera este proyecto trae los siguientes beneficios:

#### **3.1 Beneficios Tecnológicos**

Se desarrolló una aplicación Android para el datáfono modelo NEWPOSS K2 que permitirá consultar un API del servidor de Recaudos Bogotá para que el dispositivo detecte, lea y cargue la nueva información o saldo en la tarjeta al ponerla en frente, esta incluye la tecnología NFC por medio de la norma ISO 14443 para la transmisión encriptada de datos. Lo que se tarda un punto de recargo actual en recargar la tarjeta es de 2 minutos aproximadamente, al implementar esta tecnología disminuirá a 15 segundos aproximadamente desde que el usuario pone su tarjeta sobre el Newposs K2 hasta que la retire, esto será lo que tarde el sistema en depositar el nuevo saldo sobre la tarjeta Tullave.

#### **3.2 Beneficios Económicos**

La aplicación para la actualización del saldo de la tarjeta NFC funciona de manera automática en el dispositivo, para crear más puntos de recargo en las estaciones del SITP que funcionen sin necesidad de la intervención de un operario contribuye la reducción del presupuesto en empleados para este cargo. Por otro lado, implementar esta tecnología incentiva a los usuarios a utilizar más el servicio por la facilidad del

recargo lo que se vería reflejado en mayor ingreso para la empresa.

### **3.3 Beneficios Sociales**

Actualmente Recaudos Bogotá cuenta con plataformas de pagos online para que sus usuarios puedan recargar la tarjeta Tullave, pero es necesario llevar la tarjeta a los puntos de recargo que son locales comerciales que se encuentran estratégicamente ubicados en la ciudad de Bogotá, o realizar la recarga en las estaciones de Transmilenio. Los puntos comerciales prestan su servicio en los horarios definidos por el dueño, en las estaciones los usuarios suelen hacer cola en horas pico ya que se hace frecuente el uso del sistema. Según estudios de Recaudos Bogotá, el mes de febrero del presente año cerca de 50.407.721 validaciones (uso del servicio) fueron efectuadas en el horario antes mencionado. (Transmilenio S.A, 2021).

Al implementar las Newposs K2 en las estaciones reduce las colas de usuarios agilizando el proceso de recarga en horas pico para algunas estaciones de Transmilenio. De la misma manera, más establecimientos se verán favorecidos ya que la red de puntos de recargo abarca más zonas, incentivando las visitas a sus establecimientos favoreciendo el comercio.

## 4. Alcances

Este aplicativo es instalado dispositivos datafonos NEWPOS K2 que están ubicados en diferentes puntos de ingreso al Transmilenio, para que todo usuario del SITP en la ciudad de Bogotá pueda ingresar su saldo recargado en el sistema de pago virtual hacia su tarjeta de proximidad al posicionarla en frente del dispositivo.

### 4.1 Tipo De Proyecto

El proyecto fue de carácter investigativo ya que estuvo encaminado al desarrollo de nuevos conocimientos basándose en los procedimientos ya existentes por diferentes autores para ciertos puntos del desarrollo de la aplicación Android, por eso fue necesario tener una metodología aplicada en el desarrollo del proyecto.

### 4.2 Resultados Obtenidos

Los resultados que se obtuvieron de este proyecto constan de las siguientes funcionalidades:

- La aplicación establece conexión con el servidor de Recaudos Bogotá para consultar y validar recargas pendientes por parte del usuario.
- La aplicación se reinicia diariamente a las 4am.
- El Apicore se comunica con esta aplicación.
- La aplicación es intuitiva y autónoma, incentiva al usuario a que debe hacer por medio de las interfaces y pantallas de dialogo.
- Integra un menú el cual puede ser accedido por los operarios de Recaudos Bogotá.



## 5. Limitaciones Y Delimitaciones

### 5.1 Limitaciones

Para el desarrollo del proyecto se necesitó un software específico llamado Android Studio con licencia de prueba gratuita que provee las herramientas necesarias para cumplir con los requerimientos del cliente.

El uso inadecuado, mal cuidado y manipulación indebida de la tarjeta puede incurrir en problemas de lectura en el sensor por parte del dispositivo, por lo que interfiere en la transacción eficaz del dispositivo hacia la tarjeta.

De la misma manera la conexión a internet es un factor importante para el buen funcionamiento del aplicativo.

### 5.2 Delimitaciones

**ESPACIAL:** Esta aplicación fue desarrollada en las instalaciones de WPOSS en la sede en Cúcuta DMA.

**TEMPORAL:** El proyecto se desarrolló en 4 meses.

**ELEMENTOS:** se instala en el sistema operativo K2 5.0 cargado en los 500 datafonos modelo NEWPOS K2 que cuentan con ranura SIMCARD para establecer conexión a internet, una ranura SAM para el acceso seguro a los datos del servidor y un sensor que detecta por medio del protocolo ISO 14443 (RF) las tarjetas de los usuarios.

## **6. Objetivos**

### **6.1 Objetivo General**

Desarrollar un aplicativo Android que interconecte la aplicación APICORE con el servidor web de Recaudos Bogotá para la recarga de saldo en tarjetas NFC por medio de un datafono NEWPOS K2.

### **6.2 Objetivos Específicos**

- Diseñar las interfaces gráficas que contiene la aplicación para recaudos Bogotá.
- Implementar la conexión entre el aplicativo de recargas web con la aplicación APICORE.
- Desarrollar el consumo de servicios para realizar la recarga web.

## 7. Marco Referencial

### 7.1 Antecedentes

**TÍTULO:** Desarrollo de una aplicación de pago a través de la tecnología NFC.

El objetivo de este proyecto fue desarrollar un aplicativo móvil para simular funciones que permita por medio de la tecnología NFC el pago de bienes y servicios de una tienda. Para esto, utilizaron el lenguaje de programación Java, permitiendo crear entorno del aplicativo como lo es la entidad bancaria en la cual se deposita los pagos. (Campa Ruiz, 2011).

**TÍTULO:** modelo de negocio b2c alternativo para la adquisición al medio de pago de los usuarios del sistema integrado de transporte público con tecnologías de e-commerce.

Este proyecto fue una investigación estructurada, para evaluar el problema que vivió el Sistema Integrado de Transporte Público de la ciudad de Bogotá en el 2018 para cubrir las necesidades de los usuarios, gracias al ineficiente modelo de acceso a los medios de pago, para eso se diseñó un modelo de negocio B2C alternativo, para la adquisición del medio de pago de los usuarios del SITP con tecnologías de m-commerce, para contribuir a su eficiencia y potencializar los efectos del uso del Smartphone; aprovechando las tendencias tecnológicas como NFC “Near Field Communication”. El resultado de la investigación permitió ofrecer a los usuarios frecuentes del SITP un servicio más personalizado, únicamente para que el acceso al sistema de pago. (Hernandez Medina, 2016).

**TÍTULO:** Contraste de los riesgos valorados para los tipos de tarjetas que han sido utilizadas como medio de pago en el sistema integrado de transporte público.

El objetivo de este proyecto consistió en evaluar el riesgo de las tarjetas utilizadas en el ingreso a Transmilenio entre las cuales se estudiaron: la tarjeta monedera, cliente frecuente y la actual tarjeta tu llave bajo la licitación de recaudo Bogotá. Gracias a diferentes sucesos presentados, vulneración del medio de pago del sistema de transporte, se realizó una valoración de riesgos por medio de la norma ISO 27005 enfocada a seguridad de la información de activos, basándose en la identificación de amenazas, vulnerabilidades y riesgos de las tarjetas utilizadas en el sistema. (Mojica Sanchez & Andres, 2019).

**TÍTULO:** Diseño de dispositivo de pago electrónico utilizando tarjetas NFC con autenticación biométrica.

Este trabajo presenta el diseño de un dispositivo electrónico de pago que utiliza tarjetas NFC para el saldo y autenticación (identificación) por medio del rasgo biométrico de la huella digital, en lo cual se usa las tecnologías RFID y NFC, los tipos de sistemas biométricos existentes y sus aplicaciones. El objetivo es desarrollar el sistema de pago integrando estas tecnologías. (Rodriguez Mayen, 2017).

**TÍTULO:** Desarrollo de una guía para la implementación de aplicaciones basadas en NFC.

Este proyecto consiste en un acercamiento a las características y especificaciones de NFC y la implementación de su uso con ejemplos de dicha tecnología sobre diferentes plataformas como lo son Python, Arduino, Raspberry

Pi, Android y Java. Explica en qué consiste cada tecnología, cuáles son sus particularidades, sus modos de funcionamiento y como las etiquetas NFC pueden ser utilizadas para realizar diversas acciones de manera automática. (Albiña Martínez, 2016).

## 7.2 Marco Teórico

En los siguientes ítems se presentan las definiciones y conceptos relacionados con los temas más relevantes en los que se fundamenta el proyecto.

**DATÁFONO:** es un dispositivo móvil que es utilizado comúnmente para sistemas de pago ya que dispone los siguientes elementos:

- Lector de tarjetas de crédito
- Teclado numérico para la entrada de datos
- Pantalla para visualizar los datos.
- Impresora alfanumérica para imprimir recibos.

Con esto se puede realizar transacciones en tiempo real entre el cliente y el Comerciante. (Huidobro Moya & Luque Ordoñez, 2014).



Figura 1. NEWPOSS K2.

**TECNOLOGÍA NFC:** Es una tecnología que se emplea para establecer conexiones inalámbricas muy rápidamente hacia ambos sentidos, de muy corto alcance, y se basa en una tecnología de radiofrecuencia que permite a un dispositivo leer pequeñas cantidades de datos de otros dispositivos o etiquetas cuando se aproximan el uno al otro. La tecnología NFC surge pues a partir de una combinación de identificación sin contacto físico o RFID y tecnologías de interconexión. (Albau & Riera, 1992).

**SIM CARD:** La tarjeta SIM o Subscriber Identity Module es una pequeña tarjeta de plástico que tiene un chip pegado a ella, y que tienes que insertar en tu teléfono móvil o smartphone. En este chip, almacena de manera segura tu número de teléfono, así como las claves de acceso de un usuario concreto en una operadora de telefonía. (Meyes, 2008).

**SAM:** Es un módulo hardware preparado por el proveedor del sistema de recaudo que se utiliza en los lectores de tarjetas sin contacto (lectores de escritorio o antenas, lectores del sistema de recarga, lectores del sistema de personalización, etc.). El módulo SAM es una memoria segura y soporta funciones criptográficas (cifrado, cuenta de firmas electrónicas, verificación de firmas electrónicas etc.) que sirven para aumentar la seguridad de los sistemas de recaudo con tarjetas sin contacto. (Uwe Hansmann, 2002).

**ANDROID:** Es un sistema operativo móvil diseñado para dispositivos móviles con pantalla táctil como teléfonos inteligentes o tablets, pero que también lo encontramos en otros dispositivos como relojes inteligentes, televisores o incluso en los sistemas multimedia de algunos modelos de coches. Un sistema operativo desarrollado por Google y basado en el Kernel de Linux y otros softwares de código abierto y que se ha convertido en el principal responsable de la popularización de muchos dispositivos inteligentes por el hecho de facilitar el uso de una gran cantidad de aplicaciones de forma sencilla. (Montero Miguel, 2012).

**REST API:** (Interfaz de programación de aplicaciones). Es un conjunto de requisiciones que permite la comunicación de datos entre aplicaciones.



Figura 2. Logo Android

**Fuente:** Desarrollo de aplicaciones para Android. (Montero Miguel, 2012).

Para eso, la API utiliza requisiciones HTTP responsables de las operaciones básicas necesarias para la manipulación de datos.

Las principales solicitudes son:

**POST:** crea datos en el servidor;

**GET:** lectura de datos en el host;

**DELETE:** borra la información;

**PUT:** registro de actualizaciones.

Rest, que es la abreviación de Representational State Transfer, es un conjunto de restricciones que se utilizan para que las solicitudes HTTP cumplan con las directrices definidas en la arquitectura. (Hooda, 2019).

**SISTEMA DE FACTURACIÓN POS:** es un conjunto de herramientas que tiene dos componentes principales, por un lado, un hardware conformado por equipos de cómputo como lo son computadores de escritorio, cajas registradoras, lectores de códigos de barras, impresoras térmicas, display de 3 líneas, datafonos, pantallas táctiles, balanzas grameras, entre otros. Por otro lado, se encuentra el software, que en el mercado los hay de varios tipos con prestaciones diferentes pero que deben contar con un mínimo funciones básicas como lo son:

registrar las ventas en forma de factura, permitir varias formas de pago, generar informes y estadísticas, definir grupos de artículos, cotizaciones, gestionar costos, impuestos y precios de los productos, entre otros. (Lee J. Krajewski, 2013).



**APLICACIÓN MÓVIL:** Es un programa diseñado para ser ejecutados en teléfonos, tablets y otros dispositivos móviles, que permiten al usuario realizar actividades profesionales, acceder a servicios, mantenerse informado, entre otro universo de posibilidades.

Se tiene conocimiento que las primeras aplicaciones se vislumbraban a finales de los 90. No estamos hablando de apps para teléfonos inteligentes, sino para los analógicos. Estas también son consideradas aplicaciones. La agenda, juegos como el famoso snake, el Tetris, los editores de tonos de llamadas, herramientas para personalizar el teléfono, etc.... cumplían funciones muy básicas comparando con lo que tenemos en la actualidad, sin embargo para cuando salieron significaron un avance enorme en la forma en como veíamos a los teléfonos celulares más antiguos (bloques) y abrieron un mercado gigantesco, cuya competencia es, y sigue siendo tan voraz; que nos ha permitido disfrutar de herramientas cada vez más prácticas, útiles e increíbles. (Montero Miguel, 2012).

**APICORE:** Es una aplicación Android que tiene como objetivo la detección y lectura de las tarjetas que se van a utilizar para ser recargadas implementando la tecnología NFC, por lo que de manera encriptada lee y escribe los datos en la tarjeta, para que la aplicación móvil a desarrollar no pueda manipular directamente esa información. Este aplicativo es entregado por parte de la empresa Recaudos Bogotá y tiene como propósito posibilidad de escribir y leer datos sobre las tarjetas a recargar ya que esa empresa tiene el modelo en específico de las tarjetas de manera que no se puedan falsificar, esta

aplicación no tiene interfaz gráfica, por lo que corre en segundo plano en el dispositivo, trabajando en conjunto con la aplicación a desarrollar y el servidor de recaudos Bogotá.

**LIBRERÍA:** Las librerías son conjuntos de funciones que están disponibles en los programas, cuando se les incluye. Estas están pensadas con el fin de realizar operaciones que, de otra forma, deberían implementarse de nuevo completamente. (Iván, 2017)

**TESTING:** El testing es una actividad desarrollada para evaluar la calidad del producto, y para mejorarlo al identificar defectos y problemas. El testing de software consiste en la verificación dinámica del comportamiento de un programa sobre un conjunto finito de casos de prueba, apropiadamente seleccionados a partir del dominio de ejecución que usualmente es infinito, en relación con el comportamiento esperado. (Gallardo, 2009)

**SERVIDOR:** un servidor provee servicios a las aplicaciones o clientes y estos solicitan dichos servicios para poder realizar el trabajo por el que han sido programado. (M., 2002)

**PROTOCOLO:** Un protocolo de comunicación es el conjunto de normas que definen el formato y orden de los mensajes intercambiados entre dos o más entidades que se comunican entre sí, así como el conjunto de acciones que se toman durante la transmisión y la recepción de estos mensajes. (Jesús, 2010)

**JSON:** (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para

humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos. (Albau & Riera, 1992)

**TOKEN:** También llamado componente léxico es una cadena de caracteres que tiene un significado coherente en cierto lenguaje de programación. Token de seguridad, utilizado para facilitar el proceso de autenticación de usuarios. (M., 2002)

**AIDL:** El Lenguaje de definición de la interfaz de Android (AIDL) es similar a otros IDL con los que posiblemente hayas trabajado. Te permite definir la interfaz de programación que el cliente y el servicio acuerdan utilizar para establecer comunicación entre procesos (IPC). Por lo general, en Android, un proceso no puede acceder a la memoria de otro proceso. Por esto, para comunicarse, tienen que descomponer sus objetos en primitivas que el sistema operativo pueda comprender y tienen que ordenar los objetos entre esos límites para ti. Como es tedioso escribir el código para ese ordenamiento, Android se encarga de eso por ti con AIDL. (Montero Miguel, 2012)

### 7.3 Marco Legal

Este proyecto está relacionado en un marco de normas, requisitos y decretos que regulan temas de interés que se abordan en su desarrollo.

Como el comercio electrónico es de carácter mundial por su propia naturaleza, y abarca una amplia gama de actividades en su mayoría totalmente nuevas. Impulsando la revolución del internet, se expande aceleradamente y experimenta cambios radicales incluyendo nuevos negocios, mercados y organismos comerciales innovadores, en definitiva, se están creando modernas funciones y nuevas fuentes de ingresos. Bajo la denominación de comercio electrónico indirecto (pedido electrónico de bienes tangibles) como el directo (entrega en línea de bienes intangibles).

Es por esto, que la siguiente ley fomenta y regula el acceso de datos y el comercio electrónico:

Por medio de la cual se define y reglamenta el acceso y uso de los mensajes de datos, del comercio electrónico y de las firmas digitales, y se establecen las entidades de certificación y se dictan otras disposiciones. Ley 527 de 1999. (Colombia, Congreso de la Republica, 1999).

De la misma manera, la Superintendencia protege su derecho fundamental de Habeas Data (conocer, actualizar y rectificar sus datos). Por lo que la siguiente ley, es la encargada de la protección de datos personales financieros la cual incentiva el desarrollo de este tipo de aplicaciones móviles.

La Ley 1266 de 2008, también conocida como Ley de Habeas Data, se aplica a todos los datos personales financieros, crediticios, comerciales y de servicios registrados en un banco de datos. En este sentido, la aplicación de la Ley 1266 de 2008 está encaminada a regular el uso de esa información y por tanto otro tipo de datos (por ejemplo, aquellos mantenidos en un ámbito exclusivamente personal o doméstico o los que se incluyen en una historia clínica) se encuentran excluidos de la aplicación de esta norma. (Colombia. Congreso de la republica., 2008).

## 8. Diseño Metodológico

Los servicios web se deben consumir según el flujo ilustrado en la figura 3 para realizar el proceso de recarga en una tarjeta Tullave. Las cajas en color azul se refieren a los servicios que se deben consumir para llevar a cabo la recarga de saldo en la tarjeta.

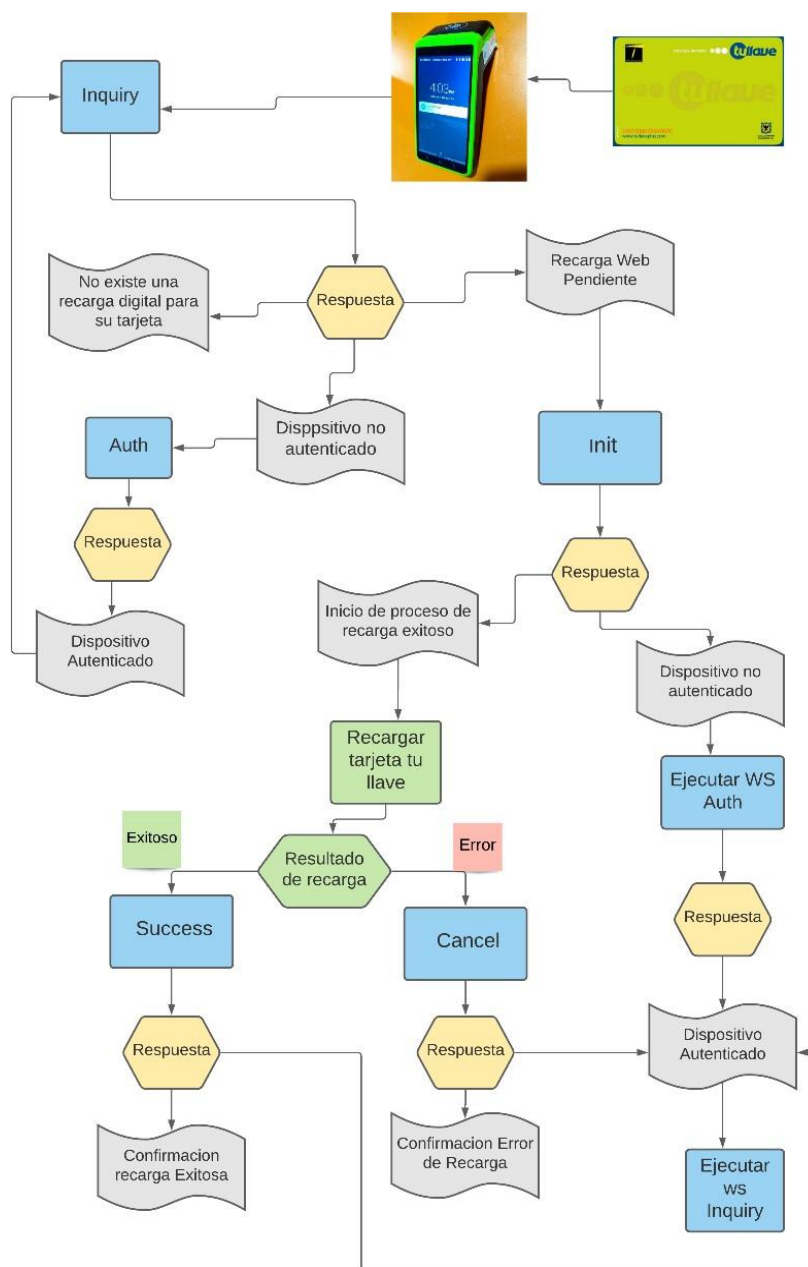


Figura 3. Diagrama de flujo funcional.

Para dar cumplimiento a los objetivos planteados en el presente documento, se presenta a continuación las actividades y metodologías como procedimiento a seguir.

### **Objetivo 1:**

Diseñar las interfaces gráficas que contiene la aplicación para recaudos Bogotá.

- Análisis de requerimientos

**Metodología:** se analizarán los requerimientos por parte de la empresa contratante con el fin de establecer posibles soluciones en el aplicativo.

- Diseño de solución

**Metodología:** se realizará la interfaz gráfica del usuario de acuerdo a la petición del cliente, usando la herramienta para desarrollo de aplicaciones móviles Android Studio en la sección de “Layout” o vistas.

### **Objetivo 2:**

Implementar la conexión entre el aplicativo de recargas web con la aplicación APICORE.

Para este objetivo se deberá implementar la conexión entre el aplicativo de recargas web con la aplicación de seguridad APICORE, para esto es necesario configurar los siguientes módulos:

- Módulo de lectura

**Metodología:** Para leer la tarjeta SAM se deberá implementar la librería en la aplicación, para que el APICORE la lea y ofrezca los datos necesarios para iniciar los servicios, también se implementará la librería que detecta la NFC y la SIMCARD, estas librerías las ofrecerán los proveedores de Recaudos Bogotá.

- Módulo de instalación para inicialización

**Metodología:** Una vez instalada las librerías se procederá a programar este módulo que confirmará existencia de las tarjetas y pedirá los datos para la inicialización de la aplicación.

- Módulo de configuración

**Metodología:** este módulo contendrá al módulo anterior, se programará para que se active al acercar una tarjeta especial que solamente la portará el personal autorizado, este módulo se encargará de pedir datos de inicio a la aplicación, también mostrará el saldo de la SAM, servirá para configurar un cambio de contraseña para la inicialización si se desea, implementará la función de bloqueo para que se desbloquee al poner en frente la tarjeta por segunda vez, también servirá para hacer un Backup de la información procesada.

**Objetivo 3:** Desarrollar el consumo de servicios para realizar la recarga web.

- Programar los servicios dados por el servidor Recaudos Bogotá.

**Metodología:** ya que el cliente proveerá el servidor en el cual se harán las consultas por medio de peticiones en formato JSON, se deberá programar cada servicio de manera lógica y secuencial, para esto empleará el servicio Auth que se encargará de autenticar que el dispositivo y la tarjeta SAM sean válidos, por medio de unas credenciales las cuales las tendrán el personal técnico de recaudos Bogotá, para esto enviara estos datos hacia el servidor por medio de este servicio, el servidor respondiendo con un Token de acceso, que es una clave para continuar al siguiente servicio, permitirá validar la autenticación para así configurar el servicio Inquiry, este hará otra consulta



enviando el Token de acceso recibido del servicio anterior para obtener la información de la tarjeta Tullave, la cual obtendrá valores propios de la tarjeta, con esta información el servicio Init validará si se tiene una recarga pendiente o no, recibiendo un código para iniciar el proceso de recarga. Si este proceso se lleva a cabo con éxito, el servicio Success será ejecutado respondiendo al servidor sobre el estado para que no recargue de nuevo con ese código, por lo que se deberá programar este servicio en la misma clase en la cual se configurara los servicios anteriores.

Si pasa algún error durante la transacción se ejecutará el servicio Cancel que avisará que ha fallado la transacción.

- Configurar manejo de mensaje de usuario

**Metodología:** Cuando se consuma los servicios se obtendrán respuestas por causa de cada petición, es por ello que es necesario validarlas para establecer en que parte del proceso se encuentra la transacción, por tanto, se deberá mostrar al usuario cómo va el estado del proceso, por lo que se diseñarán mensajes que vayan acorde al estado, para que sean mostrados en la pantalla que visualiza el usuario.

- Realizar integración de los servicios involucrados

**Metodología:** se probará la aplicación para comprobar la implementación de cada servicio y su correcto funcionamiento con el fin de llevarlo a producción.

Con estas actividades se logra obtener la recarga exitosa en la tarjeta NFC desde la web, implementando la aplicación en el dispositivo móvil, estos módulos serán desarrollados, programados e implementados por medio del software Android Studio.



Figura 4. Flujo para recarga exitosa.

Como se aprecia en la figura 5. Para llevar a cabo la recarga de la tarjeta Tullave, primero se debe realizar el pago a través de PSE o TuyaPay, después de confirmar el monto recargado, en cualquier momento se puede acercar alguna estación del SITP cuando se necesite tomar el servicio, se busca el dispositivo NEWPOS K2, se aproxima la tarjeta, el dispositivo la detecta para que a través del APICORE escriba y lea su información, la aplicación a desarrollar consulta el estado de la tarjeta Tullave comunicándose con el servidor de Recaudos Bogotá el cual responde de acuerdo al servicio que se lleva a cabo en el diagrama mostrado en la figura 3. Para esto hay una comunicación por medio del protocolo JSON entre el servidor y el aplicativo móvil.

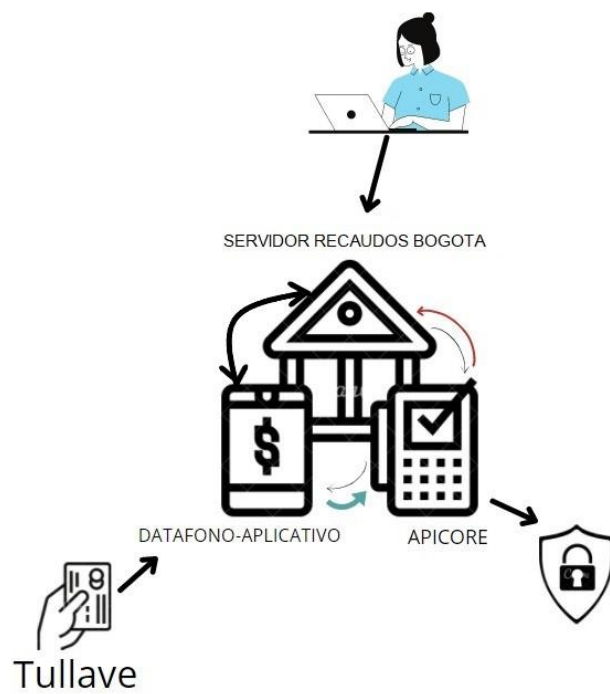


Figura 5. Partes del aplicativo.

## 8.1 Cronograma

Tabla 1. CRONOGRAMA DE ACTIVIDADES

N°	MES	ABRIL		MAYO			JUNIO			JULIO		
	ACTIVIDAD	SEMANAS										
		3	4	1	2	3	4	1	2	3	4	1
1	Diseñar las interfaces graficas que contiene la aplicación para recaudos Bogotá											
	Análisis de requerimientos	X	X									
	Diseño de solución		X									
2	Implementar la conexión entre el aplicativo de recargas web con el APICORE											
	Módulo de lectura			X								
	Módulo de instalación para inicialización			X								
	Módulo de configuración				X	X						
3	Desarrollar el consumo de servicios para realizar la recarga web.											
	Implementar servicio de Auth					X	X					
	Implementar servicio de Inquiry						X					
	Implementar servicio de Init							X				
	Implementar servicio de Succes								X			
	Implementar servicio de Cancel									X		
	Configurar manejo de mensaje de usuario									X	X	
	Realizar integración de los servicios involucrados									X	X	X

PLANEADO 

EJECUTADO 

## 8.2 Presupuesto

### *Gasto Global*

En esta sección se exponen los precios de manera porcentual indicando la financiación del proyecto por las diferentes fuentes de ingreso.

*Tabla 2. PRESUPUESTO GLOBAL DEL PROYECTO POR FUENTES DE FINANCIACIÓN.*

ITEM	ACTIVIDAD	CONTRAPARTIDA		TOTAL
		WPOSS	UFPS	
1	Gastos de Personal	\$8'480.000	\$1'600.000	\$10'080.000
2	Gastos de Equipos	\$2'021.200	0	\$2'021.200
3	Gasto de Materiales y suministros	\$920.000	0	\$920.000
4	Gastos de Infraestructura	0	0	0
5	Gastos de Servicio técnico	0	0	0
6	Gastos Administrativos	\$7.000	0	\$7.000
7	Imprevistos	\$1'302.820	0	
Subtotal		\$13'028.200	0	\$13'028.200
<b>TOTAL</b>				<b>\$ 14'331.020</b>

**NOTA:** Los imprevistos representan el 10 % del subtotal acumulado desde el rubro de personal hasta administrativo. En caso de presentarse algún imprevisto en la realización del proyecto, el gasto correrá por partes iguales tanto para la contrapartida como para la fuente externa.

*Gastos De Personal*

Tabla 3. DESCRIPCIÓN DE LOS GASTOS DE PERSONAL (EN MILES DE \$).

PERSONAL	FUNCIÓN DENTRO DEL PROYECTO	DEDICACIÓN HORAS/ SEMANA	\$/h	# de meses	FUENTES		TOTAL
					WPOSS	UFPS	
Moisés Enrique Villamizar Caballero	Pasante	46	\$5.000	3	\$3'680.000	0	\$3'680.000
Valeria Salazar	Pruebas QA	46	\$20.000	1	\$920.000		\$920.000
Fernando Romero	Ingeniero Desarrollador	46	\$15.000	2	1'840.000		\$1'840.000
Elmer Alejandro Parada	Director	4	\$25.000	4	0	\$1'600.000	\$1'600.000
Jair Angel Nova Acevedo	Codirector	20	\$15.000	4	\$4'800.000	0	\$4'800.000
<b>TOTAL</b>					<b>\$11'240.000</b>	<b>\$1'600.000</b>	<b>\$12'840.000</b>

**Gastos De Equipos**

Tabla 4. DESCRIPCIÓN DE LOS EQUIPOS DE USO PROPIO (EN MILES DE \$).

DESCRIPCIÓN	\$/h	HORAS DE USO	JUSTIFICACIÓN	FUENTES		TOTAL
				WPOSS	UFPS	
Computador Portátil	\$5.000	326	Necesario para el desarrollo del aplicativo móvil	\$1'630.000	0	\$1'630.000
Cargador de Baterías	\$200	326	Necesario para el reabastecimiento de energía de las baterías	\$65.200	0	\$65.200
Monitor	\$1.000	326	Necesario para la buena visualización a la hora de desarrollar el aplicativo	\$326.000	0	\$326.000
<b>TOTAL</b>				<b>\$2'021.200</b>		<b>\$2'021.200</b>

**GASTOS DE MATERIALES**

Tabla 5. DESCRIPCIÓN DE MATERIALES Y SUMINISTROS

ITEM	MATERIALES Y SUMINISTROS	CANTIDAD	FUENTES DE FINANCIACIÓN		TOTAL
			CONTRAPARTIDA		
			WPOSS	UFPS	
1	Datafono Newposs k2	3	\$2'700.000	0	\$2'700.000
2	Tarjeta NFC	4	\$20.000	0	\$20.000
3	SIMCARD	3	\$15.000	0	\$15.000
4	Tarjeta SAM	3	\$30.000	0	\$30.000
Subtotal			<b>\$2'765.000</b>	0	<b>\$2'765.000</b>
Total					<b>\$2'765.000</b>



## 9. Resultados

Para el desarrollo de este proyecto fue necesario ejecutar las etapas presentadas en el siguiente diagrama de flujo:

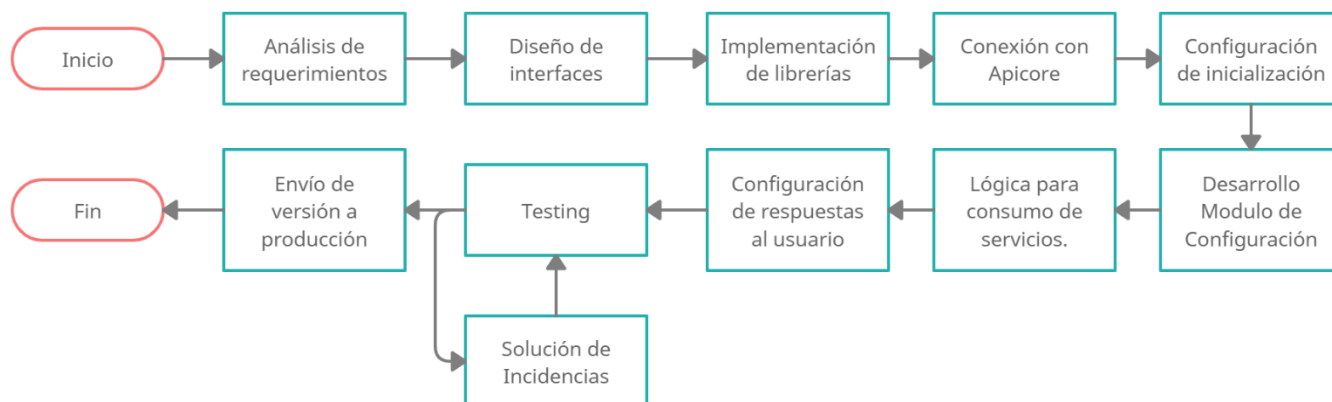


Figura 6. Diagrama de flujo del proyecto.

### 9.1 Análisis de requerimientos

Se requiere que desde la aplicación se detecte la tarjeta Tullave, cuando la detecte, el Apicore extraiga la información de la misma, envíe los datos hacia la aplicación para que pueda desarrollar el diagrama ilustrado en la figura 3. Por consiguiente, se debe abrir y cerrar el sensor NFC desde la aplicación. Como guía para controlar este sensor se toma el ejemplo a partir de una aplicación que programó la compañía fabricante del K2 que modela cómo utilizar los distintos sensores del dispositivo, este aplicativo Demo contiene el código con el cual se inicia el desarrollo.

La aplicación debe mantener bloqueada la pantalla mostrando un mensaje que diga “acerque y no retire la tarjeta del lector” en espera de que la persona la acerque al sensor, cuando sea posicionada, debe mostrar una barra de progreso circular indicando el estado de la transacción con el porcentaje de incremento y el mensaje que indica en que parte de la secuencia se encuentra,

dependiendo de la respuesta que traiga el servidor debe mostrar un cuadro de mensaje informativo que vea el usuario indicando en que finalizo la transacción.

Al finalizar, deberá volver a mostrar la pantalla inicial para repetir el proceso.

## 9.2 Diseño de interfaces

Utilizando la herramienta para desarrollo de aplicaciones móviles Android Studio, desde la sesión de layout, se diseña con base en modelos y temáticas aportadas por la empresa Recaudos Bogotá, las siguientes interfaces, a continuación, la pantalla de inicialización con la cual indica insertar una clave para iniciar la app.



Figura 7. Inicio Clave.

La clave y los parámetros de inicio los posee los operadores de recaudos Bogotá, una vez digitada, al oprimir ingresar, continuará a la siguiente interfaz:



Figura 8. Parámetros inicialización.

Ingresados los parámetros, al inicializar muestra la siguiente interfaz:



Figura 9. Pantalla de bloqueo.

Al acercarse la tarjeta se lanza la siguiente pantalla que contiene la barra de progreso circular con dos textos que indican el estado actual de la transacción.



Figura 10. Barra de progreso.

Cuando finalice el proceso, en la pantalla del K2 muestra el siguiente cuadro informativo que indica el mensaje de respuesta en la transacción:



Figura 11. Mensaje informativo.

Para asegurar que no pueda ser manipulado indebidamente el dispositivo K2 por parte de los usuarios, se le implementa un menú del cual solo puede ser accedido por personal autorizado, este menú es visible al posicionar una tarjeta de configuración como la mostrada en la siguiente figura:



Figura 12. Tarjeta de configuración y de usuario.

Al acercar la tarjeta de Control de configuración, después de que incremente un porcentaje de 16% la barra de progreso mostrada anteriormente en la Figura 9. La aplicación muestra el siguiente menú, que puede ser manipulado por la persona que este en frente, este menú contiene distintas opciones con las cuales se puede interactuar:



Figura 13. Menú.

Algunas opciones son accesos directos hacia distintas partes del software del dispositivo, otras opciones abren otras interfaces como las siguientes:



Figura 14. Interfaces del menú.

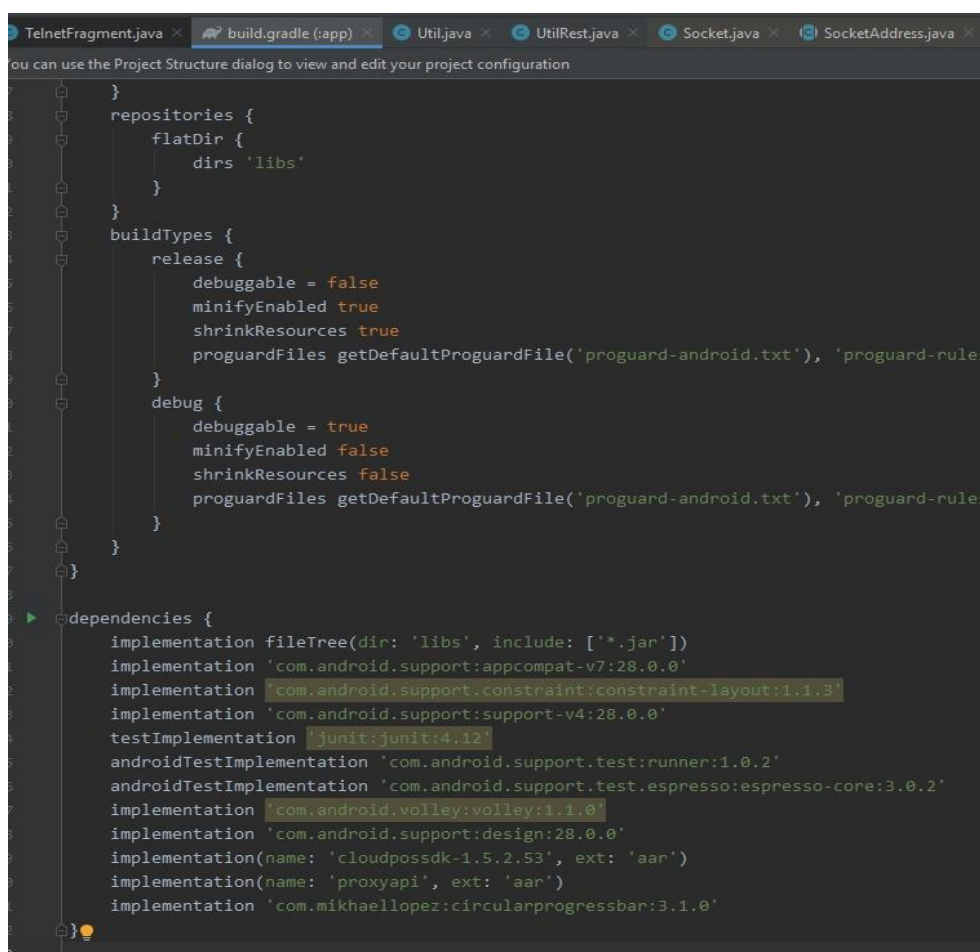
### 9.3 Implementación de librerías

Para el desarrollo de la aplicación Recargas Bogotá fue necesario implementar algunas librerías ya que con estas se puede extender las funcionalidades del aplicativo, también sirve para acceder a los periféricos del dispositivo, integrar herramientas que otras compañías tienen disponibles con el fin de dar crecimiento a un proyecto.

En una aplicación Mobile de Android Studio, se puede implementar librerías de la siguiente manera en una carpeta del proyecto llamada “RecargasBogota/app/libs” se agregaron las librerías que se implementan en la aplicación, y en el Gradle en la sesión de dependencias se incluyeron para que en el proyecto pueda dar uso de sus funciones, estas librerías tienen un formato .jar y .aar comúnmente, las librerías usadas en este proyecto son las siguientes:

**proxyapi.aar:** Esta es la que permite acceder a la aplicación Apicore, comunicándose con ella a través de métodos o funciones que se pueden llamar desde cualquier parte del proyecto intercambiando información con ese aplicativo.

**Cloudpossdk.aar:** esta librería permite acceder a los componentes del Hardware y software del dispositivo, mediante llamados a interfaces en las que se puede enviar información o mensajes hacia el sistema del pos, porque no es posible manipular directamente los procesos del sistema desde una aplicación, sino que a través de la comunicación que este archivo ofrece con el fin de que ejecute los comandos a los que se llamen desde el aplicativo.



```

}
repositories {
    flatDir {
        dirs 'libs'
    }
}
buildTypes {
    release {
        debuggable = false
        minifyEnabled true
        shrinkResources true
        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
    }
    debug {
        debuggable = true
        minifyEnabled false
        shrinkResources false
        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
    }
}
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.android.support:support-v4:28.0.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    implementation 'com.android.volley:volley:1.1.0'
    implementation 'com.android.support:design:28.0.0'
    implementation(name: 'cloudpossdk-1.5.2.53', ext: 'aar')
    implementation(name: 'proxyapi', ext: 'aar')
    implementation 'com.mikhaellopez:circularprogressbar:3.1.0'
}

```

Figura 15. Librerías del Gradle.

## 9.4 Conexión con Apicore

Después de implementar adecuadamente las librerías, compilar el proyecto, se creó una clase java que contiene funciones con la que se instancia todos los métodos que ofrece la librería de Apicore , los principales métodos a los que se pueden acceder son los que permitieron enviar y recibir información entre las dos aplicaciones, como por ejemplo, InitApi() este método hace que en la interface se muestre una barra de progreso mientras se consulta más información, GetInfoCard() este permite traer el Id, saldo actual, y la capacidad de la tarjeta que esta siendo leída para que con esa información se pueda consultar al servidor si hay recarga pendiente para esa tarjeta, GetSamId() que permite leer el saldo e Id de la tarjeta Sam, RechargeApi(final int amount) esta función manda al Apicore a recargar la tarjeta con el monto que se envía como entero, StopApi() que manda a detener la operación del Apicore entre otros.

## 9.5 Configuración de inicialización

Para inicializar la aplicación, se diseña la siguiente secuencia:

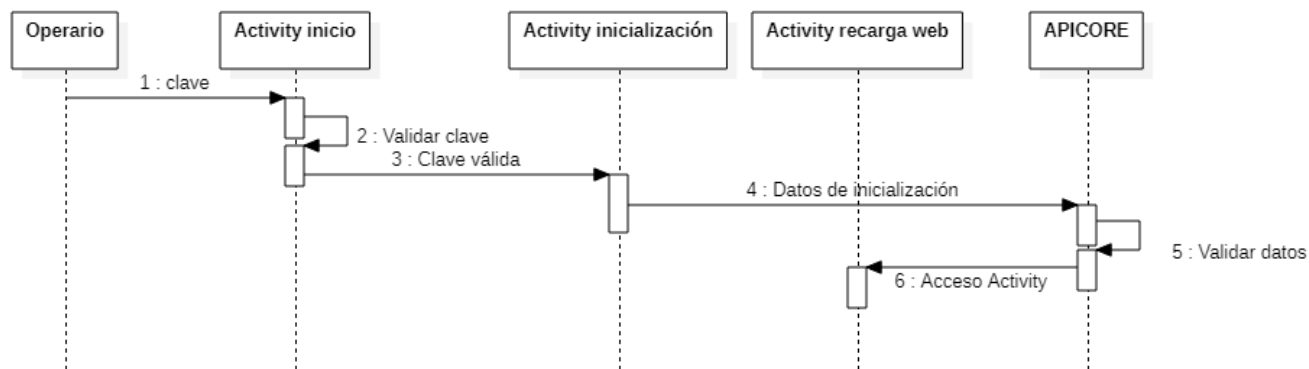


Figura 16. Diagrama secuencial

Después de haber instalado el APK en el dispositivo, al lanzarse la app, el operario debe digitar una clave, al oprimir INGRESAR la aplicación valida que sea la correcta, en caso de que



sea incorrecta muestra un mensaje el cual solicita la clave correcta, si la clave es correcta, ingresa a la pantalla de inicialización, en esta se debe ingresar los parámetros requeridos para que aplicación conecte con el servidor y con el Apicore, al ingresarlos y oprimir INICIALIZAR, el dispositivo valida la existencia de la SIM y de la SAM en sus respectivas ranuras.

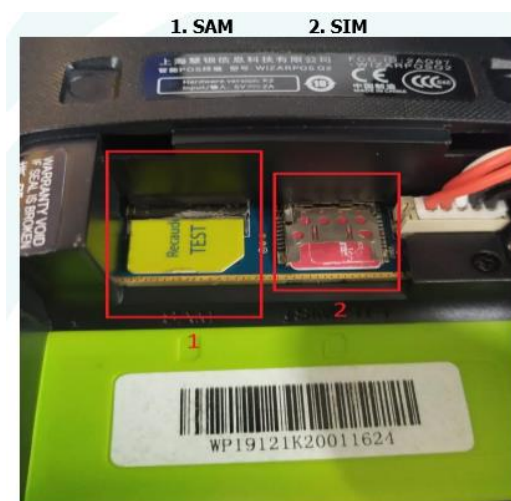


Figura 17. SIM y SAM.

Para poder acceder a la información de las diferentes tarjetas que puede leer el K2, se implementa código a partir de un demo el cual el fabricante del dispositivo entrega como guía a la empresa WPOSS y se toma como modelo para poder inicializar la app.

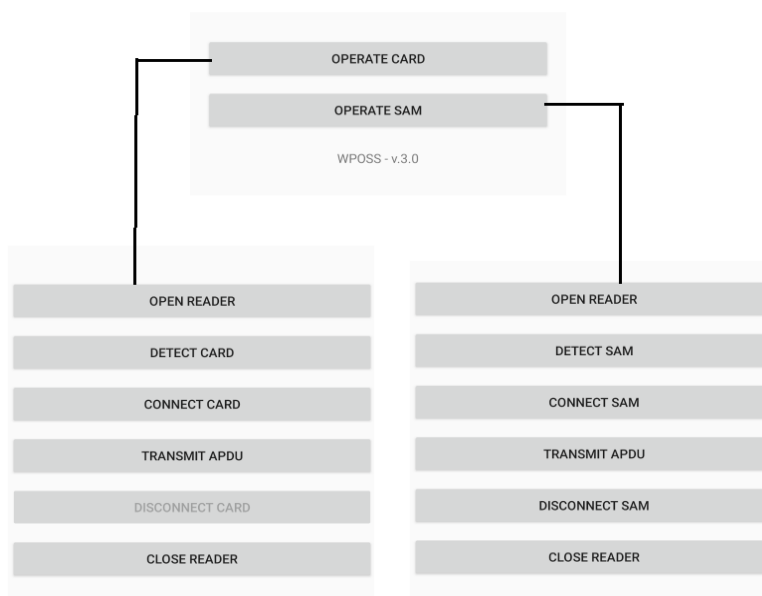


Figura 18. Demo.

Como se aprecia, este demo muestra como operar el sensor de la tarjeta NFC y SAM, abriendo, detectando, conectando, transmitiendo información a la tarjeta, desconectando y cerrando el sensor.

Se han evaluado los distintos casos que impiden inicializar la aplicación como lo son:

- No detectar y conectar con la SIM o la SAM
- Falla de lectura
- No hay internet
- Datos de inicialización erróneos
- Algún campo sin rellenar.

En caso de no inicializar, se muestra un mensaje indicando la causa de la no inicialización.

Una vez la aplicación valida que los datos de inicialización sean correctos, revisa que las tarjetas en sus respectivas ranuras, las lee y accede a internet para conectarse con el servidor de Recaudos Bogotá, enviándole la información con la cual inicializo el dispositivo, como este

servidor regresa un Token de acceso con el que la aplicación ingresa a la pantalla de inicio, la cual es la encargada de interactuar con el usuario.



Figura 19. Secuencia de gráficos.

## 9.6 Desarrollo de modulo para configuración

Para este módulo, se desarrolla desde una interfase un menú con el que al seleccionar alguna de las siguientes opciones, redirecciona a otra interfase que expande la información y funcionalidad del mismo.

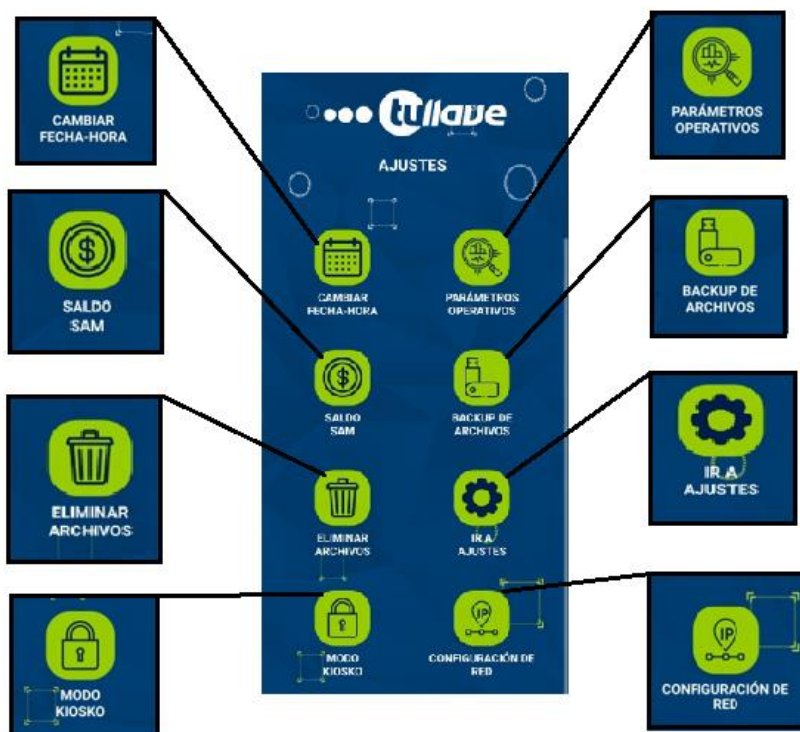


Figura 20. Módulo de configuración.

Como se explicó anteriormente, la única forma de acceder a este menú es por medio de la tarjeta de configuración que solo personal autorizado la porta.

Al Configurar cada opción se le da una funcionalidad, a continuación, se describe cada opción con su funcionalidad, sin embargo, algunas opciones son un acceso directo que direcciona a las configuraciones del sistema Android.

- **Cambiar fecha-hora**

Esta opción redirecciona hacia la configuración del sistema para poder modificar la hora y fecha del dispositivo.

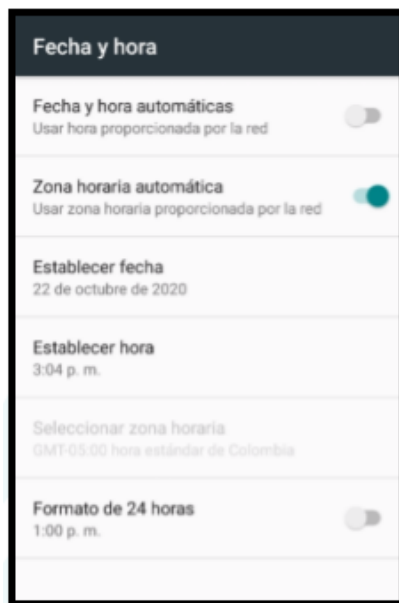


Figura 21. Configuración del sistema Fecha-Hora.

- **Parámetros operativos**

Esta opción sirve para visualizar los parámetros con los que funciona el dispositivo y la aplicación.



Figura 22. Interfase parámetros operativos.

- **Saldo SAM**

Esta opción direcciona a la interfase que visualiza el saldo disponible de la tarjeta SAM.



Figura 23. Interfase saldo SAM.

- **Backup de archivos**

Esta opción accede a la interfase que indica si desea realizar una copia del registro de transacciones que lleva hasta el momento el aplicativo, esta copia se hace en una USB.



Figura 24. Interfase Backup.

- **Eliminar archivos**

Generar un formateo de los archivos del aplicativo, dejando solo el archivo de configuración en su formato original.



Figura 25. Interfase eliminar archivos.

- **Ir a ajustes**

Esta opción accede a los ajustes del dispositivo.

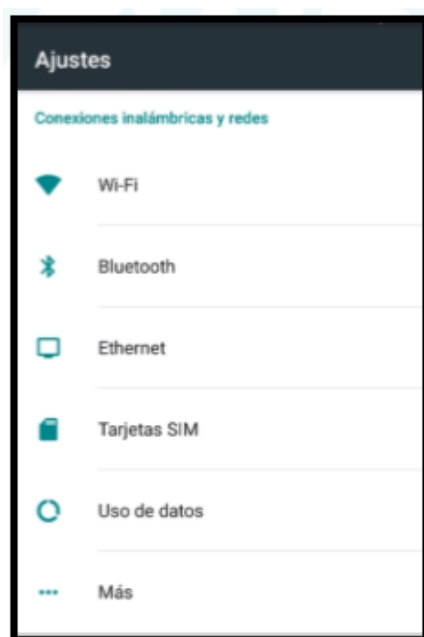


Figura 26. Ajustes del sistema.

- **Modo Kiosko**

Esta opción activa el modo kiosko en el POS, lo cual evitara que el usuario pueda salir de la aplicación.



Figura 27. Modo kiosko.

- **Configuración de red**

Esta opción permite cambiar los parámetros de red para reinicializar la aplicación.



Figura 28. Interfase parámetros de red.



## 9.7 Lógica para consumo de servicios

Para realizar la recarga a la tarjeta Tullave desde el dispositivo K2, como se mencionó antes, primero debe existir una recarga pendiente en el sistema, el servidor guarda esta recarga y está a la espera de que le soliciten autorización para que el nuevo saldo sea depositado en la tarjeta.

Cuando inicializa la app en el dispositivo, consume el servicio de autorización Auth, este permite acceder a los demás servicios, para acceder a este es necesario enviar un usuario y contraseña con el fin de que el servidor responda con un token de acceso, este token se envía como parámetro de autorización para consumir el siguiente servicio que es el Inquiry.

A continuación, se mostrará la secuencia para solicitar la verificación, autorización y confirmación al servidor para cada transacción en el aplicativo.

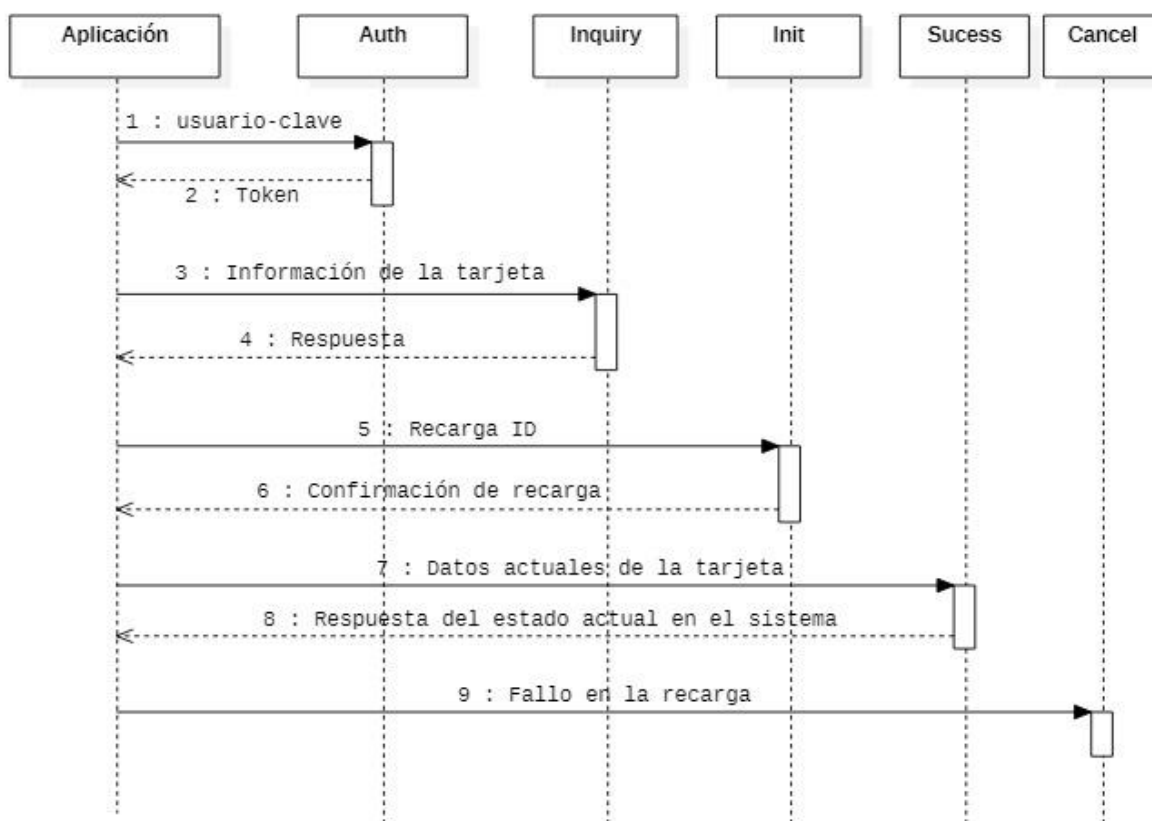


Figura 29. Secuencia de servicios.

El servicio Inquiry lo que hace es recibir los datos de la tarjeta Tullave con el fin de validar si hay una recarga disponible, después de validar, responde con un Id de recarga en caso de que haya una recarga pendiente, en caso de que no la haya o no sea posible de acceder a esa información, regresa un código o status con un mensaje especificando la respuesta.

Cuando la aplicación recibe el Id de la recarga disponible en el sistema, procede a realizar la recarga a la tarjeta, pero para eso, es necesario bloquear en el sistema la tarjeta para que no pueda ser accedida simultáneamente desde otra terminal, por esta causa, la aplicación consume el servicio de Init.

Al consumir el servicio Init, el servidor devuelve una respuesta especificando si se realizó satisfactoriamente la recarga, la respuesta consiste también en un código de estatus con un mensaje que describe si se ha bloqueado en el sistema la tarjeta, y si ya fue o no aplicada la recarga.

Cuando la aplicación Recargos Bogotá se asegura de que la recarga fue exitosa en la tarjeta, procede a consumir el servicio de Succes para que el sistema desbloquee la tarjeta, para esto, envía los datos actuales de la tarjeta como saldo actual, Id de la tarjeta, Id de la recarga, monto recargado, valor máximo y mínimo que soporta en saldo la tarjeta entre otros, con el fin de que el servidor actualice en su base de datos la información de ese usuario, y el servidor responde si se registró exitosamente la transacción o no.

En caso de que no se pudo realizar la recarga en la tarjeta, ya sea porque el servidor no lo permitió o porque el usuario quita la tarjeta del sensor del dispositivo, se calló el internet, o falla del hardware, la aplicación procede a consumir el servicio de Cancel, cuando la aplicación lo consume, envía el Id de la recarga que estaba siendo procesada y el motivo por el cual no se pudo realizar exitosamente la transacción, de esta manera en el sistema se desbloqueará la tarjeta para

que el usuario pueda intentar de nuevo esta operación.

## **9.8 Configuración de respuestas al usuario**

Cuando detecte la tarjeta NFC el Apicore extrae la información de la tarjeta como lo es el saldo, el rango máximo y mínimo que puede tener de saldo y la identificación de la tarjeta, comunicándose con la aplicación para que consulte con el servidor si es válida la tarjeta, si tiene recarga pendiente o no, en caso de que la tenga, recargue desde el saldo de la SAM, el valor recargado por el usuario previamente. En caso de que no tenga saldo pendiente, muestre al usuario el saldo de la tarjeta.

La secuencia que debe seguir la aplicación es la siguiente:

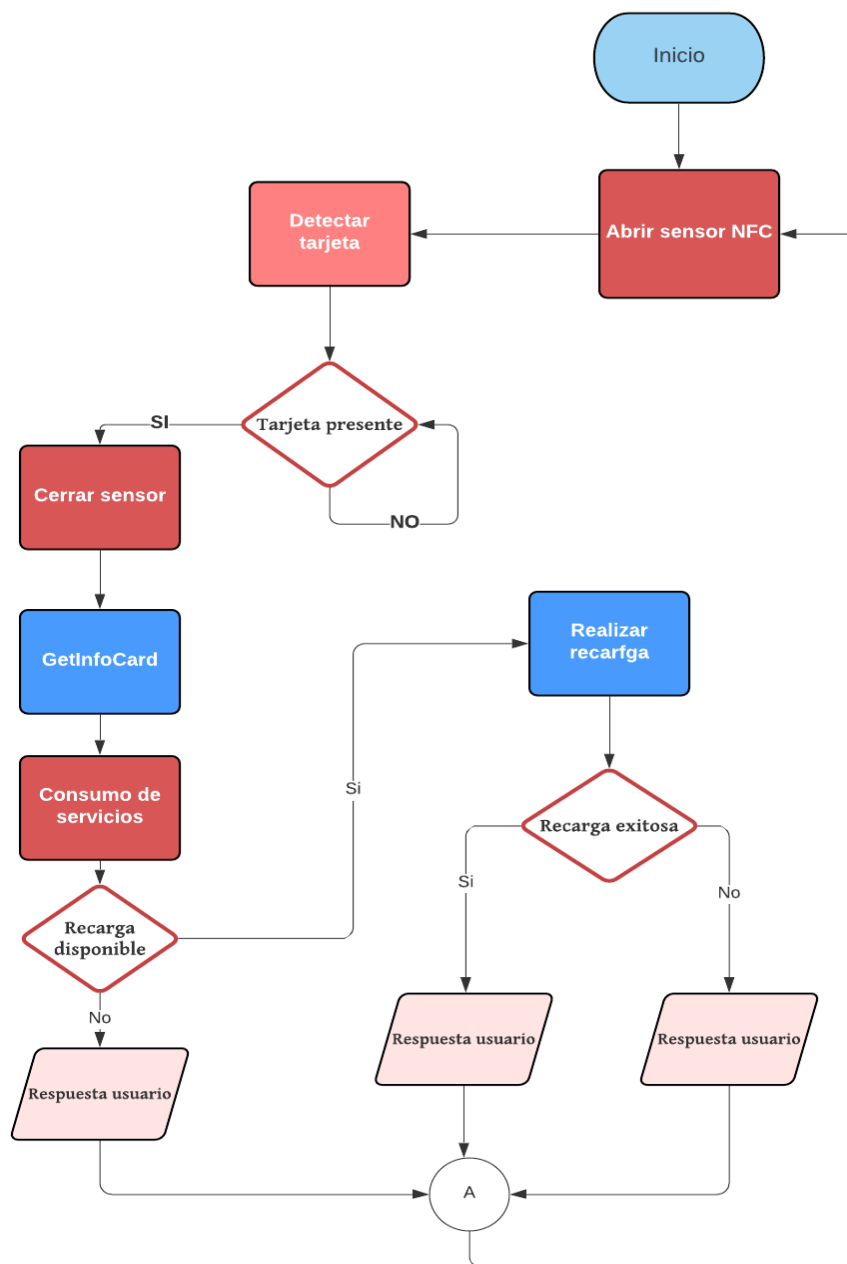


Figura 30. Secuencia de transacción.

Teniendo en cuenta la secuencia anterior y los mensajes programados en la aplicación, en la siguiente tabla se visualiza los mensajes preparados al usuario u operador de la terminal.

*Tabla 6. MENSAJES DE USUARIO.*

Causa	Mensaje
Cuando el servidor no está en servicio.	Servicio No Disponible\n en este momento
No hay repuesta del servidor.	Error confirmando pago
No se encontró recarga en el sistema para esa tarjeta.	No existe una recarga digital para su tarjeta.
Caída de internet.	Error de conexión
Se inhabilita el Pos desde el sistema.	El dispositivo no\n se encuentra en\n funcionamiento
Recarga realizada y verificada en el sistema.	Su recarga fue \n exitosa
Recarga no realizada.	Su transacción\n no fue exitosa
Fin de la transacción.	Retire su tarjeta
Tarjeta Sam sin saldo, se debe recargar desde el servidor	LSAM sin saldo
Cuando excede los límites de cupo permitido en la tarjeta NFC.	El valor de la recarga digital es superior al permitido en la tarjeta.
Falla en la lectura de la tarjeta NFC.	Acerque\n nuevamente su\n tarjeta
Está consultando al servidor.	Espere unos \n segundos
Para leer la información de la tarjeta Tullave.	Acerque su tarjeta\n al dispositivo
Consulta al servidor si son válidos los datos ingresados.	Cargando datos...
Cuando no inicializa la aplicación correctamente.	Error de Instalación, Reinicie Dispositivo
Datos incompletos.	Complete datos, por favor

No hay datos para la inicialización.	No existen datos de inicialización...
Para la pantalla de inicio.	Digite clave, por favor

## 9.9 Testing y solución de incidencias

Esta parte del proyecto permitió por medio de Testing encontrar errores o Bugs que no permitieron el correcto funcionamiento del aplicativo en el dispositivo, las pruebas se elaboraron manualmente y los errores encontrados son las siguientes:

*Tabla 7. TABLA DE PRUEBAS.*

Error	Causa	Solución
No Reiniciaba el dispositivo a las 4Am.	Mala implementación del reinicio desde la librería.	Se implemento un botón de prueba para que reinicie cada vez que se presione, cuando funcionó, se quitó y se programó desde la aplicación por medio de un subproceso, que consultara la hora del dispositivo para que en el lapso de 4:0Am a 4:02Am reiniciara el Pos.
Se cierra la aplicación durante la transacción.	Desborde de la memoria utilizada por el aplicativo.	Se implementó arquitectura Singleton para que en cada transacción no genere un subproceso o hilo más, sino

		que utilice el que ya se creó por primera vez.
Modo Kiosko no funciona.	Mala implementación del código desde la librería.	Se reprogramó esta función exitosamente.
Después de reiniciar el K2 no se lanza automáticamente la aplicación.	No programado esa función.	Se implementó un Broadcast receiver que permite estar a la escucha de las notificaciones del sistema para que cuando inicie notifique y se lance automáticamente la aplicación por medio de un Timer.
Se cierra la aplicación después de dos horas de reiniciar.	Falla en componente gráfico.	Se optimizó el circularbar o barra de progreso para que no tenga errores durante la actualización de la interface.
Se quita automáticamente el modo kioko cuando no está presente la SIM o la SAM.	Falta de validación por parte del aplicativo para revisar si está presente y se puede leer o no las tarjetas.	Se implementó validación para el módulo de inicialización con el fin de que cuando no esté presente alguna de las tarjetas o no se puedan leer, se quede en modo kioko activado hasta que un

		operario verifique las tarjetas y reinicie el dispositivo.
Las notificaciones interrumpen la aplicación.	Falta de configuración.	Se envía recomendaciones para implementar en cada dispositivo que se instalara el aplicativo para que se haga ajustes desde las configuraciones del sistema.

Como se mencionó en los beneficios, la implementación de este aplicativo logra reducir considerablemente el tiempo en el que se deposita la recarga en la tarjeta NFC Tullave, considerando que es un promedio la toma de tiempos ya que se realizó en distintas ocasiones, la siguiente tabla muestra los resultados promedio de esta prueba:



Tabla 8. PRUEBA DE TIEMPOS.

Metodos	Rangos	Tiempo por metodo
Tiempo abriendo y cerrando lector	16:38:19 9 a 16:38:20 4	1000 milisegundos
getInfoCard()	16:38:20 9 a 16:38:23 1	2500 milisegundos
CommApiProxi.getNumCard getCardIdApi(), api.lastCardID()	16:38:20 9 a 16:38:22 1	1500 milisegundos
getRechargeMin(): api.execCommand(command , parameters), api.getvalue();	16:38:22 1 a 16:38:22 6	1500 milisegundos
getRechargeMax(): api.execCommand(command , parameters), api.getvalue();	16:38:22 6 a 16:38:23 1	500 milisegundos
PaymentGateway	16:38:24 9 a 16:38:24 9	
recargarTarjeta api.recharge(amount);	16:38:24 9 a 16:38:29 6	5,500 milisegundos
setDataRecarga	16:38:29 6 a 16:38:29 6	
Tiempo total de la recarga	16:38:19 9 a 16:38:30 2	12,500 milisegundos

## 10. Conclusiones

Por medio de la documentación, requerimientos, diagramas de flujo del proyecto entregados por la empresa Recaudos Bogotá, se llevó a cabo una estrategia de organización y segmentación por etapas y objetivos, en la cual se integró metodologías ágiles para darle seguimiento al desarrollo del proyecto. Esto consiste en delegar funciones y tiempos a cada participante del equipo, esta estrategia de seguimiento, garantizo la integración de las dos aplicaciones y la comunicación con el servidor de Recaudos Bogotá; permitiendo la recarga de tarjetas Tullave desde la terminal Newpos K2 a través de los periféricos del dispositivo dando cumplimiento al objetivo principal de este proyecto.

En la etapa de Testing se pudo determinar diferentes errores gracias a la herramienta de Log.cat, la cual mostró con detalles e información el tipo de incidencia que obstaculizaba el desarrollo de la aplicación, con esta información con esta información fue posible la depuración del código.

Para el intercambio de información entre dos aplicaciones instaladas en un mismo dispositivo Android, fue necesario utilizar el mecanismo de comunicación Binder (IPC) ya que sirvió para intercomunicar procesos Android, esto permitió transmitir información entre los dos procesos por medio de memoria compartida a través del kernel de Linux.

La herramienta de software Postman permitió comprobar la comunicación con el servidor y verificar la secuencia propuesta en el presente documento, garantizando la correcta comunicación con el Backend.

Para el diseño de este aplicativo, fue importante tener en cuenta los recursos de la CPU del dispositivo, ya que la capacidad de procesamiento y la memoria es un factor determinante, fue necesario implementar la arquitectura Singleton, la cual consiste en instanciar por única vez un

proceso si no existe, esto permitió reutilizar el hilo creado por primera vez, para que en cada transacción no genere un nuevo subproceso, evitando la saturación del procesador y optimizando la aplicación.

Con la prueba de tiempo realizadas por QA en Wposs se pudo demostrar la reducción de tiempo en el cual se esperaba que la recarga fuera depositada en la tarjeta Tullave, el tiempo esperado era de 15 Seg, las pruebas demostraron un promedio de 12.5 Seg en este proceso.

## 11. Referencias

- Albau, A., & Riera, J. (1992). *Teleinformática y redes de computadores*. Marcombo S.A.
- Albiña Martínez, A. (2016). *Desarrollo de una guía para la implementación de aplicaciones basadas en NFC*. Universidad Politécnica de València, Facultad de Ingeniería.
- Campa Ruiz, A. (2011). *Desarrollo de una aplicación de pago a través de la tecnología NFC*. Universidad Carlos III de Madrid, Departamento de Ingeniería Telemática.
- Colombia, Congreso de la Republica. (1999). *Ley 527 de 1999: Por medio de la cual se define y reglamenta el acceso y uso de los mensajes de datos, del comercio electrónico y de las firmas digitales, y se establecen las entidades de certificación y se dictan otras disposiciones*. Bogota : Diario Oficial.
- Colombia. Congreso de la republica. (2008). *Sobre el Habeas Data Financiero*. Bogota: superintendencia de Industria y comercio.
- Gallardo, H. (2009). *Introducción al Testing de Sowane*. Academia.
- Hernandez Medina, D. C. (2016). *MODELO DE NEGOCIO B2C ALTERNATIVO PARA LA ADQUISICIÓN AL MEDIO DE PAGO DE LOS USUARIOS DEL SISTEMA INTEGRADO DE TRANSPORTE PÚBLICO CON TECNOLOGÍAS DE M-COMMERCE*. Universidad de la Sabana, Chia, facultad de Ingeniería.
- Hooda, A. (2019). *REST API Preguntas y respuestas de la entrevista: Automatización de API REST Preguntas y respuestas de la entrevista*. Amazon Digital Services LLC .
- Huidobro Moya, J. M., & Luque Ordoñez, J. (2014). *Comunicaciones por Radio. Tecnologías, redes y servicios de radiocomunicaciones*. Ra-Ma.
- Iván, T. B. (2017). *Programación imperativa con lenguaje C*. ECOE Ediciones.
- Jesús, M. G. (2010). *Escaneando la informática*. UOC.
- Lee J. Krajewski, L. P. (2013). *Administración de operaciones: estrategia y análisis*. Pearson Educacion.
- M., M. P. (2002). *Conceptos de sistemas operativos*. Amabar, S.L.
- Meyes, K. (2008). *Smart Cards, Tokens, Security and Applications*. Springer.
- Mojica Sanchez, I. F., & Andres, L. V. (2019). *Contraste de los Riesgos Valorados para los tipos de Tarjetas que han sido utilizadas como medio de pago en el sistema Integrado de Transporte Publico*. Universidad Catolica de Colombia, facultad de Ingenierias.
- Montero Miguel, R. (2012). *Desarrollo de aplicaciones para android*. Ra-Ma.
- Oficina de Análisis de Información y Estudios Estratégicos. (07 de Feb de 2019). <https://scj.gov.co>. Recuperado el 2021, de [https://scj.gov.co/sites/default/files/documentos\\_oaiee/PP3\\_\\_\\_TRANSMILENIO.pdf](https://scj.gov.co/sites/default/files/documentos_oaiee/PP3___TRANSMILENIO.pdf)
- Rodriguez Mayen, J. A. (2017). *DISEÑO DE DISPOSITIVO DE PAGO ELECTRÓNICO UTILIZANDO*

*TARJETAS NFC CON AUTENTICACIÓN BIOMÉTRICA* . Universidad de San Carlos de Guatemala, Facultad de Ingeniería.

Transmilenio S.A. (Febrero de 2021). *Estadísticas de oferta y demanda del Sistema Integrado de Transporte Público*. Obtenido de

<https://www.transmilenio.gov.co/publicaciones/152144/estadisticas-de-oferta-y-demanda-del-sistema-integrado-de-transporte-publico--sitp-febrero-2021/>

Uwe Hansmann, M. S. (2002). *Smart Card Application Development Using Java*. Springer.